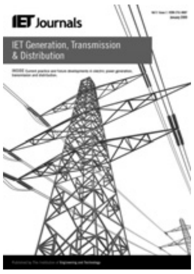
Published in IET Generation, Transmission & Distribution
Received on 31st August 2009
Revised on 8th January 2010
doi: 10.1049/iet-gtd.2009.0488

Special Issue on Wide Area Monitoring, Protection and Control



ISSN 1751-8687

PSO and ANN-based fault classification for protective relaying

J. Upendar[1,2] C.P. Gupta[1] G.K. Singh[1] G. Ramakrishna[2]

[1]Department of Electrical Engineering, Indian Institute of Technology, Roorkee, India
[2]Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada
E-mail: jallaupendar@gmail.com

Abstract: Fault classification in electric power system is vital for secure operation of power systems. It has to be accurate to facilitate quick repair of the system, improve system availability and reduce operating costs due to mal-operation of relay. Artificial neural networks (ANNs) can be an effective technique to help to predict the fault, when it is provided with characteristics of fault currents and the corresponding past decisions as outputs. This paper describes the use of particle swarm optimisation (PSO) for an effective training of ANN and the application of wavelet transforms for predicting the type of fault. Through wavelet analysis, faults are decomposed into a series of wavelet components, each of which is a time-domain signal that covers a specific octave frequency band. The parameters selected for fault classification are the detailed coefficients of all the phase current signals, measured at the sending end of a transmission line. The information is then fed into ANN for classifying the faults. The proposed PSO-based multi-layer perceptron neural network gives 99.91% fault classification accuracy. Moreover, it is capable of producing fast and more accurate results compared with the back-propagation ANN. Extensive simulation studies were carried out and a set of results taken from the simulation studies are presented in this paper. The proposed technique when combined with a wide-area monitoring system would be an effective tool for detecting and identifying the faults in any part of the system.

1 Introduction

Increased electricity supply demand and at the same time the restrictions applied on power system transmission expansion have resulted in reduced operational margins for many power utilities. There is a need to implement new monitoring, protection and communications technologies [1] to improve the security of the current power system networks. The wide-area monitoring system (WAMS) has the ability to make optimal use of the transmission networks by using real-time information [2]. The WAMS architecture consists of efficient hardware and software packages with special protection and control schemes to optimise power transmission capacity, maintain grid integrity and to ascertain acceptable power system performance at any time. An advanced, accurate and early warning wide-area protection system (WAPS) schemes for power grids that help operators must act to prevent system instabilities and overloads, as well as cascade tripping that leads to power blackouts. For this reason, the new design requirements of WAPS schemes are very demanding [3, 4].

In contrast to conventional protection devices, which provide local protection of individual equipment (transformer, generator, line, etc.), the WAPS provides comprehensive protection coverage of the entire power system. Protective relaying involves detection, classification and location of transmission line faults. Classification of faults means identification of the type of fault; this information is required for accurate fault location to carry out quick maintenance and repair work and restoration of the line to improve the reliability and availability of the power supply. Owing to this, the developments of advanced and reliable techniques for fault detection and classification have received considerable attention in the protection area.

Normally, the traditional method of signal analysis is based on Fourier transforms. Fourier transform is a process of multiplying a signal by a sinusoid in order to determine the frequency contents of a signal. The output of the Fourier transform is sinusoids of different frequencies. If there is a local transient over some small interval of time in the lifetime of the signal, the transient will contribute to the Fourier transform but its location on the time axis will be lost. It is found that Fourier transform is not appropriate to analyse faults in a power system with transient-based protection. Although the short-time Fourier transform overcomes the time location problem to a large extent, it does not provide multiple resolutions in time and frequency, which is an important characteristic for analysing transient signals containing both high and low-frequency components. Wavelet analysis [5, 6] overcomes the limitations of Fourier methods by employing analysing functions that are local both in time and in frequency, which are used to analyse the fault current waveforms in this paper. Recently, several applications of wavelet transform (WT) to power systems have appeared, namely, power quality monitoring, data compression, transient analysis, fault direction discrimination [7], fault section estimation [8], adaptive relaying [9], auto-reclosing [10], fault diagnosis and protection.

Earlier, adaptive Kalman filtering approach was proposed for the protection of transmission system; however, such protection strategy required a Kalman filter design, and if the initial estimate of the state is wrong, or if the process is modelled incorrectly, the filter may quickly diverge due to the linear model chosen. In the past, several attempts have been made for fault classification using travelling wave-based approach. However, travelling wave method [11] require high sampling rate and have problems in distinguishing between waves reflected from the fault and from the remote end of the line. Girgis and Johns [12] used an expert-system-based approach and a phasor measurement unit (PMU)-based approach was described in [13] for the analysis of fault. The drawback of these approaches is that they depend on the phasor calculation. The application of fuzzy logic to classify the faults was used in relaying [14]. The benefit of fuzzy logic is that its knowledge representation is explicit, using simple 'IF–THEN' relations. But logic-based expert systems have a combinatorial explosion problem [14] when applied to a large system. Again, the accuracy of fuzzy logic-based schemes cannot be guaranteed for wide variations in the system conditions.

Artificial neural networks (ANNs) have attracted a great deal of attention in the past two decades in areas such as pattern classification, function approximation, digital signal processing, intelligent control, power system analysis, fault detection, data compression, analysis for power quality problem solution, power quality assessment, protection, transient analysis and so on successfully, because of its computational speed and robustness and have become an alternative to modelling of physical system such as a transmission line [15]. Although there are many types of neural networks, only a few of neuron-based structures are being used commercially. One particular structure, based on multi-layer perceptron, called back-propagation neural network (BPNN), is the most popular neural network architecture, which uses supervised learning to determine a complex, nonlinear, multidimensional mathematical fitting.

This paper presents an application of particle swarm optimisation (PSO)-based multi-layer perceptron neural network in classifying faults in transmission lines with the help of WTs. The proposed method is capable of providing a reliable and fast estimation of fault types on the basis of measurement of three-phase currents using WT. The method classifies whether a normal state, single-line-to-ground, double-line-to-ground, phase-to-phase or a three-phase fault has occurred. The proposed algorithm is tested on a 400 kV two-terminal transmission line simulated using MATLAB/SIMULINK®. The performance of the proposed technique is analysed by comparing the fault classification results with the original BPNN method for the same test data considering wide variations in system operating conditions.

2 Wavelet analysis

Wavelet analysis [5, 6] is a mathematical technique for signal processing and is inherently suited for non-stationary and non-periodic wide-band signals. It helps in archiving the localisation both in frequency and in time scale. Wavelet analysis involves an appropriate wavelet function called 'mother wavelet' and performs analysis using shifted and dilated versions of this wavelet. The continuous WT (CWT) of a continuous signal $x(t)$ is defined as

$$\mathrm{CWT(a,\,b)} = \int_{\infty}^{-\infty} x(t)\Psi^{*}_{a,b}\,\mathrm{d}\,t \tag{1}$$

where $\Psi(t)$ is the mother wavelet and other wavelets $\Psi_{a,b}(t) = 1/\sqrt{a}\Psi^{*}_{a,b}(t - b/a)$ are its dilated and translated versions, the constants a and b being dilation (scale) and translation (time shift) parameters, respectively. The CWT at different scales and locations provides variable time–frequency information of the signal.

The digitally implementable counterpart of CWT known as discrete WT (DWT) is the one which is used for the proposed fault classification. The DWT of a signal $x(t)$ is defined as

$$\mathrm{DWT}(x,\,m,\,n) = \frac{1}{\sqrt{a_0^m}}\sum_m\sum_n x(k)\Psi^{*}\left(\frac{k - nb_0a_0^m}{a_0^m}\right) \tag{2}$$

where $a = a_0^m$ and $b = nb_0a_0^m$; a_0 and b_0 being fixed constants are generally taken as $a_0 = 2$ and $b_0 = 1$ and k, m and n are integers.

The actual implementation of DWT is done by multi-resolution analysis (MRA) [16]. The original signal is analysed at different frequency bands with different resolutions. The signal is decomposed into a smooth approximation version and a detail version. The approximation is further decomposed into an approximation and a detail, and the process is repeated. This decomposition of the original signal is obtained through successive high-pass and low-pass filtering of the signal. The successive stages of decomposition are known as levels. The MRA details at various levels contain the features for the detection and classification of faults.

3 PSO-based neural network algorithm

3.1 Artificial neural networks

The concept of ANNs have been around since the 1950s and are biologically inspired from the view of the human brain as a processor using interconnected neurons. The ANN is connected like the brain with artificial neurons that are interconnected and adaptive to the output of other connected nodes which have modifiable parameters. One of the most popular neural networks is the BPNN method for solving many nonlinear problems, but the original BP networks used to suffer mainly from the drawbacks of slow convergence, because they used to get trapped at local minima. Over the years, different popular, improved variations of BPNN have been proposed to specifically address several important issues, namely, reduction in convergence time, ease of computational burden, reduced memory requirement and so on [17].

