

Gerçek Zamanlı DC Motor Konum-Hız ve Konum-Hız-Tork Kaskad PID Denetleyici Performanslarının Karşılaştırılması

Hüseyin KOÇ¹, Burak Safa BERZENER¹, Tuğçe YAREN^{1*}, Selçuk KIZIR¹

¹ Kocaeli Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü, Kocaeli.

Sorumlu yazar e-posta: tugce.yaren@kocaeli.edu.tr
kou.huseyinkoc@gmail.com
burakberzener@gmail.com
selcuk.kizir@kocaeli.edu.tr

ORCID ID: <https://orcid.org/0000-0001-9937-3111>
ORCID ID: <https://orcid.org/0000-0003-4816-3231>
ORCID ID: <https://orcid.org/0000-0002-5929-5224>
ORCID ID: <https://orcid.org/0000-0002-0582-5904>

Geliş Tarihi: 27.01.2021

Kabul Tarihi: 01.08.2021

Öz

Endüstriyel servo uygulamaları genellikle çok döngülü kontrol yapısına sahiptir. Konum, hız ve akım döngüleri iç içe olacak şekilde bir kontrol yapısı oluşturulur. Oluşturulan yapıda, sistemin tüm durumları gözlemlenebildiği için sisteme etki eden dinamik yükler altında sistem yanıtında değişiklik olmaz. Klasik kontrol sisteminin bir alternatifi olan kaskad kontrol yöntemi, özellikle öngörülme bozucuların varlığında kontrol sisteminin performansını artırmaktadır. Bu çalışmada, DC motorun önce konum-hız kontrolü, daha sonra konum, hız ve tork kontrolü kaskad PID denetleyicisi tasarlanarak, gerçek zamanlı olarak uygulanmıştır. Uygulanan bu kontrol yöntemi Matlab-Simulink tabanlı Waijung blok set ile STM32F4 uygulama geliştirme kiti kullanılarak gerçekleştirilmiştir. Gerçek zamanlı uygulamada, farklı referans girişler için sistemden elde edilen sonuçlar grafiksel olarak sunulmuş ve kontrolcü performansı analiz edilmiştir. Sistem çıkışının tüm referans girişleri başarılı bir şekilde takip etmesi ile tasarlanan kontrolcünün performansı net bir şekilde ortaya çıkarılmıştır.

Anahtar kelimeler

DC Motor; Gerçek Zamanlı Kontrol; Kaskad Kontrol; STM32F4 Kiti; Waijung Blok Set.

Comparison of Real Time DC Motor Position-Speed and Position-Speed-Torque Cascade PID Controller Performances

Abstract

Industrial servo applications generally have a multi-loop control structure that has the position, velocity, and current loops. Since all states of the system can be observed in the cascade structure, there is no change in the system response under dynamic loads affecting the system. The cascade control method, which is an alternative to the classical control system, increases the performance of the control system especially in the presence of unpredictable disturbances. In this study, first the position-speed control of the DC motor, then the position, speed, and torque control of the DC motor are designed by using cascade PID controller and implemented in real-time. The designed controllers were implemented by using Matlab-Simulink based Waijung block set and STM32F4 development kit. In real time application, the results obtained from the system for different reference inputs are presented graphically and the controller performance has been analyzed. The performance of the designed controller is clearly shown by the system output following all reference inputs successfully.

Keywords

DC Motor; Real Time Control; Cascade Control; STM32F4 Kit; Waijung Block Set.

1. Giriş

Bir kontrol sisteminde asıl amaç, dış etkenlerden gelen bozucu sinyalin yok edilmesi ve referans sinyalin hatasız takip edilmesidir (Bingül ve Küçük 2017). Motor miline bağlanan ve değeri bilinmeyen bir yük veya yükteki ani değişimler, bağlantı elemanlarındaki sürtünme, boşluklar DC motorda bir takım belirsizliklere neden olabilir. Karşılaşılan

bu belirsizlikler çıkışın (konum, hız veya tork), referans yörüngesi hatalı takip etmesine neden olabilir (Karadeniz vd. 2014). Güçlü bir kontrolör yukarıda belirtilen hataları ve yukarıda belirtilen hatalardan kaynaklanan problemleri gidererek iyi bir davranış gösterebilir (Bingül ve Küçük 2017).

PID kontrol yöntemi, daha gelişmiş kontrol teknikleri olmasına rağmen, proses endüstrilerinde

hala yaygın olarak kullanılmaktadır (Zhuang 1994). Basit yapısı, yüksek verimlilik ve kolay uygulama gibi avantajları tercih sebeplerindedir (Kelekçi vd. 2019). PID kontrolörlerinin en iyi özelliklerinden biri, birden çok PID denetleyicisinin birlikte kullanılması ile kaskad PID konfigürasyonunun uygulanabilmesidir (Gucin et al. 2015). Bu konfigürasyonda, kontrolörler, bir PID' nin girişinin başka bir PID bloğu tarafından kontrol edildiği şekilde yerleştirilir.

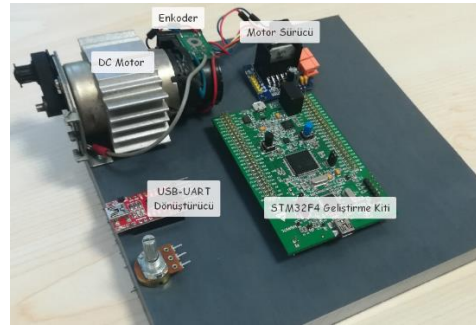
Kaskad PID yapısı, tek PID denetleyici yapısı ile karşılaştırıldığında daha iyi dinamik performans sağlamaktadır. Hem düşük hem yüksek kapasiteli motorlar için referans takibi, bozucu etki yok etme, başlama akımının azaltılması gibi özellikleri yüksek verimle sunar (Patel and Dave 2011). Tek döngüye kıyasla kaskad sistemlerin avantajları esneklik, hızlılıktır. Ayrıca, bu kontrol konfigürasyonu sistemin zaman sabitini azaltır (Kumar 2013).

Sabit mıknatıslı DC motorun hızı, torku, akımı ve konumu genellikle kaskad kontrolörler tarafından kontrol edilir. Motor sürücüler, konum, hız veya akım için bir komut sinyali alır ve kapalı döngü geri beslemesine dayanarak motora uygulanan voltaj ve akımı ayarlar. Bu tür kontrol sistemleri kaskad kontrol yapısını kullanır. Bu kontrol döngüsünün amacı, döngü dinamiklerini ortadan kaldırmaktır. Kaskad kontrol döngüsü tasarımı pratik uygulamalar için uygundur.

Bhavina ve arkadaşları, iki döngülü kaskad PID denetleyici yapısı tasarlayıp, DC motor sistemine PIC18F ile uygulamışlardır. Akım ve hız döngülerini içeren kaskad denetleyicinin, tek döngülü PID denetleyiciye göre üstün performans sağladığı sonucuna varmışlardır (Bhavina et al. 2013). Baçac ve arkadaşları, farklı kontrol teknikleri ile DC motor konum kontrol algoritmaları geliştirip, DC motor sistemine gerçek zamanlı olarak uygulamışlardır. Geliştirdikleri tekniklerden biri 2 seri PI denetleyiciye sahip kaskad kontroldür. Kaskad yapı, konum ve hız döngülerine sahiptir (Bacac et al. 2014). Robet ve Gautier, üç döngülü kaskad bir denetleyiciyi, ivme döngüsü ekleyerek dört döngülü bir kaskad denetleyici tasarlamışlar ve yöntemi deneysel bir DC motor sisteminde uygulamışlardır. İvme döngüsünün, tork bozunumlarını azaltmak için

kullanışlı olduğu sonucuna varmışlardır (Robet and Gautier 2014).

