



POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)

URL: <http://www.politeknik.gazi.edu.tr/index.php/PLT/index>

Ters sarkaç sisteminin yapay arı kolonisi algoritması ile optimizasyonu

Optimization of an inverted pendulum system by the artificial bee colony algorithm

Yazar(lar) (Author(s)): Hüseyin Oktay ERKOL

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): Erkol H. O., "Ters sarkaç sisteminin yapay arı kolonisi algoritması ile optimizasyonu", *Politeknik Dergisi*, 20(4): 863-868, (2017).

Erişim linki (To link to this article): <http://dergipark.gov.tr/politeknik/archive>

DOI: 10.2339/politeknik.369045

Ters Sarkaç Sisteminin Yapay Arı Kolonisi Algoritması ile Optimizasyonu

Araştırma Makalesi / Research Article

Hüseyin Oktay ERKOL

Karabük Üniversitesi, Teknoloji Fakültesi, Mekatronik Mühendisliği Bölümü, Karabük, Türkiye

(Geliş/Received : 07.11.2016 ; Kabul/Accepted : 31.01.2017)

ÖZ

Bu çalışmada, kullanımı gittikçe yaygınlaşan Yapay Arı Kolonisi Algoritması kullanılarak ters sarkacı denetleyen Oransal-İntegral-Türevsel kontrolcünün optimizasyonu yapılmıştır. Ters sarkacın modellenmesi ve kontrolü simülasyon ortamında yapılmış olup, simülasyon sonucunda elde edilen veriler Yapay Arı Kolonisi algoritmasının geleneksel yöntemlerden daha iyi ve hızlı sonuç verdiğini göstermektedir.

Anahtar Kelimeler: Yapay arı kolonisi, ters sarkaç, oransal-integral-türevsel kontrolcü, optimizasyon.

Optimization of an Inverted Pendulum System by the Artificial Bee Colony Algorithm

ABSTRACT

In this study, optimization of a Proportional-Integral-Derivative controller for an Inverted Pendulum System was realized by the Artificial Bee Colony algorithm which is becoming widely used. A computer simulation was made to test the optimized system. The simulation results show that the Artificial Bee Colony algorithm is faster and give better results than traditional methods.

Keywords: Artificial bee colony, inverted pendulum, proportional-integral-derivative controller, optimization.

1. GİRİŞ (INTRODUCTION)

Optimizasyon “en iyi duruma getirme” anlamına gelmektedir ve bir problem için muhtemel çözümler içerisinde en iyisini bulma işlemidir. Problem karmaşıklıkça, en iyi çözümü bulma işi de zorlaşmaktadır. Bu zorluğu aşmak için birçok yöntem geliştirilmiştir. Günümüzde yapay zeka tabanlı optimizasyon algoritmalarının kullanımı gittikçe yaygınlaşmaktadır. Bu algoritmalar içerisinde ise önemli bir yeri sürü zekâsından esinlenerek geliştirilen algoritmalar tutmaktadır. Doğada sürü olarak yaşayan canlılardan esinlenerek karınca kolonisi algoritması, genetik algoritma, parçacık sürüsü optimizasyonu gibi birbirinden farklı temellere dayanan algoritmalar geliştirilmiştir [1]–[3]. Bu tür algoritmalar problem için en iyi çözümün bulunacağını garanti etmezler ancak en iyi çözüme yakınsarlar [4].

Sürü zekâsını temel alan ve gün geçtikçe yaygınlaşan algoritmalarından biri de Yapay Arı Kolonisi (YAK) algoritmasıdır [5]. İşçi, gözcü ve kâşif arılardan oluşan arı sürülerinin yiyecek bulmak ve toplamak için sergiledikleri davranışlardan esinlenilerek D. Karaboğa tarafından 2005 yılında önerilmiştir [6].

