SYSTEM IDENTIFICATION OF AN ON ORBIT SPACECRAFT'S ANTENNA DYNAMICS

THESIS

Christopher M. Sylvester, Lieutenant Commander, USN

AFIT/GA/ENG/09-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GA/ENG/09-01

SYSTEM IDENTIFICATION OF AN ON ORBIT SPACECRAFT'S ANTENNA DYNAMICS

THESIS

Presented to the Faculty

The Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Christopher M. Sylvester, BS

Lieutenant Commander, USN

June 2009

AFIT/GA/ENG/09-01

SYSTEM IDENTIFICATION OF AN ON ORBIT SPACECRAFT'S ANTENNA DYNAMICS

Christopher M. Sylvester, BS

Lieutenant Commander, USN

Approved:

Dr. Meir Pachter (Chairman)	1 June 09 Date
Dr. Howard Evans (Member)	1 JUN 09 Date
Eric Swenson, LtCol., USAF (Member)	01 JUN 09 Date
Dr. Jonathan Black (Member)	01 JUN 09 Date

Abstract

As a result of previous efforts [Pachter, Barba, 2007] a tight control loop was designed to meet performance specifications while minimizing the feedback control system's gains of a spacecraft mounted flexible antenna. Emphasis is now shifted to on orbit system identification of the antenna dynamics in order to increase nominal plant knowledge, estimate plant uncertainty bounds, as well as determine the disturbance band. Non-parametric system identification is undertaken.

Knowledge of the plant dynamics along with the corresponding uncertainty bounds will provide for the design of a control system which meets the specifications (tracking and disturbance rejection) while at the same time employing the lowest possible gain. This in turn is conducive to sensor noise disturbance rejection, avoidance of actuator saturation, and excitation of high frequency modes.

Acknowledgments

First and foremost, I give thanks to God for the lessons with which I have been graced at AFIT and at home while stationed in Ohio. For keeping me firmly grounded and reminding me of what is truly important, I would like to thank my wife who, while raising the bar for every wife and mother, simultaneously earned her Master's in Nursing with a 4.0 GPA. My children constantly begged for just one more story at bedtime which will continue to serve as a beacon to guide my efforts in raising them in His Glory. My parents deserve my eternal gratitude for setting me on this path of success. As for my dear friends in Texas and California, there is no way I would have accomplished as much as I have without their willingness to answer my late night, early morning, and midday phone calls in attempt to preserve my sanity. On that note, I need to also thank the cell phone provider for unlimited minutes.

For his continued patience and guidance, I would like to offer my sincere appreciation to my thesis advisor, Dr. Meir Pachter. And to the other members of my thesis committee: Lt Col Eric Swenson for his willingness and ability to repeat the answer to my repeated questions in an infinite number of ways; Dr. Howard Evans for his subtle and sincere mentorship; and Dr. Jonathan Black not only for the time spent reviewing this paper, but for the time invested in creating impossible exams.

Christopher M. Sylvester

Table of Contents

Page

List of Figures

Page

List of Tables

Page

SYSTEM IDENTIFICATION OF AN ON ORBIT SPACECRAFT'S ANTENNA DYNAMICS

I. Introduction

Problem Statement

The underlying reason for the identification of an "unknown" system is to allow for a more precise control application for mission accomplishment. In this discussion, it is understood that there exists a satellite antenna that has been thoroughly modeled in a controlled environment. As a result of such, there exists a set of characteristic equations that define how an input will be received and what output will be delivered in terms of slewing the onboard antenna. The positional response of the antenna is controlled using measurements provided by feedback action and rate gyros and possible accelerometers. Ideally, a desired command is given to the plant (antenna) which results in an exact and immediate positioning of the antenna array. Realistically this is not possible for a host of reasons. From the time the command is initiated there are forces to overcome including internal vibrations from a variety of onboard sources (motors, pumps); external torques from observable micrometeoroids to unmodelled forces such as atmospheric drag and solar winds; and internal torques that might occur as a result of station keeping or required maneuvering.

In a logical step-wise approach, it is clear to see what is required to get from a current or actual position (y) to a desired position hereafter known as the *reference position* (r). First we must define where we are based on a known reference. And with respect to the same reference frame, we need to determine where to go. The goal is to ensure the error between the reference (r) and output (y) is zero. Had the antenna

remained in the initial controlled environment, it could be expected that the derived model would provide for the desired output. With a thorough knowledge of how the plant will map an input to an output, the command may be tailored to ensure the desired response. However, the rigors of space launch are immense let alone the subsequent required maneuvers and harsh space environment. From the time that the man-made satellite leaves the controlled environment for preparation there are a multitude of jarring forces incurred by transportation, loading within the fairing and those induced by the launch itself. Ironically, in a previous experiment detailed by B. Cooper [2] an imbedded accelerometer measured forces as great as 22 g's coming not from the launch or post-launch maneuvers, but from the shipping to the launch site! Such impacts can instantly and unsuspectingly forever change the painstakingly derived model used for control design. As a result, the derived nominal plant model is no longer valid. Once on orbit, the physical mass of the satellite will alter as fuel is expended for maneuvering and station keeping, as well as the expansion and contraction of the spacecraft components due to constant temperature cycles throughout its lifetime of successive orbits. All in all, the reliance upon a known model is only valid as long as the model itself remains unchanged. But as presented, the model changes over time and the expected input-output mapping function must be updated to compensate for the altered characteristic equation(s) and various forces and torques (internal and external) to ensure a zero-mean error (e) between (y) and (r). Therefore, there is an understood level of required compensation. And the challenge is then presented as to where to place that compensation. Detailed in the diagram, consideration here is given to providing the compensation not through high gain feedback, but rather using the insight gained through

system on orbit identification so that feedback action with lower gains would give satisfactory performance; in addition, one is better able to rely on knowledge of how the previously modeled dynamics have changed and compensate for net forces via a feed-*forward* action that more accurately anticipates the output, thus modifying the initial command in anticipation of an exact response. While a feedback loop is left in place, its anticipated use is to provide for increased robustness and an additional means of tracking error rejection to ensure precise antenna pointing. This is different than saying that the feedback loop is the only source of error correction; rather pointing error is compensated for with the positioning command and feedback action. Feedback action is available for unexpected disturbance rejection. Moreover, allowing for an input that more closely matches the expected response, an increased overall efficiency is obtained allowing for less power consumption and increased satellite life.

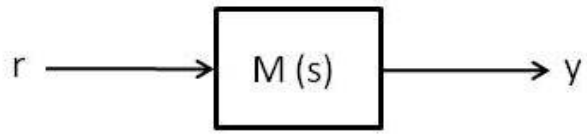


Figure 1. Open-Loop Control System

The transfer function computes the system's dynamics:

$$\frac{y(s)}{r(s)} = M(s) = \frac{\omega_n{}^2}{s^2 + 2\xi\omega_n s + \omega_n{}^2} \tag{1}$$

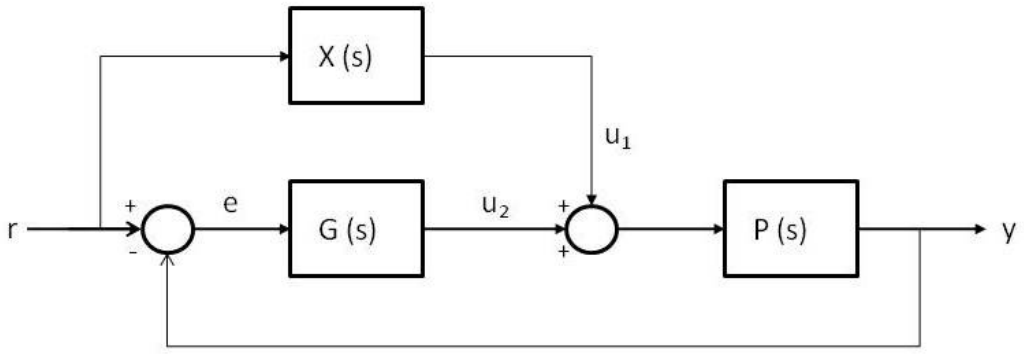


Figure 2. Control System Using Feedback Action

Figures 1 and 2 depict the same plant with different control schemes. The former is a depiction of open-loop control when the plant's dynamics are perfectly known. The latter shows a control implementation where one relies on feedback action to address plant uncertainty. Assuming that the moment of inertia of the antenna (J) remains unaltered, the rigid body mode of the plant dynamics can remain as P(s), the nominal plant's transfer function, while the unknown dynamics are accounted for in X(s), which is to be the result of SysID. Additionally, in Figure 2 a feedback loop is included. Until X(s) is characterized, the only possible way to minimize the tracking error (e) is to measure the error that exists. Without complete knowledge of the system's dynamics one cannot fully predict how an input maps to an output, and attempts to control are significantly hindered.

The following relationships of the various system elements are provided:

$$P(s) = \frac{1}{Js^2} \tag{2}$$

$$e(s) = r(s) - y(s) \tag{3}$$

$$u_1 = Xr \tag{4}$$

$$u_2 = Ge \tag{5}$$

$$y = P(u_1 + u_2) \tag{6}$$

Furthermore,

$$\frac{y(s)}{r(s)} = P\frac{G + X}{1 + PG} \tag{7}$$

As mentioned previously, the intent is to rely heavily on an accurately mapped input signal we want to use for feed forward action. Therefore, it is proposed that

$$e \equiv 0 \Leftrightarrow u_2 \equiv 0$$

So, as a result, we are left with

$$u_1 = X(s)r \tag{8}$$

$$y(s) = Pu_1 + \cancel{Pu_2} \tag{9}$$

and the following relationships are obtained

$$\frac{y(s)}{r(s)} = M(s) = P(s)X(s) \tag{10}$$

$$X(s) = \frac{M(s)}{P(s)} \tag{11}$$

Rewriting X(s) in standard form yields:

$$X(s) = \frac{\omega_n{}^2}{s^2 + 2\xi\omega_n s + \omega_n{}^2} \tag{12}$$

Which in turn can be written as

$$X(s) = \frac{\omega_n{}^2}{s^2}[1 - 2\xi\omega_n \frac{s + \frac{\omega_n}{2\xi}}{s^2 + 2\xi\omega_n s + \omega_n{}^2}] \tag{13}$$

And, after defining

$$G_1(s) \triangleq 2\xi\omega_n \frac{s + \frac{\omega_n}{2\xi}}{s^2 + 2\xi\omega_n s + \omega_n{}^2} \tag{14}$$

the entire system can be reinterpreted via Figure 3, allowing for greater insight into the individual contributors and inclusion of a feed-forward loop.

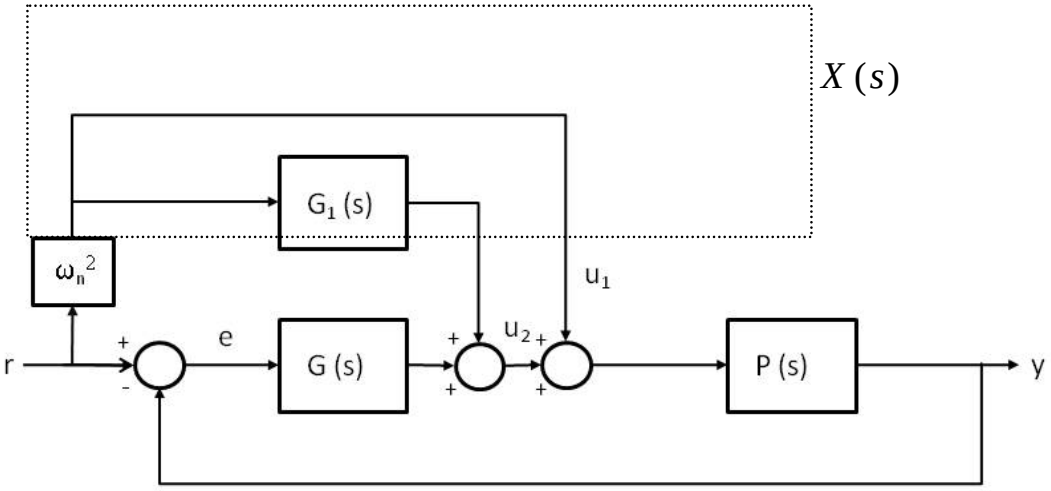


Figure 3. Inclusion of Feed-Forward Loop

Notice that X(s) is now a combination of a feed-forward element where the signal itself is introduced to the plant unaltered along with a portion of the signal that is acted upon by the function $G_1(s)$ TBD via SYS ID. Figures 2 and 3 illustrate P(s) as a portion of the nominal plant model. Within P(s), the rigid body mode is included J, which is characterized by the antenna's moment of inertia. P(s) is depicted (as per the sponsor provided model) in parallel with 20 known antenna flexible modes in Figure 4.

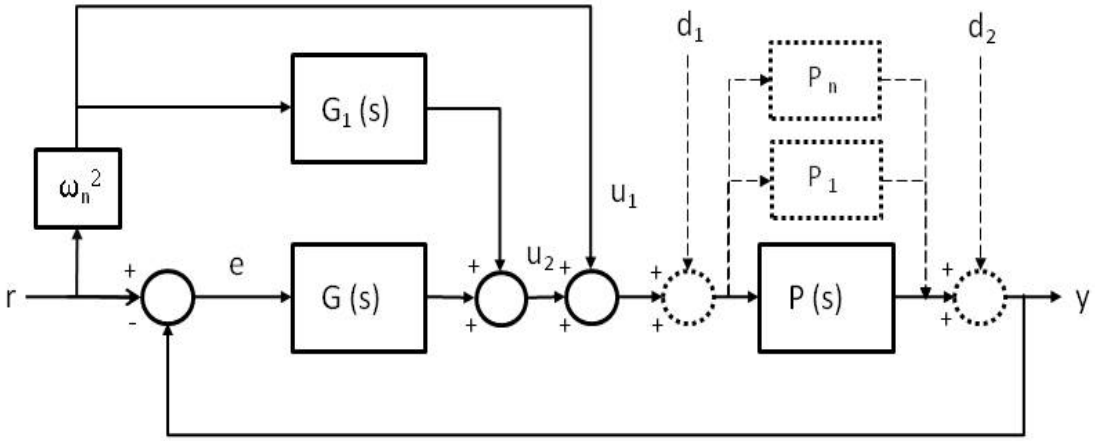


Figure 4. Control System with Feed-Forward and Feedback Action

Detailed in Figure 4 is the inclusion of two possible disturbances, d_1 and d_2. These disturbances will include transient disturbances that may occur due to satellite multi-tasking (station keeping, maneuvering) and perhaps the effects of the harsh space environment. These are not consistent enough to plan for but are significant enough to require some level of compensation. The dashed connections and boxes denote the 20 modes of the flexible antenna structure, all of which are in parallel. The information for the modes (gain, natural frequency, damping ratio) is provided in Chapter 3 and the Appendix.

