Published in Engineering Intelligent Systems, 2004,Vol 4, P 205-210
Archived in Dspace@nitr
http://dspace.nitrkl.ac.in/dspace

An accurate fault classification algorithm using a minimal radial basis function neural network

P K Dash* and S R Samantray†

*Silicon Institute of Technology, Bhubaneswar, INDIA
†National Institute of Science and Technology, INDIA

The paper presents a new fault classification scheme for high speed relaying using minimal radial basis function neural network. Unlike earlier approaches in using radial basis function network, the new approach reduces the training time drastically and provides a systematic framework for selecting the number of neurons in the hidden layer. Further the minimal radial basis function network yields an accurate fault type classification on a transmission line even in the presence of high fault resistance in the fault path. The paper also presents two different approaches in generating the inputs to the neural network with a view to simplify the training procedure and reduce the complexity in calculations. Several computer simulated test results are presented to highlight the effectiveness of the new approach.

Keywords: RBF neural network, fault classification, network input generation

1. INTRODUCTION

Distance relaying techniques based on the measurement of impedance at the fundamental frequency between the fault location and the relaying point have attracted widespread attention. The sampled voltage and current data at the relaying point are used to classify the type of fault involving the line with or without fault resistance present in the fault path. The accuracy of the fault classification also depends on the amplitude of the dc offset and harmonics in comparison to the fundamental component. Fourier transforms, Differential equations, Waveform modelling, and Kalman filters are some of the techniques used for well-established fault detection and location calculations.

In recent times due to strong pattern recognition capabilities, ANNs (Artificial Neural Networks) are considered a viable alternative to the conventional fault classification approaches. Several ANN architectures have been considered [1-7] for the classification of faults on a power transmission line from the sampled voltage and current signals at the relaying point. Amongst these the multilayered perceptrons with backpropagation learning strategy and the radial basis function neural network present very attractive solutions in classifying faults and estimating the fault distances accurately. The BP algorithm, however, has a serious drawback of overlearning and thus can produce erroneous result when the fault data falls outside the trained patterns. Similarly the conventional RBF network [5] considered for distance relaying has the demerits of selecting the number of hidden units on a trial and error approach and determination of parameters of the Gaussian function by K-means clustering algorithm. Further a large

number of training patterns have been used for providing accurate classification of transmission line faults in that approach.

This paper presents a new approach, i.e. a minimal RBF network [8, 9] in which the number of hidden units is added depending on the newness or significant change in the data pattern. The parameters of the RBF neural network are updated using an Extended Kalman Filter (EKF) [10]. A pruning strategy to identify the ineffective neurons in the hidden layer is adopted to reduce the architecture of the RBF (terming it as minimal RBF) and to improve the efficiency of learning and provide robust prediction. With this new approach, the training time has been drastically reduced from almost 24 hours to a few minutes and the number of hidden neurons also is small in comparison to the earlier RBF approaches used for distance protection. The number of training patterns required for successful classification is also reduced. A number of digital simulations on a long transmission line fed from both ends are carried out to validate its robustness and accuracy of the new approach.

2. MINIMAL RBF NEURAL NETWORK

Radial basis function neural networks (RBFNN) are well suited for solving function approximation and pattern classification problems due to their simple topological structure and their ability to reveal as to how learning proceeds in an explicit manner. In the classical approach to RBF network implementation, the basis functions are usually chosen as Gaussian and the number of hidden units are fixed apriori based on some properties of input data.The weights connecting the hidden units to the output layer are normally determined by a least mean squares algorithm. The drawback of this method is that it results in too many hidden units and is not suitable for sequential learning. To overcome this drawback Bors and Gabbouj proposed an algorithm [8] to start with only one hidden neuron and sequentially add more neurons as learning proceeds depending on some chosen criterion based on output error. Further a pruning strategy is used to select only a minimal number of hidden neurons by observing their outputs and if at any stage it is observed that the output of any neuron is insignificant it is omitted from the hidden layer.

The structure of the RBF neural network (RBFNN) is shown in Figure 1.

For each input x_n and output y_n the estimated output of the network is

$$\hat{y}(n) = f(x_n) = \alpha_{m0} + \sum_{k=1}^{K} \alpha_{mk}\phi_k(x_n) \tag{1}$$

where n represents the time index, K = number of hidden units, α_{mk} = connecting weight of the k^{th} hidden unit to output layer, α_{mo} = bias term, m is the number of outputs and is chosen as 3 (a,b,c) in this case.

