

Parçacık Sürü Optimizasyonu ve PID ile Mobil Robotun Optimum Yörünge Kontrolü

Ömer BERBER¹, Muharrem ATEŞ¹, Hussein ALRUIM ALHASAN¹, Mahit GÜNEŞ¹

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

Özetçe

Bu çalışmada, otonom bir mobil robotun modeli üzerinde MATLAB ile yörünge kontrolü Parçacık Sürü Optimizasyonu kullanılarak PID kontrolü simülasyonu yapılmıştır. Mobil robotun kinematik modeli çıkarılmış, PID ve Parçacık Sürü Optimizasyon (PSO) kontrol algoritması hakkında bilgi verilmiştir. PID katsayı parametreleri PSO kullanılarak belirlenmiştir. Ziegler Nichols (ZN) ile bulunan PID katsayıları ile katsayıları PSO ile bulunan PID kontrolünün karşılaştırılması yapılmıştır. Katsayıların PSO ile belirlenmesinden sonra optimum parametrelerin bulunduğu gözlemlenmiştir.

Abstract

In this work, a two PID controllers is designed and tuned using particle swarm optimization (PSO) in order to track a trajectory for autonomous mobile robot. The simulation and results are done using MATLAB environment. A modeling for autonomous mobile robot is done and the tuning process of the PID By using particle swarm optimization is introduced. A comparison between PID parameters with Ziegler Nichols and a PID tuned by particle swarm optimization is done in order to declare the efficient of the pervious evolutionary algorithm. The optimal PID parameters which is determined by the particle swarm optimization method is presented in a table at the conclusion section.

1. Giriş

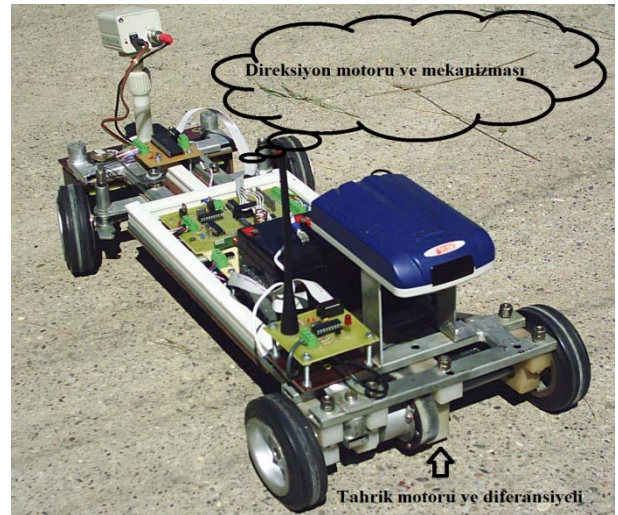
Optimizasyon, bir problemde belirli koşullar altında mümkün olan alternatifler içinden en iyisini seçme işlemidir.[1]. On tane şehri gezmek isteyen bir kişi için diğer bütün parametreler sabit kalmak şartı ile en kısa mesafe kat ederek, en masrafsız ya da en kısa sürede bütün şehirleri hangi sıralama ile gezerim gibi bir problem optimizasyon problemlerine bir örnek olarak verilebilir.

bireysel bazı problemlerin çözümüne ulaşamayan varlıklar toplu olarak hareket ettiklerinde bu problemlerin üstesinden gelebilmektedirler. Doğadaki canlıların topluluk olarak hareketlerinin incelenmesi neticesinde sürü zekasına dayalı kontrol algoritmaları geliştirilmiştir.

Bu çalışmada sürü algoritmalarından olan PSO anlatılmış, mobil robotun kinematik modeli çıkarılmış, bu robotun kontrolü için PID tekniği seçilmiş, PID'nin parametrelerinin belirlenmesi için PSO algoritması ve Ziegler Nichols yöntemi kullanılmış olup elde edilen sonuçlar değerlendirilip çalışma bitirilmiştir.