Bu çalışmada, sabit mıknatıslı bir DC motorun pozisyon, hız ve tork kontrolü için kaskad PID denetleyicisi tasarlanmış ve Matlab-Simulink destekli Waijung blok seti ile STM32F4 uygulama geliştirme kiti kullanılarak gerçek zamanlı olarak kontrolü sağlanmıştır. Düşük maliyetli oluşu, 32 bitlik ARM işlemciye sahip olması, programlayıcı ve hata ayıklama birimlerinin kit üzerinde yer alması, kolay öğrenilebilir olması ve Simulink üzerinden koddan bağımsız bloklar ile hızlı bir şekilde programlanabilir olması STM32F4 uygulama geliştirme kitinin seçilmesinin nedenlerindedir (Kizir vd. 2019).



Şekil 1. Deney düzeneği.

Kolay bulunabilir olması, düşük maliyetli olması, konum, hız, tork ve bunların birlikte ele alınabileceği (servo kontrol gibi) farklı kontrol tekniklerinin uygulanabilmesi, birçok sistem içinde yer alan bir bileşen olması ve küçük değişikliklerle farklı kontrol problemlerine dönüştürülebilmesi gibi nedenlerle DC motor kullanımı tercih edilmiştir (Kizir vd. 2019).

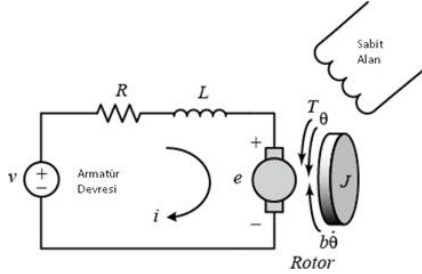
Şekil 1'de, kontrolcünün test edildiği uygulama geliştirme seti görülmektedir. Deney setinde 24V beslemeli, kayış-kasnaklı ve enkoderli, sabit mıknatıslı Canon FN38-S fırçalı DC motor kullanılmıştır. Motorda çözünürlüğü 168 CPR (Counts Per Revolution) olan iki kanallı artımlı enkoder bulunmaktadır. Aynı zamanda deney setinde STM32F4 uygulama geliştirme kiti, H köprü yapısına sahip LMD18200 motor sürücüsü ve seri haberleşme için USB-UART dönüştürücü bulunmaktadır.

2. DC Motor Modeli

Homojen bir manyetik alan varsayımı altında, doğru akım (DC) motoru, motor akımından elektriksel

torkuna doğrusal bir dönüştürücü olarak modellenmiştir. DC motorun bu klasik modeli, birleşik bir elektrik ve mekanik bir alt sistemden oluşur. Böylece DC motorun matematiksel modelinin çıkarılmasında elektriksel ve mekanik yapısından yararlanır.

Armatürün elektrik devresi ve rotorun serbest cisim diyagramı Şekil 2’de görülmektedir. DC motorun elektriksel ve mekanik eşitlikleri sırasıyla Denklem (1) – (4)’de görülmektedir.



Şekil 2. DC Motor Elektriksel ve Mekanik Devresi (Control Tutorials for MATLAB and Simulink).

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + e(t) \quad (1)$$

$$T(t) = J \frac{d^2\theta(t)}{dt^2} + b \frac{d\theta(t)}{dt} \quad (2)$$

$$T(t) = K_t i(t) \quad (3)$$

$$e(t) = K_e \omega(t) \quad (4)$$

Burada ω (rad/sn) ve i (A) sırasıyla açısal hız ve armatür akımını ifade eder. K_t (Nm/A), K_e (V/rad/sn) makine tasarımına bağlı olan tork ve zıt-emk sabitlerini ve b (Nms) viskoz sönümlenmeyi ifade eder. (Genellikle $K_t = K_e = K$ eşit kabul edilir.)

T (Nm) ve e (V) ise tork ve zıt-emk' yı temsil etmektedir. J (kgm^2) motor armatür ve şaftını içeren mekanik atalettir. L (Henry), R (ohm) ve v (V) sırasıyla endüktansı, motorun toplam bağlantı direncini ve giriş voltajını temsil eder. Görüldüğü üzere ele alınan denklemler zaman (t) uzayında yazılmıştır. Bu denklemlerin Laplace dönüşümleri alınarak s uzayındaki formları yazılır ve gerilim, zıt emk ve moment tork denklemlerinden sistem transfer fonksiyonları elde edilir.

Bilindiği üzere transfer fonksiyonu, sistemin girişi ve çıkışı arasındaki ilişkiyi temsil etmektedir. Motorun giriş sinyali motora uygulanan gerilimdir. Çıkış ise

gerçekleştirilen uygulamaya göre değişiklik gösterebilir. Denklem (1) – (4) kullanılarak gerilim/akım, gerilim/hız, gerilim/konum modelleri elde edilebilir.

Motorun akımı ile gerilim arasındaki ilişkiyi gösteren transfer fonksiyonu:

$$\frac{I(s)}{V(s)} = \frac{1}{L_a s + R_a} \quad (5)$$

Motorun hızı ile gerilim arasındaki ilişkiyi gösteren transfer fonksiyonu:

$$\frac{\omega(s)}{V(s)} = \frac{K}{L_a s^2 + (JR_a + bL_a)s + (R_a b + K^2)} \quad (6)$$

Motorun konumu ile gerilim arasındaki ilişkiyi gösteren transfer fonksiyonu:

$$\frac{\theta(s)}{V(s)} = \frac{K}{L_a s^3 + (JR_a + bL_a)s^2 + (R_a b + K^2)s} \quad (7)$$

Çizelge 1’de deney setinde kullanılan DC motorun parametreleri yer almaktadır.

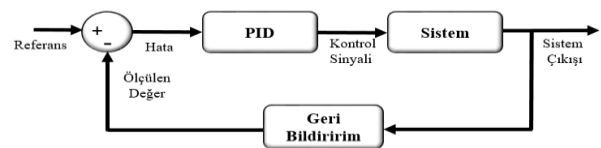
Çizelge 1. DC motor parametreleri.

Parametre	Birim	Değer
Rotor Direnci (R)	ohm	5
Rotor Endüktansı (L)	H	0.0029
Rotor Atalet Momenti (J)	kg.cm ²	0.032
Motor Tork Sabiti (K _t)	Nm/A	0.0338
Zıt Emk Sabiti (K _e)	V/rad/sn	0.0338

3. Kontrol

3.1 PID kontrol

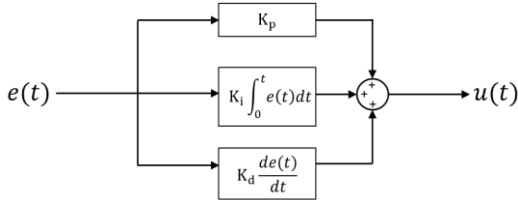
PID kontrol, geri beslemeli kontrol sistemleri arasında en yaygın olarak kullanılan denetim sistemidir. Şekil 3’de blok diyagramı görülen PID kontrol, geri besleme verisinin referans değerinden çıkartılması ile elde edilen hatayı oransal, integral ve türev etkisinde paralel olarak işleyerek, sistemin istenilen referans değerine oturmasını sağlayan bir kontrol metodudur.



