

# ROS Tabanlı Tümyönlü Mobil Robotlar için Yerel Planlayıcı Algoritmalarının Karşılaştırılması

## A Comparison of Local Planner Algorithms for a ROS-based Omnidirectional Mobile Robot

Mehmed Rafi İmamoglu<sup>1</sup>, Ecem Sümer<sup>1</sup>, Hakan Temeltaş<sup>1</sup>

<sup>1</sup>Kontrol ve Otomasyon Mühendisliği Bölümü  
İstanbul Teknik Üniversitesi, İstanbul

imamoglu20@itu.edu.tr, sumerec@itu.edu.tr, temeltash@itu.edu.tr

### Özetçe

Tümyönlü mobil robotlara uygulanabilir yerel planlama algoritmaları konusu henüz kapsamlı bir şekilde araştırılmamıştır. Bu çalışmada, Dynamic Window Approach ve Timed Elastic Band olmak üzere öne çıkan iki yerel planlayıcı algoritmasının, tümyönlü mobil robotlar için derinlemesine bir karşılaştırması yapılarak literatüre katkıda bulunulması hedeflenmiştir. Algoritmalar için karşılaştırmalı yapmış olduğumuz gerçek dünya analizleri; park manevraları, dönüş manevraları, statik engellerin etrafında gezinme ve dinamik engelden kaçınma olmak üzere dört kritik senaryoyu kapsamıştır. Algoritma performansları, her testin özel gerekliliklerini karşılamak üzere seçilen çeşitli ölçüm metrikleriyle detaylı bir şekilde incelenmiştir. Metrikler arasında toplam zaman, kat edilen toplam mesafe, robotun hareketlerinin ayrıntıları, hızdaki değişiklikler, pozisyon ve yönelim hataları bulunmaktadır. Ancak, tüm metrikler her test senaryosuna uygulanabilir olmadığından, her test durumunun gerekliliklerine dayalı olarak seçici bir şekilde kullanılmıştır. Bu karşılaştırmalı çalışmadan elde edilen bulgular, her iki algoritmanın güçlü ve zayıf yönlerini ortaya koymakta ve otonom robot teknolojisi alanındaki gelecek araştırma ve geliştirme çalışmalarını için değerli bir referans niteliği taşımaktadır.

### Abstract

The topic of local planner algorithms applicable to omnidirectional mobile robots is yet to be thoroughly investigated. In this study, it is aimed to contribute to the literature by performing an in-depth comparison of two prominent local planner algorithms, Dynamic Window Approach and Timed Elastic Band, for omnidirectional mobile robots. Our comparative real world analysis of these algorithms spanned four critical scenarios: parking and turning maneuvers, navigating around static obstacles, and avoiding dynamic obstacles. The performance of each algorithm was evaluated using a select set of metrics, tailored to the specific conditions of each test. Metrics included total time taken, total distance traveled, robot's movements, changes in velocity, position and orientation errors. However, not all metrics

were applicable to every test scenario, and as such, were employed selectively based on the requirements of each test case. The findings from this comparative study provide insights into the strengths and weaknesses of both algorithms, serving as a valuable reference for future research and development in the field of autonomous robotics.

### 1. Giriş

Son yıllarda, tümyönlü araçların kullanımı ve bu tür araçlara olan ihtiyaç giderek artmıştır. Bu araçlar, dönmeye gerek duymadan tümyönlü hareket yapabilme kabiliyeti sayesinde, endüstriyel ortamlar gibi dar alanlara uygundur.

Mobil robotlar için yol planlama algoritmalarının karşılaştırılması literatürde geniş ve detaylı bir şekilde ele alınmıştır. Cybulski ve arkadaşları [1] öne çıkan üç tane yerel planlayıcı algoritması olan TEB [2], DWA [3] ve EBand'ı [4] simülasyon ortamında ve gerçek dünya testleri yaparak karşılaştırmıştır. Gerçek dünya testlerini diferansiyel tahrikli Turtlebot 3 [5] robotunu kullanarak gerçekleştirmişlerdir. Simülasyon ortamı olarak ise Gazebo'yu [6] kullanmışlardır. Testlerini, ilk önce sadece statik engellerin bulunduğu bir ortamda, daha sonrasında ise bu ortama dinamik engeller de koyarak gerçekleştirmişlerdir. Yaptıkları testler sonucunda, üç algoritmanın da görevleri başarıyla tamamladığı ve benzer sonuçlar verdiklerini görmüşlerdir. Bir başka çalışmada ise, Naotunna ve Wongratanaphisan [7] yine aynı algoritmaları, ağır bir diferansiyel tahrikli robot üzerinde, gerçek dünya testleri yaparak karşılaştırmışlardır. Testlerini, bir önceki çalışmaya benzer olarak, ilk önce statik engellerin bulunduğu bir ortamda, daha sonrasında ise bu ortama dinamik engeller ekleyerek yapmışlardır. Testlerin sonucunda, ağır bir diferansiyel tahrikli robot için TEB ve DWA algoritmalarının, EBand'e göre daha iyi sonuç verdiğini, TEB ve DWA arasında ise belirgin bir farkın olmadığını söylemişlerdir. Benzer bir çalışmada ise Pittner ve arkadaşları [8] aynı algoritmaları, simülasyon ortamı ve gerçek dünya deneyleri yaparak karşılaştırmışlardır. Robot olarak ise diferansiyel tahrikli Turtlebot'u kullanmışlardır. Diğer çalışmalara benzer şekilde, gerçek dünya testlerini ilk önce sadece statik engellerin bulunduğu bir ortamda, daha sonrasında ise aynı ortama dinamik engel ekleyerek yapmışlardır. Yaptıkları testlerin sonucuna göre, en iyi

performans gösteren algoritma TEB'dir. DWA algoritması dinamik engellerle sorun yaşamıştır, EBand algoritması ise ilk deney olan, sadece statik engellerin bulunduğu ortamda hata vermiştir. Kim ve arkadaşları [9] yapmış oldukları çalışmada, Base, Eband ve TEB'i, küçük boyutlu diferansiyel tahrikli bir robot üzerinde gerçek dünya testleri yaparak karşılaştırmışlardır. Yaptıkları testler sonucunda, algoritmalar arasında herhangi bir üstünlük farkı görememişlerdir. Yukarıda bahsedilen yerel planlayıcı karşılaştırma testlerine bakıldığında, hepsinde diferansiyel tahrikli mobil robotlar kullanıldığı görülür. Biz ise, çalışmamızı tümyönlü mobil robot kullanarak gerçekleştirdik ve aynı zamanda, çoğu karşılaştırmada yapılmayan ancak önemli olan park etme senaryosunu da gerçek dünya testlerimizde gerçekleştirdik.

