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

Diagnóstico de Fallas Múltiples en Sistemas Eléctricos de Potencia con Cambios de Carga Dinámicos Utilizando Redes Neuronales Probabilísticas

Juan Pablo Nieto González[1], Luis Garza Castañón[2] and Rubén Morales Menéndez[3]
[1] I.T.E.S.M. Campus Saltillo, Departamento de Mecatrónica
Saltillo, Coahuila, México
juan.pablo.nieto@itesm.mx
I.T.E.S.M. Campus Monterrey,
[2]Departamento de Mecatrónica y Automatización, [3]Centro de Automatización Industrial
Monterrey, Nuevo León, México
[2]legarza@itesm.mx, [3]rmm@itesm.mx

Article received on December 18, 2007; accepted en February 27, 2009

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. It detects the faulty nodes, the type of fault in those nodes and the time when it is present. 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 normal operation, symmetrical and asymmetrical fault disturbances. The second phase is a sample magnitude comparison used to detect and locate the presence of a fault. A set of simulations are 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.
Keywords: Fault Diagnosis, Multiple Faults, Probabilistic Neural Networks, Correlation Matrix, Eigenvalues, Power System, Dynamic Load Changes

Resumen. El monitoreo de sistemas de potencia es particularmente retador debido a la presencia de cambios dinámicos de carga de los nodos de la red en modo de operación normal, así como la presencia de variables continuas y discretas, información con ruido y falta o exceso de datos. Este artículo propone un método de diagnóstico de fallas que es capaz de localizar el conjunto de nodos involucrado en eventos de fallas múltiples. El método detecta los nodos con falla, el tipo de falla y el tiempo en el cual está presente la falla. El método está compuesto de dos fases: En la primera fase una red neuronal probabilística es entrenada con los eigenvalores de los datos de voltaje obtenidos en operación normal así como con fallas simétricas y asimétricas. La segunda fase emplea una comparación entre las muestras para detectar y localizar la presencia de una falla. Se lleva a cabo un conjunto de simulaciones en un sistema eléctrico de potencia para mostrar el desempeño del método propuesto y se realiza una comparación contra un sistema de diagnóstico basado en lógica probabilística.
Palabras clave: Diagnóstico de Fallas, Fallas Múltiples, Redes Neuronales Probabilísticas, Matriz de Correlación, Eigenvalores, Sistemas de Potencia, Cambios Dinámicos de Carga

1 Introduction

The increased interest in fault diagnosis in electrical power networks (EPN) is because their complexity and high degree of interconnection can lead to an overwhelming number of alarms as a result of a disturbance. In order to manage the diagnosis task, supporting tools that help operators in this are needed. Two main features are demanded: efficiency and efficacy.

When a fault occurs in an EPN, the consequences are often not limited to the point where the fault occurred but to other points where fault can be propagated. A short-circuit fault gives rise to local damages due to a high amount of energy deployed within a limited geometrical location. The global effects of the same short circuit give rise to a voltage dip, which can lead to malfunction of other processes.

Since the impact of damage is proportional to the time that fault is presented the diagnosis time is essential. The advent of current limiting devices, which limit the fault current before it reaches dangerous levels, is an application where fast and reliable fault detection is of importance

There are many research works in the fault detection field. Many approaches are analytic methods, other are based on Artificial Intelligence (AI) or statistical methods. *[Venkatasubramanian, et al., 2003]* classifies the methods in three groups. See figure 1.

- Quantitative Model Based
- Qualitative Model Based and
- Process History Based

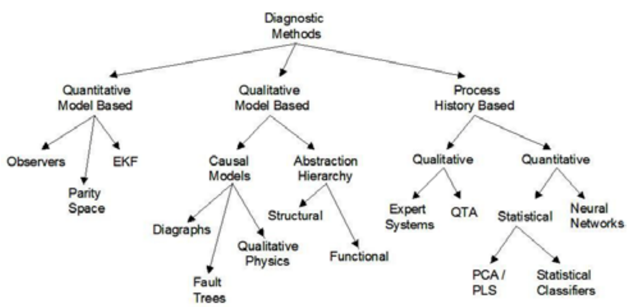

Fig. 1. Classification of Diagnostic Methods

Quantitative model based fault detection methods need a mathematical model of the system. The occurrence of a fault is captured by discrepancies between the monitored behavior and the one that is predicted by the model. These approaches exploit state estimation, parameter identification techniques, or 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 systems

Qualitative model based methods use symbolic reasoning which generally combines different kind of knowledge with graph theory. An advantage of these methods is that an explicit model of the system is not necessary. Knowledge-based approaches such as expert systems may be considered as alternative (or complementary) when analytical models are not available.

