SINERGI Vol. 22, No.3, October 2018:205-210
DOAJ:doaj.org/toc/2460-1217
DOI:doi.org/10.22441/sinergi.2018.3.009 .

TUNING FOR POWER SYSTEM STABILIZER USING DISTRIBUTED TIME-DELAY NEURAL NETWORK

Widi Aribowo
Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya
Jl. Unesa Kampus Ketintang, Surabaya 60231
Email: widiaribowo@unesa.ac.id

Abstract *-- In this paper, a Distributed Time-Delay Neural Network (DTDNN) algorithm is used to control the Power System Stabilizer (PSS) parameters to find the reliable conditions. The proposed DTDNN algorithm apply tapped delay line memory to set the PSS. In this study, DTDNN consists of a DTDNN-identifier and a DTDNN-controller. The performance of the system with DTDNN-PSS controller is compared with a Recurrent Neural Network PSS (RNN-PSS) and Conventional PSS (C-PSS). The results show the effectiveness of DTDNN-PSS design, and superior robust performance for enhancement power system stability compared to other with different cases.*

Keywords: *Power System Stabilizer (PSS); DTDNN; Recurrent Neural Network; Single machine.*

Received: April 18, 2018 *Revised: May 15, 2018* *Accepted: May 16, 2018*

INTRODUCTION

Power systems are complex multi-component dynamic systems in which the system characteristics fluctuate with varying loads and varying generation schedules. These power systems suffer by low-frequency oscillations on sudden changes in load or occurrence of fault. The transfer of bulk power across weak transmission lines is hindered due to continuous persistence of such a low-frequency oscillation (0.2–3.0 Hz) (Muammar et al., 2018; Rahayu, Sambariya & Prasad, 2015).

In early 1960s, the fast acting, high-gain automatic voltage regulators (AVRs) were applied to the generator excitation system which in-turn invites the problem of low frequency electromechanical oscillations in the power system. To reduce the low-frequency oscillations, the PSS adds a stabilizing signal to AVR that modulates the generator excitation to damping electrical torque component in phase with rotor speed deviation, which increases the generator damping.

The uniformly adopted type of PSS is known as conventional PSS (CPSS), which consists with the lead–lag-type components. In early phase of optimization, CPSS parameters have been tuned using gradient based optimization technique. It requires the computation of sensitivity and eigenvectors at the end iteration, which resulted with heavy computational burden and slow convergence rate (Sambariya, Gupta & Prasad, 2016). The CPSS suffers with some limitations as (a) these are designed off-line, therefore, requires re-tuning during commissioning, (b) these are tuned for one operating condition, therefore, may not perform properly for varying operating conditions, and (c) because of changing conditions and configuration of a power system, they require retuning for good performance, regularly (Sambariya & Prasad, 2015).

CPSS are the fixed parameter controllers, designed over a nominal operating point to get desired performance at this point as well expect over a wide range of operating conditions and varying system conditions (Sambariya ,2015)..

In case of adaptive PSS, with poor initialization, the performance during learning phase is not satisfactory. Continuity of the objective function is a prerequisite for gradient algorithm used in such applications. As alternative control theories such as the variable structure, the adaptive and linear optimal control theory has been used to design Power System Stabilizers with improved performance (Bhati & Gupta, 2013).

The rapid development of power systems and increasingly diverse problems, demands a real-time settlement. Therefore, this paper proposes the design of power system stabilizers (PSS) based on Distributed Time-Delay Neural Network (DTDNN) that can respond to changes in system performance directly. In this study DTDNN PSS was applied on a single machine system. The application of DTDNN PSS on a single machine system is emphasized on DTDNN PSS performance against low frequency oscillations and improved performance of system.

METHOD

Artificial neural networks are modeled after the computational principles of brain, with the

specific aim of understanding and replicating human activities (Gruning & Sander, 2014).

Artificial neural networks have been used in many applications such as sales forecasting, industrial process control, customer research, medical application and risk management (Ozerdm, Olaniyi, & Oyedotun, 2017).

