

## Rp-Lidar ve Mobil Robot Kullanılarak Eş Zamanlı Konum Belirleme ve Haritalama

Selman AKYOL<sup>1</sup>, Ayşegül UÇAR<sup>2\*</sup>

<sup>1</sup> Mekatronik Mühendisliği, Mühendislik Fakültesi, Türk Hava Kurumu Üniversitesi, İzmir, Türkiye

<sup>2</sup> Mekatronik Mühendisliği, Mühendislik Fakültesi Fırat Üniversitesi, Elazığ, Türkiye

\*<sup>1</sup> sakyol@thk.edu.tr, \*<sup>2</sup> agulucar@firat.edu.tr

(Geliş/Received: 09/05/2018;

Kabul/Accepted: 18/02/2019)

**Özet:** Tehlikeli olan veya insanlar tarafından ulaşılması zor olan yerlerde gerekli işlemleri yapabilmek için otonom robotlar kullanılır. Otonom robotların bu işi yapabilmesi için öncelikle çevresini ve kendi konumunu bilmesi gereklidir. Bilinen bir ortam ise, önceden hazırlanan harita otonom robota yüklenir ancak bilinmeyen bir ortam ise, eş zamanlı olarak hem ortamın haritasını hem de kendi konumunu belirlemesi gerekir. Bu çalışmada, Linux’da Robot İşletim Sistemi (ROS-Robot Operating System) yardımıyla, Eş Zamanlı Konum Belirleme ve Haritalama (SLAM-Simultaneous Localization and Mapping) işlemi gerçekleştirilmiştir. Bu amaç için, Kobuki firması tarafından üretilen Turtlebot mobil robotu kullanılmıştır. İlk uygulamada, Turtlebot bilgisayar USB arabirimi ile manuel olarak kontrol edilmiştir. Ortam haritasının çıkarılması ve robotun konumunu belirlemesi için lazer ile mesafe ölçümü yapan LIDAR sensörü kullanılmıştır. SLAM için istatistiksel kestirim yöntemlerinden biri olan Kalman Filtresi tabanlı Parçacık Filtresi uygulanmıştır. Bu yöntem için ROS içerisine Gmapping algoritması yüklenmiştir. İkinci uygulamada, ROS kullanılarak Turtlebot’un kontrolü otonom bir şekilde gerçekleştirilmiş, deney ortamının haritası aynı zamanda konumunu da belirlenerek çıkarılmıştır. Tüm uygulamalar başarılı bir şekilde gerçekleştirilmiştir. Elde edilen sonuçlar şekillerle verilmiştir.

**Anahtar Kelimeler:** Eş zamanlı konum belirleme ve haritalama algoritmaları, Lidar, Kalman filtresi, Parçacık filtresi, Robot işletim sistemi.

### Simultaneous Localization and Mapping for Mobile Robots Using Rp-Lidar

**Abstract:** Autonomous robots are used to perform necessary operations in areas that are dangerous or difficult to reach or accessible by people. In such a case, the robots must first know its surroundings and position in order to be able to do the necessary work. If it is a known environment, its map is loaded to autonomous robot. On the other hand, robot should simultaneously determine both the map of the environment and its own location. Simultaneous Localization and Mapping (SLAM) was performed by using Robot Operating System (ROS) in Linux. For this purpose, Turtlebot mobile robot, manufactured by Kobuki company, was used. In the first application, Turtlebot computer was manually controlled via USB interface. The Lidar sensor, which measures the laser distance, was used to extract the environment map and determine the position of the robot. Particle filter based on Kalman filter was used for SLAM. For this method, the Gmapping algorithm was loaded into the ROS. In the second application, Turtlebot was controlled autonomously using ROS and the experimental environment was also mapped by determining its position at the same time. All applications were successfully carried out. The obtained results were illustrated in figures.

**Keywords:** Simultaneous localization and mapping algorithms, Lidar, Kalman filter, Particle filter, Robot operating system.

