An Improved Feed-Forward Backpropagation Neural Network Based on Marine Predators Algorithm for Tuning Automatic Voltage Regulator

Widi Aribowo[1†], **Reza Rahmadian**[1], **Mahendra Widyartono**[1], **Ayusta Lukita Wardani**[1], and **Aditya Prapanca**[2], Non-members

ABSTRACT

This research will discuss the application of an automatic voltage regulator based on the feed-forward back propagation neural network (FFBNN), which is enhanced by the marine predator algorithm (MPA). The marine predators algorithm is a method that adopts marine ecosystem life that is identified in the relationship between predators and prey. MPA is adopting a natural approach to arranging the best food search strategies and finding the latest strategy. The focus of the research is on the performance of speed and rotor angle. The performance of the proposed method will be tested using hidden layer variations. In addition, the proposed method will be compared with the feed-forward backpropagation neural network (FFBNN), cascade-forward backpropagation neural network (CFBNN), Elman recurrent neural network (E-RNN), and Focused Time Delay neural network (FTDNN). The speed and rotor angle of the proposed method have good values. The MPA-FFBNN results are not much different from other methods. The experimental results show that the performance of the proposed method has promising results.

Keywords: Marine predators algorithm, feed-forward backpropagation neural network, automatic voltage regulator, metaheuristic; power system

1. INTRODUCTION

The development of technology, which is increasingly developing, has an impact on all aspects of life [1]. The most dominant is electricity. The electricity supply will increase every year. In an electric system, securing stable voltage values in assorted situations is the most prominent control issue. This is related to power quality and network reliability [2]. When interference occurs at the voltage level in the network. This will cause a system to experience very significant interference. This relates to the dynamics of the system. This will allow a decrease in the performance of the devices connected to the electricity. This will fatally damage the device. Equipment will work effectively and efficiently when conditions are in accordance with the nameplate of the equipment [3].

In power systems, keeping the terminal voltage in good condition is one focus in maintaining the stability of the electrical system [4]. In the power system, the output voltage of the generator will be recognized by the automatic voltage regulator system (AVR) [5]. The automatic voltage regulator has a function to maintain the generator terminal voltage in a stable condition. This voltage regulation uses exciter voltage in the generator. The value of the exciter voltage can be adjusted through the permissible limit. Load changes and high inductance on the generator become problems for achieving a stable and fast response [6].

The settings of the automatic voltage regulator will affect the dynamics and stability of the electricity. To fulfill control aims, the right control method has an important role. Some researchers have dealt with a lot of the control of AVR. Many kinds of control strategies are used to manage AVR, such as proportional–integral–derivative (PID) controllers. PID controllers are widely applied in controls [7, 8]. This is due to its simple structure and strong performance [9]. The disadvantages of PID are delay time and linearity. This has become an obstacle in the industry.

Complexity becomes an obstacle to finding the optimum value. The development of research on AVR is pushing towards artificial intelligence. Some AVR studies use machine learning methods, such as Particle Swarm Optimization (PSO) [10]–[12]. Harmony Search Algorithm (HAS) [13], Monarch Butterfly Optimization (MBO) [14], Kidney-Inspired Algorithm (KA) [15], Harris Hawks Optimization (HHO) [16], Salp Swarm Algorithm (SSA) [17]–[19], Ant Lion Optimizer Algorithm (ALO) [20], [21], Tree Seed Algorithm (TSA) [22], and Grasshopper Optimization Algorithm [23]–[25].

In this paper, we present improvements in the performance of feed-forward backpropagation neural networks

Manuscript received on November 15, 2022; revised on January 15, 2023; accepted on March 8, 2023. This paper was recommended by Associate Editor Yuttana Kumsuwan.

[1]The authors are with Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia,
[2]The author is with Department of Computer Engineering, Universitas Negeri Surabaya, Indonesia,
[†]Corresponding author: widiaribowo@unesa.ac.id

©2023 Author(s). This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License. To view a copy of this license visit: https://creativecommons.org/licenses/by-nc-nd/4.0/.

Digital Object Identifier: 10.37936/ecti-eec.2023212.249830

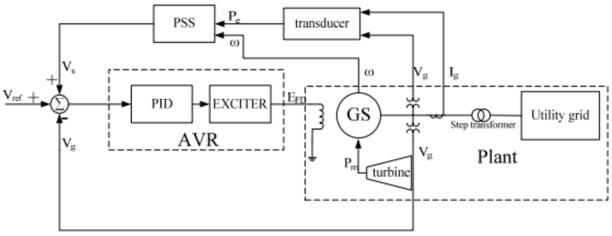

Fig. 1*: System structure.*

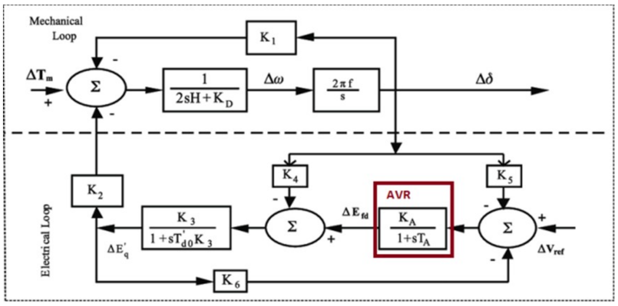

Fig. 2*: Single machine block diagram [28].*

Table 1*: Symbol list of mechanical loop.*

Parameter	Function
K_1	Heffron-Phillips model coefficients
H	Shaft inertia constant
K_D	Damping constant
T_m	Mechanical torque from turbine
ω	Rotor angular speed
δ	Rotor angle