Artificial Neural Networks are nonlinear models motivated by the physiological architecture of the nervous system which can be trained to learn approximate functions of complex non-linear systems that usually depend on a large number of inputs (Hagan, Demuth, Beale & Jesu´s,2014).

Artificial neural networks are typically organized in layers: the input layer, hidden layer and output layer, which consist of several neurons. Each neuron in a layer is connected to adjacent layers by weights. Data are presented to the network via the input layer, which are multiplied by the weights, and then go through the activation function of the neuron in one or more hidden layers. The function in the output layer computes the output of the artificial neuron (Baliyan, Gaurav & Mishra, 2015).

Back-propagation (BP) algorithm is the most frequently used, effective, and easy way to learn model for multilayered networks. It is a supervised learning technique which is based on the gradient descent method that attempts to minimize the error of the network by moving down the gradient of the error curve. Levenberg-Marquardt algorithm is one of the fastest back-propagation algorithms which works well for training small and medium sized networks.

But static (feed forward) networks have no feedback elements and contain no delays; the output is calculated directly from the input through feed forward connections.

The weakness of the backpropagation neural network is the limited function of static training and output depends only on current input conditions, so it can not afford. If there is any change of input data pattern.

According to Ibrahim, (2010), DTDNN provides a simple and efficient way of classifying data sets. To process data for classification we believe that DTDNN are best suited due to their high speed and fast conversion rates as compared with other learning techniques. Also, DTDNN preserves topological mappings between representations, a feature which is desired when classifying normal v.s. intruder behavior for network data.

That is, the relationships between senders, receivers and the protocols them, which are the primary features that we use, are preserved by the mapping. A DTDNN is it dynamic networks are generally more powerful than static networks because dynamic networks have memory, they can be trained to learn sequential or time varying patterns. In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network (Ibrahim,2010).

Each layer in the Distributed Time-Delay Artificial Neural Network is made up of the following parts:

- Set of weight matrices that come into that layer (which can connect from other layers or from external inputs), associated weight function rule used to combine the weight matrix with its
- input (normally standard matrix multiplication), and associated tapped delay line.
- Bias vector
- Net input function rule that is used to combine the outputs of the various weight functions with the bias to produce the net input (normally a summing junction)
- Transfer function

The network has inputs that are connected to special weights, called input weights, and denoted by IWij, where j denotes the number of the input vector that enters the weight, and i denotes the number of the layer to which the weight is connected. The weights connecting one layer to another are called layer weights and are denoted by LWij, where j denotes the number of the layer coming into the weight and i denotes the number of the layer at the output of the weight (Ibrahim, 2010).

A Distributed Time-Delay Artificial Neural Network structure is used to reduce overshoot of speed. The 703 features from SMIB datasets are used for input data. The DTDNN processes those given data to recognize and reduce overshoot of speed oscillation. Fig. 1 illustrates the architecture and parameters used in simulation process. The following are Proposed Distributed Time Delay Neural Network equations:

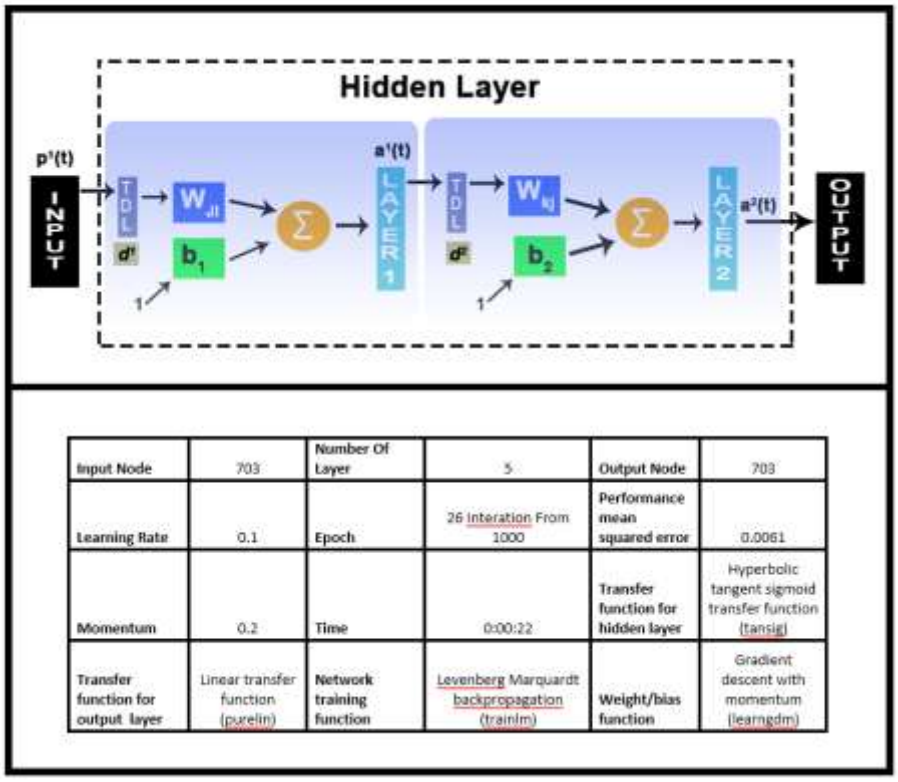
Layer 1

$$a^1(t) = \sum_{i=1}^{j} W_{ji} p^1(t - d^1) + b^1 \quad (1)$$

Layer 2

$$a^2(t) = \sum_{i=1}^{k} W_{kj} a^1(t - d^2) + b^2 \quad (2)$$

Symbol j and k are indicated j and k neuron. Where W_{ji} is the network weighted input. In layer 1, $p^1(t - d^1)$ inputs at the time $(t - d^1)$ $a^1(t - d^2)$. is the output from the hidden node; W_{kj} and $(t - d^2)$. are the weight and delay connecting in the layer 2. $a^2(t)$ is the output of the kth neuron in the lth layer at the $(t - d^2)$.



Input Node	703	Number Of Layer	5	Output Node	703
Learning Rate	0.1	Epoch	26 Interation From 1000	Performance mean squared error	0.0061
Momentum	0.2	Time	0:00:22	Transfer function for hidden layer	Hyperbolic tangent sigmoid transfer function (tansig)
Transfer function for output layer	Linear transfer function (purelin)	Network training function	Levenberg Marquardt backpropagation (trainlm)	Weight/bias function	Gradient descent with momentum (learngdm)

Figure 1. Proposed Distributed Time Delay Neural Network Structure

RESULTS AND DISCUSSION

In this paper, the power system under study composes of the single machine connected to an infinite bus (SMIB) through a transmission line. It is mentioned The Heffron-Phillips block diagram for the mathematic model of the SMIB power system as shown in Fig. 2.

Stability analysis requires the modelling of some important power system components such as excitation system, synchronous generator and AC network. The current design makes use of IEEE Model 1.0 to correspond to the synchronous generator having high gain and constant static exciter of low time.

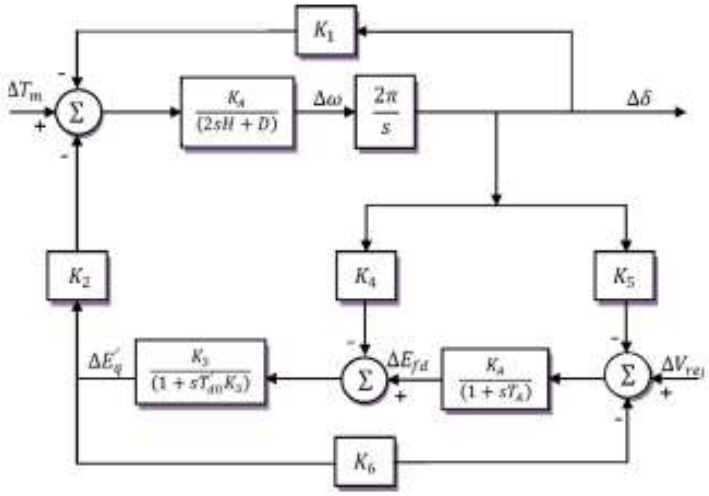


Figure 2. Heffron–Phillips block diagram for SMIB power system (Chaib L, Choucha. A. & Arif. A. 2017).

