

**ENGELLİ BİR ALAN İÇİNDE OTOMATİK OLARAK HEDEFİNİ
BULABİLEN BİR MOBİL ROBOTUN TASARIMI, İMALATI VE
HAREKET ALGORİTMALARININ GELİŞTİRİLMESİ**

**Pamukkale Üniversitesi
Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
Elektrik-Elektronik Mühendisliği Anabilim Dalı**

Erdem YILDIZ

- 1. Danışman: Yrd. Doç. Dr. Abdullah T. TOLA**
- 2. Danışman: Doç. Dr. E. Şahin ÇONKUR**

**Temmuz, 2009
DENİZLİ**

YÜKSEK LİSANS TEZİ ONAY FORMU

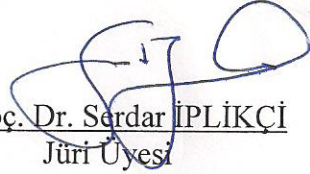
Erdem YILDIZ tarafından Yrd. Doç. Dr. Abdullah T. TOLA yönetiminde hazırlanan **“Engelli Bir Alan İçinde Otomatik Olarak Hedefini Bulabilen Bir Mobil Robotun Tasarımı, İmalatı ve Hareket Algoritmalarının Geliştirilmesi”** başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans Tezi olarak kabul edilmiştir.



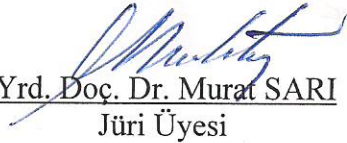
Yrd. Doç. Dr. Abdullah T. TOLA
Jüri Başkanı (1. Danışman)



Doç. Dr. E. Şahin ÇONKUR
Jüri Üyesi (2. Danışman)



Doç. Dr. Serdar İPLİKÇİ
Jüri Üyesi



Yrd. Doç. Dr. Murat SARI
Jüri Üyesi

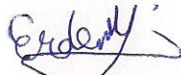


Yrd. Doç. Dr. Sezai TOKAT
Jüri Üyesi

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
29/07/2009 tarih ve 17/18.. sayılı kararıyla onaylanmıştır.

Prof. Dr. Halil KARAHAN
Müdür

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalıřmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalıřmalara atfedildiđini beyan ederim.

İmza : 
Öğrenci Adı Soyadı : Erdem YILDIZ

TEŞEKKÜR

Öncelikle lisans tez danışmanlığımı yaptıktan sonra şimdi de yüksek lisans tez danışmanlığımı yapan Yrd. Doç. Dr. Abdullah T. TOLA hocama bana yaptığı tüm yardımlardan dolayı teşekkürlerimi sunarım.

Robot konularındaki bilgi ve tecrübelerinden faydalanmamı sağlayan, programlama konusunda kendimi geliştirmeme yardımcı olan, sorduğum her soruyu cevaplamış olan, projede olumsuz bir durum oluştuğunda moral vererek performansımın düşmemesini sağlayan ve projenin tıkanan kısımlarında problemlerin çözülmesine yardımcı olan danışmanım Doç. Dr. E. Şahin ÇONKUR hocama teşekkürlerimi sunarım.

Tezle alakalı matematik konularında bana kaynak kitap önerisinde bulunarak doğru kaynağa yönelmemi sağlayan Yrd. Doç. Dr. Murat SARI hocama teşekkürlerimi sunarım.

Mobil robotun tasarımı ve imalatı konularında yardımlarından dolayı makineci arkadaşlarım Caner İNCEOĞLU ve Halil Murat ENGİNSOY'a teşekkürlerimi sunarım.

Mas Otomasyon ve Star Torna çalışanlarına da yardımlarından dolayı teşekkürlerimi sunmak istiyorum.

Teze maddi destek veren Pamukkale Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimine (PAUBAP Proje No: 2007 FBE 011) teşekkürlerimi sunarım.

Son olarak hem maddi hem de manevi destekte bulunan aileme teşekkürlerimi sunmak istiyorum.

Erdem YILDIZ

ÖZET

Yıldız, Erdem

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği ABD

Tez Yöneticileri: Yrd. Doç. Dr. Abdullah T. TOLA, Doç. Dr. E. Şahin ÇONKUR

Temmuz 2009, 116 Sayfa

Mobil robotik konusu günümüzde popüler olan bilimsel çalışma konularından biridir. Pratik olarak günlük hayatta kullanılmaya başlanmışlardır ve gün geçtikçe kabiliyetleri artmaktadır. Mobil robotik, değişik disiplinleri içeren çok sayıda alt konuya sahiptir. Bu konular yol planlama, engelden kaçınma, mobil robot tasarımı ve motor kontrolü vb. şeklindedir. Bu tezde, mobil robotik ile ilgili konulardan endüstriyel kullanımı olabilecek bir mobil robot tasarımı, imalatı ve hareket algoritmalarının geliştirilmesi konuları çalışılmıştır.

Diferansiyel sürüş yapısına sahip boyutları (exdxy) 540x600x550(mm), aks mesafesi 770 mm. olan mobil robotun tasarımı yapılmış, SolidWorks programı yardımıyla katı modeli çıkarılmış ve imalatı gerçekleştirilmiştir. Robotun tahrik mekanizması, iki adet servo motor ve bu motorlara bağlı 1/25 oranında iki adet redüktör ve 430 mm çapında iki adet scooter tekerleğinden oluşmaktadır. Robotun tekerleklerle ilettiği maksimum güç 800 W ve ürettiği maksimum tork 63,5 Nm'dir. Tekerlekler birbirinden bağımsız hareket ederek robotun doğrusal, olduğu yerde dönme veya açılı hareket gibi değişik hareketleri yapabilmesini sağlamaktadır. Robot 3000 VA değerinde UPS güç kaynağı ile beslenmektedir. Robotun kontrol sistemi, robot üzerine yerleştirilen servo motorlar, servo sürücüler, hareket kontrol kartı ve kişisel bilgisayardan oluşmaktadır.

Mobil robotun yazılımı robotun başlangıç ve hedef konumu belli bir ortamda yine konumları belli hareketli veya hareketsiz engeller arasından hedefe ulaşılabilmesi için Visual C# .NET programında geliştirilmiştir. Mobil robotun yol planlamasında potansiyel alan metodundan yararlanılmıştır. Değişik şekillerdeki engeller fare ile çizilebilmektedir. Hedef ve başlangıç konumları kullanıcı tarafından değiştirilebilmektedir. Program sınıflar kullanılarak modüler olarak yazıldığından dolayı hem fonksiyonelliği artmış hem de geliştirilmesi kolaylaşmıştır.

Anahtar Kelimeler: Mobil robot, yol planlaması, potansiyel alan metodu, engellerden kaçınma, hareketli engeller.

Doç. Dr. E. Şahin ÇONKUR

Doç. Dr. Serdar İPLİKÇİ

Yrd. Doç. Dr. Abdullah T. TOLA

Yrd. Doç. Dr. Murat SARI

Yrd. Doç. Dr. Sezai TOKAT

ABSTRACT

Yıldız, Erdem

M. Sc. Thesis in Electrical-Electronics Engineering

Supervisors: Asst. Prof. Dr. Abdullah T. TOLA, Assoc. Prof. Dr. E. Şahin ÇONKUR

July 2009, 116 Pages

Mobile robotic is one of the today's popular scientific research issues. Mobile robots are getting more abilities day by day and are being used more widely in practical life. Mobile robotic has a lot of sub-topics including various disciplines. These topics are path planning, obstacle avoidance, mobile robots design and motor control etc. In this thesis some of the sub-topics such as mobile robot design, manufacturing and motion algorithms that can be useful in industrial applications have been studied.

A mobile robot having a differential drive system with dimensions (wxdxh) 540x600x550(mm) and with axle length 770 mm has been designed using SolidWorks program, produced and controlled. Robots motion mechanism consists of two motors, two reduction gears that have 1/25 ratio and two scooter wheels whose diameter is 430 mm. Robot's maximum power transmitted to the wheels is 800 W and can produce maximum 63.5 Nm torque. The wheels can be controlled independently so the robot can go through or rotate with an angle or rotate on a point. The robot has a 3000 VA UPS power supply. Robot's control mechanism consists of servo motors, servo drivers, a motion control card and a personal computer.

The software program of the mobile robot has been designed using Visual C# .NET programming language to be able to reach its goal in an environment cluttered with moving or static obstacles. Potential field method is used for the path planning. Various types of obstacles can be drawn by the mouse. The user can easily determine obstacles positions, starting and goal position. Visual C# .NET programming language is used for the software program. Since the software is written as a modular program using classes, not only its functionality increases but also its development becomes easier.

Keywords: Mobile Robot, path planning, potential field method, obstacle avoidance, moving obstacles.

Assoc. Prof. Dr. E. Şahin ÇONKUR

Assoc. Prof. Dr. Serdar İPLİKÇİ

Asst. Prof. Dr. Abdullah T. TOLA

Asst. Prof. Dr. Murat SARI

Asst. Prof. Dr. Sezai TOKAT

İÇİNDEKİLER

	Sayfa
TEZ ONAY SAYFASI	i
BİLİMSEL ETİK SAYFASI	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER DİZİNİ	vi
ŞEKİLLER DİZİNİ	viii
TABLolar DİZİNİ	xi
SİMGELER VE KISALTMALAR DİZİNİ	xii
1. GİRİŞ	1
1.1. Literatür Taraması	1
1.1.1. Hareketsiz engellerin olduğu ortamlardaki simülasyon çalışmaları	5
1.1.2. Hareketli engellerin olduğu ortamlardaki simülasyon çalışmaları	7
1.1.3. Hareketsiz engellerin olduğu ortamlardaki uygulama çalışmaları	13
1.1.4. Hareketli engellerin olduğu ortamlardaki uygulama çalışmaları	16
1.1.5. Engelin olmadığı ortamlardaki uygulama çalışmaları	17
1.1.6. Engelin olmadığı ortamlardaki simülasyon çalışmaları	19
1.2. Mobil Robotun Tasarımı ve İmalatı	20
1.3. Mobil Robotun Kontrolü	21
1.4. Tezin Amacı ve Tanıtımı	22
2. MOBİL ROBOTUN DONANIMI	24
2.1. Mobil Robotun Parçaları	24
2.2. Mobil Robotun Sürüş Sistemi	34
3. MOBİL ROBOTUN YAZILIMI	36
3.1. Potansiyel Alan Metodunun Matematiksel Alt Yapısı	36
3.1.1. Kısmi türev tanımı	36
3.1.2. Sonlu fark yöntemi	36
3.1.3. Taylor serisi yaklaşımı ve hatalar	39
3.1.4. Sonlu fark yönteminden potansiyel alan metodunun çıkarılması	41
3.1.5. Potansiyel alanın çalışma mantığı ve hesaplanması	43
3.2. Mobil Robot ile Hedef Arasındaki Yolun Bulunması	46
3.3. Mobil Robot için Yazılan İki Adet Program	47
3.3.1. Program1: Mobil robot için yazılan ilk program	48
3.3.2. Program2: Mobil robot için yazılan ikinci program	56
3.4. Hareket Kontrol Kartı'nın (Motion Control Card) Yazılım ile Kontrolü	84
4. SONUÇ	88
KAYNAKLAR	91
EKLER	97
EK-1 İLERİ, GERİ VE MERKEZİ FARKLAR FORMÜLLERİNİN	

ÇIKARILIŞLARI.....	98
EK-2 MOBİL ROBOTUN DESTEK VE BAĞLANTI PARÇALARININ TEKNİK RESİMLERİ.....	106
EK-3 HAREKET KONTROL KARTINA AİT FONKSİYONLAR.....	112
ÖZGEÇMİŞ	116

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1 a) Elektrik süpürgesi b) Tekerlekli sandalye	1
Şekil 1.2 a) Yardımcı eleman b) Mayın tarama robotu.....	2
Şekil 1.3 a) İnsansız uçak b) Piramit içinin araştırılması.....	2
Şekil 1.4 a) Boru içinin araştırılması b) Deniz altı araştırmaları	2
Şekil 1.5 a) Uzay araştırmaları b) Robot futbolu	2
Şekil 1.6 Görünürlük grafiği	4
Şekil 1.7 Voronoi diyagramı	4
Şekil 1.8 Hücre ayrıştırması.....	4
Şekil 1.9 Potansiyel alanda mobil robotun hedefini bulması.....	5
Şekil 1.10 CAPFM ile mobil robotun hedefine ulaşması	8
Şekil 1.11 Mobil robotun hareketli engel ve hareketli hedefin olduğu durumdaki yolu.....	9
Şekil 1.12 3 mobil robotun öncelik sırasına göre geçişi	10
Şekil 1.13 Mobil robotun hareketli hedefe ulaşmasını gösteren simülasyon.....	12
Şekil 1.14 Mobil robotun hareketli bir engel varken hedefe varmasını gösteren simülasyon.....	12
Şekil 1.15 Müzenin 3 boyutlu gösterimi	13
Şekil 1.16 KURT3D mobil robotun ortamı görüşü ve ortamın geçilebilirliğini belirlemesi	16
Şekil 1.17 WEB-SUNAR.....	19
Şekil 2.1 Profil kafes sistemi.....	25
Şekil 2.2 a) UPS'in fotoğrafı b) UPS'in katı modellemesi	27
Şekil 2.3 a) Servo sürücünün fotoğrafı b) Servo sürücünün katı modellemesi.....	27
Şekil 2.4 a) Servo motorun fotoğrafı b) Servo motorun katı modellemesi.....	28
Şekil 2.5 Kablolama kartının fotoğrafı.....	28
Şekil 2.6 a) 24 volt güç kaynağının fotoğrafı b) 24 volt güç kaynağının katı modellemesi	29
Şekil 2.7 Bilgisayarın anakartı, güç kaynağı ve harddiski	29
Şekil 2.8 a) Redüktörün fotoğrafı b) Redüktörün katı modellemesi	29
Şekil 2.9 Mobil robotun SolidWorks programındaki katı modellemesi (perspektif) .	30
Şekil 2.10 Mobil robotun SolidWorks programındaki katı modellemesi (ön görünüş)	30
Şekil 2.11 Mobil robotun SolidWorks programındaki katı modellemesi (arka görünüş)	31
Şekil 2.12 Mobil robotun SolidWorks programındaki katı modellemesi (üst görünüş).....	31
Şekil 2.13 Mobil robotun fotoğrafı (perspektif).....	32
Şekil 2.14 Mobil robotun fotoğrafı (ön görünüş).....	32
Şekil 2.15 Mobil robotun fotoğrafı (arka görünüş).....	33

Şekil 2.16 Mobil robotun fotoğrafı (üst görünüş)	33
Şekil 2.17 Mobil robotun dönme merkezi.....	35
Şekil 3.1 Sonlu Fark Yöntemlerinin geometrik yorumu	37
Şekil 3.2 İki boyutlu ızgara sistemi.....	39
Şekil 3.3 P noktasının iki boyutlu bir alanda gösterimi	42
Şekil 3.4 Potansiyel alanın üç boyutlu görüntüsü	43
Şekil 3.5 Potansiyel alanın üç boyutlu görüntüsü	44
Şekil 3.6 Çalışma alanının hesabı	45
Şekil 3.7 Mobil robotun yönünün belirlenmesi	46
Şekil 3.8 Mobil robot için yazılan ilk programın ara yüzü (Program1).....	47
Şekil 3.9 Mobil robot için yazılan ikinci programın ara yüzü (Program2).....	48
Şekil 3.10 Program1 menüleri.....	48
Şekil 3.11 File menüsü	49
Şekil 3.12 Change Positions menüsü	49
Şekil 3.13 Obstacles menüsü	49
Şekil 3.14 OptionDialog Formu	50
Şekil 3.15 Alanın yeşil renkteki görüntüsü	51
Şekil 3.16 Geri, İleri ve Select and Pan butonları	52
Şekil 3.17 Fare'nin koordinatlarının gösterimi	52
Şekil 3.18 Obstacles Directions kutusu.....	53
Şekil 3.19 Çeşitli fonksiyonlar	53
Şekil 3.20 Mobil robotun programda çizimi	54
Şekil 3.21 Program1 ile ilgili kayıt defterine yapılan kayıtlar	54
Şekil 3.22 Hedefe yolun olmaması durumu	55
Şekil 3.23 Başlangıç konumunun hedef ile aynı kısma konulduğu durum.....	56
Şekil 3.24 Alanın boyutlarının ayarlanması.....	57
Şekil 3.25 Engellerin çizimi, başlangıç ve hedef konumlarının ayarlanması	57
Şekil 3.26 Form kontrolleri	57
Şekil 3.27 Form kontrolleri	58
Şekil 3.28 Form kontrolleri	59
Şekil 3.29 Form kontrolleri	60
Şekil 3.30 Form kontrolleri	61
Şekil 3.31 Yola ait noktaların koordinatları	61
Şekil 3.32 Koordinatların gösterimi	62
Şekil 3.33 Program2 ile ilgili kayıt defterine yapılan kayıtlar	63
Şekil 3.34 Mobil robotun yolu	64
Şekil 3.35 Mobil robotun yolunun büyültülmüş hali	64
Şekil 3.36 TriangleFunctions, RectangleFunctions, EllipseFunctions ve CircleFunctions sınıfları.....	65
Şekil 3.37 LineFunctions, IsInTriangle, ObstaclesMovements ve DrawGridsClass sınıfları	66
Şekil 3.38 CalculatePath, CalculatePathMethod2, DrawPath ve PathLine sınıfları ...	66
Şekil 3.39 GridsValueToExcel, Sekil, PointsForBeam ve PointsForLeastBeams sınıfları	67
Şekil 3.40 Resources, Program, Settings ve DirectionOfShape sınıfları	67
Şekil 3.41 CalculateField, Regis, Shape2 ve Form1 sınıfları	68
Şekil 3.42 M, Mo ve IO sınıfları	68
Şekil 3.43 Program2'ye ait sınıflar	69
Şekil 3.44 Elipsin parametreleri	70
Şekil 3.45 Çizgi şeklindeki engel.....	71

Şekil 3.46 Yumuşatılmış yol ve orijinal yol	74
Şekil 3.47 10 kez yumuşatılmış yol ve orijinal yol.....	75
Şekil 3.48 Işın analizi yöntemi.....	76
Şekil 3.49 Çeşitli şekillerdeki engellerin olduğu ortamda ışın analizi yöntemi.....	77
Şekil 3.50 Gridlerin durumu	77
Şekil 3.51 En kısa ışınların gösterilmesi.....	78
Şekil 3.52 Hedefin labirentin içinde olduğu durum (Program2)	79
Şekil 3.53 Hedefin labirentin içinde olduğu durum (Program1)	79
Şekil 3.54 Hareketsiz engellerin olduğu bir durumda Program2'den anlık görüntüler	80
Şekil 3.55 Hareketli hedefin olduğu bir durum (başlangıç konumu).....	81
Şekil 3.56 İkinci engel dönülürken programdan bir görüntü	82
Şekil 3.57 Hareketli engelin olduğu durum (başlangıç konumu)	82
Şekil 3.58 Hareketli bir engelin olduğu durumda mobil robotun hareketi ($t = 1$)	83
Şekil 3.59 Hareketli bir engelin olduğu durumda mobil robotun hareketi ($t = 3$)	83

TABLolar DİZİNİ

	Sayfa
Tablo 2.1. UPS'in özellikleri	25
Tablo 3.1. 6x6'lık bir alandaki grid noktalarının sayısal değerleri	45

SİMGELER VE KISALTMALAR DİZİNİ

α	Yön açısı
θ	Açısal hız
ϕ_r	Sağ teker hızı
ϕ_l	Sol teker hızı
Φ	Grid nokta operatörü
∂	Kısmi türev operatörü
Δ	Aynı eksen üzerindeki ardışık iki nokta arasındaki fark operatörü
4D	4 Dimensional (4 Boyutlu)
AHC	Adaptive Heuristic Critic (Adaptif Sezgisel Kritik)
API	Application Programming Interface (Uygulama Programlama Arayüzü)
CAPFM	Chaotic Artificial Potential Field Method (Kaotik Yapay Potansiyel Alan Metodu)
CMOS	Complementary Metal Oxide Semiconductor (Tamamlayıcı Metal Oksit Yarı İletken)
CMP	Compare Equal (Eşitliği Kıyasla)
CVT	Centroidal Voronoi Tessellations (Merkezi Voronoi Mozaığı)
DC	Direct Current (Doğru Akım)
DWENN	Dynamic Wave Expansion Neural Network (Dinamik Dalga Yayılımlı Yapay Sinir Ağı)
ERC	Error Counter (Hata Sayıcı)
eX	Fare'nin x koordinatı
exdxy	enxderinlikxyyükseklik
eY	Fare'nin y koordinatı
FIN	Finish (Bitiş)
Hz	Hertz
ICC	Instantaneous Center of Curvature (Anlık Eğrilik Merkezi)
IHDR	Incremental Hierarchical Discriminant Regression (Artımsal Hiyerarşik Diskriminant Regresyon)
IO	Input Output (Giriş Çıkış)
i, j	İndis numaraları
kg	Kilogram
KVA	Kilo Volt Amper
KW	Kilo Watt
l	aks uzunluğu
L	İlerleme boyu
LCD	Liquid Crystal Display (Sıvı Kristal Gösterge)
LDR	Light Dependent Resistor (Işığa Bağımlı Direnç)
LINQ	Language-Integrated Query (Dil Bütünlemeli Sorgu)
MIMO	Multi Input Multi Output (Çok Giriş Çok Çıkış)

Nm	Newton metre
PIC	Peripheral Interface Controller (Çevresel Arabirim Denetleyicisi)
PVO	Probabilistic Velocity Obstacle (Olasılıksal Engel Hızı)
R_{icc}	Mobil robotun etrafında döndüğü nokta ile aks merkezi arasındaki mesafe
r	Tekerlek yarıçapı
SVON	Servo On (Servo Açık)
$u_{i,j}^{m+1}$	m+1 iterasyon sonrasında oluşan (i, j) konumundaki gridin değeri
UPS	Uninterruptable Power Supply (Kesintisiz Güç Kaynağı)
V_f	Doğrusal hız
VA	Volt Amper
VAC	Volt Alternative Current (Volt Alternatif Akım)
VGM	Visibility Graph Method (Görünebilirlik Grafik Metodu)
W	Watt
xg	Dairenin veya elipsin merkezinin x koordinatı
xm	Elips üzerindeki bir noktanın x koordinatı
xr	Elipsin x eksenindeki yarıçapı
yg	Dairenin veya elipsin merkezinin y koordinatı
ym	Elips üzerindeki bir noktanın y koordinatı
yr	Elipsin y eksenindeki yarıçapı

1. GİRİŞ

1.1. Literatür Taraması

Mobil robotların gün geçtikçe kullanım alanları artmaktadır. Mobil robot teknolojisinin gelişmesi, kullanım oranının artmasında önemli bir faktördür. Mobil robotların kullanım alanları ve çeşitleri aşağıdaki gibi sıralanabilir:

Hastanelerde, havaalanlarında ve evlerde; elektrik süpürgesi (Şekil 1.1a), tekerlekli sandalye (Şekil 1.1b) ve yardımcı eleman (Şekil 1.2a) görevi yapmaktadırlar. Askeri alanlarda mayın tarama robotu (Şekil 1.2b), insansız uçak (Şekil 1.3a) ve asker olarak görev yapabilmektedirler. Bilimsel ve endüstriyel araştırmalarda kullanılmaktadırlar. Bunlara örnek olarak piramit içinin araştırılması (Şekil 1.3b), boru içinin araştırılması (Şekil 1.4a), deniz altı araştırmaları (Şekil 1.4b) ve uzay araştırmaları (Şekil 1.5a) verilebilir. Biyolojik esinlenmeler sonucu oluşturulan mobil robotlar uçan böcekler, örümcekler ve karıncalar olarak sıralanabilir. Ayrıca mobil robotlarla robot futbolu yarışmaları da yapılmaktadır (Şekil 1.5b).



(a)

Şekil 1.1 a) Elektrik süpürgesi
(WEB_1 2007)



(b)

b) Tekerlekli sandalye (WEB_1 2007)



(a)

Şekil 1.2 a) Yardımcı eleman
(WEB_1 2007)



(b)

b) Mayın tarama robotu (WEB_1 2007)



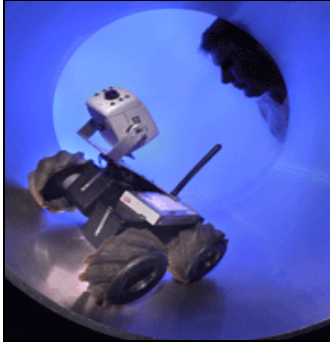
(a)

Şekil 1.3 a) İnsansız uçak
(WEB_1 2007)



(b)

b) Piramit içinin araştırılması (WEB_1 2007)



(a)

Şekil 1.4 a) Boru içinin araştırılması
(WEB_1 2007)



(b)

b) Deniz altı araştırmaları (WEB_1 2007)



(a)

Şekil 1.5 a) Uzay araştırmaları
(WEB_1 2007)



(b)

b) Robot futbolu (WEB_1 2007)

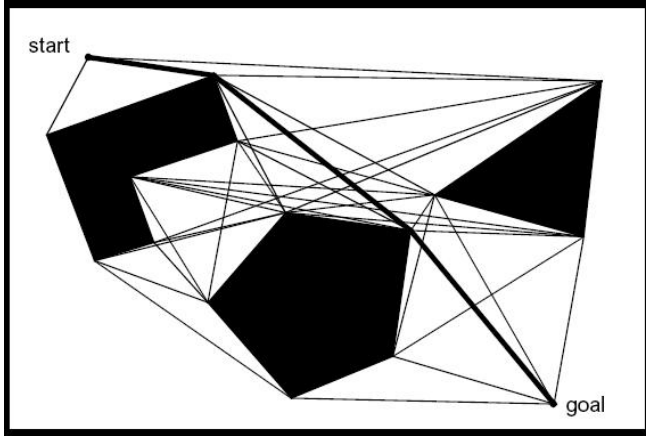
Robot futbolunda karşı takımın oyuncularından kaçabilmek, elektrik süpürgesinde kapalı bir alanın tümünü temizleyebilmek ve insansız uçakların bir kısmında belirli bir hedefe gidebilmek için yol planlaması yapılmaktadır. Diğer mobil robot örneklerinde ise şimdilik elle kontrol tercih edilmektedir. Mobil robotların otonom bir şekilde çalışabilmeleri için yol planlama algoritmalarının geliştirilmesi gerekmektedir. Yol planlama algoritmaları, mobil robotların hedeflerini engeller arasından engellere çarpmadan otomatik olarak bulmasını hedeflemektedir. Ayrıca yol planlama algoritmaları mobil robot araştırmalarında önemli bir yere sahiptirler. Yol planlama algoritmalarını 3 ana gruba ayırabiliriz:

- 1) Yol haritalaması: Alanın engelsiz kısımlarında bir takım yollar oluşturur.
 - (a) Görünürlük Grafiği
 - (b) Voronoi Diyagramı
- 2) Hücre ayrıştırması: Alanı parçalara böler.
- 3) Potansiyel alan: Alan üzerine matematiksel bir fonksiyon uygulanır.

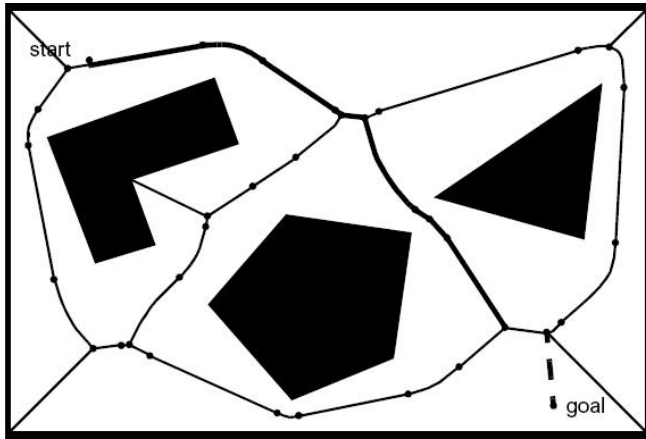
Görünürlük grafiğinde başlangıç noktası, hedef noktası ve engellerin köşe noktaları arasında düz yollar çizilmektedir. Çizilen yolların üstünde engel bulunmamalıdır (Şekil 1.6). Başlangıç noktasından hedef noktasına ulaştıran yollar arasında en kısa olanı en uygun yoldur.

Voronoi diyagramında engellerden eşit uzaklıkta olacak şekilde yollar çizilmektedir ve bu yolların en kısa olanı en uygun yoldur (Şekil 1.7).

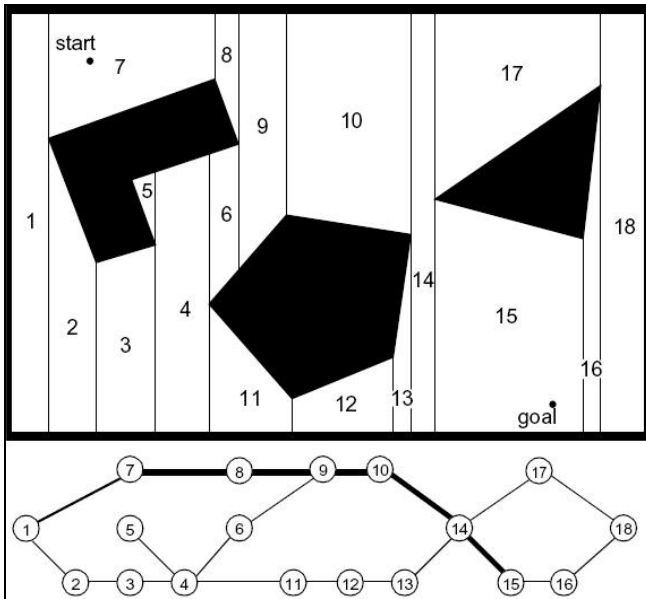
Hücre ayrıştırmasında Şekil 1.8'deki gibi alan hücrelere ayrıştırılır ve mobil robot komşu hücreler üzerinden giderek hedefe ulaşmaktadır. Burada mobil robotun hücrenin neresinden geçeceği önemli değildir.



Şekil 1.6 Görünürlük grafiği (Siegwart ve Nourbaksh, 2004)

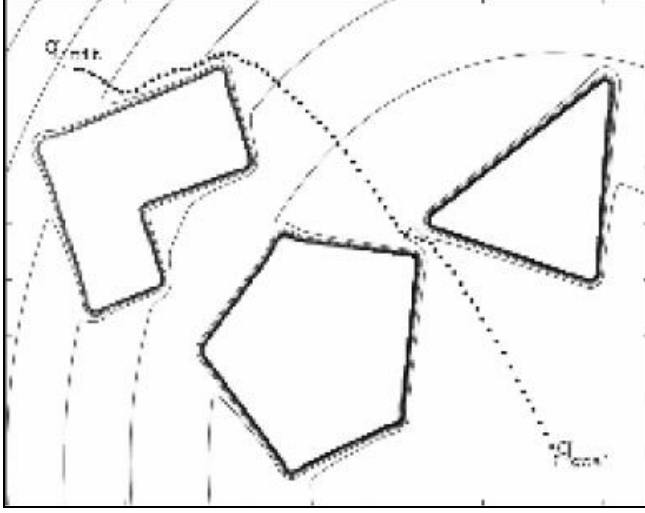


Şekil 1.7 Voronoi diyagramı (Siegwart ve Nourbaksh, 2004)



Şekil 1.8 Hücre ayrıştırması (Siegwart ve Nourbaksh, 2004)

İlk olarak Khatib (1986) tarafından ortaya konulan potansiyel alan metodunda hedef mobil robot üzerinde çekme etkisi yaratırken engeller itme etkisi yaratmaktadır. Bu iki etkinin sonucu olarak robot engellerden kaçınırken hedefe yönelmektedir (Şekil 1.9). Potansiyel alan metodu ile ilgili daha fazla bilgi bölüm 3.1’de anlatılacaktır.



Şekil 1.9 Potansiyel alanda mobil robotun hedefini bulması (Siegwart ve Nourbaksh, 2004)

Mobil robotlarla ilgili literatür çalışmaları incelenirken konunun popülerliği bir kez daha görülmüştür. Kitaplar, yurtiçi ve yurtdışındaki yüksek lisans ve doktora tezleri, makaleler ve sunumlar olmak üzere mobil robotlarla ilgili yaklaşık 1300 belge toplanmıştır. Toplanan her belge bu tezin içeriğine konulmamıştır. Aşağıda mobil robotik konusunda bu tezdeki çalışmalara yakın olan çalışmalar hakkında özetler verilecektir.

1.1.1. Hareketsiz engellerin olduğu ortamlardaki simülasyon çalışmaları

Bekmen (2006) tarafından 2 sürücü tekerlekli holonomik (herhangi bir yönde dönme ihtiyacı duymadan hareket etme) bir mobil robot için simülasyon programı geliştirilmiştir. Mobil robotta bulunan sensörler sayesinde engellere çarpmadan hareket sağlanmıştır. Koordinasyon amaçlı yapay sinir ağları kullanılmış, davranış modülleri için ise bulanık kontrol kullanılmıştır. Örneğin sensörlerden gelen bilgilere göre tekerleklerin hızlarının ayarlanması bulanık kontrol ile sağlanmıştır.

Çonkur (1997) tarafından gereğinden çok serbestlik dereceli robot kolları için yörünge planlaması üzerine çalışılmıştır. Yörünge planlamasında potansiyel alan metodundan yararlanılmıştır. Bu çalışmada esas olarak robot kolları üzerine çalışılmıştır. Ancak link sayısı bire düşürüldüğünde aynı zamanda mobil robot kontrolü de yapılabilmektedir.

Gümrükçü (2003) tarafından mobil robotun kapalı bir ortamda harita çıkarması NOMAD200 simülasyon programı ile gerçekleştirilmiştir. Haritası çıkarılan ortamda herhangi iki nokta arasında en kısa mesafe belirlenebilmiştir. Harita çıkarmada Genelleştirilmiş Voronoi Grafiğinden yararlanılmıştır. Mobil robotun tekerleğinin kaymasından dolayı oluşabilecek hatalar, haritası çıkarılmış ortamdaki bazı referans noktaları ile giderilmiştir.

Gündoğdu (2006) tarafından mobil robotun ultrasonik sensörler ile hedefine ulaşması üzerine çalışılmıştır. Harita grid, quadtree ve framed-quadtree yöntemleriyle, yol ise A* ve D* algoritmalarıyla oluşturulmuştur. Bu yöntemler birbirleriyle karşılaştırılmıştır. MATLAB programı ile simülasyonlar gerçekleştirilmiştir. İleride gerçek bir mobil robot üzerinde yapılan çalışmaların denenebileceği belirtilmiştir.

Hwang ve Ming (2002) tarafından 3 boyutlu alanda mobil robotun engellere çarpmadan hedefe ulaşması üzerine çalışılmıştır. Matematiksel gösterimi basitleştirmek için, çarpışmanın olup olmayacağını anlamak için ve yapılan hesaplamayı azaltmak için engeller elipsoit olarak seçilmiştir. Mobil robotun hedefe ulaşması için 11 adet bulanık model kullanan öğrenme algoritması kullanılmıştır. Mobil robot belli sayılarda denemeler yaptıktan sonra hedefe giden yolu bulabilmektedir.

Saraç (2002) tarafından davranış tabanlı mobil robot kontrolü üzerine çalışılmıştır. Webots simülatörü üzerinde Khepera mobil robotu ile çalışılmıştır. Khepera mobil robotu 5 cm çapındadır ve birbirinden bağımsız kontrol edilebilen 2 adet tekerleğe sahiptir. Mobil robotun engellerden kaçınarak enerji kaynağına ulaşması amaçlanmıştır. Burada enerji kaynağı olarak ışık seçilmiştir. Ayrıca AHC (Adaptive Heuristic Critic) öğrenmesi ile Q-öğrenmesi karşılaştırılmış ve AHC öğrenmesinin Q-öğrenmesine göre genel olarak daha iyi performans gösterdiği belirtilmiştir.

Şirin (2005) tarafından holonomik olmayan bir mobil robotta davranış temelli bulanık kontrol uygulanmıştır. Çalışmalar MATLAB programında yazılıp simüle edilmiştir. Sonar algılayıcılardan alındığı düşünülen veriler üzerinde bulanık mantık kullanılmıştır. Holonomik olmayan mobil robot için engelden sakın, hedefe git, duvarı izle, yola teğet ilerle, avare gez davranışları oluşturulmuştur.

Velagic vd (2006) tarafından 3 seviyeli mobil robot navigasyon sistemi üzerine çalışılmıştır. Alt seviyede mobil robotun lineer ve açısal hızlarıyla ilgilenilmiştir. Orta seviyede mobil robotun pozisyon kontrolüyle ilgilenilmiştir. Üst seviyede ise sensör verilerinin yorumlanması, harita oluşturma ve yol planlaması ile ilgilenilmiştir. Bulanık mantık ve Dempster-Shafer delil teorisi ve işgal gridleri yöntemleri harita oluşturmada kullanılmıştır. Ayrıca yol planlamasında modifiye edilmiş bir potansiyel alan metodu kullanılmıştır.

Yusufoğlu (2002) tarafından nesne sıralama oyununda mobil robot davranışlarının birleştirilmesi üzerine çalışılmıştır. Mobil robotun cisme yaklaşması ve cismi alıp taşınması için gerekli olan ayrı ayrı potansiyel fonksiyonları bu çalışmada tek bir yapay potansiyel fonksiyonu ile gerçekleştirilmiştir. Simülasyonlar Sun Spare Station 5 programında yapılmıştır ve potansiyel alan grafiklerinin çiziminde MATLAB programından yararlanılmıştır.

1.1.2. Hareketli engellerin olduğu ortamlardaki simülasyon çalışmaları

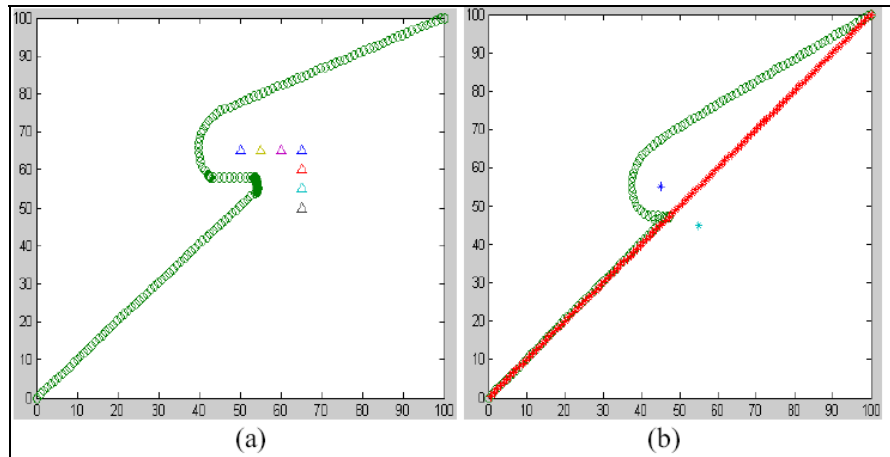
Bicho vd (2005) tarafından motorlu balon için otomatik uçuş güzergahı oluşturma üzerine çalışılmıştır. MATLAB programında simülasyonlar yapılmıştır. Simülasyonlarda CMOS kamerası ile engellerin hangi yönde ve uzaklıkta olduğunun öğrenilebildiği varsayılmıştır. Engellerin durağan veya hareketli olabileceği belirtilmiştir. Yapılan çalışmanın ileride gerçek bir motorlu balon üzerinde uygulanacağı belirtilmiştir.

