

57424



YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ENDÜSTRİYEL ROBOTLARIN
BİLGİSAYAR YARDIMIYLA HAREKET
DENKLEMLERİNİN ELDE EDİLMESİ VE SİMÜLASYONU**

Makina Müh. Suat ARAPOĞLU

F.B.E. Makina Mühendisliği Anabilim Dalı Konstrüksiyon Programında

Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof.Dr.Faris KAYA

İSTANBUL - Eylül 1996

İÇİNDEKİLER

	<u>SAYFA NO</u>
TEŞEKKÜR	III
TÜRKÇE ÖZET	IV
İNGİLİZCE ÖZET	V
BÖLÜM 1	
1.1 Giriş	1
1.2 Robotun Tanımı	4
1.3 Endüstriyel Robotların Genel Yapısı	5
1.4 Endüstriyel Robotlar Üzerine Yapılmış Çalışmalar	8
1.5 Tezin Konusu	15
1.6 Tezin İçeriği	16
BÖLÜM 2	
2.1 Robot Kinematiği	17
2.2 Uzunlar, Mafsallar, Uzun Parametreleri, Mafsal Koordinatları	17
2.3 Uzun Eksen Takımları	19
2.4 Uzun Dönüşüm Matrisi	22
2.5 Robotlarda Konum Analizi	25
BÖLÜM 3	
3.1 Robot Dinamiği	28
3.2 Lagrange - Euler Formülasyonu	31
3.2.1 Robotun Kinetik Enerjisi	31
3.2.2 Robotun Potansiyel Enerjisi	37
3.3 Hareket Denklemleri	39

BÖLÜM 4	
4.1 Bilgisayar Programı	42
4.2 Matematiksel Model Programı	42
4.2.1 Alt Programlar	44
4.3 Simülasyon Programı	47
4.3.1 Alt Programlar	48
BÖLÜM 5	
5.1 Sonuçlar	50
5.2 Matematiksel Modelin Elde Edilmesi	50
5.3 Robot Hareketinin Simülasyonu	51
5.4 Gelecekte Yapılabilecek Çalışmalar	51
KAYNAKLAR	53
EKLER	57
ÖZGEÇMİŞ	

TEŞEKKÜR

Tezimin hazırlanmasından önce ve hazırlanması sırasında göstermiş olduğu yakın ilgi ve vermiş olduğu maddi - manevi destekten dolayı değerli hocam **Prof. Dr. Faris KAYA** ' ya en içten teşekkürlerimi sunarım .

Ayrıca , yine vermiş oldukları her türlü destekten dolayı anabilim dalı başkanımız , değerli hocam **Prof. Necati TAHRALI** ' ya ve arkadaşım **Arş. Gör. Ersun YALÇIN** ' a teşekkürlerimi borç bilirim.

İstanbul
Eylül 1996

Suat ARAPOĞLU
Makina Mühendisi

ÖZET

Endüstriyel robotlar ; sahip oldukları özelliklerden dolayı son otuz yıldır seri üretim hatları ve insan sağlığı açısından zararlı alanlar başta olmak üzere sanayide oldukça yaygın kullanım alanı bulmuşlardır . Bunun sonucu olarak endüstriyel robotların imali de büyük önem kazanmıştır .

Robotlardan mali ve üretim açısından maksimum verim alabilmek için robotun konstrüksiyon, kontrol ve iş yapabilme açısından çok iyi vasıflara sahip olması gerekmektedir .Bu nedenle robot imal edilmeden önce konstrüksiyon, kinematik, dinamik ve kontrol açısından çok iyi analiz edilmelidir . Bu analizi en kolay şekilde yapabilmek için robotun matematiksel modelinin elde edilmesi gerekmektedir .

Bu tezde robotun matematiksel modeli bilgisayar yardımıyla elde edilmeye çalışılırken aynı zamanda robotun yapacağı hareketler de simüle edilmeye çalışıldı. Dinamik analiz açısından sadece Lagrange - Euler Formülasyonu esas alındı.

Birinci kısımda kısaca robotlar hakkında bilgi verilirken ikinci kısımda robot kinematiği ve üçüncü kısımda da robot dinamiği incelendi. Dördüncü kısımda ise esas amaç olan bilgisayar programı incelendi. Son kısımda ise çalışma kısaca değerlendirildi.

ABSTRACT

Industrial robots, thanks to their special characteristics, have been widely used in industry for thirty years, especially on serial production lines and those sectors having harmful effects to human health. Therefore, the production of industrial robots have gained great importance, too.

In order to take maximum performance from robots with respect to financial and manufactural aspects, they must provide best quality in relation with their construction, control and performance capabilities. Therefore, before the production, a robot must be perfectly analyzed in respect of construction, kinematics, dynamics and control. And to analyse these aspects, it needs to build up the mathematical model of the robot.

In this study , it is tried to build up the mathematical model of the robot by the aid of a computer, beside this the possible motions of the robot is tried to be simulated. For the dynamic analysis, only the Lagrange - Euler Formulation is taken into consideration.

The first chapter gives a general information on robots ; the second and third chapters cover the kinematics and dynamics of the robots , respectively. The fourth chapter focuses on the computer program, which is the main subject of the study. Finally, a brief evaluation is made in the last chapter.

BÖLÜM 1

1.1 GİRİŞ

Teknolojinin başlangıcı insanlık tarihi kadar eskidir. İnsanođlu tarih boyunca, insanların ve hayvanların yerine kullanılmak üzere birçok mekanizmalar geliřtirmiş ve kullanmıştır. Avrupa Rönesansı esnasında birçok alanda kaydedilen ilerleme ile birlikte Bilimsel Arařtırma Metodlarının geliřtirilmesinde de ilerleme sağlanmış ve bu metodların üretim sahasında kullanılması sonucu 18.yy'da Sanayi Devrimi gerçeğeleşmiştir. Sanayi Devrimi ile birlikte üretimde, bilimde ve teknolojiye büyük bir ilerleme başlamıştır. Üretimin, bilimin ve teknolojinin birbirlerini etkilemeleri her üç alanda da büyük gelişmelerin meydana gelmesine sebebiyet vermiştir. Bu gelişmeler günümüzde Otomasyon Devriminin doğmasına yol açmıştır.

Bilim ve teknoloji sahalarında yaşanan gelişmelerin üretim sahasına yansması sonucu üretimde kullanılan araçlarda da büyük gelişmeler görülmeye başlanmıştır. İnsan, üretimi hem gücüyle hem de zekasıyla sağlar. Gücüyle üretir, zekasıyla kontrol eder. Buhar makinasının icadı ve elektriğin bulunmasından sonra bu iki güç kaynağı üretimde insan gücünün yerine kullanılmaya başlanmıştır. Bunlara paralel olarak kontrol tekniklerinin gelişmesi ve elektronikteki ilerlemeler de üretim hattının mekanik olarak daha etkili şekilde kontrol edilebilmesine imkan sağlamıştır.

Üretimin artmasıyla beraber özellikle seri üretim hatlarında seri üretim tekniklerine, sağlıklı bir ortamda yapılması gereken işler için de insan faktörünü en aza indiren üretim araçlarına ihtiyaç duyulmaya başlanmıştır. 1947 yılında transistörün icad edilmesi elektronikte bir sıçrama meydana getirmiş ve minikompütürlere yol açılmıştır. Minikompütürlerin mekanik sistemlerde kontrol amacıyla kullanılmaya başlaması ise robotların gündeme

gelmesini sağlamıştır. 1947 yılında elektrik gücüyle çalışan ilk teleoperatörün geliştirilmesiyle başlayan endüstriyel robot üretimi, günümüzde en büyük robot imalatçılarından olan Unimation Company'nin 1956 yılında kurulmasıyla başlı başına bir sektör haline gelmiştir.

Endüstriyel robotların sanayide çok geniş kullanım alanı bulması değişik işleri yapması arzulanan, farklı özelliklere sahip bir takım robotlara ihtiyaç duyulmasına sebep olmuştur. Bunun sonucu olarak da farklı özelliklere sahip ve farklı fonksiyonları yerine getirmesi istenen, çeşitli robotların dizayn ve imal edilmesi gündeme gelmiştir. Robotlar çok kompleks mekanik sistemler olduğundan dizayn aşamasında, imal edilecek robotu göz önünde canlandırmak imkansız derecede zordur. Bu nedenle robotların imalinin gündeme gelmesiyle birlikte, robotların dizaynını ve simülasyonunu yapabilecek çeşitli araçlara ihtiyaç duyulmuştur. İhtiyaç duyulan bu araçları karşılamak için robot modelleme ve simülasyonu konusunda yeni bir saha açılmış ve bu sahada birçok çalışmalar yapılmıştır. Robot modelleme ve simülasyonu sahasındaki gelişmeler, bilim adamlarının matematiksel ve fiziksel kanunları yeni metodlarla yorumlamaları sonucu sağlanmıştır.

Bir sistemi en iyi şekilde inceleyip analiz etmek için, o sistemi ilk önce imal etmek ve daha sonra o sistem üzerinde çalışmak gerekir. Bu ise oldukça pahalı ve meşakkatli bir yoldur. Fakat gerçek bir sistemin üzerinde çalışmaktan ziyade onun modelini elde edip onun üzerinde çalışmak daha kolay ve ucuzdur. Bir sistemin modelini elde etmek için kullanılacak en uygun yol, matematik ve fizik bilgilerinden yararlanarak o sistemin matematiksel modelini elde etmektir. Robotik alanında da robotların matematiksel modelini elde etmek için çeşitli çalışmalar yapılmış ve farklı metodlar geliştirilmiştir. Bugün, endüstriyel robotların matematiksel modelini elde etmek için en çok üç yöntem kullanılmaktadır. Bunlar :

1 - Lagrange - Euler (L - E) Formülasyonu : Lagrange dinamiğini esas alır ve kapalı formda fakat hızlı değerlendirilemeyen matematiksel model verir.

2 - Newton - Euler (N - E) Formülasyonu : Newton - Euler dinamiğini esas alır ve kapalı formda adım adım çözülmesi gereken yapıda fakat hızlı değerlendirilebilen matematiksel model verir.

3 - Genelleştirilmiş d'Alembert (G - D) Formülasyonu : d'Alembert prensibini esas alır ve hem kapalı formda, hem de N - E Formülasyonundaki kadar olmamakla beraber hızlı değerlendirilebilen matematiksel model verir.

Bu konuda halen araştırmalar devam etmektedir.

Endüstriyel robotların hareket denklemlerini elde edilmesinde, daha sonra görüleceği gibi oldukça fazla ve karışık matematiksel işlemlerin yapılması gerekmektedir. Bu işlemlerin elle yapılması durumunda hem oldukça fazla zaman harcanacak hem de insan faktöründen kaynaklanan hatalar oluşacaktır. Bu işlemleri hızlandırmak ve hatasız sonuç elde etmek için bilgisayarlardan yararlanma yoluna gidilmiş ve bu sahada da çeşitli çalışmalar yapılmıştır. Bunun sonucu ilk önemli bilgisayar programı REDUCE2 yapılmıştır (Hearn, 1977). Bu konuda yapılan çalışmalar sonucu daha etkin ve kısa sürede sonuç veren programlar (Megahed, 1984) gerçekleştirilmiştir.

Endüstriyel robotların hareket denklemlerinin elde edilmesindeki amaç; robotu analiz etmek ve robottan en etkin şekilde faydalanabilmek için gerekli parametreleri tesbit etmektir. Bunun için de robotun çeşitli parametreleri üzerinde yapılan değişikliklerin ne gibi sonuçlar verdiğini görmek gerekir. Bunu sağlamanın en uygun ve kolay yolu da robotu sanal ortamda görüntülemektir(Simülasyon).

Robot hareketinin bilgisayar ortamında simüle edilebilmesi için de çeşitli çalışmalar yapılmıştır. Bu çalışmaların sonucu olarak çeşitli paket programlar geliştirilmiştir. GEOMAP (Sata, Kimura, Amano - 1981), GRASP (Yong,Boney -

1983), ROFAC (Stauffer - 1984), CATIA (Dombre,Borrel,Liegeois - 1985) ve en son arařtırmalardan biri olan SMAPS (Gosselin, Pereault, Vaillancourt - 1995). Bunlardan bazılarıdır.

1.2 Robotun Tanımı

Robot sözcüğü. Çek oyun yazarı Karel Capek'in 1920'de yazdığı *Rossum's Universal Robots* adlı oyunundan gelmektedir ve Çek dilinde *Robota* sözcüğü işçi manasına gelmektedir.

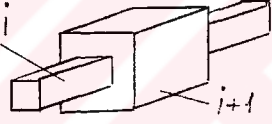

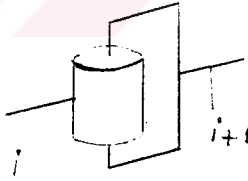
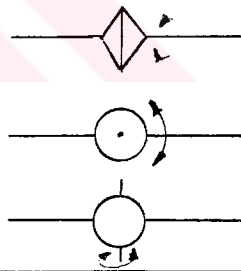
Endüstriyel robotların kullanım alanına girmesiyle beraber robotun ne olup ne olmayacağını ayırdedebilmek için tanımı yapılmıştır. Fakat bu tanım çeşitli ülke ve kuruluşlara göre değişmektedir. Robot için en kapsamlı tanımı JIRA (Japan Industrial Robot Association) yapmaktadır ve robotu, "bir operatör, programlanabilir elektronik kontrolör veya her hangi bir logic sistemle kontrol edilebilen, her hangi bir cismi tutmaya veya taşımaya yarayan ve birbirine mafsallarla bağlanmış veya birbiri üzerinde kayarak hareket eden mekanizmalardan oluşan bir manipölatör " olarak tarif eder. AFRI (Association Française de Robotique Industrielle) da JIRA gibi aynı tarifi kabul ederken, robotu diğer otomatik cihazlardan servo kontrollu olması şartıyla ayırır. Bununla beraber RIA (USA Robotic Industries Association) ve BRA (British Robot Association) daha sınırlı bir tanım getirirler ve "tekrar programlanabilir" olmayı şart koşarlar. Bu iki kuruluşa göre ise robot; "belirli bir görevi yapmak için geliştirilmiş, değişken , programlanmış hareketler yaparak bir malzemeyi taşıyan veya manipüle eden ve tekrar programlanabilen çok fonksiyonlu bir manipölatör " olarak tarif edilir.

1.3 Endüstriyel Robotların Genel Yapısı

Bir robot manipülatör, açık kinematik zincir içinde birbirine mafsallarla bağlanmış kollardan oluşur. Mafsallar genellikle döner (revolute) veya kayar (prizmatik) mafsal şeklinedirler.

Robot mafsallarının her birinde, ilgili uzvun bağlı hareketini sağlayan bir tahrik kaynağı bulunmaktadır. Bunlar uygulamada genellikle elektrik motoru olmaktadır. Fakat büyük yüklerin ve kirli çalışma ortamlarının söz konusu olması durumunda hidrolik tahrik elemanları da kullanılmaktadır.

TABLO 1 Robot Mafsalları

Adı	Sembol	Şekli	Kinematik Gösterilimi
Kayar (Prizmatik)	P		
Döner (Revolute)	R		

Mafsallar sabit uzva yakınlık sıralarına göre 1 den n e kadar numaralandırılır. 1 - 3, 4 - 6, 7 - n mafsallarının ana işlevleri farklı olduğundan üç ayrı grup altında toplanırlar.

TABLO 2 Robot Mafsallarının Gruplandırılması

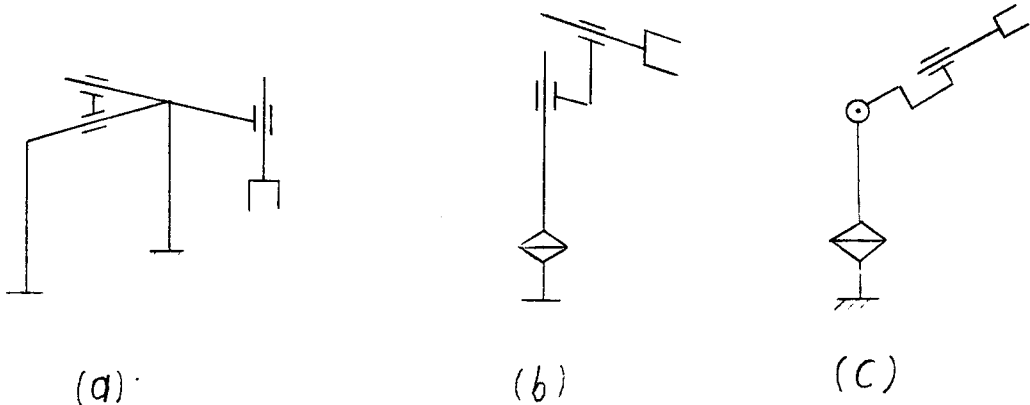
Mafsalsal No	Adı	İşlevi
1 - 3	Ana Mafsallar	Bileği konumlandırılması
4 - 6	İkincil Mafsallar	Taşınan cismin duruşunun ayarlanması
7 - n	Fazladan Mafsallar	Çalışma uzayındaki, engellerin aşılması

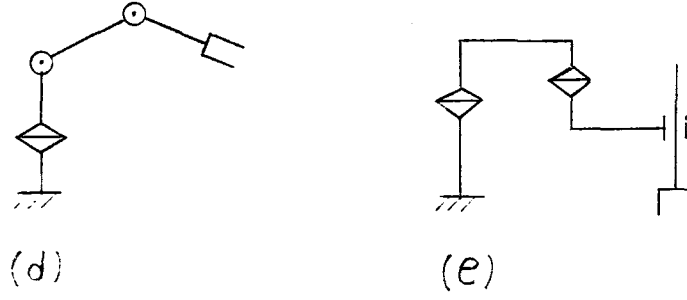
Bir robotun hareket özellikleri, geniş ölçüde ana mafsallarca belirlenir. Bu nedenle robotun ana mafsallarının bulunduğu kısım *temel robot* diye isimlendirilir. Mümkün ana mafsalsal düzenleri PPP, PPR, PRP, PRR, RPP, RPR, RRP, RRR şeklinde sekiz çeşittir. Bunların yalnız dördünden (PPP, RPP, RRP, RRR) uygun eksen seçimleriyle türetilmiş beş tip temel robot uygulanmakta ve buna bağlı olarak ta mevcut robotlar beş tip altında sınıflandırılmaktadır.

TABLO 3 Robotların Sınıflandırılması

Robot Tipi ve Mafsallar	Kartezyen Koordinat	Silindirik Koordinat	Küresel Koordinat	Düşey Eksenli	Eklemlilik
1	P	R	R	R	R
2	P	P	R	R	R
3	P	P	P	P	R

Bu robot tipleri Şekil 1 de gösterilmiştir.





(a) PPP - Kartezyen

(b) RPP - Silindirik

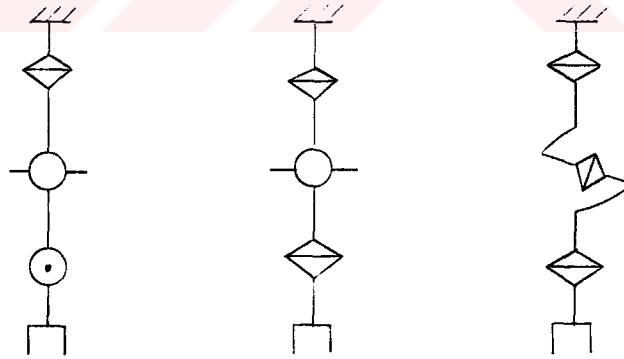
(c) RRP - Küresel

(d) RRR - Eklemlı

(e) RRP - Düşey Eksenli

Şekil 1.1 Robot Tipleri

Temel robotun uç noktası bir mafsallar kombinasyonundan oluşur ve bu mekanizma bilek olarak isimlendirilir. Bileğe, robotun iş yapan elemanı olan tutucu (el) bağlanır. Şekil 2 de en çok kullanılan bilek mekanizmaları gösterilmiştir.



