



Kahramanmaraş Sütçü İmam University Journal of Engineering Sciences



Geliş Tarihi : 04.12.2019
Kabul Tarihi : 30.12.2019

Received Date : 04.12.2019
Accepted Date : 30.12.2019

ADAPTİF PARÇACIK SÜRÜ OPTİMİZASYONU (APSO) ve KARINCA KOLONİSİ ALGORİTMASI (KKA) KULLANARAK ÖZ AYARLAMALI PID KONTROLÖR TASARIMI

DESIGN OF SELF TUNING PID CONTROLLER USING ADAPTIVE PARTICULAR SWARM OPTIMIZATION (APSO) and ANT COLONY OPTIMIZATION (ACO)

Oğuzhan KAPUKAYA *¹(ORCID:0000-0003-2551-6364)

Mahit GÜNEŞ ¹(ORCID: 0000-0002-1552-3889)

¹ Kahramanmaraş Sütçü İmam Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, Kahramanmaraş, Türkiye

*Sorumlu Yazar / Corresponding Author: Oğuzhan KAPUKAYA, kapukaya.oguzhan@gmail.com

ÖZET

Bu çalışmada, adaptif parçacık sürü optimizasyonu (APSO) ve karınca kolonisi algoritması (KKA) kullanarak öz ayarlamalı oransal integral türevsel (PID) kontrolör tasarımı yapılmıştır ve DA (Doğru Akım) Motoru üzerinde test edilmiştir. PID kontrolör elle ayarlandığı zaman, parametrelerinin ayarlanması zor ve sıkıntılıdır, ayrıca iyi bir performans sağlanamaz. Bu yüzden, parametre ayar işlemi aşamasında optimizasyon yöntemleri kullanmak işi kolaylaştırır ve daha iyi performans sağlar. Bu çalışmada PID kontrolörün parametre ayar işlemi, klasik PID, adaptif parçacık sürü optimizasyonu (APSO) ve karınca kolonisi algoritması (KKA) algoritması ile yapılmış ve sonuçlar karşılaştırılmıştır.

Anahtar Kelimeler: PID, Adaptif Parçacık Sürü Optimizasyonu (APSO), Karınca Kolonisi Algoritması (KKA)

ABSTRACT

In this study, Self-tuning proportional integrator derivative (PID) controller design using the adaptive particle swarm optimization (APSO) and ant colony optimization (ACO) algorithm and tested on DC (Direct Current) Motor. When the PID controller is manually set, setting parameters are difficulty and do not have a good performance. Therefore, using optimization methods in the parameter setting process makes the easier and gives better results. In this study, the parameter setting process of the PID controller was performed and compared with classical PID, adaptive particle swarm optimization (APSO) and ant colony optimization (ACO) algorithm results.

Keywords: PID, Adaptive Particle Swarm Optimization (APSO), Ant Colony Optimization (ACO)

GİRİŞ

DA motorlar birçok ülkede yaygın olarak kullanılmaktadır. Basit ve geniş kontrol özellikleri sayesinde, santrifüj pompalar, fanlar ve robot manipülatörleri gibi birçok uygulamada tercih edilmektedir. DA Motorları kontrol etmek için kullanılan farklı kontrol tipleri vardır. PID Kontrolör, LQR Kontrolör, Bulanık Mantık (Özçalık ve vd., 2014) bunlardan birkaçıdır. Endüstriyel uygulamalarda kolay uygulanabilirliği ve etkili oluşu ile en çok kullanılan PID Kontrolördür (Kılıç ve vd., 2016a). PID kontrol cihazının performansı tamamen Oransal, İntegral ve Türev ayarlama işlemine bağlıdır, Elle ayar ile PID kontrolörlerde iyi performans sağlamak konusunda deneyimli personel gerektirir (Alhasan, 2017). PID kontrol cihazının özelliklerini geliştirmek ve parametrelerini ayarlamak için PSO tabanlı PID (Berber Ö., 2016), Karınca kolonisi algoritmasına dayalı PID, Genetik Algoritma tabanlı PID (Nagarajvd., 2010) ve Yapay Sinir Ağları tabanlı PID (Erdoğan P. vd., 2015; Kilic ve vd., 2016b) gibi birçok yaklaşım öne sürülmüştür. Parçacık sürüsü optimizasyonu popülasyon tabanlı bir arama algoritmasıdır. Kuş ve balık sürülerinin hareketlerinden esinlenerek optimizasyon yöntemi geliştirilmiştir. Karınca kolonisi algoritması ise doğadaki karıncaların yem arama hareketlerinden esinlenerek geliştirilmiştir.

Bu iki çalışmada PID kontrolörün optimizasyonu amacıyla kullanılmaktadır. Bu çalışmada DA motor için matematiksel bir modelleme yapılmış ve klasik PID, Adaptif parçacık sürü optimizasyonu ve Karınca kolonisi algoritması için model üzerinde sistem tepkisi incelenmiş ve performans araştırmaları yapılmıştır.

DA MOTOR MATEMATİKSEL MODELLEMESİ

Bu model de DA Motor dinamikleri idealdir; örneğin, manyetik alanın sabit olduğu varsayılır. Şekil 1 Atalet yükü ile sürülen ayrı uyarılmış DA Motorun şematik diyagramını göstermektedir. DA Motor tarafından üretilen T_m torku, uygulanan voltaj tarafından indüklenen armatür akımı(i) ile orantılıdır.

$$T_m(t) = K_m i(t) \quad (1)$$

Sembol K_m , armatür sabitidir. Geri elektromotor kuvveti V_{emf} , milin açısal hızına orantılı bir gerilimdir.

$$V_{emf}(t) = K_b w(t) \quad (2)$$

Emf sabiti olan K_b , Newton denklemleri tarafından türetilen Motor denkleminin mekaniksel kısmıdır.

$$J \frac{dw}{dt} = -K_f w(t) + K_m i(t) \quad (3)$$

Burada K_f viskoz sürtünme için doğrusal bir yaklaşımdır. Aşağıdaki denklem, motorun elektrik kısmını açıklar.

$$V_{app}(t) - V_{emf}(t) = L \frac{di}{dt} + Ri(t) \quad (4)$$

Yukarıdaki denklemleri basitleştirdikten sonra, Motor dinamiklerini tanımlayan iki diferansiyel denklem elde ettik.

