

OTONOM ELEKTRİKLİ BİR ARACIN YÖRÜNGE TAKİP PERFORMANSININ DEĞERLENDİRİLMESİ

Muhammet Can TÜRÜT,^{1,a)} Levent MALGACA,^{1,b)} ve Aytaç GÖREN^{1,c)}

¹Dokuz Eylül Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği Bölümü 35390, İZMİR

^{a)} muhammetcan.turut@deu.edu.tr, ^{b)} levent.malgaca@deu.edu.tr, ^{c)} aytac.goren@deu.edu.tr,

Otonom elektrik araçların yörünge kontrolünde çeşitli algoritmalar ve denetleyici tipleri kullanılır. Bu çalışmada diferansiyel tahrikli bir aracın üç farklı kontrol algoritması ile yörünge takibinin performansı değerlendirilmiştir. Değerlendirilen denetleyiciler sırasıyla Saf Takip, PID ve Lyapunov temelli bir denetleyicidir. Aracın takip etmesi için seçilen yörünge, sabit yarıçapa sahip bir çember olarak seçilmiştir. Seçilen bu yörüngeyi araç sabit bir doğrusal hızla takip etmiştir. Benzetim için Matlab/Simulink programları ve Robotics System Toolbox, Simscape ve Simscape Multibody kütüphaneleri kullanılmıştır. Yapılan analizlerde seçilen üç denetleyici de belirli hata toleransları ile referans yörüngeyi takip etmiştir. Çalışmada elde edilen sonuçlara göre RMS yönelim hatası ve ortalama sapma değeri en düşük olan denetleyici Lyapunov temelli denetleyici ve en yüksek olan denetleyici PID denetleyicidir

Anahtar Kelimeler: Otonom elektrikli araç, yörünge takibi, Saf Takip, PID, Lyapunov

EVALUATION OF THE PATH FOLLOWING PERFORMANCE OF AN AUTONOMOUS ELECTRIC VEHICLE ABSTRACT

Different algorithms and controllers are used in the path control of autonomous electric vehicles. In this study, the performance of the path following of a differential drive vehicle was evaluated with three different control algorithms. The evaluated controllers are Pure Pursuit, PID and a Lyapunov based controller, respectively. The path chosen for the vehicle to follow is a circle with constant radius. The vehicle follows the path with a constant linear velocity. For the simulation, the Matlab/Simulink program and Robotics System Toolbox, Simscape, and Simscape Multibody libraries were used. In the analysis conducted, all three controllers followed the reference path with certain error tolerances. According to the results obtained in the study, the controller with the lowest RMS orientation error and average deviation value is the Lyapunov-based controller, and the controller with the highest is the PID controller.

Keywords: Autonomous electric vehicle, path following, Pure-Pursuit, PID, Lyapunov

1. GİRİŞ

Otonom yörünge takibi, araçların daha önce belirlenen bir yörüngeyi insan müdahalesine gerek kalmadan takip etmesidir. Yörünge takibi lojistik sektöründe hızlı şekilde gelişen akıllı depolar gibi sivil uygulamaların yanı sıra çeşitli askeri uygulamaları da mevcuttur. Kullanıldıkları alanlarda gerek personel güvenliği gerekse verimliliğin artması için araçların yörüngelerini en az sapma ile takip etmelerini gerektirmektedir. Bu durumdan dolayı farklı operasyon koşulları ve farklı taşıtlar için uygun kontrol algoritmalarının tasarlanmasını gerektirmektedir.

Diferansiyel tahrikli araçlar kinematik modelleri kabul görmüş literatür notasyonunda açıklanmıştır. [1-3] araç modeline ait kinematik denklemleri açıklar.

Geometrik bir yaklaşım olan Saf Takip algoritması için yapılan ilk çalışmalardan biri olan [4] algoritmanın temel özelliklerini detaylı bir şekilde çalışmıştır. Daha sonra yapılan çalışmalarda Saf takip ile doğrultu hatası oranı [5] incelemiştir.

Yapılan bir çalışmada, PID denetleyici basit yapısı ve düşük hesaplama gereksinimleri nedeniyle düşük güçlü sistemlerde etkili olduğu gösterilmiştir.[6]

Micaelli, A., & Samson tarafından yapılan çalışmada tek tekerlekli ve iki direksiyonlu mobil robotlar için Lyapunov temelli bir yörünge takibi yöntemi önerilmiş ve bu yaklaşımın non-holonomik sistemlerde asimptotik kararlık sağladığı gösterilmiştir [7]. [8] ise otonom robotlar için yol takibi stratejilerini incelemiş, Lyapunov temelli kontrolün yüksek doğruluk sunduğunu belirtmiştir.

Bu çalışmada diferansiyel tahrikli bir araç modelinin yörünge takibi için kontrol algoritmaları incelenmiştir. Kontrol algoritmaları olarak PID denetleyici, Saf Takip ve Lyapunov temelli bir denetleyici kullanılmıştır. Denetleyicilerin analizleri Matlab/ Simulink programında Robotics Systems Toolbox ve Simscape-Simscape Multibody kütüphaneleri kullanılarak yapılmıştır.

