Proceedings of the American Control Conference
Albuquerque, New Mexico June 1997
0-7803-3832-4/97/$10.00 © 1997 AACC

NEURAL NETWORKS IN FAULT DETECTION: A Case Study *

D.R. Hush, C.T. Abdallah, G.L. Heileman, and D. Docampo*
EECE Department, University of New Mexico, Albuquerque, NM 87131, USA.
* Departamento de Tecnologias de las Comunicaciones
ETSI Telecomunicacion, Universidad de Vigo, 36200-VIGO, SPAIN

Abstract

In this paper we study the applications of neural nets in the area of fault detection in real vibrational data. The study is one of the first to include a large set of real vibrational data and to illustrate the potential as well as the limitations of neural networks for fault detection.

1. Introduction

There has been considerable work in the areas of fault detection and isolation which were reviewed in [2], [6], [3]. There are basically two ways to approach the analytical fault detection problem: The model-based approach and the data-based approach. In the model-based approach, the engineer has access to a model of the system whose behavior is being monitored. In the data-based approach one bypasses the step of obtaining a mathematical model and deals directly with the data. This is more appealing when the process being monitored is not known to be linear or when it is too complicated to be extracted from the data. It is this approach which we will concentrate on in this paper in order to evaluate the potential of neural networks as fault detectors. This paper discusses the potential, as well as the limitations, of neural nets usage in fault detection and possible accommodation in vibrational systems. Section 2. provides a study of NN in fault detection where our approach to the fault detection and isolation problem is presented and our results obtained using real data are given. We have also investigated a trending approach using Fuzzy ART, the results of which are presented in section 3. Finally, our conclusions and recommendations are provided in section 4.

2. Detecting Faulty Bearings

It is conceivable that a neural net can be used as a monitoring device, in order to detect major changes in the operation of the system. In some cases, the neural net may also be used to accommodate the change of behavior as part of hierarchical control system [5]. In others, it is used simply as a fault detection device where the clustering capabilities of networks such as ART or CMAC are called upon. The general idea behind using a NN for fault detection can be summarized in the following steps: 1) Use a signal processing techniques to obtain a figure of merit f (Spectrum, Cepstrum, k factor, etc) for the different time signals. 2) If the figure of merit is high dimensional, use a feature extraction algorithm to reduce its dimensionality while keeping most of its information content. The resulting signal is f_e. 3) Train the neural network on f_e either in supervised or unsupervised mode as discussed below. The purpose of this study is to explore automated methods for detecting faults in the viscous damper bearing of a helicopter drive shaft. This portion of the paper describes the design of a system that accomplishes this task using *neural networks.*

Data Description: The data used in this portion of the study are *time series data* measured by an accelerometer located on the outer bracket housing of the helicopter shaft (measured with respect to the shaft 1P). The time series were sampled at a rate of 44,100 sps. The bandwidth of the anti–aliasing filter was approximately 20 KHz. Spectrograms of the data (shown below) suggest that the signals are *stationary*, and that they were sufficiently oversampled. A total of 21 runs were made using 17 different bearings. The duration of the runs varied from 4 to 7 seconds. Runs 1 and 11 were discarded because they represent "start–up" runs from the two different data gathering trips, and were both suspect in one way or another. Thus, the input data for this portion of the study consists of time series data collected from the outer bracket accelerometer for Runs 2–10 and 12–21. All experiments described in this section used 4 seconds of data from each run. Approximately 2.8 seconds (70%) were used for training and 1.2 seconds (30%) for testing. Table 1 summarizes the two levels of groupings for the data.

System Design Our focus in this paper is on the Level II grouping (i.e. *good* verses *bad*). It has been our experience (as well as others [4]) that there is wide variability in the signature produced by both good and faulty bearings and that the overlap in these signatures is quite large. This problem stems from the fact that there are *numerous factors* that contribute to the the variability among bearing signatures, and the *quality* of the bearing is *only one of these factors* (and not always the dominate one). The basic approach followed here is outlined in Figure 1. This is a traditional pattern recognition system consisting of three

*The research of D. Hush, C.T. Abdallah, and G. Heileman was supported by a Grant from Chadwick-Helmuth under Contract W-300445. The authors are grateful to JP Cain of Chadwick-Helmuth for his help in collecting the data.