Jing (2005) tarafından mobil robotlarda davranış dinamikleri ile engellerden kaçınarak hedefe ulaşma çalışılmıştır. Sadece durağan engellerin olduğu ortamlarda, sadece hareketli engellerin olduğu ortamlarda ve hem durağan hem de hareketli engellerin olduğu ortamlarda mobil robotun hedefine ulaşabildiği belirtilmiştir.

Holonomik olmayan bir mobil robot ile çalışmaların yapıldığı kabul edilmiştir. Engellerin boyut ve hız bilgilerinin gerekli olmadığı belirtilmiştir.

Malhotra ve Sarkar (2005) tarafından sabit veya hareketli engellerin bulunduğu bir ortamda mobil robotun hedefine ulaşması üzerine çalışılmıştır. Yol planlamasında potansiyel alan metodundan faydalanılmıştır. Ayrıca mobil robotun davranışlarının belirlenmesinde bulanık mantıktan yararlanılmıştır. Bulanık sistemi MIMO (Multi Input Multi Output-Çok Giriş Çok Çıkış) şeklinde ayarlanmıştır. Bulanık sistemde 37 kural kullanılmıştır. Örneğin, bir kuralda hedef ve engeller mobil robota çok uzak ise yüksek hızda düz bir şekilde gitmesi belirtilmiştir.

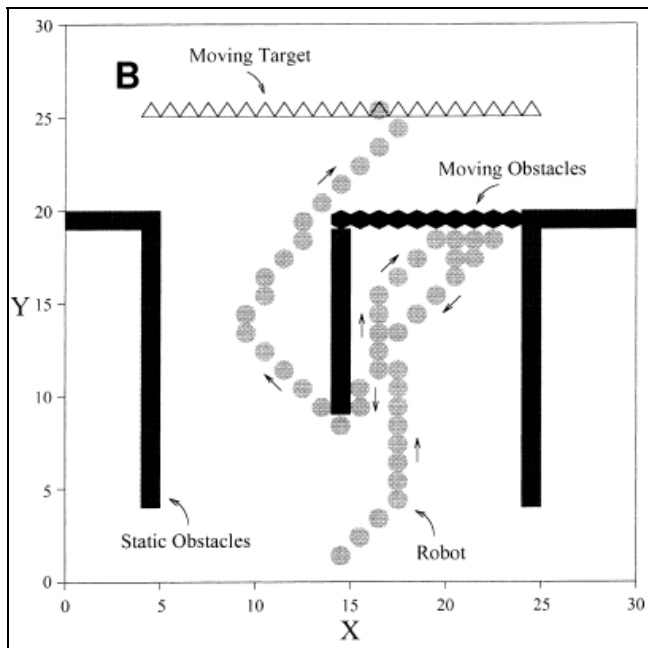
Shi vd (2006) tarafından mobil robotlara yol planlamasında uygulanabilecek kaotik potansiyel alan metodu olarak adlandırılan yeni bir optimizasyon metodu geliştirilmiştir. Geleneksel olarak bilinen potansiyel alan metodundan farklı olarak hareketli engellerden yerel olarak kaçınmada daha uygun yol bulunduğu söylenmiştir. Ayrıca bu yöntemin robot futbolu simülasyon programında da denendiği belirtilmiştir. Şekil 1.10a'da mobil robotun (0,0) noktasından (100,100) noktasına CAPFM ile bulunan yol gösterilmiştir. Burada engeller hareketsizdirler ve konkav şekindedirler. Şekil 1.10b'de (100,100) noktasından (0,0) noktasına doğru bir hareketli engelin geldiği durum gösterilmiştir.



Şekil 1.10 CAPFM ile mobil robotun hedefine ulaşması (Shi vd, 2006)

Yang ve Meng (2000) tarafından yapay sinir ağı ile hareketli ve hareketsiz engellerin olduğu bir ortamda mobil robotun hedefine ulaşması üzerine çalışılmıştır.

Yapılan çalışmada öğrenmeye ihtiyaç olmadığı ve hareketli engeller ile ilgili önceden bir bilgiye de gerek olmadığı belirtilmiştir. Şekil 1.11’de hedef sağa ve sola doğru 10 blok/dk hızda hareket etmektedir. Şekilden de görüldüğü gibi mobil robotun hedefe ulaşabilmesi için 2 adet ana yolu vardır. İlk başta hareketli engel sol yolu tamamen kapatmaktadır ve burada 0,5 dakika durmaktadır. Daha sonra 20 blok/dk hızla sağa doğru hareket etmektedir ve sağdaki yolu tamamen kapatmaktadır. Mobil robot da 20 blok/dk hızla hareket etmektedir ve hareketli engelin sağ yolu kapatmasından dolayı geriye dönüp sol yoldan hedefine ulaşmaktadır.



Şekil 1.11 Mobil robotun hareketli engel ve hareketli hedefin olduğu durumdaki yolu (Yang ve Meng, 2000)

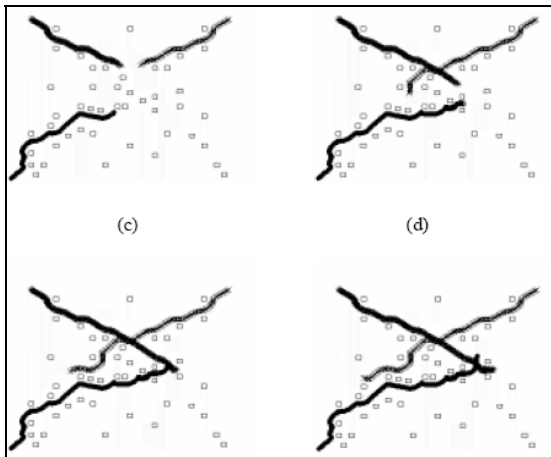
Yannier (2002) tarafından bir veya birden çok mobil robotun engellerden sakınarak hedefe ulaşması üzerine çalışılmıştır. Bu çalışmada potansiyel alan metodundan yararlanılmıştır. Engellerin oluşturduğu itici kuvvet ile hedefin oluşturduğu çekici kuvvetin ayrı ayrı işlendiği belirtilmiştir.

Agirrebeitia vd (2005) tarafından yeni bir yapay potansiyel alan metodu geliştirildiği bahsedilmiştir. Uzaklığın N . üssü ile ters orantılı bir potansiyel yoğunluk seçildiği belirtilmiştir. Daha sonra yol planlamasında yarı-jeodezik metodu (quasi-geodesic method) olarak adlandırdıkları metot ile engellerden kaçınıldığı

açıklanmıştır. Bu metot ile mobil robotun hem hareketsiz hem de hareketli engellerden kaçınabildiği belirtilmiştir.

Wuwei vd (2004) tarafından mobil robotlar için yeni bir navigasyon metodu önerildiği belirtilmiştir. Bu metotta bir eğri üzerinde noktalar araştırılmıştır. Eğri olarak mobil robotun ön tarafındaki 180°'lik kısım ve arka tarafındaki 180°'lik kısım ayrı ayrı alınmıştır. Bulanık yapay sinir ağı ile eğri üzerindeki hangi noktanın seçileceği belirlenmiştir. Hareketli veya hareketsiz engellerin olduğu bir ortamda mobil robotun hedefe ulaşabildiği belirtilmiştir.

Zheng ve Zhao (2006) tarafından yeni bir yapay potansiyel alan metodu tabanlı bir metot geliştirildiği belirtilmiştir. Birden fazla mobil robot karşılaştınca öncelik sırasına göre geçiş yapılmıştır. Önceliği fazla olandan az olana göre bir geçiş sırası takip edilmiştir. Eğer aynı öncelik durumu olursa rastgele bir öncelik sırası oluşturularak geçiş yapılmıştır. Öncelik durumunda görev önceliği, hız ve boyut rol oynamıştır. Şekil 1.12'de sol üst köşeden başlayan 2 numaralı, sağ üst köşeden başlayan 3 numaralı ve sol alt köşeden başlayan mobil robot 1 numaralı mobil robottur. 2 numaralı mobil robot 1 numaralı mobil robot ile aynı görev önceliğine sahiptir ve 2 numaralı mobil robot 3 numaralı mobil robot ile aynı büyüklüğe sahiptir. 1 numaralı mobil robot 3 numaralı mobil robota nazaran daha küçüktür ama görev önceliği daha yüksektir. 2 ve 3 numaralı mobil robotlar karşılaştıklarında 2 numaralı mobil robot görev önceliğinden dolayı önce geçmektedir. 1 ve 2 numaralı mobil robotlar karşılaştıklarında 2 numaralı mobil robot daha büyük olduğundan önce geçmektedir.



Şekil 1.12 3 mobil robotun öncelik sırasına göre geçişi (Zheng ve Zhao, 2006)

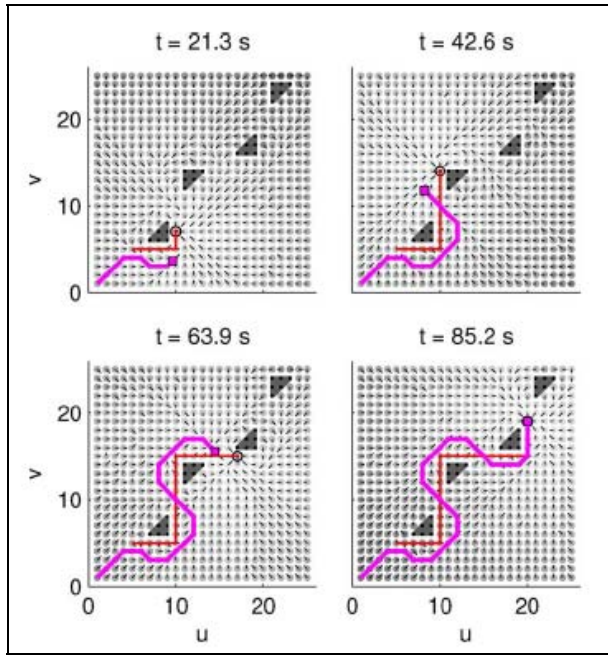
Zhao vd (2007) tarafından öğrenme ile hareketli engellerden kaçınma üzerine çalışılmıştır. Artımsal Hiyerarşik Diskriminant Regresyon (Incremental Hierarchical Discriminant Regression(IHDR)) olarak adlandırılan yöntem ile öğrenme sağlanmıştır. Mobil robota 16 senaryoda öğrenme yapıldığı ve bunun da 1451 örnek kazandırdığı vurgulanmıştır. 30 deneme boyunca %90 başarı sağlandığı belirtilmiştir.

Ya-Chun ve Yamamoto (2006) tarafından hareketli veya hareketsiz engellerin olduğu bir ortamda mobil robotun hedefine ulaşması üzerine çalışılmıştır. Potansiyel alan metodu tabanlı çarpışmadan kaçınma algoritması oluşturulmuştur. Konkav ortamlarda hedefe ulaşamama durumundan dolayı Azaltılmış Genelleştirilmiş Voronoi Grafiği (Reduced Generalized Voronoi Graph) yönteminden yararlanılmıştır. Bu yöntem ile alt hedef oluşturulmuş ve mobil robot bu alt hedefe gittikten sonra ana hedefe yönlendirilerek hedefe ulaşamama durumundan kurtarılmıştır.

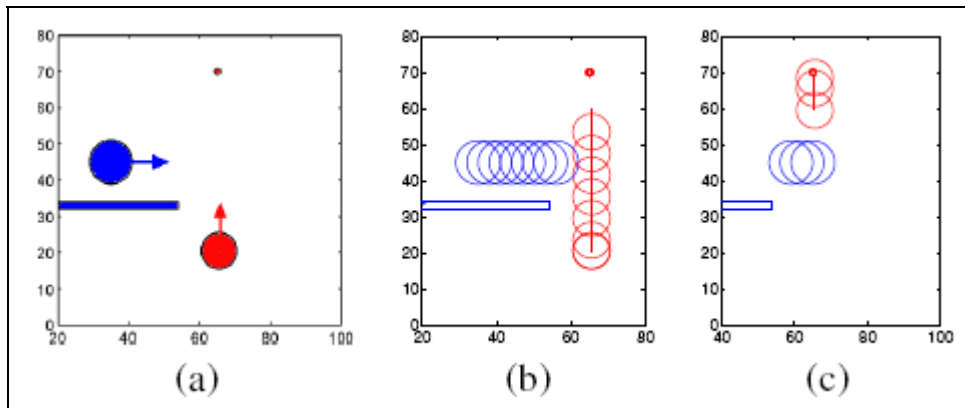
Willms ve Yang (2008) tarafından D* algoritmasına benzer fakat daha verimli olduğu belirtilen bir algoritma önerilmiştir. Bu yöntem ile hareketli veya hareketsiz engellerin olduğu bir ortamda hareketli veya hareketsiz hedefe ulaşılabilirdiği belirtilmiştir. Şekil 1.13'te hedef 0,35 grid/s hızla hareket etmektedir ve hareketine (5,5) noktasından başlamaktadır. Mobil robot ise 0,5 grid/s hızla hareket etmektedir ve hareketine (1,1) noktasından başlamaktadır. Mobil robot hedefi (20,19) noktasında yakalamıştır.

Lebedev vd (2005) tarafından Dinamik Dalga Yayılımlı Yapay Sinir Ağı (Dynamic Wave Expansion Neural Network(DWENN)) olarak isimlendirdikleri hareketli engellerin olduğu bir ortam için yeni bir yapay sinir ağı metodu geliştirildiği belirtilmiştir.

Fulgenzi vd (2007) tarafından Olasılıksal Engel Hızı (Probabilistic Velocity Obstacle (PVO)) ve işgal gridleri yöntemiyle mobil robotlar için hareketli engellerden kaçınma üzerine çalışılmıştır. Şekil 1.14'te kırmızı renkle mobil robot ve mavi renkle hareketli engel gösterilmiştir. Mobil robot hızını maksimum hıza arttırarak engele çarpmadan hedefe varmıştır.



Şekil 1.13 Mobil robotun hareketli hedefe ulaşmasını gösteren simülasyon (Willms ve Yang, 2008)

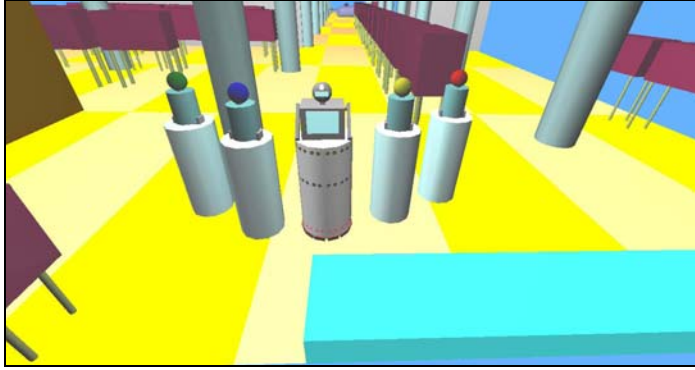


Şekil 1.14 Mobil robotun hareketli bir engel varken hedefe varmasını gösteren simülasyon (Fulgenzi vd, 2007)

Chen vd (2006) tarafından 2 boyutlu zehirli bir gazın yayılımını en aza indirecek şekilde mobil robotların kontrolü üzerine çalışılmıştır. Merkezi Voronoi Mozaïği (Centroidal Voronoi Tessellations(CVT)) yöntemi ile optimal yol planlaması çıkarılmıştır. Hareketli engellerden kaçınmak için de potansiyel alan metodundan yararlanılmıştır.

Sarı (2006) tarafından müzede tur rehberi olarak kullanılacak bir mobil robot tasarımı ve benzetimi üzerine çalışılmıştır. Mobil robot diferansiyel sürürlü olarak

tasarlanmıştır. Müze olarak Rahmi Koç müzesi ele alınmıştır. Bulanık kontrol ile mobil robot kontrol edilmiştir. Davranış olarak 2 temel davranış belirlenmiştir. Biri hedefe gitme davranışdır diğeri ise engelden kaçınma davranışdır. Şekil 1.15'te müzenin 3 boyutlu tasarımı gösterilmiştir.



Şekil 1.15 Müzenin 3 boyutlu gösterimi (Sarı, 2006)

1.1.3. Hareketsiz engellerin olduğu ortamlardaki uygulama çalışmaları

Ateş (2004) tarafından 6 bacaklı bir mobil robot tasarlanıp, gerçekleştirilmesi yapılmıştır. Bacakların hareketi için 2 adet adım motoru kullanılmıştır. Bu adım motorlarını kontrol edebilmek için bir adet mikrodenetleyici kullanılmıştır. Engelleri algılayabilmek için kızıl ötesi engel algılayıcı kullanılmıştır. Bu engel algılayıcı, DC motor ve mikrodenetleyici ile sağa veya sola döndürebilme sağlanmıştır.

Aydın (2003) tarafından global-zaman optimal yörünge bulma üzerine çalışılmıştır. Sabit engellerin olduğu kabul edilerek çalışmalar yapılmıştır. Ayrıca tüm alanın bilgisinin bilindiği varsayılmıştır. Görünebilirlik grafik metoduyla (VGM) yol planlanılmış ve planlanan yol üzerinde optimal yörünge bulma çalışması yapılmıştır. Yolun düz ve eğri kısımlardan oluşacağı belirtilmiş ve eğri kısımlar yapay sinir ağları/bulanık mantık yöntemleriyle oluşturulmuştur.

Bayram (2006) tarafından 4 tekerlekli bir mobil robot tasarlanıp gerçekleştirilmiştir. 2 adet adım motoru ve 4 adet ultrasonik mesafe algılayıcısı kullanılmıştır. Motorlardan biri ile ileri geri hareketi sağlanırken diğeri ile sağ sol hareketi sağlanmıştır. Ultrasonik mesafe algılayıcıları sayesinde engellere çarpmadan hareket ettirilmiştir. Mobil robottaki tüm kontroller için PIC 16F877 kullanılmıştır.

Erođlu (2006) tarafından gezgin robotlar için ultrasonik mesafe algılayıcılarıyla robot davranışlarının kontrolü ve çevre haritalaması üzerine çalışılmıştır. MobilSim simülatörü ve Pioneer 3-dx robotu ile testler yapılmıştır. Bayes güncellemeli doluluk ızgara yöntemiyle çevre haritası oluşturulmuştur. Duvar takibi ve engelden kaçınma davranışları geliştirilmiştir. Pusula bilgisinin sağlıklı alınamamasından dolayı harita çıkarmada kaymalar olduğu belirtilmiştir.

Karaman (2004) tarafından mobil robotlarda çevre haritalama ve yol planlaması üzerine çalışılmıştır. Çevre haritalamada ultrasonik sensörlerden faydalanılmıştır. Ultrasonik sensörden gelen veriler bulanık mantık ve Dempster Shafer yöntemleri ile işlenilmiştir. Yol planlamasında ise A* yönteminden faydalanılmıştır. Çalışmalar NOMAD200 mobil robotu üzerinde denenmiştir.

Küçükceylan (2007) tarafından mobil robotların engelli bir alanda hedefine ulaşması problemi üzerine çalışılmıştır. Mobil robot donanımı için Lego MINDSTORM RIS seti kullanılmıştır. Bu setten çıkan parçalar birleştirilerek çeşitli yapılarda mobil robotlar yapılmıştır. Engellerin tanınması için bütün alanı gören bir tepe kamerası kullanılmıştır. Engeller siyah renkte ve engel olmayan alanlar beyaz renkte olacak şekilde alan ayarlanmıştır. Engellerin yerleri kamera sayesinde alındıktan sonra enine tarama algoritmasından yararlanılarak hedefe olan yol bulunmuştur.

Özcan (2005) tarafından kapalı bir mekanda taranmadık yer kalmayana kadar gezinebilen bir mobil robot üzerine çalışılmıştır. Engellerin tanınmasında mobil robotun etrafına yerleştirdiđi 8 adet fotoselden yararlanılmıştır. Mobil robotun hareketi için 2 adet adım motoru kullanılmıştır. Ayrıca bu adım motorlarının ne kadar döndüğünü takip etmek için sayıcı sensörler kullanılmıştır. Mobil robotta karar mekanizması olarak PIC16F877 kullanılmıştır. Mobil robotun engel karşısında nasıl davranacağını yapay sinir ađı ile belirlenmiştir. Yapay sinir ađında 8 adet giriş (8 adet sensör vardır), 2 adet çıkış (2 adet motor vardır) kullanılmıştır. EasyNN-Plus ve JavaNNS programları ile yapay sinir ađı modeli oluşturulup elde edilen katsayılar mobil robot programında kullanılmıştır.

Özkan (2007) tarafından farklı özelliklere sahip gezgin robot gruplarının dağıtık kontrolü ile ilgilenilmiştir. Uygulamalarında 3 adet Pioneer-3DX mobil robotunu

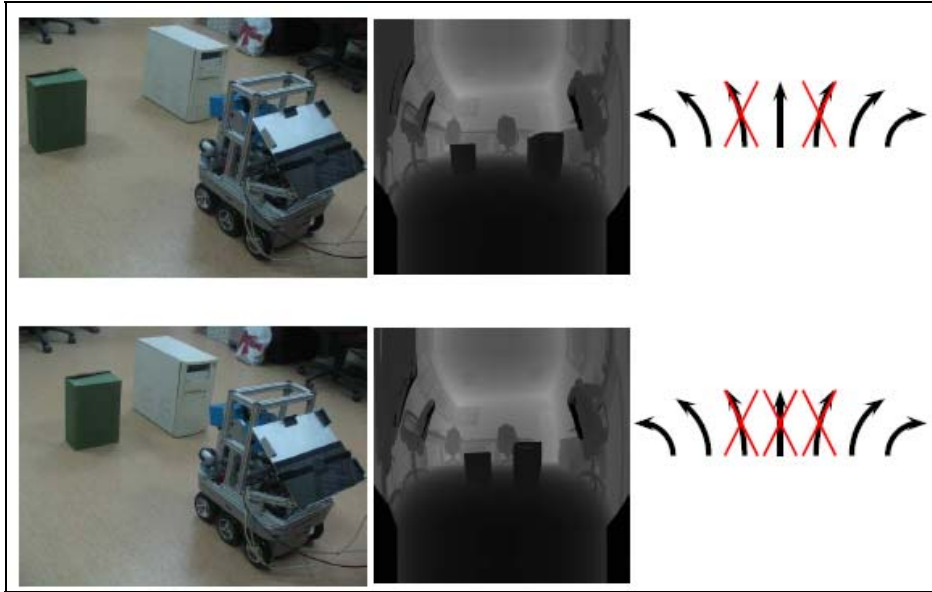
kullanmıştır. Buradaki 3 mobil robot da nesne taşıma amacıyla tutma mekanizmasına sahiptir. Mobil robotlardan bir tanesi hassas mesafe ölçümü için lazer duyucusu ile donatılmıştır. Mobil robotların konumlandırılması için tepe kamerası kullanılmıştır. Uygulamalardan bir tanesinde 3 farklı yapıdaki nesnenin 3 mobil robot tarafından belli bir konuma götürülmesi amaçlanılmıştır. Nesnelerin bir tanesi yapısı nedeniyle 2 mobil robot tarafından ittirilebilmektedir. Başka bir nesne ise tek mobil robot tarafından ittirilebilmektedir. Son nesne ise sadece tutup taşınabilmektedir. Böyle bir uygulamada mobil robotlara amaç belirtilmiştir ve hangi robotun ne iş yapacağını kararı mobil robotlara bırakılmıştır.

Şengül (2006) tarafından mobil robotlarda lazer mesafe ölçümü ile harita çıkarımı üzerine çalışılmıştır. Kapalı bir ortamda lazer sensörü yardımıyla harita çıkarılmasında işgal gridleri yönteminden faydalanılmıştır. İşgal gridi durum olasılıklarını belirlemek için Bayes Tahmin Prosedüründen yararlanılmıştır. Lazer sensörü olarak SICK LMS200 modeli kullanılmıştır.

Uğur (2006) tarafından bir mobil robot için elde edilen uzaklık görüntüleri ile ortamın geçilebilirliği üzerine çalışılmıştır. Uzaklık görüntüleri 3 boyutlu lazer tarayıcı ile alınmıştır. Lazer tarayıcısı olarak SICK LMS 200 modeli kullanılmıştır. Mobil robot olarak KURT3D modeli kullanılmıştır. Simülasyonlar için de KURT3D mobil robotlarına özel bir simülasyon programı olan MACSim simülasyon programı kullanılmıştır. Şekil 1.16'da mobil robotun ortamı nasıl gördüğü ve bu görüntüye göre ortamın geçilebilirliği ortaya koyulmuştur.

Yalçın (2004) tarafından ışığa bağımlı dirençlerden (LDR) alınan veriler yapay sinir ağında kullanılmıştır ve mobil robot için yön bilgisi elde edilmiştir. Mobil robot ışığa doğru yönlendirilmiştir. C++ Builder programı ile yazılmış arayüz ile LDR'lerin durumları ve mobil robotun yön bilgisi gösterilmiştir.

Yalçın (2003) tarafından engellerden kaçınan bir mobil robot üzerine çalışılmıştır. Mobil robotun ana kontrol mekanizması olarak PIC16F877 mikrodenetleyicisi seçilmiştir. Engellerden kaçınmak için kızılötesi sensörlerden faydalanılmıştır. Ayrıca mobil robotun kullanıcıya bilgi vermesi amacıyla LCD kullanılmıştır.



Şekil 1.16 KURT3D mobil robotun ortamı görüşü ve ortamın geçilebilirliğini belirlemesi (Uğur, 2006)

1.1.4. Hareketli engellerin olduğu ortamlardaki uygulama çalışmaları

Baştan (2004) tarafından 2 tekerlekli bir mobil robot için tepe kamerasından alınan bilgiler ile hedefe ulaşma çalışılmıştır. Yolun oluşturulmasında potansiyel alan metodundan faydalanılmıştır. Robot futbollarındaki gibi bir sistem üzerine çalışılmıştır. Tepe kamerasından alınan veriler bir bilgisayarda işlendikten sonra motorlar için gerekli olan bilgiler kablosuz haberleşme ile mobil robota gönderilmiştir.

Guleç (2005) tarafından bir grup mobil robotun koordineli hareketinin modellenmesi ve kontrolü üzerine çalışılmıştır. Yapılan çalışmaların simülasyonu MATLAB programında oluşturulmuştur. Boe-Bot isimli mobil robot ile de gerçek ortamda denemeler yapılmıştır. Bir grup mobil robotun belli bir hedefe giderken birbirlerine çarpmamaları üzerine çalışılmıştır.

Kaplan (2003) tarafından robot futbolu yarışmalarına katılabilecek bir kontrol sistemi üzerine çalışılmıştır. Robot futbolunun modellenmesinde potansiyel alan metodundan faydalanılmıştır. En iyi şekillendirilmiş kontrolör modüllerini bulabilmek için genetik algoritmalarından yararlanılmıştır. Genetik algoritmada uygunluk fonksiyonlarını bulmak için takımların kendi aralarındaki maçlardan faydalanılmıştır.

Simülasyonlarda kendi takımlarının her bir takımla 15000'den fazla maç yaptığı belirtilmiştir.

Myers vd (2005) tarafından holonomik olmayan mobil robotlar için hareketli engellerin olduğu bir ortamda hedefe ulaşılması üzerine çalışılmıştır. Çalışmalar hem simülasyon ortamında hem de gerçek ortamda denenmiştir. Mobil robot engelleri Ladar sensörü ile tespit etmektedir. Sensör 180°'lik bir görüş alanında ve 20m uzaklıktaki engelleri fark edebilmektedir. Gradyan Hız Engel algoritmasını (Gradient Velocity Obstacle Algorithm) kullanarak engellerden kaçılması sağlanmıştır. 248 farklı durumdaki simülasyon çalışmalarında mobil robotun %89.1 oranında hedefe ulaşmada başarı gösterdiği belirtilmiştir.

Xu vd (2003) tarafından E'GV olarak isimlendirilen prototip bir mobil robot ile engellerden kaçınma üzerine çalışılmıştır. Lazer sensörden gelen verilere göre kutupsal obje tablosu ile nerelerden geçilebileceği çıkarılmıştır. Daha önceden bilinmeyen engellerle karşılaşıldığında bunlara çarpmadan mobil robotun istenen yolda devam etmesi sağlanmıştır.

Ge ve Cui (2002) tarafından hareketli engellerin olduğu bir ortamda mobil robotu potansiyel alan metodu ile hedefe ulaştırma üzerine çalışılmıştır. Hem simülasyon ortamında hem de gerçek bir mobil robot üzerinde denemeler yapılmıştır. Kullanılan diferansiyel sürürlü mobil robotun boyutları 7,5cm x 7,5cm x 7,5cm dir.

Bruijnen vd (2007) tarafından hareketli bir çevrede gerçek zamanlı yol planlaması üzerine çalışılmıştır. Bu çalışmada önerilen yaklaşım potansiyel alan metodu ile karşılaştırılmış ve daha iyi sonuçların elde edildiği vurgulanmıştır. Denemelerin sadece simülasyon ortamında değil gerçek ortamda da yapıldığı belirtilmiştir.

1.1.5. Engelin olmadığı ortamlardaki uygulama çalışmaları

Gürel (2006) tarafından mobil robotun nerede olduğunun bulunması ve hareketinin takibi çalışılmıştır. Mobil robot olarak Pioneer 3-dx robotu kullanılmıştır. Kamera ile alınan bilgiler işlenerek hedef tespit edilmiştir. Ana renkler (kırmızı-yeşil-mavi) kullanılarak bir ayırım yapılmıştır. Kamera olarak web kamerası kullanıldığından

görüntü kalitesinin düşük olduğu belirtilmiştir. Dolayısıyla da görüntü işlemede bazen hatalı durumların oluştuğu söylenmiştir.

Öktem (2005) tarafından üzerine ileride başka modüller eklenebilecek bir mobil robot tasarımı çalışılmıştır. 3 tekerlekli bir mobil robot tasarlanmıştır. Arkadaki 2 tekerleğin her birine 1 adet DC motor kullanılmıştır. Motorların kontrolünde hem PIC16F877 hem de bilgisayar kullanılmıştır. Microsoft Visual C++ 6.0 programında yazılan yazılım ile bilgisayardan mikrodenetleyiciye komutlar yollanarak motorlar kontrol edilmiştir. Gerilim, devir ve oyunkolu olarak adlandırılan 3 farklı modda mobil robot hareket ettirilmiştir.

Özdemir (2006) tarafından çizgi takibi yapan bir mobil robot üzerine çalışılmıştır. Çizgi siyah bir zemin üzerinde beyaz renkte seçilmiştir. Mobil robot 5 tekerlekli yapılmıştır. Hareket için 2 adet DC motor kullanılmış ve kontroller için PIC 16F84 tercih edilmiştir. Çizgiyi anlayabilmek için CNY 70 yansımali renk sensörü kullanılmıştır. Bu sensörlerden 3 adet kullanılarak alınan verilere göre mobil robotun ne tarafa döneceği tespit edilmiştir.

Türker (2005) tarafından diferansiyel sürüslü bir mobil robotta odometri hatalarının belirlenmesi ve azaltılması üzerine çalışılmıştır. Mobil robot 2 adet DC motora sahiptir. Burada bahsedilen odometri, mobil robotun tekerlek dönüş bilgisinden elde edilen yer değiştirme bilgisidir. Bu çalışmada sistematik odometri hataları ile çalışılmıştır. Sistematik olmayan hataların düzeltilmesiyle uğraşılmamıştır. Sistematik hatalar tekerleğin kat ettiği mesafe ile orantılı olarak artmaktadır. Sistematik olmayan hatalara örnek olarak tekerleğin olduğu yerde dönmesi verilebilir. Sistematik odometri hatalarının azaltılması için mobil robota üç yörünge takip ettirilmiş ve kare testine tabi tutulmuştur.

Yavaşoğlu (2005) tarafından endüstriyel bir mobil robot gerçekleştirilmiştir. Bu mobil robot, tekstil tezgâhı makarasını bir yerden başka bir yere taşıyacak şekilde tasarlanmıştır. Mobil robot hareketinde çizgi izlemeden yararlanılmıştır ve çizgi izleme kısmında bulanık mantıktan yararlanılmıştır. Bulanık mantık kısmı MATLAB programı ile yapılmıştır. Mobil robotun donanım kısmında 2 adet DC motor ve 1 adet servo motor

kullanılmıştır. DC motorlar mobil robotun hareketi için kullanılmıştır. Servo motor ise makaranın değiştirilmesinde kullanılmıştır.

Yılmaz (2005) tarafından web tabanlı bir mobil robot sistemi üzerine çalışılmıştır. Tasarlanan mobil robotta 3 adet tekerlek bulunmaktadır. Arkada bulunan 2 tekerlek serbest tekerlek olarak tasarlanmıştır ve öndeki tekerlek ile hareket ve yönlendirme verilmiştir. Ayrıca mobil robotun üzerine bir adet dizüstü bilgisayar yerleştirilmiştir. Bu bilgisayar ile internet bağlantısı, mobil robot hareketleri ve görüntü işleme fonksiyonları yapılmıştır. Mobil robotun kamerasından alınan görüntüler internetten canlı olarak gösterilmiştir. Mobil robotun internet üzerinden programlanabileceği belirtilmiştir. Şekil 1.17'de WEB-SUNAR isimli bu mobil robotun bir görüntüsü verilmiştir.



Şekil 1.17 WEB-SUNAR (Yılmaz, 2005)

Yüksel (2004) tarafından mobil robotta görüntü işleme ile istenen hedefe gidilmesi üzerine çalışılmıştır. Bu çalışmada mobil robot, Lego MindStorm RIS (Robotics Invention System) seti ile oluşturulmuştur. Tepe kamerasından alınan görüntü bilgisi ile siyah renkte olan hedefin konumu belirlenmiştir. Konumu belirlenen hedefe mobil robotun gitmesi sağlanmıştır. Bu çalışmada ortamda engel olmadığı varsayılmıştır.

1.1.6. Engelin olmadığı ortamlardaki simülasyon çalışmaları

Barlas (2004) tarafından Mars yüzeyinde araştırma yapan mobil robotlar için yeni bir süspansiyon mekanizması tasarlanılmıştır. Bu mekanizmanın kinematik analizleri yapılmıştır. Bulunan sonuçların Visual Nastran 4D programı ile simülasyonu gerçekleştirilmiştir. Tezin son kısmında günümüzde mekanik tasarımın güçlü kontrol algoritmalarından dolayı popülerliğini kaybettiği belirtilmiştir.

Literatür araştırmasından da görüldüğü gibi mobil robot konusunda çok fazla çalışma yapılmıştır. Çalışmaların büyük bir kısmı yol planlaması üzerinedir. Ayrıca sadece simülasyon olarak yapılan çalışmaların, hem simülasyon hem de gerçek bir mobil robota uygulama şeklinde yapılan çalışmalara nazaran daha fazla olduğu görülmektedir. Bununla birlikte sadece simülasyon olarak yapılan çalışmaların bir kısmında da gerçek bir mobil robota uygulama hedefi konulmuştur. Aslında tüm simülasyon çalışmalarının nihai hedefi bir pratik çalışmada uygulanabilmesidir. Bu sebeple bu çalışmada da hem teorik hem de pratik çalışma yapılmıştır. Literatür araştırmasından görülmüştür ki sabit engelli bir alanda çalışma yerine hareketli bir alanda çalışma önem kazanmıştır. Bu sebepten dolayı buradaki yapılan çalışmada hem sabit hem de hareketli engellerin olabildiği bir ortam oluşturulmuştur. Hareketli engellerin olduğu bir ortamda çalışmanın literatürdeki popülerliği görüldüğünden dolayı burada yapılacak bir yeniliğin makale olabileceği fark edilmiştir. Bu çalışma kapsamında ortaya bir yenilik konulamamasına rağmen bu çalışmanın devamı durumunda makale çıkacağı tahmin edilmektedir. Ayrıca literatür araştırması sonucunda şimdiye kadar ki çalışmalarda tahrik mekanizması olarak servo motorların kullanılmadığı görülmüştür.

1.2. Mobil Robotun Tasarımı ve İmalatı

Bu tezde diferansiyel sürüş sistemine sahip bir mobil robotun tasarımı yapıp imal edilmiştir. Mobil robotun tasarımında SolidWorks programı kullanılmıştır. Mobil robotta 4 adet tekerlek bulunmaktadır. Bu tekerleklerden ikisi 90 mm çapında avare tekerlerdir. Mobil robotun dengesinin sağlanması için kullanılmışlardır. Diğer iki tekerlek ile ise mobil robotun hem dönme hem de öteleme hareketi sağlanmaktadır. Her iki tekere ayrı ayrı tahrik verilmektedir. Bu iki tekerlek scooter tekerleğidir ve çapları 430 mm'dir. Mobil robotun tahrik sistemi, motorların uçlarına redüktör ve redüktörlerin uçlarına da tekerleklerin bağlanması şeklinde yapılmıştır.

Mobil robotun boyutları (exdxy) 540x600x550(mm) ve aks mesafesi 770 mm'dir. Mobil robotu 3 tabaka şeklinde ayırabiliriz. Alt tabakada kesintisiz güç kaynağı, 2 adet motor, 2 adet redüktör ve 2 adet de servo sürücüsü (driver) bulunmaktadır. Orta tabakada bilgisayarın anakartı, güç kaynağı ve harddiski, 24 volt güç kaynağı, hareket kontrol kartı (motion control card) ve kablolama kartı (wiring board) bulunmaktadır.

Üst tabaka da ise acil durma butonu, bilgisayarın ekranı, klavyesi ve faresi bulunmaktadır.

Kullanılan redüktörler 3,5 kg ağırlıktadırlar ve 15 dakika boşlukludurlar. Redüktörlerin oranı 1/25'tir ve bu oran sayesinde motorların ürettikleri torklar 25 kat güçlendirilmektedirler. Maksimum 63,5 Nm tork elde edilebilmektedir.

Servo sürücülerinin boyutları (exdxy) 59x147x162(mm) ve ağırlıkları 1,2 kg'dır. Servo sürücüler 220V ile çalışmaktadırlar. Servo sürücülerin güç kaynağı UPS'tir. Servo motorların boyutları (exdxy) 60x60x161(mm) ve ağırlıkları 1,6 kg'dır. Servo motorlar ayrıca 400W'lıktır. Servo motorlar 360°'yi 10000'e bölebilmektedirler yani 0.036°'lik bir hassasiyetleri vardır. Kullanılan UPS'in boyutları (exdxy) 192x455x330(mm) ve ağırlığı 33kg'dır. UPS'in gücü 3000VA'lıktır.

Hareket kontrol kartının boyutları (enxboy) 122x175mm'dir. Hareket kontrol kartı bilgisayar ile servo sürücüler arasındaki haberleşmeyi sağlamaktadır. Kablolama kartının boyutları (enxboy) 107x200mm'dir. Kablolama kartı hareket kontrol kartındaki uçların dışarıya alınabilmesini ve diğer parçalarla bağlantı yapılabilmesini sağlamaktadır.