Bu çalışmada YAK algoritması, kontrol teorisinin temel problemlerinden biri olan ters sarkaç sistemine uygulanmıştır. Sistem PID (Proportional-Integral-Derivative, Oransal-İntegral-Türevsel) denetleyici ile kontrol edilmektedir [7]. Kullanılan denetleyici üç farklı katsayıya

sahiptir ve bu katsayılar denetleyicinin karakteristik özellikleri üzerinde etkilidir. Sistemin belirlenen kriterler ile kontrol edilebilmesi için bu katsayıların doğru ayarlanması gerekir. Bu katsayıları ayarlamak için çeşitli yöntemler tanımlanmış olmakla birlikte en iyi katsayıları garanti eden bir yöntem yoktur. Ele alınan arabalı ters sarkaç sistemi simülasyon ortamında modellenmiş, Ziegler-Nichols yöntemi ve YAK algoritması ile optimize edilmiştir. Çalışma sonunda YAK algoritması ile sistem daha kısa sürede, daha yüksek performans gösterecek şekilde optimize edilmiştir.

2. GEREÇ VE YÖNTEM (MATERIAL AND METHODS)

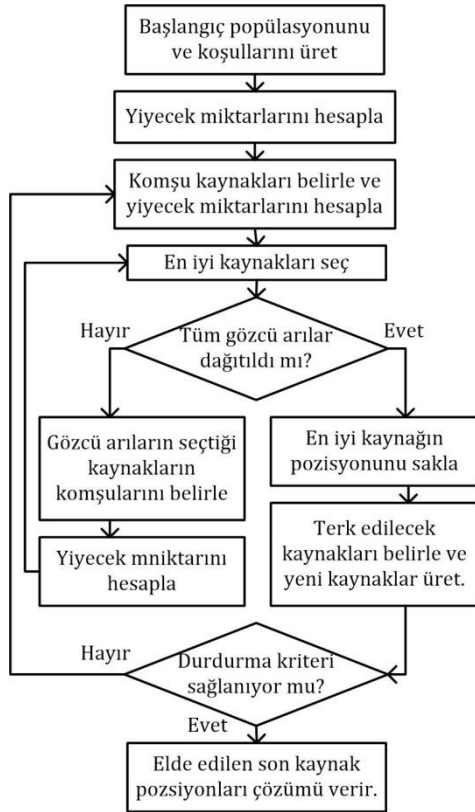
2.1. Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm)

Yapay Arı Kolonisi algoritması doğada bulunan arıların davranışlarından esinlenilerek geliştirilmiş bir optimizasyon algoritmasıdır. İlk olarak Derviş Karaboğa tarafından 2005 yılında önerilmiştir [6]. Bir arı kolonisinde üç tür arı vardır. Bunlar işçi arı, gözcü arı ve kâşif arıdır. Bilinen bir kaynaktan yiyecek taşıyan arıya işçi arı denir. Arılar yiyecek kaynağı bulduklarında bunu diğer arılara dans ederek bildirirler. Dans alanında bekleyerek, gelen bilgilere göre yiyecek kaynağını seçen arılara gözcü arı denir. Rastgele dolaşarak yiyecek arayan arıya ise kâşif arı adı verilir. Koloninin yarısı işçi, yarısı gözcü arılardan oluşur. Her yiyecek kaynağında yalnızca bir işçi arı vardır. Dolayısı ile kolonideki işçi arı veya gözcü arı sayısı çözüm sayısına eşittir. Yiyecek kaynağı tükenen her arı

*Sorumlu Yazar (Corresponding Author)
e-posta : oktayerkol@karabuk.edu.tr

kâşif arıya dönüşür ve yeni bir kaynak arar. Yiyecek kaynağının pozisyonu optimize edilen problem için çözümlerden birini temsil eder. Kaynaktaki yiyecek miktarı ise çözümün uygunluğunun bir ölçüsüdür.

Algoritmanın genel yapısı Şekil 1’de verilmiştir [8]. Öncelikle başlangıç popülasyonu ve işçi arılar için başlangıç pozisyonları belirlenir. İşçi arıların gönderildiği yiyecek kaynağındaki (pozisyon) yiyecek miktarı ve bu kaynağın etrafındaki muhtemel yiyecek kaynaklarında bulunan yiyecek miktarları hesaplanır. En fazla yiyecek bulunan kaynağı işçi arı gözcü arılara haber verir. Gözcü arılar çeşitli olasılık hesapları ile kendilerine bildirilen kaynaklardan birini seçer ve kaynağa giderler. Eğer tüm gözcü arılar kendine bir kaynak bulamamışsa, kendine kaynak bulan gözcü arıların komşulukları belirlenir ve buralardaki yiyecek miktarları hesaplanır. Belirlenen yeni kaynaklar tekrar gözcü arılarla paylaşılır ve gözcü arıların kendine yiyecek kaynağı bulması sağlanır. Bu işlem tüm gözcü arılar bir yiyecek kaynağı bulana kadar devam eder.



