Multiple Fault Diagnosis in Electrical Power Systems with Dynamic Load Changes using Probabilistic Neural Networks

Juan Pablo Nieto González[1], Luis E. Garza Castañón[2], Rubén Morales Menendez[3]

I.T.E.S.M. Saltillo Campus Mechatronics Department
[1]Prol. Juan de la Barrera No. 1241 Ote. Colonia Cumbres,
25020, Saltillo, Coahuila, México
juan.pablo.nieto@itesm.mx

I.T.E.S.M. Monterrey Campus
[2]Mechatronics and Automation Department, [3]Center for Industrial Automation
Ave. Eugenio Garza Sada No. 2501 Sur, Colonia Tecnológico
64849, Monterrey, Nuevo León, México
legarza@itesm.mx, rmm@itesm.mx

(Paper received on June 22, 2007, accepted on September 1, 2007)

Abstract. Power systems monitoring is particularly challenging due to the presence of dynamic load changes in normal operation mode of network nodes, as well as the presence of both continuous and discrete variables, noisy information and lack or excess of data. This paper proposes a fault diagnosis framework that is able to locate the set of nodes involved in multiple fault events and detects the type of fault in those nodes. The framework is composed of two phases: In the first phase a probabilistic neural network is trained with the eigenvalues of voltage data collected during symmetrical and unsymmetrical fault disturbances. The second phase is a sample magnitude comparison used to detect and locate the presence of a fault. A set of simulations is carried out over an electrical power system to show the performance of the proposed framework and a comparison is made against a diagnostic system based on probabilistic logic.

1 Introduction

As processes become more complex, the monitoring of them is very important in order to improve process performance, efficiency and product quality. Monitoring of industrial processes plays a substantial role in system safety, availability and production quality. Early detection of faults can help to avoid major breakdowns and incidents. In order to tackle those problems, fault detection and system diagnosis has been an active research domain since years.

There exist a lot of research works related with fault detection. Most of the methods used are analytic, based on artificial intelligence (AI) or statistical methods. [1] classifies fault detection and isolation methods in three groups. 1)Quantitative Model Based, 2) Qualitative Model Based and 3) Process History Based.

© H. Sossa, R. Barrón, and E. Felipe (Eds.)
Special Issue in Neural Networks and Associative Memories
Research in Computing Science 28, 2007, pp. 139-150

Quantitative Model Based fault detection methods are based on a mathematical model of the system. The occurrence of a fault is captured by discrepancies between the observed behavior and the one that is predicted by the model. These approaches make use of state estimation, parameter identification techniques, and parity relations to generate residuals. Fault localization then, rest on interlining the groups of components that are involved in each of the detected discrepancies. However, it is often difficult and time-consuming to develop accurate mathematical models that characterize all the physical phenomena occurring in industrial processes.

Qualitative Model Based fault detection methods use symbolic reasoning which generally combines different kinds of knowledge with graph theory to analyze the relationships between variables of a system. An advantage of these methods is that an explicit model of the system to be diagnosed is not necessary. Knowledge-based approaches such as expert systems may be considered as alternative or complementary approaches where analytical models are not available.

Process History Based fault detection methods only require a big quantity of historical process data. There are several ways in which these data can be transformed and presented as prior knowledge of a system. These transformations are known as feature extraction and could be qualitative, as those used by expert systems, and qualitative trend analysis methods or quantitative, as those used in neural networks, PCA, PLS or statistical pattern recognition.