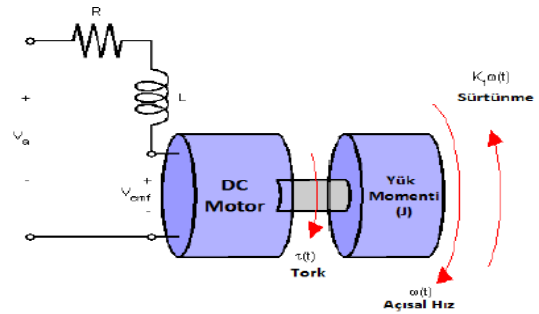
2. Mobil Robotun Modellenmesi

Mobil robotta iki adet DC motor ve dört adet tekerlek bulunmaktadır. Bu DC motorlardan birisi robotun hareketinin sağlanması amacıyla kullanılmış olup arka iki tekerleğin tahriğinde kullanılmaktadır. Diğer DC motor ise robotun yönünün (direksiyonunu) belirlenmesinde kullanılmakta olup ön iki tekerleğin açılarının değişmesinde kullanılmıştır. Bu amaçla öncelikle DC motorun ve robotun simulink modeli çıkarılmıştır.



2.1. DC Motorun Modeli

Bir DC Motora ait model Şekil 1' de görülmektedir.



Şekil 1: DC Motor Modeli

Semboller

Va = Armatür gerilimi (V)

Ra = Armatür direnci (Ω)

La = Armatür endüktansı (H)

Ia = Armatür akımı (A)

Eb = Zıt EMK (V)

 ω = Açısal hız (rad/s) θ = Konum (rad)

Tm = Motor torku (Nm)

Jm = Rotor eylemsizliği (kgm^2)

Bm = Vizkoz sürtünme katsayısı (Nms/rad)

Kt = Tork sabiti (Nm/A)

Kb = Zıt EMK sabiti (Vs/rad)

2.1.1.DC Motorun Çalışmasını İfade Eden Denklemler:

Denklem (1) ve (2)'de motor üzerine düşen gerilimle açısal hız arasındaki ilişki görülmektedir. Denklem (3) ve (4)' te de motorun akımı ile açısal hız arasındaki bağıntı mevcuttur. Denklem (6)' da motorun açısının türevi, motor milinin açısal hızına eşit olduğu görülmektedir.

$$V_a = R_a \cdot I_a + L_a \frac{dI_a}{dt} + E_b \quad (1)$$

$$E_b = K_b \cdot \omega \quad (2)$$

$$T_m = K_t \cdot I_a \quad (3)$$

$$T_m = J_m \frac{d\omega}{dt} + B_m \cdot \omega + T_{yük} \quad (4)$$

$$K_t \cdot I_a = J_m \frac{d\omega}{dt} + B_m \cdot \omega + T_{yük} \quad (5)$$

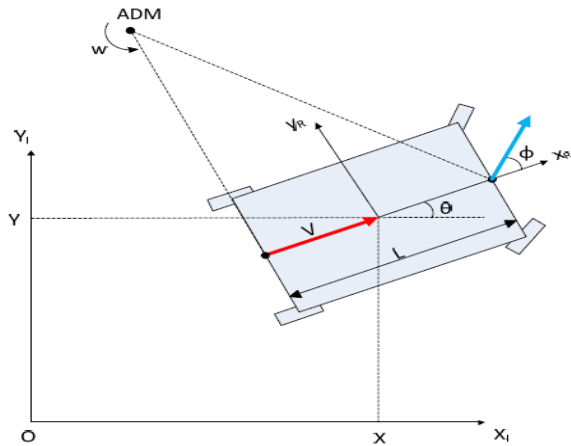
$$\omega = \frac{d\theta}{dt} \quad (6)$$

2.2. Mobil Robotun Modeli

Mobil robotun X eksenini ile yaptığı açı θ olarak belirlenmiştir. Buna göre robotun X ve Y eksenlerinde aldığı yol denklem (7) ve denklem (8)' de verilmiştir.

$$\dot{X} = V \cdot \cos(\theta) \quad (7)$$

$$\dot{Y} = V \cdot \sin(\theta) \quad (8)$$



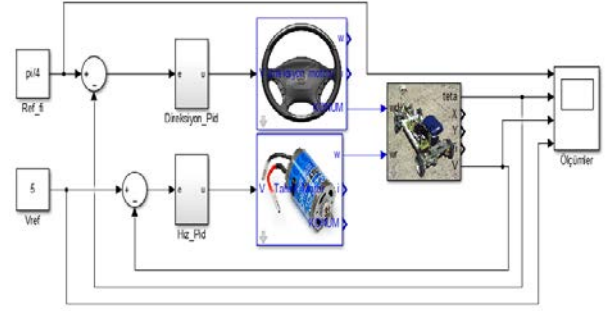
Şekil 2: Mobil robotun X ve Y koordinatlarındaki gösterimi

Mobil robotun hızı, tahrik motorunun açısal hızı ve robotun teker yarıçapı ile olan bağıntısı denklem(9)'da verilmiştir. Aracın hızı (V), θ açısı, aracın ön ve arka teker arasındaki mesafe (l) ve direksiyon açısı (ϕ) arasındaki bağıntı ise denklem(10)' da verilmiştir.