Run	Level I	Level II
7,12,14	New	Good
8,13,15,16	Used	Good
2,9,17,19	Ball Spall	Bad
10,18,20	Inner Race	Bad
5	Outer Race	Bad
6,21	Other	Bad

Table 1: Summary of Categorizations for Runs 2–10 and 12–21.

Figure 1: Pattern Recognition System.

major components: preprocessing, feature extraction and classification. The only preprocessing performed on the bearing data was to normalize each of the runs so that it had zero mean and unit standard deviation. The second stage is generally the most difficult to optimize. The purpose of feature extraction is "to extract features from the preprocessed data which provide the *greatest discrimination* between pattern classes". This is often difficult because the "best features" are usually not known ahead of time. To this end we settled on the following three feature sets: *Spectrograms:* A time–varying estimate of the magnitude of the Fourier Transform of the data. *Linear Prediction Coefficients (LPC):* Coefficients of an optimal M^{th} order (FIR) linear predictor for the data. *Cepstrum:* The inverse Fourier transform of the logarithm of the magnitude of the Fourier transform of the data. In addition to their ability to carry discriminatory information, a good feature set should also have the following properties: **Invariance:** The features should be *invariant* to superfluous variations in the data. **Dimensionality Reduction:** The training process and generalization performance of the classifier (the last stage) suffer from the curse of dimensionality. It is therefore important to reduce the dimensionality (i.e. the number of features) as much as possible at the feature extraction stage. **Simplified Representation:** Ideally, the features should take on a representation that permits optimal discrimination with the simplest possible classifier.

The final stage in Figure 1 is the classifier. We investigated a wide variety of classifiers in this study, with a focus on the following *neural network* classifiers: Multilayer Perceptrons (MLPs), Radial Basis Functions (RBFs), and Fuzzy ARTMAP. In addition to these we investigated the traditional *linear*, *quadratic* and *nearest–neighbor* classifiers.

Feature Computation The computational aspects of the three feature sets, Spectrograms, LPC coefficients and Cepstral coefficients, are described in the full study [1].

Feature Set	Input Dimension	Projected Dimension
Spectral	16	9
LPC	16	13
Cepstral	16	13

Plots of one representative spectrogram of the 19 runs used in this study is shown in Figure 2. The complete runs are described in [1].

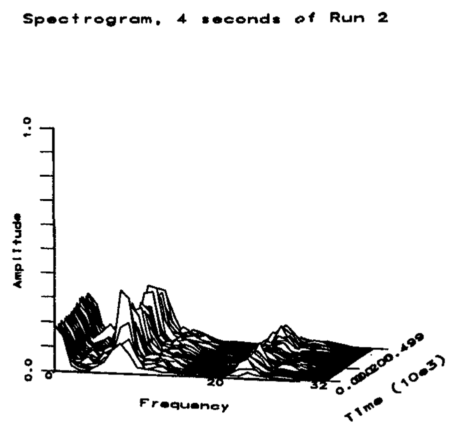

Figure 2: Run 2 (Ball Spall)

Feature Selection The purpose of feature selection is to determine which of the *individual* features in the three feature sets are useful for discrimination. The method used here forms an estimate of the Bayes Classification Error (i.e. the minimum attainable classification error) for each individual feature, and discards features with a high (close to 50 %) error.

Dimensionality Reduction After discarding individual features, we reduce the dimensionality of the features retained by projecting them to a lower dimensional space. We use a *linear* method that projects feature vectors onto directions with the largest *separability* The overall projections, are summarized in Table 2..