Tümyönlü mobil robotların otonom navigasyonunda, hassasiyet ve tekrarlanabilirlik özellikleri, üretim hatları, havaalanı, lojistik sistemler vb. gibi çeşitli uygulamalarda çok kritiktir. Bu özellikler sadece etkin yol navigasyonunu sağlamakla kalmaz, aynı zamanda sistemin güvenilirliğine ve sağlamlığına da katkıda bulunur. Bu araştırmanın temel amacı, mobil robot teknolojisinde iki önde gelen yerel planlama algoritması olan Timed Elastic Band ve Dynamic Window Approach'u, tümyönlü mobil robot kullanarak karşılaştırmalı bir analizini gerçekleştirmektir. Yazarların bildiği kadarıyla, böyle bir karşılaştırmalı çalışma mevcut literatürde daha önce incelenmemiştir. TEB ve DWA, Robot İşletim Sistemi (ROS) [10] 'de önceden hazırlanmış paketler olarak mevcuttur. Bu çalışmanın sonuçları, gelecekteki çalışmalarda tümyönlü mobil robotlar için yerel planlama algoritması seçerken önemli bir referans sağlayacaktır.

Makale, aşağıdaki şekilde düzenlenmiştir: İlk olarak, bu çalışmada kullanılan yerel planlayıcı algoritmaları kapsamlı bir şekilde açıklanmaktadır. Daha sonra; kullanılan robot, deney ortamları ve deney bulguları açıklanmaktadır. Son olarak, sonuç bölümünde, elde edilen sonuçlar karşılaştırmalı olarak tartışılmıştır.

## 2. Planlayıcı Algoritmaları

Bir mobil robotun belirlenmiş bir hedef noktaya ulaşması için doğru yol planlaması ve bu yolu izlemesinin hayati öneme sahip olduğunu belirtmek gerekir. Robot, istenen yere ulaşmak için hem global hem de yerel yollar oluşturur. Global planlayıcı, statik haritayı ve robotun boyutunu göz önünde bulundurarak, robotun mevcut pozisyonundan hedefe bir yol oluşturur. Ancak, oluşturulan yolu takip etmek için gerekli olan spesifik adımları dikkate almaz. Öte yandan, yerel planlayıcı, hedefe ulaşmak için gereken uygun yönlendirmeyi belirlemek amacıyla, robotun kinematiği, dinamik özellikleri ve sensör verileri hakkındaki detaylarla birlikte global yolu takip eder. Global yol planlamasını kullanmanın temel avantajı, hedef noktaya giden tüm yolu bir seferde oluşturabilme yeteneğidir. Ancak, global yol planlamasının hesaplamaya yükü ve global yol ayarlandıktan sonra ortaya çıkabilecek yeni engelleri dinamik olarak yanıt verememe gibi bazı dezavantajları vardır. Yerel planlayıcı, robotun hareket modelini göz önünde bulundurur ve global yolu takip etmek için gerekli olan etkin ve güvenli manevraları hesaplar.

Holonomik ve holonomik olmayan robotlar arasında hareket kabiliyetleri farklılık gösterir ve bu da hız aralıklarını etkiler. Holonomik olmayan robotlar için hareketler, robotun x-ekseni

boyunca hareketi ve robotun z-ekseni etrafında dönüşü ile sınırlıdır. Öte yandan, holonomik robotlar için hareketler, robotun hem x hem de y eksenlerindeki hareketini, aynı zamanda robotun z-ekseni etrafındaki dönüşünü kapsar.

Robotun hedef noktaya doğru hareketi, her iki planlayıcının koordinasyonuna dayanır. Robotun istenen hedef noktaya ulaşırkenki stabilitesi ve hassasiyeti, kullanılan yerel planlayıcı algoritmalarına bağlı olacaktır. Bu çalışmada, Timed Elastic Band ve Dynamic Window Approach olmak üzere iki yerel planlayıcı algoritması test edilmek üzere seçilmiştir. Algoritmalar ROS ile uyumludur ve algoritmaların ayrıntıları aşağıda verilmiştir.

### 2.1. TEB Yerel Planlayıcı

Timed Elastic Band (TEB) yerel planlayıcı algoritması, otonom araç navigasyonu alanında sıkça kullanılan bir algoritmadır [11]. TEB algoritması, global yolu takip ederken en kısa mesafeyi göz önünde bulunduran ve yol boyunca engellerden kaçınan Elastic Band (EBand) [12] algoritmasının gelişmiş bir versiyonudur [13]. Ancak, hedefe olan en kısa mesafeye öncelik vermek yerine, zaman etkin rotayı belirlemeye odaklanılır [14]. Planlayıcı, gerçek zamanlı olarak global planlayıcı tarafından oluşturulan yörüngeyi rafine eder ve zaman verimliliğini optimize eden bir yol oluşturur. TEB algoritması çalışma zamanı boyunca uyarlanabilir olması nedeniyle dinamik ortamlar için daha uygundur [15]. TEB algoritmasının en iyi şekilde çalışabilmesi için; robotun kinematiğinin, dinamiğinin, geometrik şeklinin ve ivme sınırlarının, ayrıca engellerden kaçınmak için gerekli güvenli mesafenin bilinmesi gerekir [1].

