

Rastgele Sıralı Ardışık Bileşim Kullanarak Hareket Planlama

Ferhat Gölbol*, Emre Ege†, Afşar Saranlı*, M. Mert Ankaralı*

*Elektrik ve Elektronik Mühendisliği Bölümü
Orta Doğu Teknik Üniversitesi
Ankara, Türkiye

Email: {ferhatg, afsars, mertan}@metu.edu.tr

†Desistek Robotik, Ankara, Türkiye

Email: emre.ege@desistek.com.tr

Özetçe —Rastgele sıralı ardışık bileşim yöntemi, hem örnekleme tabanlı algoritmalar gibi hızlı sonuç vermekte, hem de noktalar yerine alanları örneklediği için çok daha az sayıda örnekleme işlemi gerektirmektedir. Yakın zamanda bu yöntemi kullanarak hareket planlama yapan iki çalışma, Ege ve Ankaralı'nın insansız su üstü aracı için gerçekleştirdiği Lyapunov fonksiyonu tabanlı yöntemi (2019) ve Gölbol ve diğerlerinin ivme kontrollü noktasal robot için referans yönetici kullanarak gerçekleştirdiği çalışmasıdır (2018). Bu bildiri de bu iki çalışma ele alınmış, benzer yönleri ve farklılıkları açıklanmıştır.

Anahtar Kelimeler—rastgele sıralı ardışık bileşim, hareket planlama, insansız su üstü aracı, referans yönetici.

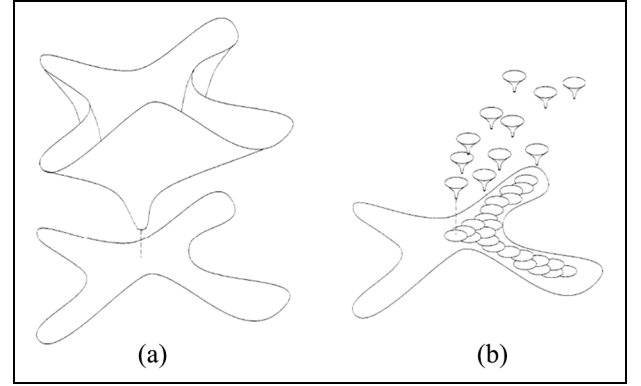
I. GİRİŞ

Hareket ve yörünge planlama, mobil robotların gerçekleştirmesi gereken en temel görevlerden biridir. Bu görev aşağıdaki şekilde ifade edilebilir: Robotun hareket dinamikleri, içinde bulunduğu ortamın modeli, başlangıç ve bitiş konfigürasyonları verildiğinde hareket planlayıcı, robotu engellere çarpmadan hedef konfigürasyona götürecektir. Geleneksel olarak, hareket planlama algoritmaları bu görevi iki aşamada çözer. İlk aşamada robotun hareket modeli göz ardı edilerek, engellere çarpmayan, açık döngü bir yörünge elde edilir. İkinci aşamada ise geri beslemeli kontrol algoritması bu yörüngeyi mümkün olduğunca yakından takip etmeye çalışır [1].

Robot yüksek boyutlu durum uzayına sahipse ya da içinde bulunduğu ortam karmaşıksa, hareket planlama görevini deterministik olarak gerçekleştirmek oldukça güçtür. Örneğin, potansiyel alan yöntemi [2] yerel minimumlara takılırken [3], navigasyon fonksiyonu yönteminde ise [4] ortam karmaşıklıkça hesaplama için gereken zaman oldukça artmaktadır. Bu gibi durumlarda örnekleme tabanlı algoritmalar yeterince kısa sürede güzel sonuçlar vermektedir [1], [5], [6]. Sıralı ardışık bileşim olarak bilinen bir diğer yöntem ise, karmaşık olan ortamı basit bölgelere bölerek, bir bölgeden bir sonraki bölgeye gidecek geri beslemeli kontrolcülerini tasarlamaya dayanır. Bu sayede karmaşık bir hareket planlama problemini daha çok sayıda ancak basit hareket planlama problemlerine indirir [1], [7], [8].

