ISSN 0280-5316
ISRN LUTFD2/TFRT--5820--SE

Using Structure-borne Sound Analysis to Predict Bearing Damages in High-speed Ventilators

Anders Mattson

Department of Automatic Control
Lund University
June 2008

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
June 2008

Document Number
ISRN LUTFD2/TFRT--5820--SE

Author(s)
Anders Mattson

Supervisor
Marcus Hoch, AOA Gauting in Germany
Tore Hägglund Automatic Control, Lund (Examiner)

Sponsoring organization

Title and subtitle
Using Structure-borne Sound Analysis to Predict Bearing Damages in High-speed Ventilators (Analys av ljudmätningar för att förutse skador i kullager i höghastighetsfläktar)

Abstract

By measuring structure-borne sound from bearings, a conclusion about their condition can be made.

The measurements are done with accelerometers which should be attached as close as possi-ble to the bearing. The signal from the accelerometer may have to be lowpass filtered before being A/D-converted and stored. These steps have to be done in real time.

The analysis of the signal is done afterwards. First, the frequency spectrum of the measure-ments is produced. This has to be done in a certain way to maintain the amplitudes. A window function has to be applied to reduce the effect of spectral leakage.

When the frequency spectrum is at hand, frequencies can be identified. Bearing damages cause special frequencies (rollover frequencies) which can be calculated if the geometry data of the bearing is known. The frequencies, their harmonics and sidebands of them and their harmonics can be observed in the spectrum. Other frequencies which are either depending on or independent of the rotation speed will also be found in the spectrum.

The conclusion of the bearing condition can be made by observing the amplitudes of the bear-ing frequencies or by using a broadband diagnostic method, which measure the “peakness” of the whole spectrum.Within this dissertation project, many trials were done to study the theory. In the beginning many failures disturbed the results, but the experiences from the trials made it possible to pre-vent failures later on. The analysing software was developed in Scilab which is an open sources version of Matlab.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
59

Security classification

Recipient’s notes

http://www.control.lth.se/publications/

Preface

The idea when writing this report was to make it possible for people with different kinds of background to read it. The disadvantage by this strategy is that some readers may find some parts too basic. For employees at AOA the presentation of the ventilators might be unnecessary to read. For readers who have used Matlab on regular basis, the chapters about Scilab will seem pretty uninteresting. Some chapters focus on the theory and some make it possible for the reader to follow my work.

Every section of this report starts with a short introduction. My intention with these introductions is to give the reader the possibility to choose which parts in the section that might be of special interest for her/him. I have also used a lot of references to other chapters, to make it possible for the reader to go back if she/he excluded something which is required for further understanding.

I would like to take the opportunity to thank all the people who supported me with my master dissertation, especially my supervisor Marcus Hoch at AOA and my examiner Tore Hägglund at Lund University.

Anders Mattson, F03
March 2008

Contents

1. Introduction

AOA apparatebau GmbH is mainly manufacturing high speed ventilators for cockpits, freight spaces, avionic bays and cabin areas in aeroplanes. The lifetime of the bearings used in the fans is considerable smaller than the estimated lifetime. The problem is hard to explain but it has been there for years. A breakdown of a bearing can result in large damages on the fan.

Until now the bearing is monitored by a stationary temperature sensor. When the bearing breaks the temperature will increase due to the friction. If the temperature increases above the threshold value, the fan should automatically be turned off. The system is not operating in a sufficient way. One reason is the wide range of temperatures the fan has to endure.

The technique of using structure-borne sound (vibration) measurements to diagnose bearings is widely spread in the industry. But commonly the machines to be diagnosed are operated with much lower rotational speed, than the high speed ventilators manufactured by AOA.

The objective of this master’s dissertation is to examine the possibilities of using vibration measurement to identify bearing damages in the fans as early as possible. The problem demands an interdisciplinary solution and knowledge in the fields of measurement technology, digital signal processing, statistics, software development, control engineering, mechanics and electronics are useful. The results of the dissertation will be used to implement a condition monitoring system for high speed ventilators developed for A350.

Some of the results from the vibration measurements gave explanations to the main problem, how the bearing damages occur. A discussion around different theories is to be found in chapter 9.1.

2. Measuring Objects

To get started a basic overview of the measuring objects is required. Chapter 2.1 is a brief presentation of the different ventilator models. In chapter 2.2 bearing will be introduced.

2.1 The ventilators

AOA manufactures three different types of fans in the 41 family. All of them suffer from bearing problems.

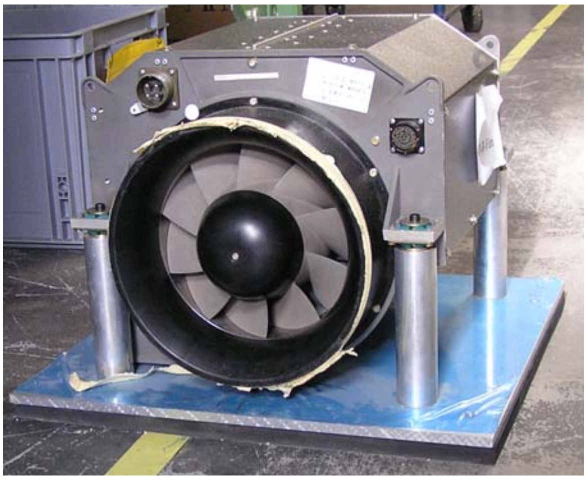


figure 1, front side VA4111

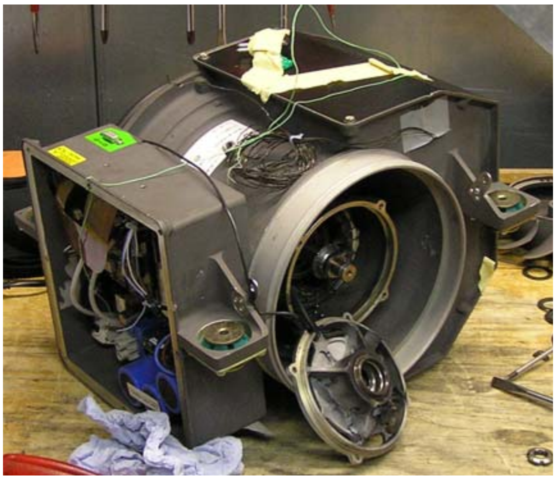


figure 2, back side VA4130

2.1.1 VA4111
VA4111 is an older fan than the others and has a maximum rotation speed limited at 13500 rpm (revelations per minute). The fan has a transformer with an air-cooling element, which generates special frequencies (chapter 5.5.3). In the coming models of Airbus, VA4111 is going to be replaced in with VA4115 and VA4130.

2.1.2 VA4130
VA4130 is smaller than VA4111 and VA4115 and has a weight at only 11.7 kg. The maximum rotation speed is 17500 rpm. There is no air-cooling element in VA4130. The fan is developed for A400M and is still not on the market. The next generation of VA4130, developed for A350, will include a monitoring system based on structured-borne sound.

2.1.3 VA4115
VA4115 is also developed for A400M and will replace VA4111 in the future. Both of the fans have about the same size. The cooling element of the transformer has a different design compared to VA4111. No measurements were done on any VA4115 within the dissertation project.

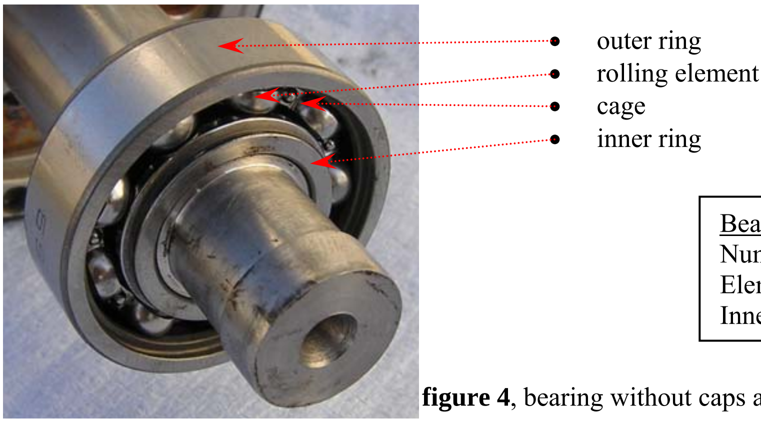
2.2 The bearing

2.2.1 Bearing structure
The bearings used in the fans are simple one row bearings and provided by the Japanese manufacturer NTN. They are delivered sealed and lubricated (figure 3).

The parts of a bearing are:
- outer race
- inner race
- rolling element
- cage (train)
- two caps

The caps are used to protect the bearing from dust and dirt. They are not loaded, and unlike the other parts, a broken cap will not directly affect the function of the bearing. In figure 4 the two caps are removed and the cage and the rolling elements are cleaned from lubrication.

figure 3, NTN bearing



Bearing dimensions
Number of elements (Z): 9 balls
Element diameter (D_E): 4.7 mm
Inner ring raceway diameter (D_I): 18.7 mm

figure 4, bearing without caps and lubrication

The cage is used to keep the elements on equal distance from each other. It is not constructed to bear any larger force and breaks if the forces increase. The rolling elements used in the NTN-bearing are ceramic balls. As result the bearing is not electrically conductive.

If damage occurs on the outer race, inner race, rolling element or cage, vibrations depending on the rotation speed will appear. The frequencies of these vibrations are different for each of the four parts and can be theoretical predicted (chapter 5.3.1).

2.2.2 Lubrication

In a bearing without lubrication, the surfaces of the rolling elements will be direct in contact with the outer and inner races, which will cause rapid bearing breakdown. A protecting layer of lubrication between the surfaces is required. Too much lubrication may also decrease the lifetime of the bearing.

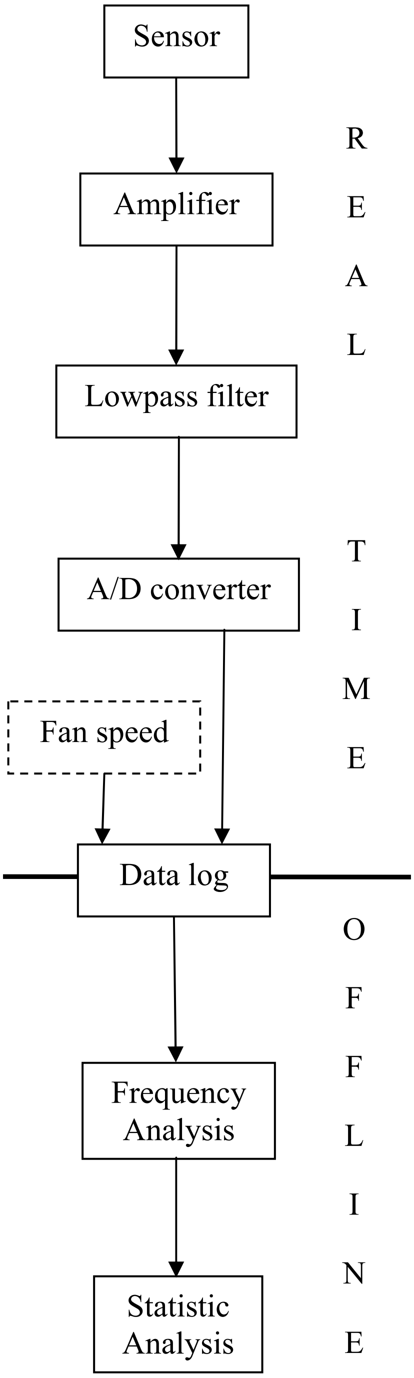
The lubrication has to be chosen very carefully. It must be able to sustain a wide interval of temperatures and air pressures. The fan may be started at -40 ºC on the Russian tundra or at 70 ºC in the Sahara Desert. During operation the lubrication becomes even hotter.

3. Measuring Technique

In this section the measuring technique needed for structure-borne sound measurement is presented. Chapter 3.1 illustrates the measurement chain. Some of the parts in the chain need a deeper explanation. How to choose the sensor is explained in chapter 3.2. In chapter 3.3 the questions of how to place and attach the sensor will be highlighted. Why and how to use a filter will be studied in chapter 3.4.

3.1 Measurement Chain

A measurement chain has to consist of some basic blocks. Some of them have to work in real-time. Directly after the A/D-conversion data can be stored and real time analysis is not required.



3.1.1 Sensor
The first block is the sensor. It receives the structure-borne sound and transforms it into analogue electrical signals.

3.1.2 Amplifier
If the specification of the sensor requires it, an amplifier can be placed right after the sensor.

3.1.3 Lowpass filter
An analogue active lowpass filter can be applied to reduce the effect of aliasing (chapter 3.4.1).

3.1.4 A/D converter
The A/D converter is used to convert analogue data into discrete. The throughput rate of the converter has to be done after some considerations. Chapter 4.2 will discusses this subject further.

3.1.5 Data log
If the data is logged (e.g. in a SRAM) the analysis does not have to work in real time. The rotation speed can be logged continuously to simplify the frequency identification.

3.1.6 Frequency Analysis
A frequency analysis is needed to interpret the signals. The technique of creating a frequency spectrum is explained in section 4 of this dissertation.

3.1.7 Statistic Analysis
To make a conclusion of the bearing condition a statistic analysis is needed. This can be done by monitoring certain frequencies (identified using theory in section 5) or statistical methods of the whole vibration spectrum (section 6).

3.2 Sensor types

The objective of the sensor is to transmute mechanical quantities into electrical signals. The most common sensor used for vibration monitoring is the accelerometer[4]. It measures the forces (F) on a known mass (m) and uses Newton's second law of motion (formula 1) to calculate the acceleration (a). If the velocity or displacement is of interest, those quantities can be calculated out of the acceleration (chapter 5.7).

$$F = m \cdot a \tag{1}$$

The most important property of an accelerometer is its bandwidth. The bandwidth depends on the design of the sensor and manner in which it is attach to the surface.

3.2.1 Piezoelectric sensors

When a piezoelectric material is affected by a force it generates a voltage, which can be measured.

A piezoelectric sensor can measure higher frequencies than any other accelerometer. The typical frequency response from a sensor is to be seen in figure 5[18]. The highest frequency in the frequency range can exceed 50 kHz, but most of the sensors on the market have measures maximum 10 kHz. The amplitude of frequencies slightly over the frequency range will be amplified. Frequencies around the resonant frequency will be seen in the frequency spectrum as well.

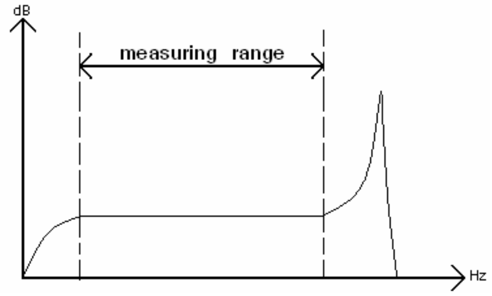


figure 5, typical frequency response piezoelectric sensor

The price of a piezoelectric sensor is $100-600.

3.2.2 MEMS sensors

MEMS (Micro Electro Mechanical System) sensors are small mass spring systems implemented in chips. In the industry they are use e.g. to trigger airbags[19].

figure 6, MEMS-sensor

The frequency range of a MEMS sensor rarely exceeds 3 kHz. The cost pro piece is much lower than that of piezoelectric sensors. The price range spans from a few dollars to $35.

3.3 Capturing signals

3.3.1 Load zone

To be able to position the sensor, knowledge of where the structure-borne sound occur is needed. The bearing can be divided in to three zones dependent upon load on the rolling elements[14]. The zones are pictured in figure 7.

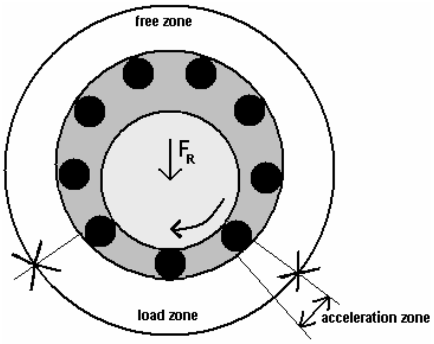


figure 7, free zone, load zone and acceleration zone (schematic picture).

Due to the centripetal force the rolling elements will always be in contact with the outer race during fan rotation. In the free zone no force from the rotating inner race will affect the rolling elements. The speed of the balls will decrease.

When the elements reach the load zone they will be accelerated within a short period. This is the acceleration zone. During this moment of a force increase, the risk of damages occurring or strengthened at the rolling elements or the surfaces in contact with them is significant. From the acceleration zone the bearing structure-borne sound originates.

3.3.2 Sensor position

The metal structure of a machine can be modelled as a mass spring system. If the sensor is placed on the exterior of the machine, the different vibration frequencies will be strained or weaken on their way from the bearing to the sensor. To prevent the phenomenon the sensor has to be placed as close as possible to the bearing and its acceleration zone (figure 8). This will also reduce the structure-borne sound from other sources compared to the sound from the bearing[14].

Due to its source, the structure-borne sound has different directions. All of the bearing tones have at least one component in radial direction[17]. If a 1-axis accelerometer is used it should be placed so as to measure radial waves.

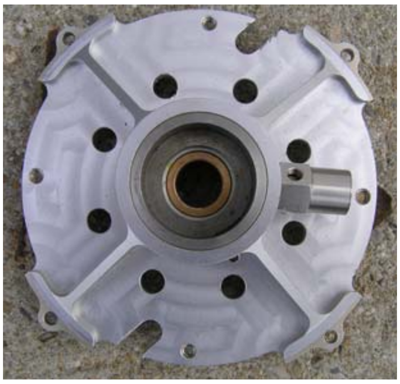


figure 8, placement of the sensor at a bearing sleeve of a VA4111.

