





# Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

## Otonom Araçla Genetik Algoritma Kullanılarak Haritalama ve Lokasyon

 Merve Nur DEMİR<sup>a,\*</sup>,  Yusuf ALTUN<sup>b</sup>

<sup>a</sup> Elektrik-Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

<sup>b</sup> Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

\* Sorumlu yazarın e-posta adresi: demirmnur@gmail.com

DOI: 10.29130/dubited.640063

### ÖZET

Teknolojik gelişmeler ve bu zamana kadar biriken bilgilerin ışığında otonom sistemlerde muazzam bir ilerleme kaydedilmiştir. Bu sayede otonom sistemler çarpışmadan kaçınma, trafik işareti tespiti, haritalama vb. sayısız akıllı işlevleri gerçekleştirebilmektedir. Gerçek zamanlı otonom araçların en zorlu problemi aracın kendi kendine haritalandırma ve lokasyon işlemlerini yapabilmesidir. Genetik Algoritma (GA) kullanarak optimize edilmiş lokasyon uygulaması ile otonom araçlar için sürüş güvenliğinin artması beklenmektedir. Bu çalışma da lazer tabanlı bir lokalizasyon ve haritalama tekniğinin üzerine odaklanılmıştır. Gerçekleştirilen sistemde sanal bir test ortamı kurulmuş ve bir otonom araç üzerinde denemeler yapılmıştır. Çalışma kapsamında sanal makineler oluşturularak üzerlerine Linux işletim sistemi kurulmuştur. Sonra bu sanal makinelere ROS ortamında TurtleBot3 kurulmuş ve iç mekân lokalizasyonu yapılarak bir harita elde edilmiştir. Bu harita genetik algoritma ile en kısa mesafelerin bulunmasını sağlamak için kullanılmaktadır. Gözlemler neticesinde simülasyon ortamındaki robot yüksek başarımla istenilen konuma gidebildiği sonucuna ulaşılmıştır.

**Anahtar Kelimeler:** Genetik Algoritma, Haritalama, Lokasyon, Otonom Araç, SLAM

## Mapping and Location Using Genetic Algorithm with Autonomous Vehicle

### ABSTRACT

Significant progress has been made in autonomous systems in the light of technological advances and accumulated knowledge to date. In this way, autonomous systems, collision avoidance, traffic sign detection, mapping and so on. It can perform numerous intelligent functions. The most challenging problem of real-time autonomous vehicles is that the vehicle can perform self-mapping and location operations. Optimized location application using Genetic Algorithm (GA) is expected to increase driving safety for autonomous vehicles. This study focuses on a laser based localization and mapping technique. In the system, a virtual test environment was established and experiments were performed on an autonomous vehicle. Within the scope of the study, virtual machines were created and Linux operating system was installed on them. Then, TurtleBot3 was installed in these virtual machines in ROS environment and a map was obtained by localizing the interior. This map is used to find the shortest distances by genetic algorithm. As a result of the observations, it was concluded that the robot in the simulation environment can go to the desired position with high performance.

**Keywords:** Genetic Algorithm, Mapping, Location, Autonomous Vehicle, SLAM.

# I. GİRİŞ

Teknolojik gelişmeler ve bu zamana kadar biriken bilgilerin ışığında otonom hareket edebilen mekanizmalar dünya çapında yüzlerce alanda kullanılmaya başlamıştır. Robotik sistemlerin geliştirilmesinde çevresine daha hızlı ve daha kararlı yanıtlar verebilen mekanizmalar üretebilmek ve bunun için de insanların sahip olduğu algılayıcılara benzer sensörler ile donatılmış robotların çalışma hızının; gerçek zamanlı ve insanın algılama ve yanıt verme hızına yakın olması bilim insanlarının en büyük hedeflerinden biridir. Mevcut çalışmalar ve gelecekteki eğilimler bize, son zamanlardaki destek sistemlerinin gelecekte tamamen otonom bir araca yönelik sürücü yardım sistemi olarak önemli rol oynayacağını göstermektedir [1].

Son on yılda, Gelişmiş Sürücü Yardım Sistemi (GSYS) [2] ve otonom sürüş [3], [4] konusunda muazzam bir ilerleme kaydedilmiştir. Bu tür sistemler çarpışmadan kaçınma, şerit ayrılma uyarısı, trafik işareti tespiti, haritalama vb. sayısız akıllı işlevleri [5] gerçekleştirebilir. Otomasyon derecesini standartlaştırmak için, Uluslararası Otomotiv Mühendisleri Derneği (OIMD- Society Of Automotive Engineers- SAE), 2014 yılında bir sınıflandırma sistemi yayınladı ve bu sistem 2016 yılında güncellendi. SAE'nin sınıflandırması, tamamen manüelden tamamen otonom araçlara kadar değişen altı otomasyon seviyesi tanımlar. Her seviyenin kısa açıklaması aşağıda verilmiştir [6]:

- Seviye 0: Sistem uyarıları idare eder ancak araç kontrolü devam etmemiştir.
- Seviye 1: Sürücü ve sistem birlikte çalışır, örneğin adaptif hız sabitleyici ve park yardımı.
- Seviye 2: Sistem hızlanma, frenleme ve yönlendirmeyi kontrol eder, ancak sürücünün süreci izlemesi ve sistem arızalandığında kontrolü ele almaya hazır olması gerekir.
- Seviye 3: Sürücü, sürüş sırasında “gözleri kapalı” olarak adlandırılan eylem (bir metin okuma veya bir filmi izleme gibi) yapılabilir.
- Seviye 4: Artık güvenlik sorunu yok. Sürücü, yolcu koltuğuna oturabilir veya uyuyabilir.
- Seviye 5: Artık insan katılımı yok. Olası bir örnek robotik taksidir.

Bu sistemleri uygularken, makine öğrenmesi, sensör verilerinin yorumlanması ve çevreyi anlama ve anlamlandırma önemli bir rol oynar. Otonom sürüşün başlıca zorluklarından biri, çoğu otonom sürüş sisteminin ağır görsel algılamaya dayanmasıdır. Sensörlerin, örneğin yağmur, sis, ışık yoğunluğu veya eksikliği vb. çevresel değişikliklere duyarlı olmasıdır. Otonom bir sürüş sistemi kusursuz olamamaktadır. Sağlam bir otonom sürüş sistemi oluşturmak için, olası tüm çevresel senaryolar üzerinden araç performansını doğrulamak çok önemlidir.

