Research of Order Processing Capabilities for Oscillatory Signals in Rotating Equipment

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Nathan Harrison

May 2018

ACKNOWLEDGEMENTS

I would like to thank everyone who has taught, mentored, and supported me in the writing of this thesis. Thank you to Dr. Rob McAmis for believing in me and helping me get started in graduate school. I would like to give a special thank you to Steve Arnold for mentorship and teaching me aeromechanical analysis techniques, which led me to the topic of this thesis. I would also like to thank Bryan Hayes, Terry Hayes, Phil Voyles, Stephen Powell, Andrew Redmon, Andrew Jackson, and Jonathan Lister for their willingness to providing their technical expertise. Thank you to Gerald Burnett with QuantiTech and Aerospace Testing Alliance for funding my graduate school tuition. I also want to thank Harry Clark at AEDC for funding the time to research and write this thesis. Thank you to my advisor and the chair of my thesis committee, Dr. Trevor Moeller, and thesis other committee members, Dr. Milton Davis and Dr. James Simonton for providing their technical expertise and guidance.

I would also like to thank my father, Dugg Harrison, and mother, Anne Harrison, who have always believed in and supported me in countless ways throughout the years.

I would like to thank my wonderful wife, Hope Harrison, who has been a steady source of love and encouragement spanning my first undergraduate course to the completion of this master’s thesis.

Most importantly, I give thanks and praise to Christ Jesus, my Lord and Savior.

ABSTRACT

The Arnold Engineering Development Complex (AEDC) has identified a need to process data from oscillatory signals on a revolution basis, also known as order processing. Such oscillatory data is hereafter referred to as dynamic data. Order processing would serve to improve dynamic data accuracy as reported in the frequency and order domains, capture momentary integral responses, and facilitate organic comparisons between various types of oscillatory signals. Organizing data by revolutions would also be beneficial for time domain analysis.

This paper explores the need for order processing, reviews similar methods employed in other data acquisition applications such as blade tip timing, and discusses options for making order processing possible for any oscillatory signal generated by rotating equipment. This paper primarily deals with turbine engine vibratory instrumentation and data acquisition. The content discussed herein may also be extended to other types of rotating equipment such as motors, compressors, and turbines. Order processing deals primarily with integral (synchronous) responses, which are forced responses as a function of rotational speed and natural frequencies. Non-synchronous responses, also known as non-integral (NIV) responses, are not considered.

The focus of this paper lies in the research of conditioning time domain data sets of various sizes to be transformed to the frequency domain by means of FFTs with standard 2^x sizes. This is to be accomplished while varying numbers

of samples per revolution for a full range of rotational speeds. Succinctly stated, a comparison is made between standard FFT processing results and simplistic order processing methods. Since improved accuracy is one of the major drivers for developing this capability, the focal points are the acquisition, conditioning, and processing of virtual data sets that are of different sizes than specified FFT sizes. More specifically, the effects of decimation, zero padding, windowing functions, and other types of processing variables are evaluated. This research serves as a precursor for development of comprehensive order processing capabilities for turbine engines at AEDC.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/NOMENCLATURE

EO	Engine Order
f	Frequency
FFT	Fast Fourier Transform
FFTs	FFT size
f_{max}	Maximum frequency of interest
f_o	Cyclic frequency in cycles per unit time
f_1	Fundamental Frequency
f_s	Sampling Frequency
IEO	Integral Engine Order
m	Summation integer
M	Number of equally spaced points
n	Integer count
N	Rotation speed
$x(t)$	Instantaneous value at time t
$x_c(t)$	Instantaneous value at time t of complex waveform
x_m	m^{th} amplitude in Fourier series
X	Amplitude
T_p	Period
TWL	Time Waveform Length
Δt	Sampling time spacing
Δf	Bin Width or Frequency Resolution

Θ	Initial phase angle with respect to the time origin in radians

CHAPTER ONE

INTRODUCTION AND GENERAL INFORMATION

Arnold Engineering Development Complex (AEDC) is world-renowned for aerospace testing capabilities. AEDC conducts testing related to aerodynamics, missiles, space environments, hypersonics, ballistics, and turbine engines. Turbine engine testing takes place in the Engine Test Facility (ETF), also known as the aero-propulsion branch, and includes multiple simulated altitude and sea-level test facilities. Among several common genres of turbine engine testing objectives is aeromechanics. Aeromechanics is the science of the interaction between structures and fluid movement, usually in the context of oscillatory responses. These responses are referred to as aeromechanical excitations. A good example of an aeromechanical excitation, as given by Dr. Kurt Nichol [1], is a semi-trailer truck passing under a suspended highway sign. The passing truck causes an air disturbance which induces a damped vibratory response on the sign. In turbine engines, aeromechanics primarily refers to the characterization of responses on airfoils of rotors and stators, guide vanes, and other internal turbine engine components that are affected by fluid movement. Like the example given, turbine engine rotors cause aeromechanical excitations on stators, and vice-versa. This is due to the airflow disturbance created by the relative velocity delta between stators and rotors. These airflow disturbances induce vibratory stresses on turbine engine components. Figure 1 shows rotors passing stators in a typical turbine engine fan or compressor.

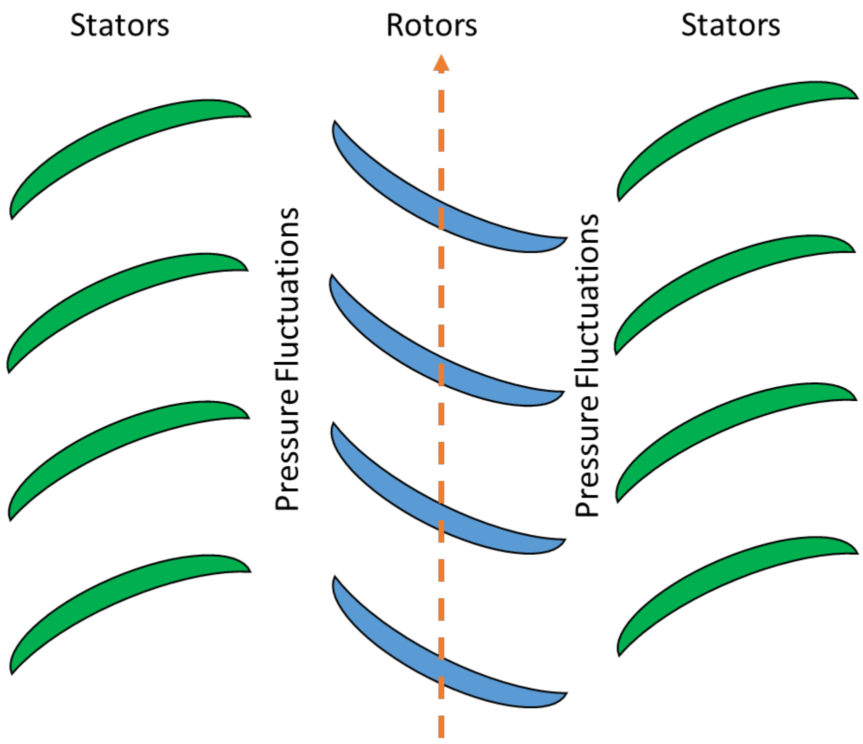

Figure 1. Rotors Passing Stators in an Axial Fan or Compressor

In order to characterize these responses, various types of instrumentation are used. AEDC employs strain gages, tip timing laser systems, and accelerometers to measure and quantify these responses on turbine engine components. These types of instrumentation produce high-response oscillatory signals, commonly referred to as dynamic data. When one views dynamic data as measured in the time domain, it is seemingly random and meaningless. This is because there are usually several discreet responses and noise occurring simultaneously.

Figure 2 shows time domain data, a complex time waveform, with realistic low amplitude noise. Though not readily apparent, there are 4 discreet responses (200 Hz, 450 Hz, 2000 Hz, and 2500 Hz) in the signal shown.

Dynamic data becomes especially useful after being transformed to the frequency domain. Once data is in the frequency domain amplitude, frequency, order, and phase can be viewed and analyzed. The transformation from the time domain to the frequency domain is accomplished by means of one of the Fourier Transforms (FT), usually the Fast Fourier Transform (FFT). The FFT takes samples from the time domain in fixed intervals and effectively applies multiple periodic curve fits, yielding frequency content. This will be covered in more detail in Chapter 2. Figure 3 shows the data from Figure 2 after being transformed to the frequency domain (processed in MATLAB). As can be observed, there are 4 discreet responses (200 Hz, 450 Hz, 2000 Hz, and 2500 Hz) that were not readily apparent by looking at the time domain alone (Figure 2).

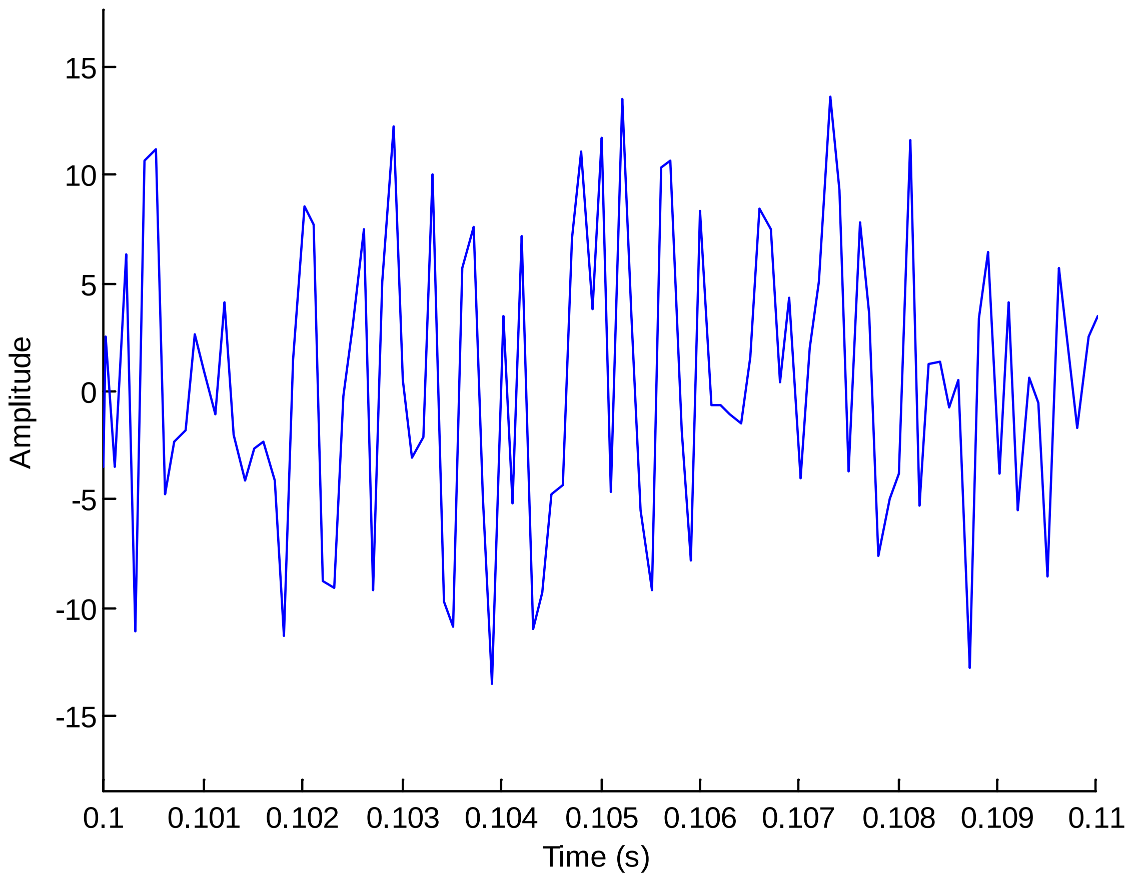

Figure 2. Example of Time Domain Data with Noise

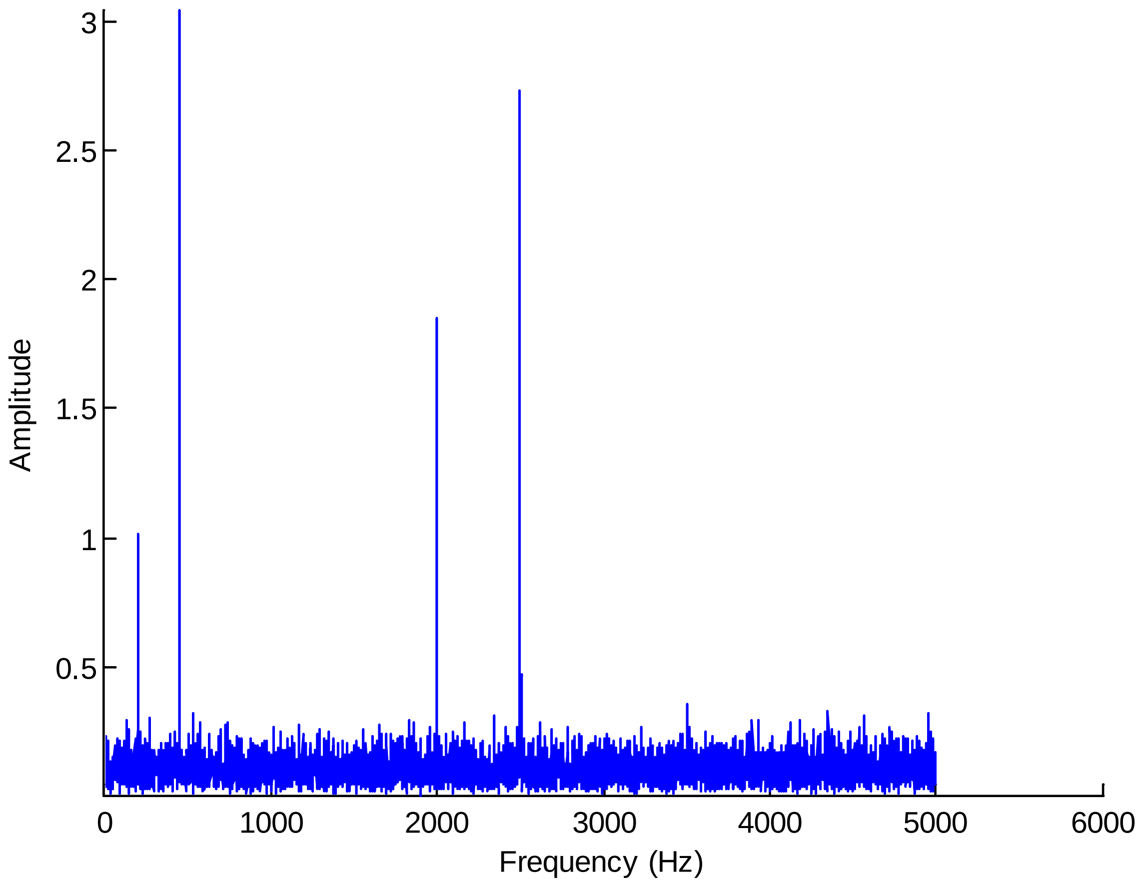

Figure 3. Example of Frequency Domain Data

In standard FFT processing, the samples used in each FFT (sometimes referred to as blocks) have arbitrary start and stop times, and have no relationship with circumferential locations of rotating components. Multiple revolutions are often included in a single FFT, which can skew the data in the frequency domain. It has been observed that momentary responses, that are sometimes not evident with standard processing methods, can cause serious damage to rotating components. For this reason and others AEDC desires the capability to process dynamic data on a single revolution basis. This would reduce averaging that can drastically decrease the amplitudes of momentary responses. This type of processing is on a once-per-revolution basis (1/REV) and is referred to as “rev processing,” order processing, or order tracking. The term used in this paper henceforth will be order processing. The objective is to include exactly 1 revolution of data in each FFT for the entire range of speeds. This would be accomplished by “stamping” the data using a 1/REV engine sensor and dividing up the data by revolutions instead of arbitrary, successive blocks. The FFT samples would then start and stop at the same approximate circumferential location (when the 1/REV pulses), yielding exactly 1 rotation of frequency domain data for each successive revolution.