In a typical multi-layer feed-forward perceptron ANN, there exists a nonlinear mapping between the input vector and the output vector via a system of simple interconnected neurons. It is fully connected to every node in the next and previous layer. The output of a neuron is scaled by the connecting weight and fed forward to become an input through a nonlinear activation function to the neurons in the next layer of the network. In the course of training, the perceptron is repeatedly presented with the training data. The weights in the network are then adjusted until the errors between the target and the predicted outputs are small enough, or a pre-determined number of epochs are passed. The training processes of the ANN are usually complex for high-dimensional problems. A major drawback of the commonly used gradient-based BP algorithm, which is a local search method, is its easy entrapment at a local optimum point, especially for those nonlinearly separable pattern classification problems or complex function approximation problem [18], so that BP may lead to failure in finding a global optimal solution. Another drawback is that the convergence speed of the BP algorithm is too slow even if the learning goal can be achieved. The important point to be stressed here is that the convergence behaviour of the BP algorithm depends very much on the choices of initial values of the network connection weights as well as the parameters in the algorithm such as the learning rate and the momentum [19].

To improve the performance of the original BP algorithm, many researchers have concentrated on the following two factors: (1) selection of better energy function [20]; (2) selection of dynamic learning rate and momentum [21, 22]. However, these improvements have not completely removed the disadvantages of the BP algorithm getting trapped into local optima in essence. In particular, with ANN's structure becoming more complex, its convergence speed will become even slower.

Genetic algorithm (GA) has been also used in training ANNs recently, but in the training process, this algorithm needs various GA operators. Usually, there are three kinds of complicated evolutionary operators associated with this algorithm, i.e. selection, crossover and mutation. Even though its training results may be better than the ones using the BP algorithm, but when the neural network structure is large, the GA's convergent speed becomes very slow.

The PSO algorithm is an alternative to GA in training the perceptrons of the ANN. The advantage of the PSO over the GA is in terms of simplicity. While there are several options for implementing a GA, for example, one may choose tournament or proportionate selection. In the PSO, however, there is one simple operator: velocity calculation. With fewer operators, the number of computations can be reduced significantly and this eliminates the need for selection procedure for choosing the best operator for a given optimisation.

3.2 Particle swarm otimisation

The PSO was originally introduced by Kennedy and Eberhart in 1995. PSO optimises an objective function by undertaking a population-based search. The PSO algorithm has been recognised as a computational intelligence technique and is closely related to evolutionary algorithms. Moreover, it is tailored for optimising difficult numerical functions and based on metaphor of human social interaction [23]. PSO is a populated search method for the optimisation of continuous nonlinear functions resembling the movement of organisms in a bird flock or fish school. It is computationally inexpensive in terms of both memory requirements and speed. It lies somewhere between evolutionary programming and GAs.

As in evolutionary computation paradigms, the concept of fitness is employed and candidate solutions to the problem are termed particles or sometimes individuals, each of which adjusts its flying based on the flying experiences of both itself and its companions. It keeps track of its coordinates in hyperspace which are associated with its previous best-fitness solution, and also of its counterpart

doi: 10.1049/iet-gtd.2009.0488

corresponding to the overall best value acquired thus far by any other particle in the population. Vectors are taken as a presentation of particles since most optimisation problems are convenient for such variable presentations. It is adaptive corresponding to the change of the best group value. The allocation of responses between the individual and group values ensures a diversity of response. The higher-dimensional space calculations of the PSO concept are performed over a series of time steps. The population is responding to the quality factors of the previous best individual values and the previous best group values. The principle of stability is adhered to since the population changes its state if and only if the best group value changes.

A similarity between PSO and a GA [24] is the initialisation of the system with a population of random solutions. Instead of employing genetic operators, the evolution of generations of a population of these individuals in such a system is by cooperation and competition among the individuals themselves. Moreover, a randomised velocity is assigned to each potential solution or particle so that it is flown through hyperspace. While the stochastic factors allow a thorough search of spaces between regions that are spotted to be relatively good, the momentum effect of modifications of the existing velocities leads to the exploration of potential regions of the problem domain. In this way, the adjustment by the particle swarm optimiser is ideally similar to the crossover operation in GAs while the stochastic processes are close to evolutionary programming. Since the PSO algorithm has been found to be able to find the global optimum with a large probability and high convergence rate, it is adopted to train the multi-layer perceptrons in this study.

A detailed description of PSO is provided below.

Step 1: Initialisation. The PSO contains 's' individual swarms called particles. Each particle represents a possible solution to a problem with d-dimensions and its genotype consists 2^*d parameters. First d-parameters, $x_i = (x_{i1}, \; x_{i2} \ldots x_{id})$, represent the particle positions and next d-parameters, $V_i = (V_{i1}, \; V_{i2} \ldots V_{id})$ represents velocity components. The velocity and position of all particles are randomly set to within pre-defined ranges.

Step 2: Velocity updating. Velocity for particle i, at iteration $t+1$, can be updated using velocity contained in previous iteration t, to be represented as

$$\vec{V}_{id}(t+1) = w \cdot \vec{V}_{id}(t) + c_1\phi_1(P_{\text{best}}^{id} - P_{\text{id}}(t)) + c_2\phi_2(G_{\text{best}}^{id} - P_{id}(t))$$

where $i = 1, 2, 3, \ldots, s$, $d = 1, 2, 3, \ldots, m$, $m =$ the number of input variables to be optimised. One set of all input variables is called as one particle, $s =$ the number of particles in a group, $w =$ inertia weight factor, varies linearly from $w_{\min}$ to $w_{\max}$, ranges between (0, 1), c_1, $c_2 =$ cognitive and social acceleration factors, respectively, compromise the inevitable tradeoff between exploration and exploitation, φ_1, $\varphi_2 =$ uniformly distributed random numbers in the range of (0, 1). The inclusion of random variables endows the PSO with the ability of stochastic searching, $P_{id}(t) =$ current position of ith particle of input variable d at iteration t, P_{best}^{id} and G_{best}^{id} are the position with the 'best' objective value found so far by particle i and the entire population, respectively; After updating, V_i should be checked and secured within a pre-specified range to avoid violent random change.

Step 3: Position updating. Then the new position of each particle is evaluated as the sum of its previous position and the corresponding updated velocity obtained in the previous step can be represented as

$$\vec{P}_{id}(t+1) = \vec{P}_{id}(t) + \vec{V}_{id}(t+1)$$

Step 4: Memory updating. Update P_{best}^{id} and G_{best}^{id} when the condition is met

$$\begin{aligned} P_{\text{best}}^{id} &= P^{id} \quad \text{if } f(P^{id}) > f(P_{\text{best}}^{id}) \\ G_{\text{best}}^{id} &= G^{id} \quad \text{if } f(G^{id}) > f(G_{\text{best}}^{id}) \end{aligned}$$

where $f(x)$ is the objective function subject to maximisation.

These parameters move with an adaptable velocity within the search space and retain its own memory with the best position it ever reached. The parameters change when moving from the present iteration to the next iteration. At every iteration, the fitness function as a quality measure is calculated by using its position vector. Each particle keeps track of its own position, which is associated with the best fitness it has achieved so far. The best position obtained so far for particle i keeps the track as $P_{\text{best}}^{i} = (P_{\text{best}}^{i1}, P_{\text{best}}^{i2}, P_{\text{best}}^{i3}, \ldots, P_{\text{best}}^{id})$. The best global version particle among the entire group of particles keeps the track of G_{best}.

Step 5: Termination checking. The algorithm repeats Steps 2–4 until certain termination conditions are met, such as a pre-defined number of iterations or a failure to make progress for a certain number of iterations. Once terminated, the algorithm reports the values of G_{best} and $f(G_{\text{best}})$ as its solution. Computational process of the proposed algorithm depicts as shown in Fig. 1.