2. ARAÇ MODELİ VE YÖRÜNGE DENKLEMİ

Bu bölümde araca ait kinematik denklemlerin türetilmesi, Simscape modelinin oluşturulması ve referans yörünge denklemleri açıklanmıştır.

Kinematik denklemler ile araca girdi olarak verilen çizgisel hız ve açısal hız değerleri ile aracın konum ve yönelimi açıklanır.

Modellenen aracın tekerlek yarıçapı 15 cm ve tekerlekler arası mesafe 40 cm'dir.

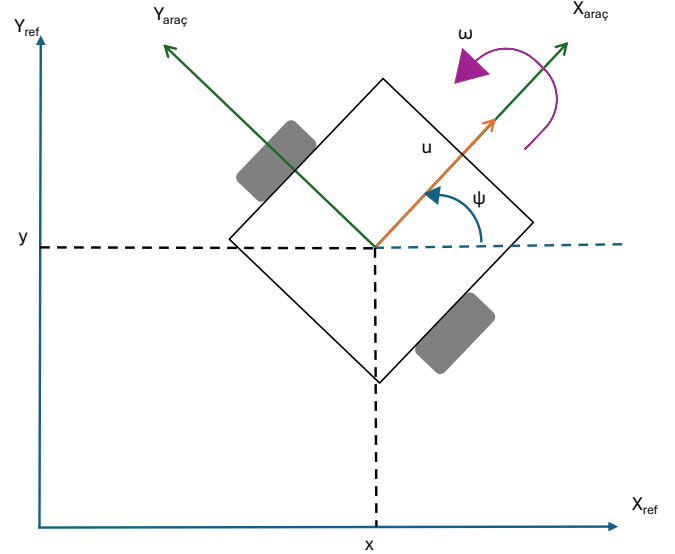
Aracın eyleyici modeli ve dinamiği bu çalışmada dikkate alınmamıştır. Denetleyiciden elde edilen açısal ve doğrusal hız değerleri araca anlık olarak etki ettirilmiştir.

2.1. Araç Kinematik Modeli

Bu çalışmada kullanılmak üzere diferansiyel tahrikli bir araç modeli seçilmiştir. Basitliği ve tekerlerin bağıl hızlarına hassasiyeti nedeniyle oluşan düşük yörünge takip hatası bu araç modelinin seçilme nedenleridir. Seçilen modelin non-holonomik özelliği göz önünde bulundurularak aracın aşağıda verilen matematik modeli çıkartılmıştır.

Global eksen takımının x ve y eksenleri sırasıyla, yatay ve düşey eksenlere gelecek şekilde, aracın yerel eksen takımı ise x eksen

aracın çizgisel hızı doğrultusunda olacak şekilde yerleştirilmiştir. Kullanılan eksen takımları Şekil 1.'de gösterilmiştir.



Şekil 1. Diferansiyel tahrikli araç modeli

Aracın global koordinat sistemindeki konum ve yönelim açısını içeren konum-yönelim vektörü (p_{global}) Denklem 1.'de verilmiştir.

$$p_{global} = \begin{Bmatrix} x \\ y \\ \psi \end{Bmatrix} \quad (1)$$

Denklem 1.'in zamana göre ilk türevi ile bulunan aracın çizgisel hız ve açısal hızı ise Denklem 2.'de verilmiştir.

$$\dot{p}_{global} = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{Bmatrix} \quad (2)$$

Aracın non-holonomik'liğinden kaynaklı aracın Y_{arac} eksenin doğrultusunda bir hızı olmayacaktır. Bu nedenle aracın yerel koordinat sistemine göre hız değerleri Denklem 3.'teki gibidir.

$$\dot{p}_{yerel} = \begin{Bmatrix} u \\ 0 \\ \omega \end{Bmatrix} \quad (3)$$

Çizgisel ve açısal hız ile global koordinat sistemindeki hızlar arasındaki ilişkiyi, θ yönelim açısı ve dönüşüm matrisi kullanarak Denklem 4.'te verildiği gibi yazılabilir.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} u \\ 0 \\ \omega \end{pmatrix} \quad (4)$$

Denklem 4. kullanılarak aracın kinematik denklemleri Denklem 5,6 ve 7 bulunur.

$$\dot{x} = u \cos \psi \quad (5)$$

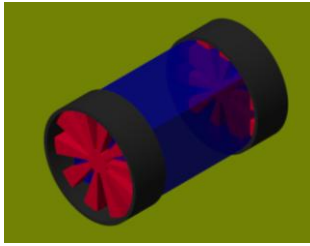
$$\dot{y} = u \sin \psi \quad (6)$$

$$\dot{\psi} = \omega \quad (7)$$

2.2. Araç Simscape Modeli

Simulink benzetiminde kullanılmak üzere aracın kinematik modeli özel bir ‘‘Simscape Component’’ olarak modellenmiştir. Bu modelde önceki başlıkta elde edilen Denklem 5,6 ve 7 kullanılarak bulunan araca ait hız değerlerinin integrali alınarak aracın x ve y eksenindeki konumları ve yönelimi bulunmuştur. Araç kinematik modeli çizgisel ve açısal hızın girdi, konum ve yönelimi ise model çıktısı olacak şekilde kodlanmıştır.

