

A QUICK INTRODUCTION TO BEARING ENVELOPE ANALYSIS

Eric Bechhoefer
Green Power Monitoring Systems
Cornwall, VT 05752
Telephone: (802) 377-0160
ebechhoefer@gmail.com

Abstract: Bearing envelope analysis (BEA) is a powerful technique for the detection of bearing faults. The improper selection of the envelope window frequency and window bandwidth can render the analysis ineffective. This can reduce the ability to perform condition monitoring to correctly identify a degraded bearing. This paper is an analysis of how BEA works in the detection of damage bearings. A description of the BEA is given, methods for window selection, such as spectral kurtosis, is described, and example algorithms are given to facilitate experimentation.

Key words: Bearing Analysis, Envelop, Heterodyne, and Condition Monitoring Systems;

INTRODUCTION: Rotating equipment in general and equipment such as wind turbines are dependent on a transmission to condition the power for useful work. In the case of wind turbines, the power of a high torque, low speed main rotor shaft is converted into a high speed, low torque generator shaft. Integral to the reliable operation of this transmission, are bearings. Reliability and availability of the wind turbine are improved if monitoring techniques are developed, which can detect when a degrading or faulty bearing requires maintenance. This is the essence of condition monitoring for bearings.

Because the vibration signals of a faulty bearing are small compared to shaft order and gear mesh frequency, detection of faults at the bearing rate frequencies using Fourier analysis is difficult. Fault detection at the baseband frequencies of the bearing rate is “Stage 1” fault detection. Bearing faults detected using these types of analysis are Late Stage; the bearing can be close to catastrophic failure. At the very least, a bearing in this state is generating metal which can cause damage to other components within the gearbox.

Ultrasonic emission can detect bearing inner and outer race roughness (a “Stage 3” fault), but the remaining useful life of a bearing at this stage is relatively long compared to the overall life of the bearing. Bearing envelope analysis (BEA) can typically detect bearing faults 100s if not 1000s of hours prior to when it is appropriate to do maintenance. It is for this reason that many condition monitoring systems manufacturers are using envelope analysis techniques.

This paper reviews the envelope algorithm and describes techniques for developing the envelope signal. In addition, methods for determining the optimal window for envelope analysis are provided. Finally this paper describes the accompanying bearing fault data

set to allow researchers to develop more robust bearing analysis techniques are methods to threshold the spectrum of the envelope analysis.

THE BEARING ENVELOPE ANALYSIS: BEA is based on demodulation of high frequency resonance associated with bearing element impacts. For rolling element bearings, when the rolling elements strike a local fault on the inner or outer race, or a fault on a rolling element strikes the inner or outer race, an impact is produced. These impacts modulate a signal at the associated bearing pass frequencies, such as: Cage Pass Frequency (CPF), Ball Pass Frequency Outer Race (BPFO), Ball Pass Frequency Inner Race (BPFI), and Ball Fault Frequency (BFF). Figure (1) is an Outer Race Fault, where the BPFO is approximately 80 Hz. Note that the modulation rate, T1, is approximately .0125 seconds (e.g. 1/80 Hz). The time T2, the period of the resonance, is approximately 1.12e-4 seconds, or about 9000 Hz. Note that the time domain representation is the super position of many resonances of the bearing itself.

