Adaptive Resampling - Transforming From the Time to the Angle Domain

Jason R. Blough, Ph.D.
Assistant Professor
Mechanical Engineering-Engineering Mechanics Department
Michigan Technological University
1400 Townsend Drive, Houghton, MI 49931
Email: jrblough@mtu.edu

NOMENCLATURE:

a_m is the Fourier coefficient of the cosine term for o_m.
b_m is the Fourier coefficient of the sine term for o_m.
o_m is the order which is being analyzed, $m\Delta o$.
O_{max} is the maximum order that can be analyzed.
$O_{nyquist}$ is the Nyquist order.
O_{sample} is the angular sample rate at which the data is sampled.
N is the total number of sample points over which the transform is performed.
R is the total number of revolutions that are analyzed.
Δo is the order spacing of the resulting order spectrum.
$\Delta\theta$ is the angular spacing of the resampled samples.
Δt is the time spacing between original sampled data points.

ABSTRACT:

This paper details two different methods of Adaptive Resampling as a method to convert Time domain data to the Angle domain. These methods include upsampling/linear interpolation and using an Upsampled Interpolation Filter. Details of each method are presented and include the appropriate choice of algorithm parameters, alias prevention, and computational complexity. This paper begins with a basic review and proceeds to the details necessary to implement the methods in practice.

INTRODUCTION:

Adaptive resampling is a method commonly used to digitally resample data acquired on rotating machinery with a constant time spacing, Δt, to a constant angular spacing, $\Delta\theta$. This process requires a reference to determine the angular spacing times, typically a tachometer signal is used. Adaptive resampling is also used to remove Doppler shifts for instance from vehicle noise pass-by data, in this case the reference signal is an instantaneous vehicle location channel. While these are the two most common applications of adaptive resampling other applications certainly exist.

Adaptive resampling differs from the far more commonly used decimation and interpolation methods of resampling because the locations of the new samples do not have a constant time relationship to the original samples. For example, in the process of interpolating a sampling rate by a factor of 2, each new sample will be half way between each original sample. In adaptive resampling the new locations are determined from the reference signal and can fall anywhere on the original time history's axis. The details of adaptive resampling algorithms which are implemented in the various hardware and software on the market are typically considered proprietary and therefore not commonly published.

The process of adaptive resampling original data in which the frequencies of rotating components or in the case of pass-by data, moving components, have frequencies which vary as a function of a time and are therefore time varying in frequency will be made to appear as though they are time invariant or stationary. Advantages of this process are that time varying frequency components require specialized digital signal processing, DSP, algorithms such as tracking filters for analysis whereas time invariant frequency components do not require any specialized DSP algorithms. This means that the Fourier transform, standard non-tracking digital filters, wavelets, statistics, …etc. can be used with no errors beyond that present in stationary data which has not been adaptively resampled.

The first published material using adaptive resampling for rotating machinery based analysis was Potter, et al from Hewlett Packard [ref. 1-7] in the presentation of a digital order tracking. Hewlett Packard considers the exact implementation of the technique to be proprietary and as such has not published many of the details but did hold a U.S. patent for their implementation. Both the small and large channel count dynamic signal analyzers that HP manufactures have this type of order tracking available either as a standard feature or as an option. Recently many other dynamic signal analyzer manufacturers have begun to offer a type of resampling based order tracking. Again, these manufacturers consider their exact implementation to be proprietary and have not published their methods of implementation.

This paper presents several different adaptive resampling procedures which may be used in anywhere adaptive resampling is required. For simplicity's sake many of the examples are presented from a rotating machinery perspective but it should be realized that simply changing the reference tachometer signal to the reference signal of choice is the only change which must be made to use the procedures elsewhere.

JUSTIFICATION FOR ADAPTIVE RESAMPLING:

A stationary time invariant system is shown in Figure 1, in this case the frequency of the data depicted in the top of this figure does not vary as a function of time. The second plot in this figure represents the spacing relative to time of the samples of the sampled waveform, in this case this spacing is constant and would normally be referred to as Δt. The third plot from the top in this figure shows the spacing of the data samples relative to a shaft angle, each instance of the shaft shows 8 samples/revolution indicated by 'x's, this rate does not change as time increases moving from left to right in this plot. The bottom plot shows the sampled waveform plotted relative to the time domain sampling that it has undergone. In this case, sampling relative to Δt or $\Delta\theta$ results in the same apparent waveform. This is exactly the case for all stationary signals and allows DSP methods which are developed relative to constant Δt's to work well because the kernel of the Fourier transform, or any other transform looks like the signal of interest.

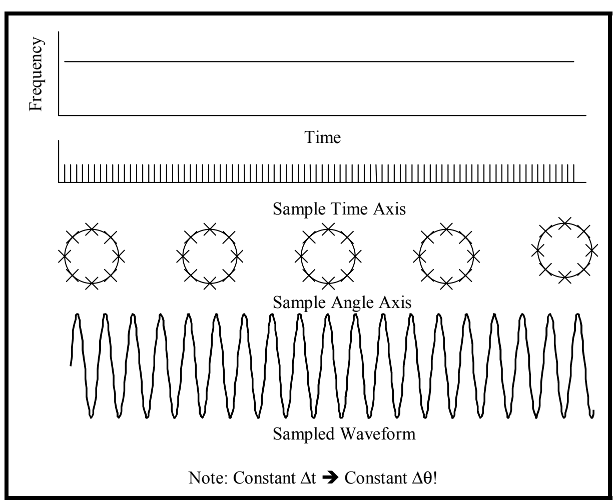

Figure 1: Graphical representation of sampled constant frequency sine wave, indicating constant time and constant angular sampling yield same waveform.

Figure 2 shows the sampling situation when the frequency of the signal of interest can vary as a function of time, in this increasing with time, and a constant Δt is used to acquire the data. In this case it can be seen that the samples relative to angular position vary as a function of time. Obviously, this sampled waveform does not resemble the form of the kernel of the Fourier transform. Frequency components which are varying in this manner will not be stationary on one spectral line of a Fourier transform. This problems manifests itself as leakage an smearing in the frequency domain. For this reason a Hanning window is normally applied to reduce leakage and in turn compromises the frequency resolution of the FFT.

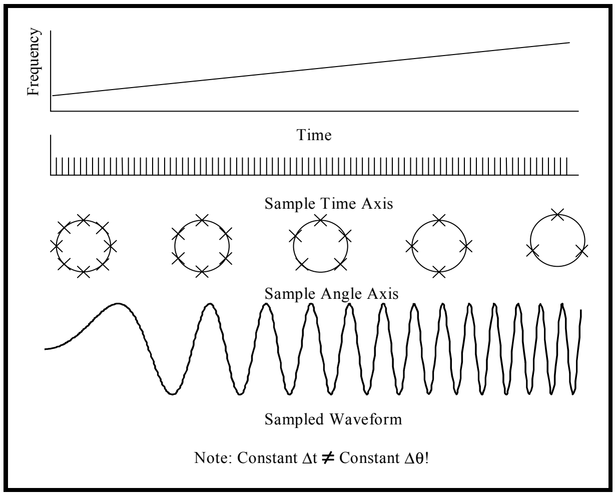

Figure 2: Varying frequency sine wave sampled with a constant Δt.

