

Araștırma Makalesi

Research Article

BİLİNMEYEN MARKOV ATLAMALI SİSTEMLERİN MODELLEMESİ VE ERGEN KİMLİK ARAMA ALGORİTMASI İLE AYARLANMIŞ PID KONTROLÜ

Bedri BAHTİYAR^{1*}, Meriç ÇETİN², Selami BEYHAN³

¹Pamukkale Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, Denizli, Türkiye ²Pamukkale Üniversitesi, Bilgisayar Mühendisliği Bölümü, Denizli, Türkiye ³Yildiz Teknik Üniversitesi, Yapay Zekâ ve Veri Mühendisliği Bölümü, İstanbul, Türkiye

Anahtar Kelimeler	Öz				
Markov Atlamalı Sistemler, Markov atlama sistemlerinin (Markov Jump System-MJS), bilinmeyen dina					
Aşırı Öğrenme Makineleri,	rastgele geçişler ve çevresel gürültüler nedeniyle denetlenmesi zordur. Bu				
PID Denetim,	makalede, gerçek zamanlı doğrusal MJS'ler optimizasyon yöntemleri kullanılarak				
Meta-Sezgisel Optimizasyon,	genel modelleme ve denetim performansını iyileştirmek için gözden geçirilmiştir.				
AISA.	Bu çalışmayla elde edilen katkılar iki başlıkta değerlendirilmektedir: i) gerçek				
	zamanlı bir RLC devresinden toplanan veriler kullanılarak kara-kutu tanımlama, ii)				
	oransal-integral-türev (Proportional-Integral-Derivative - PID) denetleyicinin				
	tasarımında sezgisel optimizasyon yöntemi olan Ergen Kimliği Arama				
	algoritmasının (AISA) ilk kez kullanımı. Bu amaçla, bilinmeyen MJ'lerin				
	dinamiklerini modellemek ve tahmin etmek için bir Aşırı Öğrenme Makinesi				
	(Extreme Learning Machine- ELM) modeli oluşturulmuştur. Ardından, yığın				
	optimizasyon içerisinde ELM modeli kullanılarak en uygun PID parametreleri				
	kümesi bulunmuştur. Denetleyicinin parametrelerini optimize etmek için				
	literatürde yaygın olarak kullanılan meta-sezgisel algoritmalar AISA ile				
	karşılaştırılmıştır. Simülasyon sonuçlarına göre en iyi uygunluk değerine en kısa				
	sürede ulaşan AISA ile gerçek zamanlı PID denetleyicisine ait parametreler 0.005				
	hata oranı ile tahmin edilmiştir. Önerilen yaklaşım, Markov davranışı sergileyen				
	deneysel bir RLC devresinin modellenmesi ve denetimi için uygulanmıştır.				

MODELLING OF UNKNOWN MARKOV JUMP SYSTEMS AND PID CONTROL TUNED BY ADOLESCENT IDENTITY SEARCH ALGORITHM

Keywords	Abstract
Markov Jump Systems,	Markov jump systems (MJS) are difficult to control due to unknown dynamics,
Extreme Machine Learning,	random transitions and environmental noises. In this paper, real-time linear MJSs
Pid Control,	are reviewed to improve general modeling and control performance using meta-
Meta-Heuristic Optimization,	heuristic optimization methods. Contributions are twofold as: i) black-box
AISA.	identification using collected data from a real-time RLC circuit, ii) first use of the
	Adolescent Identity Search algorithm (AISA), which is a meta-heuristic optimization
	method in the design of a proportional-integral-derivative (PID) controller. For this
	purpose, an Extreme Learning Machine (ELM) model is constructed to model and
	predict the dynamics of unknown MJs. Then, the optimal set of PID parameters are
	found using the ELM model in batch optimization. To optimize the parameters of the
	controller, meta-heuristic algorithms commonly used in the literature are compared
	with AISA. According to the simulation results, the parameters of the real-time PID
	controller have been estimated with an error rate of 0.005 with AISA, which
	achieved the best fittness value in the shortest time. The proposed approach is
	applied to model and control an experimental RLC circuit with Markovian behavior.
Alıntı / Cite	
Bahtiyar, B., Çetin, M., Beyha	n, S., (2025). Bilinmeyen Markov Atlamalı Sistemlerin Modellemesi ve Ergen Kimlik
Arama Algoritması ile Avarla	nmış PID Kontrolü, Mühendislik Bilimleri ve Tasarım Dergisi, 13(1), 1-16.

Yazar Kimliği / Author ID (ORCID Number)	Makale Süreci / Article Process		
B. Bahtiyar, 0000-0002-8679-095X	Başvuru Tarihi / Submission Date	05.07.2024	
M. Çetin, 0000-0002-7871-4850	Revizyon Tarihi / Revision Date	21.10.2024	
S. Beyhan, 0000-0002-9581-2794	Kabul Tarihi / Accepted Date	21.11.2024	
	Yayım Tarihi / Published Date	20.03.2025	

^{*} İlgili yazar / Corresponding author: bedribahtiyar@pau.edu.tr, +90-505-409-5975

MODELLING OF UNKNOWN MARKOV JUMP SYSTEMS AND PID CONTROL TUNED BY ADOLESCENT IDENTITY SEARCH ALGORITHM

Bedri Bahtiyar^{1†}, Meriç Çetin², Selami Beyhan³,

¹Pamukkale Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, Denizli, Türkiye
²Pamukkale Üniversitesi, Bilgisayar Mühendisliği Bölümü, Denizli, Türkiye
³Yildiz Teknik Üniversitesi, Yapay Zekâ ve Veri Mühendisliği Bölümü, İstanbul, Türkiye

Highlights

- Modelling of unknown Markov jump systems
- Adolescent identity search algorithm (AISA)
- Meta-heuristic optimization based PID parameter identification
- System modelling with Extreme Learning Machine via input-output data

Graphical Abstract



Purpose and Scope

In this paper, real-time linear Markov jump systems (MJSs) are reviewed to improve general modeling and control performance using meta-heuristic optimization methods. Contributions are twofold asi) black-box identification using collected data from a real-time RLC circuit, ii) first use of the Adolescent Identity Search algorithm (AISA), which is a heuristic optimization method in the design of a proportional-integrator-derivative (PID) controller.

Design/methodology/approach

To achieve the determined objectives, first the ELM model of the RLC circuit, which acts as a MJS, is obtained. Then, PID controller parameters are generated by simulating the ELM model with various well-known metaheuristic algorithms. Finally, the PID controller parameters obtained by AISA are applied to the RLC circuit for real-time experiment.

Findings

Simulation results illustrate that the AISA has an acceptable performance to find PID controller parameters by using ELM model. Furthermore, real-time experiment results present that PID controller, parameters of which are obtained by AISA, can manipulate the RLC circuit, which behaves as MJS, to track a reference signal containing staircase and sinusoidal signals.

Originality

In this study, black-box identification of a RLC circuit by ELM and identification-model based design of PID controller using AISA are proposed. Comparative simulations and real-time results show that AISA is a very effective meta-heuristic optimization to find the parameters of PID controller to control RLC circuit which acts as MJS.

⁺ Corresponding author: bedribahtiyar@pau.edu.tr, +90-505-409-5975

1. Giriș (Introduction)

PID denetleyici hem akademik araştırmalar hem de endüstri uygulamaları için basit, verimli ve uygulanabilir bir denetleyicidir. En önemli avantajlarından biri, gerçek zamanlı uygulamalar için çok önemli bir özellik olan sistem modelinin uygulanmasına gerek olmamasıdır. Temel amacının çıkış geri besleme denetimi olması nedeniyle sadece çıkış ölçümü denetleme işlemi için yeterlidir. Bir PID denetleyicisinin temel sorunu, istenen bir performansı karşılamak için parametrelerinin uygun değerlere ayarlanması gerektiğidir (Cetin ve Iplikci, 2015). Literatürde kararlılık, izleme ve düzenleme performansı, tesis modelleme belirsizliğine karşı dayanıklılık, gürültü azaltma ve çevresel belirsizliğe karşı dayanıklılık gibi kriterleri dikkate alan birçok farklı PID tasarım yöntemi bulunmaktadır (Ang vd., 2005).

