

An Adaptive RTRL Based Neurocontroller for Damping Power System Oscillations

K.C. Sindhu Thampatty*, P.C.Reghu Raj**

* Department of Electrical and Electronics Engineering, Amrita Viswa Vidyapeetham, Ettimadai, Coimbatore, Amritanagar (P.O), Tamilnadu, India.

** Department of Computer Science and Engineering, Govt. Engineering College, Sreekrishnapuram, Palakkad, Kerala, India.

Article Info

Article history:

Received Jan 2, 2015

Revised Feb 10, 2015

Accepted Mar 23, 2015

Keyword:

Thyristor Controlled Series Capacitor (TCSC) Thyristor controlled Power Angle Regulator (TCPAR) Real Time Recurrent Learning Algorithm (RTRL) Recurrent Neural Network (RNN) Dynamic Neural Network (DNN)

ABSTRACT

The main objective of this paper is to present the design of an adaptive neuro-controller for series connected FACTS devices like Thyristor Controlled Series Capacitor (TCSC) and Thyristor controlled Power Angle Regulator (TCPAR). This control scheme is suitable for non-linear system control, in which the exact linearised mathematical model of the system is not required. The proposed controller design is based on Real Time Recurrent Learning (RTRL) algorithm in which the Neural Network (NN) is trained in real time. This control scheme requires two sets of neural networks. The first set is a neuro-identifier and the second set is a neuro-controller which generates the required control signals for the thyristors. Performance of the system is analysed with the proposed controller using standard simulation environments like MATLAB/SIMULINK and it has been observed that the controller is robust and the response is very fast. Performance of the system with proposed controller is compared with conventional PI controllers and GA based PI controllers. Performance of the proposed controller is extremely good.

Copyright © 2014 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

K.C.Sindhu Thampatty
Department of Electrical and Electronics Engineering,
Amrita Viswa Vidyapeetham, Ettimadai,
Coimbatore, Amritanagar (P.O), Tamilnadu, India- 641112.
Email : kc_sindhu@yahoo.co.in

1. INTRODUCTION

In large interconnected power systems, low frequency electromechanical oscillations with frequency 0.2 Hz to 2 Hz have been observed frequently, which will reduce the efficiency of the system considerably. Conventionally, these oscillations are damped using Power System Stabilizers (PSS). In order to damp these electromechanical oscillations in power systems, supplementary control action, power oscillation damping (POD) can be applied to some of the FACTS devices.

Most of the controllers designed for damping power system oscillations [1]. [3] are based on a linearised model, the performance of which will deteriorate for wide varying operating conditions in presence of large disturbance. In order to overcome this, intelligent adaptive controllers are designed which does not require a perfect mathematical model of the system [4, 5]. The variation in plant parameters and plant structures can be effectively updated in an intelligent control strategy so that the system is more robust. A neural network based controller is such an intelligent controller which can operate very fast.

Most of the commonly available neural networks have drawbacks such as large training time, large number of neurons and more hidden layers required to deal with complex problems [6, 7]. The training techniques available for these neural network [8]. [10] are unable to retain the information about the infinite past which is essential for real-time applications. Dynamic Recurrent Neuron (DRN) with a feedback

connection from the output to the input is the most suitable architecture for real-time learning. DRN represents the internal or hidden states, in a potentially distributed fashion, which leads to capabilities that are similar to those of an observer in modern control theory.

Different training methods are available for recurrent neurons [11],[13]. Real time Recurrent Learning (RTRL) algorithm [14] is one of the most suitable algorithm for training the Recurrent Neural Network(RNN) for real-time application.

In all there ported works, a fully connected RNN is seldom used for the damping of power system oscillations. The proposed neuro-controller architecture is suitable for Multiple-Input Multiple-Output (MIMO) non-linear systems which can be trained by RTRL algorithm. It is an optimal algorithm which minimises the instantaneous squared error at the output of RNN for every discrete time, while the network is running. Number of neurons in the output layer of RNN is equal to the number of states of the system and the number of neurons in the controller network must besame as the number control inputs.

This paper examines the improvement in damping of power system oscillations with RTRL based neurocontroller for series connected FACTS devices like TCSC and TCPAR. A systematic procedure for modeling and simulation of a single-machine infinite-bus (SMIB) power system installed with TCSC and TCPAR is developed. The developed power system model is simulated using MATLAB/SIMULINK for different operating conditions and its performance verified.

The paper is organised as follows. Section 2 briefly explains the mathematical modelling of power system with TCSC and TCPAR. The detailed design of the proposed adaptive neuro-controller is explained in section 3 followed by the discussion on simulation results in section 4. Section 5 gives the conclusion.

2. POWER SYSTEM MODEL

In this study, SMIB system installed with TCSC and TCPAR is investigated. The synchronous generator is delivering power to the infinite-bus through series compensated transmission line as shown in Figure.1 in which the line is connected with TCSC and TCPAR. In Figure.1, V_s and V_r are the generator terminal and infinite bus voltage respectively. In this work, the mechanical system, electrical system, transmission lines and FACTS devices are modeled separately and are then interconnected to form the complete system [14].