To overcome the limitations of using standard DSP on frequency varying signals the concept of adaptive resampling is introduced. Figure 3 shows the result of sampling the same varying frequency sine wave of Figure 2 with a constant angular sampling interval, $\Delta\theta$. Notice how the sampled waveform in the bottom portion of Figure 3 once again looks like the sampled waveform in the bottom portion of Figure 1. This waveform, though now sampled relative to angle as opposed to time may now be treated as though it is actually sampled relative to time with the understanding that the results of various DSP algorithms will not be frequency dependent but instead dependent on multiples of the varying frequency component. In rotating machinery applications we define the reference frequency to be 1st order and the multiples to be 2nd, 3rd, or 4th orders if they are 2, 3, or 4 times the reference frequency respectively. If this data were acquired during a pass-by type of event than the source frequencies which did appear as though they were changing, and were due to the Doppler shift, now appear as they would if the pass-by microphone had been mounted on the moving vehicle and there had been no Doppler shift.

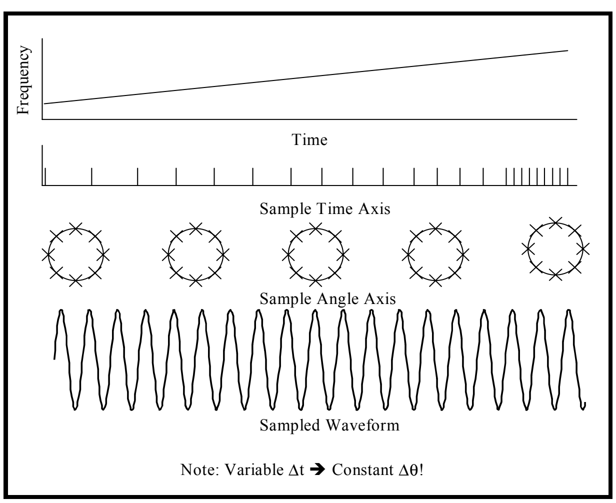

Figure 3: Varying frequency sine wave sampled with a constant angular interval, $\Delta\theta$.

Examples of the sampling parameters for the Fourier transform in terms of angle are given in Equation 1.

$$\Delta o = \frac{1}{R} = \frac{1}{N * \Delta\theta} \qquad\qquad R = N * \Delta\theta$$

$$O_{nyquist} = O_{\max} = \frac{O_{sample}}{2} \qquad\qquad O_{sample} = \frac{1}{\Delta\theta} \tag{1}$$

where: Δo is the order spacing of the resulting order spectrum.
R is the total number of revolutions that are analyzed.
N is the total number of sample points over which the transform is performed.
$\Delta\theta$ is the angular spacing of the resampled samples.
O_{sample} is the angular sample rate at which the data is sampled.
$O_{nyquist}$ is the Nyquist order.
O_{max} is the maximum order that can be analyzed.

As can be seen in Equation 1, there are analogous quantities for each of the time domain sampling parameters. In the angle domain, the order resolution is related to the number of revolutions that the machine turns through over the transform period. It should also be noted that the same sampling criteria applies to avoid aliasing, a minimum of two samples per cycle of the highest order of interest

The kernels of the Fourier transform are reformulated in terms of the uniform angular intervals. These kernels are presented in Equation 2.

$$a_m = \frac{1}{N}\sum_{n=1}^{N} x(n\Delta\theta)\cos(2\pi o_m n\Delta\theta) \qquad\qquad b_m = \frac{1}{N}\sum_{n=1}^{N} x(n\Delta\theta)\sin(2\pi o_m n\Delta\theta) \tag{2}$$

where: o_m is the order which is being analyzed, $m\Delta o$.
a_m is the Fourier coefficient of the cosine term for o_m.
b_m is the Fourier coefficient of the sine term for o_m.

ADAPTIVE RESAMPLING BACKGROUND

Understanding that adaptive resampling can be very beneficial in transforming time varying frequency components into stationary frequency components is a powerful concept and leads to the desire to apply the transforming technology in many instances. Figure 4 shows a skeleton flow chart representing the adaptive resampling process as used for rotating machinery.

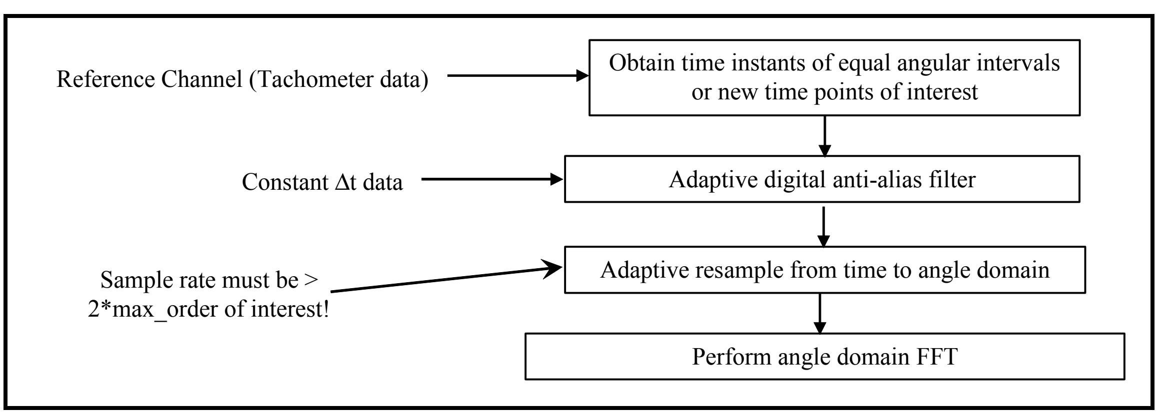

Figure 4: Flow chart of adaptive resampling process.

Adaptive resampling methods process digital time domain data with a uniform Δt after anti-alias filtering with a single frequency analog anti-alias filter. The uniformly sampled time data is then digitally resampled to the angle domain through the use of an adaptive resampling algorithm.

To enable the transformation from the time domain to the angle domain, a reference signal of some type must be measured to allow the determination of the times of the uniform angular intervals. This reference signal is typically a tachometer signal measured on a reference shaft of the operating machine. Through the use of numerical integration and an interpolation algorithm, this tachometer signal can be processed to obtain the time instants of the uniformly spaced angular points. Various tachometer processing algorithms are explained in detail in references [1-9].

The time instants of the angular intervals are then used with a digital interpolation algorithm to obtain the estimates of the response channels at these points in time. Several different interpolation algorithms may be used to estimate these new data samples.

The angle domain data is then analyzed with the use of angle domain Fourier transforms, as shown in Equations 1 and 2 above, to obtain amplitude and phase estimates of the orders.

The result from this angle domain Fourier transform is that the orders fall on spectral lines, regardless of the speed variations over which the transform is applied. Typically the Discrete Fourier Transform, DFT, is applied over the number of points which corresponds to an integer number of revolutions of the machine's rotation and hence is leakage free, as opposed to being restricted to a power of two samples. Since the data is sampled in the angle domain there is a constant number of orders present in the data regardless of the rotational speed of the machine.