Hedeflenen kriterlere göre farklılık gösteren PID tasarımları analitik, sezgisel, frekans yanıtı tabanlı, optimizasyon tabanlı veya uyarlanabilir ayar tabanlı olarak geliştirilebilir. PID parametrelerini hesaplamak için analitik ilişkileri kullanan performans ve dayanıklılık kriterlerine dayalı farklı frekans özelliklerine sahip genel bir denetlevici (Lennartson ve Kristiansson, 2009)'de önerilmistir. Ziegler-Nichols (Z-N) PID denetleyicileri, islemin gecikme ve birim reaksiyon hızı değerlerini kullanarak parametreleri en uygun şekilde ayarlar (Ziegler ve Nichols, 1942). Tasarım hedefleri arasında bir denge sağlayan yapay zekâ yöntemlerine (uzman sistemler, evrimsel hesaplama, bulanık mantık ve sinir ağları dahil) dayalı birçok PID denetleyici tasarımı vardır. Örneğin, (Chiou vd., 2012)'te bir süspansiyon sisteminin en uygun bulanık PID denetleyici parametrelerini belirlemek için Q-öğrenmeye dayalı evrimsel hesaplama sonuçları gösterilmiştir. (Paul vd., 2018)'da bilinmeyen belirsizlikleri telafi etmek için PD ve PID denetleyici ile birleştirilmiş bir tip-2 bulanık sistem uygulanmış ve denetleyicilerin kararlılığı kanıtlanmıştır. Hasan ve Abbas (Hasan ve Abbas, 2022), bir su altı robotik aracına uygulanan bozucu etkileri veya bilinmeyen dinamiklerinin etkilerini ortadan kaldırmak için doğrusal olmayan uyarlanabilir bulanık bir PID denetleyicisi önermiştir. (Van, 2018)'de belirsiz doğrusal olmayan sistemler için önerilen adaptif bulanık PID tabanlı tekil olmayan hızlı terminal kayma modu denetimi kullanılarak sistemin dayanıklılığının arttırılması ve kararlı durum hatasının azaltılması amaclanmıştır. Başka bir calışmada, sinir ağlarının ağırlıklarının verel optimale takılmasını önlemek için uyarlanabilir bir PID sinir ağı ve parçacık sürüsü optimizasyonu tabanlı denetleyici (Kang vd., 2014)'da sunulmuştur. Gerçek zamanlı tanımlama için genellikle çevrimiçi ayarlama yöntemleri tercih edilmektedir. Uyarlanabilir bir PID denetleyicisi (Chang ve Yan, 2005), Duffing-Holmes kaotik sistemi için bozucu etkilerin varlığında sistemin kapalı döngü kararlılığı garanti eden kayma modu denetimine dayalı olarak tasarlanmıştır. Kararlı bir gercek zamanlı tanımlamaya dayalı yeni bir Radial Basis Function (RBF) ağı kullanılarak cevrimici uyarlanabilir PID ayarı önerilmiştir ve (Beyhan ve Alci, 2010)'deki sıvı seviye sisteminin denetiminde uygulanmıştır. Son zamanlarda, karmaşık eşlenik sıralı sistemler için (Cetintas ve Hamamci, 2022)'de PID kararlılık yöntemi sunulmuştur.

Öte yandan Markov atlama sistemleri, rastgele geçiş olasılıklarına sahip stokastik sistemlerin bir sınıfıdır (Saravanakumar ve Ali, 2022). Geçişlere göre, sistemin davranışı aniden değişir ve bu da tahmin edilmesini ve denetlenmesini zorlaştırmaktadır (Kang vd., 2018). (Zhu vd., 2020)'deki farklı gürültü durumları için zamanlama şeması için dağıtılmış bir filtre tasarımı önerilmiştir. Belirsiz zaman gecikmeli Markov atlama sistemlerini denetlemek için (Fang vd., 2019)'da kayan modlu sağlam bir denetleyici önerilmiştir. Ayrıca, (Zhang vd., 2021)'de eşzamanlı olmayan gözlemci tabanlı kayma modu denetimi ile zaman gecikmeli ve konik tip doğrusal olmayan özellikte bir ayrık zamanlı Markov atlama sistemi incelenmiştir. (Vargas vd., 2017)'de, Markov atlama lineer sistemlerinin denetim problemi, mod gözlemi olmadan gerekli optimal koşulu hesaplayan sayısal bir yöntemle tartışılmıştır.

Sistem modelinin bilinmediği durumlarda bir denetleyici tasarlamak belirsizlik, performans, uygulanabilirlik ve doğruluk gibi farklı kriterler açısından zorlu bir süreçtir. Bu zorlukları başarmak için, sistem modellenmeden önce tasarımda kullanılacak yöntemin belirlenmesi ve model tahmini için mümkün olan en iyi verilerin kullanılması gerekmektedir. ELM (Huang vd., 2006), bir sistem modelleme problemini çözerken kullanılabilecek hızlı ve verimli bir yapay sinir ağıdır. ELM, model doğruluğunu artırmak için girdi verilerinin özelliklerini tanımlayan farklı özellik çıkarma yöntemleriyle birlikte kullanılır. Bu yöntemler, veri kümesindeki değişkenlerin önem sıralamasını belirleyerek model performansını artırır. ELM, mevcut durum ve girdi verileri verilirken MJS'nin tüm olası durumlarını modellemek ve tahmin etmek için eğitilebilir. Bu tahmin daha sonra geçiş olasılıklarına göre MJS'nin durumunu güncellemek için kullanılabilir.

Bir MJS'nin çıkış geri besleme denetimi, sistem modelini kullanmayan denetleme yöntemleri için zorlu bir iştir. Bu nedenle, yazarların en iyi bilgisine göre, bu sistemler özellikle gerçek zamanlı deneylerde zamanla değişen referans sinyalleri için denetlenmemiştir. Bu çalışmada, bilinmeyen MJS'lerin denetimi için verimli parametre tasarımlarına sahip modelleme tabanlı bir PID denetim cihazı önerilmektedir. Bu makalenin katkısının ardındaki temel fikir, önce bilinmeyen MJS'ler için verimli bir sistem tanımlama modeli tasarlamak ve geleneksel PID denetleyicisini yığın optimizasyon yoluyla optimize etmek, ardından modelleme tabanlı PID denetleyicisinin avantajını gerçek zamanlı uygulamada sergilemektir. Bu amaçla, ergen kimlik arama algoritması (AISA) optimizasyon yöntemini kullanarak ilk kez optimal bir PID denetleme cihazı tasarlanması amaçlanmıştır. Karşılaştırma amacıyla, elde edilen ELM modeli kullanılarak benzer bir PID denetleyicisi parçacık sürüsü optimizasyonu (Particle Swarm Optimization - PSO), yapay arı kolonisi algoritması (Artificial Bee Colony - ABC), gri kurt optimizasyon algoritması (Grey Wolf Optimization - GWO) ve balina optimizasyonu (Whale Optimization - WO) gibi popüler meta-sezgisel algoritmalar ile gerçeklenmiştir. Simülasyon sonuçları, önerilen yöntemin tercih edilebilirliğini göstermiş ve AISA ile elde edilen PID parametreleri MJS olarak davranan bir RLC devresinin gerçek zamanlı denetiminde kullanılmıştır.

Makalede önce 2. bölümde Markov atlamalı sistemler tanıtılmış ve ardından 3. bölümde PID denetleyicisi ve ELM ile model oluşturma tanıtıldıktan sonra, 4. bölümde meta-sezgisel optimizasyon algoritmaları tanıtılmıştır. Daha sonra 5. bölümde simülasyon çalışmaları ve deneysel çalışma detaylandırılmış ve son olarak 6. bölümde elde edilen sonuçlar değerlendirilmiştir.

2. Markov Atlama Sistemleri (Markov Jump Sistems)

Markov atlama sistemleri, gelecekteki sistem durumlarının yalnızca mevcut sistem durumuna göre belirlendiği rastgele değişen süreçleri modellemek için kullanılan bir tür stokastik süreçtir. Bir geri besleme mekanizması ile MJS durumunu istenen duruma yaklaştırmak için çeşitli denetim yöntemleri vardır. Sürekli zamanlı tek girişli tek çıkışlı doğrusal olmayan bir sistem

şeklinde verilebilir. Burada $x \in \Re^n$ durum vektörü, $u \in \Re$ giriş işareti, $y \in \Re$ çıkış işareti ve r_t Markov atlama girdisidir. $f(.) \in \Re^n$ ve $h(.) \in \Re$ doğrusal-olmayan türevlenebilir fonksiyonlardır. r_t teriminin, $r \in S =$ {1,2, ..., N} şeklinde verilen bir sonlu durum kümesinde değerler alan, ayrık-zamanlı Markov stokastik sürecine dayalı zamanla-değişen bir girdi olduğu varsayılmaktadır. Bu girdinin durum geçiş matrisi $\Pi = \pi_{rk}, r, k \in S$ ile verilmektedir. Geçiş olasılığı

$$P_r\{r_{t+\Delta t} = k \mid r_t = r\} \quad \begin{cases} \pi_{rk}\Delta t + o(\Delta t), & r \neq k\\ 1 + \pi_{rr}\Delta t + o(\Delta t), & r = k \end{cases}$$
(2)

olarak verilebilir. Burada $\Delta t > 0$ ve $\lim_{\Delta t \to 0} o(\Delta t) / \Delta t$ şeklindedir. $\pi_{rk} \ge 0$ ifadesi ise $\pi_{rk} = -\sum_{k=1,k \neq r}^{N} \pi_{rk}$ olmak üzere t anında r kipinden ve $t + \Delta t$ anında k kipinden geçiş oranlarını göstermektedir.

3. Yöntemler (Methods)

Bu çalışmada, dinamiklerinin bilinmediği varsayılan MJS'lerin çıkışlarının, zamanla değişen bir referans işaretini izleyebilmesi için verimli basit bir denetleyici olan PID denetleyicisinin parametrelerinin meta-sezgisel optimizasyon yöntemleri ile belirlenmesi ele alınmıştır.

