

Çoktörel Gezgin Robot Gruplarında
Müzayede Yöntemi İle Görev Dağıtımı

Uğur Gürel

DOKTORA TEZİ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Şubat 2014

Auction-Based Task Allocation
for Heterogeneous Mobile Robot Groups

Uğur Gürel

DOCTORAL DISSERTATION

Department of Electrical Electronics Engineering

February 2014

Çoktörel Gezgin Robot Gruplarında
Müzayede Yöntemi İle Görev Dağıtımı

Uğur Gürel

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik Elektronik Mühendisliği Anabilim Dalı
Kontrol ve Kumanda Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Osman Parlaktuna

Şubat 2014

ONAY

Elektrik Elektronik Mühendisliđi Anabilim Dalı Doktora öđrencisi Uđur Gürel'in DOKTORA tezi olarak hazırladıđı "Çoktürel Gezgin Robot Gruplarında Müzayede Yöntemi İle Görev Dađıtımı" başlıklı bu çalıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek kabul edilmiřtir.

Danıřman : Prof. Dr. Osman Parlaktuna

İkinci Danıřman :-

Doktora Tez Savunma Jürisi:

Üye: Prof. Dr. Osman Parlaktuna (Danıřman)

Üye: Yrd. Doç. Dr. Nihat Adar

Üye: Yrd. Doç. Dr. Selçuk Canbek

Üye: Yrd. Doç. Dr. Ahmet Yazıcı

Üye: Yrd. Doç. Dr. Muammer Akçay

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

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Günümüzde market tabanlı görev paylaşırma sistemleri giderek ilgi çekmektedir. Bu çalışmada, gezgin robot gruplarında görev paylaşırma problemi için market tabanlı bir yaklaşım önerilmiştir. Ortamda bulunan görevlerin daha yüksek bir performans ile yapılmasını sağlamak amacı ile tek robot yerine çok robotlu grubun kullanılması düşünülmüş ve çok robotlu bir sistem önerilmiştir. Ortamda birden çok robot bulunduğundan, bu robotların birbirleri ile haberleşebilmeleri sağlanmış ve bu haberleşme ortamı üzerinden müzayede yöntemini kullanarak görevler paylaşırılmıştır. Bu görev paylaşırma yaklaşımının yanı sıra, görevler paylaşıldıktan sonra robotların rotalarında birbirleriyle çakışmayı önleyici bir yapı da önerilmiştir. Ortamda bulunan her robot görevleri dağıtmak için müzayede düzenleyebilmektedir bu nedenle, çalışmadaki görev dağıtımı merkezî değildir. Burada birden fazla robot kullanıldığından, robotlardan biri bozulsa dahi sistem kalan robotlar ile çalışmaya devam eder. Bu çalışmadaki yaklaşımda kullanılan alt yordamlar erkin tabanlı olarak yazılmıştır. Benzetim ortamında gerçekleştirilen örnek uygulamalar, önerilen mimarinin çok robotlu yapılarda etkinliğini göstermiştir.

Anahtar Kelimeler: Gezgin robotlar, erkin tabanlı sistemler, müzayede tabanlı görev atama, çakışmasız rota planlama, çoktürel robotlar.

SUMMARY

Market-based task allocation systems are increasingly drawing attention. In this study, a market based approach is recommended for the multi robot task allocation problem. Since, there is more than one robot in the environment, the communication of those robots is ensured, and the tasks are allocated by the use of auction method above this communication environment. Besides the task allocation approach, subsequent to task allocation, collusion preventive structure is offered for the robots routes'. Each robot in the environment can hold the auction process; therefore in this study task allocation is not centralized. Since more than one robot and agent is used in the proposed architecture, the system may continue to work with the remaining robots and agents, even if one of them corrupted. The sub-functions used in this study are programmed as agent based. The sample implementations conducted in the simulation environment to show the effectiveness of the suggested architecture in multi robot structures.

Keywords: Mobile robots, agent-based systems, auction-based task allocation, collision free route planning, heterogeneous robots.

TEŞEKKÜR

Lisans yüksek lisans ve doktora eğitimim süresince benden sonsuz desteğini esirgemeyen, beni yönlendiren, cesaret veren en umutsuz anlarımda bile bana bir çıkış yolu gösteren çok değerli hocam, danışmanım Prof. Dr. Osman PARLAKTUNA'ya sonsuz teşekkür ederim.

Doktora tez çalışmalarım sırasında verdikleri faydalı öneriler ve yönlendirici eleştirileri ile sağladıkları katkılardan dolayı, çok değerli hocalarım Yrd. Doç. Dr. Nihat ADAR, Yrd. Doç. Dr. Selçuk CANBEK ve Yrd. Doç. Dr. Ahmet YAZICI'ya teşekkür ederim.

Tanıştığım ilk günden beri beni sevgisi ile koşulsuz destekleyen bana sürekli moral veren, güler yüzü bitmeyen enerjisi ile hayatıma mutluluk katan güzel eşim Dilek GÜREL'e. Her daim evimizin neşe kaynağı olan “baba artık sen çekil birazda ben yazayım” diyerek bana destek olan güzel oğlum Tuna GÜREL'e sonsuz teşekkür ederim.

Beni dünyaya getiren tüm yaşamım boyunca bana inanan sevgili annem Nuriye GÜREL ve babam Mehmet GÜREL'e teşekkür ederim.

Lisans ve yüksek çalışmalarım süresince, benim bu noktaya gelmemde katkısı olan tüm hocalarıma ve asistan arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xiv
1. GİRİŞ.....	1
1.1 Problemin Tanımı.....	4
1.2 Önerilen Yaklaşım	5
1.3 Çalışmanın Katkıları	6
2. GEZGİN ROBOT YAPILARI	8
2.1 Robot Kontrol Yapıları	8
2.1.1 Bilinçli / sıradüzensel kontrol	10
2.1.2 Tepkisel kontrol	13
2.1.3 Melez kontrol.....	19
2.2 Çok Robotlu Sistemler	20
2.2.1 Çok robotlu sistemlerin avantajları ve dezavantajları.....	21
2.2.2 Çok robotlu sistemlerde kontrol mimarilerinden beklenen özellikler	22
2.3 Yapısal Kontrol Yaklaşımları	23
2.3.1 Merkezi yapılar	24
2.3.2 Dağıtık yapılar	25
2.3.3 Market tabanlı yapılar	25
2.4 Robot-Görev Sınıflandırmaları	28
2.5 Robotların Rotalarındaki Çakışmaların Önlenmesi	29

İÇİNDEKİLER (devam)

3.	PROBLEMİN TANIMI VE ÖNERİLEN YAKLAŞIM	32
3.1	Çok Erkinli Yapılar	33
3.2	Önerilen Mimari ve Sistemde Bulunan Erkinler	35
3.2.1	Kullanıcı arayüzü erkini:	36
3.2.2	Veri tabanı erkini	37
3.2.3	Müzayede erkini	38
3.2.3.1	Teklif alma rolü	38
3.2.3.2	Teklif verme rolü	39
3.2.3.3	Maliyet hesaplama	40
3.2.3.4	Müzayede süreci	41
3.2.4	Konumlandırma erkini	42
3.2.5	Hareket erkini	44
3.2.6	Planlama erkini	45
3.2.7	Rota çakışma erkini	45
3.2.7.1	Rota çakışmasını zaman gecikmesi ekleyerek gidermek	47
3.2.7.2	Rota çakışmasını yeniden yol planlaması yaparak gidermek	48
4.	UYGULAMA	51
4.1	Pioneer P3-DX Robot Platformu	52
4.2	Benzetim Ortamı	54
4.3	Açık Erkin Mimarisi Ortamının Kurulması ve Test Edilmesi	55
4.4	Market Yapısının Oluşturulması	57
4.4.1	İş üretme erkini	58
4.4.2	İş alma erkini	59
4.4.3	Müzayede erkini	59

İÇİNDEKİLER (devam)

4.4.3.1	Müzayede düzenleme rolü.....	59
4.4.3.2	Müzayedeye teklif verme rolü.....	60
4.4.4	Hareket erkini	61
4.4.5	Veri tabanı erkini	61
4.4.6	Market yapısı uygulaması.....	61
4.5	Market Yapısında Görevlerin Uzaklığa ve Robotların Yeteneklerine Göre Dağıtılması.....	62
4.5.1	Görev ve robotların tanımı.....	63
4.5.2	Çalışmada kullanılan erkinler	63
4.5.3	Çalışmanın uygulanması ve sonuçları	65
4.6	Görevlerin Uzaklığa ve Öncelik Değerine Göre Dağıtılması	67
4.6.1	Görev ve robotların tanımı.....	68
4.6.2	Çalışmada kullanılan erkinler	68
4.6.3	Çalışmanın uygulanması ve sonuçları	71
4.7	Market Yapısı ile Görevlerin Rotalarda Çakışma Olmadan Dağıtılması.....	74
4.7.1	Görev ve robotların tanımı.....	74
4.7.2	Çalışmada kullanılan erkinler	78
4.7.2.1	Müzayede erkini	78
4.7.2.2	Rota çakışma erkini	80
4.7.3	Çalışmanın uygulanması ve sonuçları	81
	SONUÇ VE ÖNERİLER	85
	KAYNAKLAR DİZİNİ.....	88
	ÖZGEÇMİŞ	96

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
Şekil 1.1. Elle idare edilen manipülatör örneği	1
Şekil 2.1 Üç temel değerler dizisi a) Sıradüzensel değerler dizisi, b) Tepkisel değerler dizisi, c) Melez (tepkisel/bilinçli) değerler dizisi	8
Şekil 2.2 Shakey robotu	10
Şekil 2.3 Sıradüzensel yöntem akış şeması	10
Şekil 2.4 İç içe sıradüzensel denetleyici (Nested Hierarchical Controller NHC).....	12
Şekil 2.5 Gerçek-zamanlı denetim sistemi	13
Şekil 2.6 Tepkisel kontrol.....	13
Şekil 2.7 Davranış yapısı	14
Şekil 2.8 Serbest bırakıcı mekanizma ile davranış	14
Şekil 2.9 Şema teori ile kurbağanın beslenme davranışı	15
Şekil 2.10 Tepkisel değerler dizisinde duyucu organizasyonu.....	16
Şekil 2.11 Beş temel potansiyel alan a) düzgün dağılmış, b) dikey, c) çekici, d) itici, e) teğet halinde	17
Şekil 2.12 Potansiyel alan örneği.....	18
Şekil 2.13 Melez (hybrid) kontrol.....	19
Şekil 2.14 FIPA standardında sözleşme ağı iletişim kuralına ait sıralama diyagramı....	27
Şekil 2.15 Sabit engellerden sakınmak için kullanılacak yol	30
Şekil 3.1 Erkin ve çevrenin ilişkisi	33
Şekil 3.2 Erkin tabanlı mimarinin genel gösterimi	34
Şekil 3.3 OAA mimarisi	34
Şekil 3.4 Gezgin robotun içinde bulunan program birimler	36
Şekil 3.5 Kullanıcı arayüzü erkini	36
Şekil 3.6 Kullanıcı arayüzü erkini durum diyagramı.....	37
Şekil 3.7 Veri tabanı erkini durum diyagramı	37
Şekil 3.8 Müzayede erkini ve veri tabanı erkininin müzayede başlatma sıralama diyagramı	42
Şekil 3.9 Hareket erkini düğüm hesaplama akışı.....	44

<u>Şekil</u>	<u>Sayfa</u>
Şekil 3.10 Robot tabanlı çizge/zaman çizelgesi.....	46
Şekil 3.11 Rotada çakışma örneği.....	47
Şekil 3.12 Çakışma çözüldükten sonra çizge/zaman çizelgesi.....	47
Şekil 3.13 Rota çakışması çözüm diyagramı	50
Şekil 4.1 Pioneer P3-DX robot platformu	53
Şekil 4.2 Örnek dünya haritası.....	54
Şekil 4.3 Sıralama diyagramı.....	55
Şekil 4.4 MobileSim benzetim ortamı ve uygulanan rota	57
Şekil 4.5 Benzetim Ortamı.....	61
Şekil 4.6 Müzayede erkini teklif verme akış diyagramı	65
Şekil 4.7 Çalışmada kullanılan harita	66
Şekil 4.8 Müzayede erkini teklif verme akış diyagramı	70
Şekil 4.9 Çalışmada kullanılan harita	71
Şekil 4.10 Görevin öncelik değerini tanımlamada kullanılan olasılıklar.....	72
Şekil 4.11 Görev tanımlamada kullanılan olasılıklar.....	77
Şekil 4.12 Uygulamada kullanılan harita.....	81
Şekil 4.13 Robot görev kazanılan görev sayılarının grafiği	82

ÇİZELGELER DİZİNİ

Çizelge 2.1 Strips algoritması için uzaklık hareket çizelgesi	11
Çizelge 4.1 Bir görev ataması için örnek teklifler ve sonuç	62
Çizelge 4.3 Uygulamanın sonuçları	67
Çizelge 4.4 Robot görev çizelgesi (✓ robot görevi yapabilir, ✗ robot görevi yapamaz)	68
Çizelge 4.5 Uygulamanın sonuçları	73
Çizelge 4.6 Görevler ve gerekli olan yetenek/algılayıcı	75
Çizelge 4.7 Robotlara ait yetenekler	76
Çizelge 4.8 Robotlar ve gerçekleştirebilecekleri görevler	76
Çizelge 4.9 Önerilen üç yöntem için parametre değerleri ve koşulları	80
Çizelge 4.10 Görevlerin dağılımları	82
Çizelge 4.11 Önerilen üç yöntemde robotların kazandığı görev sayıları ve oranları	82
Çizelge 4.12 Önerilen üç yöntem için rotalardaki çakışmalar dikkate alınmadan işlerin bitiş süreleri	83
Çizelge 4.13 Önerilen üç yöntem için rotalardaki çakışmalar dikkate alındığında işlerin bitiş süreleri ve çakışma sayıları	83

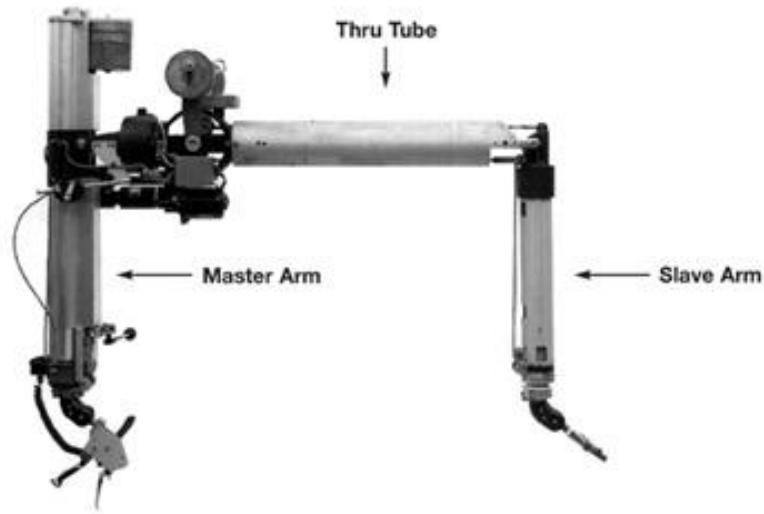
SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
R_i	i indisli robot
n	Çok robotlu sistemde bulunan robot sayısı
E_{ij}^R	i indisli robota ait j indisli erkin
ρ	Çok robotlu sistemde bulunan robotların sahip olduğu erkin sayısı

1. GİRİŞ

Tarihte “robot” kelimesi ilk defa 1921 yılında Çek yazar Karel Čapek tarafından Rossum’un Evrensel Robotları (Rossum’s Universal Robots) isimli eserinde ortaya atılmıştır (Čapek, 1921). Eserde görünmez bir güç olan Rossum tarafından yaratılan yapay insanlar robot olarak tanımlanmışlardır. Rossum’un eserinde robotlar insanlar tarafından her türlü işte köle olarak kullanılmak üzere tanımlanmıştır.

Robot kelimesi günümüzde kullanılan anlamı ile, 1979 yılında Amerika Robot Enstitüsü tarafından çeşitli görevleri gerçekleştirmek için programlanmış değişik/çeşitli hareketlerle malzeme, parça, alet veya özel cihazları taşımak amacıyla tasarlanmış çok işlevli, tekrar programlanabilir düzenek şeklinde tanımlanmıştır (Deb, 2009). Robotlar insan için tehlikeli olan ya da insanların yapmasının mümkün olmadığı görevleri başarı ile gerçekleştirebilirler. Şekil 1.1’de görünen MSM-8 ele idare edilebilen manipülatör, Merkezi Araştırma Laboratuvarı (Central Research Laboratories) tarafından etrafı tamamen ayrılmış (izole) olan bir odada bulunan nükleer maddelerin kullanıcı tarafından bir yerden bir yere taşımayı yapan kişiye zarar vermeden taşınması için geliştirilmiştir (Merkezi Araştırma Laboratuvarı, 1945).



Şekil 1.1. Elle idare edilen manipülatör örneği

1960'lı yıllara gelindiğinde özellikle otomotiv endüstrisinde endüstriyel robotlar sıklıkla kullanılmaya başlanmıştır. Endüstriyel robotlar aynı hareketi tekrar ederek paketleme, kaynak yapma, bir ürünü bir yerden bir yere taşıma gibi işlerin yapılabilmesi için herhangi bir zekâ gerektirmeyen belli görevleri yapabilen, sabit bir konumda duran robotlardır. Ancak 1970 yılların başlarına gelindiğinde gerek gelişen bilgisayar işlemcileri gerekse gelişen kontrol dizisi (paradigma) yöntemleri ile çevresinden topladığı bilgileri kullanarak hareket ve davranış planlarını kendisi yapabilen otonom gezgin robotlar ortaya çıkmaya başlamıştır. Gezgin robotlar günümüzde birçok alanda kullanılmaktadırlar: tıp alanında hastalara ilaç, yemek gibi temel ihtiyaçların taşınmasında gezgin robotlar kullanılmaktadır (Evans, 1994). Ayrıca insan sağlığı için zararlı olan zehirli, kimyasal veya nükleer atık gibi maddelerin temizlenmesi sırasında ya da yüksek gerilim hatlarının bakımı sırasında kullanılırlar (Cox, 2002). Robotlar mayın temizleme, bomba imha, güvenlik amaçlı gözlem, belirli bir noktanın keşfi ve fotoğraflarının çekilmesi ya da belirli bir yere yük taşımak için kullanılabilir (Zheng, 2009). İnsanoğlunun ulaşmasının mümkün olmadığı kadar derin yerlerde batık gemi arama, enkaz kurtarma veya deniz altı kablo döşeme ve bakımı işlerinde kullanılır (Yoerger et al., 2000). Ayrıca robotlar deprem, sel yangın gibi doğal ya da doğal olmayan afet durumlarında insanları bulmak ve eğer mümkünse kurtarmak için kullanılabilirler (Kibler ve Raskovic, 2012).

Yukarıda bahsi geçen alanlarda kullanılması düşünülen robotların bir ya da birden fazla insan kullanılarak uzaktan kontrol edilmesi değil, kendi zekâsını kullanarak işleri otonom olarak yapması beklenir. Dolayısıyla çalışmada gezgin robot denildiğinde algılayıcılar ile ortamı algılayan, algıladığı verilerden bilgi üreten, ürettiği bilgileri kullanarak aldığı görevleri kendi başına yapabilen akıllı gezgin robot akla gelmelidir.

Daha önceki çalışmalar ya tek robotlu sistemleri içeriyordu ya da içerisinde robotik bir bileşen içermeyen dağıtık problem çözme sistemleri ile ilgiliydi (Hur, 2007). Ancak gelişen teknoloji ile birlikte araştırmacılar aynı problemleri çok robotlu sistemler ile cevap aramaya başladılar. Çok robotlu çalışmalar 1990'ların ortalarından itibaren araştırmacılar tarafından oldukça ilgi çekmeye başlamıştır. Çok robotlu sistemlerin tek

robotlu sistemlere göre birçok avantajı vardır bu avantajlar aşağıda listelenmiştir (Jones ve Mataric', 2005).

- Tek bir adet pahalı ve gelişmiş bir robot yerine ucuz birçok robot kullanılarak sistemin toplam maliyeti düşürülebilir.
- Çok robot kullanmak sistemi daha esnek ve gürbüz yapabilir. Çünkü robotlar birbirlerinin yedeği gibi davranır ve bir robot devre dışı kaldığında diğer robotlar o robotun işini yapabilir.
- Bazı görevlerin çok robot tarafından gerçekleştirilmesi tek robota göre daha hızlı ve daha verimli olabilir, çünkü robotlar işleri kendi aralarında küçük parçalara ayırarak görevi daha hızlı bir biçimde gerçekleştirebilirler.
- Bazı görevleri gerçekleştirebilmek için mutlaka birden fazla robota ihtiyaç vardır.

Ancak çok robotlu sistemlerin avantajlarının yanı sıra dezavantajları da vardır. Birbirlerinin amaçlarına karşı iş yapmaya çalışan kötü tasarlanmış çok robotlu sistem yarardan çok zarara sebep olabilir. Böyle kötü tasarlanmış çok robotlu bir yapının tek robotlu bir sistemde karşılaşılan sorunların robot sayısı ile çarpımından çok daha fazla problem ortaya çıkarması olasıdır. Sonuç olarak belirli bir problemi çok robot kullanarak çözmek için tek robotta kullanılan yapının çok robota ölçeklendirilmesi yeterli olmayacaktır (Jones ve Mataric', 2005).

Literatürde ilgi çeken bir diğer konu da gezgin robotlarda görev dağıtımını problemidir. Robot sistemlerinde görev dağıtımının temelleri 1980 yılında Smith tarafından ortaya konulan Sözleşme Ağı İletişim Kuralı (Contract Net Protocol) ile atılmıştır (Smith, 1980). Sözleşmeli ağ protokolü araştırmacılar tarafından daha sonra oldukça kullanılmış ve geliştirilmiştir (Malone et al., 1988; Sandholm 1993; Golfarelli 1997). Çok robotlu sistemlere olan ihtiyaçla birlikte çok robotlu sistemlerde koordinasyonu ve görev dağıtımını sağlamak için çalışmalar ortaya konulmuştur. Parker tarafından 1998 yılında ortaya konulan ALLIANCE isimli mimari çok türel robot takımları için başarısızlık durumlarında görevleri yeniden dinamik olarak atayabilen davranış tabanlı bir mimaridir. Mimaride uygun hareketlerin seçimi motivasyon kavramı üzerine kurulmuştur. Motivasyon kavramı ise sabırsızlık ve

kabullenme ile tanımlanmıştır. Sabırsızlık kavramı kendisi haricinde diğer robotların başarısızlık durumlarında ortaya çıkarken, kabullenme kavramı robotun aldığı bir görevi kendisinin başaramaması durumlarında ortaya çıkar. ALLIANCE mimarisi içerisinde bulunan robotlar motivasyon kavramını kullanarak hem kendi iç hem de diğer robotların başarısızlık durumlarını kolladıklarından, herhangi bir başarısızlık durumunda iş başka bir robot tarafından alınır. Bu durumdan dolayı ALLIANCE mimarisi robot başarısızlıklarına karşı dayanıklıdır (Parker, 1998). Werger ve Matarić tarafından ortaya konulan BLE (Broadcast of Local Eligibility) göreve en uygun robot tarafından görevin alınması esasına dayalı davranış tabanlı bir mimaridir. BLE mimarisi ortamda bulunan değişikliklere oldukça hızlı tepki verebilmektedir (Werger ve Matarić, 2000). Görev dağıtımı konusunda market tabanlı bir mimari Stentz ve Diaz tarafından 1999 yılında ortaya konmuştur. Stentz ve Diaz çalışmalarında ilk-fiyat (first price) müzayede yöntemini kullanmışlardır (Stentz ve Diaz, 1999). Gerkey ve Matarić “MURDOCH” adı verilen bir mimari ortaya koymuşlardır. Önerilen mimari hataya dayanıklı bir yaklaşımdır. Çalışma, yayınl/abone ol (publish/subscribe) mesajlaşma yapısı ve müzayede yapısını içerisinde barındırmaktadır. İçerisinde barındırdığı bu yöntemler ile görevleri hataya dayanıklı bir biçimde dağıtık olarak dağıtmaktadır (Gerkey ve Matarić, 2000). Gerkey ve Mataric (2002), çalışmalarında görevler müzayede yöntemini kullanarak robotlar arasında paylaşılmıştır. Çalışmada görevlere tüm robotlar teklif verebilmektedir, müzayede sonrası görevi kazanan robot görevi yapmaktadır. Çalışmada robotlar yayınl/abone ol (publish/subscribe) haberleşme yöntemini kullanmışlardır.

1.1 Problemin Tanımı

Çalışmada çözüm bulunmaya çalışılan problem, daha önceden belirlenmiş bir harita üzerinde rasgele konumlara yerleştirilmiş olan çoktörel robot grubuna ortamdaki işleri müzayede yöntemi ile robotlara en uygun şekilde paylaşmak ve görevlerin paylaşılması sonucunda ortaya çıkabilecek olan robotların rotalarındaki çakışmaları en aza indirecek şekilde yeniden planlama yapmaktır.

Çok robotlu heterojen robot gruplarında görevleri paylaşırabilmek için öncelikle robotların birbirleri ile haberleşebileceği bir haberleşme ortamı ve birbirlerini anlayabilecekleri bir haberleşme protokolünün olduğu bir mimari olması gereklidir. Çalışma kapsamında robotların çalışacakları yazılım mimarisinin bazı özelliklere sahip olması gerekmektedir. Geliştirilecek mimari sistemin daha gürbüz olması için dağıtık olarak karar vermeyi desteklemelidir, robotlar kolaylıkla sisteme dâhil olabilmelidirler ve sistemden herhangi bir şekilde çıktıklarında kalan robotlar ile sistem çalışmaya devam edebilmelidir.

Çalışmada, görevler gezgin robot kaynaklarını en iyi şekilde kullanacak şekilde dağıtılacaktır. İşler robotlara dağıtılırken hem çoktörel robot grubunu oluşturan robotların yetenek iş uyumu göz önünde tutulacak hem de işlere ait öncelik ve robota olan uzaklık yakınlık değerleri de görev dağıtımını sırasında hesaba katılacaktır.

Görevler robotlara dağıtıldıktan sonra her robot kendi aldığı görevleri yapmak için kendi rotasını oluşturur. Bu oluşturulan rota bazı zamanlarda diğer robotların oluşturduğu rotalar ile çakışabilmektedir. Çalışmanın bir diğer ana amacı, robotların rotalarında olan çakışmaları en aza indirmek, olabiliyorsa tüm çakışmaları ortadan kaldırmaktır.

1.2 Önerilen Yaklaşım

Yukarıda açıklanan probleme çözüm olarak, öncelikle robotların birbirleri ile haberleşmesi, görevlerin paylaşılması ve robotun kendi içerisinde çalıştıracağı birçok yazılımın ortak çalışması için erkin tabanlı bir mimari önerilmiştir. Erkin tabanlı yapılar kullanıcıya hem yazılım geliştirme sırasında hem de mevcut yazılıma yeni eklentiler ekleme sırasında oldukça kolaylık sağlamaktadır. Yazılım geliştirme sırasındaki avantajlarının yanı sıra erkin tabanlı mimariler, mimaride oluşabilecek hatalara karşı oldukça dayanıklıdır. Mimari içerisinde bulunan her robot kendi içinde çeşitli görevleri gerçekleştirmek için özelleşmiş erkinleri barındırır. Her bir erkinin

kendine özgü bir görevi vardır ve erkinler özel olarak kodlanmış cümleler ile haberleşip bilgi paylaşımında bulunabilirler.

Önerilen yapı erkin tabanlı olduğu için robotun içerisindeki bir erkin bir sebepten ötürü çalışmayı durdurur ise robot kalan erkinler ile çalışmaya devam edebilir. Bu özellik robotların gürbüzlüğünü arttıran bir durumdur. Çalışmada görevler aşağıdaki kriterlere göre dağıtılacaktır.

- En yakın robot
- İşlerin öncelik değeri
- Robot / görev uyumu

Görevlerin dağıtılması sırasında tek çevrimli müzayede yöntemi kullanılacaktır. Sistemde bulunan her robot müzayedeyi düzenleyen robot olabilmektedir, bu sayede eğer herhangi bir robotun müzayede erkini görev yapamaz hale gelir ise diğer robotların müzayede erkinleri ortamdaki işleri dağıtabilirler.