Process history based methods only require historical process data. There are several ways in which these data can be transformed as prior knowledge of a system. These transformations are known as feature extraction. They could be qualitative, as those used by expert systems, quantitative, as those used in neural networks, Principal Component Analysis (PCA), PLS or statistical pattern recognition

Every approach has some drawbacks; the need for better results motivates the design of hybrid methods that combines several approaches. In the Electrical Power Networks (EPN) domain several applications have been developed. *[Zhang, et al., 2000]* incorporates model based diagnosis and signal analysis with neural networks. *[Bouthiba, 2005]* proposed an approach based on four independent artificial neural networks (ANN) for real time fault detection and classification in power transmission lines. The technique uses consecutive magnitude current and voltage data at one terminal as inputs to the ANN. The ANN outputs are used to indicate simultaneously the presence and the type of the fault. *[Hartstein, et al., 2007]* developed a methodology using wavelet transform for phase to ground fault detection in primary distribution systems, but it is an efficient methodology only for single phase fault detection in unbalanced distribution systems. *[Yongli, et al., 2003]* presented 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 *[Xu and Chow, 2005]* a research is performed about the use of logistic regression and neural networks to classify fault causes. *[Ren and Mi, 2006]* proposed 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 *[Peng, et al., 2006]* 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. *[Nieto J.P., et al 2007]* presented a fault detection framework based on history process data that uses a combination of PCA, control charts and statistic operation limits to make comparisons between a suspicious sample against its normal values giving in this way the detection of a single or multiple faults existing in the system.

In other areas of application, *[He Q. Peter, et al., 2004]* proposed a process monitoring which is composed of three parts: preanalysis, visualization and diagnosis, where the proposed method integrates PCA, FDA and clustering analysis taking advantage of each technique for a complete solution. *[Liang W., et al., 2005]* combined the use of signed directed graph to make a classification model, PCA and fuzzy knowledge to form a qualitative and quantitative model and compares the grade of the patterns needed to be diagnosed to the given fault patterns. *[Gentil S., et al., 2004]* proposed a method based on the interaction between AI and control techniques. It uses a causal graph representation of the process, enabling decomposition into subsystems and reducing the diagnostic computational complexity. After that, at local level, FDI techniques based on numerical residual generation and analysis are carried out. *[Liang J., et al., 2003]* showed how PCA and statistical control charts are used to detect process operating faults on an industrial rolling mill reheating furnace. The Q statistic and Hotelling T^2 statistic are used to calculate the control limits of the statistical control chart. *[Shi W., et al., 2005]* proposed a fault diagnosis model based on machine learning which extracts multi-dimension features from the detected signal to supervise the different features of it simultaneously.

The goal of this research work is to build a full diagnostic system. The system must be able to detect single or multiple, simultaneous or non-simultaneous faults, as well as be capable to diminish the false alarms rate using only historical data. The main advantages of this framework are first, the relatively easy way to obtain historical data from systems and processes controlled by computers, and second, to have an alternative approach when the modeling of the system is very difficult or even impossible due to the lack of experience of the diagnostic system designer or the high degree of complexity of the system itself. In this work Probabilistic Neural Network (PNN) was selected as a fault detector, mainly due to the simplicity of its learning procedure. The PNN needs just a few data to be trained, tackling in this way the problems of time consuming when learning and the storage capability of the training samples, being these two great advantages of the PNN over other networks architectures. In addition, when comparing our approach with those presented above, it could be seen that ours is easier to implement as we only have one neural network to train and when the system changes we only have to update the information of the new nodes or delete the information of those nodes that were taken away from the system.

A multiple fault diagnosis framework composed of two phases is proposed. In a previous step, eigenvalues are computed from the correlation matrix which is built from historical data, and then they are used as the inputs of the probabilistic neural network. In the first phase, the most likely 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 the fault.

The organization of the paper is as follows. Section 2 explains PNN fundamental basis and gives the correlation matrix and eigenvalues definitions. Section 3 gives a general description of the framework. Section 4 shows a case study. Section 5 concludes the paper.

2 Preliminary

2.1 Probabilistic neural network basis

PNNs are conceptually similar to K-Nearest Neighbor (KNN) models *[Duda, et al., 2001]*. A predicted value of an item is likely to be about the same as other items that have close values of the predictor variables.

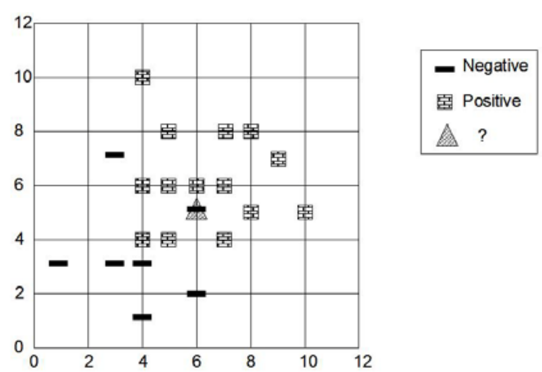

Fig. 2. PNN are conceptually similar to KNN

Figure 2 shows that each case in the training set has two predictor variables: *x* and *y*. The cases are plotted using their values as coordinates. 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 noted 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 RBF is a function whose output depends on the distance to a point called center. Gaussian RBF are symmetric functions with respect to x=0. See figure 3.

The PNN architecture is shown in figure 4. The model has two layers:

a) Radial Basis Layer and
b) Competitive Layer

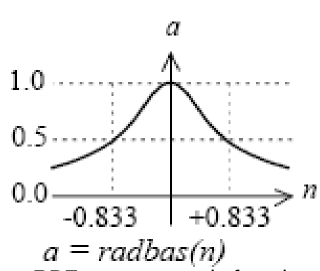

Fig. 3. Gaussian RBF are symmetric functions with respect to x=0.