Bulunan konum bilgileri kullanılarak benzetimi görselleştirmek adına aracın Simscape Multibody çıkarılmıştır. Yapılan benzetimde aracın denge sağlaması gerekmediğinden dolayı araç sadece iki tekerlek kullanılarak modellenmiştir. Aracın modeli Şekil 2’dedir.



Şekil 2. Araca ait Simscape Multibody Modeli

2.3. Yörünge Denklemi

Takip edilen yörüngeye ait denklem bir açısal değişken olan γ değişkeni kullanılarak hesaplanmıştır. Açısal değişken γ , $[0,2\pi]$ aralığında sınırlıdır. Bu sınırlamanın nedeni yörüngeye sürekli çember şeklinde çalışması ihtiyacıdır.

Referans yörüngeye ait global koordinat sistemindeki X ve Y eksenlerindeki bileşenleri sırasıyla Denklem 8 ve 9 kullanılarak hesaplanır. Referans çember 5 m yarıçapında

bir çember olduğu için Denklem 8 ve 9 yarıçap ile çarpımıyla verilmiştir.

$$x = 5 \cos \gamma \quad (8)$$

$$y = 5 \sin \gamma \quad (9)$$

Lyapunov temelli denetleyici yörüngeye ait Denklem 10. ve 11.’de verilen γ ’ya göre birinci türevleri ve Denklem 12 ve 13’te verilen ikinci türevlerine de ihtiyaç duyduğu için referans yörünge bloğunun çıktılarına eklenmiştir.

$$\frac{dx}{d\gamma} = -5 \sin \gamma \quad (10)$$

$$\frac{dy}{d\gamma} = 5 \cos \gamma \quad (11)$$

$$\frac{d^2x}{d\gamma^2} = -5 \cos \gamma \quad (12)$$

$$\frac{d^2y}{d\gamma^2} = -5 \sin \gamma \quad (13)$$

PID denetleyici için gerekli olan yörüngeye ait yönelim açısını yörünge teğet vektörünün açısı olarak tanımlanır. Bu açı Denklem 14 ile hesaplanmıştır.

$$\theta = \text{atan2}(5 \cos \gamma, -5 \sin \gamma) \quad (14)$$

3. KONTROL ALGORİTMALARI

Tüm kontrol algoritmaları için analizler yapılmıştır. Referans yörüngeye bir ‘‘Matlab Function’’ bloğu kullanılarak kodlanmıştır.

Yapılan benzetimlerde aracın başlangıç koşulları Denklem 15’te verilmiştir.

$$p_{\text{başlangıç}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (15)$$

3.1. Saf Takip Algoritması

Saf Takip algoritması yaygın ve basit bir yörünge takibi yöntemidir. Bu yöntem geometrik bir hesaplama yapar. Bu algortmada araçtan ileri görüş mesafesi (l_d) adı verilen bir mesafe kadar uzakta ve yörünge üzerinde olan bir hedef konum seçilir. Seçilen bu konum ile aracın mevcut konumu arasında bir yay çizilir. Aracın çizilen yayı takip etmesi için gerekli olan yönelim açısı sürekli olarak hesaplanır.

Yönelim açısı aşağıdaki formülle hesaplanır:

$$\psi = \text{atan} \left(\frac{2L \sin \alpha}{l_d} \right) \quad (16)$$

Burada ψ aracın sahip olması gereken yönelim açısı, L iki teker arası mesafe, α araç yönelimi ile araçtan hedef konuma olan vektörün arasındaki açıdır.

Küçük ileri görüş mesafesi aracın yörüngeyi daha yakından takip etmesini ve dolayısıyla daha hızlı dönebilmesini sağlar ancak yüksek doğrusal hızlarda ilerleyen bir araç seçilen bu noktayı yeterince iyi takip edemez ve araç yol etrafında salınım yapar. Büyük bir ileri görüş mesafesi ise aracın yörüngeyi daha yumuşak bir şekilde takip etmesini sağlar ama yörüngeyi eğriliği arttıkça yörüngeyi takip etmekte zorlanabilir. İleri görüş mesafesi sabit veya hıza bağlı olmak üzere iki farklı şekilde seçilebilir. Sabit ve/veya yavaş hızlarda giden araçlarda sabit ileri görüş mesafesi sıklıkla kullanılırken, değişken hıza sahip araçlarda genellikle hıza bağlı değişken ileri görüş mesafesi kullanılır. Değişken ileri görüş mesafesi araç yüksek hızlara ulaştığında artarak yörünge takibinde stabilite sağlar.

Çalışmada kullanılan araç sabit doğrusal hızda çalışacağı için 0,5m'lik bir sabit ileri görüş mesafesi seçilmiştir.

