

Control of the Automatic Voltage Regulator System with a Novel Stability-based Artificial Intelligence Method

Rahma Tabakh, Hasan Tiryaki*

*Department of Electrical and Electronics Engineering, Istanbul University-Cerrahpasa,
Avcilar/Istanbul, Turkey
hasan.tiryaki@istanbul.edu.tr*

Abstract—This paper proposes a novel Stability-Based Artificial Intelligence Method for predicting the optimum parameters of the proportional-integral-derivative controller in an automatic voltage regulator system. To implement the stability-based artificial intelligence method, first, parameters which are of great importance for the control of the system are applied to the system randomly, data are collected, and then artificial intelligence studies are carried out. The suggested approach has been applied to the system and compared with other control methods in the literature, namely the improved Kidney Inspired algorithm, Jaya algorithm, Tree Seed algorithm, Water Wave Optimization, and Biogeography-Based Optimization to test the robustness of the new method. The numerical results indicate that the proposed method significantly outperforms all other methods.

Index Terms—Artificial intelligence; Automatic voltage regulator; Optimum parameters; PID controller; Predict.

I. INTRODUCTION

In power systems, conventional voltage control can be grouped into three different stages. These stages are named as primary, secondary, and tertiary voltage control for a real-time, middle, and long-term stage. Automatic voltage regulators regulate the voltage of the generator bus bar by controlling the injection or absorption of reactive power for synchronous generators [1].

An automatic voltage regulator (AVR) is a system that is utilized in generators to automatically regulate the voltage by converting a voltage level from fluctuating to a constant. AVRs systematically take a wide range of input voltages and ensure a stable output voltage. The essential function of the AVR system is to automatically regulate generator voltage and maintain the output constant in the required range of voltage level for the generator without regard to the existing load. AVRs protect the surges from electrical surges and generator overload, and help generators resist overloads to avoid shorting [2], [3].

The Proportional-Integral-Derivative (PID) controller is the most used type in the control process field and the reason behind this is the simplicity of its structure and the high efficiency of its performance under several operating conditions. The topic of transient stability analysis has

received a lot of attention in the literature. Several methods have been applied to optimize the PID controller parameters in an AVR system. Ekinci and Hekimoglu [2] proposed an improved Kidney-inspired algorithm to optimize the transient response of an AVR system. Mosaad, Attia, and Abdelaziz [3] presented a Cuckoo search algorithm and a new time-domain output criterion to introduce a novel PID controller tuning design to be used in the automatic voltage regulator system. Bingul and Karahan [4] used and analysed a whale optimization algorithm to tune the PID controller parameters using root locus and bode diagram. Ekinci, Hekimoglu, and Kaya [5] presented an optimization method named the “salp swarm algorithm” based on artificial intelligence. Bhookya and Jatoth [6] employed a sine-cosine algorithm in a fractional PID controller to optimize its parameters. Ekinci, Hekimoglu, and Eker [7] introduced a Harris Hawks Optimization method as a novel parameter approach to optimize the gains of a PID controller. Spoljarić, Lušetić, and Simovic [8] applied an ant lion optimizer algorithm for optimal tuning of the AVR system in a closed-loop generator control design. Blondin, Sanchis, Sicard, and Dura [9] proposed an optimized gain tuning method for the PID controller based on a fusion between an Ant Colony Optimization algorithm and a Nelder-Mead method. Gong [10] suggested an approach to obtain the optimum values of a PID controller using the Jaya algorithm. Govindan [11] verified the effectiveness of an optimization algorithm based on an evolutionary algorithm to design a PID controller. Kose [12] implemented an optimal Tree-Seed algorithm (TSA) to optimize PID controller gain values. Pachauri [13] was inspired by nature, a Water Cycle algorithm (WCA) is implemented to evaluate the parameters of a PID controller. Zhou, Zhang, Yang, and Ling [14] applied a Water Wave Optimization (WWO). Al Gizi [15] combined particle swarm optimization and fuzzy logic control. Guvenc, Isik, Yigit, and Akkaya [16] proposed a Biogeography-Based Optimization (BBO) as a self-tuning method to find the best parameters for the PID controller in an AVR system.