İlki indüklenen armatür akımı için,

$$\frac{di}{dt} = -\frac{R}{L} i(t) - \frac{K_b}{L} w(t) + \frac{1}{L} V_{app}(t) \quad (5)$$

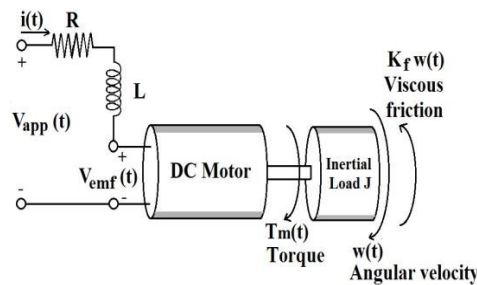
İkinci denklem açısal hız için,

$$\frac{dw}{dt} = -\frac{K_f}{J} w(t) + \frac{K_m}{J} i(t) \quad (6)$$

Yukarıdaki (5) ve (6) numaralı denklemler DA Motor sisteminin dinamiklerini göstermektedir. DA Motorun Durum Uzay sisteminin temsili iki lineer diferansiyel denklemden elde edilir. İndüklenen armatür akımı(i) ve Açısal hız(w) durum değişkenleridir. Uygulanan gerilim V_{app} sistem girişi, Açısal hız w ise sistem çıkışıdır.

$$\frac{d}{dt} \begin{bmatrix} i \\ w \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_m}{J} & -\frac{K_f}{J} \end{bmatrix} \begin{bmatrix} i \\ w \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_{app}(t) \quad (7)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ w \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} V_{app}(t) \quad (8)$$



Şekil 1. Atalet Yükü ile Sürülen Yabancı Uyarılmış DA Motorun Şematik Diyagramı

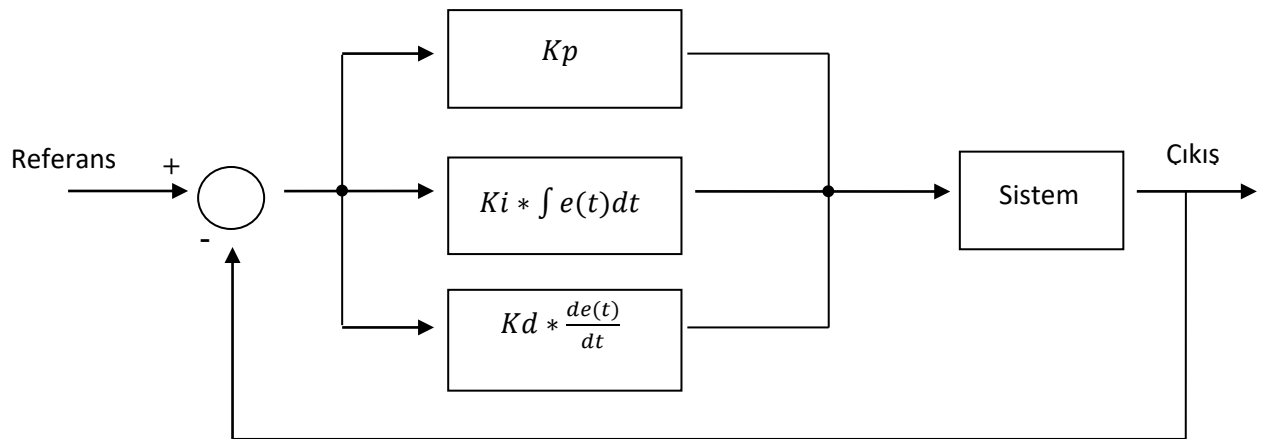
Simülasyonda kullanılan DA Motor parametre ve katsayılar Tablo 1 içerisinde.

Tablo 1. YabancıUyartımlı DA Motor Modelinin Parametreleri

Parametreler	Sembol	Değer	Birim
Armatürdirenci	R	2	Ω
Armatürİndüktansı	L	0.5	Henry
ArmatürSabit	K_m	0.1	-
EMF Sabiti	K_b	0.1	-
SürtünmeKatsayısı	K_f	0.2	N.m.s
EylemsizlikMomenti	J	0.02	$Kg.m^2$

PID KONTROLÖR

PID yaygın olarak kullanılan geri beslemeli bir kontrol yöntemidir. Kontrolör ve sistemin genel yapısı Şekil 2’de verilmiştir. Denetlenecek sistemin çıkışı ile kontrolörün girişi olan referans değer farkı hata olarak hesaplanır. Hata K_p parametresiyle, hatanın türevi K_d parametresiyle, hatanın integrali K_i parametresiyle çarpılarak elde edilen değerlerin toplamı sisteme giriş olarak uygulanır. Sistemin çıkışı tekrar referans değer ile karşılaştırılarak yeni bir hata değeri hesaplanır ve kontrolörün üreteceği yeni çıkış sisteme giriş olarak uygulanır. Bu işlemler hata sıfır olana kadar devam eder. Kontrolörün K_p , K_d ve K_i parametreleri sisteme özeldir ve bu parametreleri belirlemek için birçok yöntem geliştirilmiştir. Kullanılan en klasik yöntem Ziegler-Nichols yöntemidir (Ziegler ve Nichols, 1993).



Şekil 2. PID Kontrolör Kontrol Şeması

Aşağıdaki denklemde ise PID kontrolörün matematiksel modeli görülmektedir.

$r(t)$ =Referans Giriş, $y(t)$ =Çıkış

$e(t)=y(t) - r(t)$

$$u(t) = K_p \cdot e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (9)$$

PARÇACIK SÜRÜ OPTİMİZASYONU(PSO) ALGORİTMASI

Parçacık Sürü Optimizasyonu (PSO) 1995’te Dr. Eberhartve Dr. Kennedy tarafından geliştirilmiş popülasyon temelli sezgisel bir optimizasyon tekniğidir. Kuş veya balık sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir (Alhasan H., 2017).

Parçacık Sürü Optimizasyonunda öncelikle rastgele belirlenmiş olan olası bir çözüm kümesi oluşturulur. Burada oluşturulan çözüm kümesinin her satırı bir parçacığı her sütunu ise sürüyü oluşturmaktadır. Burada oluşturulan matris aslında rastgele belirlenmiş K_p , K_i ve K_d değerleridir.