Denetleyicinin çalışabilmesi için takip edilecek yörüngeye ait koordinatlar, aracın anlık konumu-yönelimi ve 5 m/s'lik sabit doğrusal referans hız değerleri denetleyiciye girildi verilmiştir. Aracın çizgisel ve açısal hızı bloğun çıktılarıdır.

3.2. PID Kontrol

PID denetleyici, yörünge yönelimi ve aracın anlık yönelimi arasındaki fark olarak tanımlanan yönelim hatasına bağlı olarak bir kontrol sinyali oluşturur. Hesaplanan yönelim hatası $[-\pi, \pi]$ aralığında olması için normalize edilir. Normalizasyon işlemi hata açısının çok yüksek sıçramalar yapmasını önlemek için yapılır. Normalizasyon işlemi Denklem 16. kullanılarak yapılmıştır.

$$e_\theta = \text{mod}(e_\theta + \pi, 2\pi) - \pi \quad (16)$$

Oluşturulan bu sinyal PID bloğunda işlenerek aracın direksiyon açısı belirlenir.

PID denetleyici benzetimi için katsayılar "PID Tuner" kullanılarak hesaplanmıştır. PID

Tuner'da yanıt süresi 0,2 s ve geçici davranış 0,9 olarak seçilmiştir. PID Tuner sistemi 7,5. saniyede karşılık gelen durumlar etrafında doğrusallaştırmıştır. Bulunan katsayılar sırasıyla $K_p=9,538$, $K_i=16,847$ ve $K_D=0,181$ 'dir.

3.3. Lyapunov Temelli Denetleyici

İlk olarak 1993 yılında Micaelli ve Samson tarafından tasarlanan bu denetleyicide amaç, araç ile yörünge arasında konum takip hatası ve yönelim hatası türetilip bu hataları doğrusal olmayan kontrol teknikleri ile sıfıra götürmektir.

Konum takip hatası, Denklem 17. kullanılarak hesaplanır.

$$e = R(\psi_y)(p - p_y) \quad (17)$$

Burada p aracın global koordinat sistemindeki konumu, p_d yörüngeye referans noktanın global koordinat sistemindeki konumu ve ψ_y referans noktanın yönelimidir. R matrisi, global eksen takımından referans eksen takımına dönüşüm matrisidir ve Denklem 18.'de verilmiştir.

$$R = \begin{bmatrix} \cos \psi_y & \sin \psi_y \\ -\sin \psi_y & \cos \psi_y \end{bmatrix} \quad (18)$$

Hesaplanan konum takip hatası s_1 ve y_1 olmak üzere iki bileşenden oluşur. Bu bileşenler sırasıyla iz yönü (along-track) ve iz dışı(cross-track) hata olarak adlandırılır.

Referans nokta, aracın merkezinin yol üzerindeki dik düşümü olarak seçilir ve böylelikle s_1 benzetim boyunca sıfır olur.

Aracın yönelim hatası ise aracın doğrultusundan referans noktanın doğrultusunun çıkarılması ile hesaplanır.

$$\psi_e = \psi - \psi_y \quad (19)$$

Aracın yönelim hatasını, aracın iz dışı hatasına bağlı olarak düzelten bir düzeltme fonksiyonu ile çıkartarak $\tilde{\psi}$ değeri Denklem 20. ile hesaplanır.

$$\tilde{\psi} = \psi_e - \delta(y_1, u) \quad (20)$$

Düzeltilme fonksiyonu δ Denklem 21.'de verilmiştir.

$$\delta = -\theta_0 \tanh(k_\delta y_1 u_h) \quad (21)$$

Bu denklemde θ_0 yanal sapma düzeltmesinin maksimum açısını belirler ve genellikle $0 < \theta_0 < \frac{\pi}{2}$ aralığında seçilir. k_δ düzeltme hızını ayarlayan pozitif bir kazanç katsayısı, u_h ise istenilen doğrusal hızdır.

Denklem 22’de verilen Lyapunov fonksiyonu y_1 ve $\tilde{\psi}$ kullanılarak tanımlanır.

$$V = \frac{1}{2}y_1^2 + \frac{1}{2}\tilde{\psi}^2 \quad (22)$$

Denklem 22’nin zamana göre türevi

$$\dot{V} = y_1\dot{y}_1 + \tilde{\psi}\dot{\tilde{\psi}} \quad (23)$$

Denklem 17 ve Denklem 19’un zamana göre birinci türevlerinin alarak Denklem 24 ve Denklem 25 bulunur.

$$\dot{y}_1 = u \sin(\psi_e) \quad (24)$$

$$\dot{\psi}_e = \omega - \kappa \frac{u \cos(\psi_e)}{1 - \kappa y_1} \quad (25)$$

Yörünge eğriliği κ ’nın hesaplanması için Denklem 26 kullanılmıştır.

$$\kappa(\gamma) = \frac{x'_y(\gamma)y''_y(\gamma) - x''_y(\gamma)y'_y(\gamma)}{\left(\sqrt{x'_y(\gamma)^2 + y'_y(\gamma)^2}\right)^3} \quad (26)$$

