

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**MERDİVEN ÇIKIP İNEBİLEN İKİ AYAKLI ROBOTUN DİNAMİK
MODELLENMESİ VE DENETİMİ**

DOKTORA TEZİ

İsmail Hakkı ŞANLITÜRK

Makina Mühendisliği Anabilim Dalı

Makina Mühendisliği Programı

TEMMUZ 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**MERDİVEN ÇIKIP İNEBİLEN İKİ AYAKLI ROBOTUN DİNAMİK
MODELLENMESİ VE DENETİMİ**

DOKTORA TEZİ

**İsmail Hakkı ŞANLITÜRK
(503122018)**

Makina Mühendisliği Anabilim Dalı

Makina Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hikmet KOCABAŞ

TEMMUZ 2024

İSTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**DYNAMIC MODELING AND CONTROL OF A BIPEDAL ROBOT THAT CAN
CLIMB AND DESCEND STAIRS**

Ph.D. THESIS

**İsmail Hakkı ŞANLITÜRK
(503122018)**

Department of Mechanical Engineering

Mechanical Engineering Programme

Thesis Advisor: Prof. Dr. Hikmet KOCABAŞ

JULY 2024

İTÜ, Lisansüstü Eğitim Enstitüsü'nün 503122018 numaralı Doktora Öğrencisi İsmail Hakkı ŞANLITÜRK, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “MERDİVEN ÇIKIP İNEBİLEN İKİ AYAKLI ROBOTUN DİNAMİK MODELLENMESİ VE DENETİMİ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Hikmet KOCABAŞ**

İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Mehmet Salih DOKUZ**

İstanbul Teknik Üniversitesi

Doç. Dr. Hakan YAZICI

Yıldız Teknik Üniversitesi

Doç. Dr. Zeki Yağız BAYRAKTAROĞLU.....

İstanbul Teknik Üniversitesi

Prof. Dr. Semih SEZER

Yıldız Teknik Üniversitesi

Teslim Tarihi : **31 Mayıs 2024**

Savunma Tarihi : **24 Temmuz 2024**





Eşime ve Kutay'a,



ÖNSÖZ

İnsansı robotlar geçmiş çağlardan beridir insanların ilgisini çekmiştir. Bu ilgi o kadar yüksektir ki bilim insanları çağlarının imkanlarını sonuna kadar kullanıp işlevsel insansı robotlar üretmeye çalışmışlardır. Günümüz teknolojisi gelişinceye kadar geliştirilen insansı robotlar daha çok sanat eseri olarak değerlendirilebilir. Teknik imkanların gelişmesi ve maliyetlerin katlanılabilir seviyeye gelmesiyle günümüzde fabrika ortamlarında çalışan robotlar artık bir gerçektir.

İnsansı robotlara yüksek ilgili bir mühendis olarak bendeniz de konuyla ilgili bir çalışma yapmış bulunuyorum. Her ne kadar kapakta ismim yazıyor olsa da, çok değerli insanların mevcut ve önceki katkıları olmasa bu çalışma tamamlanamazdı.

Öncelikle danışman hocam Prof. Dr. Hikmet KOCABAŞ'a danışmanım olmayı kabul etmesinden dolayı çok teşekkür ederim. Sunduğu imkanlardan, kendisinin mesleki değerlerinden faydalanma hakkına sahip olabildim.

Tez ilerleme jürimdeki hocalarım İstanbul Teknik Üniversitesi'nden Prof Dr. Mehmet Salih DOKUZ ve Yıldız Teknik Üniversitesi'nden Doç. Dr. Hakan YAZICI'ya öğrenciliğim boyunca çalışmama yaptıkları katkılardan dolayı teşekkür ederim.

Doç. Dr. Zeki BAYRAKTAROĞLU'na İTÜ Biped üzerinde çalışma izni vermesinden ve çalışma sürem boyunca yardımlarından ötürü teşekkür ederim.

Katkılarından faydalandığım arkadaşlarım Arş. Gör. Dr. Sertaç Emre KARA, Dr. Öğr. Üy. Murat ŞEN, Dr. Öğr. Üy. Tayfun ABUT, Arş. Gör. Dr. Celal KISTAK ve Araş. Gör. Mithat Can Özın'e teşekkür ederim.

Elektronik donanım ve kablolamada yardımlarını esirgemeyen Elektrik ve Elektronik öğretmeni Tuncay KANDEMİR'e teşekkür ederim.

Mayıs 2024

İsmail Hakkı ŞANLITÜRK

Makine Yüksek Mühendisi



İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ	ix
İÇİNDEKİLER	xi
KISALTMALAR	xiii
SEMBOLLER	xv
ÇİZELGE LİSTESİ.....	xvii
ŞEKİL LİSTESİ.....	xix
ÖZET	xxi
SUMMARY	xxv
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.2 Temel Kavramlar.....	1
1.3 Literatür özeti	3
1.4 Hipotez	7
1.5 Problem Tanımı ve Bilimsel Katkıları	8
2. YÖNTEM.....	11
2.1 Robotun Yapısı.....	11
2.2 Kinematik	13
2.3 Robot Dinamiği	15
2.4 Kütle merkezi ve ZMP Hesabı	18
2.5 Kütle merkezi Hesap Yöntemi	19
2.6 Dinamik Model	21
2.7 Denetim Sistemi	22
3. BENZETİMLER.....	25
3.1 Açık Çevrim Olarak Düzlemde Yürüyüş	25
3.2 Açık Çevrim Olarak Merdiven Çıkıp İnme.....	28
3.3 Kapalı Çevrim Düzlemde Yürüyüş	32
3.4 YKM Hesap Katsayılarının Etkisi.....	35
4. İTÜ BİPED ÜZERİNDE YAPILAN ÇALIŞMALAR.....	37
4.1 Robotun Durumu.....	37
4.2 Yazılım Güncellemeleri	38
4.3 Mekanik Tasarım Güncellemesi.....	42
4.4 Güncelleme Sonuçları	45
5. YENİ ROBOTUN TASARIM VE ÖZELLİKLERİ.....	47
5.1 Mekanik Tasarım.....	47
5.1.1 Mekanik sınırlar	50
5.2 Elektronik Donanım	52
5.3 Yazılım	53
6. SONUÇLAR	59
KAYNAKLAR	61
EKLER	67
EK A: Robot Yörüngesi Dönüştürme Programı.....	69
ÖZGEÇMİŞ.....	75



KISALTMALAR

2D	: 2 Dimensional – 2 Boyutlu
3D	: 3 Dimensional – 3 Boyutlu
3D-LIP	: 3 Dimensional Linear Inverted Pendulum- 3 boyutlu doğrusal ters sarkaç
AFC	: Adaptive Fuzzy Controller - Uyarlamalı Bulanık Denetleyici
CoM	: Center of Mass – Kütle merkezi
CNS	: Central Neural System – Merkezi Sinir Sistemi
CP	: Center of Pressure – Basınç Merkezi
CPG	: Central Pattern Generator – Merkezi Örüntü Oluşturucu
CWS	: Contact Wrench Sum - Temas Kuvvet Kolu Toplamı
DH	: Denavit-Hartenberg
DSP	: Double Support Phase – Çift destek fazı
FLC	: Fuzzy Logic Controller – Bulanık Mantık Denetleyici
FLS	: Fuzzy Logic System - Bulanık Mantık Sistemi
GDC	: Generalized Direction Curve - Genelleştirilmiş Doğrultu Eğrisi
IMU	: Inertial Measurement Unit – Atalet Ölçüm Birimi
İTÜ	: İstanbul Teknik Üniversitesi
LIP	: Linear Inverted Pendulum – Doğrusal Ters Sarkaç
LIPFM	: Linear Inverted Pendulum with Flywheel Model - Kasnaklı ters sarkaç modeli
MPC	: Model Predictive Control – Model Ön Görümlü Denetim
PDO	: Process Data Object (İşlem Veri Nesnesi)
SD	: Serbestlik Derecesi
SSP	: Single Support Phase – Tek destek fazı
YKM	: Yakınsayan Kütle Merkezi Farkı
ZMP	: Zero Moment Point (Sıfır Moment Noktası - SMN)



SEMBOLLER

O	: Koordinat sistemi orijini
x_0, y_0, z_0	: Robot kalça uzunluk parametreleri
l_1, l_2, l_3	: Robot uzuv uzunluk parametreleri
p_x, p_y, p_z	: ZMP konumu
c_x, c_y, c_z	: Kütle merkezi konumları
$\mathbf{c}$	: Kütle merkezi vektörü
s_x, s_y, s_z	: Adım parametreleri
h_{bas}	: Basamak yüksekliği
$h_{adım}$	: Adım yüksekliği
θ	: Eklem açısı
T	: Link dönüşüm matrisi
k_x, k_y	: Ters sarkaçın hareket ettiği yüzey eğimi
G	: Hata vektörü



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1: Uzunluk parametreleri.....	12
Çizelge 2.2: Kütle merkezi konumları ve kütleler.....	13
Çizelge 2.3: Denavit-Hertenberg yöntemine göre link dönüşüm tablosu.....	13
Çizelge 2.4: Merdiven ve düz yüzeyler için yörünge oluşturma programı öz kodu..	17
Çizelge 2.5: Yakınsayan Kütle Merkezi (YKM) algoritması.....	20
Çizelge 2.6: Çalışmada kullanılan parametreler.....	22
Çizelge 5.1: Uzunlukların hareket sınırları.....	50
Çizelge 5.2: Denavit-Hertenberg yöntemine göre link dönüşüm matrisi.....	52
Çizelge 6.1 Literatürde verilen kütle merkezi takip hataları.....	60



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1: Yürümenin sınıflandırılması	2
Şekil 2.1: Robot görünüşleri. (a) yan, (b) ön ve arkadan görünüşü.....	11
Şekil 2.2: Robot eklem açıları ve temel boyutlar. (a) eklem ve uzunluk parametreleri; (b) bacak indisleri ve yaklaşık kütle merkezi konumları.	12
Şekil 2.3: 3 Boyutlu Lineer ters sarkaç modeli (3BLTSM).	16
Şekil 2.4: Merdiven çıkıp inmede kullanılacak parametreler.	16
Şekil 2.5: Düzlemde yürüme için kütle merkezi yörüngesi ve sabit ayak konumları.	17
Şekil 2.6: Merdiven çıkıp inme için kütle merkezi yörüngesi ve sabit ayak konumları.	18
Şekil 2.7: ZMP hesabında kullanılan parametreler.	18
Şekil 2.8: Robot ve kütle merkezinin grafik gösterimi.....	21
Şekil 2.9: Simulink model diyagramı.	21
Şekil 2.10: Ayak uç noktalarına yerleştirilmiş temas küreleri.....	22
Şekil 2.11: YKM gerçek zamanlı kapalı çevrim denetim sistemi.	23
Şekil 2.12: Kapalı çevrim sistem için referans kütle merkezi yörüngesi.....	23
Şekil 3.1: Simülasyon adımları : YKM’li(a), YKM’siz(b).....	25
Şekil 3.2: Hesaplanan ve simüle edilen yörüngeler.....	26
Şekil 3.3: YKM’li ZMP ve kütle merkezi yörüngesi.	27
Şekil 3.4: YKM’siz ZMP ve kütle merkezi yörüngesi.	27
Şekil 3.5: YKM algoritmasının kütle merkezi takip hatası.	28
Şekil 3.6: Merdiven çıkıp inme simülasyon ortamı t=0 s.	29
Şekil 3.7: Merdiven çıkıp inme yürüyüş pozları.....	30
Şekil 3.8: Hesaplanan yörünge (noktalar), simülasyonda oluşan yörünge (çizgi). Yandan görünüş.	31
Şekil 3.9: Hesaplanan yörünge (noktalar), simülasyonda oluşan yörünge (çizgi). Üstten görünüş.	31
Şekil 3.10: İlerleme hızı.....	32
Şekil 3.11: Ölçülen kütle merkezi konumları.	33
Şekil 3.12: Kütle merkezi takip hatası.....	33
Şekil 3.13: Sağ bacak eklem açı değerleri.	33
Şekil 3.14: Sol bacak eklem açı değerleri.....	34
Şekil 3.15: Sağ kalça konum değişimi.....	34
Şekil 3.16: Sol kalça konum değişimi	34
Şekil 3.17: Ağırlık merkezi hatasının etkisi	35
Şekil 3.18: Kalça hareket miktarının etkisi	36
Şekil 4.1: Parçalı üçüncü dereceden eklem yörüngesi.....	39
Şekil 4.2: Parçalı üçüncü eklem yörüngesi, sürekli hale getirilmiş hız geçişleri.	41
Şekil 4.3: Tasarlanan gövde (a), alt ve üst levhalar hariç monte edilmiş gövde (b)..	42
Şekil 4.4: Gövde ek parça detayı.	42

Şekil 4.5: Mevcut tasarım (a), güncellenmiş tasarım(b).....	43
Şekil 4.6: Güncel ayak tasarımı.	43
Şekil 4.7: Statik analiz sınır şartları	44
Şekil 4.8: Şekil değiştirme sonuçları(a), emniyet katsayısı(b).	44
Şekil 5.1: Robotun üç görünüşü ve eklemlerdeki eksen takımlarının yerleşimi.	48
Şekil 5.2: Ayak gövde mesneti montaj kesiti.	49
Şekil 5.3: Tek parça gövde.	49
Şekil 5.4: 4 noktadan temaslı ayak tasarımı.	49
Şekil 5.5: Ön görünüş kalça yalpalama hareket sınırları üst sınır(a); alt sınır(b).	50
Şekil 5.6: Sol görünüş kalça yunuslama hareket sınırları üst sınır(a); alt sınır(b).....	50
Şekil 5.7: Sol görünüş diz hareket sınırları üst sınır(a); alt sınır(b).....	51
Şekil 5.8: Sol görünüş bilek yunuslama hareket sınırları üst sınır(a); alt sınır(b).	51
Şekil 5.9: Arka görünüş bilek yalpalama hareket sınırları üst sınır(a); alt sınır(b). ..	52
Şekil 5.10: Elektronik akış şeması.....	53
Şekil 5.11: Sol ayak için yatay eksen potansiyometre, dikey eksen motor konumu grafikleri.....	54
Şekil 5.12: Sağ ayak için potansiyometre-motor konum haritalama grafikleri.	55
Şekil 5.13: Sol ayak motor yörüngeleri _m ekli ve potansiyometre değerleri _p ekli. Yatay eksen zaman (s).....	56
Şekil 5.14: Sağ ayak motor yörüngeleri _m ekli ve potansiyometre değerleri _p ekli. Yatay eksen zaman (s).....	57

MERDİVEN ÇIKIP İNEBİLEN İKİ AYAKLI ROBOTUN DİNAMİK MODELLENMESİ VE DENETİMİ

ÖZET

İnsansı robotlar insanın eklem yapısında ve insanın yaptığı işlemleri gerçekleştirmesi beklenen robotlardır. Endüstride, uzayda, servis ve sağlık sektöründe, çalışma ortamında değişiklik yapmayı gerektirmeyen, insan gibi hareketliliğe sahip olacak robotlar insanların yaşam kalitesini arttırabilecektir. İnsansı robotlar günümüzde endüstriyel ortamlarda kullanılacak kadar gelişmiştir. İnsansı robotlar temel olarak iki kısımdan oluşur. İki adet bacağın kalçaya bağlanmasıyla oluşan ve gövdeyi taşıyan alt kısım yürüme işleminden sorumludur. İki kolun bağlandığı, merkezi bilgisayar, algılayıcılar ve güç kaynağından oluşan üst gövde de yapılacak görevleri gerçekleştirir. Bu çalışma iki bacak bir kalçadan oluşan alt kısmın yürüme işlemini kapsamaktadır.

İnsansı robotların yürümesinde mekanik açıdan statik ve dinamik, denetim açısından da aktif ve pasif yürüme biçimleri mevcuttur. Dinamik modelleme tam veya basitleştirilmiş model kullanılarak gerçekleştirilebilir. Basitleştirilmiş model kütle merkezi ve ayak yörüngelerini oluştururken, ayrıca dengeleyici döngüsünde, tam dinamik model ise görev gerçekleştirme ve simülasyon çalışmalarında kullanılmaktadır.

Robotun yere temas ettiği ayağın veya ayakların dışlarını çevreleyen bölgeye destek çokgeni denir. Ayak düzleminin yerde bulunan dönme merkezine ise sıfır moment noktası denir.

Statik yürümede kütle merkezi izdüşümü destek poligonu içinde kalacak şekilde yörünge oluşturulur. Kütle merkezi destek poligonu dışına çıktığında robot düşer. Dinamik yürümede ise sıfır moment noktası destek poligonu içinde kalacak şekilde yörünge oluşturulmalıdır. Dinamik yürümede şart sıfır moment noktasının destek poligonu içinde kalmalıdır, sıfır moment noktası destek poligonu içinde kaldığı müddetçe kütle merkezi izdüşümü destek poligonu içinde olmayabilir.

Robot yürümesinde ilk önce kütle merkezi ve ayak yörüngeleri oluşturulur. Sonra robot eklemleri bu yörüngeleri takip edecek şekilde oluşturulur. Eklem boşlukları, yerdeki düzensizlikler, sürtünme kuvvetinin ideal olmaması, uzuv esneklikleri oluşturulan yörüngeye olduğu gibi uygulanmasına izin vermemektedir. Bu nedenle eklem açılarını oluşturduktan sonra yürüme dengeleyicisi tasarlanır. Dengeleyicinin görevi ideal olmayan durumlarda robotun dengesini sağlamaktır.

Robota dışarıdan kuvvet uygulanması veya yürürken ön görülemeyen derecede bozuk zeminle karşılaşılması gibi durumlarda yürümeyi devam ettirebilmek için de refleks denetleyici tasarlanır.

İnsansı robotların gerçek zamanlı kontrolünde görevlerin zamanlaması kritik öneme sahiptir. İki ayaklı robotların yürümesi sırasında kütle merkezinin ters kinematik hesabı kısa sürede hesaplanabilmektedir. İnsansı bir robotun kütle merkezinin ters

kinematiğinin hesaplanmasında sayısal yöntemler kullanılmaktadır. Bu yöntemlerde birleşim açıları küçük miktarlarda değiştirilir. Daha sonra kütle merkezinin istenilen konuma gelip gelmediği kontrol edilir. Ayak ve kütle merkezi yörüngelerine ek olarak kalça ve dizlerin konumu ve yönelim yörüngeleri de bilinmelidir. Sayısal yöntemlerle yapılan hesaplamalardaki hatalar 0,03-0,4m civarında olabilir. Bu hata bazı durumlarda robotun yürümesini engelleyebilir. Oluşan hata dengeleyicinin kullanılmasıyla robotun yürümesini engellemeyecek hale gelebilir.

Bu çalışmada kütle merkezi ve ayak yörüngesi oluşturma adımından sonraki eklem açılarının hesaplanması adımı arasında, robot kütle merkezinin hassas olarak takip edilmesini sağlayacak iteratif bir yöntem geliştirilmiştir. Yöntemin etkinliği sayısal bir model üzerinde dengeleyici ve refleks denetleyici kullanmadan açık çevrim olarak çalıştırılarak incelenmiştir.

Geliştirilen iteratif yöntemde eklem açıları yerine kalçanın kütle merkezinin yön hata vektörü doğrultusundaki konumu ve yönelimi parametre olarak kullanılmıştır.

Ayak ve kütle merkezi yörüngesi bilindiğinde robot konfigürasyonu kullanılarak eklemlerin başlangıç konumları elde edilir. Referans kütle merkeziyle mevcut kütle merkezi arasındaki hata vektörü hesaplanır. Hata vektörünün birim vektörü kullanılarak robot kalçasının hareket etmesi istenen doğrultu belirlenir. Bu doğrultuda birim vektör doğrultusunda belirli bir miktar kalça hareket ettirilir. Diğer kalça eklemi de hareket ettirilerek robot ağırlık merkezi tekrar hesaplanır. Hata vektörü tekrar hesaplanıp istenen hata miktarından daha küçük bir değere ulaşıp ulaşılamadığı belirlenir. Ulaşılamamışsa kalça indisleri değiştirilir ve diğer kalça hatanın birim vektörü doğrultusunda hareket ettirilir. Böylece robotun bütün eklemleri, kütle merkezini istenen konuma ulaştırmak için kullanılmış olur. Bu işlem hata miktarı önceden tanımlanan değerden daha küçük olunaya kadar devam eder. Çalışma sonucunda kütle merkezinin konumu 0,001 m veya daha iyi bir doğrulukla hesaplanmıştır. 2415 hesaplamanın bir saniyede tamamlanabilmesi nedeniyle geliştirilen yöntem gerçek zamanlı kontrol uygulamalarında kullanıma uygundur. Yöntemde gerekli olan tek yörünge ayak ve kütle merkezidir.

Çalışma sayısal, deneysel ve imalat olarak üç bölüme ayrılabilir.

Sayısal kısım için İTÜ Makina Mühendisliği Otomatik Kontrol Laboratuvarında geliştirilen İTÜ Biped'in Sayısal modeli MATLAB Sim MultiBody'de hazırlanmıştır. Yerçekimi ve temas etkileri sayısal modele dahil edilmiştir. İki ayaklı yürüyen robot olarak geliştirilen robot için Lineer Ters Sarkaç modeli ile dinamik bir yürüme yörüngesi oluşturulmuştur. Yürüme yörüngesi 2 basamakla merdiven çıkma, 2 basamakla düz zeminde yürüme, 2 basamakla merdiven inme ve durmadan oluşur. Yürüme operasyonu açık döngülü bir sistem olarak modellenmiştir. Robot yürüme yolunu düşmeden başarıyla tamamlamıştır. Ortalama yürüme hızı 0,2257 m/s olmuştur. Ortalama takip hatası ise literatürde 0.03-0.4m arasında verilen değerlerden daha küçük bir değer olan 0.0238m olmuştur.

Deneysel kısımda İTÜ Biped robotunun hareket sürücü interpolasyon moduna geçirilmiştir. Bu modda kullanılan parçalı yörüngelerin robotu sarsıntılı çalıştırdığı görülmüştür. Bu sarsıntıları engellemek için ara noktalı 3. Dereceden yörünge interpolasyonu yapılan bir yöntem geliştirilmiştir. Literatürde belirli bir hatayla poligona eğri uydurma şeklinde kullanılan yöntem kullanılmamıştır. Eklem konumları robot kütle merkezine doğrudan etki ettiği için konumda herhangi bir hata oluşturmayacak, hız sürekliliğinin sağlandığı çok geçişli bir ara noktaları 3. Dereceden yörünge hesaplama yöntemi geliştirilmiştir.

İmalat kısmında yeni bir robotun tasarımı ve üretimi yapılmıştır. İTÜ Biped üzerinde yapılan çalışmalar sonucunda robot üzerine bir takım değişiklikler yapmak gerektiği sonucuna ulaşılmıştır. Bu değişiklikler geriye döndürülemeyecek yapıda olduğu için yapılacak değişiklikleri içerecek, yeni bir robotun tasarımı ve üretimi yapılmıştır. Bu değişiklikler rijit bir mekanik tasarım, motor denetim sinyalleriyle sensör okuma hatlarının ayrılması, gerçek zaman denetime uygun bir yazılım alt yapısının hazırlanmasıdır. İmal edilen robot yapılması düşünülen değişikliklerin etkilerini içeren bir tasarım olmuştur.





DYNAMIC MODELING AND CONTROL OF A BIPEDAL ROBOT THAT CAN CLIMB AND DESCEND STAIRS

SUMMARY

Humanoid robots are robots that are expected to perform the joint structure of humans and the operations performed by humans. In industry, space, service, and health sectors, robots that do not require changes in the working environment and have mobility like humans will be able to increase people's quality of life. Humanoid robots are advanced enough to be used in industrial environments today. Humanoid robots basically consist of two parts. The lower part, which is formed by connecting two legs to the hips and carries the body, is responsible for the walking process. The upper body, which consists of the central computer, sensors and power source to which two arms are connected, also performs the tasks to be done. This study covers the walking process of the lower part, which consists of two legs and one hip.

There are static and dynamic walking styles in terms of mechanics and active and passive walking styles in terms of control in the walking of humanoid robots. Dynamic modelling can be performed using a full or simplified model. The simplified model creates the centre of mass and foot trajectories and is also used in the balancer cycle, while the full dynamic model is used in task execution and simulation studies.

The area surrounding the outsides of the foot or feet where the robot touches the ground is called the support polygon. The centre of rotation of the foot plane on the ground is called the zero moment point.

In static walking, the trajectory is created in such a way that the centre of mass projection remains within the support polygon. The robot falls when the centre of mass goes outside the support polygon. In dynamic walking, the trajectory should be created in such a way that the zero-moment point remains within the support polygon. In dynamic walking, the condition is that the zero moment point must remain within the support polygon; as long as the zero moment point remains within the support polygon, the centre of mass projection may not be within the support polygon.

In robot walking, the centre of mass and foot trajectories are first created. Then, the robot joints are created to follow these trajectories. Joint gaps, irregularities on the ground, friction force not being ideal, and limb flexibility do not allow the created trajectory to be applied as it is. For this reason, the walking balancer is designed after creating the joint angles. The task of the balancer is to ensure the balance of the robot in non-ideal situations.

A reflex controller is also designed to continue walking in situations such as external force being applied to the robot or encountering unpredictably uneven ground while walking.

The timing of tasks is of critical importance in the real-time control of humanoid robots. The inverse kinematic calculation of the centre of mass of bipedal robots during walking can be calculated in a short time. Numerical methods are used in the

calculation of the inverse kinematics of the centre of mass of a humanoid robot. In these methods, the joint angles are changed in small amounts. Then, it is checked whether the centre of mass has reached the desired position. In addition to the foot and centre of mass trajectories, the position and orientation trajectories of the hips and knees should also be known. Errors in calculations made with numerical methods can be around 0.03-0.4 m. This error can prevent the robot from walking in some cases. The error that occurs can become less likely to prevent the robot from walking with the use of a balancer.

In this study, an iterative method was developed to ensure that the robot centre of mass is followed precisely between the calculation of joint angles after the centre of mass and foot trajectory creation step. The effectiveness of the method was investigated by running it as an open loop on a numerical model without using a balancer and reflex controller.

In the developed iterative method, the position and orientation of the centre of mass of the hip in the direction of the direction error vector were used as parameters instead of joint angles.

When the foot and mass centre trajectory are known, the initial positions of the joints are obtained using the robot configuration. The error vector between the reference mass centre and the current mass centre is calculated. The desired direction in which the robot hip will move is determined using the unit vector of the error vector. In this direction, a certain amount of hip is moved in the direction of the unit vector. The other hip joint is also moved, and the robot's centre of gravity is recalculated. The error vector is recalculated, and it is determined whether a value smaller than the desired error amount can be reached. If not, the hip indices are changed, and the other hip is moved in the direction of the error unit vector. Thus, all joints of the robot are used to reach the centre of mass to the desired position. This process continues until the error amount is smaller than the previously defined value. As a result of the study, the location of the centre of mass was calculated with an accuracy of 0.001 m or better. Since 2415 calculations can be completed in one second, the developed method is suitable for use in real-time control applications. The only trajectory required in the method is the foot and mass centre. The study can be divided into three parts: numerical, experimental and manufacturing.

