

Kapalı çevrim PID kontrolör tasarımında birim basamak cevabı çoklu ölçüt performans haritalaması

Furkan Nur Deniz^{*1}, Cemal Keleş², Barış Baykant Alagöz³, Nusret Tan⁴

19.11.2013 Geliş/Received, 03.02.2014 Kabul/Accepted

ÖZ

Bu çalışmada, (k_p, k_i, k_d) hiperküpünden örneklenen PID kontrolörleri için kapalı çevrim birim basamak cevabı çoklu ölçüt ortalama performans haritalaması sunulmuştur. Kontrol sistemlerinin birim basamak cevaplarının analizi için; karesel hata, maksimum aşım, yükselme ve yerleşme zamanları ölçütlerinin ağırlıklı ortalaması ile ifade edilen çoklu ölçüte dayalı bir performans değerlendirmesi kullanılmıştır. Böylece, (k_p, k_i, k_d) PID tasarım hiperküpü örnekleme ile elde edilen performans haritaları değerlendirilerek, bu hiperküp içinde en iyi PID tasarımı elde edilmiştir. Performans haritalamasının, bir PID tasarım hiperküpü için çoklu ölçüte dayalı performans durumlarını ortaya koyabildiği ve bu haritalarının kontrol sisteminin performans analizi ve tasarımı için önemli bilgiler sağlayabildiği görülmüştür.

Anahtar Kelimeler: PID kontrolör, birim basamak cevabı, performans haritası

Multi-criteria performance mapping of unit step response in closed loop PID controller design

ABSTRACT

In this study, a multi-criteria average performance mapping of closed-loop unit step response for PID controllers sampled from (k_p, k_i, k_d) hypercube is presented. A multi-criteria performance evaluation method based on the weighted average of mean squared error, maximum overshoot, rise and settling time criteria is used for the analysis of unit step responses of control systems. Thus, the best PID controller is designed by evaluating the performance maps obtained from sampled (k_p, k_i, k_d) design hypercube. It is observed that the performance mapping demonstrates the performance status of PID controllers for a PID design hypercube, and it can provide useful information for performance analyses and the design of control systems.

Keywords: PID controller, unit step response, performance mapping

* Sorumlu Yazar / Corresponding Author

1 İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Malatya - furkan.deniz@inonu.edu.tr

2 İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Malatya - cemal.keles@inonu.edu.tr

3 İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Malatya

4 İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Malatya - nusret.tan@inonu.edu.tr

1. GİRİŞ (INTRODUCTION)

Kontrol sistemi tasarım problemleri genellikle iki açıdan ele alınmaktadır; kararlılık ve optimizasyon [1]. Gelişmiş kontrol sistemleri tasarımı için birçok çalışmada çözüm yolları önerilmesine karşın pratikte endüstriyel kontrol sistemlerinin büyük bir kısmı hala geleneksel PID kontrolör yapısı tasarımına dayanmaktadır [2]. Bunun başlıca nedeni, geleneksel PID kontrolör yapısının basit ve kararlı çözümler sunabilmesidir. Buna rağmen, pratik PID kontrol tasarımı uygulamalarında çözüm bekleyen sorunlar güncelliğini korumaktadır. Endüstriyel proseslerin kontrolü için tasarlanan PID kontrolör parametrelerinin optimal değerlerde seçilmiş olması ve mümkün olduğunca geniş bir parametre değişim aralığında sistemin kararlılığını sürdürebilmesi gerekmektedir. Özellikle parametre belirleme yöntemlerinin birçoğunda manuel tekniklerin kullanılması, en iyi parametrelerin belirlenmesini hem zor, hem de zaman alıcı kılmaktadır [3]. Kontrol sistemi devreye girmeden önce kontrolör parametrelerinin güvenilir olarak ayarlanması gerekebilmektedir [4]. Bu itibarla, PID kontrolörlerin performanslarını iyileştirebilecek pratik yöntemlerin geliştirilmesi önem arz etmektedir.

Kontrolör parametrelerini hesaplama ve ayarlama (tuning) teknikleri için önerilen yöntemler, analitik ve nümerik çözümlere dayanmaktadır. Analitik çözüm yöntemleri, kağıt üstünde yapılabilecek kontrolör tasarım uygulamalarında büyük öneme sahiptir. Ancak bilgisayar destekli tasarım yapılırken nümerik yöntemler ön plana çıkar. Nümerik yöntemlerin işlem maliyeti analitik çözümlere kıyasla fazla olmasına rağmen, bilgisayarların bilgi işlem hızının artmış olması, nümerik çözüm tekniklerinin gelişimini mümkün kılmıştır. Analitik çözümler sınırlı sayıda değişkenin, belirli değer ve koşulları için analiz olanağı sunmakta iken nümerik çözümler çok sayıda değişkenin çok sayıda durumu ve değeri için analiz imkânı sağlar. Bu ise tasarımda etkin rol oynayan parametrelerin, geniş bir aralıktaki değişimleri için kontrol performansının değişimini değerlendirmeye imkân sağlar.

