

**FARKLI DETEKTÖRLER TAŞIYAN
ROBOTLARIN KARA MAYINI TESPİT
ALGORİTMASININ GELİŞTİRİLMESİ**

ALPER TOL

**MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANA BİLİM DALI**

YÜKSEK LİSANS TEZİ

**MERSİN
EKİM-2007**

**FARKLI DETEKTÖRLER TAŞIYAN ROBOTLARIN KARA
MAYINI TESPİT ALGORİTMASININ GELİŞTİRİLMESİ**

ALPER TOL

**Mersin Üniversitesi
Fen Bilimleri Enstitüsü**

**Elektrik Elektronik Mühendisliği
Ana Bilim Dalı**

YÜKSEK LİSANS TEZİ

**Tez Danışmanı
Yrd.Doç. Dr. Hüseyin CANBOLAT**

**MERSİN
EKİM-2007**

Bu tezin gerek bilimsel içerik, gerekse elde edilen sonuçlar açısından tüm gerekleri sağladığı kanaatine ulasan ve aşağıda imzaları bulunan biz jüri üyeleri, sunulan tezi oy birliği ile Yüksek Lisans Tezi olarak kabul ediyoruz.

Tez Danışmanı
Yrd. Doc. Dr. Hüseyin CANBOLAT

Jüri Üyesi
Yrd.Doc. Dr. Ali AKDAĞLI

Jüri Üyesi
Yrd. Doc. Dr. Hüseyin MUTLU

Bu tezin Fen Bilimleri Enstitüsü yazım kurallarına uygun olarak yazıldığı Enstitü Yönetim Kurulu'nun/...../..... tarih ve/..... sayılı kararıyla onaylanmıştır.

Pof.Dr.Mahir TURHAN
Enstitü Müdürü

Not: Bu tezde kullanılan özgün bilgiler, şekil, çizelge ve fotoğraflardan kaynak göstermeden alıntı yapmak 5846 sayılı Fikir ve Sanat Eserleri Kanunu hükümlerine tabidir.

ÖZ

Bu çalışmada Metal ve GPR mayın arama detektörlerine sahip robotların kullanıcı tarafından belirlenen bir alanın robotlar arasındaki görev tanımlarına ve sıralamalarına uygun olarak incelemesi için gerekli algoritma ve bunun test edilmesini görsel olarak sunumunu sağlayacak programlama oluşturulmuştur.

Mayınlı alan taramasının en temel gereksinimi olan bütün alanın taranmasına olanak sağlamak amacıyla algoritma kapsama yol planlamasına göre tasarlanmıştır. Ana alan kullanıcı tarafından 4'ün katları şeklinde belirlenebilmektedir. Belirlenen ana alan program tarafından 4x4'lük alt alanlara bölünmektedir. Arama belirlenen 4x4'lük alt alanlar temel alınarak yapılmaktadır. 4x4'lük alt alanlar üzerinden yapılan benzetimler alanın bütününe yansıtılarak bütünlük sağlanmaktadır. Alt alanlar arasındaki geçişler satır veya sütun kaydırılması sonucu oluşturulan geçici alanlar ile yine 4x4'lük alanlar üzerinden sağlanmaktadır. Oluşturulan algorithmada tanımlanan alandaki bütün hücelere uygun veri atanmasının sağlanması amaçlanmaktadır. Bu amaç doğrultusunda yapılacak hareketlere güvenlik açısından sınırlamalar konulmuştur. Öncelikle robotların ayak izleri sadece temiz olan güvenli hücelerde bulunabilmektedir. Robotlar ayak izlerinin komşuluğundaki kareleri tarayabilmektedir. Kontrol edilen hüceler temiz, şüpheli ve riskli olarak belirlenmektedir. Kullanılan robotlar birbirleriyle iletişim içinde olup belirlenen şartlara göre görev tanımlarına uygun şekilde birbirlerini harekete geçirebilmekte ve yönlendirebilmektedir.

Robotların hareket doğrultuları, kontrol edilmemiş hücelere doğru yönelmesini sağlayan bir potansiyel hesaplaması yöntemiyle belirlenmektedir. Potansiyel hesaplaması robotun bulunduğu hücreye göre her bir yöndeki hücelere üzerinden yönsel hesaplama ile yapılmaktadır. Mayın taraması için belirlenen alandaki bütün hücelere veri atanması yapıldığında tarama işlemi sonuçlanmaktadır.

Anahtar Kelimeler: Mayın, Robot, Mayın Tarama, Detektörler, Algoritma.

ABSTRACT

In this study an algorithm that enables the analysis of a field by the robots that have metal and GPR detectors and software for the visual testing of the algorithm are constructed. The robots conduct the analysis of the field consistently with the job descriptions and rankings among them. An algorithm is designed based on the coverage path planning in order to be able to scan the entire field, which is the basic requirement for mined area detection.

The main area can be determined as the multiples of 4 by the user. The determined main area is divided into 4x4 subareas by the software and the scanning is conducted based on the 4x4 subareas. Integrity is acquired through the application of the simulation based on the 4x4 subareas on the entire area. The movement between the subareas is achieved through the temporary areas that are created by shifting the rows or the columns. The constructed algorithm aims to provide cell definition for all the cells in the defined area. The movements that are performed to accomplish this task face some limitations due to safety reasons. Firstly, the foot prints of the robots can only be found in the clean and safe cells. The robots are able to scan the squares neighboring their foot prints. The cells that are controlled will be labeled as clean, suspicious and risky. The robots that are used is in communication with each other and depending the conditions determined they are able to activate each other and be directed as well. The robots determine their movement direction by a potential calculation technique that enables them to move towards uncontrolled cells.

The potential calculation is the directional calculation which is made by looking at the location of the cells in each direction vis-a- vis the cell where the robot is. The scanning process is completed when all cell definition in the area that is determined for mine scanning is complete.

Key Words: Landmine, Robot, Landmine Detection, Detectors, Algorithm.

TEŐEKKÜR

Bu alıőmada öncelikle yardımlarından dolayı danışmanım Yrd.Do.Dr. Hüseyin CANBOLAT'a, her zaman yanımda olan aileme, maddi manevi desteęini esirgemeyerek bu amaca ilerlememi teşvik etmesi ve bu yoğun alıőmada gösterdięi anlayıő ve destekle hep yanımda olmasından dolayı bilhassa eőime teşekkür ediyorum.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vi
KISALTMALAR DİZİNİ	viii
1.GİRİŞ	1
2.KAYNAK ARAŞTIRMALARI	4
2.1. MAYIN ÇEŞİTLERİ	4
2.1.1. Anti Personel Mayınları	4
2.1.2. Anti-Tank Mayınları	8
2.2 MAYIN TESPİT YÖNTEMLERİ	11
2.2.1 Patlayıcıyı Tanımayan Belirleme Yöntemi	12
2.2.1.1 Manyeto statik ve elektro statik	12
2.2.1.2 Doğru kım (DC) iletim metodu	13
2.2.1.3 Elektromanyetik indüksiyon tekniği	13
2.2.1.4 Mikrodalga tekniği	14
2.2.1.5 Optik teknikler	16
2.2.1.7 Ses ötesi tespit	20
2.2.2 Patlayıcı tespitiyle belirleme tekniği	22
2.2.2.1 Radyo frekansı rezonans soğurma spektroskopisi (RRAS)	22
2.2.2.2 Çekirdek dörtlük rezonansı	23
2.2.2.3 Nükleer ışınım metodu	24
3.MATERYAL VE METOT	30
3.1 ÇOKLU DETEKTÖR KULLANIMI	30
3.2 ROBOTLARIN KULLANIMI	37
3.2.1 Robotla Arama Yönteminde Kullanılan Metotlar	41
3.2.1.1. Yol planlamaları	41
3.2.1.2 Robotik kara mayını araştırması için probalistik yöntemler	41
3.2.1.3 Robot öngörülen bağımsız hareketi	46
4. BULGULAR VE TARTIŞMASI	48

4.1 GUI MODÜLÜ	48
4.2 MAYINTARLASI MODÜLÜ	48
4.3 UYGULAMA MODÜLÜ	49
4.4 TEMEL HAREKETLER	56
5. SONUÇLAR VE ÖNERİLER	61
KAYNAKLAR	63
EKLER:	67

ŞEKİLLER DİZİNİ

ŞEKİL	SAYFA
Şekil 2.1. Tip 72 Plastik AP mayını.....	5
Şekil 2.2. Solda PMN AP mayının ve PMN2 AP mayınının metal içerikleri	5
Şekil 2.3. M14 Plastik AP mayını ve kesiti.	6
Şekil 2.4. M-16 AP mayını ve kesiti.....	6
Şekil 2.5. BLU43B AP mayını.....	7
Şekil 2.6. PMD 6 AP mayını.....	8
Şekil 2.7. M86 PDM mayını (Pursuit Deterrent Munition)	8
Şekil 2.8. M15 AT Mayını	9
Şekil 2.9. M15 AT mayının toprağa yerleştirilmiş hali	10
Şekil 2.10. M19 AT mayını.	10
Şekil 2.11. Toprağa yerleştirilmesinin üzerinden zaman geçmiş bir AT mayını.....	11
Şekil 2.12. Tipik Elektro Manyetik İndüksiyon Sistemi.....	14
Şekil 2.13. GPR sistemi tarafından oluşturulan görüntü.....	15
Şekil 2.14. 5 GHz kullanılarak elde edilen görüntü.....	16
Şekil 2.15. Gündüz (14:15) IR görüntüsü	17
Şekil 2.16. Gece (04:45) IR görüntüsü.....	18
Şekil 2.17. Yandaki şeklin solundaki IR sağındaki görünür dalga boylarının.....	19
Şekil 2.18. Su içindeki AP mayının 2D görüntüsü.	20
Şekil 2.19. Su içindeki AP mayının 3D görüntüsü	21
Şekil 2.20. Bahçe toprağında 5cm derinliğe gömülmüş plastik mayının.....	22
Şekil 2.21. TNA Detektörü	25
Şekil 2.22. Geri saçılım detektörü şematik diyagramı	27
Şekil 3.1 Mayın Tarama Aracı alan çalışması sırasında	32
Şekil 3.2 Sıralı metal detektörden alınan görüntüler.....	32
Şekil 3.3 Kum içine yerleştirilmiş 3 madde.....	33
Şekil 3.4 1/1.5/2 ns'deki kum içindeki AP mayının görüntüleri.	33
Şekil 3.5 3/3.5/4 ns'deki kum içindeki taşın görüntüleri.	34
Şekil 3. 6 Metal detektörünün taş ve AP mayını görüntüleri.....	35

Şekil 3.7 Metal detektöründeki 3.5g'lık metal parçasının görüntüsü.	35
Şekil 3.8 3 objenin Metal Detektörüne ve GPR detektöründeki görüntüleri	36
Şekil 3.9 Katmanlı Mobil Robot mimarisi.....	41
Şekil 3.10 Eşit aralıklı düzenli paterne örnek.	43
Şekil 3.11 Düzenli dağılım paterni örneği.	43
Şekil 4.1. Mayın tarlası Modülü sınıf çizeneği.	49
Şekil 4.2. Uygulama modülü sınıf çizeneği.	50
Şekil 4.3. Temiz alanın taranmasına ait sonuç.....	57
Şekil 4.4. Şüpheli ve riskli hücreler içeren alt alanın taranmasına ait sonuç	57
Şekil 4.5. R_M 'nin önüne bir engel çıktığında R_G 'yi çağırmasına ait sonuç.	58
Şekil 4.6. Tam bir mayın taramasına ait sonuç.	59

KISALTMALAR DİZİNİ

AP	: Anti Personnel
AT	: Anti Tank
TNT	: Three Nitro Toluene
PDM	: Pursuit Deterrent Munition
PD	: Probability of Determination
FAR	: False Alarm Rate
UXO	: Unexploded Ordnance
CW	: Continuing Wave
GPR	: Ground Penetrating Radar
IR	: Infra Red
UV	: Ultraviolet
HSTAMIDS	: Handheld Standoff Mine Detection System
MURI	: Multidisciplinary University Search Initiative
DARPA UHF	
UWB SAR	: Defense Advanced Research Projects Agency ultra-high-frequency ultra- wide-band synthetic aperture radar
RFI	: Radio Frequency Interference
ODIS	: Ordnance Detection and Identification System
ILDC	: Improved Landmine Detector Concept
RRAS	: Radio Frequency Resonance Absorption Spectroscopy
NQR	: Nuclear Quadra pole Resonance
MMA	: Molecular Microwave Absorption
NMR	: Nuclear Magnetic Resonance
MR	: Magnetic Resonance
RF	: Radio Frequency
ILDC	: Improved Landmine Detection Concept
EMI	: Electro Magnetic Induction
NMA	: Nuclear Magnetic Absorption
EPR	: Electron Paramagnetic Resonance
TNA	: Thermal Neutron Activation

IMS	: Ion Mobility Spectrometry
MS	: Mass Spectrometry
GC	: Gas Chromatography
POMDP	: Partially Observable Markov Decision Process
MDP	: Markov Decision Process
SIHO	: Simplified Infinite Horizon Optimization.
R_M	: Metal detektörüne sahip robot
R_G	: GPR detektörüne sahip robot

1.GİRİŞ

Kara mayın harbinde, yer araçları, gemiler, veya hava araçlarını tahrip etmek ya da hasar vermek veya personeli yaralamak, öldürmek veya diğer şekillerde tesirsiz kılmak için tasarlanan, normal olarak koruyucu bir kaplama malzeme içinde bulunan, bir patlayıcı veya diğer tür bir malzeme. Kurbanının faaliyeti, zaman geçmesi veya kontrol araçları ile patlatılabilir.

Deniz mayın harbinde, gemilere hasar verme ya da batırma niyeti ile veya bir giriş bölgesine gemilerin yaklaşmasını önlemek amacıyla denize dökülen bir patlayıcı aygıt. Terim gemilerin veya liman tesislerinin altlarına, dalgıçlar tarafından iliştirilen aygıtları ve yerleştirilmesinden belirli bir süre sonra patlayacağı tahmin edilen aygıtları içermez.

Toprak üstüne yerleştirilen veya biraz gömülen, içi infilak maddesi veya kimyasal maddelerle dolu mayın. Kara mayını, genel olarak, üzerinden geçen araçların veya kıtaların ağırlığı ile infilak eder

Kara mayını, toprak altına veya üstüne yerleştirilen, doğrudan doğruya çarpma veya basınç etkisiyle patlayarak zarara yol açan patlayıcı silah. Düşmanı:

- İmha etmek,
- İstenen bölgeden uzak tutmak,
- Yön değiştirmek zorunda bırakmak,
- Geciktirmek,
- Psikolojik olarak rahatsız etmek amaçlarıyla kullanılır.

Kara, antipersonel ya da tanksavar mayınları, İkinci Dünya Savaşı'nda çokça kullanılan bu mayınların özelliği, karşı tarafın toprakları üzerinde ilerleyen düşmanın temasıyla patlayabilmesidir. Özellikle tanklar, savaş alanına girdiklerinde onları

durdurmaya ya da ortadan kaldırmaya yarayan mayını harekete geçirirler. Almanlar daha savaş başlamadan ve tüm savaş boyunca bu konuda üstün olduklarını gösterdiler. Daha 1935’de, yıldırım savaşında mayın kullanımı üzerine eksiksiz bir sistem geliştirmişlerdi. Metal mahfazalı bir tanksavar mayını olan Tellermine 35, bir tankın yatak takımını ağır bir biçimde tahrip etmeye ya da onu hareket edemez duruma getirmeye yetecek kadar, aşağı yukarı 5 kg patlayıcı içeriyordu; ateşleyicisi 150 kg’lık bir basınçla çalışıyordu; bu da yaya bir savaşçının ağırlığıyla patlamasını önlüyordu. Ama mayının, etkisiz bir duruma getirilmesini engelleyen bir tuzakla donatılması mümkündü. Ayrıca, 100 g ile 10 kg arasında değişen patlayıcılara takılabilen standart telli çeşitli ateşleyicileri olan sabit tipte antipersonel mayınları kullanıldı. S-35 sıçrar antipersonel mayınında bulunan ek bir patlayıcı, mayını telle önceden saptanan bir yüksekliğe sıçratıyordu; mayın bu yüksekliğe ulaşınca 100 m.’den daha uzağa her yönde öldürücü şarapneller fırlatarak patlıyordu [1].

Libya ve Mısır’da çöl savaşı sırasında, doğal engellerin bulunmadığı yerlerde engeller oluşturmak için milyonlarca mayın kullanıldı. Tunus, İtalya ve Rusya mayının en çok kullanıldığı bölgelerdi. Yavaş ve kullanışsız olan elektromanyetik detektörlerin sağladığı koruma, 1942’den itibaren Almanların bulduğu, detektörle saptanamayan mayınların (ağaç, bakalit, bitüm, cam, beton) yaygınlaşmasıyla iyiden iyiye gözden düştü. Almanlar beşli kümeler halinde mayın yerleştirme yöntemini kullandılar ve mayın tarlalarının yoğunluğunu istenen amaca uygulamayı her zaman başardılar. Ucuza mal oluşu, kullanım kolaylığı ve etkinliği bu mayının tüm savaşanlarca benimsenmesini sağladı. Mayının taşıdığı tehlikenin belirsizliği ve sürekliliği, savaşanları büyük ölçüde etkiledi ve alınması gereken ek önlemler zırhlı araçların ve piyadelerin hızını azalttı. Mekanik araçlar, patlayıcılar, hatta hayvanlarla yapılan mayın çalışmaları, sondayla karış karış incelemekten vazgeçmeyi gerektirecek kadar verimli olmadı [2].

Dünyanın önemli ordularında eksiksiz bir tanksavar ve antipersonel sistemi vardı. Bazıları, önceden hesaplanan bir gecikmeyle, hatta elektronik olarak uzaktan harekete geçirilen çeşitli ateşleyicilerden ve plastikten yararlanarak, mekanik olarak ya

da helikopterle yerleřtirme yntemleri ve etkisi ayarlanabilen imla hakları (ukur ya da dz imla hakları) geliřtirerek mayınların gizlenmesi temel ilkedir. Buna karřılık, mayınlar da belirli bir sre sonra etkisiz kılınmasına ya da kendi kendini imha etmesine de olanak veren aygıtlarla donatılmıřtır. Nihayet, elinde nkleer g bulunan lkelerce, doęal olarak kullanımı gizli tutulan, ama zorunlu geiř noktalarında kesin bir durdurma etkisine sahip, dřk gte nkleer haklı mayınlar zerinde arařtırmalar yapılmaktadır. Ayrıca sınırlı ve sreli kullanımlar iin retilen zayıflatılmıř uranyumlu (Depleted Uranium-DU) kara mayınlarıda retilmektedir [3].

AP ve AT mayınları Őekil ve byklk aısından farklılıklar gsterdięi gibi metal, plstik ya da tahta gibi eřitli kasa ve kılıf Őekillerinde de imal edilebilirler. Bu mayınları kendi sınıflarında incelememiz gerektięi taktirde Őu sınıflandırmayı yapmamız yerinde olacaktır.

Anti-Personel: Bu mayın eřitleri personel amalı dřk tahribatlı mayınlardır yani personele ynelik kullanım alanları pusu blgeleri nceden tespit edilmiř devriye geiř blgeleri ve daraltılmıř atıřma alanları olarak ayrılabilir ama personele verilebildięi kadar hasar verip ateř stnlęn ele geirmektedir.

Anti-Tank: Bu mayın eřitleri ise dięer mayınların aksine yksek tahribat gcne sahip olmakla beraber toplu ve yalnız seyir halindeki devriye aralarında kullanılır ama verilen en yksek hasar personel tařıyıcıları etkisiz kılmaktır fnyesiz olarak ta alıřabilme zellięine sahiplerdir birbirinden baęımsız iki ateřleyici tetik tertibatıyla da alıřabilirler [4].

2.KAYNAK ARAŐTIRMALARI

2.1. MAYIN ÇEŐİTLERİ:

Mayınlar önce kara ve deniz mayınları olarak ikiye ayrılırlar. Araőtirmalar, Dünya üzerinde çok sayıda bulunması, gündelik hayatta yaşam çevremizde (özellikle savaş ve terörizmin sık olduđu bölgelerde yaşayan insanlar için) bulunması dolayısıyla kara mayınlarına yöneliktir.

Kara mayınları iki ana grupta toplanır. İnsanlar için olan türü Anti Personel (AP) ve taşıtlar için olanı Anti Tank (AT) olarak sınıflandırılır.

2.1.1. Anti Personel Mayınlar:

Bu mayın çeőtileri personel amaçlı düşük tahribatlı mayınlardır yani personele yönelik kullanım alanları pusu bölgeleri önceden tespit edilmiş devriye geçiő bölgeleri ve daraltılmış çatıőma alanları olarak ayrılabilir amaç personele verilebildiđi kadar hasar verip ateő üstünlüğünü ele geçirmektir.

AP mayınları genellikle disk şeklinde ya da silindir biçiminde olup; çapları 20 ile 125 mm ve yükseklikleri 50 ile 100 mm arasında deđiőir. Ađırlıkları 30 g. civarında olan AP mayınlarında TNT, Tetryl ve Comp B gibi farklı patlayıcı malzemeler kullanılabilir [6].

AP mayınları gövdelerinde kullanılan malzemeye göre plastik, metal veya ahőap mayınlar, etkilerinin cinsine göre parça tesirli, patlama etkili olarak sınıflandırılabilir. Genelde plastik malzemedен yapılan mayınların metal detektörü ile bulunmaları düşük metal içeriklerinden ötürü zordur.

Tip 72 Plastik AP Mayını: Düşük metal içerikli olan bu mayının sadece ateşleme pimi metaldir. Mayın patlama etkili olup, 5-10 kg.lık bir basınçla aktif hale gelir ve imla hakkı 51 g. TNT'dir. 78 mm çap, 38 mm yüksekliğinde ve 140 g ağırlığındadır (Şekil 2.1).



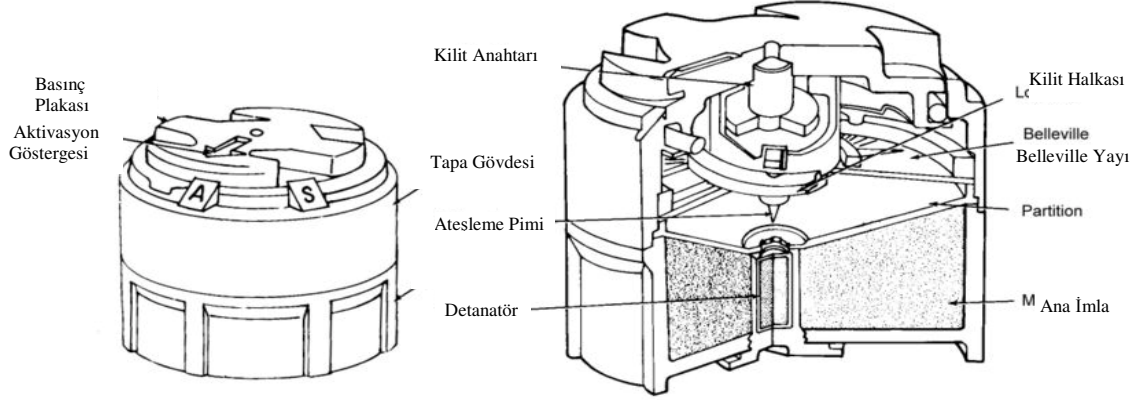
Şekil 2.1. Tip 72 Plastik AP mayını.

PMN AP Mayını: 100g TNT RDX karışımı 550g toplam ağırlık 53 mm yüksekli ve 120 mm çap [7].



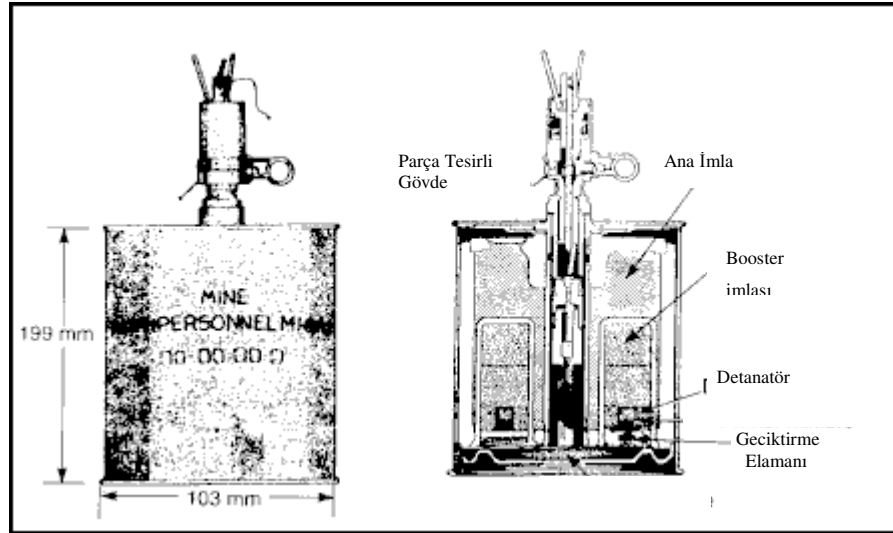
Şekil 2.2. Solda PMN AP mayınının sağda PMN2 AP mayınının metal içerikleri

M14 Plastik Mayını: Mayın patlama etkili olup, 9-16 kg.lık bir basınçla aktif hale gelir ve imla hakkı 29 g. Tetryl'dir. 56 mm çap, 40 mm yüksekliğinde ve 100 g ağırlığındadır. Şekil 2.3'te mayının şekli ve kesiti verilmiştir [8].



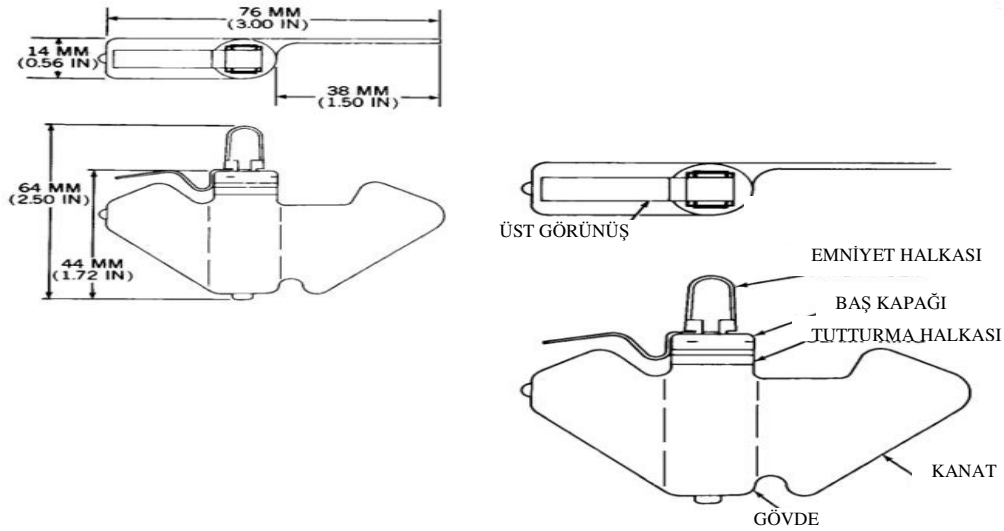
Şekil 2.3. M14 Plastik AP mayını ve kesiti.