#### 1. Giriş

Otonom robotlar endüstri dışında insanların giremeyeceği yerlere veya insanlar için çok tehlikeli olduğu düşünülen yerlere gönderilerek araştırmalar yapılmaktadır. Örneğin deprem sonrası hasarlı binalarda, karanlık ve keşfedilmemiş mağaralarda, okyanus derinliklerinde veya aktif volkanların çevresinde gerekli araştırmalar yapmak için otonom robotlar kullanılabilir. Gerekli araştırmaların yapılabilmesi ve uygulamaların gerçekleştirilebilmesi için, ortamın robot tarafından tanınması ve robotun ortam içinde nerede olduğunu bilmesi önemlidir.

Bilimsel yazında yukarıda bahsedilen bu durumlar Eş Zamanlı Konum Belirleme ve Haritalama (SLAM-Simultaneous Localization and Mapping) problemi olarak adlandırılmaktadır. SLAM problemi kısmen bilinen ya da hiç bilinmeyen bir ortamda eş zamanlı olarak haritalama ve konum belirlemenin yapılması olarak tanımlanır. SLAM hakkındaki ilk çalışma 1987 yılında Smith ve Cheesman tarafından başlamış ve günümüze kadar devam etmiştir [1-5].

\* Sorumlu yazar: agulucar@firat.edu.tr. Yazarların ORCID Numarası: <sup>1</sup> 0000-0002-1910-7671, <sup>2</sup> 0000-0002-5253-3779

Haritalama yapılırken çevre algılamasında ışık algılama ve ölçme sistemi (LIDAR-Laser Imaging Detection and Ranging) kullanılmıştır. LIDAR, çevresindeki nesnelerin konumlarını algılamak için kullanılan radar benzeri bir sensördür [6]. LIDAR, hem havadan harita oluşturmak için hem de karadan haritalama ve konum belirleme için kullanılabilir. LIDAR sensörü ile birçok alanda, yüzeylerin iki ve üç boyutlu modellerinin çıkarılması sağlanmaktadır [7-9].

SLAM problemiyle ilgili çalışmalar araştırılırken çoğunda istatistiksel çözüm yöntemleri kullanılmıştır. Bunların başında Kalman Filtresi gelmektedir [10]. Ancak Kalman Filtresi algılayıcının aldığı verileri belirli işlemlerden geçirdiği için ve gauss gürültüsüne sahip olması şartlarını koştığı için farklı yöntemlere yönelim olmuştur. [11-12]'de Beklenti Enbüyüktme (BE) yöntemi kullanılmıştır. Bu yöntem, Kalman Filtresinde olduğu gibi istatistiksel bir çözüm yöntemidir. Kalman Filtresinden farklı olarak veri gürültüsü problemini ele alarak başarılı bir sonuç ortaya koymuştur. Çok başarılı olan bu yöntemin eksikliği gerçek zamanlı olarak uygulanamaması olmuştur. Bu yüzden gerçek zamanlı uygulamalarda kullanılmamıştır. [13]'de parçacık filtreleme yöntemi önerilmiştir. Bu yöntem, Kalman Filtresi gibi aktif ve hızlı olmamasına rağmen gerçek zamanlı ve çoğu probleme çözümler getirdiği için kullanılmıştır.

Bu çalışmada, diferansiyel tekerlek modeline sahip Turtlebot mobil robotu kullanılarak üzerine yerleştirilen Robo Peak firmasının LIDAR (RP-LIDAR) sensörü sayesinde hem otonom olarak hem de manüel olarak kontrolünü sağlayıp SLAM uygulamasının yapılması amaçlanmıştır [14-15]. Bu amaca uygun deneysel düzenek oluşturulmuştur. Çalışmada hem SLAM hem de mobil robot kontrolü Linux tabanlı işletim sistemine sahip iki bilgisayarda gerçekleştirilmiştir. Bilgisayara Linux tabanlı işletim sistemine uyumlu ROS kurulmuştur [16]. Mobil robot kontrolü ROS üzerinden yapılmıştır. ROS üzerinde Parçacık Filtresi tabanlı Gmapping paketi ile SLAM uygulaması gerçekleştirilmiştir.