TEB algoritması, *teb\_local\_planner* paketi kullanılarak ROS'a entegre edilmiştir [16]. ROS Kinetic sürümünden önce, planlayıcı algoritması yalnızca holonomik olmayan robotları desteklemektedir. Ancak, ROS Kinetic'ten itibaren holonomik robotları da desteklemektedir. TEB algoritmasının performansı [17] çalışmasında vurgulanmıştır, burada araştırmacılar, araç kinematik modelinin mükemmel şekilde ayarlanmamış olması veya aracın yolundan sapması durumunda bile TEB'nin uygun bir seçenek olabileceğini bulmuşlardır.

### 2.2. DWA Yerel Planlayıcı

Dynamic Windows Approach (DWA) robotik alanında yaygın olarak kullanılan, robotun dinamik ve kinematik kısıtlarını dikkate alan gerçek zamanlı bir hareket planlama algoritmasıdır [3]. DWA, çevresel koşulları göz önünde bulundurarak robotun hız komutlarını dinamik olarak ayarlayıp çarpışma olmayan yörüngeler sunar. DWA, robotun hız uzayındaki bir dinamik pencere içinde hareketini dikkate alma prensibine dayanır. Bu dinamik pencere, robotun herhangi bir anda sahip olabileceği uygun hız aralığını temsil eder.

Algoritma, optimal hız komutunu belirlemek için iki aşamalı bir süreç gerçekleştirir. "Arama alanı" olarak bilinen ilk adımda, DWA, robotun dinamik kısıtlarına dayanarak uygulanabilir bir dizi hareket komutu oluşturur. Bu kısıtlar genellikle maksimum öteleme ve dönme hızlarını, ayrıca ivme limitlerini içerir. Ayrıca, her komutun güvenliği, robotun o komutta, yörüngesindeki en yakın engeli bulmadan önce durup duramayacağını belirleyerek değerlendirilir. Sonuç olarak, elde edilen arama alanı yalnızca uygulanabilir ve güvenli komutlardan oluşur [18]. Sonraki "optimizasyon" aşamasında, DWA en uygun

Tablo 1: Park testi pozisyon ve yönelim hataları

| Testler        | X-Ekseni Hata (cm)    | Y-Ekseni Hata (cm)    | Yaw-Ekseni Hata (derece) |
|----------------|-----------------------|-----------------------|--------------------------|
| Test 1         | TEB: 0.331 DWA: 0.202 | TEB: 0.372 DWA: 0.559 | TEB: 0.138 DWA: 0.811    |
| Test 2         | TEB: 0.737 DWA: 2.178 | TEB: 0.914 DWA: 0.573 | TEB: 0.030 DWA: 250.355  |
| Test 3         | TEB: 0.125 DWA: 2.605 | TEB: 0.421 DWA: 0.750 | TEB: 0.464 DWA: 75.796   |
| Test 4         | TEB: 0.906 DWA: 0.239 | TEB: 0.775 DWA: 0.125 | TEB: 0.182 DWA: 0.174    |
| Test 5         | TEB: 0.056 DWA: 0.318 | TEB: 0.758 DWA: 0.660 | TEB: 0.924 DWA: 22.535   |
| Test 6         | TEB: 0.168 DWA: 0.618 | TEB: 1.133 DWA: 0.211 | TEB: 0.477 DWA: 0.028    |
| Test 7         | TEB: 0.325 DWA: 0.359 | TEB: 0.942 DWA: 1.422 | TEB: 0.453 DWA: 87.399   |
| Test 8         | TEB: 0.149 DWA: 0.253 | TEB: 0.129 DWA: 3.197 | TEB: 0.709 DWA: 52.569   |
| Test 9         | TEB: 0.169 DWA: 2.493 | TEB: 0.059 DWA: 0.797 | TEB: 0.274 DWA: 32.658   |
| Test 10        | TEB: 0.037 DWA: 1.029 | TEB: 1.037 DWA: 0.181 | TEB: 1.408 DWA: 6.895    |
| Standart Sapma | TEB: 0.293 DWA: 0.998 | TEB: 0.382 DWA: 0.908 | TEB: 0.415 DWA: 76.332   |
| Ortalama       | TEB: 0.300 DWA: 1.029 | TEB: 0.654 DWA: 0.848 | TEB: 0.506 DWA: 52.922   |

komutu seçer. Her komutun uygunluğu, hedef fonksiyonun değerine göre değerlendirilir. Orijinal Dynamic Window Approach'ta, hedef fonksiyon, denklem (1)'de belirtildiği gibi tanımlanır [19].

$$F(\vec{v}) = k_1 \cdot \text{pozisyon}(\vec{v}) + k_2 \cdot \text{engel}(\vec{v}) + k_3 \cdot \text{vel}(\vec{v}) \quad (1)$$

Burada,  $k_1$ ,  $k_2$  ve  $k_3$  ağırlık parametrelerini temsil eder.  $\text{pozisyon}(\vec{v})$  yörüngenin hedefe doğru açılma yakınlığını ölçer.  $\text{engel}(\vec{v})$   $\vec{v}$ 'nin yörüngesinde karşılaşılan en yakın engelin mesafesini,  $\text{vel}(\vec{v})$  ise robotun hızını belirtir.

DWA, `dwa_local_planner` paketi kullanılarak ROS'a entegre edilmiştir [20].



(a) Park testi (senaryo I) test alanı. (b) Dönme testi (senaryo II) test alanı.

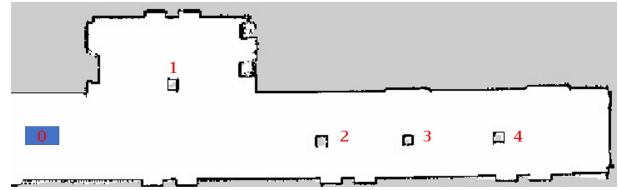


(c) Statik engel testi (senaryo III) test alanı. (d) Dinamik engel testi (senaryo IV) test alanı.