Research Focus: Assumptions and Limitations

The scope of this research is limited to identifying the response of the system via data provided from an onboard rate-sensing gyro. Placed at the most outward tip of the antenna (allowing for greatest magnification of small excursions), this data will serve as the primary motion indicator for the satellite for a terrestrial-based system identification effort, and thusly serving as the only means for which antenna response to an input

command is measured and subsequently evaluated. This underlying assumption firmly defines the observation of the antenna response as described by the rate-sensing gyros with no intent to identify the individual response of various subcomponents. In sum, the only unknown is the on-station system transfer function. Both the uploaded input command signal intended for slewing of the antenna and the resulting motion of the satellite as relayed from the onboard gyros are known.

> *Sidebar: The angular rate gyros are capable of measuring angular motion in 3 dimensions. The dimensions of a commercially available gyro may be as small as 23mm x 23mm x23mm. Given the below table, it is obvious that the specifications of the device easily allow it to overcome the rigors of launch and orbit insertion, the environment of space, and provide measurements which are not expected to exceed a bandwidth of 152Hz and an angular rate of no more than .2 rad/s, or approximately 6 deg/sec.*

Parameter	***Min***	***Typical***	***Max***
3 dB Bandwidth		330Hz	
Dynamic Range (°/sec)	±300°	±350°	
Acceleration, any axis			2000 *g*
Operating Temperature Range	−40°C		+105°C

Figure 5. ADIS16360/ADIS16365 iSensor, Six Degrees of Freedom Inertial Sensor [3]

The groundwork for this documentation was laid by the efforts of Barba in his thesis titled *Controller Design for Accurate Antenna Pointing Onboard a Spacecraft* [1]. The subject matter was in support of minimizing the effect of onboard disturbances on antenna pointing onboard a spacecraft/satellite, allowing for an increase in pointing accuracy to an accuracy of ±5 micro radians (μrad/s). The overarching goal of the

research effort remains steadfast to the previous work, however now designing a system ID method for post-launch plant modelling in an effort to identify the frequency response of the multi-modal antenna control system. The end product of this research is a MATLAB algorithm that will produce a Bode plot of system/antenna response, depicting magnitude and phase shift.

Methodology

The approach explored in this paper, at first, relies heavily on "known" parameters to provide for the evolution of a working model, and then obtain confidence in said model. With both input and outputs known, and the associated transfer function recoverable, random white noise will be introduced and consistently increased to the point of failure of the system ID algorithm, or rather, to the level from which the knowledge of the input and output signals alone are not sufficient and the system's transfer function/dynamics is/are no longer recoverable.

Preview

The organization of the thesis is designed to seamlessly walk the reader through the thoughts and processes of the writer. Chapter 2 reviews what information is available for the various elements of signal analysis and system identification. Chapter 3 provides insight as to how the frequency response was analyzed throughout various simulation runs using MATLAB and included various MATLAB toolboxes. The results of the simulations are provided and discussed in greater detail in Chapter 4 and include a comparison and contrast of the methods used for analysis. Chapter 5 provides concluding remarks with suggestions for continued work.

II. Background Information

Literature Review

The precise topic of "Spacecraft System Identification" does not readily return as vast search results as "unspecified" system ID. However, there are sufficient topics addressed to offer some discussion.

In an attempt to estimate the attitude, velocity, and bearing of a spacecraft, VanDyke, Schwatz and Hall focused on using an Unscented Kalman Filter [4]. The Kalman filter is a Linear Quadratic Estimator, the emphasis being placed on "Linear" and that excludes many practical control systems, which are inherently nonlinear. While there exists an Extended Kalman Filter (EKF) which is considered to specifically address non-linearities, in the end it relies on linearization in order to provide a solution as to the current state. The Unscented Kalman Filter (UKF) utilizes more statistical data in order to formulate a more accurate representation of the current state estimate. As a whole, Kalman filtering is ideal for working with non-stationary signals, which accurately describes the result of measuring the antenna response to a command for rotation. Because there is a mapping function, a.k.a. transfer function, that relates the input to an end state, the response data is dependent upon the time at which it is observed [5]. As a tool for SYS ID, the end product offered by a robust Unscented Kalman Filter can serve to identify where the system was versus where the system is. Correlating the change over time with the input over time might lead to the development of a relationship between the output and input. As detailed in a separate report by Julier and Uhlmann [6], the implementation of the UKF, while yielding accurate results for the distance, velocity, and

bearing of a spacecraft, the samples were limited to 10Hz. This precludes further incorporation of the UKF in this discussion as the bandwidth of interest is 152Hz.

Published in 2003, a NASA Ames Research team put forth a method for SYS ID that focuses on mass and thruster identification [7]. Their process digs much deeper than the simulations discussed later, as it looks to quantify the following properties of an on-station spacecraft: center of mass, inertia matrix, inverse inertia matrix and the corresponding thrust generated by each thruster. All of this information is the foundation of a model that would map a command to a response. Using the data from onboard rate sensors and MATLAB based algorithms, the information detailing the motion, angular acceleration, and thruster data is obtained and without any additional equipment and requiring only the software.

An ugly issue that must be faced head on deals with the presence of sensor noise and disturbances, e.g. vibrations. Regardless of how exact the actual derived model may be in relation to the physical characteristics of the orbiting spacecraft, there will always be the presence of noise. This data is invariably corrupted with noise. What is received back on Earth must be corrected for noise. Schoukens and Pintelon suggest using the geometric mean in lieu of the Power Spectral Density and Cross Power Spectral Density analysis [8]. They illustrate that in a noise free environment, the ratio of the CrossPSD to the AutoPSD equals that of the AutoPSD to the CrossPSD. Therefore at low signal-to-noise ratio, there could exist problems in the resulting solution which is overcome only if there is no noise in the input. The paper continues to detail the advantages of using H_{geom} and H_{arith} for S/N$\geq$3dB and provides the following:

	Input Noise	Output Noise
$\mathbf{H_1}$	Not allowed	Free
$\mathbf{H_2}$	Free	Not allowed
$\mathbf{H_{arith}}$	Limited	Free
$\mathbf{H_{geom}}$	Limited	Limited

Figure 6. Summary of Geometric Mean Options [8]

Where H, H_1 and H_2 are defined:

$$H(f) = \frac{Y(f)}{X(f)} \tag{15}$$

$$H_1(f) = \frac{G_{xy}(f)}{G_{xx}(f)} \tag{16}$$

$$H_2(f) = \frac{G_{yy}(f)}{G_{yx}(f)} \tag{17}$$

Where $X(f)$ and $Y(f)$ are the Fast Fourier Transforms of the noise free input and output respectively while G_{xx}, G_{xy}, G_{yy} and G_{yx} are auto and cross correlations of $X(f)$ and $Y(f)$. Also,

$$H_{arith} = \frac{1}{n}\sum_{i=1}^{n} H_{mi} \tag{18}$$

$$H_{geom}(f) = \prod_{i=1}^{n} \sqrt[n]{H_{mi}(f)} \tag{19}$$

where $H_m(f)$ is the measured values of $H(f)$ with the addition of noise

$$H_m(f) = \frac{Y(f) + N_{output-noise}(f)}{X(f) + M_{input-noise}(f)} = \frac{Y_m(f)}{X_m(f)} \tag{20}$$

A final manipulation serves to provide simplified insight into the results of a low Signal-to-Noise Ratio with high input noise content.

$$H_m(f) = H(f)\frac{1 + \dfrac{N_{output-noise}(f)}{Y(f)}}{1 + \dfrac{M_{input-noise}(f)}{X(f)}} \tag{21}$$

Another method entails an algorithm based on Markov chain Monte Carlo methods which allows for a nonparametric approach to identifying subsystems in the presence of noise [9]. Published by a team from UC, Berkeley, the approach deals well with outliers, has proven feasible with nonlinear dynamics and is remarkably accurate in the linear domain.

Brown [10] focuses his research on introducing the ideal input signal. After reviewing briefly the tenants of communication theory, he highlights the importance of capturing as much “useful” data as possible. The utility of the signal is measured in terms of how much information is transmitted from the plant to the receiver which is attempting to identify the system, and that is dependent on the response of the system. In the end, the object is to elicit the greatest response via a command signal that excites the various modes of the plant. Brown strives for maximum "excitation" of the unknown

plant via a frequency-rich input. Detailing the communication theory as per its originator, Claude Shannon, Brown denotes a dichotomy of what seems too obviously simple on one hand and yet very logical and possibly complicated on the other hand.

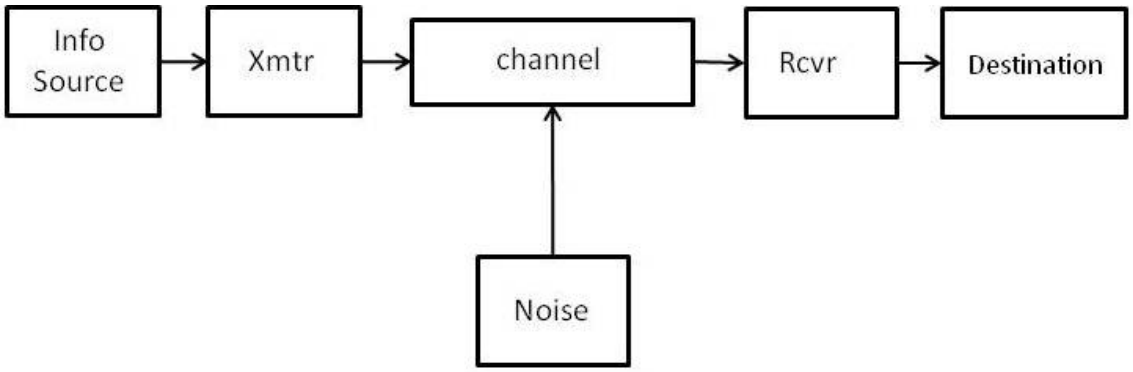


Figure 7. Communication Theory as per Shannon [11]

While the logic is undisputable, the simplicity however is challenged when the same theory is molded to fit our antenna and SYS ID process.

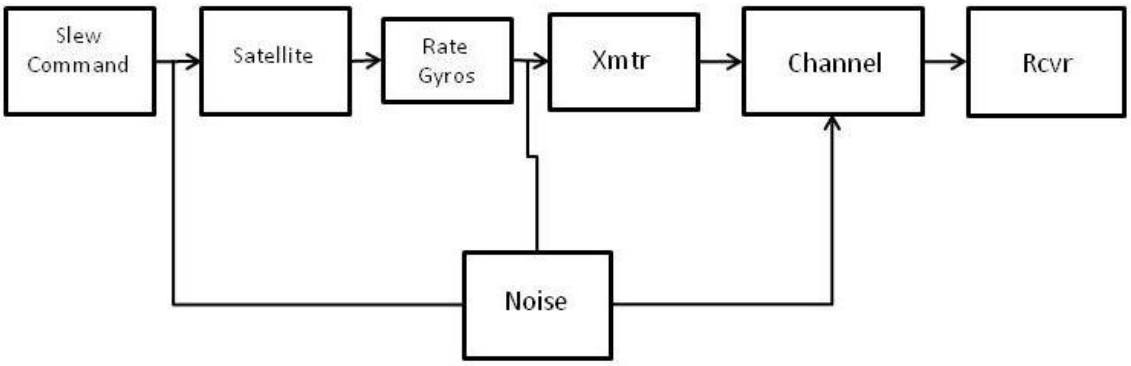


Figure 8. Communication Theory for SatComm Scenario

Moreover, focusing strictly on the power of noise corruption, the digital signal upon which we rely, itself, depends on a stream of varying “1”s and “0”s, where a flip can be the difference between “True” and “False”.

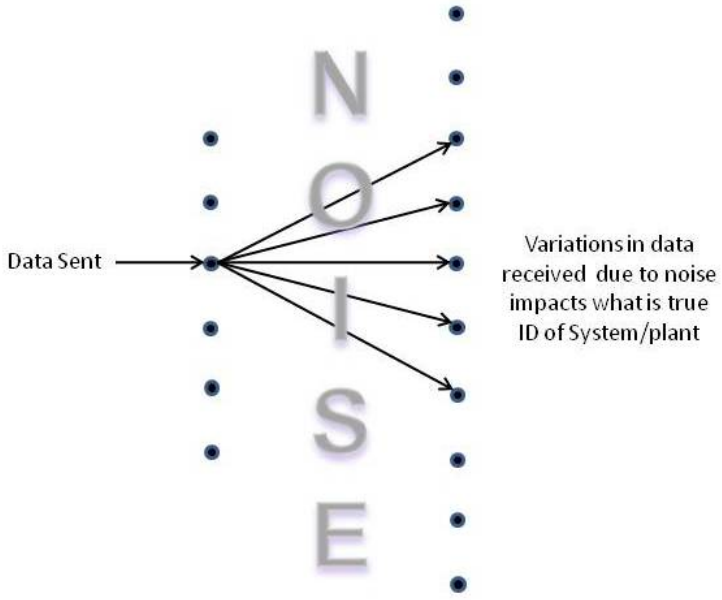


Figure 9. Data Affected by the Presence of Noise [12]

Brown’s focused input initiative must be tempered with a firm set of boundaries in order not to overexcite the plant. As Barba’s [1] thesis points out, this can lead to actuator saturation. As a precursor to this paper, *Controller Design for Accurate Antenna Pointing Onboard a Spacecraft* [1] details the necessity of awareness when operating around the first mode, which typically responds with the max gain of the various frequency responses. It will be shown later that the frequency for the first mode of the theoretical spacecraft is 3Hz .

Frequency Response

This thesis is centered on the concept of "nonparametric" system ID. As defined by Merriam-Webster, it is that which is "not involving the estimation of parameters of a statistical function". The source of the data which is used to analyze the "unknown" plant is not model-based (hence our requirement to conduct SYS ID), but rather from a data record. Due to the lack of known parameters, there is no other way to obtain the transfer function that describes the response of the satellite to a given command. Essentially, the data received is essentially a streaming collection of random variables and it is how those random variables are analyzed that will allow for future parametric studies via a derived accurate transfer function. The data from the rate gryos must travel through space and this becomes a source of measurement error. Each value transmitted is digital. It is improbable that every "1" and "0" will arrived true to their value, and there will exist a certain degree of white noise that will be received by the equipment tuned to receive the plant data.

The data received will be used to define the motion of the antenna as it responds to an input command that will be in terms of a "rate/time". Ideally, when the antenna is under no command, the data will reflect no change in rate, and the converse is obviously true, but there must be consideration given to the fact that the antenna structures are flexible appendages. They were modeled as such in a controlled environment, and if successfully deployed once on station will remain flexible. So it is to be expected that the given modes of the flexible appendages will be excited at various input frequencies. All together, the end state of any given command will be a collection of data points that

depict a rate of change, which itself is a collection of movement from the repositioning of the antenna assembly and the response of the flexible structures.

The earlier mention of random variables sets this thesis on a path of statistical analysis commonly associated with a wide array of stochastic processes. This thesis will rely on the Power Spectral Density Analysis and Cross-Power Spectral density as provided via the Fourier transform [13] in capturing the auto- and cross-covariances of the output and reference signal. This allows us greater insight into the frequency response of the plant, the heart of SYS ID. This thesis has a distinct advantage of working with a known quantity that allows the result to be easily compared to where it should be. The tools used to recover the known quantity become a working algorithm for system analysis and identification.

