Applying the Inverse FFT for Filtering, Transient Details and Resampling

Howard A. Gaberson, Oxnard, California

The discrete Fourier Transform (DFT) can be developed from the Fourier series, which results in the most useful form for machinery diagnostics. This form presents the transform of an even number of data values as the amplitude and phase of a set of (N/2 + 1) sine waves or N complex exponentials. This point of view presents the inverse DFT (IDFT) conceptually as the addition of correctly phased discrete complex exponentials. The inverse Fast Fourier Transform (IFFT) is needed to perform the 'add' because of the economies it employs. The complex exponentials defined by the DFT can be viewed as continuous, but when evaluated at times corresponding to the original data samples and added together, they exactly duplicate the original data sequence. The sum of the complex exponentials form a continuous band limited curve exactly passing through the data points. This article explains this point of view and applies it to transient detection, filtering, resampling and bin centering.

The Fourier Transform (FT) has a myriad of uses. Some of them are central to machinery diagnostics. Hidden periodicities are the key – those repeated or cyclic parts of a machinery vibration signal. Their characteristics can be a clue for diagnosing machine condition. "What is cyclic?" It is a fabulous sales angle. Good pitch charlatans have been snowing us with it forever: ". . . I've gone back and investigated the price of oober goobers over the last couple of centuries and have noticed a cyclic pattern, repeated maybe 12 or 13 times. It is a wonderful opportunity to get rich if you can discern this complicated pattern, which I know all about. Now is the perfect time to invest, and if you'll give me all your money, I'll make you rich."

How to Look at the DFT

Anyway, back to our Fast Fourier Transform (FFT), DFT really. The algorithm to calculate it does not matter as long as you perform Eqs. 1 and 2. I taught myself the DFT starting from the complex exponential form of the Fourier series as Newland[1] suggests. Equally space digitize the signal and straight line numerically plan the integration to get each coefficient, and you end up with:

$$X_k = \frac{1}{N}\sum_{n=0}^{n=N-1} x_n e^{\frac{-i2\pi kn}{N}} \tag{1}$$

Out of thin air pluck this similarly indexed synthesis equation

$$x_n = \sum_{k=0}^{N-1} X_k e^{\frac{i2\pi nk}{N}} \tag{2}$$

and try it out. Insert the value of the X_k from Eq. 1 into Eq. 2, carefully keeping track of the indices, and by using the algebra formula for the sum of a geometric series,[2] you find that they have to work exactly. Eq. 1 transforms a list of x's into a list of complex X's, and Eq. 2 exactly transforms the complex X's back to the original **x**'s. Amazing! But notice, the $1/N$ is in the position that results from the complex Fourier series derivation, not where it is usually placed.

To use Eqs. 1 and 2 for vibration analysis there are some ground rules. The data list is sampled at sampling rate f_s. The x_n of Eq. 1 are samples from a signal that was accurately sampled and band limited to $f_s/2$; this means its Fourier Transform is zero for all frequencies greater than half the sampling rate. There is an even number of N samples in the list. The time interval between the samples is:

$$h = \frac{1}{f_s} \tag{3a}$$

In Eqs. 1 and 2, the complex exponential is really:

$$e^{-\frac{i2\pi kn}{N}} = \cos\frac{2\pi kn}{N} - i\sin\frac{2\pi kn}{N} \tag{3b}$$

A sine wave can be written as $\cos 2\pi ft$, where f is the frequency and t is time. In the $2Bkn/N$ of Eq. 3b, if we multiply it by hf_s (which equals 1, by Eq. 3a), we group the terms of the cosine and the exponential as in Eq. 3c:

$$\cos\left(2\pi\frac{kf_s}{N}nh\right),\ e^{-12\pi\frac{kf}{N}nh} \tag{3c}$$

Comparing this with $\cos 2\pi ft$, in digitized terms, nh is the discrete time, and kf_s/N is the frequency.

Think of the DFT as an exact transformation of a digitized vibration signal. It transforms the data into discrete (or sampled) complex exponentials (or equivalently, sinusoids). The transform is the list of their amplitudes as a function of frequency. Eq. 1 transforms the signal into N sine waves; each X_k is the amplitude and phase of a complex sine wave. Figure 1 attempts to show one of the k discrete complex sinusoids in 3D. X_k is its complex amplitude, shown as a vector from the origin to the beginning point of the discrete spiral. The little circles on the spiral represent the values of the discrete sinusoid for the sequence of n values. The smooth spiral on which the data lie is part of the underlying curve, the curve with time taken to be continuous or with nh replaced by t in Eq. 3c. Each X_k is the complex amplitude of one of the discrete spirals. By Eq. 2, their summation yields the original signal. The transform is the list of complex X's. Since it is an exact transformation, we are able to exactly inverse transform the DFT back to the original data. The transform is a set of amplitudes and phases of complex sine waves whose summation forms a continuous curve. When the curve is evaluated at the signal sampling instants, it exactly reproduces the signal.

The continuous curve, which is the sum of N sine waves, is the continuous periodic band limited curve from which the original x's were sampled. The inverse transform evaluates all these sine waves at the signal sample instants and adds them. I will use the word reconstruction for the inverse transform operation, the IFFT.