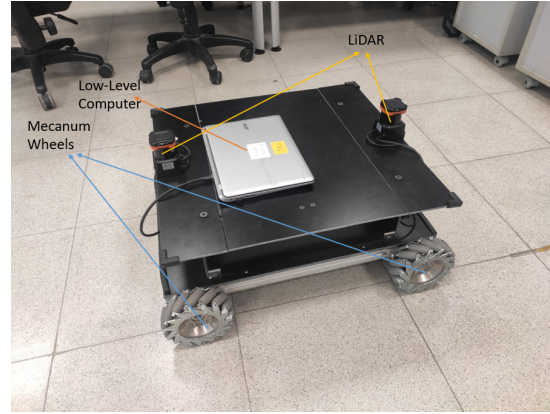
Şekil 1: Gerçek dünya deney ortamları.

### 3. Gerçek Dünya Deney Test Sonuçları

Daha önce belirtildiği gibi, bu çalışmanın amacı, tümyönlü araçların otonom navigasyonu için iki yerel planlayıcı algoritmasının doğruluğunu ve tekrarlanabilirliğini değerlendirmek ve



Şekil 2: 2B harita üstünde statik engel testinde (senaryo III) izlenen yol.



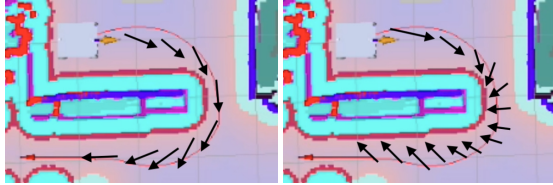
Şekil 3: İTÜ tümyönlü araç (ITU omnidirectional vehicle - ITU ODV).

karşılaştırmaktır. Bu hedef için dört farklı senaryo gerçekleştirdik: park manevraları, dönüş manevraları, statik engellerin etrafında navigasyon ve dinamik engellerden kaçınma. Bu senaryolar, Şekil 1'de gösterilen ortamlarda gerçekleştirilmiştir.

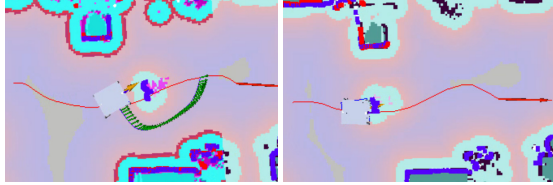
#### 3.1. ITU ODV'nin Özellikleri

Araştırma robotu, İTÜ Robotik Lab'da tasarlanıp üretilen İTÜ tümyönlü aracı (ITU omnidirectional vehicle - ITU ODV) 'dir. Robot, iki adet 240 derece Hokuyo UTM-30LX LIDAR, dört adet mecanum tekerlek, dört adet tekerlek kodlayıcı, dört adet Maxon DCX32L motor, dört adet Maxon ESCON 50/5 motor kontrolörü, bir adet Xsens MTi-30-2A8G4-DK IMU sensörü ve Arduino mega mikrodenetleyici içerir, hepsi bir adet 12V 22Ah

jel batarya ile çalışır. Şekil 3, ITU ODV'nin genel bir görünümünü göstermektedir. ROS Kinetic çalıştıran iki bilgisayar, ROS Master üzerinden bağlanmıştır. İlk bilgisayar, tekerleklerden, LiDAR'dan, IMU'dan veri toplama ve servo kontrolü gibi düşük seviye sistem işlemlerinden sorumludur. İkinci bilgisayar ise navigasyon, haritalama ve konumlama işlemleri ile ilgilenmek üzere ayrılmıştır. ITU ODV'yi kullanarak bir dizi test gerçekleştirdik ve bu testlerin kapsamlı bir açıklaması takip eden bölümlerde sunulacak.



(a) TEB algoritması kullanılan dönüş senaryosunda (senaryo I) robotun hareketi. (b) DWA algoritması kullanılan dönüş senaryosunda (senaryo II) robotun hareketi.



(c) TEB algoritması kullanılan dinamik engel testinde (senaryo III) robotun hareketi. (d) DWA algoritması kullanılan dinamik engel testinde (senaryo IV) robotun hareketi.

Şekil 4: TEB ve DWA algoritmaları kullanılan dönüş (senaryo II) ve dinamik engel senaryolarında (senaryo IV) robotun hareketi.

### 3.2. Senaryo I - Park Etme

Park etme, özellikle malzemelerin etkin bir şekilde yüklenmesi ve boşaltılması için hassas ve doğru konumlandırmanın gerekli olduğu endüstriyel ortamlarda, mobil robotlar için hayati önem taşıyan ve sıkça karşılaşılan bir durumdur. Robot, park etmesi için belirlenen hedef noktaya ulaştığında konumu ve yönelimi minimum hataya sahip olmalıdır. Yerel planlayıcı algoritmalarını test etmek için, gerçek dünya park senaryosunu şekil 1.a'daki ortamda gerçekleştirdik ve işlemi on kez tekrarladık. Bu yaklaşım, test edilen algoritmaların güvenilir bir analizini sağlamak ve test süreci sırasında ortaya çıkmış olabilecek potansiyel hatalar veya anomalileri hafifletmek için uygulanmıştır. Tablo 2, deneyde kullanılan yerel planlayıcı algoritmalarının parametrelerini sunmaktadır.