There are several acquisition and processing challenges associated with order processing, especially for rotating equipment with large ranges of rotational speeds. At a fixed sample rate, a single revolution of data at a low rotational speed will contain a larger sample size than a revolution of data at a high speed.

Furthermore, sample sizes will vary proportionally with engine speed. Due to the nature of computer processors, FFTs typically employ fixed sample sizes that are 2^x. With a fixed sample rate and changing engine speeds, the number of samples included in one revolution of data (i.e. one FFT) will rarely add up to a sample size equaling exactly 2^x. The number of samples can be resolved to 2^x via sample padding, decimation (downsampling), or a combination of both. Resampling is also an option. Another option is to simply execute FFTs with sample sizes $\neq 2^x$, while accepting a processing lag as a trade-off. While this may be feasible for post-processing applications, due to the processing lag time, it may not be feasible for real-time applications (i.e. monitoring) unless expensive high-performance computers are used. Each of these factors are considered in greater detail.

Order processing can bring benefits to dynamic data analysis in both the time and frequency domains. This paper discusses the research of existing dynamic data processing concepts and methods and draws on existing synchronous processing methods such as blade tip timing algorithms. More specifically the effects of decimation, zero padding, windowing functions, and some other types of processing variables are evaluated. This research serves as a precursor for development of comprehensive order processing capabilities for turbine engines at AEDC.

CHAPTER TWO

REVIEW OF SELECT SIGNAL PROCESSING CONCEPTS

Time Domain

General Information

This section covers pertinent concepts, standards, and practices related to vibration data acquisition and analysis. It by no means covers every aspect of dynamic data analysis, but simply lays a foundation for the development of the described order processing approach.

As stated by Bendat and Piersol [4], sinusoidal data are those types of periodic data that can be defined mathematically by a time-varying function of the form

$$x(t) = X \sin(2\pi f_0 + \theta)$$

Equation 1.

The period and frequency are directly related by

$$T_P = \frac{1}{f_0}$$

Equation 2.

These equations are the primary foundations for vibration acquisition and analysis.

In turbine engines, vibratory responses that are related to rotational speed are referred to as integral responses (synchronous). Integral responses have an “order” associated with them. In non-mathematical terms an order is defined as how often an event (oscillation) occurs in a single shaft revolution. In turbine engine analysis they are referred to as engine orders (EO). For integral responses, orders are integers and are referred to as Integral Engine Orders (IEO). Orders are expressed mathematically as:

$$EO = \frac{60f}{N}$$

Equation 3.

Similarly, frequency can be expressed mathematically as:

$$f = \frac{N(EO)}{60}$$

Equation 4.

For example, vibratory stresses are being measured on a compressor rotor (i.e. rotating) blade that is just downstream of a stator (i.e. stationary) stage with 20 blades. Each time the compressor blade passes a stator, the compressor blade will experience a vibratory excitation. In a single rotation, the compressor blade will experience 20 excitations from the stators. This would be referred to as a 20 EO response. For integral responses such as this, the frequency of the response measured on the compressor blade would be proportional to engine speed. If the engine rotates at 6,000 RPM, the frequency of the 20 EO response on the compressor blade would be 2,000 Hz. In standard FFT processing, orders are secondary calculations that are only possible after data has been transformed to the frequency domain; orders are not yielded directly from FFTs.

It should also be mentioned that every component has multiple vibratory modes, as determined by analytical or test analysis. Each mode has a different mode shape and resonance frequency. When an engine is accelerating, following constant EO lines, different modes will be excited as changing rotational speeds changes the excitation frequency proportionally. Eventually, the excitation frequency will approach and equal the resonance frequencies, which leads to a relatively sharp amplitude increase. The rotational speeds at which this happens are referred to as resonance speeds or crossing speeds (where the excitation frequencies "cross" the natural resonance frequencies). Figure 4 shows a Campbell Diagram, which includes rotational speed (N), frequency (f), order (O), and amplitude in a single figure. The diagonal lines are lines of constant order numbers, and the red blooms are modal resonances and their vertical lengths correspond to amplitude (z axis).

Instrumentation and Sampling

As previously mentioned, vibratory stresses are of major concern in rotating equipment, especially with respect to aircraft turbine engines. These vibratory responses can be measured and quantified with strain gages, accelerometers, and tip timing systems (also known as Non-Contact Stress Measurement Systems, NSMS). Common amplitude types include stress (KSI or PSI), deflection (mils), velocity (in/sec), and acceleration (G).

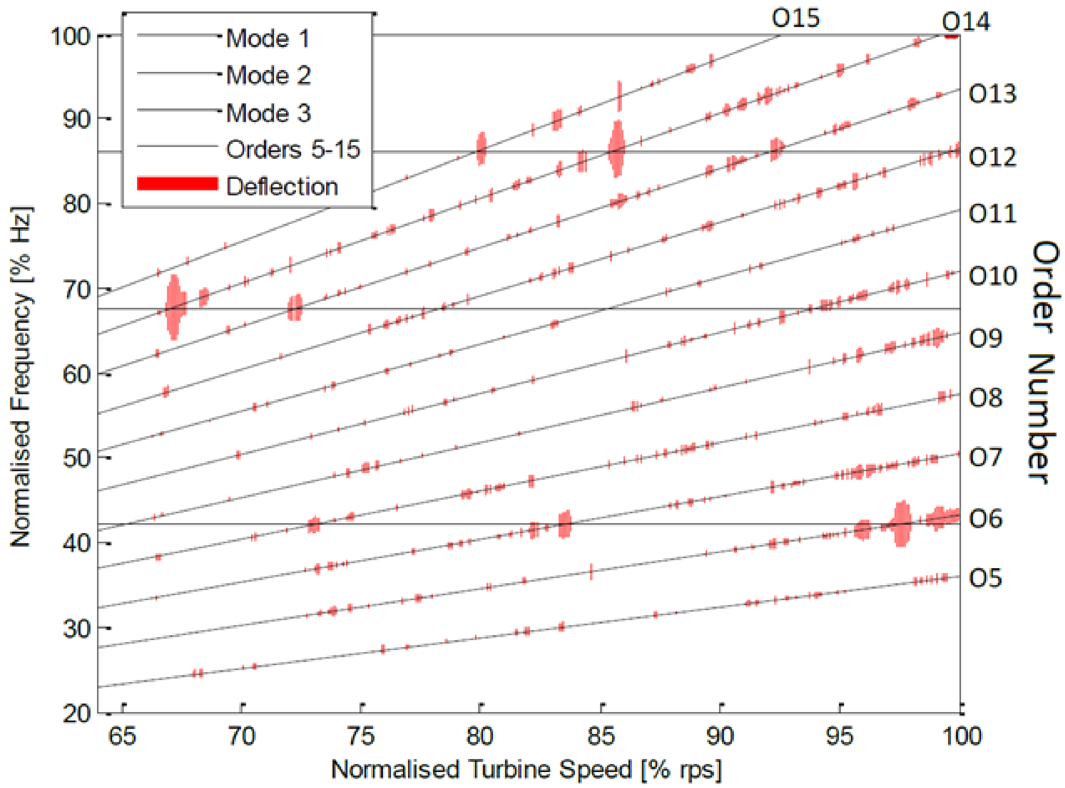

Figure 4. Typical Campbell Diagram [11]

While each of these instrumentation types produce high-response, oscillatory signals, the focus henceforth will primarily be on aeromechanical excitations as measured by strain gages, with the assumption that all applications are dynamic in the presence of component rotation. Strain gages are devices that measure surface strain. They are installed, usually by some kind of epoxy, to the surface of both rotating and stationary components. Strain gages have a small integrated circuit that has a calibrated electrical resistance. As the surface containing the strain gage is compressed or stretched, the electrical resistance of the strain gage changes. This results in varying amounts of voltage, usually in millivolts, at different strains. For stationary components, strain gages can be wired directly to the acquisition system. Rotating components are usually wired by means of slip rings (which can induce high levels of noise) or transmitted via a wireless telemetry system.

When a turbine engine is operating, strain gages produce continuous oscillatory signals. Acquisition systems “sample” these continuous signals. Sample rates are usually quantified in terms of Hertz or samples/second. The higher the sample rate is, the closer the data points will come to representing the true continuous signal as produced by the dynamic instrumentation. Figure 5 shows a comparison between a continuous 500 Hz signal with sample rate of 5 kHz and a lower sample rate of 1.2 kHz. It can be observed in Figure 5 that the curve generated from the lower sample frequency cannot be made to accurately represent the true 500 Hz signal.

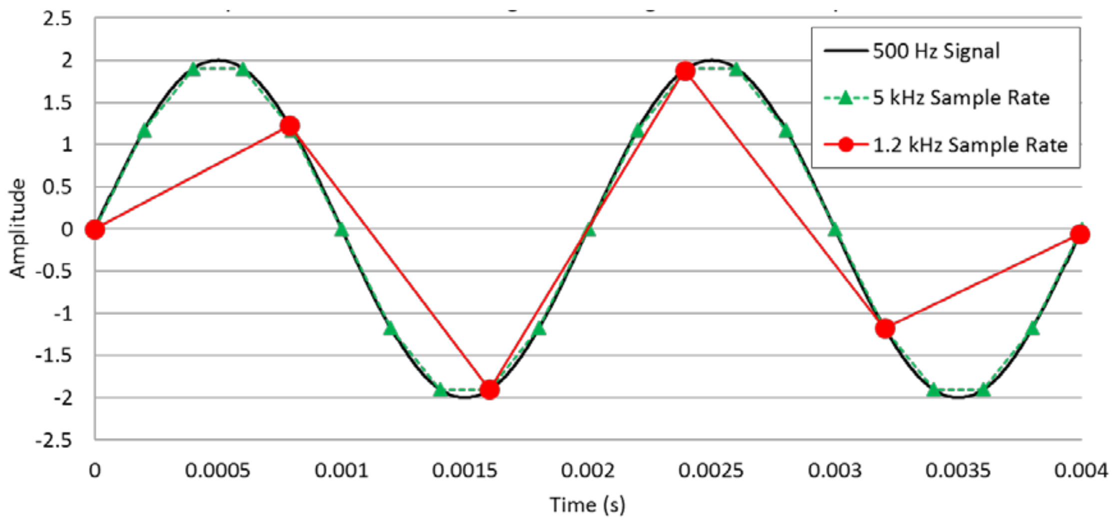

Figure 5. Continuous Curves and Sample Generated Time Waveforms

As the sample rate increases, the generated curve approaches the true 500 Hz signal curve, as shown in the 5 kHz sample rate curve.

While it might be desirable to have exceptionally high sample rates, there are hardware and data storage limitations to consider. A suitable sample rate must be determined for each application – one that is high enough to capture all frequencies of interest, but not superfluous so as to create unnecessary data storage requirements.

Fourier Transforms (Frequency Domain)

Overview

Sampling of data takes place in time domain which generates a complex time waveform like the one shown in Figure 2. While there are some special analysis techniques that make use of the time waveform, most dynamic data analysis takes place in the frequency domain. Data acquired (and usually stored) in the time domain must be transformed to the frequency domain via the FFT process. As a result of the FFT process, the frequency spectrum will contain peaks representing the amplitude and frequency of each periodic cycle of vibration contained in the waveform [2]. The main advantage of knowing frequencies is the ability to relate responses to specific components of machinery. This allows the analyst to see vibratory responses on a component basis and allows for tracking of multiple components' amplitudes in a single spectrum. Figure 6 shows a good representation of the how the frequency domain data is derived from the complex time domain wave form [5].

Summation of Periodic Functions

A key premise for Fourier transforms is the Fourier series. Complex periodic data may be expanded into a Fourier series. This is made evident in the Discrete Fourier Transform (DFT) summation equation.

$$X(f,T) = \Delta t \sum_{m=0}^{M-1} x_m e^{-j2\pi f m \Delta t}$$

Equation 5.

Figure 7 shows a visualization of the summation of sinusoids.

Nyquist Frequency and Aliasing

One very important consideration in aeromechanics (and vibration analysis in general) is the Nyquist Frequency. The Nyquist Sampling Theorem, also known as the Sampling Theorem, states that the sample frequency must be at least 2 times the highest frequency of interest [4]. Due to limitations of signal filters, the industry standard Nyquist Frequency is now considered to be 2.56 times the highest frequency of interest, known as the guard band ratio [5, 12]. In other words, there needs to be an average of at least 2.56 samples in each period of each frequency of interest. This is employed to avoid aliasing when executing Fourier Transforms.

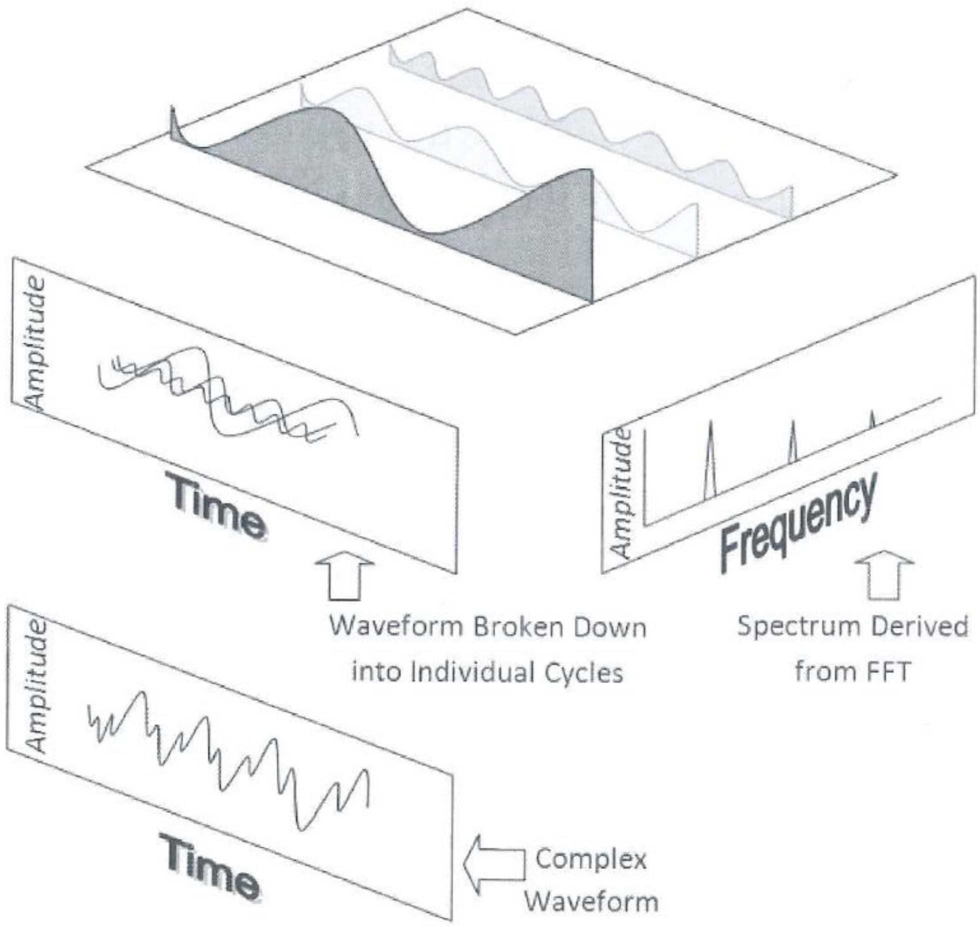

Figure 6. Visualization of the Time and Frequency Domains [5]