Şekil 1.2 Bilek Mekanizmaları

Piyasada mevcut olan bir çok ticari robot Şekil 1.1 deki temel robotlardan birine Şekil 1.2 deki bilek mekanizmalarından biri üzerinden bir tutucu bağlanmasıyla oluşturulmuştur. Piyasada bulunan ticari robotların çoğu 4, 5 veya 6 mafsallıdır. Nadir olarak da 6 dan çok mafsallı robotlara rastlanır.

Robotlar ayrıca kontrol sistemine de sahiptir. Kontrol sistemi, robota ait sensörler tarafından elde edilen ve/veya dışarıdan bir operatör tarafından girilen bilgileri işleyen bir birimdir. Bu kontrol birimi üç ana fonksiyonu yerine getirir:

- a - Bilgiyi hafızada depolama, sıralama ve işleme
- b - Robot hareketini başlatma ve durdurma
- c - Dış çevreyle uyumlu çalışma

Günümüz robotlarında kontrol ünitesi olarak genellikle mikrokomputürler kullanılmaktadır.

1.4 Endüstriyel Robotlar Üzerine Yapılmış Çalışmalar

Robot teknolojisi bugünkü haline bir çok aşamalardan geçerek gelmiştir. Bugüne kadar robotik alanında yapılan çalışmalar tek bir alanla sınırlı kalmamıştır. Bir taraftan daha hafif, daha hızlı ve daha rijid konstrüktif yapıya sahip robotlar geliştirilmeye çalışılırken bir yandanda robotların modellenmesi ve simülasyonu konularında da çalışmalar yapılmıştır ve halen de yapılmaktadır. Robotların konstrüktif özellikleri bu tezin konusu dışında olduğundan bu alanda yapılmış çalışmalara değinilmemiştir.

Robotun matematiksel modelini sembolik formda bilgisayar yardımıyla elde edebilmek için ilk ciddi çalışmaların yetmişli yıllarda yapıldığını görmekteyiz. Bu yıllarda yapılan çalışmalar genellikle, sembolik formda analitik hesaplamalar yapan bilgisayar programları üzerine olmuştur. (Dilon , 1973), (Sturges , 1973), (Hearn, 1973), (Khalil , 1978) bu çalışmalardan bazılarıdır.

Hesaplamaların sembolik formda yapılması oldukça fazla aritmetik işlem gerektirirken, aritmetik işlemlerin yapılması esnasında kısaltma yapılamadığından oldukça uzun analitik ifadeler elde edilmekte ve bu da yüksek bilgisayar hafızası gerektirmektedir. Sembolik formda analitik hesaplama yapan bilgisayar programları Tablo 4 de verilmiştir.

Endüstriyel robotların kompleks yapılarından dolayı, bu sembolik hesaplama yapan programların kullanılmasıyla elde edilen matematiksel modeller kullanılmayacak kadar uzun olmaktadır. Bu olumsuzluğu aşmak için en önemli çalışmalardan biri Megahed tarafından 1984 de yapılmış ve bir takım programlar geliştirilmiştir. Megahed'in geliştirmiş olduğu, robot kollarının matematiksel modelini sembolik formda veren bilgisayar programları Tablo 5 de verilmiştir.



TABLO 4 Sembolik Hesaplama İçin En Önemli Bilgisayar Programları

Program	Özellik	Uygulama Alanı	Dil	Hafıza	Süre	Artık İşlem	Sonuç	Referans
REDUCE 2		Genel	LISP	Büyük	Çok Uzun	Çok	FORT.	Hearn, 1973
MACSYMA		Genel	LISP	Büyük	Çok Uzun	Çok	FORT.	Barr, Feigenbaum, 1982
TOAD	Teleoperatör Kol Dizaynı	Robotik	PL/1	Büyük	Çok Uzun	Çok	?	Sturges, 1973
OSSAM	Ohio State Sembolik ve Cebrik Manipülatörler	Robotik	LISP	Büyük	Uzun	Çok	?	Dilon, 1973
EDYLMA	Manipülatörlerin Dinamik Denklemleri	Robotik	PL/1	Büyük	Uzun	Çok	?	Khalil, 1978
EGAM	Eklemlmiş Mekanizmaların Denklemlerinin Elde Edilmesi	Robotik	PL/1	Büyük	Uzun	Çok	?	Zabala, 1978
DYMIR	Endüstriyel Robotların Dinamik Modeli	Robotik	LISP	Büyük	?	?	FORT.	Cesareo, Nicolò, Nicosia, 1981
DIRKIN	Homojen Dönüşüm Matrisi T ¹⁷	Robotik	?	?	?	?	*	Ardayfio et al, 1983
RIVKIN	Özel Ters Kinematik	Robotik	?	?	?	?	?	Ardayfio et al, 1983
MPLCS	Homojen Dönüşüm Matrisi T ¹⁷	Robotik	PL/1	Küçük	Kısa	Az	PL/1	Megahed, 1984a

TABLO 5 Robotlar İçin Sembolik Hesaplama Yapan En Önemli Bilgisayar Programları

Program	Özellik	Mafsal	Dil	Hafıza	Süre	Artık İşlem	Sonuç	Referans
MGLCS	Robot Kolunun Düz Kinematik Pozisyon Modelinin Elde Edilmesi	R/P	PL/1	Küçük	Çok Kısa	Az	PL/1	Megahed, 1984b
MDFLCS	Robot Kolunun Jacobiyen Matrisinin Elde Edilmesi ve Düz Kinematik Hızının Modellenmesi	R/P	PL/1	Küçük	Çok Kısa	Az	PL/1	Megahed, 1984c
MDLCS	Robot Kolunun Düz Dinamik Modelinin Elde Edilmesi	R/P	PL/1	Küçük	Çok Kısa	Az	PL/1	Megahed, 1984d
MGLCC	Robot Kolunun Dinamik Modelinin Elde Edilmesi	R/P	PL/1	Küçük	Çok Kısa	Az	PL/1	Megahed, 1984e
MDLAT	Robot Kolunun Dinamik Modelinin Elde Edilmesi	R/P	PL/1	Küçük	Çok Kısa	Az	PL/1	Megahed, 1984f

Endüstriyel robotların matematiksel modellerinin bilgisayar yardımıyla elde edilmesi için bir çok çalışma yapıldığı gibi robot hareketlerinin bilgisayarda simüle edilebilmesi için de çalışmalar yapılmış halen de yapılmaktadır. Bugün piyasada robot simülasyonu için bir çok paket program bulunmaktadır. En önemli simülasyon programları Tablo 6 da verilmiştir.

TABLO 6 Ticari Simülasyon Programları

Program	Referans	Şirket veya Araştırma Merkezi
ROBEX	Weck,Eversheim,Zuehlke	Aachen, Almanya
GEOMAP	Sata,Kimura,Amano, 1981	Tokyo, Japonya
GRASP	Yong,Bonney, 1983	Nottingham Üniver. , İngiltere
PLACE	Howie, 1984	McAuto Manufact. , Amerika
Robot-SIM	Stauffer, 1984	Calma Co. , Amerika
ROBOGRAPHIX	Stauffer, 1984	Computer Vision Co. , Amerika
AutoMod and Autogram	Stauffer, 1984	Auto Simulation, Amerika
I-GRIP	Stauffer, 1984	Deneb Co. , Amerika
RCODE	Stauffer, 1984	SRI International, Amerika
ROFAC	Stauffer, 1984	Science Management, Amerika
XPROBE	Summers, Grosmann, 1984	IBM Research Center
CATIA	Dombre,Borrel,Liegeois, 1985	LAMM, Fransa
SMAPS	Gosselin,Perreaut,Vaillancourt, 1995	Qubec Üniver. , Kanada

Bu programların en önemli beşi hakkında kısaca şu bilgiler verilebilir.
(Megahed , 1993)

CATIA : CATIA, robot hareketini simüle etme ve robot programlama için geliştirilmiş bir paket programdır. CATIA ile yapılan robot modelleme programlarının bütünü, her hangi bir robotun ve çevresinin geometrik modelini oluşturmada etkin bir araç olmaktadır. Robotun programlanmış hareketlerini ve çevresiyle olan ilişkilerini ekranda görüntülemek mümkündür. Bu özelliklerinin yanı sıra CATIA gerçek robotlarda da kullanılabilir.

GRASP: GRASP, bir çok farklı durumdaki değişik robot işleri için amaçlanmış genel bir paket programdır. GRASP, herhangi bir cismi basit şekillerin blokları halinde gösteren, basit gövde modelleme programı olarak düşünülmüştür. Robotun yapısı ve modeli Denavit - Hartenberg prensiplerine göre elde edilmektedir. GRASP, robot hareketini ters veya düz modda kontrol etmek için kullanılabilir.

XPROBE: XPROBE, Bir örnek vasıtasıyla robot programlama için geliştirilmiş bir deneysel sistemdir. XPROBE, hem metin yazarak hem de hareket sıralarını göstererek programlama yapabilen yeni bir yaklaşıma sahiptir. XPROBE'un en önemli özelliği, robota basit hissetme tekniklerinin öğretilmesine müsaade etmesidir.

PLACE: PLACE, robotun çalışma hücresinin geliştirilmesi ve dizayn edilmesi için geliştirilmiş bir paket programdır. PLACE, hücre dizaynı ve analizi için bir temeldir. PLACE kullanıcılar için dört özelliğe sahiptir. Bunlar; bilgisayar simülasyonu, kullanıcı kontrollu animasyon, dinamik üç boyutlu tablo kontrolü ve robotun bir kütüphanesini oluşturma. Bununla beraber kullanıcı program vasıtasıyla şu işlemleri de yapabilir:

- a - Hücre elemanlarını dinamik olarak pozisyonlama
- b - Robot kolun hareketini tanımlama
- c - Tutucunun ulaşma sınırlarını kontrol etme
- d - Hareket sırasını oluşturma
- e - Taşınan cismin gittiği yolu simüle etme
- f - Çevrim zamanını analiz etme

SMAPS: SMAPS, en son geliştirilmiş paket programlardan biridir(1995). Bu program sadece üç serbestlik dereceli paralel manipülatörler için geliştirilmiştir. Kullanıcının boyutsal değerleri girmesine müsaade eder ve bunları bir dosyada saklar. Bu değerler girildikten sonra manipülatör ekranda görüntülenip animasyon yapılabilir. Ters ve düz kinematik analiz yapabildiği gibi bunların bütün olasılıklarını verir. Sürekli modda çalıştırıldığı takdirde gerçek zamanda da simülasyon yapabilir. Manipülatörün tekillik analizini yaparak tekillik durumlarını da tesbit edebilir. Yine manipülatörün çalışma uzayını belirleyip bunu kartezyen uzayda gösterebilir. Sonuç olarak manipülatörün performansını gösterebilir.

1.5 Tezin Konusu

Önceki bölümden de görüldüğü gibi endüstriyel robotların matematiksel modellerinin elde edilmesi ve simülasyonu hakkında çeşitli çalışmalar yapılmış bulunmaktadır. Robotların matematiksel modelini sembolik formda elde etmek için daha önce yapılmış programlar, analitik ifadeleri oldukça uzun formda vermektedir. Bunun en önemli sebeplerinden biri de, bugüne kadar elde edilmeye çalışılan formüllerin bilgisayar ortamında kullanılmaya pek uygun olmayışıdır. Bu nedenle bu tezde, robotların matematiksel modelini elde etmek için Lagrange - Euler Formülasyonunun bilgisayar ortamında kullanılacak şekilde getirilmiş formülleri esas alınarak bir bilgisayar programı geliştirilmesi amaçlandı. Bunun sonucunda daha kısa formda sembolik analitik ifadeler elde edilmeye çalışıldı. Ayrıca programı 1 - 7 arası serbestlik derecesine sahip her tip robot için matematiksel model verebilecek bir yapıya sahip olmasını planladı.

Ayrıca robotun matematiksel ifadesini sembolik formda elde etmenin yanında robotun yapacağı hareketleri simüle edebilecek bir program geliştirilmesi hedeflendi. Bir önceki bölümde de görüldüğü gibi piyasada mevcut olan ticari paketler oldukça geniş kapsamlı ve bazı robot tipleriyle sınırlı. Bu tezde simülasyon programı için her hangi bir robot tipi sınırlaması olmaksızın 1 -7 arası serbestlik derecesine sahip her robotun hareketlerini simüle edilebilmesi amaçlandı.

1.6 Tezin İçeriđi

Tez beş bölümden oluşmaktadır. Birinci bölümde robot teknolojisini gelişmesi anlatılarak robotların önemi belirtilmekte ve endüstriyel robotların genel yapısı hakkında bilgi verilmektedir. Ayrıca bu sahada daha önce yapılmış çalışmalara geniş bir yer verilerek daha önceki çalışmalar anlatılmaktadır. Ayrıca böyle bir çalışmaya niçin gerek duyulduđu açıklanmaktadır.

İkinci bölümde ise robot kinematiđi hakkında bilgi verilerek, programlama esnasında kullanılacak formüllerin elde edilmesi için gerekli olan uzuv, mafsalsal ve bunlara ait parametreler hakkında bilgiler verilmekte ve uzuvlara takılacak eksen takımları açıklanmaktadır. Ayrıca homojen dönüşüm matrisleri hakkında bilgi verilmektedir. Bununla birlikte robot dönüşüm matrisinin elde ediliđi anlatılmaktadır.

Üçüncü bölümde de robot dinamiđi incelenmekte ve bu tezde esas alınan Lagrange - Euler Formülasyonu geniş bir şekilde açıklanmakta ve programlama esnasında kullanılacak formüllerin elde ediliđi gösterilmektedir. Ayrıca düz ve ters dinamik analize kısaca değinilmektedir.

Dördüncü bölümde ise bu tezin esas konusu olan, robotun matematiksel modelini sembolik formda analitik olarak verebilen bir programın ve ayrıca basit bir programlama ile D - H Tablosu oluşturulmuş bir robotun yapması istenilen hareketleri ekranda simüle edilebilecek bir çizim programının nasıl geliştirildiđi anlatılmaktadır.

Beşinci ve son bölümde ise yapılan çalışma hakkında kısa bir değerlendirme yapılmakta ve bundan sonra yapılabilecek çalışmalara ışık tutulmaya çalışılmaktadır.

BÖLÜM 2

2.1 ROBOT KİNEMATİĞİ

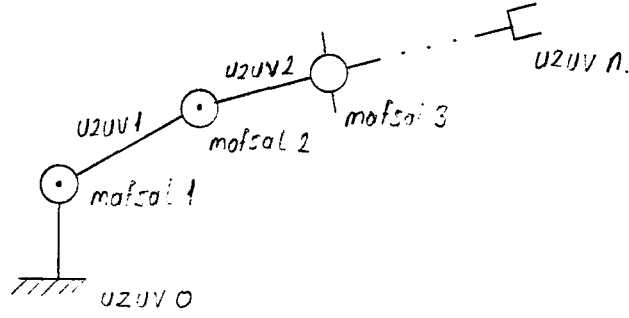
Bilindiği gibi kinematik, hareketi, hareketi meydana getiren kuvvetleri dikkate almaksızın inceleyen ilim dalıdır. Amacı, mekanik sistemlerin konum, hız ve ivmelerini incelemek ve bunlar arasındaki ilişkileri ortaya koymaktır.

Robotlar için ise bu amaç, robotun uzuvlarını oluşturan, rijid cisimlerin konum, hız ve ivmelerini incelemektir. Robot uzuvları arasında, bu uzuvların oluşturduğu açık kinematik zincirin açık ucunda yer alan ve robotun iş yapan elemanını oluşturan el, apayrı bir öneme sahiptir. Diğer uzuvlardan farklı olarak elin konum, hız ve ivmesi doğrudan doğruya robotun yapacağı iş ile bağlantılıdır. Bu nedenle bir çok durumda robot kinematığı denildiğinde sadece elin kinematığı anlaşılmaktadır.

Robot kinematığına ait incelemelerin yapı taşlarını rijid cisimler kinematığına ilişkin bilgiler oluşturur. Robot kinematığı, bu bilgilerin belli bir sistematik içerisinde değerlendirilmesinden ibarettir. (Turhan, 1994).

2.2 Uzuvlar , Mafsallar , Uzuv Parametreleri , Mafsal Koordinatları

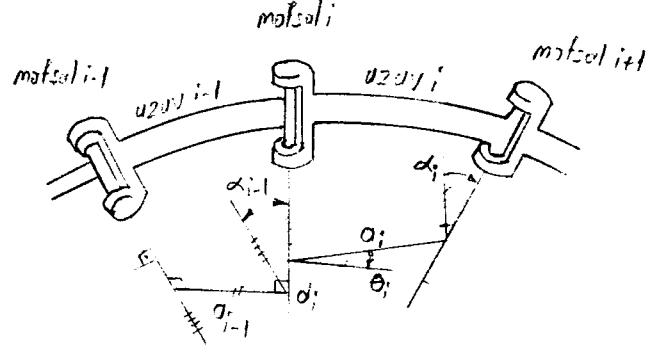
Robot uzuv ve mafsalları, kendilerine birer numara verilerek tanımlanırlar uzuvlar , sabit uzuv 0 olacak ve hareketli uzuvlar sabit uzva yakınlık sıralarına göre 1 den n ye kadar numaralandırılırken mafsallar, i. uzvu i-1. Uzva bağlayan mafsal i. mafsal olacak şekilde numaralandırılır. (Turhan ,1994) (Şekil 2.1)



Şekil 2.1 Robot Uzuve ve Mafsallarının Numaralandırılması

Robot uzuv ve mafsalları çok değişik geometrilere sahip olabilir ve bu geometriler ancak çok sayıda konstrüktif boyutun verilmesi ile tanımlanabilir. Ama uzuv ve mafsalların kinematik bakımdan tanımlanması için bütün konstrüktif boyutların verilmesine gerek yoktur. Her hangi bir uzvu ve üzerindeki mafsalı kinematik bakımdan tanımlamak için toplam dört kinematik parametrenin verilmesi yeterli olmaktadır. (Turhan, 1994). Bu dört adet kinematik parametrenin sistematik bir şekilde elde edilebilmesi geliştirilen çeşitli metodlardan biri de Denavit - Hartenberg Prensipleri' dir. (Denavit, Hartenberg, 1955). Bu tezde kullanılan formüllerin elde edilmesinde Denavit - Hartenberg Prensipleri esas alınmıştır. Aşağıda Denavit - Hartenberg parametreleri kısaca tanıtılmaktadır.

Mafsal eksen; döner mafsallarda dönme eksen, kayar mafsallarda ise kayma doğrultusuna paralel her hangi bir doğrunun oluşturduğu eksenidir. (Şekil 2.2). Bir robotta i . ve $i+1$. mafsal eksenlerinin ortak normal, kinematik bakımdan, robotun i . uzvunu temsil eder. Eksenlerin birbirine paralel olması durumunda ortak normal tek olarak belli olmayacağından keyfi olarak seçilir.



Şekil 2.2 Mafsali Eksenleri, Uzun Parametreleri, Mafsali Koordinatları

Şekil 2.2 de de görüldüğü gibi, i . ortak normalin a_i boyuna i . uzvun boyu, i ve $i+1$. mafsali eksenleri arasındaki, i . ortak normale dik düzlem içerisinde ölçülen α_i açısına ise i . uzvun burulma açısı denir ve bu iki parametre, kinematik bakımdan i . uzvun boyut ve şeklini tanımlamaya yeterli olur.

Öte yandan i . uzvun $i-1$. uzva göre konumu tanımlanmak istenirse, bu kez, $i-1$. ve i . ortak normallerin i . mafsali eksenini kestikleri noktaların d_i mesafesinin ve bu iki ortak normal arasındaki, i . mafsali eksenine dik düzlem içinde ölçülen θ_i açısının verilmesi yeterlidir. d_i ye i . mafsali boyu, θ_i ye ise i . mafsali açısı adı verilir ve bu iki parametre i . mafsali kinematik olarak tanımlar. (Turhan, 1994).