The DFT is most economically computed using the FFT. I believe MATLAB's® algorithm is able to deliver efficient results for all N values, not just powers of 2. The FFT will compute N Fourier coefficients from a sequence of N numbers. The Xs can be numbered from 0 to N–1. If N is even, X_0 is the DC or average value; it is real and contains one value. $X_{N/2}$ also contains one real value; it is special. It is the content at the Nyquist frequency or half the sampling rate. It turns out to be the sum of the sequence with alternate signs reversed. In between these two are $N/2$–1 unique complex values, each containing two values. The X values from $X_{N/2+1}$ to X_{N-1} are not unique, but are complex conjugates of the values from $X_{N/2-1}$ down to X_1. The values symmetrical about $X_{N/2}$ are complex conjugate pairs. Thus for the N values of the sequence, we get N unique values from the transform.

Relation of DFT *X*'s to Harmonic Content or Spectrum

If you find time to look up Fourier Series theory from any old book (25 years or so), you will find that they use a's and b's for the analysis of a segment of a signal, assumed periodic.

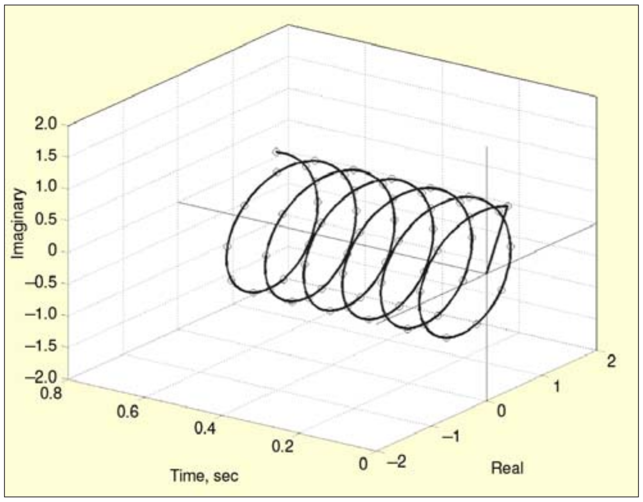

Figure 1. This represents one of the discrete complex sinusoids in 3-D. The straight line vector from the origin represents, X_k. The little circles on the spiral beginning at the tip of the straight line are the values for the sequence of n values.

(However, the analysis will work almost no matter what you assume; it will make a periodic function of that segment.) Then they reconstruct the signal in terms of these *a*'s and *b*'s as follows. If $x(t)$ is periodic with period T, it can be exactly synthesized from its Fourier series coefficients, a_k, b_k:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left(a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T}\right) \tag{4a}$$

for k = 0, 1, 2,...,

The Fourier series coefficients are given by:

$$\frac{a_k}{2} = \frac{1}{T}\int_o^T x(t)\cos\frac{2\pi kt}{T}dt \tag{4b}$$

$$\frac{b_k}{2} = \frac{1}{T}\int_o^T x(t)\sin\frac{2\pi kt}{T}dt \tag{4c}$$

and

The a_k and b_k are the harmonic content. The k[th] harmonic is given by:

and

$$x_k(t) = a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T} \tag{5a}$$

Its frequency is k/T. We can also write the k[th] harmonic in terms of amplitude and phase:

$$x_k(t) = A_k \cos\left(\frac{2\pi kt}{T} - \phi_k\right) \tag{5b}$$

Here, A_k is the amplitude and ϕ_k is the phase. The phase is the angle in radians to the first positive peak. The amplitude and phase are given by:

$$A_k = \sqrt{a_k^2 + b_k^2} \quad \text{and} \quad \phi_k = \tan^{-1}\left(\frac{b_k}{a_k}\right) \tag{5c,d}$$

A_k is the content or the amplitude of the k[th] harmonic or tone; ϕ_k is its phase. This is the quantity shown on signal analyzers and data collectors as the spectrum. More manipulating leads us to the following. The X_k's from k = 1 . . . (N–1)/2 that we get from the DFT are related to the content as:

$$X_k = \frac{a_k}{2} - i\frac{b_k}{2} \tag{6a}$$

$$A_k = 2|X_k| \qquad \phi_k = \tan^{-1}\left(\frac{\mathrm{Re}(X_k)}{-\mathrm{Im}(X_k)}\right) \tag{6b,c}$$

More Samples by Adding Zeros to the Transform

It may be helpful to note that the signal is reconstructed (inverse transformed) by adding up values of complex sine waves. The IFFT very efficiently adds up the samples at the desired, equally spaced time instants to return the original sampled signal.

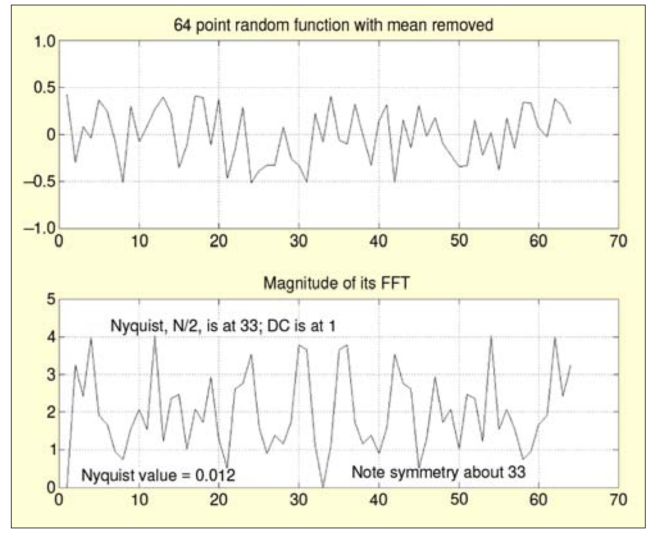

Figure 2. The upper curve is a string of 64 random values selected by the MATLAB command 'rand' with the mean removed. The bottom half of the figure is the magnitude of its FFT as returned by MATLAB.[3]