As indicated, there are many valuable studies in the literature on the control of AVR systems that are crucial to generators. Control of the system must be fast, responsive, and robust. In addition, the controller design for practical

applications should be implemented effectively. For this reason, it is imperative to determine the most appropriate control method for the AVR system, as in any system. To meet these requirements, the parameters of the PID controller used for controlling an AVR system in this study were estimated with the Stability-Based Artificial Intelligence Method (S-AIM) and applied to the system and compared to other control methods in the literature. According to the results obtained, it is demonstrated that the method we propose is highly superior to other methods in terms of rise time, settling time, and overshoot.

II. PID-BASED AVR SYSTEM

The AVR is a solid-state electronic component that is part of the generator's excitation mechanism. The entire excitation system is designed to build voltage at starting and maintain it at a predetermined level during operation. The AVR measures the voltage in the main stator, and if the voltage decreases when a load is applied, it boosts the excitation to raise the voltage again, and in case if the voltage rises above a required value, the AVR reduces the excitation to reduce the voltage again [17].

An AVR is a feedback control system that tests the

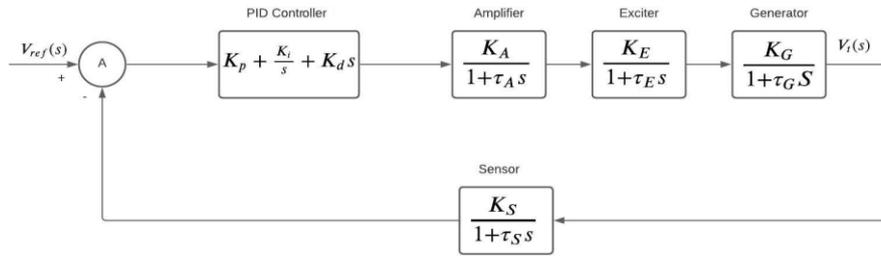


Fig. 1. Block diagram of the PID-based AVR System.

A. Amplifier

An amplifier aims to increase the magnitude of the error signal obtained when the terminal voltage is compared to the reference voltage. The first-order transfer function system can be used to design the amplifier

$$G_{\text{Amplifier}}(s) = \frac{K_A}{1 + \tau_A \times s}. \quad (1)$$

B. Exciter

The generator excitation component uses the input as the amplified error voltage to regulate the excitation voltage (DC) in a synchronous machine rotating field. Gain (K_E) and time constant (τ_E) can be used to model the transfer function

$$G_{\text{Exciter}}(s) = \frac{K_E}{1 + \tau_E \times s}. \quad (2)$$

C. Generator

Gain (K_G) and time constant (τ_G) can be used to model the transfer function of the generator

$$G_{\text{Generator}}(s) = \frac{K_G}{1 + \tau_G \times s}. \quad (3)$$

generator output voltage (V_g), compares it to a set point (V_{ref}) and produces an error signal to regulate the generator excitation. The main functions of an AVR system are maintaining the generator output voltage, regulating and controlling an overvoltage in parallel generators by stabilizing the response time in all generators, and being used as a safety and protection system [18].

The AVR system is presented with four main components. These components are mainly Amplifier, Exciter, Generator, and Sensor as introduced in Table I and shown in Fig. 1 [19], [20].

TABLE I. SELECTED VALUES OF GAIN AND TIME IN THE AVR SYSTEM.

Component	Range	Used values
Amplifier	$10 \leq K_A \leq 40$ $0.02 \leq \tau_A \leq 0.1$	$K_A = 10$ $\tau_A = 0.1$
Exciter	$1 \leq K_E \leq 10$ $0.4 \leq \tau_E \leq 1$	$K_E = 1$ $\tau_E = 0.4$
Generator	$0.7 \leq K_G \leq 1$ $1 \leq \tau_G \leq 2$	$K_G = 1$ $\tau_G = 1$
Sensor	$0.9 \leq K_S \leq 1.1$ $0.001 \leq \tau_S \leq 0.06$	$K_S = 1$ $\tau_S = 0.01$

D. Sensor

The first-order transfer function can be used to present the gain of a sensor

$$G_{\text{Sensor}}(s) = \frac{K_S}{1 + \tau_S \times s}. \quad (4)$$

The range of gain and time values of an AVR system are summarized as shown in Table I.

