43rd Turbomachinery & 30th Pump Users Symposia (Pump & Turbo 2014)
September 23-25, 2014 | Houston, TX | pumpturbo.tamu.edu

FUNDAMENTALS OF SIGNAL PROCESSING APPLIED TO ROTATING MACHINERY DIAGNOSTICS

Gaston H. Desimone
Latin America Technical Leader
Machinery Diagnostic Services, GE Measurement & Control
Buenos Aires, Argentina

Mr. Gaston Desimone is the Latin America Technical Leader for Machinery Diagnostic Services at General Electric Measurement and Control, in Buenos Aires, Argentina. His organization is responsible for solving machinery vibration problems mainly in the Oil and Gas and Power Generation Industry, as well as any other industry involving critical rotating machinery. He joined Bently Nevada Corporation in 2000, continuing his career at GE Optimization and control. Since 2000, he has gained experience as Systems and Instrumentation field Engineer, Machinery Diagnostic field Engineer and both internal and external instructor in the Machinery Diagnostic area. Mr. Desimone obtained his Bachelor´s degree in Mechanical Engineering from Buenos Aires University.

ABSTRACT

Rotating machinery diagnostics have been, and will be, a critical area in the Oil and Gas Industry. Though in the early days oscilloscopes and spectrum analyzers were typically used to analyze vibration data, in recent years, technological advances in computers have made it possible to make use of them in solving machinery problems. However, it becomes critical to accept the fact that in order to manipulate vibration signals in a software environment, they have to be previously transformed into digital format. Even though information on the various signal processing techniques are readily available, most of the time, the manner in which this topic is discussed tends to be somehow theoretical. This tutorial intends to provide a summary of these techniques, but most importantly, the link between the basic theory and real applications. It also emphasizes how choosing the wrong parameters in the data acquisition device will definitely affect the quality of vibration data, critical to solving machinery problems.

INTRODUCTION

The use of analog oscilloscopes and spectrum analyzers to diagnose machinery problems is now a thing of the past. As computers got more powerful, they started being used more frequently as diagnostic tools. However, when using computers, the Machinery Diagnostic Engineer has to consider an important fact: they work in a digital world. For this reason, the various vibration signals coming from the rotating equipment, which are “analog” in nature, need to be transformed into the digital domain. The term “analog” refers to the continuous nature of the signal; it contains an infinite number of amplitude levels, separated by infinitesimal time intervals. A digital signal, on the other hand, is not continuous, as it is formed by a finite number of amplitude levels, separated by finite time intervals (see figure 1). Even though a great amount of published information can be found on this subject, it tends to be too theoretical for the machinery diagnostic Engineer, making it difficult to evaluate the consequence of choosing the wrong parameters in the quality of the data, which in turn drives the effectiveness of the diagnosis.

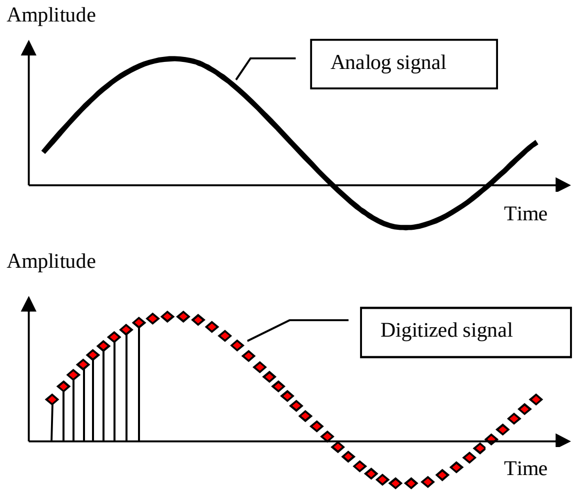


Figure 1: Comparison between an analog (top) and a digitally sampled (bottom) signal.

The process through which an analog signal is transformed into digital is usually referred to as “sampling”. Before getting into further detail, let´s take a look at the whole sampling process, which starts with an analog signal, and ends up as a digital signal, along with one of the most common operations on the digital signal, the Fast Fourier Transform (FFT). In Figure 2, the analog signal goes through a low pass filter, known as “Antialiasing filter” (AAF), which attenuates frequency components above the specified frequency. Then the voltage level of the signal is “quantized” through an Analog – to – Digital converter (ADC), wherein the voltage is divided into discrete levels. After having “measured” the signal voltage, it is stored in some type of memory, at a certain rate specified in samples per second. Once the signal has been sampled, it becomes available for either direct representation or further processing.

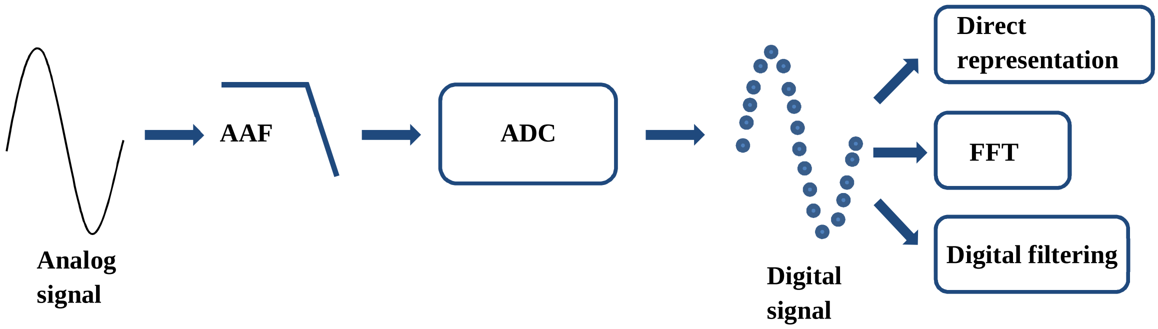


Figure 2: Flow diagram for converting an analog signal into digital.

Let´s take a look at each part of the process with a little more detail. We start with the ADC and shall cover the antialiasing filters later.

ANALOG TO DIGITAL CONVERTER

An “Analog-to-digital” converter or ADC is a device incorporated in all modern data acquisition instruments, and is used to convert the input analog signal into a digital or discrete signal, so that it can be further processed. For the sake of keeping this paper limited to the practical world, we will only focus on the most important parameter for ADCs, the bit number. This number directly affects the amplitude resolution of the ADC, i.e. how finely it slices the configured full-scale measurement range. Roughly, the number of digital levels in which the full scale is divided is 2^n, with n being the number of bits. The higher the bit number, the higher the resolution. For example, an 8-bit ADC divides the vertical full scale into 256 levels, while a 16-bit ADC achieves 65536 levels. It is also important to have in mind that this parameter is fixed, once the data acquisition instrument has been selected. Figure number 3 shows four examples with different amplitude resolutions, for a 20 mils pp full scale range.

For practical applications involving vibration amplitude measurements, anything at or above 12 bits will certainly do the job. Even though bit numbers as high as 24 or even 32 are available today, it has to be considered that high numbers of bits do have a price: memory space, in addition to being overkill. Let´s consider the 12-bit case in Figure 3. When trying to record a vibration signal for a typical machine, 20 mils of full scale is usually enough. Then, using 12 bits will yield a resolution as high as 0,0049 mils. At this point, we must ask ourselves; do we really need more resolution?.

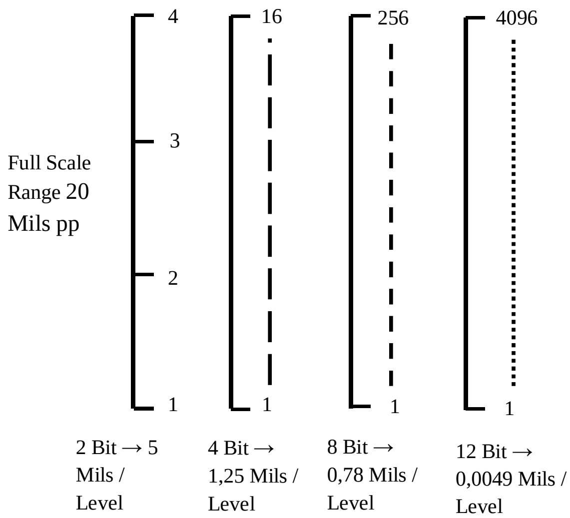


Figure 3: Comparison between numbers of bits for the ADC (resolution) for a full scale range of 20 mils pp.

TIME RESOLUTION – SAMPLING FREQUENCY

Time resolution is basically controlled by how fast the instrument collects the data (sampling frequency). In contrast with the amplitude resolution, time resolution can be modified when configuring the data acquisition device up to some maximum limit. Because of this, the Machinery Diagnostic Engineer is required to have some knowledge on its effect on data quality. Let´s examine two basic examples shown in Figure 4, one with low time resolution and the other with a high time resolution. In this example, the black line corresponds to the analog signal being sampled. For case a), by using a good sampling frequency, the reconstructed digital waveform (connecting the red dots) results in a closer representation of the original signal. However, when we look at case b), in which the sampling frequency is three times slower, the reconstructed signal becomes a poor representation of the analog one.

