Adaptif Hız Kontrol (AHK) Sistemindeki Mesafe Kontrol Sisteminin Sabit Mesafe Ve Sabit Zaman Yöntemleriyle Uygulamalı Olarak Karşılaştırılması

Hasan ŞAHİN

İstanbul Teknik Üniversitesi, Makina Fakültesi, Makine Mühendisliği, İstanbul, Türkiye

(Geliş/Received : 04.06.2016 ; Kabul/Accepted : 07.07.2016)

ÖZ

Bu çalışmada AHK sistemindeki mesafe kontrol sistemi farklı iki kontrolcü ile test edilmiştir. İlk kontrolcü mesafe kontrolünü PD algoritmasıyla araçlar arasındaki mesafeyi sabit tutmaya çalışarak sağlamıştır. İkinci kontrolcü ise Sabit Zaman Algoritması (SZA) ile mesafe kontrolünü araçlar arasındaki süreyi sabit tutmaya çalışarak sağlamıştır. MATLAB/Simulink programıyla yapılan simülasyonlar sonucunda çıkan sonuçlara göre PD algoritması ile yapılan mesafe kontrolü daha az hata ile daha konforlu geçişler sağlarken; SZA algoritması ile yapılan mesafe kontrolü daha çok hata ile daha keskin geçişler sağlamıştır. Sonuç olarak PD algoritması ile donatılan AHK sistemi daha güvenli ve konforlu bir sürüş sağlayabilir.

Anahtar Kelimeler: MATLAB/Simulink, Adaptif Kız Kontrolü, Otonom Kontrol Sistemleri, Araç Kontrol Sistemleri, Aktif Güvenlik Sistemleri.

Comparison of A Distance Controller Design Via Constant Distance And Constant Time Gap (CTG) Methods In An Adaptive Cruise Control (ACC) System

ABSTRACT

In this study, the distance control system of the ACC system was tested via two different controllers. The first controller is the PD controller which tries to keep the distance between vehicles according to the reference distance. Moreover, the second controller is the CTG controller which tries to keep the time gap between vehicles according to the reference time gap. The simulations were done via MATLAB/Simulink. Results show that the PD controller provided smoother transitions with less error however, the CTG controller provided sharper transitions with more error. Therefore, the ACC system which was developed with the PD controller could be much safer and comfortable than the CTG controller.

Keywords: MATLAB/Simulink, Adaptive Cruise Control, Autonomous Control Systems, Vehicle Control Systems, Advanced Driver Assistance Systems.

1. GİRİŞ (INTRODUCTION)

Araçlardaki aktif güvenlik sistemleri yıllardır daha da gelişerek kaza önleyici ve sürücü destekleyici sistemler olarak literatürde yerini almıştır [1]. Bu sistemler kazayı engellemek adına sürücüyü destekleyerek sürücüden kaynaklanan hataları en aza indirgemeye çalışmaktadır [1]. Bunların yanı sıra AHK sistemi hem konfor hem de bir güvenlik sistemidir [2]. Bu sistem aracı sürücünün belirlediği hızda sürerken aynı zamanda öndeki araçla mesafeyi de yine sürücünün belirlediği oranda ayarlayarak otomatik bir sürüş sağlamaktadır [3]. Sistem öndeki araçla olan mesafesini aracın ön kısmına yerleştirilmiş bir radar vasıtasıyla gerçek zamanlı olarak monitör ederek sürücünün istediği mesafeyi belirlenen oranda gaz ve fren pedalının otomatik kontrolü ile gerçekleştirmektedir [4]. Mesafe kontrolü için literatürdeki bir çok algoritma kullanılabilir. Bunlardan en önemlileri bu çalışmada kullanıldığı üzere SZA ve PD algoritmalarıdır [5]. AHK sistemindeki mesafe kontrolünün devreye girmesi için öndeki araçla mesafenin sürücü tarafından belirlenen değerden daha az olması gerekir [6]. Eğer mesafe sürücü tarafından belirlenen değerden fazlaysa AHK sistemindeki mesafe kontrolü yerine hız kontrolü devrededir [6]. Hız kontrolü sürücünün belirlediği hıza ulaşana kadar aracı otomatik olarak hızlandırır ve aracı o hızda sabit tutmaya çalışır [6]. Bu durumda aracın önüne başka bir araç çıkarsa ve bu araçla olan mesafe sürücü tarafından belirlenen mesafeden az olursa hız kontrolü devreden çıkar ve mesafe kontrolü devreye girer [6]. Bu çalışmada hız

**Sorumlu Yazar (Corresponding Author)*
e-posta: sahinhasan@itu.edu.tr
Digital Object Identifier (DOI) : 10.2339/2017.20.1 205-210

kontrolü olarak PI algoritması seçilmiştir. Hız kontrolü ile mesafe kontrolü arasındaki geçişi AHK sistemi kodlandığı şekilde otomatik olarak gerçekleştirmektedir.

