Northumbria Research Link

Farrag, M. and Putrus, G. (2011) 'An on-line training radial basis function neural network for optimum operation of the UPFC', European Transactions on Electrical Power, 21 (1), pp. 27-39.

An On-line Training Radial Basis Function Neural Network for Optimum Operation of the UPFC

M. E. A. Farrag[(1)]		G. A. Putrus[(2)]
BSc, MSc, PhD, MIET, MIEEE		BSc, MSc, PhD, CEng, MIET

(1) Institute of Energy and Sustainable Development, De Montfort University, Leicester, LE1 9BH, UK

(2) School of Engineering and Technology, Northumbria University, Newcastle upon Tyne, NE1 8ST, UK

ABSTRACT

The concept of Flexible A.C. Transmission Systems (FACTS) technology was developed to enhance the performance of electric power networks (both in steady-state and transient-state) and to make better utilization of existing power transmission facilities. The continuous improvement in power ratings and switching performance of power electronic devices together with advances in circuit design and control techniques are making this concept and devices employed in FACTS more commercially attractive. The Unified Power Flow Controller (UPFC) is one of the main FACTS devices that have a wide implication on the power transmission systems and distribution.

The purpose of this paper is to explore the use of Radial Basis Function Neural Network (RBFNN) to control the operation of the UPFC in order to improve its dynamic performance. The performance of the proposed controller compares favourably with the conventional PI and the off-line trained controller. The simple structure of the proposed controller reduces the computational requirements and emphasizes its appropriateness for on-line operation. Real-time implementation of the controller is achieved through using dSPACE ds1103 control and data acquisition board. Simulation and experimental results are presented to demonstrate the robustness of the proposed controller against changes in the transmission system operating conditions.

1. INTRODUCTION

The Unified Power Flow Controller (UPFC) is the only versatile device within the concept of Flexible A.C. Transmission Systems (FACTS) technology that can provide direct control of the three transmission system parameters: voltage, impedance and phase-angle. It can independently and dynamically control the real and reactive power flow through a transmission line while regulating the system voltage [1-3].

The UPFC consists of two voltage source converters sharing a common d.c. link, as shown in Figure 1. As can be seen, one converter (inverter 2) is connected in series with the transmission line while the other (inverter 1) is shunt connected. The former injects a series voltage which may be controlled in magnitude and phase. This voltage can be decomposed into quadrature and in-phase components (with respect to the system bus voltage) and the two components may be used to influence the flow of real and reactive power flow in the line, respectively. Inverter 1, generates the shunt compensation voltage which can also be viewed as two orthogonal components. The in-phase component is used to control the reactive power exchange with the system (to provide voltage support) whilst the quadrature component is used for regulating the UPFC d.c. link voltage. Methods to determine optimum values for the compensation voltages have been investigated [4, 5] and robustness against changes in grid conditions has always been a major concern.

The operating conditions of a real power system continuously change as a result of network and load changes or system faults and/or disturbances. It follows that a conventional controller with a set of fixed control parameters which provide an acceptable dynamic performance under certain operating conditions may no longer do so when there is a change in the system conditions, e.g. system Short Circuit Level (SCL).

Previous studies have showed that PI controllers could not always perform satisfactorily over a wide operating range and knowledge based systems using the classical fuzzy logic control have been proposed [6, 7]. However, complexities to adapt membership functions and computation requirements for defuzzification hindered its application. Recent studies turned to artificial neural networks (ANN) to achieve the desired robustness [8, 9]. A multi-layer ANN to emulate the damping control of well-tuned PI controllers under different conditions has been reported [10]. The proposed network requires a large number of neurons in the hidden layer. Hence, a Radial Basis Function Neural Network (RBFNN) was proposed which features a much simpler structure [11]. Its application in real-time control however has not been studied.

This paper presents an analysis of the dynamic performance of the UPFC when used for power flow control and voltage support. Conventional and advanced control techniques are investigated. Accordingly, due to its simple structure and good performance, an RBFNN for on-line adaptive control of a UPFC is proposed. To meet the desired performance of the UPFC, a single neuron is designated for each of the control variables. A new gradient descent algorithm is proposed for on-line tuning of the neurons. Kalman filtering could also be used, but was avoided in the present study due to the intensive computation involved. The control algorithm is simulated and experimentally implemented using a dSPACE ds1103 real time control system working in Matlab/Simulink environment. Comparative results are presented to demonstrate the performance of the proposed system.

2. THE UPFC MODEL

A single-line diagram of a simplified transmission system including a transmission line, two voltage sources and a UPFC is shown in Figure 2. The UPFC is modelled by two controllable voltage sources, V_{ser} for the series converter and V_{sh} for the shunt converter. The reactance (kX) behind the system busbar at the sending-end determines the supply Short Circuit Level (SCL). Figure 3 shows the vector diagram for the system voltages. The series voltage has two components βV_i and γV_i where β and γ represent the level of compensation relative to the nominal voltage V_i. Similarly, the shunt converter voltage has components $(1+\eta)V_i$ and ξV_i. The real and reactive power transmitted at the sending-end of the transmission line as functions of these compensation voltage components may be expressed as:

$$P = \frac{VV_j}{(1+K)X}\sin(\theta_{ij}+\theta) \qquad (1)$$

$$Q = \frac{1}{(1+K)^2 X}\left\{V^2 - VV_j\cos(\theta_{ij}+\theta\right\} - \frac{1}{(1+K)^2 X}\left\{V_j^2 - VV_j\cos(\theta_{ij}+\theta)\right\} \qquad (2)$$

where $V^2 = V_i^2 + V_{ser}^2 + 2V_iV_{ser}\cos\phi_{ser}$, $\theta = \tan^{-1}\dfrac{\gamma}{1+\beta}$ and $\phi_{ser} = \tan^{-1}\dfrac{\gamma}{\beta}$.

Similarly, the active and reactive power exchanged by the shunt converter may be expressed as:

$$P_{sh} = \frac{V_iV_{sh}}{X_{sh}}\sin\varepsilon \qquad (3)$$

$$Q_{sh} = \frac{V_i^2}{X_{sh}} - \frac{V_iV_{sh}}{X_{sh}}\cos\varepsilon \qquad (4)$$

where $\varepsilon = \tan^{-1}\dfrac{\xi}{1+\eta}$ and $V_{sh}^2 = (1+\eta)^2 V_i^2 + \xi^2 V_i^2$

3. THE UPFC CONTROL SYSTEM

3.1 Open loop control

A block diagram of an open loop control scheme for the UPFC is shown in Figure 4. Two controllers are shown; one for the series converter and the other for the shunt converter. The level of series voltage injection, represented by β and γ, is calculated from Equations 1 and 2. For the shunt converter, its quadrature component (ξV_i) is calculated by rearranging Equation 3 in order to balance, in an open loop sense, the real power flow between the two converters. The in-phase component [$(1+\eta)$ V_i] is used to adjust the exchange of reactive power with the system, as given by Equation 4.

Practically, the series converter may operate in an open loop control to provide any desired power change in the line (within system limits). However, the shunt converter must operate in a closed loop control in order to regulate the d.c. link voltage.

3.2 Closed loop control

Series converter controller

The injected series voltage components are controlled such that the real and reactive power flow in the transmission line follow their reference values. In this case, the in-phase (βV_i) and quadrature (γV_i) components are continuously updated by the controller. As given by Equations 1 and 2, the desired real power change is used to obtain the control signal γ whilst the desired reactive power change is used to determine β. For a standard PI controller, the control signals for the series converter may be expressed as follows:

$$\gamma = (P_{ref} - P)\left(K_{PP} + \frac{K_{IP}}{S}\right) \tag{5}$$

$$\beta = (Q_{ref} - Q)\left(K_{PQ} + \frac{K_{IQ}}{S}\right) \tag{6}$$

where, K_{PP}, K_{IP}, K_{PQ} and K_{IQ} are the corresponding gains of the controller.