Şekil 3. PID kontrol yapısı (Kizir vd. 2019).

Denetim yapısı içerisindeki P, I ve D işlemleri (Şekil 4) farklı formlarda birleşerek P, PI veya PD gibi kontrol yöntemlerine de dönüşebilmektedir.

PID denetleyicilerinin amacı referans ile ölçülen değer arasındaki hatayı zamanla azaltarak sistemin referans değerine ulaşmasını sağlamaktır. PID denetleyiciler, yüzde aşım, oturma zamanı ve yükselme zamanı gibi sistem davranışları açısından önemli parametreleri, sistemin ve PID bloğunun transfer fonksiyonlarını kullanarak istenilen seviyelerde tutmak için de tasarlanabilir.



Şekil 4. PID bileşenleri (Kizir vd. 2019).

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (8)$$

$$e(t) = r(t) - y(t) \quad (9)$$

$u(t)$, PID kontrolör çıkış sinyalini, $y(t)$ çıkış sinyalini, $r(t)$ referans sinyalini ve $e(t)$ ise hata sinyalini göstermektedir. Denklemden yer alan K_p , K_i ve K_d katsayıları sırasıyla oransal, integral ve türev sabitleridir. PID denetleyici içerisindeki K_p , K_i ve K_d katsayılarının her biri sistem parametrelerine karşı farklı etkiye sahiptir (Çizelge 2).

Çizelge 2. PID parametrelerinin sisteme etkisi.

Artan Parametre	Yükselme Zamanı	Yüzdellik Aşım	Oturma Zamanı	Kararlı Durum Hatası
K_p	↓	↑	-	↓
K_i	↓	↑	↑	↓
K_d	-	↓	↓	-

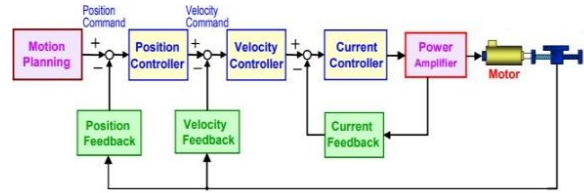
3.2 Kaskad PID kontrol

Bu çalışmada kaskad PID denetleyici tasarımı yapılmıştır. Servo sürücüler genellikle kaskad kontrolör (birden fazla kontrol döngüsü) yapısına sahiptir, akım döngüsü konum döngüsünün içine yerleştirilmiş olan hız döngüsünün içine yerleştirilmiştir. Bu denetleyici sayesinde sistemin tüm durumları gözlemlenebilmektedir.

Şekil 5’de kaskad PID denetleyici yapısı görülmektedir. Kontrolör, komut verilen değeri (konum, hız veya tork) elde edilen değerle karşılaştırmak için geri besleme cihazından çıkış

kullanır ve hataları düzeltmek için sürücüye komutlar verir.

Servo sürücü mantığı en fazla üç tip servo döngüsü içerebilir - akım, hız ve konum. Bu servo devreleri, istenen sonucu elde etmek için döngünün çıkışını ayarlamak üzere geri besleme sinyalleri kullanır.



Şekil 5. Kaskad PID denetleyici yapısı (Gross 1983).

3.2.1 Hız Döngüsü

Hız döngüsü en yaygın servo kontrol döngüsüdür. Bir enkoder aracılığıyla komut verilen hızı gerçek hız ile karşılaştırır ve motorun hızını buna göre artırmak veya azaltmak için komutlar verir. Hız döngüsüne, düzeltme komutunu belirlemek için tipik olarak hem orantı kazancını (K_p) hem de integral kazancını (K_i) kullandığı için PI denetleyicisi de denir.

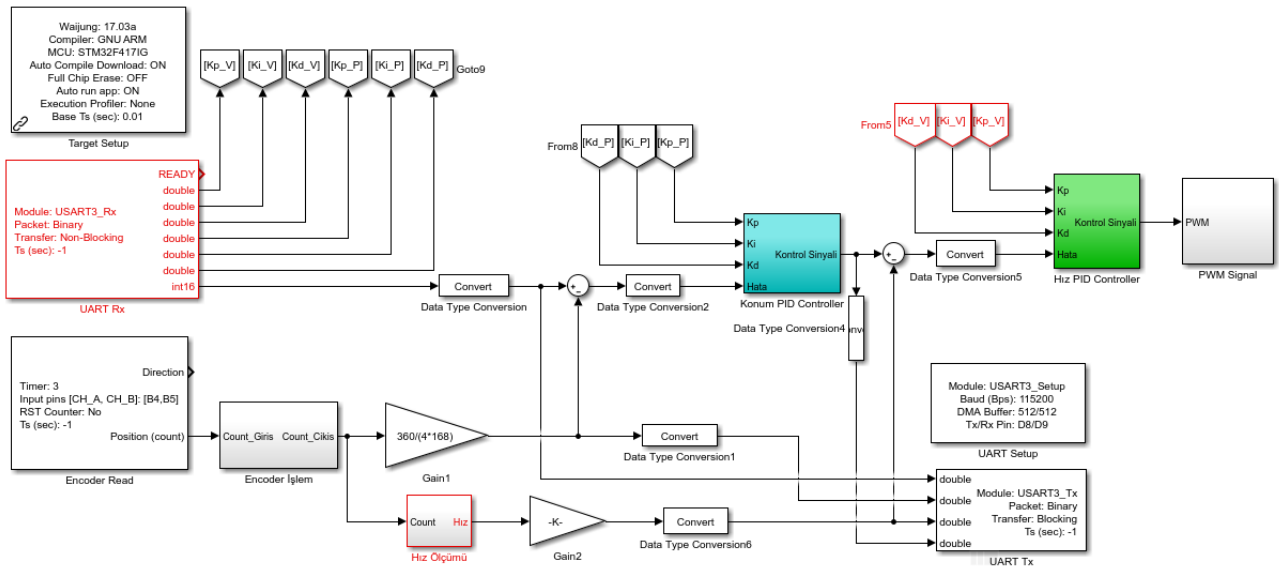
3.2.2 Konum Döngüsü

Konum kontrolünü gerektiren uygulamalar için, kaskad konum/hız döngüsü olarak bilinen, hız döngüsünün çevresine bir konum döngüsü eklenir. Pozisyon döngüsü, gerçek ve referans verilen konumlar arasındaki hatayı belirler ve hatayı azaltmak veya ortadan kaldırmak için hız komutları verir. Basamaklı bir sistemde, konum döngüsü tipik olarak yalnızca orantılı bir kazanç kullanır.

3.2.3 Akım Döngüsü

Birçok endüstriyel servo uygulamasında olduğu gibi, gerekli tepki süresi yüksek olduğunda akım kontrolü gereklidir. Akım döngüsünün birincil amacı, hızı ve dolayısıyla konumu etkileyen torku kontrol etmektir. Akım döngüsü tipik olarak hız döngüsünün içine yerleştirilir, bu da akımı en içteki döngü yapar, hız döngüsü ortada ve konum döngüsü en dıştaki döngüdür. Akım döngüleri tipik olarak hem orantılı hem de integral kazançları olan PI kontrolörleridir.