The network begins with no hidden units and as observations are received, new hidden units are added by taking some of the input data. The algorithm for developing a network structure is given below in detail.

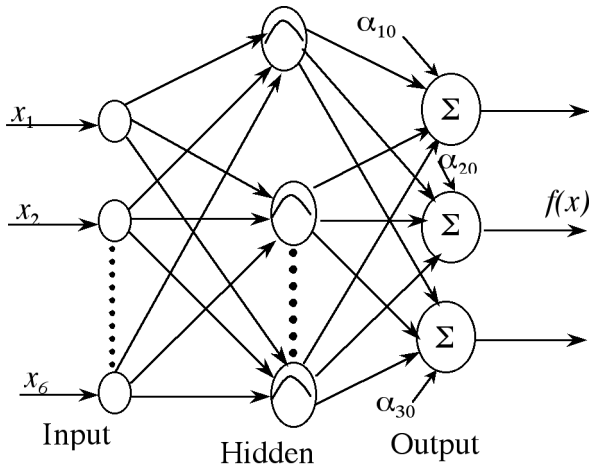

Figure 1 RBF neural network structure

Step 1: The value of $\phi_k(x_n)$ in equation (1) is the output from each hidden unit and is given by

$$\phi_k(x_n) = \exp\left(-\frac{1}{\sigma^2_k}\|x_n - \mu_k\|^2\right) \tag{2}$$

where μ_k is the centre vector for the kt^h-*hidden* unit and σ_k is the width of the Gaussian function; $\|\ \|$ denotes the Euclidean norm.

The error at the nth time step between estimated output $\hat{y}(n)$ and desired output $y(n)$ is $e(n) = |y(n) - \hat{y}(n)|$.

Define a quantity e_{rmsn} (RMS error magnitude) as

$$e_{rmsn} = \sqrt{\frac{\sum_{i=n\,M+1} [y(i) - \hat{y}(i)]^2}{M}} \tag{3}$$

If $e(n) > e_{min}$ and $\|x_n - \mu_{nr}\| \in_n$ and $e_{rmsn} > e'_{min}$ then allocate a new hidden unit with corresponding weights of

$$\alpha(k+1) = e_n,\ \mu_{k+1} = x_n \text{ and}$$
$$\sigma_{k+1} = \rho\|x_n - \mu_{nr}\| \tag{4}$$

where the threshold values are:

$$\in_n = \max\{\gamma^n \in_{\max}, \in_{\min}\},\ (0 < y < 1)$$
$$\in_{\max} = 0.05,\ \in_{\min} = 0.02,\ \gamma = 0.96$$
$$e_{\min} = 0.02 \text{ and } e'_{\min} = 0.2$$

and ρ is an overlap factor = 1.0, μ_{nr} is a centre of a hidden unit whose distance from x_n is the nearest among those of all the other hidden unit centres, M, the size of a sliding data window which covers a number of latest observations for calculating the RMS output errors e_{rmsn} =10.

Step 2: If the above conditions are not fulfilled the weights

of RBFNN are updated as

$$W_n = W_{n-1} + K_n e_n \tag{5}$$

where the vector W for a single output case is given by

$$W = [\alpha_0, \alpha_1, \mu_1^T, \sigma_1, \ldots. \alpha_K, \mu_K^T, \sigma_k] \tag{6}$$

the Kalman gain

$$K_n = P_{n-1} a_n [R_n + \alpha_n^T P_{n-1} \alpha_n]^{+1} \tag{7}$$

$$P_n = [I - K_n a_n^T] P_{n-1} + Q_n I$$

P_n = filter covariance, Q_n and R_n are state and observation error covariance matrices.
I: is an unit matrix of appropriate dimension, and
T = transpose of a quantity, and the Jacobian a_n is

$$a_n = \left.\frac{\partial f(x_n)}{\partial w}\right|_{W=W_{n/n-1}} \tag{8}$$

$$= [1, \phi_1(x_n), \phi_1(x_n) \frac{2\alpha_1}{\sigma_1^2}(x_n - \mu_1)^T,$$

$$\phi_1(x_n) \frac{2\alpha_1}{\sigma_1^3} \|x_n - \mu_1\|^2, \ldots\ldots \phi_K(x_n),$$

$$\phi_K(x_n) \frac{2\alpha_k}{\sigma_k^2}(x_n - \mu_K)^T, \phi_K(x_n) \frac{2\alpha_k}{\sigma_k^3} \|x_n - \mu_K\|^2]^T$$