3.1 Oransal-İntegral-Türev (PID) Denetimi (Proportional-Integral-Derivative Control)

Tasarımının sadeliği, verimliliği, işlevselliği nedeniyle endüstriyel uygulamalarda halen sıklıkla tercih edilen geleneksel PID denetleyicileri, bir sistemde istenen bir set noktasına ulaşmak veya bir işlemi istenen bir şekilde yönetmek için kullanılır. Geleneksel bir PID denetleyicisi, aktarım işlevi genellikle aşağıdaki gibi "paralel biçimde" yazılan bir denetleyici olarak da bilinir (Astrom ve Hagglund, 1995).

$$G(s) = K_p + K_i \frac{1}{s} + K_d s$$
⁽³⁾

Burada *s* terimi Laplace değişkenidir. K_p ; genel denetim eylemini sağlayan oransal-kazanç girdisidir. K_i ; süreklihal hatasını azaltmak için integral terimidir. K_d ; geçici hal cevabını iyileştiren türev terimidir (Ang vd., 2005). Bir geleneksel PID denetleyicisinin genel gösterimi

$$u(t) = K_p e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt},$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$
(4)

şeklindedir. Burada $T_i = K_p/K_i$ olmak üzere integral zaman sabitini, $T_d = K_d/K_p$ olmak üzere türev zaman sabitini, e(t) = r(t) - y(t) olmak üzere izleme hatasını, $y(t) \in \Re$ sistemin gerçek çıkışını ve r(t) referans işareti ya da ayar-noktasını ifade etmektedir. Bir PID denetleyicisinin başarımı en uygun PID girdilerinin seçimine bağlıdır. PID denetleyicisinin kazanç değerleri doğru bir şekilde ayarlanmazsa özellikle gerçek zamanlı sistemlerde bazı sorunlar ortaya çıkabilir (Abd-Elazim ve Ali, 2018). Örneğin, yanlış ayarlanmış PID kazançları, sistemde osilasyonlara ve titresimlere neden olabilir ki bu durum, kontrol sinyali üzerinde aşırı tepki ve kararlılık problemleriyle sonuclanabilir (Özay vd., 2022). Düsük kazanc değerlerine sahip sistem istenen set noktasına ulasmakta zorlanabilir veva tepki süresi uzavabilir. PID denetlevicisinin kazancları vanlıs avarlandığında. sistemde istenmeyen geçici tepkiler ortaya çıkabilir. PID denetleyicisinin kazançlarının doğru bir şekilde avarlanması, istenen performansı elde etmek ve kontrol sistemini kararlı tutmak icin önemlidir. PID denetlevici parametrelerinin optimum değerlerine ayarlanması bir optimizasyon problemi olarak düşünülebilir. Denetleyici parametrelerinin optimal değerlerinin bulunması sürecinde meta-sezgisel algoritmaların kullanılması denetleyicinin daha verimli çalışmasını sağlar (İzci ve Ekinci, 2023; Vincent ve Nersisson, 2017; Hekimoğlu, 2019; Qi vd., 2019; Joseph vd., 2022). Bu algoritmaların temel amacı, bir popülasyondaki bireylerin genetik operatörler aracılığıyla evrimleştirilmesi sonucunda yeni ve daha iyi bireyler oluşturmak ve bu sayede optimizasyon problemlerinin üstesinden gelmektir. Dördüncü bölümde, PID denetleyicisinin en uygun parametrelerini bulmak için bu çalışmada kullanılan bazı meta-sezgisel yöntemler özetlenmiştir.

3.2 Aşırı Öğrenme Makineleri (Extreme Learning Machines)

ELM ilk olarak gizli katman ileri-beslemeli sinir ağları (Hidden Layer Feedforward Neural Network - HLFN) için en küçük eğitim hatası, en küçük ağırlık normu ve en iyi genelleme performansı ile daha hızlı öğrenme oranları elde etmek için (Huang vd., 2004)'de önerilmiştir. ELM algoritması diğer HLFN algoritmalarından daha hızlı çıktı üretebilmektedir. ELM algoritması, bir tek katmanlı ileri-beslemeli sinir ağının (Single Layer Feedforward Neural Network – SLFN) giriş ağırlıkları ve giriş ofset değerleri keyfi olarak seçilirse, gizli katman çıkış matrislerinin genelleştirilmiş Moore-Penrose tersi kullanılarak çıkış ağırlıklarının elde edilebileceğini önermektedir. Şekil 1'de ELM modelinin yapısı görülmektedir.



Giriş katmanı $\mathbf{x} = [x_1, ..., x_M]^T \in \mathbb{R}^M$ şeklinde tanımlanan *M* boyutlu giriş vektöründen ve (*S* × *M*) boyutlu ve

$$\mathbf{W}_{i} = \begin{bmatrix} w_{i}^{(1,1)} & w_{i}^{(1,2)} & \cdots & w_{i}^{(1,M)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i}^{(S,1)} & w_{i}^{(S,2)} & \cdots & w_{i}^{(S,M)} \end{bmatrix}$$
(5)

şeklindeki giriş ağırlıklarından oluşmaktadır. Gizli katman, her biri toplama blokları ve aktivasyon işlevlerine sahip olan *S* adet sinirden oluşmaktadır. Son olarak, çıkış katmanı, $\mathbf{w}_o = \left[w_o^{(1)}, \dots, w_o^{(S)}\right]^T$ çıkış ağırlıkları ve toplama bloğundan oluşmaktadır. Böylece ELM modelin çıkışı \hat{y}

$$\hat{y} = \mathbf{w}_o^T \, \mathbf{f}(\mathbf{W}_i \mathbf{x} + \mathbf{b}) \tag{6}$$

şeklinde hesaplanabilir. Burada $\mathbf{b} = [b_1, ..., b_S]^T$ ofset vektörünü ifade etmekte olup, giriş ağırlıkları gibi değerleri rastgele seçilebilir ya da ihmal edilebilir. $\mathbf{f} = (x_1, ..., x_{M,} w_i^{(1,1)}, ..., w_i^{(S,M)}, b_1, ..., b_S)$ ise aktivasyon işlevidir ve tanjant-hiperbolik, Gaussian vb. işlevler bu amaçla kullanılabilir. NARX model biçiminde olduğu varsayılan bir bilinmeyen sistem,

$$y[n] = f(u_n, \dots, u_{n-n_u}, y_{n-1}, \dots, y_{n-n_y})$$
(7)

şeklinde verilebilir. Burada $[u_n, ..., u_{n-n_u}]$ giriş ve $[y_{n-1}, ..., y_{n-n_y}]$ çıkış vektörleri sırasıyla sistemin $(n_u + 1)$ adet girişi ve (n_y) adet çıkışını içermektedir. N adet giriş ve çıkış veri çifti için ELM modeli elde edilmek istenirse ilk olarak ϕ regresyon matrisi

$$\boldsymbol{\phi} = \begin{bmatrix} f\left(\mathbf{w}_{i}^{(1)}\mathbf{x}_{1} + \mathbf{b}_{1}\right) & \cdots & f\left(\mathbf{w}_{i}^{(S)}\mathbf{x}_{1} + \mathbf{b}_{S}\right) \\ \vdots & \ddots & \vdots \\ f\left(\mathbf{w}_{i}^{(1)}\mathbf{x}_{N} + \mathbf{b}_{1}\right) & \cdots & f\left(\mathbf{w}_{i}^{(S)}\mathbf{x}_{N} + \mathbf{b}_{S}\right) \end{bmatrix}$$
(8)

şeklinde tanımlanır. Burada $\mathbf{w}_i^{(1)}, ..., \mathbf{w}_i^{(S)}$; \mathbf{W}_i giriş ağırlıkları matrisinin $(n_u + n_y + 1)$ adet bileşenli bir satır vektörünü ifade etmektedir. $[\mathbf{x}_1, ..., \mathbf{x}_N]^T$ ise

$$\mathbf{x}_{1} = \begin{bmatrix} u_{0} \\ \vdots \\ u_{-nu} \\ y_{-1} \\ \vdots \\ y_{-n_{v}} \end{bmatrix}, \mathbf{x}_{2} = \begin{bmatrix} u_{1} \\ \vdots \\ u_{-nu+1} \\ y_{-0} \\ \vdots \\ y_{-n_{v+1}} \end{bmatrix}, \dots, \mathbf{x}_{1} = \begin{bmatrix} u_{N} \\ \vdots \\ u_{-nu+N} \\ y_{-1+N} \\ \vdots \\ y_{-n_{v}+N} \end{bmatrix}$$
(9)

şeklinde verilmiş olup, bir doğrusal olmayan (Nonlinear AutoRegressive eXenegous - NARX) modelin girişlerini temsil etmektedir. Buradan ELM model

$$\boldsymbol{\phi} \mathbf{w}_o = \mathbf{y} \tag{10}$$

olarak oluşturulabilir ki burada $\mathbf{y} = [y_1, ..., y_N]^T$ terimi çıkış vektörünü ifade etmektedir. Son olarak, problem bir doğrusal regresyon problemi biçimine dönüşür çünkü giriş ağırlıkları ve ofset değerleri rastgele olarak seçilmiştir ve geriye sadece bilinmeyen çıkış ağırlıklarının bulunması kalmıştır. Çıkış ağırlıkları da

$$\mathbf{w}_o = \boldsymbol{\phi}^\dagger \mathbf{y} \tag{11}$$

şeklinde elde edilebilir ki burada $\phi^{\dagger} = (\phi^{T} \phi)^{-1} \phi^{T}$ Moore-Penrose tersidir ve örneğin tekil değer ayrıştırması ile elde edilebilir.

