

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ALTI EKSENLİ ROBOT KOLUN HAREKETSSEL
KARAKTERİSTLİĞİNİN GÖRSEL PROGRAMLANMASI VE
GERÇEK ZAMANLI UYGULAMALAR**

Mehmet Serdar GÜZEL

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2008**

Her Hakkı Saklıdır

ÖZET

Yüksek Lisans Tezi

ALTI EKSENLİ ROBOT KOLUN HAREKETSEL KARAKTERİSTİĞİNİN GÖRSEL PROGRAMLANMASI VE GERÇEK ZAMANLI UYGULAMALAR

Mehmet Serdar GÜZEL

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı

Danışman: Prof. Dr. Baki KOYUNCU

Bu Tez çalışması kapsamında Beş ve Altı Eksenli Robot kolların matematiksel analizinin yapılması, görsel tabanlı yazılımlar geliştirerek; Robot kolların yönetilmesi ve gerçek zamanlı akıllı uygulamalarda kullanılması hedeflenmiştir. Uygulama ve Teorik kısımları için robot kol modeli olarak Lynx şirketinin Beş Serbestlik Derecesine sahip (5 DOF) Altı eklemlili robot kolu kullanıldı. Matematik analiz kapsamında robot kolun hareket karakteristiği incelenmiştir. Bu analiz, Robot kolların hareket karakteristiğinin Matematiksel Modellenesi; Ters ve Düz kinematik çözümlerin hesaplama yöntem ve tekniklerini içerir. Ayrıca özel olarak Lynx-6 robot kollar için esnek ve programlanmasi kolay kinematik çözümler Denavit-Hartenberg (D-H) dönüşüm yöntemi kullanılarak bu çalışma kapsamında geliştirilmiştir. Teorik Çalışma kapsamında kinematik analizin yanında robot eklemlerini kontrol eden Doğru Akım (DC) servo motorların genel dinamik modelleri ve kontrol tipleri üzerine de çalışılmıştır. Uygulama kısmında ise dört farklı yazılım geliştirildi. Bu yazılımların ilk iki tanesinde Lynx robot kolunu fiziksel olarak kontrol edilip sırasıyla gerçek zamanlı ve akıllı uygulamalar için kullanılmıştır. Bunun ile beraber geliştirilen üçüncü yazılım Lynx robot kollar için kinematik çözüm benzetim yazılımıdır. Son geliştirilen yazılım ise Doğru Akım servo motorlarının ve Kontrol Kartının işleyişini ve hareket karakteristiği ile ilişkisini taklit eden bir benzetim uygulamasıdır.

Mayıs 2008, 116 sayfa

Anahtar Kelimeler: Kinematik, Robot Kol, Manipülatör, Özgürlük Derecesi, Ters Kinematik, Düz Kinematik, Yapay Zekâ, İmge işleme, Servo Motor, Kontrol Kartı, Doğru Akım Motoru,

ABSTRACT

Master Thesis

VİSUAL PROGRAMMING OF SIX AXES ROBOT ARM MOTIONAL CHARACTERİSTİCS AND REAL TIME APPLICATIONS

Mehmet Serdar GÜZEL

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. Baki KOYUNCU

The scope of this thesis study directly aims to make mathematical analysis of Five and Six axes Robot Arms and to develop visual based software packages for managing robot arm and using this robot arm for different applications such as Real Time applications and Intelligent applications. Lynx-6, product of Lynx Company, is used in the Theoretical and Application parts of this study as a robot arm model .Within Mathematical analysis , Forward and Inverse kinematics of robot arm is analyzed and limitations of robot's joints are calculated. Moreover a unique kinematics solution, easy to implement, is developed for Lynx-6 robot arms with using Denavit-Hartenberg (D-H) transformation methods. In addition to Kinematics analysis, Dynamic modeling of Direct Current (DC) Motors and their control techniques are also studied in the scope of theoretical study. In the application part four different software packages are developed. First and second software packages are developed for physical controlling of Lynx-6. First one is used for Real Time applications whereas second one is for intelligent applications. Third developed software package is for kinematics simulation package of Lynx arms, and the last one is for Dynamic modeling of DC motors were introduced. Their combined motion within the arm kinematics and control card were analyzed and displayed visually.

May 2008, 116 pages

Key Words: Kinematics, Robot Arm, Manipulator, DOF, Inverse Kinematics, Forward Kinematics , Artifical Intelligence , Image Processing , Servo Engine , Control Card , DC Motor

TEŞEKKÜR

Çalışmalarımı yönlendiren, araştırmalarımın her aşamasında bilgi, öneri ve yardımlarını esirgemeyerek akademik ortamda olduğu kadar beşeri ilişkilerde de engin fikirleriyle yetişme ve gelişmeme katkıda bulunan danışman hocam sayın Prof. Dr. Baki KOYUNCU'YA ve çalışmalarım süresince birçok fedakârlıklar göstererek beni destekleyen aileme en derin duygularla teşekkür ederim.

Mehmet Serdar GÜZEL

Ankara, Mayıs 2008

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ VE TEŞEKKÜR.....	iii
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ.....	ix
1. GİRİŞ.....	1
2. ROBOT TEORİ BİLGİSİ.....	4
2.1 Robotiğe Giriş.....	4
2.1.1 Robot tipleri ve sınıflandırılması.....	5
2.2 Robotla İlişkili Matematik.....	9
2.2.1 Kordinata sistemleri.....	9
2.2.2 Matris matematiği.....	13
3. PROJEDE KULLANILACAK ROBOT KOLUN SEÇİMİ VE YAPISI.....	15
3.1 Robot Kol Seçimi.....	15
3.2 Lynx Altı Eksenli Robot Kol.....	16
3.2.1 Robot kolun geometri ve limiti.....	16
3.2.2 Robot kol için servo motorlar.....	18
3.2.3 Robot kol için servo kontrol kartı.....	19
4. KİNEMATİK ANALİZ VE POZİSYON BELİRLEME.....	21
4.1 Dönme Matrisi.....	21
4.1.1 Euler açıları ve gösterimi.....	23
4.1.2 Roll/Pitch/Yaw açıları ve gösterilimi.....	24
4.2 Öteleme Vektörü.....	25
4.3 Homojen Dönüşümler.....	25
4.4 Düz Kinematik Analiz.....	26
4.4.1 Denavit-Hartenberg(Dh) dönüşümü.....	28
4.4.1.1 Altı eksenli robot kolun düz kinematik analizi.....	30
4.4.1.2 Lynx-6 robot kolun düz kinematik analizi.....	32
4.5 Ters Kinematik Analiz.....	35
4.5.1 Üç eklemlili düzlemsel robot kolun ters kinematik çözümü.....	37
4.5.2 Altı eklemlili düzlemsel robot kolun ters kinematik çözümü.....	38
4.5.3 Lynx-6 düzlemsel robot kolun ters kinematik çözümü.....	43
5. DOĞRU AKIM MOTORLARI.....	47
5.1 Doğru Akım Servo Motorlara Giriş.....	48
5.1.1 Doğru akım motorları temel denklemleri.....	49
5.1.1.1 Elektrik denklemleri.....	50
5.1.1.2 Mekanik denklemleri.....	51
5.2 Dinamik Model ve Kontrol Tipleri.....	52
6. UYGULAMALAR.....	55
6.1 Yame(Gerçek Zamanlı Uygulama).....	55
6.1.1 Yazılım uygulama ve içeriği.....	56
6.1.1.1 Yazılımın yapısı ve sistematigi.....	60
6.2 Üç Taş Oynayan Türk Robotu (Dinamik Uygulama).....	66
6.2.1 Yazılım uygulama ve içeriği.....	67
6.2.2 Min –Max algoritması.....	72
6.2.3 İmge çıkarma ve bilgisayar algılaması.....	76

6.3 Kinematik Simülasyon Yazılımı	78
6.3.1 Yazılım ve uygulama içeriği	78
6.4 Lyds (Dinamik Ve Kinematik Simülasyon Yazılımı)	84
6.4.1 Yazılım ve uygulama içeriği	85
6.5 Uygulama Sonuç Kısmı	90
7 TEST VE UYGULAMA SONUCU	92
7.1 Yame	92
7.2 Üç Taş Oynayan Robot	95
7.3 Kinematik Simülasyon Yazılımı	100
7.4 Lyds (Dinamik ve Kinematik Simülasyon Yazılımı)	102
8 SONUÇ KISMI	106
8.1 Tezin Amacı	106
8.2 Tezin Sonucu	106
8.3 Bölümlerin Amacı ve Özeti	110
8.4 Gelecekteki Çalışmalar İçin Tavsiyeler	111
KAYNAKLAR	113
EK 1 C# Kodları	115
ÖZGEÇMİŞ	116

ŞEKİLLER DİZİNİ

Şekil 2.1	Dört derece özgürlüğe sahip robot örneği.....	6
Şekil 2.2	Üç derece özgürlüğe sahip robot modeli	10
Şekil 2.3	P noktasının a noktasına göre tanımı	11
Şekil 2.4	P noktasının {a} ve {c} koordinat sistemlerine göre tanımı.....	11
Şekil 2.5	A noktası ve uç noktanın koordinatları sistemleri	12
Şekil 2.6	B koordinat sisteminin a sistemine göre yönelmesi.....	13
Şekil 2.7	Yönelme ve pozisyon matrisi	13
Şekil 2.8	Üçüncü eklemin yönelme ve pozisyon matrisi	14
Şekil 3.1	Lynxarm robot kol	15
Şekil 3.2	Altı eksenli lynxarm robot kolun 3 boyutlu modellenmesi	16
Şekil 3.3	Lynx-6 robot kolun yerel koordinat sistemi.....	17
Şekil 3.4	Lynx-6 robot kolun eklem uzaklık değerleri	17
Şekil 3.5	Grip eklemine hareketini sağlayan hs – 81 mikro servo motoru.....	19
Şekil 3.6	Ssc-32 Servo Kontrol Kartı.....	19
Şekil 4.1	Z0 eksenini etrafında dönme	22
Şekil 4.2	Dönme matrisi.....	22
Şekil 4.3	Euler açıları gösterimi.....	23
Şekil 4.4	Euler dönüşüm matrisi	23
Şekil 4.5	Roll / pitch /yaw açıları gösterilimi	24
Şekil 4.6	Roll / pitch /yaw açıları öteleme matrisi	24
Şekil 4.7	Ötelenmiş çerçeve	25
Şekil 4.8	Öteleme vektörü.....	25
Şekil 4.9	Homojen dönüşüm matrisi.....	26
Şekil 4.10	Temel homojen dönüşüm matrisleri	26
Şekil 4.11	Eklem değişken tipleri ve parametreler	27
Şekil 4.12	N+1 uzva sahip kinematik zincir oluşturan manipülatör	27
Şekil 4.13	Denavit-hartenberg çerçeve atanması	29
Şekil 4.14	Altı eksenli robot kolun yapısı.....	30
Şekil 4.15	Altı eksenli robot kol için dönüşüm matrisleri	31
Şekil 4.16	Altı eksenli robot kol için düz kinematik çözüm.....	32
Şekil 4.17	Lynx-6 robot kol için dönüşüm matrisleri	33
Şekil 4.18	Lynx-6 robot kol için düz kinematik çözümler	37
Şekil 4.19	Üç eklemlili düzlemsel manipülatör	37
Şekil 4.20	Geometrik yaklaşımla θ_2 'nin çıkarılması.....	37
Şekil 4.21	Geometrik yaklaşımla θ_1 ve θ_3 açılarının çıkarılması.....	38
Şekil 4.22	Kinematik çözüm ve sembol matrisi.....	38
Şekil 4.23	Altı eksenli için ters kinematik ilk adım	39
Şekil 4.24	θ_1 açısının elde etme işlemi	39
Şekil 4.25	$\cos(\theta_3)$ açısının elde etme adımları	39
Şekil 4.26	θ_3 açısının elde etme adımları	40
Şekil 4.27	θ_{234} açısının çıkarılma işlemlerinin ilk aşaması	40
Şekil 4.28	θ_{234} açısının çıkarılma işlemlerinin	41
Şekil 4.29	$\sin(\theta_2)$ ve $\cos(\theta_2)$ çıkarılma işlemleri	41
Şekil 4.30	θ_5 açısının elde edilmesi.....	41
Şekil 4.31	θ_6 açısının elde edilmesi.....	42
Şekil 4.32	Altı eksenli robot kolun ters kinematik çözümleri.....	43
Şekil 4.33	Kinematik çözüm ve sembol matrisi.....	44

Şekil 4.34	Kinematik çözüm ve sembol matris eşitliği	44
Şekil 5.1	Doğru akım motor eşdeğer devresi	47
Şekil 5.2	Hs-81 servo motoru ve modeli	48
Şekil 5.3	Doğru akım elektiriksel eşitlik denklem şeması	50
Şekil 5.4	Doğru akım motoru blok diyagramı	52
Şekil 5.5	Doğru akım motoru ve armatör akım kontrolü	53
Şekil 6.1	Gerçek zamanlı hareket ekranı.....	56
Şekil 6.2	Ters kinematik ekranı	57
Şekil 6.3	Satranç kontrol ekranı	58
Şekil 6.4	Satranç Tahtası Gerçek Zamanlı Operasyonlar Ekranı.....	59
Şekil 6.5	Satranç tahtasında durumlar.....	59
Şekil 6.6	Tahta üzerinde hareket işlemleri.....	60
Şekil 6.7	Yazılımın sistematığı ve yapısı.....	60
Şekil 6.8	Tüm eksenler için (x;y;z) dönüş fonksiyonu	61
Şekil 6.9	Çizim için veri yapısı ve dizi nesnesi	62
Şekil 6.10	Simülator kod parçaçığı	63
Şekil 6.11	Seri giriş nesnesi ve tanımları.....	64
Şekil 6.12	Cconverter fonksiyon içeriği.....	65
Şekil 6.13	Taban açısının hesaplanması	66
Şekil 6.14	Üç taş oyunu çalışma alanı ve robot kolun durumu.....	67
Şekil 6.15	Üç taş oyununun yazılım paket ara yüz ekranı	68
Şekil 6.16	Sistemin ana mimarisi.....	69
Şekil 6.17	Eklenti yapısı akış diyagramı.....	71
Şekil 6.18	Üç taş oyunu için arama ağacı	73
Şekil 6.19	Üç taş oyunu için örnek bir pozisyon	74
Şekil 6.20	Üç taş oyunu için örnek arama ağacı	75
Şekil 6.21	Minmax en iyi hareket algoriması	75
Şekil 6.22	İmge çıkarma işlemi.....	77
Şekil 6.23	Threshold değeri için limit grafiği	78
Şekil 6.24	Akış şeması diyagramı.....	80
Şekil 6.25	Ana ekran (düz kinematik ekranı)	81
Şekil 6.26	Düz kinematik fonksiyonu.....	82
Şekil 6.27	Ters kinematik ekranı	83
Şekil 6.28	Ters kinematik fonksiyonları	84
Şekil 6.29	Lyds sistem akış diyagramı.....	86
Şekil 6.30	Lyds ana veri girişi ekranı.....	87
Şekil 6.31	Doğru akım motoru simülasyon ekran kodu.....	89
Şekil 6.32	Doğru akım motoru simülasyon ekranı	90
Şekil 6.33	Doğru akım motoru veri grafikleri.....	91
Şekil 7.1	Satranç tahtası üzerinde ki hareket	92
Şekil 7.2	Sistemin üç temel hareketi	93
Şekil 7.3	Hs-475 ve hsr-5995 servo motorlarının karşılaştırılması	95
Şekil 7.4	Üç taş oynayan robot kol	95
Şekil 7.5	Yapay zeka ve insan oyun sonuçları	96
Şekil 7.6	Oyun sistemi ve görme mekanizması	98
Şekil 7.7	Görüntü veritabanı	98
Şekil 7.8	Ortam ışığı az olduğu zaman elde edilen grafik	99
Şekil 7.9	Ortam ışığı çok yoğun olduğu zaman elde edilen grafik	99

Şekil 7.10 Ortam ışığının normalolduğu zaman elde edilen grafik	100
Şekil 7.11 Düz kinematik çözüm ekranı	101
Şekil 7.12 Ters kinematik çözüm ekranı.....	102
Şekil 7.13 Taban eklemi açısı - zaman grafiği (23 derece)	104
Şekil 7.14 Taban eklemi açısal hız - zaman grafiği (23 derece)	105
Şekil 7.15 Hareket matrisleri ve sonuç matrisleri	105

ÇİZELGELER DİZİNİ

Çizelge 2.1	Kinematik zincir tablosu	6
Çizelge 2.2	Ana robot tipleri ve çalışma uzayları	7
Çizelge 2.3	Robot kollar konfigürasyon karşılaştırılması.....	9
Çizelge 3.1	Eklemlerin limitleri	18
Çizelge 3.2	Ssc-32 için komut format tablosu.....	20
Çizelge 4.1	Altı eksenli robot kolun d-h parametre tablosu.....	31
Çizelge 4.2	Lynx-6 robot kolun d-h parametre tablosu	33
Çizelge 4.3	Lynx-6 Robot Kol için Dönüşüm Matrisleri.....	34
Çizelge 6.1	Oyun ağacı karmaşıklıkları	76
Çizelge 7.1	Taş taşıma modunda	93
Çizelge 7.2	Taş bırakıp / alma modunda.....	93
Çizelge 7.3	Sapma çizelgesi omuz için.....	94
Çizelge 7.4	Yapay zeka motoru için test (2 denekle).....	97
Çizelge 7.5	Simülatör örneği için d-h parametre tablosu.....	101
Çizelge 7.6	Ölçülen değerlerin karşılaştırılması	103
Çizelge 7.7	Örnek için motor sabit değerleri	104

1. GİRİŞ

Savunma ve Endüstriyel sanayide yaşanan gelişmeler, insan gücü maliyetlerinin düşürülmesi ve kalite arttırılmasının amaçlanması, insanları robot teknolojilerini kullanmaya zorunlu bırakmıştır. Robotlar günümüz teknolojisinin en güçlü elemanlarıdır. Robot sistemlerin çok geniş kullanım alanları olduğundan dolayı tanımlanmalarında bazı ufak ayrıntılar önem kazanmıştır. Robot sistemlerin en yaygın kullanım alanlarından biri de endüstri sahalarıdır. İngiliz Robot Birliği (BRA) ve Japon Endüstriyel Robot Birliği (JIRA) gibi ulusal robotik birliklerinin yayınladığı resmi robot tanımları, çoğunlukla endüstriyel robotların tanımıdır. Robot sistemlerin endüstri alanlarında kullanılmalarıyla ilgili olarak robot tanımı, Amerikan Robot Enstitüsü (RIA) tarafından şu şekilde yapılmıştır. "Robot, çeşitli görevleri yapmak maksadıyla değişik şekillerde programlanmış hareketlerle; nesnelerin, gereçlerin, gereçlerin ya da özel düzeneklerin taşınması için tasarlanmış çok işlevli bir manipülatördür (Candemir 2007, Craig 2005). Robot sistemlerin kontrollerinde ya da programlarında meydana gelebilecek yanlışlıklar, ilgili alt sistemlerin çalışmasında büyük hatalara sebep olabilir. Örneğin, robot kol sistemine ait pnömatik piston tipindeki bir sürücü hassas kontrol edilememesinden dolayı, meydana gelen hata neticesinde hem çevresine zarar verebilir, hem de kullanılmaz duruma gelebilir. Bu nedenle, robot sistemler içerisinde kullanılan bütün sürücüler ve alt sistemlerin en hassas biçimde kontrolleri sağlanmalıdır. Zamanla sistemdeki mekanik aşınmalar ve sürücü hatalarından meydana gelecek aksaklıklar, robot sistemin pozisyonunu kontrol eden denetleyicilerle, konum hatası sınır toleransları içerisinde tutulmalıdır.

Robot teknolojilerinin en popüler alt sınıflarından biride robot kollarıdır. Günümüzde hemen hemen her fabrikanın üretim bandında, robot kollar insanların yerini almıştır ve almaya da devam edecektir. İşte bu çalışmaların ve analizlerin ışığında bu tez çalışmasında Lynx robot kolları teorik olarak incelenmiş kinematik çözümleri hesaplanmıştır. Teorik çalışma grafik ara yüzü destekleyen yazılım ortamlarında uygulama olarak geliştirilip teorik bilgiler uygulama ortamında test edilmiştir. Lynx şirketinin Beş Serbestlik Derecesine sahip (5 DOF) Altı eklemlili robot kolu teorik ve uygulama kısımlarında model olarak kullanılmıştır. Bu robot kol bu tip bir tez çalışması

iin en uygun robot kol olarak belirlenmiřtir. Gerek hızı gerekse hassasiyeti, uygulamalar iin yeterli verimi saėlamaktadır.

Bu tez kapsamında teorik alıřma kısmında Beř ve Altı Eksenli Robot kollar hakkında detaylı bir matematiksel alıřma gerekleřtirilmiřtir. Bu alıřma kapsamında Beř ve Altı Eksenli Robot Kolların matematiksel modellenmesi, hareketsel karakteristiėin incelenmesi dz ve ters kinematik denklemlerinin elde edilip bu denklemlerin özlmesi saėlanmıřtır. Ayrıca Beř Serbestlik Derecesine sahip (5 DOF) Lynx-6 Robot kolu iin esnek ve gerek uygulamalarda kullanımı kolay dz ve ters kinematik özm bu alıřma kapsamında geliřtirilmiřtir. Robot kolların pozisyon analizi iin deėiřik konfigrasyonları destekleyen yntemler vardır, bu alıřmada geliřtirilen kinematik özmler iin Denavit-Hartenberg (D-H) dnřm yntemi kullanılmıř ve geliřtirilen kinematik özm algoritmalarında bu yntem temel model olarak alınmıřtır (Denavit and Hartenberg 1955).

Teorik alıřma kapsamında ayrıca, her bir eklemin hareketini kontrol eden Robot Doėru Akım (DC) servo motorların genel dinamik modelleri ve kontrol tipleri zerine de alıřılmıřtır. Dinamik modelleme kapsamında motorların elektriksel ve mekaniksel denklemleri ıkarılmıř sistemin blok diyagramları zerine alıřılmıřtır. Genel olarak Doėru Akım (DC) motorlarını kontrol etmek iin  eřit kontrol yntemi tanıtılmıř ve uygulama kısmı iin Armatr–Akım (Armature-Current) tekniėi kullanılmıřtır (Zaccarian 2005).

Uygulama kısmında ise Visula Studio.Net platformunda C# programlama dili kullanılarak drt farklı grsel tabanlı yazılım programı gerekleřtirilmiřtir. Bu alıřmalardan birinci ve ikincisinde Lynx-6 Robot kol fiziksel olarak kontrol edilip; gerek zamanlı ve akıllı uygulamalarda kullanılmıřtır. Birinci uygulama yazılımı, Robot kolun servo motorlarını ve uygulamalarda kullanılacak elektro mıknatısı servo kontrol kartı zerinden ynetirken, Kinematik Analizin gerek zamanlı uygulamada kullanılmasını saėlamaktadır. İkinci uygulama yazılımı, Lynx-6 robot kolun Kinematik Analizi ile Yapay Zek (Artificial Intelligence) ve İmge İřleme (Image Processing) teknikleri birleřtirilerek, gerek zamanlı kamera destekli akıllı bir uygulamada kullanılması saėlanmıřtır. Bu yazılım robot kol ile insanın  Tař (Tic-Tac-Toe) oynamasını saėlamaktadır. nc ve drdnc yazılımlar ise Lynx

Manip lat rlerini model alan benzetim yazılımlarıdır.     nc  yazılım iki,    , d rt ve be  eksenli Lynx robot kolların d z ve ters kinematik   z mlerini analiz edilmi , hareket matrisleri modellenmi tir. Geli tirilen di er sim lasyon yazılımında ise Lynx-6 Robot kolun her bir eklemini kontrol eden Do ru Akım servo motorların dinamik modeli sim le edilmi  ayrıca; Dinamik verilerin, hareketse  karakteristi e  evrilmesi sayısal veriler grafikler ve sim lat r yardımıyla g sterilmi tir.

2. ROBOT TEORİ BİLGİSİ

Bu bölüm kapsamında robot ile ilgili temel teori bilgisi ve ileriki kısımlarda kullanılacak robot giriş bilgisi anlatılmaktadır. Ayrıca değişik tipteki robotlar ve onların çalışma mantıklarından bahsedilmiştir. Robot için kullanılan matematik bilgisinden de bahsedilmiştir. Bu kısım, okuyanın ileriki bölümleri anlamalarına yardımcı olmak için konuyla ilişkilendirilmiştir.

2.1 Robotiğe Giriş

Robotik, Makine Mühendisliği, Elektrik ve Elektronik Mühendisliği ve Bilgisayar Mühendisliği disiplinlerinin ortak çalışması alanıdır. Robotlar bir yazılım aracılığı ile yönetilen ve yararlı bir amaç için iş ve değer üreten karmaşık makinelerdir. Robot fikri insanoğlu için binlerce yıldır varılmakla birlikte asıl atılımını 1900'li yıllarda yapmıştır. "Robot" terimi, ilk olarak Çek yazar Karel Capek tarafından 1920 yılında kaleme alınan ve "R.U.R" veya "Rossum's Universal Robots" (Rossum'un evrensel robotları) isimli tiyatro oyununda kullanılmıştır (Özsoy 2002)

Robot, programlanarak, taşıma veya yer ve yön değiştirme işlemlerini gerçekleştirebilen bir sistemdir. Taşıma görevi, gövde boyutlarına kıyasla uzak bir mesafeye hareket olarak tanımlanır. Manipülasyon ise, gövdenin pozisyonunda meydana gelen değişikliktir. Manipülatör, bir küme eklemin birbirine bağlandığı bir küme rijit kol içerir. Eklemlere motorlar takılmıştır. Böylece mekanizma verilen bir görevi gerçekleştirmek için kontrol edilebilir. Robotun analizi; matematik, mekanik ve elektronik gibi pek çok bilim dalına ait bilgi gerektirmektedir. Robotlar, endüstride, tıpta, haberleşmede ve daha birçok alanda kullanılmaktadır. Ayrıca, askeri uygulamalarda da robot kullanımı yaygındır. Robotları, kullanılan kontrol tekniğine ve içerdikleri eklem türlerine göre iki farklı şekilde sınıflandırabiliriz. Kontrol tekniğine göre robotlar: Adaptif olmayan robotlar, adaptif robotlar ve akıllı robotlardır. Akıllı robotlarda, adaptif robotlarda bulunan sensor donanımına ek olarak geniş bir bellek ve çevrenin ayrıntılı bir modeli bulunmaktadır. Robotlar da; döner, prizmatik, silindirik, küresel, düzlemsel veya helisel eklemlerden biri kullanılır ve robot, bu eklem türüne göre de sınıflandırılır. Döner ve prizmatik eklemler robotikte en çok kullanılan eklem türleridir. Bir robot, mekanik bölümler, hareketlendiriciler ve kontrol birimlerinden

oluşmaktadır. Robotun mekanik bölümleri; yapısal parçalar, güç ileten parçalar, (rot,dişliler vs.), taşıyıcılar ve akuplaj bölümleri olarak sayılabilir. Mekanik bölümler, temel yapı parçaları ve servis parçaları olarak ikiye ayrılabilir. El veya diğer gereçler (kaynak, boyama, öğütme, sıkıştırma araçları) mekaniğin servis parçalarını oluşturur. Modern robotların hareketlendiricileri, çoğunlukla elektrikseldir (DC sürücüler). Endüstride kullanılan robotlar olarak bahsettiğimiz robot kollar, bugün birçok uygulama için yeterli hız, kararlılık ve kolaylığa erişmiş durumdadır. Su an bu robot kollar için en zayıf parça, el: uç noktada bulunan el sıklıkla pnömatik, 2-pozisyonlu pense seklindedir. Bu yapı, uygulamalarda farklı parçaları tutmaya ve yönlendirmeye fazla izin vermemektedir. Bu konudaki çalışmalar gelecekteki önemli bir uğraş alanı olarak görünmektedir. Bir robot Temel olarak beş kısımdan oluşur. Aşağıda verilen kısımlar robotu oluşturan temel unsurlardır, robotun işlevini yerine getirmesi ve sınıflandırması bu beş aşamaya göre yapılır.

1: Mekanik Kısım: Robotun İskelet Kısımını oluşturur

2: Tutaç: (End Effector) :Bu eleman gerçek işi yapan kısımdır, robotun en uç Noktasıdır ve uygulamada aktif olarak yer alır

3: Motorlar: Eklemleri ve tutacı hareket ettirmek için kullanılır, en çok

Kullanılanları servo ve hidrolik motorlardır

4: Kontroller: Girişi işleyip robotun yapması gereken görevini gerçekleştirir.

5: Sensörler: Kontrollere bağlıdır, robotun görevini yapması için robota geri dönüş ve giriş verisi sağlarlar. Her zaman gerekli değildir.


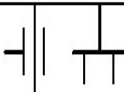




2.1.1 Robot tipleri ve sınıflandırılması

Robotun nasıl çalıştığını anlamak için önce robotun nasıl hareket ettiğini algılamak gerekir robotların yaptığı 2 temel hareket vardır.

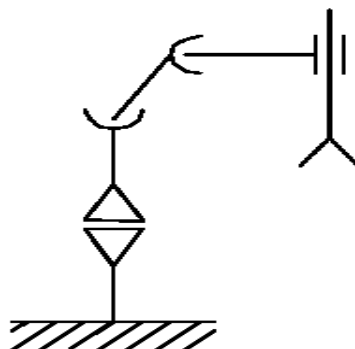
- Enine (Transverse) Hareket
- Dönme (Rotational) Hareket

Enine hareket ileri doğru sabit bir hareket veya teleskopik bir harekettir. Buna karşın dönme hareketi bir eksen çevresinde dönme veya menteşe hareketidir, bu hareketlerin sembolleri Çizelge 2. 1’de görülmektedir.

Çizelge 2.1 Kinematik zincir tablosu

Sistem		Sembol
Transfer Ekseni	Teleskop	
	Enine	
Dönme Ekseni	Eksen Çevresi Dönme	
	Mentese Hareketi	
		
		

Bu sembol tablosu kullanılarak 4 derece özgürlüğe sahip robot modelin Kinematik zincir tablosu şekil 2.1’de gözlemlenebilmektedir.


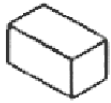




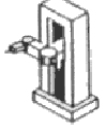





Şekil 2.1 Dört derce özgürlüğe sahip robot örneği

Dört derece özgürlüğe ve bir grip nesnesine sahip robot modelin kinematik zincir modeli, bir eksen çevresi dönme hareketi, iki menteşe hareketi ve bir enine hareketten oluşur. Robotları sınıflandırmanın bir diğer yolu da onları özgürlük derecelerine göre (DOF) sınıflandırmaktır. Her bir enine veya dönme hareketi için robota bir adet özgürlük derecesi eklenir. Yukarıda ki robot kol modeli 4 özgürlük derecesine sahiptir çünkü grip nesnesin pense hareketi özgürlük derecesi olarak kabul edilmez. Özgürlük derecesi bir robot sistemin ne kadar karmaşık olduğunu ve onun yönetiminin ne kadar karmaşık olduğunu bize gösteren önemli bir sınıflandırma metodudur (Banka and Lin 2002).

Eğer bir robot kolun özgürlük derecesi üç veya daha fazla ise çalışma uzayında verilen hedef noktaya ulaşmanın birden çok yolu var demektir. Bu da robot mekanizmasının karmaşıklığını iyice artırır, Eğer robotun özgürlük derecesi üç veya üstü ise çalışma uzayında ki herhangi bir noktaya herhangi bir doğrultuda ulaşmanın birden çok yolu var demektir (Niku 2001). Çalışma uzayı tutaç'ın (end effector) ulaşabileceği noktalardan oluşur. Çizelge 2.2'de değişik tipteki robotların çalışma grupları ile sınıflandırmalarını gösteriliyor.

Çizelge 2.2 Ana Robot Tipleri ve Çalışma Uzayları

Tip	Örnek Resim	Çalışma Uzayı
Cartesian		
Cylindrical		
Spherical		
Horizontal articulated		
Vertical articulated		

Kartezyen(Cartesian) manipülatör: Bu tip bir manipülatör üç tane kayar tip eklem ile elde edilir. Mekanik yönden çok sağlamdır fakat çalışma uzayındaki hareket yeteneği bakımından zayıftır. Bu tip manipülatörler çok büyük boyutlarda ve ağırlıklarda nesneleri hareket ettirmek ve taşımak için idealdir. Kartezyen manipülatörlerde eklemleri hareket ettiren motorlar çoğunlukla elektrik bazen de pönomatik motorlarıdır

Silindirik(Cylindrical) manipülatör: Bu tip bir manipülatör bir tane döner ve iki tane kayar tip eklem ile elde edilir. Bu tip manipülatörler de mekanik yönden sağlamdır fakat bilek konum doğruluğu(accuracy) yatay harekete bağlı olarak azalır. Benzer şekilde büyük boyutlu nesnelerin taşınmasında kullanılırlar. Bu tip manipülatörlerde hidrolik motorları tercih edilir.

Küresel(Spherical) manipülatör: Bu tip bir manipülatör iki tane döner ve bir tane kayar tip eklem ile elde edilir. Bu tip manipülatörler mekanik yönden diğer iki tipten daha zayıf, mekanik yapı yönünden daha karmaşıktır. Çoğunlukla makine montajlarında kullanılırlar. Bu tip manipülatörlerde elektrik motorları tercih edilir.

Mafsallı (Vertical articulated) manipülatör: İnsan kol yapısı esas alındığı için bu isim verilmiştir. Bu tip manipülatörler tüm eklemleri döner olduğundan çalışma uzaylarında en yetenekli manipülatörlerdir. Endüstriyel uygulamalarda geniş kullanım alanına sahiptirler.(Boyama, kaynak yapma, montaj, yüzey temizleme vb.) Bu tip manipülatörlerde elektrik motorlar tercih edilir (Craig 2005).

Endüstriyel robot, genel amaçlı, insana benzer özelliklere sahip ve programlanabilir bir makinedir. Bir robotun insana benzeyen en önemli özelliği onun koludur. Tutma ve yerleştirme işlemlerinde robot kolu kullanılır. Robot kolu, başka bir makineyle birleştirilerek, malzemenin yüklenmesi ve bir takım değiştirme işlemini yapmaktadır. Robotlar, kesme, şekil verme, yüzey kaplama, silindirik ve düzlem yüzey taslama gibi imalat işlemlerini gerçekleştirir; montaj ve kontrol uygulamalarında da kullanılmaktadır. Robot kolların kullanım Alanları ve kullanım sonuçlarını da içeren konfigürasyonlarının daha detaylı karşılaştırılması Çizelge 2.3’de verilmektedir.

Çizelge 2.3 Robot Kolların Konfigürasyonlarının Karşılaştırılması

Robot Tipi	Mafsal Tipleri -ilk eklem -ikinci eklem -üçüncü eklem	Kullanım Alanları	Kullanım Sonuçları
Kartezyen Robot Kollar	Prizmatik Prizmatik Prizmatik	Demiryolu, köprü inşaatları Büyük makine montajları	Kinematik modelleri basittir. Rijit bir gövdeye sahiptir. Hareket analizi basittir. Çalışması için büyük alan gerekir. Büyüklüğüne göre iş alanı küçüktür.
Silindirik Robot Kollar	Dönel Prizmatik Prizmatik	Büyük makine montaj sanayi Basit montaj-demontaj hatları	Kinematik modelleri basittir. Hareket analizi basittir. Güçlü hidrolik elemanlar kullanılır. İş alanları sınırlıdır. Tozlu ve ıslak ortamlarda çalışmaları zordur.
Küresel Robot Kollar	Dönel Dönel Prizmatik	Montaj sanayi Nükleer santraller	Büyük alanlara uzanabilirler. Zeminden uzaktaki nesneleri tutabilirler. Kinematik modelleri karışıktır. Hareket analizi zordur.
Mafsallı Robot Kollar	Dönel Dönel Dönel	Otomobil sanayi Otomobil boya sanayi Elektronik montaj sanayi Nükleer santrallerde Tıbbi araç-gerek yapım sanayi	Maksimum esnekliğe sahiptir. İş alanı robot büyüklüğü ile orantılıdır. Elektrik motorları kullanılabilir. Cisimleri altlarından tutabilirler. Kinematik yapıları karmaşıktır. Hareket analizleri zordur. Kolların rijitlik ayarı zordur.

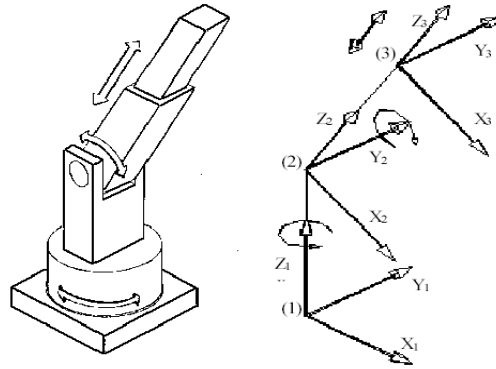
2.2 Robotla İlişkili Matematik

Bu bölüm kapsamında, robot bilimi ile ilgili temel matematiksel kavramlar ele alınmıştır. Temel olarak robotun elamanlarını çalışma uzayında konumlanması ve matris matematiğinden bahsedilmiştir.

