

# Automatic failure detection and monitoring of ventilation and cooling systems\*

Balázs Bánhelyi, Balázs L. Lévai

University Of Szeged, Institute of Informatics  
bánhelyi@inf.u-szeged.hu, levaib@inf.u-szeged.hu

## Abstract

A ventilation and cooling service firm posed the problem to develop a computational system for detecting certain types of machinery failure based on regularly collected sensor data. In this paper, we present this industrial sample classification problem, the structure and operation of the developed solution including the basic theoretical background. In addition, we discuss the possibilities of reducing the number of monitoring devices with an acceptable loss of detection rate. The test results of our technique and its future applicability are also presented in detail.

*Keywords:* sample classification, monitoring, failure detection

*MSC:* 94-06

## 1. Introduction

Nowadays, due to the general market conditions and the ever increasing competition, industrial companies are pushed more than ever to optimize their operational costs, and improve the product and service related business processes. In addition to the regular requirements, service continuity and availability are also expected. Maintaining a high level quality requires the detection of occasional failures of the included service elements as soon as possible. Consequently, the automatization of this aspect of support could be essential to minimize the time, personnel, and financial costs in a company's everyday life.

A local service company, which maintains and installs ventilation and cooling systems for a wide range of customers, including hospitals, office buildings, and

---

\*The publication is supported by the European Union and co-funded by the European Social Fund. Project title: "Telemedicine-focused research activities on the field of Mathematics, Informatics and Medical sciences." Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073. The authors are also grateful to Szilveszter Pletl and Zoltán Kincses (University of Szeged) for building the monitoring hardware.

private homes too, asked us to develop a computer system for failure detection purposes in cooperation with the engineering department of our institute.

Considering the operation of cooling and ventilation systems, one may see or even hear that malfunctions affect and alter the normal vibration caused by the moving parts, mainly the engine and the fans, therefore it is a sound way to monitor such systems by motion sensors. These regularly measure the vibration over a fixed-length of time interval in terms of signed one dimensional acceleration. The moving parts work on specific frequency which means that the collected data of each sensor is a periodic one dimensional time series, or signal in other words.

Our task is the classification of the current condition of the monitored system based on the information of the last measurement using predefined malfunction classes. This is a supervised learning problem called sample classification, therefore we need to design and implement a feature extraction method and a classifier in order to create a solution.

## 2. Feature extraction

The first step in a classification process is to extract relevant distinctive information from the raw data. This information form is called a feature vector or sample which has to be unique and characteristic to every single class.

Periodic signals are studied by frequency analysis. It is based on the theorem that every periodic signal can be created by summing up sinusoids with the appropriate frequency and amplitude. Theoretically, an infinite number of different sinusoids should be required to perfectly recreate a complex signal in most cases. In practice, we only determine the frequency and amplitude of a finite number of sinusoids. While the former could be accomplished by Fourier Transformation, the latter is executed by Discrete Fourier Transformation, or DFT in abbreviation, a realizable finite version of the theoretic technique.

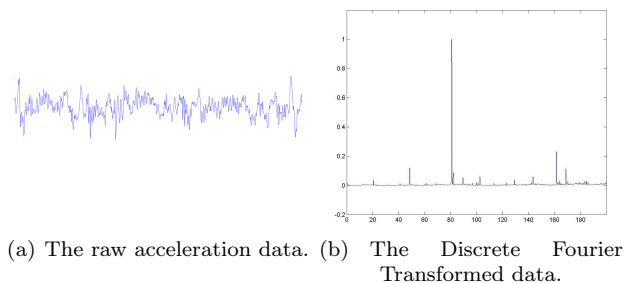


Figure 1: Illustration of frequency analysis.

Discrete Fourier Transformation calculates the amplitude of these sinusoids with a frequency from a specific range whose lower bound is zero, while its upper bound and the number of the sinusoids in the result depend on the parameters of the

input data, namely the length and the sampling rate in case of time series. For detailed information about frequency analysis see [2]. The output of the DFT is called the frequency decomposition of the studied signal. From now on, we will call its elements as the components of the signal. We will also refer the amplitude and the frequency of a component as the value and the place of it.

We use the DFT of every time series in the monitoring data to compose a sample. First and foremost, let us have a look over some of these decompositions from the reference, corresponding to the normal operation, and every malfunction class to get a basic picture of the overall problem. Recommended by the engineer colleagues, we narrow down the study to the frequency range between 20 Hz and 200 Hz. One may have two conclusions immediately. First, the decompositions in this interval have only a few dominant components, whose value is relatively high compared to the others. Second, there are remarkable differences between the dominant components of different classes: a component can be dominant in one class while completely insignificant in another one, or its place or value is slightly, but recognisably, changing from class to class.

