Chapter 4

Design of PSS and TCSC Using ANFIS and ANN

4.1 Introduction

The Adaptive Neuro-Fuzzy Inference System and Levenberg-Marquardt Artificial Neural Network algorithm for the development of the control strategy for thyristor control series capacitor based damping controller and power system stabilizer has been discussed in this chapter. In order to achieve the appreciable damping, the series capacitor has been suggested in addition to power system stabilizer. The non-linear simulations of single machine infinite bus system (SMIB) have been carried out using individual and simultaneous application of PSS and TCSC. The comparison between intelligent control strategies based damping controllers has been carried out. The results have shown efficacy and capability of proposed control schemes under the various operating conditions, disturbances and fault conditions, and also have demonstrated the improvement in the dynamic performance of the system with proposed control algorithm.

4.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Combining the learning power of neural network with knowledge representation of Fuzzy logic gives a Neuro-Fuzzy system. It gives the advantage of neural networks as well as of fuzzy logic system and it removes the individual disadvantages by combining them on the common features. Fuzzy logic has tolerance for imprecision of data, while neural networks have tolerance for noisy data. Fuzzy logic provides a structure within which the learning ability of neural networks is employed and neural network can be used to generate the membership functions for a fuzzy system and to tune them. There are two ways of hybridization; one is to endow NNs with fuzzy capabilities, thereby increasing the network's expressiveness and flexibility to adapt to uncertain environment. The second aspect is to apply neuronal learning capabilities to fuzzy system to make the fuzzy systems more adaptive to changing environment.

4.2.1 ANFIS- Architecture

It is assumed that the fuzzy inference system under consideration has two inputs x and y and one output z. For a first order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is the following:

Rule 1: If x is A_1 and y is B_1 then $f_1 = p_1x+q_1y + r_1$

Rule 1: If x is A_2 and y is B_1 then $f_2 = p_2x+q_2y + r_2$

The reasoning mechanism for the Sugeno model has been shown in Figure 4.1 and corresponding equivalent ANFIS architecture has been shown in Figure 4.2. The ANFIS algorithm has been described in literature [13, 87].

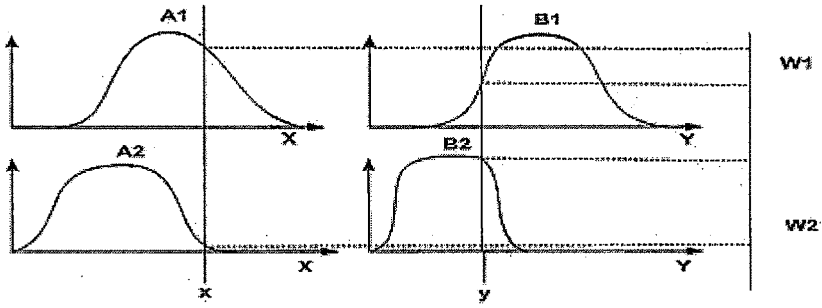

Figure 4.1: Takagi-Sugeno Fuzzy Model

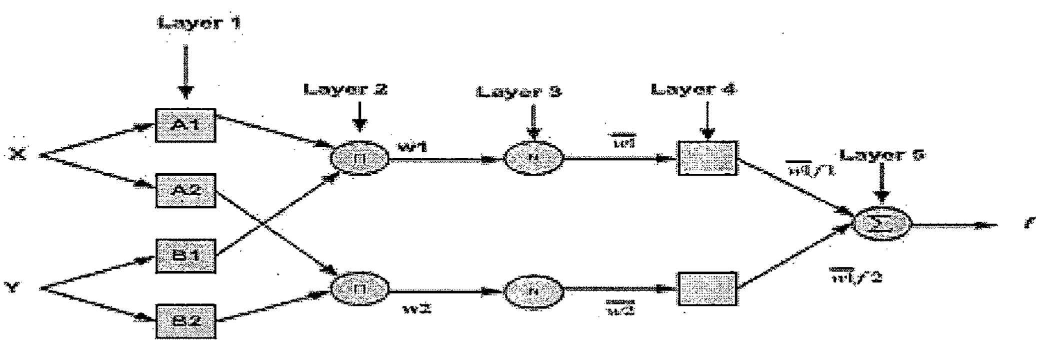

Figure 4.2: Adaptive Neuro Fuzzy Architecture

Layer 1: Every node i in this layer is an adaptive node with a node function

$O_{1,i} = \mu A_i(x)$, for $i = 1, 2$or

$O_{1,i} = \mu B_{i-2}(y)$, for $i = 3, 4$

Where x (or y) is the input to node A_i(or B_{i-2}) is a linguistic label associated with this node. Here the membership functions of A can be appropriate parameterized membership function such as generalized bell function:

$$\mu_{A(x)} = \frac{1}{1 \left| \frac{x-c_i}{a_i} \right|^{2b}} \tag{4.1}$$

Where $\{a_i, b_{i,} c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped function varies accordingly, thus exhibiting various forms of membership functions for fuzzy set A. Parameters in this layer are referred to as premise parameters.

Layer 2: Every node in this layer is a fixed node labeled II, whose output is the product of all the incoming signals:

$$O_{2,i} = w_i = \mu_{Ai}(x)\mu_{Bi}(y), \quad i = 1, 2. \tag{4.2}$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy AND can be used as the node function in this layer.

Layer 3: Every node in this layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules's firing strengths:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \tag{4.3}$$

Outputs of this layer are called normalized firing strengths.

