

# Sezgisel Algoritmalar Kullanılarak Bulanık Mantık Denetleyici Optimizasyonu

## Optimization of Fuzzy Logic Controller by Using Heuristic Algorithms

Sinan ÜNSAL<sup>1</sup>, İbrahim ALIŞKAN<sup>1</sup>

<sup>1</sup>Elektrik-Elektronik Mühendisliği Bölümü, Bülent Ecevit Üniversitesi, Zonguldak, Türkiye  
{sinanunsal, ialiskan}@beun.edu.tr

**Özetçe**—Bulanık mantık denetleyicilerin yapısında birçok tasarım parametresi bulunmaktadır. Bu parametrelerin belirlenmesinde çoğunlukla sistematik bir yaklaşıma sahip olmayan geleneksel yöntemler kullanılmaktadır. Ancak, denetleyici parametrelerinin bu şekilde ayarlanması uzun denemelere yol açmakta ve bu durum oldukça fazla zaman almaktadır. Bu nedenle, bulanık mantık denetleyici tasarım parametrelerinin belirlenmesinde genellikle sezgisel algoritmalar kullanılmaktadır. Çünkü; sezgisel algoritmalar kesin çözümün elde edilemediği problemlerde optimum çözümün çok yakınında çözümler sunabilen algoritmalarlardır. Ele alınan çalışmada, bir bulanık mantık denetleyicinin çıkış üyelik fonksiyonları parçacık sürüsü optimizasyonu ve genetik algoritma kullanılarak optimize edilmiştir. Tasarım ve optimizasyon aşamaları detaylı bir şekilde anlatılarak elde edilen sonuçlar birbiriyle karşılaştırılmıştır.

**Anahtar Kelimeler**—Bulanık Mantık; Genetik Algoritma; Parçacık Sürüsü Optimizasyonu; Sezgisel Algoritmalar.

**Abstract**—There are many design parameters in the structure of fuzzy logic controllers. Conventional methods that don't have a systematic approach are often used in determining of these parameters. However, setting the controller parameters in this way leads to long experiments and this takes a lot of time. For this reason, design parameters of the fuzzy logic controller are usually determined by using heuristic algorithms. Because, heuristic algorithms can offer solutions that are very close to the optimal solution for the problems where exact solution cannot be obtained. In this study, output membership functions of a fuzzy logic controller are optimized using particle swarm optimization and genetic algorithm. Design and optimization stages are explained in detail and results are compared with each other.

**Keywords**—Fuzzy Logic; Genetic Algorithm; Particle Swarm Optimization; Heuristic Algorithms.

### I. GİRİŞ

Bulanık mantık denetleyicilerin geleneksel yöntemlerle tasarımında geliştirilmiş bir yöntem bulunmamaktadır. Üzerinde çalışılan konuda uzman olan kişinin bilgisi ve becerisi tasarımın en önemli parçasıdır. Geleneksel yöntemler kullanılarak yapılan tasarımda, başlangıçta tasarımı yapılan denetleyici parametreleri istenilen performans seviyesine ulaşılan kadar değiştirilerek daha iyi değerlere ayarlanmaktadır. Uzman kişilerin tecrübelerinden yararlanarak denetleyici parametrelerinin değerleri belirlenebilir. Ancak, çalışılan konuda uzman kişinin bulunması her zaman mümkün olmadığı gibi uzman kişinin çözüm konusunda ne kadar başarılı olabileceği de belirsizdir. Ayrıca, bulanık mantık denetleyici parametrelerinin belirlenmesini doğrudan uzman bilgisine bırakmak denetleyici tasarımında yetersiz kalabilmektedir. Bu durum, bulanık mantık denetleyicilerin etkin olarak kullanımında önemli bir sorun oluşturmaktadır. Tasarım sürecinde karşılaşılan bu problemlere çözüm olarak bulanık mantık denetleyici parametrelerinin belirlenmesinde çoğunlukla sezgisel algoritmalar kullanılmaktadır. Yapay sinir ağları, genetik algoritma, diferansiyel gelişim algoritması, karınca kolonisi algoritması, arı kolonisi algoritması ve parçacık sürüsü optimizasyonu popüler sezgisel algoritmalar arasında bazılarıdır. Bu algoritmalar bulanık mantık denetleyici optimizasyonunda birçok araştırmacı tarafından kullanılmıştır. Sezgisel algoritmalar kullanılarak üyelik fonksiyonlarının optimize edilmesi [1-5], bulanık mantık denetleyici kural tablosunun optimizasyonu [6-8], ölçekleme faktörlerinin optimizasyonu [9-11] ve optimal denetleyici tasarımının yapılması [12-16] bu çalışmalardan bazılarıdır.

