

**ROBOT KİNEMATİK DENKLEMLERİNİN FPGA
İLE ÇÖZÜLMESİ VE ÇOK EKLEMLİ BİR
ROBOTA UYGULANMASI**

**2015
DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

Hüseyin Oktay ERKOL

**ROBOT KİNEMATİK DENKLEMLERİNİN FPGA İLE ÇÖZÜLMESİ VE
ÇOK EKLEMLİ BİR ROBOTA UYGULANMASI**

Hüseyin Oktay ERKOL

**Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalında
Doktora Tezi
Olarak Hazırlanmıştır**

**KARABÜK
Temmuz 2015**

Hüseyin Oktay ERKOL tarafından hazırlanan “ROBOT KİNEMATİK DENKLEMLERİNİN FPGA İLE ÇÖZÜLMESİ VE ÇOK EKLEMLİ BİR ROBOTA UYGULANMASI” başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Hüseyin Demirel

Tez Danışmanı, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Elektrik-Elektronik Mühendisliği Anabilim Dalında Doktora Tezi olarak kabul edilmiştir. 15/06/2015

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Prof. Dr. Mehmet AKBABA (KBÜ)

Üye : Yrd. Doç. Dr. Hüseyin DEMİREL (KBÜ)

Üye : Yrd. Doç. Dr. Selim ÖNCÜ (KBÜ)

Üye : Doç. Dr. Nurettin TOPALOĞLU (GÜ)

Üye : Doç. Dr. Bünyamin CİYLAN (GÜ)

...../...../2015

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Nevin AYTEMİZ

Fen Bilimleri Enstitüsü Müdürü

“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Hüseyin Oktay ERKOL

ÖZET

Doktora Tezi

ROBOT KİNEMATİK DENKLEMLERİNİN FPGA İLE ÇÖZÜLMESİ VE ÇOK EKLEMLİ BİR ROBOTA UYGULANMASI

Hüseyin Oktay ERKOL

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. Hüseyin DEMİREL

Temmuz 2015, 122 sayfa

Kinematik hesaplar, karmaşık olması nedeniyle, uzun zaman almakta ve çok eklemlili yapıların gerçek zamanlı hareket kontrollerinde tek işlemci ile istenen doğrulukta, hassasiyette ve sıklıkta sonuç almak zor olmaktadır. Özellikle serbestlik derecesinin artmasıyla birlikte ayrı bir hesaplama birimi kaçınılmazdır. Bununla birlikte, birbirinden bağımsız çok sayıdaki hareket elemanının kapalı çevrim kontrolleri merkezi işlem birimine ek yükler getirmektedir. Ayrıca, çok sayıdaki bu birimler için kullanılan kablolama, tasarlanan fiziksel yapıyı karmaşık hale getirmektedir. Bu çalışmada bu problemlere çözüm getirecek bir donanım tasarımı, çok yüksek hızlı entegre devre tanımlama dili (VHDL) ve alanda programlanabilir kapı dizileri (FPGA) kullanılarak gerçekleştirilmiştir. Altı bacaklı bir robot için tasarlanan bu donanım, teknolojiadaki eğilimlere paralel olarak yaygınlaşan, seri arabirimden kontrol edilebilen servo motorlar ile çalışmaktadır. Seri arabirimden aldığı ayak pozisyonları için gerekli eklem açılarını hesaplayıp, bunları yine seri arabirimden

servo motorlara göndermektedir. Bu donanım kinematik denklemlerin çözümü için bir yardımcı işlemci içermekte ve elektromekanik karmaşayı azaltacak donanımlar ile kullanılabilir. Bu donanımın robotlar için tasarımdan uygulamaya geçişi hızlandıracak bir araç olarak kullanılması öngörülmektedir.

Donanımın uygulanması için altı bacaklı bir robot ve yürüme algoritması geliştirilmiş ve başarı ile uygulanmıştır. Robot üzerinde çeşitli mesafe sensörleri ve ayaklarında zemini algılayabilmesi için temas sensörleri bulunmaktadır. Ayrıca uzak bilgisayar ile veri alış verişi yapabilmesi için bir radyo modülüne sahiptir. Uzak bilgisayardan yön komutları almakta ve bilgisayara sensör bilgilerini göndermektedir. Tüm bu donanımı yönlendirmek ve robot üzerindeki sensörlerden alınan verileri yorumlamak için bir bilgisayar arayüzü tasarlanmıştır.

Anahtar Sözcükler: Çok serbestlik dereceli sistemler, yardımcı işlemci, kinematik, kinematik, seri iletişim, altı bacaklı robot, FPGA.

Bilim Kodu : 905.1.035

ABSTRACT

Ph. D. Thesis

SOLUTION OF ROBOT KINEMATIC EQUATIONS USING FPGA AND ITS APPLICATION ON A MULTI-DEGREE-OF-FREEDOM ROBOT

Hüseyin Oktay ERKOL

**Karabük University
Graduate School of Natural and Applied Sciences
Department of Electric-Electronic Engineering**

Thesis Advisor:

Assist. Prof. Dr. Hüseyin DEMİREL

July 2015, 122 pages

As kinematic calculations are complicated, it takes a long time and is difficult to get the desired accurate result with a single processor in real-time motion control of multi-degree-of-freedom (MDOF) systems. Another calculation unit is needed, especially with the increase in the degree of freedom. The main Central Processing Unit (CPU) has additional loads because of numerous motion elements which move independently from each other and their closed-loop controls. Designed systems are also complicated because they have many parts and cabling. This paper presents the design and implementation of a hardware that will provide solutions to these problems. It is realized using the Very High Speed Integrated Circuit Hardware Description Language (VHDL) and Field-Programmable Gate Array (FPGA). This hardware is designed for a six-legged robot and has been working with servo motors controlled via the serial port. The hardware on FPGA calculates the required joint angles for the feet positions received from serial port and sent the calculated angles

to the servo motors via the serial port. This hardware has a co-processor for the calculation of kinematic equations and can be used together with the equipment that would reduce the electromechanical mess. It is intended to be used as a tool which will accelerate the transition from design to application for robots.

A six legged robot and a walking algorithm was developed for experimental study and taken good results. The robot has some distance sensors on the body and touch sensors on the feet. It has also a radio module for communication with a personal computer. The robot can take command for guidance from the computer and send sensor information to the computer. A computer interface was designed to control the robot and displaying the sensor information.

Keywords : Multi-degree-of-freedom systems, co-processor, kinematics, serial communication, six legged robot, FPGA.

Science Code : 905.1.035

TEŐEKKÜR

Bu tez alıőmasının planlanmasında, araőtırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle alıőmamı bilimsel temeller ışığında őekillendiren sayın hocam Yrd. Do. Dr. Hüseyin Demirel'e sonsuz teşekkürlerimi sunarım.

Sevgili aileme manevi hiçbir yardımını esirgemedен yanımda oldukları için tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ	xv
SİMGELER VE KISALTMALAR DİZİNİ	xvi
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	5
LİTERATÜR	5
BÖLÜM 3	18
ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA)	18
3.1. FPGA'İN YAPISI	19
3.2. DOĞRULUK TABLOSU	20
3.3. ARA BAĞLANTILAR	22
3.3.1. Komşular Arası Bağlantı	22
3.3.2. Gruplar Arası Bağlantı.....	23
3.3.3. Hiyerarşik Bağlantı	25
3.4. FPGA'İN PROGRAMLANMASI	26
3.5. FPGA'LERİN SAĞLADIĞI AVANTAJLAR	26
BÖLÜM 4	28
ROBOT KİNEMATİĞİ	28

	<u>Sayfa</u>
4.1. İLERİ KİNEMATİK	31
4.1.1. Denavit–Hartenberg Yöntemi.....	33
4.2. TERS KİNEMATİK.....	37
BÖLÜM 5	42
UYGULAMA ROBOTUNUN MEKANİK TASARIMI	42
5.1. ROBOTUN TASARIMI VE ÇİZİMLER	43
5.2. KİNEMATİK DENKLEMLERİN OLUŞTURULMASI	47
5.2.1. Üç Eklemlili Bacağın İleri Kinematiği	48
5.2.2. Üç Eklemlili Bacağın Ters Kinematiği.....	50
BÖLÜM 6	53
ELEKTRONİK DONANIM VE KUMANDA ARAYÜZÜ TASARIMI.....	53
6.1. ANAKART	54
6.2. SERVO MOTORLAR	59
6.3. DİĞER DONANIM VE SENSÖRLER	61
6.4. UZAKTAN KUMANDA ARAYÜZÜ	62
BÖLÜM 7	65
FPGA ÜZERİNDE GELİŞTİRİLEN DONANIM	65
7.1. SERİ İLETİŞİM	66
7.1.1. Servo Motorlara Ait İletişim Paketinin Yapısı	67
7.1.2. Tasarlanan Donanımda Kullanılan Paket Yapısı	69
7.2. TASARLANAN DONANIMIN GENEL YAPISI	70
7.3. SERİ PORT DONANIMI	72
7.4. KİNEMATİK YARDIMCI İŞLEMCİ	73
7.5. TASARLANAN DONANIMIN SİMÜLASYONU	75
7.5.1. Alıcı Simülasyonu	75
7.5.2. Gönderici Simülasyonu	77
7.5.3. Yardımcı İşlemci Simülasyonu.....	78

	<u>Sayfa</u>
BÖLÜM 8	80
YÜRÜME ALGORİTMASININ GELİŞTİRİLMESİ.....	80
8.1. BACAKLARIN GÖVDE ÜZERİNDEKİ KONUMLARI	83
8.2. YÜRÜME ALGORİTMASI	84
8.2.1. Ayak Pozisyonları Üretme Bloğu	85
8.2.1.1. Karar Bloğu	86
8.2.1.2. Hareket Koordinasyon Bloğu.....	93
8.2.1.3. Bacak Kontrolleri Bloğu	95
8.3. SİMÜLASYON ÇIKTILARI.....	97
BÖLÜM 9	100
UYGULAMA VE TEST SONUÇLARI.....	100
9.1. FPGA ÜZERİNDE GELİŞTİRİLEN DONANIMIN TESTLERİ	100
9.2. UYGULAMA ROBOTU ÜZERİNDE YAPILAN TESTLER.....	102
9.3. PERFORMANS DEĞERLENDİRMESİ	106
BÖLÜM 10	111
SONUÇ VE ÖNERİLER	111
KAYNAKLAR	114
ÖZGEÇMİŞ	122

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 3.1. FPGA'in genel yapısı.....	19
Şekil 3.2. Doğruluk tablosunun yapısı	21
Şekil 3.3. Hafıza elemanı ile birlikte doğruluk tablosu.....	22
Şekil 3.4. Ada tipi yapının şematik olarak tarifi	22
Şekil 3.5. Komşular arası bağlantı örneği	23
Şekil 3.6. Bağlantı kutuları ve anahtarlama kutuları içeren FPGA yapısı	24
Şekil 3.7. Bağlantı kutusunun yapısı.....	24
Şekil 3.8. Anahtarlama kutusunun yapısı.....	25
Şekil 3.9. Hiyerarşik bağlantı	25
Şekil 4.1. Koordinat sisteminin Z eksenini etrafında alfa açısı kadar dönmesi	29
Şekil 4.2. Üç işlevcinin konumu ve yer değişimi.....	32
Şekil 4.3. Denavit-Hatenberg parametreleri.....	34
Şekil 4.4. Uzuv uzunluğu ve bükülme açısı.....	35
Şekil 4.5. Uzuv açıklığı ve dönme açısı	35
Şekil 4.6. Robot kolun hareket uzayı	37
Şekil 4.7. İki farklı ters kinematik çözüm	38
Şekil 4.8. Üç eklemlili bir düzlemsel robot kol.....	39
Şekil 5.1. Gerçek örümcek ve mekanik tasarım.....	42
Şekil 5.2. İlk tasarım ve burulma problemi.....	43
Şekil 5.3. İlk ve ikinci tasarımın karşılaştırması	43
Şekil 5.4. İlk tasarımın bacak kesiti	44
Şekil 5.5. Altı bacaklı robotun gövde yapısı	45
Şekil 5.6. Robotun ön kısmına yerleştirilen sensörler	45
Şekil 5.7. Tasarlanan iki farklı ayak ucu.....	46
Şekil 5.8. Farklı açılardan robot tasarımının son hali	46
Şekil 5.9. Robotun mekanik ve elektronik montajının tamamlanmış hali	47
Şekil 5.10. Üç eklemlili bacak modeli	48
Şekil 5.11. Temel bacak yapısı ve eklemler.....	48

Sayfa

Şekil 6.1. Sistemin genel elektriksel bağlantı yapısı.....	54
Şekil 6.2. Lojik tampon ve gerilim bölücüler	55
Şekil 6.3. Anakart güç katı.....	56
Şekil 6.4. Radyo modülü bağlantıları.....	56
Şekil 6.5. Motor ve diğer çevre birimlerin bağlantı soketleri	57
Şekil 6.6. Dijital sensör devresi	57
Şekil 6.7. Anakart PCB çizimi (üstten görünüm)	58
Şekil 6.8. Anakart ve bileşenleri	58
Şekil 6.9. Sistem üzerinde anakart ve bileşenleri.....	59
Şekil 6.10. Sistem üzerinde DSP kart ve sensörler	59
Şekil 6.11. Motorların seri hatta bağlanması	60
Şekil 6.12. Tez uygulaması için geliştirilen arayüz	64
Şekil 7.1. Seri hat üzerinden veri iletim sinyali	67
Şekil 7.2. Genel donanım yapısı	71
Şekil 7.3. Genel akış şeması.....	71
Şekil 7.4. Seri port alıcı ve verici akış şemaları.....	72
Şekil 7.5. Rotor pozisyon ayrıntısı.....	75
Şekil 7.6. Alıcı simülasyonu	76
Şekil 7.7. Bir komut paketinin alınması.....	77
Şekil 7.8. Gönderici simülasyonu	77
Şekil 7.9. Oluşturulan paketin gönderilmesi.....	78
Şekil 7.10. Hesaplama zamanı	79
Şekil 7.11. “Calculation and packaging are completed”, “Send created packs” ve “Package is transmitting” sinyalleri	79
Şekil 8.1. Yürüme algoritmasına genel bir bakış	81
Şekil 8.2. Hareket sırasında ayakların yer değiştirme yönleri	81
Şekil 8.3. Gövde üzerinde bacakların yerleşimi.....	83
Şekil 8.4. DSP üzerinde gömülü sistemin genel yapısı.....	85
Şekil 8.5. Ayak hareketleri bloğu.....	86
Şekil 8.6. Cisim algılama sensörleri.....	87
Şekil 8.7. Ayağın zemine basma kontrolü	87
Şekil 8.8. Karar mekanizması blok diyagramı	88

	<u>Sayfa</u>
Şekil 8.9. Blok diyagram 2	90
Şekil 8.10. Ölçme bloğu.....	91
Şekil 8.11. Yardım bloğu	92
Şekil 8.12. Hareket koordinasyon bloğu	94
Şekil 8.13. Bacak kontrolleri bloğu.....	96
Şekil 8.14. İleri hareket sırasında oluşan ayak yer değişimleri.....	97
Şekil 8.15. Geri hareket sırasında oluşan ayak yer değişimleri	98
Şekil 8.16. Ayağın yüksek bir cisme basması sırasındaki yer değişimleri	99
Şekil 9.1. Ayağın X,Y,Z pozisyonları.....	101
Şekil 9.2. Eklem açılarındaki değişim.....	102
Şekil 9.3. Simülasyonun ürettiği sonuçlar ile FPGA donanımının ürettiği sonuçların karşılaştırılması.....	102
Şekil 9.4. Yüksek engelli parkur örneği.....	103
Şekil 9.5. Alçak engelli parkur örneği	103
Şekil 9.6. Robotun ileri giderken engel aşması sırasında kaydedilen kareler.....	104
Şekil 9.7. Robota üç ayak üzerinde etkiyen temel kuvvetler	107
Şekil 9.8. Hız testi sırasında alınan görüntüler	109

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 5.1. Robotun ölçüleri	47
Çizelge 5.2. Üç eklemlili bacak için Denavit–Hartenberg parametreleri.....	49
Çizelge 6.1. Motor soketi pin açıklamaları	60
Çizelge 6.2. Motor iletişim paketinin genel yapısı	61
Çizelge 7.1. İletişim paketi ayrıntıları	68
Çizelge 7.2. Yeşil LED’i yakan veri dizisi	68
Çizelge 7.3. Rotoru 90° konuma götürecekt olan veri dizisi.....	68
Çizelge 7.4. Genel komut yapısı	69
Çizelge 7.5. İkinci grup (pozisyona git) komut yapısı	70
Çizelge 7.6. Konum parametresinin yapısı	70
Çizelge 7.7. Pozisyon komutu veri dizisine örnek.....	74
Çizelge 8.1. Komut sinyali veri yapısı	85
Çizelge 9.1. Robotun ölçüleri.....	107

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

- A : Genel dönüşüm matrisi
T : Robot ayağı dönüşüm matrisi
R : Dönme matrisi
P : Uzayda herhangi bir nokta
X : Kartezyen koordinat sisteminin x eksenini
Y : Kartezyen koordinat sisteminin y eksenini
Z : Kartezyen koordinat sisteminin z eksenini
O : Koordinat sisteminin merkez noktası
Px : Uzayda tanımlı bir noktanın x eksenini üzerinde konumu
Py : Uzayda tanımlı bir noktanın y eksenini üzerinde konumu
Pz : Uzayda tanımlı bir noktanın z eksenini üzerinde konumu
 a_i : i numaralı uzuv uzunluğu
 d_i : i numaralı uzuv açıklığı
 α_i : x_i ekseninin referansına göre z_{i-1} ve z_i arasındaki açı
 β_i : z_{i-1} ekseninin referansına göre x_{i-1} ve x_i arasındaki açı

KISALTMALAR

ASIC	: Uygulamaya Özel Bütünleşmiş Devre (Application Specific Integrated Circuit)
CNC	: Bilgisayarlı Nümerik Kontrol (Computer Numerical Control)
CMOS	: Bütünleyici Metal Oksit Yarı İletken (Complementary Metal Oxide Semiconductor)
CPGs	: Merkezi Örnek Üreticiler (Central Pattern Generators)
DSP	: Dijital Sinyal İşleme (Digital signal processing)
FPGA	: Alanda Programlanabilir Kapı Dizileri (Field Programmable Gate Array)
HDL	: Donanım Tanımlama Dili (Hardware Description Language)
LUT	: Doğruluk Tablosu (Look up table)
MDOF	: Çok Serbestlik Dereceli (Multi-Degree-Of-Freedom)
PCI	: Çevresel Cihaz Bağlantı (Peripheral Component Interconnect) yuvası
PID	: Oransal İntegral Türevsel (proportional, integral, derivative) Denetleyici
VHDL	: VHSIC Donanım Tanımlama Dili
VHSIC	: Çok Hızlı Bütünleşik Devre (Very High Speed Integrated Circuit)

BÖLÜM 1

GİRİŞ

Algoritmaların işletilmesinde geleneksel iki yöntem vardır. Bunlardan ilki uygulamaya özel bütünleşik (hardwired) teknolojidir. Bu uygulamaya özel bütünleşmiş devre (Application Specific Integrated Circuit, ASIC) veya geleneksel devre elemanlarından oluşan bir elektronik kart anlamına gelmektedir. Bu tür donanımlar amaçlanan işi yerine getirirken çok hızlı ve efektiftir [1]. Bununla birlikte bu bütünleşmiş devreler üretimden sonra değiştirilemezler. Herhangi bir donanımsal değişiklik gerektiğinde yeniden tasarım ve üretim işlemlerinin gerçekleşmesi zorunludur. Bu özellikle büyük sistemler içerisindeki ASIC'lerin değiştirilmesi söz konusu olduğunda oldukça pahalı bir süreçtir [2]. Kart düzeyindeki uygulamalar biraz daha esnektir ancak yine her değişiklikte yeni bir kart tasarımı ve kartın eskisiyle değiştirilmesi söz konusudur.

İkinci bir yöntem ise tekrar programlanabilir mikroişlemcilerdir ki bu ilk alternatiften (ASIC veya elektronik kart) daha esnektir. Belirli sınırlar dahilinde yazılım değiştirilerek sistemin fonksiyonları değiştirilebilir. Bu esnekliğin yanında saat hızı, ilgili işin yapılması için yeterince yüksek değilse performans kötüleşebilir [2]. İşlemci her defasında önce hafızadan kodu okumalı, çözümlmeli ve sonra işi yapmalıdır. Bu her defasında yerine getirilecek olağan komutlar için ek zaman kaybı demektir. Bununla birlikte işlemci üretildiği sırada belli bir komut seti oluşturulur ve bundan sonra yazılacak tüm programlar bu komut seti ile icra edilmek zorundadır.

Yeniden programlanabilir yapılar bu iki seçenek arasındaki boşluğu doldurmaktadır [3]. Bunu donanımsal yapısı sayesinde yazılımdan hızlı olması ve üzerinde yeni donanımlar oluşturulabilmesi sayesinde kazandığı esneklik ile yapmaktadır [2–4]. Yeniden düzenlenebilir donanımlar sayesinde geleneksel olarak yazılıma atfedilen esneklik aynı derecede donanıma da uygulanabilmektedir [5]. Birçok uygulama için

en hızlı sonuçları üreten ve en az kaynak harcayan alternatif FPGA (Field Programmable Gate Array, Alanda Programlanabilir Kapı Dizileri)'dir [6]. Bu anlamda FPGA'ler uygulamaya özel bütünleşmiş devrelerin bir evrimi niteliğindedir. İlk ticari FPGA 1986 yılında Xilinx firması tarafından üretilmiştir [7].

FPGA'lerin diğer programlanabilir elemanlar ile kullanımı da yaygındır. Programlanabilir komponentlerde hız artışı ve esneklik endüstriyel uygulamalara yönelik dijital çözümlerde birçok imkân sunmaktadır. Bu özellikle mikroişlemci ve DSP'ler üzerinde geliştirilen yazılım uygulamaları için geçerlidir [8]. Bununla birlikte FPGA'ler gibi uygulamaya özel donanım geliştirme teknolojileri, bu kontrolcülerin hız ve performansının yukarı çekilmesinde önemli rol oynamaktadır [8]. Bu kapsamlı komponentler düşük maliyetli ürün geliştirme, uygun ve pratik yazılım geliştirme araçları ve çok büyük lojik element yoğunluklarını (dolayısıyla çok büyük donanımların en az FPGA ile geliştirilmesi imkânını) bir arada sunmaktadır [8,9].

FPGA teknolojisi tasarımcılar tarafından birçok alanda kullanılmakta ve yeni alanlarda kullanımı her geçen gün artmaktadır. Bunların başında ısrarla ve sürekli olarak daha fazla çip yoğunluğu isteyen telekomünikasyon, görüntü ve sinyal işleme uygulamaları gelmektedir [8,10,11]. Son zamanlarda büyüyen diğer uygulama alanları ise medikal cihazlar, robotik, otomotiv ve uzay/uçak kontrol sistemleri olarak sıralanabilir [12–21]. Bütün bu uygulamalarda güç tüketiminin azalması, termal yapı ve paketleme, fonksiyonel güvenilirlik ve solar radyasyona karşı dayanıklılık gibi özellikler FPGA'lerin avantajları arasındadır [22–27].

FPGA üzerindeki devreler uygulama için özelleştirilmiştir, böylece güç tüketimleri geleneksel işlemcilerden çok daha az olmaktadır [1]. Yapılan bir çalışmada, farklı FPGA modülleri ile yapılan denemelerde, bu güç farkı %35-%70 arasında değişmiştir ve bununla birlikte FPGA 3-7 kat daha hızlı işlem yapmıştır [1]. FPGA'leri cazip kılan bir diğer özellik ise paralel işlem yeteneğidir. FPGA içerisinde çok sayıda birbirinden bağımsız donanım yer alabilmektedir. Bu paralel yapı ses tanıma, görüntü ve ses işleme gibi sinyal işleme uygulamalarında yüksek örnekleme zamanları ve daha az çevrimde sonuca ulaşma gibi avantajlar sağlamaktadır [10].

Bahsedildiği gibi FPGA'lerin uygulama alanlarından biri de robotiktir. Hayvanların değişen doğal şartlara adaptasyon yeteneği araştırmacıları her zaman cezbetmiş ve benzer yeteneklere sahip makineler yapmaya yöneltmiştir. İki ayaklı insansı robotlardan başlayarak çeşitli memeli, böcek ve diğer türleri de içeren çalışmalar yapılmıştır. Teorik olarak tekerlekli ve paletli sistemlerin sürekli yerle temasa ihtiyacı olmasından ötürü, bacaklı sistemler düzgün olmayan arazi koşullarında daha iyi ilerleme potansiyeline sahiptirler [28,29]. Eğimli alanlarda daha iyi tırmanabilirler. Bacak sayısı ne kadar çok ise ortama adaptasyon o kadar iyi olmakta ancak, bacak sayısının artmasıyla kontrol sistemi karmaşıklaşmaktadır [28,30]. Bununla birlikte bacaklı robotların, düz zeminlerde tekerlekli robotlardan çok daha yavaş hareket etmeleri ve kontrollerinin daha karmaşık olması gibi bazı dezavantajları vardır. Tüm bacaklı robotlar içerisinde ortama adaptasyon ve kontrol sistemlerinin karmaşıklığı bir arada değerlendirildiğinde altı bacaklı robotlar mantıklı bir seçim olmaktadır ve büyük ilgi çekmektedir [28]. Bununla birlikte üç serbestlik derecesi, bu bacağına sahip bir robotun tüm yönlerde yürüyebilmesi ve dar alanlarda yeterince esnek olabilmesi için gerekli minimum derecedir [31].

Robotik uygulamalarda görüntü işleme, sinyal işleme, kontrol algoritmaları gibi, FPGA'in kullanılabileceği çok sayıda konu vardır. Bunlardan biri de kinematik denklemlerin çözümüdür [32–34]. Robotlarda izlenecek konum ve yönelimler Kartezyen koordinatlarda verilmekte, serbestlik derecesi arttıkça kinematik denklemlerin çözümü karmaşıklaşmakta ve sarsıntısız ve hızlı hareketlerin üretilebilmesi için çok hızlı hesaplamalar gerekmektedir [35]. Böyle durumlarda bu fonksiyonların çözümü için yardımcı işlem birimleri kullanılarak ana işlemci üzerindeki yük hafifletilebilir [35].

Bu çalışmada çok eklemlili robotik yapılara ait kinematik denklemlerin FPGA ile çözülmesine yönelik bir çalışma yapılmış ve bir robot üzerinde uygulanmıştır. Donanım seri port üzerinden kontrol edilebilen servo motorlar ile kullanılmak üzere özelleştirilmiştir. Geliştirilen donanım seri haberleşme birimlerine ve kinematik denklemlerin çözümü için yardımcı işlem birimine sahiptir. Bu donanımın uygulaması için bir robot tasarlanmıştır. Robot altı bacaklı yapıya sahiptir ve her bir bacağı üç serbestlik derecesine sahiptir. Üzerinde çeşitli sensörler ve uzaktaki

bilgisayar ile veri alış verişi yapabilmesi için kablosuz haberleşme birimi mevcuttur. Ayrıca robot için bir uzaktan kumanda arayüzü tasarlanmış ve sistem üzerinde bir yürüme algoritması geliştirilmiştir.

Bu çalışmanın giriş bölümünde konuyla ilgili temel bilgiler verilmiştir. İkinci bölümde konuyla ilgili literatür taraması verilmiş ve yapılan çalışmadan bahsedilmiştir. Üçüncü bölümde FPGA'in yapısı hakkında bilgiler yer almaktadır. Dördüncü bölümde robot kinematiği ele alınmıştır. Beşinci bölümde, tez çalışmasının bir parçası olan, robotun tasarımı anlatılmıştır. Altıncı bölümde tasarlanan ve kullanılan elektronik donanım ve uzaktan kumanda arayüzü açıklanmıştır. Yedinci bölümde FPGA üzerinde geliştirilen donanımın tasarımı ve ayrıntılarından bahsedilmiştir. Sekizinci bölümde, uygulama robotu için geliştirilen, yürüme algoritması ele alınmıştır. Dokuzuncu bölümde FPGA üzerinde geliştirilen donanım ve robot ile ilgili uygulama ve test sonuçları ile performans değerlendirmesine yer verilmiştir. Sonuçlar ve öneriler ise onuncu bölümde tartışılmıştır.

BÖLÜM 2

LİTERATÜR

Huang ve arkadaşları 2009 yılında yaptıkları bir çalışmada FPGA üzerine gömülen bir sistem ile CMOS (Complementary Metal Oxide Semiconductor, Bütünleyici Metal Oksit Yarıiletken) kameradan alınan robot kolu görüntülerin işlenmesiyle bir tür “Visual Servo” sistemi geliştirilmiştir [36]. Uygulama eş zamanlı olarak robot kolunun konumunu izleyerek o anki pozisyonunu belirlemekte ve verilen komuta göre bir sonraki konuma ulaşması için, gerekli kinematik denklemleri çözerek, eklemlerin açılarını hesaplayabilmektedir. Çalışmada bilgisayar ile iletişim NiosII mikroişlemcisi ile yapılmıştır. Kinematik denklemler ve diğer algoritmaların çözümü Verilog ile geliştirilen FPGA üzerindeki donanımlar ile gerçekleştirilmiştir. Yapılan bu sistemin uygulaması “Gobang” isimli bir oyun ile denenmiştir. Bu çalışmada FPGA üzerinde geliştirilen özel donanımlar mikrodenetleyici ile birlikte kullanılmıştır.

2011 yılında Yu ve arkadaşları tarafından yapılan çalışma ise renkleri takip eden grup robotlar ile ilgilidir [37]. Görüntü işleme ile ilgili bu uygulamada FPGA ile çalışılmıştır ve bunun amacı az enerji harcayan gerçek zamanlı bir kontrolcü elde etmek olmuştur.

Paul ve ekibi 2012 yılında yaptıkları çalışmada, yürüyebilen robotlar için çevredeki hareketi algılamalarına yardımcı olacak bir donanımı FPGA ile gerçekleştirilmiştir [38]. Bacaklı robotların, arazi koşullarında dahi esnek bir şekilde hareket edebileceği vurgulanan çalışmada, bu yeteneğin daha da iyileştirilmesi için tek kamera kullanılarak etrafındaki hareketliliği algılamaya yönelik bir donanım geliştirilmiştir. Geliştirilen bu donanımın zorlu şartlarda bile yumuşak ve düzgün bir ilerlemeye yardımcı olacağı belirtilmiştir.

Wang ve arkadaşlarının 2013 yılında yaptığı merdiven tırmanan robot çalışmasında FPGA ve DSP (Digital Signal Processor) bir arada kullanılmıştır [39]. Motorlar ve diğer birimlerin kontrolü için DSP kullanılırken, kameradan gelen görüntünün işlenip hareket için gerekli parametrelerin belirlenmesinde FPGA kullanılmıştır.

Bazı çalışmalarda ise FPGA üzerine çeşitli yapay zekâ algoritmaları gömülmektedir. 2011 yılında yapılan bir araştırma beş eklemlili bir kola sahip yangın söndürme robotu üzerinedir [40]. Yapılan çalışmada kolun kontrolü için uygun parametrelerin belirlenmesinde Paralel DNA (Deoksirübo Nükleik Asit) algoritması kullanılmış ve bu yolla en kısa sürede gerekli hareketleri yapabilmek için parametreler hesaplanmıştır. FPGA üzerine gömülen algoritmalar ile geleneksel yazılım tabanlı uygulamalardan daha yüksek performans alınmıştır.

2009 yılında Vachhani ve ekibi tarafından yapılan çalışmada mobil bir robotun hareket hesaplamaları için FPGA tabanlı bir kontrolcü geliştirilmiştir [41]. Donanım robotun 2 boyutlu düzlemde dönmesini ve bir vektör boyunca hizalanmasını sağlayan bir algoritma işletmektedir. Robot ile duvar takip uygulaması yapılmıştır.

Huang 2013 yılında yaptığı çalışmalarda çok yönlü dört tekerleğe sahip bir mobil robotun kinematik hesaplarını FPGA tabanlı yapay zekâ uygulamaları ile gerçekleştirmiştir. Bu çalışmalarda bağışıklık sistemi algoritması (artificial immune system algorithm, AIS) ve karınca kolonisi yaklaşımlarını FPGA'in paralel işlem yeteneği ile birleştirerek kullanmıştır [42,43]. Yine Baturone ve arkadaşları da, 2014 yılında, neuro-fuzzy teknikleri FPGA üzerinde uygulayarak tekerlekli bir mobil araç için yer bulma çalışması yapmıştır [44].

Huang 2012 yılında yaptığı bir çalışmada FPGA tabanlı genetik algoritma ve parçacık sürüsü optimizasyonu kullanarak mobil robotlar için rota belirleme işlemi yapmıştır [45]. Bu çalışmada tasarlanan donanım bir NiosII mikro işlemciye destek görevi görmektedir. Çalışmada bu tür uygulamalar için FPGA'in bilgisayar tabanlı uygulamalardan daha uygun olduğu belirtilmiştir.

2012 yılında Mingas ve arkadaşları tarafından yayınlanan bir çalışmada mobil robotların eşzamanlı konum belirleme ve haritalama (simultaneous localization and mapping, SLAM) problemi için geliştirilen bir yöntem FPGA ile uygulanmıştır [46]. Uygulamanın diğer yazılım tabanlı uygulamalardan, hiçbir hassasiyet kaybı olmadan, 14,83 kat daha hızlı olduğu gösterilmiştir.

C.-H. Huang ve arkadaşları tarafından, 2011 yılında yapılan çalışmada, iki tekerlekli bir ters sarkacın gerçekleştirilmesinde, FPGA üzerine gömülen bulanık mantık algoritmalarını kullanılmıştır [47]. Bulanık mantık temelli denge ve ilerleme algoritmalarından oluşan kontrol sistemi FPGA üzerinde gerçekleştirilirken, yazılımsal işlemci ve çevre donanımlar kullanılmıştır. Sistem bilgisayar ile seri port aracılığı ile haberleşmektedir.

2005 yılında Wong ve arkadaşları ise genetik algoritma tabanlı bulanık (fuzzy) bir sistemi omni-directional bir robota uygulamıştır [48]. Genetik algoritma tabanlı bulanık sistem bir FPGA üzerine gömülmüştür. Bulanık denetleyici için uygun parametreler genetik algoritma yardımı ile bulunmaktadır. Bulanık denetleyici ise robotun kinematik modeli kullanılarak oluşturulan veri yığını içerisinde uygun sonuçları seçerek robotu hareket ettirmektedir. Sistem kablosuz bir kumanda ile uzaktan kontrol edilmekte ve kumanda sinyaline göre istenilen yönde ve hızda hareket edebilmektedir.

Talaşlı imalat işlerinde, işleme hızı kalite için önem arz etmektedir. Hızı dinamik olarak kontrol etmeye yönelik uygulanan yöntemler zaten vardır. Bununla birlikte dijital sistemlerde hız değişimi belirli periyodlar ile yapılmaktadır ve iki hız değeri arasında sıçramalar meydana gelmektedir. Özellikle karmaşık şekillerin işlenmesi için yapılan yönlendirmelerin hesaplanması sistemin işlem yükünü, dolayısı ile işlem süresini çok fazla artırmaktadır. Bu problemi çözmek için 2014 yılında yapılan bir çalışmada, daha iyi bir hızlanma/yavaşlama devresi FPGA üzerinde tasarlanmış ve uygulaması yapılmıştır [49]. Sistem 3 eksenli bir sisteme uygulanmış ve daha iyi performans elde edildiği bildirilmiştir.

Osornio-Rios ve arkadaşları yüksek hızlı CNC tezgâhlar için FPGA tabanlı bir kontrolcü geliştirmişlerdir [50]. Bu kontrolcü üzerinde yüksek hızlı (tepki süresi kısa) PID (proportional-integral-derivative, oransal-integral-türev) kontrolcüler geliştirilmiştir. Bu çalışmada FPGA'in, servo kontrol uygulamalarında çok daha kısa bir döngü zamanı elde edilebildiği için, diğer mikroişlemci ve DSP uygulamalarından daha uygun bir seçenek olduğu belirtilmiştir.

Rodriguez-Donate ve ekibi tarafından 2011 yılında, endüstriyel uygulamalarda sıklıkla kullanılan Flexible Manipülatörler üzerine yapılan bir çalışmada ise yıpranmayla oluşan hataları önlemeye yönelik çalışmalar yapılmıştır [51]. Bu robotlarda performans konumun ve hareketin uygun doğrulukta bilinmesine bağlıdır; buda ileri kinematik (forward kinematics) ile yapılır. Bu çalışmada zamanla robot eklemlerinde oluşan ve eklem hareketi için kullanılan sensörler ile fark edilemeyen, eklemdeki mekanik yıpranmalar sonucu hatalara yol açan etkileri kaldırmak için akıllı bir sensör ağı uygulaması yapılmıştır. Bu ağda, eklemlerdeki sensörlere ek olarak üç eksenli ivmeölçerler kullanılmış ve her bir eklem için konumu yüksek doğrulukla hesaplanmıştır ve yoğun işlem yükü nedeniyle FPGA'ler kullanılmıştır.

Cho ve arkadaşları 2009 yılında FPGA tabanlı bir multi-axis hareket kontrol çipi üzerine çalışma yaptılar [52]. Bu çalışmada ise robotik kol ya da CNC (Computer Numerical Control) makine gibi çok eksenli sistemlerin nümerik kontrolü amaçlanmıştır. Bu çip kinematik denklemleri çözmekte ve üzerinde kapalı döngü çalışan PID kontrolcü bulundurmaktadır. Bu çip yüksek örnekleme oranına sahip, hassas, esnek, kompakt, düşük güçlü ve düşük maliyetli uygulamalara izin vermektedir. Bu çipin scara tipi bir robot üzerinde uygulaması yapılmıştır.

2008 yılında Koca ve ekibi uzaktan kumandalı robot kollar üzerine bir çalışma yapmıştır [53]. Kumanda kolundan gelen sinyalleri yorumlayarak robotun hareket edeceği yeni konumu kestirip, bu konuma ulaşması için, kinematik denklemleri çözerek, gerekli eklem açılarını hesaplayan bir sistem üzerine çalışılmıştır. Sistem FPGA kullanılarak gerçekleştirilmiştir.

2013’de Kim ve Ahn tarafından yapılan çalışmada bir “bilateral teleoperation system” geliştirilmiştir [54]. Bu sistemde paralel çalışan iki robot kol bulunmaktadır. İlk kola operatör tarafından yaptırılan hareketler aynen diğer kola aktarılmaktadır. Pozisyon ve kuvvet takibi problemlerinin çözümü için kumanda kolundan alınan bilgiler kullanılarak uzaktaki kolun kinematik denklemleri çözülmekte ve gerekli parametreler hesaplanmaktadır. National Instruments firmasının FPGA modülleri kullanılmış ve uygulaması yapılmıştır.

Juang ve arkadaşlarının 2012 yılında yaptığı bir çalışmada dört serbestlik dereceli bir robot kolun kinematik denklemlerinin çözülmesi için FPGA üzerinde geliştirilen bir donanım kullanılmıştır [35]. Burada FPGA ile gerçekleştirilen donanım, ana işlemcinin yükünü azaltan bir yardımcı işlem birimi olarak kullanılmıştır. Hesaplama elde edilen hız sayesinde gerçek zamanlı olarak çok yumuşak geçişli hareketler elde edildiği belirtilmiştir. Tasarlanan sistemde FPGA bilgisayara USB arayüzü kullanılarak bağlanmaktadır. Bilgisayardan konum bilgileri FPGA üzerinde tasarlanan donanıma gönderilmekte ve FPGA gerekli açı bilgilerini hesaplayarak servo sürücülere iletmektedir. Servo sürücüler de motorları istenen konuma getirmesiyle hareket sağlanmaktadır.

Zhu ve ekibi tarafından 2013 yılında yapılan uygulamada, modüler bir robot kolun hassas hareket kontrolü, FPGA tabanlı bir yaklaşım ile gerçekleştirildi [55]. Bu tasarımda sanal analiz kontrolü (virtual decomposition control, VDC) yaklaşımını kullanıldı. Bu uygulamayla modüler robotların geleneksel yeniden düzenlenebilme, esneklik, kolay kullanım ve çok yönlülük gibi özelliklerinin yanında hassas kontrol yeteneğinin de kazandırıldığı rapor edilmiştir. Robotu oluşturan her bir modül için bir FPGA kullanılmış, FPGA’ler arasındaki iletişim bir tür seri iletişim protokolü olan Space Wire (IEEE 1355) ile gerçekleştirilmiştir. Ayrıca bu yöntemin FPGA tabanlı kontrolcü kullanan diğer robot kollara da uygulanabileceğini belirtilmiştir.

Chou ve arkadaşları tarafından 2012 yılında yapılan çalışmada 3 eksenli bir gıda robotu için kontrolcü ve sürücü tasarlanmıştır [56]. Yapılan çalışmada üç tane PMSM motor kullanılmıştır. FPGA üzerine gömülen bir soft işlemci kinematik hesapları yapmış ve yine FPGA üzerinde üç adet vektör kontrol yapabilen motor

sürücü tasarlanmıştır. Bu uygulamada işlem yükü motor sürücülerde daha fazla olduğu için FPGA üzerinde motor sürücü devreleri tasarlanmış ve üç eklemlili yapının matematiksel modeli mikro işlemci üzerine gömülmüştür.

Martinez-Prado ve arkadaşlarının 2013'de yaptıkları çalışmada mikro işlemci kullanan robot kontrolcülerine alternatif olarak FPGA üzerinde bir kontrolcü geliştirdiler [57]. Bu kontrolcü ile iletişim ve kontrol algoritmaları tek çip üzerine gömüldü ve sistem gereksinimlerine göre serbestlik derecesinin artırılmasını sağlayacak şekilde tasarlandı. Geliştirilen sistem hareket komutlarını bir bilgisayardan alıyor ve üç eksenli bir robot kol için gerekli hesaplamaları yaparak gerekli değişkenleri üretiyor. Çalışmada iyi bir performans elde edildiği belirtilmiştir.

2007 yılında Shao ve arkadaşlarının yaptığı çalışmada yüksek hızlı bir robot kontrolcü tasarımı, FPGA ve DSP birlikte kullanılarak gerçekleştirildi [58]. Çalışmada DSP üzerindeki yükü azaltmak ve sistemin hesaplama yeteneğini artırmak için böyle bir tasarım yapıldığı vurgulanmıştır. Sistemin kontrolünde kullanılacak olan algoritmalar lineer olanlar FPGA, non-lineer olanlar ise DSP üzerinde gerçekleştirilmiştir. Böylelikle FPGA tabanlı bir hareket kontrolcüsü geliştirilmiş ve endüstriyel bir robot kol üzerinde uygulanmıştır. Sistem DSP üzerindeki seri port aracılığıyla bilgisayarla haberleşmektedir.

2008 yılında J.B. Huang ve arkadaşları tarafından dört serbestlik dereceli esnek bir robot için DSP ve FPGA kullanarak bir kontrolcü tasarımı yapıldı [59]. Bu çalışmada FPGA üzerine, motorların kontrolü için, uzay vektör modülasyon algoritması gömüldü. Diğer algoritmalar ise DSP üzerinde uygulandı. DSP ile FPGA arasında iletişim seri haberleşme ile sağlandı. Uygulamada kolun direnç kapasitesinin artırıldığı vurgulanmıştır.

