Corrective action planning using RBF neural network

Daya Ram a, Laxmi Srivastava b,*, Manjaree Pandit b, Jaydev Sharma c

a Indian Railways, Bilaspur, India
b Electrical Engineering Department, MITS, Gwalior, India
c Electrical Engineering Department, IIT Roorkee, India

Received 28 April 2006; received in revised form 10 August 2006; accepted 17 October 2006
Available online 20 November 2006

Abstract

In recent years, voltage limit violation and power system load-generation imbalance, i.e., line loading limit violation have been responsible for several incidents of major network collapses leading to partial or even complete blackouts. Alleviation of line overloads is the suitable corrective action in this regard. The control action strategies to limit the line loading to the security limits are generation rescheduling/load shedding. In this paper, an approach based on radial basis function neural network (RBFN) is presented for corrective action planning to alleviate line overloading in an efficient manner. Effectiveness of the proposed method is demonstrated for overloading alleviation under different loading/contingency conditions in 6-bus system and 24-bus RTS system.

Keywords: Line loading limit; Line-overloading alleviation; Conjugate gradient method; Local optimization; Generation rescheduling; Load shedding; Radial basis function neural network

1. Introduction

As power systems have become more heavily loaded due to increased load and large interconnections, there will be an increase in cases of voltage limit violation and line loading limit violation, particularly during contingencies like line outage, etc. Electric power systems operate under the influence of various parameters, which may vary with time and circumstances. In fact, the performance of power system operation is affected by the change experienced by parameters when some major disturbance, such as loss of transmission line or generation failure or a sudden deviation in load occurs.

Under emergency conditions of line overloading, operator has to make quick decisions for corrective action without caring much for the optimality of the operating point. In this condition, a direct approach could be line-overloading alleviation with minimum number of control actions, i.e., rescheduling of generators/load shedding.

The experience of several incidents of major network collapses has concluded a strong motivation to alleviate line overloading and voltage limit violation [1], so that a secure state may be recovered. Kheddache et al. [2] presented a review of optimal load shedding techniques and the importance of relay setting for this purpose. In Ref. [3], conjugate gradient search technique has been used to minimise line overloads in conjunction with the concept of local optimization. The concept of local optimization is effective in the way that only buses in the vicinity of the overloaded line are processed for optimization process. This reduces the problem size drastically.

The control action can be performed within the security limits in a minimum time by generation rescheduling and/or load shedding, neglecting the economy consideration. These are several methods based on optimal load flow for the corrective and preventive control action along with economy and security functions on the basis of system planning. These optimization methods include a large number of variables and constraints. Hence, these methods are unsuitable for on-line implementation, due to their large computer storage and time requirements.