Ege ve Ankaralı, örnekleme tabanlı yöntemlerle sıralı ardışık bileşimi birlikte kullanarak, insansız su üstü araçları

Türkiye Robotik Konferansı, 2019



Şekil 1. Sıralı ardışık bileşim. (a) Bütün noktaları hedefe götüren evrensel, karmaşık 'huni'. (b) Yerel, basit hunilerin sıralı ardışık bileşimi. Bu bileşimde her huni robotu bir sonraki huniye, en son huni de hedefe götürür. Şekil, [7]'den alınmıştır.

(İSA) için simülasyon ortamında yörüngesiz bir hareket planlama algoritması geliştirmiştir [9]. Bu yöntem suyun sönümlenme etkisinin yüksek olduğunu varsaydığından İSA'nın hız kontrollü, lineer olmayan tek tekerli bisiklet modelini ele almıştır. Bu çalışmada kullanılan kontrolcü Lyapunov fonksiyonu tabanlıdır. Bunu takip eden çalışmada Gölbol ve diğerleri de yine simülasyon ortamında ivme kontrollü lineer bir robot için, referans yönetici [10] tabanlı bir algoritma tasarlamıştır [11]. Bu bildirinin amacı, yukarıda bahsi geçen iki yöntemi açıklayıp, bu iki yöntemin birbirlerine göre güçlü yanlarını ve zayıflıklarını karşılaştırmaktır.

Bu bildirinin düzeni aşağıdaki gibidir. Bölüm II ve Bölüm III'te sırasıyla örnekleme tabanlı yöntemler ve sıralı ardışık bileşim hakkında bilgi verilmiştir. Hareket planlama algoritmaları Bölüm IV'te açıklanmıştır. Bölüm V'te simülasyon sonuçları verilmiş, Bölüm VI'da ise bu iki hareket planlama algoritması karşılaştırılmış, gelecekte yapılması planlanan çalışmalara yer verilmiştir.

II. ÖRNEKLEME TABANLI YÖNTEMLER

Robot yüksek boyutlu konfigürasyon uzayına sahipse ya da karmaşık bir ortamda çalışıyorsa, deterministik hareket planlama algoritmalarının hesaplama süreleri oldukça artmaktadır. Bu durumda daha hızlı sonuç elde etmek için örnekleme tabanlı yöntemler geliştirilmiştir [6], [12]. Bu yöntemler, yüksek

boyutlarda ve karmaşık ortamlarda da etkili sonuçlar vermektedir.

Deterministik yöntemlerden farklı olarak örnekleme tabanlı yöntemler, rastgele üretilen bir örnek konfigürasyonun engellere çarpıp çarpmadığını kontrol eden bir modüle ihtiyaç duyar. Sonrasında engele çarpmayan noktalar birleştirilerek bir yol haritası [12] ya da ağaç yapısı [6] oluşturulur. Bu yöntemlerle elde edilen sonuçlar genellikle açık döngü yörüngelerdir. Örnekleme tabanlı yöntemler olasılıksal tamamlık özelliğine sahiptir: eğer başlangıç ve bitiş konfigürasyonlarını birleştiren bir yol varsa, rastgele örnek sayısı arttıkça bu yolu bulamama olasılığı 0'a yakınsar.

A. Hızlıca Keşfeden Rastgele Ağaçlar

Bu bildiride ele alınan iki çalışma da örnekleme tabanlı olan hızlıca keşfeden rastgele ağaçlar (HRA, RRT) algoritmasını kullanır. HRA, yüksek boyutlu konfigürasyon uzaylarında rastgele seçtiği engele çarpmayan örnekleri birleştirerek bir ağaç yapısı oluşturur. Ağacı oluşturmaya başlangıç ya da hedef konumundan başlar ve daha önce aramadığı bölgelerden örnekler alarak haritanın tamamını kaplamaya çalışır. Bu sayede başlangıçla hedef arasında açık döngü bir yol oluşturur.