Çalışmada tüm görevler dağıtıldıktan sonra robotların gideceği yollar üzerinde bir çakışma olup olmadığı kontrol edilir, eğer çakışma varsa bu çakışmalar en aza indirilecek şekilde robotların yolları üzerinde gerekli değişiklikler yapılır.

1.3 Çalışmanın Katkıları

Çalışma, çok robotlu yapılarda robot görev dağıtımını problemine ve erkin tabanlı gezgin robot kontrolü anlamında çeşitli katkılar sağlamıştır. Bu katkılar aşağıda listelenmiştir.

- *Modülerlik ve Gürbüzlük*: Çalışmanın geliştirilmesi sırasında modüler bir yapı kullanılmıştır. Sistemde ihtiyaç duyulan her bir belirli iş için bir erkin geliştirilmiş ve kendi içinde test edilmiştir. Test edilen her bir erkin daha sonra genel sisteme uygulanmış ve bir kez daha işlevleri ve diğer erkinler ile uyumu test edilmiştir. Çalışmada her bir erkinin görevi ayrı olduğu için karmaşık olan

süreçler daha basit alt süreçlere indirgenerek gerçekleştirilmiş ve oluşabilecek hataların önüne geçilmiştir. Sistemde birçok erkin ve birçok robotun bir arada çalışması sağlanmıştır. Çalışma zamanı sırasında robotlarda veya herhangi bir erkinde olabilecek bir arıza sadece ilgili birim ile sınırlandırılmıştır. Bu sayede yapı kalan robot/erkinler ile çalışabilir halde tutulmaya çalışılmıştır.

- *Görev dağıtımı:* Görevlerin dağıtımı sırasında görevlere verilecek olan maliyetler, robotun yapılacak olan işe uzaklığı, işin öncelik değeri ve robot görev uyumu derecesine göre hesaplanıp görevler bu üç kritere göre dağıtılmaktadır. Görevler dağıtıldıktan sonra robotların görevleri gerçekleştirmek için gidecekleri rotalar üzerindeki çakışmalar en aza indirilmeye çalışılmıştır, robotlar için çakışma olmayan yeni yollar oluşturulmuştur.

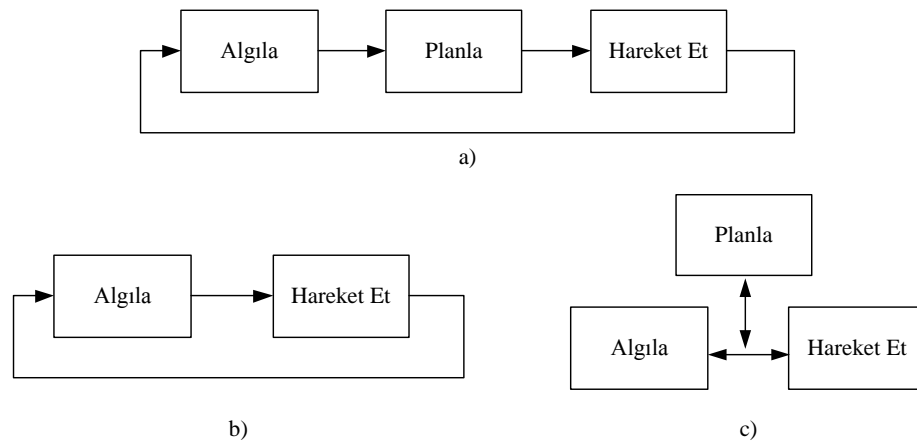
Tezin takip eden bölümlerinde şu konulara yer verilmiştir. Bölüm 2’de gezgin robotların içerdiği sistemler ve kontrol mimarilerinin yanı sıra robotların rotalarındaki çakışmaları önlemek için gerekli yapılar anlatılmıştır. Bölüm 3’de problemin geniş bir tanıtımı yapılmış ve ilgili problem için önerilen yaklaşım anlatılmıştır. Bölüm 4’de önerilen yaklaşımın etkinliğini göstermek için yapılan uygulamalar ve sonuçları gösterilmiştir. Son olarak Sonuç ve Öneriler’de yapılan çalışma ile ilgili sonuçlar tartışılmıştır.

2. GEZGİN ROBOT YAPILARI

Çalışmada otonom gezgin robotlar kullanılacaktır. Otonom gezgin robot, bulunduğu ortamda doğrudan bir insan müdahalesi olmadan kendisine verilen görevleri yapabilen akıllı makine olarak tanımlanır (Gerkey, 2003). Gerkey'in tanımında geçen akıllı makine kavramı ise bulunduğu çevreden aldığı bilgiyi kendi amacı için kullanabilen makine olarak Arkin tarafından kitabında tanımlanmıştır (Arkin, 1998).

2.1 Robot Kontrol Yapıları

Robotik kavramı ortaya çıktığından beri robotları kontrol etmek için birçok kontrol yöntemi ve birçok mimari ortaya atılmıştır. Robot kontrol yapılarını anlamak için robot değerler dizisini (paradigmayı) tanımlamak gereklidir. Murphy değerler dizisini, bir grup soruna yaklaşımı tanımlayan varsayımlar ve/veya teknikler felsefesi olarak tanımlamıştır (Murphy, 2000). Robotik değerler dizisi içerisinde robotların yapabileceği fonksiyonlar temelde üç eylemi içerisindedir bunlar algı, plan ve harekettir. Robotik kontrol yapılarında kabul görmüş üç temel değer dizisi Şekil 2.1'de verilmiştir.



Şekil 2.1 Üç temel değerler dizisi a) Sıradüzensel değerler dizisi, b) Tepkisel değerler dizisi, c) Melez (tepkisel/bilinçli) değerler dizisi

Sıradüzensel değerler dizisi bilinen en eski değerler dizisidir. Sıradüzensel değerler dizisi 1967-1990 yılları arasında oldukça yaygın bir biçimde kullanılmıştır (Murphy, 2000). Sıradüzensel değerler dizisinde robot ilk önce çevresini algılar daha sonra algılamayı bırakarak planlama aşamasına geçer ve planlama aşamasından sonra da harekete geçer. Ancak bu üç safha işletilirken herhangi bir anda sadece bir safha etkin olduğundan hem yavaş hem de kesikli çalışan bir yöntemdir. Ayrıca gerçek dünya uygulamalarında değişen dünya modeline hızlı tepki veremediğinden çeşitli sorunlar ortaya çıkmaktadır. Yavaşlık problemini ortadan kaldırmak için tepkisel değerler dizisi geliştirilmiştir. Tepkisel değerler dizisi planlama safhasını ortadan kaldırdığı için hızlı çalışmaktadır. Robot ortamı algıladığı gibi gelen algıya bir hareket ile yanıt vermektedir. Oldukça hızlı çalışmasına rağmen planlama aşamasının olmaması bu mimarinin uygulama alanlarını sınırlamıştır. Tepkisel değerler dizisi, melez tepkisel /bilinçli kontrol sıra dizisi için bir temel oluşturmuştur. Melez tepkisel/bilinçli kontrol sıra dizisi her iki yönteminde iyi yanlarını içerisinde barındırır. Melez değerler dizisinde robot önce bir plan yapar ve yapılan plan tepkisel sıra dizisinde olduğu gibi işletilir.

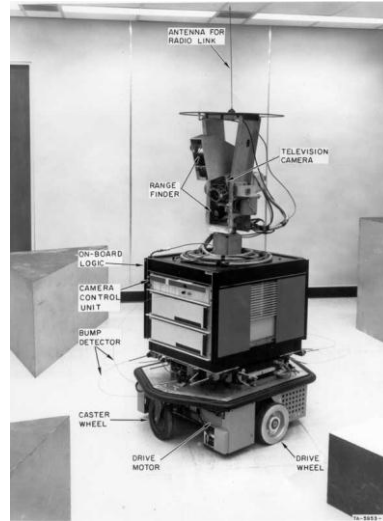
Yapılması düşünülen robotik uygulaması için yukarıda anlatılan değerler dizisinden uygun olanı seçmek uygulamayı gerçekleştirmek için yapılması gereken ilk adımdır. Yapılacak ikinci adım ise uygun kontrol yöntemini (mimari) seçmektir. Robotik uygulamalarında mimari kontrol sistemini organize bir şekilde çalışmasını sağlayan yapı olarak tanımlanabilir. Bir başka tanımda ise mimari, mimariyi oluşturan elemanların ne olacağını ve birbirleri ile ilişkilerinin nasıl olacağını tanımlayan yapı olarak tanımlar. Hangi mimarinin seçilmesi gerektiğine aşağıdaki değerlendirme ölçütlerine bakıp karar verebiliriz (Murphy, 2000).

- Modülerlik desteği: Yazılım mühendisliği açısından iyi mi?
- Uygunluk: Yapılacak olan uygulamaya uyumlu mu?
- Diğer alana kolay taşınabilirlik: Diğer robotla ya da diğer geliştirilecek olan uygulamalarla uyumlu çalışabilecek mi?

- Grbzlk: Uygulama alıřma zamanı sırasında evreden gelecek dıř etkilere dayanıklı mı?

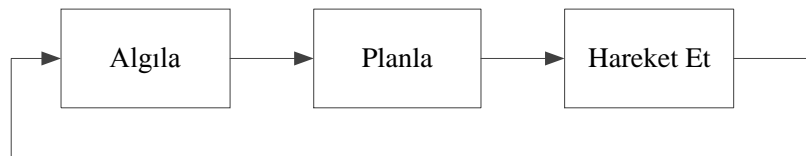
2.1.1 Bilinli / sıradzensel kontrol

Geleneksel yapay zekâda bilinen en eski yntemdir. Yntem 1967 yılındaki robotik uygulamalarına dayanmaktadır. İlk temelleri, Stanford Arařtırma Enstits (Stanford Research Institute) tarafından Őekil 2.2'de grnen Shakey isimli robot ile atılmıřtır (SRI uluslararası yapay zekâ merkezi, 1969).



Őekil 2.2 Shakey robotu

Sıradzensel yntem  temel adımın ardıřık ve aynı sırada alıřması prensibi zerine kurulmuřtur. Bu  temel adım *Algıla – Planla – Hareket Et* olarak isimlendirilir ve alıřma sırası Őekil 2.3'de grldg gibidir.



Őekil 2.3 Sıradzensel yntem akıř Őeması

Sıradüzensel yöntemde robot çevreyi algılar ve buna bağlı olarak çevrenin haritasını oluşturur. İkinci adımda ortamdan hiçbir yeni bilgi almayarak planını yapar ve son olarak da harekete geçer. Bu çevrim tamamlandıktan sonra robot yeniden dünyayı algılar, yeniden plan yapar ve harekete geçer. Bu tip yöntemlerin en büyük dezavantajı algılama sırasında kullanılacak olan dünya modelinin doğru, yeterince detaylı oluşturulmasının ve sürdürülebilirliğinin zor olmasıdır. Robot tüm planını ve hareketlerini algılama adımında algıladığı dünya modeline göre yapar, modelde olacak anlık değişiklikler plana ancak bir sonraki çevrimde dâhil olabilir. Bu durum, gerçek dünya uygulamalarında oldukça sorun yaratabilir. Çünkü robot gerçek dünyada belirli görevleri gerçekleştirirken planlarında anlık değişiklikler yapmak gerekli olabilir. Bir diğer dezavantaj da ilk adımdaki algılamanın oldukça uzun zaman almasıdır.

Sıradüzensel yöntem Shakey isimli robotta Strips adı verilen genel problem çözme algoritması olarak uygulanmıştır. Algoritmanın temeli bir hedefe ulaşmak için mevcut durumundan hedefe olan uzaklığı azaltacak şekilde bir sonraki hareketin seçilmesi esasına dayanır.

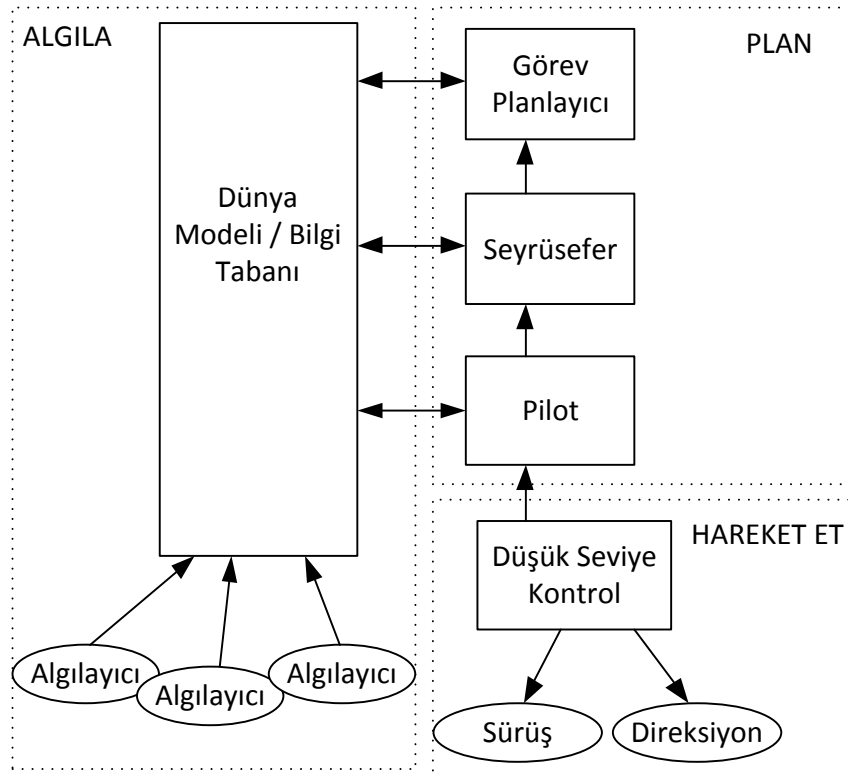
Çizelge 2.1 Strips algoritması için örnek uzaklık hareket çizelgesi

Uzaklık Farkı	Hareket	Ön Koşullar
$d > 200$	Uçarak git	
$100 < d < 200$	Trene bin	
$d < 100$	Araba kirala	Havaalanındaysan
	Kendi arabanla git	Evdeysen
$d < 1$	Yürü	

Çizelge 2.1’de görüldüğü gibi oluşturulan örnekte, uzaklık farkına bağlı seçilebilecek hareketler ve ön koşulları listelenmiştir. Robotun hedefine olan uzaklığına ve eğer var ise önkoşuluna bağlı olarak robotun yapacağı hareket seti değişecektir. Örneğin uzaklık 200 birimden büyük ise “uçarak git” komutu işletilir. Uzaklık 100 birimden küçük ve hava alanında olma önkoşulu aktif iken “araba kirala” komutu çalışır, bunun yanında uzaklık 100 birimden küçük ama “evde olma” ön koşulu aktif ise

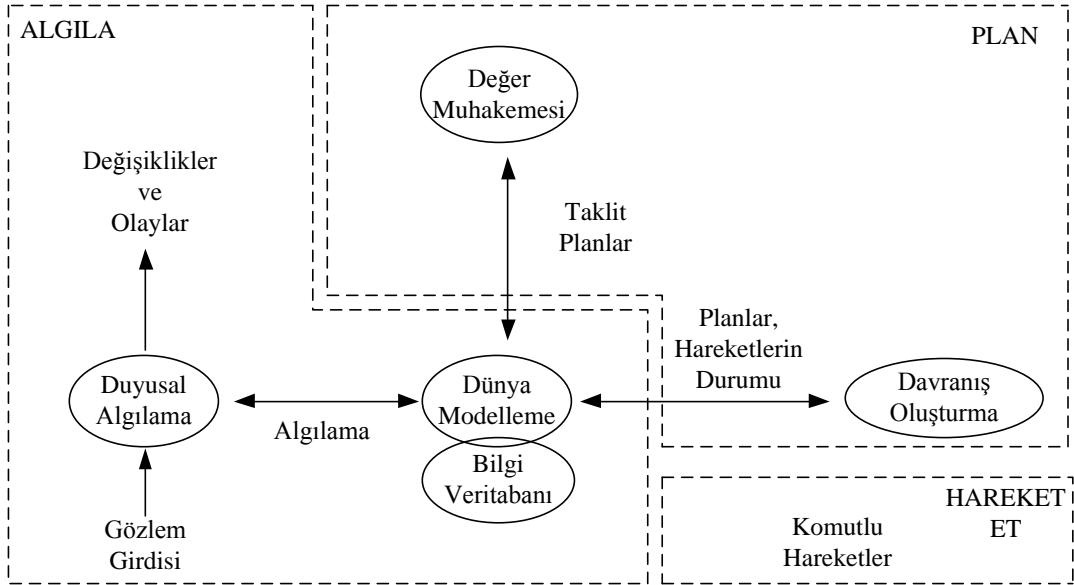
“kendi arabanla git” komutu aktif olur. Gerçek dünya uygulamaları için Sıradüzensel yöntem bazı durumlarda zayıf kalabilir bundan dolayı daha karmaşık durumları çözebilmesi için çeşitli denetleyici mimariler geliştirilmiştir.

Literatürde en çok bilinen iki denetleyici mimarisi vardır. Bunlardan ilki Meystel tarafından 1990 yılında geliştirilmiş olan İç İçe Sıradüzensel Denetleyicisi (Nested Hierarchical Controller NHC) ile kara kutu halinde olan *Algıla – Planla – Hareket Et* yapısını daha anlaşılır bir yapıya döndürmüştür (Meytsel, 1990). Oluşturulan denetleyici içinde bulunduğu dünya modelini içeren veri yapısının oluşturulması ile başlar. Bu aşama *algıla* aşamasına karşılık gelir. Daha sonraki aşama olan *planla* aşaması içerisinde üç adet alt yöntemi barındırır; bunlar görev planlayıcısı, seyrüsefer ve pilottur. En son aşama olan *Hareket Et* aşamasında ise motorların doğru çalışması için gerekli alt parametrelerin, komutların oluşturulmasını kapsar. İç içe sıradüzensel denetleyici Şekil 2.4’de gösterilmiştir.



Şekil 2.4 İç içe sıradüzensel denetleyici (Nested Hierarchical Controller NHC)

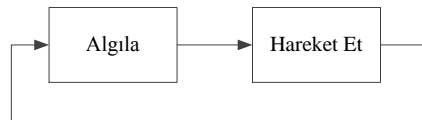
Literatürdeki diğer yöntem ise, Albus tarafından geliştirilen Gerçek-Zamanlı Denetim Sistemidir (Real-time Control System). Geliştirilen yöntem Birleşik Sıradüzen Denetleyicisine benzemektedir, temel fark duyuşsal algı program biriminin, duyucu ve veri birleştirmeye arasında bir ön işleme adımı içermesidir (Albus, 1993). Kurulan yapının Şekil 2.5’te görölmektedir.



Şekil 2.5 Gerçek-zamanlı denetim sistemi

2.1.2 Tepkisel kontrol

Tepkisel kontrol, sıradüzenel yönetime alternatif olarak geliştirilmiştir. Sıradüzenel yöntem ile en belirgin farkı planla aşamasının olmayışdır. Şekil 2.6’da yöntemin iki temel adımı olan *Algıla – Hareket Et* gösterilmiştir.



Şekil 2.6 Tepkisel kontrol

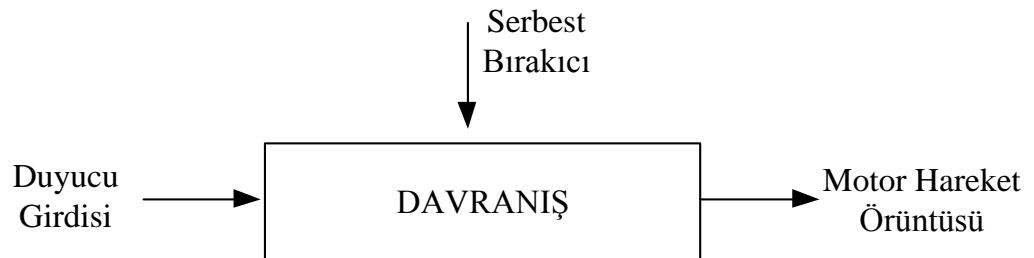
Geliştirilen yöntem planlama aşamasını içermediğinden oldukça hızlı çalışmaktadır. Yöntem doğadaki canlıların basit davranışlarını temel alır. Bu davranışlar, gerçekleştirilecek robot hareketlerinin temel yapıtaşları olarak algılanmalıdır. Tepkisel kontrolün temel yapıtaşı olan davranışların grafiksel gösterimi Şekil 2.7’de verilmiştir.



Şekil 2.7 Davranış yapısı

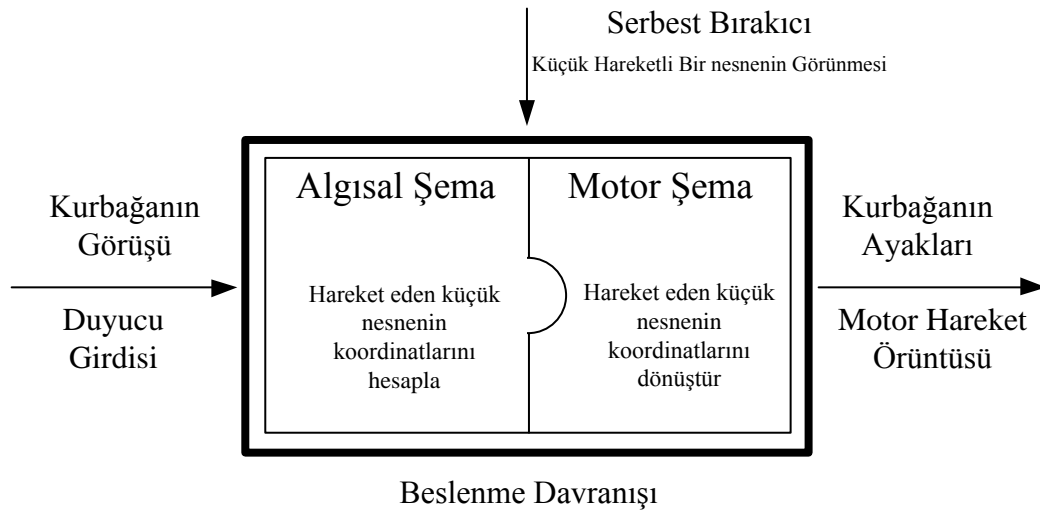
Davranış, belirli bir duyuşal girdiye karşı verilen bir dizi hareket örüntüsüdür. Örneğin bir kedi, bir köpeği görürse sırtını kambur yapar, tüylerini diker ve pençelerini çıkarır. Eğer bu yeterli olmaz ise tüm gücü ile kaçar. Kedi için, köpeği görme duyuş bilgisi kedinin birçok hareketi düşünmeden yapmasını sağlamıştır.

Ancak bazı durumlarda davranışın aktif hale geçmesi için sadece duyucu girdisi bilgisi yeterli olmaz, örneğin yemek gören bir insan sürekli her gördüğü yemeği yemez, yemek yeme davranışının aktif hale gelmesi için bir serbest bırakıcı yapıya ihtiyaç vardır. Yemek yeme davranışı için serbest bırakıcı aç olma durumudur bu yapı Şekil 2.8’de gösterilmiştir. Şema (schema) teorisi psikologlar tarafından 1900’lerin başından beri kullanılmakta olan bir yöntemdir.



Şekil 2.8 Serbest bırakıcı mekanizma ile davranış

Şema teorisinin ilk kez robotik alanına Micheal Arbib tarafından uygulanmıştır (Arbib, 1998). Daha sonra alanının tanınan kişileri tarafından da oldukça yoğun olarak kullanılmıştır. Yöntemin bu kadar ilgi çekmesinin sebebi, şemanın hem bilgiyi hem de yapılacak aksiyonları içerisinde barındırıyor olmasının yanında nesne tabanlı programlamaya (Object Orient Programming) da kolaylıkla uygulanabilir olmasıdır. Şema bir işin yapılabilmesi için gerekli olan genel bir şablondur. Bir şema sınıfı ilgili aktivitenin yapılabilmesi için gerekli olan tüm metot ve bilgiyi içerisinde barındırır. İki çeşit şema vardır algısal şema ve hareket şema. Algısal şema ortamın algılanması ile ilgili metotları içerisinde barındırırken motor şema ise aktivitenin nasıl yapılacağına dair metotları içerisinde barındırır. Şema teorisinde de serbest bırakıcı kavramı mevcuttur (Murphy, 2000).

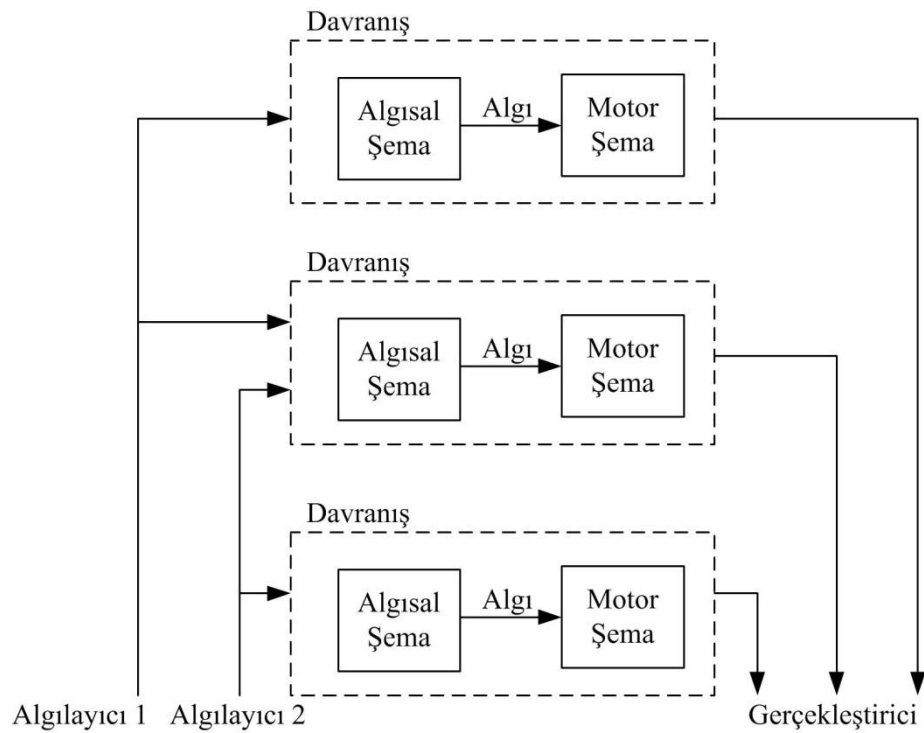


Şekil 2.9 Şema teori ile kurbağanın beslenme davranışı

Şekil 2.9'da bir kurbağanın bir sinek ile beslenme davranışı şema teorisi kullanılarak gösterilmiştir. Davranışın serbest kalabilmesi için öncelikle kurbağa hareketli küçük bir nesne görmelidir, bu hareketli küçük canlı nesne sinek ya da kurbağanın yiyebileceği bir kelebek olabilir. Ortamda böyle bir varlık tespit edildiğinde ilk olarak algısal şema çalışarak hareket eden küçük nesnenin koordinatları göz ile aşağı yukarı hesaplanır. Algısal şemada ortamın algılanması ve alınan algının hesaplanması işi vardır. Koordinatlar hesaplandıktan sonra motor şema canlılığın olduğu yönde doğru ne kadar dönülmesi gerektiğini ve uçan nesneyi yakalamak için ne kadar zıplanması

gerektiğini hesaplayarak kurbağanın bacaklarına emir verir ve kurbağanın ağız ile uçan nesnesi yakalamasını ve yemesini sağlar (Murphy, 2000).

Tepkisel kontrolde, aynı algılayıcılardan gelen bilgiler farklı davranışlar tarafından, farklı şekillerde işlenerek davranışların işletilmesi sırasında kullanılabilir. Öte yandan bir davranış farklı algılayıcılardan gelen algıları birleştirerek de kullanılabilir. Şekil 2.10'da tepkisel kontroldeki duyucu organizasyonu görülmektedir (Murphy, 2000).