Tanaka ve arkadaşları 2009 yılında 12 serbestlik dereceli, cerrahi amaçlı bir robot kolunun bilateral kontrolü için FPGA tabanlı bir sistem geliştirdiler [15]. Robotun serbestlik derecesi, hassasiyet gereklilikleri ve uygulanan kontrol sistemi çok kısa örnekleme periyodlarına ihtiyaç duymaktadır. Bu ihtiyaç FPGA üzerinde tasarlanan donanımlar ile giderilmiştir. Tasarlanan donanım bilgisayar ile PCI portu üzerinden

iletişim kurmaktadır. Uygulamanın gerçekleştirilmesi için birden fazla FPGA kullanılmıştır. Tasarlanan robot, operatörün yönlendirmesi ile hastaya müdahale etmektedir.

S.-J. Huang ve J.-C. Huang tarafından 2011’de yapılan çalışmada robot kolların kontrolü ve görüntü işleme yapılmıştır [60]. Kontrol algoritmaları FPGA üzerine, görüntü işleme algoritmaları ise DSP üzerine gömülmüştür. İki adet yedi serbestlik dereceli robot kol kullanılarak kamera ile tespit edilen objenin bir yerden bir başka yere taşınması işlemi gerçekleştirilmiştir. FPGA üzerindeki sistem hareket kontrol algoritmaları, PID algoritmaları ve seri iletişim algoritmalarından oluşmaktadır.

2013 yılında Yen-Fang Li ve Li-Lan Chuang yaptıkları çalışmada antroformik robot ile piyano çalma çalışması yapılmıştır [61]. Sistemde iki el ve on parmak bulunmaktadır. İki el ve parmakların paralel kontrolü için FPGA kullanılmıştır. Bilgisayar üzerinde oluşturulan müziği çalmak için gerekli komutlar FPGA’ye gönderilmekte ve FPGA elleri kumanda ederek müzik çalmaktadır. Eller konveyör üzerinde sağa sola hareket ettirilmektedir.

2013’de yapılan bir çalışmada ise bir iskelet-kas destek sistemi (motion assistive robotic-exoskeleton) geliştirilmiştir [62]. Bu sistem insanlara ağır işlerde destek olarak yüksek kas gücü gerektiren işleri rahatça yapmalarını sağlamaktadır. Yedi serbestlik dereceli olarak tasarlanan bu sistemin uygulanmasında yüksek örnekleme ve işlem hızı gerektiren kısımlarda FPGA, diğer kısımlarda ise bir gerçek zamanlı bilgisayar kullanılmıştır [62]. Böylelikle FPGA bilgisayar için bir yardımcı işlemci olarak kullanılmıştır.

2007 yılında İspanya’da yapılan bir araştırma ise çok bacaklı robotlar üzerinedir. Yapılan çalışmada Gregor isimli hexapod robota biyolojik benzerlerini taklit eden bir kontrol metodu yapay sinir ağları ile uygulanmıştır [63]. Asıl yapılan şey adım üretimi (gait generation)’dir. Bu robotun tırmanma kabiliyetini artırabilmek için yeni bir bacak tasarımı da yapılmıştır. Bu robotun kontrolü de FPGA ile yapılmıştır.

Çok eklemlı robotlar ¼zerine yapılan alıřmalar genellikle y¼r¼me algoritması (gait generation) ¼zerine yapılmıřtır. ¼lkemizde de bu alanda doktora seviyesinde yalnızca bir alıřma vardır. Erden tarafından yapılan bu alıřmada da gait generation ¼zerinde durulmuřtur [64].

2012 yılında Barron-Zambrano ve arkadaşları tarafından yapılan bir alıřma Merkezi ¼rnek ¼reticiler (Central Pattern Generators, CPGs) ¼zerinedir [65]. CPGs bacaklı robotların biyolojik y¼r¼me prensiplerine g¼re kontrol edilmesi iin etkili bir kontrol mekanizması olarak tanımlanmıřtır. Bacaklı robotların y¼r¼yebilmesi iin gerekli eklem koordinasyonunu yapay sinir ađı tabanlı bir sistem ile yapmaktadır. B¼ylelikle adım ¼retimi iřlemine yerine getirmektedir. Y¼ntem d¼rt ve altı bacaklı iki robot ¼zerinde, sim¼lasyon ortamında denenmiřtir. Uygulama iin FPGA kullanılmıřtır. FPGA ¼zerinde gerekleřtirilen CPGs donanımı UBlaze mikroiřlemci iin bir yardımcı donanım olarak g¼rev yapmaktadır. alıřma sim¼lasyon ortamında, 2 serbestlik dereceli robotlar kullanılarak, test edilmiřtir.

Kenton ve Leang tarafından yapılan 3 eksenli bir seri-kinematik nano-pozisyonerin y¼ksek hızlarda alıřma ihtiyacından ¼t¼r¼, kontrolc¼s¼n¼n FPGA ile gerekleřtirilmesi uygun g¼r¼lm¼řt¼r [66]. Bu nano-pozisyoneler mikroskoplarda, fiber optik hizalayıcılarda, mikro ve nano iřleme gibi alanlarda kullanılmaktadır.

Wong ve Liu tarafından 2013 yılında yapılan bir alıřmada, bipedal robotun kinematik denklemlerinin öz¼m¼nde FPGA kullanılmıřtır [67]. FPGA ¼zerinde kullandıkları donanımın hızı ile yine FPGA ¼zerinde gerekleřtirdikleri yazılımsal mikroiřlemci iin yazdıkları programın hızı karřılařtırıldı. FPGA donanımı ve iřlemcinin saat hızları aynı iken FPGA paralel iřlem yapmakta, mikro iřlemci ise ardışık iřlem yapmaktadır. Sonuç olarak FPGA yazılımsal öz¼mden yaklaşık 7595 kat daha hızlı olduđu rapor edilmiřtir.

Zhu ve ekibinin 2011 yılında yaptıđı sinyal iřleme uygulamasında kalman filtreleri kullanılarak bir donanım tasarlanmıřtır. FPGA ¼zerinde geliřtirilen donanım bilgisayar ¼zerinde uygulanan alternatifinden, hibir hassasiyet kaybı olmadan, 24,45 kat daha hızlı alıřtıđı rapor edilmiřtir [26].

Franchini ve arkadaşları tarafından 2013 yılında, geometrik modellemede kullanımı yaygınlaşmakta olan çok boyutlu Clifford matematiksel işlemlerinin yapılabilmesi için bir yardımcı-işlemci tasarlanmış ve prototipini gerçekleştirilmiştir [68]. Çalışma sonunda geliştirilen donanım PowerPC üzerindeki yazılımsal kütüphaneler ile karşılaştırıldığında, dört boyutlu objeler ile ilgili işlemlerde ortalama 5 kat, beş boyutlu objeler ile yapılan işlemlerde ise ortalama 4 kat hızlı olduğu görülmüştür.

Ayrıca altı bacaklı robotlar ile ilgili yapılmış çalışmalar da vardır. Candini ve arkadaşları 2009 yılında altı bacaklı bir robot geliştirip uygulamasını yaptılar [69]. Robot tasarımında iki eklemlili bacak yapısı ve analog giriş sinyali ile çalışan servo motorlar kullanılmıştır. Robot ayrıca basınç sensörü, akım sensörü gibi algılayıcılara ve bilgisayar ile kablosuz iletişim kurmak için bir radyo modülüne sahiptir. İki eklemlili bacak yapısı ile kinematik denklemler kullanılmadan, açık çevrim kontrol ile ayakların hareketi sağlanmıştır. Üzerine bir gözlem kamerası yerleştirilmiş ve görüntü uzaktan izlenmiştir.

Pa ve Wu tarafından, 2011 yılında yapılan bir çalışmada, Candini ve arkadaşlarının yaptığı çalışmanın bir benzeri, bilgisayar tabanlı bir uzaktan kontrol arayüzü ile birlikte geliştirildi [70]. Robot mesafe sensörü, sıcaklık sensörü, ivme sensörü gibi algılayıcılara ve bilgisayar ile kablosuz iletişim kurmak üzere bir radyo modülüne sahiptir. Yine iki eklemlili bir bacak yapısı ve kinematik denklemler kullanılmadan açık çevrim bir kontrol ile ayakların hareketi sağlanmıştır. Sensörlerden alınan veriler bilgisayar ekranında gösterilmiş ve bu ekrandan robotun kumandası sağlanmıştır. Çalışmada düşük maliyetli bir keşif robotu yapmak amaçlanmıştır.

2014 yılında Chen ve arkadaşları tarafından yapılan çalışmada ise her bir bacağı üç serbestlik dereceli altı bacaklı bir robot gerçekleştirilmiştir [28]. Çalışmada esas olarak üç eklemlili bacağın salınımı (havada iken izlediği yörünge) üzerinde durulmuştur. Bu robotun yapımında PID ile kontrol edilen servo motorlar tasarlanmış ve her ayağa basınç sensörü eklenmiştir. Her bir motor için (toplam 18 adet) sürücü ve her üç motor için bir ARM tabanlı bir mikrodenetleyici kullanılmıştır. Ayrıca bacakların koordinasyonunu sağlamak için de bir adet merkezi DSP kullanılmıştır.

Robot basınç sensörü dışında herhangi bir sensör ya da iletişim birimine sahip değildir.

Yine 2014 yılında Agheli önderliğinde yapılan çalışma da altı bacaklı bir robot üzerinedir [71]. Bu robot Chen'in önderliğinde yapılan çalışmaya çok benzemekle birlikte bir temel fark vardır. Bu çalışmada bacaklar beş serbestlik derecelidir. Bacağın her iki parçasına birer kayar eklem eklenerek gerektiğinde bacak boyunun uzaması sağlanmıştır.

Bu çalışmalara ek olarak son yıllarda uluslararası konferanslarda sunulan çalışmalara göz atabiliriz.

Petko ve arkadaşları 2013 yılında yaptıkları çalışmada, tasarladıkları bir CNC tezgâh için takım yolu belirleyici (trajectory generator) olarak FPGA tabanlı bir donanım geliştirdiler [72]. Düzgün takım hareketleri ve yumuşak hız geçişleri için yüksek hesap hızının gerekli olduğu vurgulanan çalışmada uygulama üç eksenli bir paralel robot üzerinde yapıldı.

2013 yılında, Ghorbel ve arkadaşları, tarafından kamera ile bir mobil robotun bir konumdan belirlenen bir başka konuma gitmesini sağlayan FPGA tabanlı bir donanım geliştirildi [73]. Liu ve ekibi 2012 yılında altı eksenli bir robot kol için cordic algoritması kullanarak, FPGA tabanlı bir kontrolcü geliştirdiler. Bu robot kolda FPGA ve DSP birlikte kullanıldı. Gac ve arkadaşları 2012 de bir paralel robotun ters kinematik hesaplarının hızlı yapılabilmesi için FPGA tabanlı bir donanım geliştirdiler [74].

Ding ve ekibi tarafından 2011 yılında yapılan çalışmada, FPGA altı serbestlik derecesine sahip bir Stewart Platformunun kontrolü için kullanılmıştır [75]. Altı serbestlik derecesi için altı adet motor kullanılmış ve her bir motorun kontrolü paralel olarak aynı çip ile yapılmıştır. Bu robotun istenen konumu alması için gerekli matematiksel işlemler yine aynı çip üzerinde çözülmüştür. Kullanılan FPGA konum kontrolü için motorların kodlayıcı (enkoder) bilgilerini ve yük algılaması için yük

hücrelerinden gelen bilgileri girdi olarak almaktadır. Tüm bunlar birleştirilerek biyomekanik bir test sistemi oluşturulmuştur.

Lee ve ekibi tarafından 2010 yılında bipedal robotlar üzerine yapılan araştırmada yeniden düzenlenebilir (reconfigurable) bir donanım tasarlamayı amaçlamıştır [76]. Robotun esnekliğinin temelinde ana kontrolcüsünün FPGA tabanlı olması yatmaktadır. Bu FPGA üzerine kullanıcı tarafından değiştirilebilecek lojikler yüklenebildiği gibi kendine özel gerçek zamanlı işletim sistemleri de yüklenebilmektedir. Bu sistem üzerinde çok çeşitli algoritmalar yürütülebilmektedir. Bu sistem özellikle saha uygulamalarında çok verimli olmuştur. Önerilen bu sistem basit bir yürüyen robot üzerinde başarıyla uygulanmıştır.

2009 yılında yapılan bir araştırmada Sanchez ve ekibi ileri kinematik denklemlerin çözümü için FPGA tabanlı bir kontrolcü geliştirerek Spherical Robot Manipulator üzerinde uygulama yapmıştır [77]. Bu uygulamada ağırlıklı olarak matematiksel çözümler ele alınmıştır ve simülasyon çalışması yapılmıştır.

Burada sunulmakta olan çalışmada, çok eklemlili robotik yapılara ait kinematik denklemlerin FPGA ile çözülmesine yönelik bir çalışma yapılmış ve bir robot üzerinde uygulanmıştır. Kinematik hesaplar karmaşık olması nedeniyle uzun zaman almakta ve çok eklemlili yapıların gerçek zamanlı hareket kontrollerinde tek işlemci ile istenen doğrulukta, hassasiyette ve sıklıkta sonuç almak zor olmaktadır. Özellikle serbestlik derecesinin artmasıyla birlikte ayrı bir hesaplama birimi kaçınılmazdır. Serbestlik derecesine paralel olarak elektrik motorları, hidrolik ya da pnömatik sistemler gibi eyleyicilerin de sayısı da artmaktadır. Bu eyleyicilerin kapalı çevrim kontrolleri, merkezi işlem birimine ek yükler getirmektedir. Ayrıca eyleyicilerin kontrolü ve beslemesi için kullanılan elektronik birimler ve bu birimler ile kontrolcü arasındaki kablolama tasarlanan fiziksel yapıyı karmaşık hale getirmektedir. Bu çalışmada bu problemlere çözüm getirecek bir kontrol donanımının tasarımı VHDL dili ve FPGA kullanılarak gerçekleştirilmiştir. Bu donanım, çevre birimlerle haberleşmek için seri haberleşme portlarına ve kinematik denklemlerin çözümü için yardımcı işlem birimine sahiptir. Tasarlanan bu donanım teknolojideki eğilimlere paralel olarak yaygınlaşan seri arabirimden kontrol edilebilen servo motorlar ile

çalıřacak řekilde özelleřtirilmiřtir. Seri arabirimden aldıđı ayak pozisyonları iin gerekli eklem aıllarını hesaplayarak bunları yine seri arabirimden servo motorlara gndermektedir. Bu donanım kinematik denklemlerin zm iin bir yardımcı iřlemci iermekte ve elektromekanik karmařayı azaltacak donanımlar ile birlikte kullanılabilir. Bu donanımın robotlar iin tasarımdan uygulamaya geiři hızlandıracak bir ara olarak kullanılması ngrlmektedir [78].

alıřmada bu donanımın uygulaması iin bir robot tasarlanmıřtır. Robot her bir bacağı  serbestlik derecesine sahip, altı bacaklı, rmcek tipi bir robottur. zerinde etrafındaki cisimleri algılayabilmesi iin drt adet kızıltesi sensr yerleřtirilmiřtir. Ayak ularına, yer ile teması ve yere uygulanan kuvveti algılamak amacıyla basın sensrleri yerleřtirilmiřtir. Yine ayak ularına, adım atma sırasında robotun karřılařabileceđi kk engelleri algılamak ve robotun grř aısını artırmak iin kızıl tesi iki sensr yerleřtirilmiřtir. Bu sensrler sadece n ayaklarda bulunmaktadır. Arka ayaklar ise n ayaklardaki sensr bilgisi dođrultusunda kontrol edilmektedir. Ayrıca robot zerine, uzaktan kumanda bilgisayarı ile iletiřim kurmak iin bir radyo modl yerleřtirilmiřtir. Bu modl vasıtası ile robot uzaktan ynlendirilebilmekte ve robot zerindeki sensr bilgileri bilgisayara iletilebilmektedir. Uzaktan kumanda bilgisayarı zerinde, robotu kontrol etmek ve sensr verilerini izleyebilmek iin bir kumanda arayz geliřtirilmiřtir.

Tasarlanan sistem zerinde bir yrme algoritması geliřtirilmiřtir. Algoritma, kural tabanlıdır, Matlab/Simulink ortamında geliřtirilmiř ve bir DSP zerine gmlmřtr. Bu algoritma ileri gitme, geri gitme, sađa dnme, sola dnme, sađa ve sola yan yrme gibi temel hareketleri yaparken ayakların koordinasyonunu sađlamaktadır. Algoritmanın ıktısı her bir ayak iin retilen ardıřık ayak pozisyonlarıdır. Bu pozisyonların FPGA zerinde geliřtirilen donanıma iletilmesiyle hareket geekleřtirilmiř olur. Algoritma ayak pozisyonlarını retirken, srekli olarak ayak ularındaki basın sensrlerini de kontrol etmekte ve zemin zerindeki engellere gre ayak yksekliklerini deđiřtirerek ortama adaptasyon sađlamaktadır. Algoritma aynı zamanda robot zerindeki kızıl tesi sensr verilerini de kullanmaktadır. Bu verileri deđerlendirerek, etraftaki nesnelere ve engelleri algılamakta ve yn tayinini buna gre yapmaktadır. Yaptıđı lmler sonucunda, engelin zerinden

aşılacağına karar verirse, ayak yüksekliklerini değiştirerek engeli aşmaktadır. Üzerinden geçemeyeceğini anlarsa, etrafından dolaşmayı denemektedir. Her iki şekilde de başarılı olmazsa, robotu durdurup, kumanda bilgisayarından yardım istemektedir.

Bu tez çalışmasını, yukarıda bahsedilen bazı çalışmalar ile kıyaslamak gereklidir. Barron-Zambrano ve ekibi tarafından yapılan çalışmada yapay sinir ağı tabanlı bir sistem kullanılmış ve iki serbestlik dereceli robotlar üzerinde, simülasyon ortamında denenmiştir [65]. Dolayısı ile bu tezde sunulan çalışmadan tamamen farklıdır ancak amaç ve kullanılan donanım açısından benzerlik göstermektedir.

Bir başka çalışmada, Candini ve arkadaşları tarafından, 2 serbestlik dereceli, 6 bacaklı bir robot tasarlanmıştır [69]. Bu çalışmada, kinematik denklemler ne ayrı bir donanım vasıtasıyla ne de mikrodenetleyici üzerinde kullanılmamıştır. Sistem analog servolar ile gerçekleştirilmiştir. Çalışmada yürüme algoritmasından bahsedilmemiştir. Burada sunulan çalışmada ise 3 serbestlik dereceli, 6 bacaklı bir robot tasarlanmıştır. Robot üzerinde, kinematik denklem çözümleri ve seri arabirimden kontrol edilebilen servoların kontrolü için tasarlanan, FPGA tabanlı bir donanım kullanılmış ve engellerin bulunduğu zeminlerde yürümeyi sağlayacak bir yürüme algoritması geliştirilmiştir. Pa ve Wu tarafından gerçekleştirilen çalışma da Candini önderliğinde yapılan çalışmaya benzemektedir [70].

Chen ve ekibi tarafından yapılan çalışmada, temel olarak bacak tasarımı ve salınımı ile ilgilenilmiştir [28]. Mekanik açıdan daha sağlam bir bacak yapısı önerilmiş ve bu bacakların uygun salınma yörüngeleri üzerinde durulmuştur. Agheli ve ekibi tarafından yapılan çalışmada ise uzayabilen bir bacak tasarımı gerçekleştirilmiştir [71]. Her iki tasarımda da çok sayıda motor sürücü ve mikro işlemci kullanılmıştır. Sensör donanımları zayıf, uzaktan kumanda birimi ve uzaktan kumanda arayüzünden mahrumdurlar.

BÖLÜM 3

ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA)

Bilgisayar ve elektronik alanında hesaplamaların yapılması için kullanılan çözümleri temel olarak donanımsal ve yazılımsal çözümler şeklinde ikiye ayırabiliriz. Uygulamaya özel bütünleşik devreler (ASICs) gibi donanımsal yapılar kritik hesapları bile çok hızlı gerçekleştirebilir ve uygun kaynak kullanımına sahiptir. Bununla birlikte belirli bir uygulama için özelleştirilmiştir, üretimden sonra geliştirilemez veya değiştirilemez. Tasarım ve üretimi milyonlarca dolara mal olmaktadır. Yazılımlar ise daha esnek bir yapıya sahiptir, uygulamaya göre yeniden düzenlenebilir ve çok sayıda işlevi yerine getirebilir olmalarıyla birlikte, ASIC ile kıyaslandığında, performans ve güç tüketimi açısından önemli oranda kötüdür [4].

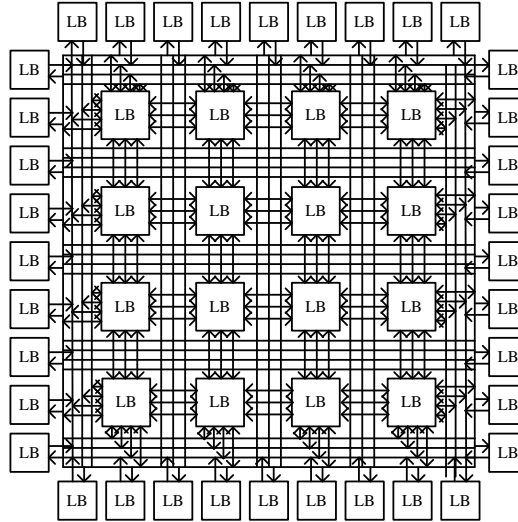
FPGA'ler ise geleneksel donanım ve yazılım kavramlarının birleşimidir. Donanımın hızına ve yazılımın esnekliğine sahiptir ve düşük maliyetlidir. Alanda programlanabilir ismi verilmesinin nedeni, mantık bloklarının ve ara bağlantıların imalat sürecinden sonra programlanabilmesidir.

FPGA'ler tasarımcının ihtiyaç duyduğu fonksiyonların işletilmesi, problemlerin çözülmesi, durum makinelerinin oluşturulması gibi birçok temel kavramın ve bu kavramların bir araya gelmesiyle oluşan büyük ölçekli çözümlerin, uygun bir maliyetle, yazılımsal olarak tasarlanıp, donanımsal olarak uygulanması amacıyla geliştirilmiş bir teknolojidir. İçerisinde bulunan her bir mantık bloğunun işlevi ve bu bloklar arasında bulunan bağlantıların düzeni kullanıcı tarafından değiştirilebilmektedir. Bu teknoloji kullanılarak başlı başına yazılımsal ya da donanımsal ihtiyaca yönelik çözümler üretilebileceği gibi, geleneksel yöntemlerle geliştirilen yazılımsal ya da donanımsal çözümlerin bir parçası olarak da kullanılabilir. Bugün FPGA'ler iletişim sistemlerinden robotik uygulamalara, tıp elektroniğinden savunma sanayine kadar birçok alanda kullanılmaktadır.

3.1. FPGA'İN YAPISI

FPGA'in genel yapısı lojik bloklar ve genel bağlantılardan oluşmaktadır. Lojik bloklar basit lojik denklem ve devreleri gerçekleştirmek için gerekli elemanları içerirler. Beş veya altı girişli fonksiyonların her biri bir lojik blok içerisinde gerçekleştirilebilir. Ara bağlantılar ise daha fazla girişe veya daha fazla lojik elemana ihtiyaç duyulduğunda blokları birbirine bağlamak için kullanılmaktadır. FPGA'in genel yapısı Şekil 3.1'de verilmiştir. Şekilde LB kısaltması lojik blokları, çizgiler ve oklar ise bloklar arasındaki tek ve çift yönlü bağlantıları göstermektedir.

FPGA içerisindeki mantıksal elemanların konfigürasyonu ve bu elemanlar arasındaki bağlantılar, programlanabilir devre elemanları ile düzenlenmektedir. Bunlar genellikle Flash veya SRAM teknolojisine sahip sigortalardır. SRAM teknolojisine sahip olan FPGA'ler piyasada en çok tercih edilen modellerdir. Tasarımdan sonra oluşturulan bir yapılandırma dosyası yardımı ile FPGA üzerindeki SRAM'lerin alması gereken değerler, ilk çalışma sırasında SRAM'lere yazılarak FPGA'in tasarlanan donanıma dönüşmesi sağlanır.



Şekil 3.1. FPGA'in genel yapısı.

FPGA üzerinde bir donanım tasarımı yapılırken genellikle, Verilog veya VHDL gibi, bir donanım tanımlama dili (hardware description language, HDL) kullanılır. Bununla birlikte bilgisayar programlamada kullanılan C/C++ gibi diller veya

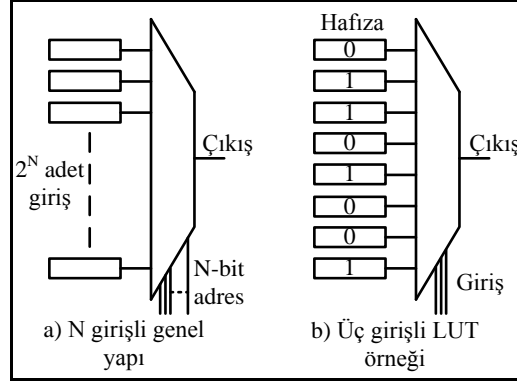
kullanılan diđer programlama yöntemleri ile de FPGA üzerinde donanım geliřtirmeyi sađlayan uygulamalar mevcuttur. Tasarım araçları ile oluşturulan donanım, kullanılacak FPGA'in özelliklerine göre optimize edilir. Bu sırada uygulanan lojik sentezleme işlemi ile yüksek seviyeli lojik bloklar mantık kapıları seviyesinde devrelere dönüřtürülür; sonra haritalama (mapping) adı verilen teknoloji ile bu kapılar FPGA'in kaynaklarını en uygun şekilde kullanmak için gruplara ayrılır. Daha sonra gruplanan bu kapılar, bloklar içerisine uygun şekilde yerleřtirilir ve bu işlemden sonra lojik bloklar arasındaki bađlantılar düzenlenir. Son işlemden sonra bir yapılandırma dosyası, ya da bit haritası oluşturulur. Bu dosyanın FPGA'e yüklenmesi ile tasarım gerçekleştirilmiř olur. Bu yükleme işlemi bir mikrodenetleyici ile veya FPGA'in programlama portuna bađlı bir EPROM (Erasable Programmable Read Only Memory, Silinip Programlanabilen Salt Okunur Bellek) ile gerçekleştirilir. Donanımın her çalıřtırılmasında bu yapılandırma dosyasının FPGA'e yüklenmesi gerekir ancak çalıřtırıldıktan sonra FPGA tasarlanan donanım olarak görev yapar.

3.2. DOĐRULUK TABLOSU

Herhangi bir hesaplama lojik eřitlikler kullanılarak yapılabilir. Her bir lojik eřitlik ise dođruluk tablosu ile ifade edilebilir. Aritmetik işlem birimleri, toplayıcılar veya karar verme eylemi gerektirecek diđer karıřık yapılar ise, if-then-else gibi řartlı ifadeler ile oluşturulabilir. Dolayısıyla herhangi bir algoritma dođruluk tabloları kullanılarak oluşturulabilir [4].

Dođruluk tabloları FPGA'in kalbidir. FPGA üzerinde bir dođruluk tablosunu işletebilecek genel amaçlı en küçük donanıma "lookup table" veya kısaca "LUT" denir. Devresel olarak düşünürsek LUT, N giriřli ve 1 çıkıřlı bir multiplexer ile N-bit hafıza elemanının birleřimidir. řekil 3.2.a'da tipik bir N-giriřli LUT görülmektedir. Bugün kullanılan FPGA'lerin içi bu LUT'lar ile doludur. řekil 3.2.b'de gösterilen 3 giriřli LUT (3-LUT) yapısını incelersek; giriř "000" ise çıkıř 0, giriř "010" ise çıkıř 1 olmakta ve bu şekilde dođruluk tablosunun karřılıđı olan fonksiyon ortaya çıkmaktadır. Kısaca N-bit giriř verisinin karřılık geldiđi adresteki 1-bit deđer çıkıřa

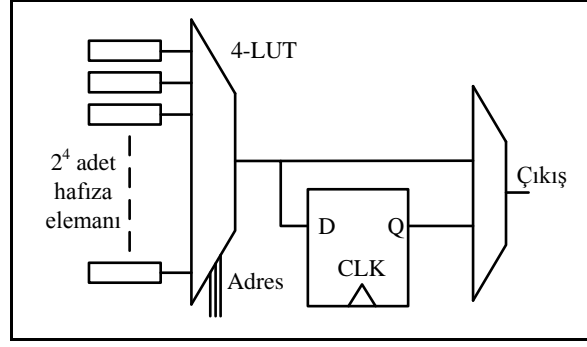
verilmektedir. Bu LUT'ların birçoğu bir araya getirilerek çok girişli ve daha karmaşık fonksiyonların gerçekleştirilmesi mümkündür.



Şekil 3.2. Doğruluk tablosunun yapısı.

Burada LUT birimlerinin giriş sayılarını değerlendirecek olursak; az sayıda giriş küçük fonksiyonların tek bir LUT ile gerçekleştirilmesi anlamına gelecektir. Büyük fonksiyonlar için çok sayıda LUT kullanılacak ve buda ara bağlantılardan kaynaklı sinyal gecikmelerini artıracaktır. Diğer taraftan LUT birimleri çok girişli olacak olursa; küçük fonksiyonlar kullanıldığında yapının bir kısmı atıl kalacaktır. Yapılan çalışmalarda fonksiyon büyüklüğü ve ara bağlantılardaki sinyal gecikmeleri birlikte göz önünde bulundurulduğunda en uygun yapının 4-LUT olduğu görülmüştür. Ancak ticari uygulamalarda farklı sayılarda girişlere sahip LUT içeren FPGA'ler mevcuttur. Ayrıca bazı uygulamalarda LUT giriş sayısını artırmak yerine birden fazla LUT aynı blok içerisine konarak hem birimlerin optimum kullanılması hem de, aynı birim içinde bağlantıların kısa olması nedeniyle, bağlantılardan kaynaklanan gecikmelerin daha az olması sağlanmaktadır.

Buraya kadar anlatılan yapı anlık çıkış verecek fonksiyonları çözmek için yeterlidir. Ancak sıralı lojik problemler söz konusu olduğunda veya kullanılan fonksiyonda önceki değerlere ihtiyaç duyulduğunda bu yapı yetersiz olmaktadır. Bunu sağlamak için her bir LUT çıkışına 1-bit D flip-flop bağlanmaktadır. Böylelikle Şekil 3.3'deki hali alan blok ile ister direkt olarak 4-LUT çıkışı, istenirse de D flip-flop ile oluşturulan hafıza çıkışı kullanılarak işlemler yapılabilir.

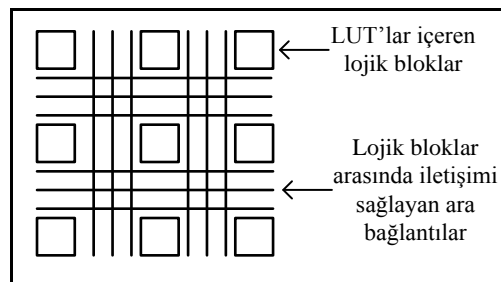


Şekil 3.3. Hafıza elemanı ile birlikte doğruluk tablosu.

3.3. ARA BAĞLANTILAR

LUT ve D flip-flop yapısı birlikte lojik blok veya fonksiyon bloğu olarak adlandırılır. Bu kısımda bu blokların büyük bir fonksiyonu gerçekleştirmek için nasıl bağlandıkları incelenecektir.

Bağlantılar için çoğunlukla kullanılan yapı “ada tipi” (island style) yapıdır. Şekil 3.4’de görülen bu yapıda lojik bloklar birer ada gibi düşünülebilir ve bu adalar ara bağlantılardan oluşan bir denizin içerisinde yer almaktadır. Burada görülen bağlantıların ayrıntıları komşular arası bağlantı, gruplar arası bağlantı ve hiyerarşik bağlantı olmak üzere üç kısımda incelenebilir.



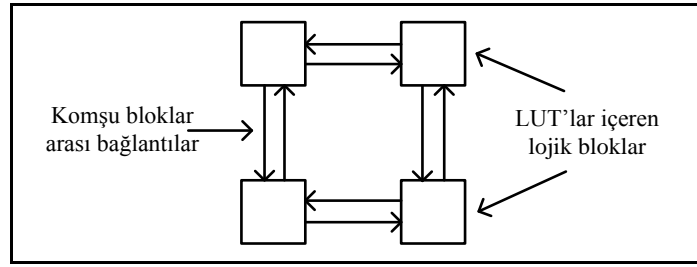
Şekil 3.4. Ada tipi yapının şematik olarak tanımı.

3.3.1. Komşular Arası Bağlantı

En basit bağlantılardan biridir. Bu bağlantıyı anlamak için Şekil 3.5’de verilen 2x2 lojik blok dizisine göz atalım. Görüldüğü gibi dört yönde komşular arası iki yönlü

bağlantı vardır. Bu bağlantılar sayesinde bloklar birbirleri ile hızlı bir şekilde iletişim kurabilirler. Bu eski FPGA’lerde kullanılan bir yapıdır ve zorlu bağlantılar ve yüksek gecikmeler gibi bazı dezavantajları vardır. Bu yapının 2048 x 2048 boyutlarında bir matrise uygulandığı düşünüldüğünde problemlerin ne kadar büyüdüğü görülebilir. En yakın komşu ile kurulan bağlantı dışında, sinyal gecikmesi bağlantı uzunluğu ile birlikte doğrusal olarak artacaktır. Çünkü sinyal ilerlerken birçok hücreden ve anahtardan geçerek gitmek zorundadır. Bu anahtarlar lojik bloklar arasındaki bağlantıları özelleştirmek ve sinyali yönlendirmek için kullanılan anahtarlardır.

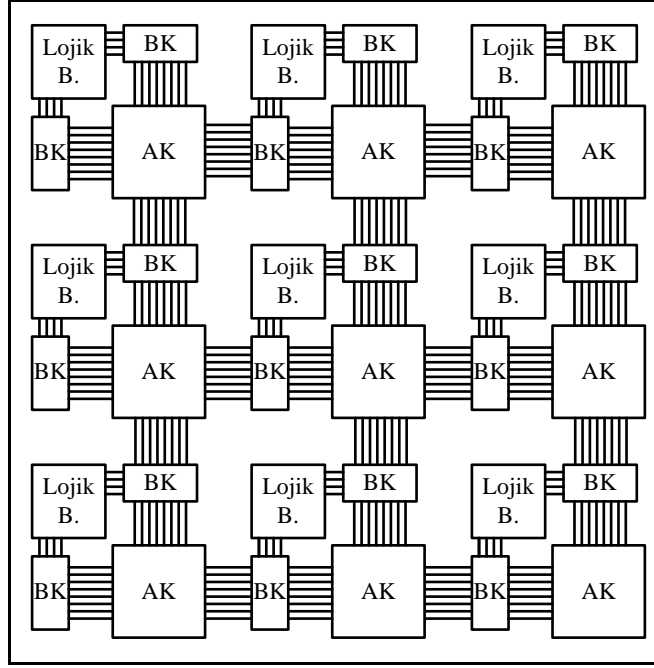
Bu yapı günümüzde tek başına kullanılmamaktadır. Yakın komşular arasındaki etkinliğine karşın artan mesafe ile birlikte artan dezavantajlar nedeniyle grup ve hiyerarşik yapılar ile birlikte kullanılmaktadır.



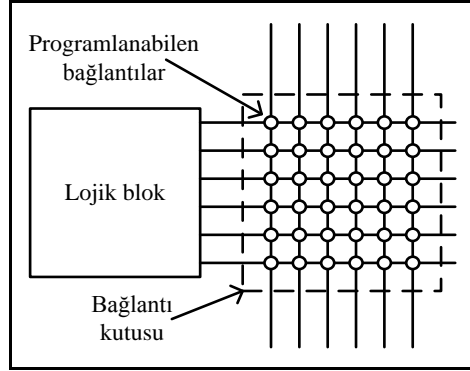
Şekil 3.5. Komşular arası bağlantı örneği.

3.3.2. Gruplar Arası Bağlantı

Günümüzde kullanılan FPGA yapıları daha çok Şekil 3.6’da verilen yapıya benzemektedir. Görüldüğü gibi bağlantı kutuları (BK) ve anahtarlama kutuları (AK) adı verilen yapılar kullanılmıştır. Bağlantı kutusu aracılığı ile lojik bloklar (Lojik B.) yakınındakiler ile bağlantı kurmaktadır. Bu kutular içerisinde, blokların giriş-çıkış uçlarını birbirine bağlamak için programlanabilen anahtarlar ve çoklayıcılar (multiplexer) mevcuttur. Ayrıntısı Şekil 3.7’de verilen bu blok, giriş ve çıkışları birbirine bağlamak için yatay ve dikey olarak yerleştirilmiş hatları kullanır ve bu sayede oldukça esnek bağlantılar oluşturabilir.

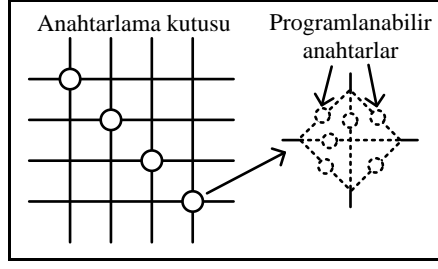


Şekil 3.6. Bağlantı kutuları ve anahtarlama kutuları içeren FPGA yapısı.



Şekil 3.7. Bağlantı kutusunun yapısı.

Şekil 3.6'da verilen yapıda yatay ve dikey bağlantıların kesiştiği yerlerde görülen anahtarlama kutularının yapısı Şekil 3.8'de verilmiştir. Bu yapı programlanabilen anahtarlardan oluşmakta ve bir yolu diğerine bağlamak için kullanılmaktadır. Örnek olarak gelen sinyalin tam, bağlantı noktasında, düz bir şekilde ileri gitmesi veya yönü değiştirilerek aşağı doğru ilerlemesi sağlanabilir. Burada örnek olarak verilen bu yapı çok daha karmaşık haller alabilmektedir.

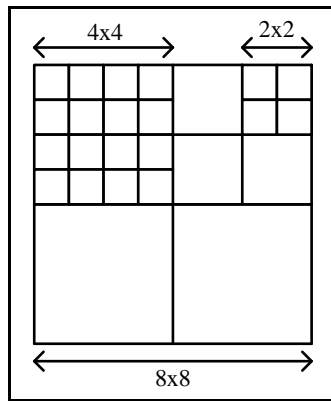


Şekil 3.8. Anahtarlama kutusunun yapısı.

Görüldüğü gibi gruplar arası kurulan bağlantılar ile sistem daha esnek ve verimli hale getirilmektedir. Komşular birbirine direkt olarak, kısa yollar üzerinden bağlanabilmekte ve gerektiğinde komşulardan oluşan gruplar birbirleri ile bağlantı kutuları kullanılarak, ne kadar uzakta olsa da bağlanabilmektedir. Böylelikle sinyal gecikmeleri azaltılmakta ve sistem daha fonksiyonel olmaktadır.

3.3.3. Hiyerarşik Bağlantı

Sinyal gecikmesini azaltmak için kullanılan bir başka yapıda hiyerarşik bağlantıdır. Şematik bir gösterimi Şekil 3.9’da verilen bu yapıyı inceleyecek olursak; hiyerarşinin en altında 2x2’lik bir diziyi oluşturan lojik bloklar tek bir küme olarak gruplandırılmıştır. Bu blok içerisinde bütün komşular arası bağlantılar mevcuttur. Bir üst seviyede 4x4’lük kümelerin oluşturduğu yeni küme 16 lojik blok içermektedir. Bu seviyede kümeler arasındaki bağlantılar bir alt seviyeye göre daha uzundur. Bu yapı tekrarlanarak daha büyük kümeler oluşturulur ve kümeler büyüdükçe ara bağlantıların uzunlukları artar.



Şekil 3.9. Hiyerarşik bağlantı.

Bu yapı içerisinde iyi tasarlanmış ve FPGA üzerine iyi yerleştirilmiş bir devre çoğunlukla yerel bağlantıları kullanır ve az sayıda sinyalin uzun mesafeler kat etmesi gerekir. Bu yapıda hiyerarşik gruplar arası bağlantıları sağlamak için var olan anahtar kutuları genişletilebilir ya da tamamen yeni anahtar kutuları oluşturulabilir.

3.4. FPGA'İN PROGRAMLANMASI

Ticari olarak kullanılan FPGA'lerde, lojik bloklarda olduğu gibi, ara bağlantıları oluşturan her bir anahtarlama noktası programlanabilir. Bağlantı kutuları içerisinde bulunan programlanabilir çoklayıcılar (multiplexer), her bir lojik bloğun giriş-çıkış uçlarını yönlendirirler. Anahtarlama kutusunda ise yatay ve dikey bağlantılar arasında sinyaller anahtarlar aracılığı ile yönlendirilir. Sonuç olarak bağlantı yolları üzerindeki anahtarlar kullanılarak farklı uzaklıktaki gruplar hiyerarşik seviyeye göre birleştirilmiş olur. Bütün bu birleşim ve yönelimler programlanabilir anahtarlar ile yapılır. Programlanabilen anahtarların durumları modern FPGA'lerde, SRAM'ler kullanılarak hafızada tutulur ve bütün çalışma süresince uygulanır.

3.5. FPGA'LERİN SAĞLADIĞI AVANTAJLAR

FPGA'ler donanımsal yapısı sayesinde yazılımdan hızlı olması ve üzerinde yeni donanımlar oluşturulabilmesi sayesinde kazandığı esneklik ile diğer geleneksel yazılım ve donanım geliştirme yöntemlerine güçlü bir alternatiftir. Yeniden düzenlenebilir donanımlar sayesinde geleneksel olarak yazılıma atfedilen esneklik aynı derecede donanıma da uygulanabilmektedir [5]. Birçok uygulama için en hızlı sonuçları üreten ve en az kaynak harcayan alternatif FPGA'lerdir [6]. Bu anlamda FPGA'ler uygulamaya özel bütünleşik devrelerin (ASIC) bir evrimi niteliğindedir. Ayrıca üzerlerine yazılımsal mikrodenetleyiciler gömülebildiği için klasik yazılım tabanlı çözümler ile birlikte donanım tasarımlarının tek bir çip üzerinde gerçekleştirilebilmesine imkan verirler.

Tüm bunlar ile birlikte FPGA'ler için bazı avantajlar şu şekilde özetlenebilir [8]:

1. En az şu üç nedenden ötürü maliyeti düşürür;

- Algoritmaların ve özel ihtiyaçların giderilmesine yönelik geliştirilen bir mimari içinde kullanılması ve uygulanması
 - Çok yüksek seviyede uzmanlık gerektiren metotların veya spesifik metotların geliştirilip uygulanma ve pazara sürülme zamanı
 - Analog arayüzler dahil, bir kontrol sisteminin tamamının, bir bütünleşik devre olarak geliştirilmesinin istenmesi
2. Gizlilik derecesini artırır; firmanın özel becerileriyle birleştiğinde, özel bir tasarımın kopyalanması kolay değildir.
 3. Uçak uygulamalarında olduğu gibi birçok kısıtlamanın bulunduğu gömülü sistemler; güç sarfiyatı, ısı açığa çıkarma, güvenilirlik, elektromanyetik geçirmezlik gibi avantajları vardır.
 4. Kontrol performansının artırılmasını sağlar. Örnek olarak paralel komut işletme zamanı inanılmaz derece azaltılabilmekte, FPGA tabanlı kontrolcülerde analog parçaların performans seviyesine ulaşılmasına izin verilmektedir ve dinamik olarak yeniden programlanabilmektedirler.