Ortaya konulan çalışmada, bulanık mantık denetleyici çıkış üyelik fonksiyonları sezgisel algoritmalar içerisinde yer alan parçacık sürüsü optimizasyonu ve genetik algoritma ile optimize edilmiştir. Optimizasyon işlemleri farklı hata performans endeksleri kullanılarak gerçekleştirilmiştir.

## II. SEZGİSEL ALGORİTMALAR

Sezgisel algoritmalar, herhangi bir problemi çözmek veya bir hedefe ulaşmak için doğadaki olaylardan esinlenilerek geliştirilen algoritmalar. Bu algoritmalar, çözüm uzayında optimum çözümün çok yakınındaki çözümleri garanti edebilmektedir [17-18]. Sezgisel algoritmalar yapılarına göre biyolojik tabanlı, fizik tabanlı, kimya tabanlı, müzik tabanlı, sosyal tabanlı ve sürü zekası tabanlı olmak üzere altı temel grupta değerlendirilmektedir [18]. Bu çalışmada bulanık mantık denetleyicinin optimizasyonunda biyolojik tabanlı yöntemlerden birisi olan genetik algoritma ile sürü zekası tabanlı yöntemlerden birisi olan parçacık sürüsü optimizasyonu kullanılmıştır.

### A. Parçacık Sürüsü Optimizasyonu

Parçacık sürüsü optimizasyonu (PSO) algoritması, balık ve kuş sürülerinin sosyal davranışlarından esinlenilerek geliştirilen bir sezgisel optimizasyon yöntemidir [19]. Balık ve kuş sürülerinin yiyecek ararken etkileşim içerisinde oldukları görülmüştür. Sürüdeki bireylerden birisi yiyecek bulduğunda sürünün diğer bireyleri sürü içerisindeki etkileşim yardımıyla yiyeceğin olduğu yöne doğru yönelmektedirler [19]. PSO algoritması temel olarak aşağıdaki adımlardan oluşur;

- Rastgele üretilen başlangıç pozisyonları ve hızları ile başlangıç sürüsü oluşturulur.
- Sürü içerisindeki tüm parçacıkların uygunluk değerleri hesaplanır.
- Her bir parçacık için mevcut sürü içerisinde yerel en iyi ( $p_{en\_iyi}$ ) bulunur.
- Sürü içerisindeki yerel en iyiler arasında küresel en iyi ( $g_{en\_iyi}$ ) seçilir.
- Pozisyon ve hızlar güncellenir.

$$V_{i^{k+1}} = w \times V_{i^k} + c_1 \times r_1 \times (g_{en\_iyi_{i^k}} - X_{i^k}) \dots \quad (1)$$

$$\dots + c_2 \times r_2 \times (p_{en\_iyi_{i^k}} - X_{i^k})$$

$$X_{i^{k+1}} = X_{i^k} + V_{i^{k+1}} \quad (2)$$

Burada;  $i$  parçacık sayısını,  $k$  iterasyon sayısını,  $X_i^k$  parçacık pozisyonunu,  $V_i^k$  parçacık hızını,  $r_1$  ve  $r_2$  0-1 arasında rastgele üretilen sayıları,  $w$  eylemsizlik katsayısını,  $c_1$  bilişsel katsayı ve  $c_2$  sosyal katsayı değerlerini ifade etmektedir.

vii. Her iterasyonda, parçacıkların konumları ve hızları güncellenir. Böylece parçacıklar her iterasyonda istenilen çözüme doğru yaklaşacaklardır. Durdurma kriteri sağlanıncaya kadar adım ii, iii, iv, v tekrarlanır.