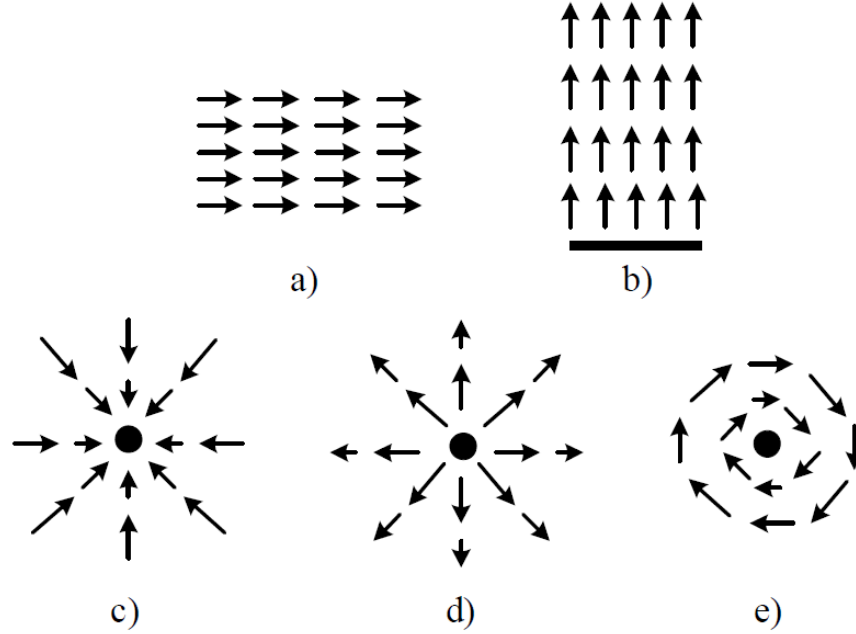


Şekil 2.10 Tepkisel değerler dizisinde duyucu organizasyonu

Rodney Brooks tarafından bulunan kapsama (*subsumption*) mimarisi tepkisel kontrolde en çok kullanılan yöntemler arasındadır (Brooks, 1986). Kapsama mimarisinde davranışlar kalıplar halinde robotun mikro işlemcisine yüklenmektedir. İhtiyaç anında herhangi bir planlamaya gerek kalmadan ilgili davranış çalışmaktadır. Kapsama mimarisinde davranışlar en üst seviyede en karmaşık davranışlar en altta en basit ve hayati davranışlar olacak şekilde oluşturulurlar. En basit davranışa örnek ortamda rasgele dolaşma olarak verilebilir, bu davranışın üzerine engellerden sakınma

davranışı konulabilir. Her zaman en alttaki davranış (etrafta dolaşma) çalışırken gerektiğinde onun üzerindeki davranış (engellerden sakınma) ya alt davranışı tamamen kapatır ya da alt davranışı kapsayarak daha karmaşık bir işlevi (engellerden sakınarak etrafta rastgele dolaşma) gerçekleştirir.

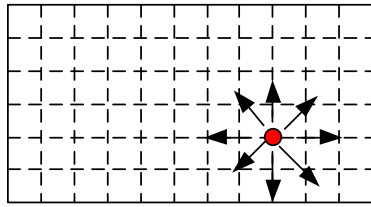
Tepkisel kontrolü içerisinde kullanan bir diğer yöntem ise potansiyel alan metodudur. Potansiyel alan yönteminde davranışlar vektörler tarafından gösterilmektedir. Birden fazla davranışın birleşmesi ortaya çıkan davranışlar vektörlerin birleşimi olarak ele alınmıştır. Temelde Şekil 2.11’de görüldüğü gibi ortamdaki davranışlar beş temel vektör ile ifade edilmektedir (Murphy, 2000).



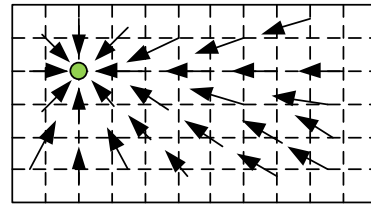
Şekil 2.11 Beş temel potansiyel alan a) düzgün dağılmış, b) dikey, c) çekici, d) itici, e) teğet halinde

Şekil 2.11a’da görünen düzgün dağılmış potansiyel alan, genellikle belirli bir yöne gitme davranışını göstermek için kullanılır. Şekil 2.11b’de görünen dik potansiyel alan ise robotun dik olarak konumlandığı durumları gösterir. Bu potansiyel alan daha çok robotun bir duvar gibi düzlemsel bir alandan uzak durması sağlar. Şekil 2.11c’de görünen çekici alan şeklindeki potansiyel alan, genellikle gezgin robot için ulaşılması gereken hedef noktanın gösteriminde kullanılır. Buradaki vektörlerin büyüklükleri

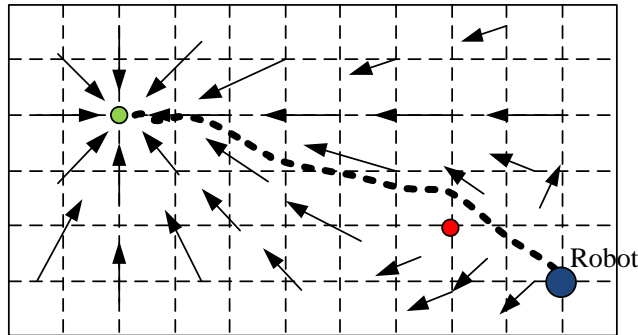
merkeze yaklařtıřa azalır. Uzakta olan bir robot için çekim oldukça fazla iken tam ortasında çekim çok çok azdır. Şekil 2.11d' de görülen itme potansiyel alanı ise çekim alanının tam tersi şeklinde çalışmaktadır. İtme değeri uzakta bulunan bir gezgin robot için düşük iken merkeze yaklařtıřa itme değeri artmaktadır. İtme potansiyel alan daha çok robotun sakınması istenilen engel vs. gibi yapıları göstermek için kullanılır. Son olarak Şekil 2.11 e'de gösterilen teğet halinde olan potansiyel alan bir nokta etrafında dönen vektörler ile ifade edilmektedir. Dönüş biçimi saat yönü veya saat yönünün tersi istikamete olabilir. Robotun bir engelin etrafından dolařması ya da o bölgede araştırma yapması sırasında kullanılır.



a) Engel noktası için itici potansiyel alan



b) Hedef noktası için çekim potansiyel alan



c) Her iki potansiyel alan birleřtiğinde robotun hareketi

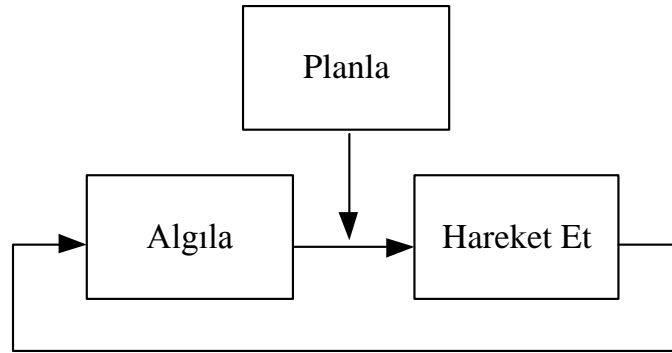
Şekil 2.12 Potansiyel alan örneđi

Şekil 2.12'de bir robotun potansiyel alan yöntemine göre bir robotun engelden sakınarak hedefe gitme davranışı gösterilmiştir. Şekil 2.12a'da engel noktası için potansiyel alan çizgileri verilmiştir Şekil 2.12b'de ise hedef noktası için potansiyel alan çizgileri verilmiştir. Bu iki şeklin bir arada birleřtirilerek gösterilmesi sonucunda Şekil 2.12c'de görünen potansiyel alan çizgileri olmuştur, robot başlangıç noktasından

(mavi nokta) hedef noktası olan yeşil noktaya gitmeye çalışıldığında potansiyel alanların birbirlerine etkilerinden dolayı Şekil 2.12c’deki gibi bir rota izlemektedir.

2.1.3 Melez kontrol

Geliştirilen melez yapı, hem tepkisel hem de bilinçli kontrol yöntemlerinin iyi yönlerini içerisinde barındıran bir yöntemdir. Günümüzde de araştırmacılar salt bilinçli ya da salt tepkisel yapılar yerine melez yapıları tercih etmektedirler. Yöntem içerisinde barındırdığı bilinçli kontrol yapıları ile ortamdaki bilginin en iyi şekilde soyutlanmasını, bir plan dâhilinde hareket etme gibi yetenekleri kullanırken aynı zamanda da içinde barındırdığı reaktif bileşen ile de gürbüzlük, değişen dünyaya hızlı tepki verebilme gibi yetenekleri içerisinde barındırır. Melez kontrol yapısı Şekil 2.13’de gösterilmiştir.



Şekil 2.13 Melez (hybrid) kontrol

Melez yapının altında yatan fikir sisteminin Planla kısmını ayrı *Algıla – Hareket Et* kısımlarının paralel olarak çalıştırmasında yatmaktadır. Melez kontrol yapılarında robot ilk önce mevcut dünya modeline göre görevi nasıl yapacağına dair plan yapar ve görevi gerçekleştirmek için gerekli davranışlar/alt metotları çalıştırır aynı anda da dünyayı algılamaya devam eder eğer dünya modelinde bir değişiklik olursa bununla ilgili yeniden planlama yapar. Murphy’ye göre melez yapılar genellikle aşağıdaki program birimlerini içlerinde barındırırlar (Murphy, 2000):

- *Sıralayıcı (Sequencer)*, belirli bir görevi gerçekleştirmek için gereken davranışları, davranışların sıralarını ve davranışların çalışabilmesi için gerekli olan koşulları belirler.
- *Kaynak yöneticisi (Resource manager)*, robotun sahip olduğu kaynakları ilgili davranışlara atar.
- *Haritacı (Cartographer)*, robota gerekli olan harita, bilginin oluşturulması, kaydedilmesi güncellenmesi ile sorumludur. Dünya modeli ve bilgi gösterimlerini de bu erkin belirler.
- *Görev planlayıcı (Mission planner)*, insan ile etkileşime girerek dış dünyadan konutları alır ve görev planını oluşturur.
- *Performans gözlemleyici ve problem çözücü (Performance monitoring and problemsolving)*, verilen görevin yapılıp yapılmadığını, görevin yapılmasında bir ilerleme olup olmadığını denetler.

2.2 Çok Robotlu Sistemler

Tanım olarak, iki ya da daha fazla robottan oluşan takım ya da topluluklar çoklu robot yapıları olarak adlandırılır (Murphy, 2000). Literatürde çok robotlu yapılar ile ilgili, birçok çalışma mevcuttur. Daha düşük maliyetli birden fazla robot eğer iyi bir yapı ile koordine edilirse bir tane pahalı ve güçlü robotun yapabileceği işten daha fazla görevi daha ucuz maliyet ile başarı bir şekilde yapabilir. Örneğin ucuz maliyetli çok robota sahip sistemler, bir tane pahalı ve güçlü robotun tarayabileceğinden daha fazla alanı tarayabilir ve çok robotlu sistemlerin bakım maliyetleri tek ve pahalı robot sistemine göre daha düşük olabilir.

2.2.1 Çok robotlu sistemlerin avantajları ve dezavantajları

Bazı görevlerin yerine getirilmesinde çok robotlu yapılar tek robotlu yapılara göre kullanıcıya bazı avantajlar sağlayabilir. Bu avantajlar aşağıdaki gibi sıralanabilir (Özkan, 2007).

- Çok robotlu sistemler tek robotun yerine getiremeyeceği işleri başarı ile gerçekleştirebilir. Örneğin tek robot tarafından itilemeyecek kadar ağır bir kutu, iki ya da daha fazla robot tarafından başarı ile itilebilir.
- Çok robotlu sistemlerde her robot tarafından hızlı bir şekilde yapılamayan hesaplamalar, grup içerisinde bulunan işlem kabiliyeti yüksek robotlar ile yapılarak, düşük işlem güçlü robotlar ile paylaşılabilir. Ya da bir binanın taranmasında çok robot tarafından yapılan tarama işlemi çok daha kısa sürede bitebilir.
- Çok robotlu sistemlerde her robotta bulunmayan lazer, kamera gibi özel algılayıcılardan alınan bilgiler diğer robotlar ile paylaşılarak robot grubunun ortam hakkında daha iyi bilgiye sahip olması sağlanabilir.
- Çok robotlu gruplarda iletişim kapsama alanı robotların iletişim sırasında röle gibi kullanılması ile artırılabilir.
- Çok robotlu yapılarda genellikle güçlü özelliklere sahip ve pahalı tek bir robot yerine işlem gücü düşük ve ucuz maliyetli birden fazla robot kullanıldığından, robot grubundan herhangi bir robot bozulduğunda, bozulan robotu yerine koymak için çok büyük bir maliyet gerekmez.
- Çoklu robot grupları birden fazla robottan oluştuğundan bir robotun bozulması tüm grubu etkilemez, grup kalan yetenekler ile işleri gerçekleştirmeye devam eder. Bu yüzden tek robota göre çok robotlu sistemler daha gürbüzdür.

Ancak çok robotlu yapılar getirdiği avantajların yanı sıra bazı dezavantajları da içerlerinde barındırır. Arkin eserinde bu dezavantajları aşağıdaki gibi belirtmiştir (Arkin, 1998).

- *Müdahale*: Birden fazla robotun ortamda bulunması robotların birbirlerini fiziksel olarak engellemelerine, çarpışmalarına sebep olabilir.
- *İletişim Maliyeti ve Gürbüzlük*: Çok robotlu yapılarda robot sayısı arttıkça robotların birbirleri ile haberleşme maliyetleri hem yeni donanım gereksinimler hem de aynı anda birçok robotun iletişim halinde olabilmesi için yeni ek yazılımların geliştirilmesi gerekeceğinden artar.
- *Diğer Robotlarla Etkileşim*: Birden çok robotun olduğu bir ortamda eğer robotlar iyi bir şekilde koordine edilmezler ise robotlar birbirleri ile yardımlaşmak yerine birbirleri ile rekabet ederler.
- *Tüm Robot Sistemin Maliyeti*: Bazı görevleri yapmak için bir robot yerine birden fazla robot kullanmak maliyetleri arttırabilir.

2.2.2 Çok robotlu sistemlerde kontrol mimarilerinden beklenen özellikler

Çok robotlu sistemler tek robotlu sistemlerin yeterli olmadığı durumlarda problemleri çözmek için kullanılabilirler, ancak çok robotlu sistemlerin başarıya ulaşabilmesi için belli başlı özellikleri karşılaması gerekmektedir. Aşağıda bu özellikler sıralanmıştır (Özkan, 2007).

- *Gürbüzlük*: Robot grubu içerisinde bulunan robotlardan biri ya da birkaçı tamamen ya da kısmen işlevini yitirir ise robot grubu kalan robot ya da kalan yetenekler ile görevleri yerine getirmeye devam edebilmelidir.
- *Genişleyebilirlik*: Sisteme yeni yetenek eklemek veya var olan yeteneği sistemden çıkarmak oldukça kolay olmalıdır.
- *Ölçeklendirilebilirlik*: Sistem her sayıdaki robot için ölçeklendirilebilir olmalı, bazı uygulamalar tek robot gerektirirken bazı uygulamalarda birden fazla robot kullanımı kaçınılmazdır. Robot sayısı arttığında sistem haberleşme, görev paylaşma gibi artan robot sayısı ile artan yükü belirli bir sayıya kadar kaldırabilir olmalı.
- *Esneklik*: Farklı işletim sistemlerinde koşan ya da farklı amaçlar için yazılmış olan yazılımlar, donanımlar birbirleri ile uyumlu bir biçimde çalışabilir olmalı.

Farklı yapıların birbirleri ile kolayca haberleşebilmeleri için ortak bir dil kullanılmalı.

- *Güvenilirlik*: Sistemde bulunan her bir eleman güvenilir olmalı her zaman doğruyu söylemeli.
- *Çoktürelilik*: Sistem hem çoktüreli robotlarla hem de tektürel robot grupları ile çalışabilmeli.
- *Haberleşme*: Sistemde her bir yapı birbiri ile haberleşebilmeli. Haberleşmenin mümkün olmadığı durumlarda robot en son veriler ile çalışmaya devam etmeli.
- *Uygun Kaynak Kullanımı*: Sistemde bulunan elemanlar ortamda bulunan kaynakların en iyi şekilde kullanımı için seçimler yapmalıdırlar.
- *Görev Dağıtımı*: Sistemde bulunan işler robotlara ve robot grubuna göre en uygun şekilde dağıtılmalıdırlar.
- *Roller*: Sistemde birden fazla rol bulunabilir. Her robot gerektiğinde yetenekleri çerçevesinde mevcut rolünü bırakıp yeni bir role kolaylıkla geçebilmelidir.
- *Yeni Girdilere Tepki Verme*: Sistem yeni tanımlanan görevler, robotlar ve rollere dinamik olarak tepki verebilmelidir.
- *Dinamik Koşullara Hızlı Tepki Verme*: Sistemin yeniliklere tepkisi hızlı olmalıdır.
- *Uygulama*: Öngörülen sistem benzetim ortamında ve/veya gerçek dünya ortamlarında test edilmiş olması gerekmektedir.
- *En Uygun Tepkinin Verilmesi*: Sistem çalışma zamanı sırasında en uygun tepkiler vermelidir.

2.3 Yapısal Kontrol Yaklaşımları

Çok robotlu sistemler, karar mekanizmasının mimaride bulunma şekline göre merkezi veya dağıtık olmak üzere iki temel başlık altında toplanabilir. Market tabanlı yapılar içerisinde hem merkezi hem de dağıtık sistemlerin avantajlarını barındırırlar. Son yıllarda market tabanlı yapılar literatürde oldukça ilgi çekmeye başlamış ve kontrol yapıları arasında yerini almıştır (Dias, 2004).

2.3.1 Merkezi yapılar

Merkezi yapılarda, tüm robot grubunun davranışları lider olan tek bir erkin tarafından yönetilir (Dias ve Stenz, 2003). Merkezi yapıların en önemli avantajı, ortamda bulunan tüm robotlardan gelen bilginin merkezi karar vericide toplanması ve bu bilginin kullanılarak en uygun çözümün üretilmesidir. Ancak teoride geçerli olan bu durum, dinamik ortamlarda aşağıdaki nedenlerden dolayı iyi sonuçlar vermeyebilir. Bazı problemler; Çok dinamik ortamlardaki merkezi planlayıcı, ortamdaki tüm değişiklikleri kapsayacak kadar hızlı yeniden planlama yapamayabilir. Bazı problemler; (Dias ve Stenz, 2003)

- Ortamdaki robot sayısının artışı ile robotlar için birçok kısıtı olan karmaşık problemlerin en iyi çözümleri bulunamayabilir.
- Merkezi yapılarda robotlar sürekli merkezi karar verici ile iletişim halinde olmalıdırlar. Eğer iletişim kesilir ya da kısıtlanırsa merkezi karar verici ortamdan yeterince veri alamadığı için robot grubu için en iyi çözümü üretemez, değişen koşullara göre planı yenilenemeyebilir.
- Merkezi yapıların başarısı, merkezi karar vericinin performansına oldukça bağlıdır. Eğer merkezi karar verici çalışmaz duruma gelirse bütün grup çalışmaz hale gelebilir.

Yukarıda sıralanan sebeplerden dolayı merkezi yapılar için daha çok robot sayısının az olduğu gruplar ve ortam değişkenlerinin az değiştiği problemler için daha uygun olduğu düşünülmektedir (Dias ve Stenz, 2003).

2.3.2 Dağıtık yapılar

Dağıtık yapılar merkezi yapıların, dezavantajlarının ortadan kaldırılması için ortaya atılmış olan kontrol yapılarıdır. Merkezi yapılarda, merkezde bulunan robot işleri dağıtır, kararları verir ve tüm grubu belirli bir amaç için yönetir. Dağıtık yapılarda ise, her robot kendi kararını verir ve diğerlerinden bağımsız bir biçimde hareket edebilir. Dağıtık yapılarda herkes kendi yaptığı/yapacağı işlerden sorumlu olduğundan robot grubunda oluşacak bir aksaklığa karşı dağıtık yapı, merkezi yapıya göre daha gürbüzdür. Çünkü robot grubu içerisinde bir robotun çalışmaz duruma geldiğinde sadece ilgili robotun yeteneği gruptan kaybolur, ancak robot grubu kalan yetenekler/robotlar ile verilen görevi yerine getirmeye devam edebilir.

Dağıtık yaklaşımların en temel dezavantajı, ilgili probleme çözüm üretirken robotlar sadece kendilerinin bilgilerini kullandığından ve takım ile ilgili bir bilgiye sahip olmadığından bazen yerel çözümlere (local minimum) yakınsayabilir. Ancak dağıtık yapılar özellikle robot sistemleri için de uygun olduğu düşünülmektedir (Jennings ve Kirkwood-Watts, 1998)

2.3.3 Market tabanlı yapılar

Market tabanlı yapılar içerisinde barındırdığı dağıtık ve merkezi yapılardan dolayı son yıllarda araştırmacıların oldukça ilgisini çekmiştir. Market tabanlı sistemler dağıtık ve merkezi yapıların iyi yönlerini bir araya getirmek için ortaya çıkmışlardır. Market tabanlı yapılarda robotlar tek başlarına plan yapabildikleri gibi, robot grubu olarak da belirli bir görevi gerçekleştirebilirler.

Market tabanlı görev dağıtım sistemlerinde görevin dağıtılması için ortamda müzayedeyi düzenleyen ve bu müzayedeye teklif verebilecek en az bir robotun

bulunması gerekir. Müzayedeye süreci, müzayedeye bir işin çıkarılması ile başlar, robotlar dağıtılacak olan iş için düzenlenen müzayedeye kendi maliyetlerini hesaplayarak gönderir. Müzayedeyi düzenleyen robot amaç fonksiyonuna en uygun olan teklifi seçerek tüm robotlara kazanan robotu bildirir. İşe en uygun teklifi veren robot dolayısıyla işi yapma hakkına da sahip olur ve kazandığı işi kendi iş listesine ekleyerek gerçekleştirir.

2.3.3.1 Sözleşme ağı iletişim kuralı

Birçok market tabanlı çalışmada, robotlar arasındaki etkileşim Smith tarafından 1980 yılında geliştirilen Sözleşme Ağı İletişim Kuralı (Contract Net Protocol) yöntemi kullanılarak gerçekleştirilir. Sözleşme ağı iletişim kuralı yöntemi erkinler arası etkileşimli problem çözme yöntemi olarak adlandırılır (Weiss, 2000). Yöntem, erkinler arasında iş dünyasında olduğu gibi mal veya hizmetlerin değiş tokuşunu yönetmek için kullanılan sözleşme mekanizmasından esinlenerek modellenmiştir. Sözleşme ağı iletişim kuralı yönteminde temel problem görevin yapılmasını isteyen bir yönetici ve bu görevi yapabilecek uygun yüklenici/yüklenicilerin bulunmasıdır.

Sözleşme ağı iletişim kuralında temel olarak iki rol bulunur bunlardan ilki işin yapılmasını isteyen yönetici diğer rol ise diğeri işi yapacak olan yüklenici rolüdür. Her iki rolünde kendine özgü yapması gereken iş kalemleri mevcuttur ve aşağıda listelenmişlerdir.

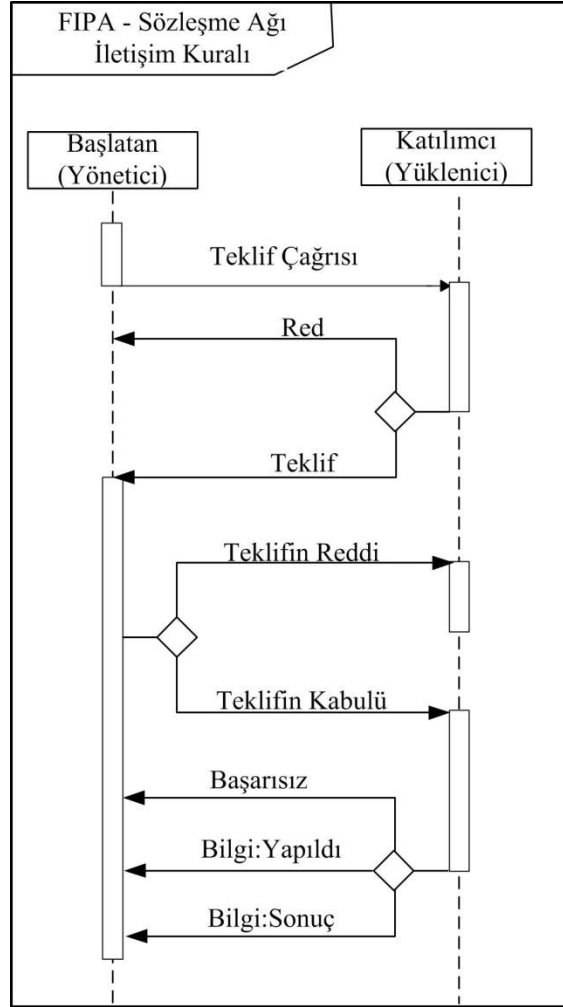
Yönetici Rolünün Görevleri:

- Yapılması istediği işi anons eder.
- Potansiyel yüklenicilerden gelen teklifleri kabul eder ve değerlendirir,
- En uygun teklifi gönderen potansiyel yükleniciye sözleşmeyi gönderir,
- Sonuçlar alınır ve birleştirilir.

Yüklenici Rolünün Görevleri:

- Görev anonslarını dinler,
- Gelen görev anonslarını yanıtlamak için yeteneklerini değerlendirir,
- Gelen görev anonslarını yanıtlar (Geri çevir / teklif ver),
- Teklifi kabul edilirse görevi gerçekleştirir,
- Sonuçları rapor eder.

Bu çalışmada da görev paylaşımı sırasında Sözleşme Ağı İletişim Kuralı kullanılacaktır. Akıllı fiziksel erkinler kuruluşu (Foundation for Intelligent Physical Agents, FIPA) tarafından belirlenen standartlara göre sözleşme ağı iletişim kuralına ait sıralama diyagramı Şekil 2.14’te verilmiştir (Akıllı fiziksel erkinler kuruluşu, 2002).



Şekil 2.14 FIPA standardında sözleşme ağı iletişim kuralına ait sıralama diyagramı

2.4 Robot-Görev Sınıflandırmaları

Gerkey ve Mataric (2004) çalışmalarında, çok robotlu sistemlerde görev paylaşırma problemi için inceleme ve sınıflandırma yapmışlardır. Çalışmalarda kullanılan görev kavramı oldukça çeşitli olabilmektedir. Çalışmalarda bazı görevler ayrıık olabileceđi gibi örneđin “x nesnesini A odasından B odasına taşı”, bazı görevler ise sürekli örneđin “A odasını gözlemler” olabilmektedir. Gerkey ve Mataric’ çalışmalarında herhangi bir görev ayırımı yapmadan genel anlamda çok robotlu sistemler için sınıflandırma yapmışlardır. Çalışmada çok robotlu sistemlerde görev paylaşırma problemlerinin bilinen eniyileme problemlerine benzerlikleri gösterilmiştir. Böylelikle, hangi görev paylaşırma probleminin hangi eniyileme yöntemleriyle çözülebileceđi belirlenmiştir (Kaleci, 2010). Bu kapsamda, çok robotlu sistemler için görev paylaşırma problemi üç temel eksen (robot, görev ve zaman) altında toplanmıştır.

- Robotların aynı anda yapabildikleri iş sayısı: Aynı anda tek iş yapabilen robotlar (Tek Görev) ve aynı anda birden fazla iş yapabilen robotlar (Çok Görev)
- İşlerin aynı anda yapılabildikleri robot sayısı: Aynı anda tek bir robot tarafından yapılabilen işler (Tek Robot) ve aynı anda birden fazla robot tarafından yapılması gereken işler (Çok Robot)
- Robotların geleceđe ait bilgilerinin durumu: Anlık Atama (AA), Zamana Yayılmış Atama (ZYA). Eğer robotlar, görevler ve ortam ile ilgili bilinen bilgi sadece o anı kapsıyor ve geleceđe dair hiçbir şey içermiyorsa anlık atama yapılır, ancak eđer gelecekte oluşacak görevler, ortam bilgisi, robotların durumları gibi bilgiler biliniyorsa zamana yayılmış atama yapılabilir.

Çalışmada, çoklu robot sistemlerinde görev atamaları yukarıda tanımlanan eksenlere bađlı olarak aşağıda listelenen sekiz farklı kategoriye ayırmışlardır.

- TR-TG-AA : Tek robot, tek görev, anlık atama.
- TR-TG-ZYA : Tek robot, tek görev, zamana yayılmış atama.
- ÇR-TG-AA : Çok robot, tek görev, anlık atama.
- ÇR-TG- ZYA : Çok robot, tek görev, zamana yayılmış atama.
- TR-ÇG-AA : Tek robot, çok görev, anlık atama.
- TR-ÇG-ZYA : Tek robot, çok görev, zamana yayılmış atama.
- ÇR-ÇG-AA : Çok robot, çok görev, anlık atama.
- ÇR-ÇG-ZYA : Çok robot, çok görev, zamana yayılmış atama.