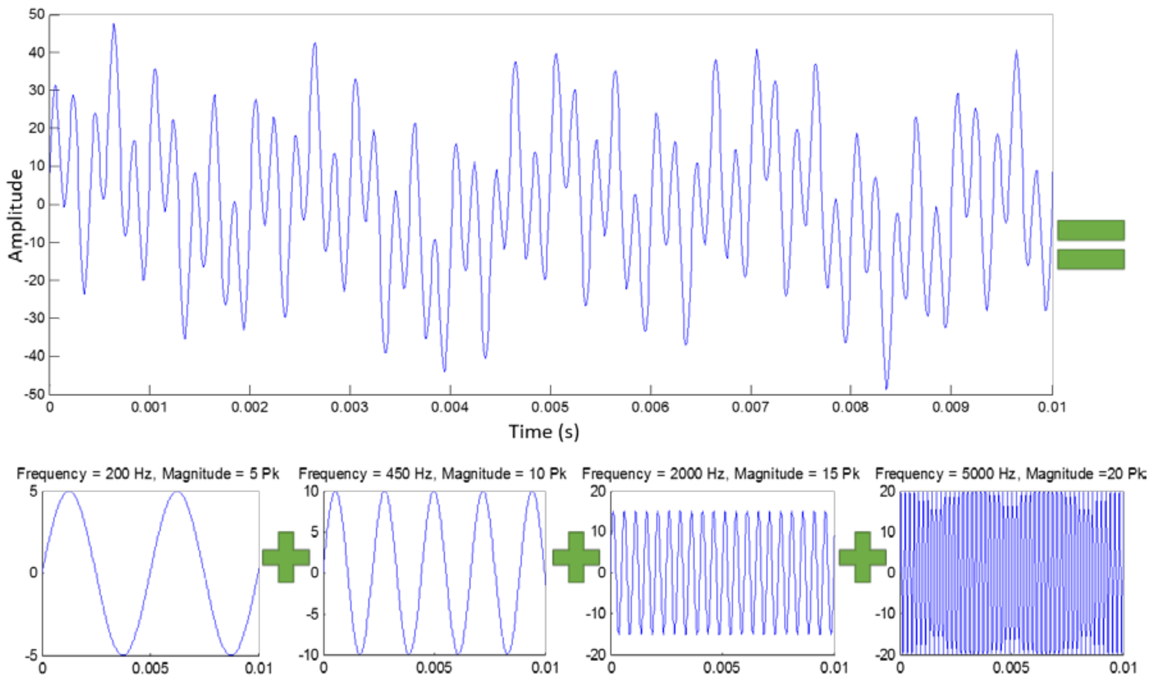

Figure 7. Visualization of Summation of Sinusoids

Aliasing is the misidentification of frequencies by FFTs. For example, the true frequency of a continuous signal is 1000 Hz. The Nyquist sampling frequency would be 2560 Hz. If the sample rate was less than 2560 Hz, the true 1000 Hz signal could be misidentified as 500 Hz (1/2 true frequency) or 250 Hz (1/4 true frequency) in the frequency domain. Figure 8 shows a signal frequency of 1000 Hz overlaid with an aliased frequency of 250 Hz (¼ of the true frequency), resulting from undersampling at a rate of 2507 Hz.

FFT Parameters

For most applications, FFTs are executed successively in a stream of data, in both monitoring (real-time) and post-analysis (playing back and processing recorded data files). In order for an FFT to yield accurate results, various FFT parameters must be correctly specified.

- *FFT Sample Size*, also referred to as Block Size or FFT size, is simply the number of samples that are included in the FFT. Due to the base-2 nature of computers, FFT sizes are almost exclusively 2^x (…256, 512, 1024, 2048…), making real-time FFTs (i.e. monitoring capabilities) possible. If FFT sizes $\neq 2^x$ were used, there would be a significant processing lag time unless high-performance computers were used.
- *Sampling Frequency* or *Sample Rate* (F_S) is the rate at which a signal is sampled. The units are usually in Hz (samples/second).
- *Sample Time Spacing* (Δt) is the time between samples taken in time, which is determined by the sample frequency.

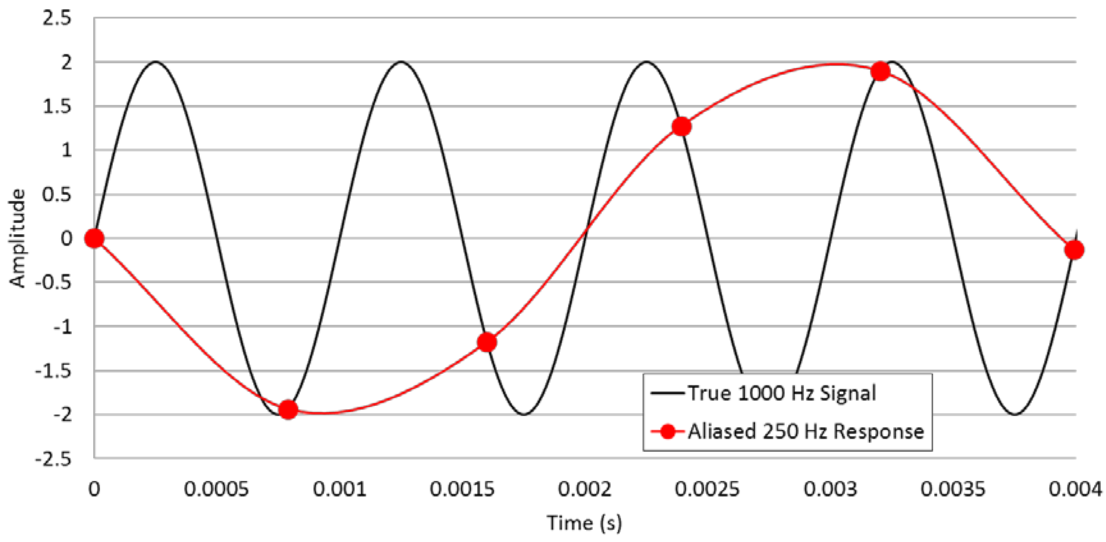

Figure 8. Example of Aliasing

- *Sample Size (SS),* is sometimes used interchangeably with FFT size. However the sample size and FFT can be different.
- *Time Waveform Length* (TWL) is the time it takes to sample one FFT of data. The TWL is directly proportional to sample rate and FFT size (FFTs).
- *Max Frequency* (F_{max}) is the highest frequency of interest. This parameter is usually used to specify the low pass filter (anti-aliasing filter) threshold and also determines the sampling rate (via Nyquist Sampling Theorem).
- *Bin Width* or *Frequency Resolution* (ΔF) is the space between discreet frequency lines (in Hz) in the frequency domain. In general, discreet amplitudes become more accurate as bin width becomes smaller. Frequency resolution is directly dependent on sample rate and FFT size (sample size).
- *Lines of Resolution* (LOR) are the number of discreet frequency values shown in the spectrum. LOR increases as Fmax becomes smaller and vice-versa. Sometimes the true frequency falls between LOR. Therefore, the effects of that are minimized with better frequency resolutions. Some data acquisitions allow the user to specify the LOR directly.
- Pertinent equations are as follows:

$$\Delta \mathrm{F} = \frac{Fmax}{LOR} = \frac{Fs}{FFTs}$$

Equation 6,

$$F_s = 2.56(Fmax)$$

Equation 7,

and

$$SS = F_s(TWL)$$

Equation 8.

Windowing and FFT Overlap

Another very important consideration for FFTs is Windowing. The FFT process assumes that the time wave form is a continuous and periodic [6]. So the first and last sample of the FFT must be resolved to be the same value [6]. Due to noise and dynamic system changes, the beginning and end sample values of each FFT are almost always different in sampled data sets. Windowing functions act as pre-FFT multipliers (usually from 0 to 1) and resolve the beginning and end points of the sampled data to be the same (usually 0), that is to make the sampled data periodic. National Instruments describes windowing as “reducing the amplitude of the discontinuities at the boundaries of each finite sequence acquired by the digitizer” [6]. The windowing function type directly affects the data as reported in the frequency domain. For simplicity’s sake, consider a sampled data set where every value is 1. If a window is applied to this data set, the new set will include values ranging from 0 to 1 along a curve that starts and ends at 0. Figure 9 shows the Hanning window (as applied to a data set with values all equaling 1), which is by far the most commonly used window type.

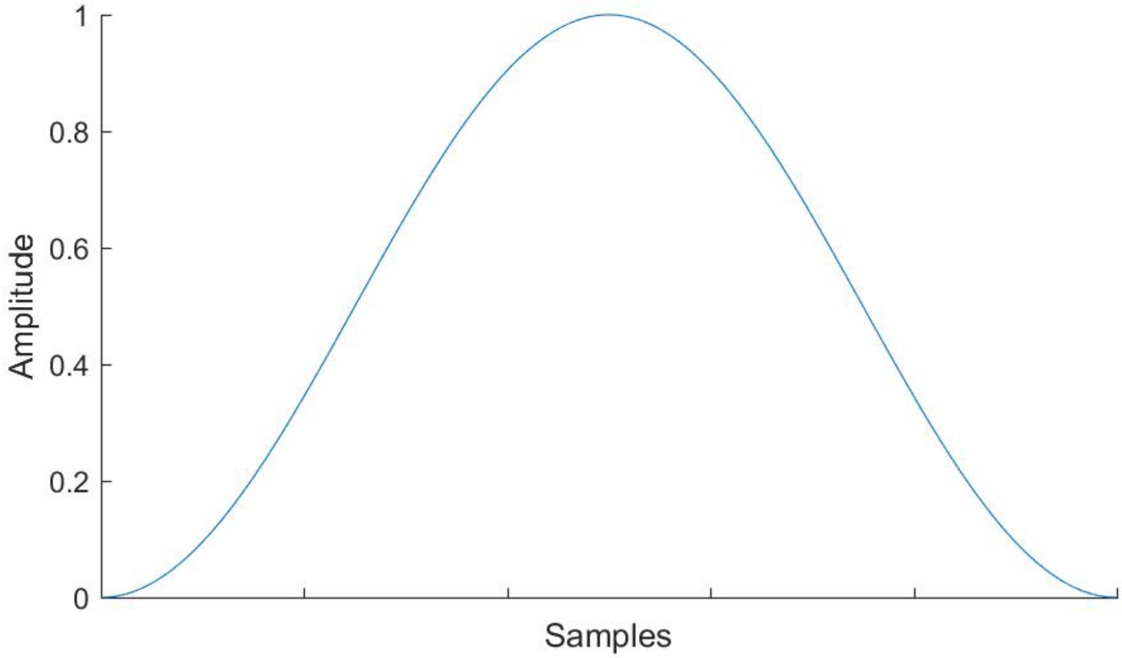

Figure 9. Hanning Window Curve

Figure 10 shows many of the various types of window functions that can be applied. Figure 11 shows a complex time waveform before a windowing function is applied. Figure 12 shows the same complex time waveform shown in Figure 11 after applying the Hanning window. This is a realistic view of how windowing "prepares" the sample set for the FFT.

After a window is applied to a sample set, it can be observed that the "best" or least-skewed data is in the middle where the multiplier is closest to 1. This means that the data on the edges of the window are skewed. In order to account for this problem, overlap is often applied. This means that samples are retained so that a portion of the samples used in the previous FFT can be used in the subsequent FFT. This allows each sample to be processed in the less-skewed portion of the window. If 50% overlap is specified, the current FFT sample set will begin with the middle sample of the previous FFT sample set. Depending on which type of window is applied, overlap may not be necessary. The square window for example has an immediate jump to 1 after the 1st sample that is made to be 0, stays at 1 throughout the sample set, then returns to 0 at the last sample. Figure 13 illustrates 50% overlap.

One of the problems associated with windowing is that the overall energy of the signal is decreased by the window function. In other words, each window has its own inherent gain, and most window's gains are less than 1. Therefore, the amplitudes as reported in the spectrum are significantly lower than the true signal's amplitudes. This can be accounted for by applying a gain.

