Sıralı Monte Carlo Yöntemine Dayalı Eş Zamanlı Konum Belirleme ve Haritalama Algoritması

A Simultaneous Localization and Map Building Algorithm Based on Sequential Monte Carlo Method

Zeyneb Kurt, Sırma Yavuz

Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
{zeyneb,sirma}@ce.yildiz.edu.tr

Özetçe

Bu çalışmada, mesafe ölçümünde sadece kızılötesi sensörlerin kullanıldığı kısıtlı algılama kapasitesine sahip bir robotun, bir bilgisayarla birlikte oluşturdukları otonom sistem vasıtasıyla, kapalı bir mekanda, eş zamanlı olarak kendi konumunu hesaplaması aynı zamanda bu alanın haritasını çıkarması amaçlanmışdır. Robotlara bu becerileri kazandırmak için geliştirilen algoritmalar literatürde Eş Zamanlı Konum Belirleme ve Harita Oluşturma algoritmaları olarak bilinmektedir. Ortamın haritasının çıkarımı ve robotun konumunun belirlenmesi esnasında kızılötesi algılayıcılar kullanılmakta ve robot gideceği yöne geliştirilen algoritma sayesinde karar verebilmektedir. Algılayıcı gürültüsü ve ölçüm belirsizliği gibi durumlar karşılaşılan en temel problemlerdir. Bu çalışmada parçacık filtresi uygulamalarından en yaygın olarak kullanılanı Sıralı Monte Carlo (SMC) algoritması kullanılmış Eş Zamanlı Konum Belirleme ve Harita Oluşturma problemine başarılı bir çözüm önerilmiştir.

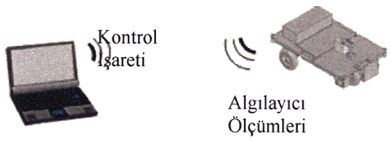
Abstract

In this study, a statistical estimation algorithm is developed to solve the SLAM (Simultaneous Localization and Map Building) problem, by using a robot equipped with only simple and cheap sensors. During map building and simultaneous localization, the robot can sense its environment with infrared sensors and can decide the path to follow by using the developed SLAM algorithm. The most frequent problems in SLAM algorithms are sensors' noise and odometry errors. To solve this problem, Sequential Monte Carlo (SMC) method which is a well known particle filter application is used and promising results were obtained for the SLAM problem.

1. Giriş

Eş zamanlı konum belirleme ve haritalama probleminin lazer ve diğer gelişmiş algılayıcılar kullanılarak çözülmüş bir problem olduğu söylenebilir [1,2]. Literatürde kameralar, lazer algılayıcılar ve sonar algılayıcıların birlikte ya da tek başlarına kullanıldığı maliyetli çözümler yer almaktadır [3]. Ancak, ucuza mal edilebilen, kısıtlı bilgi toplayan kızılötesi algılayıcılar kullanılarak yapılan çalışmalar hala gündemdeki bir konudur. Dünyada bu konuda yapılan çalışmalarda kızılötesi algılayıcılar; robotbilim ve otomasyon, süreç kontrolü, uzaktan algılama ve güvenlik sistemlerinde yaygın olarak kullanılmaktadır [4,5].

Bu çalışmada kullanılan robot, eş zamanlı yer belirleme ve haritalama çalışmalarında kullanılmak üzere Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nde tasarlanmış ve gerçeklenmiştir [6,7]. Sunulan çalışmada gerçeklenen algoritma yardımı ile robot çevrenin ön topolojik bilgilerine ya da birtakım referans nesnelerinin yer bilgisine sahip olmadan ve ihtiyaç duymadan eş zamanlı yer belirleme ve haritalama işlemi yapabilecektir[1]. Robot aynı zamanda kendi yerini tahmin etmekte ve başlangıç noktasına döndüğünü de algılayabilmektedir. Tasarlanan robot bu seyri sırasında etrafını algılayıp nereye gideceğine yine kendi başına karar verecektir. Kullanılan robot sadece kızılötesi algılayıcılar kullanarak etrafındaki cisimlere olan uzaklığını ölçmektedir. Bu ölçümlerden yola çıkarak yoluna nasıl devam edeceğine karar vermekte ve konumunu bildirmektedir. Bu amaçla bir bilgisayar ve robot ile birlikte otonom bir sistem oluşturulmuştur (Şekil 1).



