

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**EŞ ZAMANLI KONUM BELİRLEME VE
HARİTALAMAYA YÖNELİK AKILLI
ALGORİTMALARIN GELİŞTİRİLMESİ**

Bilgisayar Mühendisi Zeyneb KURT

**FBE Bilgisayar Mühendisliği Anabilim Dalında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı:

Yrd. Doç. Dr. Sırma YAVUZ

İSTANBUL, 2007

İÇİNDEKİLER

	Sayfa
KISALTMA LİSTESİ	iv
ŞEKİL LİSTESİ	v
ÇİZELGE LİSTESİ	vi
ÖNSÖZ	vii
ÖZET	viii
ABSTRACT	ix
1. GİRİŞ	1
2. TEKERLEKLİ MOBİL ROBOTLAR	2
2.1. Gezgin Robot Kinematiği	7
2.2. Gezgin Robotlarda Çalışma Alanı	8
2.3. Gezgin Robotlarda Serbestlik Derecesi	8
2.4. Robotun Sahip Olduğu Mekanizmalar	9
2.4.1. Merkezi Kontrol Birimi	10
2.4.1.1. Microchip 18F452 mikrokontrolörü	11
2.4.1.2. Microchip 12F675 Encoder Data Alıcısı	12
2.4.2. Hareket Mekanizması	12
2.4.3. Yön Mekanizması	12
2.4.4. Elektronik Hız Kontrolü (ESC)	14
2.4.5. Robot ile Bilgisayar Arasındaki Haberleşmenin Sağlanması	15
2.4.5.1. Protokol ve Paket Yapısı	15
2.4.6. Çevre Objelerin Mesafelerinin Ölçülmesi	16
2.4.7. Alınan Yolun Ölçülmesi	20
2.4.8. Robotun Hareket Yönünün Algılanması	21
2.4.9. Çarpma Durumunun Algılanması ve Aracın Durdurulması	21
2.4.10. Enerji Kaynağı	22
2.4.11. Kullanılan Arayüz	22
3. EŞ ZAMANLI KONUM BELİRLEME ve HARİTA OLUŞTURMA ALGORİTMALARI	24
3.1. Eş Zamanlı Konum Belirleme ve Haritalama Algoritmalarında Karşılaşılan Problemler	25
3.1.1. Algılayıcı Ölçüm Gürültüsü	25
3.1.2. Harita Boyutunun Çok Geniş Olması	26
3.1.3. Veri İlişkilendirme Problemi	26
3.1.4. Çevrenin Zamanla Değişmesi	27
3.1.5. Otonom Robot Gezinimi	27
3.2. Eş Zamanlı Konum Belirleme ve Haritalama Probleminde Kullanılan Temel Yöntemler	28

3.2.1	Kalman Filtresi Yöntemi	28
3.2.2	Expectation Maximization – Beklenti Enbüyültme (BE) Yöntemi.....	30
3.2.3	İşgal Edilen Alanlar Haritaları	31
3.3	Kullanılan Yöntemlerin Kıyaslanması	32
4.	SIRALI MONTE CARLO YÖNTEMİ	33
4.1	Sıralı Monte Carlo Yönteminin Özellikleri	33
5.	SİSTEMDE KULLANILAN EŞ ZAMANLI KONUM BELİRLEME ve HARİTA OLUŞTURMA ALGORİTMASI.....	36
5.1	Robotun sahip olması gereken temel davranışlar	36
5.2	Gezinme Algoritması.....	37
5.3	Algılayıcı verisi modellemesi	37
5.4	Robot Yörünge Takibi	37
5.5	Eş zamanlı konum belirleme ve harita oluşturma amacıyla geliştirilen SMC algoritması	43
5.5.1	Sistem Modeli.....	44
5.5.2	Ölçüm Modeli.....	44
5.5.3	SMC algoritması.....	46
6.	ÇEVİRİMDİŞİ TOPLANAN VERİ ile HARİTALAMA ve DEĞERLENDİRME ..	49
7.	SONUÇLAR.....	52
	KAYNAKLAR.....	55
	ÖZGEÇMİŞ.....	57

KISALTMA LİSTESİ

ACK	Acknowledge
A/DC	Analog to Digital Converter
BEC	Battery Eliminator Circuit
BEL	Belief
CMU	Carnegie Mellon University
CRC	Cyclic Redundancy Check
DC	Direct Current (doğrusal akım)
DIP	Dual – In Package
EM	Expectation Maximization (Beklenti Enbüyültme – BE)
ESC	Electronic Speed Control (Elektronik Hız Kontrol)
GPS	Global Positioning System
ICR	Instantaneous Center of Rotation
I/O	Input / Output
IR	Infrared
MMC	Main Micro Controller
PDF	Probability Density Function
PIC	Peripheral Interface Controller
PWM	Pulse Width Modulation
RAM	Random access memory
SFD	Start Frame Delimeter
SLAM	Simultaneous Localization and Map Building
SMC	Sequential (Sıralı) Monte Carlo
RF	Radio Frequency (Radyo Frekansı)
USART	Universal Synchronous Asynchronous Receiver Transmitter

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1a Standart teker	3
Şekil 2.1b Nakliye tekeri	3
Şekil 2.1c İsveç Tekeri	3
Şekil 2.1d Yuvarlak veya küresel teker	3
Şekil 2.2 Robotun ön tasarımına ait çizimler	6
Şekil 2.3 Robotun son görünümü	7
Şekil 2.4 Bilgisayar ve robot arasındaki iletişim	10
Şekil 2.5 Futaba S3003 Servo	13
Şekil 2.6 “J” tipi bağlayıcı	13
Şekil 2.7 Paket Yapısı	15
Şekil 2.8 Kızılötesi Mesafe Ölçümü	16
Şekil 2.9 Sharp GP2D12 Kızılötesi Algılayıcı	17
Şekil 2.10 Kızılötesi algılayıcı verisi dağılımı	17
Şekil 2.11 Kızılötesi algılayıcı verisine uydurulan model	20
Şekil 2.12 Normal durumda ekran görüntüsü	22
Şekil 2.13 Çarpma Durumunda Ekran Görüntüsü	23
Şekil 3.1 Dönme hatasının sonuca etkisi	26
Şekil 3.2 Dairesel bir ortamda gezinirken oluşan veri ilişkilendirme hatası	27
Şekil 3.3 Beklenti Enbüyültme ile elde edilen harita	31
Şekil 3.4a Ham sonar algılayıcı verisi kullanan occupancy grid algoritması sonucu	31
Şekil 3.4b BE’den sonra occupancy grid algoritması uygulanmasının sonucu	31
Şekil 5.1 Duvar bulma adımından duvar izleme adımına geçiş	36
Şekil 5.2 Robotun dönme hareketi esnasında oluşan ani dönme merkezi	39
Şekil 5.3 Ön teker orta noktasının çizdiği yay ve yayın ait olduğu çember	39
Şekil 5.4 Oluşan ikizkenar üçgen	40
Şekil 5.5a Sürekli sola dönerken izlenen yörünge	42
Şekil 5.5b Sürekli sağa dönerken izlenen yörünge	42
Şekil 5.5c Ters P çizerken izlenen yörünge	43
Şekil 5.6 Ölçüm değerleri tahmini	46
Şekil 5.7a Bir iç köşe bulma senaryosu	48
Şekil 5.7b Bir dış köşe bulma senaryosu	48
Şekil 6.1 Ters P yörüngesi ile ters L şeklinde bir engelin bulunması	49
Şekil 6.2 Kare şeklinde kapalı bir mekanın haritalanması	50
Şekil 6.3 Üç kenarlı ve düzleme eğik olan bir mekanın haritalanması	50
Şekil 7.1 Kare şeklindeki kapalı bir mekanın haritalanması	52
Şekil 7.2 Altı kenarlı kapalı bir mekanın haritalanması	53

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1 Tekerli taşıtlarda teker konfigürasyonu örnekleri.....	4-5-6
Çizelge 2.2 Robotun hareket kabiliyeti	9
Çizelge 2.3 Mikroişlemciler	11
Çizelge 2.4 Servonun özellikleri	13
Çizelge 2.5 Farklı darbe genişliklerine karşılık elde edilen servo açıları	14
Çizelge 2.6 Kontrol byte'ları.....	15
Çizelge 2.7 Artımlı encoder ucu tanımlamaları	21
Çizelge 3.1 Yöntemlerin kıyaslanması	32



ÖNSÖZ

Robotların insan hayatındaki yeri azımsanamayacak ölçüde artmaktadır. Robotlar insanlara hizmet etmek amacıyla çok çeşitli mekanlarda, çeşitli amaçlarla kullanılırlar. Örneğin deprem, yangın gibi afet durumlarında insanların erişemeyeceği yerlerde canlı tespitinde etkin bir şekilde kullanılması veya temizlik robotu gibi akıllı ev aletlerinde ve çeşitli endüstriyel alanlarda kullanılması mümkündür.

Ben de tez konumu seçerken insanların yaşam kalitesini artırmayı amaçlayan ve sosyal sorumluluk kaygısı olan bir çalışma yapmak istediğim için robotik alanına yönelerek sunulan çalışmayı tamamladım.

Aynı amaçlarla Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü'nde benim de dahil olduğum bir Mekatronik Grubu oluşturulmuş ve üzerinde çeşitli makine öğrenmesi algoritmalarının uygulanabileceği bir robot geliştirilmiştir.

Tezimi gerçekleştirme sürecinde desteklerinden dolayı öncelikle aileme; değerli hocam ve tez danışmanım Yrd. Doç. Dr. Sırma YAVUZ'a ve Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü Mekatronik Grubu üyelerinden başta M. Fatih AMASYALI ve Tarkan DİNÇ olmak üzere, Muhammet BALCILAR, Gökhan BİLGİN ve Hilmi Kemal YILDIZ'a teşekkür ederim. Ayrıca zor zamanlardaki arkadaşlıkları ve yardımlarından dolayı meslektaşlarım Can YÜKSEL, İbrahim ÖK ve Necmi ÖZTÜRK'e de teşekkür ediyorum.

Bu araştırma Yıldız Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'nce desteklenmiştir. Proje No: 27-04-01-01 .

ÖZET

Robotlar, algıladıkları verileri yorumlayıp nasıl tepki vereceklerine otonom olarak karar verebilen mekanizmalardır. Otonom robotların en önemli özelliği, insanlara nazaran sınırlı da olsa, kendi kendine yetebilme ve algılama özelliklerinin olmasıdır.

Robotların insanlarla ve içinde bulundukları ortamla etkileşime girmesi kaçınılmazdır ve çoğunlukla insanların girmesinin tehlikeli olduğu ortamlarda yapılması gereken işlemlerde kullanılmaktadır. Dolayısıyla robotların içinde bulundukları ortamı ve kendi konumlarını bilmesi önemlidir. Bunun için öncelikle ortamın haritasının çıkarılması gerekmektedir.

Bu amaçla bilinmeyen bir ortamda, bilinmeyen bir noktadan harekete başlayarak bir taraftan bu ortamın haritasını çıkaran, bir taraftan da kendi yerini tahmin edebilen ve başlangıç noktasına döndüğünü de algılayabilen bir robot Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü'nde geliştirilmiştir.

Sonuçlandırılan bu tezin amacı da bu robotun kapalı bir mekanda, bir bilgisayarla birlikte oluşturdukları otonom sistem vasıtasıyla eş zamanlı olarak kendi konumunu hesaplaması aynı zamanda bu alanın haritasını çıkarmasıdır. Robotlara bu becerileri kazandırmak için geliştirilen algoritmalar literatürde SLAM (Simultaneous Localization and Map Building – Eş Zamanlı Konum Belirleme ve Harita Oluşturma) algoritmaları olarak bilinmektedir. Ortamın haritasının çıkarımı ve robotun konumunun belirlenmesi esnasında kızılötesi algılayıcılar kullanılmakta ve robot gideceği yöne geliştirilen algoritma sayesinde karar verebilmektedir. Algılayıcı gürültüsü ve ölçüm belirsizliği gibi durumlar karşılaşılan en temel problemlerdir. Çözüm olarak çoğunlukla istatistiksel kestirim yöntemleri kullanılır. Sunulacak tezde de bu yöntemlerden biri olan parçacık filtresi uygulamalarından en yaygın olarak kullanılanı Sıralı Monte Carlo (SMC) algoritması kullanılmış ve Eş Zamanlı Konum Belirleme ve Harita Oluşturma problemine başarılı bir çözüm önerilmiştir.

Anahtar Kelimeler: Eş zamanlı konum belirleme ve haritalama algoritmaları, sıralı Monte Carlo yöntemi, parçacık filtreleri, otonom robot kinematiki.

ABSTRACT

Robots are autonomous mechanisms that can interpret the data they sensed from the environment and can decide how to react to the environment. The most common properties of an autonomous robot are: self sufficiency and sensing its surroundings. However, robots can have these abilities limited when compared to humans.

It is unavoidable for the robots to interact with humans and the environment surrounding them. Commonly robots are used for risky processes in dangerous environment. Hence, they should be able to recognize the environment surrounding them and their own position within this environment. To achieve these abilities robots need to build a map of the environment.

An autonomous robot, which is constructed in Yıldız Technical University, Computer Engineering Department, is used in this study. It begins its movement in an unknown position, builds a map of the environment, and at the same time estimates its new position. It is also able to recognize the starting point when it arrives that point again.

In this study, a computer and the robot construct an autonomous system. This autonomous system successfully builds the map while calculating the new perceived robot location simultaneously. In literature the algorithms which provide these abilities to the robots, are called SLAM (Simultaneous Localization and Map Building) algorithms. The aim of this thesis is to develop a successful SLAM algorithm by using a robot equipped with only simple and cheap sensors. During map building and simultaneous localization, the robot can sense its environment by its infrared sensors and can decide the path to follow by using the developed SLAM algorithm. The most frequent problems in SLAM algorithms are sensors' noise and odometry errors. To solve this problem in the SLAM, statistical estimate methods are used very often. In the thesis Sequential Monte Carlo (SMC) algorithm which is a well known particle filter application is used and promising results were obtained for the SLAM problem.

Keywords: Simultaneously localization and mapping algorithms, sequential Monte Carlo approach, particle filters, mobil robot kinematics.

1. GİRİŞ

Bu çalışmada kullanılan robot geliştirilen algoritma yardımı ile, çevrenin ön topolojik bilgilerine ya da birtakım referans nesnelerinin yer bilgisine sahip olmadan ve ihtiyaç duymadan eş zamanlı yer belirleme ve haritalama yapmaktadır. Dolayısıyla robot bilinmeyen bir ortamda bilinmeyen bir noktadan harekete başlayarak bir taraftan bu ortamın haritasını çıkarırken, bir taraftan da kendi yerini tahmin etmekte ve başlangıç noktasına döndüğünü de algılayabilmektedir. Tasarlanan robot bu seyri sırasında etrafını algılayıp nereye gideceğine yine kendi başına karar verecektir. Kullanılan robot sadece kızılötesi algılayıcılar kullanarak etrafındaki cisimlere olan uzaklığını ölçmektedir. Bu ölçümlerden yola çıkarak yoluna nasıl devam edeceğine karar vermekte ve konumunu bildirmektedir. Bu amaçla bir bilgisayar ve robot ile birlikte otonom bir sistem oluşturulmuştur.

Robotun çevresindeki engellere olan mesafesini ölçebilmek amacıyla kızılötesi algılayıcılardan alınan bilgi bilgisayara bir RF (Radyo Frekansı) modülü üzerinden aktarılmaktadır. Robot bir objeye çarptığında üzerindeki dokunmatik algılayıcılar etkilenerek objenin konumunu belirleyebilmektedir. Ön teker üzerinde bulunan artımlı encoder alınan mesafeyi ölçmekte, tekerin üst kısmındaki potansiyometre ise ön tekerin dönme miktarını göstermektedir. Robotun üzerinde yer alan merkezi kontrol birimi kızılötesi algılayıcılardan, dokunmatik algılayıcılardan, artımlı encoderdan ve potansiyometreden aldığı verileri bilgisayara yollamakta bilgisayardan gelen kontrol verilerini ise ön tekerdeki DC motora ve servo motora uygulamaktadır. Bilgisayar tarafında ise RF modülü vasıtasıyla alınan algılayıcı, artımlı encoder ve potansiyometre verileri, geliştirilen algoritmaya aktarılarak hız ve açı kontrol işaretleri algoritma tarafından üretilmekte, üretilen kontrol işaretleri yine RF modülü aracılığıyla robota gönderilmektedir.

Eş Zamanlı Konum Belirleme ve Haritalama algoritmaları geliştirilirken çoğunlukla istatistiksel kestirim yöntemlerinden yararlanılır. Literatürde gerçek zamanlı çalışmalar için etkin ve hızlı olmasından dolayı çoğunlukla istatistiksel kestirim yöntemlerinden olan “parçacık filtresi” yöntemi tercih edilmektedir. Parçacık filtresi uygulamalarından en yaygını Sequential (Sıralı) Monte Carlo (SMC) yöntemidir. Bu çalışmada da SMC yöntemiyle bir Eş Zamanlı Konum Belirleme ve Haritalama algoritması geliştirilmiş ve sürecin başarılı olduğu gözlenmiştir.

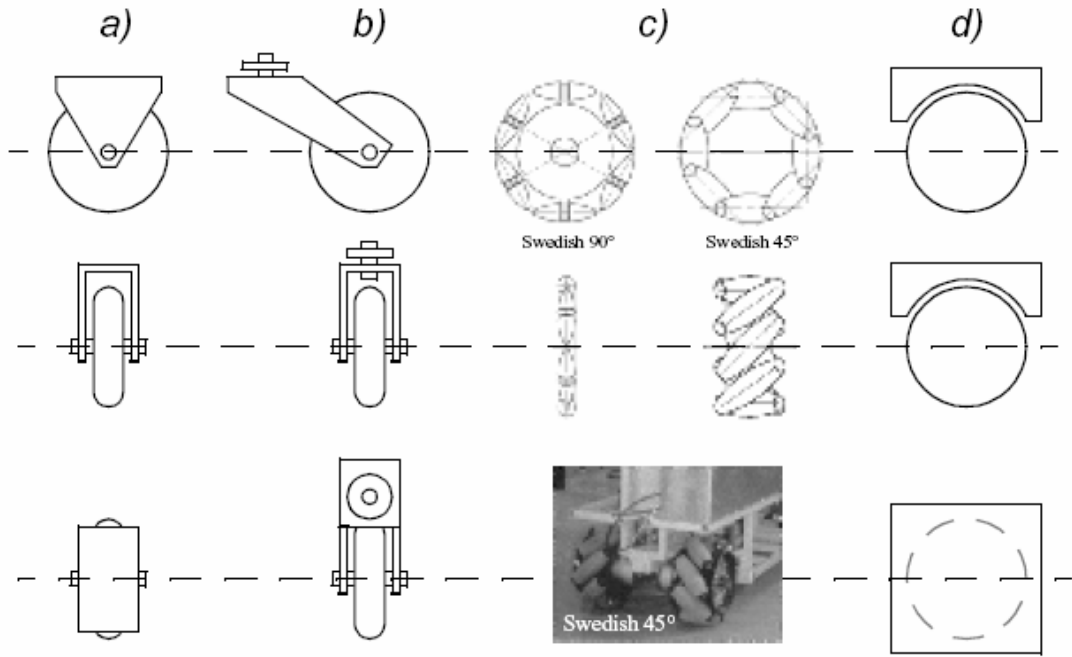
2. TEKERLEKLİ MOBİL ROBOTLAR

Bir gezgin robotun çevresini dilediği biçimde dolaşabilmesi için hareket mekanizmasına ihtiyacı vardır. Hareket kabiliyetinin yürümek, tırmanmak, sürünmek, yüzmek, uçmak ve yuvarlanmak gibi çok sayıda çeşidi mevcuttur. Dolayısıyla öncelikle robotun hareket kabiliyetlerinin belirlenmesi gerekmektedir.

Çoğu hareket mekanizması insan ya da hayvanlarda bulunan birtakım biyolojik özelliklerden esinlenerek oluşturulmuştur. Ancak düz bir yüzeyde etkin bir şekilde hareket etmeyi sağlayan, enerji ile çalışan ve insanlığın önemli icatlarından biri olan tekerlekler bu durumun dışında kalır. Aslında tekerlekli mekanizmalar da tamamen biyolojik sistemden bağımsız değildir. İki ayak üzerinde yürüme sisteminde adım uzunluğu azaldıkça yürüme hareketi de tekerleğin yuvarlanma hareketine yaklaşacaktır. Ancak tabii ki doğada tekerleğin yuvarlanma hareketinin birebir karşılığı yoktur, tekerleğin hareket mekanizmasını gerçeklemek için teknoloji gereklidir.