Gerçek zamanlı otonom araçların iki önemli problemi haritalandırma ve lokasyonunu araç tarafından tanımlanmasıdır. Günümüzde sürücüler, araçları üzerinde tam kontrol sahibidirler. Sürüş, tamamen görsel bilgiye dayanan bir iş olduğu açıktır [7]. Sürücü etrafına bakarak kafasında o anki duruma göre nerede olduğunu, gitmek istediği yere nasıl gidebileceğini bilebilmektedir. Peki, otonom araç hiç bilmediği bir ortama bırakıldığında ortamın haritasını ve kendi konumunu çıkararak istenilen yere gidebilir mi? Bilimsel literatürde “Eş Zamanlı Lokalizasyon ve Haritalama” yöntemi olarak bilinen SLAM bu probleme bir çözüm önerisi olabilmektedir. 1986 yılında, Durrant-Whyte, Crowley ve Cheeseman yapmış olduğu SLAM çalışmaları olasılıksal yöntemler içeren bir çalışmadır [8].

Otonom araçlar için sağlam ve hassas bir konumlandırma ve haritalama yöntemi gerekmektedir. Konum tahminlerin sağlamak için çeşitli sensörler ve algoritmalar kullanılabilir [9]. Harita oluşturma işleminin doğruluğu, başlangıç pozisyonuna dayanmaktadır [10]. Dış mekânda otonom sürüş yapan robotlar, Küresel Konumlandırma Sisteminin (GPS) kullanarak zamanlama sinyallerine göre konumlarını boyalam ve enlem açısından hesaplayabilir. Fakat iç mekânda bulunan robotlar GPS sinyallerini alamaz ve mesafe sensörlerinin yardımına ihtiyaç duyarlar.

Haritalama yöntemleri içinde bazı SLAM algoritmaları geliştirilmiştir. Örneğin, Myung-Jin Jung ve arkadaşları [11] 'de parçacık ağırlığı bazlı doluluk izgara haritasını kullanan bir hızlı-SLAM yöntemi sunmuştur. Performansı, Monte Carlo Lokasyon algoritması veri ilişkilendirmeleriyle birleştirilen

parçacık başına maksimum olabilirlik olasılığı ile geliştirilmiştir. Yusuke Misono [12], dış mekân mobil robotları için Lazer Menzil Bulucu kullanan göreceli engel gözlem profillerini kullanarak SLAM uygulamıştır. Küresel bir harita, göreceli gözlemler kullanılarak adım adım inşa edilir ve aynı zamanda genişletilmiş Kalman filtresiyle robotun bulunduğu yerin sınırlı bir tahminini hesaplar. Momotaz Begum [13], SLAM için bulanık mantık ve GA bütünleştiren bir yöntem geliştirmiştir.

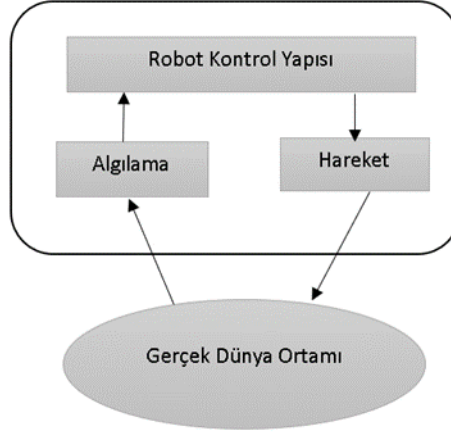
GA, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar ve sayısal optimizasyon yöntemlerinde sıkça kullanılmaktadır. Diğer klasik yöntemlere göre daha iyi sonuç vermektedir. Sezgisel bir yöntem olması sebebiyle, bulunan çözüm en iyi çözüm olmayabilir. Ancak, geleneksel yöntemler ile çok uzun sürebilecek problem çözümlerinin, gözle görülür bir şekilde kısa bir sürede elde edilmesinde faydalı olmaktadır [14]. Son zamanlarda, karmaşık ve oldukça doğrusal olmayan çok amaçlı optimizasyon problemlerini başarılı bir şekilde çözmek için uygulama alanlarında evrimsel yöntemler uygulanmaktadır. GA, bugüne kadar robot lokalizasyonu ya da haritalama problemlerine uygulanan en yaygın evrimsel algoritmalar olmuştur [15]. Bir önceki çalışmalarını geliştirme amacıyla temel fikri, örnek güçsüz/yetersiz kromozomların sayısını azaltmak ve hesaplama maliyetini düşürmek için harita parçacığını / kromozomlarını sıkıştırmak olan Rao-Blackwellized GA Filtreli SLAM tekniği önerilmiştir [16].

Bu çalışmada lazer tabanlı bir lokalizasyon ve haritalama tekniğinin optimizasyonu üzerine odaklanılacaktır. Amaç otonom araçlar için haritalama algoritmalarının incelenmesi ve daha etkili bir algoritmanın geliştirilmesi; gerçekleştirilen sistemin simülasyon ortamında sanal bir otonom araç üzerinde test edilmesidir. Yapılacak olan bu haritalama uygulaması ile otonom araçlar için sürüş güvenliğini ve verimliliğini artırması beklenmektedir. Çalışma kapsamında bilgisayarda işletim sistemi Linux olan sanal makineler kurulmuştur. Sonra ki adımda bu sanal makinelere ROS işletim sistemi ve TurtleBot3 kurulmuştur. Gazebo simülasyon ara yüzü ile iç mekân lokalizasyonu yapılarak bir harita elde edilmiştir ve eş zamanlı olarak Rviz ortamında izlenmiştir. Elde edilen bu harita GA ile en kısa mesafelerin bulunmasını sağlamak için kullanılmaktadır. Gözlemler neticesinde simülasyon ortamındaki robot yüksek başarımla istenilen konuma gidebildiği sonucuna ulaşılmıştır.