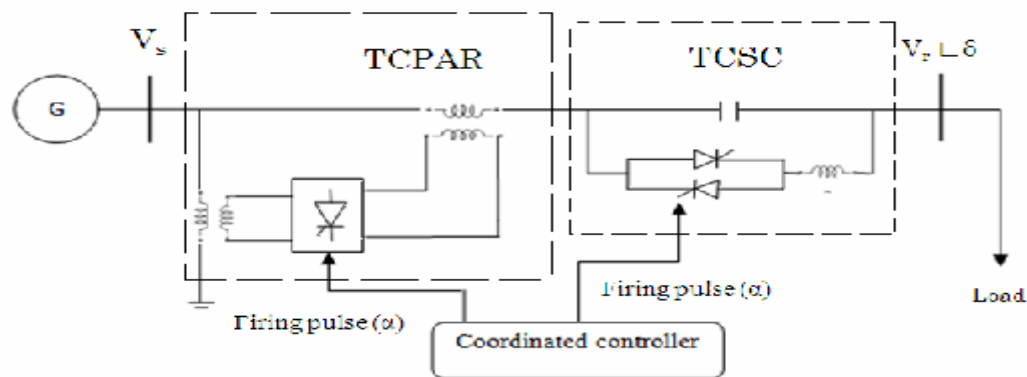


Figure 1. One Line Diagram of The System

2.1. Modelling of TCSC and TCPAR

Generally, the series connected FACTS devices are used to improve the power flow through the transmission lines. Due to the development in power electronics, the control of power flow can be done much faster. Also the enhancement of power system stability can be achieved with these fast acting switches. In this work, two series connected FACTS devices, TCSC and TCPAR are controlled by controlling the firing angles of the switches. A Quadrature Booster Transformer (QBT), injecting voltage in 90° with phase voltage is considered as the Power Angle Regulator. Modelling of TCSC and TCPAR are explained in the following sessions.

$$\frac{dV_a}{dt} = \frac{1}{C} \sqrt{\frac{2}{3}} [i_D \cos(\omega_0 t) + i_Q \sin(\omega_0 t) + \frac{i_0}{\sqrt{2}}] - \frac{i_{ra}}{C} \quad (1)$$

$$\frac{di_{Ta}}{dt} = \frac{v_a}{L} \quad (2)$$

where,

i_{Ta} : Current through TCR branch of TCSC
 L : Inductance of TCR and
 C : Capacitance of the TCSC.

When the thyristors are turned off, the current through the inductor is zero. In this condition, the TCSC can be modeled as a simple series capacitor circuit. The corresponding voltage equation for phase a can be written as:

$$\frac{dV_a}{dt} = \frac{1}{C} \sqrt{\frac{2}{3}} [i_D \cos(\omega_0 t) + i_Q \sin(\omega_0 t) + \frac{i_0}{\sqrt{2}}] \quad (3)$$

Combining equations 1, 2 and 3, the state equation of TCSC for phase a can be written as:

$$\dot{x}_a = [A]x_a + [B_a]I_{DQ0} \quad (4)$$

$$\dot{y}_a = PB_a I_{DQ0} \quad (5)$$

where

$$x_a = \begin{bmatrix} V_a \\ I_{Ta} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & -1/C \\ 1/L & 0 \end{bmatrix}, \quad I_{DQ0} = \begin{bmatrix} I_D \\ I_Q \\ I_0 \end{bmatrix},$$

$$B_a = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \omega_0 t / C & \sin \omega_0 t / C & 1/C \sqrt{2} \\ 0 & 0 & 0 \end{bmatrix}, \quad P = [1 \quad 0]$$

Similar state space equations can be written for other phases also. In order to operate the TCSC in the capacitive region, the firing angle α must be varied from 150° to 175° .

2.1.2. Modelling of Phase Angle Regulator

We adopt a quadrature voltage injection for regulating the phase angle [15]. This can be achieved using QBT. The injected voltage is given as :

$$\Delta V_p = kV_a \quad (6)$$

where,

p : Change in voltage phase due to voltage injection

V_a : voltage of phase a

k : the controlled variable which varies with respect to time.

QBT is connected through a controlled switch and the injected voltage can be adjusted by adjusting the firing angle α of the controlled switch. If P_a and P_b are the power flow at buses **a** and **b** respectively, then the power flowing through the transmission line can be represented as:

$$P_{ab} = \frac{V_a V_b}{X} \sin \delta_{ab} \quad (7)$$

where δ_{ab} is the angle difference between nodes a and b which is a function of θ where θ is the angle shift due to voltage injection. The equation for the change in real power flow through the transmission line by connecting PAR can be written as:

$$P_{ab} = \frac{V_a V_b}{X} \sin \delta_{ab} \quad (8)$$

where ΔP and $\Delta \delta$ are the changes in real power flow through the transmission line and voltage angle respectively. H is the matrix consisting of partial derivatives of power with respect to angle $\partial P_i / \partial \delta_j$ [1]. After modelling all other power system components [?], they are interconnected with TCSC and TCPAR. Then the overall state space equation can be written as:

$$\Delta \dot{X}_S = [A_S] \Delta X_S + [B_S] \Delta V_{DQ0} \quad (9)$$

$$\Delta I_{DQ0} = [C_S] \Delta X_S \quad (10)$$

where X_S includes all the system states. This system is having 16 states and some of the states are not observable. Hence a neural network based observer is designed which is a part of the proposed controller which is explained in detail in the next section.

