

# YAPAY SİNİR AĞLARI İLE KESİR DERECELİ PID DENETLEYİCİ KATSAYILARININ AYARLANMASI

Hüseyin ARPACI \*

İnönü Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Bölümü, Malatya, Türkiye

## ÖZET

Bu makalede, Yapay Sinir Ağları (YSA) kullanarak uyarlamalı kesir dereceli PID (FOPID) kontrolörün katsayılarının ayarlanmasına yönelik bir çalışma sunulmaktadır. Kontrolör katsayıların uyarlamalı olarak ayarlanması pratik kontrol uygulamalarında dayanıklılık için çok önemlidir. Çünkü parametre belirsizlikleri ve dış etkenler nedeniyle sistemin katsayılarının değişmesi kontrolörün dayanıklılığını olumsuz yönde etkileyebilir. Bu çalışmada YSA ile kontrolörün oransal, integral ve türev kazanç katsayıları kontrol süreci devam ederken geri beslemeler ile duruma göre yeniden ayarlanabilmektedir. Bu da kesir dereceli PID kontrolörüne hem uyarlamalı hemde dinamik bir yapı kazandırmaktadır. YSA' lar eğitildikten sonra kontrolörün kazanç katsayılarını en uygun şekilde ayarlarlar. Kontrol işlemi sırasında kontrol edilen sistemin parametrelerinde bozulma gerçekleşirse, bu bozulmaya karşı YSA' lar kontrolör katsayılarını yeniden en uygun değere getirerek uyarlama yaparlar. Önerilen yöntem pratik kontrol uygulamaları için FOPID kontrolörün uygulanabilirliğini kolaylaştırır. İki benzetim örneğinde önerilen uyarlamalı kontrol yönteminin performansını göstermek için MATLAB/Simulink kullanılarak YSA içeren FOPID kontrolörün tasarımı gerçekleştirilmiş ve birim basamak cevapları sunulmuştur. Bu cevaplar da (Şekil 14, 16, 19, 21, 23) göstermiştir ki önerilen yöntem hız ve dayanıklılık bakımından daha başarılı olmuştur.

*Anahtar Kelimeler:* Kesir dereceli PID kontrolör, Yapay Sinir Ağları, uyarlamalı kontrol, dayanıklılık

## ADJUSTMENT OF FRACTIONAL ORDER PID (FOPID) CONTROLLER COEFFICIENTS WITH ARTIFICIAL NEURAL NETWORKS

### ABSTRACT

In this article, we present a study for adjusting the coefficients of the adaptive fractional PID controller using Artificial Neural Networks (ANN). Adaptive adjustment of the controller coefficients is very important for the robustness in practical control applications. Because of the parameter uncertainties and external factors, changing the system's coefficients can affect the durability of the controller negatively. In this study, proportional, integral and derivative gain coefficients of the controller can be readjusted by the feedbacks with Artificial Neural Networks while the control process is continuing. This gives the fractional PID controller an adaptive and dynamic structure. If deterioration occurs in the parameters of the controlled system during the control process, the ANN adapt the controller coefficients automatically by optimizing their values against this deterioration. To show the control performance and robustness of the proposed adaptive control method, first the FOPID controller including ANN designed by using MATLAB / Simulink and then the unit step responses for two simulation examples are presented. These responses (Figures 14, 16, 19, 21, 23) have shown that the proposed method has been more successful in terms of speed and durability.

*Keywords:* Fractional order PID controller, Artificial Neural Networks, Adaptive control, durability

\* e-posta: [huseyin.arpaci@inonu.edu.tr](mailto:huseyin.arpaci@inonu.edu.tr)

## 1. Giriş

PID kontrolörler yapısal basitliği ve yeterli derecede performansı nedeniyle kontrol uygulamalarında yaygın olarak tercih edilirler. Bu durum PID kontrol sistemlerini aktif araştırma alanlarından biri konumuna getirmiştir. Ayrıca araştırmacılar PID kontrolörden en iyi performansı almak için kontrol sisteminin kararlılık ve kontrolör katsayılarını ayarlama problemlerine odaklanmışlardır. Ziegler-Nichols (ZN), Cohen-Coon kuralları, Åström-Hägglund yöntemi ve ileri Ziegler-Nichols yöntemi gibi en temel PID kontrolör katsayılarını belirleme ve ayarlama yöntemleri sunulmuştur [1-4]. Gerçek zamanlı kontrol sistemlerinde parametre belirsizlikleri ve dalgalanmaları, öngörülemez sistem bozucu gibi olumsuz durumların etkilerini azaltmak ve gerçek dünya koşullarında dayanıklı kontrol performansı sergilemek için PID kontrolör tasarımları istenmektedir. Bu nedenle PID denetleyici katsayılarının ( $K_p$ ,  $K_i$ ,  $K_d$ ) otomatik veya kendiliğinden ayarlanması için çeşitli yöntemler önerilmiştir. Bu yöntemlerden bazıları; sinir ağı tabanlı algoritmalar [5-7], bulanık mantık teknikleri [8], genetik algoritmalar [9], parçacık sürüşü yöntemleri [10], tabu arama algoritmaları [11], elektromanyetizma benzeri algoritmalar [12], bakterilerin besin arama optimizasyonu algoritması [13], uyarlamalı ayarlama mekanizmasıdır [14].

Son yirmi yılda, kesir dereceli PID (FO-PID, Fractional Order PID) kontrolöre olan ilgi zaman geçtikçe artmıştır. FOPID kontrolörde kazanç katsayılarına ( $K_p$ ,  $K_i$ ,  $K_d$ ) ek olarak iki yeni tasarım parametresi (kesirli integratör- $\lambda$  ve kesirli türev- $\mu$ ) daha eklenmiştir. Bazı çalışmalarda kesir derecelerinin ( $\lambda$ ,  $\mu$ ) kontrolörün performansını arttırdığı gösterilmiştir [15-16]. Bununla birlikte, otomatik ayarlama yöntemleri, FOPID kontrolörün dayanıklılığının iyileştirilmesi için gereklidir. Literatürde, FOPID kontrolörlerinin dayanıklılığını arttırmak ve katsayıların pratik olarak ayarlanması için değişik yöntemler önerilmiştir [17-21]. Kesir dereceli türev ve integral, kontrolörün performansını önemli ölçüde artırmasına karşılık, FOPID kontrolöründe parametre sayısındaki artış geleneksel PID kontrolöre göre katsayıların kendiliğinden ayarlanmasını zorlaştırmaktadır. Parametre sayısındaki bu artış sisteme uygun parametrelerinin belirlenmesinde ve dolayısıyla FOPID kontrolörün kullanımında zorluklar oluşturmaktadır. Bu zorlukları azaltmak için klasik PID' de olduğu gibi FOPID kontrolörün parametrelerini hesaplayan metotlar geliştirilmiştir. S-cevabına dayalı ayarlama kuralları ve kritik kazanca dayalı ayarlama kuralları bu metotlardan bazıları teşkil etmektedir. [17, 22-24].

Sinir ağı kullanarak kendinden ayarlamalı kontrolör tasarımları, güçlü doğrusal olmayan modelleme kabiliyeti ve uyarlanabilirliği nedeniyle çeşitli kontrol probleminde yaygın olarak kullanılmaktadır [6]. Bu çalışmada, sinir ağları kullanılarak sadece kazanç katsayılarının kontrol sürecinde oluşabilecek yeni durumlara göre ayarlanması, yeni duruma kontrolörün katsayılarını uyarlaması sağlanmıştır. Böylece hem uyarlamalı FOPID kontrolör tasarımlarındaki karmaşıklık azaltılmıştır hem de kontrolörün sistemde oluşabilecek değişimlere karşı dayanıklılığı artırılmıştır. YSA' lar kontrol uygulaması için nöronların esnekliğini ve doğrusal olmamasını önemli ölçüde arttıran aktivasyon fonksiyonunun şekilsel olarak çevrimiçi ayarını yaparlar [6-7]. Başlangıçta YSA' lar ZN yöntemine göre belirlenen FOPID kontrolörün katsayıları kullanılarak eğitilirler, eğitilmiş YSA' lar kontrol sırasında aldıkları geri bildirimleri de kullanarak sistemin kontrolü için en uygun katsayı optimizasyonunu yaparak kontrol işlemini sürdürürler.

Bu çalışmada YSA kullanarak FOPID kontrolörün kazanç katsayıları ( $K_p$ ,  $K_i$ ,  $K_d$ ) ayarlanır. Kontrolörün kazanç katsayılarını ayarlamak için SOM (self organizing map - özörgütlemeli harita ağı) YSA modeli kullanılır [25]. Bu makalede iki farklı tasarım örneği mevcuttur; birinci örnekte FOPI, FOPD ve FOPID kontrolörlerin performansı ve kararlılığı test edilmiştir. İkinci örnekte ise zaman gecikmeli sistem için FOPID kontrolörünün performansı gösterilmektedir. Ayrıca her iki örnekte, YSA kullanarak FOPID kontrolörün birim basamak yanıtları ile ZN ile ayarlanmış sabit katsayılara sahip FOPID kontrolör birim basamak yanıtları karşılaştırılır [17].

Bu makale aşağıdaki gibi organize edilmiştir: İlk bölümde FOPID kontrolör ve YSA yapısı kısaca anlatılır. Sonraki bölümde YSA ile FOPID kontrolör tasarım yöntemi gösterilir. En son bölümde ise örnek tasarımlar ve benzetim sonuçları sunulur.