The SMIB criterion linear model, known as Heffron-Phillips model (also called K-constant model) can be achieved by linearizing the system equations around operating conditions. Conversion of the machine equations present in Park reference frame to the Kron reference frame rotating synchronously can establish an interface of synchronous machine with the external network.

The performance of system must know before analyze the system. The Flowchart is shown in Fig. 3. Data sample take from simulation without and with C-PSS. When Training and develop model had finished. The DTDNN-PSS has applied to SMIB to reduce speed overshoot.

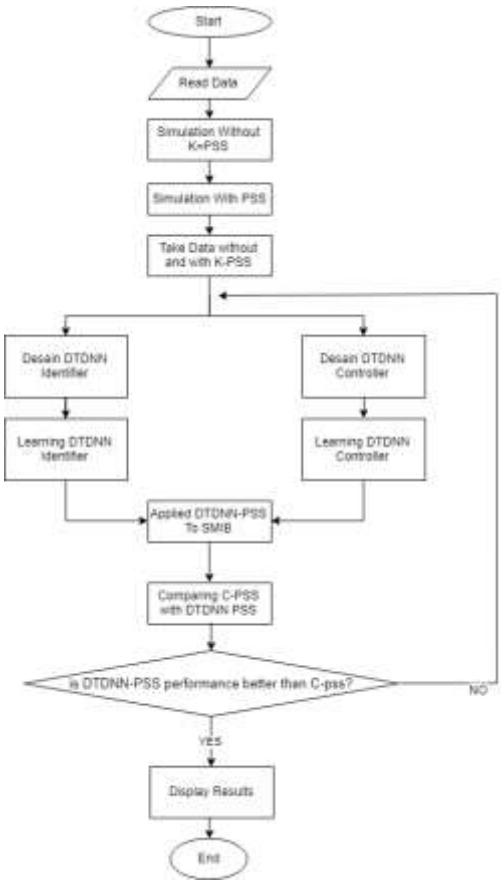


Figure 3. Flowchart Diagram of Design DTDNN-PSS

The DTDNN reads the output of the power system and attempts to duplicate the waveform of the output power system by comparing the output of the power system with the DTDNN output. If there is deviation error, the error signal is sent back to DTDNN for the learning process to minimize error.

DTDNN has two inputs Δω and Δu. Δω is the output of the plant and Δu is the output of PSS, as the initial input DTDNN uses the output from conventional PSS for the trainning process. Mathematically can be written:

$$Xi(t) = [\Delta\omega, \Delta u] \quad (3)$$

Where Δω taken from the rated value of the last synchronous machine which is censored with a constant time interval of 100 ms. And, Δu is is taken from the last control that has been done using conventional pss. (C-PSS simulation). T is the sampling period, ω is the deviation of the angular velocity against the sync speed in rad / s. u is the output of the controller. The network structure used in this training consists of three layers, namely the input layer, the hidden layer and the output layer. When the mapping process has done, DTDNN PSS is installed to the system. The model of DTDNN P S is shown in Fig. 4.

