

Otonom Araçlarda Clothoid Tabanlı Lineer Zamanla Değişen Model Öngörülü Kontrol

Clothoid Based Linear Time-Varying Model Predictive Control In Autonomous Vehicles

Mustafa Caner Sezer¹, Leyla Gören Sümer²

¹Kontrol ve Otomasyon Mühendisliği Bölümü
İstanbul Teknik Üniversitesi, İstanbul
{sezermus}@itu.edu.tr

²Kontrol ve Otomasyon Mühendisliği Bölümü
İstanbul Teknik Üniversitesi, İstanbul
{goren1}@itu.edu.tr

Özetçe

Bu çalışmada, otonom araçlardaki temel problemlerden rota planlama ve kontrol blokları ele alınmıştır. İlk aşamada aracın izleyeceği rota daha konforlu bir sürüş sağladığı için clothoidler kullanılarak yeniden oluşturulmuştur. Yeni rotada, haritadaki noktaların değerlendirilmesi yapılmış ve aynı karakteristiği taşıyan ardışık noktalar yerine direksiyon açısından değişimlerin olduğu dönüm noktaları ele alınmıştır. Yalnızca bu noktalar kullanılarak bilgi kaybı olmadan rota seyrekleştirilmiştir. Böylece hem clothoidler sayesinde daha konforlu hem de seyrekleştirme sayesinde aynı öngörü ufku ile daha uzun mesafenin kapsanabileceği bir rota elde edilmiştir. İkinci aşamada, bu rotanın izlenmesi adına bir kontrolör tasarlanmıştır. Bu kontrolör, hem lineer model öngörülü kontrolden farklı olarak sistemi birçok noktada lineerleştirilmesi ve dinamiği daha iyi temsil etmesi hem de lineer olmayan model öngörülü kontrolördeki yerel minimuma yakınsama, gerçek zamanlı çalışmama gibi olumsuzlukları içermemesi nedeniyle lineer zamanla değişen model öngörülü kontrolör olarak belirlenmiştir. Bu yaklaşım iki farklı rotada test edilmiş ve sonuçları gösterilmiştir.

Abstract

In this study, route planning and control for autonomous vehicles are discussed. Initially, the route is reconstructed using clothoids as it provides a more comfortable ride. In the new route, instead of using consecutive points with the same characteristic, the points where steering angle changes are considered. Therefore, the route becomes sparse without loss of information. Thus, a route that is more comfortable thanks to clothoids and that covers longer distances with the same horizon thanks to sparseness has been obtained. Next, a controller has been designed as a linear time-varying model predictive controller because it linearizes the system at many waypoints and represents

the dynamics better, unlike the linear model predictive control, and does not contain the disadvantages such as convergence to the local minimum or real-time problems in the non-linear model predictive controller. This approach has been tested in two different routes and the results are shown.

1. Giriş

Rota planlama ve kontrol, aracın konumu ve çevresini bildikten sonra sürüş için gerekli rotanın belirlenmesi ve bunun izlenmesidir. Bu iki ister için birçok çözüm geliştirilmiştir. Rota planlama problemi için, her adımda araca takip etmesi gereken bir rota iletilmelidir ve bu işlem gerçek zamanlı olabilmelidir. Buna göre getirilen çözümlerden ilki örneklem temelli rota planlama algoritmalarıdır. Örneklem temelli algoritmalarda başlangıç ve hedef noktaları arasında rastgele noktalar seçilir. Seçilen noktalar, önceki seçilen noktaların en yakın olanıyla birleştirilir ve böylece bir ağaç yapısı oluşmuş olur. Burada başlangıç ve hedef noktalarından başlayıp rastgele seçilen noktalarla oluşturulan ağaçların bir noktada birbirine yaklaşması ve birleştirilmesiyle rota planlanmış olur. Bu yöntemi çift yönlü rota arama algoritması denir [1]. Örneklem bazlı algoritmalar rastgele arama mantığıyla çalıştığı için buldukları yolun optimal olmasını garanti etmez. Fakat yine de, aramadaki düğüm sayısı arttıkça optimalliğe yaklaşırlar. Bu algoritmalara örnek olarak ağaç yapılarını rastgele ve anlık keşfeden RRT ve RRT* algoritmaları verilebilir. RRT algoritmasında alınan rastgele noktalar ağaç yapısındaki en yakın düğüm ile birleştirilir ve böylece harita rastgele düğümlerle büyür. Bu algoritmanın diğer bir versiyonu olan RRT* algoritmasında ise RRT'den farklı olarak seçilen noktalar en yakın düğümlerle birleştirilmez. Bunun yerine bir çember çizerek çemberin içindeki diğer düğümler ile ağaç yapısını bozmadan hedefe ulaşabilen bir yol varsa, bu düğümlerle birleştirilir [2]. RRT* algoritması da RRT gibi optimalliği garanti etmese de düğüm sayısı arttıkça eldeki harita keşfedilmiş olur ve RRT'den daha az düğüm kullanarak hedefe ulaşır. Örneklem bazlı arama