Performans haritaları, kontrolör tasarım katsayılarının belirli bir değişim aralığında kontrolör performansının dağılımını gösterir. Bu dağılım, performansın düştüğü ya da yükseldiği bölgelere işaret eder. Böylece, katsayıların en iyi performans sergilediği katsayı değişim bölgeleri elde edilebilir ve kontrolör tasarımı bu bölgeden seçilen katsayılar ile gerçekleştirilebilir. Sezgisel (heuristic) optimizasyon yöntemleri, iyi sonuca yakınsayabilen fakat optimal çözümü verebilmesi her zaman kesin olmayan yöntemlerdir. Çünkü bir çok optimizasyon tekniğinin en önemli yetersizliği, çalıştığı optimizasyon bölgesi içinde verdiği çözümün en iyi çözüm olduğuna

garanti edememesidir. Diğer bir ifade ile, arama bölgesi içinde global ekstremum noktaları yerine yerel ekstremumlara takılma riski her zaman söz konusudur. Kontrolör performans haritaları ise, katsayıların değişim bölgesini düzenli olarak örneklemediği için, optimal performans noktalarını daha yüksek bir doğrulukla tespit edebilmektedir. Bu nedenle, bilgisayar destekli kontrolör tasarımında, çok karmaşık sistem fonksiyonlarına dahi kolaylıkla uygulanabilen performans haritalaması, aranan bölge içinde en iyi kontrolör parametrelerine daha etkili ve güvenilir olarak ulaşılabilmesini sağlamaktadır.

PID kontrolöre ait parametrelerin ayarlanmasına ilişkin maksimum yüzde aşım, yerleşme zamanı, yükselme zamanı, ortalama karesel hata gibi birim basamak cevabı performans ölçütlerini dikkate alan birçok yöntem önerilmiştir. Bu yöntemlerden bazıları optimizasyon işleminde bu ölçütlerden sadece birini dikkate alırken, bazıları da bahsedilen ölçütlerden bir kaçını birlikte göz önünde bulundurur. Bu yöntemlerin çoğu önceden belirlenmiş kriterlere bağlı olarak sistem için en iyi olan PID parametrelerinin belirlenmesini sağlamaktadır [5]. Literatürde, PID kontrolör parametreleri iyileştirme yöntemi olarak Ziegler-Nichols ve CHR (Chien-Hrones-Reswick) kuralları sıklıkla kullanılmıştır [2]. Ancak bu yöntemler daha çok temel PID kontrol yapıları için önerilmiş olup, doğrusal olmayan kontrolör yapıları için optimal çözümler sunmamaktadır. Referans [6]'da optimal PID parametrelerinin bulunması için Parçacık Küme Optimizasyonu (Particle Swarm Optimization-PSO) önerilmiştir. Bu yöntemde yüzde aşım, yerleşme zamanı, yükselme zamanı gibi ölçütler için optimizasyon işlemini gerçekleştirilmiştir. Referans [7]'de Genetik algoritma yardımı ile çoklu ölçütlü PID kontrolör optimizasyonu gösterilmiştir. Diğer bir çalışmada kontrolör performansının değerlendirilmesi ile PID kontrolörün otomatik olarak katsayılarını ayarlaması sağlanmıştır [8].

Bu çalışmada, bilgisayar destekli analiz ile PID kontrol sistemi birim basamak cevabının maksimum aşım, yerleşme zamanı, yükselme zamanı ve ortalama karesel hata ölçütlerine dayalı ağırlıklı ortalama performans haritası çıkarılmıştır. Önerilen performans haritalama metodu, sistem karmaşıklığından bağımsız olarak karmaşık kontrol sistemine kolaylıkla uygulanabilmektedir. Bu avantaj, kontrolör tasarım problemini önemli ölçüde kolaylaştırmış ve sisteme göre analitik çözümleme ihtiyacını ortadan kaldırmıştır. Bu makalede, önem derecesine bağlı ağırlıklandırılmış performans ölçütlerine göre katsayıların değişim bölgesi içinde en iyi (k_p , k_i , k_d) katsayılarının elde edilmesi gösterilmiştir.

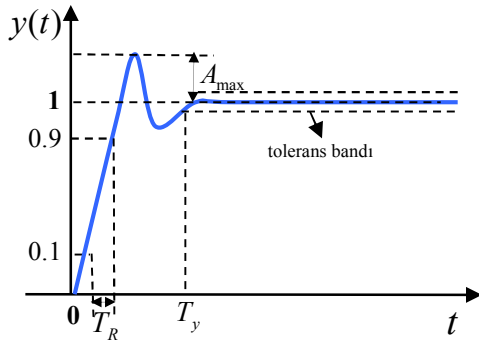
Çalışma içerisindeki bölümler şu şekilde düzenlenmiştir: 2. Bölümde birim basamak performans ölçütlerinin

belirlenmesi ve ağırlıklı ortalama performans haritasının çıkarılması için önerilen yöntem anlatılmıştır. 3. Bölümde yöntem, bir örnek üzerinde uygulamalı olarak gösterilmiş, performans haritaları çıkarılmış ve elde edilen en uygun PID katsayıları için birim basamak cevabı sunulmuştur. 4. Bölümde önerilen yöntemin işlem maliyetinin analizi yapılmış ve işlem maliyetini azaltmaya dönük çözümler önerilmiştir.

2. YÖNTEM (METHOD)

2.1. Birim Basamak Performans Ölçütleri (Unit Step Performance Criteria)

Kontrol sisteminin birim basamak cevabı performansının değerlendirilmesinde yaygın olarak dört temel performans ölçütü kullanılmaktadır. Bunlar, yükselme zamanı, maksimum aşım, yerleşme zamanı ve birim basamak ortalama karesel hata parametreleridir. Örneklenmiş bir birim basamak sistem cevabı için, bu dört performans parametresinin bilgisayar destekli analizi, istenen performans özelliklerini sergileyebilen kontrolör tasarımında önemli kolaylıklar sağlamaktadır. Bu bölümde, bu performans ölçütlerine ait hata fonksiyonları tanıtılmıştır.