Layer 4: Every node i in this layer is an adaptive node with a node function:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{4.4}$$

Where $(\bar{w}_i)$ is a normalized firing strength from layer 3 and $\{p_{i,} q_{i,} r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled Σ, which computes the overall output and the summation of all incoming signals :

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{4.5}$$

From the ANFIS architecture as shown in Figure 4.2, we observe that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. The output f in Figure 4.1 can be expressed as below:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_1}{w_1 + w_2} f_2 \tag{4.6}$$

$$= \bar{w}_1(p_1 x + q_1 y + r_1) + w_2(p_2 x + \bar{q}_2 y + r_2)$$

$$= (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + \bar{w}_1 r_1 + (\bar{w}_2 x)p_2 + (\bar{w}_1 y)q_2 + \bar{w}_2 r_2$$

4.3 Steps for Designing of ANFIS based PSS and TCSC

1. To generate the input data pattern and corresponding the target data pattern.
2. To develop the fuzzy inference system (FIS).
3. To select the number and type of the membership functions.
4. Application of rules extracted algorithm to initialize the rules of FIS.
5. Training of the FIS using learning algorithm such as hybrid learning or backpropagation learning algorithm [87, 13].
6. Testing of FIS through testing data.
7. If desired solution is achieved then stop training, else to change the fuzzy membership functions, fuzzy membership function types, change the rules extracted algorithm, FIS learning algorithm, no. of epochs, error tolerance and repeat the algorithm from the step 2.
8. Implementation of ANFIS in real system and compute the output of the system.

4.3.1 ANFIS-Power System Stabilizer

In this work, CPSS is replaced by the ANFIS based PSS. The mathematical model of the CPSS has been used for the generation of the training data for the ANFIS. The Takagi-Sugeno FIS [87] is used for the design of ANFIS based PSS. Sugeno has high computational efficiency and it works well with optimization and adaptive techniques [89, 90]. The network has been trained using 2000 sample training data, which are generated under the consideration of the different operating conditions and dynamic behavior of the power system. The two inputs and one output have been used for the training of ANFIS. The dynamic inputs are speed $\Delta\omega_m(t)$ and change in speed $(\frac{\Delta d\omega_m(t)}{dt})$ and corresponding ΔV_{pss} has been selected as output value of the ANFIS. Figure 4.3 shows the fuzzy logic controller (FLC) based PSS with inputs and output.

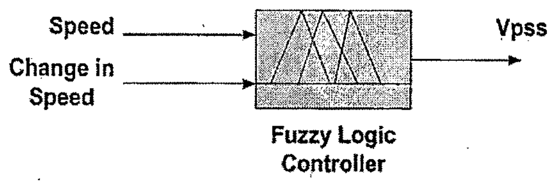

Figure 4.3: Fuzzy Logic Control based PSS

Generally gbell and gauss types membership functions (MFs) are preferable for ANFIS controller. No any convention for selection of type of MFs. The general rule is to produce a satisfactory response of the system with ANFIS controller in minimum time and also to obtain minimization of error with minimum ANFIS training parameters. The selection of number of membership function such that less number of membership function produce satisfactory response of the system. Consequently memory utilization should be reduced and ANFIS controller should produce quick response in less time. Table 4.1 shows the generating error after the execution of PSS-ANFIS structure with inputs and output data. Here four types of membership functions such that gbell, gauss, gauss2 and dsig are selected with different number of rules for designing of inputs and output variables of ANFIS controller. The triangle and trapezoidal membership functions are not suitable for ANFIS structure. Normally they are used for design of fuzzy logic controller based system. For two inputs, 3, 4, 5 and 7 variables are selected and corresponding 9, 16, 25 and 49 rules are developed. Here forty nine rules based structure of the PSS-ANFIS controller is not preferable as it takes high computational time for the execution compared to the other ANFIS structures. The best ANFIS-PSS structure has been selected after the testing with real power system. The gbell type four membership functions are selected and total sixteen rules are developed for inputs and output.

Speed signal range is selected between -0.002527 to 0.003344 and change in speed signal -0.06441 to 0.1094 for defining the inputs membership functions. The output membership functions are varied between -0.09217 to 0.1112. Figure 4.4 represents the ANFIS-PSS structure with inputs and output. Figure 4.5 and 4.6 have represented gbell type four linguistic membership functions. Figure 4.7 shows the decision surface viewer of inputs and output of the PSS.

For initializing of FIS rules, the grid partition method has been used and the initial rules are extracted. The hybrid learning algorithm has been used for training to modify FIS parameters after obtaining the application of grid partition method. The hybrid algorithm combines the least square and backpropagation gradient descent algorithm. In the hybrid algorithm [87], as shown in Figure 4.2, the node outputs go forward until layer 4 and consequence parameters are estimated by least-squares method. In the backward phase, the error signals propagate backward and the algorithm iteratively learns the premise parameters by gradient descent. The training is continued until the error becomes constant. 10 numbers of epochs are selected for training of ANFIS based PSS. The numbers of epochs are selected such that expected goal may be achieved with minimum value of error. After the 10 number of iteration, the error between actual output and ANFIS output is minimized. The training is continued until the desired error becomes constant. The training is completed when the constant error 0.00248381has been reached.