1.3. Mobil Robotun Kontrolü

Mobil robotun yazılımı Visual C# .NET programında yazılmıştır. Yazılım ile hareket kontrol kartına erişilmektedir. Hareket kontrol kartı da servo sürücülere komutları yollamaktadır. Servo sürücülere yollanan komutlara göre de servo motorlar dönmektedirler.

Hareket kontrol kartına ait fonksiyonlar ile bu karta erişim yapılabilmektedir. Ayrıca yazılımın proje dosyasına kartla ilgili dll dosyalarının da eklenmesi gerekmektedir.

Yazılıma Visual C# .NET 2005 ile başlanmıştır ve daha sonra yeni sürümün çıkmasıyla (Visual C# .NET 2008) bu sürümle yazılmaya devam edilmiştir. 2008 sürümüyle birlikte gelen LINQ'da yazılımda kullanılmıştır. LINQ engellerin konumlarının sorgulanmasında kullanılmıştır. Yazılım nesne tabanlı (object-oriented)

olarak yazılmıştır. Yazılım daha sonra geliştirilebilmesi için uygun sınıflara ayrılmıştır. Örneğin engel çizimiyle alakalı olan elips, üçgen vb. şekiller için sınıflar hazırlanmıştır. Geliştirici başka şekiller eklemek istediğinde veya bu şekillere ait özellikleri arttırmak istediğinde kolay adapte olabilecektir. Şekillerin çizimi, 2 boyutlu bir çizim programındaki gibi rahat kullanıma sahip yapılmaya çalışılmıştır.

Mobil robotun yol planlamasında potansiyel alan metodundan faydalanılmıştır. Potansiyel alan metodu ile noktalardan oluşan bir yol elde edilmektedir. Mobil robot bu noktalardan geçerek hedefe ulaşabilmektedir. Diferansiyel sürüş sistemi sayesinde mobil robot olduğu yerde dönebilmektedir. Bu da mobil robotun manevra ihtiyacı duymadan yolu oluşturan noktalar üzerinde gidebilmesini sağlamaktadır. Mobil robotun hareket kontrolünün oluşturulmasında redüktörün döndürme oranı, aks mesafesi, tekerlek yarıçapı ve diferansiyel sürüş sistemi dikkate alınmıştır.

1.4 Tezin Amacı ve Tanıtımı

Bu tezin amacı engelli bir alanda gerçek bir mobil robotun hedefine varabilmesini sağlamaktır. Ayrıca bu tezin başka bir amacı da ileride üzerinde çalışabilecek bir mobil robotun yapılmasıdır. Bu proje kapsamında yeni bir mobil robot yapılmıştır. Hazır olarak satılan mobil robotlardan alınmamıştır. Mobil robotun yazılımı engellerin hem hareketli hem de hareketsiz olabileceği durumlara uygun olarak yazılmıştır.

Tezin ikinci bölümünde mobil robotun parçaları tanıtılacaktır. Bu parçaların fotoğrafları ve SolidWorks programındaki katı modellemeleri verilecektir. Ayrıca mobil robotun bitmiş halinin fotoğrafları ve SolidWorks programındaki katı modellemesi verilecektir. Bunlardan başka ikinci bölümde mobil robotun sürüş sisteminden bahsedilecektir.

Tezin üçüncü bölümünde potansiyel alan metodunun matematiksel alt yapısından ve potansiyel alan metodunun uygulanmasından bahsedilecektir. Daha sonra yazılımdaki kontroller ve sınıflar, ışın analizi yöntemi, yolun yumuşatılması ve farklı şekillerin nasıl engel olarak algılandığı anlatılacaktır. Bunlardan başka mobil robotun hedefine ulaştığı bazı örnekler verilecektir. Son olarak da hareket kontrol kartı'na yazılım ile erişim anlatılacaktır.

Tezin dördüncü bölümünde yapılan çalışmanın özeti verilecektir. Proje sırasında karşılaşılan zorluklar ve ileride bu çalışmaya ek olarak neler yapılabileceğinden bahsedilecektir.

2. MOBİL ROBOTUN DONANIMI

2.1 Mobil Robotun Parçaları

Mobil robotun donanımı temel olarak aşağıdaki parçalardan oluşmaktadır.

- 1 adet bilgisayar
- Profil kafes
- 1 adet UPS
- 2 adet servo sürücü
- 2 adet servo motor
- 2 adet redüktör
- 1 adet hareket kontrol kartı
- 1 adet kablolama kartı
- 1 adet 24V güç kaynağı
- 2 adet motor tekeri
- 2 adet avare teker
- 1 adet monitör

Yukarıdaki parçalara ek olarak profil kafese alt, orta ve üst plaka yaptırılmıştır. Ayrıca plakalara bazı parçaların tutturulması için ek parçalar yaptırılmıştır. Profil kafes Şekil 2.1’de görülmektedir.

Profil kafes sistemi 45x45(mm)’lik alüminyum profilden yaptırılmıştır. Profil kafesin en, derinlik ve yükseklik değerleri sırasıyla 540x600x550(mm) şeklindedir. UPS olarak Newtech’in 3 KVA’lık modeli seçilmiştir. UPS’in özellikleri Tablo 2.1’de verilmiştir. Şekil 2.2’de UPS’in fotoğrafı ve katı modellemesi verilmiştir.

Tablo 2.1 UPS'in özellikleri (WEB_8 2009)

Model	3 KVA
Giriş	
Gerilim	160 - 300 VAC, 1Faz
Frekans	50/60 Hz \pm %8
Güç Faktörü	> 0.98
Çıkış	
Güç VA / W	3000 VA / 2100W
Gerilim	220 VAC \pm %3.1 Faz
Frekans(Aküden)	50 Hz \pm %0.5 Hz
Dalga Şekli	Gerçek Sinüs, THD < %3 tam yükte
Güç Faktörü	0.7
Crest Faktörü	3.1
Verim(AC / AC)	>%85
Aşırı Yük Kapasitesi	>%105 60s, >%130 5s
Akü	
Akü Tipi	Tam bakımsız kuru tip kurşun asit UPS aküsü
Akü Adedi	8x12V / 7 Ah
Akü Şarj Süresi	8 saat (Nominal kapasitenin %90'ına)
Fiziksel Özellikler	
Sıcaklık	0° C - 40° C
Nem	Max %95, yoğunlaşmayan bağıl nem
Boyutlar(ExDxY, mm)	192x455x330
Net Ağırlık(kg)	33

**Şekil 2.1** Profil kafes sistemi

Delta Electronics, Inc. firmasının ASD-B0421-A model servo sürücüsü ile ECMA-C30604GS model servo motoru kullanılmıştır. Şekil 2.3'te servo sürücünün, Şekil 2.4'te servo motorun fotoğrafı ve katı modellemesi bulunmaktadır. Servo sürücü ve servo motor 0,4 KW'lıktır. Servo sürücü, servo motora hem güç sağlamaktadır hem de kontrol sinyallerinin yollanmasını sağlamaktadır. Servo sürücünün en, derinlik ve yükseklik değerleri 59x147x162mm şeklindedir. Servo motorun ise en, derinlik ve yükseklik değerleri 60x60x160mm şeklindedir. Servo sürücü 1,2 kg ve servo motor 1,6 kg ağırlığındadır. Servo sürücüye hem kablolama kartı ile hem de servo motor ile kablo bağlantısı yapılmaktadır. Servo motora ise sadece servo sürücü ile kablo bağlantısı yapılmaktadır. Servo motor 360°'yi 10000 adıma bölebilmektedir. Bu da bir adımın 0.036°'ye karşılık geldiğini göstermektedir. Servo motorun torku 1,27 Nm'dir (WEB_3 2009). Bu tork ayrıca redüktör ile güçlendirilmektedir. Böylece mobil robot için yeterince güçlü bir teker döndürme sistemi elde edilmektedir.

Redüktör olarak planetRoll firmasının PD085 – eAH025 – 1AA0 model redüktörü kullanılmıştır. Şekil 2.8'de redüktörün fotoğrafı ve katı modellemesi bulunmaktadır. Redüktörün çevirme oranı 1/25 olduğundan dolayı motor 25 tur attığında redüktör 1 tur atmaktadır. Redüktör tur sayısını azaltırken aynı oranda torku güçlendirmektedir. Redüktörün ucundaki tork motorun ucundaki torka nazaran 25 kat daha güçlü olmaktadır. Motorun 1,27 Nm'lik torkunu $1,27 \times 25 = 31,75$ Nm'ye çıkarmaktadır. 2 motor olduğu için toplam tork $2 \times 31,75 = 63,5$ Nm olmaktadır. Redüktörün ağırlığı 3,5kg'dır (WEB_2 2008). Redüktörün boşluk miktarı 15 dakikadır.

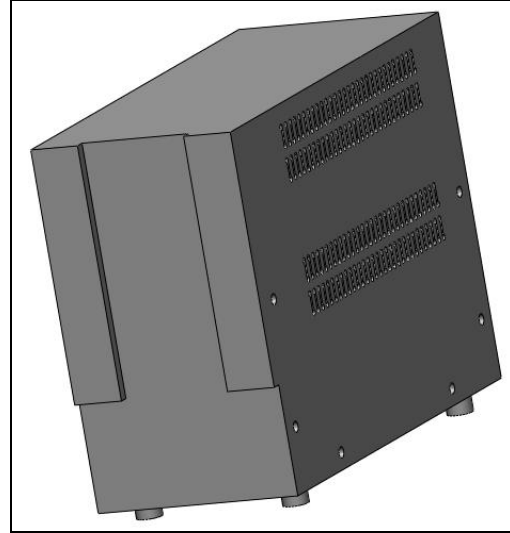
Kablolama kartı hareket kontrol kartının bilgisayarın dışına alınabilmesini sağlamaktadır. Şekil 2.5'te kablolama kartının fotoğrafı gösterilmektedir. Kablolama kartı'ndan servo sürücüye ve 24 volt güç kaynağına kablo bağlantısı yapılmaktadır. Kablolama kartı'nın en ve boy değerleri 107x200mm şeklindedir. Hareket kontrol kartının en ve boy değerleri 122x175mm şeklindedir (WEB_5 2008). Şekil 2.6'da 24 volt güç kaynağının fotoğrafı ve katı modellemesi, Şekil 2.7'de bilgisayarın anakartı, güç kaynağı ve harddiski gösterilmektedir. Mobil robotun SolidWorks programındaki katı modellemesindeki perspektif görüntüsü Şekil 2.9'da, ön görünüşü Şekil 2.10'da, arka görünüşü Şekil 2.11'de ve üst görünüşü Şekil 2.12'de verilmektedir. Mobil robotun perspektif görüntüsünün fotoğrafı Şekil 2.13'te, ön görünüşünün fotoğrafı Şekil 2.14'te,

arka görünüşünün fotoğrafı Şekil 2.15'te ve üst görünüşünün fotoğrafı Şekil 2.16'da verilmektedir.



(a)

Şekil 2.2 a) UPS'in fotoğrafı



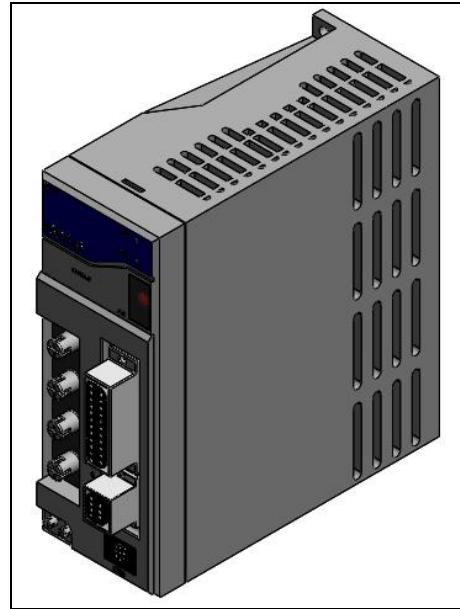
(b)

b) UPS'in katı modellemesi



(a)

Şekil 2.3 a) Servo sürücünün fotoğrafı



(b)

b) Servo sürücünün katı modellemesi



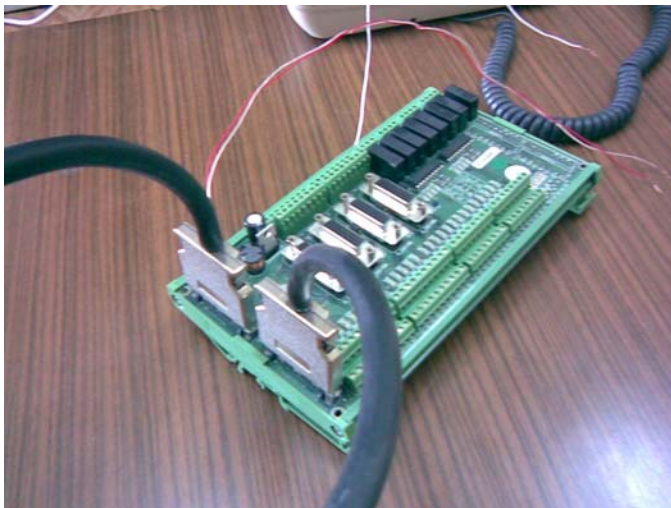
(a)

Şekil 2.4 a) Servo motorun fotoğrafı

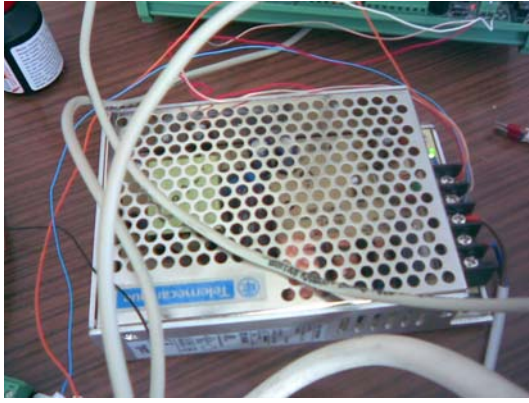


(b)

b) Servo motorun katı modellemesi

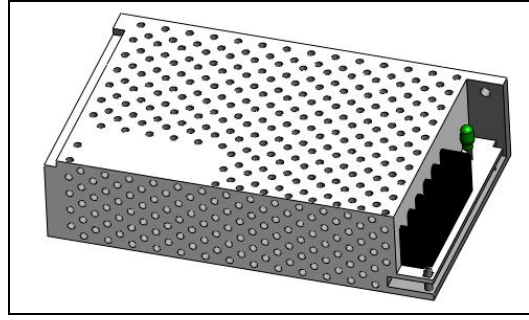


Şekil 2.5 Kablolama kartının fotoğrafı



(a)

Şekil 2.6 a) 24 volt güç kaynağının fotoğrafı



(b)

b) 24 volt güç kaynağının katı modellemesi

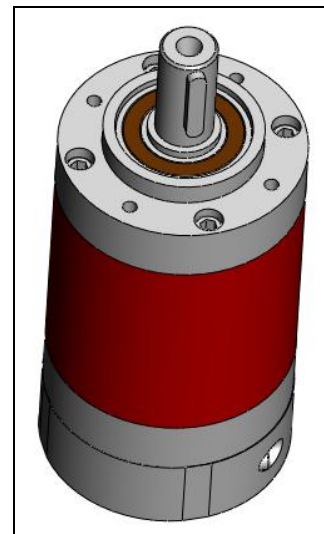


Şekil 2.7 Bilgisayarın anakartı, güç kaynağı ve harddiski



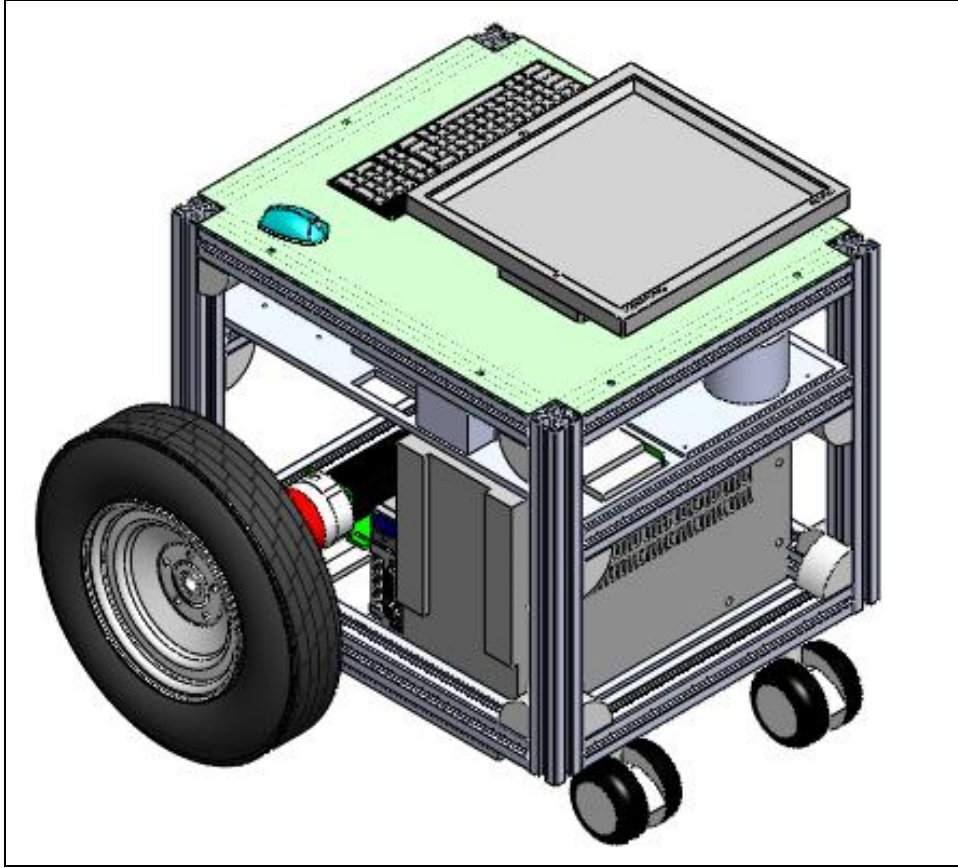
(a)

Şekil 2.8 a) Redüktörün fotoğrafı

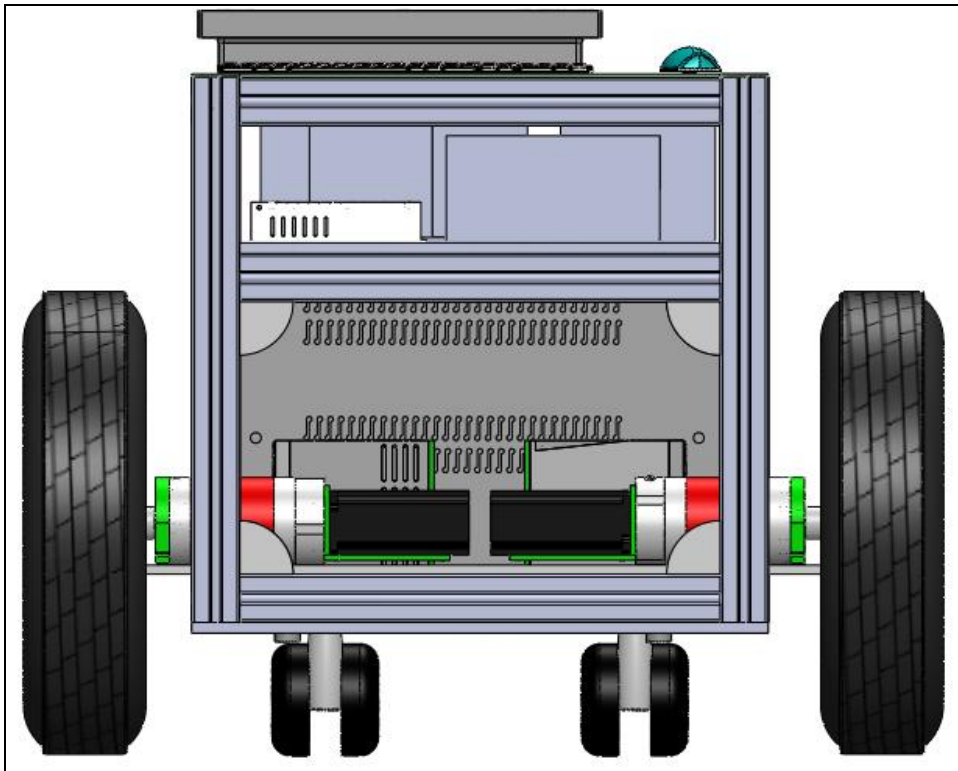


(b)

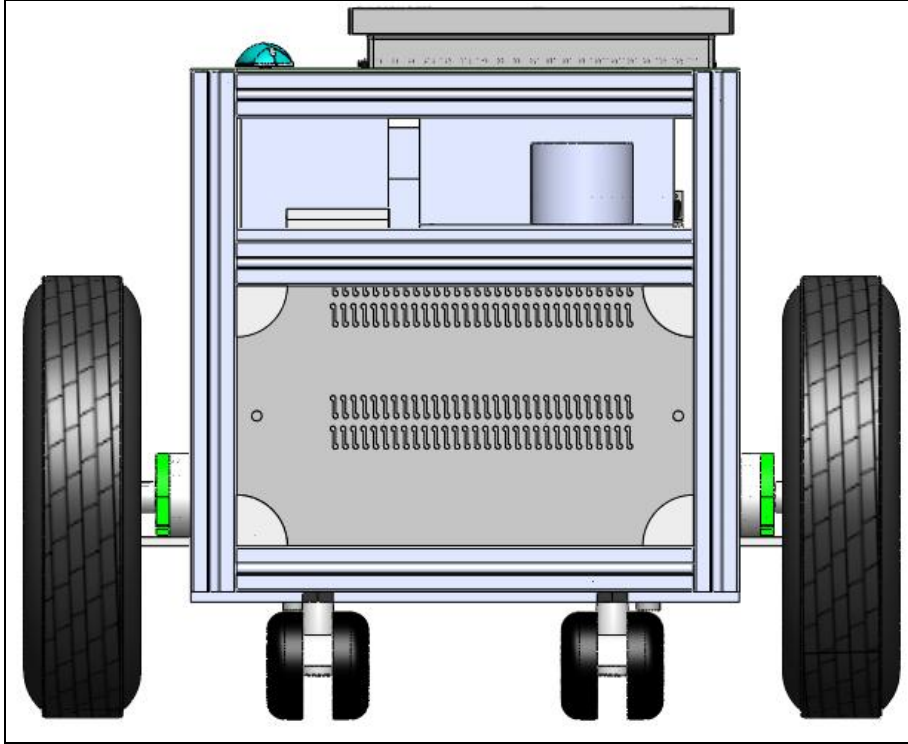
b) Redüktörün katı modellemesi



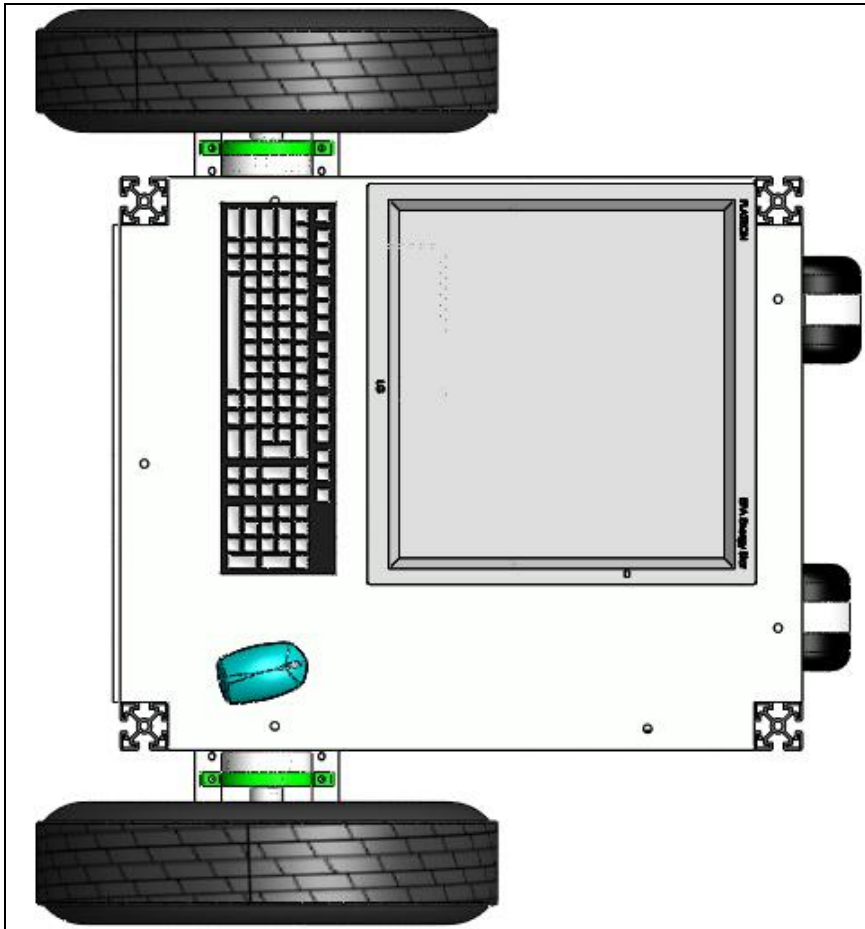
Şekil 2.9 Mobil robotun SolidWorks programındaki katı modellemesi (perspektif)



Şekil 2.10 Mobil robotun SolidWorks programındaki katı modellemesi (ön görünüş)



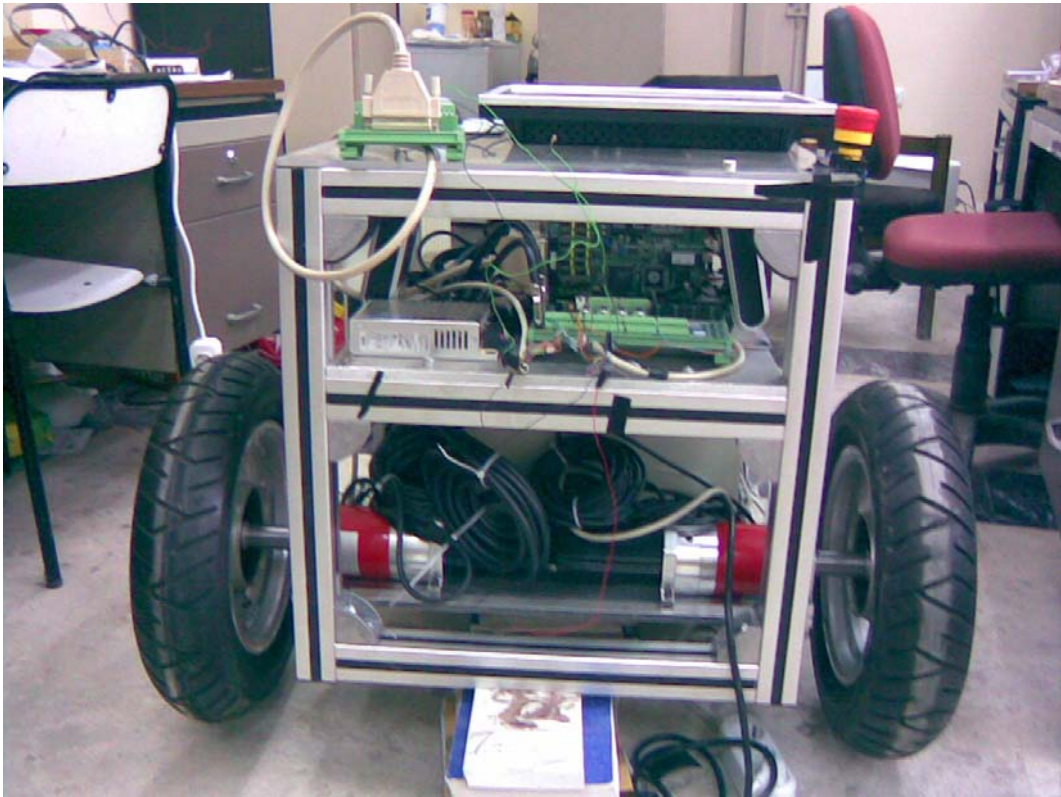
Şekil 2.11 Mobil robotun SolidWorks programındaki katı modellemesi (arka görünüş)



Şekil 2.12 Mobil robotun SolidWorks programındaki katı modellemesi (üst görünüş)



Şekil 2.13 Mobil robotun fotoğrafı (perspektif)



Şekil 2.14 Mobil robotun fotoğrafı (ön görünüş)



Şekil 2.15 Mobil robotun fotoğrafı (arka görünüş)



Şekil 2.16 Mobil robotun fotoğrafı (üst görünüş)

Donanım ile ilgili yukarıda verilen fotoğraf ve resimlere ek olarak Ek-2’de mobil robotun destek ve bağlantı parçalarına ait teknik resimler verilmiştir.

2.2. Mobil Robotun Sürüş Sistemi

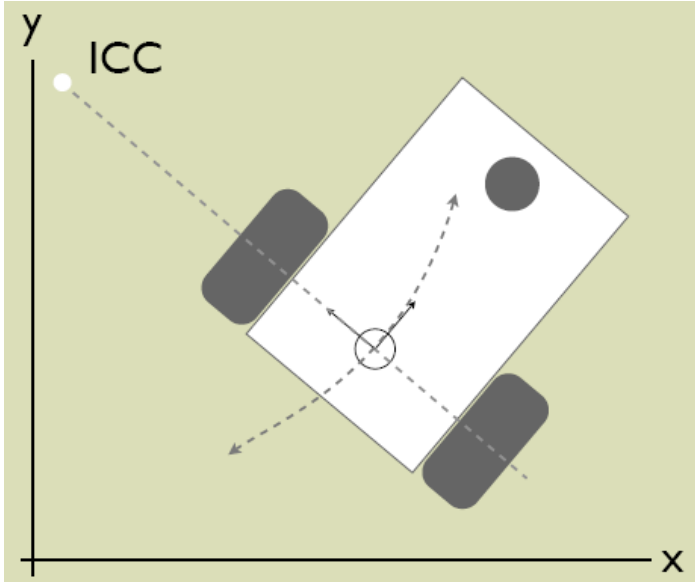
Mobil robota sürüş sistemi olarak diferansiyel sürüş sistemi uygulanmıştır. Diferansiyel sürüş sisteminde 2 adet tekerlek ayrı ayrı hareket ettirilebilecek şekilde tasarlanmıştır. Denge için bu iki tekerlek dışında birkaç adet avare teker takılabilmektedir. Buradaki mobil robotta ön tarafa 2 adet avare teker takılmıştır. Aşağıdaki denklemlerde V_f doğrusal hızı, θ açısal hızı, ϕ_r sağ teker hızı, ϕ_l sol teker hızı, l aks uzunluğunu, r tekerlek yarıçapını ve R_{ICC} mobil robotun etrafında döndüğü nokta ile aks merkezi arasındaki uzaklığı simgelemektedir (WEB_4 2008). Mobil robotun etrafında döndüğü nokta Şekil 2.17’de gösterilmiştir.

$$V_f = \frac{r}{2}(\phi_r + \phi_l) \quad (2.1)$$

$$\theta = \frac{r}{l}(\phi_r - \phi_l) \quad (2.2)$$

$$R_{ICC} = \frac{l}{2} \frac{\phi_r + \phi_l}{\phi_r - \phi_l} \quad (2.3)$$

Buradaki formüller üzerinden bazı durumları inceleyelim. Eğer sol ve sağ tekerlek farklı yönlerde ve aynı hızda hareket ederlerse ($\phi_r = -\phi_l$) doğrusal hız sıfıra eşit olmaktadır. Açısal hız da en büyük değerini almaktadır. Bu da mobil robotun kendi etrafında döndüğünü açıklamaktadır. Ayrıca R_{ICC} mobil robotun dönme merkezine uzaklığı da sıfır çıkmaktadır. Sol ve sağ tekerlek aynı yönde ve aynı hızda hareket ederse ($\phi_r = \phi_l$ durumunda) $\theta = 0$ çıkmaktadır. Bu da mobil robotun düz bir şekilde ilerlediğini ifade etmektedir. Ayrıca $R_{ICC} = \infty$ çıkmaktadır. Bu sonuç da mobil robotun herhangi bir nokta etrafında dönmediğini belirtmektedir.



Şekil 2.17 Mobil robotun dönme merkezi(WEB_4 2008)

3. MOBİL ROBOTUN YAZILIMI

Mobil robotun yazılımının temelini Potansiyel Alan Metodu oluşturmaktadır. İlk önce Potansiyel Alan Metodunun matematiksel alt yapısı anlatılacaktır. Daha sonra Potansiyel Alan Metodunun nasıl uygulandığı anlatılacaktır. Mobil robot yazılımı olarak 2 adet program yazılmıştır. Temelde bu iki program da aynı işlevi yapmaktadırlar. İkinci program daha profesyonelce yazılmıştır ve dolayısıyla ilk programdaki kod karmaşasından kurtulunmuştur.

3.1. Potansiyel Alan Metodunun Matematiksel Alt Yapısı

3.1.1. Kısmi türev tanımı

x ve y 'ye bağlı bir H fonksiyonunun herhangi bir (x,y) noktasındaki x 'e göre kısmi türevi;

$$\frac{\partial H}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{H(x + \Delta x, y) - H(x, y)}{\Delta x}$$

şeklindedir. Benzer şekilde H fonksiyonunun y 'ye göre kısmi türevi;

$$\frac{\partial H}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{H(x, y + \Delta y) - H(x, y)}{\Delta y}.$$

3.1.2. Sonlu fark yöntemleri

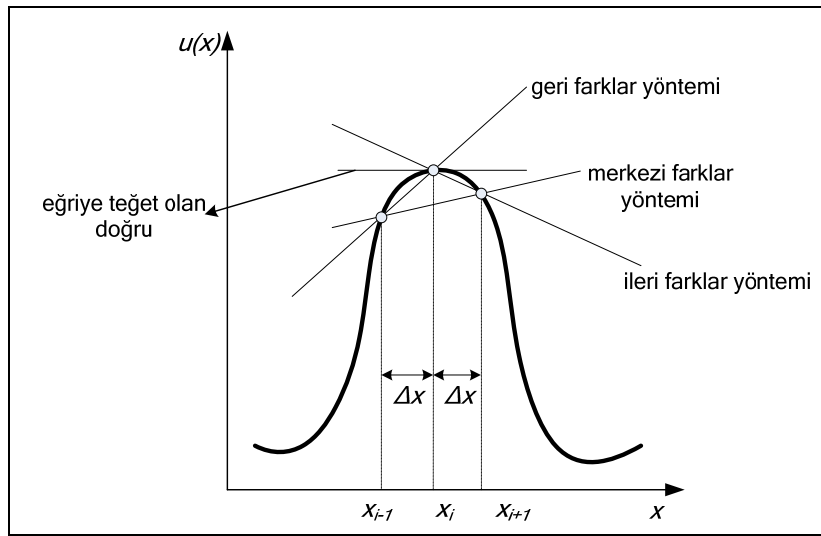
Sonlu fark yöntemleri kısmi diferansiyel denklemlerin nümerik olarak çözülmesi için kullanılan yöntemlerdendir. Sonlu fark yöntemleri temel olarak üçe ayrılmaktadır. Bunlar geri fark yöntemi, merkezi fark yöntemi ve ileri fark yöntemidir. Sonlu fark yöntemleri geometrik yorumla Şekil 3.1'deki gibi izah edilebilir. Şekil 3.1'den de görüldüğü gibi x_i noktasındaki eğrinin tanjantı üç farklı şekilde izah edilebilir. Ayrıca

bu tanjantın değeri x_i noktasındaki birinci mertebeden türevi vermektedir. Kısmi türev tanımından yola çıkılarak elde edilen ileri fark, geri fark ve merkezi fark yöntemleri sırasıyla aşağıdaki gibi tanımlanmaktadır:

$$\left(\frac{du}{dx}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x} \quad (3.1)$$

$$\left(\frac{du}{dx}\right)_i \approx \frac{u_i - u_{i-1}}{\Delta x} \quad (3.2)$$

$$\left(\frac{du}{dx}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (3.3)$$



Şekil 3.1 Sonlu Fark Yöntemlerinin geometrik yorumu

Birinci mertebeden türevler için formüller yukarıda yazılmıştır. Şimdi de bu üç yöntemin ikinci mertebeden türev için çıkarılışını yazalım. Aşağıdaki formüllerde $\Delta x = h$ olarak alınmıştır. İleri fark için;

$$\begin{aligned} \left(\frac{d^2u}{dx^2}\right)_i &= \frac{d}{dx} \left(\frac{du}{dx}\right)_i \\ &\approx \frac{1}{\Delta x} \left[\left(\frac{\Delta u}{\Delta x}\right)_{i+1} - \left(\frac{\Delta u}{\Delta x}\right)_i \right] \\ &= \frac{1}{h} \left[\frac{u_{i+2} - u_{i+1}}{h} - \frac{u_{i+1} - u_i}{h} \right] \end{aligned}$$

$$= \frac{u_{i+2} - 2u_{i+1} - u_i}{h^2} \quad (3.4)$$

Birinci mertebeden geri fark formülünden yararlanarak aşağıdaki gibi ikinci mertebeden geri fark formülünü de çıkarabiliriz:

$$\begin{aligned} \left(\frac{d^2 u}{dx^2} \right)_i &\approx \frac{1}{\Delta x} \left[\left(\frac{\Delta u}{\Delta x} \right)_i - \left(\frac{\Delta u}{\Delta x} \right)_{i-1} \right] \\ &= \frac{1}{h} \left[\frac{u_i - u_{i-1}}{h} - \frac{u_{i-1} - u_{i-2}}{h} \right] \\ &= \frac{u_i - 2u_{i-1} + u_{i-2}}{h^2} \end{aligned} \quad (3.5)$$

Birinci mertebeden merkezi fark formülünden yararlanarak aşağıdaki gibi ikinci mertebeden merkezi fark formülünü çıkarabiliriz:

$$\begin{aligned} \left(\frac{d^2 u}{dx^2} \right)_i &\approx \frac{1}{2\Delta x} \left[\left(\frac{\Delta u}{\Delta x} \right)_{i+1} - \left(\frac{\Delta u}{\Delta x} \right)_{i-1} \right] \\ &= \frac{1}{\Delta x} \left\{ \frac{1}{2} \left[\left(\frac{\Delta u}{\Delta x} \right)_{i+1} + \left(\frac{\Delta u}{\Delta x} \right)_i \right] - \frac{1}{2} \left[\left(\frac{\Delta u}{\Delta x} \right)_i + \left(\frac{\Delta u}{\Delta x} \right)_{i-1} \right] \right\} \\ &= \frac{1}{\Delta x} \left[\left(\frac{\Delta u}{\Delta x} \right)_{i+1/2} - \left(\frac{\Delta u}{\Delta x} \right)_{i-1/2} \right] \\ &= \frac{1}{h} \left[\frac{u_{i+1} - u_i}{h} - \frac{u_i - u_{i-1}}{h} \right] \\ &= \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \end{aligned} \quad (3.6)$$

Geometrik gösterimden bulunan kısmi türevler, belli bir hata ihmal edilerek çıkarılmaktadır. Bu hatalara Taylor serisinden bakılabilir.