Step 3: The pruning strategy removes the hidden units which make insignificant contributions to the overall network output consecutively over a number of training observations. The following procedure is adopted:

(i) compute the hidden unit output as

$$O_k^n = \alpha_k \exp\left(-\frac{1}{\sigma_k^2}\|x_n - \mu_k\|^2\right) \tag{9}$$

The outputs O_k^n (k = 1, 2, ... K) of all hidden units are then computed and the largest absolute hidden unit output $|O_{\max}^n|$ is found out.

(ii) calculate the normalized value for each hidden unit: $r_k^n = |O_k^n / O_{\max}^n|$ (k = 1, 2,..... K). If $r_k^n < \delta$, for M consecutive observations, then prune the k^{th} hidden neuron. The values of the parameters $P_0 = I$, $Q = 0.05$, $R = 1.0I$ and $M = 12$ are chosen in this paper.

3. FAULT CLASSIFICATION USING MINIMAL RBF NEURALNETWORK (RBFNN)

The RBFNN is used to identify the type of fault located in the first protection zone of the transmission line covering 80% of the line length from the sending end data only. In this investigation two approaches are considered for classification of fault type in a faulty power system. The first approach uses the peak values of the current and voltage waveforms within half a cycle time period as the input to the minimal RBFNN; the total number of inputs is six comprising three phase-voltages and three phase-currents. Here the peak value is calculated by finding the maximum sampled value from a string of voltage and current samples recorded by the data acquisition system. The peak value of the voltage or current obtained in this way is not the true peak of the fundamental voltage or current phasors due to the presence of decaying dc and harmonic components in the fault voltage and current waveforms. Thus another approach is presented below.

In the second approach, the peak values of the fundamental components of the voltage and current samples in the discrete form are used as inputs to the neural network. An extended Kalman filter algorithm (EKF) is used to compute the peak values of the voltage and current phasors from the sampled waveforms. This approach involves more computation than the first approach, but it provides better accuracy for detection and classification of fault types as the true values of the fundamental voltage and current phasors are used. The outputs of both networks represent the phase(s) involved. In both the approaches a sampling rate of 20 samples per cycle (1 kHz) is used. To identify whether ground is involved in the fault, a zero sequence-based indicator is proposed as in [11].

To generate data for the application of proposed strategy a 230kV, 190miles long transmission system as shown in Figure 2 is considered. EMTDC software package [12] is used to simulate fault data for different power system conditions. Only P end (as shown in Figure 2) data are being used for training and testing the RBF networks.

3.1 Training and testing of RBFNNs with fault data

3.1.1 First approach

The training of the first neural network was carried out with peaks of current and voltage signals of each phase within half a cycle after fault as input vector. The three outputs of the network reflect the involvement of the phase(s) in that particular fault situation. During training any phase involved in the fault is assigned '1' else '0'. Data are generated for different fault locations (varying from 10% to 80% of the line), fault types, fault resistances (R_f), fault inception angles, and system loading conditions. The peaks of the voltage and current waveforms in each phase within half a cycle of fault inception are computed using a simple comparator. The data sets thus generated

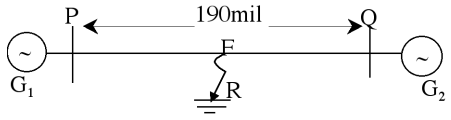

Figure 2 Power system for EMTDC simulation

Table 1 Fault at 10% of line at 45^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	1.1336	0.0402	-0.017	1.1274	0.1204	0.0396
bg	0.0247	0.9748	0.0662	0.2074	0.7858	0.0974
cg	0.0575	0.1187	1.0010	0.0065	-0.007	1.0905
ab	1.0860	0.8111	0.1634	1.0906	0.8309	0.1628
bc	0.2033	0.7789	0.8298	0.0698	0.9348	0.9884
ca	0.9951	0.0401	1.0293	1.0010	0.0410	1.0071
abc	0.9218	0.7007	0.9939	0.9608	0.6158	1.0187