Bu senaryoda, yerel planlayıcı algoritmaların performansı; toplam mesafe, toplam zaman, x- ve y-eksenindeki pozisyon hataları ve yaw-eksenindeki yönelim hataları karşılaştırılarak değerlendirilmiştir. TEB algoritmasının kullanıldığı durumlarda, robotun park işlemini tamamlaması ortalama olarak 16.5 saniye sürmüştür ve bu süre zarfında robot yaklaşık 1.88 metrelik bir yol kat etmiştir. Öte yandan, DWA algoritmasının kullanıldığı durumlarda, park işlemi ortalama olarak 59.5 saniye sürmüştür ve bu süreçte robot ortalama olarak 2.33 metre yol almıştır. Şekil 6'da ilgili grafikler görülebilir. Bu uzun süre, çoğunlukla

DWA algoritmasının hedef noktaya ulaştıktan sonra aracı hassas bir şekilde istenilen yönelimde konumlandırması için gereken ek süreden kaynaklanmaktadır. Pozisyon ve yönelim hatalarını bulmak için hedef noktanın konum bilgilerinden, robotun park işlemini tamamladığı andaki konum bilgilerini çıkarıp mutlak değerini aldık. Robot, konum bilgilerini LiDAR sayesinde elde etmiştir, bu yüzden ufak bir hata payını unutmamak gerekir. Her iki algoritmada da robot başarıyla istenilen hedef noktaya ulaşsa da tablo 1'de gösterilen x- ve y-eksenindeki pozisyon hatalarına daha dikkatli bakıldığında, TEB algoritmasının doğruluk ve hassasiyet açısından DWA algoritmasını geride bıraktığı görülür. Algoritma performansındaki farklar, yönelim hatalarını karşılaştırırken daha belirgin hale gelir. Tablo 1'de gösterildiği gibi, DWA algoritmasının, yaw-eksenindeki yönelim hatası test 1, test 4 ve test 6'da 1 derecenin altında olsa dahi, diğer testlerde çoğunlukla 50 derecenin üstündedir. 52.9 derece ortalama hata payına ve 76.3 derece standart sapmaya sahip olan DWA algoritması, yönelim konusunda çok güvenilir değildir. Buna karşılık olarak, analizimiz, TEB algoritmasının yaw-ekseni yönelim hatalarının 10. test haricinde 1 derecenin altında kaldığı gösterir. 0.41 derece standart sapmaya ve 0.50 derece ortalama hata payına sahip olan TEB algoritması, yönelim bakımından da üstün performans gösterir.

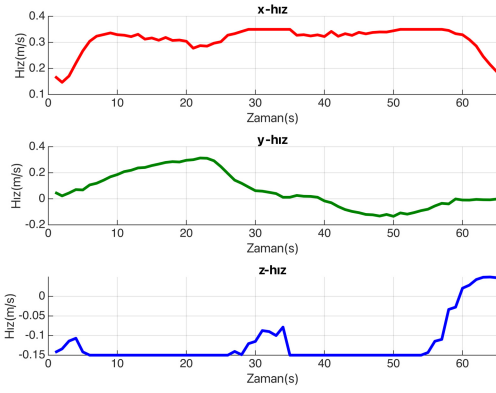
Tartışılan tüm faktörler göz önüne alındığında, park senaryosu için TEB algoritmasının, DWA algoritmasına göre daha üstün bir performans sergilediği açıktır.

Tablo 2: Seçilmiş parametreler

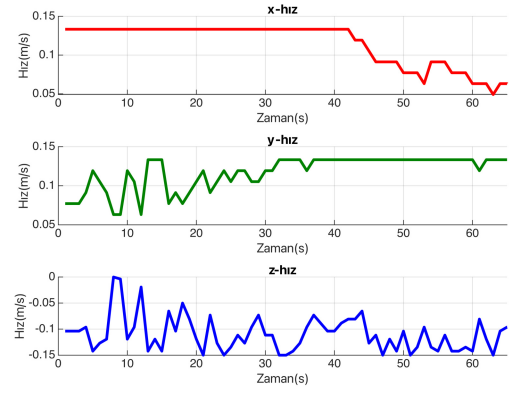
| Parametreler       | Değerler (TEB/DWA) | Tanım   |
|--------------------|--------------------|---|
| max_vel_x          | 0.35 / 0.35        | Mutlak maksimum öteleme hızı (metre/saniye)                 |
| min_vel_x          | 0.35 / -0.35       | Mutlak minimum öteleme hızı (metre/saniye)                  |
| max_vel_y          | 0.35 / 0.35        | Maksimum yanıl hareket hızı (metre/saniye)                  |
| max_vel_theta      | 0.15 / 0.20        | Maksimum açısal hız (radyan/saniye)                         |
| acc_lim_x          | 0.10 / 0.80        | Mutlak maksimum öteleme ivmesi (metre/saniye <sup>2</sup> ) |
| acc_lim_y          | 0.10 / 0.80        | Maksimum yanıl hareket ivmesi (metre/saniye <sup>2</sup> )  |
| acc_lim_theta      | 0.01 / 0.80        | Maksimum açısal ivme (radyan/saniye <sup>2</sup> )          |
| xy_goal_tolerance  | 0.01 / 0.01        | Hedef konuma izin verilen maksimum Öklit mesafesi (metre)   |
| yaw_goal_tolerance | 0.01 / 0.01        | İzin verilen maksimum yönelim hatası (radyan)               |

### 3.3. Senaryo II - Dönme Testi

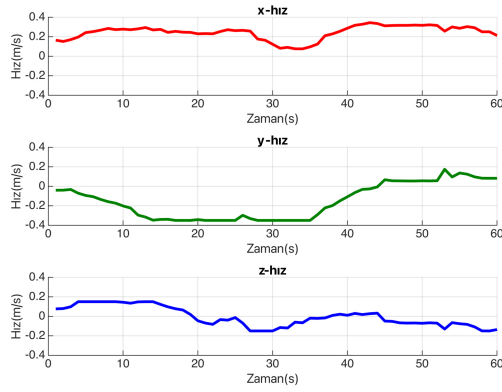
Dönüş senaryoları, mobil robotun hedef noktaya ulaşmak veya yolunu tıkayan engellerden kaçınmak için hareket yönünü değiştirmesi gereken durumları içerir. Mobil robotlar için güvenli ve çarpışmasız dönüşler yapabilmek yeteneği, özellikle endüstriyel ortamlar, kamusal alanlar ve depolar gibi, robotun engel-



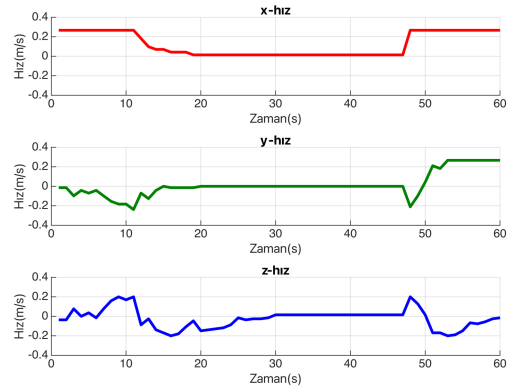
(a) TEB algoritması kullanılan dönüş senaryosunda (senaryo II) robotun hız değişimi.



