



## Karınca Kolonisi Optimizasyon Algoritması ile Bir AVR Sistemi için Optimum PID Denetleyici Tasarımı

Taner Yurdusever<sup>1\*</sup>, Yahya Danayiyen<sup>2</sup>

<sup>1</sup> Elektrik-Elektronik Mühendisliği, Karadeniz Teknik Üniversitesi, Türkiye

<sup>2</sup> Elektrik-Elektronik Mühendisliği, Karadeniz Teknik Üniversitesi, Türkiye

\*([taneryurdusever@outlook.com](mailto:taneryurdusever@outlook.com)) Başlıca yazarın mail adresi

**Özet** – Bu çalışmada bir otomatik gerilim regülatörü sisteminde kullanılmak üzere bir denetleyici tasarımı üzerinde durulmuştur. Literatürde denetleyici tasarımında metasezgisel optimizasyon algoritmalarının sıklıkla tercih edildiği görülmektedir. Bu çalışmalar incelendiğinde bu sistem için Karınca Kolonisi Algoritmasının (ACO) kullanılmamış olduğu görülmüştür. Ancak Karınca Kolonisi Algoritması ile yapılmış başka çalışmalara dayanarak ACO algoritmasının bu sistemde de başarılı sonuçlar vereceği öngörülmüştür. Bu sebeple bu çalışmada bir otomatik voltaj regülatör sistemi için ACO algoritması kullanarak bir PID denetleyici tasarımı örneği sunulmuştur. Sistem performansı iyileştirmek için ağırlık katsayıları ile integral zaman çarpma mutlak hatası (ITAE), maksimum aşma ve yerleşme zamanını kullanan bir amaç fonksiyonu önerilmiştir.

*Anahtar Kelimeler* – Karınca Kolonisi Optimizasyonu, Otomatik Voltaj Regülatörü, Yerleşme Zamanı, Maksimum Aşma, PID

### I. GİRİŞ

Kontrol sistemlerinin daha kararlı çalışabilmesi için denetleyiciler kullanılır. Bu çalışmada PID (İngilizce: Proportional Integral Derivative) denetleyiciler üzerinde durulmuştur. PID denetleyicilerde seçilen KP, KI ve KD katsayılarının alacağı değerlere göre sistem çıkışının davranışı değişmektedir. Ancak bu parametrelerin deneme-yanılma yoluyla seçilmesi oldukça zaman alıcı ve zor olmaktadır. Bu sebeple çeşitli algoritmalar ile bu katsayılar bulunmaya çalışılır. Literatürde metasezgisel optimizasyon algoritmalarının sıklıkla tercih edildiği görülmektedir. Örneğin Karınca kolonisi algoritması, Balina optimizasyon algoritması, genetik algoritmalar, parçacık sürü optimizasyonu bu algoritmalara örnek olarak verilebilir.

Literatürdeki çalışmalar incelendiğinde Karınca Kolonisi Algoritmasının (İngilizce: Ant Colony Optimization, ACO) birçok farklı kontrol

sisteminde kullanıldığı görülmektedir. Örneğin Herlambang ve ark. Parçacık Sürü Optimizasyonu ve Karınca Kolonisi Algoritmasını bir su altı aracının kontrol sistemi için PID katsayılarını en iyilemek için kullanmışlardır [1]. Chen ve ark. geliştirilmiş bir ACO algoritması ile çok bölgeli güç sistemlerinde yük frekans kontrolü için PID denetleyici tasarlamışlardır [2]. Şimşek ve ark. DC motorun hareket kontrolünde yükselme, yerleşme zamanı ile maksimum aşmayı düşürmek amacıyla ACO algoritmasını kullanmışlardır [3]. Mughees ve Mohsin benzer şekilde aynı algoritmayı manyetik kaldırma sistemlerinin denge kontrolü için tasarlan kesir dereceli PID katsayılarını bulmak için kullanmışlardır [4]. Bu çalışmalarda çoğunlukla sistem çıkışının yerleşme zamanı ve maksimum aşma değerlerinin azaltılması çalışılmış ve ACO algoritmasının başarılı sonuçlar verdiği görülmüştür. Karami ve ark. titreşimli aktüatörlerle donatılmış bir mikro robotun hareket kontrolünü

sağlamak için ACO tabanlı doğrusal olmayan optimal bir PID denetleyici tasarlamışlardır [5].