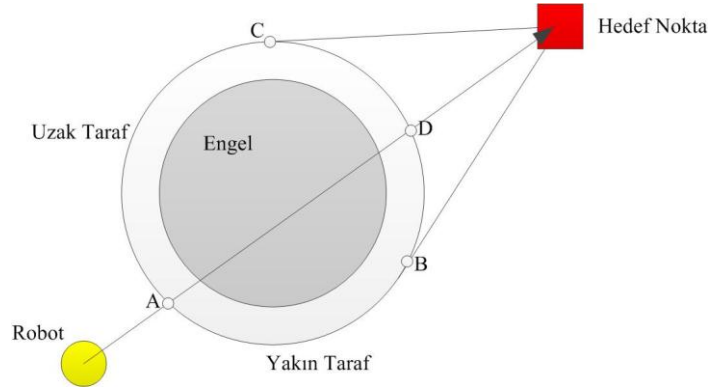
Gerkey ve Maratic çalışmalarında her bir durum tek tek ele alınmış, ilgili sınıflandırmadaki literatürdeki çalışmaları anlatılmış ve problem çözümü için en iyileme yöntemleri anlatılmıştır. Mevcut çalışmada görevler ÇR-ÇG-AA (çok robot, çok görev, anlık atama) şeklinde market tabanlı bir yöntem kullanılarak robotlara dağıtılacaktır.

2.5 Robotların Rotalarındaki Çakışmaların Önlenmesi

Çok robotlu yapılarda görevlerin dağıtılması oldukça önemli bir konudur. Görevlerin dağıtılması kadar önemli olan bir diğer konu ise robotların rotalarında oluşabilecek çakışmaları önlemektir. Robotların rotalarında olabilecek olan çakışmalar aynı anda sadece tek robot tarafından kullanılabilen yollarda oluşabilir. Robotların rotalarında oluşan çakışmalar robotların birbirlerini sonsuz süre ile kilitleme (deadlock) durumunda beklemelerine, görevlerin yapılmamasına, birbirleri ile fiziksel çarpışarak robotların zarar görmelerine hatta çalışamaz duruma gelmesine sebep olabilir. Bu gibi sakıncalardan dolayı robotların rotalarındaki çakışmalar mümkün olduğu kadar önlenmelidir. Robotlar yapacakları görevleri aldıktan sonra ilgili görevleri yapmak için her bir robot kendi yolunu oluşturur robotlar aynı zamanda aynı yolu kullanmak istediklerinde birbirlerini engellemeleri kaçınılmazdır.

Çarpışmayı önlemek için Cai ve arkadaşları (Cai et al., 2007) çalışmalarında, ortamda bulunan engelleri ve diğer robotları fark etmek için tüm yönlü (omnidirectional) kamera kullanmışlardır. Robotlar ortamda hareket ederlerken kamera

ile engelleri ve diğer robotları algılayarak çakışmaları tahmin etmişlerdir. Çalışmalarında sabit engeller ve hareket halinde olan robotlar ile olan çarpışmaları gidermek için iki yöntem önerilmiştir. Sabitler engeller ile oluşabilecek olan çarpışmalar teğet böcek algoritması (tangentbug algorithm) ile çözülmüştür.



Şekil 2.15 Sabit engellerden sakınmak için kullanılacak yol

Sabit engellerden sakınmak için, robotun bulunduğu noktadan ulaşmak istediği noktaya doğru sanal bir doğru çizilir ulaşılacak istenen noktadan engel içerecek şekilde sanal bir daire çizilir daha sonra hedef noktasından sanal çembere iki adet teğet çizilir hangi teğetin uzunluğu hedefe daha yakınsa o taraf seçilir. Örneğin Şekil 2.15’de robot hedefe ulaşmak için AB yayı seçilerek engel aşılmalıdır. Çalışmada diğer robotlarla olabilecek olan çakışmalar da tüm yönlü kamera ile tespit edilmiştir. Hareketli robotların rotalarındaki çakışmaları engellemek içinse olası çakışma fark edildiği anda öncelik değeri düşük olan robot durarak ya da hızını düşürerek öncelik değeri yüksek olan robota yol verme yöntemini kullanmışlardır. Bu sayede robotların rotalarındaki çakışmaların önüne geçmişlerdir.

Yuhong ve arkadaşları çalışmalarında denizde seyrüsefer yapan araçlar için rotalarındaki çakışmaları tahmin eden ve çakışmayı önleyen bir uzman sistem önermişlerdir. Gemilerin seyrüseferi sırasında oluşabilecek çakışmaları tahmin etmek için üç aşamalı bir yol önermişlerdir. İlk aşama geminin bir sonraki hareketinin ne olduğunun tahmin edilmesidir, daha sonra geminin rotasının diğer bir gemi rotası ile çakışmasının riskinin ve geminin ne kadar tehlikeli olduğunun tahmin edilmesidir, son aşamada ise geminin tipine ve hareketlerine göre çakışmadan nasıl kurtulacağını

tahmin edilmesidir. akışma noktası tahmin edildikten sonra akışmanın oluşmaması için daha az tehlikeli olan gemi rotasını deęiştirir (Yuhong et al., 2008).

3. PROBLEMİN TANIMI VE ÖNERİLEN YAKLAŞIM

Çalışmada, çok robotlu çöktürel robot grupları için kontrol mimarisi ve görevlerin dağıtılabilmesi için müzayede tabanlı ve merkezi olmayan (decentralized) bir görev atama yaklaşımı önerilmiştir. Öngörülen mimarinin merkezi olmaması, birden fazla farklı yazılımın paralel çalışacak olması ve birbirinden farklı donanımın birlikte kullanılacak olmasından dolayı erkin tabanlı mimari kullanılmıştır. Erkin tabanlı mimarinin aşağıda yazıldığı gibi birçok avantajları vardır (Özkan, 2007).

- *Gürbüzlük ve güvenilirlik:* Mimaride bulunan erkinlerden biri ya da birkaçı çalışmamaya başladığında sistem çalışan erkinlerin yetenekleri ile çalışmaya devam eder. Bu durum mimarinin gürbüzlüğünü arttıran bir durumdur.
- *Hızlanma ve etkinlik:* Sistem mevcut problemler üzerinden paralel ve asenkron hesaplama yapabilir.
- *Ölçeklenebilirlik ve esneklik:* Yeni erkinler sisteme hızlıca eklenebilir ve yeni yetenekler eklenen erkinler ile hızlıca kullanılabilir.
- *Geliştirme ve tekrar kullanılabilirlik:* İlgili erkinlerin modüler bir yapı ile yazılması hem hataların önüne geçilmesinde hem de yazılımda yapılacak olan bakımda kolaylık sağlar.

Erkin temelli mimariler, mobil robot uygulamaları, uçak bakımı, askeriye, tedarik zincir yönetimi, finansal yönetim gibi alan da kullanılmaktadır. (Ray et al., 2011; Kawamura et al. 2001, Onn et al. 1999; MacKenzie et al., 2012; Li, 2010; Hancock ve Lamont, 2011; Chaib-draa ve Müller 2006; Boer-Sorb'an, 2008)

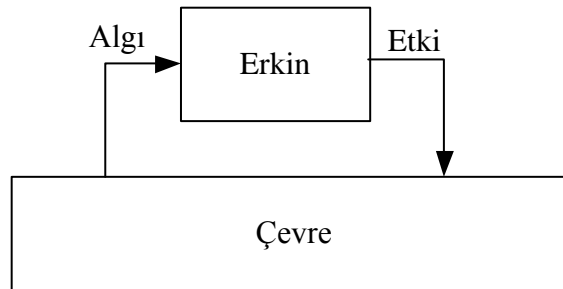
Çalışmanın gerçekleşmesi sırasında bazı varsayımlar yapılmıştır. Bu varsayımlar, ilgili problem özelinde olmayıp mobil robot literatüründe çeşitli çalışmalarda kabul edilmiş varsayımlardandır.

Çalışmada kullanılan varsayımlar aşağıda listelenmiştir:

- Çoklu robot sistemi birden fazla gezgin robottan oluşmaktadır,
- Çalışmada fiziksel robotların yanı sıra, fiziksel olmayan yazılım erkinleri de bulunmaktadır,
- Robotlar, çok türeldir. Birbirlerinden farklı/aynı yetenek/algılayıcıya sahip olabilirler,
- Çoklu robot sisteminde bulunan her robot ve her yazılım erkinini birbirleri ile açık haberleşme yapabilmektedirler,
- Robotlar dürüst ve işbirlikçidir,
- Robotların algılayıcıları her zaman doğru bilgi vermektedir,
- Robotlar aldıkları bir işi her zaman tamamlarlar.

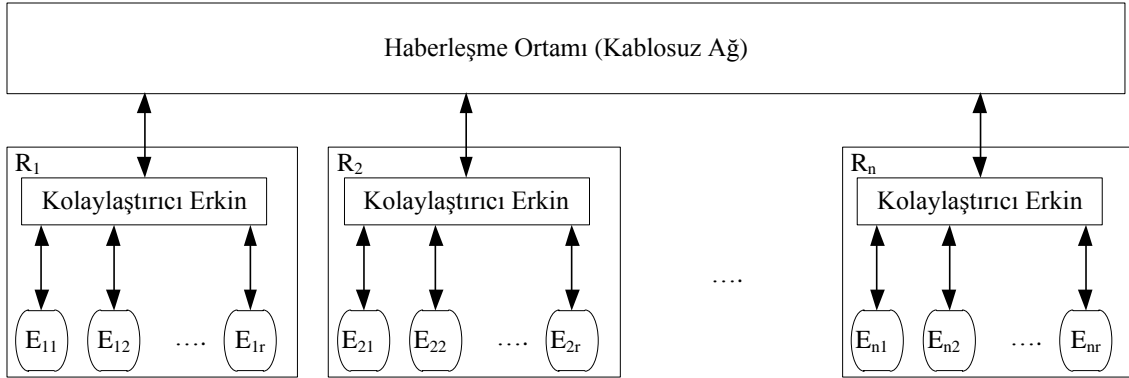
3.1 Çok Erkinli Yapılar

Erkin, kendine ait bir hesaplama yeteneği bulunan otonom varlık olarak tanımlanabilir. Erkinler algılayıcıları sayesinde çevreden aldıkları bilgileri yorumlayıp, gerek fiziksel gerekse ortama bilgi üreterek bulunduğu ortamı değiştiren ve otonom bir şekilde hareket edebilen varlıklardır (Weiss, 2000). Bu çalışmada önerilen yöntemin doğruluğunu test etmek için geliştirilmesi önerilen yazılım yukarıda anlatılan nedenlerden dolayı erkin tabanlı olarak geliştirilmiştir. Önerilen çözüm mimarisinde mobil robot fiziksel bir erkin olarak düşünülmüştür. Ancak bazı erkinler fiziksel olmayan, sadece yazılımdan ibaret olan erkinlerdir. Şekil 3.1’de gösterildiği gibi erkin çevreden sürekli bir duyu alır ve bulunduğu ortamı etkileyecek hareket çıktıları üretir.



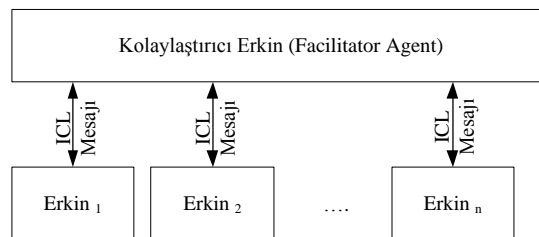
Şekil 3.1 Erkin ve çevrenin ilişkisi

Çalışmada kullanılan erkin mimarisi Şekil 3.2’de verilmiştir. Mimaride robotlar R_i , $i = 1, 2 \dots n$ şeklinde tanımlanmıştır. Her bir robotun içerisinde çeşitli görevleri gerçekleştirmek için barındırılan bir ya da birden fazla erkin bulunmaktadır, erkinler E_{ij}^R , $i \in [1, n]$ ve $j \in [1, \rho]$ şeklinde tanımlanmıştır (Özkan, 2007).



Şekil 3.2 Erkin tabanlı mimarinin genel gösterimi

Çalışmada erkinleri birbirleri ile haberleştirmek için Açık Erkin Mimarisi (Open Agent Architecture, OAA) olarak adlandırılan yazılım kullanılmıştır (SRI uluslararası yapay zekâ merkezi, 2007). Açık erkin mimarisi, Stanford Araştırma Enstitüsü (Stanford Research Institute) Yapay Zekâ Laboratuvarı tarafından geliştirilmiştir. Yazılım internet ortamından açık kaynak olarak indirilebilmektedir. Erkinler birbirleri ile ICL (Interagent Communication Language) adı verilen özel cümleler ile iletişim kurmaktadır. ICL mesajları platform ve yazılım bağımsız oluşturulur, bu sayede farklı işletim sisteminde koşan, farklı yazılım dilleri ile yazılmış olan erkinler birbirleri ile ortak haberleşme dili olan ICL mesajları üzerinden haberleşebilirler. Açık erkin mimarisine ait mesajlaşma yapısı Şekil 3.3’de gösterilmiştir.



Şekil 3.3 OAA mimarisi

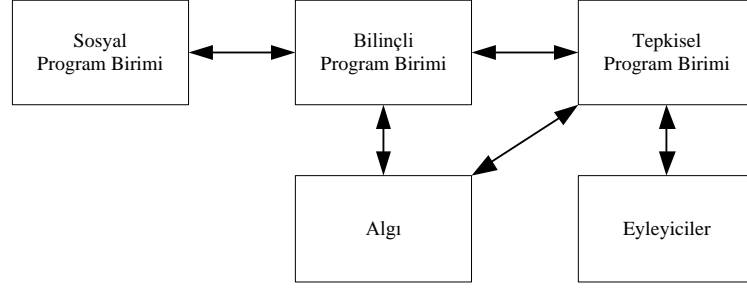
OAA ortamında haberleşmeler kolaylaştırıcı erkin (facilitator agent) üzerinden yapılmaktadır. Her erkin kendi sunduğu servisleri (yeteneklerini) kolaylaştırıcı erkine kaydettirmekte, bu sayede kendine gelen mesajları alıp, gelen mesajlara ilgili cevabı üretilip ilgili erkin/erkinlere iletebilmektedir. OAA mimarisi aşağıdaki teknik özelliklerinden, açık kaynak kod olmasından, java, c++, prolog yazılım dillerini desteklediğinden ve SunOs/Solaris, Linux, Windows 9x/NT/2000/XP işletim sistemlerinde çalışabildiğinden dolayı tercih edilmiştir (SRI uluslararası yapay zekâ merkezi, 2007).

OAA mimarisinin teknik özellikleri:

- Erkinler birçok programlama dilinde yazılabilirler,
- Çalışma zamanı sırasında erkinler güncellenebilir,
- Erkinler belirli bir ağda dağıtık olarak barındırılabilir,
- Erkinler belirli bir işin yapılması için paralel çalıştırılabilir,
- Erkinler PDA gibi mobil platformlarda da çalıştırılabilir.

3.2 Önerilen Mimari ve Sistemde Bulunan Erkinler

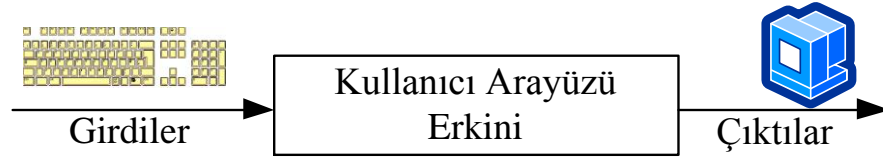
Bir gezgin robotun içinde bulunan program birimleri aşağıda Şekil 3.4'te gösterilmiştir. Sosyal program birimi robotun kullanıcı ile etkileşime girmesini sağlayan birimdir. Bilinçli program birimi hesaplamaların ve planlamanın yapıldığı birimdir. Tepkisel program birimi robotun kendini sakınmak için verdiği anlık tepkilerin oluşturulduğu birimdir. Algılama birimi ise robotun çeşitli algılayıcılarından verileri alan birimdir. Son olarak eyleyici (actuator) program birimi gezgin robotun motorlarının kontrol edildiği birimdir.



Şekil 3.4 Gezgin robotun içinde bulunan program birimleri

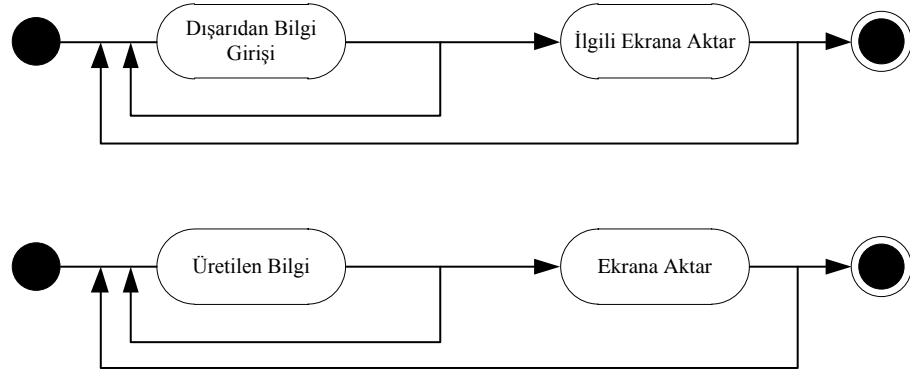
3.2.1 Kullanıcı arayüzü erkini:

Sosyal program birimi kapsamında geliştirilen kullanıcı arayüzü erkini insan ile robotların etkileşime girmesi ile görevli olan erkindir. Erkin, terminal ekranından aldığı komutları/görevleri ilgili erkinlere ulaştırır, erkinin etkileşimi Şekil 3.5’de gösterilmiştir. Kullanıcı arayüzü erkini programın çalışma zamanında ortaya çıkan uyarıları, üretilen bilgileri yine terminal ekranı sayesinde kullanıcılar ile paylaşır.



Şekil 3.5 Kullanıcı arayüzü erkini

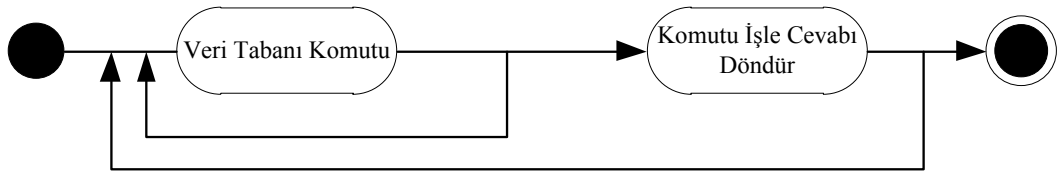
Kullanıcı arayüzü erkini sürekli dışarıdan bir bilgi girişi olup olmadığını ya da gösterilmesi gereken bilgi olup olmadığını kontrol eder. Eğer böyle bir bilgi var ise ya girilen komutu ilgili birime ulaştırır ya da üretilen bilgiyi ekranda gösterir. Erkinin durum diyagramı Şekil 3.6’da gösterilmiştir.



Şekil 3.6 Kullanıcı arayüzü erkini durum diyagramı

3.2.2 Veri tabanı erkini

Çalışma zamanı sırasında oluşan bazı bilgilerin ve ortak erişilmesi gereken parametrelerin tutulduğu bir karatahta (blackboard) yapısına benzer bir yapı kurulmuştur. Ortak erişilmesi gereken bilgiler veri tabanı adı verilen dosyaya kaydedilmektedirler. Ancak veri tabanı dosyası önerilen mimaride tek bir tane olduğu için herhangi bir anda yalnızca bir erkin dosyayı okuyup, yazabilir. Ancak çok erkinli sistemlerde aynı anda gelen birden fazla erişim isteğinin gelmesi ihtimali çok yüksektir. Aynı anda oluşan erişim isteklerinin bir kuyruk yapısı içerisinde çözülmesi gereklidir. Veri tabanı erkini veri tabanı dosyasına okuma ya da yazma isteğinde bulunan erkinleri “ilk gelene ilk hizmet esası” (first come first served) prensibi ile kuyruğa dizer ve erkinlerin ilgili isteklerini veri tabanı dosyasında çalışmasını sağlar. Eğer veri tabanı dosyasında çalıştırılan komut okuma komutu ise cevap metni soruyu soran erkine veri tabanı erkini tarafından yanıt olarak gönderilir. Veri tabanı erkini durum diyagramı Şekil 3.7’de gösterilmiştir.



Şekil 3.7 Veri tabanı erkini durum diyagramı

3.2.3 Müzayede erkini

Çoklu robot sistemlerinde, iş yapma veriminin yüksek olması için aynı anda birden fazla görev robotlar tarafından paylaşılarak yapılmalıdır. Literatürde çoklu robot sistemleri ile ilgili yapılan çalışmaların çoğu robotların görev paylaşım teknikleri üzerine yoğunlaşmıştır.

Çalışmada görevler, robotlara müzayede yapısı kullanılarak dağıtılmaktadır. Müzayede yapılarında temel olarak iki rol bulunmaktadır, teklif alma ve teklif verme. Teklif alma ve teklif verme süreçleri her bir robotta bulunan müzayede erkini tarafından yürütülmektedir. Ortamda bulunan robotlar müzayede erkini kullanarak hem dağıtılmamış işler için müzayedeler düzenleyerek işleri diğer robotlara dağıtabilir, hem de var olan bir iş için teklif verebilir.

3.2.3.1 Teklif alma rolü

Müzayede erkini teklif alma rolünü sistemde dağıtılmamış işleri diğer robotlara dağıtmak için kullanır. Müzayedeyi açan erkin (teklif alan erkin) önceden belirlenmiş bir zaman aralığında rasgele bir zaman bekler ve veri tabanı erkinine dağıtılmamış bir iş olup olmadığını sorar. Eğer ortamda hâlihazırda dağıtılmamış bir iş varsa veri tabanı erkini soruyu soran erkine özel bir cümle ile cevap verir. Bu özel cümle, işin numarasını, işin adını, işin öncelik değerini, işin yapılabilmesi için robotların hangi yeteneklere sahip olması gerektiği bilgisini taşıyan ister kelimesini ve işin koordinat bilgilerini içerir. Eğer dağıtılmamış bir iş yoksa “İş_Yok” kelimesi cevap olarak soruyu soran erkine iletilir. Veri tabanı erkininden gelen yanıtla göre erkin ya gelen işi dağıtmak için müzayede düzenler ya da hiç bir şey yapmaz. Yeni bir müzayede düzenlemek için müzayede erkini ortamda bulunan tüm müzayede erkinlerine (kendisi dâhil) özel bir cümle gönderir. Bu özel cümlenin başına “Yeni_İş” kelimesini bulunur. Diğer erkinlere gönderilen cümlenin yapısı aşağıda verilmiştir.

Yeni_İş, İş_No, İş_Ad, İş_Öncelik, İş_İster, İş_X_poz, İş_Y_Poz

Bu özel cümle müzayede erkini tarafından algılanarak müzayedeye teklif verme süreci başlar. Çalışmada belirli bir işi tanımlayabilmek için, işin numarasını, işin adını, işin öncelik değerini, işin isterlerini, işin bulunduğu konum parametrelerini tanımlamak yeterlidir. Önerilen yaklaşımda her robot her işi başarı ile gerçekleştirebilecek yeteneğe sahip olmayabilir.

İş ister kelimesi üç karakterden oluşmaktadır. Her bir karakter 1 ya da 0 değerini alabilmektedir, eğer karakter 0 ise ilgili yetenek o işi yapmak için gerekli değildir ama eğer 1 ise mutlaka iş yapacak olan robot ilgili yeteneğe sahip olmalıdır. İster kelimesini oluşturan karakterlerden en sağdaki birinci karakter olarak kabul edilmiştir. İster kelimesindeki karakterler aşağıdaki yetenekleri göstermek için kullanılmıştır.

1. Karakter: Hareketli olma ve kendini konumlandırabilme yeteneği
2. Karakter: Tutucuya sahip olma
3. Karakter: Kameraya sahip olma

Örnek olarak işe ait ister kelimesi “101” şeklinde olmuş olsun. Bu ister kelimesine baktığımızda ilk karakterin “1” olduğunu görürüz bu bize teklif verecek robotun mutlaka hareketli olmasını ve kendini konumlandırabilmesi gerektiğini gösterir. İster kelimesinde ikinci karakter “0” dır, bu karakterin “0” olması ilgili iş için tutucu ile hiçbir işlem yapılmayacağını dolayısıyla robotun tutucuya sahip olması ya da olmamasının hiçbir önemi olmadığını gösterir. Son olarak üçüncü karakterin “1” olması işin yapılabilmesi için mutlaka robot üzerinde kameranın olması gerektiğini gösterir.

3.2.3.2 Teklif verme rolü

Müzayede erkini, hem teklif alma hem de teklif verme rolünü de içerisinde barındırır. Robot müzayede erkininin teklif verme rolünü kullanarak, açık olan bir müzayedeye teklif verir. Müzayedeye teklif verme süreci teklif alma rolü tarafından gönderilen “Yeni_İş” kelimesi ile başlayan cümle ile başlar. Bu cümleyi alan erkin,

yeni bir müzayede olduğunu ve müzayedeye konu olan işe ait detay bilgilerini gelen cümleden ayırır.

Çalışmada robotlardaki çoktürelilik robotların birbirlerinden yetenek olarak farklı olması ile sağlanmıştır. Robotlarda hangi yeteneklerin bulunduğu robotta kayıtlı olan bir robot yetenek kelimesinde tutulmaktadır. Robot yetenek kelimesi işin tanımındaki ister kelimesi ile oldukça benzerdir. Yetenek kelimesindeki karakter 1 ise robotun ilgili yeteneğe sahip olduğu, 0 ise ilgili yeteneğin robotta bulunmadığı anlamına gelmesidir. Robot yetenek kelimesini oluşturan karakterlerden en sağdaki birinci karakter olarak kabul edilmiştir. Robot yetenek kelimesindeki karakterler aşağıdaki yetenekleri göstermek için kullanılmıştır.

1. Karakter: Hareketli olma ve kendini konumlandırabilme yeteneği
2. Karakter: Tutucuya sahip olma
3. Karakter: Kameraya sahip olma

Örneğin eğer robota ait yetenek kelimesi “110” ise robot hareketlidir, kendini konumlandırabilir ve tutucusu vardır ancak ilgili robot kameraya sahip değildir. Dolayısıyla eğer müzayedeki işte kamera kullanılması isteniyorsa bu işi yapmak için herhangi bir teklifte bulunamaz.

Robot kendi yetenek kelimesi ile müzayedeki işin ister kelimesini karşılaştırır, eğer kendi yetenekleri gelen işi yapmaya yetiyor ya da fazla ise gelen iş için maliyetini hesaplayarak müzayedeyi düzenleyen robota bildirir. Eğer robotun yetenekleri müzayedeye olan iş için yeterli değilse robot ilgili müzayede için teklif vermez.

3.2.3.3 Maliyet hesaplama

Ortamda bulunan her bir robot açık olan müzayedeye teklif verebilir. Teklif verme süreci yukarıda da belirtildiği gibi, müzayede erkini tarafından gönderilen özel cümle ile başlar. Bu cümleyi alan her robot önce kendi yetenek kelimesi ile gelen işin

ister kelimesini karşılaştırır, eğer kendi yetenekleri gelen işi yapmaya yetiyor ya da fazla ise gelen iş için maliyetini hesaplayarak müzayedeyi düzenleyen robota bildirir.

Robot müzayedeki işin maliyet değerini görevlerin hangi kısıtlara göre dağıtılması isteniyorsa ona uygun maliyet fonksiyonu ile belirler. Uygulamada görevlerin dağıtılabileceği kısıtlar aşağıda belirtilmiştir.