Classifier Design All of the training and testing performed in this section and the next assumes that the occurrence of good and bad bearings is equally likely, i.e. the prior probability for both of these events is assumed to be 0.5. A critical component in the design of the 1–NN, RBF and MLP classifiers is the determination of their size. A sequential search was performed, starting with the smallest possible size and then increasing it until the classifier performance began to level off. A specific size (and corresponding performance) must be chosen for the 1-NN, RBF and MLP classifiers so that they can be compared

Optimal Size Classifiers for Spectral Features		
Classifier	Size	% Error
1–NN	6	27.77/26.09
RBF (LS)	18	33.75/34.44
RBF (Pocket)	18	27.76/27.39
MLP	10	23.79/23.73

	% Error	
Classifier	Training	Test
Linear (LS)	24.65	24.26
Linear (Pocket)	22.07	22.63
Quadratic	24.09	24.56
1–NN (LVQ)	27.77	26.09
RBF (LS)	33.75	34.44
RBF (Pocket)	27.76	27.39
MLP (BP)	23.79	23.73
MLP (Constructive/LS)	22.56	23.83
MLP (Constructive/Pocket)	21.26	22.15
Fuzzy ARTMAP	4.2	27.5

Table 2: Classification Results for Spectral Features.

with the others in the next section. Table 2. shows the sizes chosen for this purpose. Similar results were obtained for the LPC and Cepstrum features [1]. **Classification Results** The classification results for the Spectral feature set are summarized in Table 2. The best classifier for this feature set is the *linear classifier trained with the Pocket algorithm.* Although one of the MLP networks achieved slightly better performance, the additional complexity of this classifier does not justify its choice given such a small increase in performance. The classification results for the LPC and Cepstral features may be found in [1].
Tests on Unknown Bearings In this section we describe the results of tests performed on Runs 3 and 4 whose true classifications are unknown (but believed to be "normal"). Neither of these runs were used in the training and testing above. Both runs were processed by the three different pattern recognition systems (one for each feature type). In each system the optimal classifier (determined in the previous section) was used, i.e. the linear, MLP and RBF classifiers were used for the Spectral, LPC, and Cepstral features respectively. The classification results are summarized in Table 3. Several observations can be made regarding these results. 1) There is a large variation across feature sets. 2) There is a large variation in the consistency of the results between Run 3 and Run 4. 3) None of the results are in close agreement with the classification error rates predicted in previous sections. The explanation for these poor results is that the data used to design these systems was not representative of all future data. These results tell us that the systems designed here are not likely to produce meaningful classifications of future data. A more promising approach is investigated in the next section.

Run	Feature	% Good	% Bad
Run 3	Spectrum	25.2	74.8
	LPC	85.4	14.6
	Cepstrum	14.4	85.6
Run 4	Spectrum	32.2	67.8
	LPC	35.0	65.0
	Cepstrum	60.1	39.9

Table 3: Classification Results for Runs 3 and 4.

3. Trending Using Fuzzy ART

We next investigated a trending approach to the detection of faulty bearing. In order to perform a trending analysis, it is necessary to monitor the *same* bearing over an extended time period. In four of the runs supplied to us, the same bearing was used; these include Runs 13 (normal), 16 (normal), 18 (inner race), and 20 (inner race severe). Note that the last two runs correspond to damaged bearing.

Our experimental setup for the trending analysis consisting of a single fuzzy ART (clustering) module. The spectrogram data computed for Run 13 was supplied as input, and the network formed a single cluster. A histogram of the output node value for this cluster, for each of the 500 time time segments is shown in Figure 3. Notice that the output node produced the same value for all 500 inputs. At this point the fuzzy ART network weights were "frozen", and the spectrogram data computed for Run 16 was supplied as input. The histogram for Run 16 is shown in Figure 4. This histogram is only slightly different from the previous one (i.e., it is nearly flat). The fuzzy ART histogram corresponding to the spectrogram data computed for Run 18 is shown in Figure 5. In this (damaged bearing) case, the histogram is significantly different from the previous two. For the final case (severe damage), Run 20, the histogram (Figure 6) is extremely "ragged". This experiment suggests that a trending approach, which is based on learning the features associated with a normal bearing, and monitoring these features over time, may be a feasible alternative to the other experiments performed in this study.