Table 2 Fault at 40% of line at 45^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	1.0686	0.1336	0.0223	1.0839	0.1370	-0.0192
bg	0.1198	0.8106	0.1347	0.1436	0.7820	0.1308
cg	0.3616	0.0050	1.1055	0.3890	0.0245	1.0945
ab	0.9694	0.6951	0.0525	0.9694	0.6951	0.0525
bc	0.3584	0.5541	0.5947	0.3543	0.5882	0.5217
ca	0.8714	-0.0677	1.1484	0.8599	-0.0727	1.1507
abc	1.0261	0.5678	0.7314	1.0211	0.5410	0.7312

Table 3 Fault at 60% of line at 30^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 100\Omega$		
	a	b	c	a	b	c
ag	1.0649	0.0257	-0.0832	0.6290	0.1811	0.1710
bg	0.0913	0.8156	0.1084	0.2351	0.5730	0.1879
cg	0.1219	-0.0458	1.0358	0.2914	0.055	0.6650
ab	0.7582	0.7106	0.0196	0.7705	0.7761	0.0092
bc	-0.1851	0.8624	0.7457	-0.1747	0.7249	0.5893
ca	0.8285	-0.1221	1.0171	0.8160	-0.0932	0.9490
abc	0.9135	0.5847	0.5952	0.9112	0.6485	0.5528

Table 4 Fault at 80% of line at 90^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	1.0055	0.0190	-0.0600	0.9539	0.0463	-0.0489
bg	0.0813	0.9601	-0.0612	0.0897	0.9134	-0.0260
cg	0.0769	0.0155	0.8819	0.0445	0.0087	0.8835
ab	0.9932	0.7926	-0.0432	1.0416	0.1501	-0.0797
bc	0.0377	0.9263	0.6369	-0.0716	1.0121	0.6572
ca	0.8262	0.0197	1.0411	0.8162	0.0164	1.0496
abc	0.8553	0.9083	1.0243	0.8547	0.9083	1.0271
abg	0.7781	0.5059	0.0148	0.8822	0.3971	-0.0205
bcg	0.0375	0.9266	0.6367	0.0673	0.9807	0.6179
cag	0.7948	-0.0036	1.0954	0.7977	-0.0077	1.1057

are used for the training and testing purposes.

The training sets are 49 in number which include data for 10%, 40% and 80% fault locations for 0^0 and 90^0 fault inception angles and at different situations of the system without fault resistance and for only seven types of fault (ag, bg, cg, ab, bc, ca and abc). In comparison to earlier approaches [5] the choice of hidden neurons is no longer arbitrary, rather a sequential learning algorithm and a pruning strategy optimally fix the number of such neurons. Again to update the parameters while training the network an EKF is used unlike K-means clustering for μ, heuristic approach for σ and multiple linear regression for weights as in [5]. This clearly shows the simplicity of the strategy in the learning process.

The network so obtained was tested by data sets generated from EMTDC [11] which were not used during training. The data sets include all shunt faults at different locations, inception angles and prefault conditions of the system. Tables 1–4 present some of the test results for the faulted transmission line. Table 1 shows the performance of minimal RBFNN for different fault types at 10% of line for 45^0 inception angle at $R_f = 0$, and $R_f = 10\Omega$. The respective values in the a, b and c columns which are network outputs reflect the state of involvement of those phases. Say for '*ca*' case with $R_f = 0\Omega$ the values '*a*' = 0.9951, '*b*' = 0.0401 and '*c*' = 1.0293 depict that the phases associated with the fault are '*a*' and '*c*' only. This classification approach takes a particular phase to be involved with a fault if any of the RBFNN output value *a, b* or *c* is greater than 0.5 or else it categorizes the phase to be 'undisturbed'. Table 2 provides the fault classification results for different faults at 40% of the line at a different source impedance of ratio of 90. Table 3 shows fault classification for a different condition of $\delta = 20^0$ at 30^0 inception angle and at 60% of the line with a high resistance $R_f = 100\Omega$ in the fault path. Table 4 presents for a 900 inception angle and a fault at 80% of the line. These results demonstrate the suitability of the network even for the untrained categories of fault; '*abg*', '*bcg*' and '*cag*' etc. which are included in Table 4 (rows 8 through 10).

3.1.2 Second approach

Conventional distance relaying takes the trip decision based on current and voltage phasors. In the proposed approach we used these phasors as input vector to classify the fault by minimal RBFNN. The fundamental peaks of the voltage and current samples in each phase within half a cycle of fault inception are computed using Kalman filtering techniques where the decaying dc, third and fifth harmonics are modeled. As mentioned in section 3, this approach is expected to result in better accuracy for fault type classification. The target outputs remaining same as the first minimal RBFNN, the second network took only 28 sets of data for training. The network performance is studied at different situations of the power system and a comparison is made with the first RBFNN in the following.