The reasons behind the increased interest in fault diagnosis in power networks are the complexity and high degree of interconnection present in electrical power networks, that can lead to an overwhelming array of alarms and status messages being generated as a result of a disturbance. This can have a negative impact on the speed with which operators can respond to a contingency. Therefore, in order to increase the efficiency of diagnosis, it is necessary to use automated tools, which could help the operator to speed up the process. Very recently, the need to develop more powerful approaches has been recognized, and hybrid techniques that combine several reasoning methods start to be used [2]. This approach incorporate model based diagnosis and signal analysis with neural networks. [3] presents Bayesian networks (BNs) to estimate the faulty section of a transmission power system. Simplified models of BNs with Noisy-Or and Noisy-And nodes are proposed to test if any transmission line, transformer, or busbar within a blackout area is faulty. In [4] an investigation is performed about the use of logistic regression and neural networks to classify fault causes. This paper also discusses about data insufficiency, imbalanced data constitution and threshold setting. Ren and Mi [5] propose a procedure for power systems fault diagnosis and identification based on Petri Nets and coding theory. They tested the approach with simulations over the IEEE 118-bus power system and highlight the great advantage to handle very easily future expansions. In [6] a Fault diagnosis system is presented, based on multi-agent systems. By using a negotiation mechanism between decision-making agent and a cooperative agent, fault diagnosis results can be obtained.

In this paper it is proposed a multiple fault diagnosis framework composed by two phases. Eigenvalues are computed from the correlation matrix which is built from historical data, and then are used as a probabilistic neural network inputs. In first phase a most probably component state of each node is given and in second phase the comparison of each sample against a constant value gives the real component state and the location of a fault, finally the diagnosis of the system is carried out.

The organization of the paper is as follows: section 2 explains probabilistic neural networks basis and gives the correlation matrix and eigenvalues definitions. Section 3 gives the framework general description. Section 4 shows how the framework works in a simulation example with single and multiple faults as well as a comparison of the general performance of it against a diagnostic system based on probabilistic logic. Section 5 concludes the paper.

2 Preliminary

2.1 Probabilistic Neural Network Basis

PNN are conceptually similar to K-Nearest Neighbor (KNN) models [7]. The basic idea is that a predicted value of an item is likely to be about the same as other items that have close values of the predictor variables.

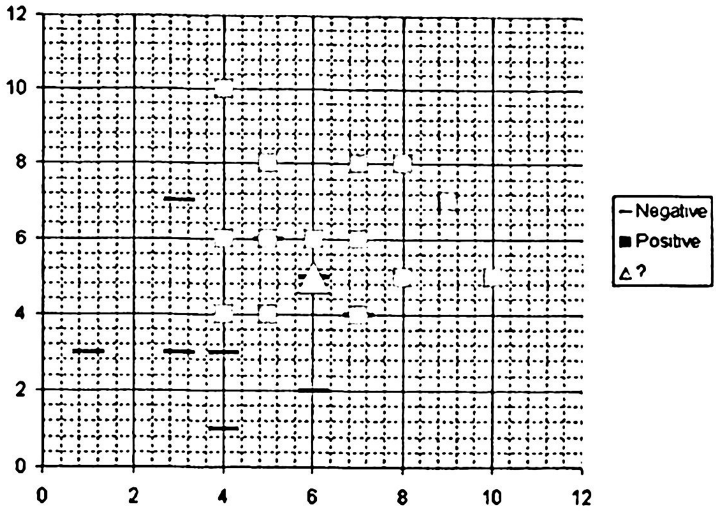

Fig. 1. PNN are conceptually similar to KNN

From Fig. 1 it is assumed that each case in the training set has two predictor variables, x and y. The cases are plotted using their x,y coordinates as shown in the figure. Also it is assumed that the target variable has two categories, positive which is denoted by a square and negative which is denoted by a dash. It can be note that the triangle is positioned almost exactly on top of a dash representing a negative value. But that dash is in a fairly unusual position compared to the other dashes which are clustered below the squares and left of center. So it could be that the underlying negative value is an odd case. The nearest neighbor classification will depend on how many neighboring points are considered. If 1-NN is used and only the closest point is considered, then the new point should be classified as negative since it is on top of a known negative point. On the other hand, if 9-NN classification is used, the closest 9 points are considered and then the effect of the surrounding 8 positive points may overbalance the close negative point. A probabilistic neural network builds on this foundation and generalizes it to consider all of the other points. The distance is computed from the point being evaluated to each of the other points, and a radial basis function (RBF) (also called a kernel function) is applied to the distance to compute the weight (influence) for each point. The radial basis function is so named because the radius distance is the argument to the function. *Weight=RBF(distance)* the further some other point is from the new point, the less influence it has. Different types of radial basis functions could be used, but the most common is the Gaussian function. The PNN architecture is shown in figure 2. The model has two layers: radial basis layer and competitive layer.