- Yakınlık derecesi
- Öncelik değeri
- Görev-robot uyumu

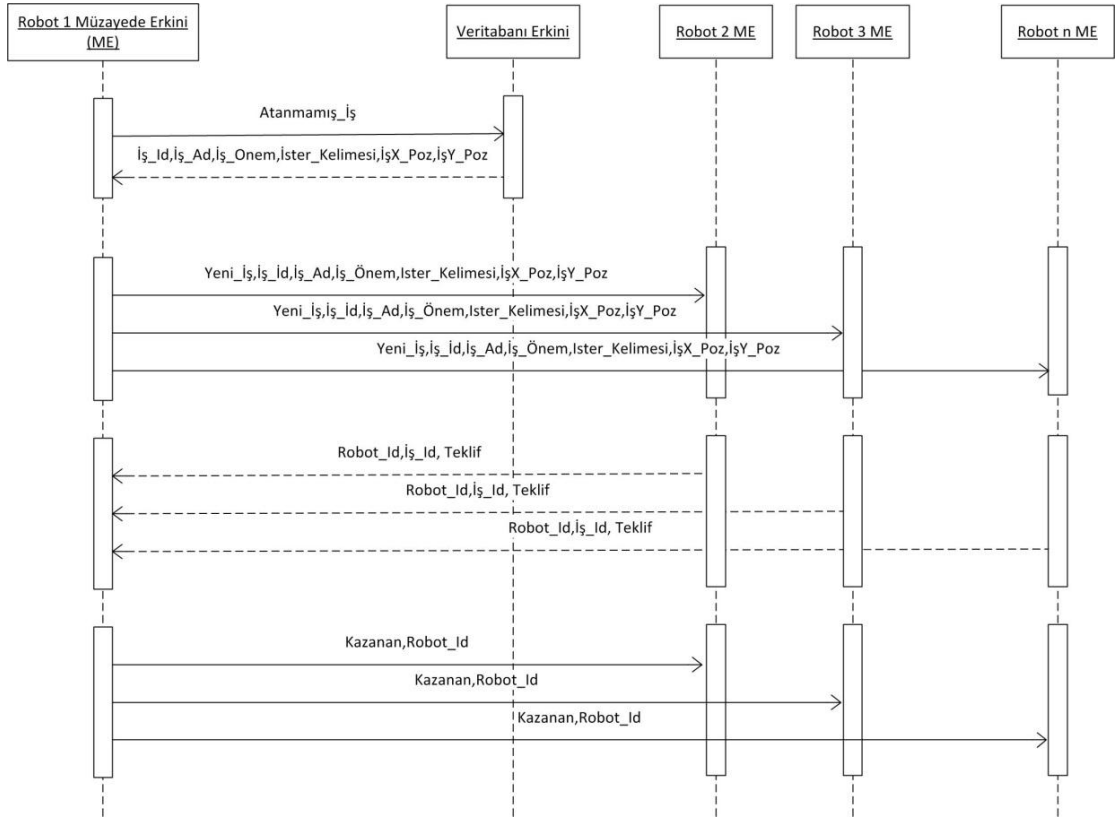
Görevleri robotlara dağıtmak için, yukarıda listelenen kısıtlar tek tek ya da aynı anda birden fazla kullanılabilir.

Müzayedede maliyet hesaplamada kullanılacak uygun araçlardan birisi de robotun kat edeceği mesafedir çünkü gezgin robotlar enerjilerini üzerlerinde taşırlar ve robotun kat edeceği mesafe kullanacağı enerji ile doğru orantılıdır, dolayısıyla çalışmanın tamamında kullanılan temel maliyet robotun kat edeceği yolun ilgili robotun hıza bölünmüş halidir, yani o işi yapmak için harcayacağı zamandır. Farklı kısıtlara göre görevler dağıtılmak istenildiğinde ilgili zaman hesaplamasının yanına bazı ek parametreler konulmuştur. Maliyet fonksiyonu çalışmanın ilerleyen bölümlerinde anlatılmıştır.

3.2.3.4 Müzayede süreci

Çalışmada görev dağıtmak için kullanılan müzayede süreci müzayede erkininin veri tabanı erkinine atanmamış iş olup olmadığını sorması ile başlar eğer ortamda atanmamış bir görev var ise veri tabanı erkini müzayede erkinine atanmamış işe ait id, ad, önem değeri, ister kelimesi ve işin pozisyonunu içeren bir cümle döner. Bu cümleyi alan müzayede erkini gelen cümleyi tüm robotlara bildirir. Gelen müzayede cümlesine göre her robot kendi maliyetini hesaplayarak müzayedeyi düzenleyen robota kendine ait robot id'sini, iş id'sini ve hesapladığı maliyetini içeren bir cümle döner. Müzayede

erkini atanmamış iş mesajını gönderdikten sonra belirli bir zaman robotlardan gelecek olan maliyet değerlerini bekler, bu zaman bittikten sonra gelen teklifler arasından en iyi olan teklifi seçerek işi yapacak olan yani kazanan robotu belirler ve kazanan robotu tüm robotlara bildirir. Kazanan robot da müzayedesini düzenlenen işi kendi iş listesine ekler. Robotların müzayede başlatma sıralaması ve diğer erkinler ile olan iletişimi aşağıdaki sıralama diyagramı Şekil 3.8’de verilmiştir.



Şekil 3.8 Müzayede erkini ve veri tabanı erkininin müzayede başlatma sıralama diyagramı

3.2.4 Konumlandırma erkini

Konumlandırma erkini robotun aldığı görevi sağlıklı yapabilmesi için hayati öneme sahiptir. Konumlandırma erkini robotun konumlandırılmasından, ortamın haritasının saklanmasından ve robotun konum ve yön bilgisine ihtiyaç duyan diğer erkinler ile haberleşip istenilen bilginin ilgili erkin ile paylaşılmasından sorumlu olan erkindir. Gezgin robotun sağlıklı bir şekilde çalışabilmesi ve verilen görevleri yerine

getirebilmesi için, çevresini algılaması ve kendini bulunduğu çevrede konumlandırması gereklidir. Gezgin robot üzerinde çevresini algılayabilmesi için ses üstü algılayıcı, lazer algılayıcı ve kamera gibi algılayıcılar bulunmaktadır.

Gezgin robot kendini en kolay kör sayma yöntemi olarak adlandırılan her iki tekerleğinde bulunan adım sayıcılardan gelen bilgileri kullanarak konumlandırabilir. Adım sayıcılar robotun merkez noktasının başlangıç pozisyonuna göre konumunu (x,y) ve dönme açısını (θ) değerlerini tekerleklerden aldığı dönme sayılarına göre hesaplar. Ancak tekerleklerin patinaj yapması, kayması ya da adım sayıcıların yeteri hassasiyette okuyamaması gibi sebeplerden adım sayıcılardan alınan konum ve yön bilgisinde belirli bir hata oluşur, oluşan bu hata zamanla toplanarak artar. Eğer robotun seyrüseferinde sadece adım sayıcılardan gelen bilgiler kullanılacaksa bu bilgi zaman içerisinde diğer algılayıcılardan hesaplanan konum bilgisi ile düzeltilmesi gereklidir.

Robotun konumlandırmasında kullanılacak bir diğer yöntem robotun konum ve yön bilgisinin bir kamera ile bulunup robota kablosuz iletişim ile gönderilmesidir.

Robotun kendini konumlandırmasındaki bir diğer yöntem ise lazer ve sonar algılayıcısı ile aldığı uzaklık verilerinin önceden robota yüklenmiş ortam haritası ile karşılaştırılmasıdır. Alınan verilerdeki bazı örüntüler ortamın haritasında aranarak robotun konumunun tam olarak bulunması sağlanır.

Çalışmada kullanılan gezgin robotun kendini konumlandırması için lazer algılayıcısından alınan bilgiler uzaklık bilgileri ortam haritası ile karşılaştırması yöntemi kullanılmıştır. Bu yöntem ile konum bilgisinin alınması için gezgin robotların üzerinde koşan işletim sisteminin içerisindeki ArLocalizationTask sınıfı ve metotları kullanılmıştır. Robotun yön bilgisi robot üzerinde takılı olan pusuladan okunarak elde edilmiştir.

3.2.5 Hareket erkini

Hareket erkininin görevi, robotun aldığı görevleri gerçekleştirmek için yapması gereken fiziksel hareketler ile ilgili komut setini oluşturarak, robotun mikro denetleyicisine bildirmektir. Hareket erkini diğer erkinler ile ortak bir şekilde çalışır. Robot aldığı görevleri gerçekleştirmek için yapması gereken fiziksel hareketleri planlamakta Şekil 3.9'da gösterilen akış şemasını kullanır.



Şekil 3.9 Hareket erkini düğüm hesaplama akışı

Robot hareket erkini daha önceden sıralanmış olan iş listesini okur, ilk işten başlayarak son işe kadar ardışık olan işleri yapmak için ziyaret edilmesi gereken düğüm noktalarını Dijkstra'nın en kısa yol algoritmasını kullanarak hesaplar. Tüm görevleri gerçekleştirmek için ziyaret edilmesi gereken düğümler hesaplandıktan sonra robot harekete başlar.

3.2.6 Planlama erkini

Gezgin robotta bulunan planlama erkini, diğer tüm erkinler ile etkileşim halinde kalarak robotun planlama ile ilgili olan işlerinden sorumlu erkindir. Gezgin robotta yapılacak olan görev ve rota planlama işleri ve diğer erkinlerin yapması gereken görevlerin senkronizasyonu planlama erkini tarafından kontrol edilir. Planlama erkini gerektiğinde diğer erkinler ile haberleşerek onlardan bilgi de alır. Planlama erkini alınan görevler için iki çeşit planlama yapar bunlardan ilki görev planlamadır. Görev planlama, robot tarafından yapılmak üzere alınan görevlerin gerçek zamanlı olarak öncelik, yakınlık gibi kısıtlara göre gerçekleştirilme sırasına konulmasıdır. Planlama erkini görevleri sıraya koyarken müzayede erkini ile ortak çalışır. İkinci yapılan planlama ise rota planlamasıdır. Rota planlaması ise alınan görevlerin koordinatlarına bakarak, istenilen noktaları içerecek şekilde en kısa yolun çıkarılmasıdır.

Planlama erkininin görevleri aşağıda listelenmiştir.

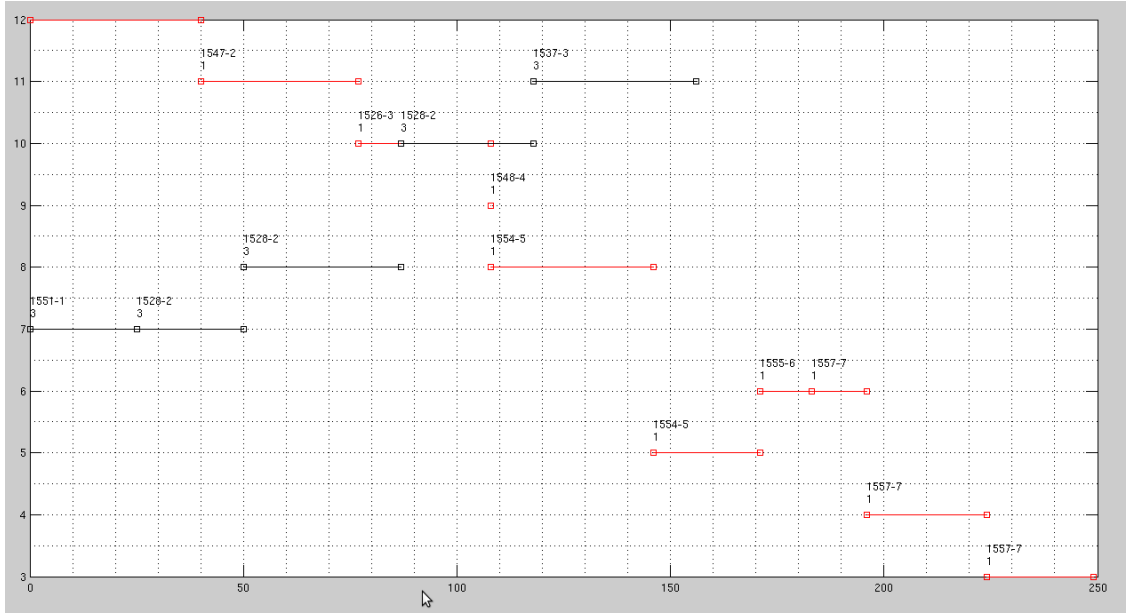
- Robotun aldığı işlerin listesini tutar ve müzayede erkini ile ortak çalışarak alınan işleri sıraya koyar.
- Sıralanmış işlerin rota planlamasını yapar.
- Robotun aldığı işleri belirlenen sıraya göre yapılması için hareket erkini ile haberleşerek ilgili komutları hareket erkinine iletir.
- İşin yapılıp yapılmadığını kontrol eder eğer bir görev yapılmış ise ilgili görevi listeden silerek başka bir listeye taşır.

3.2.7 Rota çakışma erkini

Rota çakışma erkininin görevi robotların görev planlarına gerektiğinde müdahale ederek, çalışma zamanı sırasında oluşabilecek robotların rotalarındaki çakışmaları önlemektir. Her robot yeni bir iş aldığı anda rota çakışma erkinine yeni görev planını bildirir, rota çakışma erkini de gelen iş listesini elindeki diğer işler ile karşılaştırarak rotalarda bir çakışma olup olmadığını tespit eder, eğer bir çakışma varsa mevcut

çakışmayı çözebiliyorsa çözerek yeni çakışma olmayan görev planını ilgili robot/robotlara bildirir.

Rota çakışma erkini kendisine gelen tüm görevleri robot no, görev düğüm numarası ve görev önceliğini barındıran bir listede tutar. Rota çakışma erkini robottan gelen tüm işlere robot tabanlı çizge/zaman listesi oluşturur ve çakışmaları bu çizge zaman listelerinde arar.



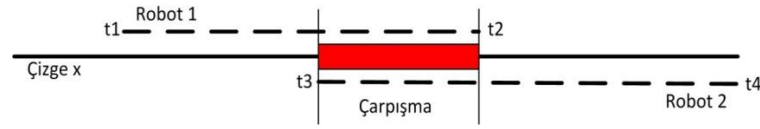
Şekil 3.10 Robot tabanlı çizge/zaman çizelgesi

Yukarıdaki Şekil 3.10'da rota çakışma erkinin tuttuğu verilerden oluşturulmuş çizge/zaman grafiği görülmektedir. Grafikte x eksenini zamanı gösterirken y eksenini robotun ilgili zamanda kullanacağı çizgeleri göstermektedir. Örneğin 0 ile 50 saniye arasında, 3 numaralı robot 7 numaralı çizgeyi kullanacaktır, aynı zaman zarfında 1 numaralı robot 12 ve 11 numaralı çizgeleri kullanacaktır. Grafikten de görüleceği gibi 10 numaralı çizge 80 ve 105. saniyeler arasında hem 1 hem de 3 numaralı robot tarafından kullanılmak istenmektedir. Dolayısıyla 10 numaralı çizgede 80 ve 105. saniyeler arasında bir çakışma mevcuttur. Bu çakışmanın, rota çakışma erkini tarafından çözülmesi beklenmektedir.

Rota çakışma erkini çalışma zamanında oluşan çakışmaları çözmek için iki yöntem izler. Bunlardan birincisi çakışmayı ortadan kaldırmaya yetecek kadar bir robotu bekletmek, ikincisi ise çakışma olan çizgeyi çözümde olmayacak şekilde yeni bir en kısa yol bulmak. Rota çakışma erkini hem araya gecikme eklemenin maliyetini hem de alternatif rotanın ek maliyetini hesaplar ve daha az maliyetli olan çözümü kabul eder.

3.2.7.1 Rota çakışmasını zaman gecikmesi ekleyerek gidermek

Çakışmayı çözmek için ilk kullanılan yöntem belirli bir zaman ilgili robotu bekletmektir. Robotun kaç saniye boyunca bekletileceği tamamı ile çakışmanın hangi saniyeler arasında olduğuna bağlıdır. Örneğin Şekil 3.11’de görüleceği gibi, çizge x robot 1 ve robot 2 tarafından kullanılmak istenmiştir. Fakat t3 ve t2 zamanları arasında robotların rotalarında çakışma mevcuttur.

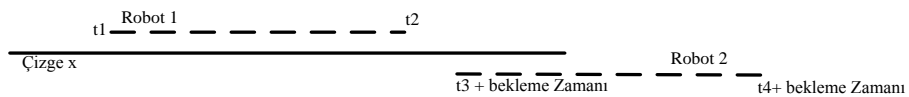


Şekil 3.11 Rotada çakışma örneği

Bu çakışmayı çözmek için Robot 2’nin çizge x’i kullanmaya başlama zamanının ötelenmesi gerekmektedir. Ne kadar beklemesi gerektiği aşağıdaki Denklem 3.1 ile hesaplanır.

$$\text{Bekleme Zamanı} = (t2-t3) + \Delta t \quad (3.1)$$

Burada Δt bekleme zamanına ekstra bir güvenlik aralığı olması için eklenmiştir. Robot 2’nin planında ilgili değişiklikler yapıldıktan sonra Şekil 3.12’de görüleceği üzere ortada herhangi bir çakışma kalmamıştır.



Şekil 3.12 Çakışma çözüldükten sonra çizge/zaman çizelgesi

3.2.7.2 Rota çakışmasını yeniden yol planlaması yaparak gidermek

Rotadaki çakışmayı önlemekteki bir diğer yöntem ise çakışmanın olduğu çizge/çizgeleri hesaplanacak yeni yolda olmayacak şekilde yeni bir yol bulmaktır. Çalışmada yeni bir rota hesaplamak için Dijkstra'nın en kısa yol algoritması kullanılmaktadır. Dijkstra'nın en kısa yol algoritması robotik alanında oldukça çok kullanılan bir kısa yol bulma algoritmasıdır.

Dijkstra'nın en kısa yol bulma algoritması 1959 yılında Dijkstra tarafından yayınlanmıştır (Dijkstra, 1959). Algoritma bir çizge üzerinde iki düğüm arasındaki en kısa yolu bulmayı garanti eder.

Algoritmada

$d_{ij} (\geq 0)$ olmak üzere i düğümünden j düğümüne olan uzaklığı gösterebilir.

Dijkstra'nın en kısa yol bulma algoritması çözümü bulmak için çizge üzerindeki bütün düğümleri başlangıç noktasına göre etiketler. Her bir düğüm üzerindeki etikette iki bilgi bulunur. Bu bilgiler:

- u_i başlangıç düğümünden i düğümüne kadar olan en kısa yolun uzunluğu.
- k başlangıç düğümünden i düğümüne kadar olan en kısa yoldaki i 'den önceki düğümün numarası.

Yukarıdaki iki bilgi birleştirildiğinde j düğümü için etiket aşağıdaki gibi oluşturulur.

$$[u_j, i] = [u_i + d_{ij}, k], (\geq 0)$$

Çizge üzerindeki her bir düğüm geçici etiket veya kalıcı etiket olmak üzere işaretlenir. Geçici etiket aynı düğümüne daha kısa uzaklığa sahip başka bir etiket var ise daha kısa uzunluktaki etiketle değiştirilir. Eğer yoksa en kısa uzunluğa sahip olan geçici etiket kalıcı etiket olarak işaretlenir. Tüm düğümlerin etiketleri kalıcı etiket olana dek algoritma çalışır. Tüm etiketler kalıcı etiket olduktan sonra kalıcı etiket

listesinde ulařılmak istenen noktadan bařlangıç noktasına geriye gidilerek en kısa yol ıkarılır.

Algoritmanın adımları ařađıda verilmiřtir

0. adım :

Bařlangıç dğümü kalıcı etiketle $[0,-]$ řeklinde iřaretlenir.

i. adım :

j 'nin kalıcı dğüm etiketlenmemiř olması kořuluyla, i . dğümden ulařılabilen her j dğümü iin geici $[u_i + d_{ij}, i]$ etiketleri hesapla.

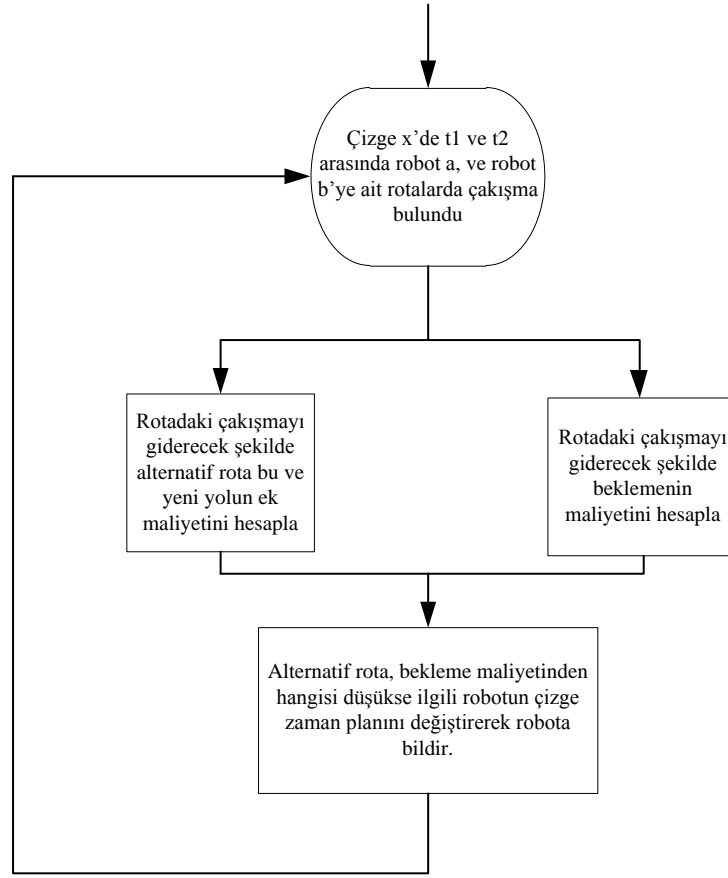
j dğümü bařka bir k dğümü iin de $[u_j, k]$ ile zaten etiketli ise ve $u_i + d_{ij} < u_j$ ise $[u_j, k]$, $[u_i + d_{ij}, i]$ ile deđiřtirilir.

Tüm etiketler kalıcı ise iřlem durdurulur. Aksi halde tüm geici etiketler $[u_r, s]$ arasından en kısa mesafeli olanı seilir (uzaklıklarda eřitlik varsa herhangi bir geici etiket seilebilir.)

$i = r$ olarak atanır ve i . adım tekrarlanır.

Dijkstra'nın en kısa yol bulma algoritmasının en zayıf olduđu konu bařlangıç noktasından bařlayarak tüm dğümleri gezip etiket listesini oluřturmasıdır. Ancak Dijkstra'nın en kısa yol bulma algoritması izge üzerinde en kısa yolu bulmayı garanti eder (2008, Arıcı).

Rota akıřma erkini robotların izge/zaman grafiklerinde herhangi bir akıřma bulunduđunda ařađıdaki řekil 3.13'deki gibi bir akıř diyagramı izleyerek ilgili akıřmayı özmeye alıřır.



Şekil 3.13 Rota çakışması çözüm diyagramı

4. UYGULAMA

Bu çalışmanın amacı, çoktürel robot grupları için müzayede yöntemi ile görevleri istenilen kısıtlara göre paylaştırmak ve görevler paylaşıldıktan sonra robotların rotalarında oluşan çakışmaları gidermektir. Bu bölümde çözüm aranan probleme önerilen yaklaşımın etkinliğini göstermek için yapılan uygulamalar ve sonuçları anlatılacaktır.

Önerilen yöntem Ubuntu 12.04 LTS işletim sistemi üzerinde C++ programlama dili kullanılarak kodlanmıştır. Geliştirilen kod, literatürde sıkça kullanılan ActivMedia tarafından geliştirilen MobileSim benzetim ortamında test edilmiştir. Benzetim ortamında P3-DX robotları kullanılmıştır.

Çalışmada önerilen çözüm yolunun etkinliğini göstermek için birçok alt uygulama yazılmıştır. Yazılan her uygulama bir önce yazılan uygulamanın üzerine yazılmıştır, böylelikle geliştirilen her bir alt uygulama ile tanımlanan problem adım adım çözülmeye çalışılmıştır.

Çalışmanın sonuca ulaşabilmesi için ilk yapılan uygulama, literatürde erkinlerin haberleşmesi için sıklıkla kullanılan açık erkin mimarisinin kurulup basit bir uygulama ile önerilen yöntem için kullanışlı olup olmadığının test edilmesidir. Uygulamada bir robot ve bir tane yol planlama erkinini vardır robot yol planlama erkinine mevcut konumunu ve gitmek istediği konumu açık erkin mimarisi üzerinden bildirir. İsteği alan yol planlama erkinini A* arama algoritmasını kullanarak robota izlemesi gereken rotayı bildirir. Yapılan bu uygulamanın sonucunda yazılan erkinlerin açık erkin mimarisi üzerinden haberleşebildiği, birbirleri ile veri paylaşabildiği görülmüştür.

Bir önceki uygulamada erkinlerin birbirleri ile haberleşmeleri ve rota planlama verisini doğru bir şekilde paylaşabildikleri test edilmişti. Bu uygulamanın üzerine müzayede sisteminin çalıştırılacağı market yapısı geliştirilmiş ve test edilmiştir. Uygulamada oluşabilecek hataları görebilmek için robot tipi ve görev paylaşılacak görev sayısı sınırlı tutulmuştur. Geliştirilen müzayede yapısını test etmek için uygulamada üç tane birbirinin aynı özelliklerine sahip robotlar kullanılmıştır, bu

robotlar tek bir görevi paylaşmak için teklif vermişler ve en düşük teklife sahip olan robot görevi kazanmıştır. Bu uygulama ile market yapısı, robotların nasıl görevleri alacağı müzayedenin nasıl düzenleneceği, erkinlerin çalışma sıraları belirlenmiştir.

Bir önceki uygulamada market yapısı oluşturularak müzayede düzenlenmiştir. Yeni yapılan uygulama ile market yapısına yapabilecekleri işler birbirlerinden farklı olan üç robot konulmuş ve rasgele oluşturulan 10, 15 ve 30 görev paylaştırılmıştır. Bu uygulamada ise bitiş süreleri, işlerinin geliş sıralarına göre yapılması ile görevlerin yakından uzağa göre sıralanarak yapılması karşılaştırılmıştır. Bu uygulama ile birlikte çoktörel bir robot grubu işleri müzayede yöntemi ile paylaşabilir ve işleri yakın olan önce, uzak olan daha sonra olacak şekilde yapabilme yeteneğine sahip olunmuştur.

Bir önceki uygulamada görevler sadece robotların göreve olan uzaklıklarına göre gerçekleştirilme sırası değişiyordu, yeni geliştirilen uygulamada robotun göreve olan uzaklığının yanı sıra görevin önem derecesi işin yapılma sırasını etkiler hale gelmiştir. Uygulamada göreve ait önem derecesinin robotların işleri bitirme sürelerine olan etkilerine bakılmıştır.

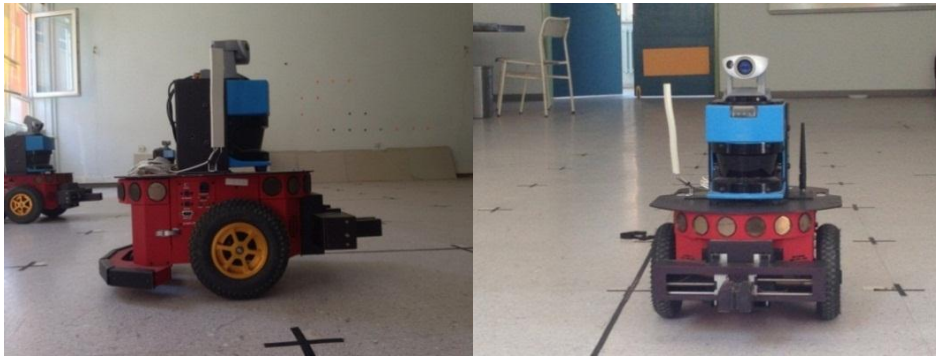
Çalışmada son olarak geliştirilen uygulamada nitelikleri birbirlerinden farklı dört robot kırk tane görevi paylaşmışlardır. Görevler robot görev uyumu ve yakınlığa göre paylaşılmıştır. Ayrıca görevler paylaşıldıktan sonra robotların rotalarında oluşan çakışmalar giderilmiştir.

Uygulamada kullanılan P3-DX robot platformunun teknik özellikleri, görevler ve robot tanımları, kullanılan ortam ile ilgili bilgiler, yapılan deneyler ile sonuçları ilerleyen bölümlerde detaylı olarak verilecektir.

