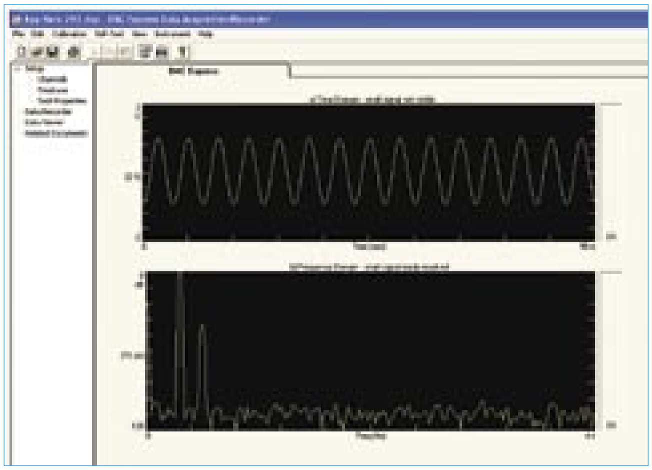


Figure 2.7

Fundamentals of Dynamic Signal Analysis

Introduction

This note is a primer for those who are unfamiliar with the advantages of analysis in the frequency and modal domains and with the class of digitizers and analyzers designed for Dynamic Signal Analysis.

The Time, Frequency and Modal Domains:

The traditional way of observing signals is to view them in the time domain. The time domain is a record of what happened to a parameter of the system versus time. It was shown over one hundred years ago by Baron Jean Baptiste Fourier that any waveform that exists in the real world can be generated by adding up sine waves. By picking the amplitudes, frequencies and phases of these sine waves correctly, we can generate a waveform identical to our desired signal. Since we know that each line represents a sine wave, we have uniquely characterized our input signal in the frequency domain. This frequency domain representation of our signal is called the spectrum of the signal. Each sine wave line of the spectrum is called a component of the total signal. It is very important to understand that we have neither gained nor lost information, we are just representing it differently. We are looking at the same three dimensional graph from different angles. This different perspective can be very useful.

Suppose we wish to measure the level of distortion in an audio oscillator. Or we might be trying to detect the first sounds of a bearing failing on a noisy machine. In each case, we are trying to detect a small sine wave in the presence of large signals. Figure 2.7a shows a time domain waveform which seems to be a single sine wave. But Figure 2.7b shows in the frequency domain that the same signal is composed of a large sine wave and significant other sine wave components (distortion components). When these components are separated in the frequency domain, the small components are easy to see because they are not masked by larger ones.

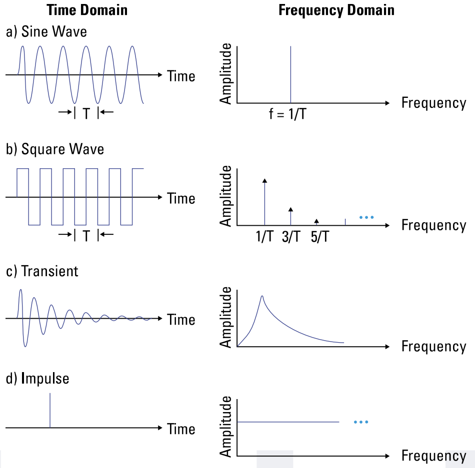


Figure 2.10

Spectrum Examples

Let us now look at a few common signals in both the time and frequency domains. In Figure 2.10a, we see that the spectrum of a sine wave is just a single line. We expect this from the way we constructed the frequency domain. The square wave in Figure 2.10b is made up of an infinite number of sine waves, all harmonically related. The lowest frequency present is the reciprocal of the square wave period. These two examples illustrate a property of the frequency transform: a signal which is periodic and exists for all time has a discrete frequency spectrum. This is in contrast to the transient signal in Figure 2.10c which has a continuous spectrum. This means that the sine waves that make up this signal are spaced infinitesimally close together. Another signal of interest is the impulse shown in Figure 2.10d. The frequency spectrum of an impulse is flat, i.e., there is energy at all frequencies. It would, therefore, require infinite energy to generate a true impulse. Nevertheless, it is possible to generate an approximation to an impulse which has a fairly flat spectrum over the desired frequency range of interest. We will find signals with a flat spectrum useful in our next subject, network analysis.

Network Analysis

If the frequency domain were restricted to the analysis of signal spectrums, it would certainly not be such a common engineering tool. However, the frequency domain is also widely used in analyzing the behavior of networks. Network analysis is the general engineering

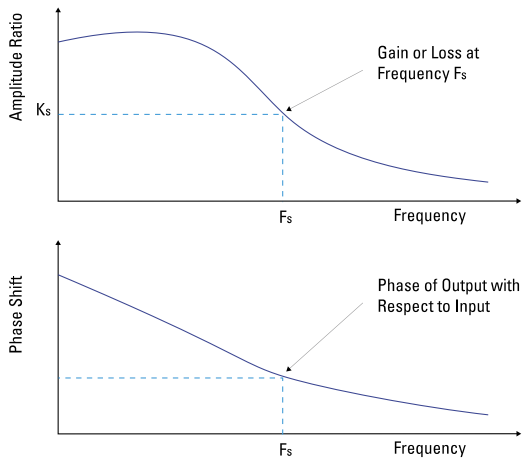


Figure 2.18

Fundamentals of Dynamic Signal Analysis

problem of determining how a network will respond to an input. Network Analysis is sometimes called Stimulus/Response Testing. The input is then known as the stimulus or excitation and the output is called the response. It is easy to show that the steady state response of a linear network to a sine wave input is a sine wave of the same frequency. The amplitude of the output sine wave is proportional to the input amplitude. Its phase is shifted by an amount which depends only on the frequency of the sine wave. As we vary the frequency of the sine wave input, the amplitude proportionality factor (gain) changes as does the phase of the output. If we divide the output of the network by the input, we get a normalized result called the frequency response of the network.

As shown in Figure 2.18, the frequency response is the gain (or loss) and phase shift of the network as a function of frequency. Because the network is linear, the frequency response is independent of the input amplitude; the frequency response is a property of a linear network, not dependent on the stimulus. The frequency response of a network will generally fall into one of three categories; low pass, high pass, bandpass or a combination of these. As the names suggest, their frequency responses have relatively high gain in a band of frequencies, allowing these frequencies to pass through the network. Other frequencies suffer a relatively high loss and are rejected by the network. To see what this means in terms of the response of a filter to an input, let us look at the bandpass filter case.

In Figure 2.20a, we put a square wave into a bandpass filter. We recall from Figure 2.10 that a square wave is composed of harmonically related sine waves. The frequency response of our example network is shown in Figure 2.20b. Because the filter is narrow, it will pass only one component of the square wave. Therefore, the steady-state response of this bandpass filter is a sine wave. Notice how easy it is to predict the output of any network from its frequency response. The spectrum of the input signal is multiplied by the frequency response of the network to determine the components that appear in the output spectrum. This frequency domain output can then be transformed back to the time domain. In contrast, it is very difficult to compute in the time domain the output of any but the simplest networks. A complicated integral must be evaluated which often can only be done numerically on a digital computer. If we computed the network response by both evaluating the time domain integral and transforming to the frequency domain and back, we would get the same results. However, it is usually easier to compute the output by transforming to the frequency domain.

Transient Response

Up to this point we have only discussed the steady-state response to a signal. By steady-state we mean the output after any transient responses caused by applying the input have died out. However, the frequency response of a network also contains all the information

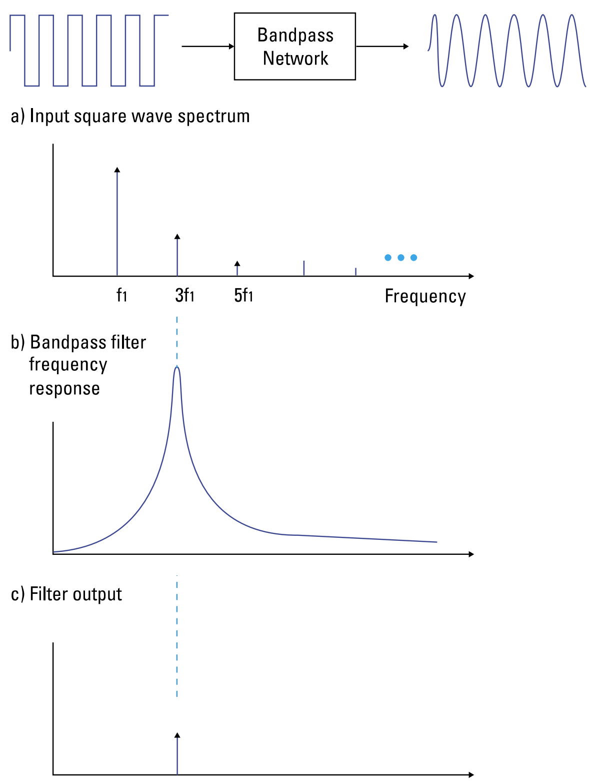
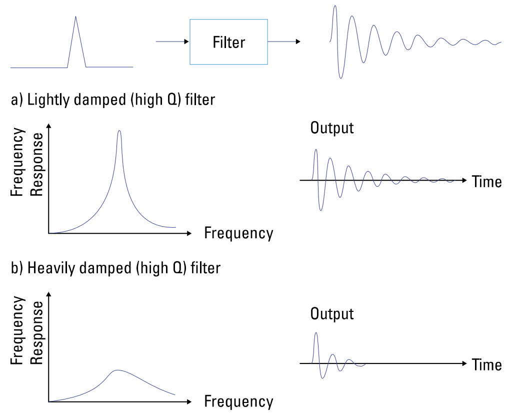


Figure 2.20

Figure 2.21



necessary to predict the transient response of the network to any signal. Let us look qualitatively at the transient response of a bandpass filter. If a resonance is narrow compared to its frequency, then it is said to be a high "Q" resonance*. Figure 2.21a shows a high Q filter frequency response. It has a transient response which dies out very slowly. A time response which decays slowly is said to be lightly damped. Figure 2.21b shows a low Q resonance. It has a transient response which dies out quickly. This illustrates a general principle: signals which are broad in one domain are narrow in the other. Narrow, selective filters have very long response times.