$r = 0,05\text{m}$ ve $l = 0,60\text{m}$ dir.

$$V = \omega \cdot r \quad (9)$$

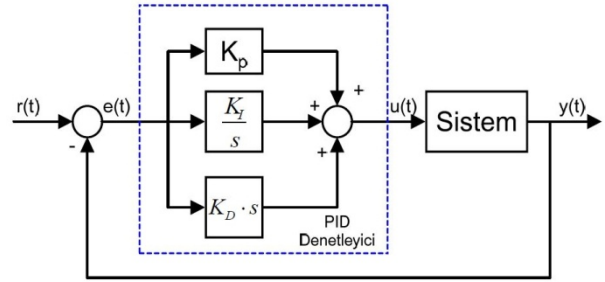
$$\dot{\theta} = \frac{V}{l} \cdot \tan(\phi) \quad (10)$$



Şekil 3: Robotun Matlab Simulink Modeli

2.3. PID Kontrol Algoritması

PID denetleyiciler yapısı çok basit olması, ayarlanacak değişken sayısının az olması ve fiziksel gerçekleşmenin kolay yapılması sayesinde çok tercih edilir. PID (Oransal-İntegral-Türev) kontrol basit yapısı nedeniyle endüstride sıkça kullanılan ve çok bilinen klasik kontrol yöntemlerinden biridir.[2] PID kontrole ait genel blok şeması Şekil 4'te gösterilmektedir.



Şekil 4:Pid Kontrol Şeması

PID'nin matematiksel denklemleri Denklem (11) ve Denklem (12)' de verilmiştir. Burada r(t) giriş (referans) sinyali, u(t) kontrol sinyali, e(t) hata sinyali ve y(t) sistemin çıkışıdır. Kp, Kd, Ki katsayıları ise sırası ile oransal, türevsel, integral katsayılarıdır.

$$u(t) = K_p \cdot e(t) + K_i \int e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (11)$$

$$e(t) = r(t) - y(t) \quad (12)$$

2.4 Parçacık Sürü Optimizasyonu Algoritması

Parçacık Sürüsü (particle swarm) Optimizasyonu (PSO); 1995 yılında J.Kennedy ve R.C.Eberhart tarafından; kuş sürülerinin davranışlarından esinlenilerek geliştirilmiş popülasyon tabanlı stokastik optimizasyon tekniğidir [3].

Nonlineer sistemlerin kontrollerinde kullanılan bir yöntemdir. Parametresi çok olan ve çok değişkenli sistemlere uygulandığında başarılı sonuçlar vermektedir.

PSO, bir grup rastgele çözümle (parçacık sürüsü) başlatılır ve güncellemelerle en uygun çözüm bulunmaya çalışılır.[4] Bütün iterasyon boyunca en iyi iki duruma göre çözümler güncellenir. Bu iki durum pbest ve gbest olarak adlandırılır. Pbest o ana kadar aynı parçacığın elde ettiği en iyi durumu gösterir. Gbest ise tüm parçacıklar arasında o ana kadarki en iyi durumu gösterir. Yani Pbest bireysel, Gbest ise global en iyi çözümdür.

N adet parametrelili D parçacığın matrisel gösterimi aşağıdaki gibidir.

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1D} \\ P_{21} & P_{22} & \ddots & P_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N1} & \cdots & P_{ND} \end{bmatrix}_{N \times D} \quad (13)$$