Kaskad kontrol uygulamalarında iç döngünün tepki süresi dış döngüden daha hızlı olmalıdır. Aksi takdirde iç döngünün dış döngüye bir etkisi olmayacaktır.



Şekil 6. Konum-Hız kontrolü Simulink modeli.

4. Gerçek Zamanlı Uygulama

4.1 Konum – hız kontrolü

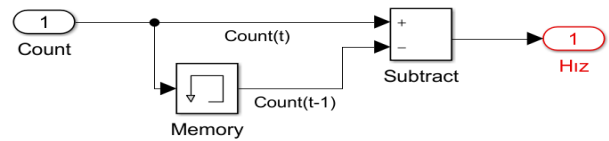
Bu çalışmada öncelikli olarak DC motor konum ve hız kontrolü gerçekleştirilmiştir. Kontrolü gerçekleştirmek için Waijung blok set kullanımı ile Şekil 6’ da görülen Simulink uygulama modeli oluşturulmuştur.

Oluşturulan kontrol yapısı kaskad PID denetleyicisi tasarımına dayanmaktadır. PID kontrol katsayıları deneysel gözlemlere dayalı olarak öncelikle iç döngüden (hız döngüsü) başlanıp daha sonra dış döngüden (konum döngüsü) devam edilerek belirlenmiştir. Deneysel gözlemler sonucunda bulunan PID katsayıları Çizelge 3’de verilmiştir.

Çizelge 1. Deneysel gözlemler sonucu bulunan kontrol katsayıları.

Parametre	Konum Kontrol	Hız Kontrol
K_p	2.12	0.015
K_i	0.015	0.625
K_d	0.0065	0.000001

Enkoderden okunan sinyaller ile gerekli işlemler yapılarak DC motorun gerçek zamanlı konum ve hız verisine ulaşılabilmektedir. Konum bilgisi enkoder bloğundan okunan sinyal sayacının $360/(4xCPR)$ ile çarpılmasıyla elde edilmiştir. Hız bilgisi ise enkoder bloğundan okunan sinyal sayacının $T=0.01s$ ’de saydığı değer $60/(4xCPRxT)$ ile çarpılmasıyla elde edilmiştir. Şekil 7’de hız hesabı için oluşturulmuş *Hız Ölçümü* blok yapısı görülmektedir.



Şekil 7. Hız Ölçümü blok yapısı.

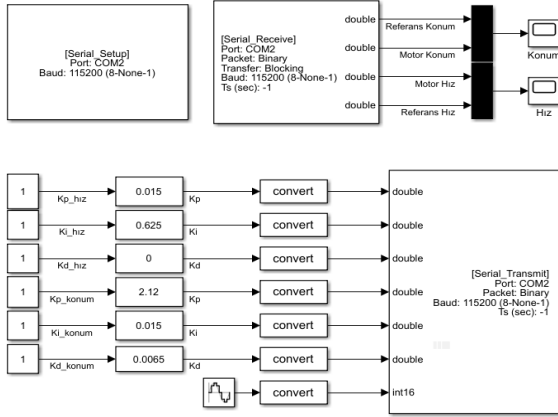
Konum bilgisi konum kontrolcüsüne, hız bilgisi de hız kontrolcüsüne geri besleme olarak verilmiştir. Kaskad PID denetleyici yapısında konum kontrolcüsünün çıkışı hız kontrolcüsünün referansı olur, hız kontrolcüsünün çıkışı da sistemin çıkışıdır.

Simulink modelinde görülen PID kontrolcü blok (*Konum – Hız PID Controller*) yapıları Şekil 4 esas alınarak oluşturulmuştur. Dolayısıyla konum kontrolcü blok girişini, hata ve PID parametreleri (konum) oluşturmakta ve blok içerisinde hata sinyali K_p, K_i, K_d parametreleri ile gerekli işleme sokulup, toplanarak kontrol sinyali elde edilmektedir (Şekil 4). Benzer şekilde elde edilen konum kontrol sinyali ve PID parametreleri (hız) hız kontrolcü blok girişine uygulanıp, hız kontrol sinyali elde edilmektedir.

Denetleyici referans sinyallerini ve parametrelerini (K_p, K_i, K_d) bilgisayardan göndermek ve ölçüm verilerini sistemden alabilmek için, Şekil 8’ deki ana bilgisayar modeli (host model), 0,01 örnekleme zamanı ile UART iletişim protokolü kullanılarak tasarlanmıştır.

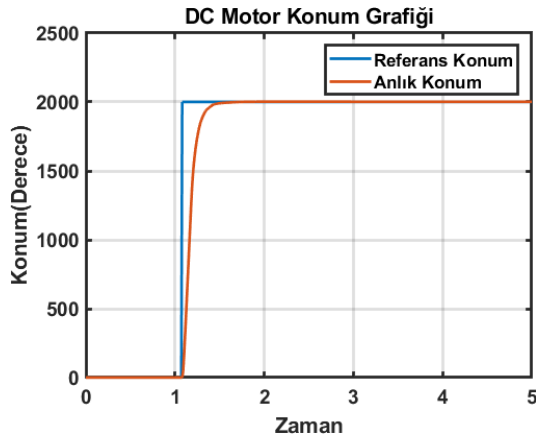
Kaskad denetleyici performansının gözlenebilmesi için sisteme farklı referans girişler verilmiştir. Şekil 8’de görülen ana bilgisayar modeli aracılığıyla

gerçek zamanlı olarak deney düzeneğine referans giriş sinyali uygulanmış ve sistem yanıtları (DC Motor konum ve hız bilgisi) aynı model üzerinden incelenerek veri toplama işlemi gerçekleştirilmiştir.

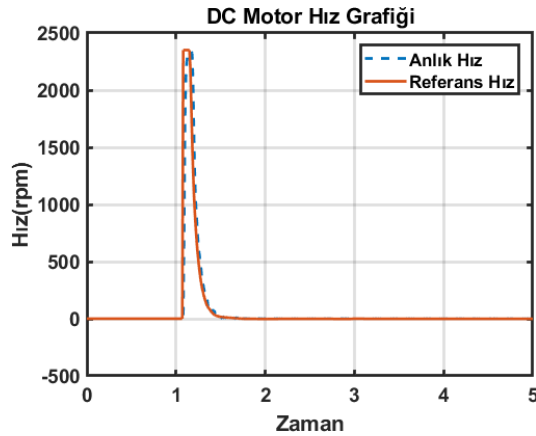


Şekil 8. PID kontrol – ana bilgisayar modeli.

Öncelikle sisteme referans giriş sinyali olarak basamak fonksiyonu uygulanmıştır. Bu durumda sistem çıkış sinyallerinin referans sinyalleri takip etmesi beklenmektedir.



Şekil 9. Basamak giriş / sistem konum yanıtı.