Algoritma başlangıç (bitiş) noktasını ağacın kökü olarak seçer. Bundan sonra her adımda harita üzerinde rastgele bir örnek nokta alır. O ana kadar oluşturmuş olduğu ağaç üzerinde bu noktaya en yakın düğümü bulur. Eğer örnek noktaya en yakın düğüm arasındaki uzaklık önceden belirlenmiş bir eşik değerindeyse, bu noktayı aynı yönde ve düğüme eşik değeri kadar uzaklıktaki bir noktaya değiştirir. Eğer düğümü noktaya bağlayan yol engellere çarpmıyorsa, algoritma noktayı, ata düğümü en yakındaki düğüm olacak şekilde, ağaca ekler. Bu döngü bitiş (başlangıç) noktasına yeterince yakın bir düğüm oluşturulana kadar devam eder.

III. SIRALI ARDIŞIK BİLEŞİM

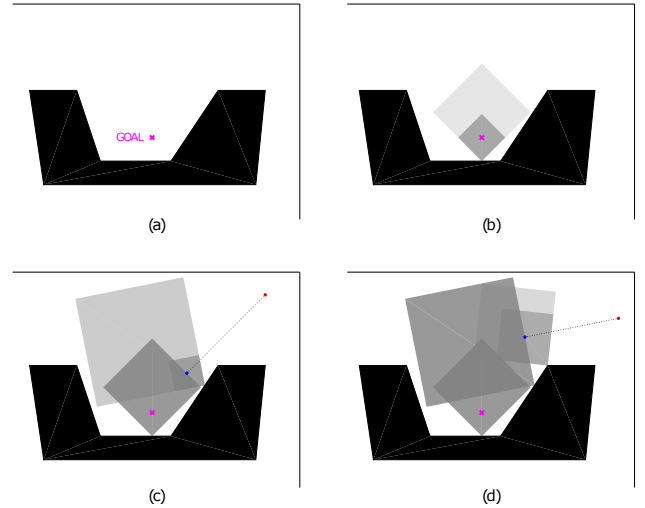
Mason, geribeslemeli kontrol algoritmaları için huni benzetmesini kullanır [13]. Bu benzetmede huninin ağzı kontrolcünün çekim havzasına denk gelirken, huninin boşaltma noktası ise kontrolcünün kararlı denge noktasıdır. Ancak robotun içinde bulunduğu ortam karmaşıksa, ortamdaki bütün noktaları hedef konfigürasyonuna götürecek evrensel bir huni bulmak zordur. Bunun yerine, sıralı ardışık bileşimde, birden fazla sayıda basit huni (kontrolcü) kullanılır. Huniler birbirine öncelik sırasıyla bağlıdır. Her bir huninin görevi, kendi çekim havzasındaki noktaları toplayıp bir üst öncelikli huninin çekim havzasına götürmektir. En üst öncelikteki huni ise çekim havzasına gelen bütün noktaları hedef konfigürasyona götürür. Huni benzetmesi Şekil 1'de verilmiştir.

Sıralı ardışık bileşim, sonuç olarak basit hunilerin ağzlarının birleşim kümesindeki bütün noktaları geriye doğru zincirleme yöntemiyle hedef noktaya taşır. Ancak temel formunda bu yöntem de deterministiktir. Engellerin konfigürasyon uzayındaki gösterimlerine ihtiyaç duyar ve planlama için sonlu elemanlar yöntemini kullanır. Bu nedenle hesaplama karmaşıklığı yüksektir. Bu bildirideki çalışmalar, sıralı ardışık bileşimi örnekleme tabanlı yöntemlerle birleştirilerek daha hızlı sonuç elde eden algoritmalar elde etmiştir.

