Abstract: The present work analyses the use of the Probabilistic Neural Network (PNN) as an automatic diagnosis system for detecting defects in rolling bearings using vibration signals. The influence of the metric and that of the mono and multi sigma in the PNN performance is analyzed and discussed. Genetic algorithms are used in order to optimize the sigma set that maximizes the PNN detection and classification performance. Two forms of training sets were constructed, one using the power spectral density of the signals and another using a blend of scalar parameters. Different classification complexities were employed. The results allow obtaining excellent classification rate as well as analyzing the influence of the main PNN parameters in its performance.

Keywords: predictive maintenance, fault detection, rolling bearings, probabilistic neural network.

1. Introduction

In order to improve industrial competitiveness, cost reduction is becoming increasingly important in industrial plant maintenance. Many methodologies, that have been used to improve production reliability and to reduce operational costs, are based on condition monitoring. In this context, it is important to develop optimized systems of monitoring and diagnosis in order to increase the precision of diagnoses and to reduce human errors.

The number of monitored points in an industrial plant can be as high as a dozen of thousands in what concerns bearings (rolling, journals, etc.), gearboxes, motors, pumps [1,2]. The most common commercial diagnostic systems compare some parameters of the measured signals, vibration mostly, to specific standard reference values. In general, the alarm threshold is established based upon energy parameters as the RMS of the complete vibration signals or that of certain frequency bands. An alarm is activated when the measured parameters exceed these values and, in this case, a specialist should perform a more accurate analysis (spectral analysis, envelope, time-frequency and time scalar methods, etc.) to confirm the diagnosis. The amount of erroneous diagnoses is large and frequent, because either an actual alarm does not occur or a conclusive diagnosis is not obtained, or else because the diagnosis is false.

More sophisticated diagnostic systems use classification methods, such as statistical and geometrical classifiers [3,4], neural networks [5,6,7,8] and fuzzy logic [6,9,10,11,12], to recognize machinery condition. These methods allow developing more automatic and reliable diagnostic systems. These automatic diagnostic systems should be robust to the point of dealing with both, a diversified source of information coming from different equipment and a diversified set of defects.

The present work describes automatic diagnostic systems for rolling bearing fault identification based on the Probabilistic Neural Network (PNN). The PNN is based on the
Bayesian statistical decision criterion, and it is not so well known especially in the case of machinery element fault classification.

Some characteristics of this kind of neural network may have important influence in the network performance. This is the case of the metric used in the Parzen function in order to compare the input and target vectors. Another important aspect of the PNN is how to model the statistical contribution of the elements of the training set. Two models are investigated in this paper, the monosigma and the multisigma.

A genetic algorithm is used in order to optimize the set of sigma that maximize the PNN performance.

The PNN performance under different conditions is tested by means of a database composed of vibration signals collected in several rolling bearing operation conditions (shaft speeds and load conditions) under nine different fault features. Three diagnostic strategies with different classification complexity were studied.

2. **Probabilistic Neural Network**

The PNN [7,8,13,14] is not a truly neural network paradigm, but an adaptation of the statistical Bayesian decision rule to the neural network formalism. Four layers compose the PNN architecture. The first one, the input layer, has its dimension dictated by the dimension (p) of the input vector. The second, the pattern layer, has the dimension of the number of examples in the training set, while the third one, a summation layer, has the dimension (k) equal to the number of classes in the set of the examples, and one output or decision layer.

An example of a PNN architecture is shown in Fig. 1. The number of layers and the number of nodes for each layer are rigidly fixed and are defined by the training set dimension.