Bu matrisin her bir elemanı optimize edilecek sisteme uygulanır ve uygulanan parçacıkların en iyisi olan pbest ve sürünün en iyisi olan gbest hesaplanır.

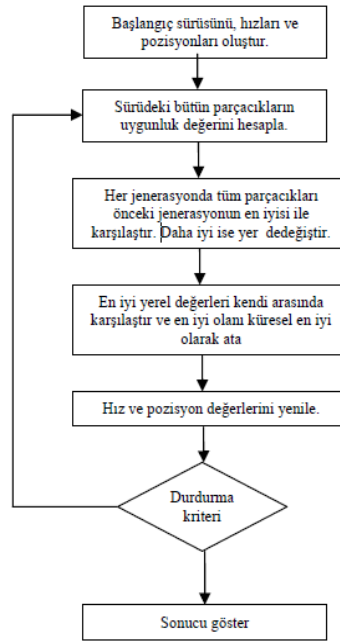
İlk iterasyonda her bir parçacığın pbest kendisine eşit olmaktadır. Buradan elde edilen en iyi değere karşılık gelen parçacık gbest olarak seçilir. Bu iki en iyi değer bulunduktan sonra;

Parçacık, hızı ve konumu sırasıyla aşağıdaki (10) ve (11) denklemlerine göre güncellenir. Yeni konum değerleri hesaplandıktan sonra bu değerler kullanılarak tekrar iterasyonun başına dönülür ve uygunluk değerleri hesaplanıp, pbest ve gbest değerleri güncellenir. Sonlanma şartı sağlanıncaya kadar işlemlere devam edilir (Doğmuş vd.,2017; Develi, 2004).

$$V_i^{k+1} = V_i^k + c_1 * rand_1^k * (pbest_i^k - X_i^k) + c_2 * rand_2^k * (gbest^k - X_i^k) \quad (10)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (11)$$

Burada rand(0-1) arasında rastgele üretilen bir sayıyı, i parçacık numarasını, k ise iterasyon sayısını gösterir. c_1 ve c_2 öğrenme faktörleridir. Bunlar parçacıkları pbest ve gbest konumlarına doğru yönlendiren sabitlerdir. c_1 parçacığın kendi tecrübelerine göre, c_2 ise sürüdeki diğer parçacıkların tecrübelerine göre hareketi yönlendirir. Düşük değerler seçilmesi parçacıkların hedef bölgeye doğru çekilmeden önce, bu bölgeden uzak yerlerde dolaşmalarına imkân verir. Ancak hedefe ulaşma süresi uzayabilir. Diğer yandan, yüksek değerler seçilmesi, hedefe ulaşmayı hızlandırırken, beklenmedik hareketlerin oluşmasına ve hedef bölgenin es geçilmesine sebep olabilmektedir.



Şekil 3.PSO Akış Diyagramı

ADAPTİF PARÇACIK SÜRÜ OPTİMİZASYONU

Kennedy ve Eberhart tarafından Parçacık sürüsü ilk gelişmesinden bu yana, araştırmacılar tarafından optimizasyon algoritmasını geliştirmek için yöntemler önerilmiştir. Yapılan çalışmalar ışığında atalet ağırlığı(w)nın küresel arama yeteneği ve yerel arama yeteneği arasındaki dengede kilit rol oynadığı anlaşılmıştır. Atalet ağırlığı(w) yükseldiğinde küresel arama yeteneği güçlenmekte fakat yerel arama yeteneği zayıflamaktadır. Atalet ağırlığı(w) düşürüldüğünde ise tam tersi gerçekleşmekte küresel arama yeteneği zayıflamakta, yerel arama yeteneği güçlenmektedir. Aradaki bu denge PSO algoritmasının performansını artırmaktadır.

Eberhart ve Shi (Eberhart ve Shi, 2001), rastgele atalet ağırlığı stratejisini önerdiler ve deneysel olarak bu stratejinin PSO algoritmasını daha hızlı yakınsadığını keşfettiler.

$$w = 0.5 + \frac{rand()}{2} \quad (12)$$

Doğrusal olarak azalan strateji(Xin, J., ve ark. 2009), PSO nun performansını ve verimliliğini arttırır. Deneysel olarak atalet ağırlığının 0.9 dan 0.4 e düşürülmesinin optimizasyon sonuçlarına katkı sağladığı keşfedilmiştir.

$$w_k = w_{max} - (w_{max} - w_{min}) \times \frac{iter}{iter_{max}} \quad (13)$$

Çalışmada kullanılan strateji, parçacıkların başarısına dayalı atalet ağırlığı güncelleme yöntemidir. (Berber Ö., 2016),

$$\lambda_{1,i}^K = \frac{1}{1+e^{\alpha(F_{p,i}^K - F_{pbest,i}^K)}} \quad (14)$$

$$\lambda_{2,i}^K = \frac{1}{1+e^{\alpha(F_{pbest,i}^K - F_{gbest}^K)}} \quad (15)$$

$$w_i^K = w_{max} - (w_{max} - w_{min}) \times (\lambda_{1,i}^K + \lambda_{2,i}^K) \quad (16)$$

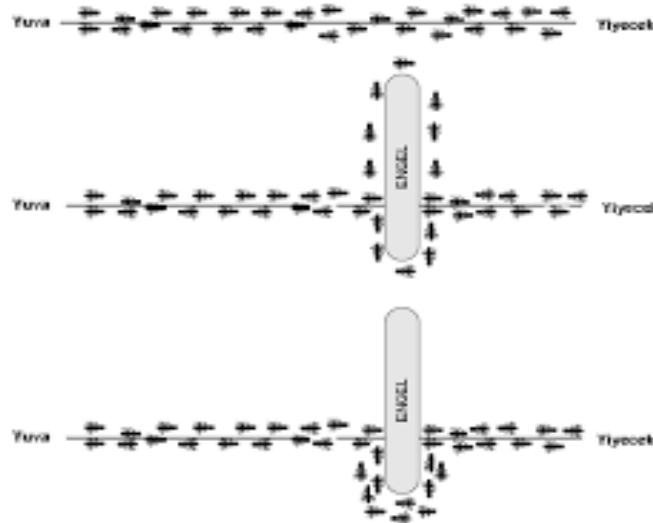
λ_1 ve λ_2 değerleri sürünün optimum sonuca yakınsama ve uzaklaşma durumunu kontrol etmektedir. α katsayı ise sigmoid fonksiyonun dikliğini ayarlanmaktadır. $\alpha < 1$ olması öngörülmüş ve $\alpha = 0,1$ olarak belirlenmiştir.