Buçalışmanın simülasyonu MATLAB/Simulink üzerinde herhangi bir katılımcı olmaksızın yapılmıştır. Katılımcılarla yapılabilecek bir deneyde AHK sisteminin sürücüler üzerine etkisi de gözlemlenebilir [7]. Bu sistemin bir araca entegre edilmesinin olası en önemli avantajı ise sistemin reaksiyon zamanının insanın reaksiyon zamanından daha kısa olmasıdır [8]. Böylelikle sürüş güveninirliği de artmış olacaktır [9]. Örnek vermek gerekirse AHK sistemi içermeyen bir araçla AHK sistemini içeren bir araç kıyaslanabilir. AHK sistemini içermeyen araçla seyreden bir sürücünün önüne aniden bir araç çıktığında sürücü profesyonel bir sürücü dahi olsa reaksiyon zamanı yüzünden fren pedalını aktif hale getirmesi AHK sistemi içeren bir araca göre daha geç olacaktır. Bu sebeple de olası bir kazayı engelleme şansı daha az olacaktır [9]. Bunun yanı sıra AHK sistemi içeren bir çok araçla bir araç konvoyu yaratılabilir. Bu konvoydaki araçların birbiriyle bağlantıları belirlenen bir ses dalgası ile sağlanarak araçlar arasındaki mesafe istenilen oranda hatasız sağlanabilir [10].

Bu çalışmada tek bir araca AHK sistemi entegre edilerek bu aracın önündeki araçla olan mesafe kontrolü iki farklı algoritmayla kıyaslanarak yapılmıştır. Bu algoritmalar PD ve SZA algoritmalarıdır. Bu algoritmaların içeriği bir sonraki bölümde detaylı olarak açıklanacaktır. Yapılan simülasyonlardan elde edilen sonuçlara göre PD algoritması ile yapılan mesafe kontrolü daha az hata ile daha konforlu geçişler sağlarken; SZA algoritması ile yapılan mesafe kontrolü daha çok hata ile daha keskin geçişler sağlamıştır.

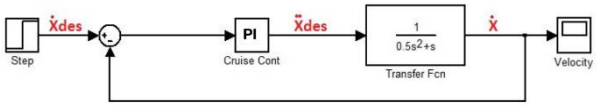
2. DENEY YÖNTEMİ (METHOD)

Daha önce bahsedildiği üzere AHK sistemi hız kontrolü ve mesafe kontrolü olmak üzere iki farklı kontrol sistemini içermektedir. Bu çalışmada mesafe kontrolüne odaklanılmış olsa da hız kontrolü olmadan AHK sistemi düşünülemez. Dolayısıyla deney yönetiminde hız kontrolü ve mesafe kontrolü detaylı olarak açıklanacaktır.

Taşıt modeli ise üst ve alt kontrol modeli olmak üzere iki modele ayrılmıştır.Modeller [3] nolu kaynaktan referans alınarak tasarlanmıştır. Üst kontrol modelinde taşıtın istenen hıza ulaşması için gerekli olan anlık ivme değeri hesaplanmıştır. Bu makalenin amacı farklı algoritmalarla bu anlık ivme değerini en sağlıklı şekilde belirlemeye çalışmaktır. Alt kontrol modelinde ise belirlenen bu anlık ivmeyi sağlamak için taşıt parametreleri göz önünde bulundurularak gerekli olan motor torku hesaplanmaktadır. Taşıtıntümözelliklerideğişse bile bu değişiklikler motor torku değiştirilerek tolore edilebilir. Bu makaledeki tüm simülasyonlar taşıtın sadece üst kontrol modeli kullanılarak yapılmıştır. Motor torkunu kontrol etmek ayrı bir çalışma konusudur. Dolayısıyla bu makalede vurgulanan üst kontrol modelindeki ivme çıktısını doğru bir şekilde kontrol edebilmektir. Taşıtın üst kontrol modeli PI,PD ve SZA algoritmalarının açıklandığı bölümlerde detaylı olarak yer almaktadır. Alt kontrol modelindeise 1 numaralı denklemde görüldüğü üzere T_{net} motorun net yanma torkunu, $J_e$motorun efektif atalet katsayısını, R çevrim oranını, $\ddot{x}_{des}$üst kontrol modelinde hesaplanan taşıt için anlık istenen ivme değerini, r_{eff}efektif lastik yarıçapını, c_a aerodinamik hava katsayısını ve ω_e motorun açısal hızını göstermektedir. Buradaki anlık ivme değeri belirtildiği üzere üst kontrol modelinden gelmektedir. Ona göre de gerekli olan motor torku ayarlanabilir.