3.3 Training neural networks with PSO

In this application, a three-layered perceptron is considered with four input parameters and four output parameters as shown in Fig. 2. When designing a neural network, one major difficulty is to determine the appropriate number of neurons in the hidden layers. The hidden layer is responsible for the internal representation of the data and the information transformation between input and output layers. If there are too few neurons in the hidden layer, the

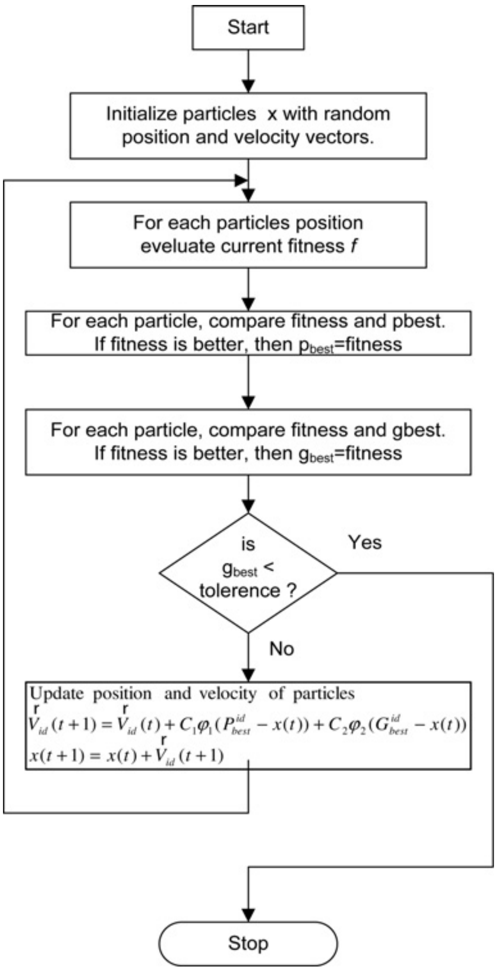

Start
Initialize particles x with random position and velocity vectors.
For each particles position eveluate current fitness f
For each particle, compare fitness and pbest. If fitness is better, then p_{best}=fitness
For each particle, compare fitness and gbest. If fitness is better, then g_{best}=fitness
is g_{best} < tolerence ?
Yes
No
Update position and velocity of particles
$\overset{r}{V}_{id}(t+1)=\overset{r}{V}_{id}(t)+C_1\varphi_1(P^{id}_{best}-x(t))+C_2\varphi_2(G^{id}_{best}-x(t))$
$x(t+1)=x(t)+\overset{r}{V}_{id}(t+1)$
Stop

Figure 1 *Flowchart for the PSO algorithm*

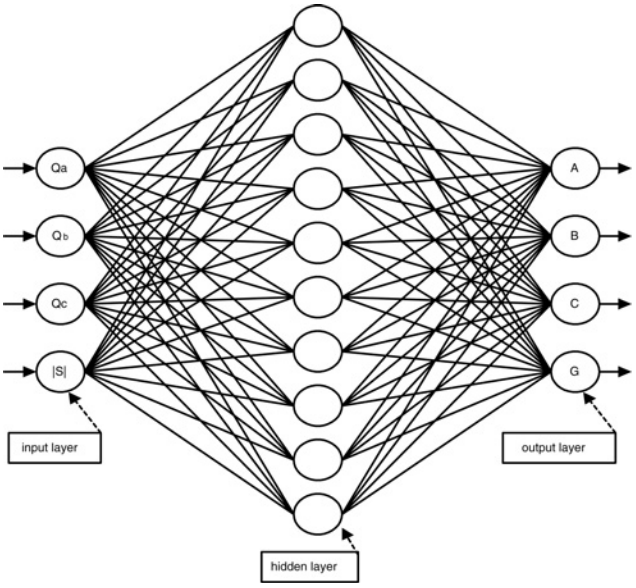


Figure 2 *Architecture of multi-layer perceptron neural network*

network may not contain sufficient degrees of freedom to form a representation. If too many neurons are used, the network might become over trained. Therefore an optimum design for the number of neurons in the hidden layer is required. To achieve good performance, it is evaluated for various values of neurons in hidden layer. A topology with five neurons in hidden layers showed the best performance for the problem. To evolve the network weights with PSO, the particle will be represented as group of weights; there are $4^*5+5^*4=40$ weights, so the particle consists of 40 real numbers.

In the training of the multi-layer perceptrons by the PSO, the representation of the connection weight matrix of the ith particle is as

$$\boldsymbol{W}=\left\{\boldsymbol{W}_i^{[1]},\ \boldsymbol{W}_i^{[2]}\right\} \tag{3}$$

where $W_i^{[1]}$ and $W_i^{[2]}$ represents the connection weight matrix of the ith particle between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively.

Moreover, the vector of the position of the previous best fitness value of any particle is represented by

$$\boldsymbol{P}=\left\{\boldsymbol{P}_i^{[1]},\ \boldsymbol{P}_i^{[2]}\right\} \tag{4}$$

where $\mathrm{P}_i^{[1]}$ and $\mathrm{P}_i^{[2]}$ represent the position of the previous best-fitness value of the ith particle, between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively.

The index of the best particle among all the particles in the population is represented by the symbol b. So the best matrix is represented by

$$P_b=\left\{P_{\mathrm{b}}^{[1]},\ P_{\mathrm{b}}^{[2]}\right\} \tag{5}$$

where $\mathrm{P}_b^{[1]}$ and $\mathrm{P}_b^{[2]}$ represent the position of the best particle among all the particles, between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively.

The velocity of the particle i is denoted by

$$V=\left\{V_i^{[1]},\ V_i^{[2]}\right\} \tag{6}$$

If m and n represent the index of matrix row and column, respectively, the manipulation of the particles are as follows

$$\begin{aligned}\boldsymbol{V}_i^{[j]}(m,n)={}&\boldsymbol{V}_i^{[j]}(m,n)+c_1\phi_1(\boldsymbol{P}_i^{[j]}(m,n)-\boldsymbol{W}_i^{[j]}(m,n))\\&+c_2\phi_2(\boldsymbol{P}_b^{[j]}(m,n)-\boldsymbol{W}_i^{[j]}(m,n))\end{aligned} \tag{7}$$

$$\boldsymbol{W}_i^{[j]}=\boldsymbol{W}_i^{[j]}+\boldsymbol{V}_i^{[j]}(m,n) \tag{8}$$

doi: 10.1049/iet-gtd.2009.0488

where $j = 1, 2$; $m = 1, \ldots, M_j$; $n = 1, \ldots, N_j$; M_j and N_j are the row and column sizes of the matrices $\boldsymbol{W}, \boldsymbol{P}$ and $\boldsymbol{V}$; c_1 and c_2 are cognitive and social acceleration factors, respectively, and ϕ_1 and ϕ_2 are uniformly distributed random numbers in the range of (0, 1).

Equation (7) is employed to compute the new velocity of the particle based on its previous velocity and the distances of its current position from the best experiences both in its own and as a group. Updated velocity must be within the specified range. If it violates the limits, it is set to a popular value. The maximum velocity is assumed as '5' and the minimum velocity as '−5'. The initial velocities of the initial particles were generated randomly in the range of (0, 1). After each iteration, if the calculated velocity was larger or smaller than the maximum velocity or the minimum velocity, it would be reset to '5' or '−5'.

The velocity change of PSO in (7) consists of three parts. The first part is the *momentum* part, which prevents the velocity to be changed abruptly. The second part is the cognitive part, which represents the private thinking of itself, meaning learning from its own flying experience. The third part is the social part, which represents the collaboration among the particles learning from the group best flying experience. The balance among these three parts determines the balance of global and local search ability. Then the new position of each particle is evaluated as sum of its previous position and corresponding updated velocity using (8).The fitness of the *i*th particle is expressed in terms of an output mean-squared error of the neural networks as follows

$$f(\mathrm{W}_i) = \tfrac{1}{S}\sum_{k=1}^{S}\left[\sum_{l=1}^{O}\{t_{kl} - p_{kl}(W_i)\}^2\right] \qquad (9)$$

where f is the fitness value, t_{kl} is the target output, p_{kl} is the predicted output based on W_i, S is the number of training set samples and O is the number of output neurons. These operations are repeated for a predefined number of iterations or until fitness is reached.

4 Application of the proposed PSO method in WAMS

4.1 PMU data

The Bonneville Power Administration was the first utility to implement synchrophasors (PMUs) in its WAMS. The essential feature of the PMU technique is that, during any operating conditions, it measures voltages and currents of a power system in real-time with precise-time synchronisation, with which to monitor and control a power system. PMU allows accurate comparison of measurements over various locations in a wide area, as well as potential real-time measurement-based control actions. This system performance can be improved with the help of information availability and information interpretation provided by system dynamic information through PMUs [25] from all over the grid. Very fast recursive discrete Fourier transform (DFT) are normally used in phasor calculations [26].

The different PMUs over a wide area power system are synchronised through a Global Positioning Satellite (GPS) system. The GPS system provides continuous precise timing at less than 1 μs level.

