I hereby recommend that the thesis prepared under my supervision by Paul J. Crabill entitled An Investigation of Signal Processing Techniques Used for Rotor Dynamic Fault Detection be accepted as fulfilling this part of the requirements for the degree of Master of Science

Approved by:

Robert W. Rost
Randall J. Alleman
David L. Brown
AN INVESTIGATION OF SIGNAL PROCESSING TECHNIQUES
USED FOR ROTOR DYNAMIC FAULT DETECTION

a thesis submitted to the

Department of Mechanical Industrial and Nuclear Engineering
College of Engineering

UNIVERSITY OF CINCINNATI

in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

1992

by
Paul John Grabill
BSME University of Cincinnati 1986

Committee Chair: Dr. R. Rost
ABSTRACT

Several signal processing techniques were evaluated for their ability to detect rotor dynamic faults. A rotor rig was constructed to test several typical rotor dynamic faults. These faults included: a bearing inner race defect, a bearing outer race defect, a bearing rolling element defect, rotor unbalance, and shaft misalignment. Vibration data was acquired from a baseline (no-fault) condition, as well as from each of the known fault conditions. The vibration data was processed with the following signal analysis algorithms:

- Narrow band spectrum analysis
- Shock pulse analysis
- Cepstrum analysis
- Signal demodulation
- Singular value decomposition

Some of the methods were tested using analytical fault data to demonstrate the different features of the algorithms. The algorithms were then used to process the rotor rig fault data. Each of the different methods were evaluated for the ability to detect the faulty condition, as well as identify the location of the fault. A relative comparison was made of each of these methods for the application to health monitoring and fault detection.

It is shown in this thesis how the cepstrum and signal demodulation techniques were effective in locating the fault, while the shock pulse method and singular value decomposition were used to identify the start of a unhealthy condition.
ACKNOWLEDGMENTS

The author wishes to express his gratitude and appreciation to Dr. Robert Rost, Dr. Randy Allemang, and Dr. Dave Brown for their help and guidance at U.C.'s Structural Dynamics Research Lab.

The author is grateful for the support given by NASA's Health Monitoring Technology Center for Space Propulsion Systems, which sponsored this research, and for the encouragement and support from GE Aircraft Engines, especially Larry Bach.

Special thanks go to the Bently Nevada corporation for their help in constructing the rotor rig, and for the health monitoring training.

The author wishes to thank the Barden Corporation for the help with providing bearings with known faults installed.

The author would like to express his deep gratitude to his mother and father for their ever present encouragement throughout all stages of his education.

Finally the biggest thanks go to my wife, Barbara. Without her patience, sacrifices and encouragement, this work would have been impossible. To her, my most sincere gratitude for her help with the graphic design of this thesis.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>viii</td>
</tr>
<tr>
<td>1.0 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Types of faults</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Signal processing techniques</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Test rig description</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Scope of thesis</td>
<td>6</td>
</tr>
<tr>
<td>2.0 Signal Processing Techniques Theory</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Spectrum analysis</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Shock pulse analysis</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Cepstrum analysis</td>
<td>10</td>
</tr>
<tr>
<td>2.4 Demodulation analysis</td>
<td>14</td>
</tr>
<tr>
<td>2.5 Singular value decomposition</td>
<td>21</td>
</tr>
<tr>
<td>2.6 Other analysis techniques</td>
<td>26</td>
</tr>
<tr>
<td>3.0 Rotor Fault Testing</td>
<td>28</td>
</tr>
<tr>
<td>3.1 Rotor rig</td>
<td>28</td>
</tr>
<tr>
<td>3.2 Test sequence</td>
<td>33</td>
</tr>
<tr>
<td>3.3 Data acquisition</td>
<td>34</td>
</tr>
</tbody>
</table>
4.0 Test Results.................................................................40
  4.1 Spectrum analysis..................................................40
  4.2 Shock pulse analysis...............................................45
  4.3 Cepstrum analysis................................................49
  4.4 Demodulation analysis..........................................53
  4.5 Singular value decomposition.................................57

5.0 Conclusion...............................................................61

6.0 References......................................................................68

Appendices
A. MATLAB Script Files.....................................................A-1
B. Spectrums from all sensors.........................................B-1
C. Rotor Fault Frequency Identification Program..................C-1
D. Spectrums for Load and Unbalance Configurations...........D-1
E. Cepstrums for Load and Unbalance Configurations...........E-1
F. Demodulation for Load and Unbalance Configurations........F-1
G. SVD for Load and Unbalance Configurations...................G-1
List of Figures

1) Cepstrum example - time history ................................................. 11
2) Cepstrum example - power spectrum ............................................ 11
3) Cepstrum example - power cepstrum ........................................... 12
4) One fault pulse - time history ..................................................... 12
5) One fault pulse - spectrum .......................................................... 13
6) Sequence of fault pulses - time history ........................................ 13
7) Sequence of fault pulses - spectrum .............................................. 13
8) Modulation example - time history .............................................. 15
9) Modulation example - spectrum ................................................... 15
10) Demodulation example - time history .......................................... 17
11) Demodulation example - spectrum of raw data ............................... 17
12) Demodulation example - high pass filter .................................... 18
13) Demodulation example - spectrum of filtered data .......................... 18
14) Demodulation example - rectified data ....................................... 18
15) Demodulation example - low pass filter ..................................... 19
16) Demodulation example - spectrum of demodulated data .................. 19
17) Fault pulses with noise - time history ........................................ 20
18) Fault pulses with noise - spectrum .............................................. 20
19) Fault pulses with noise - demodulated time history ........................ 21
20) Fault pulses with noise - demodulated spectrum ............................. 21
21) Singular value decomposition - system model .................................. 22
22) SVD example - sine wave input ................................................. 25
23) Picture of rotor rig ..................................................................... 29
24) Picture of inner race fault ............................................................ 30
25) Picture of outer race fault ............................................................ 31
26) Picture of ball fault .................................................................... 32
27) Rotor rig instrumentation diagram .............................................. 35
<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>28)</td>
<td>Picture of instrumentation setup.</td>
<td>38</td>
</tr>
<tr>
<td>29)</td>
<td>Data acquisition equipment.</td>
<td>39</td>
</tr>
<tr>
<td>30)</td>
<td>Fault frequency program output.</td>
<td>41</td>
</tr>
<tr>
<td>31)</td>
<td>Spectrum analysis - baseline.</td>
<td>42</td>
</tr>
<tr>
<td>32)</td>
<td>Spectrum analysis - inner race fault.</td>
<td>43</td>
</tr>
<tr>
<td>33)</td>
<td>Spectrum analysis - outer race fault.</td>
<td>43</td>
</tr>
<tr>
<td>34)</td>
<td>Spectrum analysis - ball fault.</td>
<td>44</td>
</tr>
<tr>
<td>35)</td>
<td>Spectrum analysis - misalignment.</td>
<td>44</td>
</tr>
<tr>
<td>36)</td>
<td>Shock pulse analysis - baseline.</td>
<td>45</td>
</tr>
<tr>
<td>37)</td>
<td>Shock pulse analysis - inner race fault.</td>
<td>46</td>
</tr>
<tr>
<td>38)</td>
<td>Shock pulse analysis - outer race fault.</td>
<td>46</td>
</tr>
<tr>
<td>39)</td>
<td>Shock pulse analysis - ball fault.</td>
<td>47</td>
</tr>
<tr>
<td>40)</td>
<td>Shock pulse analysis - misalignment.</td>
<td>47</td>
</tr>
<tr>
<td>41)</td>
<td>Shock pulse analysis - peak vibration levels.</td>
<td>48</td>
</tr>
<tr>
<td>42)</td>
<td>Cepstrum analysis - baseline.</td>
<td>50</td>
</tr>
<tr>
<td>43)</td>
<td>Cepstrum analysis - inner race fault.</td>
<td>51</td>
</tr>
<tr>
<td>44)</td>
<td>Cepstrum analysis - outer race fault.</td>
<td>51</td>
</tr>
<tr>
<td>45)</td>
<td>Cepstrum analysis - ball fault.</td>
<td>52</td>
</tr>
<tr>
<td>46)</td>
<td>Cepstrum analysis - misalignment.</td>
<td>52</td>
</tr>
<tr>
<td>47)</td>
<td>Demodulation - baseline.</td>
<td>54</td>
</tr>
<tr>
<td>48)</td>
<td>Demodulation - inner race fault.</td>
<td>55</td>
</tr>
<tr>
<td>49)</td>
<td>Demodulation - outer race fault.</td>
<td>55</td>
</tr>
<tr>
<td>50)</td>
<td>Demodulation - ball fault.</td>
<td>56</td>
</tr>
<tr>
<td>51)</td>
<td>Demodulation - misalignment.</td>
<td>56</td>
</tr>
<tr>
<td>52)</td>
<td>SVD - inner race fault and baseline.</td>
<td>58</td>
</tr>
<tr>
<td>53)</td>
<td>SVD - outer race fault and baseline.</td>
<td>59</td>
</tr>
<tr>
<td>54)</td>
<td>SVD - ball fault and baseline.</td>
<td>59</td>
</tr>
<tr>
<td>55)</td>
<td>SVD - misalignment fault and baseline.</td>
<td>60</td>
</tr>
<tr>
<td>List of Tables</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>(1) Fault Test Configurations</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>(2) Test Runs for Each Configuration</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>(3) Data Acquisition Speeds</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>(4) Rotor Rig Instrumentation</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>(5) Signal Processing Summary</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>Nomenclature</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>$f_{sh}$</td>
<td>fundamental shaft frequency</td>
<td></td>
</tr>
<tr>
<td>BPFI</td>
<td>ball pass frequency of the inner race</td>
<td></td>
</tr>
<tr>
<td>BFBO</td>
<td>ball pass frequency of the outer race</td>
<td></td>
</tr>
<tr>
<td>BSF</td>
<td>ball spin frequency</td>
<td></td>
</tr>
<tr>
<td>FTF</td>
<td>fundamental train frequency (bearing cage)</td>
<td></td>
</tr>
<tr>
<td>Nb</td>
<td>number of balls or rollers</td>
<td></td>
</tr>
<tr>
<td>Bd</td>
<td>ball or roller diameter</td>
<td></td>
</tr>
<tr>
<td>Pd</td>
<td>pitch diameter</td>
<td></td>
</tr>
<tr>
<td>$\Phi$</td>
<td>contact angle</td>
<td></td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>x(t)</td>
<td>time history</td>
<td></td>
</tr>
<tr>
<td>x(f)</td>
<td>frequency spectrum</td>
<td></td>
</tr>
<tr>
<td>$C_x(t)$</td>
<td>power cepstrum</td>
<td></td>
</tr>
<tr>
<td>X(s)</td>
<td>Laplace transform of input function</td>
<td></td>
</tr>
<tr>
<td>F(s)</td>
<td>Laplace transform of forcing function</td>
<td></td>
</tr>
<tr>
<td>$[A]_r$</td>
<td>finite sequence of coefficient matrices</td>
<td></td>
</tr>
<tr>
<td>$[B]_r$</td>
<td>finite sequence of coefficient matrices</td>
<td></td>
</tr>
<tr>
<td>$s^r$</td>
<td>temporal operator in the Laplace domain</td>
<td></td>
</tr>
<tr>
<td>$\alpha_r$</td>
<td>$r^{th}$ autoregressive coefficient</td>
<td></td>
</tr>
<tr>
<td>2N</td>
<td>order of ARMA model</td>
<td></td>
</tr>
<tr>
<td>[Co]</td>
<td>coefficient matrix</td>
<td></td>
</tr>
<tr>
<td>SVD</td>
<td>singular value decomposition</td>
<td></td>
</tr>
<tr>
<td>[A]</td>
<td>matrix of time shifted response points</td>
<td></td>
</tr>
<tr>
<td>[U]</td>
<td>unitary matrix of left singular vectors</td>
<td></td>
</tr>
<tr>
<td>[V]</td>
<td>unitary matrix of right singular vectors</td>
<td></td>
</tr>
<tr>
<td>[S]</td>
<td>diagonal singular value matrix</td>
<td></td>
</tr>
<tr>
<td>$f_s$</td>
<td>sampling frequency</td>
<td></td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

This thesis summarizes a work effort to evaluate several signal processing algorithms that could be sensitive to various rotor dynamic faults. The methods were evaluated for their ability to detect the presence of the fault as well as identify the location of the fault. The faults included: an inner race bearing fault, an outer race bearing fault, a rolling element bearing fault, rotor unbalance, and shaft misalignment. Spectrum analysis, which has been in use for many years for fault detection, was used as a baseline technique. Newer techniques, such as signal demodulation and singular value decomposition, were compared for their sensitivity to detect the rotor dynamic faults. All the signal processing algorithms were evaluated by processing the same vibration data acquired on a controlled rotor rig that had known faults induced into the system.

There are many reasons to be interested in fault detection of rotor dynamic systems. If a fault can be detected in the early stages, corrective action may be taken before any further secondary damage or catastrophic failure will occur. The benefits from health monitoring of machinery also include the following:

- Production losses are reduced.
- Machinery efficiency, reliability, availability, and longevity are improved.
- Maintenance costs are reduced (overtime payments for labor are reduced, spare parts and stocked inventory costs are decreased, fuel costs are reduced).
- Work load planning is improved.
- Safety and environmental standards are improved.

Investigating the health of the rotating system can be divided into three categories. The first level of monitoring is detecting the difference between a healthy system and an unhealthy system. The second level attempts to identify the type and location of the
fault. The third level of fault detection is involved in determining the time to catastrophic failure. For example, a rocket engine might start to exhibit a unique vibration, which can be identified as the start of an unhealthy condition. The next critical task is to determine the source of vibration, for example the High Pressure Oxygen Turbo Pump, No 3 Bearing, outer race. The last and the hardest part of fault detection is determining how far the fault has progressed, and when failure will occur. Many times, the vibration environment can provide information on the properties of the fault. The key is extracting the fault information from a very complex and information rich vibration signal.

1.1 Types of Faults

An important property of fault detection is the identification of the fault as early as possible. The longer the time span over which a developing fault can be observed, the greater the accuracy of the predicted failure time. For this reason it is important to understand the fault mechanisms. A complete literature search was made on rotor dynamic faults, health monitoring, and signal processing [1-28].

Unbalance is a common malfunction in rotating machines. When a rotor is unbalanced, the rotor mass centerline does not coincide with the axis of rotation. During rotation the unbalance will generate an inertial centrifugal force which rotates at the rotor rotational frequency. The unbalance excitation force is determined from the response measured at the rotational speed [9].

