



Parameter identification of a non-minimum phase second order system with time delay using relay test and PSO, CS, FA algorithms

Murat Erhan Çimen*^{ID}, Ali Fuat Boz^{ID}

Department of Electrical and Electronics Engineering, Sakarya Applied Science University, Sakarya, 54187, Turkey

Highlights:

- Using feedback relays for system identification
- Introducing different heuristic algorithms
- Parameter optimization of a dead time system with heuristic algorithms

Keywords:

- Cuckoo search
- Pso
- Firefly
- Relay feedback
- Modelling

Article Info:

Research Article
Received: 20.07.2017
Accepted: 06.03.2018

DOI:

10.17341/gazimmfd.416507

Acknowledgement:

Correspondence:

Author: Murat Erhan Çimen
e-mail:
muratcimen@sakarya.edu.tr
phone: 0 264 616 09 46

Graphical/Tabular Abstract

In this study, the parameters of a non-minimum phase second-order system with time delay, which is accepted as gray box, are found out using dual channel relay test and Cuckoo Search, Particle Swarm Optimization, Firefly Algorithms. For this purpose, gray box was implemented as two channel test systems to obtain signals belong to system inputs and outputs. After that, these signals are used to identify system parameter according to a performance criteria, which is integral absolute error between difference of real system and model outputs that was tried to be minimized by Cuckoo search, Particle Swarm Optimization and Firefly Algorithms. General system block diagram and its input and output are given at Figure A.

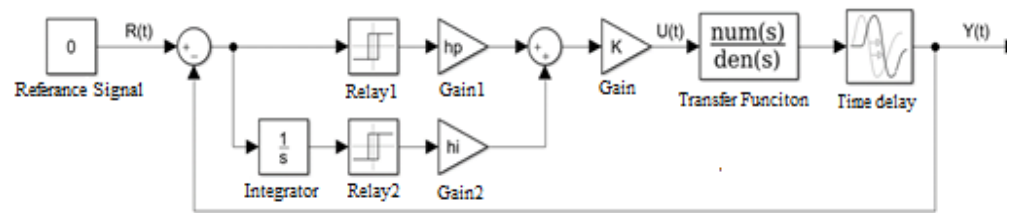


Figure A. Two Channel System Block Diagram

Purpose:

Cuckoo Search, Particle Swarm Optimization and Firefly Algorithms are heuristic algorithms which are inspired from nature itself. Also these algorithm are capable of to solve benchmark problems. In this study a non-minimum phase second order system with time delay is tried to be identified by using Cuckoo Search, Particle Swarm Optimization and Firefly algorithms according to two channel relay test and performances of these algorithms are compared with each other for this problem.

Theory and Methods:

Firstly, a real system is implemented into the two channel relay test. This test allows the system to enter a graded oscillation in a limited range. Thus, different dynamics of the system can be stimulated. After that the model parameters are identified by the same test using Cuckoo Search, Particle Swarm Optimization and Firefly Algorithms. These algorithms were tried to minimize or maximize a performance criterion. In this problem, this criterion is integral absolute error between the model output and the system output that is tried to be minimized to identify the system parameters.

Results:

It is found that the obtained results using Cuckoo Search, Particle Swarm Optimization and Firefly Algorithms gave closer results to the real system than the results obtained with the Genetic Algorithm cited in the literature.

Conclusion:

In this study, a non-minimum phase second order system with time delay was tried to be identified using cuckoo search, particle swarm optimization and firefly algorithms with two channels relay test. Each algorithms run 10 times and their standard deviations and expected values were calculated. As a result, it is shown that these algorithms gave better performances than the performances cited in the literature.



İkinci dereceden ölü zamanlı ve geri tepmeli sistem parametrelerinin, röle testi ve PSO, CS, FA algoritmaları ile belirlenmesi

Murat Erhan Çimen*^{ID}, Ali Fuat Boz^{ID}

Sakarya Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Esentepe, Sakarya, 54187, Türkiye

Ö N E Ç İ K A N L A R

- Sistem parametrelerinin belirlenmesi için geri beslemeli röle kullanımı
- Farklı sezgisel algoritmaların tanıtılması
- Sezgisel algoritmalar ile ölü zamanlı bir sistemin parametre optimizasyonu

Makale Bilgileri

Araştırma Makalesi
Geliş: 20.07.2017
Kabul: 06.03.2018

DOI:

10.17341/gazimmfd.416507

Anahtar Kelimeler:

Cuckoo search,
pso,
firefly,
röle geri-besleme,
sistem modelleme

ÖZET

Bu çalışmada, gri kutu olarak kabul edilen ikinci dereceden ölü zamanlı ve geri tepmeli bir sistem parametreleri, çift kanallı röle testi ile birlikte Guguk Kuşu, Parçacık Sürü Optimizasyonu ve Ateşböceği Algoritmaları kullanılarak belirlenmiştir. Bunun için öncelikle gri kutu, çift kanallı röle testine tabi tutularak sistem giriş ve çıkışlarına ait sinyaller elde edilmiştir. Ardından bu sinyaller arasındaki hata, amaç fonksiyonu olarak hatanın mutlak değerinin integrali fonksiyonunun kullanıldığı Guguk Kuşu, Parçacık Sürü Optimizasyonu ve Ateşböceği Algoritmaları ile minimize edilerek, sisteme ait parametreler belirlenmiştir. Elde edilen sonuçlar analiz edilmiş ve literatürde var olan Genetik Algoritma ile yapılan çalışmalar ile kıyaslanmıştır. Guguk Kuşu, Parçacık Sürü Optimizasyonu ve Ateşböceği Algoritmaları kullanılarak bu çalışmada elde edilen sonuçların, Genetik Algoritma ile elde edilen sonuçlara göre gerçeğe daha yakın olduğu görülmüştür.

Parameter identification of a non-minimum phase second order system with time delay using relay test and PSO, CS, FA algorithms

H I G H L I G H T S

- Use of feedback relays for system identification
- Introducing different heuristic algorithms
- Parameter optimization of a dead time system with heuristic algorithms

Article Info

Research Article
Received: 20.07.2017
Accepted: 06.03.2018

DOI:

10.17341/gazimmfd.416507

Keywords:

Cuckoo search,
pso,
firefly,
relay feedback,
modelling

ABSTRACT

In this study, the parameters of a non-minimum phase second-order system with dead-time, which is accepted as gray box, are found out using dual channel relay test and Cuckoo Search, Particle Swarm Optimization, Firefly Algorithms. For this purpose, gray box was implemented as two channel test systems to obtain signals belong to system inputs and outputs. Then, the error value between the input and output signals was minimized according to the determined objective function and then system parameters are determined using Cuckoo Search, Particle Swarm Optimization and Firefly Algorithms. The obtained results are analyzed and compared with the Genetic Algorithm based studies, which are cited in the literature. It is found that the obtained results using Cuckoo Search, Particle Swarm Optimization and Firefly Algorithms gave closer results to the real system than the results obtained with the Genetic Algorithm.

*Sorumlu Yazar/Corresponding Author: muratcimen@sakarya.edu.tr, afboz@sakarya.edu.tr / Tel: +90 264 295 6472

1. GİRİŞ (INTRODUCTION)

Röle ile yapılan geri besleme testi, sistem tanımda uzun süredir kullanılmaktadır. Bu test aslında sistem tanıma da hiç bilinmeyen siyah kutu yerine, sistemin belli başlı özelliklerinin veya parametrelerinin bulunduğu gri kutu olarak adlandırılan ve sistem parametrelerinin bulunmasını sağlayan bir testtir. Temeli ise Ziegler-Nichols'un kullanmış olduğu kapalı döngüdeki birinci dereceden ölü zamanlı bir sistem kazancının artırılarak osilasyona sokulması ve ardından bu noktadaki sisteme ait kritik kazanç ve kritik frekans değerlerinin bulunarak, kontrolör tasarlanmasına dayanmaktadır [1]. Röle testinde ise kritik kazanç ve frekans değerleri, röle ile birlikte kapalı döngüye girdirilen sistemin osilasyona sokulması sayesinde bulunmaktadır. Bu röle testinde, sistem kutuplarının sanal eksene gelmesi sağlanmakta ve sınırlandırılmış röle çıkışı ile çok yüksek kontrol sinyali ihtimali giderilmektedir. Böylece sistemin kararsızlığa gitme ihtimalide ortadan kaldırılmaktadır. Geri beslemeli röle testi, Tsytkin'in önerdiği ve Aström-Hagglund'un kritik kazanç ve fazla göre kontrolör tasarlanmasıyla geliştirilen bir tekniktir [2-4]. Bu süreç, nonlinear sistemlere ise ilk olarak 1987 yılında Luyben tarafından damıtma sistemleri örneğinde uygulanmış ve bu sistem tanımlama işlemine otomatik ayarlanabilme (Autotune Variation ATV) metodu ismi verilmiştir [5-6]. Shih-Haur vd. da asimetrik röle kullanarak sistem tanıma işlemi yapmışlardır [7]. Wei vd. yapmış oldukları çalışmada röle testini ikiye bölmüşler; ilkinde normal röle testi, ikincisinde ise röleye ölü zaman ekleyerek yani sistemin fazını kaydırarak, farklı durumlar için veriler toplamışlar ve en küçük kareler yöntemi ile önerilen modelin parametrelerini bulmuşlardır [8]. Kaya vd. sistem tanımda asimetrik röle kullanmakla birlikte, sistem parametrelerini A-function metodu ile bulmuşlardır [9]. Friman vd. ise biri normal diğeri integratör içeren röleleri birbirine paralel bağlamışlar ve sistemin faz ve kazancını Nyquist Eğrisi üzerinde kaydırarak, kayan noktaya göre kontrolör tasarımı önermişlerdir [10]. Kristian vd. ise çift röleli test kullanarak birinci derece ölü zamanlı sistemi (FOPDT) tanımak için bir amaç fonksiyonu tanımlamış ve ardından bu amaç fonksiyonunu optimizasyon tekniği ile minimum yapacak şekilde sistem parametrelerini belirlemişlerdir [11]. Josefin vd. yapmış oldukları çalışmada, sisteme göre adaptif olarak ayarlanabilen bir asimetrik röle ile birden fazla test yapmışlardır. Ardından, yaptıkları bu test sonuçlarını kullanarak, sisteme göre ayarlanmış rölenin genlik oranına ve zaman sabitine bağlı olarak değişen bir grafik elde etmişler ve bu grafiğe göre bir modelin önerilebileceğini ve parametrelerinin bulunabileceğini göstermişlerdir [12]. Kaya vd. ise Genetik Algoritma (Genetic Algorithm-GA) ile ters tepmeli sistemlerin tanımlanmasında, farklı zaman aralıkları için sistem tanıma yapmış ve sonuçlarını karşılaştırmışlardır [13].

Doğadan esinlenilerek geliştirilen problem çözme algoritmalarına sezgisel algoritmalar denilmektedir. Bu anlamda yapılan ilk çalışmaya, 2. Dünya savaşı sırasında

şifreleri kırmak için kullanılan ve 1942 yılında Alan Turing'in geliştirdiği Turing makinesi örnek olarak verilebilir. Günümüzde ise geliştirilen bu algoritmalara Genetik algoritma, Benzetilmiş Tavlama (Simulated Annealing-SA), Karınca Kolonisi Algoritması (Ant Colony Optimization-ACO), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization-PSO), Diferansiyel Evrim (Differential Evolution-DE), Uyum Araması (Harmony Search-HS), Yapay Arı Kolonisi (Artificial Bee Colony-ABC), Ateşböceği Algoritması (Firefly Algorithm-FA) ve Guguk Kuşu Algoritması (Cuckoo Search-CS) örnek olarak verilebilir. Bu algoritmaların birbirlerine göre üstünlükleri olduğu gibi dezavantajları da vardır. Bu üstünlükler ve dezavantajlar, uygulanan probleme göre değişmektedir. Çözülmesi istenen probleme uygun optimizasyon tekniği seçildiği takdirde, problemin çözülmesi daha hızlı ve daha hassas olacaktır ve en önemlisi daha doğru sonuçlar üretecektir [14].