IV. RASTGELE SIRALI ARDIŞIK BİLEŞİM İLE HAREKET PLANLAMA

Bu bildiride ele alınan yöntemler hareket planlama algoritmasını iki aşamada ele alır. İlk aşamada Bölüm II-A'da verilen hızlıca keşfeden rastgele ağaçlar yöntemiyle Bölüm III'te verilen sıralı ardışık bileşim yöntemi birleştirilerek, düğümleri engel içermeyen hunilerden oluşan bir ağaç elde edilir. Bu ağacın düğümleri alan kaplayan bölgelerden oluştuğu için standart HRA algoritmasına göre çok daha az sayıda düğümler hareket planlama tamamlanır. Yöntem örnekleme tabanlı olduğu için de standart ardışık sıralı bileşim metodundan daha hızlı çalışır. Bu ilk aşama sırasında Lyapunov fonksiyonu tabanlı kontrolcü kullanılacaksa, bu tür kontrolcülerin çekim havzası genelde elipsoid olduğundan düğüm olarak elipsoid huniler kullanılır. Ege ve Ankaralı'da [9] bu huniler daire olarak seçilmiştir. Öte taraftan referans yöneticinin çekim havzası konveks çokgenlerdir [10]. Eğer bu yöntem kullanılacaksa düğümler konveks çokgen şeklinde huniler olmalıdır. Gölbol ve diğerlerinde [11] hesap kolaylığı için karesel huniler kullanılmıştır. Huni şekli seçiminin dışında ağaç oluşturma aşaması robotun hareket denklemlerinden tamamen bağımsızdır. Sadece harita, başlangıç ve hedef konumlarını kullanır. Bu bildiride anlatılan her iki yöntem için ağaç oluşturma aşaması hemen hemen aynıdır ve Bölüm IV-A'da açıklanmıştır.

Eğer hedef konumun kök düğümdeki hunide, başlangıç konumunun da bir yaprak düğümdeki hunide bulunduğu varsayılırsa, robot sürekli içinde bulunduğu düğümden ata düğüme giderek hedefin bulunduğu kök düğüme ulaşabilir. İkinci aşamanın görevi de robotu bulunduğu huninin dışına çıkmadan, bu huniyle ata düğümün hunisinin kesişim alanına iletmektir. Robot kesişim bölgesine geldiğinde ata düğüme girmiş sayılır ve görev artık robotu bu düğümlerle onun ata düğümünün kesişimine taşımaktır. Bu şekilde robot kök düğüme kadar ulaşır. Kök düğümdeki görev de robotu hedef konuma taşımaktır. Düğümler içlerinde engel olmaya-



Şekil 2. Ağaç oluşturma algoritması. (a) İlk huni hedef noktasında oluşturulur. (b) Bunun için merkezi hedef noktası olan ve en yakındaki engele tek noktada değen kare (daire) oluşturulur ve bu şekil engellere ikinci bir noktada değene kadar karşıt yönde büyütülür. (c,d) Haritada rastgele bir örnek nokta alınır (kırmızı) ve kaplanmış alan üzerinde bu örneğe en yakın nokta (mavi) hesaplanır. Yeni şeklin merkezi mavi nokta olacak şekilde b'deki işlem tekrarlanır. Bu şekil, ağaca yeni düğüm olarak eklenir.

çak şekilde üretildiğinden, robotun bulunduğu huninin içinde kaldığı sürece engele çarpmayacağı garanti edilmiştir. Huninin içinde kalma durumu ele alınan iki yöntem arasında farklılık gösterir. Bu durum Bölüm IV-B ve IV-C’de ayrı ayrı açıklanmıştır.

A. Ağaç Oluşturma

Ağaç oluşturma aşaması iki yöntemde benzerlik gösterir. Bu aşama, HRA algoritmasının, düğümleri noktalar yerine daireler ya da kareler olacak şekilde değiştirilmesidir. Ağaç oluşturma aşaması Şekil 2’de gösterilmiştir.

HRA’daki gibi ilk düğüm hedef nokta x etrafında oluşturulur. Bu nokta etrafında huni oluşturmak için, engeller üzerinde bu noktaya en yakın nokta o belirlenir. Lyapunov tabanlı yöntem için merkezi x olan ve engellere o noktasında değen daire oluşturulur. Referans yönetici tabanlı yöntemde ise merkezi x ve bir köşesi o olan kare çizilir. Oluşturulan şekil, engellere ikinci bir noktada değene kadar, o noktası sabit kalacak biçimde ox eksenini boyunca genişletilir. Bu işlemin sonucunda elde edilen şekil, ağacın kök düğümü olur.