Misalignment faults will cause a constant radial force that will push the rotor to one side. The joint where the misalignment is concentrated can generate a 2X rotational frequency reaction force. This is because of non uniform stiffness to bending at the misaligned joint. For one complete shaft rotation, each segment of the misaligned joint will undergo both compression and tension. The non uniform segment will generate a reaction force for both the compression and tension parts of the cycle, thus introducing
the 2X force [1].

Rolling element bearings generate characteristic vibration signatures in several ways. It is important to note that even a healthy fault-free bearing can introduce vibration components at the characteristic fault frequencies. There can be variable flexural compliance of the bearing components. This means that the bearing components will deform non-uniformly and will produce the fault frequencies. Also there can exist geometrical imperfections that will generate fault frequencies from a new “good” bearing. The key to determining the start of a faulty condition is to identify the changes in the vibration signal that the faults will produce.

A rolling element bearing has a finite life and will eventually fail due to fatigue, even if operated under ideal conditions. The life limits can be calculated and are known for most bearings [20]. Unfortunately, many bearings will fail before reaching their design life. Most premature bearing failures can be attributed to one or more of the following causes:

1. Excessive loading.
2. Insufficient lubrication.
3. External contamination.
4. Improper installation.
5. Incorrect design or load rating.
6. Exposure to vibration while not rotating (false brinelling).

The rolling element bearing failure progression can be classified into three stages. In the pre-failure stage hairline cracks or microscopic spalls are formed that are not visible to the human eye. The bearing usually has a significant amount of operating life remaining, and it is not economical to replace the bearing at this time. In the second stage, the failure stage, the bearing develops spalls that are visible to the human eye. The bearing can produce an audible sound. This is the bearing fault stage that was investigated with the rotor rig. At this time it is very important that a health
monitoring system detects the fault before it progresses to the third fault stage, the catastrophic failure stage. When the bearing enters the catastrophic stage, rapid failure is imminent. Audible noise produced from the bearing significantly increases. The bearing temperature will increase until the bearing overheats. Loss of bearing support and rotor to stator rubs could lead to extensive secondary damage [2].

If a roller or a ball has a defect such as a pit, each revolution will result in a impact which is transmitted to the bearing housing. The fundamental frequency of these impacts is called the ball spin frequency (BSF). If the bearing inner race has a defect, then each ball will produce a shock as it passes giving rise to a fundamental vibration frequency called the ballpass frequency, inner race (BPFI). Likewise a fault on the bearing outer race will produce a frequency at the ballpass frequency, outer race (BPFO). A fault in the bearing cage will produce a frequency at the fundamental train frequency (FTF) [20].

The faults tested in this thesis are only used as a tool to evaluate the different signal processing techniques. The faults were chosen to be representative of common faults. There are a number of other rotor dynamic faults that were not tested, but still would be applicable to the fault detection algorithms. These other types of faults include rotor to stator rubbing, fluid induced instabilities, loose stationary and rotating parts, cracked shafts, gear faults, and bearing cage faults.

1.2 Signal Processing Techniques

The different fault detection techniques investigated in this thesis attempt to enhance the known vibration characteristics from the fault. The narrow band spectrum analysis is used to identify the fundamental fault frequencies previously outlined. The shock pulse method attempts to quantify the high frequency short impulse nature of the fault. The cepstrum analysis will evaluate the harmonic content created by the short duration of the fault impulses. The demodulation technique will translate down the fault frequencies that are present in the high frequency spectrum. And lastly the singular
value decomposition will identify the added sources due to the fault frequencies.

Testing from the rotor rig showed that some of the signal processing methods were better at identifying the fault verses no fault condition, where other methods were found to be better at determining the location of the fault. However, for this test, no attempt was made to address the problem of determining the severity of the fault. The rotor rig had artificial faults introduced that were not at different stages in the fault. A baseline no fault condition was tested and compared to three different types of bearing faults and a misalignment fault. Therefore, the progression of one type of fault could not be tracked.

The vibration data was processed using MATLAB™ [28] on a personal computer. MATLAB is a powerful software package for engineering numeric computations. The "programs" or script files for each of the processes investigated in this thesis are found in Appendix A.

1.3 Test Rig

A rotor rig was constructed to acquire real vibration data from a controlled fault condition. Bearing faults were introduced into disassembled ball bearings. The faulty bearings were re-assembled by the bearing manufacturer. Each faulty bearing was tested individually and was compared to a no fault condition. The static axial load and unbalance level was varied for each fault condition. The rotor rig could be operated at speeds from 0 to 10,000 rpm. A misalignment could be introduced in the system by moving the motor relative to the main shaft.

The vibration data was measured by using accelerometers mounted at various positions on the rotor rig. There are many other methods to measure the vibration that were not investigated in this thesis. Some of the other methods include: placing a proximity transducer on the bearing outer race [21] and using a fiber optic bearing deflectometer [27].
1.4 Scope of Thesis

The objective of the thesis is to evaluate several different signal processing algorithms for the ability to detect typical rotor dynamic faults.

Chapter 2 describes the theory behind the different signal processing techniques that were investigated. Examples, using analytical data, are included to help show how some of the algorithms work.

A description of the test rig that was used to acquire data for the rotor faults is given in Chapter 3. This includes the details of the different faults, and a description of the data that was recorded.

Chapter 4 contains the results from processing the rotor rig test data with the fault detection algorithms.

In Chapter 5 a summary and conclusion of the results, which includes a comparison between the different signal analysis methods, takes place.
2.0 **Signal Processing Techniques Theory**

Five (5) different techniques were investigated in this thesis. The first two, spectrum analysis and shock pulse energy are the most common techniques currently used for fault detection. The last three methods: cepstrum analysis, signal demodulation, and singular value decomposition are newer methods that were evaluated.

2.1 Spectrum Analysis

Spectrum analysis of vibration data is simply the method where the time domain data is Fourier transformed into the frequency domain. Each fault is known to create a vibrational component at the fault impact frequency. The spectrum is plotted and inspected for the magnitude at the fault frequency. The fault frequencies are calculated from the kinematics of the bearing, which are determined by the geometry of the bearing [26]. The equations for the three bearing fault frequencies tested on the rotor rig are as follows:

\[
\text{Inner Race Fault Frequency} = \frac{(Nb/2)(f_{sh})(1+(Bd/Pd) \cos \phi)}{(BPFI)}
\]  

(1)

\[
\text{Outer Race Fault Frequency} = \frac{(Nb/2)(f_{sh})(1-(Bd/Pd) \cos \phi)}{(BPFO)}
\]  

(2)

\[
\text{Ball Fault Frequency} = \frac{((Pd/2Bd)(f_{sh})(1-(Bd/Pd)^2 \cos^2 \phi)}{(BSF)}
\]  

(3)

where: Nb = Number of balls or rollers.

\( f_{sh} \) = Fundamental shaft frequency.

Bd = Ball or roller diameter.

Pd = Pitch diameter.

\( \phi \) = Contact angle.
A misalignment fault will usually generate a 2X shaft speed component which can be identified in the frequency spectrum. The 2X frequency is generated because of an asymmetric stiffness at the misaligned joint. As the shaft rotates, the misaligned segment will undergo first compression then tension for each rotation. This 2X unequal deformation causes a 2X reaction at the bearings and mounts, which can be measured in the vibration spectrum [1].

To implement the narrow band spectrum analysis, the time history data should be windowed with a hanning window to reduce leakage errors associated with the Fast Fourier Transform (FFT). Next the FFT of the data block is computed. The best units to view bearing fault data is in velocity units. Acceleration data will exaggerate the high frequency components unnecessarily while displacement data will attenuate the high frequency characteristics. Thus, the standard spectrum for fault detection has units of velocity. To view the smaller vibration components on the same plot as the larger components, a log amplitude scale should be used for the vibration amplitude. The best way to view log data is to convert the spectrum data into dB. The equation used to convert the velocity signal into dB is as follows:

\[
\text{velocity dB} = 20 \log_{10}(x(f)/\text{ref})
\]  

where: \(x(f)\) = velocity spectrum  
\(\text{ref} = \text{reference (1E-8 m/sec rms)}\)

This way the y axis is linear and much easier to interpolate values. The frequency axis should be linear. If a log frequency axis is used, the high frequency data of interest becomes hard to evaluate because it is compressed.

An advantage to the spectrum analysis method is that the algorithm is very easy to use and interpret the results. If a fault frequency is detected, then the location and type
of fault is immediately known. The data can be analyzed fairly rapidly depending on the resolution required. The spectrum analysis method will identify the unbalance and misalignment as well as all the different types of bearing faults.

There are several disadvantages to the spectrum analysis method. The main problem is that it is not very sensitive to bearing faults when they are in the early stages. Many times the fundamental bearing fault frequencies are very low in amplitude and will become lost in the background noise. The high frequency fault components can become hard to identify if there are any speed fluctuations of the rotor. The speed fluctuations can cause smearing in the frequency domain if the speed variations become too large. Also, if there is any slip in the bearing, the fault frequencies do not appear where they are calculated. For a bearing that is experiencing a great deal of slip, the fundamental train frequency (or cage pass frequency) is needed before any fault frequency can be identified.

2.2 Shock Pulse Method

The shock pulse method measures the high frequency components that are a result of the bearing faults. As a ball or roller passes over a faulty area in the bearing, the impact sends a small shock pulse through the system. This pulse excites all frequencies in the system. An accelerometer, which is basically made of a mass that reacts on a piezo-electric crystal, has a natural frequency where the mass and crystal will resonate. This sensor natural frequency is usually very high and is above the typical frequency range of interest. However, this frequency can be useful when looking for the shock pulse from a faulty bearing, since the fault will excite the natural frequency of the sensor.

This method is fairly simple to implement for fault detection. The spectrum of the data is viewed in the high frequency range (at the sensors natural frequency). The amplitude of these components in the spectrum can be recorded and trended. As the
fault progresses, the amplitudes will increase.

The advantage of the shock pulse method is that it is very easy to implement with standard data acquisition equipment. The increasing levels at the sensors natural frequency will give advance warning that some impulsive forcing function is present in the system. This method is useful for rolling element bearing types of faults.

A disadvantage of this method is that it will not identify the source of the fault. The source could be any type of bearing fault such as inner race or outer race, and it could be originating from any number of different bearings in the system. The shock pulse method will not detect low frequency faults such as unbalance or misalignment.

2.3 Cepstrum Analysis

Cepstrum analysis is basically the Fourier transform of a frequency spectrum. This analysis technique is used to detect echo and harmonic patterns in a spectrum. Historically the cepstrum analysis was utilized for acoustic data interpretation. Overlapping harmonic data is difficult to analyze in the frequency domain. When the cepstrum is computed, peaks in the cepstrum domain show which frequencies are repeated in the spectrum.

The cepstrum technique used for the rotor rig data was the power cepstrum. This is a “one way” analysis that is implemented by taking the Fourier transform of the log power spectrum. Once the power cepstrum is calculated, it can not be reversed like the complex cepstrum which retains the real and imaginary components. The complex cepstrum is used when filtering (liftering) and inverse Fourier transforming is performed to reconstruct the spectrum with individual sources removed. The equation for the power cepstrum is as follows:

\[
C_x(t) = \mathcal{F}\{\log[|\mathcal{F}\{x(t)\}|^2]\}
\]

where \(\mathcal{F}\) = the fast Fourier transform (FFT).
An example of how the cepstrum technique works is illustrated here. First a signal is constructed by adding 7 sine waves separated by 20 Hz as is shown in Figure 1 and Figure 2.

The power cepstrum, shown in Figure 3, clearly shows the spacing between the seven frequencies by the large peak at 0.05 seconds. The x axis is called quefrenty and has the units of 1/Hz or time. The y axis is used for a relative amplitude and has no real physical meaning. This first peak in the cepstrum plot corresponds to the 20 Hz spacing in the spectrum (1/0.05 = 20).

A rolling element bearing fault will create a spectrum that has many harmonics of the fault frequency. This is because the fault will create a series of repetitive impulses. Consider the dynamics as one ball rolls over a faulty area in the outer race. The discontinuity will create a short impulse that will excite the natural frequency of the bearing structure. This will be a very high frequency with a lot of damping.
Analytical data was created to simulate this impulse response. Figure 4 shows the high frequency response as the one ball excites the bearing. Figure 5 shows the spectrum of this one pulse where the natural frequency is at 10,000 Hz.

When the shaft rotates, a sequence of impacts will excite the structure as can be seen in Figure 6. The Fourier transform of this signal will result in a spectrum shown in Figure 7. Notice how the time between impacts (t) in Figure 6 results in the harmonic spacing of the fault frequency components in Figure 7.

The cepstrum can be viewed as a deconvolution process. The fault response is convolved in the time domain with a sequence of delta functions corresponding to the fault impact rate. This convolution in the time domain results in a multiplication in the frequency domain. The spectrum of the delta functions is multiplied by the spectrum of the fault pulse which results in the discrete line spectrum seen in Figure 7. The
Figure 5: One Fault Pulse Spectrum

Figure 6: Sequence of Fault Pulses Time History

Figure 7: Sequence of Fault Pulses Spectrum

\[ \Delta f = \frac{1}{t} \]
The cepstrum will deconvolve this process to show the sequence of the delta functions.

The cepstrum analysis algorithm is especially good at measuring the harmonic content of bearing faults. Many times the fault frequencies can not be identified in the spectrum because they are at such a low level when compared to the background noise. The cepstrum analysis will enhance the fault frequencies and improve the ability to identify the exact location of the fault.

A disadvantage of the cepstrum analysis technique is that the output needs to be interpreted correctly to maximize the usefulness of the algorithm. Harmonics in the frequency domain (called rahmonics) are a common side effect of the technique. Another disadvantage of this algorithm is that it is not effective in finding non harmonic faults such as unbalance.

2.4 Demodulation of Fault Information

Amplitude modulation can be described as the multiplication of one signal by another in the time domain. This process will give rise to new frequencies in the frequency domain that are not present in either of the signals involved in the modulation. The new frequencies are called sidebands. The process of modulation is very different than addition or superposition of signals where the Fourier transform is used to separate the signals. Instead, the modulation is a multiplication process that is not well suited for the standard Fourier transform techniques.