Şekil 1. Yapay arı kolonisi algoritması akış diyagramı (Flow chart of the artificial bee colony algorithm)

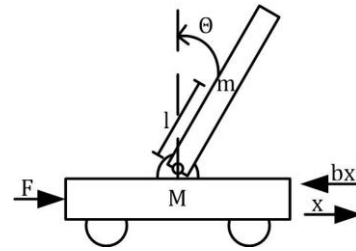
Tüm gözcü arılar yiyecek kaynaklarına dağıtıldıktan sonra en iyi kaynağın pozisyonu hafızaya alınır ve terk edilecek kaynaklar belirlenir. Yiyecek kalitesi ve miktarı bakımından yetersiz görülen kaynaklar terk edilir. En iyi kaynak, optimize edilen problem için çözüme en yakın olan pozisyonudur. Terk edilen kaynaklar ise çözüme uzak olan pozisyonlardır.

Terk edilen kaynaktaki işçi arı kâşif arıya dönüşür ve kendisine rastgele yeni bir yiyecek kaynağı bulur. Temel

ABC algoritmasında her çevrimde yalnızca bir kâşif arının çıkmasına izin verilir [8]. Bulunan tüm kaynaklar değerlendirilir ve en iyi kaynağın durdurma kriterlerini sağlayıp sağlamadığına bakılır. Eğer durdurma kriterleri sağlanıyorsa işlem sona erer. Eğer durdurma kriterleri sağlanmıyorsa elde edilen kaynakların komşulukları incelenerek yeni kaynaklar belirlenir ve algoritma bu döngü içerisinde devam eder. Burada bahsedilen hesaplamaların çoğu çeşitli olasılık işlemleri içermektedir. Ayrıntılar için [8]–[11] numaralı kaynaklar incelenebilir.

2.2. Arabalı Ters Sarkaç (Inverted Pendulum on a Cart)

Ters sarkaç kontrol teorisinde en çok kullanılan problemlerden biridir [12]. Bu çalışmada ele alınan arabalı ters sarkacın yapısı Şekil 2’de, özellikleri Çizelge 1’de verilmiştir. M kütleli bir araba üzerine yerleştirilmiş m kütleli ve $2l$ uzunluğunda ters sarkaçtan oluşan sisteme F kuvveti etki etmektedir. Sarkacın ağırlık merkezi ile arabaya bağlı olduğu mafsal arasındaki mesafe l ile gösterilmiştir. Sürtünme kuvveti bx' F kuvvetine ters yöndedir ve F kuvveti ile aynı yönde x yer değiştirmesi meydana gelmektedir. Arabaya herhangi bir zaman etkiyen F kuvveti nedeniyle araba kuvvet sıfırlansa dahi sarkaç dengesini kaybederek devrilir. Devrilmeye başlayan sarkacın başlangıç pozisyonunu koruması için arabayı ileri geri hareket ettirerek sarkacın başlangıç pozisyonuna gelmesini sağlamak ve üzerine etkiyen yatay kuvvetlerin bileşkesinin de sıfır olmasını sağlamak gerekmektedir.



Şekil 2. Arabalı ters sarkaç sistemi (Inverted pendulum on a cart)

Çizelge 1. Ters sarkaç sisteminin fiziksel özellikleri (The physical properties of the inverted pendulum system)

Sembol	Açıklama	Değer
M	Arabanın ağırlığı	0.5kg
m	Sarkacın ağırlığı	0.2kg
l	Sarkaç uzunluğunun yarısı	0.3m
g	Yer çekim ivmesi	9.81m/s ²
I	Sarkacın atalet momenti	0.006kg.m ²
b	Sürtünme katsayısı	0.1N/m.s ⁻¹
F	Arabaya uygulanan kuvvet	- N
θ	Sarkacın düşey düşey eksenle yaptığı açı	- derece

