

# Slime Mould Algorithm Training Neural Network in Automatic Voltage Regulator

**Widi Aribowo**

*Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia*

**(Corresponding author's e-mail: widiaribowo@unesa.ac.id)**

*Received: 11 September 2020, Revised: 23 May 2021, Accepted: 31 May 2021*

## Abstract

The research is proposed a new method of artificial intelligence (AI) to control automatic voltage regulators. A neural network has improved using a metaheuristic method, namely the slime mould algorithm (SMA). SMA has an algorithm based on the mode of slime mold in nature. SMA has characteristics that use adaptive weights to simulate the process to generate feedback from the movement of bio-oscillator-based slime molds in foraging, exploring, and exploiting areas. The performance of the proposed method is focused on speed and rotor angle. To know the competence and potency of the proposed method, a comparison with feed-forward backpropagation neural networks (FFBNN), cascade-forward backpropagation neural networks (CFBNN), Elman-recurrent neural networks (E-RNN), focused time delay neural network (FTDNN), and Distributed Time Delay Neural Network (DTDNN) method are applied. It can be concluded that the proposed method has the best ability. The Proposed method has ability to reduce the overshoot speed with an average value of 0.78 % and the overshoot rotor angle with an average value of 2.134 %.

**Keywords:** Slime mould algorithm, Focused time delay, Neural network, Artificial intelligence, Automatic voltage regulator

## Introduction

Electric load system is a function of the demand for active and reactive power which depends on various variables including time, weather, geography, and economy. The diversity of electricity grid loading has received more attention in recent years. This is influenced by the increase in renewable resources and an increasing number of widespread electric vehicles [1]. The complexity of power systems caused by interconnection and high demand for charging through transmission and distribution networks. Stability is a major problem in the performance of the operation of the power system. Instability will result in island formation, voltage drop, and blackout. This often occurs due to faults, overloads, lightning strikes, etc. [2]. Controllers with high stability performance are designed to be able to maintain stable operation in various operating conditions. This is handy to reduce unwanted conditions.

Electric power systems will be displaying unusual patterns when experiencing major disruptions. This depends on the system structure, loading conditions, and the location of the disturbance. The modern electric power systems have a high multivariable character which is demanded to be dynamic with different response rates [3]. The electrical system is designed to operate at a specified value. Changes in the system will be allowed with tolerance limits [4].

A constant output voltage on a generator is very important to produce the expected power supply. Changes in the output voltage of a generator are influenced by various kinds of interference factors, one of which is a change in load. Therefore, we need a special regulator equipment to keep the generator output voltage constant even when the generator is affected by these interference factors. In addition, with the aim of maintaining the stability of the system, this regulator must be able to regulate the production or absorption of reactive power from the network at every change in load. This voltage regulator can be controlled both manually and automatically. In large-scale interconnection systems, manual regulators are never used and instead an automatic regulator device called an Automatic Voltage Regulator (AVR) is installed in each generator. AVR is set under certain operating conditions. AVR is useful for maintaining the frequency and magnitude of the voltage within its limits. Small changes depend on changes in rotor angle and speed [5].

Many researchers are interested in the field of power system stability. Several researchers have updated the method to improve AVR performance such as Particle Swarm Optimization (PSO) [6-8], salp optimization algorithm [9,10], Teaching-Learning-Based Optimization (TLBO) [11,12], Cuckoo Search Algorithm [13,14], Sine-cosine algorithm [15,16], and neural network [17-19].

The particle swarm optimization (PSO) is principally based on 2 main component methodologies. This is generally associated with artificial life. In particular, it relates to flocks of birds, schools of fish, and swarm theory. Miavagh *et al.* [6] presented the PSO method which is applied to 4 controllers, namely proportional integral derivative (PID), fractional proportional integral derivative (FPID), Tilt Integral Derivative (TID) and improved tilt-integral-derivative (I-TID). This method is applied to 4 error functions, namely Integral time absolute error (ITAE), Integral square error (ISE), Integral time square error (ITSE) and Integral absolute error (IAE) in the AVR system. Elumalai *et al.* [7] presented PID tuning using the PSO method to set various synchronous generator parameters. The results obtained from the proposed controller were compared with those estimated by the genetic algorithm. Li *et al.* [8] conducted a research on fractional order PID controller. Particle swarm optimization (PSO) is applied to find the best solution from the IAE criteria, namely the FOPID controller parameter.

The Salp Swarm Algorithm (SSA) is an algorithm that emulates the navigating and foraging behavior of salp fish in the oceans. Khan *et al.* [9] and Sirsode *et al.* [10] conducted research on the Automatic Voltage Regulator (AVR) using SSA to select the best FOPID controller parameters to achieve optimal dynamic response and increased stability of the AVR system.

The teaching-learning-based optimization (TLBO) method is duplicated from the teaching method that is influenced by a teacher on the learning outcomes of students in the classroom. The paper based on the TLBO method applied to AVR as conducted by Chatterjee *et al.* [11]. The proposed TLBO algorithm is applied to get the best value for the proportional integral derivative (PID) controller gain with the 1<sup>st</sup> order low pass filter on the AVR. Ayyıldız *et al.* [12] conducted a TLBO-based AVR research which was applied to the PID and FOPID methods. In this study, an objective function consisting of overshoot, rise time, settlement time, and steady-state error is formulated to achieve a satisfactory trade-off between dynamic response characteristics.