algoritmaları tekrar sayısı yükseldikçe tüm haritayı keşfeder ve haritadaki noktalar için yollar hesaplanmış olur. Böylelikle eğer otonom araç sabit bir ortamda hareket ediyorsa, çevrimdışı olarak harita keşfedilip, ardından çevrimiçi çalışırken iki nokta arasındaki yolun hızlıca bulunması sağlanabilir.

Diğer aşama olan rotanın izlenmesi için de farklı teknikler vardır. Örneğin yanal kontrol için DARPA'nın düzenlediği yarışmada etabı ilk defa bitirebilen Carnegie Mellon Üniversitesi'nin kullandığı pure pursuit algoritması buna örnek olarak gösterilebilir, [3]. Başka bir örnek olan [4]'de, bu ister nöral ağlarla güncellenen PID kontrolör ile sağlanmaya çalışılmıştır. Dayanıklı H_∞ kontrol yöntemi de problemi çözmek için [5] ve [6]'da kullanılmıştır. Son olarak lineer zamanla değişen model öngörülü kontrolörler de bu doğrultuda kullanılmıştır [7].

Çalışmada ilk olarak aracın takip etmesi gereken rota daha az noktayla ifade edilecek ve clothoidler yardımıyla oluşturulacaktır. Sonrasında model öngörülü kontrol için gereken araç modeli elde edilecektir. Son olarak ise aracın rotayı takip etmesi için gerekli lineer zamanla değişen model öngörülü kontrolör tasarlanacak ve kontrolöre ait simülasyon sonuçları verilecektir.

2. Referans Rotası Oluşturma

Bu bölümde aracın izlenmesi istenen rotanın daha yumuşak direksiyon hareketleriyle takibi için yeniden oluşturulması hedeflenmiştir.

2.1. Euler Spiralleri

Clothoid ya da euler spiralleri eğimi herhangi bir eğrinin uzunluğuyla doğrusal olarak ilişkili fonksiyonlar olup, doğal bir araç hareketi elde edilmesi ve yumuşak bir sürüş sağlanması gibi konularda avantajlıdır. Bu yüzden çalışma kapsamında rota clothoidlerle ifade edilecektir. Clothoidler, matematiksel olarak denklem 1'deki gibidir.

$$\kappa(s) = cl + \kappa_i \quad (1)$$

Burada c , eğriliğin değişimi yani keskinliği, κ_i başlangıç anındaki eğrilik, l ise eğrinin uzunluğunu ifade eder.

2.2. Referans Rotası Seyrekleştirme

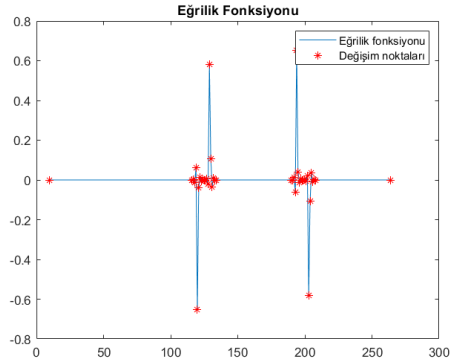
Rota oluşturulurken, örneklere bağlı olarak takip edilecek noktalar sık ya da seyrek olabilir. Fakat alınan sık örnekler, yolun yapısına ait farklı bilgiler sağlamaz. Dolayısıyla aynı bilgiyi taşıyan ardışıl noktalar, öngörü ufku boyunca yola ait bilginin çok farklı olmamasına neden olur. Başka bir deyişle, aynı yol karakterini taşıyan noktalar kullanmak, öngörü ufku boyunca daha uzun bir alanda öngörü sahibi olmak için öngörü ufku artırılabilir. Fakat bu da öngörü için kullanılan matrislerin boyutlarını büyüteceğinden, daha fazla işlem gücü ve çözüm için daha fazla zamana ihtiyaç duyar.