Şekil 2 'de verilen sisteme ait eşitlikler denklem (1) ve denklem (2)'de verilmiştir. Bu denklemler lineer değildir. Sistemi lineerleştirmek için $\sin\theta \approx 0$, $\cos\theta \approx 1$, $(\theta')^2 \approx 0$ varsayımları yapılmıştır. Bu varsayımlar sarkacın kararlı olduğu hale göre büyük değişimler göstermediği sürece doğrudur. Bu varsayımla birlikte denklem (1) ve (2) yeniden düzenlenirse (3) ve (4) numaralı denklemler elde edilir.

$$F = (M + m) \frac{dx^2}{dt^2} + b \frac{dx}{dt} + ml \frac{d\theta^2}{dt^2} \cos\theta - ml \sin\theta \quad (1)$$

$$(I + ml^2) \frac{d\theta^2}{dt^2} = mgl \sin\theta - ml \cos\theta \frac{dx^2}{dt^2} \quad (2)$$

$$\frac{dx^2}{dt^2} = F \frac{I + ml^2}{I(M + m) + Mml^2} - \frac{dx}{dt} \frac{b(I + ml^2)}{I(M + m) + Mml^2} - \theta \frac{m^2 l^2 g}{I(M + m) + Mml^2} \quad (3)$$

$$\frac{d\theta^2}{dt^2} = -F \frac{ml}{I(M + m) + Mml^2} - \frac{dx}{dt} \frac{mlb}{I(M + m) + Mml^2} - \theta \frac{mgl(M + m)}{I(M + m) + Mml^2} \quad (4)$$

2.3. PID Denetleyici (PID Controller)

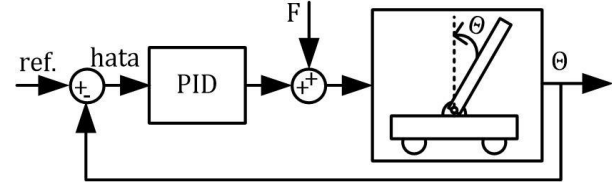
En yaygın kullanılan kontrol yöntemlerinden biri PID'dir. PID ile kontrol için sistemin kapalı çevrim olarak tasarlanması gerekir. Kontrolcü girişine, sisteme girişi olarak verilen referans ve geri besleme olarak alınan sistem çıkış değerinin karşılaştırılmasıyla elde edilen hata değeri verilir ve kontrolcü bu hatayı sıfırlamak için gerekli çıkışı üretir. PID kontrolcünün zaman domeninde genel yapısı denklem (5)'de verilmiştir [13]. Burada K_p oransal kazanç, K_d türevsel kazanç ve K_i integral kazanç katsayısıdır. Bu katsayıların her biri kontrolcünün karakteristik yapısını etkiler ve ilgili sistemin kontrol edilebilmesi için uygun katsayıların belirlenmesi gerekir. Bu belirleme işlemi için kullanılan en klasik yöntem Ziegler-Nichols yöntemidir [14].

$$U(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(\tau) d\tau \quad (5)$$

3. BULGULAR VE TARTIŞMA (FINDINGS and DISCUSSION)

Modellenen arabalı ters sarkaç ve PID denetleyiciden oluşan sistem Şekil 3'de verilmiştir. Sistemin çıkışı sarkacın düşey eksen ile yaptığı açıdır. Sistemin girişine verilen referans değeri, amaç sarkacı dik tutmak olduğu için, sıfırdır. Sistem çıkışındaki değişiklikler referans ile karşılaştırılarak bir hata değeri elde edilir ve bu hata PID

denetleyicinin girişine uygulanır. PID denetleyicinin çıkışına eklenen F kuvveti, herhangi bir zamanda araca etkiyerek sarkacın dengesini bozan F kuvvetidir. Burada ki uygulama için F kuvveti sürekli değildir ve sisteme bozucu giriş etkisi yapmaktadır. Şekil 3'de verilen sistemin, Matlab programı kullanılarak, simülasyonu yapılmış ve sistemi kontrol eden PID denetleyicinin katsayıları hem Ziegler-Nichols hem de ABC algoritması ile belirlenerek sonuçlar tartışılmıştır.