3.1.3. Taylor serisi yaklaşımı ve hatalar

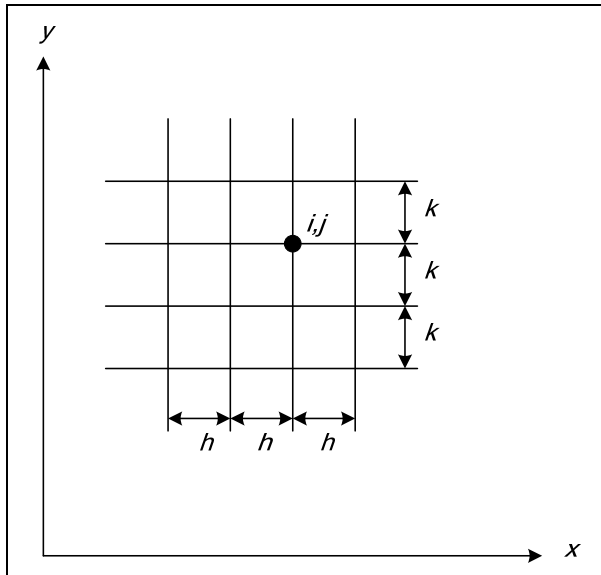
Daha önce çıkarılan formüller bir boyutluydu ve sadece x 'e göre türev alınıyordu. İki boyutlu formülleri çıkarmak için y 'ye göre de türev alınması gerekmektedir. Şekil 3.2'de iki boyutlu bir ızgara sistemi gözükmemektedir. $u(x, y)$ şeklindeki fonksiyonumuz x_i etrafında, (x_i+h) ve (x_i-h) noktalarında Taylor serisine açılırsa;

$$u(x+h, y) = u(x, y) + h \frac{\partial u(x, y)}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u(x, y)}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u(x, y)}{\partial x^4} + \dots \quad (3.7)$$

$$u(x-h, y) = u(x, y) - h \frac{\partial u(x, y)}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u(x, y)}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u(x, y)}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u(x, y)}{\partial x^4} + \dots \quad (3.8)$$

elde edilir.

Burada h ızgara boyunu (grid size) temsil etmektedir. $u(x+h, y)$ yerine $u_{i+1, j}$, $u(x-h, y)$ yerine $u_{i-1, j}$, $u(x, y)$ yerine $u_{i, j}$ yazılarak bundan sonraki denklemler elde edilmektedir. $u_{i+1, j}$ gösteriminde $i+1$ u fonksiyonunun x eksenindeki karşılığını, j ise y eksenindeki karşılığını göstermektedir.



Şekil 3.2 İki boyutlu ızgara sistemi

Denklem 3.7 ve 3.8'i indisli şekilde düzenleyecek olursak;

$$u_{i+1,j} = u_{i,j} + h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} + \dots \quad (3.9)$$

$$u_{i-1,j} = u_{i,j} - h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \quad (3.10)$$

Denklem 3.9'da $\partial u(x, y)/\partial x$ yalnız bırakılırsa;

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i,j}}{h} - \frac{h}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^2}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} - \dots \\ &= \frac{u_{i+1,j} - u_{i,j}}{h} + O(h) \end{aligned} \quad (3.11)$$

Yukarıda denklem 3.11 ile birinci mertebeden $O(h)$ kesme hatalı ileri fark formülü elde edilmiştir. Benzer şekilde Taylor serisi açılımı ile ilgili denklem 3.10'da $\partial u_{i,j}/\partial x$ çekilirse;

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h} + \frac{h}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^2}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \dots \\ &= \frac{u_{i,j} - u_{i-1,j}}{h} + O(h) \end{aligned} \quad (3.12)$$

Denklem 3.12 birinci mertebeden $O(h)$ kesme hatalı geri fark formülüdür. Denklem 3.11 ve 3.12 toplanıp ikiye bölünürse denklem 3.13'deki $O(h^2)$ hatalı birinci mertebeden merkezi fark formülü elde edilir.

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i-1,j}}{2h} - \frac{h^2}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} - \dots \\ &= \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2) \end{aligned} \quad (3.13)$$

Burada $O(h^2)$ kesme hatasını belirtmektedir. Izgara boyu(h) yarıya indirilirse hata değeri eski hata değerinin $1/4$ 'ü olmaktadır. Şekil 3.1'de görüldüğü gibi eğriye teğet olan doğrunun eğimi teğet olduğu noktadaki birinci mertebeden türevi vermektedir. Ayrıca bu eğimin değerine en çok merkezi fark yöntemi ile yaklaşılmaktadır. Taylor serisine

açılım ile elde edilen denklem 3.9 ve 3.10 toplanıp $\partial^2 u_{i,j}/\partial x^2$ çekilirse ikinci mertebeden merkezi fark formülü elde edilir.

$$\begin{aligned}\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} - \frac{2h^2}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\ &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2)\end{aligned}\quad (3.14)$$

Benzer şekilde y 'ye göre türev alınarak aşağıdaki formüller çıkarılabilir.

$$\frac{\partial u_{i,j}}{\partial y} = \frac{u_{i,j+1} - u_{i,j}}{k} + O(k) \quad (\text{İleri fark}) \quad (3.15)$$

$$\frac{\partial u_{i,j}}{\partial y} = \frac{u_{i,j} - u_{i,j-1}}{k} + O(k) \quad (\text{Geri fark}) \quad (3.16)$$

$$\frac{\partial u_{i,j}}{\partial y} = \frac{u_{i,j+1} - u_{i,j-1}}{2k} + O(k^2) \quad (\text{Merkezi fark}) \quad (3.17)$$

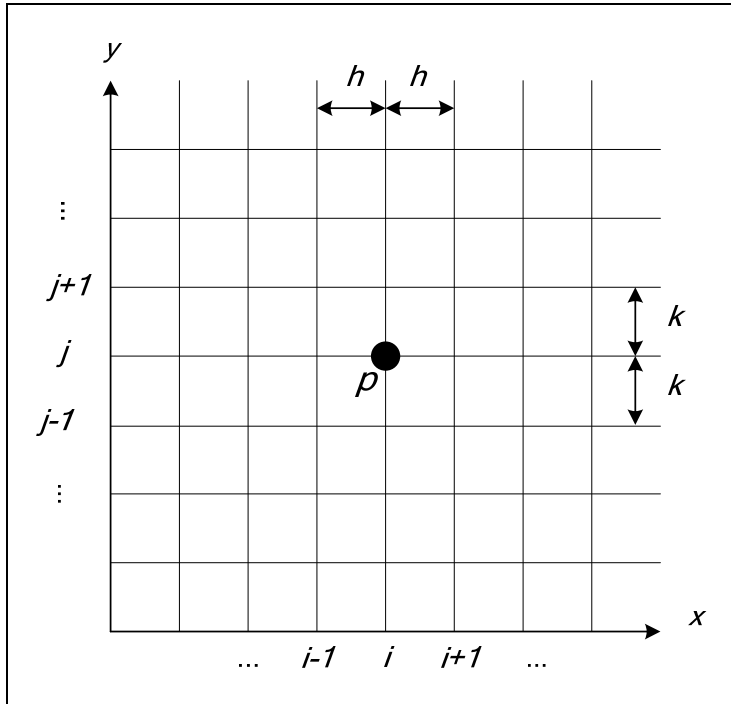
$$\frac{\partial^2 u_{i,j}}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} + O(k^2) \quad (\text{Merkezi fark}) \quad (3.18)$$

Yukarıda Taylor serisi açılımı ile birinci ve ikinci mertebeden sonlu fark denklemleri elde edildi. Yüksek mertebeden sonlu fark denklemlerini de aynı yöntemlerle elde edebiliriz. Fiziksel problemleri temsil eden çoğu kısmi diferansiyel denklemler ikinci mertebeden olduğundan yüksek mertebeli sonlu fark denklemleri daha seyrek kullanılmaktadır. Bu bölümde sonlu fark denklemlerinde kesme hatalarının olduğu gösterilmiştir. Bu hatalar aslında küçük hata değerlerine sahip olduklarından mobil robot için yazılacak programda ihmal edilmiştir. Ek-1'de sonlu fark yöntemlerinin formülleri ayrıntılı bir şekilde verilir ve çıkarılışları yapılmıştır.

3.1.4. Sonlu fark yöntemlerinden potansiyel alan metodunun çıkarılması

İkinci mertebeden kısmi diferansiyel denklemlerinin sonlu fark metodu ile çözümünde, ilk önce çözüm uzayı bir ızgara sistemiyle (grid system) ifade edilir. Izgara sistemi olarak genellikle dikdörtgensel bir ızgara sistemi kullanılmaktadır (Şekil 3.3).

P noktasındaki ikinci mertebeden merkezi fark denklem 3.18 yardımıyla x ve y 'ye göre denklem 3.19 ve 3.20'deki olarak yazılabilir.



Şekil 3.3 P noktasının iki boyutlu bir alanda gösterimi

$$\frac{\partial u_{i,j}^2}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \quad (3.19)$$

$$\frac{\partial u_{i,j}^2}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2}. \quad (3.20)$$

İki boyutlu Laplace denklemi aşağıdaki gibi yazılabilir:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \quad (3.21)$$

Denklem 3.19 ve 3.20, denklem 3.21'de yerine koyulursa;

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = 0, \quad (3.22)$$

denklemini elde edilir. Eğer $h = k$ olursa denkleminiz aşağıdaki hali alır:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0, \quad (3.23)$$

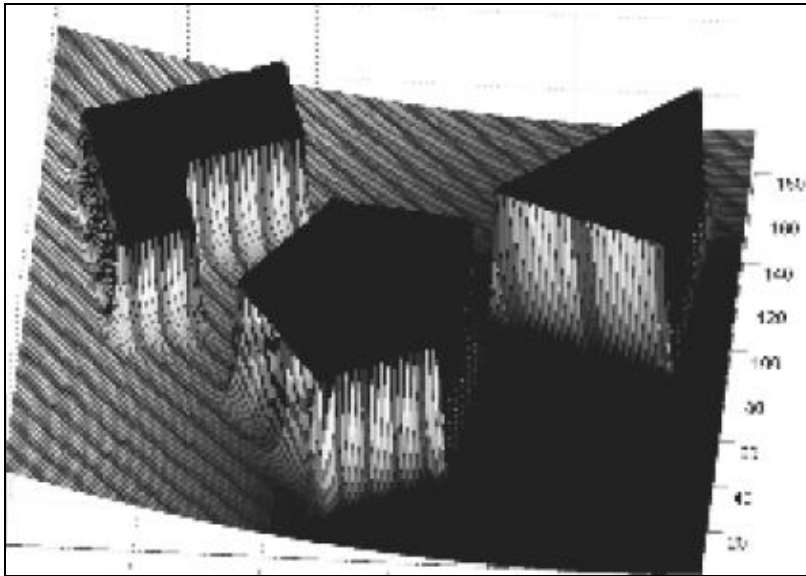
veya

$$u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}}{4}. \quad (3.24)$$

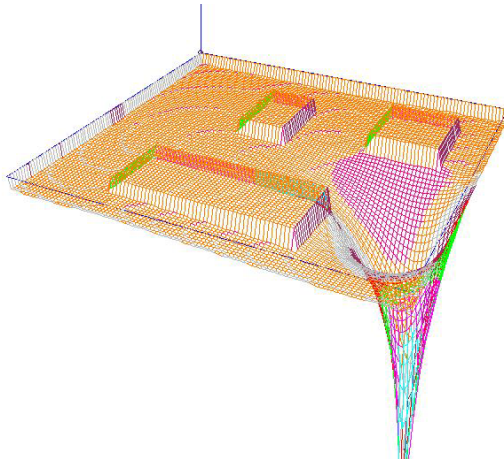
Denklemin 3.24'e potansiyel alan denkleminin denilmektedir. Eğer potansiyel alanındaki sınır değerleri biliniyorsa buna Dirichlet sınır koşulları altındaki potansiyel alan denkleminin denilmektedir.

3.1.5. Potansiyel alanın çalışma mantığı ve hesaplanması

Potansiyel alan denkleminin tüm çalışma alanına uygulanmaktadır. Bu uygulama sonucunda hedef çukurda kalmaktadır, engeller tepe etkisi yaratmaktadır ve alanda hedefe doğru bir eğim oluşmaktadır (Şekil 3.4, Şekil 3.5).



Şekil 3.4 Potansiyel alanın üç boyutlu görüntüsü (Siegwart ve Nourbaksh, 2004)

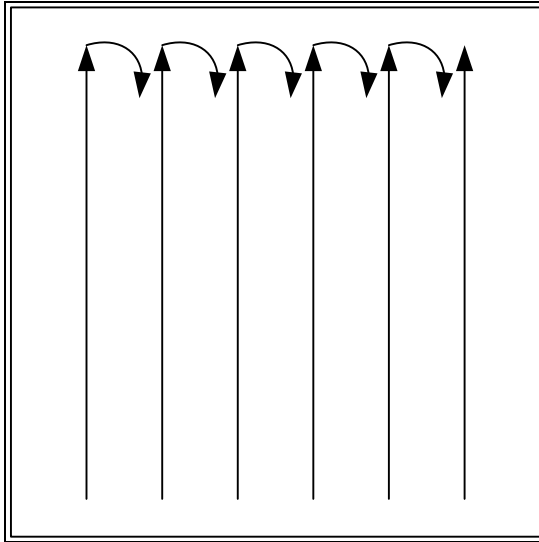


Şekil 3.5 Potansiyel alanın üç boyutlu görüntüsü (WEB_9. 2009)

Çalışma alanını çevreleyen grid noktalarına ve engelleri temsil eden grid noktalarına sıfır değeri, hedef noktasına ise -2^{124} değeri verilmektedir. En dıştaki grid noktaları, engellere ait grid noktaları ve hedefe ait grid noktası dışındaki grid noktaları potansiyel alan denklemi ile hesaplanmaktadır. Bir iterasyon yapıldığında bu bahsedilen grid noktaları bir defa hesaplanmaktadır. İterasyon sayısı kadar grid noktaları hesaplanmaktadır.

$$u_{i,j}^{m+1} = \frac{1}{4}(u_{i-1,j}^m + u_{i+1,j}^m + u_{i,j-1}^m + u_{i,j+1}^m) \quad (3.25)$$

Denklem 3.25'deki m iterasyon sayısını göstermektedir. Şekil 3.6'da da alanın hesaplanma şekli gösterilmektedir. Şekil 3.6'da aşağıdan yukarıya doğru hesaplama gösterilmiştir. Hesaplama yönü yukarıdan aşağıya, soldan sağa veya sağdan sola şeklinde de seçilebilir. Eğer alanın indis değerleri $(0,0)$ 'dan başlıyorsa (i,j) indisi $(1,1)$ 'den başlamaktadır. Benzer şekilde alanın en büyük indis değerleri (n,n) ise $(n-1,n-1)$ indisli grid noktasına kadar hesaplama yapılmaktadır.



Şekil 3.6 Çalışma alanının hesabı

Örnek olarak 6x6'lık bir çalışma alanında (4,3) indisli nokta hedef olarak seçilmiştir. Alanın sayısal değerleri Tablo 3.1'deki gibi ortaya çıkmaktadır.

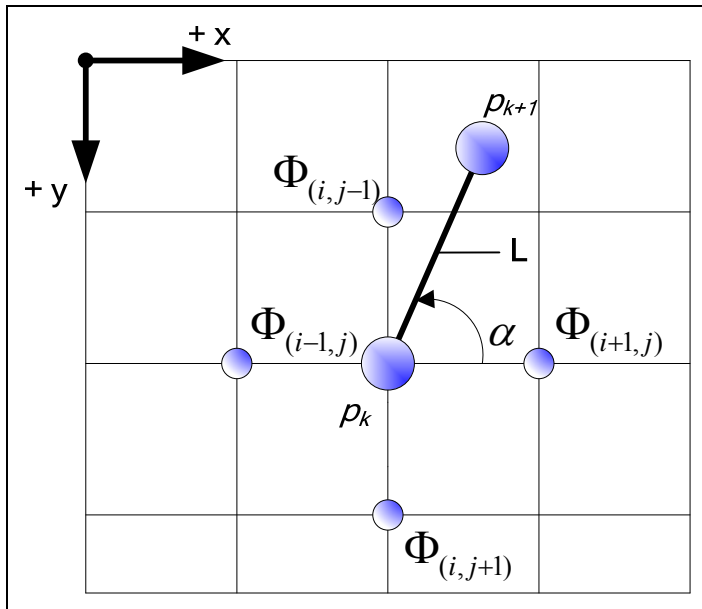
Tablo 3.1 6x6'lık bir alandaki grid noktalarının sayısal değerleri

0	0	0	0	0	0
0	-1,9E+36	-4,5E+36	-7,36E+36	-3,7E+36	0
0	-3,3E+36	-8,7E+36	-2,13E+37	-7,4E+36	0
0	-2,5E+36	-5,6E+36	-8,66E+36	-4,5E+36	0
0	-1,3E+36	-2,5E+36	-3,29E+36	-1,9E+36	0
0	0	0	0	0	0

Tablo 3.1'de sol üst köşe (0, 0) indisli nokta olarak kabul edilmiştir. x indisleri'nin sola doğru ve y indisleri'nin aşağıya doğru arttığı varsayılmıştır. Hedefin değeri -2,13E+37 şeklinde çıkmıştır. Bu değer yaklaşık olarak -2^{124} değerine eşittir. Programın içerisinde hesaplama yapılırken -2^{124} değeri alınmaktadır, fakat alanın sayısal değerleri gösterilirken -2,13E+37 gibi gösterilmiştir. Bunun sebebi de Excel programında bu şekilde formatlanmasıdır. Tablo 3.1'den de görüldüğü gibi (1,5) indisli noktanın değeri(-1,3E+36), (4,4) indisli noktanın değerinden(-8,66E+36) büyüktür. Bu da daha önce anlatıldığı gibi alanda bir eğimin olduğunu göstermektedir. Hedefe yakın olan grid noktalarının değerleri diğerlerine nazaran daha düşük olmaktadır.

3.2. Mobil Robot ile Hedef Arasındaki Yolu Bulunması

Hedefin bir çukur etkisi yarattığından ve alanda hedefe doğru bir eğim olduğundan bahsedilmişti. Mobil robotu da bu eğimli alanda bir bilye gibi düşünebiliriz. Bilye eğimli bir alanda nereye konulursa konulsun çukura doğru gidecektir. Benzer şekilde mobil robot da alanda nereye konulursa konulsun hedefine varacaktır. Mobil robotun izleyeceği yolu bulabilmek için alandaki eğimden faydalanmamız gerekmektedir. Şekil 3.7’de yönün nasıl belirlendiği gösterilmektedir.



Şekil 3.7 Mobil robotun yönünün belirlenmesi (Çonkur, 2005)

Şekil 3.7’deki α ’nın değerinin bulunması aşağıdaki gibi olmaktadır.

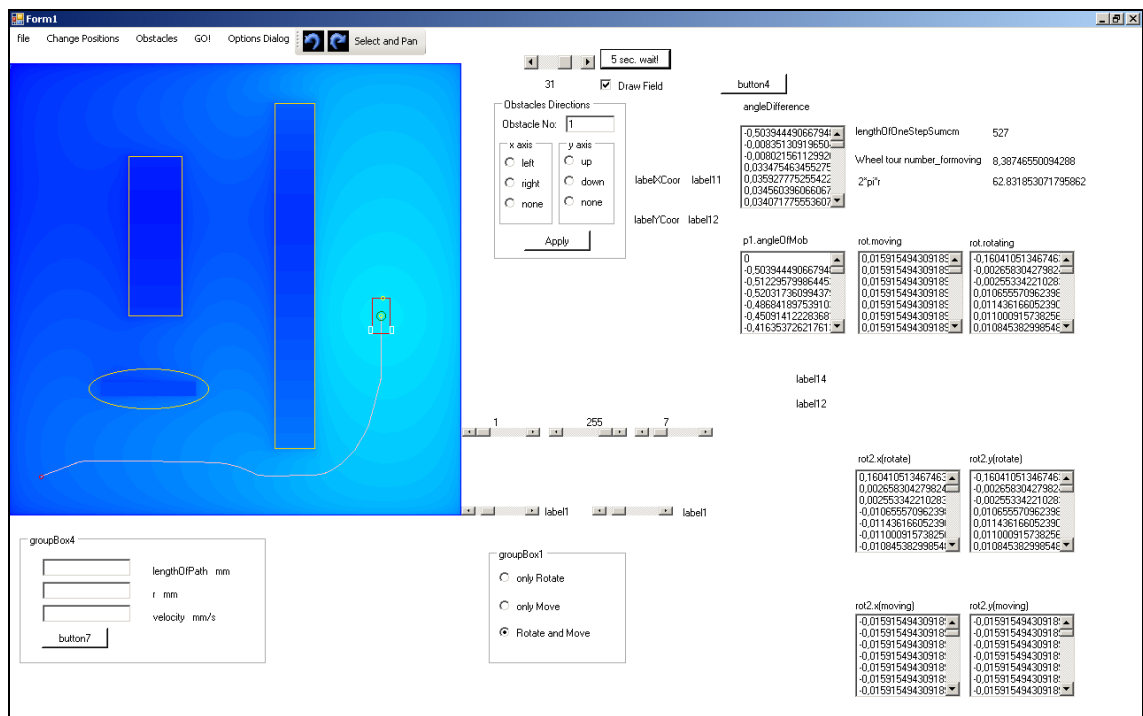
$$\alpha = a \tan 2 \left(\frac{\Phi_{(i,j-1)} - \Phi_{(i,j+1)}}{\Phi_{(i-1,j)} - \Phi_{(i+1,j)}} \right) \quad (3.26)$$

Bulunan α açısı ile yön bulunmaktadır ve bu yön doğrultusunda L boyunda gidilerek p_{k+1} noktası elde edilmektedir. Buradaki p_{k+1} noktasını bulmadaki işlem hedefe erişilinceye kadar tekrarlanmaktadır. Bulunan p_{k+1} noktası genelde grid noktasının üzerine düşmemektedir. Bu yüzden aralardaki potansiyel alan değerlerine de ihtiyaç duyulmaktadır. Bu değerler de lineer interpolasyon yoluyla elde edilmektedir. L boyuna

bağlı olarak bulunan yol kaba veya yumuşak olmaktadır. L boyu kısa tutulursa yol daha yumuşak bir şekilde çıkmaktadır.

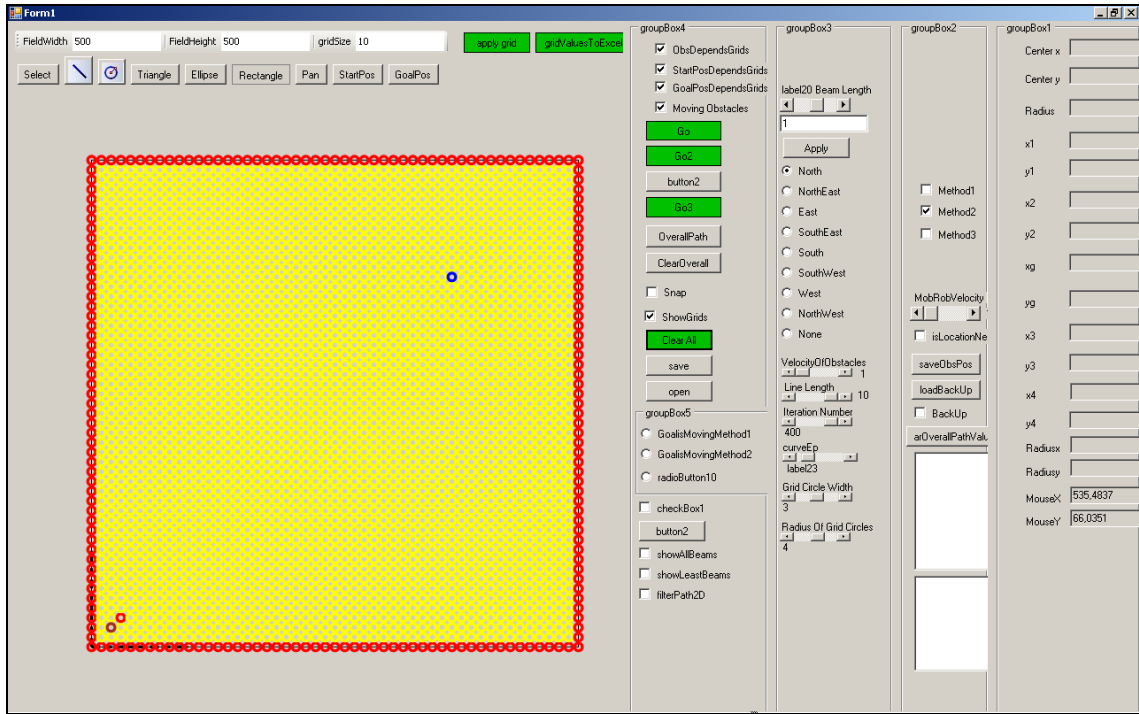
3.3. Mobil Robot için Yazılan İki Adet Program

Mobil robot için iki adet program yazıldığından daha önce bahsedilmişti. Şimdi bu programları anlatacağız. Bu programlardan bundan sonra Program1 ve Program2 olarak bahsedilecektir. İlk önce yazılan program Program1 ve daha sonra yazılan da Program2 olarak isimlendirilmiştir.



Şekil 3.8 Mobil robot için yazılan ilk programın arayüzü (Program1)

Program1'in arayüzü Şekil 3.8'de, program2'nin arayüzü de Şekil 3.9'da verilmiştir. Program2, Program1'in daha gelişmiş bir versiyonudur. Program2'ye engellerin ekrana çizilmesini ve gerektiğinde değişiklikler yapmayı sağlayan gelişmiş bir çizim kısmı eklenmiştir. Ayrıca program2 optimize edilmiş ve kod kısmı daha düzenli ve daha profesyonelce yazılmıştır.



Şekil 3.9 Mobil robot için yazılan ikinci programın ara yüzü (Program2)

3.3.1. Program1: Mobil robot için yazılan ilk program

Program1'in ana menüleri şu şekildedir; *File*, *Change Positions*, *Obstacles*, *GO* ve *Options Dialog*'tur (Şekil 3.10).

File menüsünün altında *Open* ve *Save* alt menüleri bulunmaktadır (Şekil 3.11).



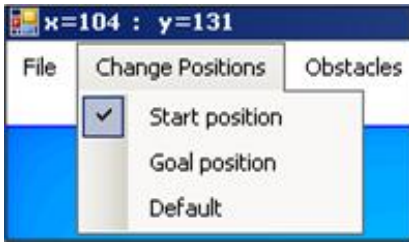
Şekil 3.10 Program1 menüleri

Save menüsü mobil robotun başlangıç konumunu, hedefin konumunu ve engellerin konumlarını mobilrob uzantılı dosyalara kaydetmeye yaramaktadır. *Open* menüsü ile de kaydedilmiş dosyalar açılabilir.



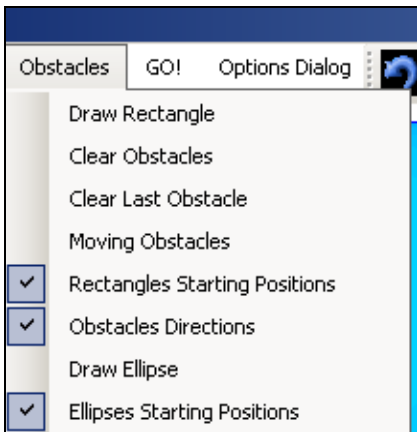
Şekil 3.11 File menüsü

Change Positions menüsünün altında *Start position*, *Goal position* ve *Default* alt menüleri bulunmaktadır (Şekil 3.12).



Şekil 3.12 Change Positions menüsü

Start Position menüsü ile mobil robotun başlangıç konumu değiştirilebilir. *Goal Position* menüsü ile hedefin konumu değiştirilebilir. *Default* menüsü ile mobil robotun geçtiği yol silinmektedir ve hareketli engeller ilk konumlarına geri dönmektedir.

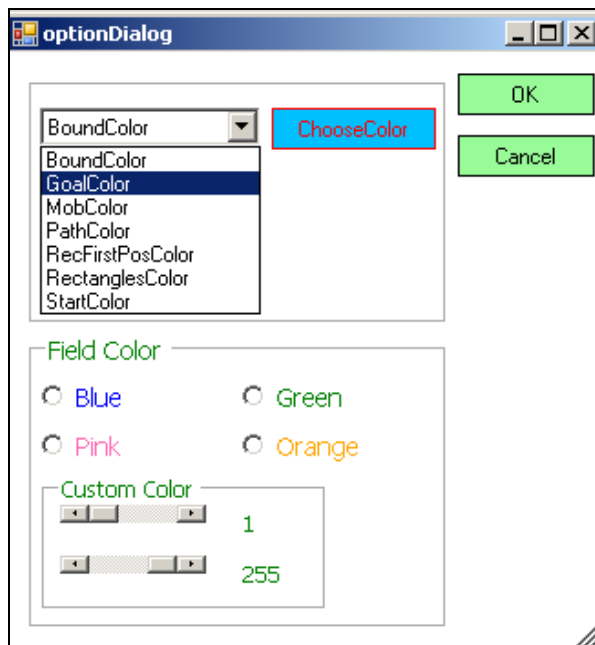


Şekil 3.13 Obstacles menüsü

Obstacles menüsünün altında *Draw Rectangle*, *Clear Obstacles*, *Clear Last Obstacle*, *Moving Obstacles*, *Rectangles Starting Positions*, *Obstacles Directions*, *Draw Ellipse* ve *Ellipses Starting Positions* alt menüleri bulunmaktadır (Şekil 3.13).

Draw Rectangle menüsü ile dikdörtgen engeller çizilebilmektedir. *Clear Obstacles* menüsü ile hem tüm engeller hem de mobil robotun geçtiği yol silinebilmektedir. *Clear Last Obstacle* menüsü ile son çizilen engel silinmektedir. *Moving Obstacles* menüsü ile engellerin hareketli olup olmayacağı belirlenmektedir. *Rectangles Starting Positions* menüsü ile dikdörtgen şeklindeki engellerin başlangıç konumlarının gözükmüp gözükmeyeceği belirlenmektedir. *Obstacle Directions* menüsü ile *Obstacle Directions* grubunun gözükmüp gözükmeyeceği ayarlanmaktadır. *Draw Ellipse* menüsü ile elips şeklinde engeller çizilebilmektedir. *Ellipses Starting Positions* menüsü ile elips şeklindeki engellerin başlangıç konumlarının gözükmüp gözükmeyeceği belirlenmektedir.

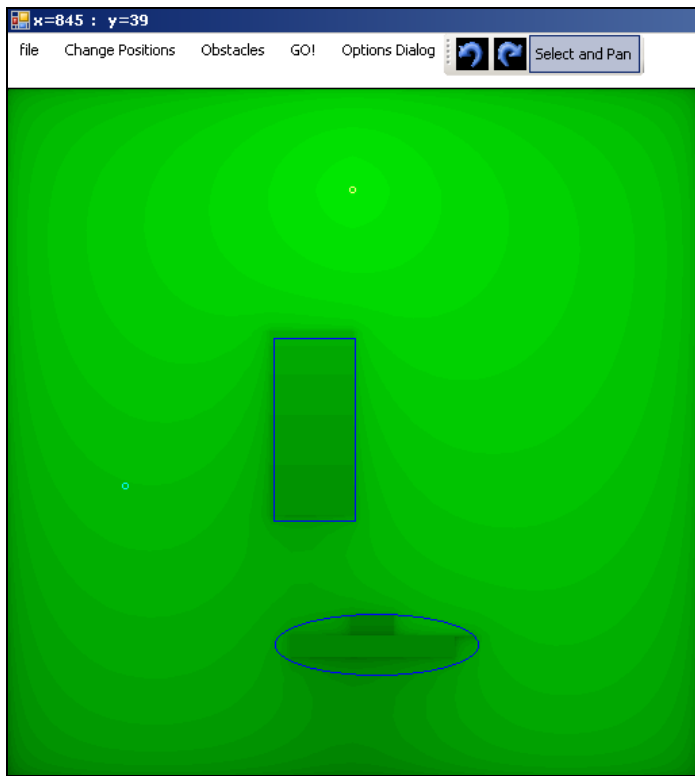
Go menüsü ile mobil robotun hareketi başlamaktadır. Aynı zamanda eğer hareketli engeller varsa onlar da hareket etmektedirler.



Şekil 3.14 OptionDialog Formu

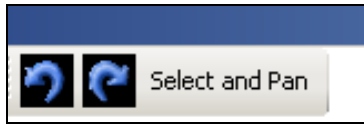
Options Dialog menüsüne tıklayınca *optionDialog* isimli form açılmaktadır (Şekil 3.14). Bu form ile alanı sınırlayan çizgilerin, hedef noktasının, mobil robotun ortasındaki dairenin, mobil robotun yolunun, engellerin başlangıçtaki konumlarını gösteren çizgilerin, engellerin ve mobil robotun başlangıç noktasının rengi ayarlanabilmektedir. Ayrıca *Field Color*'ın altındaki kısım ile alanın rengi mavi, yeşil, pembe veya turuncu yapılabilir. *Custom Color*'ın altındaki kaydırma çubukları ile alanın rengi ara renklerden yapılabilir.

Alanın rengi yeşil olarak seçildiğinde Şekil 3.15'teki gibi bir görüntü de olmaktadır. Şekil 3.15'e bakıldığında alanın görüntüsünde sanki izohips çizgileri gibi çizgilerin çıktığı fark edilecektir. İzohipsler eş yükselti için kullanılmaktadır. Burada da benzer şekilde aynı çizgi üzerindeki noktaların eşit değerlere sahip olduğunu söyleyebiliriz. Alan potansiyel alan denklemi sayesinde sayısal değerlerden oluşmaktadır. Hedefin en küçük değere, engellerin ve dış çerçevenin en yüksek değere sahip olduğundan bahsedilmişti. Alandaki değerler bu değerler arasında yer almaktadır. Ayrıca engellerin tepe ve hedefin çukur gibi düşünülebileceğinden bahsedilmişti. Çukur yani hedef en açık renkte çıkmaktadır ve tepeler yani engeller en koyu renkte çıkmaktadırlar. Diğer alan kısımları ise ara tonlarda çıkmaktadırlar.



Şekil 3.15 Alanın yeşil renkteki görüntüsü

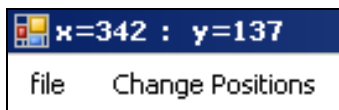
Options Dialog menüsünün hemen yanında Geri, İleri ve *Select and Pan* düğmeleri bulunmaktadır (Şekil 3.16).



Şekil 3.16 Geri, İleri ve Select and Pan butonları

Geri ve İleri butonları engellerin çizimiyle alakalıdır. Örneğin üç adet engel çizilirse 3 kere geri alabilme imkânı vardır. Geri alındığı kadar ileri de alınabilmektedir. Örneğin 3 defa geri alındığı durumda 3 defa da ileri alınabilmektedir. Engel çizildiği sürece ileri butonu aktif olmamaktadır. Geri butonu da geri alınabilecek bir durum olmadığında aktif olmamaktadır. Tüm engeller silindiğinde hem geri butonu hem de ileri butonu aktif olmamaktadır. Hem geri hem de ileri butonları aktif iken yeni engel çizilirse ileri butonu inaktif olmaktadır yani ileriye almak için kayıtlı engeller hafızadan atılmaktadır.

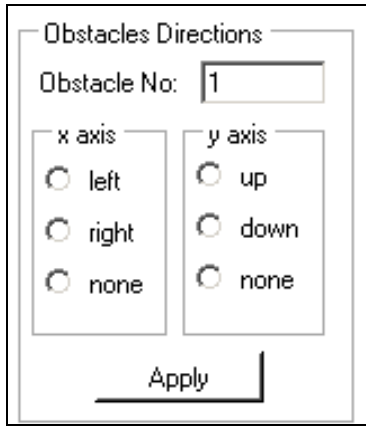
Select and Pan butonu aktif iken engeller üzerlerine fare ile gelinerek taşıma yapılabilir.



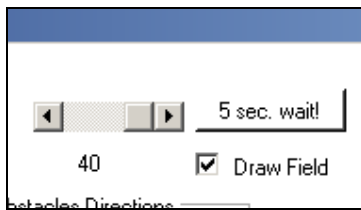
Şekil 3.17 Fare'nin koordinatlarının gösterimi

Şekil 3.17'deki gibi formun sol üst köşesinde fare'nin koordinatları gösterilmektedir. Fare'nin x koordinatı sağa doğru artmaktadır ve y koordinatı aşağıya doğru artmaktadır. (0, 0) noktası normalde formun sol üst köşesidir fakat burada alanın sol üst köşesine kaydırılmıştır.

Şekil 3.18'deki gösterilen *Obstacles Directions* kısmı ile engellerin hangi yönde hareket edeceği belirlenmektedir. Engellerin hareketleri 45°'lik açı hassasiyeti ile yapılabilir. Örneğin x ekseninde sol ve y ekseninde yukarı seçilirse engelin hareketi sol-yukarı çapraz şeklinde olmaktadır. Olmayan bir engel numarası girilip butona tıklanırsa hiçbir şey yapılmamaktadır. Eğer rakam dışında bir şey girilirse "Enter a number"(Bir sayı giriniz) şeklinde uyarı gelmektedir.



Şekil 3.18 Obstacles Directions kutusu

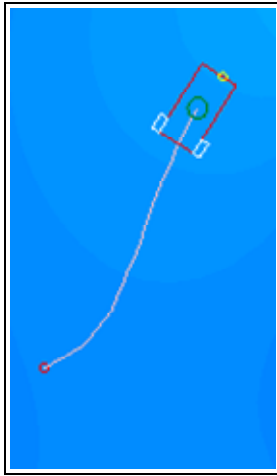


Şekil 3.19 Çeşitli fonksiyonlar

Şekil 3.19'daki kaydırma çubuğu mobil robotun bir seferde kaç nokta ilerleyeceğini göstermektedir. Simülasyonda mobil robotun hangi aralıklarla çizdirileceği ve hareket edeceği kaydırma çubuğu ile belirlenmektedir. Yolun noktalardan oluştuğunu belirtmiştik, mobil robot bu noktalar üzerinde ilerlemektedir. Şekil 3.19'da kaydırma çubuğunun değeri 40 olarak görülmektedir. Mobil robot 40 nokta giderken engeller eğer hareketli iseler 15 pixel veya $15\sqrt{2} \text{ pixel}$ hareket etmektedirler. Burada yoldaki her nokta arası 1 pixel olduğundan mobil robot da 40 pixel hareket etmektedir diyebiliriz.

Şekil 3.19'daki *Draw Field* onay kutusu ile her hareketten sonra alanın çizdirilip çizdirilmeyeceği belirlenmektedir. Eğer onay kutusu seçili değilse alanın bitmap çizimi sadece güncellenmemektedir. Mobil robot, mobil robotun yolu ve engellerin hareketi güncellenmektedir. Şekil 3.19'daki *5 sec. wait* butonu ise mobil robotun ve engellerin 5 saniye duraklamasını sağlamaktadır. Bu sayede o anki durum daha rahat bir şekilde incelenebilmektedir. Programda mobil robotun tekerleklerinin döndürülmesi ile ilgili kontroller de bulunmaktadır. Bu durum ayrıca anlatılacaktır.