Instrumentation for the Modal Domain

Dynamic Signal Analyzer

Dynamic signal analyzers are based on a high speed calculation routine which acts like a parallel filter analyzer with hundreds of filters and yet are cost competitive with swept spectrum analyzers. In addition, a two or more channel (DSA) dynamic signal analyzer or Digitizers with DSP are in many ways better network analyzers than traditional network analyzers that are limited to slow tracking filters that limit measurement speed. The DSA provides the ideal instrumentation for time, frequency and network measurements.

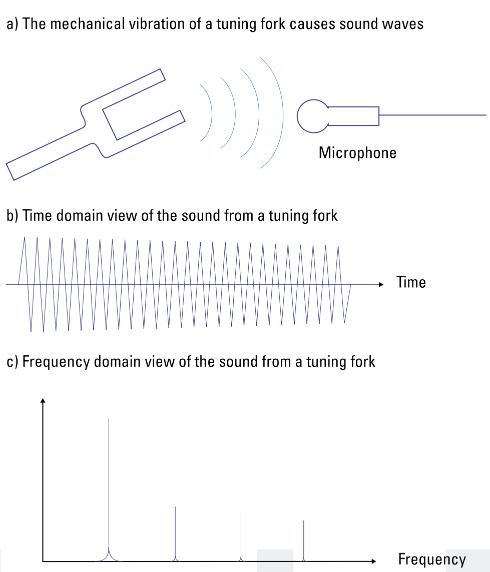


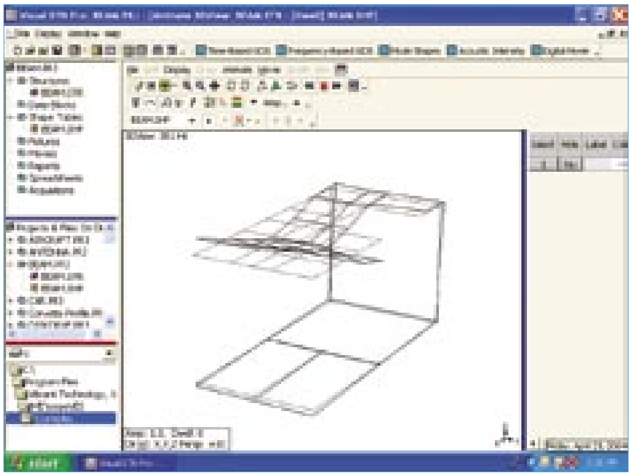
Figure 2.27

The Modal Domain

To understand the modal domain let us begin by analyzing a simple mechanical structure, a tuning fork. If we strike a tuning fork, we easily conclude from its tone that it is primarily vibrating at a single frequency. We see that we have excited a network (tuning fork) with a force impulse (hitting the fork). The time domain view of the sound caused by the deformation of the fork is a lightly damped sine wave shown in Figure 2.27b. In Figure 2.27c, we see in the frequency domain that the frequency response of the tuning fork has a major peak that is very lightly damped, which is the tone we hear. There are also several smaller peaks. Each of these peaks, large and small, corresponds to a "vibration mode" of the tuning fork. We can express the vibration of any structure as a sum of its vibration modes. Just as we can represent a real waveform as a sum of much simpler sine waves, we can represent any vibration as a sum of much simpler vibration modes. The task of "modal" analysis is to determine the shape and the magnitude of the structural deformation in each vibration mode. Once these are known, it usually becomes apparent how to change the overall vibration. Experimentally, we have to measure only a relatively few points on the structure to determine the mode shape. Each vibration mode is characterized by its mode shape, frequency and damping from which we can reconstruct the frequency domain view.

Instrumentation for the Modal Domain

To determine the modes of vibration, basically we determine the frequency response of the structure at several points and compute at each resonance the frequency, damping and what is called the residue (which represents the height of the resonance).
This is done by a curve-fitting routine to smooth out any noise or small experimental errors. From these measurements and the geometry of the structure, the mode shapes are computed and drawn on a computer display. These displays may be animated to help the user understand the vibration mode. From the above description, it is apparent that a modal analyzer requires some type of network analyzer to measure the frequency response of the structure and a computer to convert the frequency response to mode shapes. This can be accomplished by connecting a dynamic signal analyzer through a digital interface to a computer furnished with the appropriate software. This capability is also available



Modal Display

Fundamentals of Dynamic Signal Analysis

as an integrated system with digitizers designed for dynamic signal analysis. Because larger structures require more measurement channels the computer can become overloaded by the amount of data needed to process frequency response functions. Digitizers that include on board DSP, offload the computer and make it possible to use this technique with hundreds or even thousands of channels.

Summary

In this chapter we have developed the concept of looking at problems from different perspectives. These perspectives are the time, frequency and modal domains. Phenomena that are confusing in the time domain are often clarified by changing perspective to another domain. Small signals are easily resolved in the presence of large ones in the frequency domain. The frequency domain is also valuable for predicting the output of any kind of linear network. A change to the modal domain breaks down complicated structural vibration problems into simple vibration modes. No one domain is always the best answer, so the ability to easily change domains is quite valuable. Of all the instrumentation available today, only dynamic signal analyzers can work in all three domains. In the next chapter we develop the properties of this important class of analyzers.

Understanding Dynamic Signal Analysis

In this section we will develop a fuller understanding of dynamic signal analysis. We begin by presenting the properties of the Fast Fourier Transform (FFT) upon which dynamic signal analyzers are based. We then show how these FFT properties cause some undesirable characteristics in spectrum analysis like aliasing and leakage. Having demonstrated a potential difficulty with the FFT, we then show what solutions are used to make practical dynamic signal analyzers. Developing this basic knowledge of FFT characteristics makes it simple to get good results with a dynamic signal analyzer in a wide range of measurement problems.

FFT Properties

The Fast Fourier Transform (FFT) is an algorithm for transforming data from the time domain to the frequency domain. Since this is exactly what we want a spectrum analyzer to do, it would seem easy to implement a dynamic signal analyzer based on the FFT. However, we will see that there are many factors which complicate this seemingly straight-forward task. Note, however, that we cannot now transform to the frequency domain in a continuous manner, but instead must sample and digitize the time domain input. This means that our algorithm transforms digitized samples from the time domain to samples in the frequency domain as shown in Figure 3.3.

Time Records

A time record is defined to be N consecutive, equally spaced samples of the input. Because it makes our transform algorithm simpler and much faster, N is restricted to be a multiple of 2, for instance 1024. As shown in Figure 3.3, this time record is transformed as a complete block into a complete block of frequency lines. All the samples of the time record are needed to compute each and every line in the frequency domain. This is in contrast to what one might expect, namely that a single time domain sample transforms to exactly one frequency domain line. With a dynamic signal analyzer we do not get a valid result until a full time record has been gathered (samples). Another property of the FFT is that it transforms these time domain samples to N/2 equally spaced lines in the frequency domain. We only get half as many lines because each frequency line actually contains two pieces of information, amplitude and phase. We have not discussed the phase information

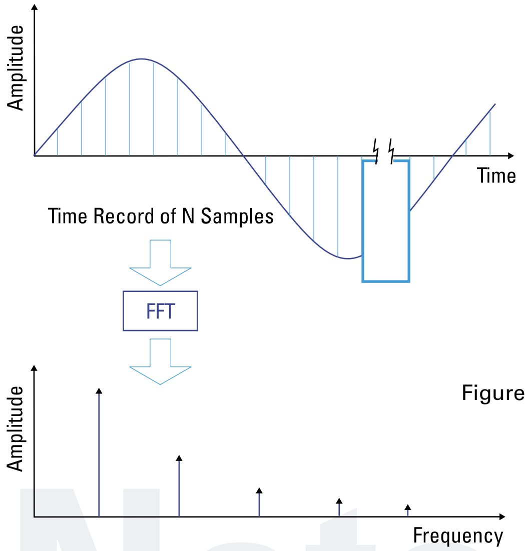


Figure 3.3

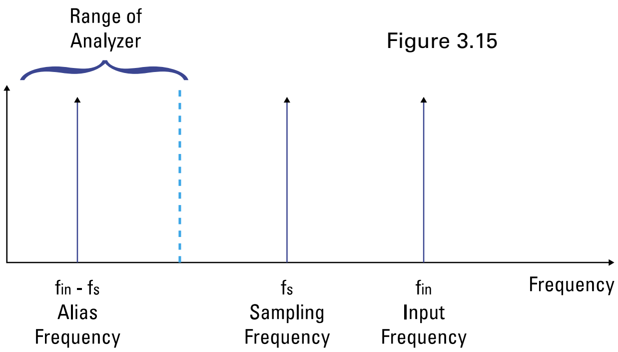


Figure 3.15

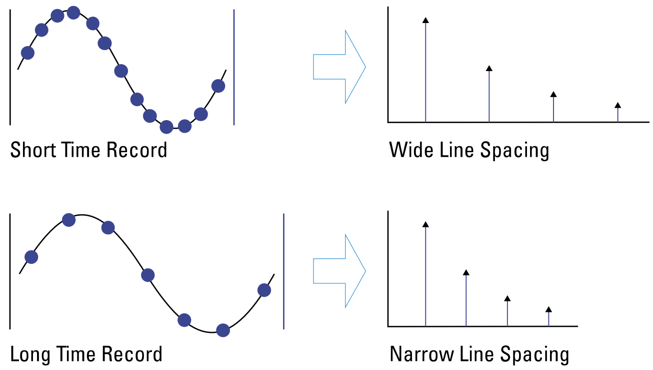


Figure 3.9

contained in the spectrum of signals until now because none of the traditional spectrum analyzers are capable of measuring phase. When we discuss measurements later, we shall find that phase contains valuable information in determining the cause of performance problems.