Due to its simple implementation and high-performance behaviour, the PID controller is counted as the most commonly utilized controller in industrial systems. The main objective of the PID controller is to improve the dynamic response performance of a system and to reduce or eliminate steady-state errors.

The PID controller transfer function can be determined as in (5)

$$G_{\text{PID}}(s) = K_p + \frac{K_i}{s} + K_d \times s. \quad (5)$$

A block diagram of an AVR system based on a PID controller is shown in Fig. 1.

The transfer function of a PID-based AVR system can be extracted from Fig. 1 and shown as in (6)

$$G_{AVR}(s) = \frac{G_{PID}(s) \times G_A(s) \times G_E(s) \times G_G(s)}{1 + G_{PID}(s) \times G_A(s) \times G_E(s) \times G_G(s) \times G_S(s)} \quad (6)$$

III. PROPOSED METHOD

To implement the stability-based artificial intelligence method (S-AIM), the K_p , K_i , and K_d PID controller parameters are applied to the system randomly, data are collected, and then artificial intelligence studies are carried out.

A. Data Collection System

For the proposed method, a data collection system, shown in Fig. 2, was first created.

As shown in Fig. 2, the set value for the AVR system to be controlled is recorded continuously, provided that the

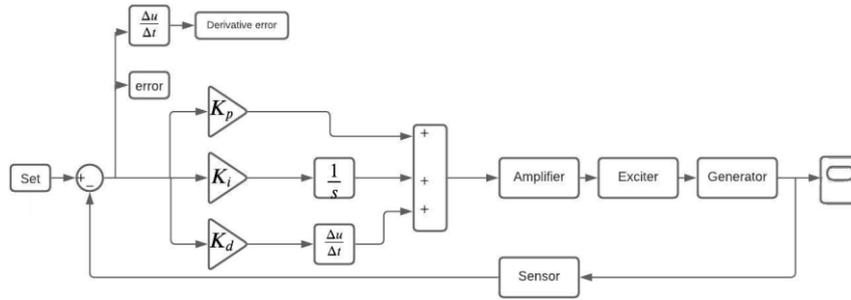


Fig. 2. Data collection system.

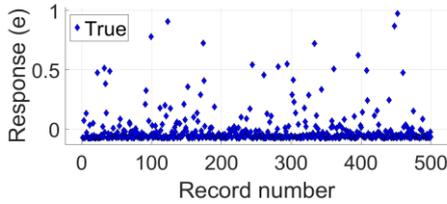


Fig. 3. Error distribution.

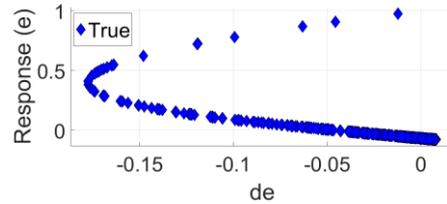


Fig. 4. Data review of e opposed de .

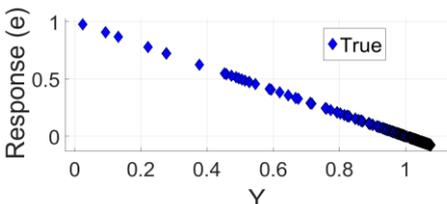


Fig. 5. Data review of e opposed Y .

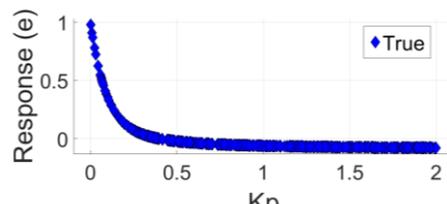


Fig. 6. Data review of e opposed K_p .

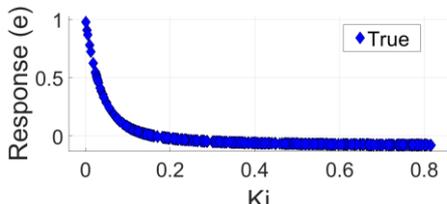


Fig. 7. Data review of e opposed K_i .

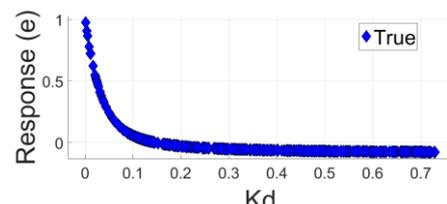


Fig. 8. Data review of e opposed K_d .