For the numerical part, the numerical model of the ITU Biped developed in the ITU Mechanical Engineering Automatic Control Laboratory was prepared in MATLAB Sim MultiBody. Gravity and contact effects were included in the numerical model. A dynamic walking trajectory was created for the robot, which was developed as a bipedal walking robot with the Linear Inverted Pendulum model. The walking trajectory consists of climbing stairs with two steps, walking on flat ground with two steps, descending stairs with two steps and stopping. The walking operation was modelled as an open-loop system. The robot completed the walking path without falling. The average walking speed was 0.2257 m/s. The average tracking error was 0.0238m, which is smaller than the values given in the literature between 0.03-0.4m.

In the experimental part, the motion driver of the ITU Biped robot was switched to interpolation mode. It was observed that the piecewise trajectories used in this mode caused the robot to operate jerkily. To prevent these tremors, a method has been developed that uses 3rd-degree orbit interpolation with intermediate points. The method used in the literature to fit a curve to a polygon with a certain error has not been used. Since the joint positions directly affect the robot's centre of mass, a multi-

pass 3rd-degree orbit calculation method has been developed that will not create any errors in the position and ensures speed continuity. A new robot has been designed and manufactured in the manufacturing section. As a result of the studies conducted on the ITU Biped, it has been concluded that some changes need to be made to the robot. Since these changes are irreversible, a new robot has been designed and manufactured, and the changes to be made will be included. These changes are a rigid mechanical design, separation of motor control signals and sensor reading lines, and preparation of a software infrastructure suitable for real-time control. The manufactured robot has been designed to include the effects of the changes that were considered to be made.





1. GİRİŞ

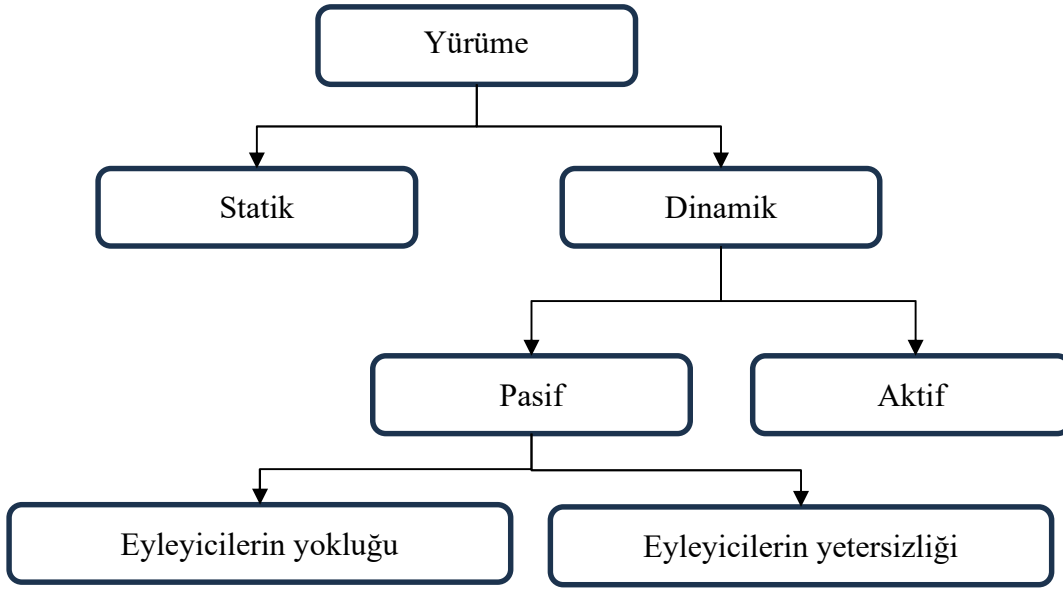
İnsansı robotların kullanımının yaygınlaşabilmesi için, insanların yürüyerek hareket etmek için kullandığı yolların tamamını kullanabilir olmaları gerekmektedir. İTÜ Biped robot üzerinde bu amaçla yapılan sayısal çalışmalar vardır (Bayraktaroğlu et al., 2018; Gerçek, 2011; TAN, 2012; Tasasız, 2011). Bu çalışmalardan edinilen bilgi birikimi kullanılarak, ek sensörler, mikro denetleyiciler ve yürüyüş planlama yöntemleriyle, robotun deneysel olarak merdiven çıkıp inmesinin sağlanabilmesi halinde, geliştirilen İTÜ Biped robotu günlük kullanıma bir adım daha yaklaşabilir.

1.1 Tezin Amacı

Yürüyen robotun merdiven çıkıp inebilmesini sağlayacak bir yürüyüş planlama ve kararlılık denetimi yöntemi geliştirilmesi amaçlanmıştır. Çalışmalar İTÜ Biped robotunun mekanik olarak rijitleştirilmesini, yörünge planlama yazılımının yenilenmesini ve daha önce sayısal ortamda yapılan uygulamaların deneysel ortamda yapılmasını kapsamaktadır. Ancak ulaşılan bulgular, İTÜ Biped üzerinde geri alınamayacak değişikliklerin yapılmasını gerektirdiğinden, bu değişikliklerin etkilerinin izlenebileceği, yeni bir robotun tasarımı ve üretimi yapılmıştır.

1.2 Temel Kavramlar

İnsansı robotlar yerçekimi, temas kuvvetleri ve eklemlere uygulanan torklarla hareket etmektedir (Sugihara & Morisawa, 2020a). Robot yürüyüşündeki en temel kavram Zero Moment Point'dir (Sıfır Moment Noktası – SMN). Robot ayaklarının yerle temas eden bölgesinin sınırlarına destek poligonu adı verilmektedir. Temas kuvvetleri ve torkları robota yerle temas ettiği bu bölgeden uygulanmaktadır. Bu bölge içindeki öyle bir nokta vardır ki, torkların yatay bileşeni sıfır olmaktadır. Ayağın yerle temas düzlemi XY düzlemi olarak ele alındığında SMN, torkun X ve Y bileşeninin sıfır olduğu noktada olmaktadır (Vukobratovic & Borovac, 2012).



Şekil 1.1: Yürümenin sınıflandırılması

Yürüme Şekil 1.1’de görüldüğü gibi sınıflandırılmaktadır. Statik yürümede kütle merkezinin izdüşümü sürekli olarak destek poligonu içinde kalmak zorundadır. Dinamik yürümede ise kütle merkezi izdüşümü destek poligonu içinde kalmak zorunda değildir. SMN destek poligonu içinde kalmalıdır, yoksa robot yürüyememekte ve düşmektedir. Aktif yürüme robotun bütü eklemlerinin tahrik edildiği ve eyleyici güçlerinin robottan beklenen bütün hareketleri gerçekleştirebilecek güçte olduğu durumdur. Pasif yürüme ise bütün eklemlerin tahrik edilmediği veya eyleyici güçlerinin istenen hareketleri gerçekleştirmede yetersiz kaldığı durumu ifade etmektedir.

Yürüme işleminde ilk olarak yürüyüş ve adım planlama yapılır (Chevallereau et al., 2018; Rokbani et al., 2009; Yin et al., 2009;). Bu işlemde robot kütle merkezinin ve ayakların yörüngesi elde edilir. Robotun, kütle merkezine göre yürümesi istendiğinde ters sarkaç yöntemi kullanılır (Tang & Er, 2007). Ayak konumlarına göre yürümede ise sepet-masa modeli kullanılmaktadır (Kajita et al., 2003a). Her iki yöntemde de sabit ayağın yerle bağlantı noktası olarak ZMP (Zero Moment Point – Sıfır Moment Noktası, SMN) kullanılır (Vukobratovic & Borovac, 2012).

Aktif robot yürümesinde tam ve basitleştirilmiş dinamik modeller kullanılmaktadır (Sugihara & Morisawa, 2020b).

Uzuvları hareket uzayında istenen konuma yerdeğiştirebilmek için ters kinematik, uzuv uzayında ise ileri kinematik yöntemi kullanılır (Lavor et al., 2018). İkinci adımda robotun bu yörüngeleri takip edebileceği şekilde ters kinematik hesaplamalar yapılır. Ayak ve kalça konumu bilindiğinde eklem açıları ters kinematik yöntemle hesaplanabilmektedir. Bu hesaplamanın yapılmasında analitik veya sayısal yaklaşımlar bulunmaktadır (Alba et al., 2010; Atmeh & Subbarao, 2018; Hildebrandt et al., 2018; Kobayashi et al., 2018; Lim et al., 2014; Maalouf et al., 2020).

Ancak bu, robot kütle merkezinin istenen hassasiyette takip edildiğini garanti etmez.

1.3 Literatür özeti

Çalışmaların tamamına yer vermek mümkün değildir. Bu kısımda sciencedirect.com, webofknowledge.com ve scholar.google.com’da “bipedal walking robot”, “stairs”, “dynamic walking”, “humanoid robot”, “ground reaction force based control” anahtar kelimeleriyle arama yapılmış ve çıkan sonuçlardan uygun olduğu değerlendirilenler listelenmiştir.

Her bir ayağın SSP (Single Support Phase – Tek destek fazı) durumu için ZMP ayrı ayrı hesaplanmıştır. Sonra birleşik durum için DSP (Double Support Phase – Çift destek fazı) oluşturulmuştur. Bu durum kullanılarak dinamik denge aralığı ve en düşük güç tüketimini birlikte sağlayacak çok amaçlı optimizasyon yapılmıştır. Optimizasyon hem Genetik Algoritmalar hem de Parçacık Sürü Optimizasyonu teknikleri kullanılarak yapılmıştır (Rajendra & Kumar Pratihar, 2015).

3 boyutlu iki ayaklı yürümede enerji verimliliği, kararlılık ve bir eksik eyleyicili tahriği optimize etmek için bir yöntem geliştirilmiştir. Eksik eyleyicili tahrikte sayısal simülasyon yapmadan robot kararlılığının belirlenebilmesine çalışılmıştır. Bunu yapabilmek için noktasal kütle ayak yerine çizgisel ve kütsüz ayak kabulü yapılmıştır. Çizgisel ayak için temas ifadeleri çıkarılmış ve RABBIT isimli iki ayaklı yürüyen robota uygulanmıştır. Robotun 0,2 m/s ile 1,5m/s arasındaki hızlarda yüksek enerji verimliliğiyle yürüyebildiği ortaya koyulmuştur (Chen et al., 2016).

Atlas insansı robotun karmaşık ortamlarda dinamik planlaması, kontrolü ve durum tahmini için optimizasyon algoritmaları geliştirilmiştir. Optimizasyon algoritmaları adım planlamadan, tüm gövde dinamiğine kadar uygulanmıştır. Geliştirilen algoritmalar Boston Dinamik tarafından üretilen Atlas üzerinde denenmiştir. Mevcut

yaklaşımlar adım planmanın ayırık veya optimizasyonla yapılması üzerinedir. Çalışmada iki yaklaşım da planlamadaki farklı problemleri çözmek üzere bir arada kullanılmıştır. Adımların konveks yüzeylere yerleştirilmesi sınırlayıcı olarak kullanılmış, konveks yüzey tespit edilemediğinde düz yüzey aranmıştır. Hiçbir zaman konkav bir yüzeye adım atılmamıştır. Ortam LIDAR (Laser imaging, detection, and ranging – Lazer görüntüleme, algılama ve mesafelendirme) yöntemiyle algılanmıştır (Kuindersma et al., 2016).

İki ayaklı bir robotun yürüebilmesi için gerekli donanım ve yazılım altyapısı verilmiştir. Yürüme fazlarını, bu fazlar arasındaki geçiş sıralamasının durumlar cinsinden verildiği çalışma, yürüme algoritmasını geliştirmede kolaylık sağlamıştır. Donanımın hangi birimlerden oluşması gerektiği, bu birimler arasında iletişimin sağlanma yöntemi verilmiştir. ZMP, CoM ve CP arasındaki ilişkiler, bilek, diz ve kalça eklemi hesaplamalarıyla yürüme bağlantıları tanımlanmıştır. Ters sarkaçla yürüyüş planlama ve kapalı çevrim kontrol diyagramları dışında iki ayaklı bir robotun yürüebilmesi için gerekli teknik ve matematik bilgisi çalışmada verilmiştir. Sonuç olarak robotun boyutlarına göre ihmal edilebilecek bir hatayla düz ve zig-zag çizerek yürüyebildiği ortaya koyulmuştur. Çalışma ayrıca dinamik yürüme konusunda temel çalışmalara referanslar içermektedir (Kien et al., 2017).

Yürüyüş örüntü oluşturucuların çoğunlukla ZMP tabanlı olduğu belirtilmiştir. MABEL ve ATRIAS insansı robotlarının ise farklı olarak non-lineer kontrol ve Sonlu Olay Makinesi kullanılarak denetlendiği, bu denetim tipinin dinamik yürüyüş ve kararlılıkta oldukça başarılı olduğu belirtilmiştir. 3D ve fiziksel uygulamasının ise çok nadir olduğu belirtilmiştir. Çalışmada adım konumları oluşturulduktan sonra doğrudan ZMP konumları hesaplanmamış, öncelikle non-lineer sınırlayıcılı optimizasyon yöntemiyle adım konumları izlenecek yörüngeye göre güncellenmiştir. Oluşturulan ZMP yörüngesinin izlenmesi için de MPC (Model Predictive Control – Model Ön Görümlü Denetim) kullanılmıştır. Çalışmada robotun genel denetimi ve takibi için bir yazılım altyapısı geliştirilmiştir. Yazılım altyapısında farklı yürüme modülleri (dönme, yan yan yürüme vs.) gibi eklentiler kullanılabilir. Ayrıca IDS (Interactive Data Server - Etkileşimli Veri Sunucusu) Robotun bütün bilgilerine her an ulaşılabilmesini sağlamaktadır. Yazılım altyapısı COMAN insansı robotunda uygulanmıştır. Sonuç olarak kararlılığı yüksek bir dinamik yürüme elde edildiği ortaya koyulmuştur (Zhou et al., 2017).

Hy-DAC (Hybrid Dynamic Autonomous Control – Hibrit dinamik otonom denetim) yöntemi dinamik yürümeyi içerek şekilde geliştirilmiştir. Yer temas merkezi tabanlı bir robot hız düzenleyici yönteme eklenmiştir. Yöntemde yapılan geliştirmelerle düzensiz bir merdiven yapısı sayısal ortamda robota izletilmiştir. Belirli aralıklarda rasgele değerler alan basamak parametreleriyle oluşturulan 2D merdivende robotun 0,5m/s hızla ilerleyebildiği gösterilmiştir. Robot ayrıca ayak uzunluğunun 1,5 katı kadar yükseklikte olan basamaklardan tırmanabilmiştir (Zachariah & Kurian, 2017).

Adım uzunluğunu gerçek zamanlı olarak değiştirebilen bir algoritma geliştirilmiştir. Algoritma dış etkilere karşı dengeyi etkili bir biçimde koruyabilmektedir. Bozucu etki Capture-Point Error (Yakalama Noktası Hatası) olarak belirlenmiştir. Gerekli ZMP, MPC tabanlı bir ZMP denetleyicisiyle hesaplanıp güncellenmiştir. ZMP'yi destek poligonu içinde tutabilmek için quadratik programlama ile oluşturulan bir amaç fonksiyonu kullanılmıştır. Algoritma DRC-HUBO+ insansı robotunun dış etkilere karşı dengesini koruyabilmesini ve yürüme esnasında bilinmeyen engellere çarptıktan sonra dengede kalabilmesini sağlamıştır (Joe & Oh, 2018).

Yürüme tek destek ve çift destek fazı olarak ikiye ayrılmıştır. Fazlar arasında geçişin dinamiğinin hesaplanabilmesi için optimizasyon yöntemlerine ihtiyaç duyulmaktadır. Genelleştirilmiş bir ters sarkaç modeliyle bu probleme çözüm getirilmeye çalışılmıştır. Geliştirilen ters sarkaç modeli hem tek hem de çift destek fazında kullanılabilmiştir (Tiseo et al., 2019).

Gövde konumunu en yüksek konumda tutma amaçlı optimizasyon probleminin çözümü yapılırken robot ikisi ayakları diğeri de gövdeyi temsil eden 3 kütle ile temsil edilecek şekilde bir dinamik model oluşturulmuştur. Bu modelde robotun ayakları 2 boyutlu olarak modellenmiştir. Yürüme işleminin insan yürümesine benzer olabilmesi için hem gövde yüksekliği hem de eklem hızları optimize edilmiştir. Deneysel çalışma sonucunda 2 boyutlu modelde kalça eklemleri dikkate alınmadığı için deneysel çalışmayla 15° lik bir fark olduğu tespit edilmiştir (Hildebrandt et al., 2018).

Adım planlanmasında önce robot kütle merkezinin yörüngesi kullanılarak ayak konumları oluşturulmaktadır. Belirtilen yöntemde kullanılan sayısal iterasyon ve optimizasyon yöntemlerinin kullanılmasını gereksiz kılan bir yöntem geliştirilmiştir. Başlangıç konumundan bitiş konumuna kadar olan yörüngede ZMP'nin harmonik olarak değiştiği kabulü yapılmıştır. İki konum arasında adım aralığı ve uzunluğu

kullanılarak ZMP yörüngesi oluşturulmuştur. ZMP yörüngesini takip edecek şekilde robot ters kinematiği hesaplanıp eklem yörüngeleri oluşturulmuştur. Sonuç olarak sınır koşullarının tek bir parametreyle belirlenebildiği bir yöntem oluşturulmuştur. Yöntemin adı GDC (Generalized Direction Curve - Genelleştirilmiş doğrultu eğrisi) olmuştur (Jatsun et al., 2018).

Referans hızlar robotun geçerli durumuna göre belirlenmelidir. Buna göre bir robotun sanal dinamik modeli oluşturulmuş ve kendisine verilen hedefe ulaşacak şekilde yörünge planlaması yapılmıştır. Robotun ulaşmak istediği hedef sanal çekme, ortamdaki engeller ise sanal itme kuvveti uygulayarak engellere çarpmadan hedefe ulaşılması sağlanmıştır. Bu esnada robotun yaşayabileceği ani hareket değişimleri ise sanal dinamik model kullanılarak gerçek zamanlı önlem alınarak yumuşatılmıştır (Kobayashi et al., 2018).

İnsanın düşmeye verdiği tepkilere benzer tepkiler üretebilecek bir yapı geliştirilmiştir. Görüntü algılaması LIDAR kullanılarak yapılmıştır. Kulağın işlevi ise bir IMU (Inertial Measurement Unit – Atalet ölçüm birimi) kullanılarak gerçekleştirilmiştir. Parçalı ve hareketli bir platform üzerine robot yerleştirilmiş, ayakların temas durumları, platformun eğim açısı gibi değişkenler değiştirilerek robotun düşme durumunu algılaması sağlanmıştır. Deneyler sayısal ortamda yapılmıştır (Subburaman et al., 2019).

Hibrit dinamik kontrolde birbiriyle iç içe iki denetleyici kullanılmaktadır. Birisi çevrimdışı olarak yürüme yörüngelerini oluştururken, diğeri de bu yörüngelerin takibinde kullanılmaktadır. Bu tip birleşik denetleyicilerin bozucu etkilere karşı yeterince başarılı olmadığı görülmüştür. Buna karşı adım uzunluğunu giriş parametresi olarak alan bir gürbüz denetleyici geliştirilmiştir. Adım uzunluğu değiştirilebildiği için bozucu etkilere karşı çevrimiçi düzeltmeler uygulanabilmektedir. Denetleyici 5 serbestlik dereceli bir alt gövde halindeki iki ayaklı yürüyen robota uygulanmıştır. Uygulanan bozucu etkilere karşılık robotun normal yürüme çevrimine geri dönebildiği Poincare haritalarıyla ortaya koyulmuştur (Arcos-Legarda et al., 2019).

Yürüme esnasında oluşan dönme momenti robotun düşmesine neden olduğundan izlenmesi gerekmektedir. Dönme momentinden dolayı robotun düşmesini engelleyecek ve enerji verimliliğini iyileştirecek bir çalışma yapılmıştır. Çalışmada FLS (Fuzzy Logic System - Bulanık mantık sistemi) kullanılarak bir ara denetleyici

tasarlanmış ve kullanılmıştır. Doğal bir yürüme görüntüsü için ZMP sabit değil hareketli kabul edilmiştir (Yang et al., 2019).

Yürüyüş kararlılığını yükseltmek için atik bir ayak tasarımı geliştirilmiştir. Ayak hızlı hareket edebildiği zaman denge bozulmalarına daha hızlı karşılık verilebileceğinden yola çıkılmıştır. Bu amaçla ayak eklemi 2 serbestlik dereceli bir universal paralel robot bağlantısıyla yapılmıştır. 5 ve 10 derecelik eğimlerde robotun yürümesi deneysel olarak incelenmiş, 0.6m/s hızla kararlı bir şekilde yürüme gerçekleştirilmiştir (Jeong et al., 2020).

Adım planlama için planlama ve ön hesaplama yapılması gerekmektedir. Sonra da adım uzunluklarına göre robot kinematiği, ZMP ve CoM konumu hesaplanmaktadır. Adım planlama işlemine bir optimizasyon problemi olarak yaklaşan bir çalışma yapılmıştır. Çalışmada robot kinematiği, adım uzunlukları, CoM ve CP sınırlayıcıları kullanılmıştır. Bu sayede adımlar gerçek zamanlı olarak planlanabilmektedir. Çalışmada ayrıca yürüyüş ve adım konumu planlama hakkında sınıflandırılmış çalışmalara referanslar da mevcuttur (Hong & Lee, 2020).

İnsan yürümesinin stratejik, taktik ve uygulama katmanlarından oluştuğu belirtilmiştir. Stratejik katmanın beyincik, taktik katmanın nöronlar ve uygulama katmanının da kaslar ve eklemlerden ibaret olduğu belirtilmiştir. Stratejik katmanın çalışması irdelenerek enerji dengesi üzerine bir yöntem geliştirilmiştir. Kinetik enerjinin en küçük olduğu yerde potansiyel enerjinin en büyük olması gerekmiştir. Bunun için 3D-LIP modelindeki enerji hesaplamaları kullanılmıştır (Kajita et al., 2001). Sonuç olarak simülasyonda %10, deneysel çalışmada ise %1,8 oranında enerji tasarrufu yapılmıştır. Çalışmadaki 27 ve 28 numaralı kaynaklar yürümenin açık ve kapalı çevrim durumlarını incelemesi bakımından önemlidir (Maalouf et al., 2020.).

1.4 Hipotez

İnsansı robot eklem açılarına göre robot kütle merkezi bulunmak istendiğinde, ileri kinematik yöntemle hesap edilebilmektedir. Robotun belirli bir kütle merkezi konumuna sahip olacak şekilde eklem konfigürasyonunun ayarlanması ise doğrudan yapılamamaktadır. Bu işlem için geliştirilmiş sayısal yöntemler vardır. Bu yöntemler farklı sınırlayıcılar kullanan optimizasyon yöntemlerine dayanmaktadır. Yapılan optimizasyon sonucu eklem açıları belirlenir ve robota uygulanır. Robot belirli bir

hatayla başta belirtilen kütle merkezi konumuna sahip olur. Literatürde bu hesaplamalar eklem açılarını değişken kabul edilerek yapılmaktadır. Halbuki, ayak ve kütle merkezi konumları bilindiğinde ters kinematik yöntemle eklem açıları belirlenebilir ve ileri kinematik hesaplamalarla robot kütle merkezi bulunabilir. Bu çalışmada ayak ve kalça konumları değişken kabul edilerek robot kütle merkezi konumunu milimetre doğruluğunda hesaplayabilen bir yöntem geliştirilmiştir.

1.5 Problem Tanımı ve Bilimsel Katkılar

Robotların günlük hayatta kullanımının arttırılabilmesi için geliştirme ortamlarının da günlük hayatta kullanılmalarının beklendiği ortamlara benzemesi gerekmektedir. Yalnızca sayısal ortamda yapılan çalışmalar, deneysel yapıya uygulandığında aynı sonucu vermemektedir.

Kütle merkezi robotun iki ayağının bağlantı noktalarının ortası veya daha yüksek bir konumda olmaktadır (Carpentier et al., 2016; Chae, n.d.; Khomariah et al., 2016; Mandava & Vundavilli, 2016; Olcay & Özkurt, 2017; Park et al., 2006; Tevatia & Schaal, 2000; Yang et al., 2018). Kütle merkezi konumu robot harekete başlamadan hesaplanır. Hareket boyunca kalçanın konumu ile kütle merkezi konumu arasındaki fark sabit kabul edilir (Kajita et al., 2003). Robot kütle merkezinin destekleyen ayağa göre sabit olduğu varsayımı da kullanılmaktadır (Efrain et al., 2014).

Kütle merkezi konumu hassas olarak hesaplanmadığında temas kuvvetinden kaynaklanan kütle merkezinin konumundaki sapma robotun yürümesini engellemektedir (Shibata & Natori, 2000). Literatürde bu problemin çözümü için uygulanan yöntemler sonraki paragrafta listelenmiştir.

Robot ayakları büyütülmek zorunda kalmıştır (Sugihara, 2011). Hesaplama doğruluğunu arttırmak için kütle merkezi ve ayak yörüngelerine ek olarak, kalça konum, kalça konum ve oryantasyon yörüngelerini gerektiren çalışmalar yapılmıştır (Bajrami et al., 2016; Mandava & Vundavilli, 2016). Ayak ağırlığının toplam robot ağırlığına göre yüksek olduğu durumlarda salınan ayağın yürüme dinamiğine etkisi büyük olmaktadır. Bu etkiyi azaltmak için salınan ayağı da sarkaç olarak modelleme yöntemi kullanılmaktadır. Yine de kütle merkezinin takibinde hata olmuş ve bahsedilen kabuller nedeniyle robotun hareketleri kısıtlanmıştır (Buschmann et al., 2007).

Uygulamada kontrolcünün bu hatalara karşı robotu istenen sınırlarda hareket ettirebilmesi hedeflenmektedir. Ulaşılabilen en küçük hata ters dinamik kontrol yöntemiyle 0.18m (Nakanishi et al., 2007) non-lineer optimizasyon kontrol yöntemiyle 0.03-0.4m olmuştur (Clever et al., 2018).