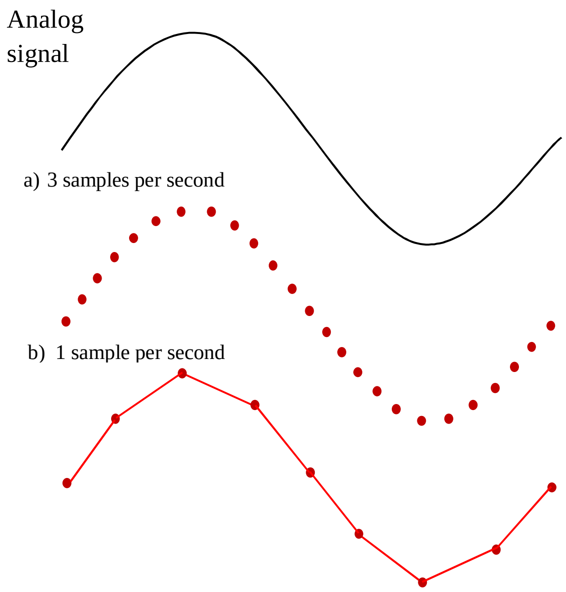


Figure 4: Comparison between good time resolution (case a) and poor time resolution (case b).

Now let´s take a look at more realistic examples. Let´s consider a typical Time base plot, which is a direct representation of the digitized signal. In this particular example, the machine is running at 1700 rpm, and the dominant frequency is synchronous or 1X (the main frequency equals running frequency). The same analog signal coming from the vibration transducer has been sampled in two different ways: 640 Hz or samples per second (Figure 5), and 25600 samples per second (see Figure 6). For comparison purposes, both plots use the same vertical and horizontal scales. The signal that has been digitized at 640 Hz appears to be quite smooth, exhibiting overall amplitude of 6 mils pp. However, when we look at the other signal, it turns out that the waveform is not as uniform as the previous case, showing some spikes that increase the overall amplitude from 6,6 to 8,5 mils pp. Due to the low sampling rate in figure 5, those spikes go undetected. In other words, using a low sampling rate will prevent us from getting all the information from the original signal.

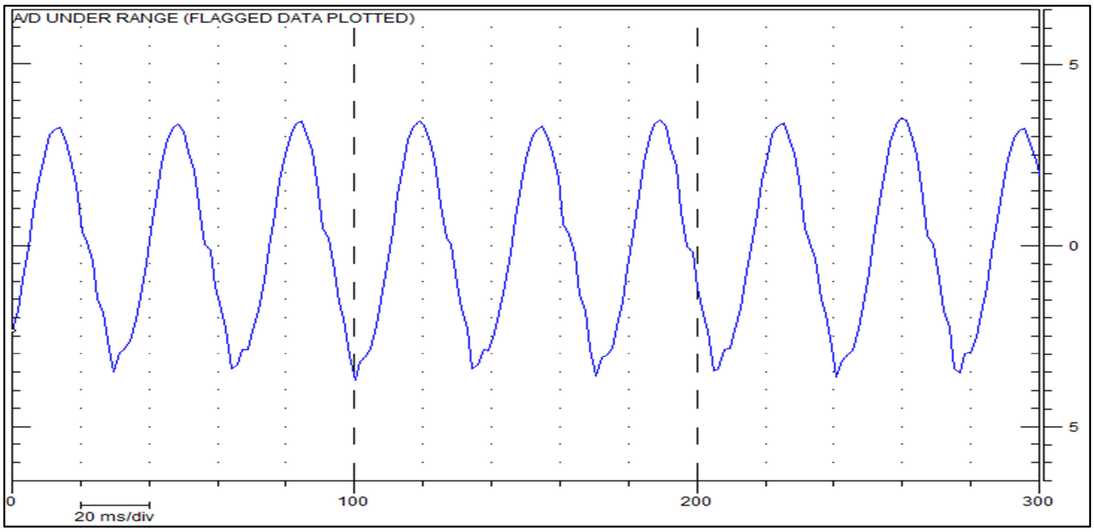


Figure 5: Digital signal sampled at 640 samples per second or 640 Hz.

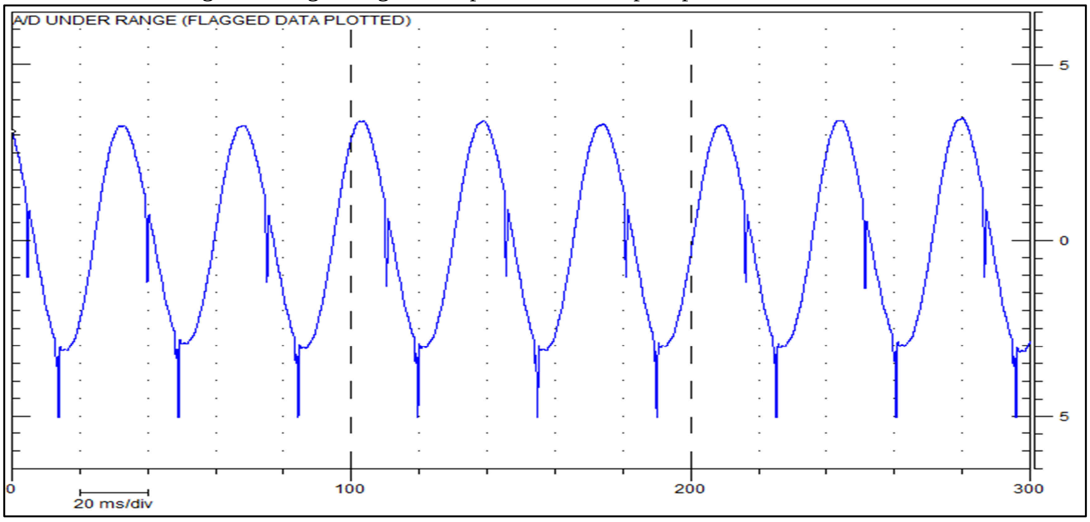


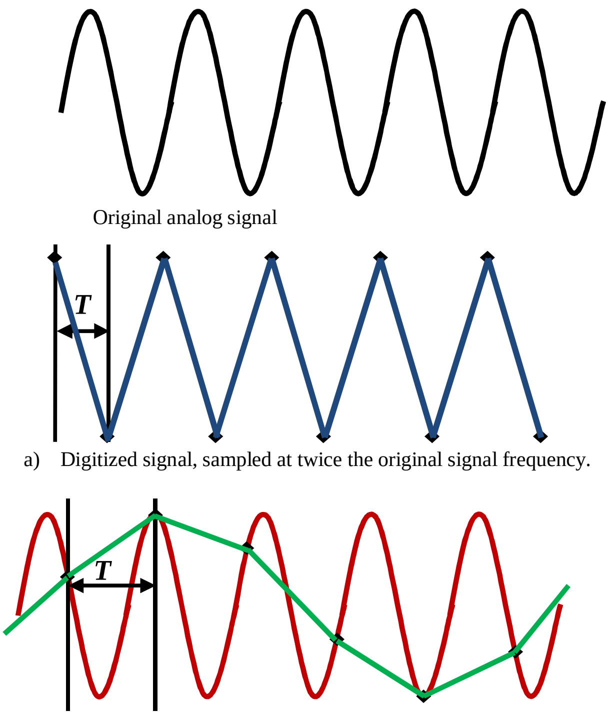
Figure 6: Digital signal sampled at 25600 samples per second or 25600 Hz.

Now that we have already seen some real examples, it is important to define another term associated with the subject of this paper. In both previous examples, we can see that only 300 milliseconds worth of signal are being displayed. Generally speaking, when a signal is being sampled, we are dealing not only with samples being collected at discrete time intervals, but also with a finite observation time, or a finite number of collected samples. Each group of samples being actually stored in the acquisition device is called "time record", and the amount of samples included in it is referred to as "time record size N". We will come back to this concept later on this paper, when we discuss the FFT.

ALIASING – NYQUIST THEOREM

When we talked about time resolution, it was made clear that the faster the samples are collected, the better quality we get for the reconstructed signal. Going the other way, the slower the samples are collected the lower quality we achieve for the digital signal. If we continue to decrease the sampling frequency, we get to the point in which we are not able to reconstruct the original signal, but an "alias" of that signal. This phenomenon is known indeed as "Aliasing". In order to better understand this concept, we will consider the following scenario. Let´s suppose we need to sample or digitize the analog signal a) shown in Figure 7. For the purpose of this example, we will assume that the signal to be sampled is a single frequency sinusoidal signal.

Part b) of Figure 7 shows black dots representing the discrete samples collected by the acquisition device, overlaid on the original signal. In this case, the sampling frequency is such that we are collecting two samples per signal cycle. In other words, we are collecting samples at twice the signal frequency. By connecting the black dots, it is easy to see that we are identifying alternating amplitudes at the frequency of the original signal. Let´s see now what happens if we collect samples at a lower frequency. This lower sampling frequency is characterized by a longer period between samples ***T***. Part c) of Figure 7 shows again the black dots representing the samples, which are now being captured at different locations in the original signal. A direct consequence of this is that when we try to reconstruct the digitized signal, we will "see" a different signal (green). By comparing it to the original signal (red), it is evident that the frequency of the reconstructed signal is lower than the original. The original high frequency signal is said to have "aliased" to a lower frequency signal. At this point, it is important to state that once the aliasing of a signal has occurred, there is no way to know whether the reconstructed signal is real or not.



Original analog signal

a) Digitized signal, sampled at twice the original signal frequency.

b) Aliased lower frequency signal (green), resulting from a low sample rate.

Figure 7: Aliasing of a digitized signal.

Going back to case b) of the previous example, the fact that we chose to collect two samples per signal cycle is actually based on the "***Nyquist sampling theorem***", a corner stone of signal sampling. This theorem states that in order to accurately extract the frequency information from the original signal, the sampling frequency or sampling rate must be "***at least***" twice the highest frequency of interest in the original signal. As shown in the previous example, this requirement reduces the possibility of erroneously representing the original signal with another signal of a lower frequency or "aliasing".