2.2.1 Koordinat sistemleri

Robotlar kendilerinin ve çevrelerindeki nesnelerin bulunduğu 3 boyutlu uzayda hareket ederler. Robotun ve çevresindeki nesnelerin konumlarını ve birbirlerine göre yönelimlerini belirlemek için robotun ve çevresindeki nesnelerin merkezlerine birer koordinat sistemi yerleştirilir. Tanımlanacak bütün konum ve yönelimler evrensel çerçeveye veya evrensel çerçeve içindeki diğer Kartezyen koordinat sistemlerine göre gerçekleştirilir. Robot sistemlerinin çalışma uzaylarında belirlenen noktalara gitmesini sağlamak için koordinat sistemleri ile ilişkilendirilmişlerdir. XYZ koordinat sistemi genelde kullanılan koordinat sistemidir, bu çalışma içinde bu koordinat sistemi tercih edilmiştir. Robotlar her bir özgürlük derecesi için bir küresel(global) bir yerel(local)

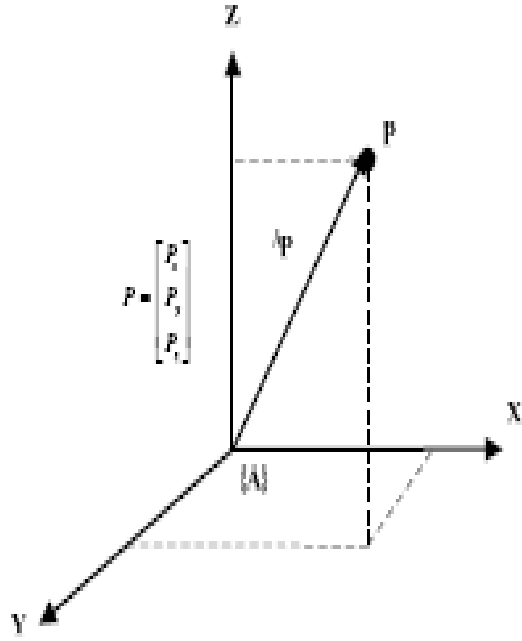
koordinat sistemine sahiptirler. Global koordinat sistemi robot ya robot üzerinde sabit bir noktaya bağlıdır, ya da çalışma uzayında uygun bir noktadadır. Üç dereceli özgürlüğe sahip robot kol Şekil 2.2’de gösterilmektedir.



Şekil 2.2 Üç özgürlük derecesine sahip robot modeli.

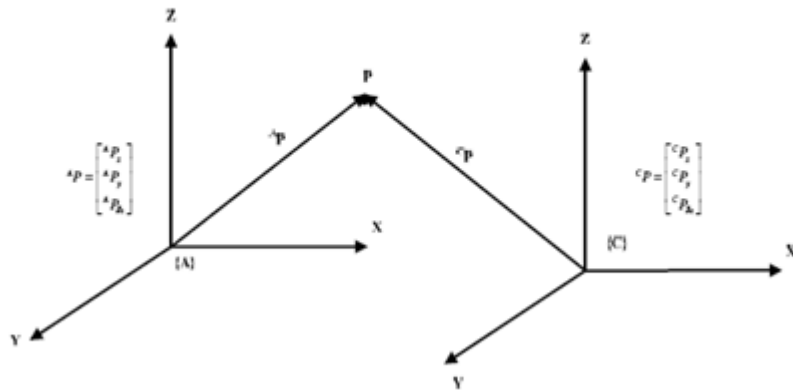
Robotun yerel koordinat sistemini belirlemek için Şekil 2.2’de robot kolun sağ tarafında koordinat sistemleri çizilmiştir. Birinci koordinat sistemi z ekseninde dönme hareketini sağlamaktadır (1) . İkinci koordinat sistemi ise kendi x ekseninde dönme hareketini göstermektedir (2) . Enine hareketi için üçüncü koordinat sistemi dönme değil öteleme hareketini yapar (3) . Matrislerin kullanımı Robotun her hangi bir ekleminin yönelme ve ilerleme hareketinin küresel(global) koordinatlarla ilişkisini sağlar robot uygulamalarında ki matris yaklaşımı bir sonraki kısımda anlatılmıştır.

Konum tanımı: Bir evrensel koordinat çerçevesi içerisine birçok koordinat sistemi yerleştirilebilir. 3 boyutlu uzayda bir nokta, bu koordinat sistemlerinin merkezine göre tanımlanmış 3x1 boyutlu vektörle gösterilebilir. Evrensel koordinat çerçevesi içerisinde bir $\{A\}$ koordinat sistemi ve P noktası olduğunu düşünelim. P noktasının $\{A\}$ koordinat sistemine göre tanımı Şekil 2.3’de verilmiştir.



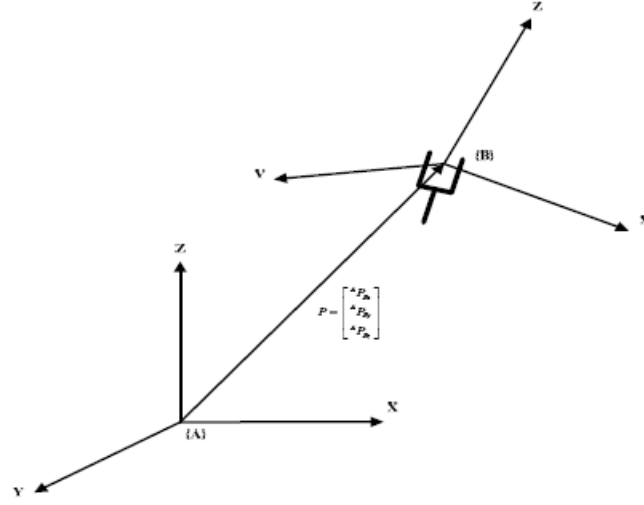
Şekil 2.3 P Noktasının A Noktasına Göre Tanımı

Aynı P noktası hem $\{A\}$ hem de $\{C\}$ koordinat sistemlerine göre de tanımlanabilir. Burada P noktasının $\{A\}$ ve $\{C\}$ koordinat sistemlerine uzaklığı eşit olmak zorunda değildir.



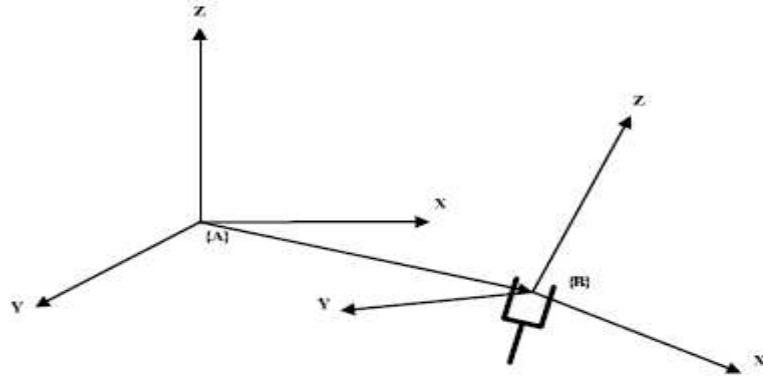
Şekil 2.4 P noktasının $\{A\}$ ve $\{C\}$ koordinat sistemlerine göre tanımı

Daha önce de belirtildiği gibi robot ve çevresindeki nesnelere koordinat sistemi yerleştirilir. Şekil 2.5'te bir robotun uç işlevcisinin A noktasına uzaklığını tanımlamak için, A noktasına ve robotun uç işlevcisine koordinat sistemleri yerleştirilmiştir. A noktası ile uç işlevcisine yerleştirilen koordinat sistemlerinin merkezleri arasındaki uzaklık şekilde görülmektedir (Küçük 2005).



Şekil 2.5 A Noktası ve uç noktanın koordinat sistemleri

Yönelim tanımı: 3 boyutlu uzayda, bir noktanın herhangi bir koordinat sistemine göre konumunun yanında yönelimi de tanımlanır. Yönelim, bir koordinat sisteminin başka bir koordinat sistemine göre dönme miktarıdır ve 3x3 boyutlu matris olarak ifade edilir. Bir katı cismin yönelimini başka bir referans koordinat sistemine göre tanımlamak için katı cisme bir koordinat sistemi yerleştirilir. Şekil 3.4'te uç işlevcisine, {B} koordinat sistemi yerleştirilerek {A} referans koordinat sistemine göre yönelimi tanımlanır (Küçük 2005).



Şekil 2.6 {B} Koordinat sisteminin {A} sistemine göre yönelimi

2.2.2 Matris matematiği

Matrisler noktaları(points), vektörleri(vectors) , çerçevelerin (frames) , öteleme, dönme ve aktarma işlemlerini sunmak için kullanılırlar. Robot kolun parçalarının çalışma uzayında küresel(global) koordinat sistemleri ile ilişkisini belirlemek için matris işlemleri yapılır.

Yönelme işlemi parametrik hale getirilip homojen dönüşümler kullanılır. Robot sistemlerinde her bir eklemin pozisyonu ve yönelmesi matris gösteriminde sunulur. $P = [i,j,k]$ vektörü o anki koordinat sisteminin bir önceki koordinat sistemine göre yerini; $X=[x_1,x_2,x_3]$, $Y=[y_1,y_2,y_3]$, $Z=[z_1,z_2,z_3]$ ise bir önceki koordinat sistemine göre yönelmesini temsil eder. X, Y ve Z vektörleri birim vektörleridir. A Matrisi, yönelme ve pozisyon alma işlemlerini temsil eden matrisi göstermektedir.

$$A = \begin{bmatrix} \bar{x} & \bar{y} & \bar{z} & \bar{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 & i \\ x_2 & y_2 & z_2 & j \\ x_3 & y_3 & z_3 & k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 2.7 Yönelme ve pozisyon matrisi

Her bir eklemin koordinat sisteminin bir önceki koordinat sistemi ile ilişkisi incelenerek; homojen aktarım matrisleri elde edilir. Robotun uç noktasının küresel koordinat sistemi ile ilişkisini anlamak için bütün matrisler çarpılır. Şekil 2.8’de son eklemin başlangıç durumuna göre konumunu ve yönelimini bulmak için yapılan matris çarpma işlemleri görülüyor (Mikkelsen 1998) .

$$M(AC) = A * B * C = \begin{bmatrix} x_1 & y_1 & z_1 & i \\ x_2 & y_2 & z_2 & j \\ x_3 & y_3 & z_3 & k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 2.8 Üçüncü eklemin yönelme ve pozisyon matrisi

$M(AC)$: Üçüncü Eklemin Küresel koordinat sistemi ile ilişkisini gösteren Matris

A: Küresel koordinat sistemi ile ilişkili birinci eklemin matrisi gösterimi

B: Birinci Eklemin koordinat sistemi ile ilişkili ikinci eklemin matris gösterimi

C: İkinci Eklemin koordinat sistemi ile ilişkili üçüncü eklemin matris gösterimi

x: Üçüncü eklemin x koordinatının küresel koordinat ile ilişkisi

y: Üçüncü eklemin y koordinatının küresel koordinat ile ilişkisi

z: Üçüncü eklemin z koordinatının küresel koordinat ile ilişkisi

i,j,k: Üçüncü eklemin küresel koordinat sistemine göre pozisyonu

3. PROJEDE KULLANILACAK ROBOT KOL SEÇİMİ VE YAPISI

Bu tez kapsamında Lynx firmasının ürettiği, Altı eklemlilik Beş Özgürlük Derecesine sahip (5 DOF) Lynx-6 Robot Kolu kullanılmıştır.

3.1 Robot Kol Seçimi

Bu robot kolun tercih edilme sebebi, yüksek lisans seviyesinde bu tip bir tez çalışmasında kullanılabilecek en uygun robot kol olmasıdır. Daha yüksek hız ve daha hassas hareket sağlayan robotlar olmasına rağmen bu üstün robotların sadece endüstride kullanıldığından ve bu tarz robotların maliyetinin çok yüksek olması sebebiyle bu tez kapsamında piyasada bulunanlar arasında her yönden en iyisi olan, Beş Özgürlük Derecesine sahip (5 DOF) Lynx-6 robot kolu kullanılmıştır. Bu robot kolu hızlı, hassas ve tekrarlı hareketler yapmaya uygundur Seçilen Robot Kol aşağıda ki şekilde görülmektedir.



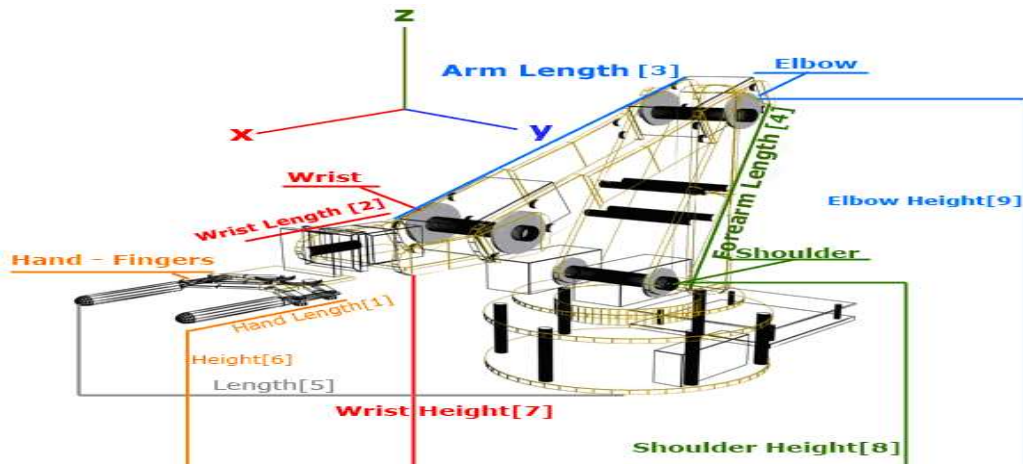
Şekil 3.1 Lynx-6 robot kol

3.2 Lynx Altı Eksenli Robot Kol

Bu bölümde Lynx firmasına ait robot kolun limitleri ve imkânları ortaya çıkarılmış, robot geometrisinden bahsedilip, servo motorlar ve bu motorları yöneten servo kontrol kartından bahsedilmiştir, kartın yönetilmesi için gereken parametre tablolarından ve kart ile motorlar arasında ki ilişkilerden bahsedilmiştir.

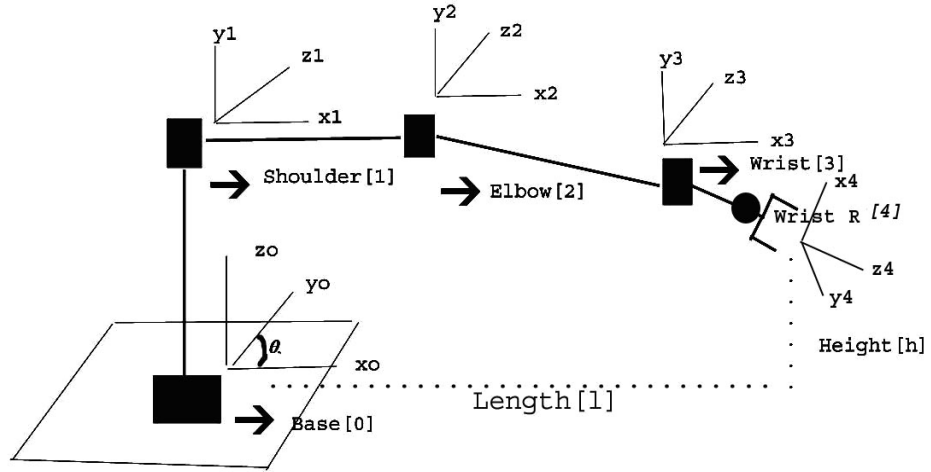
3.2.1 Robot kolun geometri ve limiti

Bu çalışmada Lynx firmasına ait, Lynx-6 robot kolu kullanılmaktadır. Bu robot kol hobi ve akademik çalışmalarda kullanılmak için tasarlanmıştır, bu yüzden aynı işlevi gerçekleştiren endüstriyel bir robotla hareket doğruluğu bakımından karşılaştırılmaz; Fakat bu tez kapsamında ki uygulamalar için yeterli niteliğe sahiptir. Lynx-6 altı ekleme sahiptir. Bu eklemlerin üç tanesi menteşe (hinge movement) yapmaktadır. Buna karşı diğer iki eklem eksen etrafında dönme (Pivot movement) hareketi gerçekleştirmektedirler; son eklemden tutuş (grip) eklemidir. Bu robot kol dikey mafsallı (vertical articulated) bir robot yapıya sahiptir. Robot kolun eklem isimleri ve üç boyutlu modellenmesi Şekil 3.2’de gösterilmektedir.



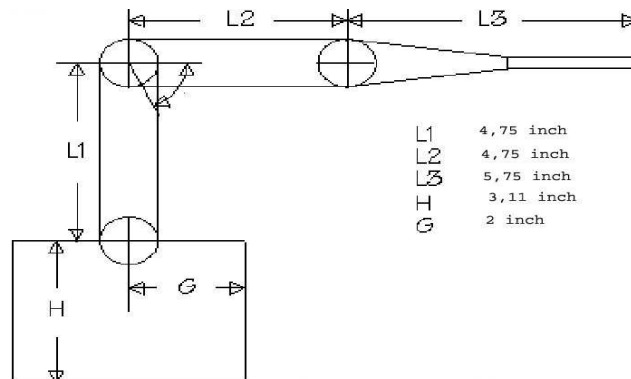
Şekil 3.2 Lynx-6 Robot Kolun üç boyutlu modeli

Lynx-6 robot kolun üç boyutlu modeli yukarıda ki figürde görülmektedir. Robot kolun altı işlevsel eklemi sırasıyla Taban, Omuz, Dirsek, Bilek, Bilek Dönmesi ve Tutaç'tır. Robot kolun yerel koordinat sistemi Şekil 3.3'de görülmektedir.



Şekil 3.3 Lynx-6 Robot kolun yerel koordinat sistemi

Global koordinat sistemi ve eklemler arası uzaklıktan şekil 3.3'te görülen yerel(local) koordinat sistemleri çıkarılır. Robotun eklemleri arasında ki uzaklık değerleri xz düzleminde şekil 3.4'de gösterilmiştir.



Şekil 3.4 Altı eksenli robot kolun eklem uzaklık değerleri

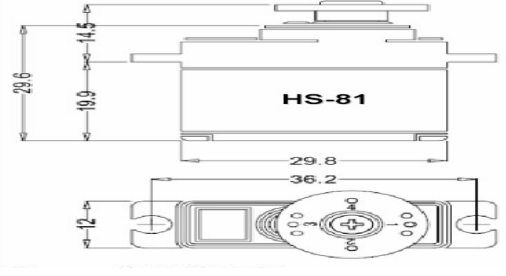
Lynx Robot kolun her bir ekleminin, en büyük ve en küçük açı değerleri, Çizelge 3.1’ de gösterilmektedir. Tabloda da görüldüğü gibi, bir eklemin sahip olduğu aralık değeri - 90 derece ile + 90 derece arasındadır. Kol toplam olarak 180 derecelik hareketi yapabilmektedir (www.lynxmotion.com 2007).

Çizelge 3.1 Eklemlerin Limitleri

EKLEM	EN KÜÇÜK AÇI	EN BÜYÜK AÇI
TABAN	—90	+90
OMUZ	—90	+90
DİRSEK	—65	115
BİLEK	—90	90
BİLEK DÖNME	—90	90
TUTAÇ	—45	45

3.2.2 Robot kol için servo motorlar

Lynx-6 manipülatörünün eklemlerini hareket ettirmek için altı titreşim orantılı servo motor kullanılmıştır. Bu motorlar radyo kontrollü araba uçak ve bot için tasarlanmış motorlardır. Normal menzili 90 derece, olmasına karşın Lynx-6 için motorlar 180 dereceye menzile çıkarılmıştır. 60 Hz kare dalga titreşim(pulse) bu motorları kontrol etmek için kullanılmaktadır. Servo motorların pozisyonu sinyal genişliğine göre belirlenmektedir. 0.5 ms ve 2.53 ms arası sinyal genişliği 180 derece dönüşü kontrol etmektedir. Bir sonraki bölümde anlatılacak olan servo kontrol kartı (SSC-32) servo motorları titreşim sinyalleri ile besler (www.lynxmotion.com 2007). Lynx-6 robot kolu 3 adet HS-475 HB servo motoru omuz dirsek ve bilek eklemleri için kullanır. Bir adet HS-85 BB servo motoru taban eklemleri için; Bir adet HS-422 servo motoru bilek dönmesi ve bir adet HS-81 servo motoru tutaç hareketi için kullanır şekil 3.5’de HS-81 görülmektedir.



Şekil 3.5 HS – 81 mikro servo motoru

3.2.3 Robot kol için servo kontrol kartı

Lynx-6 robot kol için kullanılan kontrol kartı SSC-32 (Seri Servo Kontrol) kartıdır. Bu kart bilgisayardan gelen sinyali servo motorların kullanacağı kare dalgaya çevirir. Kontrol kartı aşağıda ki şekilde görülmektedir.



Şekil 3.6 SSC-32 Servo kontrol kartı

SSC-32 Servo kontrol kartı otuz iki servo motoru kontrol edecek kare dalgayı çıkış olarak verme kapasitesine sahiptir. Servo motorları kontrol etmek için bilgisayardan seri giriş noktası gerekmektedir. SSC-32 Servo kontrol kartı için Çizelge 3.2 ile aşağıda gösterilen formatta verilen komutlarla servo motorlar kontrol edilip yönetilirler.

Çizelge 3.2 SSC–32 için Komut Format Tablosu

<ch>	Kanal Numarası 1 – 31 arası
<pw>	Pulse Genişliği 500 – 2500 mikro saniye arası
<spd>	Her bir kanalın hareket hızını belirler
<time>	Aktif olan hareketler için toplam zamanı öngörür
<cr>	ASCII 13 karakterini karşılar
<esc>	Aktif komutu iptal eder ASCII 27

Örnek : “#5 P1500 T1000 <cr>” komutu, Beşinci kanalda ki servo motorun pozisyonunu 1500 (Başlangıç Pozisyonu) yapar ve bu işlem 1 saniye sürer.

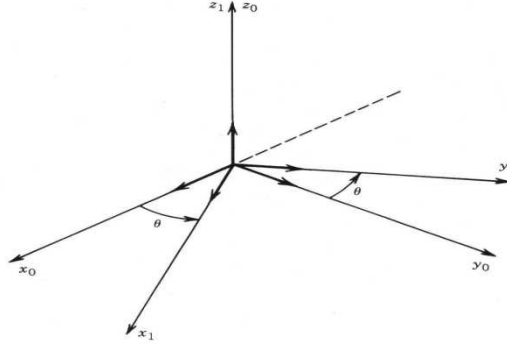
4. KİNEMATİK ANALİZ VE POZİSYON BELİRLEME

Bir robot kola ilişkin iş planlaması, yörünge planlaması, dinamik ve kontrol problemleri ele alındığı zaman ilk gereksinme duyulan hususlardan biri, bu robot kolun kinematik modelinin oluşturulması ve buna dayanarak gerekli kinematik ilişkilerin ve analizin elde edilmesidir. Robot kollar için kinematik ilişkiler, iki farklı yönde elde edilirler: Birincisi El'in zemine göre konum ve yönelimini eklem değişkenleri cinsinden belirleyen "düz kinematik" ilişkileridir. İkincisi ise El'in zemine göre verilen konum ve yönelimini sağlayacak olan eklem değişkenlerini belirleyen "ters kinematik" ilişkileridir. Ters kinematik ilişkilerin çözümü, Robot kolun serbestlik derecesine ve topolojik yapısına (döner ve kayar eklemlerinin sayısına ve sıralanışına) bağlı olan ve genelde lineer olmayan bağlaşıklık denklemlerin çözümünü kapsar. Robot Kol eklemlerle birbirine bağlanmış rijid cisimlerin açık uçlu kinematik zinciri olarak kabul edilir. Zincirin bir ucu yere bağlı iken diğer ucu sonlandırıcıya bağlıdır. Sonuçta bu yapının hareketi her bir uzvun diğerine göre hareketlerinin toplamından oluşturulur. Bunun için önce bir rijid cismin uzaydaki konumunu ve yönelimini belirtmek için dönme (rotation) matrisi ve öteleme (translation) vektörü oluşturulur. Bu matris ve vektörü birleştirmek için daha sonra homojen dönüşüm gösterimi kullanılır.

4.1 Dönme Matrisi

Şekil 4.1' de gösterildiği üzere $\{i_0, j_0, k_0\}$ $ox_0y_0z_0$ koordinat çerçevesi için , $\{i_1, j_1, k_1\}$ $ox_1y_1z_1$ koordinat çerçevesi için birim vektörler olsun. 1. koordinat çerçevesi 0. koordinat çerçevesinde z_0 eksenine etrafında θ açısı kadar döndürülerek elde edilmiştir. Bu iki koordinat çerçevesi arasındaki dönüşüm Eşitlik 4.1 ile bulunur. Buradaki R_0^1 matrisi 1. koordinat çerçevesinden 0. koordinat çerçevesine dönme matrisini göstermektedir. 1. koordinat çerçevesi 0. koordinat çerçevesinden belli bir dönme ile elde edildiğinden bu matris dönme matrisi adını alır.

$$R_0^1 = \begin{bmatrix} i_0 \cdot i_1 & j_0 \cdot i_1 & k_0 \cdot i_1 \\ i_0 \cdot j_1 & j_0 \cdot j_1 & k_0 \cdot j_1 \\ i_0 \cdot k_1 & j_0 \cdot k_1 & k_0 \cdot k_1 \end{bmatrix}$$



Şekil 4.1 z_0 eksenini etrafında dönme

Buradan $i_0 \cdot i_1 = \cos \theta$, $j_1 \cdot i_0 = -\sin \theta$, $j_0 \cdot j_1 = \cos \theta$, $i_1 \cdot j_0 = \sin \theta$ $k_0 k_1 = 1$

$$R_0^1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

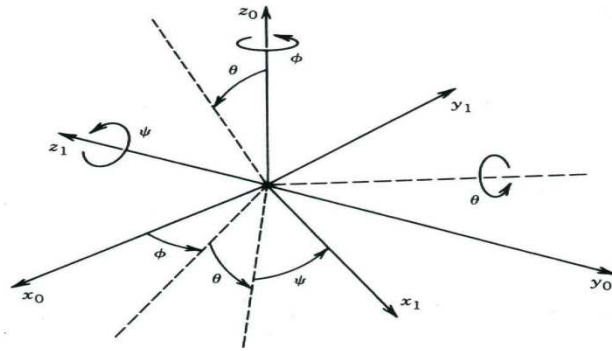
Şekil 4.2 Dönme matrisi

Dönme matrisi elde edilir, şekil 4.2 de görülmektedir, R_0^1 gösterilimi yerine dönme eksenini ve açısını belirten $R_{z,\theta}$ gösterilimi de kullanılabilir. Koordinat çerçeveleri etrafında dönme tek olmayabilir, böylece devam eden şekilde $ox_2y_2z_2$, $ox_3y_3z_3$, $ox_4y_4z_4$... Koordinat çerçeveleri elde edilebilir. Aslında sonuçta elde edilen koordinat çerçevesi sadece üç eksen etrafında (x,y,z) belirli açılarla dönmüştür. Bu nedenle elimizde bulunan rastgele bir koordinat çerçevesinin temel koordinat çerçevesine göre dönme matrisini elde etmek için üç tane açı değeri yeterlidir. Bu açıların tanımlanması için iki tane gösterilim mevcuttur.

1. Euler açıları gösterilimi
2. Roll / Pitch / Yaw açıları gösterilimi

4.1.1 Euler açıları ve gösterilimi

Euler açıları sırasıyla z eksenini etrafında Φ açısı kadar, y eksenini etrafında θ açısı kadar, tekrar z eksenini etrafında Ψ açısı kadar dönmelere karşılık gelmektedir. Euler açıları gösterilimi Şekil 4.3'te gösterilmiştir.



Şekil 4.3 Euler açıları gösterimi

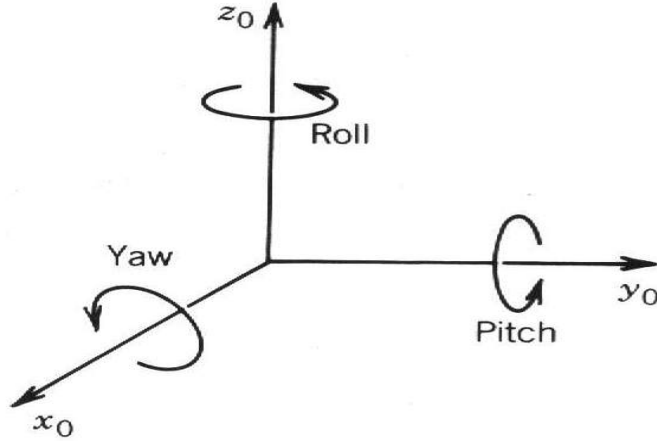
Burada $\cos = c$, $\sin = s$ olmak üzere dönme matrisi şu şekilde elde edilir. Toplam, Euler matrisi aşağıda ki şekilde görülmektedir.

$$\begin{aligned}
 R_0^1 &= R_{z,\phi} R_{y,\theta} R_{z,\psi} \\
 &= \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\phi c\theta c\psi - s\phi s\psi & -c\phi c\theta s\psi - s\phi c\psi & c\phi s\theta \\ s\phi c\theta c\psi + c\phi s\psi & -s\phi c\theta s\psi + c\phi c\psi & s\phi s\theta \\ -s\theta c\psi & s\theta s\psi & c\theta \end{bmatrix}
 \end{aligned}$$

Şekil 4.4 Euler dönüşüm matrisi

4.1.2 Roll/Pitch/Yaw açıları ve gösterilimi

Roll / Pitch / Yaw açıları sırasıyla z eksenini etrafında Φ açısı kadar, y eksenini etrafında θ açısı kadar ve x eksenini etrafında Ψ açısı kadar dönmelere karşılık gelmektedir. Roll / Pitch / Yaw açıları gösterilimi şekil 4,5 'de verilmiştir.



Şekil 4.5 Roll / Pitch /Yaw açıları gösterilimi

Burada toplam öteleme matrisinin oluşumu ve değeri şekil 4.6'da gösterilmektedir.

$$R_0^1 = R_{z,\phi} R_{y,\theta} R_{x,\psi}$$

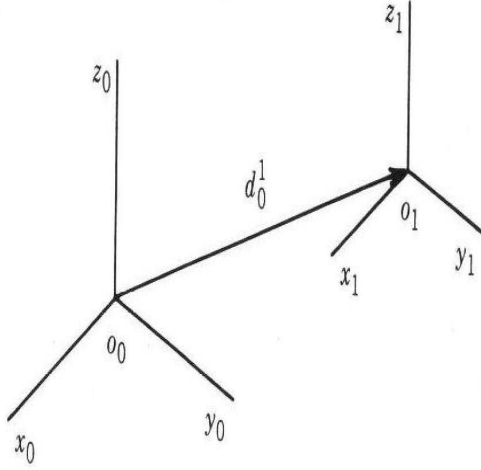
$$= \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi & -s\psi \\ 0 & s\psi & c\psi \end{bmatrix}$$

$$= \begin{bmatrix} c\phi c\theta & -s\phi c\psi + c\phi s\theta s\psi & s\phi s\psi + c\phi s\theta c\psi \\ s\phi c\theta & c\phi c\psi + s\phi s\theta s\psi & -c\phi s\psi + s\phi s\theta c\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}$$

Şekil 4.6 Roll / Pitch /Yaw açıları öteleme matrisi

4.2 Öteleme Vektörü

Şekil 4,7 'da gösterildiği üzere $oxyz$ 'den $|d|$ kadar bir öteleme ile elde edilmiş $oxyz$ koordinat çerçevesi düşünelim. Bu iki koordinat çerçevesi arasındaki dönüşüm öteleme vektörü d_0^1 ile tanımlanır ve şekil 4,8 de gösterildiği gibi öteleme vektörü ifadesi ile temsil edilir.



Şekil 4.7 Ötelenmiş çerçeve

$$d_0^1 = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

Şekil 4.8 Öteleme vektörü

4.3 Homojen Dönüşüm

Üç boyutlu uzayda koordinat çerçeveleri arasındaki dönüşüm dönme matrisleri ve öteleme vektörleri yardımıyla yapılır. Her ikisinin birlikte gösterilimi için yani koordinat çerçeveleri arasında hem dönmenin, hem de ötelemenin var olduğu durumda homojen dönüşüm matrisleri kullanılır. Homojen dönüşüm matrisleri aşağıdaki gibi elde edilir.

$$T_0^1 = \left[\begin{array}{ccc|c} & & & \\ & R_0^1 & & d_0^1 \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right]_{4 \times 4}$$

Şekil 4.9 Homojen dönüşüm matrisi

Burada elde edilen T_0^1 matrisi 1. koordinat çerçevesinden 0. koordinat çerçevesine homojen dönüşüm matrisini göstermektedir. T_0^1 Matrisini 4x4 boyutundan kare matris olduğuna dikkat edilmelidir. Bu homojen matris oluşturulurken matrisin tersinin alınabilmesi için yapılmıştır. T_0^1 Matrisindeki $T_{4,4}$ elemanı olan 1 tüm elemanların bire bir ölçeklendiğini göstermektedir. Bu elde ettiğimiz genel kalıba bağlı olarak temel homojen dönüşüm matrisleri şekil 4.10 ‘da gösterilmektedir (Jeppe Mikkelsen,1988).

$$\begin{aligned} R_{x,\alpha} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & R_{y,\phi} &= \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & R_{z,\theta} &= \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ d_{x,a} &= \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & d_{y,b} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & d_{z,c} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Şekil 4.10 Temel Homojen Dönüşüm Matrisleri

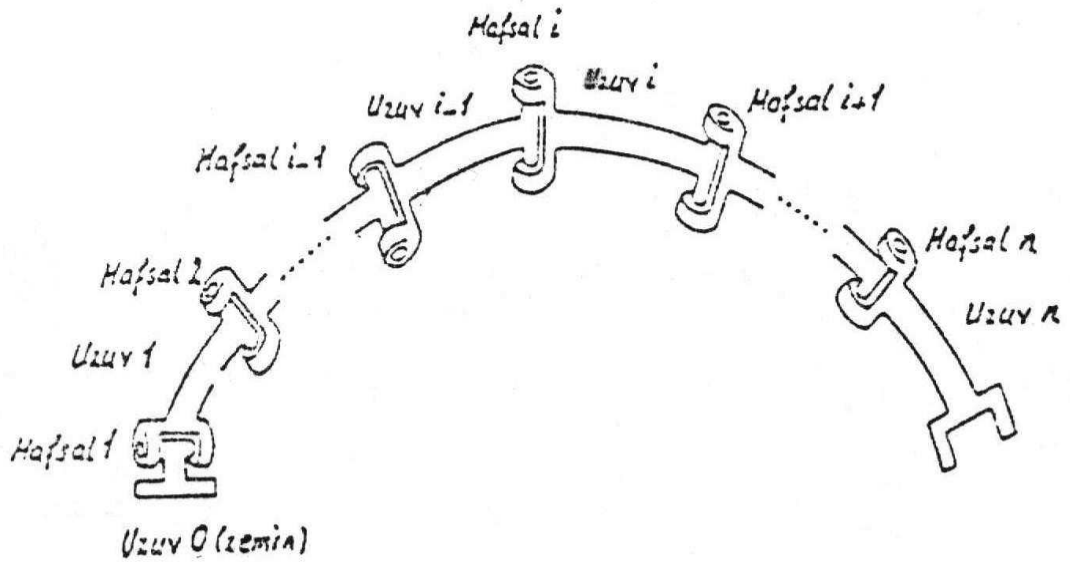
4.4 Düz Kinematik Analiz

Robotikte düz kinematikler manipülatörün verilen eklem değişken değerleri için sonlandırıcının konumunu ve yönelimini bulmak olarak tanımlanabilir. Eklem değişkenleri, eklemin döner olması durumunda uzunlar arasındaki açı, eklemin kayar olması durumunda uzuv uzanma miktarıdır. Bu değişkenler için gösterilimler aşağıda şekil 4.11’de verilmiştir.

$$q_i = \begin{cases} \theta_i & \text{döner eklem için} \\ d_i & \text{kayar eklem için} \end{cases}$$

Şekil 4.11 Eklem değişken tipleri ve parametreler

Manipülatorlerin eklemlerle birleştirilmiş bir dizi uzuv olduğundan yola çıkarak Şekil 4.12’de gösterildiği üzere $n+1$ uzva sahip bir manipülatörümüz olduğunu ve her bir uzva bir koordinat çerçevesi atandığını düşünelim.



Şekil 4.12 $n+1$ uzva sahip kinematik zincir oluşturan manipülator

Şimdi T_{i-1}^i 'in çerçeve i 'den çerçeve $i-1$ 'e homojen dönüşüm matrisi olduğunu düşünelim. Burada T_{i-1}^i matrisinin sabit olmadığına, manipülatorün hareketiyle değiştiğine dikkat edilmelidir. Bu ifadelerle ilgili olarak sonlandırıcının konumunu ve yönelimini temel(base) koordinat çerçevesinde elde etmek için homojen dönüşüm matrisine H dersek, şu şekilde ifade edebiliriz: $H = T_0^n = T_0^1 \cdot T_0^2 \cdot T_{n-1}^n$

4.4.1 Denavit-Hartenberg(DH) dönüşümü

Eşitlik 4.4.1 ile elde edilen dönüşüm kolay gibi görünse de her bir eklem için koordinat çerçevelerinin yerleştirilmesi ve birbirlerine göre yorumlanması anlam karmaşasına yol açmaktadır. Bu anlam karmaşasını ortadan kaldırmak için Denavit ve Hartenberg 1955 yılında sistematik bir yöntem önermişlerdir. Bu yöntemde aşağıdaki kurallara göre önce koordinat çerçeveleri atanır, daha sonra dönüşüm için gerekli uzuv ve eklem parametreleri bulunur. Yöntem dokuz adımdan oluşmaktadır ve bu adımlar aşağıda verilmiştir (Hartenberg 1955).

1. Eklem eksenleri $z_0 \dots z_{n-1}$ 'i konumlandır.
2. Temel çerçeveyi sağ el kuralına göre düzenle.
 $i = 1, \dots, n-1$ için adım 3-5'i tekrarla.
3. Merkez o_i 'yi yerleştir. Eğer z_i z_{i-1} ile kesişiyorsa o_i 'yi bu noktaya yerleştir.
Eğer z_i z_{i-1} paralelse o_i 'yi eklem i üzerine yerleştir.
4. x_i 'yi yerleştir. Eğer z_i z_{i-1} kesişiyorsa ikisinin oluşturduğu düzleme dik olarak x_i 'yi yerleştir. Eğer z_i z_{i-1} paralelse bunların ortak normalleri boyunca o_i 'ye doğru x_i 'yi yerleştir.
5. Sağ el çerçevesini tamamlayacak şekilde y_i 'yi yerleştir.
6. Sonlandırıcı çerçevesi $o_n x_n y_n z_n$ 'i yerleştir. Bu yerleşim sonlandırıcı tipine göre değişir.