3.3.3 Applying sensor

There are different ways of connecting the sensor at the bearing sleeve. But the only stationary solution is to use a threaded pin. A magnet is a poor choice. The quality of the transmitted signals will depend on the roughness of the surface and the shape of the magnet. A better choice is to use two-component adhesive. But the glue has to resist a large range of temperatures and is not a permanent solution.

If a MEMS-sensor (chapter 3.2.2) is preferred a mounting has to be used. In figure 9 a threaded pin mounting developed for the VA4111 is shown. This solution makes it possible to measure vibrations with two sensors at the same time. If 2-axis sensors are used signals will be measured in radial and in either axial or tangential direction, according to how the mounting is turned.

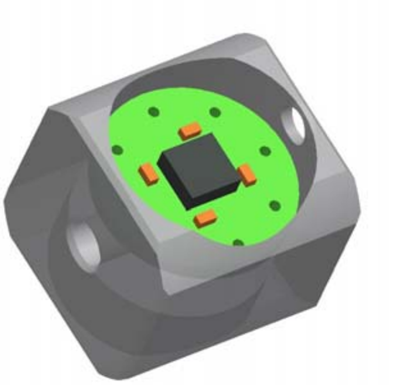


figure 9, mounting for MEMS-sensor

3.4 Filtration

The purpose of a filter is to deposit unwanted frequencies from the measuring signal. There are two main types of filters, analogue and digital. An analogue filter is applied for the A/D-conversion and a discrete afterwards, maybe as code in the fan software. A filter integrated in the software is of course extremely easy to implement, compared to the extra hardware for an analogue filter. The reason why an analogue filter is recommended in the measurement chain for structure-borne sound measurements is the risk of aliasing.

3.4.1 Aliasing

Aliasing is an effect which can appear during the conversion of analogue signals to digital signals. If the sample rate is not sufficient, higher frequencies will be interpreted as lower ones and appear in the spectrum at the wrong places.

If the highest measured frequency is known, the sample frequency can be chosen high enough to measure it. How big the sample rate has to be, is given by the Nyquist-Shannon theorem. According to the theorem, the frequency of the measured wave has to be lower than half the size of the sampling rate[3].

If the highest frequency is unknown or if a lower sample rate is required (chapter 4.2) an active analogue lowpass filter can be used. The filter is applied before the A/D-conversion and will reduce the higher frequencies which cause aliasing.

3.4.2 The Butterworth-Filter

The Butterworth-Filter is a commonly used analogue lowpass filter. The transfer function for Butterworth filters is given by equation 2[19].

$$G^2(f) = \frac{G_0^2}{1+\left(\frac{f}{f_c}\right)^{2n}} \qquad \textbf{(2)}$$

G = The gain of the filter.
G_0 = The gain at the frequency zero.
f_c = cutoff frequency
n = order of the filter.

The cutoff frequency (approximately the same as -3 dB frequency) defines the midpoint in a zone where the frequencies get oppress by the filter. Lower frequencies outside this zone get unchanged und higher get wiped out. How big this zone is, is given by the order of the filter. When the order approaches infinity the zone disappears and the filter gets shaped as a rectangle. If n is equal to infinity the filter is called ideal. In this case all frequencies under the cutoff frequency will pass through the filter and all the ones above will be rejected.

4. Creating a Frequency Spectrum

To identify the sources of structure-borne sound, it is very helpful to know how frequently the vibrations appear. When many different vibrations occur at the same time, only the eye is not enough to identify the different frequencies in the time domain. To solve the problem in a mathematical way, spectrum analysis is required. With spectrum analysis the signal will be shown in the frequency domain. Figure 10 and 11 show the same signal in time and frequency domain.

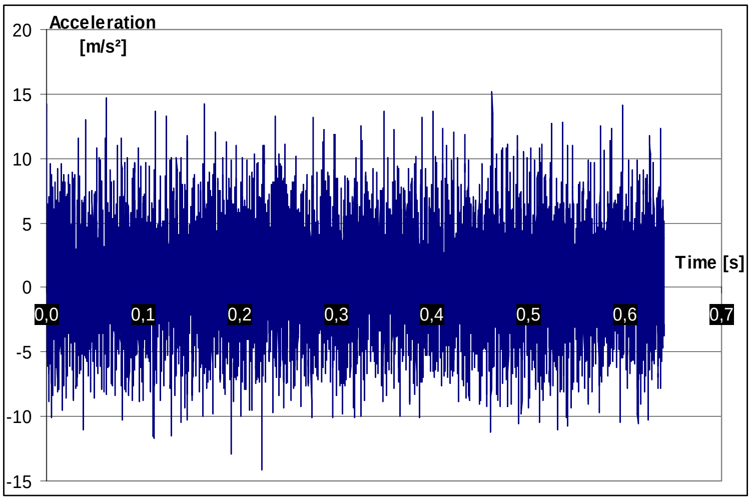


figure 10, measuring data/time domain signal. 0.64 s of 2 s captured 15st of January 11:39 at 16513 rpm.

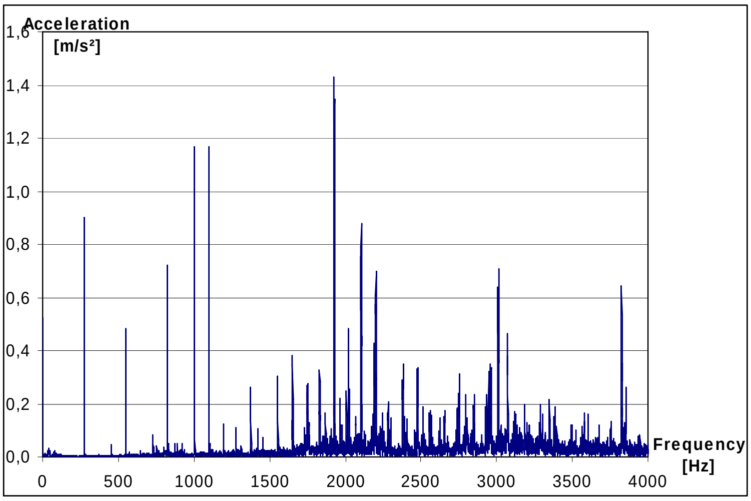


figure 11, frequency spectrum of the 2 s measured data partly shown in figure 10.

Spectrum analysis includes a couple of steps that have to be carefully followed to maintain the true amplitude at the waves. In chapter 4.1 the steps will be introduced and a rough explanation how they are performed. Chapter 4.2 is a discussion about how the sample rate and the amounts of samples during one measurement affect the result and how the parameters can be chosen. The rest of section 4 explains the theory behind the steps in 4.1. The phenomenon of spectral leakage is explained in chapter 4.3. How to prevent spectral leakage with window functions is discussed in 4.4. The theory behind the step of lowpass compensation can be studied in chapter 4.5.

4.1 The technique step by step

4.1.1 Apply Window

First of all a window function has to be applied on the array of samples (x_n). This is done to prevent the spectral leakage phenomena (chapter 4.3). The choice of window function (w_n) can be made in different ways (chapter 4.4). If a Hann-window (formula 14, chapter 4.4.3.2) is chosen formula 3 is used.

$$x_n^w = w_n \cdot x_n = 0.5\left(1-\cos\left(\frac{2\pi n}{N-1}\right)\right)\cdot x_n \tag{3}$$

n = sample number
N = total amount of samples

4.1.2 DFT/FFT

The Digital Fourier Transformation (DFT) is a transformation for transforming a discrete signal from the time domain to a discrete signal in the frequency domain. It is defined as in formula 4[19].

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi j\frac{nk}{N}} \tag{4}$$

k = sample number in frequency domain

X_k turns out to be N-periodic, which makes it unnecessary to calculate X_k for $k<0$ or $k>N-1$.

To evaluate a DFT O(N^2) operations are needed. The Fast Fourier Transformation (FFT) is a faster alternative, which only requires O($N \cdot \log N$) operations. Therefore the algorithm is the first choice in digital signal processing.

There are more versions of FFT, but they are all based on the principle of decomposing the x-array and split it up into smaller and smaller sequences[6,9]. To programme a FFT algorithm right from the start is not easy. At the homepages of gnu scientific library and http://www.fftw.org, examples are to be found. In Scilab the command *fft()* is recommended.

4.1.3 Window function compensation

After transformation the result has to be compensated for the use of window function. Every window function has a scaling factor (table 2, chapter 4.4.4). For the Hann-window the scaling factor is 0.50. The compensation is done by dividing the transformed array with the factor, as with Hann-window compensation in formula 5.

$$X_k = \frac{X_k^w}{sf} = \frac{X_k^w}{0.50} \tag{5}$$

sf = scaling factor from table 2

4.1.4 Amplitude and phase calculation

X_k is a complex array and includes information both about the amplitude (A_k) and the phase (φ_k) of the vibrations[15]. To calculate these quantities formulas 6 and 7 are to be used.

$$A_k = \frac{Mag(X_k)}{N} = \frac{\sqrt{real(X_k)^2 + imag(X_k)^2}}{N} \tag{6}$$

$$\varphi_k = \arg\left(\frac{imag(X_k)}{real(X_k)}\right) = \arctan\left(\frac{imag(X_k)}{real(X_k)}\right) \tag{7}$$

φ_k is the phase relative the start of the time-domain signal. By creating the frequency spectrum, the amplitude is of interest.

4.1.5 Single side spectrum

In measured vibration data the *x*-array in the time domain always is pure real. As a result X_0 (called the DC-element) will turn out as a real number after the transformation. If *N* also is an even number, $X_{N/2}$ will be real as well and is called the Nyquist-element. For real *x*-arrays the relation 8[19] can be observed.

$$X_k = X^*_{N-k} \qquad \textbf{(8)}$$

It means that the second half of the transformed result actually is a reflection of the first half, but complex conjugated. By magnitude calculations the conjugation can be removed. From formula 6 and 8, formula 9 can be derived.

$$A_k = A_{N-k} \qquad \textbf{(9)}$$

The result is that the A_k-array is symmetric around the Nyquist-element, and is called a double side spectrum. The mirror image ($N/2< k <N$) does not provide any extra information and can be departed. At the same time all the peak amplitudes besides the DC-element and the Nyquist-element have to be multiplied by two. The result is called a single side spectrum[15].

Double Side Spectrum		Single Side Spectrum
A_0		A_0
$A_{0<k<N/2}$	⟶	$2\cdot A_{0<k<N/2}$
$A_{N/2}$		$A_{N/2}$
$A_{N/2<k<N}$		0

4.1.6 Sample to frequency

When the single side spectrum is created the sample number may be exchanged with the corresponding frequency. This is done in formula 10.

$$f = k\cdot\frac{f_s}{N} \quad 0 \le k \le \frac{N}{2} \qquad \textbf{(10)}$$

N = total amount of samples
f_s = sample rate

The amplitudes for the frequencies between $f_s/2$ and f_s can not be calculated. This corresponds to the Nyquist-Shannon sampling theorem, which says that a wave period has to be sampled at least twice to be detected.

4.1.7 Lowpass filter reconstruction

If an active lowpass filter is used the spectrum may be compensated for the loss of amplitudes at high frequencies. The theory is further explained in chapter 4.5. If a second order Butterworth filter with 3000 Hz cutoff frequency is used formula 11 provides the answer.

$$F(f) = F_f(f)\cdot\sqrt{1+\left(\frac{f}{3000}\right)^4} \qquad \textbf{(11)}$$

F = reconstructed frequency spectrum
F_f = filtered spectrum

As a result of all the steps a frequency spectrum is provided where the amplitudes corresponds the amplitudes of the vibration waves.

4.2 Choice of sample rate and amounts of samples

The choices of sample rate (f_s) and amount of samples measured at one time (N) have to be done after some considerations.

4.2.1 Preventing aliasing

When choosing the sample rate the risk of aliasing has to be considered. The effect is described in chapter 3.4.1. According to the Nyquist-Shannon theorem the sample rate has to be at least twice as big as the highest frequency of interest.

4.2.2 Sample interval and measuring time

The choices of sample rate and number of samples in the time signal also affect the intervals of frequencies in the frequency domain. With bigger intervals the grade of accuracy of the calculated frequency gets worse. The size of the intervals is given as Δf[15] in formula 12.

$$\Delta f = \frac{f_s}{N} \qquad (12)$$

N = total amount of samples
f_s = sample rate

The size of time window (measuring period) is the inverse of formula 12, and is given as T_w in formula 13.

$$T_w = \frac{N}{f_s} \qquad (13)$$

A large time window has the benefit of reducing disturbances. One disadvantage to be considered is that the frequencies of interest might oscillate during the time. Bearing damage frequencies are linearly depending on the rotating speed of the fan. The speed is controlled and chances regularly to match the set-point (chapter 5.2.2). If the rotation speed is changed during the measure procedure the frequencies generated by the bearing damage will become blurry in the spectrum. Then the software may have problems to recognise the peaks.

4.2.3 Software problems as result of big amounts of samples

By using Scilab for analysing the results, the data has to be stored as ASCII-code. Not until the data is inserted in Scilab, it can be resaved and processed in binary form. To transfer the data from the oscilloscope to a computer in ASCII-form requires a considerable amount of time. The waiting time is proportional to N. By a smaller choice of N the problem can be prevented.

4.3 Spectral leakage

The measured data is naturally a finite sequence (N samples). The FFT algorithm (as well as the DFT algorithm) is depending on the possibility of repeating the sequence an infinite amount of times. This might cause problems in the connections between the sequences[12].

If a wave is measured exactly an integer number of periods (figure 12), the wave will smoothly connect to itself in the next sequence[15]. In this case the frequency of the wave will be observed as a peak in the frequency spectrum (figure 13).

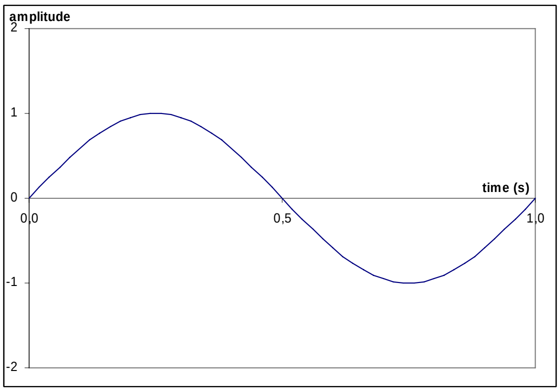


figure 12, the wave length fits an integer (one) amount of times in the measuring period.

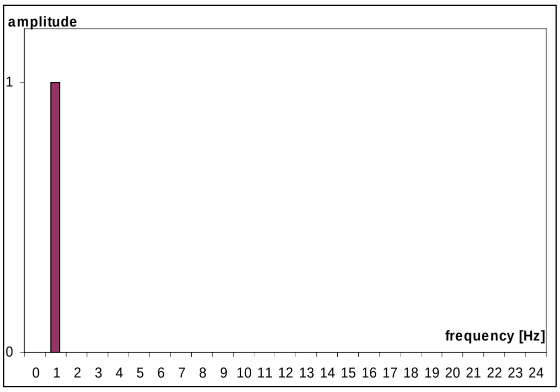


figure 13, the frequency spectrum of the wave in figure 12.

But when a wave appears a non-integer number of times during the measurement (figure 14) there will not be any smooth connection between the sequences. The result is called spectral leakage and can be observed in figure 15.

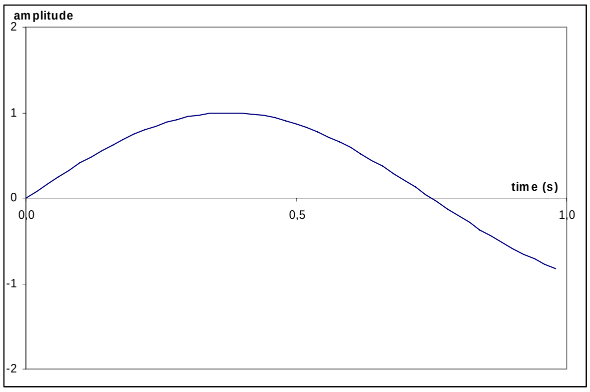


figure 14, the wave length does not fit an integer amount of times in the measuring period.

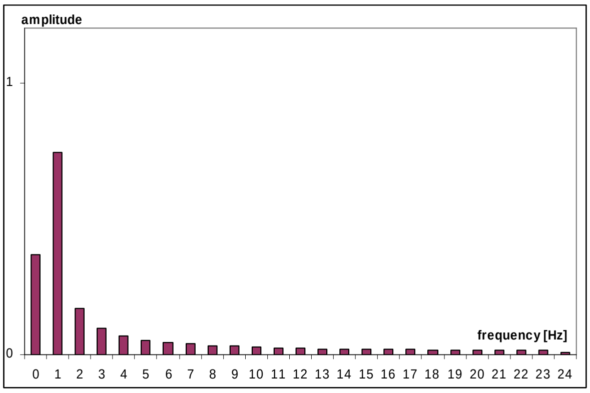


figure 15, the frequency spectrum of the wave in figure 14.

If the measurement includes just one wave, the length of the sequence (time window) can be adjusted to fit an integer amount of periods of the wave. In measured vibration data, where many waves appear at the same time and the frequencies of the waves change constantly, it is impossible to make this adjustment. In this case the only way to counteract spectral leakage is to apply a window function.