PSO algoritmasında parçacıkları değerlendirmek için hataların mutlak değerleri toplamı(IAE) fonksiyonu kullanılmıştır. IAE formülü ise;

$$IAE = \int_0^{\infty} |e(t)| dt \quad (17)$$

KARINCA KOLONİSİ ALGORİTMASI

KKA algoritması, gerçek karıncaların yuvaları ile yiyecek noktaları arasındaki en kısa yolu bulma kabiliyetlerinden esinlenilerek geliştirilmiştir. Alternatif yolların söz konusu olduğu durumlarda karıncalar, öncelikle bu yollara eşit olasılıkla dağılırken belli bir süre sonra en kısa olan yolda yoğunlaşmaktadır. Zaman geçtikçe tüm karıncaların en kısa olan yolu kullandıkları görülmektedir (Şekil 4). Bunu yaparken önceki geçişlerden yollarda kalan feromon izlerinden faydalanmaktadırlar. Temel kural, feromon miktarının yoğun olduğu yolun tercih edilme olasılığının yüksek olmasıdır. Görme duyuları çok gelişmemiş olan karıncalar yol tercihlerini feromon izlerine göre yapmaktadırlar.



Şekil 4. Karıncaları En Kısa Yolu İzlemesi

Kısa olan yolda feromon miktarı uzun yollara nispeten daha fazla birikmektedir. Kısa olan yoldan geçiş daha hızlı gerçekleşeceğinden, birim zamanda geçiş yapan karınca sayısı uzun yola göre daha fazla olacaktır. Dolayısıyla herhangi iki düğüm arasındaki yol üzerinde bulunan feromon miktarı, yolun uzunluğuyla ters orantılıdır.

Karınca kolonisi algoritmasında yol tercihi belli bir olasılığa bağlı olarak iki şekilde gerçekleştirilir: İlk seçenek q_0 olasılıkla feromonun en yoğun olduğu yolun seçilmesidir. q_0 parametresi genellikle % 90 olarak belirlenir. $\tau(i,j)$ i ve j noktaları arasındaki feromon miktarı, seçilebilirlik parametresi $\eta(i,j)$, i ve j noktaları arasındaki mesafenin tersi ($1/\delta(i,j)$), α ve β ayarlanabilir parametreler olmak üzere, i noktasında bulunan bir karıncanın gideceği nokta aşağıdaki gibi seçilmektedir:

$$j = \max_{u \in J_k(i)} \left\{ [\tau(i,u)]^\alpha \times [\eta(i,u)]^\beta \right\} \quad \text{eğer } q \leq q_0 \quad (18)$$

İkinci seçenek ise gidilmesi mümkün olan yollardan birini, yollardaki feromon izleriyle orantılı olarak seçmektir. Bu şekilde yol seçimi olasılığı $(1-q_0)$ oranındadır. $J_k(i)$, i noktasındaki karıncanın gidebileceği noktalar yani ziyaret edilmemiş şehirleri temsil eder. Tüm şehirler için seçilme olasılıkları aşağıdaki gibi hesaplanmaktadır:

$$p_k(i,j) = \begin{cases} \frac{[\tau(i,j)]^\alpha \times [\eta(i,j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i,u)]^\alpha \times [\eta(i,u)]^\beta} & \text{eğer } j \in J_k(i) \\ 0 & \text{Diğer durumlarda} \end{cases} \quad (19)$$

Karınca Kolonisi Algoritmasında feromon yenilemesi yerel ve küresel olmak üzere iki kısımda gerçekleşmektedir. Çözüm uzayındaki bir yol üzerindeki toplam feromon düzeyi; yerel ve küresel feromon düzeyinin toplamından oluşmaktadır.

Yerel feromon güncellemesi;

$\tau(i,j)$, t iterasyonuna kadar biriken feromon düzeyi, $\Delta\tau_{ij}^k(t+1)$ t iterasyonundaki feromon düzeyi ve ρ $0 \leq \rho \leq 1$ feromon buharlaşma parametresi olmak üzere yerel feromon düzeyi aşağıdaki formülle hesaplanır.

$$\tau(i,j)(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t+1) \quad (20)$$

$\sum_{k=1}^m \Delta\tau_{ij}^k(t+1)$, aşağıdaki formülle hesaplanır:

$$\Delta\tau_{ij}^k(t+1) = \begin{cases} 1/L^k(t+1) & \text{k karıncası (i, j) yolunu kullanmışsa,} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (21)$$

$L^k(t+1)$ karıncasının toplam tur uzunluğudur. Yerel feromon güncellemesi, turları dinamik olarak değiştirerek geçiş yapılan yolları cazip hale getirir. Karıncalar değişen feromon miktarlarına bağlı olarak her iterasyonda turlarını da değiştirmektedirler. Böylelikle sürekli olarak daha kısa turları bulmak amaçlanmaktadır.

KKA' da, küresel feromon güncellemesi, geçerli iterasyondaki en iyi sonuca sahip karıncanın izlediği yolun feromon düzeyinin artırılmasından oluşur ve iterasyonlarda bulunan en iyi sonuçların belli bir oranda ileriki iterasyonlara aktarılmasını sağlar.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^k(t+1) \quad (22)$$

$$\Delta\tau_{ij}(t+1) = \begin{cases} \frac{1}{L_{best}(t+1)} & (i,j) \text{ en iyi tura ait ise} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (23)$$

$L_{best}(t + 1)$ geçerli iterasyonda bulunan en iyi turun uzunluğudur.

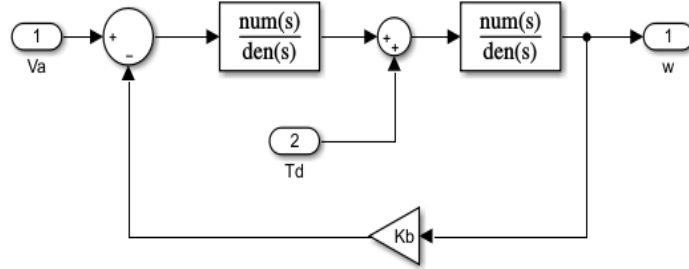
q_0 değeri en iyi çözümün sonraki iterasyonlara aktarılması olasılığını belirleyen parametredir.

