A New Study in Maintenance for Transmission Lines

LIJIA REN, XIUCHEN JIANG, GEHAO SHENG, WU BO
Department of Electric Engineering
Shanghai Jiaotong University
Shanghai 200240
CHINA
ljren@sjtu.edu.cn http://www.sjtu.edu.cn

Abstract: - Because of increasingly common damage and thefts of the transmission towers, a new function is proposed to enhance the state maintenance of transmission lines. It relies on the monitoring of the high-voltage transmission towers to ensure the reliability and safety of the power grid operation. An approach which combines Independent Component Analysis (ICA) with neural network based on Particle Swarm Optimization (PSO) algorithm is presented. Its purpose is to extract the vibration source signals caused by the theft of the towers. Initially, the proposed algorithm separates the source vibration signals from the observed mixing signals based on the FastICA of negentropy and Exploratory Projection Pursuit (EPP). In order to distinguish the vibration pulse signal from other similar interference pulses, the algorithm of the feed forward neural network (FFNN) is used to identify the vibration pulses. It is trained online with particle swarm optimization (PSO), which incorporates the pulse extraction algorithm based on an adaptive threshold. Numerical results show that the algorithm is effective in extracting and identifying the vibration signals, and it also can suppress the interference signals.

Key-Words: - Transmission tower; ICA; Negentropy; Adaptive threshold; Particle swarm optimization (PSO); Neural network

1 Introduction

The monitoring system of the high-voltage transmission tower is a new feature of the state maintenance for transmission lines [1]. It is designed to prevent the materials of transmission tower from being destroyed or stolen. A three-axis acceleration sensor is installed on the transmission tower to measure the vibration due to motion of the tower during damage or theft. The system combines a common alarm system with a video monitor. The control system is sensitive to vibration caused by the motion of the tower. When the sensor detects theft-induced vibration signals, an alarm is broadcast, and local information can be transmitted by video communication in order to visually attest to the state of the transmission tower in question, and quickly make the correct decision.

The goal is to suppress the interference signals, detect and differentiate the signature vibration caused by human activity, especially the signals from sawing the transmission tower and knocking on it. Because of the electromagnetic interference around transmission towers and other interference signals, such as wind force and influence of the transmission line galloping, it is difficult to distinguish the vibration signals caused by human activity and suppress the background noise.

To address that issue, we extend the ICA model to the situation where noise is present. Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. It is applicable to the non-stationary vibration signals [2].

The technique of ICA, although not yet the name, was introduced in the early 1980s by J. H´erault, C. Jutten, and B. Ans [3]. The problem first came up in 1982 in a neurophysiological setting. In a simplified model of motion coding in muscle contraction, the outputs $x_1(t)$ and $x_2(t)$ were two types of sensory signals measuring muscle contraction, and $s_1(t)$ and $s_2(t)$ were the angular position and velocity of a moving joint. Then it is not unreasonable to assume that the ICA model holds between these signals. The nervous system must be somehow able to infer the position and velocity signals $s_1(t)$, $s_2(t)$from the measured responses $x_1(t)$, $x_2(t)$. One possibility for this is to learn the inverse model using the nonlinear decorrelation principle in a simple neural network. H´erault and Jutten proposed a specific feedback circuit to solve the problem.

All through the 1980s, ICA was mostly known among French researchers, with limited influence internationally. The few ICA presentations in

international neural network conferences in the mid-1980s were largely buried under the deluge of interest in back-propagation, Hopfield networks, and Kohonen's Self-Organizing Map (SOM), which were actively propagated in those times. Another related field was higher-order spectral analysis, on which the first international workshop was organized in 1989. Cardoso used algebraic methods, especially higher-order cumulant tensors, which eventually led to the JADE algorithm. The use of fourth-order cumulants has been earlier proposed by J.-L. Lacoume.

In signal processing, there had been earlier approaches in the related problem of blind signal deconvolution. In particular, the results used in multichannel blind deconvolution are very similar to ICA techniques.

However, until the mid-1990s, ICA remained a rather small and narrow research effort. Several algorithms were proposed that worked, usually in somewhat restricted problems, but it was not until later that the rigorous connections of these to statistical optimization criteria were exposed.

ICA attained wider attention and growing interest after A.J. Bell and T.J. Sejnowski published their approach based on the infomax principle in the mid-90's. This algorithmwas further refined by S.I. Amari and his co-workers using the natural gradient, and its fundamental connections to maximum likelihood estimation, as well as to the Cichocki-Unbehauen algorithm, were established. A couple of years later, the fixed-point or FastICA algorithm was presented , which has contributed to the application of ICA to large-scale problems due to its computational efficiency [2].