4.4 Window functions

Window functions are used to prevent the spectral leakage phenomena (chapter 4.3) when transforming data from time domain to frequency domain. There are many different windows and all of them have their own characteristics. The choice of window function is not easy to make, but there are some rules of thumb to follow.

The first two questions to be asked are:

- What is known about the properties of the time domain data?
- Which will be the most important property of the frequency domain data after the transform, amplitude accuracy or frequency accuracy?

4.4.1 Analysing the frequency spectrum of the window function

To understand the characteristics of the different window functions, their frequency responses are studied. Figure 16 shows the frequency response of a window function. There are three parameters of interest possible to identify in the frequency response. These are the main lope width, the maximum side lope level and the side lobe roll off rate. The values on the x-axis are FFT-bins (frequency lines) and the quantity on the y-axis is gain measured in dB.

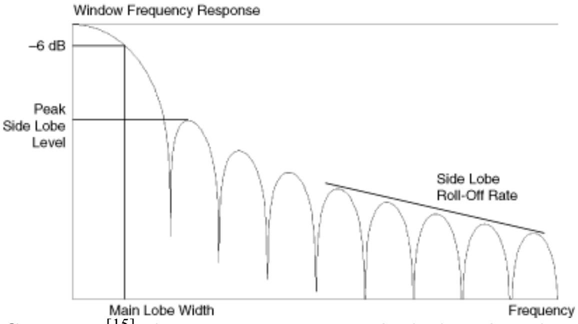


figure 16[15], frequency response of window function

4.4.1.1 Main lope width

The spectral resolution of the window depends on the width of the main lope. A window with a narrow main lope makes it easier to separate two spectral peaks very close to each other. The main lobe width is measured in FFT-bins and labelled with a gain. The -6 dB main lope is the distance from the true frequency until the amplitude is reduced to the half of the real amplitude, multiplied by two.

4.4.1.2 Maximum side lope level

How an interfering signal close to the frequency of interest affects the amplitude of this frequency depends on the side lope characteristics of the chosen window. If the interfering signal is close to the frequency, a window with small maximum side lobe level is a good choice. This means that the amplitude of the interfering signal will be damped quickly.

4.4.1.3 Side lobe roll off rate

The side lobe roll off rate decides how an interfering signal far away from the frequency of interest affects it. A high side lobe roll off rate is good to prevent this affection.

4.4.2 Worst-case amplitude error

A forth parameter to be considered before choosing window function is the worst-case amplitude error. The different windows affect the amplitude unequally. If the accuracy of the amplitude is important this has to be considered.

4.4.3 Some common window functions

There are numerous window functions which can be used to prevent spectral leakage. In table 1 the data of five of the most common window functions are to be found.

Window function	-6 dB main lope width (FFT bins)	Maximum side lobe level (dB)	Side lobe roll off rate (dB/decade)	Worst-case amplitude error (dB)
Uniform (none)	1.21	-13	20	3.92
Hann	2.00	-32	60	1.42
Hamming	1.81	-43	20	1.75
Blackman	2.30	-58	60	1.10
Flat top	3.56	-44	20	<0.01

table 1, characteristics of some common window functions[15]

4.4.3.1 Uniform Window – None window

If no window is applied on the time domain data, it can theoretical be seen as multiplying the values with a rectangular window with height one. This window is also called uniform window.

The frequency response of a uniform window explains why not using windows, in the most cases, increase the spectral leakage. The high maximum side lope level and the low side lope roll off rate of the uniform window make it possible for other signals close and far away from the interesting frequencies to interfere with them. The high worst-case amplitude error is another argument for using nonrectangular window functions.

Despite these facts the uniform window has suitable characteristics for some special types of signals. The main lope width is smaller than by any other window. If the signal includes two very closely spaced sine waves, this characteristic can be used. Another case when the uniform window is suitable is when the signal is random broad banded data (white noise).

4.4.3.2 Hann Window

The Hann window (sometimes called Hanning or raised cosine window) is expressed by formula 14[19].

$$w_n = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \qquad \textbf{(14)}$$

N	= total amount of samples
n	= sample number

The window is the first choice when the properties of the data are unknown[15]. It has a pretty small main lope width, a medium maximum side lope level and a high side lope roll off rate. This makes the Hann window good for identifying frequencies in random signals and suitable for spectral analysis of vibration data. The only catch is the big worst-case amplitude error.

4.4.3.3 Hamming Window

The Hamming window (not to be confused with Hanning window) is expressed by formula 15.

$$w_n = 0.53836 - 0.46164 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \qquad \textbf{(15)}$$

The similarities in the characteristics of the Hamming window and the uniform window are noticeable. The small main lobe width makes them both suitable for the same type of closely spaced sine waves. A benefit by the Hamming window compared to the uniform is the smaller maximum side lope level.

4.4.3.4 Blackman Window

The Blackman window is expressed by formula 16[16].

$$w_n = 0.42 - 0.5 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cdot \cos\left(\frac{4\pi n}{N-1}\right) \qquad \textbf{(16)}$$

The window has the same good side lobe roll off rate as the Hann window. The maximum side lope level is much better, but the main lope width is worse than the Hann window. The worst-case amplitude error is low compared to other windows.

4.4.3.5 Flat top Window

The Flat top window is expressed by formula 17[19].

$$w_n = 1.0 - 1.93 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 1.29 \cdot \cos\left(\frac{4\pi n}{N-1}\right) - 0.388 \cdot \cos\left(\frac{6\pi n}{N-1}\right) + 0.0322 \cdot \cos\left(\frac{8\pi n}{N-1}\right) \quad \textbf{(17)}$$

If a single tone or sine wave is measured and the accuracy of the amplitude is of interest the flat top window is the best choice. It has a worst-case amplitude error which can be approximated to zero. But the wide main lope makes the flat top window impossible to use for spectrum which includes narrow peaks.

4.4.4 Compensation

After the transformation, the spectrum has to be compensated for the use of window function[15]. This is done by dividing the whole transformed array with a scaling factor of the window which was used. If no window is used (uniform window), this compensation off course is unnecessary. The scaling factors are given in table 2.

Window	Scaling Factor
Uniform (none)	1.00
Hann	0.50
Hamming	0.54
Blackman	0.42
Flat Top	0.22

table 2, scaling factors for some common window functions

4.5 Lowpass filter compensation

The purpose of the lowpass filter used in the trials is to eliminate the high frequencies which cause aliasing (chapter 3.4.1). If the filter is of a low order, the cutoff frequency has to be chosen lower, with a bigger safety margin to the Nyquist-Shannon frequency. This might cause a reduction in frequencies in the interesting part of the spectrum.

To compensate this effect the filter reduction can be mathematically modelled and the exposed frequencies can be restored. A requirement for the procedure is of course that the frequencies to be reconstructed are not totally wiped out by the filter.

When a measured signal is filtered in the time domain, this can mathematically be seen as a discrete convolution between the signal and the filter parameters. Convolution in the time domain corresponds to multiplication in the frequency domain. The transfer function of the filter is multiplied by the measured frequency spectrum. The idea of the compensation is to divide the frequency spectrum with the transfer function of the filter to restore the filtering effect.

The transfer function for a Butterworth-Filer was given in chapter 3.4.2 as formula 2. Out of this formula the expression for compensation (formula 18) can be developed.

$$F(f) = F_f(f) \cdot \frac{G_0}{G(f)} = F_f(f) \cdot \sqrt{1 + \left(\frac{f}{f_c}\right)^{2n}} \quad \textbf{(18)}$$

F = reconstructed frequency spectrum
F_f = filtered spectrum

5. Identifying Frequencies

When the frequency spectrum is constructed it is time to identify the sources of the measured vibration. Some frequencies will be found in many places of the spectrum. To understand how, some basic spectral phenomena are studied in chapter 5.1. The rotation speed is identified in chapter 5.2. In chapter 5.3 the bearing frequencies and their behaviour are explored. Chapter 5.4 and 5.5 discuss different identifiable frequencies and their origin. To practise the ability of identifying frequencies an example is given in chapter 5.6. The last chapter shows methods of converting the spectrum consisting of acceleration data to a spectrum with velocity or displacement on the y-axis.

5.1 Spectral phenomena

5.1.1 Harmonics

Harmonics are integer multiples of the fundamental frequencies. They occur commonly in the frequency spectrum. Especially the harmonics of the rotation speed are prominent up to an order of ten. Also the bearing frequencies have harmonics.

5.1.2 Amplitude Modulation

Amplitude Modulation is a phenomenon when two waves amplify each other according to figure 17. The wave with the higher frequency of two waves is called carrier wave (*c(t)*), and the one with lower frequency is called modulation wave (*m(t)*).

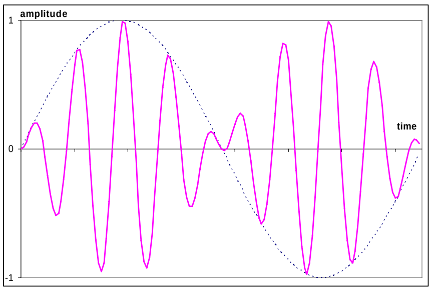


figure 17, amplitude modulation time signal.

The effect was discovered in the 1870s and is used to transmiter AM-radio[19]. The carrier wave has a frequency that remains constant (the frequency of the radio station). By radio transmitting the frequency of the modulation wave is chanced to transfer the sound.

If the modulation frequency is constant as well, amplitude modulation will appear in the frequency spectrum as in figure 18. The carrier frequency is easy to identify. Symmetrical around the carrier frequency two sidebands appear[11]. The difference between the carrier frequency and one sideband is the modulation frequency. The behaviour can be derived with the formula 19[12].

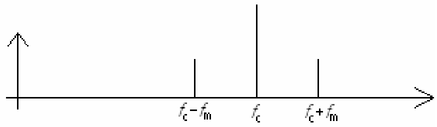


figure 18, amplitude modulation in frequency domain signal

$$y(t) = c(t)\cdot[1+m(t)] = \hat{y}_C \cos(2\pi f_C t + \varphi_C)\cdot\left[1 + \hat{y}_M \cos(2\pi f_M t + \varphi_M)\right]$$

$$= \hat{y}_C \cos(2\pi f_C t + \varphi_C) + \frac{\hat{y}_C\,\hat{y}_M}{2}\left[\cos(2\pi(f_C - f_M)t + (\varphi_C - \varphi_M)) + \cos(2\pi(f_C + f_M)t + (\varphi_C + \varphi_M))\right] \quad \textbf{(19)}$$

c(t)	= carrier wave	f_C	= frequency carrier wave
m(t)	= modulation wave	f_M	= frequency modulation wave
$\hat{y}_C$	= amplitude carrier wave	φ_C	= phase carrier wave
$\hat{y}_M$	= amplitude modulation wave	φ_M	= phase modulation wave

If the modulation frequency is not linear depending on the carrier frequency the position of the sidebands in the spectrum will not be either. This is one of the reasons why sidebands can be hard to identify.

Bearing inner race damages usually cause amplitude modulation[14]. When the damage passes the load zone (figure 7, chapter 3.3.1), the structure-borne sound gets stronger than it is in the free zone. The frequency from the bearing damage is the carrier frequency and the rotational speed the modulation frequency.

5.2 The rotation speed

5.2.1 Finding the rotation speed in the spectrum

The easiest frequency to identify in the spectrum is the fan rotation speed. This is mostly the first large peak. If this peak has a lot of harmonics, it is another cause to suspect it to be. Another way to find the rotation speed is to use the demanded value given in rpm (revelations per minute). To calculate the frequency (in Hertz or revelations per second) expression 20[17] is used. The calculated value will give a hint of where to find the measured rotation speed.

$$f[Hz] = \frac{v[rpm]}{60} \tag{20}$$

To get the rotation speed as exactly as possible, the frequency of the harmonic of seventh or eight order can be identified and divided by its order. In this way the accuracy of the fan speed is bigger than the interval between the frequencies in the spectrum.

5.2.2 The swinging behaviour of the rotation speed

When identifying the peaks the rotation speed has to be known as exactly as possible. In the fans three hall sensors are used to measure the present speed. The result is needed to control the fan speed according to a demanded value on the CAN-bus. The measured speed will always oscillate around the demanded value, and never stay absolutely constant. For a low rotation speed the oscillation is smaller compared to what it is close to the maximum speed. If the fan is situated in a narrow space it will result in high swinging of the rotation speed as well.

Under the advantageous circumstances at the test bench, the rotation speed oscillates approximately ±30 rpm (±0.5 Hz) around the demanded speed. This corresponds to a 10 Hz movement for the peak at 10 times the rotation speed. Under real circumstances the oscillations might increases to ±200 rpm (±3.3 Hz), which corresponds to a movement for the 10 times peak of 67 Hz.

The swinging behaviour of the fan speed does not only make it hard to know the present speed. If the time window is big, the rotation speed may changes within the measurement period. The result in the frequency spectrum is wider or segmented peaks. This effect can not be ignored when peak amplitudes are compared.

When the condition monitoring system is implemented in the fan, it is possible to log the rotation speed during the measuring period and calculated the average or the most common rotation speed.

5.3 Bearing frequencies

5.3.1 Rollover frequencies for bearings

In the structure-borne sound of a damage bearing, special frequencies can be tracked. These frequencies are called rollover frequencies, and can be calculated out of the geometry data of the bearing. There are four different types of possible bearing damages to be identified in the structure-borne sound spectrum:

- race outer race damages
- race inner race damages
- rolling element damages
- cage damages

According to the theory all rollover frequencies are proportional to the rotational frequency of the rotor (f_n), in the standard case when the inner race is rotating and the outer race is fixed. The frequencies can be calculated by the expressions 21-24[14]. In the formulas it is assumed that the rolling elements roll and not slide. In reality the rotation of the bearing elements is a combination of rolling and sliding[7].

Outer race:

$$BPFO = f_n \cdot \frac{Z}{2}\left[1-\frac{D_E \cdot \cos\alpha}{D_I + D_E}\right] \tag{21}$$

Inner race:

$$BPFI = f_n \cdot \frac{Z}{2}\left[1+\frac{D_E \cdot \cos\alpha}{D_I + D_E}\right] \tag{22}$$

f_n	= rotational frequency of the rotor
Z	= number of elements
D_E	= element diameter
D_I	= inner ring raceway diameter
α	= angle of contact
BPFO	= Ball Pass Frequency Other Race
BPFI	= Ball Pass Frequency Inner Race
BSF	= Ball Spin Frequency
FTF	= Fundamental Train Frequency

Rolling element:

$$BSF = f_n \cdot \frac{D_I + D_E}{D_E}\left[1-\left(\frac{D_E \cdot \cos\alpha}{D_I + D_E}\right)^2\right] \tag{23}$$

Cage:

$$FTF = f_n \cdot \frac{1}{2}\left[1-\frac{D_E \cdot \cos\alpha}{D_I + D_E}\right] \tag{24}$$

The angle of contact is defined according to figure 19, and depends on the axial contra the radial force on the bearing. If the angle is unknown, cos(α) can be approximated as 1.

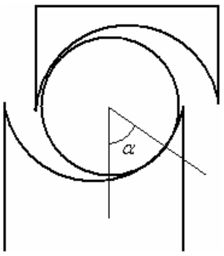


figure 19, angle of contact

5.3.2 Spectral behaviour of the rollover frequencies

The outer race damage frequency is mostly the strongest of the rollover frequencies. The reason is that BPFO does not have to be transplanted through the rolling elements as the other rollover frequencies[4].

Rollover frequencies for outer and inner race damages usually have harmonics. Harmonics from rolling element and cage damages frequencies are rarer[8].

As mentioned in chapter 5.1.2 inner race damages commonly cause amplitude modulation, with the rollover frequency as carrier frequency and the rotation speed as modulation frequency. The sidebands are also identifiable for the harmonics of BPFI. Sometimes the amplitudes of the sidebands are stronger than the rollover frequency itself.

If the outer race is fixed, amplitude modulation of the outer race rollover damage frequency is very abnormal. When sidebands at n times the rotation speed occur at the side of BPFO, the most common explanation is rotating outer race.

The cage rollover frequency is commonly hard to identify. But as modulation frequency with any of the other rollover frequencies as carrier frequency, it commonly appears. Especially the rolling element damage frequency is modulating with the rollover frequency for cage damages.

5.4 Other frequencies dependent on the rotation speed

The bearing frequencies are not the only frequencies linear dependent on the rotation speed. Many physical phenomena can result in frequencies which follows the movement of the rotation speed in the spectrum.

5.4.1 Blade pass frequency

When measuring the structure-borne sound vibrations with the frequency rotation speed times amount of blades on the fan wheel can be captured[13]. This frequency can occur under normal circumstances and does not need to indicate any failure. Sometimes it is also seen as a sideband in the spectrum.

5.4.2 Imbalanced rotor

If the rotor is imbalanced a radial force will be generated given by formula 25[2].

$\omega =$ angular velocity
$r_R =$ radial distance from the axis of rotation to the centre of gravity
$m =$ total mass of rotor
$f_n =$ rotational speed

$$F_R = \omega^2 \cdot r_R \cdot m = 4\pi^2 \cdot f_n^{\,2} \cdot r_R \cdot m \qquad \textbf{(25)}$$

The expression is derived from Newton’s second law (formula 1). According to formula 25 the force will grow quadratically with the value of the rotation speed. Therefore a carefully balanced rotor is especially important for high speed ventilators.

The vibrations generated by the imbalance force will have a frequency one time the rotation speed[1]. A imbalanced rotor is the most common source of vibration at this frequency.

During the lifetime of the fan the imbalance may increase due to dirt or dust, which gets stuck on the blade wheel. A high rotation speed makes the force very sensitive to small changes of mass.