Shunt converter controller

The shunt converter output voltage is controlled to generate or absorb a certain amount of reactive power required to maintain the system bus voltage at a set value. In addition, the shunt converter should be controlled to provide, through the d.c. link, the real power demand of the series converter. Therefore, the quadrature component of the shunt voltage (ξV_i) may be determined from the deviation of the d.c. bus voltage from its reference value. Hence, for a standard PI controller, the control signal is given as:

$$\xi = (V_{dcref} - V_{dc})\left(K_{Pdc} + \frac{K_{Idc}}{S}\right) \tag{7}$$

The control signal (η) to generate the in-phase component [$(1+\eta)\ V_i$] is driven by the deviation of the system bus voltage from its reference value. That is:

$$\eta = (V_{iref} - V_i)\left(K_{Pvi} + \frac{K_{Ivi}}{S}\right) \tag{8}$$

where $K_{Pdc}, K_{Idc}, K_{Pvi}$ and K_{Ivi} are the corresponding gains of the controllers.

4. THE RADIAL BASIS FUNCTION NEURAL NETWORK (RBFNN)

The RBFNN is a supervised neural network which may be realised in different ways. The back propagation training algorithm for the design of Muti-Layer Perceptron (MLP) neural network is widely used in load forecasting. However, for control purpose, the training process may not be acceptable as it can be very time consuming. In this paper, a different approach is described which consider the neural network design as an approximation problem in a multi-dimensional space. In this approach, the concept of the RBFNN is used where the learning process is equivalent to defining a surface that provides the best fit to the training data set and interpolate between the data. The RBFNN offers a simple structure and gives an understanding of how the learning process is achieved. As will be shown in section 6, for a real-time implementation of the RBFNN to control a UPFC, it is found that a single neuron network trained on-line by gradient descent will be suitable from the computation time and memory space point of view, as to be shown below.

An RBFNN involves three layers of different rules, as shown in Figure 5. The input layer connects the network to its environment. The second layer, the only hidden or radial basis layer in the network, applies a non-linear transformation from the input space to the hidden space. The output layer is linear, providing the response of the network to the active input. The output is described as:

$$a_2 = B_2 + W * \phi(x, \mu, B_1) \tag{9}$$

where μ and B_1 are the centre and spread of the hidden neuron; B_2 and W are the bias and weight of the linear layer. Note that, the network free parameters μ, B_1, W and B_2 need to be tuned during the learning process.

Each neuron in the radial basis layer calculates the Euclidean distance between the centre vector (μ) and the network input vector (x). The result is passed through a non-linear function; a Gaussian activation function is used in this study. As shown in Figure 6, this function is normalised, radially symmetrical around its centre and can well approximate a power integrable function. The output of the radial basis layer is determined according to how close is the input vector to the neuron centre. Thus, a radial basis network with a centre quite different from the input vector will have an output which is near to zero. In contrast, a radial basis neuron with a centre close to the input vector will produce a value near to one. The output of the hidden layer is then scaled and biased through W and B_2, respectively, to produce the network output.

In general, there are two learning paradigms for artificial neural networks. The first is *batch learning* (off-line learning) where the training data is available from analysis or previous operation. The updating action of network parameters takes place only after the whole training data set has been presented to the network. The second paradigm is *pattern learning* (on-line learning) where the network parameters are updated after each new input pattern has been presented.

For the RBFNN batch learning, there are two learning schemes depending on how the centres of the neurons are specified. The centres of the neurons in the hidden layer may be chosen randomly from the data set which covers the input space with a fixed spread. The Gaussian activation function in this case is expressed as;

$$\phi(x) = \exp\left(-\frac{m}{d_{\max}} \| x - \mu_i \|^2\right) \qquad i = 1,2...m \tag{10}$$

where m is the number of centres and $d_{\max}$ is the maximum distance between the chosen centres. The spread σ is then fixed and may be expressed as;

$$\sigma = \frac{d_{\max}}{\sqrt{m}} \tag{11}$$

The only parameters that would need to be trained in this approach are the weights and biases of the network linear layer. This may require a large training data set for determining the best centre locations and achieve a good performance. One way to overcome this limitation is to use a hybrid learning process [14].

For the RBFNN pattern learning, the network free parameters (μ, B_1, W and B_2) are updated as a result of minimising a certain cost function $\Im$, e.g. the error function given below;

$$\Im = \frac{1}{2}\sum_{j=1}^{N} e_j^2 \tag{12}$$

where N is the size of the training sample and

$$e_j = T_j - a_{2_j} \tag{13}$$

is the error signal defined as the difference between the system target and the network output for pattern j.

5. DESIGN OF THE OFF-LINE RBFNN FOR UPFC CONTROL

The network shown in Figure 5 is used to determine the compensation voltage of the series converter in the UPFC. Batch learning strategy with modified fixed centres is introduced to train the network. The learning algorithm built in MATLAB environment introduces random centres for the neurons, and then it adjusts these centres in a way to reduce the network output error. For the simple transmission system model shown in Figure 2, training data are obtained from the analytical relationship (described in section 2) between the UPFC compensation voltages and the resultant change in the real and reactive power. The data presented to the network's input layer comprises two main vectors representing the possible changes in reactive and active power values [15]. The target for the network output is represented by the desired inserted voltage components β and γ. Therefore, two neural networks are required which are trained in a similar way.

The algorithm implemented for training is shown in Figure 7, where the network iteratively creates one neuron at a time. The input pair ($\Delta Q_j, \Delta P_j$) which represent an accepted network error, is used to create a neuron. Neurons are added to the network until the Mean Squared Error (MSE) falls within an error goal margin or a maximum number of neurons has been reached (determined by the size of the input data set).

In order to minimize the error, the centres (weights) of the radial basis layer are assigned to be the transpose of the input vector. That is:

$$\mu_j = \begin{bmatrix} \Delta Q_j & \Delta P_j \end{bmatrix}^T \tag{14}$$

The spread (σ) determines the width of an area in the input space to which the neuron responds. σ is chosen to be fixed in this training algorithm in order to reduce the number of tuning parameters and network complexity. The biases are set to be:

$$B_1 = 0.8326/\sigma \tag{15}$$

Therefore, the Gaussian function output crosses 0.5 at weighted inputs of $\pm\sigma$ which reflect the closeness between the input data and the centres. The output of the radial layer for each network is described as:

$$a_{1_{\beta,\gamma}} = \phi_{\beta,\gamma}(B_{1_{\beta,\gamma}} * \mu) \tag{16}$$

The weights (W) and biases (B_2) of the output linear layer are determined in such a way to minimise the sum-squared error ($\sum_{\forall j}(T_{\beta,\gamma} - a_2)^2$) between the target values of β or γ and the network output. In order to define W and B_2, the following equation is solved in every iteration.

$$\begin{bmatrix} W & B_2 \end{bmatrix} * a_1 = T \tag{17}$$

RBFNN training parameters for the UPFC series converter application are summarized in Table 1. It is obvious that the training time for the RBFNN is relatively short. The RBFNN produces an adequate approximation of the given test data as the MSE of the training data is close to that of the test data.

Table 1 RBFNN parameters.

Network parameters	β	γ
Training data set	961	961
Test data set	441	441
Training time (sec)	6.12	5.50
Training error	2.2140e-4	2.4456e-4
Test error	2.2138e-4	2.4454e-4

6. ON-LINE RBFNN TRAINED BY GRADIENT DESCENT

As described in the previous section, the performance of the RBFNN depends on the number and centres of the radial basis functions in the hidden layer, their spread and method used for learning the input-output mapping. In this section, a simple RBFNN is designed using only one neuron to control each of the UPFC parameters. This simple network is easy to implement in a real-time control system, as the computation time is relatively short. Also, there is no need for prior training of the system data since the network parameters are dynamically adjusted every sample step. The gradient descent learning algorithm is used to update the network free parameters (μ, σ, W and B_2). The error generated from the system controlled variables (change in real and reactive power for series converter and deviation of d.c. link voltage and bus voltage for the shunt converter) are used to update the network parameters. Therefore, there is no need for training data to generate the inserted voltage components of the series and shunt converters, as it will be defined on-line at every sample step.