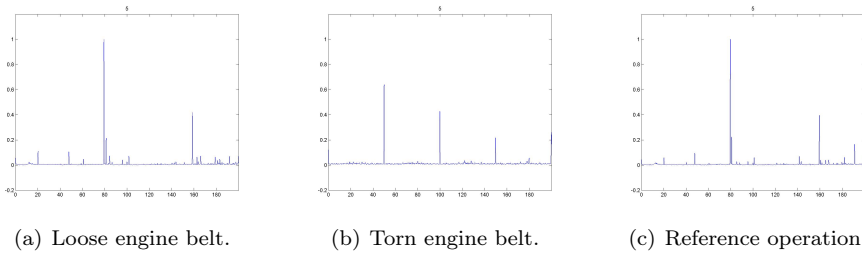


Figure 2: Discrete Fourier Transformations of engine belt problems.

In order to create a feature vector, which reflects the observations above, we adaptively mark out intervals as regions of interest around the dominant frequencies. These intervals are disjoint and also varying in width. Their actual position depend entirely on our previous manual study of the decompositions. Instead of our human instinct, some sort of machine learning algorithm can be used to determine the intervals.

In each interval, we simply choose the place and the value of the most dominant component to be features. We could use more sophisticated characteristics, like the energy, but these ones turned out to be the most reliable ones according to our experience. Eight to ten intervals on average, two feature elements in each, proved to be sufficient for classification.

### 3. Classification

The samples are classified by neural networks. They mimic the structure and operation of the human nerve system. A network consists of neurons arranged into

layers. The classical artificial neurons are individual computational units which calculate the weighted summation of the values they are given, and return a pre-defined function of it. In a feed-forward network constructed for classification, one gives the sample, containing numerical data, to the neurons of the first or input layer. When the computation is done, their output is passed to another layer whose neurons repeat this process. This data flow continues through subsequent layers until it reaches the final, output layer in which every neuron belongs to a class. The output of a neuron is the membership of the sample in the represented class. The intermediate layers are called hidden layers, because we cannot interact with them directly.

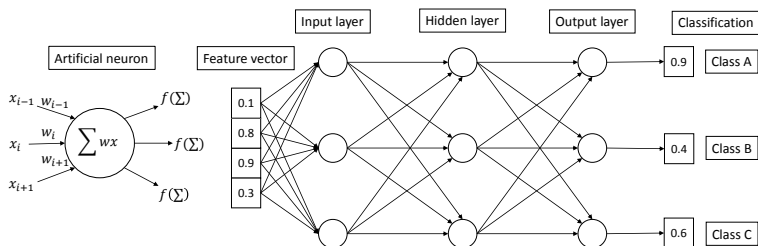


Figure 3: Illustration of the operation and structure of a neural network for classification.

Networks are configured to a specific classification task through the weights of the neurons. This configuration process is called the training of a network. We used back propagation, a common training method. It is an iterative algorithm whose point is that we use the network to classify samples for which we know what should be the correct output. We compare the network actual output to the expected one and based on its error we modify the weights in such a way that repeating the calculations with the same samples would result in less error. This process is iterated until the error of the classification is acceptable or does not improve significantly (or at all). You can read more about neural networks and classification in [1].

Our classifier networks use two hidden layers beside the input and the output one. We initialize the weights by the Nguyen-Widrow method which generates values uniformly from a defined interval and transforms them to obtain a steep, Gaussian like distribution. During training, the weight modifications are made according to the scaled conjugate gradient algorithm. It performs better than the other alternatives especially when we do not have prior information about the proper parametrization. We experienced that the performance of the network depended heavily on the initialization, therefore training multiple networks and using the best one is advisable. Considering the training as a global optimization task, the main difficulty of this problem is probably the large number of local optima in the space of allowed weight settings.

## 4. Results

We tested our detection technique implemented in MATLAB [3] on a real ventilation and cooling system designed for larger buildings like state offices. The entire system has four vital components, two ventilation fans and two condensers, separated from each other either in the operation and the location, thus we monitor them accordingly.