Table 4.1: Membership Function of ANFIS-PSS

No.of Rules	Type of Membership Functions			
	gbell	gauss	gauss2	dsig
	Error			
9	0.00305385	0.00295805	0.00265287	0.00287214
16	0.00248381	0.00303126	0.00298624	0.00320831
25	0.00325109	0.00285967	0.00240166	0.00406903
49	0.0016727	0.00158654	0.00207032	0.0033497

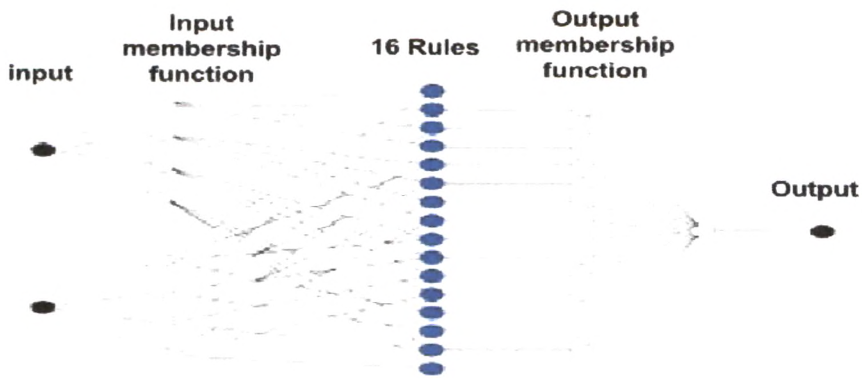

Figure 4.4: ANFIS-PSS Structure

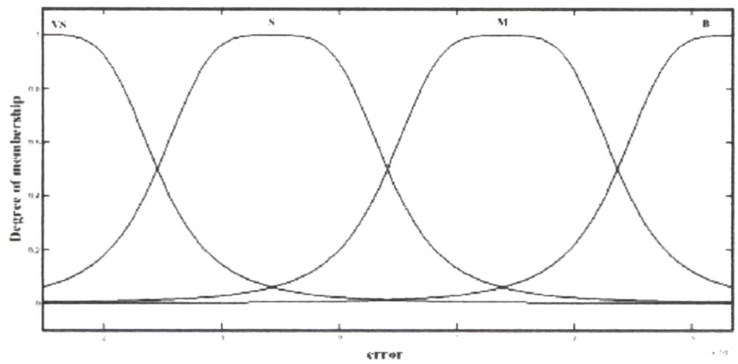

Figure 4.5: Membership Functions of Speed (Input :1)-PSS

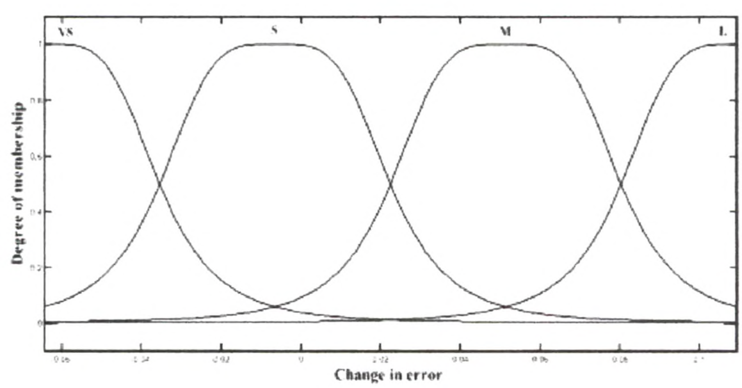

Figure 4.6: Membership Functions of Change in Speed (Input :2)-PSS

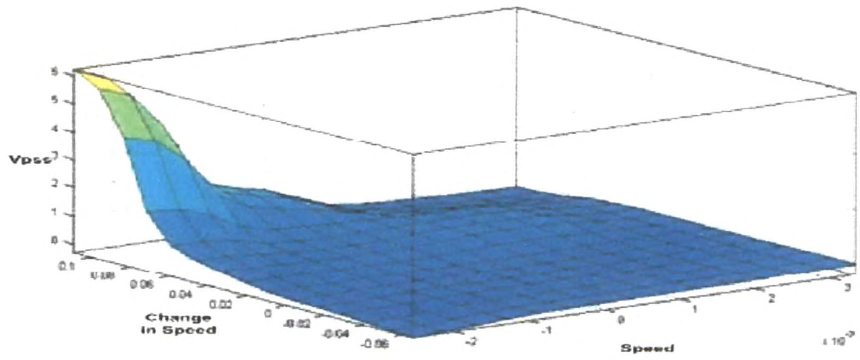

Figure 4.7: Decision-Surface Viewer of PSS

4.3.2 ANFIS-Thyristor Control Series Capacitor

Here, the stability control loop of the TCSC has been designed and replaced by sugeno ANFIS. The stability control loop of the TCSC has been used for generation of the training data pair of ANFIS under the consideration of plant dynamics. The inputs are speed $\Delta\omega_m(t)$ and change in speed $\frac{\Delta d\omega_m(t)}{dt}$, and corresponding X_{mod} has been selected as output value of the ANFIS. The input signals of TCSC are speed and acceleration, so communication delay has been taken into consideration to compensate the time lag between generator signal and transmission side input of the TCSC. The first order transfer function with 0.5 second delay has been used as communication time delay for generation of input and output data. The comparative analysis between types of MFs with different number of rules are illustrated by Table 4.2. Table 4.2 shows the generating error after the execution of ANFIS structure with inputs and output data. Here three types of membership functions are selected with different number of rules for designing of inputs and output variables of ANFIS controller. It has been experimentally verified that forty nine rules based structure of the ANFIS controller is not preferable as it takes high computational time for the execution compared to the other ANFIS structures. Type of MFs, number of MFs and number of FIS rules are selected such that generating error should be minimized. The best ANFIS-TCSC structure has been selected after the testing with real power system. The TCSC model has been implemented in power system with ANFIS based stability control loop. Here the gauss type five membership functions are selected and total twenty five rules are designed for inputs and output. Speed signal range is selected between -0.003313 to 0.003808 and change in speed signal -0.05169 to 0.1172 for defining the input membership functions. The output membership functions are varied between -0.2143 to 0.234. After 10 number of epochs, the error between actual output and ANFIS output is minimized. Figure 4.8 represents structure of ANFIS based TCSC controller. Membership functions for the ANFIS based TCSC controller have been represented in Figure 4.9 and 4.10. The decision surface viewer of ANFIS-TCSC controller has been presented by Figure 4.11. The training is continued until the desired error becomes constant. The training is completed when the constant error 0.000805447 has been reached.