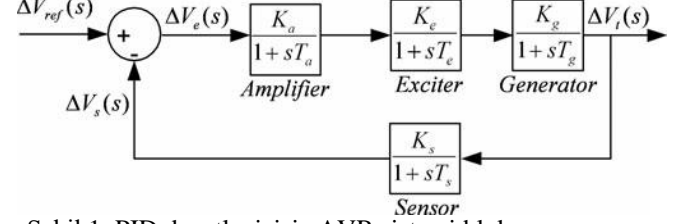
Bir AVR (Automatic Voltage Regulator) sisteminin temel görevi çıkış voltajını bir referans değere göre sabit tutmaya çalışmaktır. Literatürde birçok farklı optimizasyon algoritmasının AVR sistemlerinin çıkış gerilimini kontrol etmek amacıyla PID tasarımında kullanıldığı görülmektedir. Örneğin, Ekinci ve Hekimoğlu Kidney-Inspired Algorithm (IKA) algoritmasını kullanarak PID denetleyicili çıkıştan %15 maksimum aşma, 0.753 s yerleşme zamanı değerlerine ulaşmışlardır. Bir diğer çalışmada [7] Balina Optimizasyon Algoritması kullanılmış olup, %1.07 maksimum aşma, 0.555 s yerleşme zamanı elde edilmiştir. Yine bir başka çalışmada [8] kesir dereceli PID tasarımında Jaya Algoritması ile %13.2 maksimum aşma, 0.453 s yerleşme zamanı elde edilmiştir. Altbawi ve ark. Gradyen Tabanlı Optimizasyon Algoritması ile denetleyici katsayılarını optimize ederek çıkıştan %11.3 maksimum aşma, 0.653 s yerleşme zamanı değerleri elde etmişlerdir [9]. Başka bir güncel çalışmada [10] ise, genetik algoritma kullanılarak %0.73 maksimum aşma, 0.276 s yerleşme zamanı (%2 lik dilime göre) elde edilmiştir.

Literatürde yapılan incelendiğinde AVR sistemlerinin PID tasarımında Karınca Kolonisi Algoritmasının kullanılmamış olduğu görülmüştür. Ancak Karınca Kolonisi Algoritması ile yapılmış başka çalışmalara dayanarak ACO algoritmasının AVR sisteminde de başarılı sonuçlar vereceği öngörülmüştür. Bu sebeple bu çalışmada bir AVR sistemi için ACO algoritması kullanılarak bir PID denetleyici tasarımı örneği sunulmuştur. Bu sebeple çalışmamızın yenilikçi olduğu söylenebilir.

## II. OTOMATİK GERİLİM REGÜLATÖR MODELİ

Bir AVR sisteminin temel görevi jeneratörün çıkış gerilimini sabit bir seviyede tutmaktır [11]. Güç sisteminde meydana gelen bozucu salınımlar nedeniyle bir jeneratör her zaman dengeli bir şekilde çalışmaz. Bu tür salınımlar, güç sisteminin kararlılığı konusunda zararlı olabilecek frekans ve voltaj sapmalarına sebebiyet verebilir [12]. Bir AVR sistemi sırasıyla yükseltici, uyarıcı, üreteç ve sensör bloklarından oluşmaktadır. PID

denetleyicisiz AVR sistemi blok şeması Şekil 1’de gösterilmiştir.