The interference signals on tower, or transmission line galloping caused by the wind, are sometimes similar to the theft-induced vibration pulses, both in time and frequency domains. Based on the typical characteristics of those different disturbance signals, the paper proposes an algorithm which combines a PSO-driven neural network with a pulse extraction algorithm based on an adaptive threshold. The result proves that the algorithm could extract and identify the useful vibration pulses effectively, and effectively suppress all other pulse interference.

2 Independent Component Analysis

2.1 The model of ICA

ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. The data variables of the model are assumed to be linear or nonlinear mixtures of the unknown latent variables, and the mixing mechanism is also unknown. The latent variables are assumed to be non-Gaussian and mutually independent, and they are called the independent components of the observed data [2].

We observe n random variables $x_1,\ldots,x_n$, which are modeled as linear combinations of n random variables $s_1,\ldots,s_n$:

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \cdots + a_{in}s_n \quad (4)$$

where the a_{ij}, i, j=1,…,n are some real coefficients. By definition, the s_i are statistically mutually independent.

This is the basic ICA model. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components s_j. The independent components s_j (often abbreviated as ICs) are latent variables, meaning that they cannot be directly observed. Also the mixing coefficients a_{ij} are assumed to be unknown. All we observe are the random variables xi, and we must estimate *both* the mixing coefficients a_{ij} *and* the ICs s_j using the x_j. This must be done under as general assumptions as possible.

ICA is very closely related to the method called blind source separation (BSS) or blind signal separation. A "source" means here an original signal, i.e., independent component, like the speaker in the cocktail-party problem. "Blind" means that we know very little, if anything, of the mixing matrix, and make very weak assumptions on the source signals. ICA is one method, perhaps the most widely used, for performing blind source separation.

It is usually more convenient to use vector-matrix notation instead of the sums as in the previous equation. An ICA model can be formulated as:

$$x = As \quad (1)$$

where A is an unknown $m \times n$ matrix, called the mixing matrix; x is an m-dimensional observed vector x=[x_1, x_2,…,x_n]^T; s is an n-dimensional (latent) unknown source vector s=[s_1, s_2,…,s_n]^T, in which components are as independent as possible. For the purpose of feature extraction, the columns of A represent features, and s_i is the coefficient of the i-th feature in the observed data vector x.

Sometimes we need the columns of matrix A; if we denote them by aj the model can also be written as

$$x = \sum_{i=1}^{n} a_i s_i \quad (2)$$

The definition given here is the most basic one. in many applications, it would be more realistic to

assume that there is some *noise* in the measurements, which would mean adding a noise term in the model

The independent components are assumed statistically independent. This is the principle on which ICA rests. Surprisingly, not much more than this assumption is needed to ascertain that the model can be estimated. This is why ICA is such a powerful method with applications in many different areas.

2.1.1 Whitening

A useful preprocessing strategy in ICA is to whiten the observed variables. A zero-mean random vector $z=(z_{1...}z_n)^T$ is said to be white if its elements z_i are uncorrelated and have unit variances:

$$E\{z_i z_j\} = \delta_{ij} \tag{3}$$

In terms of the covariance matrix, this obviously means that $E\{zz^T\}=I$, with I the unit matrix. The best-known example is white noise; then the elements z_i would be the intensities of noise at consequent time points i =1,2,…and there are no temporal correlations in the noise process. The term "white" comes from the fact that the power spectrum of white noise is constant over all frequencies, somewhat like the spectrum of white light contains all colors.

A synonym for white is sphered. Because whitening is essentially decorrelation followed by scaling, the technique of PCA can be used. This implies that whitening can be done with a linear operation. The problem of whitening is now: Given a random vector x with n elements, find a linear transformation V into another vector z such that $z=Vx$ is white (sphered).

The problem has a straightforward solution in terms of the PCA expansion. Let $E(e_1...e_n)$ be the matrix whose columns are the unit-norm eigenvectors of the covariance matrix $Cx=E\{xx^T\}$. These can be computed from a sample of the vectors x either directly or by one of the on-line PCA learning rules. Let $D=\text{diag}(d_1 \ldots d_n)$ be the diagonal matrix of the eigenvalues of C_x. Then a linear whitening transform is given by

$$V = D^{-1/2}E^T \tag{4}$$

This matrix always exists when the eigenvalues d_i are positive; in practice, this is not a restriction. Because C_x is positive semidefinite, in practice positive definite for almost any natural data, so its eigenvalues will be positive.

2.1.2 Orthonormal in ICA

In ICA algorithms, we know that in theory the solution vectors are orthogonal or orthonormal, but the iterative algorithms do not always automatically produce orthogonality. Then it may be necessary to orthogonalize the vectors after each iteration step, or at some suitable intervals.

given a set of n-dimensional linearly independent vectors $a_{1,}\ldots,a_m$, with $m \leq n$, compute another set of m vectors $w_{1,}\ldots,w_m$ that are orthogonal or orthonormal (i.e., orthogonal and having unit Euclidean norm) and that span the same subspace as the original vectors. This means that each w_i is some linear combination of the a_j.