An example of signals that are modulated was created using analytical data. Figure 8 is the resulting wave form from a low frequency sin wave at 150 Hz amplitude modulating a high frequency sin wave at 1200 Hz. The low frequency, or modulating signal causes the amplitude of the high frequency signal to fluctuate at the rate of the modulating signal.

The frequency spectrum of the modulated wave form can be seen in Figure 9. The high frequency carrier is shown along with two new side band frequencies. Notice that
there is no component at 150 Hz! The spacing between the carrier and the sidebands is equal to the original modulating frequency of 150 Hz.

A trigonometric identity can be used to describe this modulation effect. Consider a fault frequency \( w_1 \) (that has a DC component) and a carrier frequency \( w_2 \). The multiplication process can be represented by an addition process as follows:

\[
(cos(w_1)t + DC)(cos(w_2)t) = 1/2cos(w_1 + w_2)t + 1/2cos(w_1 - w_2)t + (DC)cos(w_2)t \tag{6}
\]

It can be seen that frequencies of \( w_1 + w_2 \), \( w_1 - w_2 \), and \( w_2 \) are the only frequencies that appear in the spectrum when the multiplication process is modeled as an addition process.

Bearing faults can become modulated because the fault rotates into and out of the load zone. The load zone could result from the gravitational force on the rotor, or could be a misalignment force on the bearing. For every rotation
of the shaft the amplitude of the fault frequency will modulate in amplitude. For the faults that are located on a rotating component (moving reference frame) such as the inner race fault and the ball fault, this modulation mechanism is easy to verify. The outer race fault will always develop in the load zone, so the amplitude will not vary with shaft rotation.

The process of demodulation involves 4 steps. First the signal is high pass filtered. This step eliminates any low frequency data that could contaminate the demodulated data after it is shifted down in frequency. The cutoff frequency for the filtering should be just above the frequency range of interest. For example if you are interested in the ball pass spin frequency (BSF) and 2X BSF, then you would select the filter cutoff to be about 2.5X BPF. It is important that a structural mode at high frequency (like the bearing modes) be included in the high pass frequency range.

The second step in signal demodulation is to rectify the data. This is simply setting all the negative components to zero. The effect of rectifying is to enhance the low frequency modulation on the signal.

The third step involves low pass filtering of the previously rectified data. This insures that all the data that is being viewed is the low frequency modulated waveform. All the discontinuities that are a result of the rectifying process are eliminated.

The last step is simply to FFT the filtered and rectified data. The fault frequencies that were modulated to high frequencies will be enhanced and self evident. All the low frequency noise in the original signal will have been eliminated so the fault frequencies can be clearly identified.

An example of this process used on analytical data is shown below. A fault frequency was constructed at 150 Hz that was modulated on a carrier frequency at 1200 Hz. A small DC component was included on both signals to identify their position in the spectrum. Random noise was added to the signal to simulate real vibration data. The time domain signal and spectrum can be seen in Figures 10 and 11.
Next the signal was high pass filtered at 1000 Hz. A plot of the digital filter response and the resultant spectrum after convolution with the filter is shown in Figure 12 and Figure 13.

The signal was then rectified which involved zeroing out the negative components. This rectified signal was low pass filtered at 1000 Hz. The rectified data and filter response are shown in Figure 14 and Figure 15 respectively.

The process outlined above was performed entirely in the time domain. The spectrums were plotted for demonstration of the frequency content of the intermediate steps. The only time an FFT is performed on the data is at the last step when the demodulated frequencies are identified. A hanning window was applied to the data before the FFT was performed. A plot of the resultant spectrum is shown in Figure 16. Note how the fault frequency at 150 Hz is now very apparent when compared to the first spectrum shown in Figure 11.

The demodulation process has the effect of extracting the envelope information
Figure 12: Demodulation Example - High Pass Filter

Figure 13: Demodulation Example - Spectrum of Filtered Data

Figure 14: Demodulation Example - Rectified Data
from the high frequency periodic impulses that are present in outer, inner, and ball faults. The short duration impulses will cause the high frequency structural modes of the bearing to be excited. These pulses have very little energy content at the fundamental frequency. If the ringing impulses are passed through a full-wave rectifier and then low-pass filter to smooth the shape, then the demodulated pulse lasts longer and therefore the energy content at the fundamental and lower harmonic frequencies is increased. An example of this effect is shown with analytical data that was created to simulate the dynamics of a fault. The sequences of shock pulses shown in Figure 6 in Section 2.3 was used for this example. Random noise was added to better simulate actual vibration data as can be seen in Figure 17. The time between fault pulses is 0.0039 seconds which corresponds to a fault frequency of 256 Hz. The spectrum of this data shown in Figure 18 shows the difficulty finding this frequency with the added noise. When the signal is
processed with the
demodulation technique the
envelope of the pulses is
increased and smoothed as
can be seen in Figure 19.
The spectrum of this data
shown in Figure 20 clearly
shows the fault frequency at
256 Hz.

The demodulation
technique is very good at
finding bearing faults. The
exact location of the fault is
determined by the fault
frequencies that are
enhanced. Since bearing
faults in the early stages of
development have very low
energy at the fundamental
frequency, the
demodulation technique can
be useful in identifying the faults at this stage.

Some disadvantages of this method are that it requires extra computations and some
judgement by the user. The high pass and low pass filters need to be set so that the
information desired is included in the bandwidth. This method is not effective for low
frequency rotor faults such as unbalance and misalignment.
2.5 Singular Value Decomposition

Singular Value Decomposition (SVD) is used to indicate how many significant components (contributions) are present in a waveform. A significant component can be defined by a 1/rev excitation response, a response to a natural frequency, or any type of bearing fault. Any deterministic vibration source will create a component that can be estimated by the Singular Value Decomposition technique.

SVD is used extensively in modal analysis for determining the order (DOF) of the model being represented by the vibration test data. In many ways, the mathematical tools used for this task in modal analysis are the same for determining the number of
significant inputs in a forced (rotor dynamic) response. A brief description of the complex exponential method of modal parameter estimation utilizing impulse response functions is needed to justify the formation of the matrix which is used in the SVD routine. More information on the modal analysis application can be found in [10, 11].

The response data measured from the rotor dynamic system can be viewed as a linear combination of damped complex exponential functions. Some of these functions are the response to natural frequencies that are being excited by the system. Other functions (and the functions of interest here) are the system’s response to forced excitation. These response components have a frequency corresponding to the forcing frequency and can be viewed as having zero damping because there is no decay of this component. Figure 21 shows this combination of responses.

Any impulse response function can be analyzed to determine the frequency, damping, and amplitude of the measured modes of the system. This can be shown by using an ARMA model that represents a set of finite difference equations specific to the time domain data. A high order ARMA model in the Laplace domain (matrix polynomial model) can be described as:

\[
\sum_{r=0}^{n} ([A]_r s^r) X(s) = \sum_{r=0}^{m} ([B]_r s^r) F(s)
\]  

(7)

where: \([A]_r\) is a finite sequence of coefficient matrices
\([B]_r\) is a finite sequence of coefficient matrices
\(s^r\) is a temporal operator in the Laplace domain
\(X(s)\) is the Laplace transform of input functions
\(F(s)\) is the Laplace transform of the forcing functions
For the homogeneous case (used for determining frequency and damping) the right hand side can be set to zero. The ARMA model for a single response point in the time domain reduces to the scalar equation:

\[ \sum_{r=0}^{2N} [ \alpha_r x_{(m+r)}(t)] = 0 \]  \hspace{1cm} (8)

where: \( \alpha_r \) is the \( r \)th Autoregressive Coefficient
\( x_{m+r}(t) \) is the \( m+r \) time sample of the response
\( m \) is the index for starting time sample
\( 2N \) is the order of the ARMA model

If the forced response components are viewed as pseudo poles of the system, then the ARMA model representation becomes simplified. The B terms which describe the forces acting on the system can be eliminated. These terms are lumped as other A terms which permit the equation to be solved as a homogeneous equation. As mentioned before, these pseudo poles will have a frequency at the forcing frequency and zero damping (poles on the unit circle in the Z domain).

Additional equations can be generated by taking different parts of the same response function. Since the coefficients for the difference equation are not unique, the \( \alpha \)'s can be determined by assuming that one of the \( \alpha \) terms is equal to one. The standard approach is to assume that \( \alpha_{2N} \) is equal to one. The resulting equations become:

\[ \sum_{r=0}^{2N-1} [ \alpha_r x_{(m+r)}] = -x_{(m+2N)} \]  \hspace{1cm} (9)

When the starting times are taken sequentially, a set of Toeplitz equations are generated:
\[
\begin{bmatrix}
X_0 & X_1 & X_2 & \cdots & X_{2N-1} \\
X_1 & X_2 & X_3 & \cdots & X_{2N} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
X_{2N-1} & X_{2N} & X_{2N+1} & \cdots & X_{4N-2}
\end{bmatrix}
\begin{bmatrix}
\alpha_0 \\
\alpha_1 \\
\alpha_2 \\
\alpha_{2N-1}
\end{bmatrix}
= \begin{bmatrix}
X_{2N} \\
X_{2N+1} \\
\vdots \\
X_{4N-1}
\end{bmatrix}
\]

It is possible to generate as many equations as necessary (2N or more) from a single time history. The \( \alpha \) terms can now be determined. The roots of the associated polynomial equation yield the eigenvalues of the system. A standard method for solving for these roots involves taking the Z transform of the ARMA model. For more information on the Z transform, see reference [10].

A key step in the above process is knowing how many coefficients to use. If too many coefficients are used then there are more roots of the characteristic equation than there are actual modes of the system. This results in computational modes that are not real to the system. If not enough coefficients are used, then some of the poles or pseudo poles are missed. The method of determining how many coefficients to use involves using the SVD algorithm.

The Toeplitz equations (Equation 8) can be written in matrix form as:

\[
\{ \text{Co} \} \{ \alpha \} = \{ \text{D} \}
\]

where \( \{ \text{Co} \} \) is the coefficient matrix

\( \{ \text{D} \} \) is the constant vector

The Singular Value Decomposition can be performed on the coefficient matrix \( \{ \text{Co} \} \) to estimate the number of independent characteristics contained in the data.

The singular value decomposition theorem states that every matrix \( \mathbf{A} \) (mxn) can be decomposed as:
\[ A = USV^H \]  \hspace{1cm} (11)

Where \( U \) is \((m \times m)\) and unitary, \( V \) is \((n \times n)\) and unitary, and \( S \) is \((m \times n)\) and "diagonal" in that each term \( S_{ij} = 0 \) unless \( i = j \), in which case \( S_{ii} = s_i \) with \( s_i \) real and nonnegative. The columns of \( U \) are called the left singular vectors of \( A \), and they form a unitary basis for the column space of \( A \). The rows of \( V^H \) are called the right singular vectors of \( A \) and they form a unitary basis for the row space of \( A \). The diagonal elements of \( S \) are called the singular values of \( A \), and the number of nonzero singular values is equal to the rank of matrix \( A \).

For measured data, due to random errors and inconsistencies in the data, the singular values will not become zero but will be very small when the rank is exceeded. Therefore, the rate of change of the singular values is used as an indicator rather than the absolute values. To demonstrate how the SVD routine works, analytical data that composed of 2 sine wave components and 4 sine wave components was evaluated. Figure 22 shown here illustrates how the magnitude of the singular values change for 2 sine waves and for 4 sine waves (the inflection point is at 4 and 8 respectively because each sine wave has a real and imaginary component).

The SVD routine was used on the rotor rig data to determine if the added vibration sources from the faults could be detected. The time history data was formed into matrices shown in Equation 11 above. The Singular Values were calculated and plotted.
using MATLAB.

The main advantage of using the SVD routine for fault detection is that this routine will work for all the faults. Any fault that will introduce an added vibration response will show up in the singular values. This includes the low frequency faults such as unbalance and misalignment as well as the high frequency bearing faults. This method could be used as an advanced warning that some extra vibration sources are present, and indicate a detailed study to find them with another technique such as cepstrum or signal demodulation.

The disadvantage of using this routine is that the exact location of the source cannot be identified. Also, the routine could be susceptible to noise on the signal. Random noise will always increase the rank of the matrix. This will increase the amplitude of all singular values and might make it harder to find the inflection points where the singular values change amplitude. Also, a disadvantage of this routine is that initial order size (matrix size) needs to be chosen. This order size could be low for a simple system with a few sources, or could be large for a very complex system.

2.6 Other Signal Processing Techniques

There are many signal processing techniques that were not used on this data. However, there are two methods that were reviewed but not utilized. These two methods are used primarily on rotor data that does not have repeatable speeds or has transient conditions. Because the rotor rig data had a very good speed control, it was decided not to analyze the data with these methods. The two methods are mask spectrum analysis and Prony residue calculations.

Mask spectrum analysis is a frequency domain technique that is used to detect a faulty rotor dynamic condition. The process involves taking a spectrum from a healthy condition and making a envelope or template around this spectrum. A bandwidth is created around each major peak so that subsequent data that is not acquired at the exact
same rotor speed will have room to fall within this template. When a fault occurs the fault frequency will peak above the health template and trigger an alarm. This method is good for data that is trended over a long period of time where the rotor speeds may vary. It was decided to not be used for this test because the rotor speeds were easily controlled to be very repeatable. This made the fault frequencies easy to identify by the narrow band spectrum analysis technique.

The Prony residue technique is a parameter estimation routine that can be used to track sin waves that change frequency rapidly. This algorithm is very similar to the ARMA model that was described in the SVD section. Difference equations are solved for the coefficients which roots can be solved for the frequencies. The advantage of this technique is that only 4 data points are needed for each root of the system. This is very attractive when compared to FFT methods that require a large number of points to get good resolution. The problem with this method is that it is a parameter estimation routine that requires the order of model to be known. This routine also is sensitive to noise on the data. This method was not used on the rotor rig data because the data analyzed was steady state response rather than the transient response. On some test in the future it would be interesting to track a fault frequency with this method during a rotor transient.
3.0 Rotor Fault Testing

3.1 Rotor Test Rig

A rotor rig was built to investigate rotor dynamic faults. The rig featured a 1/10 HP electric motor mounted on a 31 x 6 inch steel base. This motor was connected to the rotor shaft with a rubber flexible coupling. Many parts were used from the Bently Nevada Rotor Kit (RK-3). A specially made steel shaft was supported by two Barden angular contact thrust bearings. The thrust bearings were held in aluminum supports that were bolted to the steel base. The bearing supports could be loaded axially by tightening connecting rods between them. A picture of the test rig and setup can be seen in Figure 23.