Şekil 1: Bilgisayar ve robot arasındaki iletişim

Robotun çevresindeki engellere olan mesafesini ölçebilmek amacıyla kızılötesi algılayıcılardan alınan bilgi bilgisayara bir RF (Radyo Frekansı) modülü üzerinden aktarılmaktadır. Ön teker üzerinde bulunan artımlı kodlayıcı alınan mesafeyi ölçmekte, tekerin üst kısmındaki potansiyometre ise ön tekerin dönme miktarını ölçmektedir..

2. Sıralı Monte Carlo Yöntemi

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

Literatürde yer alan istatistiksel yöntemlerden özellikle gerçek zamanlı uygulamalarda en yaygın olarak kullanılanı

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

978-1-4244-1999-9/08/$25.00 ©2008 IEEE

Kalman Filtresidir. Smith vd. 1985-1990 yılları arasında SLAM problemine bir çözüm getiren ve günümüzde hala etkin olarak kullanılan bir Kalman Filtresi uygulaması önermişlerdir [8]. Daha sonra birçok araştırmacı bu yöntemi geliştirmiştir. Ancak Kalman Filtresi yöntemi "veri ilişkilendirme" adıyla bilinen, "robotun geçtiği bir yerden tekrar geçtiğini anlaması" problemine çözüm getiremez. Ayrıca Kalman filtresi yönteminde ham algılayıcı verileri kullanılamaz, veri ön işlemlerden geçirilmelidir ve algılayıcı verisİnin sahip olduğu gürültünün dağılımı Gauss tipinde olmak zorundadır.

Veri ilişkilendirme problemine güçlü bir çözüm getiren; ham algılayıcı verisini direk kullanabilen ve veri gürültüsünün nasıl olduğuyla ilgilenmeyen başka bir istatistiksel yöntem ise "Beklenti Enbüyültme" dir. Thrun vd. BE yöntemini kullanarak SLAM algoritması geliştirilirken karşılaşılan veri ilişkilendirme problemine bugüne kadar getirilmiş en başarılı çözümü sunmuşlardır [9]. Ancak bu yöntem gerçek zamanlı uygulanamamaktadır.

Literatürde uygulanan çoğu "parçacık filtresi" yöntemi gerçek zamanlı çalışırken Kalman filtresi kadar etkin ve hızlı olamasa da, hem gerçek zamanlı çalışması hem de veri ilişkilendirme problemine çözüm getirebilmesi nedeniyle Yuen ve MacDonald tarafından da tercih edilmiştir [10]. Yuen ve MacDonald'ın SMC çalışmalarının bu çalışmadan farkı kullandıkları algılayıcı tipi ve gerçek robot üzerinde değil, simülasyon üzerinde çalışmalarıdır.

2.1. Sıralı Monte Carlo Yönteminin Özellikleri

Sıralı Monte Carlo yöntemi Bayes kuralına dayanan, uygulaması kolay ve paralelleştirilebilir bir yöntemdir. Ayrıca, verilerin Gauss (normal) dağılıma sahip olması şart değildir.