(b) DWA algoritması kullanılan dönüş senaryosunda (senaryo II) robotun hız değişimi.



(c) TEB algoritması kullanılan dinamik engel senaryosunda (senaryo IV) robotun hız değişimi.



(d) DWA algoritması kullanılan dinamik engel senaryosunda (senaryo IV) robotun hız değişimi.

Şekil 5: TEB ve DWA algoritmaları kullanılan dönüş (senaryo II) ve dinamik engel (senaryo IV) senaryolarında robotun hız değişimi.

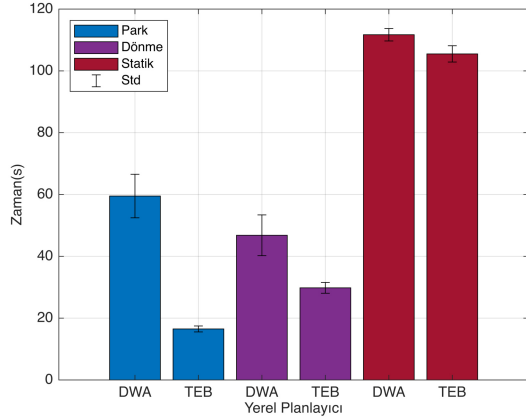
lere çarpmadan engellerin etrafında dolaşmaları gereken yerlerde hayati bir yetenektir. Bu tür yerler, sınırlı dönüş alanlarına sahip oldukları için mobil robotları dönüş manevrası yaparken zorlarlar. Eğer dönüş manevraları güvenli ve doğru bir şekilde gerçekleştirilmezse, bu durum mobil robotun kendi yapısına, yayalara, çevreye veya taşınan mallara zarar vermesi gibi ciddi sonuçlara sebebiyet verir.

Yerel planlayıcı algoritmalarının hassasiyetini doğrulamak için, şekil 1.b'deki ortamda gerçek dünya dönüş senaryosunu gerçekleştirdik ve işlemi on kez tekrarladık. Tablo 2, deneyde kullanılan yerel planlayıcı algoritmalarının parametrelerini göstermektedir. Hedef toleransları, yalnızca dönüş manevralarına odaklanmak istediğimizden dolayı x- ve y-ekseni için 0.5 metreye ve yaw-ekseni için 6.283 radyana ayarlanmıştır. Araç tüm-yönlü olduğu için, yaw hedef toleransı 360° olarak belirlenmiştir. Bu düzenleme, dönüş senaryosu testinin hassasiyetine odaklanmak için yapılmıştır. Deneylerimizde, yerel planlayıcı algoritmalarının performansını değerlendirirken, toplam mesafe ve zaman, x, y ve z eksenlerindeki hız değişimlerini ve robotun dönüş manevrası sırasındaki davranışını kapsamlı bir şe-

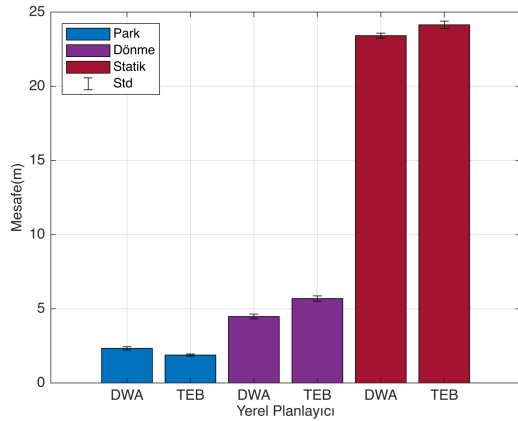
kilde karşılaştırdık ve analiz ettik. Sonuçlar, hem TEB hem de DWA algoritmalarının dönüş manevrası sırasında çarpışmalardan başarıyla kaçındığını gösteriyor. Ancak, Şekil 6.a ve 6.b'de gösterildiği gibi, TEB algoritması daha kısa toplam süreye sahipken, DWA algoritması, robotun dönüş sırasındaki hareketi nedeniyle daha küçük toplam mesafeye sahiptir. Bu bulgular, bu iki algoritma için zaman ve mesafe optimizasyonu hakkında bilgi verir. Analizimiz, beklendiği üzere, TEB algoritmasının zaman-optimal yolu önceliklendirdiğini, DWA algoritmasının ise mesafe-optimal yolu önceliklendirdiğini göstermektedir. Robotun x, y ve z eksenlerindeki hızları sırasıyla Şekil 5.a ve 5.b'de gösterilmektedir. İki algoritma arasındaki dikkat çekici fark, DWA algoritmasında gözlenen ani hız geçişleridir. TEB, DWA'ya göre daha yumuşak bir hız geçişi sergiler. Şekil 4.a ve şekil 4.b'de sırasıyla TEB ve DWA algoritmalarını kullanan robotun dönüş esnasındaki hareketi gösterilmiştir. DWA algoritmasında robotun çapraz bir şekilde hareket ettiği görülebilir, Şekil 5.b'deki ani hız değişimleri robotun bu hareketinden kaynaklanmaktadır. TEB algoritmasında ise robot, hedef noktaya ön tarafını denk getirerek hareket eder ve bu yüzden Şekil

5.a'da ani hız değişimleri yoktur.