BÖLÜM 4

ROBOT KİNEMATİĞİ

Robot kinematığı eklemlerin hareketi ile bu hareketler neticesinde robotta meydana gelen genel hareket arasındaki ilişkiyi tanımlar [79]. Robot, mekanik bir bakış açısıyla, birbirinden bağımsız katı parçaların döner veya kayar eklemler ile birbirine bağlanmasından oluşan bir zincirdir [32]. Zincirin bir ucu sabit bir gövdeye, diğer ucu ise uzayda serbestçe hareket eden ve kontrolü hedeflenen uç işlevciye bağlıdır. Bu zincirde her bir eklemin hareketi bir diğerini etkiler ve tüm bu hareketlerin bileşimi uç işlevcinin hareketini oluşturur. Dolayısıyla kinematik denklemler uç işlevci ile eklemler arasındaki ilişkiyi tanımlar. Robotun kuvvet, hız ve ivme analizinin yapılmasında robot kinematığı kullanılır.

Eklemlerde dönme hareketinden kaynaklanan yer değiştirmeye eklem açısı, doğrusal yer değiştirmeden kaynaklanan ötelemeye ise eklem kayması denir. Üç boyutlu uzayda nesnelere yönelimi ve konumu, her bir nesnenin merkezine yerleştirilen koordinat sistemine göre tanımlanır. Robotlarda da bu işlem her bir ekleme koordinat sistemi tanımlayarak yapılır. Tanımlanan bu koordinat sistemleri sayesinde hem robot zeminine göre uç işlevcinin konum ve yönelimi hem de aynı uzayda bulunan diğer nesnelere göre uç işlevcinin konum ve yönelimi belirlenebilir.

Uzayda herhangi bir nokta denklem (4.1) gibi bir matris ile ifade edilir. Uzayda bir noktanın ötelenmesi, bir başka noktaya taşınması, işlemi ise basitçe, denklem (4.2) 'de verildiği gibi, ötelenecek olan noktanın öteleme vektörü ile toplamından ibarettir.

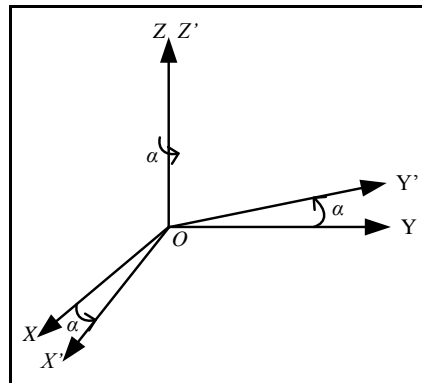
$${}^A P = \begin{bmatrix} {}^A P_X \\ {}^A P_Y \\ {}^A P_Z \end{bmatrix} \quad (4.1)$$

$$P_{yeni} = P_{eski} + P_{öteleme} \quad (4.2)$$

Uzayda bir noktaya öteleme işlemi dışında birde, bir nokta etrafında döndürme işlemi uygulamak gerekebilir. Bu işlem dönüşüm matrisi ile yapılır ve dönüşüm matrisi R, denklem (4.3)'de verildiği gibi tanımlanır. İşlem yapılan noktayı koordinat sisteminde belirli bir açı miktarınca döndürme işlemini gerçekleştirir. R matrisinde yer alan her sütun üç boyutlu uzayda bir noktayı ve satırlar sırası ile X, Y ve Z eksenlerini temsil etmektedir. Noktalar yerine noktaların ilgili eksenler üzerindeki yerlerini belirleyecek fonksiyonlar yerleştirildiğinde dönüşüm matrisi elde edilmektedir [79]. Çok eklemli yapılarda her bir eklem başlangıç noktasına yerleştirilen koordinat sistemi yardımı ile her uzvun konumu tanımlanır ve bu konuma öteleme ve döndürme işlemleri yapılarak uzayda hareketi sağlanır.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (4.3)$$

Bir referans koordinat sisteminin bir eksen etrafında α açısı kadar döndürüldüğünü düşünelim. Örnek bir işlem Şekil 4.1'de gösterilmiştir. O-X'Y'Z' koordinat sisteminin birim vektörleri referans sistem olan O-XYZ sistemine göre denklem (4.4), (4.5) ve (4.6) 'da olduğu gibi gösterilir.



Şekil 4.1. Koordinat sisteminin Z eksenini etrafında alfa açısı kadar dönmesi.

$$x' = \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix} \quad (4.4)$$

$$y' = \begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{bmatrix} \quad (4.5)$$

$$z' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.6)$$

Referans sistemden O-X'Y'Z' sistemine geçmek için gerekli dönüşüm matrisi ise denklem (4.7) verildiği gibi olacaktır. “Z” alt indisi sistemi Z eksenini etrafında döndürdüğümüzü göstermektedir. Benzer şekilde Y ve X eksenleri etrafındaki dönüşümler (4.8) ve (4.9) numaralı denklemlerde gösterildiği gibi olacaktır.

$$R_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

$$R_y(\alpha) = \begin{bmatrix} \cos(\beta_1) & 0 & -\sin(\beta_1) \\ 0 & 1 & 0 \\ \sin(\beta_1) & 0 & \cos(\beta_1) \end{bmatrix} \quad (4.8)$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta_2) & -\sin(\beta_2) \\ 0 & \sin(\beta_2) & \cos(\beta_2) \end{bmatrix} \quad (4.9)$$

Birden fazla dönüşüm matrisinin kullanılması durumunda, son eklem başlangıca göre dönme miktarını bulmak için, bu matrisler uzuv sırasıyla baştan sona doğru çarpılırlar. Artarda gelen üç eklem dönüşüm matrisleri sırasıyla R_1^0 , R_2^1 , R_3^2 ise

bunların çarpımları bize ilk referansa göre son koordinat sisteminin dönme miktarını verecektir.

$$R_1^0 = R_1^0 . R_2^1 . R_3^2 \quad (4.10)$$

Bir nokta için dönüşüm matrisi ile birlikte yönelim vektörü tek bir matris kullanılarak ifade edilebilir. Denklem (4.11)'de gösterilen bu matrise Homojen Dönüşüm Matrisi (Homogeneous Transformations Matrix) adı verilir. 0_1^0 ifadesi, 0 numaralı referans sisteminin koordinatları içerisinde yer alan 1 numaralı sistemin merkez noktasını ifade etmektedir.

$$A_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix} \quad (4.11)$$

Bu denklemi daha açık bir şekilde yazacak olursak (4.12) numaralı denklemi elde ederiz.

$$A_1^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & Px \\ r_{21} & r_{22} & r_{23} & Py \\ r_{31} & r_{32} & r_{33} & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

Robot kinematiği iki kısımda incelenir:

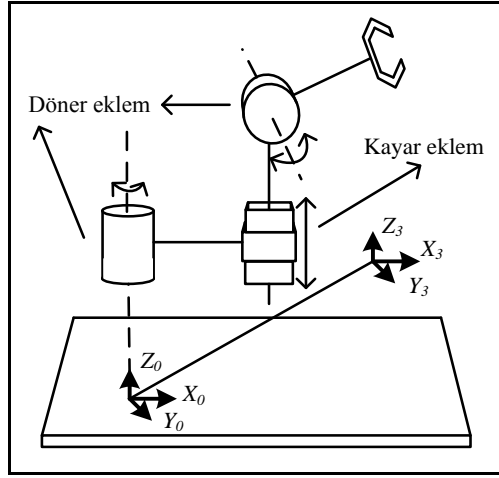
1. İleri Kinematik
2. Ters Kinematik

4.1. İLERİ KİNEMATİK

Çok eklemlili yapılar birbirine eklemler ile bağlanırlar. Bu eklemler döner eklem veya kayar eklem olabilir. Bu eklemlerin geleneksel gösterimleri Şekil 4.2'de verilmiştir. İleri yön kinematiğinde eklem dönüşüm matrislerinin ardı ardına çarpılmasıyla ana çerçeve ile araç çerçevesi arasında ilişki tanımlanır [80]. Kinematik zincirin bir

ucunda zemin, diğer ucunda ise istenen konuma taşınabilecek uç işlevci yer almaktadır.

Böyle bir yapının matematiksel modeli serbestlik derecesi ile karakterize edilir. Her bir serbestlik derecesi, değişkenleri de içeren eklem tanımlamalarına sahiptir. İleri kinematikğin amacı uç işlevcinin konum ve yönelimini eklem değişkenlerinin bir fonksiyonu olarak tanımlamaktır.



Şekil 4.2. Uç işlevcinin konumu ve yer değişimi.

Şimdi $n+1$ parçadan oluşan bir yapı düşünelim. İlk parça genellikle zemine sabitlenmiştir ve yapı n serbestlik derecesine sahiptir. Bu durumda uç işlevcinin ilk parçaya göre pozisyon ve yer değişimi denklem (4.13) ile verilir.

$$A_n^0 = A_1^0(\alpha_1)A_2^1(\alpha_2)\dots A_n^{n-1}(\alpha_n) \quad (4.13)$$

Görüldüğü gibi ileri kinematik hesabı, her biri tek bir ekleme ait değişkenlerin fonksiyonu olan ve sistematik bir biçimde tekrarlayan homojen dönüşüm matrislerinin çarpımından oluşmaktadır.

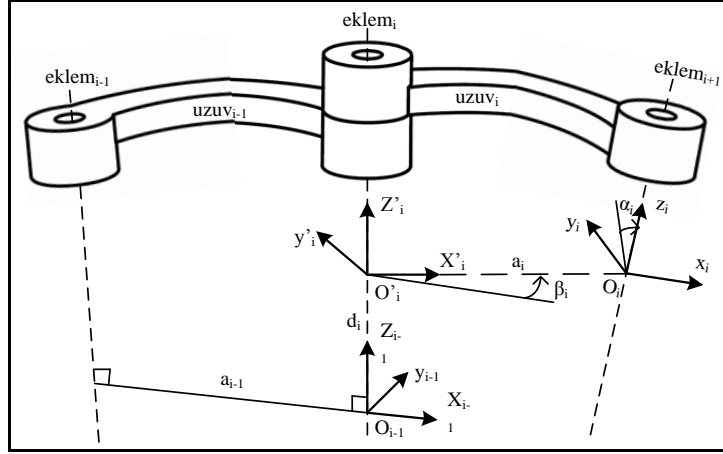
4.1.1. Denavit–Hartenberg Yöntemi

İleri kinematik denklemleri oluşturup hesaplarırken (4.13) denklemi gibi kendini tekrarlayan hesaplamalar için genel ve sistematik bir metot tanımlanabilir. Burada problem iki bağlantı arasında parametrelerin belirlenmesi ve dönüşümlerin hesaplanmasıdır. Bu yöntem ile her bir uzva bağlı bir koordinat sistemi bulunur. Ardışık iki koordinat sistemi arasındaki ilişkiyi tanımlayan homojen transformasyon matrisleri kullanılarak, mafsallardaki her bir uzvun koordinat sistemi bir önceki uzvun koordinat sistemine dönüştürülür.

Ardışık dönüşüm ile tüm uzuvların transformasyonu yapılarak, her bir uzvun konum ve yönelimi, taban koordinatları cinsinden ifade edilir. Bu durumda her bir ekleme, X_i, Y_i, Z_i ($i=0, 1, 2, 3...N$) şeklinde koordinat sistemi yerleştirilir. N serbestlik derecesini gösterir. N serbestlik derecesine sahip bir yapı için $N+1$ koordinat sistemi tanımlıdır. Denavit Hartenberg yöntemi ile bu koordinat sistemleri kullanılarak sistemin en uç noktasında bulunan uzvun konum ve yönelimi, zemine yerleştirilmiş olan sabit koordinat sistemi üzerindeki konum ve yönelim olarak elde edilir. Bu eksen takımları kullanılarak uzuvların birbirine göre durumları da tanımlanabilir. Her bir eksen takımı ait olduğu uzvun numarasını alır. Numaralandırma sıfırdan başlayarak yapılır, sıfır numaralı eksen zemine, robot gövdesinin hareket etmeyen bir yerine yerleştirilir. Genellikle sıfır numaralı eksen takımı ile 1 numaralı eksen takımları merkezleri çakışık olarak yerleştirilirler.

Şekil 4.3’de verilen şekil üzerinden tanımlayacak olursak, $i-1$ uzvu ile i uzvunun birleştiği eksen i ekseni olarak tanımlanır [32,79].

1. $i+1$ ekseni boyunca z_i ekseni belirlenir.
2. O_i merkezi, z_{i-1} ile z_i arasındaki normalin z_i ile kesişim noktasına yerleştirilir, O_i' ise z_{i-1} ile normalin kesişim noktasına yerleştirilir.
3. x_i ekseni z_i ve z_{i-1} arasındaki normal doğrultusunda, yönü i uzvundan $i+1$ uzvuna doğru olacak şekilde yerleştirilir.
4. Sağ el kuralına göre y_i ekseni yerleştirilir.

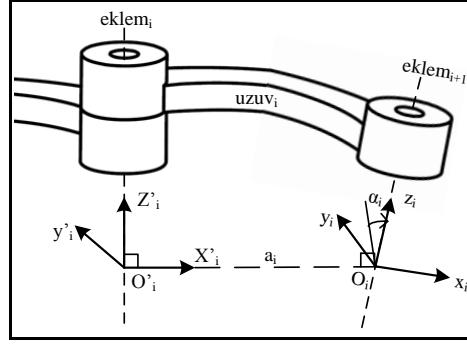


Şekil 4.3. Denavit-Hartenberg parametreleri.

Aşağıda Denavit-Hartenberg tanımlamalarına göre ortaya çıkan özel durumlar verilmiştir.

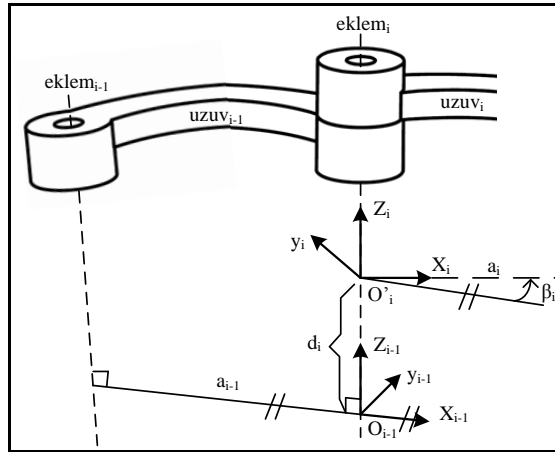
1. 0 numaralı kısımda, z_i eksenini belirlendikten sonra O_0 ve x_0 keyfi olarak seçilebilir.
2. n numaralı kısımdan sonra $n+1$ numaralı uzuv yok ise, x_n z_{i-1} numaralı eksene normal olmak koşulu ile z_n ayrıca tanımlanmak zorunda değildir.
3. Tipik olarak z_n döner eksenidir ve z_{n-1} yönünde yerleştirilmiştir.
4. Artarda gelen eksenler birbirine paralel ise bu eksenler arasında tanımlı birden fazla normal mevcuttur.
5. Artarda gelen iki eksen kesişiyorsa, x_i 'nin yönü keyfi olarak belirlenir.
6. i numaralı eklem kayar eklem ise z_{i-1} ekseninin keyfi olarak tanımlanabilir.

Eklemelere yerleştirilen bu eksenlerin birbirine göre tanımlanması için uzuv uzunluğu (a_i) ve uzuv açısının (α_i) bilinmesi gerekir. Uzuv uzunluğu iki eklem eksenindeki mesafedir. Bu mesafe, Şekil 4.4'de gösterildiği gibi, her iki eksene de dik olacak şekilde belirlenir. α açısı (bükülme açısı) ise (i)'inci uzvun z ekseninin, ($i+1$)'inci uzvun z eksenini ile yaptığı açı olarak tarif edilebilir.



Şekil 4.4. Uzunluk ve bükülme açısı.

Uzuvların birbiri ile ilişkisini incelerken tanımlanması gereken diğer kavramlar ise uzuv açıklığı (d_i) ve dönme açısı (θ_i)'dir. Bu kavramlar Şekil 4.5'de gösterilmiştir. Uzunluk açıklığı ardışık iki ekleme yerleştirilen koordinat sistemlerinin, x eksenleri arasındaki dikey (z eksenini doğrultusunda) mesafedir. Ekleme açısı ise bu x eksenlerinin uzantıları arasındaki açıdır.



Şekil 4.5. Uzunluk açıklığı ve dönme açısı.

Görüldüğü gibi bir robota ait kinematik tanımlamaların yapılabilmesi için, her ekleme ait dört parametrenin bilinmesi gerekmektedir. Bu parametrelerden ikisi uzunluğunu kendisini, diğer ikisi de başka uzuvlar ile olan ilişkisini tanımlamaktadır. Doğabilecek bazı belirsizlikler kuralları basitleştirmek suretiyle ortadan kaldırılabilir, mesela artarda gelen iki eksen her zaman paralel yapılabilir [32]. Bağlantılara eksenler yerleştirildikten sonra i numaralı kısım ile $i-1$ numaralı kısım arasında ki parametreler aşağıdaki şekilde tanımlanır.

1. a_i O_i ve O_i' arasındaki uzaklık,
2. d_i z_{i-1} boyunca O_i' 'nin konumu,
3. α_i x_i ekseninin referansına göre z_{i-1} ve z_i arasındaki açı (saatin tersi yönünde pozitifdir),
4. β_i z_{i-1} ekseninin referansına göre x_{i-1} ve x_i arasındaki açı.

a_i ve α_i daima sabittir ve uzuv ve eklemlerin geometrik yapısına bağlıdır. Diğer iki parametreden ise, eklem yapısına göre, yalnızca biri değişkendir; eğer eklem döner eklem ise β_i kayar eklem ise d_i değişkendir.

Bu noktadan sonra iki eklem arasındaki dönüşüm tanımlanabilir. İlk olarak z eksenini boyunca d_i kadar ilerleme ve yine z eksenini etrafında β_i kadar dönmeyi sağlayacak homojen dönüşüm matrisi (4.14) numaralı denklemde verilmiştir.

$$A_i^{i-1}(v_i) = \begin{bmatrix} \cos(\beta_i) & -\sin(\beta_i) & 0 & 0 \\ \sin(\beta_i) & \cos(\beta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

Daha sonra aynı noktayı x eksenini boyunca, a_i kadar ilerletecek ve yine x eksenini etrafında, α_i kadar dönmeyi sağlayacak homojen dönüşüm matrisi (4.15) numaralı denklemde verilmiştir.

$$A_i^i(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

Bu iki denklemin çarpımı ise bize eklem genel dönüşüm matrisini verecektir.

$$A_i^{i-1} = A_i^{i-1} \cdot A_i^i \quad (4.16)$$

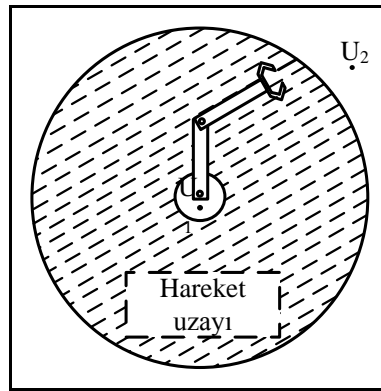
4.2. TERS KİNEMATİK

İleri kinematik uç işlevcinin konumu ve yöneliminin eklem değişkenleri ile olan bağlantısını vermektedir. Ters kinematik ise verilen bir uç işlevcinin konum ve yönelimi için eklem değişkenlerini belirlemeyi konu alır [80]. Bu problemin çözümü, uç işlevciyi uzayda istediğimiz yörüngede hareket ettirebilmemiz için temel gerekliliktir.

İleri kinematik denklemler kolay bir şekilde ortaya konabilmektedir ancak ters kinematik denklemler aşağıdaki nedenlerden ötürü daha karmaşık olmaktadır;

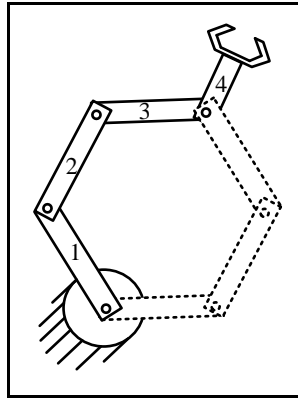
1. Denklem çözümleri genellikle doğrusal değildir,
2. Herhangi bir değişken için birden fazla sonuç ortaya çıkabilmektedir,
3. Kinematik yapıdan ötürü uygulanamaz sonuçlar ortaya çıkabilmektedir,
4. Kinematik yapıdan ötürü sonsuz sayıda çözüm ortaya çıkabilmektedir.

Uç işlevcinin ulaşması istenen konum hareket uzayı içerisinde ise çözümün varlığından söz edilebilir. Şekil 4.6'da gösterilen robot kolun hareket uzayı taralı alan ile ifade edilmiştir. U_1 ve U_2 noktaları çalışma uzayı dışındadır ve bu noktalar için ters kinematik çözüm mevcut değildir. Taralı alan ise robotun fiziksel olarak erişebileceği alanı ifade etmektedir ve bu alanda alacağı her bir pozisyon için kinematik çözüm veya çözümler mevcuttur.



Şekil 4.6. Robot kolun hareket uzayı.

Bazı durumlarda serbestlik derecesine ek olarak belirsiz Denavit–Hartenberg değişkenlerinin sayısındaki artış da bir değişken için bulunan sonuçların sayısının artmasına neden olmaktadır. Dört eklemlü düzlemsel bir robot kol için iki farklı sonuç Şekil 4.7’de gösterilmektedir. Görüldüğü gibi robot elin aynı konum ve yönelimi için birbirinden farklı iki kol pozisyonu meydana gelmektedir. Böyle bir durumda, yani üretilen çözümlerden birden fazlası robotun fiziksel sınırları dahilinde uygulanabilecekse, daha kısa zamanda elde edilebilecek olan, daha az enerji harcayan ya da robotun yapacağı iş için daha uygun olan sonuçlar seçilir ve uygulanır. Bazı durumlarda çözümlerden birisi robotun fiziksel yapısı veya çevredeki nesnelerin konumundan dolayı uygulanamaz durumdadır ve doğal olarak diğer çözümler değerlendirilir ve uygun olan seçilir. Genelde eklem sayısı arttıkça muhtemel çözüm sayısı da artmaktadır. Altı serbestlik dereceli bir robot kol için uygulanabilir 16 çözüm vardır.

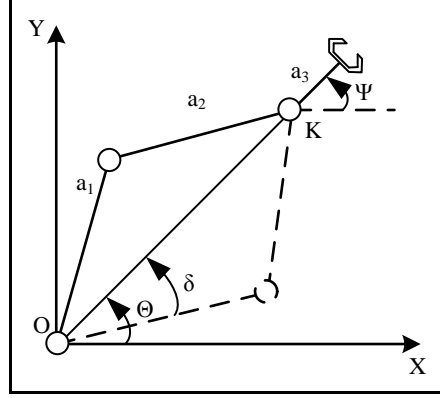


Şekil 4.7. İki farklı ters kinematik çözüm.

Ters kinematik denklemlerin çözümleri için genel bir yöntem yoktur. Genellikle denklemler doğrusal değildir ve dereceleri robotun serbestlik derecesine göre değişmektedir. Kullanılan yöntemler genellikle geometrik veya cebirsel yöntemler olmaktadır ancak her zaman net bir ayırım yapılamaz; yöntemler birçok zaman beraber kullanılmaktadır.

Ters kinematik denklemlerin çözümüne örnek olması için üç eklemlü bir düzlemsel robot kol ele alınmıştır [32]. Robot kolün örnek çizimi ve tanımlanan parametreler Şekil 4.8’de verilmiştir. Öncelikle cebirsel yöntem ile çözüm verilmiştir. Bu yöntemi

uygulamak için (4.17) denkleminde verilen yönelim tanımlaması yapılmıştır. Bu bir kısıtlamadır ve yapılmadığı takdirde sonsuz çözüm ortaya çıkmaktadır.



Şekil 4.8. Üç eklemlili bir düzlemsel robot kol.

$$\psi = \beta_1 + \beta_2 + \beta_3 \quad (4.17)$$

En başta (4.18) ve (4.19) denklemleri kullanılarak K noktasının x ve y bileşenleri elde edilmiştir.

$$P_{Kx} = P_x - a_3 c_\psi = a_1 c_1 + a_2 c_{12} \quad (4.18)$$

$$P_{Ky} = P_y - a_3 s_\psi = a_1 s_1 + a_2 s_{12} \quad (4.19)$$

K noktasının x ve y bileşenlerinin kareleri toplamını elde edip,

$$P_{Kx}^2 + P_{Ky}^2 = a_1^2 + a_2^2 - 2a_1 a_2 c_2 \quad (4.20)$$

Buradan c_2 (4.21) denkleminde verildiği gibi elde edilir.

$$c_2 = \frac{P_{Kx}^2 + P_{Ky}^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad (4.21)$$

Bir açının sinüs ve kosinüs değerlerinin kareleri toplamının bire eşit olmasından yola çıkılarak s_2 , (4.22) denklemi ile elde edilir.

$$s_2 = \pm\sqrt{1-c_2^2} \quad (4.22)$$

Artık s_2 ve c_2 bilindiğine göre β_2 atan2 fonksiyonu kullanılarak (4.23) denklemi elde edilir.

$$\beta_2 = A \tan 2(s_2, c_2) \quad (4.23)$$

s_1 ve c_1 'i bulmak için (4.18) ve (4.19) denklemleri kullanılarak (4.24) ve (4.25) denklemleri elde edilir. Bu denklemlerde s_2 ve c_2 zaten bilinmektedir.

$$s_1 = \frac{(a_1 + a_2 c_2) P_{Ky} - a_2 s_2 P_{Kx}}{P_{Kx}^2 + P_{Ky}^2} \quad (4.24)$$

$$c_1 = \frac{(a_1 + a_2 c_2) P_{Kx} + a_2 s_2 P_{Ky}}{P_{Kx}^2 + P_{Ky}^2} \quad (4.25)$$

Bu şekilde s_1 ve c_1 de bulunduğuna göre β_1 atan2 fonksiyonu ile (4.26) denklemi kullanılarak hesaplanabilir. Başta, (4.17) denklemi ile yapılan kısıtlamadan yola çıkarak, β_3 (4.27) denklemi ile hesaplanabilir ve böylece tüm bilinmeyenler bulunmuş olur.

$$\beta_1 = A \tan 2(s_1, c_1) \quad (4.26)$$

$$\beta_3 = \psi - \beta_1 - \beta_2 \quad (4.27)$$

Alternatif bir yöntem olarak geometrik yöntem ile çözüm incelenebilir. Öncelikle uzvun yönelimini belirleyen, (4.17) denklemi ile verilen kısıtlama ve K noktasının konumunu belirlediğimiz (4.18) ve (4.19) denklemleri burada da geçerlidir. Şekil 4.8'de verilen çizimi göz önünde bulundurarak, a_1 ve a_2 uzuvlarının oluşturduğu üçgene kosinüs teoremini uygulanırsa (4.28) denklemi elde edilir.

$$P_{Kx}^2 + P_{Ky}^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos(180 - \beta_2) \quad (4.28)$$

Bu denklem yeniden düzenlenerek c_2 (4.29) numaralı denklemde görüldüğü gibi elde edilir ve c_2 kullanılarak β_2 (4.30) numaralı denklemde gösterildiği gibi bulunur.

$$c_2 = \frac{P_{Kx}^2 + P_{Ky}^2 - a_1^2 - a_2^2}{2a_1a_2} \quad (4.29)$$

$$\beta_2 = \pm \cos^{-1}(c_2) \quad (4.30)$$

Robot kola ait β_1 açısını bulmak için Şekil 4.8'deki θ ve δ açılarını kullanılabilir. Görüldüğü gibi θ , P_{Kx} ve P_{Ky} 'nin işaretlerinden etkilenmektedir. Dolayısı ile θ açısı (4.31) numaralı denklemde verildiği gibi atan2 fonksiyonu ile hesaplanmalıdır.

$$\theta = A \tan 2(P_{Kx}, P_{Ky}) \quad (4.31)$$

Bir diğer açı olan δ açısını bulmak için yine kosinüs teoremi bize (4.32) numaralı denklemi verir ve bu denklemin de düzenlenmesi ile (4.33) numaralı denklem ile verilen δ açısı elde edilir.

$$c_\delta \sqrt{P_{Kx}^2 + P_{Ky}^2} = a_1 + a_2 c_2 \quad (4.32)$$

$$\delta = \cos^{-1} \left(\frac{P_{Kx}^2 + P_{Ky}^2 - a_1^2 - a_2^2}{2a_1 \sqrt{P_{Kx}^2 + P_{Ky}^2}} \right) \quad (4.33)$$

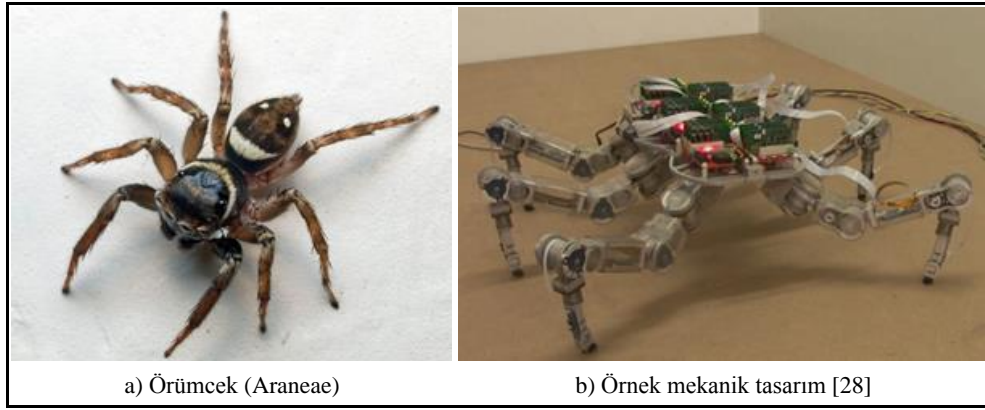
θ ve δ açıları bulunduktan sonra β_1 açısı (4.34) numaralı denklemde gösterildiği gibi hesaplanır.

$$\beta_1 = \theta \pm \delta \quad (4.34)$$

BÖLÜM 5

UYGULAMA ROBOTUNUN MEKANİK TASARIMI

Tez konusu çalışmanın uygulamasının yapılması için altı bacaklı ve her bacağı üç eklemlili olan bir robot tasarımı yapılmıştır. Bu tasarım yapılırken biyolojik benzerleri ve daha önce yapılmış olan çalışmalardan esinlenilmiştir. Şekil 5.1.a'da gerçek bir örümcek, Şekil 5.1.b'de ise mekanik bir örümcek örnek olarak verilmiştir.



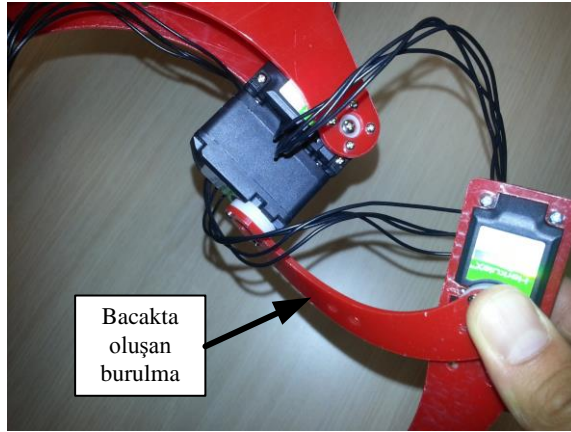
Şekil 5.1. Gerçek örümcek ve mekanik tasarım.

Bütün mekanik tasarım ve çizimler SolidWorks üç boyutlu çizim ortamı kullanılarak yapılmıştır. Yapılan çizimlerin ardından parçaların bir kısmı Plexiglas malzemenin kestirilerek, bir kısmı ise üç boyutlu yazıcı ile imal edilerek mekanik yapı gerçekleştirilmiştir. Çalışma için iki farklı tasarım yapılmıştır. İlk tasarım FPGA üzerindeki donanım ve yürüme algoritmalarının denenmesi sırasında uygulama yapmak için kullanılmıştır. Denemeler için yeterli olan bu tasarım yürüme testlerinin ilerleyen kısımlarında yetersiz kalmıştır. Tasarlanan bacak yapılarının fiziksel mukavemetinin düşük olması nedeniyle meydana gelen burulmalar yürüme eyleminin başarısız olmasına neden olmuştur. Bunun için daha dayanıklı bacaklara sahip ikinci bir tasarım yapılmıştır. İkinci tasarımda yine iki farklı ayak sensörü kullanılmıştır. İlk kullanılan sensör mekanik bir anahtar ile yapılan dijital sensördür.

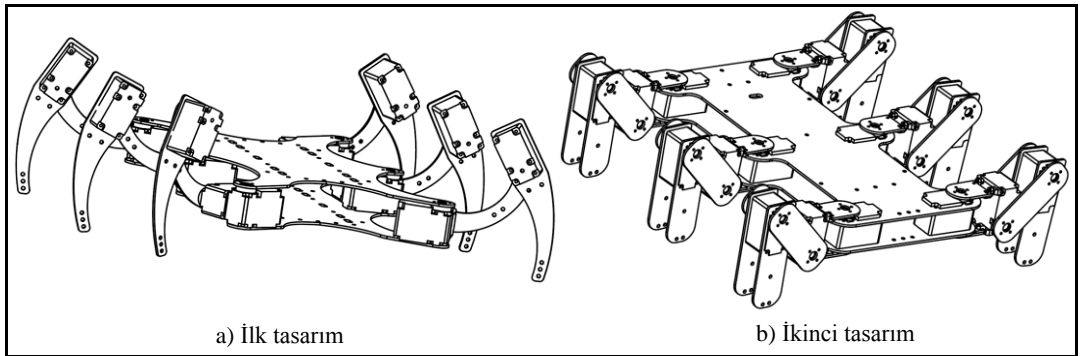
Bu sensör kullanılarak ayağın yer ile teması yeterince sağlıklı algılanamamıştır. Bunun üzerine basınç sensörü kullanılarak ikinci bir tasarım yapılmış ve elde edilen analog sensör ile ayağın yer ile teması hassas bir şekilde algılanmıştır.

5.1. ROBOTUN TASARIMI VE ÇİZİMLER

Çalışmada iki farklı tasarım yapılmıştır. İlk tasarımın bacaklarında, yürüme sırasında meydana gelen burulmadan ötürü ikinci bir tasarıma ihtiyaç duyulmuştur. Bahsedilen burulmaya bir örnek Şekil 5.2’de verilmiştir. Gösterilen bu burulma bacak pozisyonuna bağlı olarak yürüme sırasında meydana gelmekte ve hareketi bozmaktadır. Bu durumu gidermek için yapılan ikinci tasarımda bacakların mukavemeti artırılmıştır. Yapılan her iki tasarım Şekil 5.3’de verilmiştir.

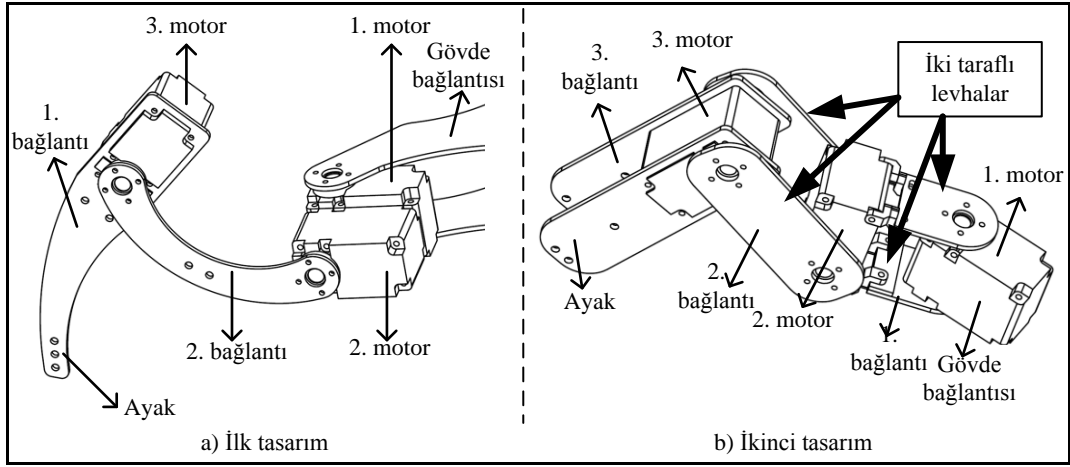


Şekil 5.2. İlk tasarım ve burulma problemi.



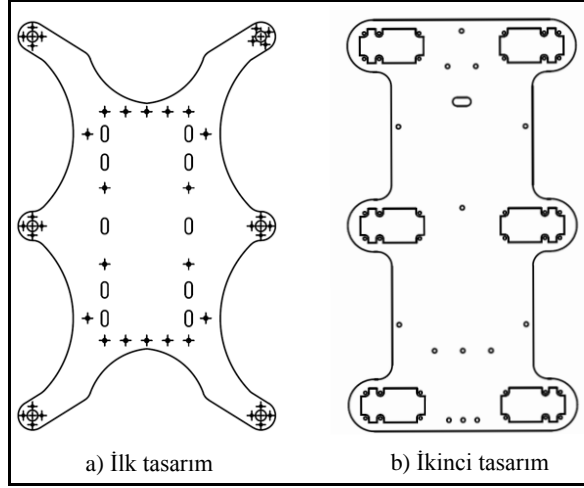
Şekil 5.3. İlk ve ikinci tasarımın karşılaştırması.

Yapılan ilk ve ikinci bacak tasarımları Şekil 5.4’de karşılaştırmalı olarak verilmiştir. Her bir bacak üç eklemlilik olarak çizilmiştir. Her ekleme bir servo motor yerleştirilmiştir. İlk tasarımda ikinci bağlantı bacağın yukarı doğru hareketinde gövdeye çarpmadan daha fazla hareket edebilsin diye yay şeklinde çizilmiştir. Yeni tasarımda ikinci bağlantı daha kalın ve biraz daha kısa çizilmiştir. Bu şekilde dayanımı artırılmıştır. Birinci motor yatay düzlemde dönme sağlamakta, ikinci ve üçüncü motorlar bacağın dikey düzlemde hareketini sağlamaktadır. Yapılan ikinci tasarımda iki taraflı levhalar (Plexiglas) kullanılarak daha fazla dayanım sağlanması amaçlanmıştır ve burulmanın önüne geçilmiştir.



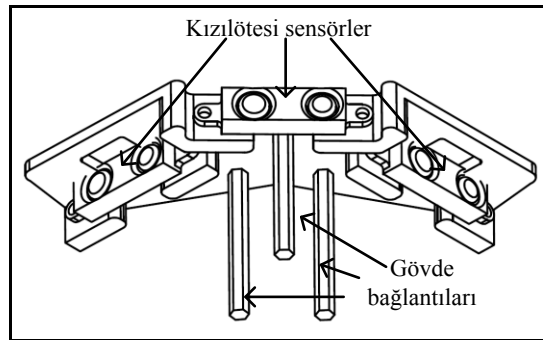
Şekil 5.4. İlk tasarımın bacak kesiti.

Tasarlanan robotların gövde çizimleri Şekil 5.5’de verilmiştir. Gövde tasarımı yapılırken, üzerine montajı yapılan motorlar hareket sırasında birbirine engel olmayacak şekilde yerleştirebilmek için gerekli mesafeler bırakılmıştır. Gövde üzerindeki eklemin daha geniş açıda hareket edebilmesi için eklemler biraz dışarı çıkıntılı yapılmıştır. Montaj ve vidalama için gerekli delik ve boşluklar çizilmiştir. Malzemenin daha hafif olması için bu delik ve boşluklar gerekenden fazla çizilmiştir. Bu parçadan iki tanesi motorlar ile birleşerek robotun gövdesini oluşturmaktadır. İkinci tasarımda dayanımı artırmak için boynuz gibi ileri uzayan bacak bağlantıları kısaltılmış ve kalınlaştırılmıştır. İlk tasarımda gövde bağlantısı bir numaralı motorun rotoru ile sağlanmaktadır. İkinci tasarımda ise motor tamamen gövdeye bağlanarak daha sağlam bir yapı elde edilmiştir.

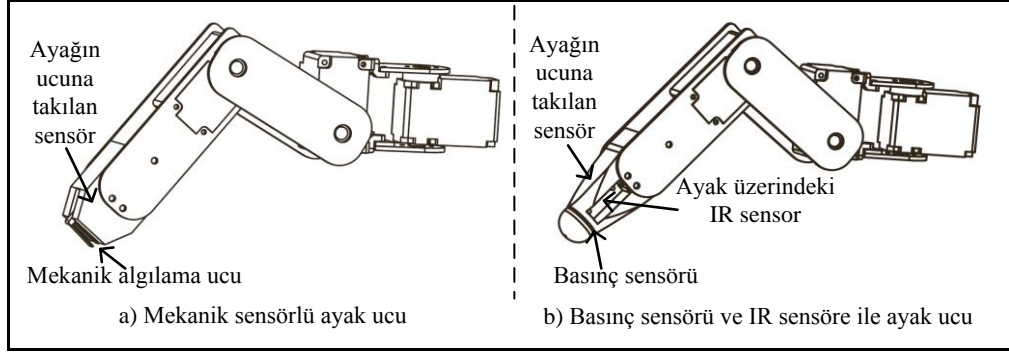


Şekil 5.5. Altı bacaklı robotun gövde yapısı.

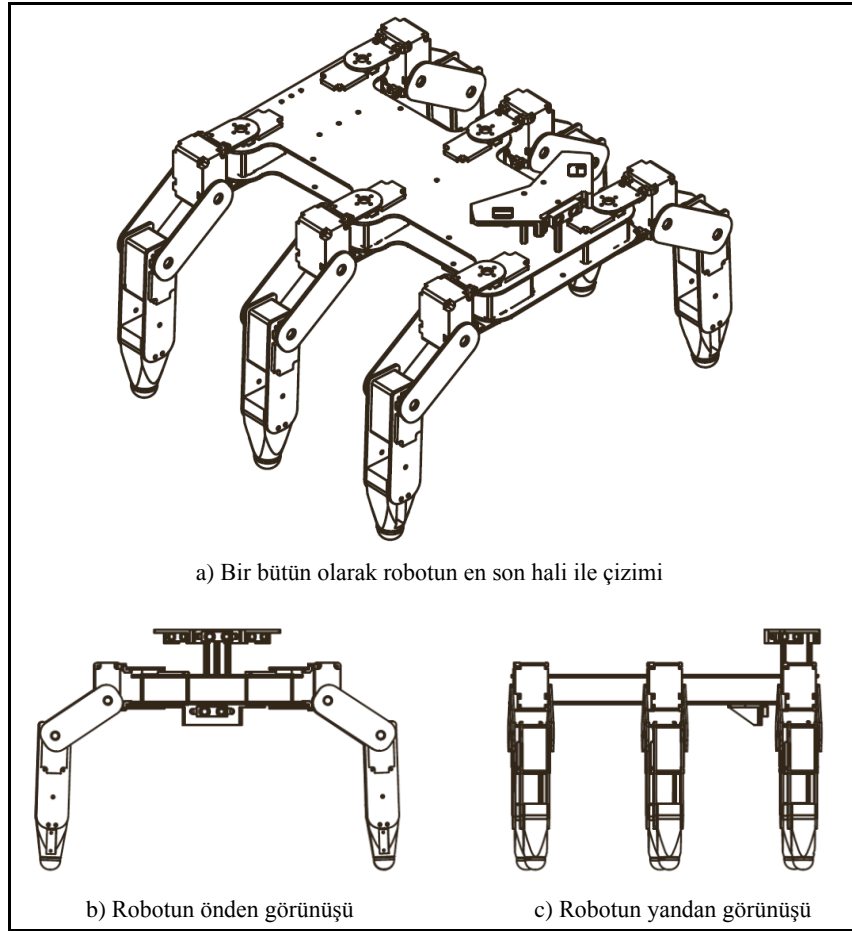
Çizimleri yapılan parçalar lazer CNC tezgâhta, Plexiglas malzeme işlenerek elde edilmiştir. Ayrıca ayakların yere bastığını algılamak için ayaklara ve etraftaki nesnelere fark edebilmek için robotun ön tarafına sensörler yerleştirilmiştir. Ayrıntısı BÖLÜM 6’da verilen bu sensörler, robotun ön kısmına, kafaya benzer bir yapı üzerine yerleştirilmiştir. Sensörlerden oluşan bu kafa Şekil 5.6’da verilmiştir. Ayrıca tasarlanan ayak yapıları Şekil 5.7’de gösterilmiştir. Şekil 5.7.a’da verilen çizim mekanik anahtar ile yapılan uygulamadır. Şekil 5.7.b’de verilen çizim ise basınç ve kızılötesi sensörleri üzerinde bulunduran tasarımdır. Ayak ucu ve kafa yapıları üç boyutlu yazıcı ile imal edilmiştir. Basınç sensörü kullanan ayak ucu ile elde edilen en son tasarım Şekil 5.8’de farklı açılardan gösterilmiştir.