α değeri ilgili yolun feromon miktarının önemini belirler ve önceki iterasyonların sonuçlarının ilerleyen iterasyonlara aktarılmasını temin eder. α değerinin yüksek olması feromonun yoğun olduğu yolların seçilme olasılığını artırırken tesadüfiliği azaltmaktadır.

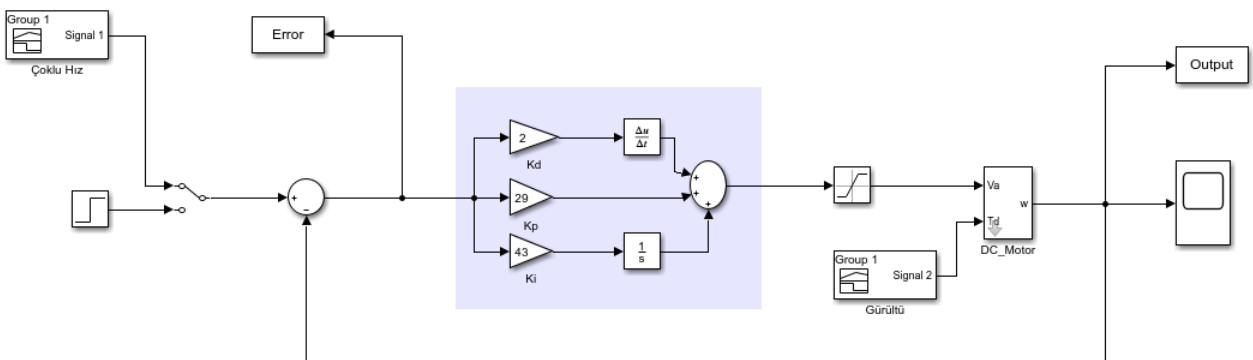
β değeri ise yol uzunluklarının, bir sonraki noktanın seçimindeki etkisini belirlemektedir. β değeri arttıkça bir sonraki yolun seçiminde tesadüfilik artmaktadır. β 'nin düşük olması ise alternatif çözümlerin araştırılması ihtimalini azaltır (Keskintürk vd.,2006).

SİMÜLASYON SONUÇLARI

Bu bölümde, DA Motorun PID kontrolör ile kontrolü için 3 farklı yöntem uygulanmış ve sonuçlar birbirleriyle karşılaştırılmıştır. PID kontrolör, analiz edilen yöntemler neticesinde bulunan, optimum PID kazanç parametrelerine göre tasarlanmıştır. Uygulanan yöntemlerin sonucunda elde edilen K_p , K_i ve K_d parametreleri birbirleriyle karşılaştırılmıştır. PID kontrolör ve DA Motor Matlab/Simulink ile modellenmiştir. Şekil 5 te DA motor modeli gösterilmiştir. Şekil 6 da ise sistemin modeli gösterilmiştir. Bu model üzerinde üç farklı optimizasyon test edilmiş ve sonuçlar karşılaştırılmıştır.



Şekil 5. DA Motor Simulink Modeli



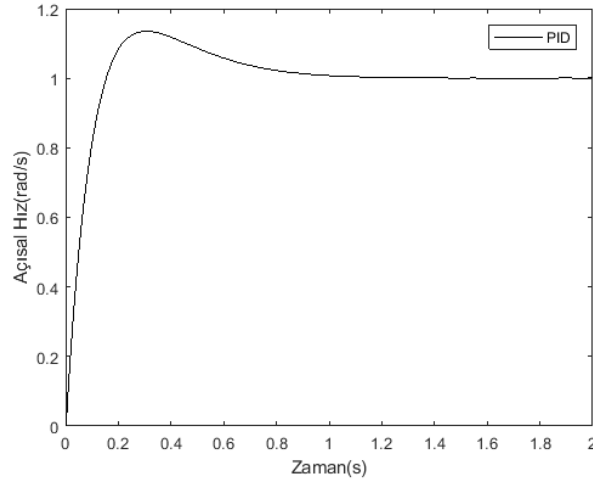
Şekil 6. Uygulanan Sistem Simulink Modeli

Simülasyon sonuçlarında sistemin optimum sonuca ulaşması için elde edilen PID parametreleri Tablo 2'deki gibidir.

Tablo 2. Optimizasyon Yöntemleri Neticesinde Elde Edilen PID Parametreleri

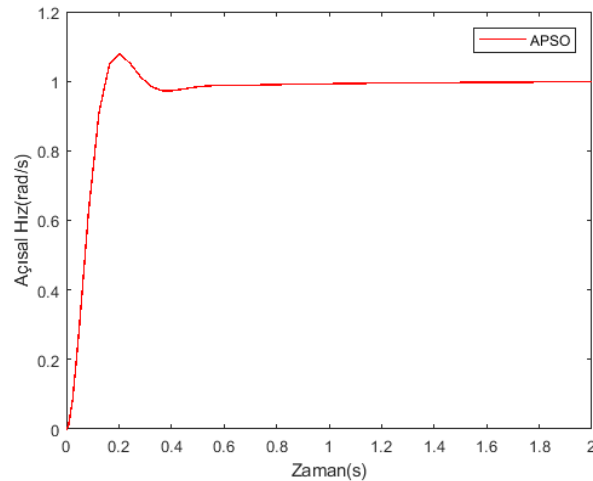
	K_p	K_i	K_d
PID	750	2467	57
APSO	365.68	61.135	1.079
KKA	47	49	2

Test edilen ilk method, Klasik PID algoritmasıdır. DA Motor sistem modelinin Klasik PID kontrolcüsü ile tasarlanan zamana bağlı açısal hız grafiğın Şekil 7deki gibidir.