M16 Mayını: Mayın metal bir gövdeye sahip olup parça tesirlidir. 1.4 - 4.5 kg çekme veya 3.6 - 20 kg ağırlıkla aktif hale gelir ve imla hakkı 575 g. TNT'dir. 103 mm çap, 203 mm yüksekliğinde ve 3.75 kg ağırlığındadır. Şekil 2.4'te mayının şekli ve kesiti verilmiştir.



Şekil 2.4. M-16 AP mayını ve kesiti.

BLU 43 Mayını: Bu mayınlar uçaklardan akrgo tipi mühimmatlarla atılır belirli bir yükseklikte patlatılan bombarın içindeki mayınlar mayınlanması istenilen bölgeye saçılır. Mayın toprağın üzerinde kalır fakat zamanla doğanın etkisi ile üzeri toprak veya birtki örtüsü ile kaplanabilir. Mayın patlama etkili olup, 9-16 kg.lık bir basınçla aktif hale gelir ve imla hakkı 9 g. Nitromethane/Nitroethane karışımıdır. 75 mm uzunluğunda, 45 mm genişliğinde (aktif hale geçince), 45 mm yüksekliğinde ve 20 g ağırlığındadır. Mayının normal şartlarda kendi kendini imha özelliği vardır. Şekil 2.5'te mayının şekli ve kesiti verilmiştir.



Şekil 2.5. BLU43B AP mayını

PMD 6 Mayını: Ahşap gövdeli olan bu mayın 190 mm uzunluğunda, 65 mm genişliğinde ve 65 mm yüksekliğindedir. Ağırlığı 400 g ve imla hakkı 200 g TNT'dir (Şekil 2.6.).

M86 PDM Mayını: Zayıflatılmış uranyumlu (Depleted Uranium-DU) mayın türüdür. Bu mayın topçu mühimmatları ile istenilen bölgeler atılarak bölge mayınlanmaktadır. Mayın kendi kendini imha etme özelliğine sahiptir. Mayının imla hakkı 0.101 g DU'dur (Şekil 2.7) [9].



Şekil 2.6. PMD 6 AP mayını



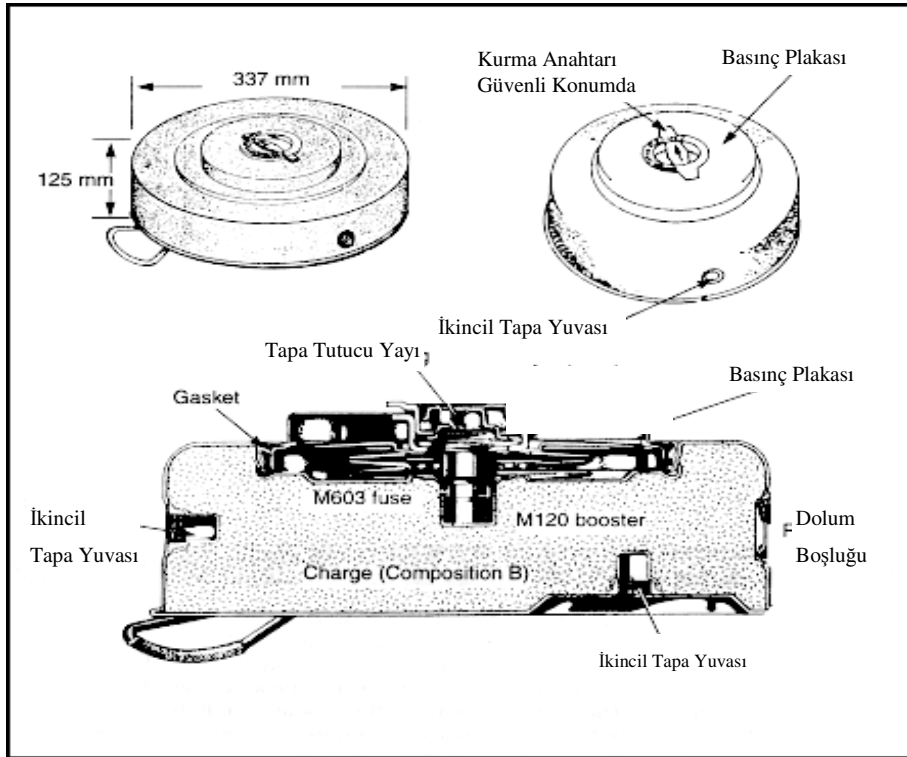
Şekil 2.7. M86 PDM mayını (Pursuit Deterrent Munition)

2.1.2. Anti-Tank Mayınları:

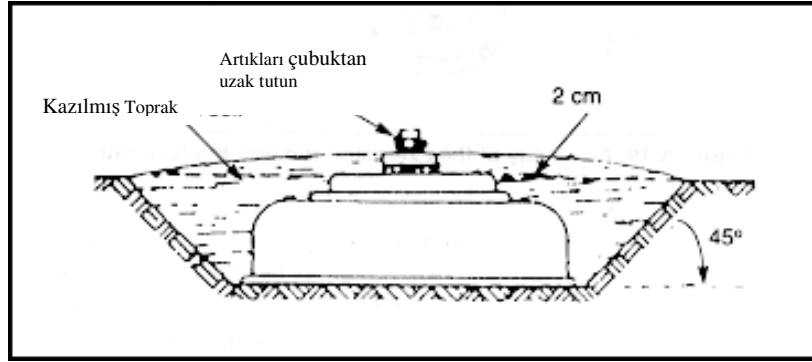
Bu mayın çeşitleri ise diğer mayınların aksine yüksek tahribat gücüne sahip olmakla beraber toplu ve yalnız seyir halindeki devriye araçlarında kullanılır amaç verilen en yüksek hasar personel taşıyıcıları etkisiz kılmaktır fünyesiz olarak ta çalışabilme özelliğine sahiplerdir birbirinden bağımsız iki ateşleyici tetik tertibatıyla da çalışabilirler.

AT mayınları genelde yassı silindirik şekilli ya da yuvarlak köşeli dörtgenler şeklinde üretilir. Çapları 150 ile 300 mm, kalınlıkları ise 50 ile 90 mm arasında değişir. Patlayıcı içerik ise genellikle TNT, Comp B ya da RDX gibi patlayıcılardan oluşur. AT mayınları genellikle savaş ekipmanı olarak değerlendirilir ve savaş alanlarında gizlenirler. Kullanım derinlikleri de değişken olup yüzeye aynı seviyede veya yüzeyin 150 mm altına kadar olan bölgeye yerleştirilirler. Çoğu AT mayını personel basmasına karşı duyarsızdır ve patlamaz. AT mayınlarını tetiklemede birim alana uygulanan kuvvet önemli rol oynar.

M15 Mayını: Metal gövdeye sahip bu mayın 337 mm çapında ve 125 mm yüksekliğindedir. İmla hakkı 10.35 kg Comp B'dir. Mayının kesiti ve toprağa yerleştirilmiş hali Şekil 2.8 ve 2.9'da verilmiştir.



Şekil 2.8. M15 AT Mayını

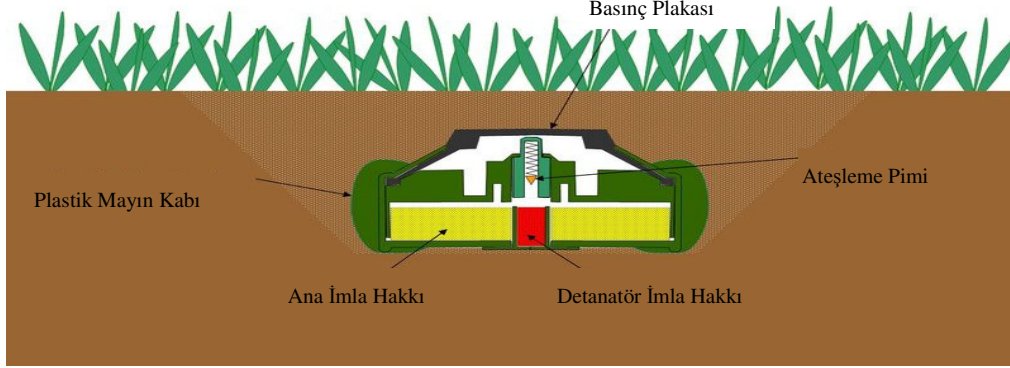


Şekil 2.9. M15 AT mayının toprağa yerleştirilmiş hali

M19 Mayını: Plastik gövdeye sahip kare şeklindeki bu mayın boyu 332 mm ve 94 mm yüksekliğindedir. İmla hakkı 9.45 kg Comp B'dir. Mayın son derece düşük metal içeriği (3 g) ile metal detektörlerince bulunması zordur (Şekil 2.9). Mayınları bir süre sonra toprağın ve bitki örtüsünün altında kalması ile bulunması daha da zorlaşır. Şekil 2.11'de toprağa yerleştirilmesinin üzerinden zaman geçmiş bir AT mayınının durumu verilmektedir [10].



Şekil 2.10. M19 AT mayını.



Şekil 2.11. Toprağa yerleştirilmesinin üzerinden zaman geçmiş bir AT mayını

2.2 MAYIN TESPİT YÖNTEMLERİ:

Bir kara mayını detektöründe esas gereksinim yüksek belirleme olasılığı (PD) olasılığının yanı sıra düşük yanlış alarm oranına (FAR) sahip olmasıdır. Her türlü hava ve toprak şartlarında ve gömülü mayınının büyüklüğün ile tipinden bağımsız olarak sonuç elde edilebiliyor olmalıdır.

Askeri operasyonlarda kullanılacak olan detektörlerde yoğun muharebe alanında hızlı çalışması zorunludur. Barış ortamındaki insani mayın temizleme çalışmalarında ise daha iyi PD ve FAR oranları beklenir.

Genel olarak kara mayınlarını belirleme teknolojileri 2 gruba ayrılabilir. 1) mayının kabını belirleme (yani mayının şeklinden bulma) 2) Doğrudan patlayıcının içeriğini tanımlama [11][12][13].

Mayın Belirleme Teknolojileri:

1- Manyeto Statik ve Elektro Statik

- 2- Doğru Akım (DC) İletim Metodu
- 3- Elektromanyetik İndüksiyon Spektroskopisi
- 4- Mikrodalga Tekniği
- 5- Optik Teknikler
- 6- Sismik Akustik Algılayıcılar
- 7- Ses ötesi
- 8- Radyo Frekans Rezonans Soğurma Spektroskopisi (RRAS)
- 9- Nükleer Işınım Metodu
- 10- Patlayıcı İzi Tespiti
- 11- Çok Algılayıcı ve Veri Birleştiricili sistemler.

2.2.1 Patlayıcıyı Tanımayan Belirleme Yöntemi:

Bu yöntemde patlayıcının niteliği belirlenmez şekilde tanımlanır. Burada sıklıkla boşluk ve anomali belirlemesiyle yapılır, yani topraktaki boşluklar ve toprağın özelliğindeki süreksizlikler taranır. Bu yöntemi kullanan tanımlama tekniklerinden bazıları şunlardır.

- 1- Manyeto Statik ve Elektro Statik
- 2- Doğru Akım (DC) İletim Metodu
- 3- Elektromanyetik İndüksiyon Spektroskopisi
- 4- Mikrodalga Tekniği
- 5- Optik Teknikler
- 6- Sismik Akustik Algılayıcılar
- 7- Ses ötesi

2.2.1.1 Manyeto statik ve elektro statik:

Manyeto statik teknik manyetik anomali tespiti olarak bilinir. Dünyanın sabit ve düzgün manyetik alanını ölçen manyeto metrelerdeki demir içerikli malzemelerden kaynaklanan bozulmalar vasıtasıyla tespit edilir. Bu yöntem sadece demir içerikli

mayınlar içindir. Toprağın manyetik özellikleri bu tekniğin kullanımına etki edecektir. Sadece demir içermeyen veya paslanmaz çelik parçalardan imal edilen mayınların keşfinde kullanılamazlar. Bununla birlikte UXO'ların (Unexploded Ordnance-Patlamamış Mühimmatlar) belirlenmesinde oldukça kullanışlıdır.

Elektrostatik yöntem hem metalik hem de metalik olmayan malzemelerin tanımlanmasında ortamdaki elektrostatik alandaki anomali ölçümleriyle kullanılabilir. Fakat yüzeye yakın yerlerdeki elektrostatik alanının çok düzensiz olması nedeniyle bu yöntem ancak çok sınırlı alanlarda kullanılabilir [5].

2.2.1.2 Doğru akım (DC) iletim metodu:

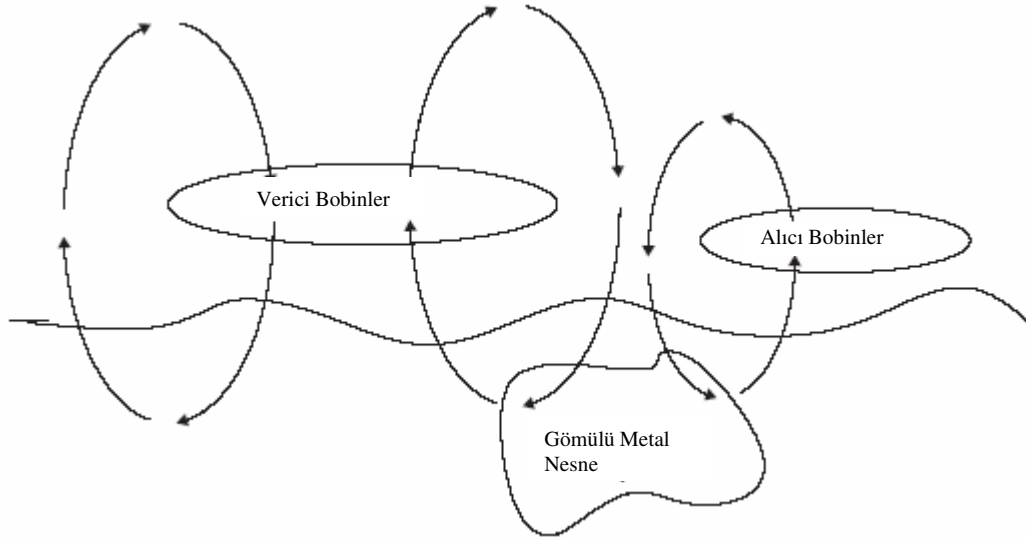
Bu metotta yüzeyin içine akım verip yüzey potansiyelinin ölçülmesiyle ortamda bulunan alanın etkin elektriksel empedansını ölçmeye dayanır. Bu akım verilen noktaların arttırılmasıyla ölçülen yüzey potansiyelleri ölçülmesiyle oluşan direnç metodunun genişlemesiyle Empedans Tomografi veya İletim Görüntüleme teknikleri oluşturulmuştur. Bu yöntemle elde edilen veriler değerlendirilerek empedans dağılımının görüntülenmesi sağlanır. Bu yöntemde de toprağın iletimselliği operasyon üzerinde etkilidir [14].

2.2.1.3 Elektromanyetik indüksiyon tekniği:

Metal detektörleri basit elektromanyetik indüksiyon tekniğine dayanarak oldukça geniş bir alanda mayın temizleme işlemlerinde kullanılır. Bu yöntemde dışarıdaki manyetik alana malzemelerden etki eden çevrinti akıntısının (eddy current) tespit edilmesiyle belirlenir. Toprağın manyetik hassasiyeti ve daha az olarak elektriksel iletimi metal detektörün çalışmasına etki eder [14]

Verici ve Alıcı bobinlerinden oluşan tipik bir metal detektörü Şekil 2.12'de verilmektedir. Verici halkalardan çıkan ilk manyetik alan metal ve onun bulunduğu

ortama nüfuz eder ve ilk manyetik alanda zaman değişiminin gömülü objede eddy akımına neden olacaktır ve bu akım alıcı tarafından ikinci bir manyetik alan gibi algılanacak. Metalik nesnenin alıcıda belli bir eşik değerinin üzerinde oluşturacağı gerilim ses tonu üretmesine neden olacaktır [16]. Modern metal detektörleri oldukça hassastır ve gömülü düşük metal içerikli mayınları (birkaç gram metal içeren) bile bulabilmektedir [16]. ODIS (Ordnance Detection and Identification System) Projesinde yapılan bir çalışmada 50 cm derinlikteki 1 cm³'ten küçük metal parçaları bile şeklini ve metalik hacmini manyetik merkezini ± 2 cm, derinliğini $\pm 10\%$ hassasiyette ve ayrıca ferromanyetik olup olmadığını belirlenebilmiştir [17].

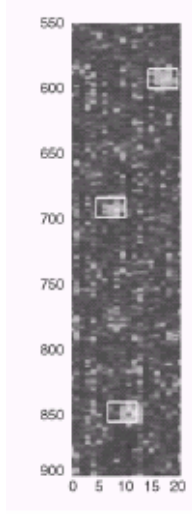


Şekil 2.12. Tipik Elektro Manyetik İndüksiyon Sistemi

2.2.1.4 Mikrodalga tekniği:

Aktif: Gömülü kara mayınlarının tespitinde, literatürde zemin keşif radarı GPR (Ground Penetrating Radar) veya yüzey altı radarı olarak geçen, aktif mikrodalga sistemleri geniş bir araştırma konusudur. Aktif sistemler önümüzdeki yıllarda kara mayınlarının tanımlanmasında çeşitli dizaynlara (impuls, sürekli dalga- CW), (frekans

modülasyonlu sürekli dalga (FM-CW), step frekans gibi) sahip olacaktır. Bu sistemlerin çoklu askeri sistem sensörlerin bir parçası olması sayesinde belli ortam koşullarında gömülü metalik olmayan AT mayınlarının bulunmasına olanak tanıyacak [18] [19].

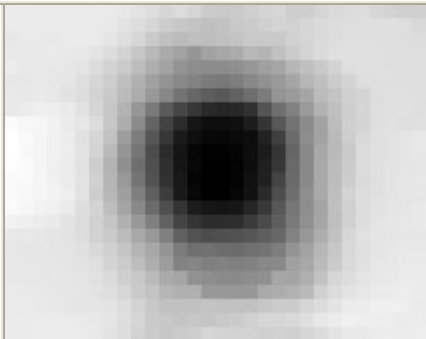
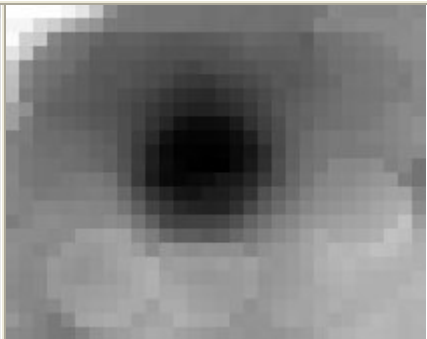
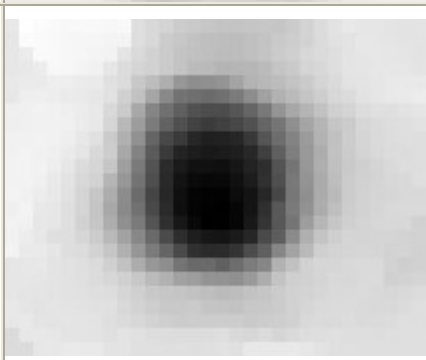
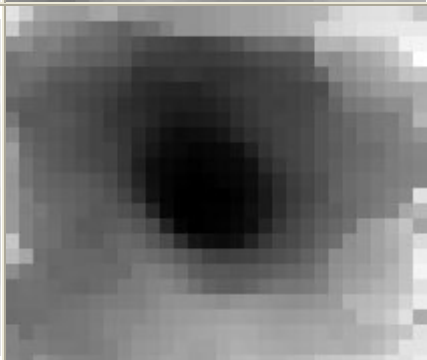
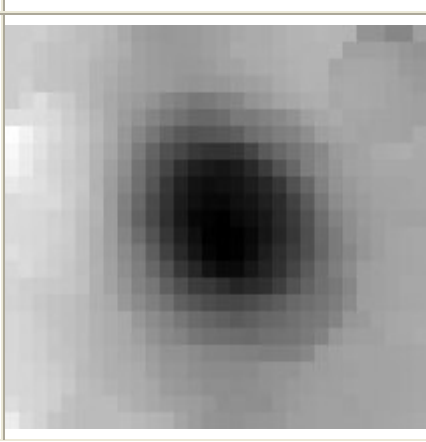
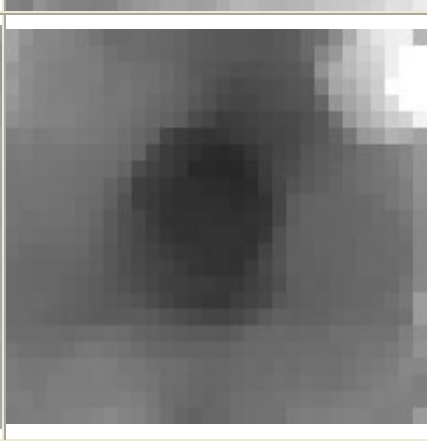


Şekil 2.13. GPR sistemi tarafından oluşturulan görüntü. En alttaki metalik mayın diğer ikisi metal olmayan mayın

Pasif: Pasif mikrodalga veya mikrodalga radyometrik tekniği, ilgilenilen cismin veya bölgenin mikrodalga bandındaki doğal siyah cisim ışımalarının ölçülmesi tekniğini içermektedir. Basitçe gömülü olan kara mayının radyometrik yöntemle belirlenmesi tekniği mayının gömülü bulunduğu ortamda oluşturduğu ısı salımındaki anomalinin belirlenmesine dayanır. Toprağın karışıklığı da sıcaklık veya ısı salımında değişikliğe sebep olabilir. Güneş ışınlarının yansımaları da yüzeydeki kara mayınlarının tespitinde kullanılır.

Her iki yöntemde de toprağın elektromanyetik özelliğiyle ilintilidir. Geçirgenlik ve iletkenlik toprağın rutubetli içeriği ile oldukça sıkı ilintilidir. [6].

Laboratuvar şartlarında 44 GHz ile metalik objeleri kuru kumda 7.5cm derinliğe kadar tespit yapabiliyor. Fakat çok küçük oranlardaki suyun varlığı bile girebilirliğini çok azaltıyor [20].

Manolya yaprakları altındaki mayın		
Kuru toprak altındaki Mayın (1cm'den 3 cm'ye kadar) $m_v < 0.05$		
Toprak altındaki Mayın (1cm'den 3 cm'ye kadar) $m_v = 0.15$		

Şekil 2.14. 5 GHz kullanılarak elde edilen görüntü. Sol sütundaki M20 metalik AT mayını sağ sütundaki PMN2 plastik AP mayını (m_v = su hacmi oranı)

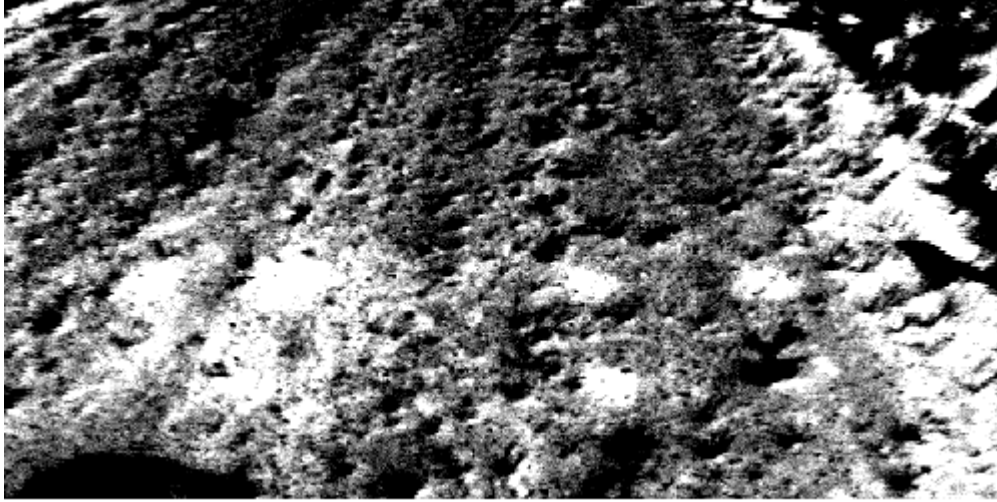
2.2.1.5 Optik teknikler:

Işık geçirmeyen saydam malzemelerde elektromanyetik yayılımın optik dalga boyunun nüfuz edebilirliği 1mm. den az olduğundan hareketle, kara mayının gömülü olmasının toprağın yüzeyine etkisini ölçmeye dayanır.

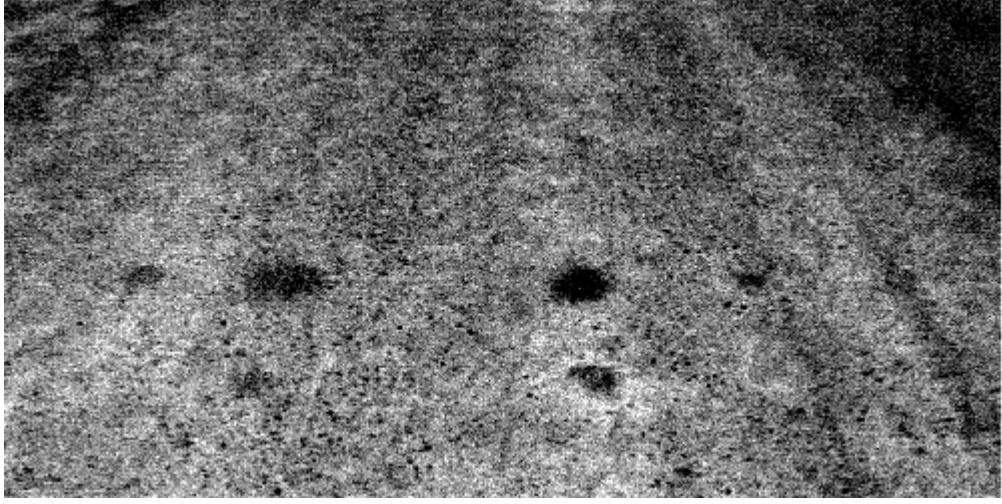
Kızılötesi (IR): Termal kızılötesi veya kızılötesi radyometresi kara mayınının üstündeki toprak ile bozulmamış toprağın spektrumu veya kızılötesi ışınımını yayması veya yansıtması arasındaki değişimi ölçmesi esasına dayanır [18].

Isı akışının yayılımdaki değişiklik kara mayının varlığından, toprağın bozulmuş olmasından ve yüzey ısı yayımının toprak bozulmuş olduğu için değişmesinden kaynaklanır (kızılötesi görüntülerdeki sıcaklık çözünürlüğü 0.1°C 'dir). Toprağın termal direnci, termal geçirgenliği ve ısı kapasitesi içindeki nem miktarına bağlıdır ve bunlar kızılötesi sistemin çalışmasıyla ilintilidir.

Genellikle yola döşenen AT mayınlarının belirlenmesinde kullanılır. Yüzeyin 3m yukarısında aşağı doğru 40 derecede konumlandırılan IR detektörden Şekil 2.15'de gündüz ve Şekil 2.16'de gece alınan görüntüler verilmektedir. Maksimumum 10-15 cm.ye gömülmüş mayınlar tespit edilebilmektedir [11].