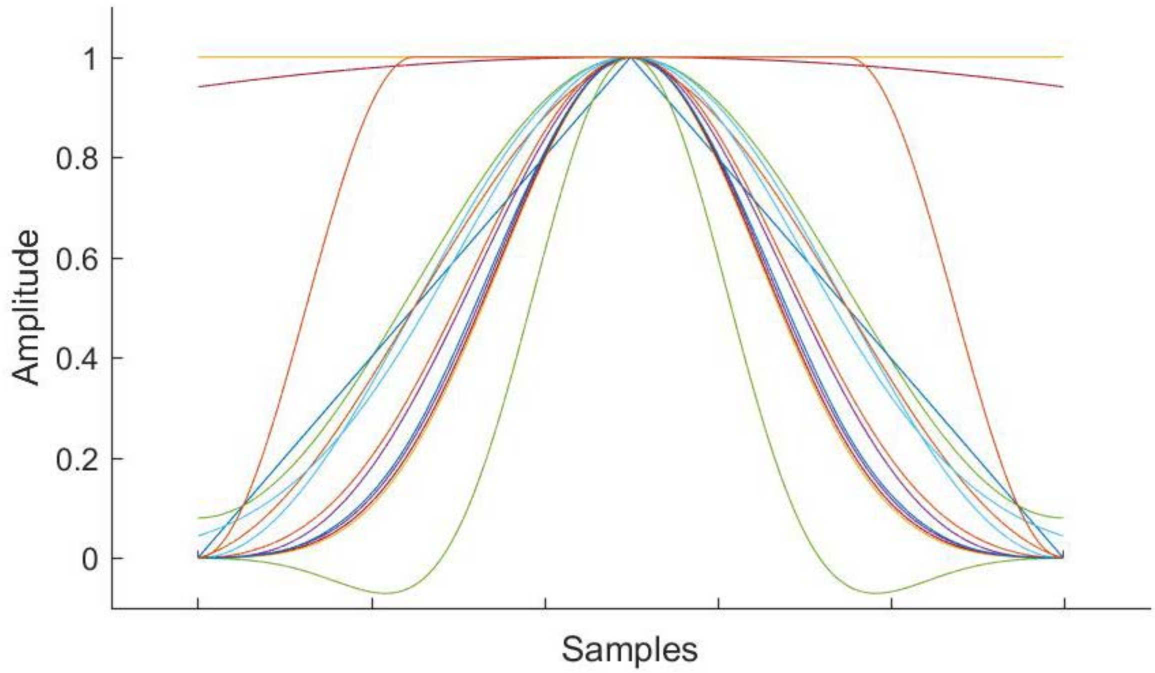

Figure 10. Various Windowing Curves

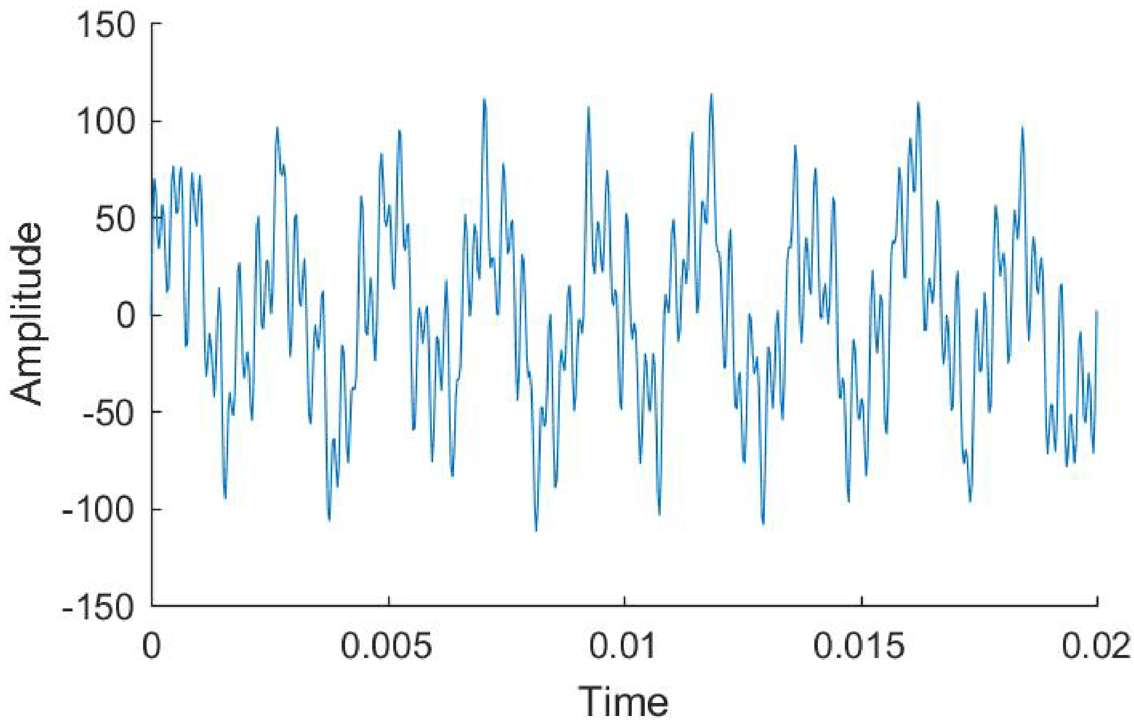

Figure 11. Complex Time Waveform with No Windowing Function

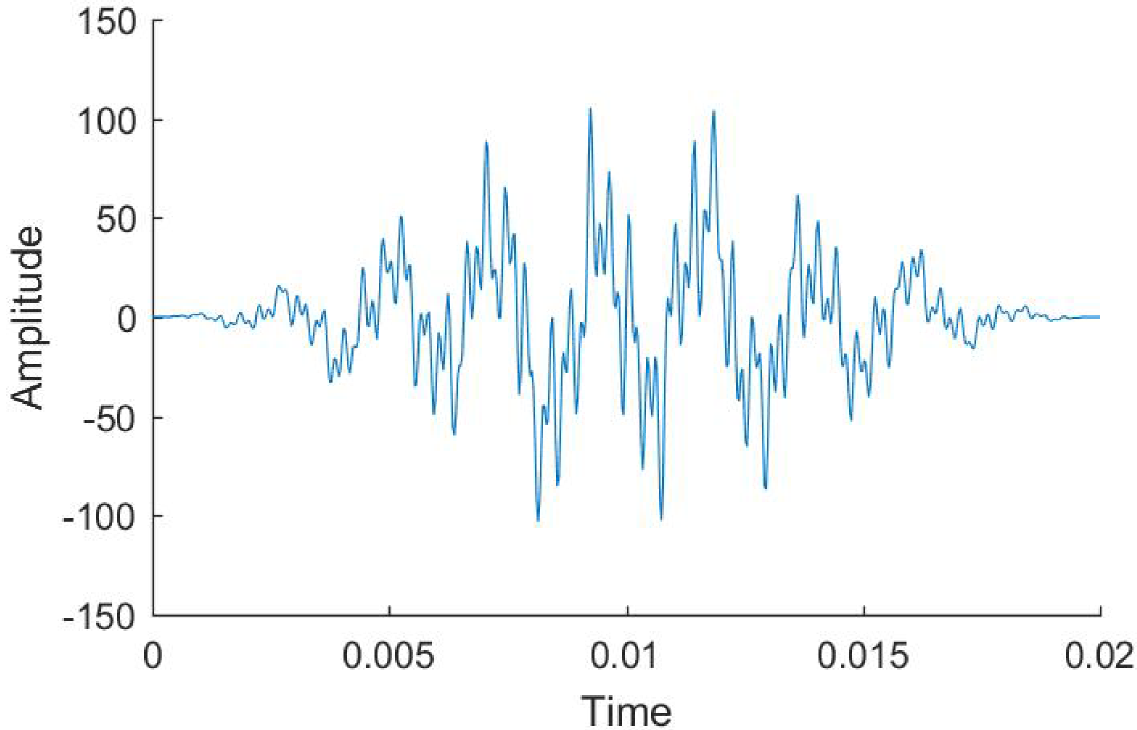

Figure 12. Complex Time Waveform with Hanning Window Applied

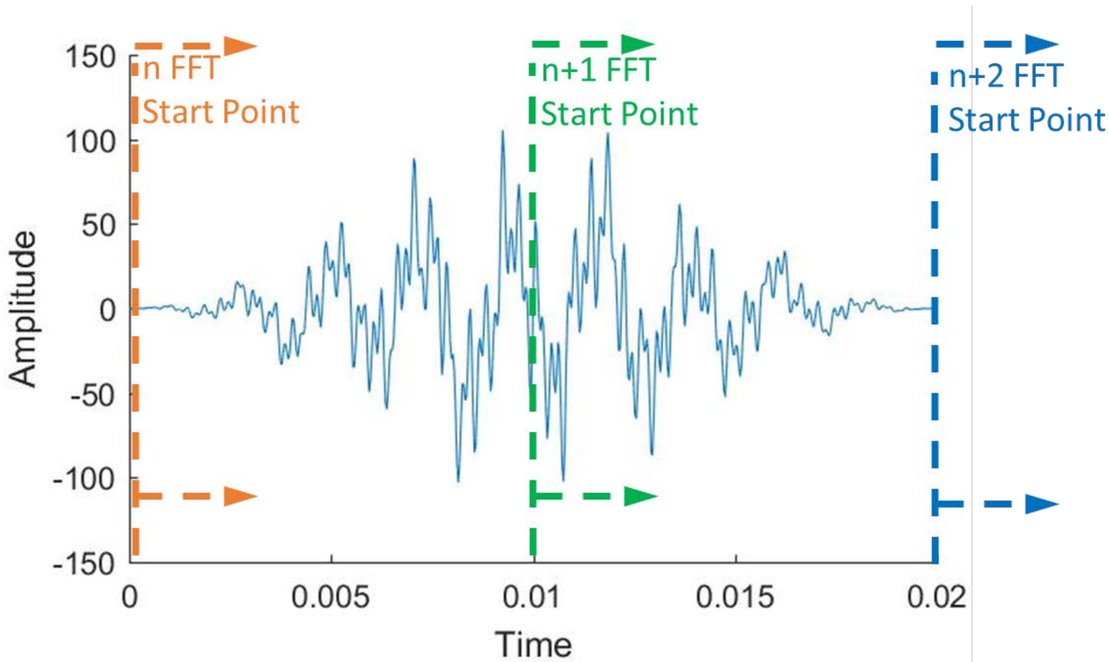

Figure 13. Illustration of 50% FFT Overlap

For the Hanning window, the correction gain is 1/0.49, which counters the inherent Hanning window gain of 0.49. The rectangular windows inherent gain is 1.0, so no correction gain is required.

Spectral Leakage

One of the reasons the Hanning window is so commonly used is because, when compared to other windowing types, it has significantly lower spectral leakage [6]. Spectral leakage is when vibration energy at a discreet frequency is erroneously presented in the frequency domain as tapering magnitudes at frequencies adjacent to the true frequency. This results in a smearing of amplitudes in the frequency domain. Ideally, all responses shown in the frequency domain would be vertical lines with no width [6]. However, this is not possible because in reality, the number of *periods* sampled for a given frequency are not integers. Windowing strives to make amends for this offense, but also results in some level of spectral leakage [6].

Among many other commonly used windows are the Hamming, Blackman-Harris, Flat Top, and Square windows. As specified by National Instruments Table 1 [6], there are different types of windowing functions, each optimized for different scenarios.

FFT Refresh Rates

The FFT is employed for both real-time (monitoring) and post-analysis (play back) applications.

Table 1. Appropriate Windows for Various Signal Contents [6]

Signal Content	Window
Sine wave or combination of sine waves	Hanning
Sine wave (amplitude accuracy is important)	Flat Top
Narrowband random signal (vibration data)	Hann
Broadband random (white noise)	Uniform
Closely spaced sine waves	Uniform, Hamming
Excitation signals (hammer blow)	Force
Response signals	Exponential
Unknown content	Hanning
Two tones with frequencies close but amplitudes very different	Kaiser-Bessel
Two tones with frequencies close and almost equal amplitudes	Uniform
Accurate single tone amplitude measurements	Flat Top

In the real-time settings, the refresh rate is directly related to the FFT size and sampling rate. Each FFT must "wait" for the required number of samples to be acquired. So, intuitively, as FFT size decreases and/or as the sample rate increases, the real-time refresh rate increases. The refresh rate can increase to a point that cannot be supported by standard computer hardware; there may be a computer "muscle deficit." Sometimes commercial monitoring software can impart refresh limits as well. When the refresh limit is reached, the FFT size must be increased. It is unlikely that the sample rates will be changed to lower the refresh rate because of the Nyquist frequencies which drive the required sample rates. On the other hand, as FFT size gets larger or as the sample rate decreases, the real-time refresh rate decreases. The most common FFT sizes employed in turbine engine testing with strain gages are 1024 and 2048. According to computer engineers and signal processing engineers at AEDC, lower refresh rates are usually considered to be a non-issue, since the refresh rate changes from 1024 and 2048 FFTs are hardly noticeable by those looking at real-time frequency domain data.

CHAPTER THREE

ORDER PROCESSING

Order Processing vs. Standard FFT Processing

The primary difference between standard FFT processing and order processing is how samples are collected and organized before undergoing the FFT process. In the standard FFT process the number of samples grouped for FFTs is solely dependent on the FFT size, and the time required to acquire those samples is fixed by the sample rate. This means that the first and last sample of the FFT blocks are completely arbitrary as related to engine rotation and integral vibration periods. This results in multiple revolutions being included in each successive FFT. For example, if an engine was rotating at 16,000 RPM, being sampled at 38 kHz, and an FFT is executed with 1024 samples, 7.19 revolutions will be included in the FFT.

The simplistic approach to order processing being considered is to execute a single FFT for each shaft revolution. At a fixed sample rate, this would mean that the number of samples in each revolution of data would change with varying rotational speeds. In standard FFT processing, the FFT “waits” for a set number of samples to be acquired. Order processing FFTs would “wait” for the 1/REV signal to pulse before being executed. This creates a data set challenge because FFT sizes are set to be a set value of exactly 2^x. For example, the sample rate of a piece of dynamic instrumentation is 102,400 Hz. At an engine speed of 5,000 RPM the number of samples acquired in a single revolution would be roughly 1229 with a waveform time of 0.012 seconds. If the engine accelerates to 15,000 RPM, the number of samples in a single revolution would drop to roughly 410, with a waveform time of 0.004 seconds. As expected, there is an inverse relationship between rotational speed and total samples per revolution. In this case it is a factor of 3. In order processing, each sample size must be resolved to be exactly the fixed FFT size. Windowing is also required in order processing FFTs. Since the first and last samples are no longer arbitrary in order processing, FFT overlap is no longer a simple option with a 1/REV sensor. However, there are sensors that pulse at evenly spaced angular increments that would allow for accurate FFT overlap.

Existing methods that make use of 1/REV processing, such as blade tip timing systems, can offer valuable information to help address the challenges associated with generalized order processing.

Review of Blade Tip Timing Processing Methods

Tip timing systems, also known as Non-contact Stress Measurement Systems (NSMS), is a method of analyzing blade tip deflections of turbine engine fan, compressor, and turbine blades. NSMS is employed in both monitoring (real-time) and post-analysis contexts. NSMS data acquisition is all based on single revolutions of data, which is the reason this method is being researched. NSMS uses lasers located in asymmetric circumferential locations, pointing inwardly radially at rotating blade tips. NSMS uses 1/REV sensors to act as a time reference signal and controls the start and stop each block of data [7]. The result is exactly 1 revolution of data for each successive revolution. One key difference between NSMS and other acquisition methods is that the geometry of the test article must be known and specified in the NSMS system - blade count, radius, rotational speeds, and 1/REV sensor locations among other information [7].

When the 1/REV “fires,” it begins recording data for that revolution. Lasers are pointed toward the blades, and when blades pass, the response is reflected back to the detectors surrounding the laser probes. This occurs for all blades on all NSMS sensors. An algorithm then calculates the deviation from an ideal, non-vibrating rotating stage of static rotors blade tips [7]. FFT overlap is not possible because FFT sample set start and stop times are always in reference to the 1/REV. Therefore, NSMS systems run 2 concurrent FFTs with different windowing functions: one for determining frequencies and a different one for determining amplitudes, then merging the data in the frequency domain. Through years of research, development, and testing, the best FFT windowing options have been determined for NSMS 1/REV FFTs. To minimize skewing of calculated frequencies, NSMS uses the square window (also known as the rectangular window). Similarly, the Hanning window is used for running FFTs for amplitude.

One of the main challenges associated with order processing is that of managing FFT sample sizes in the midst of changing rotational speeds while sampling at a fixed frequency. In NSMS applications, this FFT size management is accomplished by means of a combination of “zero padding” and decimation. Zero padding is adding zeros to an FFT data set when there aren’t enough samples acquired in a single revolution of data to fill the FFT sample slots. Decimation is the opposite in that more samples than are needed are acquired and excess samples are effectively removed.

Order Processing for Turbine Engine Testing

Capability Interests

Order processing for both real-time (monitoring) and post-processing has applications in the context of turbine engine testing. Order processing capabilities have been requested verbally for many years, and are being considered for a data acquisition system that is being developed. As previously mentioned, there have been failure events in the past that were not captured by standard FFT processing methods [11]. The capability to run FFTs for each individual revolution (synchronous with rotation) on demand would be a valuable addition to existing standard FFT processing methods. Order processing would provide a different perspective of dynamic data.

In the time domain the benefit would be to see exactly one revolution of data in digital oscilloscope plots. Unique responses could then easily be correlated to angular locations on the engine, with a small amount of error that comes from averaging angular accelerations. With standard FFT methods the highest peak shown in the frequency domain may be significantly less than the true response’s highest peak. Order processing would ensure that averaging across multiple revolutions does not occur.

One of the primary challenges associated with this type of FFT processing is the management of sample sizes for fixed FFT sizes, and the effects thereof.

Windowing

Following the NSMS method, two windows and FFTs were to be used concurrently for order processing. The Hanning window and the rectangular window would be used and the frequency and magnitude accuracies would be compared. Additionally, since the first and last samples are no longer arbitrary but are at the same exact location, overlap is no longer a straight-forward option for order processing.