4. Meta-sezgisel Optimizasyon Yöntemleri (Metaheuristic Optimization Methods)

Birçok alanda önemli uygulamaları olan bilinmeyen sistem tanımlama, özellikle sistem modeli kullanılmadan denetleyici tasarımı gerektiğinde daha da zor bir görev haline gelmektedir. Bilinmeyen sistem tanımlama sürecinde sağlamlıkları, hızlı yakınlaşmaları ve yerel minimum sorunlarından kaçınmaları nedeniyle meta-sezgisel yöntemler tercih edilebilmektedir (Deng ve Liu, 2017). Bu bölüm, bilinmeyen MJS'lerin denetimi için çeşitli meta-sezgisel parametre tasarım metodolojilerine sahip modelleme tabanlı denetleyiciler önermektedir. Tasarım sürecinde, sistem modelinin ve geçişlerinin bilinmediği, ancak zamanla değişen referans sinyalinin bilindiği varsayılmaktadır. Daha sonra verimli bir yığın optimizasyon yöntemi ile denetleyicinin en uygun parametrelerinin bulunması hedeflenmiştir. Şekil 2'de uygulanan yönteme ait blok şema verilmiştir.



Şekil 2. Meta-sezgisel optimizasyon temelli PID denetleyici parametrelerinin ayarlanması (Tuning of PID controller parameters based on meta-heuristic optimization)

Aşağıdaki alt bölümlerde, PID denetleyicisinin en uygun parametrelerini bulmak için çeşitli meta-sezgisel yöntemler özetlenmiştir. Bu yöntemler arasında bulunan, ergen kimliği arama yaklaşımına dayanan AISA optimizasyonu denetleyici tasarımı amacıyla ilk kez bu çalışmada kullanılmıştır.

4.1 Ergen Kimlik Arama Algoritması (Adolescent Identity Search Algorithm - AISA)

Ergen Kimlik Arama Algoritması (Bogar ve Beyhan, 2020), ergenlerin kimlik geliştirme/arama sürecinden esinlenen ve optimizasyon problemlerini çözmek için matematiksel olarak modellenen meta-sezgisel bir optimizasyon algoritmasıdır. Ergenlerin kimlik keşfi sırasında üç farklı davranış ortaya çıkabilir. Bir ergen, akran grubunun davranışını gözlemleyerek ve akıl yürüterek kendi kimliğini oluşturabilir. Ergenlerin akran grubundaki en iyi özellikleri tanımlayıp taklit ettiklerini varsayarsak, en iyi özellikler ortogonal işlev yaklaşımı kullanan bir özellik seçim süreci ile bulunur (Durum 1). Ergenin kimliği, güç düzeyi yüksek bir rol modeli taklit edilerek oluşturulursa, en uygun uygunluk değerine sahip birey akran grubunda rol model olarak seçilebilir (Durum 2). Ergen negatif bir kimlik özelliği benimserse, bu kimlik özelliği algoritmayı keşfedici hale getirmek için kullanılır (Durum 3). AISA 'da kısıtlanmamış tek amaçlı bir tercih

$$\min f(x_1, x_2, \dots, x_n) \underline{b}_j \le x_j \le \overline{b}_j$$
 (12)

şeklinde ifade edilebilir. Burada f(.) hedef işlevdir. x_j terimi j = 1, 2, ..., n olmak üzere j^{inci} karar değişkenidir. \underline{b}_j terimi alt sınır ve \overline{b}_i terimi de üst sınır olmak üzere x_i terimini sınırlandırmaktadır.

AISA'da, çözüm uzayının sınırları içerisinde, N ve n'nin sırasıyla ergenleri ve her ergenin kimliğini temsil ettiği yapay bir akran grubu ile rastgele bir başlangıç popülasyonu (**X**) oluşturulur. Her **x**_jⁱ içinde *i*^{inci} ergen kimlik vektörü ({**x**ⁱ}_{i=1,2,...,N}), kimlik özelliği $x_j^i = \underline{b}_j + U(0,1)_j \times (\overline{b}_j - \underline{b}_j)$ şeklinde olan j^{inci} ergenin kimliğini tanımlar. Burada U(0,1), [0, 1] aralığındaki rastgele bir sayıdır. Daha sonra, Ortogonal Chebyshev Polinomları (OCP) ({ $T_k(X)$ }_{k=0,1,2,...}) kimlikleri eşlemek için kullanılır (Cetin vd., 2019) ve ardından ağırlık parametreleri, regresyon matrisi (**Ψ**) kullanılarak tahmin edilir. Gruptaki bireylerin, $\hat{x}_j^i = 2\frac{x_j^i - \underline{b}_j}{\overline{b}_j - \underline{b}_j} - 1$ olarak ifade edilen bir işlev ile normalleştirilmesi sağlanır. Ardından, regresyon matrisi (**Ψ**) tanımlanır ve ağırlık parametreleri en küçük kareler hata yaklaşımı $\hat{\omega} = (\Psi^T \Psi)^{-1} \Psi^T f$ kullanılarak tahmin edilir. Her bir \hat{x}_j^i T için uygunluk değeri hesaplanır ve kaydedilir. Son olarak, algoritmanın yinelemelerinin her adımında her bir ergenin rastgele bir sekilde önerilen

üç durumdan birini seçtiği varsayılır. AISA'ya ait işlem adımları Algoritma 1'de sunulmuştur.

7

	Algoritma 1. AISA algoritması (AISA algorithm)				
1	N, n, k, m_{in} girdilerini belirle,				
$x_j^i, i = 1,, N, j = 1,, n$ olacak şekilde ergen gruplarını rasgele başlat.					
	Her grubun uygunluk işlev değerini hesapla				
	$t < m_{in}$ YA DA en iyi uygunluk işlevi< ϵ olduğu sürece				
2					
	 Normalize edilmiş giriş matrisi X oluştur, Ağırlık vektörlerini (ŵ) hesapla. 				
	 Kısmi uygunluk matrisi (Ê) tanımla, 				
	• En iyi uygunluk değerine sahip ergen grubunu belirle (x *).				
	<i>Her</i> 1'den N'ye kadar i <i>için</i>				
3	$r_4 \sim U(0,1)$ güncellemesi yap				
	Eğer $r_4 \leq \frac{1}{3}$ ise				
4	• $r_1 \sim U(0,1)$ güncellemesi yap				
	• $x_{yeni}^i = x_i - r_1(x^i - x^*)$				
5	Değil Fakat $r_4 > \frac{1}{3}$ VE $r_4 \le \frac{2}{3}$ ise				
6	• $r_2 \sim U(0.1)$ güncellemesi vap				
	• \mathbf{x}^{rm} rol modelini bul				
	• $p \neq rm$ olacak şekilde rastgele bir ergen seç				
	• $\lambda_{yeni} - \lambda_i = \ell_2(\lambda - \lambda_j)$				
7	Değil ise				
8	• $r_3 \sim U(0,1)^n$ güncellemesi yap				
	• Negatif kimlik vektörü (\mathbf{x}^q) oluştur.				
	• $x_{yeni} = x_i - r_3(x^2 - x^4)$				
9	Son				
10	Sınır değerleri denetimi ve güncelleme mekanizmalarını uygula				
11	Son				
12	Son				

4.2 Balina Optimizasyonu Algoritması (Whale Optimization Algorithm- WO)

Balonlu ağ saldırısını simüle ederek kambur balinaların avlanma davranışlarını taklit eden WO (Mirjalili ve Lewis, 2016), son zamanlarda çok değişkenli karmaşık işlevleri optimize etmek için popüler hale gelmiştir. Kambur balinalar küçük balıkları avlarlar, kabarcık ağı besleme yöntemi olarak bilinen özel bir balık avı türü ile yüzeyi kapatırlar. Bu beslenme yönteminde kabarcıkla ilgili iki manevra vardır: "yukarı doğru spiraller" ve "çift döngüler". İlk manevrada balinalar yaklaşık 12 m derinliğe dalarlar ve yüzeye yüzerek avın etrafında spiral kabarcıklar oluştururlar. Diğer manevra 3 farklı aşamadan oluşur: mercan döngüsü, lobtail ve yakalama döngüsü. (Mirjalili ve Lewis, 2016)'de kambur balinaların balon ağı besleme manevrası matematiksel olarak modellenmiştir. WO'ye göre rastgele değerlerle oluşturulan bir balina grubundaki lider balina (x^*) av aramak için farklı hızlarda ve yönlerde manevralarla yüzer. x^* 'ın hareketlerini taklit eden gruptaki en düşük fonksiyon değerine sahip balina diğer balinalara (x) rehberlik eder. Balinalar hareketlerini ve konumlarını x^* 'dan olan uzaklıklarına göre belirler. Balina optimizasyonu için işlem adımları Algoritma 2'de sunulmuştur (Mirjalili ve Lewis, 2016).