HRA’ya benzer biçimde, bir döngü içerisinde, engel veya kaplanmış alan üzerinde olmayan rastgele bir nokta r örneklenir. Kaplanmış alan üzerinde bu noktaya en yakın nokta x ve engel üzerinde x ’e en yakın nokta o hesaplanır. Önceki paragrafta açıklandığı gibi yeni bir huni oluşturulur ve bu huni, ata düğümü x ’in üzerinde bulunduğu düğüm olacak şekilde ağaca eklenir. Bu işlem, hedef noktası bir düğüm içerisinde kalana dek devam ettirilir.

B. Linear Olmayan Sistemde Lyapunov Tabanlı Hareket Planlama

Ege ve Ankaralı, makalelerinde algoritmalarını insansız su üstü aracı (İSA) için geliştirmiştir [9]. Bu makalede İSA’yı 2 boyutlu düzlemde tek tekerli bisiklet olarak modellemişlerdir. Suyun sönümlemesinin yeterince yüksek olduğunu varsaydıklarından, İSA’nın doğrusal ve açısız hızları doğrudan kontrol edilebilmektedir. Bu modelin durum uzayı iki boyutta konum ve tek boyutta yönelim olmak üzere 3 boyutludur. İSA’nın hareket denklemi

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ \omega \end{bmatrix} \quad (1)$$

olarak verilmiştir. v ve ω kontrol edilebilmektedir. İSA gövdesine sabitlenmiş koordinat sisteminde hedef noktanın uzaklığı ρ ve açısı ϕ olmak üzere,

$$\begin{bmatrix} \dot{\rho} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\cos \phi & 0 \\ \frac{\sin \phi}{\rho} & -1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

olarak yazılabilir. Bütün huniler için ortak kontrol kuralı

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} K_v \rho \cos \phi \\ K_\phi \phi \end{bmatrix} \quad (3)$$

olarak seçilsin. Lyapunov fonksiyon adayı

$$V(\rho, \phi) = \alpha \rho^2 + \phi^2, \quad \alpha > 0 \quad (4)$$

için

$$\dot{V}(\rho, \phi) = -2\alpha K_v \rho^2 (\cos \phi)^2 - 2K_\phi \phi^2 + K_v \phi \sin(2\phi) \quad (5)$$

olur. Eğer $K_\phi > K_v$ seçilirse $\dot{V}(\rho, \phi) < 0 \quad \forall (\rho, \phi) \neq (0, 0)$ bulunur. Dolayısıyla kontrolcü kararlıdır. İSA hız kontrollü olduğundan da robot, bulunduğu huninin dışına çıkmadan denge noktasına varabilmektedir.

Hunilerdeki denge noktası, huninin kapladığı dairesel alanın merkezi olarak seçilmiştir. Merkez noktası, ata düğümün dairesel bölgesinin üzerinde olduğundan, bu kontrolcü robotu bulunduğu düğümden ata düğüme taşıyabilmektedir. Robot ata düğüme gide gide en son hedefin içinde bulunduğu kök düğüme kadar yetişir.

C. Linear Dinamik Sistemde Referans Yönetici ile Hareket Planlama

Referans yönetici, kapalı döngü kararlı bir sistemin üzerine eklenen, sistemin referans sinyalini değiştirerek sistemin durum değişkenleri üzerinde kısıtlamaların gerçekleştirilmesini sağlayan bir yöntemdir [10].

Robotun hareket modeli ayrık zamanlı durum uzay modelinde

$$\begin{aligned} q_{k+1} &= Aq_k + Br_k \\ y_k &= Cq_k + Dr_k \end{aligned} \quad (6)$$

ve durum değişkenleri üzerindeki kısıtlamalar

$$y_k \in Y \quad \forall k \quad (7)$$

olarak verilmiş olsun. Bu halde, robotun ilk durum değişkeni q_0 ve referans sinyali \tilde{r} olmak üzere, eğer referans sinyali sabit tutulduğunda (7)’de verilen herhangi bir kısıtlama ihlal edilmiyorsa, (q_0, \tilde{r}) çifti “çıkışı kabul edilebilir” olarak adlandırılır. Çıkışı kabul edilebilir bütün durum-referans çiftlerinin oluşturduğu küme ise “maksimal çıkışı kabul edilebilir küme (MÇKK)” denir. MÇKK’nin nasıl hesaplanabileceği [14]’de anlatılmıştır. Eğer