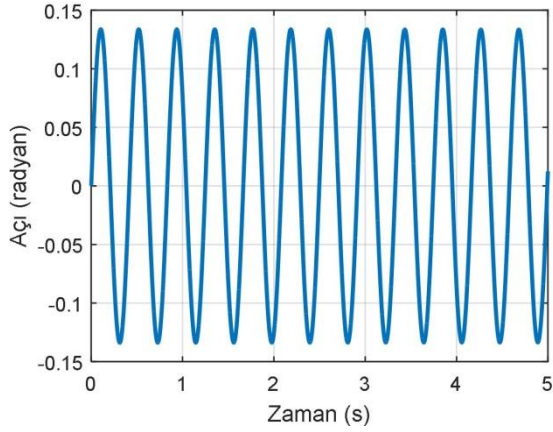
Şekil 3: Denetleyici ve sarkaçtan oluşan kapalı çevrim sistem (Closed-loop system composed of the controller and the pendulum)

PID denetleyici denklem 5'de görüldüğü gibi K_p , K_d ve K_i olmak üzere üç farklı katsayıya sahiptir. Ziegler-Nichols yöntemi uygulanırken başlangıç olarak bu katsayıların hepsi sıfırlanır. Daha sonra sistem çıkışı sabit bir salınım yapana kadar K_p değeri artırılır ve sabit salınım yapmaya başladığı andaki K_p değeri K_u olarak adlandırılır. Sistem çıkışında meydana gelen dalgalının periyodu ise P_u olarak adlandırılır. Bu iki değer yardımı ile PID denetleyicinin katsayıları Çizelge 2'de gösterildiği gibi hesaplanır [15].

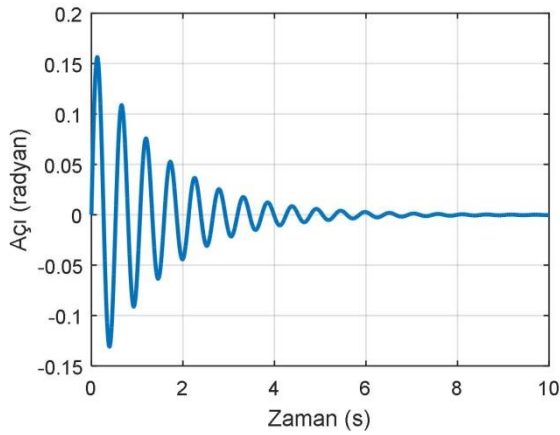
Çizelge 2. Ziegler-Nichols yöntemine göre PID katsayılarının hesabı (Ziegler-Nichols calculation for the PID parameters)

	K_p	T_i	T_d
P	$K_u/2$		
PI	$K_u/2.2$	$P_u/1.2$	
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

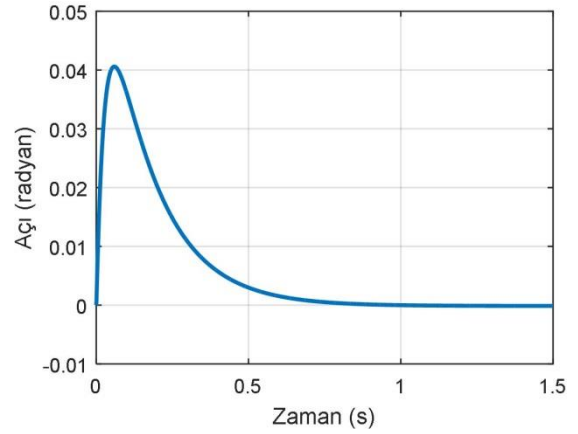
Bu çalışmada sunulan sistemin çıkışının osilasyona girdiği K_p değeri 106, P_u değeri ise 0.43 olarak belirlenmiştir. Osilasyona giren sistem çıkışı Şekil 4'de verilmiştir. Dolayısı ile $K_p=62.352$, $K_i=0.215$ ve $K_d=0.654$ olarak hesaplanmıştır. Bu katsayıların kullanıldığı PID denetleyici ile elde edilen sistem çıkışı Şekil 5'de verilmiştir. Sistemin oturma zamanı 8 saniyeden uzundur ve salınım oldukça fazladır. Sistem parametreleri deneysel olarak değiştirildiğinde sistem çıkışında bir iyileşme gözlenmektedir ve salınım azalmakta, oturma zamanı ise yaklaşık 0.9 saniye olmaktadır. Elde edilen sistem çıkışları; araç pozisyonu, araç hızı, sarkaç açısal pozisyonu ve sarkaç açısal hızı birlikte Şekil 6 ve Şekil 7'de verilmiştir.