Cuckoo search optimization (CSO) is a method based on the cuckoo bird. This is a pretty bird that has a lovely tone. On the other hand, they have active reproductive strategies. The CSO-based AVR study to obtain FOPID controller was carried out by Sikander *et al.* [13], in this study the characteristics of the AVR response time with and without a controller were investigated to determine the effectiveness of the proposed controller. Bingul [14] conducted a CSO study to obtain optimal PID control. In the research conducted by Bingul [14], the proposed method was compared with the Particle Swarm Optimization (PSO) Algorithm and Artificial Bee Colony (ABC) for the PID design of the AVR system.

Sine Cosine Algorithm (SCA) is a population-based optimization technique. This technique begins with a random number search agent. Bhookya *et al.* [15] applied the SCA method to optimize Fractional Order PID (FOPID) control on the AVR. The SCA technique is used to optimize the 5 parameters of the FOPID controller based on the minimum value of the objective function of the controller design. Hekimoğlu *et al.* [16] applied the SCA method to determine the PID controller parameters in the AVR system. Research conducted by Hekimoğlu *et al.* [16] conducted comparisons with PID controllers based on the Ziegler-Nichols (ZN) adjustment method, differential evolution (DE), artificial bee colony (ABC) and bio-geography-based optimization (BBO).

The application of neural network methods to AVR control also varies in types. Bhutto [18] has used Probabilistic Neural Network (PNN) as AVR control. Aribowo [19] has applied a Focused Time Delay Neural Network (FTDNN) which functions as an AVR control.

The article will be an investigation to perform automatic voltage regulator tuning using an improved neural network. The method to improve neural network is slime mould algorithm (SMA). The SMA is presented by Li *et al.* [20] in 2020. SMA is based on the oscillation mode of slime mould in nature. Artificial neural network used in this study is focused time delay neural network (FTDNN). The generator is modeled using heffron-phillips. This article is focuses on rotor speed and angle. s. The contribution of this paper is

1. The implementation of the latest and promising metaheuristic methods namely Slime mold algorithm to unite with FTDNN. The Slime mould algorithm method is used to improve the ability of the neural network through weighting.

2. To find out the performance of the proposed method will be validated. Validation is performed by comparing with the methods of feed-forward backpropagation neural networks (FFBNN), cascade-forward backpropagation neural networks (CFBNN), Elman-recurrent neural networks (E-RNN), FTDNN (19), and Distributed Time Delay Neural Network (DTDNN).

The paper is organized as follows: The 2<sup>nd</sup> part provides a complete study of a brief description of the metaheuristic methods used in this paper, focused time delay neural network and automatic voltage regulators. Section 3 presents the results and performance comparison analysis. In the last section, a conclusion is drawn.

## Materials and methods

### Slime mould algorithm

Slime Mould Algorithm (SMA) method is an optimization algorithm that duplicates the changing habits and morphology of the slime mould *Physarum polycephalum* in finding food. Slime mould are eukaryotes that occupy cold and damp areas. The main stage of the slime mould namely plasmodium, which is the stage of organic matter in the slime mould finding for food, twining it, and detaching enzymes to eat it. Slime mould has unique patterns that can grow more than 900 square centimeters if there is an excess of food resources in the surrounding environment. Slime mould has the ability to optimize for finding food. Slime mould can choose the most food sources. On the other hand, slime mould also takes into account speed, accuracy and risk in finding food [20].

Slime mould with inadequate information must decide when the right time to move to other areas when looking for food. When slime mould occupy areas of high food supply, the pressure to move decreases. Slime mould has unique characteristics that can utilize a variety of food sources simultaneously. The adaptability of slime molds can adjust search patterns based on food quality. Slime molds will search in restricted areas when food supplies are plentiful. However, the slime mould will enlarge the search area when the food supply in the area is little. The slime mold algorithm generally consists of 3 parts:

### Phase 1: Approach food

Slime mould searches for and recognizes food from odors through the air. This method can be modeled as follows:

$$\overrightarrow{P}(t+1) = \begin{cases} \overrightarrow{P}_b(t) + \overrightarrow{vl} \cdot (\overrightarrow{W} \cdot \overrightarrow{P}_A(t) - \overrightarrow{P}_B(t)), r < p \\ \overrightarrow{vd} \cdot \overrightarrow{P}(t), r \geq p \end{cases} \quad (1)$$

$$p = \tanh|H(i) - BF| \quad i \in 1, 2, \dots, n \quad (2)$$

$$\overrightarrow{vl} = [-a, a] \quad (3)$$

$$a = \operatorname{arctanh}\left(-\left(\frac{t}{\max\_t}\right) + 1\right) \quad (4)$$

Where the unique spot with the best smell concentration currently established is  $\overrightarrow{P}_b$ .  $\overrightarrow{vl}$  is measurable variable with a limit  $[-a, a]$ , linearly decreased variables ranging from 1 to 0 is  $\overrightarrow{vd}$ .  $t$  is the current iteration, the position of the slime mould is  $\overrightarrow{P}$ , 2 values chosen randomly are  $\overrightarrow{P}_A$  and  $\overrightarrow{P}_B$ ,  $\overrightarrow{W}$  is the weight of slime mould.  $H(i)$  is the fitness of  $\overrightarrow{P}$ ,  $BF$  is the best fitness obtained in all iterations.