When an input vector \( x = \{x_1, x_2, x_3, \ldots, x_p\} \) is presented to the network, the input layer transfers these \( p \) values to all the pattern layer nodes. This second layer is fully connected to the input layer, with one neuron for each pattern in the training set. In each node of the pattern layer, the distance \( Z_i \) between the input vector \( x \) and the target vector \( x_i \), is calculated by some norm. This value is the input of a transfer function \( g \) which output is the output of the node. The transfer function most widely used is the Parzen's probability density function (PDF) estimator, with a Gaussian kernel function. Parzen proved that the estimated PDF of a population converges to the actual PDF as this population increases, independently of the population presenting or not a normal distribution. The transfer function of each neuron \( i \) of the pattern layer is given by:

\[
g(Z_i) = \exp \left( \frac{-Z_i^2}{\sigma^2} \right)
\]

A Gaussian curve with unit area is determined for each example of the training set of a given class. This curve depends on two parameters: centralization (position of the pic) according to a certain metric and a scaling parameter \( \sigma \) that defines the width of the bell curve. This \( \sigma \) value is an adjustable parameter that needs to be determined for each training set.

It is possible to use other types of transfer functions. The comparative study of others functions with the Parzen function as well as the analysis of their influence in the PDF estimation are open subjects to future researches.

The type of distance, norm or metric, used in the Parzen function to compare the input vector and the target vector can have an important impact in the classification performance of the PNN, for it represents the resemblance or the proximity of both
vectors. Several different types of metric can be used. In this paper the following three are put in practice and their impact in the classification performance is analysed in section 7:

- **Euclidean distance**
  \[ Z_i = -\sum_{j=1}^{P} (x_j - x_j^i)^2 \] \tag{2}

- **Cityblock**
  \[ Z_i = -\left( \sum_{j=1}^{P} |x_j - x_j^i| \right)^2 \] \tag{3}

- **Dot product**
  \[ Z_i = x \cdot x_i - 1 \] \tag{4}

The transfer function, shown in equation (1), uses only one \( \sigma \) for all the samples of the training set, and is named monosigma case. Not all the samples have the same influence in the estimation of the real PDF, and therefore, Cacoullos [8,15] extended the Parzen method to the multisigma case. In this new approach, each sample of the training set has its own parameter \( \sigma_i \). Therefore, equation (1) takes the form:

\[ g(Z_i) = \exp \left[ \frac{Z_i}{\sigma_i^2} \right] \] \tag{5}

This approach can improve the performance of pattern classification, but has the cost of increasing the training time since, now, it is necessary to find a set of \( \sigma_i \) that maximizes the classification error. Therefore, to train a PNN means to find a \( \sigma \) or a set of \( \sigma_i \) that maximizes the pattern classification in a supervised training operation.

The output of each neuron of the class \( j \) of the pattern layer is added at the summation layer and is available at the node \( S_j \):

\[ S_j = \sum_{i=1}^{n_j} g_{ji}(Z_i) \] \tag{6}

where \( j \) is the class number, \( i \) is the neuron of the class \( j \) and \( n_j \) is the number of examples of the class \( j \).

At the output layer, the \( S_j \) values are then compared. The index \( j \) of the maximum \( S_j \) among the \( S_1, S_2, ..., S_k \) values indicates to which class the input vector belongs.

### 3. Genetic Algorithm

The Genetic algorithm (GA) is a mathematical tool for optimization of complex systems. The algorithm seeks to mimic mathematically the species biological evolution with the characteristics and advantages of this random natural process [16,17]. The set of possible solutions of a given optimization problem is named as the population of individuals. For an individual, the degree of adaptation to the environment (or the proximity of the solution) is its fitness. Each individual is represented by its chromosomes; which are composed by genes. Each gene is a code related to each problem variable.
The general implementation of a GA is performed in five basic steps: initializing population; evaluating fitness, creating a new population by reproduction, mutation, and selection of a new population. The reproduction phase seeks to generate a new population by means of a combination of the previous ones. In order to avoid the local minimums in the optimization problem, it is used the mutation phase that creates small perturbations in the genes of the new population. After reproduction and mutation a new population is formed (by the tournament selection, in this paper) and its fitness is evaluated. Over successive generations, the population evolves toward the optimal solutions or, in other words, until the fitness stopping criterion is reached.