Şekil 1. Birim basamak cevabı performans ölçütü parametrelerinin temsili gösterimi (Representation of performance criteria parameters for unit step response)

a) Maksimum Aşım Performans Ölçütü

Kontrol sistemi çıkışının hedeflenen noktaya yerleşmeden önce, maksimum aşım miktarını ölçer. Kontrol uygulamalarında maksimum aşımın %20 den küçük olması istenir [9, 10]. Ancak, aşımın tolere edilemediği kritik kontrol uygulamalarda, aşımın mümkün olduğunca küçük olması istenir. Bir kontrol sisteminin birim basamak cevabının $(y(t))$, Δt örnekleme periyodu ile örnekleme durumu da, ayrık domainde $y(n) = y(n\Delta t)$ ile gösterilsin. Bu birim basamak cevabı için maksimum aşım,

$$A_{\max} = \max\{y(n)\} - 1, \quad n = 1, 2, 3, \dots, n_s \quad (1)$$

ile hesaplanır [9]. Burada n_s birim basamak cevabı örnek sayısıdır. Mutlak fark hata ölçütü ile aşım hatası (E_A) ise maksimum aşım hatasına (A_{\max}) bağlı olarak,

$$E_A = |H_A - A_{\max}| \quad (2)$$

ile ifade edilmiştir. Burada, H_A hedeflenen aşım miktarını gösterir. Maksimum aşım Şekil 1'de temsili olarak gösterilmiştir. Bilgisayar destekli hesaplama için programlama dili bağımsız (raw) örnek kod Program 1'de sunulmuştur:

```
MaksimumAşım = (max(y) - 1);
MaksimumAşımHatası = Mutlak_Deger
(HedefAşım - MaksimumAşım);
```

Program 1: E_A hesaplaması için programlama dili bağımsız örnek kodu

b) Yerleşme Zamanı Ölçütü

Kontrol sistemi çıkışının hedeflenen yerleşme bandına girene kadar geçen zaman periyodunu ölçer [9, 10]. Kontrol uygulamalarında maksimum aşım için yerleşme bandı genellikle %105-%95 olarak kabul görür. Yerleşme zamanı, sistem cevabının arzu edilen noktaya yerleşebilmesi için ihtiyaç duyulan zamanı ölçer ve kontrol sisteminin cevap süresi hakkında bilgi verir. Hızlı cevap gereken kontrol uygulamalarında yerleşme zamanı mümkün olduğunca küçük belirlenmelidir. Birim basamak cevabı için yerleşme zamanı,

$$T_y = (n_s - n_z)\Delta t, \quad \text{burada } 0.95 \leq y(n) \leq 1.05 \wedge n \geq n_z \quad (3)$$

ile ifade edilebilir. Burada n_z , kontrol sistemi cevabının hedeflenen yerleşme bandına yerleşene kadar geçen örnek sayısıdır. Yerleşme zamanı Şekil 1'de temsili olarak gösterilmiştir. Yerleşme zamanı hatası (E_z) ise mutlak fark hata ölçütü kullanılarak,

$$E_z = |H_z - T_y| \quad (4)$$

ile ifade edilmiştir. Burada H_z hedeflenen yerleşme zamanını göstermektedir. Yerleşme zamanı hatası, bilgisayar destekli hesaplama için programlama dili bağımsız (raw) örnek kod aşağıda sunulmuştur:

```

nz=0;
Döngü t=1:ns
    Eğer y(t)>0.95 ve y(t)<1.05
        nz= nz+1;
    Diğer
        nz= 0;
    EğerSonu
    Eğer nz==0
        YerleşmeZamanı=
            ToplamBirimBasamakSüresi;
    Diğer
        YerleşmeZamanı=
            ToplamBirimBasamakSüresi
            -nz*ÖrneklemePeriyodu;
    EğerSonu
DöngüSonu
YerleşmeHatası= Mutlak_Değer
(HedefYerleşme-YerleşmeZamanı);

```

Program 2: E_z hesaplaması için programlama dili bağımsız örnek kod

c) Yükselme Zamanı Ölçütü

Kontrol sistemi çıkışının hedeflenen değerin %10'dan %90'a ulaşmaya kadar geçen süre yükselme zamanı olarak [9, 10]. Yükselme zamanı, sistem cevabının hızlı değişimlere uyum kabiliyetini ölçer. Sürekli değişkenlik gösteren kontrol işaretlerinin başarılı bir şekilde takibi için sistemin yükselme zamanının mümkün olduğunca küçük olması istenir. Birim basamak cevabı için yükselme zamanı,

$$T_R = (n_u - n_d)\Delta t, \text{ burada } 0.90 \leq y(n_u) \wedge 0.10 \leq y(n_d) \quad (5)$$

ile ifade edilebilir. Burada n_u ve n_d hedeflenen değerlerin %10 ve %90 ulaşılabilmesi için gereken örnekleme sayısıdır. Yükselme zamanı Şekil 1'de temsili olarak gösterilmiştir. Yükselme zamanı hatası (E_R) ise mutlak fark hata ölçütü kullanılarak,