4.1 Pioneer P3-DX Robot Platformu

Uygulamanın gerçekleştirilmesi sırasında ActivMedia tarafından üretilen Pioneer P3-DX gezgin robot platformu kullanılmıştır. P3-DX gezgin robot platformu 44,5 x 39,3 x 23,7 cm boyutlarındadır. Gezgin robotun hareketi robotun sağında ve

solunda bulunan ve birbirinden bağımsız hareket etme yeteneğine sahip iki servo motor tarafından sağlanmaktadır. Ayrıca robot platformunun dengede kalması için platformun arkasında hiçbir motora bağlı olmayan bir adet sarhoş tekerlek bulunmaktadır. Gezgin robotun sağ ve sol tekerleklerinin diferansiyel sürüş yeteneğine sahip olması gezgin robota oldukça yüksek hareket üstünlüğü sağlamaktadır. Gezgin robot platformunda bulunan 19 cm çapındaki iki adet tekerlek ile robot 1.6 m/s maksimum hıza ulaşabilir. Robot 23 kilograama kadar yararlı yük (payload) taşıyabilme kapasitesine sahiptir. Gezgin robot platformunun üzerinde bulunduğu ortamı algılamak için çeşitli algılayıcılar bulunmaktadır. Gezgin robotun her iki tekerleğinde gittiği mesafeyi kör sayma yöntemi ile hesaplamasını sağlayan 500 adımlı adım sayıcı (encoder) bulunmaktadır. Mesafeyi algılamak için 8 adet önde ve 8 adette arkada olmak üzere toplam 16 adet ses üstü mesafe algılayıcısı mevcuttur. Yine mesafe ölçmek için 1 derece hassasiyetinde ve 180 derecede algılama yapabilen SICK LMS200 lazer uzaklık ölçer bulunmaktadır. Ayrıca robotun yönünü bulmasını sağlayan sayısal pusula ve sayısal eğimölçeri bulunmaktadır. Robotun arka kısmında, çalışma zamanı sırasında oluşan çarpışmaları algılamak için tampon anahtarları (bumper switches) bulunmaktadır. Robotun üzerinde ortamı izlemek için Canon VC-C4 video kamerası bulunmaktadır. Üzerinde bulunan video kamerasının dönme, eğilme ve görüntüyü yakınlaştırma özelliği bulunmaktadır. Gezgin robotun gerektiğinde bir nesneyi tutup taşımak için ön tarafında iki serbestlik derecesinde sahip bir tutucu mekanizması bulunmaktadır. Robot platformunun resmi Şekil 4.1’de gösterilmiştir.

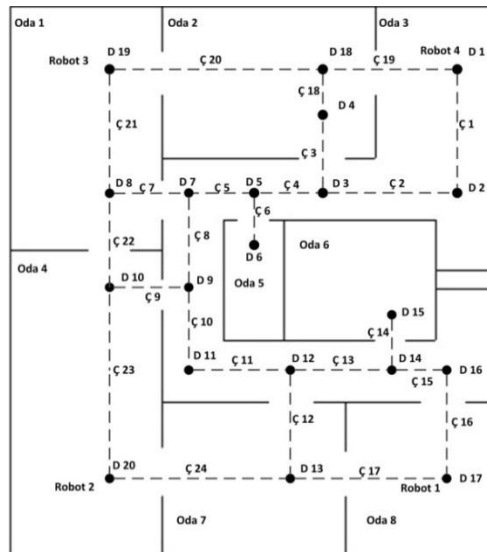


Şekil 4.1 Pioneer P3-DX robot platformu

Gezgin robot üzerinde Redhat 7.3 işletim sistemi kurulu olan Pentium tabanlı bir bilgisayar bulunmaktadır. Bu bilgisayar 802.11bgn kablosuz internet adaptörü sayesinde hem dış dünya ile iletişim kurulmasını sağlar hem de robotun donanımını kontrol eden mikrokontrolör kartını seri port üzerinden kontrol eder. Mikrokontrolör kartı üzerinde Activmedia tarafından geliştirilmiş olan ARCOS işletim sistemi kuruludur. ARCOS işletim sistemi sayesinde bilgisayar özel komutlar kullanarak donanımı kontrol edebilir ve donanımdan bilgi alabilir.

4.2 Benzetim Ortamı

Önerilen yöntemin etkinliğinin gösterilmesi için Activmedia tarafından geliştirilen ve literatürde sıklıkla kullanılan MobileSim benzetim ortamı kullanılmıştır. MobileSim benzetim ortamında kullanılan örnek bir dünya haritası Şekil 4.2’de gösterilmiştir. Haritada 8 adet oda ve 4 adet gezgin robot bulunmaktadır. Haritadaki her bir (•) bir düğümü göstermektedir. Her iki düğüm (D) arasında kalan kesikli çizgiler bir çizgeyi (Ç) göstermektedir. Çizgeler üzerinde herhangi bir yön kavramı yoktur, çizge üzerinde bir gezgin robot her iki yöne de hareket edebilmektedir.

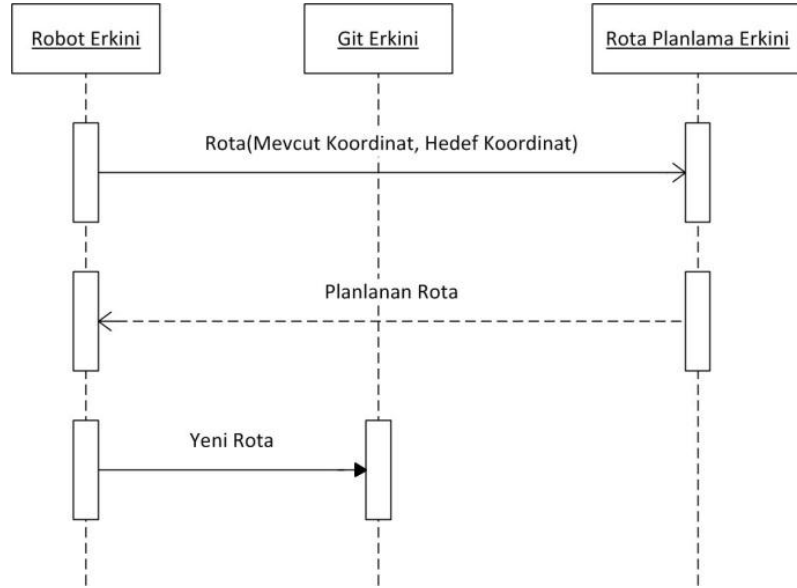


Şekil 4.2 Örnek dünya haritası

4.3 Açık Erkin Mimarisi Ortamının Kurulması ve Test Edilmesi

Çalışmada önerilen yaklaşımın gerçekleşmesi ve test edilebilmesi için birçok alt uygulama yazılmıştır. Yazılan alt uygulamalar belirli bir olgunluğa geldiğinde konu ilgili konferanslarda sunularak, geçerliliği test edilmiştir. Çalışmada erkinlerin birbirleri ile haberleşmesi için Açık Erkin Mimarisi (Open Agent Architecture, OAA) olarak adlandırılan yazılım kullanılmıştır.

Çalışmada, açık erkin mimarisinin test edilebilmesi için bir uygulama geliştirilmiştir. Geliştirilen uygulamada üç adet erkin bulunmaktadır, bu erkinler gezgin robotun kendisi (fiziksel erkin), git erkini (yazılım erkini) ve rota planlama erkini (yazılım erkini). Uygulamada robot şu anki koordinatını ve gitmek istediği hedef koordinatı rota planlama erkinine açık erkin mimarisi üzerinden gönderir. Rota planlama erkini robotun mevcuttaki koordinatına ve hedef noktanın koordinatına göre A* arama algoritması kullanarak bir rota planlar ve robota planlanan rotayı açık erkin mimarisi üzerinden geri gönderir. Robot gelen rota planını git erkinine iletir ve robotun istenilen noktaya en kısa yol üzerinden ulaşması sağlanır (Gürel, 2009). Uygulamanın sıralama diyagramı Şekil 4. 3' de gösterilmiştir.



Şekil 4.3 Sıralama diyagramı

Çalışmada en kısa yolu bulmak için kullanılan A* algoritması, literatürde iki nokta arasındaki en kısa yolu bulmak için sıkça kullanılan sezgisellerdendir.

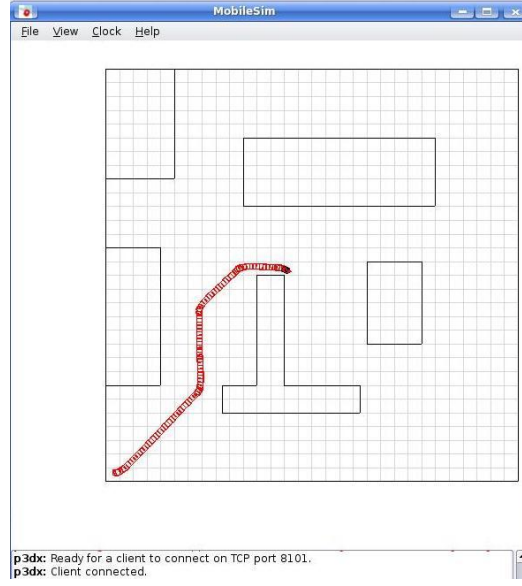
A* sezgiseli ilk defa 1968 yılında Peter Hart, Nils Nilsson ve Bertram Raphael tarafından yayınlanmıştır (Hart et al., 1968). A* sezgiseli aslında bir arama sezgiselidir başlangıç düğümünden başlayarak bitiş düğümüne kadar olan en az maliyetli yolu bulmak için kullanılır.

Sezgisel en az maliyetli yolu hesaplamak için iki maliyetin toplamını kullanır. Toplam maliyet fonksiyonu $f(x)$, uzaklık fonksiyonu $g(x)$ ile sezgisel fonksiyonu $h(x)$ 'in toplamı ile elde edilir.

$$f(x) = g(x) + h(x)$$

Denklemden bulunan $g(x)$ başlangıç düğümünden diğer düğüme olan yol uzunluğudur. Diğer bir parametre olan $h(x)$ ise bulunulan noktadan ulaşılmak istenen hedef noktaya olan kuş uçuşu uzaklıktır. A* sezgiseli karelere ayrılmış olan ortamlarda çalışır ortamdaki kareler ne kadar küçük ise o kadar yüksek çözünürlük elde edilir. Sezgisel bulunduğu karenin etrafındaki sekiz kare için $f(x)$ maliyet fonksiyonunu hesaplar ve en düşük maliyetli olanı seçip yine işlemi tekrarlar, bu işlemler hedef noktasına varıncaya kadar devam eder.

Çalışmada yukarıda anlatıldığı gibi robotun bir noktadan bir noktaya ulaşmak için kullanacağı rotayı, A* sezgiselini kullanarak rota hesaplama yeteneğine sahip rota planlama erkinine hesaplatır. Çalışmada kullanılan benzetim ortamı Şekil 4.4'de verilmiştir, ortam 1x1 metrelik karelere bölünmüştür ve her karenin orta noktası çizgedeki düğüm noktalarını olarak kabul edilmiştir. Robot başlangıçta (0,0) noktasında bulunmaktadır ve (13,15) noktasına gitmek istemektedir. Rota planlama erkinine gelen bu istek üzerine robotun izlemesi gereken rota hesaplanmış ve robota bildirilmiştir. Robotun izlediği rota Şekil 4.4'de gösterilmiştir.



Şekil 4.4 MobileSim benzetim ortamı ve uygulanan rota

Sonuç olarak geliştirilen uygulamada, erkinlerin birbirleri ile açık erkin mimarisi üzerinden haberleşirebildiği, farklı yeteneklere sahip erkinlerin belirli bir amaç için birlikte çalışabildiği görülmüştür. Ayrıca erkinlerden üretilen bilgilerin MobileSim benzetim ortamında da kullanılabilirdiği gösterilmiştir.

4.4 Market Yapısının Oluşturulması

Bir önceki çalışmada, erkinlerin haberleşmesi ve belli bir amaç için ortak çalışmaları için açık erkin mimarisinin uygun olduğu gösterilmiştir. Bu çalışmada market tabanlı uygulamalar için bir mimari önerilip gerçekleştirilecektir (Gürel, 2011). Market tabanlı sistemler görevlerin dağıtılması için hem merkezi hem de dağıtık yapıların avantajlarını içlerinde barındırırlar. Çalışmada önerilen market yapısı üzerine kurulmuş mimarideki tüm erkinler haberleşmelerini, açık erkin mimarisi üzerinden gerçekleştirmektedirler. Önerilen mimaride her bir gezgin robot içerisinde dört adet yazılım erkinini barındırır. Bu yazılım erkinlerinin her biri belirli bir görevde

uzmanlaşmıştır. Her bir gezgin robotun içerisinde bulunan erkinler aşağıda listelenmiştir.

- İş Üretme erkini
- İş Alma erkini
- Müzayede erkini
- Hareket erkini

4.4.1 İş üretme erkini

İş üretme erkininin temel görevi ortama belirli zaman aralıklarında rasgele zamanlarda, rasgele işler üretmektir. Üretilen rasgele işler market yapısı kullanılarak ortamda bulunan robotlara dağıtılır. Uygulamada erkinler birbirleri ile açık erkin mimarisi üzerinden Interagent Communication Language (ICL) mesajları ile haberleşirler. Erkinlerin birbirlerine gönderdikleri mesajlar özel bir formatta olmalıdır aksi takdirde hangi parametrenin ne amaçla gönderildiği anlaşılamayacağından erkinlerin birbirleri ile haberleşmeleri pek mümkün olmaz. İş üretme erkini tarafından rasgele oluşturulan işin parametreleri daha sonra dağıtılmak için ICL mesaj formatında veri tabanı erkinine gönderilir. Gönderilen ICL mesajının formatı aşağıdaki gibidir.

Yeni iş, İşin X koordinatı, İşin Y koordinatı

Mesajda bulunan ilk parametre “Yeni İş” veri tabanı erkinine gönderilen mesajın hangi amaç ile gönderildiğini söyler. Gönderilen ilk mesajda “Yeni İş” komutunu alan veri tabanı erkini gelen işi daha sonra dağıtılmak için veri tabanı dosyasına kaydeder. İş üretme erkininden gelen mesaja, veri tabanı erkini herhangi bir yanıt dönmez.

4.4.2 İş alma erkini

İş alma erkisinin görevi rasgele zamanlarda veri tabanı erkinine ortamda dağıtılmamış iş olup olmadığını sorar. Dağıtılmamış iş olup olmadığını sormak için aşağıdaki ICL mesajı formatını kullanır.

Al İş, Robot No

Veri tabanı erkini eğer dağıtılmamış iş var ise soruyu soran robota dağıtılmamış işin parametrelerini gönderir. Cevabı gönderirken aşağıdaki ICL mesajı formatını kullanır.

Al İş, Robot No, İş No, İşin X koordinatı, İşin Y koordinatı

İş alma erkinine gelen cevap, aynı zamanda müzayede erkini tarafından da alınır.

4.4.3 Müzayede erkini

Müzayede erkini, müzayede işlemini, iş alma erkinine gelen cevap ile başlatır. Çalışmada yapılan müzayede tipi tek çevrimlidir. Dağıtılacak olan iş anons edildikten sonra robotlardan gelen teklifler alınır, en düşük teklife sahip olan robot görevi olarak kendi görev listesine ekler. Çalışmada müzayede erkisinin iki rolü bulunmaktadır bunlar dağıtılmamış bir işi dağıtmak için müzayede düzenleme rolü ve var olan bir müzayedeye teklif verme rolü.

4.4.3.1 Müzayede düzenleme rolü

Robot, dağıtılmamış bir görevi dağıtmak için müzayede erkisinin müzayede düzenleme görevini kullanır. Müzayede düzenlemek için müzayede erkini üzerinden diğer robotlara özel formatta hazırlanmış olan ICL mesajını gönderir. Bu mesaj,

mesajın tipini, müzayede yapılacak olan işin numarasını ve işin koordinat bilgilerini içerir. Mesajın formatı aşağıda verilmiştir.

Müzayede, İş No, İşin X koordinatı, İşin Y koordinatı

Yukarıdaki mesajı alan diğer robotlar teklif verme rolünü kullanarak belirli bir zaman içerisinde kendi maliyetlerini, müzayedeyi düzenleyen erkine gönderirler. Müzayede erkini de belirli bir süre tekliflerin gelmesi için bekler ve gelen teklifler arasından en düşük maliyetli olanı seçer ve kazanan robotu diğer tüm robotlara duyurur. Kazanan robotun duyurulması yine özel formatlı bir ICL mesajı ile olur. Mesajda kazanan robot ve yapılacak olan işin numarası bulunur. Bu mesajı tüm robotlar alır ama sadece kazanan robot işi kendi listesine ekler. Mesajın formatı aşağıda verilmiştir.

Müzayede Kazanan, Robot No, İş No

4.4.3.2 Müzayedeye teklif verme rolü

Çalışmada her bir robot, açık müzayede mesajı aldığı anda kendi mevcut konumuna göre bir maliyet üreterek, müzayedeyi açan robota ilgili ICL mesajını gönderir. Robot işi yapma maliyetini, bulunduğu mevcut konumdan görev noktasına olan en kısa yolun uzunluğu olarak hesaplar. En kısa yolu hesaplarken Dijkstra'nın en kısa yol sezgiselini kullanır. Kendi maliyetini gönderirken robot diğer mesajlarda olduğu gibi özel bir ICL mesaj formatında gönderir. Mesajda mesajın tipi, robot ve işin numarası ve maliyet değeri bulunur. Mesajın formatı aşağıda verilmiştir.

Müzayede Teklif, Robot No, İş No, Maliyet

4.4.4 Hareket erkini

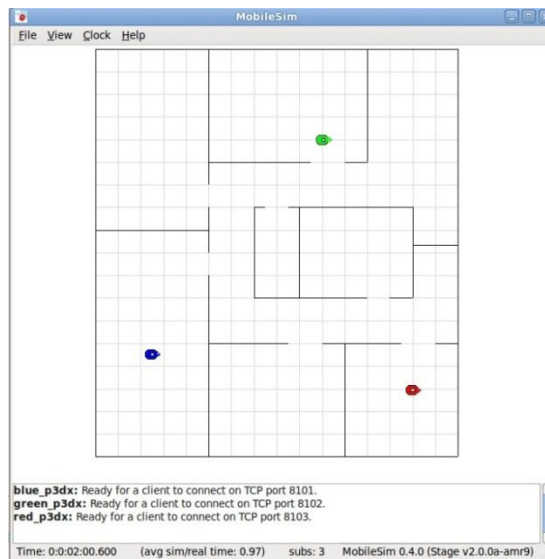
Hareket erkini robotun mevcut konumundan hedef konumuna kadar olan rotasında robotu hareket ettirmekle görevlidir. Hareket erkini rotaya bağılı olarak gerekli motor komutlarını mikrokontrolör kartına verir.

4.4.5 Veri tabanı erkini

Veri tabanı erkininin görevi gelen sorguları veri tabanı dosyasına kaydetmek ya da veri tabanı dosyası üzerinde diğer erkinlerin sorgu yapmasını sağlamaktır. Veri tabanı erkini sorguları çalıştırarak cevaplarını ilgili erkine iletir.

4.4.6 Market yapısı uygulaması

Kurulan yapının test edilmesi için bir yukarıda anlatılan erkinler kullanılarak bir uygulama yazılmıştır. Uygulamada 3 adet robot bulunmaktadır. Robotların ilk konumları Şekil 4.5’de gösterilmiştir.



Şekil 4.5 Benzetim Ortamı

Ortamda sadece bir adet görev (6000,8500) koordinatında bulunmaktadır. Robotlar sırası ile (2000,4500) (14000,3000) ve (10000,14000) noktalarında bulunmaktadır. Müzayede anonsu geldiğinde robotlar sırası ile 7656, 13500 ve 1059 teklifleri gelmiştir. Müzayede erkini en düşük teklife sahip olan 3 numaralı robotu kazanan robot olarak ilan etmiş ve robot görev noktasına gitmiştir.

Çizelge 4.1 Bir görev ataması için örnek teklifler ve sonuç

(6000,8500) noktasındaki görev			
Robot No	Mevcut Pozisyonu	Teklif	Durum
1	2000,4500	7656	
2	14000,3000	13500	
3	10000,14000	1059	KAZANAN ROBOT

Uygulama ile ilgili veriler ve sonuç Çizelge 4.1’de gösterilmiştir.

4.5 Market Yapısında Görevlerin Uzaklığa ve Robotların Yeteneklerine Göre Dağıtılması

Bir önceki çalışmada, market yapısı gerçekleşmiş ve tek bir tip iş ve aynı özelliklere sahip üç robot için yapı test edilmiştir. Şimdiki çalışmada ise robotlar ve işler farklılaştırılmıştır (Gürel, 2012). Çalışmada birden fazla farklı özelliklere sahip robot kullanılmış ve birden fazla görev dağıtılmıştır. Görevler market yapısı kullanılarak dağıtılmışlardır. Önceki çalışmadan farklı olarak robotlar sadece yeteneklerine uygun olan işler için teklif verebilmektedirler. Ayrıca robotlar gerçekleştirmek için birden fazla görev alabilmektedirler alınan işler her robotun iş listesine kaydedilmektedir. Robot kendi görev listesini enerjisini verimli kullanmak için en yakın komşu (nearest neighbor heuristic) sezgiselini kullanarak sıralamaktadır. Bu çalışmada da bir önceki çalışmada oluşturulan erkin tabanlı market mimarisi temel alınmıştır.

4.5.1 Görev ve robotların tanımı

Çalışmada iki çeşit görev bulunmaktadır bunlar görev 1 ve görev 2 olarak adlandırılmışlardır. Bu görevleri yapmak içinde ortamda üç adet robot bulunmaktadır. Çalışmada her robot her işi yapamamaktadır. Hangi robotun hangi işi yapabildiği Çizelge 4.2’de gösterilmiştir. Çizelge 4.2’de gösterilen ✓ robot görevi yapabilir olduğunu gösterirken, ✗ robot görevi yapamaz olduğunu göstermektedir.

Çizelge 4.2 Robot görev çizelgesi

	Robot 1	Robot 2	Robot 3
Görev 1	✗	✗	✓
Görev 2	✓	✓	✗

Görevler iş üretme erkini tarafından rasgele üretilmektedir. Üretilen işler market yapısı kullanılarak diğer robotlar tarafından paylaşılmaktadır.

4.5.2 Çalışmada kullanılan erkinler

Her bir gezgin robotun içerisinde bulunan erkinler aşağıda listelenmiştir.

- İş Üretme erkini
- İş Alma erkini
- Müzayede erkini
- Hareket erkini

Çalışmada kullanılan iş üretme erkini, iş alma erkini ve hareket erkini bir önceki çalışma ile aynı olduğundan yeniden anlatılmamıştır. Ancak kullanılan müzayede erkini bir önceki çalışmadan farklıdır. Müzayede erkini aşağıda detaylı bir biçimde anlatılmıştır.

4.5.2.1 Müzayede erkini

Çalışmada, müzayede erkini robotun müzayede düzenlemesi ve düzenlenen müzayedeye teklif vermesinden sorumlu olan erkindir. Müzayede düzenleme ve değerlendirme yöntemleri daha önceki çalışmalarda anlatıldığı gibidir. Bu çalışmayı diğer çalışmalardan farklı kılan robotun kendi iş listesinde iş varken, mevcut iş listesini gözeterik yeni bir işe teklif vermesi ve eğer müzayedeyi kazanırsa da yeni gelen işi kendi iş listesine sıralı biçimde eklemesidir. Dolayısıyla müzayedeye teklif verirken kullanılan müzayede verme rolünde deęişiklik yapılacaktır.

Robot açık olan bir müzayedeye kendi maliyetini doğru bir biçimde gönderebilmesi için kendi iş listesinin sıralı olması gerekmektedir. Robot işleri sıralarken en yakın komşu sezgiselini kullanır. Sezgisel temel olarak başlangıç düğümünden başlayarak, başlangıç düğümüne en yakın düğümü seçer bu durum hiç sıralanmamış nokta kalmayınca kadar devam eder. En yakın komşu sezgiselinin adımları aşağıda verilmiştir.

Adım 1: Gezgin robotun mevcut düğümünü başlangıç düğümü olarak al

Adım 2: Mevcut düğümüne en yakın olan ziyaret edilmemiş düğümü

Dijkstra'nın en kısa yol algoritması kullanılarak bul ve n olarak adlandır

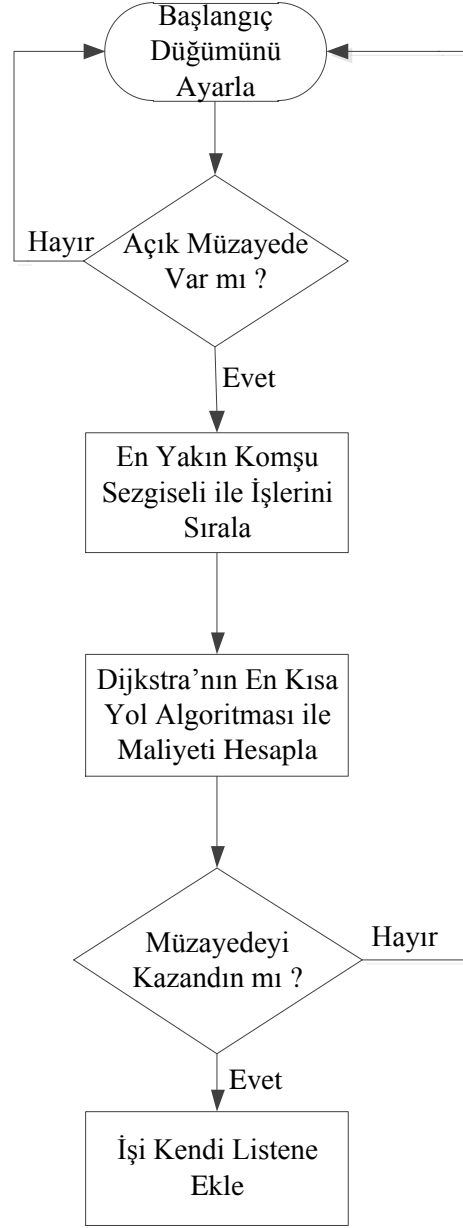
Adım 3: n düğümünü başlangıç düğümü olarak ata

Adım 4 n düğümünü ziyaret edilmiş olarak kaydet

Adım 5: Eğer ziyaret edilmemiş hiç düğüm kalmamışsa bitir

Adım 6: Adım 2 ye geri dön

Önceki çalışmada olduğu gibi açık olan müzayedeye her robot kendi maliyetini teklif olarak gönderir. Bu çalışmada maliyet, müzayededeki işi gerçekleştirebilmesi için robotun gideceği yolun uzunluğu olarak hesaplanmıştır. Robotun gideceği yol Dijkstra'nın en kısa yol algoritması kullanılarak hesaplanmıştır. Müzayede erkisinin açık olan bir müzayedeye teklif verme rolünde iken kullandığı akış şeması Şekil 4.6'da gösterilmiştir.

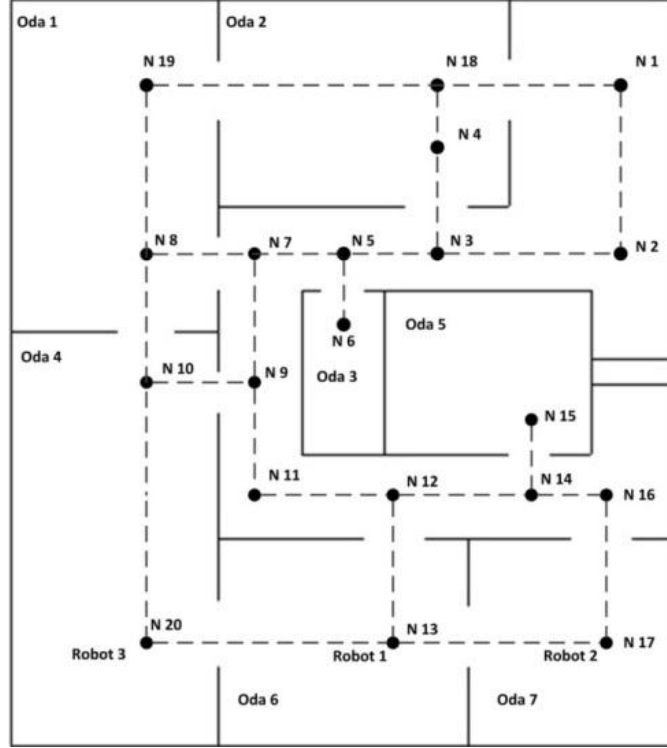


Şekil 4.6 Müzayede erkini teklif verme akış diyagramı

4.5.3 Çalışmanın uygulanması ve sonuçları

Çalışmada önerilen yöntemin test edilmesi için iş üretme erkini tarafından rasgele olarak sırasıyla 10, 15 ve 30 adet iş üretilmiş ve bu işler robotlar tarafından paylaşılmıştır. Çalışmanın test edilmesi için Şekil 4.7'deki gibi bir harita kullanılmıştır.

Benzetim ortamında robotların başlangıç noktaları sırası ile 13, 17 ve 20 nolu düğümlerdir. Haritada toplam 20 adet düğüm bulunmaktadır.