### B. Genetik Algoritma

Genetik algoritma (GA), evrim teorisi ve doğal seleksiyon mekanizması üzerinden geliştirilen bir optimizasyon yöntemidir [20]. Doğal seleksiyon, çevre

koşullarına daha iyi uyum sağlayan bireylerin hayatta kalma ve üreme şanslarının yüksek olması prensibine dayanan bir mekanizmaya göre çalışmaktadır. Yaşadığı çevre koşullarına uyum sağlayamayan bireyler hayatta kalamayacak ve böylece kalıtsal özelliklerini sonraki kuşaklara aktaramayacaklardır [20-21]. Genetik algoritma temel olarak aşağıdaki adımlardan oluşur;

- Başlangıç popülasyonu oluşturulur. Başlangıç popülasyonu tamamen rassal oluşturulabileceği gibi problemin olası çözümleri başlangıç değerleri olarak kullanılabilir.
- Popülasyondaki her bireyin uygunluk değeri hesaplanır.
- Sırasıyla seçim, çaprazlama ve mutasyon işlemleri uygulanır.
- Durdurma kriteri sağlanıncaya kadar adım ii ve iii tekrarlanır.

## III. OPTİMİZASYON ÇALIŞMALARI

Optimizasyon çalışmalarında uygunluk fonksiyonu için sırasıyla integral mutlak hata (IAE), integral karesel hata (ISE) ve integral zaman ağırlıklı mutlak hata (ITAE) performans endeksleri kullanılmıştır. Bu performans endeksleri için kullanılan matematiksel denklemler aşağıda sırasıyla ifade edilmektedir.

$$IAE = \int |e(t)| dt \quad (3)$$

$$ISE = \int e^2(t) dt \quad (4)$$

$$ITAE = \int t|e(t)| dt \quad (5)$$

Tablo 1'de parçacık sürüsü optimizasyonu için kullanılan parametreler, Tablo 2'de ise genetik algoritma için kullanılan parametreler görülmektedir. Her iki optimizasyon algoritmasında da aynı büyüklükte sürü ve popülasyon kullanılmıştır.

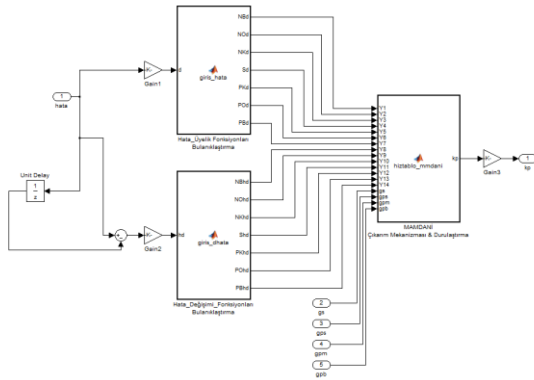
PSO Parametreleri	Değeri
Sürü Büyüklüğü	40
İterasyon Sayısı	100
Problem Boyutu	4
Bilişsel Katsayı	1
Sosyal Katsayı	1
Eylemsizlik Katsayısı	0.1
Sınırlar	[0-1]

Tablo I. PSO Parametreleri

GA Parametreleri	Değeri
Popülasyon Büyüklüğü	40
İterasyon Sayısı	100
Problem Boyutu	4
Çaprazlama Oranı	0.9
Mutasyon Oranı	0.005
Sınırlar	[0-1]

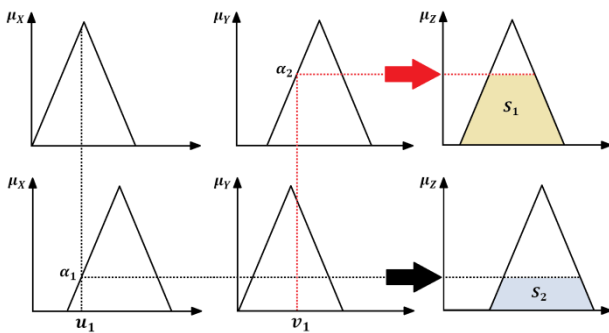
Tablo II. GA Parametreleri

Optimizasyon çalışmalarında iki ayrı sezgisel algoritma ile çıkış üyelik fonksiyonlarının optimize edilmesi hedeflenmiştir. Optimize edilecek çıkış üyelik fonksiyonları genetik algoritma için popülasyondaki bireylerle, parçacık sürüşü optimizasyonu için ise sürüdeki parçacıklarla ifade edilmiştir. Burada, bireyler ve parçacıklar optimize edilecek çıkış üyelik fonksiyonlarının ağırlık merkezlerini temsil etmektedir. Belirlenmiş olan hata performans kriterlerine göre optimize edilen üyelik fonksiyonlarının alanları iterasyonlar boyunca otomatik olarak güncellenmektedir. Çalışmalarda örnek bir sistem olarak kalıcı mıknatıslı senkron motorun hız kontrol sistemi seçilmiştir. Optimizasyon çalışmaları, kalıcı mıknatıslı senkron motorun MATLAB/Simulink ortamında kurulan devresi üzerinde yapılmıştır. Kullanılan motora ait parametreler ( $R_s=0.96\Omega$ ,  $L_d=L_q=0.00525H$ ,  $B=0.0003035Nms$ ,  $J=0.00064kgm/s^2$ ,  $\psi_m=0.1827Wb$ ,  $p=4$ ).