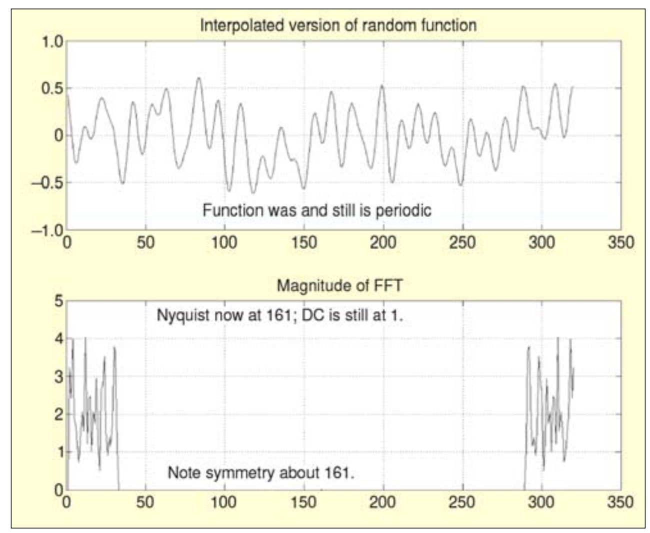

Figure 3. The lower half of the figure shows the modified transform magnitude with 255 zeros and a half Nyquist value added in the center. The upper half is the inverse transform of this zero padded transform. It shows a more detailed view of the underlying continuous curve defined by the original samples.

However, we can add high frequency zeros to the transform because the signal was band limited. This forces the IFFT to compute equally spaced samples of our band limited function at a higher sampling rate; the number of values in the transform sets the number of signal values the IFFT computes. This is not interpolating, as I have heard commented; it actually calculates more values of the true band limited function defined by the original data. The values are exact and contained in the continuous sum of the complex sine waves curve; there is an infinite number of values available. I will not deal with fewer samples, because doing so involves fewer frequency components. You would have to discard the higher frequency sine waves, which would in effect be decimating.

I am going to plot some spectra using the magnitude of the X's, $|X_k|$, as opposed to plotting the content, A_k, as defined in Eqs. 5b, 5c and 5d. Content is what we use in signal analyzers and data collectors. The X's (which the FFT computes) have values for *k* values or frequencies either side of Nyquist. The magnitudes must be symmetrical about the Nyquist frequency as explained above. We need to look at the X's for reconstruction.

To take a closer look, consider an arbitrary 64 value signal in MATLAB. The top half of Figure 2 is a plot of 64 values of a random sequence with the mean removed. The bottom half of

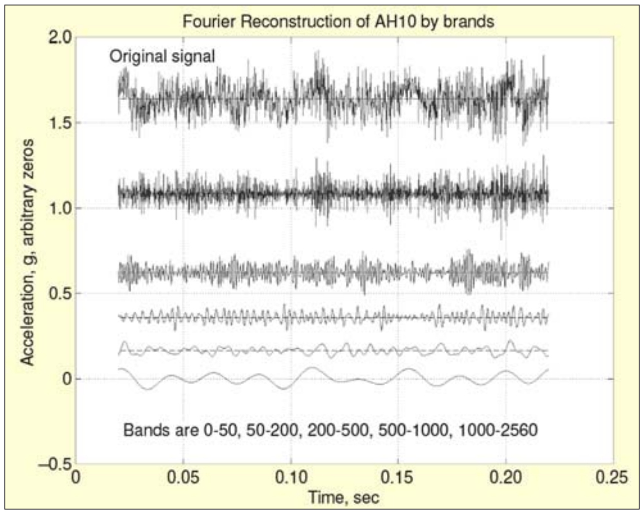

Figure 4. This shows a band wise partial reconstruction of a signal with an arbitrary band pass arrangement. Each signal segment contains content from only the frequencies indicated. Values of the transform outside the frequencies indicated were made zero prior to inverse transforming or reconstruction. Notice that the sums of components within the bands show various transient details.

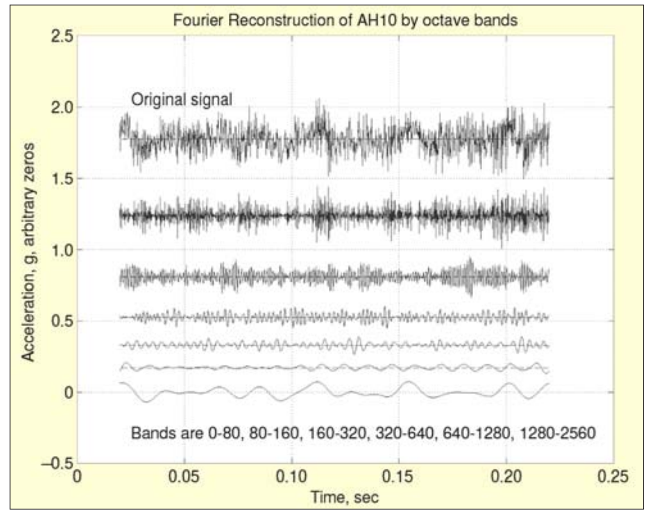

Figure 5. This also shows a partial reconstruction with the bands having an octave relationship. Again transient details can be seen. Notice that the sixth reconstruction from the bottom is a high pass filtering of the data, the bottom reconstruction is a low pass filtering and the reconstructions in between are band pass filterings of the data. Each of these filterings are brick wall filtering with no roll off.