## 2. Materyal ve Metod

### 2.1. Kesir Dereceli PID Kontrolör

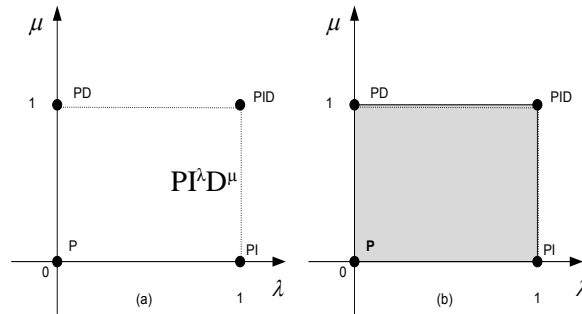
Kesirli cebir işlemlerinde kesirli türev ve  $D^\alpha$  integral olarak gösterilir.  $\alpha > 0$  için kesirli türev gerçekleşir.  $D^\alpha f(t) = \partial^\alpha f / \partial t^\alpha$  ve  $\alpha < 0$  için  $D^\alpha f(t)$  operatörü kesirli integrali ifade eder. FOPID denetleyicisi yaygın olarak şu şekilde ifade edilir,

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (1)$$

Kesirli türevin Laplace  $L(D^\alpha f(t)) = s^\alpha F(s)$  dönüşümü ise, FOPID kontrolör transfer fonksiyonu genel olarak Denklem 2' de ki gibi tanımlanır,

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (2)$$

Burada  $\lambda$  integral teriminin ve  $\mu$  türev teriminin kesirli derecesidir. Kazanç katsayıları  $K_p$ ,  $K_i$  ve  $K_d$  sırasıyla oransal, integral ve türev kazanç katsayılarıdır. [15]'de Podlubny, FOPID kontrolörün hem tamsayı dereceli ve hem de kesir dereceli sistemlerin kontrolünde etkili bir şekilde kullanılabileceğini belirtmiştir. Petras, FOPID kontrolörün, kontrol sistemleri için daha iyi kontrol performansı ve kararlılık sağladığını ifade etmiştir [26]. Aslında,  $\lambda = 1$  ve  $\mu = 1$  tamsayı dereceli durumu, yani klasik PID kontrolörü ifade eder. Şekil 1'de,  $\lambda - \mu$  düzleminde PID kontrolör yapıları gösterilmektedir. Tamsayı dereceli kontrolörler Şekil 1.a' da 4 köşedeki noktalar ile temsil edilmesine karşın, kesir dereceli denetleyiciler ise Şekil 1.b.2' deki gibi taralı alandaki sonsuz sayıda nokta ile temsil edilirler.



**Şekil 1.a.**  $\lambda - \mu$  düzleminde tamsayı dereceli, **(b)**  $\lambda - \mu$  düzleminde kesir dereceli PID kontrolör yapıları

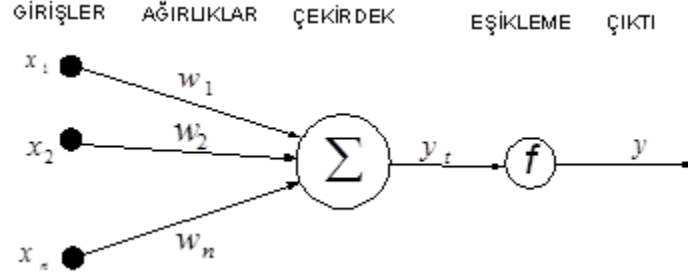
Kontrol sisteminden beklenen performans özelliklerine ulaşılabilmesi için kontrolör parametrelerinin iyi ayarlanması (tuning) gerekir. Bu işleme kontrolör parametrelerinin elde edilmesi denir. Kontrolör parametreleri için uygun değerlerin bulunması kontrolör tasarım çalışmalarının en önemli konularından biridir. Dolayısıyla bu konu üzerinde oldukça fazla durulmuştur [16-24].

Bir tam sayı dereceli PID kontrolör tasarlamak için gerekli parametre sayısı 3 tanedir ( $K_p$ ,  $K_i$ ,  $K_d$ ). Fakat bir FOPID kontrolör tasarlamak için gerekli parametre sayısı 5'i bulmaktadır ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ). Bu durum her ne kadar FOPID kontrolöre daha uygun denetleyici cevapları vermesine neden olsa da, tasarım parametrelerinin artması FOPID kontrolörün katsayı ayar sorununun karmaşık ve zor duruma

gelmesine neden olur. Kesir dereceli  $PI^{\lambda}D^{\mu}$  kontrol uygulamaları sadece kesirli integral veya türev uygulamaları olabileceği gibi kesirli  $PI^{\lambda}$  veya  $PD^{\mu}$  de olabilir. Ya da kesirli  $PI^{\lambda}D^{\mu}$  veya  $PID^{\mu}$  şeklinde de uygulanabilmektedir.

## 2.2. Yapay Sinir Ağı (YSA) ve Özörgütlemeli Harita Yapay Sinir Ağ Modeli (Self-Organizing Map - SOM)

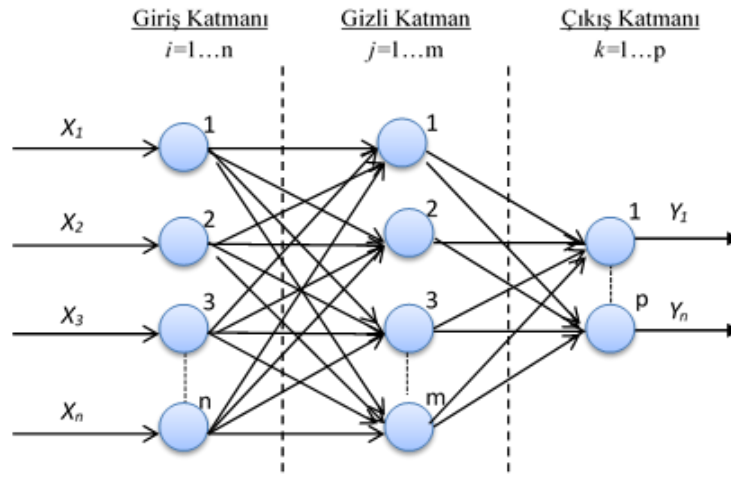
Biyolojik hücrenin çalışma ilkeleri referans alınarak Şekil 2' deki yapay sinir hücresi oluşturulmuştur [27].



Şekil 2. Yapay sinir hücre yapısı

Şekil 2' de ki yapay sinir hücresinin girişleri  $x_n$  ile gösterilmiştir ve her bir girişin ağırlık katsayısı (bu katsayı giriş ile hücre arasındaki bağlantının kuvvetini gösterir)  $w_n$  ile gösterilmiştir.  $x_n$  girdi sinyallerini gösterirken,  $w_n$  ise sinyallerin hücre ile olan bağlantı ağırlık katsayılarının değerlerini taşır. Girdi sinyalleri ağırlıklarla çarpıldıktan sonra çekirdekte toplanarak tüm girdi sinyallerinin ağırlıklı toplamları elde edilir.  $y_t$  çekirdekte elde edilen toplam sinyal göstermektedir ve bu toplam sinyal eşikleme (f) fonksiyonuna girdi olarak yönlendirilir. Sinapsis üzerindeki eşikleme fonksiyonundan çıkan sonuç sinyali  $y$  ile belirtilmiş ve diğer hücreye yönlendirilmiştir [28].

Yapay sinir hücresinin görevi;  $x_n$  girişlerine karşılık  $y$  çıkış sinyalini bulmak ve bu sinyali de diğer hücrelere iletmektir.  $x_n$  giriş sinyali ile  $y$  çıkış sinyali arasındaki ilişkiyi temsil eden  $w_n$  ağırlıkları, her yeni giriş ve çıkış sinyaline göre tekrar ayarlanır. Bu ayarlama sürecine yapay sinir hücresinin öğrenme süreci denir. Öğrenmenin tamamlanabilmesi için  $w_n$  ağırlıklarındaki değişim kararlı olana dek yeni girdilerle sistem beslenir. Kararlılık sağlandığı zaman hücre öğrenmesini tamamlamıştır.



Şekil 3. Çok katmanlı yapay sinir ağı

Görevi yukarıdaki gibi tanımlanan yapay sinir hücrelerinin birleşmesi ile oluşan katmanlı yapının tümü yapay sinir ağı olarak nitelendirilir. Böylece  $m$  adet yapay sinir hücresinin katmanlı bir yapıda bir araya gelmesi ile yapay sinir ağı oluşur.

Yapay sinir hücresindeki öğrenmenin ilk basamağı aktivasyon olarak adlandırılır. Sinir hücresine giren sinyallerin toplamına bakılır. Sinyallerin toplamı o hücreyi aktif hale getirebilecek bir değere sahip midir, değil midir? Eğer toplam sinyal eşik değerini geçebilecek kadar yüksek ise o hücre aktiftir ( $y=1$ ) değilse hücre pasiftir ( $y=0$ ). Bu sorunun cevabı yardımıyla yapay sinir hücresinin sınıflandırma yapabildiği sonucuna ulaşılır. Çok basit anlamda; girdilere 1 ya da 0 cevabını vererek sınıflandırma yapabilen bir hücre, hangi girdilere 1 hangilerine 0 diyeceği hakkında karar vermiş sayılmaktadır. “Karar vermek” ve “sınıflandırmak”, öğrenme sürecinin temel yapı taşlarını oluşturmaktadır [25, 27].