$$T_{net} = \frac{J_e}{R\,r_{eff}}\ddot{x}_{des} + [c_a R^3 r_{eff}{}^3 \omega_e{}^2 + R(r_{eff} R_x)] \quad (1)$$

2.1 PI algoritmasıyla hız kontrolü



Şekil 1. PI algoritmasıyla hız kontrol sistemi (Cruise Control System via PI Controller)

PI algoritması oransal ve integral denetleyici PID kontrol döngüsü yönteminin kısaltması olarak verilmiştir. PI algoritmasıyla hız kontrolü yapılacak olan taşıtın üst kontrol modeli 2 numaralıdenklemde gösterilmiştir. Bu denklemde PI algoritması kullanılarak taşıta uygulanması gereken doğrusal ivme $\ddot{x}_{des}$ hesaplanmaktadır. k_p değeri oransal katsayıyı, k_i değeri ise integral katsayısını göstermektedir. $V_x$taşıtın gerçek hızı ve V_{ref} ise taşıtın olması gereken hızıdır. 3 numaralı denklemde 2 numaralı denklemin basitleştirilmiş hali görülmektedir.

$$\ddot{x}_{des}(t) = -k_p(V_x - V_{ref}) - k_i \int_0^t (V_x - V_{ref})dt \quad (2)$$

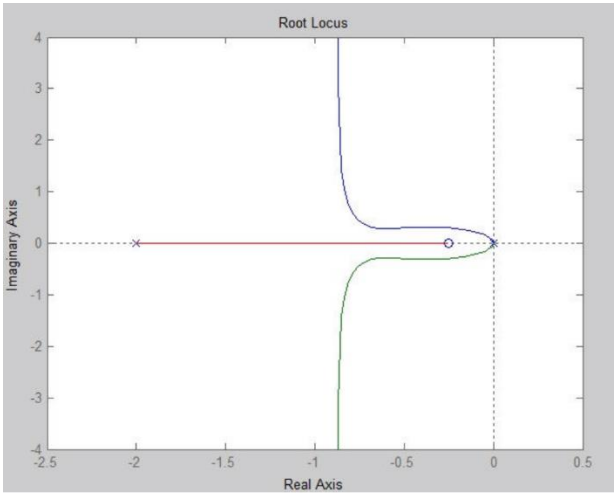
$$\ddot{x}_{des} = -k_p(\dot{x} - \dot{x}_{des}) - k_i(x - x_{des}) \quad (3)$$

Bu denklemlerle yapılan kapalı çevrim kontrol modeli Şekil 1'de gösterilmiştir.Kapalı çevrim modelinde 2.dereceden bir transfer fonksiyon bulunmaktadır. Bu fonksiyon 4numaralı denklemde gösterilmiştir. Bu denklemdeki τ değeri simülasyondaki gecikme süresini saniye cinsinden temsil etmektedir.

$$P(s) = \frac{1}{s(\tau s+1)} \quad (4)$$

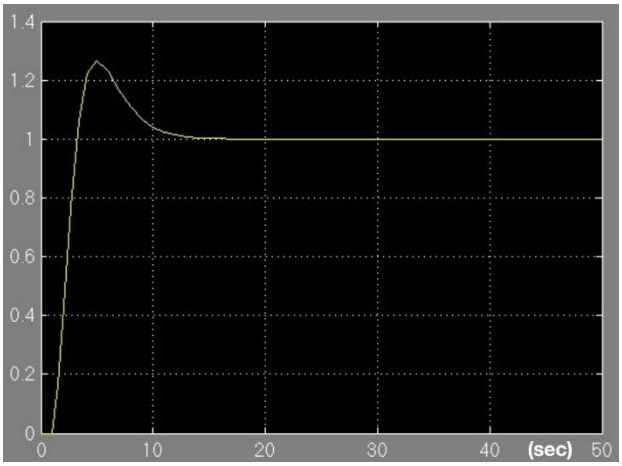
Dolayısıyla Şekil 1'de gösterildiği üzere bu transfer fonksiyon sistemde 0.5 saniyelik bir gecikme süresi olacak şekilde kontrolcü tarafından belirlenen ivmeyi hıza dönüştürecek şekilde tasarlanmıştır [3]. Olası gecikme süresi sistemin reaksiyon zamanı olarak da kabul edilebilir [3]. Transfer fonksiyonun çıktısı aracın gerçek hızıdır. Bu hızla hız kontrol sistemine sürücü tarafından girilen referans hız arasındaki fark PI algoritmasıyla kontrol edilerek transfer fonksiyona ivme çıktısı olarak iletilmektedir. Şekil 1'de referans hız olarak bir basamak fonksiyonu tanımlanmıştır. Hız kontrolü içerisindeki PI algoritmasına ait katsayılar "Köklerinin