Zero Padding

As rotational speed changes, the total number of samples per revolution will inversely change. At higher speeds, zero padding would be required to fill all the FFT sample slots. These additional zeros can be placed before, after, or split before and after the acquired samples.

Zero padding can be used for several different reasons. In the case of order processing, the purpose of padding in this initial approach is simply to get a sample size of 2^x. A distinction needs to be made between zero padding and upsampling. Upsampling does increase the sample size, but it also includes a function that creates synthetic samples with interpolated values. Zero padding is sampling the addition of zeros before, after, or before and after the sampled data. Upsampling inserts the interpolated values *between* the acquired samples. *It is important that zero padding be applied after the window is applied*. In most cases windowing results in a smooth curve of the data from and to a value 0. If zero padding is applied before the window function is applied, the real data will experience a sharp increase in amplitude, which would defeat the purpose of windowing. Since the zeros are added after the windowing function, it does not matter if the zeros are placed before, after, or split before and after the data. However, the zeros must be placed at the same location as the padding on the time array. For the sake of simplicity, zeros would be placed after the acquired samples. Figure 14 shows a visualization of zero padding after the window has been applied in the time domain [9].

Decimation (Down Sampling)

At high sample rates and at lower rotational speeds, there can potentially be an excess of samples (i.e. more than the FFT calls for) acquired in a single shaft revolution. In this case, a combination zero padding and decimation would be required to resolve the samples into exact FFT size. When applying decimation, or downsampling, it is important to note that the effective sample rate is thus reduced using a skip-sampling method (i.e. every other sample, every third sample, etc.) and the effective Nyquist frequency must be considered. If this is overlooked, frequencies of interest could be aliased to lower frequencies. Figure 15 shows a visualization of decimation of time domain data [8].

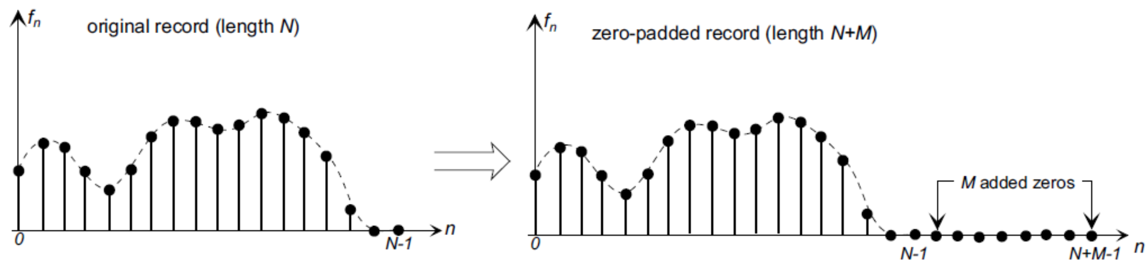

Figure 14. Visualization of Zero Padding [9]

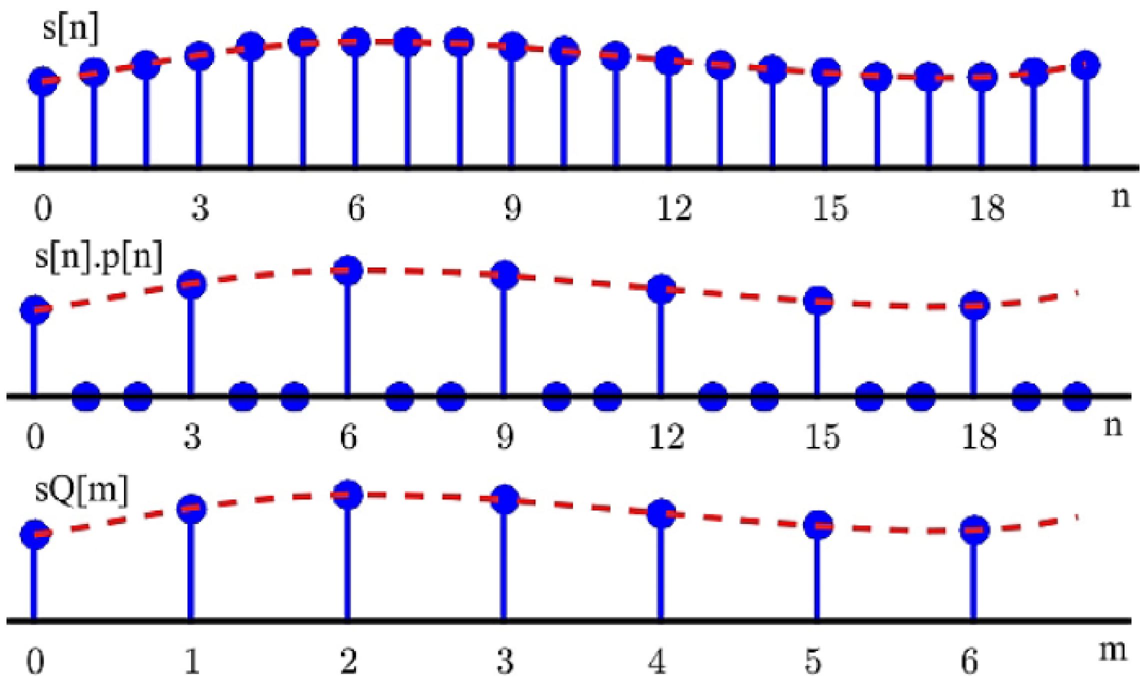

Figure 15. Downsampling (Decimation) s[n] by Q = 3 in the time Domain [8]

CHAPTER FOUR

PROCESSING ALGORITHMS USING MATLAB

In order to test the order processing approach discussed in the previous chapters, MATLAB was used to generate various data sets and conduct analysis in both the time domain and frequency domain. MATLAB is a software suite that "combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly," © MathWorks®. In order to compare the frequency domain results from standard FFT processing and order processing FFTs, it is important to start with data sets where the exact answer was known beforehand. Using data acquired in a real testing environment alone would not have given conclusive results. This is partly due to the large margin of error associated with dynamic data instrumentation and processing.

Various types of data sets can be generated using MATLAB, using values that represent typical turbine strain gage engine responses, including realistic random noise. These data sets can vary in size, as if created at varying rotational speeds. The Hanning window, rectangular window, and no window were applied to the data sets. The data sets that were not equal to 2^x were resolved to a 2^x size by means of padding or decimation as required. MATLAB was then used to run FFTs of different sizes. This process is explained in greater detail below.

Generation of Time Domain Data Sets

Synthetic data sets were generated in MATLAB. In turbine engine strain gage analyses, the highest frequency of interest is typically 5,000 Hz, resulting in the conservative Nyquist sampling frequency of 12,800 Hz. It should be noted that in reality, the sample rate is usually well above the Nyquist sampling frequency. Each data set included discreet frequencies of 200 Hz, 450 Hz, 1200 Hz, and 2300 Hz, 3700 Hz, and 5000 Hz. The amplitudes were set at 10, 50, 15, 30, 8, and 25, respectively. MATLAB’s random noise generator was also used and incorporated into the data sets. Every data set generated contained these responses and low amplitude noise. Figure 16 shows a typical data set in the time domain before applying a windowing function.

MATLAB FFTs

MATLAB includes native FFT capabilities, and includes the ability to execute FFTs with block sizes that are not equal to 2^x. If the sample size is not equal to the FFT size specified by the user, MATLAB automatically adds zero padding or truncation. This automatic MATLAB function was not used in the analyses described herein. Instead, zero padding and decimation were applied manually. In the interest of order processing, which seeks to include exactly one revolution of data in each FFT, truncation is not permissible. At low speeds the sample size may be larger than the specified FFT size, but downsampling and zero padding was used in lieu of truncation.

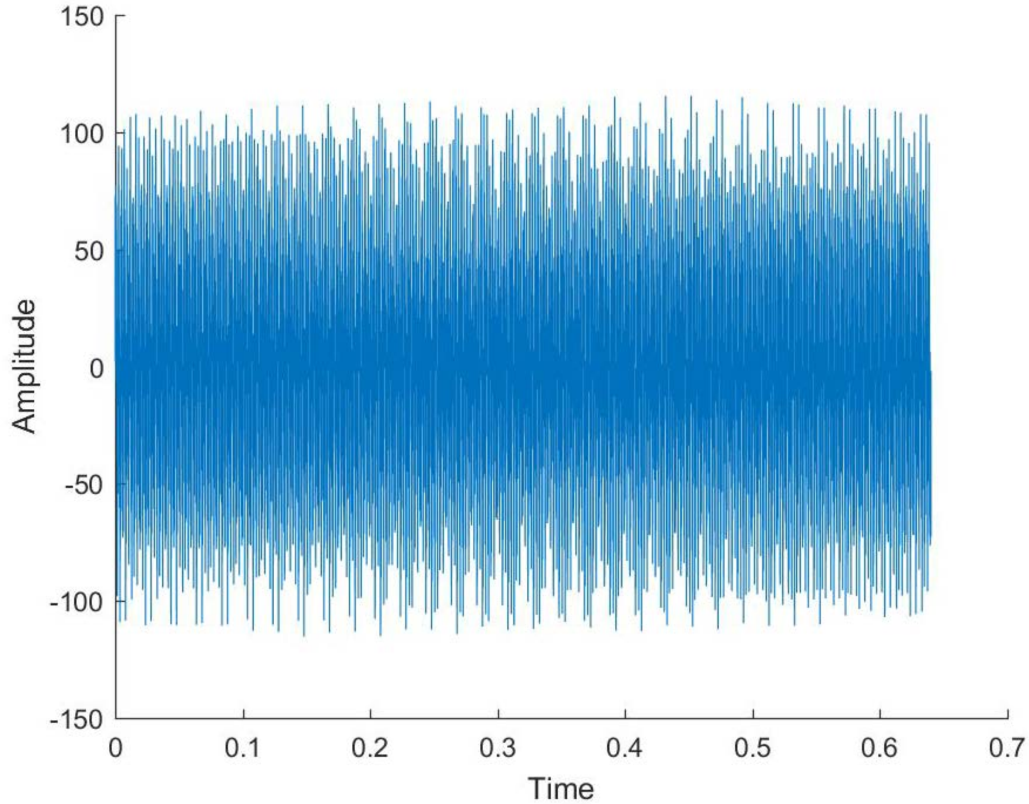

Figure 16. Typical Time Domain Data Set without Windowing

In order to track the amplitude changes with zero padding or downsampling, all sample sizes were resolved to exact FFT sizes before applying the FFTs in MATLAB. Data acquisition systems can support sample rates ranging from a few thousand hertz up to 204,800+ Hz. For standard FFT processing, the sample rate used in all cases was 12,800 with FFT sizes ranging from 64 samples all the way up to 8192 samples. This sample rate makes use of the industry standard of 2.56 times the highest frequency of interest (5,000 Hz).

Order Processing (MATLAB)

Generation of Revolution-Based Data Sets (MATLAB)

When order processing is implemented in the acquisition system in real-world applications, the 1/REV sensor will “mark” the raw data, providing a sampling start and stop point for each FFT. This “marking” of data is a relatively simple task to implement via programming, but is not necessary for the analyses covered in this paper. Instead of synthetically marking data, data sets of varying sample sizes would be generated as they would be at various engine speeds. Therefore, data sets were generated based on the same fixed sample rate of 12,800 Hz and various typical engine fan and core speeds of 2000 RPM (384 samples), 4000 RPM (192 samples), 6000 (128 samples), 9000 RPM (85 samples), 12000 RPM (64 samples), and 17000 RPM (45 samples). For the sake of comparison, a data set of exactly 1024 samples was generated (750 RPM) to exactly match the specified FFT size of 1024. Figure 17 shows a typical padded time domain data set with a Hanning window applied.

Windowing (MATLAB)

Generated data sets of varying sizes can have various windowing functions applied to them. “Applying” simply means that the dot product is taken between the two arrays. Different windows were used to check for the minimum deviation of true accuracy of magnitude and frequency, as compared to the true input values. The windows that were used compared for amplitude accuracy between standard FFT processing and order processing were the Hanning window and rectangular window. The Hanning window had a correction gain of 1/0.49 applied to it to counter the inherent Hanning gain of 0.49. The rectangular window’s inherent gain is 1.0, so no correction gain was necessary.

Order FFT Scripts (MATLAB)

The challenge presented with order processing is not whether or not a stream of data can be “marked” and organized by a 1/REV signal. What is in question is the effect of the accuracy of frequency domain data when FFTs with fixed sizes are executed with varying sample sizes and different windowing types. The order processing scripts in MATLAB were made to read the generated data sets of various sizes after the windowing functions were applied. When a sample set was less than the specified FFT size, zero padding was employed *after* applying the window function to the data set.

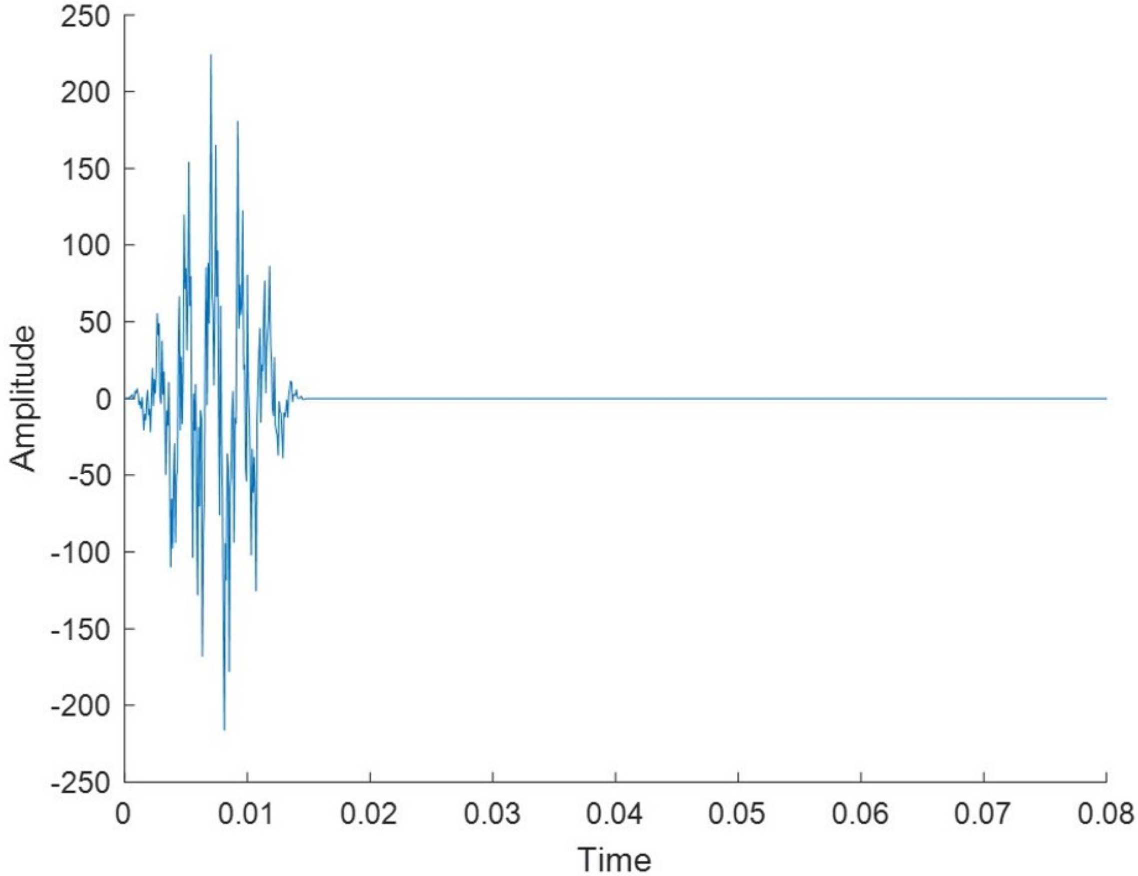

Figure 17. Typical Padded Time Domain Data Set without Windowing

Zeros were placed in the arrays *after* the sampled data. As previously explained, truncation is not an option for order processing.