Şekil 4.7 Çalışmada kullanılan harita

Çalışmada elde edilen sonuçlar Çizelge 4.3'de verilmiştir. Görevleri sıralamanın sonuçlara olan etkisini gösterebilmek için robotlar aynı iş setini, işleri sıralayarak ve sıralamadan paylaşmışlardır. Çizelge 4.3'de sonuç isimli kolonda sıralamanın robotların işlerinin bitiş zamanına etkisi ölçülmüştür bu ölçüm sırasında 4.2 nolu eşitlik kullanılmıştır.

$$Sonuç = \frac{\text{Sırasız Listede Toplam Zaman}}{\text{Sıralı Listede Toplam Zaman}} \times 100 \quad (4.2)$$

Çizelge 4.3 Uygulamanın sonuçları

Deney	Robot	Sıralı Listede Toplam Zaman	Sırasız Listede Toplam Zaman	Sonuç
10 Görev Paylaşımı	Robot 1	84	83	Fark Yok
	Robot 2	287	393	Sıralı liste %137 daha iyi
	Robot 3	371	521	Sıralı liste %140 daha iyi
15 Görev Paylaşımı	Robot 1	171	171	Fark Yok
	Robot 2	579	817	Sıralı liste %140 daha iyi
	Robot 3	371	908	Sıralı liste %244 daha iyi
30 Görev Paylaşımı	Robot 1	346	384	Sıralı liste %110 daha iyi
	Robot 2	701	2319	Sıralı liste %330 daha iyi
	Robot 3	371	908	Sıralı liste %244 daha iyi

Sonuçlarda da görüleceği üzere, işler sıralanarak teklif verildiğinde robotlar daha kısa sürede kendi iş listelerini bitirebilmektedirler. Önerilen yöntemde kullanılan sıralama algoritması işlerin bitim sürelerinin kısaltılması ile ilgili olumlu sonuç vermiştir.

4.6 Görevlerin Uzaklığa ve Öncelik Değerine Göre Dağıtılması

Bir önceki çalışmada, market yapısı içerisinde robotlar işleri kendi içlerinde yapıma sırasına koyarak paylaşmışlar ve bunun sonucu olarak işleri sıralamadan geliş sırasına göre yapmanın sıralayarak yapmaya göre daha kötü sonuçlar verdiği ortaya çıkmıştır. Bu çalışmada ise robotlar kendi iş listelerini hem sıra hem de işin öncelik değerine göre sıralayıp ilgili müzayedeye maliyetlerini bildirmektedirler (Gürel, 2013a).

Çalışmada dört farklı robot ve dört farklı görev dağıtılmıştır. Robotlar sadece kendi yapabildikleri işlere teklif vermişlerdir. Görevler en yakın komşu sezgiselinin değiştirilmiş hali ile işlerin öncelik değerleri de dikkate alınarak sıralanmışlardır. Bu çalışmada da daha önceki çalışmalarda kullanılan erkin tabanlı market mimarisi temel alınmıştır.

4.6.1 Görev ve robotların tanımı

Çalışmada dört farklı görev bulunmaktadır görevler görev 1, görev 2, görev 3 ve görev 4 olarak adlandırılmışlardır. İlgili görevleri de yapmak için ortamda dört adet robot bulunmaktadır. Çalışmada bazı görevler tüm robotlar tarafından yapılabilmekteyken bazı görevler ise sadece belirli robotlar tarafından yapılabilmektedir. Hangi robotun hangi işi yapabildiği Çizelge 4.4’de gösterilmiştir.

Çizelge 4.4 Robot görev çizelgesi (✓ robot görevi yapabilir, ✗ robot görevi yapamaz)

	Görev 1	Görev 2	Görev 3	Görev 4
Robot 1	✓	✓	✓	✓
Robot 2	✓	✓	✓	✗
Robot 3	✓	✓	✗	✗
Robot 4	✓	✗	✗	✗

4.6.2 Çalışmada kullanılan erkinler

Görevlerin öncelik değeri de gözetilerek dağıtılacağı çalışma, daha önceki erkin tabanlı market yapısı üzerine kurulmuş olduğundan gezgin robotta bulunan erkinler önceki çalışmadaki erkinler ile aynı çalışma prensibine bağlıdır. Her bir gezgin robot içerisinde bulunan erkinler aşağıda listelenmiştir.

- İş Üretme erkini
- İş Alma erkini
- Müzayede erkini
- Hareket erkini

Mevcut çalışmada öncelik değerini de gözeterek sıralama yapmak ve maliyeti doğru hesaplayabilmek için müzayede erkininin teklif verme rolünde değişiklik yapılmıştır.

4.6.2.1 Müzayede erkini

Çalışmada, müzayede erkini robotun müzayede düzenlemesi ve düzenlenen müzayedeye teklif vermesinden sorumlu olan erkindir. Müzayede düzenleme ve değerlendirme süreçleri daha önceki çalışmalarda anlatıldığı gibidir. Bu çalışmayı önceki çalışmalardan farklı kılan robotların kendi iş listelerini hem işlerin yakınlığa hem de öncelik değerine göre sıralayarak bir maliyet üretmesi ve bu maliyet ile müzayedeye katılmasıdır. Robot bir görevi aldığı anda yine aynı şekilde alınan görevi kendi iş listesine değiştirilmiş en yakın komşu sezgiselini kullanarak eklemektedir.

En yakın komşu sezgiseli yukarıdaki çalışmada açıklanmıştır. Bu çalışmada öncelik değerini de maliyet hesabına katabilmek için sezgiselin uzaklık hesaplama fonksiyonunda değişiklik yapılmıştır. Değiştirilmiş en yakın komşu sezgiselinin adımları aşağıda verilmiştir.

Adım 1: Gezgin robotun mevcut düğümünü başlangıç düğümü olarak al

Adım 2: Mevcut düğüme en yakın olan ziyaret edilmemiş düğümü değiştirilmiş

Dijkstra'nın en kısa yol uzaklık değerini (D-DEKY) (4.3 nolu eşitlik)

kullanılarak bul ve n olarak adlandır

Adım 3: n düğümünü başlangıç düğümü olarak ata

Adım 4: n düğümünü ziyaret edilmiş olarak kaydet

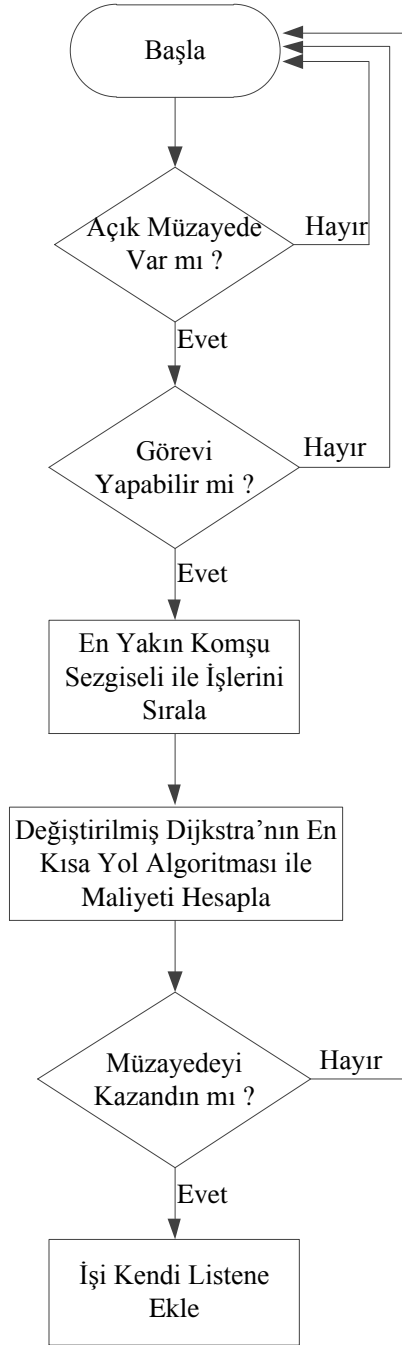
Adım 5: Eğer ziyaret edilmemiş hiç düğüm kalmamışsa bitir

Adım 6: Adım 2 ye geri dön

$$D-DEKY = \frac{\text{Dijkstra'nın en kısa yol uzaklık değeri}}{1 + \text{İşin Öncelik Değeri} \times \beta} \quad (4.3)$$

4.3 nolu eşitlikte bulunan β değeri öncelik değeri üzerinde kontrol parametresi olarak kullanılmıştır. β değeri 0 ile 1 arasında değer almaktadır, 0 değeri aldığı anda görevler öncelik değerine bağlı olmadan dağıtılırken sıfırdan farklı pozitif bir değer olduğunda işler sıralanırken önemli işler diğer işlerin önüne geçebilmektedir. Ancak burada dikkat edilmesi gereken başka bir husus ise bir işe ait öncelik değeri sadece değiştirilmiş Dijkstra'nın en kısa yol uzaklık değerini düşürmektedir, ama önemli bir

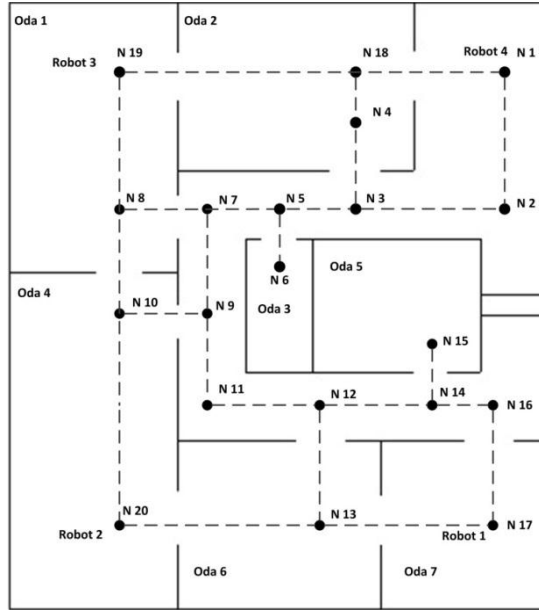
işin mutlaka ilk önce yapılacağını garantisini vermemektedir. Çok yakında olan az önemli bir iş, işlerin toplam bitirilme zamanını azaltmak adına önemli bir işten daha önce de yapılabilir. Müzayede erkininin teklif verme rolünün akış şeması Şekil 4.8’de gösterilmiştir.



Şekil 4.8 Müzayede erkini teklif verme akış diyagramı

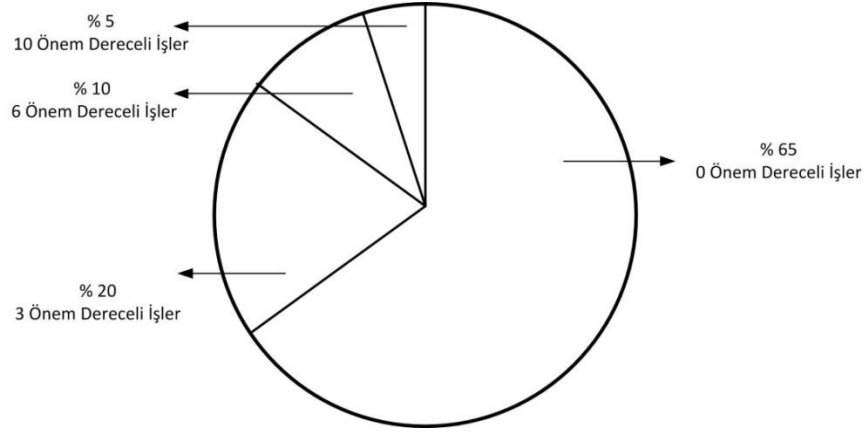
4.6.3 Çalışmanın uygulanması ve sonuçları

Önerilen yöntemin test edilmesi için iş üretme erkini tarafından rasgele olarak 20,40 ve 60 adet iş üretilmiştir. Üretilen bu işler robotlar tarafından paylaşılmıştır. Çalışmanın test edilmesi sırasında benzetim ortamında Şekil 4.9'daki harita kullanılmıştır. Başlangıç anında robotlar 17, 20, 19 ve 1 nolu düğümlerde bulunmaktadır.



Şekil 4.9 Çalışmada kullanılan harita

Çalışmada işlerin öncelik değeri 0, 3, 6 ve 10 değerlerini alabilmektedir. Öncelik değeri eğer 0 ise işin hiçbir önceliği yoktur, eğer 10 ise en önemli iş manasına gelmektedir. İşlerin öncelik değerlerinin ne olacağına rasgele karar vermek için rulet tekerleği yöntemi kullanılmıştır. Rulet tekerleği yönteminde kullanılan olasılıklar Şekil 4.10'da verilmiştir. En önemli işler %5 ihtimalle oluşturulurken önemi olmayan işlerin ihtimali %65'tir.



Şekil 4.10 Görevin öncelik değerini tanımlamada kullanılan olasılıklar

Holland tarafından 1975 yılında ortaya atılan rulet tekerleği tekniği (roulette wheel technique) olarak adlandırılan seçim yöntemi, adayların olasılık dağılımlarına göre seçilmesine dayanır (Holland, 1975). Bu yönetime göre olasılık değeri yüksek olan adayın seçilme şansı daha yüksek olmaktadır. Hangi görevin oluşturulacağı belirlenirken, 0 ile 100 arasında rasgele bir sayı üretilir, üretilen sayı (x) Denklem 4.4'deki fonksiyona sokularak görevin öncelik değerinin ne olacağı bulunur.

$$\text{Önem_Derecesi}(x) = \begin{cases} 0 \text{ Öncelik Değeri} & 0 \leq x \leq 65 \\ 3 \text{ Öncelik Değeri} & 66 \leq x \leq 85 \\ 6 \text{ Öncelik Değeri} & 86 \leq x \leq 95 \\ 10 \text{ Öncelik Değeri} & 96 \leq x \leq 100 \end{cases} \quad (4.4)$$

İş üretme erkini Denklem 4.4'ü kullanarak rasgele olarak görevlerin öncelik değerini atar. Çalışmada önerilen yöntemin test edilmesi için robotlar aynı iş setini, işleri öncelik değeri hesaba katarak ve katmayarak paylaşmışlardır. Çalışmanın sonuçları Çizelge 4.5'de verilmiştir. Çizelge 4.5'de sonuç isimli kolonda öncelik değeri hesaba katılıp katılmama durumunun robotların işlerinin bitiş zamanına etkisi ölçülmüştür bu ölçüm sırasında Denklem 4.5 kullanılmıştır.

$$\text{Sonuç} = \frac{\text{Öncelik değeri hesaba katılmış}}{\text{Öncelik değeri hesaba katılmamış}} \times 100 \quad (4.5)$$

Çizelge 4.5 Uygulamanın sonuçları

Deney	Robot	Robotların tamamlanma zamanı (sn). Öncelik değeri hesaba katılmış	Robotların tamamlanma zamanı (sn). Öncelik değeri hesaba katılmamış	Sonuç
20 Görev Paylaşımı	1	178	178	Robotların ortalama tamamlama zamanı öncelik değeri hesaba katıldığında, hesaba katılmadığı duruma göre %15 daha uzamaktadır.
	2	420	387	
	3	163	112	
	4	559	465	
	Ortalama	330	285	
40 Görev Paylaşımı	1	409	406	Robotların ortalama tamamlama zamanı öncelik değeri hesaba katıldığında, hesaba katılmadığı duruma göre %17 daha uzamaktadır.
	2	393	286	
	3	366	362	
	4	560	414	
	Ortalama	432	367	
60 Görev Paylaşımı	1	363	258	Robotların ortalama tamamlama zamanı öncelik değeri hesaba katıldığında, hesaba katılmadığı duruma göre %21 daha uzamaktadır.
	2	513	430	
	3	472	440	
	4	669	528	
	Ortalama	504	414	

Çalışmada görevler değiştirilmiş en yakın komşu sezgiseli ve market tabanlı erkin yapısı kullanılarak dağıtılmıştır. Deneyler sırasında her bir iş seti hem öncelik değeri hesaba katılarak hem de hesaba katılmadan dağıtılmıştır. Sonuç çizelgesindeki robotların işleri tamamlama sürelerini gösteren kolonlar aynı sayıdaki farklı işleri içeren deneylerin beş defa tekrarlanıp sonuçların ortalaması alınarak bulunmuştur. Sonuçlara bakıldığında öncelik değeri hesaba katıldığında işlerin tamamlanma sürelerinin uzadığı görülmektedir, öncelik değeri işlerin sırasını değiştirdiğinden bu beklenen bir durumdur. Ayrıca işlerin sayıları arttığında işlerin ortalama bitiş zamanları da lineer olarak arttığı görülmüştür. Sonuç olarak yapılan uygulamada işlerin öncelik değerine göre sıralandığındaki ortalama bitiş sürelerinin sıralanmadığı durumu göre kabul edilebilir bir oranda arttığı görülmüştür.

4.7 Market Yapısı ile Görevlerin Rotalarda Çakışma Olmadan Dağıtılması

Önceki çalışmalarda erkin tabanlı market yapısı kurulmuştur ve görevler önerilen mimariye göre uzaklığa, öncelik değerine göre dağıtılmıştır. Şimdiki çalışmada, görevler dağıtıldıktan sonra çalışma zamanı sırasında ortaya çıkacak rotalarındaki çakışmaları ortadan kaldıracak şekilde yeniden planlama yapılması planlanmaktadır. Rotalardaki çakışmaları önlenmesinin yanı sıra görevler dağıtılırken işler en uygun yetenekli ve dağıtılacak işe en yakın rotadan geçecek olan robota dağıtılması planlanmaktadır.

Çalışmada birbirinden yetenek olarak farklı robotlar ve birbirinden farklı yetenek ile gerçekleştirilebilecek işler bulunmaktadır. Robotun yeteneklerine uygun işi alması yetenek israfının önlenmesi adına oldukça önemlidir. Robotlar kendi yeteneklerinden daha az yetenek gerektiren işleri aldıklarında belki ileride yapılması gereken üst düzey bir işi enerjisi yetmediği için alamayacaktır. Dolayısı ile üst düzey yetenek gerektiren iş hiçbir robot tarafından yapılamayacaktır (Gürel, 2013b).

Çalışmada dağıtılacak olan işler en uygun yetenekli ve dağıtılacak olan işe en yakından geçecek robota göre dağıtılacaktır, işler dağıtıldıktan sonra da çakışmaları giderme rutini çalıştırılarak gerekiyorsa robotların rotalarında yeniden planlama yapılarak rotalardaki çakışmaların önüne geçilecektir. Yapılan bu yeni planlamanın çakışmalara etkisi ve ortalama iş bitirme zamanını ne kadar etkileyeceği tartışılacaktır. Çalışma, birbirinden yetenek olarak farklı dört robottan oluşan çöktürel robot grubunda test edilmiştir.

4.7.1 Görev ve robotların tanımı

Ortamda dağıtılması için dört farklı görev tanımlanmıştır bu görevler aşağıda listelenmiştir.

- Harita üzerinde bir düğümün ziyaret edilmesi
- Bir odanın ya da bir noktanın gözlenmesi

- Bir nesnenin bir odadan diğer odaya taşınması
- Bir nesnenin bir odadan diğer odaya kamera kullanarak da taşınması

Çalışmada belirtilen görevleri gerçekleştirebilmeleri için robotlar belirli bazı yeteneklere sahip olmak zorundadırlar. Eğer bu yeteneklere sahip değillerse ilgili görevleri almak için müzayedeye herhangi bir teklif veremezler. Çalışmada görevleri gerçekleştirmek için gerekli olan yetenekler gezgin olma, tutucu ve kamera olarak belirlenmiştir. Çizelge 4.6’da görevler ve görevlere gerçekleştirmek için gerekli olan yetenekler verilmiştir. Çizelge 4.6 ’de görünen 1 ve 0 ilgili yeteneğin görev için gerekli olup olmadığını gösterir. Eğer değer 1 ise ilgili yetenek görev için gereklidir 0 ise gerekli değildir. Çizelge 4.6’de ilgili görev için görünen 1 ve 0’ların görünen sıra ile bir araya gelerek oluşturduğu kelime çalışmada ister kelimesi olarak adlandırılır. İster kelimesi, ilgili görev için hangi yeteneklerin gerekli olup olmadığını özetleyen özel bir kelimedir.

Çizelge 4.6 Görevler ve gerekli olan yetenek/algılayıcı

Görev Adı	Gerekli Yetenekler				
	İster Kelimesi	Gezgin Olma	Tutucu	Kamera	
Bir düğümün ziyaret edilmesi	100	1	0	0	
Bir odanın ya da bir noktanın gözlenmesi	101	1	0	1	
Bir nesnenin bir odadan diğer odaya taşınması	110	1	1	0	
Bir nesnenin bir odadan diğer odaya kamera kullanarak taşınması	111	1	1	1	

Çalışmada önerilen yöntemin test edilmesi için dört farklı robottan oluşan çok türel bir takım kullanılmıştır. Görevlerin tanımlanmasında olduğu gibi robotların sahip olduğu özellikler ve yetenekler Çizelge 4.7’de verilmiştir. Çizelgede bulunan 1 ve 0’lar ilgili yeteneğin robotta mevcut olup olmadığını gösterir eğer değer 1 ise ilgili yetenek robotta mevcuttur eğer 0 ise mevcut değildir. Robotun sahip olduğu yeteneklerdeki 1 ve 0’ların bir araya gelerek oluşturduğu kelime ise çalışmada yetenek kelimesi olarak

adlandırılır. Bu kelime ilgili robot için hangi yeteneklerin mevcut olup olmadığını özetleyen özel bir kelimedir.

Çizelge 4.7 Robotlara ait yetenekler

Robot	Yetenekler			
	Yetenek kelimesi	Gezgin Olma	Tutucu	Kamera
Robot 1	100	1	0	0
Robot 2	101	1	0	1
Robot 3	110	1	1	0
Robot 4	111	1	1	1

Gezgin robotun bir işi gerçekleştirebilmesi için görevde istenen yeteneklerin tümüne ya da daha fazlasına sahip olması gereklidir. Eğer ilgili görev için bir yetenek bile robotta mevcut değilse robot o görevi yapamaz dolayısıyla yapamayacağı bir iş için müzayedede teklif vermez. Çizelge 4.6 ve Çizelge 4.7’de bulunan bilgileri birleştirdiğimizde, Çizelge 4.8’ü elde ederiz. Çizelge 4.8 bize hangi robotun hangi işi gerçekleştirip, hangisini gerçekleştiremeyeceğini, ilgili işi kaç robot tarafından yapılacağını özetler.

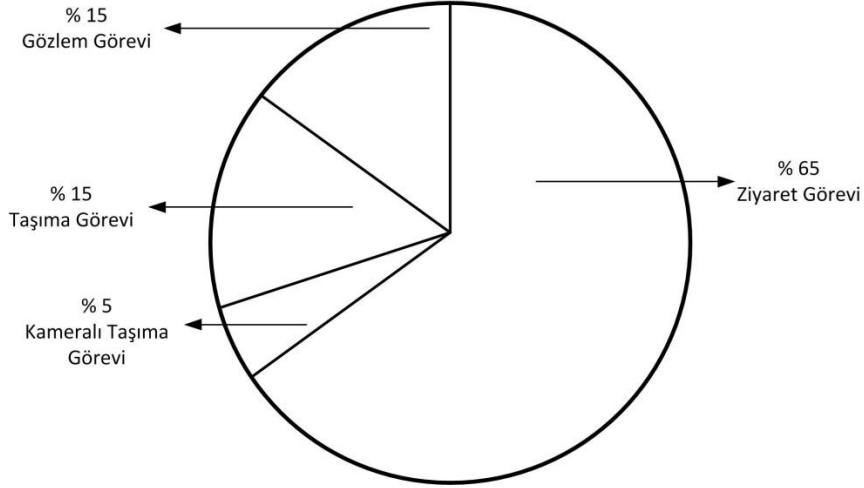
Çizelge 4.8 Robotlar ve gerçekleştirebilecekleri görevler

	Ziyaret Görevi	Gözlem Görevi	Taşıma	Kameralı Taşıma
Robot 1	✓	✓	✓	✓
Robot 2	✓	✓	✓	×
Robot 3	✓	✓	×	×
Robot 4	✓	×	×	×
Görevi Yapabilecek Robot Sayısı	4	3	2	1

Çizelge 4.8’e baktığımızda Robot 1 sadece ziyaret görevini yapabilmektedir. Ama Robot 4 görevleri gerçekleştirebilmek için tüm yeteneklere sahip olduğundan tüm

görevleri yapabilmektedir. Robot 2 ziyaret ve gözlem görevini yapabilmektedir. Robot 3 ise kamera ile taşıma görevi haricindeki tüm görevleri yapabilmektedir.

Uygulamada görevlerin tipleri rulet tekerleği tekniği yöntemi kullanarak iş üretme erkini tarafından oluşturulmuştur. İş oluşturma sırasında rulet tekniğinde kullanılan olasılıklar aşağıda verilmiştir. En yüksek olasılık her dört robot tarafından da yapılabilen ziyaret görevine verilmiştir, en az olasılıkta sadece Robot 4 tarafından yapılabilen kameralı taşıma görevine verilmiştir. Uygulamada kullanılan olasılıklar Şekil 4.11’de gösterilmiştir.



Şekil 4.11 Görev tanımlamada kullanılan olasılıklar

Denklem 4.6’deki fonksiyon kullanılarak iş üretme erkini tarafından hangi görevin üretileceği seçilir.

$$Görev_Oluştur(x) = \left\{ \begin{array}{ll} Ziyaret_Görevi & 0 \leq x \leq 65 \\ Gözlem_Görevi & 66 \leq x \leq 80 \\ Taşıma_Görevi & 81 \leq x \leq 95 \\ Kameralı_TaşımaGörevi & 96 \leq x \leq 100 \end{array} \right\} \quad (4.6)$$

4.7.2 Çalışmada kullanılan erkinler

Çalışmada daha önceki çalışmalarda kullanılan erkin tabanlı market yapısı kullanılmıştır. Dolayısıyla önceki çalışmalarda anlatılan erkinler ile aynı şekilde çalışmaktadırlar. Çalışmada görevler önceki çalışmalardan farklı kısıtlara göre dağıtılacağından sadece müzayede erkininde bazı değişiklikler yapılmıştır. Her bir gezgin robot içerisinde mevcut olan erkinler aşağıda listelenmiştir.

- İş Üretme erkini
- İş Alma erkini
- Müzayede erkini
- Hareket erkini

Yukarıda listelenen erkinler haricinde çalışma zamanı sırasında ortaya çıkacak olan rotalardaki çakışmaları engellemek için sistemde rota çakışma erkini isminde yeni bir erkin oluşturulmuştur.

4.7.2.1 Müzayede erkini

Çalışmada kullanılan müzayede erkininin müzayede düzenleme rolü önceki çalışmalarda aynı prensipte çalıştığından yeniden anlatılmayacaktır. Mevcut çalışmada görevler hem yakınlık hem de robotun yeteneklerine uygun olarak dağıtılacağından müzayede erkininin müzayedeye teklif verme rolünde değişiklik yapılması gerekmektedir. Görevleri robotlara uygunluk derecesine göre dağıtmanın etkilerini gözlemlemek için aynı görev seti üç farklı robot görev uyum kriterine göre dağıtılmış robotların aldığı işlerin bitiş zamanları ve robot görev uyum değerinin etkileri tartışılmıştır. Görevler aşağıdaki görev uyum yöntemlerine göre dağıtılmışlardır.

- Tam Görev Reddi
- Yarı Görev Reddi
- Görev Reddi yok

Tam Görev Reddi: Bu yöntemde, robot kendi yetenek kelimesi ile aynı ister kelimesine sahip olan işlere sadece teklif verebilir. Farklı olan hiçbir işe teklif vermez. Bu yöntemin amacı işi yaparken daha fazla yeteneğe sahip robotun enerjisini kullanmayıp, daha fazla yeteneğe sahip robotları daha fazla yetenek gerektiren işler için bekletmektir. Bu yöntem sayesinde işe ait ister kelimesi ile yetenek kelimesinin uyumu en yüksek düzeyde olur.

Yarı Görev Reddi: Bu yöntemde, robot teklif vereceği iş için robot yetenek ve iş ister konularında tam uyum aranmaz. Robot teklif vereceği iş için istenen yetenekleri sağlıyor ise ilgili işe teklif verebilir. Ancak burada robot yeteneği israfının azalması için maliyet fonksiyonunda bir ceza katsayısı gelir. Daha fazla yeteneğe sahip robot işe aynı uzaklıktaki daha düşük yetenekli bir robota göre daha büyük maliyet hesaplar. Böylelikle sistem kısmi olarak işi daha az yetenekli robota yönlendirir ama tam görev reddi yöntemindeki gibi mükemmel bir ister yetenek kelime uyumu aramaz.