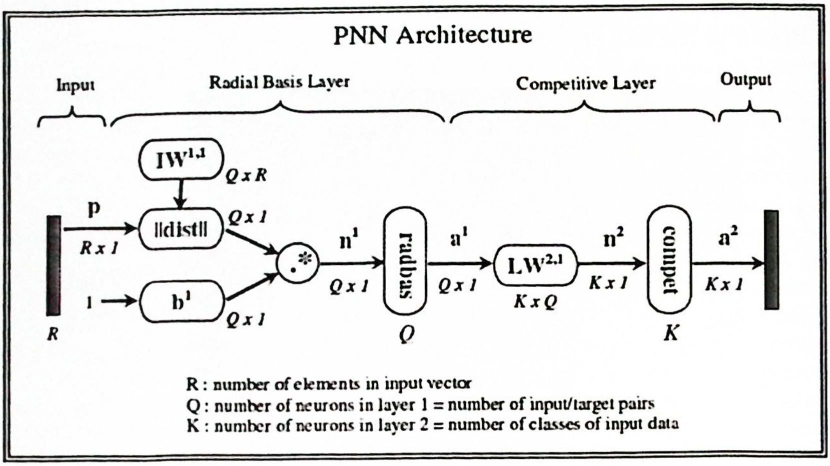

Fig. 2. PNN architecture.

There are Q input vector/target vector pairs. Each target vector has K elements. One of these element is 1 and the rest is 0. Thus, each input vector is associated with one of K classes. When an input is presented the $||dist||$ box produces a vector whose elements indicate how close the input is to the vectors of the training set. An input vector close to a training vector is represented by a number close to 1 in the output vector a^1. If an input is close to several training vectors of a single class, it is represented by several elements of a^1 that are close to 1. Each vector has a 1 only in the row associated with that particular class of input, and 0's elsewhere. The multiplication Ta^1 sums the elements of a^1 due to each of the K input classes. Finally, the second layer, produces a 1 corresponding to the largest element of n^2, and 0's elsewhere. Thus, the network has classified the input vector into a specific one of K classes because that class had the maximum probability of being correct.

2.2 Correlation Matrix and Eigenvalues Definitions

Correlation matrix definition. A Correlation matrix describes correlation among M variables. It is a square symmetrical M x M matrix with the (ik)th element equal to the correlation coefficient r_{ik} between the (i)th and the (k)th variable. The correlation coefficient is obtained as

$$r_{ik} = \frac{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)}{\sqrt{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)^2}\sqrt{\sum_{j=1}^{n}(x_{jk} - \bar{x}_k)^2}} \quad (1)$$

The diagonal elements (correlations of variables with themselves) are always equal to 1.00 [8].

Eigenvalue definition. Let A be a k x k square matrix and I be the k x k identity matrix. Then the scalars

$$\lambda_1, \lambda_2, \ldots, \lambda_k \quad (2)$$

satisfying the polynomial equation

$$|A - \lambda I \quad (3)$$

are called the eigenvalues or characteristic roots of a matrix A. The equation $|A$ - ▯ $I| = 0$ is called the characteristic equation, thus similar matrices and A and its transpose matrix have same eigenvalues [8].

3 Framework Description

The proposed detection framework is shown in figure 3. As the framework is a Process History Based fault detection method, this only requires a big quantity of historical data of the power system's nodes containing normal operation data and faulty data samples from the different types of faults that could be present in the system. These data sets are used as prior knowledge of the power system to perform the detection process.

The first step is to obtain several data sets from the power system's nodes. These data sets are matrices formed by windows of m samples and n power system's nodes where the voltage of each line of a certain node is monitored, that means three readings per node. Such matrices are constructed with normal node operation and different node faults present in system. For each node data sets its correlation matrix is obtained to see how their three lines are related. Once having the correlation matrix, their corresponding eigenvalues are computed as shown in section 2.2, such that in this way a signature to each of the different possible component states or fault types of the power system's nodes (K in figure 2) are given. The eigenvalues of the correlation matrix for each fault signature, then serve as the training vectors of the PNN corresponding to Q as described in section 2.1.