The rotor dynamic faults included an inner race bearing fault, an outer race bearing fault, a ball fault, and a misalignment fault. The unbalance and bearing load were varied for each fault. The bearing faults were introduced in a controlled and systematic procedure. Three disassembled bearings were received from the bearing manufacturer. A inner race fault was introduced by placing a small scratch in the first bearing inner race rolling surface as can be seen in Figure 24. The size of the scratch corresponds to the failure stage as outlined in reference [21]. The outer race fault was scratched into the second disassembled bearing in a similar fashion. Figure 25 shows a magnified photograph of this fault. The last fault bearing had one ball scratched as seen in Figure 26. The faulty ball on the left is shown with a good ball on the right.

The bearings were returned to the manufacturer to be re-assembled. This insured that no extra faults were introduced from an improper assembly process. A fourth bearing with no faults (as received) was used to test the baseline condition. The fault test bearings were placed in the aft bearing support which was the support farthest from the motor. The bearing in the forward support was not disturbed for all the testing. The bearings were lubricated with standard light weight motor oil (10W-30).

The rotor rig included features for two plane unbalance and shaft misalignment.
Figure 23: Rotor Rig for Fault Testing
The unbalance for the fault testing used 1.2 gram inches of single plane unbalance at the mass closest to the faulty bearing. The misalignment fault was introduced by shifting the motor relative to the shaft on the steel base. The resultant misalignment angle at the rubber flexible joint was 5 degrees. The rotor rig was designed to have the first critical speed in the operating speed range. This critical was at 6100 rpm with no load and at 6400 rpm with axial load.

3.2 Test Sequence

The test sequence involved four runs for each of the fault configurations. The different runs varied the load and unbalance of the rotor rig. The tests performed are summarized in Table 1.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Baseline</td>
<td>balance</td>
<td>unbalance</td>
<td>load + bal</td>
<td>load + unbal</td>
</tr>
<tr>
<td>2) Inner Race fault</td>
<td>balance</td>
<td>unbalance</td>
<td>load + bal</td>
<td>load + unbal</td>
</tr>
<tr>
<td>3) Outer Race fault</td>
<td>balance</td>
<td>unbalance</td>
<td>load + bal</td>
<td>load + unbal</td>
</tr>
<tr>
<td>4) Ball Fault fault</td>
<td>balance</td>
<td>unbalance</td>
<td>load + bal</td>
<td>load + unbal</td>
</tr>
<tr>
<td>5) Misalignment</td>
<td>balance</td>
<td>unbalance</td>
<td>load + bal</td>
<td>load + unbal</td>
</tr>
</tbody>
</table>

The bearings were loaded in the axial direction by tightening four rods that connected both bearing supports. The rods were instrumented with strain gages to measure the load. This enabled a repeatable load to be placed on the bearings for each fault configuration. When the four rods were fully loaded the bearing was under a 300 lb axial load.

For each of the 20 runs above there were 4 separate tests where the data acquisition parameters varied. The data acquisition test points are listed in Table 2.
Table 2: Tests for each configuration

<table>
<thead>
<tr>
<th>Test 1</th>
<th>Steady state data acquired at 4 speed points (see table 3) with a low sampling frequency ( f_s = 4096 ) Hz. 14 tracks of data were recorded.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 2</td>
<td>Steady state data acquired at 4 speed points (see table 3) with a high sampling frequency ( f_s = 32768 ) Hz. 4 tracks of data were recorded.</td>
</tr>
<tr>
<td>Test 3</td>
<td>Transient data acquired from 100 to 10000 rpm. 14 tracks of data were recorded with a low sampling frequency ( f_s = 4096 ) Hz.</td>
</tr>
<tr>
<td>Test 4</td>
<td>Steady state data acquired on 1 sensor for shock pulse information at 5000 RPM. Very high sampling frequency ( f_s = 50000 ) Hz.</td>
</tr>
</tbody>
</table>

The speed points where the steady state data was acquired were chosen so to have a low speed point, a point below the shaft critical, a point at the rotor critical, and a point above the shaft critical. The steady state data points are listed in Table 3.

<table>
<thead>
<tr>
<th>Data point 1</th>
<th>1000 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data point 2</td>
<td>5000 rpm</td>
</tr>
<tr>
<td>Data point 3</td>
<td>6100 or 6400 rpm</td>
</tr>
<tr>
<td>Data point 4</td>
<td>10000 rpm</td>
</tr>
</tbody>
</table>

Table 3: Data Acquisition Speeds

5.3 Data Acquisition Hardware and Method

The rotor rig was instrumented with 14 accelerometers arranged at locations corresponding to standard rotor monitoring positions. The bearing support vibrations were measured in the radial, tangential, and axial positions. The base and the motor were also instrumented. A sketch of the rotor rig with the sensor locations marked is shown in Figure 27. A complete list of the sensors is shown in Table 4.

The vibration data was measured using PCB accelerometers type 203H. A PCB Flexcell type 336-805 was used for the shock pulse data because the natural frequency
Figure 27: Rotor Rig Instrumentation Diagram
(not to scale)

Side View:

Top View:

End View:

KEY
1 - FWD BEARING SUPPORT,
   VERTICAL, AXIAL, LATERAL
2 - AFT BEARING SUPPORT,
   VERTICAL, AXIAL, LATERAL
3 - BASE, VERTICAL
4 - FWD BRG SUPPORT, LATERAL
5 - AFT BRG SUPPORT, LATERAL
6 - MOTOR, VERTICAL
7 - MOTOR, LATERAL
8 - AFT BRG SUPPORT, LATERAL
   (SHOCK TEST)
Table 4: Rotor Rig Instrumentation List

### SENSOR LIST FOR LOW SAMPLING RATE

<table>
<thead>
<tr>
<th>DATA TRACK</th>
<th>SENSOR CODE</th>
<th>SENSOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N1</td>
<td>Rotor Rig Speed</td>
</tr>
<tr>
<td>2</td>
<td>FV</td>
<td>Front Bearing Support Vertical</td>
</tr>
<tr>
<td>3</td>
<td>FT</td>
<td>Front Bearing Support Tangential</td>
</tr>
<tr>
<td>4</td>
<td>FA</td>
<td>Front Bearing Support Axial</td>
</tr>
<tr>
<td>5</td>
<td>AV</td>
<td>Aft Bearing Support Vertical</td>
</tr>
<tr>
<td>6</td>
<td>AT</td>
<td>Aft Bearing Support Tangential</td>
</tr>
<tr>
<td>7</td>
<td>AA</td>
<td>Aft Bearing Support Axial</td>
</tr>
<tr>
<td>8</td>
<td>BV</td>
<td>Base Vertical</td>
</tr>
<tr>
<td>9</td>
<td>FL</td>
<td>Forward Bearing Support Lateral</td>
</tr>
<tr>
<td>10</td>
<td>AL</td>
<td>Aft Bearing Support Lateral</td>
</tr>
<tr>
<td>11</td>
<td>MV</td>
<td>Motor Vertical</td>
</tr>
<tr>
<td>12</td>
<td>ML</td>
<td>Motor Lateral</td>
</tr>
</tbody>
</table>

### SENSOR LIST FOR HIGH SAMPLING RATE

<table>
<thead>
<tr>
<th>DATA TRACK</th>
<th>SENSOR CODE</th>
<th>SENSOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N1</td>
<td>Rotor Rig Speed</td>
</tr>
<tr>
<td>2</td>
<td>AV</td>
<td>Aft Bearing Support Vertical</td>
</tr>
<tr>
<td>3</td>
<td>AA</td>
<td>Aft Bearing Support Axial</td>
</tr>
<tr>
<td>4</td>
<td>BV</td>
<td>Base Vertical</td>
</tr>
</tbody>
</table>
of this accelerometer is low and within the sampling frequency of the ADC. The output from the accelerometers was conditioned with a PCB amplifier model 483A07. A 14 track DIFA SCADAS adjustable analog low pass filter (48 dB/octave) was used to eliminate an amplitude scaling problem associated with the ADC. The filter was set at 2,000 Hz for the low frequency data and at 16,000 Hz for the high frequency data. The filter was not used for the shock pulse data.

The vibration data was digitized and acquired by an HP 9000, Series 380 Graphics Workstation. The 11 conditioned acceleration signals and the speed signal were input into the HP 3565 front end that used a HP 35654-A controller module and 12 HP 35652-A single channel input modules with a 50 KHz maximum sampling frequency. The data acquisition software used was the LMS Signature Monitor package from the LMS CADA processor. Figures 28 and 29 show the data acquisition setup.

All the steady state data was throughput to disk to allow all the data to be post processed in the time domain. For the low frequency testing, 82,000 time history points were stored on a hard disk for each of the 14 tracks. For the high frequency testing, 82,000 time history points were recorded for 4 vibration tracks. Transient run up data was also acquired from 1000 rpm to 10,000 rpm. This data was not throughput to disk, but was sampled in block input mode (data captured at every 100 rpm).

The data that was acquired was stored as digital time histories that resulted in the maximum flexibility for post processing. All of the signal analysis techniques were performed using MATLAB on a 386SX personal computer with a math co-processor.
4.0 SIGNAL PROCESSING RESULTS

4.1 Spectrum Analysis

The data for the spectrum analysis was acquired in time history blocks from the steady state running points. The low range frequency data was used where the sampling frequency was set at 4096 Hz. The time history data was processed first in the LMS CADA system to store the time history data. This data was then written to a file in a universal time history format of 2048 data points (universal files are a standardized ascii format file [19]). This gave a frequency resolution of 2 Hz. The universal files were then converted to MATLAB files for the spectrum analysis.

The time history data from each of the test configurations was analyzed in a four step process. First, the time history was windowed with a hanning window to reduce leakage errors. Next, the data was Fourier transformed into the frequency domain. The third step was a digital integration to convert the acceleration spectrums into velocity spectrums. The final step was to take the log of the spectrum to convert into dB. The MATLAB script that performed these steps can be found in Appendix A.

Initially data from all the sensor locations was reduced and plotted. Spectrums from the 11 monitoring locations for Run 3 (load) at 5000 RPM can be found in Appendix B. It was determined from this data that the sensor location that best describes the fault conditions from a health monitoring standpoint was the Aft Mount Vertical sensor (AV). This sensor was positioned in the radial direction very close to the faulty bearing. The Aft Mount Axial sensor (AA) was the most sensitive to the bearing faults because it was positioned in the direction of the bearing load. However, this sensor did not measure the unbalance and misalignment effects.

A program was written that calculated the fault frequencies of the bearings. The input included the geometry of the bearing and the shaft speed. The output printed the inner race, outer race, ball, and cage fault frequencies. The program also calculated the modulation frequencies for the faults. Figure 30 is an example output for the rotor
Figure 30: Fault Frequency Program Output

Bearing Name or Description = 203H
Ball diameter = 0.26563
Pitch diameter = 1.11890
Contact angle (deg) = 15.000
Number of balls = 10
Rotor Speed (rpm) = 5000.00

<table>
<thead>
<tr>
<th>Shaft Fundamental Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
</tr>
<tr>
<td>1 x Shaft Speed =</td>
</tr>
<tr>
<td>2 x Shaft Speed =</td>
</tr>
<tr>
<td>3 x Shaft Speed =</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outer Race Fault Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
</tr>
<tr>
<td>1 x Ball Pass Frequency of the Outer Race =</td>
</tr>
<tr>
<td>2 x Ball Pass Frequency of the Outer Race =</td>
</tr>
<tr>
<td>3 x Ball Pass Frequency of the Outer Race =</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inner Race Fault Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
</tr>
<tr>
<td>1 x Ball Pass Frequency of the Inner Race =</td>
</tr>
<tr>
<td>2 x Ball Pass Frequency of the Inner Race =</td>
</tr>
<tr>
<td>3 x Ball Pass Frequency of the Inner Race =</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ball Fault Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
</tr>
<tr>
<td>1 x Ball Spin Frequency =</td>
</tr>
<tr>
<td>2 x Ball Spin Frequency =</td>
</tr>
<tr>
<td>3 x Ball Spin Frequency =</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cage Fault Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
</tr>
<tr>
<td>1 x Fundamental Train Frequency =</td>
</tr>
<tr>
<td>2 x Fundamental Train Frequency =</td>
</tr>
<tr>
<td>3 x Fundamental Train Frequency =</td>
</tr>
</tbody>
</table>
rig configuration when running at 5000 rpm. The code for the program can be found in Appendix C.

The method for fault detection using spectrum analysis involved plotting the spectrum and identifying the fault frequencies calculated from the program. The amplitude of the fault frequency component with a known fault was compared to the healthy configuration spectrum. Figures 31 through 35 show the Aft Bearing Support Vertical sensor for Run 3 (loaded bearing) at 5000 RPM.

When viewing the spectrum from the baseline no fault condition (Figure 31) it can be noted that there are very many frequencies present even in the healthy condition. The rotor 1/rev fundamental frequency can be identified easily at 83.3 Hz. Figure 32 which shows the spectrum of the inner race fault has the 1X, 2X, and 3X BPFI frequencies identified. Notice how these levels are present in the baseline spectrum, but are at a higher amplitude with the fault present. Similarly Figure 33 which shows the outer race fault has the 1X, 2X, and 3X BPFO frequencies identified. Figure 34 displays the spectrum from the ball fault configuration. Here the 2X, 4X, and 6X BSF components are easily identified because of the large amplitudes. The last spectrum, Figure 35, shows the misalignment fault. The 2X shaft frequency is shown at 166.6 Hz and is higher than the baseline spectrum.

A study was made on the effects of load and unbalance for the
spectrum technique of fault detection. Data from the Aft Bearing Support Vertical (AV) sensor was analyzed for the four load and unbalance conditions and can be found in Appendix D. The unbalance always increased the 1/rev component as would be expected. The unbalance also increased the fault frequencies especially for the inner race fault and the ball fault. The load on the faulty bearing would always increase the fault frequencies. This is because the more load on the bearing causes more impact energy from the fault.

![Inner Race Fault - Sensor AV - Load (run 3)](image)

Figure 32: Spectrum Analysis - Inner Race Fault

![Outer Race Fault - Sensor AV - Load (run 3)](image)

Figure 33: Spectrum Analysis - Outer Race Fault
Figure 34: Spectrum Analysis - Ball Fault

Figure 35: Spectrum Analysis - Misalignment
4.2 Shock Pulse Analysis

The shock pulse data was acquired using a PCB Flexcell mounted on the aft bearing support in the horizontal direction. This type of accelerometer has a natural frequency at approximately 6.4 kHz which could be measured with a high sampling rate. For the shock pulse data the sampling rate was set at 50 kHz.