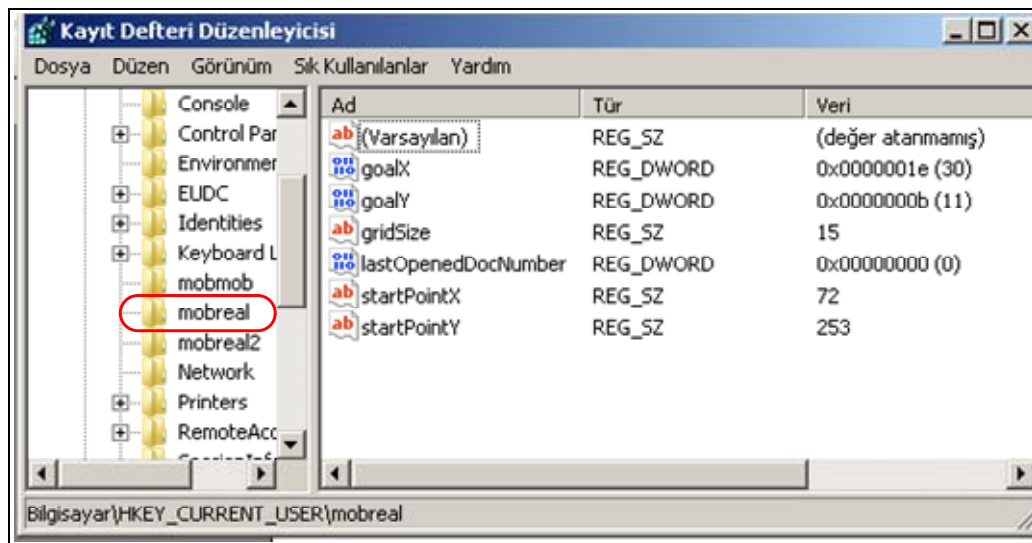
Programda mobil robotun çizimi Şekil 3.20'deki gibi olmaktadır.



Şekil 3.20 Mobil robotun programda çizimi

Şekil 3.20'den de görülebileceği gibi mobil robotun yönü gösterilmektedir. Mobil robotun yönü yolun yönü ile aynı belirlenmiştir.

Program1'deki regis sınıfının kayıt defteri işlemleri için kullanıldığını belirtmiştik. Mobil robotun başlangıç konumu ve hedefin konumu program kapatılıp açıldığında aynı kalmaktadır. Ayrıca engellerin de programın kapatıldığı andaki konumları bir dosyaya kaydedilmektedir ve program çalıştırıldığı anda engeller son konumlarında gelmektedir. Şekil 3.21'de kayıt defterine yapılan program1 ile ilgili kayıtlar gösterilmektedir. Kayıtlar mobreal adlı klasörün içine yapılmaktadır.

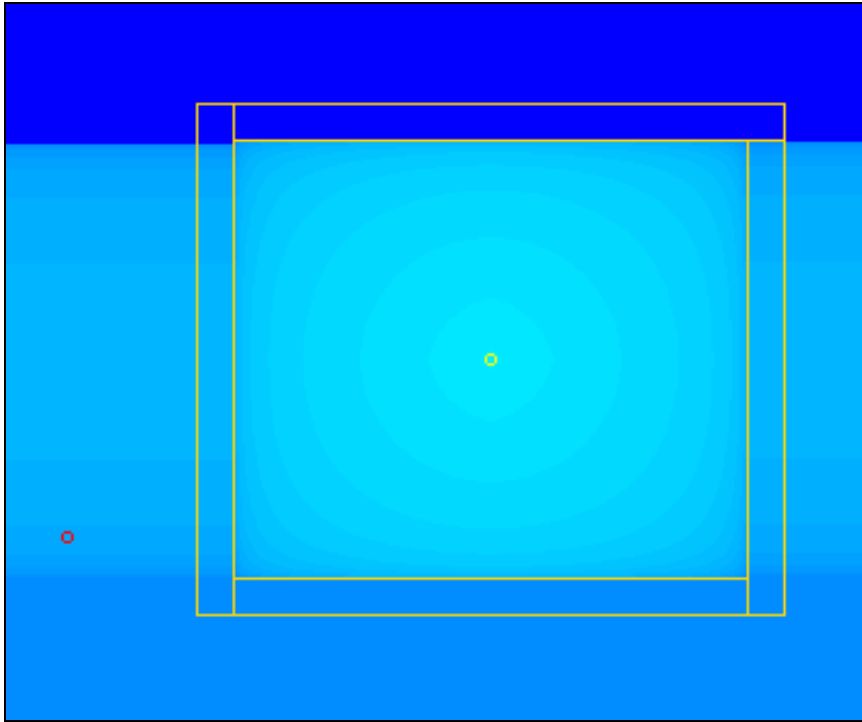


Şekil 3.21 Program1 ile ilgili kayıt defterine yapılan kayıtlar

Program1’de hedef veya başlangıç noktası çalışma alanının dışına koyulamamaktadır. Ayrıca engellerin de çalışma alanının dışına koyulması engellenmiştir.

Örneğin Şekil 3.22’de hedefe ulaşım engellerle kapatılmıştır. Böyle bir durumda “There is no solution for the current situation or an obstacle hit our mobile robot.” şeklinde bir mesaj verilmektedir. Benzer şekilde eğer bir hareketli engel mobil robota çarparsa aynı mesaj verilmektedir. Aslında Şekil 3.22’ye dikkat edilecek olursa engellerin kapattığı alanda potansiyel alanın çalıştığı fark edilecektir. Eş yükselti çizgileri Şekil 3.22’de rahatlıkla görülebilmektedir. Burada eğer başlangıç konumunu engellerin kapattığı yere koyarsak mobil robotun hedefine ulaşabildiğini görürüz (Şekil 3.23).

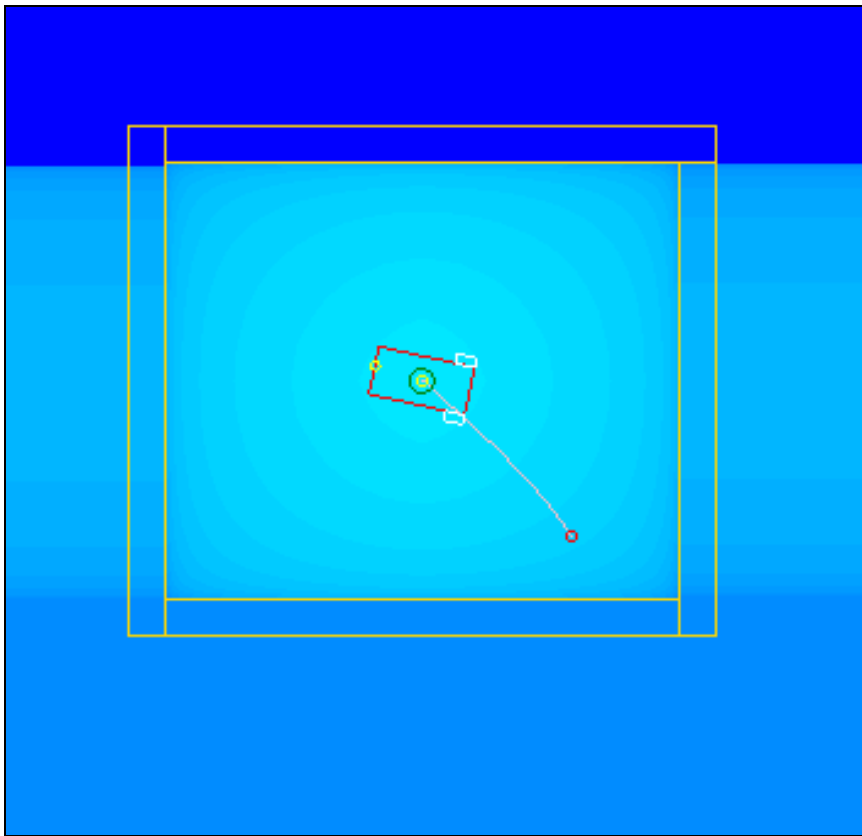
Mobil robotun hedefe yolunun olmadığını veya mobil robota bir hedefin gelip çarptığını anlamak için potansiyel alanın matematiksel yapısına bakılmıştır.



Şekil 3.22 Hedefe yolun olmaması durumu

Hedefe küçük bir sayısal değer, engellere ve alanı sınırlayan çerçeveye de hedefin değerine nazaran büyük bir değer verildiğinden bahsedilmiştir. Ara değerler ise

potansiyel alan denklemi ile iterasyon yapılarak bulunmaktadır. Eğer hedef ile alanın sınırları arasında bir geçiş yeri olmazsa ara değerler sınır ile engeller arasında oluşmamaktadır. Sınır ve engellere aynı sayısal değer verildiğinden bahsedilmişti. Aynı değerlere sahip olmalarından dolayı sınır ile engeller arası da engeller ile aynı sayısal değerlere sahip olmaktadır. Şekil 3.22'deki görüntü bu duruma örnek olarak verilebilir. Bu yüzden mobil robotun etrafında grid değerlerine bakılarak hedefe yolun olup olmadığına karar verilebilir. Benzer şekilde mobil robota engelin çarpıp çarpmadığı da mobil robotun etrafındaki grid değerlerinden anlaşılabilir.



Şekil 3.23 Başlangıç konumunun hedef ile aynı kısma konulduğu durum

3.3.2. Program2: Mobil robot için yazılan ikinci program

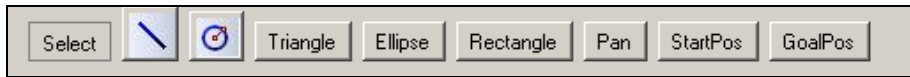
Program2'ye ilk başta bir önceki programdaki engel çizimini geliştirme amacıyla başlanmıştır. Program1'de sadece engel olarak dikdörtgen çizimi yapılabilmektedir. Daha sonra elips çizimi de Program1'e eklenmiştir. Program1'de dikdörtgen veya elips şeklindeki engeller içlerine tıklayıp taşınabilmektedir. Bu durumdan dolayı eğer engel hızlı bir şekilde taşınmak istenirse fare engelin içinden çıkabilmektedir ve engel taşınmamaktadır. Program2'de bu durum düzeltilmiştir.

Program2'nin kontrollerini açıklayalım:

Şekil 3.24'te görülen kısımlar ile alanın boyutları ayarlanabilmektedir. *FieldWidth* ile alanın genişliği ayarlanmaktadır. *FieldHeight* ile alanın yüksekliği ayarlanmaktadır. *GridSize* ile ızgara boyu ayarlanmaktadır.

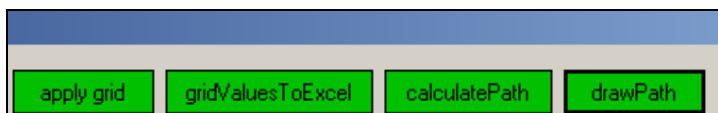


Şekil 3.24 Alanın boyutlarının ayarlanması



Şekil 3.25 Engellerin çizimi, başlangıç ve hedef konumlarının ayarlanması

Şekil 3.25'te gözüken *Select* kontrolü ile engeller seçilebilmektedir. Engeller seçildikten sonra engellerin boyutları değiştirilebilmektedir. Ayrıca engeller silinebilmektedir veya engellerin yerleri değiştirilebilmektedir. Çizgi resimli kontrol ile çizgi şeklinde engeller çizilebilmektedir. Daire resimli kontrol ile daire şeklinde engeller çizilebilmektedir. Benzer şekilde *Triangle*, *Ellipse* ve *Rectangle* kontrolleri ile de sırasıyla üçgen, elips ve dikdörtgen şeklinde engeller çizilebilmektedir. *Pan* kontrolü ile alanın yeri kaydırılabilmektedir. *StartPos* ile başlangıç konumu, *GoalPos* ile de hedef konumu değiştirilebilmektedir. Ayrıca fare'nin tekeri ile alana yakınlaştırma veya alandan uzaklaşma yapılabilmektedir.



Şekil 3.26 Form kontrolleri

Şekil 3.26'da gözüken *apply grid* butonu Şekil 3.24'te gözüken ayarların uygulanması için kullanılmaktadır. *GridValuesToExcel* butonu ile alanın grid değerleri Excel dosyasına atılabilmektedir. *CalculatePath* butonu ile alanın hesabı yaptırılabilir. *DrawPath* ile hedefe olan yol çizdirilmektedir. Bu kontroller ilk

başta çok kullanılmıştır fakat daha sonra alanı hesaplama, yolu çizdirme ve mobil robotun hareketi tek kontrol ile sağlanmıştır.

Şekil 3.27'deki kontrolleri açıklayalım. *ObsDependsGrids* kontrolü ile engellerin çizilirken gridlere bağımlı mı yoksa bağımsız mı olacağı ayarlanmaktadır. Benzer şekilde *StartPosDependsGrids* ve *GoalPosDependsGrids* kontrolleri de sırasıyla başlangıç konumu ve hedef konumu belirlenirken gridlere bağımlı mı yoksa bağımsız mı olacağı ayarlanmaktadır.

The image shows a control panel with the following elements:

- ObsDependsGrids
- StartPosDependsGrids
- GoalPosDependsGrids
- Moving Obstacles
- Go (green button)
- Go2 (green button)
- Start (grey button)

Şekil 3.27 Form kontrolleri

Moving Obstacles kontrolü ile engellerin hareketli mi yoksa hareketsiz mi olacağı ayarlanmaktadır. Bu kontrol işaretli olmadan engeller hareket edememektedirler. *Go* butonu başlangıç konumundan hedefe olan yolu buldurup çizdirmektedir. *Go2* butonu ise butona her tıklayıpta mobil robotun belirlenen hızına göre bir sonraki konumu çizdirilmektedir. Mobil robot aslında hareketi sırasında tüm yolun üzerinden geçerek hedefe ulaşması gerekmektedir fakat simülasyonda mobil robotun hızına göre belli aralıklarda çizdirilmektedir. *Start* butonu ile mobil robotun başlangıçtan hedefe olan hareketi sağlanmaktadır.

Şekil 3.28'de gözüken *OverallPath* butonu ile mobil robotun başlangıç konumundan hedefe olan yolu çizdirilmektedir. Mobil robotun hareketi sırasında yeni konumundan hedefe olan yol çizdirilmektedir. Hedefin konumu değiştirilip çalıştırıldıktan sonra *OverallPath* butonuna basılırsa mobil robotun önceki yolları da çizilmektedir. Eğer sadece o durumdaki gittiği yol gösterilmek isteniyorsa *ClearOverall* butonuna basıldıktan sonra *OverallPath* butonuna basılması gerekmektedir. *Snap* kontrolü ile

engellerin birbirlerini uç noktalardan veya merkezden yakalamaları sağlanmaktadır. *ShowGrids* kontrolü grid noktalarının gösterilmesini veya gösterilmemesini sağlamaktadır. *Clear All* butonu ile tüm engeller silinebilmektedir. *Save* butonu ile engeller, başlangıç konumu, hedef konumu, iterasyon sayısı, engellerin hızı ve mobil robotun hızı .mobilrob2 uzantılı bir dosyaya kaydedilmektedir. *Save* butonu ile kaydettiğimiz dosyaları *Open* butonu ile açabiliriz.

The image shows a software control panel with the following elements:

- Buttons: OverallPath, ClearOverall, Clear All (highlighted in green), save, open.
- Checkboxes: Snap (unchecked), ShowGrids (checked).
- GroupBox5 containing radio buttons: GoalisMovingMethod1, GoalisMovingMethod2, GoalisntMoving.
- Button: Show.
- Checkboxes: showAllBeams, showLeastBeams, filterPath2D.

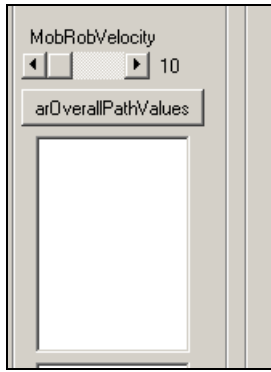
Şekil 3.28 Form kontrolleri

Program1'e nazaran Program2'deki save işlevi daha fazla özelliğe sahiptir. *GoalisMovingMethod1* ve *GoalisMovingMethod2* kontrolleri hedefi belirlenmiş bir yolda hareket ettirmektedir. *GoalisntMoving* kontrolü ise hedefin hareketsiz olmasını sağlamaktadır. *Show* butonu ile hemen altındaki kontrollerden hangileri seçili ise onlar gösterilmektedir. *ShowAllBeams* kontrolü tüm ışınların, *showLeastBeams* kontrolü engeller ile yol arasındaki en kısa mesafeli ışınların ve *filterPath2D* kontrolü de yolun yumuşatılmış halinin gösterilmesini sağlamaktadır.

Şekil 3.29’da *Beam Length* olarak yazan kaydırma çubuğu ile ışınların boyları ayarlanabilmektedir. Işınlar dik olarak atılmaktadır ve engele çarptığında daha ileriye gidememektedir. *North*, *NorthEast*, *East* şeklinde giden kısım ile engellerin yönü ayarlanmaktadır. Text kutusuna da uygulanacak engelin numarası yazılmaktadır. *VelocityOfObstacles* kaydırma çubuğu ile engellerin hızları belirlenmektedir. *Line Length* ile yoldaki noktalar arası uzaklık belirlenmektedir. Ayrıca *Iteration Number* ile iterasyon sayısı, *curveEp* ile yumuşatma katsayısı, *Grid Circle Width* ile grid dairelerinin kalınlığı ve *Radius Of Grid Circles* ile de grid dairelerinin yarıçapları ayarlanmaktadır.

The image shows a vertical panel of form controls. At the top, there is a slider labeled "Beam Length" with a value of 40. Below it is a text input field containing the number "1" and an "Apply" button. A list of radio buttons follows, with "North" selected. Below the radio buttons are several numeric spinners: "VelocityOfObstacles" (value 2), "Line Length" (value 4), "Iteration Number" (value 203), "curveEp" (value 5), "Grid Circle Width" (value 0), and "Radius Of Grid Circles" (value 1).

Şekil 3.29 Form kontrolleri



Şekil 3.30 Form kontrolleri

Şekil 3.30'daki *MobRobVelocity* kaydırma çubuğu ile mobil robotun simülasyonda bir seferde kaç nokta gideceği ayarlanmaktadır. *ArOverallPathValues* butonuna basıldığında hemen altındaki liste kutusuna ve yeni bir Excel dosyasına hedefe olan yoldaki noktaların koordinatlarını yazdırmaktadır (Şekil 3.31).

	A	B
1	95,0000000000000	45,0000000000000
2	95,3606558707451	50,9891508031521
3	95,3083305123897	56,9889226368865
4	94,6351315037053	62,9510366155610
5	93,4381526247361	68,8304278734466
6	91,6040884360971	74,5432387885700
7	89,6388627845566	80,2122676417799
8	88,6160448219818	86,1244451945463
9	88,1574683930380	92,1068951658220
10	87,8927494657935	98,1010526453648
11	87,8478752975427	104,1008848354370
12	88,1701283990070	110,0922246637720
13	88,8305596314851	116,0557664458980
14	90,0306885393986	121,9345155138880
15	91,0923030353668	127,8398499381140
16	92,6480807881260	133,6346368698330
17	94,8492326462931	139,2162970616840
18	97,6483399675284	144,5233676476290
19	100,8384109255920	149,6050452527630

Şekil 3.31 Yola ait noktaların koordinatları

Şekil 3.31'deki değerler ile iki nokta arasındaki formül kullanılırsa bizim belirlediğimiz *Line Length* değerini elde etmiş oluruz. Örneğin Şekil 3.31'deki Excel dosyasındaki gözükten 2. nokta ve 1. nokta koordinatlarına denklem 3.27'deki formülü uygulayalım. Sonucun *Line Length* için belirlediğimiz olan 6 değeri çıkacaktır. Benzer şekilde eğer 2. ve 3. nokta arasına 3.27 eşitliğini uygularsak yine aynı değer çıkacaktır.

$$\sqrt{(95.3606558707451 - 95)^2 + (50.9891508031521 - 45)^2} = 6 \quad (3.27)$$

Şekil 3.32’de koordinatların gösterimi gözükmemektedir. *MouseX* ve *MouseY* fare’nin x ve y koordinatlarını göstermektedir. Visual C#’ta x koordinatı sağa doğru ve y koordinatı aşağıya doğru artmaktadır. Geometride ise y koordinatı yukarıya doğru artmaktadır. Program2’de y koordinatı yukarıya doğru artacak şekilde ayarlama yapılmıştır. Şekil 3.32’de gözükken diğer etiketler ise engellerin koordinat gösterimini sağlamaktadır. Engellerin şekillerine göre farklı değerler gösterilmektedir. Dairede merkezinin koordinatları ve yarıçapının değeri, çizgide iki uç noktası ve merkez noktası, dikdörtgende dört uç noktası ve merkez noktası, üçgende üç uç noktası ve merkez noktası, elipste x yarıçapı, y yarıçapı ve merkez noktası koordinat gösteriminde kullanılmaktadır.

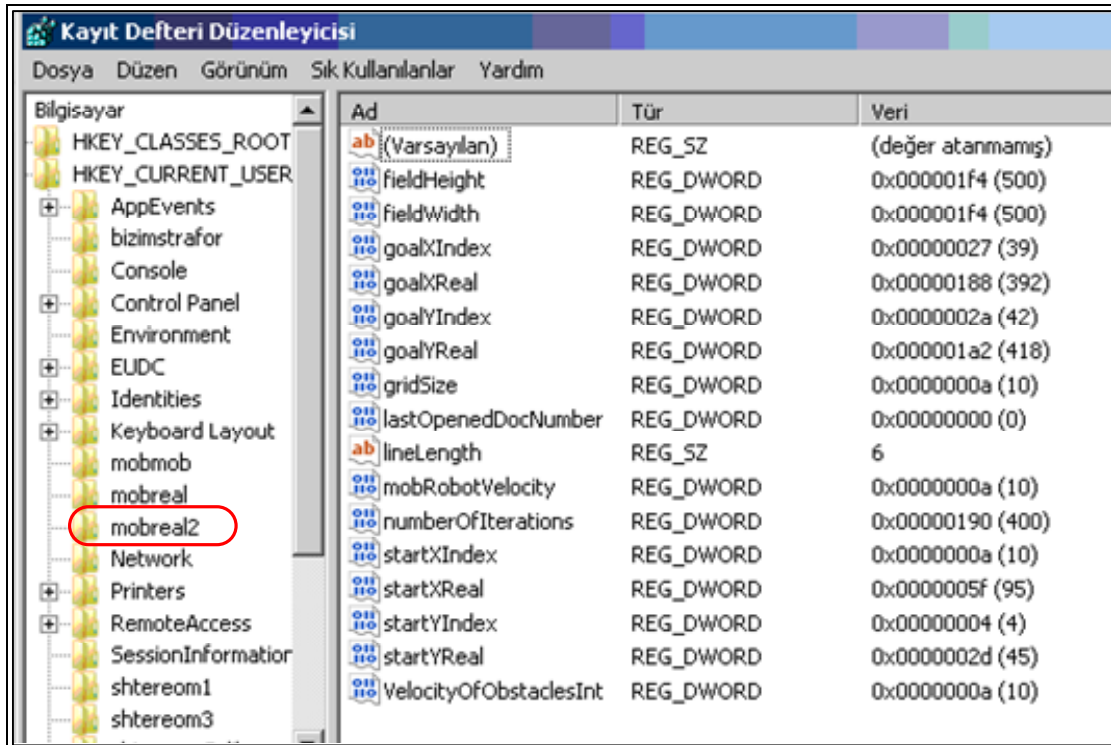
Center x	<input type="text"/>
Center y	<input type="text"/>
Radius	<input type="text"/>
x1	<input type="text"/>
y1	<input type="text"/>
x2	<input type="text"/>
y2	<input type="text"/>
xg	<input type="text"/>
yg	<input type="text"/>
x3	<input type="text"/>
y3	<input type="text"/>
x4	<input type="text"/>
y4	<input type="text"/>
Radiusx	<input type="text"/>
Radiusy	<input type="text"/>
MouseX	101,2337
MouseY	323,3594

Şekil 3.32 Koordinatların gösterimi

Şekil 3.32’de gösterilen etiketlerin kullanımı engel şekline göre değişmektedir. Daire de *Center x*, *Center y* ve *Radius*, çizgide *x1*, *y1*, *x2*, *y2*, *xg* ve *yg*, dikdörtgende *x1*, *y1*, *x2*, *y2*, *x3*, *y3*, *x4*, *y4*, *xg* ve *yg*, üçgende *x1*, *y1*, *x2*, *y2*, *x3*, *y3*, *xg* ve *yg*, elipste *Center x*, *Center y*, *Radiusx* ve *Radiusy* etiketleri kullanılmaktadır.

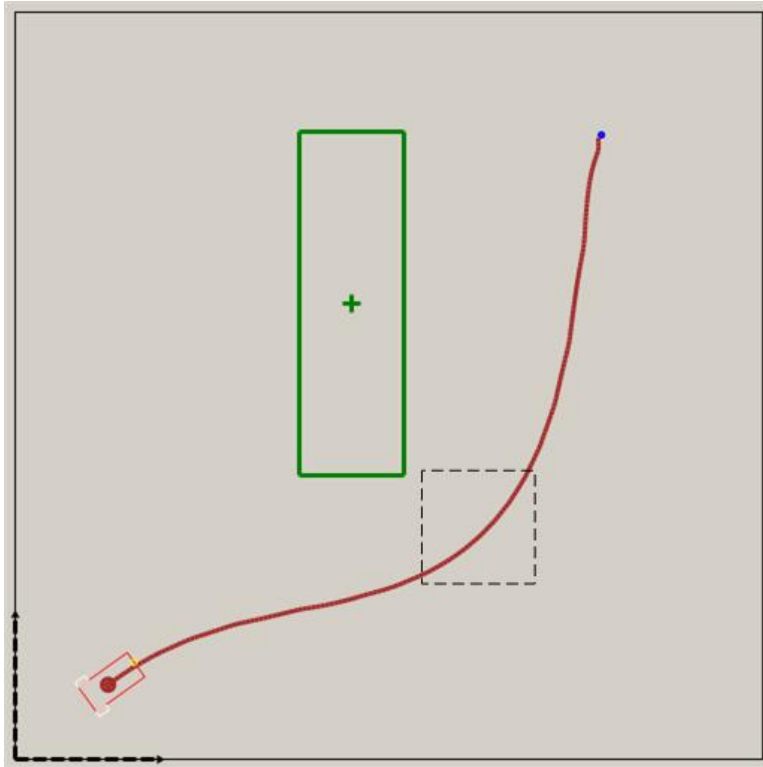
Visual C# .NET 2005 programı ile Program1 yazılmaya başlanmıştır. Daha sonra Visual C# .NET 2008 programı çıkınca bizim programımız bu yeni sürüme çevrilmiştir ve yeni sürümde yazılmaya devam edilmiştir. Visual Studio .NET 2008 sürümü ile birlikte gelen LINQ da engellerin koordinatlarının gösteriminde sorgu için kullanılmıştır.

Şekil 3.33’te Program2’de kayıt defterine yapılan kayıtlar gözükmemektedir. Bu kayıtlar sayesinde programın en son kapatıldığı haliyle gelmesi sağlanmaktadır. Ayrıca engellerin konumları da programın en son kapatıldığı haliyle gelmesi için bir dosyadan alınmaktadır. Program2’de yapılan kayıt defteri kayıtlarının Program1’de yapılanlara göre daha fazla olduğu Şekil 3.33’ten görülmektedir.



Ad	Tür	Veri
(Varsayılan)	REG_SZ	(değer atanmamış)
FieldHeight	REG_DWORD	0x000001f4 (500)
FieldWidth	REG_DWORD	0x000001f4 (500)
goalXIndex	REG_DWORD	0x00000027 (39)
goalXReal	REG_DWORD	0x00000188 (392)
goalYIndex	REG_DWORD	0x0000002a (42)
goalYReal	REG_DWORD	0x000001a2 (418)
gridSize	REG_DWORD	0x0000000a (10)
lastOpenedDocNumber	REG_DWORD	0x00000000 (0)
lineLength	REG_SZ	6
mobRobotVelocity	REG_DWORD	0x0000000a (10)
numberOfIterations	REG_DWORD	0x00000190 (400)
startXIndex	REG_DWORD	0x0000000a (10)
startXReal	REG_DWORD	0x0000005f (95)
startYIndex	REG_DWORD	0x00000004 (4)
startYReal	REG_DWORD	0x0000002d (45)
VelocityOfObstaclesInt	REG_DWORD	0x0000000a (10)

Şekil 3.33 Program2 ile ilgili kayıt defterine yapılan kayıtlar



Şekil 3.34 Mobil robotun yolu



Şekil 3.35 Mobil robotun yolunun büyültülmüş hali

Şekil 3.34'te mobil robota ait yol gösterilmiştir. Bu yolun bir kısmının büyültülmüş hali Şekil 3.35'te gösterilmiştir. Şekil 3.35'ten de görüldüğü gibi yol aslında çizgilerden oluşmaktadır. Bu çizgiler küçük seçildiğinde Şekil 3.34'teki gibi eğri şeklinde yol çıkmaktadır. Mobil robotun simülasyondaki çizimi de bu çizgilerin doğrultusuna göre yapılmaktadır.

Program2'ye ait sınıflar Şekil 3.36'dan Şekil 3.43'e kadar olan şekillerde gösterilmiştir. Şekil 3.43'te program2'ye ait tüm sınıflar gösterilmiştir.

TriangleFunctions, *RectangleFunctions*, *EllipseFunctions* ve *CircleFunctions* sınıflarından bahsedelim. Bu sınıflar Şekil 3.36’da gösterilmiştir. Bu sınıflar sırasıyla üçgen, dikdörtgen, elips ve daire çizimi ile ilgili sınıflardır.



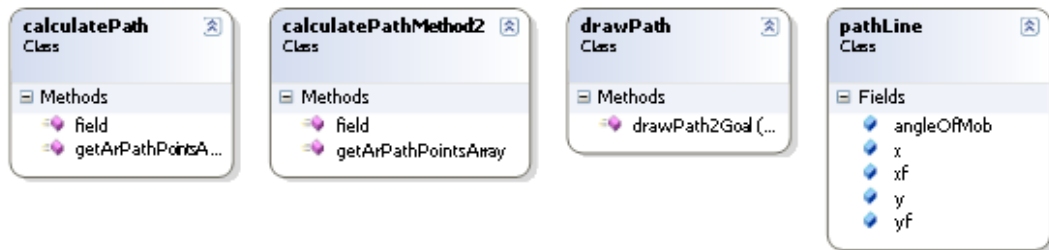
Şekil 3.36 TriangleFunctions, RectangleFunctions, EllipseFunctions ve CircleFunctions sınıfları

LineFunctions, *IsInTriangle*, *ObstaclesMovements* ve *DrawGridsClass* sınıflarından bahsedelim. Bu sınıflar Şekil 3.37’de gösterilmiştir. *LineFunctions* sınıfı çizgi şeklindeki engel çizimiyle ilgili sınıftır. *IsInTriangle* sınıfı verilen bir noktanın üçgenin içinde olup olmadığını belirlemektedir. *ObstaclesMovements* sınıfı engellerin hareketini sağlamaktadır. *DrawGridsClass* sınıfı grid noktalarının çizimini sağlamaktadır.



Şekil 3.37 LineFunctions, IsInTriangle, ObstaclesMovements ve DrawGridsClass sınıfları

CalculatePath, *CalculatePathMethod2*, *DrawPath* ve *PathLine* sınıflarından bahsedelim. Bu sınıflar Şekil 3.38’de gösterilmiştir. *CalculatePath* ve *CalculatePathMethod2* sınıfları yola ait noktaların hesaplanmasını sağlamaktadırlar. *DrawPath* sınıfı yolun çizdirilmesini sağlamaktadır. *PathLine* sınıfı yola ait noktaların hesaplanmasında kullanılan değişkenleri barındırmaktadır.

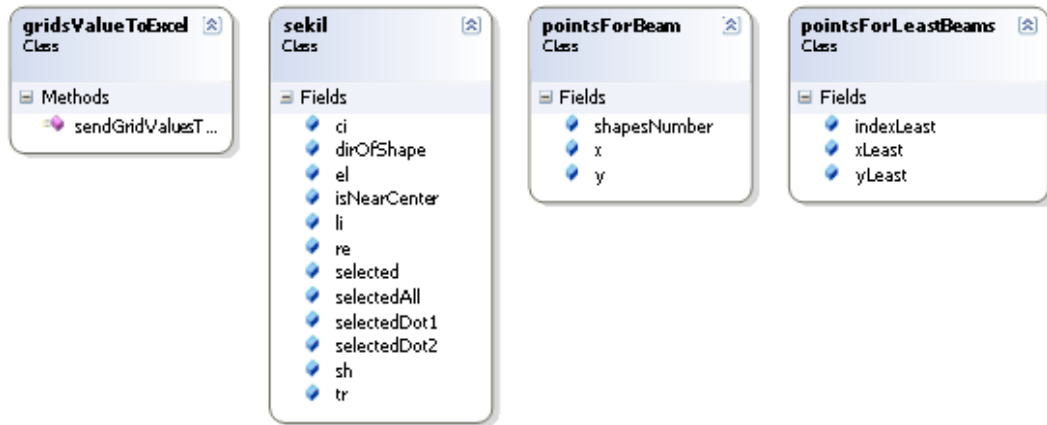


Şekil 3.38 CalculatePath, CalculatePathMethod2, DrawPath ve PathLine sınıfları

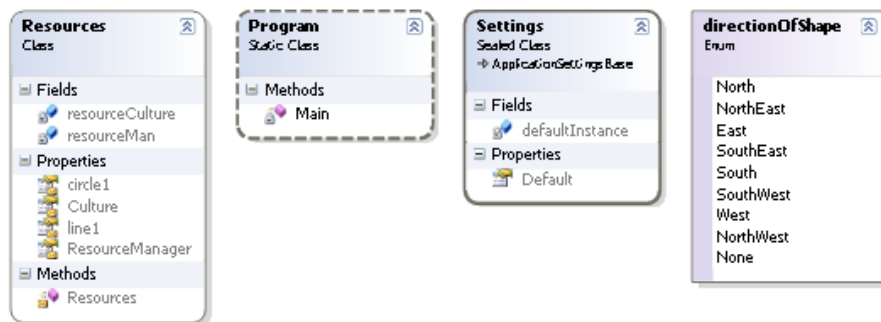
GridsValueToExcel, *Sekil*, *PointsForBeam* ve *PointsForLeastBeams* sınıflarından bahsedelim. Bu sınıflar Şekil 3.39’da gösterilmiştir. *GridsValueToExcel* sınıfı grid değerlerini Excel programına göndermeyi sağlamaktadır. *Sekil* sınıfı ile hangi tip şekil kullanılacak ise o tipe ait elemanların oluşturulması belirlenmektedir. *PointsForBeam* sınıfı ile ışın analizi yöntemindeki ışınlar ait değişkenler barındırılmaktadır. *PointsForLeastBeams* sınıfı ile engellere olan en kısa ışınların değişkenleri saklanmaktadır.

Resources, *Program*, *Settings* ve *DirectionOfShape* sınıflarından bahsedelim. Bu sınıflar Şekil 3.40’da gösterilmiştir. *Resources*, *Program* ve *Settings* sınıfları C#’ta proje oluşturulduğunda gelen dosyalardır. *DirectionOfShape* sınıfı hareketli engellerin hangi yönde hareket edeceklerinin belirlenmesinde kullanılmaktadır. *DirectionOfShape* sınıfı *Sekil* sınıfının içerisinde bulunmaktadır.

CalculateField, *Regis*, *Shape2* ve *Form1* sınıflarından bahsedelim. Bu sınıflar Şekil 3.41’de gösterilmiştir. *CalculateField* sınıfı alanın hesaplanmasını sağlamaktadır. *Regis* sınıfı ile kayıt defteri ile ilgili işlemler yapılmaktadır. *Shape2* sınıfı ile engel tipi seçilmektedir. *Shape2* sınıfı *Sekil* sınıfının içerisinde bulunmaktadır. *Form1* sınıfı ana sınıftır. *Form1* sınıfının içerisinde çeşitli işlemler yapılmaktadır. *Form1*’e ait değişkenler ve metodlar çok olduğundan dolayı Şekil 3.42’de tamamı gösterilememiştir.