Gezgin robotlarda ve insan yapımı olan diğer taşıtlarda tekerlek en sık kullanılan hareket mekanizmasıdır. Mekanik olarak tekerleğin uygulanması diğer mekanizmalara göre daha kolaydır. Ayrıca tekerlekli robot tasarımında genelde denge kurmak bir sorun teşkil etmez. Çünkü tekerlekler her an yerle temas halindedir. Ayrıca Siegwart ve Nourbakhsh'e (2004) göre dengenin korunabilmesi için ideal olarak üç tekerlek yeterlidir.

Gezgin robotlar için hareket mekanizması oluştururken çok sayıda olası teker kombinasyonu üretilebilir. Şekil 2.1'de görüldüğü üzere gezgin robotlarda dört temel tekerlek sınıfı mevcuttur.



Şekil 2.1 Dört temel teker tipi: (a) Standart teker: serbestlik derecesi ikidir. (b) Nakliye tekeri: serbestlik derecesi ikidir. (c) İsveç Teker: serbestlik derecesi ikidir. (d) Yuvarlak veya küresel teker: gerçekleştirilmesi teknik olarak güçtür. (Siegwart ve Nourbakhsh, 2004)

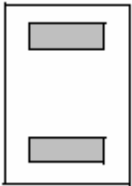
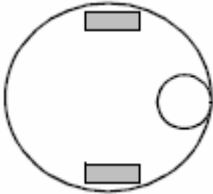
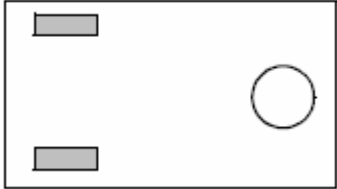
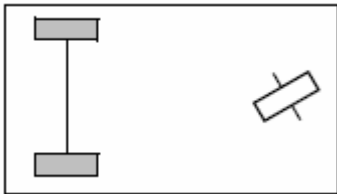
Standart tekerlek ve nakliye tekerleği ilkel bir dönme miline sahiptir. Teker farklı bir yöne dönebilmek için öncelikle dikey bir eksen boyunca doğrulma hareketi yapmalıdır. Bu iki tekerlek arasında temel bir fark vardır: doğrulma hareketi esnasında standart teker yerle etkileşim halinde bulunan dönme merkezinin kayması gibi yan etkilere maruz kalmaz, ancak taşıyıcı teker dönme merkezinden etkilenir ve doğrulma hareketi boyunca robotun gövdesine etki eden kuvvetin sahip olduğu eksen boyunca döner.

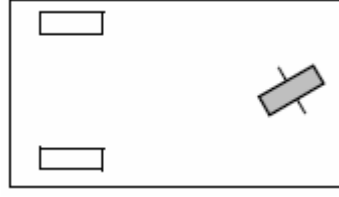
İsveç tekerleği, standart tekerlek gibi çalışır. Tekerleğin etrafına küçük silindirler eklenmiştir. Tekerleğe etkin olarak enerji veren tek parça birincil mildir. Küçük silindirler enerji için etkin olarak kullanılmazlar. Bu tasarımın en bilindik faydası, tekerleğin dönmesi için gereken enerjiyi temel bir mil sağladığından tekerlek, kinematik olarak çok küçük bir sürtünmeyle sadece ileri geri yönde değil, çok çeşitli yörüngelerde de hareket edebilir.

Küresel tekerlek ise çok yönlüdür, her yana dönebilir. Küresel tekere benzeyen bir mekanizma bilgisayarda kullanılan farelerde görülür. Fare içindeki küre dönerken kürenin üst kısmında döngüsel bir kuvvet açığa çıkar.

Hareket mekanizması tasarlanırken teker tipi ve teker geometrisinin ne olduğu önemli etkenlerdir. Çizelge 2.1’de teker sayılarına göre olabilecek konfigürasyonlardan bir kısmı verilmiştir. Çizelgede ayrıca belirli teker tiplerinin robot gövdesindeki geometrik konfigürasyonu görülmektedir. Gösterilen konfigürasyonlardan bazıları robot uygulamalarında pek sık kullanılmamaktadır. Örneğin iki tekerli bisiklet mekanizmasının manevra yeteneği orta düzeydedir, kontrol edilebilmesi zordur ve kendi başına ayakta duramaz.

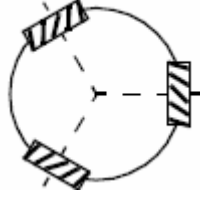
Çizelge 2.1 Tekerli taşıtlarda teker konfigürasyonu örnekleri (Siegwart ve Nourbakhsh, 2004)

Teker sayısı	Teker düzeni	Açıklama	Örnek
2		Yön belirleyen bir ön teker, gövdeyi ön tekerin çektiği yere sürükleyen bir arka teker var.	Bisiklet, motosiklet
		İki teker bir mille bağlıdır, ağırlık merkezi milin orta noktasıdır.	
		Önde bir teker, arkada farklı sürüş merkezleri olan iki teker bulunmaktadır.	
3		Önde çok yönlü bir teker, arkada iki bağımsız teker vardır.	Çoğu kapalı ortam robotları (Pygmalion ve Alice)
		Önde yön veren bir teker, arkada bir mille birbirine bağlı iki ayrı teker vardır.	



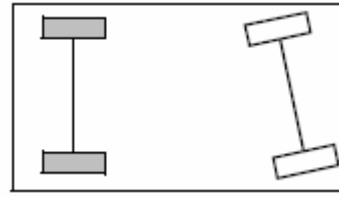
Önde yön veren bir teker, arkada iki teker vardır.

Neptune
(Carnegie Mellon Üniversitesi)



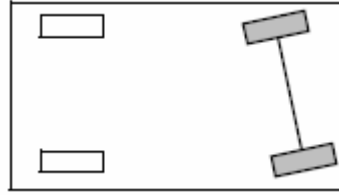
Bir üçgen düzenine oturtulmuş üç adet çok yönlü İsveç tekeri veya küresel teker.

Tribolo EPFL,
Palm Pilot Robot Kit
(Carnegie Mellon Üniversitesi)



Önde yön veren iki teker, arkada motorlu iki teker bulunmaktadır.

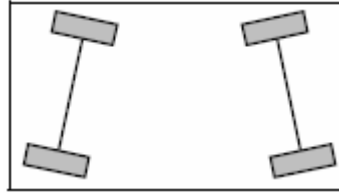
Akakadan sürüslü arabalar



Önde iki motorlu ve yön veren teker, arkada iki serbest teker vardır.

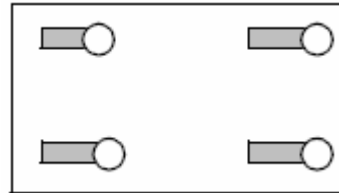
Önden sürüslü arabalar

4



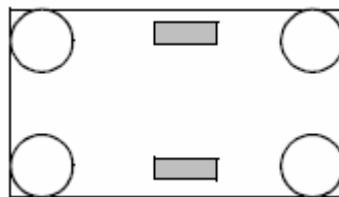
Dört adet yön belirleyen ve motorlu teker vardır.

Dört tekerli sürüşe sahip Hyperion
(Carnegie Mellon Üniversitesi)



Dört adet motorlu ve yön belirleyen taşıyıcı tekerler vardır.

Nomad XR4000



Merkezde iki adet çekici (sürükleyici) teker, her köşede birer tane çok yönlü teker vardır.

Terregator
(Carnegie Mellon Üniversitesi)

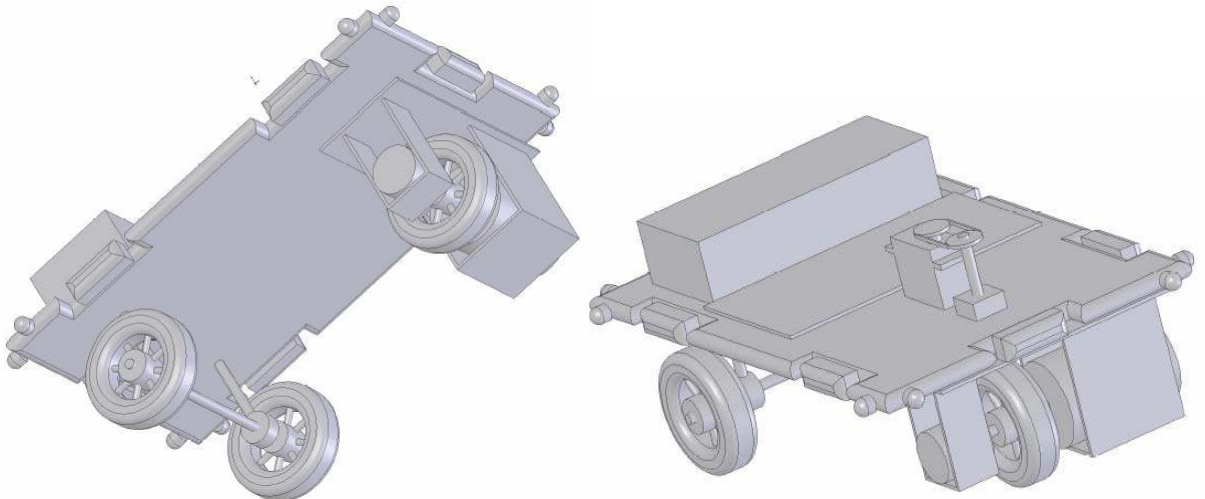
6

Yukarıdaki gösterimde kullanılan her bir teker

tipi ikonunun anlamı

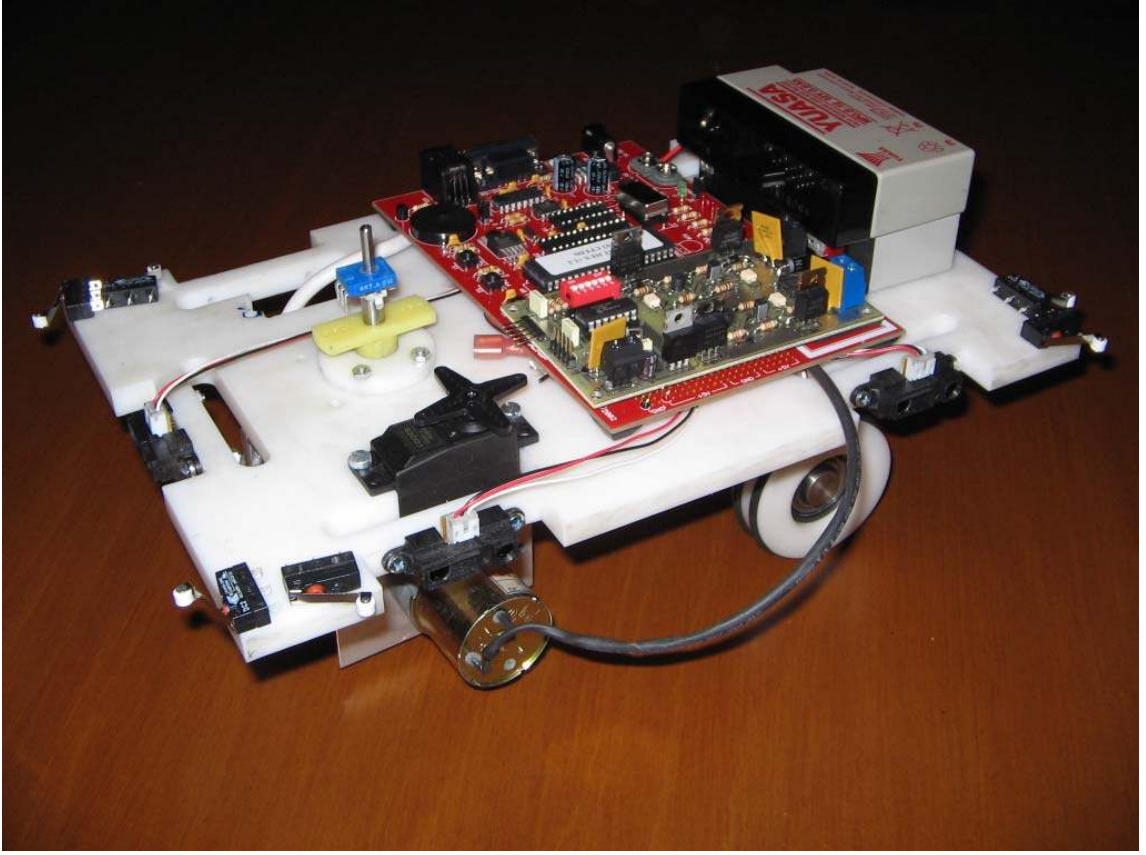
	Güç verilmemiş çok yönlü teker (küresel, nakliye tekeri, İsveç tekeri)
	Motorlu İsveç tekeri (Stanford tekeri)
	Güç verilmemiş standart teker
	Motorlu standart teker
	Motorlu ve yön belirleyen nakliye tekeri
	Yön belirleyen standart teker
	Bağlı tekerler

Uygulamamızda kullanılan robot bir ön ve iki arka olmak üzere, üç tekerlek üzerine yerleştirilen bir platformdan oluşmaktadır. Çizelge 2.1’de verilen üç tekerlekli robotlar başlığındaki üçüncü konfigürasyona sahiptir. Önde yön belirleyen ve çekme işini yapan bir teker, arkada ise birbirine bir mille bağlı iki teker mevcuttur. Ön tekerin verilen kontrol işareti ile sağa ve sola dönme özelliği vardır. Arkadaki iki teker sadece gövdeyi ön tekerin çektiği yöne götürmekten sorumludur. Robotun ön tasarımına ait bazı teknik çizimler Şekil 2.2’de görülmektedir.



Şekil 2.2 Robotun ön tasarımına ait çizimler

Robotun son görünümü Şekil 2.3'te görülmektedir.



Şekil 2.3 Robotun son görünümü

2.1 Gezgin Robot Kinematiği

Mekanik sistemlerin nasıl çalıştıklarını inceleyen en temel çalışma alanı kinematiktir. Gezgin robotların hareketleri esnasında yörüngelerini belirleyip planlayabilmeleri amacıyla kinematik denklemleri kullanılır. Belli bir amaca uygun gezgin robotlar tasarlayabilmek amacıyla robotun mekanik kabiliyetleri iyi belirlenmeli ve anlaşılır olmalıdır.

Bir robotun hareketlerini anlama süreci, her bir tekerin harekete ne kadar katkısı olduğunu tanımlamakla başlar. Her teker, robotun hareketinde etkin bir rol oynar ancak robot hareketine birtakım kısıtlar da getirir. Tekerlerin etki ettiği kuvvetler ve kısıtlar açık ve tutarlı bir biçimde ifade edilmiş olmalıdır. Gezgin robotların kendi konumlarını takibi ve dolayısıyla düzgün harita çıkarımı için tekerleklerin özelliklerini bilmek önemlidir. Kinematik denklem çıkarımında öncelikle tekerlerin tek tek kinematiği formülize edilir, daha sonra tüm robot gövdesinin kinematik denklemleri oluşturulur.

Robotun ön tekerinin gövdeyle yaptığı açısı verildiğinde, gövdenin konfigürasyonunu belirleyen problem düz kinematik problemi; robotun verilen bir gövde konfigürasyonu için ön teker açısını belirleyen problem ise ters kinematik problemi olarak ifade edilmektedir (Reuter, 1998).

Geliştirilen sistemde yörünge planlamasını gerçekleştirebilmek amacıyla robotun yapısına en uygun düz kinematik denklemleri seçilmiştir. Denklemlerin çıkarımları Bölüm 5.4'te anlatılmaktadır.

2.2 Gezin Robotlarda Çalışma Alanı

Bir robotun çalışma alanı, bu robotun serbestlik derecesinin ne olduğuna ve kendisini kontrol ederken serbestlik derecelerini nasıl kullanabildiğine bağlıdır. Serbestlik derecesi ise robotun manevra yapabilme kabiliyetidir. Çalışma alanı:

- Serbestlik derecesinden yola çıkarak robotun takip edebileceği yolların çeşidi
- Alternatif yolların oluşturduğu konfigürasyon uzayındaki olası yörüngeler

şeklinde ifade edilir.

Kısaca bir robotun çalışma alanı, gezinilen ortamın, robotun serbestlik derecesinin izin verdiği ölçüde görülebilen kısmıdır.

Çalışma alanının daha anlaşılır kılınması için serbestlik derecesi tanımlanmalıdır.

2.3 Gezin Robotlarda Serbestlik Derecesi

Bir robot gövdesinin kinematik hareket kabiliyeti, robotun ortamda gezinirken sergileyebildiği hareket yetenekleri ve bu yeteneklerin sınırlarıdır. Bir gezgin robot ani kinematik hareketlerin (ya da ani manevraların) yanı sıra yön kontrolünü yapan tekerini kullanarak kendisini daha uzaktaki hedeflere doğru yavaş yavaş yönlendirebilmelidir. Kinematik hareket kabiliyetine manevra kabiliyeti veya serbestlik derecesi de denir.

Serbestlik derecesinin bir başka tanımı da gezgin robotun yeni pozisyonunu belirlemek için gerekli olan parametre sayısıdır.

Robotlarda serbestlik derecesi uzuv sayısına, mafsallara (eklem) sayısına ve mafsal serbestlik derecesine bağlıdır, uzuv boyutuna bağlı değildir.

Sistemde kullanılan robotun gövdesinden başka uzvu yoktur ve serbestlik derecesi ikidir.

Sistemde kullanılan robotun yön kontrolü yapan ön tekerinin iki çeşit hareketi mevcuttur:

- İleri – geri yönde,
- Sağa – sola doğru

Yukarıda verilen ilk tanıma göre manevra kabiliyeti, serbestlik derecesini verdiğiinden ve iki türlü manevra yapılabildiğinden serbestlik derecesi ikidir.

Robotun ileri ve geri yönde gidebildiği en küçük hız değerleri cm/saniye cinsinden ve byte değerleri olarak Çizelge 2.2’de verilmiştir. Buna ek olarak ön tekerin sağa ve sola dönebildiği en küçük ve en büyük miktarlar derece ve byte cinsinden yine aynı tabloda gösterilmektedir.

Çizelge 2.2 Robotun hareket kabiliyeti

	İleri – geri		Sağa – sola	
	Hız (cm/sn)	Byte olarak karşılığı	Açı (derece)	Byte olarak karşılığı
En küçük	5.6 (ileri)	105	-55 (sola)	182
0 değeri	0	104	0	114
En büyük	5.6 (geri)	103	61 (sağa)	52

Çizelge 2.2’de verilen değerler Bölüm 5.4’te yörünge denklemlerinin çıkarımında kullanılacaktır. Açı değerlerinde sağ yön pozitif (+), sol yön (-) negatiftir.

Bölüm 5.4’teki yörünge denklemlerinde de görüleceği üzere robotun yeni konumunu belirleyebilmek için:

- Aldığı yol bilgisine,
- Ön tekerin gövdeyle yaptığı açı (kafa açısı) bilgisine

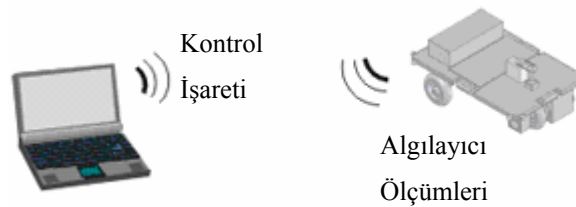
ihtiyacımız vardır. Yukarıdaki ikinci tanıma göre mekanizmanın yeni konumunu bulmak için gereken parametre sayısı da serbestlik derecesini vermekteydi, dolayısıyla bu tanıma göre de serbestlik derecesinin iki olduğunu söyleyebiliriz.