The data was processed on the LMS CADA system to plot out the frequency spectrums for each fault condition. Figures 36 through 40 show the data for Run 3 (no unbalance with loaded bearings) for the 5 different fault conditions.

The sensors' natural frequency can be seen as the large peak in the spectrum shown in Figure 36 of the baseline configuration. The spectrums of the bearing fault configurations (Figures 37, 38, and 39) all show the same shape, but with increased amplitude at the sensors' natural frequency. The misalignment fault shown in Figure 40 is not significantly different from the baseline spectrum.

The amplitude of the spectrum at the sensors natural frequency is a measure of the health of the bearing. The data above shows that this crude method is fairly sensitive to the fault condition. A chart of the maximum vibration level at the sensors natural frequency for the above faults is shown in Figure 41. The shock pulse amplitude for the baseline and misalignment tests can be seen to be much lower than the bearing fault tests. The four sets of data

![Figure 36: Shock Pulse Analysis - Baseline](image-url)
for each fault represent the different Runs that are outlined in Table 1. It can also be noticed from Figure 41 that the ball fault shock energy is only high when the bearing is loaded (Runs 3 and 4). The effects of unbalance do not change the shock pulse response (Baseline Run 1 vs Run 2).

Figure 37: Shock Pulse Analysis - Inner Race Fault

Figure 38: Shock Pulse Analysis - Outer Race Fault
Figure 39: Shock Pulse Analysis - Ball Fault

Figure 40: Shock Pulse Analysis - Misalignment
Figure 41: Shock Pulse Peak Vibration Levels

Peak Vibration at Sensor Resonance

<table>
<thead>
<tr>
<th>Fault Type</th>
<th>No Unbalance No Load</th>
<th>Unbalance No Load</th>
<th>No Unbalance Load</th>
<th>Unbalance Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outer Race</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inner Race</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ball</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Misalignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3 Cepstrum Analysis

The data for the cepstrum analysis was acquired in time history blocks from the steady state running points. The high frequency data was used where the sampling rate was set at 32,768 Hz. The time history data was processed first in the LMS CADA system to write out time history files of 4096 data points. This gave a frequency resolution of 8 Hz. The data was then converted to MATLAB format for the cepstrum analysis.

The cepstrum processing using MATLAB involved a 5 step process. First, the data was windowed using a hanning window to reduce leakage errors. Next, an FFT was performed on the windowed time history. The third step involved calculating the power spectrum. This is done by simply multiplying the real and imaginary frequency data by its complex conjugate and taking the logarithm. The power spectrum is then fourier transformed. The last step involves plotting the data. The cepstrum will always calculate a high DC or zero frequency component, so this data was intentionally omitted from the plot. The MATLAB script files that execute the analysis can be found in Appendix A.

Cepstrum plots for the Aft Bearing Support Vertical sensor for the 5 test configurations are shown in Figures 42 through 46. This data is for Run 3 (loaded bearing with no unbalance). The baseline no fault cepstrum plot shown in Figure 42 does not show any significant peaks that would correspond to the fault frequencies. The cepstrum plot for the inner race fault, Figure 43, has a peak at 0.0019 seconds which corresponds to the BPF1 of 512.2 Hz. The outer race fault cepstrum plot shown in Figure 44 does not show a significant component at the fault frequency of 321.1 Hz or 0.0031 seconds. However the cepstrum for the outer race fault in the unloaded configuration (see Appendix E) did show this fault. The cepstrum technique detected the ball fault components the best as shown in Figure 45. The 2X BSF component at
332.5 Hz can be seen readily at 0.003 seconds. It should be noted that with the ball fault a large cage passing frequency was detected in the cepstrum. This component at 32 Hz is out of the Quefrequency range of the plot at 0.031 seconds. Reference [20] states that is not uncommon to have cage passing frequencies present when a rolling element is faulty. Figure 46 shows that the misalignment cepstrum is similar to the baseline cepstrum with no large Quefrequency components to identify.

The cepstrum signal processing was effective in identifying the inner race fault and the ball fault for the loaded bearing configuration. The fault is identified by looking at the Quefrequency value that corresponds to the known fault frequency.

The cepstrum analysis for the different load and unbalance conditions showed some interesting results. The inner race fault and the ball fault showed larger cepstrum components when load was placed on the bearing. However, the outer race fault showed a much greater cepstrum fault signature when the bearing was unloaded. The unbalance faults were not detected by the cepstrum. These results can be found in Appendix E where the cepstrum plots for the aft bearing support vertical sensor (AV) are shown for all load and unbalance tests.

![Figure 42: Cepstrum Analysis - Baseline](image-url)
Figure 43: Cepstrum Analysis - Inner Race Fault

Figure 44: Cepstrum Analysis - Outer Race Fault
Figure 45: Cepstrum Analysis - Ball Fault

Figure 46: Cepstrum Analysis - Misalignment
4.4 Signal Demodulation

The signal processing for the fault demodulation was performed in several steps. First the fault data was sampled at 32,768 Hz. The time history data was processed in the LMS CADA system to write out time history files of 4096 data points. This gave a frequency resolution of 8 Hz. The time history data was then converted to MATLAB format for analysis on a personal computer.

The demodulation process using MATLAB involved a 5 step process. First, the data was high pass filtered at 900 Hz. This frequency was chosen because the 1st harmonic of the all the different fault frequencies will fall below this frequency. The filtering process was implemented by using a digital FIR filter that used 50 terms. The high passed data was then rectified. This process simply set all negative time history points to zero. The next step involved low pass filtering the rectified time history data. To perform this step a low pass FIR filter was constructed at 900 Hz similar to the high pass filter. This time domain data now contained all the fault information shifted down from the high frequencies. The data was windowed with a hanning window before an Fourier transform was performed on the data. Now the frequency domain data could be examined for the fault frequencies that had been demodulated. The MATLAB script files that execute the analysis can be found in Appendix A.

Demodulation plots for the Aft Bearing Support Vertical sensor for the 5 test configurations is shown in Figures 47 through 51. This data is for Run 3 (loaded bearing with no unbalance). The demodulated data (dash line) is plotted with the standard FFT (solid line) for comparison of the two methods. Figure 47 shows the baseline demodulated data does not show any fault components as would be expected. The large peak at 83 Hz corresponds to the shaft fundamental frequency. The inner race fault data shown in Figure 48 does show a significant increase in the 1 and 2 BPF components when compared to the standard Fourier transform. The outer race fault demodulated data shown in Figure 49 shows a very large peak at 321 Hz which
corresponds to the BPFO component. The ball fault frequencies were only slightly enhanced by the demodulation technique as can be seen in Figure 50 where the standard Fourier transform was better at identifying the 2 and 4 BSF components. The demodulation technique did not show any faults for the misalignment data shown in Figure 51.

The demodulation analysis was found to be very sensitive to the outer race fault and inner race fault. When compared to the standard Fourier Transform, the demodulated components at the fault frequencies were much higher. The ball fault frequencies are only slightly enhanced with the demodulation process.

The demodulation process was used on the data from the different load and unbalance conditions. It can be shown from the data in Appendix F that the inner race and outer race faults were always detected with this technique. The configurations with the loaded bearings had larger fault components. The ball fault data was not as clearly enhanced as the race faults, and as expected, the baseline, unbalance, and misalignment data showed no modulated fault frequencies.

![Baseline, Sensor AV - Load (run 3)](image)

Figure 47: Demodulation - Baseline
Figure 48: Demodulation - Inner Race Fault

Figure 49: Demodulation - Outer Race Fault
Figure 50: Demodulation - Ball Fault

Figure 51: Demodulation - Misalignment
4.5 Singular Value Decomposition

The data processing method for the singular value decomposition involved two variables that were determined empirically. The first variable determined was the choice of order to be used. The order determines the size of the matrix of time history points. If the order is chosen very large (100) then the data processing time becomes significant. If the order is chosen too low (5) then a significant source of vibration could be missed. After a study on the effect of different order sizes, it was determined that an order of 30 was optimal for the rotor rig data. The other variable to be determined was the sampling frequency to be used. After investigating the SVD routine on the high sampled data (\(f_s = 32768\)) and low sampled data (\(f_s = 4096\)) it was found that the routine was more sensitive to faults using the low sampled data.

The time history data was processed for Singular Value Decomposition (SVD) in two easy steps. The first step involved constructing the coefficient matrix described as Equation 10. The first 30 time points were put in the top row of the matrix. A time shift of 1 data point was used for the second row of the matrix. The second row now contained data points 2 through 31. This same procedure was continued until row 30 was filled. The second step involved performing the SVD on the coefficient matrix. This was performed as a MATLAB routine. The routine returned the diagonal singular value matrix. The values on the diagonal were plotted on a log scale to determine where a large change in magnitude was present. These points indicate how many sources of vibration are present. The MATLAB script files for this technique can be found in Appendix A.

The singular value data for the different fault configurations was plotted with the baseline no fault condition for comparison. It can be seen from Figures 52 through 55 that the time history data with the bearing faults (dashed lines) have singular values with greater magnitudes than the no fault configuration (solid lines). This is due to the higher amplitude of the fault components. Although this effect looks encouraging for
the use in fault detection, the higher amplitudes could easily be measured by using an overall rms amplitude detector. The inflection points and the relative size of the drops in the singular values are the expected indicators of the added faults. The outer race fault data shown in Figure 53 shows a couple of significant changes (drops) in singular values at 3 and 5. The ball fault data shown in Figure 54 has the largest relative changes in singular values when compared to the baseline no fault condition. The singular values for the misalignment fault were similar to the baseline data as shown in Figure 55.

The data for the different load and unbalance conditions was analyzed using the singular value decomposition routine. Appendix G shows that the unbalance runs always caused larger significant drops in the fault data. This shows that the SVD routine is effective in measuring the added source from unbalance. The added sources from the misalignment data can only be seen in the no load or unbalance case.

![Singular Value Decomposition - Sensor AV - Load (run 3)](image)

Figure 52: SVD - Inner Race Fault and Baseline
Figure 53: SVD - Outer Race Fault and Baseline

Figure 54: SVD - Ball Fault and Baseline
Figure 55: SVD - Misalignment and Baseline
5.0. Conclusion and Recommendations

The main objective of this research was to evaluate several different signal processing techniques concerning their ability to detect rotor dynamic faults. This objective was accomplished by testing five different algorithms on data acquired from a test rig where controlled faults were introduced into the system. Each algorithm was used on the same fault data so that a relative comparison could be made. The focus of the research was on the signal processing techniques and not on the actual rotor dynamic fault. The rotor rig did not test faults of different severity, but instead tested healthy verses known fault conditions. The rotor rig and the data acquired were effective tools used to investigate the different signal processing algorithms.

The different techniques can be evaluated on two main criteria. The first is on the ability to detect a faulty environment, and the second is on the ability to locate the fault. Table 5 summarizes the different signal processing techniques for the faults that were tested.

The spectrum analysis technique proved to be a simple and effective algorithm for the identification of the fault frequencies. This method was used as the standard technique that the other methods were evaluated against. The bearing fault data from the test rig simulated a fairly degraded condition were the fault frequencies were easily identified in the spectrums. However, the baseline (no-fault) condition had a frequency spectrum with a great deal of components. In order to find the fault the user must know exactly where to look in the spectrum. This process was aided by the use of the frequency identification program. A disadvantage to this method is that many times the machinery will have many different bearings and gears that would make the frequency identification very complicated. Also, if there is a significant amount of bearing slip, the fault components will not be at the frequencies where they are calculated. For the
<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Misalignment</th>
<th>Unbalance</th>
<th>Faults</th>
<th>Inner Race</th>
<th>Outer Race</th>
<th>Ball</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectrum</td>
<td>-Easy to use</td>
<td>-Not sensitive to faults in early stages</td>
<td>good</td>
<td>fair</td>
<td>fair</td>
<td>fair</td>
<td>fair</td>
<td>fair</td>
</tr>
<tr>
<td></td>
<td>-Location of fault determined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Works on all types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>of faults</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shock</td>
<td>-Easy to use</td>
<td>-Will not locate source</td>
<td>bad</td>
<td>bad</td>
<td>good</td>
<td>good</td>
<td>good</td>
<td>good</td>
</tr>
<tr>
<td></td>
<td>-Sensitive to bearing faults</td>
<td>-Not good for unbalance and misalignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cepstrum</td>
<td>-Sensitive to bearing faults</td>
<td>-Extra Processing and data interpretation</td>
<td>bad</td>
<td>bad</td>
<td>fair</td>
<td>fair</td>
<td>fair</td>
<td>good</td>
</tr>
<tr>
<td></td>
<td>-Location of fault</td>
<td>-Not good for unbalance and misalignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>determined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demodulation</td>
<td>-Sensitive to bearing faults</td>
<td>-Extra processing</td>
<td>bad</td>
<td>bad</td>
<td>good</td>
<td>good</td>
<td>fair</td>
<td>fair</td>
</tr>
<tr>
<td></td>
<td>-Location of fault</td>
<td>-Need to set filters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>determined</td>
<td>-Not good for unbalance and misalignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVD</td>
<td>-Time domain algorithm</td>
<td>-Will not locate source</td>
<td>fair</td>
<td>bad</td>
<td>fair</td>
<td>fair</td>
<td>fair</td>
<td>good</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Sensitive to noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
rotor rig data it was easy to identify the fault frequencies due to the simple structure with one rotor and two bearings. There was no bearing slip measured for both the loaded and unloaded configurations. The spectrum analysis technique is not well suited for initial detection of a faulty environment, but is better at identification of the fault location.

The shock pulse method of fault detection measured the high frequency energy that excites the accelerometers natural frequency. Although this method is extremely simple, it was shown to be sensitive to the bearing faults that produced the impulsive forces. When compared to the spectrum analysis method, the shock pulse method was found to be a better algorithm for determination of a healthy verses unhealthy condition for bearing faults. The method did not work for unbalance or misalignment faults. The main disadvantage of this method is the inability to identify the location of the fault. The only information that this analysis will produce is determining that somewhere in the system there is an impulsive force that is exciting the sensor’s natural frequency.