With a given set of data, the Fourier transform will take the signals in the time domain and convert them to a representation that depicts to what degree various frequencies are present. The Fourier transform, by its very nature, assumes a periodic signal. Thus, whatever we analyze will be treated as such by the Fourier algorithm.

III. Methodology

The concept and practice of Signal Identification is very broad, and in the end, the desired results can be just as varying. However, the overall problem remains constant: Given a set of input(s) and associated output data, determine the best way to characterize a relatively unknown system that is obtain its transfer function. The qualifier "relatively" is added to the phrase because it must be understood that there are varying degrees of baseline knowledge of the plant. It is possible to try and determine the transfer function of a plant which has never been seen. Such work can be seen by exciting particles at a molecular level and observing the output. By contrast, another scenario would focus on a manufactured structure that is physically inaccessible but none-the-less desirable to characterize. Such a structure in order to be useful must be responsive to commands. Any set of commands will ultimately determine mission success (assuming the metrics involve the ability for the antenna to be externally manipulated in support of a task, versus simply freely orbit and monitor its environment until end of lifecycle). A satellite intended for a specific purpose including communications, imaging, or conducting a wide array of experiments serves as the model provided for research detailed in this thesis.

This chapter details the steps of the process to determine the degree to which a transfer function is recoverable (in the form of a Bode frequency response plot) in the presence of white measurement noise. Roughly speaking, the process is completed twice: The first time, for the sake of calibration, a simple, second-order transfer function is used with a gain of K=1. Once the process is successful, the second set of simulations entails more complete plants.

The simple, second-order system with transfer function

$$X(s) = \frac{{\omega_n}^2}{s^2 + 2\xi\omega_n s + {\omega_n}^2} \tag{20}$$

is plotted via the tenants of Bode and yields Figure 10:

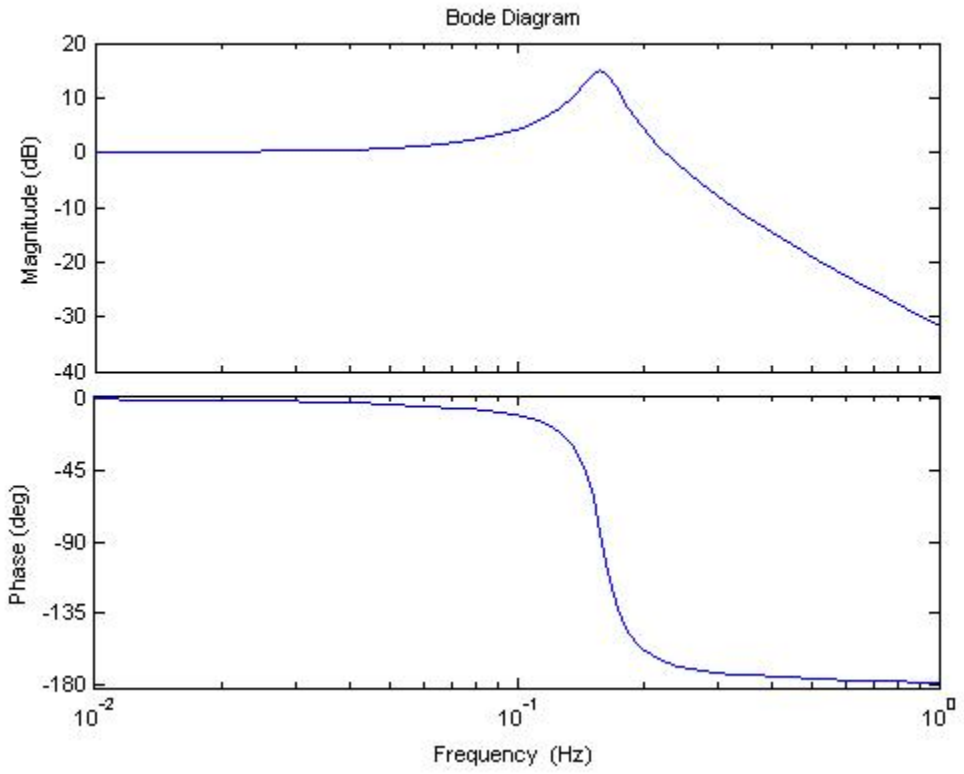


Figure 10. MATLAB Generated Bode Plot of Simple System

Note that the frequency is measured in cycles per second (Hertz, vice radians/sec) for continuity with the anticipated set of future simulations where the frequency will also be displayed in Hz for ease of comparison. ABode plot will serve as our two endpoints, that from which we start, and that which we will try to recover given simulated data. Figure 10, a standard Bode plot, reveals information at a glance about the system which serves

as motivation for its recovery. By analyzing the Bode plot, it is easy to see that this is a system that offers the following types of response to a given input:

$$y(x) = \begin{cases} \approx 0db, \quad 0.01 \le x < 0.07Hz \\ 0dB < y \le 15dB, \quad 0.07 \le x \le 0.19Hz \\ < 15db, \quad x > 0.19Hz \end{cases} \tag{21}$$

Note that there is a phase shift that occurs at the frequency coincidental to the max response (a.k.a. the corner frequency). The output signal will closely match the phase of the input until the input frequency approaches the corner frequency at which point it will get 90° out of phase and continue to reach a 180° offset as the input frequencies increase. The system has a positive gain margin and positive phase margin which, together, are indicative of stability [15].

With K=1, ξ=.09, the impulse response is as follows:

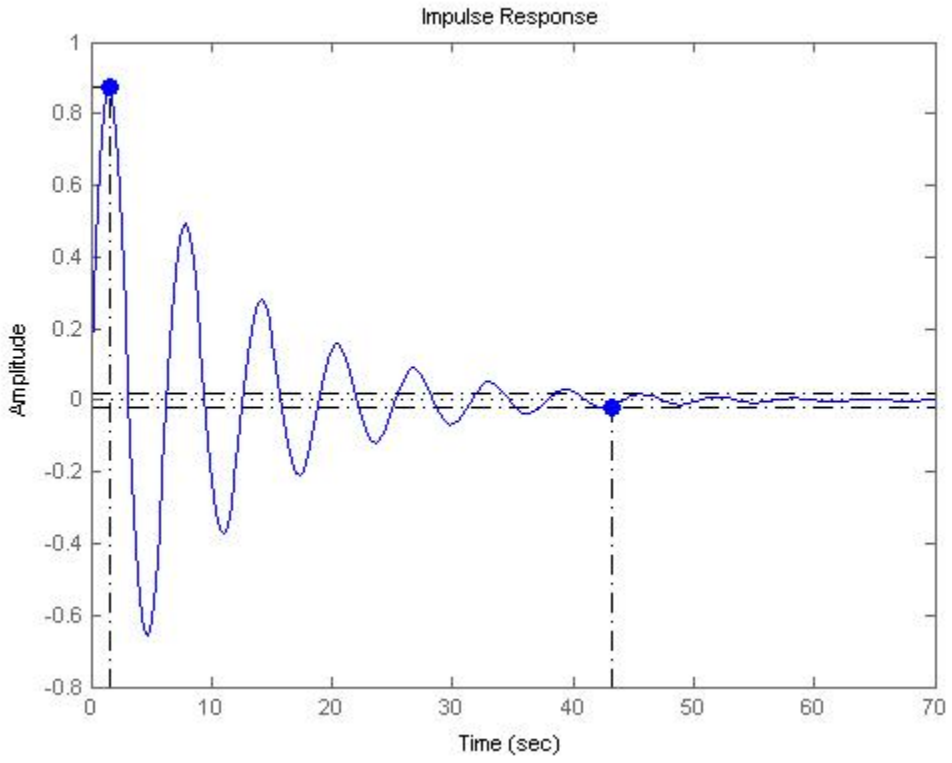


Figure 11. Impulse Response of Simple System

The purpose of this plot is to get a preliminary idea of how an open-loop control system responds. For an impulse response as seen here, we see that there is an immediate sinusoidal/periodic response. (There is a damping factor as denoted, with a peak amplitude of the output is less than that of the input.) The settling time to reach steady state (±2%) is shown to be 43seconds. The manner in which this information will be used is to establish the bounds for the first set of simulations in our attempt to establish a reference point which may be relied upon for verification or validation of a process used in successive attempts to recover an accurate frequency response plot.

To start the SysID process, the primary goal is to elicit as much information as possible. This information will serve as the building blocks for an accurate Bode representation. The information to be gained about how the output responds is directly proportional to the amount of information contained within the input signal. As a matter of control theory, and with the assumptions of linearity and time invariance, a steady state periodic (i.e. sinusoidal, cosinusoidal) input will asymptotically result in a response of equal period/frequency while the amplitude and overall phase will be a function of frequency [13]. In addition, the response to such would allow for the plotting of a single point of the entire Bode plot representing the response frequency and magnitude. The next step is to find the remainder of the points on the Bode plot. This requires more points of varying amplitude at their corresponding frequencies. Recalling a previous assumption about a steady state response having the same periodicity of the input signals, the intent now is to introduce an input signal, which in itself contains an amalgam of information that, in turn, will be mapped by a transfer function into a frequency response easily displayed and understood in a magnitude and phase plot.

The choice of input signals here needs to be what is referred to as “rich” in frequency content. The idea of introducing a simple period sine or cosine wave severely limits our frequency response as the result will mimic the input. Therefore, in order to recreate a Bode plot to cover a frequency spectrum from 0.1rad/s (.016Hz) to 100rad/s (7.9Hz) with a frequency resolution of 0.001 rad/s, we must provide an input at each frequency within the desired envelope for a total of 99901 points. Figures 12 and 13 show some various inputs and their frequency content.