	Algoritma 2. WO algoritmasi (WO algorithm)				
1	\mathbf{x}_i balina grupları vektörlerini başlat				
	Tüm balina gruplarının uygunluk işlevini hesapla ve en iyisini (x^*) bul tanımla				
	t < m _{in} olduğu sürece				
2	<i>Her</i> balina grubu <i>için</i>				
3	a, A, C, l ve p değişkenlerini güncelle				
	<i>Eğer</i> $p < 0.5$ <i>ise</i>				
4	$E \check{g} er A < 1$ ise				
5	$D = Cx^*(t) - x(t) \text{ olmak üzere } x(t+1) = x^*(t) - AD$				
6	Değil Fakat $ A \ge 1$ ise				
	Bir rastgele arama ajanı x _{rand} oluştur				
	$D = Cx_{rand} - x(t) \text{ olmak üzere } x(t+1) = x_{rand} - AD$				
	son				
	Değil Fakat $p \ge 0.5$ ise				
	$D' = x^*(t) - x(t) $ olmak üzere $x(t+1) = D'e^{bl}\cos(2\pi l) + x^*(t)$				
	son				
7	son				
	Herhangi bir balinanın arama alanının ötesine geçip geçmediğini denetle ve düzelt değiştirin				
8	Her balina grubunun uygunluk işlevi değerini belirle $x(t+1)$				
	Eğer daha iyi bir sonuç varsa en iyi balina $x^*(t)$ bilgisini değiştir				
10	son				

WO'da x^* , en iyi çözümün konum vektörüdür. A ve C katsayı vektörleridir. Bu vektörler A = 2ar - a, C = 2r olarak tanımlanır ki burada a doğrusal olarak 2'den 0'a düşürülür ve r, [0,1] cinsinden rastgele bir vektördür. b, logaritmik spiralin şeklini tanımlayan bir sabittir ve l, [-1,1] aralığında rastgele bir sayıdır. m_{in} ve t sırasıyla en çok yineleme sayısı ve geçerli yinelemedir.

4.3 Gri Kurt Optimizasyonu (Grey Wolf Optimization- GWO)

GWO, gri kurtların (Canis lupus) liderlik hiyerarşisinden ve avlanma davranışından ilham alan meta-sezgisel bir yöntemdir. GWO için işlem adımları Algoritma 3'de sunulmuştur (Mirjalili vd., 2014).

	Algoritma 3. GWO algoritması (GWO algorithm)		
1	\mathbf{x}_i gri kurt grupları vektörlerini başlat		
	α, A ve C değişkenlerini başlat		
	Her grubun uygunluk işlev değerini hesapla		
	x_{lpha}, x_{beta} ve x_{δ} bireylerini belirle		
	t < m _{in} olduğu sürece		
2	<i>Her</i> grup <i>için</i>		
3	$x_1 = x_\alpha - A_1 D_{\alpha'} x_2 = x_\beta - A_2 D_\beta \text{ ve } x_3 = x_\delta - A_3 D_\delta \text{ olmak üzere}$ $x(t+1) = \frac{x_1 + x_2 + x_3}{3} \text{ seklinde güncelle}$		
4	son		
5	α , A ve C değişkenlerini güncelle		
	Her grubun uygunluk işlevi değerini belirle		
	x_{lpha}, x_{beta} ve x_{δ} bireylerini güncelle		
6	son		

GWO, en uygun çözüme ulaşmak için her biri aday çözüm olarak kabul edilen dört tür gri kurt (alfa, beta, delta ve omega) kullanır. Avlanma, uyuma yeri, uyanma zamanı vb. konularda karar verici olan alfa'nın kararları sürüye dikte edilir. Betalar, alfa'nın komutlarını güçlendiren, alfa'ya geri bildirim veren ve ona yardım eden ikincil kurtlardır. En düşük rütbeli omega kurtları, baskın kurtların yanında sürünün dengede kalmasına yardımcı olur. İzciler, yaşlılar, avcılar ve bakıcılar kategorisindeki delta kurtları alfalara ve betalara boyun eğerler, ancak omega'ya hükmederler. Algoritmada, avlanma davranışını simüle etmek için üç davranış taklit edilir: av aramak, avı kuşatmak ve ava saldırmak. Algoritmaya göre, α en iyi çözüm olarak kabul edilir. İkinci ve üçüncü en iyi çözümler sırasıyla β ve δ çözümleridir. Kalan aday çözümler ω olarak kabul edilir. Gri kurtlar, avlarını bir grup olarak takip ettikleri ortak bir avlanma stratejisi kullanırlar. Avı kuşatarak ve yörüngesini değiştirerek, onu başarılı bir şekilde yakalama olasılığını artırırlar. Kurtların konumu α değerine göre güncellenir. β kurdu, α 'ya en yakın kurt olarak belirlenir ve liderliği altında bir grup oluşturur. δ kurdu, α ve β 'dan uzakta bir kurt olarak tanımlanır ve bir grup oluşturur. ω kurdu diğer kurtların etrafında dolaşarak yeni aday çözümler üretir. GWO algoritmasında $D_{\alpha} = |C_1 x_{\alpha} - x|, D_{\beta} = |C_2 x_{\beta} - x|, D_{\delta} = |C_3 x_{\delta} - x|. A ve C katsayı vektörleri şu şekilde hesaplanır: <math>A = 2ar_1 - a, C = 2r_2$. Yinelemeler sırasında, a bileşenleri doğrusal olarak 2'den 0'a düşerken, r_1 ve r_2 [0,1] aralığında rastgele vektörlerdir.

4.4 Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm - ABC)

ABC algoritması (Karaboga ve Basturk, 2007), çok değişkenli fonksiyonları optimize etmek için yapay arı kolonilerinin beslenme davranışından esinlenen basit ve etkili bir meta-sezgisel yaklaşımdır. Algoritma, kolonideki arıların beslenme, dans etme ve gezinme davranışlarına benzer arama yaparak arama alanında en iyi çözümü bulmayı amaçlamaktadır. Yapay arı kolonisinde, besin kaynağını seçen seyirci arılar, daha önce ziyaret edilen besin kaynağına giden istihdam edilen arılar ve rastgele arama yapan keşif arıları vardır. Algoritma başlangıcında bir yapay arı kolonisi oluşturulur, rastgele bir dizi besin kaynağı konumu seçilir ve miktarları belirlenir. İstihdam edilen arılar rastgele seçilmiş bir çözüm etrafında çalışır ve o noktaya yakın başka aday çözümler arar. Keşfedilen besin kaynağı, önceki konumdaki çalışan arının kaynağından fazlaysa, çalışan arı eski konumunu unutur ve yeni konumu günceller. Bu bilgi kovanda bekleyen seyirci arılarla paylaşılır. İzleyici arı hem açgözlü bir seçim yaklaşımıyla besin kaynağını seçer hem de miktarını denetler. Besin kaynağının konumu bir sınır toleransı ile daha da iyileştirilemezse, o besin kaynağının terk edildiği varsayılır. Arılar bir besin kaynağını terk ederse, izci arı rastgele yeni bir tane seçer. ABC algoritmasının her döngüsü, yeni bir besin kaynağı aramak için en fazla bir izci kullanır. ABC algoritmasında v_i yeni aday çözümdür ve ϕ_i [-1,1] cinsinden tekdüze bir rasgele sayıdır.

Algoritma 4. ABC algoritması	(ABC algorithm)
------------------------------	-----------------

```
Başlangıç girdilerini belirle;
```

- - ...

```
m_{in}:en fazla tekrarlama adedi, x_1, \dots, x_N: arı kolonilerini x_i = x_{min} + rand(0,1)(x_{max} - x_{min}) şeklinde oluştur
```

Her grubun uygunluk işlev değerini hesapla

	t < m _{in} olduğu sürece			
2	<i>Her</i> işçi arı <i>için</i>			
3	 v_i = x_i + φ_i(x_i - x_k) olmak üzere yeni v_i çözümlerini oluştur Her v_i için uygunluk işlevini f(v_i) hesapla x_i ve v_i arasında açgözlülüğe dayalı eleme mekanizmasını uygula 			
4	son			
5	$p_i = rac{f_i}{\sum_{n=1}^{\Delta N} f_n}$ ile olasılık değerini hesapla			
	<i>Her</i> seyirci arı <i>için</i>			
6	<i>Eğer</i> $rand < p_i$ <i>ise</i>			
7	 v_i = x_i + φ_i(x_i - x_k) olmak üzere yeni v_i çözümlerini oluştur Her v_i için uygunluk işlevini f(v_i) hesapla x_i ve v_i arasında açgözlülüğe dayalı eleme mekanizmasını uygula 			
8	son			
9	son			
10	İzci Arı fazı			
	Eğer izci için ayrı bir çözüm var ise			
11	Bu çözümü, $x_i = x_{min} + rand(0,1)(x_{max} - x_{min})$ şeklinde rastgele oluşturulan bir çözümle			
12	son			
13	son			