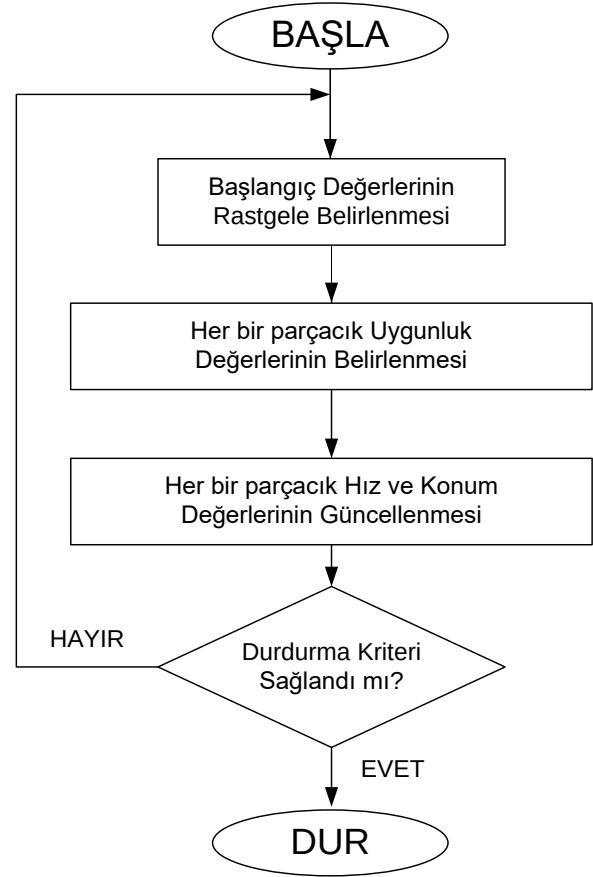
Yukarıda verilen matrisde i'ninci parçacık $P = [P_{i1} \ P_{i2} \ \cdots \ P_{iD}]'$ dir. Bu parçacık için hız değeri $V = [V_{i1} \ V_{i2} \ \cdots \ V_{iD}]$ olarak gösterilir.

Parçacık hızları ve konumlarını gösteren denklemler (14) ve (15)'te verilmiştir. Burada pbest ilgili parçacığa ait o ana en iyi konumu, gbest ise sürünün en iyi parçacığın konumunu belirtmektedir.

$$V_i^{(k+1)} = V_i^{(k)} + c_1 * r_1 * (P_{i,best}^{(k)} - P_i^{(k)}) + c_2 * r_2 * (g_{best}^{(k)} - P_i^{(k)}) \quad (14)$$

$$P_i^{(k+1)} = P_i^{(k)} + V_i^{(k+1)} \quad (15)$$

Yukarıdaki denklemlerde c_1 ve c_2 öğrenme katsayılarıdır. c_1 her bir parçacığın önceki en uygun konumuna doğru, c_2 ise sürünün en uygununa doğru çeken katsayılarıdır. $rand_1$ ve $rand_2$ sayıları ise 0 ile 1 arasında düzgün dağılımlı rastgele sayılardır. PSO' da amaç o ana kadar parçacığın kendi en iyi (pbest) ve sürünün en iyi (gbest) pozisyonunu referans alarak parçacıkları mümkün olan en optimum noktalara götürmektir.



Şekil 5: PSO Algoritması Akış Diyagramı

2.4. Ziegler Nichols Yöntemi

John Ziegler ve Nathaniel Nichols'un 1942 yılında bulduğu PID parametrelerinin belirlenmesinde kullanılan yaygın olarak kullanılan yöntemdir. Ziegler Nichols (ZN) metodunda basamak yanıtı yöntemi ve frekans cevabı yöntemi olarak iki ayrı şekilde PID parametreleri bulunabilir. Bu çalışmada frekans cevabı yöntemi kullanılmıştır.

Frekans cevabı yönteminde parametreler şu şekilde bulunur,

- 1) Kontrolör integral ve türev bileşenleri etkisiz kılınarak otomatik kontrol çevrime bağlanır.
- 2) Kontrolörün Kc kazancı kararlılık limiti aşıncaya kadar artırılır (sistem öz salınımına girer). Salınım periyodu Tc ölçülür.
- 3) Salınım periyodu Tc ve sistemin salınımına başladığı andaki P kontrolörünün kazancı Kc kullanılarak Tablo1'de görüldüğü gibi kontrolör parametreleri elde edilir. Tc : Kritik periyod, Kc : Kritik kazanç olarak anılır. [5]

Tablo 1: Ziegler Nichols frekans yanıtı yönteminde kullanılan PID parametreleri

Kontroller	K	Ti	Td
P	0,5Kc	-	-
PI	0,45Kc	0,85Tc	-
PID	0,5Kc	0,5Tc	0,125Tc

2.5. Simülasyon

Mobil robot 0 ile 20 saniye arasında x eksenini 30°, 20 ile 40 saniye arasında 0°, 40 ile 60 saniye arasında -30°'lik açı ile yol almıştır. PSO'nun sistemi en optimum şekilde kontrol edecek parametreleri aşağıdaki Tablo 1'de gösterilmiştir. Tablo 2'de ise ZN metodu ile bulunan katsayılar verilmiştir.