A number of methods like hybrid-decoupled approach, mean field theory, new dual method, etc., are developed in Refs. [4–6] for solving optimal load flow problem. A cyclic security analysis is presented for security constrained optimal power flow (SCOPF) giving the application of a new contingency screening model in Ref. [7]. An interior point quadratic-programming algorithm for solving power system optimization
### Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_i$</td>
<td>$[a_1(X_1), a_2(X_2), \ldots, a_i(X_{p_{\text{max}}})]^T$ for $i = 1, 2, \ldots, H + 1$.</td>
</tr>
<tr>
<td>$\text{AP}_{mk}$</td>
<td>Active power flow through line $mk$, from $m$ to $k$.</td>
</tr>
<tr>
<td>$\text{AP}^{\text{max}}_{mk}$</td>
<td>Active power flow through line $mk$.</td>
</tr>
<tr>
<td>$H$</td>
<td>Number of hidden layer (RBF) nodes.</td>
</tr>
<tr>
<td>$NO$</td>
<td>Number of neurons in output layer.</td>
</tr>
<tr>
<td>$O_q$</td>
<td>$[o_{q1}, o_{q2}, \ldots, o_{q_{\text{max}}}]$</td>
</tr>
<tr>
<td>$o_{q_{\text{opt}}}$</td>
<td>Output value of $q$th node in output layer for $p$th incoming pattern.</td>
</tr>
<tr>
<td>$\text{OL}$</td>
<td>Set of overloaded lines.</td>
</tr>
<tr>
<td>$P_l$</td>
<td>Real power load at $l$th bus.</td>
</tr>
<tr>
<td>$Q_l$</td>
<td>Reactive power load at $l$th bus.</td>
</tr>
<tr>
<td>$r$</td>
<td>Dimension of input vector.</td>
</tr>
<tr>
<td>$S_t$</td>
<td>Factor of safety (0.95).</td>
</tr>
<tr>
<td>$t_{q_{\text{opt}}}$</td>
<td>Target value at $q$th neuron of output layer for $p$th pattern.</td>
</tr>
<tr>
<td>$T_q$</td>
<td>$[t_{q1}, t_{q2}, \ldots, t_{q_{\text{max}}}]$.</td>
</tr>
<tr>
<td>$U$</td>
<td>Vector of the control variables available to the operator, i.e., bus power injections (generation rescheduling and load shedding).</td>
</tr>
<tr>
<td>$w_{q_{\text{opt}}}$</td>
<td>Weight between $i$th RBF unit and $q$th output node.</td>
</tr>
<tr>
<td>$w_{q_{\text{opt}}}(K)$</td>
<td>Weights connecting the hidden and output layers nodes at $K$th iteration.</td>
</tr>
<tr>
<td>$w_{q_{\text{opt}}}$</td>
<td>Biasing term at $q$th output node.</td>
</tr>
<tr>
<td>$W_{q}$</td>
<td>$W_{q1}, W_{q2}, \ldots, W_{qH}, W_{q_{\text{opt}}}$.</td>
</tr>
<tr>
<td>$x_{ji}$</td>
<td>Centre of $i$th RBF unit for input variable $j$.</td>
</tr>
<tr>
<td>$x_{jp}$</td>
<td>$j$th variable of input pattern $p$.</td>
</tr>
<tr>
<td>$X$</td>
<td>State vector of the power system consisting of bus voltage magnitudes and phase angles.</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Momentum term.</td>
</tr>
<tr>
<td>$\delta_q$</td>
<td>$[\delta_{q1}, \delta_{q2}, \ldots, \delta_{q_{\text{max}}}]$.</td>
</tr>
<tr>
<td>$\eta(K)$</td>
<td>Learning rate or adaptive size at $K$th iteration.</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>Width of $i$th RBF unit.</td>
</tr>
</tbody>
</table>

A problem with significantly less computational efforts is also presented in Ref. [8]. Mohamed et al. proposed a range of feasible corrective actions for security control, which included phase shifter control; generation rescheduling and/or load shedding to alleviate the line overload problem and transformer taps to adjust the bus voltage deviations [9]. In Ref. [10], an algorithm has been proposed for the alleviation of line overloads and voltage violations by corrective rescheduling that utilizes the decoupling of real and reactive power and the decomposition between optimization without security constraints and optimization to satisfy security constraints. Arya et al. [11] proposed an interactive line-switching algorithm for overload alleviation under line outage condition. In case, overload elimination was not possible, the overload rotation amongst two disjoint sets of overloaded lines after line switching was also suggested. A corrective switching algorithm has been proposed which relieves overloads and voltage violations as well [13]. Ioannis et al. [14] present a model and some initial results of a dynamical model for blackouts in power transmission systems. The traditional form of load control (shedding) is quite disruptive to consumers and so often avoided. In [15], a non-disruptive load control method has been developed to switch small pieces of load, so that interruptions are effectively unnoticed by consumers. These conventional methods for power system optimization and security analysis [1–15] are too slow for real-time applications in modern energy management system.

With the advent of artificial intelligence, in recent years, expert systems, pattern recognition, decision tree, neural networks and fuzzy logic methodologies have been applied to different power system problems [16–26]. Bansilal et al. [20] proposed an expert system for alleviation of network overloads using phase shifting transformers and also by rescheduling generation and/or load curtailment. For overload alleviation, fuzzy controls are suggested in Refs. [21,22], while a neural network based approach has been proposed in Refs. [23,26]. The application of artificial neural network (ANN) has shown great promise in power system engineering due to their ability to synthesize complex mappings accurately and rapidly.

Artificial neural network is the functional imitation of a human brain which simulates the human intuition in making decisions and drawing conclusions even when presented with complex, noisy, irrelevant/partial information. The information going to the input layer neurons (units) of artificial neural network is recoded into an internal representation and the outputs are generated by the internal representation rather than by the input pattern. It can model any non-linear function without knowledge of the actual model structure and during testing phase it gives the result in very short time.