The Genetic Algorithm has several advantages over traditional optimization procedures: it is not necessary to know the response surface gradient; the algorithm is relatively immune to discontinuities; local minimums do not affect the algorithm efficiency; the algorithm generates a population of solutions points at each iteration (while the standards algorithms give one point at each iteration), and it employs a random choice to go to the next population (while the standard procedures use a deterministic computation).

The values of the mains GA parameters used in this paper are: size of the population equal to 10, size of chromosomes equal to 2; mutation rate equal to 0.001 and number of generations equal to 20. These parameters were determined by a trial and error procedure, for there is not an established rule orientating their calculation.

4. The Experimental Database

A test rig was developed in order to build a database of faulty conditions in rolling bearings. An AC motor, that drives a shaft in which the rolling bearing was set up, composes the experimental apparatus. Connected to the bearing housing there is a mechanism that applies a known radial loading over the rolling bearing. A variable-speed driver controls the AC motor speed.

Rolling bearing faults may appear as a consequence of several problems: lubrication inadequacies and contamination, improper storage or installation, etc. One way to classify these faults is by means of the size of the surface defects. In order to simulate different defects sizes, four different fault types were introduced in the outer race or in the
inner race of the rolling bearing: a pit (punctual size) in either the outer race or the inner race, a localized corrosion produced by synthetic sea water during 8 and 24 hours (medium size) in the outer race or inner race and a scratched surface (distributed all over the outer race or inner race surface). Therefore, eight different types of defects were generated. A perfect rolling bearing was also used to represent the normal condition. Each one of these nine patterns is named by a code, which indicates: N for normal, Si for scratched inner race, So for scratched outer race, Pi for pit inner race, Po for pit outer race, C1i for the corrosion during 8 hours inner race, C1o for the corrosion during 8 hours outer race, C2i for the corrosion during 24 hours inner race and C2o for the corrosion during 24 hours outer race.

Vibration signals from the bearing housing were collected using a piezoelectric accelerometer, mounted vertically on the top of the bearing housing. The accelerometer was connected to an amplifier, with a low pass filter with 2 kHz of cut-off frequency. An A/D converter device digitized the filtered signal, with a sampling frequency of 5 kHz. Each signal was represented by 2048 points.

The signals were collected for each one of the nine defects, considering six different shaft speeds (400, 600, 800, 800, 1000, 1200 and 1400 rpm) and three different conditions of radial loading (200, 400 and 600N).

5. Pre-Processing Methodology

The pre-processing phase has the main objective of providing the adequate feature representation of each fault pattern to be presented to the neural network. The way in which that feature representation is carried out has an important impact in the classification performance, in the network complexity and computational effort [5,9,10,12].

Two training sets, built with different signal processing methods, were used to evaluate the PNN classification performance. In order to test the trained PNN, two test sets were built in the same way that the two training set.

In the first one, the teaching element of the training set is composed by the Power Spectral Density (PSD) of the vibration signals. These spectra were calculated by using the Welch method [18] with time Hanning window with length of 128 points, which implies in a frequency resolution of 15 Hz. Therefore each element of this training set is composed by a spectral vector of 64 points length. For this raison, this training set is referred in the text as vectorial training set.

Each element of the second training set is composed by a blend of the following scalar parameters:

a) Signal RMS [19]:

\[
RMS = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2}
\]  

where N is the signal length in number of points.

b) Signal Residual Energy [20,21]:

\[
RE = \int_{0}^{B} |S_d(f) - N(f)| df
\]  

where \(S_d(f)\) is the PSD of a signal collected in a probable faulty condition, and \(N(f)\) is the PSD of reference signal, collected in a good condition, and \(B\) is the frequency band.
c) Signal Kurtosis [19]

Kurtosis is the normalized fourth statistical moment:

$$K = \frac{1}{N\sigma^4} \sum_{i=1}^{N} x_i^{-4}$$

(7)

where $\sigma$ is the standard deviation of the signal $x$.

d) RMS, Kurtosis and RE of the signal envelope [19,21]:

The envelope of the signal is the Hilbert Transform (HT) of the signal:

$$\text{env}[s(t)] = |HT[s(t)]|$$

(8)

Once the envelope of the signal has been obtained, the RMS, Kurtosis and RE parameters are calculated.