$$E_R = |H_R - T_R| \quad (6)$$

ile ifade edilmiştir. Burada H_R hedeflenen yükselme zamanı hatasını göstermektedir. Yükselme zamanı hatası, bilgisayar destekli hesaplama için programlama dili bağımsız (raw) örnek kod aşağıda sunulmuştur:

```

alt=0; % Kontrolör cevabının %10
ilk geçiş durumu
Döngü t=1:length(y)
    Eğer y(t)>0.10 ve alt==0
        nd=t;
        alt=1;
    EğerSonu
    Eğer y(t)>0.90
        nu=t;
        break;
    EğerSonu
DöngüSonu
YükselmeZamanı=(nu-nd)*
BirimBasamakÖrneklemeAdımı;
YükselmeHatası(i,j,k)=abs
(HedefYükselme-YükselmeZamanı);

```

Program 3: E_R hesaplaması için programlama dili bağımsız örnek kod

d) Birim Basamak Cevabı Ortalama Karesel Hata Ölçütü

Kontrol sistemi çıkışının birim basamak fonksiyonuna göre ne ölçüde farklılaştığını gösteren ve optimizasyon problemlerinde yaygın olarak kullanılan bir hata ölçütüdür. Kontrolör optimizasyon algoritmalarında, minimize edilmeye çalışılan ortalama karesel hata ölçütü, minimum değer olan sıfır değerinde kontrol sistemi çıkışının ideal birim basamak cevabı olduğuna işaret eder. Pratikte bu durum sistem gecikmeleri nedeni ile olanaksızdır. Sürekli değişkenlik gösteren kontrol işaretlerinin başarılı takibi ortalama karesel hatanın minimize edilmesi ile sağlanabilmiştir [3, 6]. Ortalama karesel hata (E_2),

$$E_2 = \frac{1}{n_s} \sum_{n=0}^n (y(n) - r(n))^2 \quad (7)$$

ile ifade edilebilir. Birim basamak cevabı ortalama karesel hatayı bilgisayar destekli hesaplamak için programlama dili bağımsız (raw) örnek kod aşağıda sunulmuştur:

```

OrtalamaKareselHata=
Toplam((y(t)-1).^2)/n_s;

```

Program 4: E_2 hesaplaması için programlama dili bağımsız örnek kod

2.2. Ağırlıklı Ortalama Performans Haritasının Çıkarılması (Mapping of Weighted Average Performance)

PID kontrolörler tasarımı, (k_p, k_i, k_d) katsayıları ile temsil edilen üç boyutlu bir parametre uzayında gerçekleştirilir. Bu parametre uzayı $\Omega \in R^3$ ile gösterilsin. Ω uzayında, (k_p, k_i, k_d) parametrelerinin bir alt ve üst değer ile

sınırlanması durumunda ($\tilde{k}_p \in [\underline{k}_p, \overline{k}_p]$, $\tilde{k}_d \in [\underline{k}_d, \overline{k}_d]$, $\tilde{k}_i \in [\underline{k}_i, \overline{k}_i]$) aşağıda ifade edilen bir PID tasarım parametreleri hiperküpu (ψ) elde edilir:

$$\psi = \{(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) \in \Omega, \tilde{k}_p \in [\underline{k}_p, \overline{k}_p] \wedge \tilde{k}_d \in [\underline{k}_d, \overline{k}_d] \wedge \tilde{k}_i \in [\underline{k}_i, \overline{k}_i]\} \quad (8)$$

Burada, ψ hiperküpu içinde elde edilmiş örnekleme PID tasarım ızgarası (grid) için $(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) = (i\Delta x, j\Delta x, v\Delta x)$ $i, j, v = 1, 2, 3, \dots$ yazılabilir. Parametre Δx hiperküp örnekleme adımdır.

PID tasarım ızgarası ($\tilde{\psi}$) alınan bir parametre için ağırlıklandırılmış çoklu ölçüt performans hatası şöyle tanımlanmıştır:

$$E(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) = \alpha_R E_R(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) + \alpha_Z E_Z(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) + \alpha_A E_A(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) + \alpha_2 E_2(\tilde{k}_p, \tilde{k}_i, \tilde{k}_d) \quad (9)$$

Burada α_R , α_Z , α_A ve α_2 parametreleri, birim basamak ağırlık katsayıları E_R , E_Z , E_A ve E_2 performans ölçütleridir. Denklem 9, her bir ölçütün ortalama hatadaki önemini ağırlık parametreleri ile belirler. ψ hiperküpten örnekleme her nokta için, hataların hesaplanması sonucu örnekleme ortalama hata uzayı elde edilir. Bu uzaydan elde edilen her iki boyutlu kesit, bir çoklu ölçüt performans haritasını teşkil eder. Performans ölçütü haritası, çoklu ölçüt performans hatasının, parametre hiperküpu içinde örnekleme (ayrık uzay) dağılımını gösterir. Bu dağılım, PID kontrolörün hedeflenen ortalama birim basamak cevabı performansına, ψ hiperküpu ile tanımlı katsayı bölgesinde ne ölçüde yaklaşabildiğini gösterir ve ψ içinde en iyi PID tasarımı,

$$(k_p, k_i, k_d) = \min_{\tilde{k}_p, \tilde{k}_i, \tilde{k}_d} \{E(\psi)\} \quad (10)$$

ile elde edilebilir. Bilgisayar destekli PID tasarımı ve performans haritalaması için basit işlem adımları şöyle özetlenebilir:

Adım 1: PID tasarım hiperküpten (ψ), parametrelerin belirli bir aralıkta Δx adımı ile örnekleme $\tilde{\psi}$ ızgarası teşkil edilir.