Table 2*: Symbol list of electrical loop.*

Parameter	Function
$K_2 - K_6$	Heffron-Phillips model coefficients
K_A	DC gain of the AVR
T_A	Time constant of the AVR
ΔV_{ref}	Reference voltage of the AVR
ΔE_{fd}	Field winding voltage that from AVR output
$\Delta E_q'$	Excited voltage
T_{d0}'	d-Axis transient time constant (provided by manufacturer)

(FFBNN) using a nature-inspired optimization approach, namely the marine predator algorithm (MPA). This was inspired by the strategy of foraging marine predators [26]. In this study, we are using MPA to optimize FFBNN parameters to tune AVR. In this research, MPA-FFBNN was applied to a single-machine system. The application of MPA-FFBNN on a single machine system is emphasized with respect to feed-forward back propagation neural networks (FFBNN), cascade-forward back propagation neural networks (CFBNN), Elman-recurrent neural networks (E-RNN), and focused time delay neural networks (FTDNN) performance against speed and rotor angle.

2. METHODS

2.1 Automatic Voltage Regulator

The generator is one of the reactive power sources, and the major source of reactive power control for the generator is to manage excitation through an AVR. Increasing stability and ensuring the grade of the electricity system are seriously affected by the excitation control [20]. The control of the generator consists of the AVR and PSS. An automatic voltage regulator (AVR) and power system stabilizer (PSS) were applied to fix the transient stability of the generator. The excitation is used by AVR to maintain synchronous generator terminal voltages at the appropriate level. Figure 1 displays the system structure. AVR serves to maintain the stability of the power system in a steady state. In a transient state, the generator will experience a disturbance that results in a decrease in the generator terminal voltage [27].

In this research, generators are modeled in Heffron-Phillips. Heffron-Phillips model has two loops, namely, mechanical and electrical loops. Heffron-Phillips modeling can be seen in Figure 2. All variables and parameters of the Heffron-Phillips model are summarized in Table 1.

2.2 Feed-Forward Backpropagation Neural Network

A neural network is a network of neuron units called nodes. This computational method is already widely used in the fields of classification, optimization, control theory, and to solve regression problems. Artificial neural networks are very effective for solving classification problems by using detection and identification of targets for processing. Artificial neural networks have a dynamic character that is often used. This dynamic character is achieved by adjusting the weight of the desired reference. Weighting settings are carried out to get the desired output. Weighting arrangements are known as learning [29].

Feed-forward back propagation neural networks are one of the most popular artificial neural networks algorithms for engineering applications. The feed-forward backpropagation neural network architecture consists of an input layer, an output layer, and one or more hidden neuron layers. The feed-forward backpropagation neural network architecture can be seen in Figure 3. Each layer has several neurons, and each neuron is related to an adjustable weighting connection. The activation function in the hidden layer neurons authorizes the neural network to suit a general approach. The training process is to adjust the weighting. This allows the network to produce the desired response to the given input. Forward and backward training algorithms can be applied to minimize errors. This uses the gradient descent algorithm to decrease it. It is the average squared difference between the desired output network

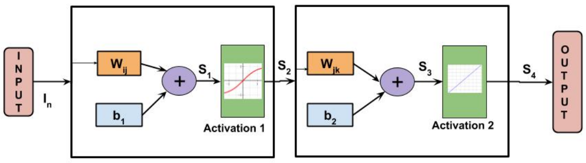

***Fig. 3**: FFBNN Structure.*

and actual [30].

In the feed-forward stage, input data (I_n) will be processed in a hidden layer. It is linking to neuron j with W_{ij}. W_{ij} . Is the connection weighty? Summation functions are recapitulating the inputs: weight (W_{ij}) and bias (b_1) in layer 1. S_2 (t) is the sigmoid function.

$$S_1(t) = \sum_{i=1}^{j} W_{ij} I_n(t) + b_1 \tag{1}$$

$$S_2(t) = f(S_1, (t)) = \frac{1}{1 + \exp^{S_1}} \tag{2}$$

In layer 2, the output from layer 1 (S_2 (t)) is connected to neurons k with connecting weights in layer 2 (W_{jk}). The addition function of layer 2 is sum of output layer 1 (S_2 (t)), weight (W_{jk}) and bias (b_2).

$$S_3(t) = \sum_{j=1}^{k} W_{jk} S_2(t) + b_2 \tag{3}$$

$$S_4(t) = f(S_3, (t)) = \frac{1}{1 + \exp^{S_3}} \tag{4}$$

Each output neuron accepts a target example that corresponds to practicing an example of input in back-propagation algorithm steps. The error estimation is obtained by multiplying by the derivation of the activation function.