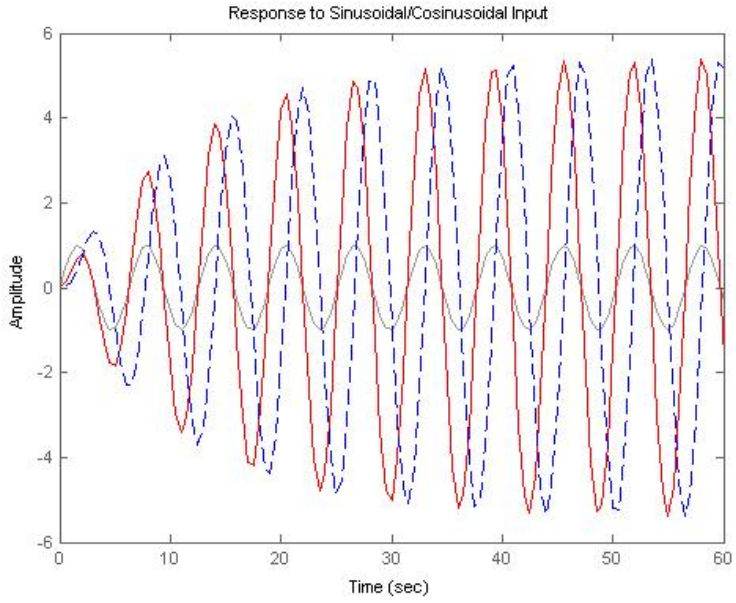


Figure 12. Periodic Input and Periodic Output, Single Frequency

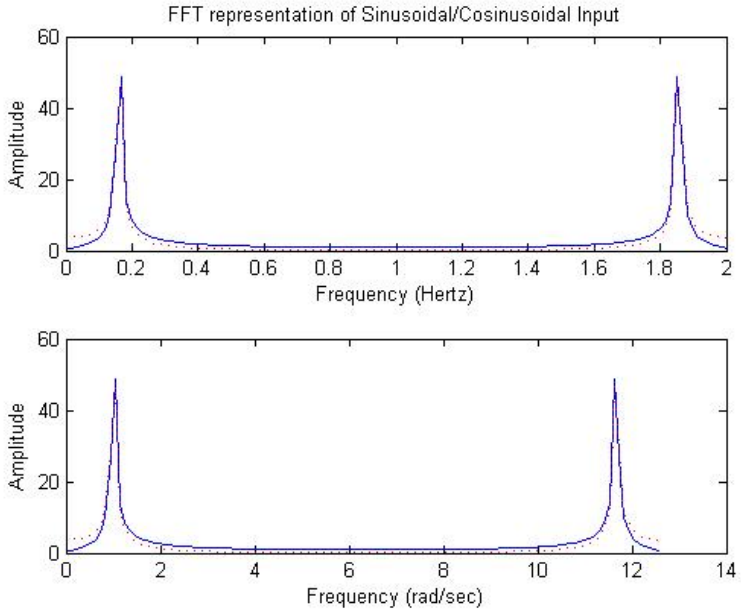


Figure 13. Frequency Content of Periodic Input

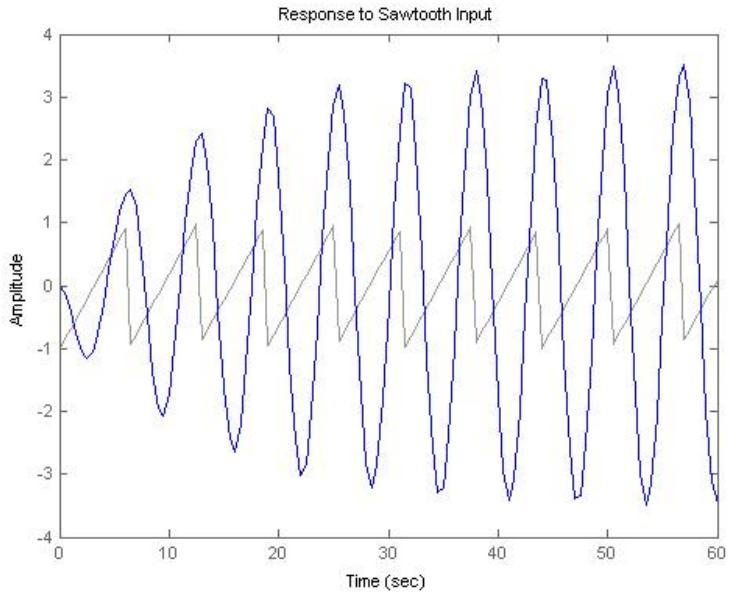


Figure 14. Sawtooth Input and Corresponding Periodic Output, Single Frequency

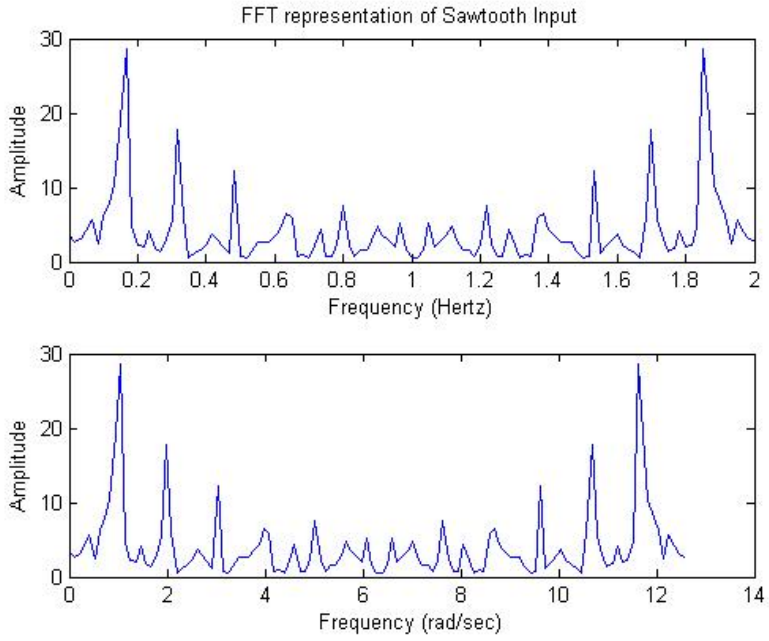


Figure 15. Frequency Content of Sawtooth Input

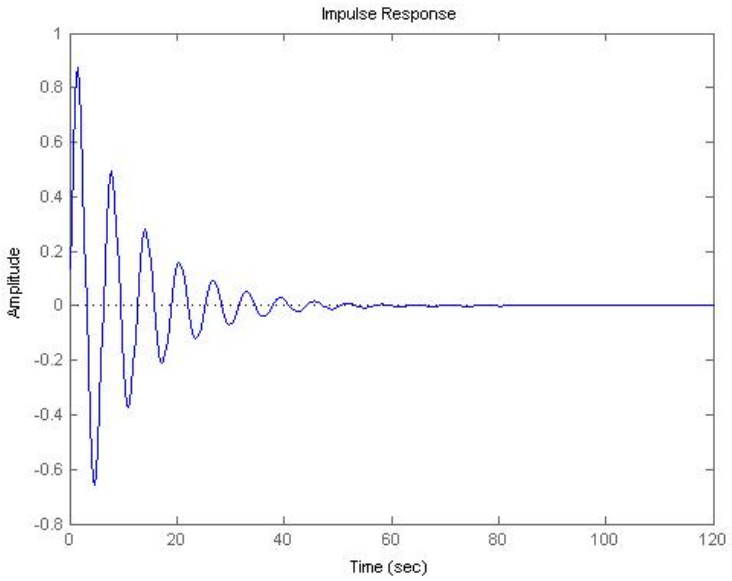


Figure 16. Response to an Impulse

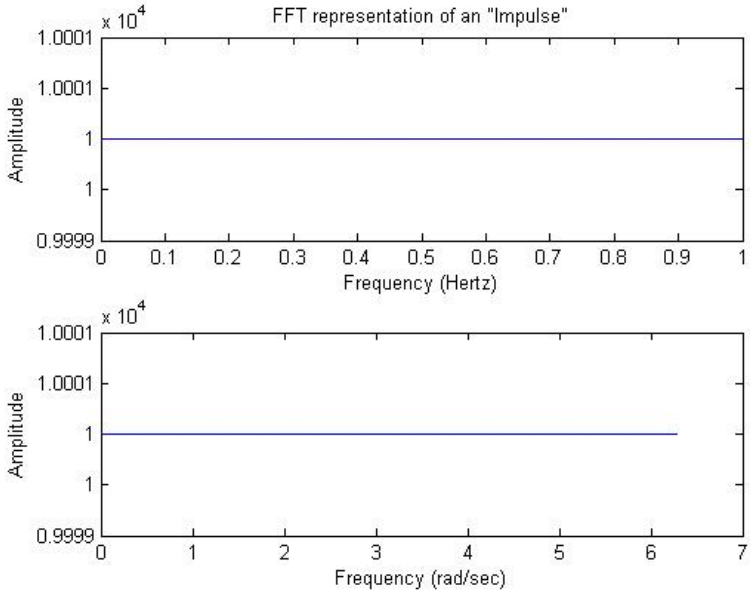


Figure 17. MATLAB Depiction of the Frequency Content of an Impulse

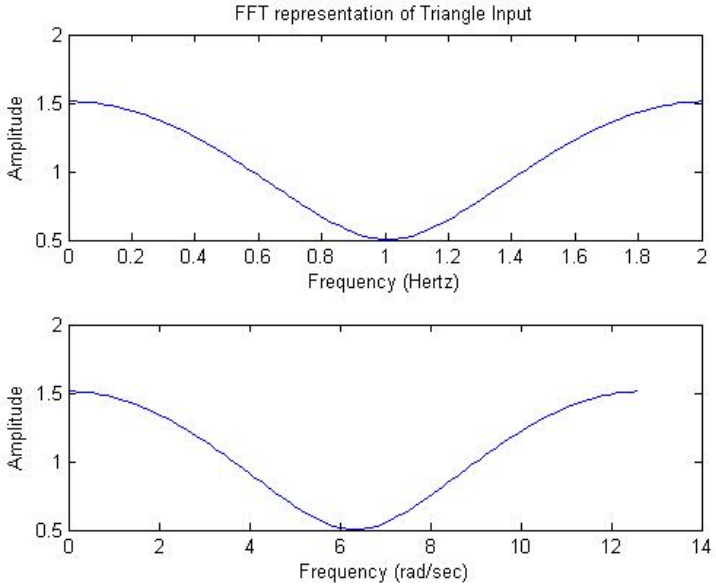


Figure 18. Triangular Wave Input, Fixed Period

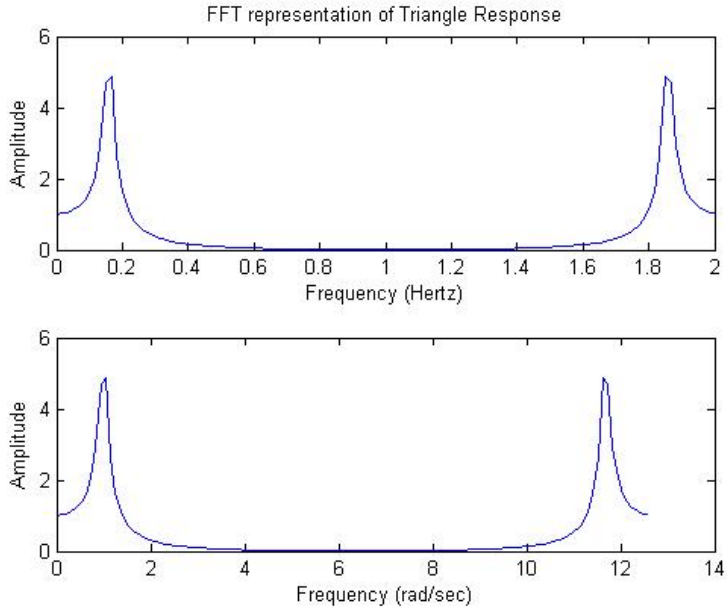


Figure 19. Frequency Content of Triangular Wave Input

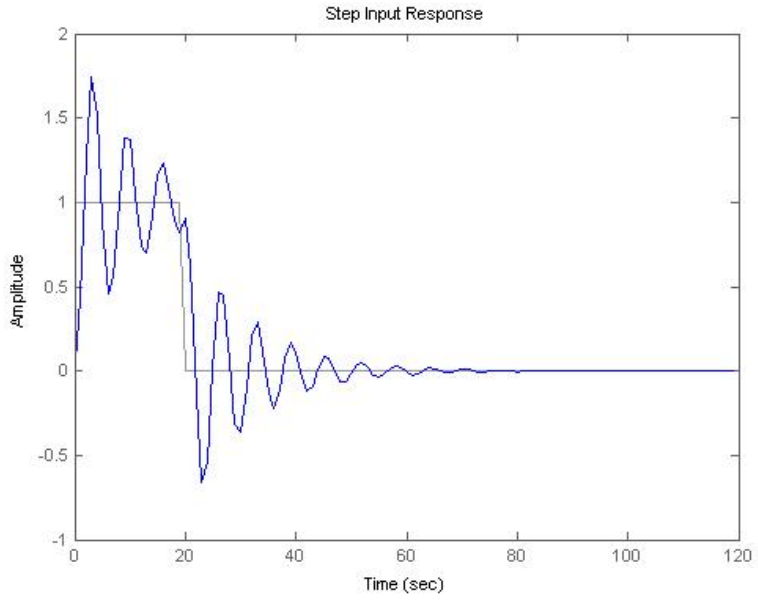


Figure 20. A Rectangular Pulse Input and Resulting Response

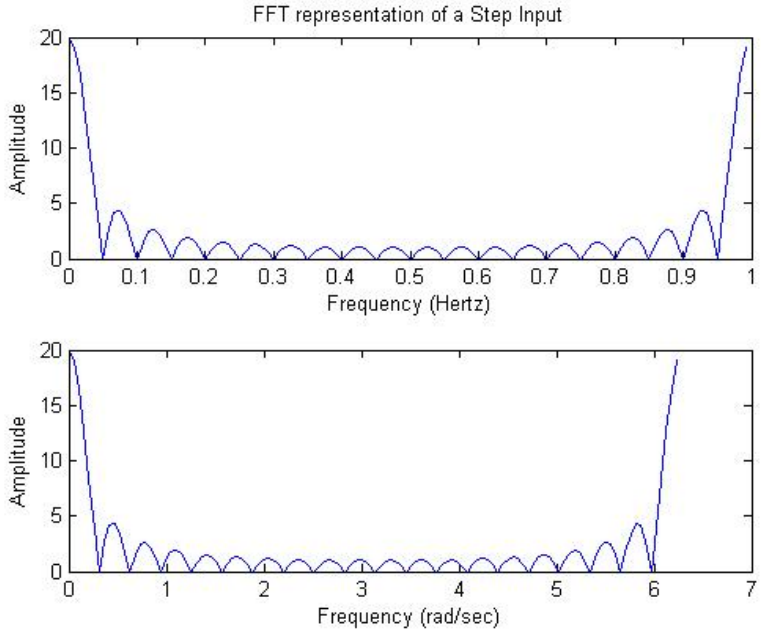


Figure 21. Frequency Content of a Rectangular Pulse Input

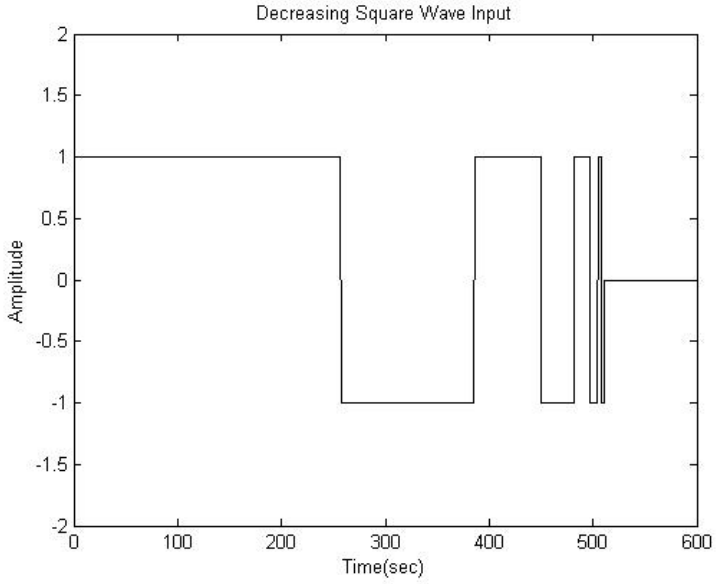


Figure 22. Decreasing Square Wave Input

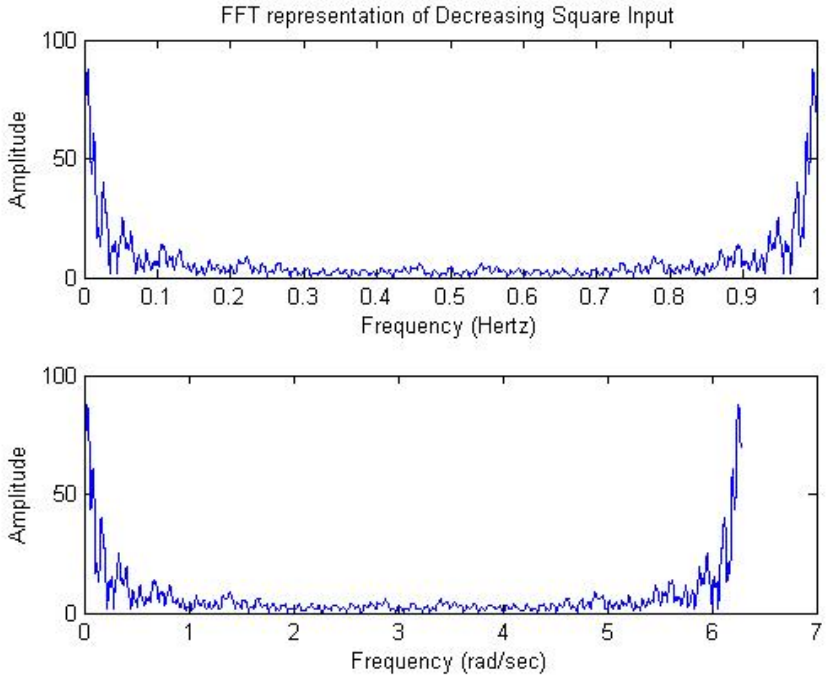


Figure 23. Frequency Content of Decreasing Square Wave

Table 1. Frequency Content of Various Input Signals

INPUT Type	**Frequency Content**
Sin/Cos	Very very low
Triangular	Very very low
Sawtooth	Very low
Impulse	By definition, an impulse will occur with delta t=0, therefore there is no associated frequency content, and presents a mathematical challenge for plotting the associated Fourier Transform. This does however elicit a useful response from a given system.
Rectangular Pulse	Moderate
Decreasing Square Wave	Very high

As highlighted in Table 1, there are two inputs that are composed of more frequencies than the others, the rectangular pulse and decreasing square wave (DSW). "More frequencies" translates into "more information". The DSW is considerably the richer of the two and this ability is due to the principle of superposition. Simply put, a square wave is actually the sum of multiple sinusoids, each of a different frequency added together to "cover" the same area as the resulting square wave.