ADAPTIVE SAMPLING RATE INTERPOLATION THEORY BASED ON UPSAMPLING AND LINEAR INTERPOLATION.

One adaptive sampling rate interpolation method implemented in multiple commercially available software packages is based on a combination of upsampling the original signal and linearly interpolating between the upsampled data values to obtain the desired data values.

The first step in this adaptive resampling approach, after the new time locations are determined from the reference signal, is to upsample the original data. The amount which the original data is upsampled has a large effect on the signal to noise ratio (SNR) of the final result. The higher the upsampling rate the higher the SNR of the final result. In determining how much to upsample the data, the SNR of the original measurement should be considered. Obviously, it is not necessary to upsample the data by an amount that will result in 80 dB of SNR if the original data only possesses a SNR of 60 dB. A typical upsample factor to obtain a SNR of approximately 60 dB is 16.

Once the data is upsampled, a linear interpolation is performed to obtain a new data value at an arbitrary instant in time as dictated by the reference signal. The upsampled data is evenly spaced in time therefore the linear interpolation is necessary to obtain values which are not evenly spaced in time. The results of the linear interpolation are the adaptively resampled data values.

Finally, the lowest sample rate in the adaptively resampled data must be greater than the original sample rate of the data. If the largest Δt between new data values is greater than the original Δt, the data will contain aliasing.

This adaptive resampling procedure is graphically represented in Figure 5 in the form of a flow chart. In this figure, an example is presented of a procedure that is used to obtain data values that are evenly spaced in the angle domain relative to the rotation of the shaft on which the tachometer signal was measured. In this case, the tachometer signal is processed to generate the reference signal and the new instants in time at which it is desired to obtain samples.

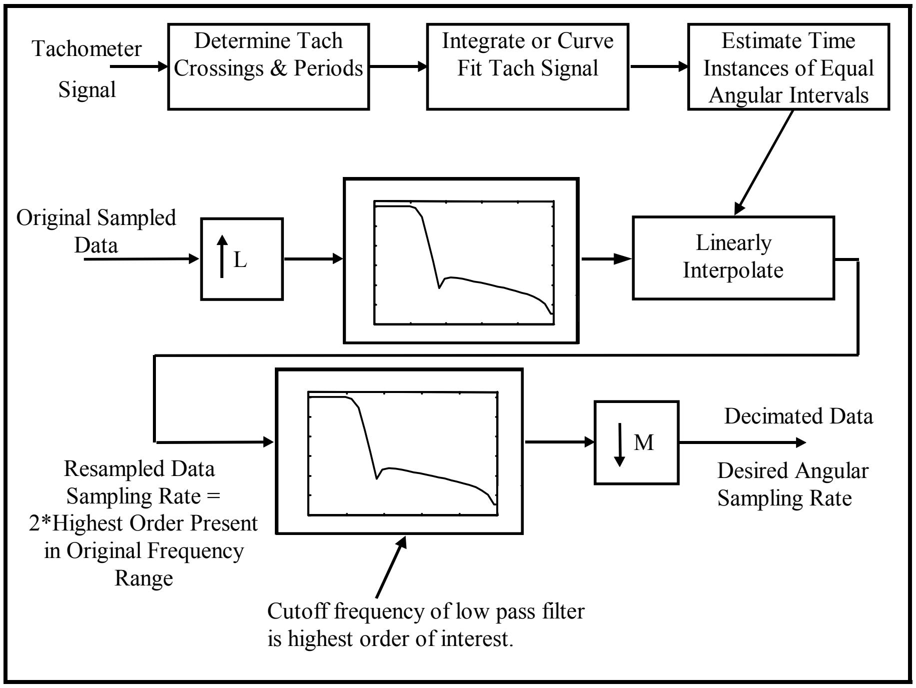

Figure 5: Graphical representation of adaptive sampling rate interpolation, transformation from time domain to angle domain.

The last step presented in this procedure is the final lowpass filtering operation and decimation. This step is often performed on data that is resampled from a rotating machine if the data is acquired during a speed sweep. Acquiring data during a speed sweep implies that if the data is to be evenly spaced in the angle domain, the original sample rate was higher than necessary at the low rpm values based on a maximum order of interest. To prevent aliasing the data must be resampled with a sample rate that prevents aliasing at these low rpm values. This necessary sample rate may result in more samples per revolution then is desired in the end result, hence a decimation operation may be performed to obtain the final desired sample rate. It should be remembered that if a certain angular sample rate is desired which will require decimation as the final step, the data should be resampled at a sample rate which is an integer multiple of this final desired sample rate. This allows a standard decimation procedure to be used on the resampled data.

NUMERICAL EXAMPLE OF ADAPTIVE SAMPLING RATE INTERPOLATION BY UPSAMPLING/LINEAR INTERPOLATION.

This section presents a numerical example of the resampling of a chirp function whose frequency is constantly increasing across the data block. This situation is representative of data acquired on a rotating machine undergoing a speed sweep from a low rotational speed to a higher rotational speed. The data is resampled with differing amounts of upsampling prior to the linear interpolation process to show the effects of upsampling rate.

Figure 6 shows both the time and frequency domain plots of the original signal. Note, how the frequency increases across the time domain trace from left to right and that the frequency domain plot shows no distinct frequency component. The minimum frequency of the chirp is approximately 3 Hz, while the maximum frequency is approximately 12 Hz. A sample rate of 50 Hz was used, a Flattop window is used for all frequency domain spectra since a sine wave type function is being analyzed. The Flattop window also allows the dynamic range of the algorithms to be estimated much better since the height of the window's sidelobes is down ~90 dB. Other windows that do not have sidelobes with such a large amount of rejection will limit the true signal to noise ratio analysis of the frequency domain signal. The sidelobe rejection of these other windows will dictate the amount of signal to noise rejection that can be observed in the frequency domain. Essentially, the signal to noise rejection of the window may be less than that of the algorithm under scrutiny.

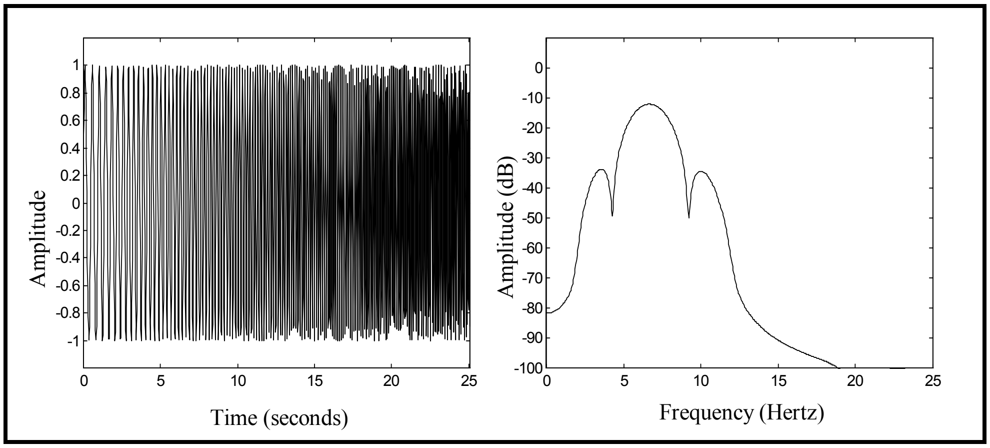

Figure 6: Original chirp signal, time and frequency domain representations.