Şekil 1. PID denetleyicisiz AVR sistemi blok şeması

AVR blok şemasında bulunan sensör istenen ve ölçülen gerilim değeri arasındaki farkı gösteren bir hata sinyali oluşturulur. Daha sonra bu hata sinyali yükselteç tarafından güçlendirilir ve uyarıcı bloğuna gönderir. Eğer bu sinyal uyarım eşiğini geçerse üreteç tarafından sağlanan güç jeneratördeki uyarım bobinlerini devreye sokar ve çıkış gerilimini dengeleme çalışır. AVR sisteminin kullanılan parametreler Tablo 1’de verilmiştir. Bu çalışmada AVR bileşenlerinin katsayı ve zaman sabiti değerleri literatürdeki diğer çalışmalar dikkate alınarak seçilmiştir [13]. Sistemin örneklem frekansı 1000 Hz dir.

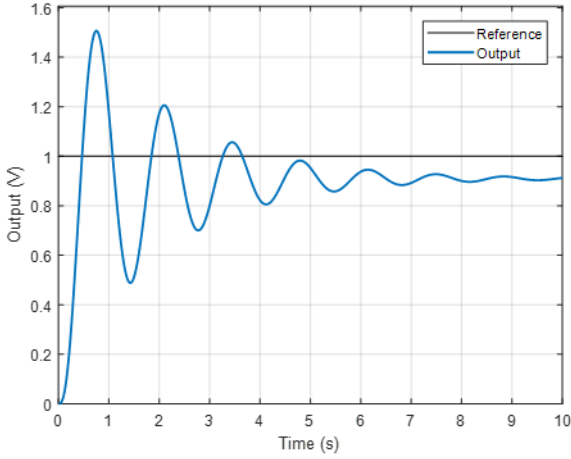
Tablo 1. AVR sisteminde kullanılan parametreler

|                  | Transfer Fonksiyonu           | Kazanç       | Zaman Sabiti |
|------------------|-------------------------------|--------------|--------------|
| <b>Amplifier</b> | $TF_a = \frac{K_a}{1 + sT_a}$ | $K_a = 10.0$ | $T_a = 0.1$  |
| <b>Excitor</b>   | $TF_e = \frac{K_e}{1 + sT_e}$ | $K_e = 1.0$  | $T_e = 0.4$  |
| <b>Generator</b> | $TF_g = \frac{K_g}{1 + sT_g}$ | $K_g = 1.0$  | $T_g = 1.0$  |
| <b>Sensor</b>    | $TF_s = \frac{K_s}{1 + sT_s}$ | $K_s = 1.0$  | $T_s = 0.01$ |

PID denetleyicisiz bir AVR sisteminin transfer fonksiyonu denklem (1)’de verilmiştir.

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1s+10}{0.0004s^4+0.0454s^3+0.555s^2+1.51s+11} \quad (1)$$

Ancak denetleyicisiz bir AVR sisteminde kararlılığı sağlamak zordur. PID denetleyicisiz AVR sisteminin birim basamak cevabı Şekil 2’de gösterilmiştir.



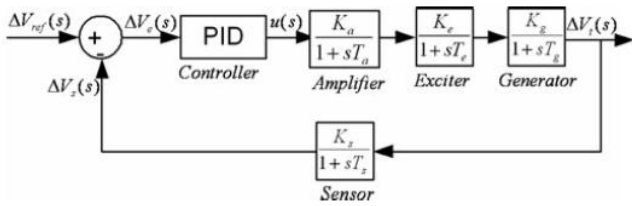
Şekil 2. PID denetleyicisiz AVR sisteminin birim basamak cevabı

PID denetleyicisiz AVR sisteminin birim basamak cevabının zaman bölgesi kriterleri Tablo 2’de verilmiştir.

