

Akıllı Tekerlekli Sandalye İçin Yüksek Seviye Kontrol Sistemi Gerçekleşmesi

Bora Akar

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliğı Anabilim Dalı

Haziran 2014

High Level Controller Implementation For Intelligent Wheelchair

Bora Akar

MASTER OF SCIENCE THESIS

Department of Electric Electronic Engineering

June 2014

Akıllı Tekerlekli Sandalye İçin Yüksek Seviye Kontrol Sistemi Gerçekleşmesi

Bora Akar

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliğı Uyarınca
Elektrik Elektronik Mühendisliğı Anabilim Dalı
Kontrol ve Kumanda Sistemleri Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. Ahmet YAZICI

Haziran 2014

ONAY

Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Bora Akar'ın YÜKSEK LİSANS tezi olarak hazırladığı “Akıllı Tekerlekli Sandalye İçin Yüksek Seviye Kontrol Sistemi Gerçeklemesi” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman : Yrd. Doç. Dr. Ahmet YAZICI

İkinci Danışman : -

Yüksek Lisans Tez Savunma Jürisi:

Üye : Yrd. Doç. Dr. Ahmet YAZICI

Üye : Prof. Dr. Abdurrahman KARAMANCIOĞLU

Üye : Prof. Dr. Osman PARLAKTUNA

Üye : Doç. Dr. Rıfat EDİZKAN

Üye : Yrd. Doç. Dr. Metin ÖZKAN

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Günümüzde gezgin robotların, iç ve dış ortamlarda kullanımı giderek artmaktadır. Askeri projelerden eğlenceye, sağlık sektöründen endüstriyel alanlara kadar uzanan geniş yelpazede yarı – otonom veya tam otonom robotlar üzerinde çeşitli uygulamalar geliştirilmektedir. Engelli veya yaşlı insanların kullanımı için olan akıllı tekerlekli sandalyeler, temizlik robotları, futbol oynayan robotlar, insansı robotlar, uzay keşif robotları, esnek otomasyonda kullanılan gezgin robotlar bu kapsamda örnek olarak kullanımda olan robotlardan bazılarıdır.

Bu çalışmada, Akıllı TEKERlekli Sandalyenin (ATEKS) özerk kontrolü için yüksek seviye kontrol sistemi ve alt davranışlar gerçekleştirilmiştir. ATEKS, öncelikle hareket engelli insanların kullanımında özerk hareket edebilen bir platform olarak geliştirilmiştir. Özerk modun yanı sıra sürüş destek amacı ile de kullanılabilir. Ayrıca, gezgin robot eğitim ve araştırmalarında kullanılacak temel bileşenlere sahiptir. ATEKS platformu, Düşük Seviye Akıllı Kontrol Birimi (AKB – DSK), Yüksek Seviye Akıllı Kontrol Birimi (AKB – YSK), Kullanıcı Arayüzü Birimi (KAB) ve gerekli algılayıcılardan oluşmaktadır. ATEKS’in özerk kontrolü melez yüksek seviye kontrol mimarisi üzerine gerçekleştirilmiştir. Bu mimaride, ATEKS platform farklı alt birimleri arasındaki koordinasyon açık ajan tabanlı robot işletim sistemi (Robot Operating System, ROS) kullanılarak gerçekleştirilmiştir. İlgili özerk davranışların koordineli kontrolü için ROS ile bütünleşik çalışan sonlu durum makinesi geliştirilmiştir. Geliştirilen yüksek seviye kontrol mimarisi, GAZEBO benzetim ortamında ve Eskişehir Osmangazi Teknopark Binasında gerçekleştirilen uygulamalarla test edilmiştir.

Anahtar Kelimeler: Akıllı Tekerlekli Sandalye, Gezgin Robotlar, Yüksek Seviye Kontrol, Robot İşletim Sistemi, Sonlu Durum Makinesi

SUMMARY

Nowadays, autonomous mobile robots are increasingly used in both indoor and outdoor environments. There is a wide range of applications of mobile robots from military projects to entertainment, or from medical to industrial applications. Some examples include intelligent wheelchairs for the usage of elderly and disabled people, space exploration robots, floor cleaning robots, soccer playing robots, mobile robots for flexible automation.

In this study, a high level controller and robotic behaviors and related robotic behaviors have been developed for autonomous control of an intelligent wheelchair (ATEKS). Firstly, the ATEKS platform is developed as an autonomous mobile platform for the usage of disabled or elderly people. The platform can also be used for driving assistance rather than fully autonomous system. In addition, it has the basic components to use in the area of mobile robot education and researches. ATEKS has Intelligent Control Unit – High Level Control (AKB – YSK), Intelligent Control Unit – Low Level Control (AKB – DSK), User Interface Unit (KAB) and sensor units (Localization, measurement sensors etc.). Autonomous control of ATEKS platform is implemented as a Hybrid Control Architecture. In this architecture, coordination of various units of ATEKS platform is achieved using Robot Operating System (ROS). Autonomous behaviors are coordinated by a Finite State Machine (FSM) which is integrated into the ROS. The developed high level control architecture is tested on both GAZEBO simulation environment and at Osmangazi Technopark.

Keywords: Intelligent Wheelchair, Mobile Robots, High Level Control, Robot Operating System, Finite State Machine

TEŞEKKÜR

Yüksek lisans tez çalışmamda bana danışmanlık ederek yönlendiren ve her türlü desteği sağlayan, tecrübelerinden yararlanırken göstermiş olduğu hoşgörü ve sabırdan dolayı değerli danışmanım Yrd. Doç. Dr. Ahmet YAZICI'ya, yüksek lisansımın başından tezimin son aşamasına kadar yanımda olan, manevi desteklerini eksik etmeyen annem Cahide AKAR, babam Mustafa AKAR ve abim Yusuf Barış AKAR'a, teknik donanım ve alt yapı desteği veren şirketim İnovasyon Mühendisliğe ve çalışanlarına teşekkürlerimi sunarım.

Bu tez, İnovasyon Mühendislik Teknoloji Geliştirme Danışmanlık Tic. Ltd. tarafından yürütülen TÜBİTAK Teydep 7120742 nolu Gezgin Robotlar İçin Akıllı Kontrol Sistemi Kobi-Arge projesi kapsamında desteklenmiştir.

İÇİNDEKİLER

Sayfa

ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ.....	xii
SİMGELER VE KISALTMALAR DİZİNİ.....	xiii
1. GİRİŞ.....	1
2. GEZGİN ROBOT KONTROL YAKLAŞIMLARI VE ARA KATMANLAR.....	4
2.1. Sıradüzensel Kontrol.....	5
2.2 Tepkisel Kontrol	6
2.3.1 Potansiyel alanlar yaklaşımı	7
2.3.2 Kapsama yaklaşımı	10
2.4 Melez Kontrol	11
2.4.1 TCA	11
2.3.2 NASREM.....	12
2.4.3 LAAS	13
2.4.4 Ajan tabanlı yaklaşım	14
2.4 Davranış Tabanlı Kontrol	16
2.5 Robotik Ara Katmanlar	20
2.5.1 ARIA.....	21
2.5.2 PLAYER.....	23
2.5.3 OROCOS	24
2.5.4 OPENRTM-aist	24
2.5.5 CLARATY	25
2.5.6 MIRO	26
2.5.7 SKILLIGENT	27
2.5.8 OAA.....	28

2.5.9 ROS.....	29
3. AKILLI TEKERLEKLİ SANDALYE PLATFORMU	31
3.1 Tekerlekli Sandalye Standart Donanımı	32
3.1.1 Kumanda kolu.....	33
3.1.2 Motor sürücüsü	33
3.2 Akıllı Kontrol İçin Entegre Edilen Çevre Birimleri	34
3.2.1 Akıllı kontrol birimi.....	35
3.2.2 Konum hesaplama birimi.....	38
3.2.3 Mesafe algılama birimleri.....	39
3.2.4 Kullanıcı arayüzü birimi	41
4. ÖNERİLEN YÜKSEK SEVİYE KONTROL SİSTEMİ	47
4.1 ATEKS Kontrol Mimarisi	47
4.2 AKB – YSK ROS Düğümleri	48
4.2.1 AKB – DSK ROS düğümü	51
4.2.2 KAB ROS düğümü	52
4.2.3 Kinect ROS düğümü.....	52
4.2.4 GAB ROS sürücü düğümü	53
4.2.5 Kalman ROS düğümü.....	53
4.2.6 İzleme ROS düğümü.....	54
4.2.7 AKB – DSK benzetim ROS düğümü	54
4.2.8 Özerk kontrol ROS düğümü	54
4.3 Özerk Kontrol Düğümü İçin Geliştirilen Davranışlar	59
4.3.1 Hedefe gitme davranışı	66
4.3.2 Koridor takibi davranışı	67
4.3.3 Kapıdan geçme davranışı.....	67
4.3.4 Cisme yaklaşma davranışı.....	71
5. ATEKS PLATFORM TESTLERİ.....	73
5.1 Benzetim Ortamı Testleri.....	76
5.2 Gerçek Ortam Testleri	81
SONUÇLAR VE ÖNERİLER.....	87
KAYNAKLAR DİZİNİ.....	88

ŞEKİLLER DİZİNİ

Şekil	Sayfa
Şekil 2.1 Robotik kontrol yaklaşımları: a) sıradüzensel b) tepkisel c) melez.....	5
Şekil 2.2 Tipik bir STRIPS arama ağacı.....	6
Şekil 2.3 Potansiyel alan vektörleri	7
Şekil 2.4 Çekici potansiyel alan vektörleri	8
Şekil 2.5 İtici potansiyel alan vektörleri	9
Şekil 2.6 Teğetsel potansiyel alan vektörleri	9
Şekil 2.7 Subsumption yöntemi	10
Şekil 2.8 TCA yaklaşımı.....	12
Şekil 2.9 NASREM yapısı	13
Şekil 2.10 Örnek LAAS yaklaşımı	14
Şekil 2.11 OAA genel yapısı	15
Şekil 2.12 Ajan çalışma mekanizması	15
Şekil 2.13 Davranış genel yapısı.....	17
Şekil 2.14 Davranışın şema olarak gösterimi: a) şemanın yapısı, b) algısal ve motor şema olarak ayrıştırılmış davranış yapısı.....	17
Şekil 2.15 Davranışların sonlu durum otomatu ile birleştirilmesi	19
Şekil 2.16 Örnek bir SRD diyagramı.....	19
Şekil 2.17 Örnek bir davranış birleştirme	20
Şekil 2.18 Robotik ara katman.....	21
Şekil 2.19 Aria yazılım mimari yapısı	22
Şekil 2.20 Player yazılım mimari yapısı	23
Şekil 2.21 RTM port yapısı.....	25
Şekil 2.22 CLARAty mimarisi	26
Şekil 2.23 MIRO soyutlama katmanı	27
Şekil 2.24 SKILLEAGENT farklı yazılım modüllerinin eklenmesi işlemi	28
Şekil 2.25 OAA genel yapısı	28
Şekil 2.26 Örnek bir ROS çalışma yapısı	30
Şekil 3.1 ATEKS sistemi genel yapısı.....	32
Şekil 3.2 Kumanda kolu.....	33
Şekil 3.3 Motor sürücü VR2	34
Şekil 3.4 ATEKS çevre ve kontrol birimleri	35
Şekil 3.5 AKB - DSK donanım prototipi.....	36
Şekil 3.6 AKB - DSK görev programlayıcı çalışma mekanizması.....	37
Şekil 3.7 AKB - YSK donanımı	37
Şekil 3.8 Konum hesaplama birimi.....	38
Şekil 3.9 Kodlayıcı	39
Şekil 3.10 Sesüstü mesafe ölçme birimi	40
Şekil 3.11 Microsoft kinect.....	41
Şekil 3.12 KAB için kullanılan donanım.....	42

Şekil 3.13 KAB sürüş destek ekranı	43
Şekil 3.14 KAB navigasyon ekranı.....	43
Şekil 3.15 Teknopark krokisi.....	44
Şekil 3.16 KAB yönetici ekranı	46
Şekil 4.1 Yüksek seviye kontrol mimarisi	48
Şekil 4.2 AKB - YSK ROS düğümleri	49
Şekil 4.3 ROS düğümleri dizi diyagramı.....	50
Şekil 4.4 Özerk mod SDM durum geçiş çizgesi	56
Şekil 4.5 VFH algoritması akış diyagramı.....	61
Şekil 4.6 VFH ızgara ve sektör örneği.....	62
Şekil 4.7 Sektör yoğunluklarının histogram şeklinde gösterimi.....	64
Şekil 4.8 Kapıdan geçme davranışı akış diyagramı	68
Şekil 4.9 Kinect algılayıcısı verisinin iki boyuta indirilmesi.....	69
Şekil 4.10 En yakın ızgara hücreleri	70
Şekil 4.11 Filtrelenmiş büyük ızgaralar örneği.....	70
Şekil 4.12 ATEKS izlenecek cisme yanaşma rotası	71
Şekil 5.1 Teknopark test alanı.....	73
Şekil 5.2 Teknopark oda1	74
Şekil 5.3 Teknopark koridor	74
Şekil 5.4 Teknopark park alanı	75
Şekil 5.5 Teknopark çıkış	75
Şekil 5.6 GAZEBO teknopark giriş katı modeli.....	76
Şekil 5.7 GAZEBO test1 alınan konum bilgileri.....	77
Şekil 5.8 GAZEBO test1 ölçülen açısız ve doğrusal hız değerleri	77
Şekil 5.9 GAZEBO test2 alınan konum bilgileri	78
Şekil 5.10 GAZEBO test2 ölçülen açısız ve doğrusal hız bilgileri	78
Şekil 5.11 GAZEBO test3 alınan konum bilgileri	79
Şekil 5.12 GAZEBO test3 ölçülen açısız ve doğrusal hız bilgileri	79
Şekil 5.13 GAZEBO test4 alınan konum bilgileri	80
Şekil 5.14 GAZEBO test4 ölçülen açısız ve doğrusal hız bilgileri	80
Şekil 5.15 Gerçek ortam kapıdan geçme testi alınan konum bilgileri	82
Şekil 5.16 Gerçek ortam kapıdan geçme testi ölçülen açısız ve doğrusal hız değerleri	82
Şekil 5.17 Gerçek ortam koridor takibi testi alınan konum bilgileri	83
Şekil 5.18 Gerçek ortam koridor takibi testi ölçülen açısız ve doğrusal hız bilgileri.....	83
Şekil 5.19 Gerçek ortam cisme yanaşma testi alınan konum bilgileri.....	84
Şekil 5.20 Gerçek ortam cisme yanaşma testi ölçülen açısız ve doğrusal hız bilgileri ..	84
Şekil 5.21 Gerçek ortam tam ortam testi alınan konum bilgileri.....	85
Şekil 5.22 Gerçek ortam tam ortam testi ölçülen açısız ve doğrusal hız bilgileri	85

ÇİZELGELER DİZİNİ

Çizelge

Sayfa

Çizelge 2.1 Davranışların sınıflandırılması	18
Çizelge 3.1 Kodlayıcı teknik özellikleri	39
Çizelge 3.2 Düğümlerin alabileceği değerler	45
Çizelge 4.1 AKB - DSK düğümü topikleri	51
Çizelge 4.2 KAB ROS düğümü topikleri	52
Çizelge 4.3 Kinect düğümü topikleri	52
Çizelge 4.4 GAB düğümü topikleri	53
Çizelge 4.5 Kalman düğümü topikleri	53
Çizelge 4.6 AKB - YSK düğümü topikleri	55
Çizelge 4.7 Özerk kontrol ROS düğümü SDM girdileri	57
Çizelge 4.8 Tüm davranışlar için VFH parametre değerleri	66
Çizelge 4.9 Davranışlar için çarpışma engelleme parametre değerleri	72

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar	Açıklamalar
AKB	Akıllı Kontrol Birimi
AKB – DSK	Akıllı Kontrol Birimi Düşük Seviye Kontrol
AKB – YSK	Akıllı Kontrol Birimi Yüksek Seviye Kontrol
ARIA	Advanced Robot Interface for Applications – Uygulama Robot Arayüzü
ATEKS	Akıllı Tekerlekli Sandalye
CLARAty	Coupled Layer Architecture for Robotic Autonomy
FSA	Finite State Automata – Sonlu Durum Otomatı
GAB	Gezgin Algılayıcı Birim
HIL	Hardware In Loop – Döngü İçinde Donanım
İÇKON	İç Ortamlar İçin Konumlandırma
JOSM	Java Open Street Map– Java Açık Sokak Haritası
KAB	Kullanıcı Arayüz Birimi
LAAS	Laboratory for Analysis and Architecture of Systems
LISP	List Processing – Liste İşleme Programlama Dili
MIRO	Middleware for Robots – Robotlar için Ara Katman
NASREM	NASA/NBS Standart Reference Model for Telerobot Control System
Architecture	
NPC	Non Player Character – Oyuncu Olmayan Yardımcı Karakter
OAA	Open Agent Architecture – Açık Ajan Mimarisi
OPENRTM	Open Real Time Middleware – Açık Gerçek Zamanlı Ara Katman
OROCOS	Open Robot Control Software – Açık Robot Kontrol Yazılımı
POI	Point Of Interest – Yer İmi
PROLOG	Programming Logic – Programlama Mantığı
ROS	Robot Operating System – Robot İşletim Sistemi
RTOS	Real Time Operating System – Gerçek Zamanlı İşletim Sistemi
SDK	Software Development Kit – Yazılım Geliştirme Seti
SDM	Sonlu Durum Makinesi
SMO15	Sesüstü Mesafe Ölçer
SRD	Stimulus Response Diagram – Refleks Karşılık Diyagramı
STRIP	Stanford Research Institute Problem Solver
TCA	Task Controller Architecture – Görev Kontrol Mimarisi
XML	Extended Mark-up Language – Genişletilebilir Biçimsel Dil
VFH	Vector Field Histogram – Vektör Alan Histogramı

BÖLÜM 1

GİRİŞ

Günümüzde gezgin robotların, iç ve dış ortamlar için kullanımı giderek yaygınlaşmaktadır. Engelli veya yaşlı insanların kullanımı için akıllı tekerlekli sandalyeler (Levine, et. al., 1999; Matsumoto, et al., 2006; Liu, et al., 2010), temizlik robotları (Palacin, et al., 2004), futbol oynayan robotlar (Weigel, et al., 2002), uzay keşif robotları (Gao, et al., 2008) bu kapsamda geliştirilmiş gezgin robotlardan bazılarıdır. Literatürde, gezgin robotların özerk kontrolü için çeşitli kontrol mimarileri, farklı ara katmanlar ve birçok kontrol yöntemi bulunmaktadır (Siciliano, et. al, 2008).

Gezgin robotlarda kontrol mimarisi, bulunduğu ortama ait tüm kısıtları dikkate alarak robota verilen görevlerin kusursuz bir şekilde gerçekleşmesini sağlayan bir yapıdır (Mataric, et al., 1992). Bu Kontrol mimarileri; ARIA, PLAYER, OROCOS ve ROS gibi işletim sistemi ve uygulamaya yazılımı arasında kalan robotik ara katmanlar (Elkady, et al., 2012) ile gerçekleştirilmektedir. Kontrol mimarileri ve robotik ara katmanlar özerk kontrolü gerçekleştirmek üzere geliştirilen farklı kontrol yöntemlerinin koordineli bir şekilde çalışmasını sağlamaktadır. Literatürde, gezgin robotların yüksek seviye kontrolünde karar verme mekanizması olarak petri ağları (Agerwala, et al., 1979) ve sonlu durum makineleri (Phillips, et. al., 1985) gibi yapılar kullanılmaktadır.

Gezgin robotların yüksek seviye kontrolünde kullanılan karar verme yapılarından biri petri ağlarıdır (Wang, et al., 1991; Moon, et al., 2010; Lee, et al., 2006; Mahulea, et al., 2012). Çalışmada (Wang, et al., 1991) akıllı gezgin robot sisteminin koordinasyonu için bir petri net modeli sunulmaktadır. Model ile rota planlama, hareket kontrolü, görüntüleme işlemlerinin dinamik bir ortamda koordineli bir şekilde çalışabilmesi petri ağları ile sağlanmaktadır. Diğer bir çalışmada (Moon, et al., 2010) kesik zamanlı (discrete) navigasyon davranışları ve bu davranışların seçimini kontrol eden bir yapı petri ağlarıyla tasarlanmıştır. Benzer bir çalışmada ise (Lee, et al., 2006)

gezgin robotlar için iç ortamlarda konumlandırma ve robot navigasyonu problemlerine petri ağı tabanlı kesik zamanlı (discrete) bir kontrol sistemi geliştirilmiştir. Diğer bir çalışmada (Kim, et al., 2007) bir servis robotunda çoklu navigasyon davranışlarının seçiminin sağlanması için petri ağı kullanılmıştır. Geliştirilen sistem dinamik ortamda çalışma zamanında ilgili davranışların seçimini yapmaktadır. Benzer bir çalışmada (Mahulea, et al., 2012) olasılıksal ayırık zamanlı petri ağı kullanarak robot kontrol mekanizması geliştirilmiştir. Robotun bulunduğu ortama ait engellerden çevrim içi (online) tabanlı bir algoritma ile kaçınması üzerine çalışma yapılmıştır. Bir diğer yüksek seviye kontrol için kullanılan karar verme mekanizması da sonlu sayıda durum ve durumlar arası geçişlerden oluşan sonlu durum makineleridir. Çalışmada (Akar vd., 2013) bir akıllı tekerlekli sandalyenin düşük seviye kontrol işlemlerini gerçeklemek için sonlu durum makinesi kullanılmıştır. Sonlu durum makinelerinde; durumlar, geçmiş hakkında bilgi tutmakta, geçişler, durum değişimlerini göstermekte ve eylemler, belirli bir zamanda gerçekleştirilen etkinliği tanımlamaktadır. Gerek petri ağları, gerekse sonlu durum makineleri literatürde bulunan, potansiyel alanlar (Khatib, et al., 1985), bulanık mantık (Berenji, et al., 1991), vektör alan histogramı (Borenstein, et al., 1991) gibi farklı yelpazede kontrol yöntemleri kullanılarak gezgin robotların kontrolü sağlamaktadır.

Potansiyel alanlar yaklaşımı (Khatib, et al., 1985; Synodinos, et al., 2010; Huang, et al., 2007; Rezaee, et al., 2011; Liu, et al., 2008; Charifa, et al., 2009), temelde ortamdaki engeller ve hedef noktasına karşılık gelen vektörler üretip, bunları toplama işlemine tabi tutarak tepkisel çıktılar üreten bir yapıya sahiptir. Potansiyel alan yaklaşımı kısıtlarını ortadan kaldırmak amacıyla geliştirilen vektör alan histogramı (Borenstein, et al., 1991; Ulrich, et al., 1998; An, et al., 2004), ortamın ızgaralara bölünüp yoğunluk histogramları çıkartılması ve bu histogramlara bağlı olarak robotun hareket ettirilmesini amaçlayan bir yaklaşımdır. Bulanık mantık yaklaşımı (Thongchai, et al., 2000; Siddaiyan, et al., 2012; Mariappan, et al., 2011; Ling, et al., 2013; Parlaktuna, et al., 2005; Bayar vd., 2014) sırasıyla bulanıklaştırma (fuzzification); çıkarsama (inference), kesinleştirme (defuzzification) adımlarını kullanarak tepkisel çıktılar üreten bir diğer kontrol yöntemidir.

Bu tez çalışmasında, Akıllı TEKERlekli Sandalye (ATEKS) platformu için yüksek seviye kontrol sistemi geliştirilmiştir. ATEKS platformu, Düşük Seviye Akıllı Kontrol Birimi (AKB – DSK), Yüksek Seviye Akıllı Kontrol Birimi (AKB – YSK), Kullanıcı Arayüzü Birimi (KAB) ve Algılayıcı birimlerinden oluşmaktadır. Bu birimler arasında haberleşme ve koordinasyonun sağlanması için ROS ara katmanı kullanılmıştır. ATEKS platformunun iç ortamlarda özerk kontrolü için ihtiyaç duyduğu: Koridor Takibi, Kapıdan Geçme, Hedefe Gitme ve Cisme Yanaşma gibi davranışlar, vektör alan histogramı yöntemi ve özel tasarlanan algoritmalar kullanılarak gerçekleştirilmiştir. Uygulamalarda bu davranışların koordineli çalışması için karar mekanizması olarak sonlu durum makinası kullanılmıştır. ATEKS platformunun yüksek seviye kontrol sistemi, GAZEBO benzetim ortamı ve Eskişehir Osmangazi Teknopark Binası'nda gerçekleştirilen uygulamalarla test edilmiştir.

Takip eden alt bölümde öncelikle gezgin robot kontrol yaklaşımları ve robotik ara katmanlar ile ilgili bilgi verilmektedir. ATEKS platformu bölüm 3'te tanıtılmaktadır. Önerilen yüksek seviye kontrol mimarisi ve geliştirilen davranışlar da bölüm 4'te verilmektedir. Bölüm 5'te, GAZEBO benzetim ortamı ve gerçek ortamda yapılan test sonuçları sunulmaktadır. Son bölümde ise çalışma değerlendirmesi ve sonuçlar vardır.

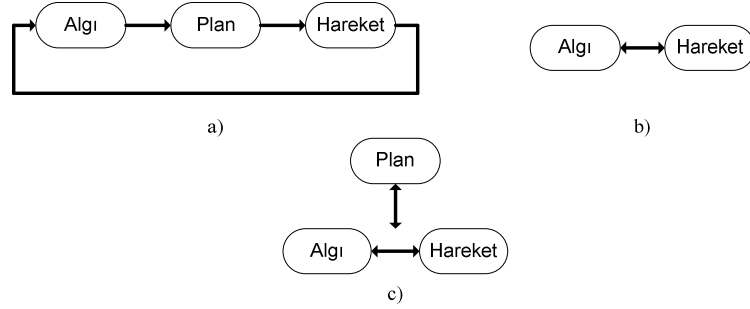
BÖLÜM 2

GEZGİN ROBOT KONTROL YAKLAŞIMLARI VE ARA KATMANLAR

Kontrol mimarisi, gezgin robotların bulunduğu ortama ait tüm kısıtları göz önünde bulundurarak robotun yerine getirmesi gereken görevlerin kusursuz bir şekilde yapılmasını sağlayan bir yapıdır (Mataric, et al., 1992). Bu yapının, algılayıcılardan elde edilen verilerin hızlı bir şekilde işlenip, verilen görevleri dikkate alarak istenen karar çıktılarını üretebilmesi için ilgili altyapıyı sağlaması gerekmektedir. Bu durumlar göz önüne alındığında, bir gezgin robot için özerk kontrol mimarisi geliştirilirken ortam değişikliklerine tepkisellik, akıllı davranış, çoklu hedefleri çözümleyebilme, güvenilirlik, gürbüzlük, esneklik, genişletilebilirlik ve uyum yeteneği özelliklerine sahip olması gerekmektedir.