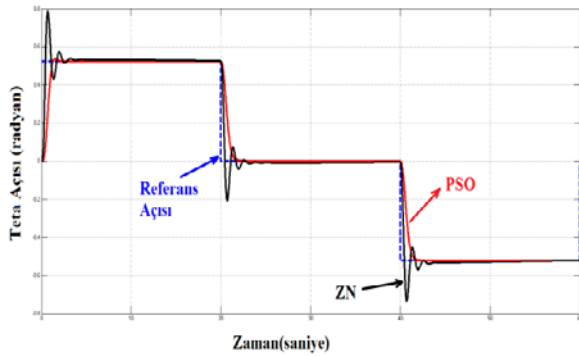
Tablo 3: ZN metodu ile bulunan PID katsayıları

	Kp	Kd	Ki
Hareket Motoru	28.3219	5.7581	573.4249
Yön Motoru	4.891	3.5360	0.01518

Tablo 2: PSO'nun bulunduğu PID katsayıları

	Kp	Kd	Ki
Hareket Motoru	5	25	1600
Yön Motoru	0.378	5	0.055366

Mobil robotun X eksenini ile yaptığı açının (θ açısının) PID parametrelerinin PSO ile hesaplanan ve ZN metodu ile bulunan simülasyon grafiği Şekil 6'da görülmektedir.



Şekil 6: Mobil Robotun Referans Açısı Grafiği

Referans açısı $\pi/6$ radyan (30°) olduğunda overshoot, settling time ve steady state error (kalıcı durum hatası) değerleri aşağıdaki tabloda verilmiştir.

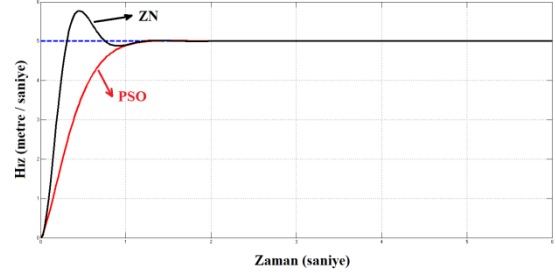
Tablo 4: PSO ve ZN ile bulunan PID katsayılarının overshoot, settling time ve steady state error değerleri

Yüzde olarak overshoot hesaplamaları :
 $PSO = \% 0,0152 / (\pi/6) * 100 = \%2,902$ aşım
 $ZN = \% 0,2657 / (\pi/6) * 100 = \%50,74$ aşım

Yüzde olarak steady state error hesaplamaları:
 $PSO = \% 0,0012 / (\pi/6) * 100 = \%0,2291$

$$ZN = \% 0,0086 / (\pi/6) * 100 = \% 1,642$$

Robotun hızının PSO ve ZN' lu PID parametreleri ile simülasyon grafiği Şekil 7'de verilmiştir. Robot yolculuk boyunca (60 saniye boyunca) 5 m/s hızla gitmektedir.



Şekil 7: Mobil Robotun Hız Simülasyon Grafiği

5 m/s hız değerine göre overshoot, settling time ve steady state error (kalıcı durum hatası) değerleri aşağıdaki tabloda verilmiştir.

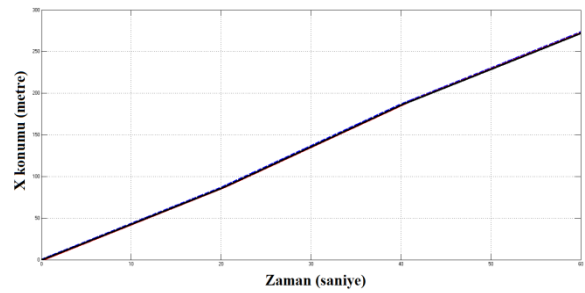
	Overshoot	Settling Time	Steady State Error
PSO	0,0106 m/s	0,8628 saniye	0,0007 m/s
ZN	0,7726 m/s	0,6595 saniye	0,0026 m/s

Tablo 5: Hız simülasyonu sonucu overshoot, settling time, steady state error değerleri