Şekil 2.15. Gündüz (14:15) IR görüntüsü



Şekil 2.16. Gece (04:45) IR görüntüsü

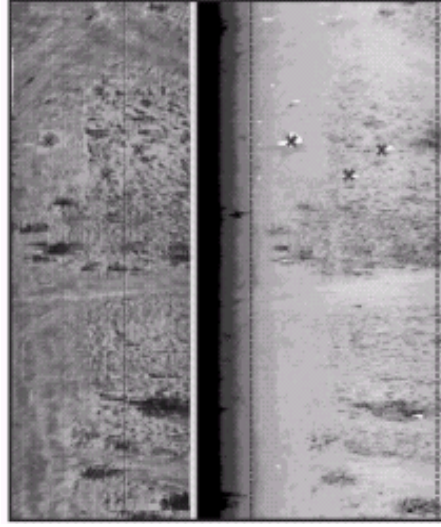
Sağ alttaki 3 karanlık bölge yeni gömülmüş mayınları belirtiyor. Büyük olan AT mayını diğer 2 küçük ise AP mayını temsil ediyor. Sol alttaki 3 karanlık nokta ise gömülmesinin üzerinden daha uzun süre geçmiş yine 1 AT ve 2 AP mayını temsil ediyor [12].

Görünür Dalga Boyları: Görünür dalga boyları yüzeyin optik özelliklerini, polarizasyon ve yansıtma karakteristikleri, bozulmuş ve bozulmamış topraklarda fark etmesinin tespitinde kullanılır. Gömülü mayının üzerinde bulunan bitki örtüsünün büyümesi bunun ölçümünü zorlaştırır [18].

Mor ötesi (UV): Yer yüzeyindeki mor ötesi yansımaların güneşle ilgili mi, yoksa yapay bir kaynaktan mı geldiğinin ölçümüne dayanır. Bu özellik gömülü kara mayınının üzerindeki toprakla bozulmamış toprak arasında farklılık göstermesine dayanır. İlerleyen zamanda bu özelliğin etkileri azalır.

Bu teknolojiye gelişmeler daha ziyade, mayınlı alanların havadan (zeplin, helikopter, balon vb.) taranması üzerine yoğunlaşmıştır.

Genelde 8–12 μ m. IR dalga boylarına duyarlı kameralarla elde edilen görüntülerin; satır, sütun ve renk bilgisinin bölünmesinden sonra elde edilen motifler yapay sinir ağından geçirilerek yorumlanır. Böylelikle başarı %90'lara varabilmektedir [13].



Şekil 2.17. Yandaki şeklin solundaki IR sağındaki görünür dalga boylarının görüntüsüdür. X işaretli yerler mayınları ifade ediyor.

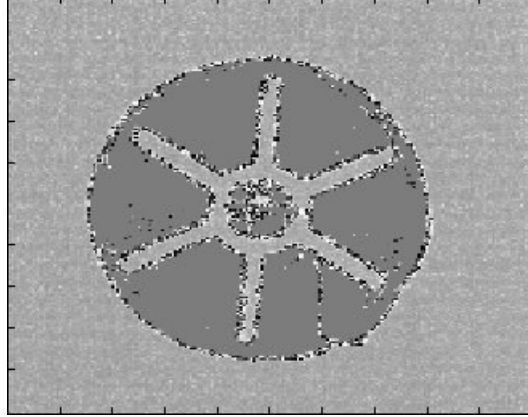
Kızılötesi görüntüleme yöntemiyle aylar önce gömülen mayınlar dahi tespit edilebilir. Kızılötesi kameradan alınan görüntünün değerlendirilme hızı; mayın tipine, toprak tipine (yoğun, kum, mil, çamur, taşlı vs) ve günün saatlerine göre değişebilir. Ancak testlerde 3m genişliğinde tesviye edilmiş bir kumlu yolda ilerlerken 10 m'deki yanlış alarm oranının %10–80 arasında olduğu kaydedilmiştir [11].

Kızılötesi sistemin başarısı çalışılan ortam şartları ve yüzey sıcaklık değişimi ile yakından ilgilidir. Yapılan denemelerde, çevre şartlarının değişimine bağlı olarak mayınların doğru tespit performansında ciddi değişimler gözlenmiştir. Donanım sistemlerinin yeterliliğine rağmen, sinyal işleme algoritmalarının yetersizliği görülmüştür ki; buradan ısıl izlerin henüz yeterince anlaşılmadığı ve tatmin edici bir modelin henüz geliştirilmediği sonucunu çıkarabiliriz. Bu teknikte yararlanılan kızılötesi elektromanyetik dalga frekans aralığı itibariyle toprak yüzeyini aşamaz ve ayrıca

mayının çevresinde hava koşullarına göre hızlı termal değişimler görülebilir. Bundan ötürü bu teknoloji ile gömülü mayınların tespiti ancak anlık ve özel koşullarda mümkün olabilmektedir.

2.2.1.6 Akustik Sismik:

Aktif akustik metodu, zemine enjekte edilen akustik enerjinin kara mayının ve veya bozulmuş toprakla bozulmamış toprak arasında ki akustik empedans farklılığından kaynaklanan yansımalarının ölçülmesine dayanır. Yer titreşimlerinin gömülü kara mayınlarının üzerindeki etkilerini belirlemeye yönelik çalışmalar devam etmektedir. Akustik empedans kütle yoğunluğuna bağlıdır. Bu özellikte toprağın nem miktarı ile ilgilidir [15].

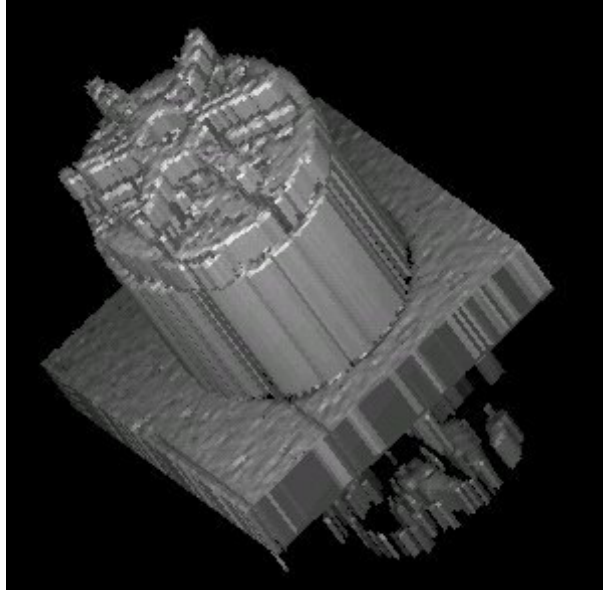


Şekil 2.18. Su içindeki AP mayının 2D görüntüsü. Bu şekil 15 MHz prop ve 0.6mm. (X,Y) tarama basamağı ile 3 cm derinlikte su içindeki mayının görüntüsü [15].

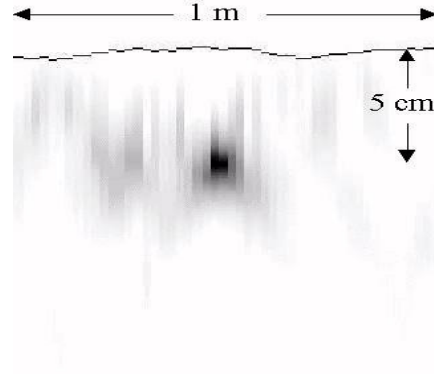
2.2.1.7 Ses ötesi tespit

Bilinen ses ötesi tespit (Ultrasound Detection) 20 KHz (insan duyum eşliğinin üzeri) ve üzeri frekanslardaki ses dalgalarının ortama salınması ve yol boyunca farklı akustik özellikli girerken yansımaya uğrayan dalgaların yakalanarak değerlendirilmesine

dayanır. Ses, çok nemli ve killi-çamurlu zeminlere verimli bir şekilde nüfuz edebilir. Ancak doğası gereği yol boyunca karşılaşılabileceği hava boşluklarının ötesini görüntüleyemez. Araştırma çalışmaları, AP mayınlarının tespit edilmesini ve diğer nesnelere ayırt edilmelerini sağlayacak "sinyal işleme" ve "model tanıma" metotları üzerinde yoğunlaşmıştır. Özel araştırma konusu da toprak ile mayın arasındaki akustik darbelerin farklarının değerlendirilmesine yöneliktir. Bu çalışmalarda; uygulanan 1 milisaniye'lik ses ötesi darbelerin yolu boyunca uğrayacağı yansıma işaretleri arasındaki farklar ölçülmüş ve başarılı sonuçlar elde edilmiştir. Ancak bu yöntemde de küçük nesnelere yansıyan sinyallerin toprak hataları ve düzensizlikleri kaynaklı sinyallerden ayırt edilmesi ile ilgili problemler mevcuttur. Sorunun üstesinden gelebilmek ve geri planda istenmeyen bu sinyallerden kurtulmak için "geri plan çıkartma" yöntemleri geliştirilmiştir.



Şekil 2.19. Su içindeki AP mayının 3D görüntüsü Düşey tarama da eklenerek elde edilen görüntü [21].



Şekil 2.20. Bahçe toprağında 5cm derinliğe gömülmüş 12cm.lik bir plastik mayının 1msec akustik pulse ile yüzey taramasından elde edilen görüntü.

2.2.2 Patlayıcı tespitiyle belirleme tekniği:

Patlayıcı tespiti metodu kitle tespiti ya da iz tespiti metodu olarak sınıflandırılır. Kitle tespiti metotları RF rezonans soğurma (mikro moleküler soğurma hariç) ve nükleer ışıdır. Patlayıcılar(%10–40) toprağa(<%0.1) göre oldukça yüksek miktarda nitrojen içerirler. Ayrıca patlayıcıların 5 ile 7 arasındaki etkin atomik numaraları, organik malzemelerin ki gibi, 11–12 olan toprakla karşılaştırılabilir. Bununla birlikte toprağın öz kütlesi ($1.0-2.5 \text{ g/cm}^3$) ve patlayıcıların ki ($1.6-1.8 \text{ g/cm}^3$) benzerdir. İz tespiti yöntemi özel bir patlayıcının moleküllerinin hava veya toprak çevresindeki örneklerde belirlenmesine dayanır [18].

- 1- Radyo Frekans Rezonans Soğurma Spektroskopisi (RRAS)
- 2- Çekirdek Dörtkutup Rezonansı
- 3- Nükleer Işınım Metodu
- 4- Patlayıcı İzi Tespiti

2.2.2.1 Radyo frekansı rezonans soğurma spektroskopisi (RRAS):

Bu metot iç veya dış alanların ve atomların elektron veya çekirdeklerinin elektrik veya manyetik momentleri arasındaki etkileşimin rezonansına bağlı oluşan elektromanyetik alandan soğurulan enerjinin ayrılmasına dayanmaktadır. Bunun için dört basit metot vardır:

Nükleer manyetik rezonans (NMR), nükleer kuadrapol rezonans (NQR), Elektron paramanyetik rezonans (EPR) ve mikro dalga moleküler soğurulma (MMA).

2.2.2.2 Çekirdek dörktutup rezonansı

Çekirdek Dörktutup Rezonansı (Nuclear Quadrupole Resonance, NQR), birçok patlayıcı ve uyuşturucu maddenin yapısında bulunan azot izotopunun (^{14}N) tespit edilmesine dayalı, Manyetik Rezonans (MR) tekniğine benzeyen özgün bir radyo frekansı (RF) tekniğidir. Son zamanlarda bu tekniğin mayın tespitinde kullanılabilmesine yönelik çalışmalar artmıştır.

Diğer tekniklerde rastlanan yanıltıcı ve ayırt edilmesi zor sinyaller yerine NQR, çok daha karakteristik sinyal algılamaya imkân vermesi bakımından üstün durumdadır. Ayrıca havaalanları ve halka açık alanlarda patlayıcı ve uyuşturucu madde tespitinde kullanılabilme potansiyeli bakımından da büyük ilgi çekmektedir.

Sinyal/gürültü oranının çok düşük olması nedeniyle girişim problemleri yaşanmasına rağmen, son yayınlanan araştırma raporlarına göre yanlış alarm durumunda bile kabul edilebilirliği çok yüksek özgün NQR işareti elde edilebilmektedir. Gözlenen NQR frekansları; çekirdeğin elektrik dörktutup momenti ile dışarıdan çekirdek çevresinde oluşturulan elektrik alanı değişimleri arasındaki etkileşimle elde edilmektedir.

TNT, RDX, ve PETN gibi azot içeren patlayıcıların ürettiği, NQR işareti patlayıcının tespit ve tanımlanmasının yanında patlayıcının miktar ve derinliğinin tahmin

edilmesini de sağlar. Elde edilen yüksek özellikli işaret nedeniyle topraktaki azot işaret gübre vb. başka malzemelerin bozucu etkisinden de sakınılır.

Metal detektörlerde olduğu gibi toprağa yakın seviyede kullanılmak üzere aranmakta olan patlayıcının NQR frekansına yakın bir frekansta RF darbeleri ile beslenen düzlemsel bir RF antenden oluşan özel tasarlanmış sistemlerdir. Aynı şekilde ikinci bir anten kullanılarak uyarımı yapılan patlayıcıdan salınan zayıf sinyaller algılanır. Doğru frekansta düşük yoğunluklu RF sinyali patlayıcıya uygulandığındaki ki bu genellikle 0,5 ile 6 MHz arasındadır, azot çekirdeğinin enerji seviyesi değiştirilir. RF uyarımı kaldırıldığında, çekirdekler başlangıç enerji seviyelerine geri dönerler ve bu sırada bir enerji salınır. Bu enerji nedeni ile bir de karakteristik radyo sinyali açığa çıkar. Elde edilecek sinyaller; RF darbelerin hemen ardından oluşacak serbest indüksiyon zayıflama işaretleri ve darbe şeridi aralarında gözlenen yankılar şeklindedir. Fakat sonraki darbelerin sayısı çok olduğu için ve ortalamaları kısa sürede alınabildiği için hassaslık artırılabilir. Algılayıcı bobinlerde elde edilecek bu zayıf indüksiyon akımlarının şiddeti patlayıcının miktarını, frekansı ise cinsini ortaya koyar.

Sadece patlayıcı tespiti amaçlı arama yapılıyor olması nedeniyle metal detektörlere göre yanlış alarm oranı düşüktür. Ancak TNT gibi yaygın kullanılan patlayıcılardan alınacak NQR sinyalleri çok zayıftır ve yüksek girişim eğilimi gösterirler [22].

2.2.2.3 Nükleer ışınım metodu:

Bu metot toprağın ve patlayıcıların parametre fonksiyonunun, nitrojenin karakteristik ışınımının geri dönüşümünün veya karakteristik olmayan ışınım saçılmasının yoğunluğunun değişiminin tespitiyle ilgilidir. Bu yöntemde aslında ortamdaki anomalinin tespitine yöneliktir. Bu yöntemde örnek X ışınları geri saçılması ve nötron moderation (hidrojen anomali). Termal nötron hareketliliği (TNA) N14 ün karakteristik gama ışınımının tespiti temeline dayanır.

Isıl Nötron Aktivasyonu (Thermal Neutron Activation – TNA): Birçok patlayıcının yapısında oldukça fazla bulunan azot çekirdeğinin, toprak yüzeyinde elektron hızlandırıcı (karşı atomdan elektron koparacak kadar enerjiye sahip iyonize elektronlar üretilir) kullanılarak elektron bombardımanına tutulması neticesinde aktive edilmesiyle salacağı özel gama ışınlarının algılanmasına dayalı bir yöntemdir. Geri saçılan gama ışınlarının algılanması çok hızlı olduğu için mayın tespiti de çok hızlıdır

Patlayıcıların (%18 ile %38) toprağa göre (%0 ile %0,1) daha fazla azot içermesi nedeniyle azot varlığının tespitinde hata yapılma olasılığı yoktur. AT mayınların, AP mayınlardan daha fazla patlayıcı içermelerinden dolayı, bu yöntemle tespitinde daha iyi netice alınmaktadır. Bu sistemin zayıf yönleri ise çok karmaşık yapıda olması ve toprak nüfuz derinliğinin düşüklüğüdür (10 - 20 cm civarı) [22].

Detektörün yaklaşık ağırlığı 180 kg. genelde AT mayınları için kullanımı daha uygun AP mayınlarındaki patlayıcı miktarı az olduğu için tespiti daha zor olmaktadır. Oldukça kompleks bir sisteme sahip olan cihaz ayrıca sınırlı bir derinliğe kadar tarama yapabiliyor (10-20cm). Şekil 2.21’de bir TNA detektörü görülmektedir.

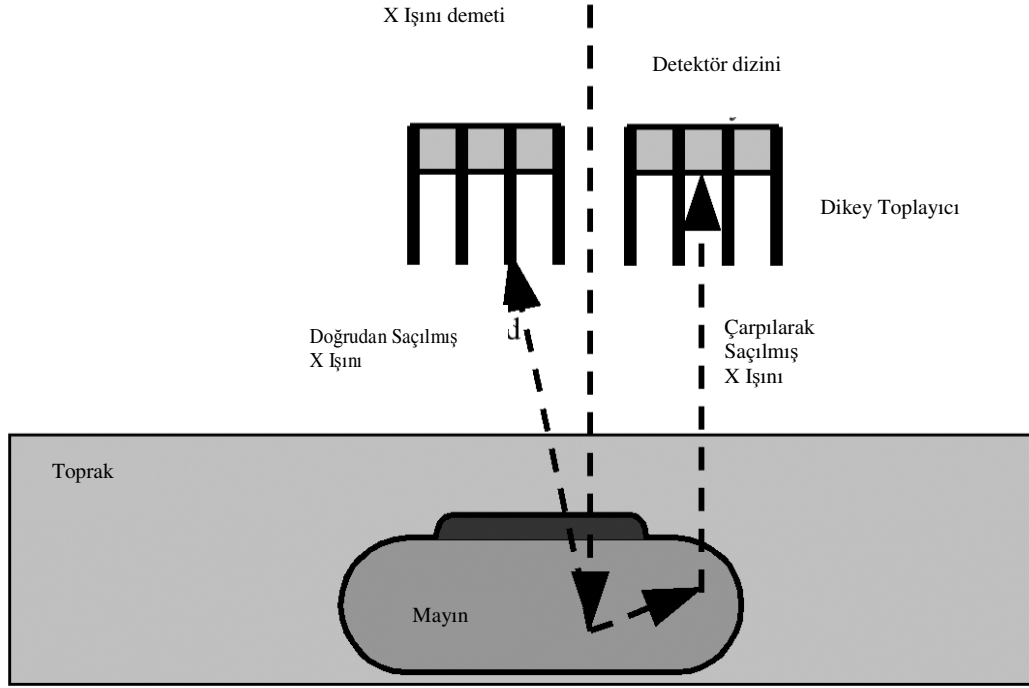


Şekil 2.21. TNA Detektörü

X ışını Gerisaçılımı (X-ray Backscatter): Gerçek zamanda AT tespitinde kullanılan bir tekniktir. Burada metal detektörlerine benzer, ancak temel olarak farklı bir yaklaşım kullanılır. Bu teknikte 2-3 cm çözünürlüklü ve iki boyutlu görüntü elde edebilmek mümkündür. Buna karşın tespit hassaslığı, düşük nüfuz derinliği ve algılama başlığının yüksek tutulması durumlarına bağlı olarak çeşitli problemler yaşanabilir

Geleneksel X ışını radyografisinde (röntgen) görüntü; hızlandırılmış fotonların doku içinden geçirilmesi ve izdüşümde yoğunluğa bağlı olarak kütlelenin arka tarafındaki ışığa duyarlı filmde oluşturulan gölgelerle elde edilir. X ışınının dalga boyu bir mayının boyutlarına göre çok küçük olduğundan çok kaliteli mayın görüntüsü elde etmek mümkündür. Ancak burada mayının arka tarafında oluşacak görüntüyü fiziksel olarak yakalamak mümkün olmadığı için, mayından geriye saçılan ışıklardan görüntü elde edilir. Mayın ve toprak arasındaki yoğunluk farkı ve iki kata varan atom numarası farkları görüntü elde etmek için yeterli ayırt edici bir özelliktir.

Gömülü mayın görüntüleme için iki temel yaklaşım kullanılmaktadır. İlki, hizalandırıcıdan geçirilmiş ve böylece odaklanmış X ışınlarının mayına çarpmasıyla saçılan fotonların başka bir hizalandırıcıdan geçerek detektöre varmasıyla görüntünün oluşturulmasına dayanmaktadır. Hizalandırma gereksinimi sistemin ağırlığını ve hacmini arttırmakla birlikte görüntü için gerekli foton sayısını azaltmaktadır. Kaynak olarak yüksek güçlü bir X ışını üreticisine gereksinim vardır. Yüksek hacim, ağırlık ve güç gereksinimi tek kişinin kullanabileceği taşınabilir bir detektörün üretimini olanaksız kılar. İkinci olarak ise hizalandırıcı kullanmadan geniş bir alanın X ışınına tabi tutulması ve yansıyan ışınların uzaysal filtreden geçirilerek benzetim yöntemleri ile hizalandırıcıdan geçmiş hale getirilmesini içerir. Bu yaklaşımda tek kişinin kullanabileceği taşınabilir bir sistem üretmek mümkündür [23]. Şekil 2.22'de geri saçılım detektörü ile patlayıcıyı atom numarasından tespit eden detektörün şematik diyagramı görülmektedir.



Şekil 2.22. Geri saçılım detektörü şematik diyagramı

2.2.2.4 Patlayıcı izi tespiti (TED):

Bu metot gömülen mayından çıkan parçanın ve tanımlamada kullanılabilen ayrılan atom veya iyon molekül bileşenlerinin belirlenmesi esasına dayanır. Bu metotta patlayıcı için yeterli hassasiyete sahip olması ve ayrılan parçanın yeterli miktarda olması gereklidir.

Biyokimyasal olmayan: Bu yöntem patlayıcı buharını tespiti dayanır. Kütle spektrometresi (MS), iyon hareketliliği spektrometresi (IMS) gaz kromatografi (GC), katı hal detektörleri, diferansiyel raman spektroskopisi bu yöntemlerden bazılarıdır.

Biyokimyasal Tespit: Bugün için patlayıcı tespitinde kullanılan en iyi detektörlerden biri de köpektir. Özel eğitim verilmiş köpeklerin patlayıcılardan havaya karışmış ve mayının karakteristik özelliklerini taşıyan gaz moleküllerini (10^{-12} ile 10^{-13} g

patlayıcı) algılayarak, gömülme tarihlerinden ve içerik oranlarında bağımsız olarak tespit edebildikleri bilinmektedir. Bu da köpekleri, metal içeriği çok düşük olduğu için metal detektörleri tarafından algılanamayan mayınların tespitinde vazgeçilmez kılmaktadır. Ek olarak köpekler; herhangi bir metal ya da patlayıcı olmayan nesneye tepki veremedikleri için detektör ile aramalarda yaşanan kusurlu arama ve zaman kaybı gibi olumsuz etkiler de ortadan kalkar.

Her türlü arazi koşulunda çalışabilen köpeklerle yapılan aramalarda sonuçlar çok güvenilirdir. Ayrıca nakledilebilirlikleri ve arazi taramalarının elle tespitten beş kat daha hızlı yapılabilmesi, mayın tespitinde köpeklerin diğer avantajları olarak sayılabilir. Güney Afrika ve Afganistan'daki köpekli aramalarla ilgili raporlarda, köpeklerin ne kadar başarılı oldukları görülebilir.

Yukarıda belirtilen avantajların yanı sıra köpeklerin dezavantajları da yok değildir. Kısıtlı periyotlar halinde günde ancak birkaç saat görev yapabilen köpekler, aynı anda birden çok noktadan patlayıcı kokusu geliyorsa şaşırabilirler. Ayrıca eğitimlerin periyodik olarak sürdürülmesi gerekmektedir. Çünkü bir köpek, zamanla patlayıcıları bulmanın dışında başka bir şey için ödüllendirildiğini zannedebilir ve asıl konuya ilgisini kaybedebilir. Yüksek miktarda patlayıcı tespitinden sonra da problem yaşanma riski vardır. Köpek, daha az miktarda patlayıcıya karşı tepki göstermeyebilir. Böyle durumların yaşanmaması için de yine periyodik eğitim öne çıkar.

Köpeklerin mayın tespitindeki başarısı havadaki patlayıcı molekül yoğunluğu, çevre ve hava koşulları ile doğrudan ilgili olsa da unutulmamalıdır ki köpeklerin etkinliği tamamı ile aldıkları eğitim, personel becerisi ve doğru yönlendirilmelerine bağlıdır.

Bu yöntem biyolojik sistemlerle patlayıcı izinin tespitine yönelik kimyasal bir süreçtir. Bu yöntemde hayvanlar (köpek, fare, arı vb.) ve özel bakteriler kullanılmaktadır.

Köpeklerin burnu patlayıcının tespiti konusunda oldukça hassastır. Gramın 10^{-12} – 10^{-13} ,te biri mertebesindeki patlayıcıyı tespit edebilir [8].

3.MATERYAL VE METOT

3.1 ÇOKLU DETEKTÖR KULLANIMI:

Sorunların aşılması için en yoğun yapılan çalışmalar birden fazla sensörün bir arada kullanılmasına yönelik yapılan çalışmalardır. Bunlar dizi sensör şeklinde veya farklı sensörleri bir arada kullanarak yapılabilmektedir. Sensör seçimlerinde hatalı alarm verme nedenleri, çalışabildikleri ortam ve tanımlama kabiliyetleri göz önüne alınarak yapılmaktadır. Bunlara ait değerlendirme Tablo 3.1’de verilmektedir.

Tablo 3.1 Seçilen mayın arama yöntemindeki hatalı alarm kaynakları

Detektör Teknolojisi	Öncelikli FAR Kaynağı
EMI	Metal atıklar, toprağın doğal iletimliliği ve manyetik değişimi
GPR	Doğal karışıklık (kökler, kayalar, su paketleri vs.
Akustik/Sismik	Boşluklar, insan yapımı objeler
Floresans Polimerler	Patlayıcı artıkları
NQR	Radyo frekans kesişimleri
IR	İşaretlerin zamanla ve çevresel durumla değişmesi, yüzey karışıklığı

Çoklu sensör kullanımı mayınların tespitindeki olasılığı arttırarak çeşitli ortamlardaki kullanım etkinliğini arttıracaktır. Tablo 3.2’de ortam, mayın tipleri açısından arama teknolojilerinin uygunluğu değerlendirilmiştir [8].

Burada karşımıza çıkan temel problem ise elde edilen verilen değerlendirilmesidir. Farklı sensörlerden gelen verilerin aktarımında ve ne şekilde değerlendirileceğinde yani verilerin karar seviyesindeki kaynaştırılmasında (decision-level fusion) sorunlar çıkabilmektedir. Bu durumda operatörün seçimi veya kullanılan programın etkinliği büyük önem taşımaktadır.

Bu konuda günümüzde karar seviyesi kaynaştırması yöntemi (decision-level fusion) bağımsız sensörlere göre yüksek tanımlama olasılığında düşük hatalı alarm verme oranı sağladığı için tercih edilmektedir. Bilgi ya da nitelik seviyesi

kaynaştırmanın çok zor, ve az olgunlaşmış olması nedeniyle karar seviyesi birleşmeleri daha çok tercih edilmektedir.

Tablo 3.2 Mayın tipine ve toprağın nem miktarına göre en uygun mayın detektörü seçimi.