Düzeltilme fonksiyonu δ ’nın zamana göre türevi Denklem 27’de verilmiştir.

$$\dot{\delta} = -\theta_0 k_\delta \frac{1}{\cosh^2(k_\delta y_1 u_h)} u_h \sin(\psi_e) \quad (27)$$

Denklem 24 ve Denklem 25, Denklem 23’te yerine yazılarak Denklem 28. elde edilir

$$\dot{V} = y_1(u \sin(\psi_e)) + \tilde{\psi} \left(\omega - \kappa \frac{u \cos(\psi_e)}{1 - \kappa y_1} - \delta \right) \quad (28)$$

Denklem 28’nin sıfırdan küçük eşit olmasını sağlayacak açısız hız değeri Denklem 29’ ile hesaplanır.

$$\omega = \kappa u_r + \delta - k_1 \tilde{\psi} - k_2 y_1 u_d \frac{\sin \psi_e - \sin \delta}{\tilde{\psi}} \quad (29)$$

Referans noktanın hızı olan u_r Denklem 30 ile hesaplanır.

$$u_r = u \frac{\cos \psi_e}{1 - \kappa(\gamma)y_1} \quad (30)$$

Aracın çizgisel hızı Denklem 31’de verilmiştir.

$$u = u_h \quad (31)$$

4. BENZETİM SONUÇLARI

Saf Takip, PID ve Lyapunov temelli denetleyiciler için sonuçlar bu bölümde incelenmiştir.

Her bir denetleyiciye, aracın anlık konumu ve referans yörüngenin o an için hedeflenen konumu girdi olarak verilmiştir.

Benzetim Matlab/Simulink 2024b sürümü, Ode14x çözücüsü 0.001s sabit zaman adımıyla çözdürülmüştür.

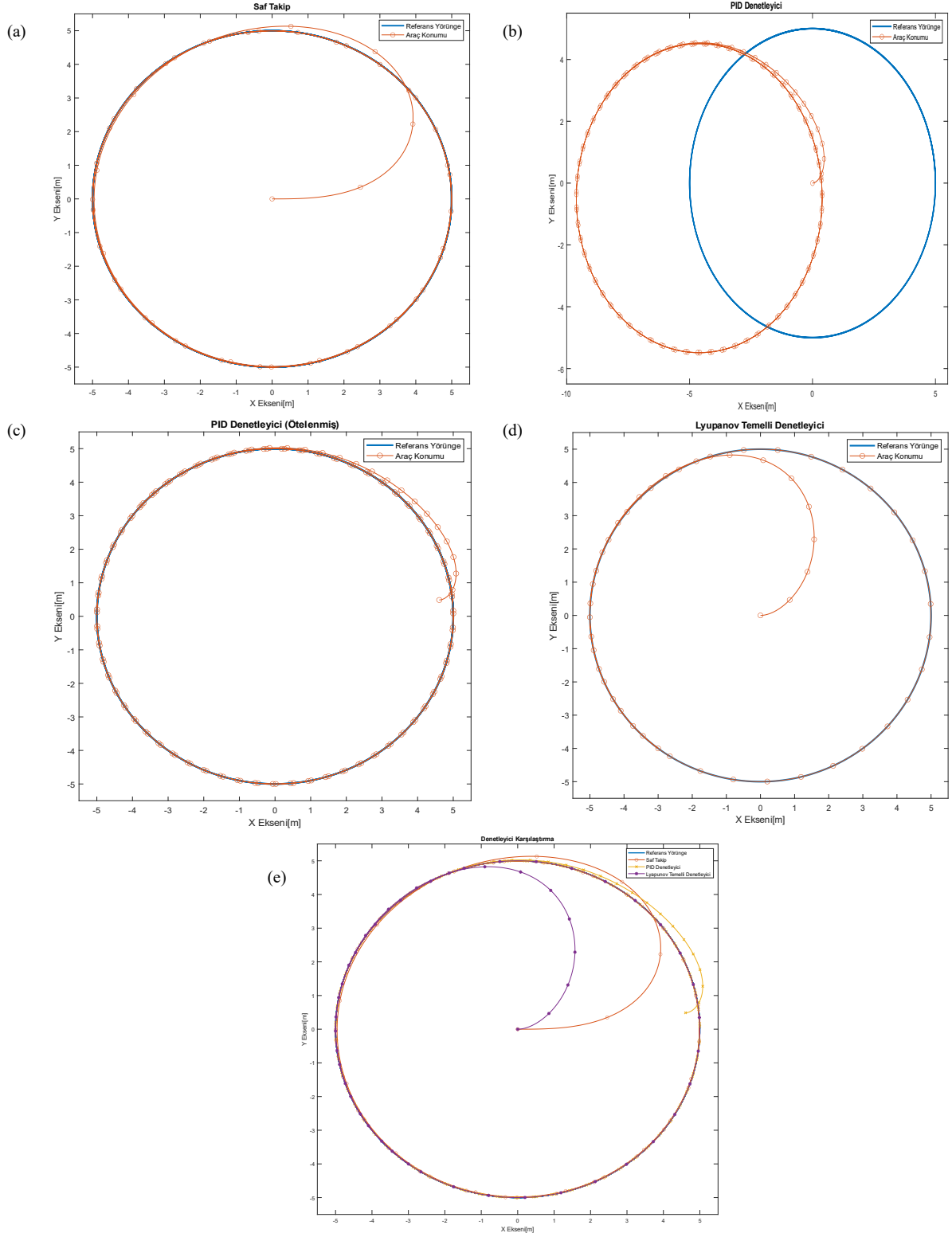
Denetleyiciler yörüngeden sapma mesafesi ile değerlendirilmiştir. Yörüngeden sapma mesafesi Denklem 32’de verilen iki nokta arası uzaklık formülü kullanılarak hesaplanır.