Bu yüzden, ardışıl ve sık örneklenmiş benzer yol karakterine sahip yol noktalarını almak yerine, aracın direksiyon açısının değişeceği noktaları referans olarak almak tercih edilebilir. Böylece aynı öngörü ufku ile yola ait çok daha uzun bir kısımda öngörü sahibi olunabilir. Aynı şekilde rota çok daha az

noktayla ifade edilmiş olur. Eğrilik bilgisinin anlam kaybetmeden seyrekleştirilmesi [8]'de bir optimizasyon problemi olarak önerilmiştir. Bu çalışmada ise eğrilik fonksiyonu dikkate alınarak denklem 2'deki gibi yapılacaktır.

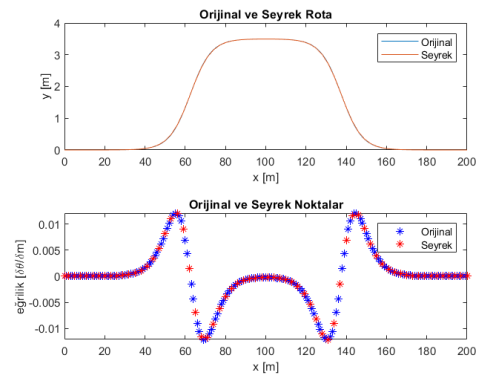
$$\kappa_{seyrek} = (i \in (2, N - 1) : \kappa_{i-1} - 2\kappa_i + \kappa_{i+1} \neq 0) \cup (1, N) \quad (2)$$

Eğrilik fonksiyonunun türevinin sıfırdan farklı olduğu noktalara değişim noktaları denir. Şekil 1'de görüleceği üzere belirtilen değişim noktaları arasında lineer fonksiyonlar vardır. Bu noktalar arasındaki doğrunun eğimi c ile, noktalar arasındaki uzaklık ise l ile gösterilir. 6'de görülebilir. Clothoidler, eğrilik



Şekil 1: Örnek bir yola ait eğrilik fonksiyonu.

fonksiyonlarında daha yumuşak bir geçiş sağladıkları için otoyol tasarımında da kullanılmıştır [9]. Benzer şekilde bu tezde yapılacak çalışmaya paralel olarak otonom araçların rota planlanmasında da kendine yer bulmuştur [10]. Çalışmada önerilen yöntem iki farklı rotada denenmiştir. Örneğin bir aracı sollarken yapılan bir manevra, Şekil 2'de gösterilmiştir. Görüldüğü üzere seçilen noktalardan oluşturulan rota ile orijinal rota üst üste gözükmemektedir. Böylelikle 201 referans noktadan oluşan orijinal rota, 63 noktaya indirilmiştir. Bu sayede %68'lik bir azalma ile aynı rota oluşturulmuştur.



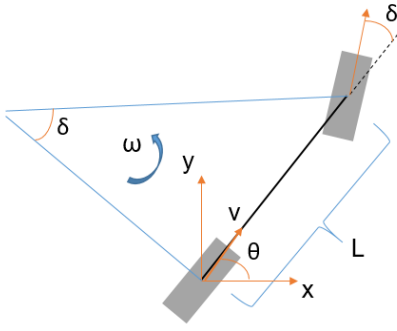
Şekil 2: Orijinal ve seyrek noktalardan oluşan sollama rotası.

3. Araç Modelleme

Model tabanlı kontrolörler çalışmak için sistem modeline ihtiyaç duyar. Bu sebeple, sistemi çalıştığı aralıkta temsil eden iyi modeli bulmak, kontrolörün iyi performansı açısından elzemdir. Bu yüzden bu bölümde araç modellenmesine odaklanılacaktır.