Sistemin t anındaki durumu x_t , t anında alınan gözlem ve üretilen kontrol işaretlerine de z_t olarak gösterilirse x_t durum değişkeni tahmin edilmek istenen tüm bilgiyi içerecektir. Veri kümesi z_t , algılayıcılardan en son alınan mesafe verisi kavramsal veri (y_t) ve bilgisayardan robota gitmek üzere üretilen en son kontrol verisi (u_{t-1}) olmak üzere, ikiye bölünürse tahmin sürecinde Bayes Teoremi ile soncul dağılım şu şekilde bulunur:

$$P(x_{t+1} \mid z_{t+1}) = \frac{P(x_t)P(z_{t+1} \mid x_{t+1})}{P(z_{t+1} \mid x_t)P(x_t)} \qquad (1)$$

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

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

$$Bel(x_t) = P(x_t \mid y_t, u_{t-1}, y_{t-1}, \ldots, u_0, y_0) \qquad (2)$$

x_t: sistemin t anındaki durumunu; $y_{1:t}$ ise: 1'den t anına kadar geçen süredeki algılayıcı verisini göstermektedir. (2) numaralı bağıntı Bayes kuralı ile şu hale gelir:

$$P(x_{0:t+1} \mid y_{1:t-1}) = \frac{P(x_{0:t} \mid y_{1:t})P(y_{t+1} \mid x_{t+1})P(x_{t+1} \mid x_t)}{P(y_{t+1} \mid y_{1:t})} \qquad (3)$$

Bayes kuralıyla yazılan "belief" denklemi, özyinelemeli bir parçacık filtresi gösterimiyle yazılarak bir kestirici elde edilebilir; *n* bir sabit olmak üzere:

$$Bel(x_{t+1}) = P(x_{t+1} \mid y_{t+1}, u_t) \qquad (4)$$

$$Bel(x_{t+1}) = nP(y_{t+1} \mid x_{t+1}) \int P(x_{t+1} \mid x_t, u_t) Bel(x_t) dx_t \qquad (5)$$

(5) ile verilen bu kestirim bağıntısı SMC algoritmasında durum ve ölçüm güncellemesinde kullanılmıştır.

E, *m* adet parçacıktan oluşan bir parçacık kümesi olmak üzere E kümesindeki her bir parçacık için çalıştırılan SMC yönteminin en genel algoritma adımları aşağıda verilmiştir:

1 Her bir parçacığa önem katsayısı atanması

2 Ölçümün güncellenmesi

3 Durumun güncellenmesi

Son olarak önem faktörü değerlerinin toplamı 1'i verecek şekilde normalizasyon yapılır.

Başlangıçta her bir parçacığa rasgele birer önem katsayısı atanır. Özyineli güncelleme filtresi algoritmada görüldüğü gibi 3 adımdan oluşmaktadır. Öncelikle bir "i" parçacığı E kümesinden seçilir. İkinci adımda sonraki durum tahmini olan x_t^i ve bunun şartlı olasılığı olan $P(x_t \mid x_{t-1}, u_{t-1})$, en son kontrol girdisi u_{t-1} ve bir önceki parçacığın durum değişkeni x_{t-1}^i kullanılarak hesaplanır. Son adımda da $P(y_t \mid x_t^i)$ ölçümü güncellenir. Bu algoritma eş zamanlı konum belirleme ve haritalama problemine çözüm getirebilmek amacıyla gereken değişikliklere tabi tutulmuştur [11].

3. Eş Zamanlı Konum Belirleme Ve Harita Oluşturma Algoritması

3.1. Gezinme Algoritması

Sistemde, Huang ve Beevers tarafından basit harita çıkarımı için önerilen gezinme algoritmasından yararlanılmıştır [12]. Bu algoritmanın adımları, istatistiksel tahmin yöntemi olan Sıralı Monte Carlo Algoritması ile gerçeklenerek robotun otonom hareket etmesi sağlanmıştır.

3.2. Yörünge Takibi

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

Uygulamamızda kullanılan robot bir ön ve iki arka olmak üzere, üç tekerlek üzerine yerleştirilen bir platformdan

oluşmaktadır. Önde yön belirleyen ve çekme işini yapan bir teker, arkada ise birbirine bir mille bağlı iki teker mevcuttur.