$$d_{sapma} = \sqrt{(x_y - x)^2 + (y_y - y)^2} \quad (32)$$

Saf Takip denetleyici, ileri görüş mesafesi 0,5 m, maksimum açısal hız 50 rad/s parametreleriyle çalıştırılmıştır. 15 saniye süren benzetim, 6,297 saniye de çözülmüştür.

PID denetleyici benzetimi 15 saniye sürmüştür ve 5,658 saniye de çözülmüştür. Yapılan PID denetleyici modelinde aracın hedef noktadan uzaklığı kontrol edilmeyip yalnızca aracın yönelimi kontrol edilmiştir.

Farklı kazanç kombinasyonlarıyla yapılan benzetim sonuçları ile araçta en iyi performansı veren Lyapunov temelli denetleyici parametreleri $k_\delta = 1$, $k_1 = 0,5$, $k_2 = 0,2$ ve $\theta_0 = \pi/4$ rad olarak seçilmiştir. Benzetim 29,932 saniyede çözülmüştür.

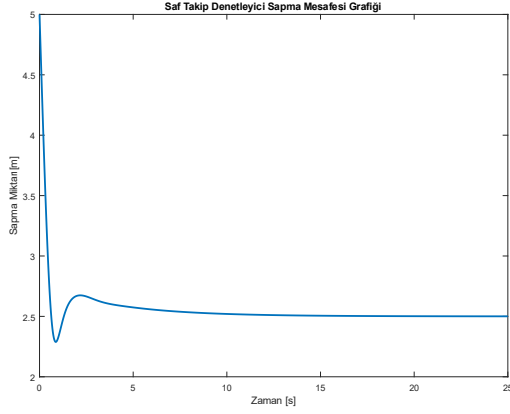


Şekil 3- Denetleyiciler için araç hareketi ve Referans yörünge a) Saf takip algoritması b) PID denetleyici c) PID denetleyici (Ötelenmiş), d) Lyapunov temelli denetleyici e) Denetleyici karşılaştırılması

Şekil 3 (a)'da verilen grafikte görüldüğü üzere Saf Takip başlangıçta, araç referans yola yaklaşırken hafif bir aşma sergilemekte, ama bu sapma hızlıca giderilmekte ve araç sonrasında

referans yörüngeyi doğru bir şekilde takip etmektedir.

Saf Takip algoritması ile sürülen aracın zamana göre sapma mesafesi Şekil 4'te verilmiştir.

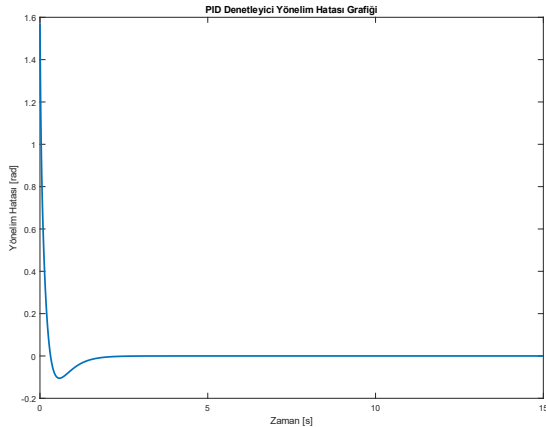


Şekil 4. Saf takip denetleyici için sapma mesafesi

Aracın son kararlı duruma geldiğinde sapma değerinin sıfıra yaklaşmamasının nedeni Saf Takip algoritmasında takip edilen hedef noktanın sürekli güncellenmesidir.

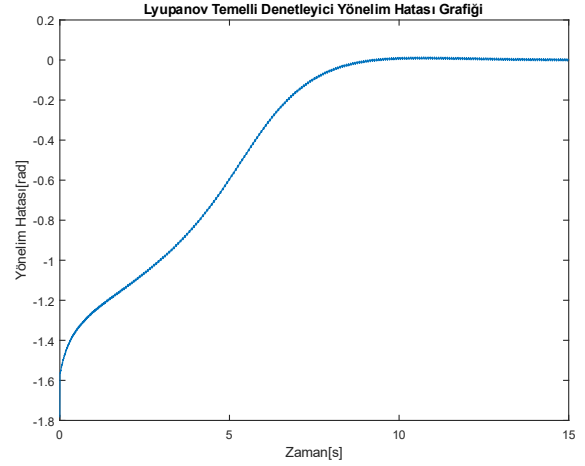
Tasarlanan PID denetleyicinin yönelim takibini yaptığını doğrulamak için hem yönelim hatası grafiği incelendi hem de elde edilen araç konumu yörünge koordinatlarına ötelendi. Ötelenmemiş araç konum ve referans konum grafiği Şekil 3(b)'de, ötelenen konum grafiği ise Şekil 3(c)'de verilmiştir.