Şekil 4. Osilasyona giren sistemin çıkış grafiği (The oscillating output signal of the system)

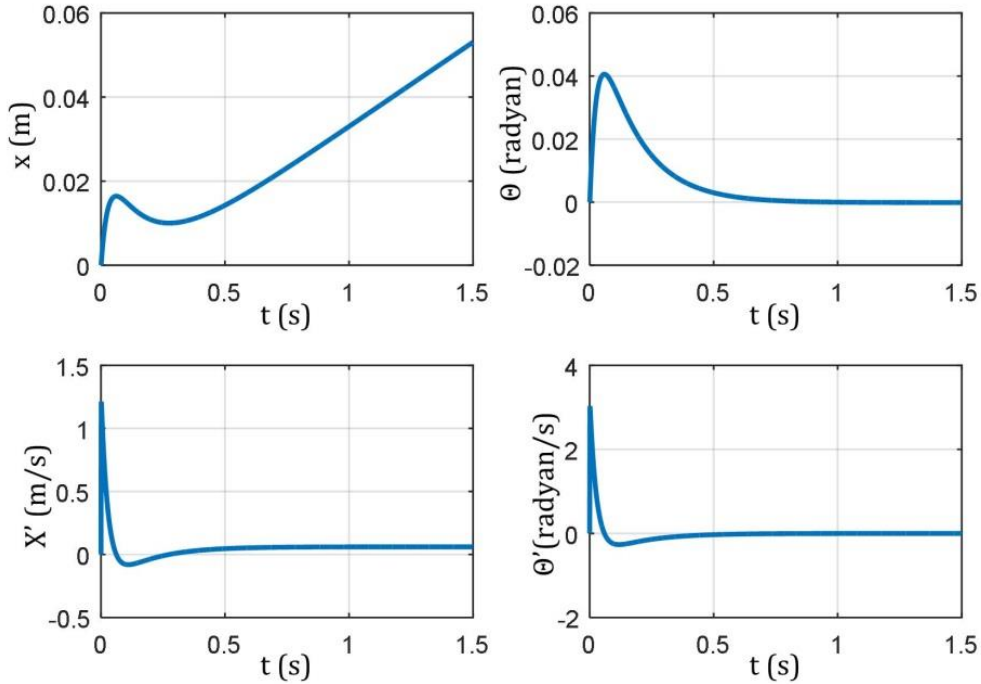


Şekil 5. Ziegler-Nichols yöntemi ile optimize edilen sistemde sarkaç açısının grafiği (Pendulum angle of the system optimized by the Ziegler-Nichols method)



Şekil 6. Deneysel olarak katsayıları iyileştirilmiş sistemde sarkaç açısının grafiği (Pendulum angle of the experimentally tuned system)