Şekil 5.6. Robotun ön kısmına yerleştirilen sensörler.



Şekil 5.7. Tasarlanan iki farklı ayak ucu.



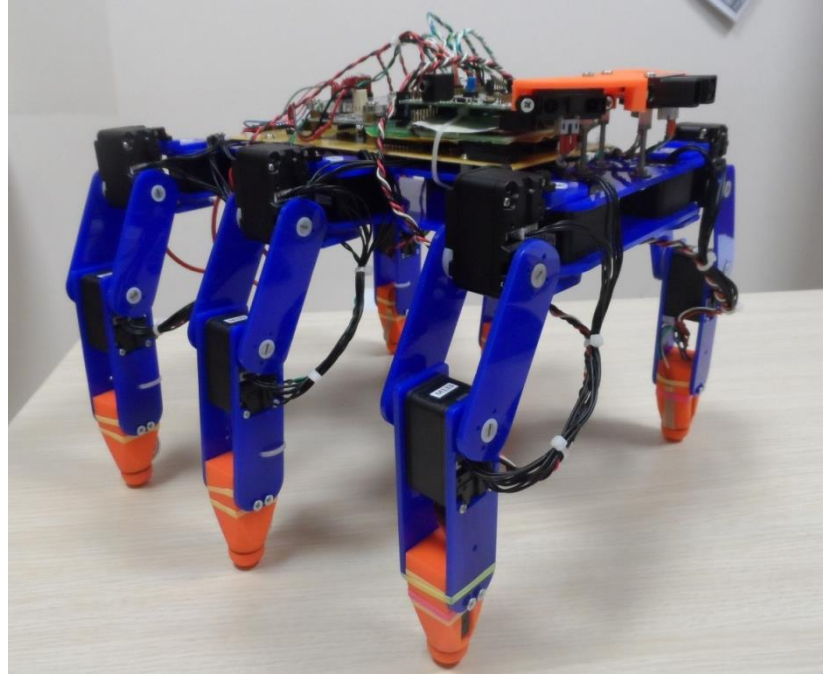
Şekil 5.8. Farklı açılardan robot tasarımının son hali.

Şekil 5.8’de verilen tasarıma ait ölçüler Çizelge 5.1’de verilmiştir. Bu ölçüler kinematik denklemlerin oluşturulması ve FPGA üzerinde geliştirilen donanım çalışmalarda kullanılmıştır. Tasarlanan robotun bir bütün olarak, mekanik ve

elektronik montajının tamamlanmasından sonra alınan görüntüsü Şekil 5.9'da verilmiştir.

Çizelge 5.1. Robotun ölçüleri.

Parça	Uzunluk	
	Sensörsüz	Sensörlü
3. bağlantı (a_3)	88mm	128mm
2. bağlantı (a_2)	60mm	
1. bağlantı (a_1)	49mm	
Gövde genişliği ($2x_{a_M}$)	125mm	
Gövde uzunluğu	291mm	
Robot ağırlığı	1790gr	
Batarya ağırlığı	303gr	

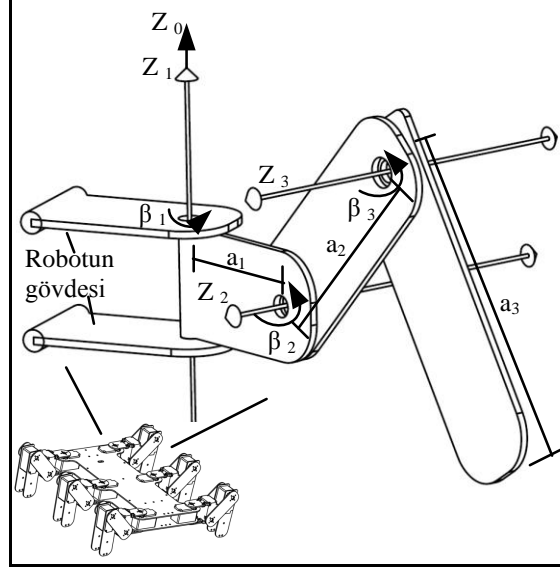


Şekil 5.9. Robotun mekanik ve elektronik montajının tamamlanmış hali.

5.2. KİNEMATİK DENKLEMLERİN OLUŞTURULMASI

Tasarlanan robota ait bacak yapısının sadeleştirilmiş hali Şekil 5.10'da verilmiştir. Bacak üç serbestlik derecesine sahiptir. Birinci eklem robot gövdesinin konumlandığı yatay düzlemde dairesel olarak hareket etmektedir. İkinci ve üçüncü eklemler robot gövdesine dik bir düzlemde, yine dairesel olarak hareket etmektedir. BÖLÜM 4'de bahsedildiği gibi, robotun kinematik denklemleri, oluşturulduktan

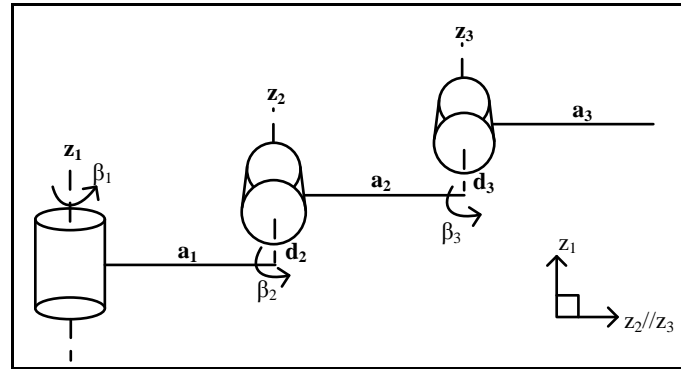
sonra ters kinematik denklemleri elde edilmiş ve bu denklemler FPGA üzerinde kullanılarak gerçek zamanlı çalışan bir donanım gerçekleştirilmiştir. Bu amaç doğrultusunda öncelikle robota ait tek bir bacağın ileri kinematik denklemleri çıkartılmış, elde edilen bu denklemlerden, ters kinematik çözüm kullanılarak, eklem açılarının ayak pozisyonuna bağlı fonksiyonları elde edilmiştir.



Şekil 5.10. Üç eklemli bacak modeli.

5.2.1. Üç Eklemli Bacağın İleri Kinematiği

Üç eklemli bacağın temel yapısı Şekil 5.11’de verilmiştir. Gövdeye bağlı olan eklem z ekseni gövdeye dik olacak şekilde konumlandırılmış, diğer iki eklem z eksenleri ise ilkinde 90° açı yapacak şekilde yerleştirilmiştir.



Şekil 5.11. Temel bacak yapısı ve eklemler.

Oluşturulan bacak yapısı için, bölüm 4.1.1’de anlatılan, Denavit–Hartenberg yöntemi ile belirlenen parametreler Çizelge 5.2’de verilmiştir. Ölçüler yapılan çizim neticesinde belirlenen kesin ölçülerdir. Üç eklemlili bu yapı için, üç Homojen Dönüşüm Matrisi oluşturulacaktır.

Çizelge 5.2. Üç eklemlili bacak için Denavit–Hartenberg parametreleri.

i	α_i	a_i	d_i	β_i
1	0	49mm	0	β_1
2	90	60mm	0	β_2
3	0	128mm	0	β_3

Çizelge 5.2’de verilen parametreler kullanılarak, üç eklem için oluşturulan, dönüşüm matrisleri (5.1), (5.2) ve (5.3) denklemlerinde verilmiştir.

$$T_1^0 = \begin{bmatrix} \cos(\beta_1) & 0 & \sin(\beta_1) & a_1 \cdot \cos(\beta_1) \\ \sin(\beta_1) & 0 & -\cos(\beta_1) & a_1 \cdot \sin(\beta_1) \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$T_2^1 = \begin{bmatrix} \cos(\beta_2) & -\sin(\beta_2) & 0 & a_2 \cdot \cos(\beta_2) \\ \sin(\beta_2) & \cos(\beta_2) & 0 & a_2 \cdot \sin(\beta_2) \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$$T_3^2 = \begin{bmatrix} \cos(\beta_3) & -\sin(\beta_3) & 0 & a_3 \cdot \cos(\beta_3) \\ \sin(\beta_3) & \cos(\beta_3) & 0 & a_3 \cdot \sin(\beta_3) \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

Bu matrislerin çarpımı ile yapının ileri kinematik denklemi ortaya çıkacaktır. Oluşturulan bu matrisler (5.4) denkleminde verildiği gibi çarpılarak, denklem (5.5) ‘de verilen dönüşüm matrisi elde edilir. Bu matris uç işlevcinin konum ve yönelimini eklem değişkenlerine göre tanımlamaktadır. Burada denklemin kapladığı yeri küçültmek için kısaltmalar kullanılmıştır; sinüs ve cosinüs fonksiyonları baş harfleri

ile gösterilmiştir ve alt indiste ulunan birden fazla rakam ilgili açı değerlerinin toplanarak fonksiyona parametre olacağını göstermektedir ($\cos(x+y)=c_{xy}$ gibi).

$$T_3^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \quad (5.4)$$

$$T_3^0 = \begin{bmatrix} c_{23} \cdot c_1 & -s_{23} \cdot c_1 & s_1 & c_1 \cdot (a_1 + a_3 \cdot c_{23} + a_2 \cdot c_2) \\ c_{23} \cdot s_1 & -s_{23} \cdot s_1 & -c_1 & s_1 \cdot (a_1 + a_3 \cdot c_{23} + a_2 \cdot c_2) \\ s_{23} & c_{23} & 0 & a_3 \cdot s_{23} + a_2 \cdot s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

5.2.2. Üç Eklemlili Bacağın Ters Kinematığı

Denklem (5.5)'de verilen matrisin son sütunu bacağa ait ayağın (bacağın en uç noktası) konumunu vermektedir. Buradan yola çıkarak ayağın x, y ve z koordinatları için (5.6), (5.7) ve (5.8) denklemleri yazılabilir. Bu noktadan sonra hedef bu üç denklemin çözülerek β açılarının bulunmasıdır. Bulunan açılar bacağa ait servo motorlara gönderildiğinde bacak ilgili pozisyonu alacaktır.

$$P_x = c_1 \cdot (a_1 + a_3 \cdot c_{23} + a_2 \cdot c_2) \quad (5.6)$$

$$P_y = s_1 \cdot (a_1 + a_3 \cdot c_{23} + a_2 \cdot c_2) \quad (5.7)$$

$$P_z = a_3 \cdot s_{23} + a_2 \cdot s_2 \quad (5.8)$$

Bu üç denklemi kullanarak β_3 açısını bulmak için, kosinüs teoreminden faydalanarak, denklem (5.9)'da verilen yeni denklem oluşturulmuştur. Ayağın konumu (P_x , P_y ve P_z) zaten bilinmektedir. Denklem çözülerek $\cos(\beta_3)$ ve $\sin(\beta_3)$ bulunmuş, ardından arctan fonksiyonu ile β_3 açısı elde edilmiştir. Denklem (5.11)'in işaretine göre iki farklı çözüm ortaya çıkmaktadır.

$$P_x^2 + P_y^2 + a_1^2 - 2 \cdot a_1 \cdot \sqrt{P_x^2 + P_y^2} + P_z^2 = a_2^2 + a_3^2 + 2 \cdot a_2 \cdot a_3 \cdot c_3 \quad (5.9)$$

$$c_3 = \frac{P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2} + P_z^2 - (a_2^2 + a_3^2)}{2.a_2.a_3} \quad (5.10)$$

$$s_3 = \pm\sqrt{1 - c_3^2} \quad (5.11)$$

$$\beta_3 = \arctan\left(\frac{s_3}{c_3}\right) \quad (5.12)$$

β_2 açısı için denklem (5.13) yazılmış ve buradan gidilen çözüm ile $\cos(\beta_2)$ ve $\sin(\beta_2)$ için (5.17) ve (5.18)'de verilen sonuçlar bulunmuştur. Bu sonuçların işaretine göre dört farklı çözüm meydana gelmektedir.

$$P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2} = (a_2.c_2 + a_3.c_{23})^2 \quad (5.13)$$

$$a_2.c_2 + a_3.c_{23} = \pm\sqrt{P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2}} \quad (5.14)$$

$$c_2(a_2 + a_3.c_3) - a_3.s_2.s_3 = \pm\sqrt{P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2}} \quad (5.15)$$

$$c_2(a_2 + a_3.c_3) - a_3.s_3\left(\frac{P_z - a_3.c_2.c_3}{a_2 + a_3.c_3}\right) = \pm\sqrt{P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2}} \quad (5.16)$$

$$c_2 = \frac{\pm\sqrt{P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2}}.(a_2 + a_3.c_3) + P_z.a_3.s_3}{a_2^2 + a_3^2 + 2.a_2.a_3.c_3} \quad (5.17)$$

$$s_2 = \frac{P_z(a_2 + a_3.c_3) \pm \sqrt{P_x^2 + P_y^2 + a_1^2 - 2.a_1.\sqrt{P_x^2 + P_y^2}}.a_3.s_3}{a_2^2 + a_3^2 + 2.a_2.a_3.c_3} \quad (5.18)$$

$$\beta_2 = \arctan\left(\frac{s_2}{c_2}\right) \quad (5.19)$$

β_1 açısı için benzer bir şekilde (5.20) ve (5.21) denklemlerinden yola çıkarak dört farklı açı bulunur.

$$P_x = \pm c_1 \cdot \sqrt{P_x^2 + P_y^2} \quad (5.20)$$

$$P_y = \pm s_1 \cdot \sqrt{P_x^2 + P_y^2} \quad (5.21)$$

$$\beta_1 = \arctan\left(\frac{s_1}{c_1}\right) \quad (5.22)$$

Bu denklemlerden elde edilen bütün açılar, ilgili pozisyon için doğru sonuçlar vermektedir. Ancak bazıları kinematik yapı, hareket uzayındaki engeller veya fiziksel yapıyla ilgili sınırlamalar nedeniyle uygulanamayabilir. Tasarımı verilen bacağın merkezindeki Kartezyen koordinatın üzerinde Y eksenini referans alınırsa; bacağı yalnızca sağ tarafta (sağ bacaklar için) veya yalnızca sol tarafta (sol bacaklar için) konumlayan çözümler kullanılabilir. Bu ayırım ise yalnızca β_1 açısına bakılarak yapılabilmektedir.

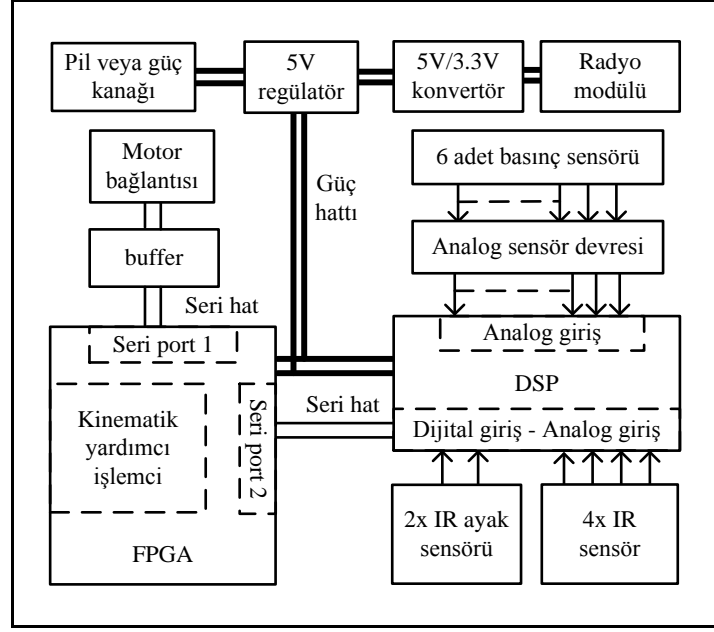
BÖLÜM 6

ELEKTRONİK DONANIM VE KUMANDA ARAYÜZÜ TASARIMI

Çalışmanın bu kısmında FPGA ile yapılan uygulamanın gerçekleştirilmesi ve robotun kontrol edilebilmesi için gerekli özellikleri taşıyan elektronik donanım tasarlanmış, ayrıca bu donanımın uzaktan kontrollü ve sensör verilerinin takibi için uzaktan kumanda arayüzü geliştirilmiştir. Tasarlanan elektronik donanım bileşenler arasındaki elektriksel bağlantıyı temin etmekte, güç gereksinimlerini karşılamakta, doğabilecek yeni gereksinimler için ek bağlantılar içermektedir.

Elektronik donanımı oluşturan temel bileşenler anakart, servo motorlar, FPGA kartı, DSP kartı, radyo modülü, sensörler ve bataryadır. Bu bölümde tasarım ve bileşenler hakkında açıklamalar bulunmaktadır. Sistemin elektriksel bağlantı yapısı Şekil 6.1’de bir bütün olarak verilmiştir.

Tasarlanan sistemde anakart, FPGA, DSP ve motorlar için pil veya güç kaynağından gelen gerilimi uygun seviyelere düşürecek devreleri, FPGA ve diğer elektronik donanım arasında elektriksel bağlantıları sağlayacak yolları ve bağlantı soketlerini üzerinde bulundurmaktadır. Sistemin elektriksel bağlantı yapısı Şekil 6.1’de bir bütün olarak verilmiştir. Kullanılan servo motorlar kendi kontrolcülerine ve iletişim için kendi seri portlarına sahiptir. Bu motorları kumanda etmek için açılış bilgisinin uygun formatta gönderilmesi yeterlidir. Sistemde DSP üzerinde koşan algoritma kablosuz haberleşme ve yürüme işlevinin gerçekleşmesi için ayak pozisyonlarının üretilmesinden sorumludur. Üretilen ayak pozisyonları FPGA üzerinde geliştirilen kinematik yardımcı işlemci ile çözülmekte ve yine FPGA üzerindeki donanımın bir parçası olarak geliştirilen seri port tarafından motorlara iletilmektedir.



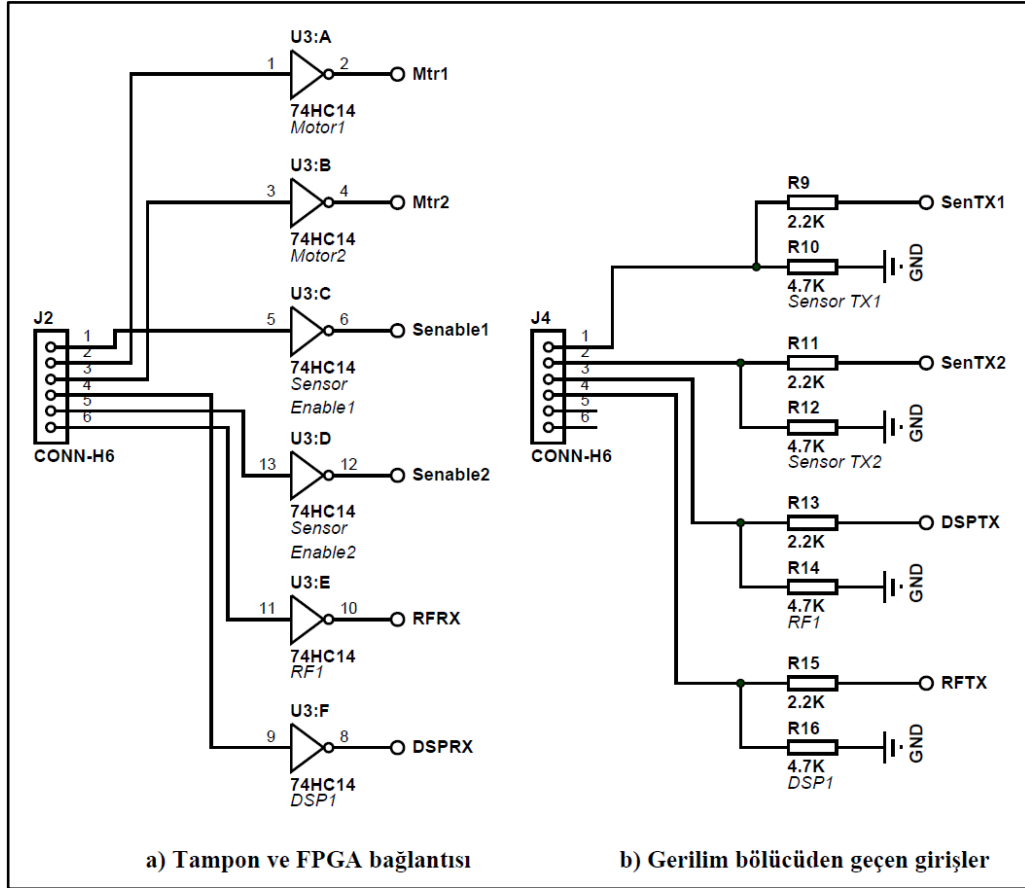
Şekil 6.1. Sistemin genel elektriksel bağlantı yapısı.

6.1. ANAKART

Uygulama sırasında gerekli elektronik donanımı bir arada tutacak ve güç ihtiyacının karşılanmasında güç kaynağı ile cihazlar arasında köprü görevi görececek bir anakartın tasarımı bu kısımda ele alınmıştır. Anakart üzerinde batarya bağlantısı ve 5V regülatör birimi, kablosuz haberleşme birimi, DSP ve FPGA bağlantıları, motor ve sensör bağlantıları bulunmaktadır. Elektriksel bağlantıların genel yapısı Şekil 6.1’de verilmiştir. Kart, 7.4-12V batarya ile veya harici bir güç kaynağı ile kullanılabilir.

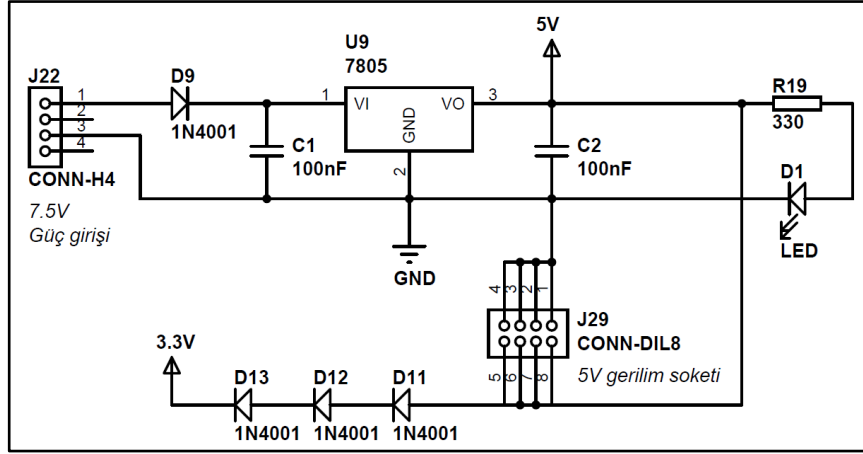
Uygulama sırasında FPGA kartının zarar görememesi ve sinyal seviyelerinin düzenlenmesi amacıyla, FPGA giriş-çıkışları ile harici bağlantılar arasında, tampon konmuştur. Bunun için 74HC14 entegresi kullanılmıştır ve ayrıntısı Şekil 6.2.a’da verilmiştir. Böylece FPGA kartı üzerindeki 0-3.3V gerilim seviyeleri diğer devre elemanlarına 0-5V olarak iletilmiştir. Ayrıca FPGA çıkışına bağlanan eleman sayısı bire indirildiği için çekilecek akımda sınırlandırılmış olmaktadır. Bu tampon aracılığı ile “motor bağlantısı 1” ve motor bağlantısı 2” için gerekli seri hat (Mtr1, Mtr2), gerekli durumlarda kullanılmak üzere iki adet sensör etkinleştirme çıkışı (Senable1, Senable2), radyo modülü ve DSP ile haberleşmek üzere seri hatlar (RFRX, DSPRX) için bağlantılar oluşturulmuştur. Bu sinyallerin hepsi FPGA’den çevre birimlere

gönderilen sinyallerdir. Yine gerekli durumlarda kullanılmak üzere çevre birimlerden FPGA'ye gelen 5V seviyesindeki sinyaller ise, Şekil 6.2'de görüldüğü gibi, gerilim bölücüler vasıtası ile 3.3V'a düşürülmektedir.



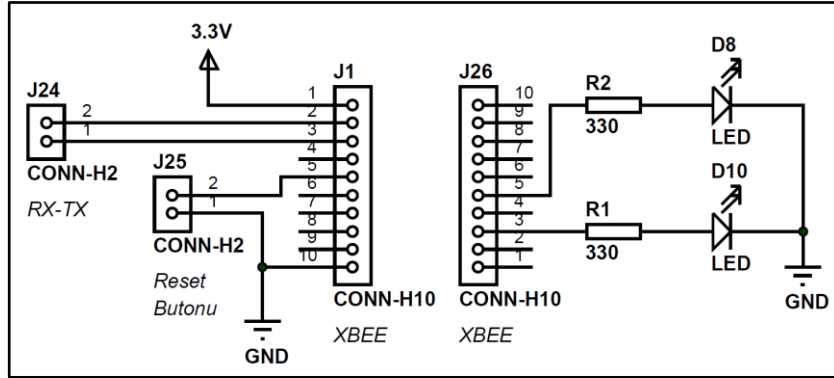
Şekil 6.2. Lojik tampon ve gerilim bölücüler.

Anakart üzerinde ihtiyaç duyulan farklı gerilim seviyelerinin elde edilmesi için Şekil 6.3'de verilen devre kullanılmıştır. Bu devrede giriş yaklaşık 7.5-12V olabilir. Giriş gerilimi güç kaynağı veya bataryadan sağlanabilir. Giriş gerilimi öncelikle 7805 entegresi ile 5V seviyesine düşürülmektedir. Bu 5V gerilimden ise 3 diyot yardımı ile $5-3 \times 0.7 = 2.9 \approx 3V$ değeri elde edilmektedir. Bu değer pratikte 3-3.3V aralığında ölçülmektedir. Elde edilen bu gerilim radyo modülünün çalışması için gereklidir. Ayrıca güç katı üzerinde bir güç gösterge LED'i ve harici kullanım için dört adet 5V gerilim soketi bulunmaktadır. Bu soketlerden birisi DSP kartı beslemek için kullanılmıştır. DSP kartı 5V gerilim ile beslenebilmekte ancak çalışma gerilimi 3.3V'dur.



Şekil 6.3. Anakart güç katı.

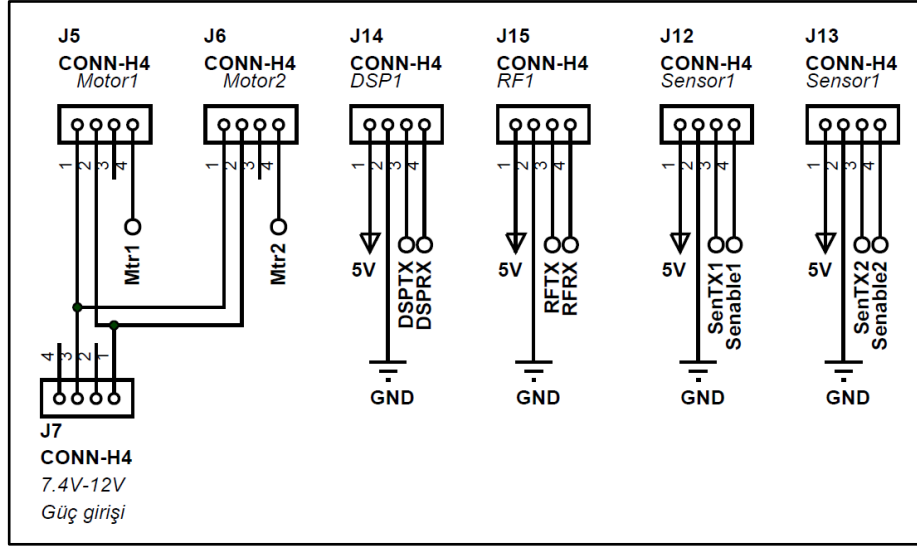
Kullanılan radyo modülünün sisteme entegrasyonu için Şekil 6.4’de verilen devre kullanılmıştır. J1 ve J29 soketleri radyo modülünün karta montajı için kullanılmaktadır. Radyo modülü DSP kartı ile iletişim kurmaktadır ve bağlantı kablo ile yapılmıştır. Kablonun bağlanacağı soket J24 soketidir. Ayrıca şekilde görülen iki adet LED, radyo modülünün çalışması sırasında kullandığı uyarı sinyallerini yansıtmak için kullanılmıştır.



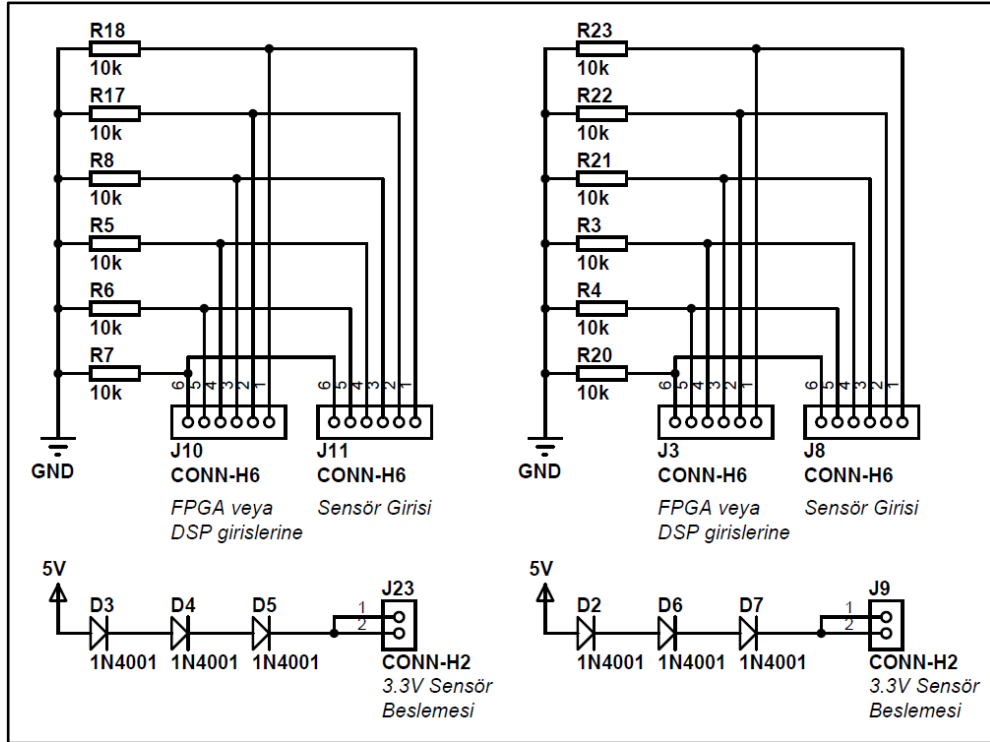
Şekil 6.4. Radyo modülü bağlantıları.

Şekil 6.5’de motor ve diğer çevre birimlerinin bağlantıları için kullanılan soketler ve pin bağlantıları verilmiştir. Şekil 6.6’da ise ayak sensörlerinin bağlantısı için kullanılan dijital sensör devresi verilmiştir. Devrede görüldüğü girişler normal durumda aşağı çekme (pull-down) dirençleri ile aşağı çekilmiştir. Kullanılan sensörlerin kontağı kapalı devre olduğunda dirençler üzerinde 3.3V gerilim

oluşmaktadır ve bu gerilim DSP kartı tarafından mantıksal 1 olarak algılanmaktadır. Devrenin beslemesi 5V ile yapılmış ve diyotlar yardımı ile gerilim 3-3.3V aralığına düşürülmüştür.

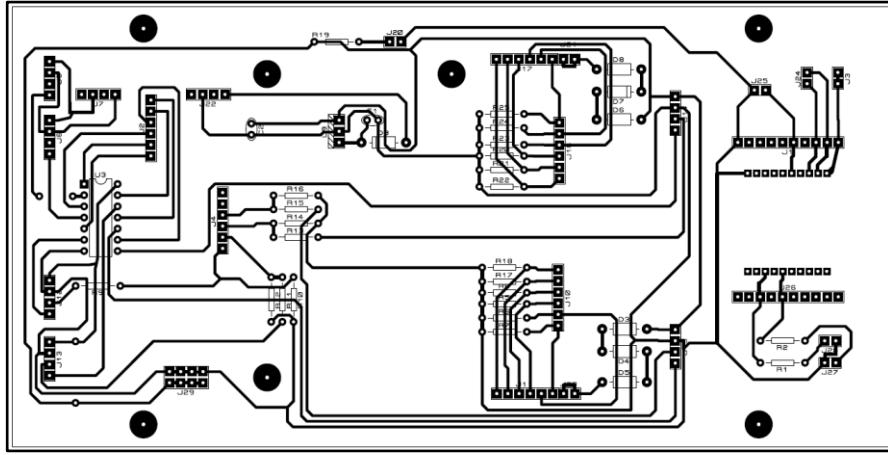


Şekil 6.5. Motor ve diğer çevre birimlerin bağlantı soketleri.

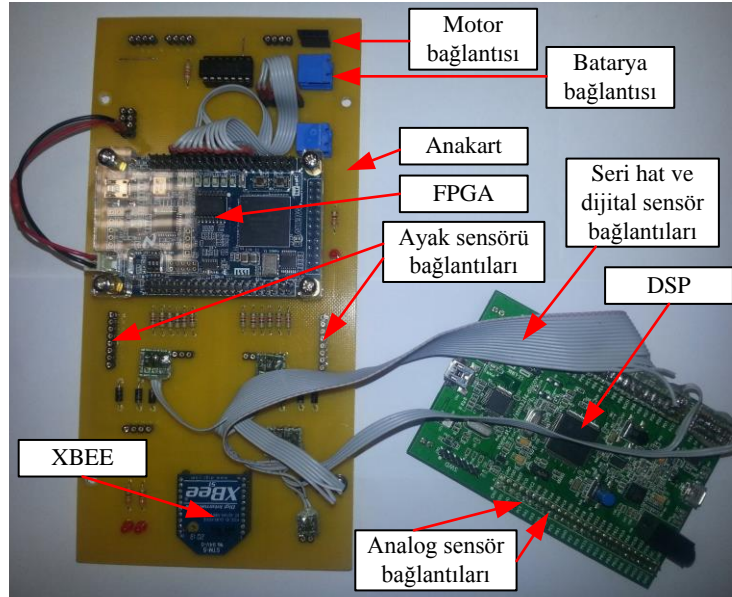


Şekil 6.6. Dijital sensör devresi.

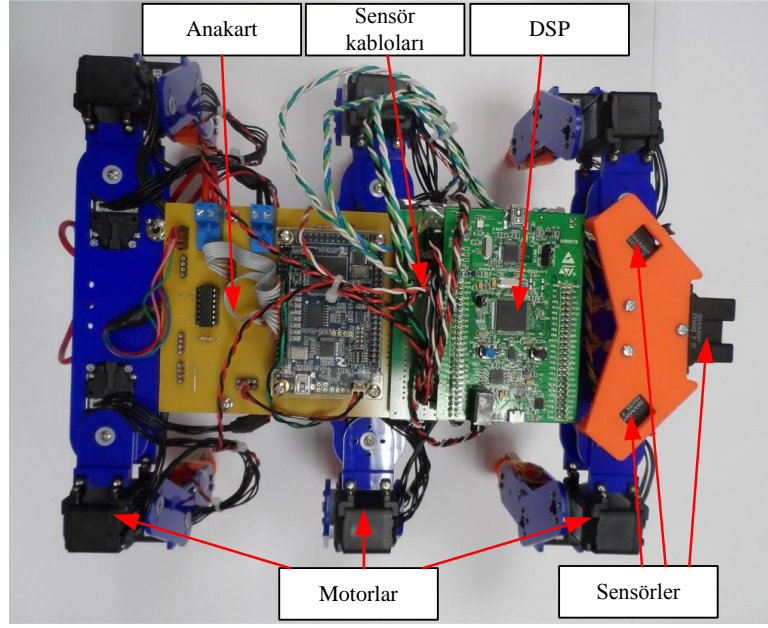
Buraya kadar verilen bütün bölümleri üzerinde bulunduran anakartın baskı devre çizimi Şekil 6.7’de verilmiştir. Şekil 6.8’de ise gerçekleştirilen sistem ve üzerindeki bileşenler verilmiştir. Görüldüğü gibi DSP ve ayak sensörlerinin anakart ile olan bağlantısı kablolar ile sağlanmıştır. Analog sensörler doğrudan DSP soketlerine bağlanmıştır. Şekil 6.9 ve Şekil 6.10’da sistem mekanik parçalar ile birlikte bir bütün olarak verilmiştir ve üzerinde elektronik bileşenler gösterilmiştir.



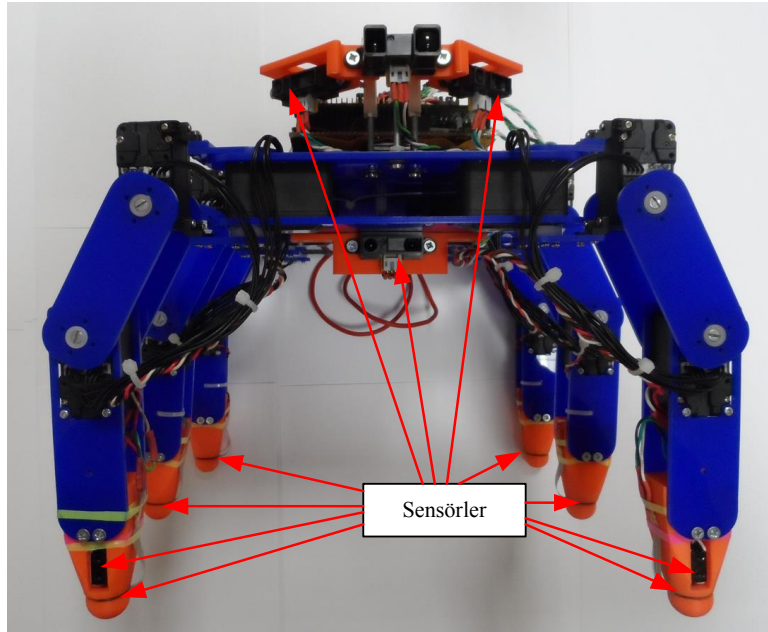
Şekil 6.7. Anakart PCB çizimi (üstten görünüm).



Şekil 6.8. Anakart ve bileşenleri.



Şekil 6.9. Sistem üzerinde anakart ve bileşenleri.



Şekil 6.10. Sistem üzerinde DSP kart ve sensörler.

6.2. SERVO MOTORLAR

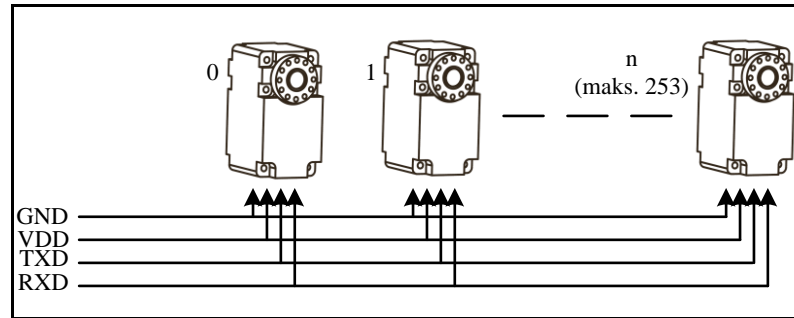
Tasarlanan bu anakart ile birlikte robotun hareketini sağlamak için servo motorlar kullanılmıştır. Kullanılan servo motorlar, HerkuleX DRS-0101 dijital servolardır

[81]. Bu motorların en önemli özelliği seri arabirimden kontrol edilebiliyor olmasıdır. Tek bir seri hatta 254 adede kadar motor bağlanıp, bu hat üzerinden kontrol edilebilmektedir. Motorun çalışma gerilimi yaklaşık 7.5-12V aralığında, haberleşme hattı sinyal seviyesi TTL standartlarında, açısal çözünürlüğü 0,325°. Rotorun taradığı alan yaklaşık 320°. Motor üzerinde hem hatalar (gerilim hatası, tork veya konum hatası gibi) meydana geldiğinde uyarı vermek için hem de kullanıcı tarafından serbest bir şekilde kullanılması için üç adet LED vardır. Robotik uygulamalar için geliştirilmiş bir motordur.

Motorlar birbirine seri olarak bağlanabilmektedir. Motor üzerindeki socketin pin açıklamaları Çizelge 6.1’de verilmiştir. Motorun kontrolü seri hat üzerinden verilen komutlar ile gerçekleştirilmektedir. Motora gönderilen açı bilgisine göre, motor üzerindeki dahili PID kontrolcü rotor açısını ayarlamaktadır. Genel bağlantı şeması Şekil 6.11’de verilmiştir.

Çizelge 6.1. Motor socketi pin açıklamaları.

Pin numarası	İsim	Açıklama
1	GND	Elektriksel toprak bağlantısı
2	VDD	Besleme gerilimi
3	TXD	Seri port veri iletim hattı
4	RXD	Seri port veri alma hattı



Şekil 6.11. Motorların seri hatta bağlanması.

Motor 8 veri biti, 1 durma biti içeren, eşlik biti içermeyen veri paketleri ile iletişim kurmaktadır. İletişim hızı 57,600bps ile 0,677Mbps hızları arasında ayarlanabilmektedir. Motor uzunluğu en fazla 223-byte olan veri paketleri

kullanılmaktadır. Paketlerin standart yapısı Çizelge 6.2’de gösterilmiştir. İletişim paketleri ile ilgili ayrıntı BÖLÜM 7’de verilmiştir.

Çizelge 6.2. Motor iletişim paketinin genel yapısı.

	Başlık	Paket büyüklüğü	pID	CMD	Check Sum1	Check Sum2	Data[n]
Değer	0xFF 0xFF	7-223	0- 0xFE	1-9	-	-	-
Uzunluk (byte)	2	1	1	1	1	1	Maks. 216

6.3. DİĞER DONANIM VE SENSÖRLER

Çalışmada FPGA ile geliştirilen donanımın uygulaması için Altera tarafından üretilen “DE0-Nano Board” isimli FPGA kartı kullanılmıştır. Bu kart gömülü sistemler, bilgisayar mimarisi çalışmaları ve robot uygulamaları için uygun bir karttır. Üzerinde Cyclone IV EP4CE22F17C6N yongası kullanılmıştır [82].