3. RTRL BASED ADAPTIVE CONTROLLER

The proposed neuro-controller shown in Figure. 2 consists of a neuro-identifier and a neuro-controller, which can be realized using a fully connected *Dynamic Recurrent Neural Network* (DRNN) architecture [18] with all the system outputs fed back to the input through a delay. The neuro-identifier tracks the dynamic properties of the system and the neuro-controller provides the necessary control signal to the plant. A set of feedback weights controls the amount of feedback to each processing neuron.

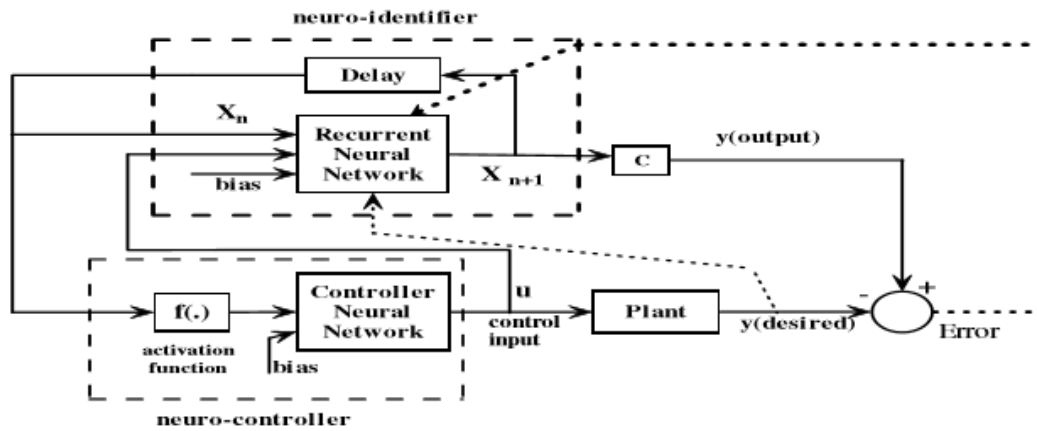


Figure 2. RTRL-Based Controller with Neuro-Identifier and Neuro-Controller

3.1. Recurrent Neural Network (RNN) Architecture

Recurrent Neural Networks is a dynamic systems which is capable of storing temporal informations [13]. The outputs of RNN are not only a function of the current inputs, but also a function of previous inputs and outputs and it is formulated for the state space realisation. The block diagram shown in Figure 3 shows the state space model of a recurrent neuron in which, the states of the system are updated with the external inputs and the activation from the previous forward propagation. It has two layers of neurons-a nonlinear hidden layer and a linear output layer as shown in Figure 3. The outputs from the hidden layer $x(n+1)$ are fed back to the input through a set of unit delays. The output of the linear output layer $y(n+1)$ is taken through unit delays which produce the output $y(n)$.

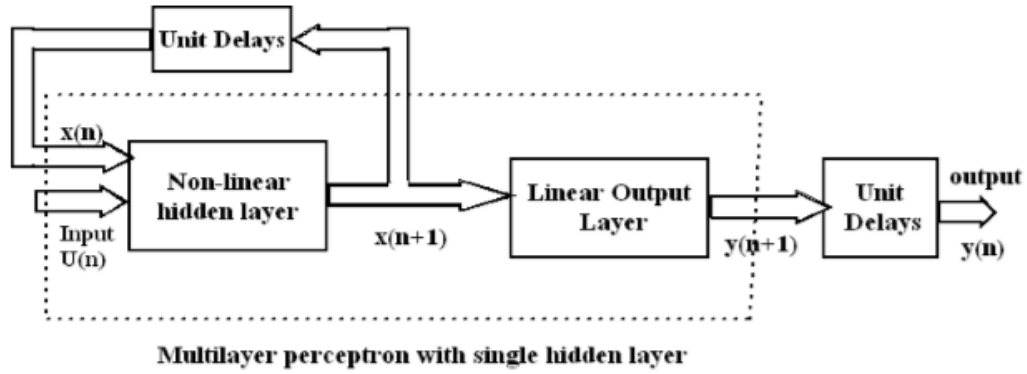


Figure 3. Block diagram representation of a state space model of recurrent neuron

The input of RNN consists of a concatenation of feedback nodes and the control nodes. Each neuron in the processing layer is activated by a non-linear activation function whereas the output neuron is activated by linear activation function. The number of unit delays (z^{-1}) used to feed the output of the hidden layer (processing layer) to the input layer determines the order of the model. For a fully connected recurrent neural network, the number of unit delays connecting between output and input will be equal to the number of states of the system. The architecture of the proposed controller using RNN is explained in detail in the next section.

3.2. Proposed Adaptive Neuro-Controller

Due to the nonlinear, time varying nature of the power system, it is essential for a controller to change its own behaviour according to the changes in the system. The controller proposed in this section is such an adaptive controller based on RTRL algorithm. RTRL-based controller includes two sub-networks. The first one is RNN which acts as a neuro-identifier that track the dynamic activities of the system. The second network acts as a neuro-controller to provide proper control signal to the system [14]. In each operation step, the controller generates a control input u that causes the plant to produce the desired output.