5.4.3 Bent shaft

If the metal coin of the rotor is heavy and the rotor axis is week, the rotor might be bowed[11]. The behaviour is dependent on lifetime and temperature of the rotor axis. In the frequency spectrum it is observed as frequencies of mainly one times the rotation frequency and secondary as two times the rotation frequency. The vibrations can be measure radial and axial.

5.4.4 Misaligned (cocked) bearing

If there is a misalignment between the bearing and the rotor axis (figure 20), this will also cause radial and axial vibrations with frequencies one or two times the rotation speed[17].

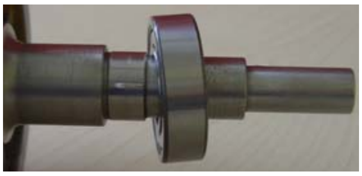


figure 20, misaligned bearing (arranged photo)

5.4.5 Looseness

Looseness causes vibrations with frequencies of n times the rotation speed (where n is an integer from one to ten). The reason is either external looseness when the fan and the foundation are not properly connected or internal looseness when the parts inside of the fan are loose. Unlike the vibrations from an imbalanced rotor the vibrations with looseness as source has tangential components. Extreme looseness can cause vibrations with frequencies at 0.5, 1.5, 2.5 etc times the rotation speed[17].

5.4.6 Rubbing rotor

If the rotor rubs against the stator, severe poundings[1] will not only be measurable, but also hearable for the ear at low rotation speeds. The frequencies of the radial structure-borne sound are fractions of the rotation speed. Due to sidebands and slipping frequency behaviour the spectrum will get very blurry. Rubbing rotor is a serious failure and can cause a burned stench. The fan should be shut down as soon as possible.

5.5 Frequencies independent from the rotation speed

When the rotation speed fluctuates some measured frequencies stay unchanged. These frequencies may originate from sources outside the fan and are hard to identify.

5.5.1 Line frequencies

The fans are provided by a line frequency at rough 400 Hz. This frequency may occur in the spectrum, especially for high fan rotation speeds. Also the line frequency times two can appear due to the symmetric requirement on north and south poles. The number of pairs of poles may be multiplied with the line frequency as well[13]. For the fans the number of pairs of poles is three.

Commonly a fixed frequency at 50 Hz appears in the spectrums during measuring trials. The phenomenon can be explained by the equipment used for the measurement. The amplifier is provided with the line frequency 50 Hz. When the signals from the transducer are strong the sensitivity of the amplifier has to be reduced to 0.001 Volt/g (chapter 8.1.2). This will result in a stronger frequency at 50 Hz. For the same reasons 150 Hz and 250 Hz may appear.

The line frequencies of the fan and the amplifier appear in the same area of the frequency spectrum as the rollover frequencies. This might lead to confusion or in worst case wrong conclusions about the bearing condition.

5.5.2 Switching frequency

The discrete current from the controller to the engine should as much as possible have the shape of a sine curve. This is of course impossible and the current will have steps with a certain space between. The size of the steps is given of the switching frequency, which is identifiable in the spectrum. The switching frequency is different for VA4130 and VA4111, but in both cases much higher than the bearing frequencies and the harmonics of them.

If the measurement is not too noisy, sidebands will appear on each side of the switching frequency situated four times the rotation frequency away. The switching frequency is the carrier frequency and four times the rotation speed is the modulation frequency. This phenomenon may be used to calculate the rotation speed, if it is unknown.

5.5.3 The cooling element of the transformer

A difference between the big VA4111 fan and the small VA4130 fan is that the VA4111 fan has an air-cooling element to keep the transformer cold. There are two different types of air-cooling elements used in the fan. The older model has nine ribs and is called K9. The improved version only has seven ribs and labelled as K7. In the airflow from the fan, the ribs start to swing. The ribs of K7 have a conical sectional view, instead of rectangular by K9. The improved design counteracts the swinging behaviour[10].

For fans with K9 the tones generated by the rips are very strong and also hearable. The ribs on K9 can be compared to tines on a tuning fork. The resonant frequency of a tuning fork (f_T) can be calculated with formula 26[5].

$$f_T = \frac{\pi}{16\sqrt{3}} \cdot \frac{w}{L^2} \cdot \sqrt{\frac{E}{\rho}} \cdot c_n \qquad \textbf{(26)}$$

w	=	width of the fork tine	n	=	mode number
L	=	length of the fork tine	c_1	=	1.194^2
E	=	Young’s modulus	c_2	=	2.988^2
ρ	=	density of the material	c_n	=	$(2n-1)^2$, $n > 2$

This formula will off course not give any exact air-cooling element frequencies, but it gives an idea of where to find them.

The VA4111 with K9 has a special group of tones at 4100-4400 Hz, which is very strong for certain rotation speeds. From approximately 9600-12800 rpm the tones, generated when the ribs swing in the air flow, are totally dominant. The energy within the vibrations is huge. A spectrum measured at 11600 rpm does only consist of these tones from the cooling element (figure 21). For rotation speed lower than 9600 rpm or higher than 12800 the tones are not to be found.

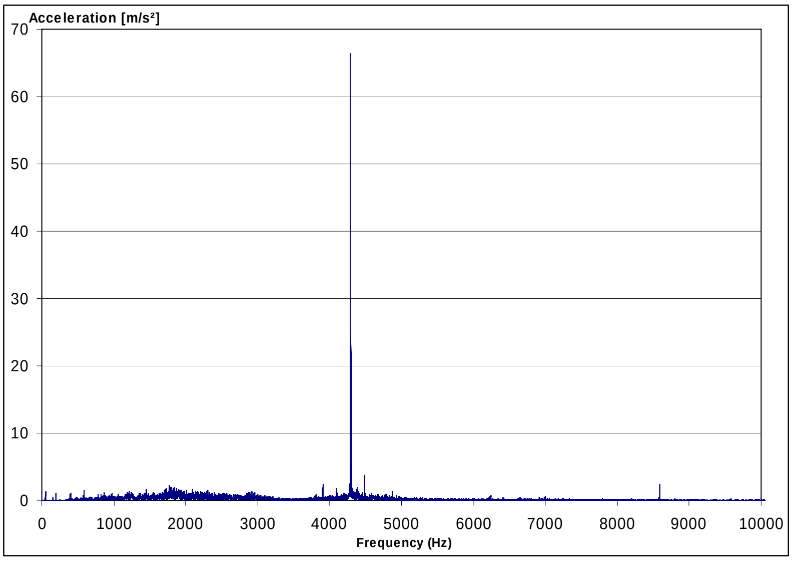


figure 21, the dominance of the cooling element tones. Measured at 11600 rpm on a VA4111, 5 December 2007.

For rotation speeds at 9000-10000 rpm tones in the area of 3750-3800 Hz occur. The amplitudes of the tones are not comparable to ones for the tones in the area of 4100-4400 Hz, but their present gives a clue of the distance between the groups of tones.

Sometimes harmonics and sidebands are generated of the tones from the cooling element. As modulation frequency the rotation speed or the blade pass frequency can be observed.

5.6 Example: Identifying frequencies

Figure 22 is the frequency spectrum of measurement data sampled 21st of January 6:28, with the commanded rotation speed of 10000 rpm. The sample frequency was 50 kS/s and the amount of samples was 100000 during the measurement. According to formula 12 and 13 the interval between the frequencies is 0.5 Hz and the measuring period is 2 s. In chapter 8.2.7 more details about the trial are found.

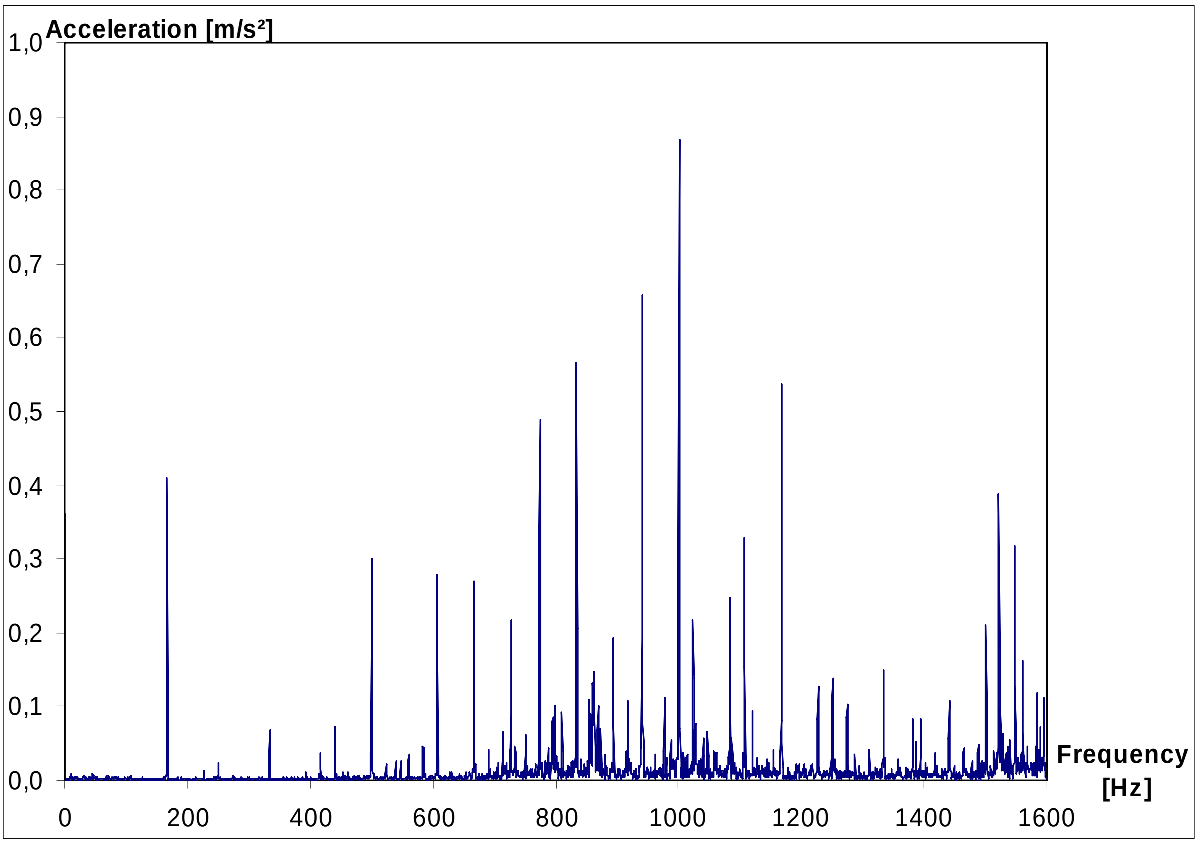


figure 22, frequency spectrum from structured-borne sound measured 21st of January 6:28

5.6.1 Rotation speed identification

With formula 20 the theoretical rotation speed is calculated to 166.67 Hz. In the spectrum (figure 22) a peak at 0.41 m/s^2 for the frequency 167 Hz is found. The value for the peak at 166.5 Hz is higher than the value for the peak at 167.5 Hz. This indicates that the true rotation frequency is lower than 167 Hz, but higher than 166.75 Hz (otherwise the value for 166.5 Hz would be higher than the value for 167 Hz).

To get a more exactly solution the harmonics of the rotational speed are identified. The peak six times the rotation speed should be close to 1000 Hz. One peak is found at 1000.5 Hz. This corresponds to a rotation speed at 166.75 Hz. For seven times the rotation speed one peak is found at 1167.5 Hz, which communicates with a fan speed of 166.79 Hz. At eight times the rotation speed the value are almost equal for the peak at 1334 Hz and 1334.5 Hz. If the peak frequency is assumed to be 1334.25 Hz, this corresponds to a rotation speed at 166.78 Hz. This value has a high accuracy and matches the values calculated before. From now on 166.78 Hz (10007 rpm) is the calculated the rotation speed.

It may seem unnecessary to do all this calculations for such a small difference as 7 rpm, and it probably is. But with higher fan speeds and under less favourable circumstances the difference will be bigger.

5.6.2 Normalisation of frequency spectrum

The next steep is to normalise the frequencies with the calculated rotation speed. This is done by division of the frequencies with the rotation speed. The result is pictured in figure 23.

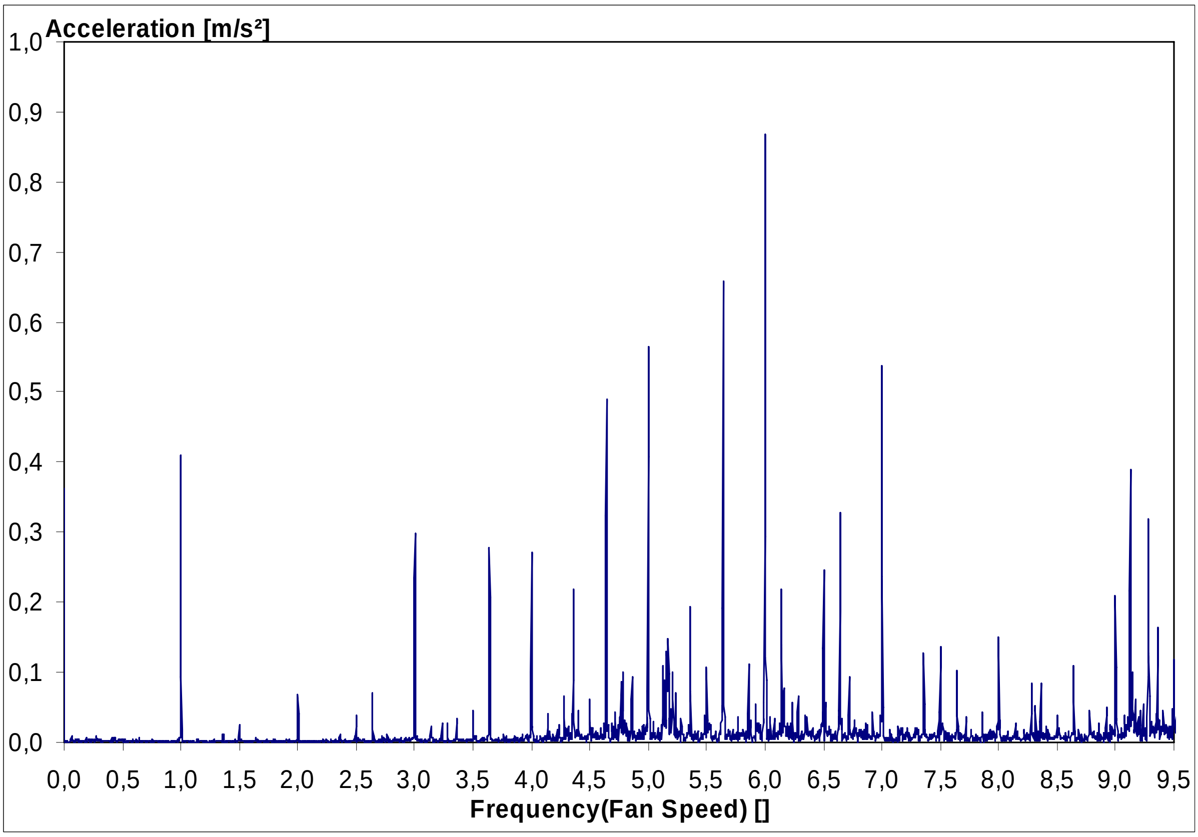


figure 23, frequency spectrum in figure 22 normalised with the calculated fan speed

On the x-axis the factors to multiply with the rotation speed to get the frequencies now is to find. After the scaling the harmonics are easy to see. 1.5, 2.5, 3.5 and so on can be identified in figure 23 as well. These frequencies indicate looseness (chapter 5.4.5).

5.6.3 Identification of outer race damage frequency

The easiest of the rollover frequencies to identify is BPFO (outer race damage frequency). If the dimensions of the bearing are known (given in chapter 2.2.1) the BPFO factor can be approximately calculated with formula 21 and α=0°. This calculation gives BPFO to 3.60 times the rotation speed. In the spectrum (figure 23) a peak at 3.64 can be found. The value corresponds with α=19°, if formula 21 is used again.

Peaks at n times the rotation speed plus 0.64 are common in the spectrum. They origin is amplitude modulation between the outer race damage frequency (carrier frequency) and the rotation speed (modulation frequency). The explanation is rotation of the outer race, which presumably is not fixed.

The first harmonic of BPFO at 7.28 is hard to identify but the sidebands at 3.28 and 4.28 can be seen in figure 23.

5.6.4 Identification of other rollover frequencies

With α known the other rollover frequencies can be calculated too and compared to the spectrum. The factor for inner race damage is given by formula 22 to 5.36. This value agrees exactly with a larger peak in figure 23. Sidebands can be identified at 1.36, 3.36, 4.36, 6.36, 8.36 and 9.36. For the first harmonic 10.72, sidebands are observed as well, e.g. at 4.72 and 6.72.

The rolling element damage frequency is calculated according formula 23 to 4.80 times the rotation speed. No peak was found at this place, which is not any surprise. Rolling element damages are used to be rare.

With α=19º formula 24 gives the cage damage rollover frequency as 0.40 times the rotation speed. This peak is very small and can not be seen with the zoom chosen in figure 23. The peak at 3.24 is a bit bigger. This peak is an amplitude modulation of the outer race damage frequency (carrier frequency) and the cage damage frequency (modulation frequency).

5.6.5 Other frequencies to be identified

Any switching frequency will not be observed in spectra with 1600 Hz as maximum frequency and do not have to be considered in this example. If any of the line frequencies is strong enough they would be identifiable in the spectrum, but none of them is.

