

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**ORTAM TARAMA İÇİN ROBOTLARLA DUYARGA AĞI
KONUMLANDIRMA**

**YÜKSEK LİSANS TEZİ
Meral CAMCI**

Anabilim Dalı : İleri Teknolojiler

Programı : Bilgisayar Bilimleri

HAZİRAN 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**ORTAM TARAMA İÇİN ROBOTLARLA DUYARGA AĞI
KONUMLANDIRMA**

**YÜKSEK LİSANS TEZİ
Meral CAMCI
(704051020)**

Tezin Enstitüye Verildiği Tarih : 4 Mayıs 2009

Tezin Savunulduğu Tarih : 4 Haziran 2009

**Tez Danışmanı : Yrd. Doç. Dr. Sanem SARIEL TALAY
(İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. A. Coşkun SÖNMEZ(YTÜ)
Yrd. Doç. Dr. D. Turgay
ALTILAR(İTÜ)**

HAZİRAN 2009

İSTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

**SENSOR NETWORK DEPLOYMENT WITH ROBOTS
FOR TERRAIN COVERAGE**

**M.Sc. Thesis by
Meral CAMCI**

**Date of submission : 4 May 2009
Date of defence examination: 4 June 2009**

**Supervisor (Chairman) : Assist. Prof. Sanem SARIEL TALAY
(ITU)
Members of the Examining Committee : Prof. Dr. A. Coşkun SÖNMEZ(YTÜ)
Assist. Prof. D. Turgay
ALTILAR(ITU)**

JUNE 2009

ÖNSÖZ

Lisans eğitimimde İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği bölümünde okurken Coşkun Sönmez'den aldığım Yapay Zeka dersinin ardından ilgi duymaya başladığım robotik konularının, beni bu derece maceralı bir akademik yolculuğa çıkaracağını tahmin edemezdim. Bir konu ile ilgili çalışırken somut bilgiye ve bu bilginin görselleştirilerek daha kesin sonuçlar üretmesine verdiğim önemin, benim o konuya tam olarak hakim olmamı sağlamasından yola çıkarak İstanbul Teknik Üniversitesi Bilgisayar Bilimleri Yüksek Lisans eğitimimde robotik konularıyla ilgili çalışmaya başladım. Bu tez çalışması kapsamında da “Ortam Tarama İçin Robotlarla Duyarga Ağı Konumlandırma” için ihtiyaç duyulan yöntemleri ve yaklaşımları araştırarak bu problemin çözümü için en iyi çözümü üreten bir uygulama geliştirmeyi hedefledim.

Sürekli yeni fikirler üretme ve geliştirme ortamının problemlerini zaman kaybetmeden çözmeye zorlayan, bu çalışmada verdiği destek için tez danışmanım Sanem Sariel Talay'a teşekkür ediyorum..

Bana gösterdikleri anlayış ve verdikleri destekleri ile unutmayacağım işyeri arkadaşlarım iyiki benimlesiniz..

Eskiden kendimden küçük kardeşler olarak baktığım ama zamanla arkadaş olduğum sevgili kardeşlerim Alper ve Burçin, sizlerin desteğiniz, anlayışınız ve güler yüzünüz yüzüme hep bir tebessüm oldu..

Ve yeniden; Bugüne kadarki hayatımda yaratabilmeyi başardığım ne kadar artı değer ve güzellik varsa bunların tamamını borçlu olduğum sevgili babam Tacettin Camcı, annem Nurşen Camcı. Sizler daima benimlesiniz...

Haziran 2009

Meral CAMCI

Bilgisayar Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR	ix
ÇİZELGE LİSTESİ	x
ŞEKİL LİSTESİ	xi
ÖZET	xiii
SUMMARY	xv
1. GİRİŞ	1
2. ROBOTLARLA ORTAM TARAMA YÖNTEMLERİ	3
2.1 Ortam Taramada Değerlendirilecek Hususlar	3
2.1.1 Ortam bilgilerinin bulunup bulunmamasına göre ortamın taranması.....	3
2.1.1.1 Çevrimdışı ortam tarama.....	4
2.1.1.2 Çevrimiçi ortam tarama	5
2.1.2 Tarama işlemini gerçekleştiren özneye göre ortamın taranması.....	7
2.1.2.1 Robot donanımı yardımıyla ortam tarama	7
2.1.2.2 Duyarga temelli ortam tarama.....	8
2.1.3 Tarama işleminde kullanılan robot sayısına göre ortamın taranması	9
2.1.3.1 Tek robotun ortamı taraması	9
2.1.3.2 Çok sayıda robotun ortamı taraması	10
2.2 Ortam Tarama Yöntemleri	11
3. PLAYER/STAGE BENZETİM ORTAMI	13
3.1 Player Yazılımı Arayüzleri	14
3.1.1 Player arayüzü	14
3.1.2 Position2d arayüzü	14
3.1.3 Simulation arayüzü	14
3.1.4 Graphics2d arayüzü	14
3.1.5 Opaque arayüzü	15
3.1.6 Sonar arayüzü	15
3.1.7 Speech arayüzü	15
3.2 Stage Yazılımı ve Bileşenleri.....	15
4. LEAST RECENTLY VISITED YÖNTEMİ	17
4.1 LRV Yöntemi.....	17
4.1.1 Düğüm döngüsü.....	17
4.1.2 Robotun güncel düğümün menzili dışında bulunması durumu	18
4.1.3 Robotun güncel düğümün menzilindeyken engelle karşılaşması durumu	20

4.1.4 Robotun güncel düğümün menzilineyken gittiği yönde engel bulunmaması durumu	21
4.2 LRV Yöntemi Detayları	21
4.3 LRV Yöntemi Gerçekleme Detayları	25
5. LEAST RECENTLY VISITED YÖNTEMİNİN GELİŞTİRİLMESİ	27
5.1 E-LRV Yöntemi	27
5.1.1 Robotun güncel düğümün menzili dışında bulunması durumu	28
5.1.2 Robotun güncel düğümün menzilineyken engelle karşılaşması durumu.....	31
5.1.3 Robotun güncel düğümün menzilineyken gittiği yönde engel bulunmaması durumu	31
5.2 E-LRV Yöntemi Gerçekleme Detayları	33
5.2.1 Genel kavramlar	33
5.2.2 Player/Stage benzetim ortamında kullanılan arayüzler.....	34
5.2.2.1 Player arayüzü	34
5.2.2.2 Position2d arayüzü	34
5.2.2.3 Simulation arayüzü.....	34
5.2.2.4 Graphics2d arayüzü.....	34
5.2.2.5 Opaque arayüzü.....	34
5.2.2.6 Sonar arayüzü	35
5.2.2.7 Speech arayüzü.....	35
5.2.3 Kullanılan veri yapıları	36
5.2.3.1 Robot ve düğümde ortak olarak kullanılan veri yapıları.....	36
5.2.3.2 Düğüm tarafında kullanılan veri yapıları	37
5.2.3.3 Robot tarafında kullanılan veri yapıları	38
5.3 E-LRV Yöntemi Bileşenleri.....	40
5.3.1 Engel tanımı ve sakınımı	40
5.3.1.1 Büyük engel tanımı	41
5.3.1.2 Engel sakınımı.....	43
5.3.2 Düğüm bırakma	44
5.3.3 Mesaj tipleri ve mesaj akışı	49
5.3.3.1 Robot düğüm haberleşmesi	49
5.3.3.2 Mesaj tipleri	51
5.3.3.3 Robot tarafında mesaj tipinin belirlenmesi	55
5.3.3.4 Düğüm tarafında mesaj tipinin belirlenmesi	56
6. DENEYSEL SONUÇLAR	59
7. SONUÇLAR VE ÖNERİLER.....	65
KAYNAKLAR.....	67
ÖZGEÇMİŞ.....	69

KISALTMALAR

LRV	: Least Recently Visited
E-LRV	: Enhanced - Least Recently Visited
STC	: Spanning - Tree Based Coverage
MSTC	: Multi - Robot Spanning Tree Coverage
MFC	: Multi - Robot Forest Coverage

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1 : Ortam tarama konusunda incelenen yöntemler.	12
Çizelge 4.1 : Çapraz yön seçme kuralı.	24
Çizelge 4.2 : Çizgi yön seçme kuralı.	25
Çizelge 4.3 : Daire yön seçme kuralı.	25
Çizelge 5.1 : Sonar duyurga numaraları ve açısal değerleri.	44
Çizelge 5.2 : E-LRV'nin gerçekleşmesinde robot ve düğüm tarafında kullanılan port, istemci ve arayüz bilgileri.	51
Çizelge 5.3 : LRV ve E-LRV'nin gerçekleşmesinde robot ve düğüm tarafında karşılıklı yollanan mesaj tipleri.	51
Çizelge 6.1 : Engelsiz ortamda 5'er kere çalıştırılan LRV ve E-LRV'ye ait sonuçlar.	59
Çizelge 6.2 : %10 engelli 5 farklı ortam için 5'er kere çalıştırılan LRV ve E-LRV ye ait ortalama değerleri içeren sonuçlar.	60
Çizelge 6.3 : %10 engelli 5 farklı ortam için 5'er kere çalıştırılan LRV ve E-LRV ye ait ortalama değerleri içeren sonuçlar.	60
Çizelge 6.4 : Heterojen dağılıma sahip %15 engelli 3 farklı ortam için 5'er kere çalıştırılan LRV ve E-LRV ye ait ortalama değerleri içeren sonuçlar.	63

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Çevrimdışı STC.....	4
Şekil 2.2 : Ortamın parçalara ayrılması.....	5
Şekil 2.3 : Süpürme yöntemiyle ortam tarama.....	5
Şekil 2.4 : Çevrimiçi STC.....	6
Şekil 2.5 : MFC.....	7
Şekil 2.6 : Çevrimdışı MSTC.....	10
Şekil 3.1 : Player/Stage mimarisi.....	13
Şekil 4.1 : LRV yöntemi adımları.....	22
Şekil 4.2 : LRV yönteminde kullanılan temel yön bilgileri ve yön seçme kuralları.....	23
Şekil 4.3 : LRV yönteminde kullanılan çapraz, çizgi ve daire yön seçme kuralları.....	23
Şekil 4.4 : Yön seçme kuralı için kullanılacak isimlendirme.....	24
Şekil 5.1 : E-LRV yöntemi yön bilgisi.....	33
Şekil 5.2 : Sonar duyargalarıyla donatılmış robotun 1 yönünde hareketi.....	33
Şekil 5.3 : Player arayüzlerinin kullanımının gösterimi.....	35
Şekil 5.4 : Robot ve düğüm tarafında ortak olarak kullanılan Nokta Yapısı.....	36
Şekil 5.5 : Robot ve düğüm tarafında ortak olarak kullanılan Temel Düğüm Yapısı.....	36
Şekil 5.6 : Düğüm tarafında kullanılan Düğüm Bilgisi Yapısı.....	37
Şekil 5.7 : Düğüm tarafında kullanılan Düğüm Yapısı.....	38
Şekil 5.8 : Robot tarafında kullanılan Robot Düğüm Bilgisi Yapısı.....	39
Şekil 5.9 : Robot tarafında kullanılan Düğüm Listesi Yapısı.....	39
Şekil 5.10 : Robot tarafında kullanılan diğer veri yapıları.....	40
Şekil 5.11 : Engel tanımı: statik nesnelere ve ortam sınırları (duvar).....	40
Şekil 5.12 : Büyük engel.....	41
Şekil 5.13 : Büyük engel değerlendirme detayı.....	42
Şekil 5.14 : Engel sakınımı algoritması akışı.....	43
Şekil 5.15 : Düğüm bırakma.....	45
Şekil 5.16 : Düğüm bırakma ön işlemi algoritması akışı.....	46
Şekil 5.17 : LRV ve E-LRV düğüm bırakma stratejileri.....	48
Şekil 5.18 : Robottan düğüme mesaj gönderilmesi.....	50
Şekil 5.19 : Düğümden robota mesaj gönderilmesi.....	50
Şekil 5.20 : Mesaj tipleri.....	52
Şekil 5.21 : Konum-Mesajı ve Konum-Mesajı-Cevabı Akışı.....	52
Şekil 5.22 : Periyodik-Bilgi-Mesajı ve Periyodik-Bilgi-Mesajı-Cevabı Akışı.....	53
Şekil 5.23 : Yön-Güncelleme-Mesajı ve Yön-Güncelleme-Mesajı-Cevabı Akışı.....	54

Şekil 5.24 : Engel-Güncelleme-Mesajı ve Engel-Güncelleme-Mesajı-Cevabı Akışı.	54
Şekil 5.25 : Robot tarafında mesaj tipinin belirlenmesi akışı.	55
Şekil 5.26 : Düğüm tarafında mesaj tipinin belirlenmesi akışı.	57
Şekil 6.1 : E-LRV yönteminin engelsiz ortamda çalıştırılması.	61
Şekil 6.2 : E-LRV yönteminin %20 engelli ortamda çalıştırılması.	62
Şekil 6.3 : LRV yönteminin %20 engelli ortamda çalıştırılması.	62
Şekil 6.4 : E-LRV yönteminin heterojen dağılımlı %15 engelli ortamda çalıştırılması.	64
Şekil 6.5 : LRV yönteminin heterojen dağılımlı %15 engelli ortamda çalıştırılması.	64

ORTAM TARAMA İÇİN ROBOTLARLA DUYARGA AĞI KONUMLANDIRMA

ÖZET

Bu çalışmada çeşitli robot uygulamaları (çim biçme, boyama, temizleme, mayın temizleme ve arama kurtarma işlemleri vb.) ve duyarga temelli ortam izleme uygulamalarında karşılaşılan ortam tarama problemi incelenmekte ve etkin bir çözüm önerilmektedir. Alan taraması bazı mayın temizleme ve arama kurtarma işlemleri gibi gerçek zamanlı uygulamalar için kritik bir temel problem olup, çim biçme, boyama, temizleme gibi günlük uygulamalar için de etkin olarak yürütülmesi gerekmektedir. Ortam tarama yöntemleri ortamın etkin bir şekilde taranmasında önemli role sahiptir. Bu çalışmada robotlarla duyarga temelli bir şekilde ortamın taranmasını sağlayan bir yöntem olan LRV (Least Recently Visited) yönteminin geliştirilmesiyle elde edilen E-LRV (Enhanced - Least Recently Visited) yöntemi anlatılmaktadır. LRV ve E-LRV duyarga düğümlerini artımlı bir şekilde bilinmeyen bir ortama bırakan tekli robot ya da çoklu robot sistemi ile bilinmeyen bir ortamı tarama işlemini gerçekleştirmek üzere tasarlanmıştır. LRV yönteminde ortamı tarayan robot, bir sonraki ziyaret edeceği yer ve gideceği yön konusunda ortama bırakılan duyargalar ve bu duyargaların önerilerini dikkate alarak karar verir. Ortam taramada duyargaların kullanılması, ortam bilgilerinin önceden bilinmesi esasına dayanan çevrimdışı ortam tarama yöntemine ihtiyaç duyulmamasını sağlar. Bilinmeyen ortamın LRV yöntemiyle taranması sırasında ortamdan elde edilen tüm bilgilerin kullanılmamasından kaynaklanan ve ortamın tarama süresinde verimsizliğe yol açan bir durum oluşmaktadır. E-LRV yöntemi LRV yönteminin bu konudaki eksikliklerini gidermek üzere tasarlanmış olup, ortamdan elde edilen bilgilerin en kısa zamanda değerlendirileceği ve ortamın taranması sırasında doğru bilgilerin kullanılmasını sağlayan bir düğüm bırakma stratejisi önermektedir. Bu stratejiye göre ortama düğüm bırakma, ortamın engel durumu değerlendirilerek gerçekleştirilir. Engel durumları için düğümlerde ek yön artırımı işlemi gerçekleştirilmektedir. Ayrıca düğüm bırakılırken robotun geçtiği yönlerin güncellenmesiyle de ortam bilgileri etkin bir şekilde kullanılmış olup elde edilen bilgi en kısa zamanda değerlendirmeye alınmaktadır. Yapılan bu ek yön artırımları ortamın tamamının taranması düşünüldüğünde mesaj trafiğini azaltıcı bir etki yapmıştır. Player/Stage benzetim ortamı kullanılarak yapılan simülasyonlarda E-LRV yönteminin LRV yöntemine göre ortamı daha kısa sürede taradığı görülmüştür.

SENSOR NETWORK DEPLOYMENT WITH ROBOTS FOR TERRAIN COVERAGE

SUMMARY

This work presents the coverage problem which is emerged for both robot applications (e.g., lawn moving, painting, cleaning, mine-removal and search and rescue operations) and sensor-based environment monitoring applications. Efficient area coverage is critical in some of these real-world applications. (e.g., mine-removal and search and rescue operations) and frequently used in daily life practical applications (e.g., lawn moving, painting, cleaning). Coverage criteria are very important for efficient coverage. For this reason E-LRV (Enhanced - Least Recently Visited), an improved version of an existing algorithm LRV (Least Recently Visited) for sensor-based robotic terrain coverage is presented in this work. LRV and E-LRV are both proposed for either a single robot or multi-robot team to cover an unstructured environment by incremental deployment of sensor nodes. The robot employing LRV makes a decision on the next unvisited area and the movement direction by means of the deployed sensors and their suggestions. Using sensors for coverage eliminates the need for mapping the environment which is needed in offline coverage. The information gained from the environment is not used efficiently when considering the coverage of the unknown environment using LRV. E-LRV is designed for the solution to this problem and evaluates the information gained from the environment in an efficient manner when deploying nodes. According to this node deployment strategy E-LRV considers the obstacle situations in unstructured environments. For these obstacle situations additional direction updates are made on sensor nodes. When deploying a node to the environment the traversed directions for the deployed node are also updated. This leads the usage of the information in an efficient manner without waiting further steps and provides accurate information through coverage. Additional direction updates during the node deployment phase decreases the total message traffic when considering the overall coverage. Simulations are implemented using Player/Stage simulator and the results after these simulations show that E-LRV complete the coverage mission in a shorter time when compared with LRV.

1. GİRİŞ

Robot bilimi her geçen gün gelişen ve kullanım alanı giderek genişleyen bir alan olup robot uygulamaları bir çok gerçek yaşam probleminin çözümünde etkinlik sağlamaktadır. Endüstriyel alanda ve hizmet sektöründe robotlar sıklıkla kullanılmakta ve insanların erişemeyeceği, yapamayacağı işleri kolaylıkla yaparak işlerini kolaylaştırmaktadır. Robotlar genelde endüstriyel alanda kullanılmaktadır. Sanayi robotları boyama, montaj, kaynak, parça yükleme-boşaltma işlemlerini gerçekleştirirken ve otomotiv, kimya, beyaz eşya endüstrilerinde kullanılmaktadır. İnsan kolunun ve eklemelerinin hareketlerinin modellenmesi esas alınarak tasarlanan bu robotlar insanların hareketlerini oldukça genişletebilmektedir. Hassasiyet, doğruluk ve güç gerektiren işleri kolaylıkla yerine getiren robotlar, aynı zamanda hızlı çalışmaları sayesinde insanların işlerini kolaylaştırmaktadır.

Hastanelerde kullanılan robotlar doktorların doğru ve hızlı bir şekilde teşhis koymalarına yardımcı olmaktadır.

Askeri alanda kullanılan robotlar hedef bulma-odaklanma ve yok etme, dost-düşman tanıma, keşif işlemlerinde kullanılmaktadır. Mayın arama cihazları, insansız hava araçları, aviyonik sistemler bu sınıfta değerlendirilmektedir.

Bilimsel araştırmalarda önemi gittikçe artan ve gelişen teknolojiyle birlikte kullanımı da artan robotlar uzay araştırmaları, yer bilimi (deprem hareketleri), su bilimi (okyanus derinliğindeki araştırmalar) alanlarında kullanılmaktadır.

Ortam tarama pek çok gerçek zamanlı uygulamada kullanılan ve kullanım alanı giderek yaygınlaşan bir problemdir. Belirli bir görevi yerine getirmek üzere kullanılan robotların belirli bir alanı gezerek (kapsayarak) görevlerini tamamlaması esasına dayanır. Ortam tarama günlük uygulamalarda (çim biçme, boyama, temizleme) kullanılabilirdiği gibi gerçek zamanlı ve kritik işlemlerde (mayın temizleme, arama-kurtarma) de kullanılabilir. Bu durumda ortamın etkin ve hatasız bir şekilde taranması gerekmektedir. Etkin tarama, zaman kritik işlemlerin

belirli bir zaman kısıtı altında gerçekleştirilmesi, işlemlerin yarıda kesilmemesi ve ortam hakkında doğru bilgilerin elde edilmesiyle sağlanmaktadır.

Bu çalışmada, gezgin (mobil) robotlar kullanılarak ortam tarama probleminin çözümüne yönelik geliştirilmiş olan Least Recently Visited (LRV) (Batalin ve Sukhatme (2007)) yöntemi tanıtılmakta ve bu yöntemin geliştirilmiş hali olan Enhanced Least Recently Visited (E-LRV) yöntemi sunulmaktadır.

Tezin bölümleri konularına göre şu şekilde organize edilmiştir:

Bölüm 2’de, robotlarla ortam tarama yöntemleri tanıtılmaktadır. Bu bölümde ortam taramada değerlendirilecek hususlar sunulmakta ve bu zamana kadar yapılan çalışmaların ortam taramada değerlendirilecek hususlar dikkate alınarak incelenmesi yer almaktadır.

Bölüm 3’de, bu tez çalışması kapsamında incelenen LRV (Batalin ve Sukhatme (2007)) yönteminin geliştirilmesiyle oluşan E-LRV yönteminin gerçekleştirilmesi için kullanılan Player/Stage’ benzetim ortamının (Gerkey ve diğ., 2001), (Gerkey ve diğ., 2003), (Url-1), tanıtımı ile gerçekleştirme sırasında kullanılan özellik ve bileşenleri yer almaktadır.

Bölüm 4’te, bu tez çalışması kapsamında incelenen LRV yöntemi ve detaylarının anlatımı yer almaktadır.

Bölüm 5’te, incelenen LRV yönteminin geliştirilmesiyle oluşturulan E-LRV yöntemi, LRV yöntemiyle farklılıkları ve gerçekleştirme detayları yer almaktadır.

Bölüm 6’da, LRV ve E-LRV yöntemlerinin Player/Stage benzetim ortamında, gerçekleştirilmesine ilişkin deneysel sonuçlar yer almaktadır.

Bölüm 7’de bu tez kapsamındaki çalışmada elde edilen sonuç ve önerilere yer verilmiştir.

2. ROBOTLARLA ORTAM TARAMA YÖNTEMLERİ

Ortam tarama, gezgin (mobil) robotlar tarafından gerçekleştirilen, pek çok gerçek zamanlı uygulamada kullanılan ve kullanım alanı giderek genişleyen bir çalışma alanıdır. Ortam tarama tek ya da çok sayıda robotun sınırları belirli ve çoğunlukla engeller içeren bir alanı, sahip oldukları donanım (duyarga ve eyleyici) bilgilerini değerlendirerek, gezerek taraması şeklinde tanımlanır (Gage, 1991).

Ortam tarama yöntemleri, uygulanan stratejiye göre farklı sınıflar altında gruplanarak bundan sonraki bölümlerde ayrıntılı olarak açıklanmaktadır.

2.1 Ortam Taramada Değerlendirilecek Hususlar

Bu bölümde ortam tarama işleminin ortam hakkındaki bilginin ortamı tarayan robota sağlanıp sağlanmamasına göre, ortamın taranmasını sağlayan özneye göre ve ortamı tarayan robot sayısına göre üç temel gruba ayrılarak incelenmesi yer almaktadır.

2.1.1 Ortam bilgilerinin bulunup bulunmamasına göre ortamın taranması

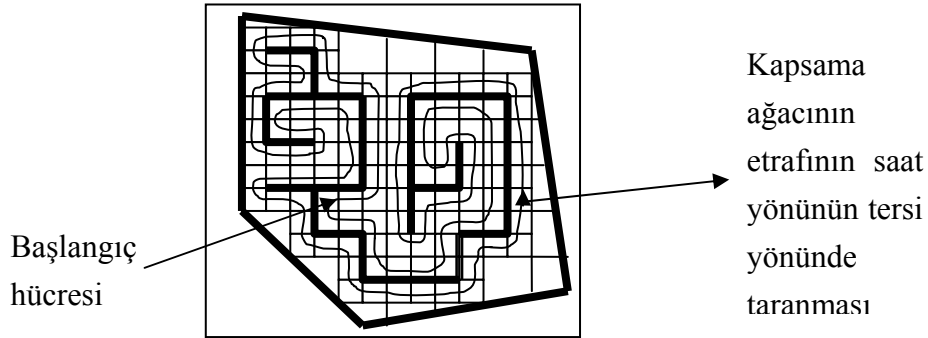
Choset (2001) robotik biliminde ortam tarama yöntemleri ile ilgili çok geniş çalışma yapmış ve ortam bilgilerinin ortamı tarayan robota sağlanıp sağlanmamasına göre ortam tarama yöntemlerini, çevrimiçi (online) ve çevrimdışı (offline) olmak üzere ikiye ayırarak incelemiştir.