Detektör Teknolojisi	Mayın Tiplerine ve Toprak Durumuna Göre Hangi Teknolojinin En Etkin Olduğu					
	Metal Mayınlar	Az Metal İçeren Mayınlar	TNT Mayınları	RDX Mayınları	Kuru Toprak	Islak Toprak
EMI	+		+	+	+	+
GPR	+	+	+	+	+	
Akustik/Sismik	+	+	+	+	+	+
Floresans Polimerler	+	+	+	+		+
NQR		+		+	+	+

Bununla birlikte 2001’de Gunatilak ve Baertlein’in EMI, GPR ve IR detektörlerinden topladığı bilgilerle karara karşı nitelik seviyesindeki kaynaşma hakkında yaptığı gösteri sırasında katı karar seviyesi kaynaşması detektörlerin bağımsız performansından daha iyi bir performans gösterememiştir. Bunla birlikte yumuşak karar seviyesi kaynaşması ve nitelik seviyesi kaynaşması daha faydalı olmuş, nitelik seviyesi algoritması gösterişsiz ama anlamlı bir katkı sağlamıştır.

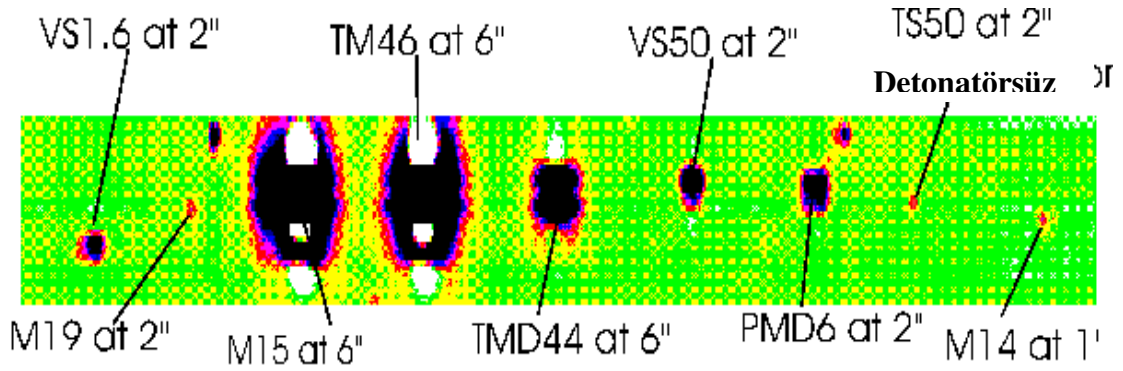
Bunun yanında dizi antenlerde sensörlerin aynı malzemeye farklı tepkiler vermesi karışıklığı yol açabilmektedir. Bu nedenle dizi antenlerde sensörlerin farklı hedeflere yönelmesi ile nitelik-seviye kaynaşması (Feature-level fusion) daha etkin bir yöntem olabilir. Örneğin Kanada’da başlatılan Geliştirilmiş Kara Mayının Detektörü Kavramı (ILDC) için 3 laboratuarda çoklu sensör sistemi ortak çalışması ile yürütülmektedir. Bu çalışmada barış ortamında yoldaki mayınların temizlenmesinde uzaktan komutalı bir araca sistem monte edilmiştir. Ön görüş için IR detektör, aşağı yönlendirilmiş 3m genişliğinde bir oldukça hassas EMI detektörü, 3m genişliğinde aşağı yönlendirilmiş GPR ve TNA detektörü olmak üzere 4 farklı tipte detektör kullanılmıştır. Bunlardan sadece TNA patlayıcıyı belirleyebilme özelliğine sahiptir ve bulunan objenin

patlayıcı içeriği konusunda bilgi sağlayarak doğrulama yapmaktadır. Bu sistem metalik olmayan orta ölçekli mayınların bulunmasında kullanılmaktadır. Sistemleri AP mayınlarından korumak için çeşitli tedbirler alınmıştır. (Şekil 3.1)



Şekil 3.1. Mayın Tarama Aracı alan çalışması sırasında

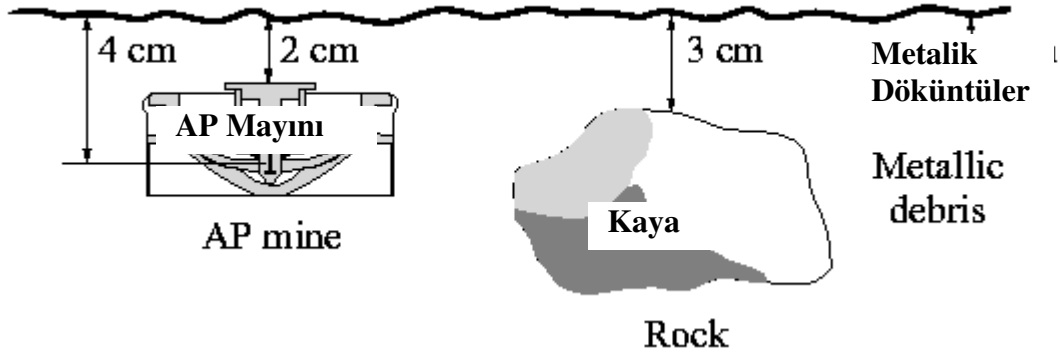
Sıralı detektör çalışmalarında ilk akla gelebilecek olan metal detektörleridir. 2m sıralı metal detektörden alınmış bir görüntü Şekil 3.2’de verilmektedir. Bu detektör 1cm³ ten küçük parçaları bile bulabilir ve çalışabilme derinliği ise yaklaşık 50 cm.’ye kadardır. Bu sıralı metal detektörü kullanılan programa bağlı olarak parçanın merkezini ± 2 cm ve derinliğini $\pm 10\%$ kesinlikle belirleyebilmektedir. Manyetik sesi ise parçanın büyüklüğü ile orantılıdır. Burada büyük olarak görülenler metalik mayınlar küçük olarak görülenler ise metalik olmayan AP mayınlardır.



Şekil 3.2. Sıralı metal detektörden alınan görüntüler.

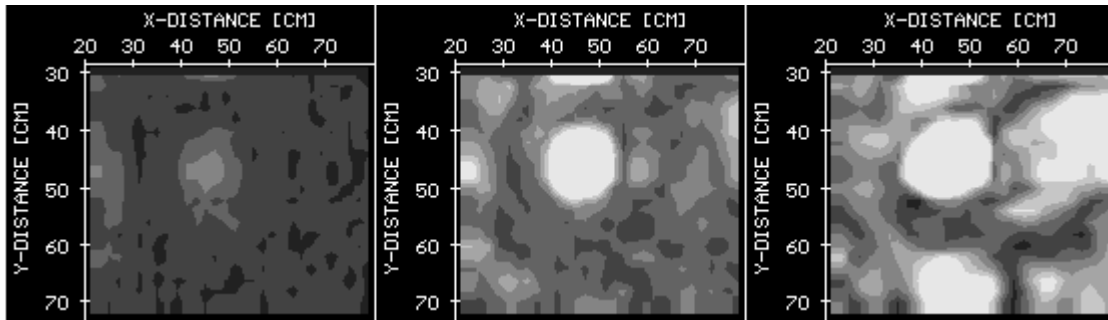
En çok kullanılan ortak kullanılan detektör çifti Metal detektörü ve GPR detektörüdür. Bunun temel nedenlerinden birisi GPR detektörünün metal detektörlerinin ortamdaki metal döküntülerine şeklin büyüklüğünü kontrol ederek yanlış alarm vermesini engellemesi, metal detektörünün ise GPR detektörünün toprak altındaki homojen olmayan taş gibi maddelerin metal miktarlarını tespit ederek GPR detektörünün yanlış alarm vermesini engellemesi.

Bu durum Şekil 3.3'deki toprak altına yerleştirilmiş taş, mayın ve metal parçasına verdikleri tepkileri incelenerek daha rahat değerlendirilebilir.

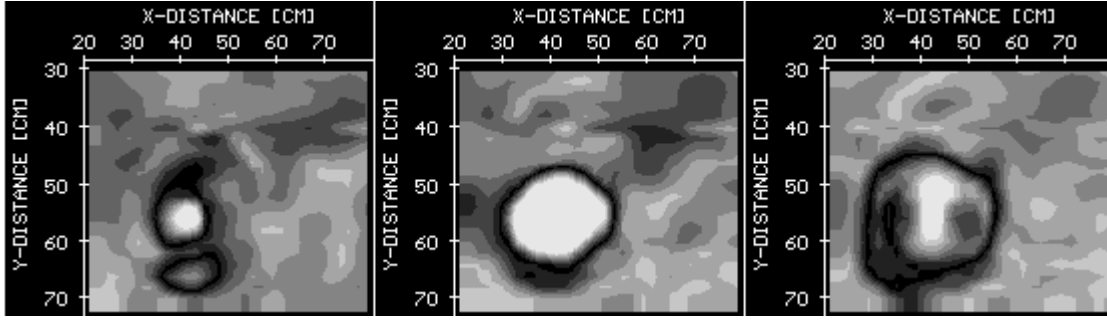


Şekil 3.3. Kum içine yerleştirilmiş 3 madde.

Bu maddeleri GPR detektörü incelenmesi sonucunda gömülü AP mayından alınan görüntüler Şekil 3.4'te ve gömülü olan taştan alınan görüntüler Şekil 3.5'te verilmiştir. Metal parçasının görüntüsü ise ihmal edilebilecek kadar küçüktür.



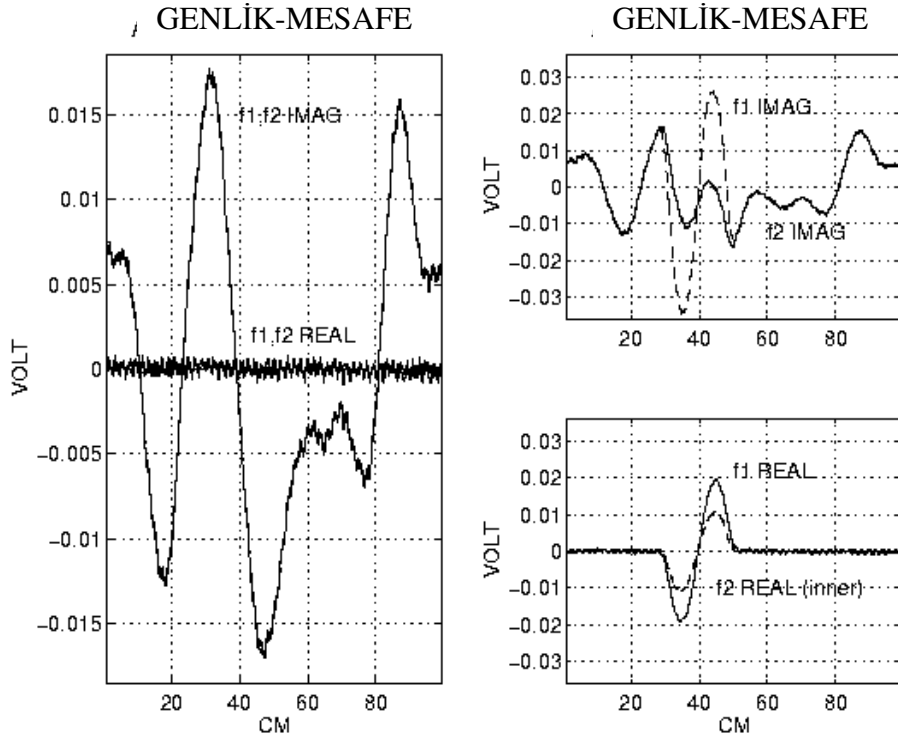
Şekil 3.4. 1/1.5/2 ns'deki kum içindeki AP mayının yatay örneklemedeki görüntüleri.



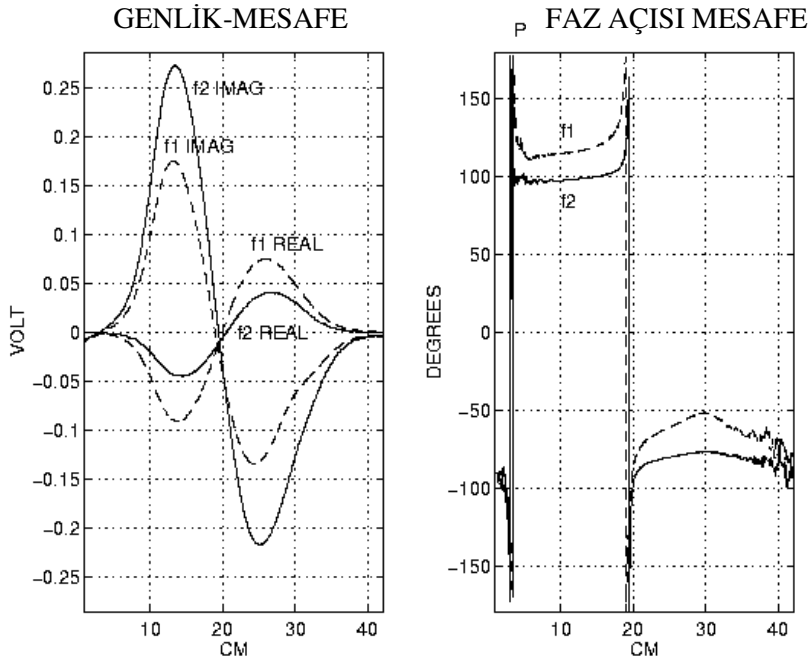
Şekil 3.5. 3/3.5/4 ns'deki kum içindeki taşın yatay örneklemedeki görüntüleri.

Aynı maddelere metal detektörünün verdiği tepkileri inceleyecek olursak, metal detektörü ferromanyetik nesnelere için 2.4 KHz ve paslanmaz çelik ve alaşımlar için 19.2KHz olmak üzere 2 adet sinyal kullanmaktadır. Alıcıda f1 ve f2'nin gerçek ve sanal görüntüleri oluşmaktadır. Şekil 3.6'da detektörün taşa karşı verdiği tepkiyi görmekteyiz, burada büyük ihtimalle toprağın homojen olmamasına bağlı olan sinyalin sanal kısmındaki büyük dalgalanmaya dikkat ediniz. Diğer taraftan frekansların üst üste binme durumu çok net bir şekilde görülüyor. Yani giden sinyalin herhangi bir malzemeye etki etmediğini gösteriyor. Bu şekilde metal detektörü GPR detektörünün taşa vereceği yanlış alarm ihtimalin ortadan kaldırıyor. Metal detektörün minimum metal içeriğine sahip bir plastik mayına verdiği yanıt aynı şeklin sağında verilmektedir. Yayınlanan sinyalin gerçek bileşeni son derece net ve kesin olarak karakterize edilmektedir. Hedefin üzerindeki bölgede üst üste binmede gözle görülür bir aykırılık fark edilmektedir. Gömülü olan bu cisme verilen tepki ile GPR detektörünün verdiği sonuç birleştirilerek mayın için detektörlerin nasıl doğru tespitte bulunabileceğinin varsayımı yapılabilir.

Aynı şekilde gömülü olan 3.5g'lık metal parçasına metal detektörün verdiği yanıt Şekil 3.7'de görülmektedir. Bununla birlikte GPR detektörünün bu metal parçasına verdiği yanıt ihmal edilebilecek kadar küçüktür. Bu şekilde sonuçların birleştirilmesi ile bu cismin ret edilmesi ile yanlış alarm verme ihtimali kaldırılmış olmaktadır. Bu da bize metal ve GPR detektörlerinin birlikte kullanımının getirdiği faydaları göstermek bakımından önemlidir [23].

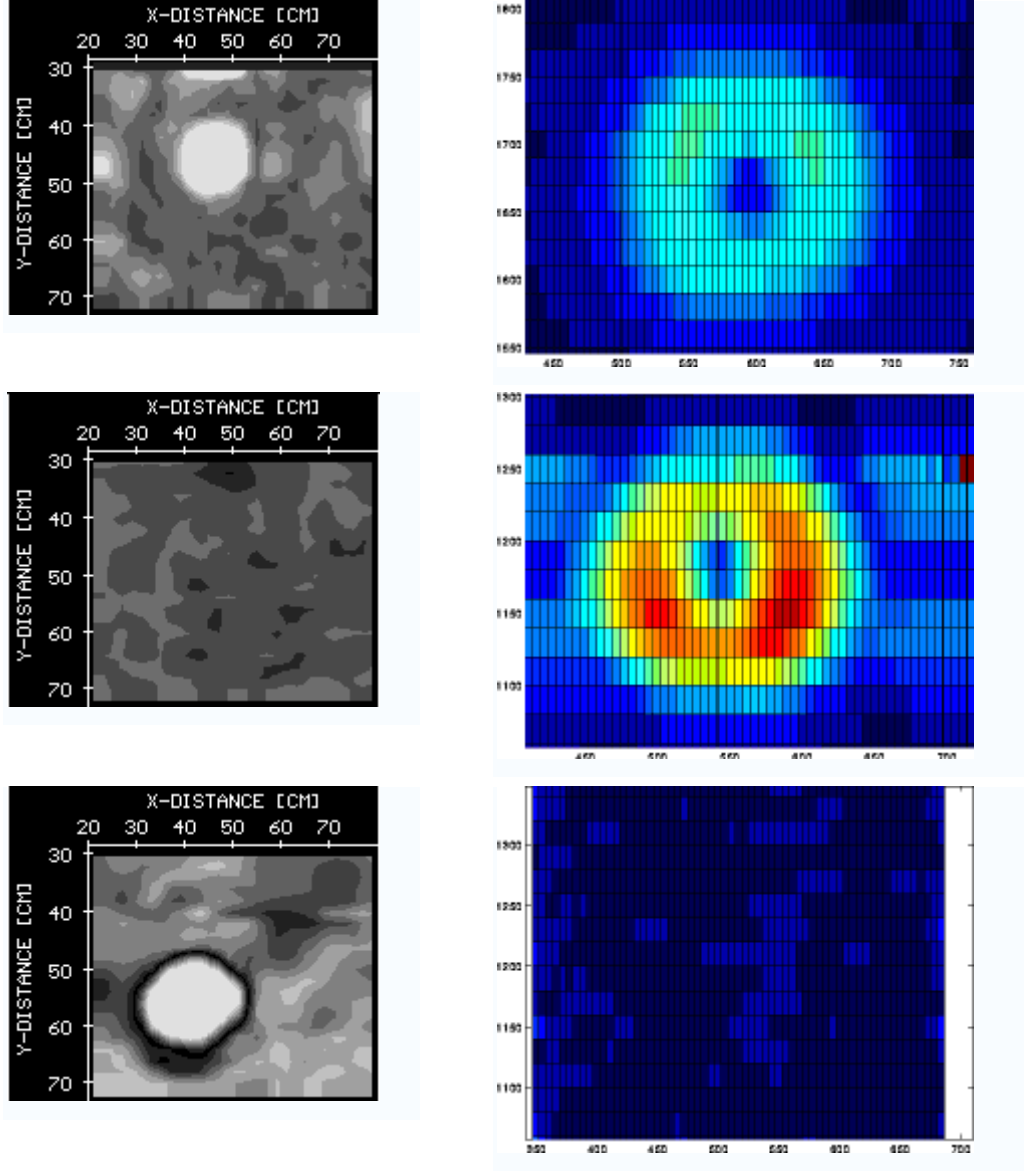


Şekil 3.6. Metal detektörün kuma gömülü olan taşa verdiği yanıt solda, AP mayınına (merkez: 40cm 'de) verdiği yanıt sağda verilmektedir.



Şekil 3.7. Metal detektörünün 3.5g'lık metal parçasına verdiği yanıt.

Bu iki detektörün aynı malzemeler için verdiği görüntülerin karşılaştırılması Şekil 3.8'dedir [24]



Şekil 3.8. 3 objenin Metal Detektörüne (soldaki) ve GPR detektörüne (sağdaki) ait görüntüleri.

3.2 ROBOTLARIN KULLANIMI

Bütün bu tekniklerin yanı sıra mayın aramada önemli bir çalışma alanı da bu sistemlerin robotlar üzerine monte edilmesiyle daha hızlı ve güvenli arama yapmaktır. Yeraltı mayınlarını ve mayın tarlalarını tarama problemi; hem sivillerin, hem de askerlerin ilgi alanına girmektedir. Mayın taraması için robotlar gibi özerk unsurları kullanmak tercih edilir çünkü tarama sırasındaki tehlikeleri ve maliyeti düşürür. Robotik araştırmalarda, tek bir robot veya birçok robottan oluşan bir grup mayın tarayıcı ekipmanlarla donatılır ve mayın olması muhtemel yerlerde görevlendirilir.

Mayın taraması işlemi insan tarayıcılar için birçok risk içermektedir çünkü küçük bir ihmâl bile ölüme sebep olabilmektedir. Elle tarama ise hem zor, hem de tekrar gerektirmektedir. Bu hareketlerin analizi içlerinden bazılarının robotik sistemlerle daha kolay ve güvenli gerçekleştirilebileceğini göstermiştir. Bu bakış açısına göre, bu robotik sistemler dünya üzerindeki milyonlarca kara mayınını bulma ve kaldırma konusunda önemli bir rol oynamaktadırlar.

Bu bağlamda, robot gruplarının tekli robot arama operasyonlarına nazaran bazı belirgin avantajları mevcuttur. Koordineli çalışan robotlar geniş bir alanın temizlenmesinde tek bir robota göre çok daha verimli ve etkili olabilir [25]. Ucuz robotların çoklu kullanımı mayın patlamalarının zararını göreceli olarak azaltılacaktır [26]. Bundan başka, robotlardan oluşan bir takım, başarısızlık noktalarını giderme sürecinden geçerek tekli veya merkezci sistemlerin kuvvetlenmesini sağlar. Açıkça görüldüğü gibi, problemin çözümü zorlu arazilerde akıcı ve etkili mayın taraması yapabilecek özerk ve de güvenilir makineler kurmaktan geçmektedir.

Amaç, robot grupları için alandaki bütün kara mayınlarını araştırarak ve algılayacak koordinasyon stratejisini geliştirmek ve haritayı minimum zamanda güncellemektir. Mayınlı alan hakkındaki bilgileri kullanarak algılama işlemini ilerletmek için araştırma teknikleri geliştirilmiştir. Robotik çalışmalarda bilgi paylaşımı da

performansı artırmak için önemlidir. Bu yüzden, robotların alanın hangi noktasında görev yapmış olduğunun belirlenmiş olması zorunludur. Bundan başka, robotlar hareket noktalarını belirleyecek global bir harita oluşturmak durumundadır. Bizim yaklaşımımız alanı ortaya koymak ve alanı bir düzlem olarak görmek gerektiğini işaret eder. Bu durum, bir robotun ayak izi ve detektörün hassaslığı üzerine inşa edilmiştir. Bizim yaptığımız tek varsayım çalışma süreci boyunca robotların pozisyonlarını bilmesi ve lokal koordinatlara göre hareket etmesidir.

Tekil çalışan robotlarla ilgili birçok detaylı çalışma yapılmış olmasına rağmen; robot grupları ile ilgili yalnızca birkaç yaklaşım bulunmaktadır. Balch ve Arkin, haberleşme etkisinin çok amaçlı robot gruplarının performansı üzerindeki etkisini araştırmıştır [27]. Çalışmalar açık haberleşmenin bulunmadığı anlarda bile (kapsama), robotların arazide birlikte çalışabildiğini işaret etmektedir. Robotlar, geçişlerinin fiziksel kanıtlarını geçtikleri çevreye bırakırlar. Bu, biyolojik sistemden etkilenilerek (örneğin karıncalardan) gerçekleştirilir. Bu iletişim türleri fiziksel iletim sistemi gerektirmedikleri için, kapalı iletişim olarak adlandırılırlar. Bir çok durumda, çevreyi fiziksel olarak işaretlemek imkansızdır veya istenmez. Ayrıca, fiziksel delil araştırması keşif verimliliğini azaltır.

Latimer, en az tekrarlar, tüm alanı kapsama ile ilgili robotlar arasında yüksek bir koordinasyon seviyesi gerektiren bir yaklaşım öngörmektedir. Robotlar beraber başlar ve ayrışmanın ilk hücrelerini kapsayan bir düzende, tüm boşluğu kat ederler. Eğer takım içindeki bir robot kritik bir noktaya denk gelirse, takım ayrı hücreleri kapsamak için bölünür ve ilerideki bir başka noktada yeniden birleşir. İşleyiş prensipleri yalnızca yarım dağılımlıdır ve ancak takımdaki üyelerden biri üstüne düşen görevi tamamlayamazsa başarısızlığa uğramış olur. Kara mayını aramalarında zaman kaybını engellemek için en uygun arama yöntemi olmakla birlikte dezavantajı bütün alanın kapsama içine alınmamasıdır [28].

Rekleitis, alanı çizgilerle ayırarak robot takımı tarafından sıra ile araştırılmasını temel alan, işbirliğine dayalı bir yöntem sunmuştur. Bir robot hareket ettiğinde, diğer robotlar sabit kalarak beklerler ve hareket eden robotu izlerler ve pozisyonunun ne olacağını değerlendirirler. Bazı robotlar sabit olduğundan dolayı, kaynakların bir kısmı sürekli boşa beklemektedir. Bu durum haritanın doğruluğunu etraflıca geliştirme avantajı getirdiği halde, arama işlemini hızlandırmaz [29].

Simmons, sınır aramasını temel alan bir çoklu robot yaklaşımı önermektedir. Bu yöntemin en önemli sakıncası sistemin merkezi haber almayla işliyor olması ve buna bağlı olarak da merkezi haber alma çöktüğünde tüm sistemin çökecek olmasıdır. Tek bir noktanın başarısızlığı sonucu çökebilecek olan bir sistem sağlam bir sistem değildir [30].

Yamauchi'nin yaklaşımı ise, robotun mevcut haritadaki en yakın sınıra hareket etmesi olarak öne çıkmıştır. Bu yaklaşım tamamen dağıtımlıdır ve tek robotun başarısızlığa uğramasını da tolere edicidir. Bununla beraber, bireysel robotların farklı sınırları seçmesi için koordinasyon bileşenleri yoktur ve buna bağlı olarak da kullanılabilir olan bütün robotları değerlendirme avantajından yoksun kalmış olur [31].

Kara mayını araması problemi için oldukça az sayıda çoklu robot koordinasyonu yaklaşımları mevcuttur. Büyük bir çoğunluk tam kapsam yöntemleri teklif etmiş ve diğerleri de probalistik tarama önermiştir. Ancak, onlar arazideki mayınları bulmak için mayınları belirleyecek örnek parametreleri kullanmışlardır. Bu daha kabul gören bir yöntemdir ancak, çevresel etki nedeniyle mayın bulma oranı azalmaktadır.

Robotlar arasındaki koordinasyonun esas amacı; takım üyelerinin arasında görev dağılımı yaparak takımın başarıya ulaşmasının sağlanmasıdır. Böylece, takım üyeleri arasında görev tanımı ve görev yapılandırması çoklu robot koordinasyonunun ana etkenleri durumuna gelmiş olur. Kara mayını tespit etme operasyonunda, görev olarak tanımlanan alan bir robot tarafından incelenecektir. Görev, kara mayınları bulma şansını

artıracak bir yöntemle belirlenir. Mayın tarlası hakkındaki bilgiler ve takım üyeleri arasındaki haberleşme takımın mayın tespit etme operasyonundaki başarısını geliştirir.