3.2.1. Neuro-identifier

The objective of this subnetwork is to identify a proper model of the system to be controlled. The main system to be identified by the controller can be represented as:

$$y(n+1) = F(y(n), u(n), bias) \quad (11)$$

where

$$y(n) = [y_1(n), y_2(n), \dots, y_p(n)] \quad : \text{output vector,}$$

$$u(n) = [u_1(n), u_2(n), \dots, u_m(n)] \quad : \text{control vector}$$

The recurrent neural network as neuro-identifier is trained using RTRL algorithm by assuming the nonlinear function F , which represent the dynamic relationship of the system. For the neuro-identifier, the input vector is :

$$[y_1(n), y_2(n), \dots, y_p(n), u_1(n), u_2(n), \dots, u_m(n)]$$

Where,

p : the total number of outputs fed back to the input side and

m : the total number of controls of the system.

The network is trained using a known desired trajectory. The predicted system output at $(n+1)^{th}$ interval is represented as $\hat{y}(n+1)$. Every discrete interval, weights of the neuro-identifier are updated using RTRL algorithm.

3.2.2. Performance Index of neuro-identifier

The *Performance Index* (PI) of the neuro-identifier shows the efficiency of the identifier in tracking the desired trajectory and it is calculated using Eqn. 12.

$$\begin{aligned} J(n) &= \frac{1}{2} \sum_{k \in \text{outputs}} [y_k(n) - d_k(n)]^2 \\ &= \frac{1}{2} \sum_{k \in \text{outputs}} [e_k(n)]^2 \end{aligned} \quad (12)$$

where $d_k(n)$ denotes the desired target value of the k^{th} output at n^{th} time interval.

The optimal value of PI after the convergence in each sampling interval is plotted with respect to time as shown in Figure. 4. Performance of this neuro-identifier is tested for various operating conditions by varying the real power from 0.2 pu to 1.2 pu. It has been observed that the proposed neuro-identifier performs well in all these conditions.

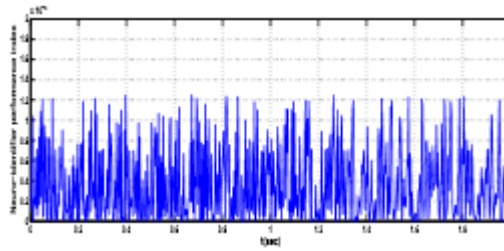


Figure 4. Performance Index of Neuro-Identifier

3.2.3. Neuro-controller

This is the second sub-network of the proposed RTRL-based controller. For the neuro-controller, the input vector is the system states which can be represented as:

$$x_1(n), x_2(n), \dots, x_q(n)$$

where x represent the updated system states and q represent the number of states of the system. These states are fed back from the output to the input side. When the desired output is given at any time interval, say $(n+1)^{th}$ instant, then the required control input is generated by the controller with the available knowledge of the states in the n^{th} time interval. The output of the neuro-controller is represented as a function of the system states and the bias as:

$$u(n+1) = f[x(n), bias] \quad (13)$$

where $x(n)$ is the state vector in n^{th} instant.

3.2.4. Performance Index of Neuro-Controller

The effectiveness of the neuro-controller is quantitatively measured using a performance index which is calculated as:

$$J_c(n) = \frac{1}{2} [\Delta \hat{y}(n) - y_d(n)]^2 \quad (14)$$

where,

$\Delta \hat{y}(n)$: $\frac{\partial \hat{y}(n)}{\partial u(n)}$: Change in predicted output with respect to a change in control input at n^{th} time interval and,
 $y_d(n)$: Desired output at n^{th} time interval.

Figure 5. shows the variation of PI of the neuro-controller with respect to time. For the neuro-controller, the desired output cannot be defined explicitly. Therefore, the neuro-controller has to be trained by driving the neuro-identifier and the error between the desired and predicted system output will be back propagated by the neuro-identifier and the states of the system are modified according to this error. The

required control to the system is generated based on this modified states. The parameters of the neuro-identifier and controller will be adjusted in every sampling period. This allows the controller to track the dynamic variations of the power system and provide the best control action. Each controller neuron is connected to all the states in the input layer and given an external bias.

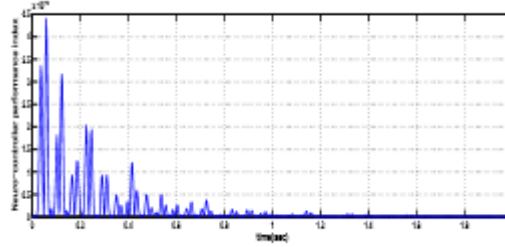


Figure 5. Performance Index of Neuro-Controller

The system states at $(n + 1)^{th}$ instant can be calculated as:

$$x(n + 1) = \phi[(W_a x(n) + W_b K)] \quad (15)$$

where,

ϕ : The non-linear activation function given by Eqn. 23

$K : [bias; control \ inputs]^T$

The desired output at $(n + 1)^{th}$ instant is given by:

$$y(n + 1) = Cx(n + 1) \quad (16)$$

where C is the output matrix of the system. The control input generated by the controller in $(n + 1)^{th}$ instant is given by:

$$u(n + 1) = \gamma \sum_i W_{ui} x_i(n) \quad (17)$$

where, W_{ui} : Synaptic weight associated with the neurons in the neuro-controller,

γ : Linear activation function used in the controller neuron and ,

$x_i(n)$: Updated i^{th} state in n^{th} instant. The execution of the algorithm is summarised as follows.