Bu alanda yapılan ve literatürde geçen çalışmalar ise kısaca şöyle özetlenebilir: PSO algoritması zor problemlerin çözülmesi için hatta nonlinear denklemlerin çözülmesi için önerilmiştir [15,16]. Özsağlam, PSO, DE ve GA'yı karşılaştırarak, PSO'nun literatürdeki bazı benchmark problemlerini daha iyi çözdüğünü ifade etmiştir [17]. Bu algoritmayı Tozan vd. de sensör yerleştirme probleminin çözümü için kullanmışlardır [18]. Ayrıca Öztürk vd. PSO algoritmasını, DC-DC inverter için kontrolör tasarımı için kullanmışlar ve Ziegler-Nichols yöntemine göre daha iyi sonuç verdiğini gözlemlemişlerdir [19]. Aydemir ise çalışmasında, hesaplamalı elektromanyetik uygulamalarında uygun çözümler bulabilmek için GA ve PSO algoritmalarını kullanmıştır [20]. Bununla birlikte Gao vd. yapmış oldukları FPGA tabanlı IIR (Sonsuz Darbe Cevaplı) tasarımı PSO algoritmasını kullanmışlar ve çok başarılı bir donanım gerçekleştirilmesi elde ettiklerini ifade etmişlerdir [21]. Öztürk vd. PSO ve ABC algoritmalarını veri kümeleme problemlerine uyarlamışlar ve başarılarını karşılaştırmışlardır [22]. Tavakoli vd. ölü zamanlı sistemler için robust PI tasarımı PSO ile yapmışlardır [23]. Satpati vd. ise yine PSO ile FOPDT bir sistem için robust PID kontrolör tasarımı yapmışlardır [24]. Kanthaswamy vd. yapmış oldukları çalışmada, PSO algoritmasını hibrit bir şekilde kullanarak, PID kontrolör tasarımı yapmışlardır [25]. Ahuja vd. yapmış oldukları çalışmada, FOPDT için 2 Serbestlik Dereceli PID (2DOF)'nin parametrelerini PSO ile optimize etmişler ve bozucuya karşı dayanıklı olduğunu belirtmişlerdir [26]. Shi vd. yapmış oldukları çalışmada, CS'yi yapı problemlerinin çözümünde kullanmışlardır [27]. Elazim vd. ise yapmış oldukları çalışmada, güç sistem düzenleyicilerinin dizaynında CS kullanmışlardır [28]. Nema vd. yapmış oldukları çalışmada, CS ile sistem tanıma ve PI-PD denetleyici tasarımı yapmışlardır [29]. Jin vd. verilen sistem için PID tasarımı CS, geliştirilmiş CS ve Ziegler-Nichols metodlarını kullanmışlardır [30]. Roeva vd. glikoz konsantrasyonunun modellenmesi için nonlinear model kestirici ve geliştirilmiş Kalman filtresi kullanmışlar ve denetlenmesi için PID kontrolör tasarımı FA

algoritması ile optimize etmişlerdir [31]. Bu çalışmada, röle tekniği ile sezgisel algoritmalar birlikte kullanılarak, ölü zamanlı bir sistemin model parametreleri belirlenmiştir. Endüstride ölü zamanlı sistemlere sık rastlanmaktadır. Yine literatürde bu sistemlerin tanınmasına ve denetlenmesine yönelik pek çok çalışma bulunmaktadır. Diğer taraftan ölü zamanlı bu sistemlerin bazıları non-minimum faza ve dolayısı ile ters cevaplı bir karakteristiğe sahip olabilmektedir. Bu ise sistem parametre sayısının artmasına neden olmaktadır. Diğer taraftan, parametre sayısının fazla olduğu bu tür sistemlerin parametre kestiriminde, literatürde bilinen bazı tekniklerin kullanılması problemler oluşturabilmektedir. Örneğin röle tekniği (otomatik ayar), sistemin sadece kritik kazanç ve kritik frekansına bağlı bir bilgi sağlamaktadır, yani doğrudan parametreleri hakkında bilgi verememektedir. Bu durumda, parametrelerin belirlenebilmesi için röle testine ilaveten farklı tekniklerin de kullanılması gerekmektedir. Bu çalışmada, ters cevaplı bir sistem parametrelerinin belirlenmesinde sezgisel algoritmalar olan PSO, FA ve CS algoritmaları kullanılmıştır. Sürü tabanlı olan bu algoritmalara bakıldığında, tamamının global olarak optimum noktayı iteratif çözümlene yöntemi ile bulduğu görülebilir. Diğer taraftan, iterasyon içerisinde her algoritmanın yaptığı işlem değişebilmektedir. Bunun nedeni esinlenen doğa olayının, matematiksel olarak o şekilde ifade edilmesinden kaynaklanmaktadır.

Dolayısı ile bir algoritmayı diğer algoritmalarından daha üstün kılanda, kendi içerisindeki bu modellenme şeklidir. Eğer esinlenen doğa olayı en iyi şekilde matematiksel olarak temsil edilip, algoritma için doğru kontrol parametreleri belirlenebilirse, bu aslında en iyi algoritma olmaya adaydır denilebilir. İstenen şey ise, en az iterasyon ile global noktanın bulunmasıdır. Bunun için algoritmaların başarıları, literatürde bulunan bir takım test problemlerinin çözdürülmesi ile kıyaslanmaktadır. Genel olarak bu algoritmalar rastlantısallık içerdikleri için, yaklaşım hızları ve doğru sonucu elde etmeleri problemden probleme de değişebilmektedir. Bu nedenle ölü zamanlı ve ters cevaplı sistem parametrelerinin belirlendiği bu çalışmada, daha önceden yapılmış bir çalışma olan GA ve tek röle testine karşılık, sistemi asıl çalışacağı bölgede osilasyona sokan çift kanallı röle, parametrelerin belirlenmesi için ise PSO, CS ve FA algoritmaları tercih edilmiştir. Bu algoritmaların seçilme nedeni, literatürde ölü zamanlı ters cevaplı sistem parametre optimizasyonunda çift kanallı röle testi ile birlikte daha önceden kullanılmamalarıdır. Bu işlem yapılırken sistem ve model offline olarak teste tabi tutulmakta ve modelin parametreleri belirlenen algoritmaya göre belirlenmektedir.

2. RÖLE TESTİ (RELAY TEST)

Röle testi, matematiksel modeli bilinmeyen bir sistemin röle ile osilasyona sokularak, bu kritik noktadaki frekans ve genliğinin bulunmasını sağlayan bir tekniktir. Bu teknik kullanılarak denetleyici tasarımı yapılabildiği için aynı zamanda otomatik ayarda (autotuning) denilmektedir. Röle tekniğinin en önemli avantajları; (i) Herhangi bir önbilgiye gerek duyulmadan sistem kritik kazanç ve frekansını bulabilmesi, (ii) Sistemi belirlenen sınırlar dışına sürüklediği için kararsızlığa gitme ihtimalini sıfırlaması, (iii) Çok hassas bir şekilde 180 derecede frekans bilgisine ulaşabilmesi olarak tanımlanabilir [5]. Diğer taraftan tek bir röle kullanılarak yapılan test sonucunda, bazı sistemlerin tüm parametreleri bulunamamaktadır. Bu durumda, sisteme farklı türde eklentiler yapılarak, sistem dinamiklerini daha iyi belirlemeye yönelik çalışmalarda yapılmaktadır [8].

2.1. Tekli Röle Testi (Single Relay Test)

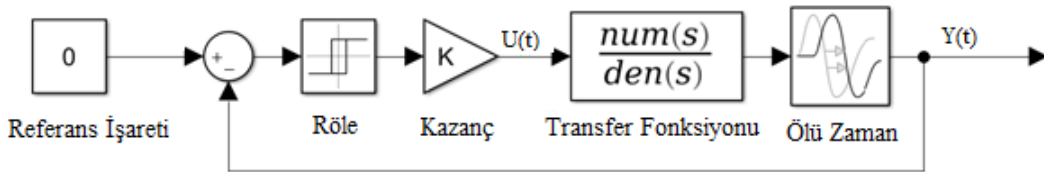
Tekli röle testinde Şekil 1'de görüldüğü gibi sistemin önüne seri bir röle bağlanmakta ve ardından sistem istenilen sınırlar içinde osilasyona sokularak, Şekil 2'de örnek olarak verilen kritik kazanç ve kritik frekans değerleri bulunabilmektedir. Sistemin kritik kazanç ve frekans değerinin hesaplanmasını sağlayan denklemler sırası ile Eş. 1 ve Eş. 2'de verilmiştir [10]. Burada h rölenin genliğini ve u 'da sistemin çıkış genliğini ifade etmektedir. Kritik frekans veren Eş. 2'de kullanılan T ise, sistem çıkışının periyodudur. Aslında bu eşitlikler, osilasyona giden cevabın birinci harmoniğini bulmayı sağlamaktadır. Yani cevap ne kadar sinüs sinyaline yakınsa, bozulma o derece az olacaktır ve sonuçlar o kadar doğru olacaktır [32].

$$K_c = \frac{4h}{u\pi} \quad (1)$$

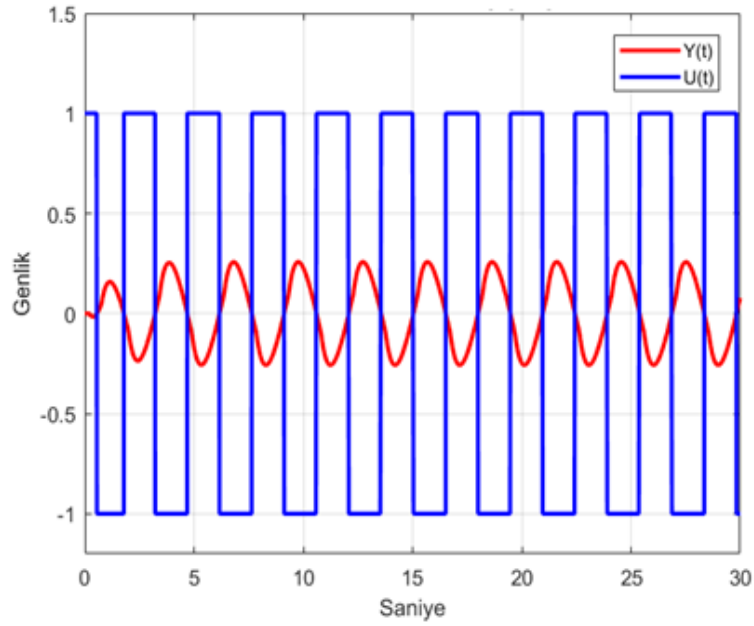
$$\omega_c = \frac{2\pi}{T} \quad (2)$$

2.2. Çift Röle Testi (Dual Relay Test)

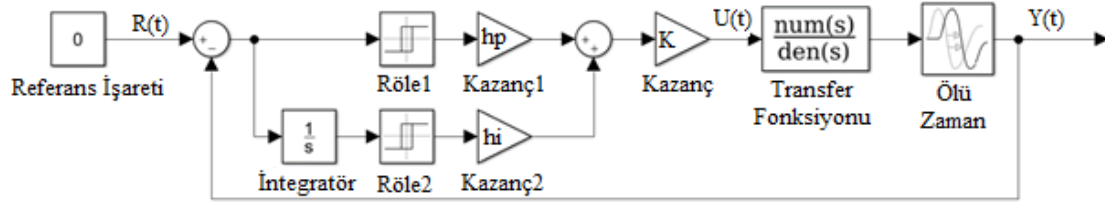
Çift röle testi Şekil 3'de gösterildiği gibi, birisine seri integratör bağlı iki paralel rölenin sisteme seri olarak bağlanması ile elde edilmektedir. Bu teknikte, tanımlama noktasının fazı 180 dereceden farklı olarak, sistemin asıl çalışması istenilen yerine göre belirlenmekte ve bu noktadaki bilgiler ile de sistem parametreleri bulunmaktadır [10]. Bu sayede, sistem dinamiklerine uygun bir noktaya göre kontrolör tasarımı yapılabilmektedir. Şekil 4'de giriş ve çıkış işaretleri görülmektedir.