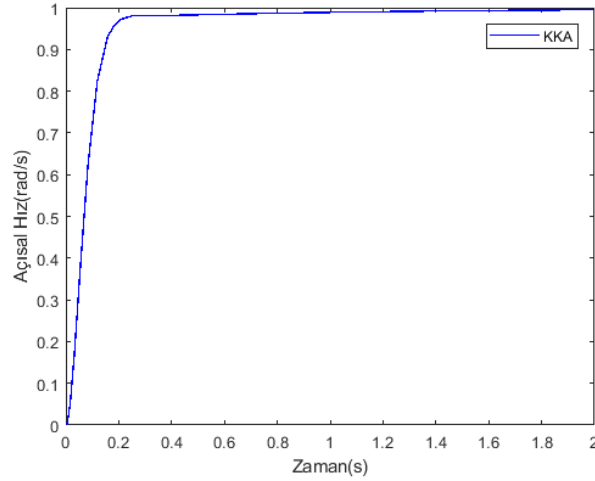
Şekil 7. Klasik PID Denetleyicili DA Motor Sisteminin Zamana göre Açısal Hız Tepkisi

İkinci yöntem Adaptif PSO, bu yöntem de w değerleri her iterasyon içerisinde parçacıkların başarısına bağlı olarak güncellenmektedir. Eylemsizlik ağırlığı w denklem (16) a göre belirlenmektedir. Hızlanma katsayıları c_1 ve c_2 '2' olarak ayarlanmıştır. DA Motor modelinin zamana bağlı tepkisi Şekil 8'deki gibidir.



Şekil 8. APSO PID denetleyicili DA Motor Sisteminin Zamana göre Açısal Hız Tepkisi

Son yöntem olarak KKA uygulanmıştır, Uygulanan algoritmada $\alpha = 0.8$, $\beta = 0.2$ ve buharlaşma katsayısı $\rho = 0.7$ olarak alınmıştır. DA motor modelinin zamana bağlı tepkisi Şekil 9 daki gibidir.



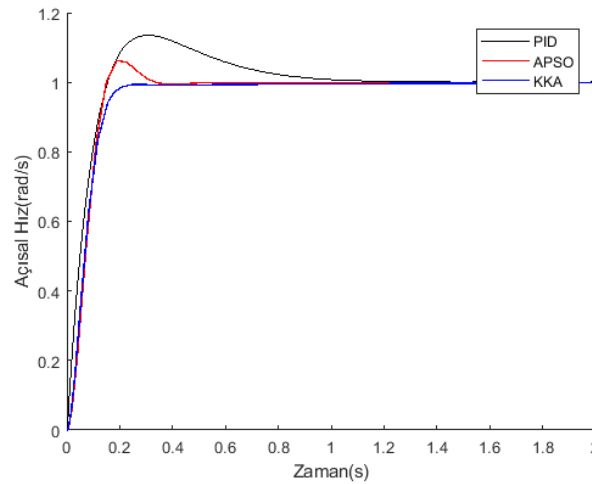
Şekil9. KKA PID denetleyicili DA Motor Sisteminin Zamana göre Açısal Hız Tepkisi

Ayar işlemi PID, APSO ve KKA ile yapılan PID denetleyicili DA Motor sisteminin zaman tepkisi Tablo 3 deki gibidir;

Tablo 3. DA Motorun Açısal Hızının Zaman Tepkisi Özellikleri

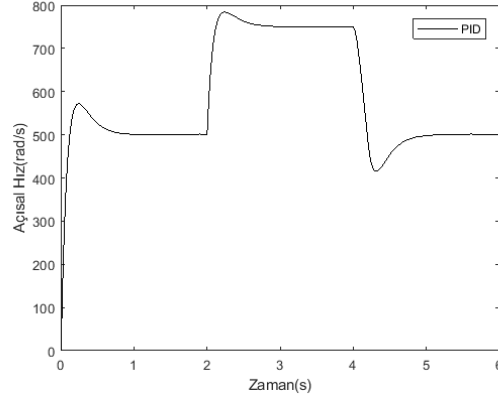
	Klasik PID	Adaptif PSO	KarıncaKolonisi Algoritması
YükselmeZamanı(s)	0.111	0.096	0.123
YerleşmeZamanı(s)	1.820	0.464	0.247
Aşım(%)	13.473	7.876	0
Kalıcı durum hatası	0.04	1.41	1.95

Üç yöntemin kıyaslamalı gösterimi Şekil 10 da görülmektedir.

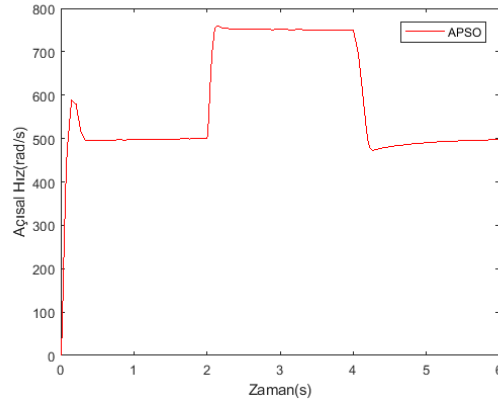


Şekil 10. Üç Yöntem ile Optimizasyon Sonucunun Kıyaslamalı Gösterimi

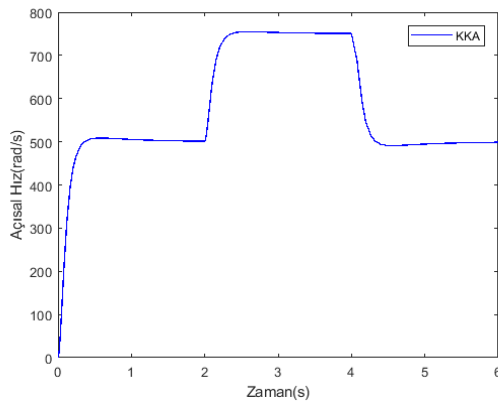
Kullanılan yöntemlerin iki farklı hız üzerindeki tepkisi incelenmiştir. İlk 2 sn. de hız 500 rad/s iken 750 rad/s ye çıkarılmış, 4. sn de ise tekrar 500 rad/s ye düşürülmüştür. Sonuçlar aşağıdaki gibidir.