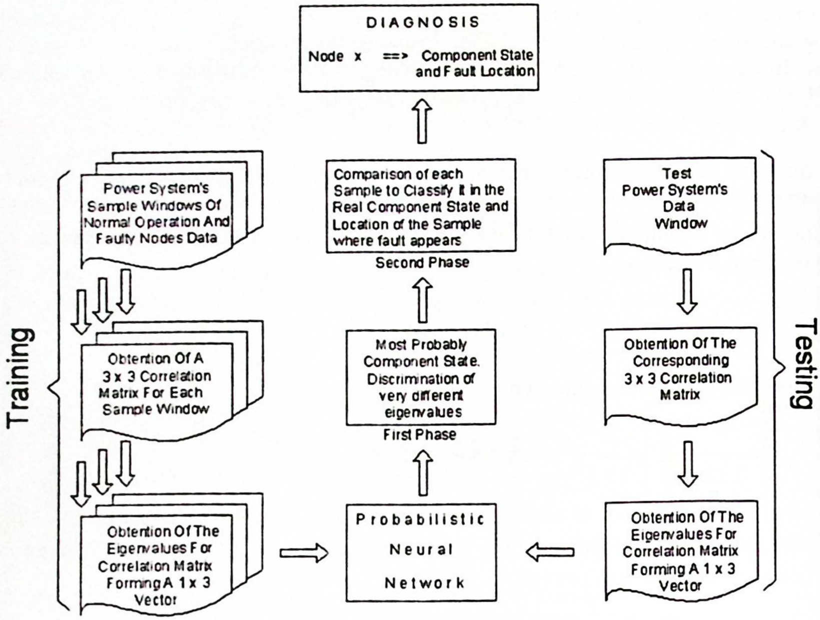

Fig. 3. General fault detection framework.

Each node will have then three eigenvalues (R components in figure 2) as they are comming from its correlation matrix that is a 3×3 matrix. Then the detection process is carried out in two phases. First phase that serves as a first filter or information discriminator. When there is a system to monitor, a window of m samples and n power system's nodes is taken, then each node is analyzed separately. m samples of the three lines corresponding to a node being monitored to find out which is its most probably state are taken. From the data set corresponding to a node being monitored its correlation matrix and its corresponding eigenvalues are obtained and then these last are used as an input vector to the PNN previously trained. It is mentioned "the most probably state" because unfortunately not all the eigenvalues of all of the node states are so different such that PNN could not classify them easily. But it has been found that for certain signature faults eigenvalues are very similar, thus there is here a discrimination/classification phase, because it is necessary to look for the real state but only comparing between just a couple of similar signatures instead of the whole bunch of node states. The output from the PNN automatically discriminates node states that are very different and gives the most probably real node state. Once the one possible node state is obtained, a second phase of the framework begins to work. In the second phase each sample of each node is taken and its magnitude is obtained, then a comparison against a constant magnitude of the probably signature faults are carried out. This comparison serves as a classifier that gives the real node component state and can be used too to locate the period of time or sample number where the fault occurs.

4 Case Study

This section shows the performance of the framework proposed in a multiple fault simulations in the IEEE network shown in figure 4.