ANTIALIASING FILTERS

Even when the Nyquist theorem ensures that we detect the frequencies of interest, undesired higher frequency components may still generate aliased components that will come up within our frequency span. Let´s suppose we want to examine frequency activity taking place between zero and 1000 Hertz. The Nyquist theorem will force us to sample at twice the maximum desired frequency, in this case, 2000 Hertz. This will warranty that any frequency component up to 1000 Hz will appear in the reconstructed signal, which would make us happy. However, what would happen if the analog signal included for example a random 1600 Hertz component? Signal sampling theory tells us that any frequency component $\boldsymbol{f_0}$ above half the sampling frequency will generate an aliased component at ***sampling frequency –*** $\boldsymbol{f_0}$. In our example, a 1600 Hz component will generate an aliased component at 2000Hz – 1600Hz = 400Hz. When this happens, we will observe a 400Hz frequency component which is not real, causing a tremendous waste of time trying to relate it to a real problem. In order to avoid this, ***antialiasing filters*** are usually applied to the original signal "before" the sampling process, as it has been depicted in Figure 1 of this paper. These are basically low pass filters that remove frequency components above our frequency span, eliminating the chance of aliasing.

However, when talking about antialiasing filters, some considerations should be made. Even when we would like these filters to look like figure # 8 a), passing all the desired frequencies and rejecting any higher frequencies, real filters usually exhibit behaviors such as the one in Figure # 8 b). Here, a transition band is present, within which frequency components are gradually attenuated. To avoid the possibility of components within this transition band aliasing into our frequency span, the sampling rate is set to twice the highest frequency in the transition band. We will come back to this later.

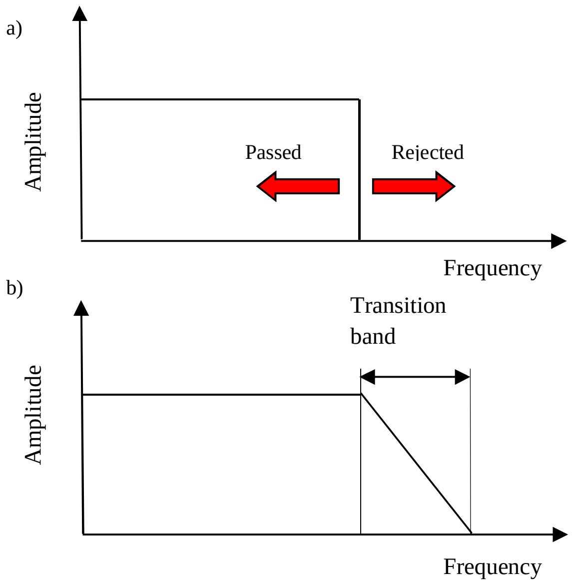


Figure 8: Ideal and real antialiasing filters frequency response.

FREQUENCY DOMAIN

So far we have been discussing the process of transforming an analog signal into a digital one. Both the analog and digital representation of a signal is usually referred to as “Time domain”. This type of plots constitute a direct representation of the digitized signal, either as a single signal or as an orbit, which is a combination of two signals coming from a pair of orthogonal transducers. In other words, no further processing, other than digitizing the analog signals, is performed when generating this type of plots. However, when dealing with vibration diagnostics, useful information can be extracted from the well-known “spectrum plots”. When we observe this type of plot, we are actually viewing the “frequency domain” of the vibration signal. In contrast with time domain, it does require additional manipulation or processing of the original digitized waveforms. Without getting into too much theoretical detail, let´s take a look at the basics associated with frequency domain processing. As we stated in the previous paragraph, the first step is to actually sample the original analog signal. In section 3, we called this group of samples “time record”. Then, a computational algorithm called Fast Fourier Transform, or “FFT”, is applied to this time record. This algorithm is based on the fact shown by Jean Baptiste Fourier, that any waveform encountered in the real world can be reproduced by adding up sine waves. By applying this algorithm, we go from a time domain representation of the signal (waveform), to the frequency domain representation of that signal, in order to get the frequency spectrum of that signal (a representation of all the frequency components that exist in the original complex signal, along with their respective amplitudes).
Let´s take a closer look now at this algorithm. Assuming that our time record is formed by ***N*** equally spaced samples, the FFT transforms them into ***N/2*** equally spaced samples in the frequency domain, called “lines”. Figure 9 shows, as an example, a digitized signal with a dominant 55Hz component with the resulting frequency spectrum. The number of samples ***N*** is always a power of 2 (2, 4, 8… 1024, 2048, etc.) as it makes the FFT algorithm simpler and faster.

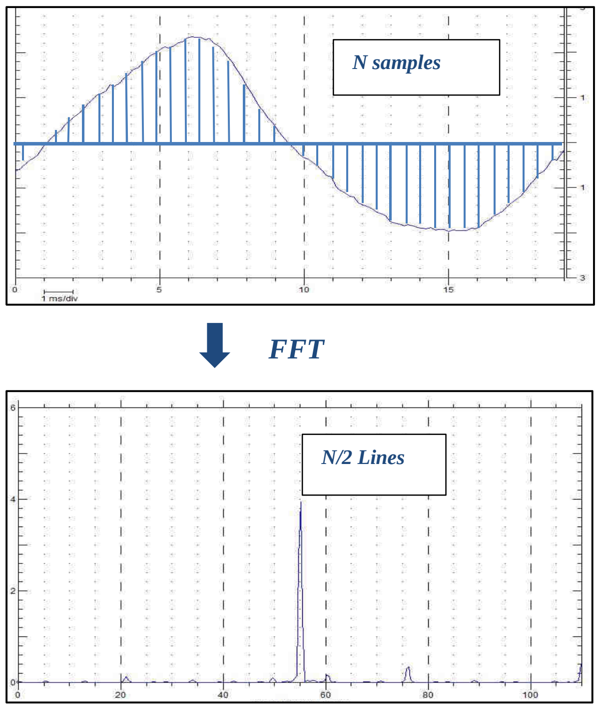


Figure 9: FFT being applied to a digitized signal.

So now we are in the frequency domain, with our ***N/2*** lines. Then the next question arises: what is the spacing between each of those lines?. As all the lines are equally spaced, then to answer the question, we just need to find out what is the spacing for the first line, i.e. between zero Hertz and the first line. As this is typically the lowest frequency we can resolve, it has the longest period ***T*** (the reciprocal of its frequency). We also need to consider that in order to precisely identify any frequency component in the digitized signal; we must be able to identify its period (the duration of one complete cycle). Then it follows that this particularly long period has to be included in our time record, otherwise, we won´t be able to properly identify this lowest frequency component. Summarizing we can conclude that the lowest frequency we can resolve with the FFT algorithm is the reciprocal of the time record period, or ***1/T*** (see figure 10).

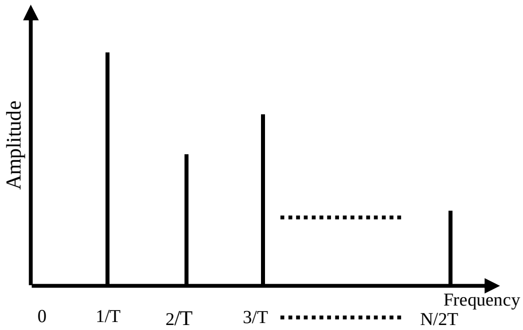


Figure 10. FFT resolution.

So far we know that the FFT delivers N/2 lines, and that those lines are equally spaced by the reciprocal of the time record period ***T***. This information allows us to easily calculate the highest frequency we can measure, using the following expression:

$$f_{\max} = \frac{N}{2} \cdot \frac{1}{T} \qquad (1)$$

Equation (1) shows that the maximum frequency we can view in our spectrum depends both on the number of samples in the time record ***N*** and the total time to complete the time record ***T***. Also, we now know that ***T*** depends on how fast the individual samples are collected, i.e. the ***sampling rate***. Additionally, we have to remember that the FFT delivers ***N/2 frequency lines***. Summarizing, we have a mathematical expression relating 5 variables: $\boldsymbol{f_{max}}$, ***N***, ***T***, sample rate and number of lines!. At this point, the reader must be wondering how all this comes down to reality, when trying to configure a data acquisition device to collect good vibration data.

Fortunately, most, if not all, data acquisition devices, limit these variables down to only 2: ***Frequency Span (FS) and number of lines (N/2).*** Note that the frequency span is not abbreviated $\boldsymbol{f_{max}}$, but ***FS***. This is not a text error. There is an explanation for this minor difference, and it involves what we have discussed in sections 4 and 5: Aliasing and Antialiasing filters. In order to better deal with this, let´s work with some real numbers in the following example.

Let´s suppose that our time record is formed by 1024 (remember, always power of 2 numbers) equally spaced samples in the time domain. If we apply the FFT to this time record, we will get 512 usable lines in the frequency domain (see figure 11). Those 512 will be distributed along our total frequency span ($\boldsymbol{f_{max}}$). However, in section 5 we learned that there is a transition band for the antialiasing filter, which is not really useful for us. For this reason, we just leave it out. But how much do we discard?. In our example, we keep 400 from the available 512 lines. The missing 112 lines used to be enough to cover the transition band of the old antialiasing filters, and even when modern data acquisition devices are capable of thinner transition bands, the ratio between 512 and 400 has been adopted. Then, our usable Frequency Span will be ***FS***.