Geliştirilen yürüme algoritmasının işletilmesi için STM32 F4 Discovery kartı kullanılmıştır. Bu kart üzerinde, 32-bit ARM Cortex-M4F çekirdeğine sahip, STM32F407VGT6 yongası bulunmaktadır [83]. Matlab/Simulink üzerinde geliştirilen yürüme algoritmaları bu kart üzerine gömülerek uygulanması yapılmıştır. Yürüme algoritmasının ürettiği çıkışlar bu kart vasıtası ile seri port üzerinden FPGA’ye gönderilmektedir. Bu kart aynı zamanda yürüme algoritmasının işletilmesi sırasında ihtiyaç duyulan ayak sensörleri, isteğe bağlı olarak kullanılacak mesafe sensörleri ve uzaktan kumanda arayüzü ile haberleşmek için radyo modülü ile de bağlantılıdır.

Radyo modülü uzaktan kumanda bilgisayarı ile iletişim sağlamak; komut almak ve bilgisayara sensör verisi göndermek için kullanılmıştır. Kullanılan modül Diji firmasının ürettiği Xbee Seri 1 radyo modülüdür [84]. Açık alanda 90 metre çekim mesafesine, dahili bir antene, 112500 Kbps iletişim hızına sahiptir. Robot üzerindeki

modül doğrudan sisteme entegre edilmiştir, bilgisayar tarafında ise “explorer” adı verilen bir aparat ile USB port üzerinden kullanılmaktadır.

Robot üzerinde iki tür sensör kullanılmıştır. Bunlardan ilki Interlink Electronics tarafından üretilen, FSR402 koduna sahip kuvvet sensörüdür [85]. 0,2N ile 20N aralığındaki kuvvetleri algılayabilmektedir. Sensör üzerine uygulanan kuvvet karşısında direnç değişimi göstermektedir. Üzerinde kuvvet yokken direnci 10Mohm'dan büyüktür. Maksimum kuvvet uygulandığında ise direnci kohm mertebesindedir. Bu sensörün kullanımı için bir gerilim bölücü devre kullanılmıştır.

Kullanılan ikinci tip sensör kızıl ötesi sensördür. Sharp markasının GP2Y0A41SK0F model koduna sahip ürünü kullanılmıştır. Sensör 4 ile 30cm arasındaki nesnelere algılayabilmektedir. Analog çıkışa sahiptir. Bu sensörden robot üzerinde üç adet kullanılmıştır. Her üçü de yakın mesafe için kullanılan bu sensörlerden bir tanesi robot gövdesinin altına, ileri bakacak şekilde yerleştirilmiştir ve diğer sensörlere göre daha alçak engellerin algılanmak amacıyla kullanılmaktadır. Diğer iki sensör ise kafaya biri sağda biri solda olmak üzere 45° açı ile yerleştirilmiştir. Bu sensörlerin amacı ise robotun yakın mesafe görüş açısını genişletmektir. Yine kafaya uzun mesafe algılama için bir adet Sharp GP2Y0A02YK0F yerleştirilmiştir. Bu sensörün algılama mesafesi ise yaklaşık 1.5m'dir. Ayak uçlarına ise iki adet Sharp GP2Y0D810Z0F sensör yerleştirilmiştir. Bu sensörler 10cm algılama mesafesine ve dijital çıkışa sahiptir.

6.4. UZAKTAN KUMANDA ARAYÜZÜ

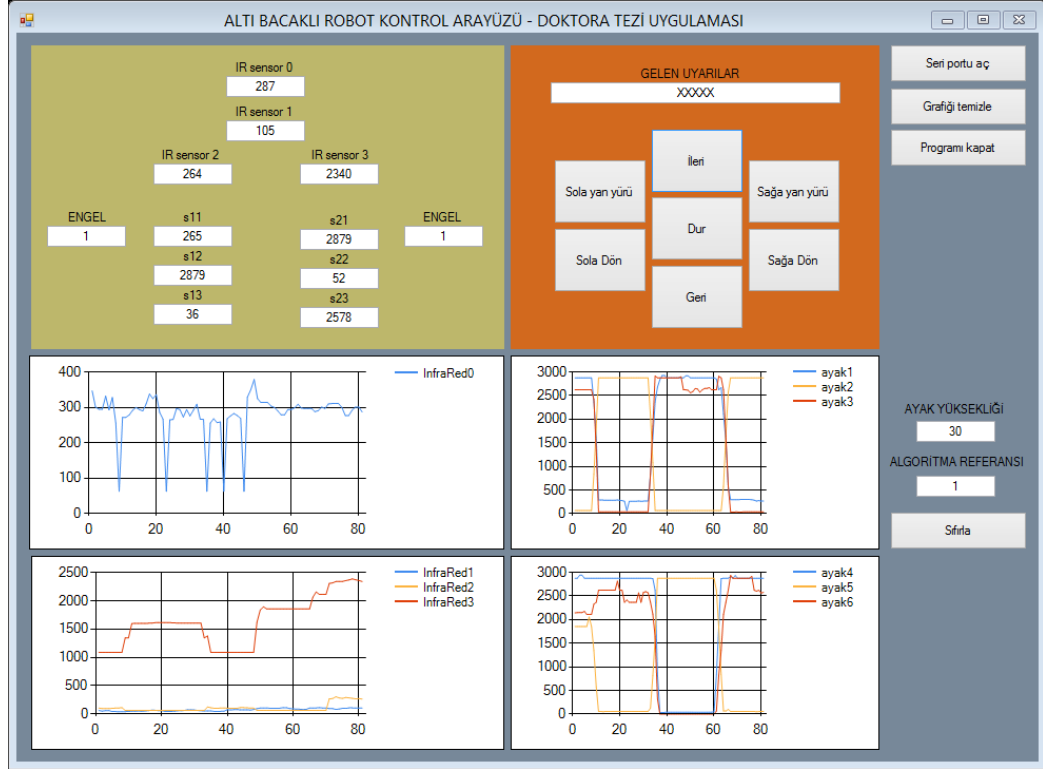
Robotu uzaktan yönlendirebilmek ve sensör verilerini izlemek için Visual Studio geliştirme ortamında C# programlama dili kullanılarak bir uzaktan kumanda arayüzü geliştirilmiştir. Bu arayüzün amacı hem robotu uzaktan yönlendirecek standart bir programa sahip olmak, hem de robottan alınan sensör verilerinin yorumlanabileceği bir yazılımsal sistem elde etmektir. Böylelikle robot üzerinde ileride yapılabilecek çeşitli yapay zekâ veya algoritma çalışmaları için bir alt yapı oluşturulması amaçlanmıştır.

Geliştirilen arayüzün ekran görüntüsü Şekil 6.12’de verilmiştir. Geliştirilen bu program bilgisayarın seri portuna bağlı olan radyo modülü aracılığı ile robota yönlendirme komutları göndermektedir. Yine aynı radyo modülü vasıtasıyla aldığı sensör bilgilerini ekranda görüntülemektedir. Alınan veriler ekranda dört farklı grafik olarak gösterilmiştir. İlk grafik kızılötesi uzun mesafe sensöründen alınan verileri göstermektedir. İkinci grafik ise alt sensör ve iki adet çapraz sensörler olmak üzere diğer kızılötesi mesafe sensörlerinden alınan verileri tek bir ekranda göstermektedir. Kızılötesi sensörlere ait verilerin birimleri mV’dur.

Kızılötesi sensör verilerinin yan tarafında bulunan iki grafik ise ayaklardaki basınç sensörlerinden alınan verileri mV cinsinden göstermektedir. Ayaklar sağ ve sol olmak üzere iki gruba ayrılmıştır. s1-s3 sol ayakları, s4-s6 sağ ayakları ifade etmektedir.

Program bilgisayarın seri portunu kullandığı için sağ üst köşeye seri portun açılmasını sağlayacak bir buton yerleştirilmiştir. Aynı butonun altına gerekli durumlarda grafik ekranını silmek için bir buton ve onun altına da programdan çıkmayı sağlayan bir buton eklenmiştir. Program ekranının sol üst kısmında bulunan yeşil alanda yer alan göstergeler, robot üzerindeki sensörlerin anlık değerleridir. Ekran üzerindeki yerleri robot üzerindeki yerleşimlerini andıracak şekilde belirlenmiştir. İnfrared 0 ile infrared 3 arasındaki göstergeler kızıl ötesi mesafe sensörlerinin anlık değerlerini göstermektedir. s11-s23 arasındaki göstergeler ayaklara yerleştirilmiş olan basınç sensörlerinin anlık değerlerini göstermektedir. Sağ ve sol tarafta yer alan ENGEL isimli göstergeler ise robotun ön ayaklarına yerleştirilen cisim algılama sensörlerinin anlık değerlerini göstermektedir.

Ekranın üst kısmında ortada, turuncu zeminli bölümde ise robotun yönlendirme komutları ve robottan gelen uyarıların gösterildiği bir metin kutusu bulunmaktadır. Bu komutlar ileri gitme için “1”, geri gitme için “2”, sağa dönme için “3”, sola dönme için “4” ve durma işlemleri için “0” sayısını, sağa/sola yan yürümek için ise 5/6 sayılarını göndermektedir. Bunu yaparken “65 65” başlığının devamında 1-byte olarak yönlendirme komutunu ifade eden sayıyı göndermektedir.



Şekil 6.12. Tez uygulaması için geliştirilen arayüz.

BÖLÜM 7

FPGA ÜZERİNDE GELİŞTİRİLEN DONANIM

Çalışmanın bu kısmında VHDL kodları kullanılarak seri port ve kinematik yardımcı işlemciyi içeren donanım tasarımı ele alınmıştır. Günümüzde teknolojik gelişmelere paralel olarak kapalı çevrim kontrol donanımını üzerinde bulunduran ve seri iletişim protokolleri ile kontrol edilebilen servo motorlar yaygınlaşmaktadır. Seri hat üzerinden kontrol edilebilen motorlar tek bir veri ve güç hattına bağlanarak kullanılabilen ve bu pratikte tasarımcıların işini kolaylaştırmaktadır. Kapalı çevrim kontrol donanımlarını üzerinde bulunduran bu motorlar ana işlemci üzerine düşen yükü azaltmaktadır. Özellikle robotik uygulamalarda, serbestlik derecesinin fazla olması nedeniyle, kullanılan servo motor sayısının fazla olduğu göz önünde bulundurulursa bu avantaj yadsınamaz.

Bununla birlikte çok eklemlili yapılar kinematik denklemler serbestlik derecesi ile orantılı olarak karmaşıklaşmaktadır. Özellikle hızlı, hassas ve düzgün hareketler istendiğinde bu hesaplar için bir yardımcı işlemci kullanılması zorunludur [86,87]. Bu yardımcı işlemci merkezi işlemciden gelen parametreleri kullanarak gerekli hesaplamaları yapar ve işlemci üzerindeki yükü hafifletir. Yardımcı işlemciler yoğun işlem gerektiren uygulamalarda ana işlemci üzerindeki yükü azaltmak için kullanılırlar. Çoğu uygulamada gerçek zamanlı olarak çalışmak üzere tasarlanırlar.

Bunları göz önünde bulundurarak bu kısımda, kontrolcüsünü üzerinde bulunduran ve seri haberleşme birimi ile kontrol edilebilen motorlar ile kullanılmak üzere, kinematik denklemleri çözerek gerekli eklem açılarını hesaplayan bir yardımcı işlem birimi tasarlanmıştır. Tasarlanan bu işlem birimi, her biri üç eklemlili olan altı bacaklı bir robot için gerekli eklem açılarını hesaplayarak seri iletişim birimi ile motorlara göndermektedir. Aynı zamanda eklem açılarını hesaplamak için gerekli parametreleri

ana işlemciden seri iletişim birimi aracılığı ile almaktadır. Tasarım VHDL dili ile yapılmış ve FPGA üzerinde çalıştırılarak test edilmiştir [88].

7.1. SERİ İLETİŞİM

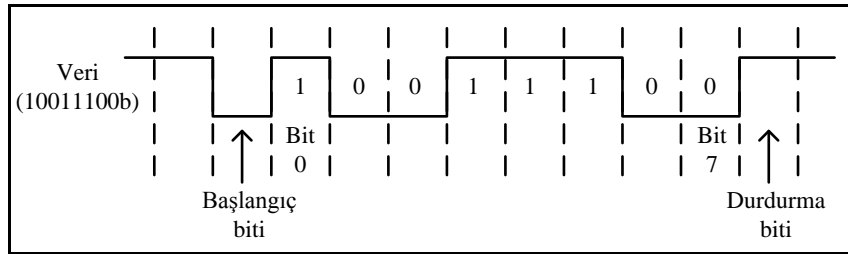
Seri port iletilecek bir verinin, herhangi bir anda, yalnızca bir bitini iletebilen elektronik bir donanımdır. Seri portlar genellikle iki yönlü (bidirectional) yapıdadır. Yani hem veri iletebilir, hem veri alabilir. Seri portun en büyük avantajları arasında, herhangi bir anda bir bit ilettiği için, az sayıdaki veri hattı ve kullanılan küçük bağlantı elemanları (konnektörler) gelmektedir [88,89].

Günümüzde seri iletişim teknolojileri gelişerek veri aktarım hızına ihtiyaç duyulan yerlerde USB (Universal Serial Bus), SATA (Serial AT Attachment) gibi teknolojiler geliştirilmiştir. Temel çalışma prensibi hepsinde aynıdır. Bu gelişim ile birlikte klasik iletişim portu RS-232 günümüzde halen kullanılmakta ve önemini korumaktadır.

RS-232 ile kurulacak seri iletişim için veri aktarım hızı 600bps ile 155200bps aralığında yaygın olarak desteklenmektedir ancak özel donanımlar daha yüksek hızlarda iletişim kurabilmektedirler. Örnek olarak bu çalışmada kullanılan motorlar 0,67Mbps hızında iletişim kurabilmektedir [81]. Burada veri aktarım hızının birimi “bps” (bit per second) saniyede aktarılacak bit sayısını ifade etmektedir. Asenkron modda veriye ait her bir bit belirli bir zaman aralığında gönderilir. Gönderilecek veri bir başlangıç biti (start bit), 7 ya da 8 veri biti, ihtiyaç duyulursa eşlik biti ve durdurma bitinden (stop bit) oluşan bir paket şeklinde gönderilir. Sıklıkla kullanılan biçim 8-N-1’dir. Bu 8 veri biti ve 1 durdurma bitinden oluşan, eşlik biti içermeyen bir formattır. Eşlik biti içerdiğinde “N” yerine “E” (even) veya “O” (odd) harfleri gelmekte ve bu harfler tek veya çift eşlik bitini ifade etmektedir [89]. Burada verilecek örnekler ve tez kapsamında yapılacak uygulama 8-N-1 biçimindedir.

Şekil 7.1’de asenkron modda 8 bitlik veri paketinin gönderilmesi sırasında oluşan sinyal gösterilmektedir. Bekleme konumunda hat mantıksal ‘1’ durumundadır. Hattın mantıksal ‘0’ konumuna geçmesiyle alıcı veri gönderileceğini anlar ve sonraki tüm

bitleri dahili saatine göre doğru zamanda okur. Seri iletişimde bps birimiyle ifade edilen, saniyede iletilen bit hızından yola çıkarak bir bitin zaman eksenindeki konumu bulunabilir. İletim hızımız 9600 bps ise bu bize veri ile birlikte başlangıç biti, durdurma ve eşlik bitleri de dahil bir saniyelik zaman diliminde 9600 bit iletileceği gösterir ve bir bitin iletimi $1/9600=0,0001s$ sürer. Alıcı başlangıç bitini alır almaz dahili saatiyle saymaya başlar ve ilk $0,0001s$ 'den sonra ilk biti, ikinci $0,0001s$ 'den sonra ikinci biti okuyup bu şekilde devam ederek durdurma bitine kadar tüm bitleri okur. Bundan sonra ikinci bir başlangıç biti gelene kadar bekleme geçer. Donanımın verici kısmı ise ayarlanmış iletim hızına göre kaydedicisine yerleştirilen 8 bitlik veriyi başlangıç, durdurma ve eşlik bitlerini üreterek seri hatta gönderir.



Şekil 7.1. Seri hat üzerinden veri iletim sinyali.

7.1.1. Servo Motorlara Ait İletişim Paketinin Yapısı

Bu çalışmada kullanılan servo motorların kontrolü seri hat üzerinden verilen komutlar ile gerçekleştirilmektedir. Motora gönderilen açı bilgisine göre, motor üzerindeki dahili PID kontrolcü rotor açısını ayarlamaktadır. Motor seri iletişimde 8 veri biti ve 1 durdurma biti kullanmakta, eşlik biti kullanmamaktadır. İletişim hızı $57,600bps$ ile $0,677Mbps$ hızları arasında ayarlanabilmektedir. Motor, uzunluğu en fazla 223-byte olan veri paketleri kullanmaktadır.

İletişim paketlerin standart yapısı Çizelge 7.1'de gösterilmiştir [81]. Paketlerde başlık standarttır ve 2-byte uzunluğundadır. Paket büyüklüğü en az 7-byte uzunluğundadır, komut ve verilere bağlı olarak maksimum 223-byte uzunluğunda olabilir. Her bir motorun kendine özgü bir kimlik numarası vardır ve buna pID denir; 0 ile 223 arasında bir değerdir. CMD motora ait kontrolcünün işleteceği asıl komuttur. Herkulex için 9 farklı komut vardır. Check Sum1 ve Check Sum2 kontrol

toplamıdır, gelen veri yığnında bir bozulma olup olmadığının kontrolü için kullanılır ve (7.1), (7.2) numaralı denklemlerde gösterildiği gibi hesaplanır [81].

Çizelge 7.1. İletişim paketi ayrıntıları.

	Başlık	Paket büyüklüğü	pID	CMD	Check Sum1	Check Sum2	Data[n]
Değer	0xFF 0xFF	7-223	0- 0xFE	1-9	-	-	-
Uzunluk (byte)	2	1	1	1	1	1	Maks. 216

$$\text{Checksum1} = (\text{PacketSize} \wedge \text{pID} \wedge \text{CMD} \wedge \text{Data}[0] \wedge \text{Data}[1] \wedge \dots \wedge \text{Data}[n]) \& 0xFE \quad (7.1)$$

$$\text{Checksum2} = \sim (\text{Checksum1}) \& 0xFE \quad (7.2)$$

Data[n] ise komuta parametre olacak veridir ve komuta göre değişmektedir. Motor üzerindeki yeşil renkli LED'i yakmak için gerekli komut 0x03'tür ve gerekli veri dizisi Çizelge 7.2'de verilmiştir. Benzer şekilde rotoru 90° konumuna taşımak için gerekli olan veri dizisi de Çizelge 7.3'de verilmiştir. Ayrıntılı bilgi için [81] numaralı kaynak incelenebilir.

Çizelge 7.2. Yeşil LED'i yakan veri dizisi.

Başlık	Paket büyüklüğü	pID	CMD	Check Sum1	Check Sum2	Data [1]	Data [2]	Data [3]
0xFF 0xFF	10	253	3	192	62	53	1	1

Çizelge 7.3. Rotoru 90° konuma götürecekle olan veri dizisi.

Başlık	Paket büyüklüğü	pID	CMD	Check Sum1	Check Sum2	Data [1]	Data [2]	Data [3]	Data [4]	pID
0xFF 0xFF	12	253	6	6	248	10	0	2	4	253

7.1.2. Tasarlanan Donanımda Kullanılan Paket Yapısı

Tasarlanan donanım iki tür paket kullanmaktadır. Birincisi parametre gerektirmeyen komutlar için kullanılmakta, ikincisi ise parametre gerektiren komutlar için kullanılmaktadır. Birinci grupta “tork aç” komut vardır. Bu komut sistemdeki tüm motorlara birden gönderilmektedir. Motorlar için tek tek bu komutun kullanılmasına ihtiyaç duyulmamıştır. Çünkü uygulamada sistem bir bütün olarak çalıştırılmakta, yine bir bütün olarak durdurulmaktadır. Sistemin normal çalışması sırasında yeterli torku üretememe, istenen pozisyona konumlanamama gibi hatalar oluşması beklenmemektedir. Motorun kontrolü ve tüm bu hatalar motor üzerindeki dahili kontrolcü tarafından kontrol edilmektedir. Hata oluştuğunda bu hatayı giderecek çözüm bu donanımı kullanmakta olan ana işlemci (bilgisayar) tarafından üretilmelidir. Burada tasarlanmış olan donanım bu tür hatalardan bağımsız olarak işlevini yerine getirmektedir. Kullanılan komutlar ve yapıları Çizelge 7.4’de verilmiştir. Parametre gerektirmeyen komut paketleri 3-byte uzunluğundadır. İlk 2-byte başlık, 3. byte komuttur. Tasarlanan sistemde 36’dan büyük bütün sayılar parametre gerektirmeyen komut olarak algılanmaktadır. 36 ve daha küçük sayılar ise konum komutu olarak algılanır (Bkz. Çizelge 7.4) ve bu sayı aynı zamanda parametre olarak gönderilecek veri sayısını belirler.

Çizelge 7.4. Genel komut yapısı.

	Başlık	Komut	Data
Tork_aç	0x41 0x41	78H	Yok
Pozisyona_git	0x41 0x41	06H-24H	6-36 byte

Konum komutu “pozisyona git” ismiyle anılan komuttur. Bu komut motorların alacağı pozisyon bilgisini içermektedir. Çizelge 7.5’de “pozisyona git” komutunun paket yapısı gösterilmiştir. Başlıktan sonra gönderilen ilk sayı 36, 30, 24, 18, 12 veya 6 ise FPGA tarafından alınan komut “pozisyona git” komutu olarak işlem görür. “Pozisyona git” komutu olarak algılanan sayılar aynı zamanda komuttan sonra gönderilen veri sayısına eşittir. Bu veriler konum parametrelerini içerir. Konum parametresinin yapısı Çizelge 7.6’da verilmiştir. Bu yapı her bir ayak için tekrarlanır.

Çizelge 7.5. İkinci grup (pozisyona git) komut yapısı.

	Başlık	Komut	Konum parametresi 1	Konum parametresi 2	Konum parametresi N (maks. 6)
Değer	0x41 0x41	6-36	-	-	-
Uzunluk (byte)	2	1	6	6	6

Çizelge 7.6. Konum parametresinin yapısı.

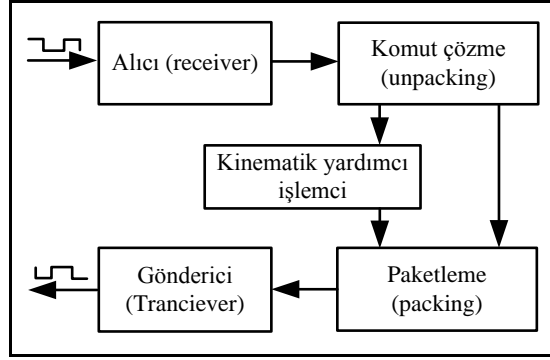
Konum parametresi						
Sembol	(+/-) _{Px}	Px	(+/-) _{Py}	Py	(+/-) _{Pz}	Pz
Değer	2B/2D	0-FF	2B/2D	0-FF	2B/2D	0-FF
Uzunluk (byte)	1	1	1	1	1	1
Anlam	İşaret	Konum	İşaret	Konum	İşaret	Konum

Konum parametresi içerisindeki değişkenler ayağın Kartezyen koordinatlardaki konumunu belirtir. İşaretleri belirten sayılar (2B/2D) +/- işaretlerinin ASCII karşılıklarıdır. Çizelge 7.6’da verilen yapı her bir bacağın X/Y/Z koordinatlarındaki mutlak konumlarını ve işaretlerini içermektedir. Her bir ayak için oluşturulan bu yapı Çizelge 7.5’de verilen yapı içerisine eklenerek FPGA’ye gönderilir.

Bütün bu paket yapıları FPGA’ye seri port üzerinden komut gönderecek olan mikrodenetleyici (veya bilgisayar) tarafından üretilir. Buradaki yapıya göre üretilen paketleri alan FPGA donanımı gerekli işlemleri gerçekleştirerek sonuçları motorlara iletir.

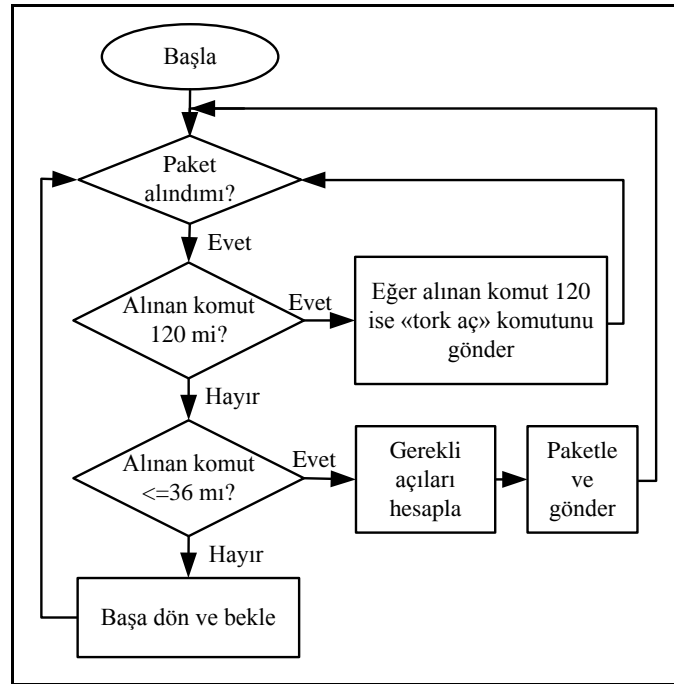
7.2. TASARLANAN DONANIMIN GENEL YAPISI

Tasarlanan donanımın genel yapısı Şekil 7.2’de verilen akış şeması ile gösterilmiştir. Öncelikle gelen veri seri port aracılığı ile alınır ve ayrıştırılır; komut ve veriler elde edilir. Eğer alınan komut bir pozisyon komutu ise alınan parametrelere göre denklemler çözülür ve elde edilen açılar, motor iletişim paket yapısına göre düzenlenerek, motorlara gönderilir. Eğer alınan komut pozisyon komutu değilse herhangi bir hesaplama ihtiyacı duyulmaz, yalnızca ilgili kodlar motorlara gönderilir.



Şekil 7.2. Genel donanım yapısı.

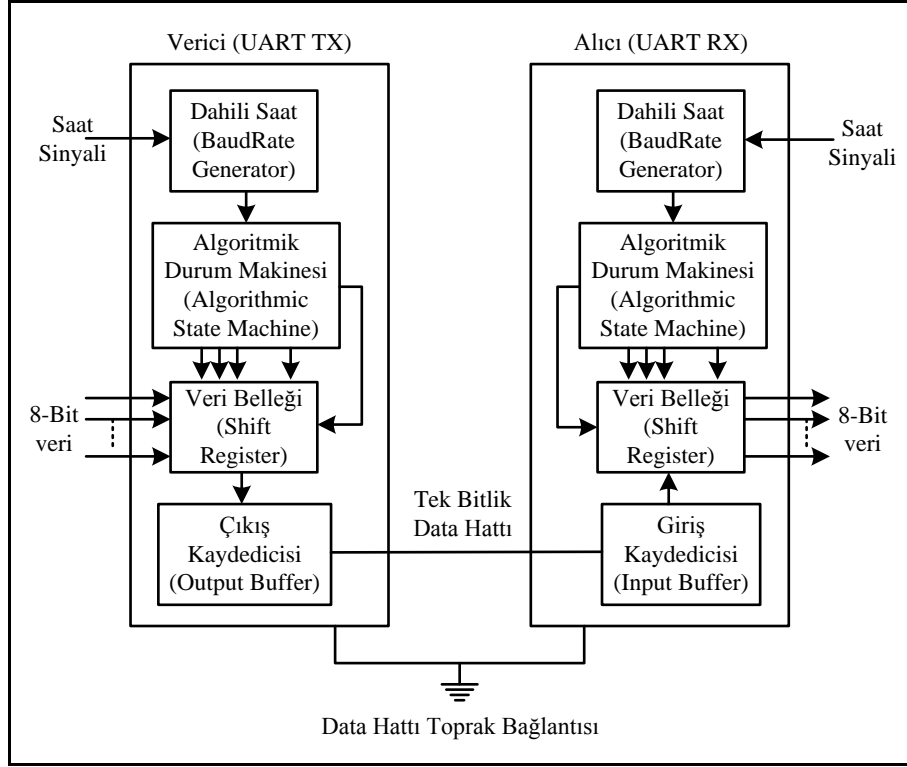
Bir paket alınmasıyla başlayan işlemler Şekil 7.3’de verilen genel akış şemasında görülen sırayla yapılır. Öncelikle başlıktan sonra gelen sayıya bakılır. Eğer sayı 78_H ise tüm motorlara ilgili komut (tork aç komutu) gönderilir ve tekrar başa dönülür. Eğer sayı ≤ 36 ise alınan veri setinden her bir ayak için konum parametreleri ayrıştırılır, bu parametreler kullanılarak kinematik hesaplar yapılır ve elde edilen açı değerleri paketlenerek motorlara gönderilir. Şemada da görüldüğü gibi algoritma hiçbir zaman son bulmamaktadır, sistem çalıştığı sürece aynı işlemleri tekrarlamaktadır.



Şekil 7.3. Genel akış şeması.

7.3. SERİ PORT DONANIMI

FPGA üzerinde tasarlanan ve uygulanan seri port donanımının alıcı ve verici kısımlarının akış şemaları Şekil 7.4’de gösterilmiştir. Blokların içerdiği donanımlar VHDL dili kullanılarak gerçekleştirilmiştir.



Şekil 7.4. Seri port alıcı ve verici akış şemaları.

Burada dahili saat (baud rate generator) ismi verilen saat, 50Mhz hızında olan sistem saatini kullanarak seri iletim için gerekli saat sinyalini oluşturur. Bu saat sinyalleri kullanarak veri belleğinde tutulan 8-bitlik veri, tasarlanan algoritmik durum makinesi vasıtasıyla, başlangıç ve durdurma bitleri de eklenerek çıkış kaydedicisine kaydedilir. Kaydedilen veriye ait her bir bit, saat sinyali kullanarak, eşit zaman aralıklarıyla veri hattına iletilir. Alıcı kısımda vericinin yaptığı işlem tersine çevrilir. Dahili saat, verici ile aynı frekansta bir saat sinyali üretir. Bu sinyal kullanarak, algoritmik durum makinesinin kontrolünde, giriş kaydedicisine belirli zaman aralıklarıyla yollanan bitler alınarak veri belleğine kaydedilir. Bunu için sistem öncelikle başlangıç bitini bekler. Başlangıç bitini aldıktan sonra, eşit zaman

aralıklarıyla hat üzerindeki sinyalleri okuyup kaydederek veriyi elde eder. Burada eşit zaman aralıkları ile kastedilen sürenin uzunluğu, sistemin iletişim hızına bağlıdır ve dahili saat tarafından ayarlanır. Örnek olarak iletişim hızı 9600bps ise bu süre yaklaşık olarak, $1/9600 = 0,0001s$ 'dir; iletişim hızı 57600bps ise, $1/57600 = 0,000017s$ 'dir.

7.4. KİNEMATİK YARDIMCI İŞLEMÇİ

Bu çalışmada tasarlanan donanımda kullanılan paketlere ait bütün veriler 8-bit uzunluğunda vektör olarak tanımlanmıştır. Alınan veriler parametre gerektirmeyen bir komuta karşılık geliyorsa ilgili komut doğrudan motorlara gönderilir. Eğer alınan veriler parametre gerektiren bir komuta karşılık geliyorsa, kullanılmadan önce, VHDL dilinin sahip olduğu fonksiyonlar ile çeşitli dönüşümlere tabi tutulurlar ve integer sayılara dönüştürüldükten sonra kullanılırlar. Elde edilen sonuçlar paketlenildikten sonra motorlara gönderilir.

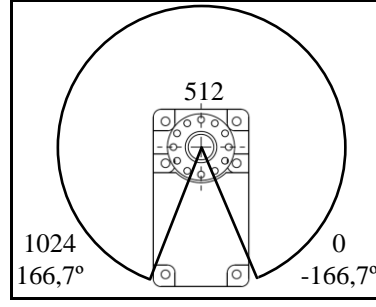
Veri alma işlemi yapılırken öncelikle başlıktan sonra gelen sayıya bakılır. Örnek bir veri dizisi Çizelge 7.7'de verilmiştir. Başlıktan sonra gelen sayı 36 veya daha küçük bir sayı ise bu bize gelen komutun bir pozisyon komutu olduğunu ifade eder ve aynı zamanda bu sayı komutun devamında alınması gereken anlamlı veri sayısını ifade eder. Örnek olarak başlıktan sonra alınan sayı 18 ise bu sayıdan sonra 18-byte daha okunmalıdır veya 36 ise bu sayıdan sonra 36-byte daha okunmalıdır. Komutun "pozisyona git" komutu olduğu anlaşılıp bütün veriler okunduktan sonra paketten P_x , P_y ve P_z noktalarının konumları sayısal olarak elde edilir. Bunun için sırayla, kayıtlı birinci veri ilk bacağın X pozisyonunun işaretini, ikinci veri ilk bacağın X konumunun işaretsiz yerini verir. Benzer bir şekilde ilk bacağın Y ve Z eksenlerindeki konum bilgileri ve aynı şekilde diğer bacakların X/Y/Z konum bilgileri bu veri dizisinden elde edilir. Dönüşüm işlemi basitçe işaret ile işaretsiz sayının çarpımından elde edilir ve bu değer matematiksel işlemlerde kullanılır. Alınan sayıların birimi milimetredir ve ayağın ilgili eksenindeki konumunu gösterir.

Çizelge 7.7. Pozisyon komutu veri dizisine örnek.

Verilerin alınma sırası	Verilerin kayıt edilme sırası	Anlamları	Açıklama
1. byte	Kayıt edilmez	Başlık	65
2. byte	Kayıt edilmez	Başlık	65
3. byte	Kayıt edilmez	Veri sayısı	36
5. byte	1	Bacağın X konumu işareti	1. bacak
6. byte	2	Bacağın X konumu	
7. byte	3	Bacağın Y konumu işareti	
8. byte	4	Bacağın Y konumu	
9. byte	5	Bacağın Z konumu işareti	
10. byte	6	Bacağın Z konumu	
...	
...	
...	
34. byte	31	Bacağın X konumu işareti	6. bacak
35. byte	32	Bacağın X konumu	
36. byte	33	Bacağın Y konumu işareti	
37. byte	34	Bacağın Y konumu	
38. byte	35	Bacağın Z konumu işareti	
39. byte	36	Bacağın Z konumu	

Px, Py ve Pz konumları alındıktan sonra bölüm 5.2’de verilen matematiksel yöntemler ile eklem açıları hesaplanır. Trigonometrik fonksiyonlar look-up-table kullanılarak gerçekleştirilmiştir. Derece olarak elde edilen eklem açılarının kullanılan servo motorlara gönderilebilmesi için bir dönüşüme tabi tutulması gerekir. Motora gönderilecek olan açı değeri 0-1024 arasındaki bir sayı olmalıdır. Denklem (7.3) kullanılarak açı değeri 0-1024 arasında ölçeklenir. Bu değerlerin karşılık geldiği açı değerleri Şekil 7.5’de verilmiştir. Ölçekleme işleminden sonra elde edilen açı değeri Çizelge 7.5’de gösterildiği gibi paketlenir ve gönderilmek üzere gönderici tamponuna yazılır.

$$\text{ÖlçeklenenAçı} = \frac{(\text{Açı} + 166,7) * 1024}{333,4} \quad (7.3)$$



Şekil 7.5. Rotor pozisyon ayrıntısı.

7.5. TASARLANAN DONANIMIN SİMÜLASYONU

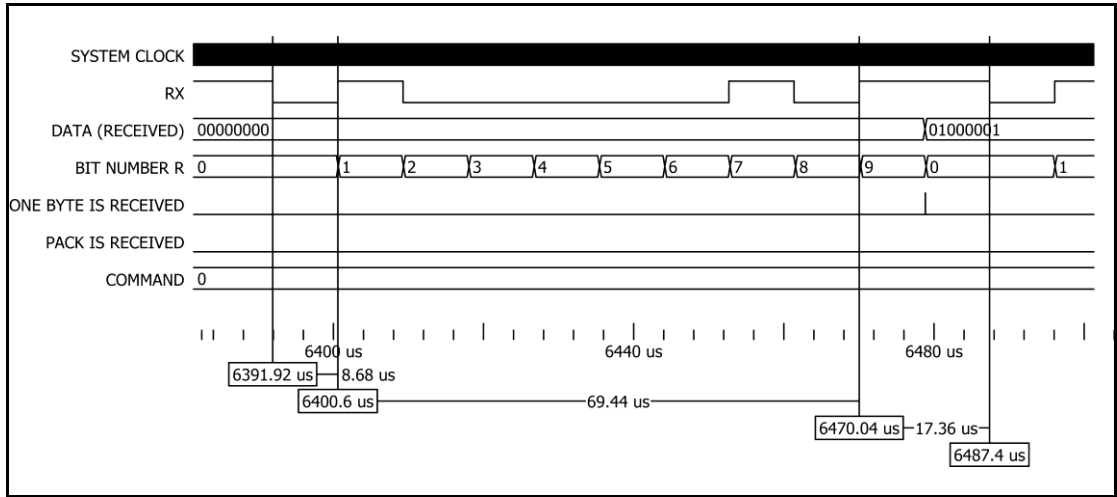
VHDL dili kullanılarak tasarlanan donanımın simülasyonu ModelSim SE-64 10.1c programı ile yapılmıştır. Sonuçları verilen simülasyon 115200 bps iletim hızında gerçekleştirilmiştir. Bu hız istenen herhangi bir değere ayarlanabilmektedir. 9600-115200 arasında iletim hızları test edilmiştir. Ana saat frekansı 50 MHz'dir.

Simülasyonda kullanılan sinyaller ve işlevleri şu şekilde sıralanabilir: “SYSTEM CLOCK” sinyali ana saat sinyalini, “RX” sinyali veri iletim hattından alınan sinyali, “TX” sinyali veri iletim hattında oluşturulan sinyali, “DATA” sinyali alınan/gönderilen 8-bitlik veriyi, “BIT NUMBER” sinyali paketi oluşturan bitlerin sırasını göstermektedir. “ONE BYTE IS RECEIVED” sinyali 8-bitlik veri alındığında üretilmekte, “PACKAGE IS RECEIVED” sinyali çok sayıda 8-bit veriden oluşan paketin alınmasıyla birlikte üretilmektedir. “ONE BYTE IS TRANSMITTED” sinyali 8-bitlik veri iletimi tamamlandığında üretilmekte ve “PACKAGE IS TRANSMITTING” sinyali çok sayıda 8-bit veriden oluşan paketin iletilmekte olduğunu göstermektedir.

7.5.1. Alıcı Simülasyonu

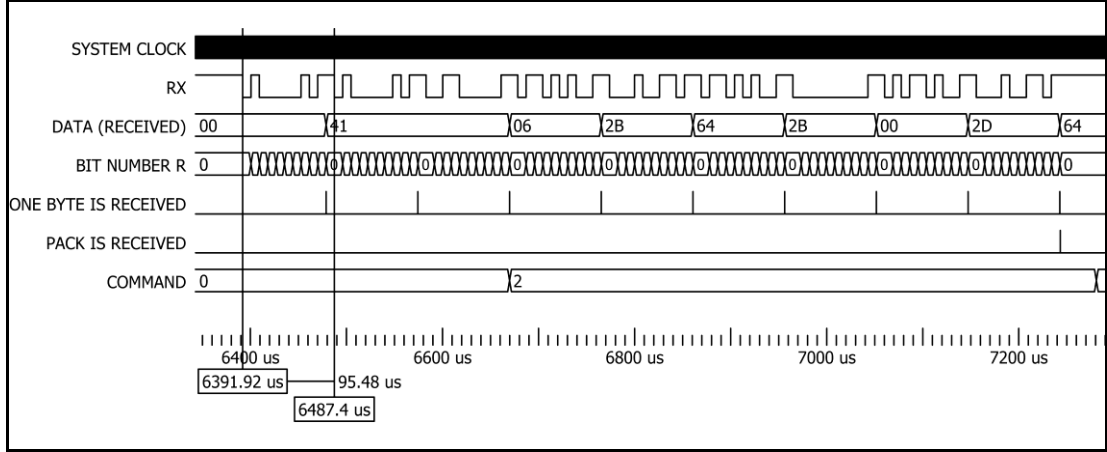
Şekil 7.6'da 8-bitlik “01000001” verisinin alınması sırasında oluşan sinyaller görülmektedir. İki durdurma biti ve sekiz veri bitinden oluşan veri paketleri kullanılmıştır. “BIT NUMBER” dahili bir sayaçtır ve başlangıç biti ile birlikte veri bitlerinin sırasını göstermektedir. “0” başlangıç bitidir. “9” numaralı bit stop bitidir. “1-8” numaralı bitler veri bitleridir. Başlangıç ve durdurma bitleri dahil her biri eşit

uzunlukta ve 115200 bps hız için $1/115200 = 8,68 \mu\text{s}$ 'dir. İlk durdurma biti geldiğinde "ONE BYTE IS RECEIVED" sinyali üretilir. Devamında bir durdurma biti daha alınmaktadır. Ancak bu sistemi etkilememektedir. Alınan iki durdurma biti, şekilde görüldüğü gibi, $2*8,68 = 17,36 \mu\text{s}$ 'dir. Başlangıç ve durdurma bitleri dahil 8-bit verinin iletimi $11*8,68 \mu\text{s} = 95,48 \mu\text{s}$ zaman almaktadır. Bu sürenin sonunda sistem bir sonraki veriyi almak için hazırdır.



Şekil 7.6. Alıcı simülasyonu.

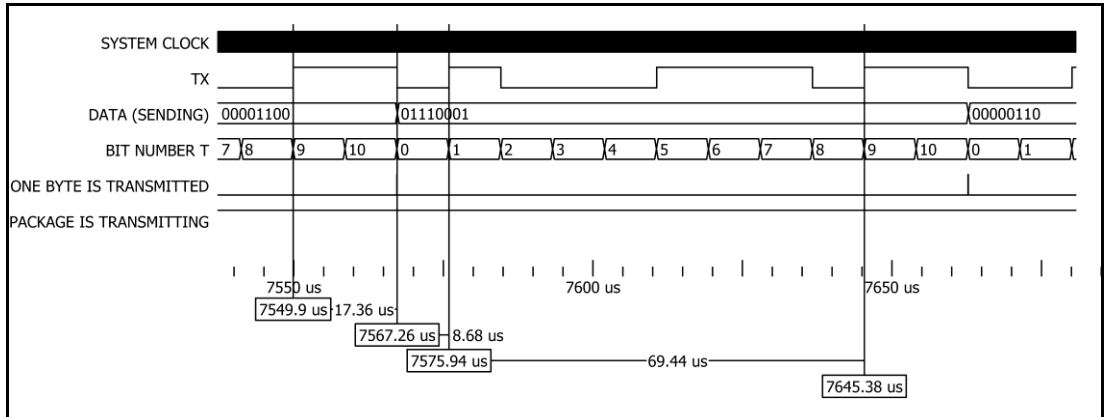
Şekil 7.7'de $(41\ 41\ 06\ 2B\ 64\ 2B\ 00\ 2D\ 64)_H$ paketinin alınması sırasında oluşan sinyaller gösterilmiştir. Bir byte verinin transferi, 1 başlangıç biti, 2 durdurma biti ve 8 veri biti ile birlikte; $9*8,68 + 17,36 = 95,48 \mu\text{s}$ sürmüştür. Her bir byte alındığında "ONE BYTE IS RECEIVED" sinyali üretilmektedir. Başlık olan 2 tane 41_H verisi alındıktan sonra 06_H sayısı alınmıştır. Bu sayı "pozisyona git" komutudur ve devamında 6-byte veri geleceğini belirtmektedir. Bu veri alınmaz alınmaz "COMMAND" sinyali 02_H (pozisyona git) değerini almıştır. Devamında 6-byte veri alınmış ve son alınan veri ile birlikte "PACK IS RECEIVED" sinyali üretilmiştir. Böylece bir paketin alınması tamamlanmıştır ve sistem bir sonraki paketi almak için hazırdır. Buradan sonra alınan komut doğrultusunda veriler işlenip motorlara gönderilecektir.