4.5 Parçacık Sürüsü Optimizasyonu (Particle Swarm Optimization - PSO)

PSO, kuş sürüleri, balıklar veya böcek sürüleri gibi belirli hayvanların kolektif davranışlarından ilham alan güçlü bir meta-sezgisel algoritmadır (Kennedy ve Eberhart, 1995). PSO algoritmasının prensibi, nesnel bir fonksiyonun optimizasyonunu ve arama alanındaki bir grup parçacığın hareketi ile en iyi çözümü bulmayı içerir. PSO'daki her parçacık, optimizasyon probleminin potansiyel bir çözümünü temsil eder ve nesnel işleve göre değerlendirilen konumu, hızı ve uygunluk değeri ile karakterize edilir. Her yinelemede bir parçacık, kendisinin ve sürünün en iyi çözümlerini kaydederek küresel optimum çözümü bulmaya çalışır. *D* boyutlu arama alanında rastgele oluşturulmuş bir gruptaki *N* adet parçacıktan her birinin ($\mathbf{x}_i = [x_{i1}, ..., x_{iD}]$) şeklinde bir konum vektörü ve ($\mathbf{v}_i = [v_{i1}, ..., v_{iD}]$) şeklinde bir hız vektörü vardır. g_{best} koordinatı, sürünün en iyi çözümünün bulunduğu arama alanındaki konumu tanımlar. Her yinelemede, uygunluk işlevi, her parçacığın önceki en iyi konumu p_{best_i} için değerlendirilir. Karşılık gelen yerel konum güncellemesi, parçacığın mevcut konumuna ve hızına göre belirlenir. Her parçacık, tüm parçacıklar arasında en iyi uygunluk değerini seçerek tüm sürüde en iyi küresel konum güncellemesinin belirlenmesine yol açar. Yeni konumlar ve hızlar güncellendikten sonra, her parçacığın uygunluk değeri yeniden hesaplanır. PSO'nun çıktısı, en iyi uygunluk değerinden ve buna karşılık gelen en iyi konumdan oluşur. PSO için belirtilen işlem adımları Algoritma 5'te sunulmuştur.



5. Uygulama Sonuçları (Application Results)

Bu çalışmada, bir RLC devresi, matematiksel modelinin bilinmediği ve parametresinde rastgele geçişlerin meydana geldiği düşünülen bir ELM modeli ile tanımlanmıştır. Ardından, yaklaşık sistem modelinin elde edildiği bu devreyi denetlemek için geleneksel bir PID kontrol cihazı tasarlanmıştır. Ek olarak, çeşitli meta-sezgisel parametre tasarım metodolojileri kullanılarak modelleme tabanlı denetleyiciler önerilmiştir.

5.1 Markov davranışlı RLC Devresinin ELM ile Tanımlanması (ELM Identification of Markovian RLC Circuit)

RLC devresinin şematik gösterimi ve gerçek zamanlı deney kurulumu sırasıyla Şekil 3(a) ve Şekil 3(b) 'de gösterilmiştir. Devrenin matematiksel modeli aşağıdaki dinamiklerle verilebilir:

$$\begin{aligned} \dot{x}_{1}(t) &= \frac{-x_{1}(t)}{C^{\theta(t)}R_{o}^{\theta(t)}} + \frac{x_{2}(t)}{C^{\theta(t)}} + d_{!}(t), \\ \dot{x}_{2}(t) &= \frac{-x_{1}(t)}{L} - \frac{R_{i}^{\theta(t)}x_{2}(t)}{L} + \frac{u(t)}{L} + d_{2}(t) \end{aligned}$$
(13)

Burada u(t) denetim işaretini temsil eder, $x_1(t)$ kapasitans voltajıdır ($v_c(t)$) ve $x_2(t)$ endüktans akımıdır ($i_L(t)$). Tasarlanan devre için denetim işareti $u_{min} = 0$ V, $u_{max} = 2$ V aralığında ve örnekleme süresi de $\tau_{min} = 0$ ms, $\tau_{max} = 1$ ms aralığında tutulmuştur. Ayrıca, $d_1(t)$ ve $d_2(t)$ parametreleri, parametre belirsizlikleri ve bilinmeyen bozucu etkiler vb. nedeniyle toplam belirsizlik işlevlerini ifade etmektedir. Denklem 13'teki sistem, (Vargas vd., 2017)'deki gibi üç Markov atlama sistemi elde etmek için her $t \ge 0$ için $\theta(t) = 1,2,3$ geçiş değişkenine bağlı olarak üç farklı mod geçişine sahiptir. Bu devrede L endüktans değeri sabit olacak şekilde 10 mH olarak seçilirken, dirençlerin değerleri $R_i^{\theta(t)} = \{10 \ \Omega, 5 \ \Omega, 3.3 \ \Omega\}$, $R_o^{\theta(t)} = \{100 \ \Omega, 50 \ \Omega, 33.3 \ \Omega\}$ ve kondansatör $C^{\theta(t)} = \{10 \ \mu f, 110 \ \mu f, 440 \ \mu f\}$. Devre alt sistemleri arasındaki geçişleri üretmek için, sürekli zamanlı Markov RLC devresi için geçiş olasılık matrisi şu şekilde kullanılmıştır:

$$P = \begin{array}{cccc} -0.2936 & 0.2016 & 0.0920 \\ P = 0.2758 & -0.5634 & 0.2876 \\ 0.1127 & 0.2343 & -0.3470 \end{array}$$
(14)



(b) Gerçek-zamanlı deney düzeneği (Real-time set up)

Sekil 3. Markovian RLC devresinin deneysel kurulumu (Experimental set-up of the Markovian RLC circuit)

Tasarım aşamasında bileşenlerin tüm değerleri bilinmesine rağmen, ilgili belirsizlikler nedeniyle RLC devresinin dinamikleri bu çalışmada bilinmeyen olarak kabul edilmektedir. Bu nedenle, PID parametre optimizasyonu için gerçek zamanlı sistem yerine bir ELM modeli kullanılmıştır. Verimli bir ELM modeli elde etmek için önce devreye $T_{son} = 100$ s olacak şekilde rastgele bir giriş sinyali uygulanır ve sistemin çıkışı $T_s = 1$ ms örnekleme periyodunda ölçülmüş ve ELM modelin eğitimi için gerekli giriş-çıkış verileri hazırlanmıştır. Giriş sinyalinin genliği, Şekil 4(a)'da gösterildiği gibi sistemin tüm dinamik özelliklerini ortaya çıkarmak için keyfi olarak değişen zaman dilimleri için keyfi olarak 0-2 V olarak değişmektedir. Sistem modelinin, Denklem 7'deki gibi NARX modeli biçiminde olduğu varsayılmıştır. Bu giriş-çıktı veri çiftleri, ELM modelinin girdilerini Denklem 9'daki gibi oluşturmak için deneysel olarak seçilen $n_u = 3$ ve $n_y = 3$ için yeniden düzenlenmiştir. Ardından, ELM modelinin rastgele giriş ağırlıkları ve ofset değerleri kullanılarak çıkış katmanı ağırlıkları tahmin edilmiştir. Daha sonra, en iyi performansa sahip yaklaşım modeli, RLC devresinin dinamiklerini temsil eden son ELM modeli olarak seçilmiştir. ELM modelinin test verilerinin 0-10 s aralığındaki test performansı Şekil 4(b)'de gösterilmiştir. Ayrıca ELM modelinin gerçek zamanlı RLC devresi ile hemen hemen aynı çıkış değerlerini ürettiğini de ortaya koyulmuştur. Sonuç olarak, test verilerinin modelleme performansı Ortalama Karesel Hata (Mean Square Error - MSE) cinsinden 1.3092 × 10⁻⁴ olarak hesaplanmıştır.





(a) Uygulanan giriş işareti (Applied control signal)
 (b) Modelleme başarımı (Modelling performance in (0-10)s)
 Şekil 4. Gerçek zamanlı veri temelli ELM modelleme (Real-time data based ELM modelling)

5.2 Optimizasyon Temelli PID Denetleyicisi Tasarımı (Optimization Based PID Controller Design)

Önceki bölümde detaylandırılan meta-sezgisel optimizasyon algoritması AISA, PID kontrol cihazının parametrelerinin optimal değerlerini bulmak için kullanılmıştır. Optimizasyon ile çözülmesi istenen hedef işlev

$$\min_{\theta} F(e) = 0.5 e^2 = 0.5 (y_{ref} - y)^2 \quad (15)$$

 $0 < \theta < 1$

şeklinde Lyapunov fonksiyonu olarak seçilmiştir. Burada y_{ref} referans işaretini, y aktif nüfusun bir üyesi kullanılarak oluşturulan PID denetleyicisi tarafından üretilen giriş işareti ile manipüle edilen sistem çıkışını ve e bunlar arasındaki farkın oluşturduğu izleme hatasını ifade etmektedir. $\theta = [K_P, K_I, K_D]$ ise şeklinde olup K_P, K_I ve K_D , Denklem 16 ile verilen PID denetleyicisinin parametreleridir. θ parametresine ait alt ve üst sınırlar ızgaraarama (grid-search) yöntemi ile deneysel olarak elde edilmiştir.

$$u[k+1] = u[k] + K_P(e(k) - e(k-1)) + K_Ie(k) + K_D(e(k) - 2e(k-1) + e(k-2))$$
(16)

AISA optimizasyonunun performansı, literatürde en çok tercih edilen meta-sezgisel optimizasyon algoritmalarından biri olan PSO (Kennedy ve Eberhart, 1995) ile karşılaştırılmıştır. PSO'ya ek olarak, literatürde popüler meta-sezgisel optimizasyon algoritmaları olan ABC algoritması (Karaboga ve Basturk, 2007), GWO (Mirjalili vd., 2014) ve WO (Mirjalili ve Lewis, 2016)'de karşılaştırıma için kullanılmıştır. Algoritmalar, 16 GB RAM bellek ile 2,60GHz'de Intel (R) Core (TM) i7-5600U merkezi işlem birimine sahip bir kişisel bilgisayarda çalıştırılmıştır.