Gabriely ve Rimon (2001)'de ortamın tek robotla çevrimiçi ve çevrimdışı bir şekilde taranması için geliştirilen Spanning Tree Coverage (STC) anlatılmaktadır. STC bütün düğümleri ve bu düğümlerin arasındaki bütün ya da bazı kenarları içeren bağlı ve yönsüz bir çizgedir. Bu yöntemde robot $2D \times 2D$ büyüklüğe sahip hücrelerden oluşan bir alanı, üzerinde bulunan D büyüklüğünde bir araç yardımıyla taramaktadır.

2.1.1.1 Çevrimdışı ortam tarama

Choset (2001)' in yaptığı araştırmaya göre çevrimdışı ortam tarama yapan robot, tarama yaptığı ortamın global bilgileri ya da haritası kendisine sağlandığı için tarama işleminde bu bilgileri veya haritayı kullanır. Çevrimdışı ortam tarama yöntemi, tarama yapılacak ortamın bilgileri gerçek hayatta ve gerçek hayata geçirilebilen uygulamalarda önceden sağlanamadığı için her zaman uygulanabilen bir yöntem değildir. Robot tarayacağı ortam hakkında ön bilgiye sahipse gideceği yolu planlayabilir ya da ortamı tararken bu bilgiyi kullanabilir.

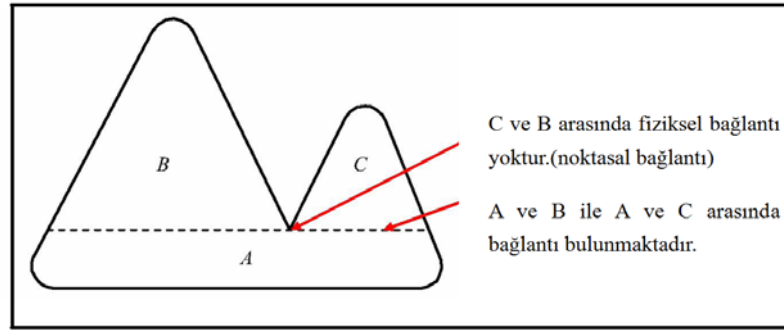
Gabriely ve Rimon (2001)'de açıklanan çevrimdışı STC yönteminde ortamın geometrik tanımı 2D'lik hücrelere çevrilmiş bir şekilde robota sağlanmış olup, parçalı olarak kapsanan hücreler çıkarılmıştır. Bu yöntem 2D büyüklüğündeki her hücrenin ortasında bir düğüm tanımlanması ve bitişik olan düğümleri birleştiren kenarların ve bu düğümlerin bir kapsama ağacı oluşturacak şekilde robota sunulması esasına dayanır. Robot belirli bir düğümün alt hücresinden başlayarak komşu düğümlere (alt hücrelere) geçer, saat yönünün tersi yönünde hareket ederek başlangıçta kullanılan alt hücreye gelene kadar bütün düğümleri kapsayacak şekilde kapsama ağacının etrafını tarar (Şekil 2.1).



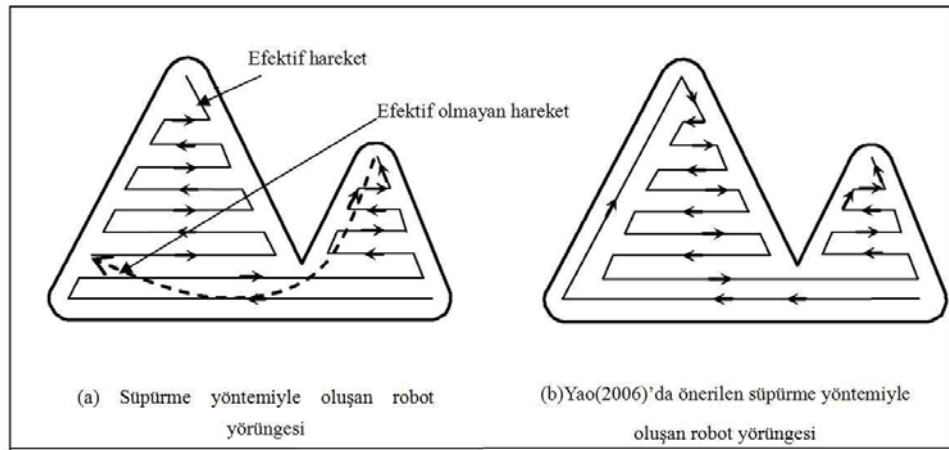
Şekil 2.1 : Çevrimdışı STC.

Yao (2006)'da ortam bilgilerine sahip olan robotun Choset (2000) ve Huang (2001)'de açıklanan süpürme yöntemiyle paralel yörüngeler izleyerek ortamı taraması verilmiştir. Bu yöntem bilinen ortamın küçük alt alanlara ayrılarak verimsizliğe neden olan fazladan yol katedilmesini önleyen bir yöntemdir. Doğrudan bağlantılı alanlar ve aralarında bağlantı bulunmayan alanların belirlenmesi için paralel çizgiler çekilerek bir çizge oluşturulur. Bu yöntemde öncelikle bağlantılı

alanların taranması ardından noktasal bağlantılı (aralarında fiziksel bağlantı bulunmayan) alanların taranması temel alınmıştır (Şekil 2.2, Şekil 2.3).



Şekil 2.2 : Ortamın parçalara ayrılması.



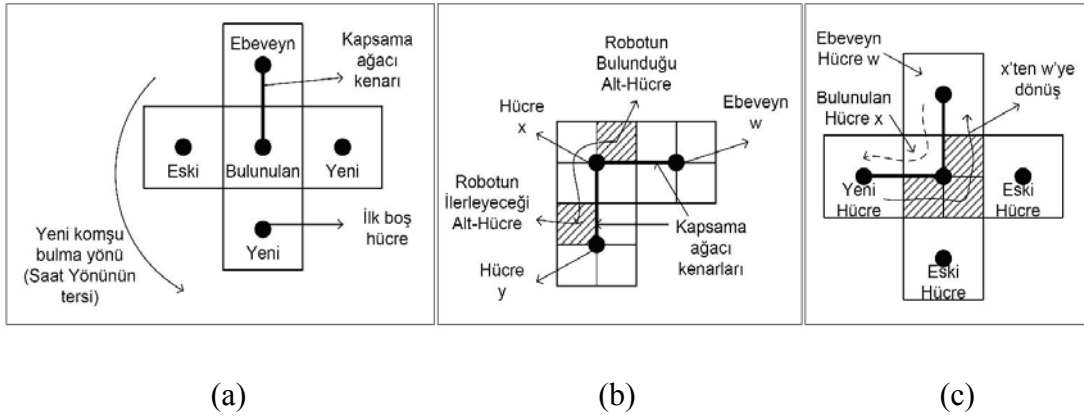
Şekil 2.3 : Süpürme yöntemiyle ortam tarama.

2.1.1.2 Çevrimiçi ortam tarama

Choset (2001)'in yaptığı araştırmaya göre çevrimiçi ortam taraması yürüten robotlar, global ortam bilgileri ya da haritasına sahip değildir. Robot, ortamı tarama işleminde hareket ettikçe aynı zamanda keşif de yapmaktadır. Ortamı tarayan robota herhangi bir ön bilginin sağlanmadığı bu yöntemde, robot karşısına çıkan engellerden sakınma ve çarpışma (çok sayıda robotun kullanıldığı ortam tarama işleminde gereklidir) durumunu engelleme işlemlerini adım adım planlayarak gerçekleştirir.

Çevrimiçi yöntem, gerçek hayata ve gerçek hayata geçirilebilen uygulamalara daha uygun bir yöntem olduğundan çevrimdışı yöntemle göre daha uygulanabilir bir yöntemdir.

Gabriely ve Rimon (2001)'de açıklanan çevrimiçi STC yönteminde robotun ortam hakkında herhangi bir bilgisi yoktur. Bu yöntemde robotun ortamda hangi $2D$ 'lik hücrede bulunduğunu anlaması için kullanılan konum ve yönlendirme duyargası bulunmaktadır. Ayrıca robotun bulunduğu hücreye komşu olan 4 hücrede engel bulunup bulunmadığını anlaması için kullanılan duyarga yer almaktadır. Bu yöntemde robot ortamdaki $2D \times 2D$ 'lik hücreleri D büyüklüğünde bölümlere ayırarak ve bu bölünen bölümler aracılığıyla ortamı taramaktadır. Robotun bulunduğu hücredeyken komşu hücrede (saat yönünün tersi yönüne göre belirlenen) engel bulunup bulunmamasına göre o hücreye olan bağlantıyı kapsama ağacı kenarı olarak eklemesiyle kapsama ağacı oluşturulur.



Şekil 2.4 : Çevrimiçi STC.

Bu yöntemde robotun bulunduğu hücrede engel bulunmayan bir komşu hücre bulunarak, robotun bulunduğu hücrede engelsiz komşu hücreye bir kapsama ağacı kenarı tanımlanır (Şekil 2.4 (a)). Robotun bulunduğu hücrenin alt hücrelerinden, bulunan yeni komşu hücrenin alt hücrelerine geçiş, tanımlanan bu kenarın sağından (saat yönünün tersine göre) gerçekleşir (Şekil 2.4 (b)). Bu şekilde başlangıç hücrelerine gelinene kadar artırımlı bir şekilde kapsama ağacı oluşturulur. Başlangıç hücrelerine dönüş yine kapsama ağacının kenarının sağından gerçekleştirilir (Şekil 2.4 (c)).

Batalin ve Sukhatme (2003a)'da yapılan çalışmada ortamı tarayan tek robota herhangi bir şekilde ortam bilgisi ya da haritası sağlanmamaktadır. Bu yöntemde robot çevrimiçi bir şekilde ortamdaki duyargalarla haberleşerek ve kendi donanımı yardımıyla edindiği bilgiyi sentezleyerek tarama işlemini gerçekleştirmektedir.

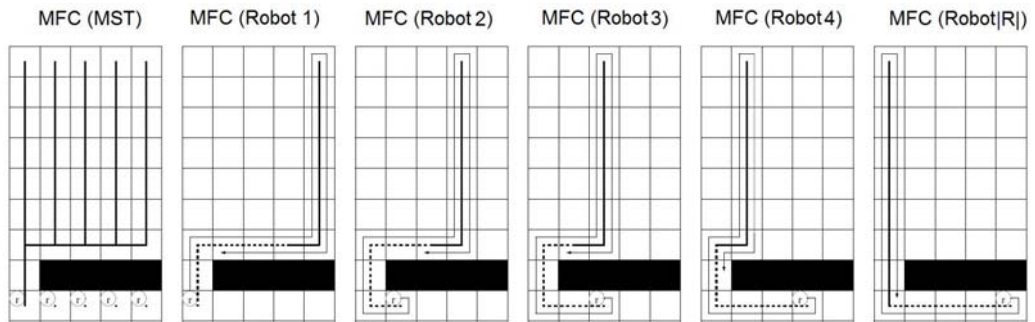
Hazon ve diğ. (2006)'da açıklanan çevrimiçi MSTC (Multirobot Spanning Tree Coverage) yönteminde birden fazla robotun herhangi bir şekilde ortam bilgisine sahip olmaksızın sadece ilk konumlarını bilmesi esasına dayalı bir tarama işlemi gerçekleştirilmektedir. Oluşturulan kapsama ağacı robotlara lokal olarak tarayacakları kapsama ağaçları şeklinde bölüştürülür. Robotlar, diğer robotların ortamı tararken kullandıkları kapsama ağacına nereden bağlanacağı bilgisini de bildikleri için herhangi bir robotun tarama işlemine devam edememesi durumunda onların görev alanlarını tarama işlemini üstlenerek çalışmalarına devam ederler.

2.1.2 Tarama işlemini gerçekleştiren özneye göre ortamın taranması

Bu kısımda ortam tarama işlemini gerçekleştiren robotun sadece kendi üzerinde bulunan donanım yardımıyla ortamı taraması yöntemi ile robotun ortamda bulunan duyargalarla etkileşim halinde bulunarak ortamı taraması yöntemleri anlatılmaktadır (Choset, 2001).

2.1.2.1 Robot donanımı yardımıyla ortam tarama

Ortam tarama işleminde gerçekleştirilen işlem ya da işlemler dizisi robotun sadece kendi sahip olduğu donanım bilgileri kullanılarak gerçekleştirilmektedir. Robot kendi üzerinde bulunan donanım yardımıyla ortam hakkında bilgi edinir ve bu bilgiyi kullanarak tarama işleminde tamamlaması gereken işlemleri tamamlar.



Şekil 2.5 : MFC.

Zeng ve diğ. (2005)'te verilen Multirobot Forest Coverage (MFC) yönteminde çok sayıda robotun çevrimdışı bir şekilde ortamı taraması anlatılmaktadır. Bu yöntem STC (Gabriely ve Rimon, 2001) yönteminin çok sayıda robota uygulanmış şeklidir.

Her robot için ayrı bir ağaç yapısı tanımlanarak robotların bu ağaçları taramasıyla ortamın tamamının taranması sağlanmaktadır (Şekil 2.5).

2.1.2.2 Duyarga temelli ortam tarama

Duyarga temelli ortam tarama, robotun ortamda bulunan statik ya da dinamik bir şekilde yerleştirilmiş duyargaların robota sağladığı bilgiler ile robotun kendi sahip olduğu donanım bilgilerini birlikte değerlendirerek ortam tarama işlemini gerçekleştirmesidir (Choset, 2001). Duyarga temelli ortam tarama, ortam bilgilerinin robota sağlanmadığı çevrimiçi ortam tarama esasına dayanmaktadır. Bu yöntemde duyargalar robotun ortamı tararken gerçekleştireceği işlem ya da işlemler dizisinde kullanacağı bilgileri robota sağlamak ve bu bilgileri alan robot, kendi donanımının sağladığı bilgilerle, duyargaların sağladığı bilgiyi değerlendirerek tamamlaması gereken görevi tamamlamaktadır.

Duyarga temelli ortam tarama, robotun kendi donanım bilgisi ve duyargaların robota sağladığı bilginin, ortamı tarama işlemini gerçekleştiren robota sağlanması esasına dayanır. Bu yüzden bu yöntemde en iyi çözüme ulaşmak duyargaların özellikleri ve çeşitliliği ile robotun donanımının özellikleri ve çeşitliliği yüzünden her zaman mümkün olmamaktadır.

Duyarga temelli ortam tarama yöntemleri duyargaların ortamda konumlandırılma zamanına göre statik (Batalin ve Sukhatme, 2002a) ve dinamik (Batalin ve Sukhatme, 2007) olarak iki temel başlık altında incelenebilir (Choset, 2001).

Statik Ortam Tarama:

Statik ortam taramada (Batalin ve Sukhatme, 2002a) ortamı tarayan robotlar ya da ortamda robotun etkileşimde bulunduğu duyargalar tarama işlemi gerçekleştirilmeden önce ortama yerleştirilir ya da belirli bir düzende ortamda konumlandırılır. Statik ortam taramada ortamın tamamı her an robotların ya da robotun etkileşimde bulunduğu duyargalarının kapsama alanında bulunur. Bir ortamın tamamının statik bir şekilde taranması için robotların ya da robotun etkileşimde bulunduğu duyargaların sayısının belirli bir değerden fazla olması gerekir. Bu değer robotların kendi sahip olduğu duyargaların ve etkileşimde bulunduğu duyargaların menziline ve ortamın büyüklüğüne göre sayısı tam olarak

belirlenemeyen bir deęer olduęu açıklanmıştır (Batalin ve Sukhatme, 2007). Ortamın önceden bilinmedięi çevrimiçi durumda, alan taramada kullanılacak robot ya da robotun etkileşimde bulunduęu duyarga sayısının belirlenmesi imkansızdır (Batalin ve Sukhatme, 2007).

Dinamik Ortam Tarama:

Dinamik ortam tarama, tarama işlemini gerçekleştirecek robotun etkileşimde bulunacaęı duyargaların zaman içinde ortama yerleştirilmesi esasına dayanır (Batalin ve Sukhatme, 2007). Dinamik ortam tarama esasına dayanan Least Recently Visited (LRV) yönteminde herhangi bir şekilde duyargaların taranacak ortama önceden yerleştirilmesi ya da yerleştirilecekleri yerlerin önceden bilinmesi mümkün değildir (Batalin ve Sukhatme, 2007). Dinamik ortam taramada, tarama işlemini gerçekleştiren robot, o anda ortamda bulunan duyargaların sağladığı bilgileri kullanarak önceden bilmedięi bir güzergahta sabit hızla ilerleyerek, her adımda (tarama işleminin döngüsü) yeni bir duyargayı taradığı ortama bırakır. Bırakılan bu yeni duyarga bir sonraki adımda robota ortam hakkında bilgi verir. Robotun bu bilgi ve ortamdaki diğer duyargaların bilgileri ile kendi donanımının sağladığı bilgiyi birleştirmesi ve tarama işlemini, sentezledięi bu bilgiyi kullanarak gerçekleştirmesi sağlanır.

2.1.3 Tarama işleminde kullanılan robot sayısına göre ortamın taranması

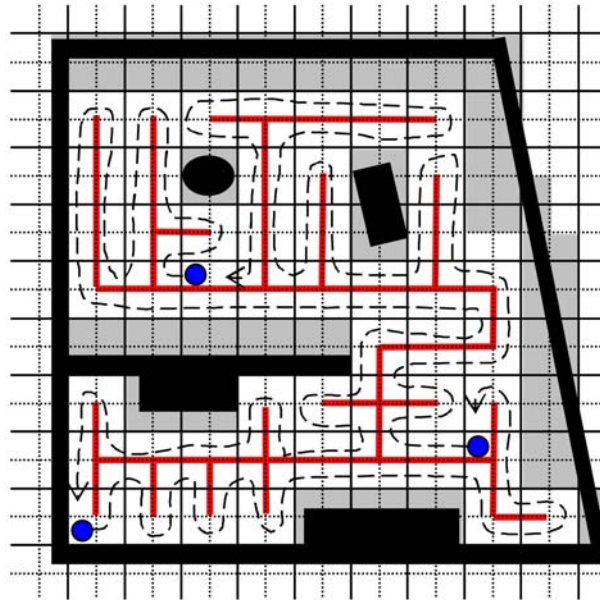
Ortam tarama, bu işlemin gerçekleştirilmesinde kullanılan robot sayısına göre tek robotun ortamı taraması ve çok sayıda robotun ortamı taraması şeklinde iki farklı gruba ayrılarak incelenebilir.

2.1.3.1 Tek robotun ortamı taraması

Bu kısımda ortam tarama işleminde tek sayıda robotun kullanılması durumu anlatılmaktadır. Ortamı tek başına tarayan robot, ortam tarama işleminde gerçekleştirilecek bütün işlemleri kendisi gerçekleştirmektedir (Yao, 2006), (Batalin ve Sukhatme, 2003a), (Batalin ve Sukhatme, 2003b). Bu yüzden ortam tarama işleminde görev paylaşımı söz konusu değildir ve ortamın taranması çok sayıda robotun ortamı taramasına göre daha uzun sürebilir.

2.1.3.2 Çok sayıda robotun ortamı taraması

Bu kısımda ortam tarama işleminde çok sayıda robotun kullanılması durumu anlatılmaktadır. Ortam tarama işleminde gerçekleştirilecek işlem ya da işlemler dizisi ortamı taramakla görevli bütün robotlara dağıtılmaktadır. Bu şekilde zaman kullanımını daha verimli bir hale getirirken yapılacak işlem ya da işlemler dizisi daha kısa zamanda tamamlanmaktadır. Aynı zamanda robotlardan herhangi bir tanesinin herhangi bir sebepten dolayı tarama işlemine devam edememesi durumunda, diğer robotlar tarama işlemine devam ettikleri için ortamın taranması için tamamlanması gereken işlem ya da işlemler dizisi daha sağlıklı bir şekilde tamamlanmaktadır (Choset, 2001).



Şekil 2.6 : Çevrimdışı MSTC.

Hazon ve Kaminka (2005)'te, çevrimdışı STC'nin (Gabriely ve Rimon, 2001) çoklu robot sistemiyle gerçekleştirilmesi yer almaktadır (Şekil 2.6). Multirobot Spanning Tree Coverage(MSTC) yönteminde ortam bilgisi kapsama ağacı şeklinde robotlara sağlanmaktadır. Robotların ilk atandıkları yerler rasgele belirlenmektedir. Bu yöntemde robotun konumlandırıldığı ilk yerden, kendinden sonraki robotun konumlandırıldığı ilk yere kadar saat yönünün tersi yönünde ortamı taraması gerçekleştirilmektedir. Herhangi bir şekilde taramaya devam edemeyen robotun görevini onun atandığı yerden önceki yeri taramak üzere atanmış robot devralmaktadır. Bu durumun haberleşme için ya robotların periyodik olarak

birbirlerine mesaj göndermesi ya da kendinden sonraki robotun başlangıç noktasına ulaşması gerekmektedir.

Bu çalışmada ayrıca robotların geriye dönüş yaptıkları ve ortam tarama işlemine devam edemeyen ya da kendisine atanan yeri tamamlayan robotların tarayacağı alanı taramaya başlamasını sağlayan backtracking MSTC ve sadece kendine atanan yeri tamamlayarak diğer robotun taramaya başlayacağı yere ulaştığında gerekirse o kısmı da tarayan geri dönüş özelliği bulunmayan non-backtracking MSTC anlatılmaktadır (Hazon ve Kaminka, 2005).

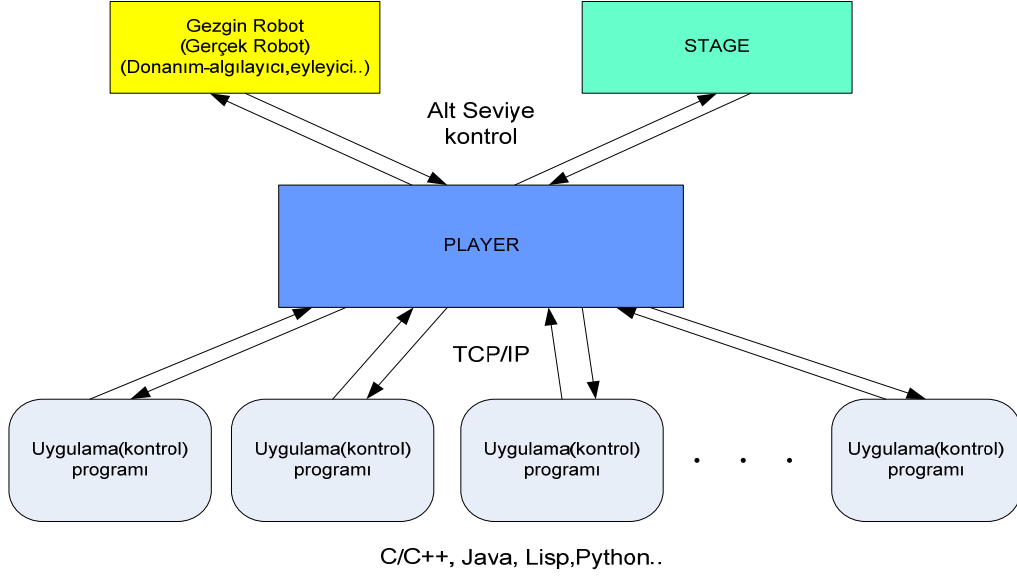
2.2 Ortam Tarama Yöntemleri

Bu tez çalışması kapsamında incelenen bazı ortam tarama yöntemleri ve bu yöntemlerin Bölüm 2.1'de anlatılan ortam taramada değerlendirilecek hususlar gözönüne alınarak değerlendirilmesi Çizelge 2.1' de özetlenmiştir.

Çizelge 2.1 : Ortam tarama konusunda incelenen yöntemler.

Yöntemin Kaynağı	Ortam Bilgilerinin Bulunup Bulunmaması		Tarama İşlemini Gerçekleştiren Özne			Tarama İşleminde Kullanılan Robot Sayısı	
	Çevrimiçi	Çevrimdışı	Robot	Duyarga Temelli		Tek Robot	Çok Sayıda Robot
				Statik	Dinamik		
Agmon ve diğ.(2006)		√	√				√
Batalin ve Sukhatme(2002a)	√			√			√
Batalin ve Sukhatme(2002b)	√				√		√
Batalin ve Sukhatme(2003a)	√				√	√	
Batalin ve Sukhatme(2003b)	√				√	√	
Batalin ve Sukhatme(2007)	√				√	√	
Gabriely ve Rimon(2001) Online STC	√		√			√	
Gabriely ve Rimon(2001) Offline STC		√	√			√	
Hazon ve Kaminka(2005) MSTC		√	√				√
Hazon ve diğ.(2006) Online MSTC	√		√				√
Latimer ve diğ.(2002)	√		√				√
Rekleitis ve diğ.(2004)	√		√				√
Yao(2006)		√	√			√	
Zheng ve diğ.(2005) MFC		√	√				√
Zheng ve Koenig(2007)		√	√				√

3. PLAYER/STAGE BENZETİM ORTAMI



Şekil 3.1 : Player/Stage mimarisi.