Yer Eğrisi Çizimi" yöntemiyle hesaplanarak seçilmiştir [3,11]. Hız kontrol sisteminin kapalı çevrim modelinin "Köklerinin Yer Eğrisi Çizim" yöntemiyle çizilen grafiği Şekil 2'de gösterilmiştir.



Şekil 2. PI algoritması köklerinin yer eğrisi çizimi (Root Locus Plot of PI controller closed loop transfer function)

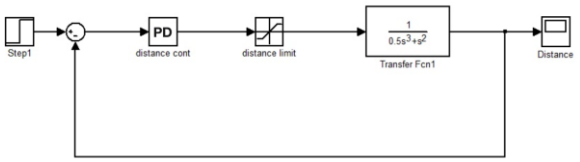
Şekil 2'deki grafiğe göre PI algoritmasındaki I değeri 0.1875 ve P değeri 0.75 olarak belirlenmiştir [3,11]. Tasarlanan hız kontrol sisteminin basamak fonksiyonu cevabı ise Şekil 3'de gösterilmiştir.



Şekil 3. PI algoritması basamak fonksiyonu cevabı(Step response of PI controller)

Şekil 3'de görüldüğü üzere sistemin basamak fonksiyon cevabında 25% oranında taşma meydana gelmiştir. Bu taşma iyileştirilebilir, ancak hız kontrol sisteminin tasarımı [3] nolu kaynaktan bir örnek olarak düzenleme yapılmadan alınmıştır. Bu çalışmanın amacı hız kontrol sisteminin tasarımını iyileştirmek değildir. Daha önce belirtildiği üzere çalışmanın asıl amacı mesafe kontrol sisteminin hassas tasarımı ve uygulamasıdır. Dolayısıyla hız kontrol sistemi değiştirilmeden belirtilen kaynaktan alınarak AHK sistemine eklenmiştir.

2.2 PD algoritmasıyla mesafe kontrolü



Şekil 4. PD algoritması ile mesafe kontrolü (Distance Control via PD Controller)

PD algoritması oransal ve türevsel denetleyici PID kontrol döngüsü yönteminin kısaltması olarak verilmiştir. PD algoritmasıyla mesafe kontrolü yapılacak olan taşıtın üst kontrol modeli 5 numaralı denklemde gösterilmiştir. Bu denklemde PD algoritması kullanılarak taşıta uygulanması gereken doğrusal ivme $\ddot{x}_{des}$ hesaplanmaktadır. k_p değeri oransal katsayıyı, k_d değeri ise türevsel katsayıyı göstermektedir. x taşıtın anlık aldığı mesafe ve x_{des} ise taşıtın anlık alması gereken mesafedir.

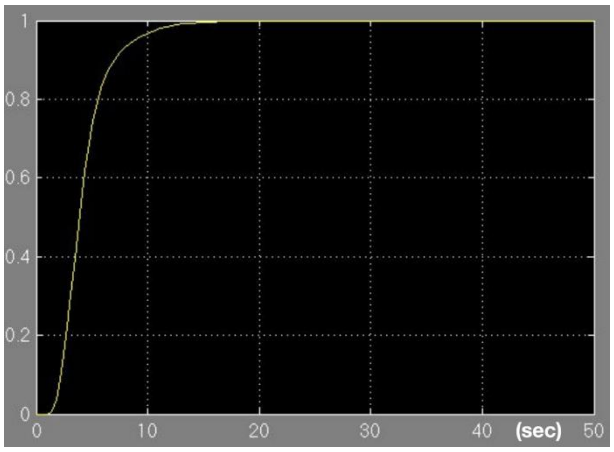
$$\ddot{x}_{des} = -k_p(x - x_{des}) - k_d(\dot{x} - \dot{x}_{des}) \quad (5)$$