Figure 8 shows the designed RBFNN controller. The network output is given as:

$$y = B_2 + W\phi(x, \mu, \sigma) \quad (18)$$

The Gaussian activation function used in the hidden layer is:

$$\phi(x, \mu, \sigma) = e^{-\left(\left|\frac{x-\mu}{\sigma}\right|^2\right)} \quad (19)$$

For UPFC application, four RBFNN's are required to determine the control variables γ , β, ξ and η . The network output set can be expressed as:

$$a_2 = [\gamma \quad \beta \quad \xi \quad \eta] \quad (20)$$

The input of each network is the error signal used to derive the corresponding control variable. Referring to Equation 20, the network input set can be expressed as:

$$x = \begin{bmatrix} \Delta P & \Delta Q & \Delta P_{tr} & \Delta V \end{bmatrix} \tag{21}$$

where, $\Delta P = P_{ref} - P$, $\Delta Q = Q_{ref} - Q$, $\Delta P_{tr} = P_{ex} - P_{sh}$ and $\Delta V = V_{ref} - V$

In this algorithm each RBFNN is trained by minimising the error $E = \hat{y} - y$. Substituting for y from Equation 18:

$$E = y - (B_2 + W\phi(x, \mu, \sigma)) \tag{22}$$

The gradient of E with respect to the neural network parameters is given as:

$$\begin{aligned} \frac{\partial E^2}{\partial W} &= 2E\frac{\partial E}{\partial W} \\ \frac{\partial E^2}{\partial \mu} &= 2E\frac{\partial E}{\partial \mu} \\ \frac{\partial E^2}{\partial \sigma} &= 2E\frac{\partial E}{\partial \sigma} \end{aligned} \tag{23}$$

Note that the bias of the linear layer is assumed to be fixed in order to reduce the number of neural network parameters to be updated.

Updating the Network Weight (W)

The change of the error with respect to the network weight is given as:

$$\frac{\partial E}{\partial W} = -\phi(x, \mu, \sigma) \tag{24}$$

The increment in the weight can be described as:

$$\frac{\partial E^2}{\partial W} = \Delta W = \lambda_W E\phi(x, \mu, \sigma) \tag{25}$$

where λ_W is the learning rate of the network weight.

The updated weight value is then given as;

$$W_k = \lambda_W E\phi_k(x, \mu, \sigma) + W_{k-1} \tag{26}$$

where k represents the current time step and (k-1) represents the previous time step.

Updating the Network Centre (μ)

The change of the error with respect to the neuron centre is described as:

$$\frac{\partial E}{\partial \mu} = -W\frac{2(x-\mu)}{\sigma^2}\phi(x,\mu,\sigma) \tag{27}$$

The increment in the centre may be given as:

$$\frac{\partial E^2}{\partial \mu} = \Delta\mu = \lambda_\mu WE\frac{(x-\mu)}{\sigma^2}\phi(x,\mu,\sigma) \tag{28}$$

where λ_μ is the learning rate of the network centre.

The updated centre value is then given as;

$$\mu_k = \lambda_\mu WE\frac{(x-\mu)}{\sigma^2}\phi(x,\mu,\sigma) + \mu_{k-1} \tag{29}$$

Updating the Network Spread (σ)

The change of the error with respect to the neuron spread may be given as:

$$\frac{\partial E}{\partial \sigma} = -W\frac{2(x-\mu)^2}{\sigma^3}\phi(x,\mu,\sigma) \tag{30}$$

The increment in the spread can be described as:

$$\frac{\partial E^2}{\partial \sigma} = \Delta\sigma = \lambda_\sigma WE\frac{(x-\mu)^2}{\sigma^3}\phi(x,\mu,\sigma) \tag{31}$$

where λ_σ is the learning rate of the network spread.

The updated spread value is then given as;

$$\sigma_k = \lambda_\sigma WE\frac{(x-\mu)^2}{\sigma^3}\phi(x,\mu,\sigma) + \sigma_{k-1} \tag{32}$$

Adaptive Learning Rate

The learning rate coefficient (λ) in the gradient descent learning algorithm determines the size of the weight adjustments made at each iteration and hence influences the rate of convergence. The value of λ is important since large variations in the learning rate can result in different choices of λ. If the chosen value of λ is too large, the network response may oscillate about the steady-state value and slowly converge or diverge. If the chosen value of λ is too small, the descent progresses very slowly and the total time for convergence is increased. Defining the best value for λ depends on the system parameters and may require some trail and error.

In this study, an adaptive mechanism for choosing the learning rate is introduced by continuously monitoring the error during the training process and adjusting the value of λ_i $(i = w, \mu, \sigma)$ to best fit the local region of descent. An exponential function is used to describe the change of the learning rate with respect to the error. For the network parameters which decrease the error, the learning rate is increased as the current value is too conservative for the local minima. Conversely, when the error is too big, the learning rate is reduced to avoid overshooting of the system response by a fast decay of the past parameter history.

7. SIMULATION STUDY

The transmission system shown in Figure 2 is used to investigate the performance of the suggested RBFNN based controllers to regulate the UPFC operation for different operating conditions. Note that when the off-line trained RBFNN is applied, the series converter parameters are controlled by the neural network controller described in Section 5 whilst the shunt converter parameters are regulated using a conventional PI controller. This is due to; the series converter control variables can be extracted from their relationship with the desired changes in the power flow which is not the case for the shunt converter control variables. However, for the on-line trained RBFNN, the controller described in Section 6 is used to control both the series and shunt converters.

The response to step changes in the power reference signal is used to test the controllers' performance. Figure 9 shows the system response to a step change in the real power flow at time 0.5 s (reactive power is unchanged) and a step change in the reactive power flow which occurs at time 1.0 s (active power is constant). These results show that the real power control loop gives similar results using either off-line trained controller or on-line controller but the former produces less interaction between the real and reactive power flow. This is due to the fact that during the training phase, changes in both power components are presented to the neural network. Hence, the correct value of β and γ are determined which minimize this interaction. However, the on-line trained controller gives faster and more robust response for the reactive power control loop. The off-line trained controller response suffers from steady-state error which may reflect poor quality training (i.e. inadequate number of neuron within the trained network). This can be explained as follows: As given by equation 2, the UPFC series converter parameters (β and γ) have almost the same effect on the reactive power flow. Therefore, the desired change in reactive power will be the result of a "competition" between these two parameters. At the other hand, γ is controlled by the changes in real power where it is the dominant component in real power flow (Equation 1).

To further improve the response of the reactive power control loop, extra training data set is required, especially around the origin, this may increase the network complexity and the required allocated dynamic memory of the host computer.

8. EXPERIMENTAL STUDY

A laboratory model of a transmission system incorporating a UPFC was used in the experimental set up, as shown in Figure 10. The experimental set up contains a host computer interfaced with the UPFC inverters and transmission system through a dSPACE ds1103 data acquisition board. The control algorithm is developed in SIMULINK platform then downloaded to the ds1103 board through the real time workshop toolbox. In this work, the shunt converter is controlled to support the bus voltage and maintain the d.c. link voltage at prescribed level while the series converter is used to control the real and reactive power flow in the transmission line. Each converter is a 6-pulse PWM inverter connected to the a.c. system through an appropriate transformer. The switching frequency pattern of both converters is set to 450 Hz to avoid even and triplen harmonics. To avoid asynchronisation of the PWM, a sampling rate of 288 samples per cycle has been used (this is 32 times the frequency modulation index which is 9 in this study).

A phase shifting transformer, consisting of a three-phase series transformer and a variac, is employed to introduce a phase difference between the sending-end and receiving-end voltages of the transmission line, respectively. The phase shifting transformer injects a 90^0 leading voltage with respect to the sending-end voltage. In this study, the transmission angle is set to 6^0 in order to produce sufficient active and reactive power flow suitable for the lab tests.

Due to large memory and computation time requirements, it was not possible to implement the off-line trained controller in real-time. Therefore, only the simple structure on-line trained control algorithm has been tested in real-time, its performance compared with an equivalent conventional PI controller. Three case studies were conducted. The first examines the UPFC capability to control the power flow and support the bus voltage; the second examines the robustness of the proposed controller against changes in the system SCL; the last examines the capability of the controller to recover the UPFC operation after short circuit fault.
For the first case, a series of step changes in the main reference signals were applied in order to investigate the UPFC capability in controlling the power flow, regulating the system bus voltage and adjusting the d.c. link voltage. Figures 11 and 12 show the resultant waveforms. It is obvious that the RBFNN controller trained on-line by gradient descent gives better response than the PI

controller, with the output smoothly reaching the desired steady-state. At the same time, the RBFNN controller reduces the interaction between the controlled variables. The PI controller parameters are chosen to produce the lowest possible overshot and to reduce the interaction between real and reactive power changes.