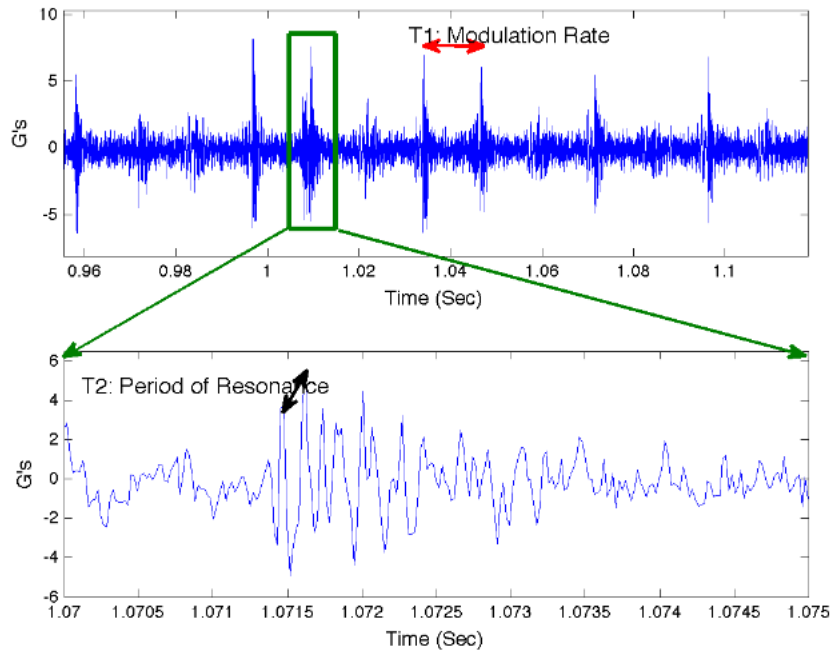


Figure 1 Example Outer Race Fault

Mathematically, the modulation is described as:

$$\cos(a) \bullet \cos(b) = \frac{1}{2} [\cos(a+b) + \cos(a-b)] \quad (1)$$

This is amplitude modulation of the bearing rate (a) with the high frequency carrier signal (resonant frequency (b)). This causes sidebands in the spectrum surrounding the resonant frequency. It is sometimes difficult to distinguish the exact frequency of the resonance. It is usually not known *a priori* and cannot be determined easily without a faulted component. However, demodulation techniques typically do not need to know the exact frequency. One method for the BEA involves multiplying the vibration signal by a

resonant frequency (example, 9 kHz). This is then low pass filtered to remove the high frequency image, decimated, and the spectral power density is estimated. (Eq 2)

$$\begin{aligned} \cos(b) \bullet \cos(a+b) &= \\ \frac{1}{2} [\cos(a+b+b) + \cos(a+b-b)] &= \\ \rightarrow H(\omega) \rightarrow \cos(a) & \end{aligned} \quad (2)$$

An alternate way to generate the envelope is to take the FFT (fast Fourier transform) of the time domain signal and calculate the frequency of each bin (e.g. sample rate/length of time series). Calculate the “window” indexes (e.g. the range of frequencies in which the bearing is resonating, say 8 to 9 kHz). Copy the Fourier coefficients from desired window (e.g. 8kHz to 9 kHz) into the base band (e.g. 0 to 1 kHz) indexes (this is a Heterodyne operation). Then copy zeros into the remaining Fourier coefficients (Fourier filtering and Hilbert transform, recalling that the Hilbert transform is defined at the real frequency of the Fourier domain). Finally, take the absolute value of inverse FFT Fourier coefficients (see Appendix for Matlab© example).

The bearing components have a number of vibration modes, which will correspondingly generate resonance at various frequencies throughout the spectrum. The selection of the frequency range used to demodulate the bearing rate signal (e.g. the window center frequency) should take into account some issues: First, the gearbox spectrum contains a number of high-energy frequencies from shaft and gear harmonics, which would mask analysis at lower bearing frequencies. Second, there are a number of accelerometers with natural resonance at frequencies that are similar to the bearing modes. Using a higher frequency window close to the accelerometer resonance can amplify the bearing fault signal, increasing the probability of fault detection.

BEA should be performed at frequencies higher than the shaft and gear mesh frequencies. This ensures that the demodulated bearing frequencies are not masked by the other rotating sources, such as shaft and gear mesh, which are present at CPF, BPFO, BPFI and BFF frequencies. Typical shaft order amplitudes of 0.1 G's and gear mesh amplitudes of 10s of G's, are common. Damaged bearing amplitudes are 0.001 G's.

A technique used to identify an appropriate envelope window is Spectral Kurtosis. Boskoski and Urevc used spectral kurtosis (SK) to determine the best window in order to determine the condition of a test gearbox with seeded faults. Dr. Robert Bond Randall's book Vibration-based Condition Monitoring gives an excellent explanation of SK.