The formula of  $\overrightarrow{W}$  is listed as follows:

$$\overrightarrow{W}(\operatorname{SmellIndex}(i)) = \begin{cases} 1 + r \cdot \log\left(\frac{bF-H(i)}{bF-wF} + 1\right), \text{condition} \\ 1 - r \cdot \log\left(\frac{bF-H(i)}{bF-wF} + 1\right), \text{others} \end{cases} \quad (5)$$

$$\operatorname{SmellIndex} = \operatorname{sort}(S) \quad (6)$$

Where condition parameter is indicating that  $H(i)$  occupies the top of the population.  $R$  is a random value between 0 and 1.  $bF$  is the best fitness measure in the current iteration process.  $wF$  is the lowest fitness value in the currently iteration process.  $\operatorname{SmellIndex}$  is the series of fitness parameters sorted.

### Phase 2: Wrap food

The phase is to duplicate the contraction mode of the slime mold structure mathematically when looking for food. The more food supply that is touched by the slime mold network. The signal generated by the bio-oscillator is getting stronger, the faster the cytoplasm flows and the thicker the net.

$$s\vec{P}^* = \begin{cases} rand \cdot (ub - lb) + lb, rand < z \\ \vec{P}_b(t) + \vec{vl} \cdot (W \cdot \vec{P}_A(t) - \vec{P}_B(t)), r < p \\ \vec{vd} \cdot \vec{P}(t), r \geq p \end{cases} \quad (7)$$

Where  $lb$  and  $ub$  is the lower and upper limits of the seeking,  $rand$  and  $r$  is the random value with range 0 to 1. The value  $z$  is an experiment to parameter settings.

### Phase 3: Grabble food

Slime mould is very dependent on the information signal sent by the biological oscillator. This will make the slime mould in the best position. For the duplication of variant widths of the slime mold parameters are used  $\vec{W}$ ,  $\vec{vl}$ , and  $\vec{vd}$ .

$\vec{W}$  is representing a duplication of the slime mould oscillation frequency when at different food concentrations. This makes slime moulds reach food faster. On the other hand, when the concentration of food is low from the position of the slime mold. Slime mould will slow down. So that the movement of the slime mould is more efficient. The selective character of the slime mold is duplicated in  $\vec{vl}$  and  $\vec{vd}$  to find the latest food sources. If the slime mould finds the latest food source. Slime mould still have organics that can be used to explore the latest and highest quality food sources.

$\vec{vl}$  process that duplicates the behavior of the mold slime to provide information whether to approach the food or find other food sources. The period will be influenced by environmental factors. On the other hand, this will increase the impulse of the slime mould to find high-quality food and avoid local traps.

### Focused time delay neural networks

Focused Time Delay Neural Networks FTDNN(FTDNN) are dynamic ANN types. FTDNN has a topology consisting of forward feed which has a time delay. This is placed on the network through which the input is passed. FTDNN was developed primarily to process temporary patterns [19]. Time delay placed on topology will support in predicting and controlling problems effectively and efficiently in dynamic ANN, the results have several variables that can be used to maintain the contextual part of the input unit. The part is available in local memory in the form of tapped time delay. The basic FTDNN topology consists of memory structures and non-linear associations [21]. The Focused Time Delay Neural Networks architecture can be seen in **Figure 1**.

The memory structure functions are to store records of related past information and the associator functions is to predict future events with the help of memory structures. Memory structure contains a time delay signal that originates from the input. Whereas, the associator contains a conventional network of feed-forward. The main structure of FTDNN is to have a special memory structure that is in the input layer. The advantage of FTDNN is having a simple topology. Where  $L$  is the length of the delay path memory, Unit input is  $I_n(t)$ , the network processes is  $I_n(t)$ ,  $I_n(t-1)$ ,  $I_n(t-2)$  ... and  $I_n(t-L)$ , Therefore, the input signal  $X_1(t)$  to neuron  $i$  (**Figure 1**) is given as:

$$X_1(t) = \sum_{i=1}^j W_{ij} I_n(t-L) + b_1 \quad (8)$$

Output  $X_2(t)$  is the result of processing  $X_1(t)$  using a non-linear activation function. A function that is often used is the sigmoid activation function.

$$X_2(t) = f(X_1(t)) = \frac{1}{1+exp^{X_1}} \quad (9)$$

At layer 2 there is no time delay. Output  $X_2(t)$  is becomes input at layer 2. It is becomes output  $X_3(t)$ . The results are then activated using the activation function.

$$X_3(t) = \sum_{j=1}^k W_{jk} X_2(t) + b_2 \quad (10)$$

$$X_4(t) = f(X_3(t)) = \frac{1}{1+exp^{X_3}} \quad (11)$$

One of FTDNN's features is a topology that hasn't backpropagation to calculate network gradients. The tapped time delay is only in the input layer and does not contain a feedback loop.

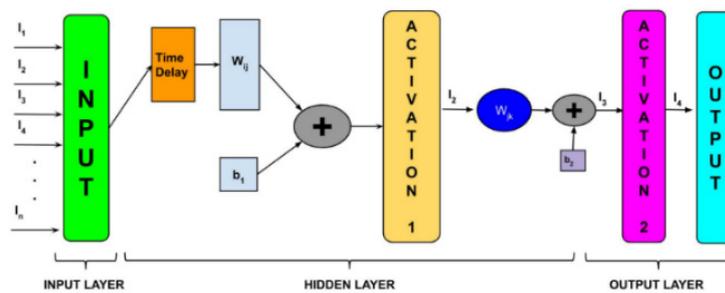


Figure 1 FTDNN topology.