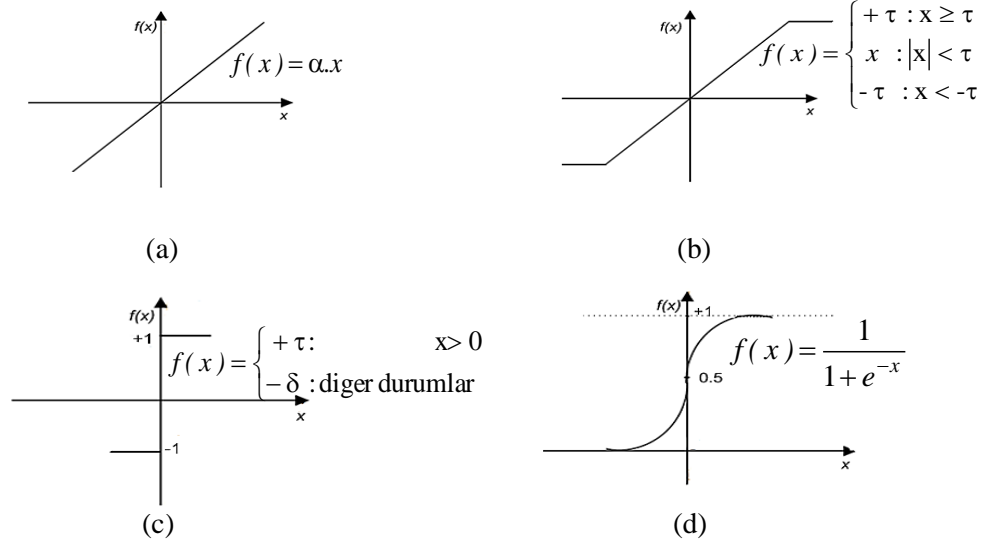
Bir yapay sinir hücresi neye göre girdilere 0 ya da 1 demektir? Sinir hücresi bunu Şekil 2’ de gösterilen ağırlıkları ( $w_n$ ) ile gerçekleştirir. Dışarıdan alınan her bir girdi sinyali ile her defasında ayarlanan bu ağırlıklar öğrenmenin gerçekleştiğinin de göstergesidirler. [27, 29]. Yapay sinir hücresine giren sinyallerin toplamı Denklem 3 ile hesaplanır.

$$y_i = \sum_{i=1}^p x_i * w_i \quad (3)$$

$x_i$  sinyali kendine ait olan ağırlık katsayısı ile çarpılarak toplam sinyale eklenmektedir. Ve  $y_i$  şeklinde toplanan toplam sinyal değeri eşikleme fonksiyonundan geçirdikten sonra çıktı değerini vermektedir.

$$y = f(y_i) \quad (4)$$

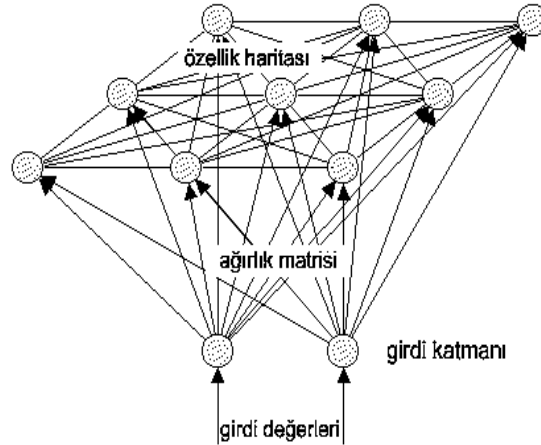
Denklem 4’ te  $f$  olarak gösterilen fonksiyon matematiksel herhangi bir fonksiyon olabilir. Fakat yapay sinir ağı modellerinde 4 tip fonksiyonun daha çok kullanıldığı görülür. Bunlar; doğrusal (a), rampa (b), basamak (c) ve sigmoid (d) fonksiyonlarıdır, Şekil 4 [27, 30].



Şekil 4. (a) doğrusal (b) rampa (c) basamak (d) sigmoid fonksiyonları

SOM ağları, Teuvo Kohonen tarafından geliştirilen eğitimsiz çalışabilen, rekabete dayalı öğrenme (competitive learning) ile giriş verisinin yapısına göre kendini organize edebilen yapılardır. Bu ağlar sınıflandırma yapmak için kullanılmaktadırlar. Bu ağlar olayları öğrenmek için bir öğretmene (eğitici)

ihtiyaç duymazlar. İleri/geri besleme türünde olabilir ve öğrenme algoritması olarak öz-örgütlenme yöntemini kullanır. Bir giriş katmanı ve bir çıkış katmanı olmak üzere iki katmandan oluşur. Öz-örgütlenme kullanan Kohonen Özellik Haritaları insan beynini taklit edecek şekilde tasarlanmıştır. İnsan beyninde öğrenme işlemi, sürekli tekrarlanan olaylar ve durumlar karşısında beyine iletilen sinyallerin, korteksin belli bölgelerinde yoğunlaşması sonucu bir hafıza oluşması şeklinde gerçekleşir [29]. SOM ağlarında yarışmayı kazanma 1 değeri ile gösterilirken, kaybetme durumu 0 değeri ile gösterilir. Ağa bir giriş verildiğinde çıkış uzayında yarışmayı kazanan ve onun etrafındaki komşuları eğitim sırasında ağırlıklarını değiştirmektedirler.



Şekil 5. Kohonen' in 2 katmanlı (girdi, çıktı) SOM ileri beslemeli ağ yapısı

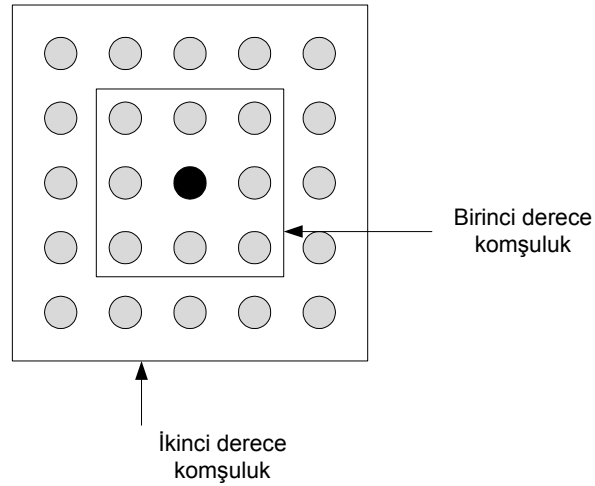
Khonen' in sunduğu ileri/geri beslemeli olan bu model [25] beynin çalışma stiline “ilişkisel bellek” karakteristiğinde olduğu teorisine dayanır. Khonen bu teorisini iki tip ilişkisel bellek ile pekiştirir. Bunlar otomatik ilişkili (auto associative), zıt ilişkili (hetero associative) bellekler. İlişkisel belleklerde öğrenme iki nesne birbiriyle ilişkilendirilerek gerçekleşir.

Bu iki tip ilişkisel bellek modeli Khonen' in modeline dayanak olan modellerdir ve bu modeller iki nesne arasındaki bağı kuvvetliliğinden yararlanırlar. Öğrenme algoritmaları ise “öğretmenli öğrenme” dir. Bu ilişkisel öğrenme modeline örnek vermek gerekirse; elma meyvesi ile kokusunun ilişkisi, limonla ekşimsi tad arasındaki ilişki. İlişkisel belleklerde öğrenme gerçekleştikten sonra örnekte verilen limon girdisine karşılık ekşilik sonucunu verecektir. Ve böylece yapı bu iki ayrı nesneyi ilişkilendirilerek (meyve - tad) öğrenmiş olacaktır.

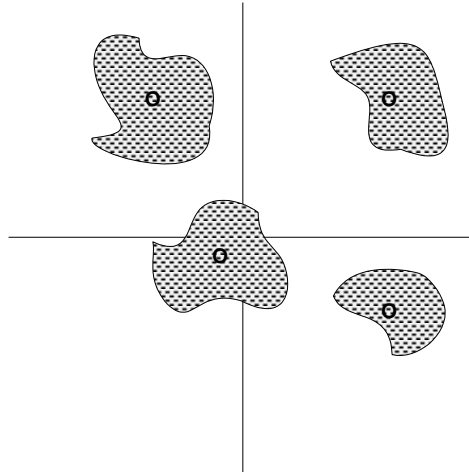
Khonen, modelinde çıkış noktası olarak bu temeli almakla birlikte “self” (kendi kendine) öğrenmeyi algoritması için temel olarak almıştır. Khonen kurduğu yapay sinir ağı modellerinde bir başka kavramdan daha söz ettirmiştir. O da “komşuluk” kavramıdır, Şekil 6. SOM yapısında “küme” (cluster) adı verilen grup merkezleri vardır. Eğer popülasyonda iyi temsilciler seçilebilirse, o popülasyon seçilmiş olan temsilcilerle ifade edilebilirdi. Ya da örneğin “x” adet elemandan oluşan girdiler “n” tane değişik sınıfa ayrılabilirdi. “n” değişik sınıf ise grup (cluster) merkezleri tarafından temsil edilebilirdi, Şekil 7 [29].

Khonen bu mimariyi geliştirirken kendi kendine öğrenmeyi hedeflediği için ele aldığı “x” adetlik girdilerin ilk “n” adedini “n” grup olarak yerleştirir. Sinir ağı tüm girdi elemanlarını bu merkez noktalarına göre öğrenir. Öğrenme sırasında alınan girdiler tüm gruplarla kıyaslanır ve girdi yakın olduğu gruba adapte edilir. Sonuçta her nokta yakın bulunduğu grup merkezini kendine doğru çekmeye çalışarak grup merkezlerinin daha iyi grup temsilcisi olabilmesini sağlamaya çalışırlar. Eğer bu işlem 100, 1000, 10000 döngü ile tekrarlanırsa grup merkezlerinin popülasyonu en iyi temsil eden grup merkezleri olduğu görülür [29]. Her grup merkezi örneklerin ayrı ayrı sınıflarını temsil ederler, Şekil 7.