The simple block diagram of synchronised phasor measurement system (PMU) is as shown in Fig. 3. The signals from GPS transmission is received by the receiver section, which delivers a phase-locked sampling clock pulse to the analogue-to-digital converter system. The sampled data are converted to a required format which represents the phasor of the sampled waveform. In WAPS schemes, data from synchrophasor placed at remote sites will be transmitted via communication links to a data concentrator in a common time reference frame, for processing and implementation of protection and control algorithms as well as being recorded for system analysis. The format of synchronised data transmission using communication links has been discussed in IEEE standard 1344-1995 [26]. With the rapid development of fibre-optic and digital-communication systems, multiplexed digital channels are often available for use with transmission-line protection. Therefore high speed for data transfer can be achieved so

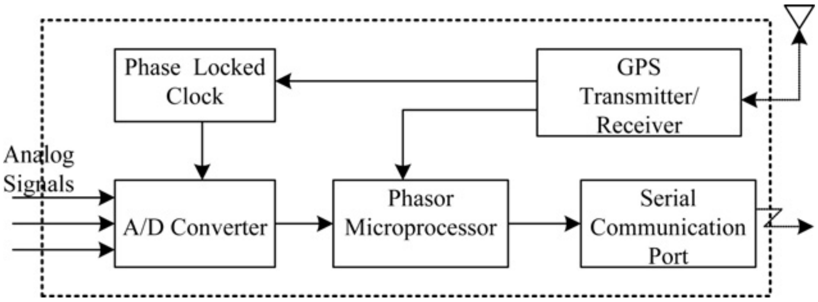


Figure 3 *Block diagram of the synchronised phasor measurement system (PMU)*

that the transmission delay can be added only a few milliseconds to the tripping decision time of the protection scheme. When currents are measured in this fashion, it is important to have a high enough resolution in the analogue-to-digital converter to achieve sufficient accuracy of representation at light loads. A 16-bit converter or equivalent generally provides adequate resolution to read load currents, as well as fault currents.

4.2 Architecture of the wide area platform

The architecture of WAMS platform for a large power system network is shown in Fig. 4. Wide area protection is a highly complex task. If a disturbance occurs in any one of the interconnected systems, it is very difficult to arrive at a diagnosis in a short period of time. The wide knowledge of the system is essential in resolving the problem. Owing to the nature of the complexity, isolated algorithms are not suitable for diagnosing the wide area disturbance. There is a need for cooperating expert system algorithms, which can assist the local experts during emergencies and help solve routine work (overall system and individual system status report, etc.) that needs to be carried out during the system wide disturbance.

PMUs measure the significant signals such as all bus voltages and line currents. These measurements are sent to its own phasor data concentrator (PDC) present at each utility centre, and then the measurements from each utility's PDC are sent to a central control centre (CC). The PDC correlates the data by time-tag to create a system-wide measurement. The PDC exports these measurements as a data stream as soon as they have been received and correlated. Multiple applications can receive these data stream and use it for display, recording, and control functions. Standard communication systems are adequate for most phasor data transmission. The issues such as speed, latency, reliability and communication speed (data rate) during data communications process depends on the amount of phasor data being sent and the number of messages/s. In case, if PMU sending 10 phasors at 30 messages/s, the data rate to be about 17 kbps. A analogue modem operating at 33.6 kbps is fast enough for this application. However, if the rate increases to 60 messages/s, a faster system, such as a 56 kbps digital link, is to be used.

The systems associated with the communication delays or latency are mainly data concentrators, transducers, data size of PMU output, multiplexing and transitions, window size of data, processing time, involved communication links. It is imperative for the utility to choose a right

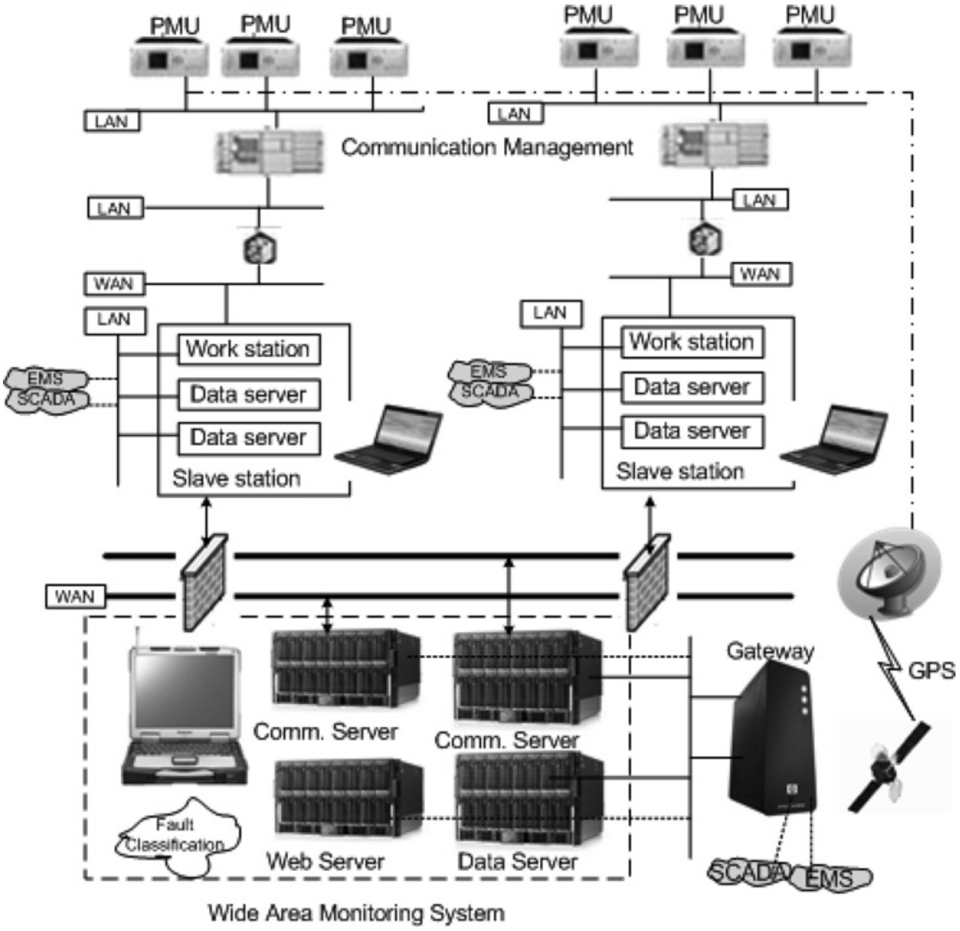


Figure 4 *Architecture of WAMS platform for large networks*

communication link depending upon the control action to be performed.

Some of the practical considerations for implementation are as follows.

1. *Complexity.* Conventional control is largely carried out locally. A wide-area control scheme introduces new interdependencies over the network that must be studied with care.

2. *Reliability.* There is a significant concern that dependence on communication systems introduces vulnerability, particularly where high-speed communication is required.

3. *Speed of response.* Some protection and control schemes that require a finite response with in a time are susceptible to communications and process delays.

Based on phasor measurement from both sides of transmission lines, WAMPCS monitors actual power flow, available transmission capacity, average temperature, thermal limits and so on, and initiates commands to local control equipment for improving the performance of the power system network and minimising congestion. It detects incipient instabilities and abnormal conditions, evaluates most suitable countermeasures and initiates automatic actions to protect power systems from large area disturbances. Automatic commands can be initiated to existing control systems (SCADA, EMS etc.) if those systems are able to perform them in due time. Otherwise, the commands are sent directly to PMUs or other control equipment in the stations. For the purpose of post-disturbance analysis, the WAMCP centre do not require to meet stringent requirements of the data networks such as real-time applications. Some delay can be tolerated during the retrieval of data on a regular schedule for the purpose of post-analysis.

Adaptive relays accept the change of their operating characteristics according to their power system conditions. The important characteristic that makes them vital to the adaptive relaying concept is that the relay parameters can be determined through the software and can be altered by communication equipment from its remote CC. Recently, with the use of combined synchronised PMUs and WAPS, adaptive protection systems can be effectively used to achieve better system security.

In this paper, a process with the knowledge base collected from WAMS using PSO–ANN is presented for ready post-fault diagnosis process immediately after the detection of fault. The approach is particularly important for post-fault diagnosis of any mal-operation of relays following a disturbance in the neighbouring line connected to the same substation/network. This may help in improving the fault monitoring/diagnosis process and coordination of the protective relays, thus assuring secure operation of the power systems. Nowadays, the traditional centralised SCADA solution with local protection systems are merging into a WAPS subject to new requirements with respect to their performance and availability. The different current signals measured with the use of PMUs at different locations and are sent to PDC at the CC. The PDC exports these measurements as a data stream as soon as they have been received and correlating the data by time-tag. The system protection centre receives data stream and makes a wide-area protection depending on the wide-area view. The proposed method can be used in WAPS to find the type of fault and pre-planned actions can be taken to minimise the wide-area disruptions and faulted section can be isolated from the healthy section. The proposed method makes the analysis of large fault disturbance data sets of WAMS. To apply this method in real time, automatic detection at the beginning of incidents is necessary. As soon as a disturbance has been detected, the recent samples of current signals are collected and taken into processing. The first goal of the approach is the identification of the type of fault occurred in the line. The approach is purely related to a substation-level power system network as shown in Fig. 5 and can be extended and implemented at CC of larger systems using WAMS as shown in Fig. 4.