Ortam değişkenlerine tepkisellik, gezgin robotun kendi görevini doğru, verimli ve güvenli bir şekilde yapabilmek için ortamda oluşan ani değişikliklere tepki verebilmesidir. Akıllı davranış, dış ortamdan gelen uyarılara karşı robot kendi ana görevinin hedefleri doğrultusunda algılayıcılarından gelen verileri işleyerek mantıklı çözümler sunabilmesidir. Çoklu hedefleri çözümleyebilme, robotun birbirinden farklı görevleri yerine getirmesidir. Gürbüzlük, gezgin robotun beklenmedik olaylar karşısında verilen görevi gerçekleştirmesidir. Güvenilirlik, robotun bir görevi gerçekleştirirken performans düşüklüğüne uğramadan davranışı tekrar edebilmesidir. Esneklik, robotlarla yapılan deneylerin gerçekleşme aşamasında tasarımları sürekli değişikliklere gereksinim duyabileceğinden, kontrol mimarisi tasarlanırken gelecekteki görevlerin kontrolünün mimari içinde kolay bir şekilde bütünleştirilebilirliğinin göz önünde bulundurulmasıdır. Uyum Yeteneği, bir robot kontrol sistemini aniden farklı kontrol stratejilerine geçmede uyumlu olmasıdır.

Literatürde bu amaçlar ile farklı robot kontrol yaklaşımları geliştirilmiştir. Şekil 2.1’de, gezgin robotlar için geliştirilmiş sıradüzensel (hierarchical), tepkisel (reactive), melez (hybrid) düşünsel/tepkisel kontrol yaklaşımları verilmektedir.



Şekil 2.1 Robotik kontrol yaklaşımları: a) sıradüzensel b) tepkisel c) melez

Takip eden alt bölümde bu yaklaşımlar detaylandırılmakta ve örnekler verilmektedir.

2.1. Sıradüzensel Kontrol

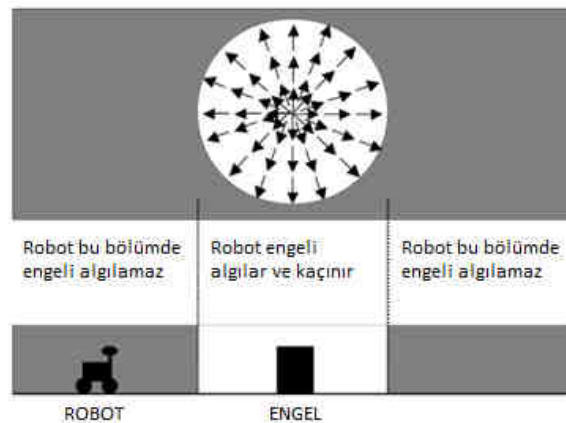
Sıradüzensel kontrolde, robot öncelikle ortamı algılar, daha sonra bir sonraki hamlesini planlar ve hareket eder. Herbir hamlede robot bir sonraki hamlesini planlar. Tüm algılanan veri, tek bir dünya modelinden toplanır ve ona bağımlı bir hal alır. Sıradüzensel yaklaşım, ayrımların her bir seviyesinde ortak bir şekilde problemin çözümünü ifade eder. Bu temelde robot kontrol mimarisi, eş zamanlı işlemler için davranışları temel alır. Bu yaklaşıma, blackboard tabanlı teknikler (Roth, et al., 1985) örnek gösterilebilmektedir. Bu yaklaşımın uygulamalarını kolaylaştırmak için STRIPS (Stanford Research Institute Problem Solver) yaygın olarak kullanılmıştır. Bu araç, Stanford Araştırma Enstitüsü tarafından geliştirilmiş bir problem çözme aracıdır. En güncel kullanımdaki hali LISP (LISP Programming Language, 2014) ile yazılmış şeklindedir. STRIPS, bir arama uzayı içerisinde bulunan dünyal modelleri içerisinde,

hayatta kalabilmesi için hızlı tepki verebilen çok basit hesaplama kabiliyetine sahip birimler kullanılmaktadır. Tepkisel kontrol yaklaşımını gerçeklemek için Potansiyel Alanlar Yaklaşımı ve Kapsama Yaklaşımı öne çıkan iki önemli yöntemdir.

2.3.1 Potansiyel alanlar

Potansiyel alan (Khatib, et al., 1985), gerçekleştirilecek davranışları tanımlamak için matematiksel vektörleri kullanan ve farklı davranışlar sonucu üretilen vektörlerin kombinasyonları sayesinde anlık bir şekilde tepkisel davranışlar üretebilen bir yapıya sahiptir. Potansiyel alanlar (Mark, et al., 2012), vektör alanlarının oluşturmuş olduğu yapay alanlardır. Potansiyel alan vektörleri, büyüklüğü ve yönü matematiksel olarak ifade edilebilen yapılardır. Basitçe bir “ok” ile temsil edilirler. Okun yönü potansiyel alanın yönünü, büyüklüğü ise potansiyel alanın kuvvetinin göstermektedir. Bir potansiyel alan vektörünün büyüklüğü 0 ile 1 arasında değişmektedir.

Ortam içerisinde algılanabilen nesneler, etrafında bulunduğu bölgeye sanal bir kuvvet uygulamaktadır (Khatib, et al., 1985). Robotun harita üzerinde noktasal olarak hareket ettiği düşünüldüğünde, o bölgelerden geçerken üzerine etki eden potansiyel alan açısı ve kuvvetini hissetmektedir. Şekil 2.3’te bir engelden çıkan potansiyel alan vektörleri gösterilmektedir.

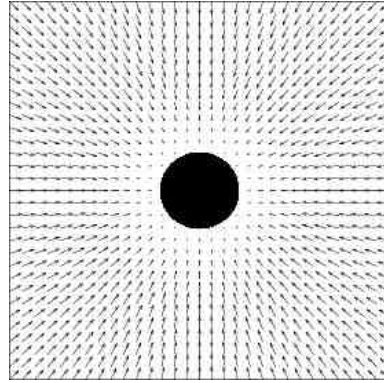


Şekil 2.3 Potansiyel alan vektörleri

Robot engele yeterince yaklaştığında, üzerine etki eden itici potansiyel alan kuvvetine maruz kalmaktadır. Her engelin etki ettiği mesafe, o engelin oluşturduğu itici potansiyel alanı oluşturmaktadır. Robot bu alan içerisine girmediği müddetçe üzerine herhangi bir kuvvet etki etmemektedir. Literatürde Çekici, İtici ve Teğetsel olmak üzere üç temel potansiyel alan kuvveti bulunmaktadır.

Çekici potansiyel alan;

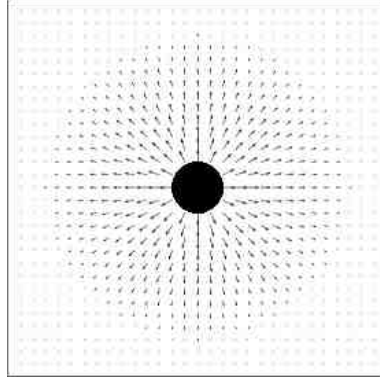
Çekici potansiyel alan, hedef noktasına doğru çizilen kuvvet doğrularının bileşkesinin oluşturduğu potansiyel alandır. Robotun ulaşmak istediği hedef noktalarından çıkan potansiyel alan vektörlerinin oluşturmuş olduğu potansiyel alandır. Şekil 2.4'te çekici potansiyel alan vektörleri gösterilmektedir.



Şekil 2.4 Çekici potansiyel alan vektörleri

İtici potansiyel alan;

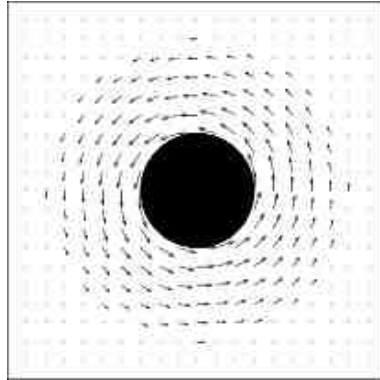
İtici potansiyel alan, engellerden dışarıya doğru çizilen kuvvet doğrularının toplamının oluşturduğu alandır (Bkz. Şekil 2.5).



Şekil 2.5 İtici potansiyel alan vektörleri

Teğetsel potansiyel alan;

Robotun etrafından dolaştığı alanları gösteren kuvvet çizgilerinin toplamının oluşturduğu alandır (Bkz. Şekil 2.6).



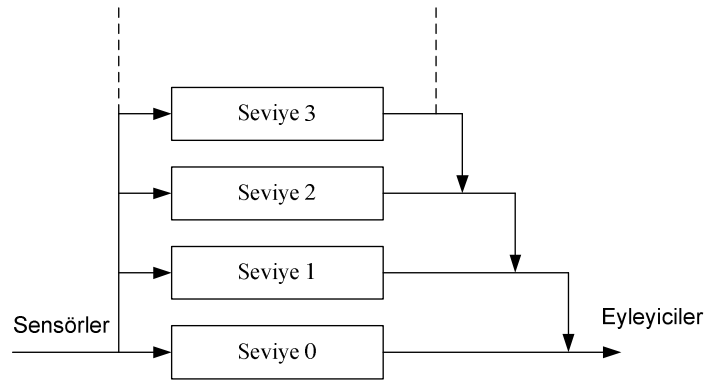
Şekil 2.6 Teğetsel potansiyel alan vektörleri

Potansiyel alanlar, uygulama geliştirme aşamasında kodlama basitliği açısından sıklıkla tercih edilmektedir. Ancak, algılayıcı verilerinden gelen veriler üzerinde herhangi bir düzeltme yapmadan, direkt olarak çıktı üretmeleri sebebiyle algılayıcıya bağımlıdırlar. Herhangi bir anda algılanan yanlış mesafe bilgileri beklenmeyen sonuçlara sebep olabilmektedir. Bunun yanında, bulunulan ortamın doğası gereği potansiyel alanlarda ortaya çıkan bazı problemler de bulunmaktadır. Bu problemlerden en fazla karşılaşılan yerel en az (local minima) problemidir. Bu problem, hedef nokta

dışında robot üzerine etki eden toplam bileşke kuvvetin büyüklüğünün sıfır olması ve robotun hiç hareket edememesi veya osilasyona girmesi durumudur. Robotun bu durumdan kurtulabilmesi için çeşitli yaklaşımlar ve algoritmik yöntemler mevcuttur (Siciliano, et al., 2008)

2.3.2 Kapsama yaklaşımı

Kapsama yaklaşımı (Subsumption Architecture) Brooks (Brooks, et al., 1986) tarafından geliştirilmiştir. Bu mimaride, kontrol sistemi katmanları seviyeler arasındaki yetki üstünlüğüne göre çalışmaktadır (Bkz. Şekil 2.7). Her bir modül kendisine ait bir görevi yerine getirmektedir. Yüksek seviye katmanlar düşük seviye katmanların giriş veya çıkışlarını bastırarak kapsayabilmektedir. Her bir yetki seviyesi, kendisinden bir önce gelen yetki seviyesini bir alt küme gibi içermektedir.



Şekil 2.7 Subsumption yöntemi

Birçok uygulamada, davranışlar doğrudan robotun donanımına ya da küçük mikroişlemcilere yüklenmektedir. Böylece, robotlar planlama işlemi olmaksızın hareket edebilmekte ve engelleri aşabilmektedir. Bu mimaride davranışlar seviyeler şeklinde tasarlanır. Öncelikle, ortamda rastgele dolaşma veya engelden sakınma gibi robotun en düşük seviyedeki basit davranışlar gerçekleştirilir. Duvar takibi veya koridorun ortasından gitme gibi daha karmaşık davranışlar, üst seviye davranışlar olarak, alt seviyedeki basit davranışların üzerine tasarlanır. Bunlar gerektiğinde alt

seviyedeki davranışın çalışmasını engeller ya da onun çalışmasını da kapsayarak karmaşık işlemi gerçekleştirir. Tepkisel yaklaşım, çok hızlı çalışma süresine sahip olmasına rağmen, hiçbir planlamanın kullanılmamasından dolayı birçok uygulamada tek başına yeterli olmamaktadır.

2.4 Melez Kontrol

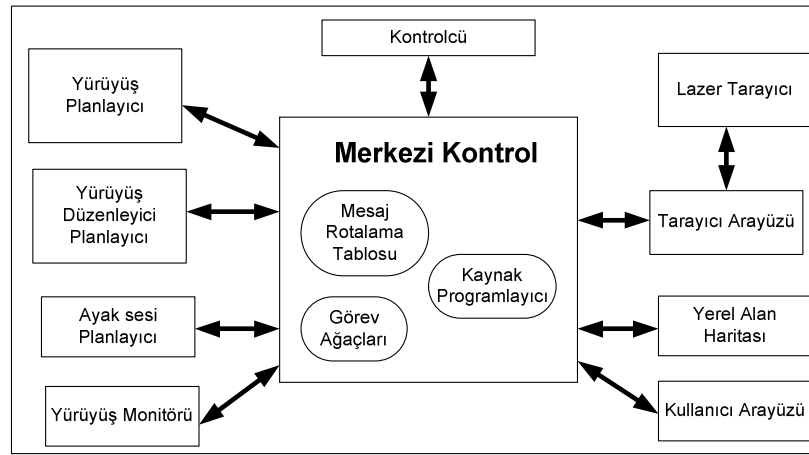
Bu yapıda algılama, planlama ve hareket koordineli olarak çalışmaktadır. Robot öncelikle görevi alt görevlere nasıl dağıtacağını planlar. Her alt görevin başarı ile tamamlanabilmesi için uygun davranışların hangileri olduğuna karar verir. Algılayıcı verileri robotun ihtiyaç duyduğu her bir davranışa girdi olarak yönlendirilir. Aynı zamanda, planlayıcı içinde bu algılayıcı verileri görev odaklı ortam modelinin çıkartılması için kullanılabilmektedir.

Melez kontrol yapısı içinde tepkisel ve planlama bileşenleri birbirleri ile çakışmazlar, ancak etkileşim içerisindedirler. Tepkisel bileşen, içinde bulunulan dünyada karşılaşılan ani ve beklenmedik durumlarda planlama bileşenin ürettiği planı aynen uygulayamayabilir. Planlama bileşeni de daha etkin hareket planlarını gerçekleştirebilmek adına tepkisel bileşene rehberlik yapabilir. Sistemdeki bu etkileşimi sağlayacak bir ara bileşen gerekmektedir. Bu yaklaşımı kullanan farklı tipte uygulamalar (Özkan vd., 2007) geliştirilmiştir. Takip eden alt bölümlerde melez yapıya sahip bazı öncü çalışmalar bulunmaktadır.

2.4.1 TCA

TCA (Task Control Architecture) (Siciliano, et al., 2008) yaklaşımında dağıtık robot sistemlerinin kontrolü için genel bir çerçeve önerilmektedir. Bu yapıda; gereksinim duyulan dağıtık haberleşmeler, görev dağılımları, kaynak yönetimi, yönetim izlenmesi ve hata giderilmesi mekanizmalarını yürütmek için tasarlanmış melez yapı

yüksek seviye robot kontrol sistemi bulunmaktadır. Şekil 2.8’de TCA kullanarak tasarlanmış örnek bir robot kontrol sistemi gösterilmektedir.

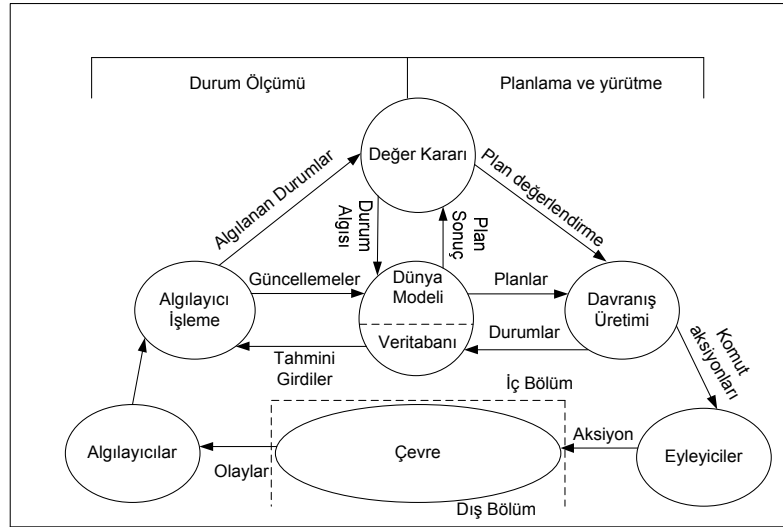


Şekil 2.8 TCA yaklaşımı

TCA yaklaşımı birçok yönü ile Blackboard mimarilerini (Albus, et al., 1989; Shieh, et al., 2004) muhafaza etmekte fakat bazı farklılıkları bulunmaktadır. TCA kontrol bilgisini merkezi olarak tutmakta iken, Blackboard mimarisinde kontrol bilgisi çözülmesi için sistemin süreçlerinin içerisinde dağıtılmaktadır.

2.4.2 NASREM

NASREM yapısında, ortam algılama, dünya modeli dikkate alarak değerlendirilme, planlama ve hareket etme koordineli olarak yapılmaktadır (Albus, et al., 1991; Albus, et al., 1989). Algılanan bilgiler o andaki durumun tutarlı bir görüntüsü elde edilene kadar birkaç işlem basamağından geçmektedir. Kontrol mimarisi, bu işlemten sonra, sürücüler tarafından istenilen hareketlerin direkt olarak çalıştırılabilmesi için bir plan geliştirilmesi ve diğer birimler tarafından dağıtılması üzerine kurulmuştur. NASREM mimarisinin genel yapısı Şekil 2.9’da gösterilmektedir.

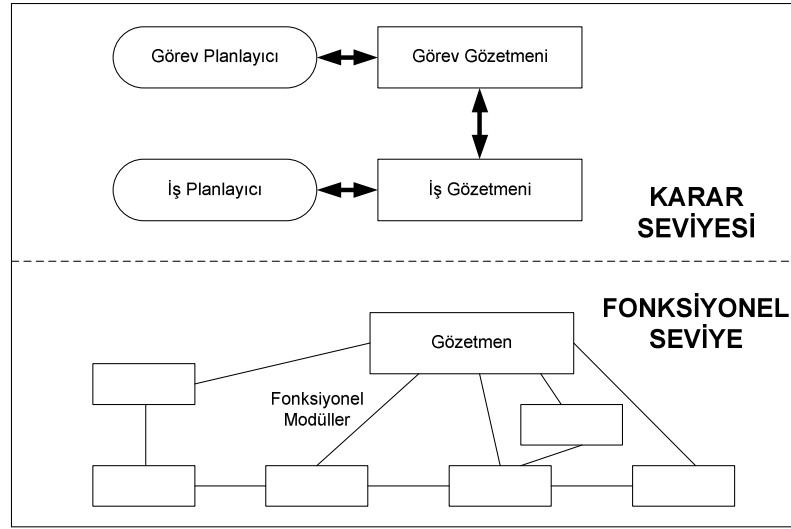


Şekil 2.9 NASREM yapısı

2.4.3 LAAS

LAAS (Laboratory for Analysis and Architecture of Systems) yaklaşımı ikisi karar verme katmanları, biri fonksiyonel katman olmak üzere üç katmandan oluşmaktadır (Bkz. Şekil 2.10). Karar verme katmanları, genellikle PRS (Alami, et al., 1998) tabanlı çalışmaktadır. Görevleri olarak senaryoları oluşturur ve senaryoların çalıştırılmasını yönetir. Fonksiyon katmanı ise temel robot görevlerinin gerçekleştirilmesi, servo motor kontrolleri ve basit robot fonksiyonlarının kontrol eder.

Fonksiyon katmanı bir takım modülleri içerisinde toplar. Bir modül ortak verileri veya kaynakları paylaşan basit robot fonksiyonlarının içerisinde gömülüdür. Bu katman yönetici tarafından yönetilir ve kontrol edilir (Elkady, et al., 2012). Modüller birbirleri ile mesajlaşma, diğer modüllerden gönderilen veriyi okuma işlemlerini gerçekleştirmektedir.

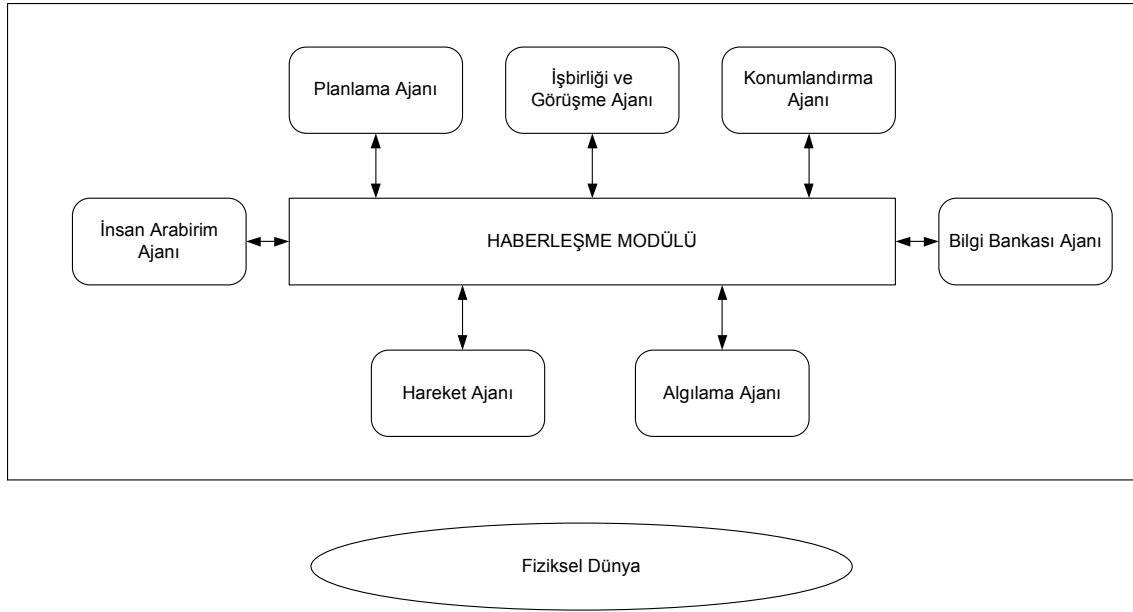


Şekil 2.10 Örnek LAAS yaklaşımı

2.4.4 Ajan tabanlı yaklaşım

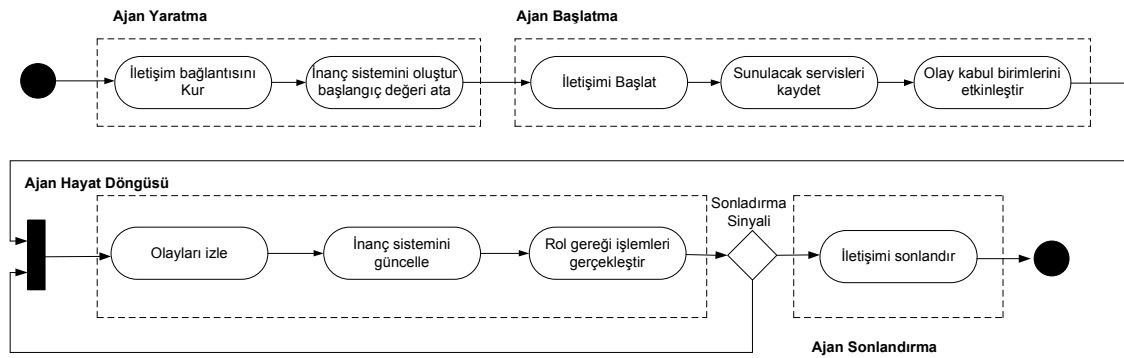
Bu yapıda, farklı rollere sahip ajanlar (erkinler), bir gezgin robotun sahip olması gereken işlevleri yerine getirmektedir. Ajanların sahip oldukları rollerin kapsamını daraltarak sistemdeki ajan sayısı artırılabilir. Örneğin; planlama ajanı, hareket planlama ajanı, yol planlama ajanı ve görev planlama ajanı şeklinde parçalanabilir.

En çok bilinen yapı, Stanford Araştırma Enstitüsü Yapay Zekâ Merkezi (Stanford Research Institute, 2014) tarafından geliştirilen açık ajan mimarisidir (Open Agent Architecture – OAA). Bu yapı geliştiriciye bir ajan geliştirme dili ve alt yapısı sağlamaktadır. Bu dille, ajanlar geliştirildikleri programlama dillerinden bağımsız olarak birbirleri ile haberleşebilmektedir. Bu dil, Prolog dilinin bir uzantısı olarak geliştirilmiştir. Şekil 2.11’de OAA genel yapısı gösterilmektedir.



Şekil 2.11 OAA genel yapısı

Robotu meydana getiren her ajan, çalışma sürecinde Şekil 2.12’de gösterildiği gibi dört temel evreye sahiptir (Özkan vd., 2007).



Şekil 2.12 Ajan çalışma mekanizması

İlk evre ajanın yaratılmasıdır. Bu evrede, haberleşme modülü ile bağlantı kurulur ve ajanın iç ve dış dünyası ile ilgili doğruluğuna inandığı bilgiler oluşturulur. Daha sonra, ajan başlatma evresi gerçekleştirilir. Bu evrede, robot iletişime başlar. Kendisine atanan rolü gereği gerçekleştirebileceği hizmetlerin kaydını haberleşme modülüne yaptırır. Haberleşme modülü bu bilgileri, diğer ajanlardan gelecek talepleri

ilgili yerlere ulařtırmakta kullanır. Ayrıca, ajanın grup içerisinde ilgilendiđi olaylar için kabul birimlerini etkinleřtirir.

Ajan hayat döngüsünde, ajanın sürekli bir döngü içinde gerçekleřtirdiđi işlemler yer almaktadır. Ajanın, ilgilendiđi çevre deđişimleri veya hizmet talepleri olup olmadığını iletişim kanalından izler. İnanç sistemindeki bilgileri günceller ve rol geređi gerçekleřtirilecek olan ve her ajan için farklılık gösteren işlemleri yapar. Bu evre, sistemden sonlandırma sinyali alana kadar sürekli devam eder.

Her bir ajan, haberleřme kanalından diđer ajanlarla iletişim halindedir ve sahip olduđu görevleri yerine getirmektedir. Eđer, sistemden ajanı sonlandırma sinyali gelirse, ajanın iletişimi sonlandırılır ve ajan yok edilir. Bu evrede, haberleřme modülü de ajanın sunduđu hizmetleri listesinden çıkartır. Böylece, diđer ajanlar bu hizmetler için beklemes ve mümkünse, diđer kaynaklardan bu hizmeti almaya devam eder.