Bu sayede girdiler kendilerini ait oldukları sınıfın merkezine daha yakın bulacaklardır. “Komşuluk” kavramı sayesinde birbirine yakın olan sınıflar yine birbirleriyle ilişkili hareket edeceklerdir



Şekil 6. SOM ağ modelinde çıktı katmanındaki komşuluk dereceleri

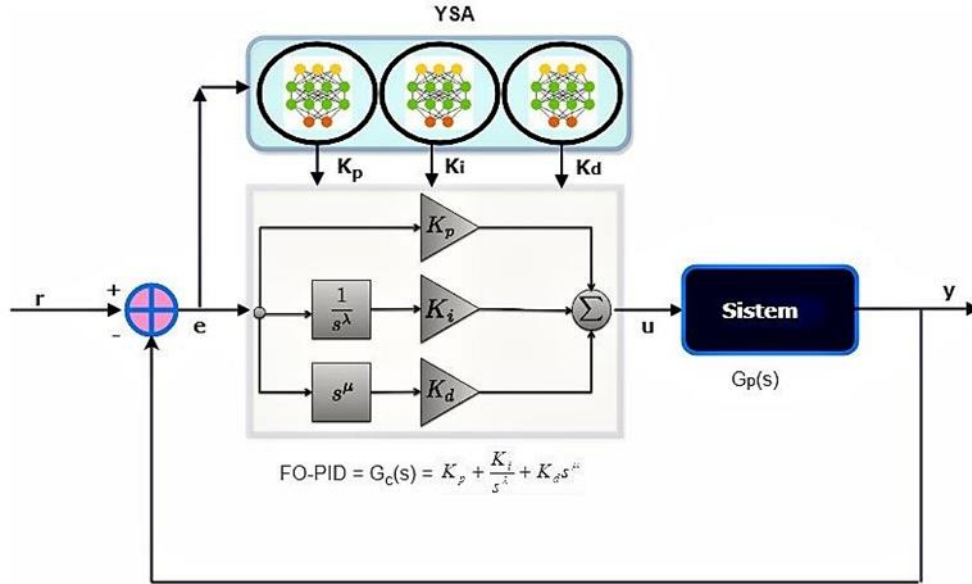


Şekil 7. SOM ağ modeli ile girdilerin sınıflandırılması

### 3. Uyarlamalı Yapay Sinir Ağları Tabanlı Kesir Dereceli PID Kontrolör

Bu çalışma, YSA ile uyarlamalı kesir dereceli PID denetleyicisini geliştirmeyi hedeflemektedir. Önerilen kontrolör sisteminin blok diyagramı Şekil 8’ de gösterilmiştir. YSA çıkışları FOPID bloğunun kazanç katsayılarını besler.

Şekil 8' de optimizasyonda kullanılan kapalı kontrol sisteminin hata sinyali  $e = r - y$  şeklinde tanımlanır. Döngüsel olarak hatanın karesinin toplamı yerel minimuma yaklaştığında, kontrol sisteminin çıkışı referans girişine yaklaşır ve bu nedenle nöronların adaptasyonu tamamlanır ve en iyi sistem cevabı elde edilir.



Şekil 8. YSA içeren FOPID kontrol sisteminin blok diyagramı

Şekil 8' de gösterilen tek girişli-tek çıkışlı bir sistemde,  $G_p(s)$  tamsayı veya kesirli dereceli sistemin transfer fonksiyonunu temsil etmektedir.  $G_c(s)$  ise YSA' lar ile katsayıları ayarlanacak olan FOPID kontrolörü göstermektedir. Şekil 8' deki kontrol sistemindeki hedef, sistemi kararlı hale getiren FOPID kontrolörün kazanç katsayılarının  $K_p$ ,  $K_i$  ve  $K_d$  çevrimiçi ayarlanmasıdır. FOPID kontrolörü zaman düzleminde Denklem 5' de görüldüğü gibi üç bileşene ayrılabiliriz.

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) = u_1(t) + u_2(t) + u_3(t) \quad (5)$$

$$u_1(t) = K_p e(t) \quad (6)$$

$$u_2(t) = K_i D^{-\lambda} e(t) \quad (7)$$

$$u_3(t) = K_d D^{\mu} e(t) \quad (8)$$

$K_p$ ,  $K_i$  ve  $K_d$  parametreleri YSA ile ayarlanan FOPID kontrolör yapısı Şekil 9' da ki gibidir.

$PI^{\lambda}D^{\mu}$  kontrolün transfer fonksiyonu Denklem 3 kullanılarak aşağıdaki gibi yazılabilir.

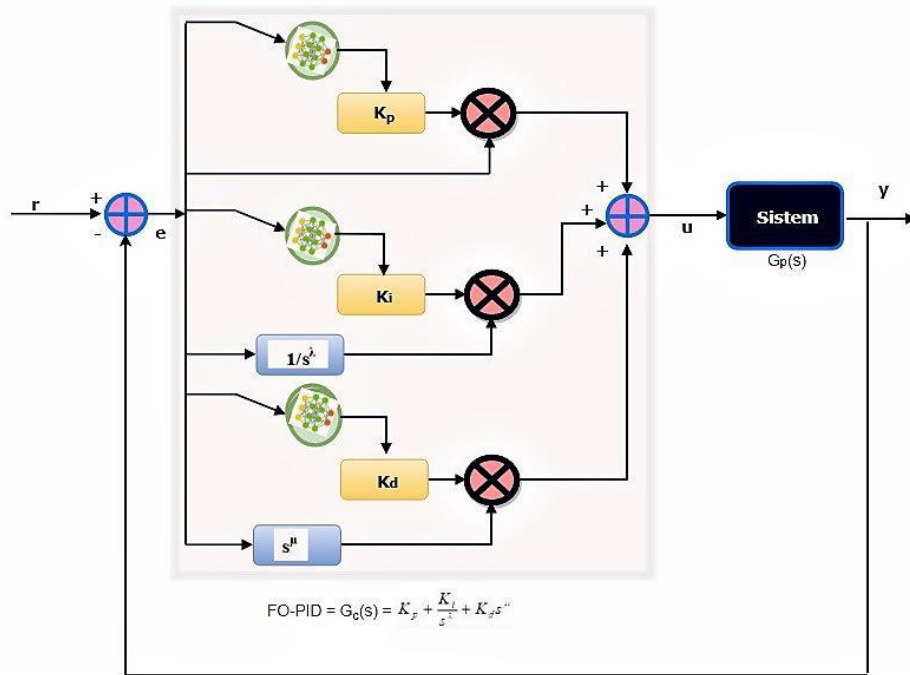
$$u(t) = f_1 \left( \sum_{i=0}^n e_i * w_i \right) * e(t) + f_2 \left( \sum_{j=0}^m e_j * w_j \right) * D^{-\lambda} e(t) + f_3 \left( \sum_{k=0}^v e_k * w_k \right) * D^{\mu} e(t) \quad (9)$$

Kazanç katsayıları aşağıdaki gibidir.

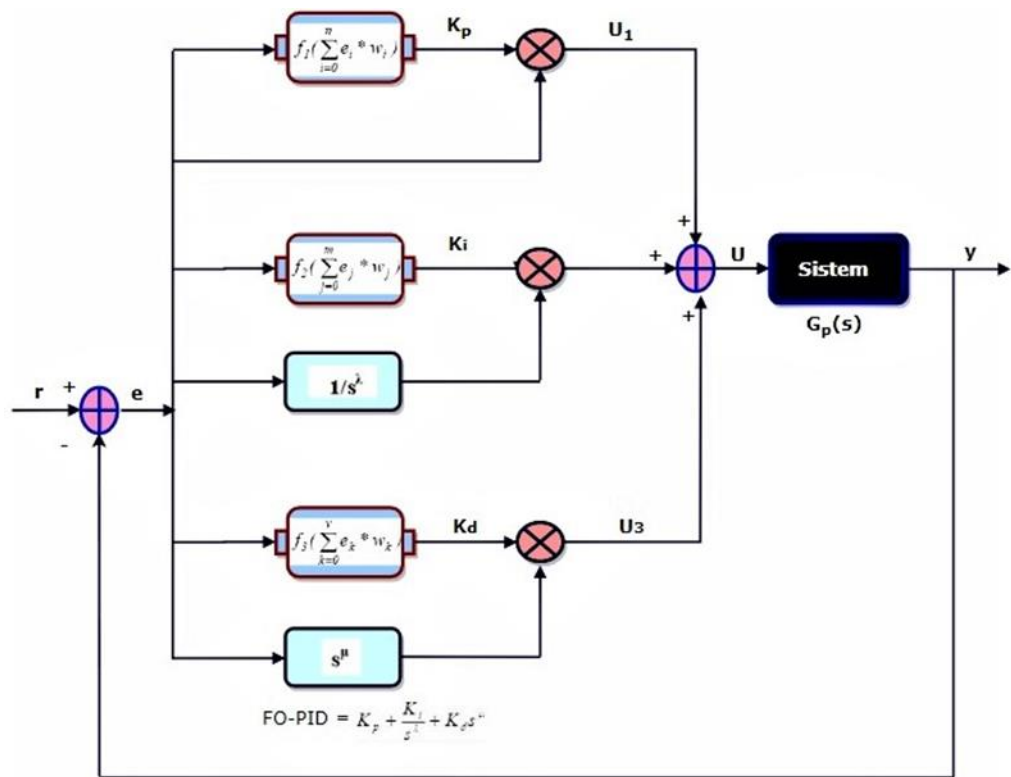
$$K_p(e) = f_1 \left( \sum_{i=0}^n e_i * w_i \right), K_i(e) = f_2 \left( \sum_{j=0}^m e_j * w_j \right), K_d(e) = f_3 \left( \sum_{k=0}^v e_k * w_k \right) \quad (10)$$

Burada sırasıyla  $f_1$ ,  $f_2$  ve  $f_3$  YSA' nın aktivasyon fonksiyonları,  $e_i$ ,  $e_j$  ve  $e_k$  YSA' nın girişleri ve  $w_i$ ,  $w_j$  ve  $w_k$  da YSA' nın ağırlıklarıdır.