Şekil 1. Bulanık mantık denetleyici yapısı

Şekil 1'de tasarımı yapılan bulanık mantık denetleyicinin yapısı görülmektedir. Tasarlanan bulanık mantık denetleyici; bulanıklaştırma, kural tabanlı çıkarım mekanizması ve durulaştırma birimi olmak üzere üç temel kısımdan oluşmaktadır. Bu birimler tümüyle yazılımsal olarak tasarlanmıştır.



Şekil 2. Mamdani bulanık çıkarım yönteminin yapısı

Çalışmalarda bulanık mantık denetleyici için Şekil 2'de yapısı görülmekte olan Mamdani çıkarım yöntemi kullanılmıştır. Mamdani çıkarım yönteminde bulanıklaştırma biriminden gelen üyelik seviyeleri minimum ilişki operatörü kullanılarak ilgili çıkış üyelik fonksiyonu ile ilişkilendirilmektedir. Mamdani çıkarım yöntemine ait bulanık ilişki operatörleri aşağıda ifade edilmektedir [22].

$$Z_i = (\mu_{xi}(u) \cap \mu_{yi}(v)) \cap \mu_{zi}(w) \quad (6)$$

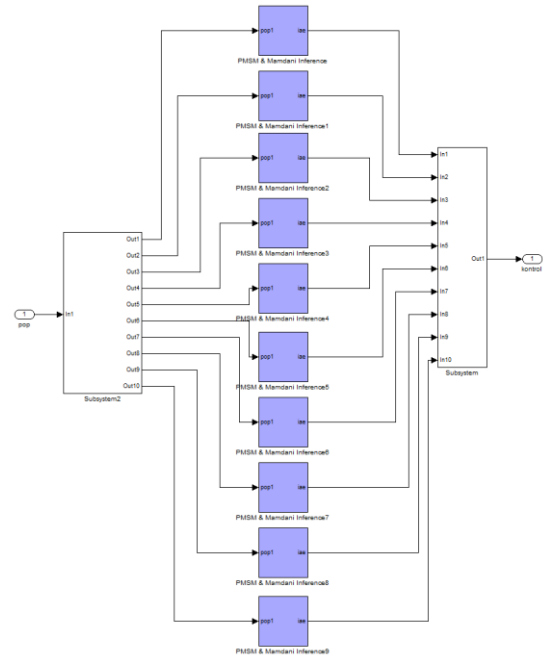
Burada,  $\mu_x$  ve  $\mu_y$  giriş üyelik fonksiyonlarını,  $\mu_z$  çıkış üyelik fonksiyonlarını,  $Z$  çıkarım sonucunu ve  $i$  kural sırasını ifade etmektedir. Çıkarım mekanizmasında,  $k$  adet kural için elde edilecek toplam sonuç tüm çıkarım değerlerinin birleşimi olarak aşağıdaki gibi ifade edilmektedir.

$$Z = \bigcup_{i=1}^k Z_i \quad (7)$$

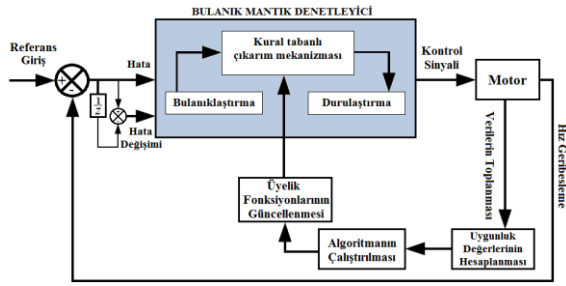
Denetleyicinin durulaştırma biriminde ağırlıklı ortalama yöntemi kullanılmıştır. Bu işlem için denklem (8)'de yer alan hesaplama yöntemi kullanılmıştır [22].

$$Z_{çıkış} = \frac{\sum_{i=1}^n \alpha_i Z_i}{\sum_{i=1}^n \alpha_i} \quad (8)$$

Optimizasyon işlemleri için durdurma kriteri olarak maksimum iterasyon sayısı kullanılmıştır. Şekil 3'te optimizasyon işlemi için tasarlanan modelde uygunluk değerlerinin elde edildiği kısım görülmektedir.