3.1. Kinematik Araç Modeli

2015’de Kong ve arkadaşları otonom sürüş kontrolör tasarımları için araçların dinamik ve kinematik modellerini karşılaştırmıştır [11]. Buna göre, 200 ms’de örneklenmiş kinematik modeller, 100 ms’de örneklenmiş dinamik modeller ile benzer performans göstermiştir. Bu çalışmada ise maksimum 15 km/h hız ile hareket eden araçlar ele alınacaktır. Dolayısıyla hem işlem yükü açısından daha basit bir model kullanmak, hem de dinamik modeli yeterince temsil ettiği için 200 ms’de örneklenmiş kinematik araç modeli uygun görülmüştür. Arka aks bazlı kinematik modelin görseli Şekil 3’de görülebilir. Araçlara ait zaman



Şekil 3: Araçın kinematik bisiklet modeli gösterimi.

domenindeki kinematik modelin matematiksel ifadesi ise, denklem 3’deki gibidir. Burada v aracın hızı, θ aracın rotadan sapma açısı, L aracın aks mesafesi ve δ aracın direksiyon açısı olarak tanımlanır.

$$\begin{aligned}\dot{x} &= v(t)\cos(\theta(t)) \\ \dot{y} &= v(t)\sin(\theta(t)) \\ \dot{\theta} &= \frac{v(t)}{L}\tan(\delta(t))\end{aligned}\quad (3)$$

Zaman domenindeki matematiksel modelin kullanılması aracın hızına bağlıdır. Bundan kurtulmak için uzaysal düzlemdeki gösterim tercih edilecektir. Bu iki domen arasında geçiş için $v(t)dt = ds$ ifadesi kullanılabilir. Böylece model, 4’deki hale gelir. Buradaki s , kat edilen mesafedir. Yani araç modeli böylece zamana bağlı değil, kat edilen mesafeye bağlı değişmektedir.

$$\begin{aligned}\dot{x} &= \cos(\theta(s)) \\ \dot{y} &= \sin(\theta(s)) \\ \dot{\theta} &= \kappa(s) = \frac{1}{L}\tan(\delta(s))\end{aligned}\quad (4)$$

Artık uzaysal düzlemdeki kinematik model ayrıık halde yazılabilir. Bunun için denklem 4’nin her iki tarafının da integralini alabiliriz. Böylelikle elde edilecek durum denklemlerinin değişimi denklem 5’deki hale gelir.

$$\begin{aligned}x_{k+1} &= x_k + \int_0^{l_k} \cos(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) ds \\ y_{k+1} &= y_k + \int_0^{l_k} \sin(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) ds \\ \theta_{k+1} &= \theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2} \\ \kappa_{k+1} &= \kappa_k + c_k l_k\end{aligned}\quad (5)$$

Denklem 5 sistemin ayrıık ve uzaysal düzlemdeki kinematik modelidir. Fakat bir sonraki durum değişkenlerini hesaplamak integral terimlerinden dolayı zor olacağı için integraller yaklaşıklık olarak da ifade edilebilir. Denklem 5’ün yaklaşık ifadesi denklem 6’daki gibidir.

$$\begin{aligned}x_{k+1} &= x_k + l_k \cos(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) \\ y_{k+1} &= y_k + l_k \sin(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) \\ \theta_{k+1} &= \theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2} \\ \kappa_{k+1} &= \kappa_k + c_k l_k\end{aligned}\quad (6)$$

Fakat model hala kosinüs ve sinüslü terimler içerir ve bu nedenle nonlineer bir modeldir. Lineer model kullanan bir kontrolör tasarımı hedeflendiği için model durum değişkenleri ve kontrol işaretleri etrafında doğrusallaştırılmalıdır. Lineerleştirirken giriş işaretleri ve durum değişkenlerine göre kısmi türev ifadeleri yazılmıştır. Sonuç olarak durum uzay modeli için oluşturulan matrislerin ifadesi denklem 7 ve denklem 8’de elde edilmiştir.

$$A_i = \begin{bmatrix} 1 & 0 & -l_k \sin(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) & -l_k^2 \sin(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) \\ 0 & 1 & l_k \cos(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) & l_k^2 \cos(\theta_k + \kappa_k l_k + c_k \frac{l_k^2}{2}) \\ 0 & 0 & 1 & l_k \\ 0 & 0 & 0 & 1 \end{bmatrix}\quad (7)$$

$$B_i = \begin{bmatrix} -\frac{l_k^3}{2} \sin(\theta_{k+1}) & \cos(\theta_{k+1}) - l_k \kappa_{k+1} \sin(\theta_{k+1}) \\ \frac{l_k^3}{2} \cos(\theta_{k+1}) & \sin(\theta_{k+1}) + l_k \kappa_{k+1} \cos(\theta_{k+1}) \\ \frac{l_k^2}{2} & \kappa_{k+1} \\ l_k & c \end{bmatrix}\quad (8)$$