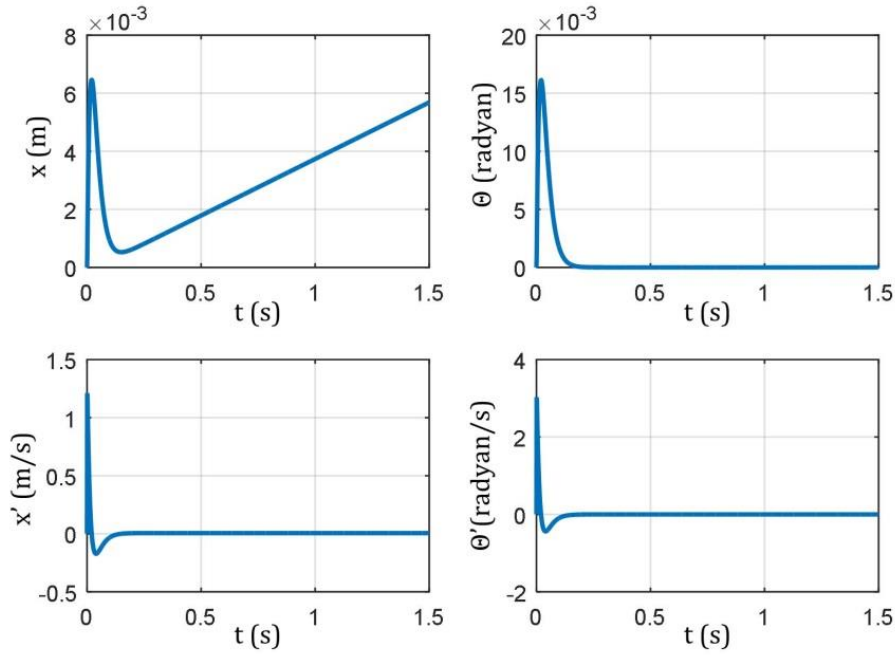
Aynı sisteme ait kontrolcünün optimizasyonu için ABC algoritmasını kullanmak mümkündür. Burada kullanılan ABC algoritmasının detayları için [9], [16], [17] numaralı kaynaklar incelenebilir. ABC ve benzeri sürü algoritmaları ile bir sistemin optimizasyonunu gerçekleştirmek için öncelikle amaç fonksiyon tanımlamak gerekmektedir. Optimizasyon algoritmasının amacı tanımlanan amaç fonksiyonu minimize etmektir. PID denetleyicinin optimizasyonu için literatürde tanımlanmış farklı amaç fonksiyonlar mevcuttur. Mutlak hatanın toplamı (Integral Absolute Error (IAE)), hata karelerinin toplamı (Integral Squared Error (ISE)), zaman ağırlıklı hata karelerinin toplamı (Integral Time-weighted Squared Error (ITSE)) gibi fonksiyonlar önerilmiştir [18]. Bu fonksiyonlar PID



Şekil 7. Katsayıları deneysel olarak iyileştirilen sistemin çıkış grafikleri (Output signals of the experimentally tuned system)

denetleyicinin kontrol ettiği sistemde meydana gelen hataların çeşitli işlemler ile toplamını vermektedir. Bu fonksiyon optimizasyon algoritmasına uygulandığında, belirlenecek PID parametreleri, sistemde minimum hata meydana getiren parametreler olacaktır. Bu çalışmada amaç fonksiyon olarak, denklem (6)'da verilen IAE fonksiyonu kullanılmıştır [18], [19]. Ayrıca kullanılan ABC algoritmasında koloni büyüklüğü 20 (işçi ve gözcü arıların toplam sayısı), maksimum iterasyon sayısı 100 olarak ayarlanmıştır. Değişkenler için başlangıç değerleri atanmamış, sınırları 0-2000 olarak ayarlanmış ve özel bir durdurma kriteri belirlenmemiştir. Algoritmanın ürettiği PID katsayıları; $K_p=1112.298$, $K_d=46.734$ ve $K_i=0$ 'dır. Elde edilen yeni katsayılar ile sisteme ait grafikler Şekil 8'de verilmiştir. Sistemin denge pozisyonuna ulaşma süresi 0.24s'ye düşmüş, ayrıca sarkacın yaptığı maksimum salınım 0.04 dereceden 0.016 dereceye gerilemiştir. Sistemde konum kontrolü yapılmadığı için konum sürekli değişmektedir.

$$f_{IAE} = \int_0^T |e(t)| dt \quad (6)$$



Şekil 8. Yapay arı kolonisi algoritması ile optimize edilen sistemin çıkış grafikleri (Output signals of the system optimized by the artificial bee colony)

4. SONUÇLAR (CONCLUSION)

Bu çalışmada arabalı ters sarkacın PID ile kontrolü gerçekleştirilmiş ve PID parametreleri Ziegler-Nichols yöntemi ve ABC algoritması ile ayarlanmıştır. Ziegler-Nichols yönteminde ilk elde edilen parametreler sistem için uygun parametreler değildir. Deneysel olarak parametrelerin değiştirilmesiyle daha iyi bir sonuç elde edilmiştir. Ancak bu deney sayısı ile orantılı olarak zaman kaybına neden olmaktadır. ABC algoritması ile elde edilen sonuçlar ise Ziegler-Nichols yöntemi ile bulunandan çok daha iyi bir performans sağlamıştır. Bu sonuçlar çok kısa bir sürede (yalnızca 100 iterasyon ile)

elde edilmiş ve tekrar müdahaleye gerek kalmamıştır. Çalışma sonunda ABC algoritmasının kullanıcıya PID optimizasyonunda kolaylık sağladığı ve zaman kazandırdığı görülmüştür. İlerleyen çalışmalarda genetik algoritma, parçacık sürüsü optimizasyonu ve karınca kolonisi gibi algoritmalar ile ABC'nin karşılaştırmalı olarak performans değerlendirmesi planlanmaktadır.