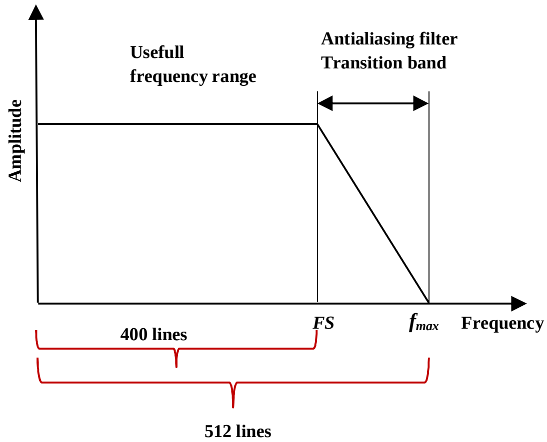


Figure 11: Theoretical and actual numbers of lines in the FFT.

As the filter characteristics were always the same, the slope of the transition band was also the same, and so was the ratio between the total and the real number of lines. Table 12 shows typical settings available in most instruments.

Table 12. Historical Numbers of Lines typically available today.

Time record size	Original FFT lines	Actually used FFT lines
256	128	100
512	256	200
1024	512	400
2048	1024	800
4096	2048	1600
8192	4096	3200
16384	8192	6400

Going back to figure 11, the Nyquist criteria requires that in order to properly identify the highest frequency in the FFT (f_{max}), we have to sample at twice this frequency. Additionally, both from figure 11 and table 12, we can see that f_{max} is exactly 1,28 x ***FS.*** Then it follows that, by Nyquist:

$$F_S = 2 \cdot f_{\max} = 2 \cdot 1{,}28 \cdot FS = 2{,}56 \cdot FS$$

or $$F_S = 2{,}56 \cdot FS \qquad (2)$$

Equation (2) represents an important relationship between the frequency span and the sampling frequency at which individual samples are collected to fill up the time record.

In table 12 it is possible to see, by comparing the time record size and the actual number of lines, the same factor of 2.56, leading to another important expression:

$$Time\,record\ size = 2{,}56 \times N^0\ lines \qquad (3)$$

Basically, Equation (3) states that the more lines, the more samples in the time record. Those samples are then collected at the sample frequency given by Equation (2). Let´s apply these relationships to a real example:

We want to analyze a frequency span of 500Hz, but we also want good resolution, so we choose 3200 line for our FFT. By using Equation (2), we determine the sampling frequency F_s of 1280 samples/sec. Then, we use Equation (3) to determine the time record size, which then equals 8192 samples. In this scenario, we will have to collect 8192 samples at a rate of 1280 samples each second. The total sampling time can easily be calculated as:

$$\frac{8192\ samples}{1280\ {}^{samples}\!/_{\sec}} = 6.4\sec \qquad (4)$$

Now 6.4 seconds may not seem a long time for everyday activities. However, when collecting data in a rotating machine, 6.4 seconds represent really a long time, especially when we are trying to detect some peak in vibration, or when we are collecting data during machine shutdown. Remember, in order to get a good quality FFT, the signal included in the time record must be periodic. Now that we have already discussed the properties and requirements for the FFT, let´s see what tools we can use when a periodic signal is not readily available.

WINDOWING

Every time we try to configure our data acquisition device in order to look at a spectrum plot, we are prompted the question of what type of window to use. In order to understand the concept of windowing, let´s go back to the FFT algorithm. This algorithm assumes that the input signal is completely periodic, repeating itself over and over from one time record to the next. In the real world, there are instances in which this is true (we will deal with these in section 8), and others in which it is not. But let´s see some examples of both cases.

Figure 13 shows a time record for a digitized signal that contains exactly 10 cycles, i.e., an integer number of cycles. This means that when the algorithm tries to copy the signal portion over and over to get a perfectly periodic waveform, it succeeds. Then, the resulting spectrum plot will show clean frequency lines with the proper information, as shown on Figures 14 and 15.

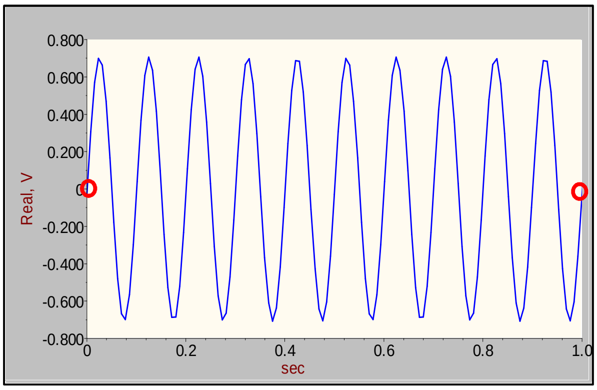


Figure 13: Signal with integer number of cycles.

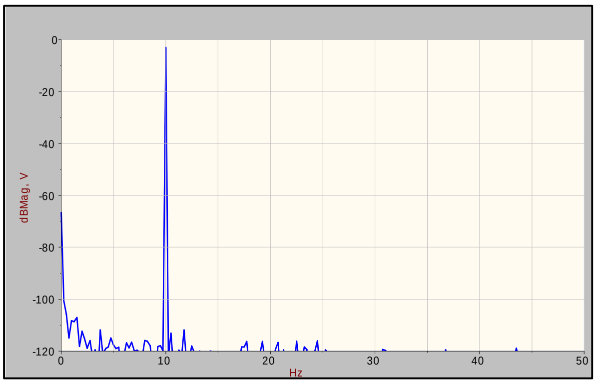


Figure 14: FFT applied to signal in Figure 13, vertical axis in dB.

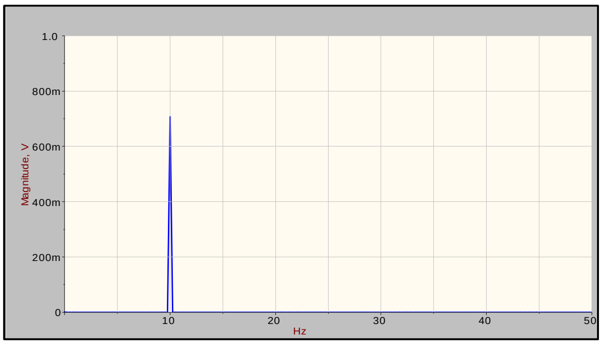


Figure 15: FFT applied to signal in Figure 13, vertical axis in magnitude.

If we now consider a signal time record that it includes a non-integer number of cycles, it is evident that there will be signal discontinuities at the edges, as shown in Figure 16. It is important to say that the original analog signal in these two examples is the same. We are only changing the way it is being sampled into two different time records. This deviation from the ideal periodic signal will cause the FFT algorithm to throw undesired results, known as "leakage" (see Figures 17 and 18). This term is associated to the fact that the energy from the original frequency line is "leaking" out to adjacent lines, even in those cases in which there is a single frequency component. If we are trying to identify specific component in order to do vibration diagnostics, leakage will definitely be an issue. If the problem we are analyzing involves frequency components that are really close, for example the two times line and two times rotation frequency in an asynchronous electric motor, leakage will make it virtually impossible to determine which one exhibits the highest amplitude.

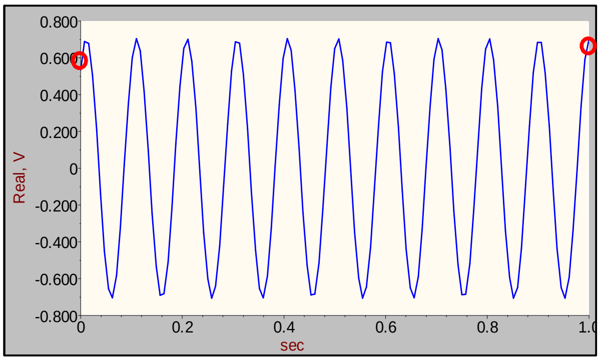


Figure 16: Signal with non-integer number of cycles.

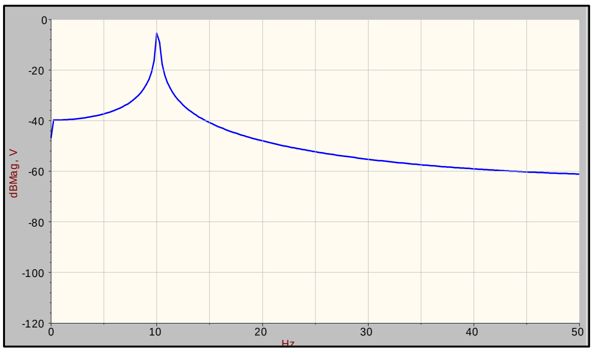


Figure 17: FFT applied to signal in Figure 16, vertical axis in dB.

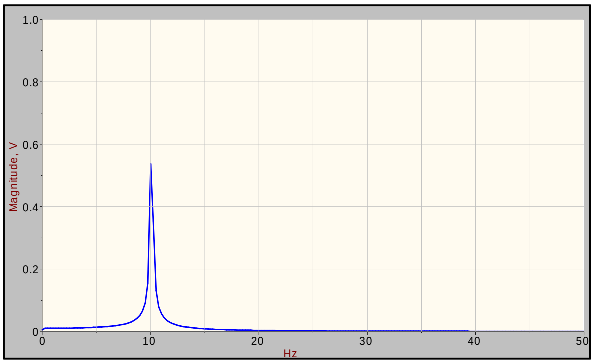


Figure 18: FFT applied to signal in Figure 16, vertical axis in magnitude.

Even though leakage cannot be fully eliminated, it can definitely be attenuated. The most common way to do this is by the use of "windows functions". The ultimate goal of using windows is to ensure that the digitized signal amplitude becomes zero at the beginning and at the end of the time record. By achieving this, we attenuate the discontinuities at the edges, responsible for causing leakage. Though there are many types of functions currently used for this purpose, let´s consider one generic windowing function that is zero at the ends and maximum amplitude in the middle. When we apply this window function to our time record signal, we are actually multiplying both, in order to get a vibration signal somehow different to the original, in the sense that its frequency remains the same, but the amplitude has been forced to zero at both ends. Figure 19 shows a comparison between not using windows and using them.