2.4 Robotun Sahip Olduğu Mekanizmalar

Beevers vd., (2005) ile Dissanayake vd., (2006)’nın da belirttiği üzere eş zamanlı konum belirleme ve haritalama probleminin lazer ve diğer gelişmiş algılayıcılar kullanılarak

çözölmüş bir problem olduđu söylenebilir. Literatürde kameralar, lazer algılayıcılar ve sonar algılayıcıların birlikte ya da tek başlarına kullanıldığı maliyetli çözümler yer almaktadır (Thrun vd., 2001). Ancak, ucuza mal edilebilen, kısıtlı bilgi toplayan kızılötesi algılayıcılar kullanılarak yapılan çalışmalar hala gündemdeki bir konudur. Dünyada bu konuda yapılan çalışmalarda kızılötesi algılayıcılar; robotbilim ve otomasyon, süreç kontrolü, uzaktan algılama ve güvenlik sistemlerinde yaygın olarak kullanılmaktadır. Özellikle de basit nesne tespitinde, sayma işlemlerinde, zemin algılamada, konum kontrolünde, derinlik ve erim gözetiminde, engel saptamada, bilgisayarla görme sistemlerinde kızılötesi algılayıcılar kullanılmışlardır (Castellanos vd., 2001; Thrun vd., 2001; Garulli vd., 2005).

Bu tezde kullanılan robot sadece kızılötesi algılayıcılar aracılığıyla etrafında herhangi bir engel olup olmadığını, varsa bu engelin kendisine olan mesafesini bulur, bunları RF modülü vasıtası ile bilgisayara aktarır. Bu bilgiler işlenerek bir yandan robotun bulunduğu mekanın haritası çıkarılırken diğer yandan da robotun kontrol işareti oluşturularak bilgisayar tarafından robota gönderilmektedir. Robotta bir ön, bir arka, ikişer tane sağda ve solda olmak üzere toplam altı adet kızılötesi algılayıcı bulunmaktadır. Ayrıca çarpmaları anlamak amacıyla her köşede ikişer tane olmak üzere toplam sekiz adet dokunmatik algılayıcı bulunmaktadır. Bilgisayar ve robot birlikte Şekil 2.4'teki gibi otonom bir sistem oluşturmuştur.



Şekil 2.4 Bilgisayar ve robot arasındaki iletişim

Yapılan gezgin robot, bir ön ve iki arka olmak üzere, üç tekerlek üzerine yerleştirilen bir platformdan oluşmaktadır. Platform ve tekerlekler derlin malzemesi ile yapılmış, ayrıca kaymayı önlemek için tekerlekler lastik ile kaplanmıştır. Robotu oluşturan temel mekanizmalar merkezi kontrol kartı, hareket mekanizması, yön mekanizması ve elektronik hız kontrolü olarak sayılabilir. Bu mekanizmaların fonksiyonları ve gerçekleşmelerinde kullanılan malzemeler aşağıda tanımlanmıştır.

2.4.1 Merkezi Kontrol Birimi

Bir mikroişlemci, robotun beyni gibi düşünülebilir. Robotun istenen görevleri gerçekleştirebilmesi için tasarımcı tarafından amacına uygun olarak programlanır. Temel

olarak, robottaki belli mekanizmalara komutlar gönderir, harici cihazlardan gelen verileri alır ve aktiviteleri düzenler. Bu çalışma için çeşitli mikroişlemciler kullanılabilirdi ancak kullanım kolaylığı, bakım yapma kolaylığı ve çalışmanın gerektirdiği her şeyi içerdikleri için PIC 18F452 ve PIC 12F675 mikroişlemcileri tercih edilmiştir. Her birinin özelliği Çizelge 2.3'te verilmiştir.

Çizelge 2.3 Kullanılan mikroişlemcilerin özellikleri

Özellikler	Microchip 18F452	Microchip 12F675
En büyük Darbe Frekansı (MHz)	40	8 (Dahili)
RAM (Byte)	1536	64
Flash Bellek (Byte)	32768	1792
EEPROM (Byte)	256	128
PWM Çıktıları	2	-
Zamanlayıcılar	5(1-8Bit, 3-16Bit, 1-WDT)	1(1-8bit)
A/D	8/10-Bit 8 Kanal	10 bit 4 Kanal
Seri Arayüzler	USART, I2C, SPI	-
I/O Pinleri	33	6
Paket	40-Pin DIP	8-Pin DIP

PIC 18F452, artımlı encoder kullanımı için gereken sayıcı ve zamanlayıcı adedine sahip değildir. PIC 12F675, artımlı encoder'dan gelen bilgileri okuyarak ön tekerin gittiği yol bilgisini belirlemekte kullanılmıştır.

2.4.1.1. Microchip 18F452

PIC 18F452 ana mikroişlemcidir ve “Ana Mikro İşlemci” anlamına gelen “Main Microcontroller (MMC)” adı verilir. MMC, robotun tüm işlevlerini kontrol eder ve yönetir. Sistemde kullanılan 18F452, USART'ını kullanarak RF alıcısıyla seri iletişim kurar ve bilgisayardan gelen veri paketlerini alır. PIC 12F675'te olduğu gibi harici cihazlarla iletişim kurmak için iki veri pinini kullanır ve algılayıcılarla iletişim halindeyken arabirim olarak sekiz adet analog'tan dijitale dönüştürücüsünü kullanır. PIC 18F452 kolay bulunan ve yaygın olarak kullanılan 40-pinli DIP (Dual In-line Package) paketiyle gelir. Robotun görevlerini yerine getirebilmesi için gerekli olan güç, darbe, toprak gibi bağlantılar kurulmuştur.

2.4.1.2. Microchip 12F675 Artımlı Encoder Veri Alıcısı

PIC 12F675'nın ana mikroişlemcinin sahip olduğu özelliklere ihtiyacı yoktur, sadece ön tekerin hareket yönünü ve miktarını belirlemekten sorumludur. Artımlı encoder adreslerini dinler, yön değişimi fark ettiğinde ana mikroişlemciyi uyarır ve yeni yönle ilgili bilgi verir. Ana mikroişlemci sadece indisi sayar ve yön değiştiğinde sonuçları kaydeder, sayıcıyı sıfırlayıp saymaya devam eder.

2.4.2 Hareket Mekanizması

Hareket mekanizması ön tekerlek mili üzerinde yer alan 1 adet DC Motor ile sağlanmıştır.

DC motorlar robotları hareket ettirmek için kullanılır. Güçlü, kolay programlanabilir olmasından ve robotun hızlı hareket etmesini sağlamasından dolayı sıklıkla tercih edilir. Ancak bu avantajlarının yanı sıra gidilecek mesafenin net ve kesin olarak kontrol edilememesi gibi önemli bir dezavantajı bulunmaktadır. Başka bir deyişle motoru açıp sürüş yaptıktan sonra birdenbire motor kapatılamaz, yavaş yavaş kapatılabilir. Ancak geliştirilen projede robot seyri esnasında aniden durmak isteyebilir. Bu sorunu, tekerin dönme sayısını belirleyip istendiğinde dışarıya verebilen bir artımlı encoder ile çözebiliriz. Kontrolün olduğu tarafta yani bilgisayar tarafında tekerin kaç tur attığı bilgisi istendiğinde artımlı encoder bu bilgiyi verir. Tekerin çevresinin kaç cm olduğu bilgisi de zaten biliniyor olduğu için kontrolün olduğu tarafta ne kadar yol gidildiği bilinmektedir.

Bu çalışmada kullanılan DC motor 12V ile sürülebilen, 47:1 dişli oranına sahip ve yüksek torklu (9000 g cm) bir motordur. Yüksek dişli oranı sayesinde robot farklı hızlarda hareket edebilme yetisine sahiptir.

2.4.3 Yön Mekanizması

Gezgin robotun istenilen yönde, istenilen açıda dönmesini sağlamak üzere bir adet servo motor kullanılmıştır. Servo motorda dişli çarklar mevcuttur ve verileri yörünge tespitinde kullanılmak üzere servonun dış miline tutturulmuş bir adet potansiyometre mevcuttur.

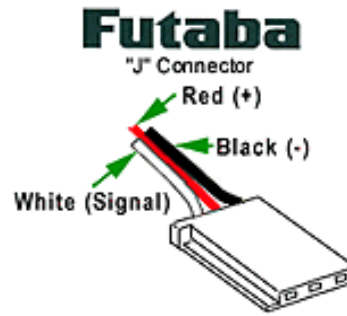
Servo düşük kuvvetle çalışan bir cihaz olduğu ve ön tekerin açisal pozisyonunu duyarlılıkla kontrol edebildiği için kullanılır. Sistemde kullanılan servo Şekil 2.5'te görülmektedir, özellikleri ise Çizelge 2.4'te verilmiştir.



Şekil 2.5 Futaba S3003 Servo

Çizelge 2.4 Servonun Özellikleri

Hız	4.8 voltta 0.23 saniye/60 derece
Torq	4.8 voltta 44.4 oz/in (3.2 kg/cm)
Boyutlar	1.59”L x 0.78”W x 1.42”H w/o dış şaftı
Ağırlık	1.31 oz (37.2g)
Bağlayıcı	“J” tipi



Şekil 2.6 “J” tipi bağlayıcı

Kullanılan servoyu yönetmek için üç uçlu “J” tipinde bir bağlayıcı kullanılır (Şekil 2.6). Uçlardan biri 4V – 6V arası güç kaynağı gerektirir, ikincisi topraklama için ve üçüncüsü de bir darbe genişlik sinyali taşıyan bir kontrol girdi sinyali içindir. Kontrol sinyali, servonun açısal pozisyonunu kontrol etmek için kullanılır. Kullanılan servonun en iyi işlem frekansı 50 Hz’dir. Servo, 20 ms’lik (50 Hz) bir periyotta 1ms’den 2 ms’ye kadar kontrol sinyalleri üreten bir mikroişlemci kullanılarak test edilmiştir. Uygulamamızda PIC18F452’den 50 Hz elde edebilmek için, PWM darbe genişliği modülasyonunun yazılım uygulaması kullanılmıştır. PWM’nin donanım uygulaması yazılım uygulamasından daha hızlıdır. Ancak servo motor,

PWM'nin donanım uygulamasıyla çalıştırılmaz. Çünkü PWM'nin donanım uygulamasından elde edilen en küçük frekans değeri servonun çalışması gereken frekans değerinden daha büyüktür. Ayrıca, en küçük frekans, devrede kullanılan kristal frekansına bağlıdır.

Farklı darbe genişlikleri ile yapılan denemelerin sonuçları Çizelge 2.5'te verilmiştir. 1.5 ms'lik darbe, servoyu merkeze getirir. 1.0 ms'lik ve 2.0 ms'lik darbeler sırasıyla saat yönünün tersine 45 derecelik ve saat yönüne 45 derecelik dönmeleri üretmek amacıyla kullanılır. Servo ve mikroişlemci aynı toprak ucunu kullanırlar.

Çizelge 2.5 Farklı darbe genişliklerine karşılık elde edilen servo açıları

Darbe Genişliği (ms)	Açı / derece
0.5	0
1.0	45
1.5	90 (merkez)
2.0	135
2.5	180

2.4.4 Elektronik Hız Kontrolü (ESC)

Motorun kontrolü bir elektronik hız kontrol ünitesi ile sağlanır. Çalışma prensibi servo motorunki ile aynıdır.

- 1.5 ms'lik bir pulse (darbe) dc motoru durdurur.
- 0.5 ms – 1.5 ms'lik darbeler ileri yönde,
- 1.5 ms – 2.5 ms'lik darbeler geri yönde hareketi sağlar.

Seçilen ESC (Electronical Speed Control) modeli ve özellikleri:

- Tekin Rebel 2,
- 4.8 -12v voltaj girişi,
- 6.0V / 5.0A BEC (Battery Eliminator Circuit) devresi,
- ileri/geri/fren kontrolleri

2.4.5. Robot Bilgisayar Arasındaki Haberleşmenin Sağlanması

Bu çalışmada robot ile bilgisayar arasındaki haberleşme bir adet alıcı ve bir adet verici RF Modülü ile sağlanmıştır. Hem bilgisayar tarafında hem de robot tarafında kullanılmak üzere Aurel'in 433.92 MHz'lik WIZ-SML-IA RF alıcısı seçilmiştir. Bu model, yarı çift yönlü (Half-duplex) modda noktadan-noktaya iletim yapan 100kbps'lik bir XTR alıcı-vericisi ile bütünleşik olan bir alıcı-vericidir. Bu cihazla veri seri iletimle alınır ve ana mikroişlemciye USART'a iletilir. Ana mikroişlemci, USART kesmeleri ile gelen veriyi alabilmek için RF modülünü dinler. Haberleşmede kullanılan protokol ve paket yapısı aşağıda açıklanmıştır.

2.4.5.1 Protokol ve Paket Yapısı

Bilgisayardan robota ya da robottan bilgisayara gönderilen verinin formatı Şekil 2.7'de görüldüğü gibidir.

Öncelikle eşzamanlama byte'ı bulunur. İkinci byte "start frame delimiter (SFD)" yani verinin başladığını gösteren bir ayıraçtır. Üçüncü byte paketin uzunluğunu verir. Dördüncü byte gönderen ve alıcının adreslerini içerir.



Şekil 2.7 Paket Yapısı

Beşinci byte kontrol verisi içerir. Çizelge 2.6'da kontrol byte'ının alabileceği değerler ve anlamları görülmektedir.

Çizelge 2.6 Kontrol byte'ları

Kontrol Byte	Anlamı
ACK (0x00)	Son paket başarıyla iletili
NACK (0x01)	Son paketi alırken zaman doldu, paket alınamadı
COLLISION (0x02)	Robot bir yere çarptı
ERROR (0x03)	Veri paketinde hata oluştu
HELLO (0x04)	Bağlantının kurulduğunu onaylamak için

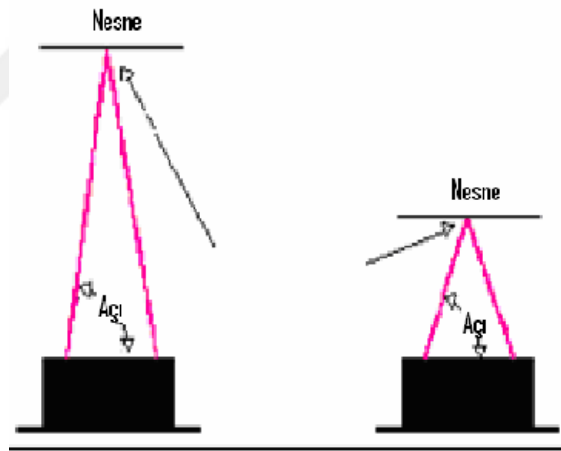
Paketin payload kısmı yollanmak istenen asıl verinin bulunduğu kısımdır.

Son kısım olan CRC (cyclic redundancy check), 16 bitlik CRC sağlaması yapan kısımdır. CRC, verinin doğru iletilip iletilmediğini kontrol eden bir çeşit hash fonksiyonudur.

2.4.6 Çevre Objelerin Mesafelerinin Ölçülmesi

Çevredeki nesnelere ya da engellere olan mesafeyi ölçmek için 6 adet IR (kızılötesi) alıcı ve verici algılayıcı kullanılmıştır.

Bu cihazlar gördükleri bir engele olan mesafeyi analog bir voltaj cinsinden gönderir. Mikroişlemciye analogdan dijitala dönüştürücü kullanılarak engele olan mesafe belirlenir. Kızılötesi algılayıcılar, insan gözüyle görülemeyecek özel kızılötesi ışık dalgaları yayarlar ve geri yansıyan ışıktan basit trigonometrik hesaplamalar yaparak engele olan mesafeyi belirlerler. Ancak ışık engellenirse veya yansıyan ışın algılayıcıya ulaşamazsa engeli yakalamak mümkün olmaz. Mesafe ölçümünün nasıl yapıldığı Şekil 2.8’de görülmektedir.



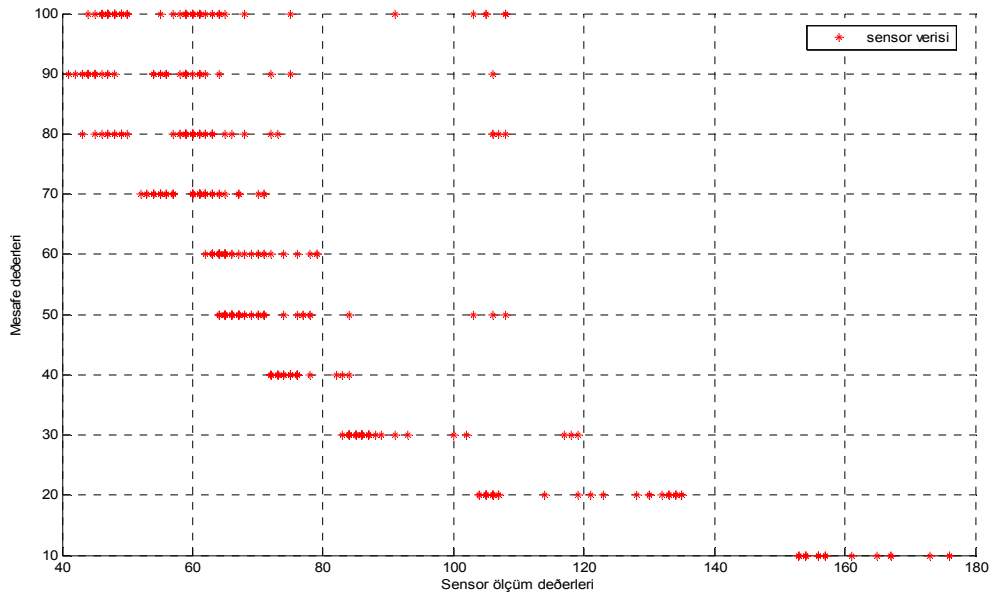
Şekil 2.8 Kızılötesi Mesafe Ölçümü

Projede kullanılması en uygun olan kızılötesi algılayıcı olarak Sharp GP2D12 seçilmiştir (Şekil 2.9). 10 cm’den 80 cm’e kadar ölçüm sınırı vardır, bu da uygulamamız için yeterlidir. Ayrıca ucuz olması da en önemli tercih sebeplerindendir. Algılayıcının en kısa mesafesi pek güvenilir değildir çünkü yansıyan ışının geri dönme açısı algılayıcının eni için çok geniştir ve algılayıcı büyük olasılıkla bu ışını yakalayamayacaktır.



Şekil 2.9 Sharp GP2D12 Kızılötesi Algılayıcı

Robotun bulunduğu konumu bilmesi ve harita çıkarabilmesi için algılayıcılardan gelen ölçüm değerlerine bakarak bir engelle karşılaşp karşılaşmadığını, karşılaşmışsa engelin ne kadarlık bir mesafede bulunduğunu anlaması gerekmektedir. Başka bir deyişle ölçüm değerleri modellenmeli ve anlaşılır kılınmalıdır. Bu amaçla robot, öncelikle gerçek zamanlı olmadan elle kontrol edilerek duvara olan mesafesi ve duruş açısı ile hangi ölçüm verisinin bu değerlere karşılık geldiği kaydedilmiş daha sonra da çeşitli eğri uydurma yöntemleri uygulanarak en uygun model seçilmiştir. Modelleme aşamasında robot elle kontrol edilerek veri toplanırken duvara olan mesafe 100 cm'den 10 cm'ye kadar 2.5 cm'lik aralıklarla ve her bir mesafe için arabanın duvara göre bakış açısı 10 dereceden 90 dereceye kadar 10'ar derece aralıklarla değiştirilmiş ve elde edilen ölçüm değerleri kaydedilmiştir. Kaydedilen ölçüm değerleri çizdirilince dağılımın Şekil 2.10'daki gibi olduğu görülmüştür.



Şekil 2.10 Kızılötesi algılayıcı verisi dağılımı

Grafikte x eksenini algılayıcılardan alınan voltaj bilgisinin 0-255 byte değerleri arasına normalize edilmiş halini göstermekte, y eksenini ise 10 cm'den 100 cm'ye kadar engele olan uzaklığı temsil etmektedir.

Bu dağılıma bakarak eğri uydurma yöntemleri ile verinin modellenebileceği düşünülmüş ve çeşitli eğri uydurma yöntemleri denenerek üstel modelin en az hatayı ve en uygun modellemeyi verdiği görülmüştür.

Üstel model, hatayı en küçültmeyi amaçlayan en küçük kareler yöntemine dayanır. Denklem:

$$y = f(x) = ae^{bx} \quad (2.1)$$

şeklindedir.