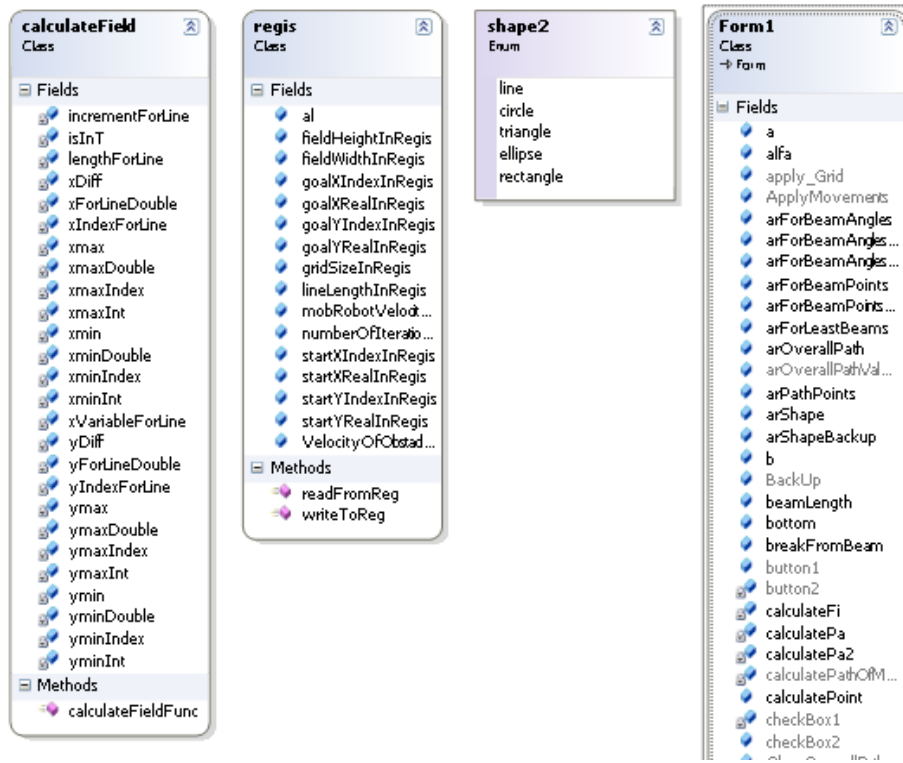


Şekil 3.39 GridsValueToExcel, Sekil, PointsForBeam ve PointsForLeastBeams sınıfları

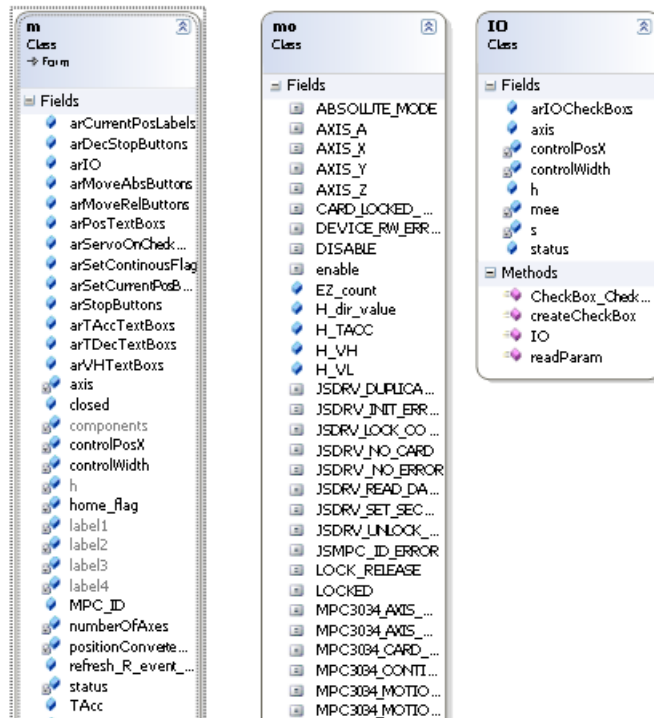


Şekil 3.40 Resources, Program, Settings ve DirectionOfShape sınıfları

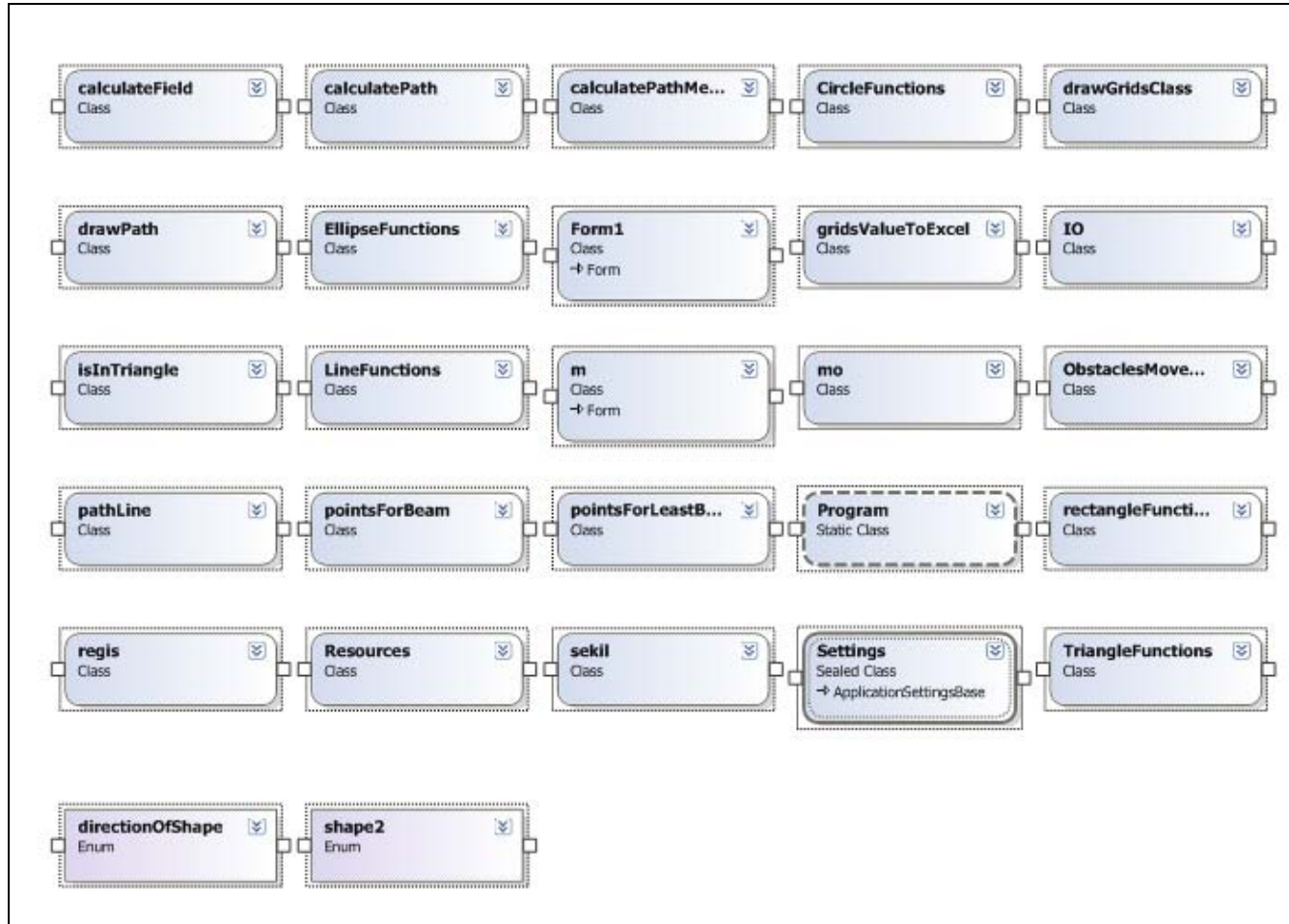
M , Mo ve IO sınıflarından bahsedelim. Bu sınıflar Şekil 3.42’de gösterilmiştir. Bu sınıflar motorların kontrolünü sağlamaktadırlar.



Şekil 3.41 CalculateField, Regis, Shape2 ve Form1 sınıfları



Şekil 3.42 M, Mo ve IO sınıfları

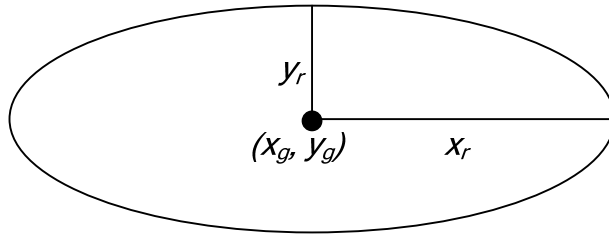


Şekil 3.43 Program2'ye ait sınıflar

Çeşitli engel şekillerinin olduğundan bahsetmiştik. Bu engellerin programda engel olarak tanınabilmeleri için matematiksel olarak içinin tanımlanması gerekmektedir. Dairede denklem 3.28 kullanılmaktadır. Burada (x,y) dairenin içinde olup olmadığına bakılan noktadır. (x_g,y_g) dairenin merkezinin koordinatlarıdır ve r de dairenin yarıçapıdır. Eğer denklem 3.28 sağlanıyorsa (x,y) noktası belirtilen dairenin içerisindeydir.

$$(x - x_g)^2 + (y - y_g)^2 \leq r^2 \quad (3.28)$$

Şekil 3.44'teki gibi elipsin parametrelerini gösterelim. Elipsin içinin tanımlanabilmesi için bu parametreler ile denklem 3.29 kullanılmaktadır. Denklem 3.29'daki (x,y) elipsin içinde olup olmadığına bakılan noktadır. Denklem 3.29 sağlanıyorsa (x,y) noktası elipsin içindedir.



Şekil 3.44 Elipsin parametreleri

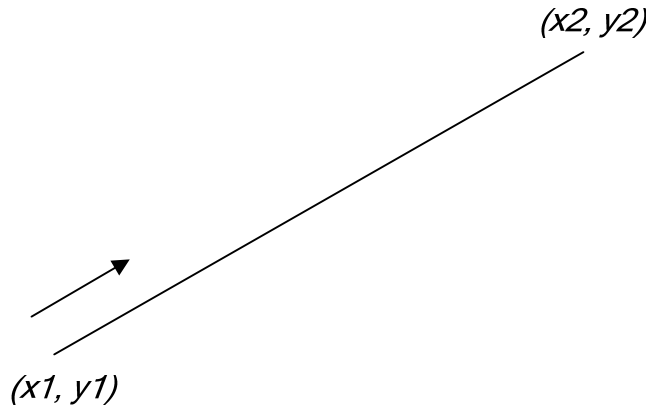
$$\frac{(x - x_g)^2}{(x_r)^2} + \frac{(y - y_g)^2}{(y_r)^2} \leq 1 \quad (3.29)$$

Çizgi şeklindeki engellerde ise doğrunun denkleminde faydalanılmıştır. Bir şeklin engel olabilmesi için denk düştüğü gridleri engele çevirmesi gerekmektedir. Çizgi şeklindeki engelde bir uçtan başlanıp diğer uca doğru gidilirken çizgi üzerindeki noktalara yakın gridler engele dönüştürülür (Şekil 3.45). Çizgi üzerinde ilerleme adımı grid boyuna ve çizginin uzunluğuna bağımlı seçilmiştir. Doğrunun denklemi olarak bilinen denklemlerden birisi 3.30'dur. Bu denklemde y 'yi yalnız bırakırsak denklem 3.31'i elde etmiş oluruz. Denklem 3.31'deki x değeri olarak iki uç noktadan bir tanesinin x koordinatı seçilir ve diğer uç nokta yönünde gidilerek (x,y) noktaları elde edilir. Bu (x,y) noktalarına en yakın gridler engel olarak değerlendirilir.

$$\frac{y - y_2}{y_2 - y_1} = \frac{x - x_2}{x_2 - x_1} \quad (3.30)$$

$$y = (y_2 - y_1) * \left(\frac{x - x_2}{x_2 - x_1} \right) + y_2 \quad (3.31)$$

Dikdörtgenin içinin belirlenmesi diğer şekillere nazaran daha basittir. Dikdörtgen şeklindeki engelimizin 4 uç noktasının koordinatları elimizde bulunmaktadır. Bu koordinatlardaki en küçük x , en büyük x , en küçük y ve en büyük y değerleri bulunmaktadır. Bu değerler elde edildikten sonra bir noktanın dikdörtgenin içinde olup olmadığı bu değerlerin arasında olup olmadığı ile belirlenmektedir.



Şekil 3.45 Çizgi şeklindeki engel

Üçgenin içinin belirlenmesi yukarıda anlatılan şekillere göre biraz daha zordur ve az bilinmektedir. Bir noktanın üçgenin içinde olup olmadığını belirlebilmesi araştırılırken birden fazla yöntem bulunmuştur. Bunlardan bir tanesinde noktadan ışın atılmaktadır. Işın üçgenin sınırlarını bir kez kesiyorsa nokta üçgenin içindedir eğer hiç kesmiyorsa veya iki kez kesiyorsa nokta ışının dışındadır. Bir diğer yöntemde ise nokta üçgenin kenarlarının sağında mı yoksa solunda mı diye bakılarak belirlenmektedir. Örneğin ABC üçgeninde A ucundan B ucuna doğru bakarken noktanın solda mı yoksa sağda mı olduğuna dikkat edilir. Benzer şekilde B ucundan C ucuna doğru bakarken ve C ucundan A ucuna doğru bakarken noktanın sol tarafta mı yoksa sağ tarafta mı olduğuna dikkat edilir. Eğer her üç durumda da nokta aynı tarafta kalıyorsa nokta üçgenin içindedir diyebiliriz. Diğer durumda nokta üçgenin içinde olmamaktadır. Son

anlatılacak yöntemde ise ilk önce üçgenin bir ucu seçilmektedir. ABC üçgeninde A köşesinin seçildiğini düşünelim. AB yönünde belli bir miktar hareketten sonra BC yönünde belli bir miktar hareket ile üçgenin içindeki tüm noktalara erişilebilmektedir. Bu mantıktan hareketle AB ve BC kenarları normalize edilmektedir. Örneğin AB kenarı 5 birim ise ve AB yönünde 2 birim gidilmişse 0,4 değeri almaktadır. AB ve BC yönündeki değerlerin toplamı 1'den küçük olmalıdır. Ayrıca gidiş yönünün doğru olması için AB ve BC yönündeki değerlerin sıfırdan büyük olması gerekmektedir. Eğer üçgenin sınır değerleri de alınmak isteniyorsa büyük eşittir, küçük eşittir şeklinde ifadelerin değiştirilmesi gerekmektedir.

Engellerin içlerinin tanımlanmasından bahsedildi. Engellerin yerinin değiştirilebilmesi veya boyutlarının değiştirilebilmesi için sınırlarından seçilmesi gerekmektedir. Bu seçilme engellerin sınır çizgilerinin tam üzerine tıklama yerine, sınır çizgilerine belli bir tolerans yakınlığıyla yapılmıştır. Kullanıcının yaptığı tıklamanın engelin sınır çizgilerine yakın olup olmadığını anlamak için sınır üzerinde nokta nokta tarama yapılmıştır. Örneğin elipste denklem 3.32 ve 3.33 kullanılmaktadır. Buradaki (x_g, y_g) elipsin merkezidir, x_r x' 'teki yarıçapı ve y_r y' 'deki yarıçapıdır. Kosinüs ve sinüsün içindeki t değeri 0 ile 2π değiştirildiğinde elipsin sınırındaki (x_m, y_m) değerleri elde edilir.

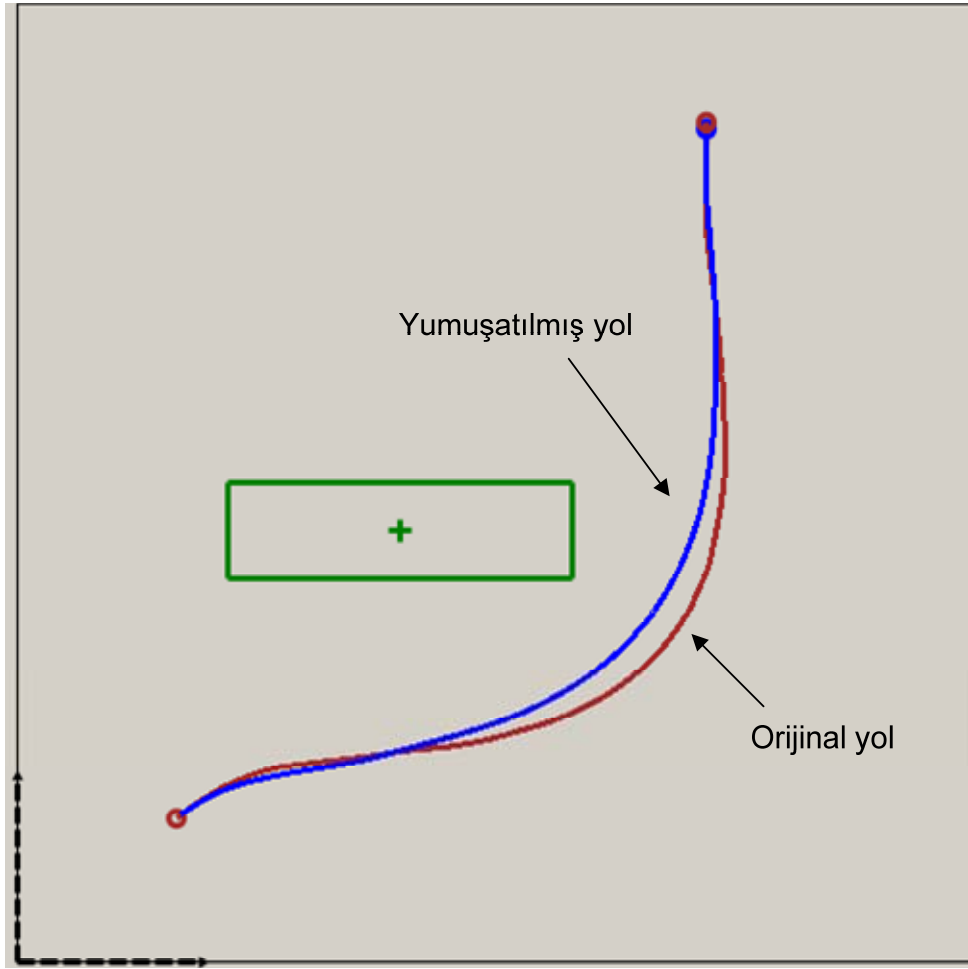
$$x_m = x_g + x_r * \cos(t) \quad (3.32)$$

$$y_m = y_g + y_r * \sin(t) \quad (3.33)$$

Benzer şekilde eğer denklem 3.32 ve 3.33'te x_r ve y_r yerine dairenin yarıçapını yazarsak dairenin sınırı üzerinde hareket etmiş oluruz. Çizgi şeklindeki engelimizde doğrunun denkleminde yararlanarak engel üzerindeki noktalara bakılmaktadır. Üçgen şeklindeki engelde de yine doğrunun denkleminde yararlanılmaktadır. Bu sefer üçgene ait üç doğruya da bakılmaktadır. Dikdörtgen şeklindeki engelde ise nokta nokta bakılmamıştır. En küçük x , y ve en büyük x , y değerleri çıkarıldıktan sonra $(eX > xmin - a$ ve $eX < xmax + a$ ve $eY > ymin - a$ ve $eY < ymax + a)$ şeklinde bir test ile dikdörtgenin bir kenarına yakın tıklanıp tıklanmadığı anlaşılmaktadır. Bu test sadece dikdörtgenin bir kenarına aittir. Diğer kenarlar için de buna benzer bir koşul

yazılmalıdır. Burada (eX , eY) fare'nin (x , y) koordinatıdır ve a değişkeni de tolerans için kullanılmıştır.

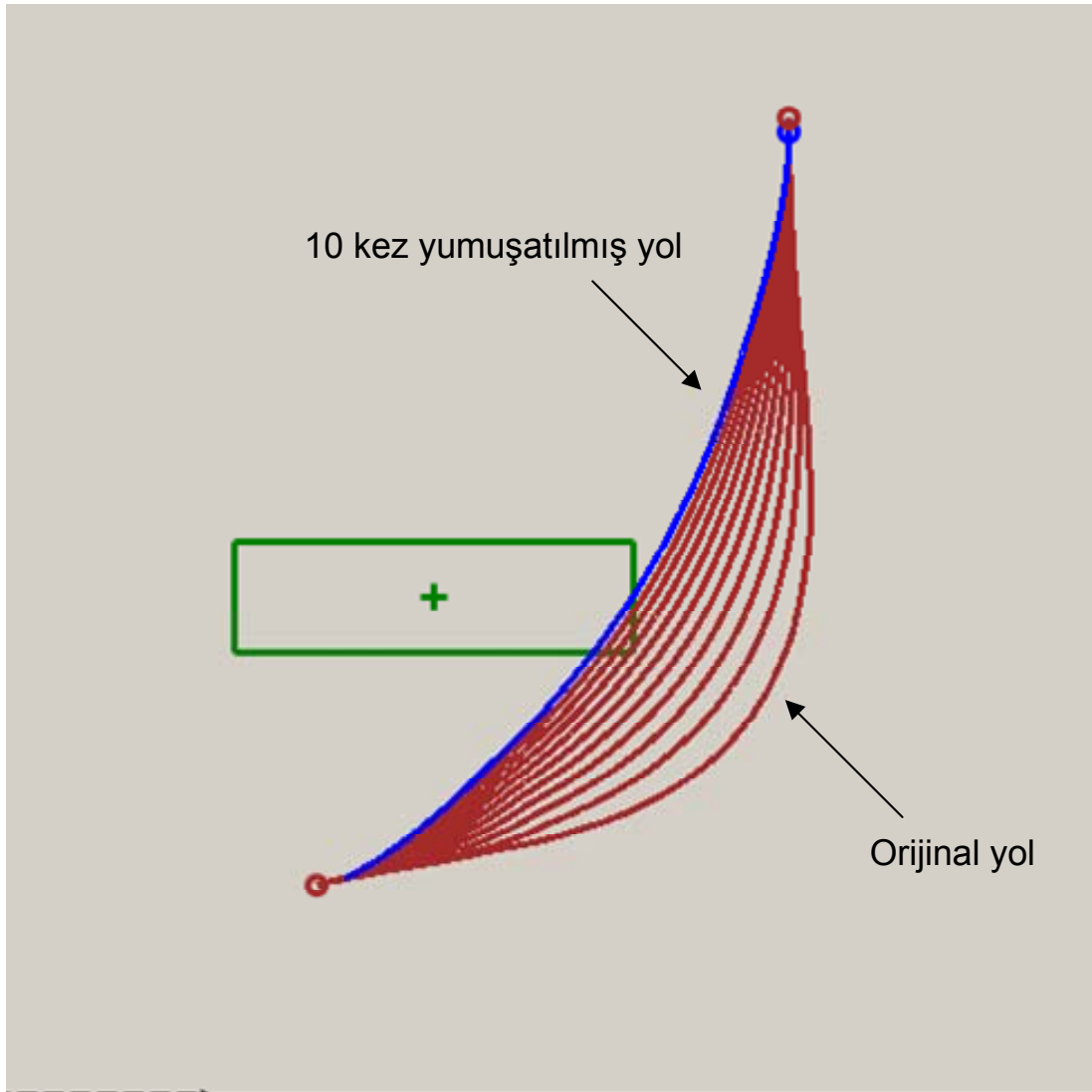
Mobil robotun izlediği yola yumuşatma uygulanmıştır. Yolu yumuşatmada Hareketli Ortalama Filtresi (Moving Average Filter) olarak bilinen bir yöntem kullanılmıştır (WEB_7 2002). Bu yöntemde belirlenen bir nokta sayısı kadar, bir noktanın arkasındaki ve önündeki noktalara ait değerlerin ortalaması alınıp, noktanın yeni değeri belirlenmektedir. Örneğin iki taraftan da ortalama girecek nokta sayısı 2 olduğu ve 20. noktanın yeni değerinin bulunması durumunu inceleyelim. Burada 18. noktadan 22. noktaya kadar olan noktaların ortalaması alınıp 20. noktanın yeni değeri belirlenmektedir. Bir taraftan ortalama girecek nokta sayısı hesaplanacak noktanın sırasından büyük ise hesaplanacak noktaya kadarki eleman kadar noktanın ilerisinden eleman hesaba katılmaktadır. Örneğin belirlenen nokta sayısı 10 olsun ve hesaplanan noktanın sırası 4 olsun. Bu durumda 4. sıradaki noktanın gerisinde sadece 3 nokta olduğundan dolayı ortalama bir tarafından 10 nokta alınamamaktadır. Bu noktanın gerisinden 3 nokta ve ilerisinden 3 nokta alınarak ortalama hesaplanmaktadır. Yumuşatma uygulanmış bir örnek Şekil 3.46'da gösterilmiştir. Şekil 3.46'dan da görüldüğü gibi yumuşatılmış yol orijinal yola göre daha düzgündür.



Şekil 3.46 Yumuşatılmış yol ve orijinal yol

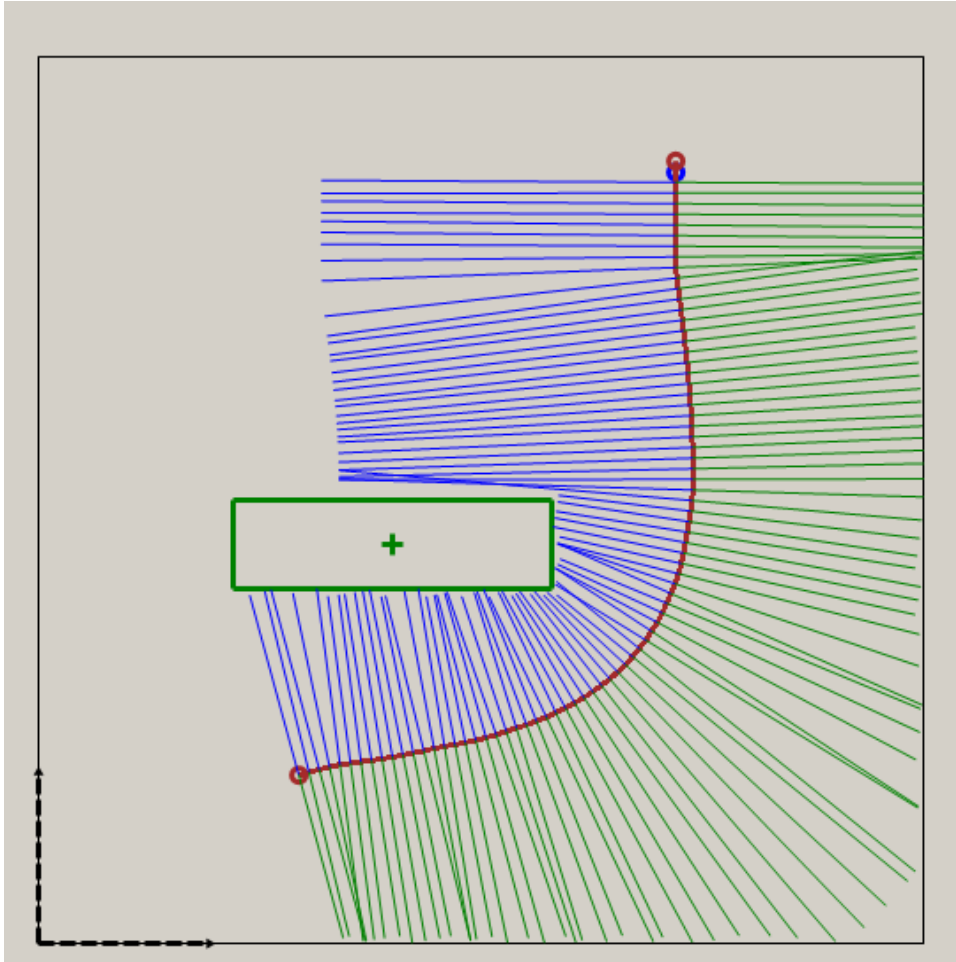
Şekil 3.46'daki durumda 36 noktalı yumuşatma yapılmıştır yani noktanın bir tarafından 18 nokta alınarak yumuşatma yapılmıştır. Yumuşatmayı eğer fazla yaparsak yol engelin üzerine gelebilmektedir (Şekil 3.47). Bu da istenmeyen bir durumdur.

Program2'de Çonkur vd'ye (2005) ait ışın analizi yöntemi (beam analysis algorithm) de kullanılmıştır. Işın analizi yönteminde yola ait noktalardan yolun o noktadaki açısına dik ışınlar atılmaktadır. Işının boyu istenildiği gibi ayarlanabilmektedir. Bu yollanan ışınlar engelle karşılaştığında veya alan çerçevesi ile karşılaştığında daha ileriye gidememektedirler.



Şekil 3.47 10 kez yumuşatılmış yol ve orijinal yol

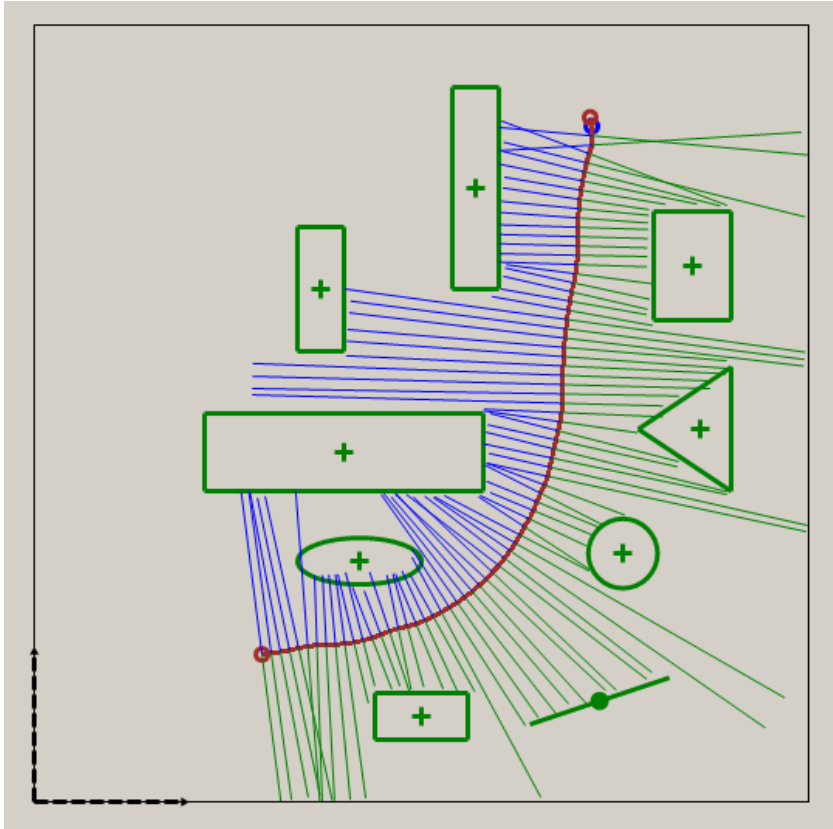
Şekil 3.48’de ışın analizi yönteminin uygulanmış hali gözükmemektedir. Bu yöntem uygulandıktan sonra engele çarpan ışınlardan en kısa olanı buldurulmuştur. En kısa ışınları hesaplamadan önce tüm engel çeşitlerine bu yöntemin uygulanabildiğini belirtmek gerekir (Şekil 3.49). Işın analizi yönteminde ışının atılacağı yön bulunduğundan sonra ışın yönünde belirlenen bir aralıkla ilerleme yapılır. İlerlenerek bulunan her noktaya en yakın gride bakılır. Eğer bu grid engelin içindeyse ışının ilerlemesi durdurulur. Şekilden bağımsız bir işlem olduğundan bu yöntem tüm engel şekillerine kolayca uygulanabilmiştir.



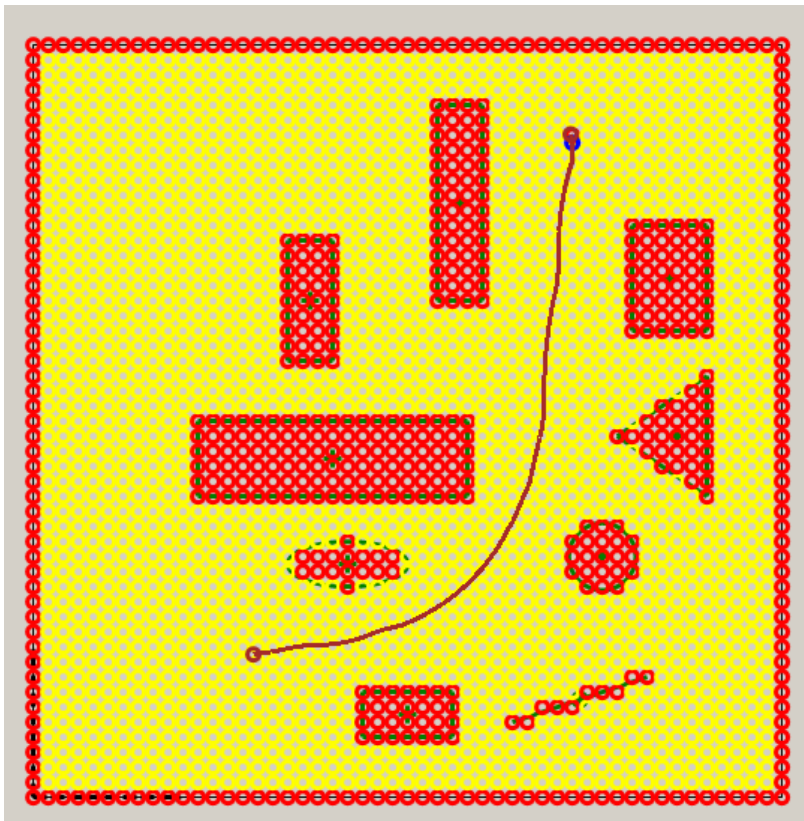
Şekil 3.48 Işın analizi yöntemi

Işın üzerinde ilerleme adımı kısa tutulursa hesaplama süresi uzayacaktır. Eğer ilerleme adımı uzun tutulursa bu sefer de engelin içinde durabilir veya çizgi şeklinde engel ise engeli atlayıp ışın devam edebilir. Bu yüzden grid aralığı ile uygun bir ilerleme adımı seçilmelidir.

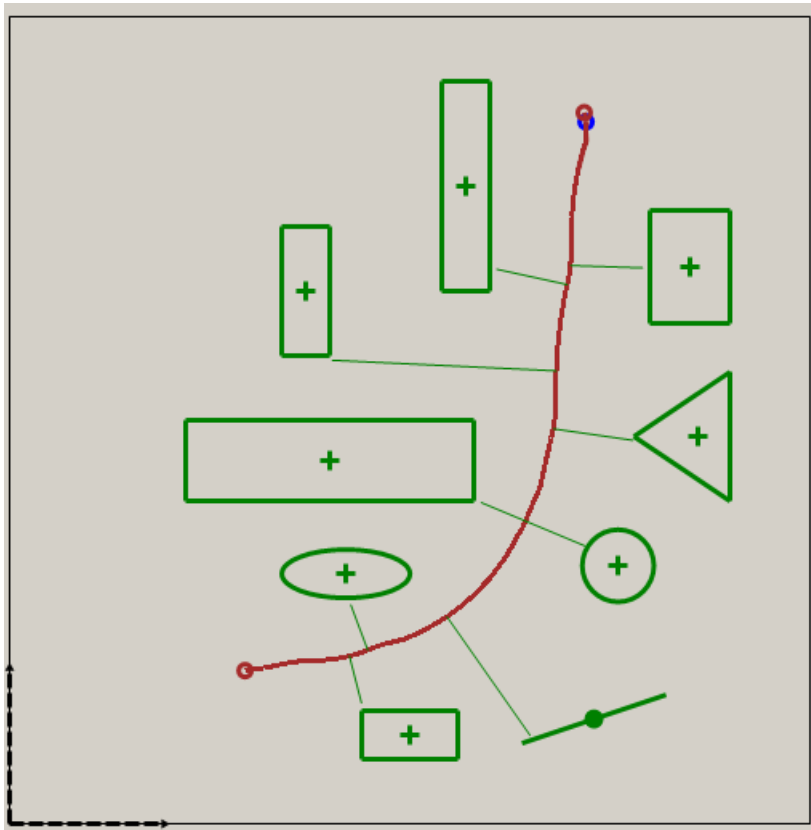
Şekil 3.50’de Şekil 3.49’daki durumda hangi gridlerin engel gibi davrandığı gözükmemektedir. Şekil 3.49’a dikkat edilirse elipste birden fazla ışının üçgende ise bir adet ışının engelin içine girdiği görülmektedir. Bunun sebebi aslında Şekil 3.50’den anlaşılabilir. Engel gibi davranan gridlere bakarsak tam olarak elips şeklini verememektedirler. Grid aralığı küçültülerek daha düzgün sonuçlar elde edilebilmektedir. Böylece gridlerin kapladığı alan daha çok çizilen engele benzeyecektir. Bu durumda da Şekil 3.49’dakinin aksine ışınların engelin içine girmesi önlenilebilecektir.



Şekil 3.49 Çeşitli şekillerdeki engellerin olduğu ortamda ışın analizi yöntemi



Şekil 3.50 Gridlerin durumu

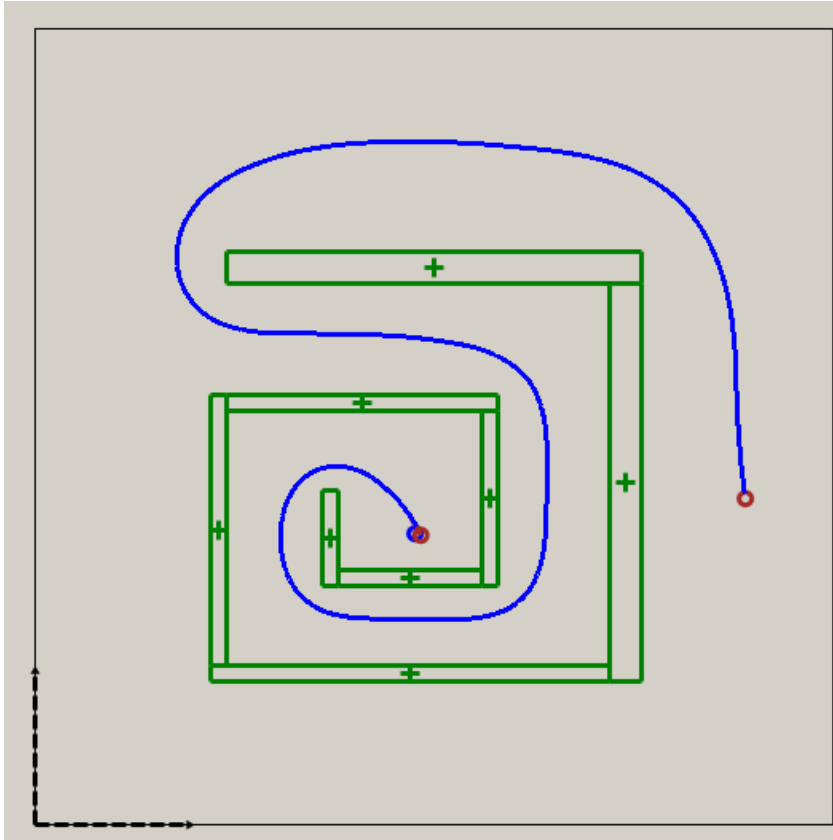


Şekil 3.51 En kısa ışınların gösterilmesi

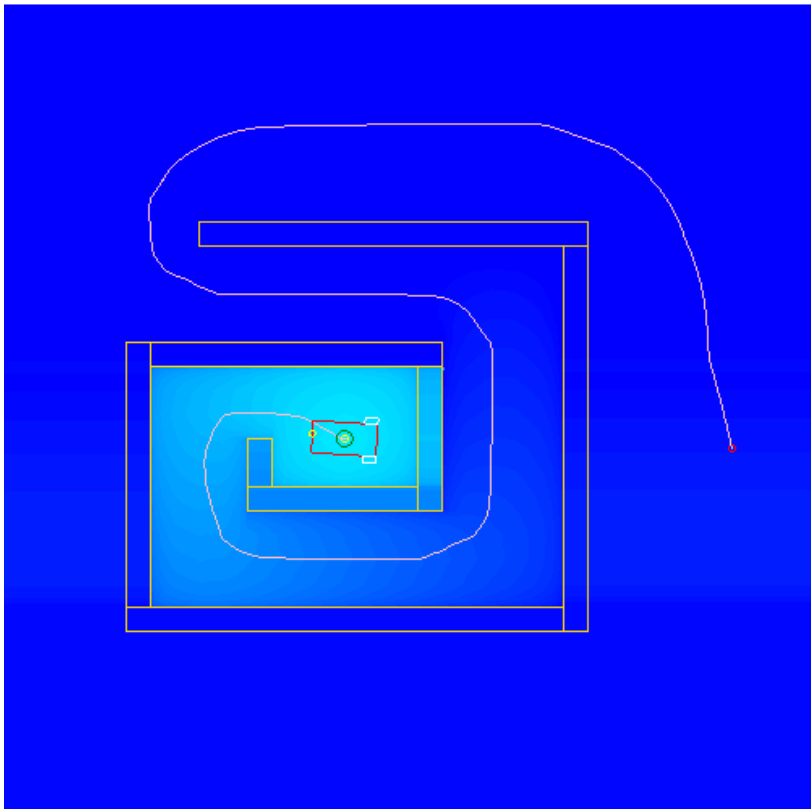
Şekil 3.51'deki gibi en kısa ışınların buldurulup gösterilebilmesi için her engel için yolun tüm noktalarından çıkan ışınlar bir kez kontrol edilmektedir. Kontrol edilen engele çarpan ışınlar bu sırada tespit edilmektedir ve bu ışınlar arasındaki en kısa olanı çizdirilmektedir.

Yazılan programlar üzerinde denemeler yapılırken mobil robotun hareketsiz engellerin olduğu durumlarda eğer hedef ile mobil robot arası geçilebilir ise her zaman uygun bir yol bulunduğu tespit edilmiştir.