$$\delta_k = (t_i - S_4) f'(S_3) \tag{5}$$

2.3 Marine Predators Algorithm

The marine predator algorithm (MPA) is the latest algorithm inspired by the behavior of predators and prey patterns in nature. MPA has an algorithm that naturally manages optimal search strategies and rate policies between predators and prey in marine ecosystems. The creatures usually aim to find their food and keep searching. Both prey and predator are seen as symbols of search because predators are looking for prey. Meanwhile, the prey itself looks for food . MPA has a character similar to most of the metaheuristics. The population-based MPA algorithm has the following uniform initialization at the beginning:

$$Z_0 = Z_{min} + rand \left(Z_{max} - Z_{min} \right) \tag{6}$$

where Z_{min} and Z_{max} are the lower and upper bounds for variables. rand is a uniform random vector in the range of 0 to 1.

According to the theory of survival of the fittest, the highest predator is a predator who is proficient in foraging. This predator is most worthy of being called elite. In the MPA algorithm, prey and predators are mentioned as agents in the search. The two variables are initialized in two matrices, namely the elite and prey matrices. Both matrices must be identified.

$$Elite = \begin{bmatrix} U^I_{1.1} & U^I_{1.2} & \cdots & U^I_{1.d} \\ U^I_{2.1} & U^I_{2.2} & \cdots & U^I_{2.d} \\ \vdots & \vdots & \vdots & \vdots \\ U^I_{n.1} & U^I_{n.2} & \dots & U^I_{n.d} \end{bmatrix} \tag{7}$$

where U is the best predator that is duplicated n times with dimension d to form an elite matrix. Predators and prey are assumed to be search agents with algorithms that predators use to search for prey and prey for their own food. Elite has an algorithm to replace the top predators with better ones. On the other hand, prey also shares the same matrix with predators. Prey become predator references to determine their position. The initiation will make the initial prey in accordance with predators to build an elite. Prey matrices are as follows:

$$Prey = \begin{bmatrix} P^I_{1.1} & P^I_{1.2} & \cdots & P^I_{1.j} \\ P^I_{2.1} & P^I_{2.2} & \cdots & P^I_{2.j} \\ \vdots & \vdots & \vdots & \vdots \\ P^I_{i.1} & P^I_{n.2} & \dots & P^I_{i.j} \end{bmatrix} \tag{8}$$

where $P^I_{i.j}$ presents the j-th dimension of the i-th prey. The metaheuristik has the main goal is to find the optimal solution. Every initialization will be updated following the algorithm applied. The MPA algorithm has three main stages in doing optimization by considering the ratio of velocity and duplication of its constituent elements.

Phase 1: High-Velocity Ratio ($v \geq 10$)

This phase is called the phase with a high speed ratio. Prey is exploring the area to find food. On the other hand, predators are waiting and monitoring the movements of prey. This scenario occurs at the beginning of an iteration. At a high speed ratio, the best strategy is that the predator does not move. This phase has an algorithm that is formulated as follows:

$$while\ iter < \frac{1}{3} \times \text{max\_iter}$$

$$\overrightarrow{Ss_i} = \overrightarrow{R_b} \otimes \left(\overrightarrow{Elite_i} - \overrightarrow{R_b} \otimes \overrightarrow{Prey_i} \right) \ i = 1, 2 \ldots .n \tag{9}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P \times \overrightarrow{R} \otimes \overrightarrow{Ss_i} \tag{10}$$

The $\overrightarrow{R_b}$ notation is a vector containing random numbers following the normal distribution that represents the movements of the Brownian. This notation $\otimes$ shows possible multiplications. This is RB by the prey, which imitates the motion of the prey. R is a uniformly conditioned random number vector with values ranging from 0 to 1, and P = 0.5 is a constant number. The scenario occurs when the size of the velocity movement is large and occurs in the initial third of the iteration.

Phase 2: Unit Velocity Ratio ($v \approx 1$)

The scenario occurs when predators and prey have the same speed. The scenario occurs in the intermediate phase. In this phase, the character of exploration changes to exploitation. This results in the same composition between the tasks of exploration by prey and exploitation by predators. The predators mimic brownian movements and duplicate Levy flight models.

$$while\ \frac{1}{3} \times \text{max\_iter} < \text{iter} < \frac{2}{3} \times \text{max\_iter}$$

- For the first half of the population

$$\overrightarrow{Ss_i} = \overrightarrow{R_L} \otimes \left(\overrightarrow{Elite_i} - \overrightarrow{R_L} \otimes \overrightarrow{Prey_i}\right) \quad i = 1, 2 \ldots .n/2 \tag{11}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P \times \vec{R} \otimes \overrightarrow{Ss_i} \tag{12}$$

where $\overrightarrow{R_L}$ is a random vector formed from the levy distribution that describes the levy movement? This session can support exploitation.