En küçük kareler yönteminde doğrusal modelin:

$$\hat{y}_i = a + b * x_i \quad (2.2)$$

şeklinde olduğunu biliyoruz.

x_i : bağımsız değişkendir, 0-255 arasına normalize edilmiş dijital algılayıcı verisidir;

$\hat{y}_i$: modelimizin bulacağı x_i 'ye bağımlı değişkenler olarak mesafe bilgisi

y_i : elle kontrollü olarak veri toplarken sahip olduğumuz gerçek mesafe bilgileri olsun.

y_i ve $\hat{y}_i$ arasındaki fark tahmin modelimizin hatasını verir:

$$e_i = y_i - \hat{y}_i \quad (2.3)$$

Toplam hata ise:

$$E = \sum_{i=1}^N e_i^2 \quad (2.4)$$

Modelimizdeki a ve b parametrelerini seçerken E'nin mümkün oldukça az olmasını amaçlarız. (en küçük kareler hata kriteri)

$$E = \sum_{i=1}^N e_i^2 = \sum_{i=1}^N (y_i - a - b * x_i)^2 \quad (2.5)$$

Bunun için E'nin a ve b'ye göre türevleri 0 olmalı:

$$\frac{\delta E}{\delta a} = \frac{\delta}{\delta a} \sum_{i=1}^N (y_i - a - b * x_i)^2 = 0 \quad (2.6)$$

Buradan a'ya göre türev alınca (2.7)'deki bağıntı elde edilir:

$$\sum_{i=1}^N y_i = a * N + b * \sum_{i=1}^N x_i \quad (2.7)$$

b'ye göre türev:

$$\frac{\delta E}{\delta b} = \frac{\delta}{\delta b} \sum_{i=1}^N (y_i - a - b * x_i)^2 = 0 \quad (2.8)$$

b'ye göre türev alınca ise (2.9)'daki bağıntı elde edilir:

$$\sum_{i=1}^N x_i * y_i = a * \sum_{i=1}^N x_i + b * \sum_{i=1}^N x_i^2 \quad (2.9)$$

Modelimiz üstel olduğundan denklem sistemleri çözümüne geçmeden önce aşağıdaki dönüşümler uygulanarak “ $y = f(x) = a * \exp(b * x)$ ” modelinden “ $y = a + b * x$ ” modeline geçeriz:

$$\ln(y) = \ln(a) + b * x \quad (2.10)$$

Denklem sistemine (2.10) ve (2.11) bağıntısındaki ön işlemleri yaptıktan sonra geçeriz.

$$\gamma = \ln(y) \quad \alpha = \ln(a) \quad (2.11)$$

Denklem sistemleri şu hale gelir:

$$\begin{pmatrix} \sum_{i=1}^N 1 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \end{pmatrix} * \begin{pmatrix} \alpha \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N \gamma_i \\ \sum_{i=1}^N x_i * \gamma_i \end{pmatrix} \quad (2.12)$$

α katsayısı bulunduktan sonra (2.11) bağıntısından a katsayısı çekilir.

Buna göre bulunan katsayılar (% 95 güven sınırlarıyla):

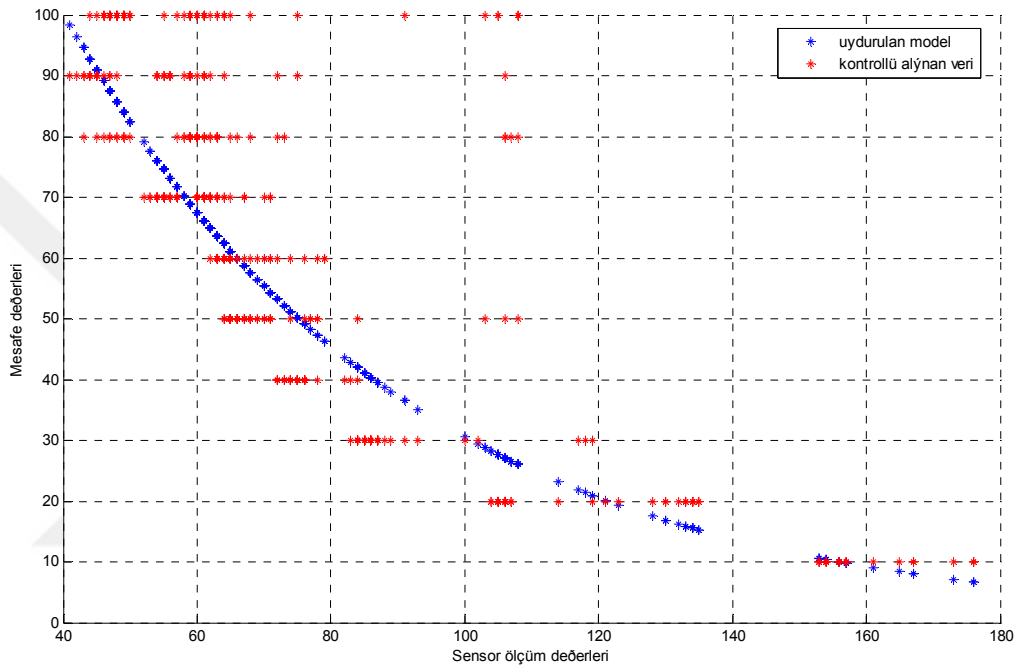
$a = 223.4$ ve $b = -0.01902$ şeklindedir.

Dolayısıyla modelimiz:

$$y = 223.4 * e^{-0.01902 * x} \quad (2.13)$$

şeklindedir.

Şekil 2.11’de uydurulan eğri mavi çizgi ile gösterilmiştir:



Şekil 2.11 Kızılötesi algılayıcı verisine uydurulan model

2.4.7 Alınan Yolun Ölçülmesi

Alınan yolun ölçülmesi ön tekerlek mili üzerine yerleştirilecek bir adet artımlı encoder ile sağlanmıştır. Artımlı encoder’lar devirsel harekete cevap olarak sinyal üretirler. Mekanik dönüşüm cihazlarıyla birlikte doğrusal hareketi ölçmek için kullanılırlar. Proje robotunun sahip olduğu artımlı encoder bir Hengstler RI 32’dir. Çözünürlüğü 1024 ve 6 uçludur. Çizelge 2.7’de gerekli tanımlamalar verilmiştir.

Çizelge 2.7 Artımlı encoder ucu tanımlamaları

Uç	Tanımlama
Kırmızı	DC 5 V/ 10 - 30 V
Beyaz	Kanal A
Yeşil	Kanal B
Sarı	Kanal N
Siyah	GND
Sarı-Siyah	Alarm

Sarı uç, bir devirde bir darbe üretir. Yeşil ve beyaz uçlar bir devirde 1024 darbe üretirler. Ancak yeşil ve beyaz uçlar arasında 90 derece faz farkı bulunmaktadır. Bu fark sayesinde artımlı encoder'ın yönünü belirleriz.

2.4.8. Robotun Hareket Yönünün Algılanması

Robotun hareket yönünün algılanması servo vasıtası ile ön teker miline bağlanan bir adet Potansiyometre ile sağlanmaktadır. Potansiyometre, robotun üzerindeki elektrik potansiyelini ölçmekte kullanılan cihazdır. Yön bilgisini elde etmek amacıyla ön tekere bağlanmıştır. Servo motorlar güvenilirdir ancak ön tekerin bir engele takılması gibi durumlarda yanlış bilgi verebilirler.

Projede seçilen potansiyometrenin model ve özellikleri:

- Top adj carbon modular pot
- 4K7 lineer 12mm

2.4.9. Çarpma Durumunun Algılanması ve Aracın Durdurulması

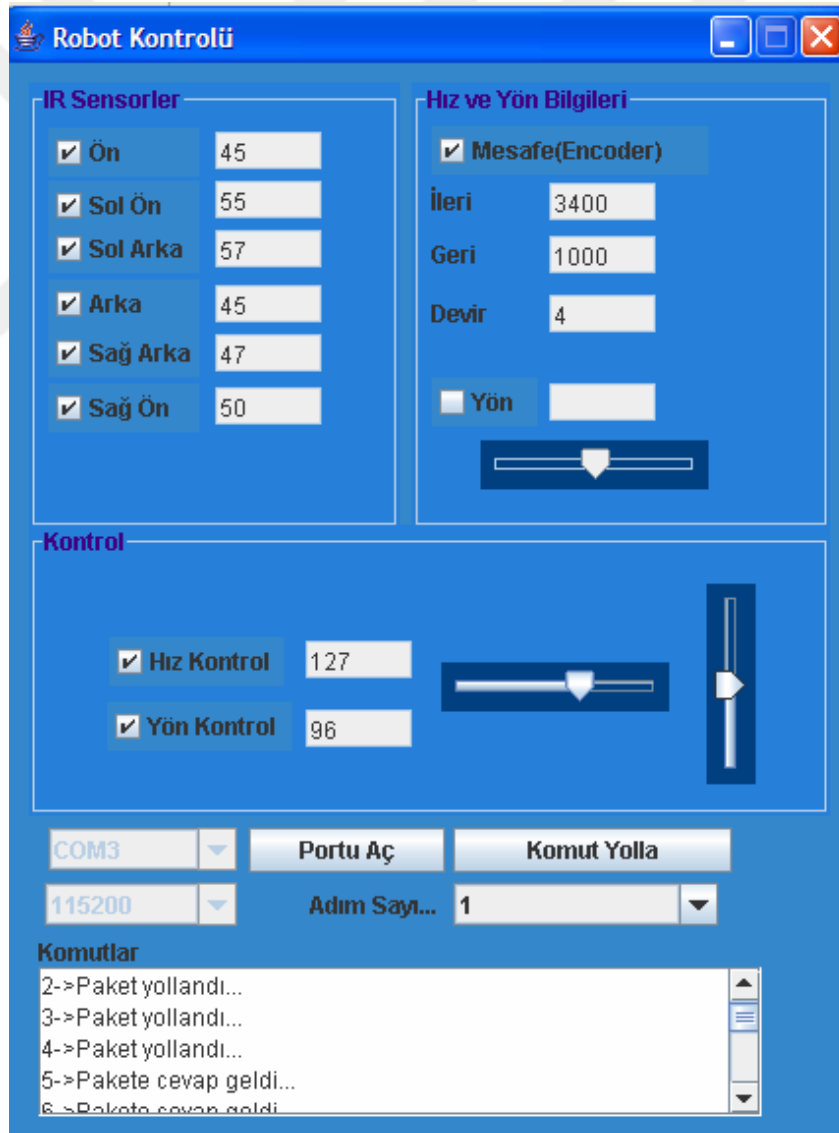
Çarpma durumunun algılanması ve aracın durdurulması, aracın köşelerine yerleştirilen sekiz adet dokunmatik algılayıcı switch (anahtar) ile sağlanmaktadır. Anahtarlar, bir elektrik akımını açan ya da kesen cihazlardır.

2.4.10. Enerji Kaynağı

Sisteme enerji sağlamak üzere gezgin platform üzerine yerleştirilen 7 Amper/saat gücünde iki akü kullanılmıştır. Motor ESC tarafından sürülmekte, ESC içindeki BEC devresinden ise servo ve kontrol kartı sürülmektedir.

2.4.11 Kullanılan Arayüz

Bölüm 2.4.6’da anlatıldığı gibi algılayıcı verisi modellenirken öncelikle robot elle kontrol edilerek gerçek zamanlı olmayan veri toplanmıştı. Bu aşama için kullanılan arayüz Şekil 2.12 ve Şekil 2.13’te görüldüğü gibidir. Sol üst tarafta herhangi bir anda alınan algılayıcı ölçüm değerleri byte cinsinden görülmekte, sağ üst tarafta potansiyometre ve artımlı encoder’dan alınan bilgiler gösterilmektedir.



Şekil 2.12 Normal durumda ekran görüntüsü

Orta kısımda gönderilen kontrol işaretleri görülmektedir. Komut göndermeden önce, RF modülün takılı olduğu port seçilir ve port açma butonuna basılır. Otomatik komut gönderme için komut yollama butonuna basılır.



Şekil 2.13 Çarpma Durumunda Ekran Görüntüsü

Çarpma durumunda Şekil 2.13'te görüldüğü gibi ekranda ilgili mesaj görülür.

3. EŞ ZAMANLI KONUM BELİRLEME ve HARİTA OLUŞTURMA ALGORİTMALARI

Harita çıkarımı, robotik ve yapay zeka alanlarında son yirmi yıldır hayli aktif olan bir araştırma konusudur. Haritalama, gezgin robotların fiziksel çevrelerini x-y düzlemindeki iz düşümlerine göre gösterme problemidir. Gerçekten otonom olan gezgin robotların en önemli problemi harita oluşturmaktır ve hala etkin bir şekilde bu probleme makul çözümler aranmaktadır. Bulunan çözümler çoğunlukla olasılıksal yöntemlerdir.

Eş zamanlı konum belirleme ve harita oluşturma problemi, ilk defa 1980'lerin sonunda, 1990'ların başında ortaya atılmıştır. İlk yıllarda temel olarak metrik ve topolojik yöntemler olarak ikiye ayrılmıştır. Metrik haritalar çevrenin geometrik özelliklerini gösterirken, topolojik haritalar farklı bölgelerin birbirine bağlı olup olmadığını gösterir. Chatila ve Laumond (1985) çevrenin geometrisini tanımlamak amacıyla bir dizi çokgen kullanarak metrik haritalama yapmışlardır. Mataric (1990) ile Kuipers ve Byun (1991) da topolojik haritalama yöntemlerini kullanan çözümler sunmuşlardır.

Metrik ve topolojik haritalama arasındaki fark yeterince belirgin değildir. Metrik haritalar topolojik olanlardan daha çok detay barındırır. Daha çok detay bulunduğundan "correspondence" (data association - veri ilişkilendirme) problemine iyi çözüm getirir ancak yüksek çözünürlük gerektirir ve hesapsal yükü ağırdır.

Eş zamanlı konum belirleme ve haritalama algoritmalarının tarihsel gelişim sürecinde dünya-merkezli ve robot-merkezli olmak üzere ikinci bir sınıflandırma daha yapılmıştır. Dünya merkezli haritalamada global bir koordinat düzleminde harita gösterimi yapılır. İlk bakışta robot-merkezli haritalama daha kolaymış gibi görünür ancak daha önce gezilmemiş yerleri tespit etmek ve birbirine benzeyen farklı lokasyonları ayırt etmek robot-merkezli haritalamada zordur. Bu sebeplerden genellikle dünya-merkezli haritalama daha sık kullanılır.

1990'lardan itibaren olasılıksal yöntemler haritalamada daha sık kullanılmaya başlanmıştır. Harita ve robot pozisyonu kestiriminde istatistiksel yöntemlerden en sık kullanılanı Kalman Filtresidir. Alternatif haritalama ve konum belirleme algoritmalarına Dempster'in (1997) geliştirdiği "expectation maximization (EM)" (beklenti enbüyültme - BE) algoritmaları örnek olarak verilebilir. Robot gezinmesi ve haritalama süreci için pahalı çözümler de geliştirilmiştir. Günümüzde üretilen çözümler çoğunlukla pahalıdır ve bilgi kazanımını maksimize etmeyi amaçlar. Eşzamanlı konum belirleme ve haritalama algoritması yöntemlerine geçmeden önce karşılaşılan problemler tanımlanmalıdır.

3.1 Eş Zamanlı Konum Belirleme ve Haritalama Algoritmalarında Karşılaşılan Problemler

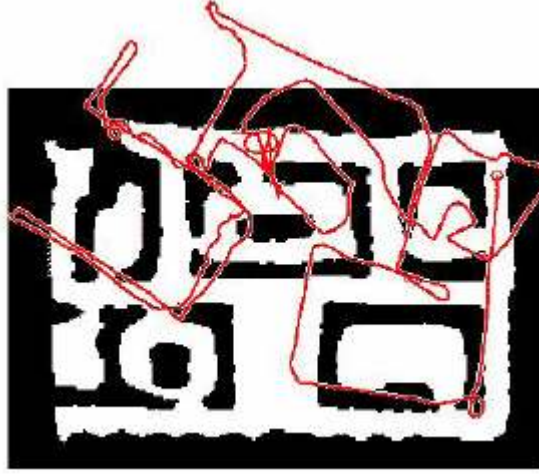
Literatürde harita çıkarma problemi, robot pozisyonu belirleme yani lokalizasyon problemi ile birlikte ele alınır. Çünkü etraftaki nesnelerin konumunu bilmek için başka bir deyişle harita çıkarabilmek için robot kendi konumunu ve buna bağlı olarak da algılayıcılarının konumunu bilmelidir; robotun kendi konumunu belirleyebilmesi için de gezindiği ortamda etrafında bulunan nesnelerin konumunu bilmelidir.

Ayrıca robot, haritasını çıkarmayı amaçladığı ortamda gezinirken dış dünyayı algılayabilmelidir. Bunun için de kamera; sonar, lazer veya kızılötesi teknolojilerini kullanan mesafe ölçerler, pusulalar veya GPS'ler kullanabilirler. Ancak algılayıcıların verisi tamamen güvenilir değildir, ölçümler gürültülü olabilir, ayrıca çoğu algılayıcının görme – algılama mesafesi kısıtlıdır. Örneğin ışık ve ses duvarlardan geçemez. Algılayıcıların bu kısıtlarından dolayı robot sabit kalarak bulunduğu ortamın haritasını çıkaramaz, bu ortamda gezinmek zorundadır. Haritası çıkarılacak ortamda gezinirken üretilen hareket (kontrol) komutları, farklı algılayıcı ölçümlerinin alındığı konumlarla ilgili bilgi içerdikleri için haritalamada önemlidir.

Robot aldığı komuta göre hareket ederken birtakım problemler çıkabilir, dolayısıyla robotun yeni pozisyonunu belirlerken sadece kontrol işaretlerine güvenmek doğru değildir. Eş zamanlı konum belirleme ve harita çıkarımı esnasında en sık karşılaşılan problemler aşağıda verilmiştir.

3.1.1 Algılayıcı Ölçüm Gürültüsü

Haritalamadaki temel problemler ölçümlerin gürültülü olmasından kaynaklanır. Eğer farklı ölçümlerde gözlenen gürültüler istatistiksel olarak bağımsız olsaydı haritalama problemi nispeten daha kolay bir hal alırdı. Ancak ölçüm hataları olasılıksal olarak bağımlıdır ve zamanla büyür. Şekil 3.1'de görüldüğü gibi başlangıçta küçük bir dönme hatası, robotun gezinimi boyunca artmış ve neticede mekanın oldukça hatalı haritalanmasına sebep olmuştur.



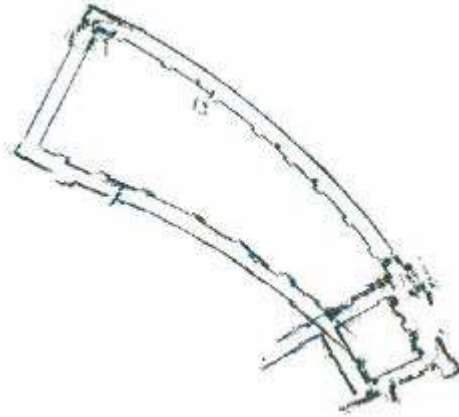
Şekil 3.1 Dönme hatasının sonuca etkisi (Thrun, 2002)

3.1.2 Harita Boyutunun Çok Geniş Olması

Haritalama ve konum belirlemede en sık karşılaşılan bir başka problem de haritası çıkarılacak mekandaki nesnelerin sayısının fazla olması dolayısıyla harita boyutunun artmasıdır. Robot, mekanın detaylı iki boyutlu planını çıkarırken genelde yüzlerce hatta binlerce engel görür. Mekanın üç boyutlu görsel bir haritasını çıkarmaya çalışırken gördüğü engellerin sayısı ise milyonlara çıkabilir. İstatistiki açıdan bakılınca haritadaki engellerin sayısı (harita boyutu), robot pozisyonu ve harita kestirimi problemi için kurulacak tahmin modelinin boyutunu da etkiler. Sonuç olarak haritalama probleminin oldukça yüksek boyutlu olduğunu söyleyebiliriz.