Bu çalışmada Player robot sunucusu ve Stage yazılım benzetim ortamı kullanılmıştır. Player/Stage yazılımı günümüzde pek çok araştırmada yaygın olarak kullanılan açık kaynak kodlu ve sürekli gelişen bir araçtır (Gerkey ve diğ., 2001), (Gerkey ve diğ. 2003), (Url-1). Player ve Stage programlarının uygulama programlarıyla ilişkileri Şekil 3.1’de gösterilmektedir. C/C++, Java, Lisp vb. programlama dillerinden herhangi bir tanesiyle yazılmış olan programlar Player sunucusu yardımıyla gezgin robot ve bu robotun sahip olduğu donanımın (örn., duyarga, eyleyici) kontrolünü sağlamaktadır. Yazılan programlar gerçek bir robot üzerinde test edilebilirken, gezgin robot benzetimine olanak sağlayan Stage programı yardımıyla da test edilebilir. Gerçek robotla yapılan çalışmalarda Player sunucusu robotla haberleşir, benzetimi yapılan gezgin robot çalışmalarında ise Player sunucusu Stage programıyla haberleşir ve gerçek robota komut yollayıp, onun donanımının sağladığı bilgileri (duyarga verileri) alıyormuş gibi çalışması sağlanır. Bu durumda yazılan program Stage programının iki boyutlu grafik arayüzü sayesinde görsel bir hale getirilerek izlenmektedir. Player/Stage yazılımı gezgin robotlar ve donanımlarını kontrol ve test

etmek için, kullanıcıların geliştireceği programlar için gerekli araçların bütünüdür. Yazılan programla robot ve donanımı arasında bir arabirim olarak çalışan Player yazılımı, robot sunucusudur. Gerçek robot kullanılmayan çalışmalarda benzetimi yapılan robotun ve donanımının alt seviye programlanmasını gerçekleştirir ve yazılan programla robotun donanımına komutlar gönderilerek, duyurga bilgilerinin alınmasını TCP soketleri yardımıyla haberleşerek sağlar.

3.1 Player Yazılımı Arayüzleri

Bütün Player haberleşmesi kendine özgü mesaj yapısı ve işlevi olan arayüzler yardımıyla sağlanmaktadır. Bu kısım altında açıklanan arayüzler tez çalışması kapsamında kullanılan arayüzlerdir.

3.1.1 Player arayüzü

Player arayüzü player sunucusunun kendisini modellerken, sunucunun davranışlarını yapılandırırken kullanılır.

3.1.2 Position2d arayüzü

Düzlemsel gezgin robot için kullanılan arayüzdür. Position2d arayüzü gezgin robotun 2 boyutlu kontrolü için kullanılır.

3.1.3 Simulation arayüzü

Player aygıtları gerçek donanım olabildiği gibi Stage (2 boyutlu) ya da Gazebo (3 boyutlu) benzetim ortamları tarafından oluşturulan sanal aygıtlar olabilir. Simulation arayüzü benzetim ortamına doğrudan erişimi sağlayan bir arayüzdür.

3.1.4 Graphics2d arayüzü

Graphics2d arayüzü grafik aygıtlar için 2 boyutlu grafik arayüzüdür. Sürücüler, istemciler ve diğer sürücülerin grafiksel çıktı oluşturmalarını bu arayüz yardımıyla gerçekleştirir. Stage modelleri bu arayüz sayesinde istemcilerin Stage penceresine çizim yapmasını sağlarlar.

3.1.5 Opaque arayüzü

Kullanıcı tanımlı mesajlar için kullanılan genel bir arayüzdür. Bu arayüz sayesinde kullanıcı tanımlı mesajların ve sürücüler ile eklentilere özel komutların gönderimi mümkündür.

3.1.6 Sonar arayüzü

Sonar duyargaları gibi sabit bir grup uzaklık duyargalarına erişim için kullanılan arayüzdür.

3.1.7 Speech arayüzü

Speech arayüzü konuşma sentezleme sistemine erişimi sağlayarak bilginin benzetim ortamında görselleştirilmesini sağlayan arayüzdür.

3.2 Stage Yazılımı ve Bileşenleri

Stage yazılımı gezgin robotlar ve duyargalar için sanal bir dünyayı modelleyen 2 boyutlu bir benzetim ortamıdır. Bu benzetim ortamı pek çok duyarga ve eyleyici modeli sağlamaktadır. Stage yazılımı bu modeli kullanılarak benzetimi yapılacak duyarga ve duyarga modellerinden oluşan bir dünyayı world dosyasının içinde tanımlayarak benzetmektedir.

Model:

Stage benzetim ortamında kullanılacak olan nesnelerin temel özelliklerinin (konum bilgisi, büyüklük bilgisi, renk bilgisi vb.) ve sahip olduğu duyargalara göre özelleşmiş özelliklerinin (sonar duyargalar için sonar duyargasının açılma yerleşimi, büyüklüğü vb.) benzetilmesidir.

World Dosyası:

Stage yazılımında benzetimi yapılacak duyarga ve duyarga modellerinden oluşan ve çözünürlük, benzetim ortamı için kullanılan süre ve güncellenme bilgileri gibi bilgileri içermektedir.

4. LEAST RECENTLY VISITED YÖNTEMİ

LRV yöntemi (Batalin ve Sukhatme, 2007) duyarğa ağı kurma ve bakımı, alan tarama ve keşfi için geliştirilmiştir. Tek bir robotun çevrimiçi (Choset, 2001), duyarğa temelli (Choset, 2001) bir şekilde ortamı taraması esasına dayanmaktadır.

Bu yöntemde robot keşif ve tarama işlemini ortama bıraktığı duyarğalarla haberleşerek ve onlardan aldığı bilgiyi kendi donanımı yardımıyla edindiği bilgiyle sentezleyerek gerçekleştirir. Robot bir duyarğa ağı kuracak ve genişletecek şekilde ve her an en az bir duyarğayla haberleşerek ortamı bu duyarğaların önerdiği yön bilgilerini değerlendirerek tarar.

4.1 LRV Yöntemi

Batalin ve Sukhatme (2007)'de gerçekleştirilen ve Algoritma-1 ile Algoritma-2 de verilen Least Recently Visited (LRV) yöntemi robot tarafında ve düğüm tarafında olmak üzere iki parçadan oluşmaktadır.

4.1.1 Düğüm döngüsü

Algoritma-1: Least Recently Visited (LRV) – Duyarğa (düğüm i) Döngüsü

D(i)=>i düğümüne ait yönler;

W(d)=> ilgili yönden geçilme sayısı(d);

ANY_OF(G)=> En az ziyaret edilen yönün önerilmesi için kullanılan yön seçme fonksiyonu

Tekrarla

if robottan **d_{update}** yönüyle ilgili bir güncelleme mesajı geldiyse **then**

W(d_{update}) = W(d_{update}) + 1 // ilgili yönün geçildiğini gösteren bilgi güncellenir

Send(NODE_INFO, n, ANY_OF(arg min_d)) // en az ziyaret edilen yön bilgisi düğüm numarasıyla birlikte robota gönderilir

Düğüm tarafında gerçekleşen kısımda (Algoritma-1) robottan düğüme gönderilen düğüm numarası ve yön bilgisinden oluşan mesajın düğüm tarafından alınmasıyla mesajda belirtilen yönün ağırlık değerinin güncellenmesi gerçekleştirilir. Bu şekilde o yönden geçilme bilgisi güncellenmiş olup sonraki aşamalarda bu güncellenen yönden geçildiği için düğümün başka bir yön önermesi sağlanmaktadır.

Her düğüm için tanımlanan yön sayısı sınırlı ve uygulamaya bağlıdır. Bu yöntemde kullanılan yön sayısı 4'tür. Her i düğümü için tanımlanan $D(i)$ kümesi o düğümde gidilebilecek yönleri göstermektedir.

Düğümler ortama bırakıldığında her düğüm kendisiyle ilgili yerel bir bilgi olan en az ziyaret edilen yön bilgisini yayınlamaya başlar. En az ziyaret edilen yön bilgisi en küçük ağırlık değerine sahip olan yön bilgisidir ($W(i,d)$). Bu yön robota önerilerek robotun en az ziyaret edilen yöne yönlendirilmesiyle etkin bir şekilde ortamın taranması gerçekleştirilmektedir. Aynı ağırlığa sahip olan yön bilgileri için $ANY_OF(T)$ fonksiyonu belirli bir kurala göre ya da rasgele bir şekilde önerilecek yönü seçer. Bu kuralların detaylı açıklaması Bölüm 4.2' de verilmiştir.

Robot tarafında gerçekleştirilen kısım başlıca 3 temel bölümden oluşmaktadır (Algoritma-2).

- Robotun güncel düğümün (n) menzili dışında bulunması durumu
- Robotun güncel düğümün (n) menzilindeyken engelle karşılaşması durumu
- Robotun güncel düğümün (n) menzilindeyken gittiği yönde engel bulunmaması durumu

4.1.2 Robotun güncel düğümün menzili dışında bulunması durumu

Robot güncel düğümün (n) menzili dışında bulunuyorsa LRV (Batalin ve Sukhatme, 2007) yönteminde açıklanan iki farklı durum söz konusudur (Algoritma-2).

- Robotun haberleşme menziline güncel düğüm dışında düğüm bulunmaması.
- Robotun haberleşme menziline güncel düğüm dışında düğüm ya da düğümlerin bulunması.

Algoritma-2: Least Recently Visited (LRV)– Robot Döngüsü

$(n, d) \Rightarrow$ (güncel düğüm ve önerilen yön);

SHORT \Rightarrow yeni düğüm bırakırken kullanılacak haberleşme mesafesi;

$R \Rightarrow$ robotun çevresindeki düğümlerden gelen veri kümesi (düğüm numarası, sinyal gücü, önerilen yön);

Opposite (d) \Rightarrow verilen d yönünün tersini geri döndüren fonksiyon.

$R =$ çevredeki düğüm(ler)den **NODE_INFO** mesaj(lar)ını al.

if n 'nin **SHORT** ile belirlenen haberleşme menzili dışında ise **then**

$(n_{closest}, d_{closest}) = R$ kümesi içindeki en büyük sinyal gücüne sahip düğüm ve önerdiği yön

if $n \neq \text{NULL}$ **then**

Send (UPDATE_DIR, $n_{closest}, d_{closest}$)

Send(UPDATE_DIR, $n_{closest}, \text{Opposite}(d_{closest})$)

else

n ' düğüm numarasına ve d ' önerilen yönüne sahip bir düğüm bırak

$(n_{closest}, d_{closest}) = (n, d)$

if $d_{closest}$ yönünde engel yoksa

$(n, d) = (n_{closest}, d_{closest})$

else

Send (UPDATE_DIR, $n_{closest}, d_{closest}$)

Cevabı bekle, kontrolü tekrarla

if robot hareket ediyorsa ve engelle olan mesafe $\leq \text{OBSTACLE_AVOIDANCE_RANGE}$ **then**

if engel büyükse ve çevrede n 'nin dışında düğüm yoksa

n ' düğüm numarasına ve d ' önerilen yönüne sahip bir düğüm bırak

$(n, d) = (n, d)$

if d yönünde engel varsa **then**

Send(UPDATE_DIR, n, d)

Cevabı bekle, kontrolü tekrarla

else engelden sakın

if $d = \text{NULL}$ **then**

d yönünde ilerle

Robot güncel düğümün (\mathbf{n}) menzili dışında ise ve o anda robotun çevresinde güncel düğümünden başka düğüm(ler) yer alıyorsa bu düğüm(ler)den alınan bilgi(ler) değerlendirilerek sinyal gücü en fazla olan düğüm (robota en yakın olan) ¹ en yakın düğüm ($\mathbf{n}_{closest}$) olarak belirlenir. En yakın düğümün önerdiği yön ($\mathbf{d}_{closest}$) ve bu yönün tersi olan yön (Opposite($\mathbf{d}_{closest}$)) ile hazırlanan UPDATE_DIR mesajları en yakın düğüme ($\mathbf{n}_{closest}$) gönderilir. Güncellenen yön bilgilerinin ardından $\mathbf{d}_{closest}$ yönünde engel yoksa robota en yakın olarak bulunan bu düğüm ($\mathbf{n}_{closest}$) ve önerdiği yön bilgisi ($\mathbf{d}_{closest}$), güncel düğüm (\mathbf{n}) ve önerdiği yön bilgisi (\mathbf{d}) olarak atanır. Bundan sonra robotun güncel (\mathbf{n}) düğümün önerdiği yönde gitmesi (bu yönde engel yok ise) gerçekleşmektedir. Bu yönlerden geçildiği bilgisi düğüm döngüsü (Algoritma-1) sayesinde güncellenerek, robotun herhangi bir şekilde bu düğümün menziline tekrar girmesi durumunda düğümün robota ziyaret edilmemiş ya da en az ziyaret edilmiş olan yönleri önermesi sağlanmaktadır. Düğümün önerdiği yönde ($\mathbf{d}_{closest}$) engel var ise bu yönle ilgili güncelleme mesajı en yakın düğüme ($\mathbf{n}_{closest}$) gönderilerek engelsiz bir yön bulunması ve bu yönün robota önerilmesi sağlanmaktadır.

Robot güncel düğümün (\mathbf{n}) menzili dışında ise ve o anda robotun çevresinde başka düğüm(ler) bulunmuyorsa yeni bir tekil düğüm tanımlayıcısı (\mathbf{n}') oluşturularak yeni bir düğüm bırakılır. Bu düğüm ve önerdiği yön en yakın düğüm ve onun önerdiği yön olarak atanır ($\mathbf{n}_{closest}$, $\mathbf{d}_{closest}$). Düğümün önerdiği yönde ($\mathbf{d}_{closest}$) engel var ise bu yönle ilgili güncelleme mesajı en yakın düğüme ($\mathbf{n}_{closest}$) gönderilerek engelsiz bir yön bulunması ve bu yönün robota önerilmesi sağlanmaktadır.

4.1.3 Robotun güncel düğümün menzilindeyken engelle karşılaşması durumu

Robotun o anda ilerlediği yönde herhangi bir engelle karşılaşması o yönde ilerlemesini engelleyecek herhangi bir nesnenin bulunması şeklinde tanımlanır. Burada dikkat edilecek husus robotun büyüklüğü ve sonar duyarga verisi değerlerine göre robotun hareketi yönünde ilerlemesini engelleyecek durumların ne şekilde değerlendirileceğidir.

¹ Bu tez çalışması kapsamında LRV ve E-LRV yöntemleri gerçekleştirirken sinyal gücü bilgisi, Öklit Mesafesine göre robot ve düğüm arası mesafe bilgisi hesaplanarak kullanılmıştır

Robot herhangi bir yönde ilerlerken engelle karşılaşsa ve o andaki sonar duyurga verileri OBSTACLE_AVOIDANCE_RANGE değerinden küçükse engelle ilgili izlenecek yol belirlenir. İzlenecek yol engelin büyüklüğü ve o anda robotun çevresinde güncel düğümden (n) başka bir düğümün bulunup bulunmamasına bağlıdır.

Algoritma-2'ye göre karşılaşılan engel büyükse ve o anda robot güncel düğümün (n) dışındaki herhangi bir düğümün haberleşme menzilinde değilse yeni bir tekil düğüm tanımlayıcısı (n') oluşturularak yeni bir düğüm bırakılır. Bu düğüm ve önerdiği yön güncel düğüm ve onun önerdiği yön olarak atanır (n, d). Düğümün önerdiği yönde (d) engel var ise bu yönle ilgili güncelleme mesajı güncel düğüme (n) gönderilerek engelsiz bir yön bulunması ve bu yönün robota önerilmesi sağlanmaktadır.

Diğer durumlarda (engelin küçük olduğu ya da ortamda güncel düğüm dışına en az bir düğümün bulunduğu) robotun o andaki sonar duyurga verileri değerlendirilerek robotun engelden sakınması sağlanmaktadır.

Robotun herhangi bir şekilde engelle karşılaştığında nasıl bir strateji izleyerek hangi verileri ne şekilde değerlendireceği detaylı bir şekilde Bölüm 5.3.1'de anlatılmıştır.

4.1.4 Robotun güncel düğümün menzilindeyken gittiği yönde engel bulunmaması durumu

Robot o anda ilerlemekte olduğu yönde herhangi bir engelle karşılaşmamışsa o yönde, sahip olduğu hızla ilerler (Algoritma-2).

Robotun o anda ilerlediği yönde bir engelle karşılaşması o yönde ilerlemesini engelleyecek bir nesnenin olması şeklinde tanımlanır. Burada dikkat edilecek husus robotun büyüklüğü ve sonar duyurga verisi değerlerine göre robotun ilerlemesini engelleyecek durumların belirlenebilmesidir. Bu kısmın gerçekleştirme detayları Bölüm 5.3.1'de anlatılmıştır.

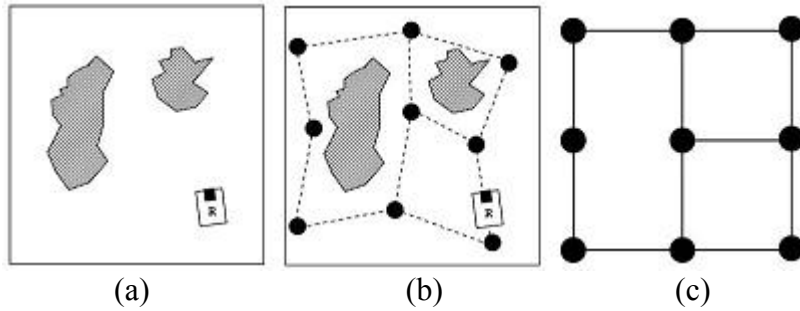
4.2 LRV Yöntemi Detayları

LRV yöntemi robot tarafında ve her duyurga düğümünde olmak üzere iki temel kısımdan oluşmaktadır (Algoritma-1 ve Algoritma-2). Robot herhangi bir ortama

birakıldığında Algoritma-2’de belirtilen robot döngüsüne göre haberleşeceği herhangi bir duyarga düğüm olmadığı için ortama bir duyarga düğüm bırakmaktadır. Bundan sonraki aşamalarda robot her adımda duyarga düğüm ağı oluşturacak ve bu ağı genişletecek şekilde çevresindeki en az bir duyarga düğümüyle haberleşebileceği bir düğüm bırakır.

LRV yönteminde düğüm tarafında yönlerin ağırlıklarının güncellenmesi robotun bir yönden geçmesinden önce ve o yönden geçmesinden sonra olmak üzere düğüm tarafında 2 kere yapılarak ilgili yönün geçilme sayısı bilgisi artırılmaktadır (Algoritma-1).

LRV yöntemine göre ortama bırakılan düğümlerin konumlanması sonlu bir çizge oluşturacak şekilde yapılmaktadır. $G(V,E)$ sonlu çizgesinde V çizge düğümlerini (LRV yöntemindeki düğümleri), E çizge düğümlerini birleştiren ayrıtları temsil etmektedir.

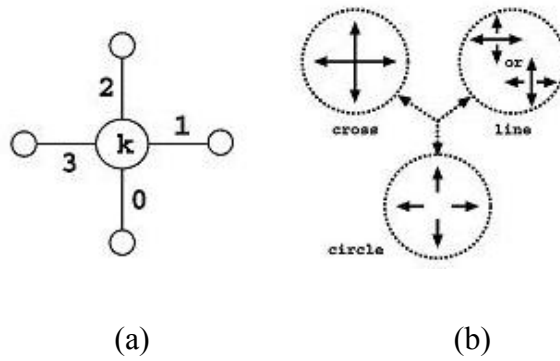


Şekil 4.1 : LRV yöntemi adımları.

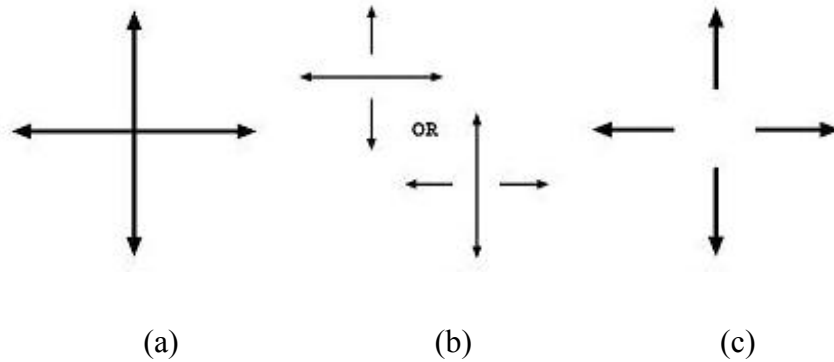
Çizge düğümlerini birleştiren ayrıtlar her i, j düğümü için bu iki düğümün birbirlerinin haberleşme menziline bulunması ve bu iki düğüm arasında fiziksel bir yol bulunması şeklinde tanımlanmaktadır. Şekil 4.1 (a)’da robotun ortama bırakılmasından önceki durum, Şekil 4.1 (b)’de düğümlerin ortama bırakılmasıyla oluşan düğüm ağı ve Şekil 4.1 (c)’de ise düğümlerin oluşturduğu bağlı çizge yer almaktadır.

LRV yönteminde aynı sayıda geçilmiş olan yönlerden hangisine gidilmesi gerektiğini belirlemek için belirli bir kurala göre ya da rasgele bir şekilde uygun yönün seçilmesini sağlayan ANY_OF(T) fonksiyonu düğüm döngüsünde kullanılmıştır (Algoritma-1). Yön seçme kuralı gidilecek olan yönlerin sıralaması şeklinde

tanımlanmaktadır. Şekil 4.2 (a)'da LRV yönteminde kullanılan temel yön bilgileri yer almaktadır. Şekil 4.2 (b)'de LRV'de tanımlı ve kullanılan çapraz (cross), çizgi (line) ve daire (circle) yön seçme kuralları genel gösterimi verilmiştir. Örneğin Yön seçme kuralı {0213} şeklinde tanımlanmışsa ve 2 ile 1 yönünden geçme bilgisi eşit ağırlıklı ise bu kurala göre öncelikle 2 yönü önerilecektir.



Şekil 4.2 : LRV yönteminde kullanılan temel yön bilgileri ve yön seçme kuralları.

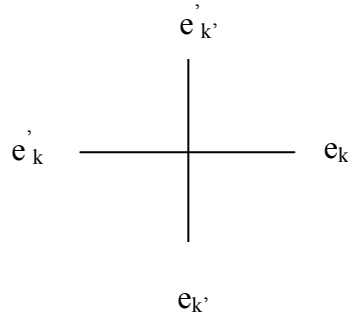


Şekil 4.3 : LRV yönteminde kullanılan çapraz, çizgi ve daire yön seçme kuralları.

LRV Yönteminde kullanılan Çapraz, Çizgi ve Daire Yön Seçme Kuralları Şekil 4.3 (a), (b) ve (c)'de verilmiş olup bu kuralların detaylı açıklamaları bundan sonraki kısımda anlatılmaktadır.

Çapraz Kural:

Şekil 4.4'te belirtilen isimlendirmeye göre $\{e_k - e'_k - e_{k'} - e'_{k'}\}$ sıralamasına sahip yön seçme kuralıdır. Şekil 4.3 (a)'da gösterilen Çapraz Yön Seçme Kuralı sınıfının tanımlı bütün öğeleri LRV'de kullanılan ve Şekil 4.2 (a)'da gösterilen temel yön bilgilerine uygulanarak Çizelge 4.1'de verilmiştir.



Şekil 4.4 : Yön seçme kuralı için kullanılacak isimlendirme.

Çizelge 4.1 : Çapraz yön seçme kuralı.

ÇAPRAZ KURAL SINIFI	
Öge Numarası	Çapraz Kural Sınıfı Öğeleri
1	{ 0, 2, 1, 3 }
2	{ 0, 2, 3, 1 }
3	{ 2, 0, 1, 3 }
4	{ 2, 0, 3, 1 }
5	{ 1, 3, 0, 2 }
6	{ 1, 3, 2, 0 }
7	{ 3, 1, 0, 2 }
8	{ 3, 1, 2, 0 }

Bu kurala göre {3,1,2,0} sıralamasına sahip olan bir öge, yön seçme kuralı olarak kullanılacaksa aynı ağırlığa sahip olan bu yönlerden öncelikle {3} yönü robota önerilecektir. Sonra sırasıyla {1},{2},{0} yönleri düğüm tarafından robota önerilecek olan yön olarak belirlenecektir. Farklı ağırlığa sahip olan yönler için en küçük olan ve bu ögeye göre belirlenen yön robota önerilecektir ({0,1,2,3} yönleri uygulamaya bağlı olarak belirlenen yönlerdir).