Kurtosis is a non-dimensional quantity that measures the relative “peakedness” of a distribution relative to the Gaussian distribution. Spectral kurtosis (SK) is a statistical parameter indicating how the impulsiveness of a signal varies with frequency. As noted, faults associated with rolling element bearings give rise to short impulse. The SK will be large in frequency bands where the fault signal is dominant and small where the spectrum is dominated by stationary signals. Antoni developed the kurtogram, which is a map indicating the optimum center frequency and bandwidth combination.

In this kurtogram example, filtering was limited between 1 kHz and 24.4 kHz (Nyquest). The lower limit was chosen because typically, below 1 kHz, there are significant gear

mesh frequencies. The kurtogram frequency space spans eight octaves, such that frequency band is approximately halved with each increase in octave:

TABLE I. KURTOGRAM FREQUENCY MAP

Octave	Bandwidth	Number of Bands	PSD Window
1	12.2 kHz	2	37768
2	8.1 kHz	3	16384
3	6.1 kHz	4	8192
4	4.1 kHz	6	4096
5	3.05 kHz	8	4096
6	2.03 kHz	12	2048
7	1.5 kHz	16	1024
8	0.76 kHz	32	512

Figure 2 shows the raw spectrum and the Kurtogram. The largest kurtosis value was in Octave 5, or a window of 1 to 4 kHz. This is compared to the second example window in Octave 8, from 2.51 kHz to 3.266 kHz.

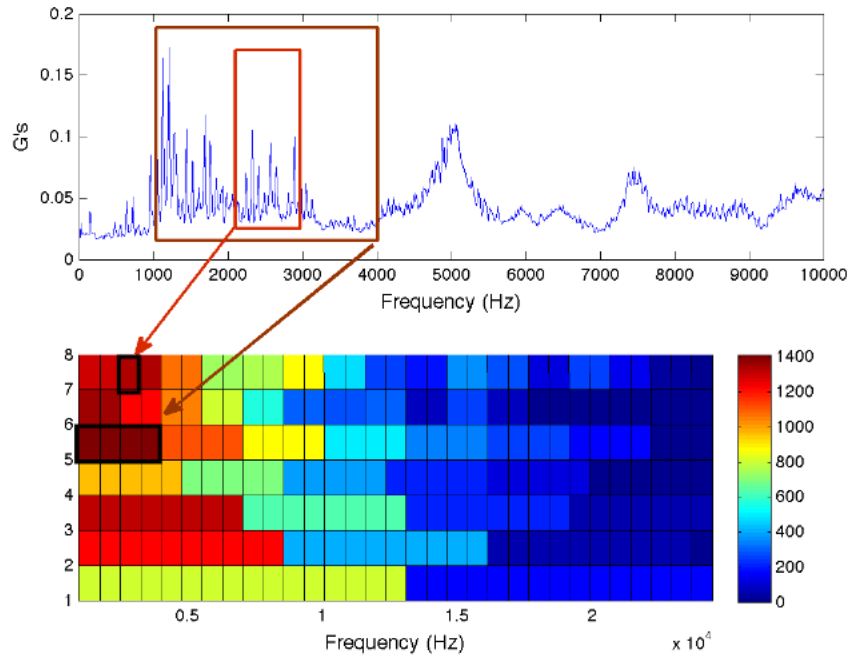


Figure 2 Spectra and Kurtogram

The envelope analysis spectrum for Window 1 and Window 2 is plotted in Figure 3. Note that a 3rd envelope analysis is also demonstrated for a poor window: 15 to 18 kHz (band

5, Octave 4). The SNR for the Window 1 was 23 dB, Window 2: 15 dB, and Window 3: 10 dB. There have been a number of reported cases where poor window selection has resulted in a missed detection of a faulted bearing.