The cepstrum analysis extracted the harmonic frequencies that resulted from the bearing faults. The harmonic content of the faulty data varied depending on the load and type of bearing fault. For this reason the cepstrum technique was sometimes more sensitive than the spectrum technique, and sometimes less sensitive. The ball fault was detected very well with the cepstrum analysis when the bearing was loaded. Conversely, the outer race fault only showed up in the cepstrum when the bearing was unloaded. The reason for the inconsistency in the technique can be speculated by the dynamics of the fault. The loaded bearing would have higher impact energy that would result in greater harmonic content. The unloaded bearing would have more clearances and non linearities that would result in harmonics of the fault frequencies. The inner race fault and the misalignment fault were detected better with the spectrum technique than the cepstrum technique. Inner race faults are always harder to detect because the
fault is furthest removed from the sensor. Also the inner race fault will rotate into and out of the load zone reducing the overall time the fault frequencies are excited. The misalignment fault was not detected by the cepstrum technique because misalignment faults do not create as many harmonics as bearing faults.

The demodulation technique was found to be very good at enhancing the inner race and outer race fault data. When compared to the spectrum analysis technique, the demodulated data was more sensitive in showing the presence of the fault. This shows that much of the fault frequency information is translated to higher frequencies outside the normal frequency range of interest. The ball fault demodulated data showed the fault components, but they were not significantly enhanced when compared to the spectrum technique. The data with the loaded bearings had higher demodulated fault frequencies. Although this method requires more computations and post processing, it was found to be worth the effort for the ability to detect the rotor dynamic bearing faults.

The singular value decomposition method was used to identify the added vibration sources from the rotor faults. This routine, which is based on theory from parameter estimation routines, was found to be sensitive to the overall amplitude of the data and to the noise on the data. The SVD routine had the most success with the ball fault data where large drops in the singular values indicated the significant sources from the fault frequencies. The use of this algorithm is similar to the shock pulse method where a faulty condition is identified, but the location of the fault is not identified. In contrast to the shock pulse method, the svd routine was intended to find low frequency faults, such as misalignment and unbalance, as well as the high frequency bearing faults. This did not work out as well as expected. The rotor rig data had very many frequency components as can be seen from the baseline no fault condition. When one extra frequency component is added from a misalignment to the hundreds of other frequencies, the SVD routine had a hard time finding this extra component. The SVD
routine was better with the bearing faults because they added several components for each fault.

Each of the different signal processing techniques had strengths and weaknesses. It was determined that the narrow band spectrum analysis is a versatile routine that is useful if the proper frequencies can be identified, and if the level of fault is large enough to result in a significant component at the fault frequency. The major problem with this routine is that it is difficult to identify faults in the early stages. Both the cepstrum technique and the signal demodulation technique were found to be very good at enhancing the bearing fault information. The cepstrum technique was the best for the ball fault while the signal demodulation technique was the best for the inner and outer race faults. The only problems with using these methods is that they require extra signal processing steps and some data interpretation. The shock pulse method and the SVD routine were used to determine a healthy verses faulty condition. The shock pulse method was simple and effective in finding the bearing faults. The SVD routine found the added components from the ball fault, but did not work as well on the other faults.

Based on the results from this investigation, several recommendations can be made for related areas of research. The next step in evaluating these different signal processing algorithms would be to test them using data where the level of fault is increasing from a healthy condition to a total failure condition. This would investigate the sensitivity at the early stages of the fault, which are the most important. Another area of further research would be to investigate the ability of the different techniques to detect faults from a transient speed conditions. Rotating machinery such as aircraft engines would need health monitoring for transients as well as steady state operation. It would also be useful to investigate the signal processing techniques on combinations of faults such as both an inner race fault and a ball fault simultaneously. Many times a typical bearing failure would have multiple faults. For example, a race will pit and
spall causing the rolling element to pick up the debris and also become faulty. There are also many other types of rotor dynamic faults that could be used to test the processing algorithms. Faults such as cracked shafts and mechanical looseness will result in unique vibration signatures that could be identified by the proper processing technique.

Some of the signal processing algorithms could improved based on the results from this research. The spectrum analysis technique could be improved by using a time domain averaging algorithm that is synchronized by the fault component. The exact speed of the bearing cage is needed for this type of analysis to work, because any bearing slip at all will cause the averaging process to smear the fault frequency of interest. The shock pulse analysis could be automated so as to band pass filter the signal at the sensors natural frequency. Then a simple overall level detector could be used to measure the shock pulse content. This would eliminate the need for the Fourier transform and the spectrum inspection that was used for this investigation. Another follow on project could be to investigate the complex cepstrum. This way many different faults could be separated by the use of filtering (liftering) the complex cepstrum. The singular value decomposition routine could be improved and fine tuned by using data with less spectral content.

There are different ways of using the singular value decomposition routine that would be interesting to investigate. The method used in this thesis involved using one sensor’s data to fill the coefficient matrix. Perhaps a better way to monitor the health of the entire structure would utilize many different sensors on the structure to fill the coefficient matrix. This method would evaluate the global or overall health of the system. By using the spatial information (where the sensors are located) the source of the fault could be localized.

The use of these algorithms is recommended for a health monitoring process. The most versatile method would utilize a combination of all the signal processing
techniques previously mentioned. The shock pulse method and SVD routine could be used to flag the start of a rotor dynamic fault. This could initiate a detailed analysis of bearing components using both the cepstrum analysis and signal demodulation analysis. The spectrum analysis technique could be used to find the low frequency unbalance and misalignment faults, and also backup the cepstrum and demodulation techniques for bearing faults in the advanced stages. For critical machinery the individual bearing components could be continuously monitored with the cepstrum and demodulation technique. A data base of each bearing component in the engine or machine could be continuously tracked. An Artificial Intelligence (AI) system could be used to identify the fault from the pre-processed data and determine the time to failure.
6.0 REFERENCES

Health Monitoring


Rotor Dynamics


Signal Processing


Bearings


[23] Bruel & Kjaer “Detecting Faulty Rolling Element Bearings” Application Notes BO0210-11, no date


[26] Taylor, J., “Determination of Antifriction Bearing Condition by Spectral Analysis” Sixth in a series of Technology Interchange on Machinery Vibration Monitoring and Analysis, Published by Vibration Institute, 1978


Other

# APPENDIX A

**MATLAB SCRIPT FILES**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Spectrum Analysis</td>
<td>A–2</td>
</tr>
<tr>
<td>2) Shock Pulse Analysis</td>
<td>A–5</td>
</tr>
<tr>
<td>3) Cepstrum Analysis</td>
<td>A–7</td>
</tr>
<tr>
<td>4) Signal Demodulation</td>
<td>A–10</td>
</tr>
<tr>
<td>5) Singular Value Decomposition</td>
<td>A–15</td>
</tr>
</tbody>
</table>
file rrspect.m
written to plot log (db) spectrums from rotor rig
fault data – Paul Grabill 1992

clear
clf

% set up independent variables
N=2048;
Fs=4096;
time=1:N;
time=time/Fs;
freq=0:N-1;
freq=freq*Fs/N;
w=2*pi*freq;
% scale factor for data acq. error
scale=5.464;
% ref for dB ips pk
ref=5.5711e-7;

axis([0 2000 0 120]);

% h= hanning window to reduce leakage
h=hanning(N);

% load baseline data
load c:\matlab\prog\data\bl_low
% window data
dat=dat(1,:).*h';

% fourier transform data
dataf=fft(dat);

% digital integration
dataf(2:N)=dataf(2:N)./(w(2:N)*scale*2/N);
% convert to log and plot
db=20*log10(abs(dataf)/ref);
plot(freq(4:N/2),db(4:N/2))
xlabel('Frequency (Hz)')
ylabel('Velocity (VdB re 1e-8 m/sec rms) ')
txt1=['Baseline – Sensor AV – No Load or Unbal (run 1) '];
title(txt1)
pause
%% process inner race fault data in the same manner

load c:\matlab\prog\data\irf_low
dat=data(1:4).*h';
datf=fft(dat);
datf(2:N)=datf(2:N)./w(2:N)*scale*2/N;
db=20*log10(abs(datf)./ref);
plot(freq(4:N/2),db(4:N/2))
xlabel('Frequency (Hz)')
ylabel('Velocity (VdB re 1e-8 m/sec rms)')
txt1=['Inner Race Fault – Sensor AV – No Load or Unbal (run 1)'];
title(txt1)
pause
meta 1

%% process outer race fault data

load c:\matlab\prog\data\orf_low
dat=data(1:4).*h';
datf=fft(dat);
datf(2:N)=datf(2:N)./w(2:N)*scale*2/N;
db=20*log10(abs(datf)./ref);
plot(freq(4:N/2),db(4:N/2))
xlabel('Frequency (Hz)')
ylabel('Velocity (VdB re 1e-8 m/sec rms)')
txt1=['Inner Race Fault – Sensor AV – No Load or Unbal (run 1)'];
title(txt1)
pause
meta 1

%% process ball fault data

load c:\matlab\prog\data\bf_low
dat=data(1:4).*h';
datf=fft(dat);
datf(2:N)=datf(2:N)./w(2:N)*scale*2/N;
db=20*log10(abs(datf)./ref);
plot(freq(4:N/2),db(4:N/2))
xlabel('Frequency (Hz)')
ylabel('Velocity (VdB re 1e-8 m/sec rms)')
txt1=['Ball Fault – Sensor AV – No Load or Unbal (run 1)'];
title(txt1)
pause
meta 1

%% process misalignment data

load c:\matlab\prog\data\mis_low
dat=data(1:4).*h';
dataf=fft(dat);
dataf(2:N)=dataf(2:N)./w(2:N)*scale*2/N;
db=20*log10(abs(dataf)./ref);
plot(freq(4:N/2),db(4:N/2))
xlabel('Frequency (Hz)')
ylabel('Velocity (VdB re 1e-8 m/sec rms) ')
txt1=['Misalignment - Sensor AV - No Load or Unbal (run 1)'];
title(txt1)
pause
meta 1
file rrshock.m
written to plot log (db) spectrums of shock pulse data
Paul Grabill  1992

clear
clg

set up independent variables

N=4096;
Fs=50000;
time=1:N;
time=time/Fs;
freq=0:N-1;
freq=freq*Fs/N;
w=6.283185*freq;
%  scale factor for data acq. error
scale=0.00708;
%  ref for dB ips pk
ref=5.5711e–7;

axis([0 25000 0 6]);

set up hanning window

h=hanning(N);

process baseline data

load c:\matlab\work\data\shock.dat
dat=data(1:end).*h';
%dataf=fft(dat);
db=abs(fft(dat))*scale*2/N;
%dataf(2:N)=dataf(2:N)./w(2:N)*scale*2/N;
%db=20*log10(abs(dataf)./ref);
plot(freq(4:N/2),db(4:N/2))
xlabel('Frequency (Hz)')
ylabel('Acceleration (g pk) ')
txt1=['Baseline – Sensor SH '];
title(txt1)
pause
meta shock

process ball fault data

dat=data(2:end).*h';
%dataf=fft(dat);
db=abs(fft(dat))*scale*2/N;
%dataf(2:N)=dataf(2:N)./w(2:N)*scale*2/N;
%db=20*log10(abs(dataf)./ref);
plot(freq(4:N/2),db(4:N/2))
% process inner race fault data

dat=data(3,:).*h';
%dataf=fliplr(dat);
db=abs(fliplr(dat)).*scale*2/N;
%dataf(2:N)=dataf(2:N)./w(2:N).*scale*2/N;
%db=20*log10(abs(dataf)./ref);
plot(freq(4:N/2),db(4:N/2))
ylabel('Acceleration (g pk) ')
txt1=['Inner Race Fault – Sensor SH '];
title(txt1)
pause
meta shock

% process outer race fault data

dat=data(4,:).*h';
%dataf=fliplr(dat);
db=abs(fliplr(dat)).*scale*2/N;
%dataf(2:N)=dataf(2:N)./w(2:N).*scale*2/N;
%db=20*log10(abs(dataf)./ref);
plot(freq(4:N/2),db(4:N/2))
ylabel('Acceleration (g pk) ')
txt1=['Outer Race Fault – Sensor SH '];
title(txt1)
pause
meta shock

% process misalignment data

dat=data(5,:).*h';
%dataf=fliplr(dat);
db=abs(fliplr(dat)).*scale*2/N;
%dataf(2:N)=dataf(2:N)./w(2:N).*scale*2/N;
%db=20*log10(abs(dataf)./ref);
plot(freq(4:N/2),db(4:N/2))
ylabel('Acceleration (g pk) ')
txt1=['Misalignment – Sensor SH '];
title(txt1)
pause
meta shock
% prog rrcep.m
% This program performs cepstrum analysis on the rotor rig data
% written by Paul Grabill 1992

clc
clear

% set up independent variables

N=4096;
Fs=32768;
time=0:N-1;
time=time/Fs;
freq=0:N-1;
freq=freq*N/Fs;
win=hanning(N);
% scale factor for data acq. error
scale=5.464;
axis([0.001 0.009 0 3.5e5]);

% load baseline data
load c:\matlab\prog\data\blr2r4h

% window data with hanning window
x=data(4,:).*win';

% Take Fourier Transform of data
datfft=fft(x)*4/length(x);

% Calculate power spectrum
power=datfft.*conj(datfft);
lpower=log10(power);

% Take Fourier transform of power spectrum
cdat=fft(lpower);

% Calculate the power cepstrum
c=abs(cdat).*abs(cdat);

% Plot data
plot(time(25:N/10),c(25:N/10))
xlabel('Queuecency (sec)')
ylabel('Amplitude')
txt1=['Cepstrum Analysis – Baseline – Sensor AV – Load + Unbal (run 4)'];
% process outer race fault data

load c:\matlab\prog\data\orfr2r4h
x=data(4,:).*win';
datfft=fft(x)*4/length(x);
power=datfft.*conj(datfft);
lpower=log10(power);
cdat=fft(lpower);
c=abs(cdat).*abs(cdat);
plot(time(25:N/10),c(25:N/10))
xlabel('Quefrency (sec)')
ylabel('Amplitude')
txt1=['Cepstrum Analysis – Outer Race Fault – Sensor AV – Load + Unbal (run 4)'];
title(txt1)
pause
meta c4

% Process inner race fault data

load c:\matlab\prog\data\irfr2r4h
x=data(4,:).*win';
datfft=fft(x)*4/length(x);
power=datfft.*conj(datfft);
lpower=log10(power);
cdat=fft(lpower);
c=abs(cdat).*abs(cdat);
plot(time(25:N/10),c(25:N/10))
xlabel('Quefrency (sec)')
ylabel('Amplitude')
txt1=['Cepstrum Analysis – Inner Race Fault – Sensor AV – Load + Unbal (run 4)'];
title(txt1)
pause
meta c4