Görev Reddi Yok: Bu yöntemde, robot teklif vereceği iş için herhangi bir yetenek ve ister kelime uyumuna bakmaz. Robotun teklif vermesi için sadece ilgili işi yapabiliyor olması yeterlidir. Bu yöntemde herhangi bir robot yetenek iş ister kelime uyumuna bakılmadığından, yüksek yetenek israfı bu yöntemde olmaktadır.

Maliyet hesaplama: Çalışmada maliyet hesaplamak için robotun kendi işleri ve müzayedede olan iş robotun başlangıç durumuna göre yakından uzağa doğru en yakın komşu sezgiseli kullanılarak sıralanır. Robotun başlangıç noktasından müzayededeki işe kadar olan rotanın uzunluğu Dijkstra'nın en kısa yol algoritması kullanılarak hesaplanır. Hesaplanan bu yol EKY (En Kısa Yol) olarak isimlendirilerek yukarıda anlatılan üç yöntemden hangisi seçilmiş ise ilgili değerler aşağıdaki denklemlere konularak robotun müzayedeye göndereceği değer belirlenir.

Maliyet hesaplamada üç denklem kullanılır. İlk denklem robotun yetenek kelimesindeki j karakter ile işe ait ister kelimesinin j karakterini karşılaştırır.

$$f(RY_k(j), GI_k(j)) = \begin{cases} 1 & \text{eğer } RY_k(j) > GI_k(j) \\ \text{Negatif } \zeta\text{BD} & \text{eğer } RY_k(j) < GI_k(j) \\ 0 & \text{eğer } RY_k(j) = GI_k(j) \end{cases} \quad (4.7)$$

Robot görev uyum değeri (RGUD) ister kelimesinin robot yetenek kelimesinin karakterlerinin karşılaştırılması ve toplamı olarak ifade edilir. Bu durum Denklem 4.8'de gösterilmiştir.

$$RGUD_k = \sum_{j=1}^3 f(RY_k(j), GI_k(j)) \quad (4.8)$$

Robot görev uyum değeri hesaplandıktan sonra müzayedeye gönderilecek teklif değeri Denklem 4.9 ile hesaplanır.

$$Robot_k \text{ Teklif} = (1 + RGUD_k)^{\tau_k} \times EKY_k \quad (4.9)$$

Yukarıda anlatılan yöntemlere göre seçilmesi gereken değişken değerleri ve koşulları aşağıda Çizelge 4.9'da gösterilmiştir.

Çizelge 4.9 Önerilen üç yöntem için parametre değerleri ve koşulları

Yöntem Adı	τ_k	Şart
Tam Görev Reddi (TGR)	Pozitif tamsayı	$RGUD_k = 0$
Yarı Görev Reddi (YGR)	$\tau_k = 1$	$RGUD_k > 0$
Görev Reddi Yok (GRY)	$\tau_k = 0$	$RGUD_k > 0$

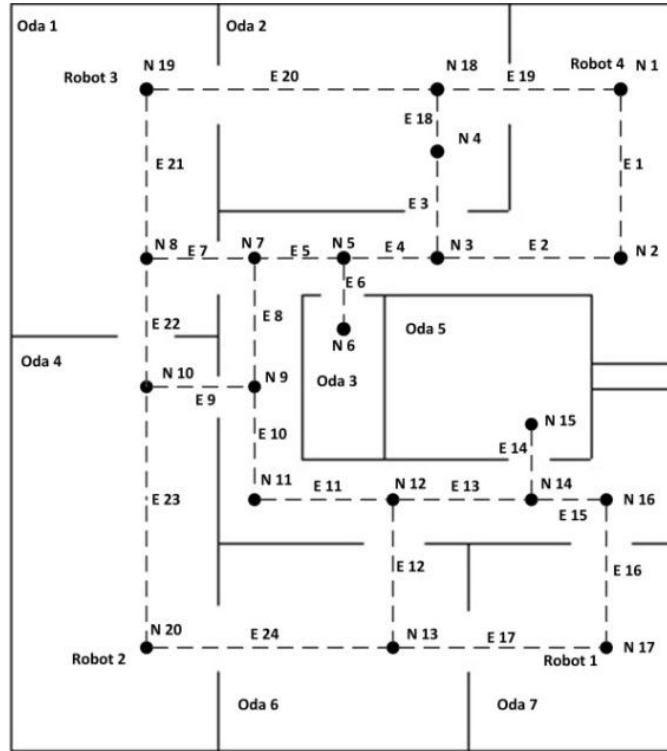
4.7.2.2 Rota çakışma erkini

Çalışmada, rota çakışma erkini diğer erkinlerden farklı olarak sistemde bir adet bulunmaktadır. Her robot işlerini ve işlerin yapılma sıralarını rota çakışma erkinine bildirmek zorundadır. Rota çakışma erkini tüm işler dağıtılabileceği kadar bekler ve tüm işler

dağıtıldıktan sonra her robotun çizge/zaman listesini oluşturur. Çakışmalar da bu çizge zaman grafiğinde aranarak bulunur. Çizge/zaman grafiğinde bulunan çakışmaların nasıl ortadan kaldırılabileceği bölüm 3.2.7.1 ve 3.2.7.2’de anlatıldığından yeniden anlatılmayacaktır.

4.7.3 Çalışmanın uygulanması ve sonuçları

Çalışmanın test edilmesi için birbirinden farklı dört gezgin robottan oluşan çoktörel bir robot grubu kullanılmıştır. Çalışmada kullanılan harita Şekil 4.12’de verilmiştir. Robotlar sırası ile 17,20,19 ve 1 numaralı düğümlerde bulunmaktadır.



Şekil 4.12 Uygulamada kullanılan harita

Uygulamada dağıtılmak üzere 40 görev rulet tekerleği yöntemi ile üretilmiştir ve robotlar tarafından paylaşılmıştır. Önerilen yöntemin doğruluğunu test etmek için deneyler yirmi defa tekrar edilmiştir. Oluşturulan görevlerin sayılarının ortalamalarına

baktığımızda Şekil 4.11’de gösterilen olasılık oranlarına yaklaştığını söyleyebiliriz. Yirmi defa farklı görev setleri ile tekrar edilen deneylerdeki ortalama görev sayıları Çizelge 4.10’da listelenmiştir.

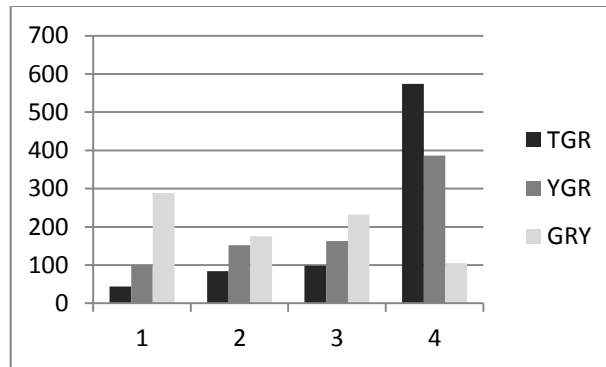
Çizelge 4.10 Görevlerin dağılımları

Görev	Ortalama Sayısı	Oran
Ziyaret Görevi	28,7	71,75
Gözlem Görevi	4,9	12,25
Taşıma Görevi	4,2	10,5
Kamerahı Taşıma Görevi	2,2	5,5

Çizelge 4.11’de tüm robotların her üç yöntemde kazandıkları görev sayıları ve bunların oranları verilmiştir Şekil 4.13’de kazanılan görevlerin sayıları grafik ile görselleştirilmiştir.

Çizelge 4.11 Önerilen üç yöntemde robotların kazandığı görev sayıları ve oranları

Robot	TGR		YGR		GRY	
	Görev Sayısı	Oran	Görev Sayısı	Oran	Görev Sayısı	Oran
1	44	5,5	99	12,375	288	36
2	84	10,5	152	19	175	21,875
3	98	12,25	163	20,375	232	29
4	574	71,75	386	48,25	105	13,125



Şekil 4.13 Robot görev kazanılan görev sayılarının grafiği

Deneylerin sonuçları Çizelge 4.12’de ve Çizelge 4.13’de verilmiştir. Çizelge 4.12’de önerilen üç yöntem için robotların bitiş süreleri robotların rotalarındaki çakışmalar dikkate alınmadan listelenmiştir. Çizelge 4.13’de ise çakışmalar dikkate alınmış ve önerilen üç yöntem için tüm robotların bitiş süreleri rotalarındaki çakışmalar ve çözülemeyen çakışmaların sayıları listelenmiştir.

Çizelge 4.12 Önerilen üç yöntem için rotalardaki çakışmalar dikkate alınmadan işlerin bitiş süreleri

	TGR	YRG	GRY
Deney	Bitiş	Bitiş	Bitiş
1	438	437	290
2	475	417	373
3	550	479	303
4	495	396	323
5	617	522	368
6	604	453	494
7	629	490	501
8	500	420	299
9	555	380	415
10	593	483	473
Ortalama	546	448	384

Çizelge 4.13 Önerilen üç yöntem için rotalardaki çakışmalar dikkate alındığında işlerin bitiş süreleri ve çakışma sayıları

	TGR		YRG		GRY	
Deney	Bitiş	Çakışma	Bitiş	Çakışma	Bitiş	Çakışma
1	438	0	476	4	295	3
2	481	8	479	16	424	10
3	727	4	668	29	316	2
4	518	6	447	12	439	11
5	723	16	605	16	378	2
6	743	9	492	10	515	4
7	691	8	724	24	634	7
8	547	8	486	12	370	12
9	593	6	521	11	471	8
10	598	4	664	19	541	28
Ortalama	605.9	6.5	556.2	16.55	438.3	9.33

Önerilen yöntemi test etmek için yapılan on deneyin sonucuna bakıldığında tam görev reddi (TGR) den görev reddi yok yöntemine gidildikçe işlerin toplam bitiş sürelerinde bir azalma görülmektedir. Bunun sebebi tam görev reddinde robotların sadece kendi yeteneklerine uygun olan işlere teklif verebilir olmalıdır. Tam görev reddi yönteminde işlerin bitiş süresi uzamaktadır ancak robotların görev israfı hiç olmamaktadır.

Görev reddi yok yönteminde en düşük iş bitirme zamanı elde ederken en yüksek robot görev israfı olmaktadır, aynı şekilde tam iş reddinde en yüksek iş bitirme zamanı elde ederken robot görev israf değeri en düşük değere sahip olmaktadır. Yarı görev reddi durumunda ise hem iş bitirme zamanı hem de iş israf değeri düşmektedir.

Çizelgeye bakıldığında rotalardaki çakışmaların ortaya çıkma sayıları ile deney yöntemi arasında herhangi bir ilişki olmadığı görülmüştür. Önerilen çakışma önleme yöntemi her üç yöntemde de başarı ile çakışmaları engellemiştir. Ancak çakışmaların ortadan kaldırılacak şekilde işlerin yeniden planlanması ortalama bitiş sürelerini arttırmıştır.

SONUÇ VE ÖNERİLER

Bu çalışmada, çoktörel robot grupları için görev atama ve çakışmasız rota planlama problemi ele alınmıştır. Ele alınan problem iki aşamaya ayrılarak çözülmüştür. İlk aşamada önerilen market tabanlı bir yaklaşım ile görevler robotlara dağıtılmış ikinci aşamada ise robotların rotalarında ortaya çıkacak olan çakışmalar ortadan kaldırılmaya çalışılmıştır. Görevler dağıtılırken müzayede yöntemi kullanılmıştır. Kullanılan müzayede yöntemi tek turlu yapılmıştır, en iyi teklifi veren robot müzayedeyi kazanmıştır. Robotlar tarafından müzayede için gönderilecek olan maliyet değerini etkileyen en önemli parametre robotun müzayedesini yapılan işe uzaklığıdır. Bundan dolayı müzayede sırasında en düşük teklifi veren robot müzayedeyi kazanan robot olarak belirlenmiştir. Görevler dağıtıldıktan sonra robotların rotalarında oluşan çakışmaların ortadan kaldırılması için literatürde bulunan çözümlerden yolun yeniden hesaplanması ve robotları bekletme yöntemlerinin maliyetleri karşılaştırılarak düşük maliyetli olan çözüm seçilerek rotalardaki çakışmalar ortadan kaldırılmaya çalışılmıştır. Önerilen yöntemin etkinliğini göstermek için birçok farklı senaryo ile değişik uygulamalar yapılmış ve sonuçları alınmıştır.

Gezgin robotların doğru çalışabilmesi için gerekli olan alt yöntemler (hareket, müzayede, planlama, vb.) erkin tabanlı olarak yazılmışlardır. Sistemin erkin tabanlı yazılmasının yazılım mühendisliği açısından avantajı sistemin hatalardan daha kolay arındırılabilmesidir. Erkin tabanlı mimarilerin en büyük avantajlarından biri ise sistemin gürbüzlüğünün artmasıdır. Sistemde bulunan tüm erkinler birbirlerinden bağımsız ancak ortak bir amaç için çalıştığından herhangi bir erkinin çalışmaz duruma gelmesi, tüm sistemin çalışmaz haline gelmesine değil sadece ilgili yeteneğin sistemden kaybolması anlamına gelir, böylelikle sistem kalan yetenekler ile çalışmaya devam edebilir.

Çalışmanın etkinliğini gösterebilmek için çeşitli senaryolarla birçok uygulama yapılmıştır. Yapılan uygulamalar basitten karmaşığa doğru birbirinin devamı şeklinde geliştirilmiştir. Yazılan uygulamalar MobileSim benzetim ortamında ve Matlab programında analiz edilmiştir. Çalışmanın amacı müzayede yöntemi kullanarak işlerin

öncelik derecesi, uzaklık, robot görev uyumu kısıtlara göre robotlara paylaştırılması ve işler dağıtıldıktan sonra robotların rotalarındaki çakışmaların giderilmesidir. Yapılan tüm alt çalışmalar bu hedefi gerçekleştirmek için gerekli olan alt yordamların oluşturulması içindir. Bu kapsamda gerçekleştirilen ilk uygulama erkin mimarisine göre kodlanmış tek bir robotun diğer bir erkinle haberleşip rota planlama yapmasıdır. Bu uygulamada amaç, geliştirilen erkin temelli yapının farklı ortamlarda koşan erkinlerin ortak bir amaç için kullanılabilir olduğunu göstermektir. Deneyden alınan sonuçlar erkinlerin belirli bir amacı gerçekleştirmek için haberleşebildiğini, bir birleri ile ortak bir amaç için çalışabildiğini göstermiştir. Gerçekleştirilen bu erkin tabanlı mimari ile artık görevleri dağıtabileceğimiz bir ortam elde etmiş olduk, bu erkin temelli mimarinin üzerine müzayede tabanlı görev dağıtım sistemi oluşturulmuş, oluşturulan bu yapı ile ortamda farklı noktalarda bulunan üç robot düzenlenen müzayede ile tek bir görevi yapmaya talip olmuşlardır. Ancak müzayedesini yapan görevi içlerinden göreve en yakın mesafede bulunan robot yapmaya hak kazanmıştır. Yazılan bu uygulama ile beraber oluşturulan müzayede tabanlı sistem kullanarak dağıtılması düşünülen görev için müzayedenin düzenlenebildiği, ortamda bulunan robotların çalıştığı düzenlenen müzayedeye teklif verebildiği ve verilen teklifler sonucunda da müzayedeki işin, işe en yakın robot tarafından alındığı görülmüştür. Geliştirilen bu uygulama bundan sonra gerçekleştirilecek müzayede tabanlı uygulamalar için temel oluşturacaktır. Çalışmada sunulan uygulamalar diğerinin üzerine konarak geliştirilmiştir, geliştirilen ilk müzayede tabanlı uygulamada tek görev paylaştırılmışken, mevcutken geliştirilen yeni uygulamada birden çok görev üç robota paylaştırılmıştır. Dağıtım sonrasında robotların aldığı işlerin tamamlanma sürelerini yakından uzağa göre sıralamış hali ve işleri alma sırasına göre karşılaştırılmıştır. İşlerin yakından uzağa göre sıralanarak yapıldığında, alınan işlerin daha kısa sürede tamamlandığı görülmüştür. Bir sonraki çalışmada ise işlere öncelik değeri verilmiş ve önceliğin işlerin tamamlanma sürelerine etkisi ölçülmüştür. Geliştirilen uygulamalar ile birden fazla robota birden fazla görevi, öncelik ve uzaklık değerine bağlı olarak müzayede yöntemini kullanarak dağıttık. Doktora tez çalışması kapsamında son olarak geliştirilen uygulamada ise robotların yetenekleri ile işlerin isterleri arasındaki uyuma bakılarak görevler dağıtılmış ve uyum değerinin işlerin bitiş süresine olan etkisi ölçülmüştür. Sonuç olarak geliştirilen yöntem

ile ortamdaki robotlar, robotun işe olan uzaklığı, işin öncelik değeri, robot-görev uyum değeri kısıtlarına göre görevleri paylaşabildikleri görülmüştür.

Çalışmanın bir diğer temel sonucu olan robotların rotalarındaki çakışma gidermek için rota çakışma erkini geliştirilmiş, geliştirilen erkin ile rotalardaki çakışmaların erkini tarafından çözüldüğü görülmüştür.

Çalışma literatüre çoktürel robot gruplarında müzayedede yöntemi ile görev dağıtımı ve dağıtım sonrasında robotların rotalarında oluşan çakışmaların giderilmesi konusunda katkı yapmıştır. Çalışmada ortaya konan erkin tabanlı mimarinin uygulanabilir olduğu ve ileride yapılacak olan çalışmalar için gelişmeye açık olduğu gösterilmiştir. Gelecekte yapılacak olan çalışmalarda sistemin etkinliğini arttırmak ve tamamen dağıtık çalışmasını sağlamak amacı ile rota çakışma erkini robotlar müzayedeye teklif verirken çalıştırılabilir. Ayrıca sistemdeki görevlerin önceliklerinin gerçek dünyada olduğu gibi zamana bağlı olarak adaptif değişmesi sağlanabilir. Ayrıca erkin tabanlı robot mimarilerinde yetenekler çoğu zaman önceden belirlenmiştir. Ancak bazı yeteneklerin işleyişi çalışma zamanı sırasında değiştirilebilir dahası robotlara çalışma sırası sırasında yeni yetenekler yazılım tanımlı (software defined) olarak robota eklenebilir ya da değiştirilebilir.

KAYNAKLAR DİZİNİ

Akıllı fiziksel erkinler kuruluđu, 2002, FIPA Contract Net Interaction Protocol Specification, <http://www.fipa.org/specs/fipa00029/SC00029H.pdf>, Ziyaret Tarihi: 2 Mart 2010.

Albus J.S., 1993, A Reference Model Architecture for Intelligent Systems Design, An Introduction to Intelligent and Autonomous Control pp. 27-56, Kluwer Academic Publishers.

Arbib M. A., 1998, Schema theory. In The handbook of brain theory and neural networks, Michael A. Arbib (Ed.), MIT Press, Cambridge, MA, USA pp. 830-834.

Arıcı V., 2008, Engellerin bulunduđu ortamda gezgin robotun en iyi yolu bulması ve izlemesi, Yüksek lisans tezi, Başkent Üniversitesi, Fen Bilimleri Enstitüsü.

Arkin C. R., Behavior-based robotics, MIT Press, Cambridge, Mass, USA, 1998.

Boer-Sorb'an K., 2008, Agent-based simulation of financial markets. A modular, continuous-time approach, PhD Thesis, Erasmus University Rotterdam.

Brooks R. A., 1986, A robust layered control system for a mobile robot, *Robotics and Automation*, IEEE Journal of Volume: 2, Issue: 1, pp. 14-23.

Cai C., Yang C., Zhu Q. and Liang Y., 2007, Collision avoidance in multi-robot systems, *Mechatronics and Automation*, ICMA 2007 International Conference on 5-8 Aug. 2007, pp. 2795-2800.

Čapek K., 1921, *Rossum's Universal Robots*.

Chaib-draa B. and Müller J., 2006, Multiagent based supply chain management, XIX, Springer, pp. 450.

Cox, D.J., 2002, Mock-up of hazardous material handling tasks using a dual-arm robotic system, *World Automation Congress*, Orlando, June 2002.

Deb S. R., 2009, *Robotics technology and flexible automation*, Tata McGraw-Hill Education Private Limited, September 29, 2009, pp. 13.

Dias M.B., 2004, *TraderBots: A new paradigm for robust and efficient multirobot coordination in dynamic environments*, PhD Thesis, Carnegie Mellon University.

- Dias M.B. and Stentz A., 2003, A comparative study between centralized, market-based, and behavioral multirobot coordination approaches, *Intelligent Robots and Systems, IROS 2003 International Conference on Volume: 3*, pp. 2279-2284.
- Dijkstra E., 1959, Two problems in connexion with graphs, *Numerische Mathematik*, pp.269-271.
- Evans J.M.,1994, HelpMate: an autonomous mobile robot courier for hospitals, *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems* , vol.3, pp.1695-1700.
- Gerkey B.P. and Mataric M. J., 2000, Murdoch: Publish/subscribe task allocation for heterogeneous agents, In *Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona, Spain*, pp. 203-204.
- Gerkey B.P. and Mataric M.J., 2002, Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation* 18(5), pp.758-768.
- Gerkey P.B., 2003, On multi-robot task allocation, PhD Thesis, University of Southern California Computer Science Department, August 2003.
- Gerkey B. P. and Mataric, M. J., 2004, A formal analysis and taxonomy of task allocation in multi-robot systems, *Journal of Robotics Research*, vol. 23(9), pp. 939-954.

- Golfarelli M., Maio D. and Rizzi S., 1997, Multi-Agent path planning based on task-swap negotiation, In: Proceedings 16th UK Planning and Scheduling SIG Workshop, Durham, England, pp. 69-82.
- Gürel U., Parlaktuna O., Adar N. and Canbek S., 2013, Task allocation in market-based applications by the nearest neighbor heuristic, *Anadolu University Journal of Science and Technology – A Applied Sciences And Engineering*, vol 14, No 2 (2013), pp. 173-182.
- Gürel U., Adar N. and Parlaktuna O., 2013a, Priority-based task allocation in auction-based applications, *Innovations in Intelligent Systems and Applications (INISTA)*, IEEE International Symposium, pp. 1-5.
- Gürel U. and Parlaktuna O., 2012, Task allocation in market-based applications by nearest neighbour heuristic, *8th International Symposium on Intelligent and Manufacturing Systems*, 27-28 September 2012, Adrasan, Turkey pp. 425-433.
- Gürel U. and Parlaktuna O., 2011, A new architecture for multi-robot teams in market-based applications, *Electrical and Electronics Engineering (ELECO)*, 7th International Conference, pp. II-430 - II-433.
- Gürel U., Parlaktuna O. and Kayır H.E., 2009, Agent-based route planning for a mobile robot, *International Advanced Technologies Symposium (IATS'09)*, May 13-15, 2009 at Karabük University, pp. 49-51.
- Hancock D.L. and Lamont G.B., 2011, Multi agent systems on military networks, *Computational Intelligence in Cyber Security (CICS)*, 2011 IEEE pp. 100-107.

- Hart P. E., Nilsson N. J. and Raphael B., 1968, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2), pp. 100-107.
- Holland J., 1975, Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor.
- Hur J., 2007, Multi-robot system control using artificial immune system, PhD, The University of Texas at Austin December 2007.
- Jennings J., Kirkwood-Watts C., 1998, Distributed mobile robotics by the method of dynamic teams, Distributed Autonomous Robotic Systems 3, pp. 47-56.
- Jones C. V., Mataric M.J., 2005, Behavior-based coordination in multi-robot systems, Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications , Geand S.S., Lewis F. L., Eds. Marcel Dekker, Inc.
- Kaleci B, 2010, Heterojen gezgin robot sistemleri için yeni bir market tabanlı görev paylaşırma yaklaşımı, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.
- Kawamura K., Wilkes D.M., Suksakulchai S., Bijayendrayodhin A., and Kusumalukool K., 2001, Agent-Based control and communication of a robot convoy, Proceedings of the 5th International Conference on Mechatronics Technology, Singapore.

Kibler S. and Raskovic, D, 2012, Coordinated multi-robot exploration of a building for search and rescue situations, System Theory (SSST), 44th Southeastern Symposium pp. 159-163.

Li X., Dang S., Hao N. and Li K., 2010, An extension to contract net protocol for military agents interactions, Computer Science and Education (ICCSE), 5th International Conference on pp. 1506-1511.

MacKenzie A., Miller J. O., Hill R. R. and Chambal S. P., 2012 , Application of agent based modeling to aircraft maintenance manning and sortie generation, Simulation Modelling Practice and Theory 20(1) pp. 89-98.

Malone W.T., Fikes E.R., Grant R.K. and Howard T.M. ,1988, Enterprise: a market-like task scheduler for distributed computing environments, In The Ecology of Computation, pp. 177-205, North-Holland.

Merkezi Araştırma Laboratuvarı, 1945, MSM-8 ele idare edilebilen manipölatör, <http://www.centres.com/images/crl-1-piece-telemanipulator.jpg>, Ziyaret Tarihi: 12 Aralık 2013.

Meystel A.,1990, Knowledge based nested hierarchical control, Advances in Automation and Robotics, vol.2, pp.63-152.

Murphy R. R., 2000, Introduction to AI Robotics, MIT Press, London, England.

Onn S., Sycara K., Gita Sukthankar and Mukherjee V., 1999, Agent aided aircraft maintenance, In Proceeding of Agents-99, pp. 306-312.

Özkan M., 2007, Farklı özelliklere sahip gezgin robot grubunun dağıtık kontrolü, Doktora Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.

Parker L. E., 1998, ALLIANCE: An architecture for fault-tolerant multirobot cooperation, IEEE Trans. Robot. Automation, vol. 14, pp. 220–240.

Ray, D.N. , Mandal, A. , Majumder, S. and Mukhopadhyay, S., 2011, Human-like gradual multi-agent Q-learning using the concept of behavior-based robotics for autonomous exploration, Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on pp. 2725-2732.

Sandholm T., 1993, An implementation of the contract net protocol based on marginal cost calculations, In Proceedings of the eleventh national conference on Artificial intelligence (AAAI'93), AAAI Press pp. 256-262.

Smith R. G., 1980, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Trans. Comput. 29, 12 (December 1980), pp. 1104-1113.

SRI uluslararası yapay zekâ merkezi, 1969, Shakey robotu, <http://www.frc.ri.cmu.edu/~hpm/book98/fig.ch2/Shakey.150.jpg>, Ziyaret Tarihi: 15 Ocak 2013.

- SRI uluslararası yapay zekâ merkezi, 2007, The open agent architecture, <http://www.ai.sri.com/~oaa/>, Ziyaret Tarihi: 1 Ocak 2011.
- Stentz A. and Dias M. B., 1999, A free market architecture for coordinating multiple robots, Technical Report CMU-RI-TR-99-42, Carnegie Mellon University.
- Weiss G., 2000, Multiagent systems: A modern approach to distributed artificial intelligence, MIT Press, USA.
- Werger, B.B. and Mataric, M.J., 2000, Broadcast of local eligibility for multi-target observation, Proceedings of DARS 4, pp. 347-356.
- Yoerger D. , A. Bradley, B. Walden, M. Cormier, and W. Ryan, 2000, Fine- scale sea floor survey in rugged deep-ocean terrain with an autonomous robot, in Proc. IEEE Int. Conf. Robot. Autom. , San Francisco, CA, Apr. 2000, vol. 2, pp. 1787-1792.
- Yuhong L., Chunsheng Y., Xuanmin D., 2008, Multi-Agent planning for ship collision avoidance, Robotics, Automation and Mechatronics, 2008 IEEE Conference pp. 549-554.
- Zheng C., 2009, Implementation of reactive control for a miniature surveillance robot, Information Technology and Computer Science ITCS. International Conference on (Volume: 2) pp. 510-513.

ÖZGEÇMİŞ

4 Mart 1980 tarihinde Edirne ilinde doğan ve Türkiye Cumhuriyeti vatandaşı olan yazar ilk, orta ve lise öğrenimini Edirne şehrinde tamamlamıştır. 1998- 2003 yılları arasında Eskişehir Osmangazi Üniversitesi Elektrik Elektronik Mühendisliği Bölümü'nde Lisans, 2004–2006 yılları arasında Elektrik Elektronik Mühendisliği Anabilim Dalı, Kontrol Kumanda Bilim Dalında Yüksek Lisans eğitimini tamamlamıştır. 2014 yılında Elektrik Elektronik Mühendisliği Anabilim Dalı, Kontrol Kumanda Bilim Dalında doktora eğitimini tamamlamıştır.