Bu makalenin geri kalan bölümleri aşağıdaki gibi organize edilmiştir. İkinci bölümde, parçacık filtresi ele alınarak SLAM algoritması incelenmiştir. Üçüncü bölümde, deneysel düzenekte kullanılan Turtlebot ve RP-LIDAR hakkında bilgiler verilmiştir. Dördüncü bölümde kullanılarak eş zamanlı haritalama yapan ve konum belirleyen mobil robot uygulamalarının sonuçları şekiller ile verilmiştir. Beşinci bölümde, elde edilen sonuçlar ve öneriler sunulmuştur.

## 2. Parçacık Filtresi Tabanlı Eş Zamanlı Konum Belirleme ve Haritalama

Bu çalışmada, Lazer mesafe sensöründen ızgara haritalama öğrenmek için Reo-Blackwellized Parçacık Filtresini etkili adımlarla uygulayan Gmapping ROS paketi kullanılmıştır [18-19]. Gmapping'in çözüme ilişkin adımları aşağıdaki gibidir.

1) Robot konumunu temsil eden  $i$  numaralı parçacık, başlangıç tahmini  $x_k^{(i)} = x_{k-1}^{(i)} * u_{k-1}$  sağlayacak biçimde bir önceki parçacığın konumundan  $x_{k-1}^{(i)}$  ve sonuncu filtre güncellenmesi esnasında alınan odometri değerlerinden oluşur. Bu formüllerde kullanılan  $*$  operatörü standart konum birleştirme operatörüdür.

2)  $m_{k-1}^{(i)}$  haritasında başlangıç konumu,  $x_k^{(i)}$  olacak biçimde tarama eşleme algoritmaları gerçekleştirilir. Tarama başlangıç konumu olan  $x_k^{(i)}$  çevresinde sınırlar olacak biçimde oluşturulur. Eğer tarama-eşleme hata ile sonuçlanırsa, konum ve ağırlıklar hareket sistemine göre hesaplanır ve 3. ve 4. adım atlanır.

3) Tarama-eşleme sonuçları ile robotun konumunun çevresinden örneklenen bir noktalar kümesi alınır. Verilerin ortalama değeri ve kovaryans matrisi, örneklenen  $x_j$  konumlarında hedef dağılım  $p(z_k | m_{k-1}^{(i)}, x_j)$   $p(x_j | x_{k-1}^{(i)}, u_{k-1})$  noktasal olarak baz alınarak hesaplanır. Aynı anda ağırlık vektörü de hesaplanır.

4)  $i$  parçacığının yeni konumu  $x_k^{(i)}$ , varsayılan dağılımın iyileştirilmiş olan Gauss yaklaşımı  $N(\mu_k^{(i)}, \Sigma_k^{(i)})$  sayesinde çizilir.

5) Parçacıkların ağırlıkları önemli olma durumuna göre güncelleştirilir.

6)  $i$  parçacığının haritası  $m^{(i)}$ , beklenen  $x_k^{(i)}$  konumu ve  $z_k$  gözlemine göre iyileştirilir.

Güncel örneklerin hesaplanmasından sonra yeni bir örnekleme adımı  $N_{eff}$  sonucu dikkate alınarak oluşturulur.  $N_{eff}$  ifadesi ise,  $w^{(i)}$  ile gösterilen  $i$  parçacığının normalleştirilmiş ağırlığı Denklem-1 de olduğu gibi hesaplanır.

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w^{(i)})^2} \quad (1)$$

Burada;  $N$  parçacık sayısıdır.

### 3. Deneysel Ortam

Bu çalışmadaki uygulamalar, LIDAR sensörü, Turtlebot mobil robot ve iki Linux işletim sistemine sahip dizüstü bilgisayar kullanılarak gerçekleştirilmiştir [20].