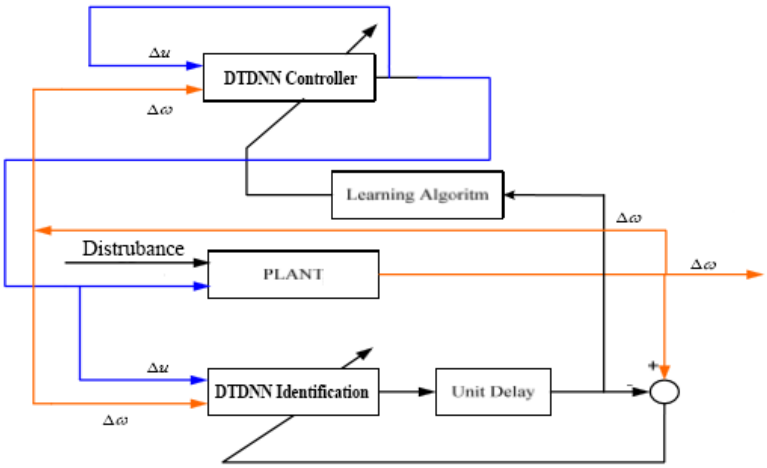


Figure. 4. Model DTDNN PSS

The training data to damp the speed oscillation is the plant output data in the form of speed with variation of the distrubance between 0.5 and 1.0 pu. The number of neurons used is 5.

In this study loading is assumed to be: P = 1.0 pu; Vt = 1.0 pu; Pf = 0.85 pu, and P = 0.5 pu; Vt = 1.0 pu; Pf = 0.85 pu. The disruption of 1 p.u is injected into the plant, and an output for the plant as shown in Fig. 5 is obtained.

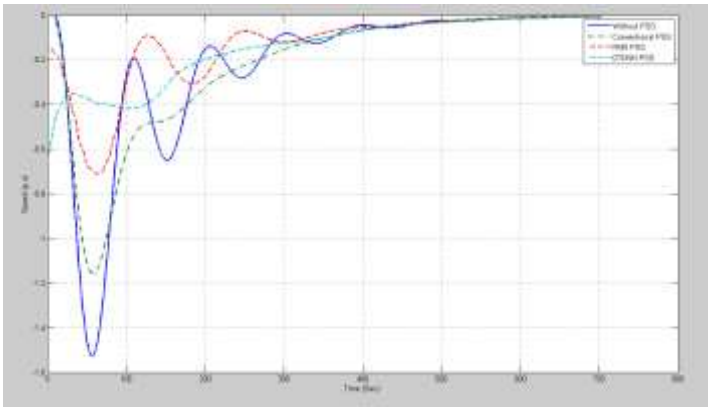


Figure 5. Speed in the nominal operating condition following disturbance 1 (p.u) with conventional PSS, RNN PSS and DTDNN PSS

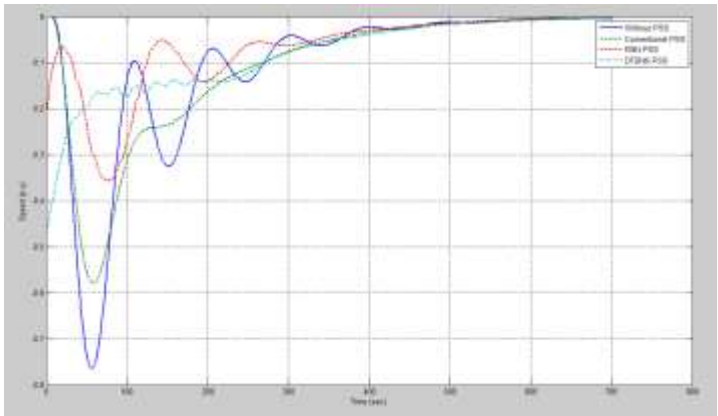


Figure 6. Speed in the nominal operating condition following disturbance 0.5 (p.u)) with conventional PSS, RNN PSS and DTDNN PSS

DTDNN PSS can decrease the overshoot speed to 0.6 p.u from its original state of 1.528 p.u. It's overshoot speed also better than RNN PSS which can only decrease by 0.7 p.u. The conventional PSS can only decrease the overshoot speed by 1.15 pu.