The classic approach is the Gram-Schmidt orthogonalization (GSO) method:

$$w_1 = a_1$$

$$w_j = a_j - \sum_{i=1}^{j-1} \frac{w_i^T a_j}{w_i^T w_i} w_i \tag{5}$$

As a result, $w_i^T wj=0$ for $i \neq j$.

If in the GSO each w_j is further divided by its norm, the set will be orthonormal.

In symmetric orthonormalization methods, none of the original vectors a_i is treated differently from the others. If it is sufficient to find any orthonormal basis for the subspace spanned by the original vectors,without other constraints on the newvectors, then this problem does not have a unique solution. This can be accomplished for instance by first forming the matrix $A=(a_{1,}\ldots,a_m)$ whose columns are the vectors to be orthogonalized, then computing $(A^TA)^{-1/2}$ using the eigendecomposition of the symmetric matrix (A^TA), and finally putting

$$W=A(A^TA)^{-1/2} \tag{6}$$

Obviously, for matrix W it holds $W^TW=I$, and its columns $w_{1,}\ldots,w_m$ span the same subspace as the columns of matrix A. These vectors are thus a suitable orthonormalized basis. This solution to the symmetric orthonormalization problem is by no means unique; again, any matrix WU with U an orthogonal matrix will do quite as well [2].

2.2 FastICA algorithm

FastICA algorithm is an effective computational method for performing the estimation of ICA. It uses a fixed-point iteration scheme, which has been found faster than conventional gradient descent methods for ICA applications, as per [4]-[7]. Another advantage of the FastICA algorithm is that it can be used to perform projection pursuit as well, thus providing a general-purpose data analysis method that can be used both in an exploratory fashion and for estimation of independent components.

2.2.1 Negentropy measure criterion

In general, Kurtosis is usually as a measure of non-Gaussianity. However, kurtosis has some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is that kurtosis can be very sensitive to outliers. Kurtosis may depend on only a few observations in the tails of the distribution, which may be erroneous or irrelevant observations. In other words, kurtosis is not a robust measure of non-Gaussianity. Thus, other measures of nongaussianity might be better than kurtosis in some situations [2].

Negentropy is based on the information-theoretic quantity of differential entropy, which we here call simply entropy. Entropy is the basic concept of information theory. The entropy of a random variable is related to the useful information that the observation of the variable contains. The more "random", i.e., unpredictable and unstructured the variable is, the larger its entropy. The differential entropy H of a random vector y=[y_1, y_2,…, y_n]^T with density $p(y)$ is defined as

$$H(y) = -\int p(y) \lg p(y) dy \qquad (7)$$

A fundamental result of information theory is that a gaussian variable has the largest entropy among all random variables of equal variance. This means that entropy could be used as a measure of nongaussianity. In fact, this shows that the gaussian distribution is the "most random" or the least structured of all distributions. Entropy is small for distributions that are clearly concentrated on certain values, i.e., when the variable is clearly clustered, or has a pdf that is very "spiky".

To obtain a measure of nongaussianity that is zero for a gaussian variable and always nonnegative, one often uses a normalized version of differential entropy, called negentropy. Negentropy J is defined as follows

$$J(y) = H(y_{gauss}) - H(y) \qquad (8)$$

where y_{gauss} is a gaussian random vector of the same covariance matrix $\sum$ as y.

Due to the above-mentioned properties, negentropy is always nonnegative, and it is zero if and only if y has a Gaussian distribution. The property of Negentropy is robust, which is better than that of kurtosis. Furthermore, negentropy has the additional interesting property that it is invariant for invertible linear transformations.

The advantage of using negentropy, as a measure of non-Gaussianity is that it is well justified by statistical theory. In fact, negentropy is in some sense the optimal estimator of non-Gaussianity[2]. So we measure non-Gaussianity by kurtosis in this paper.

In practice, we only need approximation of 1-D entropies, so we only consider the scalar case here. The classic method of approximating negentropy is using higher-order cumulants, using the polynomial density expansions. Actually, this approximation often leads to the use of kurtosis. Therefore, we develop here more sophisticated approximations of negentropy.

In the case when we use only one non-quadratic function G, the approximation becomes

$$J(y) \propto [E\{G(y)\} - E\{G(y_{gauss})\}]^2 \qquad (9)$$

Where y_{gauss} is a Gaussian variable of zero-mean and unit variance, and $G(\cdot)$is a nonlinear and non-quadratic function.

In particular, choosing a G that does not grow too fast, one obtains more robust estimators [2]. The following choices of G have proved very useful:

$$G_1(y) = \frac{1}{a} \log \cosh a_1 y, \qquad (10)$$

$$G_2(y) = -\exp(-y^2/2) \qquad (11)$$

Where $1 \le a \le 2$ is some suitable constant, often taken equal to one. Thus we obtain approximations of negentropy that give a very good compromise between the properties of the two classic nongaussianity measures given by kurtosis and negentropy. They are conceptually simple, fast to compute, yet have appealing statistical properties, especially robustness.