Çizgi Kuralı:

Şekil 4.4'te belirtilen isimlendirmeye göre { $e_k - e_{k'} - e_{k'} - e_k$ } sıralamasına sahip yön seçme kuralıdır. Şekil 4.3 (b)'de gösterilen Çizgi Yön Seçme Kuralı sınıfının tanımlı bütün öğeleri LRV'de kullanılan ve Şekil 4.2 (a)'da gösterilen temel yön bilgilerine uygulanarak Çizelge 4.2'de verilmiştir.

Çizelge 4.2 : Çizgi yön seçme kuralı.

ÇİZGİ KURAL SINIFI	
Öge Numarası	Çizgi Kural Sınıfı Öğeleri
1	{ 0, 1, 3, 2 }
2	{ 0, 3, 1, 2 }
3	{ 2, 1, 3, 0 }
4	{ 2, 3, 1, 0 }
5	{ 1, 0, 2, 3 }
6	{ 1, 2, 0, 3 }
7	{ 3, 0, 2, 1 }
8	{ 3, 2, 0, 1 }

Daire Kuralı:

Şekil 4.4'te belirtilen isimlendirmeye göre $\{ e_k - e_{k'} - e'_k - e'_{k'} \}$ sıralamasına sahip yön seçme kuralıdır. Şekil 4.3 (c)' de gösterilen Daire Yön Seçme Kuralı sınıfının tanımlı bütün öğeleri LRV'de kullanılan ve Şekil 4.3 (a)'da gösterilen temel yön bilgilerine uygulanarak Çizelge 4.3'te verilmiştir.

Çizelge 4.3 : Daire yön seçme kuralı.

DAİRE KURAL SINIFI	
Öge Numarası	Çizgi Kural Sınıfı Öğeleri
1	{ 0, 1, 2, 3 }
2	{ 1, 2, 3, 0 }
3	{ 2, 3, 0, 1 }
4	{ 3, 0, 1, 2 }
5	{ 3, 2, 1, 0 }
6	{ 2, 1, 0, 3 }
7	{ 1, 0, 3, 2 }
8	{ 0, 3, 2, 1 }

LRV yönteminde kullanılan yön seçme kurallarına göre en iyi performans çapraz kuralla, en kötü performans ise daire kuralı ile elde edilmiştir (Batalin ve Sukhatme, 2007).

4.3 LRV Yöntemi Gerçekleme Detayları

LRV yöntemi (Batalin ve Sukhatme, 2007) bu tez kapsamında KUBUNTU 8.04 işletim sisteminde, Player/Stage (Player 2.0.3 ve Stage 2.0.4) benzetim ortamı kullanılarak, Eclipse geliştirme ortamında C/C++ dilleri ile gerçekleştirilmiştir.

LRV yönteminin (Batalin ve Sukhatme, 2007) gerekleme detayları bu tez kapsamında LRV yönteminin geliştirilmesi alışmasının yer aldığı 5. Bölümde, geliştirilen bu yöntemle karşılaştırmalı olarak verilmiştir.

5. LEAST RECENTLY VISITED YÖNTEMİNİN GELİŞTİRİLMESİ

Bu tezde LRV yönteminin (Batalin ve Sukhatme, 2007) eksiklikleri üzerine gerekli görülen çalışmalar yapılmış ve geliştirilen bu yöntem ENHANCED LEAST RECENTLY VISITED (E-LRV) yöntemi olarak adlandırılmıştır.

5.1 E-LRV Yöntemi

E-LRV yöntemi, LRV yöntemi (Batalin ve Sukhatme, 2007) gibi duyurga ağı kurma ve bakımı, alan tarama ve keşfi için geliştirilmiştir. Bu yöntemde de robot, tarama ve keşif işlemini ortama bıraktığı duyurgalarla haberleşerek ve onlardan aldığı bilgiyi kendi donanımı yardımıyla edindiği bilgiyle sentezleyerek gerçekleştirir. Robot bir duyurga ağı kuracak ve genişletecek şekilde ve her an en az bir duyurgayla haberleşerek ortamı bu duyurgaların önerdiği yön bilgilerini değerlendirerek tarar.

LRV yönteminde robot ortama düğüm bırakırken bu sırada ortamın engelli olup olmaması bilgisini kullanmamaktadır. Ayrıca robot bir düğümü bırakırken yine geçtiği yönlerle ilgili yön güncelleme işlemini yapmamaktadır. Bu durumda düğüm bırakılırken ortamdaki elde edilen bilgi tam olarak kullanılmamaktadır.

E-LRV yönteminde, LRV yönteminde yer almayan düğüm bırakılması sırasında robotun geçtiği yönlerin güncellenmesi ve düğüm bırakılırken ortamdaki engel durumuna göre ilgili yönlerin güncellenmesi işlemleri yapılarak, tüm ortamın taranması işleminde yön güncellemeden kaynaklanan mesaj trafiğinin azaltılması ve ortam verilerinin daha etkin bir şekilde kullanılmasıyla daha kısa sürede ortamın taranması sağlanmaktadır.

Bütün akışı Algoritma-3-a, Algoritma-3-b, Algoritma-3-c, Algoritma-3-d şeklinde parçalara ayrılarak verilen E-LRV yönteminde, LRV yöntemini (Batalin ve Sukhatme, 2007) oluşturan başlıca üç temel kısım aynı şekilde bulunmaktadır.

- Robotun güncel düğümün (**n**) menzili dışında bulunması durumu (Algoritma-3-b).
- Robotun güncel düğümün (**n**) menzilindeyken engelle karşılaşması durumu (Algoritma-3-c).
- Robotun güncel düğümün (**n**) menzilindeyken gittiği yönde engel bulunmaması durumu (Algoritma-3-d).

Algoritma-3-a 'da E-LRV yönteminde kullanılan genel bilgiler ve her döngüde çevrede bulunan düğümlerden bilgi alınması ile ilgili akış yer almaktadır.

<u>Algoritma-3-a: Enhanced Least Recently Visited(E-LRV)- Robot Döngüsü</u>	
<u>Genel Bilgiler ve Genel Cevrim</u>	
Y_{robotYön}	=> robotun o anda gitmekte olduğu yön;
(n, d)	=>(güncelDüğüm, önerilenYön);
R	=>robotun etkileşim alanı içinde bulunan düğümlerden alınan verilerin saklandığı veri kümesi (düğümün tekil tanımlayıcısı, düğümün önerdiği yön);
SHORT	=>yeni düğümlerin bırakılacağı durumu belirleyen haberleşme menzili değeri
Y_{robotÖncekiYön}	=>robotun o anda gittiği yönden önce gittiği yön;
Y_{engelliYön}	=>düğüm bırakılırken ortamda engel varsa, engelin yönü;
Y_{engelsizYön}	=>düğüm bırakılırken ortamda bulunan engelsiz yön;
R	=çevrede bulunan düğümlerden Periyodik-Bilgi-Mesajı-Cevabı alınır.

5.1.1 Robotun güncel düğümün menzili dışında bulunması durumu

Robot güncel düğümün (**n**) menzili dışında bulunuyorsa E-LRV yönteminde LRV (Batalin ve Sukhatme, 2007) yöntemindeki ile aynı şekilde gerçekleştirilen iki farklı durum söz konusudur (Algoritma-3-b).

- Robotun haberleşme menziline güncel düğüm dışında düğüm bulunmaması durumu.
- Robotun haberleşme menziline güncel düğüm dışında düğüm ya da düğümlerin bulunması durumu.

Algoritma-3-b: Enhanced Least Recently Visited(E-LRV)- Robot Döngüsü

Robotun Güncel Düğümün Menzili Dışında Bulunması Durumu

if n 'nin haberleşme menzili dışında ise then

$(n_{enyakın}, d_{enyakın}) = R$ içinde yer alan en büyük sinyal gücüne sahip düğüm ve yön bilgisi

if $n \neq NULL$ then

Gönder(Yön-Güncelleme-Mesajı, $n_{enyakın}$, $d_{enyakın}$)

Gönder(Yön-Güncelleme-Mesajı, $n_{enyakın}$, $Y_{robotÖncekiYön}$)

if $d_{enyakın}$ yönünde engel yoksa

$(n, d) = (n_{enyakın}, d_{enyakın})$

else

Gönder(Yön-Güncelleme-Mesajı, $n_{enyakın}$, $d_{enyakın}$)

Cevabı bekle ve kontrolü tekrarla

$Y_{robotYön} = d_{enyakın}$;

else

Engel bulunmayan bir yön bul($Y_{engelsizYön}$)

$Y_{robotYön} = Y_{engelsizYön}$;

$Y_{robotYön}$ yön bilgisine sahip yeni bir düğüm bırak (n')

$(n, d) = (n', Y_{robotYön})$

Gönder(Yön-Güncelleme-Mesajı, n , $Y_{robotYön}$)

Gönder (Yön-Güncelleme-Mesajı, n , $Y_{robotÖncekiYön}$)

Engelli yönü belirle($Y_{engelliYön}$)

Gönder(Engel-Güncelleme-Mesajı, n , $Y_{engelliYön}$)

Bütün engelli yön(ler) için tekrarla

Robot güncel düğümün (n) menzili dışında ise ve o anda robotun çevresinde güncel düğümünden başka düğüm(ler) yer alıyorsa bu düğüm(ler)den alınan bilgi(ler) değerlendirilerek sinyal gücü en fazla olan düğüm (robota en yakın olan)² en yakın düğüm ($n_{enyakın}$) olarak belirlenir. En yakın düğümün önerdiği yön ($d_{enyakın}$) ve

² Bu tez çalışması kapsamında LRV ve E-LRV yöntemleri gerçekleştirirken sinyal gücü bilgisi Öklit Mesafesine göre robot ve düğüm arası mesafe bilgisine hesaplanarak kullanılmıştır.

robotun önceki yönü ($Y_{robotÖncekiYön}$) bilgilerini içeren **Yön-Güncelleme-Mesajı** hazırlanarak ilgili düğüme ($n_{enyakın}$) gönderilir. Güncellenen yön bilgilerinin ardından alınan **Yön-Güncelleme-Mesajı-Cevabı** mesajının sonucunda $d_{enyakın}$ yönünde engel yoksa robota en yakın olarak bulunan bu düğüm ($n_{enyakın}$) ve önerdiği yön bilgisi ($d_{enyakın}$), güncel düğüm (n) ve önerdiği yön bilgisi (d) olarak atanır. Bundan sonra robotun güncel (n) düğümün önerdiği yönde gitmesi (bu yönde engel yok ise) gerçekleşmektedir. Bu yönlerden geçildiği bilgisi düğüm döngüsü sayesinde güncellenerek robotun herhangi bir şekilde bu düğümün menziline tekrar girmesi durumunda düğümün robota ziyaret edilmemiş ya da en az ziyaret edilmiş olan yönleri önermesi sağlanmaktadır. Düğümün önerdiği yönde ($d_{enyakın}$) engel var ise bu yönle ilgili güncelleme mesajı en yakın düğüme ($n_{enyakın}$) gönderilerek engelsiz bir yön bulunması ve bu yönün robota önerilmesi sağlanmaktadır.

Robot güncel düğümün (n) menzili dışında ise ve o anda robotun çevresinde başka düğümler bulunmuyorsa yeni bir tekil düğüm tanımlayıcısı (n') oluşturularak LRV yönteminden farklı olarak engelsiz olarak bulunan bir yönde yeni bir düğüm bırakılır. Güncel düğüm (n) ve önerdiği yön (d) bilgileri yeni düğüm tekil tanımlayıcısı (n') ve robotun o anda gitmekte olduğu yön ($Y_{robotYön}$) olarak güncellenir. Düğüm tarafına $Y_{robotYön}$ ve $Y_{robotÖncekiYön}$ ile bırakılan düğümün tekil tanımlayıcılarından oluşan **Yön-Güncelleme-Mesajı** gönderilir. Bu şekilde düğümün sahip olduğu yönlerden geçilme sayısı bilgileri robotun geçtiği yön bilgilerine göre güncellenmiş olur. Bu sayede ortam hakkında edinilen bilgi mümkün olan en kısa zamanda değerlendirilerek, daha doğru sonuçların daha kısa zamanda elde edilmesi sağlanmaktadır.

Ayrıca LRV yönteminden farklı olarak robotun düğümü bırakırken bulunduğu yön ve daha önce bulunduğu yön bilgisinin dışında ortamda başka engel(ler) mevcut ise bu engelli yönler için de gerekli **Engel-Güncelleme-Mesajlarını** ilgili düğüme gönderilir. Bu mesajı alan düğüm mesajda belirtilen yönün geçilme sayısını oldukça büyük bir değerle güncelleyerek sonraki adımlarda bu yönün önerilmemesini sağlar.

5.1.2 Robotun güncel düğümün menzilindeyken engelle karşılaşması durumu

Algoritma-3-c: Enhanced Least Recently Visited(E-LRV)- Robot Döngüsü

Robotun Engelle Karşılaşması Durumu

```
if Robot hareket halindeyse ve engel sakınımı mesafesi <= ENGEL_SAKINIM_DEĞERİ
then
    if Engel büyükse ve çevrede herhangi bir düğüm yoksa
        engel bulunmayan bir yön bul(  $Y_{engelsizYön}$  )
         $Y_{robotYön} = Y_{engelsizYön}$ ;
         $Y_{robotYön}$  yön bilgisine sahip yeni bir düğüm bırak ( $n'$ )
         $(n, d) = (n', Y_{robotYön})$ 
        Gönder( $Yön-Güncelleme-Mesajı, n, Y_{robotYön}$ )
        Gönder ( $Yön-Güncelleme-Mesajı, n, Y_{robotÖncekiYön}$ )
        Engelli yönü belirle( $Y_{engelliYön}$ )
        Gönder( $Engel-Güncelleme-Mesajı, n, Y_{engelliYön}$ )
        Bütün engelli yön(ler) için tekrarla
    else
        Engelden sakın
```

Robotun o anda ilerlediği yönde ($Y_{robotYön}$) herhangi bir engelle karşılaşması o yönde ilerlemesini engelleyecek herhangi bir nesnenin bulunması şeklinde tanımlanır. Burada dikkat edilecek husus robotun büyüklüğü ve sonar duyuraya verisi değerlerine göre robotun ($Y_{robotYön}$) yönünde ilerlemesini engelleyecek durumların belirlenmesidir.

Robot herhangi bir yönde ilerlerken engelle karşılaşır ve o andaki sonar duyuraya verileri ENGEL_SAKINIM_DEĞERİ değerinden küçükse engelle ilgili izlenecek yol belirlenir. İzlenecek yol engelin büyüklüğü ve o anda robotun çevresinde güncel düğümün (n) başka bir düğümün bulunup bulunmamasına bağlıdır.

Karşılaşılan engel büyükse ve o anda robot güncel düğümün (n) dışındaki herhangi bir düğümün haberleşme menzilinde değilse yeni bir tekil düğüm tanımlayıcısı (n') oluşturularak LRV yönteminden farklı olarak engelsiz olarak bulunan bir yönde yeni bir düğüm bırakılır (Algoritma-3-c). Güncel düğüm (n) ve önerdiği yön (d) bilgileri

yeni düğüm tekil tanımlayıcısı (n') ve robotun o anda gitmekte olduğu yön ($Y_{robotYön}$) olarak güncellenir. Düğüm tarafına $Y_{robotYön}$ ve $Y_{robotÖncekiYön}$ ile bırakılan düğümün tekil tanımlayıcılarından oluşan **Yön-Güncelleme-Mesajı** gönderilir. Bu şekilde düğümün sahip olduğu yönlerden geçilme sayısı bilgileri robotun geçtiği yön bilgilerine göre güncellenmiş olur. Bu sayede ortam hakkında edinilen bilgi mümkün olan en kısa zamanda değerlendirilerek, daha doğru sonuçların daha kısa zamanda elde edilmesi sağlanmaktadır.

Diğer durumlarda (engelin küçük olduğu ya da ortamda güncel düğüm dışında bir düğüm bulunuyorsa) robotun o andaki sonar duyarga verileri değerlendirilerek robotun engelden sakınması sağlanmaktadır.

Robotun herhangi bir şekilde engelle karşılaştığında nasıl bir strateji izleyerek hangi verileri ne şekilde değerlendireceği detaylı bir şekilde Bölüm 5.3.1'de anlatılmıştır.

5.1.3 Robotun güncel düğümün menzilindeyken gittiği yönde engel bulunmaması durumu

Robot o anda ilerlemekte olduğu yönde herhangi bir engelle karşılaşmamışsa o yönde ($Y_{robotYön}$), belirtilen hızla ilerler (Algoritma-3-d).

Algoritma-3-d: Enhanced Least Recently Visited(E-LRV)- Robot Döngüsü

Robotun Gittiği Yönde Engel Bulunmaması Durumu

if $Y_{robotYön} \neq \text{NULL}$ then

$Y_{robotYön}$ yönünde ilerle

Robotun o anda ilerlediği yönde ($Y_{robotYön}$) herhangi bir engelle karşılaşması o yönde ilerlemesini engelleyecek bir koşulun oluşması şeklinde tanımlanır. Burada dikkat edilecek husus robotun büyüklüğü ve sonar duyarga değerlerine göre robotun ilerlemesini engelleyecek durumların belirlenebilmesidir. Bu kısmın gerçekleştirme detayları Bölüm 5.3.1'de anlatılmıştır.

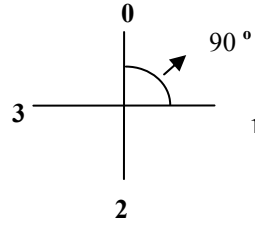
E-LRV yönteminde düğüm döngüsü, LRV yönteminde verilen düğüm döngüsüyle aynı şekildedir. Düğüm tarafında alınan yön bilgisinin geçilme sayısı güncellenerek robota en az ziyaret edilen yönün önerilmesi için gerekli mesajın gönderilmesi esasına dayanır. E-LRV yönteminde LRV yönteminden farklı olarak, engelli yönler

için geçilme bilgisinin güncellenmesi işlemi büyük bir değerle yapılmaktadır. Robottan düğümlere ve düğümlerden robota gönderilen mesajlar detaylı bir şekilde Bölüm 5.3.3’te verilmiştir.

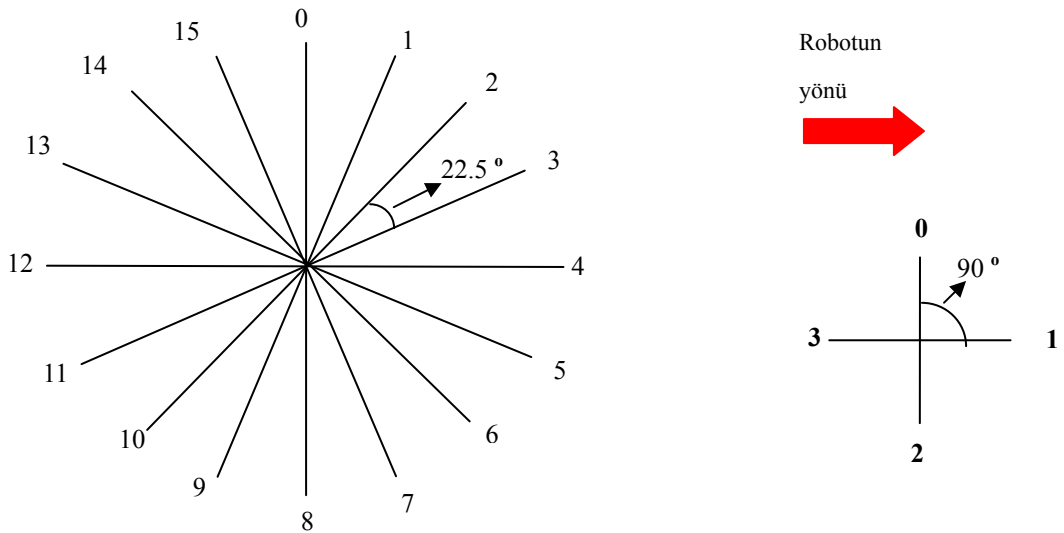
5.2 E-LRV Yöntemi Gerçekleme Detayları

E-LRV, LRV yönteminin (Batalin ve Sukhatme, 2007) eksik görülen kısımlarının geliştirilerek Player/Stage benzetim ortamının (Gerkey ve diğ., 2001), (Gerkey ve diğ., 2003), (Url-1) kısıtlarına uygun olarak gerçekleştirilmiş halidir.

5.2.1 Genel kavramlar



Şekil 5.1 : E-LRV yöntemi yön bilgisi.



Şekil 5.2 : Sonar duyargalarıyla donatılmış robotun 1 yönünde hareketi.

E-LRV yönteminde kullanılan 4 temel yön ve bu yönlerin isimlendirilmesi Şekil 5.1'de yer almaktadır. Şekil 5.2'de 1 yönünde hareket eden robotun sonar duyargalarının açısal yerleşimi gösterilmektedir.

5.2.2 Player/Stage benzetim ortamında kullanılan arayüzler

5.2.2.1 Player arayüzü

E-LRV yönteminin gerçekleşmesinde Player arayüzü robotun player sunucusu üzerinden düğümlerle haberleşmesini sağlayarak robotun davranışlarını modellemede esas alınan arayüzdür.

5.2.2.2 Position2d arayüzü

E-LRV yönteminin gerçekleşmesinde Position2d arayüzü robotun 2 boyutlu kontrolünü içeren konum bilgilerinin alınması, hız bilgilerinin değiştirilmesi gibi işlemleri gerçekleştirmek için kullanılmaktadır.

5.2.2.3 Simulation arayüzü

E-LRV yönteminin gerçekleşmesinde Simulation arayüzü bu yöntemde açıklanan robotun düğüm bırakması durumunda, düğümün Player/Stage benzetim ortamı üzerinde robot tarafından bırakıldığı yere yerleştirilmesi için kullanılır (Şekil 5.3).

5.2.2.4 Graphics2d arayüzü

E-LRV yönteminin gerçekleşmesinde Graphics2d arayüzü düğümlerin haberleşme menziline göstermek için kullanılır (Şekil 5.3).

5.2.2.5 Opaque arayüzü

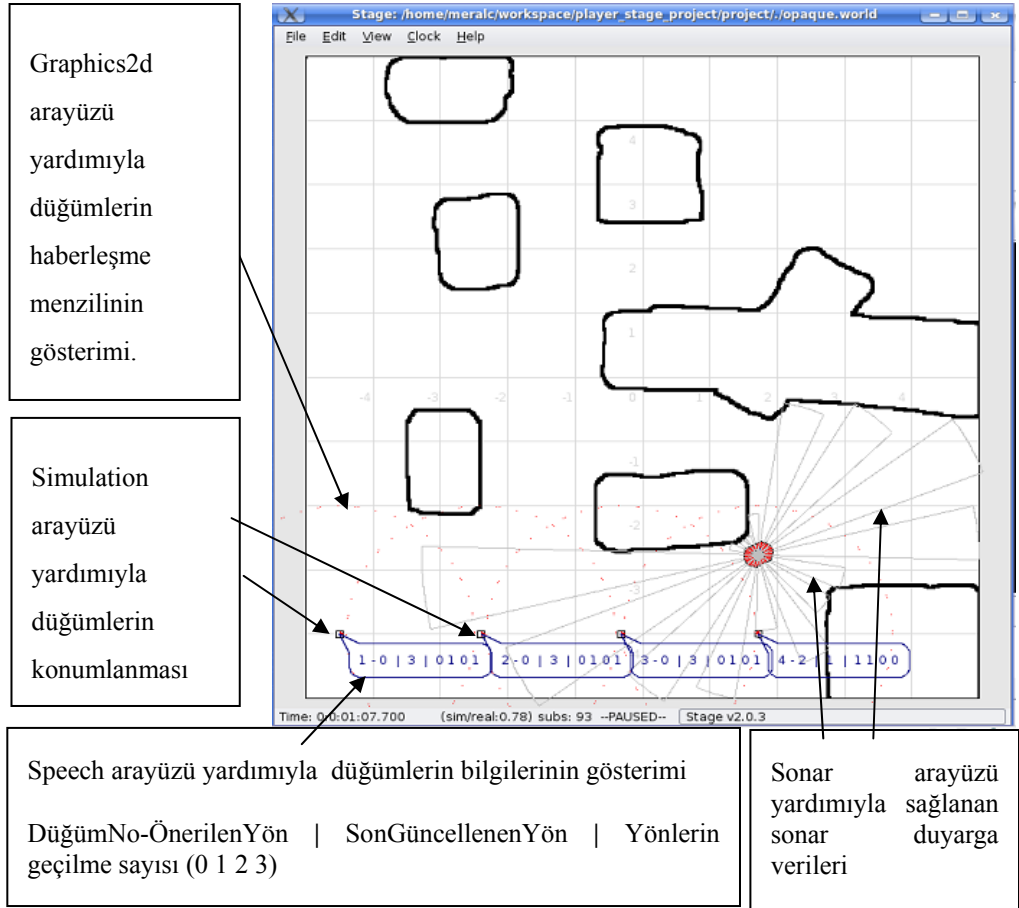
E-LRV yönteminin gerçekleşmesinde Opaque arayüzü düğümler ve robotun yine bu yöntem kapsamında tanımlanan mesajları (**Konum-Mesajı, Konum-Mesajı-Cevabı, Yön-Güncelleme-Mesajı, Yön-Güncelleme-Mesajı-Cevabı, Engel-Güncelleme-Mesajı, Engel-Güncelleme-Mesajı-Cevabı, Periyodik-Bilgi-Mesajı, Periyodik-Bilgi-Mesajı-Cevabı**) göndermesi için kullanılır.