CHAPTER FIVE

RESULTS AND DISCUSSION

Preliminary Discussion

The very nature of dynamic data brings processing difficulties, and anytime data is transformed from the time domain to the frequency domain varying degrees of amplitude and frequency inaccuracies are expected to be induced. In real world applications, special care should be taken to cater the processing parameters to meet the needs of each particular application. The analyses that follow reflect the varying error resulting from changing a few parameters, while some unconsidered parameters are left in a fixed state. Ideally each parameter is tried and compared against each other, while varying only a single variable. However, this creates an extremely large test matrix which is much larger than the scope of the research herein. In the real world, signal processing makes use of a combination of analog and digital processing tools to help refine the final data product. The goal of the following analyses was to observe the effects of changing a few key processing parameters as it relates to standard FFT processing and order processing.

A couple of points from chapter two are worth reiterating:

- Frequency resolution (or bin width) is the distance between discreet values (lines) in the frequency domain. In general smaller is better.

- In MATLAB, the maximum frequency shown in the spectrum is exactly half the sample rate, following the original Nyquist sampling theorem with a factor of 2. For example if the sample rate is 204,800 Hz, MATLAB will show a max frequency of 102,400 Hz in the spectrum.
- The number of lines shown, which is directly related to frequency resolution, is directly dependent on the FFT size by a factor of 0.5.
- While it is good to account for a margin of error, it is unnecessary to sample at a rate much higher than 2.56 the highest frequency of interest. If excessive sampling rates are used, the frequency domain will yield an unnecessarily low frequency resolution.
- The number of lines of resolution are set by the FFT size, and the way to improve frequency resolution is to lower the sample rate. One must be careful to keep a margin of error so that the sampling theorem is still satisfied.

Standard FFT Processing

General Observations

First, standard FFT processing parameters were varied for general observation. One thing that immediately became apparent is the criticality of the variables that dictate the frequency resolution. Before performing the research for this thesis, oversampling was considered to be a non-issue. Without a deeper knowledge of FFT parameters it would seem intuitive that a higher sampling rate would automatically make the results in the frequency domain more accurate. However, as previously discussed, the sample rate directly affects the maximum frequency in the spectrum. If the sampling rate is too high, the frequency domain will have the lines or resolution spread out over a larger frequency range for a given FFT size.

Figure 18 shows frequency domain data with an excessively high sample rate (102,400 Hz) and an FFT size of 1024 samples (no windowing function). Most of the lines of resolution are “wasted” on high frequencies that are above the highest frequency of interest (5000 Hz). Also worth noting is that there are significant amplitude losses associated with oversampling.

Figure 19 shows the same frequency domain data from Figure 18, but zoomed to the frequencies of interest. If the sample rate was decreased with the same FFT size, the same number of lines of resolution would be spread across a smaller frequency range, yielding a better frequency resolution. Figure 20 shows the same zoomed data with a much lower sampling rate (12,800 Hz), which is 2.56 times the highest frequency of interest (5000 Hz). The improved frequency resolution and higher amplitudes are apparent.

Recalling that frequency resolution is equal to the sample rate divided by the FFT size, at a fixed FFT size the frequency resolution gets increasingly worse (larger bin widths) as the sample rate is increased. At this point, methods of increasing the frequency resolution must be considered and applied to FFT input parameters.

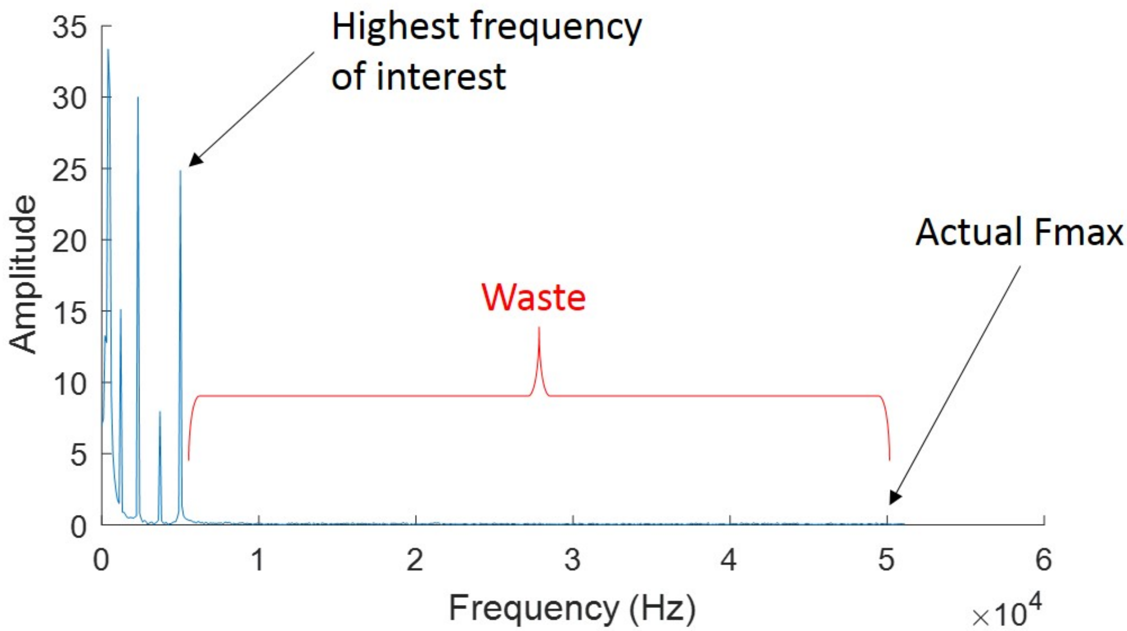

Figure 18. Spectrum with Excessive Sampling Rate

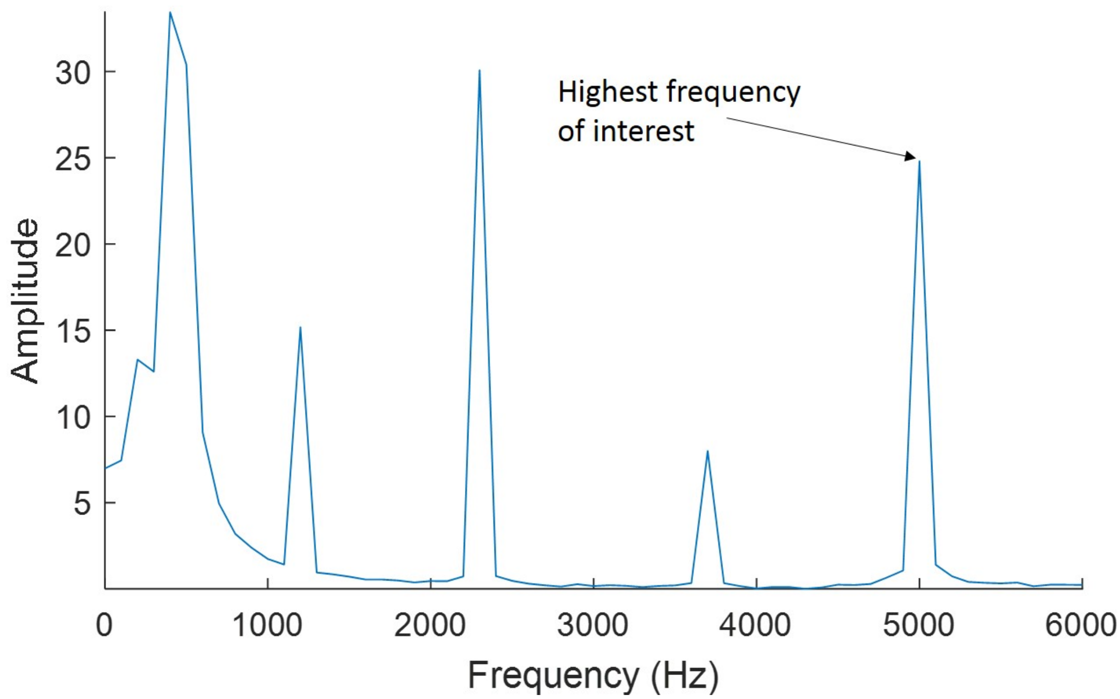

Figure 19. Zoomed Spectrum with Excessive Sampling Rate

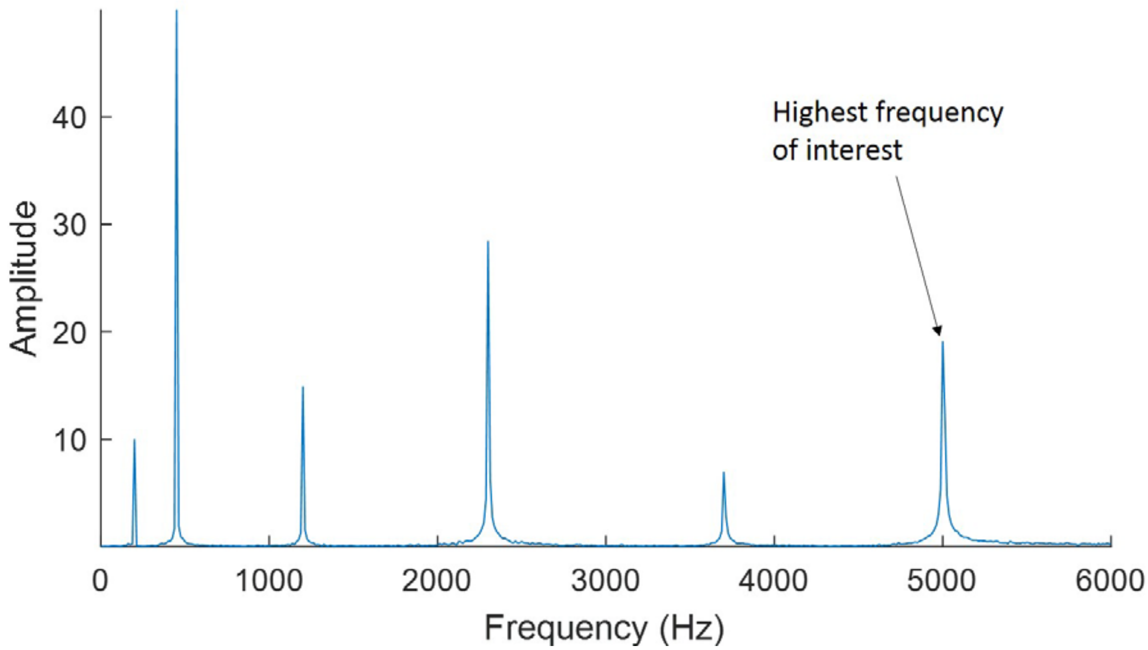

Figure 20. Zoomed Spectrum with Appropriate Sample Rate

One natural course of action would be to lower the sample rate so that the bin widths become smaller by moving the spectrum maximum frequency, while leaving the same amount of lines of resolution (same FFT size). This will in fact increase the frequency resolution. Another course of action would be to increase the FFT size, which increases the lines of resolution.

Another observation made was that amplitude error was not applied equally to all frequencies. This held true for all windows applied. Data sets that did not have windowing functions had accurate amplitudes at lower frequencies with increasing error at higher frequencies.

Data sets with the Hanning window applied also showed more accurate amplitudes at lower frequencies than at high frequencies. With the correction factor of 1/0.49 applied, the amplitudes were slightly higher than the true amplitudes at lower frequencies. At higher frequencies the amplitudes dropped beneath the true frequencies, until the error became increasingly worse at higher frequencies. As expected, data sets with the rectangular window applied were basically identical to data that had no windowing functions applied to them. Again, the rectangular window serves only to make the first and last samples 0 and effectively acts as a multiplier of 1.0 for the rest of the samples.

Frequency

In standard FFT processing, with FFT sizes at or above 256 samples, the frequency error was zero. While there was some error at lower FFT sizes, the errors were equal between the different windowing functions. This is likely due to the fact that the data was generated in MATLAB with round frequencies specified. Table 2 shows the frequency error of various frequencies and FFT sizes. **Note**: FFT sizes of 256 and larger yielded zero the frequency error; only FFT sizes of 64 through 256 are shown. It should be noted that FFT sizes this small are rarely used in turbine engine vibratory applications.

The realistically-sized FFTs, greater than or equal to 256, are shown to have no frequency error while being testing in the confines of a MATLAB environment. As the frequency resolution becomes coarser (wider bin widths) with smaller FFT sizes, the frequencies are more likely to fall on a line of resolution that is further away from the true frequency.

Amplitude

As expected, the amplitude accuracy improved for all cases as the frequency resolution increased. For FFT sizes above 64, the cases which had Hanning windows applied (with a 1/0.49 correction gain) yielded more accurate amplitudes. Table 3 shows a summary of the amplitude errors with different windowing functions with different FFT sizes. The lowest errors for each frequency are highlighted in green. If the error delta was less than 1.00%, both were highlighted. At lower frequencies the error of the Hanning window and rectangular window were close. At higher frequencies, the Hanning window error was significantly lower than the rectangular window. Real test data would have likely shown a larger spread of amplitude and frequency errors.