Bu ilk altı adım koordinat çerçevelerinin yerleştirilmesi için kullanılır, daha sonraki üç adım bize eklem ve uzuv parametrelerini verir.

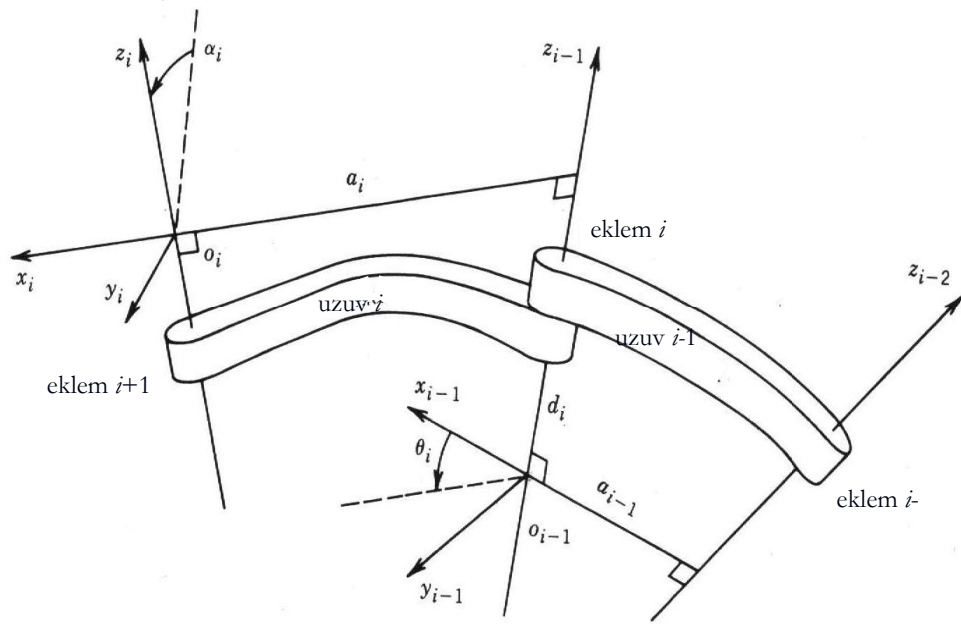
7. Daha sonra eklem ve uzuv parametreleri belirle. Bu parametreler aşağıdaki gibidir ve Şekil 4.13 üzerinde gösterilmiştir.

uzuv uzunluğu a_i : x_i boyunca o_i 'den x_i ve z_{i-1} 'nin kesişimlerine olan uzaklık

uzuv ofseti: d_i : z_{i-1} boyunca o_{i-1} 'den x_i ve z_{i-1} 'in kesişimlerine olan uzaklık.

uzuv bükümü: α_i : x_i etrafında z_{i-1} z_i arasındaki açı

eklem açısı: θ_i : z_{i-1} etrafında x_{i-1} x_i arasındaki açı



Şekil 4.13 Denavit-Hartenberg çerçeve ataması

8. Aşağıdaki matrise bağlı olarak iki çerçeve arasındaki homojen dönüşüm matrisini hesapla.

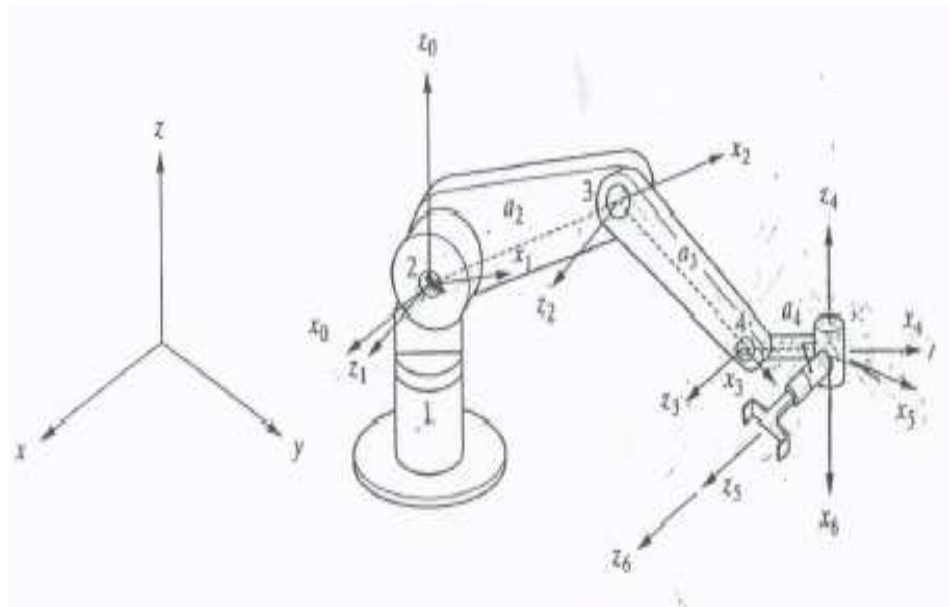
$$A_i = Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} Rot_{x, \alpha_i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

9.Daha sonra sonlandırıcı koordinat çerçevesinden temel çerçevesine dönüşüm matrisini hesapla

$$T_0^n = A_1 \dots A_n$$

4.4.1.1 Altı eklemlı robot kolun düz kinematik analizi

Bu çalışma kapsamında İnsan koluna çok yakın olan standart Mafsallı Altı eksenli Robot kol için Düz kinematik çözüm kümesine ulaşılmıştır. D-H parametre yöntemine göre eklemlere referans çerçeveleri(reference frames) atanmıştır. Referans çerçeveleri atanmış Mafsallı robot kol şekil 4.14’de görülmektedir (Niku 2001, Monacho and Canny 1992) . Bu şekilde ki modelden her bir eklemin yerel koordinatlarından önceki kısımda anlatılan D-H parametre tablosu oluşturma yöntemi kullanılarak çıkarılıyor Çizelge 4.1’de bu parametreler görülmektedir.



Şekil 4.14 Altı eksenli robot kolun yapısı

Çizelge 4.1 Altı Eksenli Robot Kolun D-H parametre Tablosu

Eklem	θ_i derece	d_i cm	a_i cm	α_i derece
1	θ_1	0	0	90
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	0	a_4	-90
5	θ_5	0	0	90
6	0	0	0	0

Birbirini izleyen her bir eklem için dönüşüm matrisleri, Çizelge 4.1'deki parametre değerlerin $A_n(n=1.....6)$ matrislerine yerleştirilmesi ile elde edilir. Örnek olarak A_1 , 0. ve 1. (küresel çerçeve, taban eklemi çerçevesi) çerçeveler arasında ki dönüşümü sağlayan matrisini temsil eder ($\sin 90 = 1$ derece; $\cos 90 = 0$ derece, $\alpha = 90$ derece, C_1 de $\cos \theta_1$ 'i temsil eder) , A_1 'den A_6 'ya bu işlemler yapılır. Her bir eklemin matris karşılığı ve toplam dönüşüm matrisi sırasıyla şekil 4.15 ve 4.16'da görülmektedir.

$$\begin{aligned}
 A_1 &= \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_4 &= \begin{bmatrix} C_4 & 0 & -S_4 & C_4 a_4 \\ S_4 & 0 & C_4 & S_4 a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_2 &= \begin{bmatrix} C_2 & -S_2 & 0 & C_2 a_2 \\ S_2 & C_2 & 0 & S_2 a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_5 &= \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_3 &= \begin{bmatrix} C_3 & -S_3 & 0 & C_3 a_3 \\ S_3 & C_3 & 0 & S_3 a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_6 &= \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Şekil 4.15 Altı eksenli robot kol için dönüşüm matrisleri

Toplam dönüşüm matrisi bu altı matrisin çarpımı ile elde edilir, yani toplam dönüşüm matrisi uç noktanın başlangıç durumuna göre değişimini sağlayan toplam sonuç matrisini verir. Buda Altı Eksenli Robot'un final düz kinematik çözümünü verir Şekil 4.16'da çözüm kümesi görülmektedir.

$${}^B T_H = A_1 A_2 A_3 A_4 A_5 A_6$$

$$= \begin{bmatrix} C_1(C_{234}C_5C_6 - S_{234}S_6) & C_1(-C_{234}C_5C_6 - S_{234}C_6) & C_1(C_{234}S_5) & C_1(C_{234}a_4 + C_{23}a_3 + C_7a_2) \\ -S_1S_5C_6 & +S_1S_5S_6 & +S_1C_5 & \\ S_1(C_{234}C_5C_6 - S_{234}S_6) & S_1(-C_{234}C_5C_6 - S_{234}C_6) & S_1(C_{234}S_5) & S_1(C_{234}a_4 + C_{23}a_3 + C_7a_2) \\ +C_1S_5C_5 & -C_1S_5S_6 & -C_1C_5 & \\ S_{234}C_5C_6 + C_{234}S_6 & -S_{234}C_5C_6 + C_{234}C_6 & S_{234}S_5 & S_{234}a_4 + S_{23}a_3 + S_7a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.16 Altı eksenli robot kol için düz kinematik çözüm

4.4.1.2 Lynx-6 robot kolun düz(FORWARD) kinematik analizi

Lynx-6 Robot kolun yapısı ve D-H parametre yöntemine referans çerçeveleri(reference frames) atanmıştır. Referans çerçeveleri atanmış Mafsallı robot kol şekil 3.3'de gösterilmektedir, bu figürde gösterilen modelden bir önceki kısımda anlatılan D-H parametre tablosu oluşturma yöntemi kullanılarak Çizelge 4.2'de gösterilen Lynx-6 robot kolu için D-H parametreleri çıkarılıyor.

Çizelge 4.2 Lynx-6 Robot Kolun D-H parametre Tablosu

Eklem	θ_i derece	d_i cm	a_i cm	α_i derece
1	θ_1	0	0	90
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	0	a_4	-90
5	θ_5	0	0	0
6	0	0	0	0

Bu çizelgeden Lynx-6 Robot kolun her bir eklemine temsil eden dönüşüm matrisleri çıkarılıyor. Birbirini izleyen her bir eklem için dönüşüm işlemi, çizelge 4.2 'de ki parametre değerlerinin $A_n(n=1.....6)$ matrislerine yerleştirilmesi ile elde edilir. Örnek olarak A_1 0 ve 1. çerçeveler arasında ilişkiyi temsil eder ($\sin 90 = 1$ derece; $\cos 90 = 0$ derece, $\alpha = 90$ derece, C_1 de $\cos@_1$ 'i temsil eder) , A_1 'den A_6 'ya bu işlemler yapılır. Her bir eklemin matris karşılığı ve toplam dönüşüm matrisi sırasıyla şekil 4.17 ve 4.18'da görülmektedir.

$$A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2 a_2 \\ S_2 & C_2 & 0 & S_2 a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3 a_3 \\ S_3 & C_3 & 0 & S_3 a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & C_4 a_4 \\ S_4 & 0 & C_4 & S_4 a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.17 Lynx-6 Robot Kol İçin Dönüşüm Matrisleri

$${}^0T_6 = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Öyle ki,

$$C_i = \cos \theta_i$$

$$S_i = \sin \theta_i$$

$$S_1 C_2 + C_1 S_2 = S_{12}$$

$$C_1 C_2 - S_1 S_2 = C_{12}$$

$$C_{234} = C_2 [C_3 * C_4 - S_3 * S_4] -$$

$$S_2 [S_3 * C_4 + C_3 * S_4]$$

$$S_{234} = S_2 [C_3 * C_4 - S_3 * S_4] +$$

$$C_2 [S_3 * C_4 + C_3 * S_4]$$

Çizelge 4.3 Lynx-6 Robot Kol için Dönüşüm Matrisleri

$n_x = C_1(C_{234} C_5 - S_{234} S_5)$	(1)	$o_x = C_1(-C_{234})S_5 - S_1 C_5$	(4)
$n_y = S_1(C_{234} C_5 - S_{234} S_5)$	(2)	$o_y = S_1(-C_{234})S_5 + C_1 C_5$	(5)
$n_z = S_{234} C_5 + C_{234} C_6$	(3)	$o_z = -S_{234} S_5$	(6)
$a_x = -C_1((C_{23} S_4 + S_{23} C_4)$	(7)	$a_y = -S_1((C_{23} S_4 + S_{23} C_4)$	(8)
$a_z = -S_4(S_{23}) + C_4(C_{23})$	(9)	$p_x = C_1(C_{234} a_4 + C_{23} a_3 + C_2 a_2)$	(10)
$p_y = S_1(C_{234} a_4 + C_{23} a_3 + C_2 a_2)$	(11)	$p_z = S_{234} a_4 + S_{23} a_3 + S_2 a_2$	(12)

4.5 Ters Kinematik Analiz

Düz kinematiklerin tersi biçimde robotikte ters kinematikler verilen sonlandırıcı konum ve yönelimi için gerekli eklem değişken değerlerini bulmak olarak tanımlanabilir. Ters kinematik problemlerinde düz kinematiklerin tersi biçimde homojen dönüşüm matrisleriyle oluşturulan doğrusal olmayan denklemlerin çözülmesi istenir. Şekil 4.12 'de verilen 6 eksenli robot kolun Şekil 4.13 'de verilen homojen matris değerlerin çarpımı sonucu oluşan toplam dönüşüm matrisi için şekil 4.14 'de gösterilen eşitlikler çözülmemelidir. Bu eşitliklerin kapalı formda doğrudan çözülmesi çok zordur, bu yüzden çözüm için etkin ve sistematik yöntemler gerektirir. Bu tip eşitliklerin çözümü için iki ana yöntem mevcuttur.

1. Kapalı Form Yaklaşımı: Kapalı form yaklaşımı homojen dönüşüm matrisinden elde edilen eşitliklere bağlı olarak genel eklem değişkenleri çözümü elde etmeyi sağlar. Böylece çok hızlı hesaplamaların gerektiği çevrimiçi robot uygulamaları için pratik ve hızlı bir yaklaşım sağlar. Ayrıca çoklu(multiple) çözümlerin olduğu durumlarda bu çözümlerden sadece birini seçerek çözüm karmaşasını ortadan kaldırır. Kapalı form yaklaşımı iki alt başlıkta incelenir.

a) Geometrik yaklaşım: Bu yaklaşım manipülatör duruşuna bağlı olarak oluşan geometrik şekilden yararlanır. Bu sebeple ters konum kinematiklerinin çözümünde tercih edilir

b) Cebirsel yaklaşım: Bu yaklaşım manipülatörün parametreleri ve eklem değişkenleri arasındaki cebirsel ilişkilerden yararlanır. Çoğunlukla ters yönelim kinematiğinin çözümünde tercih edilir.

2. Sayısal(İteratif) Yaklaşım: Bu yaklaşım manipülatörün diferansiyel kinematik eşitliklerinden ve eklem değişkenlerinin başlangıç değerlerinden yararlanarak eklem değişkenlerinin zamana sayısal değerlerini bulur.

3. Paden Kahan Alt Problemleri: Bir robotun eklem açıları verildiğinde, uç noktasını ve yönünü belirleyen probleme ileri kinematik problemi; robotun verilen bir uç nokta konfigürasyonu için eklem açılarını belirleyen problem ise ters kinematik problemi olarak ifade edilmektedir (Bayşec and Alıcı, 1983) . Robotun ileri kinematik haritası

için exponansiyellerin çarpımını kullanarak, ters kinematik problemlerin çözümüne geometrik bir algoritma geliştirmek mümkündür. Ters kinematik problem, uygun alt problemlere indirgenerek çözülebilir. Bu yöntem, ilk kez Paden (1986) tarafından sunulmuş ve Kahan'ın (1993) yılında yayınlanmamış çalışması üzerine bina edilmiştir. Aşağıda belirtilen alt problemlerin özelliği, hem geometrik olarak anlamlı olması, hem de sayısal olarak kararlı olmasıdır.

Alt problem 1: Tek eksen etrafında dönme

Σ_1 . bir eksen olsun. Verilen bir p noktasının Σ_1 eksenini etrafında θ kadar döndürülerek q noktasına ulaşmasını sağlayan ifade aşağıda verilmiştir

$$e^{\Sigma_1 \theta} p = q$$

Alt problem 2: İki sıralı eksen etrafında dönme

Σ_1 . ve Σ_2 kesişen iki eksen olsun verilen bir p noktasının önce Σ_2 eksenini etrafında θ_2 kadar ve daha sonra Σ_1 eksenini etrafında θ_1 kadar döndürülerek q noktasına ulaşmasını sağlayacak ifade aşağıdadır.

$$e^{\Sigma_1 \theta_1} e^{\Sigma_2 \theta_2} p = q$$

Alt problem 3: Verilen bir mesafeye kadar dönme

Σ . bir eksen olsun. Verilen bir p noktasının Σ eksenini etrafında, q noktasına $\&$ kadar mesafe kalana kadar döndürülmesini ifade eden denklem ise aşağıda verilmiştir.

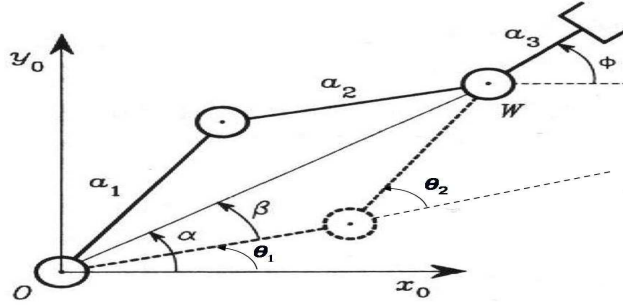
$$| q - e^{\Sigma \theta} p | = \&$$

Yukarıda verilen alt problemler, robotun en azından birkaç kesişen eklemi varsa kullanışlıdır. Ters kinematik problemlerin çözümünde, geometrik cebirin klasik eleme teorisinden yararlanılabilir. Bu yöntemle diyalektik eleme de denir Diyalektik eleme ile önce problem, sadece bir eklem açısını içeren tek değişkenli bir polinoma indirgenir. Bu polinom çözülerek eklem açılarından biri elde edilir. Bulunan açı değeri problemde yerine konular ve diğer eklem açıları bulunana kadar aynı işlem tekrarlanır. Diyalektik eleme ile hesaplanamayan ters kinematik problemler de bulunmaktadır. Günümüzde,

ters kinematik problemin çözümü için, analitik ve nümerik pek çok yöntem geliştirilmesine rağmen, genel bir yöntem yoktur. Her robot için farklı bir yöntem uygulanmaktadır.

4.5.1 Üç eklemlilik düzlemsel robot kolun ters kinematik çözümü

Şekil 4.18’de verilen üç uzuvlu düzlemsel manipülatör üzerinde geometrik ve



Şekil 4.18 Üç eklemlilik düzlemsel manipülatör

a) Geometrik yaklaşım:

b) Yukarıda ki şeklide de görüldüğü üzere $\phi = \theta_1 + \theta_2 + \theta_3$ W noktasının x ve y düzlemindeki izdüşümü değerleri figür 4.19 da gösterildiği gibi olur.

$$pw_x = a_1 c_1 + a_2 c_{12}$$

$$pw_y = a_1 s_1 + a_2 s_{12}$$

Şekil 4.19 W noktasının izdüşümü değerleri

Birinci eklem, ikinci eklem ve W noktasına merkezden çizilen doğrunun oluşturduğu üçgende kosinüs teoremini uygularsak ve $\cos(\pi - \theta_2) = -\cos \theta_2$ dönüşümünü kullanırsak şekil 4.21 ‘da görüldüğü gibi θ_2 ‘yi elde ederiz.

$$pw_x^2 + pw_y^2 = a_1^2 + a_2^2 - 2a_1 a_2 \cos(\pi - \theta_2)$$

$$c_2 = \frac{pw_x^2 + pw_y^2 - a_1^2 - a_2^2}{2a_1 a_2}$$

$$\theta_2 = A \cos(c_2)$$

Şekil 4.20 Geometrik Yaklaşımla θ_2 ‘nin çıkarılması

Burada $\theta_2 \in (-\pi, 0)$ dirsek yukarı , $\theta_2 \in (0, \pi)$ dirsek aşağı duruşuna karşılık gelmektedir. Daha sonra Şekil 4.21’de görüldüğü gibi çözüm için yerleştirilmiş α ve β değerlerinden yararlanılarak çözüm bulunur (Banka and Lin 2002).

$$\alpha = A \tan \left(\frac{pw_y}{pw_x} \right);$$

$$c_\beta \sqrt{pw_x^2 + pw_y^2} = a_1 + a_2 c_2;$$

$$\beta = A \cos \left(\frac{pw_x^2 + pw_y^2 + a_1^2 - a_2^2}{2a_1 \sqrt{pw_x^2 + pw_y^2}} \right)$$

$$\theta_1 = \alpha \pm \beta;$$

$$\theta_3 = \phi - \theta_1 - \theta_2;$$

Şekil 4.21 Geometrik Yaklaşım ile θ_1 ve θ_3 ’nin çıkarılması

4.5.2 Altı eklemlili düzlemsel robot kolun ters kinematik çözümü

$${}^R T_H = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1 A_2 A_3 A_4 A_5 A_6$$

$$= \begin{bmatrix} C_1(C_{23}C_3C_6 - S_{23}S_6) & C_1(-C_{23}C_3C_6 - S_{23}S_6) & C_1(C_{23}S_3) & C_1(C_{23}a_4 + C_{23}d_3 + C_2d_2) \\ -S_1S_3C_6 & S_1S_3C_6 & S_1C_3 & S_1(C_{23}a_4 + C_{23}d_3 + C_2d_2) \\ S_1(C_{23}C_3C_6 - S_{23}S_6) & S_1(-C_{23}C_3C_6 - S_{23}S_6) & S_1(C_{23}S_3) & S_1(C_{23}a_4 + C_{23}d_3 + C_2d_2) \\ + C_1S_3C_6 & -C_1S_3C_6 & -C_1C_3 & -C_1(C_{23}a_4 + C_{23}d_3 + C_2d_2) \\ S_{23}C_3C_6 + C_{23}S_6 & -S_{23}C_3C_6 + C_{23}S_6 & S_{23}S_3 & S_{23}a_4 + S_{23}d_3 + S_2d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.22 Kinematik çözüm ve sembol matrisi

Yukarıda ki şekil 4.22’te, hedeflenen robotun sonuç pozisyon ve yönelim matrisini ve bu matrisi temsil eden sembol matrisi görülmektedir. Bu denklemleri çözmek ve her bir eklemin temsil ettiği açı değerlerini bulmak için eşitliğin her iki yanında ki matrisler sırasıyla A_n^{-1} ile önden (premultiply) çarpılıp , (n=1.2.3...6) yeni elde edilen matrisler çözümlenerek sonuç değerleri sırasıyla elde edilir. Şekil 4.23’de iki matrisin önden A_1^{-1} çarpılması ve sonuçta oluşan matris eşitliği görülüyor

$$A_1^{-1} * \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1} * A_1 * A_2 * A_3 * A_4 * A_5 * A_6$$

Şekil 4.23 Altı eksenli için ters kinematik ilk adım

Yukarıda ki şekilde ki denklemin, eşitliklerinden [3,4] üncü elaman dikkate alınırsa şekil 4.24 de görüldüğü gibi θ_1 elde edilir.

$$p_x S_1 - p_y C_1 = 0 \rightarrow \theta_1 = \arctan(p_y/p_x) \text{ or } \theta_1 = \theta_1 + 180^\circ$$

Şekil 4.24 θ_1 açısının çıkarılma işlemi

Buna karşın aynı matrislerin [1,4] ve [2,4] elemanları dikkate alınıp eşitliklerin kareleri toplanırsa $\cos(\theta_3)$ elde edilir işlem adımları şekil 4.25’de görülmektedir.

$$p_x C_1 + p_y S_1 = C_{234} a_4 + C_{23} a_3 + C_2 a_2$$

$$p_z = S_{234} a_4 + S_{23} a_3 + S_2 a_2$$

$$(p_x C_1 + p_y S_1 - C_{234} a_4)^2 = (C_{23} a_3 + C_2 a_2)^2$$

$$(p_z - S_{234} a_4)^2 = (S_{23} a_3 + S_2 a_2)^2$$

$$C_3 = ((p_x C_1 + p_y S_1 - C_{234} a_4)^2 + (p_z - S_{234} a_4)^2 - a_2^2 a_3^2) / 2 a_2 a_3$$

Şekil 4.25 Altı Eksenli için $\cos(\theta_3)$ ifadesinin çıkarılma işlemleri.

$\sin^2 + \cos^2 x = 1$ olduğundan, şekil 4.26 ‘de gösterildiği gibi θ_3 açı değeri elde edilir.

$$S_3 = -\sqrt{1 - C_3^2}$$

$$\Theta_3 = \arctan\left(\frac{S_3}{C_3}\right)$$

Şekil 4.26 θ_3 açısının çıkarılma işlemleri

2,3 ve 4. eklemler paralel olduğundan sonraki adım A_1 'den A_4 'e ters matrislerle final matrisi çarpmaktır. Şekil 4.27'de bu adımlar sonucunda oluşan çözüm matrisi ve sembol matrisleri görülmektedir.

$$A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}[\text{RHS}] = A_5A_6$$

$$= \begin{bmatrix} C_{23}(C_1n_x + S_1n_y) + S_{23}n_z & C_{23}(C_1o_x + S_1o_y) + S_{23}o_z & C_{23}(C_1a_x + S_1a_y) + S_{23}a_z & C_{23}(C_1p_x + S_1p_y) + S_{23}p_z - C_4a_1 - C_4a_2 - a_4 \\ C_1n_y - S_1n_x & C_1o_y - S_1o_x & C_1a_y - S_1a_x & 0 \\ -S_{23}(C_1n_x + S_1n_y) + C_{23}n_z & -S_{23}(C_1o_x + S_1o_y) + C_{23}o_z & -S_{23}(C_1a_x + S_1a_y) + C_{23}a_z & -S_{23}(C_1p_x + S_1p_y) + C_{23}p_z + S_4a_1 + S_4a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_3C_6 & -C_3S_6 & S_3 & 0 \\ S_3C_6 & -S_3S_6 & -C_3 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.27 θ_{234} açısının çıkarılma işlemlerinin ilk aşaması

Yukarıda ki şekilde de görüldüğü gibi İki matrisin eşitliğinden (3,3) Elemanlardan şekil 4.28'de gösterilen yöntemle θ_{234} elde edilir.

$$S_{234}(C_{1ax} + S_{1ay}) + C_{234}a_z = 0$$

$$\Theta_{234} = \arctan\left(\frac{a_z}{C_{1ax} + S_{1ay}}\right)$$

$$\Theta_{234} = \Theta_{234} + 180$$

Şekil 4.28 θ_{234} açısının çıkarılma işlemleri

Şekil 4.23 de gösterilen eşitlik kullanılarak $\sin(\theta_2)$ ve $\cos(\theta_2)$ elde edilir, işlem sırası ve aşamaları Şekil 4.29'da gösterilmektedir.

$$\begin{cases} p_x C_1 + p_y S_1 = C_{234}a_4 + C_{23}a_3 + C_2a_2, \\ p_z = S_{234}a_4 + S_{23}a_3 + S_2a_2. \end{cases}$$

$$\begin{cases} p_x C_1 + p_y S_1 - C_{234}a_4 = (C_2C_3 - S_2S_3)a_3 + C_2a_2, \\ p_z - S_{234}a_4 = (S_2C_3 + C_2S_3)a_3 + S_2a_2. \end{cases}$$

$$\begin{cases} S_2 = \frac{(C_3a_3 + a_2)(p_x - S_{234}a_4) - S_3a_3(p_x C_1 + p_y S_1 - C_{234}a_4)}{(C_3a_3 + a_2)^2 + S_3^2a_3^2}, \\ C_2 = \frac{(C_3a_3 + a_2)(p_x C_1 + p_y S_1 - C_{234}a_4) + S_3a_3(p_z - S_{234}a_4)}{(C_3a_3 + a_2)^2 + S_3^2a_3^2}. \end{cases}$$

Şekil 4.29 $\sin(\theta_2)$ ve $\cos(\theta_2)$ çıkarılma işlemleri

$\sin(\theta_2)$ ve $\cos(\theta_2)$ ifadeleri kullanılarak $\tan^{-1}(\sin(\theta_2) / \cos(\theta_2))$ ifadesi ile θ_2 çekilir θ_2 , θ_3 ve θ_{234} bilindiğinden $\theta_4 = \theta_{234} - \theta_2 - \theta_3$ işlemiyle elde edilir,

Şekil 4.27'de ki eşitlikleri kullanarak θ_5 açısı şekil 4.30'da gösterildiği şekilde elde edilir.

$$\theta_5 = \tan^{-1} \frac{C_{234}(C_1a_x + S_1a_y) + S_{234}a_z}{S_1a_x - C_1a_y}$$

$$\begin{cases} S_5 = C_{234}(C_1a_x + S_1a_y) + S_{234}a_z, \\ C_5 = -C_1a_y + S_1a_x. \end{cases}$$

Şekil 4.30 θ_5 açısının elde edilmesi.

Şekil 4.27’de gösterilen eşitlik ifadeleri A_5^{-1} ile çarpılırsa, elde edilen yeni matris eşitliklerinin (2,1) ve (2,9) indisli elemanları kullanılarak θ_6 açısı elde edilir. Aşağıda ki şekilde bu aşamalar gösterilmektedir.

$$\begin{bmatrix} C_5[C_{234}(C_1n_x + S_1n_y) + S_{234}n_z] & C_5[C_{234}(C_1o_x + S_1o_y) + S_{234}o_z] & 0 & 0 \\ -S_5(S_1n_x - C_1n_y) & -S_5(S_1o_x - C_1o_y) & 0 & 0 \\ -S_{234}(C_1n_x + S_1n_y) + C_{234}n_z & -S_{234}(C_1o_x + S_1o_y) + C_{234}o_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\theta_6 = \tan^{-1} \frac{-S_{234}(C_1n_x + S_1n_y) + C_{234}n_z}{-S_{234}(C_1o_x + S_1o_y) + C_{234}o_z}$$

Şekil 4.31 θ_6 açısının elde edilmesi.

Altı eksenli standart bir robot kolun ters kinematik çözümleri için, yapılması gereken adımlar yukarıda anlatılmıştır, her bir açı için hesaplanan denklemler aşağıda şekil 4.32’de gösterilmektedir. Her bir uygulama için yukarıda ki hesaplamaları yaptıktan sonra aşağıda hesaplanan denklemlerin kullanılması zaman ve performanstan kazandıracaktır (Craig 2005).

$$\begin{aligned}
\theta_1 &= \tan^{-1}\left(\frac{p_y}{p_x}\right) \quad \text{and} \quad \theta_1 = \theta_1 + 180^\circ, \\
\theta_{234} &= \tan^{-1}\left(\frac{a_z}{C_1a_x + S_1a_y}\right) \quad \text{and} \quad \theta_{234} = \theta_{234} + 180^\circ, \\
C_3 &= \frac{(p_xC_1 + p_yS_1 - C_{234}a_4)^2 + (p_z - S_{234}a_4)^2 - a_2^2 - a_3^2}{2a_2a_3}, \\
S_3 &= \pm \sqrt{1 - C_3^2}, \\
\theta_3 &= \tan^{-1}\frac{S_3}{C_3}, \\
\theta_2 &= \tan^{-1}\frac{(C_3a_3 + a_2)(p_z - S_{234}a_4) - S_3a_3(p_xC_1 + p_yS_1 - C_{234}a_4)}{(C_3a_3 + a_2)(p_xC_1 + p_yS_1 - C_{234}a_4) + S_3a_3(p_z - S_{234}a_4)}, \\
\theta_4 &= \theta_{234} - \theta_2 - \theta_3, \\
\theta_5 &= \tan^{-1}\frac{C_{234}(C_1a_x + S_1a_y) + S_{234}a_z}{S_1a_x - C_1a_y}, \\
\theta_6 &= \tan^{-1}\frac{-S_{234}(C_1n_x + S_1n_y) + C_{234}n_z}{-S_{234}(C_1o_x + S_1o_y) + C_{234}o_z},
\end{aligned}$$

Şekil 4.32 Altı Eksenli Robot Kolun Ters kinematik Çözümleri

4.5.3 LYNX-6 düzlemsel robot kolun ters kinematik çözümü

Özgürlük derecesinin Üç'ten ($DOF > 3$) değerinden büyük olduğu uygulamalarda birden çok çözüm kümesinin ortaya çıkması kaçınılmazdır. Yukarıda geliştirilen ters kinematik çözümler ise Beş veya Altı Özgürlük derecesine sahip olan robot kollar olduğu için tek(unique) çözüm kümesine ulaşılmaz. Bu yüzden çözüm kümesinin indirgenmesi için değişik tekniklerin kullanılması veya kısıtlamaların gerçekleştirilmesi gerekmektedir. Bu çalışma esnasında, Geliştirilen ters kinematik çözümler GAY'in (2007) yılında Üç serbestlik derecesine sahip Lynx robot kollar için uyguladığı, Paden (1986) ve Kahan (1983) 'den esinlenerek ortaya çıkardığı ters kinematik çözümlerde kullandığı kısıtlama yöntemi kullanılmıştır. Kısıtlama parametresi olarak WARTG (Dirsek + Omuz + Bilek açısı değerlerinin) yani son noktanın başlangıç noktasına göre açısı önceden parametre olarak pozisyon değerleri ile beraber sisteme giriş olarak verilemesi yöntemi ile genişletilmiş ve uygulanan bu kısıtlama yöntemi ile ters kinematik denklemler için tek

(unique) çözüm kümesine ulaşılmıştır. Bu yüzden θ_{234} (WARTG) değeri önceden giriş verisi olarak hem Altı Mafsallı robot kollar için hem de aşağıda ters kinematik çözüm yöntemi gösterilen Beş Özgürlük Dereceli (DOF) Lynx-6 Robot Kollar için kısıtlama yöntemi olarak uygulama kısmında bu tez çalışması kapsamında kullanılmıştır.

$${}^0T_6 = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.33 Kinematik çözüm ve sembol matrisi

Yukarıda ki şekilde, hedeflenen robotun sonuç pozisyon ve yönelim matrisini ve bu matrisi temsil eden sembol matrisi görülmektedir (bkz 4.18) . Bu denklemleri çözmek ve her bir eklemin temsil ettiği açı değerlerini bulmak için eşitliğin her iki yanında ki matrisler sırasıyla A_n^{-1} ile önden(premultiply) çarpılıp , (n=1.2.3...6) yeni elde edilen matrisler çözümlenerek sonuç değerleri sırasıyla elde edilir. Şekil 4.35’de iki matrisin önden A_1^{-1} çarpılması ve sonuçta oluşan matris eşitliği görülmüyor.

$$A_1^{-1} * \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1} * A_1 * A_2 * A_3 * A_4 * A_5 * A_6$$

$$\begin{bmatrix} n_x C_1 + n_y S_1 & o_x C_1 + o_y S_1 & a_x C_1 + a_y S_1 & p_x C_1 + p_y S_1 \\ n_z & o_z & a_z & p_z \\ n_x S_1 - n_y C_1 & o_x S_1 - o_y C_1 & a_x S_1 - a_y C_1 & p_x S_1 - p_y C_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{234} C_5 & -C_{234} C_5 & C_{234} & C_{234} a_4 + C_{23} a_3 + C_2 a_2 \\ S_{234} C_5 & -S_{234} C_5 & S_{234} & S_{234} a_4 + S_{23} a_3 + S_2 a_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.34 Kinematik çözüm ve sembol matris eşitliği

Şekil 4.34’de görülen matris eşitliğinden ; $p_x S_1 - p_y C_1 = 0$ sonucunu elde edilmektedir buradan, θ_1 açısı aşağıda gösterildiği şekilde çıkarılır.

$$\theta_1 = \arctan (p_y/p_x) \text{ or } \theta_1 = \theta_1 + 180^\circ \quad (1)$$

Şekil 4.34’te görünen Matris eşitliği yeniden ele alınıp, [1,4] ve [2,4] ‘deki elemanların kareleri alınıp toplanırlarsa C_3 elde edilir. Bu işlem adımları aşağı da gösterilmektedir (Niku 2001) , θ_{234} (WARTG) değeri önceden sisteme veriliyor (GAY 2007) .

$$\begin{aligned} (p_x C_1 + p_y S_1 - C_{234} a_4)^2 &= (C_{23} a_3 + C_2 a_2)^2 \\ (p_z - S_{234} a_4)^2 &= (S_{23} a_3 + S_2 a_2)^2 \end{aligned} \quad (2)$$

$$\begin{aligned} &(p_x C_1 + p_y S_1 - C_{234} a_4)^2 + (p_z - S_{234} a_4)^2 \\ &= a_2^2 + a_3^2 + 2 * a_2 * a_3 \end{aligned} \quad (3)$$

$$\begin{aligned} C_3 &= (p_x C_1 + p_y S_1 - C_{234} a_4)^2 + (p_z - S_{234} a_4)^2 \\ &- a_2^2 - a_3^2 / 2 a_2 a_3 \end{aligned} \quad (4)$$

2 ve 3 ‘ü referans olarak alınırsa,

$$C_{12} = C_1 C_2 - S_1 S_2 \text{ and } S_{12} = S_1 C_2 + C_1 S_2$$

$$S_2 = (C_3 a_3 + a_2) (p_z - S_{234} a_4) - S_3 a_3 (p_x C_1 + p_y S_1 - C_{234} a_4) / (C_3 a_3 + a_2)^2 + S_3^2 a_3^2 \quad (5)$$

$$C_2 = (C_3 a_3 + a_2) (p_x C_1 + p_y S_1 - C_{234} a_4) + S_3 a_3 (p_z - S_{234} a_4) / (C_3 a_3 + a_2)^2 + S_3^2 a_3^2 \quad (6)$$

$$\theta_2 = \arctan (S_2/C_2) \quad (7)$$

$$\text{ve } \theta_4 \rightarrow \theta_4 = \theta_{234} - \theta_3 - \theta_4 \quad (8)$$

$$\begin{aligned} A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} * \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} &= A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} * T_0^6 \\ &= A_5 * A_6 \end{aligned}$$

θ_5 açısının çıkarılması için A_n^{-1} ($n=1 \dots 4$) ile çarpma işlemi

Şekil 4.34'teki sonuç matrisinin [2,1] ve [2,2] elemanları (Niku 2001).