3.1.3 Veri İlişkilendirme Problemi

Haritalamadaki üçüncü ve en zor problem “correspondence” adıyla da bilinen “data association” (veri ilişkilendirme) problemidir. Veri ilişkilendirme problemi, robotun farklı bir zamanda aynı yerden geçtiğini ya da aynı fiziksel nesneleri gördüğünü anlaması problemidir. Şekil 3.2’de bu probleme bir örnek görülmektedir. Robot dairesel bir mekanda gezinirken başladığı yere döndüğünü fark etmiş olmalıdır. Ancak Şekil 3.2’de de görüldüğü gibi dairesel tur bittiğinde robotun üretmiş olduğu yörünge hatası oldukça büyüktür. Robot gezinimi esnasında, aynı yerden geçilmiş olması hipotezi zamanla üstel olarak arttığı için veri ilişkilendirme problemini takip edip çözmek oldukça güçtür. Literatürde bu problemi çözmek için önerilen yöntemler çoğunlukla son beş yılda üretilmiştir. İlk yıllarda robotikle uğraşan topluluklar uzun bir dönem boyunca bu problemi yok saymışlardır.



Şekil 3.2 Dairesel bir ortamda gezinirken oluşan veri ilişkilendirme hatası (Thrun, 2002).

3.1.4 Çevrenin Zamanla Değişmesi

Dördüncü problem çevrenin zamanla değişmesidir. Çevrenin değişimi bazen bir ağacın farklı mevsimlerde görüntüsünün değişmesi gibi yavaş; bazen de bir kapının açılıp – kapanması; ya da odadaki mobilyaların, sandalyelerin yerinin değiştirilmesi gibi hızlıdır. Robotun gezindiği ortamın değişimlere açık ve dinamik bir ortam olması haritalama ve konum belirlemeyi güçleştirir. Zaten literatürde dinamik çevrelerin anlamlı ve tutarlı haritasını çıkarabilecek bir eş zamanlı konum belirleme ve haritalama algoritması bulunmamaktadır. Algoritmalarda çoğunlukla, zamanla değişen tek niceliğin robot pozisyonu olduğu, etraftaki nesnelerin statik ve zamanla değişmez olduğu kabulü ağır basar.

3.1.5 Otonom Robot Gezinmesi

Beşinci ve sıklıkla karşılaşılan problemlerin en sonuncusu robotun haritalama yaparken gideceği yolu kendisinin belirlemek zorunda olmasıdır. Seçilen gezinme stratejisi, harita çıkarımı esnasında karşılaşılabilecek risklere ve beklenmedik durumlara karşı gereken tedbirleri almış olmalıdır. Gezinme stratejisi belirleme genelde sezgisel yöntemlerle çözülen zorlu bir problemdir. Harita kestiriminde kullanılan gerçek zamanlı olmayan yöntemler bu problemi yok sayarlar.

3.2 Eş Zamanlı Konum Belirleme ve Haritalama Probleminde Kullanılan Temel Yöntemler

Eş zamanlı konum belirleme ve harita oluşturma problemi için geliştirilen algoritmaların bazısı artımlı yöntemler olduğundan datanın tümünü bilmesine gerek yoktur, sadece bir önceki adımı bilmesi yeterlidir, dolayısıyla gerçek zamanlı uygulamalarda kullanılabilirler ancak bazısı da önceden geçilen tüm adımları bilmek zorundadır dolayısıyla gerçek zamanlı uygulamalarda kullanılamaz. Bunlardan bazıları harita oluşturabilmek için robotun kesin pozisyonunu bilmek zorundadır. Diğerleri ise “veri ilişkilendirme” problemine çözüm bulabilmek için özel olarak geliştirilmiştir. Ancak kullanılan tüm yöntemlerin tek bir ortak özelliği vardır: istatistiksel yöntemlere dayanmaları.

Harita oluşturma ve robotun konumunun belirlenmesi esnasında algılayıcı gürültüsü, ölçüm belirsizliği, kontrol işaretlerinin bir problem yüzünden istendiği gibi uygulaması, vb durumlarla karşılaşılır ve çözüm olarak da istatistiksel kestirim yöntemleri kullanılır.

En çok kullanılan yöntemlerden üçü aşağıda alt başlıklar şeklinde anlatılmaktadır.

3.2.1 Kalman Filtresi Yöntemi

Kalman filtresi eş zamanlı konum belirleme ve harita çıkarımında kullanılan en eski ve klasik yöntemdir. Kalman Filtreleri soncul (posterior) durum tahmini $p(s_t; m \mid z_t; u_t)$ 'yi Gauss dağılımı ile gösteren özel Bayes Filtreleridir (Kalman, 1960; Maybeck, 1979). Robotik haritalama probleminde Gauss modeli tüm durumu içeren x vektörüdür. Bu vektör de “s” robot pozisyonunu ve “m” o anki haritayı içerir:

$$x_t = (s_t; m)^T \quad (3.1)$$

şeklinde gösterilir.

z_t : t anında gözlenen ölçüm değerleri, u_t ise t anında üretilen kontrol işaretidir.

T: vektör veya matrisin transpozisini göstermektedir. Robot düzlemsel bir yüzeyde haritalama işini yaparken robot pozisyonu “s” üç parametre ile modellenir:

- Düzlemdeki kartezyen koordinatları: s_x ve s_y ile
- Kafa açısı, bizim sistemimiz için ön tekerin robot gövdesiyle yaptığı açısı: s_θ

Kalman Filtresi yönteminde haritalar, robotun gezinirken bulunduğu nesneler kümesinin kartezyen koordinatları ile ifade edilir. Bu nesneler çoğunlukla duvar gibi sınırlar olur. Haritada bulunan nesnelerin adeti “K” olsun. Dolayısıyla x durum vektörünün boyutu $2K+3$

olur:

$$\mathbf{x}_t = (s_{x;t}, s_{y;t}, s_{\theta;t}, m_{1;x;t}, m_{1;y;t}, m_{2;x;t}, m_{2;y;t}, \dots, m_{K;x;t}, m_{K;y;t})^T \quad (3.2)$$

$m_{k;x;t}, m_{k;y;t}$ haritadaki k 'nci nesnenin kartezyen koordinatlarıdır.

Kalman filtresi yönteminde soncul durum tahmini robot pozisyonu ve o ana kadar elde edilen haritanın bileşik olasılığı ile ifade edilir: $p(s_t, m | z^t, u^t)$

Bu dağılımın ortalaması μ_t , kovaryans matrisi ise Σ_t 'dir. Ortalama vektörün boyutu $2K+3$ 'tür, dolayısıyla kovaryans matrisinin boyutu $(2K + 3)^2$ 'dir.

Kalman filtresi haritalama algoritması üç varsayıma dayanarak geliştirilir:

- Hareket modeli (bir sonraki duruma geçiş fonksiyonu) lineer olmalı ve gürültüsü Gauss dağılımına sahip olmalı
- Aynı özellik kavramsal model için de geçerlidir,
- İlk durum belirsizliği Gauss dağılımına sahip olmalı

Haritalama probleminde zaten s_t robot pozisyonu ve m_t haritası doğrusal olarak bir önceki adımdaki s_{t-1} ve m_{t-1} 'e bağlıdır. Aynı doğrusal bağıntı u_t kontrol girişi için de geçerlidir. Bayes filtresinden yola çıkarak elde edilen Kalman Filtresi denklemleri şöyledir:

$$\mu'_{t-1} = \mu_{t-1} + B u_t$$

$$\Sigma'_{t-1} = \Sigma_{t-1} + \Sigma_{\text{control}}$$

$$K_t = \Sigma'_{t-1} C^T (C \Sigma'_{t-1} C^T + \Sigma_{\text{measure}})^{-1} \quad (3.3)$$

$$\mu_t = \mu'_{t-1} + K_t(o - C \mu'_{t-1})$$

$$\Sigma_t = (I - K_t C) \Sigma'_{t-1}$$

μ_t 'nin x durumunun ortalaması, Σ_t 'nin de kovaryans matrisi olduğu verilmiştir.

B ve C : Bir durumdan diğerine geçiş matrisleridir.

Σ_{control} : kontrol girdisinin kovaryans matrisi

Σ_{measure} : ölçümün kovaryans matrisidir

K_t : Kalman kazanımıdır.

Kalman filtresi yöntemi çok sık kullanılmasına rağmen veri ilişkilendirme gibi önemli bir

probleme çözüm getirememektedir. Kalman filtresinin haritalamadaki kısıtları:

- Yapay landmark (sınır - duvar) sınırlara ihtiyaç duyulur,
- Veri ilişkilendirme problemini çözemez,
- Direk ham algılayıcı datası kullanılamaz, veriler ön işlemlerden geçirilmeli,
- Algılayıcı datasının sahip olduğu gürültü tipi Gauss tipinde olmalı,
- Başlangıçta robot pozisyonunda kayma olmadıysa harita iyi çıkar ancak kayma olduysa, zaman geçtikçe bu kayma büyüyeceğinden kötü sonuçlar alınır. Şekil-3.1’de kaymadan kaynaklanan yanlış harita çıkarımına örnek verilmiştir.

3.2.2 Expectation Maximization – Beklenti Enbüyültme (BE) Yöntemi

Son yıllarda eş zamanlı konum belirleme ve haritalama amacıyla Kalman Filtresi uygulamalarına bir alternatif olarak Beklenti Enbüyültme algoritmaları geliştirilmektedir. Beklenti Enbüyültme, maximum likelihood (en büyük olabilirlik) tahmin modelinden üretilmiş, saklı değişkenleri olan istatistiksel bir algoritmadır (Dempster vd., 1977).

Beklenti Enbüyültme yönteminde verinin işlenmesinde artımlı bir model kullanılmadığından aynı data birçok kez işlenmek zorundadır. Başka bir deyişle model sadece bir önceki adımdaki duruma bağlı değil, o ana kadar olan tüm durumlara bağlıdır. Beklenti Enbüyültme’nin Kalman’dan en belirgin farkı budur.

Verilen bir harita için robot pozisyonunun soncul durumunun hesaplandığı bir beklenti B-adımı ve verilen robot pozisyonu kullanılarak en olası haritanın hesaplandığı bir en büyültme E-adımı bulunmaktadır. Sonuçta gittikçe kesinleşen ve gerçek ortama benzeyen bir dizi harita elde edilir. $m^{[0]}$; $m^{[1]}$; $m^{[2]}$; ... Başlangıç haritası $m^{[0]}$ boş bir haritadır.

Beklenti Enbüyültme algoritmasının en önemli avantajı veri ilişkilendirme problemini çözmesidir. Bu probleme şimdiye kadar getirilmiş en iyi çözüm Beklenti Enbüyültme ile sunulmuştur. Kalman filtresi yöntemleri bu konuda Beklenti Enbüyültme’ye yetişemezler ancak Beklenti Enbüyültme gerçek zamanlı çalışamayan bir yöntemdir. Şekil 3.3’te görülen harita Beklenti Enbüyültme ile elde edilmiştir, bu haritanın elde edilmesi için BE algoritmasının saatlerce çalışması lazımdır.



Şekil 3.3 Beklenti Enbüyültme ile elde edilen harita (Thrun, 2002)

3.2.3 İşgal Edilen Alanlar Haritaları

“İşgal edilen alanlar haritaları” (occupancy grid) olarak bilinen haritalama algoritmaları ilk defa 1980’lerin sonlarına doğru Moravec (1988) ve Elfes (1989) tarafından geliştirilmiş ve popülerlik kazanmıştır. Algoritma, otonom bir robot tarafından yukarıda anlatılan algoritmalarından biri ile birlikte kullanılır. Buradaki amaç gürültülü ve eksik gelen algılayıcı verisinden yararlanarak tutarlı metrik bir harita oluşturmaktır. İşgal edilen alanlar haritaları, iki boyutlu gridlerden oluşur ancak istenirse üç boyutlu gösterim de yapılabilir. Standart işgal edilen alanlar haritalama algoritması, diğer ünlü harita algoritmaları gibi Bayes Filtresine dayanır. Bayes filtreleri özellikle her bir grid hücresinin işgal edilip edilmediğinin soncul tahminini hesaplamada kullanılır. Şekil 3.4(a)’da işlenmemiş sonar algılayıcı verileri kullanılarak oluşturulan İşgal edilen alanlar haritası; Şekil 3.4(b)’de ise BE haritalama algoritmasından sonra işgal edilen alanlar haritalama algoritmasının uygulanmasının sonucu görülmektedir.



Şekil 3.4 (a) Ham sonar algılayıcı verisi kullanan işgal edilen alanlar algoritmasının sonucu.

(b) BE’den sonra işgal edilen alanlar algoritması uygulanmasının sonucu (Thrun, 2002)

İşgal edilen alanlar haritalama algoritması son derece güvenilir ve uygulaması kolay olması nedeniyle oldukça popülerdir.

Bu yöntemin en önemli eksiği robot pozisyonu belirsizliği problemini çözemesidir. Bir diğer eksiği ise Bayes filtresinden kalmadır, gürültünün bağımsız olduğunu kabul eder. Algılayıcı gürültüsü bağımlı olduğundan, sonuç haritası hatalı çıkabilir.

3.3. Kullanılan Yöntemlerin Kıyaslanması

Çizelge 3.1’de eşzamanlı konum belirleme ve haritalama problemi için literatürdeki en bilindik üç yöntemin kıyaslaması verilmiştir.

Çizelge 3.1 Yöntemlerin kıyaslanması (Thrun, 2002)

Özellik	Kalman	BE	İşgal edilen alan algoritması
Yakınsama	Güçlü	Zayıf	Güçlü
Gerçek – zamanlı	Evet	Hayır	Evet
Artımlı algoritma	Evet	Hayır	Evet
Çok güvenilir robot konumuna ihtiyaç var mı?	Hayır	Hayır	Evet
Algılayıcı gürültü tipi	Gauss	Herhangi bir değer	Herhangi bir değer
Veri ilişkilendirme	Hayır	Evet	Evet
Ham veri kullanılabilir mi?	Hayır	Evet	Evet

Çizelge 3.1’de de görüldüğü üzere BE, sadece bir önceki duruma değil o ana kadar gözlenen tüm durumlara bağlı olduğundan artımlı ve gerçek zamanlı değildir. Kalman ve İşgal edilen alanlar artımlı ve gerçek zamanlı yöntemlerdir. Kalman’da algılayıcı gürültüsünün dağılımı Gauss olmak zorundadır, diğerlerinde böyle bir kısıt yoktur. Kalman veri ilişkilendirmeye çözüm bulamaz ancak diğerleri bulur.

4. SIRALI MONTE CARLO YÖNTEMİ

Eş zamanlı konum belirleme ve harita çıkarımının hedefi robot pozisyonunu ve ortamın haritasını eş zamanlı olarak çıkarmaktır. Bu süreçte istatistiksel bir tahmin yöntemi kullanılmalıdır.

Literatürde yer alan istatistiksel yöntemlerden en yaygın olarak kullanılanı Kalman Filtresidir. Özellikle gerçek zamanlı uygulamalarda Kalman tercih edilir. Smith vd. (1990) 1985-1990 yılları arasında SLAM problemine bir çözüm getiren ve günümüzde hala etkin olarak kullanılan bir Kalman Filtresi uygulaması önermişlerdir. Daha sonra birçok araştırmacı bu yöntemi geliştirmiştir. Ancak Kalman Filtresi yöntemi “veri ilişkilendirme” adıyla bilinen, “robotun geçtiği bir yerden tekrar geçtiğini anlaması” problemine çözüm getiremez. Ayrıca Kalman filtresi yönteminde ham algılayıcı datası kullanılamaz, data ön işlemlerden geçirilmelidir ve algılayıcı datasının sahip olduğu gürültünün dağılımı Gauss tipinde olmak zorundadır. Bu kısıtlardan ve veri ilişkilendirme gibi önemli bir probleme çözüm getiremeyişinden dolayı Kalman filtresi uygulamamızda kullanılmamıştır.

Veri ilişkilendirme problemine güçlü bir çözüm getiren; ham algılayıcı datasını direk kullanabilen ve datanın gürültüsünün nasıl olduğuyla ilgilenmeyen başka bir istatistiksel yöntem ise “Beklenti Enbüyütme” dir. Thrun vd. (1998) BE yöntemini kullanarak SLAM algoritması geliştirilirken karşılaşılan veri ilişkilendirme problemine bugüne kadar getirilmiş en başarılı çözümü sunmuşlardır. Ancak bu yöntem gerçek zamanlı uygulanamamaktadır. Sistemimiz gerçek zamanlı olacağından çözüm olarak BE yöntemi de tercih edilmemiştir.

Literatürde uygulanan çoğu “parçacık filtresi” yöntemi gerçek zamanlı çalışırken Kalman filtresi kadar etkin ve hızlı olamasa da, hem gerçek zamanlı çalışması hem de veri ilişkilendirme problemine çözüm getirebilmesi hasebiyle Yuen ve MacDonald (2002) tarafından da tercih edilmiştir.

4.1 Sıralı Monte Carlo Yönteminin Özellikleri

Literatürde Parçacık filtreleri Sıralı Monte Carlo (SMC) adıyla bilinir. SMC, sistemin soncul durum dağılımlarını hesaplamak amacıyla geliştirilen, çoğunlukla simulasyon-tabanlı, kullanışlı ve etkili bir istatistiksel tahmin yöntemidir. Temel özellikleri:

- Esnek
- Uygulaması kolay
- Parallelleştirilebilir

- Çok çeşitli veri kümelerinde kullanılabilir
- Veri, Gauss (normal) dağılıma sahip olmak zorunda değil
- Bayes Kuralı'na dayanır

Sistemin t anındaki durumuna x_t , t anında alınan gözleme ve üretilen kontrol işaretlerine de z_t diyelim.

x_t durum değişkeni tahmin edilmek istenen tüm bilgiyi içerir.

- Robot pozisyonunun tahminini ve
- Robotun çevresindeki engel pozisyonlarının (haritanın) tahminini içerir.

Daha sonra bu iki kavram ayrı ayrı değerlendirilecektir.

Veri kümesi z_t de ikiye bölünür:

- Kavramsal veri (y_t) : algılayıcılardan en son alınan mesafe verisi
- Kontrol verisi (u_{t-1}) : bilgisayardan robota gitmek üzere üretilen en son kontrol verisi

Tahmin sürecinde Bayes Teoremi ile soncul dağılımı şu şekilde bulunur:

$$P(x_{t+1}|z_{t+1}) = P(x_t) P(z_{t+1} | x_{t+1}) / P(z_{t+1} | x_t) P(x_t) \quad (4.1)$$

SMC, dinamik bir sistemin x saklı durumunu ölçümlerden ve kontrol işaretlerinden yola çıkarak tahmin eder. Robot lokalizasyonu probleminde dinamik sistemimiz gezgin robot ve robotun çevresinin iki boyutlu kartezyen koordinatlarından oluşur. Sistemimizde kullanılan gezgin robotun konumu için x_t durumu, robotun t anındaki referans vektörünün başlangıç ve bitiş noktalarının kartezyen koordinatlarını, robot gövdesinin düzleme göre eğim açısını ve o ana kadar bulunan engellerin kartezyen koordinatlarını içermektedir. SMC'nin temel dayanağı, soncul durum olasılık yoğunluk fonksiyonunu (pdf), ölçüm verisine ve kontrol verisine göre şekillenen bir durum uzayıyla tahmin etmesidir. Robotik ve yapay zeka uygulamalarında soncul duruma “belief” (kanaat) denir ve soncul x_t , gözlemlerden ve kontrol girişlerinden elde edilebilir.

z_t : y_t algılayıcı ölçümleri verisi ve u_t kontrol işaretleri verisinden oluştuğuna göre kanaat:

$$\text{Bel}(x_t) = P(x_t | y_t, u_{t-1}, y_{t-1}, \dots, u_0, y_0) \quad (4.2)$$

olur.

x_t : sistemin t anındaki durumu idi; $y_{1:t}$ ise: 1'den t anına kadar geçen süredeki algılayıcı datasını göstermektedir. Kanaat, Bayes kuralı ile şu hale gelir:

$$P(x_{0:t+1}|y_{1:t+1}) = P(x_{0:t}|y_{1:t}) P(y_{t+1} | x_{t+1}) P(x_{t+1} | x_t) / P(y_{t+1}|y_{1:t}) \quad (4.3)$$

Bayes kuralıyla yazılan kanaat denklemi, özyinelemeli bir parçacık filtresi gösterimiyle yazılarak bir estimator (kestirici) elde edilir. n:bir sabit olmak üzere:

$$Bel(x_{t+1}) = P(x_{t+1}|y_{t+1}, u_t) = n P(y_{t+1} | x_{t+1}) \int P(x_{t+1} | x_t, u_t) Bel(x_t) dx_t \quad (4.4)$$

4.4'teki bu kestirim bağıntısı SMC algoritmasında durum ve ölçüm güncellemesinde kullanılmıştır.