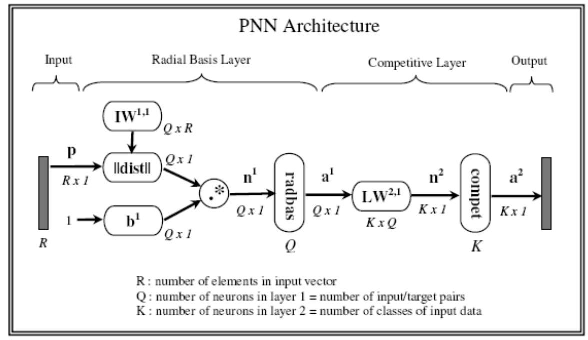

Fig. 4. PNN architecture

There are Q input vector/target vector pairs. Each target vector has K elements. One of these elements 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 Ta1 sums the elements of a1 due to each of the K input classes.

Finally, the second layer produces a 1 corresponding to the largest element of n2, 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

$$r_{ik} = \frac{\sum_{j=1}^{n}\left(x_{ji}-\overline{x}_1\right)\left(x_{jk}-\overline{x}_k\right)}{\sqrt{\sum_{j=1}^{n}\left(x_{ji}-\overline{x}_i\right)^2}\sqrt{\sum_{j=1}^{n}\left(x_{jk}-\overline{x}_k\right)^2}} \quad (1)$$

The diagonal elements (correlations of variables with themselves) are always equal to 1 [Johnson and Wichern, 2002]. In this work, for each node data set of the system being monitored, its correlation matrix is computed to see how their three lines or phases are related and to avoid false alarms due to a possible fault reflected on an non faulted node's line because of the correlation present between the three node´s lines.

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

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

satisfying the polynomial equation

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

are called the eigenvalues or characteristic roots of a matrix A. The equation $|A - \lambda I| = 0$ is called the characteristic equation, thus similar matrices to A and its transpose matrix have the same eigenvalues *[Johnson and Wichern, 2002].* The eigenvalues of the correlation matrix of normal operation data as well as for every type of fault are used for identification of (fault or normal operation) signature. Later, the eigenvalues are going to be used also as the training vectors of the PNN.

3 Framework description

The proposed detection framework is shown in figure 5. The framework is a Process History Based fault detection method. It only requires historical data of the Electrical Power Network (EPN). The amount of data will depend on the information recorded in the historic databases of the system. Thus, in order to know how big a data set is considered adequate, it is necessary to be sure that these databases contain information about normal and possible faulty operation. Then the total amount of data needed will depend on the nature of each individual problem to be solved. For instance, in the example shown in the next section, we used a 4% of the total amount available of faulty data, for each possible different fault to train the probabilistic neural network. The use of just a few quantity of examples to train, is one of the great advantages of using the probabilistic neural network over other networks architectures. In summary, these historic data sets are used as prior knowledge of the power system to perform the EPN detection process.

The first step of the training phase is to obtain historical normal and faulty data sets of the voltages of each line of the EPN's nodes (see Figure 5). These data sets are matrices formed by windows of *m* samples and *n* power system's nodes where the voltage of each line of each of the power system's node is monitored, that means three readings per each node as shown in table 1. Such matrices are built for normal and faulty operating conditions in the system.

For each node data set, its correlation matrix is obtained to see how their three lines are related. As an example we took the *m* samples of the three lines (A,B,C) of the node 1. This is shown in table 2.
Then the correlation matrix is calculated for the voltage samples of the three lines of the node 1, resulting thus a matrix (4), where the correlation coefficients are computed as in equation (1).

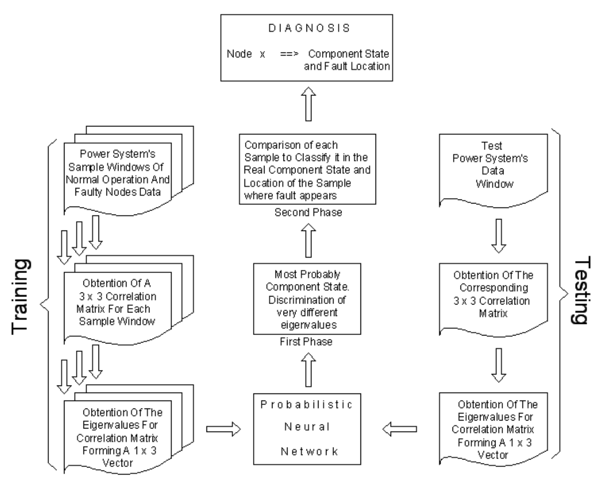

Fig. 5. General fault detection framework

Table 1. Matrix containing the three lines of each of the power system´s nodes being monitored