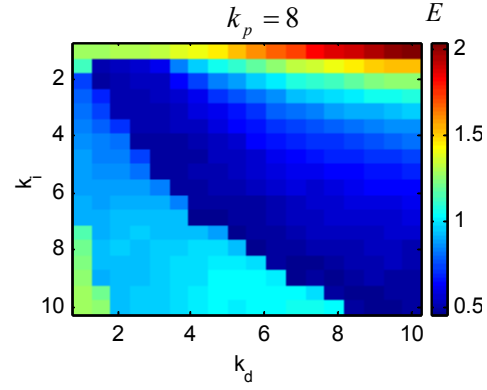
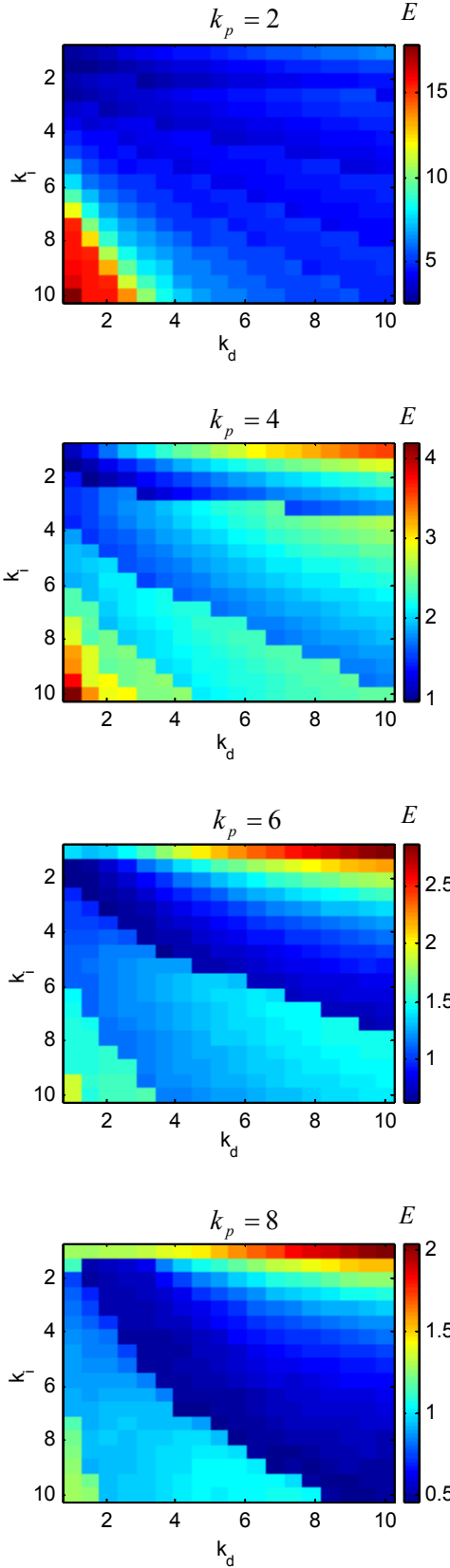
Adım 2: $\tilde{\psi}$ nin her elemanı için Denklem 9 yardımı ile ağırlıklandırılmış çoklu kriter hata uzayı (E) hesaplanır.

Adım 3: Denklem 10 ile ψ hiperküpu içinde en iyi PID katsayıları hesaplanır.

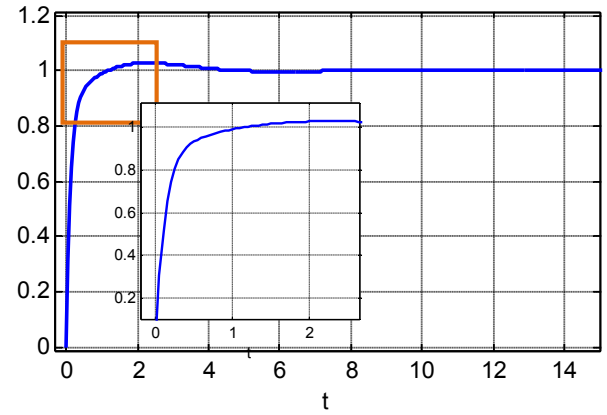
Adım 4: Üç boyutlu E uzayından 2 boyutlu kesitler alınarak, PID kontrolörünün performans haritaları elde edilir.

3. ÖRNEK UYGULAMA (APPLICATION EXAMPLE)