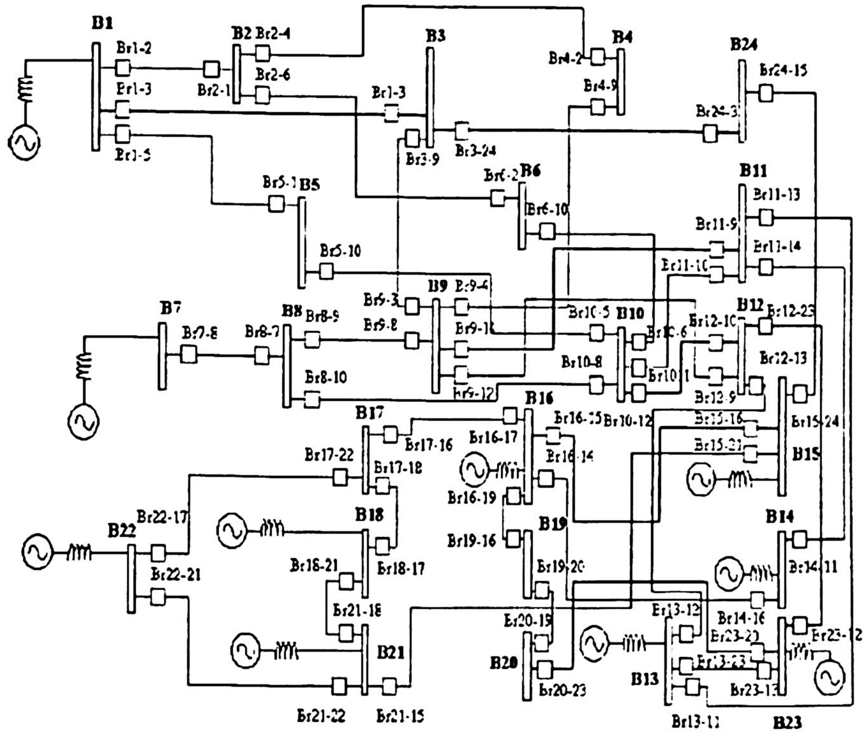

Fig. 4. IEEE reliability test system single line diagram.

Figure 4 shows an electrical power system having dynamic load changes. 24 fault simulations were carried out to determine the performance of the approach including in them symmetrical and unsymetrical faults at random nodes (3,9,10 and 13) taking into account different multiple faults scenarios combining faults such as: one line to ground (A GND), two lines to ground (A-B GND), three lines to ground (A-B-C GND), or faults between two lines (A-B or B-C) and the no fault mode (NO FAULT).

The methodology proposed is applied as follows:

1. Obtain windows of 100 samples from normal and faulty operation history process data (electrical voltage in each node's line).
2. Obtain correlation matrix for each node, which gives a *3* x *3* matrix.
3. Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), with this 3 eigenvalues form an input vector to train the PNN.
4. Take a test data set of 100 samples from the electrical power system being monitored.
5. Obtain correlation matrix for each node, which gives a *3* x *3* matrix.
6. Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), with this 3 eigenvalues form an input vector to the PNN.
7. First Phase: Take the output of the PNN as one of the two probably states of the node monitored.
8. Second Phase: Take each sample of each node monitored and obtain its magnitude, then compare it against the constant magnitude of the two probably signature faults and classify it using this simple criteria. Locate the period of time or sample number where the fault occurs.
9. Give the diagnosis of each node being monitored. If a fault is present in a specific node give the type and location of it, else print NO FAULT.

In the following tables the performance of the approach is shown taking into account three possible cases. Case 1, system is working properly during the first 25 samples from a total of 100, that means 25 samples are ok and 75 samples corresponds to fault present in system. Case 2 takes 50 samples of normal operation data and 50 samples with fault present. And case 3 takes 75 samples of normal operation and 25 with fault present. Table 1 and 2 shows a detailed example of how percentages were obtained. Tables 3 and 4 gives a summary of the obtained percentages for each of the three cases considered.

From tables 3 and 4 it can be said that the different performances are due because eigenvalues of correlation matrices are very similar when there are a major quantity of normal operation data in sample window. The more normal operation data in sample window the more difficult to classify eigenvalues by PNN because they are very similar.

Table 1. Detail performance of detection per node's component state with 25 samples ok and 75 samples with fault present (case 1).

Component State	Correct	Wrong	Accuracy
A-B-C GND	14	0	100%
A-B GND	10	0	100%
A GND	14	0	100%
A-B	18	0	100%
B-C	16	0	100%
NO FAULT	13	11	54.16%

Table 2. Detail performance of detection per node's number with 25 samples ok and 75 samples with fault present (case 1).