Power System Nodes' Lines

		Node 1			Node 2				Node n		
		Line A	Line B	Line C	Line A	Line B	Line C	•••	Line A	Line B	Line C
	1	v_{1A}	v_{1B}	v_{1C}	v_{2A}	v_{2B}	v_{2C}	•••	v_{nA}	v_{nB}	v_{nC}
Voltages	2	v_{1A}	v_{1B}	v_{1C}	v_{2A}	v_{2B}	v_{2C}	•••	v_{nA}	v_{nB}	v_{nC}
Samples	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	m	v_{1A}	v_{1B}	v_{1C}	v_{2A}	v_{2B}	v_{2C}	•••	v_{nA}	v_{nB}	v_{nC}

$$Corr_{N1} = \begin{bmatrix} 1 & r_{AB} & r_{AC} \\ r_{BA} & 1 & r_{BC} \\ r_{CA} & r_{CB} & 1 \end{bmatrix} \qquad (4)$$

Given the correlation matrix, their corresponding eigenvalues are computed having in this way a signature for each of the different faulty operating conditions (*K* in figure 4).

Table 2. Matrix of the voltages monitored from the three lines of the power system´s node 1

		Node 1		
		Line A	Line B	Line C
	1	v_{1A}	v_{1B}	v_{1C}
Voltages	2	v_{1A}	v_{1B}	v_{1C}
Simples	⋮	⋮	⋮	⋮
	m	v_{1A}	v_{1B}	v_{1C}

The eigenvalues of the matrix in expression (4) are obtained as follows:

$$\left| \begin{bmatrix} 1 & r_{AB} & r_{AC} \\ r_{BA} & 1 & r_{BC} \\ r_{CA} & r_{CB} & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right| \qquad (5)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues or roots of the characteristic equation (5).

The eigenvalues of the correlation matrix (6) are going to be used as the training vectors of the PNN corresponding to *Q* as described in section 2.1. This training vector then looks like the one shown in expression (6)

$$eigenvalues1 = [\lambda_1 \lambda_2 \lambda_3] \qquad (6)$$

Each node will have then three eigenvalues (*R* components in Figure 4) as they are coming from its correlation matrix that is a *3* x *3* matrix built as depicted in table 2 and expression (4). Up to here it has been described the testing process shown on the left of figure 5.

Then the detection process is carried out in two phases. The first phase is basically a first filter or information discriminator. When a window of *m* samples and *n* EPN's nodes is taken, each node is analyzed separately. From the data set corresponding to a particular node being monitored, its correlation matrix and their corresponding eigenvalues are obtained and used as the input vector to the PNN previously trained. It is mentioned "the most probably component state" (Figure 5) because unfortunately not all the eigenvalues of the node states are so different (Figure 7) such that the PNN could classify them easily. But we have found instead that, when certain signature faults eigenvalues are very similar, there exist here a discrimination/classification phase, because it is necessary to look for the real state but only comparing 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 likely real node state. Once the possible node state is obtained, a second phase of the framework starts. In the second phase each sample of every node is taken and its magnitude is obtained as shown in equation 7. Then a comparison of each of these results against the constant magnitudes, previously obtained (with equation 7 too), for each of the probably signature faults are carried out.

$$Cons\tan t = \sqrt{v_{1A}^{2} + v_{1B}^{2} + v_{1C}^{2}} \quad (7)$$

The constant magnitude of each type of fault has to be calculated in advance, such that for instance, a fault of three lines to ground (A-B-C GND) on a node gives a constant magnitude of cero. If the assumption is that node 1 has the type of fault where two lines, for example A and B go to ground, the constant magnitude will be calculated as:

$$Cons\tan t = \sqrt{0^2 + 0^2 + v_{1C}^{2}} \quad (8)$$

This simple comparison is used as a second classifier that delivers the real node component state and can be used to locate the period of time or sample number where the fault occurred. This second classification is needed in order to distinguish and diminish false alarms when a fault is present. This step is done in order to assure that the classification made by the PNN is a good one, because due to the similarity of the eigenvalues obtained for different types of node's states it is possible to have ambiguous diagnosis. As shown in next section in table 7, we have found similar eigenvalues for different fault signatures for the case study.

4 Case Study

This section shows the performance of the proposed framework in multiple fault scenarios simulated in the electrical network shown in figure 6.

In this power system, dynamic load changes were simulated, and also 24 different fault scenarios to determine the performance of the approach. We include in the study symmetrical and asymmetrical faults at four random nodes (3,9,10 and 13). The simulations included multiple faults scenarios with different node's states 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).

Each database for the 24 simulations contained 5000 samples, and every possible fault included 300 samples. The amount of eigenvalues used in the learning process of the probabilistic neural network was 12 examples per each node state. As we had 6 states including the no fault mode, we trained the neural network with a total amount of 72 eigenvalues, having in this way 12 of them per each state. At the same time we divided these 12 eigenvalues examples of each possible fault in 3 groups of 4 eigenvalues containing 75%, 50% and 25% of faulty data coming from windows of 100 samples. This means that the quantity of eigenvalues we needed to store for the learning process of the probabilistic neural networks were only 72 vectors, each of size 1 x 3.