2.2.2 Exploratory Projection Pursuit (EPP)

Projection pursuit is a technique developed in statistics for finding "interesting" projections of multidimensional data.

It is usually argued that the gaussian distribution is the least interesting one, and that the most interesting directions are those that show the least gaussian distribution. One motivation for this is that distributions that are multimodal, i.e., show some clustering structure, are far from gaussian [2].

An information-theoretic motivation for nongaussianity is that entropy is maximized by the gaussian distribution, and entropy can be considered as a measure of the lack of structure. This is related to the interpretation of entropy as code length: a variable that has a clear structure is usually easy to code. Thus, since the gaussian distribution has the largest entropy, it is the most difficult to code, and therefore it can be considered as the least structured.

2.2.3 A fast fixed-point algorithm using negentropy

The FastICA algorithm using negentropy finds a direction, i.e., a unit vector *w*, such that the projection $w^T z$ maximizes non-gaussianity. More rigorously, it can be derived as an approximate Newton iteration. The Newton method is a fast method for solving equations. When it is applied on the gradient, it gives an optimization method that usually converges in a small number of steps. The problem with the Newton method, however, is that it usually requires a matrix inversion at every step. Therefore, the total computational load may not be smaller than with gradient methods. What is quite surprising is that using the special properties of the ICA problem, it can been found an approximation of the Newton method that does not need a matrix inversion but still converges roughly with the same number of iterations as the real Newton method. This approximative Newton method gives a fixed-point algorithm of the form (12).

$$(1+\alpha)w = E\{zg(w^T z)\} + \alpha z \qquad (12)$$

Where α is constant.

The FastICA algorithm using negentropy combines the superior algorithmic properties resulting from the fixed-point iteration with the desirable statistical properties due to negentropy. We choose a nonlinearity g, which is the derivative of the nonquadratic function *G* used in equation (9), which give robust approximations of negentropy.

In general, we can assume that both mixture variables and the independent components have zero mean. This assumption substantially simplifies the theory and algorithms. If the assumption of zero mean is not true, preprocessing can be applied to make it hold. This is accomplished by centering the observable variables, i.e., subtracting their sample mean.

It is useful to whiten the observed variables. The observed vector x is linearly transformed into a new white vector *z* whose covariance matrix is the identity matrix: $E\{zz^T\}=I$. Whitening reduces the number of parameters to be estimated. Instead of having to estimate the n^2 parameters which are elements of the original matrix, we only need to estimate the new, orthogonal mixing matrix. In large dimensional situations, an orthogonal matrix contains only about half of the number of parameters of an arbitrary matrix.

Finally, we solve equation (9) by Newton's method, and get the fixed-point iterative algorithm for ICA

$$w_i(k+1) = E\{zG'[w_i^T(k)z]\} - E\{G''[w_i^T(k)z]\}w_i(k) \qquad (13)$$

To estimate several independent components, we need to run any of the one-unit algorithms several times (possibly using several units) with vectors $w_1, \ldots, w_m$. To prevent different vectors from converging to the same maxima, it is necessary to orthogonalize the vectors $w_1, \ldots, w_m$ after every iteration.

Based on *w*, the independent signal s_i is obtained by $y_i(k)=w_i^T z$.

From the preceding calculation, the FastICA algorithm can be described as follows :

(1) Center the data to make its mean zero, and whiten the data to provide z.

(2) Choose N, the number of ICs. Set counter $p \leftarrow 1$.

(3) Choose an initial (e.g., random) vector w of unit norm.

(4) Let $w_p \leftarrow E\{zG'[w_p^T z]\} - E\{G''[w_p^T z]\}w_p$.

(5) Do the following orthogonalization:

$$w_p \leftarrow w_p - \sum_{j=1}^{p-1} \langle w_p, w_j \rangle w_j$$

(6) Normalize:

$$w_p \leftarrow \frac{w_p}{\|w_p\|}.$$

(7) If convergence is not achieved, go back to step (4).

(8) Let $p \leftarrow p+1$ (increment the counter). If *p* is not greater than the desired number of independent components N, go back to step (3).

(9) Based on *w*, the independent signal s_j is obtained by $y_i(k) = w_i^T z$.

The FastICA algorithm has a couple of properties that make it clearly superior to the gradient-based algorithms in most cases. First of all, the convergence of this algorithm is cubic. This means very fast convergence. Second, contrary to gradient-based algorithms, there is no learning rate or other adjustable parameters in the algorithm, which makes it easy to use, and more reliable [2].

3 Recognition of the vibration pulse

It has been shown that neural networks compared well to statistical classification methods in classification of multisource remote sensing data and very-high-dimensional data [8][9]. The neural network models were superior to the statistical methods in terms of overall classification accuracy of training data.