Tablo 2. PID denetleyicisiz AVR sisteminin birim basamak cevabının zaman bölgesi kriterleri

| Parametre           | Değer  |
|---------------------|--------|
| Yükselme zamanı (s) | 0.2613 |
| Yerleşme zamanı (s) | 7.0192 |
| Maks. aşma (%)      | 65.214 |
| Tepe değeri (V)     | 1.5066 |

PID denetleyicili AVR sistemi blok şeması Şekil 3’te gösterilmiştir.



Şekil 3. PID denetleyicili AVR sistemi blok şeması

PID denetleyicisiz bir AVR sisteminin transfer fonksiyonu denklem (1)’de verilmiştir.

$$TF_{PID} = K_p + \frac{K_I}{s} + K_D s \quad (2)$$

PID denetleyicili bir AVR sisteminin transfer fonksiyonu denklem (3)’te verilmiştir.

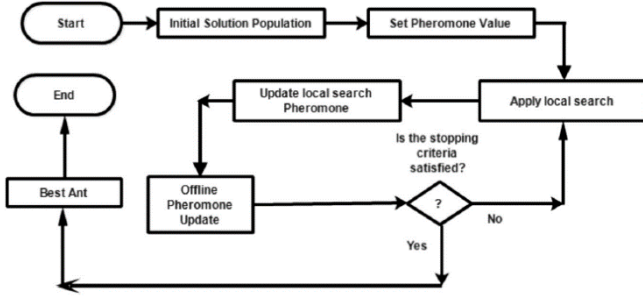
$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1K_D s^3 + (0.1K_P + 10K_D)s^2 + (0.1K_I + 10K_P)s + 10K_I}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + (1.51 + 10K_D)s^2 + (1 + 10K_P)s + 10K_I} \quad (3)$$

### III. KARINCA KOLONİSİ OPTİMİZASYONU ALGORİTMASI

Karınca kolonisi optimizasyonu (İngilizce: Ant colony optimization, ACO) algoritması, karınca sürüsü davranışından ilham alan ve ilk başta sıralama optimizasyon problemleri için tasarlanmış ayrı bir yöntemdir. [14]. ACO’nun ana fikri, problemi bir grafta minimum maliyetli bir yol aramaktır. Yapay karıncalar, iyi yollar arayarak bu grafta yürürler. Her karıncanın oldukça basit bir davranışı vardır, bu nedenle tipik olarak kendi başına en iyi yolu bulacağı garanti edilemez. Ancak feromon adı verilen çekici bir molekülün etkinliği sayesinde sürü olarak daha iyi yollar bulunur. Yapay karıncaların davranışları gerçek karıncalardan esinlenmiştir. Her bir karınca geçtiği düğümde feromon izi bırakır ve yollarını feromon izlerine bağlı olasılıklara göre seçerler ve bu feromon izleri buharlaşma ile giderek azalır. Karıncalar, yüksek miktarda feromon içeren düğümler üzerinde yol almayı tercih ederler [15].

#### A. Algoritmanın Akış Şeması

ACO’nun amacı, yuvalarından yiyecek kaynaklarına giden daha kısa yollar bulmaktır. Karıncalar, diğer karıncalarla iletişim kurmalarını sağlayan feromon adı verilen kimyasal bir bileşik bırakırlar. Bir karınca hareket ettiğinde, arkasında diğer karıncaların algılayabileceği tutarlı miktarda feromon bırakır. Her karınca kendine özgü bir şekilde seyahat eder, ancak bir feromon iziyle karşılaştığında onu takip edip etmemeye karar vermesi gerekir. Bir karınca izi takip ederse, kendi feromonu onu güçlendirir ve feromondaki bir artış, bir sonraki karıncanın onu takip etme olasılığını artırır. Sonuç olarak, bir yol boyunca ne kadar çok karınca giderse, yol sonraki karıncalar için o kadar çekici hale gelir. Ayrıca besin kaynağına daha kısa yoldan giden karınca yuvasına daha çabuk döner. Bu süreç Şekil 4’te akış şeması olarak ifade edilmiştir.