Therefore, each component of the second training set is a 6 length vector whose elements are the signal RMS, signal RE, signal Kurtosis, envelope RMS, envelope Kurtosis and envelope RE.

6. Training Methodology

In order to verify the influence of the type of metric in the classification performance of the PNN, three metrics were used, as described in section 2. Each one of these metrics was trained with the two training sets, and tested with the corresponding test set.

Another important PNN characteristic that was studied is whether the use of one smoothing parameter for all elements of the training set (monosigma case) or that of one smoothing parameter for each element of the training set (multisigma case). In both cases, the genetic algorithm was employed to optimize the set of smoothing parameters that minimizes the classification error.

In addition, the following classification strategies were used:

a) Detection: in this strategy, a PNN was trained to identify if there is fault or not. This is the simplest classification problem;

b) Identification: in this strategy, a PNN was trained to identify three fault cases: normal (no fault), fault in the outer race, and fault in the inner race;

c) Classification: in this strategy, a PNN was trained to identify each one of the nine patterns cited above. It is the most complex of the classification problems.

In each one of these classification strategies the used training sets were composed by all the nine fault features, and the three radial loads for each one of the six shaft speed.

7. Results and Discussion

In order to evaluate the influence of the metrics and the monosigma and multisigma cases in a practical study, several PNN were trained in different ways. The use of different classification strategies, and of different training set description, forms a more generic background for analysing the importance of the metrics as well as that of the used case of sigma. The results presented in Tables 1, 2 and 3 are the classification rate of the tests obtained with these trained PNN. These results were obtained using a database composed by signals measured at 400 rpm of shaft speed. For the other shaft speeds the results are similar.

Table 1 presents results obtained for the monosigma case. These results were not obtained with the GA algorithm but with a simple bisection method. From this Table it is
possible to compare the results for the three metrics in the three classification strategies, using vectorial and scalar training sets. The euclidian and cityblock metrics give similar good results while the dotprod metric gives the worst. Regarding the training set form, the vectorial one results in a better classification rate than the scalar form.

The results shown in Table 2 refer also to the monosigma case. They were obtained in the same conditions of those from Table 1, excepted that a GA was used to optimize the solution. Finally, Table 3 presents the multisigma case.

The best results, for all the Tables are highlighted in Table 2. In all the classification strategies, 100% of classification hints were reached, by using the vectorial training set, the cityblock metric and the monosigma case optimized by the GA.

On the whole, the simplest classification strategy (Detection) gives the best results, as expected. Regarding the training set description, the vectorial set gives the best results, which is also expected since it describes the best the information carried by the signals. Concerning the metric type, it is possible to notice that, in general, dotprod gives the worst results. Only in the simplest classification strategy this metric has a performance comparable with the two others. The euclidian and cityblock norms give similar good results, whose performance depends on the classification strategy (or complexity of the classification problem) and on the sigma case.

Comparing Table 2 and 3, it is possible to observe that the multisigma case gives worse results when compared with the monosigma case. This fact was not expected, since, in principle, the use of one sigma for each element of the training set should have modeled the element statistical contribution better than the use of only one (the same) sigma for all the elements. Futures researches need to be done to allow clarifying this point.