What is the Spacing of the Lines?

Now that we know that we have N/2 equally spaced lines in the frequency domain, what is their spacing? The lowest frequency that we can resolve with our FFT spectrum analyzer must be based on the length of the time record. We can see that if the period of the input signal is longer than the time record, we have no way of determining the period (or frequency, its reciprocal). Therefore, the lowest frequency line of the FFT must occur at frequency equal to the reciprocal of the time record length. In addition, there is a frequency line at zero Hertz, dc. This is merely the average of the input over the time record. It is rarely used in spectrum or network analysis. But, we have now established the spacing between these two lines and hence every line; it is the reciprocal of the time record.

What is the Frequency Range of the FFT?

We can now quickly determine that the highest frequency we can measure is:

fmax = N/2 * 1/ Period of Time Record

because we have N/2 lines spaced by the reciprocal of the time record starting at zero Hertz. Since we would like to adjust the frequency range of our measurement, we must vary fmax. The number of time samples N is fixed by the implementation of the FFT algorithm. Therefore, we must vary the period of the time record to vary fmax. To do this, we must vary the sample rate so that we always have N samples in our variable time record period. This is illustrated in Figure 3.9. Notice that to cover higher frequencies, we must sample faster.

Aliasing

The reason an FFT spectrum analyzer needs so many samples per second is to avoid a problem called aliasing.

Aliasing is a potential problem in any sampled data system. It is often overlooked, sometimes with disastrous results.

Aliasing is shown in the frequency domain in Figure 3.15. Two signals are said to alias if the difference of their frequencies falls in the frequency range of interest. This difference in frequency is always generated in the process of sampling. In Figure 3.15, the input frequency is slightly higher than the sampling frequency so a low frequency alias term is generated. If the input frequency equals the sampling frequency then the alias term falls at dc (zero Hertz) and we get the constant output.

Figure 3.16 shows that if we sample at greater than twice the highest frequency of our input, the alias products will not fall within the frequency range of our input. Therefore, a filter (or our FFT processor which acts like a filter) after the sampler will remove the alias products while passing the desired input signals if the sample rate is greater than twice the highest frequency of the input. If the sample rate is lower, the alias products will fall in the frequency range of the input and no amount of filtering will be able to remove them from the signal. This minimum sample rate requirement is known as the Nyquist Criterion. It is easy to see in the time domain that a sampling frequency exactly twice the input frequency would not always be enough. It is less obvious that slightly more than two samples in each period is sufficient information. It certainly would not be enough to give a high quality time display. Yet we saw in Figure 3.16 that meeting the Nyquist Criterion of a sample rate greater than twice the maximum input frequency is sufficient to avoid aliasing and preserve all the information in the input signal.

The Need for an Anti-Alias Filter

Unfortunately, the real world rarely restricts the frequency range of its signals. In the case of the room temperature, we can be reasonably sure of the maximum rate at which the temperature could change, but we still cannot rule out stray signals. Signals induced at the powerline frequency or even local radio stations could alias into the desired frequency range. The only way to be really certain that the input frequency range is limited is to add a low pass filter before the sampler and ADC. Such a filter is called an anti-alias filter.
An ideal anti-alias filter would look like Figure 3.18a. It

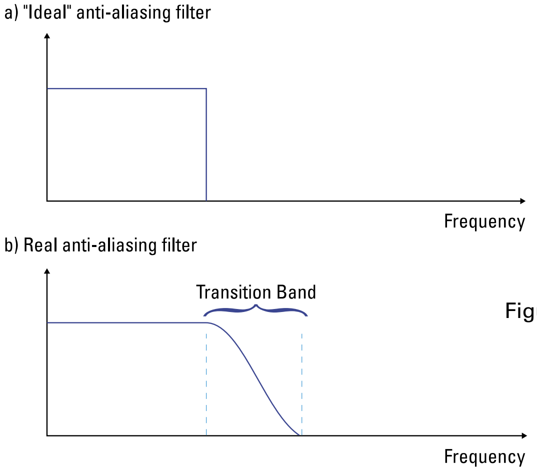


Figure 3.18

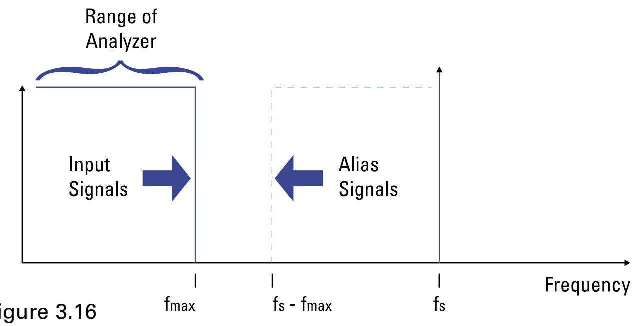


Figure 3.16

Fundamentals of Dynamic Signal Analysis

would pass all the desired input frequencies with no loss and completely reject any higher frequencies which otherwise could alias into the input frequency range. However, it is not even theoretically possible to build such a filter, much less practical. Instead, all real filters look something like Figure 3.18b with a gradual roll off and finite rejection of undesired signals. Large input signals which are not well attenuated in the transition band could still alias into the desired input frequency range. To avoid this, the sampling frequency is raised to twice the highest frequency of the transition band. This guarantees that any signals which could alias are well attenuated by the stop band of the filter. Typically, this means that the sample rate is now two and a half to four times the maximum desired input frequency.

Digital Anti-Alias Filters

Due to the properties of the FFT we must vary the sample rate to vary the frequency span of our analyzer. To reduce the frequency span, we must reduce the sample rate. From our considerations of aliasing, we now realize that we must also reduce the anti-alias filter frequency by the same amount. Typical instruments have a minimum span of 1 Hz and a maximum of tens to hundreds of kilohertz. This four decade range typically needs to be covered with at least three spans per decade. This would mean at least twelve anti-alias filters would be required for each channel. Additionally, in a multi-channel analyzer, each filter must be well matched for accurate network analysis measurements. Fortunately, there is an alternative which is cheaper and when used in conjunction with a single analog anti-alias filter, always provides aliasing protection. It is called digital filtering because it filters the input signal after we have sampled and digitized it. To see how this works, let us look at Figure 3.19. In the analog case we already discussed, we had to use a new filter every time we changed the sample rate of the Analog to Digital Converter (ADC). When using digital filtering, the ADC sample rate is left constant at the rate needed for the highest frequency span of the analyzer. This means we need not change our anti-alias filter. To get the reduced sample rate and filtering we need for the narrower frequency spans, we follow the ADC with a digital filter. This digital filter is known as a decimating filter. It not only filters the digital representation of the signal to the desired frequency span, it also reduces the sample rate at its output to the rate needed for that frequency span. Because this filter is digital, there are no manufacturing variations, aging or drift in the filter. Therefore, in a multi-channel analyzer the filters in each channel are identical. It is easy to design a single digital filter to work on many frequency spans so the need for multiple filters per channel is avoided. All these factors taken together mean that digital filtering is much less expensive and more accurate than analog anti-aliasing filtering.