2.3 Uzun Eksen Takımları

Rijid cisimlerin kinematiğine ilişkin incelemelerde, cisme bağlı bir eksen takımına gerek duyulduğu bilinmektedir. Birer rijid cisim olan robot uzuvları içinde aynı şey söz konusudur.

Robot uzuvlarının her birine bir xyz kartezyen eksen takımı bağlanması esas olmakla birlikte bunun yapılış şeklinin tek olmayacağı bellidir. Bunu keyfilikten kurtarmak için bir sistematığe bağlanması gerekmektedir. Uzuvlara eksen

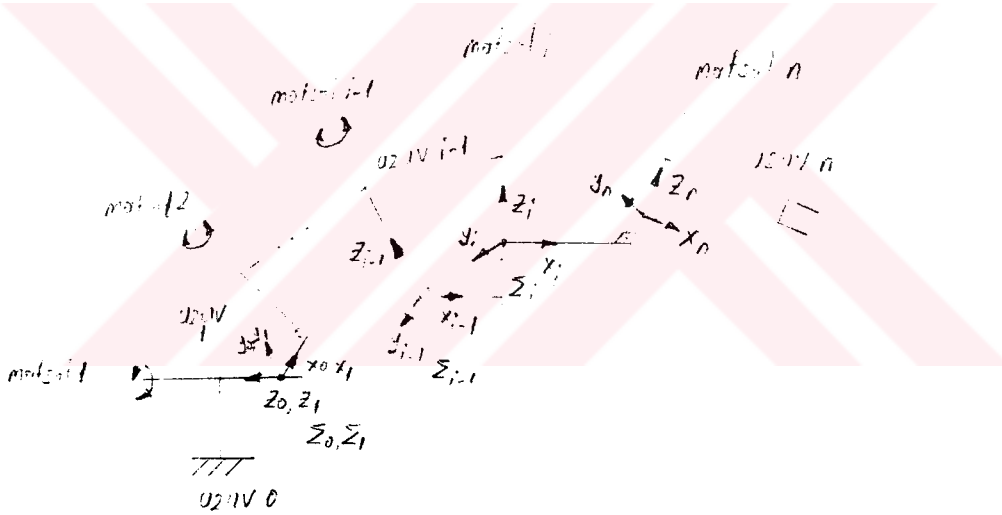
takımı bağlamakta uyulacak sistematığın unsurları aşağıda kısaca anlatılmaktadır.

A) i . uzva bağlı Σ_i eksen takımının orijini olarak i . ortak normalin i . mafsal eksenini kestiği nokta alınır. (Şekil 2.3).

B) Σ_i eksen takımının z_i ekseni i . mafsal eksen üzerinde alınır ve pozitif yönü robotun uç tarafına bakacak biçimde yönlendirilir. Uç tarafın yönü aşikar değilse yönlendirme keyfi yapılır.

C) Σ_i eksen takımının x_i ekseni i . ortak normal üzerinde ve i . mafsaldan $i+1$. mafsala yönelik olarak alınır.

D) Son olarak Σ_i nin y_i ekseni, Σ_i bir sağ eksen takımı oluşturacak şekilde seçilir.



Şekil 2.3 Uzuvlara Eksen Takımlarının Bağlanması

Yukarıda verilen kuralların uygulanmasıyla 1 den n-1. uzva kadar olan uzuvlara eksen takımlarının yerleştirilmesi mümkündür. Geriye, sabit uzva bir Σ_0 eksen takımının ve n. uzva bir Σ_n eksen takımının yerleştirilmesi kalmaktadır.

E) Σ_0 eksen takımı 1. uzuv, referans konumu olması isteniyorsa Σ_1 eksen takımıyla çakışacak şekilde seçilir.

F) Σ_n eksen takımı ise, orijini n-1. ortak normalin n. mafsal eksenini kestiği noktada olmak üzere, z_n eksenini, n. mafsal eksenini üzerinde ve robotun ucuna bakan yönü pozitif olacak şekilde, x_n eksenini n. uzuv arzu edilen referans konumunda iken x_{n-1} ile çakışacak şekilde ve y_n eksenini Σ_n eksen takımını sağ eksen takımı yapacak şekilde seçilerek oluşturulur.

Burada verilen sistematığe göre D - H Parametreleri de yeni tanımlara kavuşmuş olacaktır.

a_i : z_i den z_{i+1} e x_i eksenini boyunca ölçülen uzaklık

α_i : z_i den z_{i+1} e x_i eksenini çevresinde ölçülen açı

d_i : x_{i-1} den x_i ye z_i eksenini boyunca ölçülen uzaklık

θ_i : x_{i-1} den x_i ye z_i eksenini çevresinde ölçülen açı

Robot kinematığıne ilişkin hesapların gerçekleştirilmesi sırasında bu kinematik parametrelerde ne kadar çoğu sıfır olursa hesaplar o kadar kolaylaşmaktadır. Bu bakımdan eksen takımlarının seçilmesi sırasında en çok parametreyi sıfır yapacak şekilde seçimler yapılmaya çalışılmalıdır. (Turhan, 1994).

2.4 Uzuv Dönüşüm Matrisi

Ardışık $i-1$. ve i . robot uzuvlarına bağlı Σ_{i-1} ve Σ_i eksen takımları birbirlerine göre hem dönmüş hem de ötelenmiş eksen takımlarıdır. Böyle iki eksen takımı arasında homojen dönüşümler ve bu dönüşümlere ilişkin homojen dönüşüm matrisleri söz konusudur.

Bilindiği gibi homojen dönüşüm matrisi

$$H = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ f_{1 \times 3} & s_{1 \times 1} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{Döndürme} & \text{Öteleme} \\ \text{Perspektif} & \text{Ölçek} \end{bmatrix} \quad (2.1)$$

şeklinde ifade edilen 4×4 bir matristir. Robotlar genelinde perspektif vektörü 0 vektör kabul edilirken ölçek skaleri de 1 kabul edilir.

Buna göre Σ_{i-1} ve Σ_i eksen takımları arasındaki homojen dönüşüm matrisinin elde edilmesi problemi ele alınırsa bu problem Denavit - Hartenberg Parametreleri yardımıyla çözülebilecektir. Bunun için bölüm 2.3 deki tanımlar hatırlanır ve bunlar homojen dönüşüm matrisinde yerine konursa şu ifadeler elde edilir:

Birinci adım x_{i-1} boyunca a_{i-1} kadar öteleme. İlgili homojen dönüşüm matrisi:

$$T(a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

İkinci adım : x_{i-1} çevresinde α_{i-1} kadar dönme. İlgili homojen dönüşüm matrisi:

$$T(\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_{i-1} & -\sin \alpha_{i-1} & 0 \\ 0 & \sin \alpha_{i-1} & \cos \alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Üçüncü adım: 2. Adımdan sonra z_i haline gelmiş olan z_{i-1} boyunca d_i kadar öteleme. İlgili homojen dönüşüm matrisi:

$$T(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Dördüncü adım: z_i çevresinde θ_i kadar dönme. İlgili homojen dönüşüm matrisi:

$$T(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Her bir Denavit - Hartenberg parametresine bir dönüşüm eşleştirerek elde edilen yukarıdaki ifadeler yardımıyla Σ_i eksen takımından Σ_{i-1} eksen takımına homojen dönüşüm matrisinin

$${}^{i-1}T_i = T(a_{i-1}) \cdot T(\alpha_{i-1}) \cdot T(d_i) \cdot T(\theta_i) \quad (2.6)$$

şeklinde hesaplanabileceği açıktır. (2.2, 2.3, 2.4 ve 2.5) ifadeleri (2.6) da yerine konursa:

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\alpha_{i-1} \cdot \sin\theta_i & \cos\alpha_{i-1} \cdot \cos\theta_i & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} \cdot d_i \\ \sin\alpha_{i-1} \cdot \sin\theta_i & \sin\alpha_{i-1} \cdot \cos\theta_i & \cos\alpha_{i-1} & \cos\alpha_{i-1} \cdot d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

elde edilir. ${}^{i-1}T_i$ matrisine *uzuv dönüşüm matrisi* denir.

Robotun $i-1$. ve i . uzuvları arasındaki dönüşümleri temsil eden uzuv dönüşüm matrisleri yardımıyla ardışık olmayan uzuvlar arasında da dönüşüm matrisleri yazılabilir. Yazılabilen bu dönüşüm matrislerinin en önemlisi, hiç kuşkusuz, n uzuvlu robotun n . uzvuna bağlı Σ_n eksen takımıyla sabit uzva bağlı Σ_0 eksen takımı arasındaki dönüşümlere ilişkin olanıdır. Bu dönüşüm matrisi uzuv dönüşüm matrisleri yardımıyla

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot \dots \cdot {}^{n-1}T_n = \prod_{i=1}^n {}^{i-1}T_i \quad (2.8)$$

şeklinde hesaplanır.

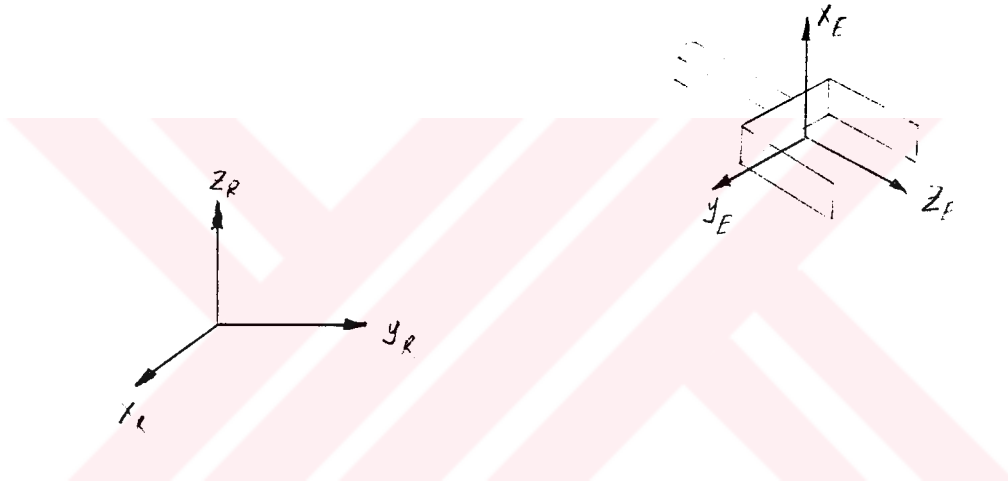
Çoğu zaman referans eksen takımı olarak, Σ_0 dan farklı bir Σ_R eksen takımının seçilmesi; n . uzva da Σ_n den farklı ve elin tutucu kısmına bağlanmış bir Σ_E eksen takımının yerleştirilmesi gerekli olur. Bu durumda Σ_E eksen takımından Σ_R eksen takımına dönüşümleri temsil eden ve (2.8) yardımıyla

$${}^R T_E = {}^R T_0 \cdot {}^0 T_n \cdot {}^n T_E \quad (2.9)$$

şeklinde hesaplanan dönüşüm matrisi önem kazanır. ${}^R T_E$ matrisine *robot dönüşüm matrisi* adı verilir. (Turhan, 1994).

2.5 Robotlarda Konum Analizi

Robotlarda konum analizi problemi, robotun her bir uzvunun yer ve duruşuyla ve bu uzuvlara ait noktaların konumlarıyla ilgili incelemeleri kapsamaktadır. Ama bir robotun konum analizi denildiğinde genellikle tutcuya bağlı Σ_E eksen takımının Σ_R referans eksen takımına göre yer ve duruşunun incelenmesi anlaşılır. (Şekil 2.4)



Şekil 2.4 Bir Robotun Referans ve Tutucu Eksen Takımları

Σ_E eksen takımının Σ_R eksen takımına göre yer ve duruşu ${}^R T_E$ homojen dönüşüm matrisi ile verilmektedir. Ancak bu matris iki şekilde ifade edilebilir. Bunlardan birincisinde ${}^R T_E$, (2.9) bağıntısıyla verilen robot dönüşüm matrisi olarak düşünülür. Bu durumda ${}^R T_E$, roboto ait Denavit - Hartenberg parametrelerinin fonksiyonu olacaktır. Fakat belirli bir robot durumunda ise Denavit - Hartenberg parametreleri olan $a_i, \alpha_i, d_i, \theta_i$ den, a_i, α_i, d_i belli olacağından ${}^R T_E$ matrisi bu durumda θ_i yani mafsallık koordinatlarının bir fonksiyonu olacaktır. Bu durumda n serbestlik dereceli bir robotun mafsallık koordinatları q_1, q_2, \dots, q_n şeklinde gösterilir ve bunlar R^n uzayında tanımlı bir

$$\bar{q} = \{q_1, q_2, q_3, \dots, q_n\}^T \quad (2.10)$$

mafsal koordinatları vektörünün yapısında bir araya getirilirse

$${}^R T_E = {}^R T_E(\bar{q}) = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

yazılabilir.

İkinci bir yol olarak da ${}^R T_E$, (2.1) ifadesindeki gibi düşünülebilir. Buradaki $R_{3 \times 3}$ dönme matrisini, rijid cisimler kinematığından bildiğimiz ve Euler açılarının (ϕ, θ, ψ) fonksiyonu olan dönme matrisi olarak düşünürsek : (Turhan, 1994). $\sin = s$ ve $\cos = c$ kısaltmasıyla:

$$R_{3 \times 3} = \begin{bmatrix} c\phi \cdot c\psi - s\phi \cdot c\theta \cdot s\psi & -c\phi \cdot s\psi - s\phi \cdot c\theta \cdot c\psi & s\phi \cdot s\theta \\ s\phi \cdot c\psi + c\phi \cdot c\theta \cdot s\psi & -s\phi \cdot s\psi + c\phi \cdot c\theta \cdot c\psi & -c\phi \cdot s\theta \\ s\theta \cdot s\psi & s\theta \cdot c\psi & c\theta \end{bmatrix} \quad (2.12)$$

dir.

Ötemele vektörü olan $P_{3 \times 1}$;

$$P_{3 \times 1} = ({}^R x_p, {}^R y_p, {}^R z_p) \quad (2.13)$$

olarak düşünülürse rijid cisimler kinematığının altı genelleştirilmiş koordinatı olan \vec{W} kartezyen koordinatlar vektörü yazılabilir:

$$\vec{W} = ({}^R x_p, {}^R y_p, {}^R z_p, \phi, \theta, \psi) \quad (2.14)$$

(2.12 ve 2.13), (2.1) de yerine konursa ;

$${}^R T_E = {}^R T_E(\vec{W}) = \begin{bmatrix} c\phi \cdot c\psi - s\phi \cdot c\theta \cdot s\psi & -c\phi \cdot s\psi - s\phi \cdot c\theta \cdot c\psi & s\phi \cdot s\theta & {}^R x_p \\ s\phi \cdot c\psi + c\phi \cdot c\theta \cdot s\psi & -s\phi \cdot s\psi + c\phi \cdot c\theta \cdot c\psi & -c\phi \cdot s\theta & {}^R y_p \\ s\theta \cdot s\psi & s\theta \cdot c\psi & c\theta & {}^R z_p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

elde edilir. (2.11) ve (2.15) birbirine eşitlenirse;

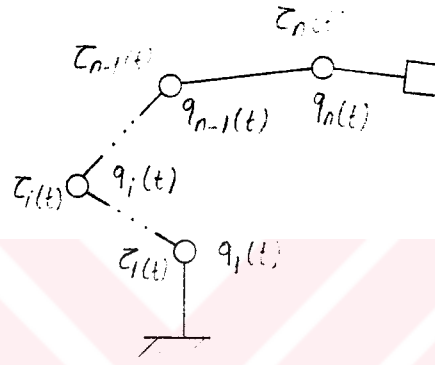
$${}^R T_E(\vec{q}) = {}^R T_E(\vec{W}) \quad (2.16)$$

şeklinde mafsalsal koordinatları vektörü \vec{q} ile kartezyen koordinatlar vektörü \vec{W} arasında bağıntılar elde edilir. Bu bağıntılar yardımıyla verilmiş \vec{q} ya karşılık gelen \vec{W} hesaplanabilir ve buna robotun *düz konum analizi* denir. Ya da verilmiş \vec{W} ye karşılık gelen \vec{q} hesaplanabilir, buna da robotun *ters konum analizi* denir. (Turhan, 1994).

BÖLÜM 3

3.1 ROBOT DİNAMIĞI

Dinamik, kuvvetler ile neden oldukları hareketler arasındaki ilişkileri incelemektedir. Robot dinamiğinde ise, mafsalsal motorlarının oluşturduğu mafsalsal kuvvetleri ile robot hareketi arasındaki ilişkiler incelenmektedir.



Şekil 3.1 Mafsalsal Kuvvet ve Koordinatları

Mafsalsal kuvvetlerini $\vec{\tau}(t) = \{\tau_1(t), \tau_2(t), \dots, \tau_n(t)\}^T$ vektörü,

robot hareketini de $\vec{q}(t) = \{q_1(t), q_2(t), \dots, q_n(t)\}^T$ mafsalsal koordinatları vektörü ve onun zamana göre türevleri olan

$\dot{\vec{q}}(t) = \{\dot{q}_1(t), \dot{q}_2(t), \dots, \dot{q}_n(t)\}^T$ mafsalsal hızları

vektörü ve $\ddot{\vec{q}}(t) = \{\ddot{q}_1(t), \ddot{q}_2(t), \dots, \ddot{q}_n(t)\}^T$ mafsalsal

ivmeleri vektörü temsil edecektir. Bu çerçevede robot dinamiği problemi

$\vec{\tau}(t)$ ile $\vec{q}(t)$ arasında

$$\vec{\tau}_{(t)} = \vec{\tau} \left(\vec{q}_{(t)}, \dot{\vec{q}}_{(t)}, \ddot{\vec{q}}_{(t)} \right) \quad (3.1)$$

şeklindeki fonksiyonel ilişkinin belirlenmesi probleminden ibarettir.

Bu ifade; n adet mafsallı bir robot için n adet (genelde bağılı ve non linner) ikinci mertebeden diferansiyel denklem verir ve bu diferansiyel denklemler iki farklı amaç için kullanılabilir :

1- Verilmiş $\vec{\tau}_{(t)}$ için $\vec{q}_{(t)}$ nin belirlenmesi , yani mafsallara belli kuvvetler uygulandığında buna karşılık robotun yapacağı hareketin bulunması istenebilir . *Robotun düz dinamik problemi* adı verilen bu problemin çözülmesi demek (3.1) diferansiyel denklem takımının çözülmesi demektir ki analitik olarak çözümü genelde imkansızdır . Bu nedenle bilgisayarda iteratif yolla sayısal çözüm elde edilmeye çalışılır . Düz dinamik problem , robot hareketinin Bilgisayar Simülasyonunda uygulama alanı bulur .

2- Verilmiş $\vec{q}_{(t)}$ için $\vec{\tau}_{(t)}$ nin belirlenmesi , yani robota arzu edilen bir hareketin yaptırılabilmesi için uygulanması gereken mafsallı kuvvetlerinin bulunması istenebilir . Bu probleme de *robotun ters dinamik problemi* adı verilir. Ters dinamik problem Robot Kontrolunda uygulama alanı bulur .

İncelenen robot için hangi problem çözülmek istenirse istensin sonuçta denklemlerin elde edilmesi gerekmektedir . Bu denklemlere Robotun Hareket Denklemi adı verilir.

Dinamiğin farklı prensipleri esas alınarak robotun hareket denklemleri farklı şekillerde elde edilebilir . Denklemler hangi yoldan elde edilirse edilsin , robotun hareketini temsil açısından aralarında hiçbir fark olmayacaktır . Fakat farklı yollardan elde edilen denklemlerin yapıları farklı olacaktır . Bu yapıların

birbirinden farklı olan temel özellikleri , kullanım amacına göre , aralarında bir tercih yapılmasını gerektirir .