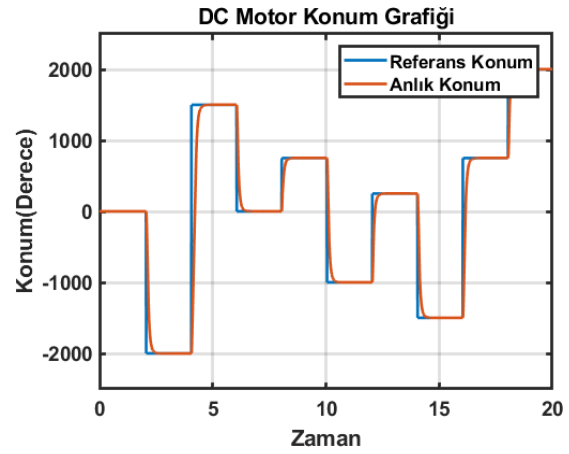


Şekil 10. Basamak giriş / sistem hız yanıtı.

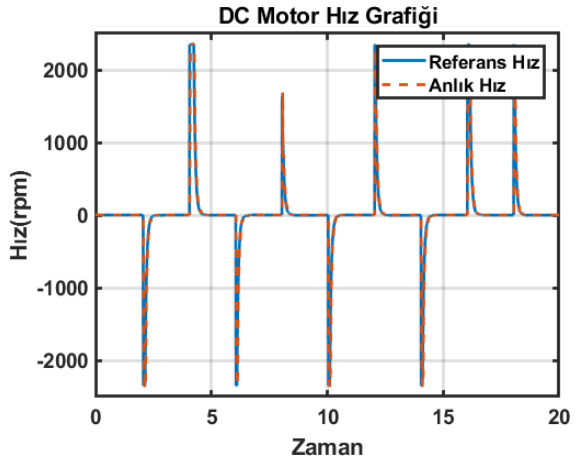
Şekil 9 ve 10 basamak girişine karşılık elde edilen sistem konum ve hız yanıtlarını göstermektedir. Grafiklerde sistem çıkışı ile birlikte referans sinyali

de görülmektedir. Grafiklerden görüleceği üzere sisteme enerji verildikten yaklaşık 1 sn. sonra referans sinyal uygulanmış ve sistem çıkışları referans sinyalleri başarılı bir şekilde takip etmiştir.

İkinci olarak ise kontrolcünün performansının farklı karakteristiğe sahip bir referans sinyal ile test edilebilmesi için sisteme rastgele basamak girişleri uygulanmıştır. Bu girişe ait sistem yanıtları, referans sinyalleri ile birlikte Şekil 11 ve 12'de görülmektedir. 20 sn. boyunca 2 sn. aralıklarla sisteme rastgele değerlerde basamak girişi uygulanmış ve sistem çıkışlarının referans sinyalleri takip ettiği gözlenmiştir.

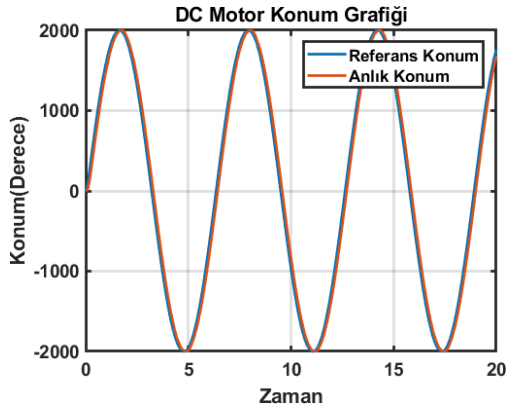


Şekil 11. Rastgele giriş / sistem konum yanıtı.

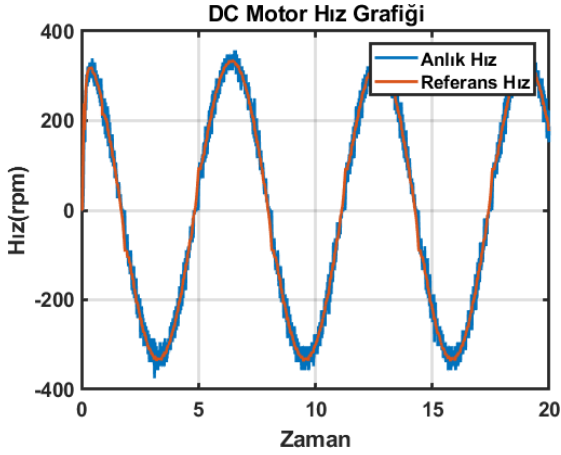


Şekil 12. Rastgele giriş / sistem hız yanıtı.

Son olarak sisteme sinüs dalgası, referans olarak verilmiştir. Şekil 13 ve 14'de referans ve sistem yanıtları görülmektedir. Sistem yanıtlarından anlaşılacağı üzere kontrolcü diğer iki referans girişinde olduğu gibi sinüs dalga referans giriş takibinde de başarılı bir performans sergilemiştir. Çıktılardan görüldüğü üzere DC motor konum-hız kontrolü deneysel gözlemler sonucu elde edilen PID parametreleri ile başarı ile sağlanmıştır.



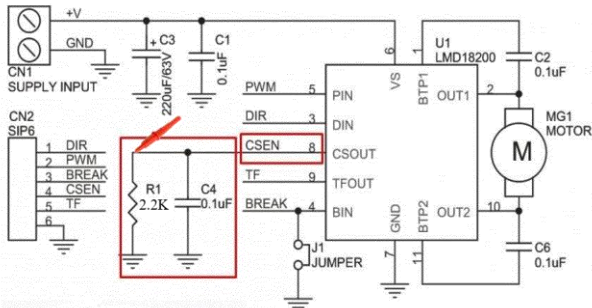
Şekil 13. Sinüs giriş / sistem konum yanıtı.



Şekil 14. Sinüs giriş / sistem hız yanıtı.

4.2 Konum – hız – tork kontrolü

DC motor konum-hız kontrolü gerçekleştirildikten sonra kaskad yapıya akım döngüsü eklenmiştir. Konum, hız ve tork kontrolü gerçekleştirmek için oluşturulan kaskad denetleyici yapısı Şekil 18’ de verilen Simulink uygulama modelinde görülmektedir.

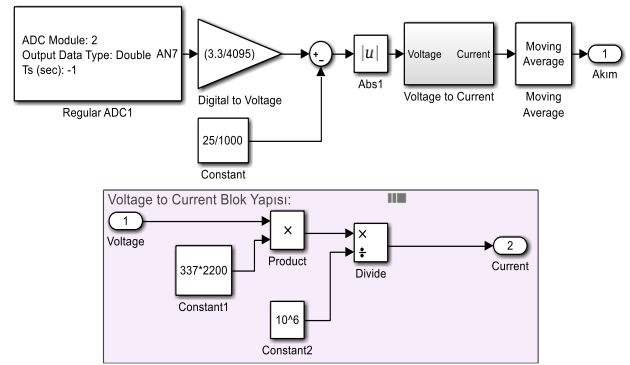


Şekil 15. LMD18200 motor sürücü şeması.

Akım geri beslemesi için devre şeması Şekil 15’de görülen LMD18200 motor sürücü kartının “Current Sense Output” pininden alınan gerilim çıkışını, STM32F4 mikro denetleyicisinin ADC birimi ile dijitalle çevirip gerekli matematiksel dönüşümlerin yapılması gerekmektedir.