- For the second half of the population

$$\overrightarrow{Ss_i} = \overrightarrow{R_b} \otimes \left(\overrightarrow{R_b} \otimes \overrightarrow{Elite_i} - \overrightarrow{Prey_i}\right) \; i = n/2, \ldots .n \tag{13}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P \times CF \otimes \overrightarrow{Ss_i} \tag{14}$$

$$CF = \left(1 - \frac{Iter}{Max\_iter}\right)^{\left(2 \frac{Iter}{Max\_iter}\right)} \tag{15}$$

where CF is an adaptive controller that is used to regulate the steps of predator mobility.

Phase 3: In Low-Velocity Ratio ($v = 0.1$)

In the low-velocity ratio phase, the predator is moving faster than the prey. This phase is at the end of the scenario and is identical to high exploitation.

$$while\ iter > \frac{2}{3} \times \text{max\_iter}$$

$$\overrightarrow{Ss_i} = \overrightarrow{R_L} \otimes \left(\overrightarrow{R_L} \otimes \overrightarrow{Elite_i} - \otimes \overrightarrow{Prey_i}\right) \quad i = 1 \ldots .n \tag{16}$$

$$\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P \times CF \otimes \overrightarrow{Ss_i} \tag{17}$$

The explanation can be summarized by saying that the predator does not move in phase 1, phase 2 movements mimic the Brownian strategy, and phase 3 movements show the Levy strategy. The scenario can also turn prey into other potential predators, e.g., sharks and tuna. Both are predatory fish. On the other hand, tuna can fall prey to sharks. The predators in the sea will experience changes in habits caused by several environmental factors, such as the formation of eddies or the effect of fish aggregating devices (FADs). The effect of FAD is formulated as follows:

$$\overrightarrow{Prey_i} = \begin{cases} \overrightarrow{Prey_i} + CF \\ \times \left[Z_0 = Z_{min} + \vec{R} \otimes \left(Z_{max} - Z_{min}\right)\right] \otimes A \\ \qquad \text{if } r \leq FADs \\ \overrightarrow{Prey_i} \\ + [FADs\ (1 - r) + r] \left(\overrightarrow{Prey_{r1}} - \overrightarrow{Prey_{r2}}\right) \\ \qquad \text{if } r > FADs \end{cases} \tag{18}$$

where the trend value is 0.2. This value indicates the search process is influenced by FAD. R is a uniform random value with a range of 0 to 1. A value is a vector containing the values 0 and 1. This value contains an array that is set up with an algorithm if the value is below 0.2, The value is assumed to be 0. On the other hand, if the value is greater than 0.2. The value will be set to 1. Marine predators are able to record the best position ever achieved. They can remember where they caught their prey. This is an important point of the MPA algorithm. The capability is implemented in memory in MPA.

2.4 The Proposed MPA-FFBNN Model

The AVR settings using the MFA-FFBNN hybrid model will be applied. The steps to adapt MPA-FFBNN for tuning AVR are illustrated in Figure 4. The initialization is the first step for all meta-heuristic algorithms. The initial values of the top predator and the weights of the FFBNN are arranged randomly at the start. Initial weighting is a random value between -1 and 1. The output of the MPA will be a potential weighter for FFBNN. Potential weights can be used on the network. The error is the difference between output and target.

3. RESULTS AND DISCUSSION

To confirm the effectiveness and efficiency of the MPA-FFBNN controller, this section discusses the results of comparisons with the FFBNN, CFBNN, Elman-RNN, and FTDNN algorithms. The parameters of the system are assumed to have the same value. In Figure 5, the novelty of the MPA-FFBNN AVR model has been plugged in to replace the regular AVR.

Performance measurement of AVR will use the maximum overshoot, understood, and settling time of the input. The first experiment is using hidden layer 4. The results of the experiment are the speed response and the rotor angle, which can be seen in Figs. 6 and 7. The results of the comparison between the 4 methods of the 5 methods used show very thin differences in speed and rotor angle. Different results are shown when using the E-RNN algorithm. The application of the proposed method gives different results for maximum undershoot and overshoot on the speed response value. The peak of the undershoot result produces the lowest value, while the peak of the overshoot value produces the highest value. The settling time value for speed response when