For the second case, the source impedance is changed on-line to represent a sudden change in the system SCL. The system response when using a PI controller and RBFNN controller is compared in Figure 13. It is clear from these results that the PI controller, not being adaptive, becomes unstable under the new system operating conditions. However, the RBFNN controller is capable of accommodating system changes and generates the appropriate control signal. This is due to the on-line tuning mechanism of the network parameters (weight and bias).

For the third case, the UPFC operation is investigated post to the clearness of a three phase short circuit fault. However, investigation of the UPFC operation during the fault is beyond the scope of this study, it is clear from Figure 14 that the RBFNN controller is capable to recover the UPFC operation faster and robust than the conventional PI.

9. CONCLUSIONS

The research presented in this paper investigates the performance of the UPFC when implementing conventional and knowledge based control schemes. A single-neuron (on-line) RBFNN controller for the UPFC is developed and its performance is evaluated. The proposed control scheme employs control signals that are locally available, $\Delta P, \Delta Q, \Delta V_i$ and ΔV_{dc}. The controller has a simple architecture and hence has the potential for real-time implementation. In the experimental work conducted in this research, a ds1103 data acquisition board has been used to interface the hardware with the host computer. The parameters of the on-line trained single neuron RBFNN controller are adjusted using the gradient descent algorithm. Simulation and experimental results show that the proposed RBFNN controller for UPFC provides good performance under different operating conditions. The on-line trained closed-loop RBFNN controller produces minimum steady-state error and reduces the interaction between the system measured quantities.

10. REFERENCES

1. Hingorani, NG,'FACTS-flexible ac transmission system', IEE 5th Int. Conf. On AC and DC Power Transmission, London, Sep. 1991.
2. Gyugyi, L, 'Dynamic compensation of ac transmission lines by solid-state synchronous voltage sources' IEEE PES Summer meeting, SM 434-1 PWRD, 1993.

3. Gyugyi, L, et al, ‘The unified power flow controller: A new approach to power transmission control’, IEEE Trans. On PWRD, Vol. 10, No. 2, April 1995.
4. Round S. D., et al.,” Performance of a unified power flow controller using a d-q control system,” AC and DC Power Transmission Conference, No. 423, 1996, PP. 357-362.
5. Papič I., et al.,” Basic control of unified power flow controller,” IEEE Trans. On Power Systems, Vol. 12, No. 4, Nov. 1997, PP. 1734-1739.
6. Lombard X., Therond P. G.,” Control of unified power flow controller: comparison methods on the basis of a detailed numerical model,” IEEE Trans. On Power Systems, Vol. 12, No. 2, May 1997, PP. 824-830.
7. Vilathgamuwa M., Zhu X., Choi S. S.,” A robust control method to improve the performance of a unified power flow controller,” Electric Power Systems Research, Vol. 55, 2000, PP. 103-111.
8. Kannan S., Jayaram S., Salama M. M.,” Application of fuzzy logic for real and reactive power flow control using unified power flow controller,” 28th North America Power Symposium, Nov. 1996, PP. 437-444.
9. Farrag, M. E. A., Putrus, G.A., ‘Advanced control of the unified power flow controller’, UPEC’99, Vol. 1, PP. 74-77, 1999.
10. Damborg M. J., et al.,” Potential of artificial neural networks in power system operation,” IEEE Int. Symposium on Circuit and Systems, Vol. 4, 1990, PP. 2933-2937.
11. Zhou Q., Davidson J., Fouad A. A.,” Application of artificial neural networks in power system security and vulnerability assessment,” IEEE Trans. On Power Systems, Vol. 9, No. 1, Feb. 1994, PP. 525-532.
12. Ma T. T., Lo K. L., Tumay M.,” A robust UPFC damping control scheme using PI and ANN based adaptive controllers,” Int. Journal for Computation and Mathematics in Electrical and Electronic Engineering, Vol. 19, No. 3, 2000, PP. 878-902.
13. Dash P. K., Mishra S., Panda G.,” A radial basis function neural network controller for UPFC,” IEEE Trans. On Power Systems, Vol. 15, No. 4, Nov. 2000, PP. 1293-1299.
14. Chen S., Billings S. A., Grant P. M.,” Recursive hybrid algorithm for non-linear system identification using radial basis function networks,’ Int. Journal of Control, Vol. 55, 1992, PP. 1051-1070.
15. Farrag, M. E. A., Putrus, G. A., Ran L.,” Design of fuzzy based-rules control system for the unified power flow controller”, IECON’02, Savile, Spain, Nov. 2002.

An On-line Training Radial Basis Function Neural Network for Optimum Operation of the UPFC

M. E. A. Farrag

BSc, MSc, PhD, MIET, MIEEE

Institute of Energy and Sustainable Development, De Montfort University, Leicester, LE1 9BH, UK

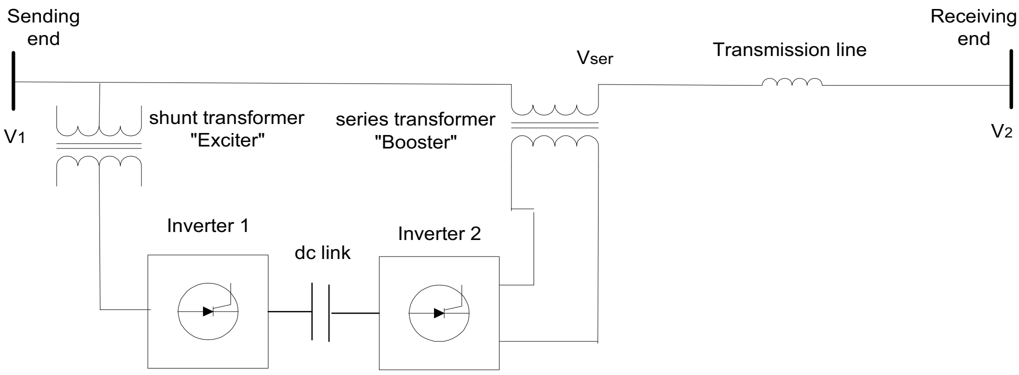


Figure 1 Schematic diagram of a transmission system, incorporating a UPFC

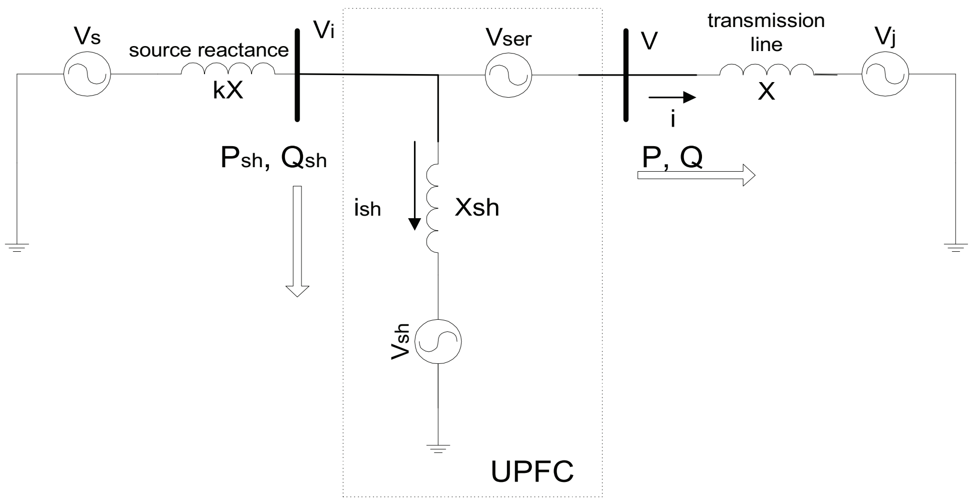
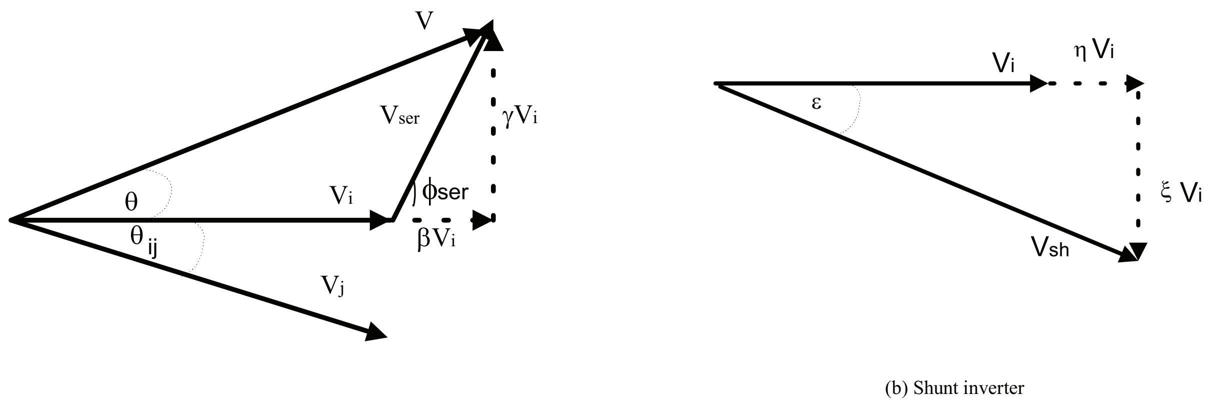