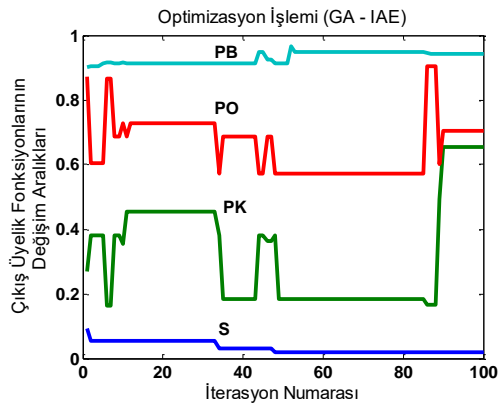
Şekil 3. Tasarlanan modelde uygunluk değerlerinin elde edildiği kısım



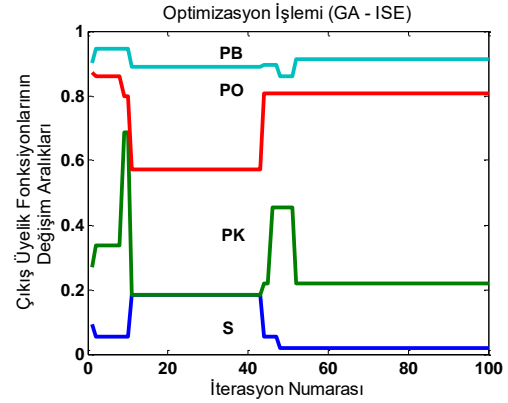
Şekil 4. Optimizasyon genel blok şeması

Şekil 4'te optimizasyon işlemlerinde kullanılan işleyişe ait genel blok şema yapısı görülmektedir. Çalışmalarda bulanık mantık denetleyici için hız hatası ve hız hatasındaki değişim giriş değişkenleri olarak kullanılmıştır. İterasyonlarda öncelikle motor için kullanılan hata endeksine göre uygunluk değerleri hesaplanmaktadır. Kullanılan sezgisel algoritma çalıştırılarak mevcut sürüdeki veya popülasyondaki uygunluk değerleri arasından en iyi değerler tespit edilir. En başarılı sonuçları sağlayan çözüm değerleri bir sonraki iterasyon için kullanılacak olan çıkış üyelik fonksiyonlarını ifade etmektedir. Çıkış üyelik fonksiyonları için Sıfır(S), Pozitif Küçük(PK), Pozitif Orta(PO), Pozitif Büyük(PB) olmak üzere dört farklı sözel değişken kullanılmıştır. Motor hız geribeslemesiyle giriş referans hızı karşılaştırılarak elde edilecek hata, bulanık mantık denetleyici için yeni giriş değerleri olmaktadır. Durdurma kriteri sağlanıncaya kadar çıkış üyelik fonksiyonları belirtilen şekilde optimize edilmeye devam edecektir.

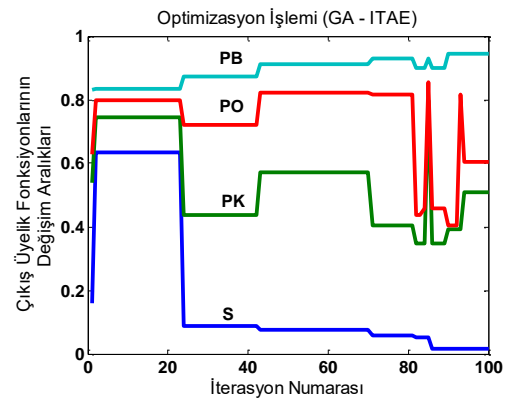
Şekil 5-10'da optimize edilen çıkış üyelik fonksiyonlarının iterasyonlar boyunca değişim aralıkları görülmektedir. Her iterasyonda elde edilen çıkış üyelik fonksiyonları bir sonraki iterasyon için denetleyicinin çıkarım işlemlerinde kullanılmaktadır.