3.1 Neural network learning algorithms based on Particle Swarm optimization

Particle Swarm optimization (PSO) is a search method for optimization of nonlinear functions based on the group behavior similar to that of a bird flock or a fish school [10]. A swarm consists of a set of particles, where each particle represents a potential solution. The changes to the position of a particle and its operation in a swarm are influenced by the experience and the knowledge of its neighbors.

It has been shown that the PSO algorithm may reduce the training time of neural network, and doesn't have the disadvantages of the back-propagation (BP) algorithm, such as slow convergence and vulnerability with respect to convergence towards the local minima [11]. Its key concept is that potential solutions are moved through the hyperspace (search space of the possible solutions) and the search is accelerated towards better, or more optimal, solutions. The search is more thorough when each particle (solution candidate) is directed towards exploring the new regions of the search space.

Initially a set of random solutions or a set of particles are considered. A random velocity is given to each particle and they are flown through the problem space. Each particle has memory which is used to keep track of the previous best position and corresponding fitness. The best value of the position of each individual is stored as 'p_{id}'. In other words, 'p_{id}' is the best position acquired by an individual particle during the course of its movement within the swarm. It has another value called the 'p_{gd}', which is the best value of all the particles 'p_{id}' in the swarm. The basic concept of the PSO technique lies in accelerating each particle towards its 'p_{id}' and 'p_{gd}' locations at each time step. Fig. 1 illustrates briefly the concept of PSO [10].

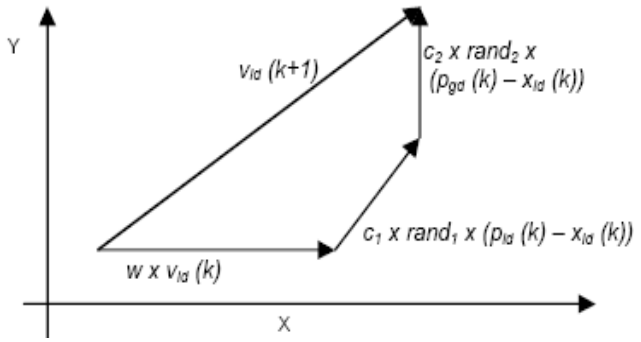

Fig. 1 PSO particle update process illustrated for a two dimensional case

where, x_{id} *(k)* is the current position of i^{th} particle with d dimensions at instant *k;* x_{id} *(k+1)* is the position of *ith* particle with d dimensions at instant *k+1*; v_{id} *(k)* is the initial velocity of *ith* particle with d dimensions at instant *k*; v_{id} *(k+1)* is the velocity of i^{th} particle with d dimensions at instant *k+1; w is* inertia weight; *Vmax* is the maximum velocity for the particles; c_1 is the cognition acceleration constant; c_2 is the social acceleration constant.

The PSO algorithm can be described as follows :

(1) Initialize a population of particles with random positions and velocities in the problem space;

(2) For each particle, evaluate the desired optimization fitness function.

(3) Compare the particles fitness evaluation with the particles p_{id}. If current value is better than the p_{id} then set p_{id} value equal to the current location.

(4) Compare the best fitness evaluation with the population's overall previous best. If the current value is better than the p_{gd}, then set p_{gd} to the particle's array and index value.

(5) Update the particle's velocity and position according to the equations shown below.

The velocity of the i^{th} particle of *d* dimension is given equations (7).

$$v_{id}(k+1) = wv_{id}(k) + c_1 rand_1(p_{id}(k) - x_{id}(k)) + c_2 rand_2(p_{gd}(k) - x_{id}(k)) \quad (14)$$

The position vector of the i^{th} particle of *d* dimension is updated based on equations (7).

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad (15)$$

(6) Repeat the step (2) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations or epochs.

In case the velocity of the particle exceeds V_{max} then it is reduced to V_{max}. Thus, the resolution and fitness of search depends on the V_{max}. If V_{max} is too high, then particles will move in larger steps and so the solution reached may not be as good as expected. If V_{max} is too low, then particles will take a long time to reach the desired solution [10].

Let us define particle swarm's positions. An element of vector $\vec{x}$ is the connection power value and the threshold value between BP neural network levels. The adaptation value function is chosen to be the mean square error target of the BP neural network, the formula is:

$$J = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C}(t_{ik} - p_{ik})^2 \quad (16)$$

where t_{ik} and p_{ik} are the target output and predicted values, respectively; N is the number of training samples; and C is the number of output neurons. The neural network is trained by minimizing the

above adaptation value function in the search space of weighting factors (parameters of the neural network).