- a. At n^{th} time step, outputs are sampled as $y(n)$.
- b. $y_1(n), y_2(n)$... etc., are used to form input vector of neuro controller at n^{th} time step. At the same time, the weights of the neuro-identifier are updated using RTRL algorithm to minimise the error between the desired output y_d and the predicted output $\hat{y}$ at n^{th} instant.
- c. $y_1(n), y_2(n), \dots, u(n)$ are used to form an input vector for the neuro-identifier and the temporal output $\hat{y}(n + 1)$ is calculated.
- d. The weights of the neuro controller are updated to minimise the error after every execution of the algorithm.
- e. The system states are updated using Eqn. 15
- f. The output of the neuro controller $u(n)$ is calculated again with the latest updated states as the input vector and the new weights calculated in the previous step.
- g. The control signal $u(n)$ is applied to the plant, and to the neuro-identifier again to calculate $\hat{y}(n + 1)$ for $(n + 1)^{th}$ time step.

3.3. Real Time Recurrent Learning (RTRL) Algorithm

This is a forward gradient algorithm, which makes use of a matrix of partial derivatives of the network state values with respect to every weight. The algorithm attempts to minimize the instantaneous squared error at the output of the neuron. The main difficulty related to the recursive training of recurrent network arises from the fact that the output of the network and its partial derivatives with respect to the weights depend on the inputs. In this method, the partial derivatives of each node with respect to each weight are computed at every iteration [14]. If q is the number of states of the system, p the number of outputs and m the number of control inputs to the system, then the dynamic system can be represented by the non-linear difference equations given as :

$$x(n+1) = \Phi(W_a x(n) + W_b u(n)) \quad (18)$$

$$y(n) = Cx(n) \quad (19)$$

where W_a is a q -by- q matrix, W_b is a q -by- $(m+1)$ matrix and C is a p -by- q matrix.

Let $\Phi : R^q \rightarrow R^q$ be a non-linear map. The neural network with m inputs, p outputs and q states can be represented in state space form as given in Eqns. (20), (21) and (22).

$$x(n+1) = \begin{bmatrix} \phi(W_1^T \zeta(n)) \\ \vdots \\ \phi(W_j^T \zeta(n)) \\ \vdots \\ \phi(W_q^T \zeta(n)) \end{bmatrix} \quad (20)$$

$$W_j = \begin{bmatrix} W_{aj} \\ W_{bj} \end{bmatrix} \quad (21)$$

$$\zeta(n) = \begin{bmatrix} x(n) \\ u(n) \end{bmatrix} \quad (22)$$

The matrices, W_a , W_b , C and the non-linear function ϕ are interpreted as follows:

- In Eqn.(18), the total weights are split into two, namely, W_a and W_b . The matrix W_a represents the synaptic weights associated with the q neurons in the hidden layer that are fed back to the input layer. The matrix W_b represents the synaptic weights associated with this hidden layer, which are connected to the input sources including the bias. Thus, the bias terms of the hidden neurons are included in W_b .
- The matrix C represents the synaptic weights of p output neurons connected to the hidden layer.
- The neurons in the hidden layer are with hyperbolic tangent nonlinear activation function, given by:

$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (23)$$

or a logistic function:

$$\phi(x) = \frac{1}{1 + e^{-x}} \quad (24)$$

For every instant, the states are observed by neuro identifier and the required control signal is generated using RTRL algorithm [14], [18]

4. SIMULATION RESULT

A mathematical model of the system shown in Figure. 1 is developed. State space model of exciter, generator, transmission line, mechanical system and series connected FACTS devices are developed separately and then interconnected together to form full system. The required compensation is provided by TCSC and TCPAR. In this analysis we calculated an optimal level of compensation provided by TCSC and TCPAR and by keeping the TCPAR compensation constant, only TCSC is controlled. There are 16 states in the given system which is simulated at different operating conditions. In analysis, the machine damping is not included so that all the damping effects are due to the FACTS devices connected in series. The analysis has been done by giving a step change in the mechanical power input to the system. Performance of the system with the proposed controller is compared with those of the conventional PI controller and GA based PI controller which is explained in detail in the following sections.

4.1. GA Based PI Controller

A conventional PI controller is widely used in power system applications because it is simple in structure, low cost and highly reliable. However, improper PI parameter tuning can lead to slow recovery, poor robustness and in worst case the failure of the system operation. Genetic Algorithm (GA) is an optimisation algorithm which can be used for finding the optimal parameters for PI controller so that the parameters will always result in global minimum rather than in local minimum conditions. This section gives a comparison of the performance of the system with conventional controller and GA based controller. Parameters used in GA are given in Table 1.