Bize ,

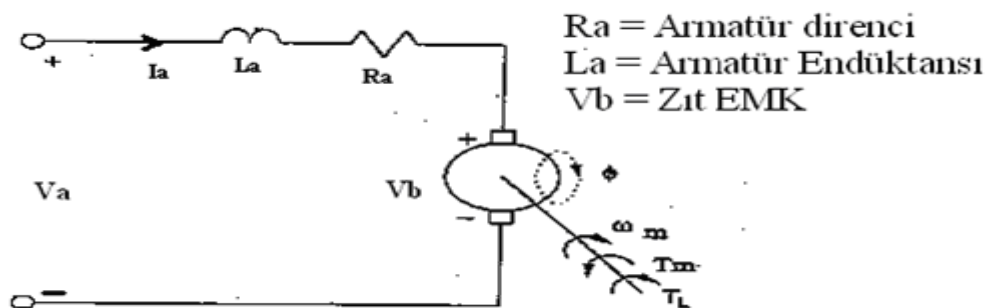
$C_5 = (C_1 n_y - S_1 n_x)$, $S_5 = (C_1 o_y - S_1 o_x)$ değerlerini verir,

Sonuç olarak,

$$\theta_5 = \arctan(S_5 / C_5) \quad (9)$$

5 DOĞRU AKIM MOTORLAR

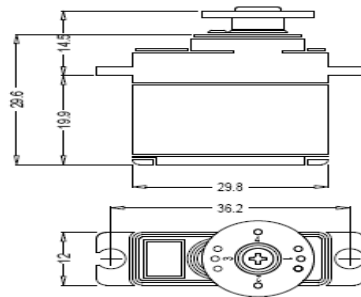
Doğru Akım (DC) Motorlar robotik uygulamalarında ve endüstriyel alanda sıklıkla kullanılmaktadırlar. Ucuz ve küçüktürler. Genel olarak 1.5V ile 100V arasında çalışabilirler. 6V, 12V ve 24V motorlar çok yaygın olarak bulunmaktadır. Birkaç bin RPM den on binlerce RPM e kadar çalıştırılabilirler. 12V ve daha küçük motorlar yapısına göre birkaç yüz mili amperden birkaç mili ampere kadar akım çekebilirler. DC motorların genel özellikleri, Yüksek hız, Düşük tork, Ters yönde kullanım ve Sürekli hareket olarak sıralanabilir. Mantık olarak bobin üzerinden geçen akımın sonucunda oluşturduğu manyetik alanlar sayesinde oluşturduğu kutuplaşmayı ileri ve geri yönlü olarak kullanarak yani zıt kutupların çekmesi ya da aynı kutupların birbirini itmesi prensibinin dairesel harekete dönüştürülmesini temel alan en basit yapıdır. Endüvi dönerken üzerindeki iletkenler de manyetik alan içerisinde döndüklerinden bu iletkenlerde bir endüksiyon elektromotor kuvveti indüklenir. Doğru akım makinesi kullanım amacına göre dinamo ya da motor olarak çalıştırılabilir. Bu formlardan birisinde çalışma, makinede herhangi bir değişikliği gerektirmez. Eğer makine dinamo olarak çalıştırılırsa tork yön değiştirir. Manyetik alan içinde etkin uzunluğu "L" ve içerisinde geçen akımı "i" olan bir iletken akı yoğunluğu B olan bir alan içerisinde kalırsa, iletken manyetik alan tarafından itilir. İletkenin alana dik olma durumunda meydana gelen itme kuvvetinin büyüklüğü "Newton" olarak $\mathbf{F}=\mathbf{B.i.L}$ olur. Alan tarafından iletken üzerinde oluşturulan itme kuvvetinin yönü iletkenin taşıdığı akımın yönüne bağlıdır. İletkende itme kuvveti olduğu sürece iletkende bir hareket veya dönme olayı meydana gelir. Doğru Akım Motorunun eşdeğer devresi Şekil 5.1'de görülmektedir.



5.1 Doğru akım motoru eşdeğer devresi

5.1 Doğru Akım Servo Motorlara Giriş

Doğru Akım servo motorları, genel olarak bir Doğru akım motoru olup, motora gerekli olan Doğru Akımı dört farklı metottan elde edilir, Bunlar sırasıyla, Bir elektrik yükselteçten; Alternatif Akımın doyumlu reaktörden geçirilmesinden; Alternatif akımın tristörden geçirilmesinden ve Amplidin, retotrol, regüleks gibi dönel yükseltgerlerin kullanılmasıdır. Doğru Akım servo motorlar çok küçük güçlerden çok büyük güçlere kadar imal edilirler (0.05 Hp den 1000 Hp ye kadar). Bu motorlar klasik Doğru Akım motorlar gibi imal edilirler. Bu motorlar küçük yapılıdır ve endüvileri (yükseklik. uzunluk / Çap oranıyla) kutup atalet momentini en küçük yapacak şekilde tasarlanırlar. Küçük çaplı ve genellikle içerisinde kompanzasyon sargısı olan kuvvetli manyetik alanı boyu uzun Doğru Akım motorların servo motor denir. Küçük çaplı ve genellikle içerisinde kompanzasyon sargısı olan, kuvvetli manyetik alanı boyu uzun doğru akım motorlarına da servo motor denir. Doğru Akım servo motor çalışma prensibi açısından aslında, Statoru Daimi Mıknatıs bir Doğru Akım motordur. Daimi mıknatıslar tarafından meydana getirilen manyetik alan ile içinden akım geçirilen iletkenler arasındaki etkileşim nedeniyle bir döndürme momenti meydana gelir. Bu döndürme momenti manyetik alan vektörü ile sargı akım vektörü arasındaki açı 90° olduğunda maksimum değeri alır. Bir Doğru Akım servo motorda fırçaların konumları, her iki dönüş yönü için de döndürme momenti açısının 90° olmasını sağlayacak şekilde belirlenmiştir. Kolektör segmentlerinin fazla olması neticesinde momentin sıfır bir noktada rotorun hareketsiz kalması engellenmiş olur. Şekil 5.2'de Lynx-6 'da kullanılan servo motorlardan Hs-81 görülmektedir.



Şekil 5.2 Hs-81 servo motoru ve modeli

5.1.1 Doğru akım motorlarının temel denklemleri

Doğru Akım (DA) Motorları başlı başına bir çalışma konusudur; Bu tez çalışması kapsamına incelenecek kısım simülasyon yazılımlarında kullanılacak olan yöntemlerin anlaşılması için gereken altyapı bilgisini içerir. Doğru Akım Motorlarında Üç temel komponent bulunmaktadır. Bunlar Armatür, Stator ve Fırçalardır. DA motorlarında dönen parçaya armatür denir. Armatür; tel sargıları, lameller, fırça, komutator ve şafttan oluşmaktadır. Lameller Eddy akımlarının yayılmasını ve akı kayıplarını önlerler, böylece ısınma engellenmiş olur. Tel sargılar manyetik alanı oluştururlar. Her tel sargı grubunun iki besleme ucu (pozitif ve negatif gerilim için) vardır. Bu uçlar, armatürün arka tarafında komutator adı verilen bir terminalde toplanırlar. Komutator, statordeki fırçalarda oluşan gerilimin armatür üzerinden geçebilmesini sağlayan parçadır. Ayrıca, armatür ve statorun kutupları birbirlerine zıt olduğu anda (kilitleme), armatürün polaritesini değiştirerek dönme hareketinin devamlılığını sağlar. Yüksek iletkenlik özelliklerinden ötürü komutatorler bakırdan üretilirler.

Her komutator segmenti arasında yalıtım elemanı bulunmalıdır. Aksi takdirde topraklamalar veya kısa devreler oluşacak, motor aşırı ısınacak ve çalışamaz hale gelecektir. Şaft, armatürün arka tarafında statore yataklanmıştır. Stator Motorda sabit duran parçadır. Gövdeyi ve sabit manyetik alanı oluşturur. Manyetik alanın oluşması için tel sargılar veya mıknatıs kullanılır. Mıknatıslı DA motorları düşük güçte çalışırlar. Stator sargısı, yalıtılmış uzun bir telin statorda lameller içindeki yuvalara sarılmasıyla oluşturulur. Bu yuvalara alan kutupları (field poles) adı verilir. Sargılar nem, toz ve diğer etkilere karşı koruyucu bir vernikte kaplanır. DA motorunun arka kapağında, komütatör sayısı kadar fırça bulunmaktadır. Fırçanın görevi, akımın komütatörler üzerinden armatür üzerinden kaynağa dönmesini sağlamaktır.. Armatürün dönmesi sırasında her komütatör sırasıyla bu fırçalara değecek ve armatürün polaritesi böylece değişecektir. Fırça motorun en çabuk aşınan kısmıdır ve periyodik bakım gerektirir (Çoşkun 2004). Doğru Akım Motorlarını modellerken ve simülasyon yazılımları gerçekleştirilirken iki tür denklem türü üzerine çalışılmıştır. Bunlar sırasıyla Elektriksel ve Mekanik Denklemlerdir.

5.1.1.1 Elektrik denklemleri

Stator kısmının matematiksel denklemi (Şekil 5.3) , 1’de görülmektedir.

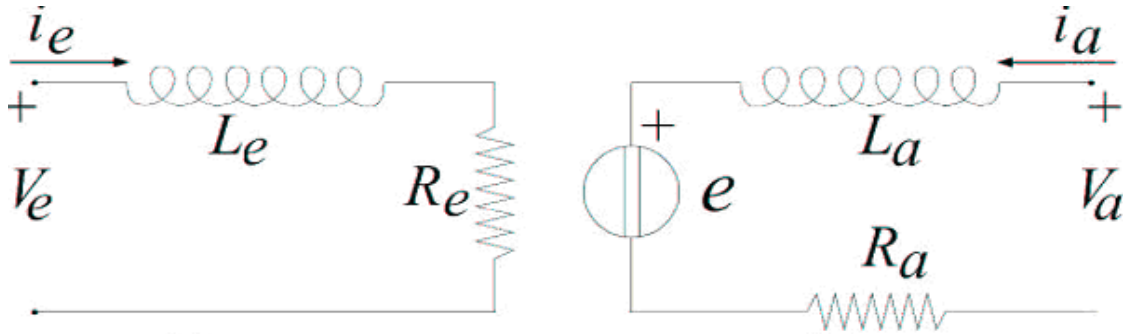
$$v_e(t) = L_e \frac{di_e}{dt} + R_e i_e. \quad (1)$$

Yukarıda ki denklemi Laplace alanına çevirirsek sinyal 2 deki şekle dönüşür.

$$\frac{i_e(s)}{v_e(s)} = \frac{K_e}{1 + \tau_e s}, \quad (2)$$

Öyle ki,

$$K_e := \frac{1}{R_e} \rightarrow \text{Stator Kazancı} \quad \&\& \quad \tau_e := \frac{L_e}{R_e} \rightarrow \text{Stator Zaman sabiti}$$



Şekil 5.3 Doğru akım elektriksel eşitlik denklem şeması

Rotar kısmının matematiksel denklemi (Şekil 5.3) 3 ’de görülmektedir.

$$v_a(t) = L_a \frac{di_a}{dt} + R_a i_a + e \quad (3)$$

Yukarıda ki denklemi Laplace alanına çevirirsek sinyal 4 deki şekle dönüşür.

$$\frac{i_a(s)}{v_a(s) - e(s)} = \frac{K_a}{1 + \tau_a s}, \quad (4)$$

Öyle ki ,

$$K_a := \frac{1}{R_a} \rightarrow \text{rotor(armatür) kazancı} \quad \&\& \quad \tau_a := \frac{L_a}{R_a} \rightarrow \text{rotor zaman sabiti}$$

(5)

Bununla beraber Motordan elde edilen Tork (Torque) T_M ile Ters Elektriksel Manyetik Güç (back EMF) e ile, denklemleri sırasıyla 6 ‘da gösterilmektedir.

$$\begin{aligned} T_M &= K i_e i_a, \\ e &= K i_e \omega, \end{aligned} \quad (6)$$

5.1.1.2 Mekanik denklemler

Motor üzerinde ki stator ve rotor kısımlarına gerilim uygulandığında motor tork oluşturur Armatür’ün oluşturduğu J eylemsizlik (inertia) ve Sürtünme Katsayısı (F) . Ayrıca motorun sahip olduğu başlangıç torku T_L ile beraber 7 deki denklem elde edilir.

$$T_M - T_L = J \frac{d\omega}{dt} + F \omega. \quad (7)$$

Yukarıdaki denklem elektrik denklemlerinde olduğu gibi doğrusal bir dönüşüm fonksiyonu ile ilişkilendirilirse 8’ deki denklem elde edilir.

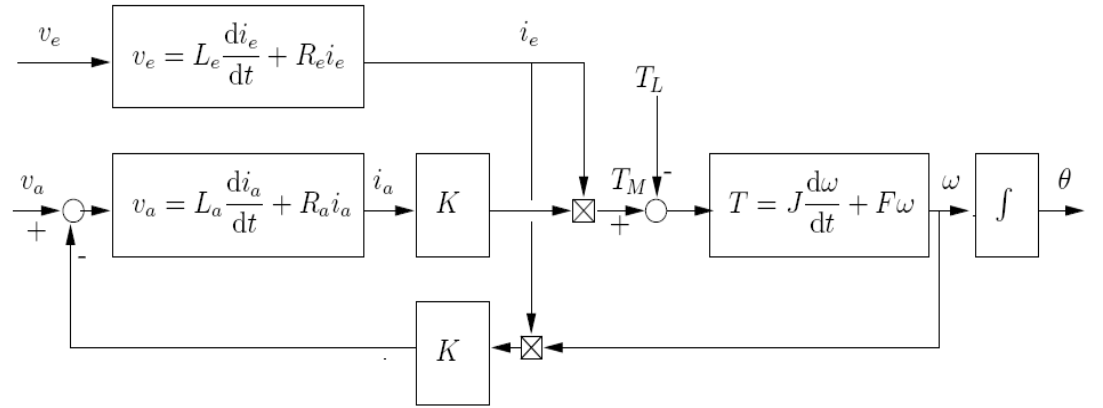
$$\frac{\omega(s)}{T_M(s) - T_L(s)} = \frac{K_m}{1 + \tau_m s}, \quad (8)$$

Öyle ki,

$$K_m := \frac{1}{F} \rightarrow \text{Mekanik kazanç} \quad \&\& \quad \tau_m := \frac{J}{F} \rightarrow \text{Mekanik zaman kazancı}$$

5.2 Dinamik Model ve Kontrol Tipleri

Aşağıda şekilde bir standart doğru akım motorunun dinamik modeli görülmektedir.

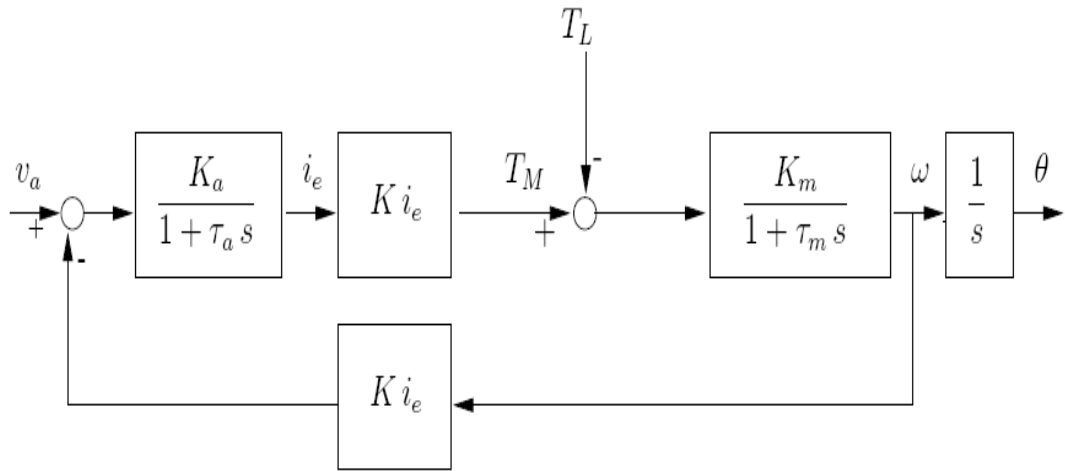


Şekil 5.4 Doğru akım motoru blok diyagramı

Şekil 5.4’de görülen doğrusal olmayan (nonlinear) blok diyagramı basitleştirmek ve doğrusallaştırmak için üç ana kontrol tekniği kullanılmaktadır.

- Sabit armatör akımı ile Stator voltaj kontrolü
- Sabit armatör voltajı ile Stator voltaj kontrolü
- Armatör Akım Kontrolü

Bu tez çalışma kapsamında Doğru Akım motorlarını kontrol etmek için Armatür Akım Kontrolü yöntemi kullanılmıştır. Armatür Akım Kontrol yöntemi en yaygın kontrol metodudur. Bu metotta stator voltajı sabit kabul edilir ve sistem sadece armatör tarafından kontrol edilir. Şekil 5.5’de Doğru Akım motorunun Armatür Akım Kontrol metodu kullanılarak doğrusallaştırılmasının blok diyagramı görülmektedir.



Şekil 5.5 Doğru akım motoru ve armatör akım kontrolü

Sistemin transfer matrisi aşağıda görülmektedir.

$$\omega(s) = [W_1(s) \ W_2(s)] \begin{bmatrix} v_a(s) \\ T_L(s) \end{bmatrix},$$

$$W_1(s) := \frac{K_a K_\Phi \Phi K_m}{(1 + \tau_a s)(1 + \tau_m s) + K_a K_m (K_\Phi \Phi)^2},$$

$$W_2(s) := \frac{K_m (1 + \tau_a s)}{(1 + \tau_a s)(1 + \tau_m s) + K_a K_m (K_\Phi \Phi)^2}$$

Genel olarak Doğru Akım Motorunun dinamik denklemleri (9...12) de gösterilmiştir.

$$L_a(di_a/dt) = -R_a i_a + e_a - e_b \quad (9)$$

$$d\theta /dt = \omega \quad (10)$$

$$e_b = K_b \omega \quad (11)$$

$$T_m = K_i i_a \quad (12)$$

Öyle ki ,

L_a = Armatür endüktansı ;

i_a = Armatür akımı (amp)

R_a = Armatür direnci (ohm)

e_a = Armatür voltajı (V)

e_b = Geri besleme voltajı (V)

K_b = Geri besleme voltaj katsayısı (sn/rad)

K_i = Akım-dönme momenti katsayısı= (N.m/amp)

T_m = Motorun ürettiği dönme momenti (N.m)

θ = Motor milinin açısal yer değiştirmesi (rad)

ω = Motor milinin açısal hızı (rad/sn) (Caner 2004)

6. UYGULAMALAR

Bu kısımda Visual Studio .Net platformunda C# programlama dili kullanılarak tez çalışması kapsamında gerçekleştirilen dört farklı görsel tabanlı yazılım programı anlatılmıştır. Birinci ve ikinci yazılımda çalışmada kullanılan (Lynx-6) robot kolun fiziksel olarak kontrol edilip sırasıyla gerçek zamanlı ve akıllı uygulamalar için kullanılması sağlanmıştır. Lynx Robot Kollarını model alan iki ayrı simülasyon uygulama yazılımı da üçüncü ve dördüncü yazılımlar kapsamında gerçekleştirilmiştir. Üçüncü yazılım Lynx robot kollar için kinematik çözüm paketi içeren bir matematiksel simülasyon yazılım uygulamasıdır. Buna karşın Dördüncü simülasyon yazılımı, robot eklemlerini kontrol eden Doğru Akım (DC) motorlarının kontrolünü, dinamik verilerin, hareketsel karakteristiğe çevrilmesi sayısal veriler grafikler ve simulator yardımıyla gösterilmesi için tasarlanmış dinamik ve kinematik analizin ortak olarak kullanılabildiği bir uygulama paketidir.

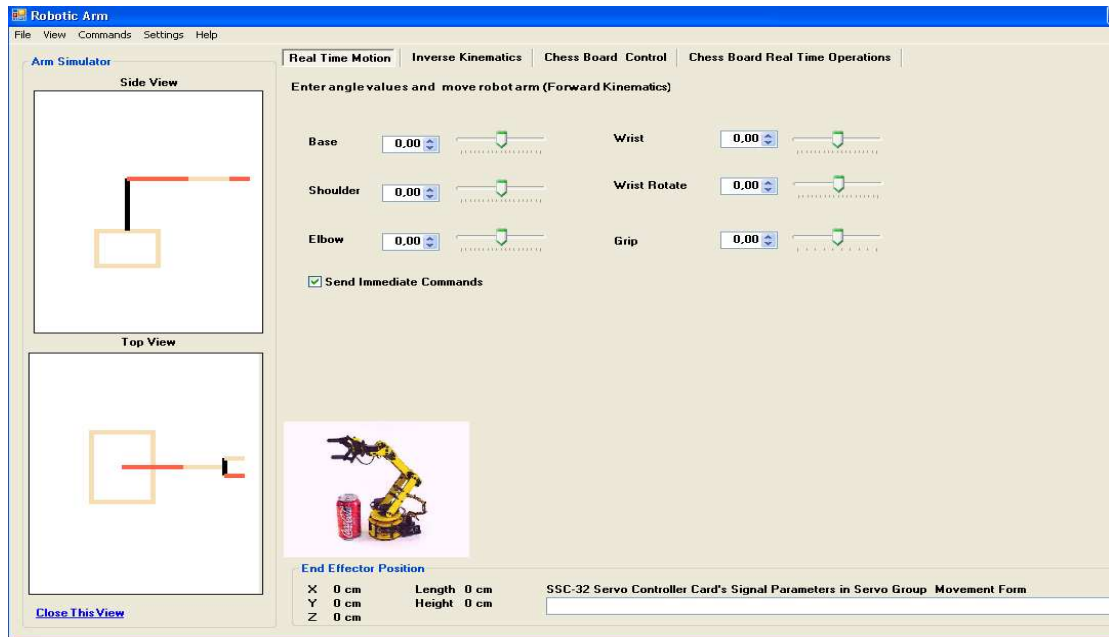
6.1 Yame (Gerçek Zamanlı Uygulama)

Yame Lynx-6 Robot Kolun kinematik çözümlerini içeren, gerçek zamanlı uygulamalarda kullanılmak için tasarlanmış. Bu tez kapsamında geliştirilmiş ücretsiz bir yazılım paketidir. Robot kolun servo motorlarını ve uygulamalarda kullanılacak elektromıknatısı servo kontrol kartı üzerinden yönetirken, kinematik analizin gerçek zamanlı uygulamada kullanılmasını sağlamaktadır. Bu yazılım paketi kapsamında Robot kolun eklemlerinin kinematik çözümlerinin uygulamada kullanılması için ilgili servo motorun veya mıknatısın aktif hale getirilip istenilen pozisyona ulaşması ve belirlenen hedefi(alma/bırakma) işlemlerinin gerçekleştirmesi sağlanmıştır. Ayrıca robot eklem limitlerinin ayarlanması için yazılımda ayrı bir alt ekran oluşturulmuş ve limitlerin ölçülmesi sağlanmıştır. Bu çalışmada ana hedef kinematik çözümlerin uygulamada kullanılmasını sağlamaktır. Bu uygulama kapsamında bir satranç tahtasında, robot kolun gerçek zamanlı olarak belirtilen konumlara ulaşması ve elin ucuna bağlanan elektromıknatıs yardımı ile o noktalara ulaşip taş bırakıp, alma işlemlerini gerçekleştirmesi sağlanmıştır. Sonuç olarak geliştirilen bu yazılım paketi Düz ve Ters Kinematik çözümler yönelme ve pozisyon bulma yetenekleri yanında, satranç tahtasında istenilen kareye gidip oradan taş alıp herhangi başka bir kareye bırakma özelliklerini de içerir. Robot Kolun hareket yeteneklerini satranç tahtasında test edilmektedir. Altı servo motor

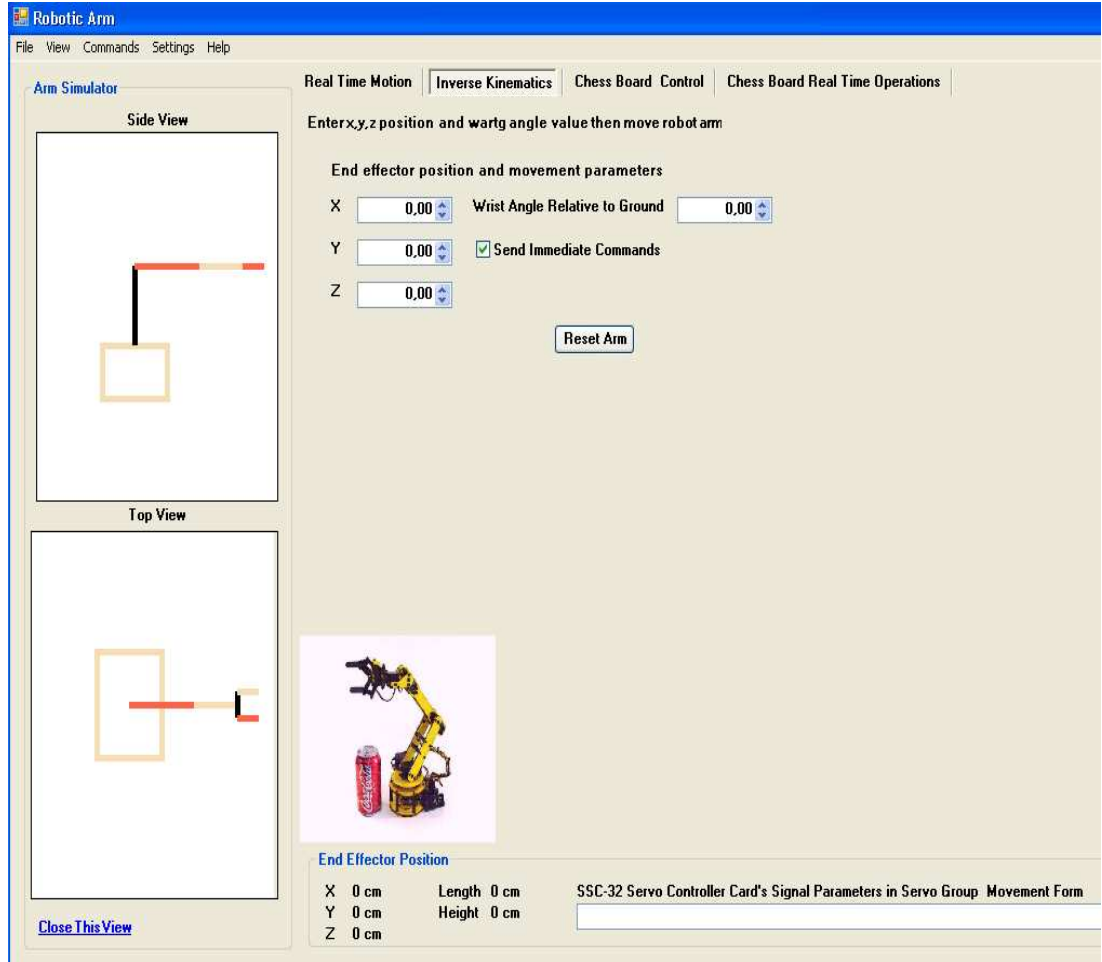
dışında satranç uygulamalarda kullanılacak elektro mıknatısta yazılım tarafından yönlendirilmektedir. Yame yazılım paketi dört ana ekran içerir bu ekranlar ve işlevleri sırasıyla Gerçek Zamanlı Hareket, Ters Kinematik, Satranç Kontrol, Satranç Gerçek Zamanlı Uygulamalar Ekranlarıdır.

6.1.1 Yazılım ve uygulama içeriği

Gerçek Zamanlı Hareket ekranı şekil 6.1’de görülmektedir. Düz kinematik çözümü içeren bu ekran, ekranda görülen çekilir bar (track bar) ve sayısal giriş kutusu (numeric up down) giriş noktalarından ilgili eklemin istenilen açı değeri girilir ve ara yüz gerekli komutu kontrol kartına gönderip eklemin istenilen açıyı yapması için eklemin servo motorunu tetikler ve hareket gerçekleşir. Ekranın solunda görülen simülatör yazılım ise hesaplanan hareketi taklit eder. Ters Kinematik (Inverse Kinematics) ekranı ise şekil 6,2 ‘de görülmektedir; Ters kinematik çözümü içeren bu ekran, Kullanıcıdan XYZ düzleminde pozisyon değeri ile sabit WARTG (omuz, eklem ve bilek açılarının toplamını) ister. Burada kullanılan ters kinematik çözüm Laurant Gay’in Lynx-6 Robot kol için geliştirdiği bir yöntemdir, üç özgürlük derecesi (DOF) ‘tan fazlasına sahip robotlar için birden çok ters kinematik çözüm vardır bu yüzden Gay’in yöntemi kullanılarak sabit tek bir çözüm elde edilmiştir (Gay 2005).

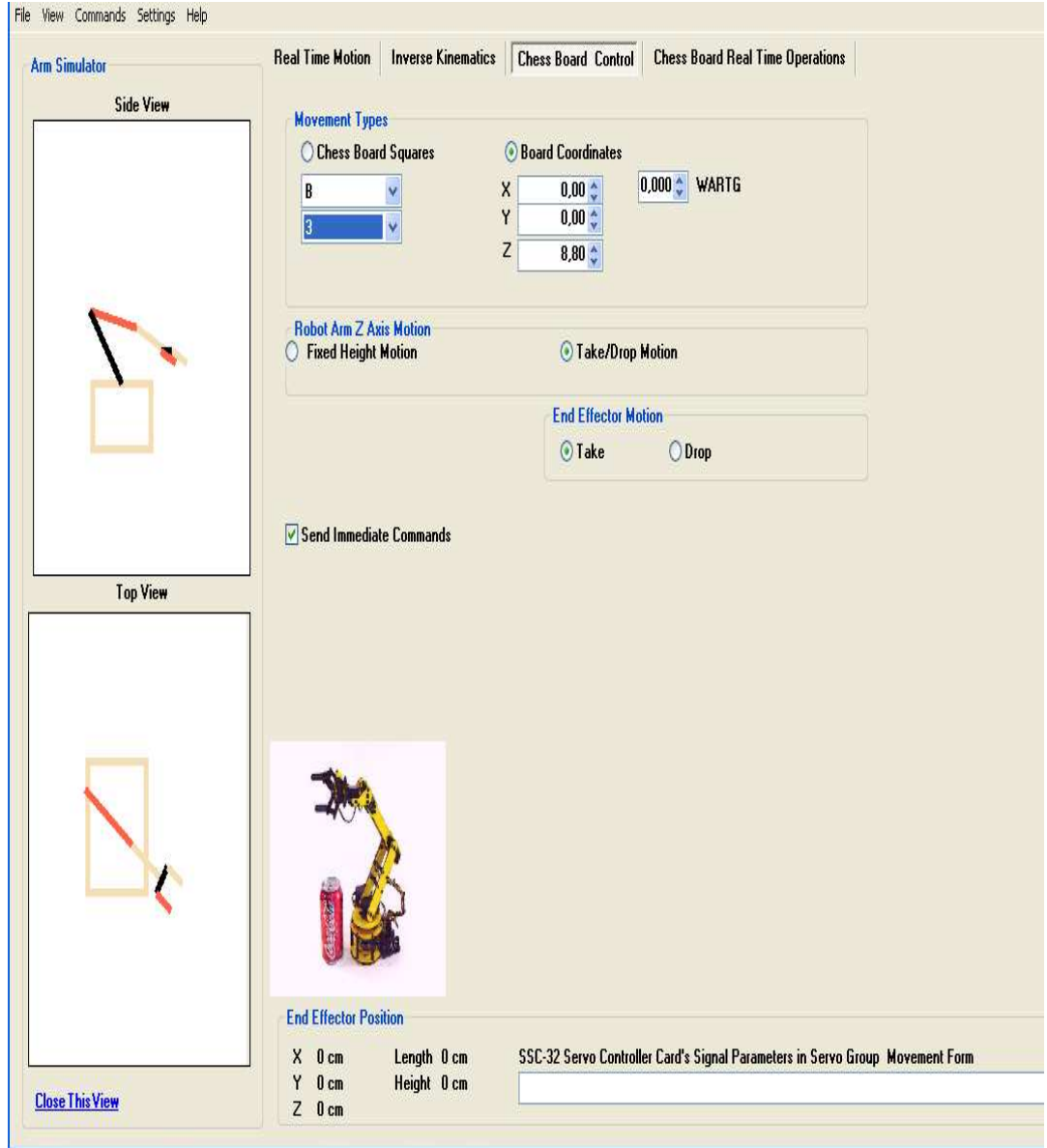


Şekil 6.1 Gerçek zamanlı hareket ekranı



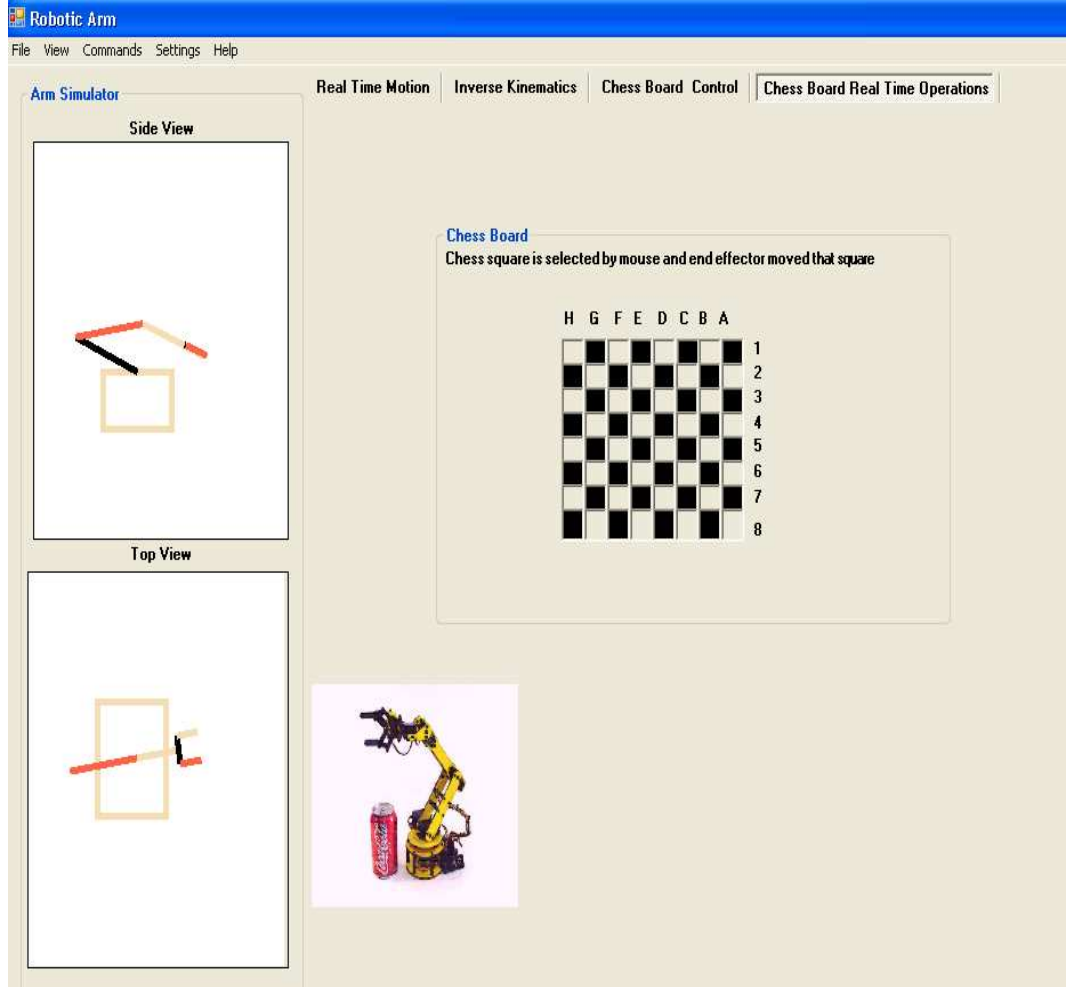
Şekil 6.2 Ters kinematik ekranı

Satranç Kontrol Ekranı ise şekil 6.3’de görülmektedir; Bu ekran iki ana işlev yapar Bu işlemlerden birincisi, ters kinematik ekranın yaptığı gibi çalışma uzayında XYZ pozisyon değerleri ve ile sabit WARTG (omuz, eklem ve bilek açılarının toplamı) giriş verilerini kullanarak ters kinematik çözümü bulup, kolun istenilen hedefe yönlendirilmesini sağlamaktır. İkincisi ise istenilen tahta koordinat sistemine göre robot kolun hareket ettirilip hedef noktada ki taşı alma veya bırakma işleminin gerçekleştirilmesidir. Taşın alınıp bırakılması sisteme bağlı elektro mıknatısa sinyal verilip, sinyalin kesilmesi ile ilişkilidir. Bu ekran sayesinde, Robot Kola gerçek zamanlı olarak satranç oynatılabilir (Arca *et al.* 2005) .