% process ball fault data

load c:\matlab\prog\data\bfrr2r4h
x=data(4,:).*win';
datfft=fft(x)*4/length(x);
power=datfft.*conj(datfft);
lpower=log10(power);
cdat=fft(lpower);
c=abs(cdat).*abs(cdat);
plot(time(25:N/10),c(25:N/10))
xlabel('Quefrency (sec)')
ylabel('Amplitude')
txt1=['Cepstrum Analysis – Ball Fault – Sensor AV – Load + Unbal (run 4)'];
title(txt1)
pause
meta c4

% process misalignment data

load c:\matlab\prog\data\misr2r4h
x=data(4,:).*win';
datfft=filt(x)*4/length(x);
power=datfft.*conj(datfft);
lpower=log10(power);
cdat=filt(lpower);
c=abs(cdat).*abs(cdat);
plot(time(25:N/10),c(25:N/10))
xlabel('Quefrancy (sec)')
ylabel('Amplitude')
txt1=['Cepstrum Analysis – Misalignment – Sensor AV – Load + Unbal (run 4)'];
title(txt1)
pause
meta c4
file rrmod.m

program to demodulate rotor rig fault test data
written by P. Grabill 1992

clear
c1g

% set up time and freq vectors

N=4096;
fs=32768;
t=0:N-1;
t=t/fs;
f=0:N-1;
f=f/fs/N;
w=6.283185*f;

% scale factor for data acq. error
scale=5.464;
% ref for dB ipss pk
ref=386.089;
track=1;
h=hanning(N);

% load baseline data
load c:\matlab\prog\data\bl_high
x=data(track,:);

% scale data and apply a hanning window
x=x.*scale;
x=x.*h';

% take FFT for comparison to final result
xft=fft(x)*2/N;

% create high pass filter with firl
term=100;
cutoff=900;
wcut=cutoff/fs*2;
filt=firl(term,wcut,'high');

% convolve filter with data
xconv=conv(x,filt);
xs=xconv(term/2:term/2+N-1);

% rectify high freq modulated data
for k=1:N
  if(xs(k)<0)
    xs(k)=0;
  end
end

% create low pass filter with fir1
wcutoff=cutoff/fs*2;
flt=fir1(term,wcutoff);

% convolve filter with data
xconv=conv(xs,flt);
xlow=xconv(term/2:term/2+N-1);

% window filtered data
maxx=N*cutoff/fs;
xlow=xlow.*h';

% FFT Demodulated data and digitally integrate
xlowft=fliplr(fliplr(xlowft(2:N)));
xlowft(2:N)=xlowft(2:N)/w(2:N);
xf(2:N)=xf(2:N)/w(2:N);

% Plot results
plot((fx:maxx),abs(xf(4:maxx)),fx(maxx),abs(xlowft(4:maxx)),'--')
xlabel('Frequency (Hz)')
ylabel('Velocity (in/sec)')
title(['Baseline - Sensor AV - No Load or Unbal (run 1)'])
text(0.65,0.8,'Solid - FFT','sc')
text(0.65,0.75,'Dash - Demodulated Signal','sc')
meta r1
pause

% process outer race fault data
load c:\matlab\prog\data\orf_high
x=data(track,:);
x=x.*scale;
x=x.*h';
xf=fliplr(fliplr(xf(2:N)));
xf(2:N)=xf(2:N)/w(2:N);
cutoff=900;
wcutoff=cutoff/fs*2;
flt=fir1(term,wcutoff,high);
xconv=conv(x,flt);
xs=xconv(term/2:term/2+N-1);
for k=1:N
    if(xs(k)<0)
        xs(k)=0;
    end
end
wcut=cutoff/fs*2;
filt=fir1(term,wcut);
xconv=conv(xs,filt);
xlow=xconv(term/2:term/2+N-1);
maxx=N*cutoff/fs;
xlow=xlow.*h';
xlowft=filtft(xlow)*2/N;
xlowft(2:N)=xlowft(2:N)./w(2:N);
xfit(2:N)=xfit(2:N)./w(2:N);
plot((4:maxx),abs(xfit(4:maxx)),f(4:maxx),abs(xlowft(4:maxx)),'—')
xlabel('Frequency (Hz)')
ylabel('Velocity (in/sec)')
title(['Outer Race Fault – Sensor AV – No Load or Unbal (run 1)'])
text(0.65,0.8,{'Solid – FFT','sc'})
text(0.65,0.75,{'Dash – Demodulated Signal','sc'})
meta r1
pause

% process inner race fault data
load c:\matlab\prog\data\irf_high
x=data(track,:);
x=x.*scale;
x=x.*h';
xfit=filtft(x)*2/N;
term=100;
cutoff=900;
wcut=cutoff/fs*2;
filt=fir1(term,wcut,'high');
xconv=conv(x,filt);
x=xconv(term/2:term/2+N-1);
for k=1:N
    if(xs(k)<0)
        xs(k)=0;
    end
end
wcut=cutoff/fs*2;
filt=fir1(term,wcut);
xconv=conv(xs,filt);
xlow=xconv(term/2:term/2+N-1);
maxx=N*cutoff/fs;
xlow=xlow.*h';
xlowft=filtft(xlow)*2/N;
xlowft(2:N)=xlowft(2:N)./w(2:N);
% Process ball fault data
load c:\matlab\prog\data\bf_high
x=data(track,:);
x=x.*scale;
x=x.*h';
xf=fft(x)*2/N;
term=100;
cutoff=900;
wcut=cutoff/fs*2;
flt=fir1(term,wcut,'high');
xconv=conv(x,flt);
x=xconv(term/2:term/2+N−1);
for k=1:N
    if(xs(k)<0)
        xs(k)=0;
    end
end
wcut=cutoff/fs*2;
flt=fir1(term,wcut);
xconv=conv(xs,flt);
xlow=xconv(term/2:term/2+N−1);
maxx=N*cutoff/fs;
xlow=xlow.*h';
xlow=fft(xlow)*2/N;
xlowf=fft(xlow);xlowf(2:N)=xlowf(2:N)/w(2:N);
xf(2:N)=fft(xlowf)/w(2:N);
plot(f(4:maxx),abs(xf(4:maxx)),f(4:maxx),abs(xlowf(4:maxx)),'--')
xlabel('Frequency (Hz)')
ylabel('Velocity (in/sec)')
title(['Ball Fault – Sensor AV – No Load or Unbal (run 1)'])
text(0.65,0.8,'Solid – FFT', 'sc')
text(0.65,0.75,'Dash – Demodulated Signal', 'sc')
meta r1
pause

% process misalignment data
load c:\matlab\prog\data\mis_high
x=data(track,:);
x=x.*scale;
x = x.*h';
xft = fft(x)*2/N;
term = 100;
cutoff = 900;
wcut = cutoff/fs*2;
fil = fir1(term, wc, 'high');
xconv = conv(x, fil);
x = xconv(term/2:term/2+N-1);
for k = 1:N
    if(x(k)<0)
        x(k) = 0;
    end
end
wcut = cutoff/fs*2;
fil = fir1(term, wc);
xconv = conv(x, fil);
xlow = xconv(term/2:term/2+N-1);
maxx = N*cutoff/fs;
xlow = xlow.*h';
xlow = fft(xlow)*2/N;
xlowf = xlowf/(2:N);
xft(2:N) = xft(2:N) ./w(2:N);
plot(f(4:maxx), abs(xft(4:maxx)), f(4:maxx), abs(xlowf(4:maxx)), '—')
xlabel('Frequency (Hz)')
ylabel('Velocity (in/sec)')
title(['Misalignment - Sensor AV - No Load or Unbal (run 1)'])
text(0.65, 0.8, 'Solid - FFT', 'sc')
text(0.65, 0.75, 'Dash - Demodulated Signal', 'sc')
meta r1
pause
file rrsvd.m

This program calculates the svd on rotor rig test data.
Written by Paul Grabill 1992

clear
clg

load baseline data

load c:\matlab\prog\data\br2r41

n=input('Enter the number of data points or 0 to quit ');
svdmat=zeros(5,n);
sv=1:n;

% shift data into file

start=10;

% choose sensor to process

k=input('Enter the vibration track ');

% load matrix

for kk=1:n
    svdata(kk,:)=data(k,kk+start:n+k-1+start);
end

% perform SVD

[v,s,y]=svd(svdata);
s=diag(s);
npts=length(s);
svdmat(1,1:npts)=s';

% process outer race fault data

load c:\matlab\prog\data\orfr2r41
for kk=1:n
    svdata(kk,:)=data(k,kk+start:n+k-1+start);
end
[v,s,y]=svd(svdata);
s=diag(s);
svdmat(2,1:npts)=s';

% process inner race fault data

load c:\matlab\prog\data\irfr2r41
for kk=1:n
    svdata(kk,:)=data(k,kk+start:n+k-1+start);
end
[v,s,y]=svd(svdata);
s=diag(s);
svdmat(3,1:npts)=s';

% process ball fault data
load c:\matlab\prog\data\bfr2r4l
for kk=1:n
    svdata(kk,:)=data(k,kk+start:n+kk-1+start);
end
[v,s,y]=svd(svdata);
s=diag(s);
svdmat(4,1:npts)=s';

% process misalignment fault data
load c:\matlab\prog\data\mistr2r4l
for kk=1:n
    svdata(kk,:)=data(k,kk+start:n+kk-1+start);
end
[v,s,y]=svd(svdata);
s=diag(s);
svdmat(5,1:npts)=s';

% plot data
semilogy(sv,svdmat(1,:),'-','sv,svdmat(2,:),--')
title('Singular Value Decomposition – Sensor AV – Load + Unbalance (run 4)')
xlabel('singular values')
ylabel('amplitude')
text(0.2,0.5,'solid – baseline','sc')
text(0.2,0.45,'dash – outer race fault','sc')
meta orf
pause
semilogy(sv,svdmat(1,:),'-','sv,svdmat(3,:),--')
title('Singular Value Decomposition – Sensor AV – Load + Unbalance (run 4)')
xlabel('singular values')
ylabel('amplitude')
text(0.2,0.5,'solid – baseline','sc')
text(0.2,0.45,'dash – inner race fault','sc')
meta irf
pause
semilogy(sv,svdmat(1,:),'-','sv,svdmat(4,:),--')
title('Singular Value Decomposition – Sensor AV – Load + Unbalance (run 4)')
xlabel('singular values')
ylabel('amplitude')
text(0.2,0.5,'solid – baseline','sc')
text(0.2,0.45,'dash – ball fault','sc')
%text(0.2,0.4,'fs=32768','sc')
APPENDIX B

SPECTRUMS FROM ALL ROTOR RIG SENSORS
CONFIGURATION: OUTER RACE FAULT
SPEED: 5000 RPM
RUN 3: NO UNBALANCE WITH LOADED BEARINGS

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Speed Sensor (N1)</td>
<td>B-2</td>
</tr>
<tr>
<td>2) Fwd Bearing Support Vert (FV)</td>
<td>B-3</td>
</tr>
<tr>
<td>3) Fwd Bearing Support Tang (FT)</td>
<td>B-4</td>
</tr>
<tr>
<td>4) Fwd Bearing Support Axial (FA)</td>
<td>B-5</td>
</tr>
<tr>
<td>5) Aft Bearing Support Vert (AV)</td>
<td>B-6</td>
</tr>
<tr>
<td>6) Aft Bearing Support Tang (AT)</td>
<td>B-7</td>
</tr>
<tr>
<td>7) Aft Bearing Support Axial (AA)</td>
<td>B-8</td>
</tr>
<tr>
<td>8) Base Vert (BV)</td>
<td>B-9</td>
</tr>
<tr>
<td>9) Fwd Bearing Support Lat (FL)</td>
<td>B-10</td>
</tr>
<tr>
<td>10) Aft Bearing Support Lat (AL)</td>
<td>B-11</td>
</tr>
<tr>
<td>11) Motor Vert (MV)</td>
<td>B-12</td>
</tr>
<tr>
<td>12) Motor Lat (ML)</td>
<td>B-13</td>
</tr>
</tbody>
</table>
Outer Race Fault - Run 3 - 5000 RPM - FWD BEARING SUPPORT AXIAL
Outer Race Fault - Run 3 - 5000 RPM - AFT BEARING SUPPORT VERT

Frequency (Hz)

Velocity (VdB re 1e-8 m/sec rms)
Outer Race Fault - Run 3 - 5000 RPM - AFT BEARING SUPPORT LATERAL

Velocity (VdB re 1e-8 m/sec rms)

Frequency (Hz)
APPENDIX C

ROTOR FAULT FREQUENCY PROGRAM
#include <math.h>
#include <stdio.h>
#define MAX 6

void main() {
    
     /* Program brgfault */
     /* This program will calculate the frequencies of classical bearing faults */
     /* Written by Paul Grable Dec 28, 1991 */

     FILE *out_ptr;    /* output file pointer */
     char string[50]; /* string for gets to scan */
     int i;           /* counter */
     int status;      /* status return from scanf */
     float ball_dia;  /* ball diameter */
     float pitch_dia; /* pitch diameter */
     float cont_ang;  /* contact angle */
     int number_balls; /* number of balls */
     float speed;     /* speed of bearing */
     double angle;    /* contact angle calculated */
     float bpfo;      /* ball pass frequency of the outer race */
     float bpsi;      /* ball pass frequency of the inner race */
     float bsf;       /* ball spin frequency */
     float fft;       /* fundamental train frequency */
     float result;    /* temp result */