The origin of frequencies at 5.12 to 5.24 times the rotation speed is not known. Also the sidebands at n+0.86 and n+0.14 are hard to identify.

5.7 Other physical quantities

The most common sensor to measure structure-borne sound is the accelerometer. For some results velocity (*v(t)*) or displacement (*s(t)*) might be of interest. According to the basic physic formula 27 and 28 are given.

$$v(t) = \int a(t)dt \qquad \textbf{(27)}$$

$$s(t) = \int v(t)dt = \iint a(t)dt^2 \qquad \textbf{(28)}$$

In signal processing there are two different ways of integrate, dividing the frequency spectrum or using running sum.

5.7.1 Dividing the frequency spectrum

If the accelerations spectrum is given the velocity spectrum can be calculated with formula 29[12].

$$V(f) = \frac{A(f)}{2\pi f} \quad f \neq 0 \qquad \textbf{(29)}$$

A(0) in the frequency domain is the offset of *a(n)* in the time domain. The magnitude of the offset is depending on the settings of the measuring equipment and not interesting for the result. Therefore *A(0)* can be set to zero.

The result of formula 29 is that vibrations at low frequencies are more responsible for the velocity of the sensor.

5.7.2 Running sum

Another way to get the *v(n)* out of *a(n)* direct in the time domain is to use running sum. Running sum (formula 30) is the digital equivalence to integration[16].

$$v(n) = \sum_{i=0}^{n} a(i) \qquad \textbf{(30)}$$

Before the calculation is made the *a(n)* has to be high pass filtered[15], otherwise the result will look like a ramp. It is not enough just to minimize the offset, the small frequencies (up to a couple of Hz) have to be excluded as well. A high pass filtration can be made as a convolution of a digital filter and *a(n)*. A digital filter can never be ideal and frequencies in the lower part of the interesting area will be affected. Another way of filtering the signal is to transform it into the frequency domain, set the amplitude of the disturbing frequencies to zero and transform it back to time domain. Also this method is not ideal, because of the spectral leakage phenomenon (chapter 4.3). If the true amplitude is calculated (formula 6, chapter 4.1.4) in frequency domain, information will get lost that is needed for the back transformation. A good advice is to set the frequencies and the mirror frequencies to zero directly after the step of FFT. After this, the back transformation can be done.

The formula 30 is an O(n^2) operation. With the recursive alternative in expression 31[16] an O(n) operation is reachable.

```
$v(0) = a(0);$
$for \quad i = 1 : (N-1)$
    $v(i) = v(i-1) + a(i);$                (31)
$end$
```

In Scilab the function *cumsum()* is the fastest possibility.

6. Statistic Analysis

Analysis of the measured structure-borne sound can be made in two different ways:

- Analysing amplitudes for special important frequencies
- Using broadband diagnostic methods to interpret the characteristic of the whole spectrum

Techniques to find the special frequencies to monitor were handled in section 5. In chapter 6.1 the broadband diagnostic methods will be introduced. Chapter 6.2 discusses how these methods and monitoring methods for special frequencies can be trended. The last chapter in section 6 will highlight advantages of combing broadband diagnostic methods with methods building on special frequencies.

6.1 Broadband diagnostic methods

Not every type of spectral analysis requires exact knowledge of the critical frequencies. The broadband diagnose methods are based on the fact that a frequency spectrum measured from a damaged bearing has higher peaks compared to the noise floor (background level), than an undamaged bearing. There are many statistical quantities used to characterise the spectrum[14].

6.1.1 Maximum absolute value

The maximum absolute value (or peak value) is calculated with formula 32.

$$y_{mav} = \max\left(\left|y_0 \ldots y_{N-1}\right|\right) \qquad \textbf{(32)}$$

In a frequency spectrum all values are real and positive. The maximum absolute value corresponds to the value of the highest peak. To make a conclusion of the bearing condition, only by looking at the peak value might be dangerous. Frequencies independent from the bearing conditions might have higher amplitudes. E.g. this is the case with the frequencies from the VA4111 cooling element at rotation speeds around 11000 rpm.

6.1.2 Root-mean-square value

Root-mean-square value (rms value) is the energy content of the signal. It is computed with formula 33[19] and can be used as a mean value for an array with measure values in time or in frequency domain.

$$y_{rms} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} y_i^2} \qquad \textbf{(33)}$$

Just observing the rms value of a broad banded signal is also not sufficient to diagnose a bearing damage. When only a few of the peaks increase, the root-mean-square value will not be affected.

6.1.3 Crest factor

The crest factor is defined as the ratio between the maximum absolute value (formula 32) and the root-mean-square value (formula 33) and given by formula 34.

$$y_{crest} = \frac{y_{mav}}{y_{rms}} \qquad \textbf{(34)}$$

Due to the composition of the crest factor, it suffers from the same imperfections as its elements. If it is clear that the waves with the highest amplitudes originate from the bearings, the crest factor can be used as a diagnostic measurement of the bearing condition.

6.1.4 Kurtosis factor

The kurtosis factor does not just compare the highest peak with the noise floor, as the crest factor. It is a measurement of the "peakness" in the entire array of values. In words it can be explained as the ratio between the fourth central statistical moment and the square of the second central statistical moment of the amplitude distribution[19]. Formula 35 is the mathematical expression for the kurtosis factor.

$$y_{kurtosis} = \frac{m_4}{m_2^2} - 3 = \frac{N \cdot \sum_{i=0}^{N-1} \left(y_i - y_{mean}\right)^4}{\left(\sum_{i=0}^{N-1} \left(y_i - y_{mean}\right)^2\right)^2} - 3 \qquad \textbf{(35)}$$

The subtraction of 3 in the end of the formula is a corrugation to make the kurtosis factor zero for the normal distribution[19]. Sometimes it is excluded in descriptions of the kurtosis factor. A positive kurtosis factor means a larger share of peakness than in the normal distribution.

6.2 Trend analysis

To know the critical values for the different broadband diagnostic methods is hard. The root mean square value is depending on the background noise, which might be extensive in an aeroplane. Instead of looking directly at the values of the broadband methods, the trend of the values can be of interest for condition monitoring.

6.2.1 K(t) value

The *K(t)*-method is based on the principle of observing the maximum absolute value (formula 32) and the root-mean-square value (formula 33), and is given by the expression 36[14].

$$K(t) = \frac{y_{rms}(0) \cdot y_{mav}(0)}{y_{rms}(t) \cdot y_{mav}(t)} \qquad \textbf{(36)}$$

$y_{rms}(0)$ and $y_{mav}(0)$ are reference values and can be averages of a numerous amount of measurements done initially. *K(0)* is according to the definition 1 and as time increases, *K(t)* decreases. When plotting the value a logarithmic scale on the *K(t)*-axis is the most proper choice.

6.2.2 Trends for special frequencies

Not only the broadband diagnostic methods are suitable to trend. The height of special frequency peaks (see section 5) can be observed and used to diagnose the condition of the bearing. Also this has to be made carefully. When the rotation speed changes, any of the monitored frequencies might approaches a frequency independent from the rotation speed (chapter 5.5). The summation of the frequencies may exceed an alarm level. To handle the problem only mean values of many measurements at different rotation speeds may be compared to alarm levels.

6.3 Combining methods for special frequencies and broadband methods

There are risks with methods based only on peak values of special frequencies. If the rotation speed fluctuates a lot within the time window of the measurement, the peaks get wider and not so high. To determine the exactly rollover frequencies might be hard as well.

The disadvantage with the broadband diagnostic methods is that they are working blind over the frequency spectrum. Frequencies with unknown origin are weighed in as much as the roll-over frequencies.

It is possible to exploit the advantages of both the groups of methods. A digital filter can be applied in the frequency domain around the frequencies of interest. This filter should consist of small windows which only let frequencies of interest and their closest neighbourhood through. These windows should be wide enough to surely include the whole width of the peaks, but exclude most of the noise between. The amplitudes of frequencies outside the windows can be set to zero. E.g. peaks situated n times the rotation speed might be excluded.

The wideness of the windows in the filter can be adjusted automatically after the fluctuation of the rotational speed. Even the shape of the windows can be changed. The BPFO-frequency, its harmonies and its sidebands can be suppressed to compensate their magnitude compared to the other rollover frequencies.

After the filter is applied a broadband diagnostic method can be used to analyse the filtered spectrum. Trends can off course be monitored as well.

7. Software

Section 7 is an explanation to the software developed within this dissertation. The software itself is to be found in Appendix A and is written in Scilab-code. To understand the code chapter 7.1 offers a brief introduction to Scilab (version 4.1.1) and chapter 7.2 explains some commands which were used. In chapter 7.3 explanations to the code-files is to be found.

7.1 Introduction to Scilab

This chapter will give a short introduction to Scilab, which hopefully makes it easier to understand the code. Scilab is an open source alternative to Matlab. For readers who earlier worked with Matlab this chapter will not bring anything new.

7.1.1 Matrixes and arrays

The main advantage with Scilab (and Matlab) compared to other programme languages is the simplicity when working with arrays and matrixes. To exemplify matrix *A* and *B* are created:

```
-->A=[1 2; 3 4]
 A  =

   1.    2.
   3.    4.

-->B=[2 3; 4 1]
 B  =

   2.    3.
   4.    1.
```

The two matrixes can easily be added or subtracted from each other:

```
-->C=A+B
 C  =

   3.    5.
   7.    5.

-->C=A-B
 C  =

 - 1.  - 1.
 - 1.    3.
```

There are two versions of multiplication. Normal matrixes multiplication is done by:

```
-->C=A*B
 C  =

   10.    5.
   22.    13.
```

The other version of multiplication is done separately for each element. The element *2,2* in matrix *A* is multiplied with element *2,2* in matrix *B* and the answer will be in element *2,2* by matrix *C*, independent from what the other elements are. This multiplication is done by adding a point before the multiplication sign.

```
-->C=A.*B
 C  =

   2.     6.
   12.    4.
```

By division between two matrixes the first matrix is multiplied with the inverse of the second.

```
-->C=A/B
 C  =

   0.7  - 0.1
   1.3    0.1
```

The point before the division sign gives a separate divisions for each element.

```
-->C=A./B
 C  =

   0.5     0.6666667
   0.75    4.
```

If a special element of a matrix is of interest, parentheses are added to the matrix. The first value in the parenthesis is the row and the second is the column.

```
-->A(1,2)
 ans  =

   2.
```

An array can be created as follows:

```
-->arr=1:0.5:4
 arr  =

   1.    1.5    2.    2.5    3.    3.5    4.
```

The first value is the start value. The second value is the step size between the values in the array. The third value is the end value. If the second value is excluded it is assumed to be one.

An array or a matrix can be transposed if it is followed by one ‘-sign.

```
-->A'
 ans  =

   1.    3.
   2.    4.
```

7.1.2 Symbols

When a semicolon is added in the end of the expression the answer will not be written in the prompt.

```
-->C=A+B;
```

If the // is written the rest of the line is a comment and will not be processed. In the following example *C* is calculated as *A*+*B* and not as *A***B*.

```
-->C=A+B; //C=A*B

-->C
 C  =

   3.    5.
   7.    5.
```

A string of text is surrounded by two ‘-signs.

```
-->a='string';
```

Constants are started with a % sign.

```
-->%pi
```

```
 %pi  =

   3.1415927
```

7.1.3 If, for and while

An *If*-statement is written as follows in Scilab:

```
if a == 0 then
        …….
elseif a > 0 then
        …….
else
        …….
end,
```

A *For*-loop in Scilab is written as:

```
for b = 1:3,
        …….
end
```

Notice that *b* in example is a scalar value. The loop will run three times and *b* will change in between. The comma is easy to forget.

A *While*-loop implemented as follows:

```
while c<10,
        …….
        c=c+1;
end
```

7.1.4 Help

Scilab has a help function to get informed of any command.

```
-->help if
```

With this command a help window will open and information about the *If*-statement is to read. Some commands are different in Scilab from Matlab. It is possible to use the help function on Matlab-commands to get translations to Scilab-commands.

7.2 Useful Scilab-commands

abs(a)	Delivers the absolute value of *a*.
clear;	Clear all variables.
clf	Opens and cleans a diagram window.
w=cumsum(v)	Sums an array as follows: *w1=v1, w2=v1+v2, w3=v1+v2+v3….*
endfunction	Ends a function.
fft(v)	Returns the FFT of the array *v*.
function o=name(i)	Initialises and starts a function. *o* is output, *i* is input and name is the name of the function.
imag(a)	Deliver the imaginary part of *a*.
legend('text',[a],%f)	Adds a small box in the corner number *a* of the diagram window, with an explanation to the curve. %f means box without frames.

load('filename.dat','A')	Loads matrix *A* from dat-file.
max(a,b)	Return the maximum value of *a* and *b*.
mean(v)	Returns the mean value of the array *v*.
ones(a,b)	Creates a matrix of size *a* rows and *b* columns all filled with ones.
plot(v,w)	Plot an array *w* as function of an array *v*.
read('filename.txt',a,b)	Reads a matrix of *a* times *b* values from a txt-file.
real(a)	Deliver the real part of *a*.
round(a)	Rounds up *a* to the closest integer.
save('filename.dat',A)	Saves a matrix *A* in a dat-file.
size(A)	Returns the size of the matrix *A*.
sqrt(a)	Provides the square root of *a*.
stacksize(a)	Sets the stacksize (app. memory space) to *a*.
subplot(a,b,c)	Splits a diagram window up in *a* times *b* pieces. *c* tells which the current window is.
sum(v)	Sums all values in the array *v* up.
write('filename.txt',A)	Saves the matrix *A* in a txt-file.
xlabel('text')	Labels the x-axis.
ylabel('text')	Labels the y-axis.
zeros(a,b)	Creates a matrix sized *a* times *b* with zeros.

7.3 Comments to software code

7.3.1 *frekAnalyser.sce* (Appendix A.1)
The program is developed to generate frequency spectra according to the theory in section 4. Besides a txt-file with the measurement, the user has to provide the program with some basic information. The sample frequency (*sampFrek*) and the number of samples (*N*) are compulsory inputs. *winMin* and *winMax* are variables to decide the frequency range at the plot window. *ampfac* is an amplification factor given at the amplifier (e.g. 0.1, 0.01 or 0.001). A timestamp can be given to simplify the comparison with outer measurement results. If the rotation speed is not known it can be guessed, the first time the program is running. When the frequency spectrum is generated the rotation speed can be calculated (chapter 5.6.1). *frekAnalyser.sce* has to run all through with the calculated fan speed at least one time.

7.3.2 *getTops.sce* (Appendix A.2)
getTops.sce is a software to generate a list with frequencies and amplitudes for the highest tops in the frequency spectrum within a window defined of *winMin* and *winMax*. As input the program needs the output from *frekAnalyser.sce*, as well as *sampFrek*, *N*, *winMin*, *winMax* and *noOfTop*. *noOfTop* is a variable which tells the program how many tops to be written in the list. E.g. *noOfTop*=10 generates a list with the 10 highest tops. In the first step all tops are identified within the given range. In the second step they get compared and stored in a special list. The list is written in the prompt.

7.3.3 *makeMatrix.sce* (Appendix A.3)

The program gets many frequency spectra from *frekAnalyser.sce* and assemblies them in the same matrix. All the spectra should belong to the same measurement run. This is done to simplify the comparison between the measurements, done by outer programs.

7.3.4 *sum.sce* (Appendix A.4)

sum.sce is a program which uses the output from *makeMatrix.sce* to find frequencies independent of the rotation speed (chapter 5.5). All measurements in a run are added together without consideration of fan speed. If a small top of the same frequency is present in many spectra they will occur clearly in the summed spectrum. Needed input data is *sampFrek*, *N*, *winMin* and *winMax*.

7.3.5 *sumNorm.sce* (Appendix A.5)

sumNorm.sce is based on the same idea as *sum.sce*. The difference between the programs is the scaling done in the *sumNorm.sce* due to the rotation speed. In the generated spectrum the frequencies independent of the rotation speed will be suppressed and the dominant scaling factors (frequency/rotation speed) will be easy to identify.

7.3.6 *trackTop.sce* (Appendix A.6)

The purpose of the program is to explore how tops develop when the rotation speed changes. *trackTop.sce* can track frequency tops or scaled tops. If frequency tops are of interest *makeMatrix.sce* and the matrix *amp* should be used as input data. When tracking scaled tops *sumNorm.sce* and matrix *ampNor* are the input data.

7.3.7 *intVelDis.sce* (Appendix A.7)

The program is developed to give the velocity and displacement measurement out of the acceleration data. The theory is explained in chapter 5.7.2. The method of lowpass filtration with transformations is used. The phenomenon of spectral leakage causes uncertainty in the accuracy of the results. The inputs needed are the measurement data, *sampFrek*, *N* and *ampfac* (amplification factor).

7.3.8 *statAnalyser.sce* (Appendix A.8)

statAnalyser.sce compares measurement runs. The data used for the figures 30 and 31 in chapter 8.2.7 were processed by the program. Both broadband diagnostic methods and combination of broadband methods and methods of special frequencies are implemented.

To simplify the reading of the program it is fragmented into two functions and one main program. The main program calls the second function and the second calls the first.

The first function includes data of the rollover frequency factors. It is only an example. The user can change which frequencies according to his interest. A weight factor for every frequency factor is attached as well. The second function gets its data from the first and composes one filter out of it. This filter has to have the same length of as the input arrays in the main program. The output from *sumNorm.sce* is used as input. The main program also needs the *topbreNorm*, *maxNorm* and *numOfMesRuns* as inputs. *topbreNorm* tells how big the windows around every frequency factor has to be. *maxNorm* tells the highest maximum norm to be processed. *nuOfMesRuns* tells how many measurement runs to be used in the calculation.