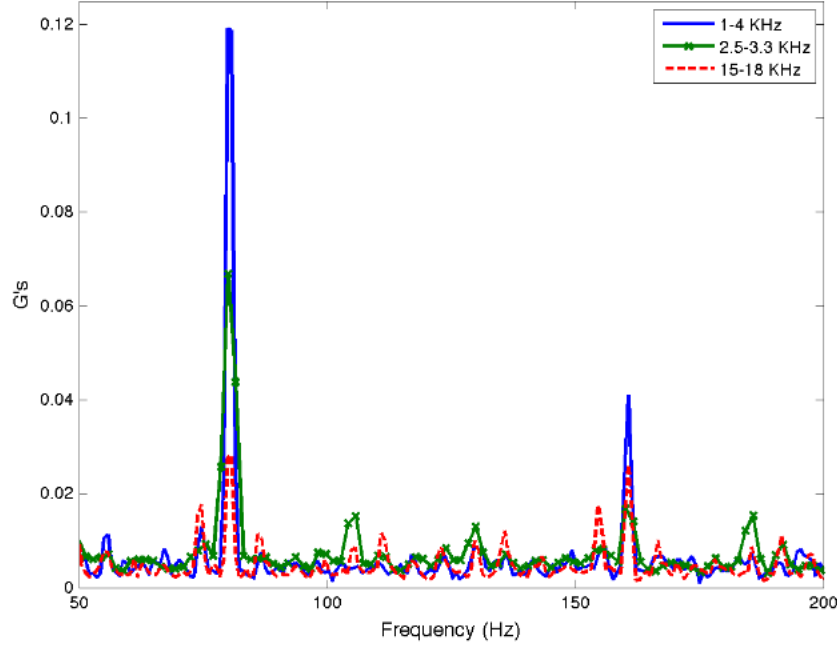


Figure 3 Envelope Spectra

CALCUALTION OF THE BEARING RATES: For BEA, the bearing rates must be known in order to calculate the energy in the spectrum associated with the fault. The bearing rates are derived from the bearing geometer and are calculated as:

Cage Pass Frequency (CPF):

$$\frac{f}{2} \left(1 - \frac{d}{e} \times \cos(\beta) \right) \quad (3)$$

Ball Pass Frequency Inner Race (BPFI):

$$\frac{b \times f}{2} \left(1 + \frac{d}{e} \times \cos(\beta) \right) \quad (4)$$

Ball Pass Frequency Outer Race (BPFO):

$$\frac{b \times f}{2} \left(1 - \frac{d}{e} \times \cos(\beta) \right) \quad (5)$$

Ball Fault Frequency (BFF):

$$\frac{e \times f}{d} \left(1 - \left(\frac{d}{e} \right)^2 \times \cos^2(\beta) \right) \quad (6)$$

where,

f	is the driving frequency
b	is the number of rolling elements
d	is the ball bearing diameter
e	is the bearing pitch diameter
β	is the bearing contact angle

The amplitude associated with the bearing rates frequency can be used as a condition indicator (CI), which reflects the “health” of the bearing components. Note that the rate does not reflect “slippage”, which can be 1% or more.

Conclusion: Bearing Envelope analysis is a powerful analysis method for the detection of bearing fault. Window selection is important – a poor window may not detect a bearing that is in the process of failing. Window selection can be guided by the use of spectral kurtosis, or by selecting part the spectrum where the bearing under analysis would have a resonant response.

Appendix:

- 1 - Example of Spectral Kurtosis Algorithm
- 2 - Example of Welch’s Method for the Calculation of the Spectrum
- 3 - Example of Envelope Method 2

Appendix:

1 - Example of Spectral Kurtosis Algorithm

```
function S = SpecKurt(v,sr)
%S = SpecKurt(v,sr)
%Inputs:
%   v   :time domain values
%   sr  :sample rate
%Outputs:
%   S   :Kurtogram
%Spectrum is calculated using psde

lvl = [2 3 4 6 8 12 16 32]; %Octave Levles

lw = 1000;%lowerest window frequency value
hi = sr/2;%Nyquest
n = length(lvl);
S = zeros(n,32);
for i = 1:n,
    cLvl = lvl(i);
    dBw = (hi-lw)/cLvl;
    cLw = lw;
    dix = floor(32/cLvl);
    sdix = 1;
    edix = dix;
    for j = 1:cLvl,
        cHi = cLw + dBw;
        [y,dt] = envhetsv(v,1/sr,128,cLw,cHi);
        [P,frq] = psde(y, 4096,1/dt, 2048);
        sk = kurtosis(P);

        idx = sdix:edix;
        S(i,idx) = sk;
        sdix = edix + 1;
        edix = edix + dix;
        cLw = cLw + dBw;
    end
end

fq = linspace(lw,hi,32);
surf(fq,1:8,S)
axis([fq(1) fq(end) 1 8])
% title(name);
colorbar
```