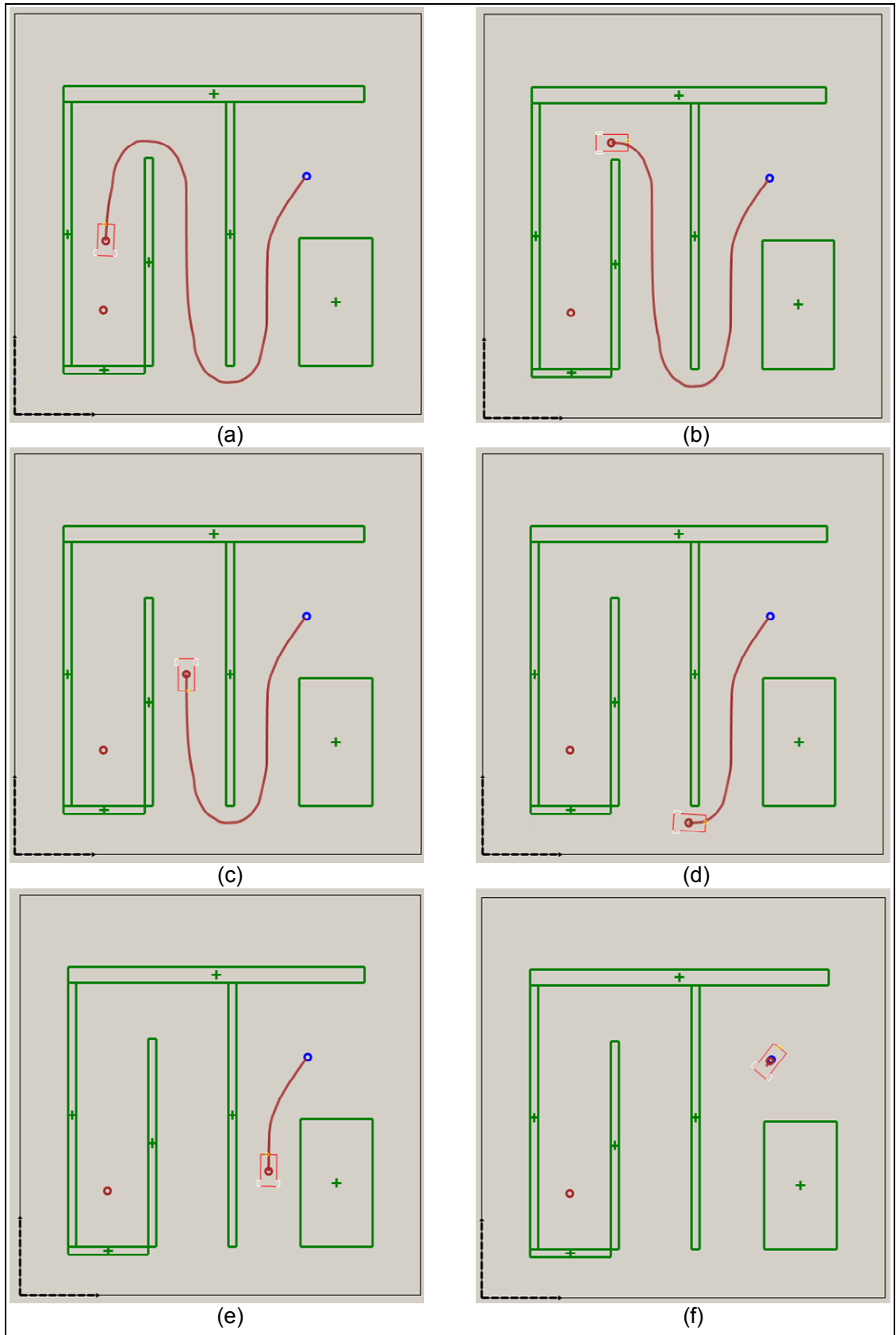
Yola ait noktalar arası uzaklık arttırılırsa yoldaki pürüzlülük daha fazla gözükmektedir. Böyle durumlarda az noktalı da olsa bir yumuşatma uygulanması iyi olmaktadır. Şekil 3.52'de hedefin labirentin içine konulduğu durumdaki yol gösterilmiştir. Şekil 3.52'deki yol 10 noktalı yumuşatma uygulanmış bir yoldur.



Şekil 3.52 Hedefin labirentin içinde olduğu durum(Program2)



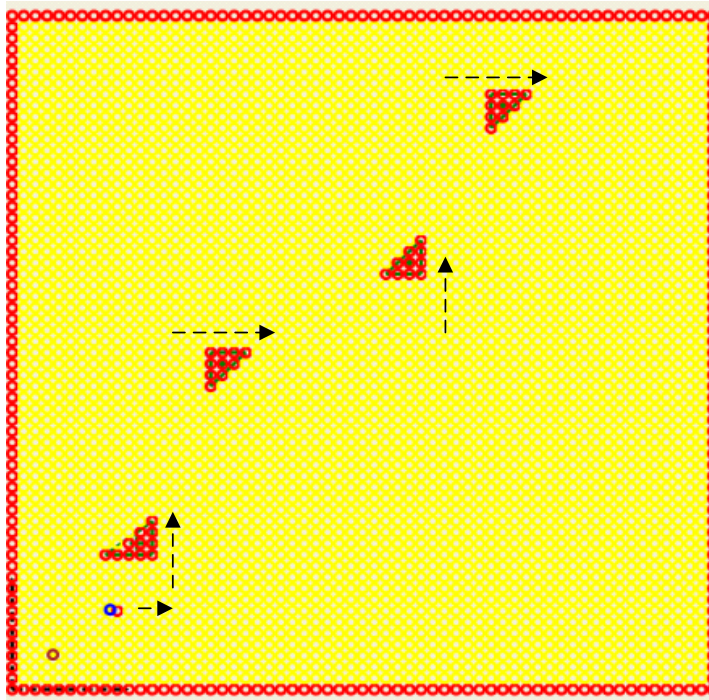
Şekil 3.53 Hedefin labirentin içinde olduğu durum(Program1)



Şekil 3.54 Hareketsiz engellerin olduğu bir durumda Program2'den anlık görüntüler

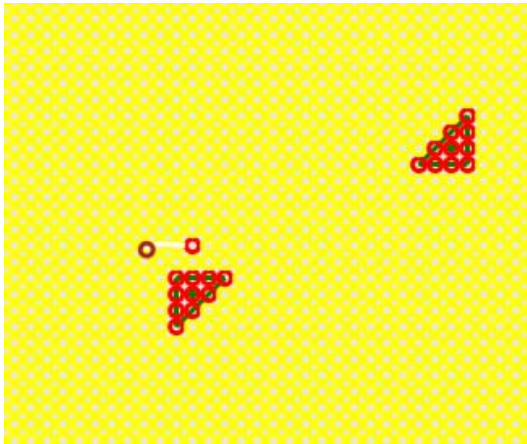
Şekil 3.53'te Şekil 3.52'dekine benzer bir durumda Program1'in görüntüsü verilmiştir. Şekil 3.54'te hareketsiz engellerin olduğu bir ortamda mobil robotun hedefine giderken anlık görüntüler gösterilmiştir.

Şekil 3.55 ve Şekil 3.56 hareketli bir hedefin olduğu bir programdan görüntülerdir. Burada hedef Willms ve Yang'ın (2008) yaptığı gibi engeller arasında zikzak hareketi yapmaktadır. Burada hedefin hızı 10 pixel/sn ve mobil robotun hızı 12 pixel/sn seçilmiştir. Ayrıca mobil robotun başlangıç konumu (40,40) ve hedefin başlangıç konumu (80,70) olarak seçilmiştir. Şekil 3.55'te hedefin izleyeceği yol oklarla gösterilmiştir.

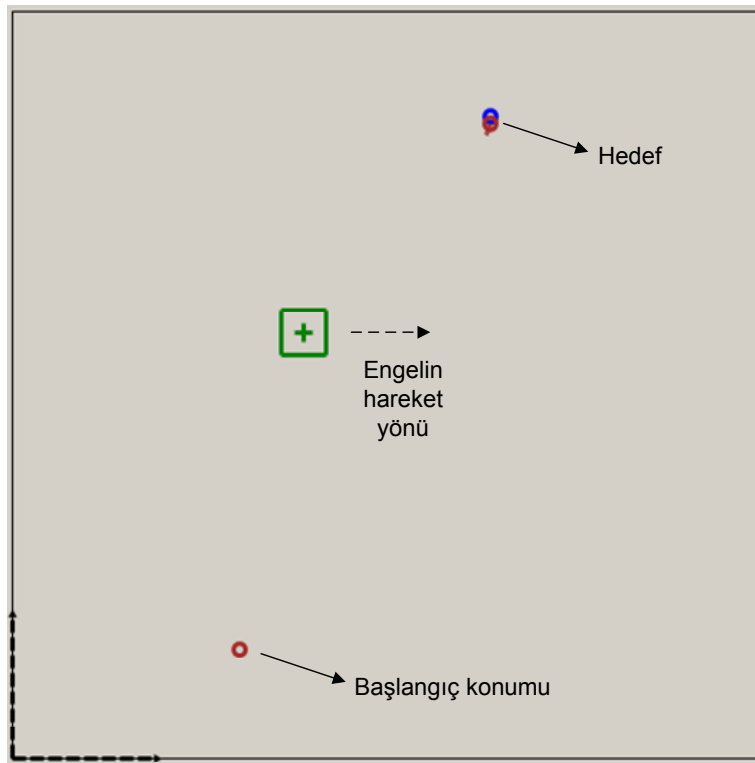


Şekil 3.55 Hareketli hedefin olduğu bir durum (başlangıç konumu)

Şekil 3.55'te verilen durumda hedef 3. engeli dönmeden mobil robot hedefi yakalamaktadır. Willms ve Yang'ın (2008) yaptığı gibi bir hedef hareketinin seçilmesinin sebebi bir karşılaştırma yapabilmektir. Karşılaştırma sonucunda Willms ve Yang'ın (2008) yaptığı örneğe nazaran yola yumuşatma yapılmadığı halde bizim programımızda daha yumuşak bir yol elde edilmektedir. Bunun sebebi Willms ve Yang (2008) yola ait noktaları hep gridler üzerinden seçmiştir. Bizim programımızda ise yola ait noktaların gridler üzerinde olma zorunluluğu yoktur.



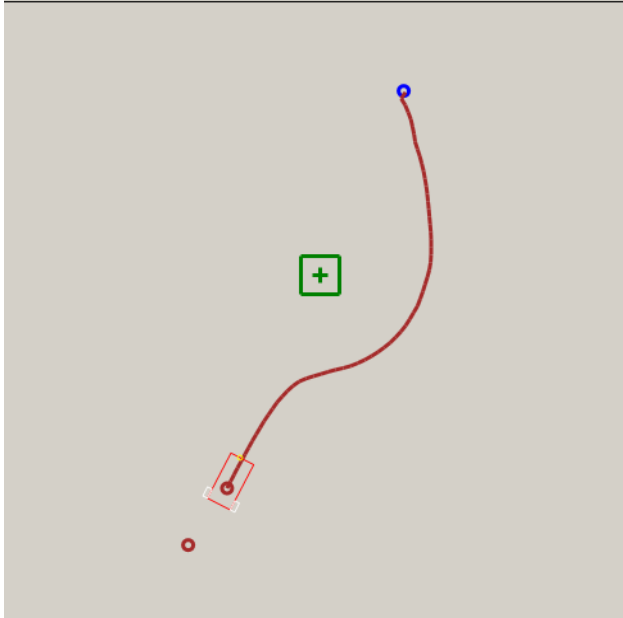
Şekil 3.56 2. engel dönülürken programdan bir görüntü



Şekil 3.57 Hareketli engelin olduğu durum (başlangıç konumu)

Şekil 3.57’de hareketli bir engelin olduğu durum sırasında başlangıç anı gösterilmektedir. Engelin hızı 30 pixel/sn ve mobil robotun hızı 60 pixel/sn olarak seçilmiştir. Şekil 3.58’den de görüldüğü gibi ilk önce mobil robot engelin sağından geçmeyi planlamaktadır. Daha sonra engelin hareketinden dolayı Şekil 3.59’da görüldüğü gibi engelin solundan geçmektedir. Potansiyel alan hareketli engellerde çoğunlukla hedefe ulaşmayı sağlamaktadır ama hareketli engel sayısı artınca veya

hareketli engellerin boyutları büyük olunca mobil robotun izlediği yol bazen optimal olmamaktadır. Bu yüzden potansiyel alan metodunun hareketli engeller olduğu durumlarda iyileştirilmeye ihtiyacı olduğu anlaşılmıştır.



Şekil 3.58 Hareketli bir engelin olduğu durumda mobil robotun hareketi ($t = 1$)



Şekil 3.59 Hareketli bir engelin olduğu durumda mobil robotun hareketi ($t = 3$)

3.4. Hareket Kontrol Kartı'nın(Motion Control Card) Yazılım ile Kontrolü

JS Automation Corporation firmasının sağladığı 4 eksenli hareket kontrol kartı (MPC3034) satın alınmıştır. Hareket kontrol kartı bilgisayar ile motorlar ve sürücülerin kontrolünü sağlamaktadır. Burada 4 eksenliden belirtmek istenen 4 adet motoru aynı anda kontrol etmektir. MPC3034 Hareket kontrol kartı servo veya adım motorları kontrol edebilmektedir. Satıcı firmanın hareket kontrol kartı ile birlikte yolladığı hazır yazılım ile hareket kontrol kartına erişilebildiği gibi isteğe göre program yazma şekliyle de hareket kontrol kartına erişilebilmektedir. Borland C/C++, Microsoft Visual C++, Microsoft Visual Basic, Microsoft Visual C# veya Windows DLL'lerini (Dynamic Link Library) çağırabilen Windows tabanlı bir program ile hareket kontrol kartına erişim sağlanabilmektedir. Belirtilen programlama dillerinden herhangi biri ile yazılmış bir programa drv3034.dll ve MPC3034.dll dosyalarının eklenmesi gerekmektedir. Ayrıca hareket kontrol kartı ile ilgili API'leri (Application Programming Interface) barındıran bir sınıfı da yazılan programa eklemek gerekmektedir. API'ler sayesinde hareket kontrol kartına erişilebilmektedir. API'lere erişim için de hareket kontrol kartını satan firma (JS Automation Corporation) tarafından fonksiyonlar yazılmıştır. 125 adet olan bu fonksiyonların listesi Ek-3'te verilmiştir (WEB_6 2008). Bu fonksiyonlardan bir kısmı aşağıda açıklanmıştır.

- MPC3034_initial()

MPC3034 kaynaklarının yüklenmesini sağlamaktadır. Program çalıştırıldığında bu fonksiyonun da çağırılması gerekmektedir.

Formatı: u32 status = MPC3034_initial(void);

- MPC3034_close()

Program kapatılırken bu fonksiyon çağırılarak MPC3034 kaynaklarının serbest bırakılması sağlanmaktadır.

Formatı: u32 status = MPC3034_close(void);

- MPC3034_set_pulse_outmode()

Darbe çıkış modunun ayarlanmasını sağlamaktadır.

Formatı: u32 status = MPC3034_set_pulse_outmode(u8 CardID, u8 Axis, u8 pulse_outmode);

- MPC3034_write_output_point()

Belirtilen çıkış uçlarını aktif veya inaktif yapmaya yaramaktadır.

Formatı: u32 status = MPC3034_write_output_point(u8 CardID, u8 Axis, u8 point_factor, u8 on_off);

Burada point_factor yazan kısım için 0: ERC, 1: SVON, 2: FIN, 3: CMP, 4: IO_0, 5: IO_1, 6: IO_2, 7: IO_3, 8: IO_4, 9: IO_5, 10: IO_6, 11: IO_7 açıklaması vardır. On_off yazan kısım için ise 0: inaktif, 1: aktif şeklinde kullanılabileceği belirtilmektedir.

- MPC3034_read_current_position()

Belirtilen eksenin o andaki pozisyonunun koordinatını okumak için kullanılmaktadır.

Formatı: u32 status = MPC3034_read_current_position(u8 CardID, u8 Axis, i32 *current_posi);

Buradaki current_posi'nin başındaki yıldız bu kısmın çıkış olarak kullanıldığını belirtmektedir. Başka bir deyişle current_posi'nin değeri verilmemektedir, alınmaktadır. Current_posi'nin değeri -134,217,728 ile 134,217,727 aralığında değişebilmektedir.

- MPC3034_T_curve_move_LINE2()

2 eksenli linear interpolasyon yapılmasını sağlamaktadır.

Formatı: u32 status = MPC3034_T_curve_move_LINE2(u8 CardID, u8 line2_index, i32 Position1, i32 Position2, u8 posi_mode, i32 VL, i32 VH, f64 Tacc, f64 Tdec);

Buradaki line2_index'in açıklaması şu şekildedir; 0: X-Y, 1: X-Z, 2: X-A, 3: Y-Z, 4: Y-A, 5: Z-A. Position1 ilk eksen için hedef pozisyonudur ve Position2 ikinci eksen için hedef pozisyonudur. Position1 ve Position2 -134,217,728 ile 134,217,727 aralığında değişebilmektedir. Posi_mode belirtilen pozisyonun nispi mi yoksa mutlak mı uygulanacağını belirler. 0 değeri verilirse nispi, 1 değeri verilirse mutlak görevi yapmaktadır. Nispi de belirtilen pozisyon kadar o eksen ilerlemektedir, mutlak görevinde ise belirtilen pozisyona o eksen gitmektedir. VL başlangıç ve bitiş hızını, VH normal ilerleme hızını belirtmektedir. Tacc hızlanma süresini ve Tdec yavaşlama süresini belirtmektedir. VL ve VH pps(pulse per second) cinsinden, Tacc ve Tdec ise saniye cinsinden verilmektedir.

- MPC3034_set_continuous_flag()

Sürekli bayrağı aktif veya inaktif durumuna getirmektedir.

Formatı: u32 status = MPC3034_set_continuous_flag(u8 CardID, u8 Axis, u8 conti_flag);

Conti_flag yazan kısma 0 gelirse sürekli mod kapatılır, 1 gelirse sürekli mod açılır.

- MPC3034_check_continuous_buffer()

Tampon(buffer) bayrağının durumunu öğrenmek için kullanılır.

Formatı: u32 status = MPC3034_check_continuous_buffer(u8 CardID, u8 Axis, u8 *buffer_full_flag);

Buradaki buffer_full_flag tamponun durumunu vermektedir. Eğer değeri 0 ise tampon dolu değildir ve hareket kontrol kartı bilgisayardan komut alabilmektedir. Buffer_full_flag'in değeri 1 ise tampon doludur ve hareket kontrol kartı bilgisayardan komut alamamaktadır.

- MPC3034_config_softlimit()

Belirtilen eksenin limit değerini aşması durumunda nasıl duracağı ayarlanmaktadır.

Formatı: u32 status = MPC3034_config_softlimit(u8 CardID, u8 Axis, u8 source_sel, u8 SL_action);

Buradaki source_sel yollanmak istenen pozisyonun mu yoksa okunacak pozisyon değerinin mi kullanılacağını belirler. Source_sel için 0: yollanmak istenen pozisyon, 1: okunan pozisyon olarak açıklanmıştır. SL_action nasıl durulacağını belirlemektedir. SL_action için 1: ani durma, 2: yavaşlayarak durma olarak açıklanmıştır.

- MPC3034_set_softlimit_data()

Eksenler için limit değerlerini ayarlamaya yarar.

Formatı: u32 status = MPC3034_set_softlimit_data(u8 CardID, u8 Axis, i32 P_limit, i32 N_limit);

Buradaki P_limit pozitif yöndeki limittir ve N_limit negatif yöndeki limittir. Her iki limitte -134,217,728 ile 134,217,727 değerleri arasında seçilebilmektedir.

- MPC3034_enable_softlimit()

Limitleri kaldırmayı veya etkinleştirmeyi sağlamaktadır.

Formatı: u32 status = MPC3034_enable_softlimit(u8 CardID, u8 Axis, u8 ON_OFF);

ON_OFF kısmı 0 değeri aldığı anda limitleri kaldırır, 1 değeri aldığı anda limitleri etkinleştirir.

Yukarıdaki fonksiyonların bir kısmında Axis değişkeni görülmektedir. Axis değişkeni o fonksiyonun hangi eksene uygulanacağını belirlemektedir. Axis değişkeninin olduğu yere 0 yazılırsa X eksen, 1 yazılırsa Y eksen, 2 yazılırsa Z eksen ve 3 yazılırsa A eksen kullanılmaktadır. Benzer şekilde çoğu fonksiyonda CardID

değişkeni görülmektedir. CardID değişkeni de hangi hareket kontrol kartına komutların yollandığını belirtmek için kullanılmaktadır. Birden fazla hareket kontrol kartı aynı programdan kontrol edilebilmektedir. Burada bir adet hareket kontrol kartı kullanıldığından CardID olarak 0 kullanılmıştır.

4. SONUÇ VE ÖNERİLER

Bu tezde diferansiyel sürüş sistemine sahip bir mobil robotun tasarımı, imali ve kontrolü yapılmıştır. Mobil robotun tasarımında SolidWorks programından yararlanılmıştır. Mobil robotta 4 adet tekerlek bulunmaktadır. Mobil robotun dönme ve öteleme hareketleri iki tekerlek ile sağlanmaktadır. Diğer iki tekerlek ise avare tekerlektir ve mobil robotun daha dengeli olması için kullanılmışlardır. Hareketi sağlayan tekerlekler ise scooter tekerleği seçilmiştir. Bu tekerleklere ayrı ayrı tahrik verilmektedir. Avare tekerleklerinin çapları 90mm ve scooter tekerleklerinin çapları 430mm'dir. Mobil robotun tahrik sistemi, motorların uçlarına redüktörlerin bağlanması ve redüktörlerin uçlarına da tekerleklerin bağlanması şeklindedir.

Mobil robotun boyutları (exdxy) 540x600x550(mm) şeklindedir. Mobil robotu 3 katmana ayırabiliriz. Üst katmanda acil durma butonu, bilgisayarın ekranı, klavyesi ve faresi bulunmaktadır. Orta katmanda bilgisayarın anakartı, güç kaynağı ve harddiski, 24 volt güç kaynağı, hareket kontrol kartı (motion control card) ve kablolama kartı (wiring board) bulunmaktadır. Alt katmanda ise kesintisiz güç kaynağı, 2 adet motor, 2 adet redüktör ve 2 adet de servo sürücüsü (driver) bulunmaktadır.

Kullanılan redüktörler 3,5kg'dır, servo sürücüler 1,2kg'dır, servo motorlar 1,6kg ve UPS 33kg'dır. Bu da toplam 45,6kg yapmaktadır. Ayrıca tekerleklerinin, alt, orta ve üst parçalarının, bağlantı parçalarının ve bilgisayarın parçalarının da ağırlıkları göz önünde tutulursa mobil robot toplamda yaklaşık 60kg ağırlığındadır. Motorların torkları 1,27 Nm'dir. Redüktörlerin oranı 1/25 olduğundan dolayı maksimum 63,5 Nm tork elde edilebilmektedir. Bu da mobil robotun hareketi için yeterli bir torktur.

Kullanılan servo motorlar 360°'yi 10000'e bölebilmektedirler yani 0.036°'lik bir hassasiyetleri vardır. Redüktörlerdeki boşluktan ve mobil robotun ağırlığından dolayı servo motorlar tam olarak istenilen yere gitmemektedir. Bu durumdan dolayı servo motor geri beslemesinde servo sürücüyeye istenilen yere gidemediğini bildirmektedir.

Servo sürücü de tekrar belirtilen konuma gitmesini istemektedir. Servo motorda yine gidemeyerek sistem titreşim durumuna girmektedir. Bu durum geri besleme katsayıları değiştirilerek düzeltilebilmektedir. Katsayılar elle ayarlanabileceği gibi sistem çalıştırılarak da otomatik olarak katsayıların ayarlanması istenebilmektedir. Sonuçta bu katsayılar ayarlanarak titreşime giren mobil robot düzeltilmiştir. Mobil robotta güç kaynağı olarak 3000 VA'lık UPS kullanılmıştır.

Mobil robotun yazılımı Visual C# .NET programı ile yazılmıştır. İlk başta 2005 sürümüyle program yazılmaya başlanmıştır. Daha sonra 2008 sürümünün çıkmasıyla bu sürüme geçiş yapılmıştır. Bu sürümle birlikte gelen LINQ da projede kullanılmıştır.

Yazılımdaki engellerin çizimi 2 boyutlu bir çizim programındaki gibi kullanışlı yapılmaya çalışılmıştır. Yazılım sınıflara ayrılarak yazıldığından programın geliştirilebilirliği artmıştır. Mobil robotun yazılımı ile robotun başlangıç ve hedef konumları belli bir ortamda hareketli veya hareketsiz engeller arasından hedefine ulaşabilmesi sağlanmıştır. Mobil robotun yol planlamasında potansiyel alan metodu kullanılmıştır.

Mobil robotun kontrolünde ilk başta *point to point* modu düşünülmüştür. Bu modda kart belirtilen noktalar arasında iki eksenli interpolasyon yapmaktadır. Bu modda mobil robot çalıştırıldığında nokta geçişlerinde motorların çok kısa bir süreliğine durup kalktığı gözlenmiştir. Bu da mobil robotun bu geçişlerde titreşmesine sebep olmaktadır. Bu yüzden *point to point* modu yerine *speed* modu tercih edilmiştir. Bu modda belirli bir süre istenilen hızda hareket yaptırılmaktadır. Hız-zaman grafiğinin altı yer değiştirme bilgisini vermektedir. Bu sayede mobil robotun gideceği yol miktarı hesaplanabilmektedir.

Simülasyon denemelerinde hareketli engellerin olduğu bazı durumlarda mobil robotun optimal olmayan yollar ürettiği görülmüştür. Bazen de engelin yerinin sürekli değişmesinden dolayı mobil robot da sürekli gideceği yolu değiştirmektedir. Bu da mobil robotun hedefine ulaşamamasına sebep olmaktadır. Bu durum potansiyel alan metodunun olumsuz bir yönüdür. Çoğunlukla hareketli engelli ortamda hedefe ulaşabilmesine rağmen yukarıda belirtildiği gibi bazı durumlarda hedefe ulaşamamaktadır.

Daha önceki çalışmaların bir kısmıyla karşılaştırıldığında simülasyon ortamındaki çalışmalarda hedefe daha uygun yollar bulunmuştur. Ayrıca servo motorun tahrik mekanizmasında daha önce kullanılmadığı da görülmüştür. Bu çalışma ile ileride üzerinde çalışılabilecek bir mobil robot imalatı yapılmıştır. İleride yapılacak çalışmalarda mobil robota sensör, kamera vb. ek cihazlar eklenerek mobil robotun kabiliyeti arttırılabilir.

KAYNAKLAR

- Agirrebeitia, J., Avile's, R., Bustos, I. F., and Ajuria, G. (2005) A new APF strategy for path planning in environments with obstacles. *Mechanism and Machine Theory*, 40: 645-658.
- Ateş, A. (2004) Mobil Robot Çalışma Prensipleri ve Mobil Robot Uygulaması, Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 130s.
- Aydın, S. (2003) Mobil Robotlarda Evrimsel Metotlar ile Optimal Hareket Planlama, Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 99s.
- Barlas, F. (2004) Design of a Mars Rover Suspension Mechanism, Master of Science Thesis, The Graduate School of Engineering and Sciences of The Izmir Institute of Technology, Izmir, 93p.
- Baştan, M. (2004) Visual Servoing of Mobile Robots Using Potential Fields, Master of Science Thesis, The Graduate School of Engineering and Natural Sciences of The Sabanci University, Istanbul, 105p.
- Bayram, Z. M. (2006) Ultrasonik Mesafe Algılayıcıları ile Mobil Robot Kontrolü, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 64s.
- Bekmen, O. (2006) A Software Environment for Behavior-Based Mobile Robot Control, Master of Science Thesis, The Graduate School of Natural and Applied Sciences of The Middle East Technical University, Ankara, 143p.
- Bicho, E., Moreira, A., Carvalheira, M., and Erhagen, W. (2005) Autonomous Flight Trajectory Generation via Attractor Dynamics, IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, p. 1379-1385.
- Bruijnen, D., Helvoort, J., and Molengraft, R. (2007) Realtime motion path generation using subtargets in a rapidly changing environment, *Robotics and Autonomous Systems*, 55: (6) 470-479.
- Chen, Y., Wang, Z. and Moore, K. L. (2006) Optimal Spraying Control of A Diffusion Process Using Mobile Actuator Networks With Fractional Potential Field Based Dynamic Obstacle Avoidance, ICNSC '06. Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, Florida, USA, p. 107-112.
- Çonkur, E. Ş. (2005) Path planning using potential fields for highly redundant manipulators, *Robotics and Autonomous Systems*, 52: 209-228.

- Çonkur, E. Ş. (1997) Real Time Path Planning And Obstacle Avoidance Algorithms For Redundant Manipulators, Doctor of Philosophy thesis, University of Bristol-Faculty of Engineering, 291p.
- Çonkur, E. Ş., Buckingham, R. and Harrison, A. (2005) The beam analysis algorithm for path planning for redundant manipulators, *Mechatronics*, 15, 67-94.
- Eroğlu, E. (2006) Gezin Robotlarda Ultrasonik Mesafe Algılayıcılarla Robot Davranışlarının Kontrolü ve Çevre Haritalama, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 99s.
- Fulgenzi, C., Spalanzani, A., and Laugier, C. (2007) Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid, 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, p. 1610-1616.
- Ge, S. S., and Cui, Y. J. (2002) Dynamic Motion Planning for Mobile Robots Using Potential Field Method, *Autonomous Robots*, 13: (3) 207-222.
- Guleç, N. (2005) Modeling and Control of The Coordinated Motion of a Group of Autonomous Mobile Robots, Master of Science Thesis, The Graduate School of Engineering and Natural Sciences of The Sabanci University, Istanbul, 137s.
- Gümrükçü, G. (2003) Improving The Accuracy of a Mobile Robot for Localization and Mapping of an Unknown Environment, Master of Science Thesis, The Graduate School of Natural and Applied Sciences of The Middle East Technical University, Ankara, 154p.
- Gündoğdu, M. (2006) Outdoor Navigation of Mobile Robots by Using Ultrasonic Sensors, Master of Science Thesis, The Institute of Science and Technology of The Istanbul Technical University, Istanbul, 70p.
- Gürel, U. (2006) Gezin Robotlar için Seyrüsefer ve Kamera Takip Sistemi, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 55s.
- Hwang, K. S., and Ming, Y. J. (2002) 3D Collision-Free Motion Based on Collision Index. *Journal of Intelligent and Robotic Systems*, 33: 45-60.
- Jing, X. J. (2005) Behavior Dynamics Based Motion Planning of Mobile Robots in Uncertain Dynamics Environments. *Robotics and Autonomous Systems*, 53: 99-123.
- Kaplan, K. (2003) Design and Implementation of Fast Controllers for Mobile Robots, Master of Science Thesis, The Institute for Graduate Studies in Sciences and Engineering of The Boğaziçi University, Istanbul, 61p.
- Karaman, Ö. (2004) Mobil Robotlarda Ultrasonik Sensörler ile Çevre Haritalama ve Navigasyon, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 90s.

- Khatib, O. (1986) Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal of Robotics Research*, 5: (10) 90-98.
- Küçükceylan, O. (2007) Gezin Robotların Engelli Bir Ortamda Hedefe Erişme Problemlerinin İncelenmesi, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü, Samsun, 99s.
- Lam, C. Y. (1994) *Applied Numerical Methods for Partial Differential Equations*, ISBN: 0130749230, Prentice Hall Southeast Asia Pte Ltd, London, 300p.
- Lebedev, D. V., Steil, J. J., and Ritter, H. J. (2005) The dynamic wave expansion neural network model for robot motion planning in time-varying environments, *Neural Networks*, 18: 267–285.
- Malhotra, R., and Sarkar, A. (2005) Development of a Fuzzy Logic Based Mobile Robot for Dynamic Obstacle Avoidance and Goal Acquisition in an Unstructured Environment, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, USA*, p. 1198-1203.
- Myers, T. J., Noel, T., Parent, M., and Vlacic, L. (2005) Autonomous Motion of a Driverless Vehicle Operating Among Dynamic Obstacles, *44th IEEE Conference on Decision and Control and the European Control Conference, Spain*, p. 5071-5076.
- Öktem, O. (2005) Mobil Robot Kontrolleri İçin Açık Platform Tasarımı, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü, Gebze, 122s.
- Özcan, S. (2005) Kapalı Mekanda Tarama Yapan Akıllı Robotun Tasarımı, Yüksek Lisans Tezi, Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü, Çanakkale, 103s.
- Özdemir, Y. (2006) Çizgi İzleyen Gezin Bir Robotun İncelenmesi ve Gerçeklenmesi, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü, Samsun, 89s.
- Özkan, M. (2007) Farklı Özelliklere Sahip Gezin Robot Grubunun Dağıtık Kontrolü, Doktora Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 134s.
- Saraç, Ö. S. (2002) A Behavior-Based Robot Control System Using Adaptive Heuristic Critic Learning, Master of Science Thesis, The Graduate School of Natural and Applied Sciences of The Middle East Technical University, Ankara, 71p.
- Sarı, S. (2006) Design and Simulation of a Mobile Tour-Guide Robot for Museum Guidance, Master of Science Thesis, The Institute for Graduate Studies in Science and Engineering of The Boğaziçi University, Istanbul, 101p.

- Shi, H., Sun, C., Sun, X., and Feng, T. (2006) Chaotic Potential Field Method and Application in Robot Soccer Game, The 6th World Congress on Intelligent Control and Automation, China, p. 9297-9301.
- Siegwart, R., and Nourbaksh, I. R. (2004) Introduction to Autonomous Mobile Robots, ISBN-13: 978-0262195027, The MIT Press, Cambridge-Massachusetts and London-England, 335p.
- Şengül, Ö. (2006) Mobil Robotlarda Laser Mesafe Ölçümü ile Harita Çıkarımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 48s.
- Şirin, M. (2005) Behavior-Based Fuzzy Control for a Mobile Robot with Non-Holonomic Constraints, Master of Science Thesis, The Institute of Science and Technology of The Istanbul Technical University, Istanbul, 70p.
- Türker, T. (2005) Tekerlekli Gezin Robotlarda Sistemik Odometri Hatalarının Belirlenmesi ve Azaltılması, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 86s.
- Uğur, E. (2006) Direct Perception of Traversibility Affordance on Range Images Through Learning on a Mobile Robot, Master of Science Thesis, The Graduate School of Natural and Applied Sciences of The Middle East Technical University, Ankara, 85p.
- Velagic, J., Lacevic, B., and Perunicic, B. (2006) A 3-level Autonomous Mobile Robot Navigation System Designed by Using Reasoning/Search Approaches. Robotics and Autonomous Systems, 54: 989-1004.
- WEB_1. (2005). Laurea Specialistica in Ingegneria Meccatronica.
<http://www.ladispe.polito.it/meccatronica/01CFI/2004-05/Slides/LabrobActivity.ppt> (05.08.2008).
- WEB_2. (2008). PLANETROLL.
<http://www.planetroll.com/pdf/PRODUKTKATALOG%20PD%20Nr.%2005%2004-01%2010.pdf> (05.01.2009).
- WEB_3. (2009). Welcome to Delta Electronics, Inc.
http://www.delta.com.tw/product/em/motion/motion_servo/download/manual/ASDA-B_manual_en.pdf (12.01.2009).
- WEB_4. (2008). Penn Engineering, Robotics Fall 2008 HomePage.
<http://fling-l.seas.upenn.edu/~cse390/wiki/?n=Main.Notes07?action=download&upname=Lecture06.pdf> (12.01.2009).
- WEB_5. (2008). JS Automation Corp.
http://ftp.automation.com.tw/download/Manual_Driver/MotionControlCard/MPC3034/manual/mpc3034e.pdf (12.01.2009).
- WEB_6. (2008). JS Automation Corp.

http://ftp.automation.com.tw/download/Manual_Driver/MotionControlCard/MPC3034/manual/sw3034.pdf (12.01.2009).

WEB_7. (2002). AVR222: 8-point Moving Average Filter.

http://www.atmel.com/dyn/resources/prod_documents/doc0940.pdf (15.05.2009).

WEB_8. (2009). Tunçmatik.

<http://www.tuncmatik.com/tr-TR/productDetail.asp?RecID=137> (05.05.2009).

WEB_9. (2009). Doç. Dr. E. Şahin Çonkur - Robot Yazılımı.

http://sconkur.pamukkale.edu.tr/zip_files/RoboKolRelease.zip (03.06.2009).

Willms, A. R. and Yang, S. X. (2008) Real-Time Robot Path Planning via a Distance-Propagating Dynamic System with Obstacle Clearance, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 38: (3) 884 - 893.

Wuwei, C., Mills, J. K., and Wenwu, S. (2004) A New Navigation Method for an Automatic Guided Vehicle. Journal of Robotic Systems, 21: (3) 129 - 139.

Xu, F., Brussel, H. V., Nuttin, M., and Moreas, R. (2003) Concepts for Dynamic Obstacle Avoidance and Their Extended Application in Underground Navigation. Robotics and Autonomous Systems, 42: 1-15.

Yalçın, M. (2004) Gezgin Robot İçin Yön Bulma Uygulaması, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 82s.

Yalçın, M. K. (2003) Mobil Robot Uygulaması, Yüksek Lisans Tezi, Niğde Üniversitesi Fen Bilimleri Enstitüsü, Niğde, 90s.

Yang, S. X., and Meng, M. (2000) An efficient neural network approach to dynamic robot motion planning. Neural Networks, 13: 143-148.

Yannier, S. (2002) Realization of Reactive Control for Multi Purpose Mobile Agents, Master of Science Thesis, The Graduate School of Engineering and Natural Sciences of The Sabanci University, Istanbul, 103p.

Yavaşoğlu, H. A. (2005) Bir Endüstriyel Taşıyıcı ve Servo Mekanizmalı Malzeme Değiştirici Mobil Robotun Gerçekleştirilmesi ve Bulanık Mantık Temelli Çizgi Takip Kontrolü, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 113s.

Ya-Chun, C. and Yamamoto, Y. (2006) Dynamic Decision Making of Mobile Robot under Obstructed Environment, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, p. 4091 - 4096.

Yılmaz, N. (2005) Web Tabanlı Mobil Robot Sistemi Tasarımı, Gerçekleştirilmesi ve Uygulamaları, Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, 150s.

- Yusufođlu, M. (2002) Combination of Behaviours in Autonomous Robot Assembly Game in 2D, Master of Science Thesis, The Institute for Graduate Studies in Sciences and Engineering of The Bođaziđi University, İstanbul, 70p.
- Yüksel, T. (2004) Özel Amaçlı Gezgin Robotların Denetimi, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü, Samsun, 94s.
- Zhao, H., Ji, Z., Luciw, M. D. and Weng, J. (2007) Developmental Learning for Avoiding Dynamic Obstacles Using Attention, ICDL 2007. IEEE 6th International Conference on Development and Learning, London, England, p. 318-323.
- Zheng, T. and Zhao, X. (2006) A Novel Approach for Multiple Mobile Robot Path Planning in Dynamic Unknown Environment, 2006 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand, p. 1-5.