Şekil 4. Karınca Kolonisi Algoritması akış şeması

## B. Amaç Fonksiyonu

Amaç fonksiyonunda en önemli parametre hata miktarıdır. Literatürde hatayı ölçmek için birçok farklı hata fonksiyonu bulunmaktadır. Bu problemde hata ITAE (Integral of Time Weighted Absolute Error) yöntemi ile ölçülmüştür. Amaç fonksiyonunda kullanılan ITAE fonksiyonu denklem (4)'te verilmiştir.

$$ITAE = \int_0^t |V_r - V_t| dt \quad (4)$$

AVR sisteminde yüksek genlikli voltaj değişiklikleri sisteme zarar verebilir. Bu sebeple amaç fonksiyonuna maksimum aşma ( $M_p$ ) eklenir. Aynı zamanda regülasyonun hızlı olması için daha küçük yerleşme zamanı ( $T_s$ ) tercih edilir. Bu parametreler sabit katsayılarla çarpılıp, toplandıktan sonra amaç fonksiyonu elde edilir. Sistemde kullanılan amaç fonksiyonu denklem (5)'te verilmiştir.

$$cost = \alpha_1 \times ITAE + \alpha_2 \times T_s + \alpha_3 \times M_p \quad (5)$$

Amaç fonksiyonunda belirtilen katsayılar sırasıyla  $\alpha_1=0.8$ ,  $\alpha_2=0.1$  ve  $\alpha_3=0.1$  olarak seçilmiştir.

## IV. BULGULAR VE TARTIŞMA

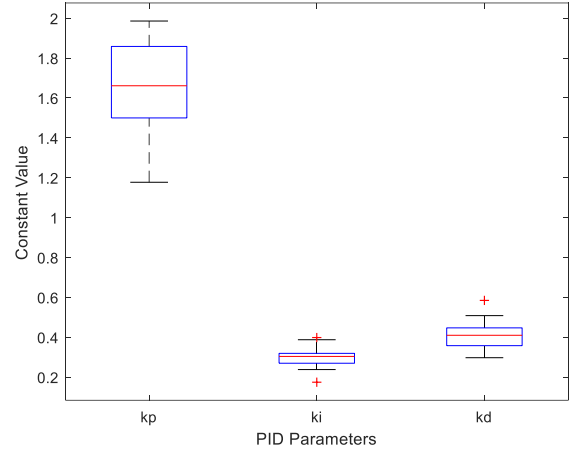
Karınca kolonisi optimizasyonu algoritması için her PID parametresi için toplam 3 düğüm oluşturulmuştur. Her parametrenin düğüm sayısı 1000 olarak belirlenmiştir. Algoritmanın arama uzayı sınırları literatürdeki çalışmalara dayanarak  $K_P=[0,2]$ ,  $K_I=[0,1]$ ,  $K_D=[0,1]$  olarak seçilmiştir. Simülasyonun iterasyon sayısı 10, her iterasyondaki karınca sayısı 50, alfa=0.8, beta=0.2 ve buharlaşma oranı ( $\rho$ )=0.7 olacak şekilde

seçilmiştir. Algoritmanın yerel minimumlara takılması olasılığını en aza indirmek için simülasyon 30 defa tekrarlanmıştır. 30 koşumun çıktıklarına ait istatistikler Tablo 3'te verilmiştir.