Table 2. Frequency Error of Various Frequencies and FFT Sizes

		No Window		Hanning Window (1/.49 Gain)		Rectangular Window	
FFT Size	True Frequency	Calculated	% Error	Calculated	% Error	Calculated	% Error
64	200	-	-	-	-	-	-
	450	400	11.11%	400	11.11%	400	11.11%
	1200	1200	0.00%	1200	0.00%	1200	0.00%
	2300	2400	4.35%	2400	4.35%	2400	4.35%
	3700	3800	2.70%	3800	2.70%	3800	2.70%
	5000	5000	0.00%	5000	0.00%	5000	0.00%
128	200	200	0.00%	200	0.00%	200	0.00%
	450	500	11.11%	500	11.11%	500	11.11%
	1200	1200	0.00%	1200	0.00%	1200	0.00%
	2300	2300	0.00%	2300	0.00%	2300	0.00%
	3700	3700	0.00%	3700	0.00%	3700	0.00%
	5000	5000	0.00%	5000	0.00%	5000	0.00%
256	200	200	0.00%	200	0.00%	200	0.00%
	450	450	0.00%	450	0.00%	450	0.00%
	1200	1200	0.00%	1200	0.00%	1200	0.00%
	2300	2300	0.00%	2300	0.00%	2300	0.00%
	3700	3700	0.00%	3700	0.00%	3700	0.00%
	5000	5000	0.00%	5000	0.00%	5000	0.00%

Table 3. Amplitude Error of Various Frequencies and FFT Sizes

FFT Size	True Amplitude	No Window		Hanning Window (1/.49 Gain)		Rectangular Window	
		Calculated	% Error	Calculated	% Error	Calculated	% Error
64	10 @ 200 Hz	-	-	-	-	-	-
	50 @ 450 Hz	45.11	-9.78%	44.27	-11.46%	45.11	-9.78%
	15 @ 1200 Hz	10.95	-27.00%	14.83	-1.13%	10.95	-27.00%
	30 @ 2300 Hz	24.91	-16.97%	28.39	-5.37%	24.91	-16.97%
	8 @ 3700 Hz	8.243	3.04%	7.759	-3.01%	8.243	3.04%
	25 @ 5000 Hz	17.45	-30.20%	23.02	-7.92%	17.45	-30.20%
128	10 @ 200 Hz	13.27	32.70%	10.28	2.80%	13.27	32.70%
	50 @ 450 Hz	32.56	-34.88%	44.22	-11.56%	32.56	-34.88%
	15 @ 1200 Hz	14.99	-0.07%	15.01	0.07%	14.99	-0.07%
	30 @ 2300 Hz	28.32	-5.60%	29.85	-0.50%	28.32	-5.60%
	8 @ 3700 Hz	6.568	-17.90%	7.6	-5.00%	6.568	-17.90%
	25 @ 5000 Hz	18.09	-27.64%	22.73	-9.08%	18.09	-27.64%
256	10 @ 200 Hz	10.1	1.00%	9.951	-0.49%	10.1	1.00%
	50 @ 450 Hz	49.65	-0.70%	50.64	1.28%	49.65	-0.70%
	15 @ 1200 Hz	14.76	-1.60%	15.2	1.33%	14.76	-1.60%
	30 @ 2300 Hz	28.05	-6.50%	29.67	-1.10%	28.05	-6.50%
	8 @ 3700 Hz	6.654	-16.83%	7.582	-5.23%	6.654	-16.83%
	25 @ 5000 Hz	18.56	-25.76%	22.9	-8.40%	18.56	-25.76%
512	10 @ 200 Hz	10.18	1.80%	10.16	1.60%	10.18	1.80%
	50 @ 450 Hz	49.95	-0.10%	50.94	1.88%	49.95	-0.10%
	15 @ 1200 Hz	14.81	-1.27%	15.18	1.20%	14.81	-1.27%
	30 @ 2300 Hz	28.31	-5.63%	29.85	-0.50%	28.31	-5.63%
	8 @ 3700 Hz	6.778	-15.28%	7.554	-5.58%	6.778	-15.28%
	25 @ 5000 Hz	18.69	-25.24%	22.89	-8.44%	18.69	-25.24%
1024	10 @ 200 Hz	10.07	0.70%	10.26	2.60%	10.07	0.70%
	50 @ 450 Hz	49.92	-0.16%	51.01	2.02%	49.92	-0.16%
	15 @ 1200 Hz	14.75	-1.67%	15.15	1.00%	14.75	-1.67%
	30 @ 2300 Hz	28.37	-5.43%	29.92	-0.27%	28.37	-5.43%
	8 @ 3700 Hz	6.787	-15.16%	7.526	-5.93%	6.787	-15.16%
	25 @ 5000 Hz	19.1	-23.60%	23.15	-7.40%	19.1	-23.60%
2048	10 @ 200 Hz	10.06	0.60%	10.25	2.50%	10.06	0.60%
	50 @ 450 Hz	49.86	-0.28%	50.93	1.86%	49.86	-0.28%
	15 @ 1200 Hz	14.69	-2.07%	15.15	1.00%	14.69	-2.07%
	30 @ 2300 Hz	28.38	-5.40%	29.99	-0.03%	28.38	-5.40%
	8 @ 3700 Hz	6.931	-13.36%	7.758	-3.03%	6.931	-13.36%
	25 @ 5000 Hz	19.07	-23.72%	23	-8.00%	19.07	-23.72%
4096	10 @ 200 Hz	10	0.00%	10.01	0.10%	10	0.00%
	50 @ 450 Hz	49.91	-0.18%	49.95	-0.10%	49.91	-0.18%
	15 @ 1200 Hz	14.76	-1.60%	14.89	-0.73%	14.76	-1.60%
	30 @ 2300 Hz	28.41	-5.30%	29.33	-2.23%	28.41	-5.30%
	8 @ 3700 Hz	6.979	-12.76%	7.616	-4.80%	6.979	-12.76%
	25 @ 5000 Hz	19.14	-23.44%	22.64	-9.44%	19.14	-23.44%
8192	10 @ 200 Hz	10.05	0.50%	10.23	2.30%	10.05	0.50%
	50 @ 450 Hz	49.91	-0.18%	51.01	2.02%	49.91	-0.18%
	15 @ 1200 Hz	14.76	-1.60%	15.18	1.20%	14.76	-1.60%
	30 @ 2300 Hz	28.45	-5.17%	30.01	0.03%	28.45	-5.17%
	8 @ 3700 Hz	6.95	-13.13%	7.735	-3.31%	6.95	-13.13%
	25 @ 5000 Hz	19.13	-23.48%	23.07	-7.72%	19.13	-23.48%

Figure 21 shows plots generated from the data in Table 3, showing how the amplitude error increases with increasing frequency at all FFT sizes. Above FFT sizes of 128, the error is relatively consistent at different FFT sizes. The error curves also flatten with increasing FFT size. **Note**: Only FFT sizes from 128 samples up to 4096 samples sizes are shown; higher FFT sizes yielded results very similar to those from the 4096 sample FFTs.

The Hanning window produced lower amplitude errors than the rectangular window for all cases. The Hanning window error increased with increasing frequency, and stayed under 10% at its worst case at 5000 Hz.

Order Processing

After the standard FFT processing, different MATLAB order processing scripts were executed for various input parameter settings to again gain an understanding of the effects on frequency domain data. As described in Chapter 4, the initial approach was to test different sample sizes that would be acquired at various simulated engine speeds. The MATLAB scripts were written so that the inputs were engine speed, sample rate, FFT size, and noise factor (which capped the amplitudes of the random noise). All parameters were varied to get an idea of the effects of changing the parameters.

Frequencies

In order FFT processing the frequency errors began to emerge with sample sets at and below 192.

Figure 21. Amplitude Error of Various Frequencies and FFT Sizes

Above this threshold the error was zero. If real test data were used there would likely be higher frequency errors for all sample sizes. It was realized that choosing round numbers as frequency values made the frequencies more likely to fall directly on lines of resolution in the spectrum. In hindsight frequency values should have been chosen that were not round numbers (e.g. 2307 Hz instead of 2300). This likely would have given a truer representation of frequency calculation errors. Also worth noting is the observation that at low simulated speeds the frequency error was the same or very close between the Hanning window, rectangular window, and no windowing. Table 4 shows the frequency error from order processing at various simulated engine speeds at a fixed FFT size of 1024. Error deltas are highlighted in gray.

Figure 22 shows frequency error as related to different simulated engine speeds. 750 RPM had zero error and is not shown. At simulated speeds at or above 4000 RPM, the frequency error between the window functions began to diverge. Besides losing the ability to discern the 200 Hz response altogether at higher speeds, the worst frequency error observed by the Hanning Window was 11.11% for the 450 Hz response at 17000 RPM. Likewise, the highest frequency error for the rectangular window was 18.75% for the 200 Hz response at 12000 RPM, which is just before this response becomes undiscernible.

Amplitudes

Table 5 shows the amplitude errors at various speeds (various sample sizes) processed with a fixed FFT size of 1024 samples.

Table 4. Frequency Error at Various Simulated Engine Speeds

Speed (RPM)	True Frequency, Hz	Hanning Window % Error (Absolute)	Rectangular Window % Error (Absolute)
750	200	0.00%	0.00%
	450	0.00%	0.00%
	1200	0.00%	0.00%
	2300	0.00%	0.00%
	3700	0.00%	0.00%
	5000	0.00%	0.00%
2000	200	0.00%	0.00%
	450	0.00%	0.00%
	1200	0.00%	0.00%
	2300	0.00%	0.00%
	3700	0.35%	0.35%
	5000	0.26%	0.26%
4000	200	0.00%	12.50%
	450	0.00%	0.00%
	1200	1.08%	1.08%
	2300	0.57%	0.57%
	3700	0.68%	0.68%
	5000	0.50%	0.50%
6000	200	0.00%	0.00%
	450	0.00%	0.00%
	1200	1.08%	0.00%
	2300	0.57%	0.57%
	3700	1.03%	1.03%
	5000	0.76%	0.76%
9000	200	-	12.50%
	450	0.00%	0.00%
	1200	1.08%	3.17%
	2300	1.65%	1.65%
	3700	1.70%	1.70%
	5000	1.50%	1.76%
12000	200	-	18.75%
	450	5.56%	0.00%
	1200	1.08%	5.25%
	2300	1.65%	1.09%
	3700	1.35%	2.03%
	5000	1.50%	1.76%
17000	200	-	-
	450	11.11%	0.00%
	1200	2.08%	4.17%
	2300	2.74%	2.17%
	3700	2.70%	4.41%
	5000	2.76%	3.00%

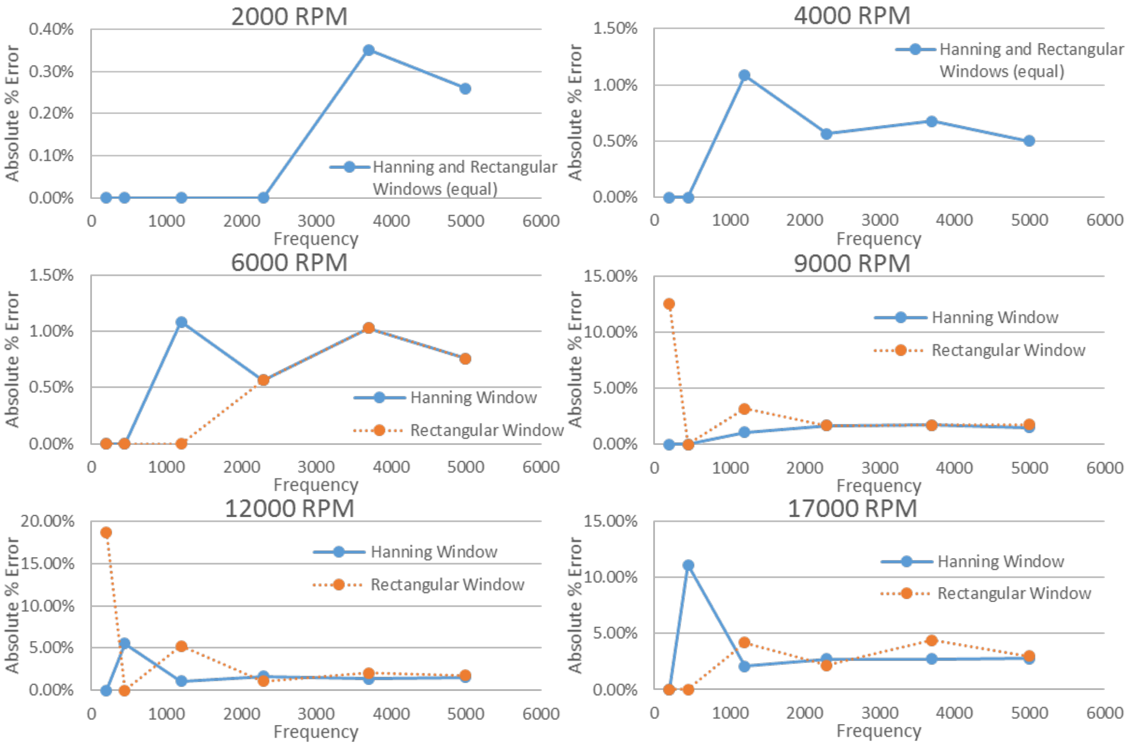

Figure 22. Frequency Error at Various Simulated Engine Speeds

Table 5. Amplitude Errors at Various Simulated Engine Speeds

Speed (RPM)	True Freq. (Hz)	Hanning Window % Error (Absolute)	Rectangular Window % Error (Absolute)
750	200	2.30	0.80
	450	1.82	0.16
	1200	1.73	1.33
	2300	0.57	2.30
	3700	4.33	14.30
	5000	7.60	23.60
2000	200	4.50	6.40
	450	1.82	0.20
	1200	2.07	0.13
	2300	0.87	5.93
	3700	2.10	1.09
	5000	1.88	0.28
4000	200	4.20	13.50
	450	0.86	0.96
	1200	0.33	4.47
	2300	1.00	2.00
	3700	0.70	2.15
	5000	1.68	0.12
6000	200	1.40	32.60
	450	0.94	0.66
	1200	1.53	1.13
	2300	0.10	2.37
	3700	5.80	2.23
	5000	1.76	0.72
9000	200	-	35.60
	450	1.50	2.64
	1200	0.33	0.87
	2300	1.43	2.83
	3700	0.56	9.40
	5000	0.56	0.48
12000	200	-	107.30
	450	4.10	1.54
	1200	1.87	15.87
	2300	1.17	3.37
	3700	3.80	3.99
	5000	1.36	2.00
17000	200	-	-
	450	9.42	9.28
	1200	7.13	1.53
	2300	0.90	4.60
	3700	2.50	9.04
	5000	1.56	1.44

The lowest error for each frequency is highlighted in green, and both are highlighted if the delta is less than or equal to 1.0%. Responses that could not be discerned in the frequency domain are marked with dashes. Figure 23 shows the amplitude errors at various speeds (various sample sizes) processed with a fixed FFT size of 1024 samples. Responses that were not discernable in the frequency domain were assigned values of 1000% error so that the error would be visible in the figures.

Based on the results shown above, at low simulated engine speeds (more samples) the amplitude error increases with frequency for both the Hanning and rectangular windows. The Hanning window yields more accurate amplitudes than the rectangular window at almost all speeds and frequencies. Another important observation is that responses at low frequencies get washed out in the frequency domain at high simulated speeds and cannot be discerned. In turbine engine strain gage applications, this would most likely not be an issue because integral responses at these high speeds will be at high frequencies. For example, if a 4 EO response was active at 17,000 RPM, the frequency would be 1133 Hz. One unexpected result was that the when the FFT size was equal to the sample size of 1024 at 750 RPM, the error was slightly elevated for responses at higher frequencies for both windowing functions. This speed was used for the purpose of comparison; turbine engines do not typically run at speeds this low. However, plant machinery could run at speeds even lower than this. Overall, the results from the cases above showed satisfactory results.