Şekil 5. IAE ile optimize edilen çıkış üyelik fonksiyonlarının değişim aralıkları gösterimi (GA)

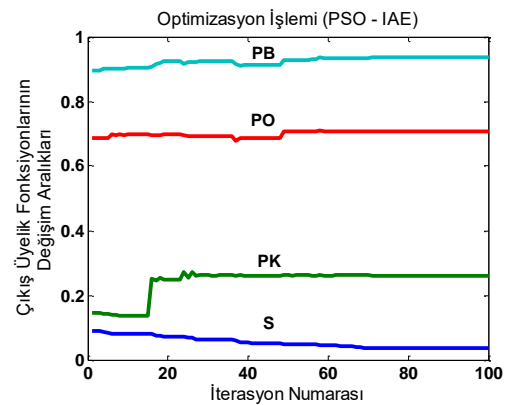


Şekil 6. ISE ile optimize edilen çıkış üyelik fonksiyonlarının değişim aralıkları gösterimi (GA)

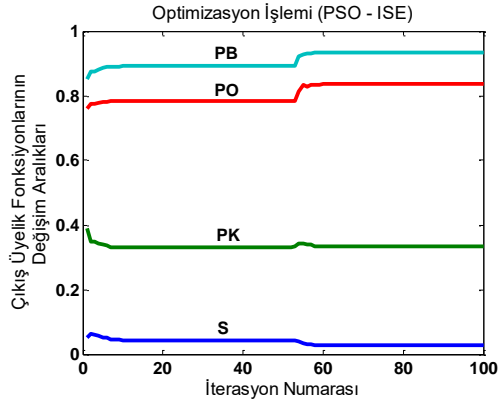


Şekil 7. ITAE ile optimize edilen çıkış üyelik fonksiyonlarının değişim aralıkları gösterimi (GA)

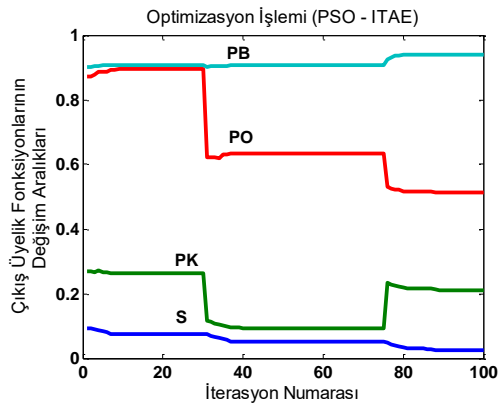
Şekil 5-7'de genetik algoritma kullanılarak optimize edilen çıkış üyelik fonksiyonlarının değişimleri görülmektedir.



Şekil 8. IAE ile optimize edilen çıkış üyelik fonksiyonlarının değişim aralıkları gösterimi (PSO)

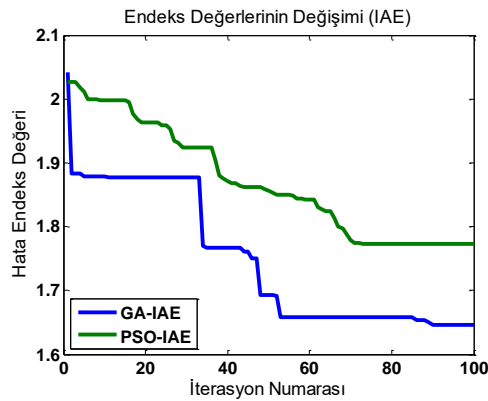


Şekil 9. ISE ile optimize edilen çıkış üyelik fonksiyonlarının değişim aralıkları gösterimi (PSO)

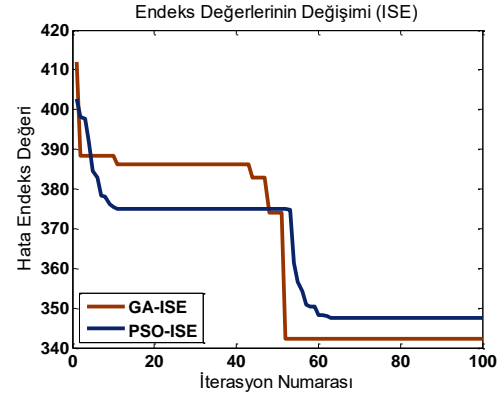


Şekil 10. ITAE ile optimize edilen çıkış üyelik fonksiyonlarının değişim aralıkları gösterimi (PSO)