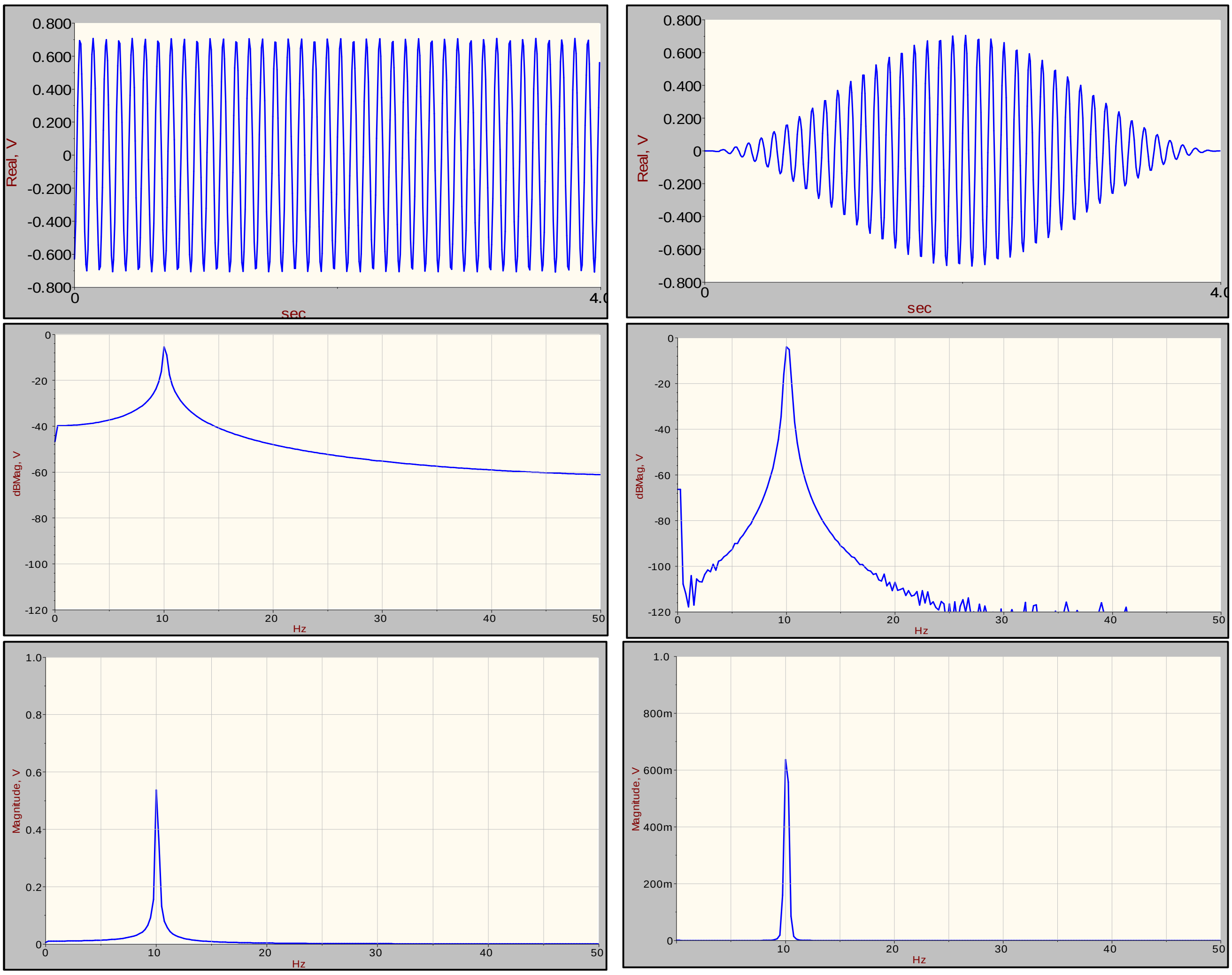


Figure 19: Non periodic signal without windowing (Left) and using Hanning Window (Right).

Even though in most cases, the time record doesn´t contain an integer number of vibration cycles, forcing us to use window functions, there are cases in which the time record does cover an integer number of these cycles. If we also consider rotating equipment in general, most of the signal frequency content is usually related to rotating speed, through some ratio (one time, two times, etc.). In cases like these, it would be useful that our sampling rate has also some type of relationship with rotation, some kind of synchronization. That type of sampling is actually available, and it is called “synchronous sampling”, and we will deal with it in the next section.

SYNCHRONOUS OR ANGULAR SAMPLING

According to what we have been discussing throughout this paper, it would be logical to arrive at the conclusion that if we want to get a good representation of the original analog signal, we just need to make sure we select a sampling frequency high enough (assuming we have a good quality ADC). However, this is not completely true. In contrast with the typical scenario for data collection, in which the machine rotating speed is fairly constant during the data acquisition (typical of small balance of plant machines), we are sometimes faced with the task of sampling our vibration signals during some transient event, such as a machine coast down. Though this type of transient events might be considered as uncommon, since it is usually expected that machines keep on running most of the time, the most useful information for diagnostic purposes is actually gathered during these events.

It turns out that the most common lateral vibration being generated in rotating equipment is the one associated with the unbalance force. This force happens to be locked into the rotation of the shaft (the heavy spot on the rotor is located at a fixed angular location). Under these conditions, the dominant vibration occurs at one time running speed, usually referred to as 1X, with X representing rotating speed. This also means that during machine shutdown, this 1X frequency component will track running speed as it decreases. Let´s suppose we want to digitize the vibration signal resulting from an unbalanced rotor, using the already discussed time based sampling, i.e. collecting samples that are equally spaced in time, during machine shutdown. Figure 20 shows the time based sampled signal. Because the speed is decreasing, and also the dominant vibration frequency, we can clearly see that each vibration cycle takes a longer time to complete. In this scenario, two major issues arise: a) due to the fact that the samples are being collected at regular time intervals, the quality of the digitized signal improves throughout the time record as rotating speed decreases; and b) the actual frequency of the signal changes throughout the time record. This last statement is actually the most important as far as quality of the data, since as we learned in the previous section, the less periodic a signal, the stronger the leakage or smearing effect in the corresponding time domain FFT (see figure 21).

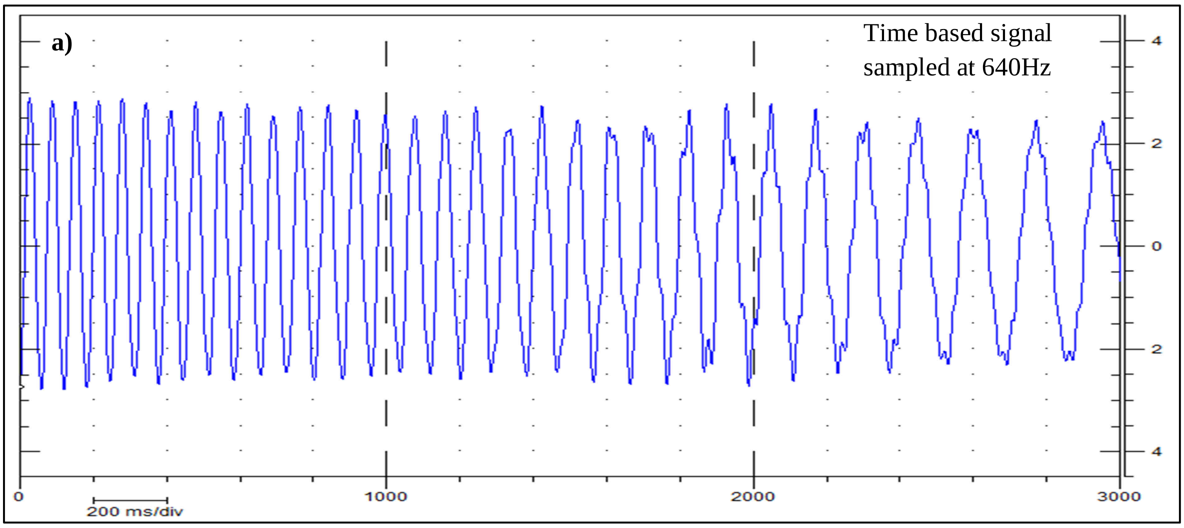


Figure 20: Time-based sampled signal.

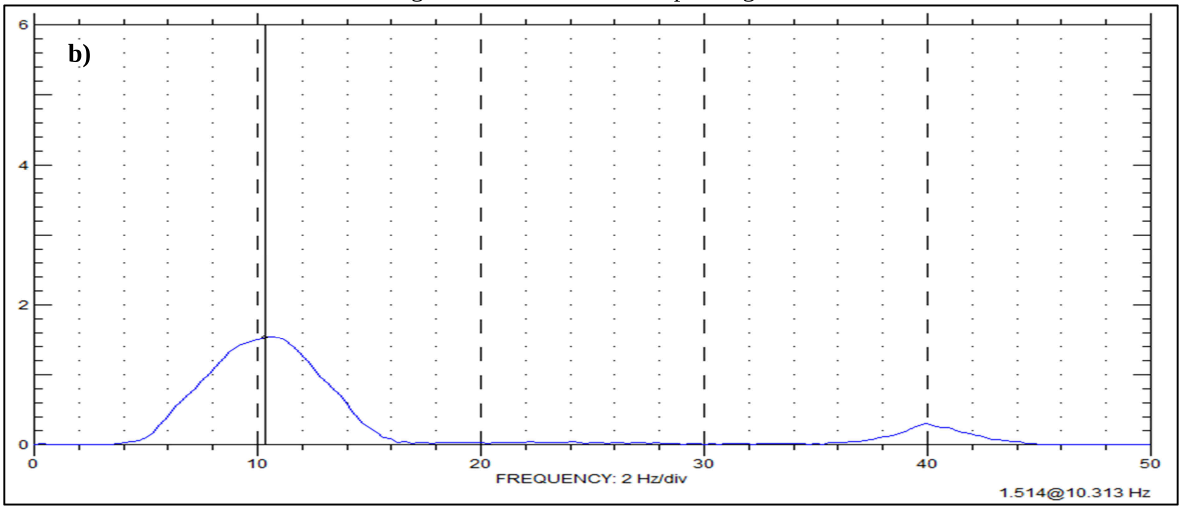


Figure 21: FFT "smearing effect".