Örnek: Bu örnekte, ikinci dereceden transfer fonksiyonu $G(s) = \frac{1}{s^2 + 2s + 1}$ ile modellenmiş bir sistem için, ($\tilde{k}_p \in [0, 10]$, $\tilde{k}_d \in [0, 10]$, $\tilde{k}_i \in [0, 10]$)—hiperküpu içinde $\alpha_R = 0.5$, $\alpha_Z = 0.3$, $\alpha_A = 0.4$ ve $\alpha_2 = 0.2$ ağırlıklı performans haritaları elde edilmiş ve bu hiperküp içinde optimal PID katsayıları belirlenmiştir. Yükselme zamanı hatası ağırlığını (α_R) çoklu ölçüt performans hatası hesaplamasında en yüksek tutmak sureti ile yükselme zamanı kriterinin performans ölçümlemesinde önemi artırılmış ve böylece elde edilen en iyi PID kontrolörünün hızlı cevap vermesi istendiği vurgulanmıştır. Burada, izin verilebilir ölçüt hataları, $H_A = 0$, $H_R = 0$ ve $H_Z = 0$ alınmıştır. Bu hiç bir birim basamak performans ölçütü için hata toleransımızın olmadığını ifade etmiştir. Şekilde 2’de k_p eksenini üzerinde alınan kesit düzlemlerine göre örnekleme periyodu $\Delta x = 0.5$ için elde edilmiş olan performans haritaları görülmektedir. Bu örnek sistemde k_p parametresinin artan düzlemlerinde çoklu ölçüt performans hatasının düzlem üzerinde azaldığı görülmektedir. Şekil 3’de hiperküp içinde elde edilen en iyi performansı veren PID katsayıları için birim basamak cevabı görülmektedir. Bu performans için $E_R = 0.35$, $E_Z = 0.55$, $E_A = 0.028$ ve $E_2 = 0.002$ dir. Toplam ortalama performans hatası $E = 0.352$ olarak elde edilmiştir.



Şekil 2. $\tilde{k}_p \in [0,10]$, $\tilde{k}_d \in [0,10]$, $\tilde{k}_i \in [0,10]$ ile belirli PID tasarım hiperküpü içinde $\tilde{k}_p = 2,4,5,6,10$ kesit düzlemleri üzerinde elde edilen performans haritaları (Performance maps on $\tilde{k}_p = 2,4,5,6,10$ section planes in PID design hypercube defined with $\tilde{k}_p \in [0,10]$, $\tilde{k}_d \in [0,10]$, $\tilde{k}_i \in [0,10]$)



Şekil 3. $\tilde{k}_p \in [0,10]$, $\tilde{k}_d \in [0,10]$, $\tilde{k}_i \in [0,10]$ ile belirli PID tasarım hiperküpü içinde elde edilen en iyi PID kontrolör katsayıları ($k_p = 10$, $k_i = 9$, $k_d = 7.5$) için birim basamak cevabı (The unit step response for the best PID controller coefficients ($k_p = 10$, $k_i = 9$, $k_d = 7.5$) obtained from PID design hypercube)

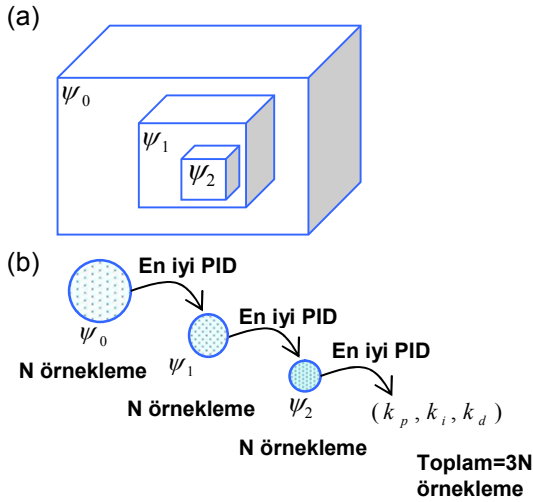
4. İŞLEM MALİYETİ ANALİZİ VE İŞLEM MALİYETİNİN İYİLEŞTİRİLMESİ (COMPUTATION COMPLEXITY ANALYSIS AND IMPROVEMENT OF COMPUTATION COMPLEXITY)

PID tasarımı uzayı (k_p, k_i, k_d) ile teşkil edilen 3 boyutlu uzayda gerçekleştirilmesi nedeni ile performansı hesaplanacak nokta sayısı, sınırları belirlenmiş bir hiperküpün boyutlarının örneklenme sayısına bağlı olarak şöyle ifade edilebilir:

$$N = \left(\frac{k_p - \bar{k}_p}{\Delta x} \right) \left(\frac{k_i - \bar{k}_i}{\Delta x} \right) \left(\frac{k_d - \bar{k}_d}{\Delta x} \right) \quad (11)$$

En iyi PID katsayısını daha hassas belirleyebilmek için Δx azaltılmalıdır. Bu durum örnekleme sayısını üstel olarak artıran bir işlem maliyeti getirmektedir. En iyi PID katsayısını daha düşük işlem maliyeti ve daha yüksek çözünürlük ile elde edebilmek için daralan hiperküp arama yöntemi kullanılabilir. Bu yöntemin adımları şöyle özetlenmiştir:

- (i) Hiperküp ψ_0 , Δx_0 örnekleme adımı ile örneklenir. ψ_0 hiperkübünün en iyi PID noktası belirlenir.
- (ii) En iyi PID noktası yakın civarında bir $\psi_1 \subset \psi_0$ hiperkübü alınır ve $\Delta x_1 < \Delta x_0$ ile örneklenir. ψ_1 hiperkübünün en iyi PID noktası belirlenir ve bu katsayı etrafında alınan yeni bir $\psi_2 \subset \psi_1$ hiperkübü $\Delta x_2 < \Delta x_1$ örnekleme adımı ile örneklenir.
- (iii) Bu işlem istenilen Δx_i çözünürlüğüne ulaşılan kadar tekrarlanır.