**Automatic voltage regulator**

The generator control consists of 2 parts, namely Automatic voltage regulators (AVR) and power system stabilizers (PSS). AVR and PSS have functions to maintain generator stability [22]. AVR functions to regulate the terminal voltage at a predetermined value. Simple AVR is composed of 4 components, namely amplifier, exciter, generator and sensor. The Schematic of AVR can be seen **Figure 2**. For mathematical modeling, components are assumed to be linear. This takes into account the main time constant and ignores nonlinearity. The control of the generator is composed of the AVR and PSS. An automatic voltage regulator (AVR) and power system stabilizer (PSS) are used to repair the transient stability.

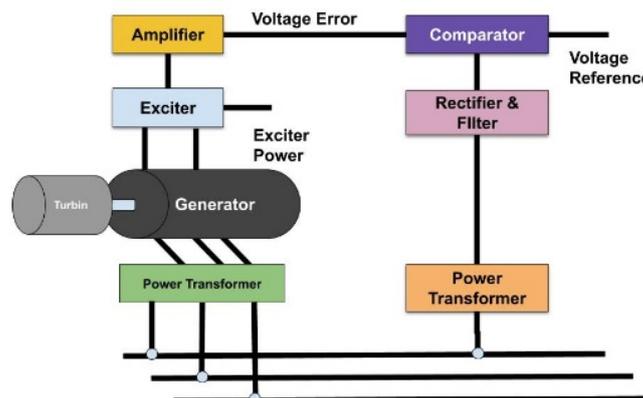


Figure 2 The schematic of AVR.

**The proposed SMA-FTDNN model**

The application of SMA and FTDNN in the tuning of AVR can be illustrated as shown in **Figure 3**. The 1<sup>st</sup> step is to model the generator in a single machine type heffron-phillips. The simulation results of the modeling are data speed and rotor angle. This data is used as input in a neural network. Random initial weight data from FTDNN is sampled. The results are processed using SMA which has 3 phases. It is a food approach, wrapping food and grabbed food. SMA results will be a potential weight for FTDNN that can be used in the network.

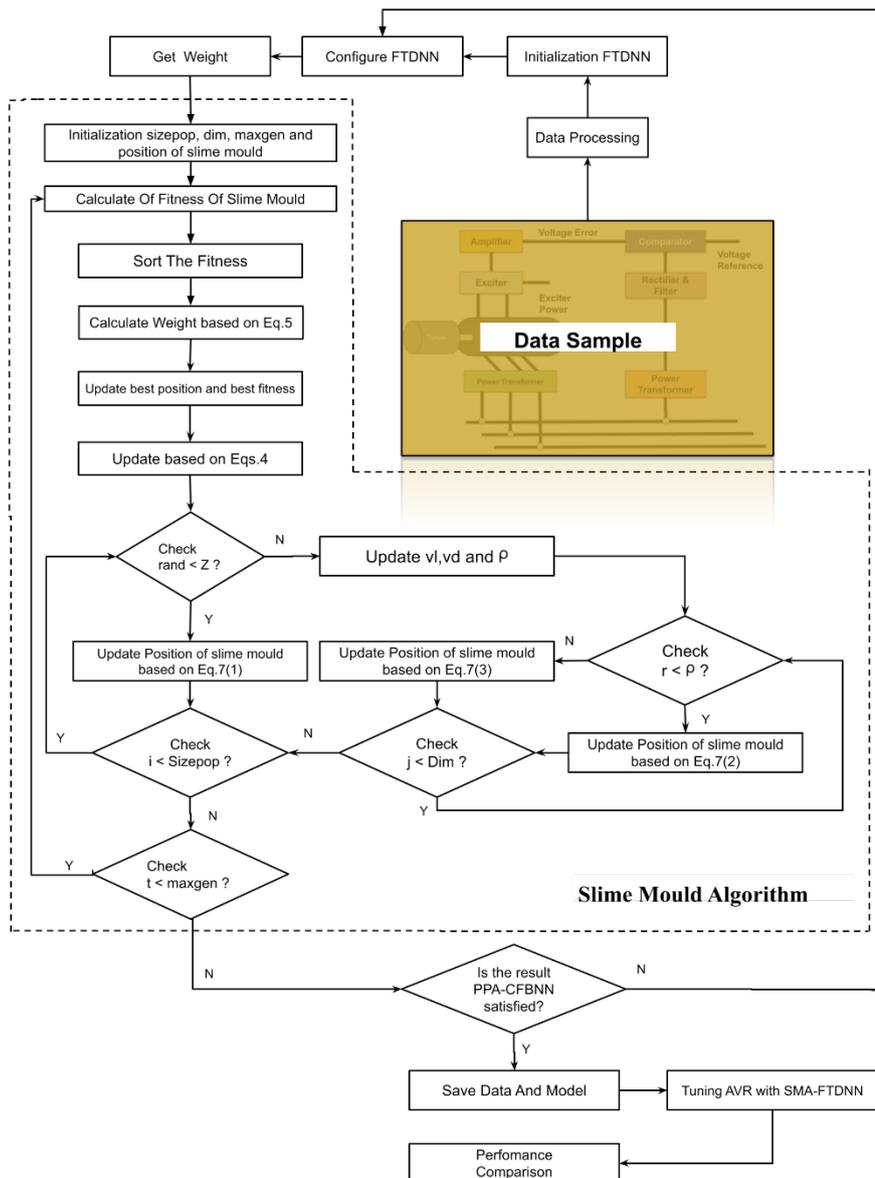
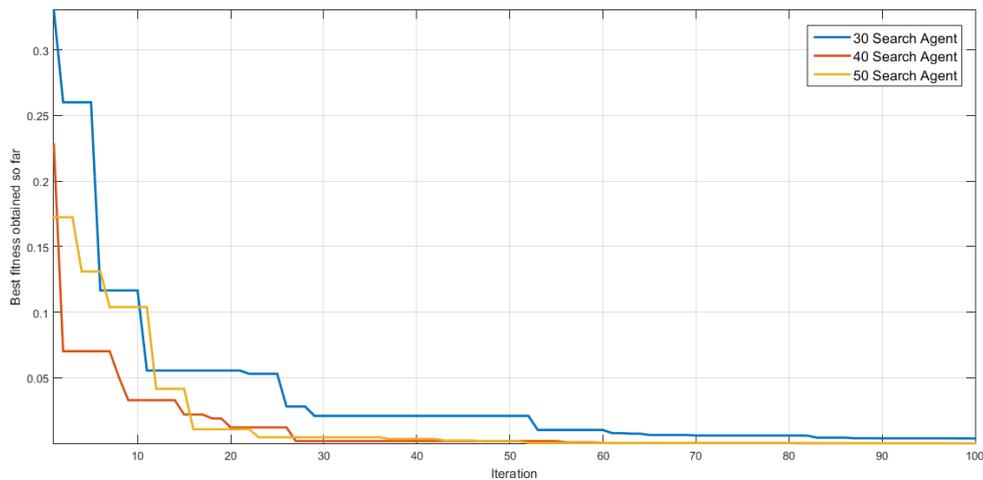