#### 3.1. Mobil robot

Kobuki firması tarafından üretilmiş olan Turtlebot mobil robotu üzerinde yapılmıştır [14]. Turtlebot'u tercih etme sebebi SLAM uygulamaları için uygun özelliklere sahip olabilmesidir. Turtlebot diferansiyel tekerlek modeli ile oluşturulmuştur. Diferansiyel tekerlek modeli SLAM uygulamalarını yapabilmek için en önemli özelliktir. Bu özellik sayesinde mobil robot olduğu yerde dönebilmektedir. Şekil 1'deki Turtlebot mobil robotun genel özellikleri Tablo 1'de verilmiştir [14].



**Şekil 1.** Uygulamada kullanılan Turtlebot mobil robotu

Turtlebotun içinde Linux tabanlı bir işletim kartı mevcuttur. Bu kart kullanılarak Turtlebot otonom olarak hareket edebilir. Ayrıca Linux tabanlı bir bilgisayara ROS işletim sistemi kurularak gerekli işlemler yapıldıktan sonra manuel kontrolü sağlanabilir.

**Tablo 1.** Turtlebot mobil robotun genel özellikleri

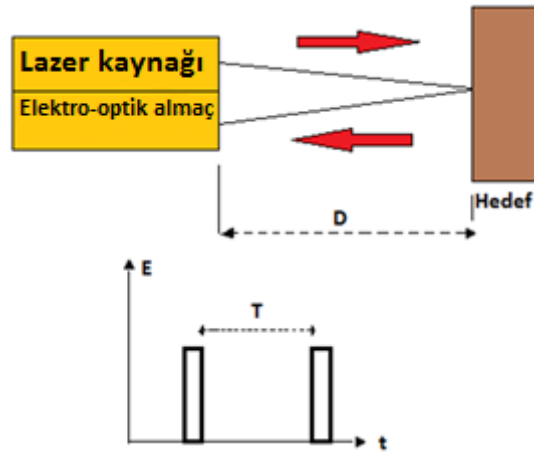
Özellikler	Açıklama
Çap	351.5 mm
Yükseklik	124.8 mm
Ağırlık	2.35 kg
Maksimum öteleme hızı	70 cm / s
Maksimum dönme hızı	180 derece / s
Taşıma kapasitesi	5 kg (sert zemin), 4 kg (halı)
Çarpma sensörleri	Sağ, orta, sol
Uçurum sensörü	Sağ, orta, sol
Uçurum	5cm daha büyük bir derinlikli bir uçurumda aktifleşir
Normal Çalışma Süresi	3/7 saat (küçük / büyük pil)
Motor	2 tane fırçalı DC motor

### 3.2. Lazer mesafe sensörü

Lazer mesafe sensörlerinin çalışma ilkesi, yaydıkları lazer ışınlarının nesneye çarparak geri gelmesine kadar geçen sürenin mesafe cinsinden hesaplanmasına dayanır. Lazer enerjisi kısa sürede ve belirli bir menzildeki nesnelere doğru yollar. Sensörün menzili içinde bulunan herhangi bir nesne, yollanan bu ışık enerjisinin belirli bir bölümünü geri yansıtır. Geri gelen ışık enerjisi, sensörün algılayıcısı tarafından algılanır. Sensörden çıkan ışık enerjisinin nesneye çarpıp geri gelmesi için geçen süre hesaplanır. Lazerdan çıkan bu ışık demetinin hızı ışık hızı bilindiğinden nesne ile sensör arasındaki mesafeyi kolaylıkla bulunabilir. Nesneden yansıyan ışık demetinin sensörde bulunan alıcı sayesinde algılanması sağlanır. Şekil 2’de sensörün basit olarak nasıl çalıştığı gösterilmektedir. Nesne ile sensör arasındaki uzaklık Denklem-2 de olduğu gibi hesaplanır.