Ötelenmiş yörünge takibinden görülebildiği üzere aracın hareketinin erken aşamalarında bir aşım görülmüştür ama bu aşım hızla sönümlenerek sistemin kararlı hale getirilmesi sağlanmıştır.



Şekil 5. PID denetleyici için yönelim hatası

Şekil 5'de görüldüğü üzere aracın doğrultu hatası Şekil 3(c)'de elde edilen sonucunda desteklediği üzere sıfıra gitmiştir.



Şekil 6. Lyapunov temelli denetleyici için yönelim hatası

Şekil 3(d)'de ve Şekil 6'da görüldüğü üzere Lyapunov temelli denetleyici sahip araç bir aşım değeri olmadan yörüngeye yaklaşmış ve Hareketi boyunca da yörüngeden sapmamıştır.

Şekil 3(e)'de bütün denetleyicilere ait araç hareketi grafiği birlikte çizdirilmiştir.

Denetleyicilerin performans kıyaslamasını yapmak için sapma değerlerinin ortalama ve RMS değerleri incelenmiştir. Hesaplanan değerler Tablo 1.'de verilmiştir.

Tablo 1. Ortalama Sapma ve RMS Yörünge Sapma Hataları

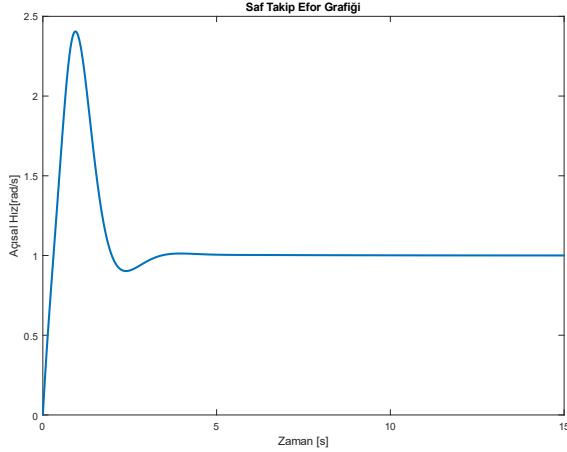
Denetleyici	Ortalama Sapma	RMS Yörünge Sapma
Saf Takip	2,5866 m	0,0212 m
PID	4,6528 m	0,1194 m
Lyapunov Temelli Denetleyici	0,3027 m	0,0095 m

Ortalama sapma değerleri kıyaslanarak denetleyicinin ne kadar verimli bir şekilde referans yörüngeyi takip ettiği hakkında bilgi edinilebilir. Bu değer düşük olması aracın düşük bir hatayla yörünge takibini sağladığı anlamına gelir. En düşük değer Lyapunov temelli denetleyici de hesaplanırken, en yüksek değer PID denetleyicide hesaplanmıştır. Bu beklenen bir sonuçtur çünkü PID denetleyici sapma hatasını değerlendirmeyip sadece yönelim hatasını değerlendirmektedir.

RMS değeri sapma mesafelerinin genliğini ve denetleyicinin tutarlılığı konusunda bir fikir verir. Bu değer kıyaslanarak denetleyicinin

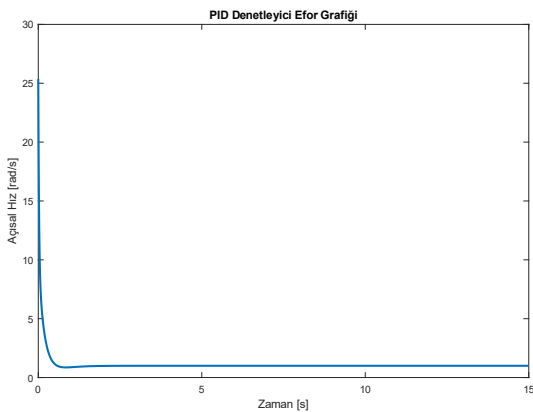
kararlılığı hakkında bir yorum yapılabilir. Düşük değer Lyapunov temelli denetleyicide en yüksek değer ise PID denetleyicide hesaplanmıştır.

Denetleyicilerin kontrol eforlarının incelenmesi, denetleyici sinyallerinin gerçekçi olup olmadığı hakkında bize bir fikir verir. Şekil 7,8 ve 9'da sırasıyla Saf Takip, PID ve Lyapunov temelli denetçilerin kontrol eforu grafikleri verilmiştir.