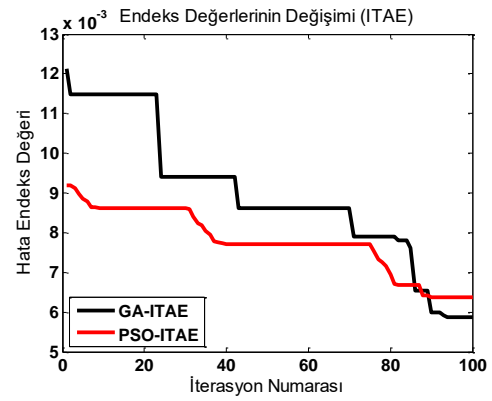
Şekil 8-10'da parçacık sürüşü optimizasyonu kullanılarak optimize edilen çıkış üyelik fonksiyonlarının değişimleri görülmektedir. Şekil 11-13'te ise optimizasyon işlemlerinde elde edilen hata performans endekslerinin iterasyonlar boyunca değişimleri görülmektedir.



Şekil 11. IAE hata endeksinin değişimi



Şekil 12. ISE hata endeksinin değişimi



Şekil 13. ITAE hata endeksinin değişimi

Şekil 11-13'te yer alan hata performans endeksi değişimleri incelendiğinde, genetik algoritma ile daha küçük hata değerlerinin elde edildiği görülmektedir. Tüm hata endeksleri için optimizasyon yöntemleri başlangıçta hızlı bir yakınsama sağlarken, ilerleyen iterasyonlarda yakınsama hızlarının belirli bir yerden sonra sabit kaldığı görülmektedir. Yapılan optimizasyon işlemleri sonucunda elde edilen hata performans endekslerinin minimum değerleri Tablo 3'te karşılaştırılmaktadır.

Hata Performans Endeksleri	Genetik Algoritma	Parçacık Sürüşü Optimizasyonu	Yüzdesel Farklılık
IAE	1.64659	1.773411	%7.15
ISE	342.3453	347.4179	%1.46
ITAE	0.005861779	0.006365	%7.9

Tablo III. Elde edilen minimum hata performans endekslerinin karşılaştırılması

#### IV. SONUÇLAR

Bu çalışmada, bulanık mantık denetleyici parçacık sürüşü optimizasyonu ve genetik algoritma ile farklı performans

kriterlerine göre optimize edilmiştir. Denetleyici tasarımı ve optimizasyon işlemlerine ait tüm aşamalar yazılımsal olarak gerçekleştirilmiştir. Sonuçlar değerlendirildiğinde, genetik algoritmanın tüm hata performans endekslerinde parçacık sürüşü optimizasyonuna göre daha başarılı sonuçlar sağladığı görülmüştür.

Yapılan çalışmada, optimum bulanık mantık denetleyici çıkış üyelik fonksiyonlarının belirlenmesinde sezgisel algoritmaların kullanımı adım adım anlatılmıştır. Optimizasyon işlemleri sonucunda çözüm uzayında daha hızlı ve etkin çözümlerin elde edilebilmesi sağlanmıştır. Böylece denetleyici parametrelerinin belirlenmesinde deneme yanılma yoluyla oluşacak zaman kaybının önüne geçilmiş ve uzman kişilere olan ihtiyacın azaltılması sağlanmıştır.

Mevcut çalışma farklı hata performans endekslerinin kullanımını içerecek şekilde genişletilebilir. Bulanık mantık denetleyicinin çıkış üyelik fonksiyonları haricindeki diğer tasarım parametrelerinin de optimize edilmesi hedef çalışmalar arasında yer almaktadır.

#### YAZAR KATKILARI

*Birinci yazar* çalışmanın sorumlu yazarı olup benzetim çalışmalarını yapmıştır. *İkinci yazar* çalışmanın genel incelemesini ve kontrolünü yapmıştır.

#### TEŞEKKÜR

Bu çalışma, Bülent Ecevit Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından desteklenmiştir.