Temel olarak, robot takımı merkezi planlama veya dağıtık planlama ile koordine edilebilir. Merkezi planlamada her robot durumunu takımdan gelen bilgilere göre değerlendirir; dağıtık planlamada ise her robot kendi planından sorumludur [32][42]. Bu tür merkezi yaklaşımların esas avantajı optimal planlamadır. Bunun yanı sıra, dinamik durumlara yavaş karşılık verme, haberleşme güçlükleri ve tek nokta başarısızlığı gibi bazı dezavantajları da beraberinde getirmiş olur. Dağılmış yaklaşım durumu tek nokta başarısızlığına karşı sağlamdır ancak, planlama en üst düzeyde uygun olmalıdır çünkü tüm planlar yerel bilgiye dayalıdır.

Her iki yaklaşımın en iyi özelliklerini birleştirmek için, sürekli bir yöneticinin veya takımı kontrol eden merkezi bir plancının bulunmadığı dağıtılmış yaklaşımı kullanılmaktadır. Diğerlerinden daha fazla bilgisi olan robot yöneticiliğe gelir, bir taslak oluşturur ve diğer bireysel robotları görevlendirir. Robotlar yeni yönetici tarafından görevlendirildiklerinde yeni yerlerinde göreve başlarlar. Yeni görev, mevcut göreve veya duruma dayalı olarak belirlenmiş olacaktır.

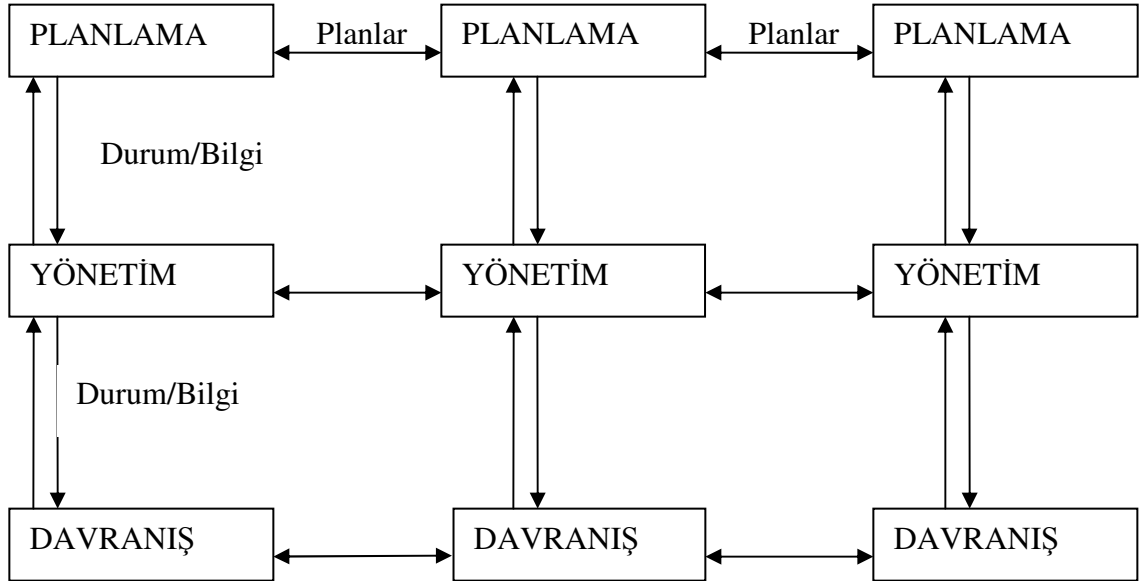
Robotlar, kendilerine verilen görev esnasında engellerden aşma, düz bir çizgide ilerleme ve mayın taraması gibi sahip oldukları yeteneklerle ellerinden geleni ortaya koyma kapasitesine sahiptir. Verilen görevin yerine getirilmesi yerel bilgiye bağlı olduğu için, iletişimdeki kopukluk görevin yerine getirilmesini etkilemeyecektir.

Çoklu robot arama planlamasında kabiliyet dağıtılmış, sıralı yapının bir parçası olarak yer alır. Bu yapı Şekil 3.9'daki gibi geleneksel üç sıralı robot yapısının genişletilmiş halidir. Bu, robotların aynı katmandaki robotlarla doğrudan birbirlerini etkilemelerine imkan verir. Davranış katmanı davranışları kontrol eder, idari katman görevi koordine eder ve tabakayı incelemek görevleri oluşturur ve belirler. Görevi en iyi şekilde yerine getirmek için araştırma teknikleri kullanılır.

3.2.1 Robotla Arama Yönteminde Kullanılan Metotlar:

3.2.1.1. Yol planlamaları:

Geleneksel yol planlaması robotun boyunu ve şeklini dikkate alarak başlangıç pozisyonundan hedefe kadar çarpışmadan gidebilmesi için gerekli unsurları araştırmak için geliştirilmiştir [33][34]. Ancak bu durum kara mayınları için yerleştirme uygulamasına yardımcı olmaz. Diğer taraftan ise kapsama yol planlaması robotun sensörünün süpürdüğü alanı belirler. Kapsamlı yol planlama problemine yaklaşım belirgin hücelere bölme metodunu da işler hale getirir [35]. Hücre, hedef alanın belirli küçük parçalara ayrılmasıdır. Tüm kapsam ise her hücrenin kapsanması ile gerçekleştirilir.



Şekil 3.9. Katmanlı Mobil Robot mimarisi

3.2.1.2 Robotik kara mayını araştırması için probalistik yöntemler:

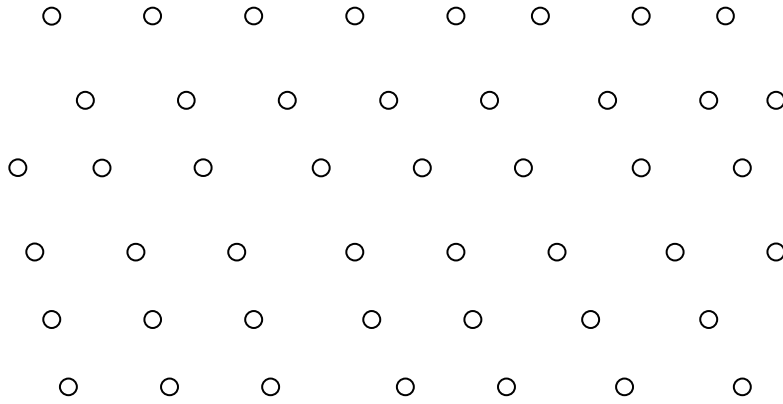
Tarama boyunca, bütün mayınların tespit edildiğinden emin olmak için, robot bölgedeki bütün alanları bir tarayıcı sensörle dolaşır. Bunu gerçekleştirmek için, robotun hedef bölge içerisinde dikkatli planlanmış bir yol takip etmesi gerekir. Geleneksel çalışan saha planlayıcıları tarama konusunda yetersiz kalmaktadır çünkü sadece iki nokta arasındaki güzergahı dikkate alırlar ve geçilen alanlara dikkat etmezler. Kapsama yol planlaması, adından da anlaşılacağı gibi, özellikle robotun detektörünün taradığı bölge ile ilgilendirir. Robotun ayak izi ile taranan alanın birleştirilmesi sonucu hedef bölgenin incelenmesi sağlanmış olur. Ayrıntılı kapsama algoritması, robotun mayın tarlasındaki bütün noktaları en azından bir kere dolaştığı kapsamlı bir yol planlamasıdır [35]. Ayrıntılı kapsam, robotun sınırsız bir zamana ve çok iyi bir mayın detektörüne sahip olması durumunda en iyi uygulanacak stratejidir.

Bununla birlikte, zamanın veya gücün sınırlı olması birçok durumda hedef alanın tamamen taranmasının önünde engel olarak bulunmaktadır. Probalistik planlama teknolojisi, mevcut sensörlerin bu tarz bir tarama operasyonunda etkin olarak yer almasını sağlar. Eğer planlayıcı mayınların yerleşiminin probalistik haritasına sahip ise, robotu rahatlıkla yönlendirebilir. Örneğin, planlayıcı mayınları toplamak yerine öncelikle bütün bölgeyi süpürmesini sağlayabilir. Bir süre sonra bölgede hiç mayın kalmadığı belirlendikten sonra; planlayıcı bölgenin mayından arındırılmış olduğunu kaydeder ve robotu başka bir alandaki görevle yönlendirir [28].

Araştırmalar mayınların yerini belirlemede uygun araştırma stratejileri problemine işaret eder. Bu durumu iki aşamada izah edebilir. Birinci aşama, hedef alandaki olası mayın yerleşimlerini belirlemek. İkinci aşama, bu haritayı kullanarak robotun mayınları belirleyeceği uygun bir güzergah belirlemek.

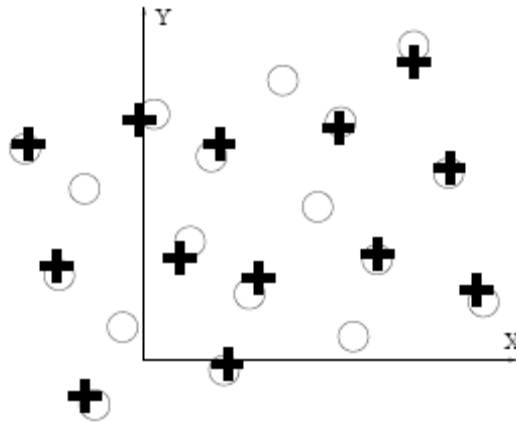
Mayın tarlasının yerleşim karakteristiğini oluşturmak, bir olasılıklar haritası oluşturmakta ve robot araştırma güzergahı tasarlamakta yardımcı olur. İki çeşit tipik dağıtım modeli mevcuttur: dağınık model ve düzenli model. Dağınık modeller genel olarak havadan veya mermilerle atılarak gerçekleştirilenler. Eliptik çarpma paternleri

darbenin doğrudan merkezi etkilediği yüksek yoğunluklu çarpışma modelidir. Mayınlar araçlar veya insanlar tarafından plana göre yerleştirildiğinde, mayın tarlalarının düzenli bir yapıya dönüşmesi askeri doktrinlere göre kuvvetle muhtemeldir. Düzenliliğin tipik karakteristik özellikleri karşılıklı olması ve eşit yer kaplamasıdır. Bu da mayınların birbirleriyle paralel sütunlar oluşturacak şekilde belirgin aralıklarla dizilmiş olmasını gerektirir. Şekil 3.10 bu modele örnektir.



Şekil 3.10. Eşit aralıklı düzenli paterne örnek.

Bir mayın tarlası düzenli bir dağılıma sahipse; istenilen mayın yerleşim şekli birçok parametrenin bir araya gelmesi şartı ile oluşturulabilir. Düzenli dağılım modeline göre yapılan grafik Şekil 3.11'dedir.



Şekil 3.11. Düzenli dağılım paterni örneği. Yuvarlaklar karakteristik fonksiyona göre beklenen yerler, (+) işaretinin olduğu yerler ise mayın tespit edilen yerler.

Model altı tane parametrenin bir araya gelmesi ile oluşturulur. Bunu oluşturmanın anahtarı ise bu süreç boyunca gerçekleşen uzaysal dağılımdır. Örnek model bilgisi, uygun araştırma stratejisi tasarımında ve robotların uzaysal yerleşimindeki gerçek zamanlı karar modellerinde kullanılabilir. Bu çalışma aynı zamanda metodolojiyi geliştirmenin başlangıç noktası olması bakımından da önemlidir.

Metodumun gereksinimleri aşağıdaki gibidir:

- Metot kesin olmayan bilgilerle işlem yapmak durumundadır. Mayın yerleşimini belirlemek için bir strateji oluşturmak zaten yeterince zordur. Detektör hataları sorunu büyötmektedir. Detektör yanlış negatif ve yanlış pozitif sinyaller üretmekte ve mayının yeri hakkında yanlış belirleme yapabilmektedir. Modelin sonuçlarının gerçek dünyadakiler ile sapmalar göstermesinin muhtemel sebepleri planlı yerleşme hataları, mayın patlaması ve modelin basitleştirilmesidir. Bütün bu hatalar temeli oluşturan yapıyı kurmakta zorluklar meydana getirmektedir.
- Metot ölçümsel olarak etkin olmalıdır. Robot üzerindeki ölçümün gerçek zamana uygun şekilde bitirilmesi oldukça önemlidir. Mayın yerleşiminin olasılık haritasını sürekli güncellemek için detektör bilgisi kullanılmalıdır. Ayrıca, güncellenen haritaya dayanan gerçek zamanlı yol planlaması robota mutlak surette yüklenmelidir.
- Metot zaman içinde farklı detektörlerden gelen sensör okumalarını birleştirmelidir. Sensör okumalarının zamanla birleştirilmesi gürültüyü tolere edebilir ve mayın tarlası modelindeki belirsizlikleri gidermek için gereklidir.

Tüm kapsam; zamanın boşa harcanması olarak değerlendirilebilir veya mevcut sınırlı bütçeyle robotlar açısından uygulanması mümkün olmayabilir. Bu yüzden, mayın yerleşiminin probalistik haritasına erişimi sağlayacak bir probalistik planlama üzerinde durulmalıdır. Planlayıcı, robotu zamanın bütün hedef alanın taranması için yeterli

olmadığı durumlarda yönlendirebilir. Gelenbe ve Cas robotun uzaysal dağılım önceliği temelinde oluşturulan önceden belirlenen alanları taraması stratejisini tartışmaktadır [36]. Mayın tarlası hakkındaki öncelikli bilgiler mayınların yerlerinin olası dağılımı ile elde edilir. Mayın bulma oranını yükseltmek için Basitleştirilmiş Sonsuz Ufuk Optimizasyonu (BSUO) tasarlanmıştır. Öncelikle, robot pozisyonlarını tarif etmek için en uygun işlem uygulamaya konulmuştur. Daha sonra, BSUO algoritması, robotun mayın bulunması olasılığı yüksek olan çeşitli alanlara yönlendirilmesi konusunda kararlar oluşturur. BSUO algoritması her adımdaki geçiş oranlarını hesaplar ve böylece robotun hedeflenen alanda gittiği her noktada mayın bulma olasılığını artırmış olur. BSUO algoritması kullanmanın avantajı ise yalnızca robotun algısına dayalı olması değil, aynı zamanda yerel bilgilerden de faydalanarak algının bütün alanı kapsamasını sağlamasıdır.

Bununla beraber, çoğu durumda, mayın yerleşiminin probalistik haritası bilinmemektedir. Bu yüzden, mayın tarlasını araştırarak probalistik haritayı oluşturmak öncelikli hedefdir. Mayın taraması ile ilgili istatistiksel sonuçlarda, mayın tarlasının yerleşim düzen karakteristiği hem düzenli, hem de düzensiz mayın tarlası çeşitlerinde ortak unsurdur. Modeller, uçaklardan alınan görüntülerle elde edilen mayın haritalarının, mayın tarlalarını analiz etmek için kullanılan diğer bir yöntemdir.

Lake ve Sadler metodu, belirli bir mayın tarlası örneğinde karşılıklılık ve düzenliliği öngörür. Bu yaklaşımın avantajı mayın tarlasındaki düzenliliği etkin bir şekilde algılamasıdır. Bu yüzden, bu metot sayesinde muhtemel örnekler hakkında ön bilgi sahibi olmadan da düzenliliği algılayabiliriz. Mayın tarlası araştırmasının iki basamaklı prosedürü, öncelikle standart yaklaşımın farklı biçimlerini kullanarak karşılıklı noktaların belirlenmesidir. Hough dönüşümü ve devir (mayınların aralığı) ise yenilenmiş Euclid algoritmasını oluşturur.

Karşılıklılığı ve düzenliliği denetlemenin bir başka yolu ise mayın tarlasını nokta işlem olarak modellemektir. Bu modelin temel önermesi mayınların birbirlerine paralel

olarak ve sabit aralıklarla dizilmiş olmasıdır [37]. Mayınlar arasındaki uzaklık, sütunların yönü ve birbirleri arasındaki mesafe de ayrıca değerlendirilmelidir.

3.2.1.3 Robot öngörülen bağımsız hareketi:

Bu yöntem, kalabalıkta veya kalabalık bir çevre içinde robot yönetim problemleri ile ilgilidir. Böyle bir çevrede çalıştırılan bir robot hareket eden insanlar ve diğer nesnelere tarafından kolayca engellenebilir. Bu problemi çözmek için insanların ve engellerin hareket yörüngesini önceden haber vermek düşünülmüştür. İki çözüm yolu ortaya konulmuştur: kısa-dönem ve uzun-vadeli. Kısa-dönem tahmini bir sonraki adımı tahmin etmeye yöneliktir. Uzun-vadeli tahmin ise engelin hareketinin belirlenmesini ön plana çıkarır. Robot hareketi, Kısmen Gözlenebilir Markov Karar İşlemi (KGMKİ) ile kontrol edilir. KGMKİ, bir robotun yeri hakkında bilgi modeli oluşturmak ve probalistik tavrındaki duyuşsal bilgi yeteneğinden dolayı kullanılır. KGMKİ'nin çözümü oldukça pahalı olduğundan dolayı; KGMKİ'nin hiyerarşik temsili kullanılmaktadır.

Mobil robotlar için; dinamik yönetim ve gerçek-alan çalışmaları zor ve karmaşık görevlerdir. Bu tür çevreler, karmaşık yapıları ve insanların hareketleri ve var olan engellerle nitelendirilir. Bu tür durumlarda, hareket eden bir robot, hareket eden insanlar ve engeller tarafından kolaylıkla engellenebilir ve hareketsiz kalabilir. Yolu boşaltılmadığı sürece gitmesi istenilen pozisyona ulaşması mümkün değildir. Bu tür bir durumun olmasını engellemek için birçok araştırmacı insanların ve engellerin hareketini önceden bildirmeye çalışmıştır. Gelecek hareketin tahmini, robotun, takip ettiği yolun engelli olup olmadığını tahmin etmesini ve eğer öyleyse engelle karşılaşmadan önce yönünü değiştirmesini sağlar.

Gelecek hareket tahmini insanlara özgü bir davranıştır. Örneğin bir caddede karşıdan karşıya geçmeye çalışan bir adamı ele alalım. Adam, yaklaşan bir aracın ne kadar süre sonra kendi bulunduğu alana varacağını tahmin eder ve sonrasında yolu geçip geçmemeye karar verir.

Genel olarak ifade etmek gerekirse, bir robot, yapması gereken hareketler konusunda her bir zaman adımında algılayıcılarının çevrenin durumu hakkında sağladığı bilgileri değerlendirerek kararlar almak zorundadır. Bu durumda, robot karar verme sırası problemini çözmek durumundadır. Karar verme sırası probleminin çözümünün, robotun her zaman kendi durumunu bildiği ve tamamen gözlemlenebilir bir çevrede gerçekleşmesine Markov Karar İşlemi (MKİ) denilir. Robotun durumunu belirlemek için yeterli bilgi yoksa veya belirsizlik söz konusuysa probleme, Kısmen Gözlenebilir Markov Karar İşlemi (KGMKİ) denilir.

İnsanların ve engellerin gelecek hareket tahmini ile KGMKİ tarafından modellenen hareket sırasını belirleme problemini birleştirmek amaçlanmıştır. Gelecek hareket tahmini konusunda yapılmış eski çalışmalar sadece yerel çarpışmadan kaçınma modüllerinde kullanılmıştır [38]. Bu yöntemde ise, global hareket modelinde tahmin bilgisini kullanmaktadır. Ayrıca, KGMKİ modeli hareket görevini formüle eder çünkü bilgiye elde etmenin probalistik yolunu sağlar ve yerleşme sorununu da çözer. KGMKİ önceleri robotikte yer almıştır fakat sadece basit çevrelerde [39][40] ve yüksek seviye görev planlayıcısı [41] olarak ayrı bir modülle çarpışmayı engellemek için kullanılmıştır. Bu yaklaşım, bağımsız robot hareketi için gelecek hareket tahmin modülü kullanımıyla yer belirlemeyi ve çarpışmayı önlemeyi birleştiren KGMKİ modelinin kullanılmasını önermektedir.

4. BULGULAR VE TARTIŞMASI:

Genel olarak kullanılan programın kurgusu şu şekilde oluşturulmuştur:

Mayın tarama robotları simülasyonu yazılım tasarımında (model-view-controller) model-arayüz-denetçi olarak da bilinen yazılım mimari şablonu kullanılmaktadır. Bu mimari şablona karşılık gelen temel yazılım birimleri gruplandırması şu şekilde yapılmaktadır:

1. Gui (arayüz)
2. Mayın tarlası (model)
3. Uygulama (denetçi)

4.1 GUI MODÜLÜ:

Bu modül içinde uygulamanın kullanıcı arayüzü ile ilgili nesnelere tanımlanmakta ve kullanıcı arayüzü servisleri ile kullanıcının uygulamanın işleyişi ile ilgili yapılandırma parametrelerini girmesi sağlanırken bir taraftan da uygulamanın çıktısı kullanıcıya gösterilmektedir. Kullanıcı gösterilen grid içinde robotların hareketlerini ve mayın tarlasında yapılan araştırmanın durumunu 1s'lik simülasyon zamanı içindeki ilerlemeyi görerek takip edebilmektedir.

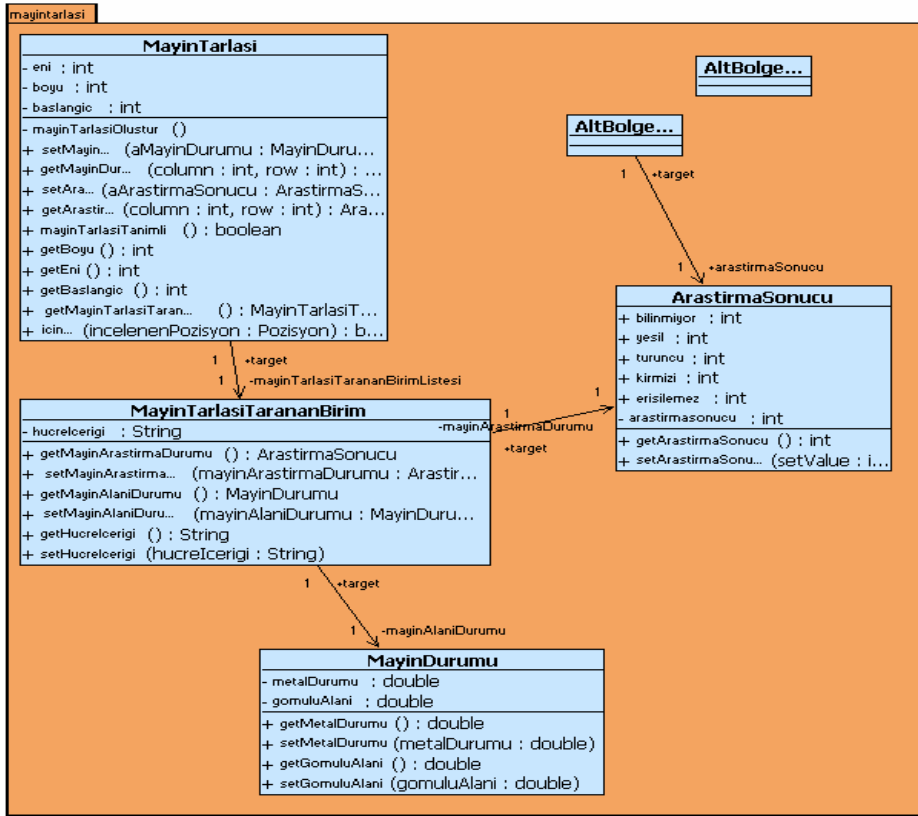
4.2 MAYINTARLASI MODÜLÜ:

Bu modülün sınıf çizeneği Şekil 4.1'de verilmektedir. Bu modülün tercih edilen mimari yaklaşım gereği rolü arayüz ile denetçi arasındaki veri alışverişini sağlamaktır. Uygulamanın iş mantığının çıktıları bu model üzerine yansıtılırken kullanıcı arayüzü modülü de bu modeldeki veri güncellemeleri ile simülasyonu gerçekleştirmektedir. Bu modülün bir diğer görevi de mayın tarlası içinde parça parça taranacak (4x4'lük) alt alanların ve bunların dolanım sırasının belirlenmesidir. Alt alanlar dolanım sırası ise

temel yönler verilen sırada göz önünde bulundurularak oluşturulmaktadır. Sıralama da esas alınan yön sırası:

1. Doğu
2. Batı
3. Kuzey
4. Güney

Bu sıra mayın tarlasını tarama ile ilgili taktik bir sıra olarak tanımlanmış ve kod içine gömülüdür.

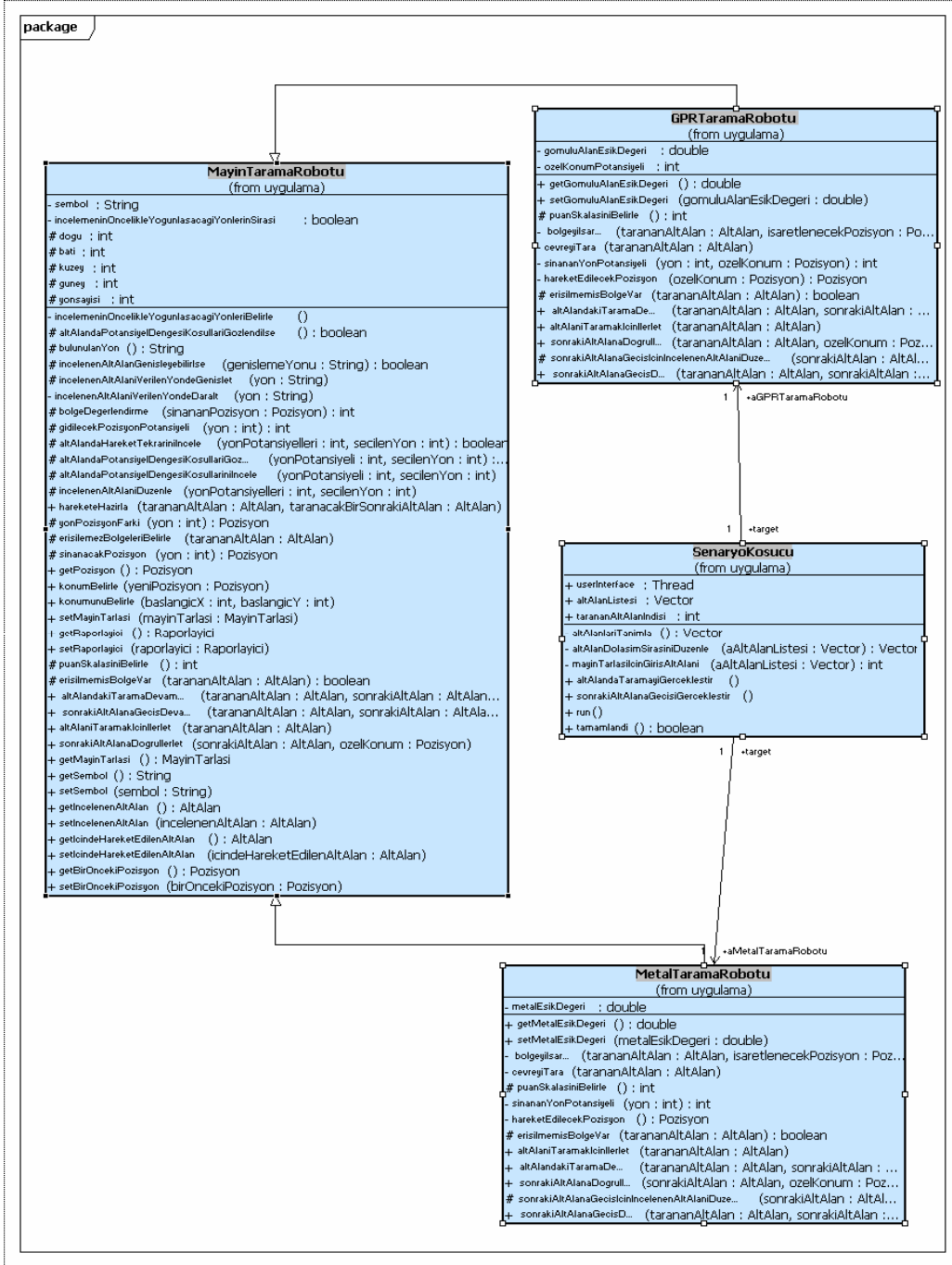


Şekil 4.1. MayınTarlasi Modülü sınıf çizeneği.