Malfunction	Class	Malfunction	Class
A bit loose engine belts	1	Very loose engine belts	2
One torn engine belt	3	Two torn engine belts	4
Loose drive fastening	5	Loose Fixation	6
Imbalanced device	7	Half-sealed shutters	8
Totally sealed shutters	9	Stuffed outer calorifer	10
Loose fan structure element	11	Jammed shutters	12
Worn bearings	13	Normal (reference) operation	14

Table 1: Error classes of the supply ventilation fan.

In the following, we present the results in part with the most complex context, the supply ventilation fan. Table 1. shows the 13 possible malfunctions we have to detect. These problems mainly concern the belts of the fan driving engine, the fan itself, the shutters, and the bearings.

10 sensors grouped into sets of size 2 and 3 composed the monitoring hardware. Sensors in the same group were positioned close to each other but directed orthogonally. First, assessing the capabilities of the elaborated detection method, we use the data of only one sensor, or all of them at a time. Table 2. shows the results. As it might be expected, a single sensor is far not enough to detect all of the malfunctions. Surprisingly, even the combined data of the ten sensors is not sufficient to achieve a 90% detection rate at least in certain cases. Possibly, we have not repeated the training enough times, and the best weights we found correspond to a rather worse local optimum. Another reason is that we are dealing with overlapping classes, and we have reached our limits.

On Figure 2. you can notice that a loose engine belt can look very similar to the normal operation. It is difficult to separate these two classes with a hyperplane. It is up to us what extent of loosing should be considered a problem. The development of some malfunctions is transitional and not instantaneous.

The industrial partner was curious about the possible reduction of the monitoring hardware to cut costs. Table 2. contains some results of the research on this question. However, the overall rate of correct classification improves, if we use an entire sensor group instead of only one element from it, there is obviously some redundancy in the provided information. Examining the best combinations of sensors or sensor pairs from every group, significant improvements can be experienced, however even the most excelling set (1,3,5,6,7,9) is far behind the performance of using every sensor, therefore the modification of the monitoring hardware is not recommended.

Finally, if we treat the classes affecting the same part of the machinery as a

	1	2	3	4	5	6	7	8	9	10	4,5,6	9,10	1,3,5 6,7,9	1-10
1	0	84	80	68	46	100	72	76	62	60	88	56	100	100
2	60	75	73	85	74	68	75	74	66	76	89	81	87	85
3	56	64	74	98	78	14	74	58	66	66	94	88	100	98
4	100	84	82	98	100	100	100	100	100	100	98	100	98	100
5	40	74	82	50	86	88	84	72	62	48	100	92	100	100
6	49	60	51	60	62	67	67	75	64	64	82	70	90	91
7	69	76	79	57	71	67	65	66	77	79	79	77	89	85
8	42	62	46	78	50	72	64	58	60	82	96	76	68	100
9	84	70	38	80	98	84	94	90	86	74	98	96	98	100
10	59	52	79	54	58	65	64	45	57	65	73	57	77	96
11	48	62	12	26	30	0	48	68	62	68	42	42	54	74
12	64	64	52	62	67	55	70	49	66	48	74	83	84	91
13	83	74	67	81	82	86	84	72	91	91	97	91	88	92
14	77	70	83	68	76	71	70	73	75	80	77	65	85	85

Table 2: Classification success rates in percents. Rows denote the classes while columns are representing the sensors used for detection.

single class, and we work with every sensor, the results are more than satisfying, having at least 93% success rate in every class, which proves that the shady class borders did cause our problems.

## 5. Conclusion

We have developed an analyzing technique which is capable of detecting specific malfunctions of ventilation and cooling systems. We have tested it on actually working machinery with success.

This classification works properly, although its applicability fully depends on the type of the fan driving engine. The studied system operates on a fixed frequency which makes the frequency decomposition possible. Lately, more and more systems are available in the market whose engine's frequency changes over time to fit the also changing performance requirements. They do not have a persistent frequency profile which renders our system inapplicable.

Lastly, we would like to mention that we have successfully integrated this technique into a fully automatic monitoring system prototype with even better success in which the determination of the frequency intervals and the construction of the neural network are also programmed.

## References

- [1] BISHOP, C. M. , *Neural Networks for Pattern Recognition*, *Oxford University Press, Inc. New York, NY, USA* (1995).
- [2] SMITH, S. W. , *The Scientist and Engineer's Guide to Digital Signal Processing*, *California Technical Publishing, San Diego, California* (1997).
- [3] THE MATHWORKS, INC. , *MATLAB*, <http://www.mathworks.com/products/matlab>.