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 the correlation matrix for each node, which gives a 3 x 3 matrix.
3.- Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), and with this 3 eigenvalues build an input vector to train a PNN.
4.- Take a test data set of 100 samples from the electrical power system being monitored.
5.- Obtain the correlation matrix for each node, which gives a 3 x 3 matrix.
6.- Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), and with this 3 eigenvalues build an input vector for 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 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 also 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, when the 100 samples windows are selected as follows:

a) 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 a fault present in the system.
b) Case 2 takes 50 samples of normal operation data and 50 samples with a fault present.
c) Case 3 takes 75 samples of normal operation and 25 with a fault present.

Tables 3 and 4 show the performance obtained just for case 1. Tables 5 and 6 give a summary of the accuracy percentages for each of the three cases considered.

Table 3. Performance of detection per node's component state with 25 samples ok and 75 samples with a fault present (case 1)

Component State	Correct	False Alarms	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 4. Performance of detection per node's number with 25 samples ok and 75 samples with a fault present (case 1)

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

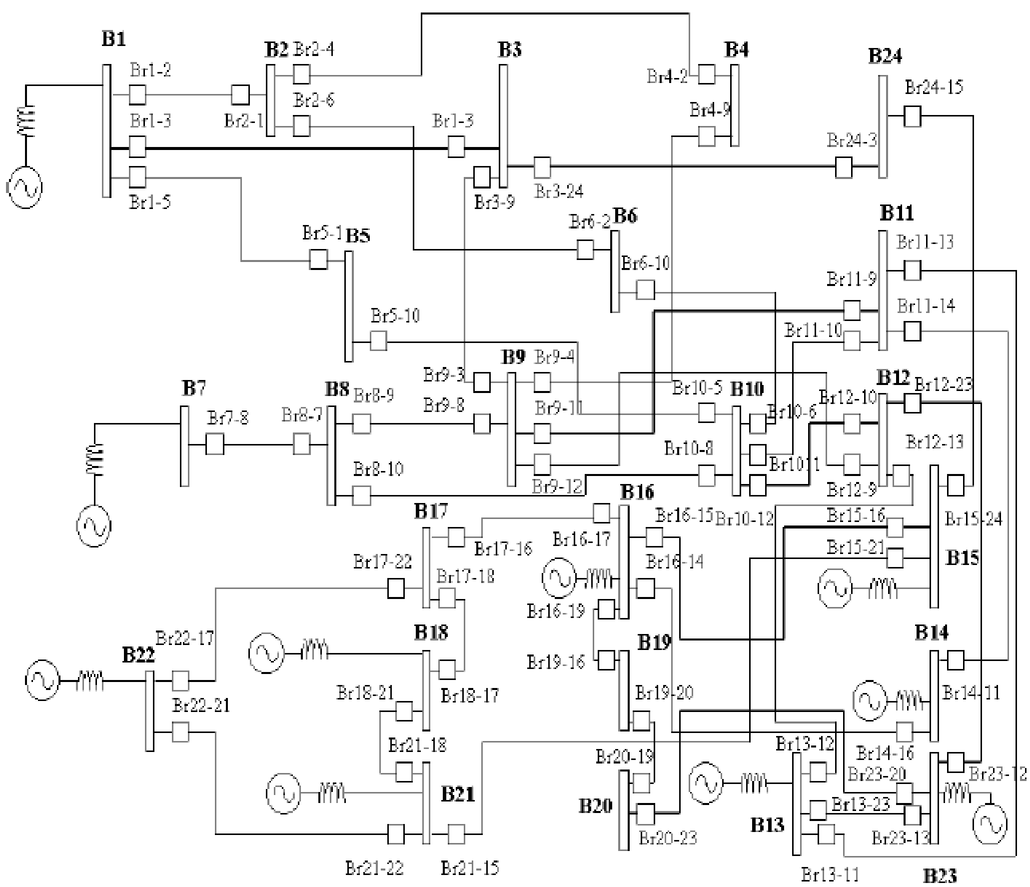

Fig. 6. IEEE reliability test system single line diagram

Table 5. Performance 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 6. Performance 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%

The difference in performance, shown in tables 5 and 6 for the same fault scenarios, are explained by considering the high similarity of eigenvalues of correlation matrices, when there are more normal operation data than fault samples, in each data window.

In this framework, when more normal operation data appear in the sample window, more difficult is to classify the eigenvalues by the PNN, because they look very similar. This can be seen on figure 7, where a simulation for case 2 shows the eigenvalues obtained for four different power system´s nodes and whose operation modes are different between themselves. In this example, the following faults were simulated:

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 are working properly.

In summary, the similarity of the eigenvalues obtained for different types of node's states, gives rise to ambiguous diagnosis, as is shown in table 7.

Table 7. Similar eigenvalues found for the different operation modes for the power system being analyzed

Eigenvalues for Fault type	Are similar to	Eigenvalues for Fault type
A-B-C GND	≈	NO FAULT
A-B GND	≈	A GND
A-B	≈	B-C

Several tests were carried out, when all data came from normal operation mode, and it has been found that the framework has detected 100% of them as NO FAULT node's component state. Percentages shown in the tables are low because criteria used in the second phase of the framework, are related to 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.