Yüzde olarak overshoot hesaplamaları :
 $PSO = \% 0,0106 / 5 * 100 = \%0,212$ aşım
 $ZN = \% 0,7726 / 5 * 100 = \%15,452$ aşım

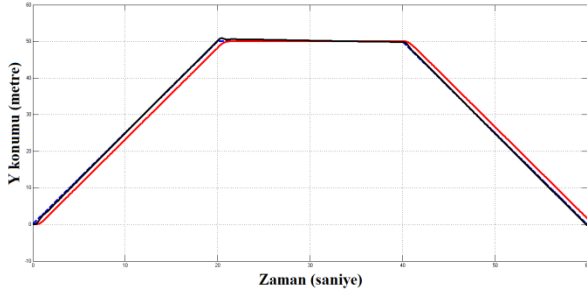
Yüzde olarak steady state error hesaplamaları:
 $PSO = \% 0,0007 / 5 * 100 = \%0,014$
 $ZN = \% 0,0026 / 5 * 100 = \% 0,052$

Şekil 8 ve 9'da mobil robotun X ve Y eksenlerindeki kat ettikleri yol gösterilmiştir.



Şekil 8 : Mobil Robotun X eksenindeki konumu

	Overshoot	Settling Time	Steady State Error
PSO	0,0152 radyan	1,1618 saniye	0,0012 radyan
ZN	0,2657 radyan	2,2318 saniye	0,0086 radyan



Şekil 9: Mobil Robotun Y eksenindeki konumu

	X eksenindeki fark	Y eksenindeki fark
PSO	1 m	1.6 m
ZN	1,45 m	0.25 m

Tablo 6: PSO ve ZN'un bitiş noktasına X ve Y koordinatlarındaki uzaklıkları

Mobil robotun yolculuğu sonunda gerçekteki bitiş noktası ile arasındaki uzaklık farkları

$$PSO = \sqrt{1^2 + 1,6^2} = 1,6 \text{ metre}$$

$$ZN = \sqrt{1,45^2 + 0,25^2} = 1,4924 \text{ metre}$$

3. Sonuç

Bu çalışmada mobil robotun kontrolü PID ile yapılmıştır. Şekil 6 ve Tablo 3' ten de görüleceği gibi PSO ile bulunan PID parametrelerinin, ZN yöntemine göre overshoot'un daha az, settling time'in daha kısa ve steady state error'un daha az olduğu görülmektedir.

Robotun hız verilerine bakıldığında (Şekil 7 ve Tablo 4) PSO'nun settling time'ının biraz daha geç olmasına karşı overshoot ve steady state error'un daha iyi sonuçlar vermektedir.

Şekil 8 ve Şekil 9'dan görüleceği gibi ZN'li PID ve PSO'lu PID arasında pek fazla bir konum farkının olmadığı görülmektedir. Her ikisi de referans bitiş konumuna yakındır.

PSO ile yapılan PID'nin açı ve hız kontrollerinde ZN' a göre özellikle overshoot bakımından daha olumlu sonuçlar verdiği görülmektedir. Dolayısı ile aracın gereksiz hızlanıp yavaşlamaları PSO ile minimize edilecektir.

Kaynakça

- [1] AKYOL S., ALATAŞ B., Güncel Sürü Zekası Optimizasyon Algoritmaları, Nevşehir Üniversitesi Fen Bilimleri Dergisi 1 (2012), s. 36-50
- [2] PID ve Bulanık Mantık ile DC Motorun Gerçek Zamanda DSPIC Tabanlı Konum Kontrolü, Gokhan Tasci, Gurkan Kucukyildiz, H. Metin Ertunc, Hasan Ocak, TOK2012,

Otomatik Kontrol Türk Milli Komitesi 2012 Ulusal Toplantısı, (2012).

- [3] Kennedy, J. and Eberhart, R. C. "Particle swarm optimization" Proc. IEEE int'l conf. on neural networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.
- [4] ÇAVUŞLU M.A., KARAKUZU C., ŞAHİN S. Parçacık Sürü Optimizasyonu Algoritması ile Yapay Sinir Ağı Eğitiminin FPGA Üzerinde Donanımsal Gerçeklenmesi, Politeknik Dergisi, Cilt:13 Sayı: 2 s. 83-92, 2010.
- [5] DEVELİ H., Ziegler Nichols Yöntemi ve Migo Yaklaşımı, İstanbul Teknik Üniversitesi Yüksek Lisans Tezi, s. 21, 2004.