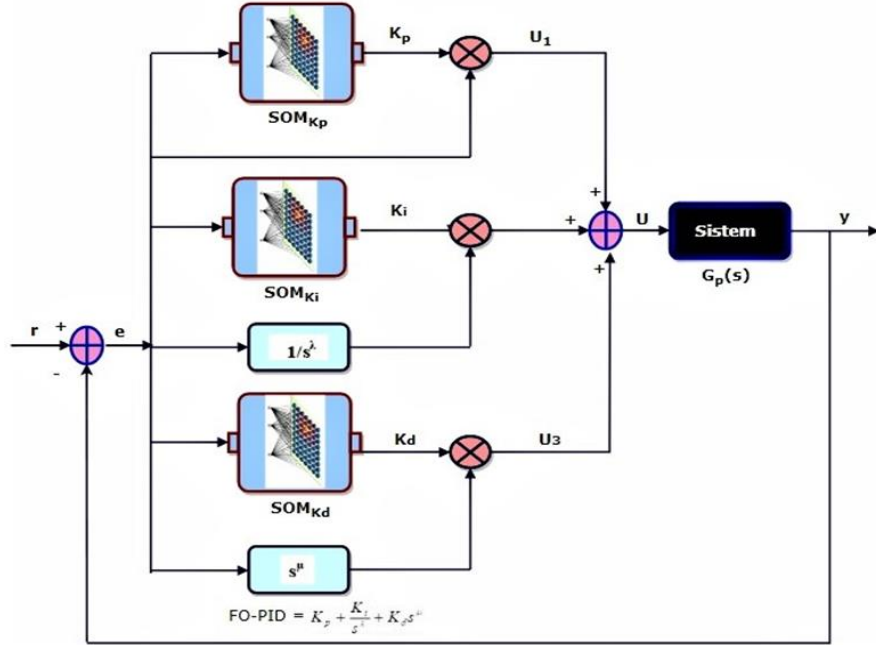


Şekil 9. YSA ile uyarlamalı  $PI^\lambda D^\mu$  kontrolörün blok diyagramı

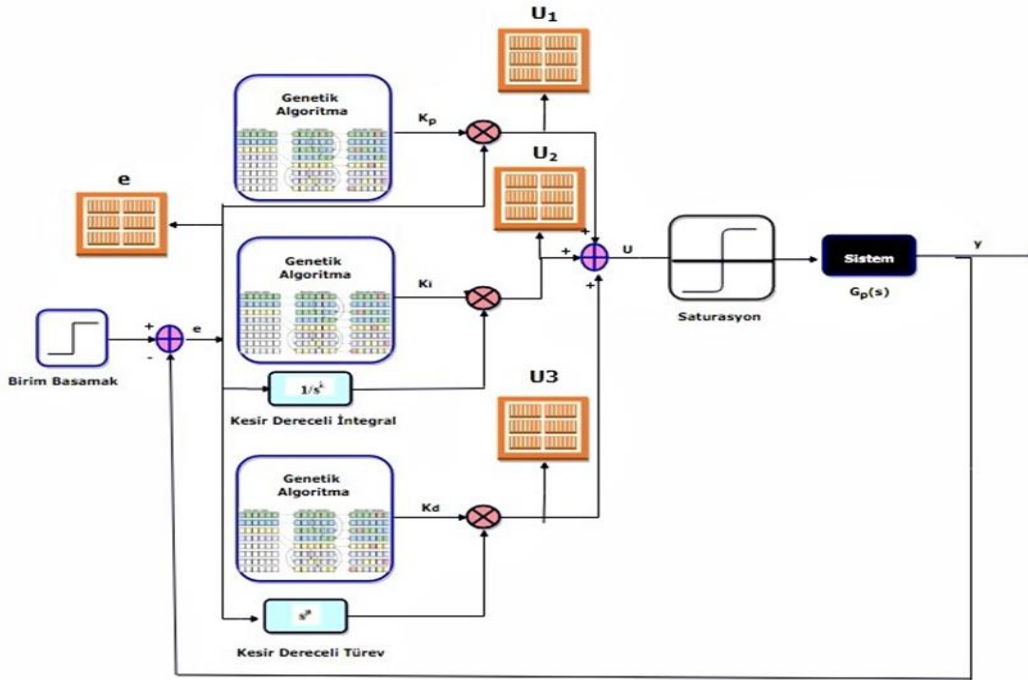


Şekil 10. YSA-FOPI kontrolör tasarımı

Denklem 9 ve 10 ile gösteren sistemin blok diyagramı, Şekil 10 ve 11’ da gösterildiği gibi olur. Burada SOM tipi YSA’ lar kullanılmıştır.  $K_p$ ,  $K_i$  ve  $K_d$  katsayıları SOM-YSA algoritması ile belirlenir. Şekil 8’ deki  $K_p$ ,  $K_i$  ve  $K_d$  yerine Şekil 11’ deki gibi SOM tipi YSA’ lar kullanılmıştır. YSA’ ların eğitimi için gerekli başlangıç verileri Şekil 12’ de verilen kontrolörden elde edilmiştir. Burada kullanılan FOPID kontrolörünün katsayıları ise Genetik Algoritma (GA) ile belirlenmiştir [30, 31]. Bu yöntemle veriler içinde iyi olanın seçilimi sağlanmıştır. Uygun olmayan değerler elenmiştir.



Şekil 11. YSA-FOPID’ nin  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerini ayarlama



Şekil 12. Öğrenme verilerini elde etmek için kullanılan FOPID kontrolörünü içeren sistemin blok diyagramı

YSA' nın eğitimi için gerekli olan başlangıç verileri  $(e, u_1)$ ,  $(e, u_2)$ ,  $(e, u_3)$  olarak Şekil 12' de ki sistemden genetik algoritma ile alınmıştır. Burada  $e$ , YSA' nın giriş öğrenme verisini temsil ederken, oransal, integral ve türev kazanç bloklarının çıkışından elde edilen  $u_1$ ,  $u_2$  ve  $u_3$ , YSA' nın çıkış öğrenme verileridir. Bu çalışmada MATLAB - YSA (SOM) fonksiyonu kullanılarak  $(e, u_1)$ ,  $(e, u_2)$ ,  $(e, u_3)$  eğitim veri çiftleri için öğrenme süreci gerçekleştirilmiştir. Eğitim verilerini elde etmek için gerekli başlangıç değerleri  $(K_p, K_i, K_d, \lambda, \mu)$  genetik algoritma kullanılarak belirlenmiştir. Eğitimi tamamlanan YSA' lar ile sistemin kontrolü sağlanmıştır.

#### 4. Bulgular ve Tartışma

Önerilen yöntemin performansı iki örnekle gösterilmiştir ve elde edilen sonuçlar aşağıdaki gibi analiz edilmiştir.

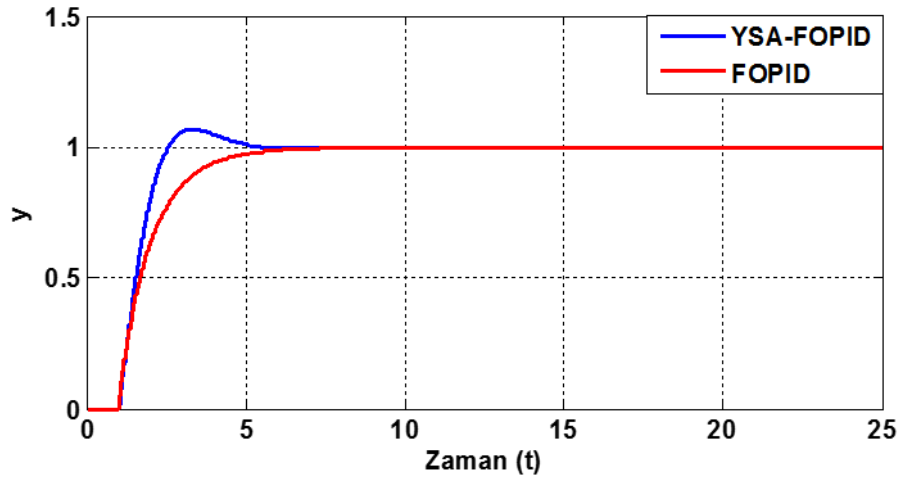
**Örnek 1.** Birinci örnek için sistemin transfer fonksiyonu Denklem 11' deki gibi alınmıştır.

$$G_p(s) = \frac{K}{s(s + \tau)} \quad (11)$$

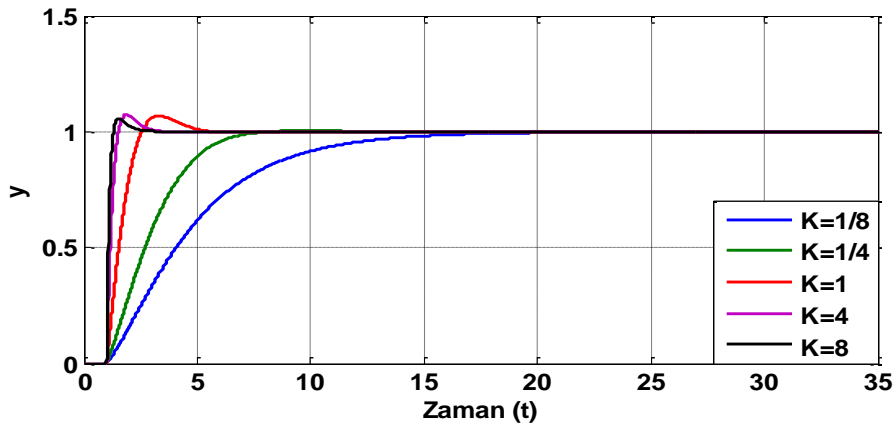
$PI^\lambda D^\mu$  kontrolör için başlangıç parametreleri (katsayıları) genetik algoritma kullanılarak seçilmiştir ve belirlenen bu başlangıç değerleri Denklem 12' deki gibidir.

$$PI^\lambda D^\mu : K_p = 0.9716, K_i = 0.3749, K_d = 0.2964, \lambda = -1.3653, \mu = 1.0017 \quad (12)$$