It has been observed that the error that occurred is in fact sensitive to the parameters of K_p , K_i , and K_d , as shown in Figs. 3–8.

B. Artificial Intelligence Studies

To determine the best artificial intelligence method in WEKA software [22], Linear Regression, Gaussian Processes, Multilayer Perceptron, Simple Linear Regression, SMOReg, IBK, K-Star algorithm, LWL,

error (e), the derivative of the error (de), K_p , K_i , K_d , and output (Y) values meet the following stability criteria, which are of great importance for the control of the system. In the code written, random values between 0 and 1000 were produced for the controller parameters K_p , K_i , and K_d and the characteristic equation of the system was examined each time with the generated values, and only the values that made the system stable were recorded. As a result of the experiments, it was decided that it would be most appropriate to choose the number of iterations as 1000 and the simulation time for each iteration as 3 seconds. As a result of the data collection simulation, 500 data were obtained that made the system stable. All data collection was performed in the MATLAB 2020b software [21]. The review of the data obtained is presented in Figs. 3–8.

Additive Regression, Randomizable Filtered Classifier, Stacking, Weighted Instances Handler Wrapper, Decision Table, Random Tree, Random Forest, and Bagging methods were applied, and Bagging was the most appropriate method in terms of training Correlation Coefficient, Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error, Root Relative Squared Error values, and simulation suitability. Training results for the Bagging method if the output is selected as an error for general inspection of the

artificial intelligence model are presented in Fig. 9 and Fig. 10.

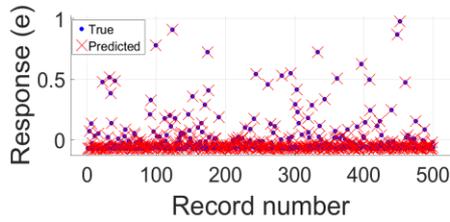


Fig. 9. True predicted data.

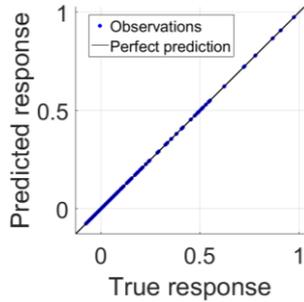


Fig. 10. Observations-predictions.

In Fig. 9 and Fig. 10, the Bagging method was observed to be almost identical to the estimated value and the actual values based on the training results.

C. Bagging

Bagging or Bootstrap aggregation is a method introduced by Breiman [23], which is useful for many regression and classification methods that improve prediction efficiency by minimizing the variance associated with prediction [24], [25].

Conceptual, bagging indicators is a system for producing different forms of an indicator and utilizing these to get a totalled indicator. The conglomeration mid points over the forms while anticipating a numerical result and does a majority vote while foreseeing a class. The different adaptations are framed by making bootstrap recreates of the learning set and utilizing these as new learning sets. Tests on real and simulated data sets using order and regression trees and the determination of subsets in linear regression demonstrate that bagging can give significant increases in accuracy. The basic component is the precariousness of the expectation strategy. On the off chance that irritating the learning set can bring about critical changes in the indicator developed, then bagging can enhance accuracy [23].

While k-closest neighbour strategies are steady, it must be moted that neural networks, grouping and regression trees, and subset choice in linear regression are unstable [26]. The

confirmation, both test and hypothetical, is that bagging can push a decent yet unstable method a critical step towards optimality. Then again, it can marginally corrupt the execution of stable techniques. There have been late works in the writing with a portion of the kind of bagging. Specifically, there has been some work on averaging and voting over numerous trees. Buntine [27] gave a Bayesian approach, Kwok and Carter [28] utilized using so as to vote over various trees created elective parts, and Heath, Kasif, and Salzberg [29] used voting over different trees produced by option angled parts. Dietterich and Bakiri [30] demonstrated that a strategy to encode numerous class issues into countless class issues expands the accuracy. There is some common characteristic of this thought with bagging.