Kullanılan sistemde artımlı kodlayıcı yardımı ile gidilen yolun miktarı (d) ve tekerin üst kısmındaki potansiyometre yardımı ile de ön tekerin gövde ile yaptığı açı (θ_1) bilinmektedir. Arka tekerlerin gövde ile yaptığı açı (θ_2) ise da basit bir geometrik bağıntıyla hareket doğrusu kullanılarak elde edilir. Verilerden yararlanarak robotun konfigürasyonunu veren referans vektörünün bir sonraki koordinatlarını bulmak için gereken kinematik denklemler çıkarılmıştır. Ön teker orta nokta koordinatlarının t anındaki değerleri biliniyorken ($t+1$) anındaki koordinatların hesaplanmasında kullanılacak denklem sistemi (6) ve arka tekerlere ait koordinatların hesaplanmasında kullanılacak denklem sistemi ise (7) ile verilmiştir :

$$\begin{bmatrix} x_{1(t+1)} \\ y_{1(t+1)} \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\cos\theta_1 \\ 0 & 1 & -\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1(t)} \\ y_{1(t)} \\ d \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} x_{2(t+1)} \\ y_{2(t+1)} \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\cos\theta_2 \\ 0 & 1 & -\sin\theta_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2(t)} \\ y_{2(t)} \\ d \end{bmatrix} \tag{7}$$

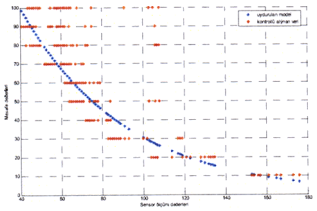
3.3. Çevre Objelerin Mesafelerinin Ölçülmesi

Robotun yanlarına ikişer adet, ön ve arkasına ise birer adet olmak üzere toplam 6 adet Sharp GP2D12 kızılötesi algılayıcılar kullanılmıştır.

Kullanılan kızılötesi mesafe algılayıcının kalibrasyon işlemi için farklı malzemelerden yapılmış duvarlar için ölçümler yapılmıştır. 2 cm aralıkla 1 metreye kadar bütün mesafeler için algılayıcının çıkış gerilimi ölçülmüştür. Robot seyri esnasında algılayıcılarından ölçülen voltajın karşılık geldiği en doğru mesafe değerini belirlemek için bir başvuru çizelgesi kullanmak yerine gerilimle mesafe ilişkisi üssel olarak modellenmiştir :

$$\boldsymbol{mesafe = a * gerilim^b + c} \tag{8}$$

Şekil 2'de kaydedilen bir grup örnek ölçüm değeri ve üstel model kullanılarak uydurulan eğri görülmektedir.



Şekil 2: Kızılötesi algılayıcı verisine uydurulan model

3.4. SMC Algoritmasının Adımları

Robot algılayıcılarının gezinirken çevrede algıladığı farklı karakteristiğe sahip her bir nesne ya da engele dönüm noktası (landmark) denir. Yeni bir dönüm noktası bulunduğunda sistemde her seferinde yeni bir parçacık oluşturulur. Sistemimizde her bir dönüm noktası veya her bir parçacık filtresi, bir duvara ya da engele karşılık gelir. Dolayısıyla algoritma sonucunda "m" adet parçacık filtresi oluştuysa "m" adet duvar bulunmuş demektir. Robot gezinirken her duvara ait çeşitli sayıda noktalar tespit eder, ve her yeni bir dönüm noktası bulunduğunda bunun iç köşe veya dış köşe oluşturduğunu belirten bir önem faktörü atanır.

3.4.1. Sistem ve Ölçüm Modeli

Sistem modeli oluşturulurken, öncelikle x durumu zamanla değişen kısım (s) ve zamandan bağımsız kısım yani engellerin haritası (ρ) şeklinde iki bileşene ayrılır. Daha sonra parametrelerin tahmini yani harita durum bileşeni olan ρ, her bir i engeli için tekrar bileşenlere ayrılır ($\rho_{i=1:L}$) ve her bir i engeli için ayrı tahminler yapılır. Ayrı parçacık filtreleri robotun pozisyonu ve her bir engelin pozisyonlarının tahmininde kullanılır. Tahmin problemi alt parçalara bölünerek ve bileşenler arasındaki bağımsızlık derecesi artırılarak hesaplama yükü azaltılmıştır [12].