2.4 Davranış Tabanlı Kontrol

Davranışlar tabanlı kontrol (Slack, 1993) gezgin robot kontrolünde kullanılan diđer önemli bir yapıdır. Gezin robot kontrol sistemlerinin oluřturulmasında, davranış yaklaşımının kullanılmasının yararları vardır. Davranışların modüler olmasından dolayı bağımsız olarak test edilmeleri kolaydır. Ayrıca, sistemdeki mevcut davranışları tutarak, yeni davranışları ekleme ile ölçeklenebilir ve genişletilebilir bir yapı sağlamaktadır.

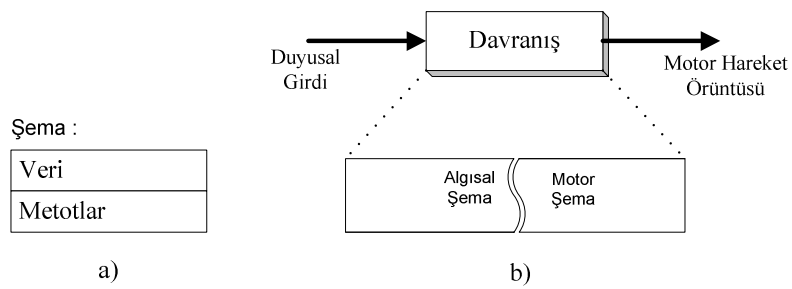
Davranışların kavranmasında hayvanların hareketlerinin gözlemlerinden faydalanılabilir. Örneğin; yırtıcı bir hayvanı fark eden bir atın yaptıđı hareket örüntüsü bir davranış olarak tanımlanabilir. At yırtıcı hayvanı görünce, kulaklarını dikleřtirir, başını eđer ve ayađı ile yeri eşelemeye başlar. Burada, yırtıcının görülmesi bir algısal davranış tetikleyicisidir. Grafiksel olarak Şekil 2.13'te gösterilen davranış, algısal girdilerin bir görevi gerçekleřtirmek üzere kullanılan motor hareketleri örüntüsüne

dönüştürülmesidir. Matematiksel bakış açısından, algısal girdileri hareket komutlarına aktaran bir aktarım fonksiyonudur.



Şekil 2.13 Davranış genel yapısı

Davranışların gezgin robotlarda uygulanmasında Şema (Schema) Teorisi yaygın olarak kullanılmıştır. Önceleri psikologlar tarafından kullanılan bu teori daha sonra gezgin robotların programlanmasında kullanılmıştır. Şemalar, bir aktiviteyi vurgulama yoludur. Yapısında bir aktivitenin oluşması için bilgi ve hesaplama süreçleri bulundurulur. Temel olarak yapısı, nesne tabanlı programlamadaki sınıf yapısı ile özdeşleştiği için robot davranışlarının programlanmasında çok kullanışlı olmaktadır. Şekil 2.14'te bir şemanın yapısının gösterimi ve şemalara ayrıştırılmış bir davranış yapısı görülmektedir.



Şekil 2.14 Davranışın şema olarak gösterimi: a) şemanın yapısı, b) algısal ve motor şema olarak ayrıştırılmış davranış yapısı

Davranış tabanlı kontrol (Mark, et al., 2012) hızlı ve tepkisel olması açısından tepkisel kontrol mimarisine yakın bir mimaridir. Tepkisel mimariden farklı yanı ise

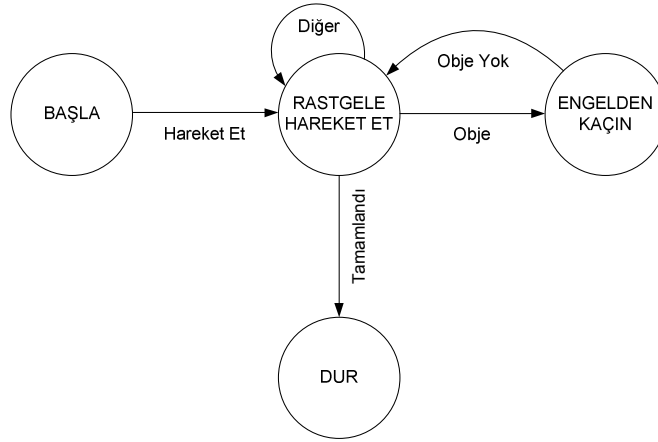
öğrenme kabiliyetine sahip olmasıdır. Davranış, algılama ve hareket arasında kalan bir yapıdır.

Davranışlar uygulamaya bağlı olarak farklı düzeylerde karmaşık yapılara sahip olabilmektedir. Örneğin; engelden kaçınma davranışı anlık bir davranış olarak çalışırken, bir bölgeye yanaşma davranışı zaman alabilen bir davranıştır. Ancak, bu iki davranış aynı zamanda çalışabilirler. Çizelge 2.1’de bazı örnek davranışlar ve sınıflandırılması verilmiştir.

Çizelge 2.1 Davranışların sınıflandırılması

Sınıflandırma	Tanım	Davranış Tipi
Keşif	Genel bir yönde hareket	Rastgele Hareket
Hedef Tabanlı	Bir noktaya doğru hareket	Rota Tayini
İtici	Çarpışma önleme	Engelden Kaçınma
Yol Takibi	Belirli bir yol üzerinde hareket	Duvar Takibi
Sosyal İşbirliği	Beraber hareket	Toplanma

Davranış tabanlı kontrol mimarisinde birden fazla davranışın koordineli çalıştırılması için alternatif karar verme mekanizmaları kullanılmaktadır (Siciliano, et. al, 2008). Örneğin; Şekil 2.15’te rastgele dolaşma ve engelden sakınma davranışlarının sonlu durum otomatu (Finite State Automata, FSA) ile birleştirilmesi gösterilmektedir. Burada Başlangıç, robotun ilk başlayacağı durumu temsil etmektedir. Robot işi bitirdiğinde ise duracağı durum Bitiş ile gösterilmektedir. Eğer robot bir engel algılsa, kaçınma durumuna geçer. Bu iki durumun dışında, rastgele durumunda hareketine devam etmektedir. Bu diyagram farklı davranışlar arasındaki ilişkiyi basit bir şekilde göstermektedir.



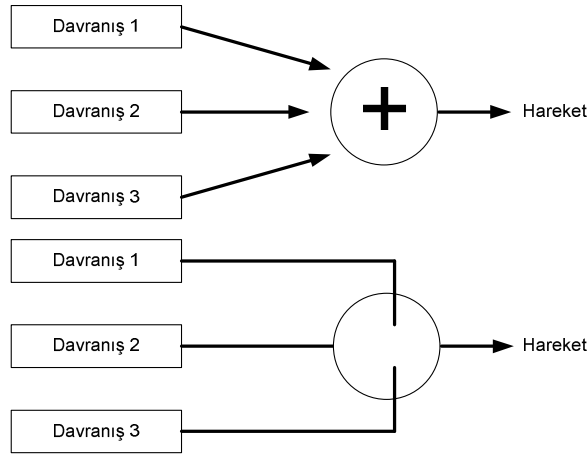
Şekil 2.15 Davranışların sonlu durum otomatu ile birleştirilmesi

Davranışlar arası koordinasyonu sağlayan diğer bir yapı da Stimulus Response Diagram (SRD) (Mark, et al., 2012) yapısıdır. Bu yapı, davranışların kendilerine atanan bir işlemi gerçekleştirmek üzere sergiledikleri sıralı davranışları göstermektedir. Şekil 2.16’da örnek bir SRD diyagramı gösterilmektedir.



Şekil 2.16 Örnek bir SRD diyagramı

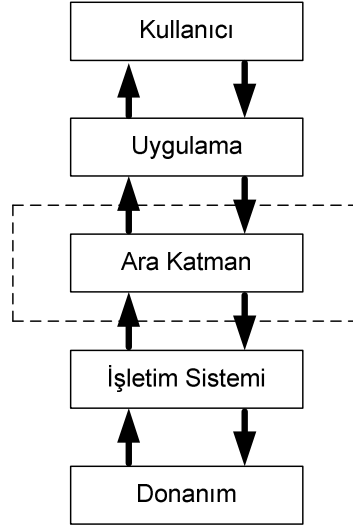
Bazı durumlarda davranışların birleştirilmesi ile daha karmaşık davranışların elde edilmesi gerekebilmektedir. Buna davranış birleştirme (fusion) işlemi adı verilmektedir. Şekil 2.17’de davranış birleştirme işlemi gösterilmektedir.



Şekil 2.17 Örnek bir davranış birleştirme

2.5 Robotik Ara Katmanlar

Gezgin robotların özerk kontrolünde farklı algılayıcılardan veri alınması, bunların farklı yazılım ortamı veya davranışlar içerisinde değerlendirilmesi ve uygulamaya alınması gerekmektedir. Ayrıca, bu alanda çalışma yapan araştırmacıların farklı robot platformları üzerinde benzer problemleri çözmeye yönelik çalışması, donanımdan bağımsız ortak bir robotik kütüphanesi geliştirilmesi ihtiyacını doğurmuştur. Bu noktada robotik ara katman kavramı ortaya çıkmaktadır. Robotik ara katmanlar (Elkady, et al., 2012), işletim sistemi ve uygulamaya yazılımı arasında kalan ve farklı platformlarda rahatlıkla kullanılan soyutlama katmanlarıdır. Ara katmanlar donanım çeşitliliğini yönetmenin yanı sıra, yazılım tasarımını kolaylaştırmak, uygulama kalitesini arttırmak ve geliştirme maliyetlerini düşürmek için de kullanılmaktadır. Şekil 2.18’de robotik ara katmanın yeri gösterilmektedir.



Şekil 2.18 Robotik ara katman

İyi bir robotik ara katman, uyarlanabilir ve geliştirilebilir bir yapıya sahip olduğu gibi, kolay anlaşılır, esnek, gürbüz ve gerçek zamanlı işlemlerin geliştirilme kabiliyetine sahip olmalıdır. Günümüzde, hem açık kaynaklı hem de ticari olan robotik ara katmanlar bulunmaktadır. Aşağıda bunların öne çıkanları verilmektedir.

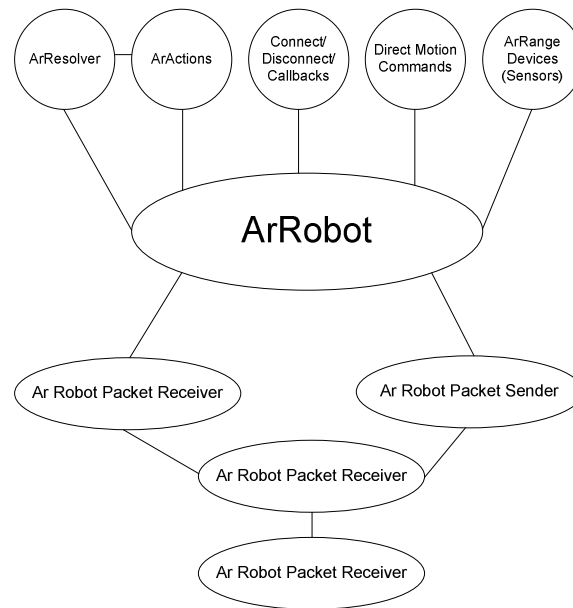
2.5.1 ARIA

ARIA, farklı yelpazede gezgin robot platformlarında kullanılan açık kaynak kodlu bir robotik ara katmandır. Robot kontrolünde, ihtiyaç duyulan çok sayıda uygulama geliştirme kütüphanesi bulunmaktadır. Bunlardan bir kısmı robot üretici firmaları tarafından sadece belirli bir robot sistemi için geliştirilmişken (Active Media Inc, 2014), bazıları ise, farklı üreticilerin robotlarını destekleyen uygulama geliştirme araçlarıdır (Aria Manuals ver. 2.5.1, 2014).

ARIA nesne tabanlı olarak geliştirilmiş olup, C++ programlama dilinde yazılmıştır. LISP, PROLOG gibi ara yüzleri bulunmaktadır. ARIA yazılım kütüphaneleri kullanılarak yazılan programlarla robotun hızı, yönlendirilmesi, bağlı yönlendirilmesi ve birçok seyrüsefer kontrolü gerçekleştirilebilmektedir. Odometri,

algılayıcı verileri ve diğer sistem verilerine de ulaşılabilir. Ayrıca, trigonometrik işlemler, geometrik işlemler, uzaysal dönüşümler ve zamanlama işlemleri gibi doğrudan robotun kontrolü ile ilgili olmayan, ancak programlama esnasında ihtiyaç duyulabilecek fonksiyonları da uygulama kütüphanesinde mevcuttur.

ARIA yazılım kütüphaneleri, hem Linux ve hem de Windows işletim sistemlerinde çalışacak programların geliştirilmesinde kullanılabilmektedir. C++ ile yazılmış olmasına rağmen, Java ve Python programlama dillerinde geliştirilecek uygulamalarda da ARIA yazılım kütüphanesi kullanılabilmektedir. ARIA yazılımının mimari yapısı Şekil 2.19’da görülmektedir.



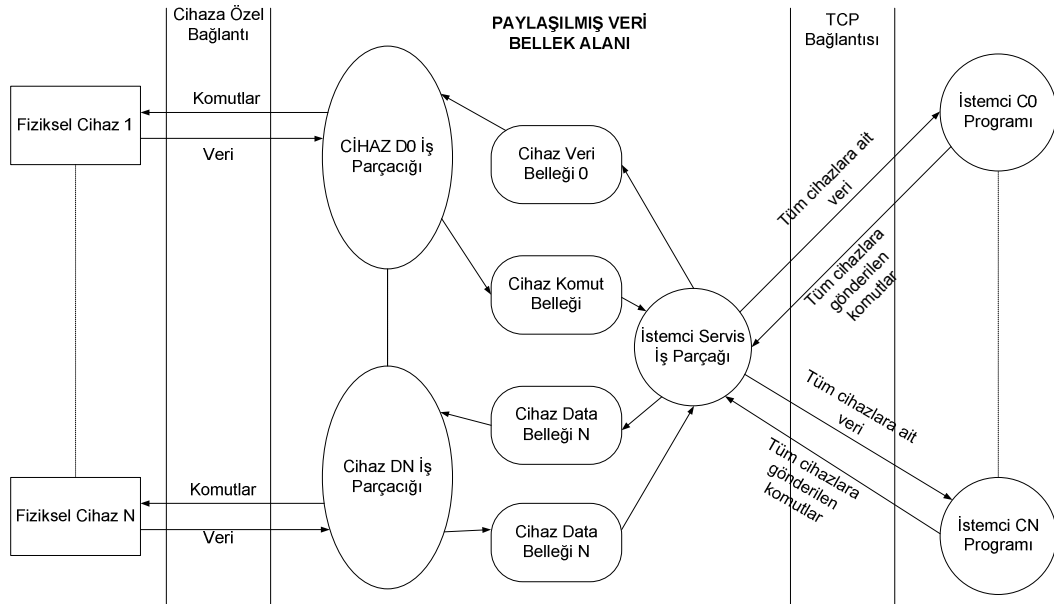
Şekil 2.19 Aria yazılım mimari yapısı

Yapıda, omurgayı ArRobot sınıfı oluşturmaktadır. Robot benzetimi ya da robot düşük seviye kontrolünü gerçekleştiren mikro denetleyici kartı ile bağlantı ve paket alışverişi ArRobot Device Connect, Packet Receiver ve Packet Sender nesneleri ile sağlanmaktadır. Ana nesnenin üstünde de, hareket komutları, algılayıcı yönetimi ve davranış tabanlı programlama için robot programlayıcısına sunulan ara yüz nesneleri bulunmaktadır.

2.5.2 PLAYER

PLAYER, açık kaynak kodlu bir ara katman olup farklı araştırmacıların katkıları ile sürekli geliştirilmiştir. Robot platformlarından bağımsız olması nedeniyle de, çok yaygın olarak robot kontrolünde kullanılmaktadır. PLAYER (The Player Project, 2014), bir robotik ara katman olup x86/Linux işletim sistemleri üzerinde geliştirilmiştir.

PLAYER, bilgisayar işletim sistemlerine de benzetilebilir. İşletim sistemleri, bilgisayar donanımını kullanıcıdan saklayarak, kullanıcıya kolay yönetilebilir bir ara yüz sağlamaktadır. PLAYER da, robot aygıtlarının detaylarını saklayarak robot programcısına bir ara yüz sağlamaktadır. PLAYER sistem mimarisinin gösterimi Şekil 2.20’de verilmektedir.



Şekil 2.20 Player yazılım mimari yapısı

Görüldüğü gibi şeklin sol tarafında robot aygıtları ve sağ tarafında ise kullanıcı programları yer almaktadır. Ortadaki yapı, kullanıcı ile aygıtlar arasındaki iletişimi yönetmektedir. Bu mimari yapı, bir robot kontrol yazılımının farklı robot platformları üzerinde çalışmasına olanak sağlamaktadır. Örneğin; robotun bir metre ileri yönde gitmesi istendiğinde, bu istek her robot için aynıdır. Ancak, farklı firmaların ürettiği robotların bu işlemi gerçekleştirmesi için gerekli düşük seviye programlamaları aynı olmayacaktır.

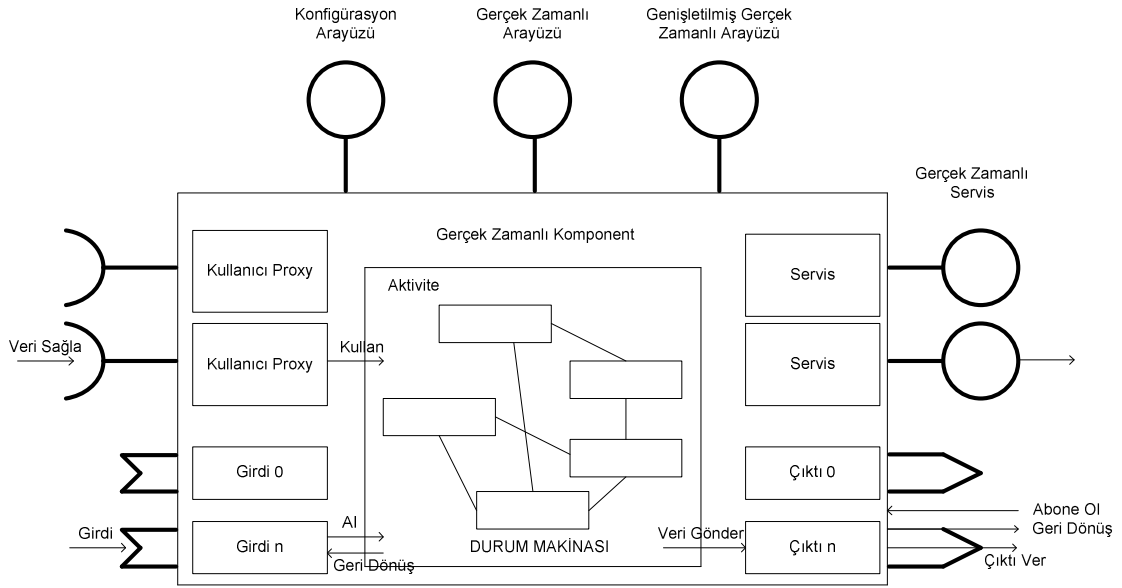
2.5.3 OROCOS

OROCOS, robot geliştiricilere gerçek zamanlı yazılım geliştirmek için gerekli araçları sunan bir ara katmandır (Elkady, et al., 2012; The OrocOS – Open Robot Control Software, 2014). OROCOS; Bileşen (Component), Kinematik – Dinamik (Kinematics – Dynamics) ve Filtreleme ve Filtreleme (Filtering) olmak üzere üç temel kütüphaneye sahiptir. Bu kütüphaneler ile farklı görevleri gerçeklemek üzere QNX (QNX Real Time Operating System, 2014) gibi gerçek zamanlı işletim sistemlerinde gerçek zamanlı olarak davranışlar geliştirilebilmektedir. Yazılım geliştirme, nesneye dayalı programlama temelli C++ dili ile gerçekleştirilmektedir.

2.5.4 OPENRTM-aist

OPEN RTM-aist (Open Real Time Middleware) (OPENRTM-aist Middleware, 2014) komponent tabanlı, gerçek zamanlı davranış geliştirmeyi sağlayan bir ara katmandır. Bir komponent, herhangi bir cihaz ünitesi, örneğin; servo motor, mesafe algılayıcısı olabileceği gibi aynı zamanda cihazların oluşturduğu daha kompleks bir yapı, örneğin robot kolu, sensör dizisi veya yazılım modülü (Kontrol algoritması, görüntü işleme algoritması) de olabilmektedir (Elkady, et al., 2012). Her bir komponentin kendisine ait ara yüzü bulunmaktadır. Bu portlar ile komponentler kendi aralarında bilgi alış-verişini gerçekleştirmektedir. Aynı şekilde karmaşık yapılar bu portların birleştirilmesiyle ortaya çıkmaktadır. Şekil 2.21’de port yapısı

gösterilmektedir. Bu gerçek zamanlı ara katman ile QNX, ART-Linux (ART-Linux, 2014), T-Kernel (T-Kernel, 2014) ve VxWorks (VxWorks, 2014) gibi gerçek zamanlı işletim sistemlerinde, gerçek zamanlı davranışlar geliştirilebilmektedir.



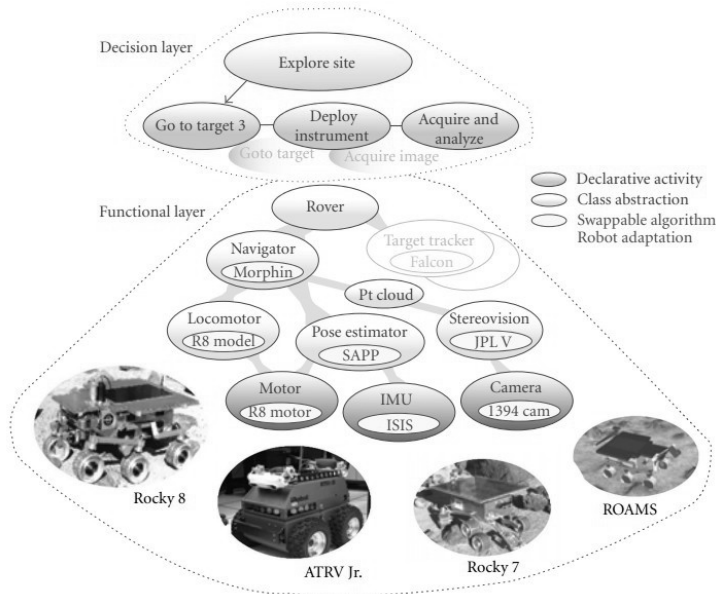
Şekil 2.21 RTM port yapısı

2.5.5 CLARATY

İçerisinde gerçek zamanlı işletim sistemi, haberleşme ara katmanı bulunduran bir ara katmandır (Elkady, et al., 2012). Fonksiyonel ve Düşünsel katmanları bulunmaktadır. Fonksiyonel katmanda genel komponentleri (dijital&analog I/O, motor kontrolü, hareket (locomotion), işletme (manipulation), görüntüleme, haritalama, yüzey oluşturma, yol planlama, yakınsama, benzetim, sistem davranışları) içerilmektedir (Elkady, et al., 2012). Kontrol algoritmaları bu katmanda gerçekleştirilmektedir.

Düşünsel katman, sistemin anlık durumu, kaynakların seviyesi ve görev kısıtları için genel bir motor görevi görmektedir. Genel planlayıcı, programlayıcı, aktivite veritabanları vb. kaynakları içerir. Tüm sistemin akış diyagramlarının çıkartılabildiği katmandır. Bu ara katmanın bir diğer özelliği de, gerçek zamanlı işletim sistemi olan

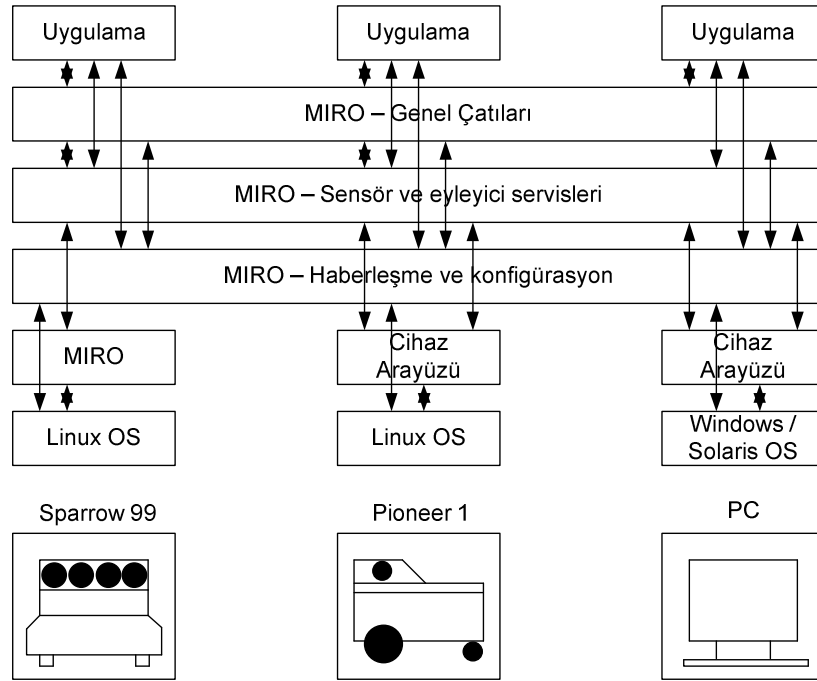
QNX gerçek zamanlı işletim sistemi ile beraber çalışabiliyor olmasıdır. Şekil 2.22’de CLARAty mimarisi gösterilmektedir.



Şekil 2.22 CLARAty mimarisi

2.5.6 MIRO

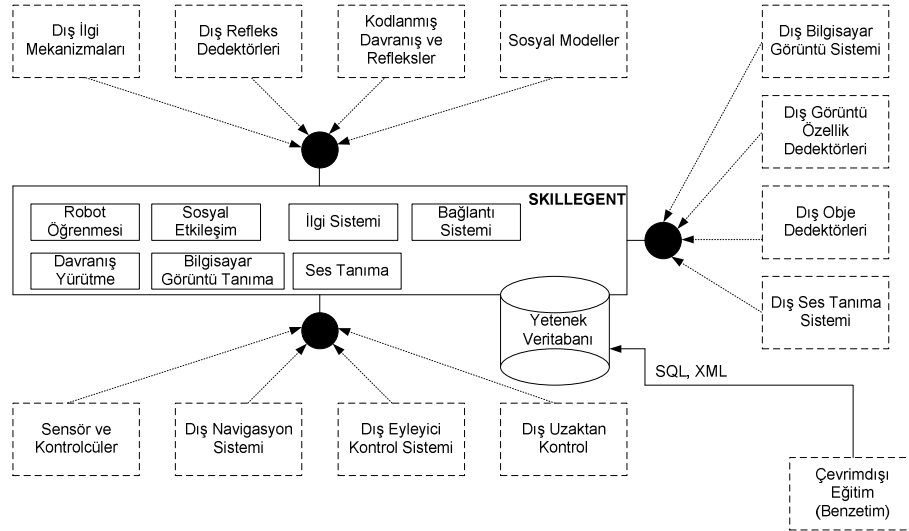
MIRO (MIRO – Middleware for Robots, 2014) ULM (The University of Louisiana) üniversitesi tarafından geliştirilmiş nesne tabanlı bir ara katmandır (Elkady, et al., 2012). Aygıt, servis ve ana yapı olmak üzere üç adet katmandan oluşmaktadır (Bkz. Şekil 2.23). Aygıt katmanı, düşük seviye donanım detaylarının gizlendiği platform bağımlı bir soyutlama ara yüzüdür. Servis, algılayıcı ve eyleyici için servis soyutlaması işlemlerinin gerçekleştiği katmandır.