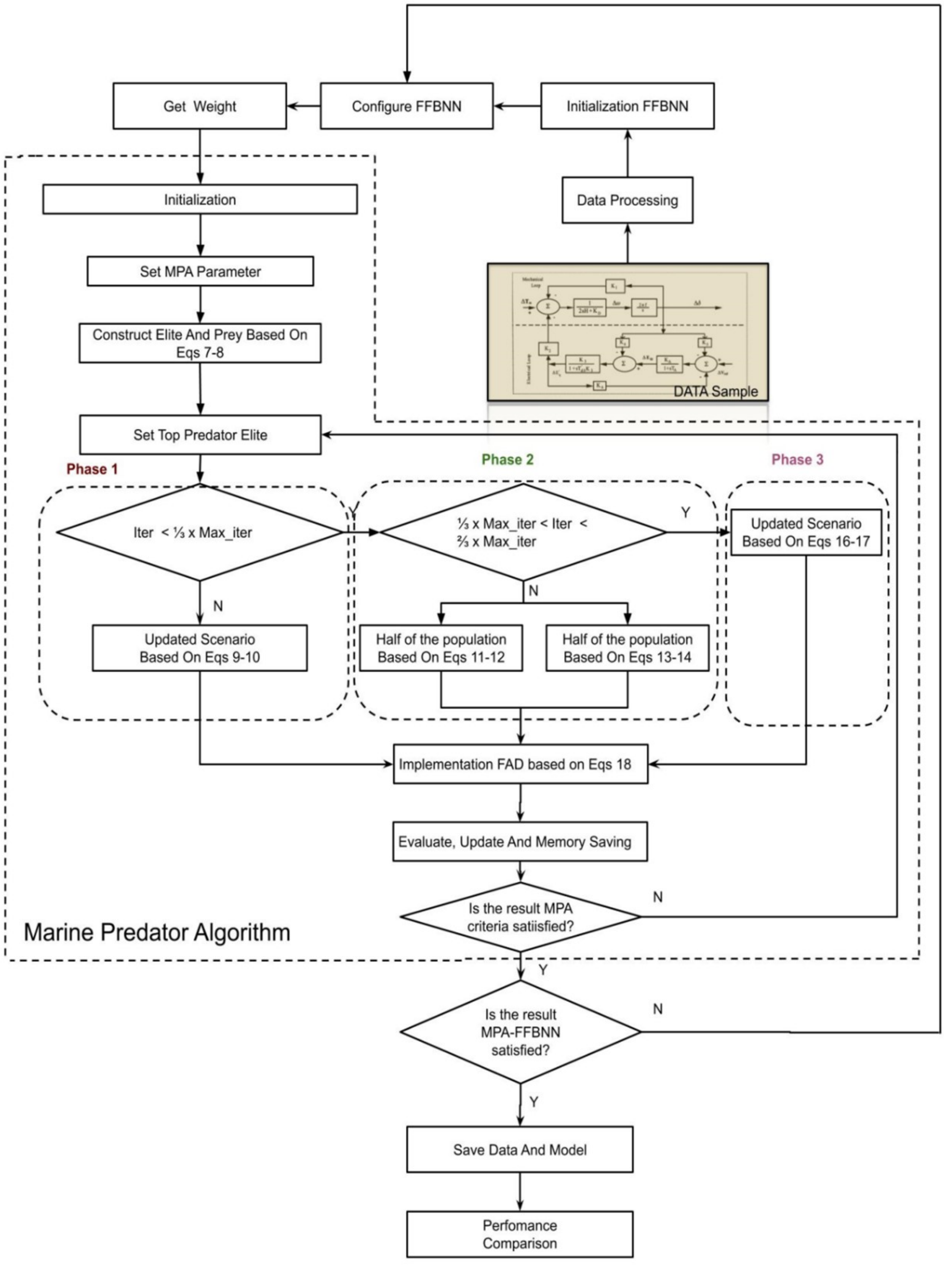

***Fig. 4**: The proposed MPA-FFBNN flowchart.*

***Table 3**: Results of training with 4 layers.*

Methods	Iteration Time	Performance	Iter	Speed Response			Rotor Angle Response		
				Under Shoot	Over Shoot	Time Settlling (s)	Under Shoot	Over Shoot	Time Settlling (s)
FFBNN	00:00:13	7.86e-09	344	-0.6557	0.4791	52	-3.6471	0.8126	48
CFBNN	00:00:02	6.45e-17	4	-0.6554	0.4821	52	-3.6900	0.8138	49
E-RNN	00:00:13	4.68e-10	927	-0.6592	0.5216	56	-3.7732	0.9865	93
FTDNN	00:00:06	9.98e-11	28	-0.65619	0.4789	52	-3.6920	0.8147	49
MPA-FFBNN	00:00:10	1.66e-10	420	-0.6512	0.4835	52	-3.6666	0.8155	49

***Table 4**: Results of training with 8 layers.*

Methods	Iteration Time	Performance	Iter	Speed Response			Rotor Angle Response		
				Under Shoot	Over Shoot	Time Settlling (s)	Under Shoot	Over Shoot	Time Settlling (s)
FFBNN	00:00:02	9.86e-11	54	-0.653	0.4847	48	-3.6471	0.8126	58
CFBNN	00:00:01	2.83e-18	5	-0.655	0.4822	48	-3.6613	0.8053	59
E-RNN	00:00:09	4.74e-10	194	-0.655	0.5204	70	-3.7708	0.9850	92
FTDNN	00:00:23	2.22e-10	731	-0.654	0.4844	48	-3.6190	0.8147	57
MPA-FFBNN	00:00:15	5.28e-12	431	-0.652	0.4821	48	-3.6130	0.8094	54