Figure 2 System model



(a) Series inverter

(b) Shunt inverter

Figure 3 Vector diagram

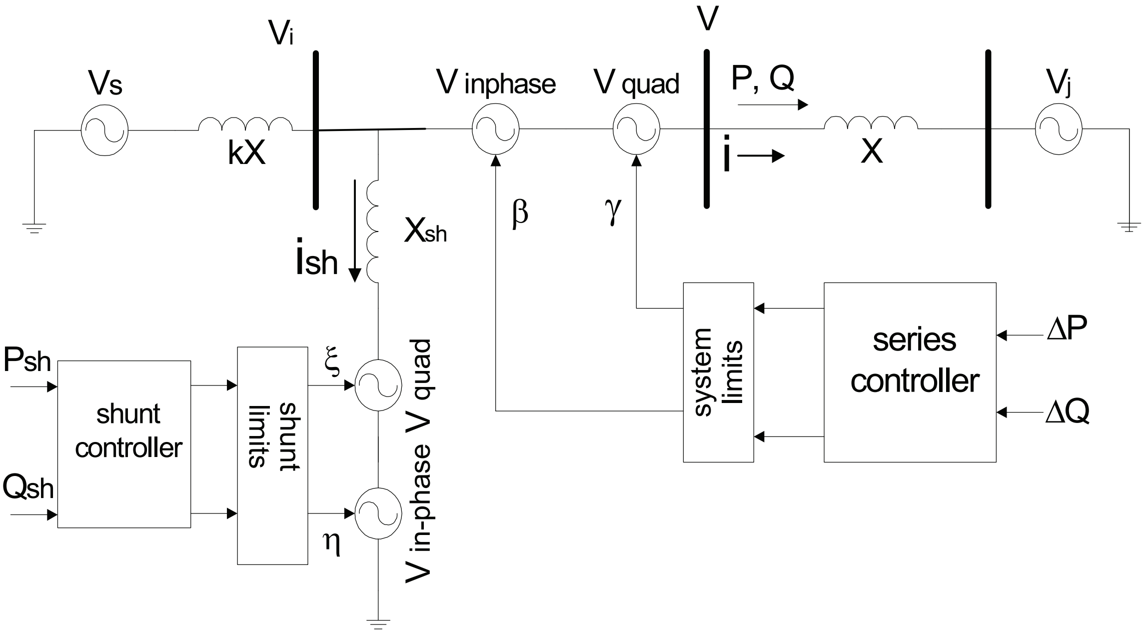


Fig. 4 Block diagram for the UPFC open-loop control system

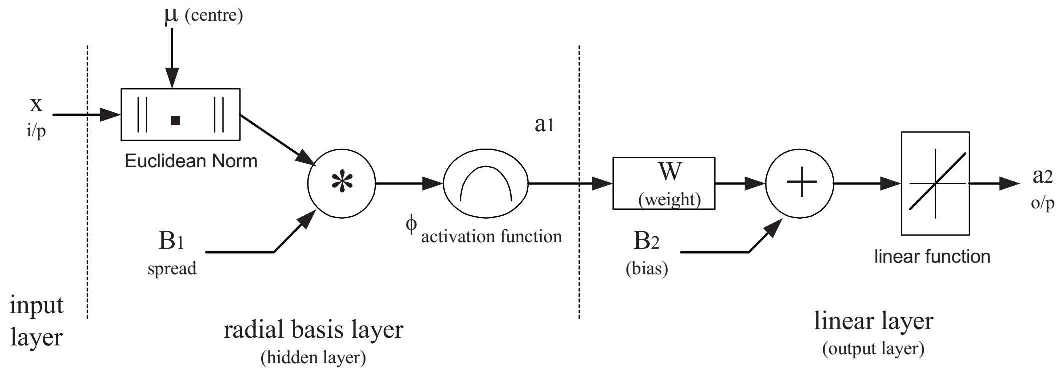


Figure 5 The RBFNN structure

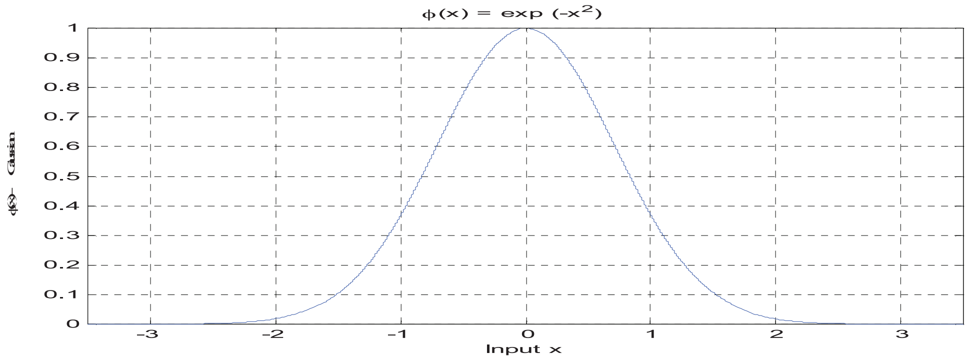


Figure 6 The Gaussian activation function

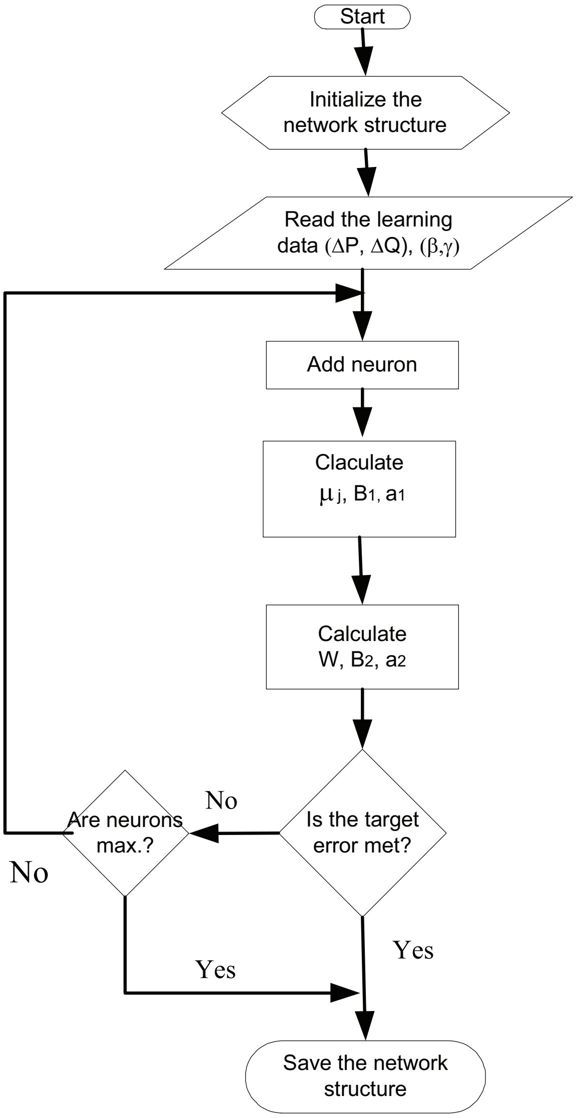


Figure 7 RBFNN learning algorithm flow chart.