## **II. LOKASYON VE HARİTALAMA**

### **A. OTONOM ARAÇLAR**

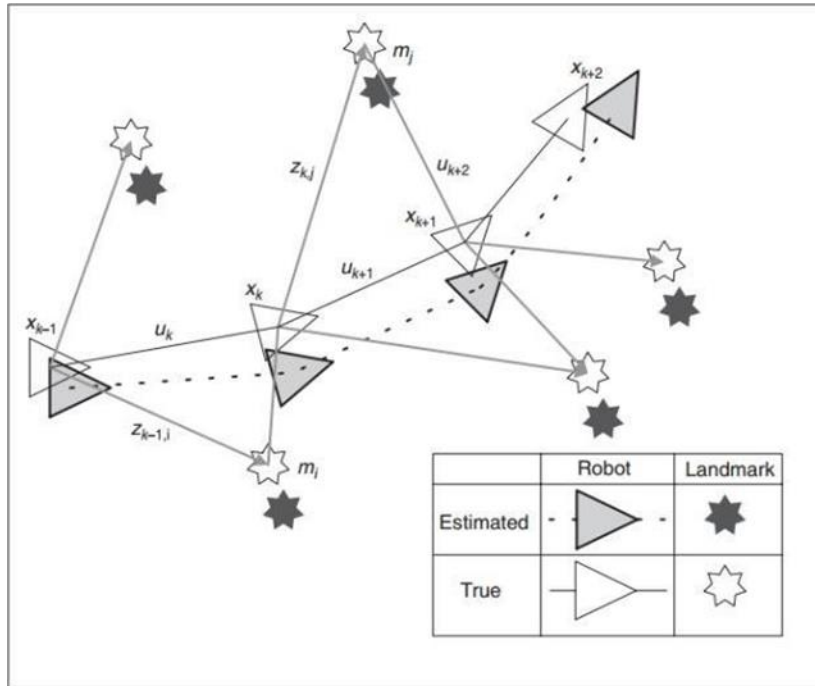
Otonom sürüş konusundaki ilk deneyler 1920'lerden başladı [11] ve ilk prototipler 1950'lerden itibaren ortaya çıkmaya başladı. 1997 yılında Japonya'da Tsukuba Makine Mühendisliği Laboratuvarı ilk gerçek otonom aracı geliştirdi. Otonom araçlar fiziksel ortamlarda hareket edebilmek için çevresinden topladıkları verileri içinde bulunan hareket sistemleri, karar mekanizmaları, ortamı algılamak amacıyla algılayıcı sistemleri ve algoritmalar sayesinde yorumlayıp kendi kendine hareket eden mekanizmalardır. Bu çalışmada TurtleBot3 Waffle Pi otonom aracı sanal ortamda hazırlanmış ve simülasyonda test edilmiştir. Mobil robot sistemlerinin bileşenleri, farklı bakış açıları olan çeşitli kaynaklarda tanımlanmaktadır [17], [18]. Örnek olarak mobil robot sistem blok şeması Şekil 1'de verilmiştir.



Şekil 1. Mobil Robot sistemi blok şeması

## B. EŞ ZAMANLI LOKASYON VE HARİTALAMA

Eş Zamanlı Lokalizasyon ve Haritalama (Simultaneous Localization and Mapping-SLAM) aynı zamanda Eş Zamanlı Haritalama ve Yerelleştirme (Concurrent Mapping and Localization-CML) olarak da bilinir ve robotun bir haritadaki yerini belirlerken bir çevre haritası oluşturma zorunluluğuyla ilgilenir. SLAM, bir mobil robotun bir çevre haritası oluşturabileceği ve aynı zamanda konumunu belirlemek için bu haritayı kullanabileceği bir süreçtir. Başlangıçta hem harita hem de araç konumu bilinmemektedir, araç bilinen bir kinematik modele sahiptir ve yapay ya da doğal yerler ile doldurulan bilinmeyen ortamda hareket etmektedir. Hem robot hem de yer işareti konumlarının eşzamanlı tahmini gerekir. Bunu yapabilmek için, araç, yer işaretleri ile aracın kendisi arasındaki göreceli konumun ölçümlerini alabilen duyuşsal bir sistemle donatılmalıdır [19].



Şekil 2. Temel SLAM çalışma prensibi

k anı için nicelikler aşağıdaki gibi tanımlanabilir:

- $x_k$  : aracın konumunu ve yönünü tanımlayan durum vektörü
- $u_k$  : aracı k zamanında  $x_k$  durumuna getirmek için k – 1 anında uygulanan kontrol vektör
- $m_i$  : gerçek konumu zamanla değişmeyen kabul edilen yer işaretinin konumunu tanımlayan bir vektör
- $z_k$  : k anında yer işaretinin bulunduğu yerin araçtan yapılmış bir gözlem. Herhangi bir zamanda birden fazla yer işareti gözlemi olduğunda veya belirli bir yer işareti olmadığında, gözlem basitçe  $z_k$  olarak ifade edilecektir.

Aracın lokasyon geçmişi;

$$X_{0:k} = \{x_0, x_1, \dots, x_k\} = \{X_{0:k-1}, x_k\} \quad (1)$$

Kontrol girişlerinin geçmişi;

$$U_{0:k} = \{u_0, u_1, \dots, u_k\} = \{U_{0:k-1}, u_k\} \quad (2)$$

Tüm yerler işaretçilerin kümesi;

$$m = \{m_1, m_2, \dots, m_n\} \quad (3)$$

Tüm yer işaretçilerinin gözlem kümesi;

$$Z_{0:k} = \{z_1, z_2, \dots, z_k\} = \{Z_{0:k-1}, z_k\} (z_0 = 0) \quad (4)$$

Olasılık formunda eş zamanlı lokalizasyon ve harita oluşturma (SLAM) problemi olasılık dağılımını tüm k anları için şu şekilde ifade edebiliriz;

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (5)$$

Bu olasılık dağılımı, kaydedilen gözlemler ve aracın ilk durumu ile birlikte k anına kadar ve k anında dahil olmak üzere, kaydedilen gözlemler ve kontrol girdileri göz önüne alındığında, yer işareti konumlarının ve araç durumunun arka eklem yoğunluğunu tarif eder. Genel olarak, SLAM problemine tekrarlamalı bir çözüm istenmektedir. Tüm k anları için SLAM problemi olasılık dağılımını ifade eden denklem (5) olasılık dağılımı için k-1 anında bir tahminle başlayarak, bir kontrol  $u_k$  ve gözlem  $z_k$ 'yi izleyen arka eklem Bayes teoremi kullanılarak hesaplanır. Bu hesaplama, sırasıyla kontrol girişi ve gözlem etkisini açıklayan bir durum geçiş modeli ve gözlem modelinin tanımlanmasını gerektirir.