Node Number	Correct	Wrong	Accuracy
3	20	4	83.33%
9	19	5	79.16%
10	22	2	91.66%
13	24	0	100%

Table 3. Accuracy of detection per node's component state for the different cases.

Component State	Case 1	Case 2	Case 3
A-B-C GND	100%	100%	100%
A-B GND	100%	100%	100%
A GND	100%	85.71%	92.85%
A-B	100%	83.33%	50%
B-C	100%	68.75%	68.75%
NO FAULT	54.16%	58.33%	79.16%

Table 4. Accuracy of detection per node number for the different cases.

Node Number	Case 1	Case 2	Case 3
3	83.33%	83.33%	83.33%
9	79.16%	75%	70.83%
10	91.66%	87.5%	62.5%
13	100%	95.83%	100%

An important thing to remark is that there has been carried out several tests when all data were from normal operation and it has been found out that the framework have detected 100% of them as NO FAULT node's component state. Percentages shown in tables are low because criteria used in the second phase of the framework has to be with the maximum magnitude value and a threshold that needs to be set as the upper limit to make the difference between two very similar signatures for the same node.

Figure 5 shows an example of the results obtained for a simulation of case 2, that has the following faults types:

1. 3 A GND, that is a fault present in node 3 of type line A to ground.
2. 9 A-B GND, that is a fault present in node 9 of type line A and B to ground.
3. {10,13} NO FAULT, that is nodes 10 and 13 working properly.

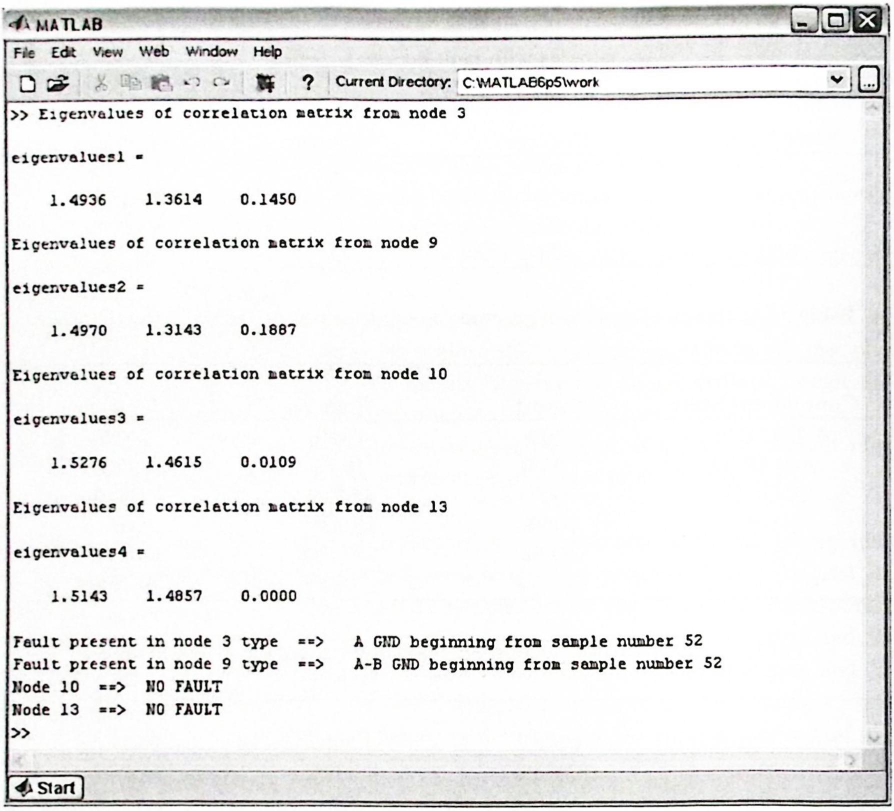

Figure 5. Example of the results given by matlab simulation

4.1 Comparison against a diagnostic system based on probabilistic logic.

To observe the general performance of the proposal, a comparison against a diagnostic system based on probabilistic logic taken from [12] has been carried out. Table 5 and 6 show the performance of the diagnostic system based on probabilistic logic.

Table 5. Performance of detection per node's component of the diagnostic system based on probabilistic logic.