***Table 5**: Results of training with 12 layers.*

Methods	Iteration Time	Performance	Iter	Speed Response			Rotor Angle Response		
				Under Shoot	Over Shoot	Time Settlling (s)	Under Shoot	Over Shoot	Time Settlling (s)
FFBNN	00:00:02	9.23e-11	45	-0.652	0.4845	54	-3.6904	0.8147	57
CFBNN	00:00:01	3.05e-16	6	-0.655	0.4822	54	-3.6613	0.8053	57
E-RNN	00:00:12	7.94e-10	109	-0.661	0.5220	70	-3.7649	0.9825	95
FTDNN	00:00:23	2.22e-10	731	-0.654	0.4785	54	-3.6881	0.8132	57
MPA-FFBNN	00:00:09	3.74e-12	249	-0.649	0.4859	54	-3.6381	0.8155	65

***Table 6**: Results of training with 16 layers.*

Methods	Iteration Time	Performance	Iter	Speed Response			Rotor Angle Response		
				Under Shoot	Over Shoot	Time Settlling (s)	Under Shoot	Over Shoot	Time Settlling (s)
FFBNN	00:00:03	9.81e-11	73	-0.651	0.4849	57	-3.6867	0.8154	58
CFBNN	00:00:01	6.67e-14	5	-0.655	0.4822	57	-3.6860	0.8121	58
E-RNN	00:00:04	9.99e-11	13	-0.659	0.5219	95	-3.7688	0.9840	98
FTDNN	00:00:23	2.22e-10	731	-0.653	0.4780	57	-3.6893	0.8177	62
MPA-FFBNN	00:00:16	1.02e-11	306	-0.650	0.4823	65	-3.6329	0.8174	66

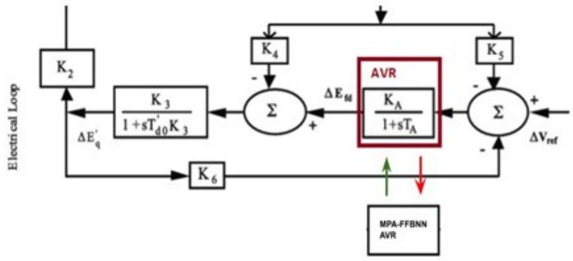

Fig. 5: *The proposed MPA-FFBNN flowchart.*

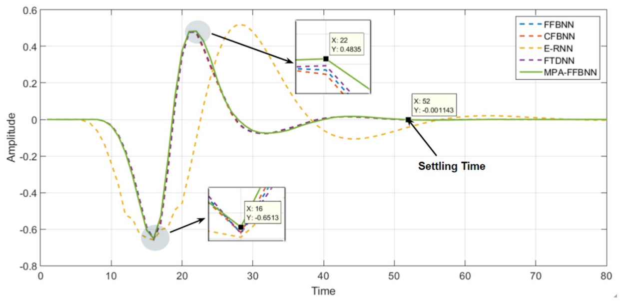

Fig. 6: *Speed response in the regular operating.*

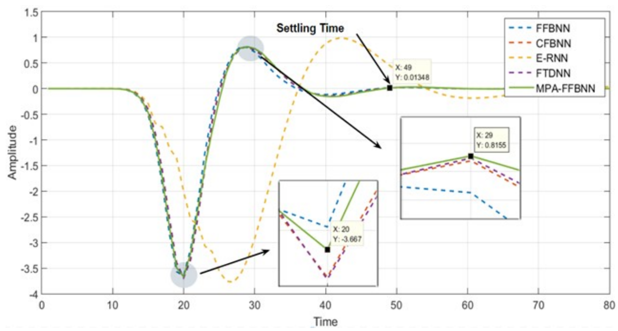

Fig. 7: *Rotor angle response in the regular operating.*

using the MPA-FFBNN method yields the same results as 4 out of 5 methods as a comparison method. The value is 52. Different results are shown when using the E-RNN method. It is 56. This can be seen in Figure 6.

In the measurement of the rotor angel, it was found that the undershoot value of the MPA-FFBNN was good, and the result was -3.667. The results of overshoot from MPA-FFBNN have the highest value of the other methods. The settling time value of the MPA-FFBNN gives the same results as the CFBNN and FTDNN methods. The highest settling time is using the E-RNN method. The value is 93. The lowest value is using FFBNN. The value is 48. This can be seen in Figure 7.

The complete results of applying the four hidden layers can be seen in Table 3. The results of the iteration and performance of the MPA-FFBNN method show better values than the FFBNN, E-RNN, and FTDNN methods. The second test is to apply eight hidden layers. The results of the simulation can be seen in Table 4. The results of overshoot and undershoot on the speed response show the best value when using the MPA-FFBNN method. On the other hand, the results of the rotor angle response are different for overshoot and undershoot. The undershoot results obtained the best value, which is equal to -3,613. While the overshoot value of the MPA-FFBNN is under the CFBNN method. The value is 0.8053.