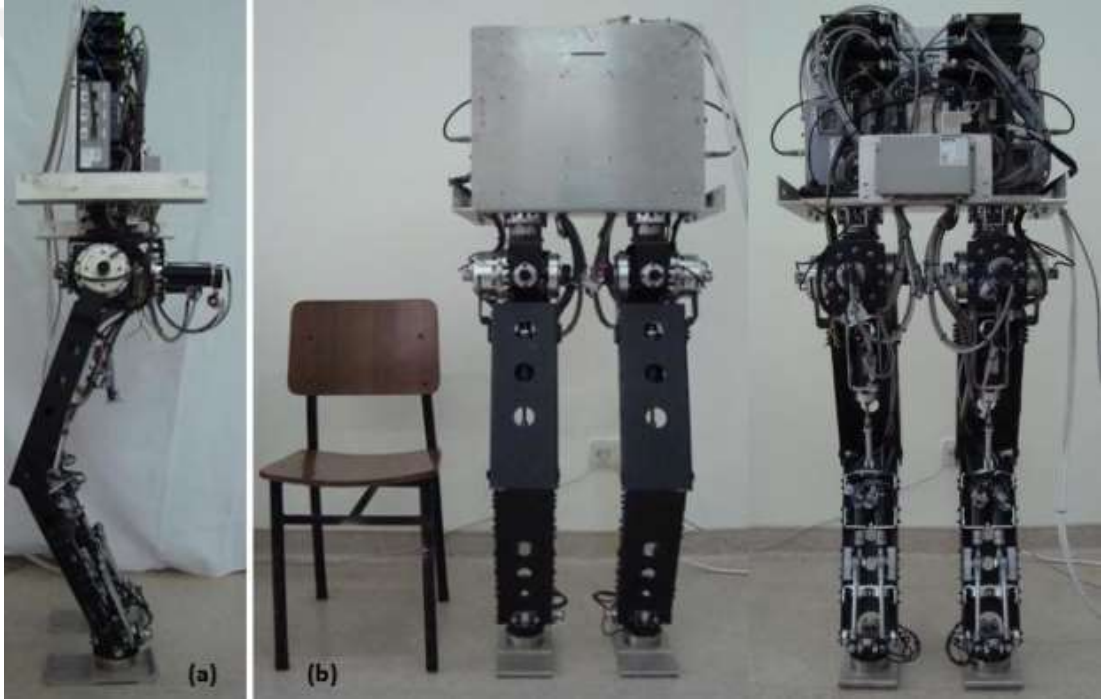
Bu çalışmada kütle merkezi konumunun hassas olarak takip edilebilmesi için eklem açılarını hesaplayan iteratif bir yöntem geliştirilmiştir. Yöntem kütle merkezini 0.001m veya daha yüksek hassasiyetle kesin olarak hesaplayabilmektedir. Hesaplama sonucunda tek bir çözüm sunulmaktadır. Çözümler içinden optimum çözümün seçilmesi gerekmemektedir. Yalnızca kütle merkezi ve ayak yörüngesinin bilinmesi yeterli olmaktadır.





2. YÖNTEM

Çalışmanın yapıldığı robot iki ayaklı yürüeyebilen bir robottur. Kalça ekleminde 3 adet motorun dönme eksenini, bilek ekseninde ise 2 adet motorun dönme eksenini tek bir noktada kesişecek biçimde yerleştirilmiştir. Üretimi yapılan robot Şekil 2.1’de görülmektedir (Yağız BAYRAKTAROĞLU et al., 2018).



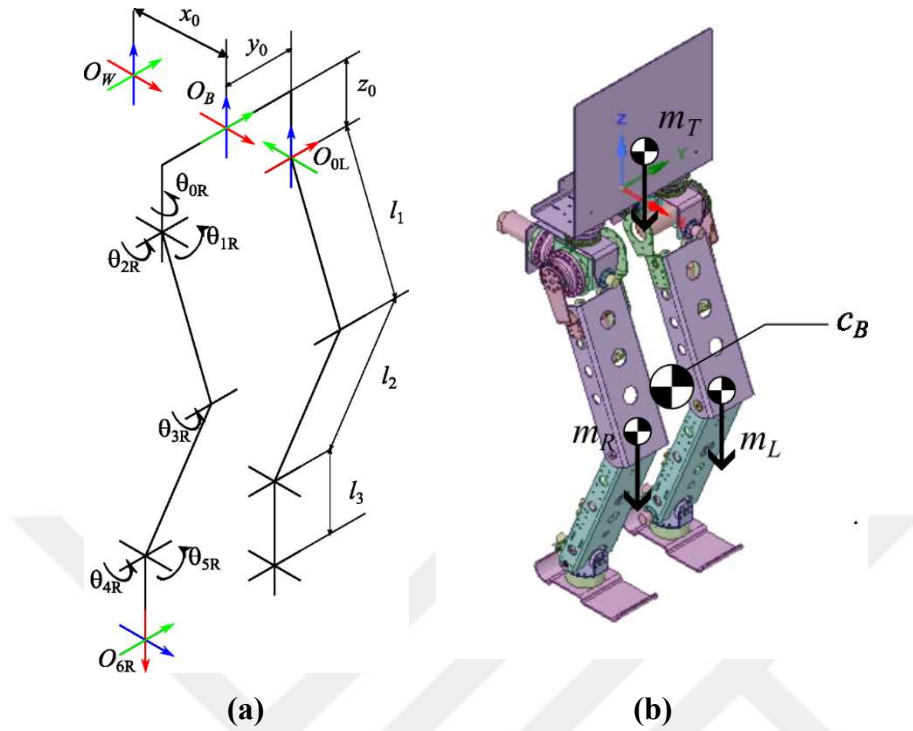
Şekil 2.1: Robot görüşleri. (a) yan, (b) ön ve arkadan görüşü.

Diz eklemi vidalı mil tahrikiyle sağlanmaktadır. Bilekte, eksenleri aynı noktada kesişen iki adet motor vardır. Baldır uzvuna yerleştirilen iki adet vidalı mil ile tahrik edilmektedir. Vidalı mil hareketi bileğe paralel kol mekanizmasıyla aktarılmaktadır.

2.1 Robotun Yapısı

Çalışmanın yapıldığı robot sayısal olarak modellenen, iki ayaklı yürüeyebilen bir robottur. Kalça ekleminde 3 adet motorun dönme eksenini, bilek ekseninde ise 2 adet motorun dönme eksenini tek bir noktada kesişecek biçimde yerleştirilmiştir.

Robotun eklem tipleri ve konumları Şekil 2.2’de gösterilmiştir.



Şekil 2.2: Robot eklem açıları ve temel boyutlar. (a) eklem ve uzunluk parametreleri; (b) bacak indisleri ve yaklaşık kütle merkezi konumları.

Şekil 2.2 (a)’da, O_W sabit koordinat sistemini, O_B robotun temel koordinat sistemini, x_0 harekete başlangıç noktasını, y_0 kalça genişliğini ve z_0 da robotun kalça yüksekliğini göstermektedir. Kırmızıyla gösterilen x eksenini, yeşil ve maviyle gösterilenler de sırasıyla y ve z eksenleridir. O_{0L} sol bacağın kalça bağlantı noktası, O_{6R} ayak koordinat sistemi ve θ_{nR} de sağ bacağın eklem açılarıdır. Okunabilirliği arttırmak için sağ bacağın kalça bağlantı noktası ve sol bacağın eklem açıları gösterilmemiştir. l_1 , l_2 ve l_3 robotun uzuv uzunluklarıdır. Uzunluk parametrelerinin değerleri

Çizelge 2.1’de verilmiştir.

Çizelge 2.1: Uzunluk parametreleri.

Simge	Değer	Birim
x_0	-0.14	m
y_0	0.13	m
z_0	0.07	m
l_1	0.422	m
l_2	0.4	m
l_3	0.075	m

Uzuv koordinat sistemine göre kütle merkezi konumları ve uzuv ağırlıkları

Çizelge 2.2’de verilmiştir.

Çizelge 2.2: Kütle merkezi konumları ve kütleler.

Parça ismi	c_x (m)	c_y (m)	c_z (m)	m (kg)
Gövde	0.000006	0.020549	0.157635	2.387902
Kalça yalpalama	0.000003	0.035534	0.0661	2.139072
Kalça dönme	-0.009912	0.002801	-0.0228	0.71518
Üst bacak	0.186095	-0.004384	-0.003505	3.804756
Alt bacak	0.180778	-0.007	0.000426	2.805996
Ayak bileği yalpalama	-0.000011	-0.001187	0.001606	0.022225
Ayak	0.058582	0.000012	0.018663	1.584761

2.2 Kinematik

Bu tezde geliştirilen yöntem bacak ağırlık merkezlerinin ayrı ayrı hesaplanmasını gerektirmektedir. Bacaklar bir matris yapısında ifade edilebildiği için kinematik hesaplamalarda Denavit-Hertenberg (DH) yöntemi kullanılmıştır (Denavit & Hartenberg, 1955). Link dönüşüm tablosu Çizelge 2.3’te verilmiştir.

Çizelge 2.3: Denavit-Hertenberg yöntemine göre link dönüşüm tablosu.

Ekleme no	θ	d	a	α
1	θ_0	0	0	$\pi/2$
2	θ_1	0	0	$-\pi/2$
3	θ_2	0	l_1	0
4	θ_3	0	l_2	0
5	θ_4	0	0	$\pi/2$
6	θ_5	0	l_3	0

Sabit koordinat sistemine göre link dönüşüm matrisleri Denklem 2.1-2.4 arasında verilmiştir.

$$O_B = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

O_B ’ ye göre O_{0R} ’nin dönüşüm matrisi. Sol ayak için ikinci satır, dördüncü sütuna y_0 yazılarak hesap yapılmaktadır.

$${}^B_0O_{0R} = O_B \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Eklem dönüşüm matrisi T ile ifade edilirse:

$${}^{n-1}_nT_n = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Bu ifadelerden sonra ayak konum ve oryantasyonu sıfırcı koordinat sistemine göre hesaplanabilir.

$${}^0RT_{6R} = {}^0RT_{1R} {}^1RT_{2R} {}^2RT_{3R} {}^3RT_{4R} {}^4RT_{5R} {}^5RT_{6R} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Yukarıda verilen ifadede denklemin sol tarafı DH matrisindeki değerlerin yerine koyulmasıyla el edilir. Sağ tarafı ise ayağın sıfırcı koordinat sisteminde elde edilen konum ve oryantasyon değerleri olmaktadır. Denklemin sol tarafının tersi alındığında ayaktan kalçaya olan zincir elde edilir. Her iki taraf ${}^5RT_{6R}^{-1}$ matrisiyle çarpılırsa bilekten kalçaya olan dönüşüm matrisi elde edilmiş olur (Ali et al., 2010).

Bilekte 2, kalçada 3 koordinat sistemi kesişmiş halde olduğundan diz eklemine değeri oluşan matrizen elde edilir. Buna göre Denklem 2.5-2.10 arasındaki ifadeler kullanılarak ters kinematik hesaplar yapılabilir.

$$\theta_4 = \frac{\sqrt{1 - \cos^2 \theta_4}}{\frac{[(p_x + l_3)^2 + p_y^2 + p_z^2 - l_1^2 - l_2^2]}{(2l_1l_2)}} \quad (2.5)$$

$$\theta_5 = \arctan\left(\frac{-p_z}{\sqrt{[(p_x + l_3)^2 + p_y^2]}}\right) - \arctan\left(\frac{l_1 \sin \theta_4}{l_1 \cos \theta_4 + l_2}\right) \quad (2.6)$$

$$\theta_6 = \arctan\left(\frac{p_y}{-p_x - l_3}\right) \quad (2.7)$$

$$\theta_2 = \arctan \left(\frac{-\sqrt{1 - (a_x \sin \theta_6 + a_y \cos \theta_6)^2}}{a_x \sin \theta_6 + a_y \cos \theta_6} \right) \quad (2.8)$$

$$\theta_1 = \arctan \left(\frac{-o_x \sin \theta_6 - o_y \cos \theta_6}{-n_x \sin \theta_6 - n_y \cos \theta_6} \right) \quad (2.9)$$

$$\theta_3 = \arctan \left(\frac{a_z}{a_x \cos \theta_6 - a_y \sin \theta_6} \right) - \pi - \theta_4 - \theta_5 \quad (2.10)$$

Robotun ileri kinematiği Denklem 2.11 kullanılarak hesaplanabilir.

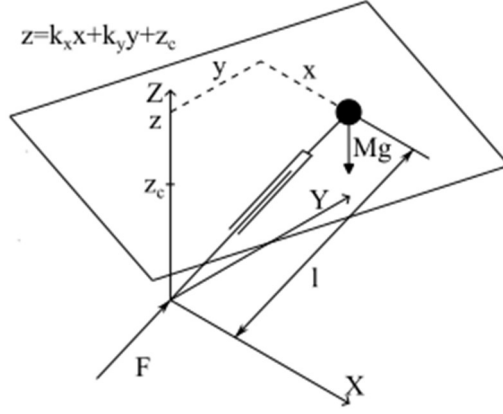
$${}^B_0O_{0R} \cdot {}^{0R}_6T_{6R} \quad (2.11)$$

Bu ifade girilen eklem açılarına göre robotun sabit koordinat sistemindeki oryantasyon ve konum matrisini ifade etmektedir.

2.3 Robot Dinamiği

Tezde geliştirilen yöntem referans kütle merkezi yörüngesini kullanmaktadır. Bu nedenle lineer ters sarkaç yöntemiyle robotun dinamiği çıkarılmıştır. Robotun merdiven çıkıp inebilmesi gerektiğinden dinamik, eğimli yüzeylerde yürüyüş temel alınarak çıkarılmıştır. Sayısal dinamik model MATLAB Multi Body yazılımında hazırlandığından tam dinamik model çıkarılmamıştır. Deneysel modelde ise yalnızca ayaklarda yük hücreleri ve gövdede atalet ölçüm sensörü bulunduğundan basitleştirilmiş dinamik model kullanılmıştır. Robotun ağırlık merkezinin yüksekliği sabit olduğundan, dizlerini bükerek yürüyeceğinden ve gövde tek parça olduğundan dolayı açısal momentum hesaba katılmamıştır (Kajita et al., 2014).

Çıkılacak merdivende iki adım arasındaki yükseklik ve adım uzunluğu kullanılarak eğim açısı elde edilir. Kütle merkezi bu eğimle hareket edecektir. Ayak yörüngeleri de merdiven basamaklarına takılmayacak şekilde oluşturulduğunda robot merdiven çıkabilecektir. Aynı işlem merdiven inilirken de yapılır. Bu defa eğim negatif olmaktadır. Şekil 2.3’de gösterilen 3 boyutlu lineer ters sarkaç modelinde (3B-LTSM). k_x yerine merdiven boyutlarından elde edilen eğim ve $k_y = 0$ konulduğunda istenen yüzey denklemi elde edilmektedir.

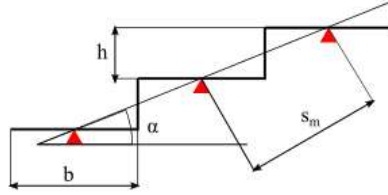


Şekil 2.3: 3 Boyutlu Lineer ters sarkaç modeli (3BLTSM).

Bu düzenlemeler yapıldıktan sonra sistemi tanımlayan diferensiyel Denklem 2.12 ve 2.13'teki gibi olur.

$$\ddot{x} = \frac{g}{z_c + k_x x} x \quad (2.12)$$

$$\ddot{y} = \frac{g}{z} y \quad (2.13)$$



Şekil 2.4: Merdiven çıkıp inmede kullanılacak parametreler.

Adım uzunluğu b , yüksekliği h olan bir merdivende k_x katsayısı Denklem 2.14 kullanılarak hesaplanır.

$$k_x = \tan(\alpha) = \frac{h}{b} \quad (2.14)$$

Merdiven inip çıkma işleminde adım uzunluğu da değişmektedir. Düz yüzeyde adım uzunluğu s_x olurken, merdiven çıkma işleminde yüzey eğiminden dolayı adım uzamaktadır. Yeni adım miktarı 2.15'te verilmiştir.

$$s_m = \frac{s_x}{\cos(\alpha)} \quad (2.15)$$

Çalışma kapsamında, merdiven çıkıp-inme ve düz yüzeylerde yürüme işlemlerinde kullanılmak üzere yalnızca adım konumlarını kullanarak kütle merkezi yörüngesini oluşturan bir program yazılmıştır. Adım konumları Denklem 2.16’da verilen p şeklinde bir matris halinde girilir. Çıktı olarak kütle merkezi yörüngesi alınır. Program öz kodu Çizelge 2.4’de verilmiştir.

$$p = \begin{bmatrix} S_{x0} & S_{x1} & S_{x(n-1)} & S_{xn} \\ S_{y0} & S_{y1} & \dots & S_{y(n-1)} & S_{yn} \\ S_{z0} & S_{z1} & S_{z(n-1)} & S_{zn} \end{bmatrix} \quad (2.16)$$

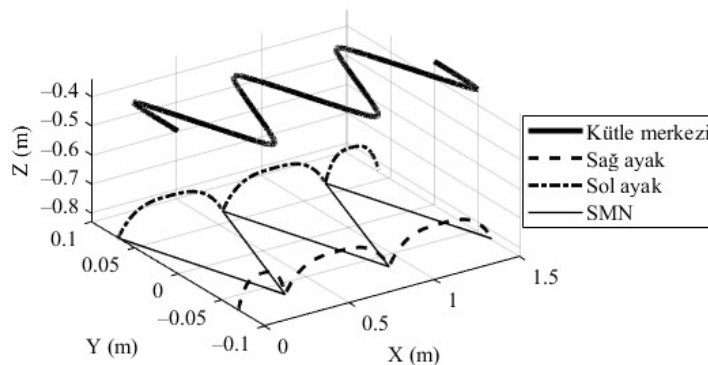
Çizelge 2.4: Merdiven ve düz yüzeyler için yörünge oluşturma programı öz kodu.

```

1 : Adım konumlarını gir,p
2 : For bütün adım konumları için
3 : İki adım arasındaki yükseklik farkını hesapla,h_fark
4 : If h_fark>0 || h_fark<0
5 : Adım uzunluğunu güncelle
6 : Eğimli kütle merkezi yörüngesini oluştur
7 : Merdiven için ayak yörüngesi oluştur
8 : Else
9 : Düz yüzey için kütle merkezi yörüngesi oluştur
10 : Düz yüzey için ayak yörüngesi oluştur
11 : End if
12 : End for

```

Bu denklemler kullanılarak adım uzunluğu 0.3m, adım genişliği 0.065m ve kütle merkezi yüksekliği 0.45m için kütle merkezi yörüngesi Şekil 2.5’teki gibi elde edilir. Ayak konumu referans olarak alındığında robot kalçasının bu durumdaki yüksekliği 0.826m’dir.



Şekil 2.5: Düzlemde yürüme için kütle merkezi yörüngesi ve sabit ayak konumları.

Robot sağ adımını atarak harekete başlamaktadır. Toplam 6 adım yürüyüp durmaktadır.

$${}^0T_n = \prod_{n=1}^6 {}^{n-1}T_n \quad (2.19)$$

$$\mathbf{c}_n = {}^0T_n \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} \quad (2.20)$$

Robot sağ ayak, sol ayak ve gövdeden meydana gelmiştir. Sembolik olarak kütle merkezi konumları Şekil 2.2 (b)'de gösterilmiştir. Algoritmada robot kütle merkezi Şekil 2.2 (b)'de gösterilen kütle merkezi konumları kullanılarak Denklem 2.21 ile hesaplanmıştır.

$$\mathbf{c}_B = \frac{\sum_{n=1}^6 (m_n \mathbf{c}_n)_R + \sum_{n=1}^6 (m_n \mathbf{c}_n)_L + m_T \mathbf{c}_T}{m_R + m_L + m_T} \quad (2.21)$$

2.5 Kütle merkezi Hesap Yöntemi

Kütle merkezinin konumunu hassas olarak elde edebilmek için önce mevcut durumdaki konumlar kullanılarak eklem açıları hesaplanır. Açılar kullanılarak robotun kütle merkezi hesaplanır. Mevcut kütle merkezinden referans kütle merkezi konumuna olan vektör Denklem 2.22 kullanılarak belirlenir.

$$(G_{hata})_n = c_B - (c_{B_{robot}})_{n-1} \quad (2.22)$$

Bu vektörün birim vektörünü hesaplamak için Denklem 2.23 kullanılır.

$$(\hat{G}_{hata})_n = \frac{(G_{hata})_n}{|(G_{hata})_n|} \quad (2.23)$$

Yerle temas halindeki ayağın bağlı olduğu kalça konumu, Denklem 2.24'ün kullanılmasıyla, birim vektörü belirli bir β katsayısıyla çarparak, bu çalışmada (0.001), kalça konumu yeni bir konuma hareket ettirilir.

$$({}^0O_3)_n = ({}^0O_3)_{n-1} + \beta \cdot (\hat{G}_{hata})_n \quad (2.24)$$

Hareketli ayağın bağlı olduğu kalça da bu harekete uygun olarak hareket ettirilir. Eklem açıları ve robot kütle merkezi yeni kalça konumlarına göre yeniden hesaplanır. Referans ve mevcut kütle merkezi arasındaki fark istenen değerden küçük oluncaya

kadar bu işlem devam eder. Bu tezde hata 0.001m olarak belirlenmiştir. Hesap önce sabit ayak için yapılır. Daha sonra sabit ve hareketli ayakların indisleri değiştirilerek hareketli ayak sabit kabul edilerek kalçalar hareket ettirilir. Böylece robotun bütün uzuvlarının kütlelerinden ve konumlarından faydalanılmış olur.

Algoritmanın öz kodu Çizelge 2.5’de verilmiştir.

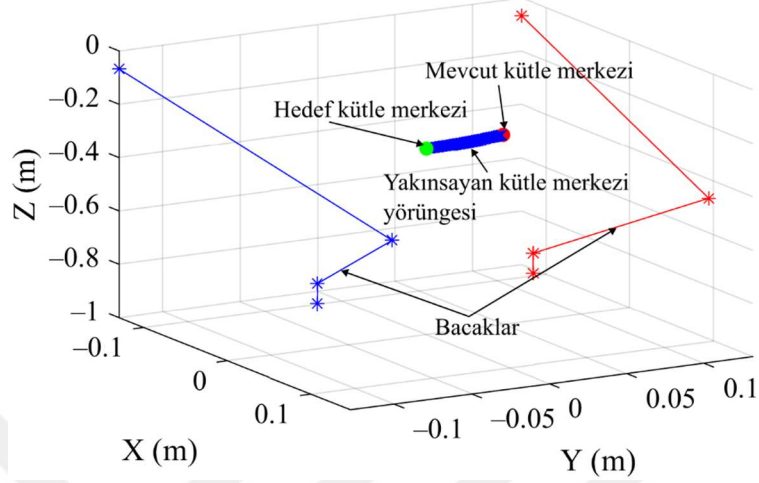
Çizelge 2.5: Yakınsayan Kütle Merkezi (YKM) algoritması.

1	:	sabit $\leftarrow 1$	
2	:	hareketli $\leftarrow 2$	
3	:	For each nokta in AgMerYor	
4	:	$G_{ref} \leftarrow \text{nokta}(1)$	
5	:	$p_{feet} \leftarrow \text{nokta}(2)$	
6	:	$G_{hata} \leftarrow 50$	
7	:	While $G_{hata} > \epsilon$	
8	:	c_B	(Denklem 2.21)
9	:	G_{hata}	(Denklem 2.22)
10	:	$\hat{G}_{hata}$	(Denklem 2.23)
11	:	$\left({}^0O_{3sabit} \right)_n = \left({}^0O_{3sabit} \right)_{n-1} + a. \left(\hat{G}_{hata} \right)_n$	(Denklem 2.24)
12	:	$\left({}^0O_{3hareketli} \right)_n = \left({}^0O_{3sabit} \right)_n - (-1)^{sabit} 2y_0$	
13	:	Eklem açılarını hesapla	(Denklem 2.5-2.10)
14	:	c_B	(Denklem 2.21)
15	:	G_{hata}	(Denklem 2.22)
16	:	If $G_{hata} < \epsilon$	
17	:	Eklem açılarını kaydet	
18	:	Else	
19	:	${}^B O_{0R} \cdot {}^{0R} T_{6R}$	(Denklem 2.11)
20	:	Döngüye devam et	
21	:	End if	
22	:	gecici $\leftarrow j$	
23	:	$j \leftarrow i$	
24	:	$i \leftarrow \text{gecici}$	
25	:	End while	
26	:	End for	

Bu yöntemde önce referans ve mevcut kütle merkezi konumlarındaki arasındaki hata hesaplanır. Eksen takımları Şekil 2.2 (a)’ya göre kullanılmıştır. Gövdeler arasındaki temasta Coulomb sürtünmesi ve kütle-sönümleyici yay modeli kullanılmaktadır.

Şekil 2.8’de robot bacakları grafiksel olarak gösterilmiştir. Sağ bacak mavi renkle, sol bacak kırmızı renkle gösterilmiştir. Yörünge hesabı başlangıcındaki kütle merkezi, mevcut kütle merkezi olarak isimlendirilmiş kırmızı daire simgesiyle gösterilmiştir.

Referans kütle merkezi konumu ise hedef olarak isimlendirilmiş ve yeşil daire simgesiyle temsil edilmektedir. Bu algoritma “Yakınsayan Kütle Merkezi Farkı (YKM)” algoritması olarak adlandırılacaktır.

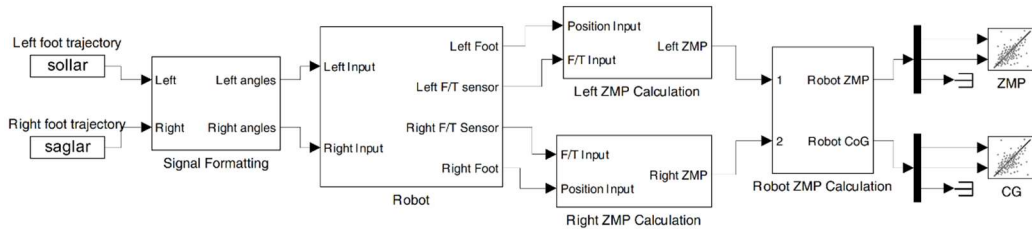


Şekil 2.8: Robot ve kütle merkezinin grafik gösterimi.

Çizelge 2.5’de verilen algoritmanın çalışma sonucu Şekil 2.8’de görülmektedir. Algoritma yörünge üzerindeki bütün değerler için çalıştırılarak yürüyüş yörüngesi elde edilir.

2.6 Dinamik Model

Bu algoritmayla hesaplanan uzuv açılarının robota giriş olarak verildiği bir dinamik model MATLAB Simulink MultiBody ürününde Şekil 2.9’deki gibi oluşturulmuştur.