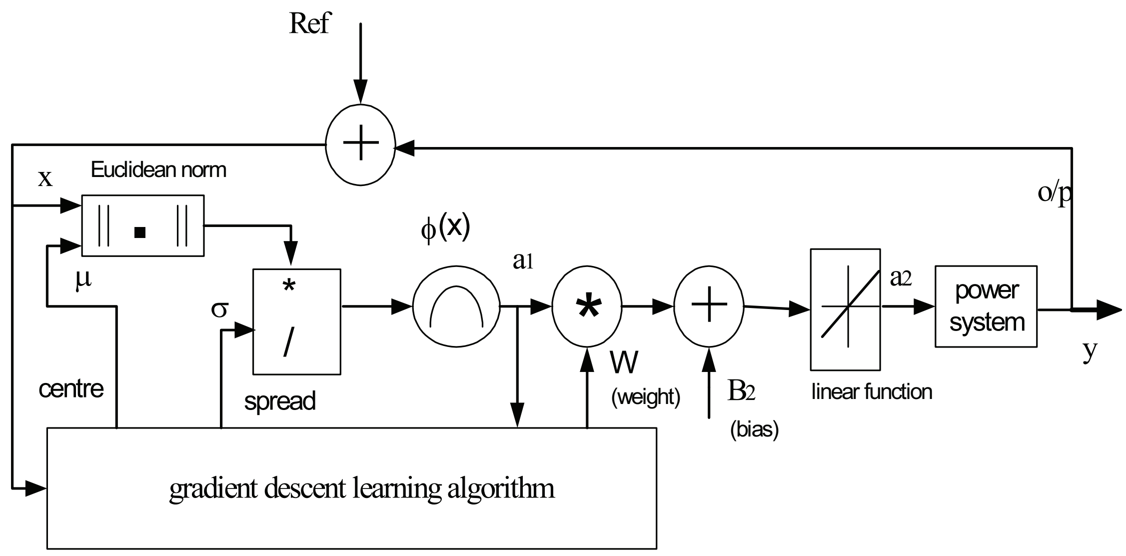
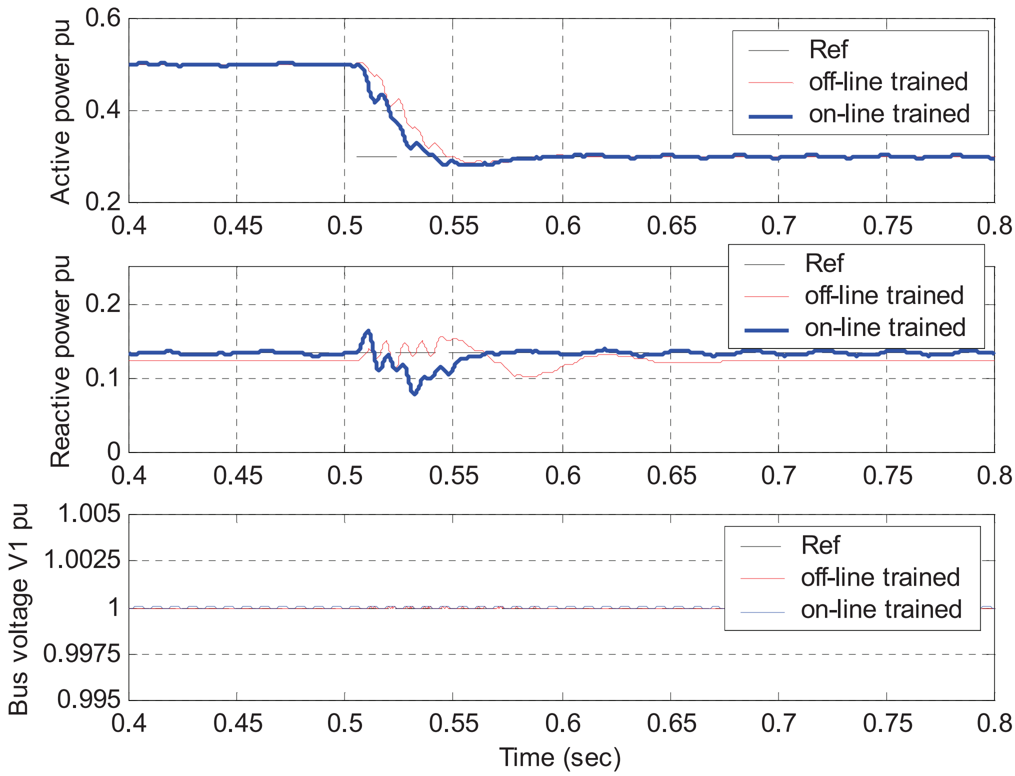
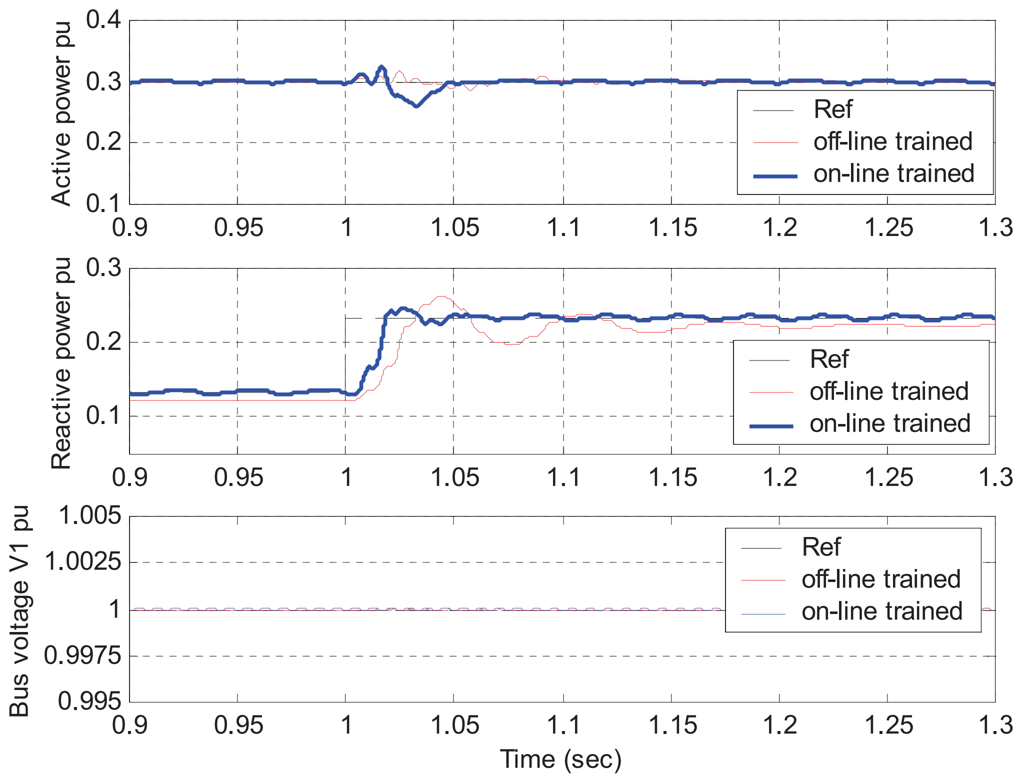


Figure 8 RBFNN trained by the gradient descent.



(a)



(b)

Figure 9 System’s response to step changes in active and reactive power.