E : M adet parçacıktan oluşan bir parçacık kümesi olsun. E kümesindeki her bir parçacık için çalıştırılan SMC yönteminin en genel algoritma adımları aşağıda verilmiştir:

For (i = 1 : M)

1. Her bir parçacığa önem katsayısı ataması
2. Durumun güncellenmesi
3. Ölçümün güncellenmesi

End for

4. Önem faktörü değerlerinin toplamı 1'i verecek şekilde normalize edilmesi.

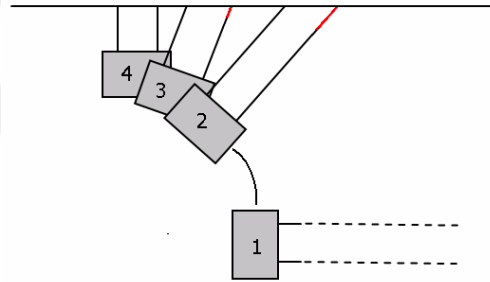
Başlangıçta her bir parçacığa rasgele birer önem katsayısı atanır. Rekürsif güncelleme filtresi algoritmada görüldüğü gibi 3 adımdan oluşmaktadır. Öncelikle bir “i” parçacığı E kümesinden seçilir. İkinci adımda sonraki durum tahmini olan x_t^i ve bunun şartlı olasılığı olan $P(x_t | x_{t-1}, u_{t-1})$, en son kontrol girdisi u_{t-1} ve bir önceki parçacığın durum değişkeni x_{t-1}^i kullanılarak hesaplanır. Son adımda da $P(y_t | x_t^i)$ ölçümü güncellenir. Bu algoritma eş zamanlı konum belirleme ve haritalama problemine çözüm getirebilmek amacıyla bölüm 5.5'te anlatıldığı gibi gereken modifikasyonlara tabi tutulmuştur. (Doucet vd., 2001)

5. SİSTEMDE KULLANILAN EŞ ZAMANLI KONUM BELİRLEME ve HARİTA OLUŞTURMA ALGORİTMASI

Robotun gezinirken belli bir algoritma izleyebilmesi için yapması beklenen bazı temel davranışlar bulunmaktadır. Öncelikle bunlara değinilmelidir:

5.1 Robotun sahip olması gereken temel davranışlar

- **Dönme:** Robot, gönderilen kontrol işaretinde belirtilen kafa açısıyla dönebilmelidir.
- **Duvar bulma:** Mekanın herhangi bir yerine bırakılan robot önce bir duvara veya engele ulaşana kadar ilerler. Duvarı bulduktan sonra kendisini duvar ile paralel hale getirir. Şekil 5.1’de gösterildiği gibi 1 konumunda öndeki algılayıcısıyla duvarı algılayan robot, yan taraflarında yer alan algılayıcı ölçümleri birbiriyle aynı olana kadar döner. 2 ve 3 konumunda algılayıcıların ölçtükleri mesafeler arasındaki farklar görülmektedir. 4 konumunda ise iki algılayıcının değerleri aynıdır. Artık robotla duvar paralel hizaya gelmiştir ve duvar izleme adımına geçilebilir.



Şekil 5.1: Duvar bulma adımından duvar izleme adımına geçiş

- **Duvar izleme:** Robotun yanlarında yer alan algılayıcılarının ölçüm değerlerinin arasındaki fark belirli bir eşik değerinden küçükse robot kendisine paralel bir duvar boyunca ilerlemeye devam edecektir.
- **Yavaşlama:** Aracın dönmeye başlamadan önce yavaşlaması beklenmektedir.
- **Hızlanma:** Dönme işleminin tamamlanmasından sonra, önünde başka bir engel yoksa araç hızlanmalıdır.

5.2 Gezinme Algoritması

Sistemde, Huang ve Beevers (2005) tarafından basit harita çıkarımı için geliştirilen algoritma kullanılmıştır. Bu algoritmanın adımları, aşağıdaki gibi özetlenebilir:

1. Robotu mekanda rasgele bir yere bırakma.
2. Duvar bulma
3. Kendini bulduğu duvara paralel hale getirme
4. Bir iç yada dış köşeye rastlayana kadar duvarı izleme. İlk bulunan köşeyi kendine *başlangıç noktası* olarak seçme
5. 90 derece dönüp yine bir köşeye rastlayana kadar duvarı izleme, duvar izlenirken bulunan noktaları haritaya ekleme.
6. Beşinci adımı başlangıç noktasına geri gelinceye kadar tekrar etme.

Yukarıdaki adımlar istatistiksel tahmin yöntemi olan Sıralı Monte Carlo Algoritması ile gerçekleştirilerek robotun otonom hareket etmesi sağlanmıştır.

5.3 Algılayıcı verisi modellemesi

Robotun, konumunu bilmesi ve harita çıkarabilmesi için algılayıcılardan byte cinsinden gelen ölçüm değerlerini alması ve ne kadarlık mesafede bir engel bulunduğunu anlaması gerekmektedir. Bu amaçla yapılması gereken veri modellemesinin tanımı ve denklemi Bölüm 2.4.6.'da anlatılmaktadır.

5.4 Robot Yörünge Takibi

Gezgin robotun kendini kontrol edebilmek, gezinebilmek ve düzgün bir harita oluşturabilmek için hareketi esnasında yörüngesini bilmesi gerekmektedir.

Gövde üzerindeki dik bir vektörün başlangıç ve bitiş noktalarının kartezyen koordinatları, vektörün düzlemle yaptığı açısı ve açının yönü gövde konfigürasyonunu belirtir. Vektörün başlangıç noktası ön tekerin orta noktası, bitiş de arka tekerleri birleştiren dik doğrunun orta noktası olarak seçilmiştir. Robotun ön tekerinin gövdeyle yaptığı açısı verildiğinde, gövdenin konfigürasyonunu belirleyen probleme ileri kinematik problemi; robotun verilen bir gövde konfigürasyonu için ön teker açısını belirleyen problem ise ters kinematik problemi olarak ifade edilmektedir (Reuter, 1998). Yörünge planlamasını gerçekleştirebilmek amacıyla robotun yapısına uygun kinematik bir denklem seçilmiştir.

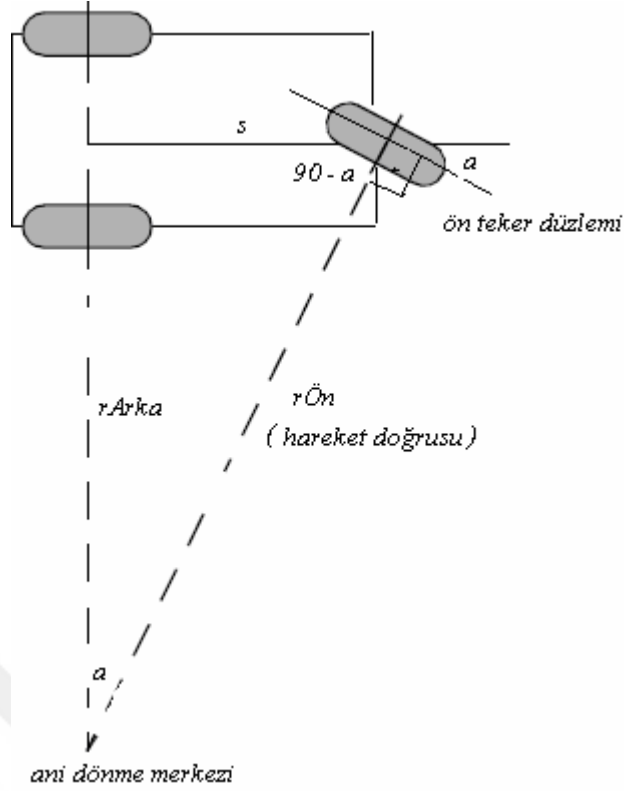
Gezgin robotlarda yörünge planlama üzerine son 20 yıldır oldukça yaygın bir şekilde çalışılmaktadır. Literatürdeki bu çalışmalar,

- a. Kapalı planlamalar ve
- b. Açık planlamalar şeklinde sınıflandırılabilir.

Planlama, gezgin robot hareket halindeyken, gerçek zamanda ve yerel olarak yapılıyorsa buna kapalı planlama denmektedir. Açık planlama da aksine global ve çevrimdışı olarak yapılmaktadır. Sistemimizdeki yörünge problemi kapalı planlama ile çözülmüştür.

Algılayıcı bilgisi ile birlikte gidilen yolun miktarı artımlı encoder'dan, ön tekerin gövde ile yaptığı açı da potansiyometreden okunarak robottan bilgisayara aktarılır. Verilerden yararlanarak robotun konfigürasyonunu veren referans vektörünün bir sonraki koordinatlarını bulmak için gereken kinematik denklemler çıkarılmıştır.

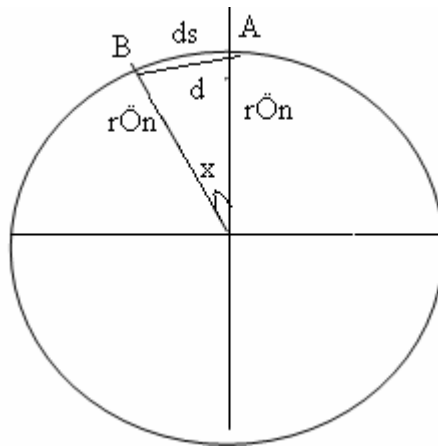
Sistemimizde kullanılan robotun bir adet önde, iki adet de arkada tekeri olduğu anlatılmıştı. Robot gövdesini belli bir doğrultuya yönlendirebilmek için ön tekere yön kontrol işareti gönderilir. Geometrik olarak ön teker düzlemine dik olan yatay bir “hareket doğrusu” (motion line) çizilir. Hareket doğrusu ile arka tekerlerin düzleminin kesiştiği nokta herhangi bir anda yapılan dönme hareketinde arabanın çizeceği yayın merkez noktasıdır. Bu noktaya “ani dönme merkezi” (instantaneous center of rotation) denir ve Şekil 5.2’de görüldüğü gibidir.



Şekil 5.2 Robotun dönme hareketi esnasında oluşan ani dönme merkezi

Ön tekerin çizdiği yayın ait olduğu çemberin yarıçapı hareket doğrusu dediğimiz doğrudur. Arka teker de dönme hareketi esnasında aynı merkezli bir yay çizer. Araba üzerinde seçtiğimiz vektörün bitiş noktası arka tekerlerin orta noktası idi, bu noktanın çizeceği yay ön tekerin çizdiği yaydan daha kısadır. Arka tekerlerin orta noktasının çizdiği yayın ait olduğu çemberin yarıçapı da basit bir geometrik bağıntıyla hareket doğrusu kullanılarak elde edilir.

Robotun ön tekerinin orta noktasının yaptığı hareket Şekil 5.3'te görülmektedir:



Şekil 5.3 Ön teker orta noktasının çizdiği yay ve yayın ait olduğu çember

Robotun ön tekerinin orta noktasının A noktasından B'ye gelene kadarki hareketinde çizdiği yayın uzunluğu, artımlı encoder'dan okunan veridir.

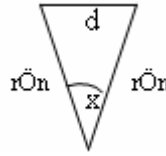
- Artımlı encoder'dan alınan yer değiştirme bilgisi “ds” olsun.
- Potansiyometre'den okunan, ön tekerin gövdeyle yaptığı açı değeri “a” olsun
- Takip edilen referans vektörünün boyu “s” olsun, dolayısıyla dairenin çapı $r_{\text{Ön}} = s / \sin(a)$ olur.
- “x” açısı ön tekerin yaptığı hareketin açısal hızıdır aynı zamanda çizilen yayı gören merkez açıdır. Aşağıdaki bağıntıdan çekilerek bulunur:

$$ds = 2 * \pi * r_{\text{Ön}} * x / 360 \quad (5.1)$$

ds ve r biliniyorken x:

$$x = 360 * ds / (2 * \pi * r_{\text{Ön}}) \quad (5.2)$$

- x'i bulduktan sonra ön tekerin orta noktasının aldığı yolu yani “d” uzaklığını bulmak için kosinüs teoremi kullanılır. Oluşan ikizkenar üçgen Şekil 5.4'teki gibidir:



Şekil 5.4 Oluşan ikizkenar üçgen

Kosinüs teoreminden d mesafesi:

$$d^2 = r_{\text{Ön}}^2 + r_{\text{Ön}}^2 - 2 * r_{\text{Ön}} * r_{\text{Ön}} * \cos(x) \quad (5.3)$$

şeklindedir.

- Vektörün harekete başladığı t anında takip ettiğimiz referans vektörünün başlangıç noktası koordinatları (yani ön tekerin orta noktası) x_{1t} ve y_{1t} ; bitiş noktasının koordinatları (yani arka tekerlerin orta noktası) x_{2t} ve y_{2t} olsun. Robot gövdesinin x-y düzlemine göre eğimi de m_t olsun. (x_{1t}, y_{1t}) ve (x_{2t}, y_{2t}) bilinirken m_t :

$$m_t = \arctan [(y_{2t} - y_{1t}) / (x_{2t} - x_{1t})] \quad (5.4)$$

- Robotun düzleme göre eğimi biliniyorken referans vektörünün başlangıç noktasının (ön teker orta noktasının) yeni koordinatları şöyle bulunur:

$$x_{1(t+1)} = x_{1t} - d * \cos (m_t - a - x/2) \quad (5.5)$$

$$y_{1(t+1)} = y_{2t} - d * \sin (m_t - a - x/2) \quad (5.6)$$

Aynı şekilde referans vektörümüzün bitiş noktasının yani arka tekerlerin orta noktasının yeni pozisyonunun da güncellenmesi gerekmektedir. Ön teker ve arka tekerlerin, dönme hareketi dönme merkezleri de aynıdır. Ancak çizilen yayın yarıçapları farklıdır ve Şekil 5.1’de de görüldüğü gibi aralarında aşağıdaki ilişki vardır:

$$r_{Arka} = r_{Ön} * \cos(a) \quad (5.7)$$

Ön teker ve arka tekerlerin, dönme hareketi boyunca sahip oldukları çizgisel hızlar farklıdır ancak açısal hız aynıdır. Açısal hız da çizilen yayları gören merkez açı yani “x” değeridir. Buna göre arka tekerlerin orta noktasının çizdiği yay:

$$ds_{Arka} = 2 * \pi * r_{Arka} * x / 360 \quad (5.8)$$

şeklinde bulunur. Arka tekerlerin orta noktasında da Şekil 5.4’teki gibi bir üçgen oluşur. Yine kosinus teoreminden arka tekerlerin orta noktasının kartezyen koordinatlardaki yer değiştirme uzunluğu bulunur:

$$d_{Arka}^2 = r_{Arka}^2 + r_{Arka}^2 - 2 * r_{Arka} * r_{Arka} * \cos(x) \quad (5.9)$$

Referans vektörünün bitiş noktasının (arka tekerlerin orta noktasının) yeni koordinatları şöyle bulunur:

$$x_{2(t+1)} = x_{2t} - d_{Arka} * \cos (m_t - x/2) \quad (5.10)$$

$$y_{2(t+1)} = y_{2t} - d_{Arka} * \sin (m_t - x/2) \quad (5.11)$$

Robot pozisyonunu güncellerken kullanılan (5.5), (5.6), (5.10) ve (5.11) bağıntıları lineer bir sisteme aittir. Bu bağıntılar çoğunlukla aynı sonucu veren fakat daha kolay çözülebilen başka bir lineer sisteme dönüştürülür ve matris çarpımı şeklinde ifade edilir. Eski robot pozisyonundan yeni pozisyona geçmek için gereken matrise rotasyon matrisi denir.

Buna göre $\theta_1 = m_t - a - x/2$ olmak üzere ön teker orta noktası için gereken rotasyon matrisi:

$$\begin{pmatrix} 1 & 0 & -\cos\theta_1 \\ 0 & 1 & -\sin\theta_1 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.12)$$

Dolayısıyla yeni ön teker orta nokta koordinatları hesaplanırken:

$$\begin{pmatrix} x_{1(t+1)} \\ y_{1(t+1)} \\ d \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\cos\theta_1 \\ 0 & 1 & -\sin\theta_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1t} \\ y_{1t} \\ d \end{pmatrix} \quad (5.13)$$

Matris sistemi kullanılır.

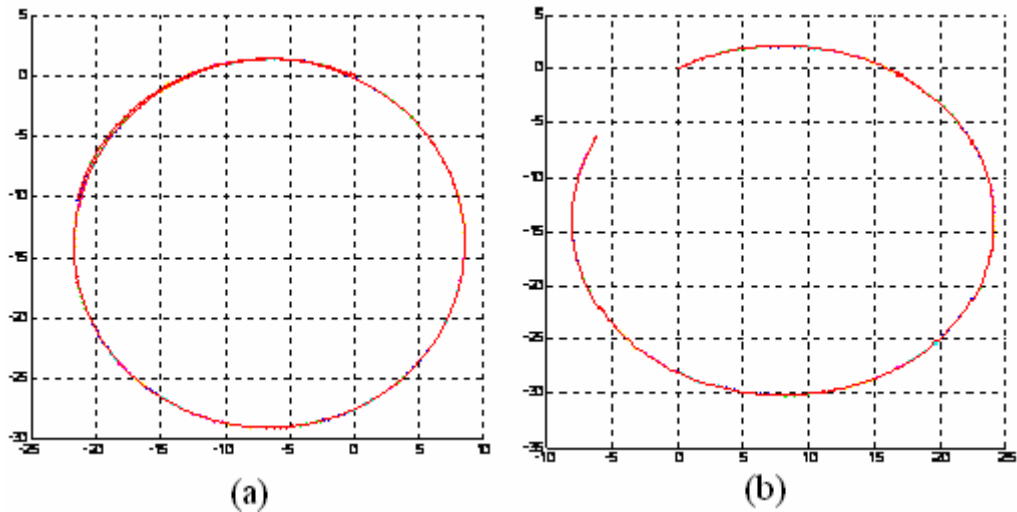
Aynı şekilde $\theta_2 = m_t - x/2$ olmak üzere arka teker için gereken rotasyon matrisi:

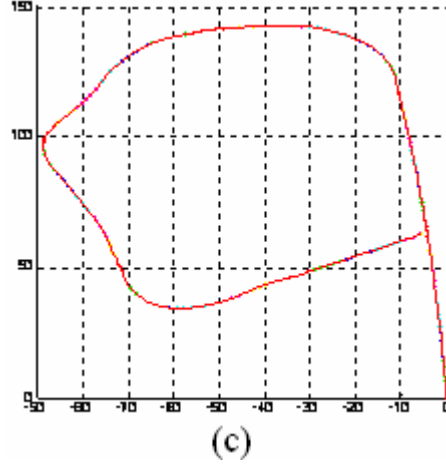
$$\begin{pmatrix} 1 & 0 & -\cos\theta_2 \\ 0 & 1 & -\sin\theta_2 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.14)$$

olur. Kullanılacak matris sistemi ise şöyle olur:

$$\begin{pmatrix} x_{2(t+1)} \\ y_{2(t+1)} \\ d \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\cos\theta_2 \\ 0 & 1 & -\sin\theta_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{2t} \\ y_{2t} \\ d \end{pmatrix} \quad (5.15)$$

Robot sabit açıyla sürekli sağa dönerken izlediği yörünge Şekil 5.5(a)'da, sürekli sola dönerken izlediği yörünge Şekil 5.5(b)'de ve ters "P" yaparken izlediği yörünge Şekil 5.5(c)'de görülmektedir. (x-y ekseninin birimi cm'dir.)





Şekil 5.5 (a) Sürekli sola dönerken izlenen yörünge; (b) Sürekli sağa dönerken izlenen yörünge (c) Ters P çizerken izlenen yörünge

5.5 Eş zamanlı konum belirleme ve harita oluşturma amacıyla geliştirilen SMC algoritması

Bölüm 4.1’de anlatılanlardan görüldüğü üzere Sıralı Monte Carlo algoritması da diğer popüler eş zamanlı konum belirleme ve harita oluşturma algoritmaları gibi Bayes filtresinden üretilmiştir. Sistemimizde kullanılan SMC algoritmasına geçmeden önce gerekli bazı tanımlamalar verilmelidir.