A neural network consists of a number of neurons, which are the elementary processing units that are connected together according to a certain pattern of connectivity. The development of artificial neural network involves into two phases, training or learning phase and testing phase. Developing a neural network is unlike developing software, because the network is trained, not programmed. Most of the published work in power system area utilizes multi-layer perceptron (MLP) model based on back propagation (BP) algorithm, which usually suffers from local minima and over fitting problems.

In this paper, an approach based on radial basis neural network [26–28] is proposed to alleviate line overloading under different loading/contingency conditions in an efficient manner. The RBFN has many advantageous features such as optimised system complexity, minimised learning and recall times. RBF model has an input layer, one hidden layer and output layer [33]. The input variables are directly fed to the hidden units without weights. The approach developed here is successfully applied to 6-bus system and 24-bus RTS system.

### 2. Methodology

The general block diagram of the work carried out in this paper is presented in Fig. 1. A large number of patterns are...
2.1. Alleviation of line overloads

The line overload alleviation problem can be formulated as an optimization problem [3]:

Minimize \( f(x) = \sum_{m_k \in \text{OL}} (\Delta P_{mk} - \Delta P^\text{max}_{mk} S_i)^2 \) \hspace{1cm} (1)

Subject to:

\[ g(X, U) = 0 \] \hspace{1cm} (2)

\[ h(X, U) \leq 0 \] \hspace{1cm} (3)

\[ U^\text{min} \leq U \leq U^\text{max} \] \hspace{1cm} (4)

Eq. (1) is a sum of the square of line overloads; Eq. (2) represents the usual load-flow equations; inequality (3) represents the system operating constraints, i.e., line loading and bus voltages, and inequality (4) represents the limitations on the control variables. The optimization problem defined in Eqs. (1)–(4) is a large-scale non-linear problem. Considering the following assumptions and simplifications the problem is reduced to a manageable size.

(i) Most of constraints (3) are ineffective in the final solution and may be excluded from the solution procedure as practical power systems are not designed to operate with many line flows or voltages near limits.

(ii) Line flow in practical power systems with heavily loaded lines is prominently active power flow that depends on the phase angles. Hence, constant voltage magnitude is assumed, and Eq. (2) is solved for phase angles only.

(iii) The buses for local optimization are selected on the basis of pre-specified step length, usually 1–2-step length, is enough for local optimization. This reduces the problem size drastically.

With the above assumptions/simplifications, the reduced problem for line overload alleviation can be presented into the following form [3]:

\[
\text{Minimize } f(\theta) = \sum_{m_k \in \text{OL}} (\Delta P_{mk} - \Delta P^\text{max}_{mk} S_i)^2
\] \hspace{1cm} (5)

where

\[
\Delta P_{mk} = G_{mk} V_m^2 - V_m V_k [G_{mk} \cos(\theta_m - \theta_k) + B_{mk} \sin(\theta_m - \theta_k)]
\] \hspace{1cm} (6)

Subject to, the net bus power equal to zero, i.e.,

\[
g_i(\theta; P) = P_i - \sum_{j \in B_i} G_{ij} V_j^2 - V_i V_j [G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)] = 0
\] \hspace{1cm} (7)

and

\[
p^\text{min}_i \leq P_i \leq p^\text{max}_i
\] \hspace{1cm} (8)

where \( i \in \text{OL} \) are the buses to be processed for local optimization except slack bus, \( B_i \) is the set of buses connected to \( i \) and \( P_i \) is bus power injection at bus \( i \).

The problem defined in Eq. (5) is solved by using conjugate gradient method [3] with \( P \) as independent variable and \( \theta \) as dependent variables subject to the constraints given (7) and (8). Load shedding is carried out on the receiving-end buses of the overloaded lines. The amount of load shedding on these buses can be controlled by adjusting the limits on \( P_i \), i.e., \( P^\text{min}_i \) and \( P^\text{max}_i \).

2.2. Training set generation

Two approaches have been used for training set generation. These are: (i) a number of loading patterns are generated by changing load at each bus randomly and overloads are induced by creating single line outage contingency cases and (ii) a large number of loading patterns are generated by changing load at each bus randomly in wide range to create overloading of some of the lines in the system.