4.1 Comparison of the general performance against several classical methods.

In order to observe the relative general performance of our proposal, a comparison against four classical and similar Process History Based fault detection methods has been carried out. We have chosen diagnostic methods that are based on the use of PCA and/or Multidimensional feature extraction of signal based on machine learning, due to the large number of references in the literature of fault diagnosis that make use of them, when dealing with storing and handling big quantities of data.

```
>> Eigenvalues of correlation matrix from node 3

eigenvalues1 =

    1.4936    1.3614    0.1450

Eigenvalues of correlation matrix from node 9

eigenvalues2 =

    1.4970    1.3143    0.1887

Eigenvalues of correlation matrix from node 10

eigenvalues3 =

    1.5276    1.4615    0.0109

Eigenvalues of correlation matrix from node 13

eigenvalues4 =

    1.5143    1.4857    0.0000

Fault present in node 3 type  ==>   A GND beginning from sample number 52
Fault present in node 9 type  ==>   A-B GND beginning from sample number 52
Node 10  ==>  NO FAULT
Node 13  ==>  NO FAULT
>>
```

Fig. 7. Example of the results given by a matlab simulation

Table 8 shows this comparison. The first column shows the capability to detect single faults, simultaneous and non-simultaneous multiple faults, as well as detection of measurement and process noise presence. The PCA method was applied as depicted in *[Liang N., et al 2003]*, the Machine Learning technique was the one proposed by *[Shi W., et al 2005]* the fourth column method is the one developed by *[Nieto J.P., et al 2007]*, the Probabilistic Logic was the method proposed by *[Garza, 2001]* and the last column shows the capabilities of the present work.

We can notice that the method of PCA used without any other technique, offers a poor data analysis, thus as pointed out before, when a combination of two or more techniques is done, a better performance should be expected. It is observed also that the rest of methodologies offer multiple fault detection.

Nevertheless, the use of Machine Learning techniques needs to be implemented for each measured signal, which generates a big quantity of data to be analyzed. Meanwhile PCA + Control Charts + Statistic Limits and Probabilistic Logic methodologies avoid this data explosion, but they could not detect non-simultaneous multiple faults. Finally an advantage that is noted immediately is that our proposed methodology could detect all kinds of faults and also noise presence in the system.

Table 8. Comparison of the general performance of our proposal against several classical methods

Detection Of	PCA Method	Machine Learning	PCA+Control Charts + Statistic Limits	Probabilistic Logic	PNN + Magnitud Comparison
Single Fault	√	√	√	√	√
Simultaneous Multiple Faults	NO	√	√	√	√
Non-Simultaneous Multiple Faults	NO	√	NO	NO	√
Measurement Noise	NO	NO	√	√	√
Process Noise	NO	NO	NO	NO	√

4.2 Comparison against the diagnostic system based on probabilistic logic

In order to test the performance of our framework, a challenging complex system was chosen in the domain of electrical power networks (see figure 6). The system is a highly interconnected process with many components and dynamic signal variations during normal operation. We decided to compare the performance of the proposed framework against the diagnostic system taken from *[Garza, 2001]*, due to the availability of data and the similar general performance of the techniques shown in table 8.

This diagnostic system consists of a modeling step followed by a diagnosis step. In the modeling step it is used the Dynamic Independent Choice Logic (DICL) to represent the diagnosis problem with causal probabilistic models that represent both, the relationships between the elements of the system, and the dynamics of the process. The diagnosis task comprises two phases. In phase one, the diagnostic system generates all possible explanations from a set of discrete observations, consistent with the process model facts. The discrete observations are taken from the statuses of protection breakers installed between node's lines. The explanations contain the suspected faulted components. Phase one uses heuristics based on probabilities, to deal with the combinatorial explosion in the number of generated explanations, due to the large quantity of available information. In phase 2, the diagnostic framework models the dynamics of the problem by specifying Dynamic Probabilistic Models within DICL. The model structure is learned from data and the inference is performed with a maximum entropy classifier. These models represent the steady state behavior of a device or component. Faults are detected by analyzing a set of filtered residuals, computed from the difference between the dynamic model and the observed measurements. Phase 2 can be considered as a refining stage where non faulted components, given by the first phase, are discarded by analyzing continuous signals that give more insight in the behavior of the component.

Table 9 and 10 show the performance of this diagnostic system based on probabilistic logic.

Comparing the results of both frameworks, we noticed that they have a very similar performance, but when comparing case 1 of our framework against the diagnostic system based on probabilistic logic, our method has a better performance.

Another important point is that our framework is relatively easier to implement and to update when power system scales up. In the probabilistic framework, new simulations are required to compute the fault detection thresholds and also a dynamic model needs to be learned for every added node.

Table 9. Performance of detection per node's component of the diagnostic system based on probabilistic logic

Component State	Correct	False Alarms	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%

Table 10. Performance of detection per node number of the diagnostic system based on probabilistic logic

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

Table 11 shows a comparison of the performance obtained per node's component with the framework presented in this paper against that of the diagnostic system based on probabilistic logic. We can see that both frameworks in general have a very similar performance, but when comparing case 1 of our proposal against the probabilistic logic diagnostic system, it is clearly noticed that the former reach a better behavior.