LMD18200 motor sürücü kartının 8.pini $337 \mu A / A$ çıkış sinyali vermektedir [6]. Şekil 15’deki şemada görüldüğü üzere CSEN pin çıkışı (akım sinyali) $2.2K \Omega$ değerindeki direnç üzerinden toprağa iletilmektedir. Böylece bağlanan direnç üzerindeki gerilim voltmetre ile kolaylıkla ölçülebilmektedir. Ölçülen gerilim (V) değeri akım bilgisini de içermektedir.

Şekil 16’da akım değerini ölçmek için Simulink uygulama modelinde kullanılan blok yapısı görülmektedir. Blok yapısının giriş sinyalini, CSEN pininden ölçülen gerilim değeri oluşturmaktadır.

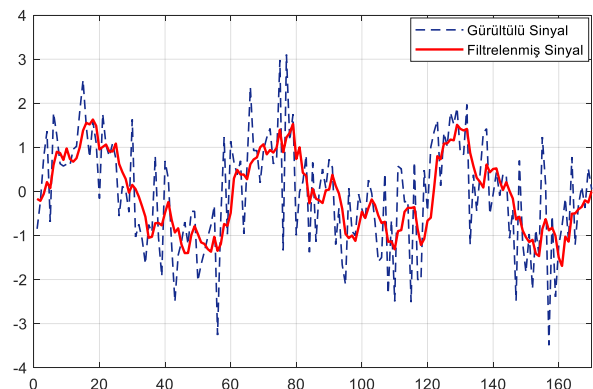


Şekil 16. Akım Bilgisinin ADC birimi ile okunması.

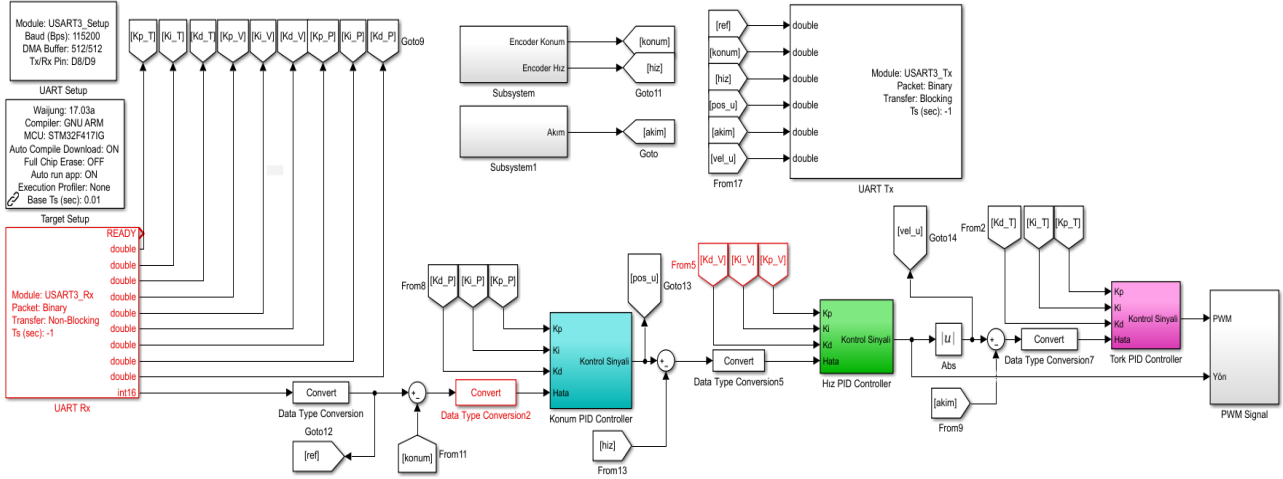
Ölçülen dijital değer gerekli matematiksel işlemler yapılarak akım bilgisine dönüştürülür. DC motordan ölçülen akım üzerinde gürültüler olduğundan ölçüm çıkışına “Hareketli Ortalama” filtresi uygulanarak gürültü seviyesi bir miktar giderilmiştir.

Hareketli Ortalama Filtresi (*Moving Average Filter*), giriş sinyalini gürültülerden temizlemek için tercih edilebilecek basit bir yumuşatma filtresidir. Filtrenin çalışma yapısı, Denklem (10)’da görüldüğü üzere çıkışın sürekli giriş değerlerinin ortalamasının alınması şeklindedir.

$$y(k) = \frac{[u(k) + u(k-1) + \dots + u(k-n+1)]}{n} \quad (10)$$



Şekil 17. Hareketli ortalama filtresinin uygulanması



Şekil 18. Konum-Hız-Tork kontrolü Simulink modeli.

Filtrenin uygulanması için önce filtre boyutu (n) kadar veri saklanmakta ve bu verilerin aritmetik ortalaması çıkış olarak kullanılmaktadır. Her yeni veri geldiğinde, en eski değer çıkarılarak yerine bu yeni veri koyulmakta ve tekrar ortalamasının alınması şeklinde devam etmektedir. Şekil 17’de hareketli ortalama filtresinin gürültülü bir sinyale etkisi görülmektedir.

DC motorun tork bilgisi elde edilmek istenirse Denklem (3)’deki matematiksel işlem yapılabilir. K_t motor tork sabitidir. Şekil 18’de gösterilen kontrol yapısında kaskad PID denetleyicisine akım döngüsü eklenmiştir. Konum-hız kontrolünde olduğu gibi PID parametreleri deneysel gözlemlere dayanılarak önce iç döngünün (akım döngüsü) daha sonra bir sonraki döngünün (hız döngüsü) ve son olarak en dış döngünün (konum döngüsü) kontrol parametreleri belirlenip PID denetleyicisine girilmiştir. Deneysel gözlemler sonucunda bulunan PID katsayıları Çizelge 4’de verilmiştir.

Çizelge 2. Deneysel gözlemler sonucu bulunan kontrol katsayıları.

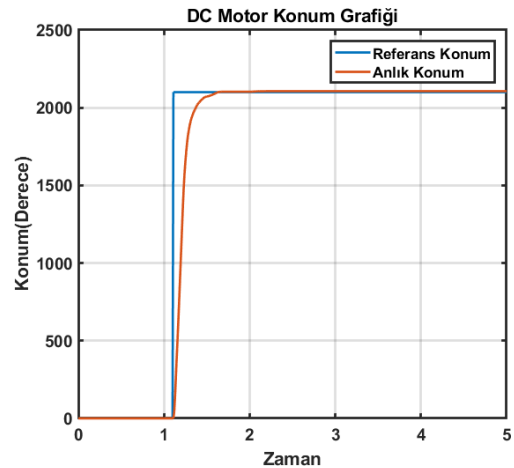
Parametre	Konum Kontrol	Hız Kontrol	Tork Kontrol
K_p	2.26	0.015	1.0275
K_i	0.0511	0.625	0.0008
K_d	0.025	0.000001	0

Şekil 8’de görülen ana bilgisayar modeli kaskad yapıya eklenen döngüye göre güncellenmiştir. Modele, tork PID denetleyici parametreleri ve sistem tork çıkışı ilave edilmiştir. Güncellenen model aracılığıyla gerçek zamanlı olarak deney düzeneğine referans giriş sinyali uygulanmış ve sistem yanıtları (DC Motor konum, hız ve tork bilgisi) aynı model

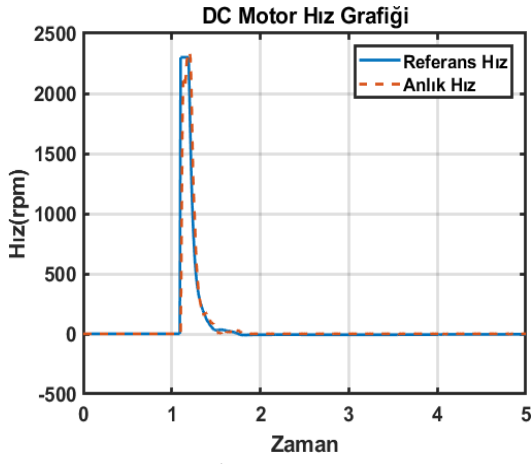
üzerinden incelenerek veri toplama işlemi gerçekleştirilmiştir.