Optimizasyon sürecinde, referans sinyali ile sistemin çıkışı arasındaki izleme hatasını grubun bir üyesinin başarımını ortaya çıkaran bir uygunluk değeri olarak hesaplamak için bir sistem modeli gereklidir. Bu amaçla, dinamikleri bilinmediği varsayıldığı için gerçek zamanlı RLC devresi yerine tasarlanmış ELM modeli kullanılmıştır. ELM modelinin cıktısı, meycut gruptaki birevlerle oluşturulan aday PID denetim cihazı tarafından manipüle edilen giriş işareti ile hesaplanmaktadır. AISA yöntemi istenen parametre optimizasyonunu sağlamak üzere çalışmaya başlamak için popülasyon miktarı, yapılacak en fazla iterasyon adedi, optimize edilecek parametre sayısı ve bu sayı kadar alt ve üst sınır bilgisinin baştan belirlenmesine ihtiyaç duymaktadır. Bu nedenle karşılaştırma amacıyla kullanılan yöntemler de bu özelliklere sahip olacak şekilde seçilmişlerdir. ABC, GWO ve WO algoritmaları da benzer özelliktedir. Bunlar dışında literatürdeki referans algoritmalar arasında gösterilebilecek olan PSO algoritması ise belirtilen temel başlangıç atamaları haricinde v_{min} , v_{max} , c_1 , c_2 gibi başka başlangıç parametreleri tanımına ihtiyaç duymaktadır. PSO'ya ait bu parametreler hedef çalışmaya uymak amacıyla standart ve literatürde kabul görmüş değerleriyle kullanılmıştır. Bu çalışmada kullanılan algoritmaların arama sürecini aynı koşullar altında başlatmasını sağlamak için öncelikle PID denetleyicinin parametreleri olan K_P, K_I ve K_D için alt ve üst sınırlar $b_{lower} = [0,0,0], b_{upper} = [1,1,1]$ ile belirlenen tarama uzayında nPop = 100 üyeli bir rastgele bireyler grubu matrisi oluşturulmuş ve her algoritma başlangıç aday grubu olarak bu grubu kullanacak şekilde başlatılmıştır. Bu sınır değerlerinin seçilmesi için literatürce bilinen ızgara arama (grid-search) algoritması kullanılmıştır. Bu algoritma sonucunda K_P , K_I ve K_D için en uygun aralık belirtilen şekilde hesaplanmıştır. Durdurma şartı olarak en çok yineleme sayısı maxiter = 500 ve en çok üretim duraklama adedi (maximum stall generation) maxStallGen = 50 olarak seçilmiştir. Her algoritmanın temel arama sürecinde rastgelelik içeren işlemler kullanması nedeniyle, başlangıç nüfusu aynı olsa bile algoritmalar her yeni çalışmada farklı bir en iyi uygunluk değeri $(F(e)_{best})$ bulabilmektedir. Bu nedenle, çalışmanın bu aşamasında, durdurma şartlarından en az birine ulaşılıncaya kadar her algoritma aynı ayarlarla RunNo = 50 kez çalıştırılmıştır. Ardından, bu süreç içinde elde edilen $F(e)_{best}$, bu uygunluk değerine ulaşılıncaya kadar geçen sn olarak süre $(T(F(e)_{best}))$ ve süreç içinde uygunluk değerinin hesaplanma adedini ifade eden (Cost Function Execution Count - $CFEC(F(e)_{hest})$) değerleri kaydedilmiş ve Tablo 1'de sunulmuştur. Benzetim çalışmaları sonucunda en küçük uygunluk değeri 10⁻⁹ hassasiyetle, $F(e)_{best} = 2.726953388$ olarak bulunmuştur. Algoritmalar RunNo = 50 olacak şekilde durdurma şartları dahilinde çalıştırılırken $F(e)_{best}$ değerine GWO ve WO ulaşamamışken, AISA 15, PSO 35 ve ABC 43 kez ulaşabilmiştir. Buna rağmen $F(e)_{best}$ değerine ulaşılırken geçen hesaplama sürelerine ait en iyi zaman değeri $T(F(e)_{best})$ AISA'ya aittir. Ayrıca $F(e)_{best}$ değerine ulaşılırken uygunluk değeri işlevinin hesaplama adedini gösteren *CFEC*(*F*(*e*)_{best}) için de AISA, en iyi değere sahip olan PSO algoritmasına çok yakın bir değere sahiptir. Bu durum Tablo 1'de belirtilmistir. Bu sonuçlar yakın zamanda önerilen AISA yönteminin, dinamikleri bilinmediği varsayılan ve ELM ile modellenen bir MJS'nin zamanla değişen bir referans işareti için çıkış denetimi yapacak bir PID denetleyicisinin parametrelerinin bulunmasında yeterli başarıma sahip olduğunu göstermektedir.

ELM modeli tasarlanırken, sistemin iç dinamiklerini harekete geçirmek için rastgele giriş sinyali uygulanmıştır. Bu nedenle, tasarlanan doğrusal olmayan ELM modeli, RLC dinamiğinin tüm olası Markov modlarını kapsamaktadır. Bu model, yığın optimizasyon içerisinde çıkış işaretini üreten sistem modeli olarak kullanılmış ve model çıkışının zamanla değişen bir referans işaretini izlemesini sağlayacak PID denetleyicisinin parametreleri de AISA optimizasyonu ile $K_P = 0.45920349$, $K_I = 0.14835754$ ve $K_D = 0.07168092$ olarak bulunmuştur. Şekil 5(a) ve Şekil 5(b), sırasıyla yığın parametre optimizasyon durumu için en iyi izleme başarımını ve uygulanan denetim işaretini göstermektedir. Optimum PID denetleme cihazı, Markov geçişlerinin etkisi altında yeterli bir denetim başarımına sahiptir. Örnek olarak, kontrol edilen sistemin t = 20 sn anındaki izleme performansı, Şekil 5(a)'daki küçük bir pencere tarafından odaklanmıştır.

Benzetim çalışmalarından sonra, AISA tarafından elde edilen PID denetleme cihazının optimum tasarımı, üç modlu bir MJS gibi davranan gerçek zamanlı RLC devresinin denetiminde kullanılmıştır. Ek olarak, PID kontrol cihazının parametreleri, karşılaştırma için geleneksel Z-N ayarlama yöntemi kullanılarak sırasıyla $K_p = 0.5$, $K_i = 0.25$, $K_d = 0.074$ olarak elde edilir. Bu PID girdileri ile yapılan denetimde tatmin edici bir performans seviyesi sağlamadığından uygulama sonuçları verilmemiştir. Optimum PID denetleyicisi, gerçek-zamanlı deneyler için Şekil 6'da gösterilen yeterli izleme performansına sahiptir. Şekil 6(a), kapasitör voltajıyla ilgili referans izlemeyi göstermektedir. Şekil 6(b), Şekil 6(c)'de belirtilen olasılık geçişlerine göre uygulanan denetleme gerilimlerini göstermektedir. Markov geçişlerinin olumsuz etkilerine rağmen, gerçek zamanlı RLC sistemi, AISA tarafından elde edilen optimum PID denetleyicisi tarafından uygun şekilde manipüle edilmiştir. Bunu daha iyi gösterebilmek için, t = 30 sn ve t = 90 sn 'daki mod geçişleri, Şekil 6(a)'daki küçük pencereler tarafından odaklanmıştır. Geçiş gerçekleştiğinde, gerçek zamanlı sistem dinamiği ve matematiksel model tam olarak uyuşmadığından simülasyonlardakinden daha fazla salınımın var olduğu görülmektedir. Performans ölçümleri Tablo 2'de verilmiştir, burada bilinmeyen MJS'lerin denetimi için yığın optimizasyon tabanlı ayarlama yönteminin geçerli olduğu görülmektedir. Gerçek zamanlı deneylerin başarım ölçümleri, karesel hataların ortalamasının karekökü (Root Mean Square Error - RMSE) olarak hesaplanmıştır.