Robot pratiğinde hareket denklemlerinin sahip olması istenen iki önemli özellik vardır . Bunlardan birincisi denklemlerin kapalı formda olması , yani $\vec{\tau}_{(t)}$ ve $\vec{q}_{(t)}$ arasındaki ilişkiyi açık ifadeler şeklinde ortaya koyması ; ikincisi ise denklemlerin sratli deęerlendirilebilir yapıda olmasıdır . Bu özelliklerden ilki kontrol ve tasarım mhendislerinin analiz yapmalarını kolaylařtırırken; ikincisi ise robotun gerek zamanda kontrol edilebilmesini kolaylařtırmaktadır .

Robot hareket denklemleri formlasyonunun bilinen  ana yolu ve bu yollardan elde edilen denklemlerin özellikleri kısaca řoyledir :

- 1- Lagrange - Euler Formlasyonu : Lagrange dinamięini esas alır ve kapalı formda fakat hızlı deęerlendirilemeyen denklemler verir .
- 2- Newton - Euler Formlasyonu : Newton - Euler dinamięini esas alır . Kapalı formda olmayan rekrsif yapıda fakat hızlı deęerlendirilebilen denklemler verir
- 3- Genelleřtirilmiř d' Alembert Formlasyonu : d' Alembert dinamięini esas alır. Hem kapalı formda hem de N-E formlasyonundaki kadar olmamakla birlikte hızlı deęerlendirilebilen denklemler verir .(Turhan, 1994).

3.2 Lagrange - Euler Formülasyonu

Robot hareket denklemlerinin elde edilmesinde ilk uygulanan yaklaşım L - E formülasyonu olmuştur . Bu yaklaşımın robot dinamiğinin yapısına nüfuz edilmesini kolaylaştıran çözümlmeye elverişli denklemlere götürme özelliği ve diğer bütün yaklaşımlardan daha sistematik bir yapısı vardır

Kısaca Lagrange Hareket Denklemini hatırlanırsa :

T : Kinetik enerji

V : Potansiyel enerji

q : Genelleştirilmiş koordinat

τ : Mafsal kuvveti

L = T - V şeklinde Lagrange Fonksiyonu olmak üzere

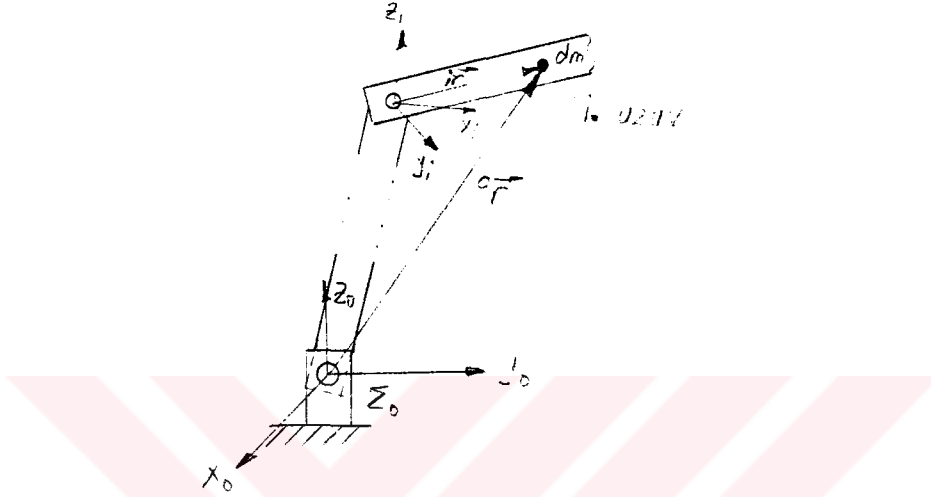
$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1.2.3.....n \quad (3.2)$$

Bu durumda (3.2) denkleminin değerlendirilebilmesi için robotun kinetik ve potansiyel enerjilerinin ifade edilmesi gerekmektedir.

3.2.1 Robotun Kinetik Enerjisi

Herhangi bir robotun kinetik enerjisini ifade etmekte robot uzuvlarını oluşturan rijid cisimlerin kinetik enerjilerini, rijid cisimler dinamiğinin bilinen sonuçlarına dayanarak ifade etmek mümkündür. Fakat robot sistemlerine uygun sistematik bir formülasyona ulaşmak da mümkündür. (Turhan, 1994).

Bu amaçla rijid robot uzuvlarından her birin dm maddesel noktalarından oluşmuş bir bütün olarak düşünülebilir. (Şekil 3.1). İncelemeye bu dm noktalarının kinetik enerjilerini hesaplamakla başlanabilir. (Turhan, 1994).



Şekil 3.1 Kinetik Enerji Hesabında Robot Modeli

i. robot uzvunun dm elemanının kinetik enerjisi Şekil 3.1 yardımıyla

$$dT_i = \frac{1}{2} \cdot {}^0\dot{\vec{r}}^T \cdot {}^0\dot{\vec{r}} \cdot dm = \frac{1}{2} \cdot iz \left({}^0\dot{\vec{r}} \cdot {}^0\dot{\vec{r}}^T \right) \cdot dm \quad (3.3)$$

şeklinde yazılabilir. Burada ikinci ifadedeki " iz " , bir kare matrisin ana köşegeni üzerindeki elemanlarının toplamıdır . Yani :

$$iz A = \sum_{i=1}^n a_{ii} \quad (3.4)$$

şeklinde ifade edilir. Şimdi (3.3) bağıntısındaki ${}^0\dot{\vec{r}}$ hesaplınsın.

$${}^0\dot{\vec{r}} = {}^0T_i \cdot \dot{\vec{r}} = {}^0T_1(q_1) \cdot {}^1T_2(q_2) \cdot \dots \cdot {}^{i-1}T_i(q_i) \cdot \dot{\vec{r}} \quad (3.5)$$

olduğu ve ${}^i\vec{r}$ - rijid cismin yer vektörü - rijid cismin tanımı gereği sabit olduğua göre:

$${}^0\dot{\vec{r}} = {}^0\dot{T}_i(q_1, q_2, \dots, q_i) \cdot {}^i\vec{r} = \left[\sum_{j=1}^i \frac{\partial {}^0T_i}{\partial q_j} \cdot \dot{q}_j \right] \cdot {}^i\vec{r} \quad (3.6)$$

yazılabilir. (3.6) ifadesi (3.3) de yerine konursa :

$$d T_i = \frac{1}{2} \cdot iz \left[\sum_{j=1}^i \frac{\partial {}^0T_i}{\partial q_j} \cdot \dot{q}_j \cdot {}^i\vec{r} \cdot \left(\sum_{k=1}^i \frac{\partial {}^0T_i}{\partial q_k} \cdot \dot{q}_k \cdot {}^i\vec{r} \right)^T \right] \cdot dm \quad (3.7)$$

elde edilir. Burada toplam alma ve iz almanın sıraları değiştirilip \dot{q}_j ve \dot{q}_k skalerleri matrisin dışına alınarak ve $(A.B)^T = B^T.A^T$ özelliği dikkate alınırsa

$$d T_i = \frac{1}{2} \cdot \sum_{j=1}^i \sum_{k=1}^i iz \left(\frac{\partial {}^0T_i}{\partial q_j} \cdot {}^i\vec{r} \cdot {}^i\vec{r}^T \cdot \frac{\partial {}^0T_i}{\partial q_k} \right) \cdot \dot{q}_j \cdot \dot{q}_k \cdot dm \quad (3.8)$$

bulunur ve buradan i. uzvun kinetik enerjisine geçilebilir. i. uzvun kinetik enerjisini veren ifadeyi elde edebilmek için (3.8) ifadesinden i. robot uzvu boyunca integral alınarak ve uzvun içindeki konumdan bağımsız olan unsurlar integralin dışında bırakılarak

$$T_i = \frac{1}{2} \cdot \sum_{j=1}^i \sum_{k=1}^i iz \left[\frac{\partial {}^0T_i}{\partial q_j} \cdot \left(\int_i {}^i\vec{r} \cdot {}^i\vec{r}^T \cdot dm \right) \cdot \frac{\partial {}^0T_i}{\partial q_k} \right] \cdot \dot{q}_j \cdot \dot{q}_k \quad (3.9)$$

elde edilir. Burada integral altındaki kısım ayrılır ve

${}^i\vec{r} = \{x_i, y_i, z_i, 1\}^T$ olduğunu hatırlanırsa :

$$H_i = \int_i \vec{r}^i \cdot \vec{r}^i \cdot \vec{r}^i \cdot dm = \begin{bmatrix} \int x_i^2 \cdot dm & \int x_i \cdot y_i \cdot dm & \int x_i \cdot z_i \cdot dm & \int x_i \cdot dm \\ \int x_i \cdot y_i \cdot dm & \int y_i^2 \cdot dm & \int y_i \cdot z_i \cdot dm & \int y_i \cdot dm \\ \int x_i \cdot z_i \cdot dm & \int y_i \cdot z_i \cdot dm & \int z_i^2 \cdot dm & \int z_i \cdot dm \\ \int x_i \cdot dm & \int y_i \cdot dm & \int z_i \cdot dm & \int dm \end{bmatrix} \quad (3.10)$$

elde edilir. m_i i. uzvun kütlesi; x_i, y_i, z_i ise i.uzvun kütle merkezinin Σ_i deki koordinatları olmak üzere

$$\int_i dm = m_i ; \int_i x_i \cdot dm_i = m_i \cdot x_i ; \int_i y_i \cdot dm_i = m_i \cdot y_i ; \int_i z_i \cdot dm_i = m_i \cdot z_i \quad (3.11)$$

eşitlikleri dikkate alınır ve rijid cisimler dinamiğinden rijid cismin atalet tansörü hatırlanırsa (3.10) ifadesi şu forma girer :

$$H_i = \begin{bmatrix} \frac{-J_{xx} + J_{yy} + J_{zz}}{2} & J_{xy} & J_{xz} & m_i \cdot x_i \\ J_{xy} & \frac{J_{xx} - J_{yy} + J_{zz}}{2} & J_{yz} & m_i \cdot y_i \\ J_{xz} & J_{yz} & \frac{J_{xx} + J_{yy} - J_{zz}}{2} & m_i \cdot z_i \\ m_i \cdot x_i & m_i \cdot y_i & m_i \cdot z_i & m_i \end{bmatrix} \quad (3.12)$$

Burada $J_{\alpha\alpha}$ lar i. uzvun Σ_i eksen takımında ifade edilmiş atalet momentleri, $J_{\alpha\beta}$ lar ise yine aynı eksen takımındaki atalet çarpımlarıdır . (3.12) deki H_i matrisi i. uzvun kütle özelliklerini göstermektedir . Atalet tansörüyle benzerliklerine dayanarak bu matrise psödo atalet tansörü adı verilir .

Buradan (3.12) ifadesi, (3.9) da yerine konursa i. uzvun kinetik enerjisini veren ifade şu şekli alır :

$$T_i = \frac{1}{2} \cdot \sum_{j=1}^i \sum_{k=1}^i \text{iz} \left(\frac{\partial {}^0 T_i}{\partial q_j} \cdot H_i \cdot \frac{\partial {}^0 T_i^T}{\partial q_k} \right) \cdot \dot{q}_j \cdot \dot{q}_k \quad (3.13)$$

Şimdi buradaki $\frac{\partial {}^0 T_i}{\partial q_j}$ ifadesi incelenirse ve yazım kolaylığı açısından

$$U_{ij} = \frac{\partial {}^0 T_i}{\partial q_j} \quad (3.14)$$

denirse :

$$U_{ij} = {}^0 T_1^{-1} T_2 \cdot \dots \cdot {}^{j-2} T_{j-1} \cdot \frac{\partial {}^{j-1} T_j}{\partial q_j} \cdot {}^j T_{j+1} \cdot \dots \cdot {}^{i-1} T_i = {}^0 T_{j-1} \cdot \frac{\partial {}^{j-1} T_j}{\partial q_j} \cdot {}^j T_i \quad (3.15)$$

yazılabilir. Buna göre U_{ij} ler, robot kinematığından bilinen uzuv dönüşüm matrisleriyle bunların $\frac{\partial {}^{r-1} T_r}{\partial q_r}$ kısmi türev matrisleri cinsinden belirlidir. Şimdi bu kısmi türev matrislerinin ifadesini elde etmek için (2.7) ifadesi hatırlanırsa

$${}^{r-1} T_r = \begin{bmatrix} \cos \theta_r & -\sin \theta_r & 0 & a_{r-1} \\ \cos \alpha_{r-1} \cdot \sin \theta_r & \cos \alpha_{r-1} \cdot \cos \theta_r & -\sin \alpha_{r-1} & -\sin \alpha_{r-1} \cdot d_r \\ \sin \alpha_{r-1} \cdot \sin \theta_r & \sin \alpha_{r-1} \cdot \cos \theta_r & \cos \alpha_{r-1} & \cos \alpha_{r-1} \cdot d_r \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

Buradan r. mafsalın döner mafsal olması halinde $q_r = \theta_r$ ile

$$\frac{\partial {}^{r-1}T_r}{\partial q_r} = \begin{bmatrix} -\sin\theta_r & -\cos\theta_r & 0 & 0 \\ \cos\alpha_{r-1} \cdot \cos\theta_r & -\cos\alpha_{r-1} \cdot \sin\theta_r & 0 & 0 \\ \sin\alpha_{r-1} \cdot \cos\theta_r & -\sin\alpha_{r-1} \cdot \sin\theta_r & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.17)$$

r. mafsalin kayar mafsali olmasi halinde $q_r = d_r$ ile

$$\frac{\partial {}^{r-1}T_r}{\partial q_r} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sin\alpha_{r-1} \\ 0 & 0 & 0 & -\cos\alpha_{r-1} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.18)$$

Buradan D_r ; r .mafsal döner mafsali (R) iken

$$D_r = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.19)$$

r .mafsal kayar mafsali (P) iken

$$D_r = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.20)$$

olacak Őekilde tanimlanan bir matris olmak üzere :

$$\frac{\partial {}^{r-1}T_r}{\partial q_r} = {}^{r-1}T_r \cdot D_r \quad (3.21)$$

şeklinde hesaplanır .Bu (3.21) ifadesini (3.15) de yerine koyarsak :

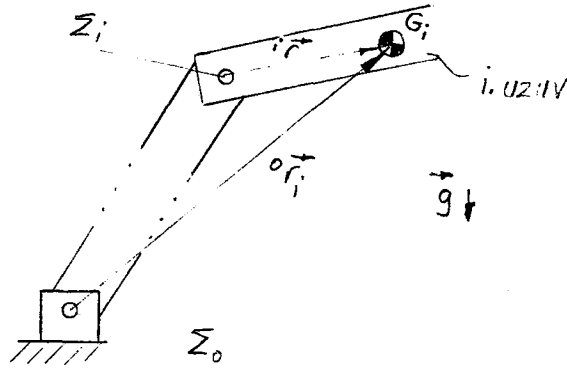
$$\begin{aligned} U_{ij} &= {}^0T_j \cdot D_j \cdot {}^jT_i \quad i \leq j \quad \text{koşuluyla} \\ U_{ij} &= 0 \quad i > j \quad \text{koşuluyla} \end{aligned} \quad (3.22)$$

elde edilir . Elde edilen bu (3.22) ve (3.14) ifadeşeri (3.13) de yerine konursa robotun toplam kinetik enerjisini veren şu ifade elde edilir .

$$T = \frac{1}{2} \cdot \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{iz} \left(U_{ij} \cdot H_i \cdot U_{ik}^T \right) \cdot \dot{q}_j \cdot \dot{q}_k \quad (3.23)$$

3.2.2 Robotun Potansiyel Enerjisi

Robotların büyük çoğunluğunda potansiyel enerji yalnızca yer çekimi alanıyla ilgilidir. Bu formülasyonda da robotun potansiyel enerjisi sadece yer çekimi potansiyeli açısından incelenmektedir. (Turhan, 1994).



Şekil 3.2 i. uzuv ve Kütle Merkezi

İlkin robotun i. uzvunun potansiyel enerjisi yazılmak istenirse Şekil 3.2 yardımıyla ve

$$\vec{r}_i = \{ x_i, y_i, z_i, 1 \}^T \quad (3.24)$$

j. uzvun kütle merkezi g_i nin yer vektörünü ,

$${}^0\vec{g} = \{ g_x, g_y, g_z, 0 \}^T \quad (3.25)$$

ise yer çekimi ivmesi vektörünün 4x1 lik bir matris şeklindeki ve Σ_0 referans eksen takımındaki ifadesini göstermek üzere :

$$V_i = -m_i \cdot {}^0\vec{g}^T \cdot {}^0\vec{r}_i = -m_i \cdot {}^0\vec{g}^T \cdot ({}^0T_{i,i} \vec{r}_i) \quad (3.26)$$

yazılabilir . Bu ifade , Σ_0 in orjininden geçen yatay düzlem, yer çekimi potansiyeli için referans düzeyi alınarak yazılmıştır. (3.26) ifadesinden robotun toplam potansiyel enerjisi hemen

$$V = \sum_{i=1}^n V_i = -\sum_{i=1}^n m_i \cdot {}^0\vec{g}^T \cdot ({}^0T_{i,i} \vec{r}_i) \quad (3.27)$$

şeklinde elde edilir.

3.3 Hareket Denklemleri

Robotun kinetik ve potansiyel enerjileri için elde edilen (3.23) ve (3.27) ifadeleri $L = T - V$ şeklindeki Lagrange Fonksiyonunda yerine konulursa

$$L = \frac{1}{2} \cdot \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{iz} \left(U_{ij} \cdot H_i \cdot U_{ik}^T \right) \cdot \dot{q}_j \cdot \dot{q}_k + \sum_{i=1}^n m_i \cdot {}^0\vec{g}^T \cdot \left({}^0T_i \cdot \vec{r}_i \right) \quad (3.28)$$

ifadesi elde edilir. Bu ifade ile (3.2) ye dönülürse gerekli hesap ve düzenlemelerden sonra

$$\tau_i = \sum_{k=i}^n \sum_{j=1}^k \text{iz} \left(U_{kj} \cdot H_j \cdot U_{ki}^T \right) \cdot \ddot{q}_j + \sum_{k=i}^n \sum_{j=1}^k \sum_{m=1}^k \text{iz} \left(U_{kjm} \cdot H_k \cdot U_{ki}^T \right) \cdot \dot{q}_j \cdot \dot{q}_m - \sum_{j=i}^n m_j \cdot {}^0\vec{g}^T \cdot U_{ji} \cdot \vec{r}_j \quad (3.29)$$

elde edilir. Bu denklemi biraz düzenlemek için

$$M_{ij} = \sum_{k=\max(i,j)}^n \text{iz} \left(U_{kj} \cdot H_j \cdot U_{ki}^T \right) \quad (3.30)$$

$$h_{ijm} = \sum_{k=\max(i,j,m)}^n \text{iz} \left(U_{kjm} \cdot H_k \cdot U_{ki}^T \right) \quad (3.31)$$

$$c_i = \sum_{j=i}^n -m_j \cdot {}^0\vec{g}^T \cdot U_{ji} \cdot \vec{r}_j \quad (3.32)$$

tanımları yardımıyla

$$\tau_i = \sum_{j=1}^n M_{ij} \cdot \ddot{q}_j + \sum_{j=1}^n \sum_{m=1}^n h_{ijm} \cdot \dot{q}_j \cdot \dot{q}_m + c_i \quad (3.33)$$

de yazılabilir. Yine bu ifade matris formunda gösterilirse:

$$\vec{\tau} = M(\vec{q}) \cdot \ddot{\vec{q}} + \vec{h}(\vec{q}, \dot{\vec{q}}) + \vec{c}(\vec{q}) \quad (3.34)$$

ifadesi de kullanılabilir. Burada $\vec{\tau}$, \vec{q} , $\dot{\vec{q}}$, $\ddot{\vec{q}}$, sırasıyla mafsal kuvvetleri, mafsal koordinatları, mafsal hızları ve mafsal ivmeleri vektörüdür. Diğer terimler ise :

$M(\vec{q})$: Elemanları (3.30) da tanımlı , $n \times n$ boyutlu simetrik bir matris olup robotun atalet matrisi adını alır.