Figure 7 shows the time domain representations of both the original and resampled signals. The signal was adaptively resampled with a sample rate of 5 samples/cycle of the chirp signal. The left hand plot shows the time signals at the beginning of the time block, the original signal marked with the symbol "*o*" and the resampled signal marked with the symbol "+". Note that the original signal contains many more samples per cycle at the beginning of the time block than the resampled signal and that the original samples are not evenly spaced across a cycle of the waveform. Later in the time block, shown in the right hand plot, the resampled signal contains nearly the same number of samples per cycle as the original signal. The sample rate of the resampled signal is constant across the entire time block with respect to samples/cycle while the sample rate of the original signal is constant with respect to samples/second. The consistent samples/cycle sample rate is also apparent from the fact that in all cycles of the resampled waveform, the samples fall at the same location of each cycle. This property can be verified by comparing the minimum and maximum values of the resampled waveform at both the beginning of the time block and further into the time block. These values are the same in all cycles.

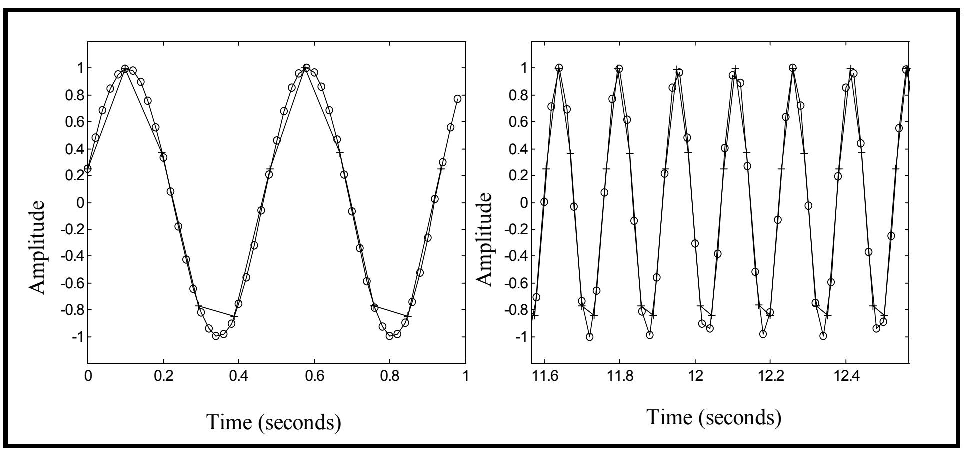

Figure 7: Original and resampled time histories, beginning and middle of time block.

The frequency domain representation of the adaptively resampled signal is shown in Figure 8. The left hand figure was adaptively resampled after upsampling the original signal by a factor of 2 prior to the linear interpolation. The right hand plot was adaptively resampled after upsampling the original signal by a factor of 16. Since the original chirp did not have frequency content greater than approximately ½ of the Nyquist frequency, this upsampling factor gives approximately 64 samples/cycle of the highest frequency of the chirp. Observe that the additional upsampling improves the SNR of the final result by approximately 10 dB. This improvement in SNR comes at the expense of considerable increased computational load.

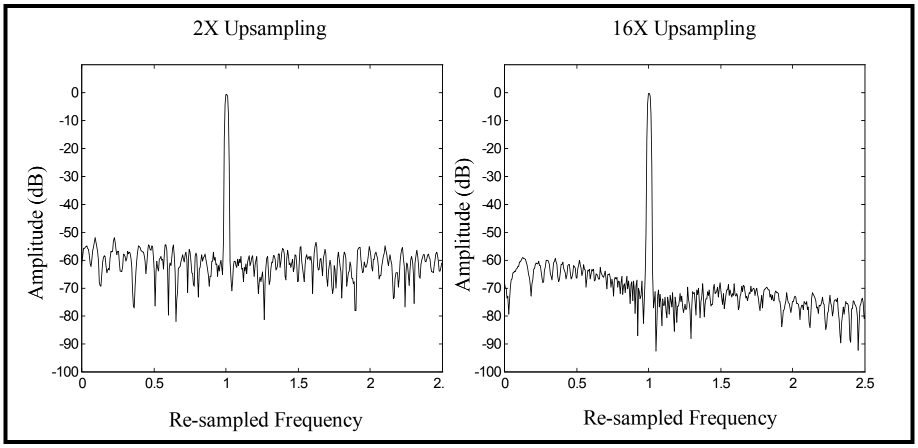

Figure 8: Adaptively resampled signals showing effect of amount of upsampling prior to linear interpolation.

This adaptive resampling method works well given a powerful computer to perform the operations and a maximum desired SNR of approximately 60 dB. A greater dynamic range is possible, however, the computational expense to improve this result increases in a non-linear fashion.

To reduce the computational load and the required memory in the implementation of this method, only the upsampled data values that are to be used in the linear interpolation process are calculated. This realization together with the use of a polyphase interpolation filter can greatly reduce the number of calculations required to implement this algorithm. However, the SNR of this method is still somewhat restricted to approximately 60 dB. The method also requires that a minimum of two upsampled points be calculated for each new data value. The linear interpolation requires one data point before and after the desired sample instant.

Further gains in the signal to noise ratio can be gained through the use of a higher order interpolation technique such as a spline or Lagrangian interpolation algorithm. However, all higher order interpolation algorithms require that more upsampled data points be calculated and require more calculations than the simple linear interpolation, again leading to a very computationally demanding implementation.

ADAPTIVE RESAMPLING BASED ON AN UPSAMPLED INTERPOLATION FILTER.

Non-adaptive decimation and interpolation methods are based simply on lowpass digital filters that must possess a specified cutoff frequency. The adaptive sampling rate interpolation based on an upsampled filter is based on a specific filter that must possess very distinct properties. This section will explain the requirements of this filter as well as its implementation. The section will conclude with an analysis of the performance of this method.

The analog sampling process can be approximated digitally through the use of the sinc (sin x/x) function [ref. 11]. The digital resampling process is accomplished by positioning the center of the sinc function at each point in time a new data sample is desired. The analytical sinc function exists for all time and acts as an allpass digital filter. Since the digitally sampled data does not exist for all time, the sinc function must be truncated in time. The frequency bandwidth of this truncated function must also be limited to eliminate aliasing errors.

One method of producing an FIR filter that approximates the ideal sinc function is through the generation of a sinc function in the time domain which is truncated in time by the application of a window function. The window that is applied to truncate the function has an effect on the frequency domain characteristics of the filter. The transition bandwidth and stopband rejection, as well as the passband ripple, are affected.