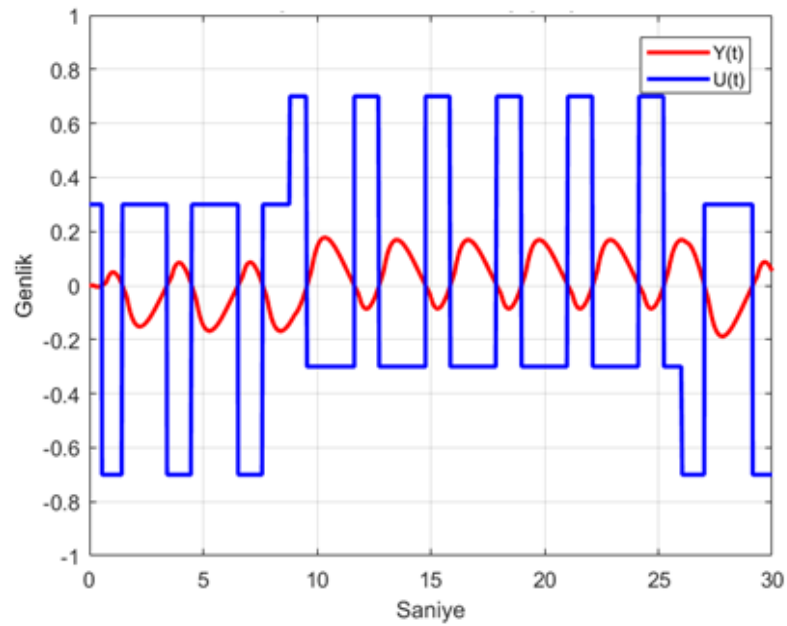
Şekil 1. Tekli Röle Testi(Single Relay Test)



Şekil 2. Tek Röle Testinde Sistem Giriş ve Çıkışı (System Input and Output in Single Relay Test)



Şekil 3. Çift Röle Testi (Dual Relay Test)



Şekil 4. Çift Röle Testinde Sistem Giriş ve Çıkışı (System Input and Output in Dual Relay Test)

Sistem tanıma deneyinde genel olarak giriş işaretinin çeşitliliği, sistemin farklı dinamiklerinin de tanınmasını sağlamaktadır [33]. Bu nedenle genelde tanıma işlemlerinde PRBS (Pseudo rastlantısal binary işareti) gibi genliği +1 ile -1 arasında ve rassal olarak değişen sinyaller kullanılmaktadır. Bu çalışmada ise uygulanan işaret, sistemin çıkışına bağlıdır. Buradan elde edilen işaretin genliğine göre röleler aktif olup, sisteme giriş olarak uygulanmaktadır. Sistemin kritik kazancı, Şekil 3'deki röle 1'in kazancı (h_p), röle 2'nin kazancı (h_i) ve sistem çıkışından elde edilen u genliğinin Eş. 3'de yerine konulması ile bulunabilir. Ayrıca sistemin kritik frekansı ise, Eş. 4'de verilmiştir. Kritik frekans sistem cevabının periyoduna yani T 'ye bağlıdır [11].

$$K_c = \frac{4\sqrt{h_p^2 + h_i^2}}{\pi a}; \quad (3)$$

$$\omega_c = \frac{2\pi}{T}; \quad (4)$$

$$\varphi = -\arctan\left(\frac{h_i}{h_p}\right); \quad (5)$$

3. ÖLÜ ZAMANLI TERS CEVAPLI SİSTEMLER (TIME DELAY SYSTEM WITH INVERSE RESPONSE)

Sıfırlardan ve kutuplardan oluşan ölü zamanlı (zaman gecikmeli) sistemlerin genel ifadesi Eş. 6'daki gibi verilebilir. Burada verilen sistemin kapalı döngü transfer fonksiyonuna ait kutuplar, sistem cevabının karakteristiğini belirlerken, sıfırlar ise sistemin başlangıç durumu ve hızı üzerinde etki etmektedir. Transfer fonksiyonuna ait herhangi bir sıfırın sağ yarı düzlemde olması, sistem cevabının negatif olma ihtimalini doğurmaktadır. Böyle bir sistemde çıkış sinyali, sisteme uygulanan girişe karşılık ters bir tepki verdikten belli bir süre sonra giriş işareti yönünde tepki vermeye başlayacaktır [34]. Bu tür sistemlere ters cevaplı sistem denilmektedir. Sistemde ilave olarak zaman gecikmesi olması durumunda ise, ölü zamanlı ters cevaplı sistem olarak adlandırılmaktadır.

$$G(s) = \frac{(s + a_m)(s + a_{m-1}) \dots (s + a_0)}{(s + b_n)(s + b_{n-1}) \dots (s + b_0)} e^{-s} \quad m < n \quad (6)$$

4. OPTİMİZASYON ALGORİTMALARI (OPTIMIZATION ALGORITHMS)

Optimizasyon, verilen kısıtlar içerisinde en iyiyi elde etme sanatıdır. Bunun için klasik manada türev tabanlı çalışan dik iniş, Newton, Quaksi-Newton, Conjugate-Newton gibi arama algoritmaları kullanılabilir. Bu tip algoritmalar pek çok lokal minimumu bulunan bir problemde, belirlenen başlangıç değerine göre amaç fonksiyonunu minimuma yönlendirecek şekilde arama yapmaktadır. Bunların yanı sıra doğadaki canlı ve cansız varlıkların davranışları modellenerek, sürü tabanlı olarak çalışan sezgisel algoritmalar geliştirilmiştir. Bu algoritmalar daha çok global arama yapmaya çalışarak, en optimum noktayı yakalamaya

çalışmaktadırlar. Temelinde ilk olarak, sürü içerisindeki her bireye olası çözüm değeri ataması yapılmaktadır. Ardından her bir bireyin olası çözümü, amaç fonksiyonu ile değerlendirilir. Akabinde algoritmanın yapısına göre sürü içerisindeki bireyler hareket ettirilir ve her bir bireyin yeni olası çözümleri, amaç fonksiyonu ile yeniden değerlendirilir. Sonra algoritmanın yapısına bağlı olarak eski ve yeni olası çözümlere göre ve/veya bunların amaç fonksiyonundaki değerlerine göre eniyileme yapılarak, sürünün olası çözümleri güncellenir. Bu şekilde optimum sonuç elde edilmeye çalışılmaktadır.

Doğanın yapısına bakıldığında, israfın en azının kendisinde olduğu görülmektedir (örneğin bir sıvının denge noktasına en az enerji harcayarak gelmesi gibi). Dolayısı ile bu tip davranışlar matematiksel formata dönüştürüldüğünde ve uygun kontrol parametreleri kullanıldığında, yeni bir sezgisel algoritma oluşturulabilmektedir. Bilim insanları da doğayı inceleyerek, en iyi optimizasyon algoritmasını aramaya devam etmektedirler. İlk olarak GA gibi insan genomunun çaprazlanması ilkesine dayanan ve sonrasında farklı davranışlar eklenen GA algoritması, doğal seçimdeki gibi en iyiyi bir sonraki nesle aktarması sağlanmıştır. Bu sayede en dayanıklı olan birey, bir sonraki nesle genlerini aktaracaktır. Bu çalışmada ise sezgisel algoritmalarından olup kullanımını basit, anlaşılması kolay ayrıca literatürde en yaygın ve/veya güncel olan PSO, CS ve FA algoritmaları seçilmiştir. CS algoritmasının tercih edilmesinin nedeni, iki aşamadan oluşmasıdır. Lokal ve global aramanın dengelenerek gerçekleştirildiği bu algoritmanın ilk aşamasında global arama, ikinci aşamasında ise lokal arama yapılmaktadır. Bu özelliğinden dolayı seçilen CS algoritması, problemde bulunan bir çok lokal çözüm içerisinde en uygun global çözümü diğer algoritmalarla göre daha iyi bulabilmektedir. Dolayısı ile bu özelliği nedeniyle, CS algoritması global yakınsamayı garanti etmektedir [35]. Böylece problem için daha doğru ve hassas sonuç bulunabilmektedir. Ayrıca global aramada standart rastgele yürüyüşten ziyade, beklenen değeri ve varyansı sonsuz olan Levy Flight bozucusu kullanılmaktadır. Böyle olduğu için global arama yaparken Levy Flight bozucusunun kullanıldığı algoritmalar, Gauss bozucusunun kullanıldığı algoritmalarından daha efektif olabilmektedir [35]. FA algoritmasının tercih edilmesinin nedeni, parametrik olarak PSO'ya benzemesinden dolayıdır. Yapısı ve uygulanması bakımından FA algoritması, ivmelendirilmiş PSO olarak da görülebilmektedir. Ayrıca FA algoritması içermiş olduğu Gauss bozucusu sayesinde genel olarak problemlerde PSO'dan daha iyi performans sergilemektedir [14].

4.1. Guguk Kuşu Algoritması (Cuckoo Search Algorithm)

CS Algoritması, guguk kuşlarının yumurtalarını başka yuvalara bırakarak, kendi yumurtalarını başka guguk kuşlarının büyümesini sağlamalarından esinlenilerek, Xin-She ve Suash-Deb tarafından geliştirilmiştir. Aslında guguk kuşlarının bu içgüdüsel davranışlarına kuluçka parazitliği (brood parasitism) denilmektedir [36]. Guguk kuşlarının olası durum içeren davranışlarına bakıldığında, guguk

kuşları başka yuvalara uçarak yumurtladıkları yuvalarda, ev sahibi yumurtalardan yavru guguk kuşu çıkma olasılığı yüksek olan yumurtaları yuvadan atmaktadırlar. Ayrıca ev sahibi guguk kuşu yuvaya geri geldiğinde yabancı yumurtayı tanırsa, o zaman ev sahibinin iki seçeneği bulunmaktadır; ya yabancı yumurtayı aşağı atmakta, ya da yuvayı terk etmektedir. Diğer taraftan, ev sahibi yabancı yumurtayı tanımaz ise yabancı yumurtaların da üzerine oturarak kuluçkaya yatmakta ve kuluçka asalaklığına maruz kalmaktadır. Guguk kuşlarının bu davranışları kurallaştırılırsa [14, 36, 37];

- Bir guguk kuşu rastgele bir yuvaya sadece bir tane yumurta bırakabilir.
- Eğer yuvada kaliteli yumurta varsa, o zaman kaliteli yumurta bir sonraki jenerasyona aktarılmaktadır.
- Ev sahibi guguk kuşu, bırakılan yumurtayı belirli bir p_a (0,1) olasılıkla tanıyabilmektedir. Eğer ev sahibi yabancı yumurtayı tanırsa, o zaman ev sahibinin 2 seçeneği bulunmaktadır; ya yabancı yumurtayı aşağı atacak ya da kendine yeni yuva kurmak üzere yuvayı terk edecektir.

CS algoritması, küresel rassal yürüyüş ve bölgesel rassal yürüyüş yaklaşımlarının dengelenerek kullanıldığı bir algoritmadır. Bu tip dengelemelerde kontrol parametreleri bulunmaktadır ve bu parametreler ayarlanmaktadır. CS algoritması için küresel rassal yürüyüş, Xin-She ve Suash Deb'in önerdiği Levy Flight ile gerçekleştirilmektedir [36]. Levy Flight sayesinde üretilen değer, α gibi bir değişkenle ağırlıklandırılıp eski konum ile toplanarak, yeni konumların bulunması sağlanmaktadır. Eş. 7'de verilen α parametresi 1 olarak alınmıştır. Ayrıca Levy Flight içinde Xin-She'nin önermiş olduğu kontrol parametreleri s ve λ 'dır. Eş. 7'de verilen ve küresel rassal yürüyüşü sağlayan Levy Flight'ın λ parametresi, genellikle 1 ile 3 arasında alınmaktadır. Bu çalışmada ise Eş. 4'deki λ , 1,5 olarak alınmıştır. Bu çalışmada s ise 0,8862 olarak alınmıştır.