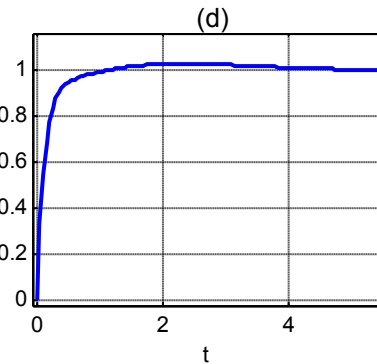
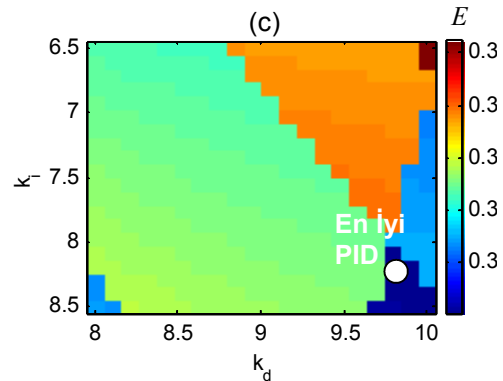
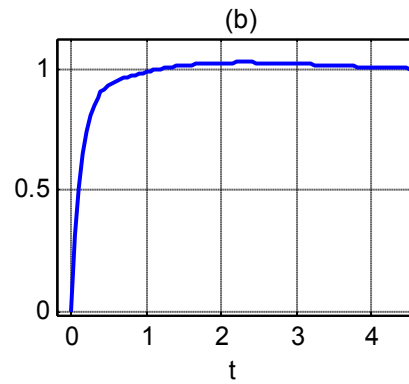
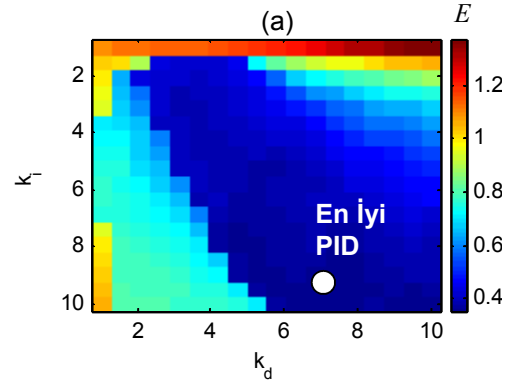
Şekil 4. (a) Daralan hiperküp yönteminde iç içe seçilen PID hiperküpleri ψ_0 , ψ_1 ve ψ_2 (b) işlem adımlarının temsili gösterimi ve örnekleme sayısının toplamsal artması (a) Nested narrowing PID hypercubes ψ_0 , ψ_1 ve ψ_2 in narrowing hypercube search method (b) Representation of the narrowing process and additive increase of sampling number)

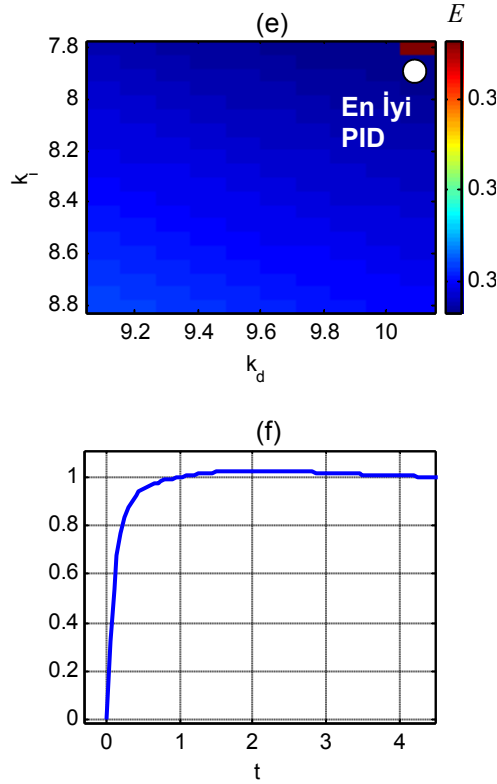
Şekil 4(a)'da daralan hiperküpler temsili olarak gösterilmiştir. Şekil 4(b)'de her bir ψ_i alt hiperkübünün aynı miktarda örneklendiği varsayıldığında, toplam örnek sayısının toplamsal artışı görülmektedir. Daralan hiperküp yönteminde toplam örnekleme sayısı için

$$N_T = \sum_{i=0}^n N_i \text{ yazılabilir. Her bir } \psi_i \text{ hiperkübü}$$

$\Delta x_i = \Delta x_{i-1} / m$, ($m > 1$ bölme oranı) örnekleme adımı

ile örneklenir ise, her bir iç hiperkübün örnekleme adımı $\Delta x_i = \Delta x_0 / m^i$ ile azalır.





Şekil 5. (a)-(b) ψ_0 hiperküpünün $\Delta x = 0.5$ ve $N = 8000$ örnekleme için performans haritası ve en iyi PID için step cevabı. (c)-(d) ψ_1 hiperküpünün $\Delta x = 0.1$ ve $N = 8000$ örnekleme için en performans haritası ve en iyi PID için step cevabı. (e)-(f) ψ_2 hiperküpünün $\Delta x = 0.05$ ve $N = 8000$ örnekleme için performans haritası ve en iyi PID için step cevabı ((a)-(b) Performance map and step response of the best PID controller for $\Delta x = 0.5$ and $N = 8000$ sampling of ψ_0 . (c)-(d); Performance map and step response of the best PID controller for $\Delta x = 0.1$ ve $N = 8000$ sampling of ψ_1 . (e)-(f) Performance map and step response of the best PID controller for $\Delta x = 0.05$ ve $N = 8000$ sampling of ψ_2 .)