$\vec{h}(\vec{q}, \dot{\vec{q}}) = \{ h_1, h_2, \dots, h_n \}^T$ vektörü Coriolis ve merkezkaç kuvvetler vektörüdür . Elemanları (3.31) yardımıyla

$$h_i = \sum_{j=1}^n \sum_{m=1}^n h_{ijm} \cdot \dot{q}_j \cdot \dot{q}_m \quad (3.35)$$

şeklinde tanımlanır ve $j = m$ halinde merkez kaç kuvvetleri $j \neq m$ halinde Coriolis kuvvetlerini gösterir .

$\vec{c}(\vec{q}) = \{ c_1, c_2, \dots, c_n \}^T$ vektörü ise elemanları (3.32) de tanımlı ağırlık kuvvetleri vektörüdür .

Robot hareket denklemlerinin (3.29) , (3.33) ve (3.34) ifadeleri kapalı formda ifadelerdir. Bu özellikleriyle robot üzerine etkiyen kuvvetlerin kökenlerine göre

ayırddilmesine imkan verdikleri gibi her kuvvet tipi üzerinde robot parametrelerinin etkilerinin çözümlenmesine de elverişli yapıdadırlar . (Turhan,1994).



BÖLÜM 4

4.1 BİLGİSAYAR PROGRAMI

Birinci bölümde de değiniildiği gibi bir robotu daha etkin ve verimli kılabilmek için her açıdan çok iyi analiz edilmesi gerekmektedir. Bunu sağlamanın en kolay yoluda robotun matematiksel modelini elde etmektir. Robotun matematiksel modelini elde etmek ise birçok karışık matematiksel işlemin yapılmasını gerektirmektedir. Bu işlemleri çok kısa sürede doğru olarak yapabilmek için ise bilgisayarlardan yararlanmak bir zorunluluk olmaktadır.

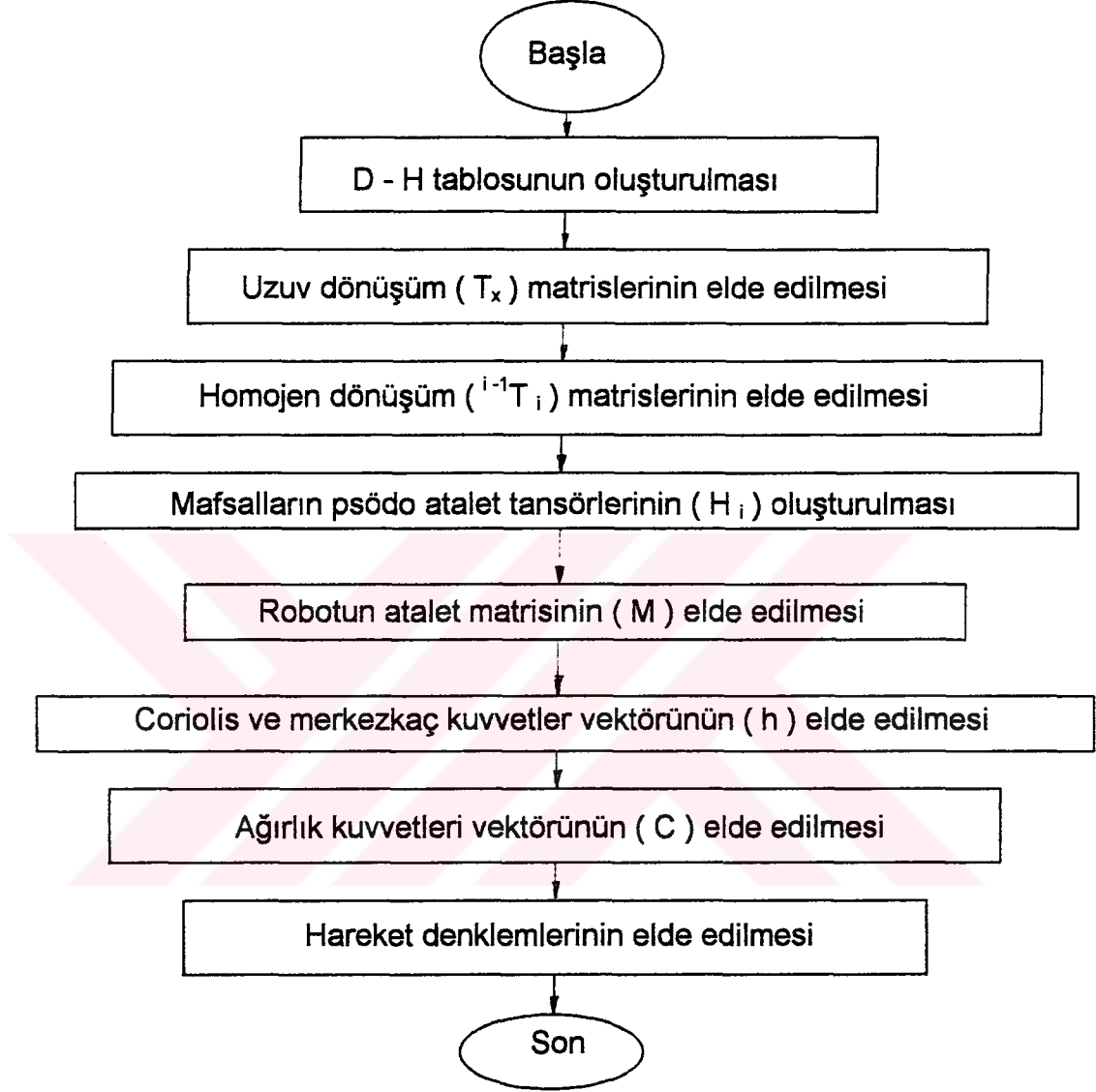
Ayrıca tasarlanan robotu imal etmeden önce sanal ortamda gerçekleştirip yapacağı hareketleri simüle etmek de birçok faydalar sağlayacaktır. Bunu yapmak için de elde edilen matematiksel modelinin kullanılması gerekmektedir. Bütün bunları yapmak için ise bilgisayar programlarına ihtiyaç duyulmaktadır. Bu nedenle bir çok araştırmacı bu konu üzerine eğilmiş ve farklı programlar geliştirmiştir.

Bu tezde her hangi bir robotun matematiksel modelini sembolik formda verebilen bir program ve robotun yapacağı hareketi simüle edebilen ayrı bir program geliştirilmeye çalışıldı.

4.2 Matematiksel Model Programı

Daha önce de değinildiği gibi robotun matematiksel modelinin elde edilmesinde sadece Lagrange - Euler Formulasyonu esas alınırken uzuvlara eksen takımı yerleştirilmesinde de Denavit - Hartenberg Prensipleri esas alınmaktadır. Bu tezde incelenen hareket denklemlerinin temelini oluşturan D - H Tablosu tamamen sembolik karakterlerden oluşmaktadır. Denklemleri kısa

formda gösterebilmek için trigonometrik ifadeler olan \sin ve \cos , s ve c olarak kısaltılmaktadır. Programın akış şeması şöyledir :



Şekil 4.1 Matematiksel Model Programı Akış Şeması

4.2.1 Alt Programlar

Her hangi bir robotun matematiksel modelini sembolik formda elde edebilmek için geliştirilmiş olan program bir çok alt programlardan (fonksiyon) meydana gelmektedir. Bu programların tamamı C++ 3.0 da yazılmışlardır. Bu alt programlar ve özellikleri kısaca şöyle açıklanabilir :

DHTABLO Hareket denklemlerinin elde edilmesinin temelini oluşturan D - H Tablosunun elemanlarının bilgisayara girilmesini ve hafızada tutularak diğer alt programlar tarafından kullanılmasını sağlar. Maksimum yedi mafsal için D - H parametresi girilebilmesine ve yine her bir parametre en fazla on karakter olmasına müsaade eder. Kinematik analiz esnasında da belirtildiği gibi hesapların kısa sürede ve hareket denklemlerinin kısa formlarda olabilmesi için D - H parametrelerinin mümkün olduğu kadar az karakterle hatta sıfır olarak ifade edilmesi gerekmektedir .Ayrıca sorulan mafsal tipine Revölüt mafsallar için r , Prizmatik Mafsallar için p değerlerinin girilmesi gerekmektedir .

TXMAT Txmat() fonksiyonu daha önce girilmiş olan D - H parametrelerini uzuv dönüşüm matrislerinde yerine koymaktadır. Kinematik analizde de açıklandığı gibi uzuv dönüşüm matrisleri; her mafsal için belirlenmiş olan dört adet D - H Parametresinin her birine karşılık gelen dönme veya öteleme hareketlerini bir önceki eksen takımına indirgeyen matrislerdir. 4x4 boyutundadırlar ve her bir mafsal dört adet uzuv dönüşüm matrisine sahiptir. Bunlar $T(a)$; $T(\alpha)$; $T(d)$; $T(\theta)$ matrisleridir. Txmat() fonksiyonu tarafından hesaplanan sembolik matrisler sonraki kullanımlar için hafızada saklanmaktadır.

HTMMAT Txmat() fonksiyonu tarafından hesaplanan $T(a)$; $T(\alpha)$; $T(d)$; $T(\theta)$ uzuv dönüşüm matrislerini sırayla çarparak ilgili

mafsalın homojen dönüşüm matrisini hesaplar. Homojen dönüşüm matrisi hem dönme hem de öteleme hareketlerini ifade ederek Σ_i eksen takımını Σ_{i-1} eksen takımına indirgeyen matristir. Matris çarpımı komutatif olmadığı ($A \cdot B \neq B \cdot A$) için matris çarpım sırası oldukça önemlidir ve programlama esnasında bu özellik göz önünde tutulmuştur.

H_i Mafsalların psödo atalet tansörlerinin bilgisayar ortamına girilmesini sağlar. H_i matrisi i . uzvun kütle özelliklerini tasvir etmektedir. Atalet tansörüyle benzerliklerinden ötürü bu isim verilir. Yine bu matrisin elemanları oluşturulurken mümkün olduğu kadar kabuller yapılarak elemanlar sıfırlanmalı veya küçültülmelidir. Aksi takdirde hareket denklemleri çok uzun olmaktadır ve bu da hafıza sorunları çıkarmaktadır. Önemli bir özellik olarak da mafsalların kütle değerleri mutlak şekilde m_1 , m_2 şeklinde olmalıdır. Aksi takdirde C ağırlık kuvvetleri vektörünün hesaplanmasında kullanılan ve program tarafından otomatik olarak verilen kütle değerleriyle uyuma sağlanamaz.

M_i Robotun atalet matrisini hesaplar. Robotun dinamik analizini inzelenirken hareket denklemlerinin genel formunu (3.34) daki ifadesiyle $\vec{\tau} = M(\vec{q}) \cdot \ddot{\vec{q}} + \vec{h}(\vec{q}, \dot{\vec{q}}) + \vec{c}(\vec{q})$ şeklinde elde edilmişti. Buradaki M matrisi robotun atalet matrisini vermektedir ve n mafsallı bir robot için $n \times n$ boyutunda simetrik bir matristir. Yani $M_{ij} = M_{ji}$ dir. İvme değerleri ise program tarafından verilmektedir. 1. Mafsalın ivmesi A_1 , 2. Mafsalın ivmesi A_2 gibi.

HCOR_i Robotun Coriolis ve merkezkaç kuvvetler vektörünün elemanlarını hesaplar. Yine Robotun dinamik analizinden hareket denklemlerinin genel formunu hatırlanırsa $\vec{\tau} = M(\vec{q}) \cdot \ddot{\vec{q}} + \vec{h}(\vec{q}, \dot{\vec{q}}) + \vec{c}(\vec{q})$ idi. Buradaki h vektörü robotun coriolis ve

merkezkaç kuvvetler vektörüdür ve n mafsallı bir robot için n boyutlu bir vektördür . Vektörün her bir elemanı (3.35) ifadesiyle $h_i = \sum_{j=1}^n \sum_{m=1}^n h_{ijm} \cdot \dot{q}_j \cdot \dot{q}_m$ şeklindedir .Program sadece $h_{ijm} \cdot \dot{q}_j \cdot \dot{q}_m$ elemanlarını hesaplıyamaktadır ve her mafsalin hızı V1 , V2 gibi program tarafından verilemektedir .Mafsal koordinatı olan q ise D - H parametrelerinden gelmektedir .

CKOORD Robotun ağırlık kuvveti vektörünü hesaplar. Yine robotun dinamik analizinden hareket denklemlerinin genel formu hatırlanırsa $\vec{\tau} = M(\vec{q}) \cdot \ddot{\vec{q}} + \vec{h}(\vec{q}, \dot{\vec{q}}) + \vec{c}(\vec{q})$ idi. Buradaki C vektörü robotun ağırlık kuvvetleri vektörüdür ve n mafsallı bir robot için n boyutlu bir vektördür. Mafsal koordinatı olan ise D - H parametrelerinden gelmektedir. Vektör elemanlarında kullanılan mafsal kütle değerleri m1 , m2 şeklinde program tarafından verilmektedir .

MAT Robot hareket denklemlerinin Lagrange - Euler Formulasyonu ile elde edilişi büyük oranda 4x4 boyutlu matrislerin çarpımına dayanmaktadır. Mat() fonksiyonu 4x4 matrislerin çarpımını sembolik olarak yapmaktadır.

UIJ Yine robotun matematiksel modeli elde edilirken bir çok

$$\text{yerde } U_{ij} = \frac{\partial^0 T_i}{\partial q_j} \text{ türevini almak gerekmektedir. Uij()}$$

fonksiyonu bu türevi sembolik olarak hesaplamaktadır.

UIJK Robotun matematiksel modelini elde ederken yine bir çok

$$\text{yerde } U_{ijk} = \frac{\partial U_{ij}}{\partial q_k} = \frac{\partial^2 T_i}{\partial q_j \partial q_k} \text{ türevini almak}$$

gerekmektedir. Uijk() fonksiyonu bu türevi sembolik olarak hesaplamaktadır.

MAIN

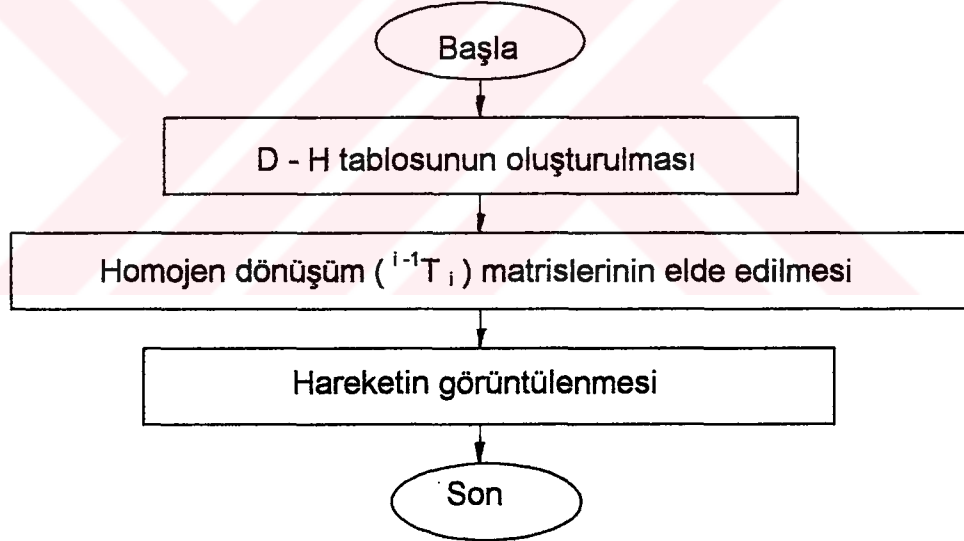
Robotun matematiksel modelini veren ana fonksiyondur.

Yukarıda açıklanan alt fonksiyonları kullanarak robotun hareket denklemlerini sembolik formda verir.

4.3 Simülasyon Programı

Bu tezde robot hareketinin simülasyonundan şu kastedilmektedir :

Robot uzuvlarına yerleştirilen eksen takımlarının konum ve duruşları, mafsal motorlarına verilen hareketlerle nasıl değişiyor ve tutucu en son hangi konumda kalıyor bunu görmek. Bunu yapabilmek için simülasyon programı tarafından tamamen nümerik değerler kullanılmaktadır ve uzuvlar ekranda, eksen takımlarını bir birine bağlayan çizgiler olarak ifade edilmektedir. Hareketin simüle edilmesi aşamasında şu işlem sırası takip edilmektedir:



Şekil 4.2 Simülasyonun Programı Akış Şeması

Matematiksel model programın da olduğu gibi robot hareketini simüle edebilmek için geliştirilmiş olan program da bir çok alt programlardan meydana gelmektedir. Bu programların da tamamı C++ 3.0 da yazılmışlardır. Bu alt programlar ve özellikleri kısaca şöyle açıklanabilir :

4.3.1 Alt Programlar

NDHTABLO Hareket denklemlerinin elde edilmesinde olduğu gibi robotun yapacağı hareketin görüntülenmesi için de yine D-H Tablosunun oluşturulması gerekmektedir. Fakat hareket denklemlerinin elde edilmesinden farklı olarak bu tablo tamamen nümerik değerlerden oluşmaktadır. Ndhtablo() fonksiyonu bu nümerik değerlerin bilgisayar ortamına girilmesini ve hafızada tutularak diğer alt programlar tarafından kullanılmasını sağlamaktadır. Görüntülenmek istenen hareketin ekranda görülebilmesi için verilecek uzuv boyutları oldukça önemlidir. Bu tezin gerçekleştirilmesi esnasında 640x480 çözünürlüklü ekran kullanılmış olup simülasyon programı bu değerler esas alınarak yapılmıştır. D - H tablosuna boyut olarak girilen değerler pixel olarak değerlendirilmektedir . Yani D - H tablosuna 100 br. olarak girilen bir boyut ekranda 100 pixel olarak görüntülenmektedir. Bunun yanında açılarının derece olarak girilmesi istenmektedir.

NHTMMAT Robotun nümerik homojen dönüşüm matrislerini hesaplayan fonksiyondur. Daha önce girilmiş olana D H Tablosundaki değerleri ilgili mafsalın homojen dönüşüm matrisinde yerine koyar. Robot uzuvlarının homojen dönüşüm matrislerini hesaplar.

GRAFİK Fonksiyon çalıştırılırken kullanıcıdan istenen robotun yapacağı harekete ait bilgileri ve daha önce girilmiş olan D - H Tablosuna göre hesaplanan nümerik homojen dönüşüm matrislerini kullanarak robotun yapacağı hareketi düzlemsel olarak görüntülüyene fonksiyondur. Robotun hareketini simüle edebilmek için herşeyden önce robotun yapacağı hareketin bilinmesi gerekmektedir. Bu nedenle robotun yapacağı hareket dört parametreden oluşan bir döngü ile kullanıcıdan istenmekte ve bu hareket programı HAR.DAT dosyasında sonraki kullanımlar için saklanmaktadır. Hareket programına geçmeden önce kullanıcıdan referans

eksen takımının konumu sorulmaktadır. Boyutlandırmada ekranın sol alt köşesi (0 , 0) koordinatlarında kabul edilirken sağ üst köşe (640 , 480) koordinatlarında olarak kabul edilmektedir. Hareket programı tarafından ilk olarak hareketi yapacak olan mafsalın numarasının girilmesi istenirken ikinci olarak yapılacak hareketin miktarının girilmesi istenmektedir. Daha önce de değinildiği gibi, eğer yapılacak hareket doğrusal ise girilen değer pixel olarak işlem görmektedir. Yani kayar bir mafsalın 175 br. hareket etmesi isteniyorsa bu hareket ekranda 175 pixel olarak görüntülenmektedir. Bu nedenle D - H tablosuna girilen doğrusal boyut değerlerin birimi ile yapılacak doğrusal hareket miktarının hızı bir birbine uygun olmalıdır. Yani hızın birimi boyut birimi cinsinden olmalıdır. Üçüncü parametre olarak kullanıcıdan mafsal hızı istenmektedir. Bu ise doğrusal hareketler için br/ms , açısal hareketler için deg/ms olması gerekmektedir. Dördüncü parametre ise ilgili mafsalın hareketsiz bekleme süresidir ki bunun birimi saniyedir ve buna göre bir değer girilmesi istenmektedir.