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \quad (7)$$

Eş. 8'de, yerel rassal yürüyüş ile aramayı sağlayan eşitlik verilmiştir. Bu denklemdeki x_k^t ve x_k^t , çözüm uzayındaki vektörlerinin rassal permütasyonlarıdır. $H(u)$, Heaviside adım fonksiyonu ve α parametresi ise Gauss dağılımdan gelen rassal reel bir sayıdır. s ise adım uzunluğudur [37]. Burada s parametresini belirlemek önem arz etmektedir. Çünkü s büyük olduğunda, istenilen nokta yakınsanamayabilmektedir. Küçük olduğunda ise yakınsama çok fazla zaman almaktadır.

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \quad (8)$$

Algoritmanın adımları Şekil 5'de verilmiştir.

4.2. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)

PSO, Kennedy ve Eberhart tarafından sürülerin davranışlarından geliştirilmiş bir algoritmadır [15, 16, 18].

Örneğin balık sürüsündeki her bir balığın çevresine adapte olması, yiyecek kaynağı bulabilmesi veya daha büyük bir balık saldırısı olduğunda avcılardan kaçabilmesi gibi davranışlarından esinlenilerek geliştirilmiş bir algoritmadır. PSO, popülasyon temellidir ve sürü içerisinde bilgi paylaşımına yöneliktir. Sürüdeki her bir birey, parçacık olarak adlandırılmaktadır. Her bir parçacığın, pozisyon ve hız bilgileri vardır. Her bir iterasyonda bu hız ve pozisyon bilgileri güncellenmektedir. Bu güncellenme, parçacığın o anki hız bilgisine, konumuna, sürünün en iyi pozisyonuna (g_{best}) ve o iterasyon içindeki en iyi pozisyona (p_{best}) göre ayarlanmaktadır. Notasyon olarak parçacıkların pozisyonu p , hızları v olarak tanımlanır ise bir sonraki iterasyon için pozisyon ve hız bilgileri Eş. 9 ve Eş. 10'daki gibi güncellenir [15, 16, 18].

$$v_{t+1} = wv_t + c_1r_1(p_{best} - p_t) + c_2r_2(g_{best} - p_t) \quad (9)$$

$$p_{t+1} = p_t + v_{t+1} \quad (10)$$

Eş. 9'da, r_1 ve r_2 , program sayesinde 0-1 arasında üretilen rastgele sayılardır. c_1 ve c_2 , öğrenme faktörüdür ve genellikle 1,8-2 arasında değer almaktadır. w ise atalet momentidir ve hızın etkisini zamanla azaltmaktadır. Diğer taraftan iterasyon sayısı arttıkça ataletin değeri küçülmektedir. Atalet değerinin çok küçülmesi de parçacıkların hareket edememesine neden olmaktadır [18,38]. PSO algoritması Şekil 6'da verilmiştir.

4.3. Ateşböceği Algoritması (Firefly Algorithm)

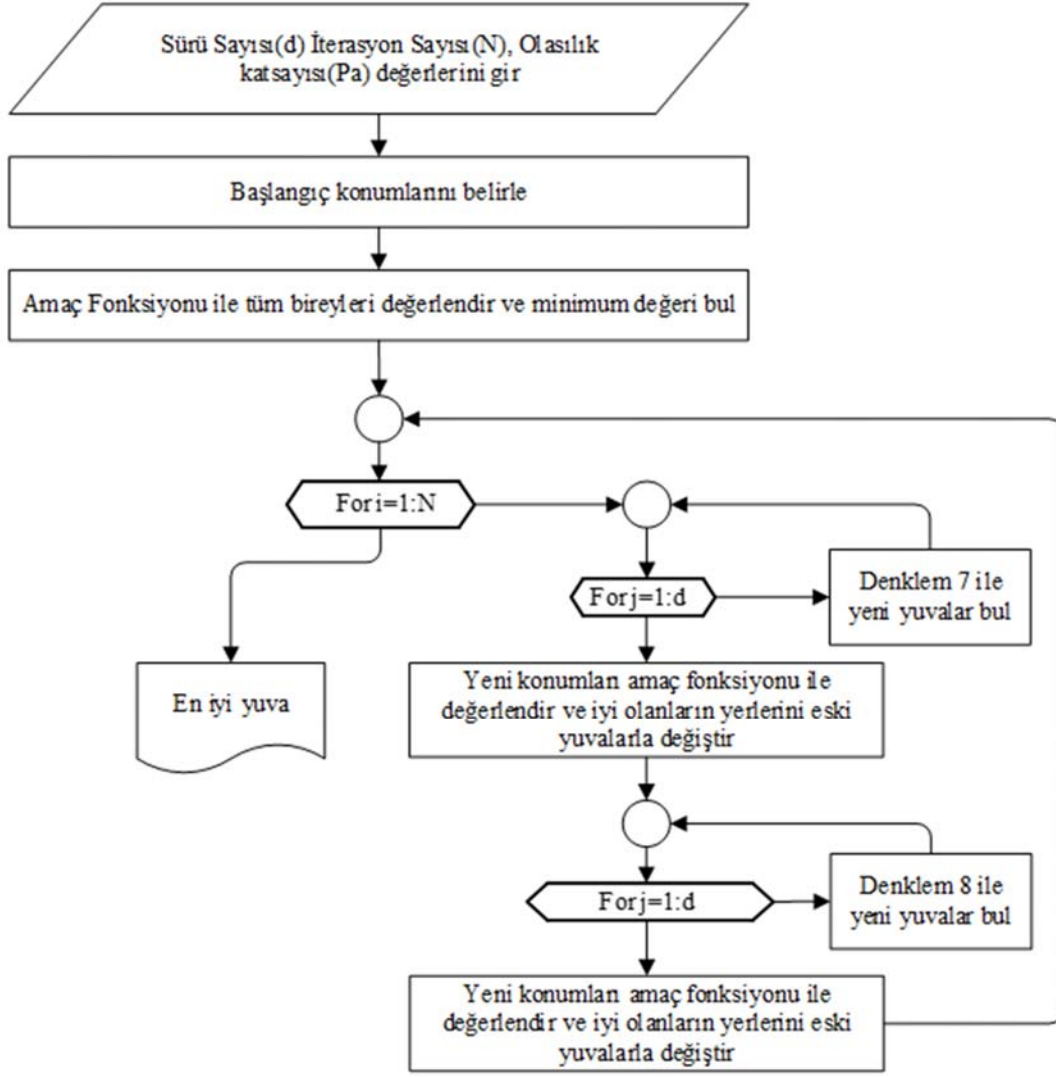
FA, Xin-She Yang tarafından 2008 yılında ateşböceklerinin davranışlarından esinlenilerek önerilmiş bir algoritmadır. Bu algoritmada ateş böceklerinin parlaklık ve hareket yönleri kontrol parametreleri olarak modellenip geliştirilmiştir. Bu davranışlar kurallaştırıldığında aşağıdaki sonuçlar ortaya çıkar.

- Ateş böcekleri cinsiyetsiz varsayımı yapılır ve bu ayrım yapılmaksızın birbirlerinden etkilenebilirler.
- Ateş böceklerinin parlaklığı çekiciliği ile orantılıdır. En parlak olan ateş böceği diğerlerini kendine doğru çeker.
- Ateş böceğinin parlaklığı amaç fonksiyonundan elde edilecek olan değere göre belirlenmektedir.

Ateş böceklerinin parlaklığı ışık yoğunluğu olarak da adlandırılmaktadır. Işık yoğunluğu (I); başlangıç ışık yoğunluğuna (I_0), gammaya (γ) ve uzaklığa (r) bağlıdır. Işık yoğunluğunun denklemi Eş. 11'de verilmiştir.

$$I = I_0 e^{-\gamma r} \quad (11)$$

Ateş böceklerinin birbirlerine olan çekiciliği (β), uzaklık ile artıp azalmaktadır. Yakın olan ateş böcekleri diğer ateş böceklerine daha çekici gelmekte ve uzaktaki ateşböceklerinin çekiciliği azalmaktadır. Dolayısı ile çekicilik Eş. 12'deki gibi hesaplanabildiği gibi, Eş. 13 kullanılarak hesaplanabilir. Seçilen bu denklemdeki başlangıç değeri (β_0), aslında iki ateşböceğinin arasındaki uzaklığın sıfır olduğu andaki değeri göstermektedir [39].



Şekil 5. Guguk Kuşu Algoritması(Cuckoo Search Algorithm) [36]

$$\beta = \beta_0 e^{-\gamma r^2} \quad (12)$$

$$\beta = \frac{\beta_0}{1 + \gamma r^2} \quad (13)$$

Ateş böceklerinin birbirine olan uzaklıkları, hem ışık yoğunluğunu hemde çekiciliği değiştirmektedir. Bu değişimler de ateş böceklerinin hareketlerini belirlemektedir. Dolayısı ile ateş böceklerinin birbirleri arasındaki uzaklıkların hesaplanabilmesi için Eş. 14 kullanılmaktadır.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (14)$$

Ateş böceklerinin içgüdüsel davranışlarını modellemek için Eş. 15 kullanılmaktadır. Bu denklemdaki ilk ifade bir önceki konumu ifade ederken, ikinci ifade ateş böceğinin başka ateş böceklerinden etkilenmesi üzerine modellenen çekicilik formülünden gelmektedir. Aslında iki tane ateş böceği

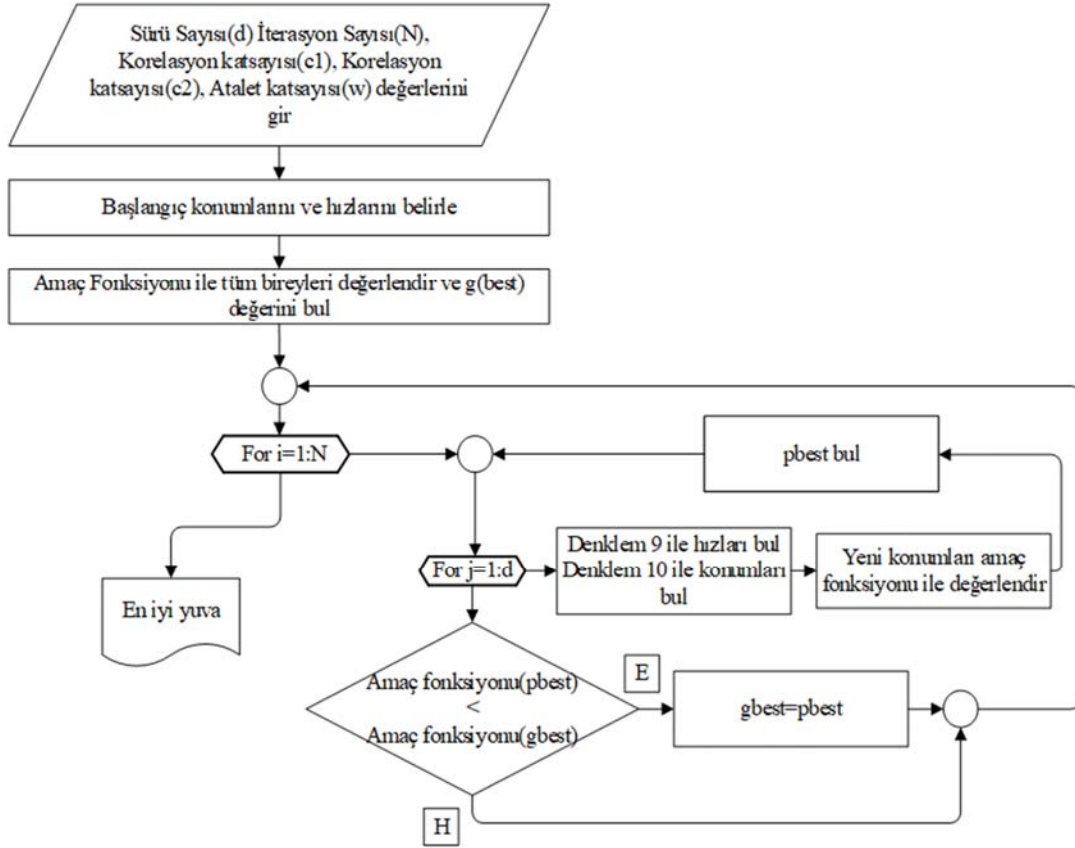
arasındaki uzaklığın ve çekiciliğin çarpılması ile elde edilmektedir. Diğer bir eklenti olan α rastlantı parametresi, ε ise Gaussian dağılımından gelen bir vektörü ifade etmektedir [14, 40].

$$x_i^{t+1} = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \varepsilon_i^t \quad (15)$$

Ateş böceği algoritmasının adımları Şekil 7'de verilmiştir.