Table 1 – Classification rates for the monosigma case

<table>
<thead>
<tr>
<th>Metric</th>
<th>Strategy</th>
<th>Euclidian</th>
<th>Cityblock</th>
<th>Dotprod</th>
<th>Metric</th>
<th>Strategy</th>
<th>Euclidian</th>
<th>Cityblock</th>
<th>Dotprod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>88,80%</td>
<td>88,80%</td>
<td>88,80%</td>
<td>Detection</td>
<td>98,15%</td>
<td>98,15%</td>
<td>88,89%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>83,30%</td>
<td>86,00%</td>
<td>44,00%</td>
<td>Identification</td>
<td>98,15%</td>
<td>98,15%</td>
<td>41,67%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td>63,80%</td>
<td>69,44%</td>
<td>11,00%</td>
<td>Classification</td>
<td>99,07%</td>
<td>98,15%</td>
<td>5,56%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 – Classification rates for the monosigma case by using GA

<table>
<thead>
<tr>
<th>Metric</th>
<th>Strategy</th>
<th>Euclidian</th>
<th>Cityblock</th>
<th>Dotprod</th>
<th>Metric</th>
<th>Strategy</th>
<th>Euclidian</th>
<th>Cityblock</th>
<th>Dotprod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>88,80%</td>
<td>88,80%</td>
<td>88,80%</td>
<td>Detection</td>
<td>97,22%</td>
<td>100%</td>
<td>88,89%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>69,40%</td>
<td>80,60%</td>
<td>22,20%</td>
<td>Identification</td>
<td>99,07%</td>
<td>100%</td>
<td>37,04%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td>52,78%</td>
<td>63,89%</td>
<td>16,67%</td>
<td>Classification</td>
<td>97,22%</td>
<td>100%</td>
<td>3,70%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 – Classification rates for the case multisigma by using GA

<table>
<thead>
<tr>
<th>Metric</th>
<th>Strategy</th>
<th>Euclidian</th>
<th>Cityblock</th>
<th>Dotprod</th>
<th>Metric</th>
<th>Strategy</th>
<th>Euclidian</th>
<th>Cityblock</th>
<th>Dotprod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>88,80%</td>
<td>88,80%</td>
<td>88,80%</td>
<td>Detection</td>
<td>88,80%</td>
<td>88,80%</td>
<td>88,80%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>83,30%</td>
<td>86,10%</td>
<td>44,40%</td>
<td>Identification</td>
<td>54,63%</td>
<td>72,20%</td>
<td>44,40%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classification</td>
<td>61,10%</td>
<td>61,10%</td>
<td>16,70%</td>
<td>Classification</td>
<td>35,19%</td>
<td>20,37%</td>
<td>18,52%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8. Conclusions

This paper presented a fault classification study in rolling bearings based on the use of the probabilistic neural network. The influence of the PNN metrics and that of the cases of mono and multi smoothing parameter sigma was analyzed. A genetic algorithm was used in order to optimize the sigma values for a good classification performance.

The classification problems analyzed in this work are much more complex than those found in the Industry. The training and test sets included nine patterns (faults), six different shaft speeds and three radial loads. Three classification strategies with increasing complexity were employed: detection (there is or not a fault), identification (fault in inner or outer race) and classification (each fault in a set of nine different types). Two pre-processing methodologies were used: one where the training set is composed by the Power Spectral Density of the vibration signal, and another composed by a blend of scalar parameters.

It worth noting that excellent results were obtained since 100% of correct faulty classifications was reached.

Regarding the importance of the metrics, two of them, the euclidian and the citiblock, led to good similar results. On the contrary, the Dotprod did not show a satisfactory performance. In what concerns the mono and multi sigma cases, the mono case gave surprisingly better results unlike it was expected, since the multi sigma could have better modeled the statistical contribution of each element of the training set. A further study needs to be carried out in order to analyze the reasons for this behavior.

On the other hand, the GA proved to be an efficient optimization algorithm, which can be adapted to the PNN topology with a good performance. A further step will be to adapt the GA to optimize a third case of sigma, one for each category or class of classification space.

Acknowledgment
The authors wish to thanks CAPES (Procad 0136/01-8) for the financial support to this work.

References

[21] Padovese, L.R.; Rocha, L.F.R.; Improving fault diagnostics by vibration analysis in industrial plants by using residual signals; 5th Int. Conf. Acoustical and Vibratory Surveillance Methods and Diagnostic Techniques; October, 2004, Senlis, France