Band Selectable Analysis

The real power of digital filters in a DSA allows you to increase the frequency resolution of your measurements. With a “zoom” measurement you can select a center frequency and a frequency span and rather than your FFT, resolution being based dc to the maximum frequency divided by the number of lines of your FFT it is based on your zoom frequency span. This can increase your frequency resolution by a factor of 10 or 100 easily. Digitizers with DSP offer on-board synchronized DAC’s for the generation of arbitrary waveforms. These signals can also be band translated to provide “mooz” signals that allow non-zero start

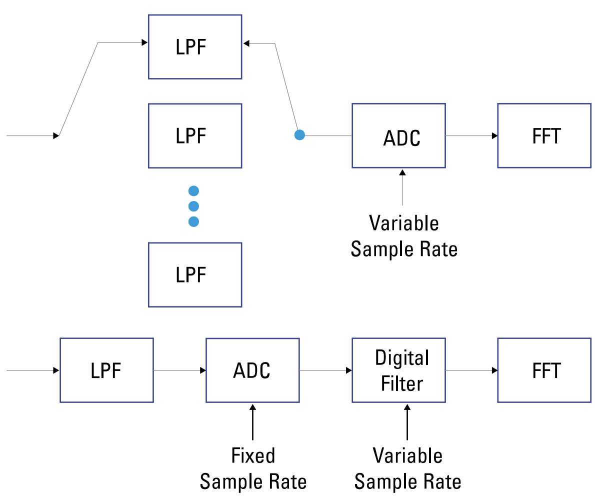


Figure 3.19

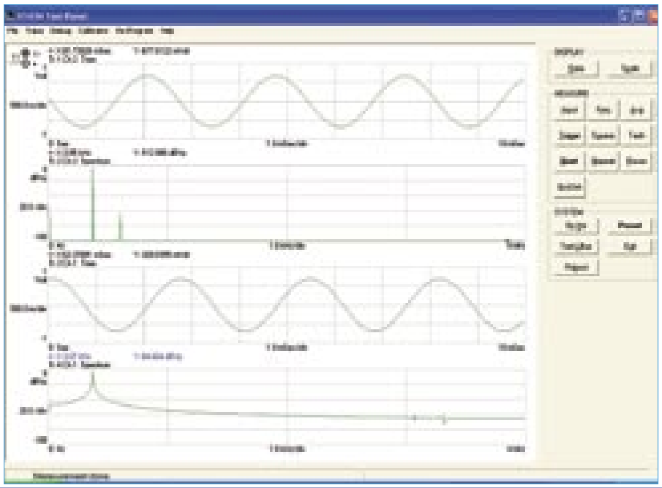


Figure 3.25

frequencies. This band-translate capability allows the source to have a bandwidth and center frequency that matches the zoom FFT, creating much higher quality measurements and avoiding troublesome resonances that might damage the DUT.

Windowing

There is another property of the fast fourier transform which affects its use in frequency domain analysis. We recall that the FFT computes the frequency spectrum from a block of samples of the input called a time record. In addition, the FFT algorithm is based upon the assumption that this time record is repeated throughout time. This does not cause a problem with the transient. But what happens if we are measuring a continuous signal like a sine wave? If the time record contains an integral number of cycles of the input sine wave, the input waveform is said to be periodic in the time record. Figure 3.24 demonstrates the difficulty with this assumption when the input is not periodic in the time record. The FFT algorithm is computed on the basis of the highly distorted waveform in Figure 3.24c. We know that the actual sine wave input has a frequency spectrum of single line. The spectrum of the input assumed by the FFT in Figure 3.24c should be very different. Since sharp phenomena in one domain are spread out in the other domain, we would expect the spectrum of our sine wave to be spread out through the frequency domain.

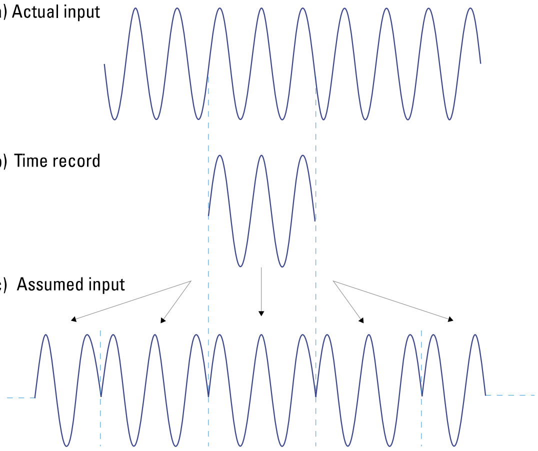


Figure 3.24

In Figure 3.25 we see in an actual measurement that our expectations are correct. In Figures 3.25 a & b, we see a sine wave that is periodic in the time record. Its frequency spectrum is a single line whose width is determined only by the resolution of our dynamic signal analyzer. On the other hand, Figures 3.25c & d show a sine wave that is not periodic in the time record. Its power has been spread throughout the spectrum as we predicted. This smearing of energy throughout the frequency domains is a phenomena known as leakage. We are seeing energy leak out of one resolution line of the FFT into all the other lines. It is important to realize that leakage is due to the fact that we have taken a finite time record. For a sine wave to have a single line spectrum, it must exist for all time, from minus infinity to plus infinity. If we were to have an infinite time record, the FFT would compute the correct single line spectrum exactly. However, since we are not willing to wait forever to measure its spectrum, we only look at a finite time record of the sine wave. This can cause leakage if the continuous input is not periodic in the time record. It is obvious from Figure 3.25 that the problem of leakage is severe enough to entirely mask small signals close to our sine waves. As such, the FFT would not be a very useful spectrum analyzer. The solution to this problem is known as windowing.

What is Windowing?

In Figure 3.24 we have again reproduced the assumed input wave form of a sine wave that is not periodic in the time record. Notice that most of the problem seems to be at the edges of the time record, the center is a good sine wave. If the FFT could be made to ignore the ends and concentrate on the middle of the time record, we would expect to get much closer to the correct single line spectrum in the frequency domain. If we multiply our time record by a function that is zero at the ends of the time record and large in the middle, we would concentrate the FFT on the middle of the time record. One such function is shown in Figure 3.29. Such functions are called window functions because they force us to look at data through a narrow window.

Figure 3.27 shows us the vast improvement we get

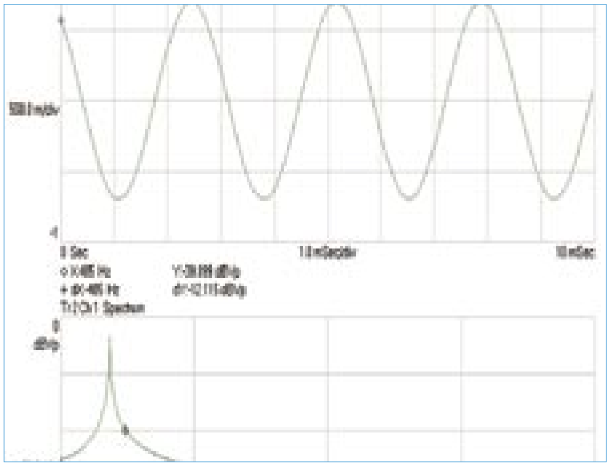


Figure 3.27

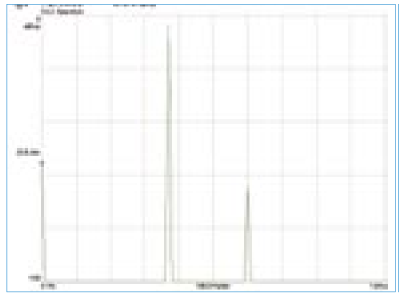


Figure 3.28a

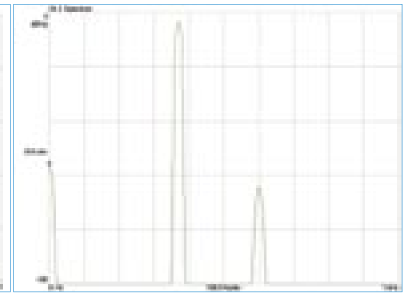


Figure 3.28b

by windowing data that is not periodic in the time record. However, it is important to realize that we have tampered with the input data and cannot expect perfect results. The FFT assumes the input looks something like an amplitude-modulated sine wave. This has a frequency spectrum which is closer to the correct single line of the input sine wave than the original, but it still is not correct. Figure 3.28 demonstrates that the windowed data does not have as narrow a spectrum as an unwindowed function which is periodic in the time record.

The Hanning Window

Any number of functions can be used to window the data, but the most common one is called Hanning. We actually used the Hanning window in Figure 3.27 as our example of leakage reduction with windowing. The Hanning window is also commonly used when measuring random noise.

The Uniform Window

We have seen that the Hanning window does an acceptably good job on our sine wave examples, both periodic and non-periodic in the time record. If this is true, why should we want any other windows? Suppose that instead of wanting the frequency spectrum of a continuous signal, we would like the spectrum of a transient event.

A typical transient is shown in Figure 3.29a. If we multiplied it by the window function in Figure 3.29b we would get the highly distorted signal shown in Figure 3.29c. The frequency spectrum of an actual transient with and without the Hanning window is shown in Figure 3.30. The Hanning window has taken our transient, which naturally has energy spread widely through the frequency domain and made it look more like a sine wave. Therefore, we can see that for transients we do not want to use the Hanning window. We would like to use all the data in the time record equally or uniformly. Hence, we will use the Uniform window which weights all of the time record uniformly. The case we made for the Uniform window by looking at transients can be generalized. Notice that our transient has the property that it is zero at the beginning and end of the time record. Remember that we introduced windowing to force the input to be zero at the ends of the time record. In this case, there is no need for windowing the input. Any function like this which does not require a window because it occurs completely within the time record is called a self-windowing function. Self-windowing functions generate no leakage in the FFT and so need no window. Impacts, impulses, shock responses, sine bursts, noise bursts, chirp bursts and pseudo-random noise can all be made to be self-windowing. Self-windowing functions are often used as the excitation in measuring the frequency response of networks, particularly if the network has lightly-damped resonances (high Q). This is because the self-windowing functions generate no leakage in the FFT. Recall that even with the Hanning window, some leakage was present when the signal was not periodic in the time record. This means that without a self-windowing excitation, energy could leak from a lightly damped resonance into adjacent lines (filters). The resulting spectrum would show greater damping than actually exists.