Table 4.2: Membership Function of ANFIS-TCSC

No.of Rules	Type of Membership Functions		
	gbell	gauss	dsig
	Error		
9	0.00169247	0.00084397	0.00115811
16	0.00029095	0.000753772	0.00158915
25	0.00131741	0.000805447	0.000930369
49	0.000256792	0.000494053	0.00674250

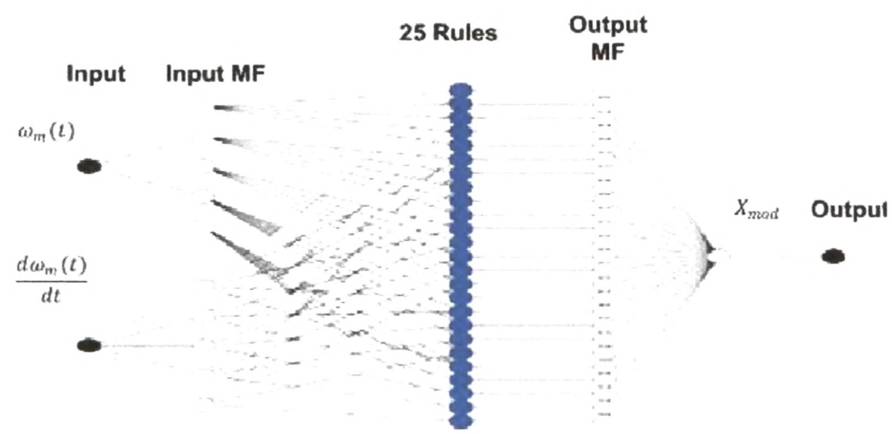

Figure 4.8: ANFIS-TCSC Structure

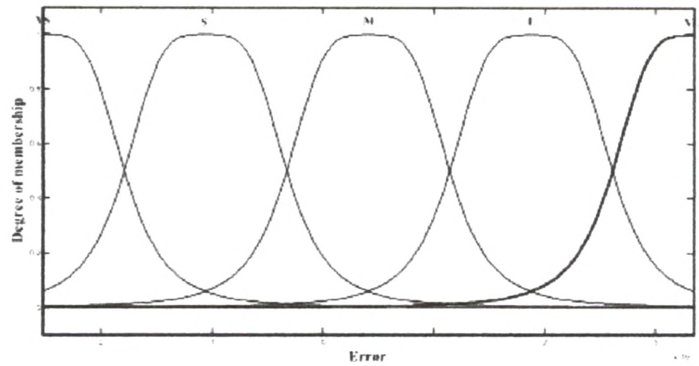

Figure 4.9: Membership Functions of Speed (Input: 1)-TCSC

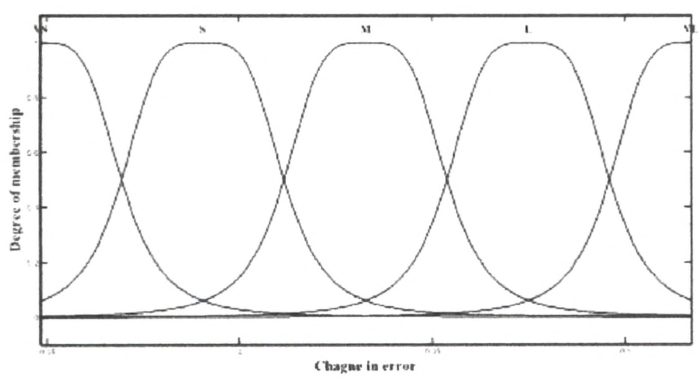

Figure 4.10: Membership Functions of Change in Speed (Input :2)-TCSC

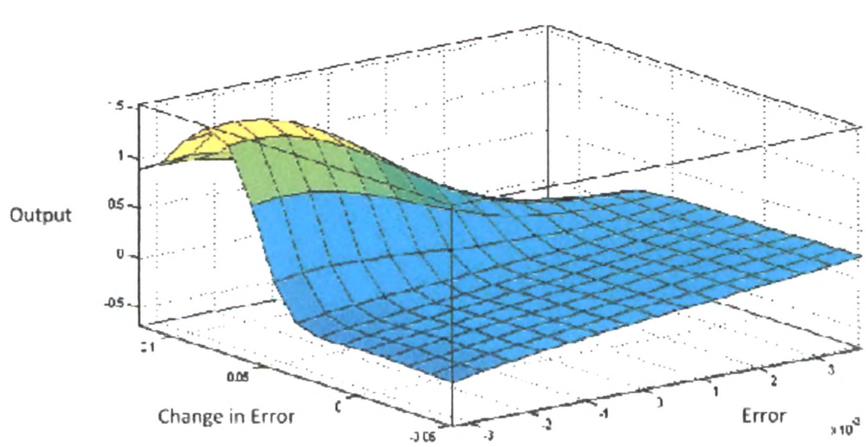

Figure 4.11: Decision-Surface Viewer of TCSC

4.4 Levenberge-Marquardt Neural Network

Artificial Neural Network has been one of the most interesting topics in the control community because they have the ability to treat many problems that cannot be handled by traditional analytical techniques. There are several approaches to neural network training, for determining an appropriate set of weights. The feedforward multilayer neural networks are the most common neural network architecture for solution of control problem. A widely used training method for feedforward multilayer neural network is the back propagation algorithm; the standard back propagation learning algorithm has several limitations. Most of all, a long and slow training process when plant is non-linear and parameters of the plant are dynamic i.e. the rate of convergence is seriously affected by the initial weights and the learning rate of parameters. Here, the learning rule is common to a standard nonlinear optimization or least-squares technique. The adjustment of weight is done at the end of each iteration and the sum of squares of all errors is used as the objective function for the optimization problem. In this problem derivative –based optimization Levenberg-Marquardt method [87, 14] is used for solving the nonlinear least squares problem. The Gauss Newton Levenberg-Marquardt method works well in practice and has become standard of nonlinear least squares routines.