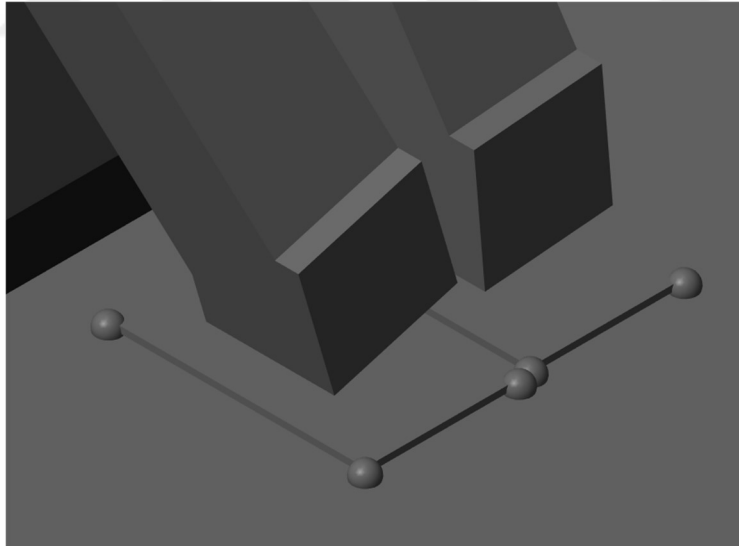
Şekil 2.9: Simulink model diyagramı.

Çalışmada kullanılan parametreler Çizelge 2.6’da verilmiştir.

Çizelge 2.6: Çalışmada kullanılan parametreler.

Parametre	Büyüklik	Birim	Açıklama
s_x	0.3	m	Adım uzunluğu
s_y	0.065	m	Ayaklar arasındaki açıklık
$h_{adım}$	0.075	m	Adım yüksekliği
r_{kure}	0.01	m	Temas küresi yarıçapı
k_{temas}	60,000	N/m	Temas rijitliği
b_{temas}	6,000	Ns/m	Temas sönüm oranı
G_{has}	0.001	m	Hesaplama hassasiyeti
β	0.001		Birim vektör şiddet çarpanı

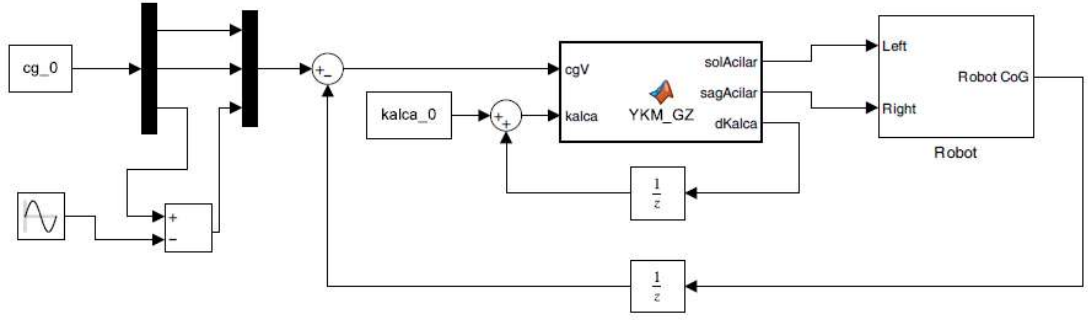
Yerle olan teması tanımlamak için Simulink MultiBody yazılımında küresel ve düzlemsel yüzeyler arasında temas tanımlama özelliği kullanılmıştır. Temas yalnızca küresel yüzeylerle düzlemsel yüzeyler arasında tanımlanabilmektedir. Bu nedenle ayakların 4 köşesine temas küreleri yerleştirilmiştir. Küreler Şekil 2.10’da görülmektedir. Robot yürümesinde ayak konumları önceden bellidir. Simülasyonda kütle merkezini elde etmek için MATLAB Multibody ürününde mekanizma kütle merkezini ölçmek için “Inertia Sensor” elemanı kullanılmıştır.



Şekil 2.10: Ayak uç noktalarına yerleştirilmiş temas küreleri.

2.7 Denetim Sistemi

Robotun denetim sistemi denetleyici olarak YKM algoritmasının kullanılacağı şekilde oluşturulmuştur. Sistemin blok diyagramı Şekil 2.11’de görülmektedir.

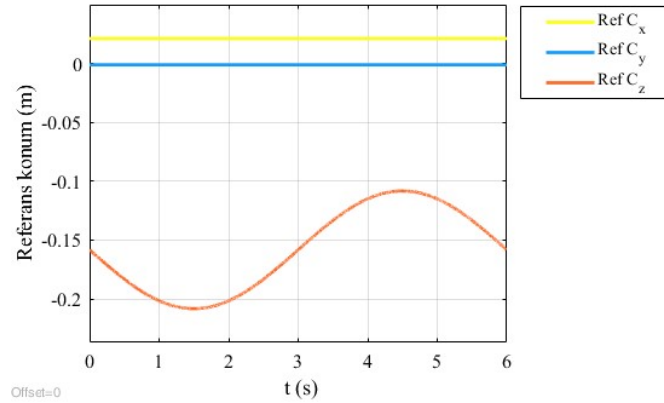


Şekil 2.11: YKM gerçek zamanlı kapalı çevrim denetim sistemi.

Şekil 2.9’da görülen sistemden farklı olarak kütle merkezi çıkışı geri besleme olarak sistemin girişine verilmiştir. Girişte referans kütle merkezi tanımlanmıştır.

Referans kütle merkezi yörüngesi Şekil 2.12’de görülmektedir. Robotun Z eksenindeki hareketi sinüzoidal bir referansla denetlenmek amaçlanmıştır. Kullanılan sinüs dalgasının genliği 0.05 m, frekansı $\pi/3$ olarak belirlenmiştir. Simülasyon süresi 6 saniyedir.

YKM_GZ bloğuna kalçanın $t=0$ s’deki değeri baştan tanımlanmıştır. Kalçanın her hareketinden sonra bu değer güncellenmiştir. Kalça konum değeri ayakla aradaki farkı hesaplamak ve hesap süresinin kısa olması için kullanılmıştır. Cebrik döngü oluşmaması için birim geri beslemelerde birim gecikme kullanılmıştır.



Şekil 2.12: Kapalı çevrim sistem için referans kütle merkezi yörüngesi.



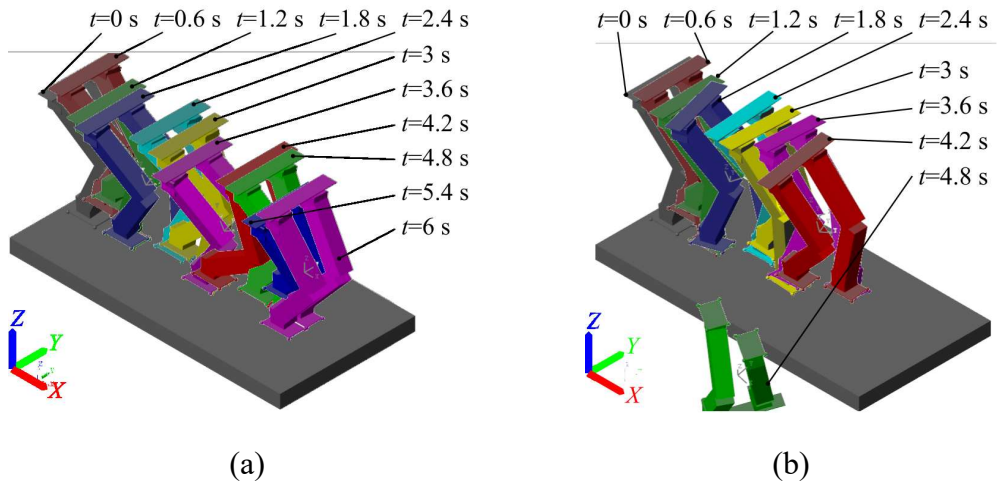
3. BENZETİMLER

3.1 Açık Çevrim Olarak Düzlemde Yürüyüş

Aynı yürüyüş parametrelerini kullanan iki simülasyon yapılmıştır. İlk simülasyonda YKM algoritması kullanılmıştır. Yörüngedeki her konum için eklem açıları YKM kullanılarak hesaplanmıştır.

İkinci simülasyonda robot harekete başlamadan önce kütle merkezi hesaplanmıştır. Kalça ve kütle merkezi arasındaki vektörün sabit kaldığı kabul edilmiştir. YKM algoritması, kabulün yapıldığı (Kajita et al., 2003)'da verilen yöntemle karşılaştırılmıştır. Diğer yöntemler robotun hareketini kısıtladığı ve ek yörüngeler gerektirdiği için kullanılmamıştır. Her iki simülasyon da açık çevrim olarak çalıştırılmıştır. Denetleyici ve dengeleyici kullanılmamıştır.

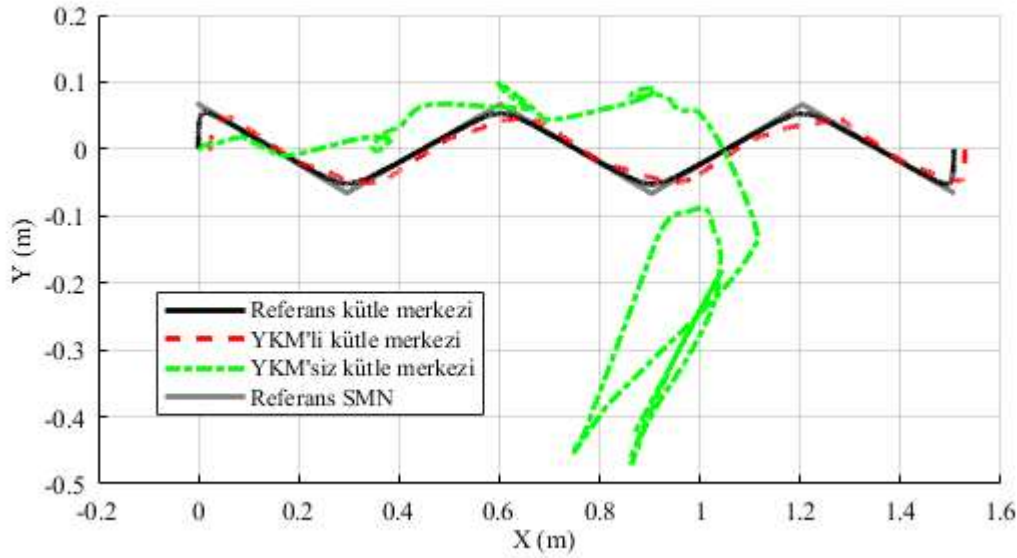
Simülasyon çalışma aralığı, Şekil 3.1'de görüldüğü gibi 10 eşit parçaya bölünmüştür. Her aralığın başlangıcında simülasyonun ekran görüntüsü alınmış ve bir öncekinin üzerine yerleştirilmiştir. Şekil 3.1'deki pozlardan en solda olanı simülasyon başlangıcındaki poza aittir. Robot Şekil 3.1'de kırmızı okla belirtilen pozitif x doğrultusunda ileri doğru hareket etmektedir.



Şekil 3.1: Simülasyon adımları : YKM'li(a), YKM'siz(b).

Şekil 3.1’de YKM’li ve YKM’siz pozlar gösterilmiştir. Şekil 3.1 (a)’da görüldüğü üzere robot düşmeden hareketini tamamlamıştır. Şekil 3.1 (b)’de gösterilen YKM kullanılmadığı durumda ise robot dördüncü adımda dengesini kaybedip düşmektedir. Düşen robot pozu Şekil 3.1 (b)’de ters dönmüş robot pozundan görülebilir.

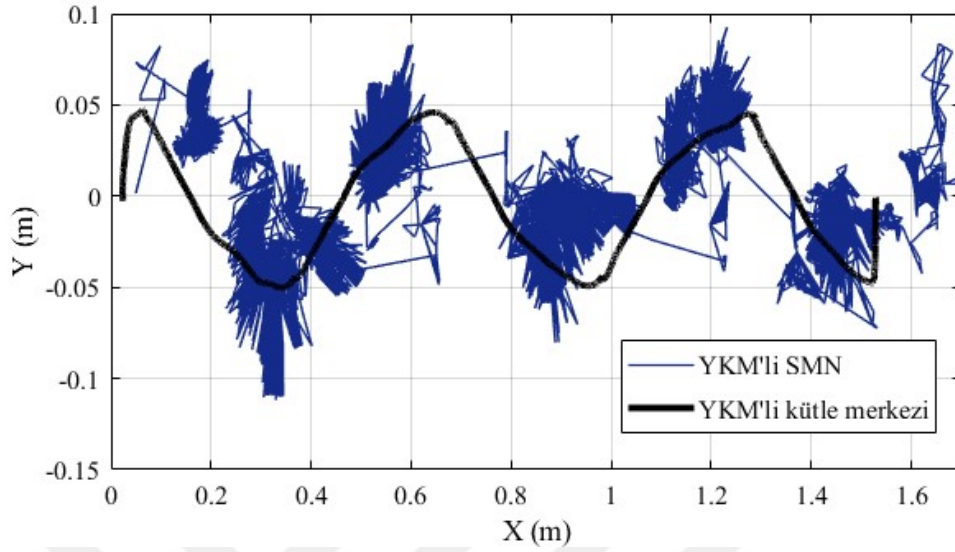
Şekil 3.2 referans ZMP ve kütle merkezi yörüngelerini, YKM’li ve YKM’siz kütle merkezi yörüngelerini bir arada göstermektedir. YKM’li simülasyon sonucu referans ağırlık merkezi yörüngesiyle uyum içindedir. YKM’siz yörünge ise ikinci adımda referans yörüngeden sapmaya başlamakta ve dördüncü adımda robotun düşmesiyle sonuçlanmaktadır. Harekete başlangıç noktaları birbirine çok yakındır. Yerle yalnızca 4 noktadan temas olduğundan ayaklarda kaymalar meydana gelmiştir. X ve Y eksenlerinde meydana gelen sapmalar bu şekilde açıklanabilir.



Şekil 3.2: Hesaplanan ve simüle edilen yörüngeler.

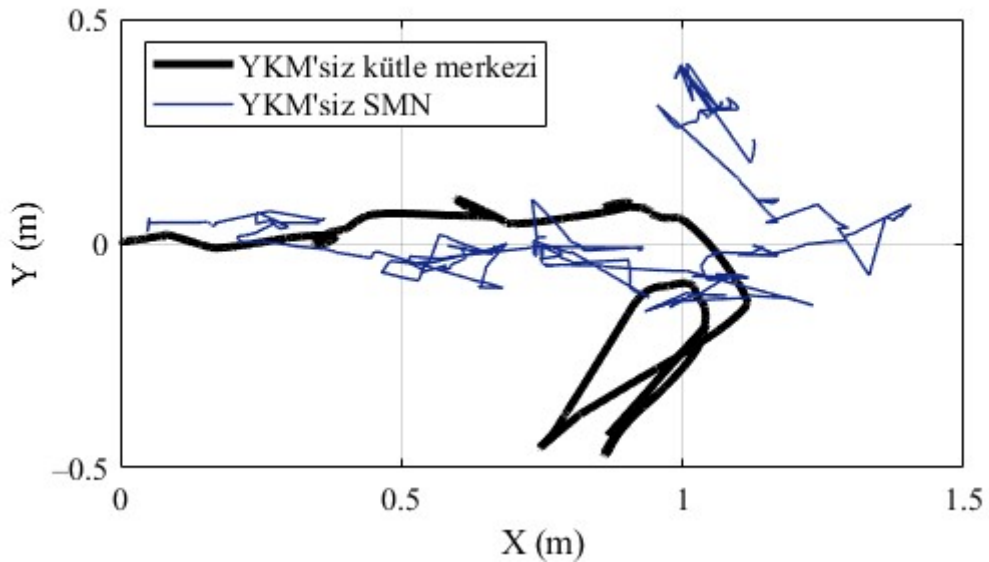
Hesaplama zamanları ayrıca elde edilmiştir. Windows 10 Pro 22H2 yüklü, Intel Core i7 2700 işlemcili, 16GB hafızalı ve 240GB SSD sabit diskli, MATLAB 2023b çalıştıran bir bilgisayarda yörünge için ilk değeri için hesaplama süresi 0.001662 s olarak elde edilmiştir. Takip eden noktaların her biri için bu süre 0.000414 s olarak elde edilmiştir. Bu durum saniyede 2415 kez hesaplama yapılabileceğini göstermektedir.

YKM'li ve YKM'si simülasyonda elde edilen ZMP ve kütle merkezi yörüngeleri ayrı ayrı Şekil 3.3 ve Şekil 3.4'te verilmiştir.



Şekil 3.3: YKM'li ZMP ve kütle merkezi yörüngesi.

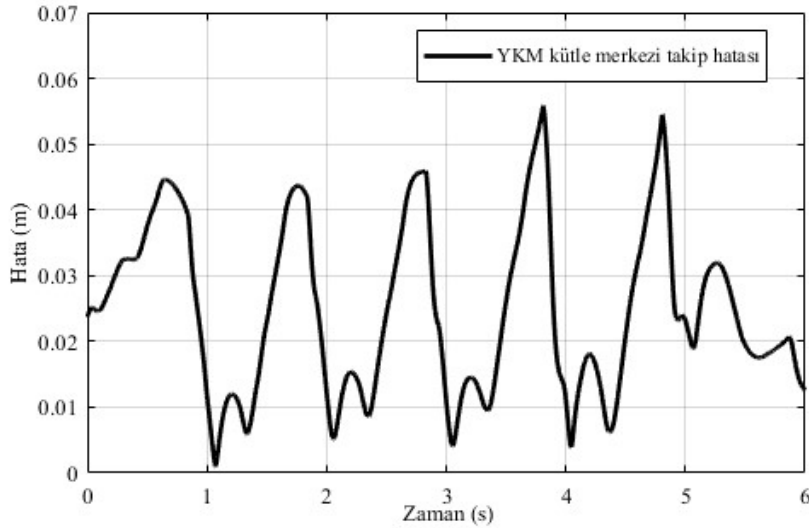
Şekil 3.3 de görülebildiği gibi ZMP ve kütle merkezi yörüngeleri iyi bir uyum içindedir. Temas kuvvetlerindeki titreşim nedeniyle ZMP değerlerinde sapmalar görülmektedir. Bu değerler destek geometrisi içinde kaldığından robot düşmemiş ve yürüyüş hareketini tamamlayabilmiştir.



Şekil 3.4: YKM'siz ZMP ve kütle merkezi yörüngesi.

YKM'siz diğ er durum i in yapılan sim lasyonda, ilk iki adım i in ZMP ve k tle merkezi y r ngeleri uyumludur. Bu uyumun    nc  adımda bozulmasıyla robot d rd nc  adımda y r y    tamamlayamadan d  m   t r.

Ek olarak YKM algoritmasıyla elde edilen k tle merkezi y r ngesinin takip edilmesindeki hata  ekil 3.5'te g sterilmi tir.



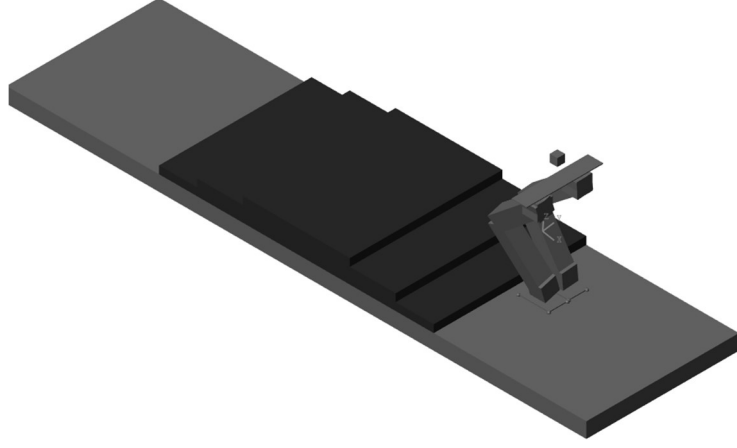
 ekil 3.5: YKM algoritmasının k tle merkezi takip hatası.

 ekil 3.5 hesaplanan referans k tle merkezi y r ngesi ve sim lasyonda elde edilen değ erler arasındaki farkı g stermektedir. En b y k hata 0.0559m, en k   ğ  9.6353e-4 m ve ortalama hata da 0.0238 m olarak meydana gelmi tir.

3.2 A ık  evrim Olarak Merdiven  ıkıp İ me

Merdiven  ıkıp inme i leminde YKM'li y ntem kullanılmı tır. Model i erisinde basamak y ksekliđi 0.075m olan merdiven yerle tirilmi tir. Robot 3 basamak  ıkarak harekete ba lamaktadır. D z alanda 1 tam ve 1 de yarım adım attıktan sonra yine 3 basamak inmektedir. Son olarak 2 ayađını da yan yana getirip durmaktadır. Sim lasyon modeli  ekil 3.6'da g r lmektedir.

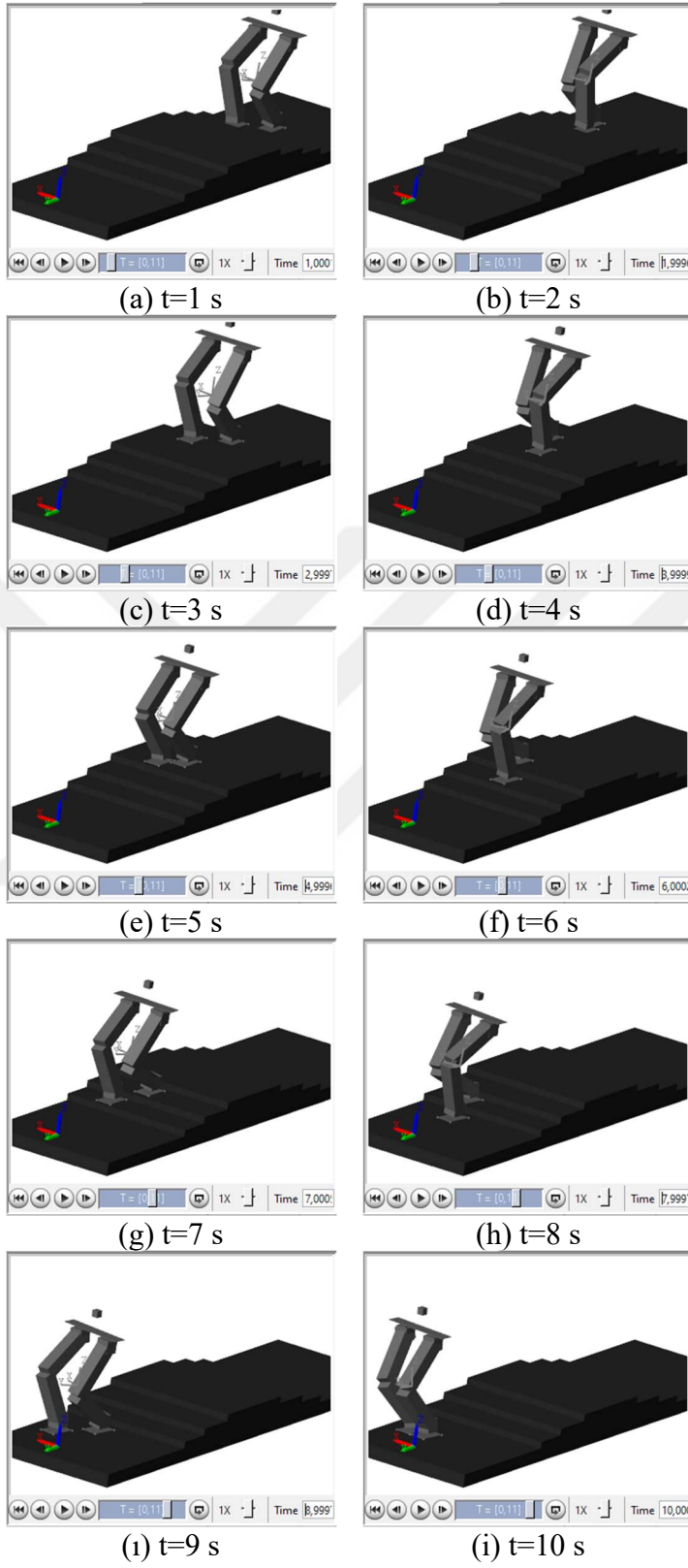
D zlemde y r y   ten farklı olarak aynı temas k relerine zemine ek olarak merdivenin basamaklarını ifade eden 3 farklı g vdeyle daha temas tanımlanmı tır.



Şekil 3.6: Merdiven çıkıp inme simülasyon ortamı $t=0$ s.

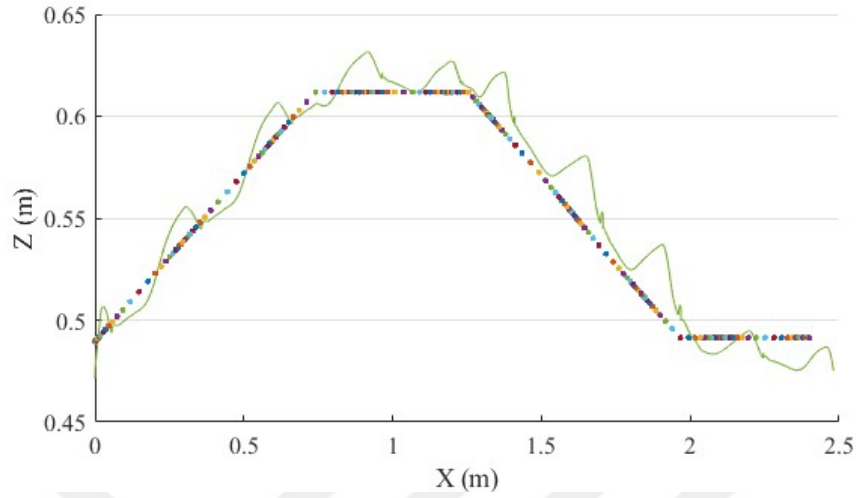
Model içerisinde basamak yüksekliği 0.075m olan merdiven yerleştirilmiştir.

Yürüyüş pozları Şekil 3.7’de görülmektedir. Robot 3 basamak çıkarak harekete başlamıştır. Düz alanda 1 tam ve 1 de yarım adım attıktan sonra yine 3 basamak inmiştir. Son olarak 2 ayağını da yan yana getirip durmuştur.

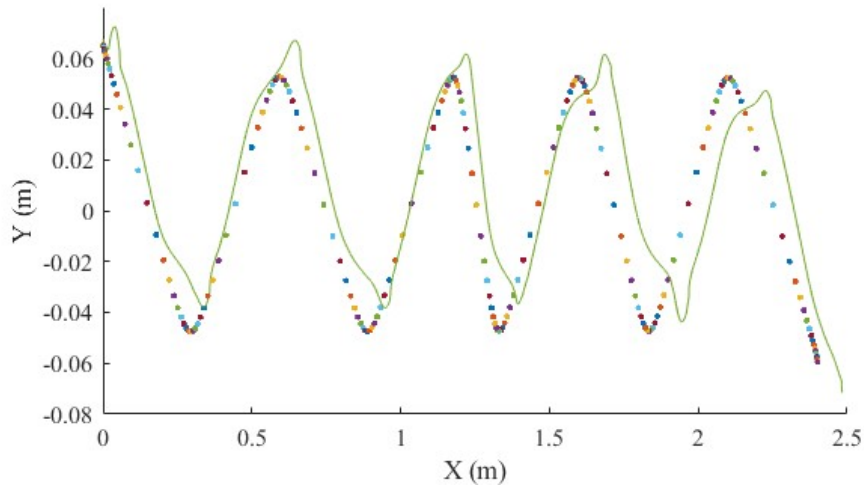


Şekil 3.7: Merdiven çıkıp inme yürüyüş pozları.