KAYNAKLAR (REFERENCES)

- [1] Blum C., "Ant Colony Optimization: Introduction and Hybridizations", *7th International Conference on Hybrid Intelligent Systems*, Kaiserlautern, Germany, 353-373, (2007).
- [2] Asteroth A. and Hagg A., "How to Successfully Apply Genetic Algorithms in Practice: Representation and Parametrization", *International Symposium on Innovations in Intelligent Systems and Applications*, Madrid, Spain, 1-6, (2015).
- [3] Marini F. and Walczak B., "Particle Swarm Optimization (PSO). A Tutorial", *Chemometrics and Intelligent Laboratory Systems*, 149: 153-165, (2015).
- [4] Akyol S. and Alataş B., "Güncel Sürü Zekâsı

Optimizasyon Algoritmaları", *Neşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 1: 36-50, (2012).

- [5] Karaboga D. and Basturk B., "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", *Journal of Global Optimization*, 39(3): 459-471, (2007).
- [6] Yan G. and Li C., "An Effective Refinement Artificial Bee Colony Optimization Algorithm Based on Chaotic Search and Application for PID Control Tuning", *Journal of Computational Information Systems*, 7(9): 3309-3316, (2011).
- [7] Kumar R. and Singla S.K., "A Comparative Analysis of Different Methods for the Tuning of PID Controller",

- International Journal of Electronics Communications and Electrical Engineering*, 3(2): 1–17, (2013).
- [8] Karaboğa D., "Yapay Zeka Optimizasyon Algoritmaları", Üçüncü baskı, *Nobel*, Ankara, (2014).
- [9] Karaboga D. and Akay B., "A Survey: Algorithms Simulating Bee Swarm Intelligence", *Artificial Intelligence Review*, 31(1-4): 61–85, (2009).
- [10] Mishra A.K., Khanna A., Singh N.K. and Mishra V.K., "Speed Control of Dc Motor Using Particle Swarm Optimization Technique", *Universal Journal of Electrical and Electronic Engineering*, 1(3): 68–75, (2013).
- [11] Verma B.K. and Kumar D., "A review on Artificial Bee Colony Algorithm", *International Journal of Engineering and Technology*, 2(3): 175–186, (2013).
- [12] Jain N., Gupta R. and Parmar G., "Intelligent Controlling of an Inverted Pendulum Using PSO-PID Controller", *International Journal of Engineering Research and Technology*, 2(12): 3712–3716, (2013).
- [13] Montiel O., Sepulveda R., Melin P., Castillo O., Porta M.A. and Meza I.M., "Performance of a Simple Tuned Fuzzy Controller and a PID Controller on a DC Motor", *IEEE Symposium on Foundations of Computational Intelligence*, Honolulu, Hawaii, 531-537, (2007).
- [14] Ziegler J.G. and Nichols N.B., "Optimum Settings for Automatic Controllers", *Journal of Dynamic Systems, Measurement, and Control*, 115(2B): 759-765, (1993).
- [15] Khan M.R., Khan A.A. and Ghazali U., "Speed Control of DC Motor Under Varying Load Using PID Controller", *International Journal of Engineering*, 9(3): 38–48, (2015).
- [16] Karaboga D., Gorkemli B., Ozturk C. and Karaboga N., "A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications", *Artificial Intelligence Review*, 42(1): 21–57, (2014).
- [17] Saab S.M., El-Omari N.K.T. and Owaied H.H., "Developing Optimization Algorithm Using Artificial Bee Colony System", *Ubiquitous Computing and Communication Journal*, 4(5): 15–19, (2009).
- [18] Campo A.B., "PID Control Design, MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1", *InTech*, (2012).
- [19] Hussain K.M., Zepherin R.A.R. and Kumar M.S., "Comparison of Tuning Methods of PID Controllers for FOPTD System", *International Journal of Innovative Research In Electrical, Electronics, Instrumentation and Control Engineering*, 2(3): 1177–1180, (2014).