The third experiment involves applying 12 hidden layers for each method. The speed response for the undershoot of MPA-FFBNN is -0,649. This value is below the CFBNN value. While the overshoot value of MPA-FFBNN is 0.4859. The value is better than the E-RNN method. The values of the rotor angle for overshoot and undershoot in MPA-FFBNN are 0.8155 and -3.681. The undershoot value of MPA-FFBNN is the best. On the other hand, the overshoot value of MPA-FFBNN is still below that of the FFBNN, CFBNN, and FTDNN methods. The details can be seen in Table 5.

The last experiment uses 16 hidden layers. The undershoot result for speed and rotor angle on MPA-FFBNN is the best. The results are -0.650 and -3.629. On the other hand, the overshoot value for the speed response is still below the CFBNN method, and the overshoot value for the rotor angle response is still below the CFBNN and FFBNN methods. The detail can be seen in Table 6.

4. CONCLUSION

This research is to introduce the new marine predator algorithm (MPA) method by Faramarzi (2020). The author conducts research by integrating with existing neural networks, namely FFBNN. The integration between MPA and FFBNN resulted in a hybrid system. Namely MPA-FFBNN. This is installed on an AVR that is on a single machine. The type of single machine is Heffron-Phillips. Tests using hidden layer variations obtained good performance from the proposed method, although not the best. From the experimental results, the proposed method has produced promising results.

REFERENCES

[1] M. S. H. Lipu et al., “Battery Management, Key Technologies, Methods, Issues, and Future Trends of Electric Vehicles: A Pathway toward Achieving Sustainable Development Goals,” *Batteries*, vol. 8, no. 9, Sep. 2022.

[2] W. Aribowo, “A Novel Improved Sea-Horse Optimizer for Tuning Parameter Power System Stabilizer,” *J. Robot. Control*, vol. 4, no. 1, pp. 12–22, Jan. 2023.

[3] K. Eltag, M. S. Aslamx, and R. Ullah, “Dynamic stability enhancement using fuzzy PID control technology for power system,” *Int. J. Control. Autom. Syst.*, vol. 17, no. 1, pp. 234–242, Jan. 2019.

[4] W. Aribowo, Supari, and B. Suprianto, “A hunger game search algorithm for economic load dispatch,” *Int J Artif Intell ISSN*, vol. 11, no. 2, pp. 632-340, Jun. 2022.

[5] M. S. Ayas and E. Sahin, "FOPID controller with fractional filter for an automatic voltage regulator," *Comput. Electr. Eng.*, vol. 90, no. 106895, Mar. 2021.

[6] A. Sikander and P. Thakur, "A new control design strategy for automatic voltage regulator in power system," *ISA Trans.*, vol. 100, pp. 235-243, May 2020.

[7] W. V. Balasaheb and C. Uttam, "Novel Intelligent Optimization Algorithm Based Fractional Order Adaptive Proportional Integral Derivative Controller for Linear Time Invariant based Biological Systems," *J. Electr. Eng. Technol.*, vol. 17, pp. 565–580, 2022.

[8] E. Can, "Extra control coefficient additive ECCA-PID for control optimization of electrical and mechanic system," *Acta Polytech.*, vol. 62, no. 5, pp. 522–530, 2022.

[9] G. Tang, J. Lei, H. Du, B. Yao, W. Zhu, and X. Hu, "Proportional-integral-derivative controller optimization by particle swarm optimization and back propagation neural network for a parallel stabilized platform in marine operations," *J. Ocean Eng. Sci.*, May 2022.

[10] K. Elumalai and S. Sumathi, "Behavior modification of PID controller for AVR system using particle swarm optimization," in *2017 Conference on Emerging Devices and Smart Systems (ICEDSS)*, Mallasamudram, India, 2017, pp. 190-195.

[11] F. Rodrigues, Y. Molina, C. Silva, and Z. Ñaupari, "Simultaneous tuning of the AVR and PSS parameters using particle swarm optimization with oscillating exponential decay," *Int. J. Electr. Power Energy Syst.*, vol. 133, no. 107215, Dec. 2021.

[12] F. Aboura, "Tuning PID Controller Using Hybrid Genetic Algorithm Particle Swarm Optimization Method for AVR System," in *2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, Istanbul, Turkey, 2019, pp. 570-574.

[13] D. K. Sambariya and D. Paliwal, "Optimal design of PIDA controller using harmony search algorithm for AVR power system," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*, New Delhi, India, 2016, pp. 1-6.

[14] D. K. Sambariya and T. Gupta, "Optimal design of PID controller for an AVR system using monarch butterfly optimization," in *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, Indore, India, 2017, pp. 1-6.

[15] S. Ekinci and B. Hekimoğlu, "Improved Kidney-Inspired Algorithm Approach for Tuning of PID Controller in AVR System," *IEEE Access*, vol. 7, pp. 39935-39947, 2019.

[16] A. H. Mary, A. H. Miry, and M. H. Miry, "An Optimal Robust State Feedback Controller for the AVR System-Based Harris Hawks Optimization Algorithm," *Electr. Power Components Syst.*, vol. 48, no. 16–17, pp. 1684–1694, Apr. 2021.