Şekil 7.7. Bir komut paketinin alınması.

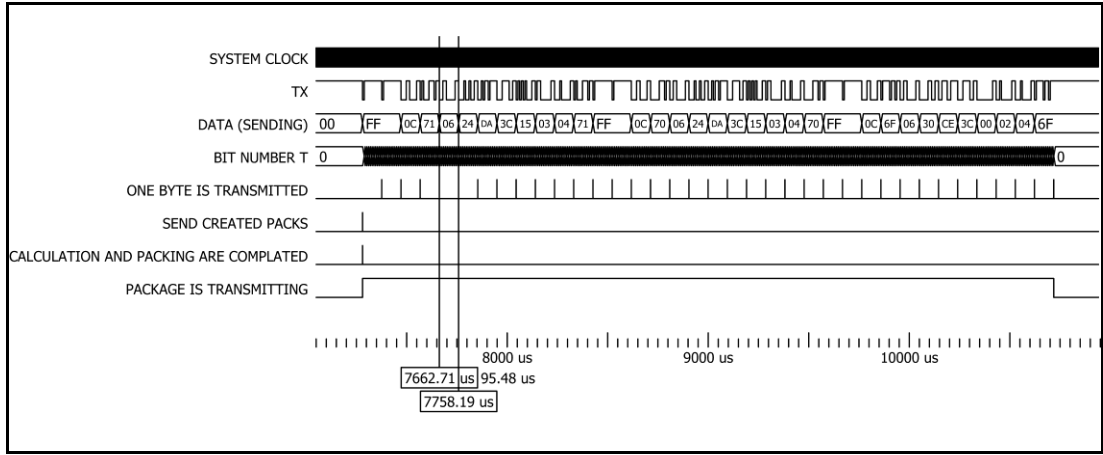
7.5.2. Gönderici Simülasyonu

Şekil 7.8’de 8-bitlik “00000110” verisinin iletim sinyalleri görülmektedir. 2 stop biti ve 8 veri biti kullanılmış, parity biti kullanılmamıştır. “BIT NUMBER” veri bitlerinin sırasını göstermektedir. “0” başlangıç bitidir. “1-8” numaralı bitler veri bitleridir. “9” ve “10” numaralı bitler durdurma bitleridir, aynı zamanda hat boşta durumudur. Başlangıç ve durdurma bitleri dahil her bir bit eşit uzunlukta ve 115200 bps hız için $81/115200 = 8,68 \mu s$ ’dir. Başlangıç ve durdurma bitleri dahil 8-bit verinin iletimi $11 * 8,68 \mu s = 95,48 \mu s$ zaman almaktadır. Bu sürenin sonunda sistem bir sonraki veriyi göndermek için hazırdır. İkinci stop bitinden hemen sonra “ONE BYTE IS TRANSMITTED” sinyali üretilmektedir. İkinci stop biti olan “10” numaralı bitten sonra sıradaki veri (00100100) gönderilmeye başlanmaktadır.



Şekil 7.8. Gönderici simülasyonu.

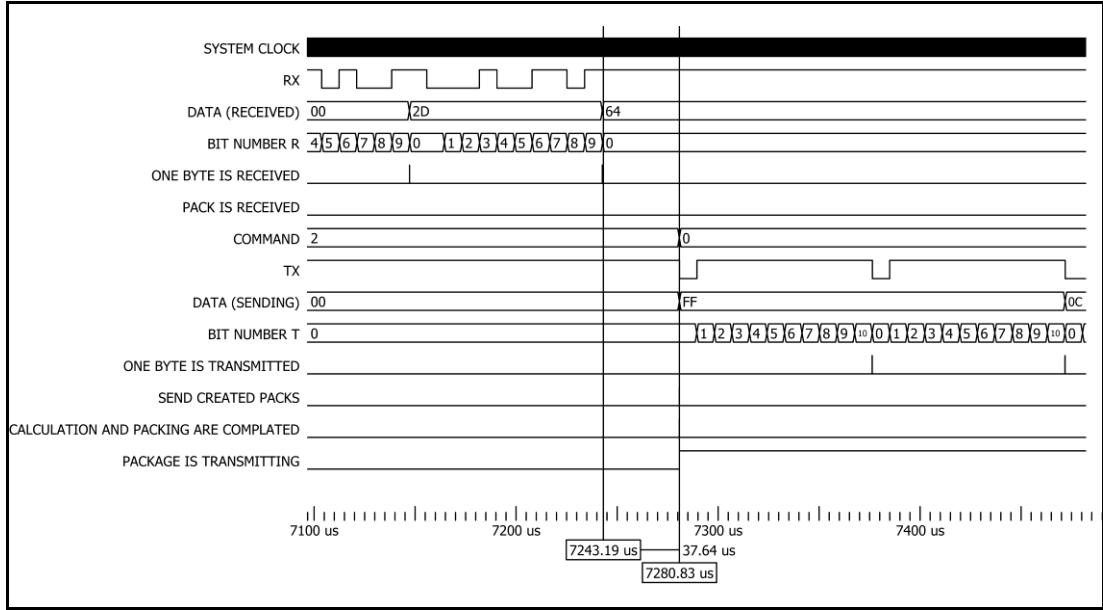
Şekil 7.9’da, “pozisyona git” komutu ile gerekli hesaplamalar yapıldıktan sonra üretilen paketin iletimi görülmektedir. “Send created packs” sinyali üretilir üretilmez sırayla paketteki veriler gönderilmektedir. Her bir byte iletiminden sonra “One byte is transmitted” sinyali üretilmektedir. Paket iletimi süresince “Package is transmitting” sinyali lojik 1 konumundadır. Paket gönderiminin tamamlanmasıyla birlikte tekrar lojik 0 olur.



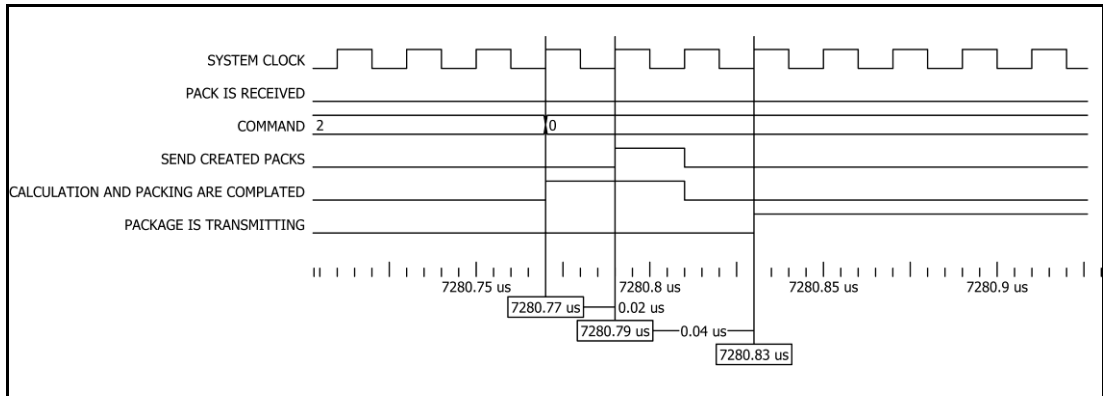
Şekil 7.9. Oluşturulan paketin gönderilmesi.

7.5.3. Yardımcı İşlemci Simülasyonu

Şekil 7.10’da paketlerin alınması ile paketlerin gönderilmesi arasındaki zaman farkı görülmektedir. “Pack is received” sinyali ile birlikte hesaplamalar ve paketleme işlemleri sırayla yapılır. İşlemin bitmesiyle birlikte “Calculation and packaging are completed” sinyali üretilir ve hemen sonrasında “Send created packs” sinyali ile birlikte paket gönderilmeye başlar. Hesaplama işleminin başlamasından veri iletiminin başlamasına kadar geçen süre 37,64 μ s’dir. Şekil 7.11’de “Calculation and packaging are completed”, “Send created packs” ve “Package is transmitting” sinyallerinin ayrıntısı görülmektedir.



Şekil 7.10. Hesaplama zamanı.



Şekil 7.11. “Calculation and packaging are completed”, “Send created packs” ve “Package is transmitting” sinyalleri.

BÖLÜM 8

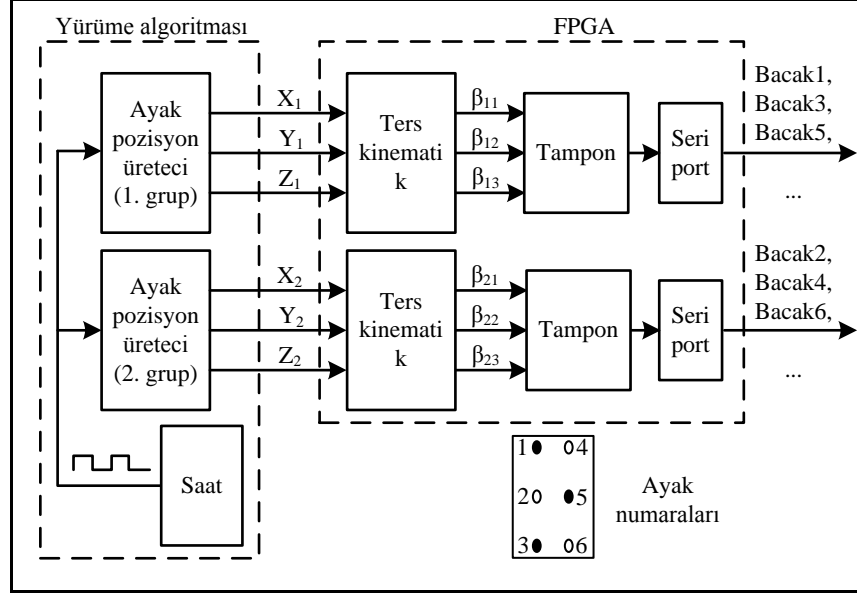
YÜRÜME ALGORİTMASININ GELİŞTİRİLMESİ

Bu kısımda FPGA üzerinde tasarlanan donanımın uygulanacağı robot için yazılan yürüme algoritması ele alınmıştır. Yürüme algoritmaları robotun yürüme hareketi yapabilmesi için gerekli adımları, ayakları birbirine uyumlu ve sıralı bir şekilde hareket ettirerek oluşturur. Altı bacaklı robotlarda tek ayak, iki ayak ve üç ayak gibi farklı yürüme şekilleri uygulanabilir. Buradaki ayak sayısı aynı anda adım atan, yani havaya kaldırılıp ileri doğru hareket eden, ayak sayısını ifade etmektedir. Aynı ortam şartlarında, ayak hareket hızlarının değişmediği düşünüldüğünde, üç ayak yürümenin daha hızlı olacağı açıktır. Bu nedenle burada üç ayak yürüme şekli uygulanmıştır. Çalışmanın bu kısmında geliştirilen algoritma ileri/geri yürümeyi, sağa/sola dönmeyi, sağa/sola yan yürümeyi sağlamakta, engelleri algılayarak üzerinden geçecek veya etrafını dolaşarak aşabilecek çıktıları üretebilmektedir.

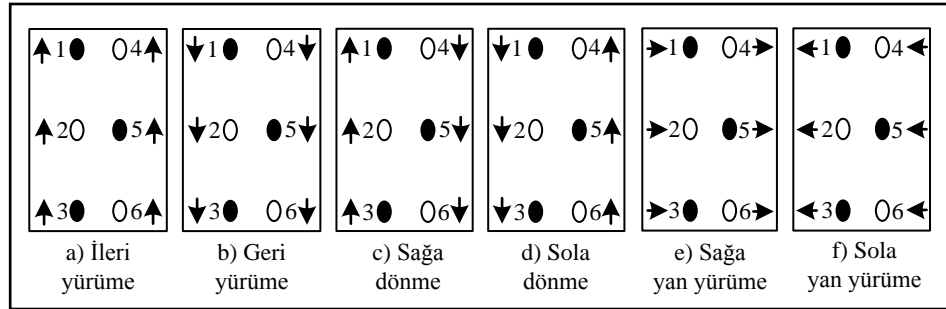
Üç ayak yürümede uygulanacak genel yöntem Şekil 8.1'de verilmiştir. Bu yürüme biçiminde altı ayak, yürürken ve dönerken, iki gruba ayrılmakta ve farklı hareketler yapmaktadır. Üç ayak yürüme düzeninde, şekilde görüldüğü gibi, 1-3-5 numaralı ayaklar (siyah ile renklendirilmiş) 1 numaralı grubu, 2-4-6 numaralı ayaklar (beyaz renkte) 2 numaralı grubu oluşturmaktadır. Gruplar için gerekli ayak pozisyonları bu bölümde açıklanan algoritmalar ile üretilir ve FPGA'ye gönderilir. FPGA üzerinde ayak pozisyonlarını elde etmek için gerekli eklem açıları hesaplanır ve bu açılar seri port üzerinden servo motorlara gönderilir. Böylelikle istenen hareket sağlanmış olur.

İleri hareket esnasında ilk ayak grubu kaldırılarak ileri hareket ettirilir ve bu ayaklar havada iken ikinci gruptaki ayakların geri hareket etmesi ile gövde ileri taşınır. Sonra havadaki ilk grup yere indirilir ve ikinci grup havaya kaldırılır. İkinci grup havada iken ilk grupta ayaklar geri hareket ettirilerek gövde ileri taşınır ve ikinci grubun yere indirilmesi ile bir çevrim sona erer. İleri giderken bütün ayaklar ve gövde ileri doğru

yer deęiřtirirken, aynı adımların ters olarak iřletilmesi ile geri hareket saęlanır ve bu durumda ayaklar ve gövdenin yer deęiřimi geriye doęru olur. Dikkat edileceęi üzere bir grup içindeki bütün ayaklar aynı yöne hareket etmektedir. İleri ve geri yürüme sırasında ayakların yer deęiřirme yönleri, Őekil 8.2.a ve Őekil 8.2.b'de verilmiřtir.



Őekil 8.1. Yürüme algoritmasına genel bir bakış.



Őekil 8.2. Hareket sırasında ayakların yer deęiřirme yönleri.

Burada dikkat edilmesi gereken bir nokta yere basan ayakların durumudur. Yere basan bir ayak ileri hareket ederse gövdeyi geriye doęru taşır, geriye hareket ederse gövdeyi ileri taşır. Ayak ile yer arasındaki sürtünme kuvvetinden dolayı ayak yer deęiřtirmez, hareketten ötürü ayaęın baęlı olduęu gövde yer deęiřtirir (zeminin kaygan olmadıęı varsayılarak bu hüküm verilmektedir; kaygan bir zeminde ise

herhangi bir hareket sırasında, ayak veya gövde yer deęişiminin ne yönde olacağını kestirmek çok zordur).

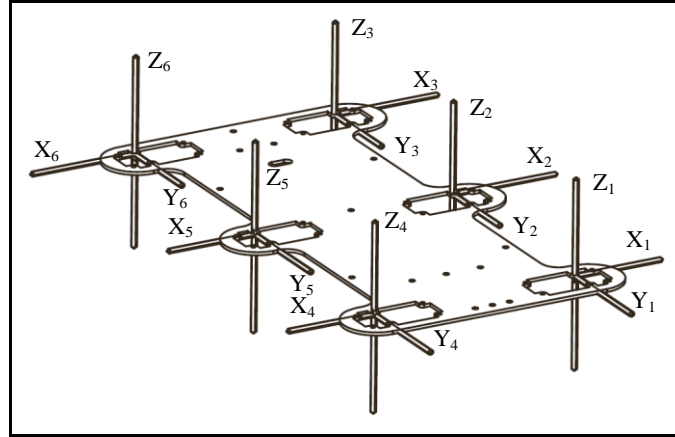
Benzer işlemler saęa ve sola dönme sırasında da yapılır. Bu hareketler esnasında ayakların yer deęiştirme yönleri, Şekil 8.2.c ve Şekil 8.2.d’de şematik olarak verilmiştir. Saęa dönme eylemi gerçekleştirilirken ilk grup havaya kaldırıldıktan sonra 1-3 ileri, 5 ise geri hareket ettirilir. Bu ayaklar havada iken ikinci grupta 2 geri 4-6 ileri hareket ettirilerek gövde bir miktar saęa döndürülür ve havadaki ayaklar indirilip dięer ayaklar kaldırılır. Havada olan 2 ileri, 4-6 geri hareket ettirilir; yerde olan 1-3 geri, 5 ileri hareket ettirilir ve havadaki ayaklar yere indirilir. Dönme eylemi devam ettięi sürece işlem tekrarlanır. Sola dönmek için ise saęa dönmek için yapılanların tam tersi yapılmaktadır.

Saęa ve sola yan yürümek için yapılan hareketler de dięerlerine benzemektedir. Bu hareketler esnasında ayakların yer deęiştirme Şekil 8.2.e ve Şekil 8.2.f’de verilmiştir. Saęa yan yürümek için ilk grup havaya kaldırıldıktan sonra yere basan ayaklar sola doğru hareket ettirilir ve böylece gövde saęa taşınmış olur. Havadaki ayaklar (ikinci grup) ise saęa doğru açılıp yere indirilir ve ilk grup havaya kaldırılıp saęa doğru hareket ettirilir. İkinci grup ise yere temas etmektedir ve sola doğru hareket ederek gövdeyi saęa taşır. Bu adımların arka arkaya tekrarlanması ile saęa yan yürüme hareketi gerçekleştirilmiş olur. Sola yan yürümek için ise bütün adımlar ters yöne doğru tekrarlanır.

Yapılan bu tanımlamalara uygun şekilde, yürüme eylemini gerçekleştirmek için gerekli bütün bu hareketleri koordineli bir biçimde kontrol eden, sensör bilgilerini okuyup yorumlayan (özellikle ayağın yere teması için) ve uzak bilgisayara gönderen, aynı zamanda uzak bilgisayardan aldığı yön komutlarına göre robotu hareket ettiren bir sistem tasarlanmıştır. Tasarlanan bu sistem farklı yükseklikteki engellerin üzerinden geçebilme yeteneğine sahiptir. Üzerinden geçemedięi engellerin ise mümkünse etrafından dolaşmakta, bunu yapamıyorsa kullanıcıdan yönlendirme için yardım istemektedir. Sistem uzak bilgisayardan gelen yönlendirme komutlarını çözerek ürettięi ayak pozisyonlarını FPGA’ye göndermek suretiyle hareketin gerçekleşmesini sağlamaktadır.

8.1. BACAKLARIN GÖVDE ÜZERİNDEKİ KONUMLARI

Yürüme algoritması geliştirilirken robot üzerindeki bacakların gövde üzerine yerleşimini Şekil 8.3’de gösterildiği tanımlanmıştır. Her bir bacak üzerindeki eksen takımı birbirinin aynıdır. Uygulama Robotunun Mekanik Tasarımı isimli bölümde oluşturulan kinematik yapı $X_4/Y_4/Z_4$ eksen takımına göre oluşturulmuştur. Eksen takımlarından 5 ve 6 numaralı olanlar 4 numaralı eksen ile aynı pozisyonda bulunmaktadır. Dolayısı ile bu üç eksen takımı üzerinde tanımlanacak konumlar eşdeğer olmaktadır. Diğer eksen takımlarını (1, 2 ve 3 numaralı) elde etmek için 4 numaralı eksen takımını Z eksenini etrafında 180° döndürülüp (motor yerleşimi bu şekilde tasarlandı); 1, 2 ve 3 numaralı bacakların merkezine taşınırsa, Y eksenlerinin Şekil 8.3’de verilen eksen takımına göre ters yönde olduğu görülür. Dolayısı ile şekildeki sistemi elde etmek için 1, 2 ve 3 numaralı eksen takımlarında, Y eksenini üzerinde bulunan noktanın -1 ile çarpılması gerekir. Bu işlem XY düzleminde hareketi temsil eden β_1 açısının -1 ile çarpılmasıyla da gerçekleştirilebilir. Böylelikle elde edilen sistem üzerinde bacak hareketlerini tanımlarken kolaylıklar sağlanmaktadır.



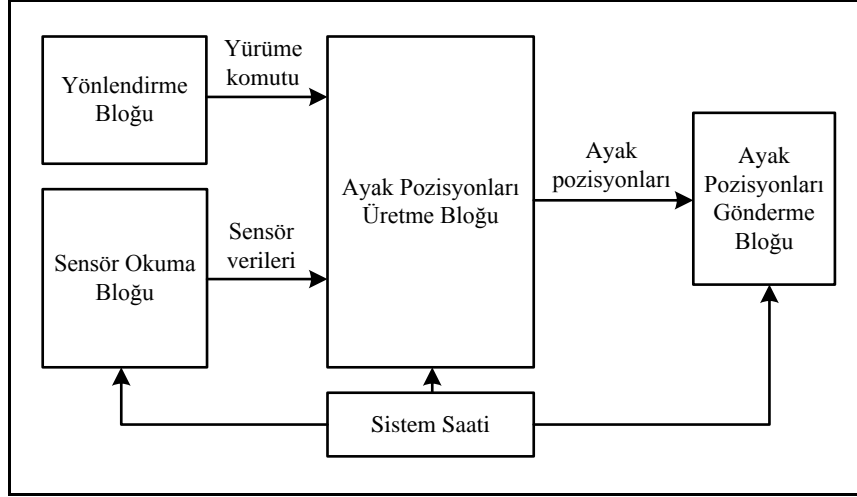
Şekil 8.3. Gövde üzerinde bacakların yerleşimi.

Örnek olarak ayakların 20mm açılması, ayakların gövdeden 20mm uzaklaşması anlamına gelir. Oluşturulan yeni eksen takımlarında, ayakların 20mm açılması istendiğinde tüm bacakların X eksenini üzerinde 20mm konumuna getirilmesi yeterlidir. Benzer şekilde ayağın ileri doğru 20mm atılması istendiğinde, tüm

bacaklar için ayağın Y eksenindeki konumun 20mm olarak ayarlanması hareketin sağlanması için yeterlidir. Aksi takdirde aynı hareketler için bacağın yönüne göre farklı işaretlere sahip çıktılar kullanmak gerekecektir. Bu nedenle robot gövdesi üzerinde şekildeki gibi bir yerleşim ve dönüşüm tercih edilmiştir.

8.2. YÜRÜME ALGORİTMASI

Geliştirilen algoritma Matlab/Simulink ortamında yazılmış ve DSP üzerine gömülerek uygulaması yapılmıştır. Oluşturulan sistemin genel yapısı Şekil 8.4’de verilmiştir. Sistemi genel olarak tanımlayacak olursak; “yönlendirme bloğu” uzaktan kumanda bilgisayarından aldığı yürüme komutunu, “ayak pozisyonları üretme bloğu”na iletir. Bu komut ileri git, geri git, sağa dön, sola dön, sağa yan yürü, sola yan yürü ve dur komutları olabilir. Bu blok aldığı komut doğrultusunda gerekli ayak hareketlerini gerçekleştirmek üzere ayak pozisyonlarını üretir. Ürettiği bu pozisyon bilgilerini “ayak pozisyonları gönderme bloğu”na iletir ve bütün pozisyon bilgileri paketlenerek FPGA’ye gönderilir. FPGA almış olduğu konum parametrelerini kullanarak ters kinematik denklemleri çözer ve eklemler için açıları hesaplar, bu açıları seri hat üzerinden motorlara gönderir ve hareketin gerçekleşmesini sağlar. Bütün bu işlemler sistem saatinin koordinasyonu ile yapılır. İşlemci saatinden farklı olan bu saat sinyalinin her çevriminde algoritma işletilmektedir. “Ayak pozisyonları üretme bloğu”, “sensör okuma bloğu”ndan gelen sensör bilgilerini de yorumlayarak ayak yüksekliğinin ayarlanması ve hareket yönünün değiştirilmesi gibi kararları verir. “Sensör bloğu” ise sistem saatinin her periyodunda sensörleri okur; okunan veriler hem kablosuz verici üzerinden kumanda bilgisayarına gönderilir, hem de algoritma tarafından kullanılır. Gönderdiği veriler arasında sensör bilgileri dışında gerçekleştirilen hareket ile ilgili bilgiler de vardır. Bu bilgiler robotun hangi işlemi yaptığını kullanıcıya bildirmek amacıyla, bilgisayar üzerindeki kullanıcı ekranında görüntülenir.



Şekil 8.4. DSP üzerinde gömülü sistemin genel yapısı.

Uzaktan kumanda bilgisayarından, “yönlendirme bloğu”na gönderilen komut, 3-byte uzunluğundadır. Komut yapısı Çizelge 8.1’de verilmiştir. 2-byte uzunluğundaki başlıktan sonra 1-byte uzunluğunda komut verisi iletilmektedir. Komut verisi olan 0, 1, 2, 3, 4, 5, 6 ve 10 değerleri sırasıyla ileri git, geri git, sağa dön, sola dön, sağa yan yürü, sola yan yürü ve dur komutlarına karşılık gelmektedir.

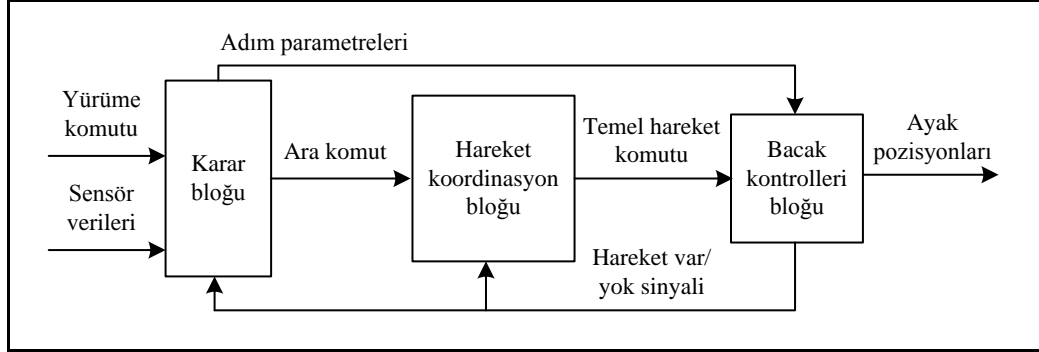
Çizelge 8.1. Komut sinyali veri yapısı.

Tür	Başlık		Komut
Değer	65	65	0, 1, 2, 3, 4, ..., 10
Uzunluk	1-byte	1-byte	1-byte

8.2.1. Ayak Pozisyonları Üretme Bloğu

Bu blok aldığı komut ve sensör bilgileri doğrultusunda, ilgili hareketlerin gerçekleşmesini sağlayacak olan ayak pozisyonlarını üretir. Genel yapısı Şekil 8.5’de verilen bu blok “karar bloğu”, “hareket koordinasyon bloğu” ve “bacak kontrolleri bloğu” olmak üzere üç alt bloktan oluşmaktadır. Karar bloğu, robota gelen yön ve sensör bilgilerine göre robotun hareketlerini belirler. Belirlediği hareketlerin gerçekleştirilmesi için ilgili komutu hareket koordinasyon bloğuna iletir ve bu blok hareketin gerçekleşmesi için gerekli komutları sıralı bir biçimde üreterek bacak

kontrolleri bloğuna iletir. Bacak kontrolleri bloğu ise bu komutlar doğrultusunda birbirinden bağımsız olarak altı bacağın hareketini gerçekleştirir.

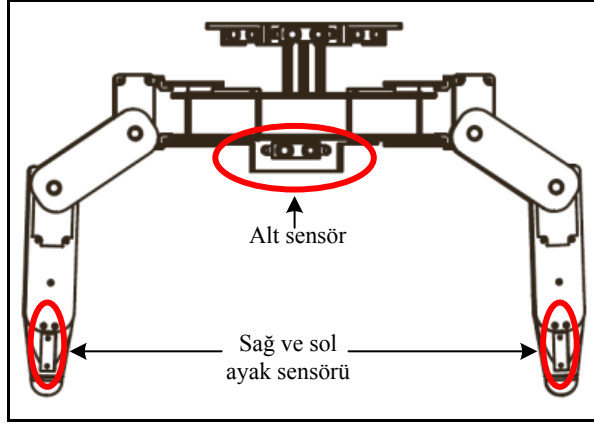


Şekil 8.5. Ayak hareketleri bloğu.

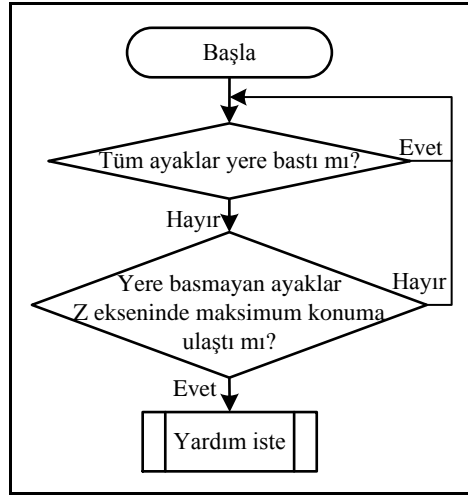
8.2.1.1. Karar Bloğu

Bu blok temel olarak robotun karar alma mekanizmasıdır. Hareket sırasında üzerinden geçebileceği veya geçemeyeceği engelleri değerlendirir ve aşabilmek için parametreleri değiştirir veya ara komutlar üretir. Girdi olarak kumanda bilgisayarından gelen komutu, yapılan son hareketin tamamlanıp tamamlanmadığı bilgisini taşıyan “hareket var/yok” sinyalini ve sensör bilgilerini kullanır. Çıktı olarak “hareket koordinasyon bloğu”na komut bilgisini, “bacak kontrolleri bloğu”na adım parametrelerini gönderir.

Karar bloğu çalışması sırasında üç temel cisim algılama sensörünü kullanır. Şekil 8.6’da yerleşimi gösterilen bu sensörlerden biri alt sensördür ve 4-30 cm algılama mesafesine sahiptir. 0-4 cm arasındaki kör bölgeyi kullanım alanından yalıtım için gövde üzerinde 4 cm geriye yerleştirilmiştir. Böylelikle sensör 0-26 cm mesafede cisim algılayabilmektedir. Diğer iki sensör ise ayak sensörleridir ve yalnızca 10 cm mesafe içerisinde bir cisim olup olmadığını belirlemekte ve ölçüm yapamamaktadır. Bunlara ek olarak basınç sensörleri de kullanılmaktadır. Bu sensörler kontrol edilerek, robot ayağının yere bastığından emin olmak için kullanılan blok Şekil 8.7’de verilmiştir. Bu blok sürekli olarak basınç sensörlerini kontrol eder. Ayaklar yere basmadıysa ve maksimum konuma geldiyse, ayağın boşlukta olduğuna veya yeterince sağlam basmadığına karar verilerek robot durdurulur ve yardım istenir.



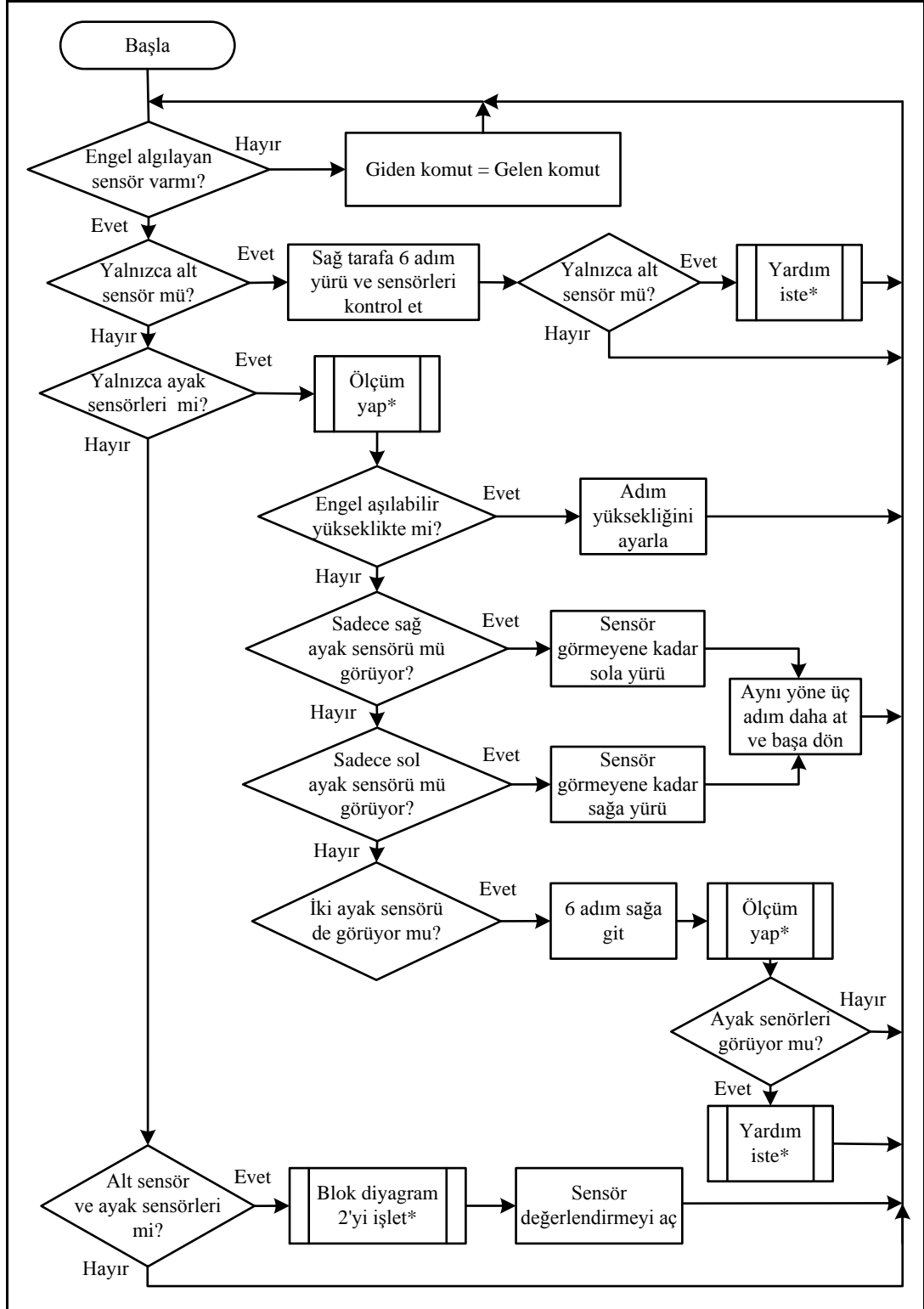
Şekil 8.6. Cisim algılama sensörleri.



Şekil 8.7. Ayağın zemine basma kontrolü.

Zemine basma kontrolü dışında, bu işleme paralel olarak yürütülen bir başka blok, karar mekanizmasının temelini oluşturur. Kullanılan üç sensörden gelen bilgileri değerlendirerek robotun engeller arasından veya üzerinden geçmesini sağlayan bu bloğun, genel blok diyagramı Şekil 8.8’de verilmiştir. Eğer hiçbir sensör cisim algılamıyorsa kumanda bilgisayarından gelen yön komutlarını değiştirmeden “hareket koordinasyon bloğu”na iletir. Eğer yalnızca alt sensör algılasa bu tam ortada bir cisim olduğunu gösterir, ayak sensörleri algılamadığı için ne yöne gideceğine karar veremez ve sağ tarafa 6 adım (bu sayı rastgele seçilmiştir, amaç cisim civarından biraz uzaklaşıp yeni durumu değerlendirmektir) ilerleyip durumu tekrar kontrol eder. Hala yalnızca alt sensör cisim algılıyorsa operatörden yardım

istemek için yardım bloğunu işletir. Durum değişti ise yeniden değerlendirmek üzere başa döner.



* Bu bloklar alt yordam şeklinde tanımlanmıştır

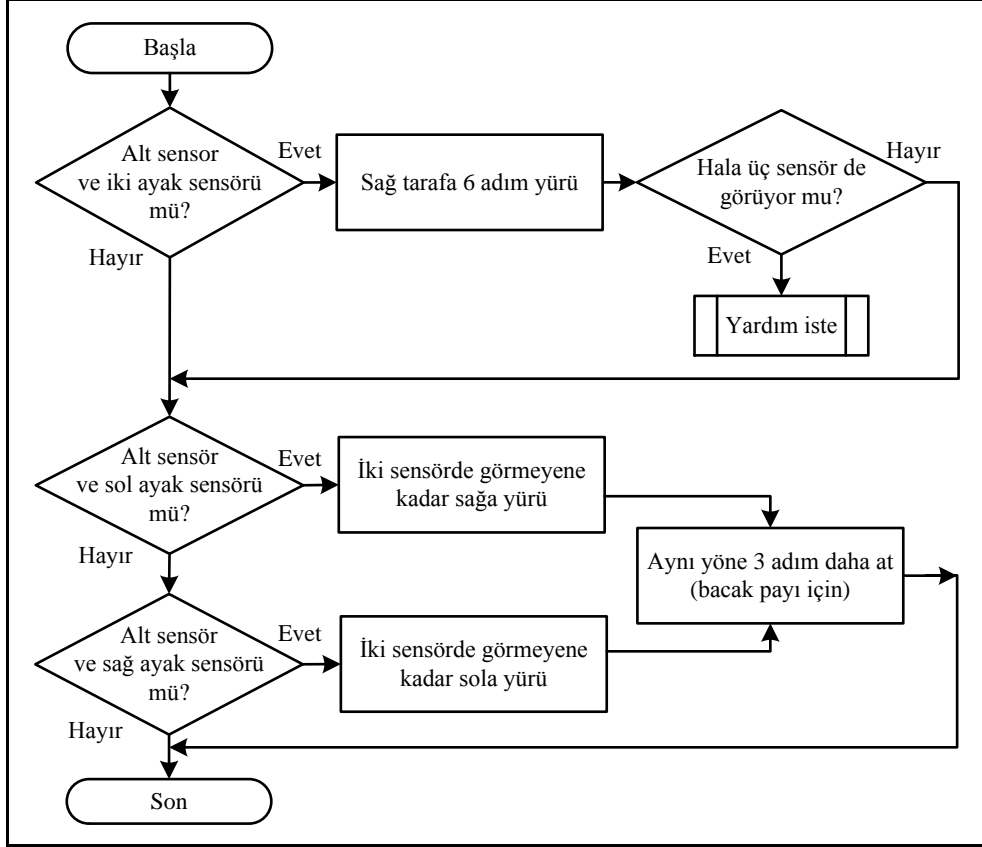
Şekil 8.8. Karar mekanizması blok diyagramı.

Yalnızca ayak uçlarında bulunan sensörlerden biri veya ikisi cisim algılıyorsa, bu durumda cismin yüksekliğini belirlemek, aşılıp aşılmayacağına karar vermek, eğer aşılabilecek ise gerekli ayak yüksekliğini belirlemek üzere ölçüm yapma bloğu işletilir. Yapılan ölçüm sonucunda cisim yüksekliği uygun ise ayak yüksekliği ayarlanarak harekete devam edilir ve cismin üzerinden geçilir. Cisim yüksekliği uygun değilse, robot üzerinden geçemeyecek demektir. Bu durumda ayak sensörlerinden hangilerinin cisim algıladığı kontrol edilir. Eğer yalnızca sağ ayak sensörü cisim algılıyorsa, boşa çıkana kadar sol tarafa yürüme komutu işletilir ve cisim geçildikten sonra; aynı yöne üç adım daha atıp, kumanda bilgisayarından gelen komutu işletilmeye başlanır. Benzer şekilde sadece sol ayak sensörü cisim algılıyorsa, boşa çıkana kadar sağ tarafa yürüme komutu işletilir ve boşa çıkıldığında; aynı yöne üç adım daha atıp, kumanda bilgisayarından gelen komut işletilmeye başlanır. Cisim aşıldıktan sonra üç adım daha atılmasının iki nedeni vardır. İlk neden sensörün tam olarak bacağın ortasında olması ve sensör görmese dahi sensör kenarlarında kalan kısmın engele çarpma olasılığıdır. İkinci neden ise ayağın hareketi sırasında üçüncü eklem dışarı doğru yer değiştirmesi ve bu kısmın engele çarpma ihtimalidir. Bu ihtimalleri sıfıra indirmek için, yola devam etmeden önce üç adım daha atılır.

Eğer her iki ayak sensörü de cisim algılıyor ise bu durumda yine gerekli kararı verecek bilgi elde olmadığı için sağ tarafa doğru 6 adım yürür ve ölçümü yeniler. Ölçüm sonrasında cisim algılanmaz ise ya da üzerinden geçebileceğine karar verirse başa dönüp kumanda bilgisayarından gelen komutu işletmeye devam eder. Eğer yine üzerinden geçemeyeceği bir engel algılar ise yardım bloğunu işletir.

Bu ihtimaller dışında eğer üç sensör de (alt sensör ve ayak sensörleri) cisim algılar ise; alt sensör gördüğü için cismin üzerinden geçilemeyecek kadar yüksek olduğu açıktır ve ölçüm yapılmaz ve Şekil 8.9'da verilen blok diyagram 2 işletilir. Üç sensör de algılama yaptığı için sağ tarafa 6 adım yürüyüp tekrar bir değerlendirme yapar. Hala üç sensör de algılama yapıyorsa yardım bloğu işletilir. Sensörlerin algılama durumu değişti ise; alt sensör ve sağ ayak sensörü algılama yapıyorsa her iki sensör de görmeyene kadar sola yürür. Benzer şekilde alt sensör ve sol ayak sensörü algılama yapıyorsa her iki sensör de görmeyene kadar sağa yürür. Sağa/sola yürüme

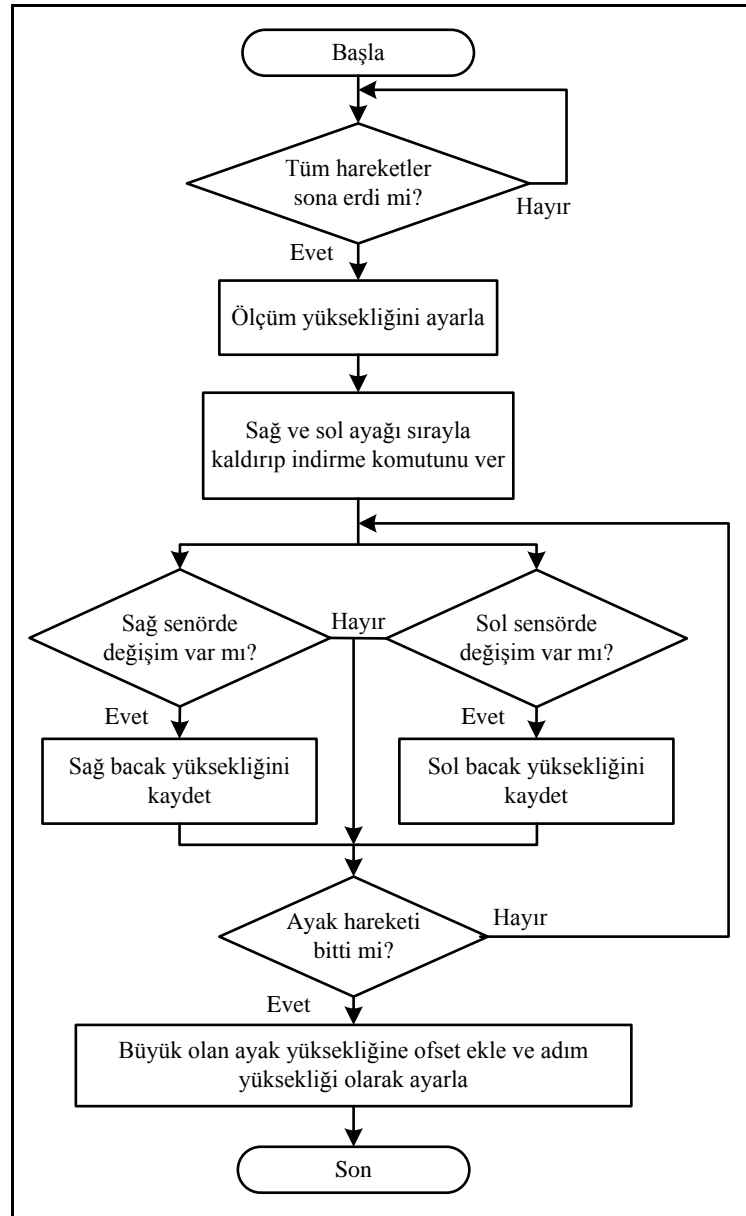
sırasında sensörlerin cismi göremedikleri adımdan sonra, bacak payı için, aynı yöne üç adım daha ilerler ve başa döner. Bu paragrafta anlatılan ihtimallerin dışında bir seçenek gerçekleşirse sistem yine başa döner ve genel karar mekanizması baştan işletilir.