Tablo 1. Yiğin optimizasyon sonuçları (Batch optimization results)					
Başarım	AISA	PSO	ABC	GWO	WO
$F(e)_{best}$	2.726953388	2.726953388	2.726953388	2.726953480	2.726981494
$T(F(e)_{best})(sn)$	163,4195	180,1056	311,6661	124,8971	31,5652
$CFEC(F(e)_{best})$	50100	49600	100274	35900	9300

 Metrikler
 Performances



Şekil 5. Yığın girdi tasarımı benzetim sonuçları, a) İzleme başarımı, b) Uygulanan denetim işareti, c) Mod geçişleri (Batch parameter design simulation results, a) Control results, b) Applied control signal, c) Mode transitions)



Şekil 6. Yığın girdi tasarımı gerçek-zamanlı sonuçlar, a) İzleme başarımı b) Uygulanan denetim işareti c) Mod geçişleri (Batch parameter design real-time results, a) Control results, b) Applied control signal, c) Mode transitions)

6. Tartışma ve Sonuçlar (Discussion and Conclusion)

Her kontrol mühendisinin bildiği gibi, sürekli Markov geçişleri ve zamanla değişen referans sinvalleri ile karşılaşıldığında Z-N yöntemi ve parametrelerin manuel olarak ayarlanması maalesef yetersizdir. Markov davranıslı atlama sistemleri, rastgele gecişlerin ve çevresel gürültülerin neden olduğu belirsiz dinamikler nedeniyle kontrol edilmesi zor süreçlerdir. Bu çalışmada, dinamiği bilinmeyen bir RLC devresi üzerinden sunulan Markov Atlama Sisteminin çıkışını kontrol etmek için PID denetleyici parametrelerinin ayarlanması, yığın optimizasyon yöntemi ile çözülmüştür. Bu nedenle, birincil adım olarak, öngörülemeyen üç Markov sıçramasının gerçekleştiği RLC devresinden temel giriş-çıkış verileri elde edilir. Bu veriler, zaman ve genlik bakımından rastgele değişen giriş sinyallerine karşılık gelen çıkış değerleri ile MJS'nin tüm dinamiklerini kapsamlı bir şekilde temsil etmektedir. Daha sonra, gerçek zamanlı RLC devresi yerine yığın optimizasyon sürecindeki simülasyonlarda kullanılmak üzere bu verilere dayanarak daha az karmaşıklığa sahip bir ELM modeli oluşturulmuştur. Bu bağlamda, optimum PID parametreleri AISA optimizasvon vöntemi kullanılarak belirlenmis ve PSO, GWO, WO, ABC gibi literatürde sıklıkla tercih edilen veva nispeten veni meta-sezgisel algoritmalarla karsılaştırılmıştır. Çalışmanın birincil katkısı, gerçek zamanlı bir RLC devresinden toplanan gerçek zamanlı verileri kullanarak kara kutu tanımlaması yapılmış olmasıdır. İkinci katkı ise, bilinmeyen bir modele sahip bir MJS'yi denetlemek için AISA optimizasyon yöntemini kullanarak ilk kez optimal bir PID denetleme cihazı tasarlanmış olmasıdır. Simülasyon sonuçları, AISA optimizasyon yönteminin kabul edilebilir bir performansa sahip olduğunu göstermiştir. Bu nedenle. Markov davranisi sergileven bir gercek zamanlı RLC devresini denetlemek için parametreleri AISA ile elde edilen PID denetim cihazı kullanılmıştır. Gerçek zamanlı denetleme sonuçları, tasarlanan yöntemlerin verimliliğini doğrulamış ve gelecekteki uygulamalarda bilinmeyen MJS'lerin denetimini sağlamak üzere modellemeye dayalı PID optimizasyonunun verimliliğini göstermiştir.

Çıkar Çatışması (Conflict of Interest)

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir. No conflict of interest was declared by the authors.

Teşekkür (Acknowledgement)

Bu makale Pamukkale Üniversitesi Bilimsel Araştırma Projeleri Konseyi tarafından 2021HZDP021 numaralı hibe kapsamında finanse edilmiştir.

Kaynaklar (References)

Abd-Elazim S.M., Ali E. S., 2018. Load frequency controller design of a two-area system composing of PV grid and thermal generator via firefly algorithm. Neural Computing and Applications, 30, 607-616.

Ang K.H., Chong G., Li Y., 2005. PID control system analysis, design, and technology, IEEE Transactions on Control Systems Technology, 13(4), 559-576.

Astrom K.J., Hagglund T., 1995. PID controllers: theory, design, and tuning, ISA-The Instrumentation, Systems and Automation Society.

Beyhan S., Alci M., 2010. Stable modeling-based control methods using a new RBF network, ISA Transactions, 49(4), 510-518.
Bogar E., Beyhan S., 2020. Adolescent identity search algorithm (AISA): A novel metaheuristic approach for solving optimization problems, Applied Soft Computing, 95, 106503.

Cetin M., Bahtiyar B., Beyhan S., 2019. Adaptive uncertainty compensation-based nonlinear model predictive control with realtime applications. Neural Computing and Applications, 31, 1029-1043.

Cetin M., Iplikci S., 2015. A novel auto-tuning PID control mechanism for nonlinear systems, ISA Transactions, 58, 292-308.

Cetintas G., Hamamci S.E., 2022. Proportional-integral-derivative stabilization of complex conjugate-order systems, Transactions of the Institute of Measurement and Control, 44(15), 2941-2952.

- Chang W.D., Yan J.J., 2005. Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems, Chaos, Solitons and Fractals, 26(1), 167-175.
- Chiou J.S., Tsai S.H., Liu M.T., 2012. A PSO-based adaptive fuzzy PID-controllers, Simulation Modelling Practice and Theory, 26, 49-59.
- Deng L., Liu S., 2023. A novel hybrid grasshopper optimization algorithm for numerical and engineering optimization problems, Neural Process Letters, 1-55.
- Fang M., Shi P., Dong S., 2019. Sliding mode control for Markov jump systems with delays via asynchronous approach, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(5), 2916-2925.
- Hasan M.W., Abbas N.H., 2022. Disturbance rejection for underwater robotic vehicle based on adaptive fuzzy with nonlinear PID controller, ISA Transactions, 130, 360-376
- Hekimoğlu B., 2019. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. IEEE Access, 7, 38100-38114.
- Huang G.B., Zhu Q.Y., Siew C.K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks, IEEE International Joint Conference on Neural Networks, 2, 985-990.
- Huang G.B., Zhu Q.Y., Siew C.K., 2006. Extreme learning machine: theory and applications, Neurocomputing, 70(1-3), 489-501.
- Izci D., Ekinci S., 2023. Optimizing Three-Tank Liquid Level Control: Insights from Prairie Dog Optimization. International Journal of Robotics & Control Systems, 3(3).
- Joseph S.B., Dada E.G., Abidemi A., 2022. Oyewola D.O., Khammas B.M. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. Heliyon, 8(5), e09399.
- Kang J., Meng W., Abraham A., Liu H., 2014. An adaptive PID neural network for complex nonlinear system control, Neurocomputing, 135, 79-85.
- Kang Y., Zhao Y.B., Zhao P., 2018. Stability analysis of Markovian jump systems. Springer.
- Karaboga D., Basturk B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization, 39(3), 459-471.
- Kennedy J., Eberhart R., 1995. Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks, 4, 1942-1948.
- Lennartson B., Kristiansson B., 2009. Evaluation and tuning of robust PID controllers, IET Control Theory and Applications, 3(3), 294-302.
- Mirjalili S., Lewis A., 2016. The whale optimization algorithm, Advances in Engineering Software, 95, 51-67.
- Mirjalili S., Mirjalili S.M., Lewis A., 2014. Grey wolf optimizer, Advances in Engineering Software, 69, 46-61.
- Özay C.A.N., Eroğlu H., Öztürk, A., 2022. FV-termal güç sistemlerinde balina optimizasyon algoritması tabanlı otomatik üretim kontrolörü. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 38(2), 915-926.
- Paul S., Yu W., Li X., 2018. Bidirectional active control of structures with type-2 fuzzy PD and PID, International Journal of Systems Science, 49(4), 766-782.
- Qi Z., Shi Q., Zhang H., 2019. Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays. IEEE Transactions on Industrial Electronics, 67(7), 5637-5646.
- Saravanakumar R., Ali M.S., 2022. Extended dissipative criteria for generalized Markovian jump neural networks including asynchronous mode-dependent delayed states, Neural Process Letters, 54, 1623-1645.
- Van M., 2018. An enhanced robust fault tolerant control based on an adaptive fuzzy PID-nonsingular fast terminal sliding mode control for uncertain nonlinear systems, IEEE/ASME Transactions on Mechatronics, 23(3), 1362-1371.
- Vargas A.N., Pujol G., Acho L., 2017. Stability of Markov jump systems with quadratic terms and its application to RLC circuits, Journal of the Franklin Institute, 354(1), 332-344.
- Vincent A.K., Nersisson R., 2017. Particle swarm optimization based PID controller tuning for level control of two tank system. In IOP Conference Series: Materials Science and Engineering, 263, (5).
- Zhang X., Zhang G., Yin Y., He S., 2021. Asynchronous sliding mode dissipative control for discrete-time Markov jump systems with application to automotive electronic throttle body control system, Computers and Electrical Engineering, 96, 107496.
- Zhu M., Chen Y., Kong Y., Chen C., Bai J., 2020. Distributed filtering for Markov jump systems with randomly occurring one-sided Lipschitz nonlinearities under Round-Robin scheduling, Neurocomputing, 417, 396-405.
- Ziegler J.G., Nichols N.B., 1942. Optimum settings for automatic controllers, Transaction ASME, 64(11), 759-768.