In Fig. 6, DTDNN PSS can decrease the overshoot speed to 0.45 p.u from its original state of 0.73 p.u. The overshoot speed also better than conventional PSS which can only decrease the overshoot speed by 0.576 p.u.

CONCLUSION

In this paper, we presented DTDNN PSS implementations for the Single Machine. DTDNN PSS is able to improve the performance of the Single Machine system in which DTDNN PSS is installed.

The comparison between RNN-PSS and DTDNN-PSS shows that DTDNN-PSS has better damping results at load 1 p.u. DTDNN-PSS can reduce the overshoot to 0.6 p.u at load 1 p.u load But RNN-PSS has better damping result at load 0.5 p.u. RNN-PSS can reduce overshoot to 0.34 p.u. The training time and iteration of DTDNN has better than RNN

The success of the design of Distributed Time-Delay Neural Network power system stabilizers (DTDNN PSS) is highly dependent on the data and the correct learning process.

ACKNOWLEDGMENT

This Work was Supported by Department of Electrical engineering, Faculty of Engineering, State University of Surabaya.

REFERENCES

Aribowo, W. (2010). Stabilisator Sistem Tenaga Berbasis Jaringan Syaraf Tiruan Berulang Untuk Sistem Mesin Tunggal. *TELKOMNIKA*, *8*(1), 65-72. http://dx.doi.org/10.12928/telkomnika.v8i1.606

Baliyan A, Gaurav K & Mishra SK (2015). A Review of Short Term Load Forecasting using Artificial Neural Network Models. *Procedia Computer Science*, *48*, 121-125. http://dx.doi.org/10.1016/j.procs.2015.04.160

Bhati, S.P. & Gupta, R. (2013). Robust fuzzy logic power system stabilizer based on evolution and learning. *International Journal of Electrical Power & Energy Systems*, *53,* 357-366. http://dx.doi.org/10.1016/j.ijepes.2013.05.015

Chaib, L., Choucha, A. & Arif, S. (2017). Optimal design and tuning of novel fractional order PID power system stabilizer using a new metaheuristic Bat algorithm. *Ain Shams Engineering Journal. 8*(2), 113-125. http://dx.doi.org/10.1016/j.asej.2015.08.003

Gruning, A. & Bohte, S.B. (2014). Spiking Neural Networks: Principles and Challenges. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Bruges, Belgium, 1-10.

Hagan, M.T., Demuth, H.B., Beale, M.H. & Jesu´s, O.D. (2014). *Neural Network Design.* 2nd Ed. Martin T. Hagan Publishing.

Ibrahim, L. M. (2010). Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN). *Journal of Engineering Science and Technology*, *5,* 457–471

Muamar, T., Amir, R. & Rahayu, Y. (2018). Power Supply Management System Design on Node Early Warning System for Peatlands Fire Mitigation. *SINERGI*, *22*(1), 29-36. http://dx.doi.org/10.22441/sinergi.2018.1.006

Ozerdem, C.O., Olaniyi, O.E. & Oyedotun, K.O. (2017). Short term load forecasting using particle swarm optimization neural network, *Procedia Computer Science*, *120*, 382-393. http://dx.doi.org/10.1016/j.procs.2017.11.254

Sambariya, K.D & Prasad, R. (2015). Optimal tuning of fuzzy logic power system stabilizer using harmony search algorithm, *International Journal of Fuzzy System*. *17*(3), 457–470. http://dx.doi.org/10.1007/s40815-015-0041-4

Sambariya, K.D. (2015). Power System Stabilizer Design using Compressed Rule Base of Fuzzy Logic Controller. *Journal of Electrical and Electronic Engineering*, *3*(3), 52-64. http://dx.doi.org/10.11648/j.jeee.20150303.16

Sambariya, K.D., Gupta, R. & Prasad, R. (2016). Design of optimal input–output scaling factors based fuzzy PSS using bat algorithm. *Engineering Science and Technology, an International Journal*, *19*(2), 991-1002. http://dx.doi.org/10.1016/j.jestch.2016.01.006