5. SİMÜLASYON ÇALIŞMASI VE DEĞERLENDİRİLMESİ (SIMULATION AND EVALUATION)

Bu çalışmada, parametreleri bilinmeyen bir sistemin Şekil 3'deki yapı ile osilasyona girmesi sağlanarak, sistemin osilasyon cevabı ($y(t)$) alınacaktır. Ardından buradan elde edilen bilgiler PSO, CS ve FA algoritmalarında kullanılarak, sistemin geri tepmeli birinci dereceden ölü zamanlı sistem (Inverse response with FOPDT) modeli elde edilecektir. Bu



Şekil 6. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) [18]

algoritmalar her iterasyonda değerlendirme yaparken, kendi ürettikleri parametre değerlerini simülasyonla sisteme uygulayarak bir sonuç üretmektedirler. Ardından bu sonuç amaç fonksiyonuna yerleştirilerek, model cevabı ile sistemin gerçek cevabı arasında bir hata değeri elde edilir. Bir sonraki iterasyonda, algoritmaların bulmaya çalıştığı parametre değerleri bu ilişkilendirilmiş amaç fonksiyon sonucuna göre değiştirilerek, en uygun parametre değerleri elde edilmeye çalışılmaktadır. Bu çalışmada sistem modeli sabit olup, sistem parametreleri yukarıda belirtilen algoritmalarla bulunmaktadır. Bunun için gerçek sistemin kendisi Eş. 16’da verilmiştir.

$$G_{\text{gerçek}}(s) = \frac{(-0.2s+1)}{(s+1)(s+1)} e^{-0.2s} \quad (16)$$

Optimizasyon teknikleri ile parametreleri belirlenmek istenen Eş. 16’daki gerçek sistemin parametrik transfer fonksiyon modeli ise Eş. 17’de verilmiştir. Transfer fonksiyonunun kazancı (K), sistem sıfırı (T_0), kutupları (T_1 , T_2) ve ölü zamanı (L) olarak sembolize edilmiştir. Kısıtlar Eş. 18’de, minimum yapılmak istenen amaç fonksiyonu ise Eş. 19’da verilmiştir. Bu sisteme ait parametreleri bulabilmek için PSO, CS ve FA algoritmaları kullanılmış ve elde edilen sonuçlar birbirleri ile kıyaslanmıştır. Bu çalışmada, optimizasyon teknikleri için 10 farklı analiz yapılmıştır. Bu sonuçlar Intel(R) Core(TM) i7-6700 HQ

CPU @ 2.60Ghz, 64 Bit, 8GB RAM’e sahip bir bilgisayardan elde edilmiştir.

$$G(s) = \frac{K(-T_0s+1)}{(T_1s+1)(T_2s+1)} e^{-Ls} \quad (17)$$

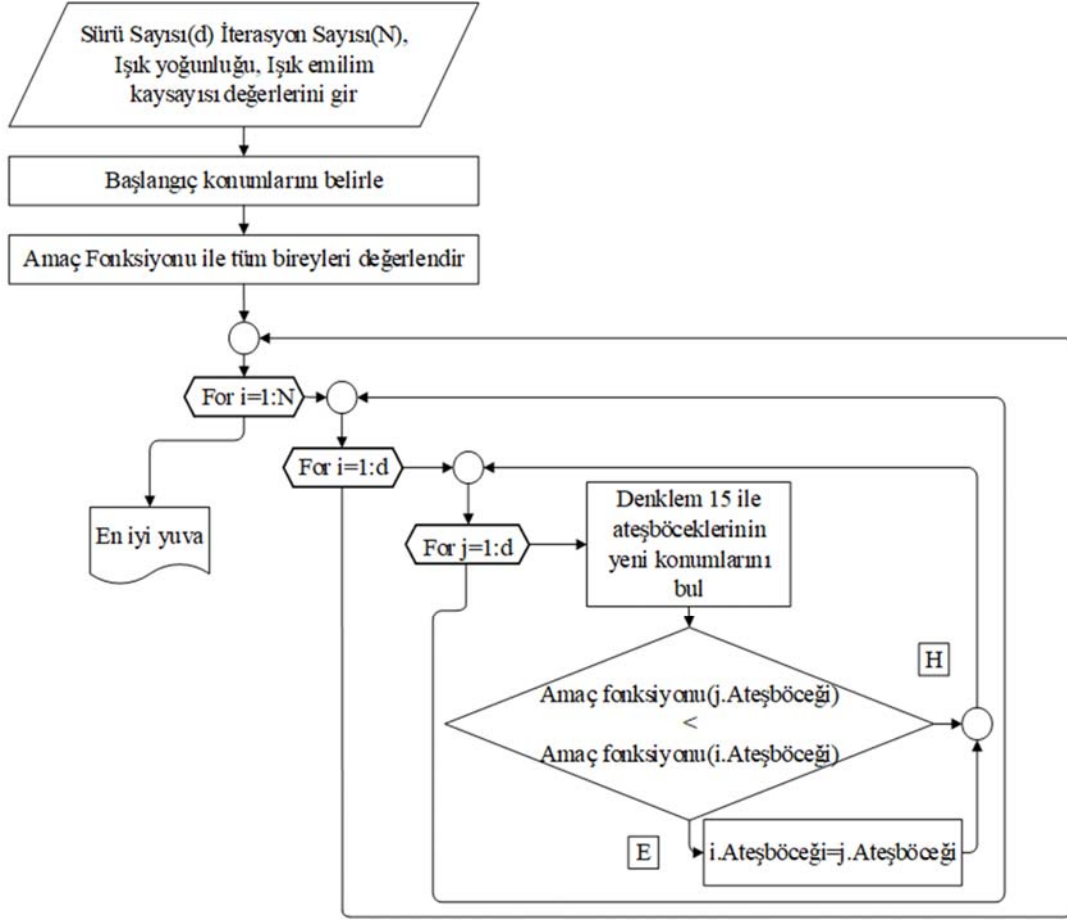
Kısıtlar;

$$0 < K, T_0, T_1, T_2 < 10 \quad (18)$$

Amaç fonksiyonu;

$$IAE = \int_{T_1}^{T_2} |e(t)| dt = \int_{T_1}^{T_2} |y(t) - \psi(t)| dt \quad (19)$$

Eş. 19’daki amaç fonksiyonunda, $y(t)$ gerçek sistemden elde edilen osilasyon cevabı iken, $\psi(t)$ ise algoritmaların her bir adımda üretmiş olduğu değişkenlere bağlı olarak, Şekil 3’de üretilen osilasyon cevabıdır. Algoritmalar bu ikisi arasındaki hatanın mutlak değerinin integralini(Integral Absolute Error-IAE) alarak, IAE indeks değerini vermektedir. Dolayısı ile bir ölçüt olarak algoritmalar, üretmiş oldukları değişkenler arasından amaç ölçütünü minimum yapan değeri vermeye çalışmaktadırlar. Doğası gereği bu algoritmalar, var olan değişkenlerden yeni değişkenler üretmekte ve bu değişkenlerin uygunluğunu Eş. 19 ile değerlendirip, yeni en iyi değeri elde etmeye çalışmaktadırlar. Bu işlem iteratif olarak gerçekleştiği için



Şekil 7. Ateş Böceği Algoritması (FireFly Algorithm) [14]

zaman almaktadır. Diğer taraftan, her algoritma kendi yapısından kaynaklı olarak amaç ölçütünü çok iyi bir şekilde minimize edemeyebilmektedir. Dolayısı ile bu algoritmalar, belirli bir iterasyon sayısına ve amaç ölçütünün belli bir değer altına düşüşüne kadar iterasyon yapmaktadırlar. Yapılan çalışmada Şekil 3’de verilen yapı kullanıldığı için, Röle 1 standart röle olarak kabul edilip, histerezis değeri 0 ve genliği 0,5 olarak seçilmiştir. Önünde integratörü olan Röle 2’nin histerezisi 0,2 ve genliği 0,4 olarak seçilmiştir. Bu ayarlama sadece Eş. 2’deki sistem için ayarlanmış olup, farklı değerlerde de seçilebilirdi. Lakin seçilen değerlerden Röle 2’nin histerezisi önem arz etmektedir, zira farklı sistemler için zaman sabiti farklı olacağından, seçilen histerezis değeri Röle 2’yi pasif yapabilmektedir. CS algoritmasının diğer algoritmalarla eş olabilmesi için popülasyon sayısı 40 ve maksimum iterasyon sayısı 1000 olarak alınmıştır. Adım uzunluğu 0,05, λ parametresi 1,5 ve p_a parametresi ise 0,25 olarak alınmıştır. Sonuçlar Tablo 1’de verilmiştir. CS ve diğer algoritmalar için 3 farklı zaman aralığında sistem cevabına göre tanıma işlemi yapılmıştır. İlk olarak 0 ile 2,5 sn aralığında geçici hal durumu için tanıma işlemi yapılmıştır. 2,5-13,5 sn aralığında kararlı duruma göre ve 0-30 sn aralığında ise tüm sinyali kapsayacak şekilde tanıma işlemleri yapılmıştır. Elde edilen sonuçlar Tablo 1’de verilmiştir. CS algoritmasının korelasyon sonuçlarına

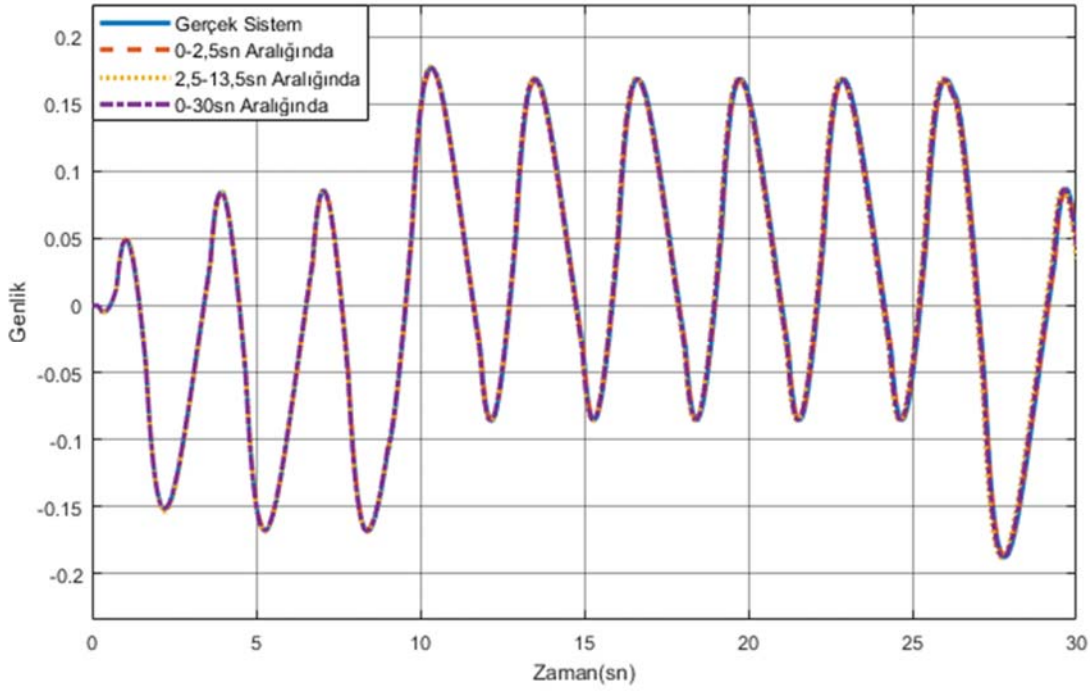
bakıldığında, en iyi yakınsayan bölgenin sinyalin tamamını içeren 0-30 sn aralığı olduğu görülmektedir. Korelasyonlara bakıldığında, en kötü yakınsayan durumun ise 0-2,5 sn aralığındaki geçici halde olduğu görülmektedir. Kararlı durumda yapılan tanıma işleminde ise geçici hale göre çok daha iyi sonuçlar verdiği görülmektedir. En kötü durumun olduğu 0-2,5 sn aralığında CS algoritmasının bu kadar kötü sonuçlar vermesinin nedeni, lokal bir bölgeye takılarak orada kalması veya veri sayısının az olmasından dolayı çözüm noktasının istenilen amaç fonksiyonunu minimum yapan başka bir noktaya takılmış olması olarak değerlendirilebilir. Ortalama iterasyon sayısına bakıldığında, genel tanıma sürelerinin artmasına bağlı olarak iterasyon sayılarının ve iterasyon ortalama değerlerinin arttığı görülmektedir. İterasyon sayısının ortalama değerden sapmalarına bakıldığında en çok sapma 2,5-13,5 sn aralığında meydana gelirken, en az sapma 0-2,5 sn aralığında meydana gelmiştir. Algoritmanın bulunduğu en iyi ve en kötü parametre değerleri Tablo 2’de verilmiştir. Aralıklara göre tanıyan sistem parametrelerinin en iyi cevapları Şekil 8’de, en kötü cevapları Şekil 9’da verilmiştir. Diğer algoritmalarla eş olabilmesi için Parçacık Sürü Optimizasyonunun popülasyon sayısı 40 ve maksimum iterasyon sayısı 1000 olarak alınmıştır. Atalet momentum katsayısı 0,9, c_1 ve c_2 2 olarak belirlenmiştir.