Robot algılayıcılarının gezinirken çevrede algıladığı farklı karakteristiğe sahip her bir nesne ya da engeli “landmark” denir. Yeni bir landmark bulunduğunda sistemde her seferinde yeni bir parçacık oluşturulur. Sistemimizde her bir landmark dolayısıyla her bir parçacık filtresi, bir duvara ya da engeli karşılık gelir. Dolayısıyla algoritma sonucunda “m” adet parçacık filtresi oluştuysa “m” adet duvar bulunmuş demektir. Robot gezinirken her duvara ait çeşitli sayıda noktalar tespit eder, dolayısıyla her bir parçacık kümesinde çeşitli sayıda eleman bulunur diyebiliriz. Her yeni bir landmark bulunduğunda bu landmarkın iç köşe mi dış köşe mi oluşturduğunu belirten bir önem faktörü atanır.

Robot pozisyonu zamanla değişir ama engeller aynı yerinde kalır. Dolayısıyla robot pozisyonu güncellemesi tipik bir durum tahmin problemidir. Ama engel pozisyonu güncellemesi parametre tahmini olarak düşünülebilir. SMC, genelde durum tahminlerinde kullanılır. Tüm durum tahmin vektörü olan x bileşenlere ayrılır ve her bir bileşen için ayrı bir kestirici geliştirilir. Her bir filtre, tüm durumun farklı bileşenlerini tahmin eder. Buna göre

öncelikle sistem modelinin ve ölçüm güncelleme modelinin nasıl olması gerektiği tanımlanmalıdır.

5.5.1 Sistem Modeli

Öncelikle x durumu iki bileşene ayrılır.

$$X = (s, \rho)$$

* s : zamanla değişen kısım yani robot pozisyonu,

* ρ : zamandan bağımsız kısım yani engellerin haritası

İkinci olarak parametrelerin tahmini yani harita durum bileşeni olan ρ , her bir i engeli için tekrar bileşenlere ayrılır:

$$\rho_i = 1:L$$

ve her bir i engeli için ayrı tahminler yapılır.

Ayrı parçacık filtreleri robotun pozisyonu ve her bir engelin pozisyonlarının tahmininde kullanılır. Tahmin problemi alt parçalara bölünerek ve bileşenler arasındaki bağımsızlık derecesi artırılarak hesaplama yükü azaltılmıştır (Doucet vd., 2001).

Robot pozisyonunu tahmin eden filtre ρ_0 durum filtresidir. i 'nci engel pozisyonunu tahmin etmek için atanan filtreler yani parametre filtreleri: $\rho_i = 1:L$ 'dir.

5.5.2 Ölçüm Modeli

Ölçüm modeline genellikle kavramsal model ya da gözlemin güncellenme modeli de denir ve $y_{t+1} = F(y_t, x_{t+1})$ şeklinde gösterilir (Doucet vd., 2001).

Ölçüm modeli, sistemin herhangi bir anda çevreden alması muhtemel olan verileri tahmin etmek amacıyla kullanılır. Gerçek gözlem verileri elde edildikten sonra ise beklenen ölçüm değerleri ile gözlenen ölçüm değerleri arasındaki farka bakılarak Bölüm 5.5.3'te anlatılan algoritma için gereken çıkarımlar yapılır. F fonksiyonu:

- Ne tip algılayıcı kullandığımıza,
- Uygulamamıza,
- İki boyutlu mu üç boyutlu mu çalıştığımıza bağlıdır.

Örneğin algılayıcı olarak kamera kullanılıp belli bir alanın görüntüsü alınıyorsa çeşitli görüntü işleme fonksiyonları kullanılıyor olabilir ya da lazer ve kızılötesi algılayıcılarda olduğu gibi

engelin robota olan uzaklığı alınıp kartezyen koordinatlara taşınıp iki boyutlu geometrik bağıntılar kullanılıyor olabilir. Sistemimizdeki gözlem modeli; algılayıcının bir önceki adımda gördüğü noktaya, algılayıcının bir önceki pozisyonuna ve şu anki pozisyonuna bağlıdır.

Ölçüm modeli güncellemesinin daha anlaşılır kılınması için Şekil 5.6'ya bakılarak gereken geometrik bağıntılar elde edilebilir. Robot (I). konumdayken sol üstteki algılayıcı bir duvar algılamış olsun. Duvarda görmüş olduğu nokta B noktası olsun. Robot aldığı kontrol işaretleri sonucunda (II). pozisyona ulaşır. Bu konumda duvarın devam ettiği düşünülerek D noktasını görmesi gerektiği düşünülür. D noktasını tahmin edilebilmesi için analitik geometri bağıntılarından faydalanılır:

(I). konumda robotun eğimi (m_{Robot}), duvarı algılayan algılayıcının kartezyen koordinatları (A_x, A_y) ve duvarın eğimi (m_{Duvar}) bilinmektedir. Çünkü harita oluşturulurken her adımda eş zamanlı olarak robot pozisyonu da güncellenmektedir. (II). konumda da duvar eğiminin sabit kalacağı aşikardır. Robot pozisyonu her adımda güncellendiğinden aynı algılayıcının yeni pozisyonu (C_x, C_y) de bilinmektedir. Oluşması beklenen CD doğrusunun eğimi, robot kartezyen koordinatlara göre eğimine diktir. Dolayısıyla:

$$m_{CD} = -1 / m_{Robot} \quad (5.16)$$

ile bulunur.

$$m_{CD} = (C_y - D_y) / (C_x - D_x) \quad (5.17)$$

denklemleri elde edilir. Buradan (D_x, D_y)'yi direk tahmin edemeyiz. Birinci dereceden ikinci bir denkleme daha ihtiyacımız vardır.

(I). konumunda ve (II). konumunda duvar eğimi sabit kalacağından:

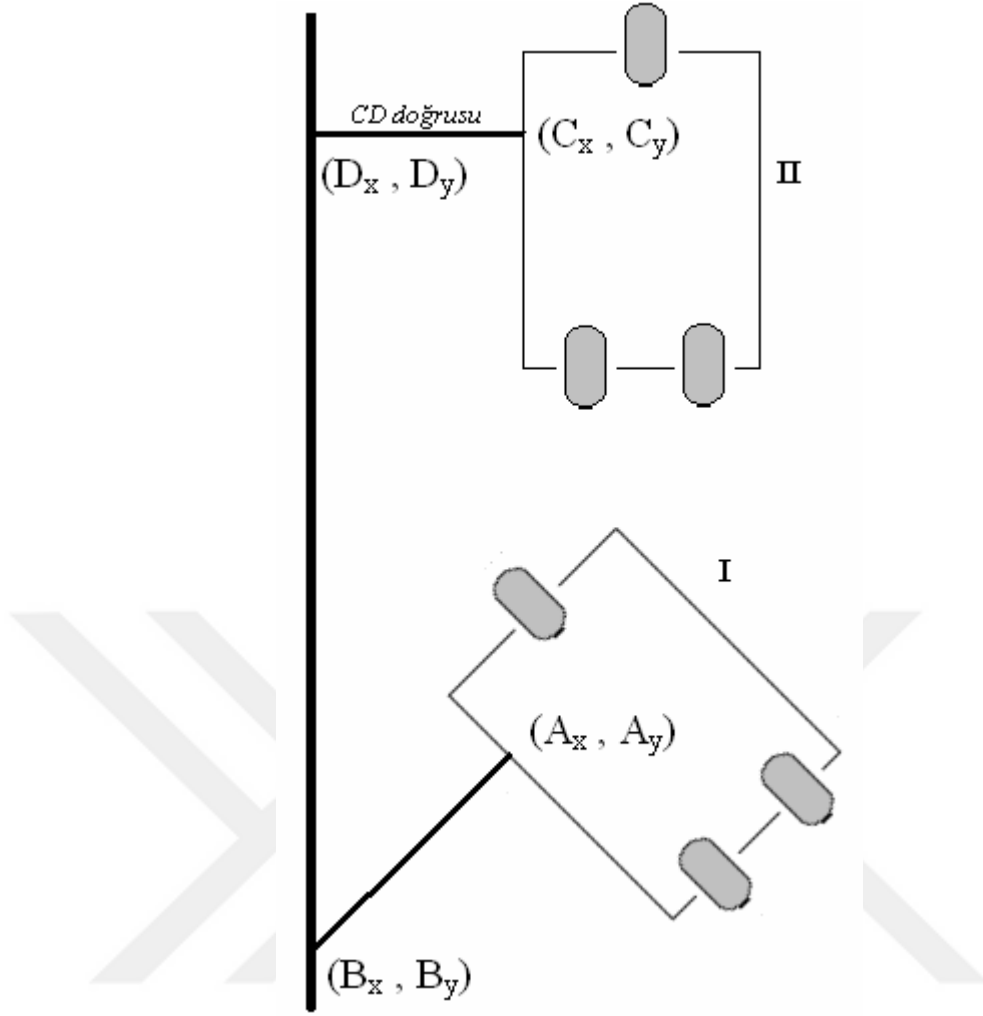
$$m_{Duvar} = (B_y - D_y) / (B_x - D_x) \quad (5.18)$$

(5.13) ve (5.14) bağıntılarında bilinmeyenler (D_x, D_y); bilinenler m_{CD} , m_{Duvar} , (B_x, B_y) ve (C_x, C_y)'dir. D_x ve D_y 'yi çekersek:

$$D_x = (B_y - C_y + m_{CD} * C_x - m_{Duvar} * B_x) / (m_{CD} - m_{Duvar}) \quad (5.19)$$

$$D_y = C_y + m_{CD} * (D_x - C_x) \quad (5.20)$$

olur.



Şekil 5.6 Ölçüm değerleri tahmini

D noktası tahmin edilen ölçüm değerimizdir. Gerçekte duvar devam etmiyorsa yani gerçek ölçüm değerlerine göre robot (II). konumda bir engel yakalayamadıysa solundaki duvarın orada bittiğini anlar, bir sonraki alt başlıkta görüldüğü üzere iç ya da dış köşe bulmuş olmasına bağlı olarak bir kontrol işareti üretir.

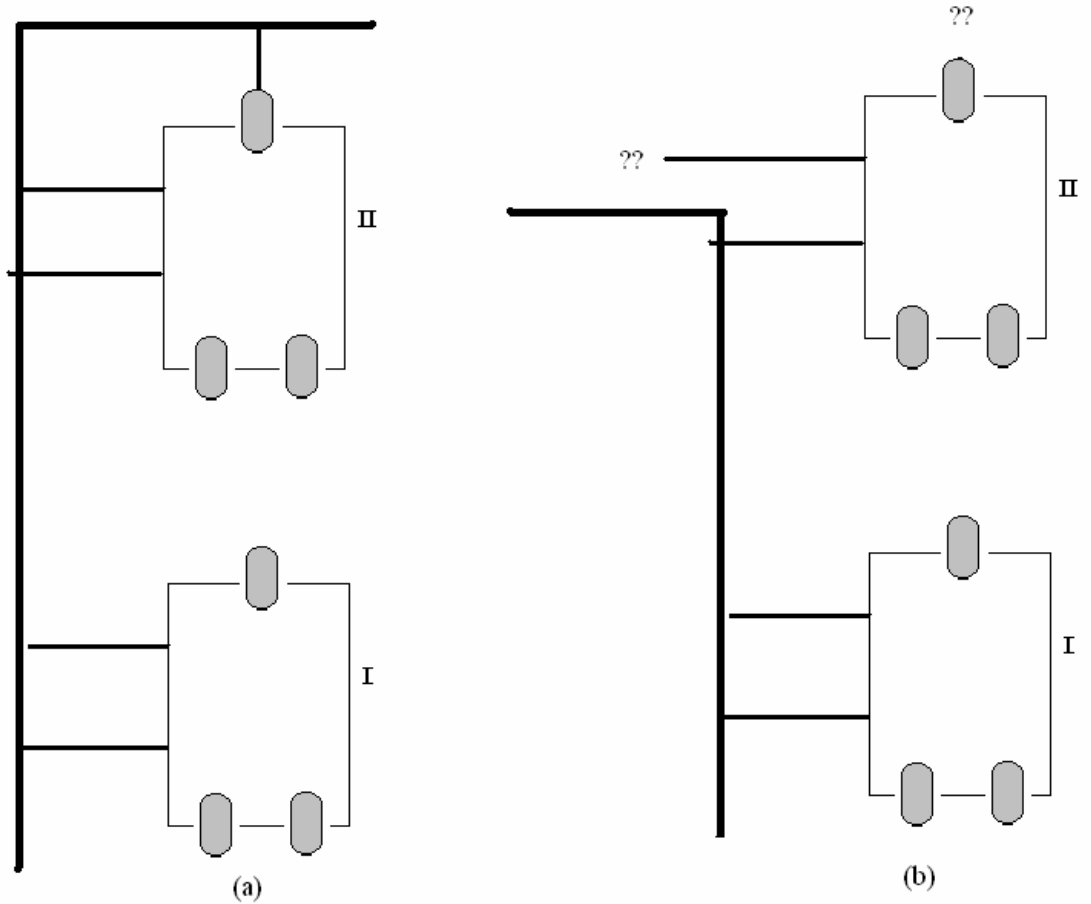
5.5.3 SMC algoritması

Sistem modellemesi ve ölçüm modellemesi tanımlandıktan sonra bölüm 5.2'deki eş zamanlı konum belirleme ve haritalama amaçlı kullanılan gezinme algoritmamız için gereken SMC algoritmasının adımları verilebilir:

- 1) Robot pozisyonu için ρ_0 durum filtresine ilk değer ataması yap.
- 2) $L = 0$ // L : yeni engellerin toplam sayısı. Başlangıçta 0'dır
- 3) Aşağıdaki adımlar robot başladığı yere dönene kadar tekrarlanır :
 - a) Robot pozisyonunun ρ^t_0 güncelle (t anında)

- b) Kavramsal haritayı $\rho_j^t = 1:L$ ile güncelle
 - c) Beklenen ölçüm değerleri y_{prj}^{t+1} hesaplanır
 - d) Gerçek ölçüm değeri y_{act}^{t+1} ile beklenen ölçüm değeri y_{prj}^{t+1} karşılaştırılır
 - e) Eğer yeni bir engel bulunursa aşağıdaki adımlar işleme konacaktır:
 - $L = L + 1$
 - Yeni parametre filtresi ρ_L^t 'yi oluşturup ilk değer ataması yapılır.
 - Engel için önem faktörü ataması yapılır.
 - f) Önem faktörüne göre kontrol işareti üretilmesi ve gönderilmesi.
- 4) Robot başladığı yere dönmediyse 3. adımdan devam edilir.

Görüldüğü üzere her adımda filtre güncelleme, yeni engel bulunup bulunmadığı kontrolleri, ölçüm güncellemeleri yapılmaktadır. Beklenen ölçüm değerlerini bulmak için 5.5.2'de anlatılan kavramsal modeldeki denklemler kullanılarak o andaki robot pozisyonu ve bir önceki adımdaki gerçek ölçümler kullanılarak t anındaki beklenen ölçümler hesaplanmaktadır. Gerçek ölçümler ve beklenen ölçümler arasında belirgin bir fark varsa yeni bir engel bulundu anlamına gelmektedir. Yeni engel bulununca yeni bir parçacık kümesi oluşturulup ilk değer ataması yapılmaktadır. Şekil 5.7 (a)'da bir iç köşe bulunması durumuna örnek; Şekil 5.7 (b)'de ise bir dış köşe bulunmasına örnek verilmiştir.



Şekil 5.7 (a) Bir iç köşe bulma senaryosu, (b) Bir dış köşe bulma senaryosu

Buna göre Şekil 5.7(a)'da (I). ve (II). konumda sol algılayıcılar aynı veriyi alıyor ancak ön algılayıcı eski konumuna göre daha kısa mesafeler algılıyorsa bir iç köşe bulunmuş demektir. Yeni bir parçacık filtresi oluşturulup, bulunan duvarın iç köşe olduğunu belirten bir önem faktörü değeri atanır. Ön algılayıcı yeni algılanan duvara yeterince yaklaşıncaya robot, kendini yeni bulunduğu duvara paralel hale getirmek için dönme hareketi yapar.

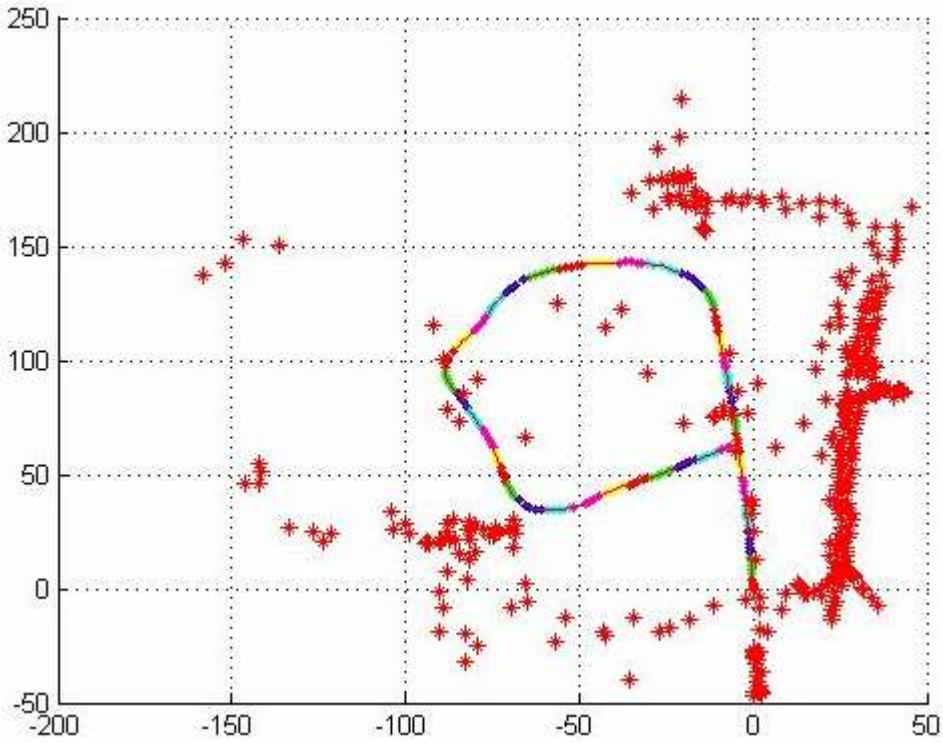
Şekil 5.7 (b)'de ise (I). ve (II). konumda sol algılayıcılar aynı veriyi algılamadığından ön algılayıcı bir engel görse de görmese de bir dış köşe bulunmuş demektir. Yeni bir parçacık filtresi oluşturulup, takip edilecek yeni duvarın dış köşe olduğunu belirten bir önem faktörü değeri atanır. Robot yine kendini yeni duvara paralel hale getirmek için dönme hareketi yapar.

Robot harekete başladığı yere döndüğünü anladığında algoritma sonlanır. Tüm işlem adımları bittikten sonra da isteğe göre parçacık filtreleri incelenebilir. Bir parametre filtresinin tahmini tatmin edici ve tutarlı görülmezse bu filtre silinir. Silinen filtreye (landmark) ait noktaların gürültüden kaynaklandığı düşünülür ve onlar da silinir.