MAIN Robotun yapacağı hareketi simüle eden ana fonksiyondur. Yukarıda açıklanan alt fonksiyonları kullanarak robotun yapacağı hareketi simüle etmektedir.

BÖLÜM 5

5.1 SONUÇLAR

Önceki bölümlerde yapılan kinematik ve Lagrange - Euler Formülasyonunu esas alan dinamik incelemeler sonucu robotun matematiksel modelini veren formüller elde edildi. Daha sonra bu formüller kullanılarak robotun matematiksel modelini sembolik formda verebilen bir bilgisayar programı geliştirildi. Ayrıca robotun yapması planlanan hareketi de simüle edebilen bir program geliştirildi. Yapılan bu çalışmaların kısa bir değerlendirmesi aşağıda yapılmıştır.

5.2 Matematiksel Modelin Elde Edilmesi

Bu tezde sadece belirli tiplerdeki robotların değil; D - H Tablosu oluşturulan herhangi tipteki bir robotun matematiksel modelini elde edilmesi planlandı. Bu düşünceden dolayı bir takım kabullerin yapılması imkan dışı kaldı ve bunun sonucu olarak çalışma biraz genişlerken bazı konularda yüzeysel kaldı ve derinlemesine bir inceleme yapılamadı. Bu nedenle geliştirilen program yardımıyla elde edilen herhangi bir robota ait matematiksel modeli sembolik formda olmakta fakat yine de uzun bir hal almaktadır. Bu ise matematiksel modelin net bir şekilde görülmesini ve analizini biraz zor hale getirmektedir. Ayrıca elde edilen sembolik formdaki hareket denklemleri program tarafından nümerik hale dönüştürülememektedir. Bu da elde edilen denklemlerin simülasyon amaçlı olarak kullanılmasını engellemektedir. Bu sebepten dolayı simülasyon esnasında hareket denklemlerinin bir kısmının (homoje dönüşüm matrisleri) nümerik olarak tekrar hesaplanması zorunluluğu doğmaktadır.

5.3 Robot Hareketinin Simülasyonu

Robot hareketinin simülasyonu demek ; robotun yapacağı hareketin sanal ortamda olduğu gibi canlandırılması demektir. Bu ise oldukça gelişmiş programlar gerektirmektedir. Daha önce değinildiği gibi bu tezde belirli robot tipleriyle sınırlamaya gidilmedi. D - H Tablosu oluşturulan her hangi bir robotun hareket denklemleri nasıl elde edilebiliyorsa yapacağı hareketler de simüle edilebiliyor. Bu sebepten dolayı yine simülasyon programı da geniş fakat yüzeysel kalmaktadır.

Robot hareketini simüle etmek için herşeyden önce robotun kendisini ekranda görüntülemek gerekmektedir. Fakat her hangi bir sınırlamaya tabi tutulmamış ve bazı değerleri verilmiş her hangi bir robotu ekrana çizmek için ne kadar büyük programlara ihtiyaç olduğu açıktır. Bir de bu robotların yapacağı hareketleri gerçek manada simüle etmek için de ne kadar fonksiyonel bir programa ihtiyaç duyulacağı da anlaşılabilir. Bu ise bu tezin kapsamını oldukça aşmaktadır. Bu sebepten dolayı sadece düzlemsel boyutta çalışma yapılması bir zorunluluk olmaktadır. Ayrıca robotun uzuvları sadece bir çizgiyle ifade edilirken robotun hareketinin simülasyonunda düz kinematik analiz açısından simülasyon yapılmaktadır.

5.4 Gelecekte Yapılabilecek Çalışmalar

Yapılan bu çalışmayla istenilen her hangi bir robotun hareket denklemlerini veren ve robotun hareketini simüle eden bilgisayar programları geliştirilmeye çalışılmıştır. Yazılan bu programlar yardımıyla elde edilen sembolik formdaki hareket denklemleri uzun bir hal almaktadır. Bunu önlemek için sembolik ifadeler üzerinde matematiksel işlemler yapabilen programlar veya fonksiyonlar geliştirilebilir. Ayrıca hareketin simülasyonunda kullanılan formüller sembolik formüllerden ayrı olarak elde edilmektedir . Bu ise ikinci bir hesaplama

demektir. Bunu önlemek için sembolik formda elde edilen denklemleri nümerik hale getirebilen ve buradan simülasyona geçebilen programlar üzerinde de çalışmalar yapılabilir. Ayrıca simülasyon programı sadece düz kinematik analiz açısından hareketi simüle etmektedir .Yani mafsallara belirli hareketler verilince tutucu en son hangi konumu alır bu görebilmektedir. Ters kinematik analiz açısından çizim yapabilen bir programın geliştirilmesi için de çalışmalar yapılabilir.



KAYNAKLAR

- 1 - **Ardayfio, D. D.** , et al. (1983), MICRAS: microcomputer interactive codes for robot analysis and simulation, Mechanical and Aerospace Engineering Department, University of Missouri-Rolla, Mo.
- 2 - **Asada, H. , Slotine, J. J.** , (1986), Robot analysis and control, Wiley - Interscience
- 3 - **Barr, A. , and Feigenbaum, E. A.** (1982), The Handbook of Artificial Intelligence, Vol. II, Heuris technical Press, Stanford, Calif. , and William Kaufmann, Los Angeles, Calif. , pp. 143-9.
- 4 - **Borisenko, A. I. , Taradov, I. E.** , (1968), Vector and tensor analysis with applications, Dover
- 5 - **Cesareo, G. , Nicolo, F. , and Nicosia, S.** (1981), DYMIR: generations automatica di modelli dinamici di robot industriali, RI.81.0, Istituto di Automatica, Universita di Roma, Italy.
- 6 - **Coban, M.** (1994), Bütün yönleriyle turbo C, İstanbul
- 7 - **Denavit, J. and Hartenberg, R. S.** (1955), A kinematic notation for lower-pair mechanisms based on matrices, Journal of Applied Mechanics, pp. 215-221
- 8 - **Dilon, S. R.** (1973), Computer assisted equation generation in linkage dynamics, PhD thesis, Ohio State University.
- 9 - **Dombre, E. , Borrel, P. , and Liegois, A.,** (1985), A CAD system for programming and simulating robots' actions. In Aleksander (ed), Computing Techniques for robots. Kogan Page, pp.223-47.
- 10 - **Fu, K. S. , Gonzales, R. C. , Lee, C. S. G.** , (1987), Robotics, control, sensing, vision and intelligence, McGraw-Hill
- 11 - **Goldstein, H.,** (1980), Classical Mechanics, Addison-Wesley
- 12 - **Gosselin, C. M. , Perreault, L. , Vaillantcourt, C.** (1995), Simulation and computer- aided- kinematic desing of three-degree-of-freedom spherical parallel manipulators. Jour. of Robotic Sys.-1995,pp.857-869.

- 13 - **Hearn, A. C.** (1973), REDUCE 2 Users Manual, Symbolic Computation Group report No. UCP-19, University of Utah.
- 14 - **Hodges, B.** (1992), Industrial robotics, Newnes
- 15 - **Howie, Ph.** (1984), Graphic simulation for off-line programming, Robotics Today, February 1984, pp. 63-6
- 16 - **Khalil, W.** (1978), Contribution a la commande automatique des manipulateurs avec l'aide d'un modele mathematique des mecanismes, These de dovtorat d'Etat, Universite des Science et Techniques du Languedoc, Montpellier.
- 17 - **Koivo, A. J.** , (1989), Fundamentals for control of robotic manipulators, Wiley
- 18 - **Komisyon,** (1992), Borland turbo C++ 3.0 User's guide - özel basım, İstanbul
- 19 - **Megahed, S.** (1984a), MPLCS: programme d'obtention automatique de la matrice de passage homogene des robots manipulateurs, Research report 84.044, LAAS-CNRS, Toulouse.
- 20 - **Megahed, S.** (1984b), MGLCS: programme d'obtention automatique du modele geometrique direct des robots manipulateurs, Research report 84.045, LAAS-CNRS, Toulouse.
- 21 - **Megahed, S.** (1984c), MDFLCS: programme d'obtention automatique du modele differantial direct et de la matrice jacobienne des robots manipulateurs, Research report 84.047, LAAS-CNRS, Toulouse.
- 22 - **Megahed, S.** (1984d), MDLCS: programme d'obtention automatique du modele dynamique des robots manipulateurs, Research report 84.046, LAAS-CNRS, Toulouse.
- 23 - **Megahed, S.** (1984e), MGLCC: programme d'obtention automatique du modele geometrique des robots manipulateurs ayant une structure de chaine complexe, Research report, LAAS-CNRS, Toulouse.
- 24 - **Megahed, S.** (1984f), MDLAT: programme d'obtention automatique du modele dynamique des robots manipulateurs ayant une structure d'arbre

- topologique, Research report, LAAS-CNRS, Toulouse.
- 25 - **Megahed, S.** (1993), Principles of Robot Modelling And Simulation, John Wiley & Sons, England.
- 26 - **Paul, R. P.** , (1981), Robot manipulators, mathematics, programming, and control- The computer control of robot manipulators, MIT Press
- 27 - **Sata, T. , Kimura, F. , and Amano, A.** (1981), Robot simulation system as a task programming tool. In Proceeding of 11th ISIR, Tokyo, Japan.
- 28 - **Schilling, R. J.** (1990), Fundamentals of Robotics, analysis and control, Prentice-Hall
- 29 - **Shabana, A. A.** , (1989), Dynamics of multibody systems, Wiley-Interscience
- 30 - **Spong, M. W. , Vidyasagar, M.** , (1989), Robot dynamics and control, Wiley
- 31 - **Stauffer, R.** (1984), Robot system simulation. Robotics Today, June 1984, pp.81-90.
- 32 - **Sturges, R.** (1973), Teleoperator arm design program, Report No. 3.27.46 MIT, Drapper Lab. , Cambridge.
- 33 - **Suhubi, E. S.** ,(1988), Rijid cisimler dinamiği, İ.T.Ü.
- 34 - **Summers, Ph. D. , and Grossman, D. D.** (1984), XPROBE: An experimental system for programming robots by example, Int. J. Robotics research, 3(1), 25-39
- 35 - **Turhan, Ö.** (1994) , Robot Kinematiği ve Dinamiği Ders Notları , İTÜ.
- 36 - **Weck, M. , Eversheim, W. , and Zuehlke, D.** (1981), ROBEX: an off-line programming tool of industrial robots. In Proceedings of 11th ISIR, Tokyo, Japan.
- 37 - **Wittenburg, J.** , (1977), Dynamics of systems of rigid bodies, Teubner
- 38 - **Yong, Y. F. , and Bonney, M. C.** (1983), Simulation-preventing some nasty snarl-UPS. In A Decade of Robotics-A Special 10th Anniversary Issue, IFS Publication, Kempston, Bedford.
- 39 - **Yoshikawa, T.** , (1990), Foundations of robotics, analysis and control,

MIT Press

- 40 - **Zabala, I. J.** (1978), *Commande des robots manipulateurs a partir de la modelisation de leur dynamique*, These de docteur de specialite EEA, Universite Paul Sabatier, Toulouse.



Ek - 1**DHTABLO Program Kodu**

```
void dhtablo()
{
    FILE *fdh;

    /* D-H.DAT dosyasının açılması. */

    clrscr();
    if((fdh=fopen("dh.dat","w"))==NULL)
    {
        printf("dh.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

    /* mafsalsayısının girilmesi. */

    do
    {
        clrscr ();
        printf("Mafsalsayısını Giriniz [1-7]:");
        scanf("%d",&n);
    }
    while(n<1 || n>7);
    fprintf(fdh,"%d",n);

    /* DH parametrelerinin okunması ve DH.DAT dosyasına kaydedilmesi. */
```

```

{
  int i,j;/* döngü değeri. */
  printf("\n\n");
  for(i=0;i<n;i++)
  {
    clrscr();
    printf("          DH Parametrelerini giriniz.\n");
    printf("          En Fazla 10 Karakter.\n\n");
    printf("          %d.mafsal için\n",i+1);
    printf("z(%d) den z(%d) ye x(%d) boyunca ölçülen uzaklık (a) [mm]:",i,i+1,i);
    scanf("%10s",dh[i][0]);
    printf("z(%d) den z(%d) ye x(%d) çevresinde ölçülen açı (α) [deg]:",i,i+1,i);
    scanf("%10s",dh[i][1]);
    printf("x(%d) den x(%d) ye z(%d) boyunca ölçülen uzaklık (d)
[mm]:",i,i+1,i+1);
    scanf("%10s",dh[i][2]);
    printf("x(%d) den x(%d) ye z(%d) çevresinde ölçülen açı (θ)
[deg]:",i,i+1,i+1);
    scanf("%10s",dh[i][3]);
    printf("(%d).Mafsalın Tipi ( r veya p ) :",i+1);
    scanf("%1s",tip[i]);
    for(j=0;j<4;j++)
      fprintf(fdh, "%10s",dh[i][j]);
  }
  fclose(fdh);
}
}

```

Ek - 2**TXMAT Program Kodu**

```
void txmat()
{
    FILE *ftx;
    int i,j,k;

    /* TX.DAT dosyasının açılması. */
    clrscr();
    if((ftx=fopen("tx.dat","w"))==NULL)
    {
        printf("tx.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

    /* tx matrislerine dh tablosundaki değerlerin atanması. */

    for(i=0;i<n;i++)
    {
        char *pdh;

        for(j=0;j<4;j++)
            for(k=0;k<4;k++)
            {
                strcpy(t1[j][k],"0");
                strcpy(t2[j][k],"0");
                strcpy(t3[j][k],"0");
            }
    }
}
```

```
strcpy(t4[j][k], "0");  
}
```

```
strcpy(t1[0][0], "1");  
strcpy(t1[1][1], "1");  
strcpy(t1[2][2], "1");  
strcpy(t1[3][3], "1");  
strcpy(t2[0][0], "1");  
strcpy(t2[3][3], "1");  
strcpy(t3[0][0], "1");  
strcpy(t3[1][1], "1");  
strcpy(t3[2][2], "1");  
strcpy(t3[3][3], "1");  
strcpy(t4[2][2], "1");  
strcpy(t4[3][3], "1");
```

```
strcpy(t1[0][3], dh[i][0]);
```

```
if(!strcmp(dh[i][1], "0"))
```

```
{  
    strcpy(t2[1][1], "1");  
    strcpy(t2[2][2], "1");  
}
```

```
else
```

```
{  
    if(!strcmp(dh[i][1], "90"))  
    {  
        strcpy(t2[1][2], "-1");  
        strcpy(t2[2][1], "1");  
    }  
}
```

```
else
{
if(!strcmp(dh[i][1], "-90"))
{
strcpy(t2[1][2], "1");
strcpy(t2[2][1], "-1");
}
else
{
pdh=dh[i][1];
if(pdh[0]!='-')
{
strcpy(t2[1][1], "cos");
strcat(t2[1][1], pdh+1);

strcpy(t2[1][2], "sin");
strcat(t2[1][2], pdh+1);

strcpy(t2[2][1], "-sin");
strcat(t2[2][1], pdh+1);

strcpy(t2[2][2], "cos");
strcat(t2[2][2], pdh+1);
}
else
{
strcpy(t2[1][1], "cos");
strcat(t2[1][1], pdh);

strcpy(t2[1][2], "-sin");
```

```

        strcat(t2[1][2],pdh);

        strcpy(t2[2][1],"sin");
        strcat(t2[2][1],pdh);

        strcpy(t2[2][2],"cos");
        strcat(t2[2][2],pdh);
    }
}
}
}

```

```

strcpy(t3[2][3], dh[i][2]);

if(!strcmp(dh[i][3],"0"))
{
    strcpy(t4[0][0], "1");
    strcpy(t4[1][1], "1");
}
else
{
    pdh=dh[i][3];
    if(pdh[0]!='-')
    {
        strcpy(t4[0][0], "c");
        strcat(t4[0][0],pdh+1);

        strcpy(t4[0][1], "s");
        strcat(t4[0][1],pdh+1);
    }
}
}
}
}

```

```

        strcpy(t4[1][0], "-s");
        strcat(t4[1][0], pdh+1);

        strcpy(t4[1][1], "c");
        strcat(t4[1][1], pdh+1);
    }
    else
    {
        strcpy(t4[0][0], "c");
        strcat(t4[0][0], pdh);

        strcpy(t4[0][1], "-s");
        strcat(t4[0][1], pdh);

        strcpy(t4[1][0], "s");
        strcat(t4[1][0], pdh);

        strcpy(t4[1][1], "c");
        strcat(t4[1][1], pdh);
    }
}

/* tx matrislerinin tx.dat dosyasına kaydedilmesi. */

for(j=0;j<4;j++)
    for(k=0;k<4;k++)
        fprintf(ftx, "%14s", t1[j][k]);
for(j=0;j<4;j++)
    for(k=0;k<4;k++)
        fprintf(ftx, "%14s", t2[j][k]);
for(j=0;j<4;j++)

```

```

        for(k=0;k<4;k++)
            fprintf(ftx, "%14s", t3[j][k]);
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            fprintf(ftx, "%14s", t4[j][k]);
    }/* i döngü sonu. */
    fclose(ftx);
}

```

Ek - 3

HTMMAT Program Kodu

```

void htmmat()
{
    FILE *ftx;
    FILE *fihtm;
    FILE *fhtm;
    int i,j,k;
    char kar;

    /* TX.DAT dosyasının açılması. */
    if((ftx=fopen("tx.dat", "r"))==NULL)
    {
        printf("tx.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

    /* ***** HTM MATRİSLERİNİN OLUŞTURULMASI. ***** */

```

```

{
/* HTM.DAT dosyasının açılması.*/
    if((fhtm=fopen("htm.dat","w"))==NULL)
    {
        printf("htm.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

/* [R,0]T matrisinin HTM.DAT dosyasına kaydedilmesi. */
    clrscr();
    do
    {
        printf("R eksen takımı için [R,0]T matrisini girmek istiyor musunuz? [e/h]:" );
        kar=getche();
    }while(kar!='e'&& kar!='h');

    clrscr();
    if(kar=='e')
    {
        char *r[4][4];
        for(j=0;j<4;j++)
            for(k=0;k<4;k++)
                r[j][k]=new char[11];
        printf("R eksen takımı için [R,0]T matrisinin oluşturulması (max. 10
kar.).\n\n");
        for(j=0;j<4;j++)
            for(k=0;k<4;k++)
            {
                printf("[R,0]T [%d,%d] = ",j+1,k+1);
                scanf("%10s",r[j][k]);
            }
    }
}

```

```

        strcpy(htm[0][i][k],r[j][k]);
    }
    delete[] r;
}
else
{
    char r[4][4][2]={"1","0","0","0","0","1","0","0","0","0","1","0","0","0","0","1"};
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            strcpy(htm[0][i][k],r[j][k]);
}
for(i=1;i<=n;i++)
{
    i1=j1=k1=4;

    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            fscanf(ftx,"%14s",t1[j][k]);
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            fscanf(ftx,"%14s",t2[j][k]);
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            fscanf(ftx,"%14s",t3[j][k]);
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            fscanf(ftx,"%14s",t4[j][k]);

    /* homojen dönüşüm matrislerinin elde edilmesi.*/
}

```

```

for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
strcpy(a[j][k],t1[j][k]);
strcpy(b[j][k],t2[j][k]);
}
mat();
for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
strcpy(a[j][k],c[j][k]);
strcpy(b[j][k],t3[j][k]);
}
mat();
for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
strcpy(a[j][k],c[j][k]);
strcpy(b[j][k],t4[j][k]);
}
mat();

for(j=0;j<4;j++)
for(k=0;k<4;k++)
strcpy(htm[i][j][k],c[j][k]);

}
fclose(fix);

/* [n,E]T matrisinin ekrandan okunması. */