4.3 UYGULAMA MODÜLÜ

Bu modülün görevi ise mayın tarlası temsili üzerindeki bilgileri kullanarak ve işleyerek robot hareketleri simülasyonunun yapılmasıdır. Bu modülde tanımlanan iş mantığı kuralları ile robot hareketlerinin minimum manevra ve en az hareket tekrarı ile

gerçeğe olabildiğince mayın tarama faaliyetlerini tamamlamaları sağlanmaya çalışılmaktadır. Şekil 4.2’de uygulama modülünün sınıf çizeneğine yer verilmektedir.



Şekil 4.2. Uygulama modülü sınıf çizeneği.

SenaryoKosucu aktif nesnesinin temel görevi bütün bir simülasyonu baştan sona koşarak mayın tarlasının taranması için robotların sürülmesidir. Bu işlevi iki temel adımda yerine getirilir. Bu adımlar alt alan içinde mayın taramasının gerçekleştirilmesi ve Metal ve GPR robotlarını taranması tamamlanmış bir alt alandan sıradaki diğer alt alana geçişlerinin gerçekleştirilmesidir. Bu işlevi mayın tarlasında tanımlı tüm alt alanlar (4x4 lük) tamamlanana kadar da bu çevrimin çalışmasını sağlamaktadır.

MayınTaramaRobotu sınıfı ise Metal ve GPR tarama robotlarının hareketinin benzetimini yapan nesnelere türediği bir sınıf olarak ortak işlevleri ve veri yapılarını kılıflamaktadır. MetalTaramaRobotu ve GPRTaramaRobotu sınıflarının ise kendi hareket karakteristiklerini yansıtacak şekilde hareket etme ve hareketin yönüne karar vermelerini sağlayan işlevlerden oluşmaktadır. Her iki robot için de hareket etme kararının oluşması koşulu ile hareketin yönüne karar verme yaklaşımları temel olarak eylem potansiyellerinin hesaplanması yaklaşımı üzerine kurulmuştur.

MetalTaramaRobotu ve GPRTaramaRobotu sınıfların çözmesi beklenen bir diğer sorun ise alt alan içindeki dolanımlar esnasında ortaya çıkabilecek karar vermelerini zorlaştıran aykırı durumların oluşması ihtimalini ortadan kaldırmaya dönük olarak gerçekleştirilen işlevlerdir. Bu kapsama giren ve ele alınarak çözülmeye çalışılan iki temel sorun olmaktadır.

1. Denge potansiyelinin giderilerek hareket yönüne sağlıklı karar verilmesi
2. Denge potansiyelinin gözlenmediği ancak kendini tekrar eden faydasız hareket çevrimlerinin gözlenmesi ve bunların giderilmeye çalışılması problemi.

Her iki tür problemin de anlaşılması için öncelikle bir hareket yönü kararının mevcut koşullar altında verilmiş olması gerekmektedir. Bu kararın verilmesinin sonrasında hareket yönü kararının alınmasında referans alınan yön potansiyelleri problem analiz işlevlerine parametre olarak geçirilmektedir.

Denge potansiyelinin gözlenmesi koşulu seçilen bir yön ile aynı potansiyelde bir başka yönün olup olmadığının belirlenmesi ve böyle bir durumun gözlenmesi durumunda incelenen alt alanın daraltılmasıyla hesaplanacak potansiyel değerlerini değiştirilmesi ve bu şekilde robotun yönlendirilmesine dayanmaktadır. Daraltılan alan içinde erişilmemiş herhangi bir hücre kalmayana kadar robot hareketi devam ettirilecek bu durum gözlendiğinde incelenen alan tekrar önceki boyutlarına çekilerek var ise geri kalan bölgeler sınanmaya devam edilmektedir. Bir kez bu işleme tabi olmuş bir hücre tanımlanan alt alan sınırların müsaade ediyorsa tekrar ortaya çıkacak bir dengesizlik giderilmek üzere daraltılmakta, bu işlem ile hareketin iyileştirilmesi söz konusu olabilmektedir.

Faydasız hareket çevrimlerinin gözlenmesi durumu ise robotun bir önceki pozisyonuna bir an sonra geri dönmesi ve bunu tekrarlaması durumudur. Bu aykırılığın gözlenmesi durumu ise önceki tür aykırı durumun gözlenmesinden bağımsız olarak gözlenebilen bir durumdur. Bu durumun gözlenmesi durumunda da yine tekrar eden hareketin doğrultusuna dik yönlerden uygun olan seçilerek incelenen alanın potansiyeli değiştirilmeye çalışılmaktadır. Bu şekilde belirlenen incelenen alanda tüm bölgelere erişildikten sonra yine parçası olduğu büyük alt alanın sınırları ile kendi sınırları eşleştirilerek tüm alt alanı kapsayacak harekete devam edilmektedir.

Metal ve GPR robotları arasında tanılanmış olan dolanım sırası da yine alt alanda hareket etme kararları üretilerek kontrol edilmektedir. Metal robotu için alt alanda harekete son verme koşulu içeriği bilinmeyen erişilebilir bir hücre kalmamış olması ya da ilerlemeyi engelleyen yan yana bir grup şüpheli alanın gözlenmiş olması koşuludur. Bahsedilen durumlardan birinin oluşması ile metal robotunun hareketi durdurulmaktadır. Buna karşın GPR tarama robotunun hareketine son vermesi koşulu ise bilinmeyen hiçbir erişilebilir alanın olmadığı bir alt alanda en az bir şüpheli alanın tespit edilmiş olması ya da metal robotunun durmasına sebep olan bir engelleyici şüpheli bölgenin gözlenmiş olmasıdır. Karşılıklı olarak hareket edilmesini sağlayan bu kurallar ile robot hareketleri kontrol altında tutulmaktadır.

Metal ve GPR robotlarının gerçekleştirdiği bir diğer ortak hareket ise tarama işleminin tamamlandığı bir alt alandan alt alan listesinde tanımlanmış bir sonraki alt alana geçiş hareketidir. Bu hareket de yine alt alan içinde hareket edilmesini sağlayan işlev ile aynıdır. Hareketin sıradaki alt alana doğru yapılabilmesi için robotları ve sonraki alt alanı da içine alacak şekilde büyük bir alt alan yaratılır ve böylece sonraki alt alandaki bilinmeyen bölgelerin yarattığı potansiyel ile metal robotunun doğal olarak ilerlemesi gereken yönde yol alması sağlanmaktadır. GPR robotu ise yine kendisi için tanımlanmış benzer özellikteki bir alt alan içinde metal robotunu takip etmesini sağlayacak potansiyel üretilerek sonraki alt alana doğru hareket ettirilir ve her iki robotun hareketi de sonraki alt alan olarak tanımlanmış alt alan sınırları içinde ulaştıkları temiz bir alana girmeleriyle son bulmaktadır.

Bu çalışmada amaç iki robot kullanarak tanımlanan alanda mayın taraması yapılmasıdır. Taramayı yaparken belirlenen başlangıç koşulları şu şekildedir:

- Kullanılacak robotlar metal ve GPR detektörüne sahiptir.
- Taranacak alanın sınırları belirlenmiştir.
- Alan düz ve engellerden arındırılmıştır.
- Yüzey örtüsü robot hareketini engellememektedir.
- Sınırları aşması engellenmek için robotun kendi haritasında alan sınırlarını belirlemektedir.
- Robotlarda ve ana merkezde aynı haritaya sahiptir.
- Her robot okuduğu değeri diğer robota ve ana merkeze bildirmektedir.
- 2. robotun tarama alanı 1. robotun kendine gönderdiği turuncu alanlardır.

Bu çalışmada ana alan (4x4)'lük görev alt karelere bölünmektedir. Robotların görevleri alt kareleri belirli bir sıraya göre tarayarak tüm alanın taramasını bitirmektir. Robot harekete başlamadan önce ana kare büyüklüğü tanımlanmaktadır. Buna göre alt karelere numaralandırılıp ve hareketin sınırları belirlenmektedir.

Metal ağırlığı ve alan değeri için belirlenen kritik değerler bilgisayara girilebilmektedir. Bizim çalışmada metal eşik değeri 2g. ve alan eşik değeri 25cm²'dir. Bu değerler gömülü mayınlardan metal detektörleri ve GPR detektörleri vasıtası ile alınan bilgilerin eşleştirilmesi mantığı ile verilmektedir. Bu değerler aslında metal detektöründe oluşan akım nedeniyle üretilen ses sinyalinin yüksekliği ile ilişkilendirilmesi sonucu elde edilmektedir. Buradaki girilen eşik değeri ağırlık yerine detektörün metal miktarına göre ürettiği ses sinyallerinin ilişkilendirilmesi ile desibel cinsinden de girilebilir. Aynı şekilde alan eşik değeri de GPR detektörünün ekranındaki kapladığı alanla mayının yüzey alanının ilişkilendirilmesi sonucunda oluşturulmaktadır. Bu alan değerini detektör ekranındaki alan üzerinden piksel olarak de vermek mümkündür. Bu çalışmadaki temel amaç okunan değerlerin türü ve büyüklüğünden çok okunan değerlere uygun hareketlerin tanımlanmasıdır.

Hareket sıralaması önceliği yukarıda belirtildiği gibi doğu, batı, kuzey ve güneydir. Taramalardaki mükerrerlikleri önlemek için taranan alanlar bellekte tutularak ve harekete ters yönde devam edilmektedir.

Alt karelerin taranması sırasında zamanı da etkin olarak kullanmak bakımından kareler renk değerlerine göre puanlanmakta ve hareket yönelimi buna göre yapılmaktadır.

Robot hareketi sırasında bulunduğu kareye göre yönsel puanlama yapmaktadır. Yapılan hesaplama sonucuna göre puanı en yüksek olan doğrultuda harekete devam edilmektedir. Böylece taranmamış bölgelere daha çabuk ulaşılarak ve görev hızlı bir şekilde tamamlanmaktadır.

Robotlar alt alanlar arasında geçişlerde en kısa yolu belirleyip ona göre hareket etmektedir. Bunu bulunduğu bölgeden gitmek istediği bölgeye gidinceye kadar geçmesi gereken temiz kareleri sayıp en kısa olanı seçerek yapmaktadır. Bu güvenlik açısından

en uygun yöntemdir. Robot bu hareketini sadece yeşil olarak tanımlanmış bölgelerden geçerek gerçekleştirmektedir.

Robotlar alt karenin taranmasını bitirdiklerinde tanımlanmamış alan kalmamasını kontrol etmektedirler. Hareket komutlarına ait program EK'tedir.

Öncelikle mayın aranacak alana ait mayın tarlası benzetimi yapılmaktadır. Burada amaç detektör verisinin gerçekliğinden çok robotu okunduğu varsayılan bu veriler karşısındaki uygun hareket tarzının belirlenmesini sağlamaktır. Bu nedenle metal ve GPR detektörlerinden okunduğu varsayılan değerler ilgili hücelere girilerek mayın tarlası benzetimi yapılabilir. Bu hücelere girilen değerler robot tarafından tehlike değerleriyle karşılaştırılmakta ve hücreye sonuca uygun renk değeri atanmaktadır. Metal robotu okunan değer sonuçlarına göre ilgili kareye;

İlk okunan değerde temiz olduğu anlaşılırsa : Yeşil

Robot şüpheli gördüyse : Turuncu

renkleri atanmaktadır.

Metal detektörüne sahip robot aramasını yaptıktan sonra şüpheli işaretlediği alanlar GPR detektörüne sahip robot tarafından kontrol edilmektedir. GPR detektörüne sahip robot ilgili kareye;

Eğer alandaki şüpheyi devam ettirecek bir değer okuduysa: Kırmızı

Alanda tehlike belirtecek bir değer okumadıysa : Yeşil

renklerini atamaktadır.

Eğer alanı ilk tarayacak olan metal detektörüne sahip robotun önüne hareketini engelleyecek bir sıralama gelmesi durumunda (sıra veya sütun şeklinde bir engel) metal detektörüne sahip robot GPR detektörüne sahip ikinci robotu çağırmakta ve onun temiz

(yeşil) olarak işaretlediği ilk bölgeden planlanan hareketine devam etmektedir. Alanda aynı anda sadece bir robot çalışmaktadır.

Oluşturulan algoritmanın test edilmesi için Java dilinde yazılan program kullanılmaktadır. Yazılan program Java Runtime Enviroment (JRE) 5.0 veya daha üzeri bir sürümü kullanmaktadır. Programın çalıştırılması ile gelen ekrana, taranacak bölgenin sınırlarının, belirlenen alandaki istenen hücelere kontrol edilecek varyasyonun verilerinin ve robotların çalışmaya başlayabileceği temiz satırın numarasının girilmesinden sonra program koşturularak sonuç test edilmektedir. Yapılan bu testler ve sonuçları temel hareketler bölümünde daha detaylı gösterilmektedir.

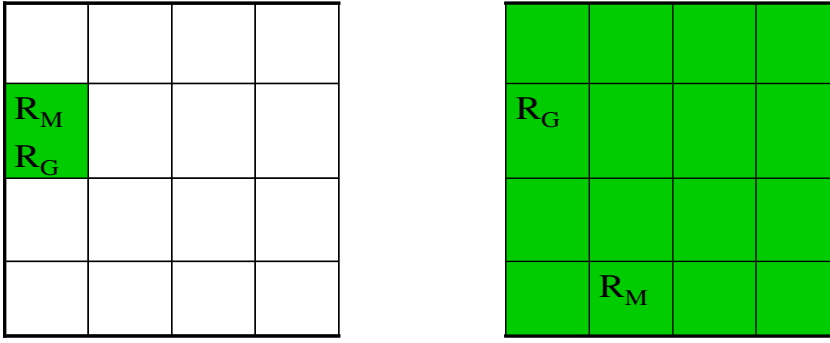
4.4 TEMEL HAREKETLER:

Bu bölümde robotların mayın tarlası benzetimi yapılan 4x4 alt alanda yapacağı temel hareketleri incelenmektedir. Mayın taramasında kullandığımız kapsama algoritmasına göre robotların yapması gerekenler;

- Mayın riski olan hüceleri belirlemesi ve oralardan geçmemesi,
- Alandaki bütün hücelerin kontrol edilmiş ve uygun işaretin atanmış olması,
- Gerektiğinde birbirleriyle iletişime geçerek hareket devamlılığının sağlanması,

şeklindedir.

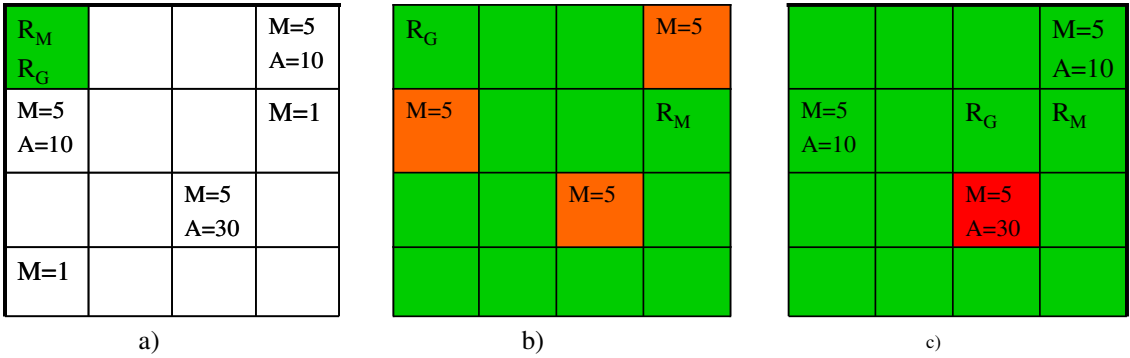
Bu kapsamda öncelikle belirlenen 4x4'lük alt alanda herhangi bir mayın riski olmadığı durumda bütün alanı tam olarak dolaştığının kontrolü yapılmaktadır. Hedef alandaki hücelere hiçbir metal değeri girilmediği durumla ilgili sonuç Şekil 4.3'te verilmektedir. Robotlar taramaya başlamadan önce güvenli olarak başlangıçta belirlenen hücrede olup ve alanın taranmasına buradan başlanmaktadır. Sonuç olarak alana hiçbir metal değeri girilmediği için R_M bütün alanı herhangi bir engelle karşılaşmaksızın taramakta ve görevini tamamlamaktadır.



Şekil 4.3. Temiz alanın taranmasına ait sonuç.

İkinci olarak test edilen durum R_M 'nin alanda şüpheli bir hücre olduğunda bunu belirleyebilmesi, buna uygun olarak şüpheli hücreyi işaretlemesi, tüm alandaki hücrelere veri ataması yapıldıktan sonra R_G 'yi çağırması, R_G 'nin şüpheli alanları kontrol etmesi ve uygun işaretlemeyi yaparak tüm alanın tam olarak taranmasının tamamlanmasıdır. Bu durumla ilgili sonuç Şekil 4.4'te verilmektedir.

Taraması yapılan alt alanın 16 hücresinden 5 tanesine metal değeri 3 tane alan değeri girilmiştir (Şekil 4.4.a). Girilen metal değerlerinden sadece 3 tanesi ve alan değerinden 1 tanesi eşik değer üzerinde. Kısaca alanda sadece 1 adet mayın olma ihtimali yüksek hücre bulunmaktadır. Buna uygun olarak R_M eşik değerinin üzerinde metal olan hücreleri turuncu olarak işaretlemekte ve aramasını tamamlamasını müteakip (Şekil 4.4.b) R_G 'yi çağırmakta ve R_G 'nin alandaki bütün şüpheli bölgelere bakarak uygun işaretlemeleri yapmaktadır (Şekil 4.4.c).



Şekil 4.4. Şüpheli ve riskli hücreler içeren alt alanın taranmasına ait sonuç.

Üçüncü olarak test edilen durum R_M 'nin önüne bir engel çıktığında, bütün satır veya sütun şüpheli olarak tanımlanırsa, R_G 'nin çağırılmasıdır. R_G 'nin R_M 'nin önünde engel teşkil eden bölgede hareketine devam edebilmesi için temiz bir geçiş sağlaması gerekmektedir. R_M 'nin bu geçişin açıldığını fark ederek bu geçitten yoluna devam ederek alanın taranmasını tamamlamaktadır. R_G 'nin diğer şüpheli bölgeleri de kontrol ederek alanın taranması tamamlanmaktadır.

Taraması yapılan alt alanın 16 hücresinden 8 tanesine metal değeri 5 tane alan değeri girilmiştir (Şekil 4.5.a). Girilen metal değerlerinden sadece 5 tanesi ve alan değerinden 2 tanesi eşik değerinin üzerindedir. Kısaca alanda sadece 2 adet mayın olma ihtimali yüksek hücre bulunmaktadır. Bu metal değeri eşik seviyesinin üzerindeki 4 hücre yan yana gelerek R_M 'nin önüne engel teşkil etmektedir (Şekil 4.5.b). Bu durumda R_M , R_G 'yi çağırmakta ve onun açtığı yoldan aramaya devam etmektedir (Şekil 4.5.c). R_M 'nin aramayı tamamlamasını müteakip R_G tarafından şüpheli olan hücreleri kontrol edilmekte ve kontrol edilen hücelere uygun işaretleme yapılmaktadır (Şekil 4.5.d).

	M=1		M=1
R_M R_G			
M=5 A=30	M=5 A=10	M=5 A=10	M=5 A=10
	M=1		M=5 A=30

a)

R_G	R_M		
M=5	M=5	M=5	M=5

b)

	R_G		
M=5 A=30	M=5 A=10	M=5 A=10	M=5 A=10
		R_M	M=5

c)

M=5 A=30	M=5 A=10	M=5 A=10	R_G
		R_M	M=5 A=30

d)

Şekil 4.5. R_M 'nin önüne bir engel çıktığında R_G 'yi çağırmasına ait sonuç.

Son olarak test edilen durum 12x12'lik bir alanda tam bir mayın taramasının yapılmasıdır. Taraması yapılan alan 9 alt alandan yani 144 hücreden oluşmaktadır. 144 hücreden 38 tanesine metal değeri 26 tanesine alan değeri girilmiştir (Şekil 4.6.a). Girilen metal değerlerinden 26 tanesi ve alan değerinden 11 tanesi eşik değerinin üzerindedir. Kısaca alanda 11 adet mayın olma ihtimali yüksek hücre bulunmaktadır. Bu taramaya ait sonuç Şekil 4.6.b'dedir.

M=1 R_M R_G M=3 A=30 M=4 A=10 M=5 A=15 M=8 A=5 M=5 A=29	M=1 M=4 A=10 M=5 A=29	M=1 M=1 M=4 A=10 M=1 M=5 A=29
M=4 A=10 M=1 M=1 M=1	M=1 M=4 A=10 M=45 A=450 M=4 A=10 M=1	M=5 A=5 M=5 A=29 M=4 A=10
M=5 A=10 M=4 A=10 M=5 A=29 M=1	M=5 A=29 M=1 M=4 A=10 M=5 A=29	M=5 A=29 M=4 A=10 M=4 A=10

(a)

Şekil 4.6. Tam bir mayın taramasına ait sonuç.

M=1					M=1				M=1
						M=5 A=29		M=1	M=4 A=10
M=3 A=30	M=4 A=10	M=5 A=15	M=8 A=5		M=4 A=10			M=1	
			M=5 A=29						M=5 A=29
		M=1	M=4 A=10						
M=4 A=10					M=45 A=450			M=5	M=5 A=29
	M=1	M=5 A=29			M=4 A=10	M=1			
	M=1				M=1				M=4 A=10
M=5 A=10		M=4 A=10			M=5 A=29				
						M=1		M=5 A=29	M=4 A=10
		M=5 A=29	M=1		M=4 A=10	M=5 A=29			R_G
M=1									R_M
								M=4 A=10	

(b)

Şekil 4.6. (devam) Tam bir mayın taramasına ait sonuç.

5. SONUÇLAR VE ÖNERİLER:

Bu çalışmada mayınlı bir bölgenin farklı detektörlere sahip iki robot tarafından taranması konusu incelenmiştir. Bilindiği üzere mayın temizleme çalışmalarındaki en temel problem detektörlerin yanlış alarm verme oranlarının yüksekliği, uzun zaman alması ve insanlarla yapılan aramlardaki hataların telafisinin zor olmasıdır. Bu nedenle bu konuda genel olarak robot ve robot gruplarının kullanımına yönelik bir eğilim vardır.

Yapılan çalışmada bu hususlar da dikkate alınarak 2 farklı detektör kullanımı ile hata oranını azaltma ve birden fazla robot kullanarak zaman kazanılması yoluna gidilmiştir. Barış zamanında yapılan mayın temizleme çalışmalarındaki gibi kapsama yol planlaması yapılarak bütün alanın tam olarak taranmasına yönelik bir algoritma geliştirilmiştir. Bu amaçla çalışmada GPR ve metal detektörüne sahip 2 robot kullanılmış ve bu robotların birbirleriyle iletişim etkileşim içinde alandaki mayınlı olma ihtimali yüksek bölgelerin tespit edilmesi sağlanmıştır. Oluşturulan algoritma Java dilinde yazılan bir program vasıtası ile test edilmiş ve belirlenen mayın tarlası verilerine göre istenilen sonuçların alındığı doğrulanmıştır. Bu program sayesinde probabilistik çalışmalarda elde edilen mayın tarlaları programa girilerek algoritmanın testleri yapılabilir ve elde edilen sonuçlara göre geliştirilmesine devam edilebilir.

Kara mayını tespitine yönelik yapılacak çalışmalar temel olarak detektör/ robot tasarımları ve geliştirmeleri olarak iki kısma ayrılmaktadır.

Detektör kısmında yapılabilecekler en geniş kullanım alanına sahip detektörü veya detektör kombinasyonlarını geliştirmek ve en iyi kalitede veri alma yöntemleri ile elde edilen verilerin gerçeğe en uygun olacak şekilde işlenmesini sağlayacak işleme teknikleri geliştirmek. Burada daha alt seviyede sınıflandırma yapılacak olursa detektör-gömülü madde etkileşimleri, tarama yapılan alandaki maddelerin gömülü olduğu ortamın özelliklerinin (fiziksel, kimyasal, elektriksel vs.) elde edilen verilere etkisini

belirlemek, kullanılan sinyalle arama derinliđi arasındaki optimizasyon sađlamak gibi alt konularda sıralanabilir.

Robotlarla ilgili kısmında yapılabilecekler ise, uygun kapsama algoritmalarının yazılması ve bu algoritmaların gereklerini yerine getirebilecek özelliklere sahip robotların tasarlanmasıdır. Algoritma konusunda yapılabileceklerin alt seviyede sınıflandırması yapılacak olursa,

- Tarama sırasında oluşabilecek anlamsız hareket tekrarlarını engellemek,
- Oluşabilecek durumları tam olarak belirleyebilmek ve uygun tanımlamaları yapabilmek, probabilistik yönteme ilginin temelinde olası durumların bilinmesi ve buna uygun çözümleri baştan tasarlama isteđidir,
- Enerjiyi verimli kullanmak bakımından kısa yol programlaması yapmak,
- Belirsiz alanın kalmaması veya bu durumun oluşması halinde uygun çözümler belirlemek, kapalı alan içinde kalan nokta teoremi kullanılabilir,
- Robotların etkin bir şekilde kullanılabilmesi için robotların bořta kalmalarını önlemeye yönelik çalışmalar yapmak,
- Ortak çalışabilecek robotlarda karar seviyesi tanımlamalarının belirlenmesi,

olarak sıralanabilir.