## B. 1. SLAM Teknikleri

Birkaç araştırmacı SLAM'da çalışmış ve en sık kullanılan sensörler lazer tabanlı, sonar tabanlı ve görsel tabanlı sistemlerde sınıflandırılabilir olduğunu ifade etmişlerdir. Robot durumu bilgisini ve dış dünyayı, pusulalar, kızılötesi teknoloji ve Küresel Konumlandırma Sistemi KKS (GPS) gibi daha iyi algılamak için ek duyuusal kaynaklar kullanılabilir olduğunu belirtmişlerdir [20]. Bununla birlikte, tüm bu sensörler, genellikle ölçüm gürültüsü olarak adlandırılan belirli hatalar taşırlar. Ayrıca çevrede gezerken gerekli olan çeşitli sınırlamalara sahiptirler; örneğin, ışık ve ses duvarlara nüfuz edemez.

- *Lazer sınıflandırma sistemleri*; hassas etkin sensörlerdir. En bilinen şekli ile uçuş prensibi sırasında nesneye doğru dar bir ışın içinde bir lazer darbesi göndererek ve darbenin hedeften yansıtılması ve gönderene geri gönderilmesi için geçen süreyi ölçerek çalışır.
- *Sonar tabanlı sistemler*; hızlıdır ve görüşe benzer miktarda bilgi içerirler ancak görünüm verisi eksikliğinde olan ölçümler ve tanıma kapasiteleri vardır. Bununla birlikte, kilometre

sayacı gibi atalet sensörlerine bağımlılığı, küçük bir hatanın sonraki konum tahminleri üzerinde büyük etkileri olabileceği anlamına gelmektedir [21].

- *Görüş sistemleri*; pasif, uzun menzilli ve yüksek çözünürlüğe sahiptirler, ancak hesaplama maliyeti oldukça yüksektir. İyi görsel özelliklerin elde edilmesi ve eşleştirilmesi daha zordur. Görüş sistemi, 3B yapısını, özellik konumunu ve robot pozunu, örneğin stereo kamera çiftleri veya hareket kurtarma işleminden yapılmış monoküler kameralar aracılığıyla tahmin etmek için kullanılır.

Robotik harita oluşturma 25 yıl öncesine kadar izlenebilir. 1990'lardan bu yana olasılıklı yaklaşımların yani, Kalman Filtreleri (KF), Parçacık Filtreleri (PF) ve Beklenti Maksimizasyonu (BM) SLAM'da baskın hale gelmiştir. Bu üç teknik tekrarlamalı Bayes kuralının matematiksel türevleridir. Bu olasılıksal teknik popülaritesinin ana nedeni, robot haritalamanın belirsizlik ve sensör gürültüsü ile karakterize edilmesi ve olasılık algoritmaları, farklı gürültü kaynaklarını ve bunların ölçümler üzerindeki etkilerini açıkça modelleyerek sorunu çözmektedir [20].

**Tablo 1.** SLAM çerçevesine uygulanan filtreleme yaklaşımlarının avantaj ve dezavantajlarının listesi [20]

Avantajlar	Dezavantajlar
<b>Kalman Filtresi/Genişletilmiş Kalman Filtresi (KF/GKF) [21]-[23]</b>	
<ul style="list-style-type: none"> <li>- Yüksek yakınsama</li> <li>- Belirsizlikle başa çıkmak</li> </ul>	<ul style="list-style-type: none"> <li>- Gauss varsayımı</li> <li>- Yüksek boyutlu haritalarda yavaş</li> </ul>
<b>Sıkıştırılmış Genişletilmiş KF (SGKF) [24]</b>	
<ul style="list-style-type: none"> <li>- Belirsizliği azaltır</li> <li>- Hafıza kullanımının azaltılması</li> <li>- Geniş alanları ele al</li> <li>- Harita tutarlılığını arttırmak</li> </ul>	<ul style="list-style-type: none"> <li>- Çok sağlam özellikler gerektirir</li> <li>- Veri birliği sorunu</li> <li>- Çoklu harita birleştirme gerektirir</li> </ul>
<b>Bilgi Filtreleme (BF) [25],[26]</b>	
<ul style="list-style-type: none"> <li>- Kararlı ve basit</li> <li>- Doğru</li> <li>- Yüksek boyutlu haritalar için hızlı</li> </ul>	<ul style="list-style-type: none"> <li>- Veri ilişkilendirme sorunu</li> <li>- Durum tahminlerini iyileştirmek gerekebilir</li> </ul>
<b>Parçacık Filtreleme (PF) [27],[28]</b>	
<ul style="list-style-type: none"> <li>- Doğrusal olmayanları ele almak</li> <li>- Gauss olmayan gürültüyü ele almak</li> </ul>	<ul style="list-style-type: none"> <li>- Karmaşıklıkta büyüme</li> </ul>
<b>Beklenti Maksimizasyonu (BM) [25],[29]</b>	
<ul style="list-style-type: none"> <li>- Harita oluşturma için en uygun</li> <li>- Veri ilişkisini çöz</li> </ul>	<ul style="list-style-type: none"> <li>- Verimsiz, maliyet artışı</li> <li>- Büyük senaryolar için kararsız</li> <li>- Sadece harita oluşturmada başarılı</li> </ul>

## C. GENETİK ALGORİTMA

Optimizasyon problemlerine çözüm önerisi sunmak isteyen araştırmacılar çoğunlukla doğayı ve canlıları taklit eden algoritmalar geliştirmişlerdir. Yapay Zekâ'nın hızla gelişen bir dalı olan Genetik Algoritma da evrimsel sürecin bilgisayar dilinde modellenmiş bir optimizasyon yöntemidir [30]. Genetik algoritmalar Darwin'in evrim teorisinden etkilenerek geliştirilmiştir. Diğer bir deyişle GA doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemidir denebilir. Bir probleme olası pek çok çözümün içerisinde en uygununu (en iyisini) bulmaya çalışan algoritmalarlardır.

GA da çözüm uzayının tamamını değil yalnızca bir kısmını tararlar. Böylece etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar. Problemi etkileyen faktörlerin fazla olduğu durumlarda sıklıkla kullanılır. GA olasılık kurallarına göre çalışır ve ne kadar iyi çalışacağı önceden bilinmemektedir. Kısıtlar değiştirilerek probleme uygun çözüm/ler aranır. GA önemli bir özelliği de muhtemel çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve yerel en iyi çözüme takılmamalarıdır. Popülasyon nesilden nesile geliştikçe kötü çözümler yok olma, iyi çözümler ise daha iyi çözümler oluşturmak için kullanılma eğilimindedirler. Diğer bir deyişle, GA en iyi çözümün hayatta kalma ilkesine bağlı olarak çözüm kümesini oluşturur.