Tablo 1. CS Algoritmasının Sonuçları(CS Algorithm Results)

Zaman Aralığı	Korelasyonu	Yüzde Hatası	Ortalama İterasyon Sayısı	Standart Sapma
0-2,5 sn	0,729	0,825	266,5	123,759
2,5-13,5 sn	0,903	0,173	514,3	221,836
0-30 sn	0,917	0,130	761,0	217,480

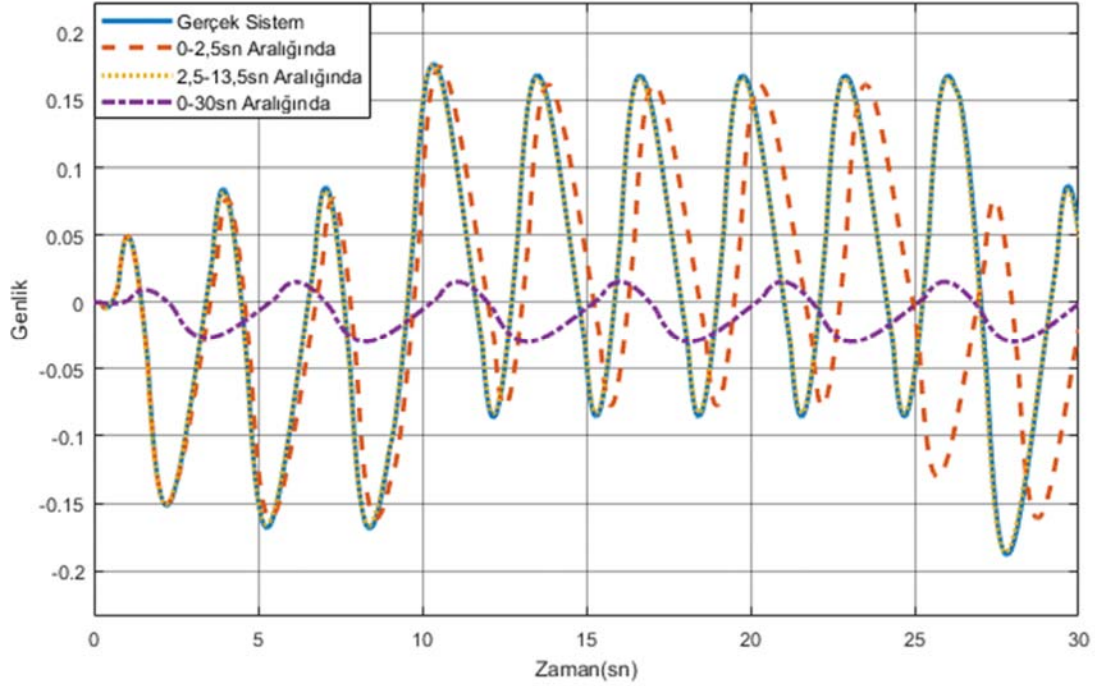
Tablo 2. CS Algoritmasının En İyi ve En Kötü Parametre Sonuçları(Best and worst parameter results of the CS Algorithm)

	0-2,5 sn		2,5-13,5 sn		0-30 sn	
	En iyi	En kötü	En iyi	En kötü	En iyi	En kötü
K	1,033	1,987	1,014	1,177	1,000	0,200
T ₀	0,202	0,199	0,204	0,197	0,200	0,381
T ₁	1,303	3,785	0,813	1,718	0,998	1,471
T ₂	0,788	0,491	1,220	0,664	1,000	2,000
L	0,198	0,201	0,199	0,198	0,199	0,198
Korelasyon	0,930	0,663	0,952	0,824	0,999	0,512

**Şekil 8.** Farklı Zaman Aralıkları için Parametreleri CS Algoritması ile Belirlenen En İyi Sistem Çıktıları
(Best outputs of the system, where its parameters are determined by the CS Algorithm for different time intervals)

Elde edilen sonuçlar Tablo 3’de verilmiştir. Parçacık Sürü Optimizasyonunda korelasyon analizlerine bakıldığında, 0-2,5 sn aralığında parametrelerin en iyi benzediği sonuçlar elde edilmiştir. En kötü benzeyen aralık 0-30 sn aralığıdır. Bu analizi yaparken bazı yerler büyük ihtimalle lokal bir noktaya girmiş ve oradan kurtulamadığı için en kötü benzeyen sonuçlar oluşmuştur. 0-2,5 sn aralığında, yüzde hataya bakıldığında daha yüksek bir hata aralığına sahip olduğu, korelasyonuna bakıldığında ise en iyi benzeyen çıktığı görülmüştür. Bunun nedeni, parametreleri istenilen kısıtların içinde kaldığı için korelasyonu sapmamış, fakat yüzde hatalar tüm çözüm kümesinin ortalaması olduğu için böyle bir sonuç vermiştir. İterasyon sayılarına bakıldığında ise yine en az iterasyon da istenilen şartları sağlayan 0-2,5 sn

zaman aralığı olmuştur. Standart sapmalarına bakıldığında en yüksek standart sapmayı yapan 2,5-13,5 sn aralığı olmuştur. Böyle olduğu için bu algoritma ile bu zaman aralığında tanıma yapmanın, başlangıç değerlerine bağlı olduğu anlaşılmaktadır. Çünkü, bu algoritmanın bu aralıkta lokal bölgelere sıkıştığı görülmektedir zira standart sapması çok yüksektir. En çok iterasyon yapan aralık yine 0-30 sn aralığıdır. Bu, sistemin çözüm yapabilmesi için aralıkta yeterli bir iterasyon sayısının olmadığını göstermektedir. Çünkü bazı deneylerde erken tanıma yapmış ama iterasyon sayısı yetmediği için standart sapması yükselmiştir. Algoritmanın bulduğu en iyi ve en kötü parametre değerleri Tablo 4’de verilmiştir. Aralıklara göre tanıyan sistem parametrelerinin en iyi cevapları Şekil 10’da, en kötü



Şekil 9. Farklı Zaman Aralıkları için Parametreleri CS Algoritması ile Belirlenen En Kötü Sistem Çıktıları (Worst outputs of the system, where its parameters are determined by the CS Algorithm for different time intervals)

Tablo 3. PSO Sonuçları(PSO Results)

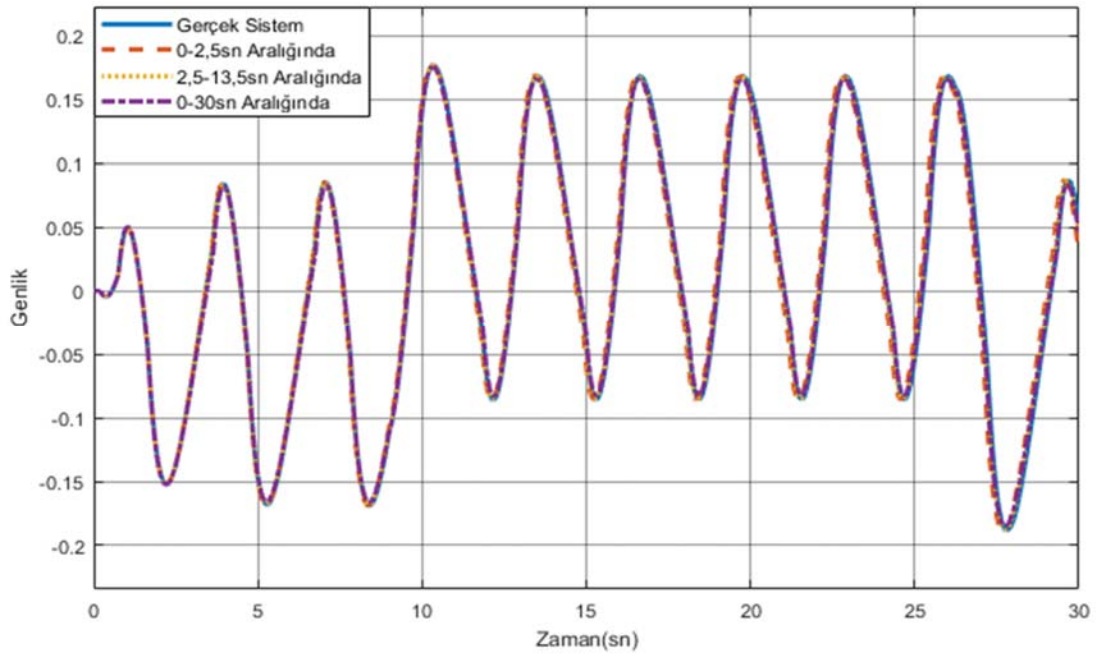
Zaman Aralığı	Korelasyonu	Yüzde Hatası	Ortalama İterasyon Sayısı	Standart Sapma
0-2,5 sn	0,909	0,209	30,1	14,151
2,5-13,5 sn	0,887	0,185	720,7	479,972
0-30 sn	0,284	0,230	941,8	324,560

Tablo 4. PSO'nun En İyi ve En Kötü Parametre Sonuçları (Best and worst parameter results of the PSO)