6.KAYNAKLAR:

- 1- “Büyük Larousse Sözlük ve Ansiklopedisi”, Gelişim Yayınları, s.:7887 (1986).
- 2- Yazıcı Y. “MSI Aylık Savunma Teknolojileri Dergisi”, sayı (014): 60-65, (2006).
- 3- Headquarters Department of the Army, Field Manuel 20-32, Washington, 185s. (1998).
- 4- Macdonald, J., ve Lookwood, J.R, “Alternatives for Landmine Detection” RAND Publication, Santa Monica, 366 s. (2003).
- 5- Nardulli, G. ve Marangi, C. “Trends in landmine warfare and landmine detection” USPID - VII Uluslararası Castiglioncello Konferansları, Castiglioncello- İtalya, (1997).
- 6- Yujiri L., Hauss B., ve Shoucri M., "Passive millimeter wave sensors for detection of buried mines", SPIE, s.: 2-6. (1995).
- 7- Smith, A., “IMAS and PPE requiriments”, Journal of Mine Action, **7(1)** (2003).
- 8- U.S. Department of Defense, “Office of Humanitarian and Refugee Affairs, Mine Facts: An Interactive CD-Rom Database”, Goss, C.F., ve Gander, T.J., (Ed), “Jane’s Military Vehicles and Logistics 1995-1996”, **(16):**239 (1995).
- 9- ARDEC, Public Affairs Office, "A Mine to Deter Pursuit", International Defense Review, s.: 1019 (1991).
- 10- Mike Croll, “The History of Landmines” s.: 37-52, Londra, (1998).
- 11- Bruschini, C. ve Gros, B. “A Survey Of Current Sensor Technology Research For The Detection Of Landmines” International Workshop on Sustainable Humanitarian Demining (SusDem'97), Zagreb, Hırvatistan, **(6):**18-27 (1997).
- 12- Russell K.L., McFee J.E., ve Sirovyak W., "Remote Performance Prediction for Infrared Imaging of Buried Mines", SPIE, s.: 762-769, (1997).
- 13- Ngan P., "Development of automatic target recognition for infrared sensor-based close range land mine detector", SPIE, s.: 881-889, (1995).
- 14- Das, Y., McFee, J. E. “Soil Properties Database for Humanitarian Demining: A Proposed Initiative” Council of International Union of Soil Science at the 17th World congress of Soil Science, Bangkok, Tayland,(2002).

- 15- Lloyd, S.R., "Electromagnetic Induction", Macdonald, J., Lookwood, J.R. (ed) "Alternatives for Landmine Detection", s.:85-92, RAND Publ., (2003).
- 16- L. S. Riggs, "Red Team Final Report: Comparison of the Sensitivity of Five Commonly Used Metal Detectors," (1999). (yayınlanmamış)
- 17- Borgwardt, C., "ODIS In Proceedings of WAPM'95", Lausanne, Switzerland, s.:37-43 (1995).
- 18- McFee, J.E. ve Carruthers, A. "A multisensor mine detector for peacekeeping- (ILDC)," in Proc.SPIE Conference on Detection and Remediation Technologies for Mines and Mine-Like Targets, **2765**, Orlando, FL, A.B.D., (1996).
- 19- Carin, L., "Ground Penetrating Radar", Macdonald, J.R.Lookwood (ed) "Alternatives for Landmine Detection" RAND Publ. s.:40-42 (2003).
- 20- Yujiri L., Hauss B. ve Shoucri M., "Microwave/ millimeter wave detection of metal and plastic mines", SPIE, s. 652-658, (1997).
- 21- van Kempen L., Nyssen E., Sahli H. ve Cornelis J., "Pattern Recognition Experiments for Ultra-Sonic and Radar AP-mine detection", SusDem, Zagreb, Hırvatistan, (1997).
- 22- Godzins, L., "X-RAY BACKSCATTER", Macdonald, J., Lookwood, J.R, (ed) "Alternatives for Landmine Detection" s.:191-204, RAND Publ. (2003).
- 23- Bruschini, C., Gros, B., Guerne, F., ve arkadaşları, "Ground penetrating radar and imaging metal detector for antipersonnel mine detection" Journal of Applied Geophysics, **40**: 59-71 (1998).
- 24- Nicoud, J.D., "Symposium on Technology and the Mine Problem", Monterey- A.B.D, (1996).
- 25- V. Kumarathanan and T. Nanayakkara "Intelligent Collaboration among Robotic Agents for Landmine Detection" Proceedings of the Annual Sessions of the Sri Lanka Association for Artificial Intelligence, SriLanka, (2005).
- 26- Cassinis R., Bianco G., Cavagnini, A., Ransenigo P., "Strategy for navigation of robot swarm to be used landmines detection", Third European Workshop on Advanced Mobile Robots, s.:211-218, Zürich, (1999).
- 27- Balch T.and Arkin R. C. "Communication in reactive multiagent robotic systems" Autonomous Robots, **1(1)**:27-52, (1994)

- 28- Latimer IV D., Srinivasa S., Hurst A., Choset H., and Shue, V. L., Jr. "Towards sensor based coverage with robot teams." International Conference on Robotics and Automation. IEEE, (2002).
- 29- Rekleitis, I. M., Dudek, G., ve Milios, E. E.. "Multi-robot collaboration for robust exploration" International Conference on Robotics and Automation IEEE, (2000).
- 30- Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Thrun, S. ve Younes, H. "Coordination for multi-robot exploration and mapping" National Conference on Artificial Intelligence, AAAI, (2000).
- 31- Yamauchi, B., "Frontier-based exploration using multiple robots" 2nd International Conference on Autonomous Agents, s.: 47-53, (1998).
- 32- Dias M. B. ve Stentz, A., "Free market architecture for distributed control of a multirobot system" 6th International Conference on Intelligent Autonomous Systems (IAS-6), s.:115-122, (2000).
- 33- Hou E.and Zheng D. "Mobile robot path planning on hierarchical hexagonal decomposition and artificial potential fields" Journal of Robotic Sys. s.: 605-614 (1994).
- 34- Acar E.U. ve Choset, H. "Critical point sensing unknown environments" IEEE ICRA 00 S.Fransisco CA (2000).
- 35- Zang, Y., Schervish M., Acar E.U. ve Choset H. "Probabilistic Methods for Robotic Landmine Search" 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems Maui, Hawaii, A.B.D., (2001).
- 36- Gelenbe,E., ve Cao, Y "Autonomous search for mines European Jurnal of Operational Resarch" **108**: 319-333 (1998).
- 37- Walsh, D., Raftery, A., "Detecting Mines in Minefields with Linear Characteristics," Technometrics, **41**: 34-44 (2002).
- 38- Foka, A.F. ve Trahanias, P.E., "Predictive Autonomous Robot Navigation" 200 IEEE/RJS Intl.Conference on Intelligent Robots and Systems EPFL, Lausanne, İsviçre (2002).
- 39- Cassandra, A. R., Kaelbling, L. P. ve Kurien, J. A.. "Acting under uncertainty: Discrete bayesian models for mobile-robot navigation" IEEE Int. Conf. Intell. Robots and Syst., (1996).
- 40- Koenig, S. ve Simmons, R. G. "Unsupervised learning of probabilistic models for robot navigation" Int. Conf. Robot. and Autom., s.:2301-2308 (1996).

- 41- Thrun, S., "Probabilistic algorithms and the inter-active museum tour-guide robot minerva" *Int. Jour. Robot. Resea.*, **19(11)**: 972-999 (2000).
- 42- Stentz, A., "Optimal and efficient path planning for partially known environments" *International Conference on Robotics and Automation*, **4**:3310-3317. IEEE, (1994).

EKLER:

JAVA PROGRAM KODLARI:

AltAlan:

package uygulama;

```
public class AltAlan {  
  
    private static final int Taranmadi = 0;  
  
    private int x1,x2,y1,y2;  
  
    private int taranmaDurumu;  
  
    public AltAlan() {  
        super();  
        taranmaDurumu = Taranmadi;  
    }  
    public AltAlan(AltAlan aAltAlan){  
        setX1(aAltAlan.getX1());  
        setX2(aAltAlan.getX2());  
        setY1(aAltAlan.getY1());  
        setY2(aAltAlan.getY2());  
        setTaranmaDurumu(aAltAlan.getTaranmaDurumu());  
    }  
    public boolean icinde (Pozisyon bolge){  
        boolean sonuc = false;  
        if ((bolge.getPosX()>= getX1())&&(bolge.getPosX()<=getX2()))  
            if ((bolge.getPosY()>= getY1())&&(bolge.getPosY()<=getY2()))  
                sonuc = true;  
        return sonuc;  
    }  
  
    public void alanSinirlariniEsitle(AltAlan aAltAlan){  
        setX1(aAltAlan.getX1());  
        setX2(aAltAlan.getX2());  
        setY1(aAltAlan.getY1());  
        setY2(aAltAlan.getY2());  
    }  
}
```

```

}

public String verilenAltAlaninBulunduguYon(AltAlan aAltAlan){
    String bulunduguYon = new String ("");
    if (aAltAlan != null){
        int doguBatiDogrultusundakiFark = getX1() - aAltAlan.getX1();
        int kuzeyGuneyDogrultusundakiFark = getY1() - aAltAlan.getY1();

        if (doguBatiDogrultusundakiFark<0)
            bulunduguYon = bulunduguYon+"dogu";
        else if (doguBatiDogrultusundakiFark>0)
            bulunduguYon = bulunduguYon+"bati";
        if (kuzeyGuneyDogrultusundakiFark<0)
            bulunduguYon = bulunduguYon+"guney";
        else if (kuzeyGuneyDogrultusundakiFark>0)
            bulunduguYon = bulunduguYon+"kuzey";
    }
    return bulunduguYon;
}

public int isTarandi() {
    return taranmaDurumu;
}

public void setTarandi(int tarandi) {
    taranmaDurumu = tarandi;
}

public int getX1() {
    return x1;
}

public void setX1(int x1) {
    this.x1 = x1;
}

public int getX2() {
    return x2;
}

public void setX2(int x2) {
    this.x2 = x2;
}

```



```

        public int getY1() {
            return y1;
        }

        public void setY1(int y1) {
            this.y1 = y1;
        }

        public int getY2() {
            return y2;
        }

        public void setY2(int y2) {
            this.y2 = y2;
        }

        public int getTaranmaDurumu() {
            return taranmaDurumu;
        }

        public void setTaranmaDurumu(int taranmaDurumu) {
            this.taranmaDurumu = taranmaDurumu;
        }

    }

```

GPRTaromaRobotu:

```

package uygulama;

import mayintarlası.ArastırmaSonucu;
import mayintarlası.MayınDurumu;
import mayintarlası.MayınTarlası;
import gui.Raporlayıcı;

public class GPRTaromaRobotu extends MayınTaromaRobotu {

    private double gomuluAlanEsikDegeri = 25;
    private final int özelKonumPotansiyeli = 5000;

    public double getGomuluAlanEsikDegeri() {
        return gomuluAlanEsikDegeri;
    }
}

```

```

public void setGomuluAlanEsikDegeri(double gomuluAlanEsikDegeri) {
    this.gomuluAlanEsikDegeri = gomuluAlanEsikDegeri;
}

public GPRTaramaRobotu(Raporlayici aRaporlayici, MayinTarlasi aMayinTarlasi) {
    super(aRaporlayici, aMayinTarlasi);
    setSembol("Rg");
}

@Override
protected int[] puanSkalasiniBelirle() {
    int puanSkalasi[] = {0,1,10,0,0};
    return puanSkalasi;
}

private void bolgeyilsaretle(AltAlan tarananAltAlan,Pozisyon isaretlenecekPozisyon){
    if (tarananAltAlan.icinde(isaretlenecekPozisyon)){
        if (getMayinTarlasi().getArastirmaAt(isaretlenecekPozisyon.getPosX(),
isaretlenecekPozisyon.getPosY()).getArastirmaSonucu()== ArastirmaSonucu.turuncu ){
            MayinDurumu mayinDurumu =
getMayinTarlasi().getMayinDurumuAt(isaretlenecekPozisyon.getPosX(), isaretlenecekPozisyon.getPosY());
            if (getGomuluAlanEsikDegeri() <=
mayinDurumu.getGomuluAlani()){
                getMayinTarlasi().setArastirmaAt(new
ArastirmaSonucu(ArastirmaSonucu.kirmizi), isaretlenecekPozisyon.getPosX(), isaretlenecekPozisyon.getPosY());
                getRaporlayici().raporla(" X: "+
isaretlenecekPozisyon.getPosX()+ " Y: "+ isaretlenecekPozisyon.getPosY()+ " pozisyonu kirmizi olarak
isaretlendi.");
            }
            else{
                getMayinTarlasi().setArastirmaAt(new
ArastirmaSonucu(ArastirmaSonucu.yesil), isaretlenecekPozisyon.getPosX(), isaretlenecekPozisyon.getPosY());
                getRaporlayici().raporla(" X: "+
isaretlenecekPozisyon.getPosX()+ " Y: "+ isaretlenecekPozisyon.getPosY()+ " pozisyonu yesil olarak
isaretlendi.");
            }
        }
    }
}

private void cevreyiTara (AltAlan tarananAltAlan){

```

```

        for (int i = 0; i < yonsayisi; i++){
            Pozisyon incelenen = sinanacakPozisyon(i);
            if (tarananAltAlan.icinde(incelenen))
                bolgeyilsaretle(tarananAltAlan, incelenen);
        }
    }

    private int sinananYonPotansiyeli(int yon,Pozisyon ozelKonum){
        int yonPotansiyeli = 0;

        switch (yon){
            case (dogu):
                {
                    for (int i=getPozisyon().getPosX()+1;i <= getIncelenenAltAlan().getX2();i++)
                        for (int j = getIncelenenAltAlan().getY1(); j <=
getIncelenenAltAlan().getY2(); j++){
                            yonPotansiyeli += bolgeDegerlendirme(new
Pozisyon(i,j));
                            if ((i ==
ozelKonum.getPosX())&&(j==ozelKonum.getPosY()))
                                yonPotansiyeli += ozelKonumPotansiyeli;
                        }
                    break;
                }
            case (bati):
                {
                    for (int i=getIncelenenAltAlan().getX1();i < getPozisyon().getPosX();i++)
                        for (int j = getIncelenenAltAlan().getY1(); j <= getIncelenenAltAlan().getY2():
j++){
                            yonPotansiyeli += bolgeDegerlendirme(new Pozisyon(i,j));
                            if ((i == ozelKonum.getPosX())&&(j==ozelKonum.getPosY()))
                                yonPotansiyeli += ozelKonumPotansiyeli;
                        }
                    break;
                }
            case (kuzey):
                {
                    for (int i=getIncelenenAltAlan().getX1();i <= getIncelenenAltAlan().getX2();i++)
                        for (int j = getIncelenenAltAlan().getY1(); j < getPozisyon().getPosY(); j++){
                            yonPotansiyeli += bolgeDegerlendirme(new Pozisyon(i,j));
                            if ((i == ozelKonum.getPosX())&&(j==ozelKonum.getPosY()))
                                yonPotansiyeli += ozelKonumPotansiyeli;
                        }
                }
        }
    }

```

```

        break;
    }
    case (guney):
    {
        for (int i=getIncelenenAltAlan().getX1();i <= getIncelenenAltAlan().getX2();i++)
            for (int j = getPozisyon().getPosY()+1; j <= getIncelenenAltAlan().getY2():
j++){
                yonPotansiyeli += bolgeDegerlendirme(new Pozisyon(i,j));
                if ((i == ozelKonum.getPosX())&&(j==ozelKonum.getPosY()))
                    yonPotansiyeli += ozelKonumPotansiyeli;
            }
        break;
    }
    default:break;
}
return yonPotansiyeli;
}

```

```

private Pozisyon hareketEdilecekPozisyon(Pozisyon ozelKonum){
    int enlyiPuan = 0;
    int secilenYon = 0;

```

```

    int puan[]={0,0,0,0} ;
    for (int yon = 0; yon < yonsayisi; yon++){
        if (gidilecekPozisyonPotansiyeli(yon) > 0)
            puan[yon] = sinananYonPotansiyeli(yon,ozelKonum);
        if (puan[yon] > enlyiPuan){
            enlyiPuan = puan[yon];
            secilenYon = yon;
        }
    }
}

```

```

    incelenenAltAlaniDuzenle(puan,secilenYon);
    return sinanacakPozisyon(secilenYon);
}

```

```

protected boolean erisilmemisBolgeVar(AltAlan tarananAltAlan){
    boolean sonuc = false;
    for (int i=tarananAltAlan.getX1(); (i <= tarananAltAlan.getX2())&&(sonuc == false);i++)
        for (int j=tarananAltAlan.getY1(); (j <= tarananAltAlan.getY2())&&(sonuc ==
false);j++)

```

```

        if (getMayinTarlasi().getArastirmaAt(i,j).getArastirmaSonucu() ==
ArastirmaSonucu.turuncu)
            sonuc = true;
        return sonuc;
    }

    @Override
    public boolean altAlandakiTaramaDevamEdiyor(AltAlan tarananAltAlan,
        AltAlan sonrakiAltAlan) {
        boolean sonuc = true;
        if (!erisilmemisBolgeVar(tarananAltAlan))
            sonuc = false;

        return sonuc;
    }

    @Override
    public void altAlaniTaramakIcinlerlet(AltAlan tarananAltAlan) {
        cevreyiTara(tarananAltAlan);
        Pozisyon sonrakiPozisyon = hareketEdilecekPozisyon(new Pozisyon(-1,-1));
        konumBelirle(sonrakiPozisyon);
    }

    @Override
    public void sonrakiAltAlanaDogrullerlet(AltAlan tarananAltAlan, Pozisyon ozelKonum) {
        cevreyiTara(tarananAltAlan);
        Pozisyon sonrakiPozisyon = hareketEdilecekPozisyon(ozelKonum);
        konumBelirle(sonrakiPozisyon);
    }

    protected void sonrakiAltAlanaGecisIcinIncelenenAltAlaniDuzenle(AltAlan sonrakiAltAlan){

        if (getIncelenenAltAlan() == null)
            setIncelenenAltAlan(sonrakiAltAlan);
        else
            getIncelenenAltAlan().alanSinirlariniEsitle(sonrakiAltAlan);

        if (sonrakiAltAlan.getXI()>getPozisyon().getPosX()) {
            getIncelenenAltAlan().setXI(getPozisyon().getPosX());
        }
    }

```

```

    }
    else if (sonrakiAltAlan.getX2() < getPozisyon().getPosX()) {
        getIncelenenAltAlan().setX2(getPozisyon().getPosX());
    }
    if (sonrakiAltAlan.getY1() > getPozisyon().getPosY()) {
        getIncelenenAltAlan().setY1(getPozisyon().getPosY());
    }
    else if (sonrakiAltAlan.getY2() < getPozisyon().getPosY()) {
        getIncelenenAltAlan().setY2(getPozisyon().getPosY());
    }
}
}

```

```

@Override
public boolean sonrakiAltAlanaGecisDevamEdiyor(AltAlan tarananAltAlan,
        AltAlan sonrakiAltAlan) {
    boolean sonuc = true;
    if (sonrakiAltAlan.icinde(getPozisyon()))
        sonuc = false;
    else {
        sonrakiAltAlanaGecisIcinIncelenenAltAlaniDuzenle(sonrakiAltAlan);
    }
    return sonuc;
}
}

```

MetalTaramaRobotu:

```
package uygulama;
```

```
import mayintarlası.ArastırmaSonucu;
import mayintarlası.MayınDurumu;
import mayintarlası.MayınTarlası;
import gui.Raporlayıcı;
```

```
public class MetalTaramaRobotu extends MayınTaramaRobotu {
```

```
    private double metalEsikDegeri = 2;
```

```
    public double getMetalEsikDegeri() {
```

```

        return metalEsikDegeri;
    }

    public void setMetalEsikDegeri(double metalEsikDegeri) {
        this.metalEsikDegeri = metalEsikDegeri;
    }

    public MetalTaramaRobotu(Raporlayici aRaporlayici, MayinTarlasi aMayinTarlasi) {
        super(aRaporlayici, aMayinTarlasi);
        setSembol("Rm");
    }

    private void bolgeyilsaretle(AltAlan tarananAltAlan,Pozisyon isaretlenecekPozisyon){
        if (tarananAltAlan.icinde(isaretlenecekPozisyon)){
            if (getMayinTarlasi().getArastirmaAt(isaretlenecekPozisyon.getPosX(),
isaretlenecekPozisyon.getPosY()).getArastirmaSonucu() == ArastirmaSonucu.bilinmiyor){
                MayinDurumu mayinDurumu =
getMayinTarlasi().getMayinDurumuAt(isaretlenecekPozisyon.getPosX(), isaretlenecekPozisyon.getPosY());
                if (getMetalEsikDegeri() <= mayinDurumu.getMetalDurumu()){
                    getMayinTarlasi().setArastirmaAt(new
ArastirmaSonucu(ArastirmaSonucu.turuncu), isaretlenecekPozisyon.getPosX(),
isaretlenecekPozisyon.getPosY());
                    getRaporlayici().raporla(" X: "+ isaretlenecekPozisyon.getPosX()+
" Y: "+ isaretlenecekPozisyon.getPosY()+ " pozisyonu turuncu olarak isaretlendi.");
                }
            }
            else{
                getMayinTarlasi().setArastirmaAt(new
ArastirmaSonucu(ArastirmaSonucu.yesil), isaretlenecekPozisyon.getPosX(), isaretlenecekPozisyon.getPosY());
                getRaporlayici().raporla(" X: "+ isaretlenecekPozisyon.getPosX()+
" Y: "+ isaretlenecekPozisyon.getPosY()+ " pozisyonu yesil olarak isaretlendi.");
            }
        }
    }
}

    private void cevreyiTara (AltAlan tarananAltAlan){
        for (int i = 0; i < yonsayisi; i++){
            Pozisyon incelenen = sinanacakPozisyon(i);
            if (tarananAltAlan.icinde(incelenen))
                bolgeyilsaretle(tarananAltAlan, incelenen);
        }
    }
}

```

```

}
protected int [] puanSkalasinBelirle (){
    int puanSkalasi[] = {1000,100,10,1,1};
    return puanSkalasi;
}

private int sinananYonPotansiyeli(int yon){
    int yonPotansiyeli = 0;

    switch (yon){
        case (dogu):
            {
                for (int i=getPozisyon().getPosX()+1;i <= getIncelenenAltAlan().getX2();i++)
                    for (int j = getIncelenenAltAlan().getY1(); j <=
getIncelenenAltAlan().getY2(); j++)
                        yonPotansiyeli += bolgeDegerlendirme(new
Pozisyon(i,j));
                break;
            }
        case (bati):
            {
                for (int i=getIncelenenAltAlan().getX1();i < getPozisyon().getPosX();i++)
                    for (int j = getIncelenenAltAlan().getY1(); j <= getIncelenenAltAlan().getY2();
j++)
                        yonPotansiyeli += bolgeDegerlendirme(new Pozisyon(i,j));
                break;
            }
        case (kuzey):
            {
                for (int i=getIncelenenAltAlan().getX1();i <= getIncelenenAltAlan().getX2();i++)
                    for (int j = getIncelenenAltAlan().getY1(); j < getPozisyon().getPosY(); j++)
                        yonPotansiyeli += bolgeDegerlendirme(new Pozisyon(i,j));
                break;
            }
        case (guney):
            {
                for (int i=getIncelenenAltAlan().getX1();i <= getIncelenenAltAlan().getX2();i++)
                    for (int j = getPozisyon().getPosY()+1; j <= getIncelenenAltAlan().getY2();
j++)
                        yonPotansiyeli += bolgeDegerlendirme(new Pozisyon(i,j));
                break;
            }
        default:break;
    }
}

```



```

    }
    return yonPotansiyeli;
}

private Pozisyon hareketEdilecekPozisyon(){
    int enlyiPuan = 0;
    int secilenYon = 0;

    int puan[] = {0,0,0,0} ;
    for (int yon = 0; yon < yonsayisi; yon++){
        if (gidilecekPozisyonPotansiyeli(yon) > 0)
            puan[yon] = sinananYonPotansiyeli(yon);
        if (puan[yon] > enlyiPuan){
            enlyiPuan = puan[yon];
            secilenYon = yon;
        }
    }

    incelenenAltAlaniDuzenle(puan,secilenYon);

    return sinanacakPozisyon(secilenYon);
}

protected boolean erisilmemisBolgeVar(AltAlan tarananAltAlan){
    boolean sonuc = false;
    for (int i=tarananAltAlan.getX1(); (i <= tarananAltAlan.getX2())&&(sonuc == false);i++)
        for (int j=tarananAltAlan.getY1(); (j <= tarananAltAlan.getY2())&&(sonuc ==
false);j++)
            if (getMayinTarlasi().getArastirmaAt(i,j).getArastirmaSonucu() ==
ArastirmaSonucu.bilinmiyor)
                sonuc = true;
    return sonuc;
}

@Override
public void altAlaniTaramakIcinlerlet(AltAlan tarananAltAlan) {
    cevreyiTara(tarananAltAlan);
    Pozisyon sonrakiPozisyon= hareketEdilecekPozisyon();
    konumBelirle(sonrakiPozisyon);
}

@Override
public boolean altAlandakiTaramaDevamEdiyor(AltAlan tarananAltAlan, AltAlan sonrakiAltAlan) {

```

```

        erisilemezBolgeleriBelirle(tarananAltAlan);
        return erisilmemisBolgeVar(tarananAltAlan);
    }

    @Override
    public void sonrakiAltAlanaDogrullerlet(AltAlan sonrakiAltAlan,Pozisyon ozelKonum) {
        Pozisyon sonrakiPozisyon= hareketEdilecekPozisyon();
        konumBelirle(sonrakiPozisyon);
        cevreyiTara(sonrakiAltAlan);
    }

    protected void sonrakiAltAlanaGecisIcinIncelenenAltAlaniDuzenle(AltAlan sonrakiAltAlan){

        if (getIncelenenAltAlan() == null)
            setIncelenenAltAlan(sonrakiAltAlan);
        else
            getIncelenenAltAlan().alanSinirlariniEsitle(sonrakiAltAlan);

        if (sonrakiAltAlan.getX1()>getPozisyon().getPosX()) {
            getIncelenenAltAlan().setX1(getPozisyon().getPosX());
        }
        else if (sonrakiAltAlan.getX2()<getPozisyon().getPosX()){
            getIncelenenAltAlan().setX2(getPozisyon().getPosX());
        }
        if (sonrakiAltAlan.getY1()>getPozisyon().getPosY()) {
            getIncelenenAltAlan().setY1(getPozisyon().getPosY());
        }
        else if (sonrakiAltAlan.getY2()<getPozisyon().getPosY()){
            getIncelenenAltAlan().setY2(getPozisyon().getPosY());
        }
    }

    @Override
    public boolean sonrakiAltAlanaGecisDevamEdiyor(AltAlan tarananAltAlan, AltAlan sonrakiAltAlan) {
        boolean sonuc = true;
        if (sonrakiAltAlan.icinde(getPozisyon()))
            sonuc = false;
        else{
            sonrakiAltAlanaGecisIcinIncelenenAltAlaniDuzenle(sonrakiAltAlan);
        }
        return sonuc;
    }