Bu denklemle yapılan kapalı çevrim kontrol modeli Şekil 4'te gösterilmiştir. Kapalı çevrim modelinde 3.dereceden bir transfer fonksiyon bulunmaktadır. Bu fonksiyon 6 numaralı denklemde gösterilmiştir. Bu denklemdeki τ değeri simülasyondaki gecikme süresini saniye cinsinden temsil etmektedir.

$$P(s) = \frac{1}{s^2(\tau s+1)} \quad (6)$$

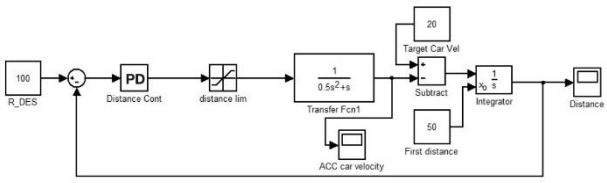
Dolayısıyla Şekil 4'te görüldüğü üzere bu transfer fonksiyon sistemde yine 0.5 saniyelik bir gecikme süresi olacak şekilde kontrolcü tarafından belirlenen ivmeyi mesafeye dönüştürecek şekilde tasarlanmıştır [3]. Transfer fonksiyonun çıktısı öndeki araçla olan gerçek mesafeyi göstermektedir. Bu mesafeyle mesafe kontrol sistemine sürücü tarafından girilen referans mesafe arasındaki fark PD algoritmasıyla kontrol edilerek transfer fonksiyona ivme çıktısı olarak iletilmektedir. Bununla birlikte PD algoritması ile transfer fonksiyonu arasında aracın üst ve alt ivmelenme değerleri belirlenmiştir. Buna göre araç hızlanırken üst ivmelenme limiti 1.0 m/s^2 ve araç yavaşlarken alt ivmelenme limiti -2.5 m/s^2 dir. Bu değerler araç hızlanırken ve yavaşlarken sürücü konforunu sarsmayacak şekilde belirlenmiştir [3]. Şekil 4'te referans mesafe olarak bir basamak fonksiyonu tanımlanmıştır [3]. Mesafe kontrolü içerisindeki PD algoritmasına ait katsayılar yine "Köklerinin Yer Eğrisi Çizimi" yöntemiyle hesaplanarak seçilmiştir [3,11]. Bu yöntemle yapılan özgün tasarımda PD algoritmasındaki D değeri 0.9495 ve P değeri 0.284 olarak belirlenmiştir.

Şekil 5'te görüldüğü üzere sistemin basamak fonksiyon cevabında sistem integral içermediği için herhangi bir



Şekil 5. PD algoritması basamak fonksiyonu cevabı (Step response of PD controller)

taşma görülmemektedir. PD algoritmasıyla yapılan mesafe kontrol sistemi ise Şekil 6'da görülmektedir.



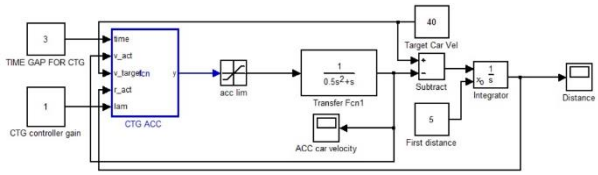
Şekil 6. PD algoritmasıyla mesafe kontrol sistemi (Distance Control System via PD controller)

Şekil 6'da Şekil 4'e ek olarak öndeki aracın hızı başlangıç anında ve simülasyon boyunca 20 m/s ve öndeki araçla olan başlangıç mesafesi ise 50 metre olarak tanımlanmıştır. Bununla birlikte sürücü tarafından öndeki araçla olan referans mesafesi de 100 metre

olarak tanımlanmıştır. Mesafe kontrolü yüklü olan ana aracın simülasyon başladığı ilk andaki hızı 0 m/s dir. Simülasyon sonucu ilerleyen bölümlerde gösterilerek tartışılacaktır.

2.3 SZA algoritmasıyla mesafe kontrolü

Şekil 7'de görüldüğü üzere Şekil 6 ile kıyaslandığında PD algoritmasının yerini SZA algoritması almıştır. Şekil 7'de mavi kutucukla gösterilen SZA algoritmasının MATLAB içerisine yüklenmiş kodu ise Çizelge 1'de görülmektedir. "%" işareti içeren satırlar formül açıklamalarıdır. Çizelge 1'deki ilk satır ise SZA algoritmasının temel formülünü göstermektedir [3]. Bu temel formül bir kaç formülün birleşmesinden oluşmaktadır.