Şekil 2.23 MIRO soyutlama katmanı

2.5.7 SKILLIGENT

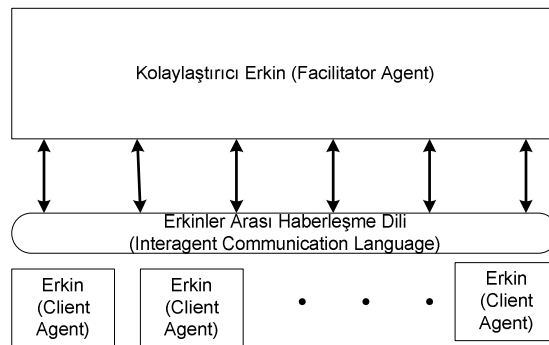
SKILLIGENT (Skilligent – Task and Skill Learning for Multifunctional Service Robots, 2014) modüler tabanlı bir ara katmandır. SKILLIGENT, robotik öğrenme ve davranış kontrol sistemleri üzerine geliştirilmiştir (Bkz. Şekil 2.24). Bu işlerin yapılabilmesi için içerisinde ses tanıma ve öğrenme, davranış işletme ve koordinasyonu gibi birçok modül barındırmaktadır. Modüler yapısı sayesinde modifiye edilebilir bir özelliğe sahiptir. Üzerine eklenmek istenen yapıyı ve modülerliği sağladığı müddetçe içerisine eklenmesine izin vermektedir.



Şekil 2.24 SKILLEAGENT farklı yazılım modüllerinin eklenmesi işlemi

2.5.8 OAA

Stanford Araştırma Enstitüsü Yapay Zekâ Merkezi tarafından geliştirilen bu ara katman (Open Agent Architecture, 2007), ajan (erkin) tabanlı programlamada kullanılmak üzere geliştirilmiştir. Bu yazılımda kullanılmak üzere, ajanlar arası haberleşme dili (ICL) geliştirilmiştir. Bu dille, ajanlar geliştirildikleri programlama dillerinden bağımsız olarak birbirleri ile haberleşebilmektedir. Bu dil, Prolog dilinin bir uzantısı olarak geliştirilmiştir. Şekil 2.25'te açık ajan mimarisinin temel yapısı gösterilmektedir.



Şekil 2.25 OAA genel yapısı

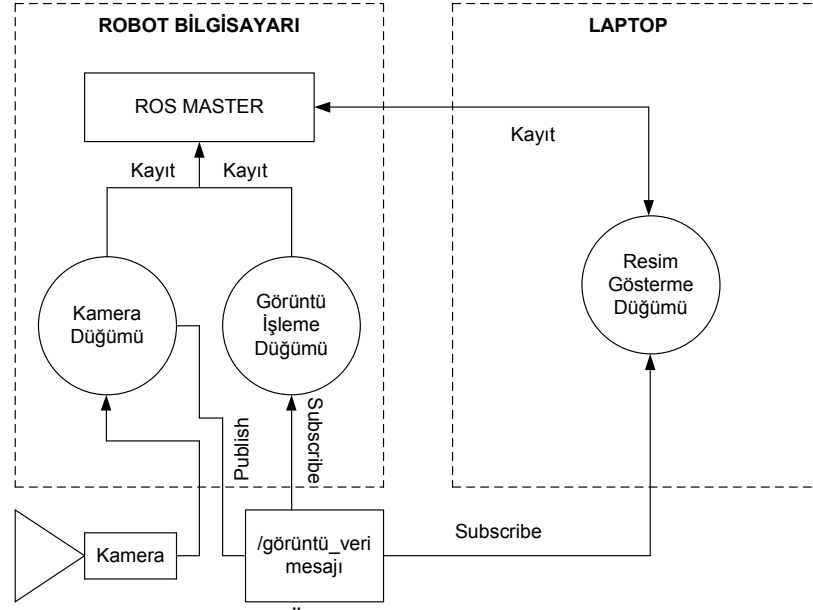
2.5.9 ROS

Robot işletim sistemi (Robot Operating System, ROS), (ROS Project, 2014) robot yazılımı geliştiriciler için kütüphane ve araçlar sağlayan, açık kaynaklı bir ara katmandır. ROS, kullanıcıya donanım soyutlama, cihaz sürücüler, yazılım kütüphaneleri, mesajlaşma ve paket yönetimi gibi özellikler sunmaktadır. Ubuntu 12.04 işletim sistemin üzerinde tüm özellikleriyle çalışabilmektedir. Bunun yanında Fedora, OS X, Windows, Gentoo ve Debian, Android gibi işletim sistemleri için deneysel olarak kullanılabilir.

Uygulama alanı oldukça geniş olup, gezgin robotlar, robot manipulatorleri (endüstriyel robot kolları), özerk arabalar, insansız robotlar ve insansız hava araçları için kullanılmaktadır. ROS, piyasada kullanılan birçok robotik algılayıcı için içerisinde kütüphaneler içermektedir. Yazılım geliştirici, ROS un gerekli kütüphanelerini kullanarak bu algılayıcılardan veri okuyabilmektedir.

Açık kaynaklı bir yazılım olmasından dolayı kullanıcı kendi geliştirmiş olduğu algılayıcılar için de ROS kütüphaneleri yazıp ortak kullanıma açabilmektedir. ROS' un diğer robotik ara katmanlardan farkı, robotik alanında çok geniş yelpazede sistem için yaygın olarak kullanılmasıdır.

ROS, açık ajan tabanlı dağıtık kontrolü gerçeklemek için geliştirilmiş bir platformdur (Bkz. Şekil 2.26). Bu yapıda, ROS içerisinde belli bir görev için çalışan alt işlemler düğüm olarak kurgulanabilmektedir. Tüm düğümler bir çizge (graph) yapısı şeklinde birbirleriyle haberleşmektedir. Düğümler arası haberleşme publish – subscribe mantığında çalışmaktadır.

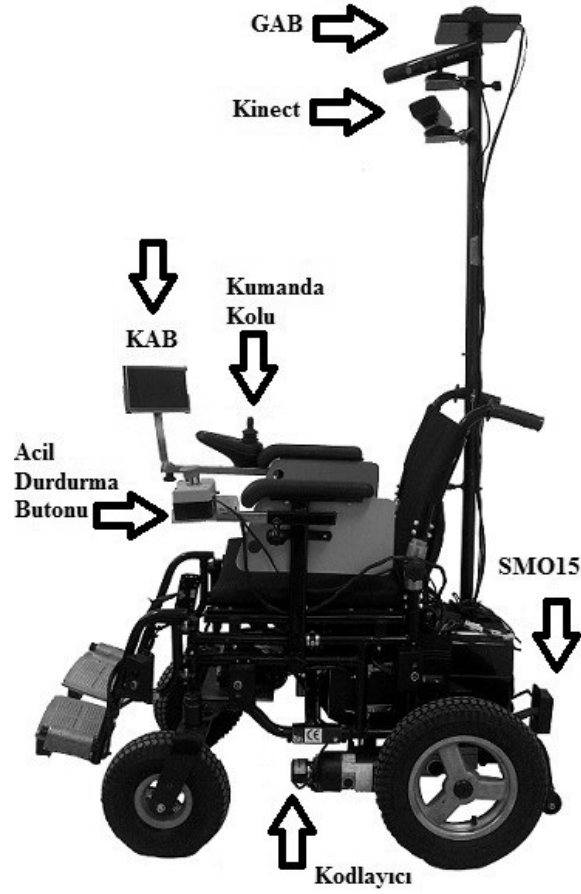


Şekil 2.26 Örnek bir ROS çalışma yapısı

BÖLÜM 3

AKILLI TEKERLEKLİ SANDALYE PLATFORMU

Akıllı TEKERlekli Sandalye (ATEKS) platformu, öncelikle hareket engelli insanların kullanımında özerk hareket edebilen bir platform olarak geliştirilmiştir. Bu mod özerk mod olarak adlandırılmaktadır. Platformun ayrıca Kullanıcı Modu ve Sürüş Destek Modu bulunmaktadır. Kullanıcı modunda, platform tamamen kullanıcının kontrolüne bırakılmıştır. Sürüş destek modunda, kullanım sırasında algılayıcılardan alınan bilgiler ile herhangi bir çarpışma ihtimali oluştuğunda ATEKS otomatik olarak durdurulmaktadır. Ayrıca, platform özerk modda gezgin robot eğitim ve araştırmalarında (Akar vd., 2013; Akar vd., 2014 a; Bayar vd., 2014 b; Özçelikörs vd., 2014) kullanılacak temel bileşenlere sahiptir. ATEKS platformu Şekil 3.1’de verilmiş olup, robotik araştırmalarda yaygın kullanılan ROS ara katmanı ile uyumlu olarak geliştirilmiştir.



Şekil 3.1 ATEKS sistemi genel yapısı

ATEKS mekanik platformu hazır alınmış olup, üzerine çevreyi algılamak için gerekli algılayıcı birimleri ve özerk kontrolü gerçekleyecek akıllı kontrol birimleri entegre edilmiştir.

3.1 Tekerlekli Sandalye Standart Donanımı

Tekerlekli sandalye standart donanımda, PG Drive kumanda kolu – motor sürücüsü ve gerekli itkiyi sağlayan fırçasız DC motorlar bulunmaktadır.

3.1.1 Kumanda kolu

Kumanda kolu, sandalyenin istenen yöne gidebilmesini sağlayan kullanıcı arayüzü birimidir. Üzerinde beş farklı ayarlanabilir hız kademesi ve çevreyi uyarabilmek için bir adet korna bulunmaktadır. Ayrıca, kumanda kolunda tekerlekli sandalyenin şarj göstergesi bulunmaktadır. Şekil 3.2’de tekerlekli sandalye üzerindeki kumanda kolu gösterilmektedir.



Şekil 3.2 Kumanda kolu

3.1.2 Motor sürücüsü

Şekil 3.3’te gösterilen PG Drive VR2 motor sürücüsü, tekerlekli sandalyenin üzerindeki iki adet motoru sürmektedir. Bu motorlar sağ ve sol tekere bağımsız hareket sağlamaktadır. Tekerlekli sandalye bu yapı itibarıyla, diferansiyel (farksal) sürüş tipine sahiptir.

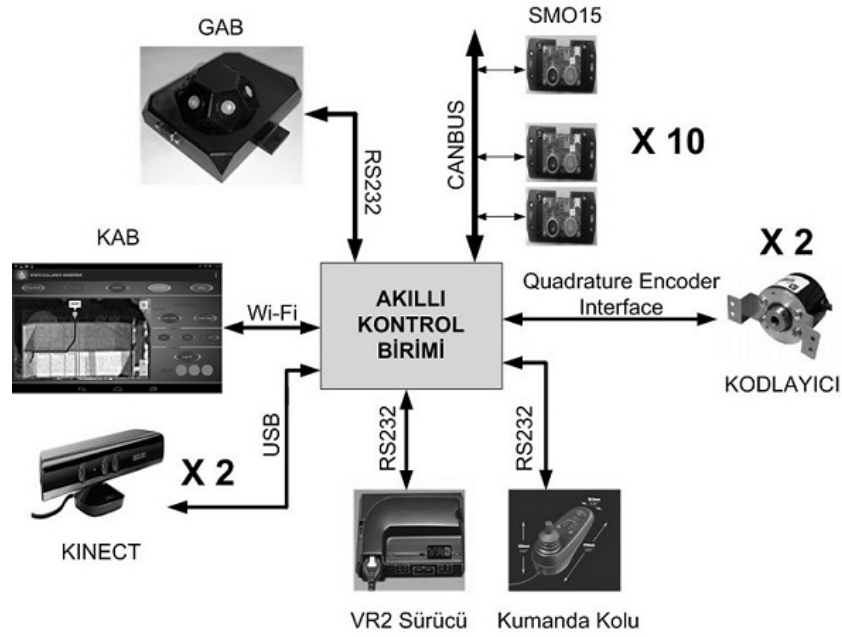


Şekil 3.3 Motor sürücü VR2

Kumanda kolu ve motor sürücü arasında özel bir haberleşme protokolü vardır. Sistemin çalışması için bu haberleşme yapısının sürekli olması gerekir. Bu haberleşmenin belli bir süre kesintiye uğraması durumunda sandalye arıza moduna geçer ve durdurulur. Herhangi bir arıza durumu, kumanda kolu üzerindeki göstergelerden kullanıcıya gösterilmektedir.

3.2 Akıllı Kontrol İçin Entegre Edilen Çevre Birimleri

Tekerlekli sandalye platformuna özerk kontrol ve sürüş destek modunu gerçeklemek için ortam algılama amaçlı değişik algılayıcılar (Microsoft Kinect, SMO15, Konum algılama birimi – GAB, Bağlı konum birimi – Kodlayıcı) alınan verilerin değerlendirilmesi, kontrol için Akıllı Kontrol Birimi (AKB) ve kullanıcı ile etkileşim için Kullanıcı Arayüzü Birimi (KAB) entegre edilmiştir. Şekil 3.4'te tekerlekli sandalye temel platformuna entegre edilen çevre birimleri gösterilmektedir.



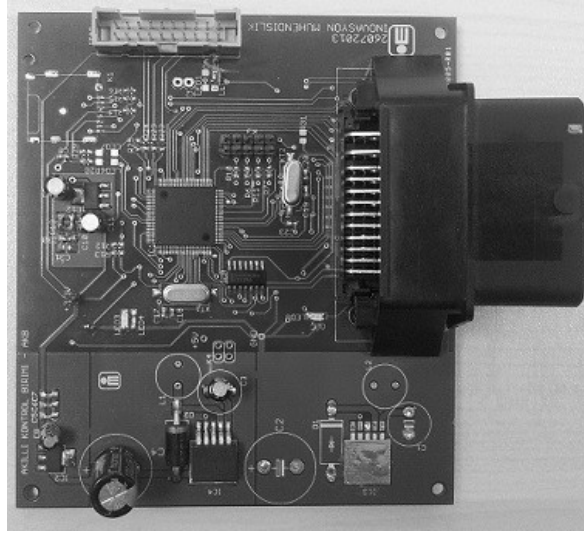
Şekil 3.4 ATEKS çevre ve kontrol birimleri

3.2.1 Akıllı kontrol birimi

Akıllı kontrol birimi (AKB), ATEKS platformunun özerk davranışlarının gerçekleştirilebilmesi için tasarlanmış bir birimdir. AKB; Düşük Seviye Kontrol (DSK) ve Yüksek Seviye Kontrol (YSK) olmak üzere iki alt birimden oluşmaktadır.

AKB – DSK;

AKB – DSK birimi, LM3S9B96 ARM® Cortex™-M3 tabanlı mikro denetleyici kullanılarak motor kontrolü, sensör denetleme ve üst seviye birimlerle haberleşebilme gibi alt seviye görevleri yerine getirmek için geliştirilmiş ve çeşitli çevre birimlerine güç sağlamak amacı ile DA - DA buck dönüştürücüler ile tümleşik olarak tasarlanmış bir geliştirme kartıdır (Bkz. Şekil 3.5).



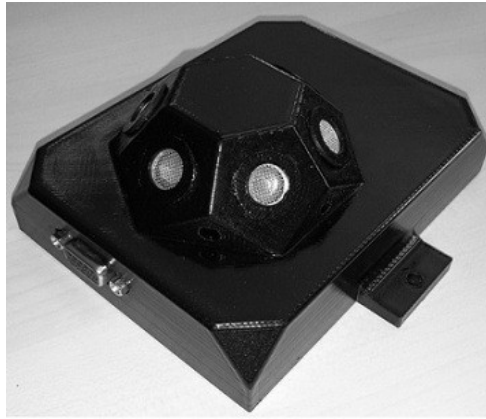
Şekil 3.5 AKB - DSK donanım prototipi

AKB – DSK birimi üzerinde çalışan gömülü yazılım, tüm girdi çıktı arayüzlerinden gelen algılayıcı bilgilerinin toplanması ve sandalye üzerindeki motorların sürülmesi işlemlerini gerçek zamanlı bir şekilde çalıştırmak üzere geliştirilmiştir. Bu zaman kısıtlarını göz önünde bulundurması açısından kritik bir yazılım mimarisine sahiptir. Yazılım, donanımsal ve yazılımsal zamanlayıcılar (timer) ve kesmeler (interrupt) kullanılarak zaman dilimli bir gerçek zamanlı işletim sistemi (RTOS) gibi geliştirilmiştir. Bu yapı, tanımlı işlerin tam istenen zamanda çalışmasını sağlamaktadır. Bu çalışma süresi içerisinde tüm giriş çıkış arayüzlerinden ilgili verileri toplayıp AKB – YSK’ya gönderebilmektedir. Şekil 3.6’da, geliştirilen zaman dilimli yazılımın bir saniye boyunca gerçekleştirdiği görevler ve bu görevlerin çalışma zamanları gösterilmektedir.

birbirleri ile sürekli olarak SMO15 bilgileri, açısal – doğrusal hız bilgileri, motor sürme verileri vb. alış verişi yapmaktadır. AKB – YSK akıllı kontrol anlamında, ATEKS platformunun özerk kontrolü için gerekli olan karar verme mekanizmaları ve kapıdan geçme, cisme yanaşma vb. davranışları gerçekleştirilmektedir.

3.2.2 Konum hesaplama birimi

ATEKS platformu, dış ortamlarda konum bilgisini GPS vb. sistemlerden alabilmektedir. İç ortamlarda ise, geliştirilmiş olan İç Ortam KONumlandırma Sistemini (İÇKON) (Yayan, 2012; Yücel vd., 2012) kullanarak konum bilgisini alabilmektedir. İÇKON sisteminden alınan sinyaller ile iç ortamda konum hesaplayan Gezgin Algılayıcı Birim (GAB) Şekil 3.8’de verilmektedir. GAB için ROS sürücü kütüphanesi geliştirilmiştir.



Şekil 3.8 Konum hesaplama birimi

İÇKON sistemi, sadece sesüstü sinyaller kullanmakta olup konumu cm hassasiyetinde hesaplamaktadır. Temel olarak sesüstü sinyallerin ortamda varış zaman farkı ölçümü ile mesafe farklarının hesaplanması ve hesaplanan mesafe farklarından hiperbol kesişimi yönteminin bir çözümü olan En Az Kareler algoritmasını kullanarak gezgin algılayıcı birimin konumu hesaplamaktadır (Yücel vd., 2012). ATEKS

platformunda ayrıca, GAB konum bilgisi ve kodlayıcılardan gelen artırimsal konum değişim bilgisi Kalman filtre ile tümleştirilerek daha hassas konum hesaplanmaktadır.

ATEKS platformundaki her iki tekere Şekil 3.9’da verilen kodlayıcı kuple edilmiştir. Kodlayıcılar, AKB – DSK birimine Quadrature Encoder Interface (QEI) üzerinden bağlanmaktadır. Kodlayıcıya ait teknik özellikler Çizelge 3.1’de verilmiştir.



Şekil 3.9 Kodlayıcı

Çizelge 3.1 Kodlayıcı teknik özellikleri

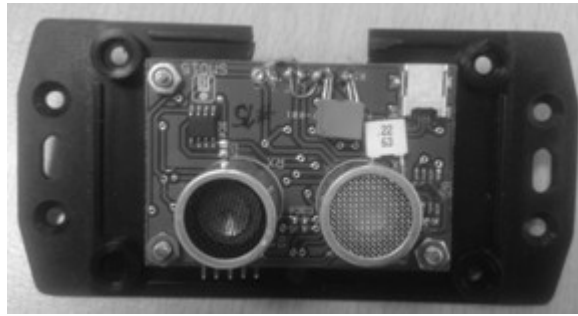
Çalışma Sıcaklığı	-25 ile 85 derece
Depolama Sıcaklığı	-40 ile 100 derece C
Koruma Sınıfı	IP54
Gövde	50 mm çap
Çalışma Frekansı	300 Khz
Çalışma Hızı	Optik: 6000 rpm maks. Manyetik: 3000 rpm maks
Delik Çapı	6 mm, 8 mm, 10 mm

3.2.3 Mesafe algılama birimleri

ATEKS platformu üzerinde etrafındaki cisimlere olan mesafeyi ölçmek üzere sesüstü mesafe algılayıcı SMO15 ve Microsoft Kinect algılayıcıları yerleştirilmiştir.

SMO15;

ATEKS platformunda tüm yönlerde olan cisimleri algılamak üzere ön tarafında dört adet, sağ sol ve arka taraflarında iki şer adet monte edilmiş toplam 10 adet SMO15 mesafe algılayıcısı bulunmaktadır. SMO15 mesafe ölçer (Sesüstü mesafe ölçer, 2014,) 2 – 500 cm arasında 1 cm hata payıyla mesafe ölçümü yapabilen ve USB/CANBUS protokolleri üzerinden hesaplanan mesafeyi gönderebilen bir algılayıcıdır (Bkz. Şekil 3.10). Bu algılayıcılar, sürüş destek modunda engellere çarpışmayı önleme, özerk modda ise geliştirilen yüksek seviye davranışların uygulanması sırasında ikinci seviye çarpışma önleme sistemi olarak kullanılmaktadır.



Şekil 3.10 Sesüstü mesafe ölçme birimi

Microsoft Kinect;

Microsoft tarafından Xbox 360 platformu için geliştirilen oyuncu hareketi algılama amaçlı piyasaya sürdüğü bir algılayıcıdır (Microsoft Kinect, 2014). Ucuz fiyatı ve sağlamış olduğu yazılımsal ara yüzler ile robotik üzerine çalışan araştırmacılar tarafından da kullanılmaya başlanmış bir derinlik ve renk algılama sensörüdür (Bkz. Şekil 3.11).



Şekil 3.11 Microsoft kinect

Kinect, sağlamış olduğu yazılım kütüphaneleri (Software Development Kit, SDK) ile Windows ve Ubuntu işletim sistemlerinde kullanılabilmektedir. ATEKS platformunda, Microsoft Kinect sensörü tarafından üretilen üç boyutlu derinlik bilgileri PCL (Point Cloud Library, 2014) kütüphanesi yardımı ile yüksek seviye davranışların ihtiyaç duyduğu ortam bilgisine dönüştürülmektedir.

3.2.4 Kullanıcı arayüzü birimi

Kullanıcı Arayüzü Birimi (KAB), ATEKS platformunun kullanıcı ile etkileşimini sağlamaktadır. KAB üzerinden ATEKS platformunun temel parametreleri değiştirilebildiği gibi, ortamda hareket ederken bulunduğu konum ve cisimlere olan mesafesi görselleştirilebilmektedir. Ayrıca KAB, ATEKS için önerilen yüksek seviye kontrol mimarisinde, planlama işlemlerinin gerçekleştirildiği birimdir. Kullanıcın bulunduğu noktadan herhangi bir hedef noktaya rota planlaması yapabilmekte ve bunları işletilmek üzere AKB – YSK'ya iletebilmektedir. Şekil 3.12'de KAB görüntüsü verilmekte olup, bu birim için de bir ROS sürücü kütüphanesi geliştirilmiştir.



Şekil 3.12 KAB için kullanılan donanım

KAB’da sürüş destek, navigasyon ve yönetici işlemleri için üç temel ekran bulunmaktadır.

Sürüş Destek Ekranı;

Sürüş destek ekranında, kullanıcıya ATEKS platformu üzerindeki SMO15 mesafe ölçerlerden gelen cisimlere olan mesafe görselleştirilmektedir (Bkz. Şekil 3.13). SMO15 algılayıcılarından ölçülen mesafe bilgilerine bağlı olarak oluşturulan görsellerin rengi ve uzunluğu değişmektedir. Ayrıca, herhangi bir algılayıcıdan bir engele çarpma riski oluşturacak kadar yaklaşmış durumda ise, görsel uyarının yanı sıra sesli uyarılar da devreye girerek kullanıcı uyarılmaktadır.



Şekil 3.13 KAB sürüş destek ekranı

Navigasyon Ekranı:

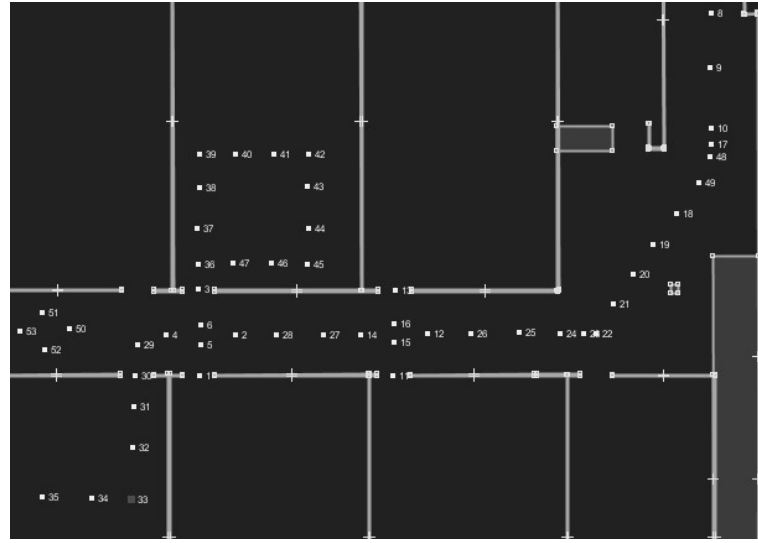
Navigasyon ekranında, aracın bulunduğu ortamın krokinin görselleştirilmesi ve bu kroki üzerinde rota planlaması yapılması gerçekleştirilmektedir (Bkz. Şekil 3.14). Ayrıca, ATEKS platformunun hareketi sırasında GAB üzerinden gelen konum bilgileri kullanıcıya kroki üzerinde sunulmaktadır.