3.2 The pulse extraction algorithm based on the adaptive threshold

The background noise and some other interference signals are variable and change frequently, and the vibration pulses caused by different people, or occurring in different locations on towers will be different. It is therefore crucial to suppress the interference and extract the integrity pulse signals to enhance the reliability of identification. Compared with the method based on hard-threshold, the advantage of adaptive pulse extraction algorithm is to confirm the onset of every pulse and extract each integrity pulse in consideration of the background noise, amplitude and minimum width of the pulse.

Because the extraction algorithm is in the threshold-adaptive mode, it could extract each integrity pulse and preserve the original waveform's characteristics. It is a very important part of the noise suppression process.

3.3 Recognition experiments

Artificial Neural Networks (ANN) is computational model that try to emulate the behavior of the human brain [12][13]. They are based on a set of simple processing elements, highly interconnected, and with a massive parallel structure. ANNs are characterized by their learning, adapting and generalization capabilities, which make them particularly suited for tasks such as function approximation.

Feed-Forward Neural Networks (FFNN) is a special class of ANNs [14], in which all the nodes in some layer *l* are connected to all the nodes in layer *l*-1 (shown in Fig.2). Each neuron receives information from all the nodes in the previous layer and sends information to all the nodes in the following layer. A FFNN is composed of the input layer, which receives data from the exterior environment, typically one hidden layer (though more layers may be used) and the output layer, which sends data to the exterior environment.

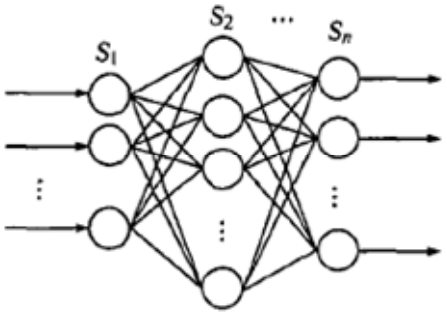

Fig.2 Architecture of the multilayer feed-forward neural networks

The links connecting each pair of neurons are given some weight. This attribution of weights to links is the job of any training algorithm. Each neuron computes an output value based on the input values received the weights of the links from the neurons in the previous layer and the neuron's transfer function. Usually, sigmoid functions are used. The capability of the FFNN for mapping input values into output values depends on the link weights. Their optimal determination is still an open problem. Therefore, iterative hill-climbing algorithms are used. Their main limitation comes from the fact that only local optima are obtained: only occasionally the global optimum can be found. In the context of ANNs, these iterative optimization algorithms are called training algorithms

Here, we used PSO-based feed forward neural network algorithm equipped with the adaptive pulse extraction algorithm for canceling the noise interference and identifying the required vibration pulse. The flowchart is shown in Fig.3. The characteristics of the extracted pulses can be classified into four categories, such as the shape of the pulse waveforms, frequency, energy, and correlation coefficient of the signal [15][16]. To classify the pulses, 22 characteristics have been chosen as input parameters. Accordingly, the three output layer nodes of the neural network represent the pulse categories. To achieve flexibility of identification, ten hidden layers have been adopted for the model of FFNN.

615 samples are used for the pulse analysis. A total of 615 samples were divided into two groups:

Table 1 Recognition results comparison of training algorithms

Testing sample numbers (Pulse classes)	Recognition numbers		Testing sample accuracy	
	PSO-based	BP	PSO-based	BP
156 (by human-generated force)	156	149	100%	93.6%
90 (by transmission line galloping)	88	82	97.7%	91.1%
120 (noise pulses)	118	109	96.6%	90.8%

249 training samples and 366 samples in the classification procedure. Set particle swarm optimization parameters as follows: the size of the swarm is 25, and the accelerarion constants are c_1= c_2 = 2. The inertia weight decreases from 0.8 to 0.2 linearly. The maximum velocity is chosen to be V_{max} = 2. Table 1 shows comparative analysis of the recognition results obtained by training with PSO and BP neural networks, respectively.

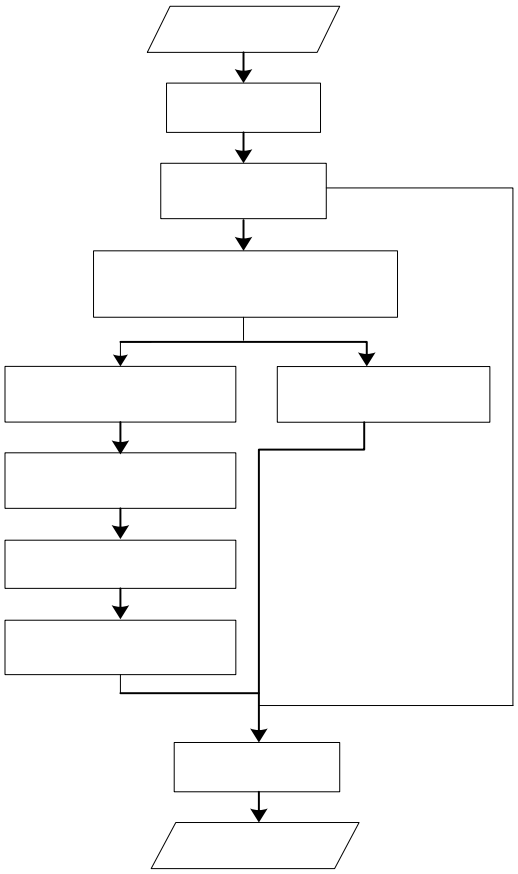

Fig.3 Flowchart for recognition of the vibration pulse using PSO-based FFNN with an adaptive pulse extraction algorithm