4.4.1 Levenberg-Marquardt Algorithm

To implement the Levenberg–Marquardt algorithm [87, 14, 94] for neural network training, the first step is calculation of Jacobin matrix and second step is organize the training process iteratively for weight updating. Suppose that we have a function $V(k)$ to minimize with respect to the parameter k vector, and then Newton's method would be

$$\Delta k = -\left[\nabla^2 V(k)^{-1}\right] \nabla V(k) \tag{4.7}$$

Where is $\nabla^2 V(k)^{-1}$ Hessian matrix and $\nabla V(k)$ is the gradient.

$\nabla\omega(k)$ - sum of square funciton

$$V(k) = \sum_{i=1}^{N} e_i^2(k) \tag{4.8}$$

Then it can be shown that

$$\nabla V(k) = J^T(k)e(k) \tag{4.9}$$

$$\nabla^2 V(k) = J^T(k)J(k) + s(k) \tag{4.10}$$

$$J(k) = \begin{bmatrix} \frac{\partial e_1(k)}{\partial \theta_1} & \frac{\partial e_1(k)}{\partial \theta_2} & \cdots & \frac{\partial e_1(k)}{\partial \theta_n} \\ \frac{\partial e_2(k)}{\partial \theta_1} & \frac{\partial e_2(k)}{\partial \theta_2} & \cdots & \frac{\partial e_1(k)}{\partial \theta_n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \frac{\partial e_N(k)}{\partial \theta_1} & \frac{\partial e_N(k)}{\partial \theta_2} & \cdots & \frac{\partial e_N(k)}{\partial \theta_n} \end{bmatrix} \tag{4.11}$$

$$s(k) = \sum_{i=1}^{N} e_i(k)\nabla^2 e_i(k) \tag{4.12}$$

The updated rule of Levenberg Marquardt to the Gauss-Newton method is

$$\theta_{k+1} = \theta_k - \left[J(k)^T J(k) + \alpha I\right]^{-1} J(k)e(k) \tag{4.13}$$

Where $J(k)$ is Jacobian matix, α is always positive called combination coefficient, I is the identity matrix. As the combination of the steepest descent algorithm and the Gauss–Newton algorithm, the Levenberg–Marquardt algorithm switches between the two algorithms during the training process. When the combination coefficient α is very small, the Gauss–Newton algorithm is used while combination coefficient α is very large; the steepest descent method is used. With the update rule of the Levenberg–Marquardt algorithm equation (4.13) and the computation of Jacobian matrix, the next step is to organize the training process.

4.5 Steps for Designing of LMNN based PSS and TCSC

1. To generate the input data pattern and corresponding the target data pattern.
2. To Develop the feedforward neural net
3. Training of the neural network using Levengerg-Marquardt algorithm
4. To Update the NN parameters through equation (4.13).
5. Calculating mean square error between actual output and targeted output.
6. Computing the output of the NN.
7. If desired solution is achieved then stop, else change the NN goal, learning rate, no. of epochs and repeat the algorithm from the step 4.

4.5.1 ANN-Power System Stabilizer

In this work, CPSS is replaced by the ANN based PSS. The time constants and gain of lag-lead compensator based CPSS has been tuned by genetic algorithm [8] and genetic algorithm tuned model of the CPSS has been used for the generation of the training data for the artificial neural network. The network has been trained using 8000 sample data, which are generated under the consideration of the different operating conditions and dynamic behavior of the power system. The training pattern for the feedforward neural network is dynamic inputs $u(t)$ and corresponding outputs $y(t)$ such that $\omega_m(t), \omega_m(t-1), \omega_m(t-2), \omega_m(t-3)$ and V_{pss} respectively and targeted value of the neural network is $\hat{y}(t)$.

Table 4.3 represents the mean square error (mse), number of iteration and training time in second with different learning rate and different combination of neurons in input layer, hidden layer and output layer. The different combinations of neurons are selected and trained the neural network such that error should be reduced with less number of iteration and less time. Each trained neural network has been tested with real power system and performance has been analyzed. Here dynamic data are used for training of neural network so large number of neurons is required for satisfactory performance of the system. Hence the feedforward network has been developed with 50 neurons in first layer, 30 neurons in hidden layer and 1 neuron in output layer with hyperbolic tangent sigmoidal transfer function in first layer and hidden layer, and linear transfer functions in output layer. The selection of [50 30 1] neurons with higher learning rate neural network has produced better response.

Table 4.3: Mean Square Error of ANN based PSS

No.of neurons	Learning rate					
	0.05			0.1		
	Iteration	t (sec)	mse	Iteration	t (sec)	mse
[10: 5:1]	35	6	2.64×10^{-5}	11	2	3.01×10^{-5}
[20: 10: 1]	89	24	2.72×10^{-5}	25	7	2.96×10^{-5}
[30: 20: 1]	16	10	2.96×10^{-5}	12	8	2.80×10^{-5}
[50: 30: 1]	24	11	3.06×10^{-5}	27	12	2.98×10^{-5}

Contd..