The Flattop Window

We have shown that we need a uniform window for analyzing self-windowing functions like transients. In addition, we need a Hanning window for measuring noise and periodic signals like sine waves. We now need to introduce a third window function, the flattop

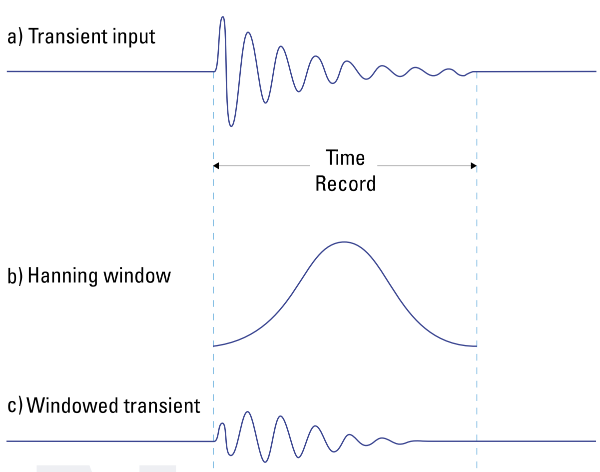


Figure 3.29

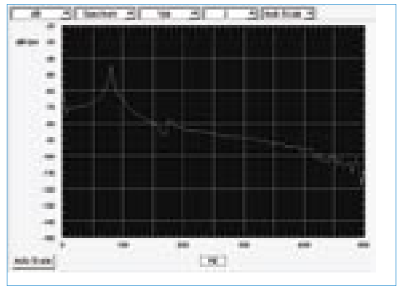


Figure 3.30a

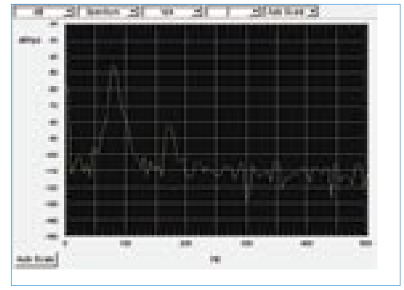


Figure 3.30b

window, to avoid a subtle effect of the Hanning window. To understand this effect, we need to look at the Hanning window in the frequency domain. We recall that the FFT acts like a set of parallel filters. Figure 3.32 shows the shape of those filters when the Hanning window is used. Notice that the Hanning function gives the filter a very rounded top. If a component of the input signal is centered in the filter it will be measured accurately. Otherwise, the filter shape will attenuate the component by up to 1.5 dB (16%) when it falls midway between the filters. This error is unacceptably large if we are trying to measure a signal's amplitude accurately. The solution is to choose a window function which gives the filter a flatter pass-band. Such a flattop pass-band shape is shown in Figure 3.33. The amplitude error from this window function does not exceed 0.1 dB (1%), a 1.4 dB improvement. The accuracy improvement does not come without its price, however. We have flattened the top of the pass band at the expense of widening the skirts of the filter. We therefore lose some ability to resolve a small component, closely spaced to a large one. Dynamic signal analyzers offer both Hanning and flattop window functions so that the operator can choose between increased accuracy or improved frequency resolution. Digitizers with DSP offer these

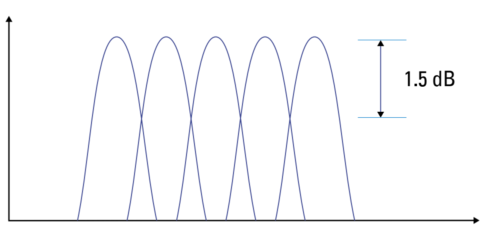


Figure 3.32

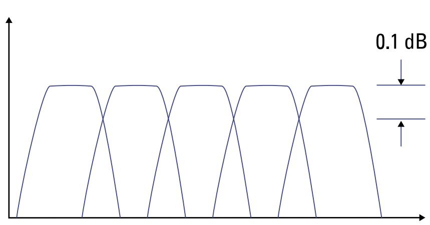


Figure 3.33

same functions with the additional benefit of increased performance for multiple channel systems by processing many channels in parallel.

Other Window Functions

Many other window functions are possible but the three listed above are by far the most common for general measurements. For special measurement situations other groups of window functions may be useful. Two windows which are particularly useful when doing network analysis on mechanical structures by impact testing are the force and response windows.

Network Stimulus

We can measure the frequency response at one frequency by stimulating the network with a single sine wave and measuring the gain and phase shift at that frequency. The frequency of the stimulus is then changed and the measurement repeated until all desired frequencies have been measured. Every time the frequency is changed, the network response must settle to its steady-state value before a new measurement can be taken, making this measurement process a slow task. Many network analyzers operate in this manner and we can make the measurement this way with a two channel dynamic signal analyzer. There are also several techniques available to improve measurement speed with a DSA.

Noise as a Stimulus

A single sine wave stimulus does not take advantage of the possible speed provided by the parallel filters of a Digitizer with DSP or dynamic signal analyzer provide. If we had a source that put out multiple sine waves, each one centered in a filter, then we could measure the frequency response at all frequencies at one time. Such a source acts like hundreds of sine wave generators connected together. This type of source is called a pseudorandom noise or periodic random noise source. From the names used for this source it is apparent that it acts somewhat like a true noise generator, except that it has periodicity. If we add together a large number of sine waves, the result is very much like white noise. However if we add a large number of sine waves, our noise-like signal will periodically repeat its sequence. Hence, the name periodic random noise (PRN) source.

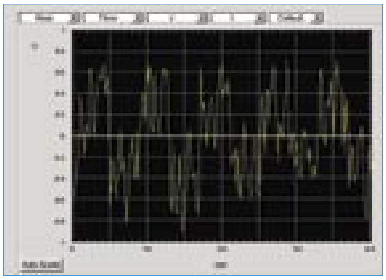


Figure 3.44 Single time record, no averaging

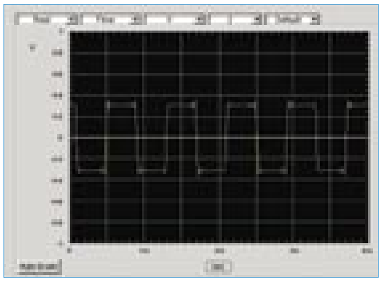


Figure 3.44b Time record 122 linear averages

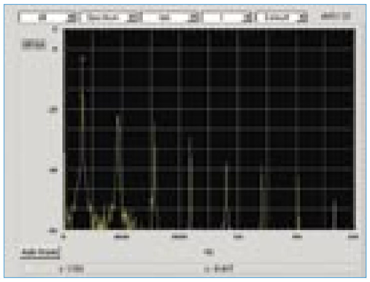


Figure 3.44c Frequency spectrum 122 linear averages

Fundamentals of Dynamic Signal Analysis

A truly random noise source would not be periodic. It is apparent that a random noise source would also stimulate all the filters at one time and so could be used as a network stimulus. Which is a better stimulus? The answer depends upon the measurement situation.

Linear Network Analysis

If the network is reasonably linear, PRN and random noise both give the same results as the swept-sine test of other analyzers. But, PRN gives the frequency response much faster. PRN, can be used to measure the frequency response in a single time record. Because the random source is true noise, it must be averaged for several time records before an accurate frequency response can be determined. Therefore, PRN is the best stimulus to use with fairly linear networks because it gives the fastest results.

Nonlinear Network Analysis

If the network is severely nonlinear, the situation is quite different. In this case, PRN is a very poor test signal and random noise is much better. This is because the sine waves that compose the PRN source are put through a nonlinear network, distortion products will be generated equally spaced from the signals. Unfortunately, these products will fall exactly on the frequencies of the other sine waves in the PRN, so the distortion products add to the output and therefore interfere with the measurement of the frequency response. This shows as a jagged response trace of a nonlinear network measured with PRN. Because the PRN source repeats itself exactly every time record, this noisy looking trace never changes and will not average to the desired frequency response. With random noise, the distortion components are also random and will average out. Therefore, the frequency response does not include the distortion and we get the more reasonable results.

This points out a fundamental problem with measuring nonlinear networks; the frequency response is not a property of the network alone, it also depends on the stimulus. Each stimulus, swept-sine, PRN and random noise will, in general, give a different result. Also, if the amplitude of the stimulus is changed, you will get a different result.

Averaging

The standard technique in statistics to improve the estimates of a value is to average. When we watch a noisy reading on a dynamic signal analyzer, we can guess the average value. But because the dynamic signal analyzer contains digital computation capability we can have it compute this average value for us. Two kinds of averaging are available, rms (or "power" averaging) and linear averaging.

RMS Averaging

rms stands for "root-mean-square" and is calculated by squaring all the values, adding the squares together, dividing by the number of measurements (mean) and taking the square root of the result. If we want to measure a small signal in the presence of noise, rms averaging will give us a good estimate of the signal plus noise. We cannot improve the signal to noise ratio with rms averaging; we can only make more accurate estimates of the total signal plus noise power.

Linear Averaging

However, there is a technique for improving the signal to noise ratio of a measurement, called linear averaging. It can be used if a trigger signal which is synchronous with the periodic part of the spectrum is available. Of course, the need for a synchronizing signal is somewhat restrictive, although there are numerous situations in which one is available. In network analysis problems the stimulus signal itself can often be used as a synchronizing signal. Figure 3.44 is an example of linear averaging.