The maximum amplitude value, the root-mean-square value, the crest factor and the kurtosis factor are calculated with and without the use of the filter for every measurement run.

8. Practical Experiences

The purposes of section 8 are to show how the tests were accomplished and to give some tips for the readers who would like to measure structure-borne sound themselves. Chapter 8.1 describes the parts at the test bench, which was used. In chapter 8.2 the trials, on which this report is based, are evaluated.

8.1 The test bench

8.1.1 Sensor

The sensor used for the structure-borne sound measurements was a piezoelectric accelerometer (figure 24), produced by Brüel & Kjær. The frequency range contains frequencies up to 10 kHz and the natural frequency of the sensor is 45 kHz.

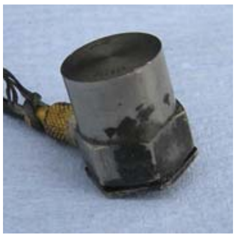


figure 24, piezoelectric sensor

8.1.2 Amplifier and filter

The sensor was connected to *input*, at the amplifier (figure 25). The amplifier is also produced by Brüel & Kjær and was calibrated before use. When the diodes indicated overload the rotating wheel in the middle, labelled *Volt/g Out*, was turned anticlockwise. The software was informed of this modification.

The rotating wheel down to the right was used to apply an active analogue lowpass filter on the signal. The lowpass filter is a Butterworth filter of second order. The choice of -3 dB frequency can be made between 1, 3, 10 and 30 kHz. *Lin.* removes the filter.

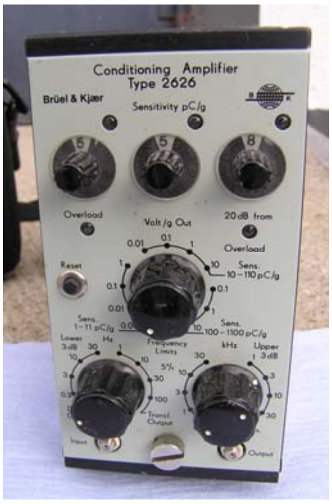


figure 25, amplifier

8.1.3 Oscilloscope and laptop

The signals from the amplifier were transferred to an oscilloscope, provided by LeCroy (figure 26). Inside the instrument an A/D-converter converted them to digital values and printed them on the screen. The time base on the oscilloscope was tuned (chapter 4.2). The oscilloscope has four memory spaces where measured data in time domain was saved. The data was transferred with a zero-cable to a laptop. The laptop had the software required for the oscilloscope.

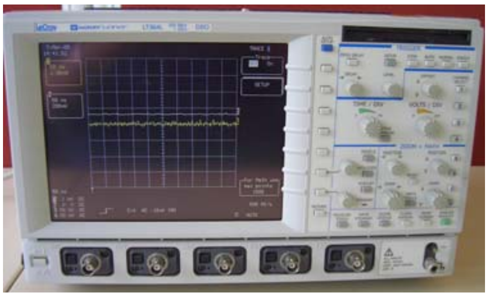


figure 26, oscilloscope

8.1.4 Run the ventilator

During the trials the ventilator was controlled over the CAN-bus by a computer with a Labview interface (figure 27).

figure 27, labview interface for fan control

8.2 Trials

8.2.1 24-25 October 2007

The first systemised measurements were captured in the end of October. The research object was a VA4111 and measurements were done at some few different rotation speeds. The sensor was applied at the exterior of the fan and no lowpass filter was used. The sample frequency (f_s) was 100 kS/s and 50 kS were sampled (N) during the measuring period.

The results of the early trials were suffering from many basic failures. Most of the theories in section 4 were unknown and no proper analysis was done. The rollover frequencies did not occur in the frequency spectrum. The position of the sensor is probably the main cause to this. Due to a software-failure all calculated frequencies were shifted $1*\Delta f$ Hz (formula 12, chapter 4.2.2). The function *fft()* in Scilab returns the value of 0 Hz in the element 1 and the value for $1*\Delta f$ Hz in element 2.

Despite the bad premises the measurements pulled the project in the right direction. The high sample frequency was sufficient to prevent aliasing, even if no lowpass filter was used. The small measurement period made the results sensible for improbable frequencies, but the frequency dependence of rotation speed was noticeable. Line frequencies, cooling element frequencies and blade pass frequencies were identified. The switching frequency was found as well, even though the origin of the frequency was unknown at the time.

8.2.2 8 November 2007

The search of the rollover frequencies was the main object for the measurements done 8 November at the same VA4111. The sensor was now attached directly on the bearing sleeve with a thread pin, where the acceleration zone of the rare bearing was assumed to be. Frequencies were measured at twelve different rotation speeds. Lowpass filter was used with the -3 dB damping frequency at 3 kHz, sample frequency was reduced to 50 kS/s and N was still 50 kS. Hann window was applied but no lowpass or window compensation and the steps with the amplitude calculation and single side spectrum were done in a wrong way (chapter 4.1).

At 9000 rpm an outer race rollover damage frequency was identified. The miscalculations when translating the measured amplitudes from mV to m/s^2, were well-known and the amplitudes were just used for comparison. Analysing tools were developed for measurement summation with and without consideration of rotation speed.

8.2.3 16 November 2007

After the measurements done 8 November, the suspiciousness over the cooling element as sours for vibrations causing bearing damages did grow. To make a proper comparison of the amplitudes new measurements were done on the 16th of November without any lowpass filter at the VA4111. To prevent aliasing the sample frequency was restored to 100 kS/s. Due to software progresses N could be chosen to 100 kS and the $1*\Delta f$-shifting could be removed. 36 measurements were done at the time. The amplitudes were still not proper calculated.

The results did show that the vibrations from the cooling element were totally dominant in the spectra. The weaker group of tones were identified (chapter 5.5.3), as well as harmonics of the stronger group. To analyse the results software tools to compare special tops from many measurement series and for producing of diagrams for energy content were developed.

8.2.4 29 November 2007

The first trial with a VA4130 was done 29 November. The sensor was initially attached with adhesive tape next to the rare bearing, but got lose. The measurements were redone the same afternoon. This time the sensor was attached with two-component adhesive. 35 measurements with each 100 kS were sampled with a velocity of 100 kS/s. No filter was used and the results were compared to the ones from the VA4111. The steps of frequency analysis to maintain the amplitudes were developed and proper followed.

The VA4130 does not have any cooling element which gave the spectra a totally different look. The data from the measurements were used as bases for a discussion about the imbalance of the rotor (chapter 5.4.2). For that purpose a software tool to get velocity spectra was manufactured. The switching frequency of the VA4130 and the four times the rotation speed sidebands were identified.

8.2.5 5 December 2007

The 5th of December was structure-borne sound measured from the VA4111 again. This time N was 100 kS and f_s was 100 kS/s. Totally 36 measurements was done.

Although a damaged bearing was inserted in the fan the rollover frequencies were not dominating the spectra at any rotation speed. Amplitude modulation between the cooling element frequencies and the blade pass frequency was observed.

8.2.6 18 December 2007

A damage bearing was inserted in the VA4130 and 36 measurements were done. The sampling frequency at 50 kS/s and the number of samples at 100 kS gave a smaller frequency interval. A lowpass filter was used, with the -3 dB damping frequency at 3 kHz. Lowpass filter compensation (chapter 4.5) was discovered and implemented.

The big measuring period (2 s) reduced the background noise. The swinging behaviour of the rotation speed did not reduce the accuracy remarkable. Rollover frequencies from inner race and cage damages were identified. Sidebands of the outer and inner race damage frequencies were observed.

8.2.7 14 January – 28 January 2008

A new bearing (which still had its caps on) was cocked in alcohol to determent the lubrication. The bearing was inserted in the VA4130 to study how damages occur and got worse. During two weeks the ventilator was running measurements were done regularly, every time at 16 different rotation speeds and at least two times a day (apart from the weekends). Between the measurements the fan speed was 15262 rpm. The same lowpass filter and sampling parameters were used as in the measurements done the 18th of December.

The idea was to run the ventilator until the bearing broke but after two weeks the experiment had to be interrupted. The method of cocking the bearing in alcohol was not successful enough and the ventilator was needed for other projects. In the frequency spectra rollover damages were found already at the first measurement, but the increasing trend of the frequencies was not as strong as predicted. The amplitude of rollover frequency for outer race damages is giving by figure 28. Groups of 16 measurements are compared. The swinging behaviour of the rotation speed may affect the measurements different and can be seen as an element of uncertainty.

When the rotor was removed and the rare bearing was visible, groves were found on the outside of the outer race (figure 29). This indicates rotation of the whole bearing in the bearing sleeve.

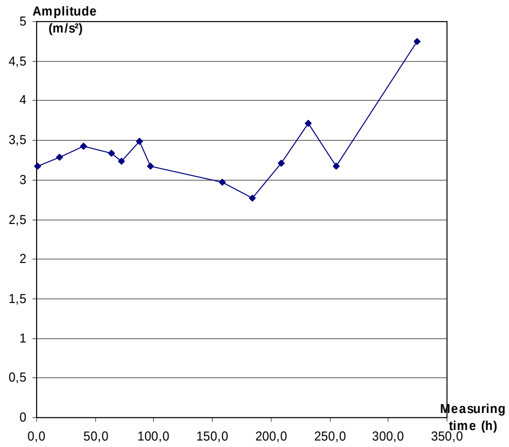


figure 28, average outer race damage frequencies

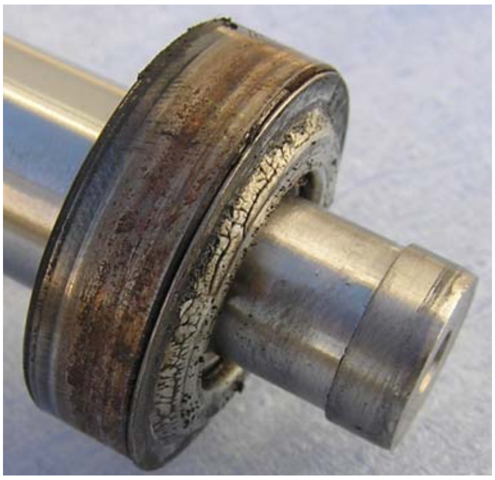


figure 29, bearing after two weeks of running

When analysing the results *statAnalyser.sce* some trends was able to observe. In figure 30 and 31 the development of the crest factor (chapter 6.1.3) and kurtosis factor (chapter 6.1.4) shows interesting tendencies. The lower curves in each of the figures are the factors calculated with the blind broadband diagnostic method. To calculate the upper curves a combination of broadband methods and methods for special frequencies have been used (chapter 6.3). The chosen frequencies are the same as in the implementation of *statAnalyser.sce* (Appendix A.8).

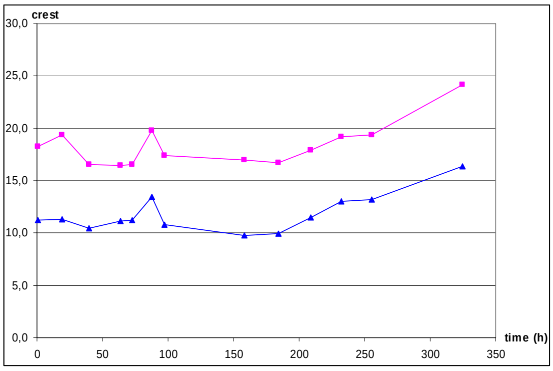


figure 30, crest. The upper curve is received with a combination of crest and a smart filter.

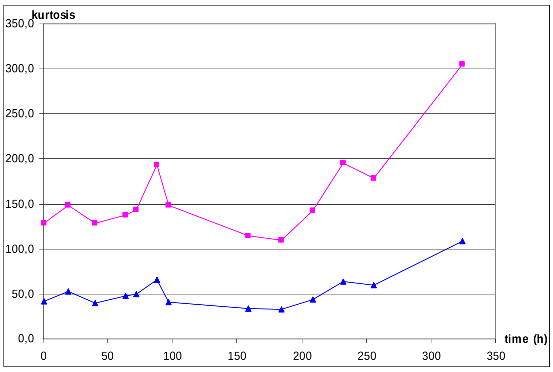


figure 31, kurtosis. The upper curve is received with a combination of kurtosis and a smart filter.

8.2.8 30 January - 4 February 2008

The two caps on a new bearing were removed and the lubrication was carefully washed out (figure 4, chapter 2.2.1). This time the bearing was not only cocked, also compressed air and solvent cleaning were used.

The bearing was inserted in the rear part of the VA4130, and same settings and filter were used as in the trial before. To discover how the temperature monitoring would react an extra temperature sensor was attached to the emergency bearing sleeve and the demanded and measured rotation speed on the CAN-bus were logged.

The 30th of January 8:48 the test was initialised. A first run of 16 measurements was captured right at the start (example of one in figure 32 and 33). 10:15 the same day the extra temperature sensor indicated an increase of 7 °C and the rotation speed decreased for a short moment before the controller restored it. At 12:49 the temperature increased 9 ºC and suddenly fell 14 ºC. At the same time the rotation speed was decreasing violently. The controller reacted to set the current to zero and start the fan over again. This was repeated a couple of times.

When the next run of measurements was captured from 14:17 until 15:01 a stench was noticeable. At low rotation speeds pounding was hearable. The shape of the time signal had changed remarkably (figure 34) and in the frequency spectra no peaks where identifiable at higher frequencies (figure 35).

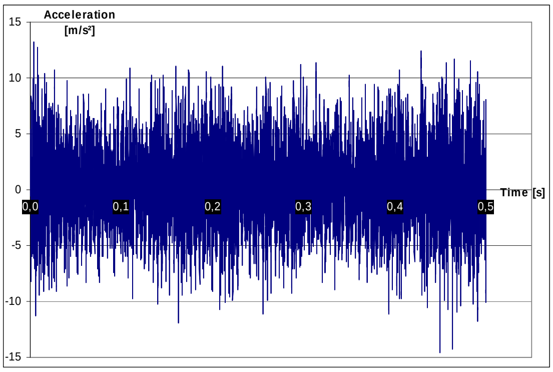


figure 32, 0.5 s of a 2 s measurement captured 9:07 30th of January 2008, at 13020 rpm.

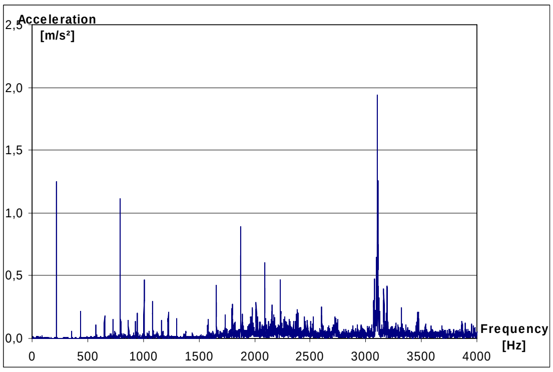


figure 33, frequency spectrum of the 2 s time signal partly shown in figure 32.

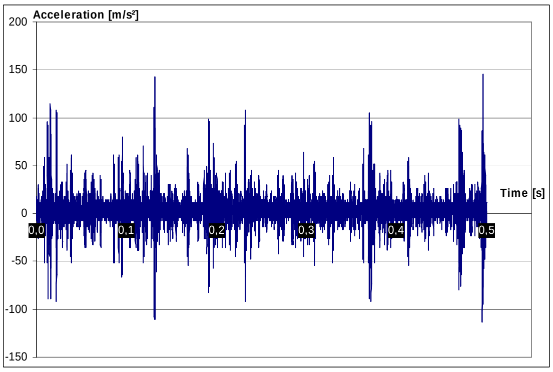


figure 34, 0.5 s of a 2 s measurement captured 14:33 30th of January 2008, 13023 rpm.

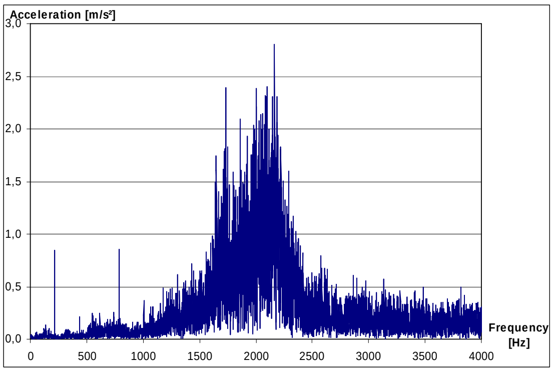


figure 35, frequency spectrum of the 2 s time signal partly shown in figure 34.

The fan was running for some more days, without any further changes in the measured data. At Monday the 4th of February, the test was stopped and the ventilator was opened. The inside of the fan was covered by black dust (figure 36) from the rotor rubbing the stator (chapter 5.4.6). The pin, which keeps the emergency bearing sleeve fix, was pressed out and the sleeve was able to rotate. Because of this the stationary as well as the extra temperature sensor were torn away. This might have happened at 12:49 when the temperature was drastically falling. The tip of the rotor (figure 37) had carved the inside of the emergency bearing sleeve (figure 38). The bearing was broken (figure 39) and from the cage only particles remained.