Simülasyon sonucunda analitik yöntemle hesaplanan kütle merkeziyle simülasyon sonucu elde edilen kütle merkezi konumları Şekil 3.8 ve Şekil 3.9’da gösterilmiştir.

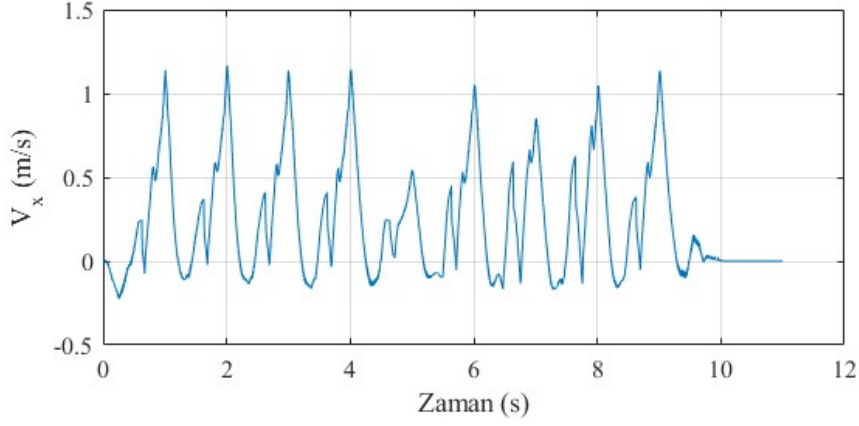


Şekil 3.8: Hesaplanan yörünge (noktalar), simülasyonda oluşan yörünge (çizgi). Yandan görünüş.



Şekil 3.9: Hesaplanan yörünge (noktalar), simülasyonda oluşan yörünge (çizgi). Üstten görünüş.

Şekil 3.8 ve Şekil 3.9’da hesaplanan kütle merkezi yörüngesi ile simülasyonda oluşan yörünge elde edilmiştir. Noktalardan oluşan yörünge hesaplanan, çizgiden oluşan yörünge ise simülasyondan elde edilen kütle merkezi yörüngesini ifade etmektedir. Harekete başlangıç noktaları birbirine çok yakın olmuştur. Ayakların yere temas ettiği anlarda Z doğrultusundaki yörüngede sapma meydana gelmiştir. Yere yalnızca 4 noktadan temas edildiğinden hareket esnasında ayaklarda kaymalar meydana gelmiştir. Bu da Y ve X doğrultusundaki yörüngelerde meydana gelen hataları açıklamaktadır.

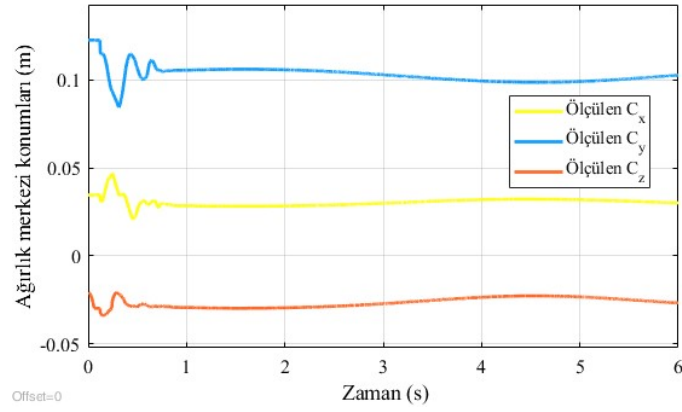


Şekil 3.10: İlerleme hızı.

Şekil 3.10’da ise robotun ilerleme hızı görülmektedir. En yüksek hız 1.1676 m/s, en düşük -0.2252 olarak gerçekleşmiştir. Yürüme hızı ise ortalama değer olan 0.2257 m/s olmuştur. Şekil 3.10’da grafiğin tepe noktaları ayak değiştirme noktalarını göstermektedir. Ters sivri noktalar ise hareketli ayağın yere temas ettiği anları göstermektedir. Negatif hızlar diğer adım için hareketli ayağı hareketlendirme esnasında meydana gelmektedir. 10 saniyede hareket tamamlanmıştır. Düz adım atarken ulaşılan en yüksek hız, merdiven çıkıp inerken ulaşılan en yüksek hızdan daha düşüktür. Eğimli yüzeylerde adım uzunluğu arttığından, daha uzun mesafeyi, düz adımla aynı sürede tamamlamak için kütle merkezinin hızı da artmak zorundadır.

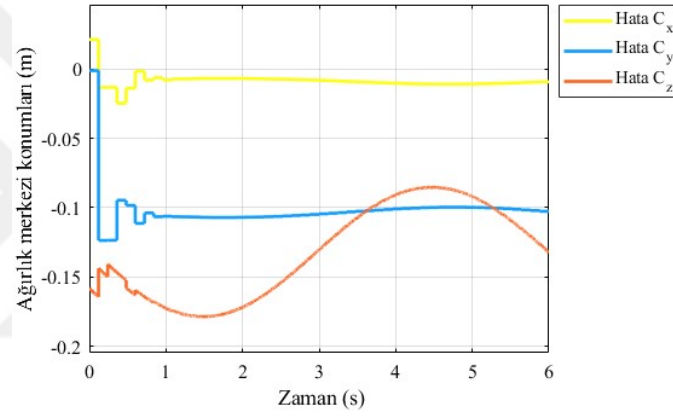
3.3 Kapalı Çevrim Düzlemde Yürüyüş

Referans kütle merkezine göre simülasyon sonucunda oluşan kütle merkezi konumları Şekil 3.11’de gösterilmiştir. Z ekseninde sinüs dalgası biçimi görülmekle beraber girilen referans takip edilememiştir. Grafiklerin başlangıcında değerlerde oluşan titreşimler birim gecikme bloğundan kaynaklanmaktadır. Başlangıç şartının 0 olmasından dolayı simülasyonda uyumlu değerler üretilene kadar değerlerde sapmalar meydana gelmiştir.



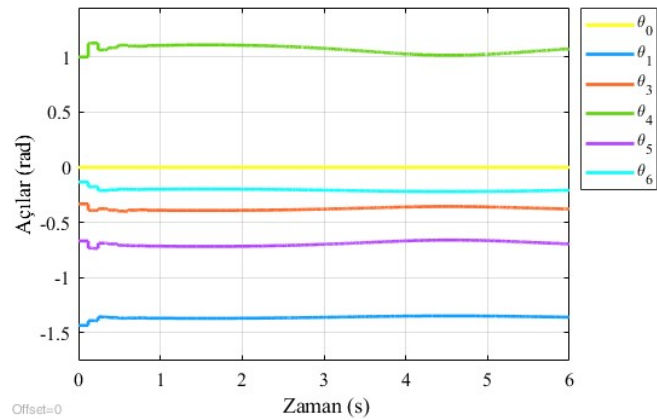
Şekil 3.11: Ölçülen kütle merkezi konumları.

Kütle merkezi takip hatası Şekil 3.12’de verilmiştir. Simülasyon boyunca hatanın düşmediği gözlenmiştir.

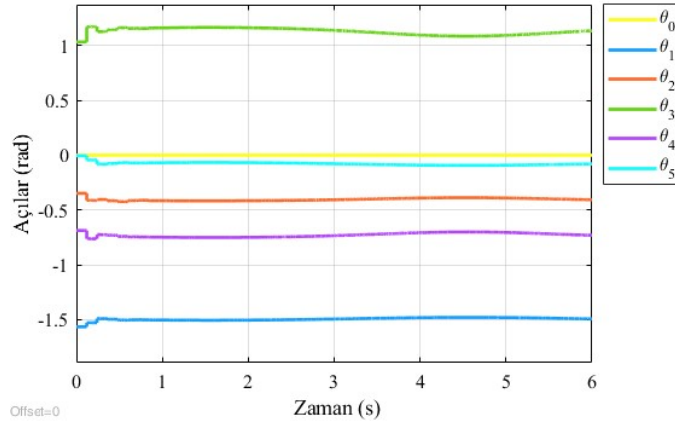


Şekil 3.12: Kütle merkezi takip hatası

Sağ ve sol bacak açı değerleri Şekil 3.13 ve Şekil 3.14’te görülmektedir. Yalnızca kütle merkezinin Z konumu değiştiğinden bacaklardaki hareket özdeş olmuştur.

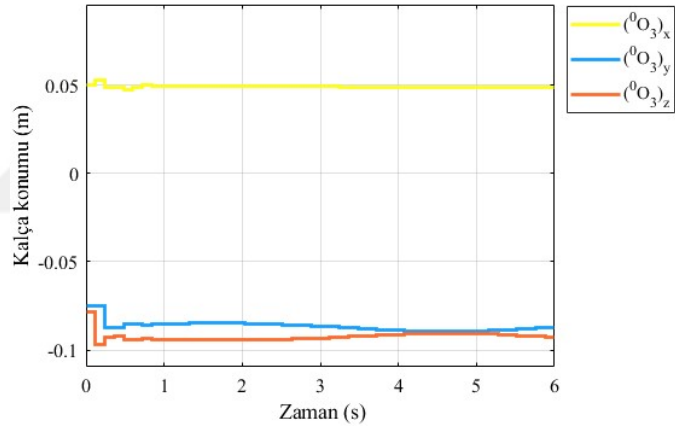


Şekil 3.13: Sağ bacak eklem açı değerleri.

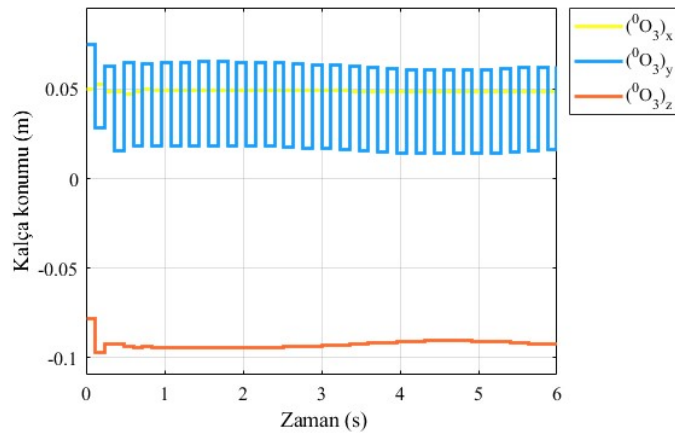


Şekil 3.14: Sol bacak eklem açı değerleri.

Kalça konumlarının değişimi Şekil 3.15 ve Şekil 3.16’da görülmektedir. YKM algoritmasının kalça konumlarını değiştirdiği görülmektedir. Sol kalça y konumundaki dalgalanma, algoritmanın önce sağ kalça konumunu değiştirmesi, sonrasında sol kalçanın konumunu değiştirmesinden kaynaklanmaktadır.



Şekil 3.15: Sağ kalça konum değişimi.



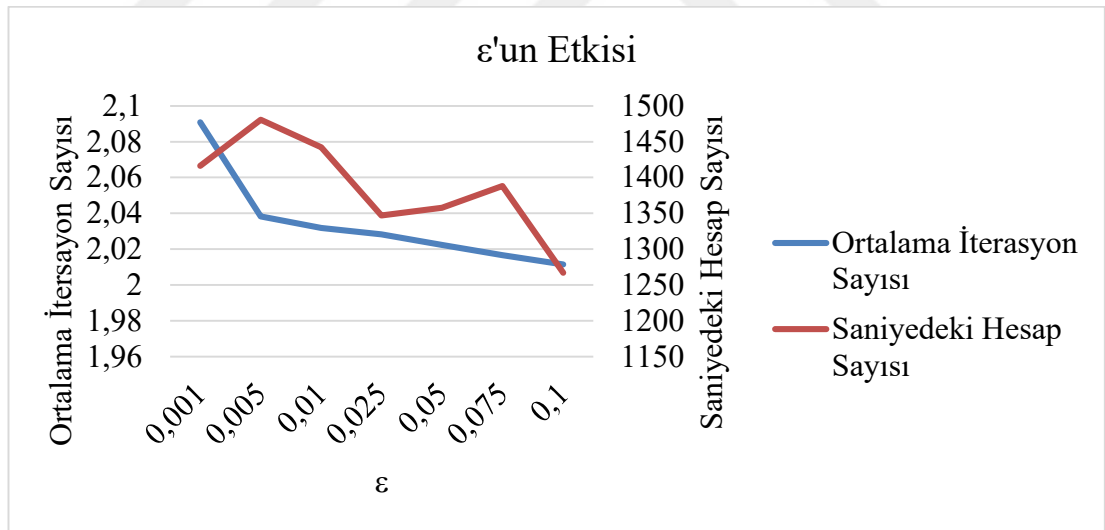
Şekil 3.16: Sol kalça konum değişimi

β katsayısının ve sinüs dalgasının frekansına göre takip ve hata yörüngeleri değişmektedir. β katsayısı büyüdükçe takip performansı iyileşmekte ancak geri beslemedeki birim gecikme nedeniyle robotun hareketleri yüksek ivmeli olmaktadır. Bu da tepki kuvvetlerinin aşırı büyümesine ve robotun düşmesine neden olmaktadır. Ancak β katsayısının büyük olması kalça konumunun bacak tarafından sağlanamayacak bir değere ulaşmasına neden olmakta, bu da eklem açılarının geçersiz değerler almasına neden olmaktadır.

YKM algoritması gerçek zamanlı denetleyici olarak kullanılmaya çalışıldığında β katsayısının kalça konumunun değişmesinden kaynaklanan ivmelerin yüksek olması nedeniyle istenen performansı verememiştir.

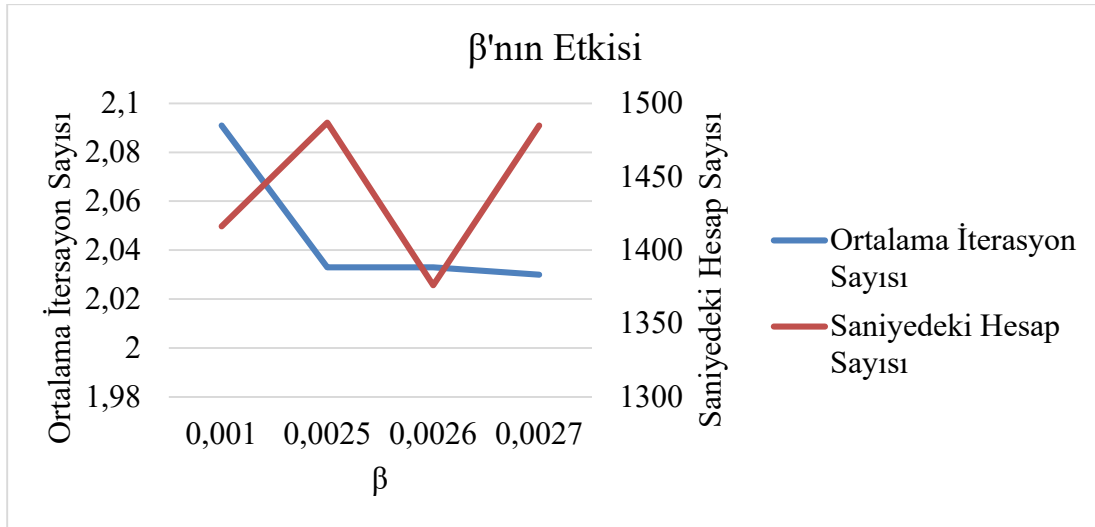
3.4 YKM Hesap Katsayılarının Etkisi

Mevcut kütle merkezi ve hedef kütle merkezi arasındaki farkı ifade eden ε ile kalçanın hareket miktarını ifade eden β katsayılarının etkisi incelenip grafik olarak gösterilmiştir.



Şekil 3.17: Ağırlık merkezi hatasının etkisi

Şekil 3.17’de ağırlık merkezi hatası değişken, kalça hareket miktarı sabit olarak alınmıştır. Ağırlık merkezi hatası 0.001’den 0.1 m’ye kadar değiştirilmiş, kalça hareket miktarı 0.001 m olarak tanımlanmıştır. ε arttıkça ortalama itersayın sayısı azalmıştır. İstenen hata miktarına ulaşmak için daha az itersayın yapılması gerekmiştir.



Şekil 3.18: Kalça hareket miktarının etkisi

Şekil 3.18’de görüldüğü üzere kalça hareket miktarı için kullanılan β katsayısının 0.0027 m’den büyük değerleri için algoritma sonsuz döngüye girmiştir. Bu da kalçanın 0.0027 m’den büyük hareketlerinin kütle merkezi konumunda kullanılan ε katsayısının değeri olan 0.001 m’den daha büyük bir değişiklik meydana getirdiğini göstermektedir.

Saniyede yapılabilecek hesap sayısı ise azalan bir karakter sergilemektedir. İterasyon sayısı azaldıkça her bir iterasyon için gereken sürenin benzer olması ve saniyede yapılabilecek hesap sayısının artması beklenmiş, ancak öyle olmamıştır. Bu durumun MATLAB’ın çalışma algoritmasıyla ilgili olduğu düşünülmektedir.

4. İTÜ BİPED ÜZERİNDE YAPILAN ÇALIŞMALAR

4.1 Robotun Durumu

Robotun elektrik bağlantısı uzun süredir kesik olduğundan sürücü ayarlarının silindiği tespit edilmiştir. Önceki çalışmalar (Acar, 2007; Gerçek, 2011; TAN, 2012; Tasasız, 2011; Yapıcı, 2008) incelenerek ayarlar belirlenip uygulanmıştır.

Mutlak sensörler her okumada farklı bir değer döndürmekte ve kararsız çalışmaktadır.

Sürücüler konum modunda çalıştırılmaktadır. CANBus'la çalışan haberleşme hattı 12 motorun eş zamanlı sürülebilmesini sağlayacak şekilde zaman yörüngelerine uyamamaktadır.

Konumları okumak için kullanılan programda okunacak dosyanın sınırlarının bilinmesi gerekmektedir. Bu sınır her seferinde kaynak kodda ilgili satırı değiştirmek suretiyle ayarlanmaktadır.

Program başlamadan önce sürücülerin algılanması işlemi yapılırken bazen hata alınabilmektedir. Bu hatalar haberleşme veya eşleşme hatası gibi yörüngenin uygulamasında kritik olmayan hatalardır. Ancak hata oluştuğu için yürüme işlemine başlanamamaktadır. Bu hatayı USB ile bağlı PC'deki yüklü programdan sıfırlamak ve sürücüyü yeniden aktifleştirmek gerekmektedir.

Robot gövdesinin orta kısmına kablo geçirmek için delikler açılmıştır. Robot tek ayağını kaldırdığında gövde esnemekte ve eğilmektedir. Bu da robotun ayaklarının birbirine değmesine neden olduğundan yürüme sağlanamamaktadır.

Diz ve bilek eklemlerindeki motor-mil bağlantısı kuvvet bağlı yapılmıştır. Adım periyodu 3 sn'in altına indiğinde bağlantı kopmaktadır.

Bilek vidalı millerinin uçlarındaki rulman bağlantılarında dirsek kullanılmamıştır. Hızlı hareketlerde mil rulmanın içinden kaymaktadır.

Kalça eklemi hareket halinde radyal doğrultuda sarsılmaktadır.

4.2 Yazılım Güncellemeleri

Mutlak sensörlerden okunan değerlerin kararsız olmasına karşın sürücünden bilgi okuma kapalı çevrimli hale getirilmiştir. Böylece mutlak ve artımlı enkoderlerin okuma hataları düzeltilmiştir. PDO okumada yaşanan çakışmalar önlenmiştir. Öncedeki çalışmalarda yazılan programlara ek olarak yazılan rogramda sürücünün sonlu durum makinesi modellenerek sürücünün aktifleştirilmesi sağlanmıştır. Programın oluşabilecek hataları farkedip sürücüyü hata modundan çıkartıp çalışma moduna geçirebilmesi sağlanmıştır.

Sürücülerin Interpolayon modu aktifleştirilmiştir. 200ms'lik periyorlarla oluşturulan üçüncü derecede yörüngeleri 1ms örnekleme hızıyla robot eklemlerine uygulanmıştır. Bu sayede hareketler yörüngeye uygun hale getirilmiştir. Sürücü yalnızca iki kayıt noktası arasında interpolasyon yapmakta, ara noktalı interpolasyon yapmamaktadır. Bu da robot hareketlerinin sarsıntılı olmasına neden olmaktadır. Şekil 4.1'de eklem yörüngesinin ve açısal hızının ilk hali görülmektedir.

$$\theta(t) = a_1 + a_2t + a_3t^2 + a_4t^3 \quad (4.1)$$

Denklem 4.1, Denklem 4.2-4.5 arasındaki sınır şartlarına göre çözülür.

$$\theta(t_0) = \theta_0 \quad (4.2)$$

$$\theta(t_f) = \theta_f \quad (4.3)$$

$$\dot{\theta}(t_0) = \dot{\theta}_0 \quad (4.4)$$

$$\dot{\theta}(t_f) = \dot{\theta}_f \quad (4.5)$$

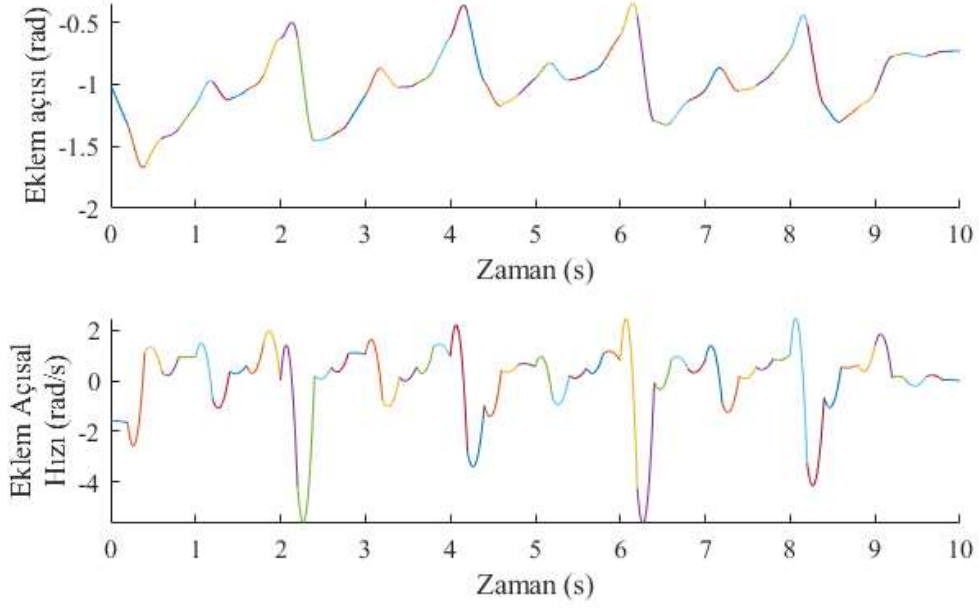
Katsayılar 4.6-4.9 denklemleriyle ifade edilir.

$$a_1 = \frac{t_0^2\theta_f(t_0 - 3t_f) - t_f^2\theta_0(t_f - 3t_0) + (t_f - t_0)(\dot{\theta}_0t_0t_f^2 + \dot{\theta}_ft_0^2t_f)}{(t_0 - t_f)(t_0^2 - 2t_0t_f + t_f^2)} \quad (4.6)$$

$$a_2 = \frac{\dot{\theta}_f(t_0^3 - 2t_0t_f^2 + t_0^2t_f) - \dot{\theta}_0(t_f^3 - 2t_0^2t_f + t_0t_f^2) + 6t_0t_f(\theta_f - \theta_0)}{(t_0 - t_f)(t_0^2 - 2t_0t_f + t_f^2)} \quad (4.7)$$

$$a_3 = \frac{3(t_0 + t_f)(\theta_0 - \theta_f) + \dot{\theta}_f(t_f^2 - 2t_0^2 + t_0t_f) - \dot{\theta}_0(t_0^2 - 2t_f^2 + t_0t_f)}{(t_0 - t_f)(t_0^2 - 2t_0t_f + t_f^2)} \quad (4.8)$$

$$a_4 = -\frac{2(\theta_0 - \theta_f) + (t_f - t_0)(\dot{\theta}_0 + \dot{\theta}_f)}{(t_0 - t_f)(t_0^2 - 2t_0t_f + t_f^2)} \quad (4.9)$$



Şekil 4.1: Parçalı üçüncü dereceden eklem yörüngesi.