The prototype system consists of a three-phase power supply, a transmission line represented by lumped parameters connecting a load. Faults were created at different places on transmission line [27].

Generator

Voltage rating: 400 kV, 50 Hz.

Total impedance of generator and transformer together: (0.2 + j 4.49) Ω.

Transmission line

- Positive and negative sequence resistance/unit length = 0.02336 Ω/km.

- Zero sequence resistance/unit length = 0.38848 Ω/km.

- Positive- and negative-sequence inductance/length = 0.95106 mH/km.

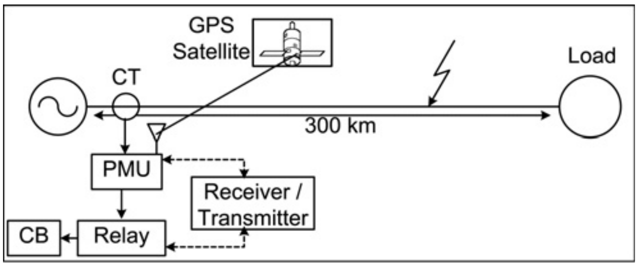


Figure 5 *Sample power system network*

- Zero-sequence inductance/length = 3.25083 mH/km.

- Positive- and negative-sequence capacitance/unit length = 12.37 nF/km.

- Zero-sequence capacitance/unit length =8.45 nF/km.

Load

Load impedance = (720 + j1.11) Ω corresponding to load of 200 MVA at 0.9 pf.

5 Implementation of the proposed fault classification

5.1 Pre-processing of the training and test patterns

Line currents I_a, I_b and I_c at a frequency of 50 Hz measured simultaneously by PMUs at the sending end of the line are used to classify the types of the fault among LG, LL, LLG, LLL and healthy (normal) conditions. The GPS clock generator can be used in the PMUs to provide an accurate and reliable external reference clock signal. These current signals are being decomposed into nine levels using the MRA algorithm. Since for N-level decomposition, 2^N samples are required. The current signals with 512 samples at a sampling period of $T_s = 7.828 \times 10^{-5}$ s are used in this work. The input contains 512 (12.77 kHz) samples which are passed through high pass filter (HPF) and low pass filter (LPF), and the corresponding approximate and detailed coefficients are recorded. The high-frequency noise signals are filtered and the corresponding seventh¯level detailed wavelet coefficients are calculated to know the second and third¯order harmonic components in the faulted current signals.

During fault condition, the detailed (HPF) coefficients of seventh level with the frequency band of 99–199 Hz have higher magnitudes owing to the presence of second and third-order harmonic content in the line currents. From the previous studies, it is found that Daubechies mother wavelet is having a good capability to capture the time of transient occurrence and the extraction of frequency features during power system faults. In the proposed algorithm, 'Db1' mother wavelet is used to get the DWT coefficients for the classification of different types of faults.

With the use of these detailed coefficients, various parameters i.e. S_a, S_b, S_c, Q_a, Q_b and Q_c are calculated,

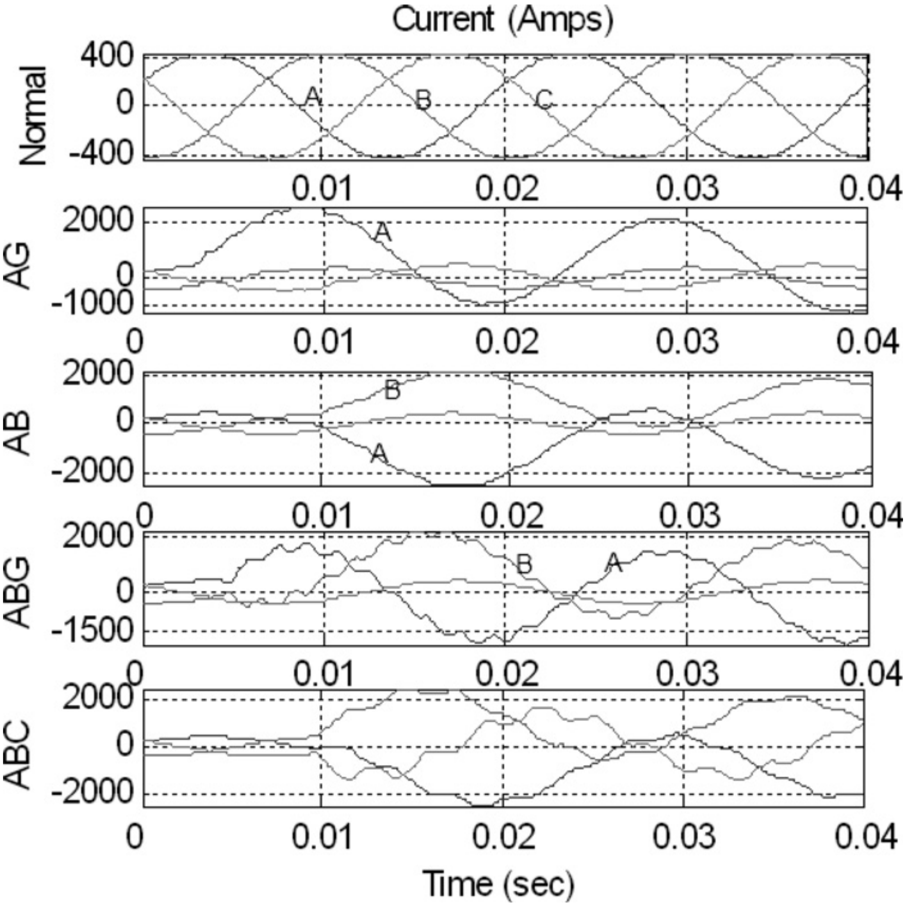


Figure 6 *Simulated line current signals measured at sending end*

where S_a, S_b and S_c are the sum of the seventh¯level detailed coefficients of line currents I_a, I_b and I_c, respectively, and Q_a, Q_b and Q_c the sum of the absolute values of the seventh-level detailed coefficients of line currents I_a, I_b and I_c, respectively.

For instance, the simulated fault current waveforms obtained for the various operating conditions at sending end for the faults types such as AG, AB, ABG and ABC including normal operating conditions are shown in Fig. 6. In this figure, the higher fault current magnitudes compared with the normal current signal during the corresponding fault can be observed. The x-axis in Fig. 6 represents the time in seconds and y-axis represents the instantaneous current signals in amperes.

The variation of indices Q_a, Q_b and Q_c and $|(S_a + S_b + S_c)|$ during LLG fault at different locations of the transmission line and various incidence angles with solid fault are depicted in Figs. 7a–7d.

5.2 Simulation of training and test cases

After calculating above parameters, the training sample of $|(S_a + S_b + S_c)|$, Q_a, Q_b and Q_c for various types of faults are given as input variables to build neural network. The data sets are created by considering different operating conditions, i.e. the different values of inception angles ranging between 0° and 360°, different values of fault resistances between 0 and 200 Ω and different fault distances from 0 to 300 km as follows:

1. Fault type: AG,BG,CG,ABG,BCG,CAG,AB,BC,CA,ABC.

2. Fault locations:

Training: 12,30, 48, 66, …, 282 km (in steps of 18 km).

Testing: 0.25, 5, 9.75, …, 300 km (in steps of 4.75 km).