Şekil 7. SZA algoritmasıyla mesafe kontrol sistemi (Distance Control System via CTG controller)

Çizelge 1. SZA algoritmasının MATLAB/Simulink içerisine yüklenmiş kodu (The code of CTG controller)

```
Y =  (-1/time) * [(V1- V2) + [L* (-D + (time*V1))]] ;
% Y = SZA algoritması tarafından hesaplanan ivme değeri (m/s2)
% time = Sürücünün belirlediği öndeki araçla olan süre farkı (s)
% V1 = Mesafe kontrol sistemi yüklü olan aracın gerçek hızı (m/s)
% V2 = Öndeki aracın gerçek hızı (m/s)
% D = Öndeki araçla olan gerçek mesafe (m)
% L = SZA algoritması hassasiyet katsayısı (1/s)
```

Çizelge 1'deki temel formül çıktısı aracın doğrusal yönde istenilen ivmesidir. SZA algoritmasında PD algoritmasına göre araçlar arası istenilen mesafe sabit değildir ve hıza bağlı olarak 7 numaralı denklemde gösterildiği üzere değiştirmektedir. h araçlar arası istenilen süre farkını, V_x ise mesafe kontrol yüklü olan aracın gerçek hızını göstermektedir.

$$\partial_{i\,des} = hV_x \quad (7)$$

Çizelge 1'deki temel formülde gösterilen (V1–V2) ifadesi 8 numaralı denklemde gösterilen ∂_iifadesinin türevinden gelmektedir. $\dot{\partial}_i$araçlar arası mesafe farkının türevi olarak araçlar arası hız farkını göstermektedir.Yine Çizelge 1'deki temel formülde gösterilen (-D + (time*V1)) ifadesi ise 8 numaralı denklemde belirtilen araçlar arası mesafe hatasını φ_i göstermektedir.

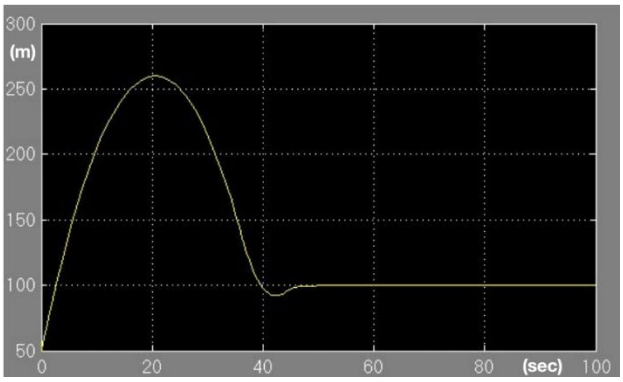
$$\varphi_i = -\partial_i + h\,V_x \quad (8)$$

7 ve 8 numaralı denklemler birleştirildiğinde Çizelge 1'deki temel formül elde edilmektedir [3]. Şekil 7'de ve Çizelge 1'de belirtildiği üzere SZA algoritması hassasiyet katsayısı 1; sürücü tarafından belirlenen öndeki araçla olan süre farkı 3 saniye; mesafe kontrol sistemi yüklü olan aracın ilk hızı 0 m/s; öndeki aracın başlangıç anından sonra simülasyon boyunca hızı 40 m/s ve öndeki araçla olan başlangıç mesafesi ise 5 metre olarak tanımlanmıştır. SZA algoritması da PD algoritması gibi bir kapalı çevrim sistemine sahiptir. PD algoritmasındaki

kapalı çevrimde sadece mesafe farkları kullanılırken; SZA algoritmasındaki kapalı çevrimde mesafe kontrol sistemi yüklü aracın gerçek hızı, öndeki araçla olan gerçek mesafesi ve öndeki aracın gerçek hızı kullanılmıştır. Bununla birlikte SZA algoritması ile transfer fonksiyonu arasında PD algoritmasında olduğu gibi aracın üst ve alt ivmelenme değerleri belirlenmiştir. Buna göre araç hızlanırken üst ivmelenme limiti yine 1.0 m/s^2 ve araç yavaşlarken alt ivmelenme limiti -2.5 m/s^2 dir. PD ve SZA algoritmalarının simülasyon sonuçları bir

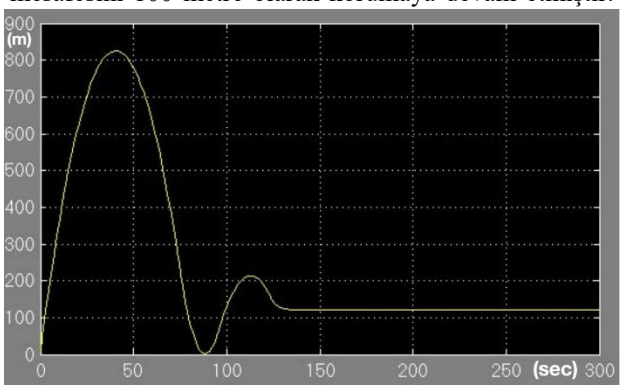
sonraki bölümde gösterilip detaylı biçimde kıyaslanacaktır.