0.3			0.5		
Iteration	t (sec)	mse	Iteration	t (sec)	mse
39	7	2.83×10^{-5}	32	6	2.98×10^{-5}
108	29	2.81×10^{-5}	11	3	2.96×10^{-5}
26	16	2.60×10-5	13	8	2.97×10^{-5}
15	34	2.88×10^{-5}	5	24	3.84×10^{-5}

During the training of ANN, the weights and bias of the network are adjusted such that the error between the actual output and targeted output is minimized and desired goal is achieved through Levenberg-Marquardt derivatives –based optimization. The optimization function can be represented by the following equation.

$$J_i(k) = \frac{1}{2}\left[\Delta\omega_m(k) - \Delta\hat{\omega}_m(k)\right]^2 \tag{4.14}$$

The Levenberg-Marquardt's direction is determined by the updated rule of Levenberg Marquardt to the Gauss-Newton method. Which one is an intermediate between the Gauss-Newton direction and the steepest descent direction. Figure 4.12 shows relation between training data versus output data and targeted data. Figure 4.13 shows the minimizing of the cost function $J_i(k)$ described by the equation (4.14). The selected structure has produced

3.84×10^{-5} mse in 24 second with 0.5 learning rate. The training is performed for 5 number of iteration through appropriate adjustment of weight and bias of neural network.

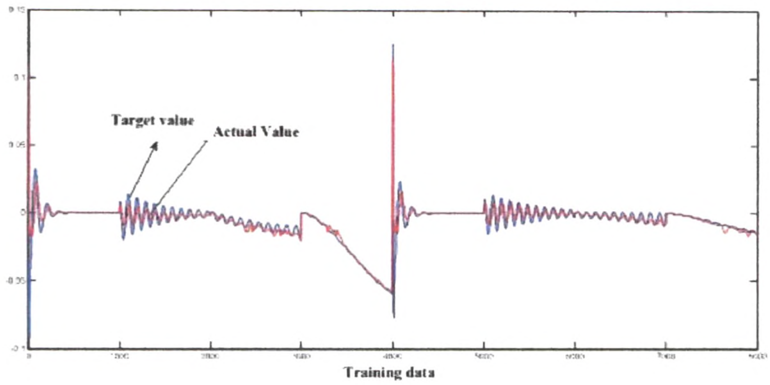

Figure 4.12: Training data and Actual data PSS

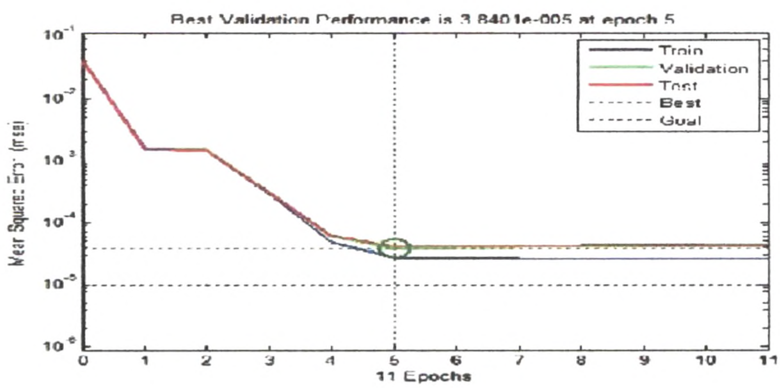

Figure 4.13: Mean Square Error

4.5.2 Design of LMNN-TCSC

Here, the stability control loop of the TCSC has been trained by artificial neural network. The gain and time constant of TCSC controller has been tuned by GA and the tuned model of the TCSC described by the equation (2.87), (2.88) and (2.89) have been used to generate the training data for the NN under the different operating condition and dynamic behavior of the power system. Procedure for selection of number of neurons in different layers of ANN-TCSC is similar to the ANN-PSS. The training pattern for the feedforward neural network is dynamic inputs $u_1(t)$ and corresponding outputs $y_1(t)$ such that $\omega_m(t), \omega_m(t-1), \omega_m(t-2), \omega_m(t-3)$ and X_{mod} respectively and targeted value of the neural network is $\hat{y_1}(t)$. The first order transfer function with 0.5 second delay has been used as communication time delay. The feedforward network has been developed with 30 neurons in first layer, 10 neurons in hidden layer and 1 neuron in output layer with hyperbolic tangent sigmoidal transfer function in first layer and hidden layer, and linear transfer functions in output layer. The optimization function can be represented by the following equation.

$$J_i(k) = \frac{1}{2}\left[y(k) - \hat{y_1}(k)\right]^2 \tag{4.15}$$

The mean square error $0.00002947 \times 10^{-5}$ has been reached after the 13 iterations through appropriate adjustment of weight and bias of neural network using Levenberg-Marquardt algorithm.

4.6 Non Linear Simulation

The nonlinear model of the power system has been used for the stability analysis of the SMIB system with generator attached PSS and transmission line connected TCSC. The initial conditions have been calculated using MATLAB programming. The non-linear simulation is carried out using non-linear dynamic model, which has been implemented using MATLAB/simulink environment. The non linear model and detail data of the power system used in this study is given in Appendix A. The comparison analysis between ANFIS and LMNN based PSS and, simultaneous application of ANFIS and ANN based TCSC - PSS are carried out under different operating condition, faults and disturbance. These disturbances are considered such as the three phase short circuit at the infinite bus, outage of transmission line, suddenly changes in mechanical input and step change in terminal voltage reference. The comparison study of intelligent techniques based TCSC and PSS has been carried out.

4.6.1 Case I

Considering operating condition 1 as defined in table 3.1 $P_t = 0.6, Q_t = 0.0224$. A three phase fault is created at 1s at the sending end of one the circuits of the transmission line and cleared after 100ms [33]. The original system restored after fault clearance. The response of speed deviation without the application of controllers and with application of intelligent controllers have been shown in Figure 3.6 and 4.14 respectively. Figures 3.6 shows that without application of controllers, the oscillation in speed deviation has been observed while using simultaneously application of ANFIS and ANN based TCSC-PSS, and individually PSS significantly diminished this oscillation in the system and provided very good damping characteristics.