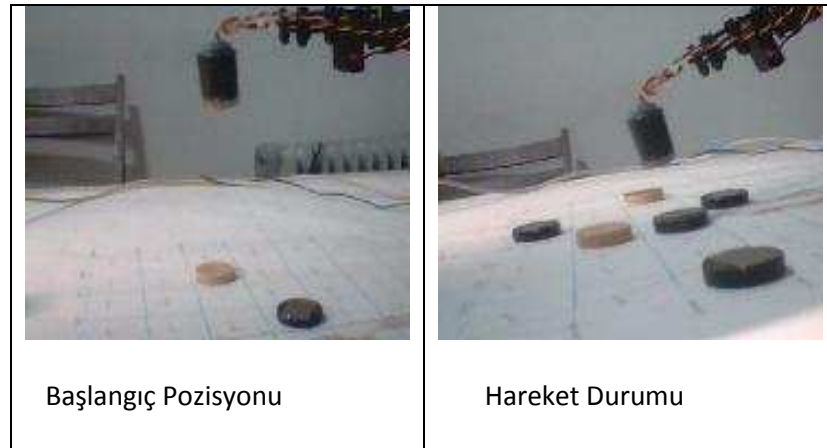
Şekil 6.3 Satranç kontrol ekranı

Son Ekran olan Satranç Tahtası Gerçek Zamanlı Operasyonlar Ekranı şekil 6.4 'de görülmektedir. Ekranın ortasında bir satranç tahtası figürü bulunmaktadır. Kullanıcı figürde istediği kareye fare yardımıyla dokunup kolun gerçek satranç tahtasında o hedef pozisyona gitmesini sağlayabilir. Simülatörde her bir hareketi taklit eder, bu bir önceki Satranç kontrol ekranının daha kolay kullanılabilir bir sürümüdür, yalnız bu ekranda, taş alıp bırakma işlevi gerçekleştirilmemektedir.



Şekil 6.4 Satranç tahtası gerçek zamanlı operasyonlar ekranı

Satranç tahtasında ki pozisyon ve hareket işlemleri şekil 6.5 - 6.6 'da görülmektedirler.



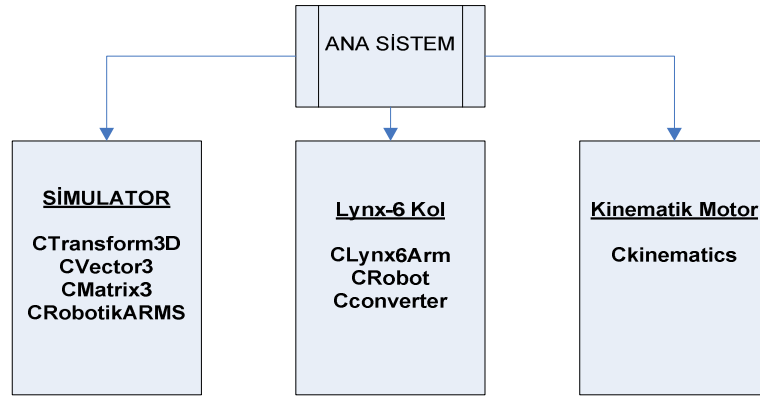
Şekil 6.5 Satranç tahtasında durumlar



Şekil 6.6 Tahta üzerinde hareket işlemleri

6.1.1.1 Yazılımın yapısı ve sistematığı

Geliştirilen yazılım üç temel modülden oluşur. Bu yapılar içerdiği alt fonksiyonlar ile sistemin çözümü ve kontrolünü yaparlar. Bu fonksiyonlar ve içerikleri açıklamaları ile aşağıda gösterilmektedir.



Şekil 6.7 Yazılımın sistematığı ve yapısı

SİMULATOR: Bu modül ve içerdiği alt fonksiyonlar Bundan sonraki dört yazılım paketinde de kullanılacaktır. Bu sistem, LYNX-6 robot kolun, 2 boyutlu (2D) bir görsel simülasyonudur. Bu yazılım sayesinde Fiziksel bir LYNX-6 robot kola ihtiyaç duyulmadan, fiziksel robotun hareket karakteristiği taklit edilir. Şekil 6.8’de simülatör nesnesi görülmektedir.

CVector3: Bu fonksiyon temel vektör işlemlerini gerçekleştirmek için geliştirilmiştir. Bu bağlamda, vektörlerin çapraz çarpımı, nokta çarpımı ayrıca vektörün uzunluğunu bulmak, birim vektör değeri gibi temel işlemleri bu fonksiyon tarafından gerçekleştirilir.

CMatrix3: Bu fonksiyon $n \times n$ 'lik matrisleri oluşturma, matrisler arası aritmetik işlemleri gerçekleştirmek amacıyla geliştirilmiştir.

CTransform3D: Bu fonksiyon, Üç boyutlu (3D) uzayda giriş olarak verilen vektör nesnesini X, Y veya Z eksenlerinden herhangi birinin etrafında veya hepsinde sırası ile istenilen açı değerinde döndürme işlemini gerçekleştirir. Şekil 6,8'de üç ekseninde de sırasıyla döndürme işlemini gerçekleştiren fonksiyonda ki ilgili kod parçasığı görülmektedir.

```
public static CVector3 Rotate( CVector3 vPos, CVector3 vAngle )
{
    CMatrix3 tx,ty,tz;
    CVector3 vRot;
    float cx,cy,cz,sx,sy,sz;
    cx = (float)Math.Cos((vAngle.X)*Math.PI/180);
    sx = (float)Math.Sin((vAngle.X)*Math.PI/180);
    cy = (float)Math.Cos((vAngle.Y)*Math.PI/180);
    sy = (float)Math.Sin((vAngle.Y)*Math.PI/180);
    cz = (float)Math.Cos((vAngle.Z)*Math.PI/180);
    sz = (float)Math.Sin((vAngle.Z)*Math.PI/180);

    tx = new CMatrix3( 1,    0,    0,
                      0,    cx, -sx,
                      0,    sx,  cx);

    ty = new CMatrix3( cy,    0,    sy,
                      0,    1,    0,
                      -sy,    0,    cy);

    tz = new CMatrix3( cz, -sz,    0,
                      sz,  cz,    0,
                      0,    0,    1);

    vRot = vPos;
    vRot = tx * vRot;
    vRot = tz * vRot;
    vRot = ty * vRot;
    return vRot;
}
```

Şekil 6.8 Tüm eksenler için (X;Y;Z) dönüş fonksiyonu

CRobotikARMS: Bu fonksiyon, .Net Çizgi kütüphanelerini kullanarak Simülatör nesnesinin görsel olarak oluşmasını ve girilen matematiksel değerlere göre tekrar ekranda çizilmesini sağlayan grafiksel bir yazılım uygulamasıdır. Bu kapsamda simülatör nesnesinde, Her bir robot eklemine temel özelliklerini temsil etmesi için bir veri yapısı nesnesi oluşturulur. Bu veri yapısı nesnesi yazılımda Line olarak adlandırılmıştır. Bu yapı hem görsel hemde matematiksel parametreleri tutar. Line veri yapısı tipinde Buffer isiminde Dizi (Array) oluşturulup, her bir eklem (Omuz, Dirsek gibi) Bu dizinin bir elemanı olarak tutulup dönüşüm ve çizim işlemleri bu dizi veri yapısı üzerinden esnek bir biçimde yapılır. Şekil 6.9’da Line veri yapısı ve bu veri yapısı tipinde ki bir dizi elemanına robotun bir eklemine (Omuz) bilgilerinin ilk değeri olarak aktarılması gösterilmektedir. Ayrıca Şekil 6.10’da örnek kod parçası, Simülatörün üstten görünüşünün oluşumu görülmektedir.

```
private struct Line {  
  
    public CVector3 vDir; // Doğrunun uzunluğu ve yönü  
    public CVector3 vPos; // Doğrunun pozisyonu  
    public CVector3 vTo; // Doğrunun son noktası  
    public string Name; // Doğrunun İsmi  
    public int Color; // Doğrunun Rengi  
  
}  
  
Buffer[0].Name = "Omuz";  
Buffer[0].vDir = new CVector3(0,-1,0) * m_OmuzUzunluğu;  
Buffer[0].vPos = vPos;  
Buffer[0].Color = 0;
```

Şekil 6.9 Çizim için veri yapısı ve dizi nesnesi


```

// top view
public void DrawXZ( Graphics g )
{
    // transform coordinates
    Transform();

    // render base
    //g.DrawRectangle(Color2, vMainPosition.X-m_BaseWidth/2, vMainPosition.Z-m_BaseLength/2, m_BaseWidth, m_BaseLength);
    // vMainPosition.Z = 120;
    g.DrawRectangle(Color2, vMainPosition.X-m_BaseWidth/2, vMainPosition.Z-m_BaseLength/2, m_BaseWidth, m_BaseLength);
    // render arm
    for( int i=0; i<7; i++ )
    {
        switch( Buffer[i].Color )
        {
            case 0:
                g.DrawLine(Color0, Buffer[i].vPos.X, Buffer[i].vPos.Z, Buffer[i].vTo.X, Buffer[i].vTo.Z);
                break;
            case 1:
                g.DrawLine(Color1, Buffer[i].vPos.X, Buffer[i].vPos.Z, Buffer[i].vTo.X, Buffer[i].vTo.Z);
                break;
            case 2:
                g.DrawLine(Color2, Buffer[i].vPos.X, Buffer[i].vPos.Z, Buffer[i].vTo.X, Buffer[i].vTo.Z);
                break;
        }
    }
}

```

Şekil 6.10 Simülör kod parçacığı

LYNX-6 KOL: Bu modül ve içerdığı alt modüller geliştirilen bu yazılım ve bir sonraki üç taş oynayan Robot kol uygulamasında ortak olarak kullanılacaktır. Bu modül Lynx-6 Robot kolun bütün fiziksel parametrelerinin tutulduğu; Bilgisayar ara yüzü ve kontrol kartı arasında ki ilişkiyi sağlayan, kart vasıtasıyla robotun fiziksel kontrolüne imkan sağlayan alt fonksiyonları içerir.

CLynx6Arm: Bu fonksiyonun Temel görevlerinden biri SSC-32 kontrol kartı ile Robot kol arasında ki bağlantıdan sorumlu olmasıdır. Kontrol Kartı ve sistem arasında ki veri iletişimi Seri Giriş (PORT) vasıtasıyla sağlanmaktadır. Kart, seri giriş soketi ile ilgili bilgisayara bağlanmaktadır. Yazılım ve kart arasında ki veri alış verişini kontrol etmek için bu fonksiyon içinde yazılımın desteklediği Seri Giriş nesnesi tanımlanmıştır. Şekil 6.11’de Seri Giriş Nesnesi ve bağlantı özelliklerini içeren kod parçası görülmektedir. Bağlantı sağlandıktan sonra kart vasıtasıyla motorlar istenilen şekilde kontrol edilebilirler. Bu fonksiyon, içerdığı alt fonksiyonlarla bağlantı sağlandıktan

sonra kullanıcıdan gelecek açı ve pozisyon değerlerine göre karta veri gönderip gerekli hareketlerin yapılmasını sağlar.

```
SerialPort m_Bağlantı; // Seri Giriş Nesnesi

if (!m_Bağlantı.IsOpen) // Bağlantı Kapalıysa
Başlangıç Değerleri Ata
{
    m_Bağlantı.PortName = "COM8";

    m_Bağlantı.BaudRate = 115200;

    m_Bağlantı.Parity = Parity.None;

    m_Bağlantı.DataBits = 8;

    m_Bağlantı.StopBits = StopBits.One;

    m_Bağlantı.Open();

}
```

Şekil 6.11 Seri giriş nesnesi ve tanımlamaları

CConverter: Bu fonksiyon temel dönüşüm işlemlerini gerçekleştirir. Radyan ve derece değerleri arasında ki dönüşüm bu fonksiyon sayesinde gerçekleşir. Ayrıca bu sayısal değerler SSC-32 kontrol kartının anlayacağı parametre değerlerine çevrilir (Dvorsky 2008). İşte bütün bu dönüşüm işlemlerinden bu fonksiyon sorumludur. Fonksiyonun sahip olduğu temel alt fonksiyonlar ve içerikleri şekil 6.12’de görülmektedir.

```

public static double DegreeToRadian( double Degree )
{
    return Degree * Math.PI / 180;
}

public static double RadianToDegree( double Radian )
{
    return Radian * 180 / Math.PI;
}

public static int DegreeToPulseExtended( double Degree )
{
    return 1500 + (int)(Degree * 11.111111111111111);
}

public static int DegreeToPulse( double Degree)
{
    return 1500 + (int)(Degree * 13.333333333333333);
}

public static int DegreeToPulse(double Degree, double MinPulse)
{
    // 1500 pulse sıfır pozisyonu
    return 1500 + (int)(Degree * MinPulse);
}

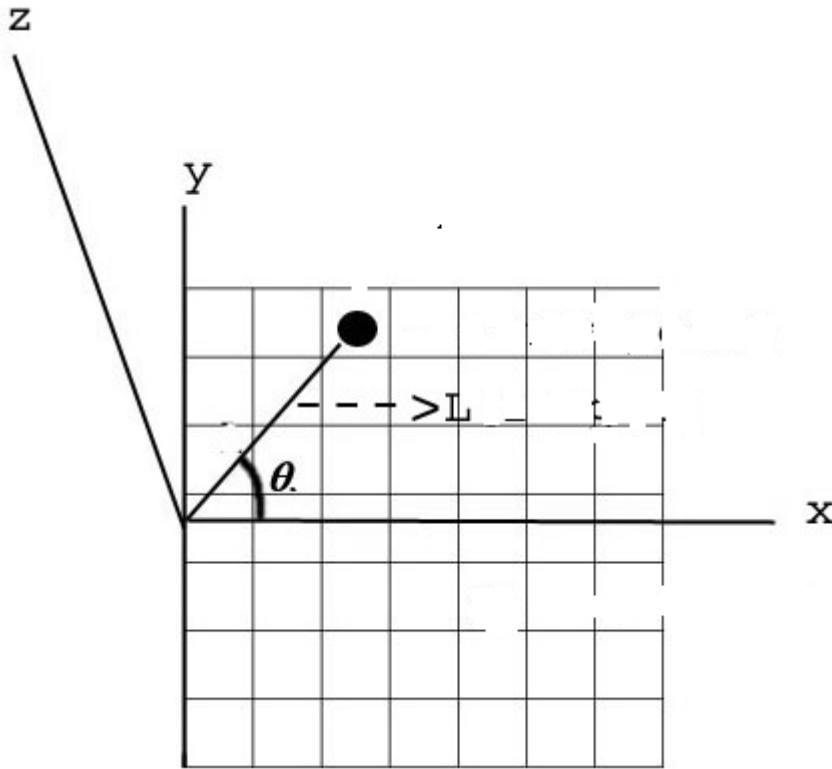
```

Şekil 6.12 CConverter fonksiyon içeriği

Kinematik Motor: Bu fonksiyon, Lynx-6 Robot kolun, Hareketsel karakteristiğini sağlamak için geliştirilmiştir. Gay'in (2007) yılında geliştirdiği Kinematik çözümler bu fonksiyonda kullanılmıştır. Ayrıca Gay'a (2007) ek olarak Taban (Base) eklemi

içinde bir çözüm yöntemi, ters kinematik çözüm kümesine eklenmiştir. Taban Ekleminin Dönme açısı, yani z eksenini etrafında dönme işlemi Şekil 6.13’de gösterilmektedir.

$L = (x^2 + y^2)^{1/2}$ ve Dönme açısı θ açısı ise, $\theta = \arctan(y/x)$ olarak hesaplanır. Bu yöntem ile Gay’in (2007) Üç özgürlük derecesine (3 DOF) göre ters kinematik hesaplama yöntemi Dört özgürlük derecesine (4 DOF) genişletilip, Robot kolun oyun z eksenini etrafında dönmesi için gereken imkân sağlanmıştır, bu da geliştirilen uygulamalarda robot kolun oyun tahtalarında sağa ve sola dönmesini sağlamaktadır.

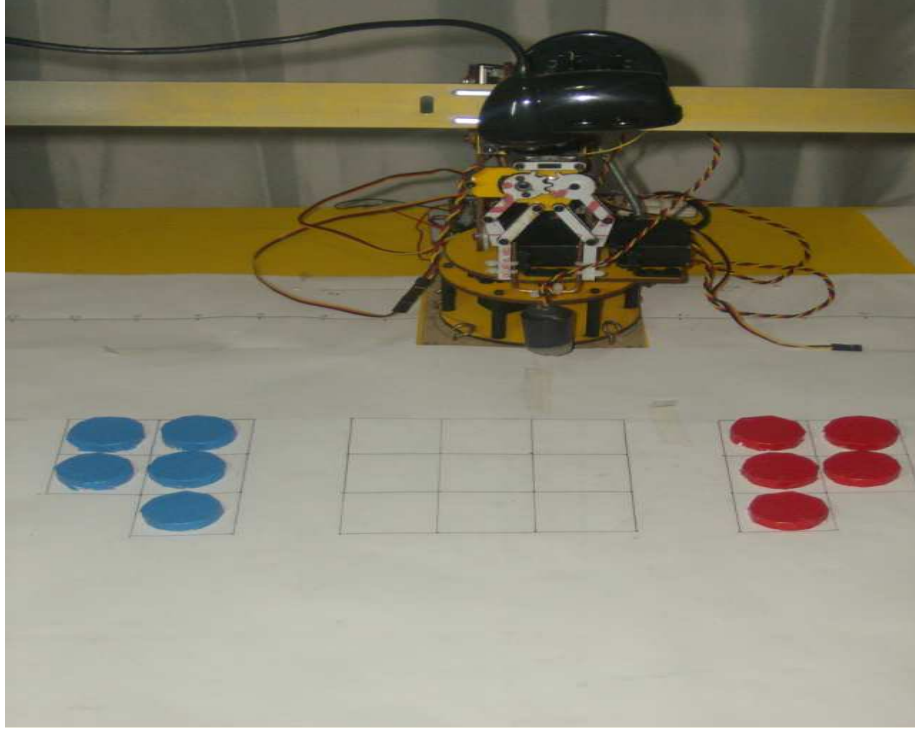


Şekil 6.13 Taban θ açısının hesaplanması

6.2 Üç Taş Oynayan Robot Kol (Dinamik Uygulama)

Yapay zekâ ve İmge işleme teknikleri kullanılarak Üç Taş oyununun gerçek zamanlı olarak robot kol ve insan arasında oynanmasını sağlayan ikinci yazılım paketi bu çalışma kapsamında gerçekleştirilmiştir Robot Kolum hareket kabiliyeti yanında karar verme ve görme işlemleri de sırasıyla, Yapay zekâ ve İmge işleme Teknikleri kullanılarak gerçekleştirilmiştir.

İmge işleme tekniklerinden çıkarma (subtraction) tekniği kullanılarak oyun tahtasında ki taşların pozisyonu belirleniyor. Buna karşın yapay zekâ yöntemlerinden Min-Max algoritması ile robotun yapması gereken en uygun hamleyi bulması yani karar vermesi sağlanıyor. Oyun alanı ve robot kolun durumu şekil 6.14 ‘ de görülmektedir.



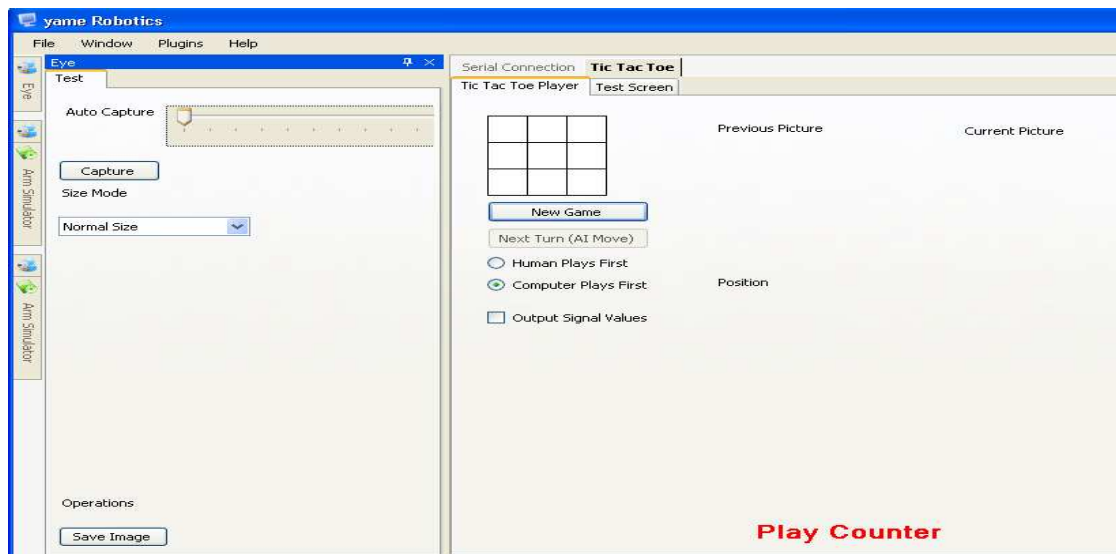
Şekil 6.14 Üç taş oyunu çalışma alanı ve robot kolun durumu

6.2.1 Yazılım ve uygulama içeriği

Geliştirilen yazılım paketi, dinamik bir uygulama olan Üç Taş oyunu gerçekleştirmesinin yanında robotun, simülasyon ek yazılımına ve devamlı çalışma alanıyla iletişim sağlayan kamera algı sistemine de sahiptir, yazılım paketinin ara yüzü Şekil 6.15 ‘de görülmektedir.

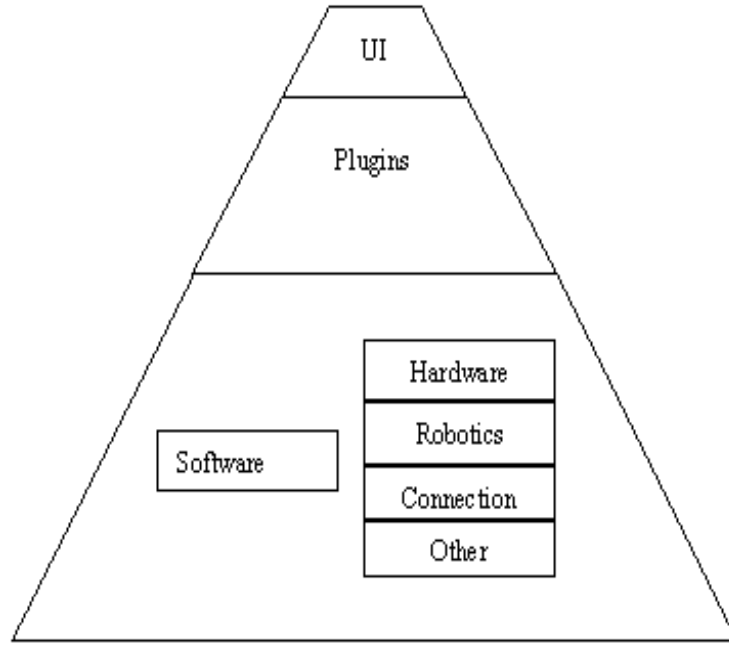
Bu yazılım kapsamında altı eksenli robot kolun kinematik analizi ile yapay zekâ (Artificial Intelligence) ve imge işleme (Image Processing) teknikleri birleştirilerek, gerçek zamanlı kamera destekli akıllı bir uygulamada kullanılması hedeflenmiştir. Bu yazılım robot kol ile insanın üçtaş (Tic-Tac-Toe) oynamasını sağlamaktadır. İnsanın

oyun tahtasında ki hareketinin kamera yardım ile belirlenip, tahtanın yeni durumunun imge işleme yöntemlerinden (Çıkarma metodu) kullanılarak gerçekleştirilir. Bu yeni konumun sistemin yapay zekâsına girdi olarak verilmesi; sisteminde Min-Max Algoritmasının kullanılarak en uygun hamlenin belirlenmesi ve bu hamlenin hareket sistemine girdi verisi olarak dönmesi, sistemin temel çalışma prensibidir. Bu gerçek zamanlı akıllı oyun yazılımında kullanılan yapay zekâ tekniği yüzünden, insanın en fazla başarabileceği oyunu beraberliğe getirebilmesidir.



Şekil 6.15 Üç taş oyununun yazılım paketi ara yüz ekranı

Geliştirilen bu esnek yazılım paketi simülasyon ve gerçek zamanlı uygulamaları destekler. Yazılım kullanıcı talebini her iki uygulama tipini de destekler. Şekil 6.15’de görülen ana ekranın sol üst köşesinde Göz Motoru (Eye Engine) Bulunmaktadır, Bu motor kamera görüntü aktarımı ve görüntü işlenmesi ile ilgili bütün işlemlerden sorumludur. Ekranın sağ tarafında ise oyun alt ekranı vardır bu ekran oyunun her anında güncellenir ve fiziksel oyun tahtasında ki taşların pozisyonunu ve yerleşimini gerçek zamanlı olarak gösterir. Sistemin Genel Mimarisi şekil 6.16’da görülmektedir.

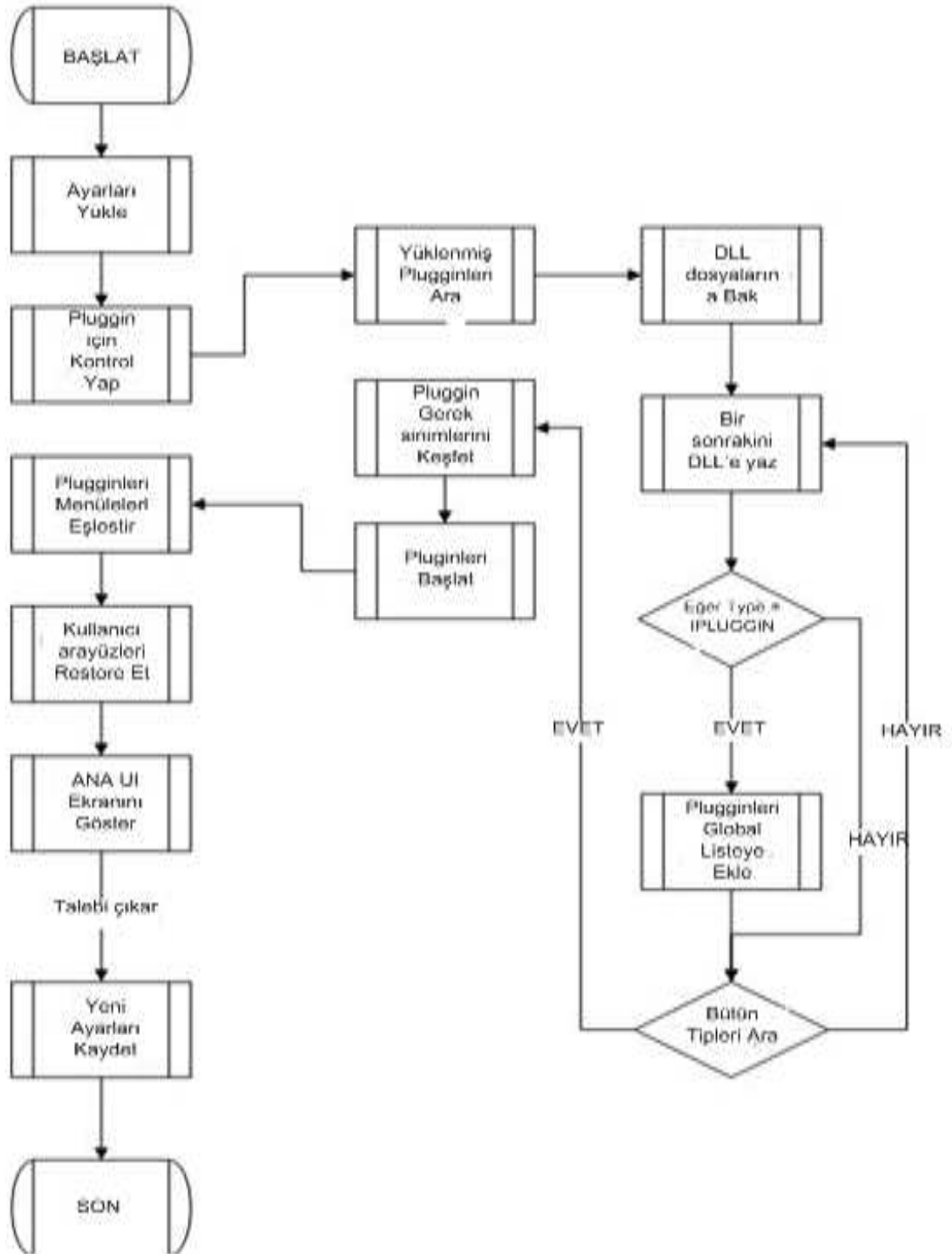


Şekil 6.16 Sistemin ana mimarisi

Yukarı da şekilde gösterilen mimari yazılımın esnek yapısını temel bir biçimde göstermektedir. En üst kısımda Kullanıcı Ara yüzü (UI) bulunmaktadır, bu katman Şekil 6.15 ‘de görülmektedir. Kullanıcı ve sistem arasında iletişim sağlayan en üst düzey arabirimdir. Bir alt kısımda Eklenti (Plugins) Katmanı vardır bu katman ise yeni eklenecek modüllerin derleme (compile) işlemine gerek duyulmadan sisteme eklenmesini sağlar, yazılımın esnek yapısı buradan gelmektedir. Geliştirilmiş yazılım paketine dışarıdan herhangi bir ek yazılım modülü çok kolay bir biçimde entegre edilebilir. Bir önceki kısımda anlatılan, Yame uygulamasının alt modülleri (Simülatör,Lynx-6 Kol ve Kinematik Motor) ile beraber bu uygulamaya özgü Oyun Motoru, Göz Motoru , gibi alt modüllerin hepsi birere Eklenti(Plugins) olarak kabul edilir. İlk üç alt modül Yame yazılım ile aynı özellikleri içerir ve aynı kod içeriğine sahiptir. Buna karşın Oyun Motoru ve Göz Motoru bu uygulamaya özgüdür ve sırasıyla bölüm 6.2.2 ve 6.2.3’de bu motorların çalışma mantıkları anlatılacaklardır. Eklenti (Plugins) Çalışma mantığının akış diyagramı 6.17’de görülmektedir. Sistem çalışmaya başlayınca, yüklenmiş Eklentiler (Plugins) aranır, bunun için uzantısı “DLL” olan bütün dosyalara bakar eğer, bulunan “DLL” uzantılı Eklenti sisteme uygun ise o an aktif Eklentiye GLOBAL nesnesine ekler. GLOBAL nesne dosyası sistemde ki Eklentilerin

(Pluggins) tutulduğu bir veri yapısıdır. Bundan sonraki adım Eklenti (Pluggin) gereksinimlerinin belirlenmesidir, örnek olarak Eye Eklentisi için sistemde fiziksel bir kamera olması gerekmektedir. Bu aşamadan sonra Eklenti (Pluggin) yapılarını Ara yüze(UI) entegre edilmesi aşaması gelir; bu bağlamda menülere yüklü Eklenti yapıları eklenir ayrıca Ekranlar, Eklenti yapılarına göre Restore edilirler. Yeni Ayarlar sisteme kayıt edilip, Derleme işlemine gerek duymadan sisteme yeni alt modüller eklenir.

En alt katmanda Yazılım(Software) Donanım (Hardware) , Fizikse Robot (Robotics) Bağlantı Nesnesi (Connection) ve Diğer gereksinimler bulunmaktadır. Bu katmanda Yazılım, sistemin sahip olduğu sabit yazılım öngörülmektedir. Bu kısım Eklenti(Pluggin) ve Ara yüz(UI) gibi katmanların dışında kalan en alt seviye yazılım sistemin temsil eder. Donanım kısmı ise Yazılımın yüklü olduğu Bilgisayar Sistemidir. Fiziksel Robot kısmı ise Lynx-6 Robot koldur. Bağlantı (Connection) Nesnesi, Fiziksel Robot ile bilgisayar sistemi arasında ki bağlantıyı kuran Kontrol Kartıdır. Diğer Gereksinimler ise Kamera, oyun tahtası ve elektro mıknatıs gibi uygulamalarda ki diğer fiziksel ve donanımsal aygıtlardır.



Şekil 6.17 Eklenti (Plugins) yapısı akış diyagram

6.2.2 Min – Max algoritması

Robot Kolun, karar vermesi bu çalışmada yapay zekâ yöntemlerinden min-max algoritması kullanılarak gerçekleştirilmiş, rakibin hareketine karşılık en uygun hareket belirlenip robotun kazancının en büyük olması hedeflenmiştir.

Min - Max algoritması esasında çok bilinen bir oyun stratejisi algoritmasıdır. Max ve Min iki kişilik oyunda taraflar olarak adlandırılır. Max ilk harekete başlayan taraftır, en sonunda oyun bittiğinde kazanan ödülü alırken kaybeden cezalandırılır. Herhangi bir oyunu bir çeşit arama problemi olarak tanımlarsak aşağıdaki maddelerle ifade edilebilir.

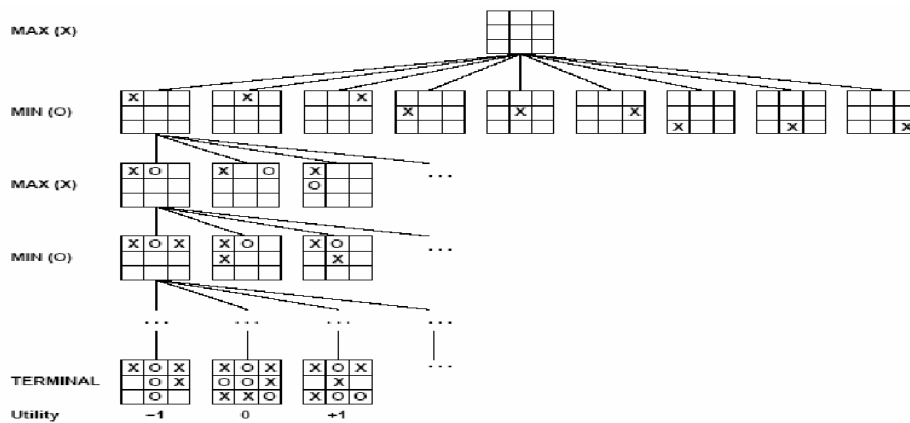
- Başlangıç Durumu: Boş Oyun Tahtası
- Başarı Fonksiyonu: Legal Hareketlerin Listesi
- Terminal Testi: Oyun Bitti mi?
- Başarı Fonksiyonu (Utility Function): Her bir terminale değer verilir galibiyet (+1) , mağlubiyet = (-1) , beraberlik (0)

Algoritmanın uygulandığı problemleri bir arama ağacı gibi düşünürsek, Her düğüme, MAX için bir kazanma ya da MIN için bir kazanma olup olmama durumuna, göre 1 ya da 0 verilecektir. Minmax işlemi ile bu değerler grafiğin yukarısına doğru ardışık ebeveyn düğümleri boyunca şöyle yayılacaktır. Ebeveyn durum MAX düğümünde ise, buna çocukları arasından maksimum değer verilir Ebeveyn durum MIN düğümünde ise, çocuklarının minimum değeri verilir. Her bir duruma iliştilmiş bir değer, bu oyuncunun başarmak için ümit ettiği en iyi durumun değeridir. Bu türetilmiş bir değerler bu olası hareketler içerisinde seçilir. Düğüm değeri yüksek olduğu sürece kazanma olasılığı artar. Değer küçüldüğü takdirde ise rakibin kazanma şansı artmış olur. Oyunculardan birisi sürekli olarak büyük değerleri diğeri ise küçük değerleri takip edecektir.

Başlangıç durumu ve legal hareketler her bir kenar için oyun ağacını tanımlar, Şekil 6.18’de Üç Taş oyunu için oluşturulan oyun ağacını gösteriliyor. Başlangıç durumundan itibaren Max’ın dokuz hamle şansı vardır. Max’ın hareketini X ile Min’in hareketini O

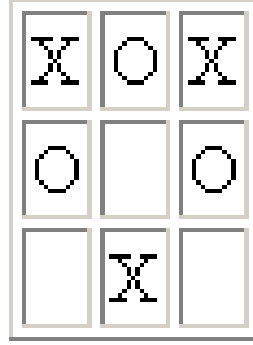
ile gösterirsek oyun her iki taraftan biri yaprak elemanlardan birine ulaşp, kendilerine kazandıracak, yani kendi taşlarının üç tanesinin yatay dikey veya çapraz olarak arka arkaya gelmesiyle oluşacak terminal durumların oluşması ile oyun sona erer.

Max 'ın görüş açısına göre bakarsak, Yaprakta ki yüksek değerler Max için atanırken düşük değerler Min için atanır. Max açısından baktığımız için, Max arama ağacını kontrol eder ve kendi için en uygun hamleyi seçip kazanımı en yükseğe çıkarırken Min 'ın kaybını da en yükseğe çıkaracak hamleyi yapar. Normal bir arama probleminde hedef duruma ulaşmak işlemi sabit sıralı hareketlerden oluşur. Buna karşın bir oyunda rakibinizde hareket yapacağı için, Rakibin her hareketi için en uygun en hatasız hareketin yapılması oyun algoritmalarının temel unsurudur. Üç Taş oyunu çok karmaşık bir oyun olmadığı için, Uygun arama ağacının oluşturtup uygun değer bulunması kolaydır. Optimum bir strateji için ağaçta ki her bir elemana Min-Max diye adlandırılan değerler atanır. Bu değerler Max'ın kazancını artırmaya yöneliktir. Max arama ağacını kontrol edip bir sonraki hamlesini belirlerken Yüksek Min-Max değerine sahip ağaç elemanını tercih eder buna karşın Min 'de tam tersini tercih edip Max'ın kaybetmesi için uğraş verir (Russell and Norving ,2001).Max Hamlesine karar verirken oluşturulan karar ağacında tekrarlı olarak rakibin hamlesine göre karşılık gelen bütün hamlelerin sonuçlarına bakar ve kendine en yüksek kazancı sağlayacak hamleyi seçer.