Time and frequency domain examples of a truncated sinc function are shown in Figure 9. This sinc function was generated directly in the time domain and truncated with a Hanning window. The zero crossings of the sinc function are spaced exactly a Δt apart. The zero crossing spacing is what determines the cutoff frequency of an FIR filter based on the sinc function. This property along with the unity amplitude at the center of the function allows any data value which falls at an original Δt to pass the through the filter unaltered.

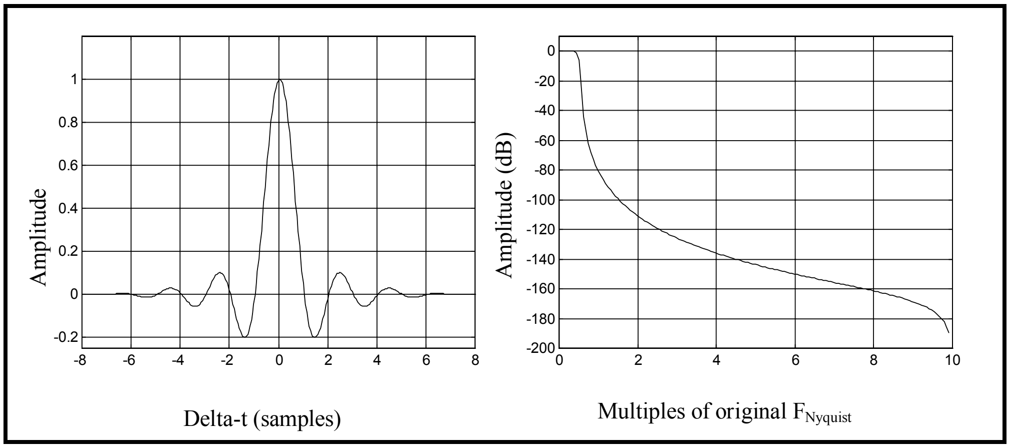

Figure 9: Time and frequency domain plots of truncated sinc function (interpolation filter).

The cutoff frequency of this truncated sinc function is the original Nyquist frequency. The filter uses 8 points prior and after the point in time at which the new data point is calculated. The filter possesses 20 values between each zero crossing and therefore could be used to upsample a signal by a factor of 20, giving approximately 80 dB of dynamic range.

To implement this filter in a computationally efficient manner only 16 (2*8) multiplications/adds per new data point are required. The first step in the application of the filter is to center the filter at the point in time that a new data value is to be estimated. Only the filter coefficients that are aligned in time with original sampled data values are used to calculate the new data sample. All other filter coefficients are multiplied by zero. This implies that only every 20th filter point is needed to calculate the new interpolated data value. This computationally efficient implementation is shown in Figure 10.

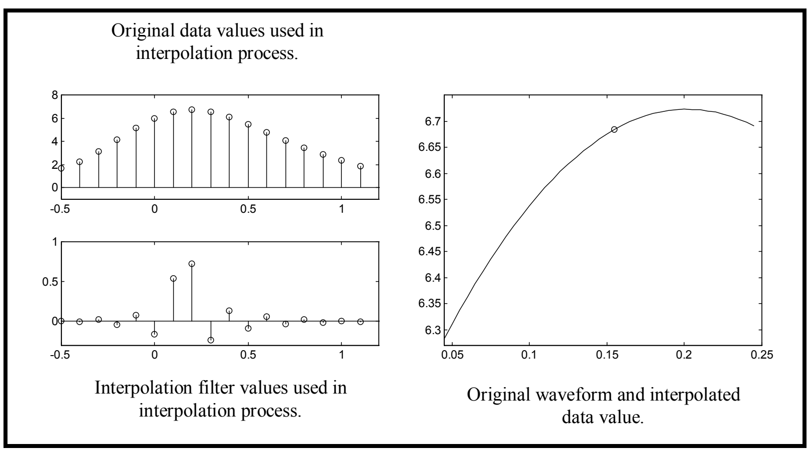

Figure 10: Example of efficient implementation and result of digital interpolation process.

As can be seen in Figure 10, this interpolation method provides very accurate results. To implement this method for adaptive resampling requires considerably more overhead as described below. Even with the necessary overhead this method is a very efficient method for calculating a new data sample at an arbitrary time.

The adaptive implementation of this interpolation method requires that a filter be pre-calculated with a very large number of filter coefficients. The procedure is based on approximating the analog sampling process, in that a new data value can be calculated at any point in time. This requires that the filter be created with, for example, 8192 values between each original Δt. This upsampled filter can then be applied in the same manner as the filter described above. Note there are still only 16 (8*2) multiply/adds per new interpolated data point. By upsampling the filter by such a large amount, an approximation of the analog sampling process is obtained by using the set of filter coefficients which are closest to the time at which it is desired to have a new data value. This is in effect applying a digital sample and hold circuit, where the original signal is upsampled by a very large amount and the new data values extracted from the upsampled data by assuming the closest point is the correct point.

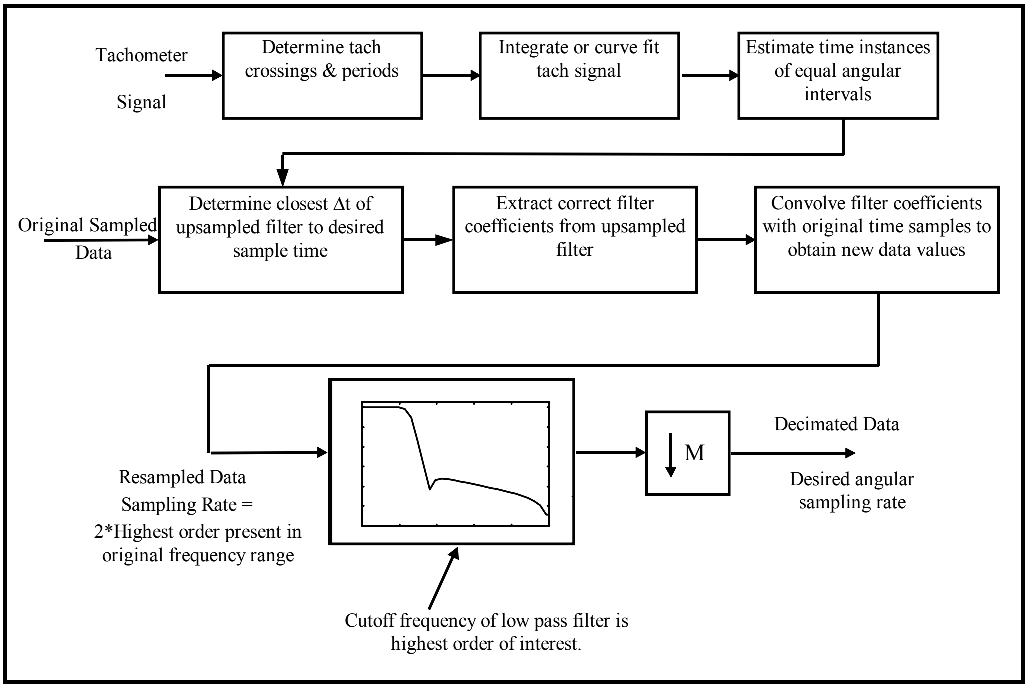

Figure 11: Graphical representation of adaptive resampling with an upsampled interpolation filter.