Table 1. Parameters Used for Tuning GA

Parameter	Value/Type
Maximum generations	100
Population size	50
Type of selection Normal	[0 0.08]
geometric	[2]
Type of crossover Arithmetic	[2 100 3]
Type of mutation Nonuniform	Maximum generation

The optimised value of the tuning parameter for PI controller used is given in Table 2 where K is the gain, T_1 and T_2 are the time constants of the PI controller. The performance of the system with GA based PI controller is compared with the conventional PI controller as shown in Figure 8. It takes almost 3 seconds to reach steady state in case of conventional controller, whereas, with the GA based controller, variation reduces to zero before 2 seconds. Simulations are carried out for different operating conditions and most of the cases the performance of the system with GA based PI controller is far better than the conventional controller.

Table 2. GA Tuned PI Parameters

Gain	Time Constants	
K_T	T_1	T_2
30.3	0.4	0.6

4.2. RTRL Based ANN Controller

Detailed simulation of the test system with RTRL-based controller is executed using MATLAB for the same step change in mechanical input to the generator. The Power Angle Regulator is set for a constant injected voltage and the TCSC is controlled smoothly for the required operating conditions. Simulations are carried out for wide operating conditions where, the real power P is varied from 0.2 pu to 1.2 pu and the thyristor firing angle of TCSC is varied from 150° to 175° (Capacitive mode of operation of TCSC). The learning rate parameter η was taken as 0.42 for training the neural network which provides fast convergence. The maximum tolerance level set for the error is 0.005. Initially, the synaptic weights associated with each neuron are generated randomly for each sample.

The nonlinear activation function ϕ (Eqn. 23) and the linear activation function γ are selected for the current neural network and the controller neurons respectively. The initial states supplied to the neurons

are set to a small value. In this study, the desired output supplied to the neural network is set to be zero during training. The control signal generated by the proposed ANN controller is given to the system, which is the required triggering pulse for the thyristors in TCSC. There sponses are simulated for an initial change of 40% increase in the shaft torque. The control signal generated by the ANN controller is shown in Figure 6. This is the required deviation of thyristor firing angle for TCSC from the initial operating point, in order to damp the power system oscillations. The squared error plot of each sample is shown in Figure 7. where the error tolerance is set to a value 0.005.

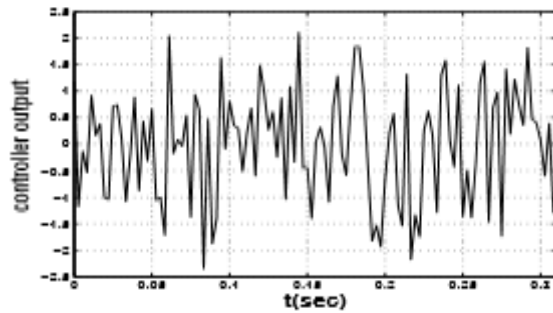


Figure 6. Expanded view of the output of the ANN controller

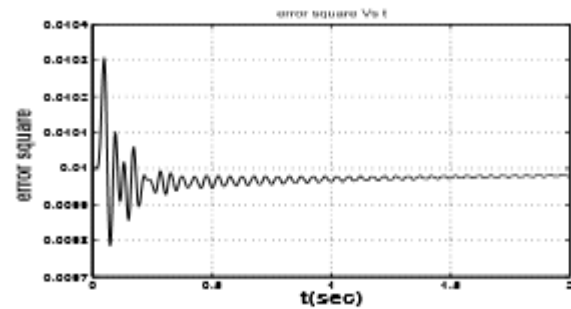


Figure 7. Squared Error

Figures 9. To 15 shows the performance of the system with different operating conditions. Fig. 9 shows the deviation of slip with GA based controller and ANN controller. The maximum deviation is 0.0015pu with ANN controller whereas 0.0038pu with GA based controller. Deviation of slip reduced to less than the tolerance value within one second in the case of ANN controller whereas it takes more than 5 seconds for GA based PI controller. Figure 10 shows the deviation of generator rotor angle δ and slip with the proposed ANN controller. In this case the real power is set to 0.8 pu and voltage = 1 pu. It has been observed from this figure that δ deviation reduces to zero before 0.1 second and slip reduces to less than tolerance level before 1.5 seconds. The deviation of torque without and with proposed ANN controller is shown in Figure 11. With ANN controller, the oscillations damped out before 0.7 seconds whereas without controller, it takes more than 2 seconds. The maximum amplitude of the oscillation is more with ANN controller but it exists only for first few cycles. Similarly in Figure 12, performance of the system is compared with GA based PI controller and ANN based controller for $P = 0.8$ pu and it has been observed that the performance of the system with ANN controller is far better than the GA based controller. A performance comparison is made with ANN controller and conventional controller as shown in Figure 13. The torque oscillations persists for a long duration with conventional controller whereas with ANN based controller, the damping is very fast eventhough the magnitude of oscillation is large during first few cycles. It has been observed that the performance of this is far better than with the conventional controller.