Corrective action in the form of generation rescheduling and load shedding in the vicinity of the overloaded lines has been implemented by applying conventional local optimization technique as discussed in Section 2.1 to obtain a new operating point that is free from any overloading. Thus, for each load scenario, the value of modified loads and rescheduled generation is obtained, which belongs to normal loading of the system. The original value of loads along with modified values of loads and slack bus generation form one training set.
pattern, i.e., input–output set to be used for training/testing of the proposed neural network.

Let $P_1, P_2, ..., P_n, Q_1, Q_2, ..., Q_n$ be representing a load scenario with overloading of one or more number of lines. By applying local optimization technique, the overloading of different lines is alleviated implementing generation rescheduling and/or load shedding in the vicinity of the overloaded lines. As line overloading is alleviated, let the new values of loads be $P_1', P_2', ..., P_n'$ and the slack bus power generation be $P_{g1}'$.

In the developed RBF neural network, $P_1, P_2, ..., P_n, Q_1, Q_2, ..., Q_n$ are taken as inputs while $P_1', P_2', ..., P_n'$ and $P_{g1}'$ are considered as output, thus making a training pattern (training pair) for the RBF neural network. A large number of such training patterns are used for optimal training of the proposed neural network.

During testing phase, for an unseen load scenario having overloading of some lines in the system, the trained RBFN model immediately gives the modified values of loads and generation, i.e., $P_1', P_2', ..., P_n'$ and $P_{g1}'$ leading to alleviation of overloading in the system. Once the outputs are known, the amount of load shedding $(P_1 - P_1'), (P_2 - P_2'), ..., (P_n - P_n')$ can be calculated. Thus, the trained RBF model is able to provide the necessary corrective action in terms of load shedding/generation scheduling.

In the development of the RBF network, the size of the input layer has been determined by the size of the input pattern. Similarly, the size of the output layer has been determined by the number of outputs, i.e., each output neuron was assigned to corrected value of real load/generation at slack bus. The size of the hidden neurons was selected to provide the best test results for the given system. It was found that 48 hidden nodes were adequate for 6-bus system, while 77 hidden neurons were adequate for 24-bus RTS system.

It has been observed that the training error decreases along with the number of iterations, while the testing error decreases at first, bounces around, and then starts to increase as shown in Fig. 2. The optimal learning and generalization are achieved close to the global minimum of the testing error [32]. During the training, the neural network is closely monitored to prevent network memorization (over training). When a neural network memorizes the training data, it reproduces acceptable results for patterns that have been used during the training, but unacceptable results with high errors when tested on unseen patterns.

At optimal learning point, the testing error produced by the trained neural network would be minimum, which is always desired for any application of a neural network. So, in this paper, training of the RBF neural networks is continued for a number of iterations until the minimum rms error is obtained for testing patterns.

2.3. Radial basis function neural network

The RBF network consists of three layers, the input layer, hidden layer and output layer. The nodes within each layer are fully connected to the previous layer as shown in Fig. 3. The input variables are assigned to each node in the input layer and are passed directly to the hidden layer without weights. The hidden nodes (units) contain the radial basis functions, and are analogous to the sigmoidal function commonly used in feed-forward multi-layer perceptron model.

The radial basis function is similar to the Gaussian density function, which is defined by a centre position and a width parameter. The width of the RBF unit controls the rate of decrease of function. The output of the $i$th unit, $a_i(X_p)$ in the
hidden layer is given by

\[ a_i(X_p) = \exp\left( -\sum_{j=1}^{r} \frac{(x_{jp} - \bar{x}_j)^2}{\sigma^2} \right) \]  

(9)

The connection between the hidden units and the output units are weighted sums as shown in Fig. 3. The output value \( o_{qp} \) of the \( q \)th output node is given as

\[ o_{qp} = \sum_{i=1}^{H} w_{qi} a_i(X_p) + w_{qo} \]  

(10)

The parameters of the RBF units are determined in three steps of the training process. First, some form of clustering algorithm determines the unit centres. Then the widths are determined by a nearest neighbor method. Finally weights connecting the RBF units and the output units are calculated using multiple regression techniques. Euclidean distance based clustering [29] technique has been employed in this paper to select the number of hidden (RBF) units and unit centres. The normalized input data are used for training of the RBF neural network.