2 - Example of Welch's Method for the Calculation of the Spectrum

```
function [Spec, freq] = psde(x, winln, Fs, noverlap)
%[Spec, freq] = psde(x, winln, Fs, noverlap);
%Inputs:
% x          :time domain data
% winln      :window length, e.g. 2048,
% Fs         :sampling frequency
% noverlap   :length of the overlap
%Outputs:
% Spec       :vector Spectrum
% freq       :corresponding frequency

n = winln;
m = n/2;
window = .5*(1 - cos(2*pi*(1:m)/(n+1))); %Hann window
window = [window; window(end:-1:1)];
window = window(:);

nfft = length(window);

n = length(x);          % Number of data points
nwind = length(window); % length of window
if n < nwind             % zero-pad x if it has length less than
the window length
    x(nwind)=0; n=nwind;
end
% Make sure x is a column vector; do this AFTER the zero-padding
% in case x is a scalar.
x = x(:);

k = fix((n-noverlap)/(nwind-noverlap)); % Number of windows
index = 1:nwind;
%KMU = norm(k*window)^2;    % Normalizing scale factor ==>
asymptotically unbiased
    KMU = k*sum(window)^2;% alt. Normalizing scale factor ==> peaks are
about right

Spec = zeros(nfft,1);
for i=1:k
    xw = window.*detrend(x(index));
    index = index + (nwind - noverlap);
    xx = fft(xw,nfft);
    Xx = abs(xx).^2;
    Spec = Spec + Xx;
end

% Select first half
if ~any(any(imag(x)~=0)), % if x is not complex
    if rem(nfft,2), % nfft odd
        select = (1:(nfft+1)/2)';
    else
        select = (1:nfft/2)';
    end
    Spec = Spec(select);
else
    select = (1:nfft)';
end
```



```

freq = (select - 1)*Fs/nfft;

% find confidence interval if needed

Spec = sqrt(Spec*(4/KMU)); % normalize: ow in Energy vs. Power

```

Example of Envelope method 1

```

function [env,dtty] = envelopel(data,dt,nfilt,lowf,highf)
% [env,dtty] = envelopel(data,dt,nfilt,lowf,highf);
%Inputs:
% data      :data vector, time domain
% dt        :sampling time interval
% nfilt     :size of filter for low pass filtering
% lowf      :low frequency limit of bandpass filter
% highf     :high frequency limit of bandpass filter
%Outputs:
% env       :Envelope of data
% dtty      :decimated sample rate

c = (highf + lowf)/2;
if 1/dt/2 < c
    y = [];
    dt = [];
else
    ndata = length(data);
    z = data(:) .*exp(-2 * pi * li * c * dt * (0:ndata-1)');
    bw = highf - lowf;
    b = fir1(nfilt,bw*dt);
    x = filter(b,1,z);
    r = fix(1/(bw*2*dt));
    env = abs( x(nfilt+1:r:ndata) ); % crop first nfilt
    elements

    dtty = dt*r;
end

```

3 - Example of Envelope Method 2

```
function [env,dtty] = envelope2(data,dt,lowf,highf)
% [env,dtty] = envelope1(data,dt,nfilt,lowf,highf);
%Inputs:
% data      :data vector, time domain
% dt        :sampling time interval
% lowf      :low frequency limit of bandpass filter
% highf     :high frequency limit of bandpass filter
%Outputs:
% env       :Envelope of data
% dtty      :decimated sample rate

n = length(data);
dfq = 1/dt/n;
idxLow = floor(lowf/dfq);
idxHi = ceil(highf/dfq);
D = fft(data);
idx = idxHi-idxLow + 1;

D(1:idx) = D(idxLow:idxHi);
D(idx+1:end) = 0;
data = abs(ifft(D));
bw = highf - lowf;
r = fix(1/(bw*2*dt));
env = data(1:r:n);    % crop first nfilt elements

dtty = dt*r;
```