Table 11. Comparison of the accuracy of detection per node's component state of the 3 cases considered in the framework proposed against the probabilistic logic system

Component State	Case 1	Case 2	Case 3	Probabilistic Logic
A-B-C GND	100%	100%	100%	100%
A-B GND	100%	100%	100%	100%
A GND	100%	85.71%	92.85%	85.7%
A-B	100%	83.33%	50%	83.3%
B-C	100%	68.75%	68.75%	100%
NO FAULT	54.16%	58.33%	79.16%	70.8%

Table 12 is a comparison of the performance obtained per node number between the two frameworks.

Table 12. Comparison of the accuracy of detection per node number for the two frameworks

Node Number	Case 1	Case 2	Case 3	Probabilistic Logic
3	83.33%	83.33%	83.33%	79.1%
9	79.16%	75%	70.83%	87.5%
10	91.66%	87.5%	62.5%	87.5%
13	100%	95.83%	100%	95.8%

5 Conclusion

This paper has presented a fault detection framework for electrical power systems with dynamic load changes. This approach uses a PNN as a first classifier to obtain the most probably operation mode of the nodes being analyzed. An advantage we have over model based methods, is that this framework needs only historical data of normal system operation as well as faulty data sets, to train the PNN, which in practice is relatively easy to obtain for computer controlled systems. We use a PNN because it is an ideal choice to work on classification problems. Its most important advantage is that it needs only a little time for its training.

Thus, the only thing that is needed for the framework presented in this paper is a big quantity of normal and different faults data sets.

When a test sample arrives, it is necessary to obtain windows of data of *m* size. Then, in the first phase, the eigenvalues of the correlation matrix obtained from the samples windows 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 carried out when eigenvalues are very similar, with the implementation of a second phase. In this phase, a simple comparison of each sample magnitude to the constant value of a certain signature fault is made, to detect the type of fault, and at the same time the location of the fault.

It can be concluded that, when more fault than normal data appear in the sample window, our proposed framework has a better performance because eigenvalues are easily classified by the PNN as they have very different values.

The most important advantage of this proposal is that as it diagnoses the status of each node, it could detect simple and multiples faults, simultaneous and non-simultaneous faults and a combination of different faults as well as their corresponding location for each node separately.

References

1. **Barigozzi A., Magni L., & Scattolini R. (2004)** A Probabilistic Approach to Fault Diagnosis of Industrial Systems. *IEEE Transactions on Control System Technology,* 12(6), 201-212.
2. **Bouthiba T. (2005).** Fault detection and classification technique in EHV transmission lines based on artificial neural networks. *European transactions on electrical power,* 15(5), 443-454.
3. **Du D., Lou X. & Wu C. (2005).** Dynamic Model of FCCU and its Application in a Hybrid Fault Diagnosis System. *International Conference on Control and Automation*, Budapest, Hungary, 2, 1014-1017.
4. **Duda R., Hart P., & Stork D. (2001).** *Pattern Classification.* New York: Willey Interscience.
5. **Garza L. (2001).** *Hybrid Systems Fault Diagnosis with a Probabilistic Logic Reasoning Framework.* PhD thesis, Intituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey.
6. **Gentil S., Montmain J. & Combastel C. (2004).** Combining FDI and AI Approaches Within Causal Model Based Diagnosis. *IEEE Transactions on Systems,Man and Cybernetics.* Part B: Cybernetics, 34(5), 2207-2221.
7. **Hartstein R., Rezende K., & Suman A. (2007).** Fault Detection in Primary Distribution Systems using Wavelets. *International Conference on Power Systems Transients (IPST'07),* Lyon, France, 572-580.
8. **He Q. Peter, Wang J., & Joe Qin S. (2004**). A New Fault Diagnosis Method Using Fault Directions in Fisher Discriminant Analysis. *TWMCC Texas-Winsconsin Modeling and Control Consortium. Technical report number TWMCC-2004-05.* Austin, USA: *The University of Texas at Austin. Department of Chemical Engineering.*
9. **Johnson R. & Wichern D. (2002).** *Applied Multivariate Statistical Analysis.* New Jersery: Prentice Hall.
10. **Liang J., & Wang N. (2003).** Fault Detection and Isolation Based on PCA: An Industrial Reheating Furnance Case Study. *International Conference on Systems, Man and Cybernetics,* Hangzhou, China, 2, 1193-1198.
11. **Liang W., Wang B., Ma L., Zhang J. & Gao J. (2005).** Fault Diagnosis Approach Based on the Integration of Qualitative Model and Quantitative Knowledge of Signed Directed Graph. *Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, China 4, 2251 – 2256.
12. **Nieto J.P., Garza L.E. & Morales R. (2007).** Fault Detection Combining PCA, Control Charts and Statistic Operation Limits. *Research on Computing Science,* 27, 267-278.
13. **Nieto J.P., Garza L.E. & Morales R. (2007).** Multiple Fault Diagnosis in Electrical Power Systems with Probabilistic Neural Networks. *6th Mexican International Conference on Artificial Intelligence.* Aguascalientes, Mexico, 71-81.
14. **Nieto J.P., Garza L.E., Garza M. & Morales R. (2008).** Fault Diagnosis of Industrial Systems with Bayesian Networks and Neural Networks. *7th Mexican International Conference on Artificial Intelligence.* Mexico, 998-1008.
15. **Nieto J.P., Garza L.E., Rabhi A., & El Hajjaji A. (2008).** Fault Diagnosis of a Vehicle with soft Computing

Methods. *7th Mexican International Conference on Artificial Intelligence*. Mexico, 492-502.