Şekil 19, 20 ve 21 sırasıyla basamak girişine karşılık elde edilen sistem konum, hız ve tork yanıtlarını göstermektedir. Grafiklerden görüleceği üzere sisteme referans sinyal uygulandıktan sonra sistem çıkışları referans sinyalleri başarılı bir şekilde takip etmiştir.

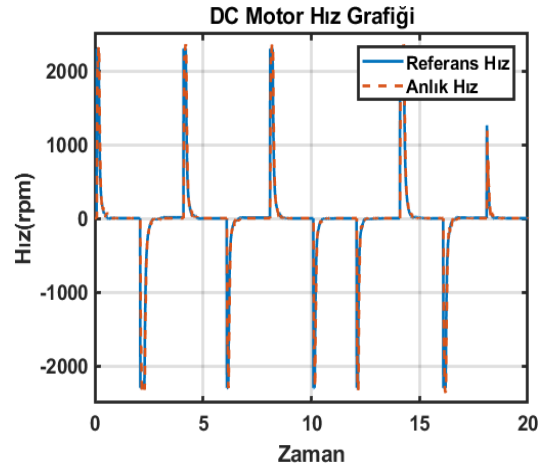
Konum ve hız verileri, doğrudan ölçülebilir sinyallerdir; tork sinyali ise akım verisi üzerinden dolaylı olarak ölçülmektedir. Dolayısıyla akım verisinin tork kontrol performansı üzerindeki etkisi açıktır. Ölçülen - gürültülü akım sinyali ile filtrelenmiş akım sinyali arasında genlik farkı ve kaymalar (Şekil 17) oluşabilmektedir. Oluşan bu durumlar tork kontrol performansını da doğrudan etkilemektedir. Bu etki, DC motor tork grafiklerinde (Şekil 21, 24, 27) yüzde aşım ve kalıcı durum hatası olarak açıkça görülmektedir.



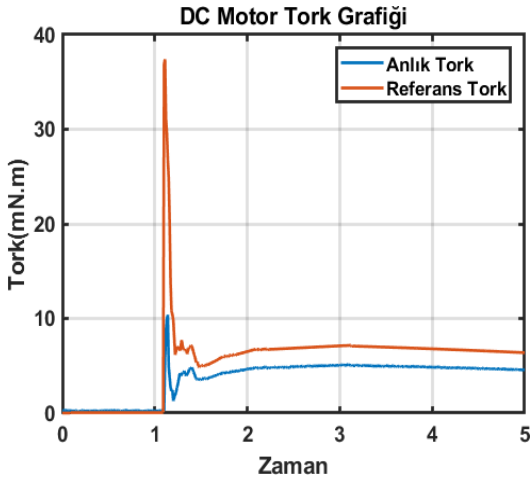
Şekil 19. Basamak giriş / sistem konum yanıtı.



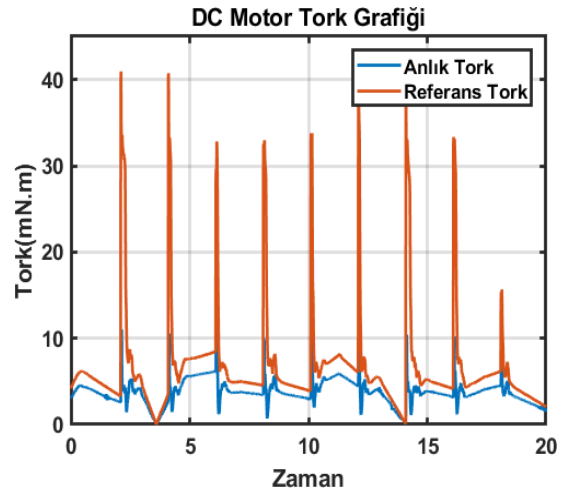
Şekil 20. Basamak giriş / sistem hız yanıtı.



Şekil 23. Rastgele giriş / sistem hız yanıtı.

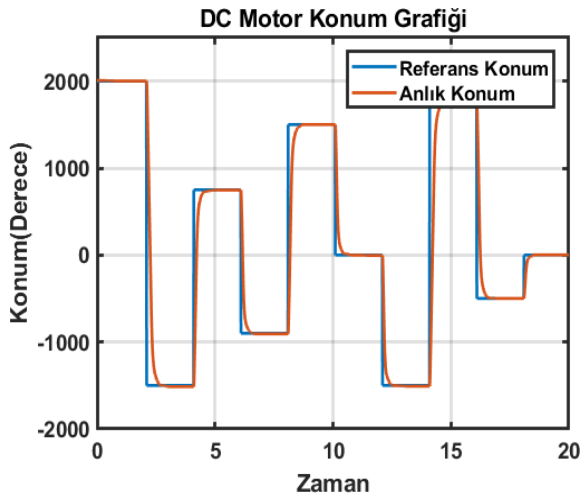


Şekil 21. Basamak giriş / sistem tork yanıtı.



Şekil 24. Rastgele giriş / sistem tork yanıtı.

Konum-hız kaskad PID denetleyici deneylerinde olduğu gibi konum-hız-tork kaskad PID denetleyicisinin test edilmesi için ikinci olarak sisteme rastgele değerlerde 2 sn. aralıklarla basamak girişler uygulanmıştır. Bu girişe karşılık sistemden alınan konum, hız ve tork yanıt grafikleri sırası ile Şekil 22, 23 ve 24'de görülmektedir.



Şekil 22. Rastgele giriş / sistem konum yanıtı.

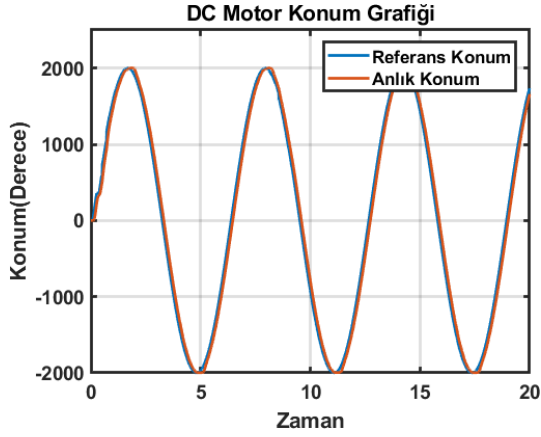
Son olarak sisteme sinüs dalgası, referans olarak verilmiştir. Şekil 25, 26 ve 27'de referans ve sistem yanıtları görülmektedir. Sistem yanıtlarından anlaşılacağı üzere kontrolcü diğer iki referans girişinde olduğu gibi sinüs dalga referans giriş takibinde de başarılı bir performans sergilemiştir.