Figure 3 The SMA-FTDNN flowchart.

**Results and discussion**

The proposed method was applied to optimize conventional controller in order to see the enhancement stability of the power system. The application for the SMA-FTDNN was written in MATLAB. The dynamic and transient condition was compared with the operation of the other method controller under 3 different loading conditions: 25, 50 and 90 %. The validation of the proposed method will be carried out by comparing it with the FFBNN, CFBNN, Elman-RNN, FTDNN and DTDNN methods. Each of these was investigated for speed and rotor angle.



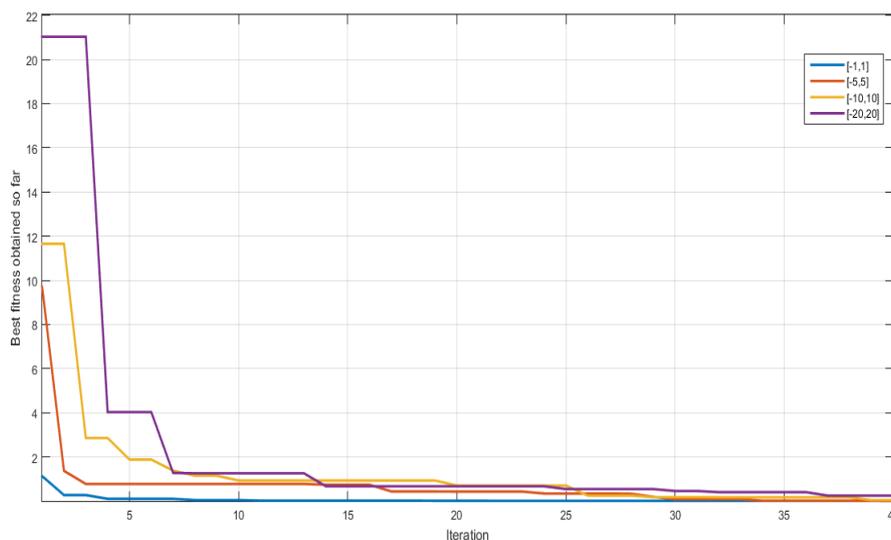
**Figure 4** Convergence curve of search agent for SMA-FTDNN.

The variables of SMA is also required to be set up before optimization of the FTDNN model, including the number of Slime Mould (Search Agent), maximum number of iterations and lower-upper limit of the optimization (L). **Figure 4** is the performance of search agent with 100 iterations. The use of the highest search agent gets the best score for fitness. The peak value is 0.2088. Meanwhile, the settling time value is 15.5781. The detailed results can be seen in **Table 1**.