5.2.2.6 Sonar arayüzü

E-LRV yönteminin gerçekleşmesinde 16 tane sonar duyargayla donatılmış gezgin robota Sonar arayüzü yardımıyla erişilerek sonar değerlerinin okunması ve değerlendirilmesi gerçekleştirilir (Şekil 5.3).

5.2.2.7 Speech arayüzü

E-LRV yönteminin gerçekleşmesinde Speech arayüzü düğümler için önerdikleri yön ve yerel olarak kendi çevrelerindeki yönlerden geçilme sayısının gösterimi için, robot içinse o anda gittiği yön ve haberleşme menziline bulunduğu düğüm bilgisinin gösterimi için kullanılmaktadır (Şekil 5.3).



Şekil 5.3 : Player arayüzlerinin kullanımının gösterimi.

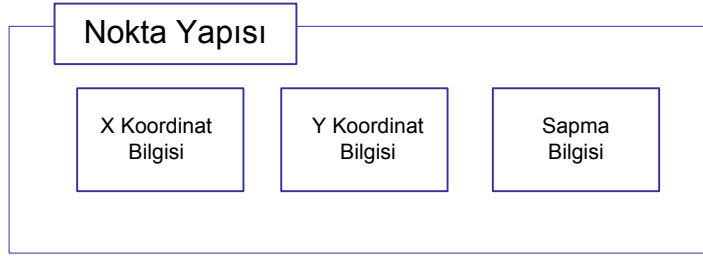
5.2.3 Kullanılan veri yapıları

5.2.3.1 Robot ve düğümde ortak olarak kullanılan veri yapıları

Nokta Yapısı:

Şekil 5.4'te gösterilen Nokta Yapısı, robot tarafında, robotun ve robotla haberleşen düğümlerin X ve Y koordinatı bilgileri ile x-y koordinat düzlemine göre sapma bilgisinin tutulduğu yapıdır.

Düğüm tarafında, sadece o düğümün X ve Y koordinatı bilgileri ile sapma bilgisinin tutulduğu yapıdır.



Şekil 5.4 : Robot ve düğüm tarafında ortak olarak kullanılan Nokta Yapısı.

Temel Düğüm Yapısı:

Şekil 5.5'te gösterilen Temel Düğüm Yapısı, robot tarafında robotla haberleşen düğümlerin tekil tanımlayıcı bilgisi ve önerdiği yön bilgisini içeren yapıdır.

Düğüm tarafında sadece o düğümün tekil tanımlayıcı bilgisi ve önerdiği yön bilgisini içeren yapıdır.



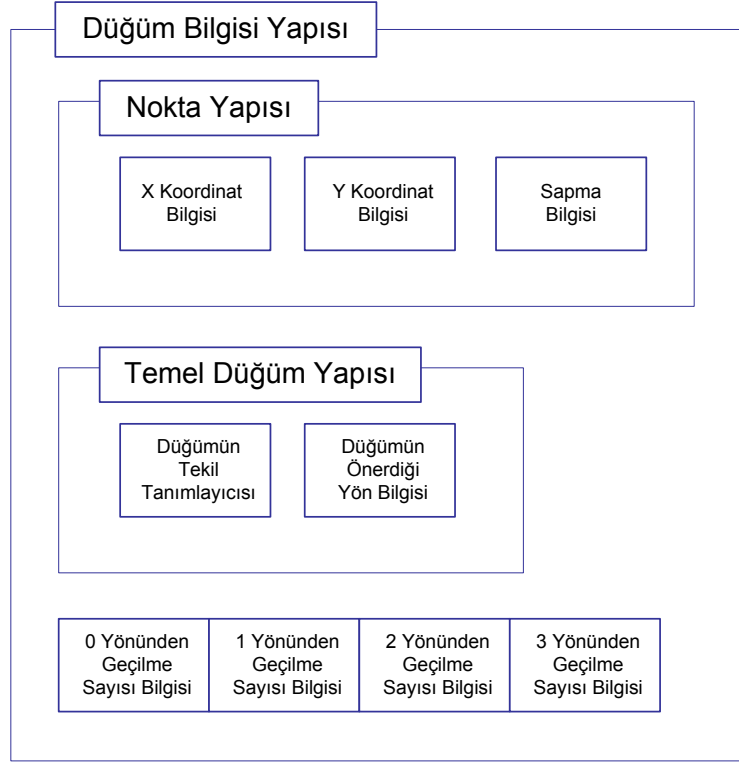
Şekil 5.5 : Robot ve düğüm tarafında ortak olarak kullanılan Temel Düğüm Yapısı.

5.2.3.2 Düğüm tarafında kullanılan veri yapıları

Nokta Yapısı ve Temel Düğüm Yapısı robot ve düğümler tarafında ortak olarak kullanılan veri yapılarıdır. Bu yapıların haricinde yine bu yapılar kullanılarak oluşturulan Düğüm Bilgisi Yapısı ve Düğüm Yapısı veri yapıları aşağıda verilmiştir.

Düğüm Bilgisi Yapısı:

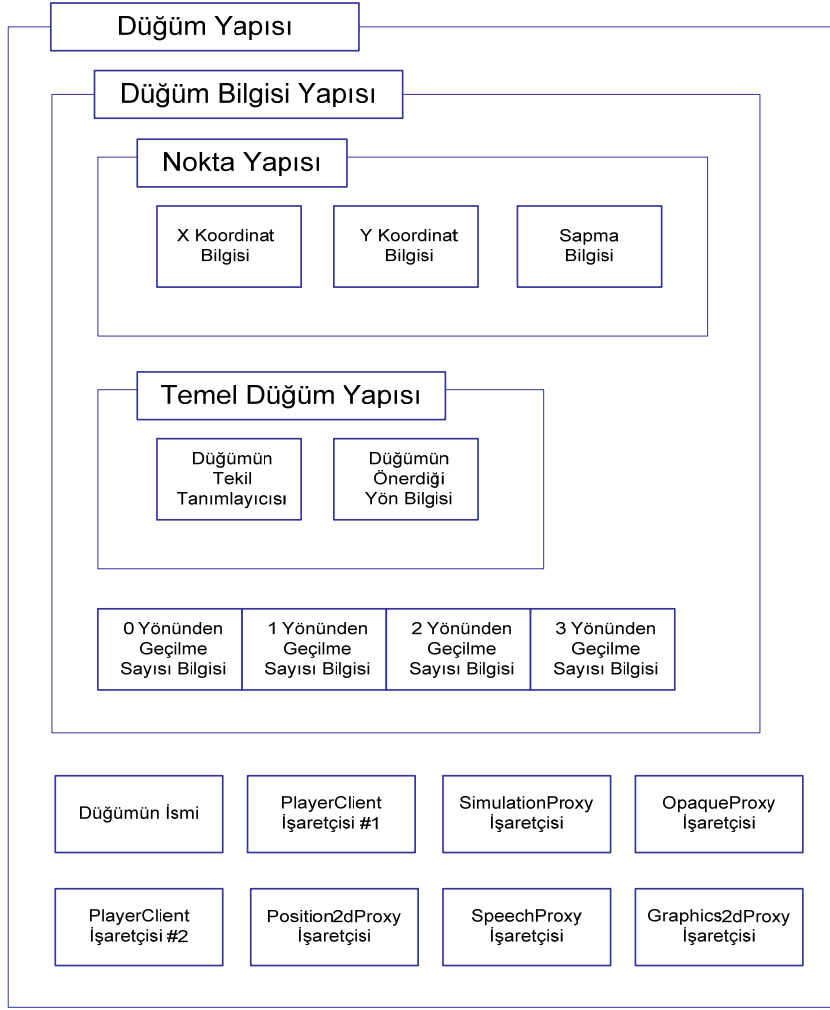
Şekil 5.6'da gösterilen Düğüm Bilgisi Yapısı, Şekil 5.1'de verilen 4 temel yön bilgisine ait geçiş sayısı bilgisinin, düğüme ait nokta bilgisinin ve düğüme ait temel düğüm bilgileri yapısının tutulduğu yapıdır.



Şekil 5.6 : Düğüm tarafında kullanılan Düğüm Bilgisi Yapısı.

Düğüm Yapısı:

Şekil 5.7'de gösterilen Düğüm Yapısı, düğüm tarafında tanımlı olan ve kullanılan Player arayüzü (2 adet), Position2d arayüzü, Opaque arayüzü, Simulation arayüzü, Speech arayüzü, Graphics2d arayüzü ve Düğüm Bilgisi Yapısı ile düğüme ait isim bilgisinin tutulduğu yapıdır.



Şekil 5.7 : Düğüm tarafında kullanılan Düğüm Yapısı.

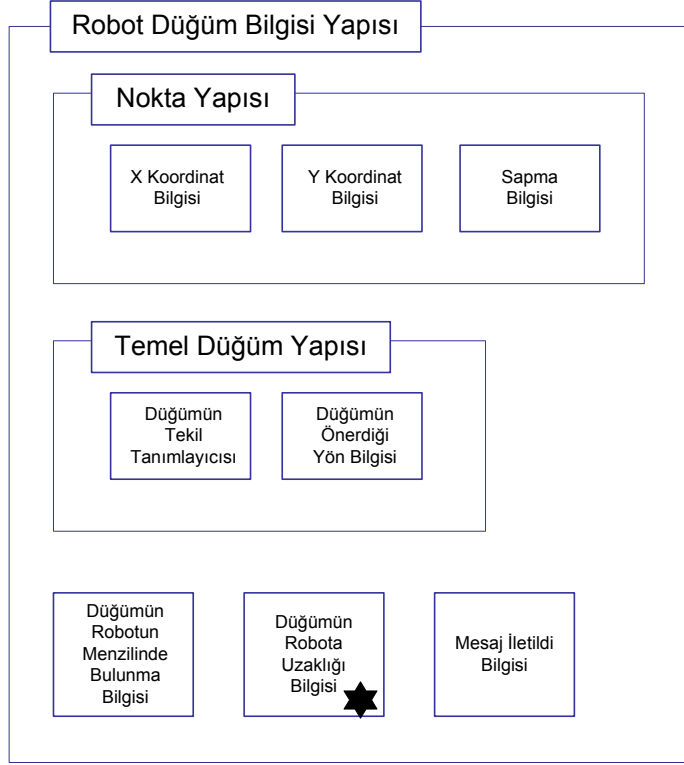
5.2.3.3 Robot tarafında kullanılan veri yapıları

Şekil 5.4'te verilen Nokta Yapısı ve Şekil 5.5'te verilen Temel Düğüm Yapısı robot ve düğüm tarafında ortak olarak kullanılan veri yapılarıdır.

Robot Düğüm Bilgisi Yapısı:

Şekil 5.8'de gösterilen Robot Düğüm Bilgisi Yapısı, robotun haberleştiği düğümüne ait nokta bilgisinin, düğümüne ait temel düğüm bilgileri yapısının, mesajın ulaştığını belirten, düğümün robota uzaklığını belirten ve robotun düğümün haberleşme menziline bulunduğunu belirten bilgilerin tanımlandığı yapıdır. Yıldızla işaretli olarak belirtilen bilgi dışındaki bütün bilgiler robotta tutulmaktadır. ★'la belirtilen düğümün robota uzaklığını belirten bilgi gerçekleştirme detayıdır ve simülasyonda

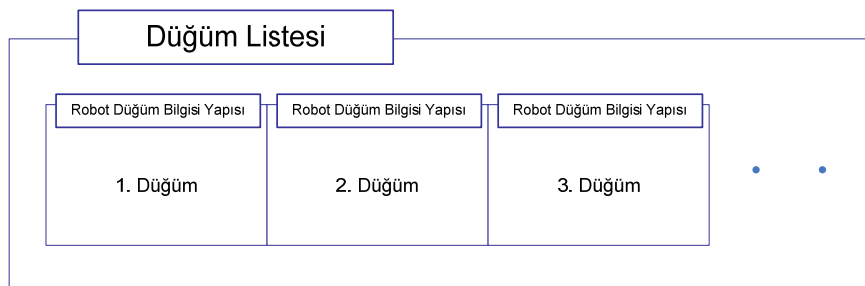
tutulmaktadır. Gerçek robot ve duyarlı bir sistemde bu bilgiye net bir şekilde ulaşamaz ama ortamın fiziksel doğası gereği işlem yerine getirilebilir. Benzetim ortamındaki gerçekleştirme için bu bilgiye ihtiyaç vardır.



Şekil 5.8 : Robot tarafında kullanılan Robot D ğ m Bilgisi Yapısı.

D ğ m Listesi:

Şekil 5.8’de verilen Robot D ğ m Bilgisi Yapı’larından oluşan listedir ve robotun haberleştiđi b t n d ğ mlere ait bilgiler D ğ m Listesi olarak adlandırılan ve Şekil 5.9’da g sterilen yapıda tutulur. Bu bilgi de robot tarafında tutulmayan bir gereklemedir ve sim lasyonda tutulmaktadır.



Şekil 5.9 : Robot tarafında kullanılan D ğ m Listesi Yapısı.

Robot tarafında tanımlı olan ve kullanılan Player, Position2d, Opaque, Simulation, Speech ve Sonar arayüzleri bilgileri ile Robotun ismi bilgisi Şekil 5.10'da gösterilen diğer veri yapılarını oluşturmaktadır.

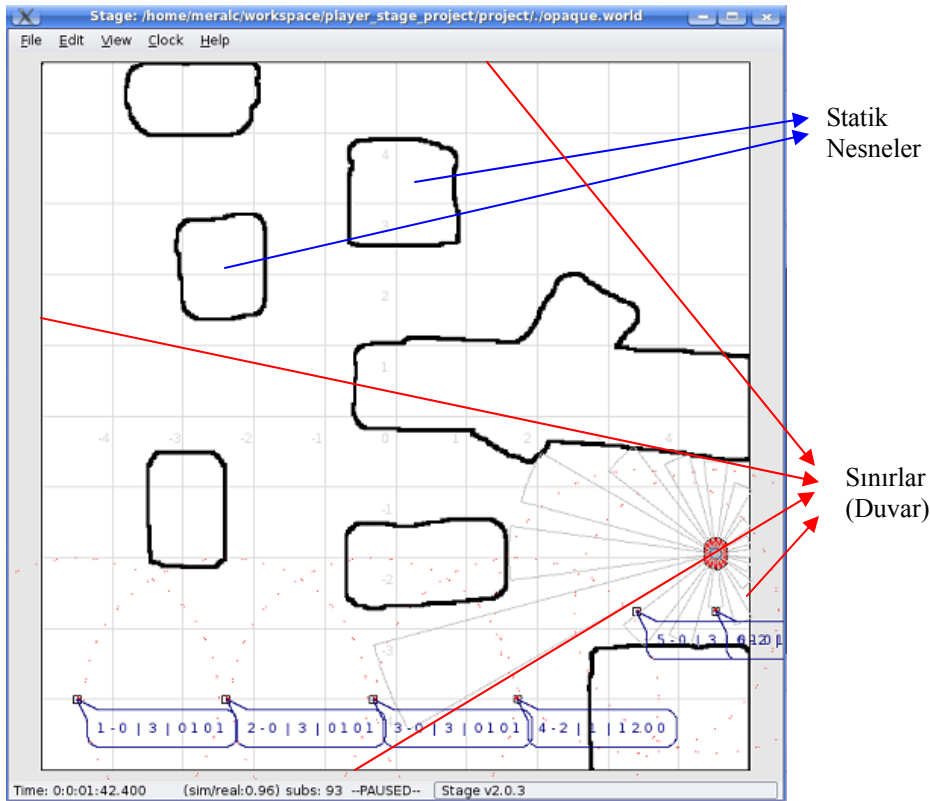


Şekil 5.10 : Robot tarafında kullanılan diğer veri yapıları.

5.3 E-LRV Yöntemi Bileşenleri

E-LRV yönteminin geliştirilmesinde Engel Tanımı ve Sakınımı, Düğüm Bırakma ve Mesaj Yapıları olmak üzere üç temel kısım üzerinde çalışılmıştır.

5.3.1 Engel tanımı ve sakınımı



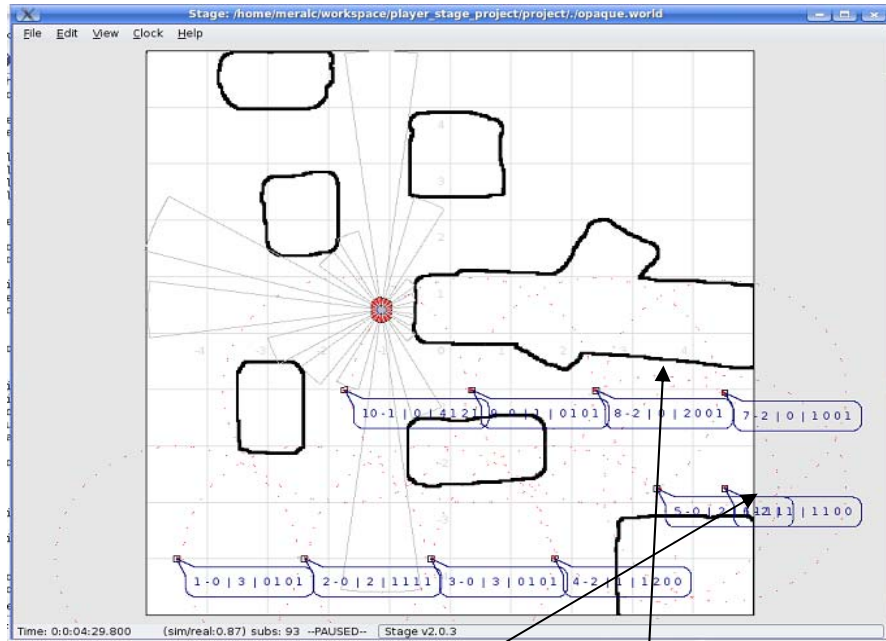
Şekil 5.11 : Engel tanımı: statik nesnelar ve ortam sınırları (duvar).

Engeller (Şekil 5.11) robotun ilerlemesini engelleyen statik nesnelere ve sınırlar (duvar) olabilirken düğümlerin kendisi de engel kapsamında değerlendirilebilir. Benzetim ortamında düğümlerin engel olarak değerlendirilmemesi için robotun engelle karşılaştığında o engelin düğüm olduğunun anlaşılması için düğümle arasındaki uzaklığa da bakılabilir. Engelle karşılaştığında düğümün engel olarak değerlendirilmemesi bu şekilde sağlanabilir. Gerçek ortamda düğümün engel olarak algılanmaması için o düğüme mesaj gönderilerek ve ondan gelen cevaba göre onun düğüm olduğu anlaşılabilir.

Robot ilerlediği yönde engel olduğunu sonar duyargalarının verilerini değerlendirerek anlar. Robotun gittiği yönde [4] numaralı sonar duyargası yer almaktadır (Şekil 5.2). Robot gittiği yönün ± 45 derecesinde bulunan sonar duyargalarını kontrol ederek engel olup olmadığını anlar.

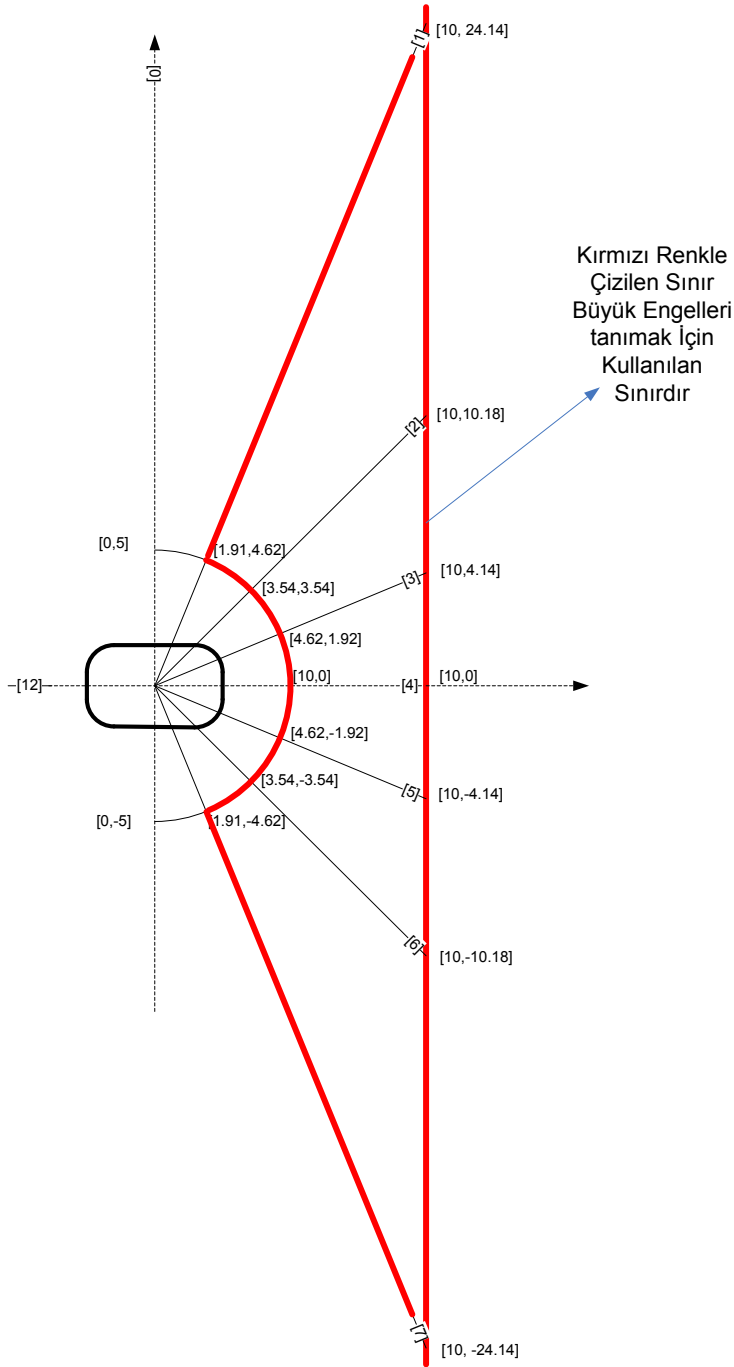
$[2(+45^\circ)]$, $[3(+22.5^\circ)]$, $[4(0^\circ = \text{robotun yönü})]$, $[5(-22.5^\circ)]$, $[6(-45^\circ)]$ yönlerindeki sonar duyargaları engel tanımı ve sakınımı için genel olarak kullanılan sonar duyargalarıdır (Şekil 5.2).

5.3.1.1 Büyük engel tanımı



Büyük engel: Sınır(duvar) ya da statik nesne

Şekil 5.12 : Büyük engel.



Şekil 5.13 : Büyük engel değerlendirme detayı.

Robotun karşılaştığı engeli büyük bir engel (Şekil 5.12) olarak tanınması için engel sakınımında ve değerlendirilmesinde kullanılan $[2(+45^\circ)]$, $[3(+22.5^\circ)]$, $[4(0^\circ = \text{robotun yönü})]$, $[5(-22.5^\circ)]$, $[6(-45^\circ)]$ sonar duyargalarının hepsinin belirli bir eşik değerinden küçük olduğunu belirlemesi gerekmektedir. Büyük engeller bu 5 sonar duyargasının değerlerinin açılara göre farklılık gösteren bir şekilde belirli bir aralığa düşmesi değerlendirilerek belirlenir (Şekil 5.13).

5.3.1.2 Engel sakınımı

```
if ([0] numaralı sonar verisi [8] numaralı sonar verisinden büyükse)

    if ([14],[15],[0],[1] numaralı sonar duyarga verileri engel sakınımı mesafesinden büyükse)

        Robot (+90 °) ([0] nolu sonar duyargası) yönüne döner.

    else if ([8],[9],[10],[11] numaralı sonar duyarga verileri engel sakınımı mesafesinden büyükse)

        Robot (-90 °) ([8] nolu sonar duyargası) yönüne döner.

    else Robot (+/-180 °) ([12] nolu sonar duyargası) yönüne döner

else ([8] numaralı sonar verisi [0] numaralı sonar verisinden büyükse)

    if ([8],[9],[10],[11] numaralı sonar duyarga verileri engel sakınımı mesafesinden büyükse)

        Robot (-90 °) ([8] nolu sonar duyargası) yönüne döner.

    else if([14],[15],[0],[1] numaralı sonar duyarga verileri engel sakınımı mesafesinden büyükse)

        Robot (+90 °) ([0] nolu sonar duyargası) yönüne döner.

    else Robot (+/-180 °) ([12] nolu sonar duyargası) yönüne döner.
```

Şekil 5.14 : Engel sakınımı algoritması akışı.