4. Conclusions and Future Directions

We have presented an overview of fault detection and isolation techniques with special emphasis on vibrational data and the usage of neural networks. We have presented applications of these techniques to real vibrational data. Based on this study, we have determined that the approach which presents the best probability of success is the trending approach where a particular system is monitored over its lifetime and faults are detected as deviation

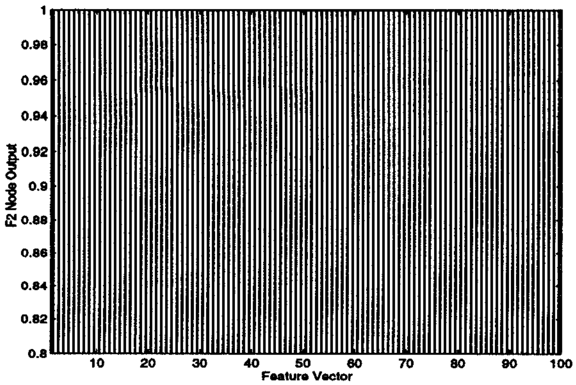

Figure 3: Histogram of Fuzzy ART Output Values for Spectral Features for Run 13.

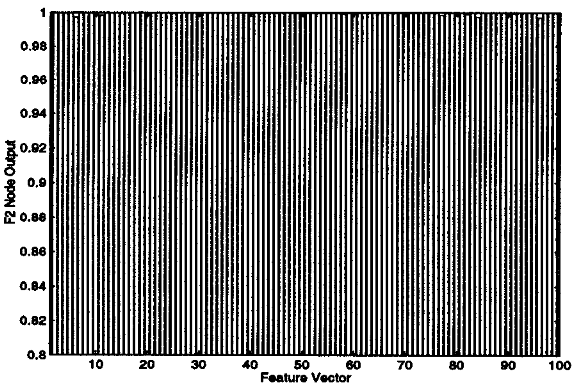

Figure 4: Histogram of Fuzzy ART Output Values for Spectral Features for Run 16.

from normal behavior. On the other hand, an approach relying on combining data from different systems is doomed to failure as shown in section 2.. As a direction of future research, we intend to study the trending approach using different feature sets and different neural networks structures.

References

[1] C.T. Abdallah, D.R. Hush, G.L. Heileman, and D. Docampo. Neural networks results for fault detection applications: Survey and implementation. Technical Report EECE-96-001, University of New Mexico, Albuquerque, NM, 1996.

[2] P.M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundacy–a survey and some new results. *Automatica*, 26:459–474, 1990.

[3] R. Isermann. Process fault detection based on modeling and estimation methods. *Automatica*, 20:387–404, 1984.

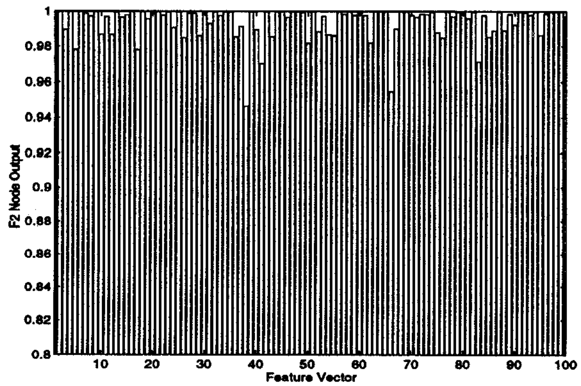

Figure 5: Histogram of Fuzzy ART Output Values for Spectral Features for Run 18.

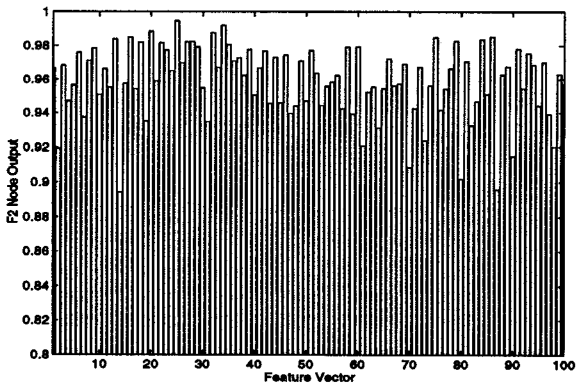

Figure 6: Histogram of Fuzzy ART Output Values for Spectral Features for Run 20.

[4] T. Petsche and et. al. A neural network autoassociator for induction motor failure prediction. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 8*. Morgan Kaufmann, 1996.

[5] M. Polycarpou and A. Vemuri. "Learning methodology for failure detection and accomodation. *IEEE Contr. Sys. Mag.*, 15(3):16–24, 1995.

[6] A.S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.