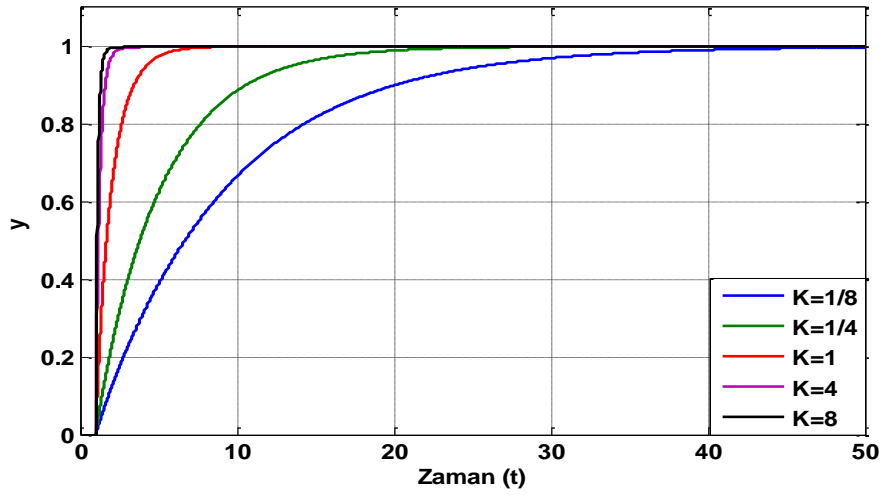
Bu parametreler,  $PI^\lambda D^\mu$  kontrolörünün başlangıçtaki katsayılarıdır ve YSA (SOM)' ları eğitmek için kullanılmıştır. Şekil 11' de gösterilen süreçte  $SOM_{K_p}$ ,  $SOM_{K_i}$  ve  $SOM_{K_d}$  YSA yapıları oluşturulduktan sonra elde edilen başlangıç değerleri referans alınarak eğitilmiştir. Daha sonra farklı  $K$  ve  $\tau$  değerleri için geleneksel  $PI^\lambda D^\mu$  kontrolör ile YSA- FOPID kontrolörün birim basamak cevapları Şekil 13 – 17' de karşılaştırılarak analiz edilmiştir.



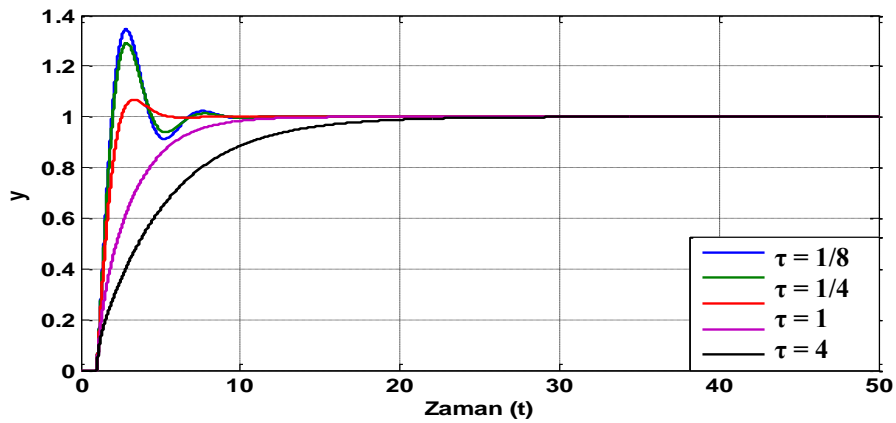
Şekil 13. Aynı  $K$  ( $K = 1$ ) ve  $\tau$  ( $\tau = 1$ ) değerleri için YSA-FOPID ile FOPID kontrolörlerinin birim basamak cevaplarının karşılaştırılması



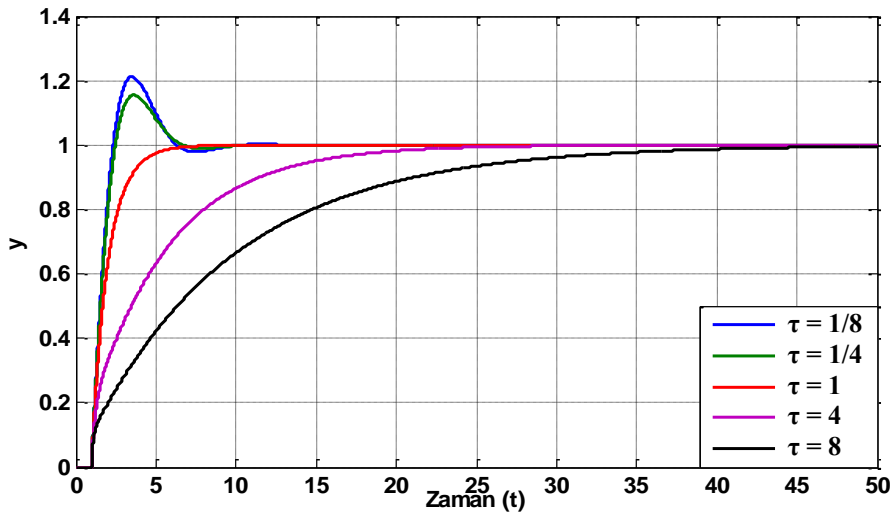
Şekil 14. Farklı  $K$  değerleri için YSA-FOPID kontrolörün birim basamak cevapları



Şekil 15. Farklı  $K$  değerleri için FOPID kontrolörün birim basamak cevapları



Şekil 16. Farklı  $\tau$  değerleri için YSA-FOPID kontrolörün birim basamak cevapları



Şekil 17. Farklı  $\tau$  değerleri için FOPID kontrolörün birim basamak cevapları

Şekil 13' te YSA-FOPID ile FOPID kontrolörlerinin aynı  $K$  ve  $\tau$  değerleri ( $K = 1$ ,  $\tau = 1$ ) için birim basamak cevapları gösterilmiştir. Bu grafikten her iki kontrolörün sistemi kararlı konuma getirmelerinin yaklaşık olarak aynı zamanda gerçekleştiği görülmektedir birlikte YSA-FOPID kontrolörün başlangıçta verdiği tepkini daha hızlı olduğu ifade edilebilir.

Şekil 14 ve Şekil 15' de ise  $\tau$  sabit tutularak ( $\tau = 1$ ) farklı  $K$  değerleri ( $K = 1/8, 1/4, 1, 4, 8$ ) için YSA-FOPID ve FOPID kontrolörlerinin birim basamak cevapları sırası ile verilmiştir. Şekil 14' deki cevaplar incelendiğinde YSA-FOPID kontrolörün çok hızlı ve kararlı cevaplar verdiği görülür. En kötü durumda bile ( $K = 1/8$ ) sistemi yaklaşık 15. saniyede kararlı konuma getirmiştir. Oysa Şekil 15 incelendiğinde ise  $K$  değerlerindeki düşüşe bağlı olarak sistemin kararlı konuma gelme süresinin de uzadığı görülür. Aynı değer için ( $K = 1/8$ ) karşılaştırıldığında sistem yaklaşık olarak 50. saniyede kararlı noktaya ulaşmıştır.

Şekil 16 ve 17' de ise  $K$  değeri sabit tutularak ( $K = 1$ ) farklı  $\tau$  değerleri ( $\tau = 1/8, 1/4, 1, 4, 8$ ) birim basamak cevapları karşılaştırıldı. YSA-FOPID kontrolörün verdiği cevaplara bakıldığında düşük  $\tau$  değerleri için taşmalar meydana gelmekle birlikte büyük  $\tau$  değerleri için yine sistem kısa sürede ( $\tau = 8$  için yaklaşık 20. sn.) kararlı noktaya ulaşmıştır. FOPID kontrolörün cevaplarında ise kararlı konuma gelme süresinin ( $\tau = 8$  için yaklaşık 45. sn) yine yüksek olduğu görülür.

**Örnek 2.** Bu örnekle de önerilen yöntemin zaman gecikmeli sistemlerdeki performansı test edilmiştir. Ham şeker kamışı suyu nötralizasyon işleminin transfer fonksiyonu olan örnek Denklem 13' deki gibidir.

$$G(s) = \frac{k}{\tau s + 1} e^{-Ls} = \frac{0.55}{62s + 1} e^{-1s} \quad (13)$$

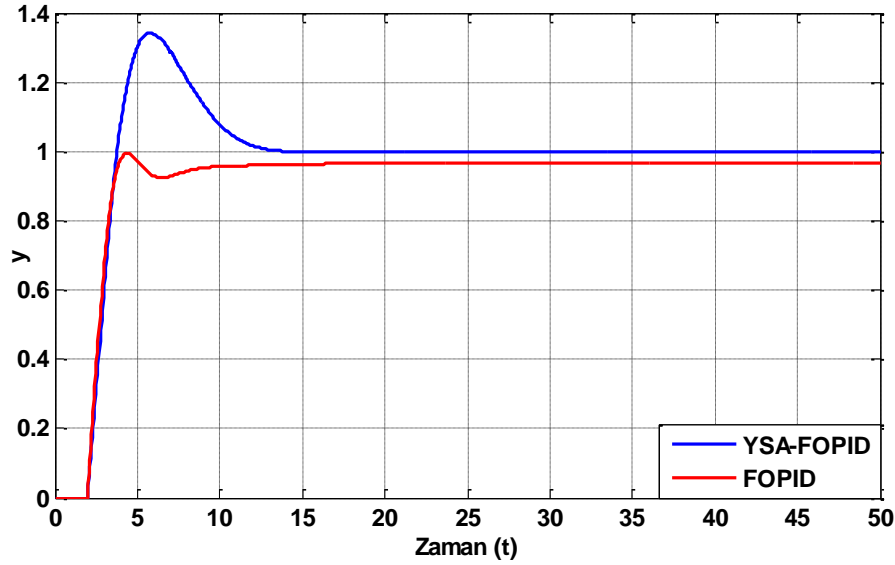
Birinci dereceden Pade yaklaşımı kullanılarak,

$$G(s) = \frac{k}{\tau s + 1} e^{-Ls} = \frac{0.55}{62s + 1} e^{-1s} = \frac{0.55}{62s + 1} \frac{1 - 0.5s}{1 + 0.5s} \quad (14)$$

$$G(s) = \frac{0.55 - 0.275s}{31s^2 + 62.5s + 1} \quad (15)$$

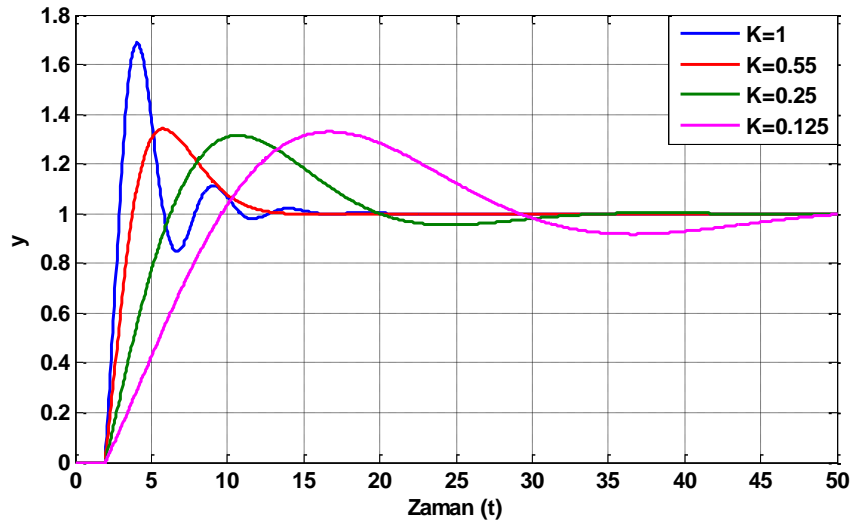
elde edilir.