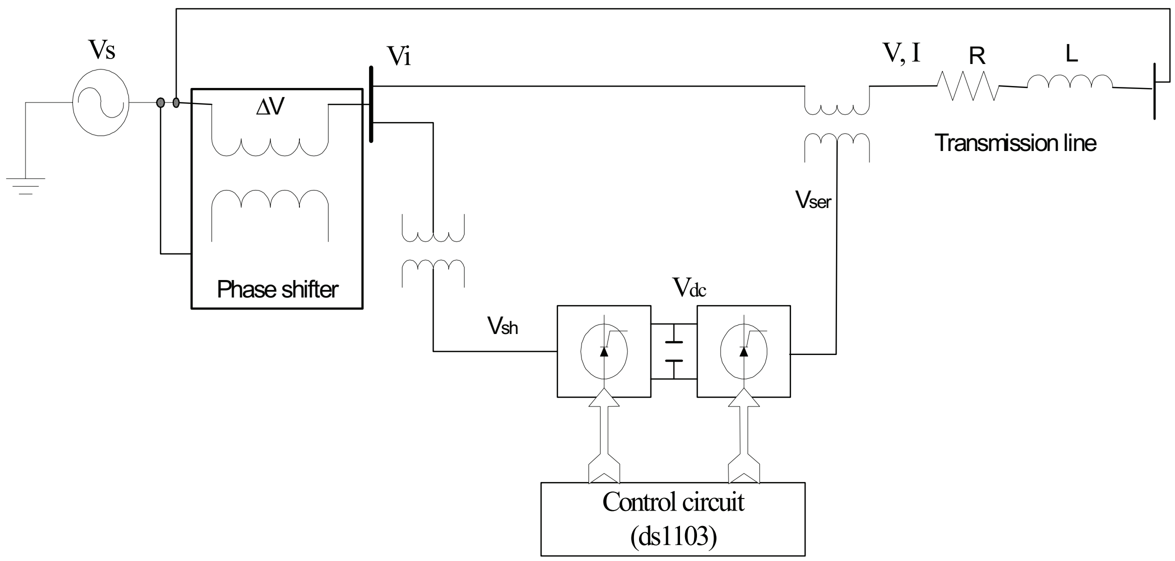


Figure 10 Block diagram of the experimental set-up.

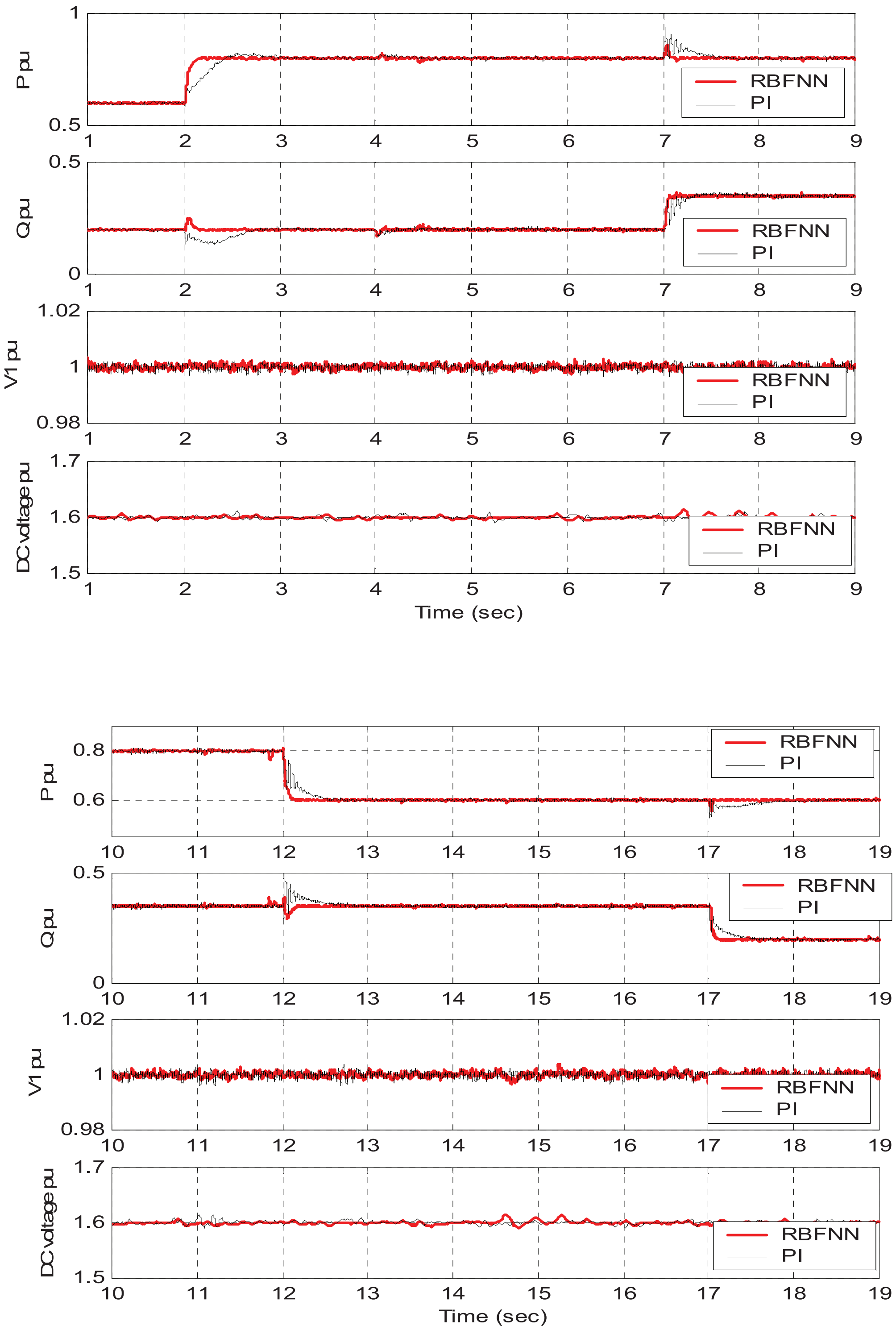


Figure 11 System's response to changes in real and reactive power.