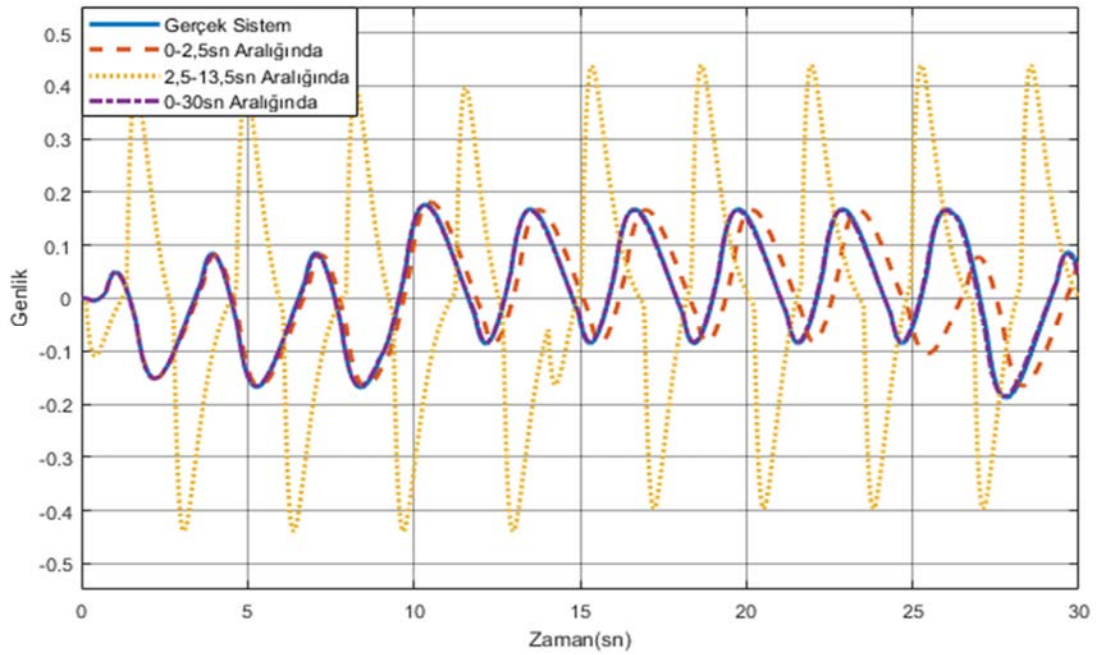
	0-2,5 sn		2,5-13,5 sn		0-30 sn	
	En iyi	En kötü	En iyi	En kötü	En iyi	En kötü
K	1,000	1,287	1,025	0,206	1,208	1,200
T ₀	0,199	0,198	0,197	2,000	0,201	0,202
T ₁	1,012	1,952	1,177	0,578	1,885	0,622
T ₂	0,977	0,660	0,866	0,198	0,605	1,837
L	0,200	0,198	0,200	0,133	0,202	0,198
Korelasyon	0,999	0,797	0,971	0,759	0,526	-0,269

cevapları ise Şekil 11'de verilmiştir. FA algoritmasında da diğer algoritmalarla eş olabilmesi için, popülasyon sayısı 40 ve maksimum iterasyon sayısı 1000 olarak alınmıştır. Çekiciliğin başlangıç değeri (β_0) 0,8, ışık yoğunluğu (γ) 0,9 olarak seçilmiştir. Sonuçlar Tablo 5'de verilmiştir. FA algoritmasının sonuçlarına bakıldığında, korelasyon

analizinden istenilen parametrelerin en iyi yakınsandığı tanıma bölgesinin 0,92 ile 2,5-13,5 sn aralığında meydana geldiği görülmektedir. Diğer değerlerinde bu değere yakın sonuçlar verdiği görülmektedir. Yüzde hataları da birbirine yakın fakat yine 2,5-13,5 sn aralığı diğerlerine göre daha düşük bir hata oranı vermiştir.



Şekil 10. Farklı Zaman Aralıkları için Parametreleri PSO Algoritması ile Belirlenen En İyi Sistem Çıktıları
(Best outputs of the system, where its parameters are determined by the PSO Algorithm for different time intervals)



Şekil 11. Farklı Zaman Aralıkları için Parametreleri PSO Algoritması ile Belirlenen En Kötü Sistem Çıktıları
(Worst outputs of the system, where its parameters are determined by the PSO Algorithm for different time intervals)

Tablo 5. FA Algoritmasının Sonuçları (FA Results)

Zaman Aralığı	Korelasyonu	Yüzde Hatası	Ortalama İterasyon Sayısı	Standart Sapma
0-2,5 sn	0,923	0,1539	161,90	122,97
2,5-13,5 sn	0,941	0,1551	861,00	271,68
0-30 sn	0,919	0,1558	997,90	76,58

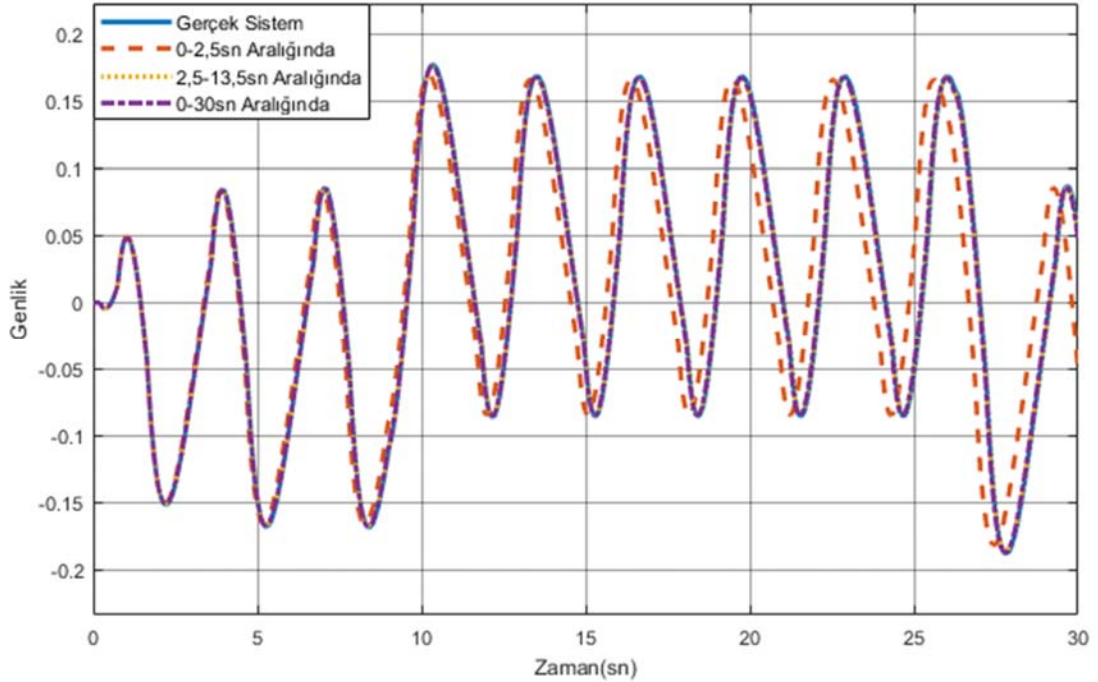
Standart sapmaya bakıldığında bu değer, iterasyon sayısının sapması olarak değerlendirilmiş ve en çok standart değerden sapma 2,5-13,5 sn aralığında meydana gelmiştir. 2,5-13,5 sn aralığındaki korelasyona bakıldığında diğer korelasyon sonuçlarına göre daha doğru çözüm bulunmuştur. Bununla birlikte ortalama iterasyon sayısının yüksek olması ise denemelerde seçilen maksimum iterasyon sayısının yetersiz olduğunu ve bazı denemelerin durdurma kriteri olan maksimum iterasyon sayısında durdurulduğunu göstermektedir. Bu aynı zamanda standart sapmanın yüksek olmasından daha iyi anlaşılmaktadır. Dolayısı ile global çözüm bulunamadan algoritma durdurulduğu için bazı sonuçlar lokal çözümde bulunmuştur.

Sinyalin tamamını almak iyi görünse de, zaman açısından ve işlem yükü bakımından verimsiz olmaktadır. Yine sinyalin tamamını almak, sistemi çok iyi tanıyamamasına neden

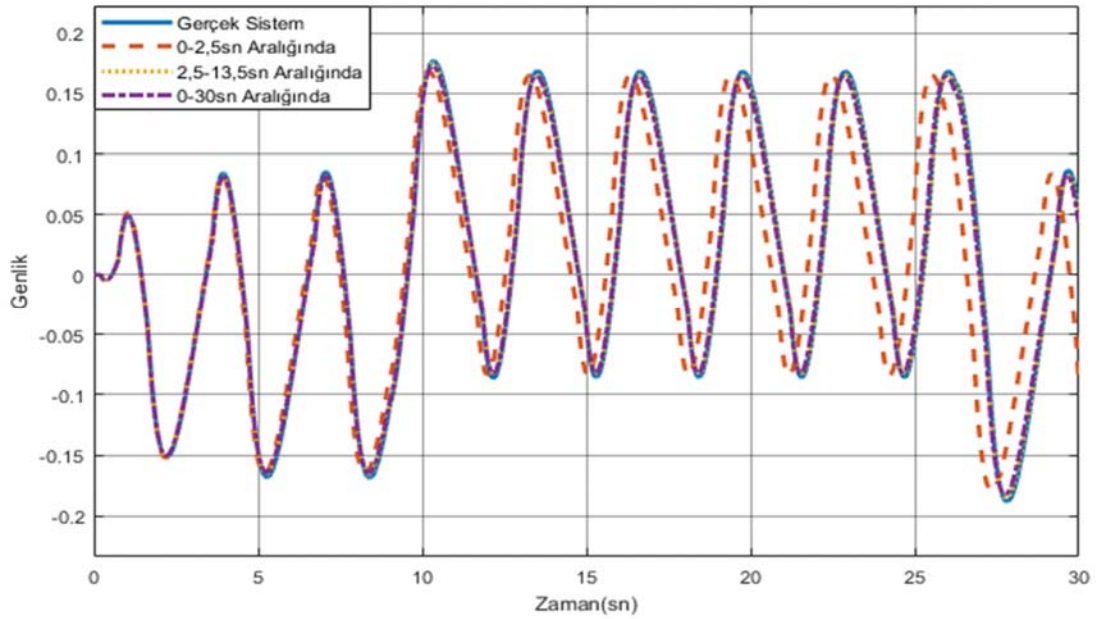
olmaktadır. Bunun nedeni iterasyon sayısının kısıtlı olmasıdır. Ortalama iterasyon sayısına bakıldığında 997,9 gibi bir değer vermiştir. Fakat algoritmanın durdurma kriteri 1000 olduğu için, 0-30 sn aralığında yapılan tanıma işlemi için iterasyon sayıları yetersiz kaldığı görülmektedir. Algoritmanın bulunduğu en iyi ve en kötü parametre değerleri Tablo 6'da verilmiştir. Aralıklara göre tanınan sistem parametrelerinin en iyi cevapları Şekil 12'de, en kötü cevapları Şekil 13'de verilmiştir. Aynı sistem için Kaya vd.'nin otomatik ayar testine göre Genetik Algoritma ile elde ettikleri ölü zamanlı ters cevaplı birinci dereceden sistem parametrelerinin sonuçları, Tablo 7'de verilmiştir. Elde edilen parametrelerin korelasyonlarına bakıldığında, aynı yönlü fakat diğer algoritmalara nazaran zayıf bir benzerliği olduğu görülmektedir. Bu parametreler röle testi ile sınanıldığında elde edilen sonuçlar ile gerçek sistem sonuçları, Şekil 14'de verilmiştir.

Tablo 6. FA Algoritmasının En İyi ve En Kötü Parametre Sonuçları (Best and worst parameter results of the FA)

	0-2,5 sn		2,5-13,5 sn		0-30 sn	
	En iyi	En kötü	En iyi	En kötü	En iyi	En kötü
K	0,970	1,074	1,057	1,187	1,007	1,174
T ₀	0,201	0,204	0,201	0,196	0,199	0,200
T ₁	1,089	0,619	1,382	0,651	1,041	0,627
T ₂	0,870	1,606	0,757	1,757	0,971	1,800
L	0,198	0,202	0,198	0,200	0,198	0,200
Korelasyon	0,984	0,815	0,999	0,813	0,998	0,797



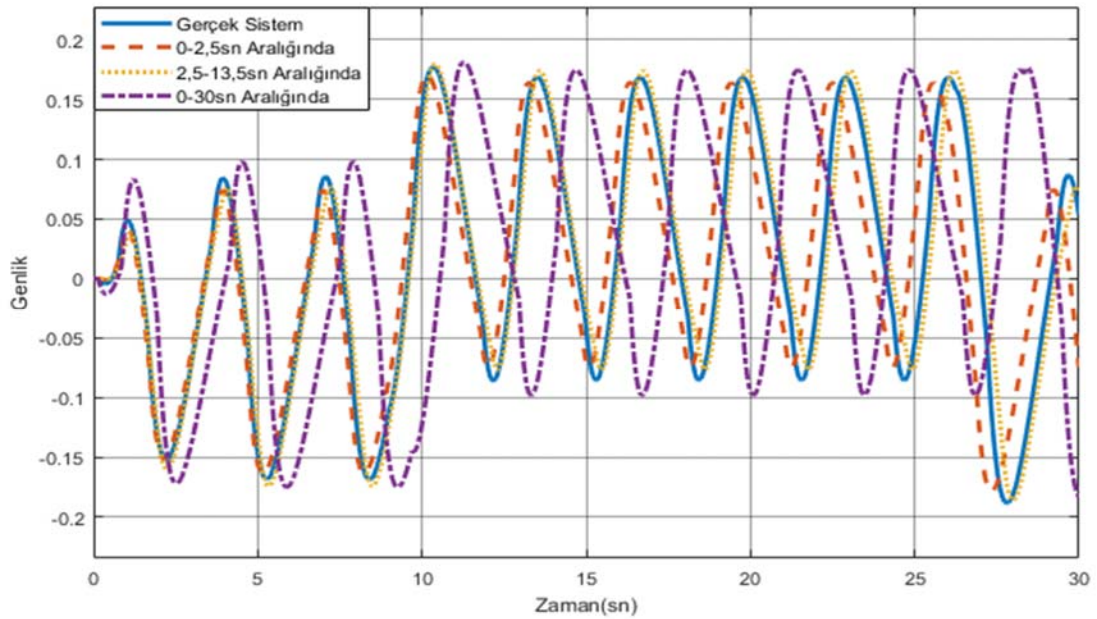
Şekil 12. Farklı Zaman Aralıkları için Parametreleri FA Algoritması ile Belirlenen En İyi Sistem Çıktıları (Best outputs of the system, where its parameters are determined by the FA Algorithm for different time intervals)