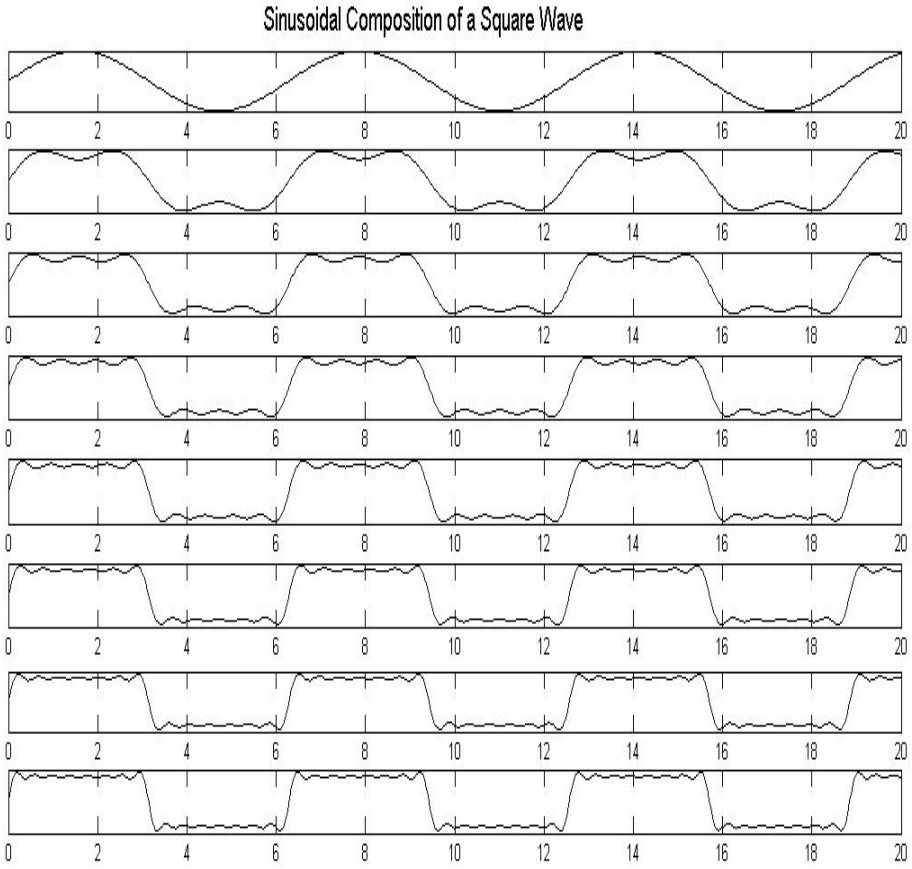


Figure 24. Sinusoidal Composition of a Square Wave [MATLAB generated]

As one might expect, varying the frequency of a sin/cos waveform alters its width by changing its period. Therefore, as the width of the square wave changes, so does the collection of its constituent frequencies. Such a changing square wave is an ideal way of acting on a given system with several inputs at once. In this thesis, the input of choice will be a decreasing square wave.

The algorithm used for the creation of such an input is as follows:

```
N=2048*2*2                % Number of Samples
BW=10*2                   % Bandwidth in Rads/sec
MaxRads=BW/2              % Max Obervable frequency in Rads/sec
MaxHz=BW/2/pi             % Max Obervable frequency in Hertz
Ts=2*pi/BW                % Sample time as per Nyquist
t=[0:N-1]*Ts;             % Time vector
lt=length(t);             % Length of time vector

u1a=ones(1,lt*.1);          % 1st cycle fraction with length 10% of time
u1b=-1*ones(1,lt*.05);      % 2nd cycle fraction with length 5% of time
u1c=ones(1,lt*.025);        % 3rd cycle fraction with length 2.5% of time
u1d=-1*ones(1,lt*.0125);    % 4th cycle fraction with length 1.25% of time
u1e=ones(1,lt*.006);        % 5th cycle fraction with length .6% of time
u1f=-1*ones(1,lt*.003);     % 6th cycle fraction with length .3% of time
u1g=ones(1,lt*.0015);       % 7th cycle fraction with length .15% of time
u1h=-1*ones(1,lt*.00075); % 8th cycle fraction with length .075% of time
U=[u1a u1b u1c u1d u1e u1f u1g u1h]; % Cycles 1-8 of DSW btwn "-1" and "1"
lU=length(U);             % Current length of DSW wave
u1r=zeros(1,lt-lU);       % Padding current length with zeros
udecsqr=[U u1r];          % Entire DSW with length of (t)
```

Figure 25. MATLAB Script for Decreasing Square Wave Input

Initially, the algorithm in Figure 25 was employed strictly as a constant collection of positive and negative multiples of unity, but it was discovered that as the number of samples increased, or as the sampling interval was altered it was possible for the input to become ineffectively small. By having each step of the DSW a constant fraction of the entire time vector, the input remains relatively constant in proportion to the entire sampled series, ensuring the excitation of the control system.

What remains is to excite the system with the DSW and recover what information we can. The information will be recovered using the fundamentals of Fourier's transform and series. The Fourier transform itself allows for the representation of a signal in the time domain to be translated into the frequency domain. The fast Fourier Transform has

greatly facilitated the way the Fourier transform is computed and as a result is an algorithm designed specifically around the binary nature of computing. It relies on a sample number that is a product of "2" to a power "n", hence, the number of samples will always be $N=2^n$.

$$F(\omega) = \int_{-\infty}^{\infty} x(t) e^{-i\omega t} dt = \int_{-\infty}^{\infty} x(t) e^{-i 2\pi f t} dt \tag{22}$$

$$X_l = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nl} \tag{23}$$

An example of how the FFT works can be seen by reviewing figures of the inputs. When in MATLAB, the command for the Fast Fourier Transform yields the result as obtained by the following method:

$$X_l = \sum_{k=0}^{\frac{N}{2}-1} x_{2k} e^{-\frac{2\pi i}{N} 2kl} + \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} e^{-\frac{2\pi i}{N}(2k+1)l} \tag{24}$$

$$= \sum_{k=0}^{L-1} x_{2k} e^{-\frac{2\pi i}{L} kl} + e^{-\frac{2\pi i}{N} l} \sum_{k=0}^{L-1} x_{2k+1} e^{-\frac{2\pi i}{L} kl} \tag{25}$$

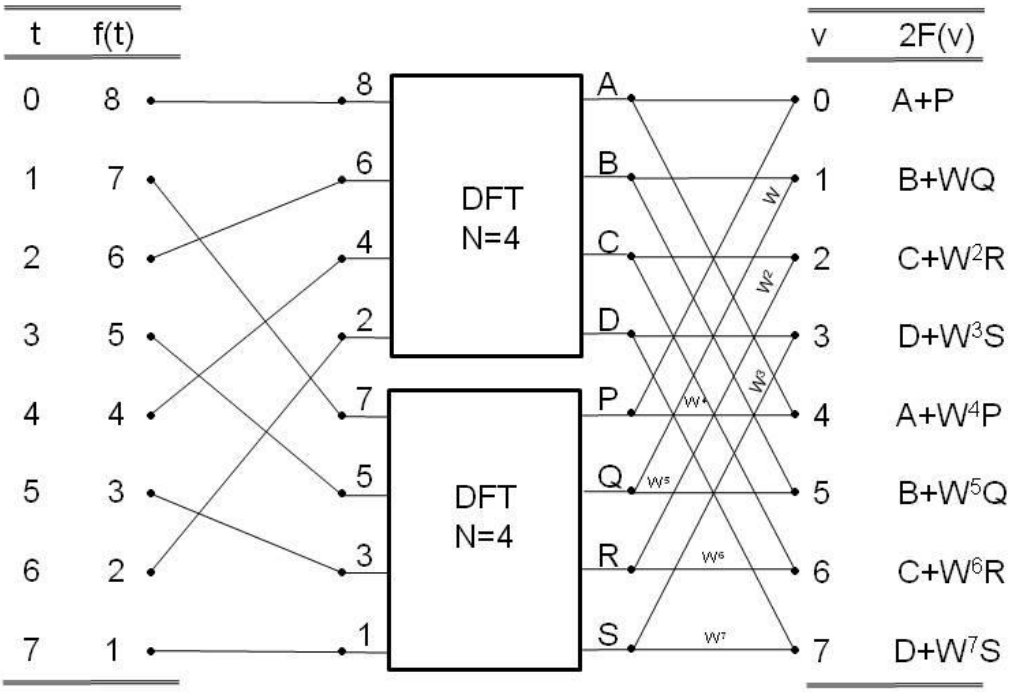


Figure 26. Fast Fourier Transform Where N=8 [13]

The diagram denotes how the FFT algorithm works. Simply stated, the given series of values (as depicted here, 8 samples) is first divided into half. And the division would continue until it is no longer possible to divide the number of samples by 2 and grouping the samples into even and odd sample indexes. The newly arranged data is then combined linearly as denoted by the lines of action. This serves to greatly reduce the number of total computations by N^2-$N\log_2 N$ [13]. If N=32, that's a savings of over 10^6 calculations, but we'll be looking at multiples of N=2048, therefore the savings will prove to be immense.

As for the samples themselves, the intervals and total numbers of samples (in our case a 2 raised to a power "n") are crucial. We are cognizant from the start with issues of

under-sampling, aliasing and oversampling. Consideration must be given as to how much of the response needs to be captured. And not one concept should be considered without the other. Specifically, while it may sound advantageous to choose a number of samples where N=2048, it will most likely prove counterproductive to choose a sampling interval of 2 seconds, especially if the overall response to a given input lasted no longer than 43 seconds. In such a case, a characterization would be attempted with only 21 samples as the remaining 2027 are of "steady state". Or conversely, a sampling rate (T_s) of .001 seconds over the course of N would leave a plot only 2.048 seconds' worth of the 43 second response. In attempt to characterize the entire response, this is useless.

A stipulation for the Nyquist sampling theorem is that the signal itself must be band limited or have a value greater than zero over its Fourier transform [14]. If this is the case, a uniform sampling frequency must be greater than the signal's entire bandwidth in order to fully reconstruct the signal up to and including the highest frequency, otherwise referred to as the cutoff frequency (f_c).

$$f_s > 2 f_c \tag{26}$$

$$2 f_c = \text{Bandwidth (BW)} \tag{27}$$

$$T_s(\text{sec}) = \frac{1}{2 f_c} = \frac{1}{BW} \tag{28}$$

$$T_s(\text{sec}) = \frac{2\pi}{2 f_c} = \frac{2\pi}{BW}[\text{rad/sec}] \tag{29}$$

Note that Nyquist (Nyquist-Shannon) merely establishes the minimum Nyquist rate and Nyquist interval for signal reconstruction. Failure to use the minimums will lead to under-sampling and the associated challenges of incomplete signal reconstruction ensue. The transform that the FFT provides assumes a periodic signal and maps that concept from the time domain to the frequency domain. So, whatever signal is present will be repeated with a period of 2π in the frequency plot. And since the sampling rate utilized is inversely related to the bandwidth/cutoff frequency, it is the cutoff frequency that determines when the signal is repeated in the graphical reconstruction. The cutoff frequency is the point beyond which no information is passed on to the Fourier transform. Should the sampling interval be excessively low, then there is a guarantee that the transformed signal will repeat at an interval that is much less than the actual interval of the original signal. There will be distortion in the reconstructed signal and this is referred to as “aliasing”, a phenomenon due to the overlapping frequency components [15]. When sampled at a rate greater than the Nyquist minimum, there is no overlapping of frequency components; rather at the end of a period of reconstruction the cutoff frequency is readily displayed. However, it must be noted that sampling at a rate equal to the cutoff frequency might yield confusion right at the end point as there will be no separation between repeating cycles. Hence, throughout all of the simulations in this thesis, N will be of a value much greater than the cutoff frequency of the characteristic equation.

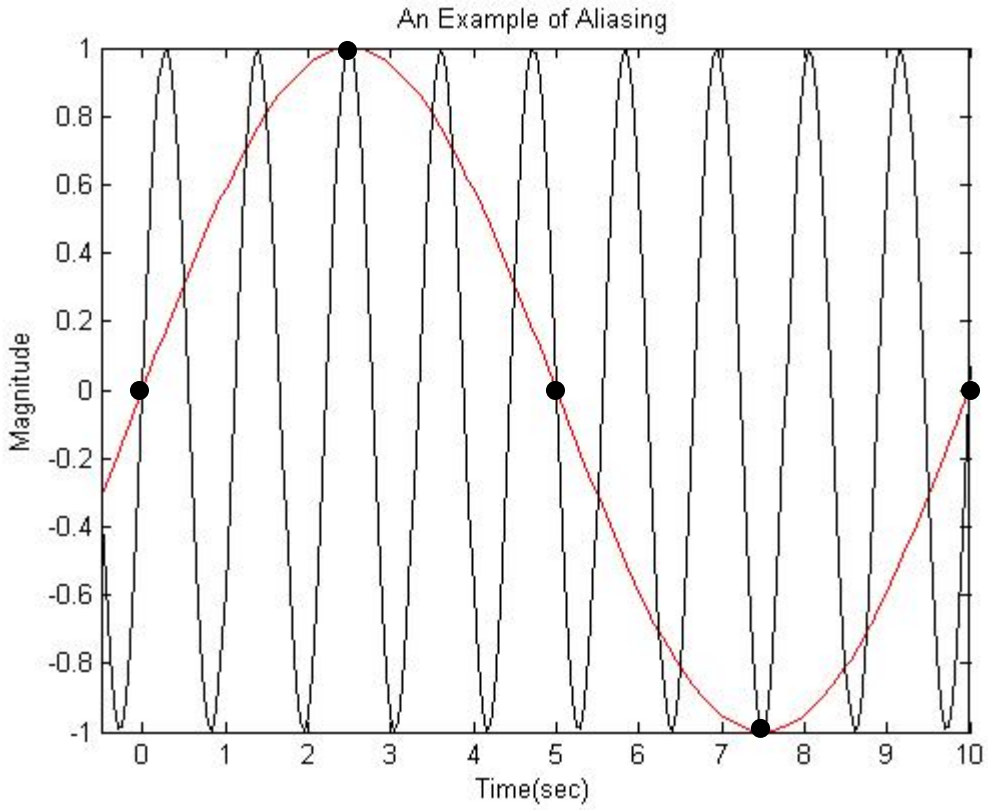


Figure 27. A High Frequency Signal Might Appear to be Something Different

Figure 27 the signal y=sin(.9t) should be sampled at a frequency equal to the bandwidth (1.8rad/s). Therefore, the sampling interval of .5556 seconds will capture the points that make up the higher frequency sinusoid. If instead the sampling interval does not adhere to the Nyquist minimum and instead is sampled at multiples of 2.5 seconds then the result is a collection of 5 points that will appear to describe the lower frequency sine wave. The true signal is "masked" or falls victim to aliasing due to inadequate information resulting from undersampling. Suppose the minimum sampling time was increased by 10, the only downside being addition of more points, thus requiring more computations. The reconstruction however is sure to recover the original signal.

With the preliminaries covered, the groundwork has been laid for choosing the number of samples, at which interval to conduct the sampling and roughly how the mapping from the time domain to the frequency domain occurs. The following highlights some of the MATLAB coding to used for signal analysis.

```
z=.09
num=[1];
den=[1 2*z 1];
sys=tf([num],[den])        %   X(s) = 1 / (s^2 + 2(.09)s + 1)

figure
bode(sys)
P = bodeoptions;           % Create plot with the options specified by P
P.FreqUnits = 'Hz';        % Set frequency units to Hz in options
figure
h = bodeplot(sys,P);

fc=.6                      % cut off frequency (Hz)...the highest frequency we want to
                             include in reconstruction

fs=OS*B                    % sampling frequency (Hz)
Ts=1/fs                    % sampling interval (sec)
MaxHz=fs/2                 % max observable frequency expected in reconstruction (Hz)
MaxRads=fs*pi              %max observable frequency expected in reconstruction (rad)
N=2^11                     % N=2048
t=[0:N-1]*Ts               % the time vector with same length as "N" with interval of Ts
```

Figure 28. MATLAB Script Establishing N and T_s

The transfer function of the simple system is identified and a cutoff frequency is arbitrarily chosen to be .6 Hz as this corresponds to a point where the output signal crosses the -20dB line, an arbitrary value beyond which we have little interest. The intent is merely to recover the portion of the Bode plot surrounding the magnitude peak at 15dB. This magnitude offers our greatest response, so it becomes the key point of

recognition in our reconstruction. We choose a frequency well beyond the corner frequency to mitigate the chance for ambiguity when repetition of the signal occurs.