Şekil 8.9. Blok diyagram 2.

Ölçüm yap bloğunun blok diyagramı Şekil 8.10'da verilmiştir. Ölçüm yapmak için öncelikle tüm hareketlerin bitmesi beklenir. Hareketler bitip robot tüm ayaklarını yere bastıktan sonra ölçüm yüksekliği ayarlanır. Ölçüm yüksekliği robotun normal yürüme sırasında ayağını kaldırdığı yükseklikten daha fazla bir yüksekliği (geçebileceği en büyük yükseklik) ifade etmektedir. Buradaki uygulamada değeri 80 olarak ayarlanmıştır. Ölçüm yüksekliğini ayarladıktan sonra sağ ve sol ayakları sırasıyla ölçüm yüksekliği kadar kaldırıp indirecek olan hareket komutu işletilir. Bu hareketin yapılması sırasında ise ayak sensörleri denetlenerek cisim yüksekliği yaklaşık olarak tespit edilir. Bunun için sürekli olarak ayak sensörü çıkışında bir değişiklik olup olmadığına bakılır. Eğer ayak sensörü bir cisim algılıyorsa ve

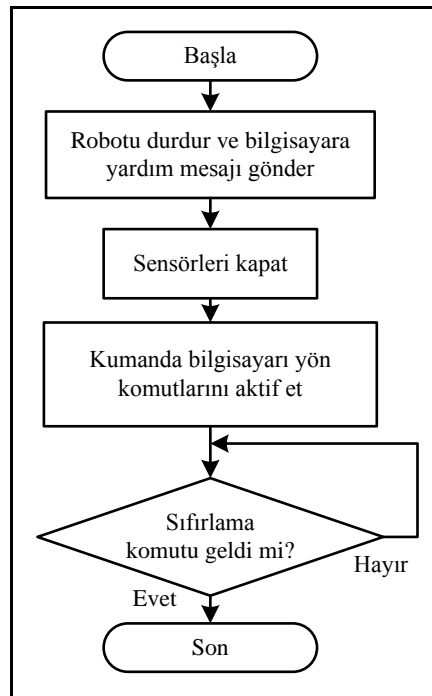
geçebileceği yükseklikte ise; ayak o yüksekliğin üzerine çıktığında ayaktaki cisim sensörünün çıkışı değişecektir, çünkü o yükseklikten sonra cisim algılamayacaktır. Tam bu değişimin olduğu andaki ayak yüksekliği her iki ayak içinde kaydedilir. Eğer cisim olmadığı için ya da cisim çok yüksek olduğu için ayak sensöründe değişim olmaz ise ilgili ayağın yükseklik değeri sıfır olarak kalır. Değerler sıfırdan farklı ise büyük olan değere bir ofset eklenerek yeni ayak yükseklik değeri olarak atanır. Eklenen ofsetin nedeni sensörün ayağın en alt noktasına yerleştirilememiş olmasıdır. Sensörün yerden yüksekliği kadar bir ofset bu değere eklenir.



Şekil 8.10. Ölçme bloğu.

Yapılan uygulamada sensörlerden ve mekanik yapıdan kaynaklanan nedenler ile kesin bir ölçüm yapılamamaktadır. Bu nedenle algılanabilecek cisim boyları üçe ayrılmıştır. İlk sırada küçük cisimler vardır. Yüksekliği yaklaşık olarak 2cm'den küçük olan cisimler ayaklardaki sensörler ile algılanamamaktadırlar ve yine bu cisimler robotun olağan yürüme sırasındaki ayak yüksekliğinden küçüktürler. Böylelikle robot bu engelleri görmese dahi aşabilmektedir. Bu 2 cm'den büyük ve sensörlerin görebildiği cisimlerin üzerinden geçmek için ise ayak yüksekliği şu şekilde ayarlanır; eğer algılanan cismin yüksekliği ayağın dikeyde 30mm'den küçük ise ayak yüksekliği 46mm olarak ayarlanır. Eğer cisim yüksekliği 45mm ile 30mm arasında ise ayak yüksekliği 65mm olarak ayarlanır.

Robotun üstesinden gelemeyip yardıma gerek duyduğu durumlarda Şekil 8.11'de verilen yardım bloğu işletilmektedir. Robot yardıma ihtiyaç duyduğunda durdurulur, bilgisayardaki kullanıcı arayüzüne yardım mesajı gönderilir, sensörler kapatılır ve kumanda bilgisayardan gelecek komutlar olduğu gibi işletilir. Bu sırada sürekli sıfırlama komutu gelip gelmediğine bakılır. Kullanıcı istediği bir anda arayüz üzerindeki sıfırla butonuna basarak yardım bloğundan çıkar ve algoritma en başından işlemeye devam eder.

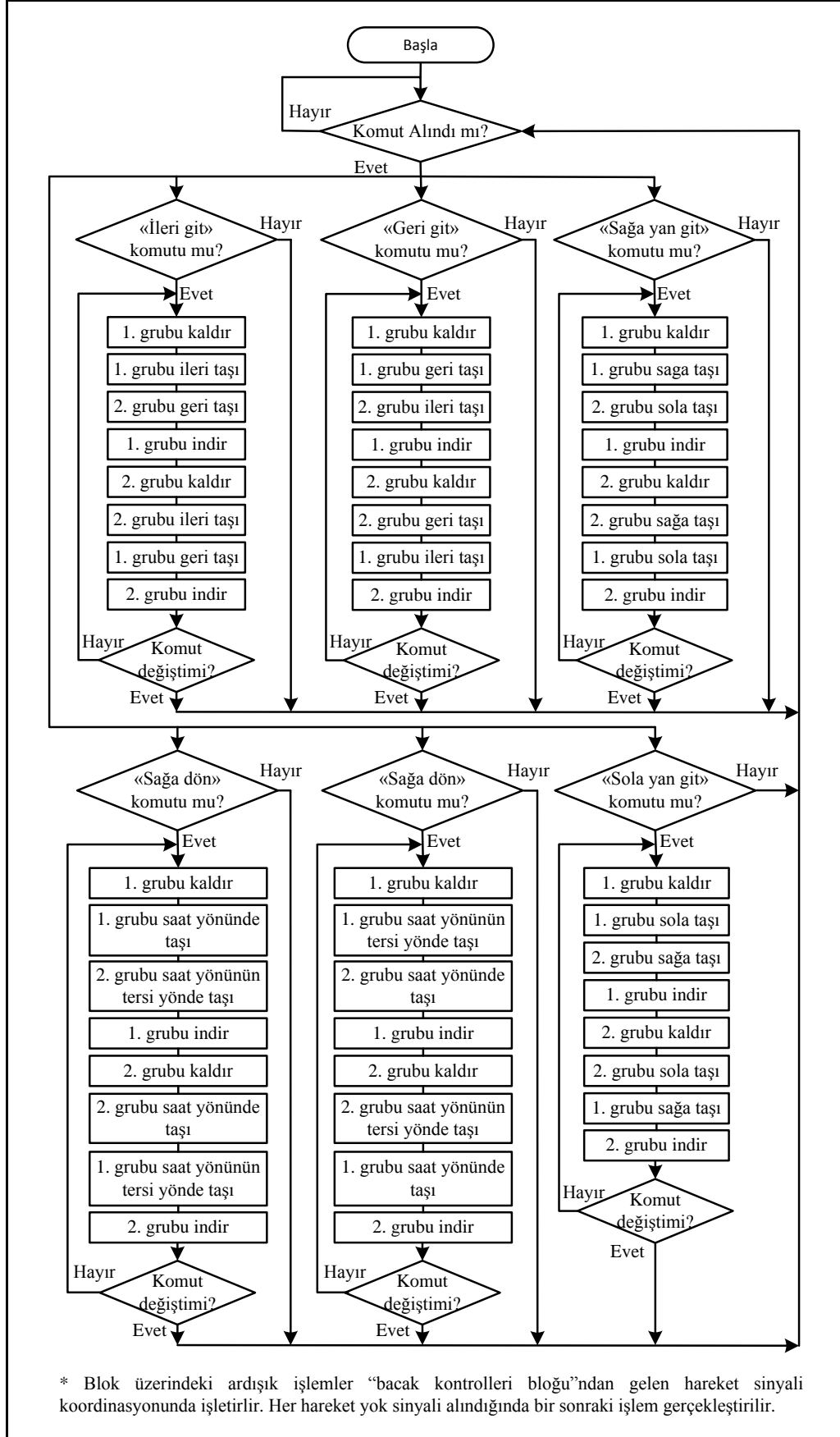


Şekil 8.11. Yardım bloğu.

8.2.1.2. Hareket Koordinasyon Bloğu

Bu bloğun görevi ileri ve geri hareket etmek, sağa/sola dönmek, sağa/sola yan yürümek için, bölümün girişinde bahsedilen şekilde, gerekli temel ayak hareketlerini sıralı ve düzgün olarak koordine etmektir. Hangi hareketin yapılacağına karar bloğundan gelen ara komut ile karar verir. Bu komut 0, 1, 2, 3, 4, 5, 6 ve 10 değerlerini alabilir. 0 durma, 1 ileri gitme, 2 geri gitme, 3 sağa dönme, 4 değeri sola dönme, 5 değeri sağa yan yürüme, 6 değeri sola yan yürüme ve 10 değeri yürüme algoritması parametrelerini ilk ana döndürme (sıfırlama) anlamına gelir. Aynı zamanda burada verilenler dışında bir değer gelirse, anlamsız bir değer olduğu için yine durma komutu olarak algılanır. Bloğun bu değerlerden birini bir an görmesi bir çevrim olarak hareketi yapması için yeterlidir. Yani blok 1 değerini alırsa ilgili ayakları havaya kaldırıp, yere basan ayaklar ile gövdeyi ileri taşır ve havadaki ayakları yere indirir; sonra bir önceki kısımda yere basan ayakları havaya kaldırır, gövdeyi ileri taşır ve tekrar havadaki ayakları yere indirerek bir çevrimi tamamlar. Bu çevrim tamamlandıktan sonra tekrar ara komutu okur ve yeni komuta göre hareket eder. Bu işlemleri yaparken ayak sensörlerinden gelen bilgileri de yorumlar. Bloğun yapısı Şekil 8.12’de verilmiştir.

Bu blok bir hareketi gerçekleştirmek için, içerisinde tanımlanmış olan temel hareketleri belirli bir sıra ile tetikleyerek birleşik bir hareketin gerçekleşmesini sağlar. Bu temel hareketler, bir ayağı kaldırma/indirme, iler/geri taşıma ve sağa/sola dönme gibi temel hareketlerdir. Bu hareketlerin hiç birisi tek başına yürüme için yeterli değildir. Bu hareketlerin belirli bir sıra ve zamanlama ile birleştirilmesi sonucu yürüme veya dönme hareketleri meydana gelecektir. Örnek olarak ileri hareket etmek için adım üretme bloğu “1. grubu kaldır”, “1. grubu ileri taşı”, “2. grubu geri taşı”, “1. grubu indir”, “2. grubu kaldır”, “2. grubu ileri taşı”, “1. grubu geri taşı”, “2. grubu indir” komutlarını, ilgili hareketler tamamlandıkça, birbiri ardına göndermektedir. Gönderilen bu komutlar bacak kontrolleri bloğu tarafından işletilerek hareket gerçekleştirilir.



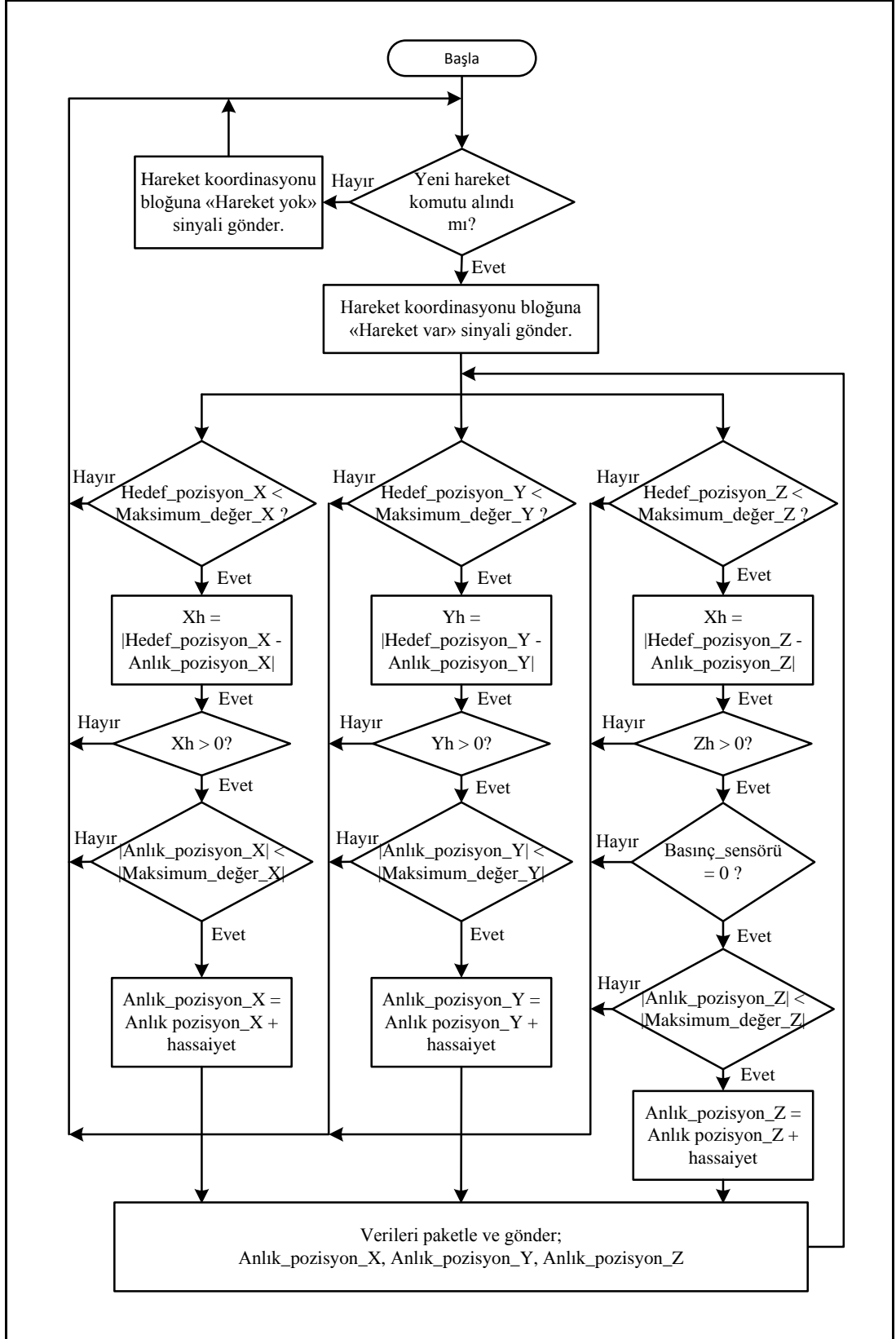
Şekil 8.12. Hareket koordinasyon bloğu.

Burada kullanılan algoritmada yürüme komutları “ayağı ileri at”, “ayağı yukarı kaldır”, “ayağı geri çek” gibi alt komutlar ile tanımlanmıştır. Doğal olarak bu hareketlerin bir ölçüsü vardır. Normal şartlarda ayak ne kadar ileri atılmalı (adım büyüklüğü ne olmalı), normal bir duruş pozisyonunda gövde yüksekliği ve ayak açıklığı ne kadar olmalı gibi ölçüler, robotun fiziksel limitleri de göz önünde bulundurularak, sisteme parametre olarak girilmektedir. Sistem bu parametreleri göz önünde bulundurarak temel adım hareketlerini oluşturmaktadır. Bu sistem için öngörülen değerler karar bloğu tarafından belirlenip “adım parametreleri” olarak “bacak kontrolleri bloğu”na gönderilir.

8.2.1.3. Bacak Kontrolleri Bloğu.

Bu blok her bir bacağın temel hareketleri yerine getirebilmesi için gerekli fonksiyonları işletir. Bu temel hareketler bacağı kaldırma, indirme, ileri ve geri taşıma gibi hareketlerdir. Bunları teker teker veya ardışık olarak bir arada yaptırabilir. Şekil 8.13’de bloğun yapısı verilmiştir. Her bir bacak için bağımsız bir birim vardır. Bu birimler tamamen birbiri ile aynı yapıya sahiptir.

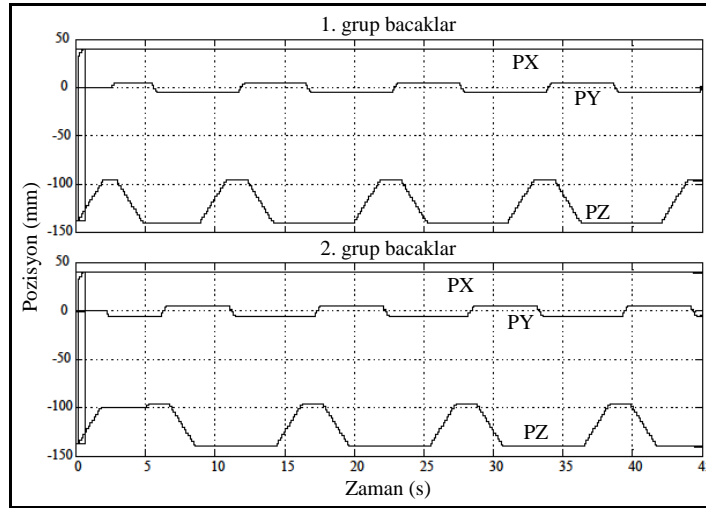
Bloğa, gerçekleştirmesi istenen komut, karar bloğundan gelir. Bu komut aslında hedeflenen X/Y/Z pozisyonlarını içermektedir. Her yeni komut geldiğinde, öncelikle hedef pozisyon değerinin, tanımlanan maksimum değerden büyük olup olmadığı kontrol edilir. Eğer değer uygunsa, ayağın o anki pozisyonu ile hedef pozisyon karşılaştırılır; arada fark var ise farkı kapatmak için, her çevrimde, anlık değere hassasiyet değeri eklenir. Hassasiyet değeri, anlık iki pozisyon arasındaki olağan değişim miktarıdır ve ön tanımlıdır. Hassasiyet değerinin eklenmesi işlemi, anlık değer maksimum değere ya da hedef pozisyona ulaşana kadar tekrarlanır. İşlem tamamlanınca başa dönülüp yeni komut beklenir. Anlık değerler her hesaplandığında, BÖLÜM 7’de verilen FPGA donanımı için, bir iletişim paketi oluşturulur ve FPGA’ye gönderilir. Z ekseni ile ilgili akışta, diğer eksenlerden farklı olarak, anlık değer her değişiminden önce ayaklardaki basınç sensörü kontrol edilir. Sensör bilgisi “0” ise işleme devam edilir. “1” ise, ayak zemine bastığı için, işlem o anda durdurulur ve yeni komut için başa dönülür.



Şekil 8.13. Bacak kontrolleri bloğu.

8.3. SİMÜLASYON ÇIKTILARI

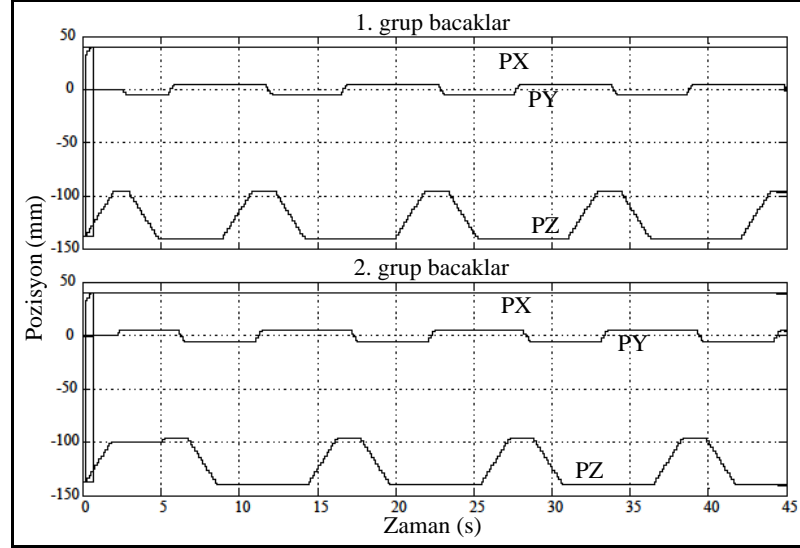
Geliştirilen yürüme algoritmasının ürettiği ayak pozisyon grafikleri Matlab/Simulink programı ile elde edilmiş ve bu kısımda incelenmiştir. Robotun ileri ve geri hareketleri sırasında elde edilen sinyaller Şekil 8.14 ve Şekil 8.15’de verilmiştir. İlk saniyede ayaklar başlangıç pozisyonunu almakta ve bundan sonraki süreçte verilen komuta göre pozisyon değiştirmektedir. Örnek bir durum Şekil 8.14’de incelenmiştir. Onuncu saniyede birinci grup ayaklar, Z ekseninde hareket ederek, havaya kalkmaktadır. Bu ayaklar havaya kalktıktan hemen sonra yere basan ikinci grup ayaklar Y ekseninde konum değiştirerek gövdeyi ileri taşırlar. Bu işlemden sonra havadaki ayaklar ileri taşınır ve yere indirilir. Birinci gruptaki ayaklar yere bastıktan sonra ikinci gruptaki ayaklar Z eksenindeki hareket ile havaya kaldırılır. İkinci grup havaya kalktıktan sonra yere basan birinci grup Y ekseninde hareket ederek gövdeyi ileri taşır; hemen sonrasında havadaki ayaklar ileri taşınıp yere indirilirler. Sonra aynı işlemler tekrarlanır. Görüleceği üzere yere basan ayakların pozitif konumdan negatif konuma doğru hareketi gövdeyi ileri taşımaktadır. Ayaklar yer ile arasında olan sürtünme kuvvetinden dolayı konum değiştirmemekte ancak hareketin gerçekleşmesine izin vermek için gövde yer değiştirmektedir.



Şekil 8.14. İleri hareket sırasında oluşan ayak yerdeğişimleri.

Geri hareket içinde aynı işlemler ters yönde yapılır. Gövde yere basan ayaklar tarafından ileri değil geri doğru taşınır. Yine onuncu saniyede ilk grup ayaklar

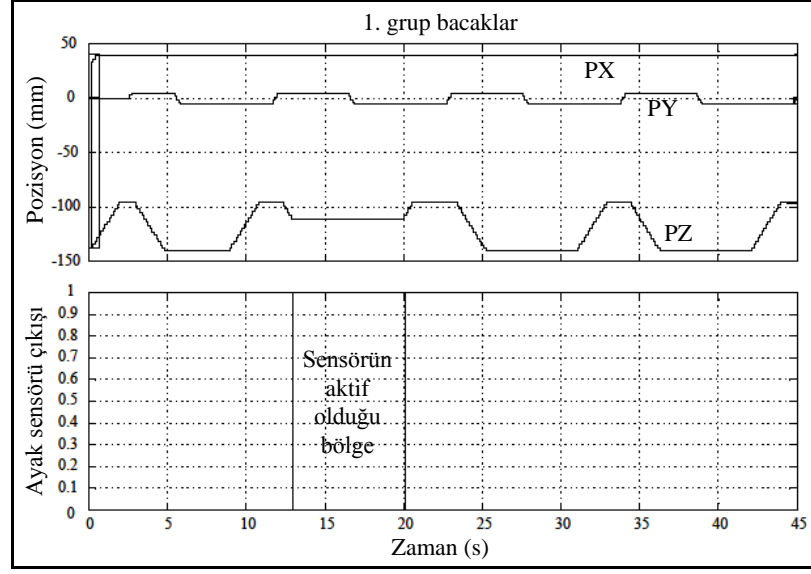
havaya kalkmaktadır. Bu ayaklar havaya kalktıktan hemen sonra yere basan ikinci grup ayaklar Y ekseninde gövdeyi geri doğru taşırlar, yani negatif pozisyondan pozitif pozisyona hareket ederler ve ayaklar yere bastığı için gövde geri taşınır.



Şekil 8.15. Geri hareket sırasında oluşan ayak yer deęişimleri.

Saęa sola dönme eylemine ait grafikler de ileri/geri gitme eyleminin grafikleri ile aynıdır. Farklı olan şey grup içerisinde ayakların yaptığı hareketlerdir. İleri/geri hareket sırasında grup içerisindeki tüm ayaklar aynı hareketi yapmaktadır. Saęa sola dönme sırasında ise robotun aynı yönde olan iki ayağı ile dięer yönde olan tek ayağı (tüm hareketler üç ayak üzerinde yapılmaktadır) ters yönlerde hareket etmektedir.

Ayak sensörü cisim algıladıęında gerçekleşen ayak pozisyonları Şekil 8.16'da verilmiştir. Ayağın 13. Saniyede cismin üzerine basınca sensörden "1" çıkışı alınmaktadır. Bu durumda ayak aşağı olan hareketini keserek gövde pozisyonu ve dengeyi korumaktadır. Ayağını cismin üzerinde tutarak gövdeyi hareket ettirmekte ve 20. saniyede ayağı kaldırmaya başlamaktadır. Bir sonraki adımında, cisim ilgili ayağın altında olmadığı için (gövdenin ileri taşınması ile birlikte ayak da pozisyon deęiştirdi) normal adım atarak yoluna devam etmektedir.



Şekil 8.16. Ayağın yüksek bir cisme basması sırasındaki yer değişimleri.

BÖLÜM 9

UYGULAMA VE TEST SONUÇLARI

Bu bölümde, FPGA üzerinde geliştirilen donanım ve bu donanımın uygulaması için geliştirilen robot ile ilgili test ve deney sonuçları sunulmaktadır. Öncelikle FPGA üzerine inşa edilen kinematik yardımcı işlemci ile ilgili test sonuçları verilmiştir. Genel uygulama için geliştirilen elektronik donanım, seri port aracılığı ile bilgisayara bağlanmış ve gerekli testler yapılmıştır. Arkasından, elektronik ve mekanik donanım bir bütün olarak test edilmiş, yürüme sırasında kaydedilen fotoğraf kareleriyle, fonksiyonel açıdan incelenmiştir. Son olarak ortaya çıkan robot prototipinin performans değerlendirmesi yapılmıştır.

9.1. FPGA ÜZERİNDE GELİŞTİRİLEN DONANIMIN TESTLERİ

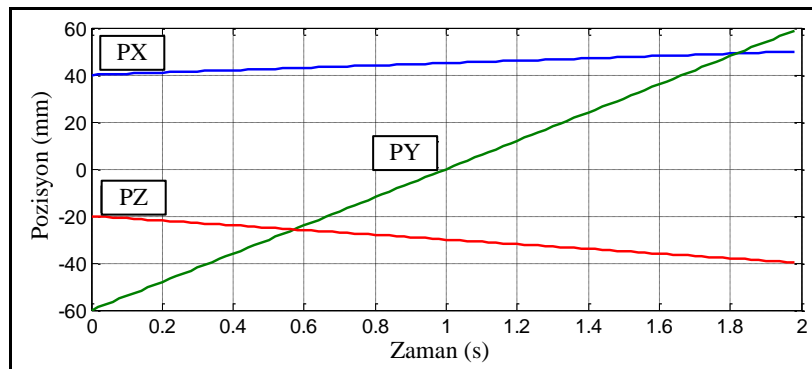
Tasarımı ve simülasyon sonuçları BÖLÜM 7’de verilen donanımın, fonksiyonel olarak test edilmesi, gerçek donanımın aldığı komutlar doğrultusunda doğru sonuçları üretip üretmediğinin görülebilmesi için bir test sistemi hazırlanmıştır. Test için BÖLÜM 6’da verilen anakart tasarımının sadeleştirilmiş hali kullanılmıştır. Şekil 6.8’de verilen anakart üzerinde yalnızca DSP’nin FPGA ile iletişimini sağlayan bağlantılar, FPGA çıkışına bağlı tampon ve güç devresi bulunmaktadır. DSP tarafından üretilen, ayak konumlarını içeren, pozisyon komutları DSP’nin seri portu üzerinden FPGA’ye gönderilmektedir. Bu komutlar doğrultusunda FPGA’in ürettiği açılı bilgileri seri port aracılığı ile bilgisayara aktarılmakta ve FPGA tarafından üretilen sonuçlar ile bilgisayar üzerinde, simülasyon tarafından üretilen sonuçlar karşılaştırılmaktadır.

Test sistemi bilgisayara seri port aracılığı ile bağlanmıştır. Test sisteminde, seri porttan veri göndermek ve gönderilen verileri görüntülemek için RealTerm isimli

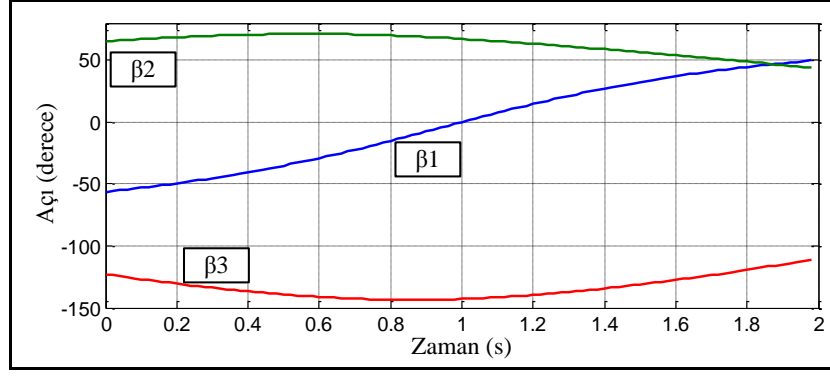
terminal programı kullanılmıştır. Bu program vasıtası ile seri port üzerindeki sinyallerin durumu gözlenebilmekte, değişik protokoller ve hızlar ile seri port üzerinden veri alış verişi yapılabilmektedir. Alınan ve gönderilen veriler değişik formatlarda (decimal, hexadecimal, ascii, vb.) kullanılabilmekte ve kaydedilebilmektedir.

Tasarlanan test için giriş sinyallerini üretecek basit bir program Matlab/Simulink programı ile oluşturulmuş ve DSP üzerine gerçek zamanlı olarak çalıştırılmıştır. Oluşturulan program, her bir ayak için, X/Y/Z ekseninde sıralı pozisyonlar üretmektedir. Üretilen bu pozisyonlara karşılık gelecek olan eklem açıları yine bu program aracılığı ile hesaplanmıştır. Üretilen pozisyonlar, FPGA üzerindeki donanıma, Matlab/Simulink üzerinde oluşturulan program aracılığı ile seri port üzerinden gönderilmiştir. FPGA donanımının ürettiği çıkışlar Realterm isimli terminal programı ile alınmış ve yine Matlab/Simulink programı ile grafik olarak görüntülenmiştir.

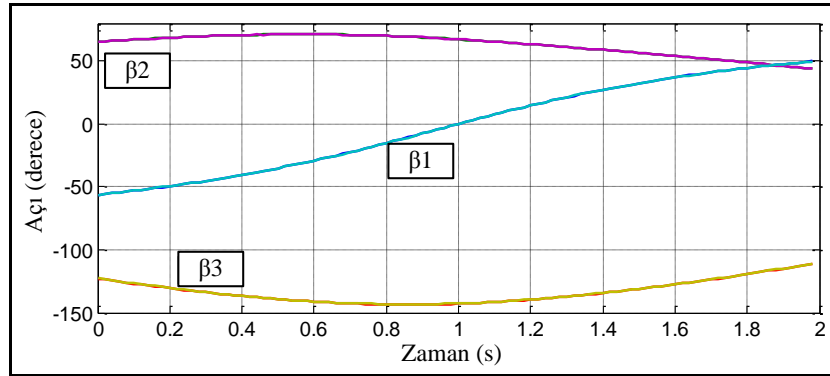
Deneme amaçlı oluşturulan giriş sinyalleri Şekil 9.1’de verilmiştir. Matlab/Simulink programı ile DSP için yazılan program aracılığı ile bu değerler seri port üzerinden FPGA’ye gönderilmiştir. Bu girişlere karşılık olması gereken eklem açıları Şekil 9.2’de verilmiştir. Şekil 9.3’de FPGA’in ürettiği çıkışlar ile simülasyonun ürettiği çıkış karşılaştırmalı olarak verilmiştir. Oluşan en büyük hata 1° ’den küçüktür. Deneme 50 komut cümlesi kullanılarak yapılmıştır.



Şekil 9.1. Ayağın X,Y,Z pozisyonları.



Şekil 9.2. Eklem açılarındaki değişim.



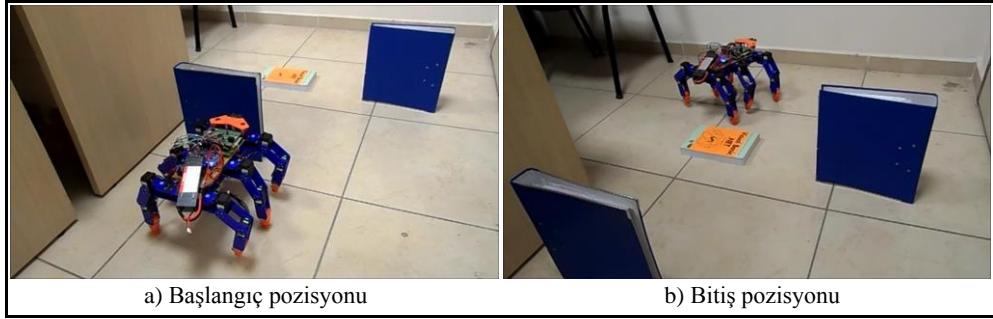
Şekil 9.3. Simülasyonun ürettiği sonuçlar ile FPGA donanımının ürettiği sonuçların karşılaştırılması.

Bu test sırasında, 56200-115600pps arasındaki değişik hızlarda denemeler yapılmıştır ve sonuçta elektronik donanımın, fonksiyonel olarak, doğru çalıştığı görülmüştür.

9.2. UYGULAMA ROBOTU ÜZERİNDE YAPILAN TESTLER

Tasarımı ve simülasyon sonuçları BÖLÜM 8’de verilen yürüme algoritmasının, robot üzerinde uygulanması başarılı bir şekilde test edilmiş ve sonuçlar burada incelenmiştir. Robot uzaktan yönlendirme ile ileri/geri gitme, sağa/sola dönme ve sağa/sola yan yürüme hareketlerini yapabilmekte ve birtakım engelleri aşarak yoluna devam edebilmektedir. Bu bölümde robotun hareketi sırasında çekilen videodan elde edilen kareler, hareketlerin ayrıntısını göstermesi amacıyla sunulmuştur.

Şekil 9.4’de robotun geçemeyeceği iki engel (mavi dosyalar) ve üzerinden geçebileceği, yeterince yüksek bir kitaptan oluşan parkur da robotun başlangıç ve bitiş pozisyonları gösterilmiştir. Robot aşamadığı iki engelin etrafından dolaşırken; ilk engelde sola yan yürüyüp engeli aştıktan sonra ileri gitmiştir. İkinci engeli aşmak için sağ tarafa yürümüş ve engeli aştıktan sonra ileri gidip kitabın üzerinden başarıyla geçerek parkuru tamamlamıştır. Şekil 9.5’de ise yalnızca ayak uçlarındaki kızılötesi sensörler tarafından görülebilecek iki engelden oluşan bir parkur vardır. Robot ilk engeli aşamayacağını anlayıp sağ tarafa yürümüştür, engeli geçtikten sonra düz gidip, engel olarak kullanılan kitabın üzerinden başarıyla geçmiştir.

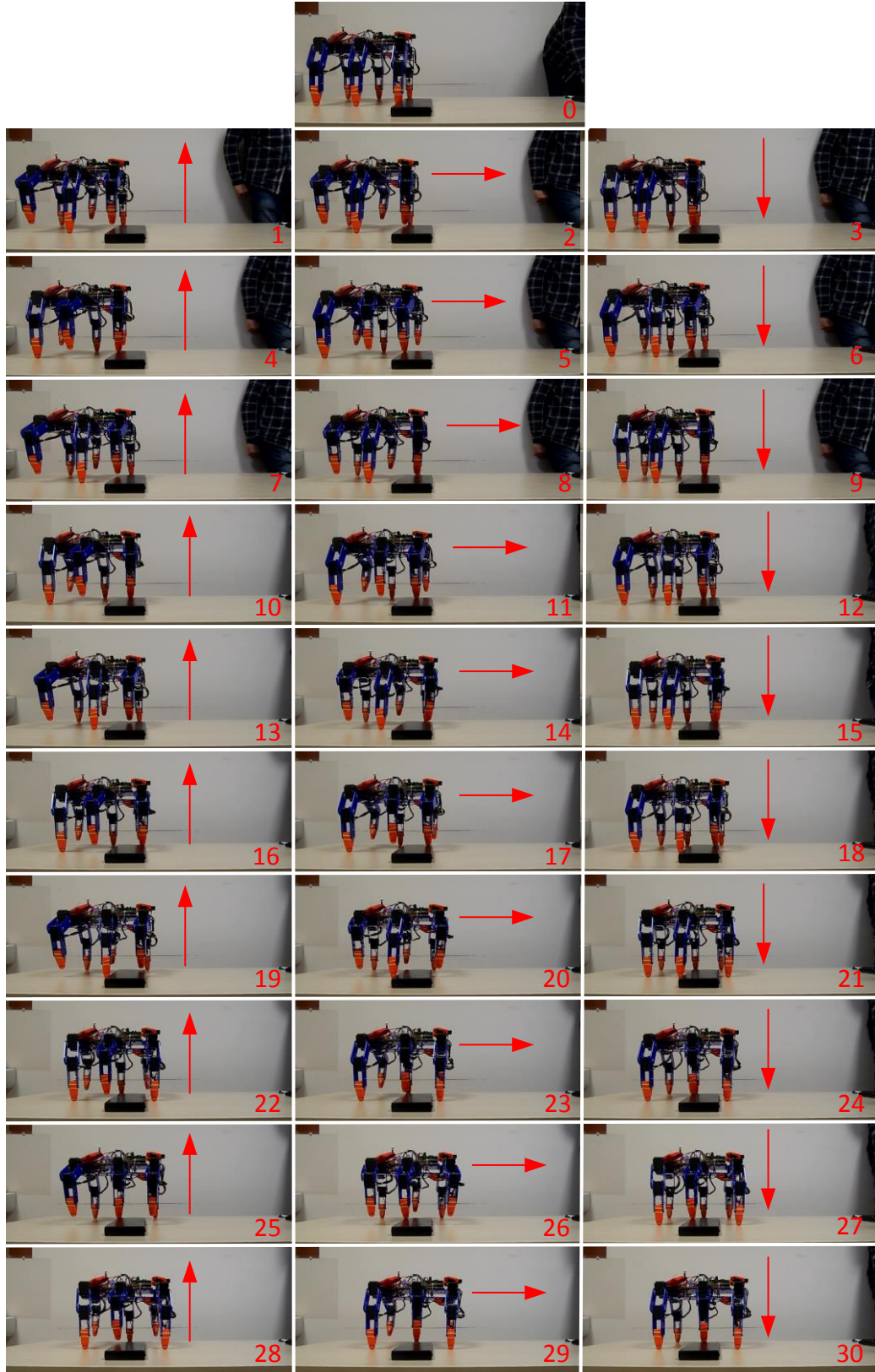


Şekil 9.4. Yüksek engelli parkur örneği.

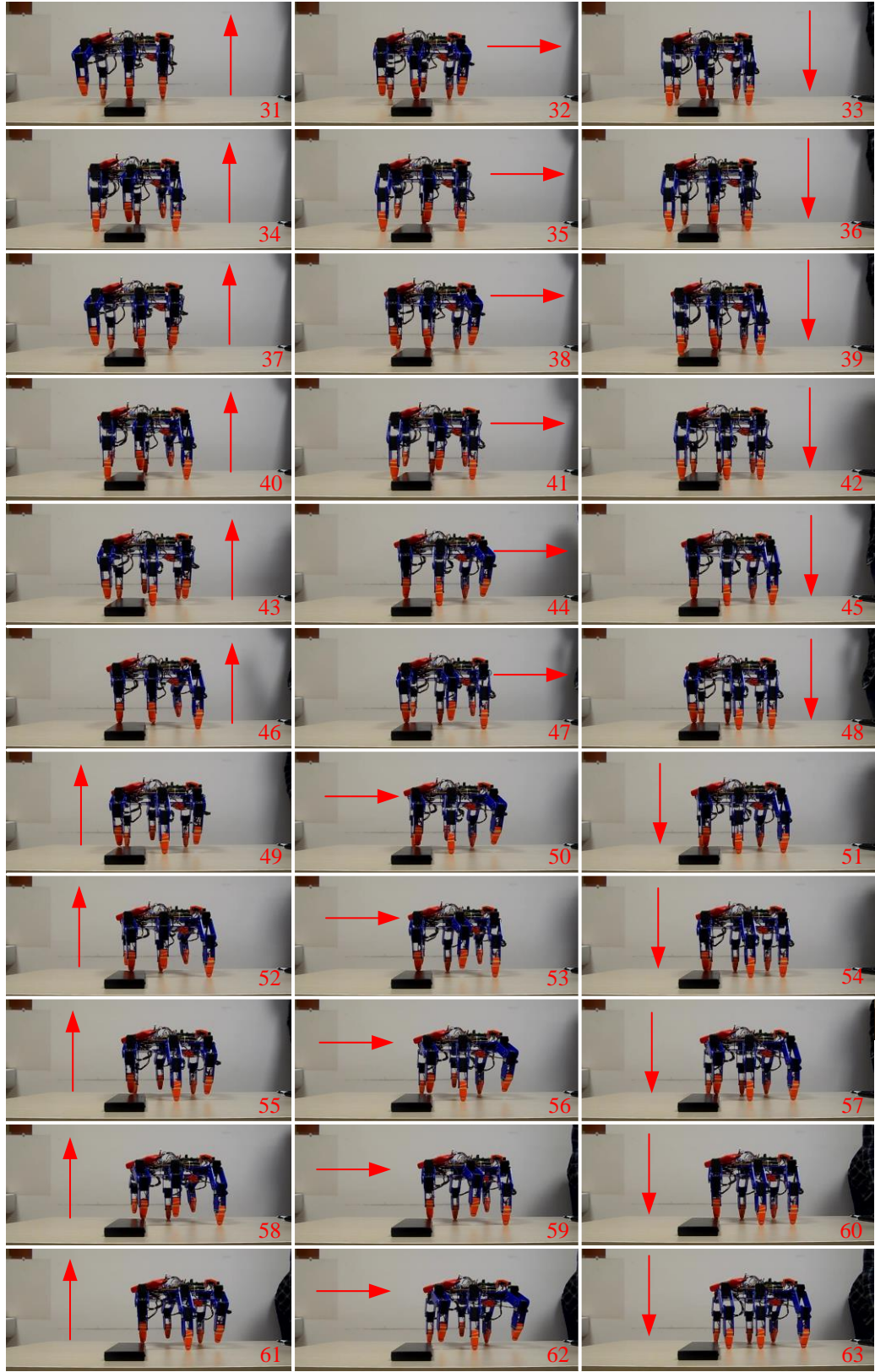


Şekil 9.5. Alçak engelli parkur örneği.