3.1.3 Comparison between the two approaches

For a meaningful assessment of the two approaches the voltage and current samples derived for situations as for Tables 1–4 are used to obtain the trip decision by the second approach. Tables 5–8 show results for fault classification as obtained by second approach using voltage current samples for cases mentioned in Tables 1–4. The results

Table 5 Fault at 10% of line at 45^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	1.2489	0.2799	0.0218	1.2480	0.2867	0.0138
bg	0.2246	1.1484	0.2027	0.2436	1.1105	0.2614
cg	0.2436	0.1440	1.0966	0.3783	0.0830	1.0763
ab	1.1580	1.0464	0.1042	1.1246	1.0468	0.0902
bc	0.1465	1.0570	1.1185	0.1030	1.0745	1.0902
ca	1.1430	0.1126	1.0172	1.0949	0.0735	1.0519
abc	1.1162	1.0655	1.0563	1.1130	1.0702	1.0560

Table 6 Fault at 40% of line at 45^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	0.9615	0.2185	0.0041	1.0366	0.4616	-0.0524
bg	0.2271	1.0370	0.4280	0.1920	1.0719	0.4341
cg	0.1352	0.0134	1.4076	0.1490	0.0096	1.4277
ab	0.6182	0.8937	0.1462	0.6185	0.8938	0.01459
bc	0.2441	0.8042	0.7451	0.2432	0.8044	0.7449
ca	0.9288	-0.0416	1.1513	0.9291	-0.0432	1.1516
abc	0.5616	0.9111	0.7572	0.5427	0.9162	0.7575

Table 7 Fault at 60% of line at 30^0 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	0.9650	-0.0051	0.0327	0.9172	0.1660	0.1096
bg	-0.0232	0.9937	0.0214	0.3588	0.6594	0.0871
cg	-0.0038	-0.0032	1.0163	0.3517	0.1246	0.6734
ab	0.9938	0.9793	0.0005	1.0265	0.9078	-0.0026
bc	-0.0001	0.9999	1.1006	0.1280	0.8138	0.8694
ca	1.0320	0.0021	1.1016	0.8046	0.0606	0.8748
abc	0.9951	0.9923	0.9861	0.8919	0.9476	0.8876

clearly demonstrate the superior performance of the second RBFNN over the first one in delivering the proper fault classification. For example, rows 4 and 8 in Tables 4 and 8 for '*ab*' and '*abg*' fault types, respectively, show the outputs of first and second RBFNNs for 90^0 inception angle situation. The classification by first RBFNN fails for phase '*b*' for R_f = 10Ω whereas second one rightly performs. In rest of the shown cases both approaches classify accurately. The test results show that the networks perform excellently even if the initial condition, inception angle, fault resistance and source impedance vary significantly. However, the second approach needs extra filtering arrangements for execution but with better accuracy.

3.2 Ground detection

In reference [10] for detecting the involvement of ground during fault, a zero sequence current based indicator of the type

Table 8 Fault at 80% of line at 900 inception angle

Fault Type	$R_f = 0\Omega$			$R_f = 10\Omega$		
	a	b	c	a	b	c
ag	0.8526	0.2917	0.2170	0.8319	0.2526	0.1909
bg	0.0820	1.0392	0.0383	0.0433	1.0595	0.0334
cg	-0.2449	0.1078	1.1816	-0.2467	0.1063	1.1599
ab	1.1661	0.7385	0.0785	1.1504	0.5742	0.0746
bc	0.0488	1.0736	0.7277	0.0388	1.1008	0.5788
ca	0.8906	-0.0061	1.1167	0.6671	-0.0366	1.1283
abc	0.9747	0.6871	0.6338	0.9483	0.7121	0.6741
abg	1.1486	0.7844	0.0854	1.1711	0.7207	0.0781
bcg	0.0467	1.0707	0.7539	0.0025	1.1317	0.6936
cag	0.8671	-0.0055	1.1274	0.8204	-0.0356	1.1423

$$Index1 = \frac{|Ia + Ib + Ic|}{\text{median}(|Ia|,|Ib|,|Ic|)}$$

is proposed. Here *Ia*, *Ib* and *Ic* are the current phasors of the three phases at the sending end. However, without resorting to Tustin approximation for finding the phasors [11], in this work the phasors are estimated by the EKF method and the corresponding *Index1* value is calculated. When the *Index1* value exceeds the threshold value of 0.05, it indicates the involvement of fault with ground. The ground-detection is carried out in parallel with the RBFNN calculations. Test results showing the values of *Index1* for 'a' – phase to – 'b' faults at a distance of 10% of the line for the power system are presented in Table 9.