Beyond simply choosing a max frequency, a factor by which to oversample is included and the number of samples to take is included. Consideration must be given to the fact, however, that the number of samples times the sampling interval will determine the sampling window. In this case, $N*T_s \approx 21.3$ seconds. Let's take a look at the response of the system to a decreasing square wave to see if this makes sense:

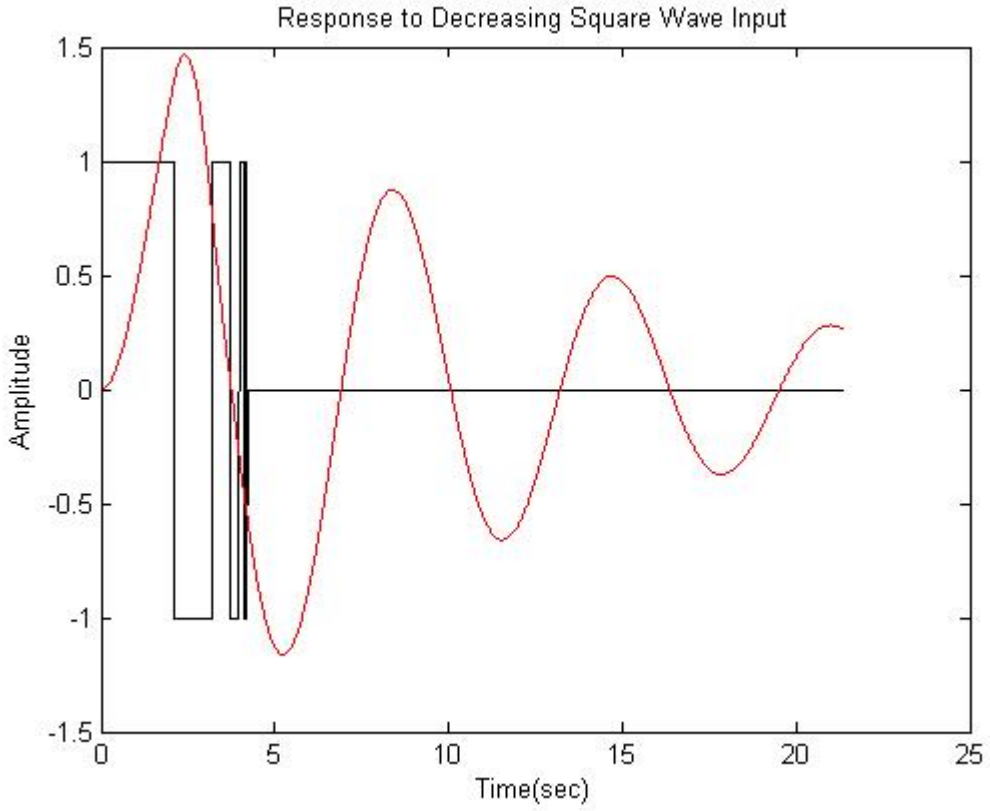


Figure 29. Response as seen over 2048-points

As displayed by Figure 29, the response is still changing at the end of our sampling window. This is not a good strategy when trying to characterize the full response to an input. By increasing N, and retaining the current value of T_s, we can

capture more of the response. The combination of N=2048 and T_s=.0104 seconds is inadequate. Increasing the number of sampling points by a factor of 4 yields the following response:

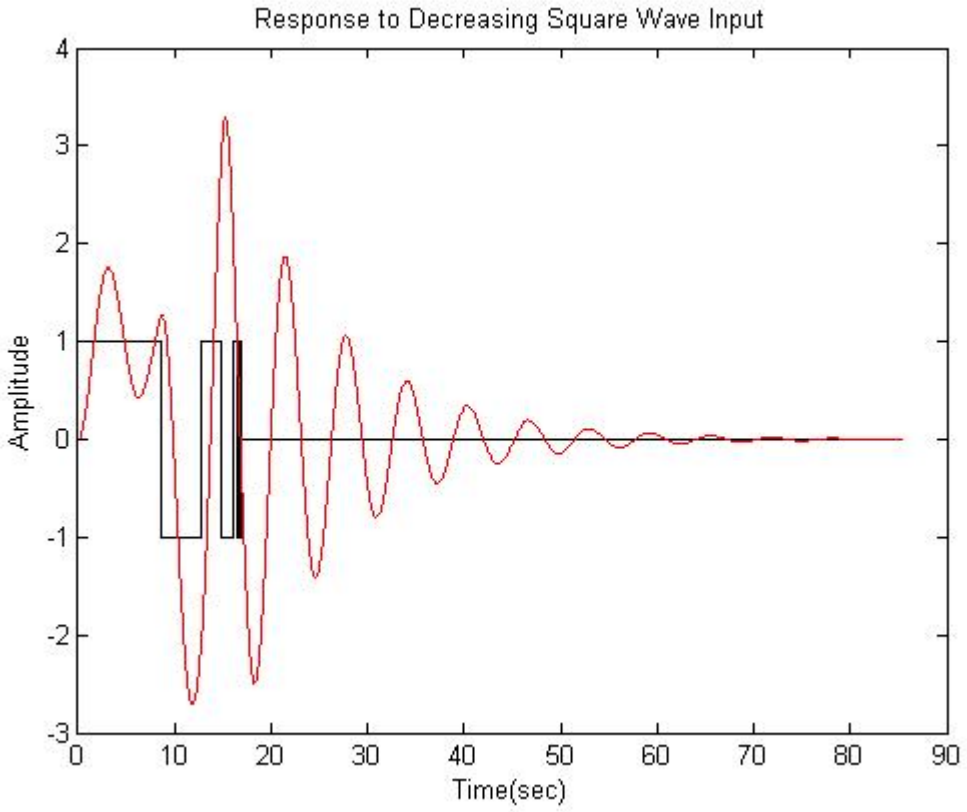


Figure 30. Response as seen over 8196-points

Figure 30 provides a total sample window of approximately 83 seconds which allows enough time for the return to a zero response after being excited by the decreasing square wave.

The next step is to capture the values of the response and transform them from the time domain into the frequency domain. This is done via the MATLAB "LSIM" and "fft" commands in the following script:

```
ydecsqr=lsim(sys,udecsqr,t);  % assigns the value of sys response to the DSW
Y=fft(ydecsqr);               % takes the values of 'ydecsqr' and runs it
                                 through the FFT algorithm
N=length(Y)
f=[0:N-1]/N/Ts;
w=[0:N-1]*2*pi/N/Ts;
figure
SUBPLOT(2,1,1), plot(f,abs(Y))
title('FFT representation of Decreasing Square Response')
xlabel('Frequency (Hertz)')
ylabel('Amplitude')
SUBPLOT(2,1,2), plot(w,abs(Y))
xlabel('Frequency (rad/sec)')
ylabel('Amplitude')
```

Figure 31. MATLAB Script to Transform Time Domain to Frequency Domain

The following plot is the result.

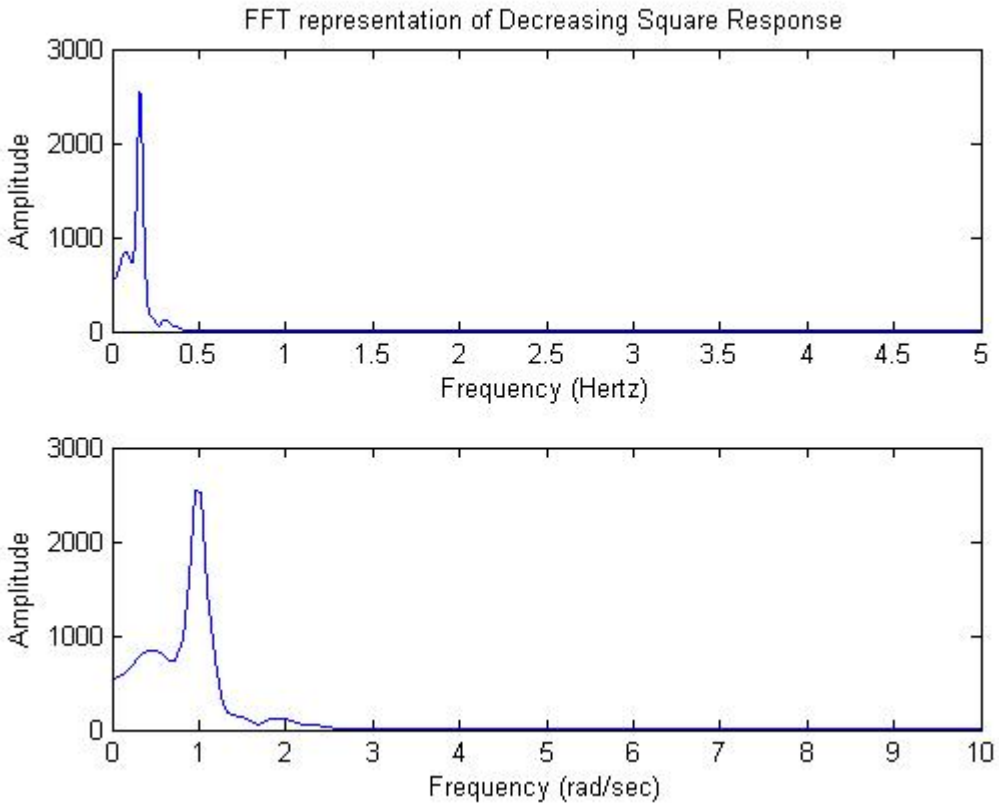


Figure 32. FFT of System Response to a Decreasing Square Wave (DSW) Input

Of particular note, it should be stated that previously we had assigned a vector for "t" with a length equal to *N*. As detailed in Figure 33 and by reviewing the MATLAB script, the transformations from a time vector to frequencies (both Hertz and rad/s) are easily made.

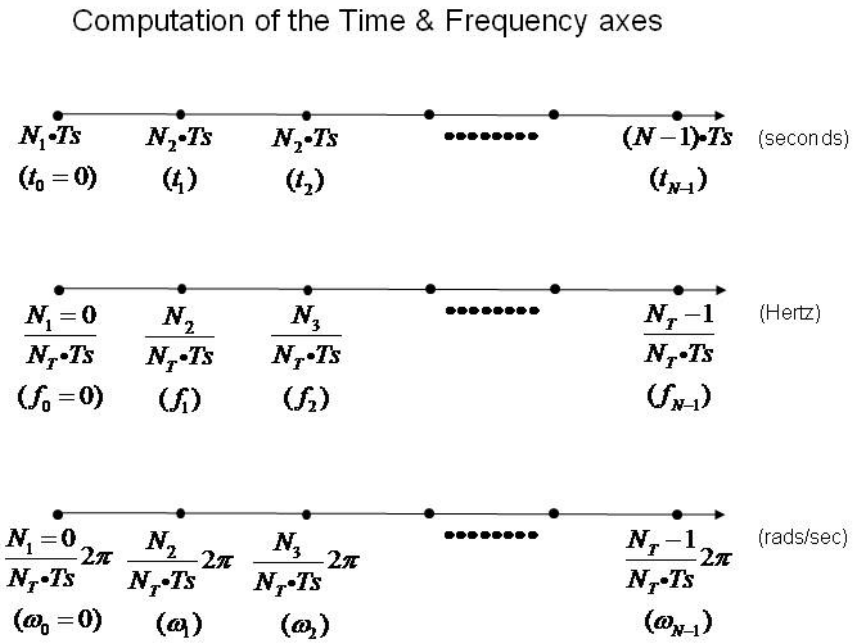


Figure 33. Time and Frequency Axis

To this point, both the input and the output have been tested separately. However, it has been mentioned that the output is related to the input and this "transfer function" is now obtained. Of the several tenants of control theory, that which is relied upon most in this paper deals with the assumption of linearity and time independence. Moreover, the response (h(t)) of a system to an impulse (δ(t)) may define the system entirely [5],[15]. Thus, once the frequency response is identified, we can apply any input and the input will be appropriately mapped to the correct output.

$$h(t) = T[\delta(t)] \tag{30}$$

It follows that the manner in which a unit impulse is transformed to an output will be the same manner in which subsequent inputs may be processed.

$$y(t) = h(t) * x(t) \tag{31}$$

$$Y(\omega) = H(\omega)X(\omega) \tag{32}$$

Where $$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-i\omega t}dt \tag{33}$$

And upper case X and Y are also their respective Fourier Transforms. Thus we have the relationship for the transfer function of the LTI defined as a ratio of the DFT (in our case we will use FFT) of the output to the FFT of the input.

$$\frac{Y(\omega)}{X(\omega)} = H(\omega) \tag{34}$$

It's important to realize however that when entering into the frequency domain, the realm of complex numbers plays a role in how the MATLAB code is scripted.

```
1  X=fft(udecsqr);
2  Y=fft(ydecsqr);
3
4  XM=abs(X);
5  YM=abs(Y);
6
7  mag=20*log10(YM./XM');
8  phi=angle(Y)-angle(X)';
9  phi=unwrap(phi);
10 phi=rad2deg(phi);
11
12 figure
13
14 SUBPLOT(2,1,1), semilogx(f,mag)
15 title('Plot of FFT(DSW Response)/FFT(DSW)')
16 xlim([.05 .35])
17 ylim([-40 20])
18 ylabel('Magnitude')
19
20 SUBPLOT(2,1,2), semilogx(f,phi)  % 'f'is previously defined as f=[0:N-1]/N/Ts;
21 xlabel('Frequency (Hertz)')
22 xlim([.05 .35])
23 ylim([-250 250])
```

Figure 34. Ratio of FFTs to Obtain 'H'

The approach in the coding (referring to the incorporation of the "ABS" and "ANGLE" commands) is due to the complex numbers within the value of the Fourier transform of both the input and output. Lines 8, 9 and 10 are collectively a result of several attempts to reign in the display of the phase shift. Initially, the attempt was made to simply define "phi" in MATLAB as the difference in phases by stating "phi=phase(Y)-phase(X)". For small phase variations this works fine, but the challenge of increasing phase shifts soon leads to spikes in the plot that throw the overall scale into a form that masks the smaller

values. Whereas the “phase” command in MATLAB delivers a value that is accurate, the values delivered by a combination of “angle” and “unwrap” are more manageable (lines 8, 9 and 10).