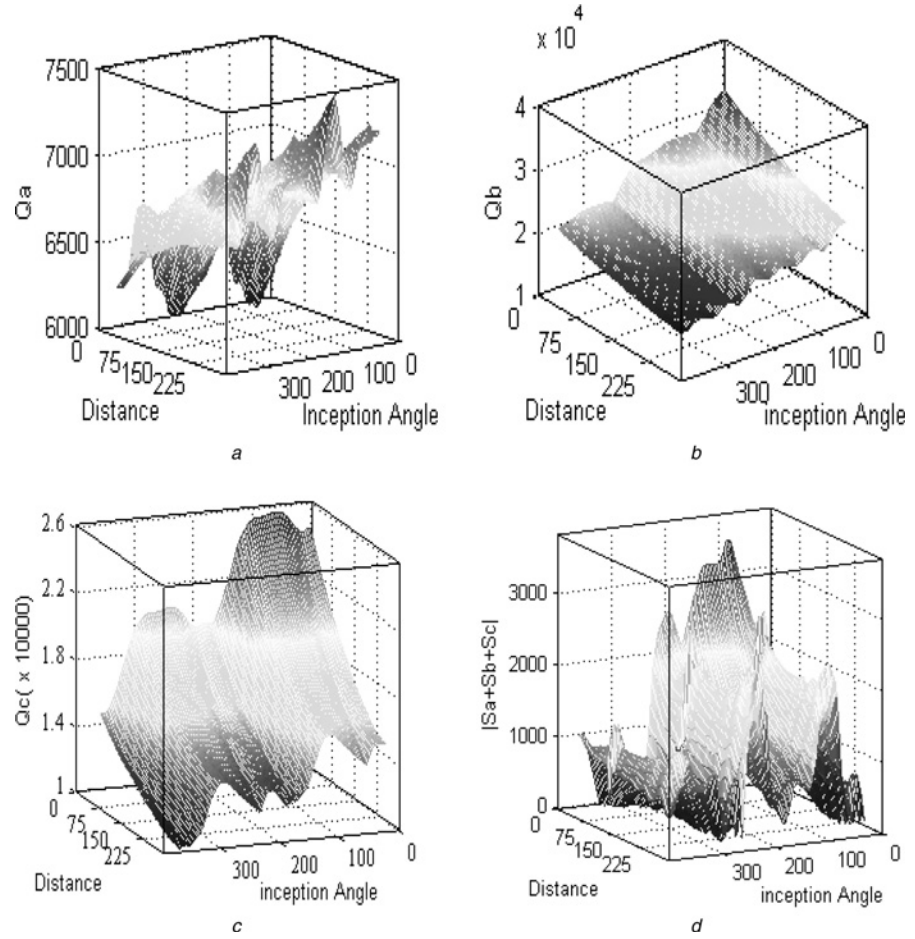


Figure 7 *Variation for different values of distances and inception angles during the LLG (BCG) fault*

a Q_a
b Q_b
c Q_c
d $|(S_a + S_b + S_c)|$

doi: 10.1049/iet-gtd.2009.0488

Table 1 Neural network desired output

Fault type	'A'	'B'	'C'	'G'
AG	1	0	0	1
BG	0	1	0	1
CG	0	0	1	1
ABG	1	1	0	1
BCG	0	1	1	1
CAG	1	0	1	1
AB	1	1	0	0
BC	0	1	1	0
CA	1	0	1	0
ABC	1	1	1	0

3. Fault inception angle:

   Training: 0°, 20°, 40°, 60°, …, 340° (in steps of 20°).

   Testing: 0°, 4°, 8°, 12°… 359° (in steps of 4°).

4. Fault resistance (Ω):

   Training: 0, 12, 24, …, 200 Ω (in steps of 12 Ω).

   Testing: 0, 10, 20, …, 200 Ω (in steps of 10 Ω).

Therefore during training of the fault data with ANN, the fault cases were carried out at 16 different locations with 18 different inception angles and with 17 various fault resistances. Thus a total of 16 × 18 × 17 = 4896 training

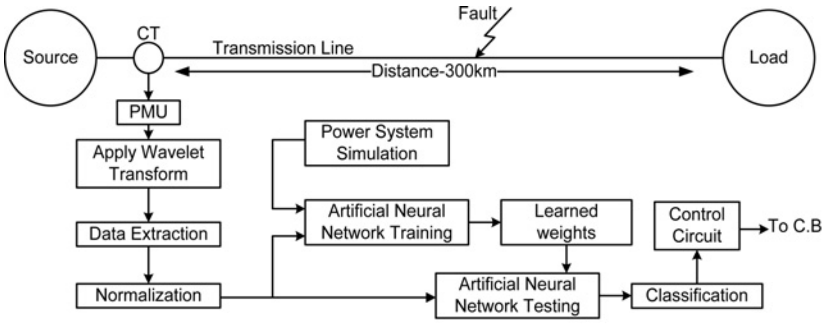


Figure 8 *Overview of proposed fault classification scheme using ANN*

Table 2 ANN output for various faults at 60° fault inception angle and at 66.75 km away from sending end

Fault type	Inception angle = 60°, fault distance = 66.75 km							
	$R_f = 0\ \Omega$				$R_f = 200\ \Omega$			
	'A'	'B'	'C'	'G'	'A'	'B'	'C'	'G'
AG	0.9064	0.0630	−0.0376	1.0237	1.0508	0.0610	0.0584	1.0896
BG	0.0150	0.9327	−0.0020	0.9900	−0.0126	0.9854	0.0551	1.0712
CG	0.0548	0.0367	0.8962	1.0275	−0.0619	−0.0206	0.9015	1.0951
ABG	0.9508	1.0664	−0.0061	1.0649	0.9251	0.9206	0.0438	1.0702
BCG	−0.0674	1.0025	0.9792	1.0759	−0.0686	1.0370	0.8928	0.8628
CAG	1.0572	0.0450	1.0148	1.0563	1.0745	−0.0506	0.9078	0.9317
AB	1.0185	0.9421	−0.0077	0.0162	1.0187	0.9283	−0.0416	−0.0422
BC	0.0409	0.9295	0.9419	0.0591	−0.0319	0.9726	0.8755	0.0145
CA	0.9882	0.0333	1.0345	−0.0453	1.0972	−0.0422	0.9616	0.0473
ABC	0.9342	1.0612	0.9963	−0.0132	0.9837	0.9401	0.9639	−0.0673

doi: 10.1049/iet-gtd.2009.0488

Table 3 ANN output for various faults at 135° fault inception angle and at 119 km away from sending end

Fault type	Inception angle = 135°, fault distance = 119 km							
	$R_f = 0\ \Omega$				$R_f = 200\ \Omega$			
	'A'	'B'	'C'	'G'	'A'	'B'	'C'	'G'
AG	1.0060	0.0254	−0.0169	1.0782	1.0549	0.0465	0.0004	0.9190
BG	0.0293	0.9690	−0.0100	1.0031	−0.0274	0.8953	−0.0434	1.0010
CG	−0.0429	0.0255	0.9255	0.9389	−0.0276	0.0058	0.9582	0.9815
ABG	0.9056	0.9448	−0.0489	0.9748	0.8821	1.0447	0.0277	0.9715
BCG	−0.0170	0.8969	0.9888	1.0033	0.0504	0.9666	1.0066	0.9899
CAG	0.8738	0.0495	0.8743	0.8987	0.9501	0.0560	0.8771	1.0862
AB	0.9482	1.0278	0.0131	0.0450	0.8999	1.0884	0.0203	0.0445
BC	0.0224	1.0346	0.9251	−0.0221	−0.0222	1.0136	1.0060	−0.0294
CA	1.0322	0.0048	0.9582	0.0318	1.0613	−0.0267	0.9020	0.0474
ABC	1.0262	0.9932	0.9738	0.0095	0.9684	0.9446	1.0090	−0.0181

samples are created for each fault resulting into a training data set of 4896 × 10 = 48 960 training samples for all ten types of faults.

The graphical view of training data set considered for training purpose is shown in Fig. 7. The training of neural network can be performed either using measurement data taken from the network or accessing historical record of fault data set. Whereas, in this paper, the training data set is generated by simulating MATLAB software program. For training purpose, the four outputs of ANN represented with 'A', 'B', 'C' and 'G' are assigned with values '1' for faulted phase and '0' for healthy phase as shown in Table 1.

In order to have a good accuracy of ANN, it is evaluated with more number of fault cases at different system conditions. Then, to test the proposed algorithm, the fault cases were carried out at 64 different locations with 90 different inception angles and with 21 various fault resistances. Thus, a total of 64 × 90 × 21 = 120 960 observations for each type of faults were simulated. As a result, total observations of 120 960 × 10 = 1 209 600

Table 4 NN output for various faults at 210° fault inception angle and at 180.75 km away from sending end

Fault type	Inception angle = 210°, fault distance = 180.75 km							
	$R_f = 0\ \Omega$				$R_f = 200\ \Omega$			
	'A'	'B'	'C'	'G'	'A'	'B'	'C'	'G'
AG	0.9286	0.0284	0.0065	0.9479	0.9133	−0.0077	0.0272	1.0035
BG	0.0170	0.9276	0.0413	0.9326	0.0640	1.0413	0.0032	1.0878
CG	0.0532	−0.0458	0.9545	1.0644	0.0672	−0.0320	0.8795	0.9293
ABG	1.0333	1.0300	−0.0347	1.0423	1.0839	0.9287	0.0526	1.0733
BCG	0.0332	1.0481	1.0323	1.0879	0.0022	0.9203	0.9949	0.8844
CAG	0.8869	0.0116	0.9966	0.9939	0.9147	−0.0232	0.9366	0.8757
AB	0.9664	0.9736	−0.0107	−0.0094	0.9679	0.8914	−0.0384	0.0112
BC	0.0365	1.0767	0.9669	0.0042	0.0197	1.0858	1.0683	−0.0407
CA	0.8635	−0.0168	0.8811	0.0397	1.0925	0.0253	0.9493	−0.0054
ABC	1.0194	1.0531	0.9664	0.0095	0.9454	1.0634	0.8777	0.0412

doi: 10.1049/iet-gtd.2009.0488

Table 5 ANN output for various faults at 330° fault inception angle and at 242.50 km away from sending end