Component State	Correct	Wrong	Accuracy
A-B-C GND	14	0	100%
A-B GND	10	0	100%
A GND	12	2	85.7%
A-B	15	3	83.3%
B-C	16	0	100%
NO FAULT	17	7	70.8%

Comparing the results of both frameworks it can be note that in general they have a very similar performance, but when comparing case 1 of the proposal against the diagnostic system based on probabilistic logic it could be said that framework propose has a better performance. Another important point is that the proposal is relatively easier to implement and to update when power system grows up.

Table 6. Performance of detection per node number of the diagnostic system based on probabilistic logic

Node Number	Correct	Wrong	Accuracy
3	19	5	79.1%
9	21	3	87.5%
10	21	3	87.5%
13	23	1	95.8%

5 Conclusions

This paper has presented a fault detection framework for electrical power systems with dynamic load changes using a PNN based on history process data. An advantage over model based methods is that this framework needs historical data of normal system operation as well as faulty data sets to train the PNN, which in practice it is relatively easy to obtain for computer controlled systems. It has been decided to use the PNN because is ideal when working on classification problems. Its most important advantage is that it needs only a little time for its training. It has been proposed a two phases relatively easy to implement fault diagnosis method.

In the first phase the eigenvalues of the correlation matrix are taken and used them as inputs for a PNN to classify the node's component state. It has been shown how this classification could be improved and carry out when eigenvalues are very similar with the implementation of a second phase where a simple comparison of each sample magnitude to the constant value of a certain signature fault has been apply and at the same time it gives the location of a fault if a it is present.

It can be concluded too, that as there is as many faulty data in the sample window the proposal has a better performance because eigenvalues are easily classified by the PNN as they have very different values. Another advantage of this proposal is that as it diagnostics each node, it could be detected simultaneous and nonsimultaneous simple, multiples, a combination of different faults as well as their corresponding location on each node separately.

References

1. V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. Kavuri (2003): A review of process fault detection and diagnosis Part I, Part II and Part III. Computers and Chemical Engineering 27(2003) 293-311.
2. Zhang D., Dai S., Zheng Y., Zhang R., Mu P. (2000): Researches and Applications of a Hybrid Fault Diagnosis System Proceedings of the 3r world Congress on Intelligent Control and Automation, Hefei, P.R. China, pp. 215- 219.
3. Z. Yongli, H. Limin, L. Jinling (2006): Bayesian Networks-Based Approach for Power Systems fault Diagnosis IEEE Transactions on power Delivery, Vol. 21, No. 2, pp. 634-639.
4. L. Xu, M. Chow (2005): Power Distribution Systems Fault Case Identification Using Logistic Regression and Artificial Neural Network Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, pp. 163-168.
5. H. Ren, Z. Mi (2006): Power System Fault Diagnosis Modeling Techniques based on Encoded Petri Nets IEEE Power Engineering Society General Meeting.
6. M. Peng, W. Hanli, C. Bin (2006): Study of Fault Diagnosis for Power Network based on MAS International Conference on Power System Technology.
7. R. Duda, P. Hart, and D. Stork (2001): Pattern Classification. Second Edition 2001 ISBN 0-471-05669-3.
8. R. Johnson, D. Wichern: Applied Multivariate Statistical Analysis. Prentice Hall, fifth edition.
9. N. Zanzouri, and M. Tagina (2002): A Comparative Study of Hybrid System Monitoring Based on Bond Graph and Petri Net Modeling. IEEE SMC.
10. D. Du, X. Lou, and C. Wu (2005): Dynamic Model of FCCU and its Application in a Hybrid Fault Diagnosis System. 2005 International Conference on Control and Automation (ICCA 2005), June 27-29, 2005, Budapest, Hungary.
11. W. Wang, X. Bai, W. Zhao, J. Ding and Z. Fang (2005): Hybrid Power System Model and the Method for Fault Diagnosis. 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China.
12. L. Garza (2001) : Hybrid Systems Fault Diagnosis with a Probabilistic Logic Reasoning Framework. PhD thesis.