### 2.4. Normalization of data

Normalization of the data is an important aspect for training of the neural network. Without normalization, the higher valued input variables may tend to suppress the influence of smaller ones. To overcome this problem neural networks are trained with normalized input data. The variable having highest value is assigned a value equal to 0.9 and that having lowest value is assigned 0.1. The normalized value \( x_n \) presented to the neural network as input is calculated using the equation

\[ x_n = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \times 0.8 + 0.1 \]  

(11)

where \( x, x_{\text{max}} \) and \( x_{\text{min}} \) are the actual, maximum and minimum values of input variable \( x \).

### 2.5. Algorithm

The solution algorithm for corrective action planning using RBFN is given below.

(i) A large number of load patterns are generated randomly in wide range of load variation at each bus and single line outage is simulated.
(ii) For each case conjugate gradient technique is applied for alleviating line-overload using the concept of local optimization and the corrected values of loads at different buses and generation at slack bus is computed.
(iii) Real power and reactive loads at different buses are selected as input features for the RBF network.
(iv) Input data are normalised between 0.1 and 0.9.
(v) The number of hidden (RBF) units and unit centres are determined using Euclidean distance based clustering technique. Then width of the RBF unit is determined.

(vi) For training of the RBF network, initialize all the connection weights between hidden nodes and output nodes.
(vii) Compute the Gaussian function at the hidden node using Eq. (9).
(viii) Calculate the output of the RBF network using Eq. (10).
(ix) Calculate the mean squared error \( e_p \) for the \( p \)th pattern using

\[ e_p = \frac{1}{2} \sum_{q=1}^{\text{NO}} (t_{qp} - o_{qp})^2 \]  

(12)

(x) Repeat steps (vii)–(ix) for all the training patterns.
(xi) Calculate the error function \( E_K \) using

\[ E_K = \sum_{p=1}^{\text{NO}} e_p = \frac{1}{2} \sum_{p=1}^{\text{NO}} \sum_{q=1}^{\text{NO}} (t_{qp} - o_{qp})^2 \]  

(13)

where \( p_{\text{max}} \) is the total number of training patterns,
(xii) Update the connection weights using equations

\[ w_{qi}(K+1) = w_{qi}(K) + \Delta w_{qi}(K) \]  

(14)

where \( \Delta w_{qi}(K) = \eta(K) \sum_{p=1}^{p_{\text{max}}} \delta_{qp} A_{pi} + \alpha \Delta w_{qi}(K-1) \)
(xiii) The procedure is continued till the error becomes negligible.

### 3. Simulation results

The proposed method has been implemented on a 6-bus system [30] and for 24-bus RTS system [3]. Overloading in sample systems has been created by simulating single-line outage conditions in 6-bus system under varying load condition, and by heavily loading the 24-bus sample system.

#### 3.1. 6-Bus system

For 6-bus system, 100 load scenarios were generated randomly for the load variation of \( \pm 30\% \) of the base case. For each loading condition, single line outages were simulated and conjugate gradient technique was applied for alleviating line-overload, and thus, the corrected values of real load and generation at different buses are computed. In 6-bus system, the solution could be obtained for 5 single-line outage cases. Thus, 500 (5 \( \times \) 100) load patterns were generated. Out of these 500 patterns, 400 patterns corresponding to 80 load scenarios were arbitrarily selected for training the RBF neural network. Once the RBFN was trained, it was tested for the remaining 100 test patterns corresponding to 20 load scenarios.

The non-zero (total 6 nos.) real and reactive power loads at different buses were selected as input features to the RBFN. There are four output nodes in the proposed RBFN representing the corrected values of real loads at three buses and generation at slack bus after overloading has been alleviated. Since, the same RBFN was developed for simulating all the five line outage cases, a string of three bi-polar digits (+1, -1) was also
applied (resulting into nine inputs) as input to the RBFN. The RBFN (9–48–4) was trained to predict real loads at the three buses and generation at slack bus such that there is no line overloaded in the power system.

The trained RBFN was tested for 100 unknown patterns corresponding to 20 load scenarios. Test results are compared with those obtained by using the conventional method, i.e., conjugate gradient method for overloading alleviation to show the effectiveness of the proposed method. Table 1 compares the loads and generation estimated, only for a particular load scenario (five test patterns) by RBFN following all the five single-line outage cases, where maximum and minimum errors are 0.0034 pu and 0.0002 pu, respectively. The rms error for all the 100 testing patterns was 0.0017 pu while maximum error was 0.0045 pu.