Tablo 2. Simülasyon sonucunun istatistikleri

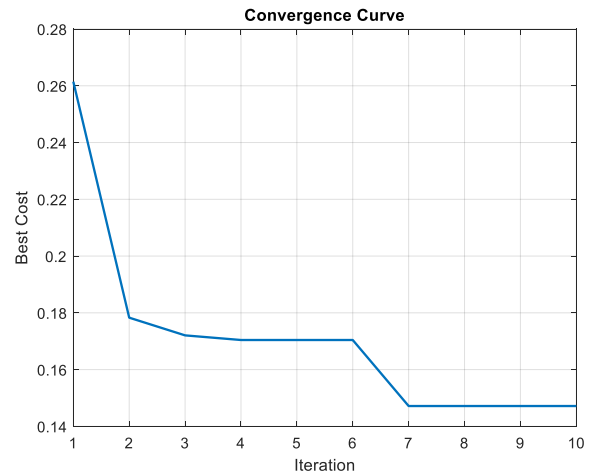
| Kriter         | Değer   |
|----------------|---------|
| En iyi çözüm   | 0.14716 |
| En kötü çözüm  | 0.1953  |
| Ortalama       | 0.16374 |
| Standart sapma | 0.01062 |

30 koşumun PID katsayı değerlerine ait istatistikler Şekil 5'te gösterilmiştir. Şekilde  $K_P$  parametresinin daha yüksek değerlerde ve daha geniş bir aralıkta aranmış olduğu görülmektedir.



Şekil 5. PID katsayı istatistikleri

Koşum sonuçları arasında elde edilen en iyi çözüme ait yakınsama eğrisi Şekil 6'da gösterilmiştir.



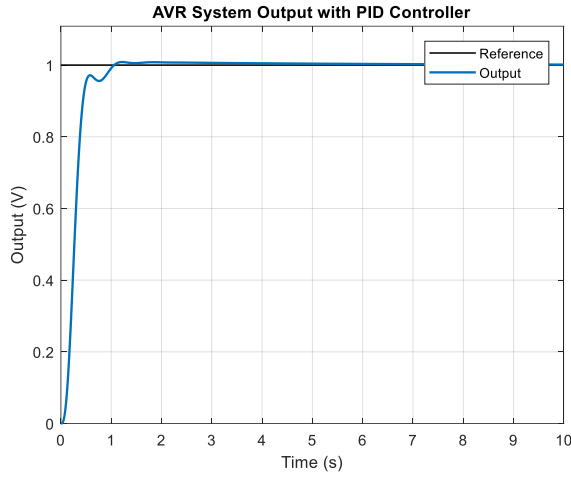
Şekil 6. En iyi çözümün yakınsama eğrisi

AVR sisteminin optimize edilmiş PID katsayı değerleri Tablo 4'te verilmiştir.

Tablo 3. Optimize edilmiş PID katsayıları

| PID Katsayıları | Değer  |
|-----------------|--------|
| En iyi $K_P$    | 1.8586 |
| En iyi $K_I$    | 0.3876 |
| En iyi $K_D$    | 0.4312 |

PID denetleyicili AVR sisteminin birim basamak cevabı Şekil 7'de gösterilmiştir.



Şekil 7. PID denetleyicili AVR sisteminin birim basamak cevabı

PID denetleyicili AVR sisteminin birim basamak cevabının zaman bölgesi kriterleri Tablo 5'te verilmiştir. Sistemin yeni durumda daha düşük maksimum aşma ve yerleşme zamanı değerlerine sahip olduğu görülmüştür.

Tablo 4. PID denetleyicili AVR sisteminin birim basamak cevabının zaman bölgesi kriterleri

| Parametre           | Değer  |
|---------------------|--------|
| Yükselme zamanı (s) | 0.3129 |
| Yerleşme zamanı (s) | 0.9466 |
| Maks. aşma (%)      | 0.7142 |
| Tepe değeri (V)     | 1.0086 |

## v. SONUÇ

Simülasyon sonuçları incelendiğinde maksimum aşma yüzdesinin ilk duruma göre yaklaşık %65 oranında düşüş gözlemlenmiştir. Bu sonuç literatürdeki yer alan en küçük sonuca oldukça yakındır. Benzer şekilde yerleşme zamanı değerinin 7 saniyeden 0.946 saniyeye düştüğü görülmüştür. Literatürde daha iyi sonuçlar yer almasına karşın algoritmanın daha çok sayıda karınca ile tekrarlanması ve klasik PID denetleyici yerine son zamanlarda sıklıkla tercih edilen kesir dereceli PID denetleyicilerin kullanılması ile daha iyi sonuçlar elde edilebilir.