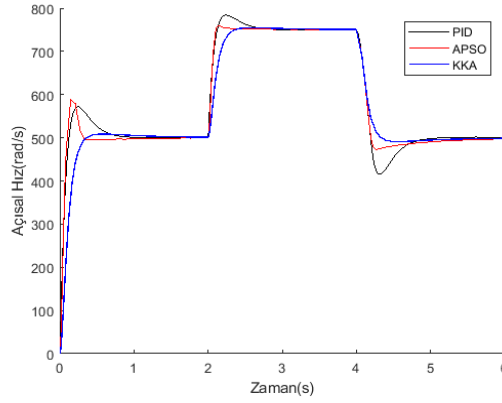
Şekil 11. Klasik PID Denetleyicili Sistem Tepkisi



Şekil 12. APSO PID Denetleyicili Sistem Tepkisi



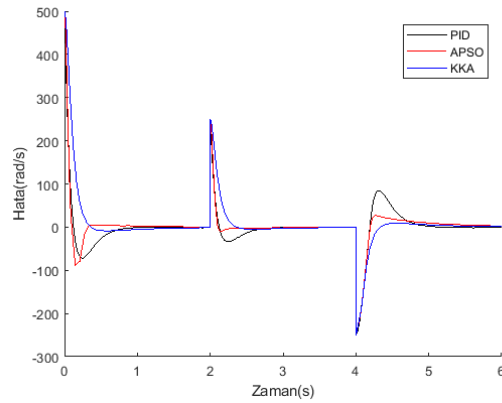
Şekil 13. KKA PID Denetleyicili Sistem Tepkisi



Şekil 14. Üç Yöntem Karşılaştırmalı Sistem Tepkisi

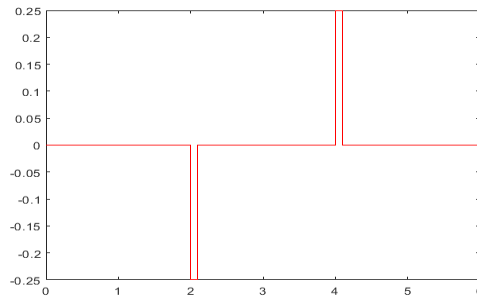
Tablo 4. DA Motorun Açısal Hızının İkili Hız Kalıcı Hal Hataları

Kalıcı Hal Hatası (%)	Klasik PID	Adaptif PSO	KarıncaKolonisi Optimizasyonu
0-2 sn.	0	0	0
2-4 sn.	0.46	0.3	0.08
4-6 sn.	0.1	0.04	0.24

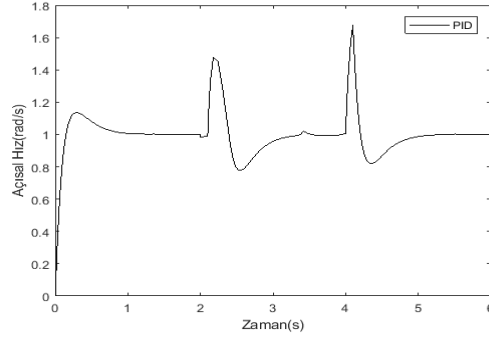


Şekil 15. Üç Yöntem Karşılaştırmalı Çoklu Hız Sistem Hatası

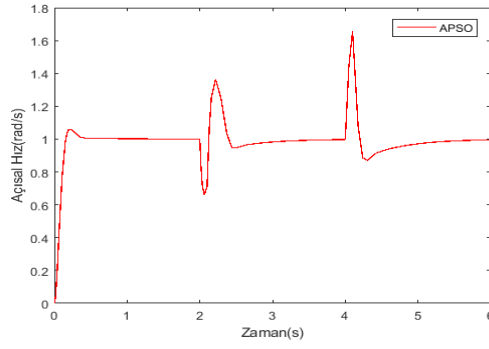
Uygulanan 3 farklı methodun Şekil 16 daki bozucu etkiye karşı sistem tepkisi test edilmiştir ve sonuçlar aşağıdaki gibidir. Bozucu giriş DA motora uygulanan gürültüyü temsil etmektedir.



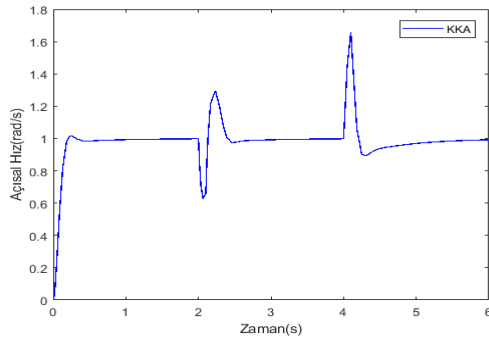
Şekil 16. Sisteme Uygulanan Bozucu Etki



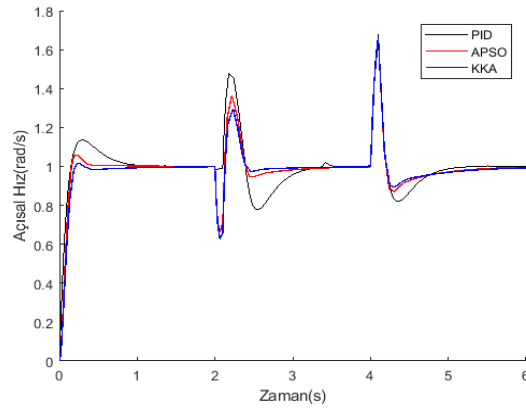
Şekil 17. Klasik PID BozucuEtkiSistemTepkisi



Şekil 18. APSO PID BozucuEtkiSistemTepkisi



Şekil 19. KKA PID BozucuEtkiSistemTepkisi



Şekil 20. ÜçYöntemBozucuEtkiSistemTepkisi

SONUÇLAR

Sistemin optimum sonuca ulaşma aşamasında, basamak girdi testinde, yerleşme zamanı, aşım göz önünde bulundurulduğunda KKA'nın en başarılı optimizasyon olduğu görülmekte fakat kalıcı hal hatasında en başarısız yöntem olduğu görülmektedir.

Çoklu hız testi gözönüne alındığında referans hız değişimlerine KKA kolonisinin daha iyi uyum sağladığı görülmektedir. APSO ve Klasik PID, KKA optimizasyonuna göre daha fazla aşım ve yerleşme zamanı ile sistemi referans hıza ulaştırmaktadır.

Gürültü testi sonucunda ise KKA algoritması diğer iki yöntemle göre daha iyi sonuçlar vermekte ve sistemi referans değere daha hızlı ve daha az bir aşım ile getirmektedir.

Bu üç testin sonucunda kullanılan yöntemlerin DA Motor hız optimizasyon işlemi başarı sırası KKA, APSO, Klasik PID olarak sıralanmaktadır.

KAYNAKLAR

Açıkgöz H., Gani A., Keçecioglu Ö., Şekeli M., Sıvı Sıcaklık Denetiminde PSO Tabanlı PID Denetleyicinin Başarımının İncelenmesi, *Otomatik Kontrol Ulusal Toplantısı, TOK*.