Engel sakınımında kullanılan 5 sonar duyarganın verisi sahip oldukları açılara göre farklılık gösteren bir şekilde kendilerine düşen aralığa göre değerlendirilir. Bu sonar duyargalarından herhangi bir tanesi kendine özgü tanımlanan değerden küçük olduğunda engel sakınımı durumu gerçekleştirilmektedir.

E-LRV’de gerçekleşen engel sakınımı algoritması akışı Şekil 5.14’te verilmiş ve aşağıda anlatılmıştır. Aynı algoritma akışı LRV yöntemi için de kullanılmaktadır.

Engel sakınımı işlemi robotun gittiği yön dikkate alınarak (Çizelge 5.1) [2],[3],[4(robotun yönü)],[5],[6] sonar duyargalarının verilerinin büyüklüklerine göre

[14],[15],[0],[1],[7],[8],[9],[10] sonar duyargalarının verileri de kontrol edilerek ilgili yöne (4 temel yönden 3'üne $(+90^\circ),(-90^\circ),(180^\circ)$) dönmesi şeklinde gerçekleştirilmektedir. Robotun o anda geldiği yön doğrultusunda engel olduğu için bu yön (0°) gidilecek yönlerin dışında bırakılmıştır.

Robot gittiği yönde engelle karşılaştığında [0] ve [8] numaralı sonar duyarga verilerinin büyüklüklerine göre [0] numaralı sonar duyarga verisi [8] numaralı sonar duyarga verisinden büyükse [0] numaralı sonar duyargasının içinde bulunduğu [14],[15],[0],[1] sonar duyarga kümesi verilerini öncelikle kontrol ederek, bu küme yönünde engel yoksa $(+90^\circ)$ yönüne gider. Bu küme yönünde engel varsa ikinci küme olan [7],[8],[9],[10] sonar duyarga kümesi verilerini kontrol ederek, bu ikinci küme yönünde engel yoksa (-90°) yönüne gider. İkinci küme yönünde de engel varsa robot geldiği yönün tam 180° tersine dönerek o yönde ilerler. [8] numaralı sonar duyarga verisi [0] numaralı sonar duyarga verisinden büyükse yukarıda anlatılan durum öncelikle 2. küme sonra 1. küme kontrol edilerek yapılır (Şekil 5.14).

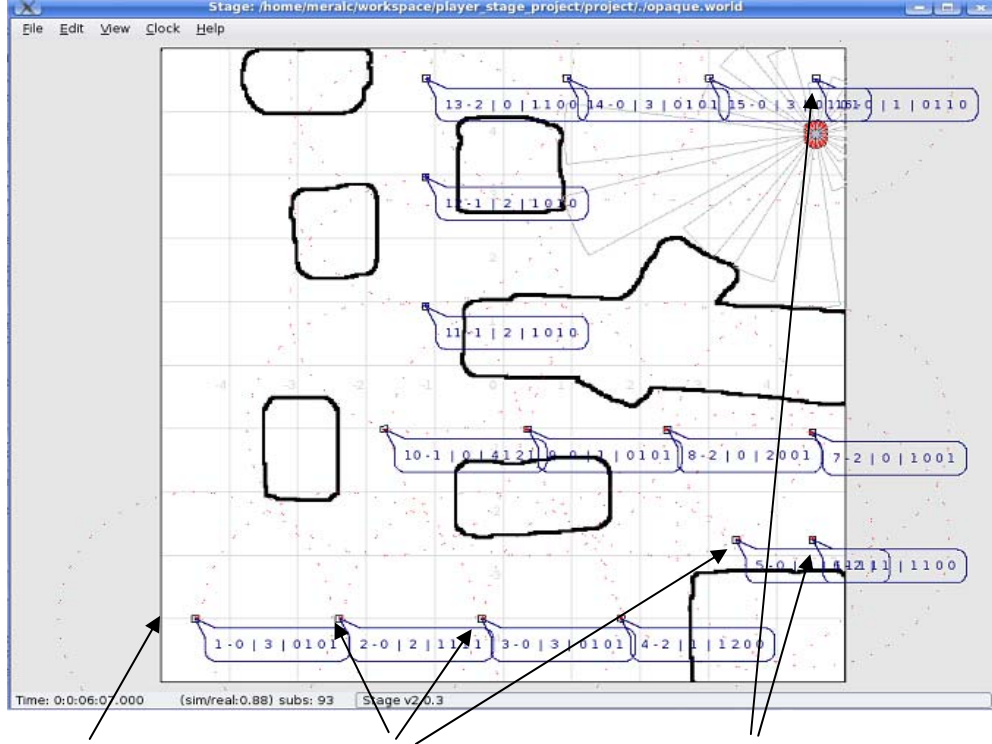
Çizelge 5.1 : Sonar duyarga numaraları ve açısal değerleri.

Sonar Duyarga Numarası	Açısal Değeri
0	(+) 90°
1	(+) 67.5°
2	(+) 45°
3	(+) 22.5°
4	$0^\circ \rightarrow$ Robot Yönü
5	(-) 22.5°
6	(-) 45°
7	(-) 67.5°
8	(-) 90°
9	(-) 112.5°
10	(-) 135°
11	(-) 157.5°
12	180°
13	(+) 157.5°
14	(+) 135°
15	(+) 112.5°

5.3.2 Düşüm bırakma

Düşüm bırakma işlemi LRV ve E-LRV yöntemlerinde iki farklı durumda gerçekleştirilir (Şekil 5.15).

- Robotun çevresinde etkileşimde bulunduğu başka bir düğüm yoktur.
- Robot büyük bir engelle karşılaşmış ve çevresinde güncel düğümden başka bir düğüm bulunmamaktadır.



Robotun ortamı taramaya başladığı yer

Robotun çevresinde etkileşimde bulunduğu başka bir düğüm yoktur

Robot büyük bir engelle karşılaşmış ve çevresinde güncel düğümden başka bir düğüm bulunmamaktadır.

Şekil 5.15 : Düğüm bırakma.

Robotun gittiği yön bilgisi, x ve y koordinatı bilgisi düğüm bırakma ön işleminde kullanılır. Akışı Şekil 5.16'da verilen düğüm bırakma ön işlemi bu çalışma gerçek bir robot uygulaması olmadığı için ve Player/Stage benzetim ortamında robot ve bıraktığı düğüm aynı yerde bulunamayacağı için robotun düğümü bırakması için öncelikle nereye bırakacağını belirlemesi ve o anda elde edilen sonar verilerine uygun olarak uygun yere gitmesi için kullanılmaktadır.

Bu işlemi gerçekleştirirken robot yukarıda bahsedilen benzetim ortamından kaynaklanan kısıtlamadan dolayı bir çeşit engel sakınımı yaparak düğümü bıraktıktan sonra gideceği yönü kontrol ederek hareket eder ve düğümü bırakır. Sonar duyurga

verileri kontrol edilerek robotun düğümü bıraktıktan sonra bulunacağı noktada engelden sakınamayacağı bir durumla karşılaşmaması için sırayla kontrol edilecek sonar duyargalarının belirli bir değerden ($D_{\text{düğümBırakma}}$) büyük olması gerekmektedir.

```

if ([4] numaralı sonar duyargası verisi >  $D_{\text{düğümBırakma}}$  ) then

    Robot  $D_{\text{düğümUzaklasma}}$  kadar uzaklaşır.

    (Bırakılacak düğümün yönü  $Y_{\text{robotYon}}$  dür. Robotun yönü değişmemiştir.
 $Y_{\text{robotYon}}$ )

else if (([0] nolu sonar duyarga verisi >  $D_{\text{düğümBırakma}}$  ) || ([8] nolu sonar duyarga verisi >
 $D_{\text{düğümBırakma}}$ ))

    if ([0] nolu sonar duyarga verisi > [8] nolu sonar duyarga verisi)

        Robot (+90°) ([0] nolu sonar duyargası) yönüne dönerek  $D_{\text{düğümUzaklasma}}$ 
kadar uzaklaşır

        if ([8] nolu sonar duyarga verisi >  $D_{\text{düğümBırakma}}$  )

            Robot ilk yönüne geri döner. ( $Y_{\text{robotYon}}$  )

        else

            Robotla bırakılacak düğümün yönü [0] nolu sonar duyargası yönü
olarak güncellenir.

        else //([8] nolu sonar duyarga verisi > ([0] nolu sonar duyarga verisi)

            Robot(-90°) ([8] nolu sonar duyargası) yönüne dönerek  $D_{\text{düğümUzaklasma}}$ 
kadar uzaklaşır

            if ([0] nolu sonar duyarga verisi >  $D_{\text{düğümBırakma}}$  )

                Robot ilk yönüne geri döner. ( $Y_{\text{robotYon}}$  )

            else

                Robotla bırakılacak düğümün yönü [8] nolu sonar duyargası yönü
olarak güncellenir.

else // robot gittiği yönün tam tersi yönüne dönerek(180°) düğüm bırakacaktır.

    Robot (180 °) ([12] nolu sonar duyargası) yönüne dönerek  $D_{\text{düğümUzaklasma}}$  kadar
uzaklaşır

    Robotla bırakılacak düğümün yönü ([12] nolu sonar duyargası yönü) olarak
güncellenir.

```

Şekil 5.16 : Düğüm bırakma ön işlemi algoritması akışı.

Öncelikle robotun gittiği yön (Y_{robotYon}) doğrultusunda yer alan (0°) (o anda [4] numaralı sonar duyargası yönü) kontrol edilerek bu duyarganın değeri ($V_{\text{sonar}[4]}$)

$D_{\text{dugumBırakma}}$ değerinden büyükse robotun düğümü bıraktıktan sonra uzaklaşması gereken mesafe kadar ($D_{\text{dugumUzaklasma}}$) gitmesi sağlanır.

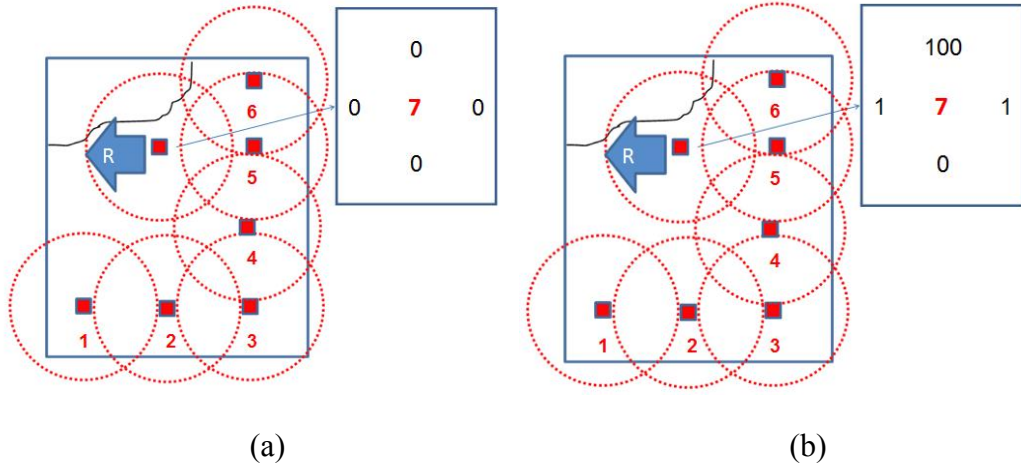
Y_{robotYon} yönünde $D_{\text{dugumBırakma}}$ mesafesinden kısa bir mesafe var ise ($+90^\circ$) (o anda [0] numaralı sonar duyargası yönü) ve (-90°) (o anda [8] numaralı sonar duyargası yönü) yönlerinden herhangi bir tanesinin duyarga değerinin ($V_{\text{Sonar}[0]}$ veya $V_{\text{Sonar}[8]}$) $D_{\text{dugumBırakma}}$ mesafesinden fazla olup olmadığına bakılarak, ($V_{\text{Sonar}[0]}$ ve $V_{\text{Sonar}[8]}$) sonar duyarga verilerinden hangisi daha büyükse robotun o yöne ($+90^\circ$ ya da -90°) dönmesi sağlanır. Robotun $D_{\text{dugumUzaklasma}}$ mesafesi kadar ilerlemesi sağlanır. Döndüğü yönün tam tersi yönündeki duyarganın değeri ($V_{\text{Sonar}[0]}$ veya $V_{\text{Sonar}[8]}$) $D_{\text{dugumBırakma}}$ mesafesinden küçük ise robot yön değiştirmez ($+90^\circ$ ya da -90° yönünde kalır). Döndüğü yönün tam tersi yönündeki duyarganın değeri ($V_{\text{Sonar}[0]}$ veya $V_{\text{Sonar}[8]}$) $D_{\text{dugumBırakma}}$ mesafesinden büyük ise Y_{robotYon} yönüne döner.

D_{robotYon} yönünde ve [0] numaralı sonar duyargası yönü ile [8] numaralı sonar duyargası yönü yönlerinde $D_{\text{dugumBırakma}}$ mesafesinden kısa bir mesafe varsa robot (180°) (o anda [12] numaralı sonar duyargası yönü) yönüne döner, $D_{\text{dugumUzaklasma}}$ değeri kadar uzaklaşır.

Düğüm bırakmadan önce gerekli işlemler Şekil 5.16'da akışı verilen şekilde gerçekleştirildikten sonra, bırakılacak olan düğüm için gerekli bilgilerle **Konum-Mesajı** oluşturularak ilgili düğümüne mesaj gönderilir. **Konum-Mesajı** gönderimi ve düğümlerin belirtilen yerlere robot tarafından bırakılması LRV ve E-LRV yöntemlerinin ikisinde de aynı şekilde gerçekleştirilmektedir. Robot tarafından gönderilen **Konum-Mesajı**'nı alan düğüm belirtilen konuma robot tarafından bırakılır ve aktif hale gelerek bundan sonra robotun bu düğümün haberleşme menziline girdiği zaman gerekli yön önerisini yine mesajla robota gönderir.

Düğüm bırakılırken E-LRV yönteminde ortamın engel durumunun gözetilmesi ve buna uygun güncelleme mesajlarının robottan düğümüne gönderilmesi, LRV yönteminde gerçekleştirilmemektedir. Aynı şekilde düğümü bırakırken geçilen yön bilgileri LRV yönteminde güncellenmezken, E-LRV yönteminde bu yönlerin güncellenmesi için düğümüne mesaj gönderilmektedir.

Şekil 5.17 (a) ve (b)' de LRV ve E-LRV yönteminde düğüm bırakma işlemlerinin basit bir senaryosu gösterilmektedir.



Şekil 5.17 : LRV ve E-LRV düğüm bırakma stratejileri.

Şekil 5.17 (a)'da verilen senaryoda LRV yöntemine göre ortama düğüm bırakıldığında robotun sağdan sola giderken bıraktığı 7 numaralı düğümünün geçilen yönleri güncellenmemektedir. Bu yöntemde ortam taramanın etkin bir şekilde gerçekleştirilmesi için gereken bir koşul olan ortamda elde edilen tüm bilgilerin kullanılmadığı görülmektedir. 7 numaralı düğümün yukarısında engel bulunmaktadır ve bu bilgi herhangi bir şekilde kullanılmadığı için robot herhangi bir şekilde bu düğümün menziline tekrar girdiğinde düğüm herhangi bir yön seçme kuralına göre önerdiği yön bilgisi olarak yukarı yönü robota önerebilmektedir. Bu durum robot bu düğümün menziline her girdiğinde gereksiz yere yön güncelleme mesajlarının düğümüne gönderilmesine yol açmaktadır. Ortamın tamamının taranması düşünüldüğünde bu durum toplam mesaj trafiğini artırıcı bir durum yaratmaktadır. Aynı zamanda robotun düğümü bırakırken geçtiği yönlerin güncellenmemesinden kaynaklanan yönler (sola ya da sağa) yönlendirilmesi de mümkün olabilmektedir.

Şekil 5.17 (b)'de verilen senaryoda E-LRV yöntemine göre ortama düğüm bırakıldığında robotun sağdan sola giderken bıraktığı 7 numaralı düğümünün geçilen yönleri güncellenerek ortamda elde edilen tüm bilgilerin kullanılması sağlanmaktadır. 7 numaralı düğümün yukarısında engel bulunmaktadır ve bu bilgi 7 numaralı düğümüne gönderilen **Engel-Güncelleme-Mesajı** yardımıyla düğüm tarafında büyük bir değerle güncellenerek, robot herhangi bir şekilde bu düğümün

menziline tekrar girdiğinde düğümün herhangi bir yön seçme kuralına göre önerdiği yön bilgisi olarak yukarı yönü robota önermesi engellenmektedir. Bu durumda her engelli yön için bir mesaj gönderimi yapılmaktadır. Ortamın tamamının taranması düşünüldüğünde bu durum toplam mesaj trafiğini azaltacak bir durum oluşturmaktadır. Aynı zamanda robotun düğümü bırakırken geçtiği yönlerin güncellenmesiyle robotun geçtiği yönlere (sola ya da sağa) yönlendirilmesi durumu çözülmektedir.

Çizelge 4.1’de verilen Çapraz yön seçme kuralının 1 numaralı ögesi {0,2,1,3} ve Şekil 5.1’de verilen yön bilgisi kullanıldığında, robot herhangi bir şekilde 7 numaralı düğümün menziline girdiğinde önerilecek olan yön LRV yöntemine göre **(0)** yönüdür (Şekil 5.17 (a)’da 7 numaralı düğümün yukarısı). Bu yönde engel bulunduğu robot tarafında gereksiz yere düğümüne **Yön-Güncelleme-Mesajı** gönderilmesine sebep olmaktadır. Farklı yön seçme kuralları uygulandığında **(1)** yönü (Şekil 5.17 (a)’da 7 numaralı düğümün sağı) ya da **(3)** yönü (Şekil 5.17 (a)’da 7 numaralı düğümün solu) önerilebilmektedir. Aynı durum E-LRV yöntemi için düşünüldüğünde **(1)** yönü (Şekil 5.17 (b)’de 7 numaralı düğümün sağı) ya da **(3)** yönü (Şekil 5.17 (b)’de 7 numaralı düğümün solu) bilgileri düğüm bırakılırken güncellendiğinden ve **(0)** yönü de (Şekil 5.17 (b)’de 7 numaralı düğümün yukarısı) engelden dolayı güncellendiğinden robota henüz ziyaret edilmemiş olan **(2)** yönü (Şekil 5.17 (b)’de 7 numaralı düğümün aşağısı) önerilecektir.

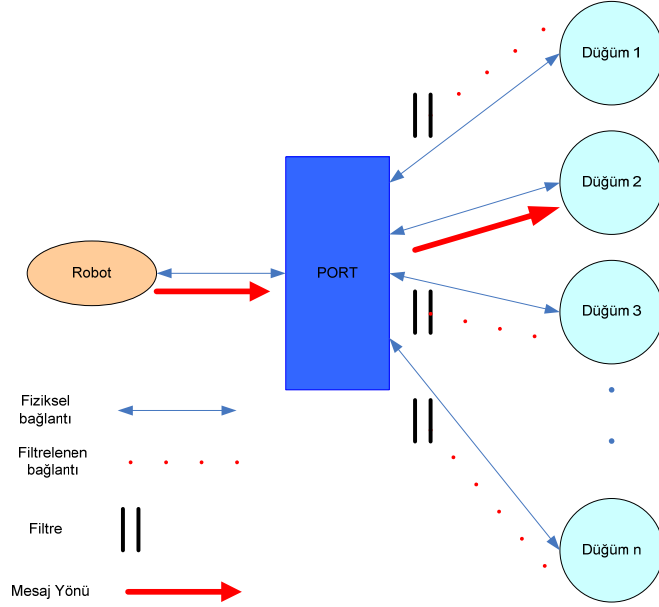
Tez çalışması kapsamında gerçekleştirilen LRV ve E-LRV yöntemlerinde kullanılan mesaj tipleri ve açıklamaları detaylı bir şekilde Bölüm 5.3.3’te yer almaktadır.

5.3.3 Mesaj tipleri ve mesaj akışı

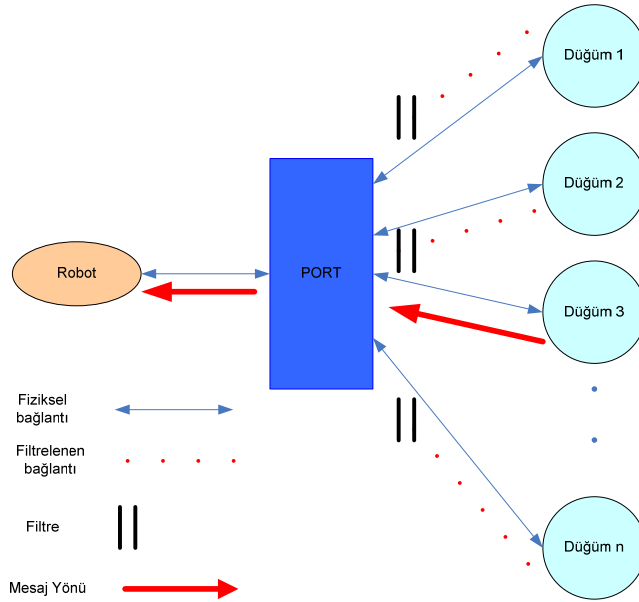
5.3.3.1 Robot düğüm haberleşmesi

Robot ve düğümler arasındaki haberleşme TCP soketleri yardımıyla sağlanmaktadır. Opaque arayüzü yardımıyla kullanıcı tanımlı mesajların gönderilmesi için robot ve düğümler tarafında belirli bir port’un haberleşme için ayrılması gerekmektedir. Şekil 5.18 robot tarafından belirli bir düğümüne gönderilen mesajın akışını göstermektedir. Robot ve düğümlerin her biri tek bir port üzerinden haberleştiği için porta yazılan her mesaj hem robot tarafından hem de düğümlerin her biri tarafından alınmaktadır.

Robot tarafından gönderilen mesajın sadece ilgili düğümde değerlendirilmesi için düğümlerin haberleşme dışında kullandıkları port numaralarına göre (düğümler için sıralı) (Çizelge 5.2) filtreleme işlemi gerçekleştirilmektedir.



Şekil 5.18 : Robottan düğümüne mesaj gönderilmesi.



Şekil 5.19 : Düğümünden robota mesaj gönderilmesi.

Şekil 5.19 belirli bir düğümün robota gönderdiği mesajın akışını göstermektedir ve Şekil 5.18’de yer alan robotun belirli bir düğümüne mesaj göndermesi işleminde kullanılan filtreleme yöntemi burada da kullanılmaktadır. Şekil 5.18 ve Şekil 5.19’da

yer alan port bilgisi, yapılan bu tez çalışmasında 6665 olarak belirlenmiş ve robot-düğüm haberleşmesinde kullanılmıştır. Bu port aynı zamanda Simulation arayüzü için de kullanılmaktadır. Opaque ve Simulation arayüzleri dışında düğümlerde kullanılan diğer arayüzler için, her düğüm için (bu tez kapsamında yapılan gerçekleştirme aşamasında kullanılan) aynı zamanda düğüm numarasını belirtecek şekilde sıralı port numaraları kullanılmıştır (Çizelge 5.2).

Çizelge 5.2 : E-LRV'nin gerçekleştirilmesinde robot ve düğüm tarafında kullanılan port, istemci ve arayüz bilgileri.

Nesne	Port	İstemci		Arayüzler				
		PlayerClient	Opaque	Simulation	Position2d	Speech	Sonar	Graphics2d
Robot	6665	√	√	√	√	√	√	
Düğüm 1	6665	√	√	√				
Düğüm 1	6666				√	√		√
Düğüm 2	6665	√	√	√				
Düğüm 2	6667				√	√		√
.....							

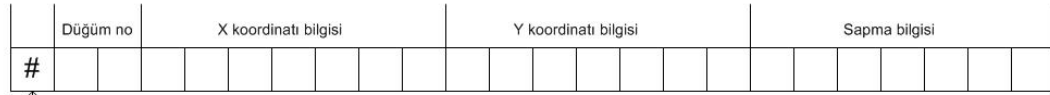
6665 -> Robot ve düğümler arası haberleşmeyi sağlayan Opaque arayüzünün ve benzetim ortamına bağlanmayı sağlayan Simulation arayüzünün Player sunucusuna bağlanması için kullanılan porttur.

5.3.3.2 Mesaj tipleri

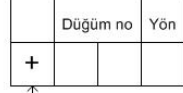
Çizelge 5.3 : LRV ve E-LRV'nin gerçekleştirilmesinde robot ve düğüm tarafında karşılıklı yollanan mesaj tipleri.