Şekil 7. Saf takip denetleyici için kontrol eforu

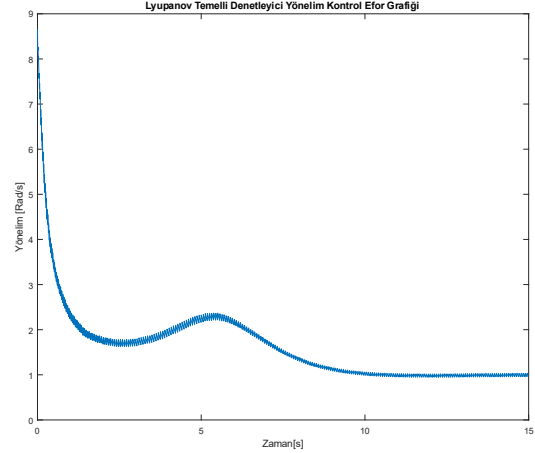
Şekil 7 ve Şekil 3(a)'da görüldüğü üzere araç öncelikle yüksek bir açısal hıza çıkarak bir aşım sergiledikten sonra düzenli rejime gelmiştir. Düzenli rejime değerine gelene kadar araçta ani keskinliklerle bir hızlanma/yavaşlanma gibi gerçek sistemlerin ataletleri nedeniyle yapamayacağı hareketler yerine sürekli bir hareketin meydana gelişi kontrol sinyalinin gerçekçi olduğu destekler.



Şekil 8. PID denetleyici için kontrol eforu

Şekil 8'de görüldüğü üzere aracın başlangıç anında yüksek bir yönelim hatasına sahip olduğu için yüksek bir açısal hız ihtiyacıyla hareketine başlamıştır. Şekil 3(c)'de de bu durum aracın diğer denetçilerden daha hızlı bir

dönüş yaptığından da anlaşılabilir. Açısal hız değerinin hızlı bir şekilde salınım yapmadan düzenli rejim ulaşması denetçinin gerçekçi bir yönelim kontrolü sağladığını gösterir.



Şekil 9. Lyapunov temelli denetleyici için kontrol eforu

Lyapunov temelli denetleyici de araçta PID denetleyiciye benzer şekilde yüksek bir hata miktarı ile başlayıp bu hatanın giderilmesi için yüksek bir açısal hızla harekete başlar.

5. SONUÇLAR

Bu çalışmada, otonom elektrikli bir araç için yörünge takibi üç farklı denetleyici kullanılarak yapılmıştır. Araç 5 m yarıçaplı çember bir referans yörüngeyi 5 m/s sabit bir doğrusal hızla takip etmiştir. Yapılan analizlerde Saf takip algoritması ve Lyapunov temelli denetleyici hem konum hem de yönelim hatalarında istenilen sonucu verirken, PID denetleyici sadece yönelim hatasında istenilen sonucu vermiştir.

PID denetleyicinin konum hatasını sıfırlamasını sağlaması için sistemde sadece yönelim hatası ile çalışan bir PID bloğu yerine sırasıyla konum hatası ve yönelim hatası ile çalışan iki ayrı PID denetleyici kullanılarak çözülebilir.

Bu çalışmada kullanılan PID Tuner yerine kök yer eğrileri kullanılarak PID denetleyicinin katsayıları elde edilirse yönelim hatası azalır.

Benzetim çözüm süresi en yüksek olmasına rağmen, ortalama sapma değeri açısından PID denetleyiciden 15,37 kat, Saf takip denetleyicisinden 8,54 kat; RMS yörünge sapma değeri açısından PID denetleyiciden 12,57 kat, Saf takip denetleyicisinden 2,23 kat daha iyi performans gösterdiğinden Lyapunov

temelli denetleyicinin en etkili denetleyici olduğu söylenebilir.

6. KAYNAKLAR

[1] Sigar, R. I., Nourbakhsh, R., Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots, Second Edition.*, The MIT Press, Cambridge/Massachusetts.

[2] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W. (2005). *Principles of Robot Motion.*, The MIT Press, Cambridge/Massachusetts.

[3] Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. (2009). *Robotics: Modelling, planning and control.*, Springer, London

[4] Coulter, R. C. (1992). *Implementation of the Pure Pursuit Path Tracking Algorithm*, Carnegie Mellon University, Robotics Institute.

[5] Lihui Wang, ZongLiang Chen, Wenxing Zhu (2022); *An improved pure pursuit path tracking control method based on heading error rate*. Industrial Robot 30 June 2022; 49 (5): 973–980. <https://doi.org/10.1108/IR-11-2021-0257>

[6] Jain, H., & Babel, P. (2024). A Comprehensive Survey of PID and Pure Pursuit Control Algorithms for Autonomous Vehicle Navigation. *ArXiv, abs/2409.09848*.

[7] Micaelli, A., Samson, C. (1993). *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots*, INRIA.

[8] Hung, N.Q., Rego, F.F., Quintas, J., Cruz, J., Jacinto, M., Souto, D., Potes, A., Sebastião, L., & Pascoal, A.M. (2022). A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments. *Journal of Field Robotics*, 40, 747 - 779. <https://doi.org/10.1002/rob.22142>