Figure 2 shows the magnitude of its FFT, (as MATLAB returns it, without putting the $1/N$ where I like it). Notice the symmetry about the Nyquist value at index 33. You can exactly calculate the harmonic content without using the X values from $X_{(N/2+1)}$ to X_{N-1}, but the IFFT of Equation 2 requires all of the X values.

Now let us add some zeros to force a more detailed plot of this band limited function (band limited because of taking its FFT). I did this in MATLAB and plotted the results in Figure 3. Notice the similarities and differences between Figure 2a and Figure 3a. My thinking on adding zeros is as follows. We want to add only high frequency zeros. We want to add an even number of values to make it cheaper to compute, and because I have not thought out the theory for an odd number of values. Nyquist, the $(N+1)^{\text{th}}$ value and highest frequency value, is real. Half of its value should be placed symmetrically on either side of the new zero Nyquist value as the highest frequency non zero values (add a complex zero to each half Nyquist). Then place an odd number of complex zeros in the center of the transform to attain the desired number of values in the reconstruction. For the plot, I added 255 complex zeros between the two new half-original Nyquist values prior to performing the IFFT. The transform had 64 values, and I added 255 zeros and one complex half Nyquist so now it has 5 times as many values. We will get 5 times as many samples back after performing the IFFT. I had to multiply the answer by 5 for the following reason. MATLAB's FFT equations are given in Equations 7a and 7b

$$X_k = \sum_{n=1}^{n=N} x_n e^{\frac{-i2\pi(k-1)(n-1)}{N}} \qquad x_n = \frac{1}{N}\sum_{k=1}^{N} X_k e^{\frac{i2\pi(n-1)(k-1)}{N}} \qquad (7a,b)$$

The original X_k were calculated with N_{orig} values. But now when we IDFT the zero padded transform (Eq. 7b) with N_{long} values, the algorithm will divide each time history value by N_{long} when it should only have been divided by N_{orig}. Thus we have to multiply the values of our interpolated time history by the factor $N_{\text{long}}/N_{\text{orig}}$. In our case, N_{long} is 320 and N_{orig} is 64; 320/64 = 5; hence we multiply by 5. Figure 3a, the resulting IFFT, is a much more detailed plot of the true band limited function defined by the original set of data, Figure 2a. Adding zeros is a sensible practical procedure.

Partial or Bandwise Reconstruction

Eq. 2 can be considered a reconstruction of the original data by adding up its harmonic components. Since the reconstruction is built from sine waves, the details of any transients have to be assembled from a build up and cancellation of the separate harmonics. If one knew which sine waves built a particular transient, one could reconstruct or add only selected harmonics to accentuate the transient and inspect its details. Intuition as to how the reconstruction assembles transient effects in the time domain can be developed by experimenting with a partial reconstruction where major portions of the transform are zeroed out.

I did two tests of band-wise reconstruction with octave groups of bin values and arbitrary bin groupings. Figure 4 shows a 5-band reconstruction of some air handler pillow block acceleration data. Figure 5 shows the same data reconstructed in octave bands. Once this is programmed, it is a trivial matter to change the bands. By noticing the differences between Figure 4 and Figure 5, one can see that experimenting with different groups of bands will bring out otherwise unnoticed signal characteristics. The segment contained 1024 samples; no window was used because I planned a reconstruction. The bottom reconstruction on each figure is a low pass filtering of the data. The uppermost band, just below the original signal, is a high pass filtering of the data. Each of the remaining band-wise reconstructions is a band pass filtering of the signal. All of these filterings are done with brick wall filters.

Reconstruction Eliminating Smallest Harmonics

Reconstruction by eliminating bands with amplitude magnitudes in excess of or less than a level, such as 5 or 10 percent, is another procedure that might have value. I believe this is done in wavelet 'compression' to transmit an image with less data. To illustrate reconstructing by selecting only the strongest or weakest Fourier coefficients, I analyzed a simulated string of impact signals. I simulated some repetitious impacts with 7 percent damped exponentially decaying sine waves. The ring-down frequency in this case was 2394 Hz, the time interval between impacts was 102/51,200 = 0.002 sec; the reciprocal of this value is the harmonic spacing of the peaks or 51,200/102 = 501.9608 Hz. Figure 6b shows 3000 samples of the 8192 sample segment analyzed. I analyzed the signal with a boxcar window and noticed that the spectrum contains a sequence of harmonics spaced at the impact frequency with a significant DC component. In this case, 35 of them are over 10 percent of the peak signal and they occur as harmonics of the pulse frequency. Figure 6c shows the reconstructed signal from the harmonics over 10 percent, and Figure 6a shows the reconstruction from the remainder of the smaller harmonics. The most striking observation to me is that the components that fell in the lower 10 percent take care of the end effects. Maybe they help handle the beginning and end discontinuity. It would be

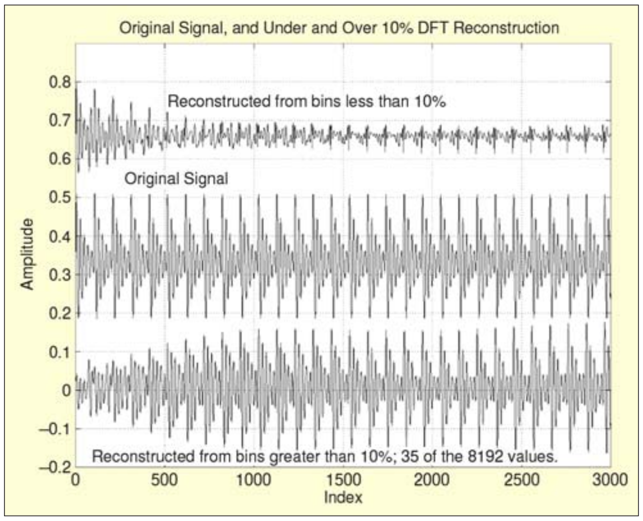

Figure 6. Reconstruction based on harmonic or transform magnitude. The middle signal is a simulated sequence of decaying exponentials. The lower figure illustrates reconstruction or inverse transformation using only Fourier components that are greater than 10% of the maximum Fourier transform magnitude. The top figure portion shows reconstruction using Fourier components with magnitudes less than 10% of the maximum Fourier transform magnitude.