Şekil 13. Farklı Zaman Aralıkları için Parametreleri FA Algoritması ile Belirlenen En Kötü Sistem Çıktıları (Worst outputs of the system, where its parameters are determined by the FA Algorithm for different time intervals)

Tablo 7. Genetik Algoritma ile Elde Edilmiş Parametre Sonuçları (Parameter Results obtained by Genetic Algorithm) [13]

	0-2,5 sn	2,5-13,5 sn	0-30 sn
K	1,500	1,088	1,199
T_0	0,111	0,010	0,377
T_1	2,765	2,000	2,711
T_2	0,428	0,400	0,420
L	0,320	0,500	0,140
Korelasyon	0,665	0,642	0,617



Şekil 14. Genetik Algoritmasının Röle Testi Sonuçları (Relay Test Results of the Genetic Algorithm)

6. SONUÇLAR (CONCLUSIONS)

Bu çalışmada ölü zamanlı ters tepmeli bir sistemin parametreleri bulunmuştur. Bunun için ölü zamanlı ters tepmeli sistem, röle testi ile osilasyona sokulmuş ve elde edilen sonuçlara göre CS, PSO ve FA algoritmaları ile sisteme uygun olan modelin parametreleri belirlenmiştir. Algoritmaların çalışabilmesi için gerekli olan amaç fonksiyonu olarak, yaygın kullanılan IAE performans kriteri seçilmiştir. Çalışmada ISE performans kriteri de seçilebilirdi, diğer taraftan literatürdeki çalışmalar göz önüne alındığında ve hata değeri birden küçük olduğunda, karesel ifade hatayı daha da küçülteceğinden dolayı IAE tercih edilmiştir. Farklı hata kriterlerinden olan zamanın momentleri şekline yazılan ITSE (Integral Time Square Error, Karesel Hatanın Zamanla İntegrali) veya ITAE (Integral Time Absolute Error, Mutlak Hatanın Zamanla İntegrali) amaç fonksiyonlarının kullanılmamasının nedeni ise, zaman küçükken hatanın ağırlığının az olması sebebi ile sistem cevabındaki geri tepmenin hataya etkisini azaltacak olmasıdır. Bu iki kriterin kullanılması durumunda, sistem sıfırının bu performans kriteri için önemi daha az olacaktır. Zira bu kriterler için ilerleyen zamanların hatası, ilk baştaki hataya nazaran daha baskındır. Algoritmaları durdurma kriteri olarak, IAE hata kriterinin 0.001 değerinin altına düşmesi veya iterasyon sayısının maksimum 1000'e ulaşması seçilmiştir. İterasyon bakımından incelendiğinde, optimize edilmek istenen zaman aralığı arttığında, belirlenen 1000 iterasyon sayısının yetersiz kaldığı görülmüştür. Doğruluk bakımından korelasyonlarına bakıldığında ise CS [0,512, 0,999] arasında, PSO [-0,269, 0,999] ve FA [0,797, 0,999] aralığında parametre değerleri bulmuştur. Buradan görüldüğü üzere, FA algoritması belirlenen kısıtlarda daha doğru sonuç bulabilmektedir.

Ayrıca Kaya ve Nalbantoğlu'nun tek kanallı röle testine göre Genetik Algoritma kullanarak yapmış oldukları çalışmadaki en iyi parametre değerleri ile bu çalışmada yapılan CS, PSO ve FA algoritmalarının en iyi parametre değerleri, gerçek sistem değerleri ile karşılaştırılmıştır. CS, PSO ve FA algoritmaları global aramada daha başarılı olduğu için korelasyonlarının daha yüksek çıktığı görülmüştür. Bunun anlamı CS, PSO ve FA algoritmalarının gerçeğe daha yakın sonuçlar vermesidir.

KAYNAKLAR (REFERENCES)

- Ziegler J.G., Nichols N.B., Rochester N.Y., Optimum settings for automatic controllers, Transaction of the ASME, 64 (11), 220-222, 1942.
- Åström, K.J. and T. Hägglund, Automatic Tuning of Simple Regulators With Specifications on Phase and Amplitude Margins. Automatica, 20 (5), 645-651, 1984.
- Tsympkin Y.Z., Frequency responses of relay systems, Automat. Telemekh., 20 (12), 1603-1610, 1959.
- Tsympkin, Y.Z., Relay Control Systems, Cambridge University Press, 1984.
- Luyben W.L., Derivation of Transfer Functions for Highly Nonlinear Distillation Columns. Ind. Eng. Chem. Res., 26, 2490-2495, 1987.
- Yu C.-C., Autotuning of PID Controllers A Relay Feedback Approach. 2nd Edition ed., Springer Science & Business Media, 2006.
- Shen S.H., Wu J.S. ve Yu C.C., Use of biased-relay feedback for system identification, AIChE Journal, 42 (4), 1174-1180, 1996.
- Li W., Eskinat E., ve Luyben W.L., An improved autotune identification method, Industrial & engineering chemistry research, 30 (7), 1530-1541, 1991.
- Kaya I. ve Atherton D., Parameter estimation from relay autotuning with asymmetric limit cycle data, Journal of Process Control, 11 (4), 429-439, 2001.
- Friman M. ve Waller K.V., A two-channel relay for autotuning. Industrial & engineering chemistry research, 36 (7), 2662-2671, 1997.
- Soltész K., Hägglund T. ve Åström K.J., Transfer function parameter identification by modified relay feedback, American Control Conference (ACC), 2164-2169, 2010.
- Berner J., Åström K.J. ve Hägglund T., Towards a new generation of relay autotuners, IFAC Proceedings, 47 (3), 11288-11293, 2014.
- Kaya İ. ve Nalbantoğlu M., Röle geri-beslemeli sistemlerde genetik algoritma ile modelleme, Dicle Üniversitesi Mühendislik Fakültesi Dergisi, 3 (1), 31-39, 2012
- Yang, X.-S., Nature-inspired optimization algorithms 1st edition, Elsevier, 2014.
- Eberhart R. ve Kennedy J., A new optimizer using particle swarm theory, In Micro Machine and Human Science, MHS'95 Proceedings of the Sixth International Symposium on, IEEE, 39-43, 1995.
- Kennedy J. ve Eberhart R., Particle swarm optimization, International Conference on Neural Networks IEEE, 4, 1942-1948, 1995.
- Özsağlam M.Y., Çunkaş M., Optimizasyon problemlerinin çözümü için parçacık sürü optimizasyonu algoritması, Politeknik Dergisi, 11 (4), 299-305, 2008.
- Tozan A., Sevilgen F.E. ve İnce O., Sensör Yerleştirme Probleminin Parçacık Sürü Optimizasyonu ile Çözümü, Elektrik Elektronik Bilgisayar Mühendisliği 12. Ulusal Kongresi, http://www.emo.org.tr/etkinlikler/ulusal/etkinlik_bildiri_leri_detay.php?etkinlikkod=27&bilkod=2989. Yayın Tarihi Kasım 18, 2007. Erişim Tarihi Haziran 20, 2017.
- Öztürk A. ve Alkan S., Parçacık sürü optimizasyonu ile fotovoltaik sistemlerde kullanılan dc-dc dönüştürücünün kontrolü, İleri Teknoloji Bilimleri Dergisi, 2 (3), 110-120, 2013.
- Aydemir M.E., Performance of some optimization methods for patch antenna problems, Journal of the Faculty of Engineering and Architecture of Gazi University, 29 (3), 579-588, 2014.
- Gao Z., Zeng X., Wang J ve Liu J., FPGA implementation of adaptive IIR filters with particle swarm optimization algorithm, Communication Systems ICCS 2008, 11th IEEE Singapore International Conference on IEEE, 1364-1367, 2008.

22. Öztürk C., Hançer E. ve Karaboga D., Automatic clustering with global best artificial bee colony algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 29 (4), 677-687, 2014.
23. Tavakoli S. ve Banookh A., Robust PI control design using particle swarm optimization. *Journal of Computer Science and Engineering*, 1 (2), 2010.
24. Satpati B., Koley C., ve Datta S., Robust PID controller design using particle swarm optimization-enabled automated quantitative feedback theory approach for a first-order lag system with minimal dead time, *Systems Science & Control Engineering*, 2 (1), 502-511, 2014.
25. Kanthaswamy G., ve Jerome J., Control of dead-time systems using derivative free particle swarm optimisation, *International Journal of Bio-Inspired Computation*, 3 (2), 85, 2011.
26. Ahuja A., Narayan S. ve Kumar J., 2 DOF Observer Based Controller for First Order with Dead Time Systems, *Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 8 (1), 106-110, 2014.
27. Gandomi A.H., Yang X.-S.ve Alavi A.H., Cuckoo Search algorithm: a metaheuristic approach to solve structural optimization problems. *Springer-Verlag*, 29 (1), 18, 2013.
28. Elazim A.S.M. and Ali E.S., Optimal Power System Stabilizers design via Cuckoo Search algorithm, *International Journal of Electrical Power & Energy Systems*, 75, 99-107, 2016.
29. Nema, S. and P. Kumar Padhy, Identification and cuckoo PI-PD controller design for stable and unstable processes, *Transactions of the Institute of Measurement and Control*, 37 (6), 708-720, 2015.
30. Jin, Q., Qi L., Jiang B. ve Wanh Q., Novel improved cuckoo search for PID controller design, *Transactions of the Institute of Measurement and Control*, 37 (6), 721-731, 2015.
31. Roeva, O. ve Slavov T., Firefly algorithm tuning of PID controller for glucose concentration control during E. coli fed-batch cultivation process, *Proceedings of the Federated Conference on Computer Science and Information Systems*, 455-462, 2014.
32. Sung, S.W., Park J.H. ve Lee I.B., Modified Relay Feedback Method, *Industrial & engineering chemistry research*, 34 (11), 4133-4135, 1995.
33. Soderstrom T. ve Stoica P., *System Identification*, Prentice-Hall, 1989.
34. Nise N.S., *Control Systems Engineering 6th ed.*, John Wiley & Sons, 2011.
35. Yang, X.S., *Cuckoo Search and Firefly Algorithm Theory and Applications*, Springer International Publishing, 2013.
36. Yang, X.S. ve Deb S., Cuckoo Search via Lévy Flights, *Nature & Biologically Inspired Computing*, 210 - 214, 2009
37. Karagül K., Guguk Kuşu Algoritması: Bir Plastik Atık Toplama Uygulaması, *15th International Symposium on Econometrics, Operations Research and Statistics*, Isparta, Turkey, 15, 775-784, 2014.
38. Civicioglu, P. ve Besdok E., A conceptual comparison of the Cuckoo search, particle swarm optimization, differential evolution and artificial bee colony algorithms, *Artificial Intelligence Review*, 39 (4), 315-346, 2011.
39. Yang, X.-S., *Firefly Algorithms for Multimodal Optimization*, 169-179, 2010.
40. Belen, M.A., Alıcı M. ve Güneş F., Ateşböceği Algoritması ile Mikrodalga Transistör Performansının Karakterizasyonu, *Elektrik, Elektronik, Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu ELECO-2014*, 491-494, 2014.