**Table 1** Parameter values for various search agent.

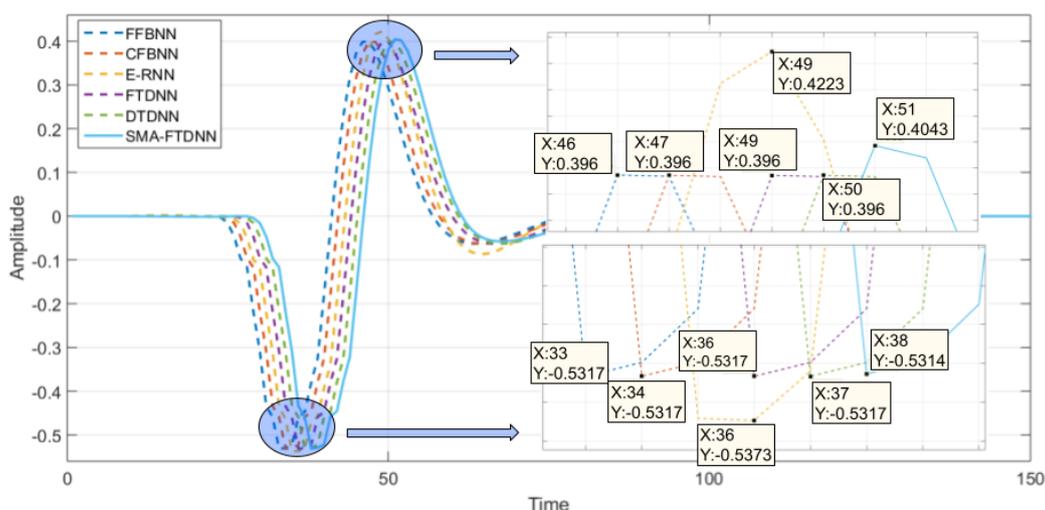
Search agent	Rise time	Settling time	Peak
30	24.204	52.998	0.3301
40	13.778	26.723	0.2280
50	7.0800	15.5781	0.2088

The lower-upper parameter also needs to be set to get a value that can be used for optimization. The test uses a random range value, namely the range  $[-1, 1]$ ,  $[-5, 5]$ ,  $[-10, 10]$  and  $[-20, 20]$ . **Figure 5** shows the convergence curve value for each lower-upper using the SMA-FTDNN method. In **Figure 5**, it can be seen that the lower-upper value quickly converges when using the value  $[-1, 1]$ . Testing of slime mold and lower-upper parameters uses 100 iterations.

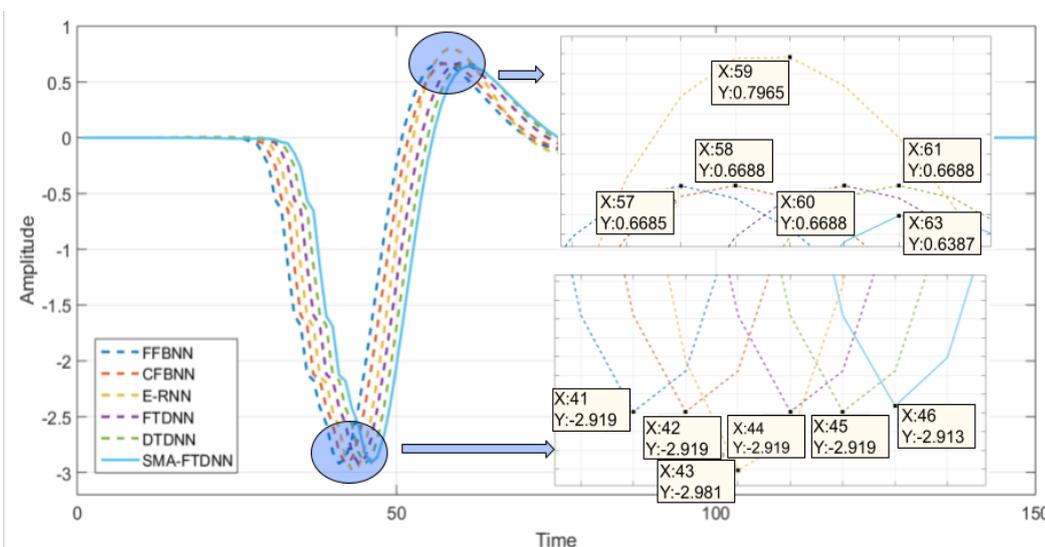


**Figure 5** Convergence curve of lower-upper.

Measurement of the performance of the generator is the speed and rotor angle. The 1<sup>st</sup> test is carried out by overloading the generator with a loading of 25 %. The results of the loading can be seen in **Figures 6** and **Figure 7**. From **Figure 6**, the undershoot values of the FFBNN, CFBNN, FTDNN and SMA-FTDNN methods are obtained the same value. The value is  $-0.5317$ . This value is  $0.056\%$  worse than the proposed method. The worst score for undershoot belongs to the E-RNN method. The value is  $-0.5373$ . Meanwhile, the overshoot value of the FFBNN, CFBNN, FTDNN and DTDNN methods has the same value, namely  $0.396$ . The worst score belongs to the E-RNN method. The value is  $0.4223$ . At  $25\%$  loading, the proposed method has a speed overshoot value of  $0.4043$ . This score is  $2.05\%$  worse than the FFBNN, CFBNN, FTDNN and DTDNN methods. On the other hand, it was  $4.26\%$  better than the E-RNN method.