4. CONCLUSIONS

The use of a minimum Radial Basis function neural network as a pattern classifier to simulate a distance relay is investigated in this paper. Using data of fast half cycle after the fault inception two approaches (with and without filtering) are demonstrated in this paper. The second RBFNN classifier is found to give fast and precise operation and provides robust fault classification in the presence of dc offset, noise and harmonics in the voltage and current signals of the faulted line. Also the approach provides excellent fault type classification for a variety of system operating conditions. The minimal RBF neural network results in a more systematic approach in choosing the number of neurons in the hidden layer and a faster learning by suitably adjusting the parameters of the computing neurons. Also the number of training data is drastically reduced with this approach which is found to be suitable for real-time applications.

Table 9 Index1 for fault at 10% of the line

Fault Type	Index1
ab	0.0036
abg(Rf=0W)	0.4771
abg(Rf=10W)	0.4592
abg(Rf=50W)	0.2405

REFERENCES

1 Sidhu, T.S., Singh, H., and Sachdev, M.S.: Design, Implementation and Testing of an Artificial Neural Network Based Fault Direction Discrimination for Protecting Transmission Lines, *IEEE Trans. on Power Delivery*, 1995, 10 (2), pp. 697–706.

2 Dalstein, T., and Kulicke,B.: Neural Network Approach to Fault Classification for High Speed Protective Relaying, *IEEE Trans. on Power Delivery*, 1995, 10 (2), pp. 1002–1011.

3 Novosel, D., Bachmann, B., Hart, D., Hu, Y., and Saha, M.M.: Algorithms for Locating Faults on series Compensated Lines Using Neural Network and Deterministic Methods, *IEEE Trans. on Power Delivery*, 1996, 11 (4), pp. 1728–1736.

4 Song, Y.H, Johns, A.T., and Xuan, Q.Y.: Artificial Neural Network Based Protection Scheme for Controllable series-compensated EHV transmission lines, *IEE Proc. Gen. Trans. Dist.*, Vol. 143, No. 6, 1996, pp. 535–540.

5 Song, Y.H., Xuan, Q.Y., and Johns, A.T.: Protection Scheme for EHV Transmission Systems With Thyristor Controlled Series Compensation using Radial Basis Function Neural Networks, *Electric Machines and Power Systems,* 1997, pp. 553–565.

6 Narendra, K.G., Sood, V.K., Khorasani, K., and Patel, R.: Application of a Radial Basis Function (RBF) Neural Network for Fault Diagnosis in HVDC Systems, *IEEE Trans. on Power Systems*, 1998, 13, pp. 177–183.

7 Coury, D.V., and Jonge, D.C.: Artificial Neural Network Approach to Distance Protection, *IEEE Trans. on Power Delivery*, 1998, 13 (1), pp. 102–108.

8 A.G. Bors, and M. Gabbouj, Minimal topology for radial basis functions neural network, *Neural Computations,* 1994, 5, pp. 954–975.

9 Yingwei, L., Sundarrajan, N., and Saratchandran, P.: Performance Evaluation of a Sequential Minimal Radial Basis Function (RBF) Neural Network Learning Algorithm, *IEEE Trans. on Neural Networks*, 1998, 9 (2), pp. 308–318.

10 Hilands, T.W., and Thomopoulos, S.C.A.: Nonlinear Filtering Methods of Harmonic Retrieval and Model Order Selection in Gaussian and Non-Gaussian Noise, *IEEE Trans. on Signal Processing* 1997, 45, (4), pp. 983–994.

11 Akke, M., and Thorp, J.T.: Some Improvements in the Three-Phase Differential Equation Algorithm for Fast Transmission Line Protection, *IEEE Trans. on Power Delivery*, 1998, 13 (1), pp. 66–72.

12 Manitoba HVDC Research Center, EMTDC – Electromagnetic Transients Simulation Program, 1988