4. Kontrolör Tasarımı

Çalışmada, gerçek zamanda kullanılabilmesi hem de lineer olmayan sistem modelini birçok farklı noktada lineerleştirip daha yakın bir temsil sağlaması nedeniyle lineer-zamanla değişen model öngörülü kontrolör kullanılacaktır.

4.1. Rota Takibi Problemi

Rotayı izlemek için aracın durum değişkenlerinin bilinmesi ve rotayla arasındaki hataya göre uygun kontrol işaretlerinin hesaplanması gerekir. Kontrol işaretleri $u = [c, l]$, durum değişkenleri $z = [x, y, \theta, \kappa]$ olarak kullanılacaktır. Sözel ifadeyle kontrol işareti aracın eğriliğinin ne kadar çok ve ne kadar uzunluk boyunca değiştirilmesi gerektiğinin hesaplanmasıdır. Clothoid tabanlı model öngörülü kontrolör için rota takip problemi 9'dan gibi tanımlanabilir.

$$J = \hat{z}_N^T Q_f \hat{z}_N + \sum_{i=1}^{N-1} (\hat{z}_i^T Q_i \hat{z}_i + \hat{u}_i^T R_i \hat{u}_i) \quad (9)$$

Şapkalı ifadeler durum değişkenleri ve kontrol işaretlerindeki hatalardır. İlk andan, öngörü ufku N 'e kadar $\hat{z}_i = z_i - z_i^{ref}$, $\hat{u}_i = u_i - u_i^{ref}$ ifadeleri hesaplanır ve Q ve R ağırlıklarıyla cezalandırılır. Tasarıma bağlı olarak bu ağırlıklar hem her öngörü ufku adımında hem de farklı değişkenler için değişik seçilebilir. Burada amaç, hem aracın uygulayamayacağı kontrol işaretlerinden, hem de yüksek hatalar içeren optimal çözümlerden kaçınmaktır. Stabiliteye olan etkisinden dolayı öngörü ufkunun sonundaki ağırlığı farklı seçmek iyi bir pratiktir.

4.2. Kontrolör Tasarımı

Rota takibi problemine uygun olarak mevcut hata ölçülerek, gelecekteki hatalar denklem 10 ile hesaplanır. Denklemde belirtilen A_i ve B_i lineerleştirilmiş sistem matrisleri, \hat{z} durum değişkenlerinin hata ifadesi, u kontrol işareti ve N ise öngörü ufkudur.

$$\hat{z}(k+1) = A_i \hat{z}(k) + B_i \hat{u}(k) \quad i = 1, \dots, N-1 \quad (10)$$

LTV-MPC baz alınarak tasarlanan maliyet fonksiyonu denklem 11'deki gibi yazılır ve hem sistem modeli hem de kısıtlar dahilinde çözüm aranır. Böylelikle öngörü ufku boyunca uygulanacak kontrol işareti serisi bulunur. Serinin ilk elemanı sisteme uygulanarak sonraki adımda problem tekrar çözülür.

$$\begin{aligned} \min_{\hat{u}(k)} \quad & J(\hat{z}(k), \hat{u}(k)) \\ \text{k.s.} \quad & \hat{z}(k+1) = A_i \hat{z}(k) + B_i \hat{u}(k) \quad i = 1, \dots, N-1 \\ & \text{kısıtlar} \end{aligned} \quad (11)$$

Çözüm iteratif bir yöntemle bulunabileceği gibi, denklemler iç içe yazılıp genişletilmiş matris ifadeleri ile de hesaplanabilir. Genişletilmiş matris ifadeleri ile de hesaplamak döngüler içermez. Böylece sadece mevcut hata ölçülüp, sistemin sonraki hataları denklem 12'deki gibi daha hızlı hesaplanabilir.