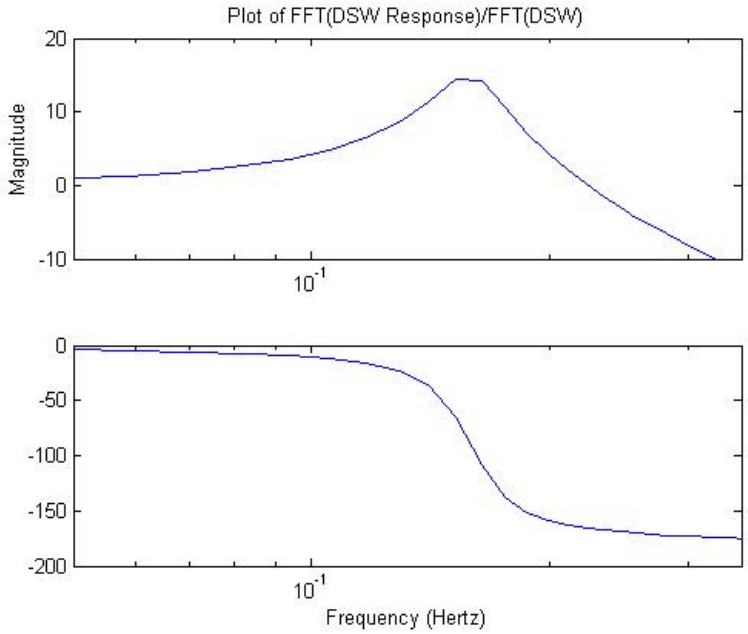


Figure 35. Reconstructed Bode Plot of a Simple System

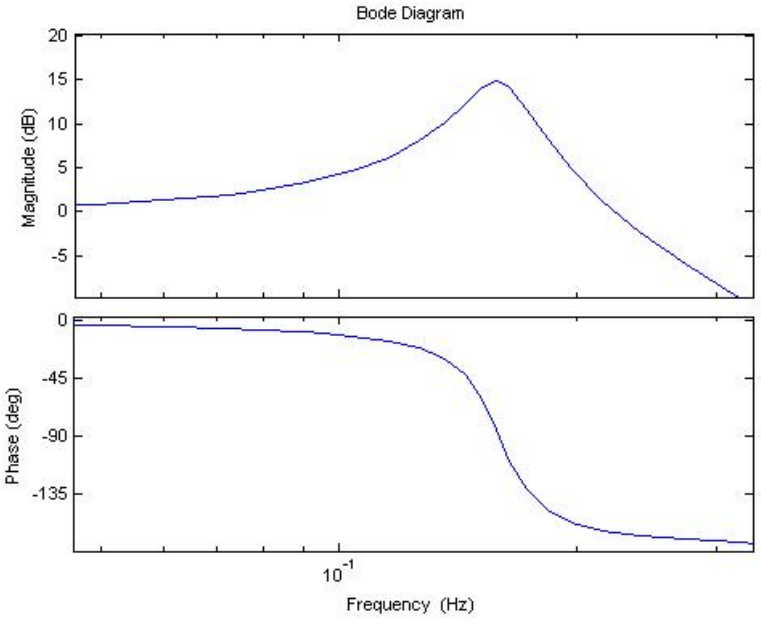


Figure 36. MATLAB Generated Bode Plot of a Simple System

The previous two plots show by that plotting the ratios of the FFTs it is possible to match the Bode plot. Moreover, what is proven is that with a frequency-rich input, it is possible to excite the frequencies of the system to such a degree that the output matches the same system response to an impulse which is used to define the transfer function in the first place. However, comparing the two plots, there is obviously room for improvement. From previous discussion we know that our method to increase resolution of our plot is to obtain an appropriate combination of samples, sampling intervals and the window of time itself. Because it is necessary to capture the response completely (a point defined as where the reponse reaches "steady state" which itself is defined as a final value ±2%), our window of time is fixed. Our remaining options are to increase the number of samples and alter the sampling interval. Since they are dependent upon each other given a fixed time interval, we will adjust both. By increasing the N from 2^13 to 2^15 and changing T_s by a factor of one-half, the following plot is obtained:

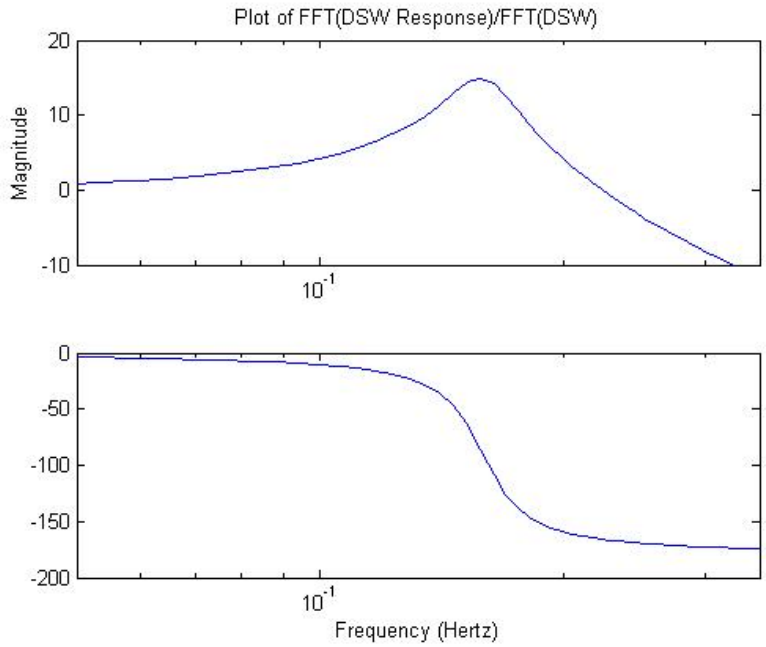


Figure 37. Improvement of Figure 35 by Increasing N and Reducing T_s

The resolution is most readily noticeable in the peak of the amplitude where the refined version is more representative of the actual response around the corner frequency. As depicted in Figure 36, the MATLAB generated Bode plot, the peak is at (.158Hz & 14.9dB). The refined reconstruction plot, Figure 37, shows the same peak at (.158Hz & 14.9dB), in contrast to Figure 35 which depicted the relative maximum as a point to be interpolated between (.152 and .164)Hz and (14.36 and 14.17)dB. Moreover, the refinement in our reconstruction is a result of an additional 24,567 data points at every .0052 seconds versus a previous T_s=.0104 seconds. Such an improvement only forces a comparison to a previous plot, that of the frequency spectrum of simply the output presented earlier. Figure 38 highlights the contrast.

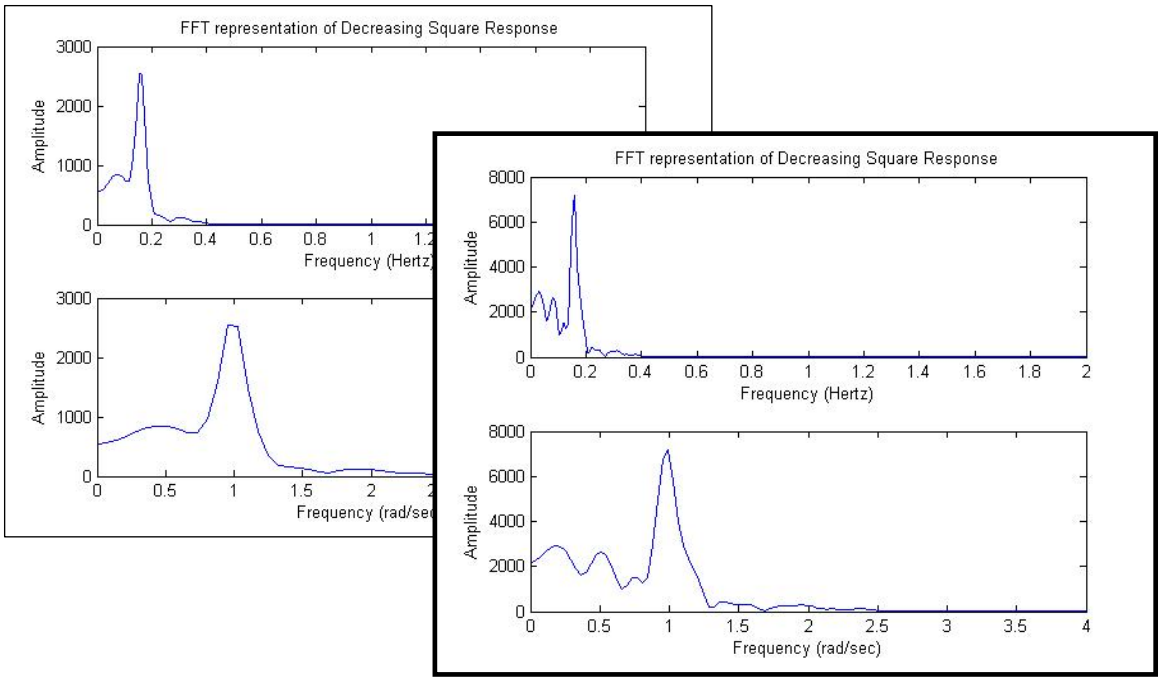


Figure 38. Increased Resolution with a Decrease in T_s

Figure 38 has the most recent reconstruction of the plot forward. An increase in frequency resolution is noticeable by additional peaks. This drives home the necessity for a constant desire for an ever increasing N and a decreasing T_s.

Up until now, all work has been done on a clean signal. But it is ludicrous to expect anything near "clean" when analyzing a signal originating from an Earth orbit. There are infinite means of signal corruption the easiest of which to deal with is white noise. In reality, the color spectrum of possible noise is far too complex to simply label as "white", but it will serve us well for the introduction of statistical uncertainty. Moreover, there will be other factors much more prevalent than random noise. The following table shows three sets of two plots. Each group shows the response to the same simple system as above, but now with the addition of various levels of "random white" noise added via the following script:

```
1    ynoisy1=ydecsqr+.1206*randn(length(ydecsqr),1);       %-80dB
2    ynoisy2=ydecsqr+.001203*randn(length(ydecsqr),1);   %-40dB
3    ynoisy3=ydecsqr+ .00001206*randn(length(ydecsqr),1);%0dB
```

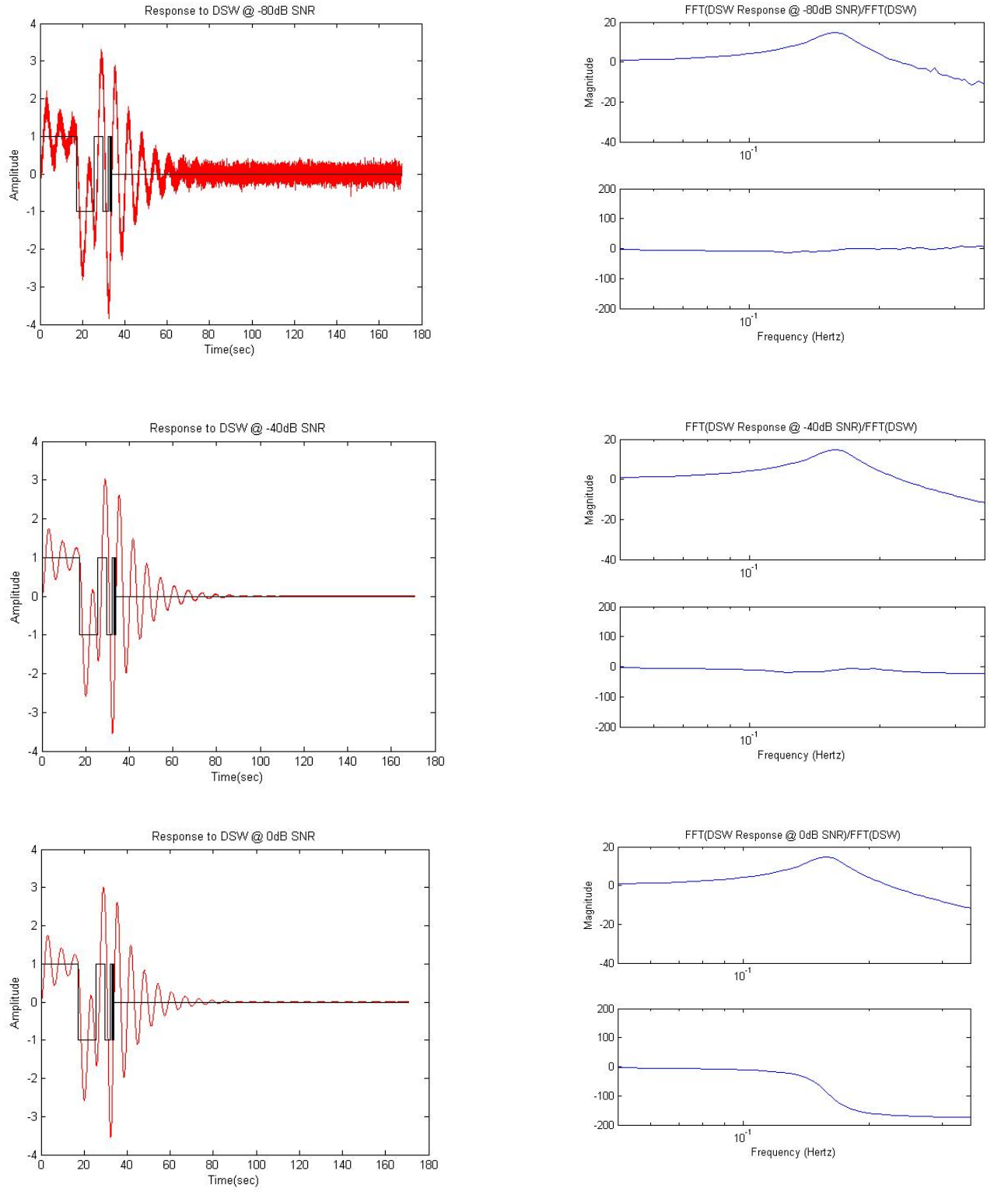


Figure 39. Reconstruction of Bode Plot Using Straight FFT Ratios

The first plot represents a signal-to-noise ratio (SNR) of -80dB, and the next two depict -40dB and 0dB respectively. While the most noticeable difference in the plots is found by comparing the 0dB plot with the previous two, it's important to highlight what is missing in the first two plots, namely, an appropriate phase shift. Recalling that all three reconstructions were created using the same algorithm based of FFT ratios and phase differences, there is a noticeable shortcoming. It is important to incorporate more statistical data into consideration. Here we modify the previous FFT ratios to ratios of Cross Power Spectral Density to the Power Spectral Density of the input, with each defined as:

$$\frac{S_{XY}(f) = \mathbb{F}\left\{R_{XY}(k)\right\}}{S_{XX}(f) = \mathbb{F}\left\{R_{XX}(k)\right\}} \tag{35}$$

The cross and auto-correlations are defined as:

$$R_{XY}(t) = \int_{t_0}^{t_f} g(u-t)\, f(t) dt \tag{36}$$

$$R_{XX}(-t) = x(-t) * x(t) = R_{XX}(t) \tag{37}$$

Where the cross-correlation is similar to the convolution of the input with the output, but there is no signal reversal [13]. And as is the case with the auto-correlation becoming unity at a point of no time shift, such can be for the cross-correlation assuming it is normalized. When using the MATLAB command "XCORR", unity at $\delta t=0$ can be specified via a defined set of options.