## KAYNAKLAR

- [1] Herlambang, T., Rahmalia, D., & Yulianto, T. (2019, April). Particle swarm optimization (pso) and ant colony optimization (aco) for optimizing pid parameters on autonomous underwater vehicle (auv) control system. In *Journal of Physics: Conference Series* (Vol. 1211, No. 1, p. 012039). IOP Publishing.
- [2] Chen, G., Li, Z., Zhang, Z., & Li, S. (2019). An improved ACO algorithm optimized fuzzy PID controller for load frequency control in multi area interconnected power systems. *IEEE Access*, 8, 6429-6447.
- [3] Şimşek, E. C., Köse, A., Şahin, M., & Irmak, E. (2019, November). Optimization of PID parameters using ant colony algorithm for position control of DC motor. In *2019 8th International Conference on Renewable Energy Research and Applications (ICRERA)* (pp. 1047-1051). IEEE.
- [4] Mughees, A., & Mohsin, S. A. (2020). Design and control of magnetic levitation system by optimizing fractional order PID controller using ant colony optimization algorithm. *IEEE Access*, 8, 116704-116723.
- [5] Karami, M., Tavakolpour-Saleh, A. R., & Norouzi, A. (2020). Optimal nonlinear PID control of a micro-robot equipped with vibratory actuator using ant colony algorithm: Simulation and experiment. *Journal of Intelligent & Robotic Systems*, 99(3), 773-796.
- [6] Ekinci, S., & Hekimoğlu, B. (2019). Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system. *IEEE Access*, 7, 39935-39947.
- [7] Mosaad, A. M., Attia, M. A., & Abdelaziz, A. Y. (2019). Whale optimization algorithm to tune PID and PIDA controllers on AVR system. *Ain Shams Engineering Journal*, 10(4), 755-767.
- [8] Jumani, T. A., Mustafa, M. W., Hussain, Z., Rasid, M. M., Saeed, M. S., Memon, M. M., ... & Nisar, K. S. (2020). Jaya optimization algorithm for transient response and stability enhancement of a fractional-order PID based automatic voltage regulator system. *Alexandria Engineering Journal*, 59(4), 2429-2440.
- [9] Altbawi, S. M. A., Mokhtar, A. S. B., Jumani, T. A., Khan, I., Hamadneh, N. N., & Khan, A. (2021). Optimal

- design of Fractional order PID controller based Automatic voltage regulator system using gradient-based optimization algorithm. *Journal of King Saud University-Engineering Sciences*.
- [10] Dogruer, T., & Can, M. S. (2022). Design and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm. *Transactions of the Institute of Measurement and Control*, 44(9), 1862-1873.
- [11] Micev, M., Čalasan, M., Ali, Z. M., Hasanien, H. M., & Aleem, S. H. A. (2021). Optimal design of automatic voltage regulation controller using hybrid simulated annealing–Manta ray foraging optimization algorithm. *Ain Shams Engineering Journal*, 12(1), 641-657.
- [12] Micev, M., Čalasan, M., & Oliva, D. (2020). Fractional order PID controller design for an AVR system using Chaotic Yellow Saddle Goatfish Algorithm. *Mathematics*, 8(7), 1182.
- [13] Ayas, M. S., & Sahin, E. (2021). FOPID controller with fractional filter for an automatic voltage regulator. *Computers & Electrical Engineering*, 90, 106895.
- [14] Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O., & Khammas, B. M. (2022). Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon*, e09399.
- [15] Dorigo Marco and Gianni Di Caro, “The Ant Colony Optimization Meta-Heuristic” IRIDIA, 1999.