Figure 11 shows a graphical representation of an adaptive implementation of the upsampled interpolation filter resampling process for synchronous sampling with respect to a reference tachometer signal. This procedure is equivalent to the procedure presented in Figure 6 for the upsampling/linear interpolation adaptive resampling process.

NUMERICAL EXAMPLE OF UPSAMPLED INTERPOLATION FILTER ADAPTIVE RESAMPLING.

The same test case used to exhibit the performance of the adaptive resampling by upsampling and linear interpolation is again used to present the performance of the adaptive resampling based on an upsampled interpolation filter. Using the same example for both algorithms allows the broadband noise levels to be directly compared. The example is a chirp function of a linear sweep. The chirp is resampled such that it should appear as a sine wave with a resampled frequency of 1.

Figure 12 shows the frequency domain representation of both the original signal and the adaptively resampled result.

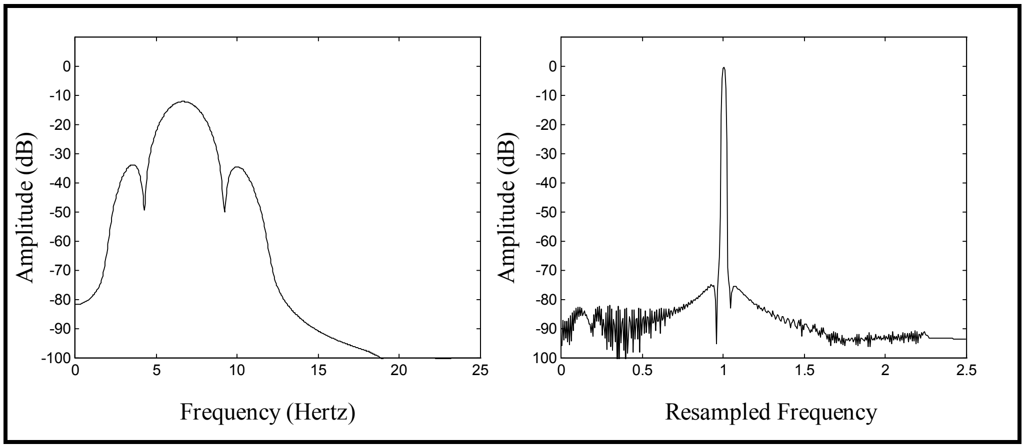

Figure 12: Frequency domain representations of original and resampled chirp function.

The upsampled interpolation filter used in this example was generated by truncating an analytical sinc function with a Hanning window. The filter used 9 points before and after each resampled data sample and contained 8192 steps between each original Δt. Note that while the adaptively resampled results from using the upsampling/linear interpolation approach had approximately 60 dB of dynamic range, the result from using the upsampled linear interpolation filter has a dynamic range of approximately 75-80 dB.

The effect of using fewer steps between each original Δt is shown in Figure 13. The left hand plot shows the result using 4096 steps with 9 original data values prior to and after each new data sample used in the interpolation process. The right hand plot shows the result using 4096 steps with 6 original data values prior to and after each new data sample used in the interpolation process. Note that the effect of using 18 data samples in the interpolation process compared with using 12 data samples results in a greater signal to noise ratio in the final result. Both plots show approximately the same SNR near the frequency of the signal, however, the plot generated by using 18 data samples has a higher SNR away from this frequency. Using either number of data samples results in a SNR of at least approximately 75 dB everywhere, surpassing that of the previously presented upsampling/linear interpolation based adaptive resampling method.

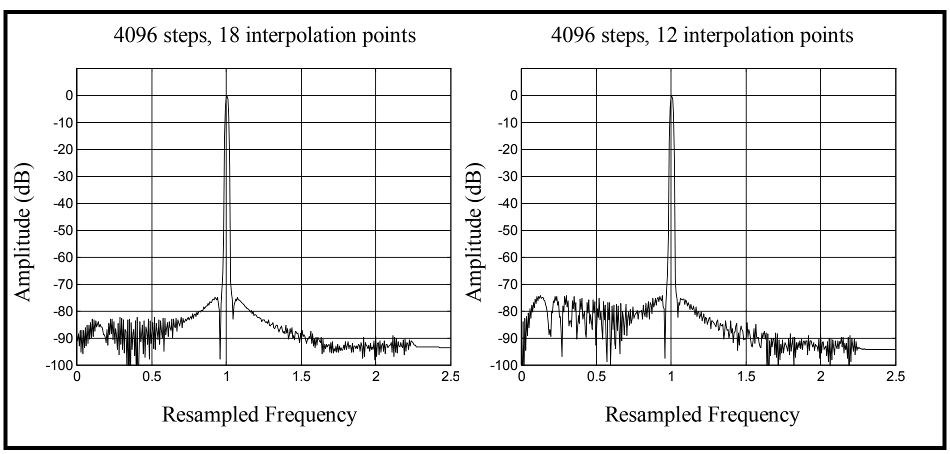

Figure 13: Frequency domain representations of resampled chirp function showing the effect of the number of interpolation points.

Figure 14 further shows the effect of the number of steps that the original Δt is subdivided into. The left hand plot uses an upsampled interpolation filter with 512 steps. The right hand plot uses an upsampled interpolation filter with 1024 steps. Both of these filters use 9 original data samples prior to and after the new data sample. Again, note that both of these filters give approximately 75 dB of signal to noise ratio. It can clearly be seen that as the number of steps in the filter is decreased the broad band noise becomes larger. The SNR is still dominated by the behavior of the filter near the large frequency component, this frequency range is controlled by the window used in generating the filter.

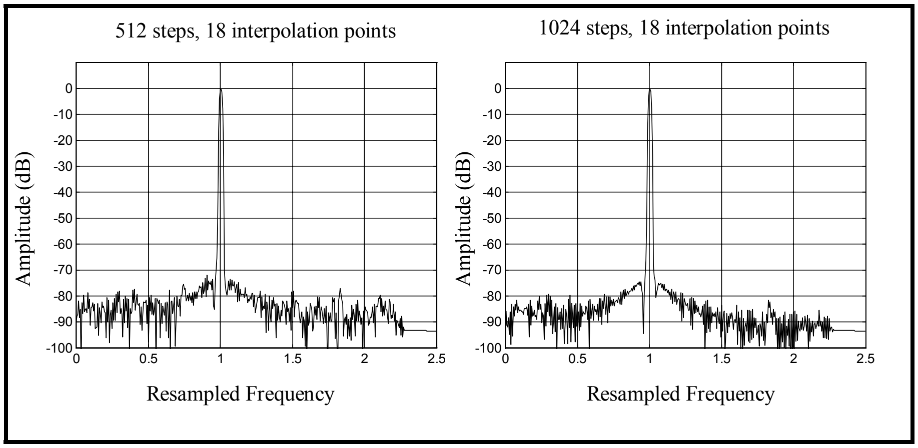

Figure 24: Frequency domain representation of resampled chirp functions showing the effect of the number of steps.