Real Time Bandwidth

Until now we have ignored the fact that it will take a finite time to compute the FFT of our time record. In fact, if we could compute the transform in less time than our sampling period we could continue to ignore this computational time. If we limit our measurements to one or two channels and low frequency spans this is possible. A reasonable alternative is to add a time record buffer to the block diagram of our analyzer. This allows us to compute the frequency spectrum of the previous time record while gathering the current time record. If we can compute the transform before the time record buffer fills, then we are said to be operating

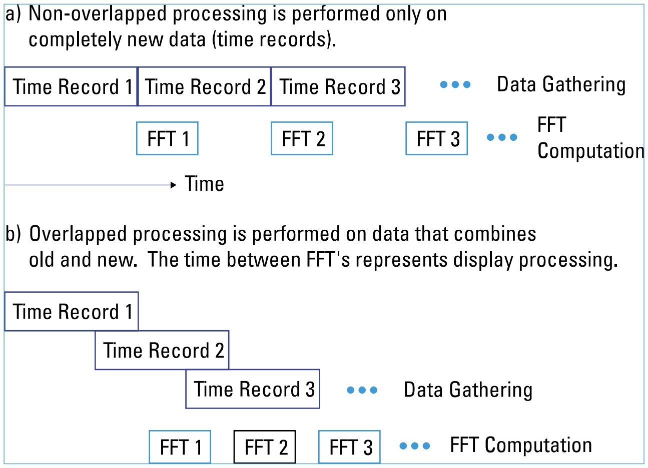


Figure 3.51

in real time. To see what this means, let us look at the case where the FFT computation takes longer than the time to fill the time record. Although the buffer is full, we have not finished the last transform, so we will have to stop taking data. When the transform is finished, we can transfer the time record to the FFT and begin to take another time record. This means that we missed some input data and so we are said to be not operating in real time. Recall that the time record is not constant but deliberately varied to change the frequency span of the analyzer. For wide frequency spans, the time record is shorter. Therefore, as we increase the frequency span of the analyzer, we eventually reach a span where the time record is equal to the FFT computation time. This frequency span is called the real time bandwidth. For frequency spans at and below the real time bandwidth, the analyzer does not miss any data.

Overlap Processing

To understand overlap processing, let us look at Figure 3.51a. We see a low frequency analysis where the gathering of a time record takes much longer than the FFT computation time. Our FFT processor is sitting idle much of the time. If instead of waiting for an entirely new time record we overlapped the new time record with some of the old data, we would get a new spectrum as often as we computed the FFT. This overlap processing is illustrated in Figure 3.51b. Overlap processing can give dramatic reductions in the time to compute rms averages with a given variance. Recall that window functions reduce the effects of leakage by weighting the ends of the time record to zero. Overlapping eliminates most or all of the time that would be wasted taking this data. Because some overlapped data is used twice, more averages must be taken to get a given variance than in the non-overlapped case.

Using Dynamic Signal Analyzers

In this section we show how to use dynamic signal analyzers in a wide variety of measurement situations. We introduce the measurement functions of dynamic signal analyzers as we need them for each measurement situation. We begin with some common electronic and mechanical measurements in the frequency domain.

Frequency Domain Measurements

Oscillator Characterization

Let us begin by measuring the characteristics of an electronic oscillator. An important specification of an oscillator is its harmonic distortion. In Figure 4.1, we show the fundamental through fifth harmonic of a 1 kHz oscillator. Because the frequency is not necessarily exactly 1 kHz, windowing should be used to reduce the leakage. We have chosen the flattop window so that we can accurately measure the amplitudes. Notice that we have selected the input sensitivity of the analyzer so that the fundamental is near the top of the display. In general, we set the input sensitivity to the most sensitive range which does not overload the analyzer. Severe distortion of the input signal will occur if its peak voltage exceeds the range of the analog to digital converter. Therefore, all dynamic signal analyzers warn the user of this condition by some kind of overload indicator. It is also important to make sure the analyzer is not under loaded. The signal going into the analog to digital converter is too small, much of the useful information of the spectrum may be below the noise level of the analyzer. Therefore, setting the input sensitivity to the most sensitive range that does not cause an overload gives the best possible results. In Figure 4.1a, we chose to display the spectrum amplitude in logarithmic form to insure that we could see distortion products far below the fundamental. All signal amplitudes on this display are in dBV, decibels below 1 V rms. However, since most dynamic signal analyzers have very versatile display capabilities, we could also display this spectrum linearly as in Figure 4.1b. Here, the units of amplitude are volts.

Power-line Sidebands

Another important measure of an oscillator's performance is the level of its power-line sidebands. In Figure 4.1c, we use band selectable analysis to "zoom in" on the signal so that we can easily resolve and measure the sidebands which are only 60 Hz away from our 1 kHz signal. With some analyzers it is possible to measure signals only millihertz away from the fundamental if desired.

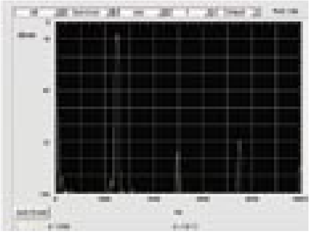
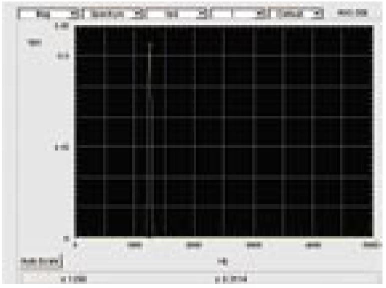


Figure 4.1a

Figure 4.1b



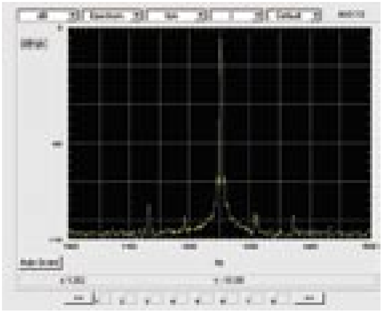


Figure 4.1c

Fundamentals of Dynamic Signal Analysis

Rotating Machinery Characterization

A rotating machine can be thought of as a mechanical oscillator. Therefore, many of the measurements we made for an electronic oscillator are also important in characterizing rotating machinery. To characterize a rotating machine we must first change its mechanical vibration into an electrical signal. This is often done by mounting an accelerometer on a bearing housing where the vibration generated by shaft imbalance and bearing imperfections will be the highest.

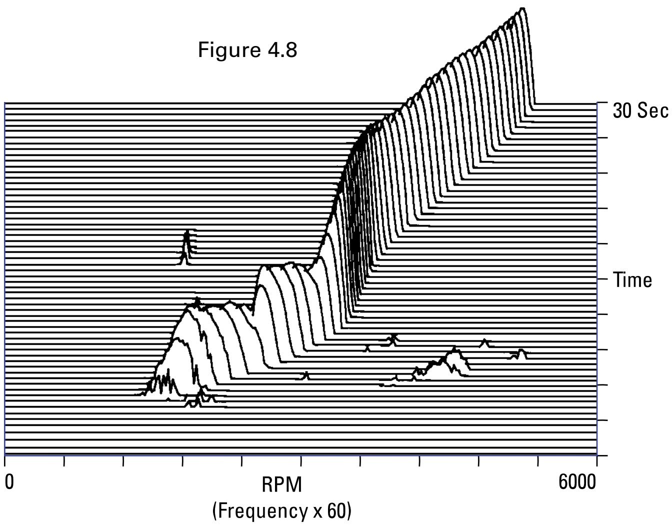
A typical spectrum might look like Figure 4.1. It would obviously be much more complicated than the relatively clean spectrum of the electronic oscillator we looked at previously. There is also a great deal of random noise; stray vibrations from sources other than our motor that the accelerometer picks up. The band selectable analysis capability of our analyzer to "zoom-in" and separate the vibration of the stator at 120 Hz from the vibration caused by the rotor imbalance only a few tenths of a hertz lower in frequency. This ability to resolve closely spaced spectrum lines is crucial to our capability to diagnose why the vibration levels of a rotating machine are excessive. The actions we would take to correct an excessive vibration at 120 Hz are quite different if it is caused by a loose stator pole rather than an imbalanced rotor.

Since the bearings are the most unreliable part of most rotating machines, we would also like to check our spectrum for indications of bearing failure. Any defect in a bearing, say a spalling on the outer face of a ball bearing, will cause a small vibration to occur each time a ball passes it. This will produce a characteristic frequency in the vibration called the passing frequency. The frequency domain is ideal for separating this small vibration from all the other frequencies present. This means that we can detect impending bearing failures and schedule a shutdown long before they become the loudly squealing problem that signals an immediate shutdown is necessary. In most rotating machinery monitoring situations, the absolute level of each vibration component is not of interest, just how they change with time. The machine is measured when new and throughout its life and these successive spectra are compared. If no catastrophic failures develop, the spectrum components will increase gradually as the machine wears out. However, if an impending bearing failure develops, the passing frequency component corresponding to the defect will increase suddenly and dramatically.

An excellent way to store and compare these spectra is by using a computer based monitoring system. The spectra can be easily compared with previous results by a trend analysis program. This avoids the tedious and error prone task of generating trend graphs by hand. In addition, the computer can easily check the trends against limits, pointing out where vibration limits are exceeded or where the trend is for the limit to be exceeded in the near future.