Şekil 5’de daralan hiperküp yöntemi ile seçilen PID hiperküpleri ψ_0 , ψ_1 ve ψ_2 için en iyi PID katsayılarını veren performans haritaları ve en iyi PID katsayısı için birim basamak cevapları gösterilmektedir. $\Delta x = 0.05$ örnekleme sıklığında en iyi PID katsayısına, toplamda $N_T = 3 \times 8000 = 24000$ örnekleme ile ulaşılmıştır. Toplam performans hatası ψ_0 için $\Delta x = 0.5$ ile elde edilmiş olan $E = 0.352$ den, ψ_2 için $\Delta x = 0.05$ ile elde edilen $E = 0.295$ ’e düşürülmüştür. Eğer sadece ψ_0 hiperküpünün $\Delta x = 0.05$ örnekleme adımı ile örneklenmesi durumunda ise toplam $N_T = (10/0.05)^3 = 8.10^6$ nokta için hesaplama yapılması gerekmektedir. Daralan hiperküp yöntemi ile ihtiyaç duyulan örnekleme sayısı $8.10^6 / 24.10^3 \cong 333$ kat

azaltılabilmektedir. PID tasarım hiperküpüde, en iyi PID katsayılarını arama probleminde, daralan hiperküp arama yöntemi ile işlem maliyetinde kayda değer oranda azalma sağlanmıştır.

5. SONUÇLAR (CONCLUSIONS)

Bu çalışmada, PID tasarım hiperküpü içinde çoklu ölçüte dayanan birim basamak performans analizleri çalışılmış ve en iyi PID’nin belirlenmesine dönük yaklaşımlar sunulmuştur. Performans haritalaması, kontrolör tasarımında ve analizinde önemli veriler sağlamaktadır. Optimal PID katsayıları, örnekleme adımı genişliğine bağlı bir duyarlılıkta kesin olarak elde edilebilmektedir. Yöntemin işlem maliyeti analizleri sunulmuştur. Çoğu optimizasyon tekniği, daha az işlem maliyeti karşılığında optimal PID katsayılarını verebiliyor olsa da, bu katsayıların optimizasyon araştırma bölgesi içinde en optimal değer (global extrimum) olmasını garanti edememektedir. Günümüzde gelişen teknoloji, yüksek işlem yüklerini makul sürelerde tamamlayabilmeye imkan sağlamaktadır. Bu nedenle, optimal çözümlerin kesinliği ve güvenilirliği, işlem yükünden daha öncelikli bir durum haline gelmiştir. Bu durum, bilgisayar destekli kontrolör tasarımı için bu çalışmanın önemini artırmaktadır.

Önerilen performans haritalama metodu, sistem karmaşıklığından bağımsız olarak her kontrol sistemine uygulanabilmektedir. Bu avantaj, her yeni kontrol sistemi için analitik çözümlere ihtiyacı ortadan kaldırmış ve kontrolör tasarımını önemli ölçüde kolaylaştırmıştır.

PID tasarım hiperküpünde en iyi PID katsayısını arama problemini makul işlem maliyetlerinde gerçekleştirebilmek için daralan hiperküp arama yöntemi önerilmiş ve yöntemin etkinliği gösterilmiştir.

KAYNAKLAR (REFERENCES)

- [1] X.-M. Liang, S.-C. Li ve A. Hassan, «A novel PID controller tuning method based on optimization technique,» *Journal of Central South University of Technology*, cilt 17, no. 5, pp. 1036-1042.
- [2] K. Aström ve T. Höggglund, «PID Controllers: Theory, Design and Tuning,» %1 içinde *Instrument Society of America*, New York, 1995.
- [3] H. Zhang, Y. Cai ve Y. Chen, «Parameter Optimization of PID Controllers Based on Genetic Algorithm,» *International Conference on E-Health Networking, Digital Ecosystems and Technologies*, Shenzhen, China, 2010.
- [4] Y. Zhi ve W. Jingling, «Auto-tuning of PID parameters based on switch step response,» *IEEE*

- International Conference on Intelligent Processing Systems*, Beijing, China, 1997.
- [5] B. Ayaz, *PID kontrolörlerinin optimal parametrelerinin belirlenmesi amacıyla bir bulanık mantık karar mekanizması tasarımı*, İTÜ Fen Bilimleri Enstitüsü, 2011.
- [6] Z. Gaing, «A Particle Swarm Optimization Approach For Optimum Design of PID Controller in AVR system,» *IEEE Trans. on Energy Conversion*, cilt 19, no. 2, pp. 384-391, 2004.
- [7] M. Rani, H. Selamat, H. Zamzuni ve Z. Ibrahim, «Multi-Objective Optimization For PID Controller Tuning Using The Global Ranking Genetic Algorithm,» *International Journal of Innovative Computing, Information and Control*, cilt 8, pp. 269-284, 2012.
- [8] M. Tokuda ve T. Zamamoto, «Self-Tuning PID Controller Based on Control Performance Evaluations,» *International Journal of Innovative Computing, Information and Control*, cilt 6, pp. 3751-3762, 2010.
- [9] K. Ogata, *Modern control engineering*, Prentice Hall PTR, 2002.
- [10] A. Visioli, *Practical PID control*, London, 2006.