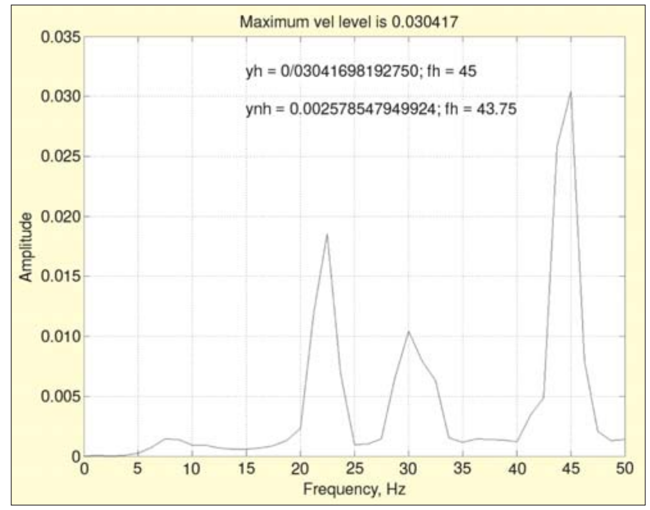

Figure 7. An expanded portion of the spectrum of air handler vibration data. FFT size: 4096; f_s = 5120/sec; Δf = 1.25 Hz; Hanning window; 6 averages. The complete spectrum has values all the way to 2560 Hz in 2049 values. Here the first 41 values are plotted to show how a typical spectrum appears when the tones are not bin centered.

interesting to resample the signal so that it was exactly periodic in the time window; I have not found time to do that.

Incidentally looking at the spectrum of the signal, the peak harmonic occurs closest to the ring-down frequency of the decaying exponential. To think about the compression ideas, I analyzed 8192 samples – 8157 of them were under 10 percent, 35 were over. Even though the harmonics are complex conjugates, we get an odd number, which is due to the DC component. Thus, if I wanted to send you the signal shown in Figure 5c, I could do so with 35 complex numbers along with where they are to be placed in a list of 8192 complex zeros. You could IFFT that spectrum and obtain Figure 5c.

Manipulating and Resampling to Bin Center a Tone

Resampling could be important to try to see the 'true' spectrum. Leakage, picket fence and the discreteness of the spectrum will creep in and contaminate or obscure my reading of the amplitude, frequency and phase of the tone I need for diagnosis. If I do an FFT of my data with a Hanning window, I can apply a frequency correction factor (see the Appendix) and calculate a good estimate of the true frequency. By resampling I can make that tone bin-centered and then re-FFT to see if the estimate is correct. After bin centering I can apply any window to reduce the leakage. Another reason for resampling might be to do after-the-fact time synchronous averaging. This would require a situation where I could assume the shaft speed remains constant over that interval.

Prior to digging into this material, I had used two methods for resampling, which I will describe here. To resample, one needs values of the data at a list of time instants that are different from the time instants at which the data were originally sampled. By analyzing digitized data at all, we have to believe it is band limited and has no frequency components greater than half of the sampling rate. MATLAB has two functions that are helpful in resampling – interp.m and decimate.m. MATLAB promises to have programmed these correctly in accordance with the authoritative IEEE bible,[4] and I have to trust them. That being the case, one resampling procedure I have used is to interpolate my signal by a huge factor of 100 and just select the nearest neighbor as the value for each new selected sampling instant. Another method I have used is to again interpolate, by at least 20, so as to feel confident that a straight line between any two interpolated points is a good approximation of the signal. Then I interpolate the values of the signal with a straight line for each of the desired instants. In each case, after the resampling, I decimate down to a reasonable sampling rate.

As I said, I now want to propose two additional methods based on a better understanding. But first let me drive home the point and help you make the leap I suggested in the paragraph after Eq. 3c. From my reading of, and belief in, the sampling theorem,[5] I am absolutely convinced of Shannon's statement: "There is one and only one function whose spectrum is limited to a band W, and which passes through given values at sampling points separated by $1/2W$ seconds apart." Here he means $W = F_s/2$ (thus $1/2W$ is $1/F_s$.). The signal must be properly band-limited and accurately digitized. I suggest you convince yourself that Shannon's statement is true. Think with me now, (and this was hard for me), only one curve can pass through our samples and be band limited. No one will ever think of another one. Look again at Eq. 2; it is the sum of N discretely evaluated continuous complex sine waves. The sum of the N complex continuous curves is a continuous curve going through the sample points. The sum of the N complex sine waves defined in Eq. 2 but with continuous time inserted as follows is the answer. Using the idea from Eq. 3c in Eq. 2 we have (Eq. 8):

$$x(t) = \sum_{k=0}^{N-1} X_k e^{\frac{i2\pi k f_s}{N} t} \tag{8}$$

This is a continuous curve implied by Eq. 2, and it is certainly band limited; it is the one and only curve.