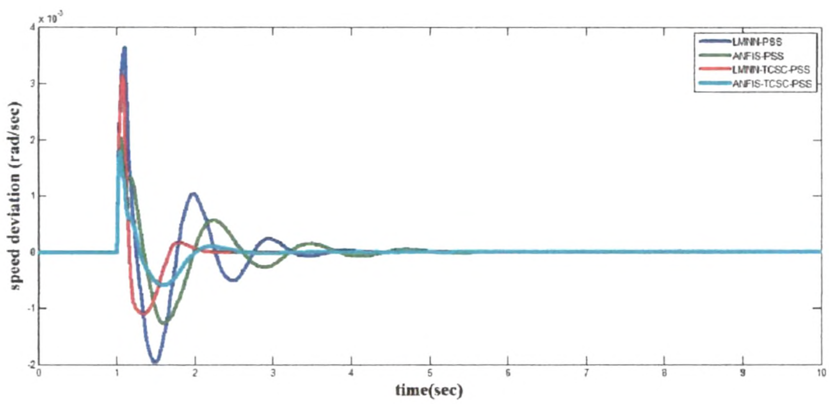

Figure 4.14: Case I: Speed response of ANFIS and LMNN based PSS-TCSC

4.6.2 case II

$P_t = 0.9, Q_t = 0.12$, Here heavy loading condition is considered. A three phase fault is created at 1s at the sending end of one of the circuits of the transmission line and cleared after 50ms.. The original system restored after the fault clearance. The response of the $\omega_m(t)$ without controller has been shown in Figure 3.22, the oscillation in the power system continuously growing with respect to the time and system has become unstable. The speed response with LMNN and ANFIS based PSS and simultaneous application of LMNN and ANFIS based TCSC-PSS has been shown in Figure 4.15. The simultaneous application of ANFIS and

ANN based TCSC-PSS, and individual applications of PSS reduced the oscillations in the system and improved stability.

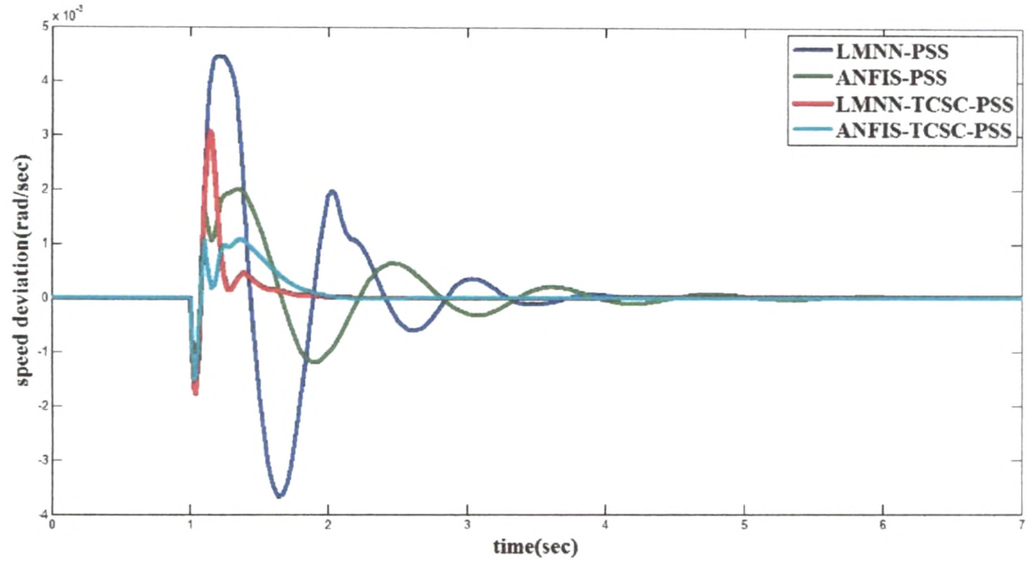

Figure 4.15: Case II: Speed response of ANFIS and LMNN based PSS-TCSC

4.6.3 Case III

$P_t = 0.9, Q_t = 0.12$,Under the heavy loading condition a 10% mechanical change applied at 1s and removed at 5 s is considered. The system lost its stability at 4s without application of controllers, which has been shown by Figure 3.26. The response of ω_m with and simultaneous application of LMNN and ANFIS based PSS-TCSC has been shown in Figure 4.16. The simultaneous application of ANFIS and ANN based TCSC-PSS, and individual applications of PSS reduced the oscillations in the system and improved stability.

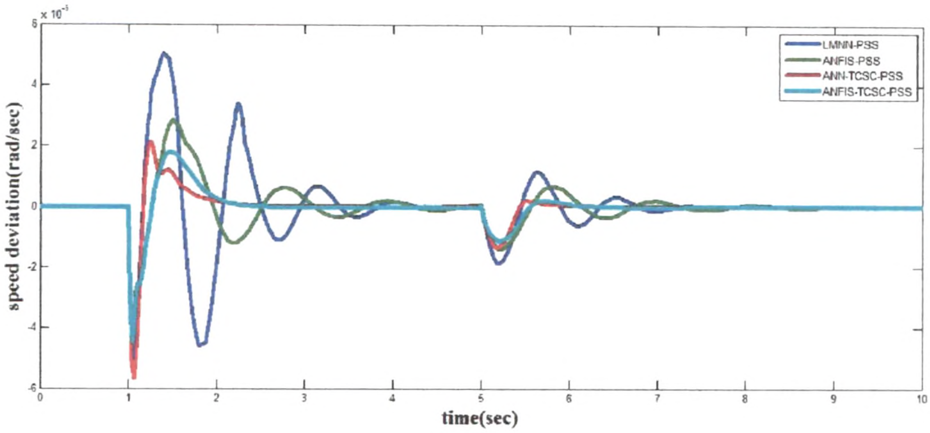

Figure 4.16: Case III: Speed response of ANFIS and LMNN based PSS-TCSC

4.6.4 Case IV

$P_t = 1.2, Q_t = 0.2$, A 0.1p.u. change in reference input voltage is applied at 1 s and removed at 5 s. The response of the ω_m without and with presence of controllers has been shown in Figure 3.30 and 4.17 respectively. Figures 4.17 shows that the simultaneous application of ANFIS-TCSC-PSS controller has improved stability of the system compared to individual application of LMNN-PSS and ANFIS-PSS.