3. SONUÇLAR VE TARTIŞMA (RESULTS AND DISCUSSION)



Şekil 8. PD algoritması ile donatılan mesafe kontrol sisteminin sağladığı mesafe değişimi (The distance between vehicles provided by the distance control system via PD controller)

İlk olarak PD algoritması sonuçları incelenecek olursa Şekil 8'de görüldüğü üzere simülasyon başlangıç anında araçlar arası mesafe 50 metredir. Sonrasında mesafe kontrol sistemi yüklü araç hızlanarak öndeki aracın hızını 20. saniyede (20 m/s) yakalamıştır. Sonrasında araçlar arasındaki mesafe referans değer olan 100 metreye gelinceye kadar mesafe kontrol sistemi yüklü araç hızlanmaya devam etmiştir. Araçlar arası mesafe 100 metreye geldiğinde mesafe kontrol sistemi yüklü araç yavaşlamaya başlamış ve sonrasında araçlar arası mesafe 92 metreye geldiğinde öndeki aracın hızını yakalamıştır. Sonrasında hızlanıp yavaşlayarak öndeki araçla mesafesini 100 metre olarak korumaya devam etmiştir.



Buradaki hata payı 8 metre olarak ölçülerek referans olarak belirlenen mesafenin 8% 'idir.

Şekil 9. SZA algoritması ile donatılan mesafe kontrol sisteminin sağladığı mesafe değişimi (The distance between vehicles provided by the distance control system via CTG controller)

SZA algoritması sonuçları incelenecek olursa Şekil 9'da görüldüğü üzere simülasyon başlangıç anında araçlar arası mesafe 5 metredir. Sonrasında mesafe kontrol sistemi yüklü araç hızlanarak öndeki aracın hızını 40. saniyede (40 m/s) yakalamıştır. Sonrasında araçlar arasındaki mesafe referans değer olan 3 saniyeye yani 120 metreye gelinceye kadar mesafe kontrol sistemi yüklü araç hızlanmaya devam etmiştir. Araçlar arası mesafe 120 metreye geldiğinde mesafe kontrol sistemi yüklü araç yavaşlamaya başlamış ve sonrasında araçlar arası mesafe yaklaşık 1 metreye geldiğinde öndeki aracın hızını yakalamıştır. Sonrasında hızlanıp yavaşlayarak öndeki araçla mesafesini 120 metre olarak korumaya devam etmiştir. Buradaki hata payı yaklaşık 119 metre olarak ölçülerek referans olarak belirlenen mesafenin 99% 'u yani neredeyse tamamı kadardır. Bu sonuç neredeyse bir kazaya yol açacak şekilde meydana geldiğinden asla kabul edilemez. SZA algoritmasının hassasiyet değeri tavsiye edilen değere yakın bir değer alınmıştır [3]. Hassasiyet değeri ne kadar değiştirilse de PD algoritmasının sağladığı 8% 'lik hata payına yaklaşamamıştır. Dolayısıyla ilk bakışta bu sonuçlara göre mesafe kontrol sisteminde PD algoritması tercih edilebilir.

4. SONUÇLAR (CONCLUSIONS)

Bu çalışmada AHK sistemindeki mesafe kontrol sistemi PD ve SZA algoritmalarıyla denenmiş ve hata oranları karşılaştırılmıştır. Çıkan sonuçlara göre otomatik pilot olarak kullanılan PD algoritmasıyla donatılan mesafe kontrol sisteminin davranışı insan davranışına çok benzer olduğundan bu algoritma tercih edilebilir [12]. Diğer bir yandan SZA algoritmasındaki hata payı çok yüksek olduğundan ve neredeyse bir kazaya sebep olacağından bu algoritmanın kontrol modeli değiştirilmeden otomatik pilot olarak kullanılamaz.