- C,A çifti gözlemlenebilir ise
- A matrisi asimptotik olarak kararlı ise ve
- Kısıtlama kümesi, denge noktasını içinde bulunduran bir politop olarak verilmişse, yani $Y = \{y | Sy \leq s\}$ ¹

MÇKK de bir politop olmaktadır:

$$MÇKK = \{(q_0, \tilde{r}) | H_q q + H_r \tilde{r} \leq h\} \quad (8)$$

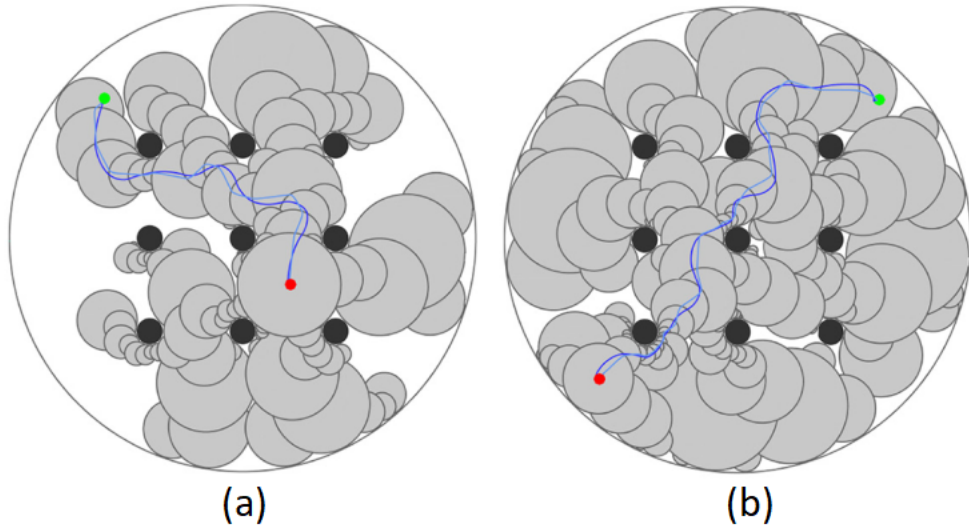
MÇKK bulunduktan sonra skalar referans yönetici aşağıdaki gibi tanımlanmıştır: Değiştirilmiş referans sinyali \tilde{r} başlangıçta çıkışı kabul edilebilir bir değere ayarlanır. Sonrasında da aşağıdaki denkleme göre güncellenir:

$$\tilde{r}_{k+1} = \tilde{r}_k + \kappa(r - \tilde{r}_k) \quad (9)$$

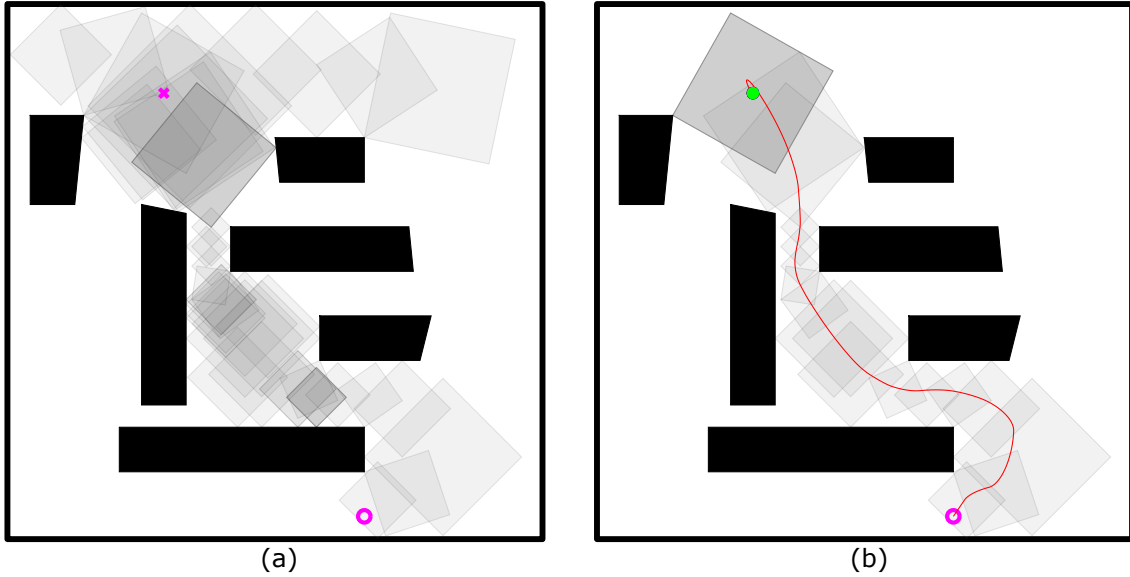
Bu denklemde κ , 0 ile 1 arasında, (q_k, \tilde{r}_k) çıkışı kabul edilebilir kalacak şekilde maksimize edilir. Bu sayede değiştirilmiş referans sinyali \tilde{r} gittikçe asıl referans r ’a yaklaşırken durumun her zaman çıkışı kabul edilebilir kalması sağlanır. Referans yöneticini hareket planlama için kullanan başka çalışmalar da mevcuttur [15], [16].

Gölbol ve diğerleri, bildirilerinde iki kere integral alan noktasal robot modelini kullanmışlardır. İvme kontrollü olan bu robotu 0.05s zamanla ayrık zamana alıp, $K_p = 4, 1$ ve $K_d =$

¹Burada \leq işareti eleman eleman karşılaştırma olarak kullanılmıştır.



Şekil 3. Lineer olmayan sistemde Lyapunov fonksiyonu tabanlı yöntemin iki farklı başlangıç ve hedef konfigürasyonu için sonucu. İçteki siyah daireler engelleri, dıştaki siyah çember haritanın sınırlarını, yeşil nokta başlangıç noktasını ve kırmızı nokta hedefi gösterir. Gri daireler algoritmanın oluşturduğu hunilerdir.



Şekil 4. Lineer dinamik sistemde referans yönetici tabanlı algoritmanın sonucu. İçteki siyah çokgenler engelleri, dıştaki siyah kare harita sınırlarını, 'o' başlangıç noktasını ve 'x' hedefi gösterir. (a) Gri kareler algoritmanın oluşturduğu hunilerdir. (b) Sadece başlangıçta hedefe götüren yoldaki huniler çizilmiştir.

2,2 parametrelerini kullanan PD kontrolcü eklemiştir. Robotun kare şeklindeki huninin içinde kalması, robotun konumu, dolayısıyla durum değişkeni üzerinde bir kısıtlamadır. Bu kısıtlama politop olarak yazılabildiği için skalar referans yönetici yöntemine uygundur. Bu yöntemde orta vadeli referans sinyali r , robotun içinde bulunduğu huniyle ata düğümün hunisinin kesişim alanının orta noktasıdır. Referans yönetici, \tilde{r} 'yi değiştirerek robotun sürekli karenin içinde kalmasını sağlarken, PD kontrolcü de robotu zamanla r 'ye doğru, dolayısıyla ata düğüme götürür.

V. SIMÜLASYONLAR

Ege ve Ankaralı, geliştirdikleri yöntemi Matlab simülasyonunda denemişlerdir. Simülasyonda kullandıkları ortam çembersel olup, alanda 9 adet dairesel engel

bulunmaktadır. Oluşturulan ağaç yapısı ve robotun izlediği yol, iki farklı simülasyon için Şekil 3'te verilmiştir. Yöntemi standart HRA algoritmasıyla karşılaştırmak için aynı haritada HRA da uygulanmıştır. Monte-Carlo simülasyonlarına göre HRA ortalama 3.4 saniyede ve 971 düğüm oluşturarak hareketi planlamışken, rastgele sıralı ardışık bileşim algoritmasında bu değerler 0.1 saniye ve 134 düğümdür.