Desktop computers are also useful when analyzing machinery that normally operates over a wide range of speeds. Severe vibration modes can be excited when the machine runs at critical speeds. A quick way to determine if these vibrations are a problem is to take a succession of spectra as the machine runs up to speed or coasts down. Each spectrum shows the vibration components of the machine as it passes through an rpm range. RPM is measured by the tachometer input of the digitizer. Tachometer inputs have signal conditioning designed to measure inputs from a proximity probe or optical tachometer. If each spectrum is transferred to the computer the results can be processed and displayed as

Figure 4.8



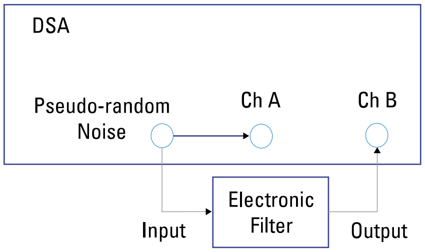


Figure 4.9

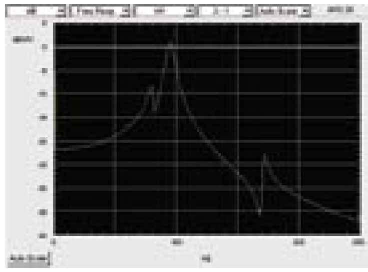


Figure 4.10a

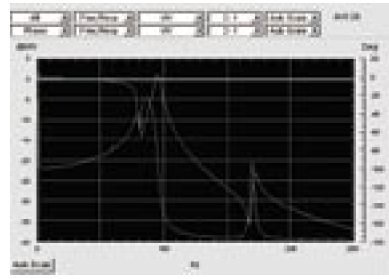


Figure 4.10b

in Figure 4.8. From such a display it is easy to see shaft imbalances, constant frequency vibrations (from sources other than the variable speed shaft) and structural vibrations excited by the rotating shaft. The computer gives the capability of changing the display presentation to other forms for greater clarity. Because all the values of the spectra are stored in memory, precise values of the vibration components can easily be determined. In addition, signal processing can be used to clarify the display. For instance, in Figure 4.8 all signals below -70 dB were ignored. This eliminates meaningless noise from the plot, clarifying the presentation.

Electronic Filter Characterization

In previous sections we developed most of the principles we need to characterize a low frequency electronic filter. We show the test setup we might use in Figure 4.9. Because the filter is linear we can use pseudo-random noise as the stimulus for very fast test times. The uniform window is used because the pseudo-random noise is periodic in the time record. No averaging is needed since the signal is periodic and reasonably large. We should be careful, as in the single channel case, to set the input sensitivity for both channels to the most sensitive position which does not overload the analog to digital converters. With these considerations in mind, we get a frequency response magnitude shown in Figure 4.10a and the phase shown in Figure 4.10b. The primary advantage of this measurement over traditional swept analysis techniques is speed. This measurement can be made in 1/8 second with a dynamic signal analyzer, but would take over 30 seconds with a swept network analyzer. This speed improvement is particularly important when the filter under test is being adjusted or when large volumes are tested on a production line.

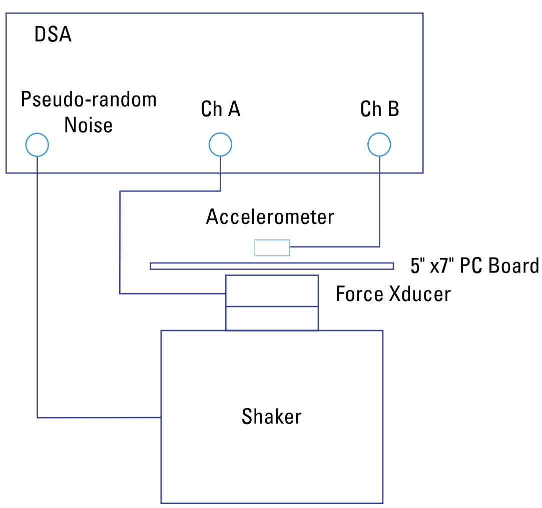


Figure 4.11

Structural Frequency Response

The network under test does not have to be electronic. In Figure 4.11, we are measuring the frequency response of a single structure, in this case a printed circuit board. Because this structure behaves in a linear fashion, we can use pseudo-random noise as a test stimulus. But we might also desire to use true random noise, swept-sine or an impulse (hammer blow) as the stimulus.

In Figure 4.12 we show each of these measurements and the frequency responses. As we can see, the results are all the same.

The frequency response of a linear network is a property solely of the network, independent of the stimulus used.

Since all the stimulus techniques in Figure 4.12 give the same results, we can use whichever one is fastest and easiest. Usually this is the impact stimulus, since a shaker is not required. In Figure 4.11 and 4.12, we have been measuring the acceleration of the structure divided by the force applied. This quality is called mechanical accelerance. To properly scale the displays to the required g's/lb, we have entered the sensitivities of each transducer into the analyzer by a feature called engineering units. Engineering units simply changes the gain of each channel of the analyzer so that the display corresponds to the physical parameter that the transducer is measuring. Other frequency response measurements besides mechanical accelerance are often made on mechanical structures. By changing transducers we could measure any of

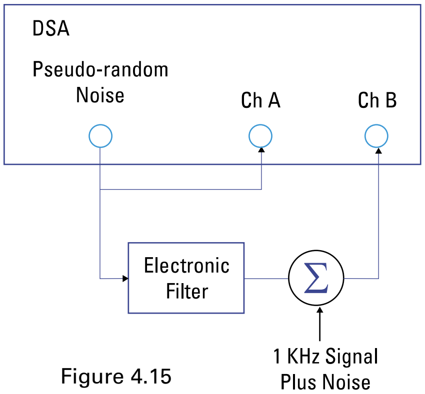


Figure 4.15

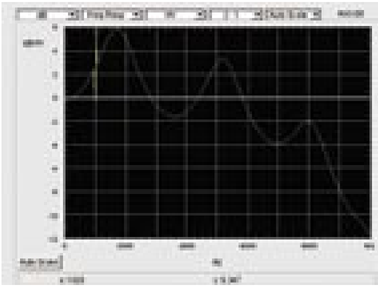


Figure 4.16

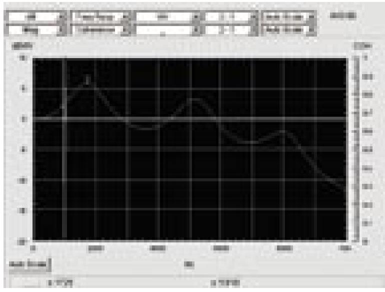
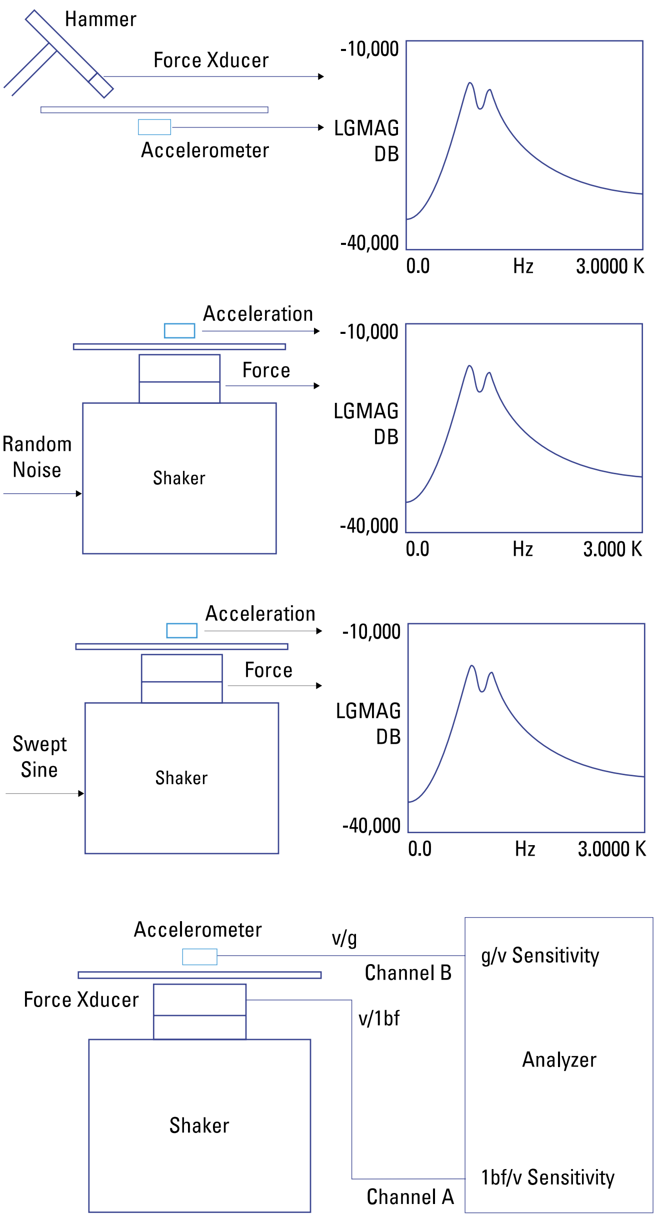


Figure 4.17

Figure 4.12



these parameters. Or we can use the computational capability of the dynamic signal analyzer to compute these measurements from the mechanical impedance measurement we have already made.

Coherence

Up to this point, we have been measuring networks which we have been able to isolate from the rest of the world. That is, the only stimulus to the network is what we apply and the only response is that caused by this controlled stimulus. This situation is often encountered in testing components, e.g., electric filters or parts of a mechanical structure. However, there are times when the components we wish to test cannot be isolated from other disturbances. For instance, in electronics we might be trying to measure the frequency response of a switching power supply which has a very large component at the switching frequency. Or we might try to measure the frequency response of part of a machine while other machines are creating severe vibration. In Figure 4.15 we have simulated these situations by adding noise and a 1 kHz signal to the output of an electronic filter. The measured frequency response is shown in Figure 4.16. rms averaging has reduced the noise contribution, but has not completely eliminated the 1 kHz interference. If we did not know of the interference, we would think that this filter has an additional resonance at 1 kHz.