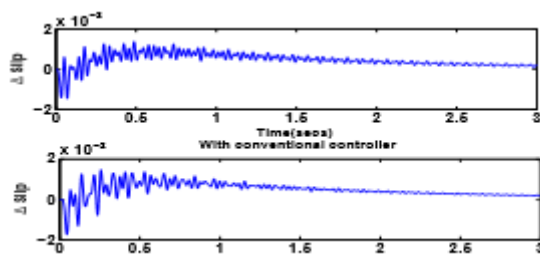


Figure 8. Variation of slip with conventional controller and GA based controller

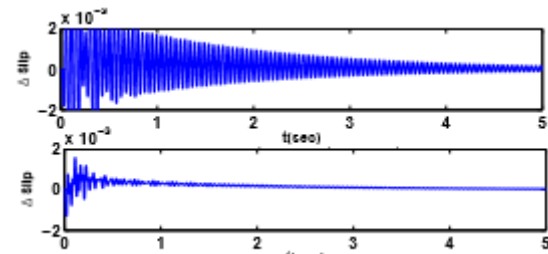


Figure 9. Variation of slip with a) GA based controller and b) with ANN controller $P = 1$ pu

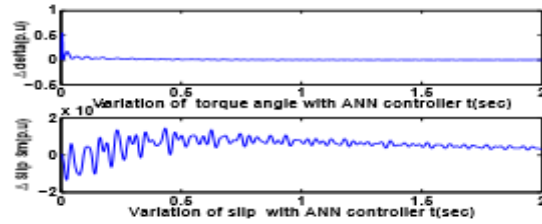


Figure10. Deviation of Rotor Angle δ and Slip With The Proposed ANN Controller for $P=0.8$ pu

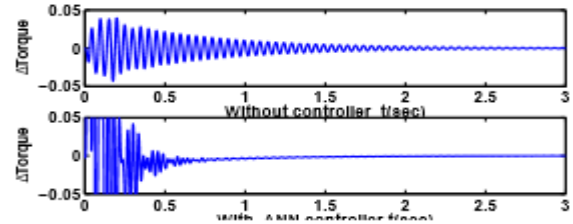


Figure 11. Variation of Torque Without Any Controller and with ANN Controller

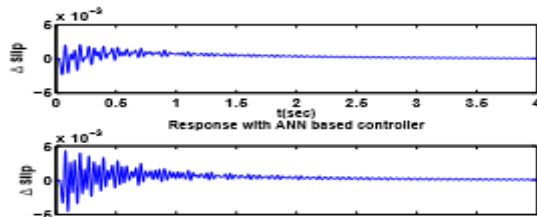


Figure 12. Variation of Slip with a) ANN Based Controller and b) with GA Based Controller for $P = 0.8$ pu

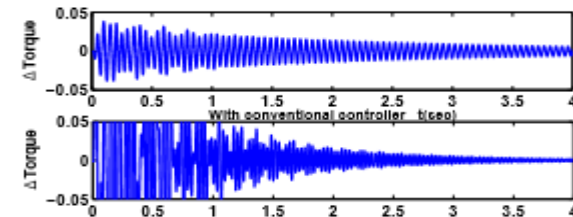


Figure 13. Variation of Torque with a) Conventional Controller and b) with ANN Based Controller $P = 0.8$ pu

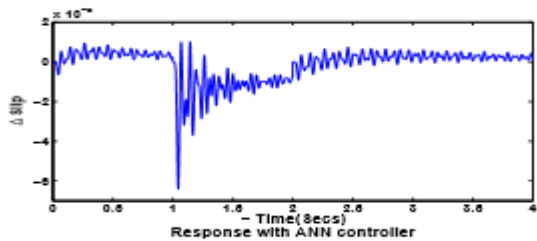


Figure 14. Variation of Slip with the Proposed ANN Controller

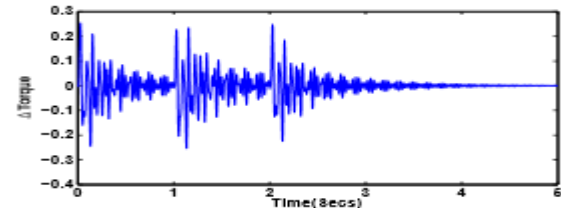


Figure 15. Response of the System with ANN Controller for Step Change in Power

Figures 14 and 15 shows the variation of slip and torque of the system with the proposed controller when the power is changed. In this case, at $t = 0$, power changes from 30% to 50%, at $t = 1$ sec, from 50% to 70% and then at $t = 2$ secs, power changes from 70% to 50%. In all these changes, it has been observed that the system damping is improved with the proposed controller.

4.3. Performace Comparison of the Power System with Conventional PI, GA Based PI and with The Proposed RTRL Based Controller

Comparison is made between the performance of the system with three different controllers conventional PI, GA based PI and with the proposed RTRL based controller for an operating voltage of $V = 1$ pu at the load side. Figure 16 and Figure 17 shows the deviation of Torque and delta for $P = 0.8$ PU and $V = 1$ pu. From all these analysis, it has been observed that the proposed ANN controller is robust and the damping of the power system oscillations can be improved to a large extent with this controller.