PD algoritmasında sürücü öndeki araçla olan referans mesafeyi metre cinsinden belirlerken; SZA algoritmasında saniye cinsinden belirlemektedir. Günümüzde AHK sistemlerinde sürücü öndeki araçla olan referans mesafeyi saniye cinsinden belirlemektedir. Araç hızlarının değişken olduğu düşünülürse bunun saniye olarak belirlenmesi çok daha mantıklıdır. Ancak bu durumda PD algoritmasının kullanılması pratik olarak mümkün değildir. Dolayısıyla gelecek çalışmalarda yapılması gereken SZA algoritmasının kontrol modelinin düzenlenerek PD algoritması gibi çalışmasını sağlamaktır. Sürücüye destek olan bu otonom sistemlerin tasarımında insanın reaksiyon zamanı da baz alınırsa sürücüyle beraber daha verimli bir sistem ortaya çıkacaktır [13,14].

KISALTMALAR (ABBREVIATIONS)

ACC	Adaptive Cruise Control
CTG	Constant Time Gap
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
SZA	Sabit Zaman Algoritması
AHK	Adaptif Hız Kontrol
T_{ne}	Motorun net yanma torku

J_e	Motorun efektif atalet katsayısı
r_{eff}	Efektif lastik yarıçapı

R	Çevrim oranı
$\ddot{x}_{des}$	Taşıtın olması gereken doğrusal ivmesi
x_{des}	Taşıtın anlık alması gereken mesafesi
x	Taşıtın anlık aldığı mesafe
$\partial_{i_{des}}$	Öndeki taşıtla istenen mesafe farkı
∂	Öndeki taşıtla gerçek mesafe farkı
$\dot{\partial}$	Öndeki taşıtla gerçek hız farkı
c_a	Aerodinamik hava katsayısı
ω_e	Motorun açısal hızı
k_p	PID oransal katsayısı
k_i	PID integral katsayısı
k_d	PID türevsel katsayısı
V_x , $\dot{x}$	Taşıtın gerçek hızı
V_{req} , $\dot{x}_{des}$	Taşıtın olması gereken hızı
h	Öndeki taşıtla istenilen süre farkı
φ	Öndeki taşıtla mesafe farkı hatası

KAYNAKLAR (REFERENCES)

[1] Thierry P., Kassaagi M. and Brissart G. "Active Safety Experiments with Common Drivers for the Specification of Active Safety Systems", 2001-06-0004***.Society of Automotive Engineers***, (2001).

[2] Tapani A. "Traffic simulation modelling of driver assistance systems". ***Advances in Transportation Studies an international Journal***, ***Section A*** 23: (2011).

[3] Rajamani R. "Vehicle Dynamics & Control". New York, USA: ***Springer,*** (2006).

[4] Bauer H. "Automotive Handbook: 7th Edition". Chichester, England: ***Wiley. Robert Bosch GmbH***, (2007).

[5] Ali Z., Popov A. and Charles G. "Model predictive control with constraints for a nonlinear adaptive cruise control vehicle model in transition manoeuvres". ***Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility***, 51:6, 943-963, (2013).

[6] Bauer H. "AHK Adaptive Cruise Control". ***Robert Bosch GmbH***, (2003).

[7] Xiong H. and Boyle L. "Drivers' Adaptation to Adaptive Cruise Control: Examination of Automatic and Manual Braking". ***IEEE Transactions on Intelligent Transportation Systems***, 13(3): (2012).

[8] Tapani A. "Vehicle Trajectory Effects of Adaptive Cruise Control", ***Journal of Intelligent Transportation Systems: Technology, Planning, and Operations***, 16:1, 36-44, (2012).

[9] Wang J., Zhang L., Zhang D. and Li K. "An Adaptive Longitudinal Driving Assistance System Based on Driver Characteristics". ***IEEE Transactions on Intelligent Transportation Systems***, 14(1), (2013).

[10] Milanes V., Shladover S., Spring J., Nowakowski C., Kawazoe H. and Nakamura M. "Cooperative Adaptive Cruise Control in Real Traffic Situations". ***IEEE Transactions on Intelligent Transportation Systems***, 15(1): (2014).

[11] Franklin G., Powel J. and Naeini A. "Feedback Control of Dynamic Systems", ***Pearson***, (2006).

[12] Moon S. and Yi K. "Human driving data-based design of a vehicle adaptive cruise control algorithm". ***Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility***, 46(8): 661-690, (2008).

[13] Lee D., McGehee V., Brown L. and Reyes L. "Collision warning timing, driver distraction, and driver response to imminent rear-end collisions in a high-fidelity driving simulator". **Human Factors**, 44: 314–335, (2002).

[14] Itoh M., Horikome T. and Inagaki T. "Effectiveness and driver acceptance of a semi- autonomous forward obstacle collision avoidance system". ***In Proceedings of the Human Factors and Ergonomics Society 54th annual meeting*** (pp. 2091-2095). Santa Monica, CA, (2010).