#### KAYNAKÇA

- [1] E. L. Olivas, O. Castillo, F. Valdez, J. Soria, "Ant Colony Optimization for Membership Function Design for a Water Tank Fuzzy Logic Controller," *Hybrid Intelligent Models and Applications (HIMA)*, Apr 16-19, 27-34, Singapore, 2013.
- [2] N. Kapetanović, N. Osmić, S. Konjicija, "Optimization of membership functions of Sugeno-Takagi fuzzy logic controllers with two inputs and one output using genetic algorithms," *International Symposium on Telecommunications (BIHTEL)*, Oct 27-29, Sarajevo/Bosnia and Herzegovina, 2014.
- [3] A. Ullah, J. Li, A. Hussain, Y. Shen, "Genetic Optimization of Fuzzy Membership Functions for Cloud Resource Provisioning," *Computer Science and Electronic Engineering (CEECE)*, 2016.
- [4] A. K. Paul, P. C. Shill, "Optimizing Fuzzy Membership Function Using Dynamic Multi Swarm-PSO," *Informatics, Electronics and Vision*, pp. 139-144, 2016.
- [5] B. Safae, S. K. M. Mashhadi, "Fuzzy Membership Functions Optimization of Fuzzy Controllers for A Quad Rotor Using Particle Swarm Optimization and Genetic Algorithm," *Control, Instrumentation, and Automation (ICCIA)*, pp. 256-261, 2016.
- [6] M. Caner, U. Gülseren, "Genetik Algoritma İle Fuzzy PSS'in Kural Tablosu Optimizasyonu," *AKÜ Fen Bilimleri Dergisi*, 2010-01, 83-92, 2011.
- [7] N. Pitalua-Diaz, R. Lagunas-Jimenez, G. Angelesa "Tuning Fuzzy Control Rules via Genetic Algorithms An Experimental Evaluation," *Electronics, Robotics and Automotive Mechanics Conference*, Vol. 2(10), pp. 81-87, 2013.
- [8] S. Yeasmin, A. K. Pau, P. C. Shill, "Optimization of interval type-2 fuzzy logic controllers with rule base size reduction using genetic algorithms," *Electrical Engineering and Information Communication Technology (ICEEICT)*, Sep 22-24, Dhaka/Bangladesh, 2016.
- [9] N.M.A Ghani, A.N.K Nasir, M.O. Tokhi "Optimization of fuzzy logic scaling parameters with spiral dynamic algorithm in controlling a stair climbing wheelchair : Ascending task," *Methods and Models in Automation and Robotics (MMAR)*, Sep 2-5, 776-781, Miedzyzdroje/Poland, 2014.
- [10] A. Sebastião, C. Lucena, L. Palma, A. Cardoso, P. Gil, "Optimal Tuning of Scaling Factors and Membership Functions for Mamdani Type PID Fuzzy Controllers," *Control, Automation and Robotics (ICCAR)*, pp. 92-96, 2015.
- [11] Z. Salleh, M. Sulaiman, R. Omar, F. A. Patakor, "Optimization of Fuzzy Logic Based for Vector Control Induction Motor Drives," *Computer Science and Electronic Engineering (CEECE)*, 2016.
- [12] D. Pelusi, "Optimization of a fuzzy logic controller using genetic algorithms," *Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 143-146, 2011.
- [13] F. Valdez, P. Melin, O. Castillo, "Particle Swarm Optimization for Designing an Optimal Fuzzy Logic Controller of a DC Motor," *Fuzzy Information Processing Society (NAFIPS)*, 2012.
- [14] T. Chaiyatham, I. Ngamroo, "A Bee Colony Optimization Based-Fuzzy Logic-PID Control Design Of Electrolyzer for Microgrid Stabilization," *International Journal of Innovative Computing*, 8(9), 2012.
- [15] S. Rajan, S. Sahadev, "Performance Improvement of Fuzzy Logic Controller using Neural Network," *International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST)*, 704-714, 2016.
- [16] D. H. Nguyen, "A Comparison of DE and SFLA Optimization Algorithms in Tuning Parameters of Fuzzy Logic Controller," *International Journal of Computer Applications*, 156(11), 17-22, 2016.
- [17] D. Karaboğa, "Yapay Zeka Optimizasyon Algoritmaları", Nobel Yayın Dağıtım, 2011.
- [18] S. Akyol, B. Alataş, "Güncel Sürü Zekası Optimizasyon Algoritmaları," *Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi* 1, 36-50, 2012.
- [19] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization," *Proc. of the IEEE Int. Conference on Neural Networks*, 4, pp. 1942-1948, 1995.
- [20] J. H. Holland, *Adaptation In Natural And Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, New York, 1989.
- [22] İ. Aışkan, S. Ünsal, "Farklı Çıkarım Yöntemlerine Sahip Bulanık Mantık Denetleyicileri Kullanarak Kalıcı Mıknatıslı Senkron Motorun Hız Denetimi," *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 24(2), 185-191, 2018.