The WEKA 3.9.5 software [22] is used to create the Bagging architecture. Several attempts have been made to select the parameters that will yield the best result. As a result of these trials, the Bagging parameters that give the best estimate are as follows: REPTree was chosen as the Classifier. 1000 iterations were made by selecting the Seed coefficient as “1.0” and Bag size Percent “100”.

D. PID Controller Parameter Prediction System

Using the obtained data set, when the weights produced for the output are considered in the WEKA software [22], it is observed that the most effective controller parameters are K_p , K_i , and K_d , respectively. Accordingly, the data set was rearranged so that its output was K_p and training was carried out using the Bagging method. Similarly, in the next step, the data set was rearranged so that the output would be K_i , and training was carried out. In the last step, the data set was rearranged so that the output was K_d and the training was carried out. Table II shows the training results.

Using the weights obtained at the end of the training, the K_p , K_i , and K_d values were predicted, respectively, so that the output (Y) is 1, the error (e) is 0, and the derivative of the error (de) is 0. The predicted K_p , K_i , and K_d parameters with the parameters used by other methods are presented in Table III. The Stability-Based Artificial Intelligence Method (S-AIM) flow diagram is given in Fig. 11.

TABLE II. PERFORMANCE OF TRAINING SET FOR PID PARAMETERS.

Training Results	PID Parameters		
	K_p	K_i	K_d
Correlation Coefficient	0.9999	0.9999	0.9999
Mean Absolute Error	0.006	0.0024	0.0022
Root Mean Squared Error	0.0077	0.0031	0.0031
Relative Absolute Error	1.2292 %	1.2255 %	1.2561 %
Root Relative Squared Error	1.3728 %	1.3593 %	1.5043 %

TABLE III. PID PARAMETERS FOR DIFFERENT CONTROLLERS.

Method	PID Parameters		
	K_p	K_i	K_d
Proposed S-AIM	0.73600	0.23700	0.15100
IKA-PID [2]	1.04260	1.00930	0.59990
Jaya-PID [10]	0.60051	0.41386	0.20138
TSA-PID [12]	1.12810	0.95670	0.56710
WWO-PID [14]	1.22990	0.84740	0.39210
BBO-PID [16]	1.24640	0.58930	0.45960