[17] I. A. Khan, A. S. Alghamdi, T. A. Jumani, A. Alamgir, A. B. Awan, and A. Khidrani, "Salp Swarm Optimization Algorithm-Based Fractional Order PID Controller for Dynamic Response and Stability Enhancement of an Automatic Voltage Regulator System," *Electronics*, vol. 8, no. 12, Dec. 2019.

[18] M. A. Labbaf Khaniki, M. Behzad Hadi, and M. Manthouri, "Feedback Error Learning Controller based on RMSprop and Salp Swarm Algorithm for Automatic Voltage Regulator System," in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, 2020, pp. 425-430.

[19] P. Sirsode, A. Tare, and V. Pande, "Design of Robust Optimal Fractional-order PID Controller using Salp Swarm Algorithm for Automatic Voltage Regulator (AVR) System," in *2019 Sixth Indian Control Conference (ICC)*, Hyderabad, India, 2019, pp. 431-436.

[20] R. Pradhan, S. K. Majhi, and B. B. Pati, "Design of PID controller for automatic voltage regulator system using Ant Lion Optimizer," *World J. Eng.*, vol. 15, no. 3, pp. 373-387, Jun. 2018.

[21] T. Spoljarić, C. Lušetić, and V. Simovic, "Optimization of PID controller in AVR system by using ant lion optimizer algorithm," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2018, pp. 1522-1526.

[22] E. Köse, "Optimal Control of AVR System With Tree Seed Algorithm-Based PID Controller," *IEEE Access*, vol. 8, pp. 89457-89467, 2020.

[23] B. Hekimoğlu and S. Ekinci, "Grasshopper optimization algorithm for automatic voltage regulator system," in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, Istanbul, Turkey, 2018, pp. 152-156.

[24] G. Saravanan, D. S. Kumar, A. M. Ibrahim, and C. Karthikeyan, "Enhanced chaotic grasshopper optimization algorithm based PID controller for automatic voltage regulator system," *Int. J. Recent Technol. Eng*, vol. 8, no. 4, pp. 348–354, Nov. 2019.

[25] M. A. L. Khaniki, M. B. Hadi, and M. Manthouri, "Tuning of Novel Fractional Order Fuzzy PID Controller for Automatic Voltage Regulator using Grasshopper Optimization Algorithm," *Majlesi J. Electr. Eng.*, vol. 15, no. 2, pp. 39–45, Mar. 2021.

[26] M. A. A. Al-Qaness, A. A. Ewees, H. Fan, L. Abualigah, and M. Abd Elaziz, "Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea," *Int. J. Environ. Res. Public Health*, vol. 17, no. 10, May 2020.

[27] W. Aribowo, "Focused time delay neural network for tuning automatic voltage regulator," *Emit. Int. J. Eng. Technol.*, vol. 7, no. 1, pp. 34–43, Jun. 2019.

[28] W. Aribowo, "Slime Mould Algorithm Training Neural Network in Automatic Voltage Regulator,"

Trends Sci., vol. 19, no. 3, pp. 1–11, Feb. 2022.

[29] A. Nagathan and M. I. Manimozhi, "Content-Based Image Retrieval System Using Feed-Forward Backpropagation Neural Network," *Int. J. Comput. Sci. Eng.*, Jul. 2013.

[30] R. Jovanović, A. A. Sretenović, and B. D. Živković, "Ensemble of various neural networks for prediction of heating energy consumption," *Energy Build.*, vol. 94, pp. 189-199, May 2015.

Widi Aribowo is a lecturer in the Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia. He is B.Sc in Power Engineering/ Sepuluh Nopember Institute of Technology (ITS) - Surabaya in 2005. He is M.Eng in Power Engineering/ Sepuluh Nopember Institute of Technology (ITS) – Surabaya in 2009. He is mainly research in the power system and control.

Reza Rahmadian received his Bachelor of Applied Science from Electronic Engineering Polytechnic Institute of Surabaya (PENS), Surabaya, Indonesia, in 2006, and his Master of Engineering Science from Curtin University, Australia, in 2013. He is currently a lecturer at the Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia. His research interests include renewable energy.

Mahendra Widyartono received his Bachelor of Engineering from Sepuluh Nopember Institute of Technology (ITS), Surabaya, Indonesia, in 2006, and his Master of Engineering from Brawijaya University, Indonesia, in 2012. He is currently a lecturer at the Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia. His research interests include power system and renewable energy.

Ayusta Lukita Wardani received her Bachelor of Applied Science from Electronic Engineering Polytechnic Institute of Surabaya (PENS), Surabaya, Indonesia, in 2011, and her Master of Engineering from Sepuluh Nopember Institute of Technology (ITS), Indonesia, in 2017. She is currently a lecturer at the Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia. Her research interests include renewable energy.

Aditya Prapanca received his Bachelor of Engineering from Sepuluh Nopember Institute of Technology (ITS), Surabaya, Indonesia, in 2000, and his Master of Computer from Sepuluh Nopember Institute of Technology (ITS), Indonesia, in 2007. He is currently a lecturer at the Department of Computer Engineering, Universitas Negeri Surabaya, Indonesia. His research interests include artificial intelligence.