In order to avoid this phenomenon, we have two options. We can select a higher sampling frequency, causing the time record to be shorter (minimizing the chance to get changes in frequency), or, we can use another type of sampling, the "synchronous sampling". In contrast with the time based sampling, synchronous sampling is based on the angle of rotation. This is why it is also called Angular sampling. Basically, the signal portion included in each revolution is digitized in such a way, that the amount of samples per revolution is always the same, independently of what the rotational speed is. In other words, the individual samples are captured at regularly spaced "angle" intervals, as opposed to regularly spaced "time" intervals. Under these conditions, while collecting data during startup or shutdown event, any component that tracks running speed (1X, 2X, etc.) will look "as if" it had a constant frequency in the angle domain. Then, the angle domain FFT will properly show a well-defined peak. This type of analysis is usually referred to as "order tracking". Focusing on the practical application point of view, this type of sampling requires that both the vibration and a robust tachometer (either once-per-turn or multi-event) signals are sampled at a very high frequency (in the MHz range). Then, using the tachometer signal as reference, the data is resampled in order to obtain the signal amplitude as a function of shaft rotation or angle.

Figure 22 shows the same signal from figure 20, but synchronously sampled at 64 samples per revolution, using a once-per-turn reference signal (see figure 23). Even though we can still observe that the dominant frequency is decreasing through the time record, the resulting FFT looks much better than the one in figure 21 (see figure 24). The reason for this is that even though the dominant frequency is decreasing as seen in the time domain, it is actually constant in the angle domain (it´s always 1X). The frequency unit used in the angle domain FFT is "orders" (1X, 2X, etc.).

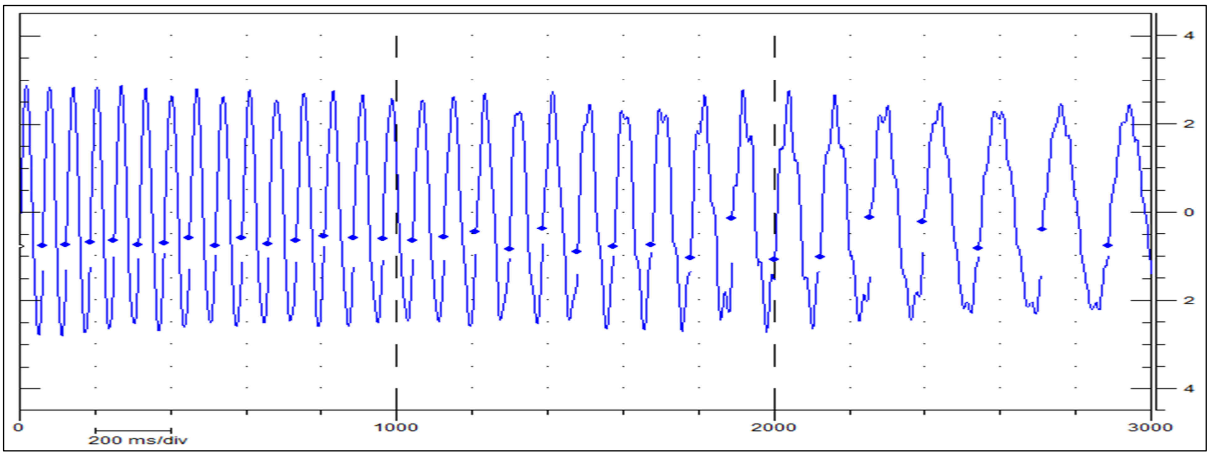


Figure 22: Angle-based sampled signal (64 samples per revolution).

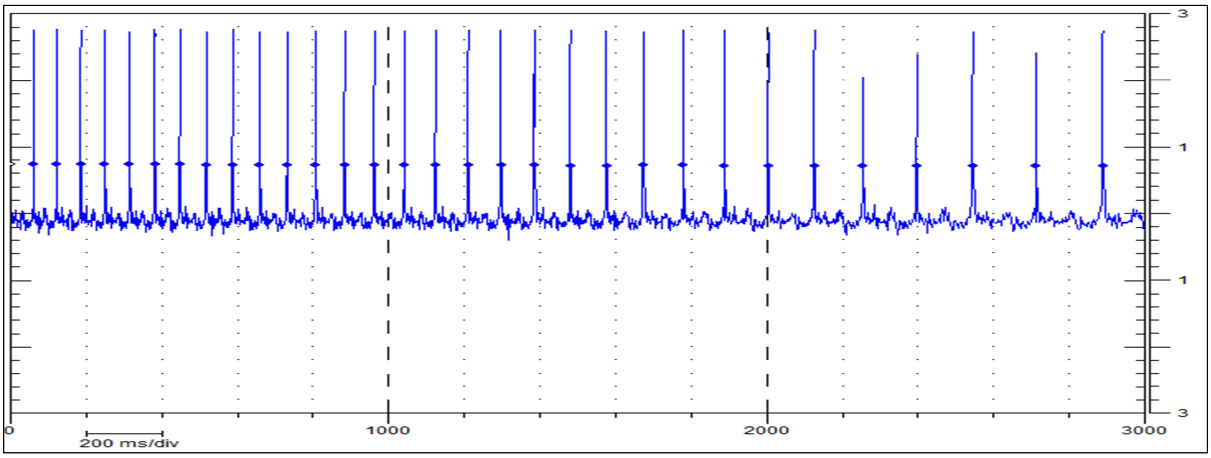


Figure 23: Once-per-turn reference signal.

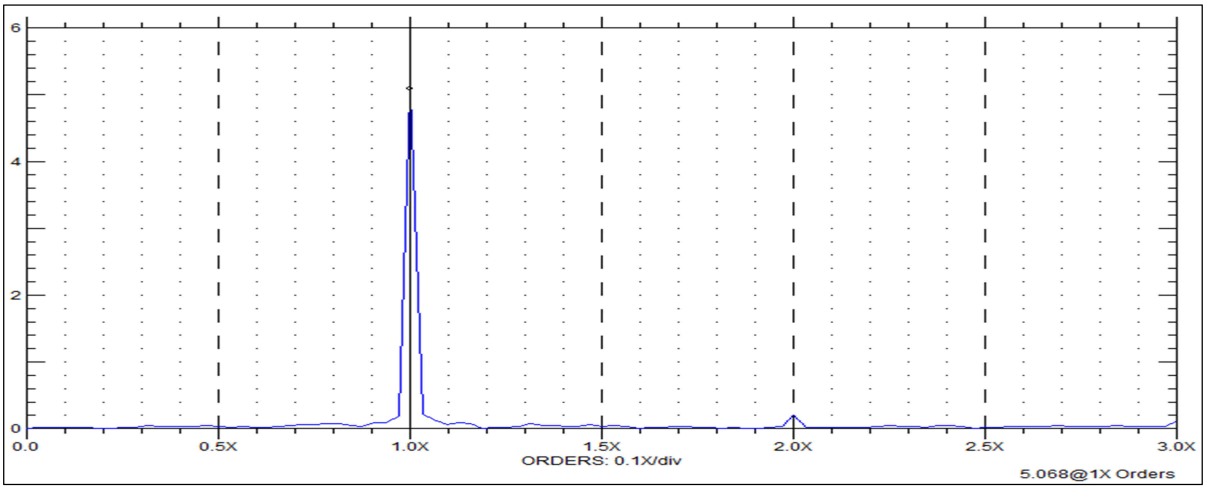


Figure 24: Angle Domain FFT.

Now that we are discussing data collection during transient events such as machine start up or shut down, we cannot go any further without talking about one of the most widely used plots: the Bode plot. This plot shows how the 1X vibration (both amplitude and phase) change as a function of rotating speed. Even though this is a well-known plot, the signal processing associated with it is sometimes overlooked. So let´s see what type of additional signal processing is required to accurately measure 1X amplitude and phase.

DIGITAL TRACKING FILTERS

We already talked about filters, when we discussed antialiasing filters. These filters are located "before" the digitizing process. In other words, they are analog low pass filters. However, once the signal has been digitized, digital filters become available (either at hardware or software level). In this case, we are interested in a specific type, "band pass filters", designed in such a way that only a frequency band is allowed to pass, attenuating higher and lower frequencies. Figure 25 shows a typical band pass filter profile, along with two important parameters: bandwidth (***B***) and center frequency (f_c).

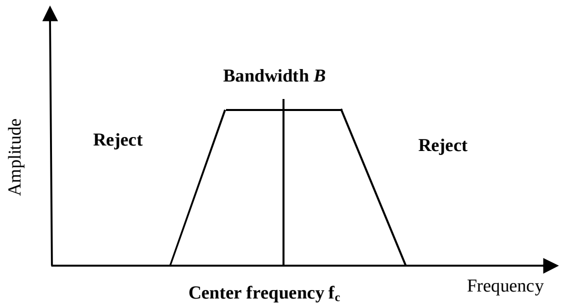


Figure 25: Typical bandpass filter profile.