Under outage of line no. 2 in base case of the 6-bus system, Line no. 1, 4 and 5 get overloaded. This overloading could be removed by modifying the real loads at bus nos. 3, 5 and 6, from 0.275, 0.15 and 0.25 pu to 0.1123, 0.0523 and 0.1108 pu, respectively, while the slack bus generation is 0.1402 pu. For this training pair, the inputs are taken as 0.275, 0.15, 0.25, 0.065, 0.090, 0.025 and a string of three bipolar digits (−1,+1,−1) to represent the line outage condition (total 9 no. of inputs), while the four outputs will be 0.1123, 0.0521, 0.1108 and 0.1402. A properly trained RBFN will be able to provide these outputs accurately. Once the outputs are known, the amount of load shedding can be calculated as 0.1627, 0.0979, and 0.1402 at the three buses, respectively. Hence, the corrective action can be planned and taken immediately without solving the time-consuming optimization problem of overload alleviation.

3.2. 24-Bus (RTS) system

The proposed method has been tested on 24-bus system [3]. The 24 bus system has, 34 lines, 17 PQ buses and 7 PV buses including slack bus. As many as 250 training/testing patterns were generated by changing the load at each bus randomly for the load variation of ±50% of the base case and under each condition, conjugate gradient technique is applied for alleviating overload limit violations using the concept of local optimization, and thus the corrected values of real load and generation at different buses are computed. Out of 250 patterns, 200 patterns are selected randomly for training and remaining 50 for testing of the RBFN.

In this case, better results were obtained when only the real power loads at different buses (total 17 in nos.) were selected as input features to the RBFN. The proposed RBF model has one input layer of 17 neurons; one hidden layer of 77 neurons (optimum no. of clusters or hidden nodes) and an output layer of 18 neurons representing the modified values of real power loads at different buses and generation at slack bus such that overloading in system is alleviated.

It was observed that a multi-output neural network performs well only in the cases where outputs are almost of similar values. In cases where outputs are spread in wide range for the same input, the performance of artificial neural network was not satisfactory [31]. Therefore, similar type of outputs is grouped in the same cluster [29] and for each cluster a separate RBF neural network was developed and trained. Excellent performance was observed when three RBF models having structures (17–77–8), (17–77–7) and (17–77–3) were developed. A bias neuron is also added in hidden layer to speed up the training.

Once trained, the RBFNs were tested for 50 unknown load scenarios. Test results are compared with those obtained by using the conventional method, i.e., conjugate gradient method based on local optimization for overloading alleviation to show the effectiveness of the proposed method. Tables 2–4 compare the loads and generation estimated only for one load scenario providing maximum error by RBFN1, RBFN2 and RBFN3, respectively due to limited space. The RBFN1 provides maximum absolute error of 0.0021 pu for pattern no. 28, while the rms error for all the 50 testing patterns is 0.0043 pu. The RBFN2 provides maximum absolute error 0.0131 pu for pattern no. 45, with the rms error for all the testing patterns as 0.0076 pu. The RBFN3 provides maximum absolute error 0.011 pu for pattern no. 6, with the rms error of 0.0045 pu for all

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Comparison of results for 6-bus system</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO</td>
<td>Load/Gen</td>
</tr>
<tr>
<td>1</td>
<td>P1</td>
</tr>
<tr>
<td>2</td>
<td>P1</td>
</tr>
<tr>
<td>3</td>
<td>P1</td>
</tr>
<tr>
<td>4</td>
<td>P1</td>
</tr>
<tr>
<td>5</td>
<td>P1</td>
</tr>
</tbody>
</table>

LO = line outage, Load/Gen = modified load/generation with no overloading.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Testing performance of RBFN1 for pattern no. 28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load/Gen</td>
<td>Target</td>
</tr>
<tr>
<td>P1</td>
<td>1.619</td>
</tr>
<tr>
<td>P2</td>
<td>1.383</td>
</tr>
<tr>
<td>P3</td>
<td>1.862</td>
</tr>
<tr>
<td>P4</td>
<td>1.463</td>
</tr>
<tr>
<td>P5</td>
<td>2.028</td>
</tr>
<tr>
<td>P6</td>
<td>1.531</td>
</tr>
<tr>
<td>P7</td>
<td>1.731</td>
</tr>
<tr>
<td>P8</td>
<td>0.14</td>
</tr>
</tbody>
</table>
the 50 testing patterns. The testing performances (pu error) of the three RBFNs for all the testing patterns are shown in Figs. 4–6, respectively. As can be observed from these figures, the trained RBF neural networks provide satisfactory results for different testing patterns.