The exact value of the curve at any instant is available. Thus samples at any desired instant or sampling rate greater than the existing sampling rate are available. The fact that the signal is reconstructed (inverse transformed) by adding up values of the N complex sine waves is a good way to look at the situation. That's a lot of sine wave evaluations. The IFFT is a very efficient way to add up the samples at the desired equally spaced time instants as in the original sampled signal. However, we can add zeros to the transform, which is in effect forcing the IFFT to compute equally spaced samples of the band limited function, Eq. 8, at a higher sampling rate. The number of values in the transform sets the number of signal values in the IFFT reconstruction.

I tested this idea in two ways, and it seems to work fine. I performed a six average, 25% overlapped, Hanning windowed, 4096 point FFT of my air handler data file, AH10C1. An expanded portion of the spectrum in the 0-50 Hz region is shown in Figure 7. This is expanded so much that you can see corners in the spectrum drawing at the individual bin values. The data were sampled at 5120/sec. The frequency spacing in a DFT is $1/T$; in the case:

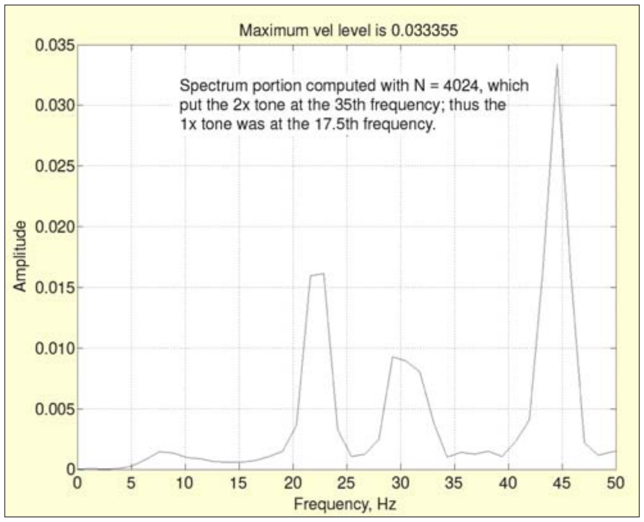

Figure 8. Result of bin centering by changing the FFT length to 4024. This first attempt shows a poor choice of FFT length. The magnitude of the peak just under 45 Hz for the signal 2×, misalignment component shows the true value of the tone. However, because this was the 35th frequency, the 1× component at half this frequency lies between the 17th and 18th frequencies, and is badly in error.

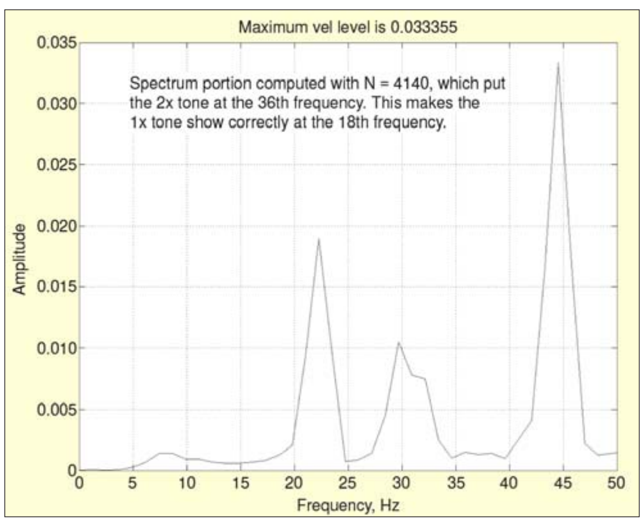

Figure 9. Here the error of Figure 8, has been corrected by changing the FFT length to 4140 which bin centers both the 1× and 2× tones such that the correct values appear in the spectrum.

$$\Delta f = \frac{1}{T} = \frac{1}{N/f_s} = \frac{5120}{4096} = 1.25Hz \qquad (9)$$

Notice that the 2× peak at 45 Hz is kind of lop-sided or cut off at an angle. This indicates that the true spectrum has a peak between 43.75 and 45 Hz and is closer to 45. In a situation like this, if you wish to assume that a cluster of lines in your Hanning windowed spectrum is from a single tone, you can use the Hanning window equations in the Appendix to calculate the peak spectrum value and its frequency. The highest line near 2 times the running speed (due to misalignment) is in bin 37, with a value of 0.03041698192750, at a frequency of 45; the next highest is in bin 36, with a value of 0.02578547949924, at a frequency of 43.75. I used Eqs. A3 and A5 to estimate the true frequency, f_{true}, to be 44.52951399872051 Hz. Eq. A4 was then used to estimate the peak spectrum value, y_{true} to be 0.03335510723959. Bin 37 means the 37th frequency value; the first is zero or DC. So the frequency of the 37th bin is $(37–1)\Delta f$ = 45.