A tracking filter is a special type of bandpass filter, in which the center frequency f_c actually tracks the machine rotational speed (and any desired multiple). In this fashion, 1X, 2X and any other "preconfigured" multiple of running speed can be used to perform order analysis, i.e. to evaluate how these frequency components behave during a startup or shutdown event. However, as usual, there is a catch. In general frequency analysis, there is a relationship between the bandwidth ***B*** and the time required to process the data ***T*** (this time is the length of the time record we have been discussing in previous sections), given by the following expression:

$$B \cdot T \geq 1 \qquad (5)$$

Equation (5) basically tells us that the narrower the filter bandwidth, the longer the time to process the data. But since the purpose of this paper is to relate theory to practical applications, let´s see some examples on how the bandwidth ***B*** affects the quality of the vibration data.

Figures 26, 27 and 28 show three Bode plots for the same vibration transducer, considering three different bandwidths for the 1X tracking filter: 2Hz (120cpm), 0.2Hz (12cpm) and 0.02Hz (1.2cpm). For this example, we are collecting data on a slow ramp rate machine, such us a 400MW steam turbine generator starting up in cold conditions. The ramp rate in this case is **228rpm/Min**. Though the 120 and 12cpm filters seem to provide similar 1X data, the 1.2cpm tracking filter is measuring a lower amplitude value at the critical speed.

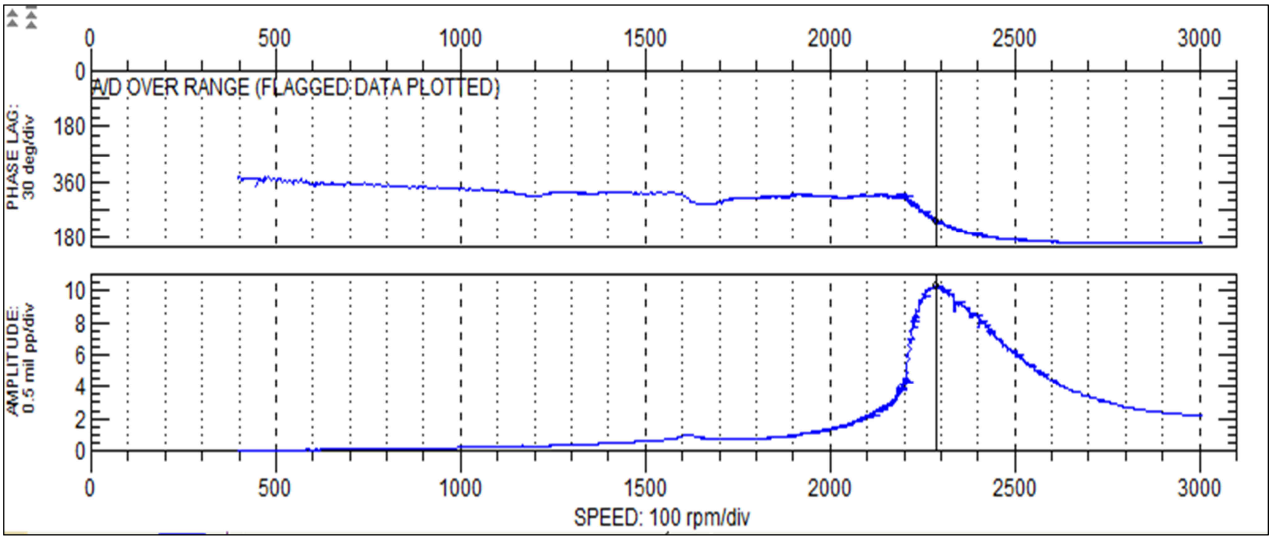


Figure 26: Bode plot for a slow ramp rate machine, using a bandwidth of 120cpm.

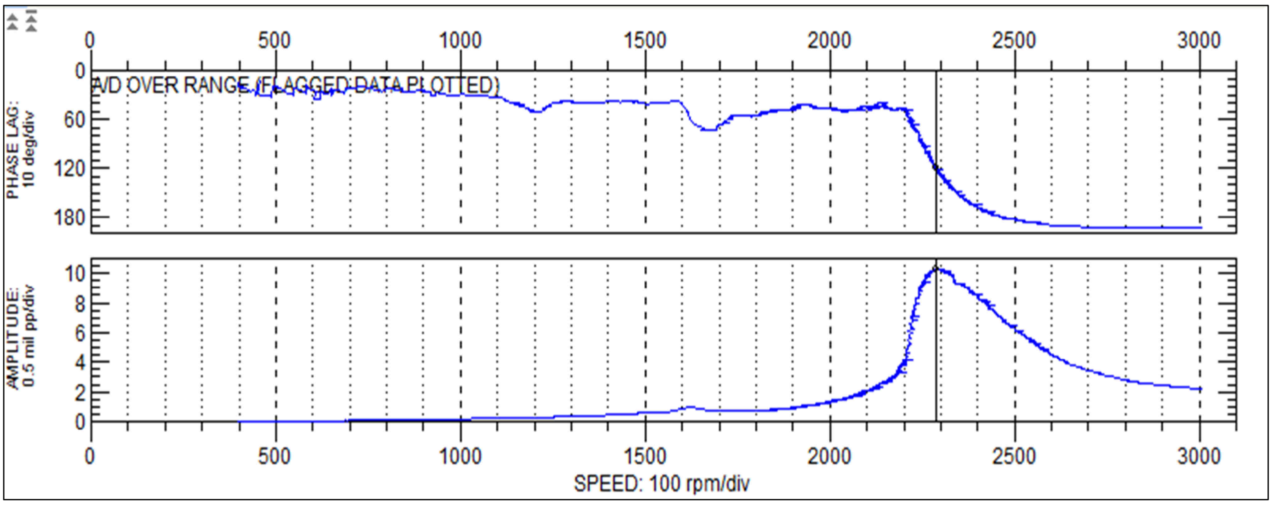


Figure 27: Bode plot for a slow ramp rate machine, using a bandwidth of 12cpm.

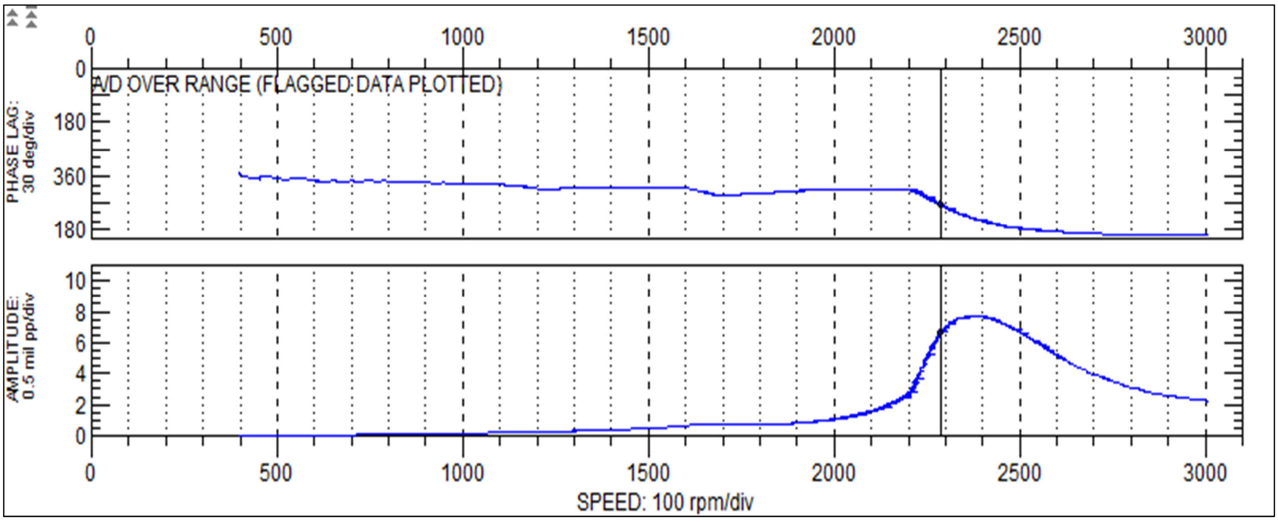


Figure 28: Bode plot for a slow ramp rate machine, using a bandwidth of 1.2cpm.

Now let´s move our data acquisition device to a fast ramp rate machine. The best example for this type of machine in the field is the direct drive asynchronous motor. Bode plots included in figures 29, 30 and 31 correspond to an **8082rpm/min** ramp rate, quite typical of these machines. Now we can easily see that only the 2Hz bandwidth filter is able to provide proper data. The 0.02Hz bandwidth filter, on the other hand, is completely unable to detect the maximum amplitude at the critical speed.