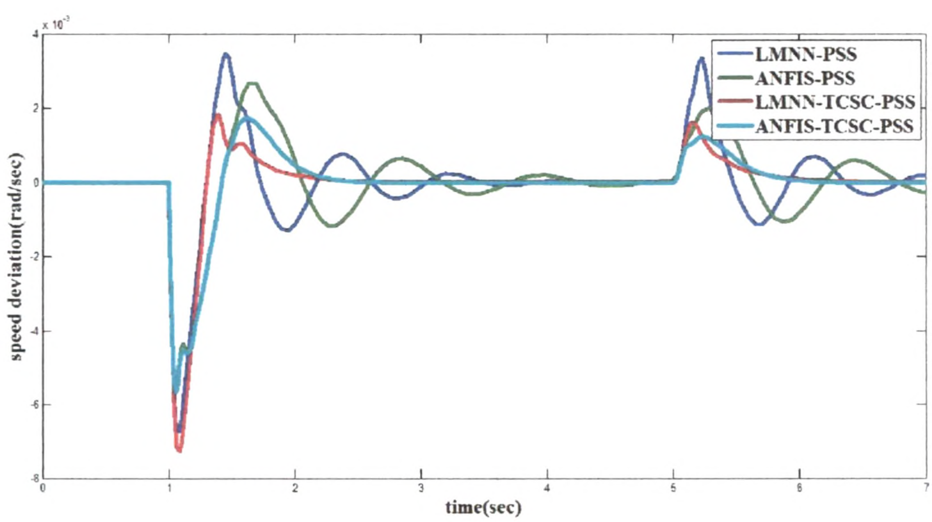

Figure 4.17: Case IV: Speed response of ANFIS and LMNN based PSS-TCSC

4.6.5 Case V

$P_t = 0.75, Q_t = 0.1$, In this case another severe disturbance is considered. One of the transmission lines is permanently tripped at 1 sec. The line reactance is significantly increased. The speed response for the above contingency has been shown in Figure 3.34 and 4.18 without and with controller respectively. Figure 4.18 shows that ANFIS based simultaneously designed TCSC and PSS have provided good stability to system compared to individual neural network based power system stabilizer.

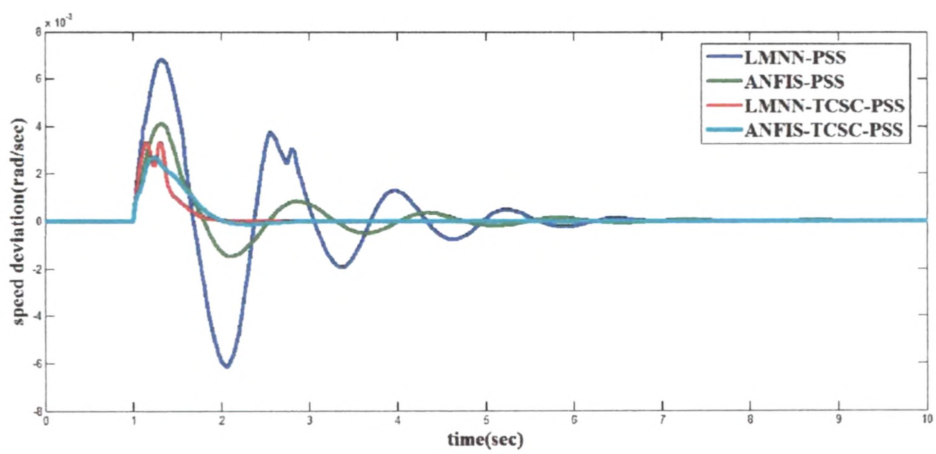

Figure 4.18: Case V: Speed response of ANFIS and LMNN based PSS-TCSC

4.7 Conclusion

In this study, the smart control strategies based TCSC damping controller and PSS have been designed. The ANFIS and LMNN based PSS, and simultaneously LMNN and ANFIS based TCSC-PSS have been applied to the dynamical power system. The non-linear simulations have been carried out for detailed analysis of the stability of the power system. The time response of speed deviation obtained by intelligent techniques based controller has been compared to the conventional power system stabilizer. Four different operating conditions are taken and the response of rotor speed deviation has been analyzed under different types of the disturbances and faults.

From the non - linear analysis,

1. Without the application of the controllers in the system, the oscillations in rotor speed deviation has been observed. Under the heavy loading condition, it has been observed that the active power and reactive power are increased; the oscillation in speed deviation is continuously growing which creates the instability of the system. The smart damping controllers have greatly diminished oscillations in system.

2. Conventional power system stabilizer does't produce satisfactory response under the different operating conditions. While simultaneous application of ANFIS and ANN based TCSC and PSS have provided very good damping characteristics compared to the individual application of PSS and almost eliminated the oscillations in system.

3. It has been observed that individual application of ANFIS-TCSC produces better response compared to the individual application of LMNN-TCSC.

4. As shown in figures, individual application of ANFIS-PSS produces better response compare to the individual LMNN-PSS.

5. Under the heavy loading condition, ANFIS based TCSC-PSS has produced good results compared to the LMNN based TCSC-PSS, and also improved the time response parameters such as settling time, rise time and delay time appreciably and decreased the overshoot in the system.