NUMERICAL EXAMPLE TO DETERMINE THE EFFECTS OF EFFECTIVE CLOCK JITTER IN THE UPSAMPLED INTERPOLATION FILTER ADAPTIVE RESAMPLING PROCEDURE.

To further analyze the effect of the number of steps that the original Δt is subdivided into, a second example was formulated. This example again uses a chirp function, however the maximum frequency of this chirp is much closer to the Nyquist frequency. This situation allows a better evaluation of the effect of the effective clock jitter that is introduced by the adaptive resampling filter. Clock jitter is introduced if the timing of the sampling intervals is not accurate and the data is sampled at non-uniform intervals. All of the filters used in this example were generated by truncating an analytical sinc function with a Hanning window. This is essentially the same filter that was used in the previous example. These filters also use an identical 9 points prior to and after each resampled data sample in the interpolation process, regardless of the number of Δt subdivisions.

Figure 15 shows the frequency domain representations of both the original chirp and the adaptively resampled chirp calculated using 8192 steps.

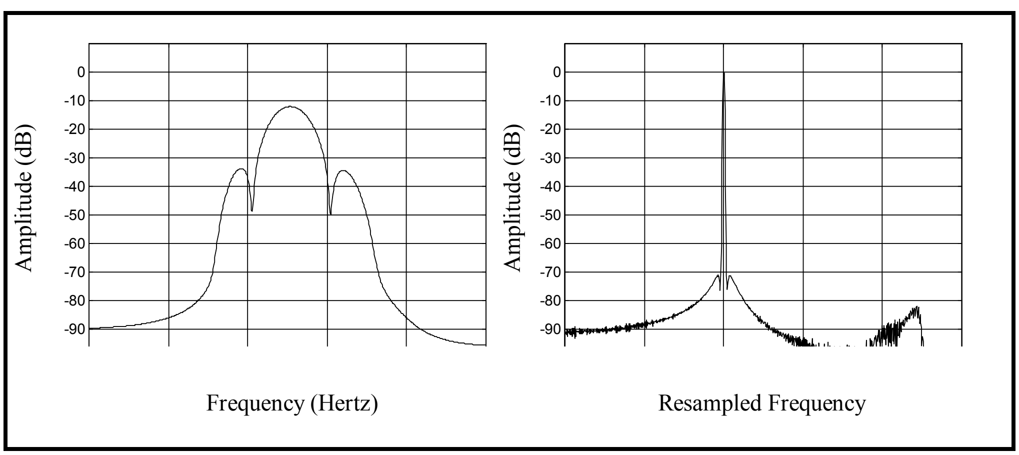

Figure 15: Frequency domain representations of original and resampled chirp.

The SNR of this resampled chirp is greater than 70 dB across the entire spectrum showing that the performance of the algorithm provides an acceptable SNR near the original Nyquist frequency.

Figure 16 shows the same chirp function resampled with interpolation filters using 512 and 1024 steps respectively. Clearly, it can be observed in this figure that neither of these results has an acceptable SNR. An acceptable SNR is determined by the measurement system and process. Typically experimental measurements have a SNR of approximately 65 dB. The amount of jitter introduced by using too few steps results in a high amount of broadband noise regardless of the number of interpolation points employed.

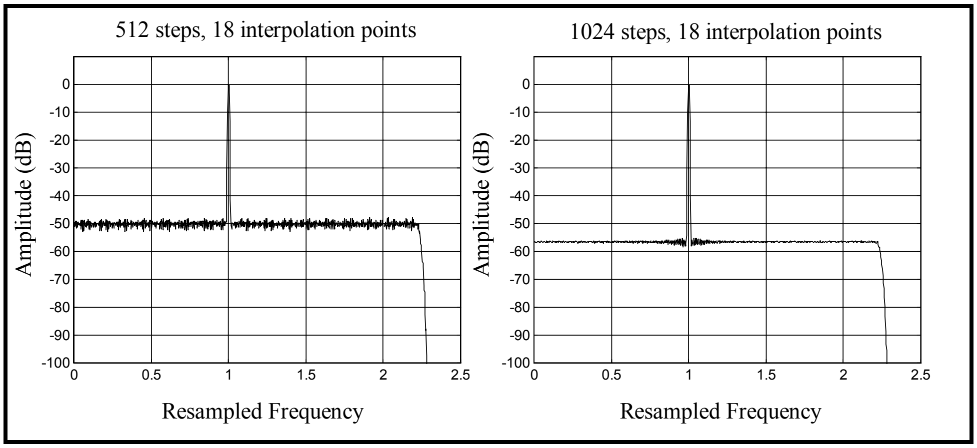

Figure 16: Frequency domain representation of adaptively resampled chirps showing the effect of the number of steps.

Figure 17 shows the same adaptively resampled chirp function. To generate this figure, 2048 and 4096 steps were used in the upsampled interpolation filter. Both of these plots show that the SNR has improved considerably over the previous two filters. Each of these results is acceptable and has over 70 dB of dynamic range. The difference between the two results is exhibited in the amount of noise away from the large frequency component at resampled frequency 1.

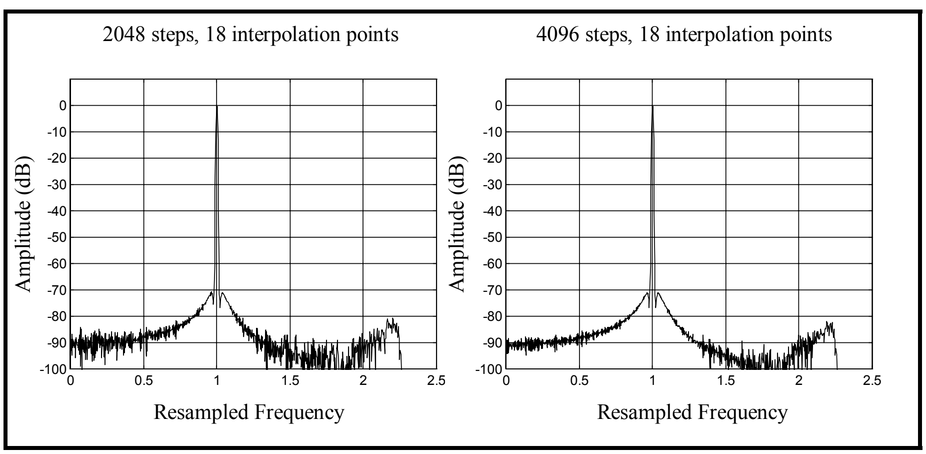

Figure 17: Frequency domain representations of adaptively resampled chirps showing the effect of the number of steps.

Finally, Figure 18 shows the same chirp function adaptively resampled with the upsampling/linear interpolation approach. These plots are included for comparison and completeness to show that this method also functions well for adaptively resampling components near the Nyquist frequency. The left hand plot was upsampled by 16 prior to the linear interpolation. The SNR of this result is approximately 60 dB. While 60 dB is acceptable in many instances with experimental data, it is not as good as the 70+ dB exhibited in the upsampled interpolation filter results. To obtain a higher SNR the same chirp was upsampled by a factor of 32 prior to linear interpolation. This result is shown in the right hand plot. Note that the SNR has improved in this case to the 70 dB exhibited by the upsampled interpolation filter results. This amount of upsampling prior to the linear interpolation process requires many more calculations than the upsampled interpolation filter.