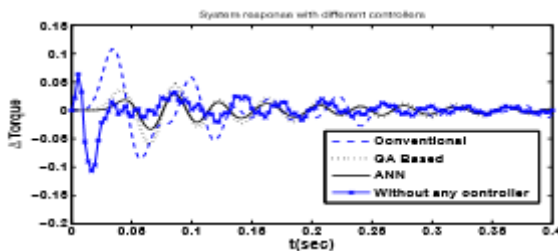


Figure 16. Variation of torque with different controllers for $P = 0.8$ pu, $V = 1$ pu

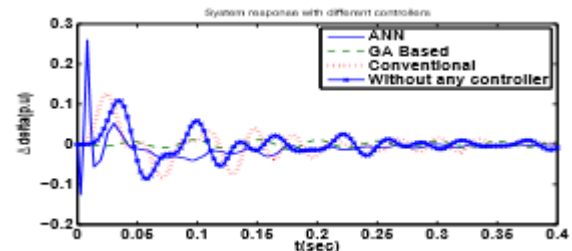


Figure 17. Variation of torque angle with different controllers for $P = 0.8$ pu, $V = 1$ pu

5. CONCLUSION

An adaptive neuro-controller based on RTRL algorithm is developed for TCSC control to enhance the damping of power system oscillations. In this controller, the neural network is trained using Real Time Recurrent Learning algorithm. From the simulation results obtained, it can be observed that the proposed neurocontroller provides better damping for power system oscillations than the conventional controllers and enhances the transient stability of the system. The proposed technique does not require an accurate modeling of the system and hence this controller is suitable for the non-linear system control. Moreover, the fast response of the controller makes the ANN based approach very attractive for on-line applications in non-linear system control.

REFERENCES

- [1] James F Gronquist, William A Sehares, Fernando L Alvarado, Robert A Lasseeter "Power Oscillation damping control strategies for FACTS devices using locally measurable quantities," *IEEE Transactions on Power Systems* , Vol 10, No. 3, August 1995 pp.1598- 1605.
- [2] SalehMBamasak,M.AAbido"Robust coordinated Design of PSS and STATCOM controllers for damping Power system oscillations," 15 th PSCC, Lege, , 22-26, August 2005.
- [3] R.D.Saxena K.D. Joshi "Application of Unified Power Flow controller(UPFC) for damping Power System Oscillations-AReview,"*International Journal of Engineering Research and Technology (IJERT)*,No.4,June 2012
- [4] L. Ender , R. Maciel Filho, "Design of multivariable controller based on neural networks," Elsevier, *Computers and Chemical Engineering*, Vol 24, 2000, pp.937-943.
- [5] T. A. Al-Zohary,Wahdan, M. A. R. Ghonaimy and A. A. Elshamy, "Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks and Approximate Models,"
- [6] Wei Qiao, Ronald G. Harley,"An Indirect Adaptive External Neuro-Controller for Series Capacitive Reactance Compensator in Damping Power Oscillations," *IEEE Transactions on Industrial Electronics*,Vol.54, No.1, 2007, pp.77-85.
- [7] N. Magaji and M.W. Mustafa, "Optimal Thyristor Control Series Capacitor Neuro-Controller for Damping Oscillations," *Journal of Computer Science*,Science Publications, Vol.5(12), 2009, pp.983-990.
- [8] Narendra K.S, Parthasarathy K., "Identification and control of dynamical system using neural networks,"*IEEE Trans on Neural Networks*, Vol.1, No. 6, 1990, pp.4-27.
- [9] Jin,L, Nikiforuk,P, Gupta M, "Approximation of discrete time state space trajectories using dynamic recurrent networks,"*IEEE Trans on Automatic Control*, Vol.40, No.7, 1995, pp.1266-1270.
- [10] Steepen W.Riche Webrose , "Steepest Descent Algorithm for Neural Network controllers and Filters," *IEEE Trans on Neural Networks*, Vol.5, No. 2, 1994, pp.198-212.
- [11] Pearlmutter.B,"Gradient calculations for Dynamic Recurrent Neural Networks: A Survey,"*IEEE Transon Neural Networks*, Vol.6, No.3, 1995, pp.1212-1228.
- [12] Williams R.J and Zipser D , "A learning algorithm for continually Running fully Recurrent Neural Networks," *Neural Computation* , Vol.1, 1989, pp.552-558.
- [13] Danilo P. Mandic, Jonathan A Chambers," A normalised real-time recurrent learning algorithm," *Signal Processing*. Elsevier, 80 (2000), pp.1909-1916.
- [14] K.C.Sindhu Thampatty, M.P. Nandakumar and Elizabeth P.Chериан, "Adaptive RTRL Based Neurocontroller for Damping Sub-synchronous Oscillations using TCSC," *Engineering Applications of Artificial intelligence*, Elsevier, Vol.24, Issue 1, 2011, pp.60-76.
- [15] Narain G Hingoraniand Laszio Gyugyi,Understabling FACTS, *IEEE Power Engineering Society*, Delhi 110006.
- [16] Prabha Kundur,Power System Stability and Control, McGraw-Hill Inc., New York, 1994
- [17] Padiyar K.R ,Power System Dynamics: Stability and Control, BS Publications, 2002
- [18] Simon Haykin, "Neural Networks-A comprehensive Foundation," *IEEE Press*, 1994.