Fault type	Inception angle = 330°, fault distance = 242.50 km							
	$R_f = 0\ \Omega$				$R_f = 200\ \Omega$			
	'A'	'B'	'C'	'G'	'A'	'B'	'C'	'G'
AG	1.0999	−0.0617	0.0144	0.9895	1.0229	−0.0630	0.0201	1.0410
BG	−0.0400	0.9292	0.0317	1.0096	0.0644	0.9064	−0.0252	1.0192
CG	−0.0123	0.0342	1.0183	1.0246	−0.0325	−0.0084	0.8871	1.0720
ABG	0.9297	1.0965	−0.0639	1.0226	0.8629	0.9263	0.0607	0.9253
BCG	−0.0068	0.8642	1.0045	1.0704	−0.0662	1.0450	1.0780	0.9607
CAG	1.0899	−0.0682	1.0052	0.8631	0.9839	−0.0262	0.8975	0.9111
AB	1.0440	1.0091	−0.0162	−0.0582	0.9700	1.0132	0.0256	−0.0570
BC	−0.0651	0.9945	0.9040	0.0157	0.0077	1.0968	1.0430	−0.0030
CA	0.8914	0.0319	1.0128	0.0279	0.9998	−0.0485	1.0332	−0.0205
ABC	0.8829	1.0573	0.9009	0.0246	0.9822	1.0874	1.0164	−0.0081

were made for all types of faults for testing purpose, whereas most of the training samples did not coincide with the testing data set.

5.3 Test results of PSO–ANN-based classifier

The overview of a proposed protective relaying solution based on PSO–ANN is shown in Fig. 8. On the occurrence of a fault, the fault detection unit activates the fault classification unit. The proposed technique is tested using simulated data obtained using MATLAB/SIMULINK® software [17]. The test cases generated various operating conditions are processed through PSO–ANN to test the proposed algorithm. While implementing PSO to train ANN, the inertia weight w which is linearly decreasing from 0.9 to 0.4, c_1 and c_2 are selected as 2 and the number of particles considered as $s = 40$. The end conditions to terminate PSO are (i) G_{best} should be below tolerance value 'ε' and (ii) the maximum number of iterations reached.

Tables 2–5 present some of the fault classification test results at various operating conditions. The performance of PSO–ANN for different fault types at 66.75 km of the line for 60° fault inception angle at different fault resistances $R_f = 0\ \Omega$ and $R_f = 200\ \Omega$ are shown in

Table 6 Summary of PSO–ANN, BPNN and SVM-based test results for nominal power system

S. no.	Faults type	PSO–ANN method		BPNN method [29]		SVM [30]
		Success	% Success	Success	% Success	% Success
1	AG	120 944	99.98	120 921	99.97	94.44
2	BG	120 927	99.97	120 892	99.94	96.29
3	CG	120 855	99.91	120 810	99.88	91.67
4	ABG	120 960	100.00	120 959	99.99	94.44
5	BCG	120 960	100.00	120 960	100.00	98.14
6	CAG	120 818	99.88	120 782	98.85	93.51
7	AB	120 746	99.82	120 677	99.77	98.14
8	C	120 960	100.00	120 960	100.00	100.00
9	CA	120 412	99.54	120 232	99.40	92.51
10	ABC	120 960	100.00	120 960	100.00	–
Total tested samples	1 209 600	1 208 542	99.91	1 208 153	99.88	96.01

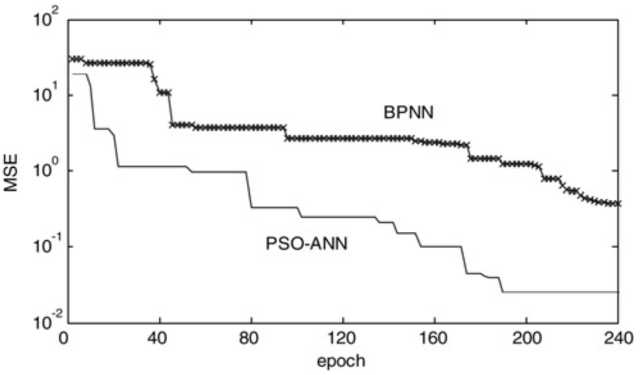


Figure 9 *Relationships between the normalised MSE and epochs during training for PSO-based ANN and BP-based perceptrons*

Table 2. The respective PSO–ANN output values 'A', 'B', 'C' and 'G' for AB fault type with $R_f = 0\,\Omega$ are 1.0185, 0.9421, −0.0077 and 0.0162 depict that the faults are involved in phases A and B only. This method takes a particular phase to be involved with fault if its corresponding PSO–ANN output value is near to '1', else it categorises the phase to be healthy. Similarly, Table 3 provides the fault classification results for various faults at 119 km of the line for 135° fault inception angle at different fault resistances. Table 4 shows the fault classification results for various fault conditions at 180.75 km of the line for 210° fault inception angle at different fault resistances, whereas Table 5 presents for 330° fault inception angle and a fault at 242.50 km of the line. The results demonstrate the suitability of the proposed method even for untrained test samples.

5.4 Comparison with the BPNN- and SVM-based classification scheme

The performance of PSO–ANN-based fault classifier for all types of faults is verified and the classification results are summarised as in Table 6. From Table 6, it is observed that the PSO–ANN provides a very good classification since the % success in fault classification varies from 99.54% to a maximum value of 100%, with 99.91% of average success including all type of operating conditions. The minimum value of % success is obtained as 99.54 while detecting CA faults, because some of these faults are detected as CAG fault and mismatch in detecting the ground involvement.

The BP under the neural network toolbox in MATLAB [28] software is employed as the benchmarking tool for comparison. In order to compare the proposed classification method with the BPNN method, the data sets that are already created during both the learning phase and testing phase are being used. In BPNN implementation [29], the mean square error (MSE) was set to 0.01 and the maximum number of times of training was set to 400. Bias weights and momentum factors are used to obtain optimum results in minimum time. The % success in fault classification using BPNN method is varies from 98.85% to a maximum value of 100%, with 99.88% of average success including all type of operating conditions. Similarly, the classification results for all types of faults (except ABC

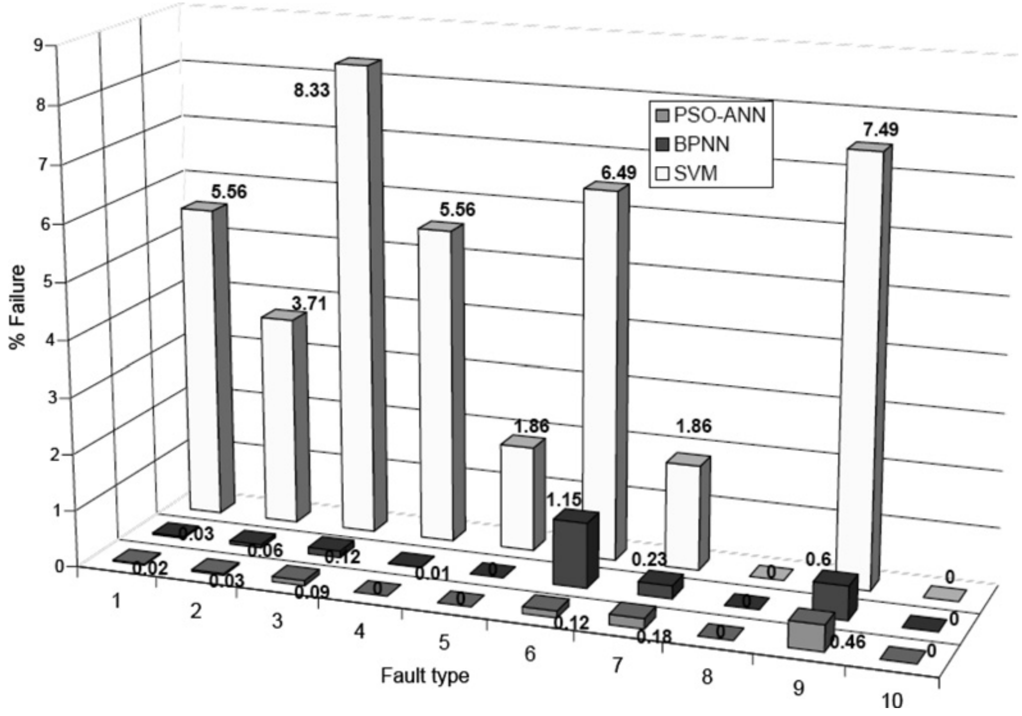


Figure 10 *Comparison of % failure in fault classification by applying PSO–ANN, BPNN and SVM methods*

doi: 10.1049/iet-gtd.2009.0488

faults) using SVM [30] classifiers are summarised in Table 6. It is observed that % success in fault classification with SVM-based method varies from 92.51 to a maximum value of 100%, with 96.01% as an average success rate.