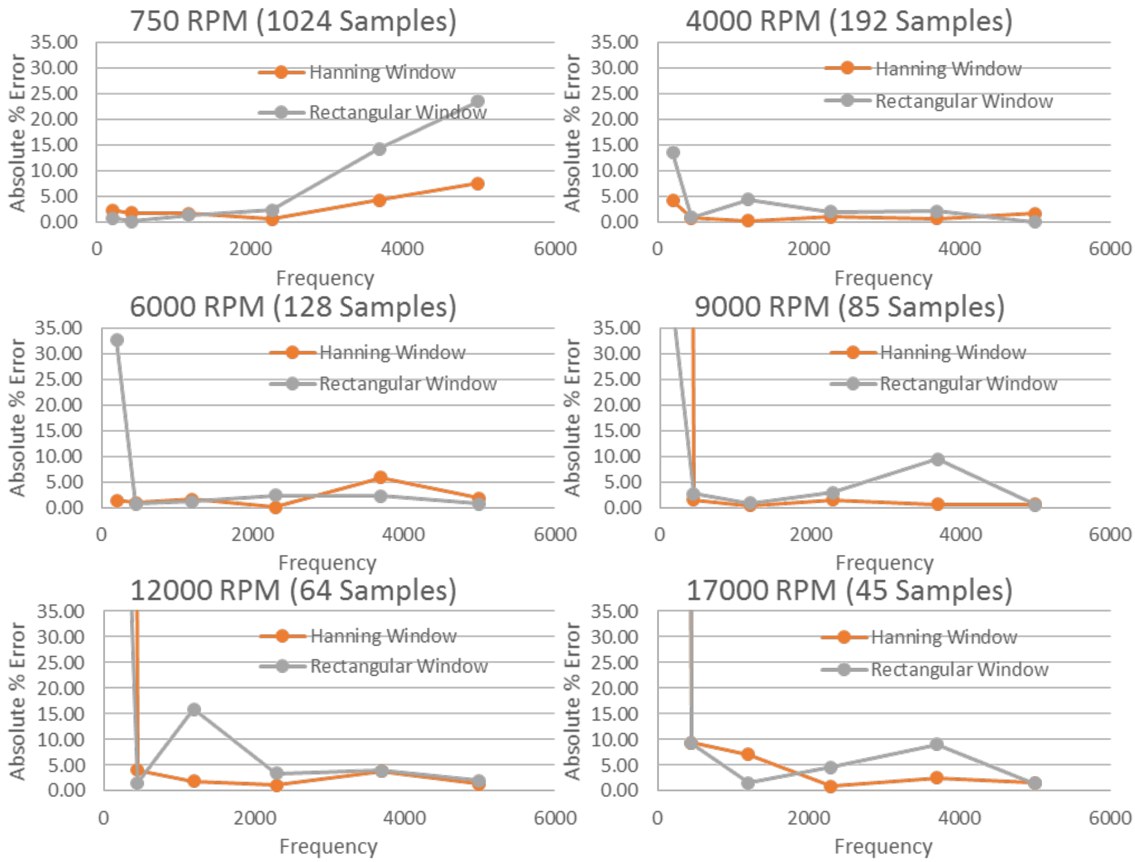

Figure 23. Amplitude Errors at Various Simulated Engine Speeds

Most error was kept under 10%, and usually 5% for the Hanning window. Also worth mentioning is the observation that the rectangular window seemed less stable at changing conditions. The Hanning window's curve is flatter and more predictable, which was also true for the results from standard FFT processing.

Zero Padding

The effect of padding was tested by varying FFT sizes (1024, 4096, 8192, and 16384) for a fixed simulated engine speeds of 6000 RPM. For sample sizes less than the specified FFT size, at any given speed and fixed sample rate, as FFT size increased the amount of padding also increased. With the exception of the low frequency 200 Hz response, the error curves of each frequency stayed relatively flat with increasing FFT size for both Hanning rectangular windows and the error was contained in a region less than approximately 1% for all FFT sizes at 6000 RPM. In reality an FFT size of 16384 samples would rarely be used. The inclusion of this FFT size was for the purpose of extending the curves for trending. Figure 24 shows the frequency error associated with zero padding at 6000 RPM with varying FFT sizes.

The amplitude accuracy stayed nominally the same (relatively flat curves) with increasing FFT size. This is because the FFT scaling and windowing are really only applied to the sampled data. The 200 Hz response error curve stayed flat, though it maintained a relatively large value of roughly 33%. Eventually the 200 Hz response becomes undiscernible in the frequency domain with increasing speed.

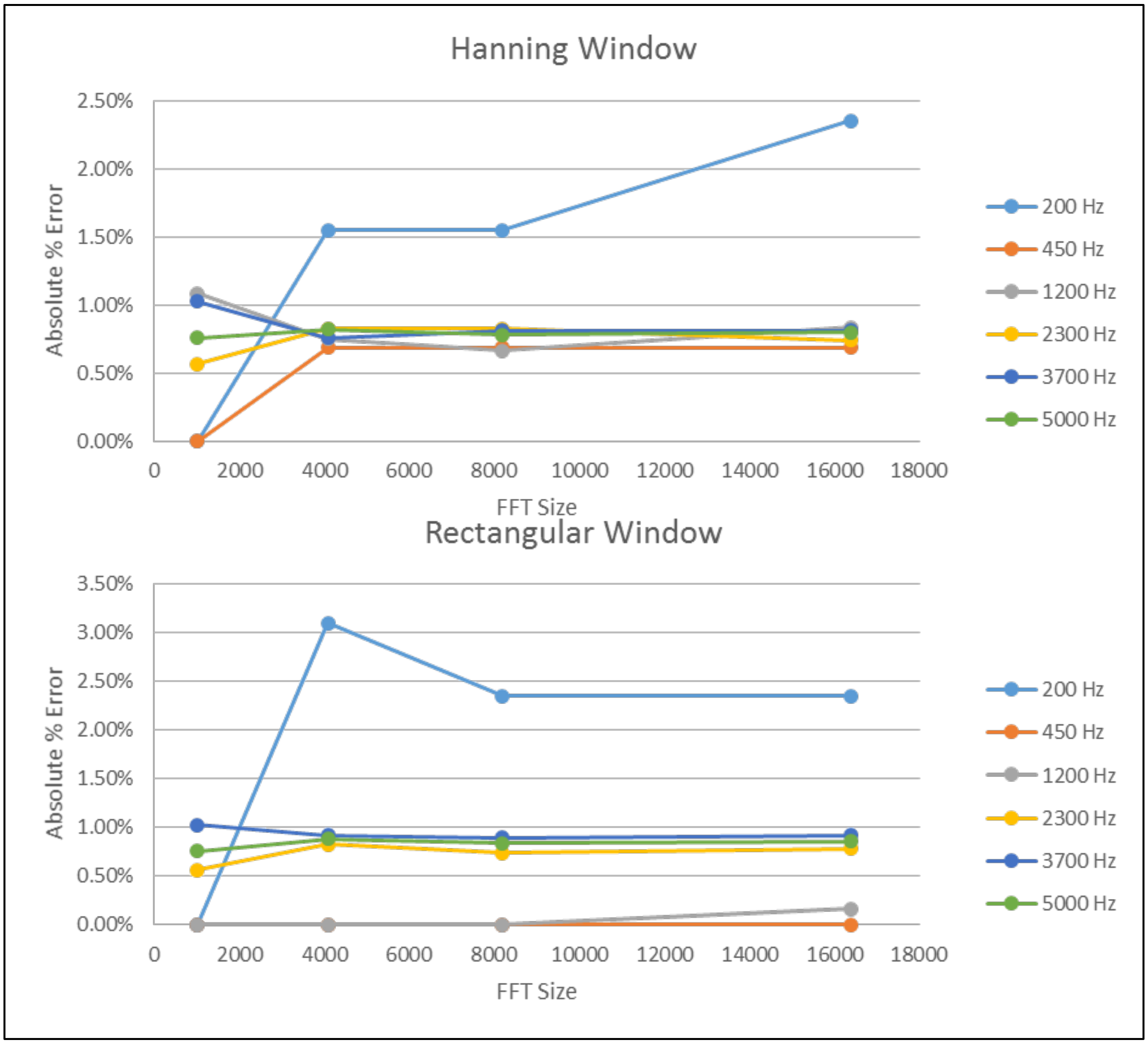

Figure 24. Frequency Error Associated with Zero Padding

As previously discussed, low frequency responses seem to be an area of limitation for this method of order processing, as is also evident in the frequency error. Figure 25 shows the amplitude error associated with padding at 6000 RPM with varying simulated engine speeds. The rectangular window was slightly less sensitive to changing FFT sizes.

Decimation

After learning that excessively high sample rates degrade frequency resolution and that high FFT sizes improve the frequency resolution, it was decided that if this was ever implemented in an acquisition system there would rarely, if ever, be a need for decimation. With that said, if very high sample rates are used then decimation might be required. When decimation is employed, the effective sample rate is simply divided by the decimation factor. There must be cognizance of the Nyquist sampling rate if ever decimation is required. In a real application there would need to be a flag to alert users if decimation resulted in effective sampling rates that violate the sampling theorem.

Figure 25. Amplitude Error Associated with Zero Padding

CHAPTER SIX

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

Summary

Standard FFT Processing

The relationships between frequency resolution, sample rate, and FFT size are critical. Sample rates should be carefully chosen for each particular application. Oversampling was initially thought not to be an issue. Excessive sampling rates cause poor frequency resolution.

Within the environment of MATLAB, the frequency error from standard FFT processing was zero for realistic FFT sizes; this would likely not be the case with real test data. This likely stems from choosing round numbers for frequencies. Still, the effect of bin width on frequency accuracy was evident in the results with smaller FFT sizes.

The amplitude accuracy was shown to be relatively stable when FFT sizes were 256 or larger; FFT sizes larger than 256 are usually used in real applications. FFT sizes larger than 256 produced results that were almost asymptotic with increasing FFT sizes at each frequency of interest. Typical FFT sizes for vibratory applications in turbine engines are usually 512, 1024, and 2048. At all FFT sizes the amplitude accuracy was higher when employing the Hanning window than when employing the rectangular window. The Hanning window showed increased error with increasing frequency, and was capped at just under 10% at the 5000 Hz response.

Order Processing

At low speeds the Hanning and rectangular windows yielded similar frequency accuracies. The frequency accuracies of the Hanning and rectangular windows were the same at and below sample sizes of 192 (4000 RPM). When the sample size was less than 192 (4000 RPM), the Hanning and rectangular window's frequency accuracies diverged at various frequencies, but still performed similarly. While frequency error did increase with simulated engine speed, the frequency error stayed under 5% for both the Hanning and rectangular windowed data sets (excluding low frequency responses).

For almost all conditions, the amplitude error was less using the Hanning window than when using the rectangular window. The amplitude error from the Hanning window stayed below 5% at all speeds except the very low speed of 750 RPM.

Zero padding had almost no effect on frequency accuracy for different FFT sizes executed at a fixed simulated engine speed. With the exception of the aforementioned low frequency responses, zero padding had a very small effect on amplitude accuracies; zero padding only changed the amplitude accuracies by 1 to 3 % within the range of realistic FFT sizes. The rectangular window was slightly less sensitive to changing FFT sizes.

Decimation, or down sampling, was not directly employed in the analyses, simply because the effect of decimation is common knowledge. In real life applications initial calculations should be made to determine if decimation would be required in the speed range. Then the effective sample rate must be checked to avoid violating the Nyquist sampling theorem when decimating sample sets.

Conclusion

The goals of this endeavor were to research order processing at a fundamental level and quantify the effects on frequency and amplitude accuracies. It was known beforehand that some level of error would be introduced by this method, but surprisingly, frequency and amplitude error were kept below 5% in most cases, which is on par or better than standard FFT processing, which maxed out at just below 10% error when applying the Hanning window. Given the minimal effects of zero padding, large FFTs can be used in lieu of decimation. Considering the effect of averaging over multiple revolutions in standard FFT processing methods and the low error, this method of order processing may very well yield more accurate results than standard FFT processing methods for momentary integral responses and integral responses in general.

After conducting this research, the results are in favor of further development and implementation of order processing for use in turbine engine test facilities. From what was learned during this research endeavor, order processing capabilities are feasible and will bring long-term, valuable benefits to dynamic data acquisition and analysis. Order processing will allow for additional analysis capabilities in the time domain, will minimize amplitude losses due to averaging over multiple revolutions, and will facilitate comparisons between data of various types of instrumentation. While order processing capabilities are not meant to replace standard FFT processing methods, order processing will bring about a different and parallel perspective of dynamic data, allowing for a fuller understanding of vibratory responses in turbine engines and other rotating machinery.

Recommendations

It is recommended that further development of order processing capabilities for both real-time and post-analysis applications for the acquisition system that is currently being developed be considered. The ability to apply various windows should be made available for order processing in future acquisition systems. The ability to run separate larger FFTs should also be a consideration.

Test data was not included in this paper. It is recommended that, as schedule and budget permit, that these capabilities be tested with real test data that has already been acquired.

It is also recommended that Computed Order Tracking (COT) abilities (not covered in this paper) be explored. COT algorithms should be tested for real-time data processing applications. Upsampling and parabolic interpolation should also be explored for order processing. In addition to the capabilities discussed herein, it is recommended that additional research be conducted to extend these capabilities for azimuth applications. This could be particularly helpful in helicopter testing applications.

The next step is to begin implementing these algorithms into a current acquisition system. There will likely be other challenges that emerge that are related to programming, data storage, and other acquisition aspects, but through collaboration of analysis engineers, signal processing engineers, instrumentation engineers, and programmers, this capability can be implemented.

LIST OF REFERENCES

[1] Nichol, Kurt. *Aeromechanics Training Course*. Aeromechanical Theory. APEX Turbine Technologies User Conference 2016, 22 Feb. 2016, New Orleans, LA, Omni Royal Orleans Hotel.

[2] Emerson, Training Team. *Basic Vibration Analysis*. Emerson Process Management Educational Services, 2014.

[3] Zion, Christopher. *The Fourier Transform and Some Applications* (2002). Masters Theses & Specialist Projects. Paper 612. http://digitalcommons.wku.edu/theses/612

[4] Bendat, Julius S., and Allan G. Piersol. *Random Data: Analysis and Measurement Procedures*. Wiley, 2010.

[5] Emerson, Training Team. *Intermediate Vibration Analysis*. Emerson Process Management Educational Services, 2014.

[6] *Understanding FFTs and Windowing*. National Instruments. www.ni.com/white-paper/4844/en/.

[7] Hayes, Bryan Will, Design, *Implementation, and Analysis of a Time of Arrival Measurement System for Rotating Machinery*. Master's Thesis, University of Tennessee, 2012. http://trace.tennessee.edu/utk_gradthes/1159

[8] Chaudhari, Qasim. *Understanding Sample Rate Conversion and Scaling Factors.* EDN Network, 10 Dec. 2016. edn.com/design/analog/4443139/ Understanding-sample-rate-conversion-and-scaling-factors.

[9] Rowell, Derek. *Signal Processing - Continuous and Discrete.* MIT OpenCourseWare. 13 Oct. 2017, Massachusetts Institute of Technology: MIT, Massachusetts Institute of Technology: MIT. https://ocw.mit.edu/courses/mechanical-engineering/2-161-signal-processing-continuous-and-discrete-fall-2008/lecture-notes/lecture_11.pdf.

[10] Vining, Charles R., Hayes, Bryan W., Arnold, Steve A., & Howard, Robert P. *Comparison of Non-contact Stress Measurement Data to Strain Gage Data, High Cycle Fatigue Conference*. New Orleans, LA. March 8-11, 2005.

[11] Janicki, G, et al. *Turbine Blade Vibration Measurement Methods for Turbocharges*. American Journal of Sensor Technology 2.2 (2014): 13-19

[12] *Digital Signal Analysis Review: Part 3 - Aliasing.* Noise and Vibration Test Blog: SignalNews from Data Physics Corporation, Data Physics, 7 Jan. 2016, blog.dataphysics.com/dynamic-signal-analysis-review-aliasing/.

VITA

Nathan D. Harrison was born and raised in Shreveport, LA. He worked in aviation as an aircraft mechanic before returning to school to pursue a career in engineering. In 2013 he graduated cum laude from the University of New Orleans with a Bachelor of Science degree in Mechanical Engineering. He immediately accepted a position as a mechanical engineer working in the Oil and Gas industry in New Orleans. Shortly after he returned to his aviation interests and accepted an analysis engineer position at AEDC testing commercial and military turbine engines. Nathan is now the lead aeromechanical analysis engineer for turbine engines testing at AEDC. He has been pursuing a Master of Science degree in Mechanical Engineering since the fall semester of 2015 and will be graduating in May of 2018 from the University of Tennessee.