Şekil 3.14 KAB navigasyon ekranı

Navigasyon ekranında görselleştirilen krokiler, Java Openstreet Map (JOSM) editörü kullanılarak belli standartlarda oluşturulmuştur. Burada iç ortama ait kapı,

merdiven, pencere, geiş noktası vb. i ortam zellikleri, dıř ortam navigasyon cihazlarındaki POI (Point of Interest) ler gibi krokiye eklenebilmektedir. KAB'ta ise, standart *.osm uzantılı oluřturulan bu dosyaları okuyan ve grselleřtiren yazılım ktphaneleri geliřtirilmiřtir. rneėin; řekil 3.15'te Eskiřehir Osmangazi niversitesi Teknopark Giriř Katı krokisi ve bazı geiş noktaları grselleřtirilmektedir. Geiş noktalarına rota planlamasında deėerlendirilmek zere etiketler eklenebilmektedir.



řekil 3.15 Teknopark krokisi

JOSM ortamında hazırlanan krokilerdeki geiş noktaları, hazırlanan yazılım ktphanesi ile rota planlamada kullanılmak zere dėm ve ayrıtlardan oluřan izge yapısına dnřtrlmektedir.

Daha nce bahsedildiėi gibi navigasyon ekranında, kullanıcı bulunduėu konumu kroki zerinden grebilmektedir. Bu noktada, gitmek istediėi hedef noktasına navigasyon ekranında kullanıcıya seim yaptırılıp, rota planlanmaktadır. Rota planlamada literatrde yaygın olarak kullanılan A* (Yazıcı vd., 2007) algoritması kullanılmıřtır. Planlanan rota, izge zerindeki dėmler dizisidir. Dėmlerin i ortamda karřılık geldiėi geiş noktaları ve bunların etiket deėerleri kullanılarak grev

planı oluşturulur. Oluşturulan görev planına ait bir düğümün alabileceği özellikler Çizelge 3.2’de gösterilmektedir.

Çizelge 3.2 Düğümlerin alabileceği değerler

Düğüm İsmi	Düğüm Alabileceği Değerler
Özellik	Boş Alan(BA), Boş Alandan Kapı Girişi(BKG), Kapı Ortası(KO), Koridordan Kapı Girişi(KKG), Dörtlü Kavşak Girişi(DKG), Üçlü Kavşak Girişi(TKG), Koridor(KRDR)
Komşular	Komşu durumundaki düğümlerin numaraları birbirinden ‘_’ ile ayrılarak bu kısma yazılmıştır.
AyrıtDavranış	KoridorTakibi, RotaTakibi, KoseyiDonme, CismeYanas, HedefeGit

Görev planı içerisinde takip edilmesi gereken düğümler ve ardışık düğümler arasında ATEKS’ in gerçekleştireceği davranışlar bulunmaktadır. XML (Extendedn Mark-up Language) formatında oluşturulan örnek bir görev planının bir kısmı aşağıda gösterilmektedir.

<Ayrıt0> --0 numaralı ayrıt

<BasN> --Ayrıtın başlangıç düğümü

<BasX>-10000</BasX> --Bu düğümünün X koordinatı

<BasY>-6000</BasY> --Bu düğümünün Y koordinatı

<Oz>OI</Oz> --Bu düğümün özelliği

<BS>1</BS> --Diğer düğümlere olan bağlantı sayısı

</BasN>

<BitN> --Ayrıtın bitiş düğümü

<BitX>-9200</BitX> --Bu düğümün X koordinatı

<BitY>-7000</BitY> --Bu düğümün Y koordinatı

<Oz>OI</Oz> --Bu düğümün özelliği

<BS>2</BS> --Diğer düğümlerle olan bağlantı sayısı

</BitN>

<Dav>HedefeGit</Dav>--Düğümler arasında ATEKS’in uygulayacağı davranış

```

<Hiz> --Düğümler arasın min, max ve ref hızları
  <Min>0.3</Min>
  <Max>0.7</Max>
  <Ref>0.5</Ref>
</Hiz>
</Ayrit0>

```

XML formatında hazırlanan görev planı, AKB – YSK birimine gönderilmektedir. Burada ayrıştırılan görev planı işletilerek davranışlar çalıştırılmaktadır.

Yönetici Ekranı:

Yönetici ekranında, ATEKS platformuna ait dinamik, kinematik, algılayıcı yerleşim vb. sistem parametreleri tanımlanabilmektedir. Ayrıca, diğer ekranlardaki sesli uyarı, engel uyarı mesafesi, algılayıcı tazeleme hızı vb. ayarlar da yapılabilmektedir. Şekil 3.16’da yönetici ekranı gösterilmektedir.



Şekil 3.16 KAB yönetici ekranı

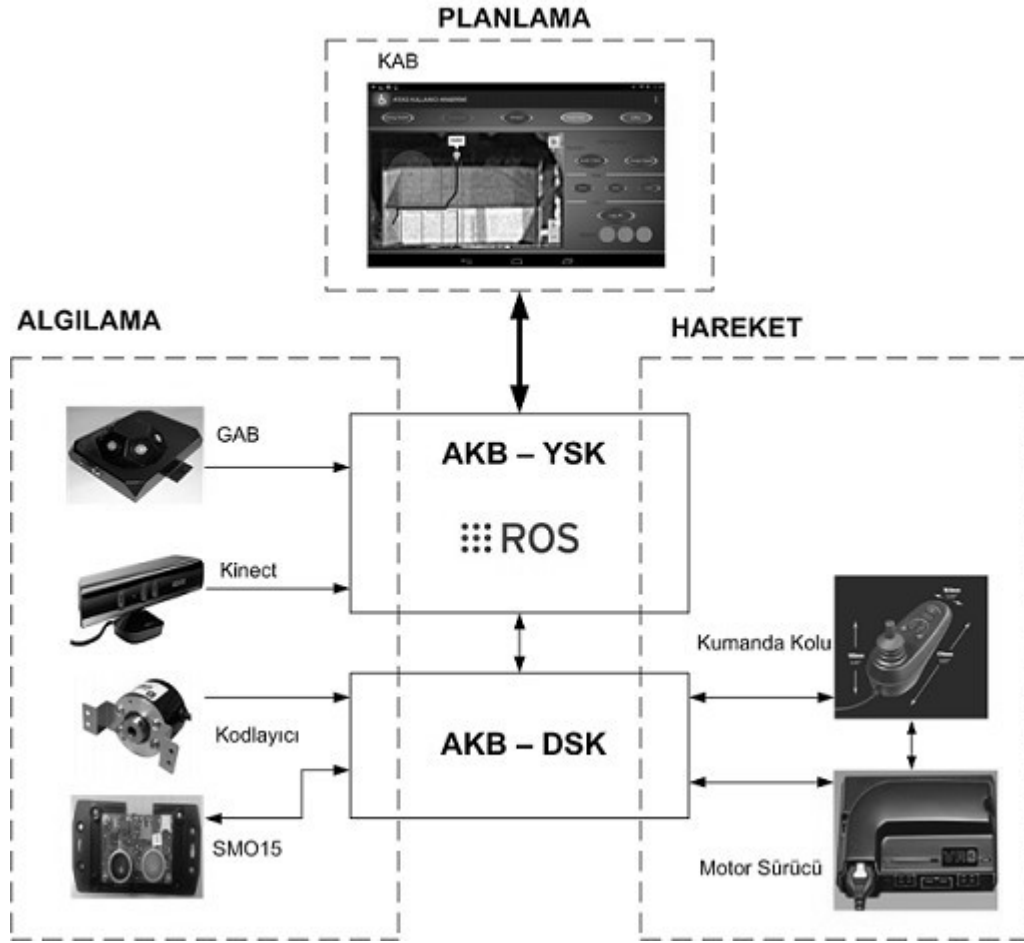
BÖLÜM 4

ÖNERİLEN YÜKSEK SEVİYE KONTROL SİSTEMİ

ATEKS platformundaki özerk kontrolü gerçeklemek üzere Akıllı Kontrol Birimi (AKB) tasarlanmıştır. AKB yüksek seviye kontrol (YSK) ve düşük seviye kontrol (DSK) olmak üzere iki alt bileşene sahiptir. AKB – YSK’da ATEKS platformunun verilen görevleri özerk olarak yapabilmesi için gerekli davranışlar gerçekleştirilmektedir. AKB – YSK’daki davranışların ürettiği doğrusal ve açısal hız değerleri, AKB – DSK’da (Akar vd., 2013) sisteme uygulanmaktadır. AKB – YSK birimi, takip eden alt bölümde verildiği gibi ROS robotik ara katmanı üzerinde çok ajanlı bir yapıda tasarlanmış ve gerçekleştirilmiştir.

4.1 ATEKS Kontrol Mimarisi

ATEKS platformu için önerilen kontrol mimarisi melez bir yapıya sahiptir. Bu yapı, Şekil 4.1’de gösterilen planlama, algılama ve hareket olmak üzere temel üç bileşene sahiptir. ATEKS platformunda yüksek seviye kontrol, AKB – YSK ve KAB birimlerinde gerçekleştirilmektedir. KAB, kullanıcı arayüzü özelliği yanı sıra, tarif edilen görevin gerçekleştirilmesi için gerekli rota planlamalarını yapmaktadır. Yapılan bu planlar, AKB – YSK’da davranışlar ile gerçekleştirilmektedir.



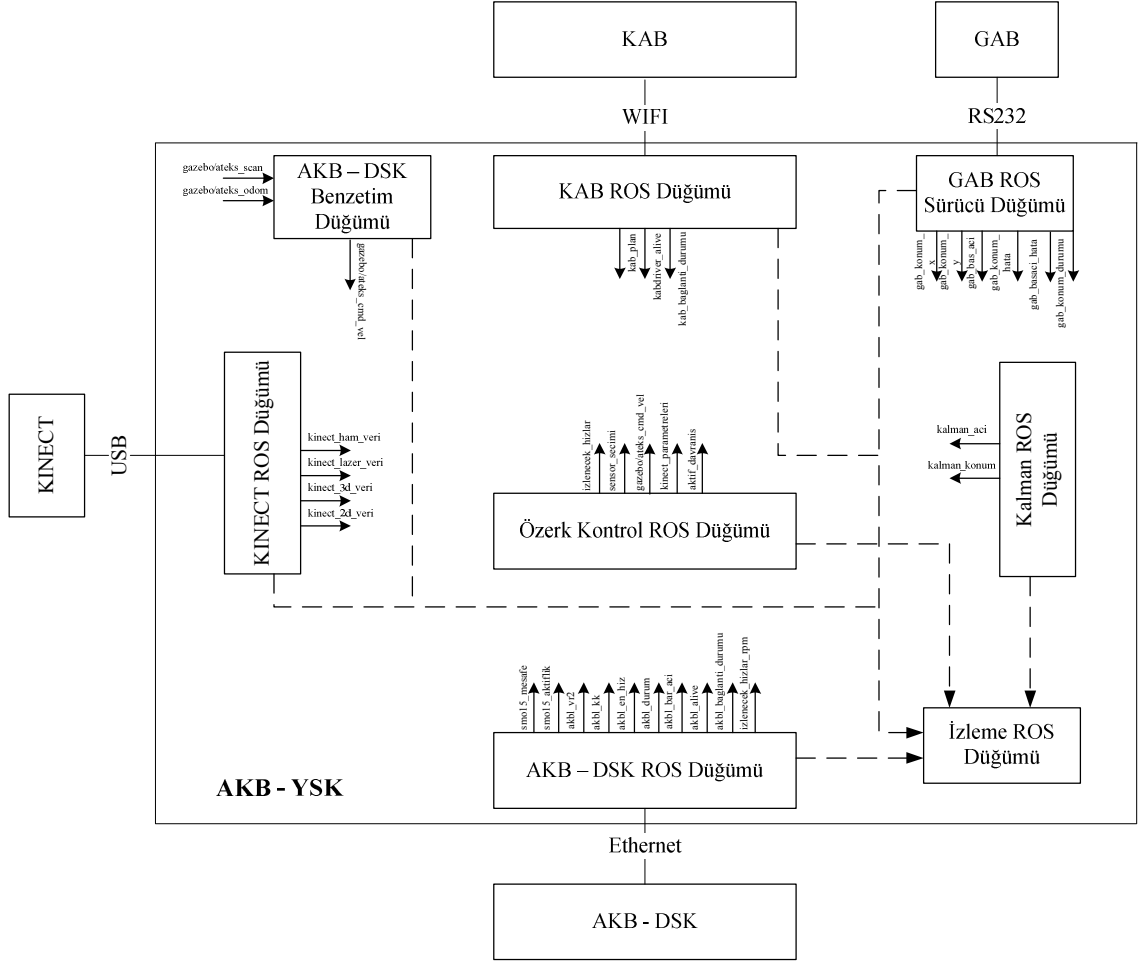
Şekil 4.1 Yüksek seviye kontrol mimarisi

AKB – YSK birimindeki davranışlar ROS ara katmanı üzerinde çalışmaktadır. Özerk mod için geliştirilen davranışların başarıyla gerçekleştirilebilmesi için gerekli algılama, hareket ve koordinasyonu sağlamak üzere ROS ara katmanında çalışan düğümler tasarlanmıştır. Önerilen yüksek seviye kontrol sistemi birçok düğümden oluşmaktadır. Takip eden alt bölümde önerilen kontrol mimarisini oluşturan düğümler verilmektedir.

4.2 AKB – YSK ROS Düğümleri

AKB – YSK birimi, yüksek seviye kontrol ile ilgili planlama dışında gerekli işlemleri gerçekleştiren bir yapıdır. Bu yapıda, farklı görevlere sahip ajanlar (erkin)

Şekil 4.2’de verilen ROS düğümleri olarak gerçekleştirilmiştir. Bu şekilde, ROS düğümlerinin yanı sıra, bu düğümlerin sunduğu topik bilgileri verilmektedir.

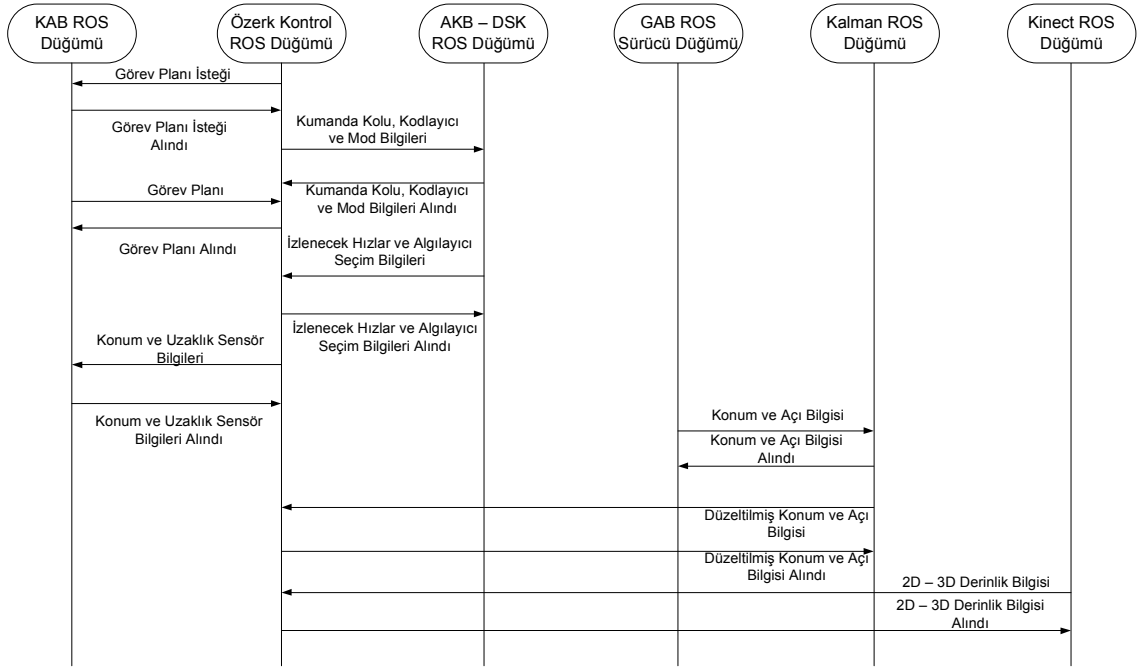


Şekil 4.2 AKB - YSK ROS düğümleri

KAB ROS Düğümü, KAB’tan gelen bilgilerin Özerk Kontrol ROS Düğümü’ne aktarılması için geliştirilmiş planlama arayüz düğümüdür. AKB – DSK ROS Düğümü, AKB – DSK birimi ile iletişim için geliştirilmiş bir arayüz düğümüdür. GAB ROS Sürücü Düğümü, GAB konumlama birimi ile Kalman ROS Düğümü ve Özerk Kontrol ROS Düğümü arası iletişimi sağlayan arayüzdür. Özerk Kontrol ROS Düğümü davranışların gerçekleştirildiği merkezi kontrol düğümüdür. Kalman ROS Düğümü,

GAB ROS Sürücüsü Düzümünden çıkan ham konum ve açı bilgilerinin işlenerek daha düzgün konum ve açı bilgileri haline getirildiği düğümdür.

Şekil 4.2’de verilen ROS düğümleri arasındaki veri alış verişini gösteren dizi (sequence) diyagramı Şekil 4.3’te verilmektedir. Bu diyagramda, KAB, Özerk Kontrol, AKB – DSK, GAB Sürücü, Kalman ve Kinect ROS düğümlerindeki ajanlar arası paylaşılan bilgiler, zaman sırası açısından gösterilmektedir.



Şekil 4.3 ROS düğümleri dizi diyagramı

Bu diyagramda Özerk Kontrol ROS Düzümü, KAB ROS düğümleri, AKB – DSK ROS Düzümü, GAB ROS Düzümü, Kalman ROS Düzümü, Kinect ROS Düzümü bulunmaktadır. Özerk Kontrol ve KAB ROS düğümleri arasında görev planı ve algılayıcı bilgileri paylaşılır. Özerk Kontrol ve AKB – DSK ROS Düzümleri arasında kumanda kolu - motor sürücü ve izlenecek hızlar bilgileri paylaşılır. Aynı zamanda, Kalman ROS Düzümünden düzeltilmiş konum ve açı, Kinect ROS Düzümünden 2 ve 3 boyutlu derinlik bilgileri, Özerk Kontrol ROS Düzümüne aktarılır. Takip eden alt bölümde bu düğümlere ait bilgiler aktarılır.

4.2.1 AKB – DSK ROS düğümü

AKB – DSK ROS Düğümü, AKB – DSK birimi üzerinden gelen algılayıcı ve motor bilgilerini alarak Özerk Kontrol ROS Düğümüne gönderen, AKB – YSK'daki davranışların çıktısı olan doğrusal ve açısal hız değerlerini de AKB – DSK'ya gönderen hareket arayüz düğümüdür. 10 Hz frekansta AKB – DSK ile haberleşmektedir. AKB – DSK ile bağlantı kopması halinde tekrar bağlantı açma isteği gönderebilmektedir. Çizelge 4.1'de AKB – DSK ROS Düğümü topikleri ve açıklamaları gösterilmektedir.

Çizelge 4.1 AKB - DSK düğümü topikleri

Topik Adı	Açıklaması
smo15_mesafe	SMO15lerin mesafelerini basmaktadır.
smo15_aktiflik	SMO15lerin aktiflik durumlarını basmaktadır.
gab_konum	GAB konum bilgisini basmaktadır.
gab_kalman	GAB kalman filtre için ilgili bilgileri basmaktadır.
akbl_vr2	Motor sürücüsünün bilgilerini basmaktadır.
akbl_kk	Kumanda koluna ait bilgileri basmaktadır.
akbl_enk_konum	Kodlayıcının hesaplandığı bağıl konum bilgisini basmaktadır.
akbl_enk_yon	Kodlayıcının hesaplandığı yön bilgisini basmaktadır.
akbl_enk_hiz	Kodlayıcının hesaplandığı hız bilgisini basmaktadır.
akbl_durum	AKB – DSK durum makinesine bağlı olarak bulunulan durum bilgisini basmaktadır.
akbl_bar_aci	BAR da hesaplanan baş açısı referans değerinin basmaktadır.
akbl_alive	ROS üzerinde çalışan düğümün ayakta olduğunu göstermektedir.
akbl_baglanti_durumu	AKB – DSK bağlantı durumu bilgisinin göndermektedir.

4.2.2 KAB ROS düğümü

KAB ROS Düğümü, KAB üzerinden gelen görev planı ve yönetici ayarları bilgilerini alarak Özerk Kontrol ROS Düğümüne gönderen, ATEKS platformu üzerindeki algılayıcı ve sistem parametrelerini KAB'a gönderen kullanıcı arayüz düğümüdür. Çizelge 4.2'de KAB ROS Düğümü topikleri ve açıklamaları gösterilmektedir.

Çizelge 4.2 KAB ROS düğümü topikleri

Topik Adı	Açıklaması
kab_plan	KAB 'ta hesaplanan görev planına ait bilgileri (düğümler, referans hızlar, davranışlar vb.) içeren topiktir.
kabdriver_alive	ROS üzerinde çalışan Node 'un ayakta olduğunu gösteren topiktir.
kab_baglanti_durumu	KAB bağlantı durumu bilgisinin gönderildiği topiktir.

4.2.3 Kinect ROS düğümü

Kinect ROS Düğümü, Kinect üzerinden gelen ilgili derinlik bilgisi içeren veri katarını ayrıştıran ve Kinect algılayıcısını kontrol eden mesafe algılama arayüz düğümüdür. Çizelge 4.3'te Kinect ROS Düğümü topikleri ve açıklamaları gösterilmektedir.

Çizelge 4.3 Kinect düğümü topikleri

Topik Adı	Açıklaması
kinect_ham_veri	Tüm bilgilerin tutulduğu topiktir. (Point Cloud Verisi)
kinect_lazer_veri	Açı ve derinlik bilgilerinin tutulduğu topiktir.
kinect_2d_veri	Engellerin konum bilgisini 2 boyutlu tutan topiktir.
kinect_3d_veri	Engellerin konum bilgisini 3 boyutlu tutan topiktir.

4.2.4 GAB ROS sürücü düğümü

GAB ROS Sürücü Düğümü, İÇKON sisteminin mobil cihaz üzerindeki GAB alt birimi için AKB – YSK ile arayüzü sağlayan sürücü düğümüdür. Bu düğüm üzerinde GAB’dan gerekli konum, açı vb. bilgiler alınmaktadır. Çizelge 4.4’te GAB ROS Sürücü Düğümü topikleri ve açıklamaları gösterilmektedir.

Çizelge 4.4 GAB düğümü topikleri

Topik Adı	Açıklaması
gab_konum_x	Sinyal verileri kullanılarak hesaplanan ham konum x koordinat bilgisidir.
gab_konum_y	Sinyal verileri kullanılarak hesaplanan ham konumy koordinat bilgisidir.
gab_konum_hata	Hesaplanan konumun hata bilgisidir.
gab_bas_acisi	Sinyal verileri kullanılarak hesaplanan açı bilgisidir.
gab_bas_acisi_hata	Hesaplanan açının hata bilgisidir.

4.2.5 Kalman ROS düğümü

Kalman ROS Düğümü, GAB’dan gelen konum bilgisi ve kodlayıcıdan gelen arttırımsal konum bilgilerinin alınarak Kalman filtreden (Yücel vd., 2012) geçirilip konum bilgisinin iyileştirmesinin yapıldığı düğümdür. Çizelge 4.5’te Kalman ROS Düğümü topikleri ve açıklamaları gösterilmektedir.

Çizelge 4.5 Kalman düğümü topikleri

Topik Adı	Açıklaması
kalman_konum	Düzeltilen GAB konum bilgisinin yayınlandığı topiktir.
kalman_aci	Düzeltilen GAB açı bilgisinin yayınlandığı topiktir.

4.2.6 İzleme ROS düğümü

Bu düğüm gerektiğinde kullanıcının verilen görevi, görevin mevcut durumunu vb. detayların izlenebildiği düğümdür. Algılayıcılardan gelen veriler de bu düğüm üzerinden kullanıcıya gösterilmektedir. Ayrıca bu düğüm, ROS düğümlerinin canlılık topiklerinin, ARIZA durumunun, sonlu durum makinesine ait bilgilerin, gerçekleşen akıllı davranışların, tüm algılayıcılara ait bilgilerin izlenebildiği düğümdür.

4.2.7 AKB – DSK benzetim ROS düğümü

AKB – DSK ROS Düğümü, HIL (Hardware In Loop) senaryolarını gerçekleyebilmek için tasarlanmış olan düğümdür. Görevi, AKB – DSK'daki düşük seviye kontrolü benzetim ortamında gerçekleştirmektir. Bu durumu sağlayabilmek için AKB – DSK ROS Düğümü yazılımı ile AKB – DSK yazılımının arasındaki haberleşmeyi taklit etmektedir. Bu düğüm, AKB – YSK'dan gelen davranış çıktısı olan doğrusal ve açısal hız değerlerini gerçek platform yerine, GAZEBO benzetim ortamındaki ATEKS platformu modeline uygulamaktadır.

4.2.8 Özerk kontrol ROS düğümü

Yukarıda verilen düğümler, ATEKS platformunun özerk kontrolü için gerekli tüm algılama, planlama ve hareket fonksiyonlarını sağlamaktadırlar. Özerk Kontrol ROS Düğümünde ise, platformun özerk kontrolü için gerekli karar mekanizmaları ve davranışlar gerçekleştirilmektedir. Çizelge 4.6'da Özerk Kontrol ROS Düğümü topikleri ve açıklamaları gösterilmektedir.