Now let's use the inverse transforming ideas to confirm these values from the equations. I'll describe two ideas, both of which make f_t very close to bin centered. The term bin centered refers to a tone or single frequency component in a segment of the signal we are analyzing. The tone will be bin centered if it has an integer or whole number of periods or cycles in the segment you are FFT'ing. If you think that out, the number of periods of a tone of frequency f, in a signal of length N sampled with a sampling rate of f_s is given by

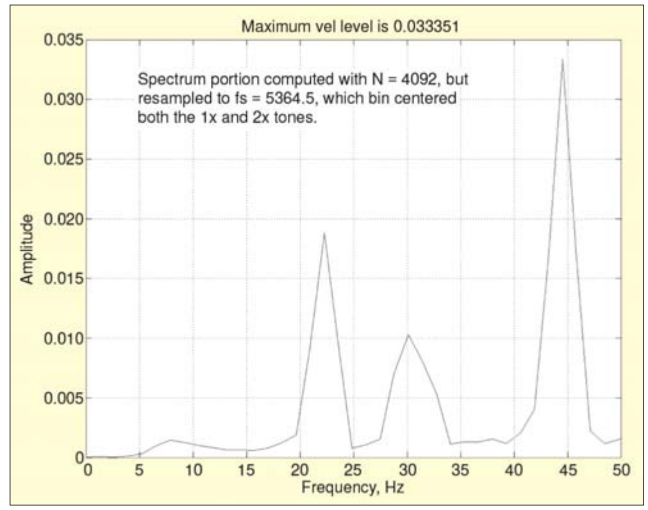

Figure 10. This is a second example of bin centering, this time by resampling. The resampling was accomplished by adding zeros to the transform to increase the sampling rate from 5120/sec to 5364.5/sec, which by Eq (10) bin centers the tone. Notice the symmetry of the spectrum shape around the bin centered peaks at 22.26, and 44.53 Hz.

$$N_p = \frac{Nf}{f_s} \qquad (10)$$

Notice also by using Eq. 9 in Eq. 10

$$N_p = \frac{f}{\Delta f} \qquad (10a)$$

Let's use Eq. 10 to check the number of periods of the tone at f_{true} in our spectrum. Substituting values for f_{true} = 44.52951399872051, N = 4096, and f_s = 5120 into Eq. 10 gives N_p = 35.62361119897641. That's not very close to a whole number. Examining Eq. 10, we have two options to make N_p a whole number: adjust N or adjust f_s. Both work.

First let's adjust N, and this is certainly not resampling. It is taking advantage of MATLAB's ability to calculate a DFT of any length sequence. Solve Eq. 10 with f_{true} = 44.52951399872051, and f_s = 5120, to find N for N_p = 35, we get

N_{35} = 4.024297233631363e+003 ≈ 4024.

Similarly, if we solve to find N for N_p = 36 we find

N_{36} = 4.139277154592259e+003 ≈ 4140.

N = 4024 looks like the best bet, since it's an even number, and only 1/3 of a sample off. If we try that, we get the spectrum portion shown in Figure 8. And this is an instructive mistake. By using the 2× tone to adjust the number of samples, I put the 1x tone right in the middle of the space between the 17th and 18th frequency; if 2× is bin centered, one would certainly want 1× bin centered as well.

To correct the problem and put the 2× tone in the 37th bin or 36th frequency, or close to it, I'll change the FFT length to 4140. This makes the number of periods from Eq. 10 with f_{true} = 44.52951399872051, f_s = 5120, N_{36} = 4140, we get N_p = 36.0063. This is still quite good. (I have experimented a little that even as much as 10% off, you get a pretty good value for the peak amplitude.) Figure 9 gives the resulting spectrum.

And as you can see both 1× and 2× appear as they should. Thus I can say that adjusting the length of the FFT is an excellent procedure for bin centering a tone.

The second procedure to get Eq. 10 to give us a whole number of periods in the segment we are FFT'ing is to modify the sampling rate, f_s. This is resampling. To increase the sample rate, I took the whole string of data, 10,138 values, FFT'd it, added an even number of values to that transform, and inverse transformed it. The new sample rate was to be the new number of values divided by the original duration of the signal. The

FFT of a real signal with an even number of values N will have N values, and the IFFT of a transform with N values also has N values. Adding zeros was explained previously.

Specifically, I wanted 34 periods in Eq. 10 when $f = f_{true}$ = 44.52951399872051, and N = 4096. The new sample rate comes out to be, f_{snew}=4096*f_t/34 = 5364.5. The original signal in AH10C1 contained 10,139 values. I skipped the last value to work with an even number. The duration of this signal is the number of values divided by the sample rate. Since the resampled signal and the original signal must have the same duration, we can say

$$\frac{10138}{5120} = \frac{Nnew}{5364.5}$$

or N_{new} = 1.0622e+004 = 10622. We must add 484 values [(10622 - 10138), 483 zeros and a half Nyquist] to the transform. This is done with the MATLAB script, resamp1.m

```
%resamp1.m is Rresamp for S&V inverse transform article
% Let datalength be the desired data length
datalength=10622;%*****Insert your value
%*****place datashort (data to be resampled) in workspace
   with an even number of values
datashort=datashort(:);
%make a column
DATASHORT=fft(datashort);
N=length(datashort);
nyq=DATASHORT(N/2+1);
%DC value is DATASHORT(1), Nyquist is DATASHORT (N/
   2+1)
%We zeropad DATASHORT to datalength, by replacing Nyquist
   with half Nyquists as last nonzero values and adding
   (datalength–N–1) complex zeros in center
addzeros=zeros(datalength–N–1,1);
czeros=complex(addzeros);
halfnyq=nyq/2;
%Don't need to complex nyq; it is already complex
DATALONG=[DATASHORT(1:N/2); halfnyq; czeros; halfnyq;
   DATASHORT(N/2+2:N)];
datalong=real(ifft(DATALONG));
%must multiply by factor datalength/N
datalong=datalong*datalength/N;
%Now datalong is our resampled data.
```

The computation went fast. When I FFT'd the resulting resampled signal, the tones at 22.26 and 44.53 Hz were bin centered, and that spectrum is shown in Figure 10. The procedure looks good. Thus I can also say that increasing the sampling rate appropriately by adding zeros to the transform is an excellent way of bin centering a tone. I believe this is a good idea and encourage you to try it when you want something resampled.