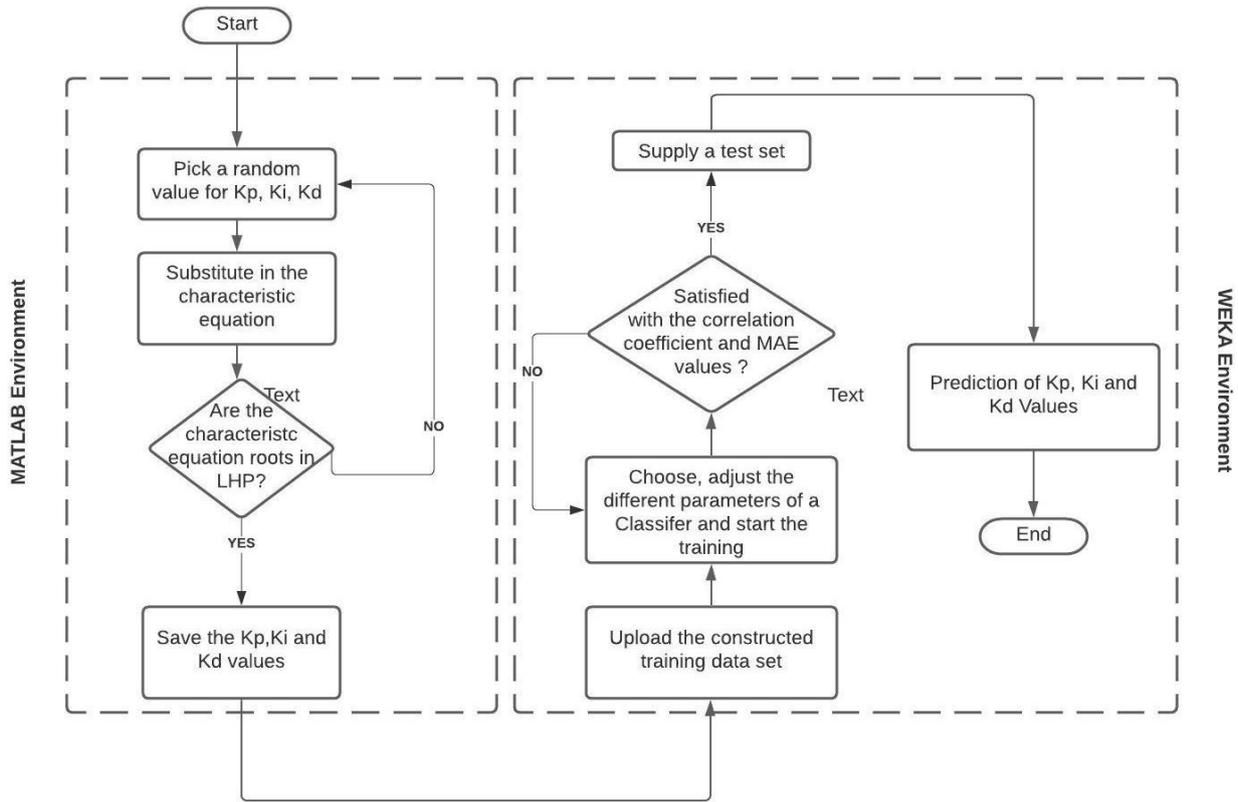


Fig. 11. Flowchart of the proposed method.

IV. RESULTS AND DISCUSSION

All simulations were performed using MATLAB/Simulink 2020b [21]. The suggested approach has been applied to the AVR system and compared to other control methods in the literature. The Stability-Based Artificial Intelligence Method (S-AIM) clearly outperforms the other alternatives, as shown in Fig. 12.

The comparison of the simulation results (rising time, settling time, and overshoot of different controllers) for the terminal voltage step response of the AVR system with various controllers is depicted in Table IV.

It is observed in Fig. 12 and Table IV that the proposed Stability-based Artificial Intelligence Method (S-AIM) method for controlling the AVR system outperforms the other methods.

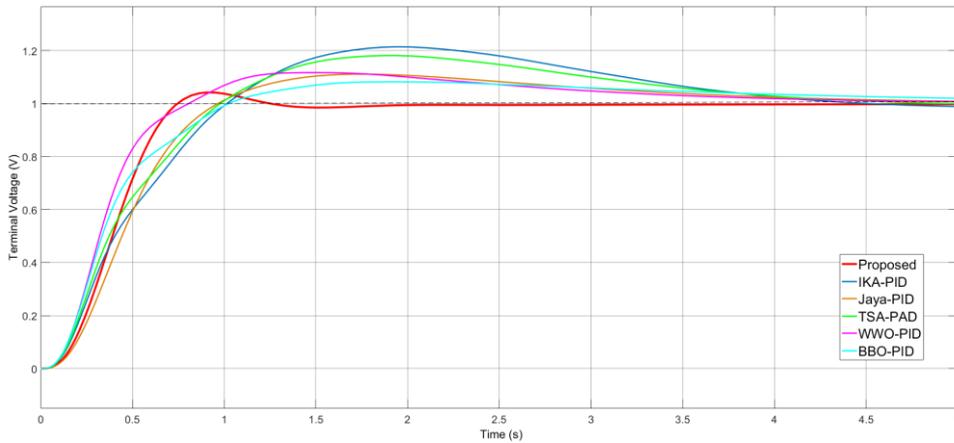


Fig. 12. Comparison of the responses of the AVR system.

TABLE IV. COMPARISON OF TRANSIENT RESPONSE FOR DIFFERENT CONTROLLERS.

Method	Rise time (s)	Settling time (s)	Overshoot (%)
Proposed S-AIM	0.439	0.675	4.737
IKA-PID[2]	0.693	3.667	21.341
Jaya-PID[10]	0.584	3.173	10.556
TSA-PID[12]	0.657	3.608	18.452
WWO-PID[14]	0.429	2.932	11.789
BBO-PID[16]	0.644	3.322	8.152

V. CONCLUSIONS

The Stability-Based Artificial Intelligence Method (S-AIM) is newly used to determine optimal PID controller tuning parameters for an AVR system. The Stability-Based Artificial Intelligence Method (S-AIM) was compared to IKA, Jaya, TSA, WWO, and BBO using some properties of transient responses, such as rising time, settling time, and overshoot. When the results are examined in detail; the rise

and settling times are considered together, it is seen that the proposed method is approximately 2 seconds faster than the closest competitor method (WVO-PID [14]). Similarly, in terms of overshoot value, it is seen that the proposed method gives a better response by approximately 4 % than the closest competitor method (BBO-PID [16]). The proposed Stability-Based Artificial Intelligence Method (S-AIM)-based PID controller in the AVR system, according to simulation results, is reliable and resilient to uncertainties. It performs exceptionally well and outperforms the other methods.