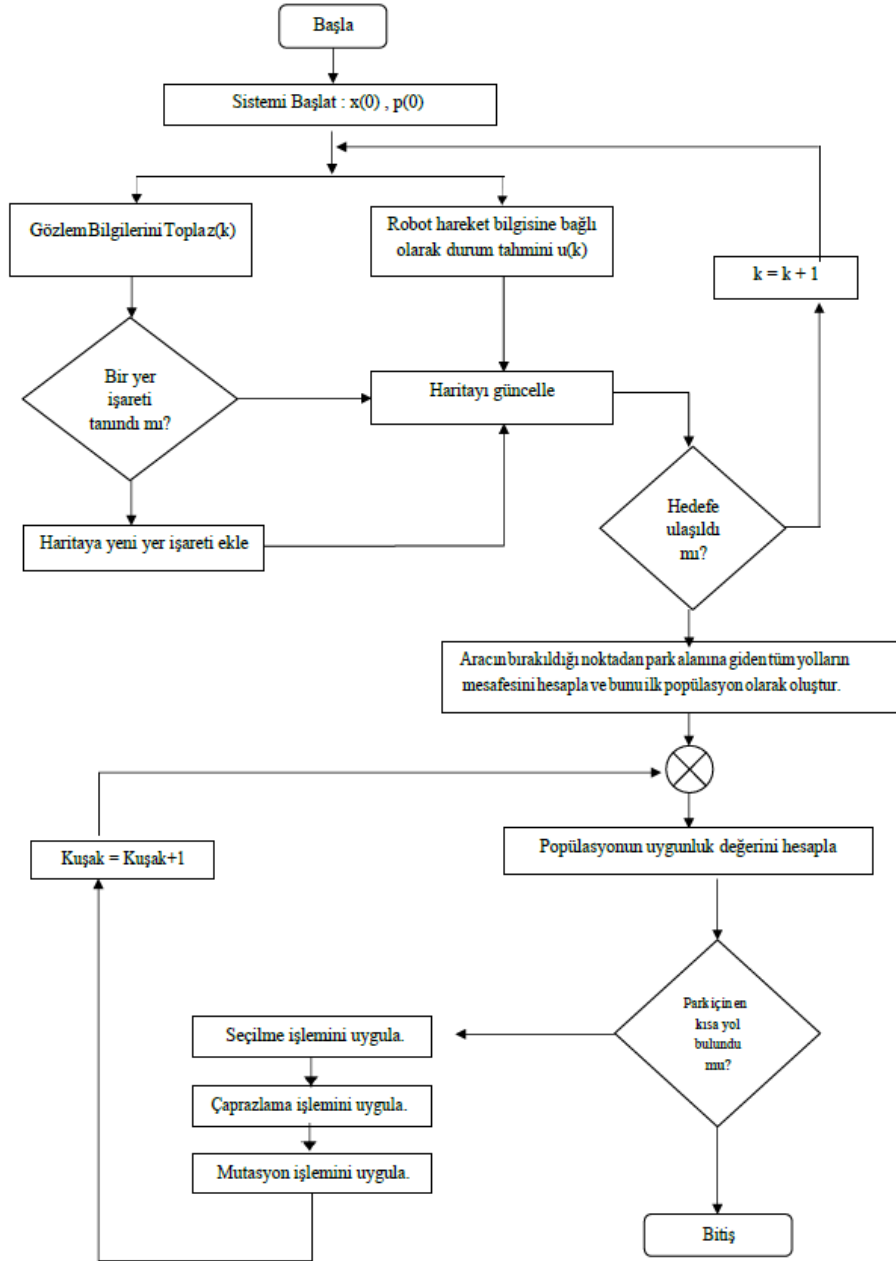
Birçok alanda uygulama imkânı ve uygulamaları olan genetik algoritmaların işleme adımları şöyle açıklanabilir;

- i. Arama uzayındaki tüm mümkün çözümler dizi olarak kodlanır.
- ii. Genellikle rastgele bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir. Başlangıç popülasyonu içindeki her çözüm bir bireyi temsil etmektedir. Popülasyonun büyüklüğü aynı zamanda çözüm kümesinin de büyüklüğünü göstermektedir.
- iii. Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
- iv. Bulunan uygunluk değerlerinin algoritmayı sonlandırma özelliğine sahip olup olmadığına bakılır. Eğer istenilen çözüme ulaşıldıysa algoritmadan çıkılır. En iyi çözüm; çözüm kümesi içerisinde ki en iyi bireydir.
- v. Eğer istenilen çözüme ulaşılmadıysa yeni popülasyonun oluşturulması gerekmektedir. Bir grup dizi belirli bir olasılık değerine göre rastgele olarak seçilip çoğalma işlemi gerçekleştirilir.
- vi. Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur. Burada amaç; seçilen çözüm kümesinden daha iyi çözümler üretebilmektir.
- vii. Çaprazlama işlemi popülasyonun çeşitliliğini artırmaktadır. Mutasyon işlemi ise çözüm kümesinin kısır döngüye girmesini engeller.
- viii. Yeni oluşturulan popülasyon içindeki her bir dizi için uygunluk değeri hesaplanır. Bulunan uygunluk değerlerinin algoritmayı sonlandırma özelliğine sahip olup olmadığına bakılır. Eğer istenilen çözüme ulaşıldıysa algoritmadan çıkılır. En iyi çözüm; çözüm kümesi içerisinde ki en iyi bireydir.
- ix. Eğer uygun çözüm bulunmadıysa en son belirlenen popülasyon ile yukarıdaki işlemler devam ettirilir.
- x. İterasyon, belirlenen kuşak sayısına ulaşıncaya kadar işlem sona erdirilir.

## III. SİMÜLASYON

Bu makalede otonom araçla genetik algoritma kullanılarak oluşturulan harita üzerinden lokasyon yapabilmek için simülasyon ortamında test yapılacaktır. Otonom araç simülasyonunda Gazebo arayüzü kullanılarak sanal bir ortam belirlenecek ve haritası çıkarılacaktır. Rviz; robotun çıkarttığı haritanın ve izlediği gezinimin görüntülenebileceği ROS paketidir. Rviz'de kullanılarak dışarıdan aktarılan robotun simülasyondaki görüntüleri ve haritalama özellikleri işlenecektir.