$$\bar{Z} = \bar{A} \hat{z}_0 + \bar{B} \bar{U} \quad (12)$$

Burada z_0 , mevcut anda ölçülen ya da kestirilen durum değişkenleri, \bar{A} ve \bar{B} genişletilmiş matris ifadeleridir. \bar{U} kontrol işaretinin, \bar{Z} hataların ufuk boyunca vektör şeklinde ifade edilmiş halleridir. Problemi standart forma çevirmek, birçok optimizasyon çözücü algoritma için gereklidir. Standart forma denklem 13'den ulaşılabilir.

$$J = \bar{Z}^T \bar{Q} \bar{Z} + \bar{U}^T \bar{R} \bar{U} \quad (13)$$

Denklem 13'deki \bar{Q} ve \bar{R} ifadeleri denklem 9'daki ağırlıkların diyagonal bir şekilde yeni bir matrise konulmasıyla oluşturulur. Burada matrislerin boyutları, $Q \in \mathcal{R}^{n \times n}$, $Q_f \in \mathcal{R}^{n \times n}$, $\bar{Q} \in \mathcal{R}^{n \cdot N \times n \cdot N}$ ve $\bar{R} \in \mathcal{R}^{l \cdot N \times l \cdot N}$ 'dir. Belirtilen n durum değişkenlerinin sayısı, l giriş işaretlerinin sayısı ve N öngörü ufkudur.

Karesel maliyet fonksiyonuna ulaşmak için gereken son adımda elde edilen matrisler birleştirilir. Karesel maliyet fonksiyonunun ifadesi denklem 14'den, bu denklemdeki H ve f ifadeleri denklem 15'den bulunabilir.

$$J = \frac{1}{2} \bar{U}^T H \bar{U} + f^T \bar{U} \quad (14)$$

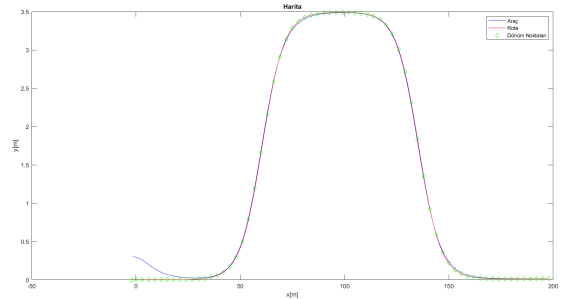
$$\begin{aligned} H &= 2(\bar{B}^T \bar{Q} \bar{B} + \bar{R}) \\ f &= 2\bar{B} \bar{Q} \bar{A} \hat{z}_0 \end{aligned} \quad (15)$$

Böylelikle her adımda oluşturulan yeni maliyet fonksiyonunda optimizasyon problemi kısıtlar dahilinde çözülür. Bu kısıtlar, kontrol işaretinin büyüklüğü, kontrol işaretinin değişimi ya da sistem çıktısı gibi büyüklükler için oluşturulabilir. Bu doğrultuda çalışmada tasarlanacak kontrolörde de, sisteme uygulanacak kontrol işaretinin büyüklüğüne kısıt getirilmesi amaçlanmıştır. Kısıtlar dahilinde tanımlanan optimizasyon probleminin ifadesi denklem 16'da görülebilir.

$$\begin{aligned} U^* &= \min J \\ \text{k.s.} \quad & \begin{bmatrix} I \\ -I \end{bmatrix} \bar{U} \leq \begin{bmatrix} u_{max} - u_i^{ref} \\ u_{max} + u_i^{ref} \end{bmatrix} \end{aligned} \quad (16)$$

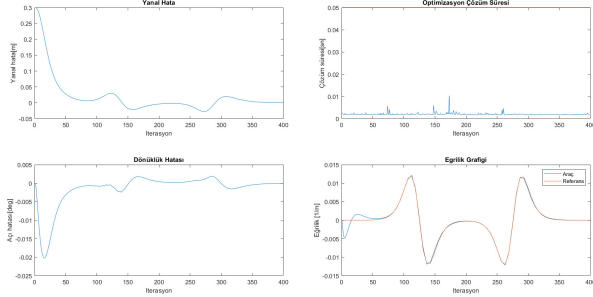
5. Simülasyon ve Sonuçlar

Önerilen yapının test edilmesi için iki farklı rota kullanılmıştır. Bunlardan ilki sollama manevrası, ikincisi ise gerçek bir pistten toplanan datalar ile oluşturulmuş bir rotadır. Her iki rotada da aracın hızı sabit 10 km/s ve aracın öngörü ufkusu N de 6 olarak tutulmuştur. Ayrıca maliyet fonksiyonunda kullanılmak üzere Q matrisinin diyagonal elemanları sırasıyla $[10, 100, 10000, 10000]$, R matrisinin diyagonal elemanları $[10, 100]$ olarak seçilmiştir. İlk rotada algoritmayı biraz daha zorlamak amacıyla başlangıç noktası rotadan 0.3 metre uzakta seçilmiştir.