Certainly, the proposed Stability-Based Artificial Intelligence Method (S-AIM) can be adapted to a wide range of different systems, and every issue that needs to be decided during the adaptation, from the data collection stage to the artificial intelligence method to be used, can be determined specifically for the system to be controlled.

As future work, we aim to apply the Stability-Based Artificial Intelligence Method (S-AIM) method, which is a new method proposed in this study, to different systems (theoretically and practically) and to develop original software with a user-friendly interface to ensure effective use of the method.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

- [1] H. Tian, Y. Liu, Y. Zhang, and X. Qin, "An improved secondary voltage control method for half-wavelength transmission system", *International Journal of Electrical Power & Energy Systems*, vol. 115, p. 105417, 2020. DOI: 10.1016/j.ijepes.2019.105417.
- [2] S. Ekinici and B. Hekimoglu, "Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system", *IEEE Access*, vol. 7, pp. 39935–39947, 2019. DOI: 10.1109/ACCESS.2019.2906980.
- [3] A. M. Mosaad, M. A. Attia, and A. Y. Abdelaziz, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system", *Ain Shams Engineering Journal*, vol. 10, no. 4, pp. 755–767, 2019. DOI: 10.1016/j.asej.2019.07.004.
- [4] Z. Bingul and O. Karahan, "A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system", *Journal of the Franklin Institute*, vol. 355, no. 13, pp. 5534–5559, 2018. DOI: 10.1016/j.jfranklin.2018.05.056
- [5] S. Ekinici, B. Hekimoglu, and S. Kaya, "Tuning of PID controller for AVR system using salp swarm algorithm", in *Proc. of 2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018, pp. 1–6. DOI: 10.1109/IDAP.2018.8620809.
- [6] J. Bhookya and R. K. Jatoth, "Optimal FOPID/PID controller parameters tuning for the AVR system based on sine-cosine-algorithm", *Evolutionary Intelligence*, vol. 12, no. 4, pp. 725–733, 2019. DOI: 10.1007/s12065-019-00290-x.
- [7] S. Ekinici, B. Hekimoglu, and E. Eker, "Optimum design of PID controller in AVR system using harris hawks optimization", in *Proc. of 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2019, pp. 1–6. DOI: 10.1109/ISMSIT.2019.8932941.
- [8] T. Spoljarić, C. Lušetić, and V. Simovic, "Optimization of PID controller in AVR system by using ant lion optimizer algorithm", in *Proc. of 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018, pp. 1522–1526. DOI: 10.23919/MIPRO.2018.8400274.
- [9] M.-J. Blondin, J. Sanchis, P. Sicard, and J. M. H. Dura, "New optimal controller tuning method for an AVR system using a simplified ant colony optimization with a new constrained Nelder–Mead algorithm", *Applied Soft Computing*, vol. 62, pp. 216–229, 2018. DOI: 10.1016/j.asoc.2017.10.007.
- [10] C. Gong, "Jaya algorithm-optimized PID controller for AVR system", in *Advances in Intelligent, Interactive Systems and Applications. IISA 2018. Advances in Intelligent Systems and Computing*, vol. 885. Springer, Cham, 2019. DOI: 10.1007/978-3-030-02804-6_52.
- [11] P. Govindan, "Evolutionary algorithms-based tuning of PID controller for an AVR system", *International Journal of Electrical & Computer Engineering*, vol. 10, no. 3, pp. 3047–3056, 2020. DOI: 10.11591/ijece.v10i3.pp3047-3056.
- [12] E. Kose, "Optimal control of AVR System with tree seed algorithm-based PID controller", *IEEE Access*, vol. 8, pp. 89457–89467, 2020. DOI: 10.1109/ACCESS.2020.2993628.
- [13] N. Pachauri, "Water cycle algorithm-based PID controller for AVR", *COMPEL*, vol. 39, no. 3, pp. 551–567, 2020. DOI: 10.1108/COMPEL-01-2020-0057.