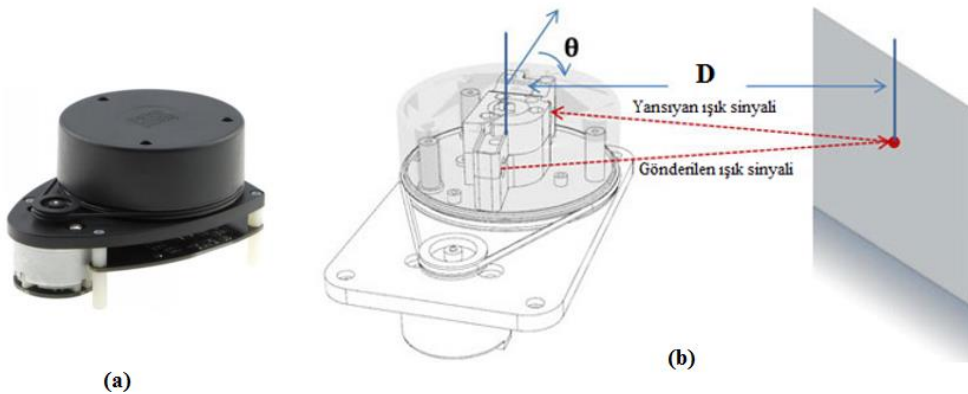
$$D = \frac{c \cdot T}{2} \quad (2)$$

Burada; D nesne ile sensör arasındaki uzaklık, C ışık hızı ve T periyodu göstermektedir.



Şekil 2. Lazer mesafe ölçümü

Mobil robotlarda en fazla tercih edilen lazer çeşidi iki boyutlu olarak adlandırılan hem nesnenin lazere olan mesafesi hem de lazerle olan açısal konumunu veren çeşitlerdir. Uygulamamızda Robo Peak firması tarafından üretilmiş Şekil 3a’daki RP-LIDAR [15] lazer sensörü kullanılmıştır. Bu sensörü Tablo 2’de özellikleri verilmiştir. 360 derece tarama yaparak 6 metrelik menzildeki tüm nesnelerin mesafe bilgisini ölçebilen bir lazer sensördür. Çalışma ilkesi [17] Şekil 3b’de verilmiştir.



Şekil 3. a) RP-LIDAR mesafe ölçüm sensörü ve b) RP-LIDAR çalışma prensibi

**Tablo 2.** RP-LIDAR sensörünün genel özellikleri

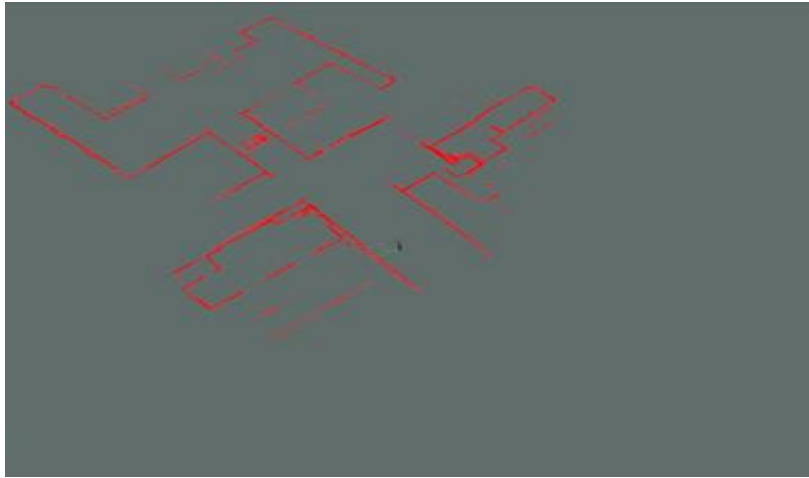
Kapsama açısı	360 derece
Hassasiyet	0.5 derecelik
Tarama hızı	2-10 Hz
Örnek alma	Saniyede 2000
Menzil	0.2-6 metrelik

#### 4. Uygulama Sonuçları

Bu çalışmada, SLAM çerçevesinin uygulaması iki farklı şekilde gerçekleştirilmiştir. Birincisi manuel olarak Turtlebot'un nerde olduğunu görüp ona yön vererek ortamın haritası oluşturulmuştur. İkincisi otonom olarak Turtlebot'a müdahale edilmeden ortamın haritası oluşturulmuştur.