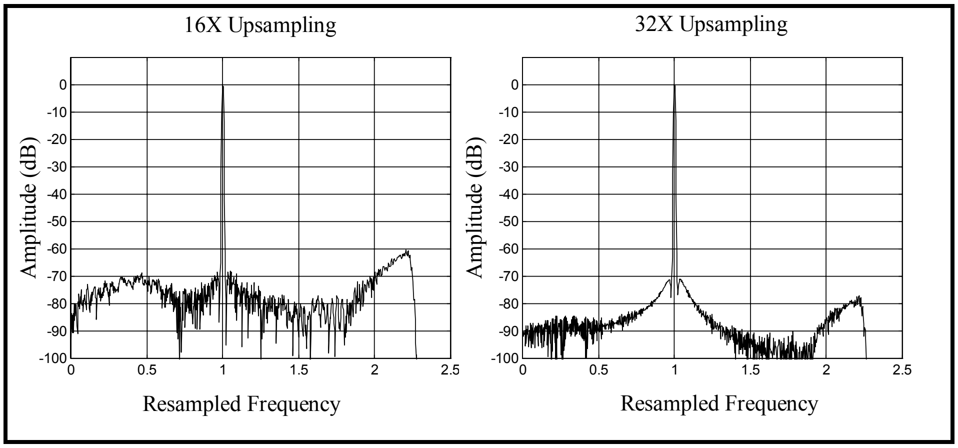

Figure 18: Frequency domain representation of adaptively resampled chirps showing the effect of the amount of upsampling prior to linear interpolation.

ADDITIONAL ADAPTIVE RESAMPLING ERRORS.

Both of the adaptive resampling procedures have errors inherent in them. Both methods presented have some amount of approximation. To implement a truly perfect adaptive resampling algorithm would require a separate digital filter to be designed for each new data sample. This filter would possess the exact amount of phase delay between the original data samples and the new data sample. This type of procedure would be very computationally demanding and is not feasible even with today's powerful computers.

The upsampling/linear interpolation method has an error that is associated with the linear interpolation process. When linearly interpolating a sinusoidal function, an error that is similar to a slight phase delay at each sample is introduced into the resampling process. The magnitude of this error is not predictable and varies as a function of the distance from the closest original time sample to the newly interpolated sample. This error is reduced through a higher upsampling factor prior to the linear interpolation as shown in the previous examples.

The upsampled interpolation filter adaptive resampling procedure introduces an error that can be described exactly by sample clock jitter. The error that sample clock jitter introduces into a signal is a broadband type of noise. This error is also not predictable in the general case and is minimized through the use of a filter that is created with a larger upsampling factor. The magnitude of the possible jitter error can be analyzed if the characteristics of the original data acquisition system that was used to acquire the data are understood. The magnitude of the error that is introduced by sample clock jitter is a function of both the sample clock instability and the number of bits which the ADC uses. An analysis of this error is presented by Watkinson in reference [10].

Both adaptive resampling procedures require that the tachometer or reference signal be very accurate. If, for example, the tachometer signal that is measured has substantial inaccuracies present, or is processed in such a manner as to have errors, the new values of time at which data values are to be estimated will be in error. This error can manifest itself as either a bias error or as a random error.

Bias errors are introduced if the sweep rate of a test sweep is estimated incorrectly. For example, a bias error will occur if the sweep rate is estimated to be too slow. The new data values will be estimated to be further apart than they actually are, resulting in a non-integer number of samples/revolution and/or a non-uniform angular spacing. This invalidates the reason for performing the resampling process. If a Fourier transform were used to analyze resampled data with this error present, leakage would be present in the results which otherwise should have been leakage free!

A random error could occur if the machine's rpm were estimated to be slowly varying when in actuality it was constant. This error in rpm would introduce a Δt spacing error that is constantly varying from too small a Δt to too large a Δt. This type of error is similar to sample clock jitter and would therefore introduce broadband random noise into the resulting data. This type of error would also occur if it were desired to analyze the torsional vibration of a system and the number of tachometer pulses/revolution was less than the highest order that was present in the data.

CONCLUSIONS

This paper has presented first the justification and background necessary to understand why adaptive resampling is performed in the analysis of rotating machinery data. Next, the sampling parameters and Fourier transform are shown in their re-formulated form for use on data sampled with a constant angular interval verses a constant time interval.

The adaptive resampling process itself is then presented using two different methods of adaptively resampling, the upsampling/linear interpolation procedure and the procedure based on the use of an upsampled interpolation filter. Examples were presented of analytical test cases to show the effective signal to noise ratio of each method based on different processing choices.

Finally, the computational complexity of the two methods is discussed, it becomes clear that there are major computational advantages in the upsampled linear interpolation procedure if implemented in an intelligent fashion.

REFERENCES:

1. Potter, R. and Gribler, M., "Computed Order Tracking Obsoletes Older Methods", Proceedings of the Society of Automotive Engineers Noise and Vibration Conference, Traverse City, MI., 1989, SAE paper no. 891131.
2. Potter, R., "A New Order Tracking Method for Rotating Machinery", Sound and Vibration Magazine, Vol. 24, No. 9., September 1990.
3. Hewlett Packard, "Appendix A – Computed Synchronous Resampling and Order Tracking", Effective Machinery Measurements using Dynamic Signal Analyzers, Application Note 243-1, pp. 64-69.
4. Potter, R.W., "Tracking and Resampling Method and Apparatus for Monitoring the Performance of Rotating Machines", United States Patent Number 4912661, March 27, 1990.
5. McDonald, D. and Gribler, M., "Digital Resampling – A Viable Alternative for Order Domain Measurements of Rotating Machinery", Proceedings of International Modal Analysis Conference 9, Italy, pp.1270-1275, 1991.
6. Kirtley, N., "The Fine Art of Order Tracking", Hewlett Packard Realtime Update, Spring 1994, pp. 1-4.
7. Hewlett Packard, "Realtime Basics: Order Analysis", Hewlett Packard Realtime Update, Fall 1996-Winter 1997, pp. 8-9.
8. Vold, H. and Leuridan, J., "High Resolution Order Tracking at Extreme Slew Rates, Using Kalman Tracking Filters", Proceedings of the SAE Noise and Vibration Conference, Traverse City, MI., 1993, SAE paper no. 931288.
9. Blough, J.R., "Improving the Analysis of Operating Data on Rotating Automotive Components", Ph.D. Dissertation, University of Cincinnati, 1998.
10. Watkinson, J., *"The Art of Digital Audio"*, Focal Press, Oxford, England, 1991.
11. Crochiere, R.E. and Rabiner, L.R., *"Multirate Digital Signal Processing"*, Prentice Hall, Englewood Cliffs, New Jersey, 1983.