Otonom araç Gazebo simülasyonunda ilk önce uzaktan kumanda ile test alanında test sürüşü gerçekleştirilir. Daha sonra yine Gazebo simülasyon ortamında otonom sürüş için gerekli işlemler yaptırılır. Araç bırakılan ortamda rastgele geçtiği yolların 2-boyutlu haritasını oluşturarak hafızasına kaydeder. Eş zamanlı olarak Rviz simülasyon ortamında oluşturulan haritayı görebilmekteyiz. Daha sonra yine belirlenen bu ortamın rasgele bir noktasına bırakılan araç belirtilecek nokta ya da Mouse yardımıyla işaretlenen yere gitmesi istenecektir. Aracın bulunduğu noktadan istenen yere en kısa yoldan gitmesi için genetik algoritma kullanılacaktır. Anlatılan yöntem için program akış diyagramı Şekil 3'te verilmiştir.

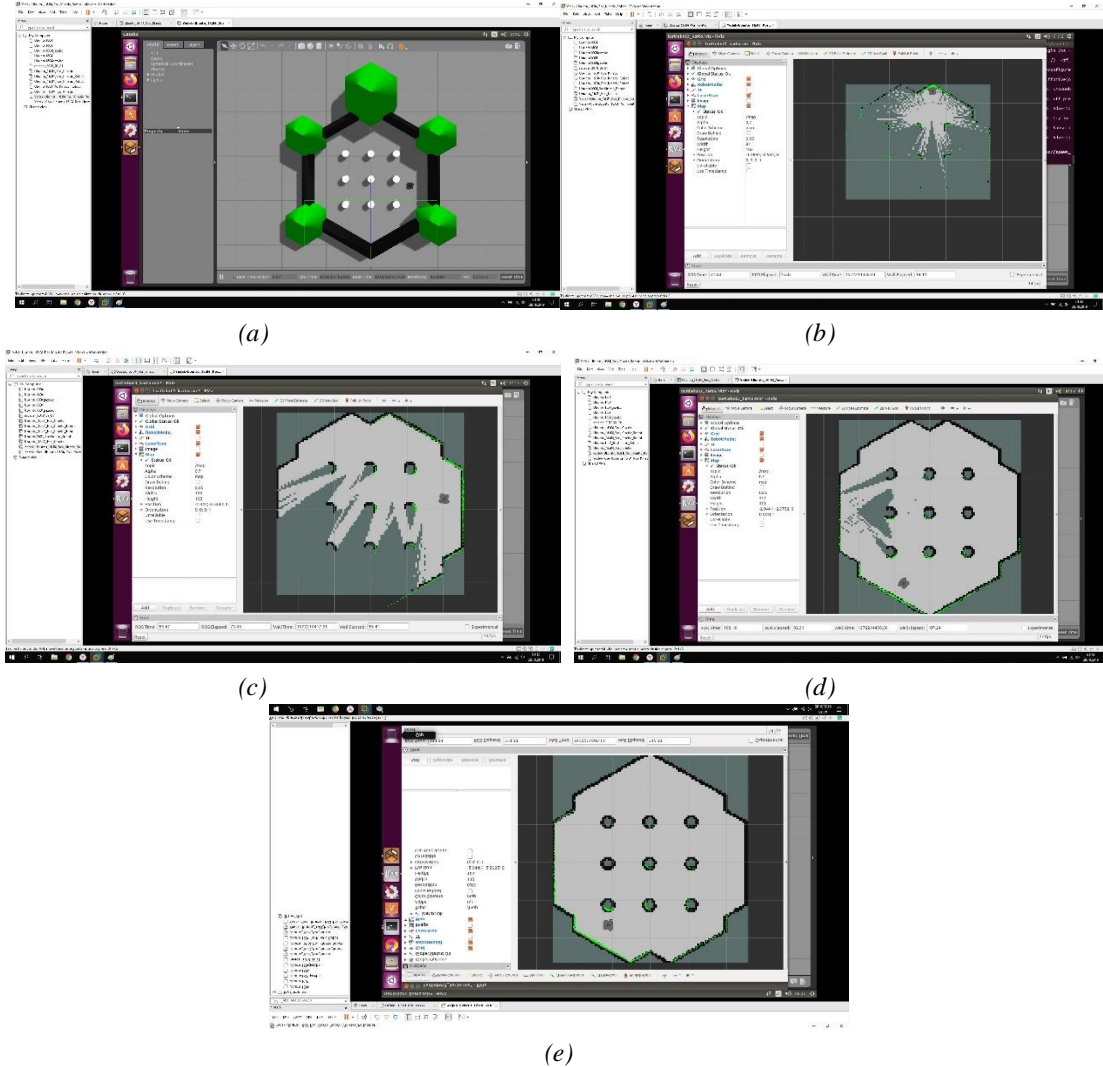


Şekil 3. Program akış diyagramı

Simülasyon deney ortamında oluşturulan sanal makineler Linux işletim sistemi (Ubuntu 16.04) ve ROS, ROS-Kinetic Paketi ve TurtleBot3 yönergeler takip edilerek kurulmuştur. Gazebo ortamında sanal