```

```

clrscr();
do
{
    printf("E eksen takımı için [%d,E]T matrisini girmek istiyor musunuz?
[e/h]:",n );
    kar=getche();
}while(kar!='e'&& kar!='h');

clrscr();
if(kar=='e')
{
    char *r[4][4];
    for(j=0;j<4;j++)
    for(k=0;k<4;k++)
    r[j][k]=new char[11];
    printf("E eksen takımı için [%d,E]T matrisinin oluşturulması (max. 10
kar.).\n\n",n);
    for(j=0;j<4;j++)
    for(k=0;k<4;k++)
    {
        printf("[%d,E]T [%d,%d] = ",n,j+1,k+1);
        scanf("%10s",r[j][k]);
        strcpy(htm[n+1][j][k],r[j][k]);
    }
    delete[] r;
}
else
{
    char r[4][4][2]={"1","0","0","0","0","1","0","0","0","0","1","0","0","0","0","1"};
    for(j=0;j<4;j++)

```

```

        for(k=0;k<4;k++)
            strcpy(htm[n+1][j][k],r[j][k]);
    }
    /* HTM matrislerinin HTM.DAT dosyasına kaydedilmesi.*/
    for(i=0;i<n+2;i++)
    {
        for(j=0;j<4;j++)
            for(k=0;k<4;k++)
            {
                fprintf(fhtm,"%35s",htm[i][j][k]);
            }
    }
    fclose(fhtm);
} /* HTM matrisleri oluşturuldu. */
}

```

Ek - 4

Hİ Program Kodu

```

void hi()
{
    FILE *fh;
    int i,j,k;

    /* H.DAT dosyasının açılması.*/
    if((fh=fopen("h.dat","w"))==NULL)
    {
        printf("h.dat dosyası açılmadı!!!");
    }
}

```

```

    getch();
    exit(0);
}

/* ***** H MATRİSİNİN EKRANDAN OKUNMASI. ***** */
for(i=0;i<n;i++)
{
    char h[4][4][26];
    clrscr();
    printf("          Psödo Atalet Tansörlerinin Oluşturulması (max. 25
kar.)\n\n\n");
    printf("          [%d]. Mafsalın Psödo Atalet Tansörü.\n\n",i+1);
    for(j=0;j<4;j++)
    for(k=0;k<4;k++)
    {
        printf("    H%d [%d,%d] = ",i+1,j+1,k+1);
        scanf("%25s",h[j][k]);
    }
    for(j=0;j<4;j++)
    for(k=0;k<4;k++)
        fprintf(fh,"%25s",h[j][k]);
}
fclose(fh);
}

```

Ek - 5**Mİ Program Kodu**

```
void mi()
{
    FILE *fm;
    FILE *fh;
    int i,j,k,d1,d2;
    char iv[7][4]={"A1","A2","A3","A4","A5","A6","A7"};
    i1=j1=k1=4;

    /* H.DAT dosyasının açılması. */
    clrscr();
    if((fh=fopen("h.dat","r"))==NULL)
    {
        printf("h.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

    /* M.DAT dosyasının açılması. */
    clrscr();
    if((fm=fopen("m.dat","w"))==NULL)
    {
        printf("m.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

    /* M atalet matrisinin hesaplanması. */
    {
        for(i=1;i<=n;i++)
```

```
{
for(j=i;j<=n;j++)
{
char m[7500]="0";
char ara[4][4][1000];
for(k=j;k<=n;k++)
{
    uij(k,j);
    fseek(fh,(k-1)*400,0);
    for(d1=0;d1<4;d1++)
    for(d2=0;d2<4;d2++)
    {
        strcpy(a[d1][d2],c[d1][d2]);
        fscanf(fh,"%25s",b[d1][d2]);
    }
    mat ();
    for(d1=0;d1<4;d1++)
    for(d2=0;d2<4;d2++)
        strcpy(ara[d1][d2],c[d1][d2]);
    uij(k,i);
    for(d1=0;d1<4;d1++)
    for(d2=0;d2<4;d2++)
    {
        strcpy(a[d1][d2],ara[d1][d2]);
        strcpy(b[d1][d2],c[d2][d1]);
    }
    mat();
    for(d1=0;d1<4;d1++)
    {
        if(strcmp(c[d1][d1],"0"))
```

```
{
    if(!strcmp(m,"0"))
    {
        strcpy(m,"");
        strcat(m,c[d1][d1]);
    }
    else
    {
        if(c[d1][d1][0]=='-')
        {
            strcat(m,c[d1][d1]);
        }
        else
        {
            strcat(m,"+");
            strcat(m,c[d1][d1]);
        }
    }
}
strcat(m,"]");
strcat(m,iv[j-1]);
}/* k */
fprintf(fm,"%7500s",m);
}/* j */
}/* i */
}/* mij fonk. sonu.*/
fclose(fm);
fclose(fh);
}
```

Ek - 6**HCORİ Program Kodu**

```
void hcori()
{
    FILE *fh;
    FILE *fhi;
    int i,j,k,m,d1,d2;
    char ara[4][4][1000];
    char v[10][4]={"V1","V2","V3","V4","V5","V6","V7","V8","V9","V10",};
    i1=j1=k1=4;
    /* H.DAT dosyasının açılması. */
    clrscr();
    if((fh=fopen("h.dat","r"))==NULL)
    {
        printf("h.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }
    /* HI.DAT dosyasının açılması. */
    clrscr();
    if((fhi=fopen("hi.dat","w"))==NULL)
    {
        printf("hi.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }
    for(i=1;i<=n;i++)
    {
```

```
int d;
for(j=1;j<=n;j++)
{
    for(m=1;m<=n;m++)
    {
        char hijm[7500]={"0"};
        /* hijm lerin hesaplanması */
        /* k değerinin bulunması k=max(i,j,m)*/
        if(i>=j)
        {
            if(j>=m)
            {
                d=i;
            }
            else
            {
                if(i>=m)
                {
                    d=i;
                }
                else
                {
                    d=m;
                }
            }
        }
        else
        {
            if(j>=m)
            {
                d=j;
            }
            else
            {
                d=m;
            }
        }
        for(k=d;k<=n;k++)
```

```

{
    uijk(k,j,m);
    for(d1=0;d1<4;d1++)
        for(d2=0;d2<4;d2++)
            strcpy(a[d1][d2],c[d1][d2]);
    fseek(fh,(k-1)*400,0);
    for(d1=0;d1<4;d1++)
        for(d2=0;d2<4;d2++)
            fscanf(fh,"%25s",b[d1][d2]);
    mat ();
    for(d1=0;d1<4;d1++)
        for(d2=0;d2<4;d2++)
            strcpy(ara[d1][d2],c[d1][d2]);
    uij(k,i);
    for(d1=0;d1<4;d1++)
        for(d2=0;d2<4;d2++)
            strcpy(b[d1][d2],c[d2][d1]);
    for(d1=0;d1<4;d1++)
        for(d2=0;d2<4;d2++)
            strcpy(a[d1][d2],ara[d1][d2]);
    mat ();
    for(d1=0;d1<4;d1++)
    {
        if(strcmp(c[d1][d1],"0"))
        {
            if(!strcmp(hijm,"0"))
            {
                strcpy(hijm,"[]");
                strcat(hijm,c[d1][d1]);
            }
        }
    }
}

```

```

else
{
    if(c[d1][d1][0]=='-')
    {
        strcat(hijm,c[d1][d1]);
    }
    else
    {
        strcat(hijm,"+");
        strcat(hijm,c[d1][d1]);
    }
}
}
}/*d1*/
}/* k */
{
    strcat(hijm,"**");
    strcat(hijm,v[j-1]);
    strcat(hijm,"**");
    strcat(hijm,v[m-1]);
}
fprintf(fhi,"%7500s",hijm);
}/*m*/
}/*j*/
}/*i*/
fclose(fh);
fclose(fhi);
}

```

Ek - 7**CKOORD Program Kodu**

```
void ckoord()
{
    FILE *fr;
    FILE *fc;
    char m[10][4]={"m1","m2","m3","m4","m5","m6","m7","m8","m9","m10"};
    char g[1][4][6]; /* ivme vektörü. */
    int i,j,k,l;
    /* R.DAT dosyasının açılması. */
    clrscr();
    if((fr=fopen("r.dat","r"))==NULL)
    {
        printf("r.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }
    /* C.DAT dosyasının açılması. */
    clrscr();
    if((fc=fopen("c.dat","w"))==NULL)
    {
        printf("c.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }
    /* g vektörünün okunması. */
    for(j=0;j<4;j++)
        fscanf(fr,"%5s",g[0][j]);
```

```

/* g yer çekim ivme vektörü ile U ij matrisinin çarpılması. */
for(i=1;i<=n;i++)
{
    char ci[1000]={"0"}
;
    /* Uij matrislerinin hesaplanması. */
    for(l=i;l<=n;l++)
    {
        char *pc;

        uij(l,i);

        for(j=0;j<4;j++)
            for(k=0;k<4;k++)
                strcpy(b[j][k],c[j][k]);
        for(j=0;j<4;j++)
            strcpy(a[0][j],g[0][j]);

        /* g vektörü ile Uij matrisinin çarpılması. */
        i1=1;j1=1;k1=4;
        mat();
        for(j=0;j<4;j++)
            strcpy(a[0][j],c[0][j]);
        fseek(fr,(20+(l-1)*40),0);
        for(j=0;j<4;j++)
            fscanf(fr,"%10s",b[j][0]);
        i1=j1=1;k1=4;
        mat();

        /* elde edilen stringin '-mi' ile çarpılması. */

```

```
pc=c[0][0];
if(!strcmp(ci,"0"))
{
    if(pc[0]=='-')
    {
        strcpy(ci,m[l-1]);
        strcat(ci,"*");
        strcat(ci,pc+1);
    }
    else
    {
        strcpy(ci,"-");
        strcat(ci,m[l-1]);
        strcat(ci,"*");
        strcat(ci,pc);
    }
}
else
{
    if(pc[0]=='-')
    {
        strcat(ci,"+");
        strcat(ci,m[l-1]);
        strcat(ci,"*");
        strcat(ci,pc+1);
    }
    else
    {
        strcat(ci,"-");
        strcat(ci,m[l-1]);
    }
}
```

```
        strcat(ci,"*");
        strcat(ci,pc);
    }
}
}
    fprintf(fc,"%1000s",ci);
}
fclose(fr);
fclose(fc);
}
```




```
else
{
    dd=1;
    if(pb[0]=='-')
    {
        strcat(d,pb);
    }
    else
    {
        strcat(d,"+");
        strcat(d,pb);
    }
}
}
else
{
    if(!strcmp(pb,"1"))
    {
        if(!strcmp(d,"0"))
        {
            strcpy(d,pa);
        }
        else
        {
            dd=1;
            if(pa[0]=='-')
            {
                strcat(d,pa);
            }
            else
```

```
{
    strcat(d,"+");
    strcat(d,pa);
}
}
}
else
{
    if(!strcmp(pa,"-1"))
    {
        if(!strcmp(d,"0"))
        {
            if(pb[0]=='-')
            {
                strcpy(d,pb+1);
            }
            else
            {
                strcpy(d,"-");
                strcat(d,pb);
            }
        }
    }
    else
    {
        dd=1;
        if(pb[0]=='-')
        {
            strcat(d,"+");
            strcat(d,pb+1);
        }
    }
}
```

```
        else
        {
            strcat(d,"-");
            strcat(d,pb);
        }
    }
}
else
{
    if(!strcmp(pb,"-1"))
    {
        if(!strcmp(d,"0"))
        {
            if(pa[0]!='-')
            {
                strcpy(d,pa+1);
            }
            else
            {
                strcpy(d,"-");
                strcat(d,pa);
            }
        }
    }
    else
    {
        dd=1;
        if(pa[0]!='-')
        {
            strcat(d,"+");
            strcat(d,pa+1);
        }
    }
}
```

```
    }  
    else  
    {  
        strcat(d,"-");  
        strcat(d,pa);  
    }  
}  
else  
{  
    if(!strcmp(d,"0"))  
    {  
        if(pa[0]=='-')  
        {  
            if(pb[0]=='-')  
            {  
                strcpy(d,pa+1);  
                strcat(d,"*");  
                strcat(d,pb+1);  
            }  
            else  
            {  
                strcpy(d,pa);  
                strcat(d,"*");  
                strcat(d,pb);  
            }  
        }  
    }  
    else  
    {  
        if(pb[0]=='-')
```

```
{
    strcpy(d,"-");
    strcat(d,pa);
    strcat(d,"*");
    strcat(d,pb+1);
}
else
{
    strcpy(d,pa);
    strcat(d,"*");
    strcat(d,pb);
}
}
else
{
    {
        dd=1;
    }
    if(pa[0]=='-')
    {
        if(pb[0]=='-')
        {
            strcat(d,"+");
            strcat(d,pa+1);
            strcat(d,"*");
            strcat(d,pb+1);
        }
        else
        {
```

```
        strcat(d,pa);
        strcat(d,"**");
        strcat(d,pb);
    }
}
else
{
    if(pb[0]=='-')
    {
        strcat(d,"-");
        strcat(d,pa);
        strcat(d,"**");
        strcat(d,pb+1);
    }
    else
    {
        strcat(d,"+");
        strcat(d,pa);
        strcat(d,"**");
        strcat(d,pb);
    }
}
}
}

    /*pb=-1 parentezi.*/
    /*pa=-1 parentezi.*/
    /*pb=1 parentezi.*/
    /*pa=1 parentzi.*/
    /*pb=0 parentezi.*/
    /*pa=0 parentezi.*/
```

```

    }/*k döng parentezi.*/
    if(dd==1)
    {
        strcpy(c[i][j], "(");
        strcat(c[i][j], d);
        strcat(c[i][j], ")");
    }
    else
        strcpy(c[i][j], d);
        delete[] d;
}/*j döng parentezi.*/
}/*fonks.parentezi.*/

```

Ek - 9

UİJ Program Kodu

```

void uij (int d,int e)
{
    FILE *fhtm;
    int j,k,l;
    char r[4][4][3]={"0","-1","0","0","1","0","0","0","0","0","0","0","0","0","0"};
    char p[4][4][2]={"0","0","0","0","0","0","0","0","0","0","0","0","0","0","1","0"};
    i1=j1=k1=4;
    /* HTM.DAT dosyasının açılması.*/
    if((fhtm=fopen("htm.dat","r"))==NULL)
    {
        printf("htm.dat dosyası açılmadı!!!");
        getch();
    }
}

```

```

        exit(0);
    }
fseek(fhfm,0,0);
for(j=0;j<4;j++)
for(k=0;k<4;k++)
    fscanf(fhfm,"%35s",a[j][k]);
for(j=0;j<4;j++)
for(k=0;k<4;k++)
    fscanf(fhfm,"%35s",b[j][k]);
mat();
for(l=2;l<=e;l++)
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
    strcpy(a[j][k],c[j][k]);
for(j=0;j<4;j++)
for(k=0;k<4;k++)
    fscanf(fhfm,"%35s",b[j][k]);
mat();
}
for(j=0;j<4;j++)
for(k=0;k<4;k++)
    strcpy(a[j][k],c[j][k]);
if(!strcmp(dh[e-1][3],"0"))
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
    strcpy(b[j][k],p[j][k]);
}
else

```



```

        /* HTM.DAT dosyasının açılması.*/
        if((fhtm=fopen("htm.dat", "r"))==NULL)
        {
            printf("htm.dat dosyası açılmadı!!!");
            getch();
            exit(0);
        }
fseek(fhtm,0,0);
for(j=0;j<4;j++)
    for(k=0;k<4;k++)
        fscanf(fhtm, "%35s", a[j][k]);
for(j=0;j<4;j++)
    for(k=0;k<4;k++)
        fscanf(fhtm, "%35s", b[j][k]);
mat();
for(l=2;l<=f;l++)
{
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            strcpy(a[j][k],c[j][k]);
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)
            fscanf(fhtm,"%35s",b[j][k]);
            mat();
}
for(j=0;j<4;j++)
    for(k=0;k<4;k++)
        strcpy(a[j][k],c[j][k]);
        if(!strcmp(dh[f-1][3], "0"))
        {

```

```

for(j=0;j<4;j++)
  for(k=0;k<4;k++)
    strcpy(b[j][k],p[j][k]);
}
else
{
  for(j=0;j<4;j++)
    for(k=0;k<4;k++)
      strcpy(b[j][k],r[j][k]);
}
mat();
for(l=f+1;l<=e;l++)
{
  for(j=0;j<4;j++)
    for(k=0;k<4;k++)
      strcpy(a[j][k],c[j][k]);
  for(j=0;j<4;j++)
    for(k=0;k<4;k++)
      fscanf(fhfm,"%35s",b[j][k]);
  mat();
}
for(j=0;j<4;j++)
  for(k=0;k<4;k++)
    strcpy(a[j][k],c[j][k]);
if(!strcmp(dh[e-1][3],"0"))
{
  for(j=0;j<4;j++)
    for(k=0;k<4;k++)
      strcpy(b[j][k],p[j][k]);
}

```

```

else
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
strcpy(b[j][k],r[j][k]);
}
mat();
for(l=e+1;l<=d;l++)
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
strcpy(a[j][k],c[j][k]);
for(j=0;j<4;j++)
for(k=0;k<4;k++)
fscanf(fhtm,"%35s",b[j][k]);
mat();
}
} /* fonks. sonu */

```

Ek - 11

MAIN (Hareket Denklemleri Fonksiyonu) Program Kodu

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
char a[4][4][750];
char b[4][4][750];
char c[4][4][750];

```

```
char t1[4][4][15];
char t2[4][4][15];
char t3[4][4][15];
char t4[4][4][15];
char htm[9][4][4][36];
char dh[7][4][11];
char tip[7][2];
int i1,j1,k1,n;
#include"dhtablo.h"
#include"txmat.h"
#include"mat.h"
#include"htmmt.h"
#include"hi.h"
#include"uij.h"
#include"uijk.h"
#include"mi.h"
#include"migor.h"
#include"hcori.h"
#include"hcorigor.h"
#include"rgvek.h"
#include"ckoord.h"
#include"ckoordgo.h"
#include"prin.h"
#include"miprm.h"
#include"hcoripm.h"
#include"ckoordpr.h"
int main(void)
{
    FILE *fdh;
    char kar;
```

```

int i,j;
clrscr();
printf("DH Tablosunu Dosyadan Okumak İçin 'd' Klavyeden Girmek İçin 'k' :");
kar=getche();
if(kar=='d')
{
clrscr();
if((fdh=fopen("dh.dat","r"))==NULL)
{
printf("dh.dat dosyası açılmadı.Lütfen yeni bir DH Tablosu
oluşturun.");
getch();
clrscr();
dhtablo();
}
fscanf(fdh,"%d",&n);/* mafsal sayısının okunması.*/
/* DH parametrelerinin okunması.*/
{
for(i=0;i<n;i++)
for(j=0;j<4;j++)
{
fscanf(fdh,"%10s",dh[i][j]);
}
fclose(fdh);
}
}
else
{
clrscr();
dhtablo();
}

```

```
}  
txmat();  
htmmat();  
hi();  
mi();  
hcori();  
rgvek();  
ckoord();  
  clrscr();  
  printf("Denklem Elemanlarının Görüntülenmesi : g  Yazılması : y\n");  
  printf("Seciminiz :");  
  kar=getch();  
if(kar=='g')  
{  
  migor();  
  hcorigor();  
  ckoordgo();  
}  
else  
{  
  mipm();  
  hcoripm();  
  ckoordpr();  
}  
return 0;  
}
```

Ek - 12**NDHTABLO Program Kodu**

```
void ndhtablo()
{
    FILE *fndh;
    FILE *ftip;
    /* NDH.DAT dosyasının açılması. */
    clrscr();
    if((fndh=fopen("ndh.dat","w"))==NULL)
    {
        printf("ndh.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }
    /* TİP.DAT dosyasının açılması. */

    clrscr();
    if((ftip=fopen("tip.dat","w"))==NULL)
    {
        printf("tip.dat dosyası açılmadı!!!");
        getch();
        exit(0);
    }