EKLER

EK-1 İLERİ, GERİ VE MERKEZİ FARKLAR FORMÜLLERİNİN ÇIKARILIŞLARI

Chung - Yau Lam'ın kitabında(Chung 1994) ileri, geri ve merkezi farklar formülleri verilmiştir. Bu formüllerin bir iki tanesinin çıkarılışı dışında diğerlerinin çıkarılışı kitaba konulmamıştır. Bu formüllerin çoğunun çıkarılışı yapılabilmektedir. Aşağıda ilk önce formüller yazılacak daha sonra bunların bir kısmının çıkarılışı yapılacaktır. Taylor seri açılımından denklem (5.1) ve (5.2) elde edilmektedir.

$$u_{i+1,j} = u_{i,j} + h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} + \dots \quad (5.1)$$

$$u_{i-1,j} = u_{i,j} - h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \quad (5.2)$$

a) İleri Fark Formülleri (hata = $O(h)$)

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i,j}}{h} \quad (5.3)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+2,j} - 2u_{i+1,j} + u_{i,j}}{h^2} \quad (5.4)$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{u_{i+3,j} - 3u_{i+2,j} + 3u_{i+1,j} - u_{i,j}}{h^3} \quad (5.5)$$

$$\frac{\partial^4 u_{i,j}}{\partial x^4} = \frac{u_{i+4,j} - 4u_{i+3,j} + 6u_{i+2,j} - 4u_{i+1,j} + u_{i,j}}{h^4} \quad (5.6)$$

b) İleri Fark Formülleri (hata = $O(h^2)$)

$$\frac{\partial u_{i,j}}{\partial x} = \frac{-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}}{2h} \quad (5.7)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{-u_{i+3,j} + 4u_{i+2,j} - 5u_{i+1,j} + 2u_{i,j}}{h^2} \quad (5.8)$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{-3u_{i+4,j} + 14u_{i+3,j} - 24u_{i+2,j} + 18u_{i+1,j} - 5u_{i,j}}{2h^3} \quad (5.9)$$

$$\frac{\partial^4 u_{i,j}}{\partial x^4} = \frac{-2u_{i+5,j} + 11u_{i+4,j} - 24u_{i+3,j} + 26u_{i+2,j} - 14u_{i+1,j} + 3u_{i,j}}{h^4} \quad (5.10)$$

c) Geri Fark Formülleri (hata = $O(h)$)

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i,j} - u_{i-1,j}}{h} \quad (5.11)$$

$$\frac{\partial u_{i,j}^2}{\partial x^2} = \frac{u_{i,j} - 2u_{i-1,j} + u_{i-2,j}}{h^2} \quad (5.12)$$

$$\frac{\partial u_{i,j}^3}{\partial x^3} = \frac{u_{i,j} - 3u_{i-1,j} + 3u_{i-2,j} - u_{i-3,j}}{h^3} \quad (5.13)$$

$$\frac{\partial u_{i,j}^4}{\partial x^4} = \frac{u_{i,j} - 4u_{i-1,j} + 6u_{i-2,j} - 4u_{i-3,j} + u_{i-4,j}}{h^4} \quad (5.14)$$

d) Geri Fark Formülleri (hata = $O(h^2)$)

$$\frac{\partial u_{i,j}}{\partial x} = \frac{3u_{i,j} - 4u_{i-1,j} + u_{i-2,j}}{2h} \quad (5.15)$$

$$\frac{\partial u_{i,j}^2}{\partial x^2} = \frac{2u_{i,j} - 5u_{i-1,j} + 4u_{i-2,j} - u_{i-3,j}}{h^2} \quad (5.16)$$

$$\frac{\partial u_{i,j}^3}{\partial x^3} = \frac{5u_{i,j} - 18u_{i-1,j} + 24u_{i-2,j} - 14u_{i-3,j} + u_{i-4,j}}{2h^3} \quad (5.17)$$

$$\frac{\partial u_{i,j}^4}{\partial x^4} = \frac{3u_{i,j} - 14u_{i-1,j} + 26u_{i-2,j} - 24u_{i-3,j} + 11u_{i-4,j} - 2u_{i-5,j}}{h^4} \quad (5.18)$$

e) Merkezi Fark Formülleri (hata = $O(h^2)$)

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} \quad (5.19)$$

$$\frac{\partial u_{i,j}^2}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad (5.20)$$

$$\frac{\partial u_{i,j}^3}{\partial x^3} = \frac{u_{i+2,j} - 2u_{i+1,j} + 2u_{i-1,j} - u_{i-2,j}}{2h^3} \quad (5.21)$$

$$\frac{\partial u_{i,j}^4}{\partial x^4} = \frac{u_{i+2,j} - 4u_{i+1,j} + 6u_{i,j} - 4u_{i-1,j} + u_{i-2,j}}{h^4} \quad (5.22)$$

f) Merkezi Fark Formülleri (hata = $O(h^4)$)

$$\frac{\partial u_{i,j}}{\partial x} = \frac{-u_{i+2,j} + 8u_{i+1,j} - 8u_{i-1,j} + u_{i-2,j}}{12h} \quad (5.23)$$

$$\frac{\partial^2 u_{i,j}^2}{\partial x^2} = \frac{-u_{i+2,j} + 16u_{i+1,j} - 30u_{i,j} + 16u_{i-1,j} - u_{i-2,j}}{12h^2} \quad (5.24)$$

$$\frac{\partial^3 u_{i,j}^3}{\partial x^3} = \frac{-u_{i+3,j} + 8u_{i+2,j} - 13u_{i+1,j} + 13u_{i-1,j} - 8u_{i-2,j} + u_{i-3,j}}{8h^3} \quad (5.25)$$

$$\frac{\partial^4 u_{i,j}^4}{\partial x^4} = \frac{-u_{i+3,j} + 12u_{i+2,j} - 39u_{i+1,j} + 56u_{i,j} - 39u_{i-1,j} + 12u_{i-2,j} - u_{i-3,j}}{6h^4} \quad (5.26)$$

Çıkarılışlar;

(5.1)'den;

$$u_{i+1,j} = u_{i,j} + h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} + \dots$$

$$h \frac{\partial u_{i,j}}{\partial x} = u_{i+1,j} - u_{i,j} - \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \dots$$

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i,j}}{h} - \frac{h}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \dots$$

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i,j}}{h} + O(h) \quad (5.3)$$

(5.3)'ten;

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i,j}}{h} + O(h)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{1}{h} \left(\frac{\partial u_{i+1,j}}{\partial x} - \frac{\partial u_{i,j}}{\partial x} \right) + O(h)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{1}{h} \left(\frac{u_{i+2,j} - u_{i+1,j}}{h} + O(h) - \left(\frac{u_{i+1,j} - u_{i,j}}{h} + O(h) \right) \right) + O(h)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{1}{h} \left(\frac{u_{i+2,j} - 2u_{i+1,j} + u_{i,j}}{h} \right) + O(h)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+2,j} - 2u_{i+1,j} + u_{i,j}}{h^2} + O(h) \quad (5.4)$$

(5.4)'ten;

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+2,j} - 2u_{i+1,j} + u_{i,j}}{h^2} + O(h)$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{1}{h^2} \left(\frac{\partial u_{i+2,j}}{\partial x} - 2 \frac{\partial u_{i+1,j}}{\partial x} + \frac{\partial u_{i,j}}{\partial x} \right) + O(h)$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{1}{h^2} \left(\frac{u_{i+3,j} - u_{i+2,j}}{h} + O(h) - 2 \left(\frac{u_{i+2,j} - u_{i+1,j}}{h} + O(h) \right) + \frac{u_{i+1,j} - u_{i,j}}{h} + O(h) \right) + O(h)$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{1}{h^2} \left(\frac{u_{i+3,j} - 3u_{i+2,j} + 3u_{i+1,j} - u_{i,j}}{h} \right) + O(h)$$

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{u_{i+3,j} - 3u_{i+2,j} + 3u_{i+1,j} - u_{i,j}}{h^3} + O(h) \quad (5.5)$$

(5.5)'ten;

$$\frac{\partial^3 u_{i,j}}{\partial x^3} = \frac{u_{i+3,j} - 3u_{i+2,j} + 3u_{i+1,j} - u_{i,j}}{h^3} + O(h)$$

$$\frac{\partial^4 u_{i,j}}{\partial x^4} = \frac{1}{h^3} \left(\frac{\partial u_{i+3,j}}{\partial x} - 3 \frac{\partial u_{i+2,j}}{\partial x} + 3 \frac{\partial u_{i+1,j}}{\partial x} - \frac{\partial u_{i,j}}{\partial x} \right) + O(h)$$

$$\frac{\partial^4 u_{i,j}}{\partial x^4} = \frac{1}{h^3} \left(\frac{u_{i+4,j} - u_{i+3,j}}{h} + O(h) - 3 \left(\frac{u_{i+3,j} - u_{i+2,j}}{h} + O(h) \right) + 3 \left(\frac{u_{i+2,j} - u_{i+1,j}}{h} + O(h) \right) - \left(\frac{u_{i+1,j} - u_{i,j}}{h} + O(h) \right) \right) + O(h)$$

$$\frac{\partial^4 u_{i,j}}{\partial x^4} = \frac{1}{h^3} \left(\frac{u_{i+4,j} - 4u_{i+3,j} + 6u_{i+2,j} - 4u_{i+1,j} + u_{i,j}}{h} \right) + O(h)$$

$$\frac{\partial^4 u_{i,j}}{\partial x^4} = \frac{u_{i+4,j} - 4u_{i+3,j} + 6u_{i+2,j} - 4u_{i+1,j} + u_{i,j}}{h^4} + O(h) \quad (5.6)$$

(5.1)'den;

$$\begin{aligned}
u_{i+1,j} &= u_{i,j} + h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} + \dots \\
h \frac{\partial u_{i,j}}{\partial x} &= u_{i+1,j} - u_{i,j} - \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} - \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i,j}}{h} - \frac{h}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^2}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} - \frac{h^3}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i,j}}{h} - \frac{h}{2!} \left(\frac{u_{i+2,j} - 2u_{i+1,j} + u_{i,j}}{h^2} \right) + O(h^2) \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{2u_{i+1,j} - 2u_{i,j} - u_{i+2,j} + 2u_{i+1,j} - u_{i,j}}{2h} + O(h^2) \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}}{2h} + O(h^2)
\end{aligned} \tag{5.7}$$

(5.2)'den;

$$\begin{aligned}
u_{i-1,j} &= u_{i,j} - h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
h \frac{\partial u_{i,j}}{\partial x} &= u_{i,j} - u_{i-1,j} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h} + \frac{h}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^2}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^3}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h} + O(h)
\end{aligned} \tag{5.11}$$

(5.11)'den;

$$\begin{aligned}
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h} + O(h) \\
\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{1}{h} \left(\frac{\partial u_{i,j}}{\partial x} - \frac{\partial u_{i-1,j}}{\partial x} \right) + O(h)
\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{1}{h} \left(\frac{u_{i,j} - u_{i-1,j}}{h} + O(h) - \left(\frac{u_{i-1,j} - u_{i-2,j}}{h} + O(h) \right) \right) + O(h) \\ \frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{1}{h} \left(\frac{u_{i,j} - 2u_{i-1,j} + u_{i-2,j}}{h} \right) + O(h) \\ \frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i,j} - 2u_{i-1,j} + u_{i-2,j}}{h^2} + O(h)\end{aligned}\quad (5.12)$$

(5.12)'den;

$$\begin{aligned}\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i,j} - 2u_{i-1,j} + u_{i-2,j}}{h^2} + O(h) \\ \frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{1}{h^2} \left(\frac{\partial u_{i,j}}{\partial x} - 2 \frac{\partial u_{i-1,j}}{\partial x} + \frac{\partial u_{i-2,j}}{\partial x} \right) + O(h) \\ \frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{1}{h^2} \left(\frac{u_{i,j} - u_{i-1,j}}{h} + O(h) - 2 \left(\frac{u_{i-1,j} - u_{i-2,j}}{h} + O(h) \right) + \frac{u_{i-2,j} - u_{i-3,j}}{h} + O(h) \right) + O(h) \\ \frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{u_{i,j} - 3u_{i-1,j} + 3u_{i-2,j} - u_{i-3,j}}{h^3} + O(h)\end{aligned}\quad (5.13)$$

(5.13)'ten;

$$\begin{aligned}\frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{u_{i,j} - 3u_{i-1,j} + 3u_{i-2,j} - u_{i-3,j}}{h^3} + O(h) \\ \frac{\partial^4 u_{i,j}}{\partial x^4} &= \frac{1}{h^3} \left(\frac{\partial u_{i,j}}{\partial x} - 3 \frac{\partial u_{i-1,j}}{\partial x} + 3 \frac{\partial u_{i-2,j}}{\partial x} - \frac{\partial u_{i-3,j}}{\partial x} \right) + O(h) \\ \frac{\partial^4 u_{i,j}}{\partial x^4} &= \frac{1}{h^3} \left(\frac{u_{i,j} - u_{i-1,j}}{h} + O(h) - 3 \left(\frac{u_{i-1,j} - u_{i-2,j}}{h} + O(h) \right) + 3 \left(\frac{u_{i-2,j} - u_{i-3,j}}{h} + O(h) \right) \right. \\ &\quad \left. - \left(\frac{u_{i-3,j} - u_{i-4,j}}{h} + O(h) \right) \right) + O(h) \\ \frac{\partial^4 u_{i,j}}{\partial x^4} &= \frac{u_{i,j} - 4u_{i-1,j} + 6u_{i-2,j} - 4u_{i-3,j} + u_{i-4,j}}{h^4} + O(h)\end{aligned}\quad (5.14)$$

(5.2)'den;

$$\begin{aligned}
u_{i-1,j} &= u_{i,j} - h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
h \frac{\partial u_{i,j}}{\partial x} &= u_{i,j} - u_{i-1,j} + \frac{h^2}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h} + \frac{h}{2!} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^2}{3!} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^3}{4!} \frac{\partial^4 u_{i,j}}{\partial x^4} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i,j} - u_{i-1,j}}{h} + \frac{h}{2!} \left(\frac{u_{i,j} - 2u_{i-1,j} + u_{i-2,j}}{h^2} + O(h) \right) + O(h^2) \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{2u_{i,j} - 2u_{i-1,j} + u_{i-2,j}}{2h} + O(h^2) \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{3u_{i,j} - 4u_{i-1,j} + u_{i-2,j}}{2h} + O(h^2)
\end{aligned} \tag{5.15}$$

(5.1) - (5.2)'den;

$$\begin{aligned}
u_{i+1,j} - u_{i-1,j} &= 2h \frac{\partial u_{i,j}}{\partial x} + \frac{h^3}{3} \frac{\partial^3 u_{i,j}}{\partial x^3} + \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i-1,j}}{2h} - \frac{h^2}{6} \frac{\partial^3 u_{i,j}}{\partial x^3} - \dots \\
\frac{\partial u_{i,j}}{\partial x} &= \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2)
\end{aligned} \tag{5.19}$$

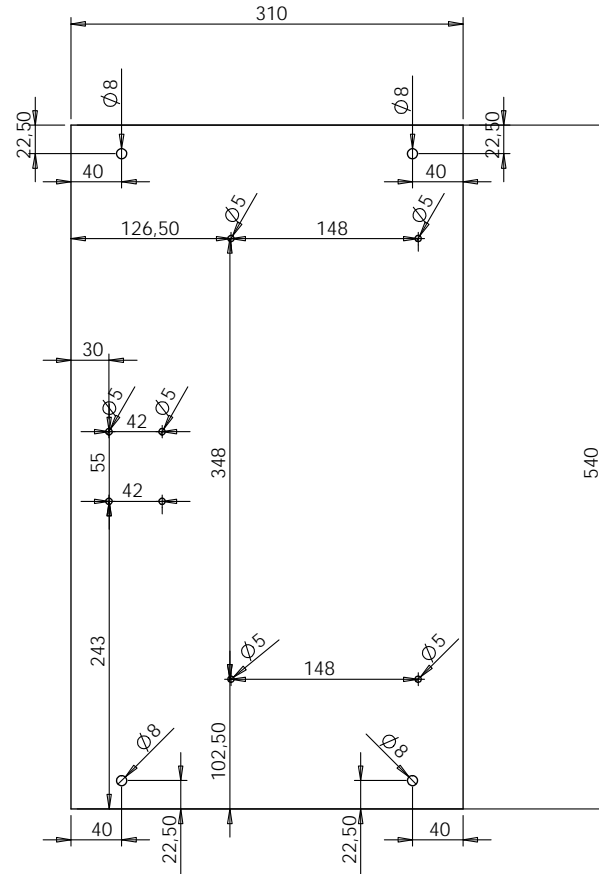
(5.1) + (5.2)'den;

$$\begin{aligned}
u_{i+1,j} + u_{i-1,j} &= 2u_{i,j} + h^2 \frac{\partial^2 u_{i,j}}{\partial x^2} + \dots \\
\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h^2} + O(h^2)
\end{aligned} \tag{5.20}$$

(5.20)'den;

$$\begin{aligned}
\frac{\partial^2 u_{i,j}}{\partial x^2} &= \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h^2} + O(h^2) \\
\frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{1}{h^2} \left(\frac{\partial u_{i+1,j}}{\partial x} - 2 \frac{\partial u_{i,j}}{\partial x} + \frac{\partial u_{i-1,j}}{\partial x} \right) + O(h^2) \\
\frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{1}{h^2} \left(\frac{u_{i+2,j} - u_{i,j}}{2h} - 2 \left(\frac{u_{i+1,j} - u_{i-1,j}}{2h} \right) + \frac{u_{i,j} - u_{i-2,j}}{2h} \right) + O(h^2) \\
\frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{1}{h^2} \left(\frac{u_{i+2,j} - u_{i,j} - 2u_{i+1,j} + 2u_{i-1,j} + u_{i,j} - u_{i-2,j}}{2h} \right) + O(h^2) \\
\frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{1}{h^2} \left(\frac{u_{i+2,j} - 2u_{i+1,j} + 2u_{i-1,j} - u_{i-2,j}}{2h} \right) + O(h^2) \\
\frac{\partial^3 u_{i,j}}{\partial x^3} &= \frac{u_{i+2,j} - 2u_{i+1,j} + 2u_{i-1,j} - u_{i-2,j}}{2h^3} + O(h^2)
\end{aligned} \tag{5.21}$$

EK-2 MOBİL ROBOTUN DESTEK VE BAĞLANTI PARÇALARININ TEKNİK RESİMLERİ
Ek Şekil 1 Ups destek parçası



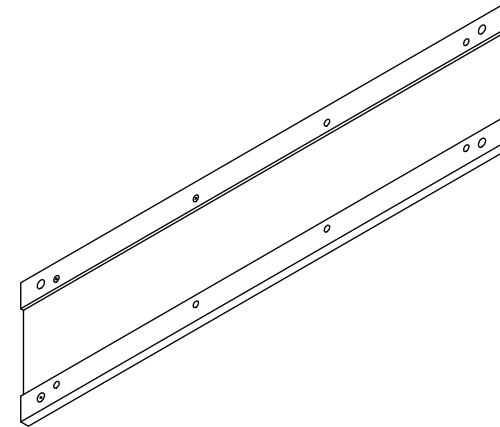
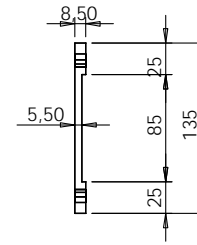
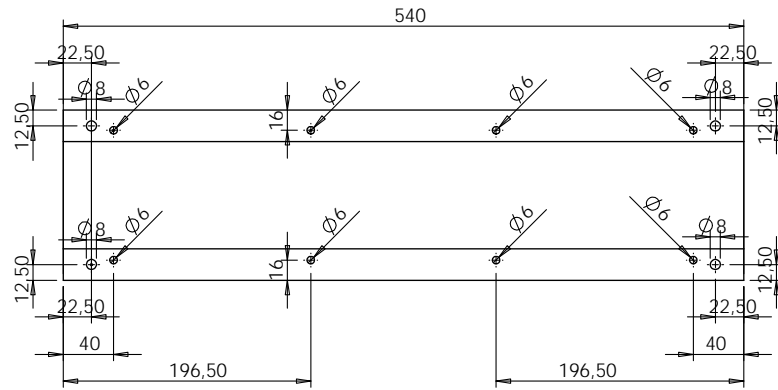
kalınlık 10 mm

PARÇA:

upsdestek

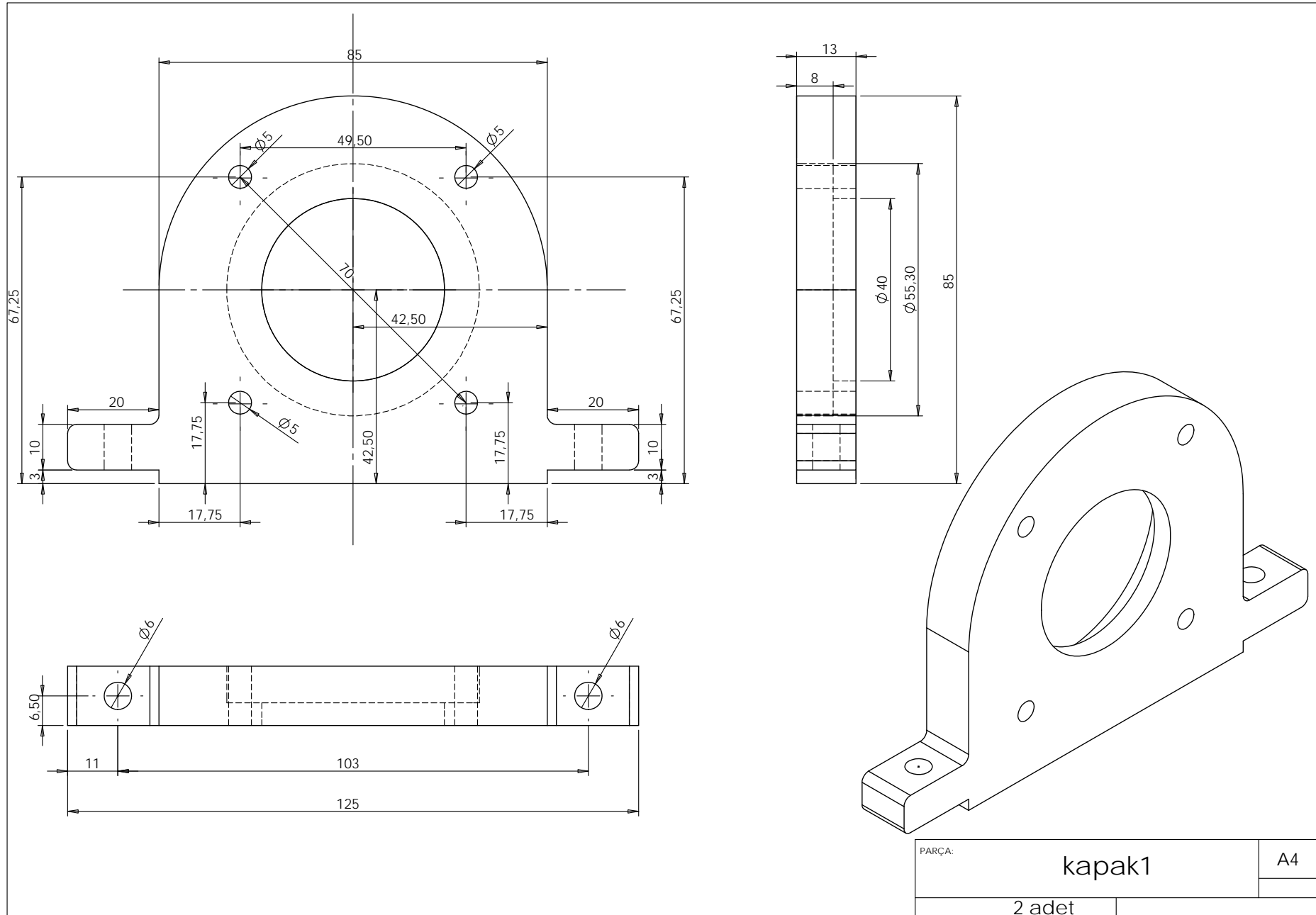
A4

Ek Şekil 2 Redüktörlerin ve motorların alt parçası

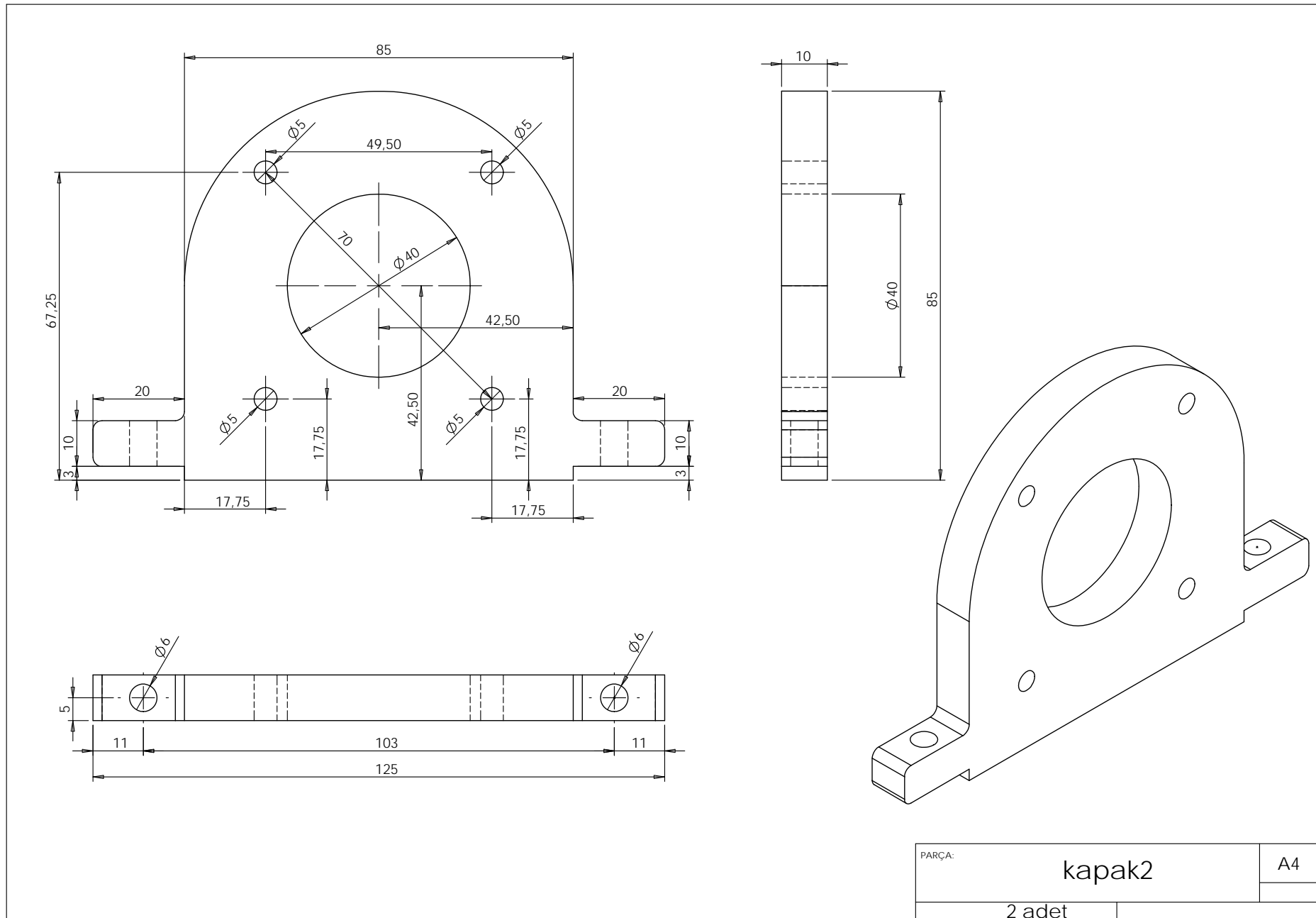


PARÇA:	altparca	A4

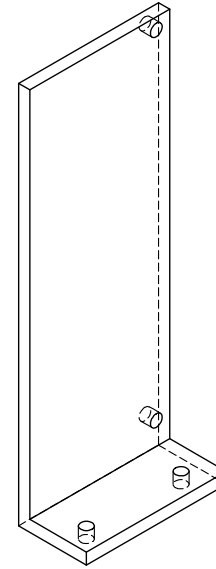
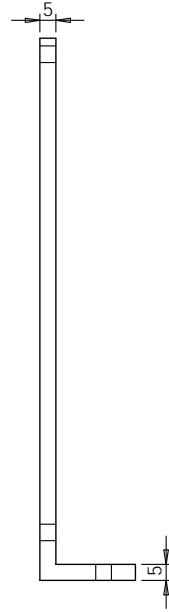
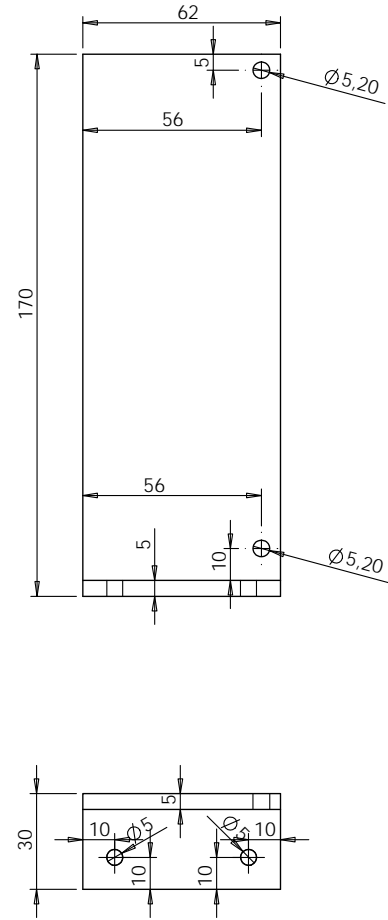
Ek Şekil 3 Redüktörün destek parçası1



Ek Şekil 4 Redüktörün destek parçası2

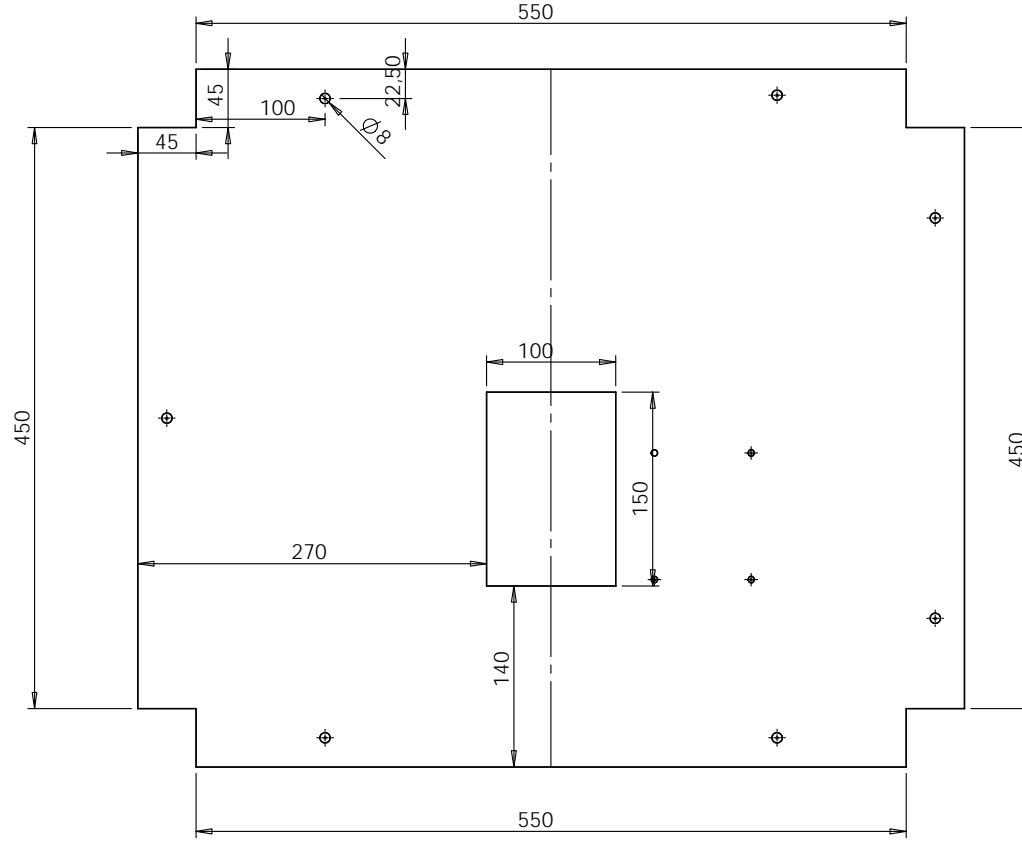


Ek Şekil 5 Driver'ların destek parçası



PARÇA:	driverdestek	A4
	2 adet	

Ek Şekil 6 Üst Levha



PARÇA:

üst levha

A4

EK-3 HAREKET KONTROL KARTINA AİT FONKSİYONLAR

- 1) MPC3034_initial()
- 2) MPC3034_close()
- 3) MPC3034_init_card()
- 4) MPC3034_info()
- 5) MPC3034_dll_Simu_mode()
- 6) MPC3034_set_pulse_outmode()
- 7) MPC3034_readback_pulse_outmode()
- 8) MPC3034_set_pulse_inmode()
- 9) MPC3034_readback_pulse_inmode()
- 10) MPC3034_config_SD_PIN()
- 11) MPC3034_readback_SD_PIN()
- 12) MPC3034_config_PCS_PIN()
- 13) MPC3034_readback_PCS_PIN()
- 14) MPC3034_config_INP_PIN()
- 15) MPC3034_readback_INP_PIN()
- 16) MPC3034_config_ERC_PIN()
- 17) MPC3034_readback_ERC_PIN()
- 18) MPC3034_config_ALM_PIN()
- 19) MPC3034_readback_ALM_PIN()
- 20) MPC3034_config_LTC_PIN()
- 21) MPC3034_readback_LTC_PIN()
- 22) MPC3034_config_CMP_OUT()
- 23) MPC3034_readback_CMP_OUT()
- 24) MPC3034_config_EL_MODE()
- 25) MPC3034_readback_EL_MODE()
- 26) MPC3034_config_TTL_IO_MODE()
- 27) MPC3034_readback_TTL_IO_MODE()
- 28) MPC3034_set_HOME_pin_logic()
- 29) MPC3034_readback_HOME_pin_logic()
- 30) MPC3034_set_EZ_pin_logic()
- 31) MPC3034_readback_EZ_pin_logic()
- 32) MPC3034_read_point_status()

- 33) MPC3034_write_output_point()
- 34) MPC3034_save_config2_file()
- 35) MPC3034_load_config_from_file()
- 36) MPC3034_fix_speed_range()
- 37) MPC3034_unfix_speed_range()
- 38) MPC3034_T_velocity_move()
- 39) MPC3034_S_velocity_move()
- 40) MPC3034_velocity_change()
- 41) MPC3034_dec_stop()
- 42) MPC3034_imd_stop()
- 43) MPC3034_read_speed()
- 44) MPC3034_config_home_mode()
- 45) MPC3034_start_homing()
- 46) MPC3034_set_current_position()
- 47) MPC3034_read_current_position()
- 48) MPC3034_start_origin_search_homing()
- 49) MPC3034_T_curve_position_move()
- 50) MPC3034_S_curve_position_move()
- 51) MPC3034_position_change()
- 52) MPC3034_backlash_comp()
- 53) MPC3034_readback_backlash_comp()
- 54) MPC3034_suppress_vibration()
- 55) MPC3034_readback_suppress_vibration()
- 56) MPC3034_T_curve_move_LINE2()
- 57) MPC3034_S_curve_move_LINE2()
- 58) MPC3034_T_curve_move_LINE3()
- 59) MPC3034_S_curve_move_LINE3()
- 60) MPC3034_T_curve_move_LINE4()
- 61) MPC3034_S_curve_move_LINE4()
- 62) MPC3034_ARC2_center_move()
- 63) MPC3034_ARC2_3P_move()
- 64) MPC3034_CIR2_3P_move()
- 65) MPC3034_T_CIR2_3P_move()
- 66) MPC3034_S_CIR2_3P_move()

67) MPC3034_set_continuous_flag()
68) MPC3034_check_continuous_buffer()
69) MPC3034_read_conti_buffer_no()
70) MPC3034_read_motion_status()
71) MPC3034_set_event_factor()
72) MPC3034_read_event_flag()
73) MPC3034_read_error_flag()
74) MPC3034_OnLine_T_curve_change()
75) MPC3034_OnLine_T_curve_change_LINE2()
76) MPC3034_OnLine_T_curve_change_LINE3()
77) MPC3034_OnLine_T_curve_change_LINE4()
78) MPC3034_OneAxis_restart()
79) MPC3034_2Axis_restart()
80) MPC3034_3Axis_restart()
81) MPC3034_4Axis_restart()
82) MPC3034_enable_FIFO()
83) MPC3034_check_FIFO_buffer()
84) MPC3034_T_curve_write_FIFO()
85) MPC3034_T_LINE2_write_FIFO()
86) MPC3034_T_LINE3_write_FIFO()
87) MPC3034_T_LINE4_write_FIFO()
88) MPC3034_Run_FIFO_CMD()
89) MPC3034_FIFO_EOI()
90) MPC3034_set_FIFO_out_ratio()
91) MPC3034_config_softlimit()
92) MPC3034_readback_config_softlimit()
93) MPC3034_set_softlimit_data()
94) MPC3034_readback_softlimit_data()
95) MPC3034_enable_softlimit()
96) MPC3034_readback_enable_softlimit()
97) MPC3034_read_softlimit_flag()
98) MPC3034_config_pulser_mode()
99) MPC3034_readback_pulser_mode()
100) MPC3034_run_pulser_Vmove()

- 101) MPC3034_run_pulser_Pmove()
- 102) MPC3034_set_pulser_counter()
- 103) MPC3034_read_pulser_counter()
- 104) MPC3034_set_pulser_Map()
- 105) MPC3034_enable_pulser_motion()
- 106) MPC3034_read_FB_counter()
- 107) MPC3034_set_FB_counter()
- 108) MPC3034_read_FBcounter_latch_value()
- 109) MPC3034_config_comparator_out()
- 110) MPC3034_config_comparator_out_W()
- 111) MPC3034_readback_comparator_out()
- 112) MPC3034_readback_comparator_out_W()
- 113) MPC3034_set_comparator_data()
- 114) MPC3034_set_comparator_data_W()
- 115) MPC3034_readback_comparator_data()
- 116) MPC3034_readback_comparator_data_W()
- 117) MPC3034_read_compare_flag()
- 118) MPC3034_read_compare_flag_W()
- 119) MPC3034_set_password()
- 120) MPC3034_change_password()
- 121) MPC3034_clear_password()
- 122) MPC3034_unlock_security()
- 123) MPC3034_read_security_status()
- 124) MPC3034_set_serial_code()
- 125) MPC3034_read_serial_code()

ÖZGEÇMİŞ

Adı Erdem

Soyadı Yıldız

Doğum yeri ve Tarihi İzmir 16.06.1984

Lisans Pamukkale Üniversitesi, Elektrik-Elektronik Mühendisliği 2006