Robottan Düğüme Gönderilen Mesajlar	Düğümde Robotla Gönderilen Mesajlar	Kullanıldığı Yöntem
Konum-Mesajı	Konum-Mesajı-Cevabı	LRV ve E-LRV
Periyodik-Bilgi-Mesajı	Periyodik-Bilgi-Mesajı-Cevabı	LRV ve E-LRV
Yön-Güncelleme-Mesajı	Yön-Güncelleme-Mesajı-Cevabı	LRV ve E-LRV
Engel-Güncelleme-Mesajı	Engel-Güncelleme-Mesajı-Cevabı	E-LRV

Bu kısımda Robot tarafından düğüme gönderilen **Konum-Mesajı**, **Periyodik-Bilgi-Mesajı**, **Yön-Güncelleme-Mesajı** ve **Engel-Güncelleme-Mesajı** ile düğüm tarafından robota gönderilen **Konum-Mesajı-Cevabı**, **Periyodik-Bilgi-Mesajı-Cevabı**, **Yön-Güncelleme-Mesajı-Cevabı** ve **Engel-Güncelleme-Mesajı-Cevabı** formatları anlatılmaktadır (Çizelge 5.3 ve Şekil 5.20).

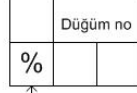


(a) Konum-Mesajı



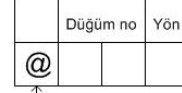
Mesaj Belirteci

(b) Konum-Mesajı-Cevabı



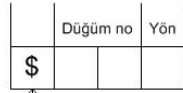
Mesaj Belirteci

(c) Periyodik-Bilgi-Mesajı



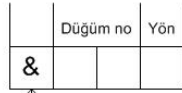
Mesaj Belirteci

(d) Periyodik-Bilgi-Mesajı-Cevabı



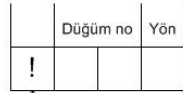
Mesaj Belirteci

(e) Yön-Güncelleme-Mesajı



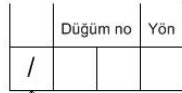
Mesaj Belirteci

(f) Yön-Güncelleme-Mesajı-Cevabı



Mesaj Belirteci

(g) Engel-Güncelleme-Mesajı

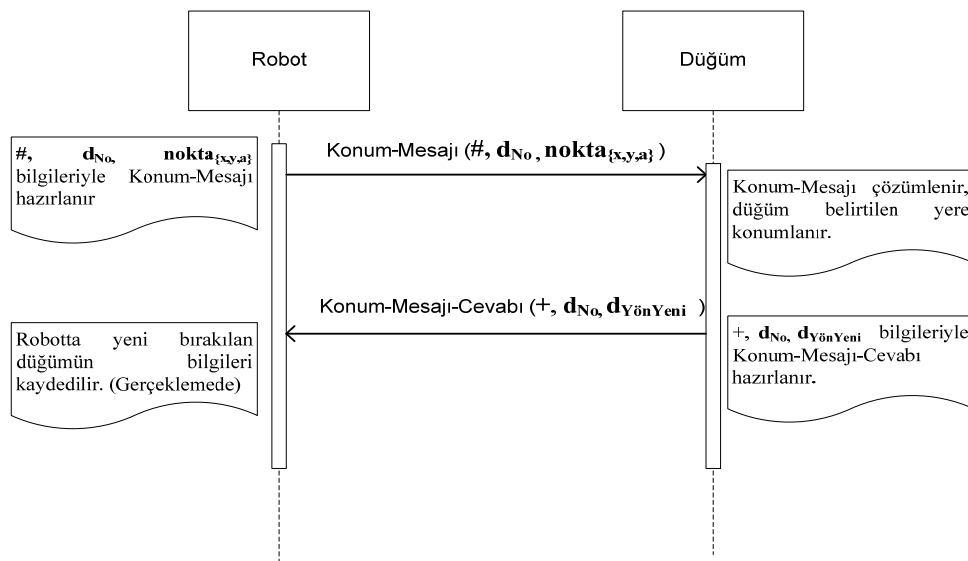


Mesaj Belirteci

(h) Engel-Güncelleme-Mesajı-Cevabı

Şekil 5.20 : Mesaj tipleri.

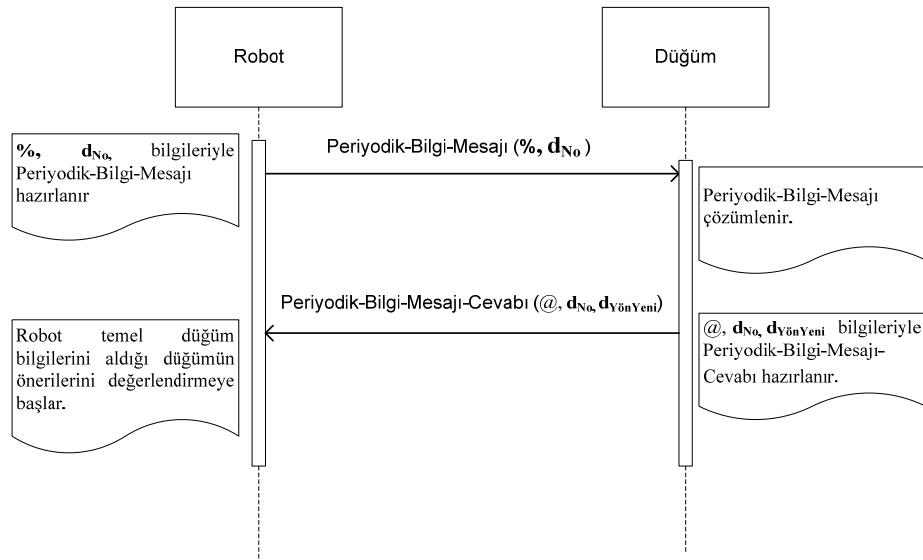
Mesajların başlangıcında yer alan ilk kısım her mesajın tekil bir şekilde değerlendirilmesi ve diğer mesajlardan ayrılması için o mesaja özgü olan bir belirteci içermektedir.



Şekil 5.21 : Konum-Mesajı ve Konum-Mesajı-Cevabı Akışı.

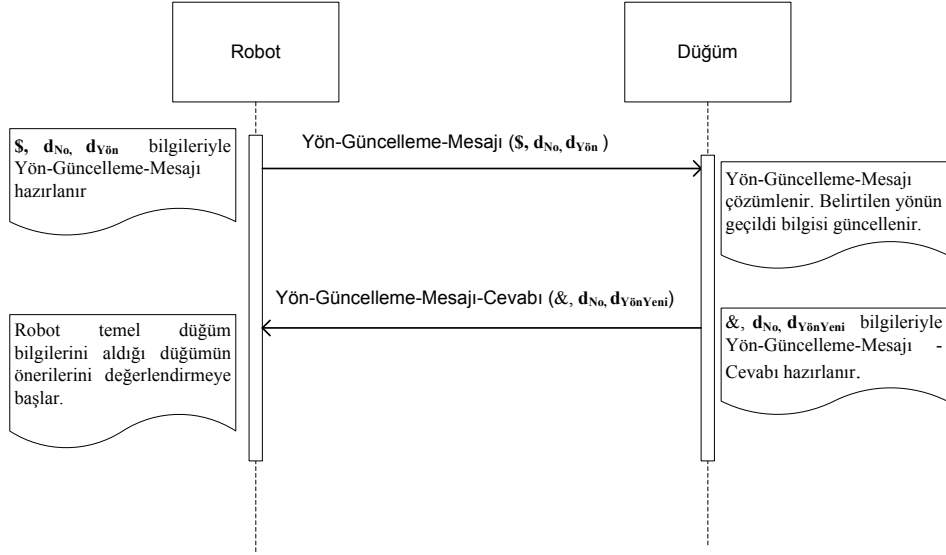
Konum-Mesajı ve **Konum-Mesajı-Cevabı** mesajları robotun ortama bir düğüm bıraktığı zaman düğümüne gönderdiği ve düğümün mesajda belirtilen yere konumlanmasının ardından robota önerdiği yön bilgisini gönderdiği mesajlardır. Bu mesaj çifti LRV ve E-LRV de aynı durum için kullanılmaktadır (Şekil 5.21).

Periyodik-Bilgi-Mesajı ve **Periyodik-Bilgi-Mesajı-Cevabı** mesajları robotun her döngüde menziline bulunduğu düğümlerden güncel bilgilerini almak amacıyla düğümlere gönderilen ve düğümlerin güncel bilgilerini robota göndermesi için kullanılmaktadır. Bu mesaj çifti LRV ve E-LRV de aynı durum için kullanılmaktadır (Şekil 5.22).



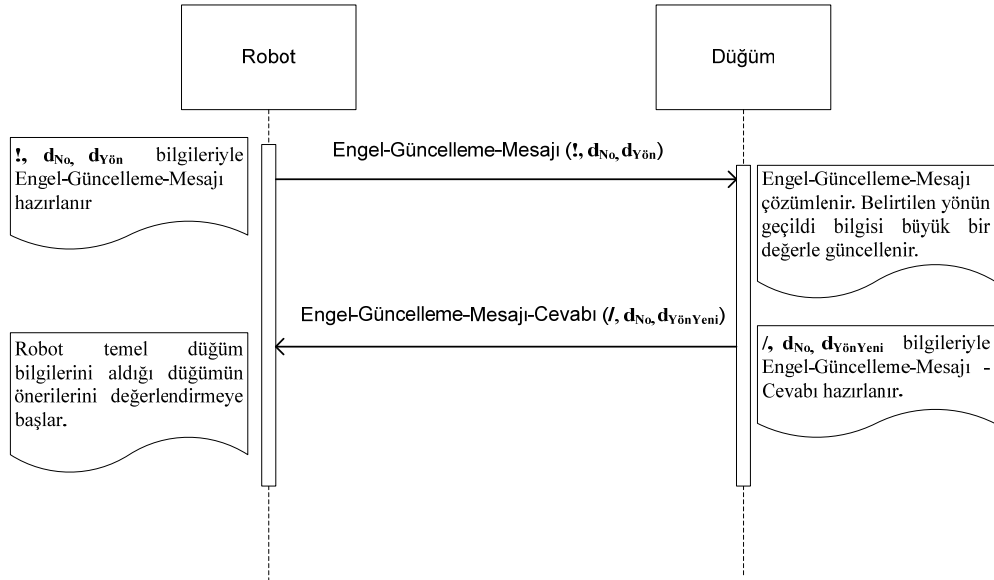
Şekil 5.22 : Periyodik-Bilgi-Mesajı ve Periyodik-Bilgi-Mesajı-Cevabı Akışı.

Yön-Güncelleme-Mesajı ve **Yön-Güncelleme-Mesajı-Cevabı** mesajları robotun herhangi bir şekilde menziline girdiği düğümün önerdiği yönde engel varsa düğümüne gönderdiği ve düğümün bu yönü güncelleyerek bütün yönleri içinde en az ziyaret edilen yön bilgisini robota gönderdiği mesajlardır. Bu mesaj çifti aynı zamanda robotun güncel düğümün menzili dışındayken ortamda kendisine en yakın olarak bulunan düğümün önerdiği yön ve robotun önceki yön bilgilerinin düğüm tarafında güncellenmesi için kullanılmaktadır. Bu iki kullanım şekli LRV ve E-LRV yöntemlerinde aynı şekilde gerçekleştirilmektedir. E-LRV yönteminde LRV yönteminden farklı olarak düğüm bırakılması sırasında robotun geçtiği yönlerin güncellenmesinde de kullanılmaktadır (Şekil 5.23).



Şekil 5.23 : Yön-Güncelleme-Mesajı ve Yön-Güncelleme-Mesajı-Cevabı Akışı.

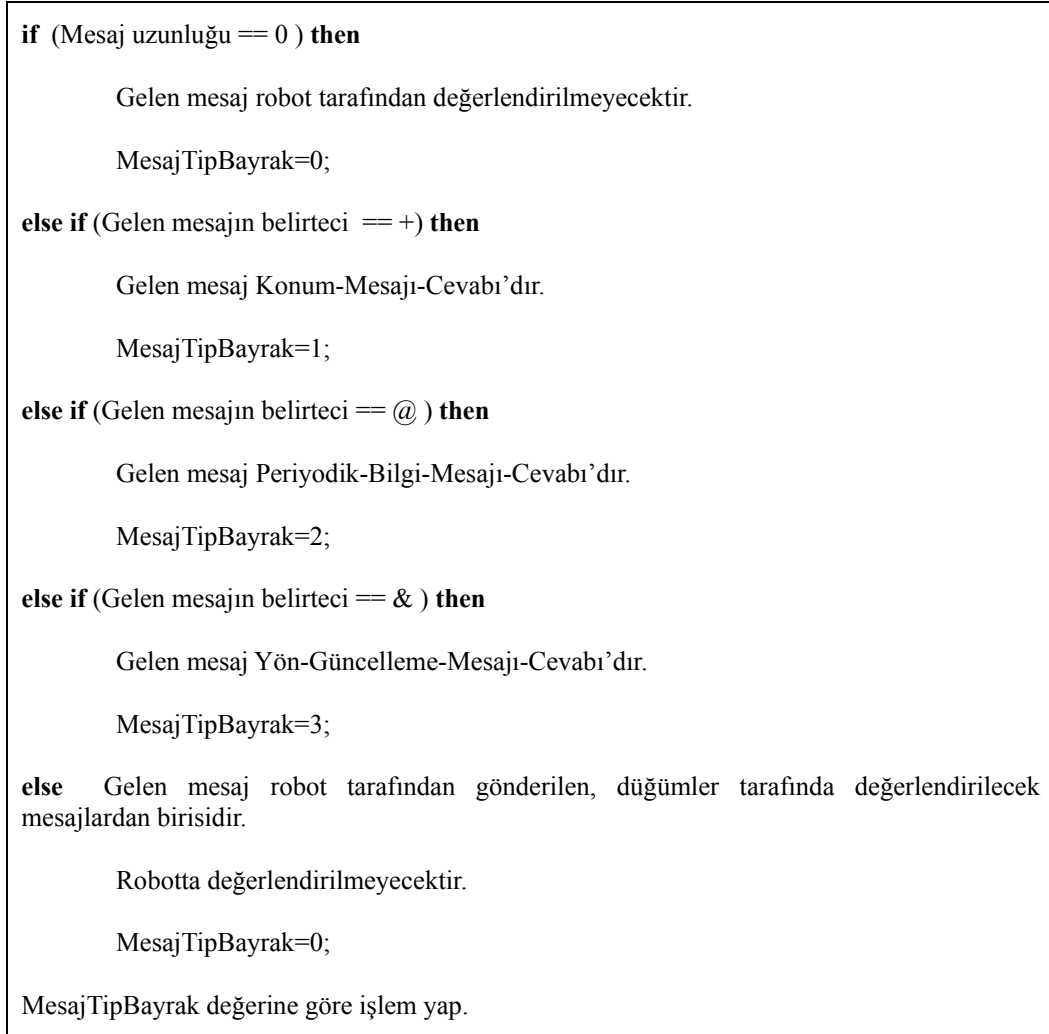
Engel-Güncelleme-Mesajı ve **Engel-Güncelleme-Mesajı-Cevabı** mesajları sadece E-LRV de kullanılan ve robotun düğüm bırakması sırasında geçtiği yönlerin dışındaki yönlerde engel bulunması durumunda düğümüne gönderilen, düğümün bu yönü büyük bir değerle güncelleyerek bütün yönler içinde en az ziyaret edilen yön bilgisini robota göndermesi için kullanılmaktadır (Şekil 5.24).



Şekil 5.24 : Engel-Güncelleme-Mesajı ve Engel-Güncelleme-Mesajı-Cevabı Akışı.

5.3.3.3 Robot tarafında mesaj tipinin belirlenmesi

E-LRV ve LRV yönteminde robot ve düğümlerin haberleşmesini sağlayan mesajları ilk olarak gönderen taraf robot tarafıdır ve bu haberleşme Opaque arayüzü kullanılarak gerçekleştirilir. Robot, düğümün yapmasını istediği iş için gerekli bilgileri düğümler için tanımlı mesajlardan herhangi bir tanesinin formatına uygun hale getirerek düğüme gönderir.



Şekil 5.25 : Robot tarafında mesaj tipinin belirlenmesi akışı.

Robota herhangi bir anda ulaşan mesaj, robot için tanımlanan ve robotta değerlendirilen **Konum-Mesajı-Cevabı**, **Periyodik-Bilgi-Mesajı-Cevabı** ve **Yön-Güncelleme-Mesajı-Cevabı** mesajlarından herhangi bir tanesinin formatına uygun değilse robotun kendi gönderdiği **Konum-Mesajı**, **Periyodik-Bilgi-Mesajı** ve **Yön-Güncelleme-Mesajı** mesajlarından birisidir, bu mesaj robotta değerlendirilmeyecektir.

Robot, gelen mesajın uzunluk kontrolünün ardından mesajın robotta değerlendirilecek mesajlardan herhangi bir tanesine uygun olup olmadığını, gelen mesajın Mesaj-Tipi-Belirtecini kontrol ettikten sonra belirler ve o mesaj geldiğinde robotta yapılacak işleri belirleyen mesaj tipi bilgisini günceller. Robot düğüme mesaj göndermeden önce o mesajı göndereceği düğüme ait tekil tanımlayıcı bilgisini saklayarak, düğüm tarafından gönderilen ve robotta değerlendirilecek cevap mesajının içerdiği düğüme ait tekil tanımlayıcı bilgisi ile karşılaştırarak farklı bir mesajın cevabı geldiyse o mesajı değerlendirmeye almaz. Tekil tanımlayıcı bilgileri aynı olan mesaj ve cevabı eşleştirilerek o mesajın gerçekleştireceği işlem tamamlanır.

Şekil 5.25'te robot tarafında, düğüm tarafından gönderilen mesajın tipinin nasıl belirleneceğinin belirtildiği akış yer almaktadır. Bu akışta LRV ve E-LRV yöntemlerinde ortak olarak kullanılan mesaj tipleri yer almaktadır.

E-LRV yönteminde kullanılan **Engel-Güncelleme-Mesajı-Cevabı**, bu yöntem kullanıldığında bu akışa eklenmektedir.

5.3.3.4 Düğüm tarafında mesaj tipinin belirlenmesi

Robot tarafından gönderilen mesajlar Opaque arayüzü yardımıyla düğümler tarafından alınır. Gelen mesajın uzunluk kontrolünün ardından geçerli bir uzunluğa sahip olan bir mesaj gelmişse ve gelen mesaj o düğümü ilgilendiren bir mesajsa (düğüm numarasına bakılır) bu mesajın düğümler için tanımlanan geçerli bir mesaj olup olmadığı kontrol edilir.

Düğüme ulaşan mesaj, düğümler için tanımlanan ve düğümlerde değerlendirilen **Konum-Mesajı**, **Periyodik-Bilgi-Mesajı** ve **Yön-Güncelleme-Mesajı** mesajlarından herhangi bir tanesinin formatına uygun değilse düğümlerin kendi gönderdiği **Pozisyon-Mesajı-Cevabı**, **Periyodik-Bilgi-Mesajı-Cevabı** ve **Yön-Güncelleme-Mesajı-Cevabı** mesajlarından birisidir ve bu mesaj, mesajı alan düğümde değerlendirilmeyecektir. Düğümler için tanımlanan mesajlardan herhangi bir tanesinin formatına uygun olan mesajdan düğüme ait tekil tanımlayıcı bilgisi elde edilerek bu bilginin mesajı alan düğümün tekil tanımlayıcısına uygun olup olmadığının kontrolünün ardından mesajın hangi mesaj tipinde olduğunu belirten Mesaj-Tipi-Belirteci bilgisi kontrol edilir. Mesajın tipine uygun mesaj tipi bilgisi

güncellenir ve bu mesaj geldiğinde yapılacak olan işlemler mesajı alan düğümde gerçekleştirilir.

Şekil 5.26'da düğüm tarafında, robot tarafından gönderilen mesajın tipinin nasıl belirleneceğinin belirtildiği akış yer almaktadır. Bu akışta LRV ve E-LRV yöntemlerinde ortak olarak kullanılan mesaj tipleri yer almaktadır.

E-LRV yönteminde kullanılan **Engel-Güncelleme-Mesajı-Cevabı**, bu yöntem kullanıldığında bu akışa eklenmektedir.

```
if ( Mesaj uzunluğu == 0 ) then
    Gelen mesaj düğümler tarafından değerlendirilmeyecektir.
    MesajTipBayrak=0;
else if ( Gelen mesajdaki düğüm no != Mesajı alan düğümün numarası)
    Gelen mesaj o anda aktif olan ve robotla haberleşen başka bir düğüme gönderilmiştir.
    Mesajı alan düğüm tarafından değerlendirilmeyecektir.
    MesajTipBayrak=0;
else if ( Gelen mesajın belirteci == # ) then
    Gelen mesaj Konum-Mesajı'dır.
    MesajTipBayrak=1;
else if ( Gelen mesajın belirteci == % ) then
    Gelen mesaj Periyodik-Bilgi-Mesajı'dır.
    MesajTipBayrak=2;
else if ( Gelen mesajın belirteci == $ ) then
    Gelen mesaj Yön-Güncelleme-Mesajı'dır.
    MesajTipBayrak=3;
else Gelen mesaj düğümlerin gönderdiği, robot tarafında değerlendirilecek mesajlardan birisidir.
    Mesajı alan düğümde değerlendirilmeyecektir.
    MesajTipBayrak=0;
MesajTipBayrak değerine göre işlem yap.
```

Şekil 5.26 : Düğüm tarafında mesaj tipinin belirlenmesi akışı.

6. DENEYSEL SONUÇLAR

LRV ve E-LRV yöntemleri Player/Stage benzetim ortamında sonar duyargaları ile donatılmış Pioneer 2DX gezgin robot modeli kullanılarak gerçekleştirilmiştir.

Bu iki yöntemin karşılaştırılmasında kullanılan kriterler kapsama oranı yüzdesi(%), toplam tarama süresi ve ortama bırakılan düğüm sayısıdır.

Toplam kapsama oranı, bir oran şeklinde hesaplanmış olup, ortamda belirli bir orana göre alınan her pixelin herhangi bir düğümün menziline girip girmemesine bağlı olarak kapsanmayan alanın hesaplanması ve bu değer toplam alandan çıkarılarak toplam alana oranlanmasıyla elde edilir. (6.1)

$$\text{Kapsama\%} = (\text{Toplam Alan} - \text{Kapsanmayan Alan}) / \text{Toplam Alan} \quad (6.1)$$

Simülasyonlar engelsiz ortamlar için bütün alanın taranmasıyla, engelli ortamlar için ise en az %98 oranında tarama gerçekleştiğinde sonlandırılmıştır.

Deneyle farklı engel yüzdesine sahip 4 farklı küme için gerçekleştirilmiştir. Yöntemler ve bu yöntemlerin değerlendirilmesinde engelsiz ortam, %10 engelli 5 farklı ortam ve %20 engelli 5 farklı ortam kullanılmıştır. %10 ve %20 engelli ortamlar engellerin rasgele ve homojen dağılımıyla oluşturulmuş ortamlardır.

1. küme için 5 farklı çalıştırmanın ortalamasıyla elde edilen sonuçlar Çizelge 6.1’de verilmiştir.

Çizelge 6.1 : Engelsiz ortamda 5’er kere çalıştırılan LRV ve E-LRV’ye ait sonuçlar.

Yöntem	Süre(dk, μ)	Tarama Oranı(%)	Düğüm Sayısı(aralık)
E-LRV	11.33	100	24-26
LRV	32.38	100	25-26

Bu çizelgede toplam tarama zamanı, tarama oranı ve ortama bırakılan düğüm sayıları(aralık şeklinde) verilmiştir. Sonuçlara bakıldığında ortama bırakılan düğüm

sayısında önemli bir değişiklik olmamasına rağmen ortamı tarama süresi göz önünde bulundurulduğunda E-LRV yöntemi LRV yönteminden daha kısa sürede ortamı taramıştır.

2. küme için %10 engelli 5 farklı ortamın her biri için 5'er defa çalıştırılan yöntemlerde elde edilen ortalama değerler ve sonuçları Çizelge 6.2'de ve 3. küme için %20 engelli 5 farklı ortamın her biri için 5'er defa çalıştırılan yöntemlerde elde edilen ortalama değerler ve sonuçları Çizelge 6.3'te verilmiştir.

Çizelge 6.2 : %10 engelli 5 farklı ortam için 5'er kere çalıştırılan LRV ve E-LRV ye ait ortalama değerleri içeren sonuçlar.