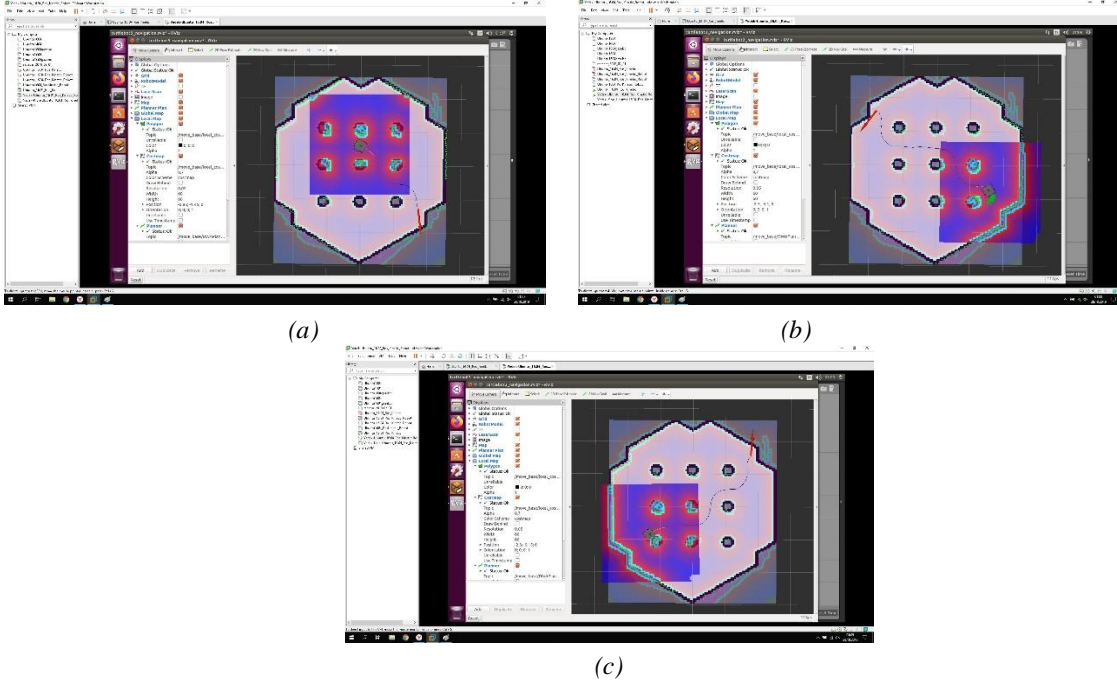
dünyalarından biri çağrılarak, araç bu ortamda otonom hareket ederek Şekil 3'te gösterilen SLAM yöntemi ile bulunduğu ortamın haritasını çıkarmaya başlar. Rviz ortamında bu işlem eş zamanlı olarak izlenebilmektedir. Araç haritalama işlemini bitirdiğinde haritayı kayıtlar. Kaydedilen harita genetik algoritma kısmında çağrılarak aracın istenilen noktaya en kısa mesafelerin bulunmasını sağlamak için kullanılmaktadır.



**Şekil 4.** (a) Gazebo ortamı ve aracın konumu. (b) Rviz ortamında aracın ilk konumu. (c) Rviz ortamında aracın haritalamaya devam etmesi. (d) Rviz ortamında aracın haritalamaya devam etmesi. (e) Rviz ortamında aracın haritalamayı bitirmesi

Şekil 4 (a)'da TurtleBot3 aracının Gazebo ortamındaki ilk konumu görülmektedir. Rviz ortamı ayrı bir terminalden çalıştırılarak araç hareketi eş zamanlı gözlenmektedir. Şekil 4 (b), (c), (d) aracın Gazebo ortamındaki otonom hareketi esnasındaki haritalama işlemleri farklı zamanlar için gösterilmektedir. Şekil 4 (e)'de Gazebo ortamının haritalamasını bitiren aracın Rviz ortamındaki görüntüsü verilmiştir. Bu aşamadan sonra elde edilen harita genetik algoritma ile birleştirilerek, sanal ortamda bulunan aracımızın istediğimiz noktaya en kısa yolu bulup belirlenen noktaya gitmesini bekliyoruz. GA'nın olasılık kurallarına göre çalıştığı unutulmamalıdır. Bu sebeple GA'nın problemlere ne kadar iyi çözüm vereceği önceden bilinmemektedir. GA en iyi çözüme en yakın çözümü sunduğunu da unutmamak gerekmektedir. Kısıtlar değiştirilerek probleme uygun çözüm/ler aranır. Bu çalışmada kullanılan kısıtların bazıları kısaca şu şekildedir;

- Nesil sayısı: 15
- İterasyon sayısı: 15
- Çaprazlama duyarlılığı: 1
- Çaprazlama olasılığı: 0.9
- Mutasyon olasılığı: 0.01
- Mutasyon tekrar sayısı: 50



**Şekil 5.** (a) Aracın bulunduğu yerden kırmızı çubukla gösterilen yere en kısa yol güzergâhı 1. (b) Aracın bulunduğu başka bir konumdan kırmızı çubukla gösterilen yere en kısa yol güzergâhı 2. (c) Aracın bulunduğu yine başka bir konumdan kırmızı çubukla gösterilen yere en kısa yol güzergâhı 3

## **IV. SONUÇ VE TARTIŞMA**

En iyi park alanını bulmak için otonom araba sistemi ve genetik algoritma uygulaması, önerilen sistemin alan kullanımı ve araba verimliliği açısından önemlidir. Genetik algoritma, önerilen sistemin çok fazla zaman kaybı olmadan en kısa yol güzergâhı için arama işlemini optimize etmesine yardımcı olur. GA tabanlı park sistemi, geleneksel park sisteminden daha güçlü bir yöntemdir ve yalnızca doğrusal kısıtlamaları olan bir işleve uygulanabilir. Önerilen yöntemin avantajı, etkili bir park yolu sağlaması ve kontrollü çalışma alanını daha geniş bir bölgeye yaymasıdır. Sunulan GA tabanlı algoritma etkili ve uygulaması kolaydır. Çözüm bilgisinin kısıtlı ikili forma kodlanmasını gerektirir ve optimal çözümü bulmak için biyolojik evrimi simüle eder. Evrimsel yinelemelerden sonra, dönüşüm vektörü elde edilebilir. Belirlenen alanda aracın otonom hareket ederek park işlemini gerçekleştirme probleminde genetik algoritmanın farklı parametreleri kullanılarak çözülmüş ve parametrelerin çözüm üzerindeki etkisi incelenmiştir. Gerçekleştirilen simülasyonlarda başlangıç popülasyonundaki birey sayısındaki artışın, sonucun bulunmasına olumlu katkıda bulunduğu gözlemlenmiştir. Bu harita genetik algoritma ile en kısa mesafelerin bulunmasını sağlamak için kullanılmaktadır. Simülasyon test ortamındaki denemeler esnasında gözlemler neticesinde robot yüksek başarımla istenilen konuma gidebildiği sonucuna ulaşılmıştır.