**Figure 6** Speed response with 25 % loads.



**Figure 7** Rotor angle response with 25 % loads.

**Figure 7** is the result of the rotor angle with  $25\%$  of loading. In **Figure 7**, the worst values for the overshoot and undershoot are  $0.7965$  and  $-2.981$ . This value belongs to the E-RNN method. Meanwhile, the undershoot value of the FFBNN, CFBNN, FTDNN and DTDNN methods obtained the same value that is  $-2,919$ . This value is  $0.2056\%$  worse than the proposed method. **Table 2** can be seen the undershoot, overshoot and settling time of the rotor angle with  $25\%$  of loading. The value of overshoot rotor angle from the CFBNN, FTDNN and DTDNN methods has the same value, namely  $0.6688$ . Meanwhile, FFBNN's rotor angle overshoot value is slightly better, namely  $0.6685$ . This value was  $4.46\%$  worse than the proposed method.

**Table 2** Result of training with 25 % of load.

Methods	Speed response				Rotor angle response			
	Under shoot	Over shoot	Rise time (s)	Settling time (s)	Under shoot	Over shoot	Rise time (s)	Settling time (s)
FFBNN	-0.5317	0.396	0.453	86.7831	-2.919	0.6685	2.8175e-07	84.4279
CFBNN	-0.5317	0.396	0.5475	86.8497	-2.919	0.6688	2.9123e-07	84.4537
E-RNN	-0.5373	0.4223	0.6947	86.7443	-2.981	0.7965	2.9123e-07	84.4537
FTDNN	-0.5317	0.396	0.4440	86.8441	-2.919	0.6688	2.9030e-07	84.4514
DTDNN	-0.5317	0.396	0.4440	86.8441	-2.919	0.6688	2.8957e-07	84.4508
SMA-FTDNN	-0.5314	0.4043	1.2288e-10	86.9444	-2.913	0.6387	0.7659	84.3022

Experiments with 50 % of loading, the results of the SMA-FTDNN method obtained the best overshoot and undershoot values of the rotor angle. The values are 0.7051 and -3.1524. The value of the overshoot rotor angle of the proposed method is 1.54 % better than the FFBNN method. Meanwhile, the undershoot rotor angle value was 0.53 % better than the CFBNN and FTDNN methods. On the other hand, the overshoot value of the speed of the proposed method has a value of 0.4138. This value is slightly better 3.63 % than the CFBNN, FTDNN and DTDNN methods. Meanwhile, the undershoot value is 0.4028 % better than the CFBNN, FTDNN, and DTDNN methods. The details of the results of the 50 % loading can be seen in **Table 3**.

**Table 3** Result of training with 50 % of load.

Methods	Speed response				Rotor angle response			
	Under shoot	Over shoot	Rise time (s)	Settling time (s)	Under shoot	Over shoot	Rise time (s)	Settling time (s)
FFBNN	-0.5711	0.4296	0.4819	88.1445	-3.1694	0.7161	2.6061e-07	85.6717
CFBNN	-0.5710	0.4294	0.5937	88.1705	-3.1692	0.7165	2.7709e-07	85.6926
E-RNN	-0.5795	0.4555	0.6714	88.0012	-3.2385	0.8592	2.4764e-06	83.8869
FTDNN	-0.5710	0.4294	0.3998	88.1720	-3.1692	0.7164	2.7593e-07	85.6923
DTDNN	-0.5710	0.4294	0.6879	88.1769	-3.1693	0.7164	2.7710e-07	85.6948
SMA-FTDNN	-0.5687	0.4138	6.7637e-10	91.2437	-3.1524	0.7051	0.1695	87.3616

With a system loading of 90 %, the output value of overshoot and undershoot of the speed and rotor angle respond for the SMA-FTDNN method is the highest. The overshoot value of the rotor angle of the proposed method is slightly better at 0.28 % than the FFBNN method. Meanwhile, the value of the overshoot rotor angle of the proposed method is better at 0.336 % than the FTDNN and DTDNN methods. In the undershoot of the rotor angle, the value of the proposed method is slightly better at 1.55 % than the FFBNN, CFBNN, FTDNN and DTDNN methods. The worst score is owned by the E-RNN method. The value is -3,553. On the overshoot of the speed, the value of the proposed method is 0.4753. this value is the best value. This value is 0.813 % better than the CFBNN, FTDNN and DTDNN methods. The worst score belongs to the E-RNN method. On the other hand, the undershoot value of the speed of the proposed method is better at 2.295 % than the FFBNN, CFBNN, FTDNN and DTDNN methods. Detailed results of 90 % loading can be seen in **Table 4**.

**Table 4** Result of training with 90 % of load.

Methods	Speed response				Rotor angle response			
	Under shoot	Over shoot	Rise time (s)	Settling time (s)	Under shoot	Over shoot	Rise time (s)	Settling time (s)
FFBNN	-0.6319	0.4793	0.6236	88.5575	-3.4696	0.7743	3.3218e-07	86.0045
CFBNN	-0.6319	0.4792	0.6678	88.6514	-3.4696	0.7748	2.8889e-07	86.0095
E-RNN	-0.6425	0.5084	0.6283	88.4080	-3.5531	0.9341	3.0519e-06	84.1942
FTDNN	-0.6319	0.4792	0.3231	88.6601	-3.4696	0.7747	2.9086e-07	86.0123
DTDNN	-0.6319	0.4792	0.6527	88.6592	-3.4696	0.7747	2.8851e-07	86.0105
SMA-FTDNN	-0.6174	0.4753	0.8365	96.7849	-3.4159	0.7721	2.7602e-08	91.0936

## Conclusions

This article discusses the development of the latest hybrid methods based on AI techniques for controlling AVR. The comparison and evaluation of the methods developed with other methods are carried out in this article. It can be concluded that the metaheuristic method has a good performance in optimizing ANN. By using the new method, the slime mold algorithm to optimize FTDNN, the best value is obtained in AVR control. From the research, it was found that the average value of the undershoot and overshoot of the speed of the proposed method was able to reduce by 0.92 and 0.78 % compared to the standard FTDNN method. Meanwhile, for the rotor angle, the proposed method is able to reduce the average undershoot and overshoot values by 0.76 and 2.134 % compared to the standard FTDNN method.