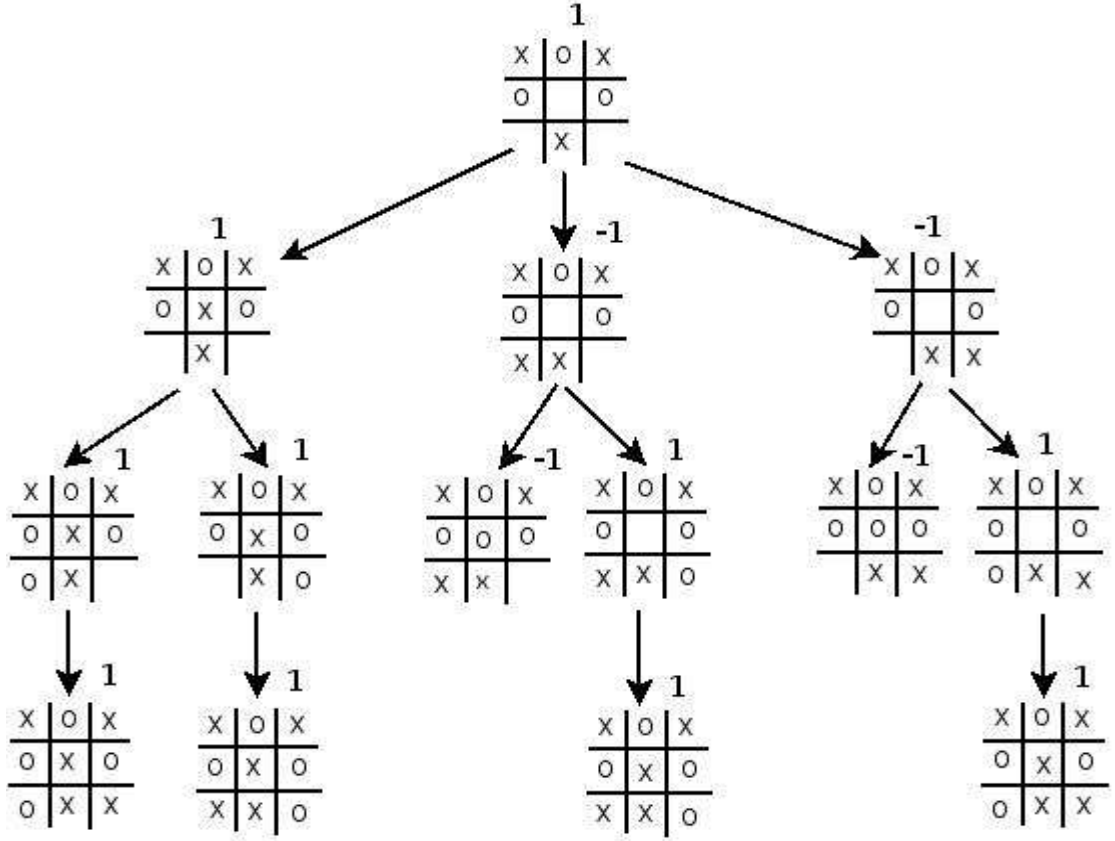
Şekil 6.18 Üç taş oyunu için arama ağacı

Algoritmanın çalışma prensibinin anlaşılması için Üçtaş oyunu için oluşan örnek bir pozisyon problem olarak ele alınmıştır Şekil 6.19’de görülmektedir. Problemin çözüm için oluşturulan arama ağacı şekil 6.20’de görülmektedir.



Şekil 6.19 Üç taş oyununu için örnek bir pozisyon

Bu örnek için Başarı Fonksiyonu (Utility Function): X kazanırsa 1; O kazanırsa -1 ve Beraberlik için 0 olarak her bir duruma atanıyor. Öncelikle yapılması gereken oyunun güncel anından itibaren bundan sonraki bütün pozisyon ihtimallerini kapsayan bir arama ağacı oluşturmak ve bu arama ağacının her bir Düğümüne (Node) yaprağa kadar bir başarı fonksiyonu değeri atamaktır. Başarı fonksiyonu atama işlemi ağaç ortaya çıkarıldıktan sonra sondan başa doğru yani yapraklardan köke doğru verilir. Şekil 6.18 ‘de yukarıda bahsedilen problem için bir sonraki hamlenin X ‘in yapacağı düşünülerek arama ağacı oluşturulmuş ve her bir Düğümüne ilgili Başarı fonksiyonu atanmıştır. Arama ağacı oluşturduktan sonra sıra karar aşamasına gelir. Sıra X oyuncusunda olduğu için ve X’in amacı bir sonraki hamlenin en iyisi olmasıdır, Bu da X oyuncusunu ‘1’ Başarı Fonksiyonuna sahip dala yönelmesidir. X oyuncusunun altında üç dal vardır üçünün içinde Başarı fonksiyonu en büyük olan sol daldır “1” . Bu nedenle X sol taraftaki yolu seçer. Bu aşamada sıra O’ya gelir O’nun amacı Rakibin bir sonraki hareketini en aza indirecek şekilde hareket emektir yani “-1” Başarı fonksiyonuna sahip Düğümüne ulaşmaktır. Ama O oyuncusu hangi hamleyi yaparsa yapsın altında ki dallarda ki Başarı fonksiyonu 1’dir Bu da X oyuncunun kazanma durumunu temsil eder. O oyuncusu iki daldan birini seçer, X oyuncusuna tekrar sıra geldiğinde altında ki dallar artık yapraktır ve daha derine inilmez her iki yapraktaki başarı fonksiyonu 1 olduğu için X oyuncusu hangi yolu takip ederse etsin oyunu kazanır. Arama işleminin algoritması Şekil 6.21’de görülmektedir.



Şekil 6.20 Üç taş oyunu için örnek arama ağacı

MINIMAX(oyuncu, tahta)

BAŞLA:

Eğer (oyun o anki aktif tahta pozisyonunda sonlanıyorsa)

Kazananı Döndür

Çocuk = Bu tahtada ki oyuncu için bütün legal hareketler

Eğer (Max oyuncusu Hamle sırası ise)

minimax fonksiyonun çağırıp çocuklar içinde en büyük

Başarı Fonksiyonuna sahip olanı döndür

Eğer (Min oyuncusu Hamle sırası ise)

minimax fonksiyonun çağırıp çocuklar içinde en küçük

Başarı Fonksiyonuna sahip olanı döndür

BİTİR:

Şekil 6.21 MinMax en iyi hareket algoritması

Yukarıda ki örnekte de uygulandığı gibi En iyi (Best-Move) Hareket Algoritması önce oyunun bitip bitmediğine bakar eğer oyun bitmişse hesaplanacak bir şey yoktur ve fonksiyon kazananı döndürür. Buna karşın eğer oyun bitmemişse algoritma her bir çocuk düğümünün değerine bakar tekrarlı (recursively) olarak ve Max için en yüksek Başarı Fonksiyonu değerine sahip olanı seçilirken Min için en düşük Başarı Fonksiyonu değerine sahip olan seçilir. Böylece her iki taraf içinde en uygun(optimum) çözümler elde edilir. Üçtaş oyunu çok fazla durum içermediği için MinMax algoritması bu oyun için mükemmel sonuca ulaşır fakat daha karmaşık oyunlar için Arama ağacını oluşturmak çok uzun süreler ister bu yüzden MinMax tek başına her uygulamada kullanılamaz. Çizelge 6.1’de değişik oyunlar için Arama ağacı Karmaşıklıkları gösteriliyor

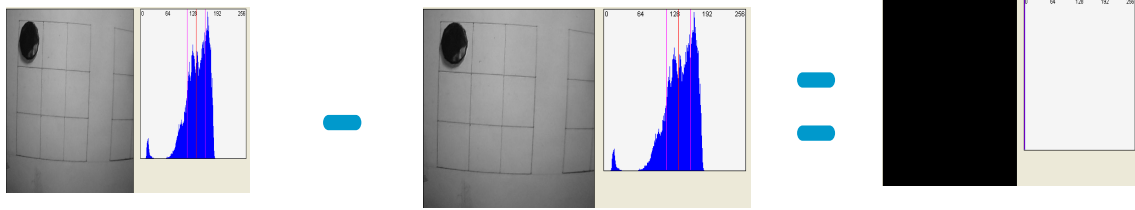
Çizelge 6.1 Oyun Ağacı Karmaşıklıkları

Üç Taş (Tic-Tac-Toe) – $9!$ – 362,280 durum
4 birleştir (Connect Four) – 10^{13} durum
Dama (Checkers) – 10^{18} durum
Satranç (Chess) – 10^{50} durum
Git (Go) – 10^{170} durum

6.2.3 İmge çıkarma ve bilgisayar algılaması

Akıllı uygulamalarda ki en önemli ihtiyaçlardan biride görme mekanizmasıdır, Bu uygulamada sensör olarak kamera kullanmıştır, kamera devamlı suretle oyunu alanını kullanıcıya gösterirken oyun esnasında ki değişimleri algılayıp tıpkı insan gözünün yaptığı gibi robot kolun görmesini sağlar. Görme işlemi kameradan devamlı veri alarak gerçekleşirken, algılama işlemi için imge işleme tekniklerinden imge çıkarma yöntemi kullanılmıştır. Şekil 6.22’de işlem gösterilmektedir. Oyun tahtasında dokuz kare olduğu için ve her bir karenin iki farklı değeri olduğundan dolayı oyunda on sekiz değişik taş varyasyonu vardır. Bu varyasyonlar önceden JPEG formatında çekilip veritabanına saklanır, oyunda yeni bir hamle yapıldığında gerçek zamanlı olarak oyun tahtasının

resmi çekilir, veritabanında saklı olan her bir resimden sırayla aktif resim çıkarılır. Oluşan fark resimlerinden en küçüğü hamleyi temsil eden resmin aktif resimden çıkarılması sonucu oluşan resimdir.



(Çekilen Resim)

(Veritabanında Olan Resim)

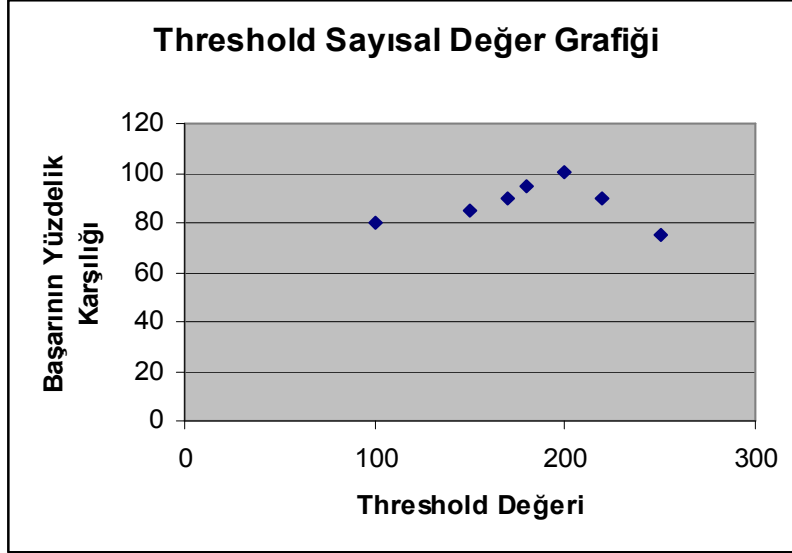
(Sonuç Resmi)

Şekil 6.22 İmge Çıkarma İşlemi

Hem Renkli hem de, Gri seviye resimde çalışmalar yapıp en doğru sonuçların renkli resimlerde sağlandığı görülmüştür. Çıkarma işleminde ki en önemli noktalardan biri uygun bir limit (Threshold) değeri belirleyip çıkarma işlemi esnasında bu değer ve üstünde ki renk değerlerini hesaba katmaktır. Aşağıda algılama işlemimizin algoritması maddeler halinde görülmektedir.

- On sekiz Pozisyonun Resimlerini Veritabanına JPEG Formatında Yerleştirilir
- Kameradan Aktif Pozisyon JPEG Resim formatında alınır
- Aktif resim veritabanında ki resimlerlerden, sırayla çıkarılır
- Fark Belirli Bir Limit (200) değerinin fazla ise ilgili değişken arttırılır
- On sekiz Resim ile çıkarma işlemi yapıldı kadar 3. ve 4. adımlar devam eder.
- En sonunda On sekiz resim değişkeni içinde en düşük olan sıralama algoritması ile seçilir bu aktif harekettir
- Algılama işlemi gerçekleştirilmiştir, yeni pozisyon bilgisi karar mekanizmasına aktarılır.

Limit (Threshold) değeri için bir dizi test yapılmış ve sonuçta iki yüz değerinin en uygun olduğu saptanmıştır, test ile ilgili grafik aşağıda ki şekilde gösterilmektedir.



Şekil 6.23 Threshold değeri için limit grafiği

6.3 Kinematik Simülasyon Yazılımı

Bu yazılım kapsamında iki, üç, dört ve beş eksenli Lynx robot kolların düz ve ters kinematik çözümlerini analiz edilmiş; Her bir eklemin hareket matrisleri oluşturulup; Eklemlerin birbiriyle ilişkileri matematiksel olarak simüle edilmiş. Geliştirilen bu yazılım Lynx manipülatörlerinin hareketsel karakteristiğinin fiziksel bir robota ihtiyaç duyulmadan gerekli deney ve araştırmaların için kullanılmasını sağlamıştır. Bu yazılım paketi 4. Kısımda da anlatılan teorik çalışmanın test edilmesini sağlarken, Lynx-6 robot kol için iyi bir alternatif kinematik çözüm sistemi ortaya koymaktadır.

6.3.1 Yazılım ve uygulama içeriği

Yazılımın Akış Şeması Şekil 6.24'de görülmektedir, Yazılımın esnek yapısı, piyasada ki her tür Lynx robot kolun (2.3.4.5 özgürlük derecesi) kinematik çözümünü destekleyen bir matematik motoruna sahiptir, bu motor düz ve ters kinematik çözümleri en az hata ile hesaplayıp istenilen hedeflere ulaşılmasını sağlar. Sistemin Matematik motoru, Düz kinematik (Forward Kinematics) çözümler için 4. Kısımda anlatılan dönüşüm matrisleri hesaplama yöntemiyle her bir eklemin dönüşüm matrisini hesaplar;

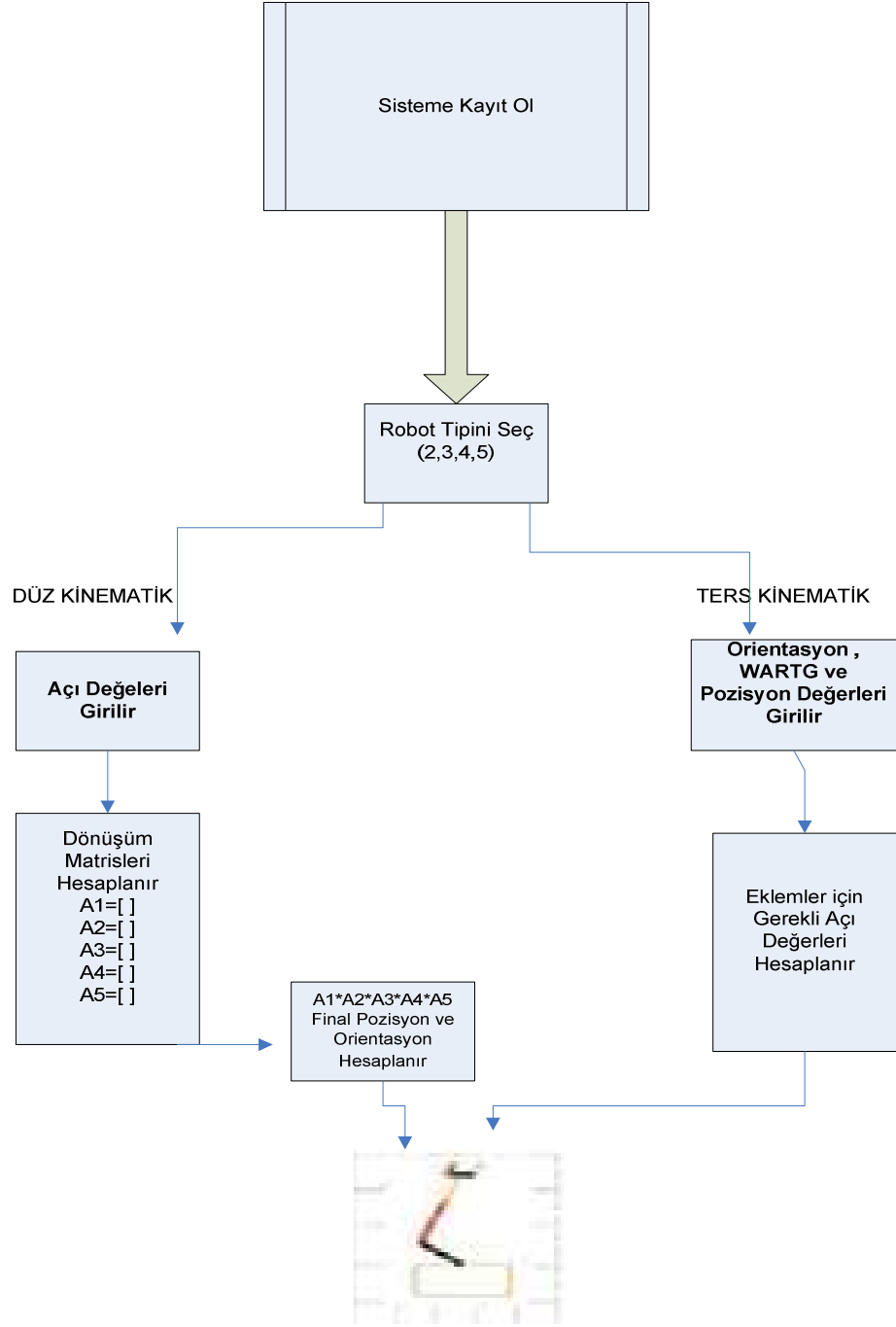
kullanıcının seçimine göre kinematik motor, verilen açı değerleri oluşan dönüşüm matrislerinden ilgili sonuç pozisyon ve yönelmesine sahip dönüşüm matrislerini hesaplar, seçim adımlar 1’den 4’e kadar görülmektedir.

$$T_s = A_1 * A_2 * A_3 * A_4 * A_5, 5 \text{ (Serbestlik Derecesi)} \quad (1)$$

$$T_s = A_1 * A_2 * A_3 * A_4, 4 \text{ (Serbestlik Derecesi)} \quad (2)$$

$$T_s = A_1 * A_2 * A_3, 3 \text{ (Serbestlik Derecesi)} \quad (3)$$

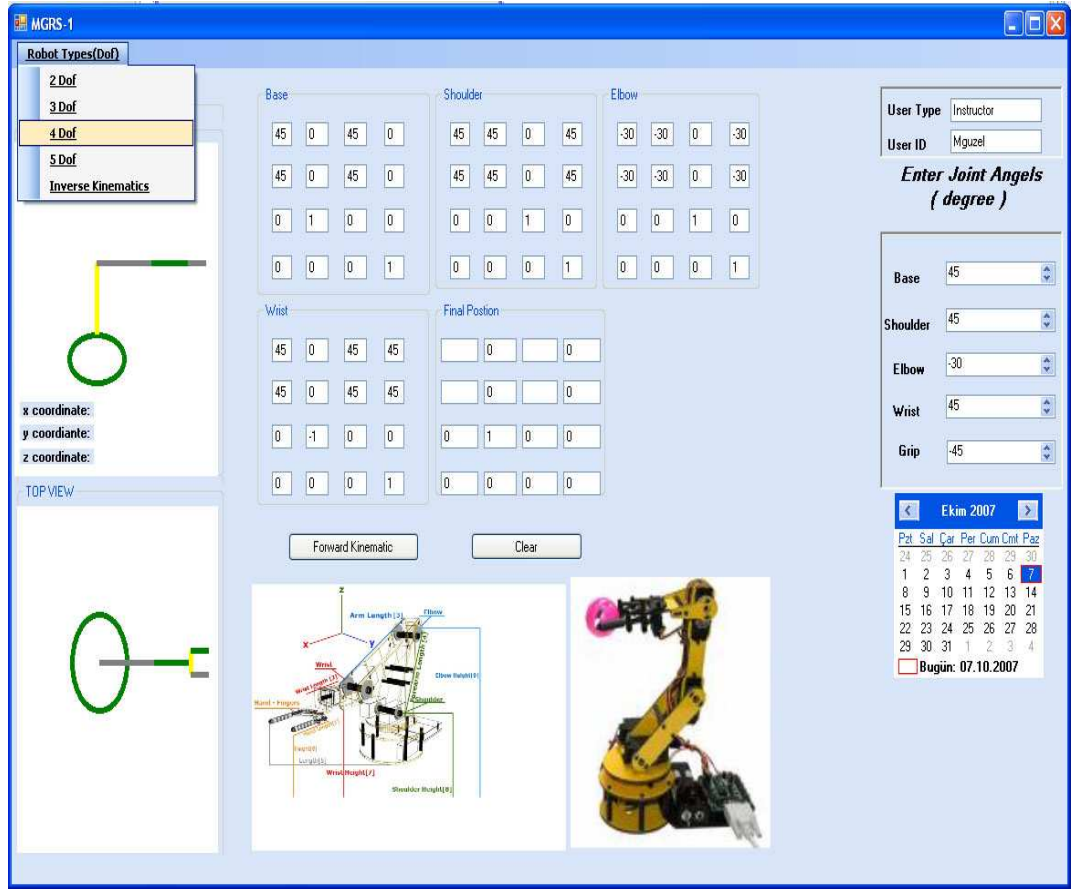
$$T_s = A_1 * A_2, 2 \text{ (Serbestlik Derecesi)} \quad (4)$$



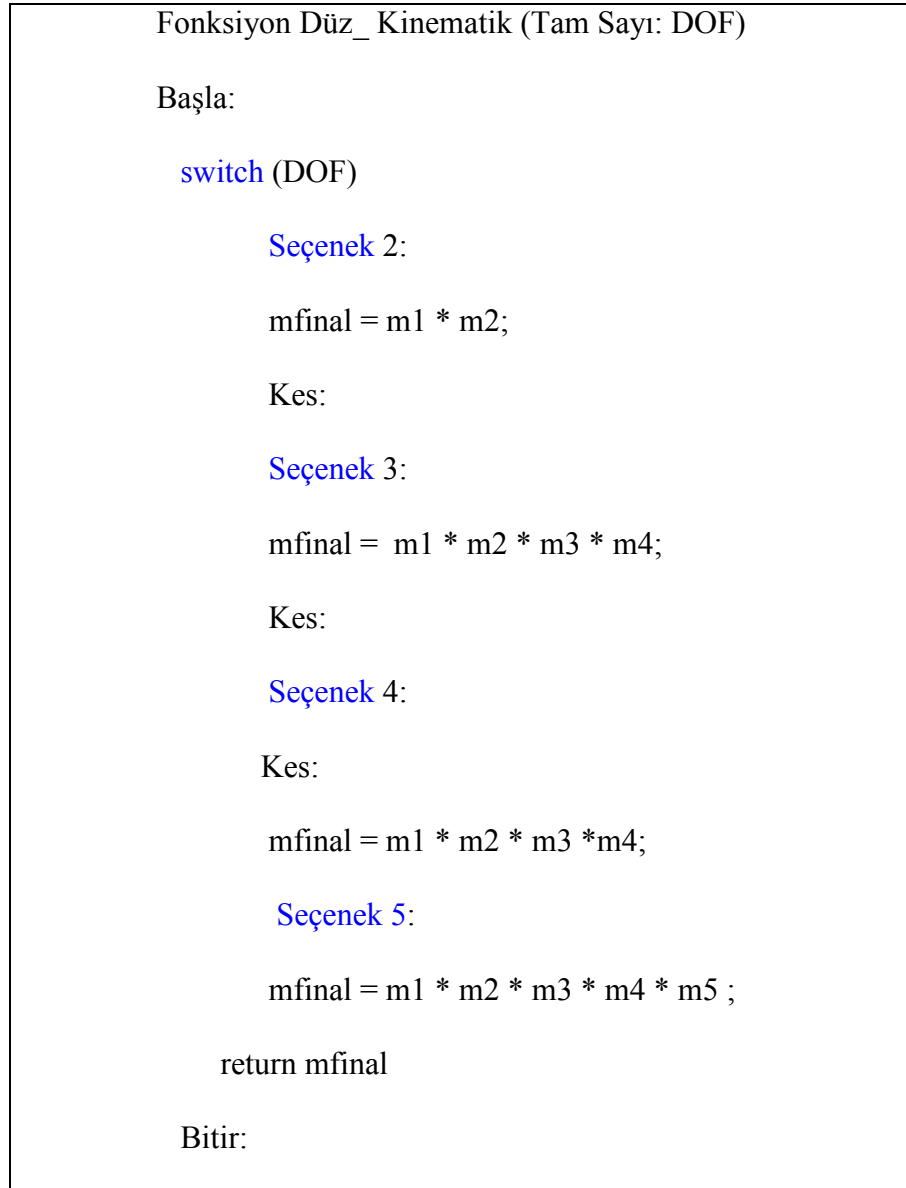
Şekil 6.24 Akış şeması diyagramı

Sistemin Ana Ekranı ve Düz Kinematik çözümlerin gerçekleştirildiği ekran olan giriş ekranı şekil 6.25’de görülmektedir. Bu ekran aktive edilen Lynx robot kolun sahip olduğu eklemlerinin dönüşüm matrislerini temsil eder, girilen her bir açı değeri için ilgili eklem matrisi oluşur ve oluşan matrislerin çarpımından sonuç pozisyon ve yönelme matrisleri oluşur. Ayrıca simülatör gerçek zamanlı olarak hareketi tekrar

ederken pozisyon değerlerini de (XYZ) simülatör ekranına gerçek zamanlı olarak görünür. Düz kinematik hesaplama sorumlu olan ana fonksiyonun temel algoritması Şekil 6.26’de gösterilmektedir. Bu fonksiyon parametre olarak özgürlük derecesini(DOF) alıp ona göre istenilen Düz Kinematik çözümü ekrana girilen açı değerlerine göre hesaplar.



Şekil 6.25 Ana ekran (DÜZ KİNEMATİK EKRANI)



Şekil 6.26 Düz kinematik fonksiyon

Ters kinematik çözümler için ise aşağıda ki şekilde görülen ekran kullanılır. Bu ekran giriş olarak T_f matrisin alırken sonuç olarak Kısım 4'dede bahsedilen ters kinematik denklem çözümleri kullanarak ilgili eklem açılarını hesaplar 5'den 10'a kadar görülmektedir. Bu denklemler Şekil 6.28'de algoritması gösterilen Ters kinematik işlemlerden sorumlu fonksiyonlarca kullanılıp çözüme ulaşılır. Bunlar **Ana Fonksiyon** ve **Ters Kinematik-Hesapla** fonksiyonlarıdır. **Ana Fonksiyonu** matris ve yerel değişkenlere ilk değerlerini verip Ters **Kinematik-Hesapla** fonksiyonunu çağırır.

Çağrılan fonksiyonda gelen özgürlük derecesi (DOF) parametresine göre aşağıda gösterilen açı hesaplama denklemlerinden (5..10) uygun olanlarını kullanıp giriş olarak verilen T_f Matris değerinden gerekli açı değerlerini çıkarır.

$$T_f = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\theta_{234} \text{ (WARTG} \rightarrow \text{)} \quad (5)$$

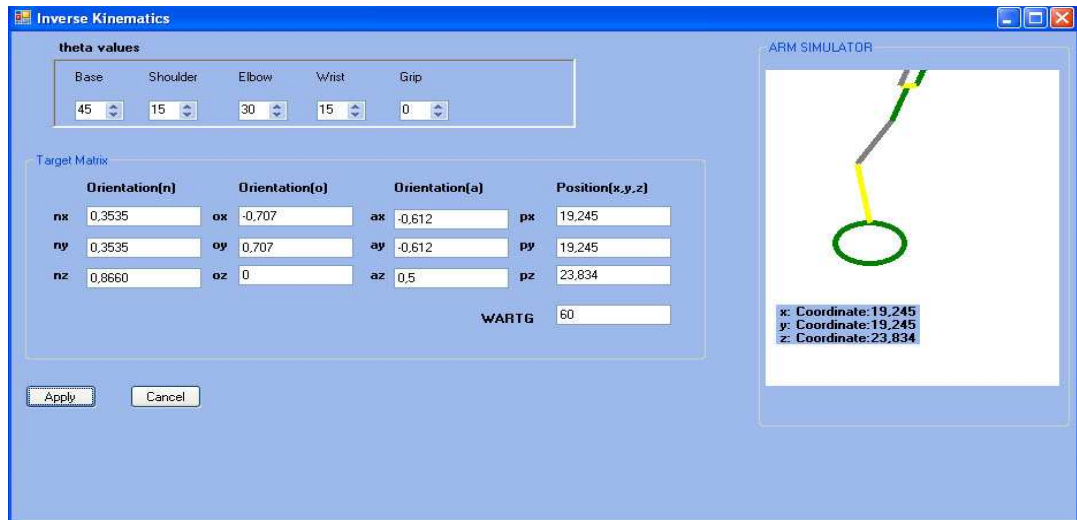
$$\theta_1 = \arctan(p_y/p_x) \text{ or } \theta_1 = \theta_1 + 180^\circ \quad (6)$$

$$\theta_2 = \arctan(S_2/C_2) \quad (7)$$

$$\theta_3 = \arctan(S_3/C_3) \quad (8)$$

$$\theta_4 = \theta_{234} - \theta_3 - \theta_4 \quad (9)$$

$$\theta_5 = \arctan(S_5/C_5) \quad (10)$$



Şekil 6.27 Ters kinematik ekranı

Ana Fonksiyon Ters Kinematik - Başlat

Her bir pozisyon ve yönelme değeri ekrandan alınır

Devam Et:

Yerel Değişkenlere İlk Değerler Atanır

Matris Değişkenlerine ilk Değerler Atanır

Kadar: Toplam Dönüşüm Matrisi Hazırlanana Kadar

Ters Kinematik-Hesapla(Tf:Matris ,Dof:Tam Sayı)

Fonksiyon Ters Kinematik-Hesapla (Tf:Matris ,Dof:Tam Sayı).

Yerel Değişkenlere İlk Değerler Atanır

Gelen DOF değerine göre

Eğer Dof = 2

İse (6,7) Kullan

Eğer Dof = 3

İse (6.7.8) Kullan

Eğer Dof = 4

İse (6,7,8,9) Kullan

Değilse

(6,7,8,9,10) Kullan

Şekil 6.28 Ters kinematik fonksiyonları

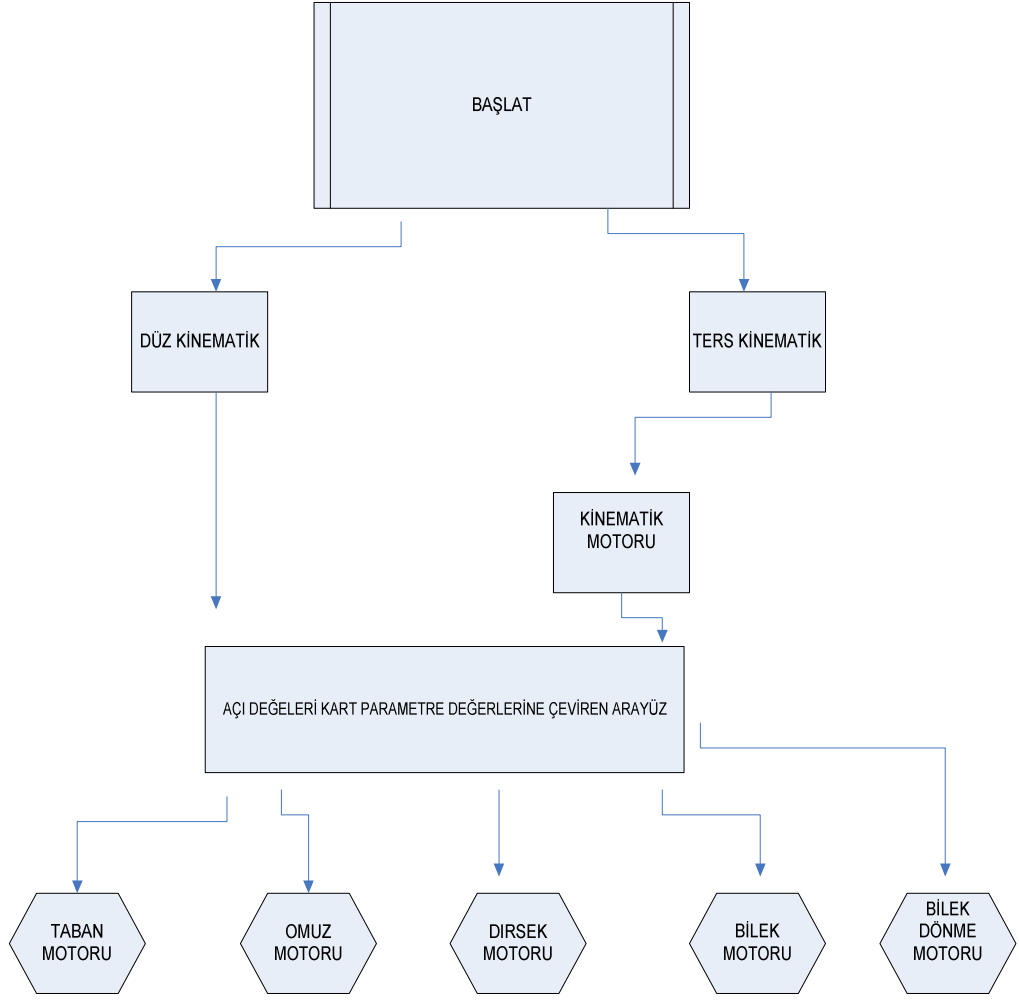
6.4 LYDS (Dinamik ve Kinematik Simülasyon Yazılımı)

Bu yazılım paketi kapsamında Lynx-6 Robot kolun her bir eklemine kontrol eden Doğru Akım servo motorların dinamik modelinin simülasyonu yapılmış ayrıca; Dinamik verilerin, hareketsel karakteristiğe çevrilmesi sayısal veriler grafikler ve simülatör

yardımıyla desteklenmiştir. Bu yazılım paketinin çalışma prensibi özetle, kullanıcı tarafından girilen sayısal pozisyon veya açı değerlerinin ilgili Doğru Akım motorları, kontrol kartı ve robotun fiziksel konumunu nasıl etkilediğinin aşamalarının simülasyonunun yapılmasıdır.

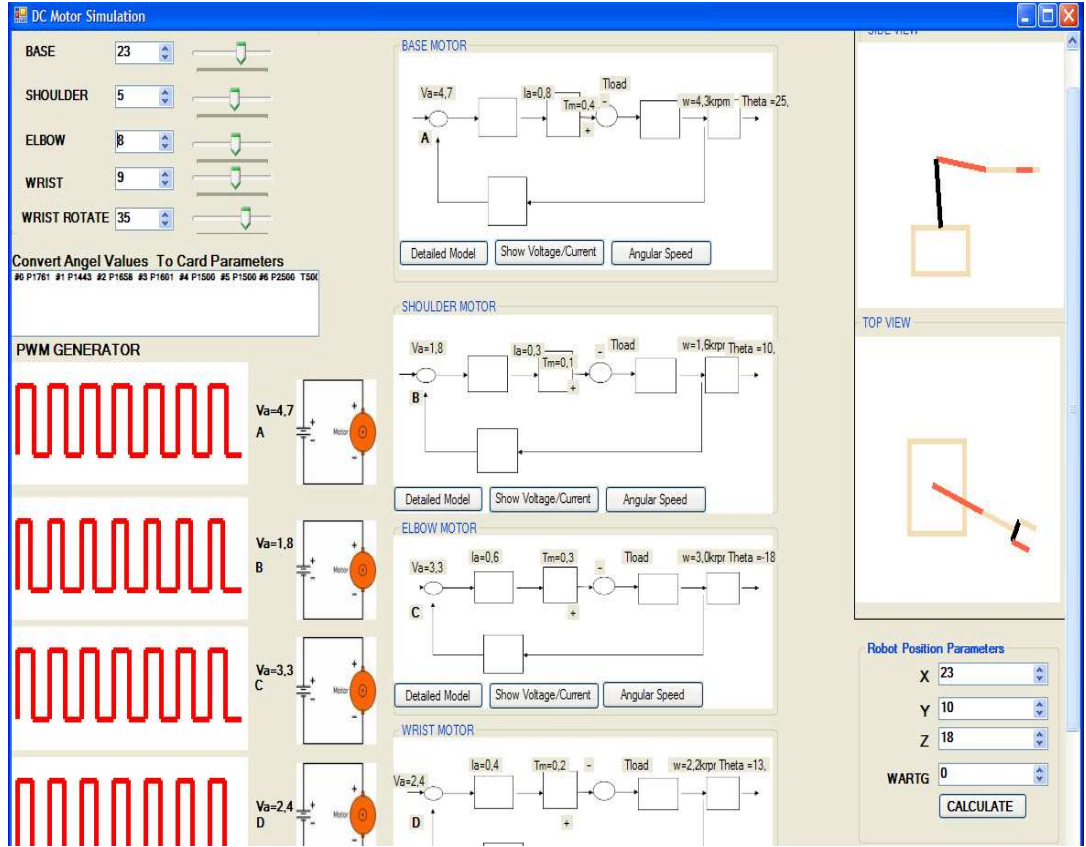
6.4.1 Yazılım ve uygulama içeriği

Bu tez kapsamında geliştirilen LYDS yazılım programı kullanıcı dostu hem eğitim hem de bilimsel çalışmalar için kullanılmak üzere geliştirilmiştir. Sistemin çalışmasının temel yapısının akış diyagramı şekil 6.29'da görülmektedir. Sisteme iki taraftan giriş sağlanabilir. Bunlardan ilki açı değerlerinin girilmesi (DÜZ KİNEMATİK YAKLAŞIM) , ikincisi ise pozisyon değerlerinin (TERS KİNEMATİK YAKLAŞIM) sisteme girilmesidir. Birinci tür bir veri girişi gerçekleştirildiği zaman girilen sayısal açı değeri geliştirilen ara yüz yardımıyla kontrol kartının anlayabileceği parametre değerlerine çevrilir. Buna karşın ikinci tür veri giriş gerçekleştirildiği zaman, Kinematik motoru devreye girerek öncelikle girilen pozisyon değerinden gerekli açısal değerler ters kinematik denklemler yardımıyla çıkarılır; çıkan bu açısal değerler ara yüz yardımıyla kontrol kartının anlayabileceği parametre derlerine çevrilirler. Dönüştürme ve ayarlama işlemi sayısal verinin motorun anlayabileceği analog değere çevrilmesi işlemidir. Her bir motorun çalışma adımları ekranda görsel olarak gösterilirken sonuçta toplam hareketin benzetim nesnesine etkisi de görülür. Böylece dinamik verilerin hareket karakteristiği üzerinde ki etkisi de test edilir.