- [14] Y. Zhou, J. Zhang, X. Yang, and Y. Ling, "Optimization of PID controller based on water wave optimization for an automatic voltage regulator system", *Information Technology and Control/Informacinės technologijos ir valdymas*, vol. 48, no. 1, pp. 160–171, 2019. DOI: 10.5755/j01.itc.48.1.20296.
- [15] A. J. H. Al Gizi, "A particle swarm optimization, fuzzy PID controller with generator automatic voltage regulator", *Soft Computing*, vol. 23, no. 18, pp. 8839–8853, 2019. DOI: 10.1007/s00500-018-3483-4.
- [16] U. Guvenc, A. H. Isik, T. Yigit, and I. Akkaya, "Performance analysis of biogeography-based optimization for automatic voltage regulator system", *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24, no. 3, pp. 1150–1162, 2016. DOI: 10.3906/elk-1311-1111.
- [17] S. Vachirasricirikul, I. Ngamroo, and S. Kaitwanidvilai, "Coordinated SVC and AVR for robust voltage control in a hybrid wind-diesel system", *Energy Conversion and Management*, vol. 51, no. 12, pp. 2383–2393, 2010. DOI: 10.1016/j.enconman.2010.05.001.
- [18] E. Celik and R. Durgut, "Performance enhancement of automatic voltage regulator by modified cost function and symbiotic organisms search algorithm", *Engineering Science and Technology, an International Journal*, vol. 21, no. 5, pp. 1104–1111, 2018. DOI: 10.1016/j.jestech.2018.08.006.
- [19] S. Panda, B. K. Sahu, and P. K. Mohanty, "Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization", *Journal of the Franklin Institute*, vol. 349, no. 8, pp. 2609–2625, 2012. DOI: 10.1016/j.jfranklin.2012.06.008.
- [20] S. Chatterjee and V. Mukherjee, "PID controller for automatic voltage regulator using teaching–learning based optimization technique", *International Journal of Electrical Power & Energy Systems*, vol. 77, pp. 418–429, 2016. DOI: 10.1016/j.ijepes.2015.11.010.
- [21] *MATLAB R2020a Reference Manual*, License No. 40827100, 2020.
- [22] *WEKA 3.9.5 Reference Manual*, 2021.
- [23] L. Breiman, "Bagging predictors", *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. DOI: 10.1007/BF00058655.
- [24] F. Harrou, A. Saidi, and Y. Sun, "Wind power prediction using bootstrap aggregating trees approach to enabling sustainable wind power integration in a smart grid", *Energy Conversion and Management*, vol. 201, p. 112077, 2019. DOI: 10.1016/j.enconman.2019.112077.
- [25] T. H. Lee, A. Ullah, and R. Wang, "Bootstrap aggregating and random forest", in *Macroeconomic Forecasting in the Era of Big Data. Advanced Studies in Theoretical and Applied Econometrics*, vol. 52. Springer, Cham, 2020. DOI: 10.1007/978-3-030-31150-6_13.
- [26] L. Breiman, "Heuristics of instability and stabilization in model selection", *The Annals of Statistics*, vol. 24, no. 6, pp. 2350–2383, 1996. DOI: 10.1214/aos/1032181158.
- [27] W. Buntine, "Learning classification trees", in *Artificial Intelligence Frontiers in Statistics*. Chapman and Hall, London, 1991, pp. 182–201.
- [28] S. Kwok and C. Carter, "Multiple decision trees", in *Uncertainty in Artificial Intelligence*. Elsevier Science, Amsterdam, 1990, pp. 327–335.
- [29] D. Heath, S. Kasif, and S. Salzberg, "Jaya algorithm-optimized PID controller for AVR system", in *Proc of The Second International Workshop on Multistrategy Learning*, 1993, pp. 1002–1007.
- [30] T. G. Dietterich and G. Bakiri, "Error-correcting output codes: A general method for improving multiclass inductive learning programs", in *Proc. of The Ninth National Conference on Artificial Intelligence (AAAI-91)*, 1991, pp. 572–577.