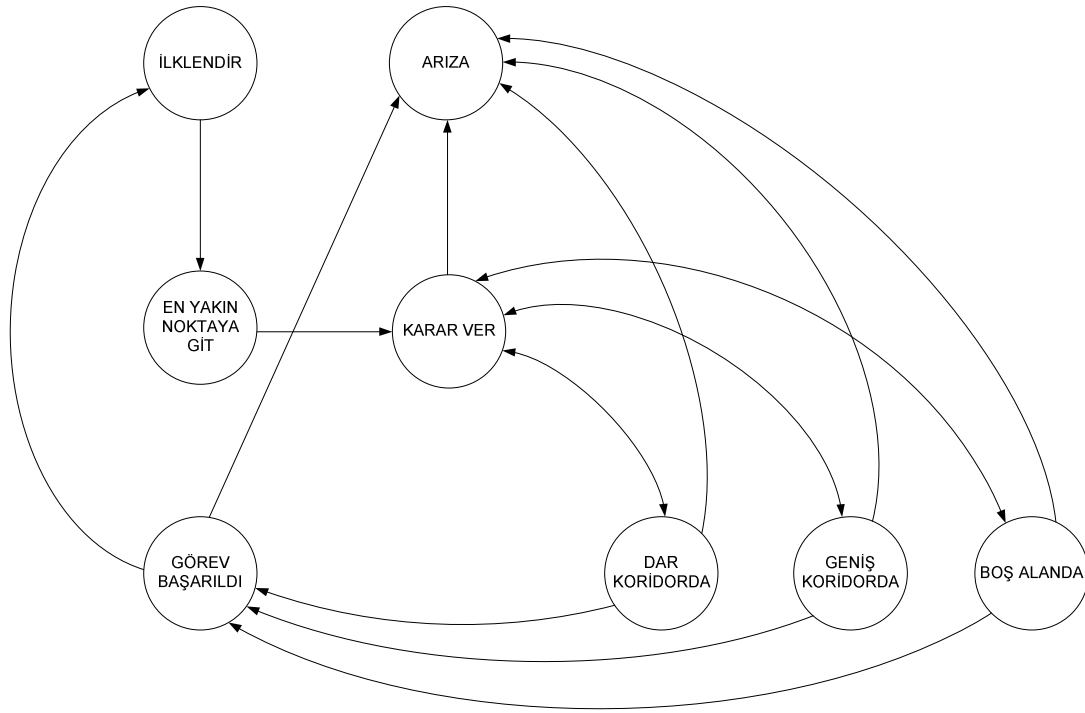
Çizelge 4.6 AKB - YSK düğümü topikleri

Topik Adı	Açıklaması
izlenecek_hizlar	AKBH yüksek seviye kontrol programı tarafından hesaplanan açısal ve doğrusal hız bilgilerini içeren topiktir.
sensor_secimi	Kullanılmak istenen sensörlere ait istek bilgilerini içeren topiktir.
akbl_den_par	AKBL a ait denetleyici parametresi bilgilerini içeren topiktir.
akbh_robot_bilgileri_durumu	İlgili dosyadan robot bilgilerinin durumu hakkında bilgi veren topiktir.
akbh_osilasyon_durumu	Otonom mod esnasında herhangi bir tıkanıklık veya osilasyon durumu ile ilgili bilgi veren topiktir.
akbh_yol_durumu	Otonom mod esnasında yol durumu hakkında bilgi veren topiktir.

Özerk Kontrol ROS Düğümü, öncelikle KAB'tan gelen görev planını almaktadır. Görev planında ATEKS' in yapması gereken davranışlar ve izleyeceği yollar bulunmaktadır. Bu görev planını gerçekleyecek şekilde Özerk Kontrol ROS Düğümü içerisindeki Sonlu Durum Makinesi (SDM) gerekli davranışları tetiklemektedir. Herhangi bir davranış tetiklendikten sonra, düğüm gerekli algılayıcı bilgilerini alıp değerlendirmek ve doğrusal ve açısal hız çıktı değerini üretmektedir. ATEKS platformunda özerk kontrolün yanı sıra, hata tespiti ve gideriminin (arıza girdileri) aynı anda kontrol edilmesi gerekmektedir. Tüm bu özellikleri gerçeklemek üzere, Özerk Kontrol ROS Düğümü içerisindeki SDM görev almaktadır.

Özerk kontrol ROS düğümü SDM yapısı;

Özerk Kontrol ROS Düğümü içindeki SDM, $M = (Q, \Sigma, \delta, q_1)$ ile gösterilsin. Burada Q, Σ, δ sırası ile SDM durumları, girdileri (alfabesi) ve durum geçiş fonksiyonunu göstermektedir. q_1 ise başlangıç durumu olup durum geçiş çizgesi Şekil 4.4'te verilmiştir.



Şekil 4.4 Özerk mod SDM durum geçiş çizgesi

Özerk Kontrol ROS Düğümündeki SDM alfabe kümesi
 $\Sigma = \{AI_1, AI_2, AI_3, AI_4, AI_5, AI_6, AI_7, AI_8, AI_9, AI_10, UI_1, UI_2, UI_3, PI_1, PI_2, PI_3, \}$

Arıza Girdileri, Plan Girdileri ve Uyarı Girdileri olmak üzere üç temel bileşen olup Çizelge 4.7’de verilmektedir.

Çizelge 4.7 Özerk kontrol ROS düğümü SDM girdileri

Girdi Grubu	Girdi	Açıklaması
PLAN	P1	Düğüm Özelliği hakkında bilgi veren girdidir.
	P2	Düğüm komşu sayısı hakkında bilgi veren girdidir.
	P3	Ayrıta ait davranış hakkında bilgi veren girdidir.
ARIZA	A1	AKB – DSK ve AKB – YSK arası bağlantının kopukluk durumunu gösteren girdidir.
	A2	AKB – YSK ile KAB arası bağlantının kopukluk durumunu gösteren girdidir.
	A3	AKB – DSK biriminin sistem kapalı durumunu gösteren girdidir.
	A4	Yol durumu hakkında bilgi veren girdidir.
	A5	AKB – DSK birimin arıza durumunu gösteren girdidir.
	A6	AKB – DSK birimin sürüş destek durumunu gösteren girdidir.
	A7	AKB – DSK birimin kullanıcı modu durumunu gösteren girdidir.
	A8	ATEKS platformunun osilasyon veya takılma durumunu gösteren girdidir.
	A9	Görev planını durumu hakkında bilgi veren girdidir.
	A10	ATEKS platformuna ait parametreler hakkında bilgi veren girdidir.
UYARI	U1	ATEKS batarya seviyesi hakkında bilgi veren girdidir.

Özerk Kontrol ROS Düğümündeki SDM durum kümesi;

$Q = \{ILKENDIR, ENYAKINNOKTA, KARARVER, BOSLAN, GENISKORIDOR, DARKORIDOR, GOREVBASARILDI, ARIZA\}$

olup SDM durumları ATEKS platformunun bulunduğu iç ortama göre özel bir davranış ihtiyacına göre belirlenmiştir. Örneğin; boş alanda gerçekleşen bir hedefe gitme davranışı ile dar koridarda gerçekleşen bir hedefe gitme davranışı farklı parametrelere sahip olmaktadır. SDM durumlarının detaylı açıklaması aşağıdadır:

İlklendir

İlklendir durumunda, özerk kontrolün başarı ile gerçekleştirilebilmesi için gerekli ROS düğümlerinin çalışırılığı, ihtiyaç olan algılayıcıların canlılığı, ATEKS platformu kontrolü için gerekli sistem parametrelerinin varlığı vb. bilgiler kontrol edilir. Herhangi bir hata durumunda ARIZA durumuna geçilmektedir. Eğer bir hata yoksa KAB ROS Düğümü üzerinden gelen plan girdisi beklenir. Bu girdi geldiğinde en yakın noktaya git durumuna geçilmektedir.

En yakın noktada

KAB'tan gelen görev planı göz önünde bulundurularak ATEKS'in bulunduğu yerden en yakın plan noktasına geçtiği durumdur. Bu durum esnasında Kalman ROS Düğümü üzerinden gelen konum bilgisi, ilgili noktaya ulaşıp ulaşılmadığını anlamak için kullanılmaktadır. Navigasyon işlemi esnasında hedefe git davranışı çalışmaktadır. En yakın noktaya ulaşıldıktan sonra "KARAR VER" durumuna geçilmektedir.

Karar ver

Herhangi bir SDM durumunda tanımlanan alt görev bittiğinde bu duruma geçiş yapılmaktadır. Görev planı dikkate alınarak daha sonra geçilecek duruma karar verilmektedir. Tüm ROS düğümlerinden gelen canlılık topikleri (*kabdriver_alive*, *kinectdriver_alive*, *abkldriver_alive*, *kalman_alive*) burada kontrol edilmekte, herhangi bir hata durumunda ARIZA durumuna geçilmektedir. Eğer hedef noktaya ulaşıldı ise GÖREV BAŞARILDI durumuna geçilmektedir.

Boş alanda

Koridor haricindeki oda, park alanı vb. yerlerdeki Hedefe Git, Cisme Yanaşma veya Kapıdan Geçme gibi davranışlar bu kapsamda değerlendirilmektedir. Bu davranışların detayları takip eden alt bölümde verilmektedir. Herhangi bir davranış bitirildiğinde tekrar KARAR VER durumuna geçilmektedir. Bu durumda ARIZA ile ilgili girdiler kontrol edilmekte, gerektiğinde ARIZA durumuna geçilmektedir.

Dar&geniş koridorda

Kapalı bir iç ortamdaki geçiş yollarında hareket sırasında bu duruma geçilmektedir. Koridor genişliği 2m üzerinde ise iki araç karşılıklı güvenli bir şekilde geçebilmekte olup buna göre hedefe git vb. davranışlar aktiflenmektedir. Eğer bu değerin altında ise, koridor ortalanarak gidilmektedir. Herhangi bir davranış bitirildiğinde tekrar KARAR VER durumuna geçilmektedir. Bu durumda, ARIZA ile ilgili girdiler kontrol edilmekte, gerektiğinde ARIZA durumuna geçilmektedir.

Görev başarıldı

Bir görev başarılı bir şekilde bitirildi ve hedef bölgeye ulaşıldı ise bu duruma geçilmektedir. Bu durumda kullanıcıya ilgili görevin bitirildiğini gösteren uyarılar verilmektedir. Daha sonra yeni görev planının gelmesi için İLKLENDİR durumuna geçilmektedir. Herhangi bir hata durumunda ARIZA durumuna geçilmektedir.

Arıza

Herhangi bir arıza durumu ile karşılaşıldığında geçilen durumdur. AKB – DSK a gönderilen “izlenecek hızlar topiğinde” açısız ve doğrusal hızlara “0” a atanmakta ve böylece ATEKS in durması sağlanmaktadır. Eğer, hata giderildi ise ATEKS kaldığı yerden görevini yapmaya devam etmektedir.

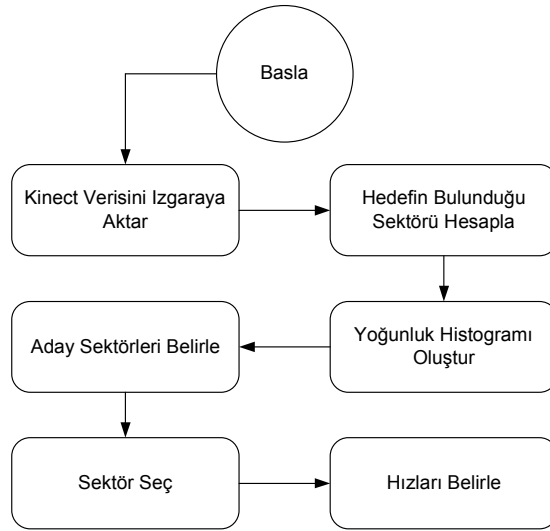
4.3 Özerk Kontrol Düzümü İçin Geliştirilen Davranışlar

Davranışlar gezgin robot sistemlerinin en önemli yapı taşıdır. Davranış, algılamadan alınan verilerin değerlendirilerek hedeflenen amaca uygun harekete dönüştürülmesini sağlar. Bu tez çalışması kapsamında, Özerk mod için geliştirilen hedefe gitme, koridor takibi, cisme yanaşma, kapıdan geçme davranışları duruma bağlı olarak farklı parametreler ile çalışmaktadır.

Davranışların geliştirilmesi kapsamında çeşitli yöntemler üzerinde inceleme ve çalışmalar yapılmıştır. Literatürde bulunan, Bölüm 2.3'te aktarılan potansiyel alanlar yaklaşımı, bug algoritmaları (Charifa, et. al., 2009), elastic band yöntemi (Quinlan, et. al., 1993), pure pursuit algoritması (Coulter, et. al., 1992), nearness diyagram (Minguez, et. al., 2004) algoritmaları incelenmiştir. Yapılan incelemeler sonucu ilk etapta davranışlar potansiyel alanlar ile kodlanmıştır. Gerçek ortam testleri esnasında yapılan kapsamlı çalışmalar neticesinde potansiyel alanlar metodu ile geliştirilen davranışlar için istenen sonuçların elde edilmediği görülmüştür. Daha sonra vektör alan histogram metodu (Borenstein, et. al., 1991) üzerinde çalışmalar yapılmıştır. Takip eden kısımlarda öncelikle vektör alan histogramı yönteminin detayları verilmekte, devamında ise farklı davranışlarda nasıl uygulandığı aktarılmaktadır.

Vektör alan histogramı (Vector Field Histogram, VFH) yöntemi

Bu tez çalışması kapsamında, hedefe gitme, kapıdan geçme ve koridor takibi davranışlarında kullanılmak üzere ortak bir vektör alan histogramı yöntemi kurgulanmıştır. Fakat, davranışa bağlı olarak bu yöntemde parametrik değişimler yapılmıştır. Kullanılan vektör alan histogramı algoritmasının herhangi bir adımda işlettiği akış diyagramı Şekil 4.5'te verilmiştir.



Şekil 4.5 VFH algoritması akış diyagramı

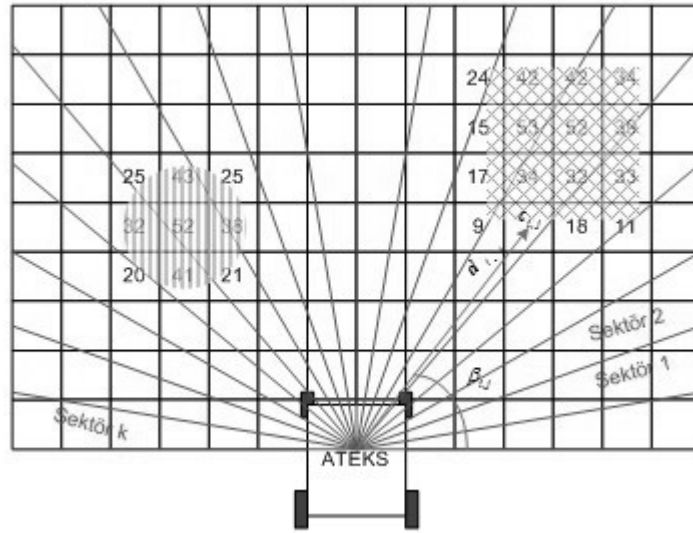
VFH algoritmasının adımlarının detayı aşağıda verilmektedir.

i. Kinect verisini ızgaraya aktarma

Kinect ROS Düşümünden gelen nesnelere ait koordinat bilgileri, Şekil 4.6'da ATEKS platformunun geometrik ortasından başlayan ızgaraya dönüştürülmektedir. Izgara, ATEKS ile birlikte hareket eden aktif bir penceredir. ATEKS platformunda iki adet Kinect kullanıldığı için derinlik verisini ızgaraya aktarmak için önce her iki algılayıcıdan gelen veri, koordinat merkezi ATEKS platformunun geometrik ortası olacak şekilde dönüşümlere tabi tutulmaktadır. Izgaradaki her bir hücrenin kapsadığı alan içerisine düşen nesne koordinat sayısına bağlı olarak ilgili hücre değeri $c_{i,j}$ yazılmaktadır. Örneğin; Şekil 4.6'da sol üst köşede bulunan hücrenin değerinin 25 olması, Kinect algılayıcısının gördüğü engel için o koordinatlar içerisinde 25 farklı nokta veri ürettiği anlamına gelmektedir.

Algoritmada, ATEKS'in nokta olarak hesaba katılması, engelden sakınma davranışı için bazı problemlere sebep olmaktadır. Bu sorunu çözmek için, Hedefe Gitme ve Koridor Takibi davranışlarına nesne büyültme (object growing) işlemi uygulanmaktadır. Bu işlem kapsamında Kinect'ten gelen veri ızgaraya doldurulurken, davranışlara göre değişiklik gösteren nesne büyültme miktarına bakılarak, nesnenin

algılandığı hücrenin etrafındaki hücrelere de engel algılanmış gibi bilgi doldurulmaktadır.



Şekil 4.6 VFH ızgara ve sektör örneği

VFH algoritmasında ızgara için ayrıca bir sektör tanımlaması yapılmaktadır. Sektör, merkezi ATEKS platformunun geometrik ortası olan k adet α derecelik açılarla bölünmüş ışınlar karşılık gelmektedir. Şekil 4.6'da sektörler ızgara üzerinde gösterilmektedir.

ii. Hedefin bulunduğu sektörün belirlenmesi

Hedefin ızgara ve sektördeki konumunun bilinmesi davranış için büyük önem arz etmektedir. Izgara, robota göre yerel konumlarda çalıştığı için küresel konumlarda olan hedef konumunun ızgaradaki yerini bulmak için küresel koordinatlardaki hedef konum denklem 4.1 kullanılarak yerel konuma dönüştürülür.

$$\begin{bmatrix} x_{h,y} \\ y_{h,y} \end{bmatrix} = \begin{bmatrix} \sin \varphi & -\cos \varphi \\ \cos \varphi & \sin \varphi \end{bmatrix} \begin{bmatrix} x_{h,g} - x_{r,g} \\ y_{h,g} - y_{r,g} \end{bmatrix} \quad (4.1)$$

Burada $(x_{h,y}$ ve $y_{h,y}$) hedefin yerel konum bilgilerini, $(x_{hg}$ ve y_{hg}) hedefin global konumunu, $(x_{r,g}$ ve $y_{r,g}$) ATEKS platformunun konumunu, φ ise ATEKS X eksenini ile yaptığı açıyı göstermektedir.

Yerel konuma dönüştürülen hedef konumun, ızgara içerisinde karşılık geldiği hücre bulunur. Daha sonra, hedef konumun denk geldiği sektör denklem 4.2 ile hesaplanmaktadır.

$$k_h = INT\left(\frac{\beta_{i,j}}{\alpha}\right) \quad (4.2)$$

Bu denklemde; $\beta_{i,j}$ hücrenin ATEKS'in geometrik merkezine olan açısı, α sektörler arası açıyı ve k_h hedefe karşılık gelen sektör numarasını göstermektedir.

iii. Sektör bazında yoğunluk histogramının oluşturulması

Kinect algılayıcısından gelen ortamdaki nesnelere ait veriler, ızgaradaki hücrelere aktarıldıktan sonra bu hücrelerin karşılık geldiği sektörler üzerindeki yoğunluk değerleri hesaplanmaktadır. Bunun için denklem 4.3 ve 4.4 kullanılarak sırasıyla ızgaradaki her bir hücrenin ATEKS'in geometrik merkezine olan uzaklığı ($d_{i,j}$) ve açısı hesaplanır.

$$d_{i,j} = \sqrt{(x_{i,j} - x_0)^2 + (y_{i,j} - y_0)^2} \quad (4.3)$$

$$\beta_{i,j} = \tan^{-1} \frac{y_{i,j} - y_0}{x_{i,j} - x_0} \quad (4.4)$$

Devamında her bir hücrenin yoğunluk değeri ($m_{i,j}$), denklem 4.5 ile hesaplanmaktadır.

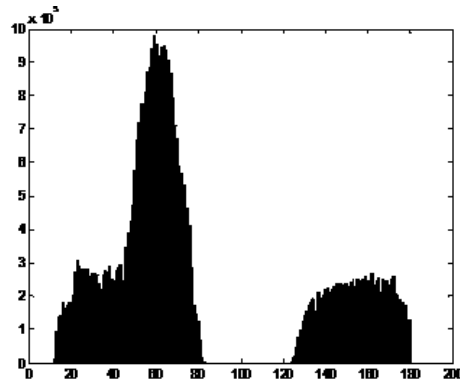
$$m_{i,j} = (c_{i,j})^2 (a - b d_{i,j}) \quad (4.5)$$

Burada $c_{i,j}$ hücre değerini, a ve b katsayıları yoğunluk hesabı yapılacak maksimum hücre uzaklık eşitliğini pozitif olacak şekilde seçilen katsayılardır.

Daha sonra Denklem 4.6 ile sektörlere karşılık gelen tüm hücreler dikkate alınarak herbir sektör için yoğunluk değeri h_k hesaplanır.

$$h_k = \sum_{i,j} m_{i,j} \quad (4.6)$$

Şekil 4.7’de örnek bir sektör yoğunluk histogramı verilmektedir. Bu histogramda engel olan sektörlerde yoğunluk değeri yüksek çıkmaktadır. Engel olmayan sektörlerle karşılık gelen yoğunluk değeri bulunmamaktadır.



Şekil 4.7 Sektör yoğunluklarının histogram şeklinde gösterimi

iv. Aday sektör aralıklarının belirlenmesi

Yoğunluğa göre oluşturulan histogram üzerinden eşik değerinin altında kalan sektörler boş alan olarak tanımlanmakta ve diğer komşu boş sektörler ile gruplanarak aday sektör aralıkları belirlenmektedir. Herbir aday sektör aralığı η_i ($i = 1, 2, \dots, p$) arasındaki sektör sayıları ile tanımlanır.

v. Sektör seçimi

Literatürde gidilecek sektör seçimi yapılırken önce aday sektörler genişliklerine göre dar ve geniş olarak gruplanmakta ardından geniş sektörlerle öncelik vererek hedefe yakınlıklarına göre seçim yapıldığı görülmüştür (Borenstein, et. al., 1991). ATEKS platformu için geçilebilecek sektör aralıklarından hedefin bulunduğu sektöre en yakın olan sektör seçilir.

vi. Hızların belirlenmesi

Algoritmada son işlem olarak robota uygulanacak doğrusal ve açısal hızlar denklem 4.7 kullanılarak hesaplanmaktadır.

$$u = u_{\max} \left(1 - \frac{\min(h_{\max}, h_r)}{h_{\max}}\right)$$

$$w = w_{\max} \left(k\alpha k_s - \frac{\pi}{2}\right) \frac{2}{\pi} \quad (4.7)$$

Burada, u uygulanacak doğrusal hızı, w uygulanacak açısal hızı, $u_{\max}$ maksimum doğrusal hızı, $w_{\max}$ maksimum açısal hızı, $h_{\max}$ maksimum yoğunluk değerini, h_r ATEKS'in hareket doğrultusundaki sektörün yoğunluğunu, k_s açısal hızın uygulama kazanç ayarını göstermektedir.

Doğrusal hız hesaplanırken, robotun önündeki sektörün yoğunluğu herbir davranış için özel olarak belirlenmiş maksimum yoğunluk değeri ile karşılaştırılarak, belirlenen maksimum doğrusal hıza bağlı olarak hesaplanır. ATEKS'in doğrultusundaki sektörün yoğunluğu, maksimum yoğunluk sabitinden büyük ise robota uygulanacak doğrusal hız sıfıra çekilerek sadece açısal hız uygulanmakta ve böylece robot önündeki yakın engelden çarpmadan sakınılması sağlanmaktadır. İzlenecek hızların belirlenmesinin ardından bu değerler AKB – DSK'ya gönderilerek ATEKS'e uygulanmaktadır.

VFH algoritmasının, Hedefe gitme, Kapıdan Geçme ve Koridor Takibi davranışları için kullanılan parametre değerleri Çizelge 4.8’de verilmektedir.

Çizelge 4.8 Tüm davranışlar için VFH parametre değerleri

Davranış	Hedefe Gitme	Koridor Takibi	Kapıdan Geçme
Parametre	Değeri	Değeri	Değeri
Doğrusal Hız Max. Yoğunluk – h_{max}	5054054,0	99058,0	2,0
Izgaradaki Hücre Boyutu (X,Y) – $c_{i,j,x}$	0,1m x 0,1m	0,1m x 0, 1m	0,01m x 0,01m
Sektörler arası Açı - α	1,0°	1,0 °	0,1 °
Maksimum uzaklık Pozitif Katsayı – a	5,0	5,0	5,0
Maksimum uzaklık Pozitif Katsayı – b	2,0	2,0	2,0
Sektör Eşik Değeri	25,0	15,0	5,0
Max. Doğrusal Hız – u_{max}	0,2 m/s	0,2	0,15
Max Açısal Hız – w_{max}	0,6 rad/s	0,6	0,9
Nesne Büyültme Miktarı	0,2m	0,15	0

4.3.1 Hedefe gitme davranışı

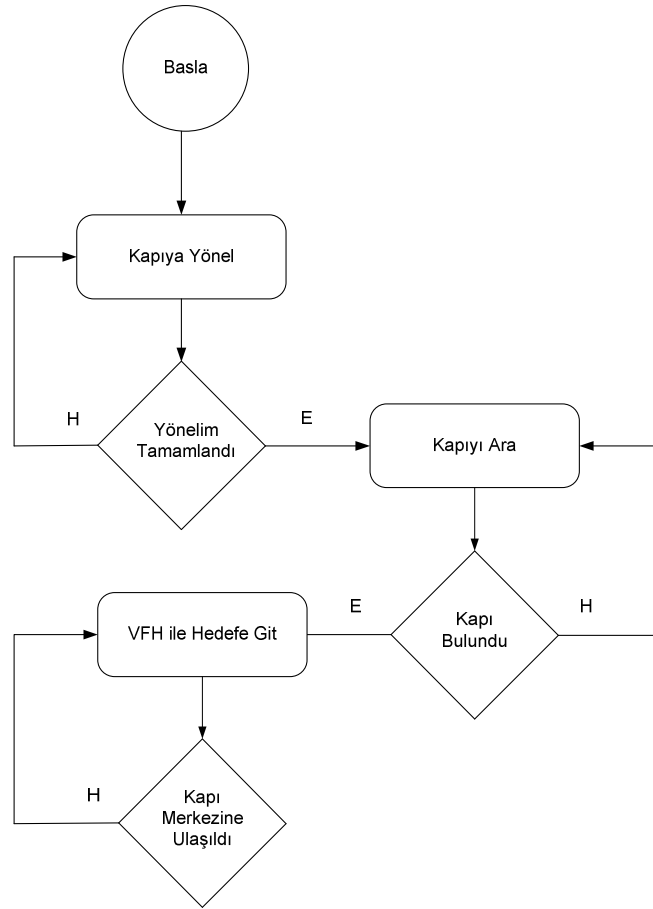
ATEKS platformu için hedefe gitme, bulunduğu noktadan hedef noktaya varmasıdır. Bu davranış için kullanılan VFH parametre değerleri Çizelge 4.8’de listelenmiştir. VFH algoritması, Kinect algılayıcısından gelen değerler kullanılarak ortamda hareket sağlamaktadır. Eğer bir noktada engel varsa, algoritmanın o noktaya aracı yöneltmemesi beklenmektedir. Hedefe gitme davranışından ikinci bir çarpışma önleyici olarak SMO15 algılayıcılarından gelen veriler kullanılmaktadır. Bu algılayıcılar için çarpışma engellemede kullanılan parametre eşik değerleri Çizelge 4.9’da verilmiştir.

4.3.2 Koridor takibi davranışı

Koridor takibi davranışı, boş veya engelli bir koridor içerisinde bir noktadan başka bir noktaya engelden kaçınarak koridor duvarlarına paralel hareket etme işlemidir. Bu davranış için kullanılan VFH parametre değerleri Çizelge 4.8’de listelenmiştir. Eğer koridor darsa, ATEKS bu davranış boyunca hareketi koridoru ortalayarak gerçeklemektedir. Eğer geniş koridor ise, sağ duvarı referans alıp hareket etmektedir. Hareket sırasında engellerden sakınma davranışı aktif olup çarpışma engelleme için kullanılan parametre değerleri Çizelge 4.9’da verilmiştir.