16. **Nieto J.P., Garza L.E., Rabhi A., & El Hajjaji A. (2008).** Fault Detection and Diagnosis of a Vehicle Combining Multidimensional Scaling, Percentiles Range and PCA. *9th International conference on Sciences and Techniques of Automatic control & computer engineering.* Tunis, Tunisia, 1-11.
17. **Nieto J.P., Garza L.E., Rabhi A., El Hajjaji A. & Morales R.(2009).** Vehicle Fault Detection and Diagnosis combining AANN and ANFIS. *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*. Barcelona, Spain. 1079-1084.
18. **Nieto J.P. & Garza L.E.(in press).** Fault Detection with Statistical and Soft Computing Methods. In Lea M. Simon (Ed), *Fault Detection: Theory, Methods and Systems.* New York: Nova Science Publishers, Inc.
19. **Ren H. & Mi Z. (2006).** Power System Fault Diagnosis Modeling Techniques based on Encoded Petri Nets. *IEEE Power Engineering Society General Meeting.* Montreal, Canada, 6, 1111-1118.
20. **Shi W., Yan H., & Ma K. (2005).** A New Method of Early Fault Diagnosis Based on Machine Learning. *Fourth International Conference on Machine Learning and Cybernetics.* Guangzhou, China, 6, 3271-3276.
21. **Venkatasubramanian V., Rengaswamy R., Yin K. & Kavuri S. (2003).** A review of process fault detection and diagnosis Part I, Part II and Part III. *Computers and Chemical Engineering.* 27, 293-311.
22. **Wang W., Bai X., Zhao W., Ding J. & Fang Z. (2005).** Hybrid Power System Model and the Method for Fault Diagnosis. *IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific.* Dalian, China, 1-5.
23. **Xu L. & Chow M. (2005).** Power Distribution Systems Fault Cause Identification Using Logistic Regression and Artificial Neural Network. *13th International Conference on Intelligent Systems Application to Power Systems.* North Carolina, USA, 163-168. **Yongli Z., Limin H. & Jinling L. (2006).** Bayesian Networks-Based Approach for Power Systems fault Diagnosis. *IEEE Transactions on power Delivery*, 21(2), 634-639.
24. **Zanzouri N. & Tagina M. (2002).** A Comparative Study of Hybrid System Monitoring Based on Bond Graph and Petri Net Modeling. *International Conference on Systems, Man and Cybernetics*. Tunis, Tunisia, vol(4), 1-6.
25. **Zhang D., Dai S., Zheng Y., Zhang R. & Mu P. (2000).** Researches and Applications of a Hybrid Fault Diagnosis System. *3r world Congress on Intelligent Control and Automation*. Hefei, China, 215- 219.

Juan Pablo Nieto González

was born on October 7th 1973, in Saltillo, Coahuila. He is graduated from the master degree in Electrical Power and Distribution Systems (EPDS) in 1998 from La Laguna Institute of Technology and Industrial Electrical Engineering in 1995 from Saltillo Institute of Technology (ITS). He worked for about 7 years in EPDS Design and in Vehicle industry areas. Now, he works as professor in the Electrical-Electronics Department at ITS and in Mechatronics Department at ITESM. He is a student of the PhD in Mechatronics at ITESM where he has acquire expertise in Fault Diagnosis field having a research visit in 2008 at Laboratoire Modélisation Information et Systemes in Amiens, France. He has published several articles for MICAI and Safeprocess conferences in Fault Detection and Diagnosis.

Luis Eduardo Garza Castañón

was born on December 15th, 1963, in Monclova, Coahuila. He holds a PhD in Artificial Intelligence from ITESM, Monterrey since 2001, and graduated from the master degree in Control Engineering in 1988, and from Electronic Systems Engineering in 1986. He worked, for about 10 years, in the Control and Automation area in Metallurgical and Metal-mechanic industry. Now, he is a full time professor and researcher in the Mechatronics and Automation department at ITESM Monterrey, and is a level 1 member of the Sistema Nacional de Investigadores (SNI). He has published more than 40 articles in Fault Diagnosis and FTC.

Rubén Morales Menéndez

received the PhD degree in Artificial Intelligence from Tecnológico de Monterrey, in 2003. From 2000 to 2003, he was a Visiting Scholar with the Laboratory of Computational Intelligence at University of British Columbia, BC, Canada. He is currently Dean of Quality Enhancement and Accreditation (Tecnológico de Monterrey) and a consultant specializing in the analysis and design of automatic control systems for continuos processes for more than 22 years. He is member of the National Researches System of Mexico (level 1), and member of IFAC TC 9.3.