Şekil 6.29 LYDS sistem akış diyagramı

Yazılımın şekil 6.30’da görülen Ana giriş ekranı kullanıcının kolay anlayabileceği bir biçimde geliştirilmiştir. Şekilde de görüldüğü gibi iki girişi olan bu sistem, Sayısal değerleri PWM kontrol metodu kullanarak dijital verilerden gerekli analog değerleri kontrol edip, motorun dinamik hareketliliği ve sonucunda pozisyon bulma ve yönelme işlemleri gerçekleştiriyor. Sistemin bütün mekanizması geliştirilen 4 adet ana motor üzerine inşa edilmiştir. Bunlar Kinematik Motor, PWM motoru Simülör Motorlarıdır ve Doğru Akım Motor Simülörüdür.



Şekil 6.30 LYDS ana veri girişi ekranı

Kinematik Motoru: Bu motor, Düz ve Ters Kinematik çözümlerin hesaplanması yani sistemin hareketsel karaktersizliğinin oluşturulmasından sorumlu olan kısımdır. Daha önceki kısımlarda çözümlerinden bahsedilen Lynx-6 Robot Kolu kinematik denklemleri bu motor kapsamında kullanılmaktadır.

PWM Motoru: Bu motorun yaptığı işlem esasen analog voltaj değerlerini sayısal değerleri kullanıp yönetmek ve düzenlemektir. Ayrıca sayısal açı değerleri ve Kontrol Kart Parametreleri arasında ki ilişkiyi yönetmek aralarında ki dönüşüm mekanizmasını sağlamaktır.

Simülasyon Motoru: Bu modül ve içerdiği alt fonksiyonlar Bundan önceki üç yazılım paketinde de anlatılan ve kullanılan LYNX-6 robot kolun, 2 boyutlu (2D) görsel simülasyon uygulamasını gerçekleştirmekten sorumludur.

Doğru Akım Motor Simülatörü: Bu modül her bir eklem için çalışmaktadır, ilişkili ekleme giriş verisi olarak gelen voltaj verisi ve sahip olduğu sabit motor sabitlerine göre doğru akım motorunun modelinin ve simülasyonunu gerçekleştirir. Bu motorun yazılımda ki karşılığı DC motor adlı bir fonksiyon ve sahip olduğu alt fonksiyonlar şekil 6.32’de görülmektedir. Bu fonksiyonda Kısım 5’de anlatılan Motor denklemlerinin çözüm yöntemleri kullanılır. Ayrıca motor davranışlarının görsel içeriğinin gösterildiği simülasyon ekranı ise Şekil 6.33’de görülmektedir, her bir eklem için yazılımda bu ekrandan birer tane mevcuttur.

LYDS yazılım sistemin bütün çalışma algoritması maddeler halinde aşağıda anlatılmaktadır.

- Program Çalışır
- Veri Düz Kinematik ekranından açı olarak girilirken, Ters Kinematik giriş ekranı kısmından pozisyon olarak girilir.
- Eğer girilen açı değerleri pozisyon ise Kinematik motor pozisyon değerlerinden açı değerlerini hesaplar
- Açı Değerleri Kontrol Kartı Parametrelerine çevrilir.
- PWM motoru sayısal verilere göre analog voltaj değerlerini düzenler
- Bütün aktif olan eklem motorları gelen voltaj değeri ve sabit motor parametre değerlerine göre dinamik bir biçimde çalışır
- Simülatör Motoru Lynx-6 Robot Kolun İki boyutlu (2D) modelini gelen giriş açılarına göre hareket ettirir.

Her bir Doğru Akım Motorunun kendi işlevsel bir ekranı vardır şekil 6.32’de bu ekran görülmektedir. Şekil 31 ’de ana algoritması gösterilen DC Motor fonksiyonu her bir ekranın simülasyon için kullandığı temel fonksiyondur. Bu ekran her gelen yeni voltaj değerine göre dinamik olarak sahip olduğu parametreleri değiştirir.

```

public DcMotor(double Vb,int id)
{
    Ra = 5;
    L = 2;
    Jm = 0.025;
    Bm = 0.1;
    kb = 0.1;
    Va = Vb;
    Id = id;
    Ki = 0.5;
    Tl = 0.0;
    Eb = 0;
}

public double VoltageCurrentG(double t) // ia(s) / (v(s) - e(s))
{
    return (1 / L) * Math.Pow(Math.E,-(Ra/L)*t);
}

public double CalculateCurrent(double t)
{
    double initialcurrent=0,m=0;
    double infinitivecurrent;
    infinitivecurrent = Va / Ra;

    m = infinitivecurrent + (initialcurrent - infinitivecurrent) * Math.Pow(Math.E, -((Ra / L) * (t)));
    return m;
}

public double CalculateAngularSpeed(double t)
{
    double initiSpeed = 0, m = 0;
    double infinitiveSpeed;

    infinitiveSpeed = GlobalA.TorqueMEL[this.Id] / Bm;
    m = infinitiveSpeed + (initiSpeed - infinitiveSpeed) * Math.Pow(Math.E, -((Bm / Jm) * (t)));
    return m;
}

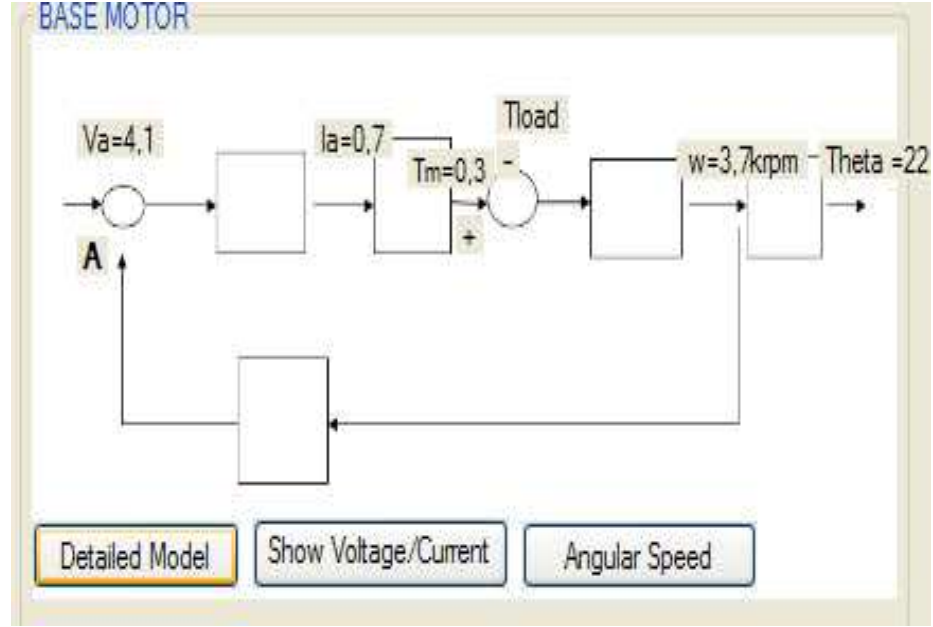
public double TorqueSpeedG(double t)
{
    return (1 / Jm) * Math.Pow(Math.E, -(Bm / Jm) * t);
}

public double CalculateTM(double ia) // calculate Torque
{
    return ia * Ki; // return Torque
}

// Calculate BackEMF
public void CalculateBackEmf(double w)
{
    Eb = w * Kb;
}

```

Şekil 6.31 Doğru akım motoru simülasyon ekranı kodu

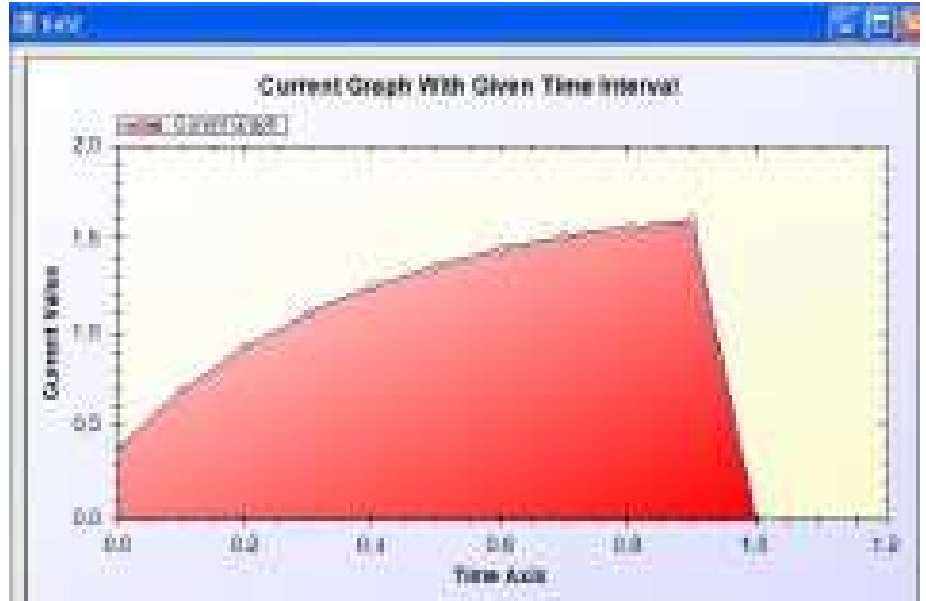


Şekil 6.32 Doğru akım motoru simülasyon ekranı

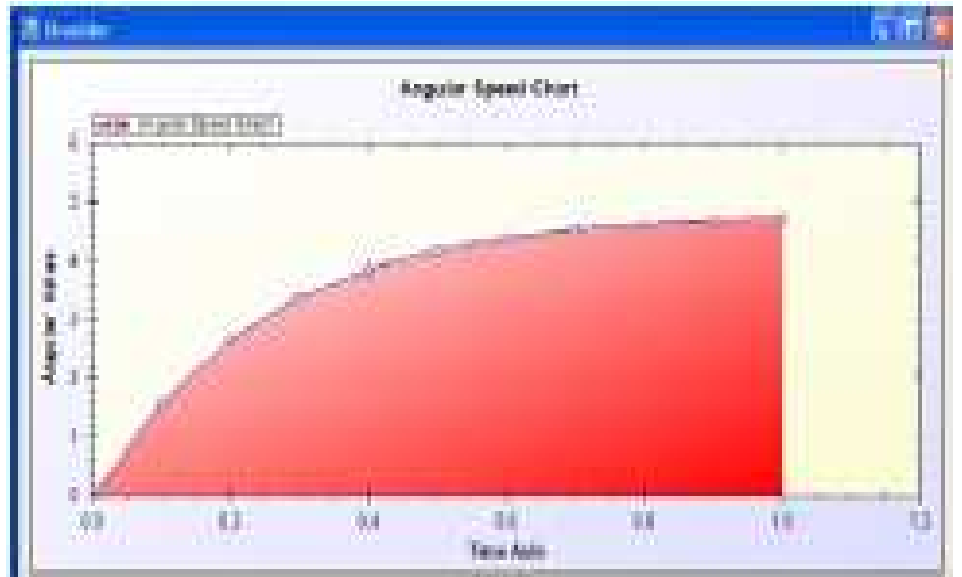
Bu ekran, Motor parametrelerinin değerlerinin göstermenin yanında iki boyutlu grafiklerle de sistemi ve olayın açıklanmasını kolaylaştırır, bu grafikler Akım - Zaman ve Açısal Hız – Zaman Grafikleridir. Bu grafiklerden birincisi Akım Zaman Grafiği, Bu grafik Akım değerinin verilen zaman aralığında Sabit Duruma (Steady State) ulaşmasını simüle ederken verilen zaman aralığında ki tepki değerleri gözlemlenir. Buna karşın ikinci grafikte verilen zaman aralığında Açısal hız değerinin (Steady State) ulaşmasını ve verilen zaman aralığındaki tepki değerleri gözlemlenir.

6.5 Uygulama Sonuç Kısmı

Bu kısım kapsamında tez için geliştirilen yazılımlar anlatılmış algoritmaları ve uygulama alanlarından detaylı bir biçimde bahsedilmiştir. Geliştirilen bütün yazılımlar hem bu tez kapsamında çalışılan konulara sanal bir test ortamı yaratırken, geliştirme algoritmalarında ki esnek yapı onların robot ve benzeri birçok bilimsel ve de eğitimsel konularda minör değişikliklerle kullanılabilmesine imkân sağlamıştır. Bu kısım ile ilgili uygulama ve test örnekleri bir sonra ki bölümünde incelenecektir. Ayrıca yazılımların bütün kod içeriği Ek 1’de mevcuttur.



a (Akım – Zaman)



b (Açısal Hız – Zaman)

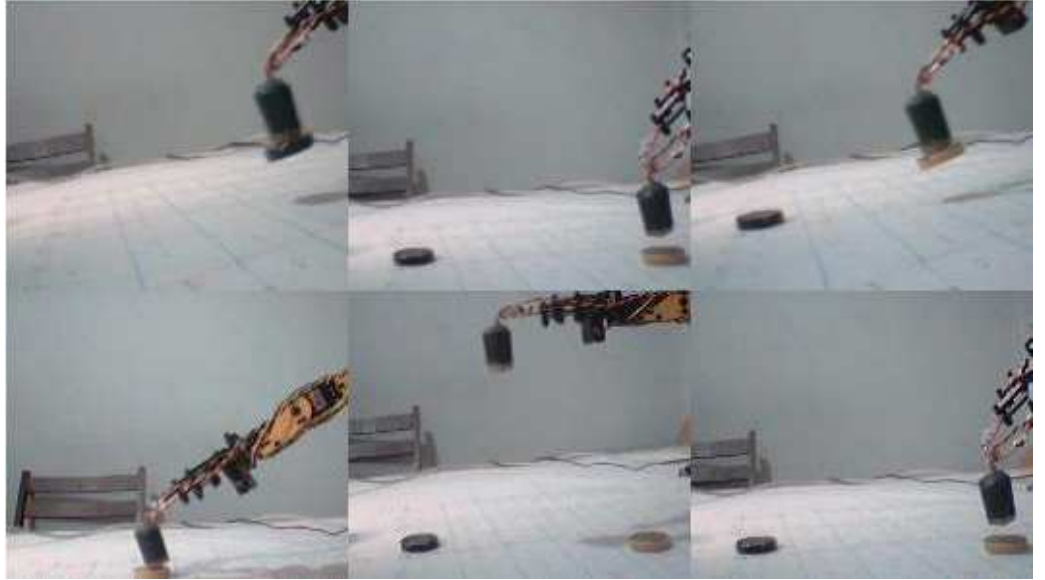
Şekil 6.33 Doğru akım motoru veri grafikleri

7. TEST VE UYGULAMA SONUÇLARI

Bu kısım tez kapsamında geliştirilen ve 6. kısımda anlatılan yazılım uygulamalarına örnekler ve bazı test sonuçlarını içerir.

7.1 Yame

Bu yazılım paketi, Lynx-6 Robot kolun fiziksel olarak yönetilmesi, hareketsel karakteristiğinin incelenmesi ve kinematik yeteneklerinin satranç tahtasında test edilmesi için tasarlanmıştır. Yazılım uygulamasının test edilmesi için, 80 cm × 80 cm 'lık Sarı Renkli bir tahta kullanıldı, tahtanın üzerinde ise 34 cm × 34 cm 'lük bir satranç tahtası çizilerek, robot son noktasının her bir oyun karesini ulaşması hem düz kinematik hem de ters kinematik çözümler denenerek test edilmiştir. Şekli 7.1'de satranç tahtası üzerinde ki bazı pozisyon ve hareketler görülmektedir. Robot kol tahta üzerinde her noktaya ulaşmaktadır. Çizelge 7.1 ve Çizelge 7.2'de oyun tahtası üzerinde ki her bir kareye ulaşılırken eklemlerin aldığı açılar iki değişik (Taş Alıp /Bırakma; Taş Taşıma) durumları için gösteriliyor.



Şekil 7.1 Satranç tahtası üzerinde hareketler

Yazılımın robot kola tahta üzerinde oyun taşlarını almak için yaptırdığı üç temel hareket vardır bunlar sırasıyla Başlangıç, Taş Alıp / Bırakma ve Taş Taşımadır. Bunlar sırasıyla Şekil 7.2'de görülmektedir.



Başlangıç

Taş Alıp / Bırkama

Taş Taşıma

Şekil 7.2 Sistemin üç temel hareketi

Çizelge 7.1 Her bir kareye ulaşmak için robot eklem açıların alması gereken değerler

Taş Taşıma Modunda Sırasıyla (Omuz , Taban , Dirsek , Bilek , WARTG)

	A	B	C	D	E	F	G	H
1	57,-56,-23,66,-20	86,-56,-52,59,-20	71,-45,-57,30,-30	88,-33,-92,5,-35	88,-33,-92,-5,-35	88,-42,-78,-45,-30	71,-45,-57,-60,-30	48,-43,-37,-67,-30
2	42,-52,-11,54,-20	66,-57,-31,45,-20	86,-56,-52,30,-20	87,-49,-65,11,-25	87,-49,-65,-12,-25	86,-56,-52,-31,-20	66,-57,-31,-45,-20	42,-52,-11,-55,-20
3	24,-44,-1,45,-20	42,-52,-11,35,-20	57,-56,-23,23,-20	66,-57,-31,8,-20	66,-57,-31,-9,-20	57,-56,-23,-24,-20	42,-52,-11,-36,-20	24,-44,-1,-45,-20
4	7,-32,4,37,-20	19,-41,1,29,-20	29,-47,-4,18,-20	35,-50,-7,6,-20	35,-50,-7,-7,-20	29,-47,-4,-19,-20	19,-41,1,-30,-20	7,-32,4,-38,-20
5	-9,-17,4,32,-20	0,-26,4,24,-20	7,-32,4,15,-20	11,-35,3,5,-20	11,-35,3,-6,-20	7,-32,4,-16,-20	0,-26,4,-25,-20	-9,-17,4,-33,-20
6	-24,3,-1,28,-20	-16,-7,2,21,-20	-11,-14,3,12,-20	-9,-17,4,4,-20	-9,-17,4,-5,-20	-11,-14,3,-13,-20	-16,-7,2,-22,-20	-24,3,-1,-29,-20
7	-41,30,-10,25,-20	-34,17,-5,18,-20	-29,10,-3,11,-20	-26,7,-2,3,-20	-26,7,-2,-4,-20	-29,10,-3,-12,-20	-34,17,-5,-19,-20	-41,30,-10,-26,-20
8	-60,64,-25,22,-20	-49,44,-16,16,-20	-44,34,-12,10,-20	-41,29,-10,3,-20	-41,29,-10,-4,-20	-44,34,-12,-11,-20	-49,44,-16,-17,-20	-60,64,-25,-23,-20

Çizelge 7.2 Her bir kareye ulaşmak için robot eklem açıların alması gereken değerler Taş

Alıp / Bırkama Modunda Sırasıyla (Omuz , Taban , Dirsek , Bilek , WARTG)

	A	B	C	D	E	F	G	H
1	39,-45,-36,66,-40	64,-50,-55,59,-40	85,-49,-78,45,-40	87,-42,-92,45,-45	87,-42,-92,45,-45	77,-45,-78,45,-45	57,-45,-58,60,-45	34,-39,-42,-67,-45
2	27,-40,-29,54,-40	47,-47,-41,45,-40	64,-50,-55,30,-40	74,-50,-66,11,-40	74,-50,-66,-12,-40	64,-50,-55,-31,-40	47,-47,-41,-45,-40	27,-40,-29,-55,-40
3	12,-30,-24,45,-40	27,-40,-29,35,-40	39,-45,-36,23,-40	47,-47,-41,8,-40	47,-47,-41,-9,-40	39,-45,-36,-24,-40	27,-40,-29,-36,-40	12,-30,-24,-45,-40
4	-3,-17,-22,37,-40	8,-27,-23,29,-40	17,-33,-25,18,-40	21,-37,-26,6,-40	21,-37,-26,-7,-40	17,-33,-25,-19,-40	8,-27,-23,-30,-40	-3,-17,-22,-38,-40
5	-18,1,-25,32,-40	-9,-10,-23,24,-40	-3,-17,-22,15,-40	1,-20,-22,5,-40	1,-20,-22,-6,-40	-3,-17,-22,-16,-40	-9,-10,-23,-25,-40	-18,1,-25,-33,-40
6	-34,25,-32,28,-40	-26,12,-28,21,-40	-20,5,-26,12,-40	-18,1,-25,4,-40	-18,1,-25,-5,-40	-20,5,-26,-13,-40	-26,12,-28,-22,-40	-34,25,-32,-29,-40
7	-59,67,-50,25,-40	-46,45,-40,18,-40	-40,34,-35,11,-40	-37,29,-34,3,-40	-37,29,-34,-4,-40	-40,34,-35,-12,-40	-46,45,-40,-19,-40	-59,67,-50,-26,-40
8	-70,68,-20,22,-20	-58,46,-9,16,-20	-53,36,-4,10,-20	-50,31,-2,3,-20	-50,31,-2,-4,-20	-53,36,-4,-11,-20	-58,46,-9,-17,-20	-70,68,-20,-23,-20

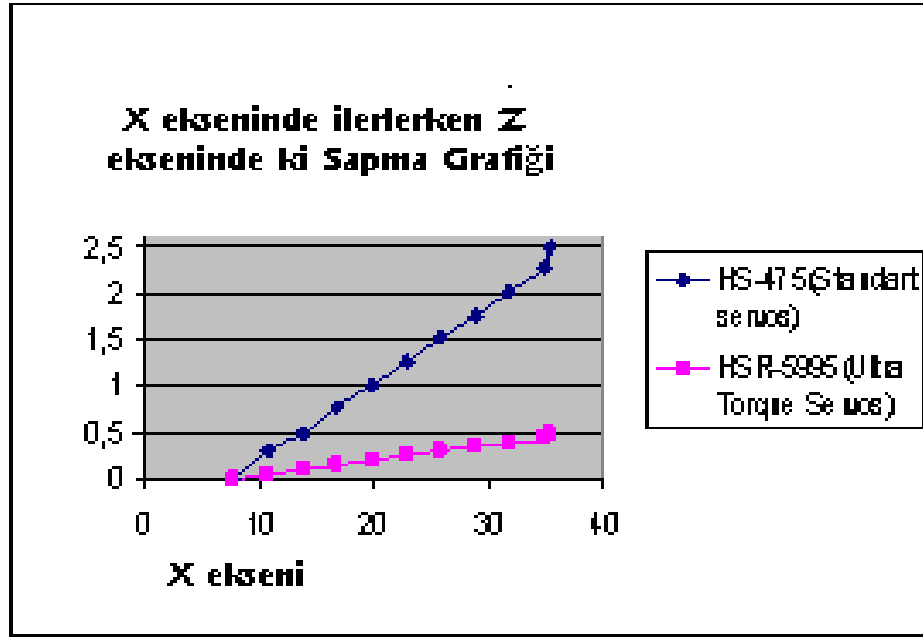
Sistemin yerçekimi etkisine karşın direncini ölçmek için çalışma uzayı olan satranç tahtasında önce x eksenini Yönünde 1'er santim; 1'er santim ilerlenmiş, bu sırada z yönünde ki konum sabit 6.8 cm olarak tutulmuştur. 30 cm lik bir hareket alanı içinde 2 ile 2,5 santimetre arasında sapma saptanmıştır. Bu sapma çok büyüktür ve istenilen pozisyondan bir bazen bir kaç kare kadar uzağa yanlış bir pozisyona konumlanılmasına sebep olmaktadır. Bu problemi çözmek için iki yöntem üzerinde çalışılmıştır. Çizelge

7.3’de Yerçekimi etkisi ile oluşan sapma değerleri omuz açısı cinsinden görülmektedir. Tabloda ki 2. kolon X pozisyonunda yerçekimi sonucu etkilenen omuz açı değerini temsil ederken 4. kolon (Z= 6.8) pozisyonunu sağlayan açı değerini temsil etmektedir. Bu yöntemde yapılan X yönünde ilerlerken Z eksenini boyunca oluşan sapmaya bağlı fark açı değerini hesaplayıp (5. kolon) X yönünde her bir santimetre ilerlemek için omuz açısına yazılım içinde hareket esnasında ekleyerek sapma probleminden kurtulmaktır.

Çizelge 7.3 Sapma Çizelgesi Omuz İçin

X	@	Z	OLMASI @	
7,31	30,828	6,8	32,5	1,672
8,31	30,636	6,8	31	0,364
9,31	27,817	6,8	29,2	1,383
10,31	24,953	6,8	26,5	1,547
11,31	23,949	6,8	25,5	1,551
12,31	20,814	6,8	22	1,186
13,31	17,624	6,8	19	1,376
14,31	17,18	6,8	18,6	1,42
15,31	13,527	6,8	15	1,473
16,31	9,8196	6,8	11,5	1,6804
17,31	11,102	6,8	12,8	1,698
18,31	7,55	6,8	9,3	1,75
19,31	3,9366	6,8	5,5	1,5634
20,31	3,2	6,8	5,2	2
21,31	-0,4	6,8	1,2	1,6
22,31	-4,09	6,8	-2,09	2
23,31	-6,06	6,8	-3,6	2,46
24,31	-10,05	6,8	-7,5	2,55
25,31	-14,1094	6,8	-11,2	2,9094
26,31	-15,45	6,8	-12,45	3
27,31	-19,5	6,8	-16,5	3
28,31	-23,6791	6,8	-20,3	3,3791
29,31	-25,99	6,8	-22,5	3,49
30,31	-30,05	6,8	-26	4,05
31,31	-34,2366	6,8	-29,7	4,5366
32,31	-38,89	6,8	-33	5,89
33,31	-44,11	6,8	-38,5	5,61
34,31	-49,7261	6,8	-43	6,7261

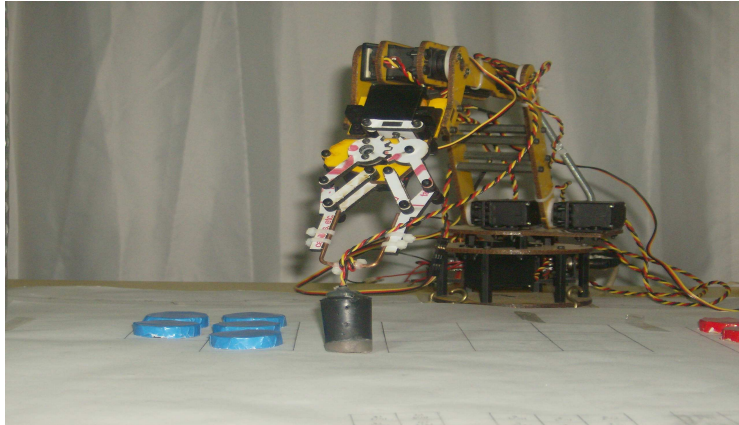
Bu çözüm iyi sonuç verse de hala bir, oyun tahtası veya robotun yeri değiştiği zaman Çizelge 7.3’deki hesaplamaların tekrar yapılması gerekmektedir ve yeni hesaplamalar göre düzenlemeler düzeltmeler yapılmalıdır. Bu yüzden sistemin alternatif bir çözüm olarak sistemin sahip olduğu Standart HS-475 Servo motorlar Daha güçlü tork veren Sayısal HSR-5995 ‘ler ile değiştirilmiş ve 2 ile 2.5 sm arası değişen sapmalar 0.2 ile 0.5 arası inmiş bu da tölere edilebilir bir aralıktır. Şekil Sonuçta maliyet olarak fazla olsa da bu yöntem en kesin ve en uygun çözümdür



Şekil 7.3 HS-475 ve HSR -5995 servo motorların performanslarının karşılaştırılması

7.2 Üç Taş Oynayan Robot

Yapay zekâ ve İmge işleme teknikleri kullanılarak Üç Taş oyununun gerçek zamanlı olarak robot kol ve insan arasında oynanmasını sağlayan ikinci yazılım paketi bu bölüm kapsamında test edilmiştir. Test kapsamında oyun yapay zekâsı ve görme sistemi test edilmiştir.



Şekil 7.4 Üç Taş oynayan robot kol

Üç taş oyunu için yapay zeka yöntemi olarak Min-Max algoritmasının kullanılması Dama ve Satranca göre seçenekleri kısıtlı olan bu oyunda mükemmel sonuç

vermektedir. Üç taş oyununda $9! = 362880$ farklı durum olabilir ki, diğer Tahta oyunları ile karşılaştırıldığında kolayca tolere edilebilecek bir sayıdır. Algoritmanın sistem üzerinde ki mükemmel sonucu, yapay zeka ile yönetilen Robot kolun insana hiç bir şekilde bu oyun boyunca mağlup olmamasıdır, insanın başarabileceği en iyi ihtimalle beraberliktir. Şekil 7.5'te insan ve yapay zeka arasında ki bazı oyun sonuçları gösterilmektedir. X insanın hamlelerini temsil ederken 0 yapay zekâyı temsil eder.

X	X	0
	0	X
0		

Yapay Zekâ Kazandı

X	X	0
0	0	X
X	X	0

Beraberlik

0	X	X
	0	
X		0

Yapay Zekâ Kazandı

0	0	0
X	X	0
X		X

Yapay Zekâ Kazandı

X	0	X
X	0	
0	0	X

Yapay Zekâ Kazandı

X	0	X
X	0	
0	0	X

Yapay Zekâ Kazandı

Şekil 7.5 Yapay zeka ve insan oyun sonuçları

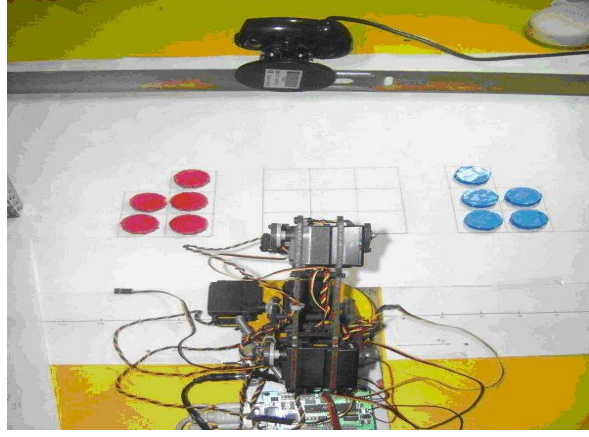
Sistem farklı oyun tecrübesine sahip iki denek ile test edilmiştir. Deneklerden birincisi Üçtaş oyununda tecrübeli, orta yaşa ulaşmış bir oyuncudur. Buna karşın ikinci denek

oyunu yeni öğrenen ve yaşı küçük olan bir oyuncudur. Test sonuçları Çizelge 7.4’de görülmektedir. Oyuncunun üçtaş oyununu iyi bilmesi veya oyun tecrübesi ona en fazla beraberlik sayısını artırma imkânı veriyor

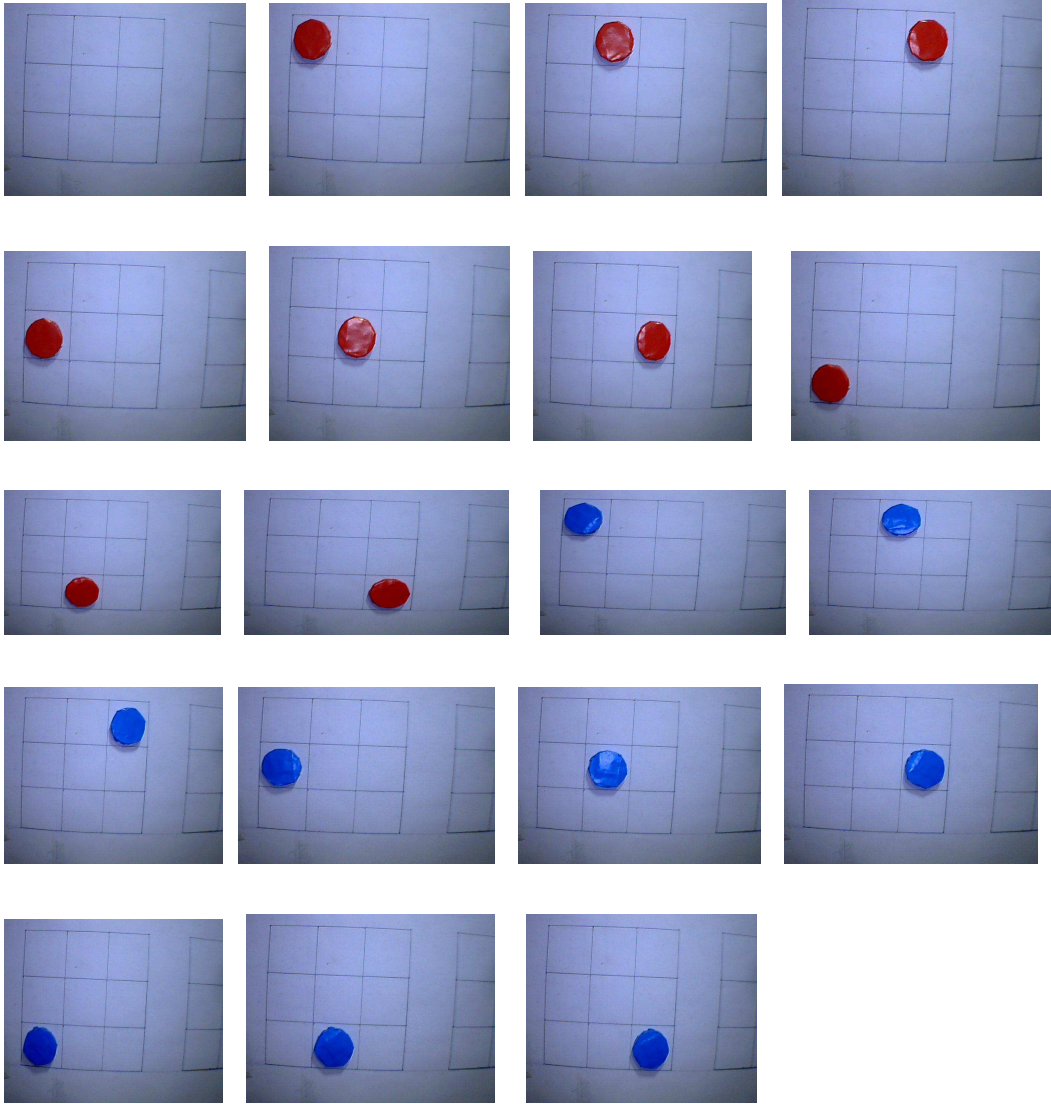
Çizelge 7.4 Yapay Zeka Motoru için Test (2 Denekle)

OYUN SAYISI: 15 , X (Yapay Zeka) , O (İnsan Zekası)			
X Kazandı	O Kazandı	Beraberlik	Yüzde (Yapay Zekâ Yönünden)
9	0	6	%60 X kazanır
13	0	2	%86 X kazanır

Oyunun Yapay Zekâsının yanında, oyunun görme mekanizması da test edilmiştir. Oyunda ki her bir taşın pozisyonu sisteme üstten bütünleşmiş olmuş bir web kamera ile kaydedilmiştir Şekil 7.6’da sistem görülmektedir. Görüntü veritabanı şekil 7.7’de görülmektedir.