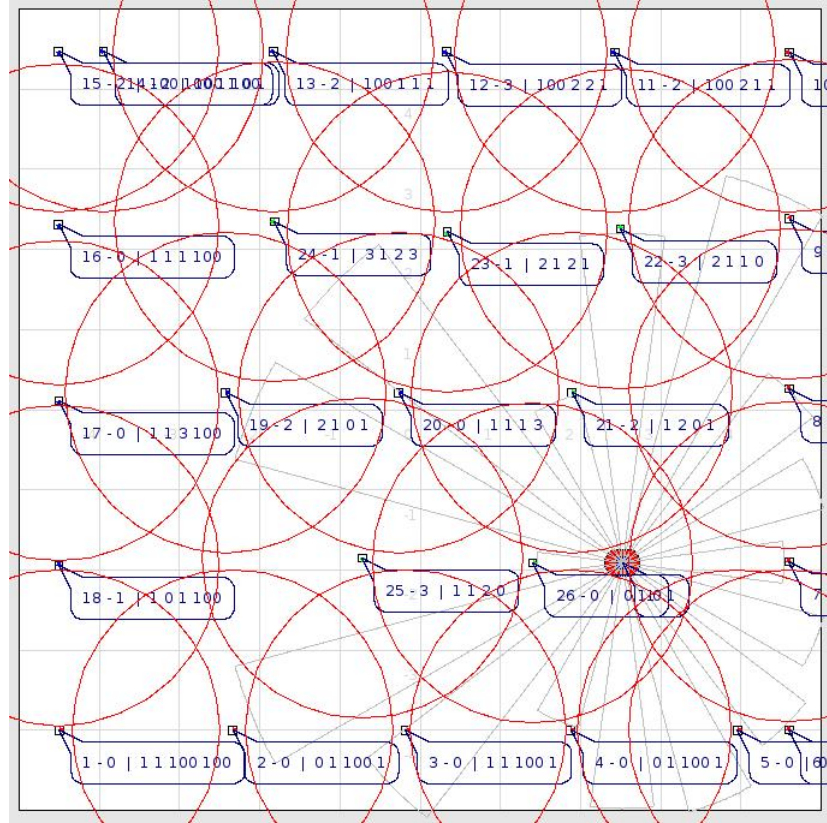
Ortam (%10 engelli)	Süre (dk, μ)		Tarama Oranı (%)		Düğüm Sayısı (aralık)	
	E-LRV	LRV	E-LRV	LRV	E-LRV	LRV
Ort1	27.96	89.58	99.71	98.23	23-24	25-26
Ort2	17.73	50.68	98.52	99.53	24-25	24-26
Ort3	12.63	34.33	99.97	98.64	24-25	25-26
Ort4	31.75	93.75	99.39	99.14	25	26
Ort5	22.21	67.93	99.76	99.25	24-26	24-25

Çizelge 6.3 : %10 engelli 5 farklı ortam için 5'er kere çalıştırılan LRV ve E-LRV ye ait ortalama değerleri içeren sonuçlar.

Ortam (%20 engelli)	Süre (dk, μ)		Tarama Oranı (%)		Düğüm Sayısı (aralık)	
	E-LRV	LRV	E-LRV	LRV	E-LRV	LRV
Ort6	11.88	46.05	99.36	98.31	23-25	26
Ort7	26.50	63.86	99.11	98.43	25	27
Ort8	21.35	56.80	98.62	98.26	24-25	26-27
Ort9	30.20	77.40	99.71	99.84	26	25-26
Ort10	28.61	74.20	99.34	99.13	25-26	25

Bütün sonuçlar değerlendirildiğinde ortama bırakılan düğüm sayısı iki algoritma için ciddi bir farklılık göstermemiştir. İki algoritma karşılaştırıldığında göze çarpan en büyük farklılık ortamı tarama süresinde görülmektedir. Engelli ya da engelsiz ortamlarda E-LRV yöntemi LRV yönteminin ortamı tarama süresinin yaklaşık %33'ü kadar bir sürede ortam tarama işlemini tamamlanmaktadır. E-LRV yöntemindeki düğüm bırakma stratejisine göre düğümü bırakırken robotun geçtiği yönlerin (düğümün önerdiği yön ve robotun o düğümü bırakmadan önceki yönü) düğüm tarafında güncellenmesi ve yine ortamdaki engelli yönlerin düğüm tarafında güncellenmesiyle E-LRV yönteminin ortamı tarama süresi LRV yöntemiyle karşılaştırıldığında belirgin bir farklılık göstermektedir. Bu yeni düğüm bırakma

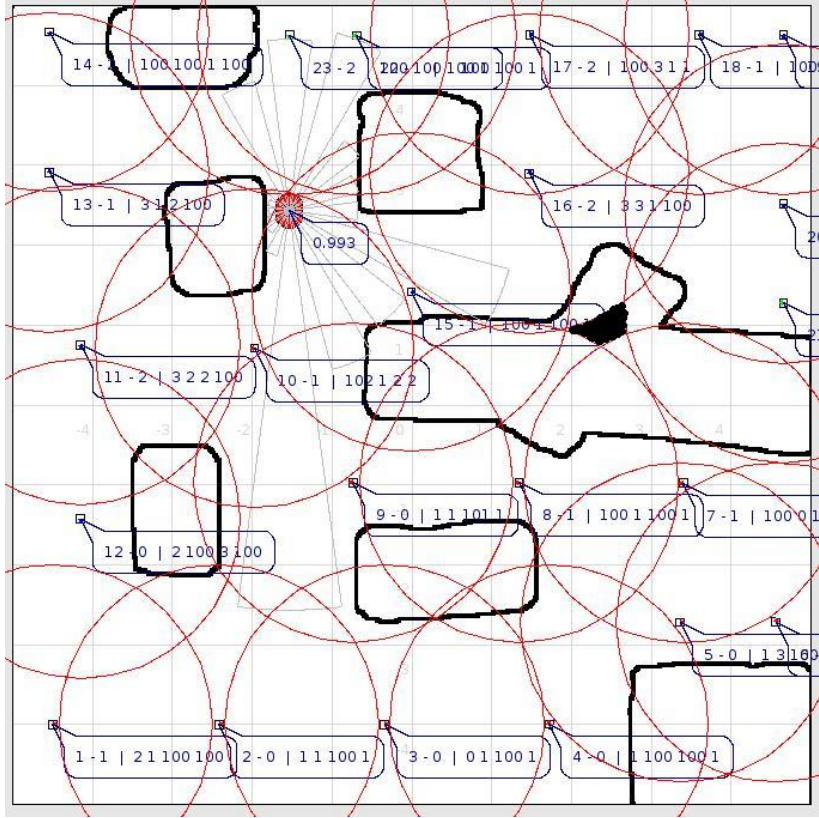
stratejisi sayesinde ortam bilgileri elde edildiği ilk anda kullanılmış olup bu bilgiler sayesinde güncellenen yön bilgileriyle robotun sonsuz döngü içine girmesi engellenmektedir. Bu yöntem sayesinde aynı zamanda toplam mesaj trafiği azaltılmış olup robota en doğru bilgi, en kısa zamanda sunulmaktadır. Sonuçlardan ayrıca ortamın engel oranının, ortam tarama süresi üzerinde etkisi olmadığı görülmektedir.



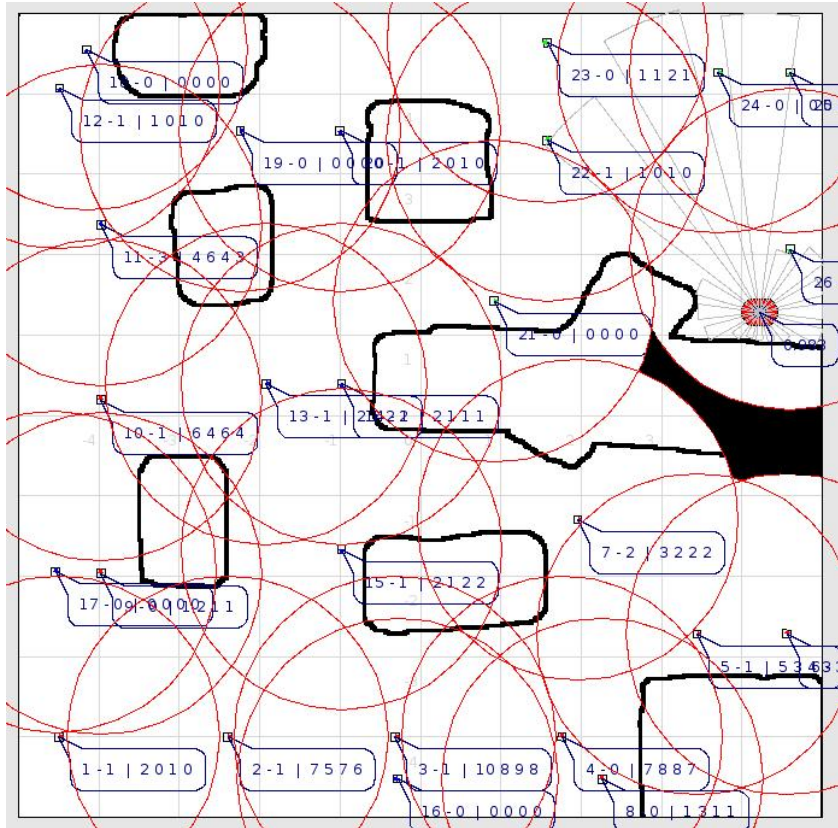
Şekil 6.1 : E-LRV yönteminin engelsiz ortamda çalıştırılması.

Şekil 6.1’de E-LRV yönteminin engelsiz ortamda çalıştırılmasıyla elde edilen simülasyon sonucu yer almaktadır. LRV yönteminde E-LRV yöntemindekine benzer bir düğüm yerleşimi gözlemlenmiş olup, Çizelge 6.1’de verilen sonuçlarda da görüldüğü gibi ortamı tarama süresi E-LRV yöntemine göre daha uzun sürmektedir.

Şekil 6.2 ve Şekil 6.3’te %20 engelli bir ortamda E-LRV ve LRV yöntemleri kullanılarak elde edilen simülasyon sonuçları verilmiştir.



Şekil 6.2 : E-LRV yönteminin %20 engelli ortamda çalıştırılması.



Şekil 6.3 : LRV yönteminin %20 engelli ortamda çalıştırılması.

Şekil 6.2 ve Şekil 6.3'te verilen simülasyon sonuçlarından da görüleceği üzere %100 bir şekilde ortamın taranması her zaman mümkün değildir. Ortamda bulunan büyük engellerin içinde taranmayan alan (taranmayan alanlar siyahla boyanarak gösterilmiştir) kalabilmektedir.

Bu deney kümelerinin dışında heterojen engel dağılımına sahip olan %15 engelli 3 farklı ortam için 4. bir küme oluşturularak deneyler yapılmış ve bu küme için elde edilen sonuçlar Çizelge 6.4'te verilmiştir.

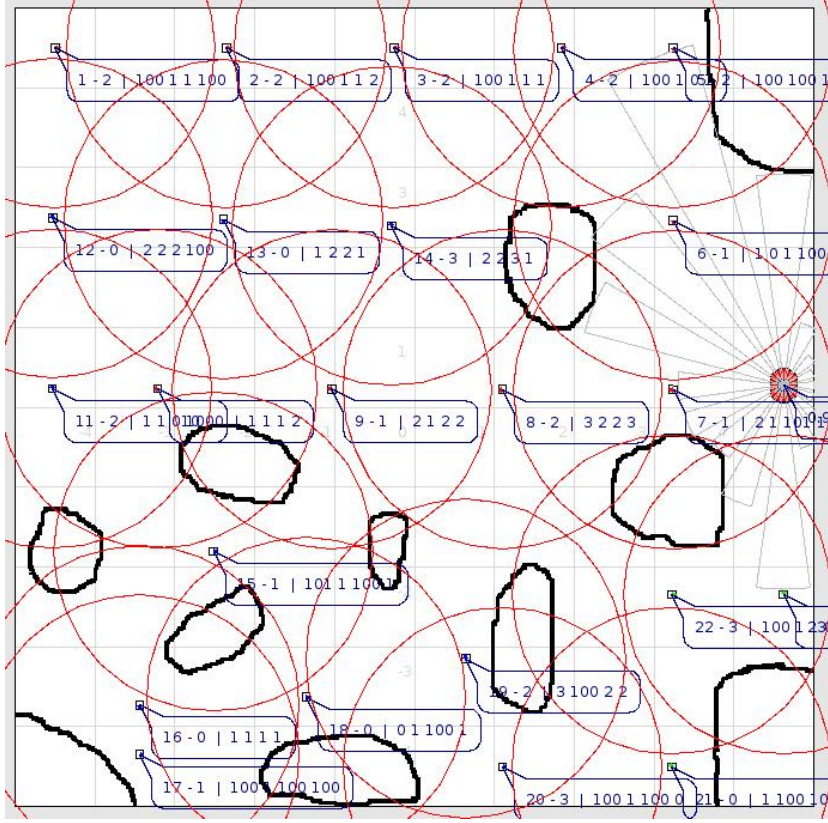
Çizelge 6.4 : Heterojen dağılıma sahip %15 engelli 3 farklı ortam için 5'er kere çalıştırılan LRV ve E-LRV ye ait ortalama değerleri içeren sonuçlar.

Ortam (%15 engelli)	Süre (dk, μ)		Tarama Oranı (%)		Düğüm Sayısı (aralık)	
	E-LRV	LRV	E-LRV	LRV	E-LRV	LRV
Ort11	18.39	60.45	98.07	89.62	22-25	18-22
Ort12	16.64	52.78	99.30	91.02	23-26	20-22
Ort13	23.47	76.35	98.11	92.56	21-24	19-22

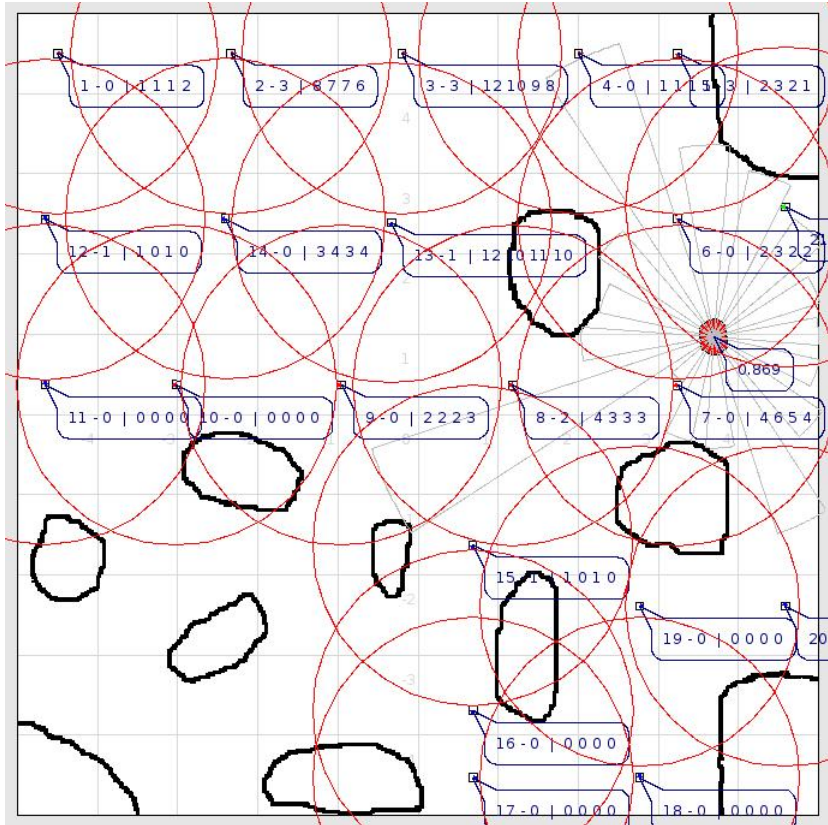
Dördüncü deney kümesi için kullanılan Ort11 engellerin belirli bir düzene göre saat yönünde arttığı bir ortamdır. Ort12 engellerin soldan sağa doğru keskin olmayan bir şekilde artış gösterdiği ortamdır. Ort13 ise engellerin soldan sağa doğru keskin bir şekilde artış gösterdiği ortamdır.

Bu deney kümesi için sonlanma kriteri diğer deney kümelerinde elde edilen tarama süresi oranıdır. Belirli bir ortam tarama yüzdesi sınırı bu yüzden belirlenmemiştir. Yöntemlerin deney sürelerinde yaklaşık %33 lük bir tarama süresi oranı elde edildiğinde deneyler sonlandırılmıştır. Bu deneyin sonuçları ve diğer deneylerde elde edilen sonuçlar değerlendirildiğinde heterojen dağılıma sahip ortamlarda E-LRV yönteminin LRV yöntemine göre daha kısa sürede ortamı taradığı görülmüştür. Bu deney kümesinde LRV yönteminin E-LRV yöntemindeki tarama yüzdesine ulaşması için Çizelge 6.4'te belirtilen süreden daha fazla süre çalışması gerekmektedir.

Şekil 6.4 ve Şekil 6.5'te 4. deney kümesi için elde edilen simülasyon sonuçları yer almaktadır.



Şekil 6.4 : E-LRV yönteminin heterojen dağılımlı %15 engelli ortamda çalıştırılması.



Şekil 6.5 : LRV yönteminin heterojen dağılımlı %15 engelli ortamda çalıştırılması.

7. SONUÇLAR VE ÖNERİLER

Ortam tarama, gezgin robotlar yardımıyla gerçekleştirilen pek çok gerçek zamanlı uygulamada karşılaşılan ve etkin olarak çözülmesi gereken bir problemdir. Ortam tarama stratejisinin etkin olarak yürütülmesi, arama kurtarma, mayın tarama gibi kritik pek çok uygulama alanında görevin tamamlanmasını, günlük hayatta da çim biçme, boyama, temizleme (süpürme) gibi pratik uygulamalarda görev başarımını etkilemektedir.

Bu tez çalışması kapsamında gezgin robotların ortama bıraktıkları duyargalar yardımıyla ortam tarama işlemini bilinmeyen bir ortamda (çevrimiçi) gerçekleştirmesi için varolan bir yöntem olan Least Recently Visited (LRV) yönteminde görülen eksiklikler iyileştirilerek E-LRV yöntemi geliştirilmiştir. Ortam bilgilerinin LRV yöntemine göre daha etkin bir şekilde kullanılması esasına dayanan bir düğüm bırakma stratejisi geliştirilmiştir. Bu stratejide robotun düğümü bırakmadan önce gittiği yön ile düğümün önerdiği yön bilgileri düğümün bırakılmasının ardından güncellenmektedir. Bu sayede ortam taranırken elde edilen bilgiler en kısa sürede değerlendirmeye alınmış olup bu yönlerden geçildiği için düğüm tarafında güncelleme sağlanarak düğümün doğru bir yön önermesi sağlanmaktadır. Düğüm bu yönlerin bilgilerini güncelleyerek robota en az ziyaret edilen yön bilgisini doğru bir şekilde sunmaktadır. Ayrıca düğümün bırakıldığı sırada bu geçilen yönlerin dışında herhangi bir şekilde engelli bir yön varsa bu yön de düğümüne gönderilen bir mesaj aracılığıyla güncellenerek robotun herhangi bir zamanda tekrar bu düğümün menziline girmesi durumunda (düğümün menziline girdiği her sefer için) bu yönün güncellenmesine gerek kalmamaktadır. Engel bulunan yönle ilgili güncelleme mesajını alan düğüm bu yönün değerini büyük bir değerle güncelleyerek bu yönün tekrar önerilmesini engellemiş olur.

Deneysel Player/Stage benzetim ortamında sonar duyargalarına sahip Pioneer 2DX robot modeli ile gerçekleştirilmiş olup E-LRV’de önerilen yeni düğüm bırakma stratejisi sayesinde E-LRV’nin aynı ortamı (homojen-engelli ya da engelsiz) LRV’nin ortamı

tarama süresinin %33'ü kadar bir sürede taradığı gözlenmiştir. Toplam süre göz önüne alındığında, mesaj trafiğinin düğüm bırakılırken 1 kereye mahsus büyük bir değerle güncellenmesinin robotun bu düğümün menziline her girişinde güncellenmesine kıyasla azaldığı ve ortamdaki bilgilerin elde edilmelerinin hemen ardından (robotun geçtiği yön ve önerilen yön bilgilerinin güncellenmesinin) kullanılmalarıyla robotun geçilen yönlere yönlendirilmesinin engellendiği görülmüştür. Ortama bırakılan düğüm sayılarında her iki yöntem kıyaslandığında ciddi bir farklılık görülmemektedir.

Simülasyonların daha farklı ortamlarda (farklı engel özelliklerine sahip) denenmesi sayesinde bu iki yöntemle ilgili daha fazla analizin yapılabileceği düşünülmektedir.

Geniş bir araştırma konusu olan duyarga ağlarının gerçekleşmesi ve bu konudaki iyileştirmeler için duyarga ağlarıyla ilgili kapsamlı çalışmalar yapılması sonraki akademik çalışmalarda düşünülmektedir.

KAYNAKLAR

- Agmon, N., Hazon, N., and Kaminka, G. A.,** 2006. Constructing spanning trees for efficient multi-robot coverage, *Proceedings of the 2006, IEEE International Conference on Robotics and Automation*, Vol. **15**, Issue. 19 May 2006, pp. 1698 – 1703.
- Batalin, M. A., and Sukhatme, G. S.,** 2002. Spreading out: A local approach to multi-robot coverage, *Proceedings of 6th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pp. 373 – 382.
- Batalin, M. A., and Sukhatme, G. S.,** 2002. Multi-robot dynamic coverage of a planar bounded environment, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Submitted), 2002.
- Batalin, M. A., and Sukhatme, G. S.,** 2003. Efficient exploration without localization, *International Conference on Robotics and Automation (ICRA2003)*, Taipei, Taiwan, May 12-17, pp. 2714-2719.
- Batalin, M. A., and Sukhatme, G. S.,** 2003. Coverage, exploration and deployment by a mobile robot and communication network, *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, Palo Alto.
- Batalin, M. A., and Sukhatme, G. S.,** 2007. The Design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment. *IEEE Transactions on Robotics*, Vol. **23**, No.4, pp. 661-675.
- Choset, H.,** 2000. Coverage of known spaces: The Boustrophedon cellular decomposition, *In the Journal of. Autonomous Robots*, Vol. **9**, No. 3, pp. 247-253.
- Choset, H.,** 2001. Coverage for robotics a survey of recent results, *Annals of Mathematics and Artificial Intelligence*, Vol. **31**, pp. 113-126.
- Gabriely, Y., and Rimon E.,** 2001. Spanning-tree based coverage of continuous areas by a mobile robot, *Annals of Mathematics and Artificial Intelligence*, Vol. **31**, pp. 77-98.
- Gage, D.W.,** 1992. Command control for many-robot systems, *19th Annual AUVS Technical Symposium*, Huntsville, Alabama, USA, pp. 22-24.

- Gerkey, B. P., Vaughan, R., Stoy, K., Howard, A., Sukhatme, G. S., and Mataric, M.,** 2001. Most valuable player: A robot device server for distributed control, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1226-1231.
- Gerkey, B., Vaughan, R., and Howard, A.,** 2003. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems, *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, pp. 317-323.
- Url-1** <<http://playerstage.sourceforge.net>>, alındığı tarih 19.12.2008.
- Hazon, N., and Kaminka, G. A.,** 2005. Redundancy, efficiency, and robustness in multi-robot coverage, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 735–741.
- Hazon, N., Mieli, F. and Kaminka, G. A.,** 2006. Towards robust on-line multi-robot coverage., *ICRA 2006, Proceedings 2006 IEEE International Conference on Robotics and Automation*, Volume, Issue, **15-19**, pp. 1710 – 1715.
- Huang, W. H.,** 2001. Optimal line-sweep-based decompositions for coverage algorithms, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21-26, pp. 27-32.
- Latimer IV, D., Srinivasa, S., Lee-Shue, V., Sone, S., Choset, H., and Hurst, A.,** 2002. Toward sensor based coverage with robotic teams, *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 961-967.
- Rekleitis, I., Lee-Shue, V., New, A. P., and Choset, H.,** 2004. Limited communication multi-robot team based coverage, *IEEE International Conference on Robotics and Automation*, pp. 3462-3468.
- Yao, Z.,** 2006. Finding efficient robot path for the complete coverage of a known space, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 9-15, pp.3369-3374.
- Zheng, X., Jain, S., Koenig, S., and Kepme, D.,** 2005. Multi-robot forest coverage. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2318–2323.
- Zheng, X., Koenig, S.,** 2007. Robot coverage of terrain with non-uniform traversability, *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 29-Nov. 2, 2007.

ÖZGEÇMİŞ

Ad Soyad: Meral CAMCI

Doğum Yeri ve Tarihi: Eskişehir / 20.03.1981

Adres: Mecidiye Mahallesi Bestekar Ahmet Çağın Sk.
Şirin Apt. No:12/7 Beşiktaş/İSTANBUL

Lisans Üniversitesi: Bilgisayar Mühendisliği Bölümü
İstanbul Teknik Üniversitesi

Yayın Listesi:

Camcı, M., Sariel-Talay, S., 2009. Utilizing Sensor-Based Robotic Terrain Coverage, *International Workshop on Robotic Wireless Sensor Networks (RWSN'09)* in The 5th IEEE International Conference on Distributed Computing in Sensor Systems(DCOSS'09), June 8-10.