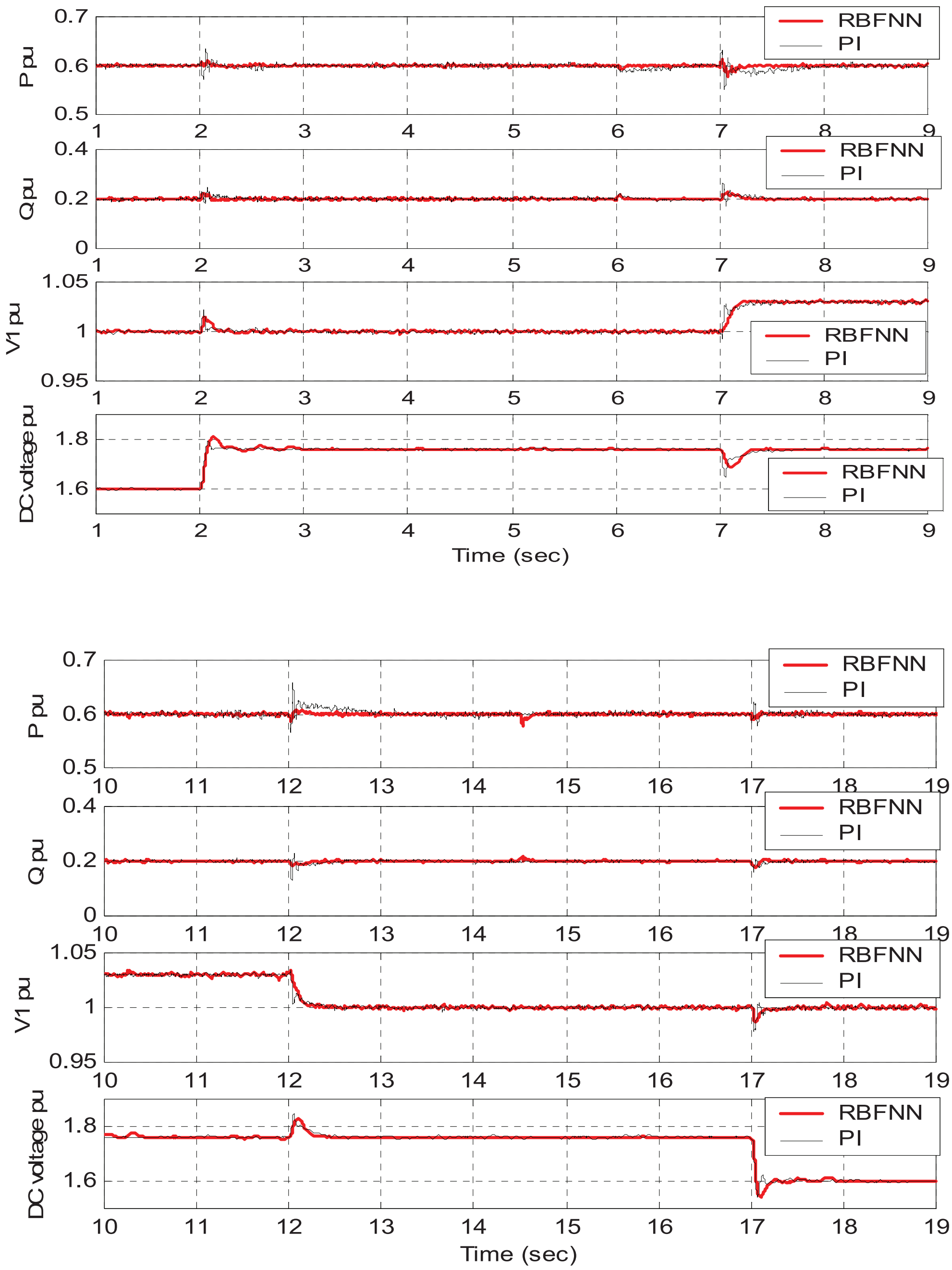


Figure 12 System's response to changes in test bus and d.c. link voltages.

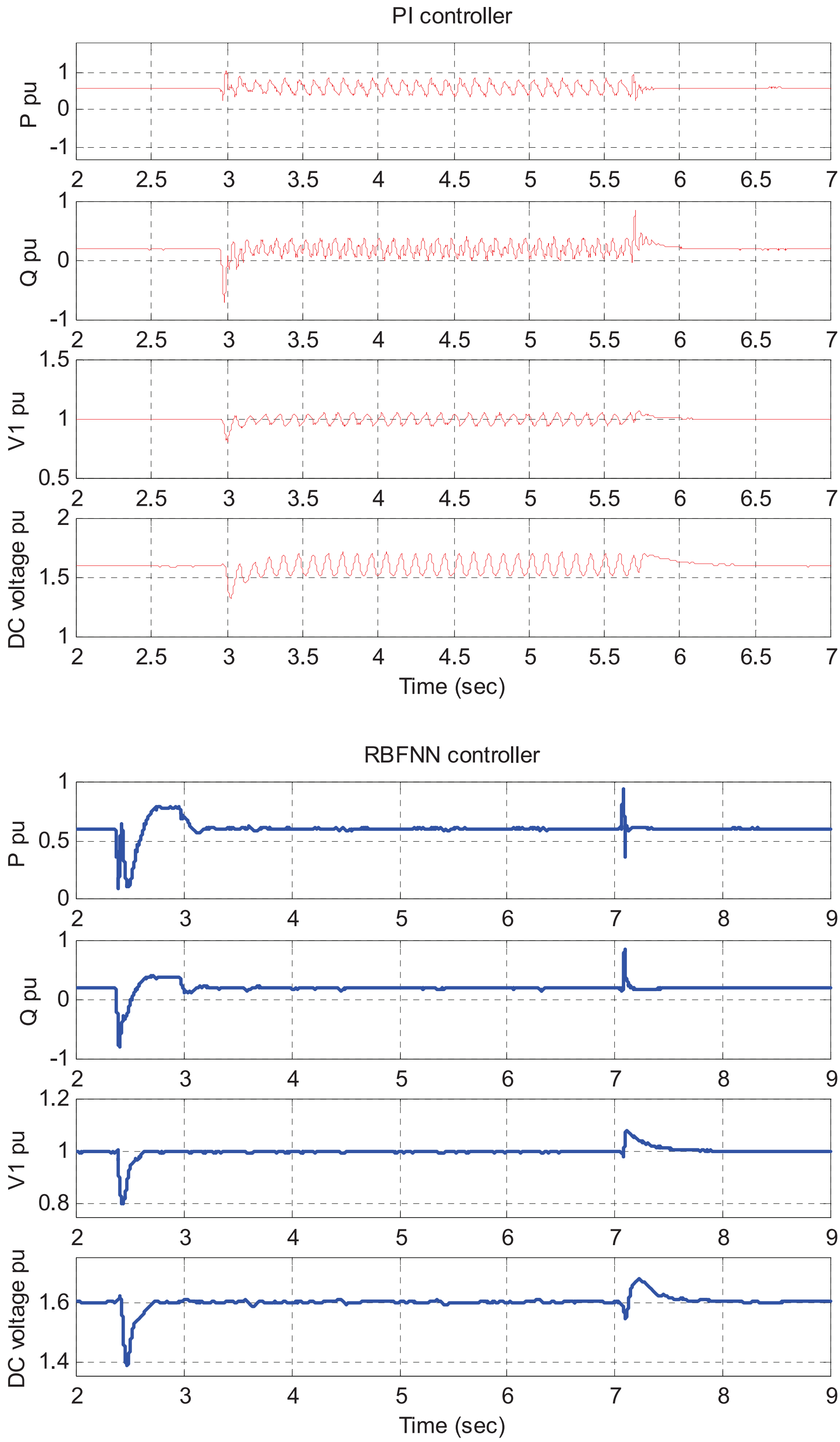


Figure 13 System's response to sudden change in the source impedance.

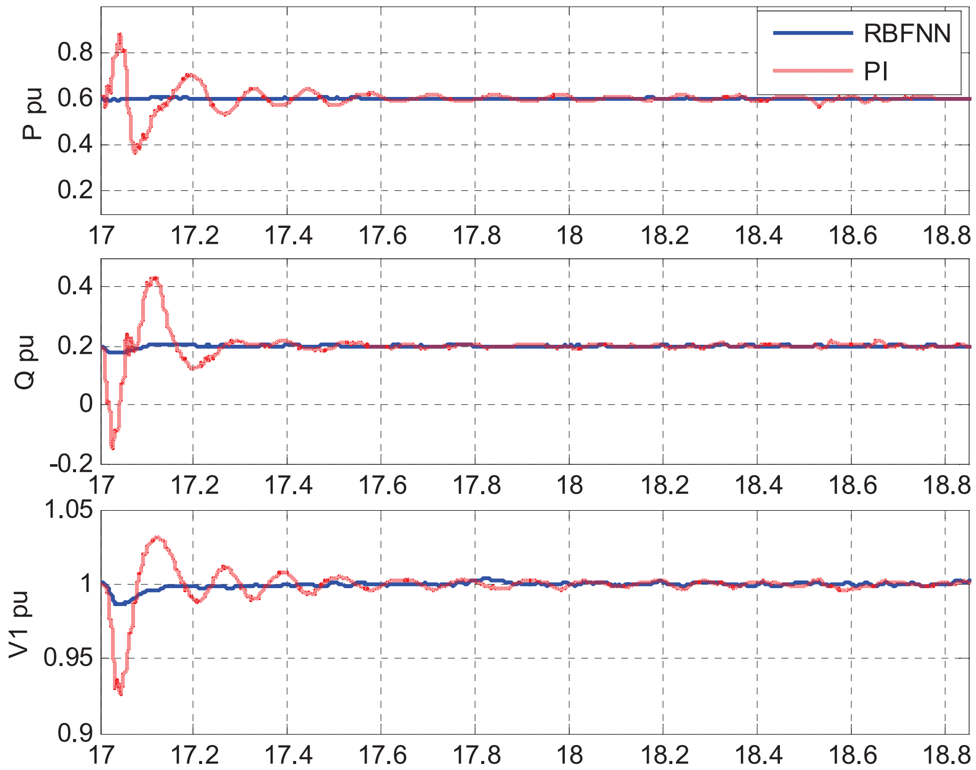


Figure 14 System’s response post to three phase short circuit fault.