    /* mafsal sayısının girilmesi. */
    do
    {
        clrscr ();
```

```

printf("Mafsal Sayısını Giriniz [1-7]:");
scanf("%f",&n);
}
while(int(n)<1 || int(n)>7);
fprintf(fndh,"%f\t",n);

```

```

/* DH parametrelerinin okunması ve NDH.DAT dosyasına kaydedilmesi. */

```


```

{
int i,j;

printf("\n\n");
for(i=0;i<int(n);i++)
{
clrscr();
printf("          DH Parametrelerini giriniz.\n");
printf("          %d.mafsal için\n\n",i+1);
printf("z(%d) den z(%d) ye x(%d) boyunca ölçülen uzaklık (a) [mm]:",i,i+1,i);
scanf("%f",&ndh[i][0]);
printf("z(%d) den z(%d) ye x(%d) çevresinde ölçülen açı (Ó) [deg]:",i,i+1,i);
scanf("%f",&ndh[i][1]);
printf("x(%d) den x(%d) ye z(%d) boyunca ölçülen uzaklık (d)
[mm]:",i,i+1,i+1);
scanf("%f",&ndh[i][2]);
printf("x(%d) den x(%d) ye z(%d) çevresinde ölçülen açı (Ú)
[deg]:",i,i+1,i+1);
scanf("%f",&ndh[i][3]);
for(j=0;j<4;j++)
fprintf(fndh,"%f\t",ndh[i][j]);
}
fclose(fndh);

```

```
clrscr();
printf("\n\n\n");
printf(" Mafsal Tiplerini Giriniz.\n");
printf("\n\n\n");
for(i=0;i<int(n);i++)
{
    printf(" %d. mafsalın tipi :",i+1);
    scanf("%s",tip[i]);
    fprintf(ftip,"%s",tip[i]);
}
fclose(ftip);
}
}
```



Ek - 13**NHTMMAT Program Kodu**

```
void nhtmmtat()
```

```
{
```

```
    FILE *fnhtm;
```

```
    int i,j,k;
```

```
    char kar;
```

```
{
```

```
    /* NHTM.DAT dosyasının açılması. */
```

```
    clrscr();
```

```
    if((fnhtm=fopen("nhtm.dat","w"))==NULL)
```

```
    {
```

```
        printf("nhtm.dat dosyası açılmadı!!!");
```

```
        getch();
```

```
        exit(0);
```

```
    }
```

```
    /* [R,0]T matrisinin ekrandan okunması. */
```

```
    clrscr();
```

```
    do
```

```
    {
```

```
        printf("R eksen takımı için [R,0]T matrisini girmek istiyor musunuz? [e/h]:" );
```

```
        kar=getche();
```

```
    }while(kar!='e'&& kar!='h');
```

```

clrscr();
if(kar=='e')
{
printf("R eksen takımı için [R,0]T matrisinin oluşturulması.\n\n");
for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
printf("[R,0]T [%d,%d] = ",j+1,k+1);
scanf("%f",&nhtm[0][j][k]);
}
}
else
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
if(j==k)
nhtm[0][j][k]=1;
else
nhtm[0][j][k]=0;
}
}
}

```

```

for(i=1;i<=int(n);i++)
{
nhtm[i][0][0]=cos(rad(ndh[i-1][3]));
nhtm[i][0][1]=-sin(rad(ndh[i-1][3]));
nhtm[i][0][2]=0;
nhtm[i][0][3]=ndh[i-1][0];
}

```

```

nhtm[i][1][0]=cos(rad(ndh[i-1][1]))*sin(rad(ndh[i-1][3]));
nhtm[i][1][1]=cos(rad(ndh[i-1][1]))*cos(rad(ndh[i-1][3]));
nhtm[i][1][2]=-sin(rad(ndh[i-1][1]));
nhtm[i][1][3]=-sin(rad(ndh[i-1][1]))*ndh[i-1][2];
nhtm[i][2][0]=sin(rad(ndh[i-1][1]))*sin(rad(ndh[i-1][3]));
nhtm[i][2][1]=sin(rad(ndh[i-1][1]))*cos(rad(ndh[i-1][3]));
nhtm[i][2][2]=cos(rad(ndh[i-1][1]));
nhtm[i][2][3]=cos(rad(ndh[i-1][1]))*ndh[i-1][2];
nhtm[i][3][0]=0;
nhtm[i][3][1]=0;
nhtm[i][3][2]=0;
nhtm[i][3][3]=1;
}

/* [n,E]T matrisinin ekrandan okunması. */

clrscr();
do
{
    printf("E eksen takımı için [%d,E]T matrisini girmek istiyor musunuz?
[e/h]:",n );
    kar=getche();
}while(kar!='e'&& kar!='h');

clrscr();
if(kar=='e')
{
    printf("E eksen takımı için [%d,E]T matrisinin
oluşturulması.\n\n",int(n));
    for(j=0;j<4;j++)

```

```

for(k=0;k<4;k++)
{
printf("[%d,E]T [%d,%d] = ",int(n),j+1,k+1);
scanf("%f",&nhtm[n+1][j][k]);
}
}
else
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
if(j==k)
nhtm[n+1][j][k]=1;
else
nhtm[n+1][j][k]=0;
}
}

/* nhtm matrislerinin NHTM.DAT dosyasına kaydedilmesi */

for(i=0;i<int(n)+2;i++)
{
for(j=0;j<4;j++)
for(k=0;k<4;k++)
fprintf(fnhtm,"%f\t",nhtm[i][j][k]);
}
fclose(fnhtm);

} /* HTM matrisleri oluşturuldu. */
}

```

Ek - 14**GRAFİK Program Kodu**

```
void grafik(void)
{
    FILE *fhar;
    clock_t ilk,son,arailk,arason;
    int gdriver = DETECT, gmode, errorcode;
    int xr,yr,zr,i,j,k,l,m,maxy,ku_fak;
    float ilks[10],ilky[10],ilksz[10],sonx[10],sony[10],sonz[10];
    float prsilks[10],prsilky[10],prssonx[10],prssony[10],alfa,beta,yo,xo;
    char msg[80];char kar;
    struct hareket {
        int maf_no;
        float har_mik;
        float hiz;
        int bekle;
    };
    hareket har[100];
    int adim=0;
    float t,ara,artis;
    int int_ara;
    clrscr();
    printf("Referans eksen takımını için koordinatları giriniz (x,y,z):\n");
    printf("xr=");
    scanf("%d",&xr);
    printf("yr=");
    scanf("%d",&yr);
    printf("zr=");
```

```

scanf("%d",&zr);
/* hareket deęerleri dosyadan mı okunacak ekrandan mı girilecek */
printf("\n\n\n");
printf("Hareket Deęerleri [Dosya :d / Klavye :k]");
kar=getche();
if(kar=='d')
{
    if((fhar=fopen("har.dat","r"))==NULL)
    {
        printf("har.dat dosyası açılmadı deęerleri klavyeden girmek
zorundasınız.");
    }
else
{
    do
    {
        adim=adim+1;
        fscanf(fhar,"%d",&har[adim].maf_no);
        fscanf(fhar,"%f",&har[adim].har_mik);
        fscanf(fhar,"%f",&har[adim].hiz);
        fscanf(fhar,"%d",&har[adim].bekle);
    }while(!feof(fhar));
    adim=adim-1;
    fclose(fhar);
}
}
else
{
    printf("\n");
do

```

```

{
    adim=adim+1;
    clrscr();
    do
        {
            printf("Hareketi yapacak olan mafsal :");
            scanf("%d",&har[adim].maf_no);
        }while(har[adim].maf_no < 1 || har[adim].maf_no > n);

    do
        {
            printf("Mafsalın yapacağı hareket miktarı (deg. veya br.):");
            scanf("%f",&har[adim].har_mik);
            do
                {
                    printf("Mafsalın hızı (deg/ms veya br/ms) :");
                    scanf("%f",&har[adim].hiz);
                }while(har[adim].hiz < 0);
            }while(har[adim].har_mik != 0 && har[adim].hiz == 0);
        printf("Hareketsiz bekleme süresi (ms) :");
        do{
            scanf("%d",&har[adim].bekle);
        }while(har[adim].bekle < 0);
        printf("Devam için 'd':");
        kar = getch();
    }while(kar == 'd');
    /* dosyaya kayıt */
    if((fhar=fopen("har.dat","w"))==NULL)
        {
            printf("har.dat dosyası açılmadı!!!");
        }

```

```

        getch();
        exit(0);
    }
    for(i=1;i<=adim;i++)
    {
        fprintf(fhar,"%d\t",har[i].maf_no);
        fprintf(fhar,"%f\t",har[i].har_mik);
        fprintf(fhar,"%f\t",har[i].hiz);
        fprintf(fhar,"%d\t",har[i].bekle);
    }
    fclose(fhar);
}
clrscr();
printf(" Adım    Maf.no    Har.mik.    Hiz    Bekleme\n");
printf("\n");
for(i=1;i<=adim;i++)
{
    printf("%6d",i);
    printf("%11d",har[i].maf_no);
    printf("%15.1f",har[i].har_mik);
    printf("%13.3f",har[i].hiz);
    printf("%12d",har[i].bekle);
    printf("\n");
}
getch();
alfa=30;
beta=30;
initgraph(&gdriver, &gmode, "");
errorcode = graphresult();
if (errorcode != grOk)

```

```

{
    printf("Grafik Ekrana Geçilemiyor: %s\n", grapherrormsg(errorcode));
    printf("Çıkış İçin Herhangi Bir Tuşa Basın:");
    getch();
    exit(1);
}

i1=j1=k1=4;
/* ilk konumun hesaplanması */
cizim:
maxy=getmaxy();
ilkx[0]=xr;
ilky[0]=yr;
ilkz[0]=zr;

for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
    na[j][k]=nhtm[0][j][k];
    nb[j][k]=nhtm[1][j][k];
}

ilkx[1]=na[0][3];
ilky[1]=na[1][3];
ilkz[1]=na[2][3];
nmat();
ilkx[2]=nc[0][3];
ilky[2]=nc[1][3];
ilkz[2]=nc[2][3];
for(i=2;i<int(n)+2;i++)
{

```

```

for(j=0;j<4;j++)
for(k=0;k<4;k++)
{
na[j][k]=nc[j][k];
nb[j][k]=nhtm[i][j][k];
}
nmat();
ilkx[i+1]=nc[0][3];
ilky[i+1]=nc[1][3];
ilkz[i+1]=nc[2][3];
}
/* ilk konumun çizimi için fonks. argumanlarının oluşturulması */
xo=120;
yo=200;
prsilx[0]=xo+ilkx[0]*cos(rad(alfa))/3+ilky[0]*cos(rad(beta))/3;
prsilky[0]=yo+ilkx[0]*sin(rad(alfa))/3-ilky[0]*sin(rad(beta))/3-ilkz[0]/3;
for(i=1;i<=int(n)+2;i++)
{
prsilx[i]=prsilx[0]+ilkx[i]*cos(rad(alfa))+ilky[i]*cos(rad(beta));
prsilky[i]=maxy-prsilky[0]+ilkx[i]*sin(rad(alfa))-ilky[i]*sin(rad(beta))-
ilkz[i];
}
clrscr();
ilk=clock();
for(i=1;i<=adim;i++)
{
t=fabs(har[i].har_mik / har[i].hiz);
ara=t/200;
int_ara=int(ara);
artis=har[i].har_mik / (int_ara+1);

```

```

for(l=1;l<=int_ara+1;l++)
{
    arailk=clock();
    /* mafsal tipinin belirlenmesi */
    if(tip[har[i].maf_no-1][0]=='r')
        ndh[har[i].maf_no-1][3]=ndh[har[i].maf_no-1][3]+artis;
    else
        ndh[har[i].maf_no-1][2]=ndh[har[i].maf_no-1][2]+artis;
    /* yeni htm[har[i].maf_no]'nun hesaplanması */
    nhtm[har[i].maf_no][0][0]=cos(rad(ndh[har[i].maf_no-1][3]));
    nhtm[har[i].maf_no][0][1]=-sin(rad(ndh[har[i].maf_no-1][3]));
    nhtm[har[i].maf_no][0][2]=0;
    nhtm[har[i].maf_no][0][3]=ndh[har[i].maf_no-1][0];
        nhtm[har[i].maf_no][1][0]=cos(rad(ndh[har[i].maf_no-
1][1]))*sin(rad(ndh[har[i].maf_no-1][3]));
        nhtm[har[i].maf_no][1][1]=cos(rad(ndh[har[i].maf_no-
1][1]))*cos(rad(ndh[har[i].maf_no-1][3]));
        nhtm[har[i].maf_no][1][2]=-sin(rad(ndh[har[i].maf_no-1][1]));
        nhtm[har[i].maf_no][1][3]=-sin(rad(ndh[har[i].maf_no-
1][1]))*ndh[har[i].maf_no-1][2];
        nhtm[har[i].maf_no][2][0]=sin(rad(ndh[har[i].maf_no-
1][1]))*sin(rad(ndh[har[i].maf_no-1][3]));
        nhtm[har[i].maf_no][2][1]=sin(rad(ndh[har[i].maf_no-
1][1]))*cos(rad(ndh[har[i].maf_no-1][3]));
        nhtm[har[i].maf_no][2][2]=cos(rad(ndh[har[i].maf_no-1][1]));
        nhtm[har[i].maf_no][2][3]=cos(rad(ndh[har[i].maf_no-
1][1]))*ndh[har[i].maf_no-1][2];
    /* son konumun hesaplanması */
    for(j=0;j<4;j++)
        for(k=0;k<4;k++)

```

```

{
  na[j][k]=nhtm[0][j][k];
  nb[j][k]=nhtm[1][j][k];
}
sonx[1]=na[0][3];
sony[1]=na[1][3];
sonz[1]=na[2][3];
nmat();
sonx[2]=nc[0][3];
sony[2]=nc[1][3];
sonz[2]=nc[2][3];
for(m=2;m<int(n)+2;m++)

```

```

{
  for(j=0;j<4;j++)
  for(k=0;k<4;k++)
  {
    na[j][k]=nc[j][k];
    nb[j][k]=nhtm[m][j][k];
  }
  nmat();
  sonx[m+1]=nc[0][3];
  sony[m+1]=nc[1][3];
  sonz[m+1]=nc[2][3];
}

```

/* son monumun çizilmesi için fonks. argumanlarının oluşturulması

*/

```

prssonx[0]=prsilx[0];
prssony[0]=prsilky[0];
for(m=1;m<=int(n)+2;m++)
{

```

```

prssonx[m]=prsilky[0]+sonx[m]*cos(rad(alfa))+sony[m]*cos(rad(beta));
prssony[m]=maxy-prsilky[0]+sonx[m]*sin(rad(alfa))-
sony[m]*sin(rad(beta))-sonz[m];
    }
    /* konumların çizilmesi */
    cleardevice();
    /* ilk */
    setlinestyle(3,1,1);
    konumciz(maxy,prsilky,prsilky);
    /* son */
    setlinestyle(0,1,1);
    konumciz(maxy,prssonx,prssony);
    arason=clock();
    delay(200-(arason-arailk)/CLK_TCK*1000);
}/* i */
/* renk çizgilerinin çizilmesi */
renkciz(int(n));
/* eksen takımının çizilmesi */
setcolor(15);
line(250,75,250,25);
outtextxy(250,5,"z");
line(250,75,250+75,75-75*tan(rad(beta)));
outtextxy(250+85,75-70*tan(rad(beta)),"y");
line(250,75,250+45,75+45*tan(rad(alfa)));
outtextxy(250+55,75+45*tan(rad(alfa)),"x");
/* tablonun oluşturulması */
setcolor(15);

```

```

tablo(ilkx[n+1],ilky[n+1],ilkz[n+1],sonx[n+2],sony[n+2],sonz[n+2]);
son=clock();
getch();
restorecrtmode();
printf("Devam etmek için 'd'ye basın :");
kar=getche();
if(kar=='d')
{
    clrscr();
    adim=0;
    /* hareket deęerleri dosyadan mı okunacak ekrandan mı girilecek */
    printf("\n\n\n");
    printf("Hareket Deęerleri [Dosya :d / Klavye :k]");
    kar=getche();
    if(kar=='d')
    {
        if((fhar=fopen("har.dat","r"))==NULL)
        {
            printf("har.dat dosyası açılmadı deęerleri klavyeden girmek
zorundasınız.");
        }
        else
        {
            do
            {
                adim=adim+1;
                fscanf(fhar,"%d",&har[adim].maf_no);
                fscanf(fhar,"%f",&har[adim].har_mik);
                fscanf(fhar,"%f",&har[adim].hiz);
                fscanf(fhar,"%d",&har[adim].bekle);
            }
            while(kar=='d');
        }
    }
}

```

```

}while(!feof(fhar));
    adim=adim-1;
    fclose(fhar);
}
}
else
{
printf("\n");
do
{
    adim=adim+1;
    clrscr();
do
{
    printf("Hareketi yapacak olan mafsal :");
    scanf("%d",&har[adim].maf_no);
}while(har[adim].maf_no < 1 || har[adim].maf_no > n);
do
{
    printf("Mafsalın yapacağı hareket miktarı (deg. veya br.):");
    scanf("%f",&har[adim].har_mik);
do
{
        printf("Mafsalın hızı (deg/ms veya br/ms) :");
        scanf("%f",&har[adim].hiz);
}while(har[adim].hiz < 0);
}while(har[adim].har_mik != 0 && har[adim].hiz == 0);
printf("Hareketsiz bekleme süresi (ms) :");
do
{

```

```

        scanf("%d",&har[adim].bekle);
    }while(har[adim].bekle < 0);
    printf("Devam için 'd':");
    kar = getch();
}while(kar == 'd');
/* dosyaya kayıt */
if((fhar=fopen("har.dat","w"))==NULL)
{
    printf("har.dat dosyası açılmadı!!!");
    getch();
    exit(0);
}
for(i=1;i<=adim;i++)
{
    fprintf(fhar,"%d\t",har[i].maf_no);
    fprintf(fhar,"%f\t",har[i].har_mik);
    fprintf(fhar,"%f\t",har[i].hiz);
    fprintf(fhar,"%d\t",har[i].bekle);
}
fclose(fhar);
}
clrscr();
printf(" Adım      Maf.no      Har.mik.      Hız      Bekleme\n");
printf("\n");
for(i=1;i<=adim;i++)
{
    printf("%6d",i);
    printf("%11d",har[i].maf_no);
    printf("%15.1f",har[i].har_mik);
    printf("%13.3f",har[i].hiz);

```

```
        printf("%12d",har[i].bekle);  
        printf("\n");  
    }  
    getch();  
    setgraphmode(getgraphmode());  
    goto cizim;  
}  
closegraph();  
}
```



Ek - 15**MAIN () Program Kodu**

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include<dos.h>
char tip[10][2];
float ndh[10][4];
float nhtm[12][4][4];
float na[4][4];
float nb[4][4];
float nc[4][4];
float n;
int i1,j1,k1;
#include"tablo.h"
#include"rad.h"
#include"nmat.h"
#include"npmek.h"
#include"renkciz.h"
#include"konumciz.h"
#include"ndhtablo.h"
#include"nhtmmat.h"
#include"grafik.h"
int main (void)
{
```

```
int i,j,k;  
ndhtablo();  
nhtmamat();  
grafik();  
return 0;  
}
```



ÖZGEÇMİŞ

Doğum Tarihi : 20 Ağustos 1967

Doğum Yeri : Sakarya

Eğitim : 1974 - 1978 Hasanpaşa İlkokulu

1978 - 1981 Yakacık Ortaokulu

1981 - 1984 Kartal Endüstri Meslek Lisesi - Makina Ressamlığı

1984 - 1986 D.E.Ü. İzmir Mes. Yük. Okulu - Makina-Resim-Konst.

1988 - 1993 Y.T.Ü. Makina Fakültesi - Makina Mühendisliği

1984 ve 1993 arası yıllarda muhtelif firmalarda makina ressamı olarak çalışmıştır. Halen İGDAŞ ' ta makina mühendisi olarak çalışmaktadır.