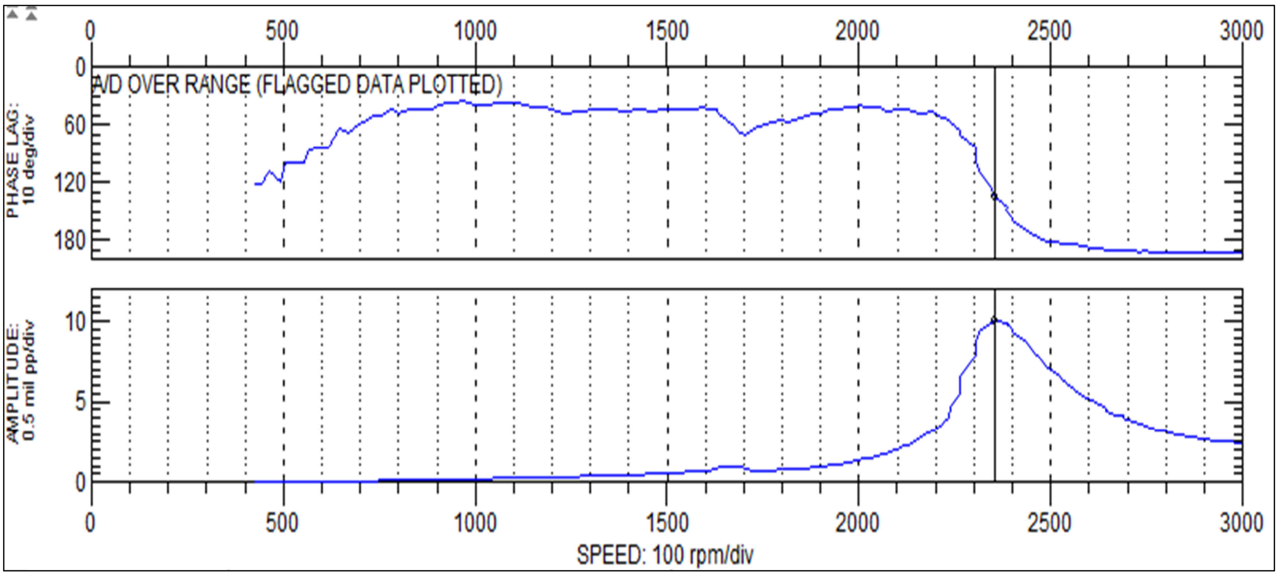


Figure 29: Bode plot for a fast ramp rate machine, using a bandwidth of 120cpm.

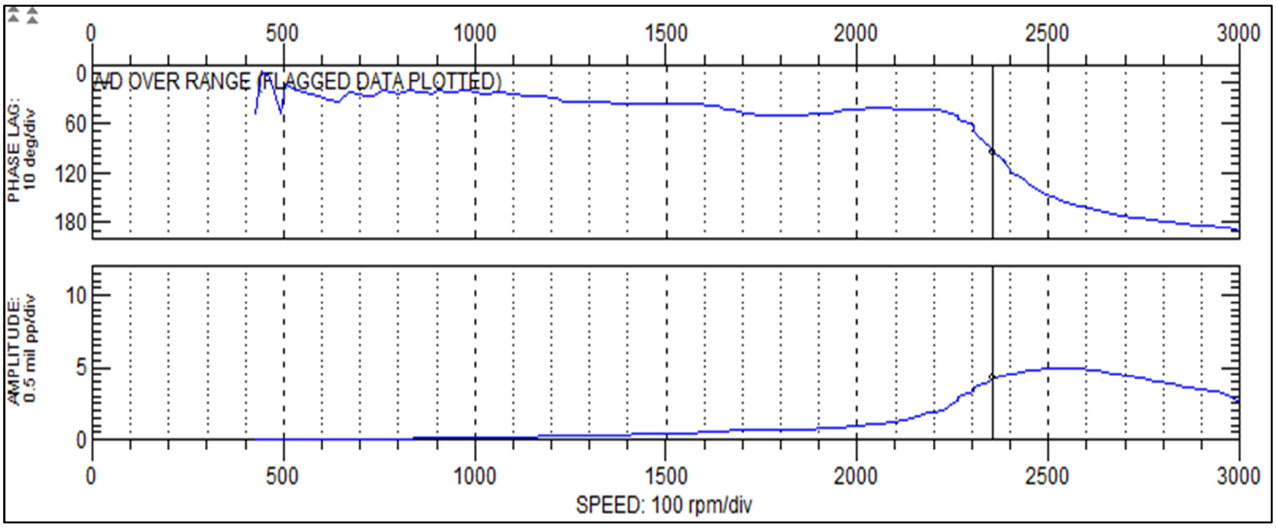


Figure 30: Bode plot for a fast ramp rate machine, using a bandwidth of 12cpm.

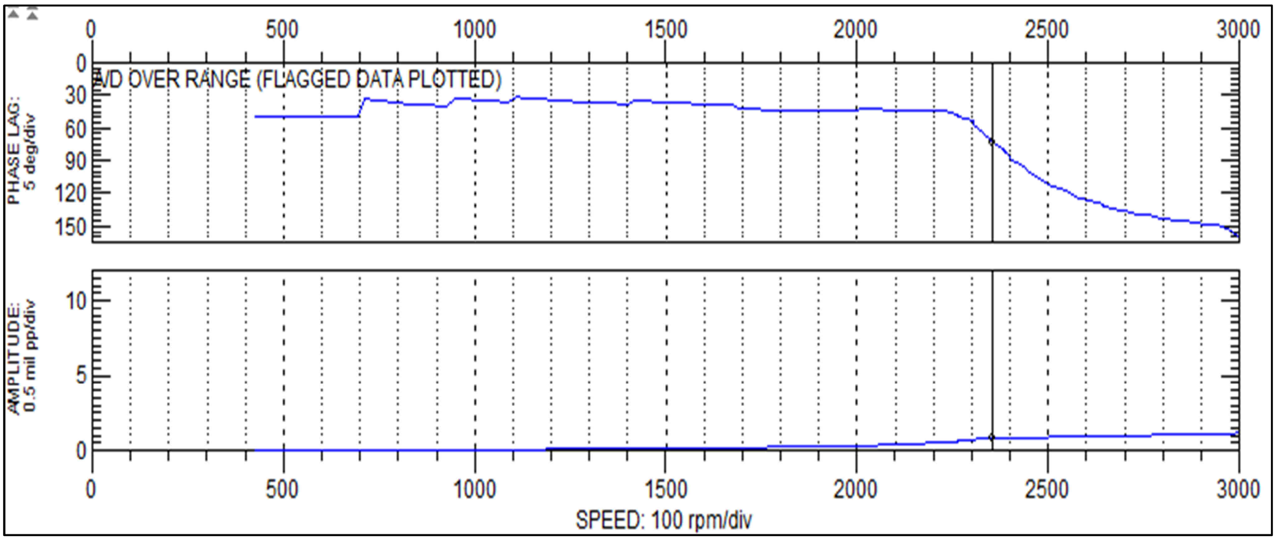


Figure 31: Bode plot for a fast ramp rate machine, using a bandwidth of 1.2cpm.

So what is wrong in the previous example?. Let´s examine equation (3) again. Let´s rearrange it in the following way:

$$B \geq \frac{1}{T} \tag{6}$$

Using equation (6), we can determine "the minimum" time ***T*** for the filter to provide the correct output, for the three bandwidths used in the examples.

$$2Hz(120cpm) \geq \frac{1}{0.5\sec} \tag{7}$$

$$0.2Hz(12cpm) \geq \frac{1}{5\sec} \tag{8}$$

$$0.02Hz(1.2cpm) \geq \frac{1}{50\sec} \tag{9}$$

From Equation (7), we can see that a 2Hz bandwidth filter requires at least a 0.5-second settling time (time required to provide a proper output). Then, a narrower filter with a bandwidth of 0.2Hz will need 5 seconds to settle (see Equation (8)), and an even narrower filter of 0.02Hz bandwidth will take as long as 50 seconds (see Equation (9)). In this last case, if we go back to the fast ramp rate example, the machine reaches nominal speed in about 21 seconds, roughly half the filter settling time. What actually happens is that the center frequency (rotating speed) is moving too fast for the filter to catch it, causing its output reading to be a frequency much lower than real 1X. This is why the displayed 1X amplitude appears to be extremely low throughout the speed range in figure 31.

CONCLUSION

The use of computer based signal processing instrumentation and software is common across practically every industry and test engineering application today. Although rotating machinery diagnostics has its foundation in mechanical engineering concepts, the diagnostics practitioner needs to be knowledgeable in signal processing concepts and methods in order to ensure that the highest quality data is always acquired and analyzed in the most optimal manner. Important concepts such as antialiasing, time and frequency resolution, windowing, synchronous sampling and filtering were presented. But most importantly, the notion that "nothing is for free in signal processing" has also been discussed. High frequency resolution always requires time to collect data, becoming a critical issue when dealing with transient conditions, such as machine startup or shutdown events. Even though current instrumentation and software provide configurable options for high resolution, the diagnostics Engineer must understand how to manage these options.

NOMENCLATURE

AAF = Antialiasing Filter
ADC = Analog to Digital Converter
FFT = Fast Fourier Transform
Mil pp = Vibration amplitude (thousand of an inch) peak to peak
T = Period, time between samples
FS = Frequency Span
F_S = Sampling Frequency
1X = Frequency equal to running speed
B = Bandwidth
f_c = Center Frequency
dB = Decibel, unit of ratio or gain

BIBLIOGRAPHY

- "The Fundamentals of Signal Analysis", Hewlett Packard, Application Note 243.
- Richardson M., 1978, "Fundamentals of the Discrete Fourier Transform", Sound and Vibration Magazine.
- Anders B. et al, 2005, "Main Principles and Limitations of Current Order Tracking Methods", Sound and Vibration.
- Southwick D., 1998, "3200 Line Spectrum…when shouldn´t you use it?", Orbit Magazine.
- Southwick D., 1993, "Sampling Waveform and Computing Spectra", Orbit Magazine.
- Desimone G., 2011, "ADRE 408 DSPI Signal Processing", Orbit Magazine.

ACKNOWLEGEMENTS

The author would like to thank Arun B. Menon, GE M&C, for his valuable help in clarifying many of the concepts included on this paper, as well as the use of signal simulating equipment.