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

Ölçüm modeline genellikle kavramsal model ya da gözlemin güncellenme modeli de denir ve $y_{t+1} = F(y_t, x_{t+1})$ şeklinde gösterilir. Ölçüm modeli, sistemin herhangi bir anda çevreden alması muhtemel olan verileri tahmin etmek amacıyla kullanılır. Gerçek gözlem verileri elde edildikten sonra ise beklenen ölçüm değerleri ile gözlenen ölçüm değerleri arasındaki farka bakılarak geliştirilen algoritma için gereken çıkarımlar yapılır. F fonksiyonu kullanılan algılayıcı tipine, uygulamaya ve kaç boyutlu çalışıldığına bağlı olacaktır. Sistemimizdeki gözlem modeli; algılayıcının bir önceki adımda gördüğü noktaya, algılayıcının bir önceki pozisyonuna ve şu anki pozisyonuna bağlıdır. Engelin robota olan uzaklığı alınıp kartezyen koordinatlara taşınıp iki boyutlu geometrik bağıntılar kullanılmaktadır.

3.4.2. Algoritmanın Adımları

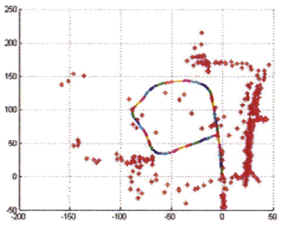
Sistem modellemesi ve ölçüm modellemesi tanımlandıktan sonra oluşturulan SMC algoritmasının adımları şu şekilde verilebilir:

1) Robot pozisyonu için ρ_0 durum filtresine ilk değer ataması yap.
2) L=0 (L: engellerin toplam sayısı, başlangıçta 0'dır)
3) Aşağıdaki adımlar robot başladığı yere dönene kadar tekrarla:
   a. Robot pozisyonunun ρ^t_0 güncelle (t anında)
   b. Kavramsal haritayı ρ^t_j =1:L ile güncele
   c. Beklenen ölçüm değerleri y^{t+1}_{prj} hesapla
   d. Gerçek ölçüm değeri y^{t+1}_{act} ile beklenen ölçüm değeri y^{t+1}_{prj} karşılaştır
   e. Eğer yeni bir engel bulunursa aşağıdaki adımlar işle:
      i. L= L + 1
      ii. Yeni parametre filtresi ρ^t_L 'yi oluşturup ilk değer ataması yap.
      iii. Engel için önem faktörü ataması yap.
   f. Önem faktörüne göre kontrol işareti üretip ve gönder
4) Robot harekete başladığı noktaya dönmediyse 3. adımdan devam et.

3.5. Çevrimdışı Toplanan Veriler ile Haritalama ve Değerlendirme

Geliştirilen sistemde eş zamanlı konum belirleme ve harita oluşturma algoritmasına uyarlanan Sıralı Monte Carlo yöntemi gerçek zamanlı bir çözümdür. Ancak gerçek zamanlı uygulamaya geçmeden önce robot elle kontrol edilmiş ve çevrimdışı veri toplanmıştır. Toplanan veri modellenerek yörünge planlaması yapılmış ve robotun gördüğü engeller haritaya aktarılmıştır.

Buna göre Şekil 3'de ters P şeklinde bir yörünge ile ters L şeklindeki bir engelin nasıl haritalandığı görülmektedir. Harekete (0, 0) noktasından başlanmıştır ve eksenler cm cinsindendir.

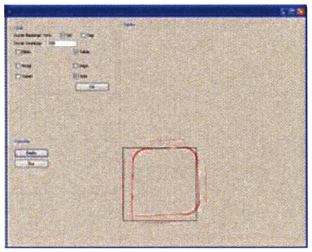


Şekil 3: Ters P yörüngesi ile ters L şeklinde bir engelin bulunması

Çevrimdışı yapılan denemeler sonucunda yörüngenin gerçek robot hareketiyle çok benzediği, bazı kaymalar dışında oldukça başarılı olduğu gözlenmiştir. Konum belirlemeye bağlı olarak çıkarılan haritaların da gerçek mekana oldukça benzediği görülmüştür.