     /* first read in data for bearing */

     printf("Enter bearing name or description (one word)\n");
     status=scanf("%s",string);

     printf("Enter the ball diameter \n");
     status=scanf("%f",&ball_dia);

     printf("Enter the pitch diameter \n");
     status=scanf("%f",&pitch_dia);

     printf("Enter the contact angle (deg) \n");
     status=scanf("%f",&cont_ang);
printf("Enter the number of balls in the bearing \n");
status=scanf("%d", &number_balls);

printf("Enter the rotor speed (RPM) \n");
status=scanf("%f", &speed);

out_ptr = fopen("brg.out", "w");

fprintf(out_ptr, "Bearing Name or Description = %s\n",string);
fprintf(out_ptr, "Ball diameter = %8.5f\n", ball_dia);
fprintf(out_ptr, "Pitch diameter = %8.5f\n", pitch_dia);
fprintf(out_ptr, "Contact angle (deg) = %6.3f\n", cont_ang);
fprintf(out_ptr, "Number of balls = %3d\n", number_balls);
fprintf(out_ptr, "Rotor Speed (rpm) = %8.2f\n\n", speed);

angle=cos(0.01745329*cont_ang);
bpfo=(speed/2)*number_balls*(1-(ball_dia/pitch_dia*angle));
bpfi=(speed/2)*number_balls*(1+(ball_dia/pitch_dia*angle));
bsf=speed*(pitch_dia/(2*ball_dia))*(1-(ball_dia/pitch_dia*angle))*(ball_dia/pitch_dia*angle);
fft=speed/2*(1-ball_dia/pitch_dia*angle);

/* shaft fundamental frequencies */

fprintf(out_ptr, "Shaft Fundamental Frequencies\n\n");
fprintf(out_ptr, "RPM    Hz    Orders\n");
for (i=1; i<MAX+1; i++){
    fprintf(out_ptr, "%d x Shaft Speed = \n", i, speed, i*speed/60, (float)i);
}

/* outer race fault frequencies */

fprintf(out_ptr, "Outer Race Fault Frequencies\n");
fprintf(out_ptr, "RPM    Hz    Orders\n");
for (i=1; i<MAX+1; i++){
    result=i*bpfo-speed;
    fprintf(out_ptr," modulation of - shaft fund. = %8.2f \ %7.3f \ %7.3f\n",result,result/60,result/speed);
    result=i*bpfo;
    fprintf(out_ptr," %d x Ball Pass Frequency of the Outer Race = %8.2f \ %7.3f \ %7.3f\n",i,result,result/60,result/speed);
result = i*bpf0 + speed;
fprintf(out_ptr," modulation of + shaft fund. = %8.2f %7.3f %7.3fn",result,result/60,result/speed);
}

/* inner race fault frequencies */

fprintf(out_ptr,"n\nInner Race Fault Frequencies\n\n");
fprintf(out_ptr," RPM  Hz  Orders\n");
for (i=1;i<MAX+1;i++){
    result = i*bpf1 - speed;
    fprintf(out_ptr," modulation of - shaft fund. = %8.2f %7.3f %7.3fn",result,result/60,result/speed);
    result = i*bpf1;
    fprintf(out_ptr,"%d x Ball Pass Frequency of the Inner Race = %8.2f %7.3f %7.3fn",i,result,result/60,result/speed);
    result = i*bpf1+speed;
    fprintf(out_ptr," modulation of + shaft fund. = %8.2f %7.3f %7.3fn",result,result/60,result/speed);
}

/* ball fault frequencies */

fprintf(out_ptr,"n\nBall Fault Frequencies\n\n");
fprintf(out_ptr," RPM  Hz  Orders\n");
for (i=1;i<MAX+1;i++){
    result = i*bfs - speed;
    fprintf(out_ptr," modulation of - shaft fund. = %8.2f %7.3f %7.3fn",result,result/60,result/speed);
    result = i*bfs;
    fprintf(out_ptr,"%d x Ball Spin Frequency = %8.2f %7.3f %7.3fn",i,result,result/60,result/speed);
    result = i*bfs+speed;
    fprintf(out_ptr," modulation of + shaft fund. = %8.2f %7.3f %7.3fn",result,result/60,result/speed);
}

/* train fault frequencies */

fprintf(out_ptr,"n\nCage Fault Frequencies\n\n");
fprintf(out_ptr," RPM  Hz  Orders\n");
for (i=1;i<MAX+1;i++){
    result = i*ff;
    fprintf(out_ptr,"%d x Fundamental Train Frequency = %8.2f %7.3f %7.3fn",i,result,result/60,result/speed);
}
APPENDIX D

SPECTRUMS FROM LOAD AND UNBALANCE CONFIGURATIONS
SPEED: 5000 RPM
SENSOR: AFT BEARING SUPPORT VERTICAL (AV)

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Baseline no load or unbal</td>
<td>D-2</td>
</tr>
<tr>
<td>2) Baseline unbalance</td>
<td>D-3</td>
</tr>
<tr>
<td>3) Baseline load</td>
<td>D-4</td>
</tr>
<tr>
<td>4) Baseline load and unbal</td>
<td>D-5</td>
</tr>
<tr>
<td>5) Inner race fault no load or unbal</td>
<td>D-6</td>
</tr>
<tr>
<td>6) Inner race fault unbalance</td>
<td>D-7</td>
</tr>
<tr>
<td>7) Inner race fault load</td>
<td>D-8</td>
</tr>
<tr>
<td>8) Inner race fault load and unbal</td>
<td>D-9</td>
</tr>
<tr>
<td>9) Outer race fault no load or unbal</td>
<td>D-10</td>
</tr>
<tr>
<td>10) Outer race fault unbalance</td>
<td>D-11</td>
</tr>
<tr>
<td>11) Outer race fault load</td>
<td>D-12</td>
</tr>
<tr>
<td>12) Outer race fault load and unbal</td>
<td>D-13</td>
</tr>
<tr>
<td>13) Ball fault no load or unbal</td>
<td>D-14</td>
</tr>
<tr>
<td>14) Ball fault unbalance</td>
<td>D-15</td>
</tr>
<tr>
<td>15) Ball fault load</td>
<td>D-16</td>
</tr>
<tr>
<td>16) Ball fault load and unbal</td>
<td>D-17</td>
</tr>
<tr>
<td>17) Misalignment no load or unbal</td>
<td>D-18</td>
</tr>
<tr>
<td>18) Misalignment unbalance</td>
<td>D-19</td>
</tr>
<tr>
<td>19) Misalignment load</td>
<td>D-20</td>
</tr>
<tr>
<td>20) Misalignment load and unbal</td>
<td>D-21</td>
</tr>
</tbody>
</table>
Inner Race Fault - Sensor AV - Unbalance (run 2)
Inner Race Fault - Sensor AV - Load + Unbalance (run 4)
Misalignment - Sensor AV - Load + Unbalance (run 4)

Velocity (VdB re 1e-8 m/sec rms)

Frequency (Hz)
APPENDIX E

CEPSTRUM FROM LOAD AND UNBALANCE CONFIGURATIONS
SPEED: 5000 RPM
SENSOR: AFT BEARING SUPPORT VERTICAL (AV)

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Baseline no load or unbal</td>
<td>E-2</td>
</tr>
<tr>
<td>2) Baseline unbalance</td>
<td>E-3</td>
</tr>
<tr>
<td>3) Baseline load</td>
<td>E-4</td>
</tr>
<tr>
<td>4) Baseline load and unbal</td>
<td>E-5</td>
</tr>
<tr>
<td>5) Inner race fault no load or unbal</td>
<td>E-6</td>
</tr>
<tr>
<td>6) Inner race fault unbalance</td>
<td>E-7</td>
</tr>
<tr>
<td>7) Inner race fault load</td>
<td>E-8</td>
</tr>
<tr>
<td>8) Inner race fault load and unbal</td>
<td>E-9</td>
</tr>
<tr>
<td>9) Outer race fault no load or unbal</td>
<td>E-10</td>
</tr>
<tr>
<td>10) Outer race fault unbalance</td>
<td>E-11</td>
</tr>
<tr>
<td>11) Outer race fault load</td>
<td>E-12</td>
</tr>
<tr>
<td>12) Outer race fault load and unbal</td>
<td>E-13</td>
</tr>
<tr>
<td>13) Ball fault no load or unbal</td>
<td>E-14</td>
</tr>
<tr>
<td>14) Ball fault unbalance</td>
<td>E-15</td>
</tr>
<tr>
<td>15) Ball fault load</td>
<td>E-16</td>
</tr>
<tr>
<td>16) Ball fault load and unbal</td>
<td>E-17</td>
</tr>
<tr>
<td>17) Misalignment no load or unbal</td>
<td>E-18</td>
</tr>
<tr>
<td>18) Misalignment unbalance</td>
<td>E-19</td>
</tr>
<tr>
<td>19) Misalignment load</td>
<td>E-20</td>
</tr>
<tr>
<td>20) Misalignment load and unbal</td>
<td>E-21</td>
</tr>
</tbody>
</table>
Cepstrum Analysis - Ball Fault - Sensor AV - Unbalance (run 2)
Cepstrum Analysis - Misalignment - Sensor AV - No Load or Unbal (run 1)
Cepstrum Analysis - Misalignment - Sensor AV - Load + Unbal (run 4)
# APPENDIX F

**DEMODULATED DATA FROM LOAD AND UNBALANCE CONFIGURATIONS**

**SPEED: 5000 RPM**

**SENSOR: AFT BEARING SUPPORT VERTICAL (AV)**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Baseline no load or unbal</td>
<td>F-2</td>
</tr>
<tr>
<td>2) Baseline unbalance</td>
<td>F-3</td>
</tr>
<tr>
<td>3) Baseline load</td>
<td>F-4</td>
</tr>
<tr>
<td>4) Baseline load and unbal</td>
<td>F-5</td>
</tr>
<tr>
<td>5) Inner race fault no load or unbal</td>
<td>F-6</td>
</tr>
<tr>
<td>6) Inner race fault unbalance</td>
<td>F-7</td>
</tr>
<tr>
<td>7) Inner race fault load</td>
<td>F-8</td>
</tr>
<tr>
<td>8) Inner race fault load and unbal</td>
<td>F-9</td>
</tr>
<tr>
<td>9) Outer race fault no load or unbal</td>
<td>F-10</td>
</tr>
<tr>
<td>10) Outer race fault unbalance</td>
<td>F-11</td>
</tr>
<tr>
<td>11) Outer race fault load</td>
<td>F-12</td>
</tr>
<tr>
<td>12) Outer race fault load and unbal</td>
<td>F-13</td>
</tr>
<tr>
<td>13) Ball fault no load or unbal</td>
<td>F-14</td>
</tr>
<tr>
<td>14) Ball fault unbalance</td>
<td>F-15</td>
</tr>
<tr>
<td>15) Ball fault load</td>
<td>F-16</td>
</tr>
<tr>
<td>16) Ball fault load and unbal</td>
<td>F-17</td>
</tr>
<tr>
<td>17) Misalignment no load or unbal</td>
<td>F-18</td>
</tr>
<tr>
<td>18) Misalignment unbalance</td>
<td>F-19</td>
</tr>
<tr>
<td>19) Misalignment load</td>
<td>F-20</td>
</tr>
<tr>
<td>20) Misalignment load and unbal</td>
<td>F-21</td>
</tr>
</tbody>
</table>
Baseline - Sensor AV - Unbalance (run2)

Solid - FFT
Dash - Demodulated Signal
Baseline - Sensor AV - Load + Unbalance (run 4)

Solid - FFT
Dash - Demodulated Signal
Inner Race Fault - Sensor AV - No Load or Unbal (run 1)

Solid - FFT
Dash - Demodulated Signal
Inner Race Fault - Sensor AV - Load + Unbalance (run 4)

Solid - FFT
Dash - Demodulated Signal
Outer Race Fault - Sensor AV - No Load or Unbal (run 1)

Solid - FFT
Dash - Demodulated Signal
Outer Race Fault - Sensor AV - Load + Unbalance (run 4)

Solid - FFT
Dash - Demodulated Signal

Frequency (Hz)

Velocity (m/sec)
Ball Fault - Sensor AV - Load + Unbalance (run 4)

Solid - FFT
Dash - Demodulated Signal
Missalignment - Sensor AV - Unbalance (run2)

Solid - FFT
Dash - Demodulated Signal
Misalignment - Sensor AV - Load + Unbalance (run 4)

Solid - FFT
Dash - Demodulated Signal
APPENDIX G

SINGULAR VALUE DECOMPOSITION FROM LOAD AND UNBALANCE CONFIGURATIONS

SPEED: 5000 RPM

SENSOR: AFT BEARING SUPPORT VERTICAL (AV)

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Inner race fault no load or unbal</td>
<td>G-2</td>
</tr>
<tr>
<td>2) Inner race fault unbalance</td>
<td>G-3</td>
</tr>
<tr>
<td>3) Inner race fault load</td>
<td>G-4</td>
</tr>
<tr>
<td>4) Inner race fault load and unbal</td>
<td>G-5</td>
</tr>
<tr>
<td>5) Outer race fault no load or unbal</td>
<td>G-6</td>
</tr>
<tr>
<td>6) Outer race fault unbalance</td>
<td>G-7</td>
</tr>
<tr>
<td>7) Outer race fault load</td>
<td>G-8</td>
</tr>
<tr>
<td>8) Outer race fault load and unbal</td>
<td>G-9</td>
</tr>
<tr>
<td>9) Ball fault no load or unbal</td>
<td>G-10</td>
</tr>
<tr>
<td>10) Ball fault unbalance</td>
<td>G-11</td>
</tr>
<tr>
<td>11) Ball fault load</td>
<td>G-12</td>
</tr>
<tr>
<td>12) Ball fault load and unbal</td>
<td>G-13</td>
</tr>
<tr>
<td>13) Misalignment no load or unbal</td>
<td>G-14</td>
</tr>
<tr>
<td>14) Misalignment unbalance</td>
<td>G-15</td>
</tr>
<tr>
<td>15) Misalignment load</td>
<td>G-16</td>
</tr>
<tr>
<td>16) Misalignment load and unbal</td>
<td>G-17</td>
</tr>
</tbody>
</table>
Singular Value Decomposition - Sensor AV - Unbalance (run 2)

- solid - baseline
- dash - inner race fault

amplitude vs. singular values
Singular Value Decomposition - Sensor AV - Load (run 3)

- solid - baseline
- dash - inner race fault
Singular Value Decomposition - Sensor AV - Load + Unbalance (run 4)

solid - baseline
dash - inner race fault
Singular Value Decomposition - Sensor AV - Load (run 3)

solid - baseline
dash - outer race fault
Singular Value Decomposition - Sensor AV - No Load or Unbal (run 1)

- solid - baseline
- dash - ball fault

amplitude

singular values
Singular Value Decomposition - Sensor AV - Load + Unbalance (run 4)

solid - baseline
dash - ball fault
Singular Value Decomposition - Sensor AV - Unbalance (run 2)

Amplitude

- solid - baseline
- dash - misalignment

Singular values
Singular Value Decomposition - Sensor AV - Load + Unbalance (run 4)

- solid - baseline
- dash - misalignment