6. ÇEVİRİMDİŞİ TOPLANAN VERİ ile HARİTALAMA ve DEĞERLENDİRME

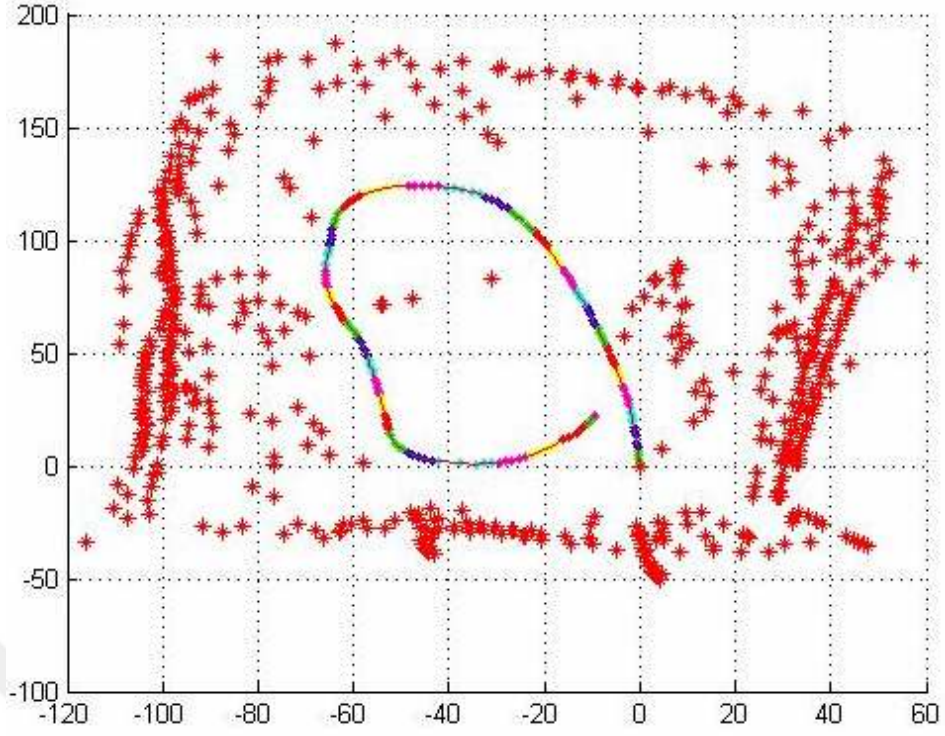
Geliştirilen sistemde eş zamanlı konum belirleme ve harita oluşturma algoritmasına uyarlanan Sıralı Monte Carlo yöntemi gerçek zamanlı bir çözümdür. Ancak gerçek zamanlı uygulamaya geçmeden önce Bölüm 2.4.11’de örneği verilen arayüz kullanılarak robot elle kontrol edilmiş ve çevrimdışı veri toplanmıştır. Toplanan veri bölüm 2.4.6’da anlatıldığı gibi modellenmiş, bölüm 5.4’teki denklemlerle yörünge planlaması yapılmış ve robotun gördüğü engeller haritaya aktarılmıştır.

Buna göre Şekil 6.1’de ters P şeklinde bir yörünge ile ters L şeklindeki bir engelin nasıl haritalandığı görülmektedir. Renkli çizgi adım adım robotun pozisyonunu, kırmızı noktalarda bulunan engelleri göstermektedir. Harekete (0, 0) noktasından başlanmıştır, eksenler cm cinsindendir.



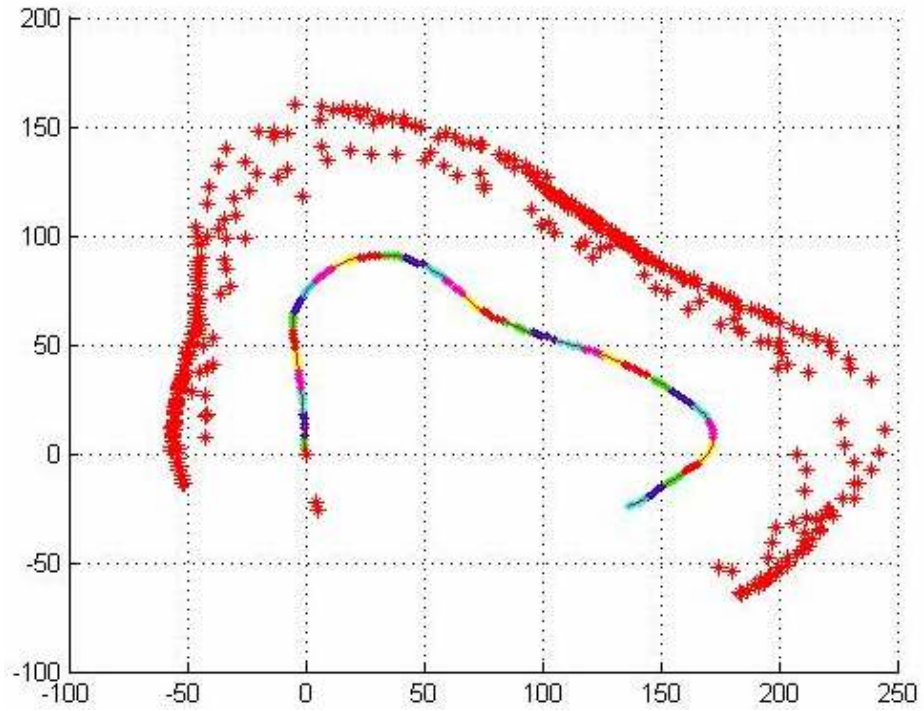
Şekil 6.1 Ters P yörüngesi ile ters L şeklinde bir engelin bulunması

Şekil 6.2’de ise dört tarafı kapatılarak oluşturulan kare şeklindeki kapalı bir mekanda robotun yörüngesi ve çıkardığı harita görülmektedir. Yine renkli çizgi yörüngeyi, kırmızı çizgiler de bulunan engelleri göstermektedir.



Şekil 6.2 Kare şeklinde kapalı bir mekanın haritalanması

Şekil 6.3’de de üç kenarlı düzleme göre eğik duran bir mekanda robotun yörüngesi ve çıkardığı harita görülmektedir. Yine renkli çizgi yörüngeyi, kırmızı çizgiler de bulunan engelleri göstermektedir.



Şekil 6.3 Üç kenarlı ve düzleme eğik olan bir mekanın haritalanması

Çevrimdışı yapılan denemeler sonucunda yörünge'nin gerçek robot hareketiyle çok benzediği, nadiren birkaç yerde birkaç santimetrelik kayma olduğu, onun haricinde oldukça başarılı olduğu gözlenmiştir. Konum belirlemeye bağılı olarak çıkarılan haritaların da gerçek mekana oldukça benzediği görülmüştür.

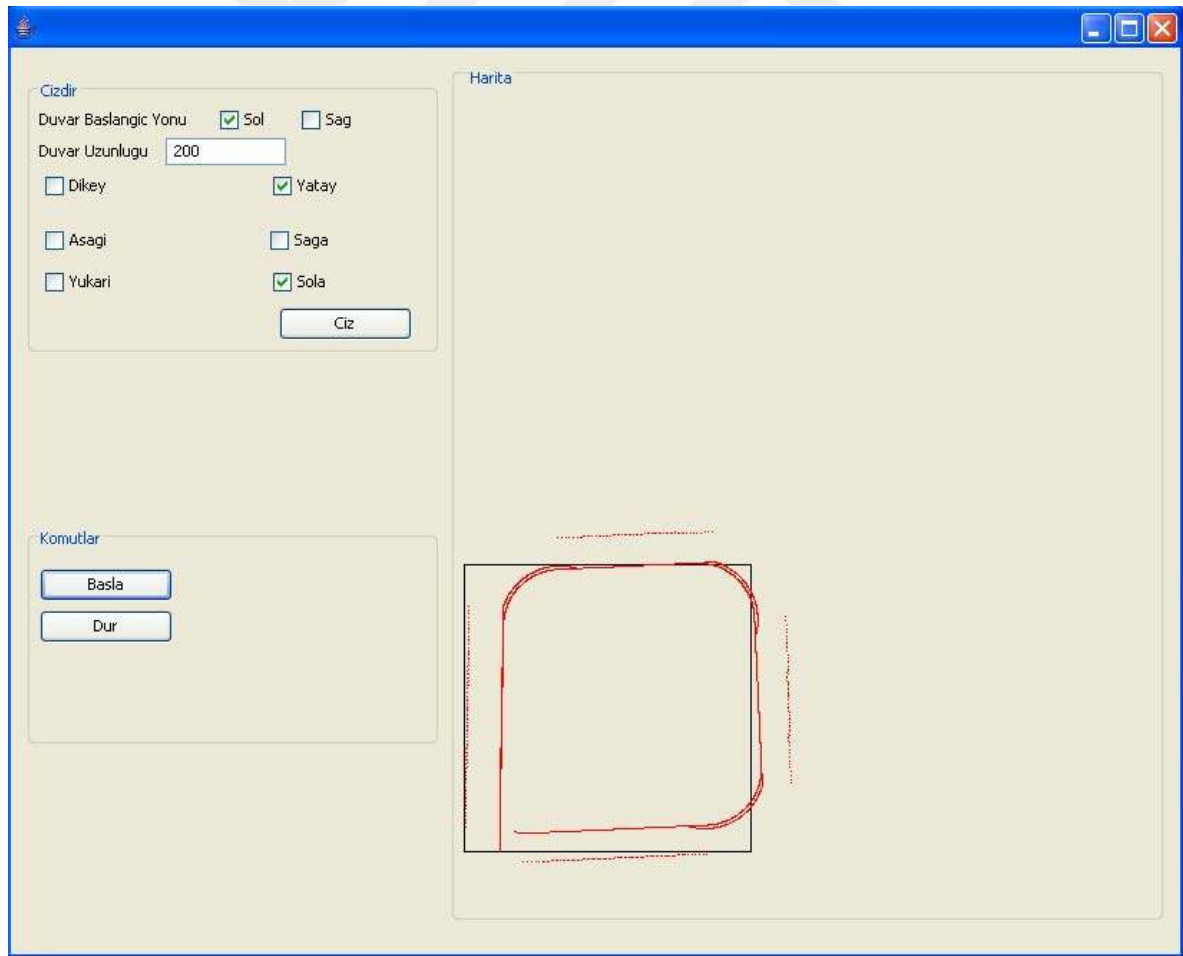


7. SONUÇLAR

Önerilen sistem, ilerde arama kurtarma çalışmalarında en kısa zamanda mümkün olan her yerin aranmasının sağlanması; deprem, yangın gibi insanların çalışmasının ve her yere erişiminin zor olduğu koşullarda canlı tespiti en etkin ve hızlı biçimde tamamlanabilmesi için kullanılabilir. Bu çalışmada elde edilecek yöntemler sadece oda, bina içi gibi küçük mekânlarda değil şehir haritalarının çıkartılmasında da kullanılabilir.

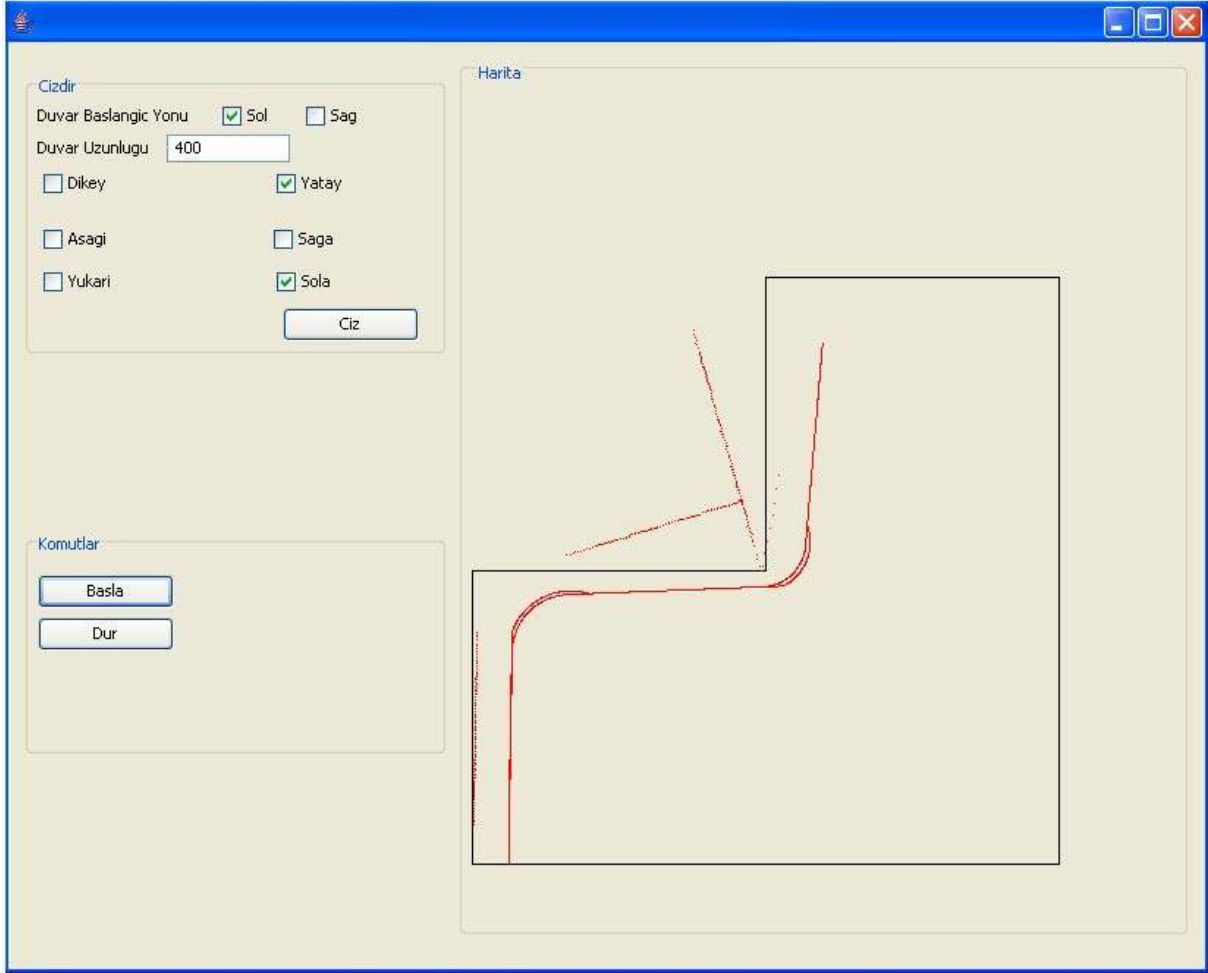
Bu amaçla bu tezde Sıralı Monte Carlo yöntemiyle bir eş zamanlı konum belirleme ve harita oluşturma algoritması önerilmiştir. SMC, gürültünün ne olduğuyla ilgilenmeden gerçek zamanlı çalışarak veri ilişkilendirme problemine çözüm bulabilmektedir. Ayrıca yapılan denemeler sonucunda SMC'nin bellek kullanımı az, uygulaması kolay ve hızlı bir yöntem olduğu görülmüştür.

SMC ile geliştirilen eş zamanlı konum belirleme ve haritalama algoritması çeşitli ortamlarda test edilmiş ve gerçek mekan ile algoritma tarafından çıkarılan harita karşılaştırılmıştır.



Şekil 7.1 Kare şeklindeki kapalı bir mekanın haritalanması

Şekil 7.1’de 200 cm * 200 cm’lik boyutlara sahip kare şeklindeki kapalı bir mekanda gezinilerek elde edilen robot yörüngesi ve gezinilen ortamın haritası görülmektedir. Şekildeki kırmızı eğriler robotun yörüngesini; kırmızı noktalar ise gezinilen mekanın haritasını temsil etmektedir. Ortamda dört adet iç köşe bulunmaktadır. Köşelerin hepsi olması gerektiği gibi tespit edilmiştir.



Şekil 7.2 Altı kenarlı kapalı bir mekanın haritalanması

Şekil 7.2’de ise başlangıç noktası sol alt köşe olan kapalı bir mekanın haritalama sonuçları görülmektedir. Haritası çıkarılacak yeni mekanın başlangıç noktası sol alt köşe olmak üzere kenar uzunlukları sırasıyla 200 cm, 200 cm, 200 cm, 200 cm, 400 cm ve 400 cm’dir. Burada ise sırasıyla bir iç, 1 dış, dört tane iç köşe bulunmaktadır. İlk üç duvarın haritalanması yine kırmızı noktalarla temsil edilmekte; robotun yörüngesi ise kırmızı eğrilerle gösterilmektedir. Görüldüğü gibi ilk iç köşe ve ilk dış köşede bir miktar kayma olmuştur. Bunun sebebi ölçüm bilgisinin bilgisayar tarafından istenildiği an ile haritanın çizildiği an arasında robotun az da olsa hareketine devam etmiş olması ve haberleşmenin gecikmeye sebep olmasıdır. Ancak kaymaya rağmen iki duvarın birbirine dik olduğu tespit edilebilmiştir.

Sistemin sadece o andaki deęil, sonraki anlardaki saklı durumunun yani robot pozisyonu ve engel pozisyonları tahminlerinin de güçlü bir kestirim modeliyle oluşturulmasının kayma sorununu çözebileceęi düşünölmektedir. Gelecekteki çalışmalarda bu amaçla řu anda kullanılan SMC algoritmasının iyileřtirilmesi hedeflenmektedir.



KAYNAKLAR

- Beevers K. (2005), "Loop Closing in Topological Maps", Proceedings of the International Conference on Robotics and Automation, 24:601-613.
- Castellanos, J. A., Neira, J. and Tardós, J.D. Multisensor (2001), "Fusion for Simultaneous Localization and Map Building", IEEE Transaction on Robotics and Automation, 17(6):908-914.
- Chatila, R. and Laumond, J.-P (1985), "Position Referencing and Consistent World Modeling for Mobile Robots", In Proceedings of the IEEE International Conference on Robotics and Automation, 2:138-145.
- Dempster, A.P., Laird, A.N. and Rubin, D.B. (1977), "Maximum likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society, Series B, 39(1):1-38.
- Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F. and Csorba, M. (2006), "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem", NSW 2006, December 2006, Sydney, Australia.
- Doucet, A., Freitas, N.D. and Gordon, N. (2001), Sequential Monte Carlo Methods in Practice, Springer Science+Business Media, Inc., New York.
- Elfes, A. (1989), "Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation", PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University.
- Garulli A., Giannitrapani A., Rossi A., Vicino A., "Simultaneous Localization and Map Building Using Linear Features", European Conference on Mobile Robots 2005, September 7-10, Ancona-Italy.
- Huang, W. H. and Beevers, K. R. (2005), "Topological mapping with sensing-limited robots", Algorithmic Foundations of Robotics VI, 17:235-250.
- Kalman, R. E. (1960), "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME, Journal of Basic Engineering, 82:35-45.
- Kuipers, B. and Byun, Y.-T. (1991), "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations", Journal of Robotics and Autonomous Systems, 8:47-63.
- Matarić, M. J. (1990), A Distributed Model for Mobile Robot Environment-learning and Navigation. Master's thesis, MIT, January, Cambridge, MA. Also available as MIT AI Lab Tech Report AITR-1228.
- Maybeck, P. (1979), Stochastic Models, Estimation, and Control, Academic Press, Inc, New York.
- Moravec, H. P. (1988), "Sensor Fusion in Certainty Grids for Mobile Robots", AI Magazine, 9(2):61-74.

Reuter J. (1998), "Mobile robots trajectories with continuously differentiable curvature: An optimal control approach", IEEE/RSJ Proc. of Int. Conf on Intelligent Robots and System, 1:38-43.

Siegwart, R. ve Nourbakhsh, I.R. (2004), Introduction to Autonomous Mobile Robots, The MIT Pres, Cambridge, Massachusetts.

Smith, R., Self, M. and Cheeseman, P. (1990), Autonomous Robot Vehnicles, Springer-Verlag New York Inc., New York.

Thrun, S., Fox, D. and Burgard, W. (1998), "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots", Machine Learning, 31:29–53. Also appeared in Autonomous Robots 5:253–271.

Thrun, S., Bücken, A., Burgard, W., Fox, D., Fröhlinghaus, T., Hennig, D., Hofmann, T., Krell, M. and Schmidt, T. (2001), "Map Learning and HighSpeed Navigation in RHINO", Research Reports on Information Science and Electrical Engineering of Kyushu University, 6(1):77-82.

Thrun, S. (2002), Robotic mapping: A survey. Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann.

Yuen, D.C.K. and MacDonald, B.A. (2002), "A Comparison between EKF and Sequential Monte Carlo Techniques for Simultaneous Localisation and Map-building", Australian Conference on Robotics and Automation, November 2002, Auckland, New Zealand.

ÖZGEÇMİŞ

Doğum tarihi 10.10.1983

Doğum yeri Antakya

Lise 1998-2000 Isparta Süleyman Demirel Fen Lisesi
2000-2001 Isparta Gazi Lisesi

Lisans 2001-2005 Yıldız Teknik Üniversitesi Elektrik – Elektronik Fak.
Bilgisayar Mühendisliği Bölümü

Çalıştığı kurumlar

08.2005-10.2005 Sesam Yazılım Hizmetleri

2005-Devam ediyor YTÜ Fen Bilimleri Enstitüsü Araştırma Görevlisi