```

```
    }  
}
```

SenaryoKosucu:

```
package uygulama;
```

```
import gui.Raporlayici;
```

```
import java.util.Vector;
```

```
import mayintarlasi.MayinTarlasi;
```

```
public class SenaryoKosucu implements Runnable{
```

```
    MayinTarlasi aMayinTarlasi = null;
```

```
    Raporlayici raporlayici = null;
```

```
    MetalTaramaRobotu aMetalTaramaRobotu = null;
```

```
    GPRTaramaRobotu aGPRTaramaRobotu = null;
```

```
    Thread userInterface = null;
```

```
    Vector <AltAlan> altAlanListesi = null;
```

```
    int tarananAltAlanIndisi = 0;
```

```
public SenaryoKosucu(    MayinTarlasi mayinTarlasi,
```

```
    Raporlayici aRaporlayici,  
    MetalTaramaRobotu metalTaramaRobotu,  
    GPRTaramaRobotu gprTaramaRobotu) {
```

```
    super();
```

```
    aMayinTarlasi = mayinTarlasi;
```

```
    raporlayici = aRaporlayici;
```

```
    aMetalTaramaRobotu = metalTaramaRobotu;
```

```

aGPRTaromaRobotu = gprTaromaRobotu;

altAlanListesi = altAlanlariTanimla();

for (int i =0;i < altAlanListesi.size();i++) {
    AltAlan element = (AltAlan) altAlanListesi.get(i);
    raporlayici.raporla(this.getClass(),"Alt Alan : "+ (i+1) + " X1: "+element.getX1()+ " , X2:
"+element.getX2()+ " , Y1: "+element.getY1()+", Y2: "+element.getY2());
}
}

```

```

private Vector <AltAlan> altAlanlariTanimla(){
    Vector <AltAlan> aAltAlanListesi = new Vector<AltAlan>();
    for (int j = 0; j < aMayinTarlasi.getEni();j=j+4){
        int Y1,Y2;
        Y1 = j;
        Y2 = Y1 + 3;
        if (Y2 > aMayinTarlasi.getEni()){
            Y2 = aMayinTarlasi.getEni()-1;
            if (Y2-3 >= 0)
                Y1 = Y2-3;
            else
                Y1 = 0;
        }
        for (int i = 0; i < aMayinTarlasi.getBoyu();i=i+4){
            AltAlan aAltAlan = new AltAlan();
            aAltAlan.setX1(i);
            aAltAlan.setX2(aAltAlan.getX1()+3);
            if (aAltAlan.getX2() > aMayinTarlasi.getBoyu()){
                aAltAlan.setX2(aMayinTarlasi.getBoyu()-1);
                if (aAltAlan.getX2()-3 >= 0 )
                    aAltAlan.setX1(aAltAlan.getX2()-3);
                else
                    aAltAlan.setX1(0);
            }
            aAltAlan.setY1(Y1);
            aAltAlan.setY2(Y2);
            aAltAlanListesi.add(aAltAlan);
        }
    }
    return altAlanDolasimSirasiniDuzenle(aAltAlanListesi);
}

```

```

}

private Vector <AltAlan> altAlanDolasimSirasiniDuzenle(Vector <AltAlan> aAltAlanListesi){

    Vector <AltAlan> dolasimSirasinaGoreAltAlanlar = new Vector<AltAlan>();
    AltAlan bulunulanAltAlan = null;

    Integer [] dolanimSirası = new Integer[4];
    dolanimSirası[0] = new Integer(DolanimDuzenleyici.dogu);
    dolanimSirası[1] = new Integer(DolanimDuzenleyici.bati);
    dolanimSirası[2] = new Integer(DolanimDuzenleyici.kuzey);
    dolanimSirası[3] = new Integer(DolanimDuzenleyici.guney);

    DolanimDuzenleyici aDolanimDuzenleyici = new DolanimDuzenleyici(dolanimSirası);

    int girisAltAlanIndisi = mayinTarlasiIcinGirisAltAlani(aAltAlanListesi);

    bulunulanAltAlan = aAltAlanListesi.get(girisAltAlanIndisi);
    dolasimSirasinaGoreAltAlanlar.add(bulunulanAltAlan);
    aAltAlanListesi.remove(girisAltAlanIndisi);

    while (aAltAlanListesi.size() > 0){
        int bulunulanAltAlanIndisi = aDolanimDuzenleyici.gidilecekBirSonrakiAltAlan(aAltAlanListesi,
bulunulanAltAlan);
        bulunulanAltAlan = aAltAlanListesi.get(bulunulanAltAlanIndisi);
        dolasimSirasinaGoreAltAlanlar.add(bulunulanAltAlan);
        aAltAlanListesi.remove(bulunulanAltAlanIndisi);
    }

    return dolasimSirasinaGoreAltAlanlar;
}

private int mayinTarlasiIcinGirisAltAlani(Vector <AltAlan> aAltAlanListesi){
    int found = 0;
    for (int i =0; (i < aAltAlanListesi.size()) && (found == 0);i++){
        AltAlan incelenenAltAlan = (AltAlan)aAltAlanListesi.get(i);
        if
((aMayinTarlasi.getBaslangic())>=incelenenAltAlan.getY1())&&(aMayinTarlasi.getBaslangic())<=incelenenAltAlan.getY
2()))
            found = i;
    }
    return found;
}

```

```

public void altAlandaTaramayiGerceklestir(){

    AltAlan tarananAltAlan = null;
    AltAlan sonrakiAltAlan = null;
    tarananAltAlan = altAlanListesi.get(tarananAltAlanIndisi);

    if (tarananAltAlanIndisi+1 != altAlanListesi.size())
        sonrakiAltAlan = altAlanListesi.get(tarananAltAlanIndisi+1);

    aMetalTaramaRobotu.hareketeHazirla(tarananAltAlan, sonrakiAltAlan);
    aGPRTaramaRobotu.hareketeHazirla(tarananAltAlan, sonrakiAltAlan);

    while (aMetalTaramaRobotu.altAlandakiTaramaDevamEdiyor(tarananAltAlan,
sonrakiAltAlan)||aGPRTaramaRobotu.altAlandakiTaramaDevamEdiyor(tarananAltAlan, sonrakiAltAlan)){

        if (aMetalTaramaRobotu.altAlandakiTaramaDevamEdiyor(tarananAltAlan,
sonrakiAltAlan))
            aMetalTaramaRobotu.altAlaniTaramakIcinIlerlet(tarananAltAlan);

        else if (aGPRTaramaRobotu.altAlandakiTaramaDevamEdiyor(tarananAltAlan,
sonrakiAltAlan))
            aGPRTaramaRobotu.altAlaniTaramakIcinIlerlet(tarananAltAlan);

        try {
            Thread.sleep(1000);
            Thread.yield();
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

public void sonrakiAltAlanaGecisiGerceklestir(){

    AltAlan tarananAltAlan = null;
    AltAlan sonrakiAltAlan = null;
    tarananAltAlan = altAlanListesi.get(tarananAltAlanIndisi);

    if (tarananAltAlanIndisi+1 != altAlanListesi.size())
        sonrakiAltAlan = altAlanListesi.get(tarananAltAlanIndisi+1);
}

```

```

        aMetalTaramaRobotu.hareketeHazirla(tarananAltAlan, sonrakiAltAlan);
        aGPRTaramaRobotu.hareketeHazirla(tarananAltAlan, sonrakiAltAlan);
        while (aMetalTaramaRobotu.sonrakiAltAlanaGecisDevamEdiyor(tarananAltAlan,
sonrakiAltAlan)||aGPRTaramaRobotu.sonrakiAltAlanaGecisDevamEdiyor(tarananAltAlan, sonrakiAltAlan)){

                if (aMetalTaramaRobotu.sonrakiAltAlanaGecisDevamEdiyor(tarananAltAlan,
sonrakiAltAlan))
                        aMetalTaramaRobotu.sonrakiAltAlanaDogrullerlet(sonrakiAltAlan, new
Pozisyon(-1,-1));
                if (aGPRTaramaRobotu.sonrakiAltAlanaGecisDevamEdiyor(tarananAltAlan,
sonrakiAltAlan))

                        aGPRTaramaRobotu.sonrakiAltAlanaDogrullerlet(tarananAltAlan,aMetalTaramaRobotu.getPozisyon());
                try {
                        Thread.sleep(1000);
                        Thread.yield();
                } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }
}

public void run(){
        while (!tamamlandi()){
                altAlandaTaramayiGerceklestir();
                sonrakiAltAlanaGecisiGerceklestir();
                tarananAltAlanIndisi++;
        }
}

public boolean tamamlandi(){
        return (altAlanListesi.size() == tarananAltAlanIndisi);
}
}

```

DolanımDüzenlevici:

```

package uygulama;

import java.util.Vector;

```

```

public class DolanimDuzenleyici {
    public final static int dogu=0;
    public final static int bati=1;
    public final static int kuzey=2;
    public final static int guney=3;

    private Integer stratejikDolanimSirasi[];

    public DolanimDuzenleyici(Integer[] stratejikDolanimSirasi) {
        super();
        this.stratejikDolanimSirasi = stratejikDolanimSirasi;
    }

    public int gidilecekBirSonrakiAltAlan(Vector aAltAlanListesi,AltAlan bulunulanAltAlan){
        int gidilecekBirSonrakiAltAlanIndisi = -1;
        for (int i = 0; (i < stratejikDolanimSirasi.length)&&(gidilecekBirSonrakiAltAlanIndisi < 0) ; i++) {
            gidilecekBirSonrakiAltAlanIndisi = gidilebilecekAltAlanIndisi(aAltAlanListesi,
bulunulanAltAlan,stratejikDolanimSirasi[i].intValue());
        }
        if (gidilecekBirSonrakiAltAlanIndisi < 0)
            gidilecekBirSonrakiAltAlanIndisi = 0;
        return gidilecekBirSonrakiAltAlanIndisi;
    }

    private int gidilebilecekAltAlanIndisi(Vector aAltAlanListesi, AltAlan bulunulanAltAlan, int yon){
        int enUygun = -1;
        int uygun = -1;
        for (int i = 0 ; (i < aAltAlanListesi.size())&&(enUygun== -1);i++) {
            AltAlan altAlan = (AltAlan) aAltAlanListesi.get(i);
            switch (yon){
                case dogu :
                    if (((bulunulanAltAlan.getYI() - altAlan.getYI()) == 0 )
&&((altAlan.getXI()-bulunulanAltAlan.getXI()) > 0)){
                        uygun = i;
                        if ((altAlan.getXI()-bulunulanAltAlan.getXI()) <= 4){
                            enUygun = uygun;
                        }
                    }
                    break;
                case bati :
                    if (((bulunulanAltAlan.getYI() - altAlan.getYI()) == 0 )
&&((altAlan.getXI()-bulunulanAltAlan.getXI()) < 0)){
                        uygun = i;
                    }
            }
        }
    }
}

```



```

                if ((altAlan.getXI()-bulunulanAltAlan.getXI()) >= -4){
                    enUygun = uygun;
                }
            }
            break;
        case kuzey :
            if (((bulunulanAltAlan.getXI() - altAlan.getXI()) == 0 )
&&((altAlan.getYI()-bulunulanAltAlan.getYI()) < 0)){
                uygun = i;
                if ((altAlan.getYI()-bulunulanAltAlan.getYI()) >= -4){
                    enUygun = uygun;
                }
            }
            break;
        case guney :
            if (((bulunulanAltAlan.getXI() - altAlan.getXI()) == 0 )
&&((altAlan.getYI()-bulunulanAltAlan.getYI()) > 0)){
                uygun = i;
                if ((altAlan.getYI()-bulunulanAltAlan.getYI()) <= 4){
                    enUygun = uygun;
                }
            }
            break;
        default :
            break;
    }
}
return uygun;
}
}

```

MayinTarmaRobotu:

```
package uygulama;
```

```
import mayintarlasi.ArastirmaSonucu;
import mayintarlasi.MayinTarlasi;
import gui.Raporlayici;
```

```
public abstract class MayinTaramaRobotu {
```

```

private Pozisyon pozisyon = null;
private Pozisyon birOncekiPozisyon = null;

private String sembol = "";
private AltAlan icindeHareketEdilenAltAlan = null;
private AltAlan incelenenAltAlan = null;
private AltAlan icindeHareketEdilecekBirSonrakiAltAlan;
private boolean incelemeninOncelikleYogunlasacagiYonlerinSirasi[] = {true,true,true,true};

protected final static int dogu=0;
protected final static int bati=1;
protected final static int kuzey=2;
protected final static int guney=3;
protected final static int yonsayisi = 4;

@SuppressWarnings("unused")
private Raporlayici raporlayici = null;
private MayinTarlasi mayinTarlasi = null;

public MayinTaramaRobotu(Raporlayici aRaporlayici,MayinTarlasi aMayinTarlasi) {
    super();
    raporlayici = aRaporlayici;
    mayinTarlasi = aMayinTarlasi;
    konumunuBelirle(0,aMayinTarlasi.getBaslangic()-1);
}
private void incelemeninOncelikleYogunlasacagiYonleriBelirle(){
    String birSonrakiAltAlaninYonu=
icindeHareketEdilenAltAlan.verilenAltAlaninBulunduguYon(icindeHareketEdilecekBirSonrakiAltAlan);

    if (birSonrakiAltAlaninYonu.contains("dogu")){
        incelemeninOncelikleYogunlasacagiYonlerinSirasi[0]=false;
    }
    if (birSonrakiAltAlaninYonu.contains("bati")){
        incelemeninOncelikleYogunlasacagiYonlerinSirasi[1]=false;
    }
    if (birSonrakiAltAlaninYonu.contains("kuzey")){
        incelemeninOncelikleYogunlasacagiYonlerinSirasi[2]=false;
    }
    if (birSonrakiAltAlaninYonu.contains("guney")){
        incelemeninOncelikleYogunlasacagiYonlerinSirasi[3]=false;
    }
}
}

```

```

protected boolean altAlandaPotansiyelDengesiKosullariGozlendilse(){
    boolean sonuc = false;

    double agirlikMerkeziPozisyonu = (1.0*(incelenenAltAlan.getX1() +
incelenenAltAlan.getX2()))/2;
    double dengeAgirligi = 0.0;

    for (int i = incelenenAltAlan.getX1(); i<=incelenenAltAlan.getX2();i++){
        double merkezeUzakligi = i - agirlikMerkeziPozisyonu;
        double kolonAgirligi = 0.0;
        for (int j = incelenenAltAlan.getY1(); j<=incelenenAltAlan.getY2();j++){
            kolonAgirligi = kolonAgirligi + bolgeDegerlendirme(new Pozisyon(i,j));
        }
        dengeAgirligi = dengeAgirligi + (merkezeUzakligi*kolonAgirligi);
    }
    if ((dengeAgirligi > -1.0)&&(dengeAgirligi < 1.0))
        sonuc = true;
    return sonuc;
}

protected String bulunulanYon(){
    String yon = new String("");

    if (getPozisyon().getPosX()-getIncelenenAltAlan().getX1() > getIncelenenAltAlan().getX2() -
getPozisyon().getPosX())
        yon = yon + "dogu";
    if (getPozisyon().getPosX()-getIncelenenAltAlan().getX1() < getIncelenenAltAlan().getX2() -
getPozisyon().getPosX())
        yon = yon + "bati";
    if (getPozisyon().getPosY()-getIncelenenAltAlan().getY1() < getIncelenenAltAlan().getY2() -
getPozisyon().getPosY())
        yon = yon + "kuzey";
    if (getPozisyon().getPosY()-getIncelenenAltAlan().getY1() > getIncelenenAltAlan().getY2() -
getPozisyon().getPosY())
        yon = yon + "guney";

    return yon;
}
protected boolean incelenenAltAlanGenisleyebilirse(String genislemeYonu){
    boolean sonuc = false;

    if (genislemeYonu.contains("dogu")){

```

```

        if (getMayinTarlasi().icinde(new
Pozisyon(getIncelenenAltAlan().getX2()+1,getIncelenenAltAlan().getY2()))
            sonuc = true;
        }
        else if (genislemeYonu.contains("bati")){
            if (getMayinTarlasi().icinde(new Pozisyon(getIncelenenAltAlan().getX1()-
1,getIncelenenAltAlan().getY2()))
                sonuc = true;
            }
        else if (genislemeYonu.contains("kuzey")){
            if (getMayinTarlasi().icinde(new Pozisyon(getIncelenenAltAlan().getY1()-
1,getIncelenenAltAlan().getX2()))
                sonuc = true;
            }
        else if (genislemeYonu.contains("guney")){
            if (getMayinTarlasi().icinde(new
Pozisyon(getIncelenenAltAlan().getY2()+1,getIncelenenAltAlan().getX1()))
                sonuc = true;
            }
        }
        return sonuc;
    }

```

```

protected void incelenenAltAlaniVerilenYondeGenislet(String yon){
    if (yon.equals("dogu")){
        getIncelenenAltAlan().setX2(getIncelenenAltAlan().getX2()+1);
    }
    if (yon.equals("bati")){
        getIncelenenAltAlan().setX1(getIncelenenAltAlan().getX1()-1);
    }
    if (yon.equals("kuzey")){
        getIncelenenAltAlan().setY1(getIncelenenAltAlan().getY1()-1);
    }
    if (yon.equals("guney")){
        getIncelenenAltAlan().setY2(getIncelenenAltAlan().getY2()+1);
    }
}

```

```

private void incelenenAltAlaniVerilenYondeDaralt(String yon){
    if (yon.equals("dogu")){
        getIncelenenAltAlan().setX2(getIncelenenAltAlan().getX2()-1);
    }
    if (yon.equals("bati")){

```

```

        getIncelenenAltAlan().setX1(getIncelenenAltAlan().getX1()+1);
    }
    if (yon.equals("kuzey")){
        getIncelenenAltAlan().setY1(getIncelenenAltAlan().getY1()+1);
    }
    if (yon.equals("guney")){
        getIncelenenAltAlan().setY2(getIncelenenAltAlan().getY2()-1);
    }
}

protected int bolgeDegerlendirme(Pozisyon sinananPozisyon){
    int puanSkalasi[] = puanSkalasiBelirle();
    return puanSkalasi[getMayinTarlasi().getArastirmaAt(sinananPozisyon.getPosX(),
sinananPozisyon.getPosY()).getArastirmaSonucu()];
}
protected int gidilecekPozisyonPotansiyeli(int yon){
    int potansiyel = 0;
    Pozisyon sinananPozisyon = sinanacakPozisyon(yon);
    if (getIncelenenAltAlan() == null)
        setIncelenenAltAlan(new AltAlan(icindeHareketEdilenAltAlan()));
    if (getIncelenenAltAlan().icinde(sinananPozisyon))
        if (getMayinTarlasi().getArastirmaAt(sinananPozisyon.getPosX(),
sinananPozisyon.getPosY()).getArastirmaSonucu() == ArastirmaSonucu.yesil)
            potansiyel++;
    return potansiyel;
}

protected boolean altAlandaHareketTekrariniIncele(int yonPotansiyelleri[], int secilenYon){
    boolean sonuc = false;
    Pozisyon gidilecekPozisyon = sinanacakPozisyon(secilenYon);
    if (gidilecekPozisyon.esittir(getBirOncekiPozisyon())){
        setIncelenenAltAlan(getIcindeHareketEdilenAltAlan());
        switch (secilenYon){
            case 0 :
                incelenenAltAlaniVerilenYondeDaralt("dogu");

                break;
            case 1 :
                incelenenAltAlaniVerilenYondeDaralt("bati");

                break;
            case 2 :

```

```

        incelenenAltAlaniVerilenYondeDaralt("kuzey");

        break;
    case 3 :
        incelenenAltAlaniVerilenYondeDaralt("guney");

        break;
    default :
    }
}

return sonuc;
}

protected boolean altAlandaPotansiyelDengesiKosullariGozlendilse(int yonPotansiyeli[], int secilenYon){
    boolean sonuc = false;
    for (int i = 0; i< yonsayisi;i++)
        if ((i!=secilenYon) && (yonPotansiyeli[i] == yonPotansiyeli[secilenYon]))
            sonuc = true;
    return sonuc;
}

protected void altAlandaPotansiyelDengesiKosullarinilncele(int yonPotansiyeli[], int secilenYon){

    if (altAlandaPotansiyelDengesiKosullariGozlendilse(yonPotansiyeli, secilenYon)){

        switch (secilenYon){
        case 0 :
            incelenenAltAlaniVerilenYondeDaralt("bati");

            break;
        case 1 :
            incelenenAltAlaniVerilenYondeDaralt("dogu");

            break;
        case 2 :
            incelenenAltAlaniVerilenYondeDaralt("guney");

            break;
        case 3 :
            incelenenAltAlaniVerilenYondeDaralt("kuzey");

            break;
        }
    }
}

```

```

        default :
        }
    }
}

protected void incelenenAltAlaniDuzenle(int yonPotansiyelleri[], int secilenYon){
    if (!erisilmemisBolgeVar(getIncelenenAltAlan()))
        getIncelenenAltAlan().alanSinirlariniEsitle(icindeHareketEdilenAltAlan);
// altAlandaHareketTekrariniIncele(yonPotansiyelleri, secilenYon);
altAlandaPotansiyelDengesiKosullariniIncele(yonPotansiyelleri, secilenYon);
}

public void hareketeHazirla(AltAlan tarananAltAlan, AltAlan taranacakBirSonrakiAltAlan){
    icindeHareketEdilenAltAlan = tarananAltAlan;
    icindeHareketEdilecekBirSonrakiAltAlan = taranacakBirSonrakiAltAlan;
    incelemeninOncelikleYogunlasacagiYonleriBelirle();
}

protected Pozisyon yonPozisyonFarki(int yon){

    Pozisyon fark = new Pozisyon(0,0);

    switch (yon){
    case MetalTaramaRobotu.dogu :
        fark.setPosX(1);
        fark.setPosY(0);
        break;
    case MetalTaramaRobotu.bati :
        fark.setPosX(-1);
        fark.setPosY(0);
        break;
    case MetalTaramaRobotu.kuzey :
        fark.setPosX(0);
        fark.setPosY(-1);
        break;
    case MetalTaramaRobotu.guney :
        fark.setPosX(0);
        fark.setPosY(1);
        break;
    default :

```

```

    }
    return fark;
}

protected void erisilemezBolgeleriBelirle(AltAlan tarananAltAlan){
    try{
        for (int i=tarananAltAlan.getX1(); i <= tarananAltAlan.getX2();i++)
            for (int j=tarananAltAlan.getY1(); j <= tarananAltAlan.getY2();j++)
                if (getMayinTarlasi().getArastirmaAt(i,j).getArastirmaSonucu() ==
ArastirmaSonucu.bilinmiyor){
                    int kapaliYonSayisi = yonsayisi;
                    for (int yon = 0; (kapaliYonSayisi == yonsayisi) && (yon <
yonsayisi); yon++){
                        Pozisyon fark = yonPozisyonFarki(yon);
                        Pozisyon incelenenKomsuPozisyon = new Pozisyon (i
+fark.getPosX(),
                            j +fark.getPosY());
                            if (getMayinTarlasi().icinde(incelenenKomsuPozisyon )){
                                if
(getMayinTarlasi().getArastirmaAt(incelenenKomsuPozisyon .getPosX(),incelenenKomsuPozisyon
.getPosY()).getArastirmaSonucu() != ArastirmaSonucu.kirmizi)
                                    kapaliYonSayisi--;
                                }
                                else{
                                    kapaliYonSayisi--;
                                }
                            }
                        if (kapaliYonSayisi == yonsayisi){
                            getMayinTarlasi().getArastirmaAt(i,j).setArastirmaSonucu(ArastirmaSonucu.erisilemez);
                            getRaporlayici().raporla(" X: "+ i + " Y: "+ j + "
pozisyonu gri olarak isaretlendi.");
                        }
                    }
                }
            }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.print(true);
        }
    }
}

```



```

protected Pozisyon sinanacakPozisyon(int yon){
    Pozisyon sinanan = new Pozisyon(getPozisyon());

    Pozisyon fark = yonPozisyonFarki(yon);

    sinanan.setPosX(getPozisyon().getPosX()+fark.getPosX());
    sinanan.setPosY(getPozisyon().getPosY()+fark.getPosY());

    return sinanan;
}

public Pozisyon getPozisyon() {
    return pozisyon;
}

public void konumBelirle(Pozisyon yeniPozisyon){
    if (getBirOncekiPozisyon() == null)
        setBirOncekiPozisyon(new Pozisyon(getPozisyon()));
    else
        getBirOncekiPozisyon().belirle(getPozisyon());
    pozisyon.belirle(yeniPozisyon);
    getRaporlayici().raporla(getSembol()+ " Robotu" + " X : " + getPozisyon().getPosX()+ " Y : " +
getPozisyon().getPosY()+ " konumuna ilerledi");
}

public void konumunuBelirle(int baslangicX, int baslangicY){
    if (pozisyon == null){
        pozisyon = new Pozisyon(baslangicX,baslangicY);
    }
    else
        pozisyon.belirle(baslangicX, baslangicY);
}

public void setMayinTarlasi(MayinTarlasi mayinTarlasi) {
    this.mayinTarlasi = mayinTarlasi;
}

```

```

public Raporlayici getRaporlayici() {
    return raporlayici;
}

public void setRaporlayici(Raporlayici raporlayici) {
    this.raporlayici = raporlayici;
}

protected abstract int [] puanSkalasiniBelirle();
protected abstract boolean erisilmemisBolgeVar(AltAlan tarananAltAlan);
public abstract boolean altAlandakiTaramaDevamEdiyor(AltAlan tarananAltAlan,AltAlan sonrakiAltAlan);
public abstract boolean sonrakiAltAlanaGecisDevamEdiyor(AltAlan tarananAltAlan,AltAlan
sonrakiAltAlan);

public abstract void altAlaniTaramakIcinllerlet(AltAlan tarananAltAlan);
public abstract void sonrakiAltAlanaDogrullerlet(AltAlan sonrakiAltAlan, Pozisyon ozelKonum);

public MayinTarlasi getMayinTarlasi() {
    return mayinTarlasi;
}

public String getSembol() {
    return sembol;
}

public void setSembol(String sembol) {
    this.sembol = sembol;
}
public AltAlan getIncelenenAltAlan() {
    return incelenenAltAlan;
}
public void setIncelenenAltAlan(AltAlan incelenenAltAlan) {
    this.incelenenAltAlan = incelenenAltAlan;
}
public AltAlan getIcindeHareketEdilenAltAlan() {
    return icindeHareketEdilenAltAlan;
}
public void setIcindeHareketEdilenAltAlan(AltAlan icindeHareketEdilenAltAlan) {
    this.icindeHareketEdilenAltAlan = icindeHareketEdilenAltAlan;
}
public Pozisyon getBirOncekiPozisyon() {
    return birOncekiPozisyon;
}

```

```

    }
    public void setBirOncekiPozisyon(Pozisyon birOncekiPozisyon) {
        this.birOncekiPozisyon = birOncekiPozisyon;
    }
}

```

Pozisyon:

package uygulama;

```

public class Pozisyon {
    private int posX;
    private int posY;

    public Pozisyon(int posX, int posY) {
        super();
        belirle(posX,posY);
    }
    public Pozisyon(Pozisyon kopya) {
        super();
        belirle(kopya);
    }

    public void belirle(Pozisyon yeniPozisyon){
        setPosX(yeniPozisyon.getPosX());
        setPosY(yeniPozisyon.getPosY());
    }

    public void belirle(int posX, int posY){
        setPosX(posX);
        setPosY(posY);
    }

    public int getPosX() {
        return posX;
    }

    public void setPosX(int posX) {
        this.posX = posX;
    }
}

```

```

public int getPosY() {
    return posY;
}

public void setPosY(int posY) {
    this.posY = posY;
}

public boolean esittir (Pozisyon sinananPozisyon){
    boolean sonuc = false;
    if (sinananPozisyon != null)
        if ((sinananPozisyon.getPosX() == getPosX()) && (sinananPozisyon.getPosY() ==
getPosY()))
            sonuc = true;
    return sonuc;
}
}

```