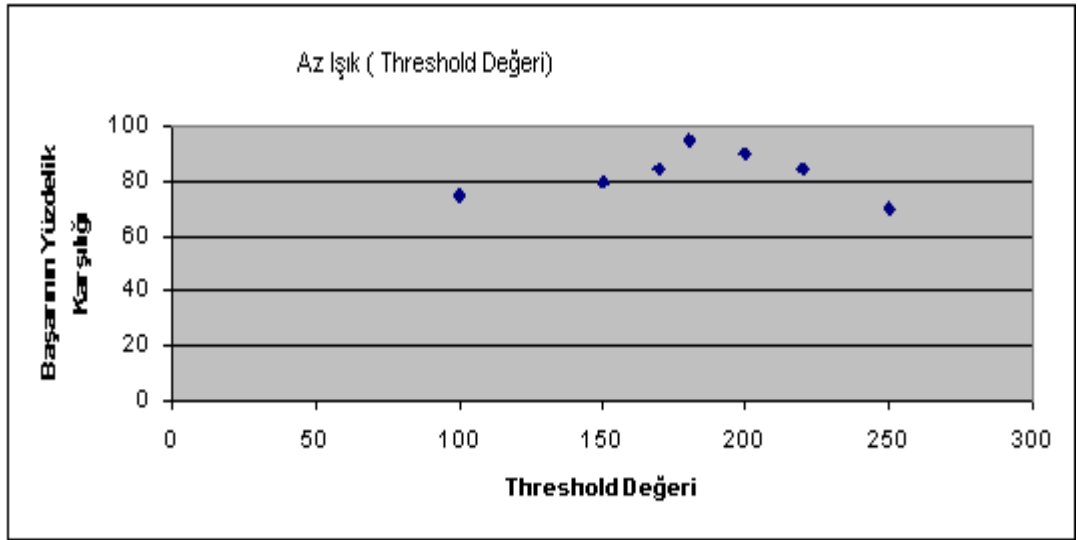


Şekil 7.6 Oyun sistemi ve görme mekanizması

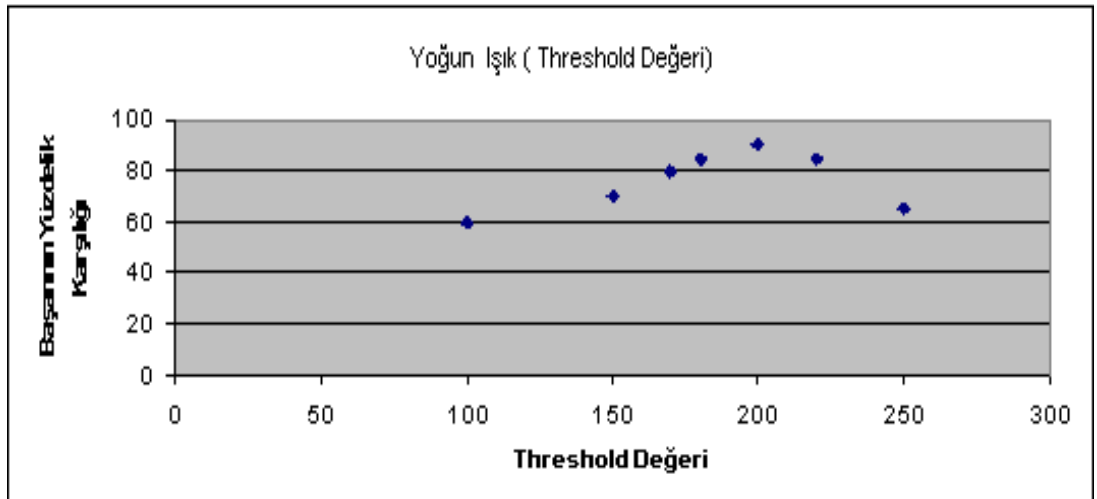


Şekil 7.7 Görüntü veritabanı

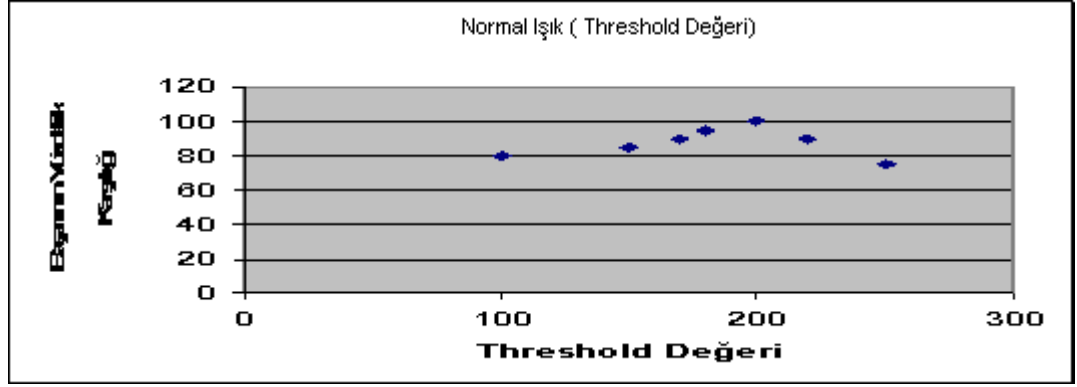
Sistemin doğruluğunu test etmek için değişik ışık yoğunluğu ortamlarında sistem test edilmiştir. Şekil 7.8’den Şekli 7.10’a kadar değişik ışık ortamlarında görme sistemi test edilmiştir. Şekil 7.8’de sistemin çok karanlık bir ortamda test edilmesi sonucu elde edilen Threshold değerlerinin başarı yüzdesi grafiği görülmektedir. Buna karşın Şekil 7.9’da çok yoğun bir ışık ortamında sistem test edilmiş en son olarak ışık yoğunluğunun normal olduğu bir ortamda sistem test edilmiş ve ideal Threshold Değerinin her türlü ortam için 200 olduğu bu test sonuçlarından gözlemlenmiştir.



Şekil 7.8 Ortam ışığının çok az olduğu zaman elde edilen grafik



Şekil 7.9 Ortam ışığının çok yoğun olduğu zaman elde edilen grafik



Şekil 7.10 Ortam ışığının normal olduğu zaman elde edilen grafik

7.3 Kinematik Simülasyon Yazılımı

Bu yazılım paketini incelerken bu kısımda bir örnek üzerinde yazılımın çalışma modeli gösteriliyor. Ayrıca yazılımın hesapladığı bazı pozisyon değerleri ile fiziksel robotun gerçek pozisyon değerleri karşılaştırılıyor. Robot Kolun Başlangıç Pozisyon Matrisi aşağıda ki matriste gösteriliyor.

$$\begin{bmatrix} 1 & 0 & 0 & 38,195 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Başlangıç matrisi yukarda görülen örnek için D-H parametreleri Çizelge 7.5’te veriliyor Buna göre yazılımın hesapladığı düz ve ters kinematik çözümler sırasıyla şekil 7.11 ve 7.12’de gösteriliyor.


Çizelge 7.5 Simülatör Örneği için D-H parametre Tablosu

Eklemler	θ degree	d sm	a sm	α degree
1	45	0	0	90
2	30	0	12.065	0
3	30	0	12.065	0
4	-45	0	14.249	-90
5	30	0	0	0
6	0	0	0	0

The image shows a software interface for kinematic simulation. It consists of six panels arranged in a 2x3 grid:

- Base:** A 4x4 grid of input fields. The first two rows contain the values 45, 0, 45, 0. The third row contains 0, 1, 0, 0. The fourth row contains 0, 0, 0, 1.
- Shoulder:** A 4x4 grid of input fields. The first two rows contain the values 30, 30, 0, 30. The third row contains 0, 0, 1, 0. The fourth row contains 0, 0, 0, 1.
- Elbow:** A 4x4 grid of input fields. The first two rows contain the values 30, 30, 0, 30. The third row contains 0, 0, 1, 0. The fourth row contains 0, 0, 0, 1.
- Wrist:** A 4x4 grid of input fields. The first two rows contain the values -45, 0, -45, -45. The third row contains 0, -1, 0, 0. The fourth row contains 0, 0, 0, 1.
- Grip:** A 4x4 grid of input fields. The first two rows contain the values 30, 30, 0, 0. The third row contains 0, 0, 1, 0. The fourth row contains 0, 0, 0, 0.
- Final Position:** A 4x4 grid of input fields. The first row contains 0,2379, -0,953, -0,183, 21,260. The second row contains 0,9450, 0,270, -0,183, 21,260. The third row contains 0,2241, -0,129, 0,9659, 20,121. The fourth row contains 0, 0, 0, 1.

Şekil 7.11 Düz kinematik çözüm ekranı


Inverse Kinematics

theta values

Base	Shoulder	Elbow	Wrist	Grip
45	30	30	-45	30

Target Matrix

	Orientation(n)	Orientation(o)	Orientation(a)	Position(x,y,z)
nx	0,2379	ox -0,953	ax -0,183	px 21,262
ny	0,9450	oy 0,270	ay -0,183	py 21,264
nz	0,2241	oz -0,129	az 0,9659	pz 20,121

WARTG 15

Şekil 7.12 Ters kinematik çözüm ekranı

7.4 LYDS (Dinamik ve Kinematik Simülasyon Yazılımı)

Bu yazılımı test etmek içinde bir örnek üzerinden gidilmiştir. Çizelge 7.6’da aşağıdaki örnek için motor sabit değer tablosu görülmektedir. Aşağıdaki örnek için 6.27’de bütün algoritma adımları işleniyor.

Çizelge 7.6 Ölçülen Değerler ile Yazılım Değerlerinin Karşılaştırılması

$R_a = 5$	Armatür Direnci
$L = 2$	endüktans
$J_m = 0.025$	motor sabiti
$B_m = 0.1$	sürtünme sabiti
$k_b = 0.1$	ters emf sabiti
$K_i = 0.5$	moment sanbit
$T_1 = 0.0$	Yükleme moment

Örnek için girilen pozisyon değerleri için $x = 23$, $y = 8$, $Z = 21$ ve $WARTG = 0$ olarak anılıyor . Giriş olarak verilen pozisyon değerlerinden Kinematik motorun çalışması sonucu açılma değerleri $TABAN = 20$, $OMUZ = 9$, $DİRSEK = -3$ ve $BİLEK = 35$ olarak hesaplanır.

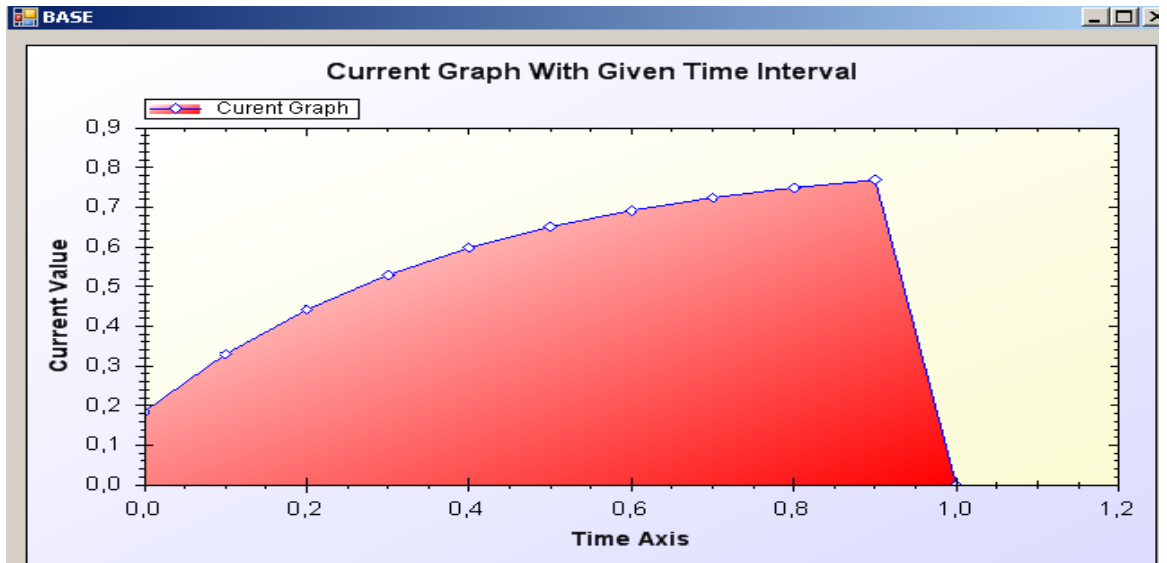
X	Y	Z	WARTG	TABAN	OMUZ	DİRSEK	BİLEK
23	8	21	0	20	9	-3	-6

Kinematik motordan hesaplanan açılma değerlerini kart parametrelerine çeviren ara yüz giriş devreye giriyor açılma değerlerini gerekli pulse değerlerine çeviriyor. Çevirme işleminde ise 1500 pulse değeri sıfır açılma değeri olarak alınıyor ve bu açılma değeri üzerine eklenen her bir derece açılma için pulse değeri için 11,1 artıyor. Çizelge 7.7’de, Hesaplanan pulse değerleri ve bunların karşılığı olan volt değerleri görülmektedir. Yapılan test çalışmalarında bazı örneklerde kinematik motorun hesapladığı açılma değerleri ile, doğru akım motoru simülatör değerlerinin hesapladığı açılma değerler arasında (1 - 4) derece arası farklar olmaktadır.

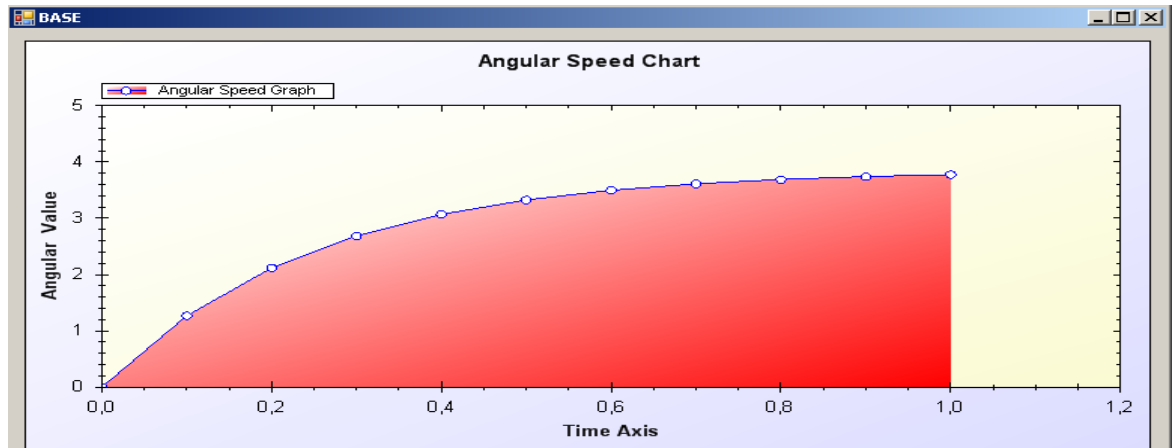
Çizelge 7.7 Çevirici Ara yüz Sonuç Değerleri

Hesaplanan Pulse Değerleri			
TABAN	OMUZ	DİRSEK	BİLEK
1713 ms	1400 ms	1438 ms	1538 ms
Hesaplanan Volt Değerleri			
TABAN	OMUZ	DİRSEK	BİLEK
4.1 V	2.5 V	1.4 V	1.8 V

Şekil 7.13’de Taban Ekleminin simülatörünün desteklediği Akımın sabit duruma ulaşma grafiği (Steady State) ve Açısal Hızın zamanla sabit duruma ulaşma grafiği (Steady State) grafiklerine bir örnek yukarıda ki örnekle ilişkili olarak sırasıyla şekil 7.13 ve 7.14 de görülmektedirler. Her eklemin motor simülasyon ekranı bu değişim grafiklerini desteklemektedir. Verilen zaman aralıklarında akım veya açısal hız değerlerinin Sabit duruma ulaşması (Steady- State) simüle ediliyor.



Şekil 7.13 Taban ekleminin açısı – zaman grafiği (23 Derece)



8. SONUÇ KISMI

Bu kısımda tezin amacı ve sonucu tartışılmış, bölümlerin kısa birer özetinden bahsedilmiş ayrıca gelecekte yapılacak çalışmalar için tavsiye ve yöntemlerden bahsedilmiştir.

8.1 Tezin Amacı

Bu Tezin Amacı Beş ve Altı Eksenli Robot kolların matematiksel analizinin yapılması, hareketsel karaktersitliğinin incelenmesidir. Teorik çalışmanın sonuçlarının görsel tabanlı bilgisayar yazılımları ile uygulama olarak test edilmesi; çalışılan teori ve algoritmik bilginin gerçek ve simülasyon uygulamalarında kullanılması hedeflenmiştir. Tez kapsamında robot kol modeli olarak Lynx-6 robot kol seçilmiştir. Bu kolun hem teorik çalışma esnasında hareketsel karakteristiğinin incelenmesi hem de pratik çalışma kapsamında gerçek zamanlı ve akıllı uygulamalar için kullanılması amaçlanmıştır.

8.2 Tezin Sonucu

Bu Tez çalışması kapsamında hedeflenen teorik çalışmalar gerçekleştirilmiştir. Beş ve Altı serbestlik derecesine sahip robot kolların hareketsel karaktersizliği ve kinematik çözümleri incelenmiştir. Tez kapsamında incelenip çalışılan Altı Serbestlik Derecesine (6 DOF) sahip robot kol bütün eklemeleri döner eklem olan temel model olarak kabul edilen ve insan kolunun hareket karakteristiği ile büyük benzerlik gösteren Mafsallı (Articulate) bir Robot Koldur. Bunun ile beraber tez kapsamında incelenen Beş Serbestlik Derecesine (5 DOF) sahip robot kol ise esnek yapısı ve piyasada bu tür akademik ve araştırma çalışmaları için yegâne tercih edilen Lynx Firmasına ait Lynx-6 Robot Koldur. Kinematik denklemlerin elde edilmesi için Denavit Hartenberg (D-H) yöntemi ile robot kollar modellenip her bir eklem için dönüşüm matrisleri oluşturulmuştur. Bu dönüşüm kolay gibi görünse de her bir eklem için koordinat çerçevesinin yerleştirilmesi ve birbirlerine göre durumlarının yorumlanması karmaşaya yol açmaktadır. Komşu bağlar arasında genel bir eklem eksenini vardır. Altı eklemlili bu Robot'ta kinematiğinin sabit kısmını tamamen tanımlamak için onsekiz parametre gereklidir. Buna karşın Beş Eklemlili Lynx-6 Robot kolun kinematiğinin sabit kısmını tanımlamak için onbeş parametre yeterlidir. Bunlar hesaplanıp Beş ve Altı eksenli

robotlar için Denavit Hartenberg (D-H) parametre tablosu oluşturulmuş ve bu parametre tablosuna göre her bir robot eklemının dönüşüm matrisleri hesaplanmıştır (Niku 2001). Dönüşüm matrisleri birbiriyle çarpılarak Düz Kinematik Denklemler her iki robot kol için hesaplanmıştır.

Altı Serbestlik Dercesine sahip Mafsallı robot kol ile Beş serbestlik derecesine sahip Lynx-6 robot kol için bu tez çalışmasında esnek ve anlaşılır bir kinematik çözüm yöntemi geliştirilmiştir. Kollar için geliştirilen kinematik çözüm yöntemi yukarıda da bahsedildiği gibi; Robot Kolların Denavit Hartenberg (D-H) yöntemiyle modellenip her bir eklem için dönüşüm matrisleri oluşturulması ve bu matrislerden aritmetik yöntemler kullanılarak Düz Kinematik Denklemlerin (Forward Kinematic Equations) sonuçlarının çıkarılmasıdır. Buna karşın Ters kinematik problemlerin çözümünde, geometrik cebirin klasik eleme teorisinden yararlanılmıştır. Bu yöntem diyalektik eleme de denir Diyalektik eleme ile önce problem, sadece bir eklem açısını içeren tek değişkenli bir polinoma indirgenir. Bu polinom çözülerek eklem açılarından biri elde edilir. Bulunan açı değeri problemde yerine konulur ve diğer eklem açıları bulunana kadar aynı işlem tekrarlanır. Diyalektik eleme ile hesaplanamayan ters kinematik problemler de bulunmaktadır. Günümüzde, ters kinematik problemin çözümü için, analitik ve nümerik pek çok yöntem geliştirilmesine rağmen, genel bir yöntem yoktur. Her robot için farklı bir yöntem uygulanmaktadır. Robotun ileri kinematik haritası için exponansiyellerin çarpımını (product of exponentials) kullanarak, ters kinematik problemlerin çözümüne geometrik bir algoritma geliştirmek de mümkündür. Ters kinematik problem, uygun alt problemlere indirgenerek çözülebilir. Bu yöntem, ilk kez Paden (1986) tarafından sunulmuş ve Kahan'ın (1983) yılında ki çalışması üzerine inşa edilmiştir. Bu çalışma için kullanılan robot kollar eklem yapısı ve özgürlük derecesi (DOF) bakımından birbirlerine çok benzedikleri için ortak bir yöntem ikisi içinde uygulanmıştır. Bu yöntem ise Niku'nun 2001'de kullandığı Sonuç Dönüşüm Matrisinin eklem matrislerin tersleri ile sırasıyla çarpılması ve hedeflenen pozisyon ve yönelmeyi sağlayacak açı değerlerinin çıkarılmasıdır. Bu yöntem her iki robot kol içinde sonuç vermek ile beraber ters kinematik çözümlerde ki temel sıkıntı olan Özgürlük derecesinin Üç'ten ($DOF > 3$) değerinden büyük olduğu uygulamalarda birden çok çözüm kümesin ortaya çıkması bu çalışma esnasında da yaşanmıştır. Bu yüzden geliştirilen ters kinematik çözümler GAY'ın (2007) yılında Üç serbestlik derecesine sahip Lynx robot kollar için

uyguladığı, Paden (1986) ve Kahan (1983)'dan esinlenerek ortaya çıkardığı ters kinematik çözümler için kullandığı kısıtlama yöntemi WARTG sayısal değerinin (Dirsek + Omuz + Bilek açısı değerlerinin) önceden parametre olarak pozisyon değerleri ile beraber sisteme giriş olarak verilemesi ile genişletilmiştir. Tez kapsamında geliştirilen ters kinematik uygulamalarda kullanılmış, çoklu çözüm kümesi problemi bu yöntemle çözümlenmiştir. Sonuç olarak ters kinematik denklemler için tek (unique) çözüm kümesine ulaşılmıştır. Mafsallı robot kol için birden çok ters kinematik yöntem geliştirilmiştir. Örnek olarak Raghavan ve Roth çözümünü içeren Manocha ve Canny (1992) tarafından geliştirdiği yöntemdir. Bu yöntem temel olarak Altı Serbestlik derecesine (6 DOF) sahip ve uzayda 16 farklı yönelme yapılandırmasına sahip robot sistemini tek (unique) çözüme indirmek için öz vektörle (eigen-vectors) ve substitution (yerine koyma) metotlarını içeren karmaşık ve güçlü bir çözüm sistemidir. Özgüren'in (2001) geliştirdiği başka bir yöntem ise analitik çözümlerin olmadığı durumlarda yarı analitik (semi-analytical) yöntemlerin kullanılması ve üstsel(dönüşüm matrisleri exponential rotation matrices) yöntemlerini içerir. Bu iki yöntem Altı Eksenli standart robot kollar için bilinen ve uygulanan standart yöntemlerdir. Ayrıca matrislerin tersinin çarpımı (matrixes inverse multiply) yöntemi Xie (2005) tarafından ters dönüşüm ve geometrik metotlara alternatif olarak kullanılmış çözüm kümesi 16 farklı yönelme yapılandırmasından 8'e indirgenmiştir. Tez kapsamında geliştirilen ters kinematik yöntem ise bu bakış açısına bir alternatif olarak sunulmaktadır. Geliştirilen sistemin en büyük avantajı ve diğerlerinden farkı ise yöntemin basit olması ve özellikle yazılım uygulamalarında kolayca kodlanabilmesidir. Çözüm Denklemlerine ulaşmak için yerine koyma ve yok etme metotları gibi basit matematiksel yöntemler ile temel matris işlemleri yeterlidir. Denklemlerin basit ve anlaşılabilir yapısı ve yazılım uygulamaların da kullanılmaya yatkın oluşu geliştirilen çözüm kümesinin sistemler için iyi bir alternatif olduğunun göstermektedir. Beş Eksenli Lynx-6 Robot kol için geliştirilen Kinematik Çözüm Kümesi, Cebirsel yöntemlerin kullanıldığı Lynx marka robot kollar için ilk Kinematik çözüm çalışmasıdır. Bundan önce ki çalışmalarda Gay (2007) ve Hong (2005)'in sırasıyla Üç ve Beş Serbestlik Derecesine (DOF) sahip robotlar için geometrik yöntemler kullanarak kinematik çözüm denklemleri elde etmişlerdir. Gay'in (2007) çözüm denklemleri sadece Üç serbestlik derecesine sahip olan (3 DOF) Lynx robot kollar için çözüm üretirken, Hong'un (2005) geliştirdiği çözüm yöntemde ise

Üçgenin Cosinus Kuralı (Cosine Rule of Triangle) kullanılarak geometrik yaklaşımla çözüm elde edilmiştir. Uygulama kısmında geliştirilen dört farklı yazılım ise teorik çalışmayı görsel bir zeminde test etme ve uygulama fırsatı vermiştir. Lynx-6 robot kolun fiziksel olarak kullanılması için iki farklı yazılım geliştirilmiştir. Bunların birincisi robot kolu gerçek zamanlı olarak kontrol ederken kinematik çözümlerini test ediyor. Bu test kapsamında Lynx-6 robot kolun hareketsetel karakteristiğı satranç tahtası üzerinde kontrol ediliyor. Buna karşı ikinci geliştirilen uygulama ile yapay zekâ ve kamera desteğıyle Lynx-6 robot kolun güncel bir uygulamada kullanılması test edilmiştir. Bu uygulamanın kapsamında Lynx-6 Robot kola insan ile Üç-Taş oyunu (Tic-Tac-Toe) oynatılıyor. Geliştirilen sistemin en temel unsurlarından biri kullanılan yapay zekâ yöntemidir. Yapay Sinir Ağları yöntemi de bu oyun için çok tercih edilen bir çözüm metodudur. Ama seçeneklerin 9! olduğı böylesi kısıtlı bir oyunda yapay sinir ağları yöntemlerinin kullanılması masraflı olmaktadır. Siegel'in (2001) yılında üçtaş oyunu için kullandığı Takviyeli Öğrenme (Reinforcement Learning) yöntemi bile en iyi sonuç olarak %94 başarı oranını yakalamıştır. Sungur ve Halici'nin (1992) yılında, Üç değişik Yapay Sinir Ağları yöntemiyle (Hopfield Ağı Boltzman Makinası ve Gaussian Makinesi) oyunun sisteme öğretilmesine çalışmışlardır. Sonuçta en uygun sonucu elde etmelerine karşın; özellikle Boltzman ve Gaussian Makinelerinde öğrenme aşamasında çok fazla tekrar olması böyle basit bir oyun için çok fazla masraf anlamına gelmektedir. Bu nedenle böyle bir oyunda en az masraflı ve %100 başarıyı yakalayan Min-Max algoritması kullanılmıştır. Min-Max algoritması sistemin Yapay Zekâsının motorudur. Oyunda ki her bir pozisyon ağacın bir dalında tutulur ve her bir dala bir fayda (Utility) değeri atanır. Bütün pozisyonlar ağaçta tutulduğı için sistemin herhangi bir hamle karşısında en iyi hamleyi yapması ikili ağaçta (Binary Tree) arama yapmasından ibarettir, bu nedenle sistemin oyunu kaybetmesi imkânsızdır. Kullanılan Min-Max Yapay zekâ yönteminden dolayı insanın oyunu kazanma şansı sıfıra indirgeniyor. İnsanın en fazla kazanacağı başarı beraberliktir. Değişik yetenek ve yaş guruplarında ki insanlarla sistem karşılaştırılıyor ve üst üste oynanan oyun sayısı artıkça insanın dikkatinin dağıldığı ve robot sisteminin üst üste galibiyetler aldığı gözlemleniyor. Görüntü isleme ve nesne tanımlama günümüzün en önemli araştırma konulardan biridir. Çeşitli uygulama alanları için değişik görüntü isleme ve nesne tanımlama yöntemleri bulunmaktadır. Görüntü isleme ve görüntüdeki nesneleri

tanımlama, daha uzun süreli araştırma ve uzmanlık gerektiren bir çalışma alanıdır. Bu çalışmada, bilinen yaygın yöntemlere nazaran daha basit ve çıkarma (subtraction) işlemine dayalı bir yöntem kullanılmıştır. Bu yöntem istenilen verimliliği sağlamakla beraber, oyun tahtasının ve robotun yerinin değişmesi durumunda görme sisteminin tekrar kurulması yani taşların tekrar yeni duruma göre pozisyonlarının çekilip sisteme kayıt edilmeleri gerekmektedir. Buna karşın görme sistemi her türlü ışık yoğunlu değişimine karşın sıfıra yakın hata değeri ile performans göstermektedir. Üçüncü ve Dördüncü uygulamalar simülasyon tabanlı matematiksel uygulama yazılımlarıdır. Bu yazılımlar için fiziksel bir robota ihtiyaç yoktur. Geliştirilen Üçüncü yazılım paketi, Lynx robot kollar için matematiksel bir kinematik çözüm paketi içeriyor. Bu yazılım paketi kapsamında Teori kısmında geliştirilen kinematik denklemler görsel bir biçimde test ediliyor ayrıca bu yazılım sadece Lynx-6 için değil piyasada ki her türlü Lynx marka robot kolun kinematik çözümüne destek veriyor. Son yazılım ise Robot eklemlerinde ki Doğru akım servo motorlarının çalışma prensibinin hareketsetel karaktersizliği ile eşleşmesinin gösteren bir uygulama yazılımıdır. Bu yazılım temelde Doğru Akım (DC) motorlarının hareketsetel karakteristiği üzerinde ki etkisini ve sonuçta bir robot kolun her bir hareket için geçirdiği aşamaları simülasyon desteği ile gösteriyor.

8.3 Bölümlerin Amacı ve Özeti

Bu Tez Çalışması 7 Bölümden oluşur. Tezin Birinci kısmı giriş ve özet bölümünden oluşur. Sonraki kısımda temel robot bilgisi, robot sınıfları ve robot teori bilgisi için gerekli matematiksel yöntemler anlatılıyor. Üçüncü kısım projede kullanılan Lynx robot kolunun özellikleri limitleri, Doğru Akım (DC) Servo motorları ve SSC-32 servo kontrol kartı anlatılıyor. Dördüncü kısımda robot kolların pozisyon analizi ve kinematik çözümler anlatılmıştır. Matris gösterimi, dönüşüm matrisleri eklemleri modelleme yöntemleri, dönüşüm yöntemleri ve düz ters kinematik çözümler anlatılıp hedeflenen robot kollar için düz ve ters kinematik çözümler elde edilmiştir. Beşinci kısımda Doğru Akım (DC) motorlar anlatılmıştır. Dinamik modelleri elektriksel ve mekaniksel denklemler çıkarılmıştır. Ayrıca motorların kontrol yöntemleri anlatılmıştır. Altıncı kısımda tez çalışması kapsamında geliştirilen yazılımlar anlatılmıştır. Kullanılan algoritmalar yöntemler bu bölümde bulunmaktadır. Uygulama kapsamında, Visual Studio.Net platformunda C# programlama dili kullanılarak dört farklı görsel tabanlı

yazılım programı gerçekleştirilmiştir. Bu yazılımlardan ilki Lynx-6 robot kolunun gerçek zamanlı olarak yönetilmesi ve kinematik hesaplamalarını içerirken, İkinci yazılım paketi kamera desteği ve yapay zekâ teknikleri içeren bir uygulamadır, sistem yardımıyla Üç-Taş oyunu (Tic-Tac-Toe) insan ve Lynx-6 robot kol arasında oynatılıyor. Geliştirilen üçüncü yazılımım paketi Lynx robot kollar için kinematik çözüm paketi içeren bir matematiksel simülasyon yazılım paketidir. Son yazılım uygulaması ise robot eklemlerini kontrol eden Doğru Akım (DC) motorlarının kontrolünü, dinamik verilerin, hareketssel karakteristiğe çevrilmesi sayısal veriler grafikler ve simulator yardımıyla gösterilmesi için tasarlanmış dinamik ve kinematik analizin ortak olarak kullanılabildiği bir uygulamadır. Yedinci kısımda ise bir önceki kısımda geliştirilen yazılımların testleri ve bazı uygulama sonuçları gösterilmiştir.

8.4 Gelecekteki Çalışmalar için Tavsiyeler

Bu tez kapsamında geliştirilen, teorik ve uygulama çalışmaları ileride birçok Çalışmaya hem teorik hem pratik anlamda alt yapı oluşturacak esnek yapıya sahiptirler. Lynx-6 robot kollar için geliştirilen kinematik yöntem herhangi bir dış parametre (WARTG) kurtarılıp sadece sonuç matrisi kullanılarak tek (unique) çözüm kümesi elde edilmesi ileride yapılması tavsiye edilen teorik çalışmalarından biridir. Hesaplanan veriler bu yazılım ile robot üzerinde denendiğinde, robotun bazı hareketleri tam olarak gerçekleştiremediği gözlenmiştir. Meydana gelen hatanın sebebi, Lynx-6 eklemlerindeki hareketlendiricilerin, sınırlı dönme ($\pm 90^\circ$) sağlayabilmesidir. Bu değerler dışındaki eklem açılarının gerçekleştirilmesi istendiğinde motorlar zorlanmaktadır. Ayrıca hareketlendiriciler üzerindeki yük miktarı da robot hareketini kısıtlayıcı bir etken olmuştur. Bu çalışmanın devamında, robotun eklem açılarını genişletmek için eklemlerdeki hareketlendiricilerin sağladıkları açılar arttırabilir. Eklem açıları hesaplandıktan sonra, robot verilen görevi gerçekleştirirken, hareketinin titreşimsiz ve düzgün olabilmesi için yörünge planlaması yapılmalıdır Yörünge planlaması yapılırken; pozisyonda, hızda ve ivmede süreklilik sağlamak için, beşinci dereceden polinomlar kullanılabilir Geliştirilen yazılım paketleri Visual Studio.Net ortamında geliştirilmiş ve dolayısıyla sadece Windows platformlarında çalışmaktadır. Geliştirilen yazılımların platformdan bağımsız Java gibi bir yazılım ortamına aktarılması, geliştirilen yazılımların sadece Windows sistemlerinde değil bir çok değişik sistemlerde (Linux ,

Machintos) kullanılmasına olanak sağlar. Geliştirilen uygulama yazılımlar birçok araştırma ve eğitim uygulamalarına altyapı oluşturacak şekilde kullanılabilirler. Örnek olarak, geliştirilen YAME yazılımı Lynx-6 Robot kolun hareketsel karaktersizliğini tam olarak sağladığı için robot kolun kullanıldığı birçok uygulama da hem eğitim hem de altyapı sistemi olarak kullanılabilir. Ayrıca geliştirilen akıllı uygulamada ki sistem farklı bir yapay zekâ motoru ile ve görme sisteminde ki küçük değişiklikler ile daha karmaşık oyunlar için kullanılabilir (Satranç, Dama vs) . Üçüncü geliştirilen kinematik çözüm paketi sadece Lynx marka robot kolları değil de piyasada ki her türlü aynı ölçekli manipülatörleri destekleyecek biçimde eğitim seti olarak geliştirilebilir (Edubotlar ve Lego Robotlar) . Son geliştirilen yazılım paketi daha karmaşık modüller ile daha etkili bir matematik, fizik motoru ile ve daha güçlü bir grafik sistemi ile LabView ve Matlab gibi profesyonel yazılımlara alternatif bir ücretsiz bir sistem olarak ortaya konulabilir.

Yeni bir uygulama yazılımı geliştirilerek Üç-Taş oyunu (Tic-Tac-Toe) web üzerinden oynatılabilir, robotun bulunduğu ortamda internet sunucusu olan bir makineye uzaktan bağlanan istemciler internet üzerinden fiziksel robot ile oyun oynayabilir ve de onu başka görevler içinde kullanabilirler. Böyle bir sistem kurulursa internet üzerinden yönetilen manipülatörler hem eğitim amaçlı hem de bilimsel çalışmalar için kullanılabilir. Uygulamalarda kullanılan Lynx-6 robot kolun simülasyonunu sağlayan simülatör yazılımı, Robot kolun 2 Boyutlu (2-D) modelini oluşturmaktadır, yeni geliştirilecek çalışmalarda Robotun 3 Boyutlu (3-D) modelinin oluşturulduğu bir simülatör yazılımı hem görselliğe katkı sağlar hem de robot hareketlerinin daha iyi anlaşılmasını sağlar. Özellikle tutacın yönelme hareketleri 3 Boyutlu bir modelle çok daha iyi gözlemlenebilir.

KAYNAKLAR

Anonymous. 2007. Web Sitesi: www.lynxmotion.com, Eriřim Tarihi : 9.10.2006.

Anonymous. 2005. Web Sitesi: www.robot2.uniroma2.it, Eriřim Tarihi : 8.11.2000.

Banka, N. and Lin, Y.J. 2002. Mechanical design for assembly of a 4-DOF arm utilizing a top-down concept, Dept. of Mechanical Engineering. The University of Akron (USA).

Bingöl, Z. ve Küçük, S. 2005. Robot Teknięi-I, Birsen Yayınevi, İstanbul, 23-59.

Bayseç, S. and Alıcı, G. 1993. “Two Dimensional Step Guidance of a Dynamic Load Avoiding Residual Vibrations by a Point Positioning Manipulator”, Computer Simulation Conference, s 425-428, Boston.

Canidemir, E. 2007. Mekatronik Klubü , Web Sitesi: www.mekatronikkulubleri.org, Eriřim Tarihi: 05.08.2007.

Canny, J. and Monaco, D. 1992. Real Time Inverse kinematics for General 6R Manipulators, Proceedings of IEEE Conference on Robotics and Automation.

Craig, J. 2005. Introduction to Robotics Mechanics and Control, pp 1-7, Prentice Hall.

Kahan, W. 1983. Lectures on Computational aspects of Geometry, Unpublished Manuscripts.

Mikkelsen, J. 1998. A Machine Vision System Controlling a Lynxarm Robot along a Path , University of Cape Town, South Africa.

Niku, S.B. 2001. Introduction to Robotics, pp 10-96 , Prentice Hall.

Özgüren, M. Kemal. 2002. Topological analysis of 6-joint serial manipulators and their IK solutions, Mechanism and Machine Theory, 7 p.511-547.

- Özsoy, Ö. 2002. Robot & Elektronik, “Yakın tarihte robotik biliminin gelişimi” Web Sitesi: <http://www.endtas.com/> , Erişim Tarihi: 05.08.2007.
- Paden, B. 1986. Kinematics and Control Robot Manipulators, PhD. Thesis, Universty of California, Berkeley.
- Russell, S. and Norving, P. 2001. Prentice Hall , Artificial Intelligence A Modern Approach , pp 67-76 .
- Sigel, S. 2001. Traninig an artifical neural network to play Tic-Tac-Toe ,Term Project.
- Sungur, M. and Halıcı, U. 1992. Optimizing Neural Networks for playing Tic–Tac–Toe , Neural Networks, IJCNN, International Joint Conference.
- Xie, J. 2006. A method for solving the inverse kinematics problem of 6-DOF space manipulators , Systems and Control in Aerospace and Astronautics.
- Zaccarian, L. 2005. DC motors dynamic model and control techniques , pp 16-22.

EKLER

Ek 1 C# Program Kodları, C# Program Kodları Tezle birlikte verilen CD içerisinde.

ÖZGEÇMİŞ

Adı Soyadı : Mehmet Serdar GÜZEL

Doğum Yeri : Erzurum

Doğum Tarihi : 04.02.1980

Medeni Hali : Bekar

Yabancı Dili : İngilizce ,Almanca

Eğitim Durumu

Lise : Erzurum Anadolu Lisesi: 1995- 1999

Lisans : Başkent Üniversitesi Bilgisayar Mühendisliği: 2000 -2004

Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar
Mühendisliği Anabilim Dalı: (Şubat 2006 - Haziran 2008)

Çalıştığı Kurumlar

ESC BİLGİ İŞLEM , Ağustos 2004 - Aralık 2004

HAVELSAN , Ocak 2005 - Aralık 2005

Ankara Üniversitesi Bilgisayar Mühendisliği Bölümü Araş. Gör. ,
Ocak 2006-