Performance of the proposed RBF neural network was compared with that of a three-layered feed-forward ANN [28] by estimating the corrected values of loads and generation at slack bus for the 24-bus system on Pentium IV, 1.5 GHz. The three-layered ANN (17-77-3) has the same number of neurons and interconnected weights as the developed RBF model (RBFN3) contains. The same initial weights, learning rate $\eta$ and momentum $\alpha$ were selected for training of the two models. The two neural network models were trained up to the optimal learning and generalization of the neural network.

As can be observed from Table 5, the error reduction properties of the RBF model are far better than those of the BP

<table>
<thead>
<tr>
<th>Load/Gen</th>
<th>Target</th>
<th>RBFN1</th>
<th>Error</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>P9</td>
<td>0.623</td>
<td>0.6211</td>
<td>0.0019</td>
<td>0.305</td>
</tr>
<tr>
<td>P10</td>
<td>0.83</td>
<td>0.8308</td>
<td>−0.0008</td>
<td>0.096</td>
</tr>
<tr>
<td>P11</td>
<td>1.268</td>
<td>1.2716</td>
<td>−0.0036</td>
<td>0.284</td>
</tr>
<tr>
<td>P12</td>
<td>0.904</td>
<td>0.897</td>
<td>0.007</td>
<td>0.774</td>
</tr>
<tr>
<td>P13</td>
<td>1.05</td>
<td>1.0535</td>
<td>−0.0035</td>
<td>0.333</td>
</tr>
<tr>
<td>P14</td>
<td>0.81</td>
<td>0.8194</td>
<td>−0.0094</td>
<td>1.161</td>
</tr>
<tr>
<td>P15</td>
<td>0.921</td>
<td>0.9079</td>
<td>0.0131</td>
<td>1.422</td>
</tr>
</tbody>
</table>

Table 3
Testing performance of RBFN2 for pattern no. 45

<table>
<thead>
<tr>
<th>Load/Gen</th>
<th>Target</th>
<th>RBFN1</th>
<th>Error</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>P16</td>
<td>3.834</td>
<td>3.8255</td>
<td>0.0085</td>
<td>0.222</td>
</tr>
<tr>
<td>P17</td>
<td>2.695</td>
<td>2.6987</td>
<td>−0.0037</td>
<td>0.137</td>
</tr>
<tr>
<td>G1</td>
<td>2.159</td>
<td>2.17</td>
<td>−0.011</td>
<td>0.509</td>
</tr>
</tbody>
</table>

Table 4
Testing performance of RBFN3 for pattern no. 6
network. The RBF neural network achieves the point of optimal training in 574 iterations, while the BP model achieves it in approximately 3000 iterations. The optimal training time required by RBF neural network is almost 6% of that required by MLP model. In other words, the learning of RBF network is about 17 times faster as compared with the three-layered ANN. Also the rms error provided by the RBF Model is considerably less as compared to that obtained by the BP model.

4. Conclusion

A radial basis function neural network based technique for corrective action planning is developed in this paper. Once trained, the proposed RBFNs provide corrective action planning for unknown patterns in terms of load shedding/generation rescheduling almost instantaneously and accurately, in order to achieve a new secure point.

The RBF based technique has been applied to only two test systems under single line outage conditions and under heavily loaded conditions. This method is applicable to practical power systems having a large number of buses and lines under generator outage, single/multiple line outage conditions and under heavily loaded condition as well. Since the approach is very fast it can prove to be more suitable for on-line applications as compared to conventional optimization techniques.

Acknowledgements

The authors would like to acknowledge the financial support of University Grant Commission (UGC), New Delhi, India, under Research grant no. F 14-16/2003 dated 27/03/03 and thank the Director, MITS, Gwalior, India to carry out this work.

References


Table 5

<table>
<thead>
<tr>
<th>S. no.</th>
<th>ANN model</th>
<th>Optimal training</th>
<th>Testing error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Iterations</td>
<td>Training time</td>
</tr>
<tr>
<td>1</td>
<td>RBF model</td>
<td>574</td>
<td>2.41</td>
</tr>
<tr>
<td>2</td>
<td>BP model</td>
<td>2986</td>
<td>41.47</td>
</tr>
</tbody>
</table>