##### 4.1. ROS kullanarak gerçek zamanlı manuel SLAM uygulaması

Bu uygulamada, Turtlebot mobil robotu manuel olarak hareket ettirilerek SLAM uygulaması gerçekleştirilmiştir. Bunun içinde ikinci bir bilgisayara yüklediğimiz ROS işletim sistemini kullanarak USB ara birimi ile Turtlebot'un X (yeşil), Y (kırmızı) ve Z (Mavi) eksen hareketi manuel olarak kontrol edilmiştir. İkinci bilgisayarın klavyesi kullanılarak mobil robotun sağa-sola, ileri-geri ve bulunduğu noktada sabit kalarak kendi etrafında dönmesi hareketleri gerçekleştirilmiştir. Turtlebot'un üzerindeki bilgisayarda ROS'un Gmapping paketi kurularak SLAM uygulaması Şekil 4'deki gibi gerçekleştirilmiştir.



**Şekil 4.** Turtlebot ile elde edilen Fırat Üniversitesi Mekatronik Mühendisliği kat haritası

##### 4.2. ROS kullanarak gerçek zamanlı otonom SLAM uygulaması

Bu uygulamada, ilk olarak, Linux tabanlı bilgisayara sanal bir işletim sistemi olan ROS'un son versiyonu Kinetic Kame yüklenmiştir. İkinci olarak RP-LIDAR'dan gerçek zamanda verilerini almak için gerekli RP-LIDAR paketleri yüklenmiştir. Üçüncü olarak Parçacık Filtre tabanlı Gmapping paketi yüklenmiştir. Son olarak da Turtlebot'a yüklenen programla otonom olarak hareket etmesi sağlanmıştır. Şekil 5'de gösterilen ortamda, otonom olarak hareket ederek SLAM uygulaması gerçekleştirilmiştir. Şekil 6'de otonom olarak hareket eden Turtlebot'un çizdiği harita verilmiştir.



Şekil 5. Uygulama ortamı



Şekil 6. Otonom olarak hareket eden Turtlebot'un SLAM uygulaması a)1. adım, b) 2. adım ve c) 3. adım

## 5. Sonuçlar

Bu çalışmada, otonom mobil robot kavramı için en yeni algoritma çerçevesi olan SLAM incelenmiş ve Turtlebot mobil robotu kullanılarak SLAM uygulamaları yapılmıştır. Otonom olarak sensörler yardımı ile çevresini algılayarak hareket edebilen ve bulunduğu ortamın haritasını çıkarıp konumunu belirleyen mobil robot uygulamaları gerçekleştirilmiştir. İlk uygulamada USB ara birimi ile manuel olarak mobil robotun kontrolü yapılarak aynı anda konum belirleme ve haritalama işlemi gerçekleştirilmiştir. Son uygulamada, mobil robotumuzda bulunan çarpma sensörleri yardımı ile tekerleklere bağlı fırçalı DC motorlar kontrol edilerek mobil robotumuzun otonom olarak hareket etmesi sağlanmıştır. Otonom mobil robotun üzerine yerleştirilen RP-LIDAR'dan anlık olarak veri alınarak ROS ortamında parçacık filtreleme yöntemini içeren Gmapping SLAM algoritmasıyla haritalama ve konum belirleme gerçekleştirilmiştir.

Gerçekleştirilen tüm uygulamalar, mobil robotun etrafını başarıyla tanıdığını ve robotun içinde yer aldığı haritaları oluşturduğunu göstermiştir. Ancak SLAM çerçevesinin uygulanmasının kolay olmadığı açıkça görülmüştür. Farklı sensörlerin birlikte kullanımı, ileri yazılım yöntemleri, kontrol algoritmaları ve filtreler gibi konular bu alandaki çalışmalar için gereklidir.