## Acknowledgements

The author would like to thank the Universitas Negeri Surabaya.

## References

- [1] W Aribowo, S Muslim, Munoto, B Suprianto, UT Kartini and IGPA Buditjahjanto. Tuning of power system stabilizer using cascade forward backpropagation. *In: Proceedings of the 3<sup>rd</sup> International Conference on Vocational Education and Electrical Engineering (ICVEE)*, Surabaya, Indonesia. 2020, p. 1-5.
- [2] X Zhao, X Zhang and B He. Study on self-organized criticality of China power grid blackouts. *Energy Convers. Manage.* 2009; **50**, 658-61.
- [3] SK Nallagalva, MK Kirar and G Agnihotri. Transient stability analysis of the IEEE 9-bus electric power system. *Int. J. Eng. Sci. Technol.* 2012; **1**, 161-66.
- [4] W Aribowo, S Muslim, B Suprianto, S Haryudo and AC Hermawan. Intelligent control of power system stabilizer based on Archimedes optimization algorithm-feed forward neural network. *Int. J. Intell. Eng. Syst.* 2021; **14**, 43-53.
- [5] H Saadat. *Power system analysis*. McGraw-Hill, New York, 1999.
- [6] FM Miavagh, EAA Miavaghi, AR Ghiasi and M Asadollahi. Applying of PID, FPID, TID and ITID controllers on AVR system using Particle Swarm Optimization (PSO). *In: Proceedings of the 2<sup>nd</sup> International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran. 2015, p. 866-71.
- [7] K Elumalai and S Sumathi. Behavior modification of PID controller for AVR system using particle swarm optimization. *In: Proceedings of the Conference on Emerging Devices and Smart Systems (ICEDSS)*, Mallasamudram, India. 2017, p. 190-5.
- [8] X Li, Y Wang and N Li. Optimal fractional order PID controller design for automatic voltage regulator system based on reference model using particle swarm optimization. *Int. J. Mach. Learn. Cyber.* 2017; **8**, 1595-605.
- [9] IA Khan, AS Alghamdi, TA Jumani, A Alamgir, AB 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* 2019; **8**, 1472.

- [10] 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: Proceedings of the 6<sup>th</sup> Indian Control Conference (ICC), Hyderabad, India. 2019, p. 431-6.*
- [11] S Chatterjee and V Mukherjee. PID controller for automatic voltage regulator using teaching-learning based optimization technique. *Int. J. Electr. Power Energy Syst.* 2016; **77**, 418-29.
- [12] BA Ayyıldız and O Karahan. Controller tuning approach with TLBO algorithm for the automatic voltage regulator system. *Eskişehir Tech. Univ. J. Sci. Tech. Appl. Sci. Eng.* 2020; **21**, 128-46.
- [13] A Sikander, P Thakur, RC Bansal and S Rajasekar. A novel technique to design cuckoo search based FOPID controller for AVR in power systems. *Comput. Elect. Eng.* 2018; **70**, 261-74.
- [14] B Zafer and O Karahan. A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. *J. Frankl. Inst.* 2018; **355**, 5534-59.
- [15] J Bhookya and RK Jatoth. Optimal FOPID/PID controller parameters tuning for the AVR system based on sine-cosine-algorithm. *Evol. Intell.* 2019; **12**, 725-33.
- [16] B Hekimoğlu. Sine-cosine algorithm-based optimization for automatic voltage regulator system. *T. I. Meas. Control.* 2019; **41**, 1761-71.
- [17] G Bal, O Kaplan and SS Yalcin. Artificial neural network based automatic voltage regulator for a stand-alone synchronous generator. *In: Proceedings of the 8<sup>th</sup> International Conference on Renewable Energy Research and Applications (ICRERA), Brasov, Romania. 2019, p. 1032-37.*
- [18] AA Bhutto, FA Chachar, M Hussain, DK Bhutto and SE Bakhsh. Implementation of probabilistic neural network (PNN) based automatic voltage regulator (AVR) for excitation control system in MATLAB. *In: Proceedings of the 2<sup>nd</sup> International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sindh, Pakistan. 2019, p. 1-5.*
- [19] W Aribowo. Focused time delay neural network for tuning automatic voltage regulator. *Emitter Int. J. Eng. Technol.* 2019; **7**, 34-43.
- [20] S Li, H Chen, M Wang, AA Heidari and S Mirjalili. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* 2020; **111**, 300-23.
- [21] M Abed, A El-Shafie and S Siti. Creep predicting model in masonry structure utilizing dynamic neural network. *J. Comput. Sci.* 2010; **6**, 597-605.
- [22] S Essallah, A Bouallegue and A Khedher. Integration of automatic voltage regulator and power system stabilizer: Small-signal stability in DFIG-based wind farms. *J. Mod. Power Syst. Clean Energ.* 2019; **7**, 1115-28.