Hız grafiğindeki süreksizlikler sarsıntının olduğu noktaları göstermektedir. Bu sarsıntıların oluşmasını engellemek için hızlar sürekli hale getirilmelidir. Bu işlem ara noktalar tanımlanarak yapılmıştır. Yörüngeler ikişerli olarak gruplanmıştır. Her ikişerli yörüngede 2 adet ek parametre tanımlanmıştır. Tanımlanan parametreler Denklem 4.10 ve 4.11’de verilmiştir. Bu denklemler her grup için sırasıyla çalıştırılmıştır. Yörüngeler ikili gruplar halinde ele alındığı için ikili gruplar halindeki yörüngeler sürekli hale gelmiştir. Ancak bunun dışında kalan yörünge gruplarında yine süreksizlik oluşmuştur. Bunu aşmak için yörünge indisleri 1 arttırılarak tekrar gruplandırma yapılmıştır. Bu işlem iki defa yapıldığında eklem yörüngeleri sürekli hale gelmiştir. Sürekli hale getirilmiş yörünge Şekil 4.2’de görülmektedir.

$$\theta(t_v) = \theta_v \quad (4.10)$$

$$\dot{\theta}(t_v) = \dot{\theta}_v \quad (4.11)$$

Ek parametrelerin kullanılmasıyla $t_0 - t_v$ arası ile $t_v - t_f$ arası farklı denklemlerle hesaplanmaktadır. Ek parametrelerle hesaplanan katsayılar Denklem 4.12 ile Denklem 4.19 arasında verilmiştir.

$$a_{0v1} = \theta_0 \quad (4.12)$$

$$a_{0v2} = \dot{\theta}_0 \quad (4.13)$$

$$a_{0v3} = -\frac{9\theta_0 + 3\theta_g - 12\theta_v + 7t_f\dot{\theta}_0 - t_f\dot{\theta}_g}{4t_f^2} \quad (4.14)$$

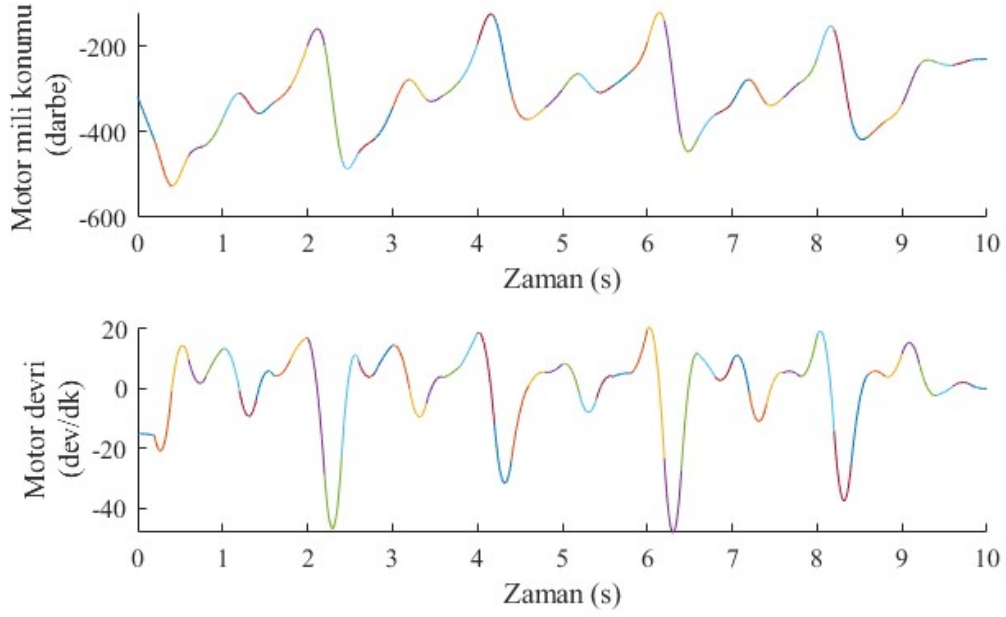
$$a_{0v4} = \frac{5\theta_0 + 3\theta_g - 8\theta_v + 3t_f\dot{\theta}_0 - t_f\dot{\theta}_g}{4t_f^3}; \quad (4.15)$$

$$a_{vf1} = \theta_v \quad (4.16)$$

$$a_{vf2} = -\frac{3\theta_0 - 3\theta_g + t_f\dot{\theta}_0 + t_f\dot{\theta}_g}{4t_f} \quad (4.17)$$

$$a_{vf} = \frac{3\theta_0 + 3\theta_g - 6\theta_v + t_f\dot{\theta}_0 - t_f\dot{\theta}_g}{2t_f^2} \quad (4.18)$$

$$a_{vf4} = -\frac{3\theta_0 + 5\theta_g - 8\theta_v + t_f\dot{\theta}_0 - 3t_f\dot{\theta}_g}{4t_f^3} \quad (4.19)$$



Şekil 4.2: Parçalı üçüncü eklem yörüngesi, sürekli hale getirilmiş hız geçişleri.

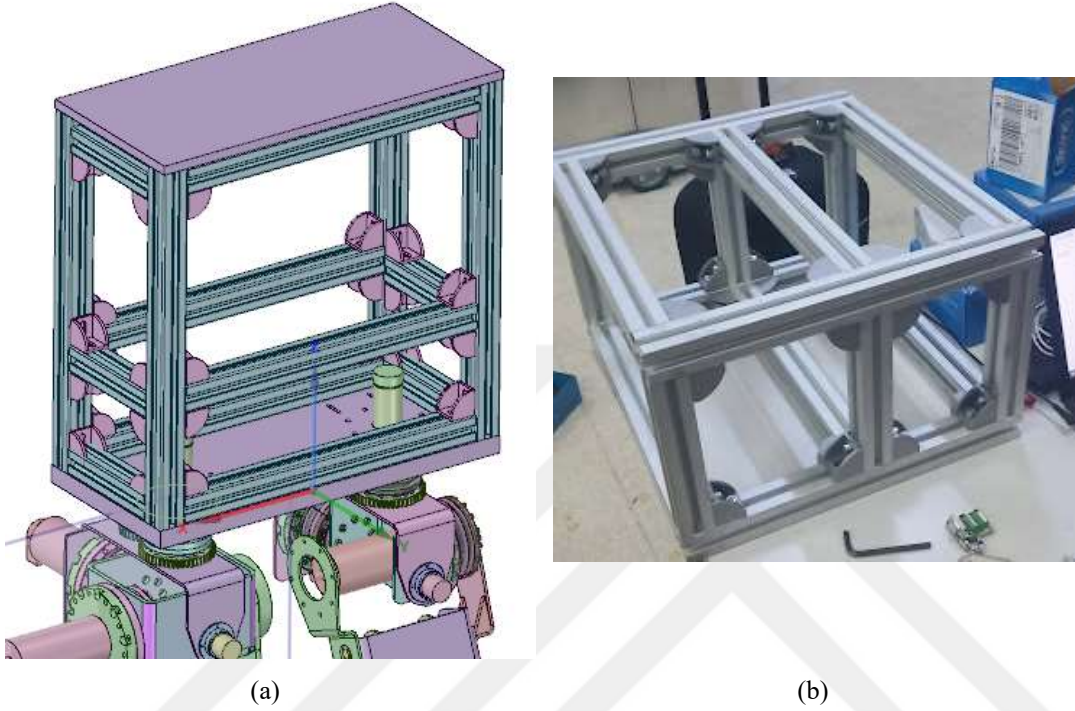
Yörünge, yürüme işlemine başlamadan önce okunarak satır sayısı elde edilmiştir. Elde edilen satır sayısı yürüme döngüsünü bitirecek bir sınırlandırma olarak uygulanmıştır. Böylece satır sayısının önceden bilinmesine gerek kalmamıştır.

Robot sıfıra gittiken sonra, robottan yapması istenen hareketlerin yörüngeleri gönderilmektedir. MATLAB’da elde edilen eklem açılarını motor yörüngelerine çeviren program EK A’da verilmiştir.

Sürücülere en düşük 40ms ayrıık zaman periyodundaki sinyaller gönderilebilmiştir. Daha düşük periyotlarda sürücü önbellekleri bilgisayar tarafından yeterli hızda doldurulamadığı için hata oluşmuştur. 40ms periyodunda sinyal gönderildiği zaman CANBus veriyolunun bant genişliği akım ve konum bilgilerinin okunmasında yetersiz kalmıştır.

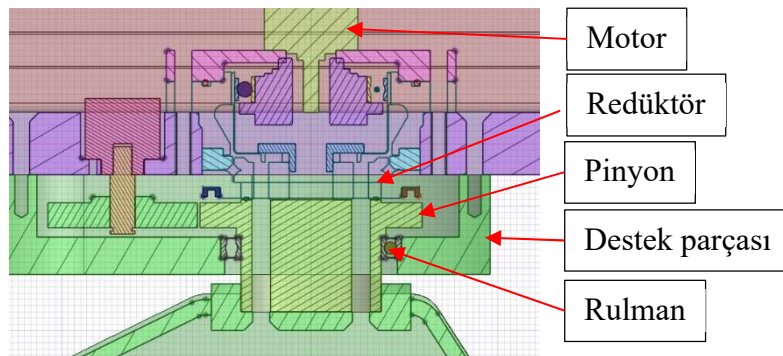
4.3 Mekanik Tasarım Güncellemesi

Rijit bir gövde tasarlanıp imal edilmiştir. Tasarlanan kafes yapı Şekil 4.3'te görülmektedir.



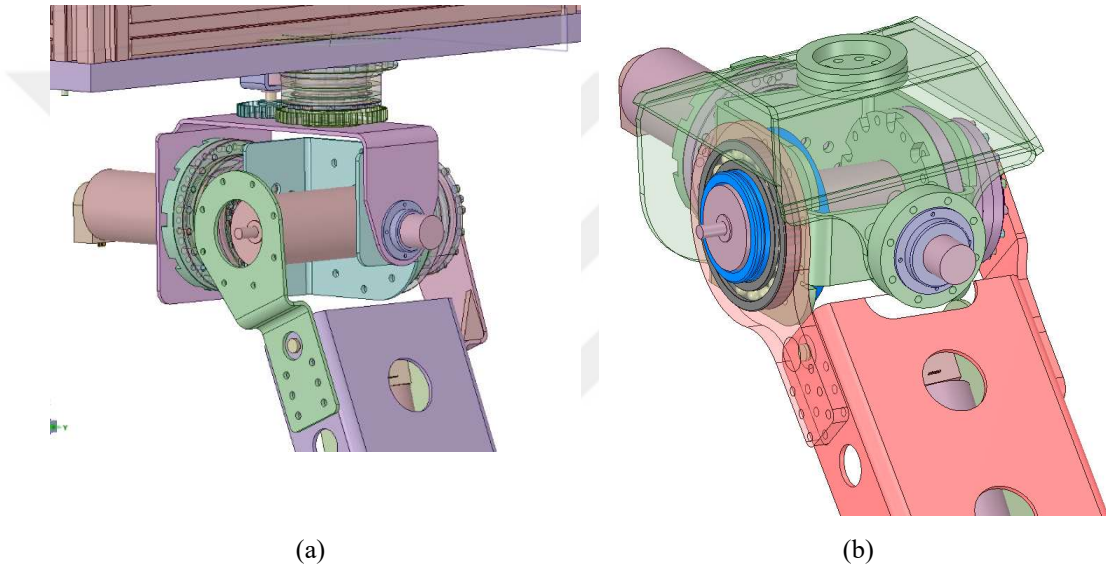
Şekil 4.3: Tasarlanan gövde (a), alt ve üst levhalar hariç monte edilmiş gövde (b).

Kalça dönme eksenini radyal olarak esnemektedir. Bu esnemeyi engellemek için gövdenin alt tarafına civatalarla bağlanan, radyal yükü alan bir parça tasarlanmıştır. Büyük dişli tasarımının da buna bağlı olarak güncellenmesi gerekmiştir. Sol görünüşte, sol ayağın motor ekseninde alınan kesiti Şekil 4.4'te görülmektedir.



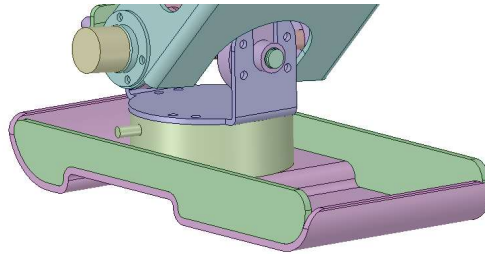
Şekil 4.4: Gövde ek parça detayı.

Kalça yuvarlanma eksenini de adım atma hareketleri yaparken esnemiştir. Şekil 4.5 (a)'da mevcut tasarım, Şekil 4.5 (b)'de güncellenmiş tasarım görülmektedir. Kalça dönme parçasının et kalınlığı 4mm'den 5mm'ye çıkarılmıştır. Sadece enkoder bağlantısı olarak görev yapan kısım yük taşıyacak şekilde yeniden tasarlanmış ve rulmanla yataklanmıştır. Böylece parça konsol kiriş yapısından basit mesnetli bir yapıya dönüştürülmüştür. Mevcut tasarımda motor da yük taşıyan bir eleman olarak tasarlanmışken güncel tasarımda mavi renkli görülen parça ile yük motorun üzerinden alınmıştır. Mevcut tasarımda C yapıda olan parça güncel tasarımda kapalı bir hale getirilip yük taşıma ve rijitliği artırılmıştır.



Şekil 4.5: Mevcut tasarım (a), güncellenmiş tasarım(b).

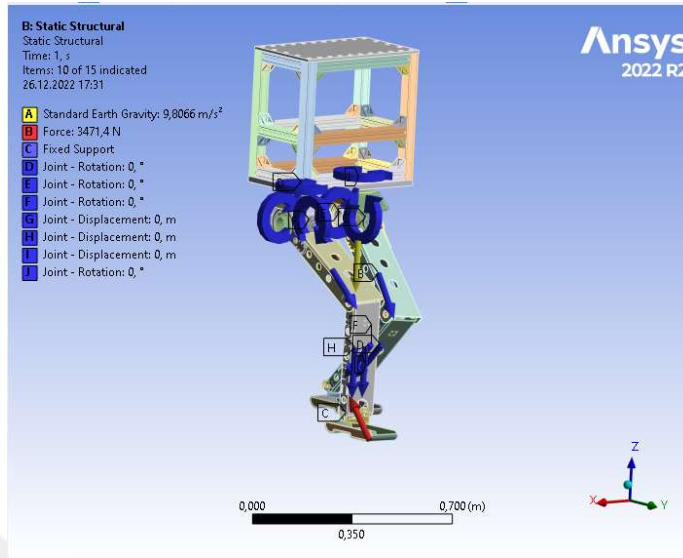
Ayağa yapılan güncelleme Şekil 4.6'da görülmektedir. Yanlara destek parçaları kaynatılarak tasarım güncellenmiştir.



Şekil 4.6: Güncel ayak tasarımı.

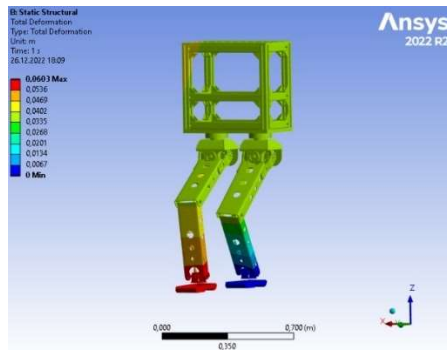
Tasarım güncellendikten sonra doğrulama için sonlu eleman yöntemiyle statik gerilme analizi yapılmıştır. Gerilme analizinde kullanılan yükler ve robot pozı MATLAB MultiBody modülünde geliştirilen simülasyonda elde edilen en yüksek tepki kuvveti

ve eklem torklarının incelenmesi sonucunda elde edilmiştir. Analizin sınır şartları Şekil 4.7’de gösterilmiştir.

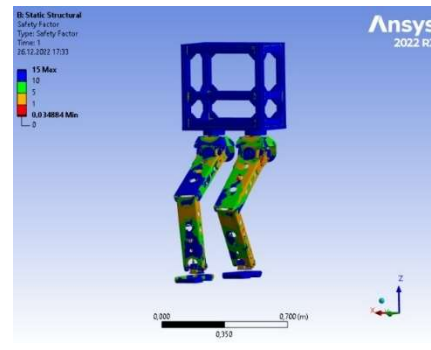


Şekil 4.7: Statik analiz sınır şartları

Şekil 4.8 (a)’ya göre analiz sonucunda meydana gelen şekil değiştirme sınır şartlarını doğrulamaktadır. Emniyet katsayısı hesaplatıldığında oluşan değerler Şekil 4.8 (b)’de görülmektedir. Buna göre emniyet katsayısı 1’den küçük bölgeler bulunmaktadır. Bu modelin emniyetsiz olduğunu gösterir. Ancak gerçekte emniyetsiz olan bölgelerde oluşan gerilme yığılmaları civatalarla taşınacaktır. Modelde civatalı birleştirme yerine “yapışma” sınır şartı uygulanmıştır. Kenarların yüzeylere yapıştığı yerlerde yüksek gerilmeler oluşmuş ve bu bölgeleri emniyetsiz olarak göstermiştir. Yine de gerilme yığılmalarının bütün eleman boyunca yayılması nedeniyle parçalar emniyetsiz olarak değerlendirilmiştir.



(a)



(b)

Şekil 4.8: Şekil değiştirme sonuçları(a), emniyet katsayısı(b).

4.4 Güncelleme Sonuçları

Adım periyodu 3 saniyenin altına indiğinde motorlar istenen hızı sağlayabilmekte fakat, akım ve sıcaklık emniyet değerleri aşılmaktadır. Bu nedenle sürücüler hata moduna geçip motorların hareketini sonlandırmaktadır.

Yine adım periyodu 3 saniyenin altına indiğinde robotun mekanik elemanlarında kayma ve titreşim meydana gelmektedir.

Üretici tarafından gerçek zamanlı PLC kullanılması, konum hız ve zaman değerlerinin bu şekilde gönderilmesi gerektiği belirtilmiştir. Cevabın ilgili bölümü:

For a 12Axis system, you would use usually a realtime CANopen PLC Master system, which is able to do the pathplanning and controlling all the axis.

With CAN bus and 12 slave device, the lowest possible data rate exchange is 4ms. Lower values will lead to an overloaded CANbus.

Mekanik güncellemenin yapılabilmesi için kabloların kesilip bağlantıların yeniden yapılması gerekmektedir.

Elektronik ve yazılım sisteminin güncellenmesi için sürücülerin giriş-çıkış ayarlarının değişmesi, kablolanın yenilenmesi gerekmektedir.

Bu aşamadan sonra yeni bir robot sayısal ortamda tasarlanmış ve üretimi yapılmıştır.

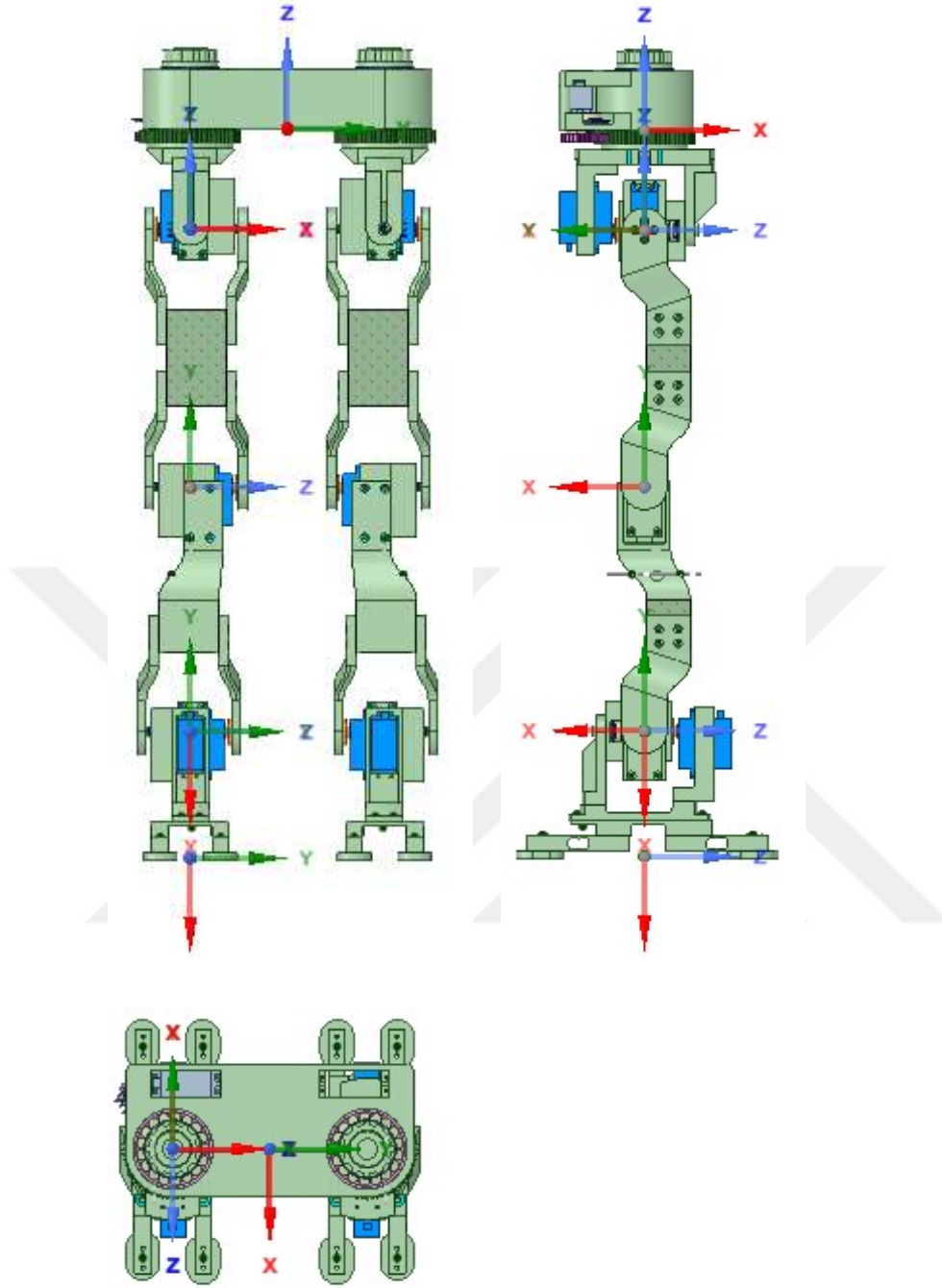


5. YENİ ROBOTUN TASARIM VE ÖZELLİKLERİ

Yeni robot tasarlanırken İTÜ Biped’de yapılması düşünülen değişikliklerin etkisini yansıtabilecek özelliklere sahip olmasına dikkat edilmiştir. Bu nedenle kinematik yapısı aynen korunmuştur. Eklemlere metal redüktörlü servo motorlar yerleştirilmiştir. Bu motorlar harmonik redüktör, DC motor ve artımlı enkoder sistemine analogdur. Eklemlerin mutlak konumu potansiyometreler kullanılarak elde edilmiştir. Ayaklarda 6 eksen tork-kuvvet sensörü yerine 4 noktadan yerle temaslı yük hücresi kullanılmıştır. Ek olarak gövdede atalet ölçüm birimi sensörü (Inertial Measurement Unit – IMU) kullanılmıştır.

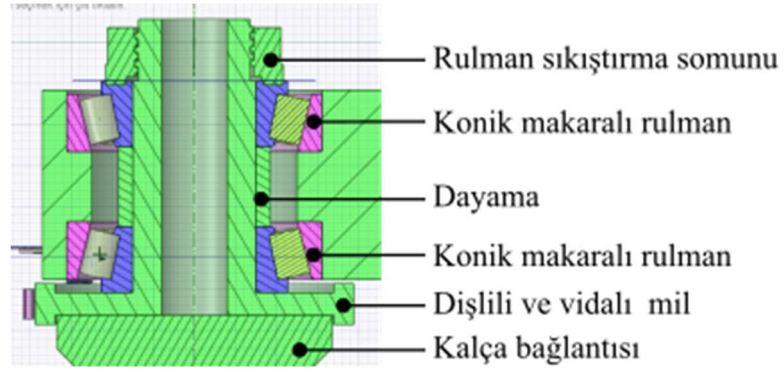
5.1 Mekanik Tasarım

Robotun mekanik tasarımı Şekil 5.1’de görülmektedir. Mekanik tasarım yapılırken kalça rijitliğinin yüksek olmasına ve hafifliğin sağlanması için bağlantıları sağlayan elemanların basit mesnetle yataklanmasına dikkat edilmiştir. Mekanik parçalar Fuse Deposition Molding (FDM) – Eriyik Yığma Modellemesi yöntemini kullanan 3 boyutlu yazıcıyla yazdırılmıştır. Malzeme olarak Polilaktik Asit (PLA) kullanılmıştır. Malzeme dolgusu %12, katman yüksekliği 0.2mm olarak seçilmiştir. Duvar ve unsur çevre kalınlığı 4 katman, alt ve üstte 2 katman yazdırılmıştır. Creality Ender 3D V2 marka model yazıcı kullanılmıştır.

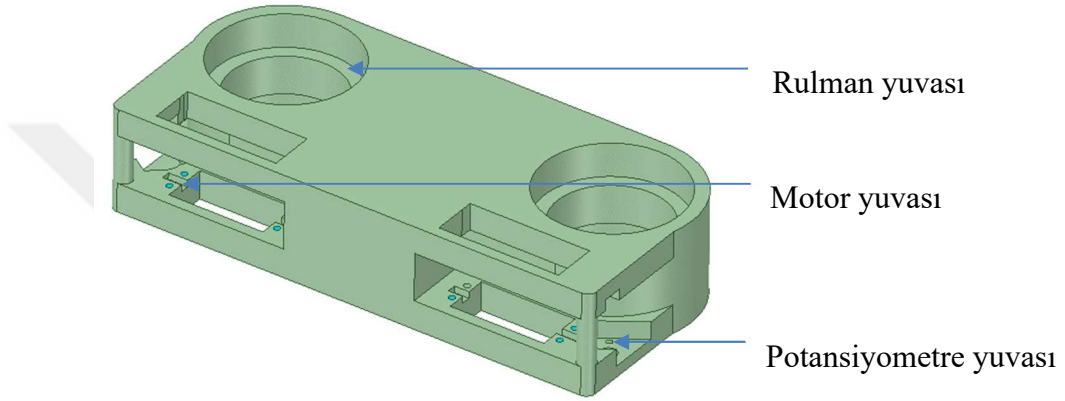


Şekil 5.1: Robotun üç görünüşü ve eklemlerdeki eksen takımlarının yerleşimi.

Kalça ekleminin gövdeye bağlantısı Şekil 5.2’de görülmektedir. Bu tasarımda ayağın kalçaya olan bağlantısı X düzeninde moment taşıyacak şekilde mesnetlenmiştir. Bacaklar tek parçadan oluşan bir gövdeye bağlanmıştır (Şekil 5.3).

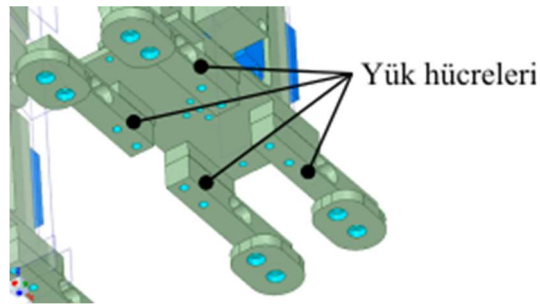


Şekil 5.2: Ayak gövde mesneti montaj kesiti.



Şekil 5.3: Tek parça gövde.

Yere 4 noktadan temas eden yeni ayak tasarımı Şekil 5.4’de görülmektedir. Dinamik simülasyonda en yüksek temas kuvveti 70N olarak elde edilmiştir. Bu nedenle 10kg kapasiteli yük hücreleri kullanılmıştır.



Şekil 5.4: 4 noktadan temaslı ayak tasarımı.

5.1.1 Mekanik sınırlar

Kullanılan servo motorlar 180° hareket edebilmektedir. Mekanik tasarım yapılırken insan uvuzlarının hareket sınırlarına yakın veya daha fazla hareket edebilmesine dikkat edilmiştir. Uzuvarların hareket sınırları Çizelge 5.1’de gösterilmiştir.

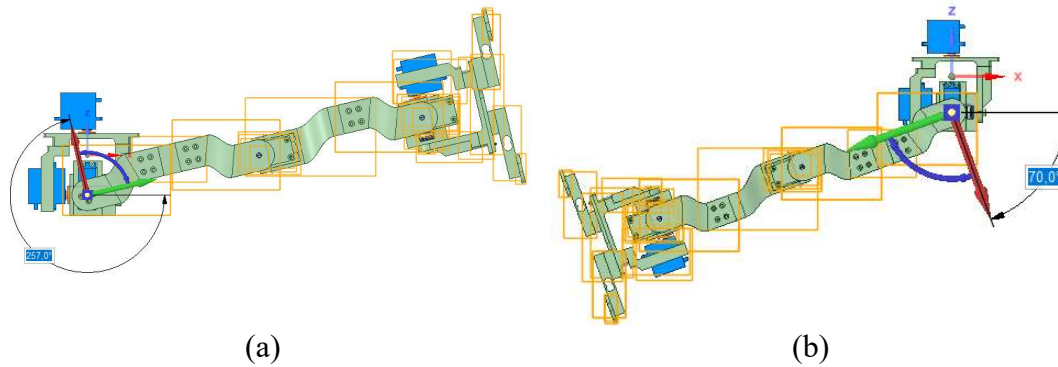
Çizelge 5.1: Uzuvarların hareket sınırları.