Bu durumda farklı  $K$  değerleri ( $K = 1/4, 1/8, 0.55, 1$ ), farklı  $L$  değerleri ( $L = 1, 0.8, 0.5, 0.1$ ) ve farklı  $\tau$  değerleri ( $\tau = 31, 62, 93$ ) için FOPID ve YSA-FOPID kontrolörlerinin Şekil 18 - 22' de elde edilen birim basamak cevaplarının karşılaştırmalı analizleri yapıldı.

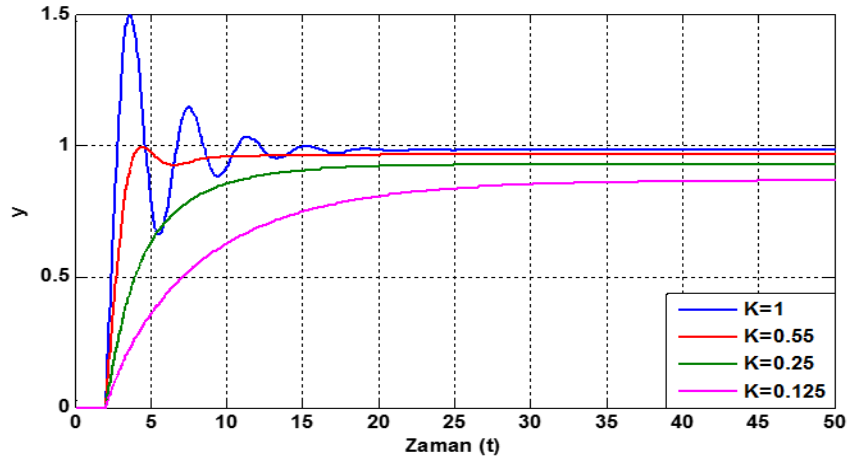


Şekil 18. Aynı  $K$  ( $K = 0.55$ ),  $\tau$  ( $\tau = 62$ ) ve aynı  $L$  ( $L = 1$ ) değerleri için YSA-FOPID ile FOPID kontrolörlerinin birim basamak cevaplarının karşılaştırılması

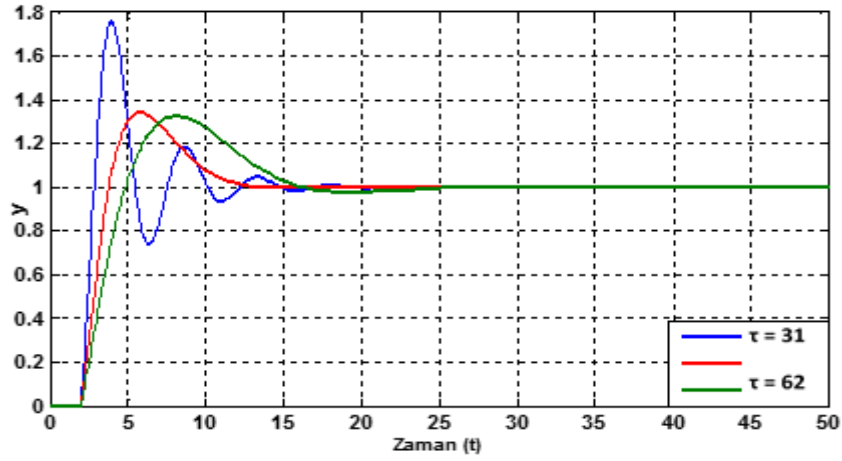
Şekil 18' de YSA-FOPID ile FOPID kontrolörlerinin aynı  $K$  ve  $\tau$  değerleri ( $K = 0.55$ ,  $\tau = 62$ ) için birim basamak cevapları gösterilmiştir. Bu grafikten YSA-FOPID kontrolörün sistemi kısa sürede (yaklaşık 12 sn.) kararlı konuma getirmesine karşın FOPID kontrolörün sistemi kararlı konuma getirme süresinin daha uzun olduğu görülmektedir. Ayrıca FOPID kontrolör ile yapılan kontrol uygulamasında referans değere (1) ulaşamama sorunu görülmektedir.



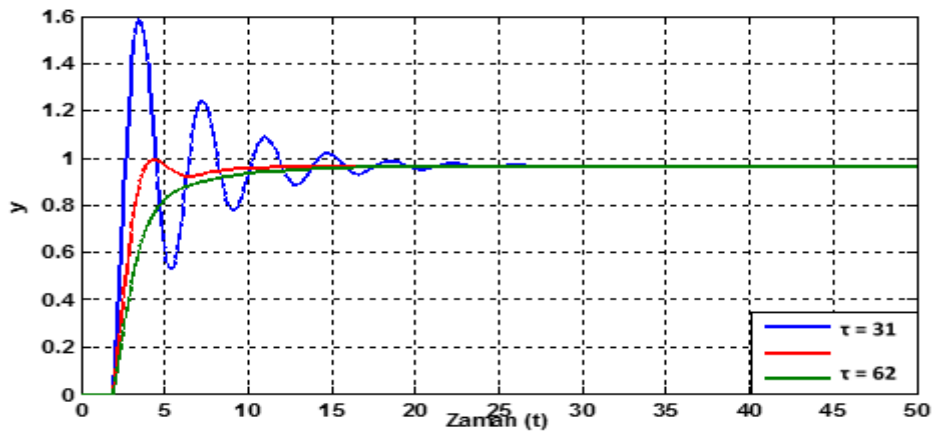
Şekil 19. Farklı  $K$  değerleri için YSA-FOPID kontrolörün birim basamak cevapları



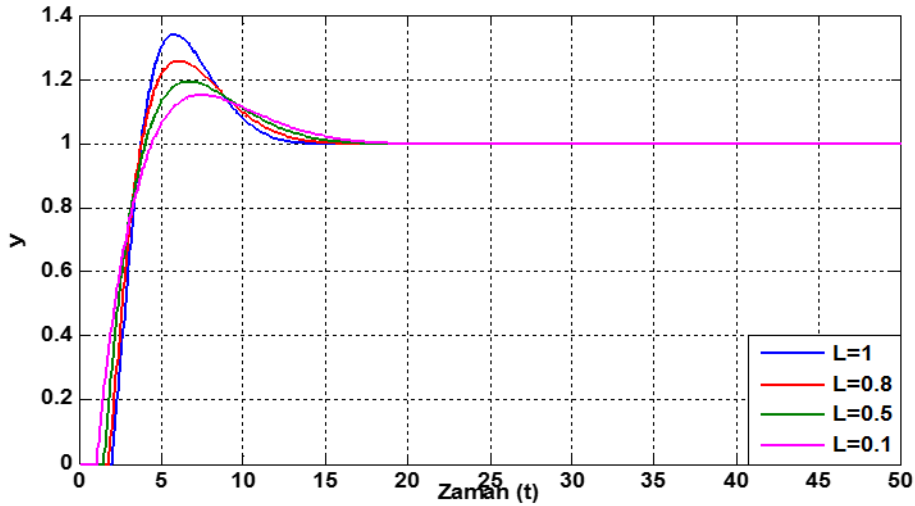
Şekil 20. Farklı K değerleri için FOPID kontrolörün birim basamak cevapları



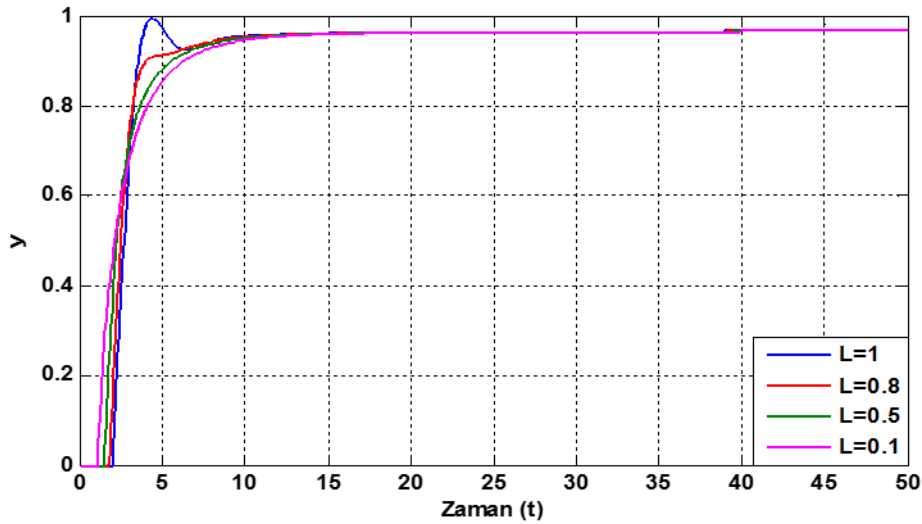
Şekil 21. Farklı  $\tau$  değerleri için YSA-FOPID kontrolörün birim basamak cevapları



Şekil 22. Farklı  $\tau$  değerleri için FOPID kontrolörün birim basamak cevapları



Şekil 23. Farklı L değerleri için YSA-FOPID kontrolörün birim basamak cevapları



Şekil 24. Farklı L değerleri için FOPID kontrolörün birim basamak cevapları

Şekil 19 ve Şekil 20' de  $\tau$  ve L sabit tutularak ( $\tau = 62$ ,  $L = 1$ ) farklı K değerleri ( $K = 1, 0.55, 0.25, 0.125$ ) için YSA-FOPID ve FOPID kontrolörlerinin birim basamak cevapları sırası ile verilmiştir. Şekil 19' da ki cevaplar incelendiğinde YSA-FOPID kontrolörün hızlı ve kararlı cevaplar verdiği görülür. FOPID kontrolörün birim basamak cevaplarının gösterildiği Şekil 20 incelendiğinde ise K değerlerindeki düşüşe bağlı olarak sistemin kararlı konuma gelme süresinin uzadığı ve referans değerden uzaklaştığı görülür.