Gelecek çalışma olarak, 3 boyutlu bilgi verebilen kinect sensörü kullanılarak SLAM gerçekleştirilebilir. Maliyeti düşürerek ve çoklu robotlarla SLAM bilgisini daha hızlı bir şekilde gerçekleştirilebilecek çalışmalar yapılabilir.

## 6. Teşekkür

Bu çalışma, “Rp-Lidar Kullanılarak Mobil Robotlar İçin Eş Zamanlı Konum Belirleme ve Haritalama” İsimli Yüksek Lisans Tez Çalışması Kapsamında, FÜBAP MF.1616 no.lu Yüksek Lisans Tez Projesi desteğiyle gerçekleştirilmiştir.

## Kaynaklar

- [1] Smith, R. and Cheesman, P. On the representation of spatial uncertainty. *Int. J. Rob. Res.*, 1987;5(4): 56-68.
- [2] Leonard, J.J. and Durrant-Whyte, H. F. Simultaneous map building and localization for an autonomous mobile robot. *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 91. Intelligence for Mechanical Systems, IROS'91, 3-5 Nov. 1991; Osaka, Japan, 1442-1447.
- [3] Bailey, T. and Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part i The essential algorithms. *IEEE Robot Autom. Mag.*, 2006; 13(2): 99-108.
- [4] Javad, Z., Cai, Y., and Majid, Y. Comparing EKF and SPKF Algorithms for Simultaneous Localization and Mapping (SLAM). *J. Robot. Netw. Artif. Life*, 2017; 3(4): 217-220.
- [5] Valencia, R., and Andrade-Cetto, J. “Active Pose SLAM” in *Mapping, Planning and Exploration with Pose SLAM*. Springer, Cham, 2018; 89-108.
- [6] Bellian, J.A., Kerans, C., Jennette, D.C. Digital out crop models: applications of terrestrial scanning lidar technology in stratigraphic modeling. *J. Sediment Res.*, 2005; 75(2): 166-176.
- [7] Surmann, H., Nüchter, A. and Hertzberg, J. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Rob. Auton. Syst.*, 2003; 45: 181-198.
- [8] Nguyen, V., Martinelli, A., Tomatis, N. and Siegwart, R. A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS, 2-6 Aug. 2005; Alberta Canada, 1929-1934.
- [9] Kim, P., Chen, J., & Cho, Y. K. SLAM-driven robotic mapping and registration of 3D point clouds. *Automat. Constr.*, 2018; 89: 38-48.
- [10] Kalman, R.E. A New approach to linear filtering and prediction problems. *J. Basic Eng.*, 1960; 82: 35-45.
- [11] Dempster, A.P., Laird A.N. and Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B. Stat. Methodol.*, 1977; 39(1): 138.
- [12] Thrun, S., Fox, D. and Burgard, W. A Probabilistic approach to concurrent mapping and localization for mobile robot. *Mach. Learn.*, 1998; 31(1-3): 2953.
- [13] Yuen, D.C.K. and MacDonald, B.A. A Comparison between EKF and sequential monte carlo techniques for Simultaneous localisation and map-building. *Proc. Australasian Conference on Robotics and Automation*, 27-29 Nov. 2002; Auckland, New Zealand, 26-28.
- [14] <http://www.turtlebot.com/> (07 Mayıs 2018).
- [15] <http://www.slamtec.com/en/Lidar/A1> (07 Mayıs 2018).
- [16] <http://www.ros.org/> (07 Mayıs 2018).
- [17] [https://cdn-shop.adafruit.com/product-files/4010/4010\\_datasheet.pdf](https://cdn-shop.adafruit.com/product-files/4010/4010_datasheet.pdf) (05.02.2018)
- [18] Murphy, K.P. Bayesian map learning in dynamic environments. *Proc. The 12th International Conference on Neural Information Processing Systems*, 29 Nov.-04 Dec. 1999; Denver, US, 1015-1021.
- [19] Grisetti, G. Stachniss, C. and Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.*, 2007; 23(1): 3446.
- [20] Akyol, S. Rp-lidar ve mobil robot kullanılarak eş zamanlı konum belirleme ve haritalama, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, 2017.