It is noted that the identification precision using PSO-based FFNN algorithm is more than 96%, while that of the BP algorithm is about 91%. Furthermore, the algorithm is a suitable tool for fast and adaptive real-time identification because of its optimization efficiency and global convergence rate.

4 Test and Analysis

In the anti-theft monitoring system of transmission tower, a three-axis accelerometer sensor is installed on the transmission tower to identify and measure the vibration caused by the application of a typical human-generated force consistent with the theft-related activities. In the test, "knock on" and "saw" actions were applied to different locations on the tower. Examples include application of the reinforcing steel bar under the sensor, adjacent angle iron and the angle iron on the cross, etc. Fig.4 shows the observed signal along the two accelerometer axes (*x*-axis and *y*-axis) in the field test.

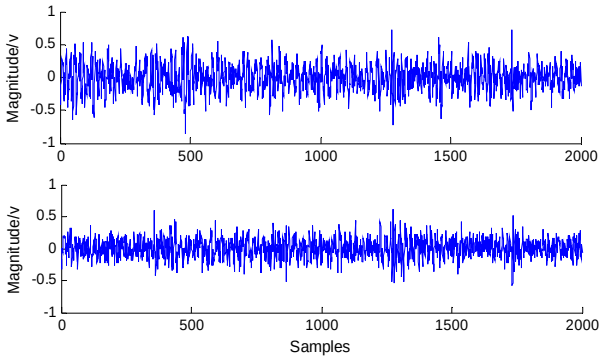

Fig.4 Waveforms of the observed signals

The result of the separation with FastICA is depicted in Fig.5.

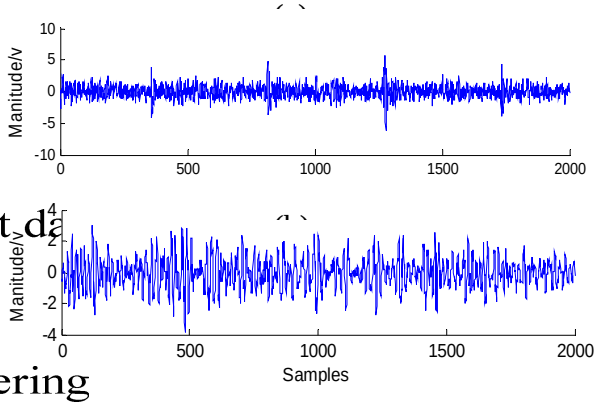

Fig.5 Source separation based on ICA. (a) The reconstructed knocking signal. (b) The reconstructed sawing signal.

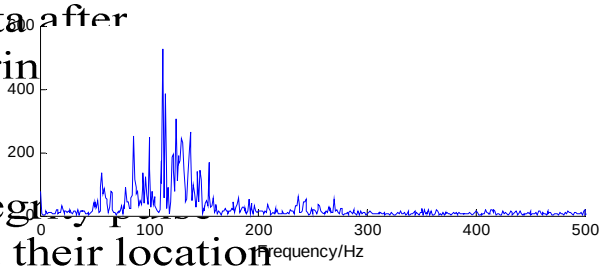

Fig.6 The spectrum of the vibration signal shown in Fig.5 (b)

From the Fig.5 we can see that the independent components (the sawing signal and the knocking signal) could be separated successfully. Though knocking signal has some interference in Fig.5 (a), it possesses distinct peak value in the time domain, while there is no evident change in the frequency domain except for the rising energy in the low frequency end of the spectrum. The sawing signal possesses high amplitude, and vibrates evidently. The sawing signal in the test has an obvious peak

value between 100Hz-150Hz in the frequency domain (Fig.6), and it is consistent with the result tested in the lab where there was no interference present in the signal. Though the amplitude of the waveform could change after processed by the ICA, the signal can still be extracted successfully. The practical tests demonstrate that FastICA is effective in solving the Blind Source Separation (BSS), and is an efficient computation method.