Eklem adı	Alt sınır	Üst sınır
Kalça dönme	-30	30
Kalça yalpalama	-55	103
Kalça yunuslama	-70	167
Diz	-15	215
Bilek yunuslama	-60	153
Bilek yalpalama	-90	90

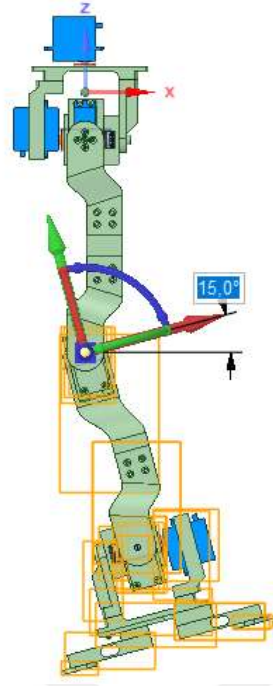
Bacakların birbirlerine ve kendilerine çarpmadan yapabileceği hareket sınırları Şekil 5.5 ile Şekil 5.9 arasındaki şekillerde gösterilmiştir.



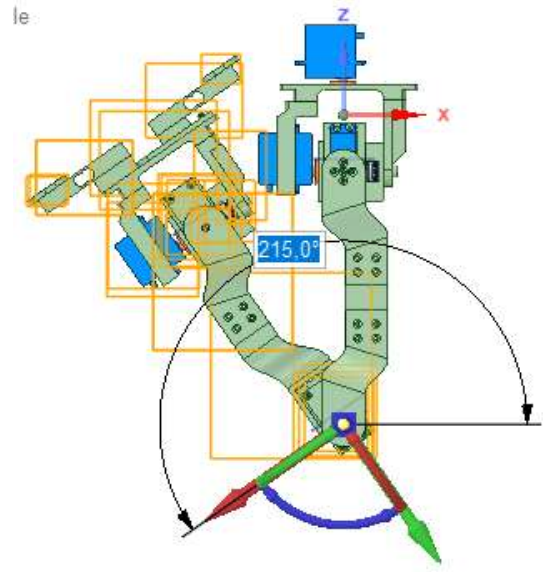
Şekil 5.5: Ön görünüş kalça yalpalama hareket sınırları üst sınır(a); alt sınır(b).



Şekil 5.6: Sol görünüş kalça yunuslama hareket sınırları üst sınır(a); alt sınır(b).

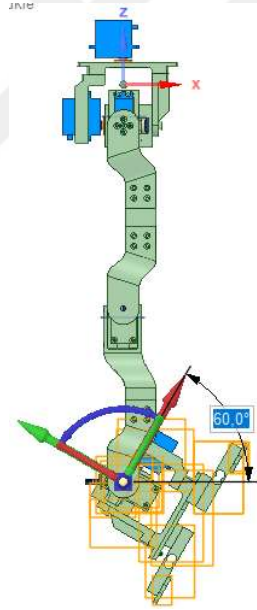


(a)

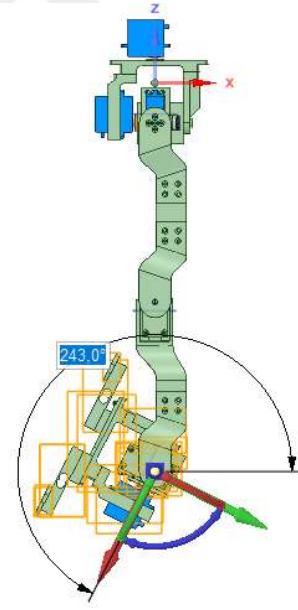


(b)

Şekil 5.7: Sol görünüş diz hareket sınırları üst sınır(a); alt sınır(b).

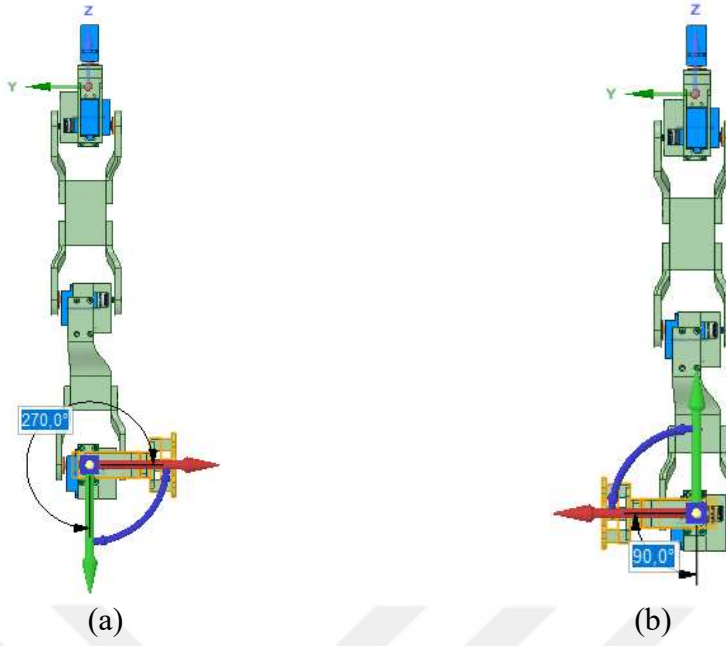


(a)



(b)

Şekil 5.8: Sol görünüş bilek yunuslama hareket sınırları üst sınır(a); alt sınır(b).



Şekil 5.9: Arka görünüş bilek yalpalama hareket sınırları üst sınır(a); alt sınır(b).

Dönüşüm tablosunda yer almayan $y_0 = 0.075m$ ve $z_0 = 0.078068m$ değerindedir. Robotun yeni link dönüşüm matrisi Çizelge 5.2’de verilmiştir.

Çizelge 5.2: Denavit-Hertenberg yöntemine göre link dönüşüm matrisi.

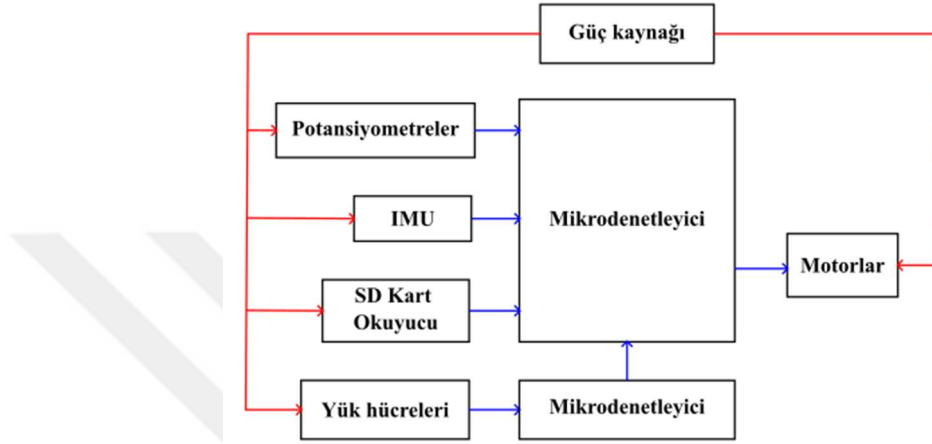
Eklem no	θ	d	a	α
1	θ_0	0	0	$\pi/2$
2	θ_1	0	0	$-\pi/2$
3	θ_2	0	0.2	0
4	θ_3	0	0.19	0
5	θ_4	0	0	$\pi/2$
6	θ_5	0	0.0980368	0

5.2 Elektronik Donanım

Elektronik donanım seçilirken parçaların yaygın olarak bulunabilmesi, kütüphane desteklerinin olması dikkate alınmıştır. Eklem mutlak konumlarını belirlemek için 10K analog potansiyometre kullanılmıştır. Kullanılan motorlar Digital Servo marka DS3218G model metal redüktörlü servo motorlardır. Atalet ölçüm sensörü olarak MPU-6050 kullanılmıştır. Yük hücresi olarak 10kg kapasiteli gerinim ölçerler kullanılmıştır. Bunlardan alınan değer HX-711 24bit analog-dijital dönüştürücüyle I2C protokolü kullanılarak işlenmiştir. Bilgisayarda oluşturulan robot yörüngesi sd karta kaydedilerek mikrodnetleyiciye aktarılmıştır. Yük hücreleri Arduino Uno model mikrodnetleyiciye bağlanmıştır. Motorlar, potansiyometreler, kart okuyucu ve atalet ölçüm ünitesi STM32F407VGT6 model mikrodnetleyiciye bağlanmıştır.

STM32F407VGT6 ana mikrodnetleyici, Arduino Uno yk mikrodnetleyicisi olarak kullanılmıřtır.

ZMP'ler ikinci mikrodnetleyicide hesaplanmıř ve ana mikrodnetleyiciye aktarılmıřtır. Ana denetleyicide yalnızca robot yrnge ve dengeleme iřlevlerinin kalması amalanmıřtır. řekil 5.10'da kırmızı renk g hattı, mavi sinyal hattı, ok ynleri de sinyal doėrultusunu gstermektedir.



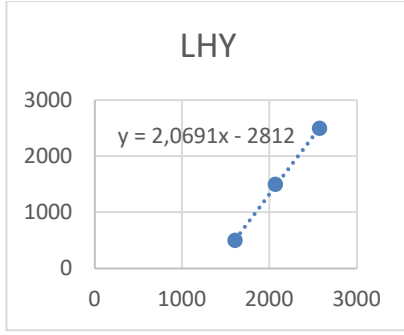
řekil 5.10: Elektronik akıř řeması.

5.3 Yazılım

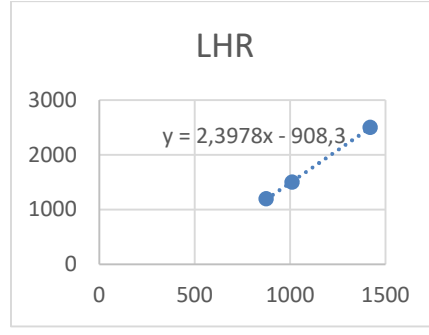
İvme lerden, potansiyometrelerden ve kuvvet sensrlerinden deėerlerin okunup motorlara yeni konum gnderilmesi 25ms srmektedir.

Robotun kinematik modelinin ve jakobiyeninin mikrodnetleyiciye gmldė yeni bir yazılım hazırlanmıřtır.

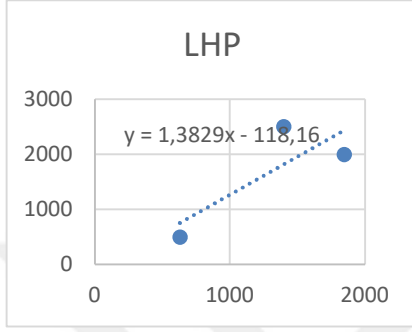
Gerek zamanlı denetim iin potansiyometre ve motorların aynı karakterde deėerler retmesi iin haritalama yapılması gerekmektedir. Bu haritalamayla ilgili grafikler řekil 5.11 ve řekil 5.12'de verilmiřtir.



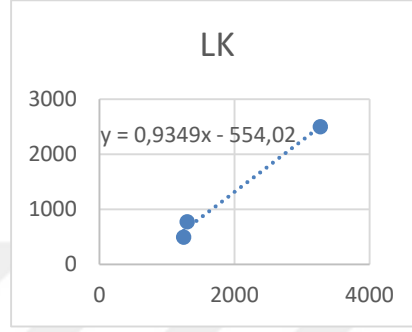
(a) Sol kalça dönme



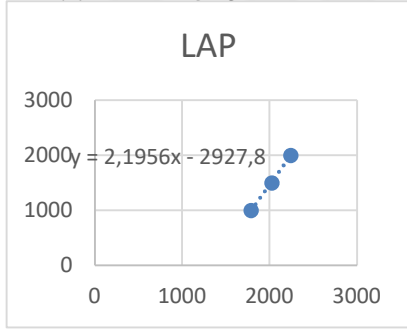
(b) Sol kalça yalpalama



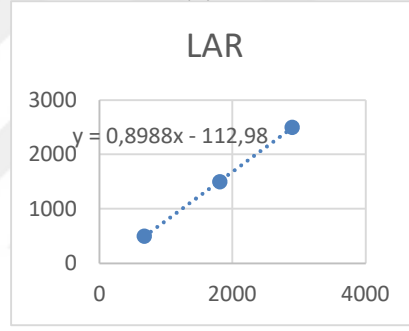
(c) Sol kalça yunuslama



(d) Sol diz

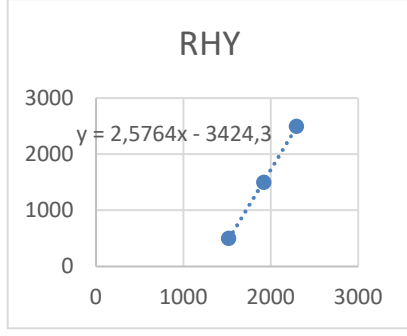


(e) Sol ayak bileği yunuslama

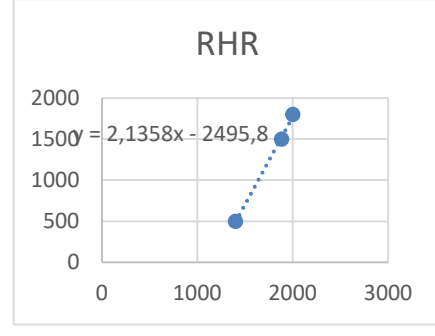


(f) Sol ayak bileği yalpalama

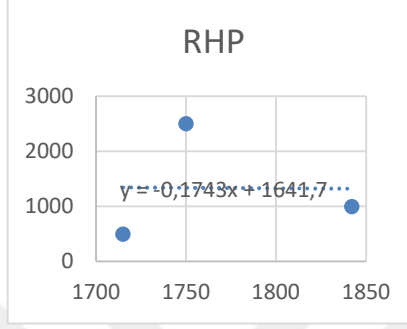
Şekil 5.11: Sol ayak için yatay eksen potansiyometre, dikey eksen motor konumu grafikleri.



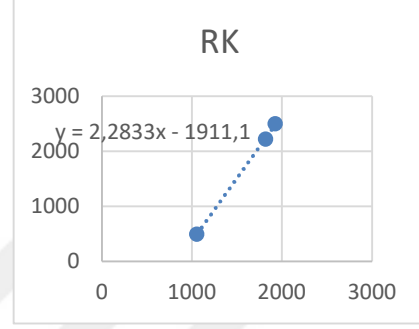
(a) Sağ kalça dönme



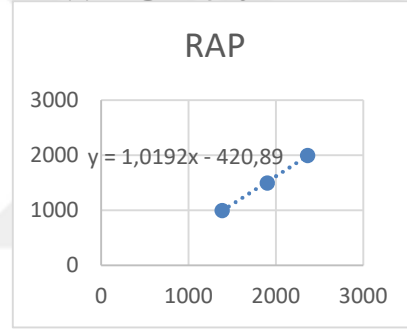
(b) Sağ kalça yalpalama



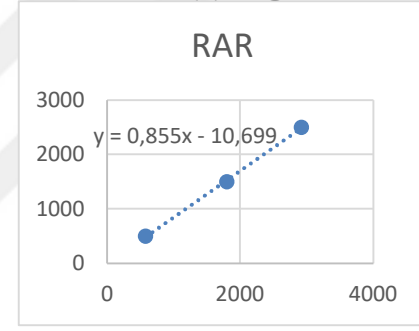
(c) Sağ kalça yunuslama



(d) Sağ diz



(e) Sağ ayak bileği yunuslama



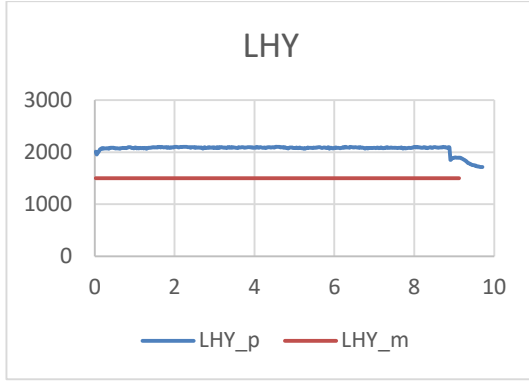
(f) Sağ ayak bileği yalpalama

Şekil 5.12: Sağ ayak için potansiyometre-motor konum haritalama grafikleri.

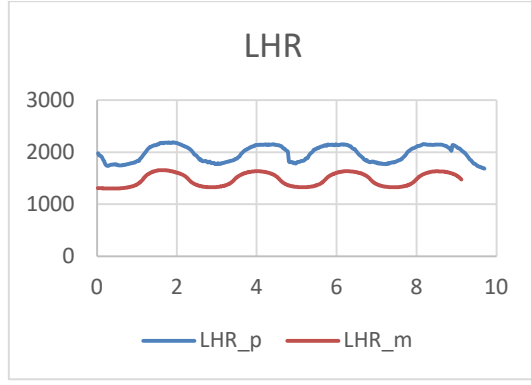
Grafikler yorumlanırken uzuvları veya ayakları birbirine yakınlaştıran hareketler “iç”, uzaklaştıran hareketler “dış” yönelim olarak tanımlanmıştır.

Kalça yalpalama bağlantılarının içe doğru hareketleri küçük, dışa doğru hareketleri büyüktür. Bu nedenle sol kalça yalpalama bağlantısının ilk 2 noktası, sağ kalçanın ise dış iki noktası birbirine yakındır. Kalça yunuslama ve diz eklemleri de benzer karakter göstermektedir. İçeri ve dışarı eşit miktarda hareket eden eklemlerin grafikleri orta noktadan geçen dikey eksene göre simetrik olmuştur.

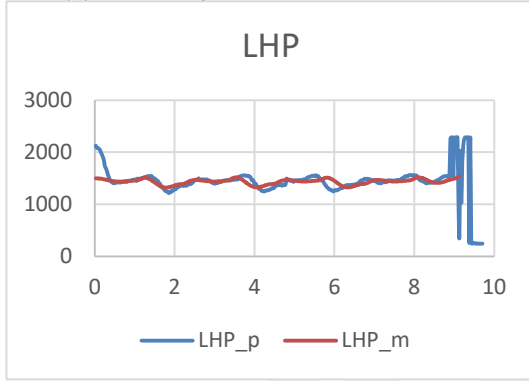
Kalibrasyon yapıldıktan sonra motorlara gönderilen yörünge ve potansiyometrelerde ölçülen değerler karşılaştırılmış, grafik olarak elde edilmiştir.



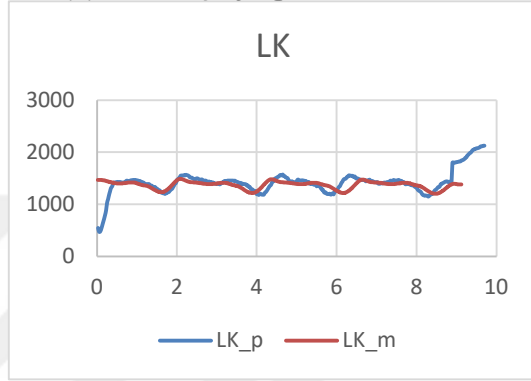
(a) Sol kalça dönme



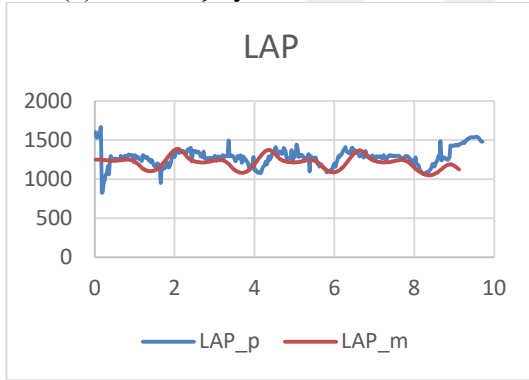
(b) Sol kalça yalpalama



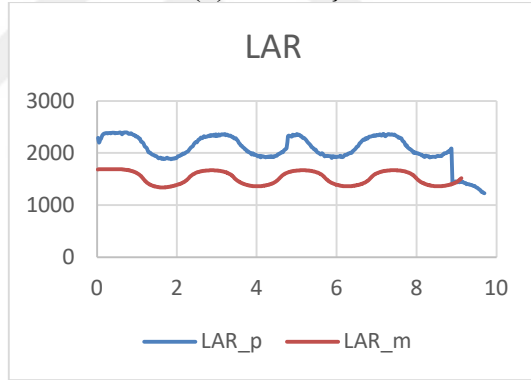
(c) Sol kalça yunuslama



(d) Sol kalça diz



(e) Sol ayak bileği yunuslama

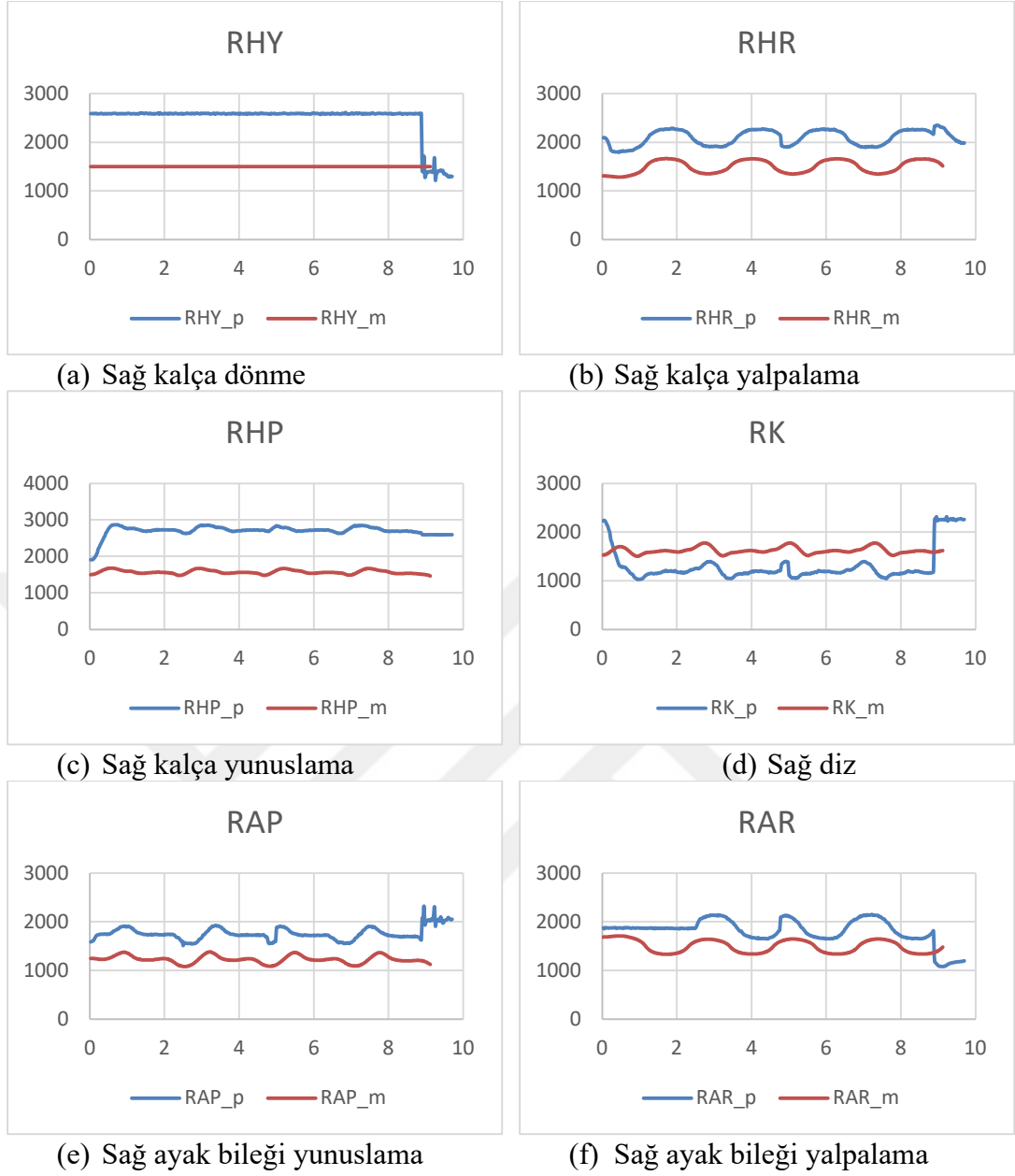


(f) Sol ayak bileği yalpalama

Şekil 5.13: Sol ayak motor yörüngeleri _m ekli ve potansiyometre değerleri _p ekli.

Yatay eksen zaman (s).

Yürüyüşün tamamlanması için motorların kumanda edildiği süre 9.12s'dir. Sonrasında motorlardaki güç kesilmektedir. Bu nedenle potansiyometre değerlerinde sıçramalar olmuştur.



Şekil 5.14: Sağ ayak motor yörüngeleri _m ekli ve potansiyometre değerleri _p ekli.

Yatay eksen zaman (s).

Şekil 5.13 ve Şekil 5.14'te verilen grafiklerde dikey eksen etiketlenmemiştir. Potansiyometrelerin verilen motor konumlarıyla genellikle uyum içinde olduğu görülmektedir. Değerler birbiriyle örtüşmemektedir. Kalibrasyon yapılırken geliştirme ortamından değişken değerlerinin izlenebilmesi için devrenin gücü USB portu üzerinden beslenmiştir. Değerler kaydedilirken ise bütün sistem güç kaynağından beslenmiştir. USB portu ile güç kaynağının gerilim seviyesi arasındaki farklar, değerlerde farklılığa sebep olmuştur.



6. SONUÇLAR

Kütle merkezi konumunun hassas olarak takip edilebilmesi için eklem açılarını hesaplayan YKM adında iteratif bir yöntem geliştirilmiştir. Yöntem kütle merkezini 0.001m veya daha yüksek hassasiyetle kesin olarak hesaplayabilmektedir. Hesaplama sonucunda tek bir çözüm sunulmaktadır. Çözümler içinden optimum çözümün seçilmesi gerekmemektedir. Yalnızca kütle merkezi ve ayak yörüngesinin bilinmesi yeterli olmaktadır.

Verilen kütle merkezi ve ayak yörüngelerine göre eklem açılarının YKM algoritmasıyla istenen hassasiyette yapılması iki ayaklı yürüyebilen bir robotun gerçek zamanlı denetimi için uygundur.