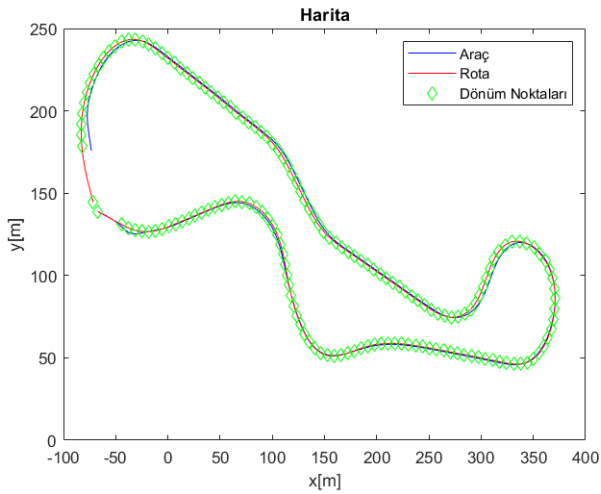
Şekil 4: LTV-MPC algoritmasının ilk rotadaki performansı.

Şekil 4'de görüldüğü üzere araç rotanın dışından takibe başlasa da ilgili rotayı başarılı bir şekilde takip edebilmiştir. Takip performansını nicelik olarak ölçmek için bazı metriklere kullanılmıştır. Bunlar, aracın yanal hatası, aracın olması gereken ile gerçek arasındaki fark olan dönüklük hatası, implemente edilen algoritmanın gerçek zamanlı çalışmasından emin olmak için kontrolörün çevrim süresi ve yolun referans eğriliği ile aracın eğriliği seçilmiştir. Bu metriklerden elde edilen sonuçlar ilk rota için şekil 5'de görülebilir. İkinci rotada gerçek bir test yolu



Şekil 5: LTV-MPC algoritmasının ilk rotadaki metrik değişimleri.

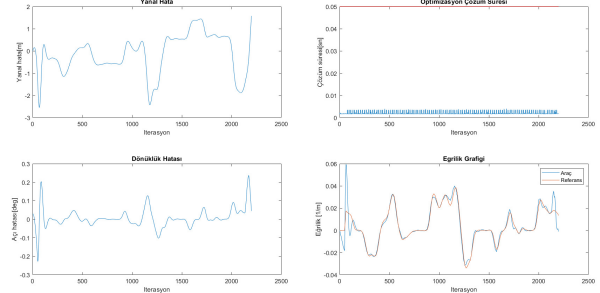
ele alınmıştır. Buradan toplanan veriler ile yola ait bir referans elde edilmiştir. Ardından toplanan referans, çalışmada belirtilen yöntem ile seyrekleştirilmiş ve dönüm noktaları tespit edilmiştir. Rota, bu dönüm noktaları kullanılarak clothoidler ile tekrar oluşturulmuştur. Yeniden oluşturulan rota, bu rotaya ait dönüm noktaları ve aracın rotada izlediği yol şekil 6'da görülebilir. İkinci rotaya ait metrik sonuçları ilk rotada olduğu gibi



Şekil 6: LTV-MPC algoritmasının ikinci rotadaki performansı.

hesaplanmıştır. Hesaplanan metrik değerler şekil 7'de görülebilir. Araç yol ile aynı noktadan başlamış, ardından istenen rotayı tamamlayabilmiştir. Her iki rotada da kontrolörün gerçek zamanlı çalışabilmesi için gereken süre koşulunu sağladığı görülebilir. Aracın izlediği yoldaki davranışını değiştirmek için Q ve

R matrisleri ya da öngörü ufkunu değiştirilebilir. Öngörü ufkunun az seçilmesi aracın kararsızlığa gitmesine sebep olur, fakat çok uzun seçilmesi performansı aynı orantıda artırmaz. Dolayısıyla yeterli bir öngörü ufkunu seçildikten sonra Q ve R matrislerinde durum değişkenlerine verilen ceza değiştirilerek aracın davranışına istenen şekil verilebilir.