Tork kontrolünün sistemden ölçülen akım sinyaline bağlı olduğu düşünülecek olursa, akımın yüksek hassasiyetle ölçülmesinin oldukça önemli olduğu ortaya çıkmaktadır. Kullanılan motor sürücü kartı, fiziki bağlantılar vb. gibi gerçek zamanlı uygulamalarda karşılaşılabilecek durumlar bu ölçümde olumsuz etkiye sebep olabilmektedir. Özellikle ölçülen sinyalde oluşan gürültü bahsedilen etkilerin başında gelmektedir. Elde edilen gerçek zamanlı sistem tork yanıtlarında, tork sinyali doğru davranışı sergilemekle birlikte akım sinyalindeki gürültüler kontrolcünün performansına olumsuz etki etmiştir. Aynı etki konum ve hız yanıtlarında söz konusu değildir.

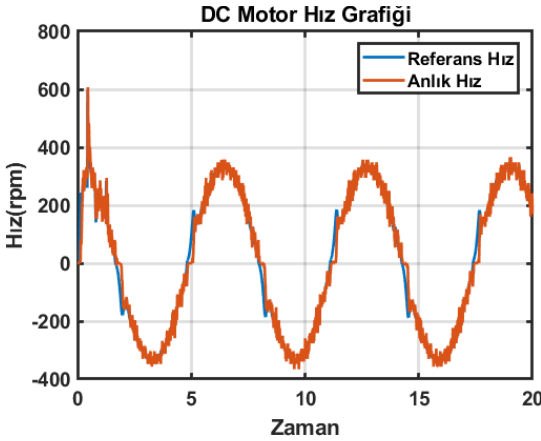
Örnek bir karşılaştırma olması açısından her iki kaskad denetleyici için basamak giriş – sistem konum yanıtının karakteristikleri Çizelge 5’de verilmiştir.

Çizelge 3. Basamak giriş – sistem konum yanıtının karşılaştırılması.

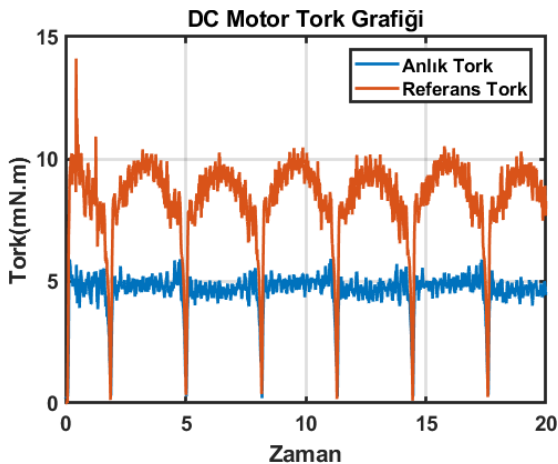
Denetleyici	Yüzde Aşım	Yükselme Zamanı	Kalıcı Durum Hatası
Konum-Hız	% 0	0.2	% 0.4
Konum-Hız-Tork	% 0	0.4	% 0.6



Şekil 25. Sinüs giriş / sistem konum yanıtı.



Şekil 26. Sinüs giriş / sistem hız yanıtı.



Şekil 27. Sinüs giriş / sistem tork yanıtı.

5. Sonuçlar

Bu çalışmada DC motorun konum, hız ve tork kontrolü için kaskad PID denetleyici tasarlanmış ve gerçek zamanlı olarak uygulanmıştır. Gerçek zamanlı uygulamalarda öncelikle konum-hız döngüsü içeren kaskad PID yapısı oluşturulmuş ve konum-hız kontrolü yapılmıştır. Kontrol katsayıları deneysel gözlemlere dayanılarak belirlenmiş olup, ana bilgisayar (host) model aracılığıyla PID denetleyicisine verilmiştir. Daha sonrasında sisteme akım döngüsü de eklenip konum, hız ve tork kontrolü için kaskad PID denetleyici yapısı oluşturulmuş ve gerçek zamanlı olarak uygulanmıştır. Yine bu uygulamada da PID katsayıları deneysel verilere dayanılarak belirlenmiştir. Denetleyici performanslarının incelenmesi için sisteme gerçek zamanlı olarak basamak, sinüs ve rastgele referans giriş verilmiş ve yanıtlar gözlenmiştir.

Tasarlanan konum-hız ve konum-hız-tork kaskad denetleyici yapılarında özellikle konum ve hız denetleyici döngüleri başarılı bir kontrol performansı sergilemiştir. Tork kontrolünde ise akım sinyalindeki gürültüler kontrolcü performansını olumsuz etkilemiştir. Gelecek çalışmalarda bu konu üzerinde durulup, denetleyici performansının geliştirilmesi hedeflenmektedir.

6. Kaynaklar

- Bacac, N., Slukic, V., Puskaric, M., Stih, B., Kamenar, E., Zelenika, S., 2014. Comparison of Different DC Motor Positioning Control Algorithms. *In Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1654–59.
- Bhavina, R., Jamliya, N. and Vashishtha, K., 2013. Cascade Control Of DC Motor with Advance Controller. *International Journal of Industrial Electronics and Electrical Engineering*, 1:1, 18–20.
- Bingül, Z. ve Küçük, S., 2017. Robot Dinamiği ve Kontrolü, Umutepe Yayınları, 10-25.
- Gross, H., 1983. *Electrical Feed Drives for Machine Tools*, Wiley, 40-69.
- Gücin, T. N., Biberoglu, M., Fincan, B. and Gülbahçe, M. O., 2015. Tuning Cascade PI(D) Controllers in PMDC Motor Drives: A Performance Comparison for Different Types of Tuning Methods. *In Proceedings of*

the 9th International Conference on Electrical and Electronics Engineering (ELECO), 1061–66.

Karadeniz, M., İskender, İ. ve Yüncü, S., 2014. Doğru Akım Motor Hızının Uyarlamalı Kutup Yerleştirme Denetimi. *Gazi Üniv. Müh. Mim. Fak. Der.*, **19:3**, 327–34.

Kelekçi, E., Yaren, T. and Kizir, S., 2019. Model Based PID Controller Design of a DC Motor Using Black Box System Identification. *In Proceedings of the 10th International Conference on Image Processing, Wavelet and Applications (IWW2019)*, 166–175.

Kizir, S., Yaren, T. and Kelekçi, E., 2019. Matlab Simulink Destekli Gerçek Zamanlı Kontrol: Teori ve Mühendislik Uygulamaları, Seçkin Yayıncılık, 188-189.

Patel, L. S. and Dave, K. C., 2011. Cascade Control Technique for DC Motor Speed Control. *In Proceedings of the International Conference on Science and Engineering (ICSE 2011)*, 599-603.

Robet, P. P. and Gautier, M., 2014. Cascaded Loops Control of DC Motor Driven Joint Including an Acceleration Loop. *IFAC Proceedings*, **47:3**, 7227–7232.

Kumar, P., 2013. Cascaded PID Controller Design for Heating Furnace Temperature Control. *IOSR Journal of Electronics and Communication Engineering*, **5:3**, 76–83.

Zhuang, M., 1994. Optimum Cascade PID Controller Design for SISO Systems. *In Proceedings of the International Conference on Control'94*, 606–611.

İnternet kaynakları

1) <https://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition§ion=SimulinkModeling>, (15.12.2021)