3.6. Çevrimiçi Haritalama ve Değerlendirme

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



Şekil 4: Kare şeklindeki kapalı bir mekanın haritalanması

Şekil 4'de kare şeklindeki kapalı bir mekanda gezinilerek elde edilen robot yörüngesi ve gezinilen ortamın haritası görülmektedir. Ortamda dört adet iç köşe bulunmaktadır. Köşelerin hepsi olması gerektiği gibi tespit edilmiştir.

4. Sonuçlar

Eş zamanlı konum belirleme ve harita oluşturma çalışmalarının geniş bir uygulama alanı vardır: arama kurtarma çalışmalarında en kısa zamanda mümkün olan her yerin aranmasının sağlanması; deprem, yangın gibi insanların çalışmasının ve erişiminin zor olduğu koşullarda canlı tespitinin en etkin ve hızlı biçimde tamamlanabilmesi için ve şehir haritalarının çıkartılmasında kullanılabilirler.

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

Sistemin sadece o andaki değil, sonraki anlardaki saklı durumunun yani robot pozisyonu ve engel pozisyonları tahminlerinin de güçlü bir kestirim modeliyle oluşturulmasının haritalarda meydana gelen kayma sorununu çözebileceği düşünülmektedir. Gelecekteki çalışmalarda bu amaçla şu anda kullanılan SMC algoritmasının iyileştirilmesi hedeflenmektedir.

5. Kaynakça

[1] Beevers K., "Loop Closing in Topological Maps", *Proceedings of the International Conference on Robotics and Automation*, 24:601-613,2005.

[2] Dissanayake, M.W.M.G., Newman, P. et al., "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem", *NSW 2006*, 2006, Sydney, Avustralia.

[3] Thrun, S., Bücken, A., Burgard, W., Fox, D. et. al., "Map Learning and HighSpeed Navigation in RHINO", *Research Reports on Information Science and Electrical Engineering of Kyushu University*, 6(1):77-82,2001.

[4] Castellanos, J. A., Neira, J. and Tardós, J.D. Multisensor, "Fusion for Simultaneous Localization and Map Building", *IEEE Transaction on Robotics and Automation*, 17(6):908-914,2001.

[5] Garulli A., Giannitrapani A., Rossi A., Vicino A.,"Simultaneous Localization and Map Building Using Linear Features", *European Conference on Mobile Robots 2005*, September 2005, Ancona-Italy.

[6] Yavuz S., Amasyalı M. F., Balcılar M., Bilgin G., Dinç T., Kurt Z., "Eş Zamanlı Konum Belirleme ve Harita Oluşturma Amaçlı Otonom Bir Robot", *Eleco 2006*, Aralık 2006.

[7] Kurt Z., "Eş Zamanlı Konum Belirleme Ve Haritalamaya Yönelik Akıllı Algoritmaların Geliştirilmesi", Yıldız Teknik Üniv. FBE Bilgisayar Mühendisliği Anabilim Dalında Hazırlanan Yüksek Lisans Tezi, 2007.

[8] Smith, R., Self, M. and Cheeseman, P., Autonomous Robot Vehnicles, Springer-Verlag New York Inc., New York,1990.

[9] Thrun, S., Fox, D. and Burgard, W., "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots", *Machine Learning*, 31:29–53,1998.

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

[11] Doucet, A., Freitas, N.D. and Gordon, N. *Sequential Monte Carlo Methods in Practice*, Springer Science+Business Media, Inc., New York,2002.

[12] Huang, W. H. and Beevers, K. R., "Topological mapping with sensing-limited robots", *Algorithmic Foundations of Robotics VI*, 17:235-250,2005.