Açıkgöz H., Keçecioglu Ö., Gani A., Şekeli M. (2014), Speed Control of Direct Torque Controlled Induction Motor By using PI, Anti-Windup PI and Fuzzy Logic Controller, *International Journal of Intelligent Systems and Applications in Engineering*.

Akyol S., Alataş B. (2012), Güncel Sürü Zekâsı Optimizasyon Algoritmaları, *Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi* 136-50.

Alhasan H. (2017), A New Adaptive Particle Swarm Optimization Based on Self-Tuning of PID Controller for DC Motor System, *Çukurova University Journal of the Faculty of Engineering and Architecture*, 32(3), pp. 243-249.

Berber Ö. (2016), Parçacık Sürü Optimizasyonu ve PID ile Mobil Robotun Optimum Yörünge Kontrolü, *KSU Mühendislik Bilimleri Dergisi*, 19(3), 167-168.

Cücü G. (2010), Yenilenen Parçacıklı Sürü Optimizasyonu ile PID Parametrelerinin Ayarlanması, *Politeknik Dergisi Cilt:13 Sayı: 2 s. 83-92*.

Çavuşlu M., Karakuzu C., Şahin S. (2010), Parçacık Sürü Optimizasyonu Algoritması ile Yapay Sinir Ağı Eğitiminin FPGA Üzerinde Donanımsal Gerçeklenmesi, *Politeknik Dergisi Cilt:13 Sayı: 2 s. 83-92*.

Develi H. (2004), Ziegler-Nichols Yöntemi ve Migo Yaklaşımı, *İstanbul Teknik Üniversitesi Yüksek Lisans Tezi*, s. 21.

Doğmuş, O., Kılıc, E., Şit, S., Güneş, M. (2017), PSO Algoritması ile Optimize Edilmiş PID Denetleyicinin Fotovoltaik MPPT Sistemine Uyarlanması. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 20 (4), 1-8.

Eberhart, R.C., Shi, Y. (2001), Tracking and Optimizing Dynamic Systems with Particle Swarms, *In Evolutionary Computation*. vol. 1, 94-100.

Erdoğan P., Yalçın E. (2015), Parçacık Sürü Optimizasyonu ile Kısıtsız Optimizasyon Test Problemlerinin Çözümü, *İleri Teknoloji Bilimleri Dergisi*, 4(1), 14-22.

Erkol H. (2017), GA ve PSO ile Kontrol Parametrelerinin Optimizasyonu, *Karaelmas Fen ve Muh. Derg.*, 7(1):179-185.

Erol H., GA ve PSO ile Kontrol Parametrelerinin Optimizasyonu, *Karabük Üniversitesi, Teknoloji Fakültesi, Mekatronik Mühendisliği Bölümü*, Karabük, Türkiye.

- Gani A., Keçecioglu Ö., Açıkgoz H., Kuyumcu M., Kapalı Bir Ortamdaki Sıvının Sıcaklık Denetiminde PID Denetleyicinin Performans Analizi, *Mühendislik ve Teknoloji Sempozyumu, Çankaya Üniversitesi*.
- Iwasaki N., Yasuda K.(2005), Adaptive Partical Swarm Optimization via Velocity Feedback, *IEEEJ Trans. EIS*, Vol.125, No.6.
- Kennedy, J. and EberhartR. C. (1995),“Particle Swarm Optimization” Proc. IEEE int'lconf. On Neural Networks Vol. IV, pp. 1942-1948. *IEEE Service Center, Piscataway, NJ*.
- Kesintürk T., Söyler H. (2006), Global Karınca Kolonisi Optimizasyonu, *Gazi Üniv. Müh. Mim. Fak. Der.*,Cilt 21, No 4, 689-69.
- Kılıç, E, Özçalık, H.R, Şit, S. (2016a), Üç Fazlı Asenkron Motor İçin Uzay Vektör Darbe Genişlik Modülasyonu Kullanan Yapay Sinir Ağı Temelli Adaptif Hız Kontrol Sistemi Tasarımı, *KSU Mühendislik Bilimleri Dergisi*, 19(3), 24-32.
- Kılıç, E, Özçalık, H.R, Doğmuş, O. (2016b), Adaptive PI based on RBF Neural Network for Speed Control of Induction Motor, *International Conference on Natural Science and Engineering (ICNASE'16)*, March 19-20, Kilis, Türkiye.
- Li X, Chen M. AndTsumotu Y. (2013), A Method of Searching PID Controller's Optimized Coefficients for Buck Converter Using Particle Swarm Optimization, 2013 *IEEE 10th International Conference on Power Electronics and Drive Systems (PEDS)*.
- NagarajB., Murugananth N. (2010), A Comparative Study of PID Controller Thning Using GA, EP, PSO and ACO,978-1-4244-7770-8/10/\$26.00 ©2010 IEEE.
- Özçalık, H.R, Gani, A, Kılıç, E, Keçecioglu, Ö.F. (2014), Farklı Kural Tabanları Kullanarak PI-Bulanık Mantık Denetleyici ile Doğru Akım Motorunun Hız Denetim Performansının İncelenmesi, *Akademik Platform Mühendislik ve Fen Bilimleri Dergisi*, 2 (1), 16-23.
- Özsağlam M., Çunkaş M. (2008),Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması,*Politeknik Dergisi Cilt:11 Sayı: 4 s.299-305*.
- Shi Y., Particle Swarm Optimization, *Electronic Data Systems, Inc. Kokomo, IN 46902, USA*.
- Solihin M. (2011), Tuning of PID Controller Using ParticleSwarmOptimization (PSO), *Proceeding of theInternational Conference on Advanced Science,Engineeringand Information Technology*.
- Varol H., Bingül Z. (2004), A New PID TuningTechnique Using Ant Algorithm, *Proceeding of the 2004 American Control Conference Boston, Massachusetts June30 -July 2*.
- Xin, J., Chen, G., Hai, Y. (2009), A Particle Swarm Optimizer With Multi-stageLinearly-decreasing InertiaWeight, *In Computational Sciences and Optimization*, 505-508.
- Zhan Z., Zhang J., Li Y., ChungH. (2009),Adaptive Particle Swarm Optimization, *IeeeTransactions On Systems, Man, And Cybernetics—Part B: Cybernetics, VOL. 39, NO. 6*.