Eklem açıları yerine kalça konumlarının, hedef ve referans ağırlık merkezleri arasındaki vektörün birim vektörü kullanılarak değiştirilmesi, kütle merkezinin hassas olarak elde edilmesini sağlamıştır. Bu yöntem kullanılarak herhangi bir geri besleme alınmadan ve kontrol uygulanmadan, robotun merdiven çıkıp inebilmesi ve düz bir yüzeyde yürüyebilmesi sağlanmıştır.

Ulaşılabilen en küçük hata, Çizelge 6.1’de listelenen literatürden daha küçük bir değer olan 0.0238m olmuştur.

Çizelge 6.1 Literatürde verilen kütle merkezi takip hataları

Başlık	En küçük takip hatası	Referans
Inverse dynamics control with floating base and constraints	0.18m	(Nakanishi et al., 2007)
Humanoid gait generation in complex environments based on template models and optimality principles learned from human beings	0.03-0.4m	(Clever et al., 2018)
A kinematics-dynamics based estimator of the center of mass position for anthropomorphic system — A complementary filtering approach	0.115m	(Carpentier et al., 2015)
Center of mass estimator for humanoids and its application in modelling error compensation, fall detection and prevention	0.1m	(Xinjilefu et al., 2015)
Controlling Center of Mass in Humanoid Robot using Sliding Mode Control	0.394m	(Nguyen et al., 2020)

YKM algoritması konum kontrollü iki ayaklı robotların yürüme işlemlerinde kullanılabilir.

Ayrıca kütle merkezi ve ayak yörüngelerinin robot harekete başlamadan bilinmesi veya yeniden oluşturulması gereken durumlarda da YKM algoritması kullanılabilir.

YKM algoritması robot harekete başlamadan önce yörünge oluşturulması gereken durumlarında kullanılmalıdır. Denetleyici olarak kullanılamamıştır.

İTÜ Biped üzerinde yapılan çalışmalardan mevcut robotun yazılımı güncellenerek belirli bir seviyede durumunun güncellenebileceği ve yeni yetenekler kazandırılabilceği, ancak kapsamlı bir donanım güncellemesi ve değişikliği olmadan uygulamanın sınırlı kalacağı görülmüştür.

İTÜ Biped'in üretildiği dönemde uygun maliyetli bir robot imal etmek mümkün olmamıştır. Mikrodenetleyicilerdeki gelişmeler, eklemeli üretimin uygun maliyetle yapılabilmesi endüstriyel ortamda olmasa bile eğitim amaçlı kullanılabilecek robotların üretilmesine imkan vermiştir. Açık kaynak yazılım felsefesinin yaygınlaşması elektronik bileşenler için gerekli kütüphanelerin daha kolay elde edilmesini, daha hızlı hazırlanabilmesini sağlamıştır. Bu nedenle bu tezde geliştirilen ve üretilen iki ayaklı robot eğitim amaçlı kullanılabilir.

KAYNAKLAR

- Acar, M.** (2007). İki Ayaklı Yürüme Hareketinin Modellenmesi ve Kontrolü.[Yüksek Lisans Tezi] İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- Alba, M., Prada, J. C. G., Meneses, J., & Rubio, H.** (2010). Center of percussion and gait design of biped robots. *Mechanism and Machine Theory*, 45(11), 1681–1693. <https://doi.org/10.1016/j.mechmachtheory.2010.06.008>
- Ali, M. A., Park, H. A., & Lee, C. S. G.** (2010). Closed-form inverse kinematic joint solution for humanoid robots. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 704–709. <https://doi.org/10.1109/IROS.2010.5649842>
- Arcos-Legarda, J., Cortes-Romero, J., & Tovar, A.** (2019). Robust compound control of dynamic bipedal robots. *Mechatronics*, 59, 154–167. <https://doi.org/10.1016/j.mechatronics.2019.04.002>
- Atmeh, G., & Subbarao, K.** (2018). A neuro-dynamic walking engine for humanoid robots. *Robotics and Autonomous Systems*, 110, 124–138. <https://doi.org/10.1016/J.ROBOT.2018.09.003>
- Bajrami, X., Dermaku, A., Likaj, R., Demaku, N., Kikaj, A., Maloku, S., & Kikaj, D.** (2016). Trajectory Planning and Inverse Kinematics Solver for Real Biped Robot with 10 DOF-s. *IFAC-PapersOnLine*, 49(29), 88–93. <https://doi.org/10.1016/J.IFACOL.2016.11.108>
- Bayraktaroğlu, Z. Y., Acar, M., Gerçek, A., & Tan, N. M.** (2018). Design and Development of the I.T.U. Biped Robot. *GAZI UNIVERSITY JOURNAL OF SCIENCE*, 31(1), 251–271. <http://dergipark.gov.tr/gujs>
- Buschmann, T., Lohmeier, S., Bachmayer, M., Ulbrich, H., & Pfeiffer, F.** (2007). A collocation method for real-time walking pattern generation. *IEEE-RAS International Conference on Humanoid Robots*, 1–6. <https://doi.org/10.1109/ICHR.2007.4813841>
- Carpentier, J., Benallegue, M., Mansard, N., & Laumond, J.-P.** (2015). A kinematics-dynamics based estimator of the center of mass position for anthropomorphic system — A complementary filtering approach. *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 1121–1126. <https://doi.org/10.1109/HUMANOIDS.2015.7363493>
- Carpentier, J., Tonneau, S., Naveau, M., Stasse, O., & Mansard, N.** (2016). A versatile and efficient pattern generator for generalized legged locomotion. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June, 3555–3561. <https://doi.org/10.1109/ICRA.2016.7487538>

- Chae, H.-S.** (2004). Development of the Modeling for Biped Robot using Inverse Kinematics. <https://api.semanticscholar.org/CorpusID:16048560>
- Chen, Z., Lakbakbi Elyaaqoubi, N., & Abba, G.** (2016). Optimized 3D stable walking of a bipedal robot with line-shaped massless feet and sagittal underactuation. *Robotics and Autonomous Systems*, 83, 203–213. <https://doi.org/10.1016/j.robot.2016.05.003>
- Chevallereau, C., Razavi, H., Six, D., Aoustin, Y., & Grizzle, J.** (2018). Self-synchronization and self-stabilization of 3D bipedal walking gaits. *Robotics and Autonomous Systems*, 100, 43–60. <https://doi.org/10.1016/j.robot.2017.10.018>
- Clever, D., Hu, Y., & Mombaur, K.** (2018). Humanoid gait generation in complex environments based on template models and optimality principles learned from human beings. *The International Journal of Robotics Research*. 37(10):1184-1204. doi:10.1177/0278364918765620
- Denavit, J., & Hartenberg, R. S.** (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Journal of Applied Mechanics*, 22(2), 215–221. <https://doi.org/10.1115/1.4011045>
- Efrain, O., Ponce, R., Mansard, N., Souères, P., Ramos, O. E., & Soû, P.** (2014). Whole-body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control. *IEEE-RAS International Conference on Humanoid Robots*, 2015-February, 707–712. doi:10.1109/HUMANOIDS.2014.7041440
- Gerçek, A.** (2011). İki Ayaklı Yürüyen Robot Tasarımı Ve Prototip İmalatı.[Yüksek Lisans Tezi] İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- Hildebrandt, A.-C., Ritt, K., Wahrmann, D., Wittmann, R., Sygulla, F., Seiwald, P., Rixen, D., & Buschmann, T.** (2018). Torso height optimization for bipedal locomotion. *International Journal of Advanced Robotic Systems*, 15(5). doi:10.1177/1729881418804442
- Hong, Y. D., & Lee, B.** (2020). Real-Time Feasible Footstep Planning for Bipedal Robots in Three-Dimensional Environments Using Particle Swarm Optimization. *IEEE/ASME Transactions on Mechatronics*, 25(1), 429–437. <https://doi.org/10.1109/TMECH.2019.2955701>
- Jatsun, S., Savin, S., & Yatsun, A.** (2018). Harmonic function-based ZMP trajectory generation for nonlinear motion of walking robots. *2018 International Russian Automation Conference, RusAutoCon 2018*, 1–6. <https://doi.org/10.1109/RUSAUTOCON.2018.8501657>
- Jeong, H., Lee, K. K., Kim, W., Lee, I., & Oh, J. H.** (2020). Design and control of the rapid legged platform GAZELLE. *Mechatronics*, 66. <https://doi.org/10.1016/j.mechatronics.2019.102319>
- Joe, H. M., & Oh, J. H.** (2018). Balance recovery through model predictive control based on capture point dynamics for biped walking robot. *Robotics and Autonomous Systems*, 105, 1–10. <https://doi.org/10.1016/j.robot.2018.03.004>
- Kajita, S., Hirukawa, H., Harada, K., & Yokoi, K.** (2014). Introduction to Humanoid Robotics. *Springer Tracts in Advanced Robotics*, 101. <https://doi.org/10.1007/978-3-642-54536-8/COVER>

- Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H.** (2003). Biped walking pattern generation by using preview control of zero-moment point. *Proceedings - IEEE International Conference on Robotics and Automation*, 2, 1620–1626. <https://doi.org/10.1109/ROBOT.2003.1241826>
- Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., & Hirukawa, H.** (2001). The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. *IEEE International Conference on Intelligent Robots and Systems*, 1, 239–240. <https://doi.org/10.1109/iros.2001.973365>
- Khomariah, N. E., Pramadihanto, D., & Dewanto, R. S.** (2016). FLoW bipedal robot: Walking pattern generation. *Proceedings - 2015 International Electronics Symposium: Emerging Technology in Electronic and Information*, IES 2015, 73–78. <https://doi.org/10.1109/ELECSYM.2015.7380817>
- Kien, E. K. T., Shanmugavel, M., & Veera Ragavan, S.** (2017). Motion planning of a bipedal walking robot with leg-mounted ultrasonic sensors-An experimental study. *International Conference on Robotics and Automation for Humanitarian Applications*, RAHA 2016 - Conference Proceedings. <https://doi.org/10.1109/RAHA.2016.7931874>
- Kobayashi, T., Sekiyama, K., Hasegawa, Y., Aoyama, T., & Fukuda, T.** (2018). Virtual-dynamics-based reference gait speed generator for limit-cycle-based bipedal gait. *ROBOMECH Journal*, 5(1), 18. <https://doi.org/10.1186/s40648-018-0115-9>
- Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., & Tedrake, R.** (2016). Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3), 429–455. Doi:10.1007/s10514-015-9479-3
- Lavor, C., Xambó-Descamps, S., & Zaplana, I.** (2018). Robot kinematics. *SpringerBriefs in Mathematics*, 75–100. https://doi.org/10.1007/978-3-319-90665-2_4
- Lim, I. S., Kwon, O., & Park, J. H.** (2014). Gait optimization of biped robots based on human motion analysis. *Robotics and Autonomous Systems*, 62(2), 229–240. <https://doi.org/10.1016/j.robot.2013.08.014>
- Maalouf, N., Elhajj, I.H., Shammass, E..** (2020). Biomimetic Energy-Based Humanoid Gait Design. *Journal of Intelligent Robot Systems*, 100, 203–221, <https://doi.org/10.1007/s10846-020-01179-z>
- Mandava, R. K., & Vundavilli, P. R.** (2016). Forward and inverse kinematic based full body gait generation of biped robot. *International Conference on Electrical, Electronics, and Optimization Techniques*, ICEEOT 2016, 3301–3305. <https://doi.org/10.1109/ICEEOT.2016.7755317>
- Nakanishi, J., Mistry, M., & Schaal, S.** (2007). Inverse dynamics control with floating base and constraints. *Proceedings - IEEE International Conference on Robotics and Automation*, 1942–1947. Doi: 10.1109/ROBOT.2007.363606
- Nguyen, X. T., Tran, T. D., Nguyen, H. H., Tong, N. P., Nguyen, T. P., & Nguyen, T. T.** (2020). Controlling Center of Mass in Humanoid Robot using Sliding Mode Control. *2020 International Conference on Advanced Mechatronic Systems (ICAMechS)*, 17–22. <https://doi.org/10.1109/ICAMechS49982.2020.9310088>

- Olçay, T., & Özkurt, A.** (2017). Design and walking pattern generation of a biped robot. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(2), 761–769. <https://doi.org/10.3906/elk-1409-19>
- Park, I. W., Kim, J. Y., Lee, J., & Oh, J. H.** (2006). Online free walking trajectory generation for biped humanoid robot KHR-3(HUBO). *Proceedings - IEEE International Conference on Robotics and Automation*, 1231–1236. <https://doi.org/10.1109/ROBOT.2006.1641877>
- Rajendra, R., & Kumar Pratihari, D.** (2015). Analysis of double support phase of biped robot and multi-objective optimization using genetic algorithm and particle swarm optimization algorithm. *Sadhana*, 40, 549-575. <https://www.ias.ac.in/article/fulltext/sadh/040/02/0549-0575>
- Rokbani, N., Ammar, B., & M., A.** (2009). Toward Intelligent Biped-Humanoids Gaits Generation. InTech. doi: 10.5772/6732
- Shibata, M., Natori, T.** (2000) Impact force reduction for biped robot based on decoupling COG control scheme, *6th International Workshop on Advanced Motion Control. Proceedings (Cat. No.00TH8494)*, 612-617, doi: 10.1109/AMC.2000.862951.
- Subburaman, R., Kanoulas, D., Muratore, L., Tsagarakis, N. G., & Lee, J.** (2019). Human inspired fall prediction method for humanoid robots. *Robotics and Autonomous Systems*, 121, 103257. <https://doi.org/10.1016/j.robot.2019.103257>
- Sugihara, T.** (2011). Solvability-Unconcerned Inverse Kinematics by the Levenberg–Marquardt Method. *IEEE Transactions on Robotics*, 27(5), 984–991. <https://doi.org/10.1109/TRO.2011.2148230>
- Sugihara, T., & Morisawa, M.** (2020a). A survey: dynamics of humanoid robots. *Advanced Robotics*, 34(21–22), 1338–1352. <https://doi.org/10.1080/01691864.2020.1778524>
- Tan, N. M.** (2012). İki Ayaklı Yürüyen Robot İçin Kontrol Sistemi Geliştirilmesi.[Yüksek Lisans Tezi] İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- Tang, Z., & Er, M. J.** (2007). Humanoid 3D gait generation based on inverted pendulum model. *22nd IEEE International Symposium on Intelligent Control, ISIC 2007*, 339–344. <https://doi.org/10.1109/ISIC.2007.4450908>
- Tasasız, B.** (2011). İki Ayaklı Yürüyen Prototip Robotun Gerçek Zamanlı Denetimi, [Yüksek Lisans Tezi], Fen Bilimleri Enstitüsü, İTÜ, İstanbul.
- Tevatia, G., & Schaal, S.** (2000). Inverse kinematics for humanoid robots. *Proceedings-IEEE International Conference on Robotics and Automation*, 1, 294–299. <https://doi.org/10.1109/ROBOT.2000.844073>
- Tiseo, C., Veluvolu, K. C., & Ang, W. T.** (2019). The bipedal saddle space: Modelling and validation. *Bioinspiration and Biomimetics*, 14(1), 015001. <https://doi.org/10.1088/1748-3190/aae7b0>
- Vukobratovic, M., & Borovac, B.** (2004). Zero-Moment Point - Thirty Five Years of its Life. *Int. J. Humanoid Robotics*, 1, 157-173.

- Xinjilefu, X., Feng, S., & Atkeson, C. G.** (2015). Center of mass estimator for humanoids and its application in modelling error compensation, fall detection and prevention. *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 67–73. <https://doi.org/10.1109/HUMANOIDS.2015.7363533>
- Yang, L., Liu, Z., & Chen, Y.** (2018). Bipedal walking pattern generation and control for humanoid robot with bivariate stability margin optimization. *Advances in Mechanical Engineering*, 10(9), <https://doi.org/10.1177/1687814018800883>
- Yang, L., Liu, Z., & Chen, Y.** (2019). Energy efficient walking control for biped robots using interval type-2 fuzzy logic systems and optimized iteration algorithm. *ISA Transactions*, 87, 143–153. <https://doi.org/10.1016/j.isatra.2018.11.018>
- Yapıcı, K. O.** (2008). 14 Serbestlik Dereceli İki Ayaklı Bir Robotun Dinamik Yürüme Hareketinin Kontrolü [Yüksek Lisans Tezi]. ITUFen Bilimleri Enstitüsü, İstanbul.
- Yin, C., Zhu, J., & Xu, H.** (2009). Walking Gait Planning And Stability Control. InTech. doi: 10.5772/6735
- Zachariah, S. K., & Kurian, T.** (2017). Hybrid-state driven autonomous control for planar bipedal locomotion over randomly sloped non-uniform stairs. *Robotics and Autonomous Systems*, 97, 18–39. <https://doi.org/10.1016/j.robot.2017.08.003>
- Zhou, C., Wang, X., Li, Z., & Tsagarakis, N.** (2017). Overview of Gait Synthesis for the Humanoid COMAN. *Journal of Bionic Engineering*, 14(1), 15–25. [https://doi.org/10.1016/S1672-6529\(16\)60373-6](https://doi.org/10.1016/S1672-6529(16)60373-6)



EKLER

EK A: Robot Yörüngesi Dönüştürme Programı





EK A: Robot Yörüngesi Dönüştürme Programı

```
solAcilar(:,1) = solAcilar(:,1)+pi;
solAcilar(:,2) = solAcilar(:,2)+pi/2;

sagAcilar(:,1) = sagAcilar(:,1)+pi;
sagAcilar(:,2) = sagAcilar(:,2)+pi/2;

acilar(1,:) =[solAcilar(1,:) sagAcilar(1,:)];
dt=0.04
j=2;
ilk=0;
son=0;
d2r = pi/180;

for i=2:length(sagAcilar)
    if (son-ilk)==(dt*1000-1)
        acilar(j,:) =[solAcilar(i,:) sagAcilar(i,:)];
        j=j+1;
        son=0;
    end
    son=son+1;
end

%J2M arada%

for i=1:length(acilar)-1
    hizlar(i,:) =[(acilar(i+1,:)-acilar(i,:))./(dt)]; %puls/s
end
hizlar(i+1,:) = zeros(1,12);

%güncellenmiş hızlarla ilk hızları karşılaştırmak için
eskiHiz = hizlar;
cikar = acilar(1,:);
acilar = (acilar-ones(size(acilar)).*cikar);

t0=0;
tf=dt;
i=1;
%% ilk halini çizdirdik
for i=1:size(acilar,1)-1
    [kciz, kdotciz]=derece3(acilar(i,3),
    acilar(i+1,3),hizlar(i,3),hizlar(i+1,3),t0, tf,0.001);
```

```

        t0=t0+dt;
        tf = tf+ dt;
end
%figure,

% 3 açı değeri arasındaki hız ve ivme sürekli olacak şekilde hız
yörüngeleri
% 1den başlayarak hesaplanır
for motor=1:12
say=1;
i=i+1;
t0=0;
tf=dt;
while i<length(acilar)
    teta0 = acilar(i-1,motor);
    tetav = acilar(i,motor);
    tetag = acilar(i+1,motor);
    v0 = hizlar(i-1,motor);
    vg = hizlar(i+1,motor);
    a10 = teta0;
    a11 = v0;
    a20 = tetav;
    a12 = -(9*teta0 + 3*tetag - 12*tetav + 7*tf*v0 -
tf*vg)/(4*tf^2);
    a13 = (5*teta0 + 3*tetag - 8*tetav + 3*tf*v0 - tf*vg)/(4*tf^3);
    a21 = -(3*teta0 - 3*tetag + tf*v0 + tf*vg)/(4*tf);
    a22 = (3*teta0 + 3*tetag - 6*tetav + tf*v0 - tf*vg)/(2*tf^2);
    a23 = -(3*teta0 + 5*tetag - 8*tetav + tf*v0 - 3*tf*vg)/(4*tf^3);

    for t=0:0.001:dt
        k(say)=a10+a11*t+a12*t^2+a13*t^3;
        kdot(say) = a11+2*a12*t+3*a13*t^2;
        say=say+1;
    end
    hizlar(i,motor)=a21;
    for t=0+0.001:0.001:0.1
        k(say)=a20+a21*t+a22*t^2+a23*t^3;
        kdot(say) = a21+2*a22*t+3*a23*t^2;
        say=say+1;
    end
    i=i+1;

end
%ikinci turu cizdik
t0=0;
tf=dt;
clear kciz
clear kdotciz
for i=1:size(acilar,1)-1
    [kciz, kdotciz]=derece3(acilar(i,3),
acilar(i+1,3),hizlar(i,3),hizlar(i+1,3),t0, tf,0.001);

```

```

end

%figure,
%aynı işlem 3üncü değerden başlanarak yapılır
%ilk seferde 1den başlayarak tek sayılı indislerin hızları yumuşatıldı
%ikinci seferde 2 den başlayarak çift sayılı indisler düzeltildi
i=3;
t0=0;
tf=dt;
while i<length(acilar)
    teta0 = acilar(i-1,motor);
    tetav = acilar(i,motor);
    tetag = acilar(i+1,motor);
    v0 = hizlar(i-1,motor);
    vg = hizlar(i+1,motor);
    a10 = teta0;
    a11 = v0;
    a20 = tetav;
    a12 = -(9*teta0 + 3*tetag - 12*tetav + 7*tf*v0 -
tf*vg)/(4*tf^2);
    a13 = (5*teta0 + 3*tetag - 8*tetav + 3*tf*v0 - tf*vg)/(4*tf^3);
    a21 = -(3*teta0 - 3*tetag + tf*v0 + tf*vg)/(4*tf);
    a22 = (3*teta0 + 3*tetag - 6*tetav + tf*v0 - tf*vg)/(2*tf^2);
    a23 = -(3*teta0 + 5*tetag - 8*tetav + tf*v0 - 3*tf*vg)/(4*tf^3);

    for t=0:0.001:dt
        k(say)=a10+a11*t+a12*t^2+a13*t^3;
        kdot(say) = a11+2*a12*t+3*a13*t^2;
        say=say+1;
    end
    hizlar(i,motor)=a21;
    % t0=t0+dt;
    % tf = tf+ dt;
    for t=0+0.001:0.001:0.1
        k(say)=a20+a21*t+a22*t^2+a23*t^3;
        kdot(say) = a21+2*a22*t+3*a23*t^2;
        say=say+1;
    end
    % t0=t0+dt;
    % tf = tf+ dt;
    i=i+1;
end

%üçüncü geçisi cizdik
t0=0;
tf=dt;
clear kciz
clear kdotciz
for i=1:length(acilar)-1
    [kciz, kdotciz]=derece3(acilar(i,3),
acilar(i+1,3),hizlar(i,3),hizlar(i+1,3),t0, tf,0.001);
end

```

figure,

```
i=4
t0=0;
tf=dt;
while i<length(acilar)
    teta0 = acilar(i-1,motor);
    tetav = acilar(i,motor);
    tetag = acilar(i+1,motor);
    v0 = hizlar(i-1,motor);
    vg = hizlar(i+1,motor);
    a10 = teta0;
    a11 = v0;
    a20 = tetav;
    a12 = -(9*teta0 + 3*tetag - 12*tetav + 7*tf*v0 -
tf*vg)/(4*tf^2);
    a13 = (5*teta0 + 3*tetag - 8*tetav + 3*tf*v0 - tf*vg)/(4*tf^3);
    a21 = -(3*teta0 - 3*tetag + tf*v0 + tf*vg)/(4*tf);
    a22 = (3*teta0 + 3*tetag - 6*tetav + tf*v0 - tf*vg)/(2*tf^2);
    a23 = -(3*teta0 + 5*tetag - 8*tetav + tf*v0 - 3*tf*vg)/(4*tf^3);

    for t=0:0.001:dt
        k(say)=a10+a11*t+a12*t^2+a13*t^3;
        kdot(say) = a11+2*a12*t+3*a13*t^2;
        say=say+1;
    end
    hizlar(i,motor)=a21;
%     t0=t0+dt;
%     tf = tf+ dt;
    for t=0+0.001:0.001:0.1
        k(say)=a20+a21*t+a22*t^2+a23*t^3;
        kdot(say) = a21+2*a22*t+3*a23*t^2;
        say=say+1;
    end
%     t0=t0+dt;
%     tf = tf+ dt;
    i=i+1;
end

%epos2'nin sistemine uygun olarak gönderilecek hız ve konum değerleri
%sonra izlenecek yörünge
t0=0
tf=dt
clear kciz
clear kdotciz
for i=1:length(acilar)-1
    [kciz, kdotciz]=derece3(acilar(i,motor),
acilar(i+1,motor),hizlar(i,motor),hizlar(i+1,motor),t0, tf,0.001);
    if i==1
        cikar = kciz(1);
    end
    % kciz = kciz.*(1000/pi)
```

```

    kciz = kciz-ones(size(kciz)).*cikar;
    kdotciz = kdotciz.*(60/2000); %puls/sn*60sn/1dk*1dev/2000puls
    subplot(2,1,1)
    plot(linspace(t0, tf,length(kciz)),kciz)
    hold on
    subplot(2,1,2)
    hold on
    plot(linspace(t0, tf,length(kdotciz)),kdotciz)
    t0=t0+dt;
    tf = tf+ dt;
end
drawnow
end

fileID = fopen('puls1ar.txt','wt+');

for ii = 1:size(acilar,1)
    fprintf(fileID,'%d\t',acilar(ii,:));
    fprintf(fileID,'\n');
end
fclose(fileID);

hizlar2 = round(hizlar.*60/2000); %puls/sn*60sn/1dk*1dev/2000puls
fileID = fopen('hizlar.txt','wt+');

for ii = 1:size(hizlar2,1)
    fprintf(fileID,'%d\t',hizlar2(ii,:));
    fprintf(fileID,'\n');
end
fclose(fileID);

```



ÖZGEÇMİŞ

Ad-Soyad : İsmail Hakkı ŞANLITÜRK

ÖĞRENİM DURUMU:

- **Lisans** : 2008, Fırat Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği
- **Yükseklisans** : 2012, Fırat Üniversitesi, Makine Mühendisliği, Konstrüksiyon ve İmalat

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2009-2018 yılları arasında Fırat Üniversitesi Mühendislik Fakültesi Makine Mühendisliği Konstrüksiyon ve İmalat Anabilim Dalı'nda araştırma görevlisi olarak çalıştı.
- 2018 yılından itibaren Tokat Gaziosmanpaşa Üniversitesinde makine mühendisi olarak çalışıyor.
- 2012 Uluslararası SpaceClaim Tasarım yarışmasında üçüncülük ödülü.

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Şanlıtürk İ.H., Kocabaş H.** 2023. Precise inverse kinematic calculation of joint angles respect to center of gravity of a bipedal robot. *International Graduate Research Symposium*, May 2-5, 2023 İstanbul, Türkiye.
- **Şanlıtürk İ.H., Kocabaş H.** 2024. Precise Calculation of Inverse Kinematics of the Center of Gravity for Bipedal Walking Robots, *Applied Sciences*, 2024; 14(9):3706. <https://doi.org/10.3390/app14093706>.