4.3.3 Kapıdan geçme davranışı

Kapıdan geçme davranışı, kapının 1,5m önünden başlayıp kapı merkezine gelindiğinde bitmektedir. Herhangi bir görevi gerçeklemek üzere ATEKS platformu ortamda ilerlerken, kapıdan geçmesi gerektiğinde kapıya belli bir mesafe kala kapıdan geçme davranışı tetiklenmektedir. Bu davranışın akış diyagramı Şekil 4.8’de verilmiştir.



Şekil 4.8 Kapıdan geçme davranışı akış diyagramı

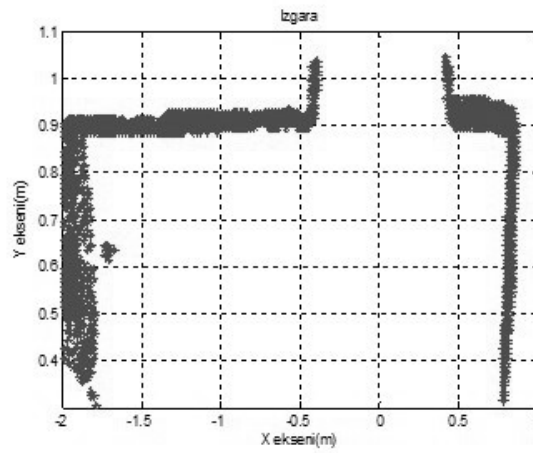
Kapıdan geçme davranışı sırasında ATEKS platformu küresel konum yerine kapının merkez noktasını Kinect algılayıcısı ile bulup onu hedef olarak ayarlar ve hareket eder. Yukarıdaki şekil 4.8’de verilen akış diyagramı detaylandırılmaktadır.

i. Kapıya yönelim

Kapıdan geçme davranışının ilk aşamasında, ATEKS platformunun kapıya yaklaşık olarak dik konuma gelmesi sağlanmaktadır. Bu, Kinect algılayıcılarının kapıyı daha iyi algılaması için gereklidir. Burada, ATEKS platformunun küresel koordinatı ve kapı merkezinin küresel koordinatı dikkate alınarak araç kapıya yönelir.

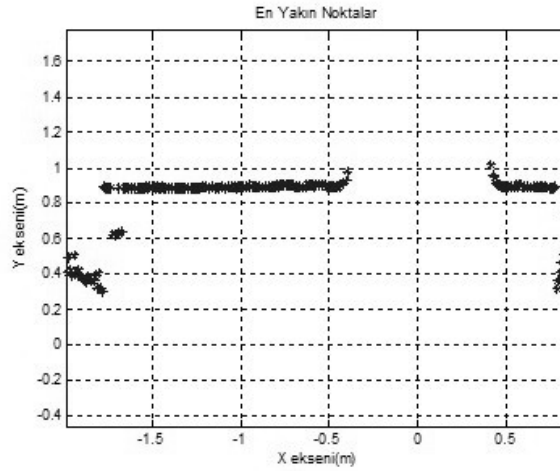
ii. Kapıyı arama

Kapı arama için Kinect algılayıcıları ile elde edilen üç boyutlu veriler kullanılmaktadır. Öncelikle bu veri şekil 4.9'daki gibi iki boyutlu veriye dönüştürülmektedir. İki boyutlu veriye dönüştürülen algılayıcı değerleri VFH algoritmasına benzer bir ızgaraya yerleştirilir.



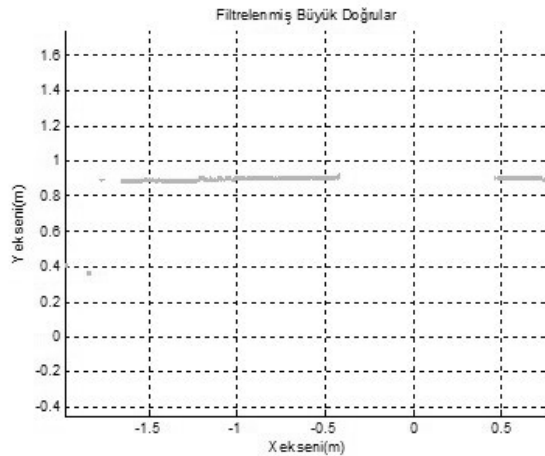
Şekil 4.9 Kinect algılayıcısı verisinin iki boyuta indirilmesi

Şekil 4.10'daki boyut dönüşümünde gerçek ortamda aynı düzlemin farklı yüksekliklerine karşılık gelen veriler ilgili noktanın Kinect algılayıcıya olan uzaklık farkından dolayı farklı mesafelerde gözükebilmektedir. Bu problemi çözmek için bu verilerden (ızgara hücrelerinden) en yakın olanları alınarak şekil 4.11 elde edilir.



Şekil 4.10 En yakın ızgara hücreleri

Şekil 4.10'daki veriler kullanılarak, belli eşik değeri geçen doğrular bulunmaktadır. Bu küçük doğrulardan uzaklıkları tanımlanmış eşik değerinden küçük ve iki doğru arasındaki eğim farkının mutlak değeri belirli bir eşik değerinin altında kalan doğrular birleştirilerek daha büyük doğrular elde edilmektedir. Şekil 4.11'de gösterilen büyük doğruların filtrelenmesinin ardından doğrular arasındaki uzaklık tanımlanan kapı genişliğinin komşuluğunda ise olası kapı olarak belirlenmekte ve bu iki doğru üzerinden olası kapının ortasının robota göre yerel koordinatlarda konumu hesaplanmaktadır.



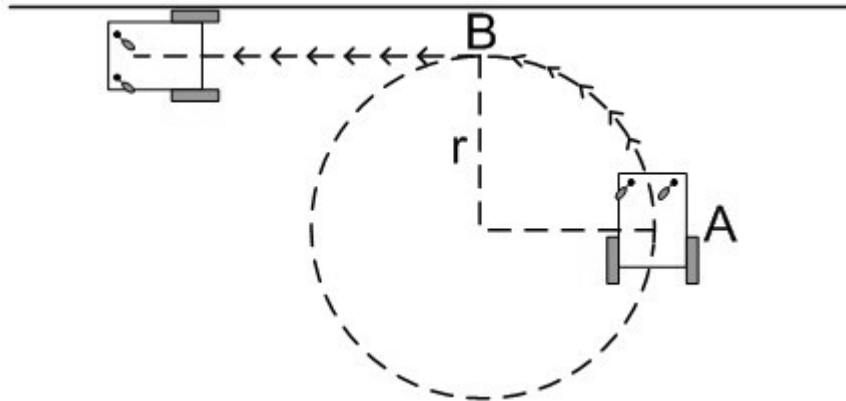
Şekil 4.11 Filtrelenmiş büyük ızgaralar örneği

iii. VFH ile hedefe gitme

Kapı tanınması ve kapı orta noktasının bulunmasının ardından, bulunan kapı ortası ızgara üzerinde yerel hedef işaretlenmekte ve Çizelge 4.8’de kullanılan VFH algoritmasının parametreleri ile hedefe gitme işlemi gerçekleştirilmektedir.

4.3.4 Cisme yanaşma davranışı

Bu davranış ile ATEKS platformunun önünde durduğu duvar vb. düz bir yüzeye sağ taraftan paralel yanaşması planlanmıştır. Bu davranışta yerel algılama için Kinect algılayıcısı kullanılmaktadır. ATEKS platformu, duvara dik olarak yöneldiği bir durumdan, yanaşma davranışının sonuna kadar Şekil 4.12’de ok işaretleri ile verilen rotayı takip etmektedir.



Şekil 4.12 ATEKS izlenecek cisme yanaşma rotası

Bu davranış başlangıcından ATEKS platformu duvara tam dik yönelimde olmayabilir. Bunun için, Kinect algılayıcısı ile duvar tanımlanmakta ve ATEKS platformunun duvara göre açısı hesaplanmaktadır. Daha sonra, platform duvara yönelimi dik olacak şekilde dönmektedir. Bu durumda, Kinect kullanılarak platform ile duvar arası mesafeyi ölçmekte ve dönme işlemi için dönüş yarıçapı hesaplanmaktadır.

ATEKS platformu B noktasına kadar, A – B yayını takip etmektedir. B noktasına geldiğinde platform duvara paralel olmaktadır. Bunun kontrolü, sağ taraftaki SMO15’ler arası mesafe farkı kontrol edilerek tespit belirlenmektedir. Bu noktadan sonra potansiyel alanlar kullanılarak duvara ve park noktasına daha hassas bir şekilde yanaşılmaktadır.

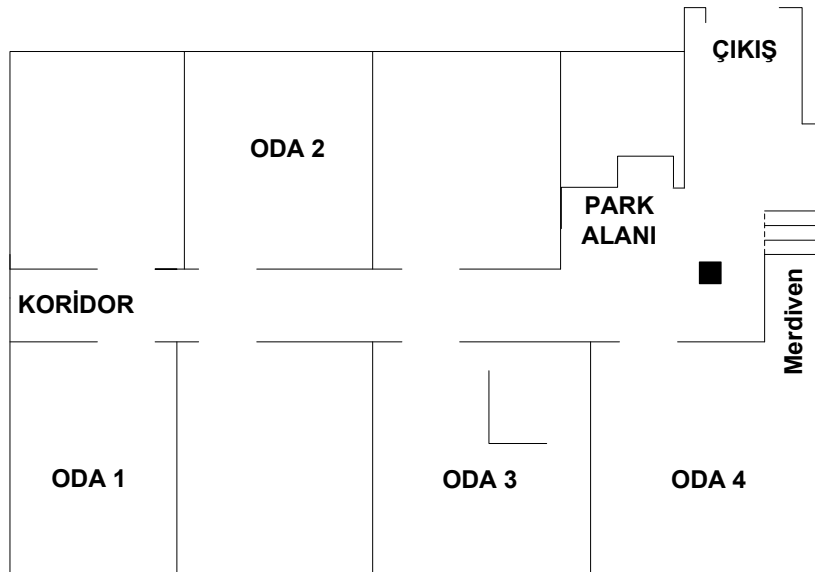
Çizelge 4.9 Davranışlar için çarpışma engelleme parametre değerleri

Parametre	Hedefe Gitme	Koridor Takibi	Kapıdan Geçme
Ön Kritik Durma Mesafesi	0,6 m	0,40 m	0,30 m
Yan Kritik Durma Mesafesi	0,21 m	0,18 m	0,08 m
Arka Kritik Durma Mesafesi	0,5 m	0,75 m	0,60 m

BÖLÜM 5

ATEKS PLATFORM TESTLERİ

ATEKS platformu için geliştirilen yüksek seviye kontrol mimarisi benzetim ortamında ve gerçek ortamda test edilmiştir. Benzetim testleri GAZEBO (GAZEBO Robot Simulation Software, 2014) robot benzetim ortamına aktarılmasıyla elde edilen ortamda yapılmıştır. Gerçek ortam testleri Şekil 5.1’de krokisi verilen Osmangazi Teknoparkı Giriş Katı’nda yapılmıştır.



Şekil 5.1 Teknopark test alanı

Bu ortam odalar, koridor, park alanı ve çıkış bölgesinden oluşmaktadır. Şekil 5.2’de krokideki Oda1’in fotoğrafı verilmektedir. Oda1, 5m X 7m boyutlarında olup, burada hedefe gitme ve kapıdan geçme davranışları gerçekleştirilmektedir.



Şekil 5.2 Teknopark oda1

Şekil 5.3'te krokide bulunan koridorun fotoğrafı gösterilmektedir. Koridor 2m genişliğinde ve 15,2m uzunluğunda olup, koridor takibi davranışı gerçekleştirilmektedir.



Şekil 5.3 Teknopark koridor

Şekil 5.4'te krokide gösterilen park alanının fotoğrafı bulunmaktadır. Park alanı, 5m X 5m boyutlarında olup, park alanı içerisinde hedefe git veya cisme yanaşma davranışları gerçekleştirilmektedir.



Şekil 5.4 Teknopark park alanı

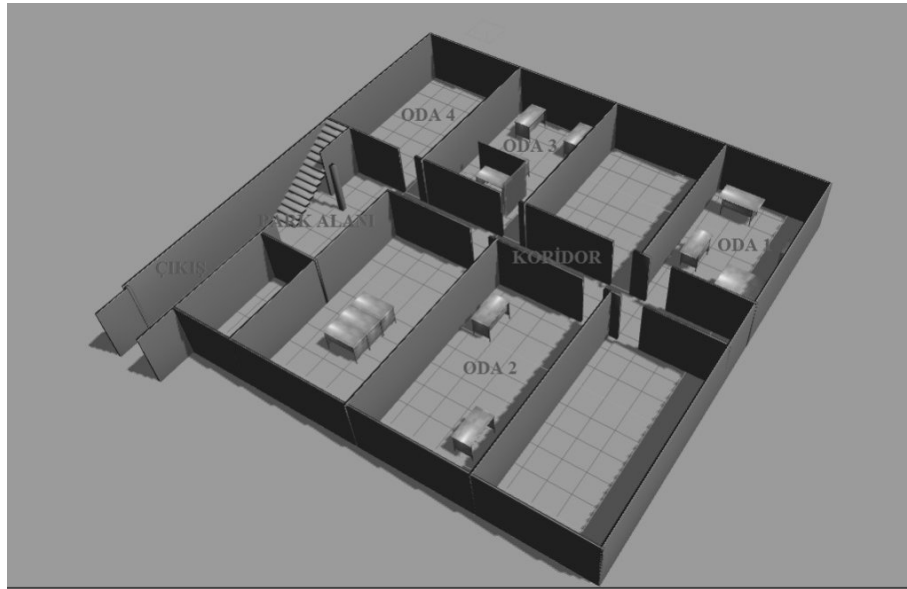
Şekil 5.5'te krokide bina çıkışı olup koridor olarak kabul edilmektedir. Burada koridor takibi davranışı yapılmaktadır.



Şekil 5.5 Teknopark çıkış

5.1 Benzetim Ortamı Testleri

Benzetim ortamı olarak robotik arařtırmalarda yaygın olarak kullanılan GAZEBO kullanılmaktadır. ATEKS platformunun dinamik ve kinematik model bileřenleri bu ortama aktarılmıřtır. Ayrıca, Osmangazi Teknopark Ortamı (bkz. řekil 5.1 – 5.5), řekil 5.6'daki gibi üç boyutlu olarak GAZEBO ortamına aktarılmıřtır.

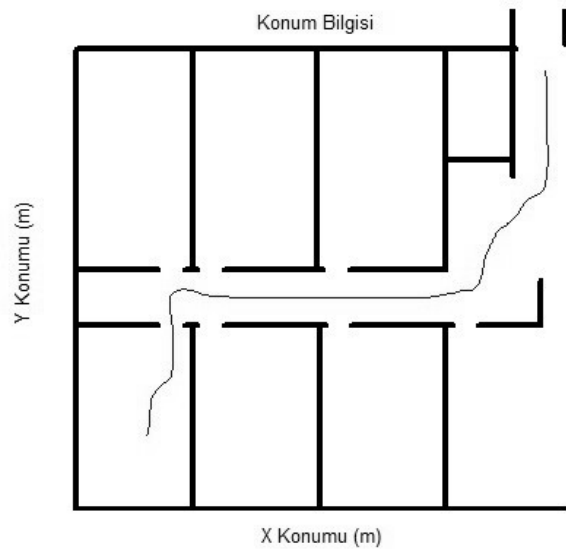


řekil 5.6 GAZEBO teknopark giriş katı modeli

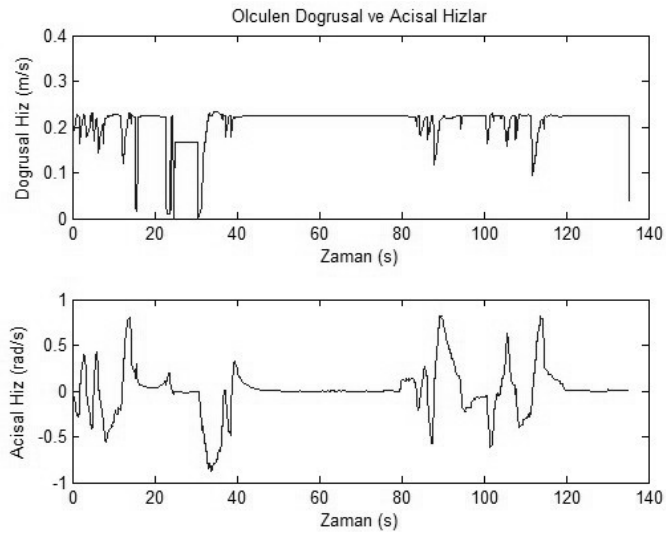
Benzetim ortamındaki testler için dört farklı senaryo belirlenmiřtir. Her bir senaryo için ölçölen açısal ve dođrusal hızlar, ATEKS konum bilgileri GAZEBO ortamında kayıt altına alınmıřtır.

Test 1 – oda1’den teknopark çıkışı;

Test1’de ATEKS Oda1’den Çıkışa doğru hareket etmektedir. Şekil 5.7’de bu hareket sırasında ATEKS platformunun bulunduğu konum değerleri gösterilmektedir. Şekil 5.8’de ölçülen açısal ve doğrusal hız bilgileri gösterilmektedir.



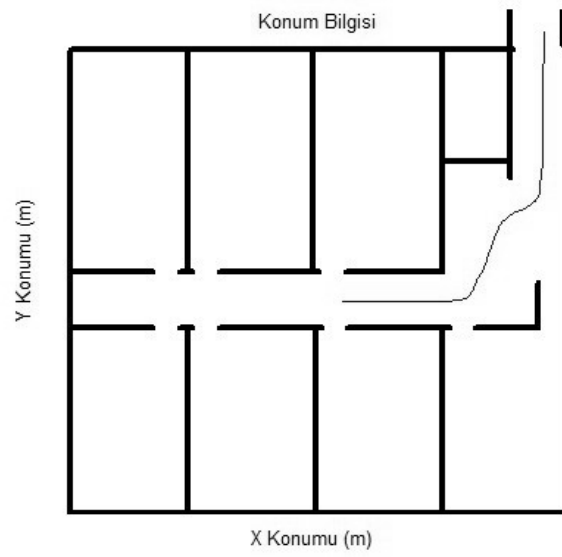
Şekil 5.7 GAZEBO test1 alınan konum bilgileri



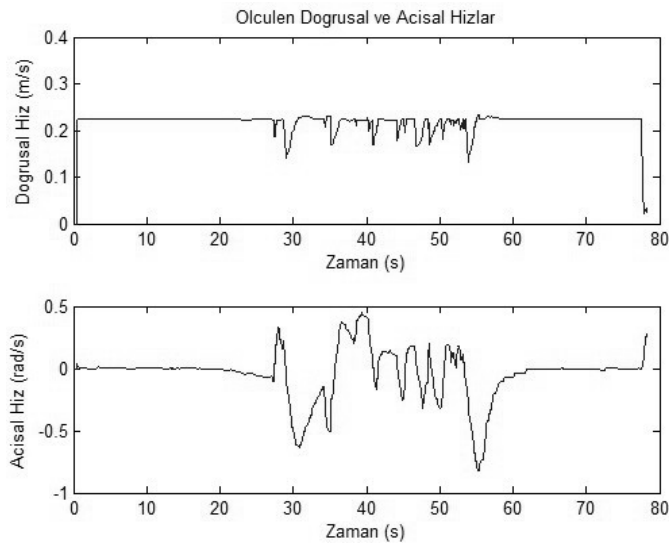
Şekil 5.8 GAZEBO test1 ölçülen açısal ve doğrusal hız değerleri

Test2 – teknopark çıkıştan oda3 öü:

Test2’de ATEKS Çıkıştan Oda3’ün önüne doğru hareket etmektedir. Şekil 5.9’da bu hareket sırasında ATEKS platformunun bulunduğu konum değerleri gösterilmektedir. Şekil 5.10’da ölçülen açısal ve doğrusal hız bilgileri gösterilmektedir.



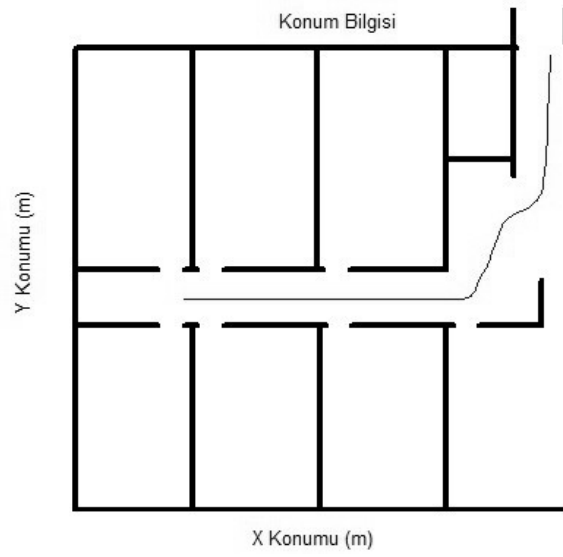
Şekil 5.9 GAZEBO test2 alınan konum bilgileri



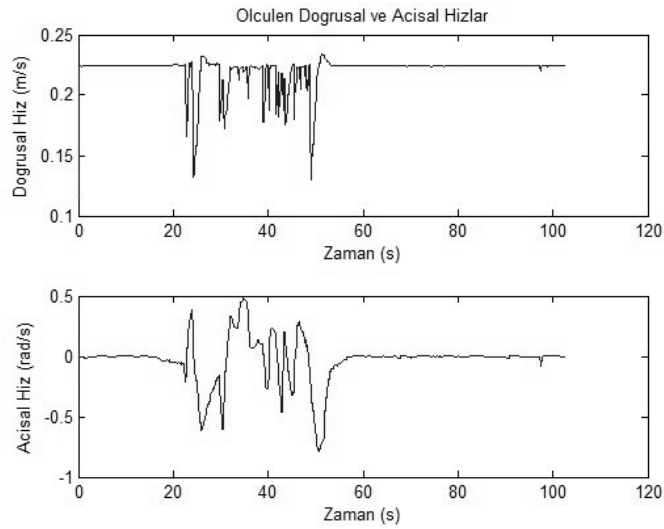
Şekil 5.10 GAZEBO test2 ölçülen açısal ve doğrusal hız bilgileri

Test3 – teknopark çıkıştan oda1 Önü;

Test3'te ATEKS, Çıkıştan Oda1 önüne doğru hareket etmektedir. Şekil 5.11'de hareket sırasında ATEKS platformunun bulunduğu konum değerleri gösterilmektedir. Şekil 5.12'de ölçülen açısal ve doğrusal hız bilgileri gösterilmektedir.



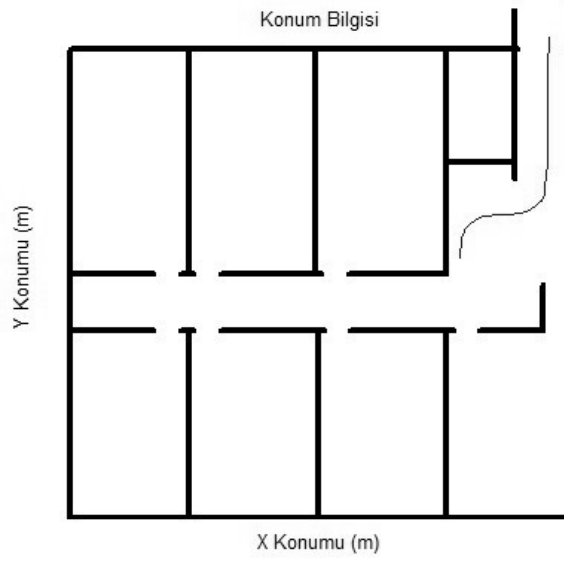
Şekil 5.11 GAZEBO test3 alınan konum bilgileri



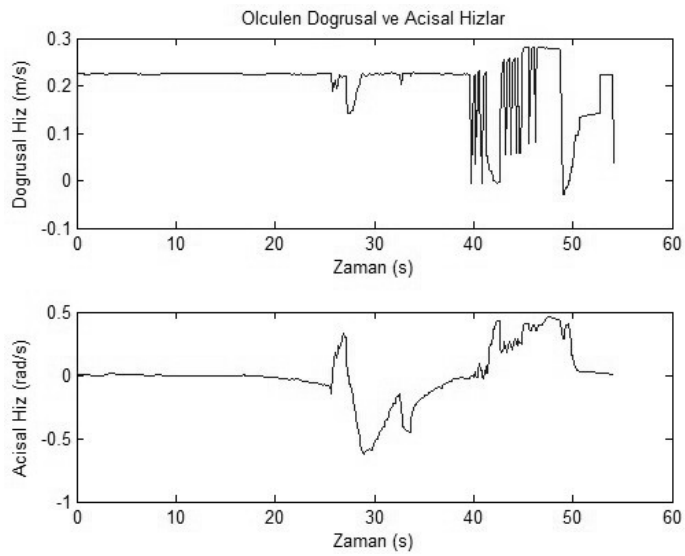
Şekil 5.12 GAZEBO test3 ölçülen açısal ve doğrusal hız bilgileri

Test4 – çıkıştan park alanında park etme;

Test4'te ATEKS, Çıkıştan Park Alanına gelip park etme davranışı gerçekleştirilmektedir. Şekil 5.13'te bu hareket sırasında ATEKS platformunun bulunduğu konum değerleri gösterilmektedir. Şekil 5.14'te ölçülen açısal ve doğrusal hız bilgileri gösterilmektedir.



Şekil 5.13 GAZEBO test4 alınan konum bilgileri



Şekil 5.14 GAZEBO test4 ölçülen açısal ve doğrusal hız bilgileri

GAZEBO da yapılan denemeler sonucunda, geliştirilen kontrol mimarisi ve davranışlar test edilmiştir. Bu aşamadan sonra gerçek ortam testlerine geçilmiştir.