Şekil 7: LTV-MPC algoritmasının ikinci rotadaki metrik değişimleri.

6. Sonuçlar

Bu çalışmada, literatürde otonom araçlarda rota planlama ve kontrol algoritmaları olarak kullanılan yapılar ve kullanılan bazı algoritmaların avantaj ve dezavantajlarına değinilmiştir. Çalışmada, planlanan rotanın clothoidler olarak adlandırılan ve otoyolların yapımında da kullanılan şekil ile yeniden oluşturulması amaçlanmıştır. Clothoidler karakterleri gereği sabit eğrilik içerdiklerinden daha konforlu bir sürüş gerçekleşmesine olanak sağlar. Ayrıca rota yeniden oluşturulurken her nokta değil, direksiyon açısının değiştiği dönüm noktaları tespit edilmiş ve bu şekilde rotada seyrekleştirme yapılmıştır. Seyrekleştirme sayesinde aynı karakteri içeren ardışık noktaları referans almak yerine, karakter değişimlerinin yaşandığı noktaları referans alınmıştır ve aynı kontrol ufkusuyla daha uzun mesafeler kapsanabilmiştir. Böylelikle işlem yükü artırılmadan daha uzun mesafede öngörü yapılabilmektedir.

Kontrol yapısı için ise otonom araçlarla yapılan çalışmalarda son yıllarda oldukça popülerlik kazanan model öngörülü kontrol yapısı üzerinde durulmuştur. Barındırdığı avantajlar dolayısıyla çalışmada kontrol yöntemi, lineer zamanla değişen model öngörülü kontrol olarak seçilmiştir. İlgili kontrolör yapısı ve referans rotanın clothoid tabanlı oluşturulması MATLAB ortamında simüle edilmiştir. Kontrolörde seçilen tüm durum değişkenlerine farklı ağırlıklar atanarak kontrolörün ilgili yoldaki performansları incelenmiştir.

Algoritmanın performansı şekil 5 ve şekil 7'de görülebilir. İlk rota görece olarak daha az eğrilik değişimi içerdiğinden 10 km/s hızla ortalama 8 santimetre hatayla tamamlanmıştır. Daha uzun ve daha fazla eğrilik içeren ikinci rotada ise araç bazı kısımlarda rotadan uzaklaşmıştır. Yine de şerit genişliği göz önüne alındığında aracın şeritte kalabileceği sınırlarda rotadan sapması sebebiyle, başarılı kabul edilebilir. Fakat daha iyi bir takip performansı için Q ve R matrisindeki ağırlıklar değiştirilebilir.

7. Kaynakça

- [1] T. A. J. Nicholson, "Finding the shortest route between two points in a network," *The computer journal*, vol. 9, no. 3, pp. 275–280, 1966.
- [2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [3] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," 1992.
- [4] G. Han, W. Fu, W. Wang, and Z. Wu, "The lateral tracking control for the intelligent vehicle based on adaptive pid neural network," *Sensors*, vol. 17, no. 6, 2017.
- [5] X. Huang, H. Zhang, G. Zhang, and J. Wang, "Robust weighted gain-scheduling h_∞ vehicle lateral motion control with considerations of steering system backlash-type hysteresis," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1740–1753, 2014.
- [6] C. Latrach, M. Kchaou, A. El Hajjaji, and A. Rabhi, "Robust h_∞ fuzzy networked control for vehicle lateral dynamics," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 905–910.
- [7] T. Juhlin-Henricson, "Implementation and analysis of a clothoid-based model predictive controller," 2016.
- [8] P. F. Lima, M. Nilsson, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Spatial model predictive control for smooth and accurate steering of an autonomous truck," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 4, pp. 238–250, 2017.
- [9] K. Baass, "Use of clothoid templates in highway design," Tech. Rep., 1982.
- [10] C. Alia, T. Gilles, T. Reine, and C. Ali, "Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method," in *2015 IEEE intelligent vehicles symposium (IV)*. IEEE, 2015, pp. 674–679.
- [11] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1094–1099, 2015.