Dynamic signal analyzers can often make an additional measurement that is not available with traditional network analyzers called coherence. Coherence measures the power in the response channel that is caused by the power in the reference channel. It is the output power that is coherent with the input power. Figure 4.17 shows the same frequency response magnitude from Figure 4.16 and its coherence. The coherence goes from 1 (all the output power at that frequency is caused by the input) to 0 (none of the output power at that frequency is caused by the input). We can easily see from the coherence function that the response at 1 kHz is not caused by the input but by interference. However, our filter response near 500 Hz has excellent coherence and so the measurement here is good.

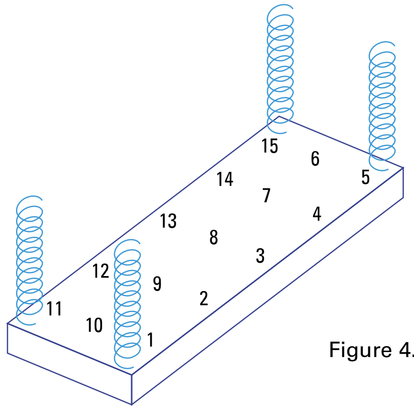


Figure 4.27

Modal Domain Measurements

Earlier we learned how to make frequency domain measurements of mechanical structures with dynamic signal analyzers. Let us now analyze the behavior of a simple mechanical structure to understand how to make measurements in the modal domain. We will test a simple metal plate shown in Figure 4.27. The plate is freely suspended using rubber cords in order to isolate it from any object which would alter its properties. The first decision we must make in analyzing this structure is how many measurements to make and where to make them on the structure. There are no firm rules for this decision; good engineering judgment must be exercised instead.

Measuring too many points makes the calculations unnecessarily complex and time consuming. Measuring too few points can cause spatial aliasing; i.e., the measurement points are so far apart that high frequency bending modes in the structure cannot be measured accurately. To decide on a reasonable number of measurement points, take a few trial frequency response measurements of the structure to determine the highest significant resonant frequencies present. The wave length can be determined empirically by changing the distance between the stimulus and the sensor until a full 360° phase shift has occurred from the original measurement point. Measurement point spacing should be approximately one-quarter or less of this wavelength. Measurement points can be spaced uniformly over the structure using this guideline, but it may be desirable to modify this procedure slightly.

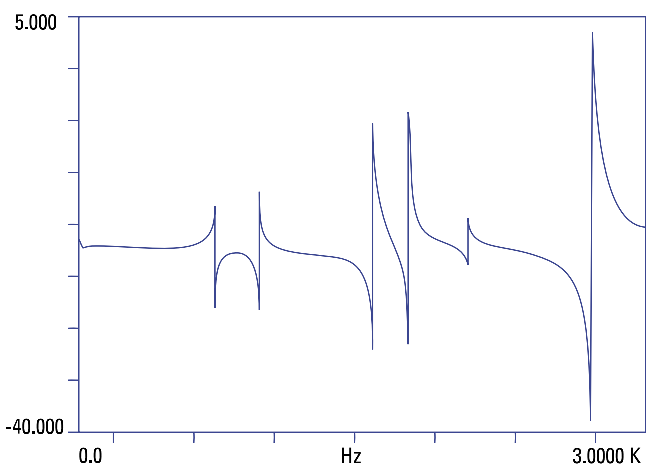


Figure 4.29

Few structures are as uniform as this simple plate example, but complicated structures are made of simpler, more uniform parts. The behavior of the structure at the junction of these parts is often of great interest, so measurements should be made in these critical areas as well. Once we have decided on where the measurements should be taken, we number these measurement points (the order can be arbitrary) and enter the coordinates of each point into our modal analyzer software. This is necessary so that the analyzer can correlate the measurements we make with a position on the structure to compute the mode shapes. The next decision we must make is what signal we should use for a stimulus. Our plate example is a linear structure as it has no loose rivet joints, nonlinear damping materials, or other nonlinearities. Therefore, we know that we can use any of the stimuli. In this case, an impulse would be a particularly good test signal. We could supply the impulse by hitting the structure with a hammer equipped with a force transducer. This is probably the easiest way to excite the structure as a shaker and its associated driver are not required. As we saw in the last chapter, however, if the structure were nonlinear, then random noise would be a good test signal.

To supply random noise to the structure we would need to use a shaker. To keep our example more general, we will use random noise as a stimulus. The shaker is connected firmly to the plate via a load cell (force transducer) and excited by the band-limited noise source of the analyzer. Since this force is the network stimulus, the load cell output is connected through a suitable amplifier to the reference channel of the analyzer. To begin the experiment, we connect an accelerometer to the plate at the same point as the load cell. The accelerometer measures the structure's response and its output is connected to the other analyzer channel. Because we are using random noise, we will use a Hanning window and rms averaging just as we did in the

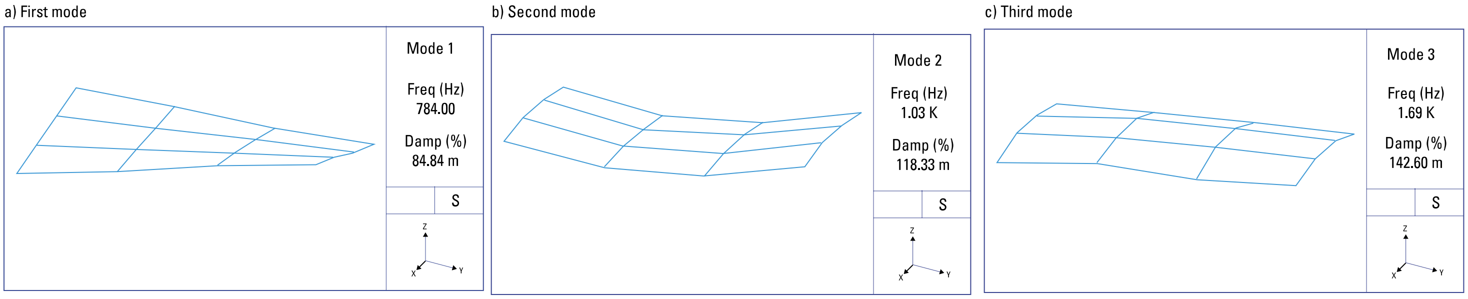


Figure 4.30

previous section. The resulting frequency response of this measurement is shown in Figure 4.29.
The ratio of acceleration to force in g's/lb is plotted on the vertical axis by the use of engineering units, and the data shows a number of distinct peaks and valleys at particular frequencies. We conclude that the plate moves more freely when subjected to energy at certain specific frequencies than it does in response to energy at other frequencies. We recall that each of the resonant peaks corresponds to a mode of vibration of the structure. Our simple plate supports a number of different modes of vibration, all of which are well separated in frequency.

Structures with widely separated modes of vibration are relatively straight forward to analyze since each mode can be treated as if it is the only one present. Tightly-spaced, but lightly-damped vibration modes can also be easily analyzed if the band selectable analysis capability is used to narrow the analyzer's filter sufficiently to resolve these resonances. Tightly spaced modes whose damping is high enough to cause the responses to overlap create computational difficulties in trying to separate the effects of the vibration modes. Fortunately, many structures fall into the first two categories and so can be easily analyzed. Having inspected the measurement and deciding that it met all the above criteria, we can store it away. We store similar measurements at each point by moving our accelerometer to each numbered point or, in the case of a multi-channel system measure all points at the same time.

Multi-channel measurements have the added advantage that the structure will have the same loading and environmental conditions at all measurement points. We will then have all the measurement data we need to fully characterize the structure in the modal domain. Recall earlier, that each frequency response will have the same number of peaks, with the same resonant frequencies and dampings. The next task is to determine these resonant frequency and damping values for each resonance of interest. We do this by retrieving our stored frequency responses and, using a curve-fitting routine, we calculate the frequency and damping of each resonance of interest. With the structural information we entered earlier, and the frequency and damping of each vibration mode which we have just determined, the modal analyzer can calculate the mode shapes by curve fitting the responses of each point with the measured resonances. In Figure 4.30 we show several mode shapes of our simple rectangular plate. These mode shapes can be animated on the display to show the relative motion of the various parts of the structure. The graphs in Figure 4.30, however, only show the maximum deflection.

Summary

This note has attempted to demonstrate the advantages of expanding one's analysis capabilities from the time domain to the frequency and modal domains. Problems that are difficult in one domain are often clarified by a change in perspective to another domain. The dynamic signal analyzer is a particularly good analysis tool at low frequencies. It cannot only work in all three domains, it is also very fast. For measurement applications that require more than a small number of channels a modular digitizer with DSP solution that is designed as a DSA is required. A digitizer designed for dynamic signal analysis should have a minimum set of capability not found in most digitizers designed for general data acquisition. These capabilities include:

- Good phase and gain match for each channel, each module and if available extended mainframes
- Analog and digital anti-alias filters that track the sampling rate of the digitizer
- Parallel data paths to insure the storage of large data sets in multi-channel systems
- Signal conditioning for popular transducers including voltage, IEPE, charge, and microphone inputs
- DSPs to provide window functions and advanced measurements like order tracking
- Parallel design that scales to large channel counts
- Integrated source to provide a range of excitation signals from swept sine to arbitrary waveforms
- Integrated tachometer input for direct readings of RPM and trigger position data
- A wide range of software applications