Conclusions

I have tried to explain the inverse Fourier transform as a Fourier series "kind of add 'em up" and suggested that when thinking about FFT's to think of the N in the position of Eq. 1, but continue to compute with MATLAB and its formulation. I've tried to convince you that a band limited underlying curve exists for any set of sampled data and defined it. I showed that a coarsely sampled curve defined a specific smooth curve going through the values. I presented three different uses for inverse Fourier Transforming vibration signals. Band-wise reconstruction, which includes low pass, high pass, and band pass filtering, was the first. Next I demonstrated reconstruction by eliminating smaller or larger harmonics. Finally, I discussed bin centering and used inverse transforms to show how a tone can be made bin centered in real data. These ideas came from looking at the DFT as in Eq. 1 and Eq. 2 with the $1/N$ in Eq. 1, where it most logically belongs for vibration analysis. The techniques were implemented in MATLAB[3], and the MATLAB m-files that draw these figures are available by contacting the author.

References

1. Newland, D. E., *An Introduction to Random Vibrations, Spectral and Wavelet Analysis*, 3rd ed., Longman Scientific & Technical, Copublished in the United States by John Wiley & Sons, NY 1993; pp 295-370.
2. "C.R.C. Standard Mathematical Tables," 12th ed. 1964; The Chemical Rubber Publishing Company, Cleveland, OH.
3. MATLAB for Windows, Version 5.3, 1999, High Performance Numeric Computation and Visualization Software, The MathWorks, Inc., 24 Prime Park Way, Natick, MA.
4. Annon. "Programs for Digital Signal Processing," IEEE Press, John Wiley & Sons, 1979.
5. Shannon, C. E., "Communication in the Presence of Noise," Institution of Radio Engineers, Vol. 37, 1949; pp 10-21.

Appendix – Frequency/Amplitude Correction Factor

Frarey has worked this out by accurate testing with a Hanning window. He recommends against its use for the box car window, and I agree.[A1] Brüel and Kjær in their Technical Review[A2] present an analytical equation in dB for the corrections without proof. Ming and Kang[A3] give a difficult analysis with which I agree.[A2] I was able to work this out, but the way I explain it takes too many pages. Burgess[A4] theoretically attacks the problem, but his paper is more difficult. However, I will present what I believe to be a correct interpretation of the results from A2 and A3 that is at least something to use. You apply this correction to the two highest lines in a cluster of a few high lines in a spectrum where you believe there is a single tone or harmonic.

Let us define:

T = duration of the signal in seconds.
Δf = line spacing ($1/T$).
Δf_c = frequency offset in Hz from highest line taken in the direction of the next highest line.
y_t = true peak amplitude.
y_h = amplitude of highest line.
y_{nh} = amplitude of the next highest line.

For the rectangular, boxcar, or no window, Frarey recommends not using this[A1] (Eqs. A1 and A2):

$$\frac{\Delta f_c}{\Delta f} = \frac{y_{nh}}{y_{nh} + y_h} \qquad \frac{y_t}{y_h} = \frac{\pi \dfrac{\Delta f_c}{\Delta f}}{\sin\left(\pi \dfrac{\Delta f_c}{\Delta f}\right)} \tag{A1,2}$$

Basically, the Hanning window causes 4 lines to appear in a Hanning windowed DFT of a tone. The zero padded DFT of the Hanning window has a main lobe four Δf's or bins wide. 4 lines always end up in the main lobe and are scaled or have relative heights matching the main lobe shape. The geometry leads to the following equations. For the Hanning window (Eqs. A3 and A4):

$$\frac{\Delta f_c}{\Delta f} = \frac{2y_{nh} - y_h}{y_{nh} + y_h} \qquad \frac{y_t}{y_h} = \left[1 - \left(\frac{\Delta f_c}{\Delta f}\right)^2\right] \frac{\pi \dfrac{\Delta f_c}{\Delta f}}{\sin\left(\pi \dfrac{\Delta f_c}{\Delta f}\right)} \tag{A3,4}$$

Finally, to tidy things up let's add that if:

f_t = true frequency and f_h = frequency of the highest line, then (Eq. A5):

$$f_t = f_h \pm \Delta f_c \qquad \begin{matrix} + \text{ if } f_{nh} > f_h \text{, and} \\ - \text{ if } f_{nh} < f_h \end{matrix} \tag{A5}$$

References for Appendix

A1. Frarey, Jack, "The FFT as a Precise Amplitude and Phase Meter," Chapter 4 of Machinery Vibration Analysis III Course Notes Volume I, Vibration Institute, 1995; Also published and co-authored with Neun, John, as "Phase and Amplitude Errors in Single-Channel FFT Analyzers," *Vibrations*, Vol 4, No. 3, Sept 1988.
A2. "Windows to FFT Analysis (Part II)," Bruel Kjaer Technical Review, n4, 1987, pp 28-31.
A3. Ming, Xie, and Kang, Ding, "Corrections for Frequency, Amplitude, and Phase in a Fast Fourier Transform of a Harmonic Signal," *Mechanical Systems and Signal Processing*, 10(2), pp 211-221, 1996.
A4. Burgess, John C., "On Digital Spectrum Analysis of Periodic Signals," *Jnl. Acoustical Society of America*, v58, n3, Sept 1975, pp 556-567.

The author may be contacted at: hagaberson@att.net.