Relationships between the normalised MSE and epochs during training of PSO-based ANN and BPNN method is shown in Fig. 9. Using the PSO–ANN method, it is found that the time taken for training the full training data set is approximately 1.462 s, and the time taken for testing the single-fault sample data is 0.036484 s. The BPNN method took 2.384 s to train the same data set. It is noted that testing cases of the PSO-based network are able to give a successful prediction rate of up to 99.91%, which is higher than BPNN (99.88%) and SVM (96.01%)-based methods. The percentage failure in classifying the testing data set using PSO–ANN and BPNN methods is shown in Fig. 10. It is observed that, with the use of PSO-based perceptron method, the highest failure of 0.18% is obtained when classifying AB fault, whereas 1.15% failure in classifying the CAG faults was obtained with the application of BP-based perceptron method. Moreover, the PSO-based perceptron exhibits much better and faster convergence performance in the training process as well as better classification than those by the BP-based perceptron. The PSO–ANN is a simple algebraic process, so it achieves the stable weights in less number of iterations and learns the new knowledge quickly compared with BPNN. It can be concluded that the PSO-based perceptron performs better than the SVM and BPNN-based perceptron. PSO–ANN method classifies fault type more perfectly during all operating conditions compared with BPNN and SVM. Moreover, the PSO–ANN network is able to identify the testing patters almost immediately.

6 Conclusion

This paper presents the application of a PSO-based perceptron approach for prediction of fault type with the help of WT. The optimisation algorithm is demonstrated to be able to provide model-free estimates in deducing the output from the input. Various case studies have been studied including the variation of fault distance, inception angle and fault resistances. Both PSO-based perceptron approach and BPNN method have been implemented. The performance shown demonstrates that the proposed technique gives a very high accuracy (99.912%) in classification of the power system faults. It is demonstrated from the training and verification simulation that the prediction results of fault type are more accurate, when compared with the conventional BP-based perceptron and SVM methods. Therefore the PSO-based perceptron approach can be used as an attractive and effective approach for classification algorithm of power system faults.

7 References

[1] ZHANG N., KEZUNOVIC M.: 'A real time fault analysis tool for monitoring operation of transmission line protective relay', *Electr. Power Syst. Res.*, 2007, **77**, pp. 361–370

[2] BEGOVIC M., NOVOSEL D., KARLSSON D., HENVILLE C., MICHEL G.: 'Wide-area protection and emergency control', *Proc. IEEE*, 2005, **93**, (5), pp. 876–891

[3] XIAO J., WEN F., CHUNG C.Y., WONG K.P.: 'Wide-area protection and its applications – a bibliographical survey'. IEEE PES Power Systems Conf. and Exposition, November 2006, pp. 1388–1397

[4] KAMWA R., HÉBERT Y.: 'Wide-area measurement based stabilizing control of large power systems – a decentralized/hierarchical approach', *IEEE Trans. Power Syst.*, 2001, **16**, (1), pp. 136–153

[5] DAUBECHIES I.: 'The wavelet transform, time-frequency localization and signal analysis', *IEEE Trans. Inf. Theory*, 1990, **36**, (5), pp. 961–1005

[6] DAUBECHIES I.: 'Ten lectures on wavelets' (SIAM, 1992)

[7] SIDHU T.S., SINGH H., SACHDEV M.S.: 'Design, implementation and testing of an artificial neural network based fault direction discriminator for protecting transmission lines', *IEEE Trans. Power Delivery*, 1995, **10**, pp. 697–706

[8] DALSTEIN T., FRIEDRICH T., KULICKE B., SOBAJIC D.: 'Multi neural network based fault area estimation for high speed protective relaying', *IEEE Trans. Power Deliv.*, 1996, **11**, pp. 740–747

[9] AGGARWAL R.K., XUAN Q.Y., DUNN R.W., JOHNS A.T., BENNETT A.: 'A novel fault classification technique for double-circuit lines based on a combined unsupervised/supervised neural network', *IEEE Trans. Power Deliv.*, 1999, **14**, pp. 1250–1256

[10] FITTON D.S., DUNN R.W., AGGARWAL R.K., JOHNS A.T., BENNETT A.: 'Design and implementation of an adaptive single pole autoreclosure technique for transmission lines using artificial neural networks', *IEEE Trans. Power Deliv.*, 1996, **11**, pp. 748–756

[11] SHEHAB-ELDIN E.H., MCLAREN P.G.: 'Travelling, wave distance protection-problem areas and solutions', *IEEE Trans. Power Deliv.*, 1998, **3**, (3), pp. 894–902

[12] GIRGIS A.A., JOHNS M.B.: 'A hybrid expert system for faulted section identification, fault type classification and selection

of fault location algorithms', *IEEE Trans. Power Deliv.*, 1996, **4**, (2), pp. 978–985

[13] JIANG J., CHEN C., LIU C.: 'A new protection scheme for fault detection, direction discrimination, classification, and location in transmission lines', *IEEE Trans. Power Deliv.*, 2003, **18**, (1), pp. 34–42

[14] OMAR A., YOUSSEF S.: 'Combined fuzzy-logic wavelet-based fault classification technique for power system relaying', *IEEE Trans. Power Deliv.*, 2004, **19**, (2), pp. 582–589

[15] PURUSHOTHAMA G.K., NARENDRANATH A.U., THUKARAM D., PARHASARATHY K.: 'ANN applications in fault locators', *Electr. Power Energy Syst.*, 2001, **23**, pp. 491–506

[16] MALLET S.G.: 'A theory for multi-resolution signal decomposition: the wavelet representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1989, **11**, (7), pp. 674–693

[17] SIMPSON P.K.: 'Artificial neural network: foundation, paradigms, applications and implementations' (Pergamon Press, 1990)

[18] GORI M., TESI A.: 'On the problem of local minima in back-propagation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1992, **14**, (1), pp. 76–86

[19] ZHANG J.-R., ZHANG J., LOK T.-M., *ET AL.*: 'A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training', *Appl. Math. Comput.*, 2007, **185**, pp. 1026–1037

[20] VAN OOYEN A., NIENHUIS B.: 'Improving the convergence of the back-propagation algorithm', *Neural Netw.*, 1992, **5**, (4), pp. 465–471

[21] JACOBS R.A.: 'Increased rates of convergence through learning rate adaptation', *Neural Netw.*, 1998, **1**, pp. 295–307

[22] WEIRS M.K.: 'A method for self-determination of adaptive learning rates in back propagation', *Neural Netw.*, 1991, **4**, pp. 371–379

[23] KENNEDY J., EBERHART R.: 'Particle swarm optimization'. Proc. 1995 IEEE Int. Conf. Neural Networks, Perth, 1995, pp. 1942–1948

[24] UPENDAR J., GUPTA C.P., SINGH G.K.: 'Discrete wavelet transform and genetic algorithm based fault classification of transmission systems'. National Power Systems Conf. (NPSC-2008), Indian Institute of Technology, Bombay, India, 16–18 December 2008, pp. 323–327

[25] JIANG J.-A., YANG J.-Z., LIN Y.-H., LIU C.-W., MA J.-C.: 'An adaptive PMU based fault detection/location technique for transmission lines, part I: theory and algorithms', *IEEE Trans. Power Deliv.*, 2000, **15**, pp. 486–493

[26] Working Group C-6, System Protection Subcommittee, IEEE PES Power System Relaying Committee; 'Wide area protection and emergency control'. Final report [online], 2002, http://www.pes-psrc.org/

[27] DAS D., SINGH N.K., SINGH A.K.: 'A comparison of Fourier transform and wavelet transform methods for detection and classification of faults on transmission lines'. IEEE Power India Conf., New Delhi, 2006

[28] Matlab/Simulink Toolbox: 'Neural network, wavelet transform', [online], www.mathworks.com

[29] UPENDAR J., GUPTA C.P., SINGH G.K.: 'ANN based power system fault classification'. IEEE Region-10 Int. Technical Conf. (TENCON-2008), Hyderabad, India, 20–23 November 2008

[30] BHALJA B., MAHESHWARI R.P.: 'Wavelet-based fault classification scheme for a transmission line using a support vector machine', *Electr. Power Compon. Syst.*, 2008, **36**, (10), pp. 1017–1030

doi: 10.1049/iet-gtd.2009.0488