Tartışılan tüm yöner dikkate alındığında, algoritmaların ikisi de dönüş senaryoları için kullanılabilir olsa da, DWA algoritmasının aksine, TEB algoritması robotta ani hız değişimlerine sebep olmadığı için DWA'ya kıyasla daha iyi bir seçenek olduğu söylenebilir.



(a) Robotun dört farklı test senaryosunu tamamlamak için aldığı toplam süre.



(b) Robotun dört farklı test senaryosunu tamamlamak için aldığı toplam mesafe.

Şekil 6: Robotun dört farklı test senaryosunu tamamlamak için aldığı toplam zaman ve toplam mesafe.

### 3.4. Senaryo III - Statik Engel

Senaryo III'te, algoritmaların statik engelleri aşma performansını test ettik. Mobil robotların, özellikle endüstriyel ortamlar, halka açık alanlar ve depolar gibi statik engellerin etrafında dolaşmaları gereken yerlerde, güvenli bir şekilde seyahat etme ve çarpışmalardan kaçınma yeteneğine sahip olmaları gereklidir. Senaryo II'de bahsedilen sorunların aynısı statik engel testi için de söylenilebilir.

Bu senaryo şekil 1.c'de gösterilen ortamda gerçekleştirilmiştir. Şekil 2'deki 0 noktası ITU ODV'nin başlangıç noktasını, 1-2-3-4 noktaları ise sırasıyla dört ardışık hedef noktayı

gösterir. Robot, koridor boyunca statik engellerden kaçınarak yol alıp sırasıyla hedef noktalara ulaşmakla görevlendirildi. Yerel planlayıcı algoritmalarının hassasiyetini analiz etmek için, gerçek dünya statik engel senaryosunu on kez gerçekleştirdik. Bu senaryoda, robotun statik engellerin bulunduğu bir ortamdaki navigasyonuna odaklanmak istediğimiz için, tablo 2'deki hedef toleranslarını x- ve y-ekseni için 0.5 metreye, yaw-ekseni için ise 6.283 radyana ayarladık. Diğer parametreler tablo 2'de gösterilen parametreler ile aynıdır. Yerel planlama algoritmalarının performansını değerlendirmek için, Şekil 6.a ve 6.b'de gösterildiği gibi toplam mesafe ve toplam zaman metriklerinin karşılaştırılmasına odaklanan deneyler gerçekleştirdik. Şekil 6.a ve 6.b'de gösterildiği gibi, DWA algoritması testi tamamlamak için yaklaşık 113 saniye gerektirdi ve bu süre zarfında robot toplamda yaklaşık 22 metre mesafe kat etti. Buna karşılık, TEB algoritması testi yaklaşık 105 saniye içinde tamamladı ve robot bu süre zarfında toplamda yaklaşık 24 metre mesafe kat etti. Ayrıca, İkinci Senaryo'daki bulgularımıza uygun olarak, TEB ve DWA algoritmaları için zaman ve mesafe optimizasyonu konusunda benzer sonuçlara ulaştık.

Tartışılan hususlar dikkate alındığında, her iki algoritma da statik engel senaryosu için uygundur. Ancak, TEB algoritması zamanı optimize etmek için daha uygunken, DWA algoritması mesafeyi optimize etmek için daha uygundur.

### 3.5. Senaryo IV - Dinamik Engel

Mobil robot navigasyonunun kritik bir parçası olan dinamik engelden kaçınma, robotun yolculuğu sırasında karşılaştığı dinamik engellerle baş etme durumlarını içerir. Temel hedef, robotun yolu üstünde karşılaştığı dinamik engelleri başarılı bir şekilde aşıp, yoluna devam etmesidir. Mobil robotlar otonom olarak hareket ettiği alanların çoğunda çok sayıda dinamik engelle karşılaşabilmektedir. Robot, bu dinamik engellerle başarılı bir şekilde mücadele edemezse, hedeflenen son noktaya ulaşması mümkün olmayacaktır.

Deneyel değerlendirmemizde, yerel planlayıcı algoritmalarının performansını, dinamik engellere karşı gösterdikleri tepkilere göre değerlendirdik. Şekil 4.c ve 4.d, sırasıyla TEB ve DWA algoritmalarının dinamik engellerle karşılaştığındaki tepkilerini görsel olarak gösterir. Yaptığımız testte, robot TEB algoritmasını kullandığı zaman, x-ekseni boyunca hareket ederken yolu üstünde bir dinamik engelle karşılaştığı zaman y-ekseni hızını artırarak dinamik engelden kaçıp, yoluna devam ettiğini gördük. Aksine, DWA algoritması, yine aynı koşullarda bir dinamik engelle karşılaştığında durur ve engel, robotun gitmesi gereken yoldan çekildikten sonra robot yoluna devam eder. Bu belirgin yaklaşım farkı, robotların hız değişimlerini gösteren Şekil 5.c ve 5.d'ye bakılarak da görülebilir. Şekil 5.c'nin incelenmesi, TEB algoritması çerçevesinde, robotun yaklaşık her 10 saniyede bir dinamik bir engelle karşılaştığını ortaya koyar. Bu karşılaşmaları yönetirken, robot, y-ekseni hızında bir artış, z-ekseni hızında bir azalış ve neredeyse sabit bir x-ekseni hızını sürdüren bir strateji kullanır. Şekil 5.d, robotun DWA algoritması altındaki davranışını gösterir. Bu, robotun yaklaşık 12. saniyede bir dinamik engelle karşılaştığını ve çarpışmayı önlemek için durduğunu gösterir, x, y ve z eksenlerindeki hızlarını 33 saniye süresince sifira indirir. Dinamik engel yolunu temizledikten sonra, robot 45. saniye civarında yoluna devam eder.