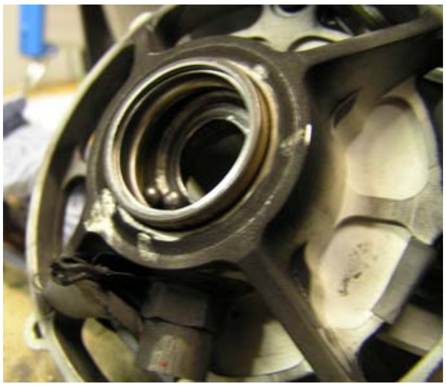


figure 36, the bearing sleeve covered with dust.

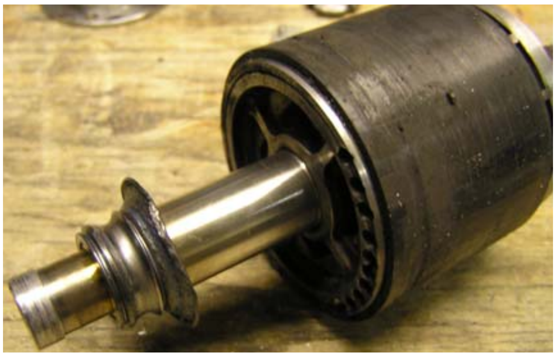


figure 37, the rubbed rotor. Notice the rubbed tip. The bearing inner race and a protective plate are still attached.

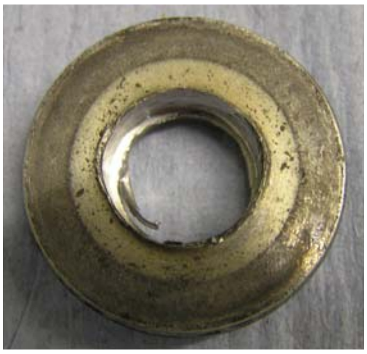


figure 38, the emergency bearing sleeve. The tip of the rotor has enlarged the hole.

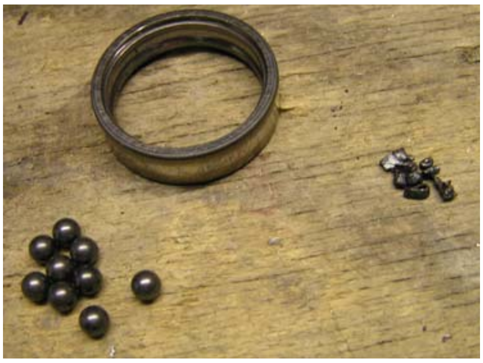


figure 39, bearing parts. From left: ceramic balls, outer race and the rest of the cage.

9. Conclusions and Further Ideas

Section 9 includes discussions about the results and possibilities for the future. In 9.1 the questions about the sources of the bearing problem are highlighted once again. Chapter 9.2 focuses on the main subject of this dissertation, how a condition monitoring system can be implemented.

9.1 Causes of bearing problem

Some results from the measurements were used as basis for a discussion of the causes of the bearing problems. This chapter sums the thoughts of four different theories up.

9.1.1 Air-cooling unit

It is a well known fact that high frequency vibration can accelerate bearing corrosion. In the VA4111 with K9 the amplitudes of the vibrations from the air-cooling unit are huge at some rotation speeds. In figure 21 (chapter 5.5.3) amplitudes measured at 11600 rpm come close to 70 m/s^2. The amplitudes correspond to acceleration which is proportional to energy. A theory of this energy causes the early bearing damages was considered.

The main argument against the theory is the experiences from other fans. VA4111 with K7 has the same problem although the amplitudes of the vibrations are much smaller from K7. VA4130 has not been on the market yet and not enough long-time-observations of the machine have been done. But the results so far indicate even more bearing problems by VA4130, even though it does not have any air-cooling element.

The air-cooling unit of the transistor does not include the solution of the problem. Even if it does affect the bearing in a negative way, there most be another main explanation to the early bearing damages.

Frequencies over 10 kHz has not been measured within this project, because of sensor limitations. But within the measurable band no other high frequency causes any vibrations with amplitudes like the ones from the air-cooling unit.

9.1.2 Imbalanced rotor

Due to formula 25 in chapter 5.4.2 the force of imbalance increases when the fan speed rises. This makes ventilators running at high speed especially sensitive for imbalanced rotors. If the forces on the bearings are exceeding the tolerances, the lifetime of the bearings will be reduced. The maximum speed of the VA4111 has been limited. Before the limitation was implemented the bearing problems were even more common.

The rotors and blades are balanced during the construction of the fans. Considerations about the reliability of the tolerance levels have been present for a long time, but nothing has been done to adjust the levels according to the bearing tolerances.

The measurements done within this project support the theory of imbalanced rotor. The amplitude level of the rotation speed frequency is high enough to cause forces which exceed the tolerance levels of the bearings. The imbalance of the rotor is not the only factor increasing the amplitude, but practical experiences from other measurements says it is the main cause.
In the future drastic measures have to be carried out to reduce the imbalance of the rotor.

9.1.3 Rotating outer race

Almost every time the rollover frequency of the outer race damage was observed, rotation speed sidebands were present as well. Mostly some of the sidebands were even stronger than the amplitude of the damage frequency. The main cause for these sidebands is a rotating outer race. The bearing is not fixed in the bearing sleeve and can slip.

The scratches on the outside of the outer race in figure 29 (chapter 8.2.7) are another proof of rotating bearing. When the other race of the bearing rubs the inside of the bearing sleeve corrosion occurs and the lifetime of the bearing gets dramatically reduced.

The assumption when the fans where developed was that the bearing would stay in is position. This assumption has been doubted for a while and within the A350 project a mechanism to maintain the bearing in its position is developed. With the results of the structure-borne sound measurements, the doubts have been justified.

9.1.4 Misaligned bearing

The bearing is running free in the bearing sleeve. If the bearing gets misaligned (figure 20, chapter 5.4.4) during the assembly, its lifetime will be reduced. To prevent the risk, special tools can be used which screw all screws on the bearing sleeve at the same time. This is not done by AOA. The mechanism to maintain the bearing in its position developed within the A350 project may reduce bearing misalignment. If misalignment is part of the explanation of the bearing problem could not be proven within this dissertation, but the possibility has to be considered in the future.

9.2 Possibilities of realising a condition monitoring system

9.2.1 Conclusions from trials

In the results of the measurements, bearing frequencies are clearly identifiable. The requirement for capturing bearing frequencies is a proper placing of the sensor according to chapter 3.3.

In the trial from 14 January – 28 January 2008 (chapter 8.2.7) the trends of the outer race damage frequency (figure 28) and two broadband diagnostic methods (figure 30 and 31) were observed. The trends of the methods were pointing in the right direction but the test was finished to early to make a conclusion of a suitable place of an alarm level.

The experiences earned from the trail 30 January - 4 February 2008 (chapter 8.2.8) points at other possibilities with a condition monitoring system. The poundings which occurred were caused by a rubbing rotor (chapter 5.4.6). The stationary temperature sensor used in the test was torn away and not able to give a signal for the ventilator turnoff. A condition monitoring system would react more efficient. In figure 32-35 rubbing rotor is recognisable both in the time domain signal and the frequency spectrum. To analyse the time domain signal broadband diagnostic methods can be used.

In the trial the fan was running almost 53 hours after the suspected point when the emergency bearing sleeve got loose. The damages on the fan (figure 36-39) would have been reduced if the fan had been turned off directly at the start point of the poundings, but sever damages would still occur. A condition monitoring system which only recognises poundings will not prevent ventilator damages, only reduce them.

9.2.2 Alarm levels

To estimate the exactly remaining lifetime of the bearing is impossible. The goal for a condition monitoring system should be to categorise the condition of the bearing. E.g. three different categories can be used: “undamaged”, “partly damaged” and “totally damaged”. The second step can result in a warning and the third in a shut down of the fan.

To develop fixed alarm levels more measurements has to be done. The alarm levels will be different for VA4130 and VA4115. Tests at different machines of the same model have to be done as well to exclude individual differences.

The way the frequencies behave in the spectrum changes with the rotation speed. A condition monitoring system can observe frequencies at a special fan speed or a number of different specified rotation speeds as done in the trials. It will probably be impossible for the system to command a fan speed, according to the safety requirements on the ventilator software. The condition monitoring system has to wait until the wanted fan speed (or an approximation of it) is commanded.

The circumstances under which the fan works, will also affect the result. The extern vibrations in an aeroplane can not be ignored. Within the phases of starting and landing unpredictable forces may affect the sensor. The recommendation is to measure the vibrations only within the flight phase.

9.2.3 The next step

To examine the possibilities, tests have to be done on the different fans on-board aeroplanes. An idea is to implement a first version of a condition monitoring system, without permission to turn the fan off, only to gather data. When enough measurements are collected the alarm levels can be set for the different ventilators at different fan speeds.

Bibliography

1. *An analytical approach to solving motor vibration problems*, W. Finley, M. Hodowanec and W. Holter, PCIC-99-20, 1999, available on the internet at: http://www2.sea.siemens.com/NR/rdonlyres/FC008550-952A-4EE3-BFFB-A78B078E1608/0/AnAnalyticalApproachtoSolvingMotorVibrationProblems.pdf

2. *Ausarbeitung_Wuchten*, R. Berger, AOA, 2007

3. *Computer-controlled systems theory and design*, K. Åström and B. Wittenmark, Prentice Hall, ISBN 0-13-314899-8, 1997

4. *Description of Peakvue and illustration of its wide array of applications in fault detection and problem severity assessment*, J. Robinson and J. Berry, 2001, available on the internet at: http://www.maintsys.com.au/images/support/PeakVueReliability2001Conference.pdf

5. *Determination of the temperature dependence of Young's modulus for stainless steel using a tuning fork*, L. Bates, T. Beach and M. Arnott, 1999, available on the internet at: http://www.jurp.org/articles/tuningforkpaper.pdf

6. *Digital signal processing a computer-based approach*, S. Mitra, McGraw-Hill, ISBN 0-07-252261-5, 2001

7. *Early warning fault detection in rolling element bearings using micrologenveloping*, SKF Condition Monitoring, CM3021, 1996

8. *Failure modes of rolling element bearings*, T Reeves, 1996, available on the internet at: http://www.compsys.com/drknow/aplpapr.nsf/06b6f5a4de2eae6285256a3f004d9758/3ebbed0041ac9072852565a2005fd9aa?OpenDocument

9. *Fast Fourier Transform (FFT)*, Reliable Software, 2006, available on the internet at: http://www.relisoft.com/Science/Physics/fft.html

10. *Kühlkörpervergleich VA4111 Schallpegel-Analyse*, H. Sturm and T. Cebulla, AOA, 2003

11. *Machinery Vibration Diagnostics*, D. Stevens, 2001, available on the internet at: http://www.vibanalysis.co.uk/vibanalysis/

12. *Maschinendiagnose an Industriegetrieben – Grundlagen*, R. Wirth, available on the internet at: http://www.maschinendiagnose.de/Fachbeitrage/Maschinendiagnose_Teil_1/Maschinendiagnose_Teil_1.pdf

13. *Maschinendiagnose an Industriegetrieben – praktische Signalidentifikation*, R. Wirth, available on the internet at: http://www.maschinendiagnose.de/Fachbeitrage/Maschinendiagnose_Teil_2/Maschinendiagnose_Teil_2.pdf

14. *Measurement of structure-borne sound of rolling element bearing in machines and plants for evaluation of state condition*, VDI (Verein Deutsche Ingenieure) 3832, 2007

15. *NI developer zone*, National Instruments, available on the internet at:
http://zone.ni.com/devzone/cda/main

16. *The scientist and engineer's guide to digital signal processing*, S. Smith, California Technical Publishing, ISBN 0-9660176-6-8, 1999

17. *Vibman*, DLI Engineering, 2008, available on the internet at:
http://www.dliengineering.com/vibman.htm

18. *Vibration monitoring and accelerometers by Wilcoxon Research*, Wilcoxon Research, available on the internet at:
http://www.me.ua.edu/me360/PDF/Wilcoxon_Research_Vibration_Notes.pdf

19. *Wikipedia - the free encyclopedia*, Wikimedia Foundation, Inc, available on the internet at:
http://www.wikipedia.org/

Appendices

A. Scilab

A.1 frekAnalyser.sce

```
//frekAnalyser.sce
//Frequency analysis for measurement data. The file reads data from txt-file.
//The user has to specify the data properties for the measurement.
//x has to be replaced with a value, filename with filenames and paths
//The frequency analysis is done according to the steps in the report chapter 4.1
//The result is plotted and saved to make other files able to use it.
//Anders Mattson, 13 Mars 2008

//Main
  clear;                    //All variables are cleared
  clf                       //The screen is cleared from Scilab-plotting-windows

  //Data properties
  sampFrek=x;               //Sample rate
  N=x;                      //Amount of Samples in measurement period
  winMax=x;                 //Window range maximum in Hz
  winMin=x;                 //Window range minimum in Hz
  ampfac=x;                 //Amplification factor at the amplifier
  time=x;                   //Timestamp (min)
  rotSpeed=x/60;            //Rotation speed in Hz=rpm/60

  //Loading data
  y=read('filename.txt',N,1);
                            //A N*1-matrix of data is read from a txt-file

  //Basic calculation
  n=0:N-1;                  //Array with sample numbers
  NF=N/2;                   //Nyquist frequency sample
  nF=0:NF-1;                //Array with sample numbers until NF
  f=nF/N*sampFrek;          //Frequency corresponding to sample number
  winMaxSamp=round(winMax/sampFrek*N);
                            //Window range maximum sample
  winMinSamp=round(winMin/sampFrek*N);
                            //Window range minimum sample

  //Amplification reduction
  y=y/ampfac;               //Compensation for amplifier setting

  //Hann window
  k=2*%pi/(N-1);            //First step of Hann window calculation
  hW=(1-cos(n*k))*0.5;//Second step of Hann window calculation
  yHannTP=y.*hW';           //Applying Hann window on measure data

  //FFT
  FHannTP=fft(yHannTP);//Fast Fourier Transformation

  //Window compensation
  FTP=FHannTP/0.5;          //Compensation for Hann window

  //Amplitude calculation
  magFTPds=sqrt((imag(FTP))^2+(real(FTP))^2)/N;
                            //Amplitude calculated

  //Single Side Spectrum
  magFTPss(1)=magFTPds(1);
                            //First element in single side spectrum
  magFTPss(2:NF-1)=magFTPds(2:NF-1)*2;
                            //All elements besides first and Nyquist in single side spectrum
  magFTPss(NF)=magFTPds(NF);
                            //Nyquist sample in single side spectrum

  //Low pass filter compensation
  G1=1+(f./3000).^4;        //First step to calculate lpfc-factor
  LPFC=sqrt(G1);            //Second step to calculate lpfc-factor
  magF=magFTPss.*LPFC';
                            //Applying lpfc-factor

  //Erasing offset
  magF(1)=0;                //Erasing uninteresting offset

  //Plotting
  plot(f(winMinSamp+1:winMaxSamp+1),magF(winMinSamp+1:winMaxSamp+1)')
                            //Plotting frequencies on x-axis and amplitudes on y-axis
  xlabel('Frequency [Hz]')
                            //Label x-axis
  ylabel('Acceleration [m/(s^2)]')
                            //Label y-axis
  legend(string(rotSpeed*60) + ' rpm at time ' + string(time),[1],%f);
                            //Names the curve in box at the upper right corner

  //Save data
  save('filename.dat',magF,rotSpeed,time);
                            //Saves amplitudes, rotation speed and timestamp in dat-file

  clear;                    //All variables are cleared
```

A.2 getTops.sce

```
//getTops.sce
//Calculates the tops in a frequency spectrum and writes the highest on the prompt.
//The user has to specify the data properties for the measurement.
//x has to be replaced with a value, filename with filenames and paths
//Anders Mattson, 13 mars 2008

//Main
  clear;                    //All variables are cleared;

  //Data properties
  sampFrek=x;               //Sample rate
  N=x;                      //Amount of Samples in measurement period
  winMax=x;                 //Window range maximum in Hz
  winMin=x;                 //Window range minimum in Hz
  noOfTop=x;                //Number of highest tops to be written on the prompt

  //Translate window ranges in samples
  winMaxSamp=round(winMax/sampFrek*N);
                            //Window range maximum sample
  winMinSamp=round(winMin/sampFrek*N);
                            //Window range minimum sample

  //Loading data
  load('filename.dat','magF','rotSpeed','time');
                            //Loading data from frekAnalyser.sce

  //Find all tops
  topCount=0;               //Top counter
  up=1;                     //Boolean which indicates direction (1=true)
  for r=winMinSamp+2:winMaxSamp+2,
                            //Loop running through window samples
    if magF(r)<magF(r-1) & up==1 then
                            //If "this amplitude"<"the last one" and direction==up then top found
      topCount=topCount+1;
                            //Top counter added with one
      top(topCount,1)=round((r-2)/N*sampFrek);
                            //Frequency saved in top-matrix
      top(topCount,2)=round(1000*magF(r-1))/1000;
                            //Amplitude with 3 decimals saved in top-matrix
      up=0;                 //Direction set to down
    elseif magF(r)>magF(r-1) then
      up=1;                 //If "this amplitude">"the last one" then direction set to up
    end,
  end

  //Get the highest tops
  for nm=1:noOfTop,         //Loop running through as many times as tops to be written
    temp=[top(1,1) top(1,2) 1];
```

```
                              //Temporary vector for a top that candidates to be the highest
    for r=2:max(size(top)),
      if top(r,2)>temp(2) then
        temp=[top(r,1) top(r,2) r];
      end,                    //All top being compared with the temporary one
    end
    maxes(nm,:)=temp;  //The highest top is found and placed in maxes
    top(temp(3),:)=[0 0];
                              //The highest top is erased
  end

  //Writing
  maxes                       //Highest tops are written in the prompt

  clear;                      //All variables are cleared;
```