## **V. KAYNAKLAR**

- [1] The Road Sign Recognition Group, Road Sign Recognition Survey [Çevrimiçi]. Erişim: <http://euler.fd.cvut.cz/research/rs2/files/skoda-rs-survey.html>, Erişim Tarihi:05.09.2019.
- [2] A. Lindgren and F. Chen, “In State of the art analysis: an overview of advanced driver assistance systems (ADAS) and possible human factors issues”, *Human Factors and Economics Aspects on Safety*, 2006, ss. 38–50.
- [3] M. Aeberhard et al., “Experience, results and lessons learned from automated driving on Germany’s highways”, *IEEE Intelligent Transportation System Magazine*, c. 7, s.1, ss. 42–57, 2015.
- [4] J. Ziegler et al., “Making bertha drive—an autonomous journey on a historic route”, *IEEE Intelligent Transportation System Magazine*, c. 6, s. 2, ss. 8–20, 2014,.
- [5] M. Fu and Y. Huang, “A survey of traffic sign recognition”, *Wavelet Analysis and Pattern Recognition –ICWAPR*, 2010, ss. 119–124.
- [6] A. Driving, “Levels of driving automation are defined in new SAE international standard J3016: 2014”, SAE International: Warrendale, PA, USA, 2014.
- [7] C.Y. Fang, S. W. Chen, C. S. Fuh, “Road-Sign Detection and Tracking,” *Vehicular Technology, IEEE Transactions* , c. 52, s. 5, ss. 1329- 1341.
- [8] K.Paslıoğlu, “Otonom Mobil Robotlarda Dağılımlı Kalman Filtresi Tabanlı Eş Zamanlı Lokalizasyon ve Haritalama”, Yüksek lisans tezi, Fizik Mühendisliği, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2010.
- [9] T.Stahl, A. Wischnewski, J. Bethz and M. Lienkamp, “ROS-based localization of a race vehicle at high-speed using LIDAR”, *E3S Web of Conferences ICPEME*, 2019, c. 95.
- [10] M. Yaktubay, “A Genetic Algorithm Based Solution Approach For Vehicle Routing Problem”, Yüksek lisans tezi, Endüstri Mühendisliği, Fen Bilimleri Enstitüsü, Adana Bilim ve Teknoloji Üniversitesi, Adana, Türkiye, 2018.
- [11] M.-J. Jung, H. Myung, S.-G. Hong, D. Park, H.-K. Lee, and S. Bang, “Structured light 2D range finder for simultaneous localization and map-building (SLAM) in home environments,” in *Proc. of the 2004 International Symposium on Micro-Nanomechatronics and Human Science, and 2004 The Fourth Symposium Micro-Nanomechatronics for Information-Based Society*, 2004, ss. 371–376.
- [12] Y. Misono, Y. Goto, Y. Tarutoko, K. Kobayashi, and K. Watanabe, “Development of laser rangefinder-based SLAM algorithm for mobile robot navigation,” in *Proc. of the SICE 2007 Annual Conference*, ss. 392–396.

- [13] M. Begum, G. K. Mann, and R. G. Gosine, “Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots,” *Applied Soft Computing*, c. 8, s. 1, ss. 150 – 165, 2008.
- [14] A. Tuncer, “Otonom Araçlar için Yol Bulma Probleminin Genetik Algoritmalar ve FPGA ile Çözümü ve Gerçekleştirilmesi”, Doktora Tezi, Elektronik ve Bilgisayar Eğitimi, Fen Bilimleri Enstitüsü, Kocaeli Üniversitesi, Kocaeli, Türkiye, 2013.
- [15] D. J. Feng, S. Wijesoma and A. P. Shacklock, “Genetic Algorithmic Filter Approach to Mobile Robot Simultaneous Localization and Mapping”, IEEE 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 2007.
- [16] D. J. Feng and S. Wijesoma, “Improving Rao-Blackwellised Genetic Algorithmic Filter SLAM Through Genetic Learning”, IEEE 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 2008.
- [17] R. R. Murphy, “Introduction to AI Robotics”, MIT Press, London, 2000.
- [18] R. C. Arkin, “Behavior-Based Robotics”, John Wiley and Sons Press, England, 2002.
- [19] J. Aulinas, Y. Petillot and J. Salvi, X. Lladó, “The SLAM problem: a survey”, Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence (CCIA), October, 2008, Spain.
- [20] S. Thrun, “Robotic mapping: A survey”. *Exploring Artificial Intelligence in the New Millenium. The Morgan Kaufmann Series in Artificial Inteligence*, ISBN ISBN-10: 1558608117, 2002.
- [21] P.M. Newman and J.J. Leonard, “Consistent, Convergent, and constant-time SLAM”, International Joint Conference on Artificial Intelligence (IJCAI), Mexico,2003, ss. 1143-U1150.
- [22] P. Jensfelt, D. Kragic, J. Folkesson and M. Björkman, “A framework for vision based bearing only 3D SLAM”, in Proc. IEEE International Conference on Robotics and Automation, ICRA,2006, ss. 1944–1950.
- [23] S. Se, D. Lowe and J. Little, “Mobile robot localization and mapping with uncertainty using scaleinvariant visual landmarks”, *The International Journal of Robotics Research*, c. 21, s. 8, ss. 735–758, 2002.
- [24] J. E. Guivan and E. M. Nebot, “Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation”, *IEEE Transactions on Robotics and Automation*, c. 17, s. 3, 2001.
- [25] S. Thrun and Y. Liu, “Multi-robot SLAM with sparse extended information filers”, 11th International Symposium of Robotics Research (ISRR’03), Sienna, Italy, 2003.
- [26] S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery-Montemerlo, D. Chakrabarti and W. Burgard, “A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor

environments with mobile robots”, *IEEE Transactions on Robotics and Automation*, c. 20, s. 3, ss. 433–442, 2004.

[27] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem”, in Proc. of the National Conference on Artificial Intelligence, 2002, ss. 593–598.

[28] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, “FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges”, 18th International Joint Conference on Artificial Intelligence (IJCAI), 2003, Mexico, ss. 1151–1156.

[29] W. Burgard, D. Fox, H. Jans, C. Matenar and S. Thrun, “Sonar-based mapping with mobile robots using EM”, 16th International Conference on Machine Learning, 1999.

[30] S.Solak, “Gezgin Robotların Konom Belirleme ve Engel Sakınım Probleminin Tek Karthı Bilgisayar Sistemi Kullanılarak Çözümü”, Doktora Tezi, Elektronik ve Bilgisayar Eğitimi, Fen Bilimleri Ensütüsü, Kocaeli Üniversitesi, Kocaeli, Türkiye, 2016.