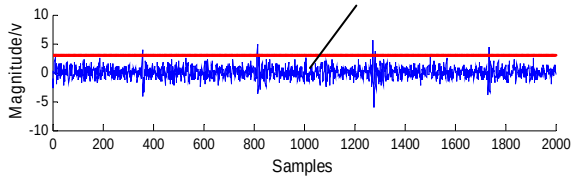

Fig.7 Independent source signal separated by ICA and the computed threshold

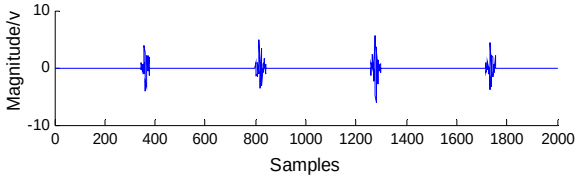

Fig.8 The result using PSO-based FFNN algorithm

Let us now apply the pulse extraction algorithm based on the adaptive threshold to the separated signal shown in Fig.5 (a), and obtain the computed threshold shown in Fig.7.

The pulse extraction algorithm and the PSO algorithm are used to process the data shown in Fig.7. The noise pulses can be identified according to the pulse classes and be eliminated finally. Fig.8 shows the results after suppressing the noise pulses. The result shows that the PSO algorithm achieves good performance in real time identification and suppresses most noise pulses effectively. The processed result is in accordance with the test.

5 Conclusion

The proposed theft monitoring system separates the vibration signals successfully from the saught signature signals (knocking and sawing) thanks to the FastICA using negentropy. The paper proposes an adaptive threshold pulse extraction method, which is capable of effectively extracting the integrity pulses. Following that, and according to the characteristics of the identified pulses, the PSO-based FFNN algorithm classifies the pulses, and suppresses the noise in the observed signals.

Field tests have been used to demonstrate that the method proposed in the paper is effective in separating, extracting and classifying the vibration signals. The system is therefore capable of detecting the desired classes of signals accurately, monitor the actions of the people under the tower and differentiate it from the measured signals of galloping of the transmission lines. The proposed monitoring scheme could enhance the reliability and safety of the power grid operation, and it would be a welcome feature of the state maintenance for transmission lines.

References:

[1] An Ling, Jiang Xiuchen, Zhu Yu, Object-oriented design and implementation of the state maintenance software system for transmission lines, Automation of Electric Power Systems, Vol.26, No,7, 2002, pp. 66-69.

[2] Aapo Hyvarinen, Juha Karhunen, Erkki Oja, Independent Component Analysis, Jone Wiley & Sons, Inc., New York, 2001.

[3] B. Ans, J. H´erault, and C. Jutten, Adaptive neural architectures: detection of primitives, In Proc. of COGNITIVA'85, Paris, 1985, pp. 593–597.

[4] Lee Te-Won, Girolami Mark, and Sejnowski Terrence J, Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources, Neural Computation, 1999, Vol.11, No.2, pp. 417-441.

[5] Yang Fusheng, Hong Bo, Principals and application of independent component analysis. Beijing Tsinghua University Press, 2006.

[6] A. Hyvarinen, E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Computation, Vol.9, No.7, 1997, pp. 1483-1492.

[7] A. Hyvarinen, Fast and robust fixed-point algorithm for independent component analysis, IEEE transaction on neural network, vol.10, No.3,1999, pp.626-634.

[8] Benediktsson J.A., Sveinsson J.R., Ersoy O.K., Parallel consensual neural networks, IEEE International Conference on Neural Networks, 1993, Vol.1, pp,27-32.

[9] Munjal Geetika, Kaur Sunint, Comparative study of ANN for pattern classification, *WSEAS Transactions on Computers*, Vol.6, No.2, 2007, pp. 236-241.

[10] Kiran R., Jetti S.R., Venayagamoorthy G.K, Online training of a generalized neuron with particle swarm optimization, International Joint Conference on Neural Networks, 2006, pp. 5088-5095.

[11] V. G. Gudise, G. K. Venayagamoorthy, Comparison of particle swarm optimization and backpropagation as training algorithms for

neural networks, IEEE on Swarm Intelligence Symposium, 2003, pp.110-117.
[12] Qian Ding, Man Zhang, Classification of recorded musical instruments sounds based on neural networks, IEEE Symposium on Computational Intelligence in Image and Signal Processing, 2007, pp.157-162.
[13] Watanapa Bunthit, Chan Jonathan H., Neural network classification of extended control chart patterns, WSEAS Transactions on Computers, Vol.6, No.1, 2007, pp.160-166.
[14] Kelemen Arpad, Liang, Yulan, Franklin Stan, Learning high quality decisions with neural networks in "conscious" software agents, WSEAS Transactions on Systems, 2005, Vol. 4, No.9, pp. 1482-1492.
[15] Chenjun Huang, Zhenyu Shao, Yakui Zhao, Application of neural network to suppress PD stochastic pulse noise for large generators, Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference, 2003, pp. 21-25.
[16] Shao Zhenyu, Huang Chengjun, Xiao Yan, Application of PSO based neural network in suppression of stochastic pulse interference for partial discharge monitoring in large gererators, Automation of Electric Power Systems, Vol.29, No.11,2005, pp. 49-52.