Göbol ve diğerleri de yöntemlerini, yine Matlab üzerinde, çokgensel engeller içeren kare şeklinde bir haritada denemişlerdir. Oluşan ağaç yapısı ve robotun izlediği yol Şekil 4'te verilmiştir. Algoritma 0.3 saniyede ve 48 düğüm üreterek ağacı oluşturmuştur. Algoritmanın ikinci aşamasında kullanılacak olan referans yöneticinin MÇKK'sinin hesaplanması da 1.2 saniye sürmüştür.

VI. SONUÇ

Rastgele sıralı ardışık bileşim yöntemi, hem örnekleme tabanlı olduğu için hızlı ve etkili sonuç verebilmekte, hem de ortamı basit 'huni'lerin birleşimi olarak ele aldığından basit kontrolcülerle kullanılabilir. Bu yöntemin hareket planlama alanında uygulamaları vardır. Bu uygulamalara yakın zamandan iki örnek olarak Ege ve Ankaralı'nın Lyapunov fonksiyonu tabanlı çalışması ve Gölbol ve diğerlerinin referans yönetici tabanlı çalışmasıdır.

Ege ve Ankaralı'nın çalışması, İSA'nın lineer olmayan hareket modeliyle çalışmaktadır. Bu yöntem, robotun hedefe olan uzaklığının karesini Lyapunov fonksiyonunda kullandığından, hunilerini dairesel olarak seçmektedir. Standart HRA algoritmasına göre daha az sayıda düğüm ile ve daha kısa sürede sonuç bulmaktadır. Yöntemin eksik kaldığı alan ise robotun dinamik modelini dikkate almamasıdır.

Gölbol ve diğerleri, ivme kontrollü dinamik bir robot için benzer bir çalışma yapmıştır. Bu yöntem referans yönetici kullandığından, karesel şekilli huniler kullanmıştır. Bu yöntemin eksikliği ise robotun hareket modelinin lineer olmasını gerektirmesidir. Gelecekte yapılması planlanan çalışma ise hem lineer olmayan, hem dinamik bir robot modeli için çalışacak bir algoritma tasarlamaktır. Hareketli engellerden oluşan dinamik bir ortamda çalışacak bir algoritma geliştirmek de bir diğer çalışma konusudur.

TEŞEKKÜR

Bu çalışma TÜBİTAK (Türkiye Bilimsel ve Teknolojik Araştırma Kurumu) tarafından 118E195 projesi kapsamında desteklenmektedir.

KAYNAKÇA

- [1] L. Yang and S. M. LaValle, "A framework for planning feedback motion strategies based on a random neighborhood graph," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 544–549.
- [2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.
- [3] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE, 1991, pp. 1398–1404.
- [4] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Departmental Papers (ESE)*, p. 323, 1992.
- [5] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [6] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.
- [7] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.
- [8] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.
- [9] E. Ege and M. M. Ankaralı, "Feedback motion planning of unmanned surface vehicles via random sequential composition," *Transactions of the Institute of Measurement and Control*, p. 0142331218822698, 2019.
- [10] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *2014 American Control Conference*. IEEE, 2014, pp. 226–241.
- [11] F. Gölbol, M. M. Ankaralı, and A. Saranlı, "Rg-trees: Trajectory-free feedback motion planning using sparse random reference governor trees," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6506–6511.
- [12] L. E. Kavraki and J.-C. Latombe, "Probabilistic roadmaps for robot path planning," 1998.
- [13] M. Mason, "The mechanics of manipulation," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 544–548.
- [14] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
- [15] O. Arslan and D. E. Koditschek, "Smooth extensions of feedback motion planners via reference governors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4414–4421.
- [16] C. Petersen, A. Jaunzemis, M. Baldwin, M. Holzinger, and I. Kolmanovsky, "Model predictive control and extended command governor for improving robustness of relative motion guidance and control," in *Proc. AAS/AIAA space flight mechanics meeting*, 2014.