Yukarıda belirtilen faktörler dikkate alındığında, TEB algoritmasının dinamik engellerden başarılı bir şekilde kaçınıp, yoluna devam ettiğini söyleyebiliriz. Öte yandan, DWA algoritması, dinamik engel aşma konusunda tam anlamıyla başarı gösteremese de, robot ile engelin çarpışmasını engellediğini ve engel robotun yolundan çekildiğinde robotun yoluna devam ettiğini söyleyebiliriz.

#### 4. Sonuç

Bu çalışmada, tümyönlü bir mobil robot kullanılarak TEB ve DWA yerel planlayıcıları arasında ayrıntılı ve kapsamlı bir karşılaştırmalı analiz yapılmıştır. Park manevrası, dönüş manevrası, statik engellerin etrafında navigasyon ve dinamik engelden kaçınma olmak üzere dört farklı gerçek dünya senaryosu dikkatlice tasarlanmış ve gerekli analizler yapılmıştır. Bu titiz testler ve sonuçların sistemli analizi, TEB yerel planlayıcının, DWA yerel planlayıcıya kıyasla tümyönlü mobil robotların yönlendirilmesi konusunda belirgin bir avantaj sağladığını ortaya koymuştur. Bu üstünlük, özellikle TEB yerel planlayıcının dinamik engellerden becerikli bir şekilde kaçınmasından ve zaman optimizasyonu konusundaki başarısından kaynaklanır. Bu nedenle, bu araştırma, tümyönlü mobil robotlar için yerel planlama algoritmalarının nadir ve kapsamlı bir karşılaştırmasını sunarak, robotik alanında önemli bir dönüm noktası olarak durmaktadır. Sonuçlarının etkileri derindir ve hem çağdaş uygulamalara hemen faydalar sağlar, hem de gelecekteki araştırmalar için umut verici yollar sunar. Bu başarılar sayesinde, sürekli evrimleşen otonom navigasyon sistemlerinin hikayesine anlamlı bir sayfa eklemiş bulunmaktayız.

#### 5. Kaynakça

- [1] B. Cybulski, A. Wegierska, G. Granosik, Accuracy comparison of navigation local planners on ros-based mobile robot, in: 2019 12th International Workshop on Robot Motion and Control (RoMoCo), IEEE, 2019, pp. 104–111.
- [2] C. Roesmann, W. Feiten, T. Woesch, F. Hoffmann, T. Bertram, Trajectory modification considering dynamic constraints of autonomous robots, in: ROBOTIK 2012; 7th German Conference on Robotics, 2012, pp. 1–6.
- [3] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEE Robotics and Automation Magazine 4 (1997) 23–33, first proposition for <b>DWA</b>. doi:10.1109/100.580977.
- [4] S. Quinlan, O. Khatib, Elastic bands: Connecting path planning and control, in: [1993] Proceedings IEEE International Conference on Robotics and Automation, IEEE, 1993, pp. 802–807.
- [5] R. Amsters, P. Slaets, Turtlebot 3 as a robotics education platform, in: Robotics in Education: Current Research and Innovations 10, Springer, 2020, pp. 170–181.
- [6] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566), Vol. 3, IEEE, 2004, pp. 2149–2154.
- [7] I. Naotunna, T. Wongratanaphisan, Comparison of ros local planners with differential drive heavy robotic system, International Conference on Advanced Mechatronic Systems, ICAMEchS 2020-December (2020) 1–6. doi:10.1109/ICAMECHS49982.2020.9310123.
- [8] M. Pittner, M. Hiller, F. Particke, L. Patino-Studencki, J. Thielecke, Systematic analysis of global and local planners for optimal trajectory planning, in: ISR 2018; 50th International Symposium on Robotics, 2018, pp. 1–4.
- [9] S. K. Min, R. Delgado, W. C. Byoung, Comparative study of ros on embedded system for a mobile robot, Journal of Automation, Mobile Robotics and Intelligent Systems (2018) 61–67.
- [10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al., Ros: an open-source robot operating system, in: ICRA workshop on open source software, Vol. 3, Kobe, Japan, 2009, p. 5.
- [11] L. Yu, H. Wu, C. Liu, H. Jiao, An optimization-based motion planner for car-like logistics robots on narrow roads, Sensors 22 (22) (2022) 8948.
- [12] S. K. Gehrig, F. J. Stein, Elastic bands to enhance vehicle following, in: ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585), IEEE, 2001, pp. 597–602.
- [13] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, T. Bertram, Trajectory modification considering dynamic constraints of autonomous robots, in: ROBOTIK 2012; 7th German Conference on Robotics, VDE, 2012, pp. 1–6.
- [14] C. Rösmann, F. Hoffmann, T. Bertram, Integrated online trajectory planning and optimization in distinctive topologies, Robotics and Autonomous Systems 88 (2017) 142–153.
- [15] M. Deshpande, N. Kamalanathan, Behavioral analysis of ros motion planners integrated with robotics middleware framework (rmf), in: 2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), IEEE, 2022, pp. 196–199.
- [16] R. Wiki, teb\_local\_planner - ROS wiki, [http://wiki.ros.org/teb\\_local\\_planner](http://wiki.ros.org/teb_local_planner).
- [17] P. Marin-Plaza, A. Hussein, D. Martin, A. d. l. Escalera, Global and local path planning study in a ros-based research platform for autonomous vehicles, Journal of Advanced Transportation 2018 (2018) 1–10.
- [18] O. Brock, O. Khatib, High-speed navigation using the global dynamic window approach, in: Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C), Vol. 1, IEEE, 1999, pp. 341–346.
- [19] P. Saranrittichai, N. Niparnan, A. Sudsang, Robust local obstacle avoidance for mobile robot based on dynamic window approach, in: 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, IEEE, 2013, pp. 1–4.
- [20] R. Wiki, dwa\_local\_planner - ROS wiki, [http://wiki.ros.org/dwa\\_local\\_planner](http://wiki.ros.org/dwa_local_planner).