Şekil 21 ve 22' de ise K ve L değerleri sabit tutularak ( $K = 0.55$ ,  $L = 1$ ) farklı  $\tau$  değerleri ( $\tau = 31, 62, 93$ ) için birim basamak cevapları karşılaştırıldı. YSA-FOPID kontrolörün verdiği cevaplara bakıldığında düşük  $\tau$  değerleri için taşmalar meydana gelmekle birlikte büyük  $\tau$  değerlerin de aşım oranı düşmüştür. FOPID kontrolörün cevaplarında ise kararlı konuma gelme süresinin yüksek olmasının yanı sıra aşım sayılarının fazlalığı ve referans değeri yakalama güçlüğü görülür.

Şekil 23 ve 24' de ise K ve  $\tau$  değerleri sabit tutularak ( $K = 0.55$ ,  $\tau = 62$ ) farklı L değerleri ( $L = 1, 0.8, 0.5, 0.1$ ) için birim basamak cevapları karşılaştırıldı. YSA-FOPID kontrolörün verdiği cevaplara bakıldığında sistemin kontrolü sorunsuzca tamamlanırken (15 sn.) FOPID kontrolörün cevaplarında ise taşım oranı azalmakla birlikte referans noktaya ulaşım sıkıntısı devam etmektedir.



## 5. Sonuçlar

Bu çalışmada Yapay Sinir Ağları (YSA) ile kesir dereceli PID (FOPID) kontrolörün kazanç katsayıları ( $K_p$ ,  $K_d$  ve  $K_i$ ) ayarlanarak sistemin en iyi şekilde kontrol edilmesi çalışılmıştır. FOPID kontrolör ile sistemleri kararlı bir şekilde kontrol edilebilirliği daha önce yapılan çalışmalarda gösterilmiştir [3 - 8]. Fakat sistemin parametrelerinde çevresel nedenler ile değişimler söz konusu olduğunda, FOPID kontrolörün sistemi kararlılıkla kontrol etmekte zorlandığı alınan benzetim sonuçlarında gözlemlenmiştir (Şekil 15, 17, 20, 22, 24). YSA kullanılarak gerçekleştirilen tasarımda (YSA-FOPID) ise duruma göre kazanç katsayılarını ayarlanması söz konusu olduğu için kontrol edilen sistemdeki parametre değişimlerine karşı dayanıklılığın arttığı ve çok daha geniş bir parametre aralığında sistemi kararlılıkla daha hızlı kontrol edebilmektedir (Şekil 14, 16, 19, 21, 23). Ayrıca kesir dereceli PID (FOPID) kontrolörlere göre daha hızlı ve kararlı cevaplar verdiği karşılaştırmalı cevap eğrilerinden gözlemlenmiştir (Şekil 13, 18). İleriye yönelik çalışmalarda yine YSA kullanarak FOPID denetleyicinin integral derecesinin ( $\lambda$ ) ve türev derecesinin ( $\mu$ ) ayarlanması sağlanabilir.

## Kaynaklar

- [1] Ziegler JG, Nichols N B (1942). Optimum Settings for Automatic Controllers, Trans. ASME, 64, 759 – 765.
- [2] Åström KJ, Hägglund T, Hang CC, Ho WK (1993). Automatic tuning and adaptation for PID controllers - a survey, Control Eng. Prac,1; 699-714.
- [3] Åström KJ, Hägglund T (1995). PID controllers: Theory, Design, and Tuning, Instrument, Society of America.
- [4] Zhuang M, Atherton DP (1993). Automatic tuning of optimum PID, IEE Proceedings D – Control Theory and Applications, 140(3), 216-224.
- [5] Mizumoto I, Tanaka H, Iwai Z (2010). Adaptive PID Control for Nonlinear Systems With a Parallel Feed forward Compensator, International Journal of Innovative Computing, Information and Control, 2901-2918.
- [6] Chang WD, Hwang RC, Hsieh JG (1998). Adaptive control of multivariable dynamic systems using independent self-tuning neurons, In Proceeding of the Tenth International Conference on Tools with Artificial Intelligence, 68-73.
- [7] Chen CT, Chang WD (1996) A feed forward neural network with function shape autotuning, Neural Networks, 9(4), 627-641.
- [8] Xu B, Pandian RS, Sakagami N, Petry F (2012). Neuro-fuzzy control of underwater vehicle-manipulator systems, Journal of the Franklin Institute, 349, 1125-1138.
- [9] Saad MS, Jamaluddin M, Darus IZM (2012). Implementation of PID Controller Tuning Using Differential Evolution and Genetic Algorithms, International Journal of Innovative Computing, Information and Control, 9(11), 7761-7779.
- [10] Gaing ZL (2004). A Particle Swarm Optimization Approach For Optimum Design of PID Controller in AVR system, IEEE Trans. on Energy Conversion, 19(2), 384-391.
- [11] Bagis A (2011). Tabu search algorithm based PID controller tuning for desired system specifications, Journal of the Franklin Institute, 348, 2795-2812.
- [12] Lee CH, Chang FK (2010). Fractional-order PID controller optimization via improved electromagnetism-like algorithm, Expert Systems with Applications, 37(12), 8871-8878.
- [13] Ali ES, Abd-Elazim SM (2011). Bacteria foraging optimization algorithm based load frequency controller for interconnected power system, International Journal of Electrical Power & Energy Systems, 33(3), 633-638.
- [14] Cetin M, Iplikci S (2015). A novel auto-tuning PID control mechanism for nonlinear systems, ISA transactions, 58, 292-308.
- [15] Podlubny I (1999). Fractional order systems and  $PI^\lambda D^\mu$  controller, Proc. IEEE Transaction on Automatic Control, 44, 208–214.

- [16] Li HS, Luo Y, Chen YQ (2010). A Fractional Order Proportional and Derivative (FOPD) Motion Controller: Tuning Rule and Experiments, *IEEE Transactions on Control System Technology*, 18, 516-520.
- [17] Valério D, Costa JS (2006). Tuning of fractional PID controllers with Zeigler-Nichols-type rules, *Signal Processing*, 86, 2771–2784.
- [18] Valério D, Costa JS (2007). Tuning-Rules for Fractional PID controllers, *Advances in Fractional Calculus*, 463-476.
- [19] Duma R, Dobra P, Trusca M (2012). Embedded application of fractional order control, *Electronics Letters*, 48(24),1526-1528.
- [20] Padula F, Visioli A (2012). Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes, *Control Theory & Applications*, 6(6), 776-786.
- [21] Arpacı H, Özgüven ÖF, Serhat M (2018). Otomatik Ayarlamalı Sinir Hücresi ile Adaptif Kesir Dereceli PID Kontrolör Tasarımı, *Journal of New Results in Engineering and Natural Science*, No:8, 100-118.
- [22] Caponetto R, Fortuna L, Porto D (2002). Parameter Tuning of a Non Integer Order PID Controller, In: *Electronic proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems*.
- [23] Chen YQ, Moore KL (2002). Discretization schemes for fractional order differentiators and integrators, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 49, No 3, 363–367.
- [24] Valério D, Sá da Costa JMG (2006). Tuning-Rules for Fractional PID controllers, 2nd IFAC workshop on fractional differentiation and its applications 06, Porto.
- [25] Kohonen T (1984). *Self-Organization and Associative Memory*, (3rd edition 1989). Springer, Berlin.
- [26] Petras I (1998). Control Quality Enhancement by Fractional Order Controllers, *Acta Montanistica Slovaca*, Vol. 3, No 2, 143-148.
- [27] Fausett LV (1994). *Fundamentals of Neural Networks*, Prentice Hall.
- [28] Base NK (1995). *Neural Network Fundamentals with Graphs, Algorithms and Applications*, McGraw-Hill Education.
- [29] Cinsdikici M, Yüçetürk AC, Öztürk Y (1997). ATM Networkleri için Yapay Sinir Ağı Çözümleri, 14.Ulusal Bilişim Kurultayı (BİLİŞİM.97) Bildirileri, İstanbul, 55-59.
- [30] Lipmann RP (1987). An introduction to computing with neural networks, *IEEE ASSP Mag.* 4, 4-22.
- [31] Cao JY, Liang J, Cao BG (2005). Optimization of Fractional Order PID Controllers Based on Generic Algorithm, 2005 International Conference on Machine Learning and Cybernetics, Vol.9, 5686-5689, 18-21.
- [32] Zhang Y, Li J (2011). Fractional-order PID Controller Tuning Based on Genetic Algorithm, *Business Management and Electronic Information (BMEI)*, 2011 International Conference on, Vol.3, 764 – 767, 13-15.