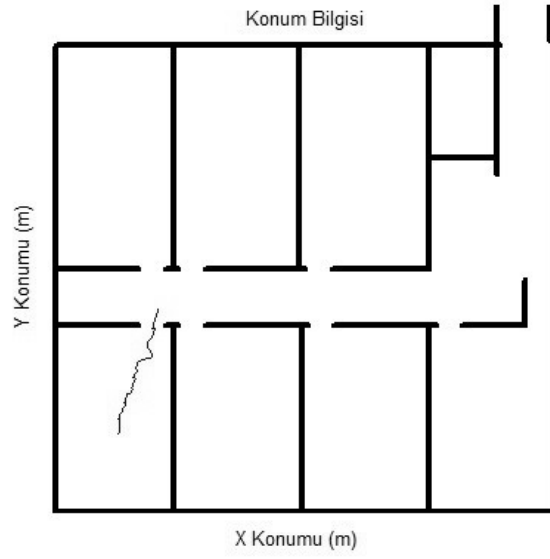
5.2 Gerçek Ortam Testleri

GAZEBO benzetim ortamında, araç kinematiği, dinamiği ve algılayıcılar oldukça gerçeğe yakın modellenmiş olsa da, bazı farklılıklar oluşabilmektedir. Örneğin; benzetim ortamındaki SMO15 algılayıcısı oldukça hassas mesafe üretirken, gerçek ortamda yansıma vb. problemler olabilmektedir. Benzetim ortamı ile gerçek ortam arasındaki diğer önemli bir fark, konum bilgisindedir. Gerçek ortamdaki konum bilgisi, benzetim ortamında alınan konum bilgisi kadar düzgün bir bilgi değildir. Bu durumdan dolayı davranışları gerçeklerken farklı problemler ortaya çıkabilmektedir. Bundan dolayı kapı geçişlerinde ve cisme yanaşma davranışlarında Kinect, SMO15 algılayıcılarından elde edilen bağıl konum değerleri kullanılmaktadır.

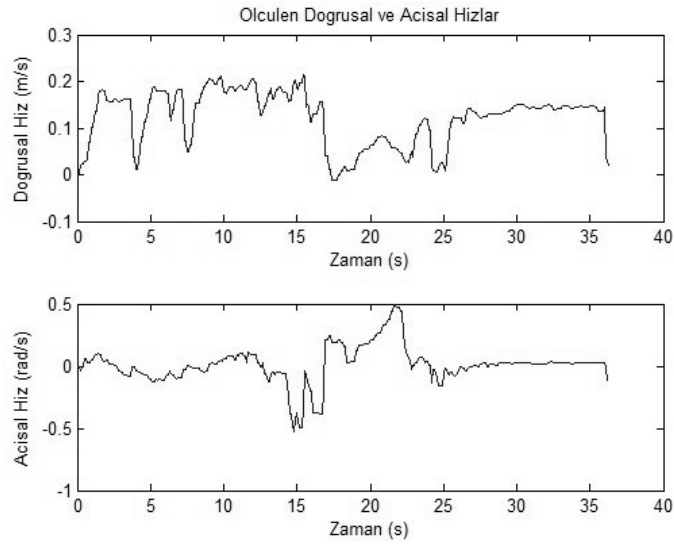
Gerçek ortamda, kapıdan geçme, koridor takibi, cisme yanaşma davranışlarını içerecek şekilde farklı testler yapılmıştır.

Test 1 – oda1'den koridora;

Şekil 5.15'te, Oda1'den başlayıp kapıdan geçip koridordaki bir hedef noktada biten ATEKS platformunun bulunduğu konum değerleri gösterilmektedir. Şekil 5.16'da ölçülen açısal ve doğrusal hız değerleri gösterilmektedir.



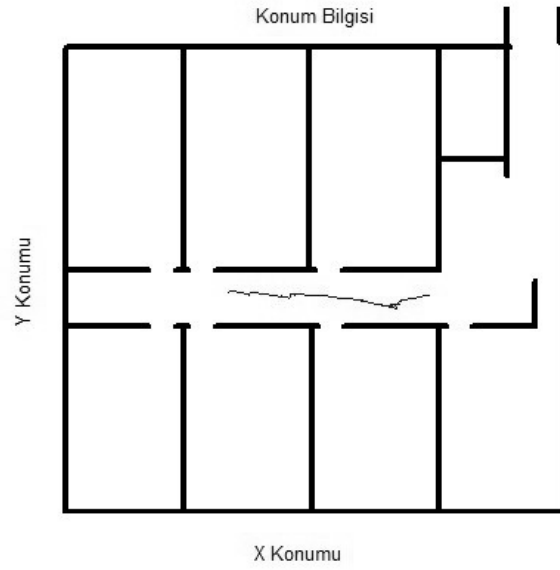
Şekil 5.15 Gerçek ortam kapıdan geçme testi alınan konum bilgileri



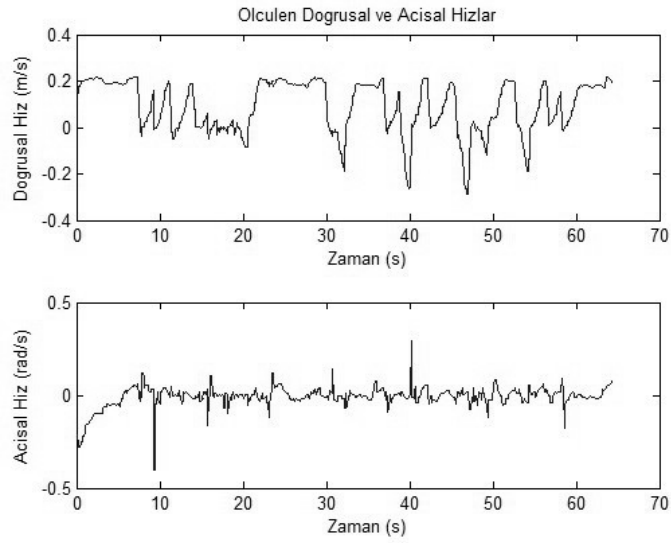
Şekil 5.16 Gerçek ortam kapıdan geçme testi ölçülen açısal ve doğrusal hız değerleri

Test 2 – koridor takibi davranışı;

Şekil 5.17’de, koridordaki bir noktadan diğer noktaya hareket esnasında ATEKS platformunun bulunduğu konum değerlerini göstermektedir. Şekil 5.18’de ölçülen açısal ve doğrusal hız bilgileri gösterilmektedir.



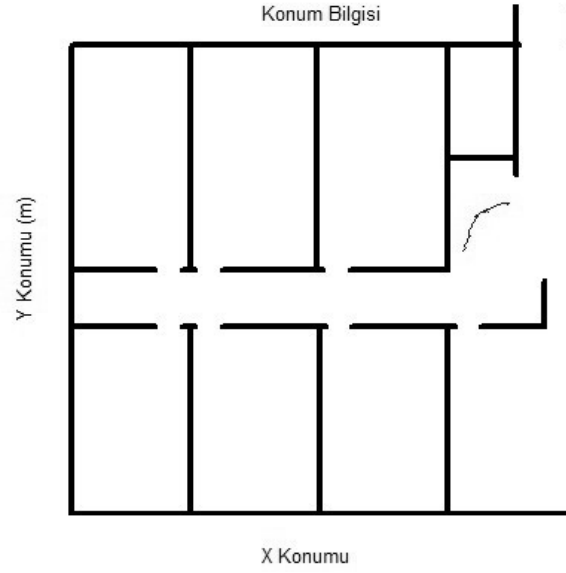
Şekil 5.17 Gerçek ortam koridor takibi testi alınan konum bilgileri



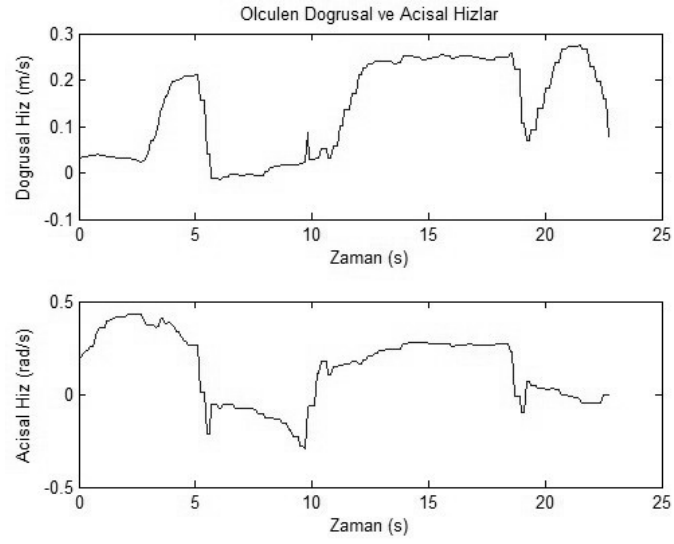
Şekil 5.18 Gerçek ortam koridor takibi testi ölçülen açısal ve doğrusal hız bilgileri

Test 3 – cisme yanaşma davranışı;

Şekil 5.19’da park alanında bir noktadan başlayıp hedeflenen duvara yanaşan cisme yanaşma davranışı sırasında ATEKS platformunun konum değerleri gösterilmektedir. Şekil 5.20’de ölçülen açısal ve doğrusal hız değerleri gösterilmektedir.



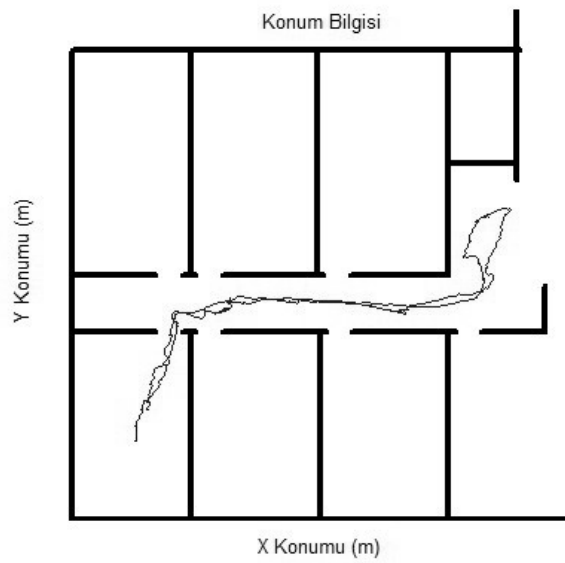
Şekil 5.19 Gerçek ortam cisme yanaşma testi alınan konum bilgileri



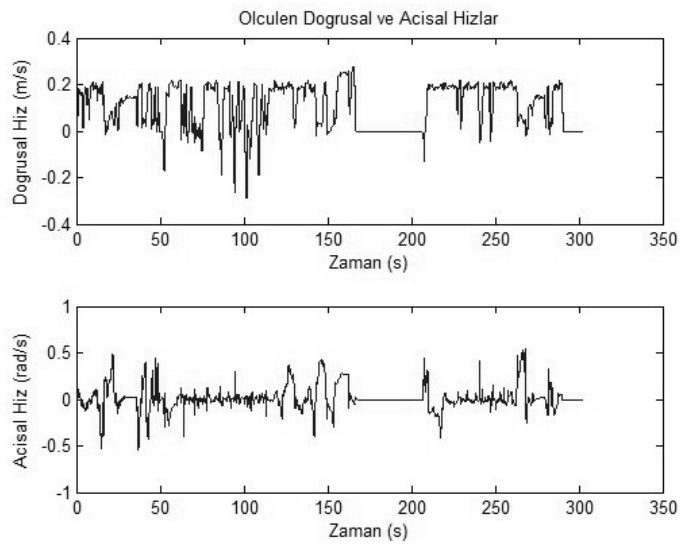
Şekil 5.20 Gerçek ortam cisme yanaşma testi ölçülen açısız ve doğrusal hız bilgileri

Test 4 – tam ortam testi;

Bu testte ATEKS platformu Oda 1’den çıkıp Park alanına gitmekte, buradan dönerek tekrar Oda1’e geri dönmektedir. Şekil 5.21’de bu test sırasında alınan konum değerleri gösterilmektedir. Şekil 5.22’de ölçülen açısal ve doğrusal hız değerleri gösterilmektedir.



Şekil 5.21 Gerçek ortam tam ortam testi alınan konum bilgileri



Şekil 5.22 Gerçek ortam tam ortam testi ölçülen açısal ve doğrusal hız bilgileri

Bu testlerin dışında birçok gerçek ortam testi yapılmıştır. ATEKS platformunun testlerine ait videolar (İnovasyon Mühendislik, 2014) kaynağından izlenebilir.

SONUÇLAR

Bu tez çalışmasında elektrikli bir sandalye platformunun özerk kontrolü için öncelikle platforma Akıllı Kontrol Birimi (AKB) donanımı ve çeşitli algılayıcılar entegre edilmiştir. Daha sonra, AKB’de özerk kontrolü sağlayacak kontrol mimarisi tasarlanmış ve gerçekleştirilmiştir. Tasarlanan kontrol mimarisi, uluslararası çalışmalarda kullanılan ROS ara katmanını kullanmaktadır. ATEKS platformuna özgün ROS düğümleri ve gerekli tüm sürücüler gerçekleştirilmiştir. Bu yapıda özerk kontrol davranışlarını gerçeklemek üzere geliştirilen ROS düğümünde, davranışların senkronizasyonu için Sonlu Durum Makinesi (SDM) kullanılmaktadır. ATEKS’in akıllı kontrolü, ROS ve SDM’nin birlikte çalıştığı melez bir yapı ile sağlanmıştır. Vektör alan histogramı algoritması parametrik girdileri olacak şekilde tasarlanmış ve farklı davranışların içerisinde parametrik değişiklikler yapılarak kullanılmıştır.

Geliştirilen kontrol mimarisi ve davranışlar, öncelikle benzetim ortamında test edilmiştir. Bu testlerden sonra, yapılan gerçek ortam uygulamalarında, farklı problemler ile karşılaşılmış ve bunlar üzerinde iyileştirmeler yapılmıştır. Benzetim ortamı ve gerçek ortam arası farklılıklar tespit edilmiştir. Geliştirilen platform, gezgin robot araştırmaları ve engelli insanlar için kullanılabilecektir.

ATEKS platformuna ortam navigasyonunda kullanılmak üzere, son yıllarda gezgin robot araştırmalarında yaygın bir şekilde kullanılmaya başlanan Microsoft Kinect algılayıcısı entegre edilmiş olup, hedeflenen davranışları gerçekleyecek şekilde iç ortamlar için bazı temel örüntü tanıma fonksiyonları geliştirilmiştir. Literatürde bu algılayıcı ile iç ortam örüntü tanımaya yönelik yoğun çalışmalar yapılmaktadır. Bu çalışmaların ATEKS platformunda entegrasyonu daha ileri seviye davranışları mümkün kılacaktır.

KAYNAKLAR DİZİNİ

Active Media Inc, 2014, <http://www.mobilerobots.com>.

Agerwala T., December 1979, Special Features Putting Petri Nets to Work, IEEE IEEE Computer Society, Vol. 12, 85 – 94 p.

Akar B., Yayan U., Yücel H., Bayar V. ve Yazıcı A., Eylül 2013, Akıllı Tekerlekli Sandalye Kontrolü İçin Sonlu Durum Makinesi Tasarımı, Otomatik Kontrol Ulusal Toplantısı (TOK2013) Bildiriler Kitabı, Malatya, 114 – 116 s.

Akar B., İnan F., Yayan U. ve Yazıcı A., Nisan 2014, Akıllı Tekerlekli Sandalye İçin İç Ortam Navigasyon Yazılımı Geliştirilmesi, Sinyal İşleme ve İletişim Uygulamaları (SIU 2014) (baskıda).

Albus J.S., Outline of a theory of intellegence, May 1991, IEEE Trans. On Systems, Man and Cybernatics, 21(3): 473 – 509 p.

Albus J.S., McCain H.G. and Lumia R., 1989, NASA / NBS Standard Reference Model for Telerobot Control System Architecture (NASREM), Technical Report 1235, National Institute of Standards and Technology.

Alami R., Chatila R., Fleury S., Ghallab M. and Ingrand F.F., Spring 1998, An Architecture for Autonomy, To appear on Internation Journal of Robotics Research.

An D. and Wang H., 2004, VPH: A New Laser Radar Based Obstacle Avoidance Method for Intelligent Mobile Robots, Intelligent Control and Automation, Vol.5. 4681 – 4685 p.

KAYNAKLAR DİZİNİ (devam)

Aria Manuals ver. 2.5.1, 2014, <http://robots.mobilerobots.com>.

ART-Linux, 2014, <http://art-linux.sourceforge.net>.

Bayar V., Akar B., Yayan U., Yazıcı A. ve Yavuz H. S., 23-25 Nisan 2014, Bulanık Mantık Tabanlı Kontrol Sisteminin ROS Ara Katmanında Gerçeklenmesi, SİU 2014, (baskıda).

Bayar V., Akar B., Yayan U., Yazıcı A. and Yavuz H. S., 23-25 Nisan 2014, Fuzzy Logic Based Design of Classical Behaviors for Mobile Robots in ROS Middleware, Inista 2014, (in press).

Berenji H. R., Yager R. R. and Zadeh L.A., 1991, An Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, 69 – 96 p.

Borenstein J. and Koren Y., June 1991, The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots, IEEE Transactions On Robotics and Automation, Vol. 7, No. 3, 278 – 288 p.

Brooks R.A., Mar. 1986, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, RA-2(1): 14 – 23 p.

Charifa S. and Bikdash M., 2009, Adaptive Boundary-following Algorithm Guided By Artificial Potential Field for Robot Navigation, Robotic Intelligence in Informationally Structured Space, RIIS '09. IEEE Workshop, 38 – 45 p.

KAYNAKLAR DİZİNİ (devam)

Coulter R. C., January 1992, Implementation of the Pure Pursuit Path Tracking Algorithm, The Robotics Institute Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

Elkady A. and Sobh T., 2012, Robotics Middleware: a Comprehensive Literature Survey and Attribute-Based Bibliography, Hindawi Publishing Corporation Journal of Robotics.

Gao J., Gao B. X., Zhu W., Zhu J. and Wei B., June 2008, Fault-tolerant and High Reliability Space Robot Design and Research, Neural Networks, IEEE World Congress on Computational Intelligence, 2413 – 2417 p.

GAZEBO Robot Simulation Software, 2014, <http://gazebosim.org>.

Huang, L., October 2007, A Potential Field Approach for Controlling a Mobile Robot to Track a Moving Target, Intelligent Control, IEEE 22nd International Symposium, 65 – 70 p.

İnovasyon Mühendislik, 2014, <http://inovasyonmuhendislik.com>.

Khatib O., March 1985, Real-Time Obstacle Avoidance For Manipulators and Mobile Robots, Robotics and Automation. Proceedings. 1985 IEEE International Conference on, Vol. 2, 500 – 505 p.

Lee D. and Chung W., October 2006, Discrete-Status-Based Localization for Indoor Service Robots, IEEE Transactions on Industrial Electronics, Vol. 53, No. 5, 1737 – 1746 p.

KAYNAKLAR DİZİNİ (devam)

Levine S.P and Bell D.A, December 1999, The NavChair Assistive Wheelchair Navigation System, IEEE Transaction Rehabilitation Engineering, Vol. 10, 443 – 451 p.

Ling W., 2013, Fuzzy Control of Obstacle Avoidance for Mobile Robot Based on Single Chip Microcomputer, International Conference on Artificial Intelligence and Software Engineering (ICAISE).

LISP Programming Language, 2014, <http://common-lisp.net>.

Liu J., Zhabg H., Fan B., Wang G. and Wu J., June 2010, A novel economical Embedded Multi-Mode Intelligent Control System for Intelligent Wheelchair, 2010 International Conference on Computing, Control and Industrial Engineering (CCIE), Vol. 1, 156 – 159 p.

Liu Z., Jing R., Ding X. and Li J., October 2008, Trajectory Tracking Control of Wheeled Mobile Robots Based on the Artificial Potential Field, Vol. 7, 382 – 387 p.

Mahulea C. and Kloetzer M., September 2012, Online Petri Net Based Algorithm for Planning and Controlling Mobile Robots, Emerging Technologies & Factory Automation (ETFA), IEEE 17th Conference, 1 – 8 p.

Mariappan M., Ganesan T. and Ramu V., 1 Dec 2011, Fuzzy Controlled Obstacle Avoidance System (OAS) for OTOROB Flexible Robotic Arm Safety System, International Journal of Robotics Research and Development (IJRRD), Vol.1, 1 – 20 p.

KAYNAKLAR DİZİNİ (devam)

Mark W. S. and Chaouki A., Series Editors, March 2012, Mobile Robotics for Multidisciplinary Study, Morgan & Claypool Publishers.

Mataric M., 1992, Behavior – Based Control: Main Properties and Implications, Proceedings of Workshop on Intelligent Control Systems, International Conference on Robotics and Automation, Nice, France.

Matsumoto O. and Komoriya K., October 2006, Autonomous Traveling Control of the ‘TAO Aicle’ Intelligent Wheelchair, Proc. of the 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing-China, 4322 – 4327 p.

Minguez J. and Montano L., Feb. 2004, Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios, IEEE Trans. Robot. Autom., Vol. 20, No. 1, 45 – 59 p.

MIRO – Middleware for Robots, 2014, <http://www.ohloh.net/p/miro-middleware>.

Moon C. and Chung W., May 3-8 2010, Design of Navigation Behaviors and the Selection Framework with Generalized Stochastic Petri Nets toward dependable navigation of a mobile robot, 2010 IEEE International Conference on Robotics and Automation, Anchorage Convention District, Anchorage, Alaska, USA, 2989 – 4729 p.

Open Agent Architecture (OAA), 2007, Developer’s Guide, v.2.3.2.

OPENRTM-aist Middleware, 2014, <http://openrtm.org/openrtm/en>.

KAYNAKLAR DİZİNİ (devam)

Özçelikörs M., Çoskun A., Say G., Yayan U., Akçakoca M. ve Kılıç İ., Potansiyel Alanlar ile Kinect Tabanlı Akıllı Tekerlekli Sandalye Navigasyonu, Sinyal İşleme ve İletişim Uygulamaları (SIU 2014) (baskıda).

Özkan M., 2007, Farklı Özelliklere Sahip Gezgin Robot Grubunun Dağıtık Kontrolü, Doktora Tezi, Eskişehir Osmangazi Üniversitesi Fen bilimleri Enstitüsü.

Palacin J., Salse J. A., Valgañón I. and Clua X., October 2004, Building a Mobile Robot for a Floor-Cleaning Operation in Domestic Environments, 1418, IEEE Transactions on Instrumentation and Measurement, Vol. 53, No. 5, 1418 – 1424 p.

Parlaktuna O., Bakla B., Özkan M. ve Yazıcı A., 16 – 18 Mayıs 2005, Bulanık Mantık Yöntemiyle Gezgin Robot Navigasyonu, SIU 2005 (IEEE 13. Sinyal İşleme ve İletişim Uygulamaları).

Phillips R.G. and Sutherland H.A., March 1985, Performance Evaluation of Alternative Robot Controller Architectures Using a Finite State Machine Emulator, Robotics and Automation. Proceedings. 1985 IEEE International Conference, Vol. 2, 638 – 643 p.

Point Cloud Library, 2014, <http://pointclouds.org>.

QNX Real Time Operating System, 2014, <http://www.qnx.com>.

Quinlan S. and Khatib O., 1993, Elastic Bands: Connecting Path Planning and Control, in Proc. IEEE Int. Conf. Robot. Autom., 802 – 807 p.

KAYNAKLAR DİZİNİ (devam)

Rezaee, H. and Abdollahi, F., July 2011, Mobile Robots Cooperative Control and Obstacle Avoidance Using Potential Field, Advanced Intelligent Mechatronics (AIM), IEEE/ASME International Conference, 61 – 66 p.

ROS Project, 2014, <http://www.ros.org>.

Roth B. H., July 1985, A Blackboard Architecture for Control, Artificial Intelligence, 26(3): 251 – 321 p.

Shieh M.Y., Hsieh J.C. and Cheng C.P., 10-13 Oct. 2004, Design of an Intelligent Hospital Service Robot and its Applications, Systems, Man and Cybernetics, 2004 IEEE International Conference on, Vol: 5, 4377 – 4382 p.

Siciliano B. and Khatib S., 2008, Springer Handbook of Robotics, Springer – Verlag Berlin Heidelberg.

Siddaiyan S. and R. Arokiasamy W., 2012, Reliable Obstacle Avoidance for Mobile Robot Navigation Coupled with A* Algorithm Through Fuzzy Logic and Knowledge Based Systems, 2012 International Conference on Computer Technology and Science (ICCTS), Singapore, IPCSIT vol. 47 IACSIT Press.

Skilligent – Task and Skill Learning for Multifunctional Service Robots, 2014 <http://www.skilligent.com>.

Slack M.G., March 1993, Navigation Templates: Mediating Qualitative Guidance and Quantitative Control in Mobile Robots, IEEE Trans., On Systems, Man and Cybernetics, 23(2): 452 – 466 p.

KAYNAKLAR DİZİNİ (devam)

SMO15 – Sesüstü mesafe ölçer, 2014,
<http://inovasyonmuhendislik.com/urun/3/ultrasonik-mesafe-olcer>.

Stanford Research Institute, 2014, <http://www.sri.com>.

Synodinos A., Aspragathos and N.A., July 2010, Path Planning of a Mobile Robot Using Solid Modeling Techniques on Potential Fields, Mechatronics and Embedded Systems and Applications (MESA), IEEE/ASME International Conference, 549 – 553 p.

T-Kernel, 2014, <http://www.t-engine.org/what-is-t-kernel/t-kernel>,

The Orocos – Open Robot Control Software, 2014, <http://www.orocos.org>.

The Player Project, 2014, <http://playerstage.sourceforge.net>.

Thongchai S. and Kawamura K., September 25-27 - 2000, Application of Fuzzy Control to a Sonar-Based Obstacle Avoidance Mobile Robot, Proceedings of the 2000 IEEE International Conference on Control Applications Anchorage, Alaska, USA, 425 – 430 p.

Ulrich I. and Borenstein J., May 1998, VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots, Robotics and Automation, IEEE International Conference, Vol. 2, 1572 – 1577 p.

VxWorks, 2014, <http://www.windriver.com/products/vxworks/>.

KAYNAKLAR DİZİNİ (devam)

Wang F.Y., Konstantinos J. K., Tsolkas A. and N. S. George, August 1991, Petri-Net Coordination Model for an Intelligent Mobile Robot, IEEE Transactions On Systems, Man, And Cybernatics, Vol. 21, No. 4, 777 – 789 p.

Weigel T., Gutmann J. S., Dietl M., Kleiner A. and Nebel B., October 2002, CS Freiburg: Coordinating Robots for Successful Soccer Playing, IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, 685 – 699 p.

Yazıcı A., Özgüner U., Fu L., Kasım 2007, OSU – ACT Otonom Otomobili için Şehir İçi Yol Planlaması, 12 Elektrik – Elektronik Bilgisayar Biyomedikal Mühendisliği Ulusal Kongresi, Anemon Otel, Eskişehir.

Yayan U., 2012, Yüksek Lisans Tezi, İç Ortamlar İçin Konumlandırma Sistemi Geliştirilmesi, ESOGU Fen Bilimleri Enstitüsü.

Yücel H., Edizkan R., 2012, Yazıcı A., Yayan U., İç Ortamlar İçin Konumlandırma Sistemi (İÇKON)’nde Kodlayıcı Desteği ile Performans İyileştirme, Otomatik Kontrol Ulusal Toplantısı (TOK’12) Bildiriler Kitabı, Niğde, 546 – 551 s.