A.3 makeMatrix.sce

```
//makeMatrix.sce
//Creates one matrix of all data in one measurement run.
//User has to give filenames with filenames and paths
//The result is saved in a dat-file
//Anders Mattson, 13 Mars 2008

//Main
  clear;                      //All variables are cleared
  stacksize(15000000);  //Memory space enlarged

    //Data properties
  fileNameVec=['filename.dat' 'filename.dat' 'filename.dat' ....];
                              //Vector with filenames
  amOfMeas=max(size(fileNameVec));
                              //Amount of Measurements in a run

  //Loading data
  for mnum=1:amOfMeas
                              //The Loop runs amOfMeas times
    load(fileNameVek(mnum),'magF','rotSpeed','time');
                              //The files are loaded
    amp(:,mnum)=magF;
                              //The amplitudes are saved in a matrix
    rotSpeVek(mnum)=rotSpeed;
                              //The rotation speeds are saved in a vector
    times(mnum)=time;  //The time stamps are saved in a vector
  end

  //Saving data
  save('filename.dat',amp,rotSpeVek,times,amOfMeas);
                              //The matrix and the vectors are saved

  //Confirmation
  ready='true'                //Writes ready on the prompt

  stacksize(5000000);    //Memory space reset
  clear;                      //All variables are cleared
```

A.4 sum.sce

```
//sum.sce
//Sums all the measurement in one run up and plots the result.
//The user has to specify the data properties for the measurement.
//x has to be replaced with a value, filename with filenames and paths
//Anders Mattson, 13 Mars 2008

//main
  clear;                      //All variables are cleared
  stacksize(15000000);  //Memory space enlarged
  clf                         //The screen is cleared from Scilab-plotting-windows

  //Data Properties
  sampFrek=x;                 //Sample rate
  N=x;                        //Amount of Samples in measurement period
  winMax=x;              //Window range maximum in Hz
  winMin=x;              //Window range minimum in Hz

  //Loading data
  load('filename.dat','amp','rotSpeVek','times','amOfMeas');
                         //Loads file made from makeMatrix.sce

  //Basic calculation
  n=0:N-1;               //Vector with sample numbers
  NF=N/2;                //Nyquist frequency sample
  nF=0:NF-1;             //Vector with sample numbers until NF
  f=nF/N*sampFrek;       //Frequency corresponding to sample number
  winMaxSamp=round(winMax/sampFrek*N);
                         //Window range maximum sample
  winMinSamp=round(winMin/sampFrek*N);
                         //Window range minimum sample

  //Summation
  for samp=1:max(size(amp)),
    sumAmp(samp)=sum(amp(samp,1:amOfMeas))/amOfMeas;
  end                    //Sums amp up and divides with amOfMeas

  //Plotting sum
  plot(f(winMinSamp+1:winMaxSamp+1),sumAmp(winMinSamp+1:winMaxSamp+1),'k')
                         //Plotting frequencies on x-axis and sums of amplitudes on y-axis
  xlabel('Frequency [Hz]')
                         //Label x-axis
  ylabel('Acceleration [m/(s^2)]')
                         //Label y-axis
  legend('Sum of all measurements in the run independent of rotation speed',[1]);
                         //Names the curve in box at the upper right corner

  //Cleaning
  stacksize(5000000);    //Memory space reset
  clear;                 //All variables are cleared
```

A.5 sumNorm.sce

```
//sumNorm.sce
//Sums all the measurement in one run norm up and plots the result.
//The user has to specify the data properties for the measurement.
//x has to be replaced with a value, filename with filenames and paths
//Anders Mattson, 13 Mars 2008

//Main
  clear;                 //All variables are cleared
  stacksize(15000000);   //Memory space enlarged
  clf                    //The screen is cleared from Scilab-plotting-windows

  //Data properties
  sampFrek=x;            //Sample rate
  N=x;                   //Amount of Samples in measurement period
  winMax=x;              //Window range maximum in Hz
  winMin=x;              //Window range minimum in Hz
  tastKoeff=0.01;        //Space between the lines in plot (0.01 recommended)
  maxNorm=12;            //Maximum Norm (12 recommended)

    //Loading data
  load('filename.dat','amp','rotSpeVek','times','amOfMeas');
                         //Loads file made from makeMatrix.sce

  //Basic calculation
  n=0:N-1;               //Vector with sample numbers
  winMaxSamp=round(winMax/sampFrek*N);
                         //Window range maximum sample
  winMinSamp=round(winMin/sampFrek*N);
                         //Window range minimum sample

  //Norming
  ampNor=zeros(maxNorm/tastKoeff,amOfMeas);
                         //Creates vector where norms can be added
  ind=ones(1,amOfMeas);
                         //Creates vector for indexes
  for koeff=1:maxNorm/tastKoeff,
                              //Loop through all coefficients
    for mnum=1:amOfMeas,//Loop through all measurements in the run
      while (((ind(1,mnum)-1)/N*sampFrek)/rotSpeVek(mnum))<=(koeff*tastKoeff),
                              //As long as the coefficient is bigger than the frequency(index)/rotation speed then
        ampNor(koeff,mnum)=ampNor(koeff,mnum)+amp(ind(1,mnum),mnum);
                              //Amplitude is added to a coefficient depending matrix
        ind(1,mnum)=ind(1,mnum)+1;
                              //Index of frequency gets bigger
      end
    end
    ampNorSum(koeff)=sum(ampNor(koeff,:))/amOfMeas;
  end

  //Plotting sum
  koeff=(tastKoeff:tastKoeff:maxNorm)';
                              //Creates a coefficient vector for the x-axis
  plot(koeff(3:maxNorm/tastKoeff),ampNorSum(3:maxNorm/tastKoeff),'b')
                              //Plotting norm factors on x-axis and sums of amplitudes on y-axis
  xlabel('Frequency(Rotation Speed)[]');
                              //Label x-axis
  ylabel('Acceleration [m/(s^2)]')
                              //Label y-axis
  legend('Sum of all amplitudes dependent on rotation speed',[1]);
                              //Names the curve in box at the upper right corner

  //Saving
  save('filename.dat',ampNor,rotSpeVek,times,amOfMeas);
                              //The new matrix are saved with the old vectors in a dat-file

  stacksize(5000000);    //Memory space reset
  clear;                      //All variables are cleared
```

A.6 trackTop.sce

```
//trackTop.sce
//Toll created to track a top, either a norm top or a frequency top.
//If the user wants to track a frequency top she/he uses the output-file from makeMatrix.sce as input.
//If the user wants to track a norm top she/he uses the output-file from sumNorm.sce as input.
//Anders Mattson, 13 Mars 2008

//Main
  clear;                      //All variables are cleared
  clf                         //The screen is cleared from Scilab-plotting-windows
  stacksize(15000000);  //Memory space enlarged

  //Loading data
  load('filename.dat','amp','rotSpeVek','times','amOfMeas')
                              //Loading data from madeMatix.scr to track frequency
                              //Loading data from sumNorm.scr to track norm top (exchange 'amp' with 'ampNor')

  //Creates plot Matrix top
  for mnum=1:ampOfMeas,
    top(1,mnum)=sum(amp(x:x+2,mnum));
                              //Choose a sample number to track x to follow
                              //exchange amp to ampNor in case of norm top tracking
    top(2,mnum)=sum(amp(x:x+2,mnum));
                              //More tops can be studied at the same time
    top(3,mnum)=sum(amp(x:x+2,mnum));
  end

  //Plotting
  plot(rotSpeVek,top(1,:),rotSpeVek,top(2,:),rotSpeVek,top(3,:))
  xlabel('rotation speed[rpm]');
  ylabel('acceleration[m/(s^2)]');

  stacksize(5000000);    //Memory space reset
  clear;                      //All variables are cleared
```

A.7 intVelDis.sce

```
//intVelDis.sce
```

```
//This file using measurement acceleration data and plot velocity curve and displacement curve
//User has to give filenames with filenames and paths and sampFrek, N and ampfac
//Anders Mattson, 13 Mars 2008

//Main
  clear;                //All variables are cleared
  clf                   //The screen is cleared from Scilab-plotting-windows
  stacksize(20000000);  //Memory space enlarged

  //Data properties
  sampFrek=x;           //Sample rate
  N=x;                  //Amount of Samples in a measurement period
  ampfac=x;             //Amplification factor at the amplifier

  //Loading data
  y=read('filename.txt',N,1);
                        //A N*1-matrix of Data is read from a txt-file

  //Basic calculation
  n=0:N-1;              //Sample array created
  t=n/sampFrek;         //Time array created

  //Amplification reduction
  y=y/ampfac;           //Compensation for amplifier setting

  //Acceleration
  a=y;                  //Renamed for educational reasons
  subplot(3,1,1)        //Splits the plotting window. Window 1 is chosen.
  plot(t,a)             //Plotting the time domain acceleration curve
  xlabel('Time [s]')    //Label x-axis
  ylabel('Acceleration [m/(s^2)]')
                        //Label y-axis

  //Lowpass filtration of acceleration curve
  Fa=fft(a);            //Fast Furrier Transformation
  Fa(1:11)=0;           //Low frequencies excluded
  Fa(N-10:N)=0;         //Low mirror frequencies excluded
  afil=real(ifft(Fa));  //Back transformation

  //Velocity
  v=cumsum(afil)/N;     //Calculate the velocity
  subplot(3,1,2)        //Window 2 is chosen.
  plot(t,v')            //Plotting the time domain velocity curve
  xlabel('Time [s]')    //Label x-axis
  ylabel('Velocity [m/s]')
                        //Label y-axis

  //Lowpass filtration of velocity curve
  Fv=fft(v);            //Fast Furrier Transformation
  Fv(1:11)=0;           //Low frequencies excluded
  Fv(N-10:N)=0;         //Low mirror frequencies excluded
  vfil=real(ifft(Fv));  //Back transformation

  //Displacement
  s=cumsum(vfil)/N;     //Calculate the displacement
  subplot(3,1,3)        //Window 3 is chosen
  plot(t,s')            //Plotting the time domain displacement curve
  xlabel('Time [s]')    //Label x-axis
  ylabel('Displacement [m]')
                        //Label y-axis

  stacksize(5000000);   //Memory space reset
  clear;                //All variables are cleared
```

A.8 statAnalyser.sce

```
//statAnalyser.sce
//Using broadband diagnostic methods or combinations from these and for analysing special frequencies.
//Inputs from many measurement runs each processed with sumNor.sce
//The choice of which special frequencies to observe can be done by the user. This is only an example.
//The weight is also a choice to be tuned by the user.
//topbreNorm, maxNorm, numOfMesRuns and filenames has to be set by the user.
//The result is plotted.
```

```
//Anders Mattson, 13 Mars 2008

clear;                  //All variables are cleared

//Function for calculation rollover frequency factors
function ubFac=getFactors()

  //Bearing parameters
  z=9;                  //Number of rolling elements
  dw=4.7;               //Diameter of rolling element
  dpw=18.7+dw;          //Pitch circle diameter
  alpha=19*%pi/180;     //Contact angle (approximation)

  //Calculating a choice of interesting frequency factors
  ubFac(1,1)=0.5*(1-(dw*cos(alpha))/dpw);
                        //Cage rollover frequency (standard case)
  ubFac(2,1)=0.5*z*(1-(dw*cos(alpha))/dpw);
                        //Outer race rollover frequency
  ubFac(3,1)=dpw/dw*(1-(((dw*cos(alpha))/dpw)^2));
                        //Rolling element rollover frequency
  ubFac(4,1)=0.5*z*(1+(dw*cos(alpha))/dpw)-1;
                        //Inner race rollover frequency sideband
  ubFac(5,1)=0.5*z*(1+(dw*cos(alpha))/dpw);
                        //Inner race rollover frequency
  ubFac(6,1)=0.5*z*(1-(dw*cos(alpha))/dpw)+2;
                        //Outer race rollover frequency sideband
  ubFac(7,1)=0.5*z*(1+(dw*cos(alpha))/dpw)+1;
                        //Inner race rollover frequency sideband
  ubFac(8,1)=0.5*z*(1-(dw*cos(alpha))/dpw)+3;
                        //Outer race rollover frequency sideband
  ubFac(9,1)=2*ubFac(2,1);
                        //Outer race rollover frequency harmonic
  ubFac(10,1)=0.5*z*(1-(dw*cos(alpha))/dpw)+4;
                        //Outer race rollover frequency sideband
  ubFac(11,1)=0.5*z*(1-(dw*cos(alpha))/dpw)+5;
                        //Outer race rollover frequency sideband
  ubFac(12,1)=0.5*z*(1-(dw*cos(alpha))/dpw)+6;
                        //Outer race rollover frequency sideband
  ubFac(13,1)=2*ubFac(5,1);
                        //Inner race rollover frequency harmonic
  ubFac(14,1)=3*ubFac(2,1);
                        //Outer race rollover frequency harmonic

  //Weight for the factors
  ubFac(1,2)=2;
  ubFac(2,2)=1;
  ubFac(3,2)=2;
  ubFac(4,2)=2;
  ubFac(5,2)=2;
  ubFac(6,2)=1;
  ubFac(7,2)=2;
  ubFac(8,2)=1;
  ubFac(9,2)=1;
  ubFac(10,2)=1;
  ubFac(11,2)=1;
  ubFac(12,2)=1;
  ubFac(13,2)=2;
  ubFac(14,2)=1;
endfunction

//Creats a filter suitible for filtering interesting frequencies.
function ubFilt=getFilter(Sampmax,fNorm,topbreNorm)
  ubFilt=zeros(Sampmax,1);
                        //Creating array with zeros corsponding every sample
  ubFac=getFactors();   //The function get factors is started
  for samp=1:Sampmax    //Loop for every sample
    for ub=1:max(size(ubFac))
                        //Loop for every frequency factor
      if ubFac(ub,1)-topbreNorm/2<=fNorm(samp) & ubFac(ub,1)+topbreNorm/2>=fNorm(samp) & ubFac(ub,2)>ubFilt(samp,1) then
                        //If the sample-loop reached an area around the current frequency factor of the
                        //factor-loop and the weight of this factor is bigger than the instantaneous.
        ubFilt(samp,1)=ubFac(ub,2);
```

```
                      //Then the filter is raist at this point
   end,
  end
 end
endfunction

//Main
 clf                  //The screen is cleared from Scilab-plotting-windows
 stacksize(20000000); //Memory space enlarged

 //Data properties
 topbreNorm=x;        //The with of the normtop (e.g. 0.04)
 maxNorm=12;          //Maximum Norm (12 recomended)
 numOfMesRuns=x;      //Number of measurement runs

 //Basic calculation
 fNorm=(0.01:0.01:maxNorm)';
                      //Creats a array with norms
 Sampmax=max(size(fNorm));
                      //Amounts of Samples in the filter

 ubFilt=getFilter(Sampmax,fNorm,topbreNorm);
                      //Creating the filter with using function getFilter

 for me=1:numOfMesRuns,
                      //Loop for every measuring run

   //Loading data
   load('filename.dat','ampNor','drehzahl','times','Mnum');

   time(me)=mean(times);
                      //Calculate average times

   for samp=1:Sampmax,
                      //Loop which sums all the amplitude in one run up
     ampNorSum(samp)=sum(ampNor(samp,:))/Mnum;
   end

   ampFil=ampNorSum.*ubFilt(:,1);
                      //Applying filter

   //Calcalation of root-mean-square with and without filter
   rms(me)=sqrt(sum(ampFil.^2)/Sampmax);
   rmsF(me)=sqrt(sum(ampNorSum.^2)/Sampmax);

   //Calcalation of maximum absolute value with and without filter
   mav(me)=max(abs(ampFil));
   mavF(me)=max(abs(ampNorSum));

   //Calcalation of crest factor with and without filter
   crest(me)=mav(me)/rms(me);
   crestF(me)=mavF(me)/rmsF(me);

   //Calculation of kurtosis factor with and without filter
   HFmen=sum(ampFil)/Sampmax;
   Fmen=sum(ampNorSum)/Sampmax;
   kurtosis(me)=Sampmax*sum((ampFil-HFmen).^4)/((sum((ampFil-HFmen).^2))^2)-3;
   kurtosisF(me)=Sampmax*sum((ampNorSum-Fmen).^4)/((sum((ampNorSum-Fmen).^2))^2)-3;
 end

 //Plotting RMS with and without filter
 subplot(4,1,1)       //Splits the plotting window. Window 1 is chosen.
 plot(time,rms,time,rmsF)
 xlabel('Time [s]')   //Label x-axis
 ylabel('RMS')        //Label y-axis

 //Plotting MAV with and without filter
 subplot(4,1,2)       //Window 2 is chosen.
 plot(time,mav,time,mavF)
 xlabel('Time [s]')   //Label x-axis
 ylabel('MAV')        //Label y-axis

  //Ploting Crest with and without filter
```

```
subplot(4,1,3)           //Window 3 is chosen.
plot(time,crest,time,crestF)
xlabel('Time [s]')       //Label x-axis
ylabel('Crest')          //Label y-axis

//Ploting Kurtosis with and without filter
subplot(4,1,4)           //Window 4 is chosen.
plot(time,kurtosis,time,kurtosisF)
xlabel('Time [s]')       //Label x-axis
ylabel('Kurtosis')       //Label y-axis

stacksize(5000000);      //Memory space reset
clear;                   //All variables are cleared
```