İleri yürüme sırasında ayakların sırayla bir engelin üzerine gelmesi durumunda yaptığı hareketleri gösteren kareler, ayrıntılı olarak, Şekil 9.6’da verilmiştir. Burada kareler robotun ayağını havaya kaldırmasıyla başlamış ve engeli aşmasıyla bitmiştir. Şekilde de görüldüğü üzere, robot yürüme işlemini başarı ile gerçekleştirmekte, üzerinden geçtiği engeli başarılı bir şekilde algılayarak gövdesini dengede tutmaktadır.



Şekil 9.6. Robotun ileri giderken engel aşması sırasında kaydedilen kareler.

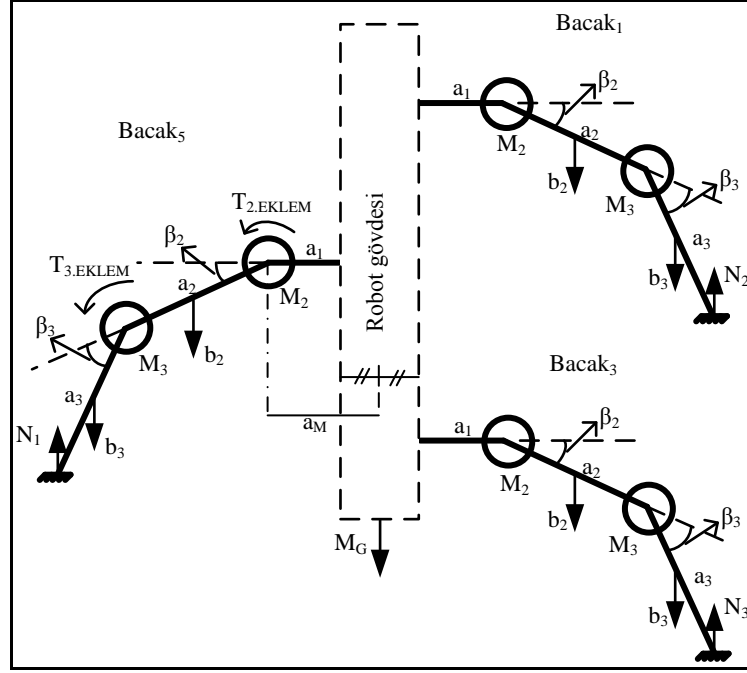


Şekil 9.6 (Devam ediyor).

9.3. PERFORMANS DEĞERLENDİRMESİ

Bu bölümde yapılan çalışma, mekanik ve elektriksel açıdan analiz edilmiştir. FPGA ile tasarlanan donanımın uygulanması için altı bacaklı bir robot tasarımı yapılmıştır. Bacaklı robotlarda bacak sayısı ne kadar fazla ise ortama adaptasyon yeteneği o kadar fazla olmaktadır ancak bacak sayısının artmasıyla birlikte sistemin kontrolü kaçınılmaz bir şekilde karmaşık bir hal almaktadır [28]. Ek olarak bacak sayısındaki artış ile robot maliyeti direkt olarak artırmaktadır. Tüm bacaklı robotlar içerisinde ortama adaptasyon ve kontrol sistemlerinin karmaşıklığı bir arada değerlendirildiğinde altı bacaklı robotlar mantıklı bir seçim olmaktadır [28]. Daha önce de belirtildiği gibi üç eklemlili bacak yapısı, robotun tüm yönlere yürüyebilmesi ve dar alanlarda yeterince esnek olabilmesi için gerekli minimum derecedir [31]. Dolayısı ile her bir bacağı üç eklemlili olan altı bacaklı bir robot ekonomi, adaptasyon yeteneği ve kontrol karmaşıklığı açısından en uygun seçenek olmaktadır.

Tasarlanan altı bacaklı robot için BÖLÜM 8’de anlatılan üç ayak yürüme biçimi seçilmiştir. Bu durumda robotun alabileceği temel iki pozisyon vardır; altı ayağında yere bastığı pozisyon ve üç ayağın yer bastığı pozisyon. Üç ayağın yere bastığı durumda robot ağırlığı bu üç ayak üzerine bineceği için, motorların diğer durumdan daha fazla tork üretmesini gerektirecektir. Robot düz bir zemin üzerinde dururken gövde üzerinde, z_1 eksenine yerleştirilen motora yük binmez. Çünkü bacaklar bu motorun hareket düzlemine diktir ve bacaklar gövdeye dik olan bu düzlemde meydana gelen hareketlerden ötürü, motor üzerine rotoru döndürmeye çalışan herhangi bir kuvvet oluşmaz. Bu nedenle bu pozisyonda Şekil 9.7’de görüldüğü gibi diğer iki eklem üzerine düşen kuvvetler incelenmiştir. Çizelge 9.1’de, Şekil 9.7’de verilen temsili çizim üzerindeki parçaların uzunlukları ve ağırlıkları verilmiştir. Görüldüğü gibi M_3 ve M_2 isimli motorların ağırlıkları ile a_3 ve a_2 isimli uzuvların ağırlık ve uzunlukları verilmiştir. Şekilde görülen üç bacakla ilgili bu değerler dışındaki tüm ağırlıklar gövde ağırlığına dâhil edilmiştir. Gövde ağırlık merkezinin tam ortada olduğu farz edilmektedir. Ayrıca gövde ağırlık merkezinin iki numaralı eklemlere uzaklığı a_M , gövde ağırlığı ise M_G ile gösterilmiştir.



Şekil 9.7. Robota üç ayak üzerinde etkiyen temel kuvvetler.

Çizelge 9.1. Robotun ölçüleri.

Parça	Uzunluk/Ağırlık
a_3/b_3	128mm/13gr
a_2/b_2	60mm/14gr
(a_M)	111,5mm
Motor ağırlığı (M1,M2,M3)	55gr
Robot + Pil ağırlığı	1790gr + 303gr=2093gr
M_G	1682gr

Burada $N_2=N_3$ dür. Ayrıca toplam ağırlık (7.4) denkleminde verildiği gibi tepki kuvvetleri toplamına eşittir. N_2 'yi bulmak için N_1 noktasına göre denge denklemini yazılırsa (7.5) denklemi elde edilir. N_2 'yi bulduktan sonra N_1 , toplam ağırlıktan faydalanılarak bulunur. N_1 ve N_2 bulduktan sonra eklemlere etkiyen torklar elde edilebilir.

$$\text{ToplamAğırlık} = N_1 + 2 * N_2 \quad (7.4)$$

$$\begin{aligned}
& 2.N_2.(2.a_3.\cos(\beta_3 + \beta_2) + 2.a_2.\cos(\beta_2) + 2.a_M) = \\
& M_3.(a_3.\cos(\beta_3 + \beta_2)) \\
& +M_2.(a_3.\cos(\beta_3 + \beta_2) + a_2.\cos(\beta_2)) \\
& +M_G.(a_3.\cos(\beta_3 + \beta_2) + a_2.\cos(\beta_2) + a_M) \\
& +2.M_2.(a_3.\cos(\beta_3 + \beta_2) + a_2.\cos(\beta_2) + 2.a_M) \\
& +2.M_3.(a_3.\cos(\beta_3 + \beta_2) + 2.a_2.\cos(\beta_2) + 2.a_M) \\
& +b_3.a_3.\cos(\beta_3 + \beta_2) / 2 + b_2.(a_2.\cos(\beta_2) / 2 + a_3.\cos(\beta_3 + \beta_2)) \\
& +2.b_2.(a_2.\cos(\beta_2) / 2 + 2.a_M + a_2.\cos(\beta_2) + a_3.\cos(\beta_3 + \beta_2)) \\
& +2.b_3.(a_3.\cos(\beta_3 + \beta_2) / 2 + 2.a_2.\cos(\beta_2) + 2.a_M + a_3.\cos(\beta_3 + \beta_2))
\end{aligned} \tag{7.5}$$

Bulduğumuz N_1 ve N_2 değerlerini kullanarak ikinci ve üçüncü eklemlere düşen torqlara bakacak olursak (7.6) ve (7.7) denklemleri elde edilir,

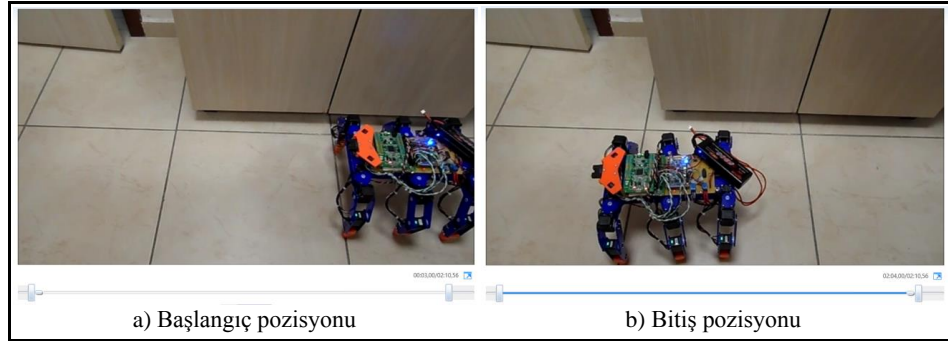
$$\begin{aligned}
T_{3.EKLEM} &= M_2.(a_2.\cos(\beta_2)) \\
& +M_G.(a_2.\cos(\beta_2) + a_M) + 2.M_2.(a_2.\cos(\beta_2) + 2.a_M) \\
& +2.M_3.(2.a_2.\cos(\beta_2) + 2.a_M) \\
& -2.N_2.(2.a_2.\cos(\beta_2) + 2.a_M + a_3.\cos(\beta_3 + \beta_2)) \\
& +b_2.a_2.\cos(\beta_2) / 2 \\
& +2.b_2.(a_2.\cos(\beta_2) / 2 + 2.a_M + a_2.\cos(\beta_2)) \\
& +2.b_3.(a_3.\cos(\beta_3 + \beta_2) / 2 + 2.a_2.\cos(\beta_2) + 2.a_M)
\end{aligned} \tag{7.6}$$

$$\begin{aligned}
T_{2.EKLEM} &= 2.N_2.(2.a_M + a_2.\cos(\beta_2) + a_3.\cos(\beta_3 + \beta_2)) \\
& -M_G.a_M - 2.M_2.2.a_M \\
& -2.M_3.(a_2.\cos(\beta_2) + 2.a_M) \\
& -2.b_2.(a_2.\cos(\beta_2) / 2 + 2.a_M) \\
& -2.b_3.(a_3.\cos(\beta_3 + \beta_2) / 2 + a_2.\cos(\beta_2) + 2.a_M)
\end{aligned} \tag{7.7}$$

Robotta kullanılan motorların maksimum tutma torku 12kg.cm'dir. Robot eklemlerine en çok yükün bineceği pozisyon eklemleri gövdeden en uzak olduğu pozisyonudur. Bu durumda açılar 0° iken gerçekleşir. Açılar 0° iken 2. ve 3. eklemlere etkiyen tork sırayla 15,53kg.cm ve 10,08kg.cm'dir. Bu pozisyonda, ikinci eklemden, motorların karşılayabileceği torkun üzerine çıkmaktadır ve dengede duramaz. Ancak robot normal yürüme ve duruş pozisyonunda ayaklar gövdeye yakın olacak ve torqlar azalacaktır. Bu çalışmada kullanılan algoritmada eklem açıları hiçbir zaman 0°

olmamaktadır. Robotun normal duruş pozisyonunda bu açılar $\beta_2=30^\circ$ ve $\beta_3=55^\circ$ olarak hesaplandığında eklemlere düşen yük $T_2=5,56\text{kg.cm}$ ve $T_3=0,874\text{kg.cm}$ olmaktadır. Örnek verilen açılar yürüme sırasında gerçekleşebilecek minimum açılardır ve daha büyük açılarda bacaklar gövdeye daha yakın duracağı için gerekli torklar bu değerlerden düşük olacaktır. Bu pozisyonda robotun taşıyabileceği en fazla yük 2,1kg olmaktadır. Bu yük ile 2. ekleme 12kg.cm, 3. ekleme 1,82kg.cm tork etki etmektedir.

Robotun yürüme hızı mekanik ve elektronik donanımın sınırlarına bağlıdır. Gerçek yürüme sırasında yapılan ölçümlerde robotun hızı engelsiz düz yolda 24,8cm/dk. olarak ölçülmüştür. Hız ölçümü video kamera ile bilinen iki nokta arasında yürüme zamanının elde edilmesi ile ölçülmüştür. Robotun ölçüm yapılan iki noktadan geçişi Şekil 9.8’da verilmiştir. İki nokta arasındaki mesafe 50cm’dir. Videodan elde edilen süre ise 121 sn’dir. Burada hız 24,8cm/dk. olarak hesaplanmaktadır.



Şekil 9.8. Hız testi sırasında alınan görüntüler.

Robotu güç tüketimi açısından inceleyecek olursak en yüksek güç sarfıyatı motorlar üzerinde olmaktadır. Tasarlanan sistem 8,4V gerilime sahip lityum-polimer batarya ile beslenmektedir. Yapılan ölçümler sonucunda sistemin akım değerleri elde edilmiştir. Robot altı bacağına üzerinde dururken çektiği ortalama akım 1.520A’dır. Hareket sırasında ise ortalama olarak 1.870A akım çekmektedir. Çekilen akımın maksimum değeri yaklaşık 2A olarak hesaplanırsa, robotun düz yolda yürümesi sırasında ihtiyaç duyacağı güç $2A \cdot 8,4V = 16,8W$ olmaktadır. Bu durumda robot 2500mAh’lik bir batarya ile 75dk çalışacaktır. Bu süre uygulamada eklemlere ön görülenden fazla yük düşmesi ve batarya kalitesi gibi nedenler ile daha kısa

olabilmektedir. Robot yürüme sırasında daima statik dengededir. Bu yüzden dururken ve yürürken ölçülen akımların farkı azdır. Dinamik hareket esnasında veya eklemleri gövdeden uzaklaştıran farklı pozisyonlarda yine robotun güç sarfiyatı artacak, pil ömrü kısılacaktır.

Uygulanabilecek alanlar açısından değerlendirildiğinde tasarlanan robotun kullanım alanlarının başında eğitim robotu olarak kullanım gelmektedir. Robot üzerinde çeşitli yapay zekâ algoritmaları, yürüme algoritmaları gibi konular çalışılabilir. Ayrıca robot üzerine kablosuz kamera entegre edilerek güvenlik amaçlı keşif robotu olarak da kullanılması mümkündür. Üzerine kamera ve mayın detektörü yerleştirilmesi ile mayın arama robotu olarak da kullanılabilir. Geliştirilen temel donanımın örneklere benzer başka alanlara da uyarlanması mümkündür. Ancak bu çalışmada üretilen prototip dış ortam uygulamaları için uygun değildir ve hızı yetersizdir.

BÖLÜM 10

SONUÇ VE ÖNERİLER

Bu tez çalışmasında çok eklemlili robotların ters kinematik denklemlerini çözmek üzere FPGA tabanlı bir donanım geliştirilmiş ve 18 serbestlik dereceli, altı bacaklı bir robot üzerinde uygulaması yapılmıştır. Geliştirilen bu donanım teknolojideki eğilimlere paralel olarak yaygınlaşan seri arabirimden kontrol edilebilen servo motorlar ile çalışacak şekilde özelleştirilmiştir. Seri hat üzerinden (mikrodenetleyici, bilgisayar, kablosuz alıcı, vb.) aldığı ayak pozisyonlarına ulaşmak için gerekli eklem açılarını hesaplayarak, yine seri hat üzerinden motorlara göndermektedir. Bu tasarımda aynı FPGA üzerinde seri iletişim modülleri ve kinematik hesaplar için yardımcı işlemci bulunmaktadır. Bu donanım sayesinde yoğun matematiksel işlemlerin getirdiği yük, merkezi işlem biriminin üzerinden alınmaktadır. Bu donanımın robotik uygulamalarda tasarımdan uygulamaya geçişi hızlandıracağı öngörülmektedir.

Seri hat üzerinden kumanda edilebilen servo motorlar kontrolcülerini kendi üzerlerinde bulundurmaktadır. Böylece kendini denetleyen merkezi işlem birimine ek bir yük oluşturmamaktadır. Dışarıdan sadece güç ve seri veri girişi ile çalışmaktadırlar, tek bir seri hat üzerine onlarca motor bağlanabilmektedir. Böylelikle merkezi kontrol birimi ve motorlar arasında yapılan elektriksel bağlantı sayısı azalmış olmakta, ayrıca seri hat sayesinde bütün motorlar için tek bir bağlantı yapılmaktadır. Böylece elektromekanik karmaşa azaltılarak, sistem fiziksel olarak basitleştirilmiş olmaktadır. Kullanılan motor sayısının fazla olduğu uygulamalarda hem tasarım hem de montaj açısından bu durum büyük avantaj sağlamaktadır.

Çalışmada bu donanımın uygulaması için bir robot tasarlanmıştır. Robot, her biri üç serbestlik derecesine sahip altı bacaklı bir robottur. Üzerinde, çevredeki cisimleri ve engelleri algılayabilmesi için, 6 adet kızılötesi sensör kullanılmıştır. Bunlardan 3

tanesi robotun kafasına, 1 tanesi gövdenin hemen altına yerleştirilen mesafe sensörleridir. Kafaya yerleştirilen sensörlerden biri ileri bakacak şekilde, diğer ikisi ise ortadaki sensör ile 45° açı yapacak şekilde yerleştirilmiştir. Böylece geniş bir bakış açısı elde edilmiştir. Gövdenin altına yerleştirilen sensör ise hem daha alçak nesne ve engellerin algılanmasını sağlamakta, hem de kafanın ortasına yerleştirilen sensörün kör bölgesinde algılama yaparak görüş alanını artırmaktadır. Diğer iki kızılötesi sensör ise, robotun ön iki ayak ucuna yerleştirilen, cisim algılama sensörleridir. Bu sensörler 10cm mesafe içerisindeki cisimleri algılayarak, mesafe sensörleri tarafından algılanamayan engelleri tespit etmekte ve bu sensörlerden alınan veriler doğrultusunda yürüme algoritması işletilmektedir.

Robotun bozuk zeminde ilerlemesi ve engellerin üzerinden geçerken gövde dengesini koruması için bastığı zemin hakkında bilgi sahibi olması gerekmektedir. Bu bilgiyi elde etmek için, robotun her bir ayağına basınç sensörü yerleştirilmiştir. Bu sensörler sayesinde ayağın yere basıp basmadığı ve zemine ne kadar kuvvet uyguladığı bilgileri elde edilmekte, ayağın zemine bastığı anadaki ayak pozisyonundan ise zemin üzerindeki yükselti belirlenebilmektedir. Bu bilgiler yürüme algoritması tarafından adım üretimi ve yön tayininde kullanılmaktadır.

Sistem üzerinde bir yürüme algoritması geliştirilmiştir. Bu algoritma ileri gitme, geri gitme, sağa dönme, sola dönme, sağa ve sola yan yürüme gibi temel hareketleri yaparken ayakların koordinasyonunu sağlamaktadır. Ayrıca bozuk zeminlerde ilerleyebilmesi ve belirli engelleri aşabilmesi için gerekli algoritmalar geliştirilmiş ve uygulanmıştır. Robot önüne çıkan engellerin büyüklüğünü değerlendirerek ne yapacağına karar verebilmektedir. Sensörler vasıtasıyla ölçümler yaparak cisim yüksekliğini belirlemekte ve üzerinden geçebilecekse, ayak yüksekliklerini ayarlayarak, cismin üzerinden geçmektedir. Üzerinden geçemeyeceğine karar verdiğinde ise etrafından dolaşmayı denemekte, bir çözüm bulamazsa kullanıcıdan yardım istemektedir.

Ayrıca robot üzerine yerleştirilen bir radyo modülü ile sensör verileri okunabilmekte, robota kumanda komutları gönderilebilmektedir. Bunun için bilgisayar üzerinde bir kumanda arayüzü geliştirilmiştir.

Çalışma geliştirmeye açık bir konudur. Elde edilen prototip robot 24,8cm/dk. ilerleme hızına sahiptir. Fiziksel olarak küçüktür ve dış ortam için uygun bir fiziksel yapıya sahip değildir. Sistem dinamikleri göz ardı edilerek tasarlanmıştır. Uzun süreli kullanım için enerji kaynağı sıkıntısı vardır.

Gelecekte yapılabilecek çalışmalardan biri, sistem dinamiklerini de hesaba katan yeni bir kontrol donanımının geliştirilmesidir. Böyle bir donanım koşmak, zıplamak gibi hareketleri gerçekleştirebilecek robotlar üzerinde uygulanabilir.

Sistem fiziksel olarak daha büyük boyutta ve mekanik açıdan daha dayanıklı olarak yeniden tasarlanabilir. Enerji sarfiyatı açısından da geleneksel enerji kaynaklarından (sıvı yakıtlar gibi) elektrik enerjisi üretilerek problem çözülebilir. Böylelikle daha hızlı, dayanıklı ve uzun çalışma sürelerine sahip bir robot gerçekleştirilebilir.

Robotun hızı ve manevra kabiliyetini geliştirecek yeni yürüme algoritmaları geliştirilebilir. Robot üzerine bir kamera entegre edilerek, görüntü işleme teknikleriyle, robotun algılama yeteneği artırılabilir. Otonom hareket için yapay zekâ algoritmaları ve birden fazla robot kullanıldığında grup davranışları gibi birçok konu sistem üzerinde çalışılabilir.

Sistem üzerine kamera ile birlikte denge sensörü, konum sensörü gibi birimler eklenerek, robotun izlenebilirliği artırılabilir, arazide kilometrelerce uzakta iken, rahatlıkla kontrolü gerçekleştirilebilir.

Böyle bir robot askeri uygulamalar, arama kurtarma uygulamaları ve bilimsel araştırmalar başta olmak üzere geniş bir uygulama alanına sahiptir.

KAYNAKLAR

1. Todman, T. J., Constantinides, G. A., Wilton, S. J. E., Mencer, O., Luk, W., and Cheung, P. Y. K., "Reconfigurable computing : Architectures and design methods", *IEE Proc.-Comput. Digit. Tech*, 152 (2): 193–207 (2005).
2. Compton, K. and Hauck, S., "Reconfigurable computing : A Survey of systems and software", *ACM Computing Surveys*, 34 (2): 171–210 (2002).
3. Shoa, A. and Shirani, S., "Run-time reconfigurable systems for digital signal processing applications : A survey", *Journal Of VLSI Signal Processing*, 39 (3): 213–235 (2005).
4. Hauck, S. and Dehon, A., "Reconfigurable computing", *Denise E. M. Penrose*, Boston, (2008).
5. Lee, C. A., Gasster, S. D., Plaza, A., Chang, C.-I., and Huang, B., "Recent developments in high performance computing for remote sensing : A review", *508 IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing*, 4 (3): 508–527 (2011).
6. Sklyarov, V. and Skliarova, I., "Hamming weight counters and comparators based on embedded dsp blocks for implementation in FPGA", *Advances In Electrical And Computer Engineering*, 14 (2): 63–68 (2014).
7. Vuillemin, J. E., Bertin, P., Roncin, D., Shand, M., Touati, H. H., and Boucard, P., "Programmable active memories : Reconfigurable systems come of age", *IEEE transactions on very large scale integration (VLSI) systems*, 4 (1): 56–69 (1996).
8. Monmasson, E. and Cirstea, M. N., "FPGA design methodology for industrial control systems - A review", *IEEE Transactions On Industrial Electronics*, 54 (4): 1824–1842 (2007).
9. Brown, S., "FPGA architectural research: A survey", *IEEE Design & Test Of Computers*, 13 (4): 9–15 (1996).
10. Tessier, R. and Burleson, W., "Reconfigurable computing for digital signal processing : A survey", *Journal Of VLSI Signal Processing*, 28 (1/2): 7–27 (2001).
11. Ovaska, S. J., Member, S., and Vainio, O., "Evolutionary-programming-based optimization of reduced-rank adaptive filters for reference generation in active

- power filters", *IEEE Transactions On Industrial Electronics*, 51 (4): 910–916 (2004).
12. Kathirvelan, J., Anilkumar, R., Alex, Z. C., and Fazul, A., "Development of low cost automatic wheelchair controlled by oral commands using standalone controlling system", *2012 IEEE International Conference On Computational Intelligence And Computing Research, ICCIC 2012*, 1–4 (2012).
 13. Yajima, S. and Katsura, S., "Multi-DOF motion reproduction using motion-copying system with velocity constraint", *IEEE Transactions On Industrial Electronics*, 61 (7): 3765–3775 (2014).
 14. Dhiman, R., Saini, J. S., and Priyanka, "Genetic algorithms tuned expert model for detection of epileptic seizures from EEG signatures", *Applied Soft Computing Journal*, 198–17 (2014).
 15. Tanaka, H., Member, S., Ohnishi, K., and Nishi, H., "Implementation of Bilateral control system based on acceleration control using fpga for multi-dof haptic endoscopic surgery robot", *IEEE Transactions On Industrial Electronics*, 56 (3): 618–627 (2009).
 16. Ugurlu, B. and Kawamura, A., "Bipedal trajectory generation based on combining inertial forces and intrinsic angular momentum rate changes", *IEEE Transactions On Robotics*, 28 (6): 1406–1415 (2012).
 17. Gabrick, M., Nicholson, R., Winters, F., and Young, B., "FPGA considerations for automotive applications", *Sae Technical Paper Series*, (724): (2006).
 18. Converter, B. B., "Electric vehicle battery life extension using ultracapacitors and an Fpga controlled interleaved buck–boost converter", *IEEE Transactions On Power Electronics*, 28 (12): 5940–5948 (2013).
 19. Chiaberge, B., Chiaberge, M., Botto, G., and Giuseppe, D., "An FPGA controlled dc/dc step-up converter for automotive applications", *EPE Journal*, 23 (3): 30–35 (2014).
 20. Ciani, L. and Catelani, M., "A fault tolerant architecture to avoid the effects of Single Event Upset (SEU) in avionics applications", *Measurement*, 54256–263 (2014).
 21. Hartley, E. N., Jerez, J. L., Member, S., Suardi, A., Maciejowski, J. M., Kerrigan, E. C., Constantinides, G. A., and Member, S., "Predictive control using an fpga with application to aircraft control", *IEEE Transactions On Cont*, 22 (3): 1006–1017 (2014).

22. Lui, H. Y., Lee, C. H., and Pate, R. H., "Power estimation and thermal budgeting methodology for FPGAs", *IEEE 2004 Custom Integrated Circuits Conference*, (2): 711–714 (2004).
23. Controllers, F. P. I. D., Chan, Y. F., Moallem, M., and Wang, W., "Design and implementation of modular", *IEEE Transactions On Industrial Electronics*, 54 (4): 1898–1906 (2007).
24. Velusamy, S., Huang, W., Lach, J., Stan, M., and Skadron, K., "Monitoring temperature in FPGA based SoCs", *Proceedings - IEEE International Conference On Computer Design: VLSI In Computers And Processors*, 2005634–637 (2005).
25. Petry Ligocki, N., Rettberg, a., Zanella, M., Hennig, a., and de Freitas Francisco, a. L., "Towards a modular communication system for FPGAs", *Second IEEE International Workshop On Electronic Design, Test And Applications*, 71–71 (2004).
26. Zhu, X., Jiang, R., Chen, Y., Hu, S., and Wang, D., "FPGA implementation of Kalman filter for neural ensemble decoding of rat's motor cortex", *Neurocomputing*, 74 (17): 2906–2913 (2011).
27. Ostler, P. S., Caffrey, M. P., Gibelyou, D. S., Graham, P. S., Morgan, K. S., Pratt, B. H., Quinn, H. M., and Wirthlin, M. J., "SRAM FPGA reliability analysis for harsh radiation environments", *IEEE Transactions On Nuclear Science*, 56 (6): 3519–3526 (2009).
28. Chen, J., Liu, Y., Zhao, J., Zhang, H., and Jin, H., "Biomimetic design and optimal swing of a hexapod robot leg", *Journal Of Bionic Engineering*, 11 (1): 26–35 (2014).
29. Moosavian, S. A. a. and Dabiri, A., "Dynamics and planning for stable motion of a hexapod robot", *IEEE/ASME International Conference On Advanced Intelligent Mechatronics, AIM*, 818–823 (2010).
30. Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H., "Walknet, a bio-inspired controller for hexapod walking", *Biological Cybernetics*, 107 (4): 397–419 (2013).
31. Klaassen, B., Linnemann, R., Spennberg, D., and Kirchner, F., "Biomimetic walking robot SCORPION: Control and modeling", *Robotics And Autonomous Systems*, 41 (2): 69–76 (2002).
32. Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., "Robotics", soft computing, *Springer*, (2009).
33. Zeng, Q., Ehmann, K. F., and Cao, J., "Tri-pyramid robot: Design and kinematic analysis of a 3-DOF translational parallel manipulator", *Robotics And Computer-Integrated Manufacturing*, 30 (6): 648–657 (2014).

34. Song, Y., Gao, H., Sun, T., Dong, G., Lian, B., and Qi, Y., "Kinematic analysis and optimal design of a novel 1T3R parallel manipulator with an articulated travelling plate", *Robotics And Computer-Integrated Manufacturing*, 30 (5): 508–516 (2014).
35. Juang, Y. S., Sung, T. Y., Ko, L. T., and Li, C. L., "FPGA implementation of a CORDIC-based joint angle processor for a climbing robot", *International Journal Of Advanced Robotic Systems*, 10 (195): 1 (2013).
36. Huang, S. J. and Wu, S. S., "Vision-based robotic motion control for non-autonomous environment", *Journal Of Intelligent And Robotic Systems: Theory And Applications*, 54 (5): 733–754 (2009).
37. Yu, Y.-H., Kwok, N. M., and Ha, Q. P., "Color tracking for multiple robot control using a system-on-programmable-chip", *Automation In Construction*, 20 (6): 669–676 (2011).
38. Paul, J., Laika, A., Claus, C., Stechele, W., El Sayed Auf, A., and Maehle, E., "Real-time motion detection based on SW/HW-codesign for walking rescue robots", *Journal Of Real-Time Image Processing*, 8 (4): 353–368 (2013).
39. Wang, M.-S., Mi, C.-M., Wu, S.-Y., and Tu, Y.-M., "DSP and FPGA based stair climbing robot design", *Mathematical Problems In Engineering*, 2013: 1–9 (2013).
40. Tsai, C. C., Huang, H. C., and Lin, S. C., "FPGA-based parallel DNA algorithm for optimal configurations of an omnidirectional mobile service robot performing fire extinguishment", *IEEE Transactions On Industrial Electronics*, 58 (3): 1016–1026 (2011).
41. Vachhani, L., Sridharan, K., and Meher, P. K., "Efficient FPGA realization of cordic with application to robotic exploration", *IEEE Transactions On Industrial Electronics*, 56 (12): 4915–4929 (2009).
42. Huang, H.-C., "Intelligent motion control for omnidirectional mobile robots using parallel ACO algorithm", *Advances In Mechanical Engineering*, (June): 131–136 (2011).
43. Huang, H.-C., "Intelligent motion control for omnidirectional mobile robots using ant colony optimization", *Applied Artificial Intelligence*, 27 (3): 151–169 (2013).
44. Baturone, I., Gersnoviez, A., and Barriga, Á., "Neuro-fuzzy techniques to optimize an FPGA embedded controller for robot navigation", *Applied Soft Computing*, 2195–106 (2014).
45. Huang, H., "FPGA-based hybrid GA-PSO algorithm and its application to global path planning for mobile robots", *Przeglad Elektrotechniczny*, 88 (7B): 281–284 (2012).

46. Mingas, G., Tsardoulis, E., and Petrou, L., "An FPGA implementation of the SMG-SLAM algorithm", *Microprocessors And Microsystems*, 36 (3): 190–204 (2012).
47. Huang, C. H., Wang, W. J., and Chiu, C. H., "Design and implementation of fuzzy control on a two-wheel inverted pendulum", *IEEE Transactions On Industrial Electronics*, 58 (7): 2988–3001 (2011).
48. Wong, C. C., Lin, Y. H., Lee, S. A., and Tsai, C. H., "GA-based fuzzy system design in FPGA for an omni-directional mobile robot", *Journal Of Intelligent And Robotic Systems: Theory And Applications*, 44 (4): 327–347 (2005).
49. Ji, S., Hu, T., Zhang, C., and Sun, S., "A parametric hardware fine acceleration/deceleration algorithm and its implementation", *International Journal Of Advanced Manufacturing Technology*, 63 (9):–(12): 1109–1115 (2012).
50. Osornio-Rios, R. A., Romero-Troncoso, R. D. J., Herrera-Ruiz, G., and Castañeda-Miranda, R., "The application of reconfigurable logic to high speed CNC milling machines controllers", *Control Engineering Practice*, 16 (6): 674–684 (2008).
51. Rodriguez-Donate, C., Osornio-Rios, R. A., Rivera-Guillen, J. R., and Romero-Troncoso, R. de J., "Fused smart sensor network for multi-axis forward kinematics estimation in industrial robots", *Sensors (Switzerland)*, 11 (4): 4335–4357 (2011).
52. Cho, J. U., Le, Q. N., and Jeon, J. W., "An FPGA-based multiple-axis motion control chip", *IEEE Transactions On Industrial Electronics*, 56 (3): 856–870 (2009).
53. Koca, H., Doğan, M., and Taplamacıoğlu, M. C., "Cartesian-specific control of a robot manipulator", *Gazi Üniv. Müh. Mim. Fak. Der.*, 23 (4): 769–776 (2008).
54. Kim, B.-Y. and Ahn, H.-S., "A design of bilateral teleoperation systems using composite adaptive controller", *Control Engineering Practice*, 21 (12): 1641–1652 (2013).
55. Zhu, W. H., Lamarche, T., Dupuis, E., Jameux, D., Barnard, P., and Liu, G., "Precision control of modular robot manipulators: The VDC approach with embedded fpga", *IEEE Transactions On Robotics*, 29 (5): 1162–1179 (2013).
56. Chou, H. H., Kung, Y. S., Tsui, T. W., and Cheng, S., "FPGA-based motion controller for wafer-handling robot", *Transactions Of The Canadian Society For Mechanical Engineering*, 37 (3): 427–437 (2013).
57. Martinez-Prado, M., Franco-Gasca, A., Herrera-Ruiz, G., and Soto-Dorantes, O., "Multi-axis motion controller for robotic applications implemented on an

- FPGA", *The International Journal Of Advanced Manufacturing Technology*, 67 (9):-(12): 2367–2376 (2013).
58. Shao, X. and Sun, D., "Development of a new robot controller architecture with FPGA-based IC design for improved high-speed performance", *IEEE Transactions On Industrial Informatics*, 3 (4): 312–321 (2007).
 59. Huang, J. B., Xie, Z. W., Liu, H., Sun, K., Liu, Y. C., and Jiang, Z. N., "DSP/FPGA-based controller architecture for flexible joint robot with enhanced impedance performance", *Journal Of Intelligent And Robotic Systems: Theory And Applications*, 53 (3): 247–261 (2008).
 60. Huang, S. J. and Huang, J. C., "Vision guided dual arms robotic system with DSP and FPGA integrated system structure", *Journal Of Mechanical Science And Technology*, 25 (8): 2067–2076 (2011).
 61. Yen-Fang Li and Li-Lan Chuang, "Controller design for music playing robot - Applied to the anthropomorphic piano robot", *2013 IEEE 10th International Conference On Power Electronics And Drive Systems (PEDS)*, 968–973 (2013).
 62. Rahman, M. H., Saad, M., Kenné, J. P., and Archambault, P. S., "Control of an exoskeleton robot arm with sliding mode exponential reaching law", *International Journal Of Control, Automation And Systems*, 11 (1): 92–104 (2013).
 63. Arena, P., Patané, L., Schilling, M., and Schmitz, J., "Walking capabilities of Gregor controlled through Walknet", *Proceedings of SPIE*, 65920K–65920K–9 (2007).
 64. Erden, M. S., "Six-Legged Walking Machine", Doktora Tezi, *Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, (2006).
 65. Kim, H. W. and Lee, S., "Design and implementation of a private and public key crypto processor and its application to a security system", *IEEE Transactions On Consumer Electronics*, 50 (1): 214–224 (2004).
 66. Nanopositioner, H., Kenton, B. J., and Leang, K. K., " Design and control of a three-axis serial-kinematic high-bandwidth nanopositioner ", *IEEE/ASME Transactions on Mechatronics*, 17 (2): 356–369 (2012).
 67. Wong, C. C. and Liu, C. C., "FPGA realisation of inverse kinematics for biped robot based on CORDIC", *Electronics Letters*, 49 (5): 332–334 (2013).
 68. Franchini, S., Gentile, A., Sorbello, F., Vassallo, G., and Vitabile, S., "Design and implementation of an embedded coprocessor with native support for 5D, quadruple-based clifford algebra", *IEEE Transactions On Computers*, 62 (12): 2366–2381 (2013).

69. Candini, G. P., Paolini, E., and Piergentili, F., "Design and manufacture of a low cost educational hexapod rover", *Acta Astronautica*, 65 (3):-(4): 525–536 (2009).
70. Pa, P. S. and Wu, C. M., "Design of a hexapod robot with a servo control and a man-machine interface", *Robotics And Computer-Integrated Manufacturing*, 28 (3): 351–358 (2012).
71. Agheli, M., Qu, L., and Nestinger, S. S., "SHeRo: Scalable hexapod robot for maintenance, repair, and operations", *Robotics And Computer-Integrated Manufacturing*, 30 (5): 478–488 (2014).
72. Petko, M., Gac, K., Karpiel, G., and Góra, G., "Acceleration of parallel robot kinematic calculations in FPGA", *Proceedings Of The IEEE International Conference On Industrial Technology*, 34–39 (2013).
73. Ghorbel, A., Jallouli, M., Ben Amor, N., and Amouri, L., "An FPGA based platform for real time robot localization", *2013 International Conference On Individual And Collective Behaviors In Robotics (ICBR)*, 56–61 (2013).
74. Gac, K., Karpiel, G., and Petko, M., "FPGA based hardware accelerator for calculations of the parallel robot inverse kinematics", *IEEE International Conference On Emerging Technologies And Factory Automation, ETFA*, 1–4 (2012).
75. Ding, B., Stanley, R. M., Cazzolato, B. S., and Costi, J. J., "Real-time FPGA control of a hexapod robot for 6-DOF biomechanical testing", *IECON Proceedings (Industrial Electronics Conference)*, 252–257 (2011).
76. Lee, B.-H., "Design of a reconfigurable humanoid robot", *RA '07 Proceedings Of The 13th IASTED International Conference On Robotics And Applications*, 425–430 (2007).
77. Sánchez, D. F., Muñoz, D. M., Llanos, C. H., and Motta, J. M., "FPGA implementation for direct kinematics of a spherical robot manipulator", *ReConFig'09 - 2009 International Conference On ReConfigurable Computing And FPGAs*, 416–421 (2009).
78. Erkol, H. O. and Demirel, H., "A VHDL application for kinematic equation solutions of multi-degree-of-freedom systems", *Journal Of Zhejiang University SCIENCE C*, 15 (12): 1164–1173 (2014).
79. Murray, R. M., "A mathematical introduction to robotic manipulation", *CRC Press*, (1994).
80. Yılmaz, D., "Bir robot kolunun bilgisayar destekli kinematik analizi", Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, (2010).

81. İnternet: Dongbu Robot, "Herkulex User Manual", <http://www.sgbotic.com/products/datasheets/robotics/herkulexeng.pdf> (2015).
82. İnternet: Altera Corporation, "Altera De0-Nano User Manual", <http://www.terasic.com.tw/> (2015).
83. İnternet: STMicroelectronics, "UM1472 user manual", <http://www.st.com/> (2015).
84. İnternet: Digi Corporation, "XBee radio module", <http://www.digi.com/> (2015).
85. İnternet: Interlink Electronics, "FSR402 force sensor", <http://www.interlinkelectronics.com/FSR402.php> (2015).
86. Zheng, Y., Liu, J., and Kan, J., "An optimal kinematics calculation method for a multi-DOF manipulator", *Przegląd Elektrotechniczny*, 88 (7B): 320–323 (2012).
87. Taira, T., Kamata, N., and Yamasaki, N., "Design and implementation of reconfigurable modular humanoid robot architecture", *2005 IEEE/RSJ International Conference On Intelligent Robots And Systems, IROS*, 1071–1076 (2005).
88. Erkol, H. O. and Demirel, H., "FPGA ile Seri Port Tasarımı ve Uygulaması", *Otomatik Kontrol Ulusal Toplantısı, TOK2013*, 132–135 (2013).
89. Axelson, J., "Serial port complete", 2. Ed., *Lakeview Research*, (2007).

ÖZGEÇMİŞ

Hüseyin Oktay ERKOL 1983 yılında Kırıkkale’de doğdu. İlk ve orta öğretimine birkaç farklı şehirde devam ettikten sonra, Ankara Abidin Paşa Lisesinden, 1999 yılında mezun oldu. Üniversite hayatına 2000 yılında Konya’da başladı. Selçuk Üniversitesinde İnşaat Mühendisliği bölümünde İngilizce hazırlık ve lisansın ilk yılını okuduktan sonra, bölüm ve üniversite değişikliği yaparak Niğde Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümünü kazandı. Yatay geçişle eğitimine Kırıkkale Üniversitesinde devam ederek 2006 yılında mezun oldu. Aynı yıl TÜBİTAK bursu ile Erciyes Üniversitesi, Mekatronik Mühendisliği Bölümünde yüksek lisansını yaptı. 2009 yılında askerlik hizmetini tamamladıktan sonra özel sektörde ar-ge ve proje mühendisi pozisyonlarında bir süre çalıştı. Bu sırada Kırıkkale Üniversitesinde doktora programına başladı. 2011 yılında özel sektörü bırakarak, ÖYP programı dahilinde, Karabük Üniversitesi, Teknoloji Fakültesi, Mekatronik Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaya başladı. Doktora eğitimini de Karabük Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliğinde tamamladı.

ADRE BİLGİLERİ

Adres : Karabük Üniversitesi
Teknoloji Fak. Mekatronik Müh. Böl.
Demir Çelik Kampüsü 100.Yıl / KARABÜK

Tel : (0554) 334 9104

E-posta : oktayerkol@karabuk.edu.tr
hoktayerkol@hotmail.com