In both of the above equations, when the transition is made from the time domain into the frequency domain the following relationships hold:

$$R_{XY}(\omega) = H(\omega)R_{XX}(\omega) \tag{38}$$

$$\frac{R_{XY}(\omega)}{R_{XX}(\omega)} = H(\omega) \tag{39}$$

Additionally, stochastic noise is defined to have an overall mean of zero, thus having a corresponding average and more importantly, a constant power. This is understood to be a gross simplification for the purpose of the thesis, and will be addressed further in Chapter 5. This assumption will eventually allow for the inclusion of a cost function that will serve as a source of error. The following MATLAB script allows for obtaining the Power Spectral Density and Cross-Power Spectral Density as well as the corresponding reconstructed Bode plots in terms of magnitude and phase shift (degs vs. Hertz).

```
1     Y=xcorr(udecsqr,ydecsqr');
2     X=xcorr(udecsqr);
3     YY=abs(fft(Y));
4     XX=abs(fft(X));
5     YT=fft(Y);
6     XT=fft(X);
7     ly=length(YY);
8     lx=length(XX);
9     mag=20*log10(YY./XX);
10    phi=angle(XT)-angle(YT);
11    phi=unwrap(phi);
12    phi=rad2deg(phi);
13    N=ly;
14    f=[0:N-1]/N/Ts;
15    w=[0:N-1]*2*pi/N/Ts;
```

```
figure
SUBPLOT(2,1,1), semilogx(f,mag)
title('Plot of FFT(Rxy)/FFT(Rxx)(DSW Response)')
xlim([.05 .35])
ylim([-40 20])
ylabel('Magnitude')
SUBPLOT(2,1,2), semilogx(f,phi)
xlabel('Frequency (Hertz)')
xlim([.05 .35])
ylim([-200 0])
Y=xcorr(udecsqr,ynoisy1');
X=xcorr(udecsqr);
YY=abs(fft(Y));
XX=abs(fft(X));
YT=fft(Y);
XT=fft(X);
ly=length(YY);
lx=length(XX);
mag=20*log10(YY./XX);
phi=angle(XT)-angle(YT);
phi=unwrap(phi);
phi=rad2deg(phi);
N=ly;
f=[0:N-1]/N/Ts;
w=[0:N-1]*2*pi/N/Ts;
figure
SUBPLOT(2,1,1), semilogx(f,mag)
title('Plot of FFT(Rxy)/FFT(Rxx)(DSW Response + .0001*noise)')
% xlim([.05 .35])
% ylim([-40 20])
ylabel('Magnitude')
SUBPLOT(2,1,2), semilogx(f,phi)
xlabel('Frequency (Hertz)')
% xlim([.05 .35])
ylim([-200 0])
Y=xcorr(udecsqr,ynoisy2');
X=xcorr(udecsqr);
YY=abs(fft(Y));
XX=abs(fft(X));
YT=fft(Y);
XT=fft(X);
ly=length(YY);
lx=length(XX);
mag=20*log10(YY./XX);
phi=angle(XT)-angle(YT);
phi=unwrap(phi);
```

```
62    phi=rad2deg(phi);
63    N=ly;
64    f=[0:N-1]/N/Ts;
65    w=[0:N-1]*2*pi/N/Ts;
66    figure
67    SUBPLOT(2,1,1), semilogx(f,mag)
68    title('Plot of FFT(Rxy)/FFT(Rxx)(DSW Response + .01*noise)')
69    % xlim([.05 .35])
70    % ylim([-40 20])
71    ylabel('Magnitude')
72    SUBPLOT(2,1,2), semilogx(f,phi)
73    xlabel('Frequency (Hertz)')
74    % xlim([.05 .35])
75    ylim([-200 0])
76    Y=xcorr(udecsqr,ynoisy3','coeff');
77    X=xcorr(udecsqr,'coeff');
78    YY=abs(fft(Y));
79    XX=abs(fft(X));
80    YT=fft(Y);
81    XT=fft(X);
82    ly=length(YY);
83    lx=length(XX);
84    mag=20*log10(YY./XX);
85    phi=angle(XT)-angle(YT);
86    phi=unwrap(phi);
87    phi=rad2deg(phi);
88    N=ly;
89    f=[0:N-1]/N/Ts;
90    w=[0:N-1]*2*pi/N/Ts;
91    figure
92    SUBPLOT(2,1,1), semilogx(f,mag)
93    title('Plot of FFT(Rxy)/FFT(Rxx)(DSW Response + 1*noise)')
94    xlim([.05 .35])
95    % ylim([-40 20])
96    ylabel('Magnitude')
97    SUBPLOT(2,1,2), semilogx(f,phi)
98    xlabel('Frequency (Hertz)')
99    % xlim([.05 .35])
100   ylim([-200 0])
```

Figure 40. MATLAB Script for PSD Ratios and Reconstructed Bode Plots

The following plots are obtained as a result:

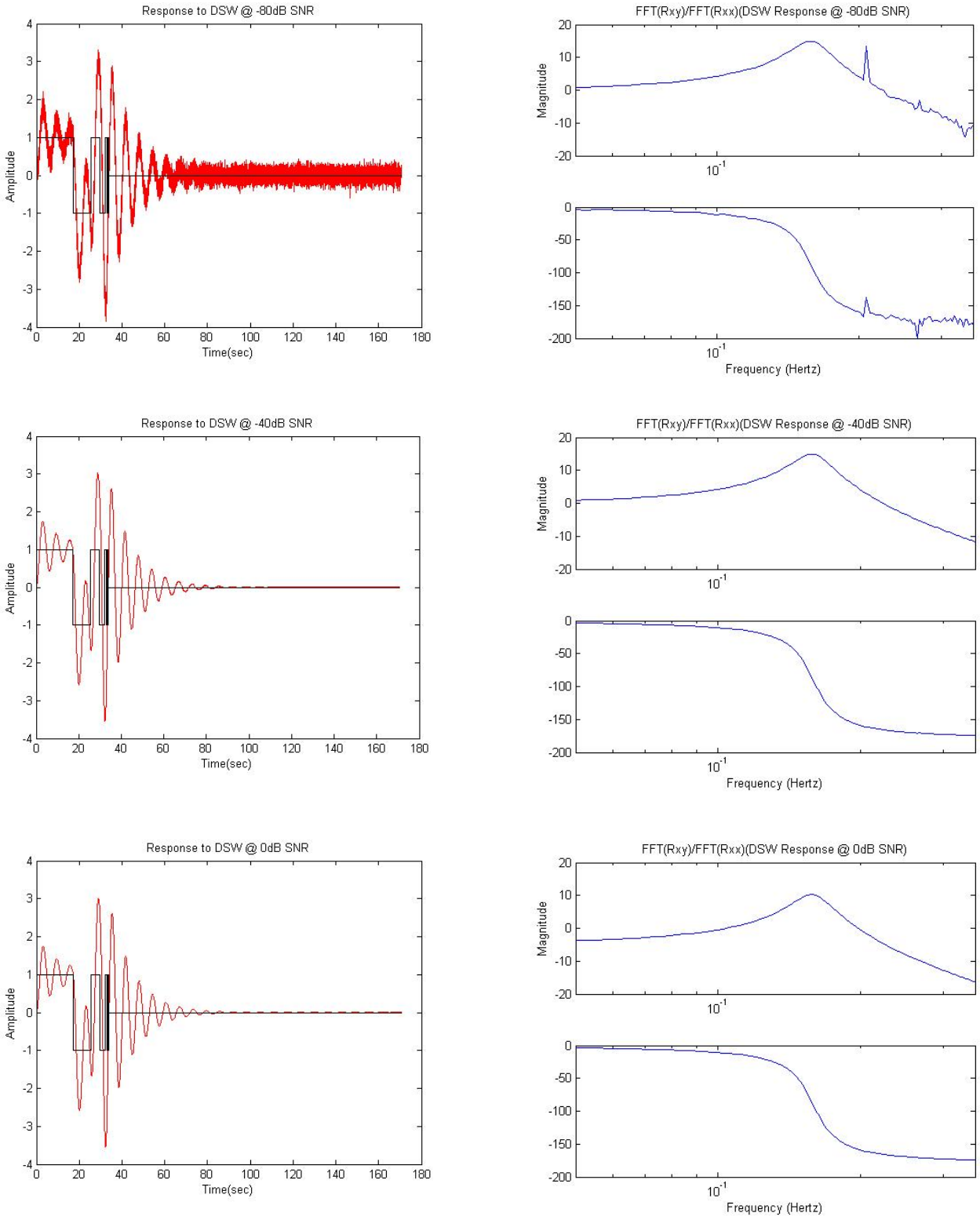


Figure 41. Reconstruction of Bode Plot Using PSD Ratios

Again, with signal-to-noise ratios (SNR) of -80dB, -40dB and 0dB respectively, the reconstructed plots now include representation of the 180-degree phase shift at the corner frequency. The SNR is determined based on the steady state value of .00001206 and the relationship of

$$SNR = 20Log_{10}\left(\frac{Signal}{Noise}\right) \qquad (40)$$

Table 2. Comparison of Two Different Methods for Frequency Range .05Hz and .35Hz

	Corner Frequency (Hz)		**Peak at Corner Frequency (dB)**		**Overall depicted Phase Shift (deg)**		**Phase Shift at Corner Frequency (deg)**	
	FFT/FFT	**Sxy/Sxx**	**FFT/FFT**	**Sxy/Sxx**	**FFT/FFT**	**Sxy/Sxx**	**FFT/FFT**	**Sxy/Sxx**
Clean Signal	0.1582	0.1582	14.93	14.93	170.4	170.52	-86.34	-86.34
SNR=-80dB	0.1582	0.1582	14.93	14.93	19.64	170.42	-10.94	-86.34
SNR=-40dB	0.1582	0.1582	14.93	14.93	27.01	171.39	-10.92	-86.33
SNR=0dB	0.1582	0.1582	15.24	6.166	191.00	170.30	-87.52	-87.53
MATLAB Bode	0.158		14.9		163.2		-85.8	

Table 2 above summarizes the differences between the two different methods with a clean signal, and the three signals corrupted by various levels of noise. The bottom row presents the values as retrieved from the MATLAB generated Bode plot of the original simple, second-order transfer function. Overall, a better quality reconstruction is possible by taking advantage of the Power Spectral Density Ratios. The

improvement over the FFT is seen in the overall depicted phase shift and the phase shift at the corner frequency.

This chapter set out to explain a method by which a known characteristic equation could be recovered to an extent suitable to reconstruct a Bode plot of a control system's frequency response. The appropriate MATLAB code was included to show how each of the plots and its required data was obtained. Throughout this entire setup, a simple transfer function was used to validity of the desired method. The next chapter will progress immediately with the PSD ratios while some backtracking will be necessary so as to highlight lessons learned.

IV. Results & Analysis

In work previously done by Pachter and Barba [1], a controller for a spacecraft mounted flexible antenna was designed. The intent was to attain a pointing accuracy of ±5 μrad/s jointly using feed-forward and feedback control action. Rather than wait for a measure of error, an attempt was made to also use feedforward action to predict the error in advance allowing for a "get it right the first time" approach. The design effort incorporated an assumption of variation that was ±10 percent of various plant parameters to allow for a change in performance due to aging and uncertainties in system identification. The latter is accomplished by feedback action. Six major parameters were considered: the natural frequencies of the flexible modes, the modal peaks at each of the structural modes, the moment of inertia of the antenna's rigid body, time delay, amplifier bandwidth and motor gain. The sponsor of the research has provided a tentative linear model of the antenna based on the following transfer functions:

$$G_{i^{th}\ \text{flexible mode}}(s) = \frac{k_i s}{s^2 + 2\zeta_i \omega_{n,i} s + \omega^2_{n,i}} \tag{41}$$

$$G_{\text{rigid body}}(s) = \frac{1}{Js^2} \tag{42}$$

$$G_{\text{structure}}(s) = G_{\text{rigid body}}(s) + \sum_{i=1}^{20} G_{\text{flexible mode, i}}(s) \tag{43}$$

As illustrated in Figure 4 (Chapter 1), the transfer function $G_{structure}$ is a result of the parallel combination of the rigid body and flexible modes. The end result is

$$G_{structure}(s) = \frac{1.009s^{40} + 23.26s^{39} + 1.305E6s^{38} \quad \cdots \quad 8.482E87s + 3.447E90}{Js^{41} + 911.7s^{40} + 5.131E7s^{39} \quad \cdots \quad 3.347E89s^{2} + 1.36E92s} \qquad (44)$$

The numerical values are obtained from the table in Appendix A. J is the moment of inertia scalar with a value of 39.46 oz·in·s^2. However, this is left as a variable and will be discussed further. The MATLAB script used to evaluate the transfer function of the plant is as follows:

```
1   J=39.46*1;
2   G_rigid_body=tf(1,[J 0]);
3   k=.001*[-1.5597 .0242 .0255 .0189 .0720 .2990 .0909 .0980 .0419 .1288 .0808 .0287 .0784
4   .1793 .0642 .0617 .0408 .01367 .0676 .3620];
5   w_n=2*pi*[3.3 11.5 15.5 16.5 19 21 24 24.5 28 32 34 38 41 43 48 49 56 62 70 76];
6   n=.0025*ones(1, length(k)-1);
7   zeta=[.02 n];
8   G_flex_mode_1=tf([k(1,1) 0],[1 2*zeta(1,1)*w_n(1,1) w_n(1,1)*w_n(1,1)]);
9   G_flex_mode_2=tf([k(1,2) 0],[1 2*zeta(1,2)*w_n(1,2) w_n(1,2)*w_n(1,2)]);
10  G_flex_mode_3=tf([k(1,3) 0],[1 2*zeta(1,3)*w_n(1,3) w_n(1,3)*w_n(1,3)]);
11  G_flex_mode_4=tf([k(1,4) 0],[1 2*zeta(1,4)*w_n(1,4) w_n(1,4)*w_n(1,4)]);
12  G_flex_mode_5=tf([k(1,5) 0],[1 2*zeta(1,5)*w_n(1,5) w_n(1,5)*w_n(1,5)]);
13  G_flex_mode_6=tf([k(1,6) 0],[1 2*zeta(1,6)*w_n(1,6) w_n(1,6)*w_n(1,6)]);
14  G_flex_mode_7=tf([k(1,7) 0],[1 2*zeta(1,7)*w_n(1,7) w_n(1,7)*w_n(1,7)]);
15  G_flex_mode_8=tf([k(1,8) 0],[1 2*zeta(1,8)*w_n(1,8) w_n(1,8)*w_n(1,8)]);
16  G_flex_mode_9=tf([k(1,9) 0],[1 2*zeta(1,9)*w_n(1,9) w_n(1,9)*w_n(1,9)]);
17  G_flex_mode_10=tf([k(1,10) 0],[1 2*zeta(1,10)*w_n(1,10) w_n(1,10)*w_n(1,10)]);
18  G_flex_mode_11=tf([k(1,11) 0],[1 2*zeta(1,11)*w_n(1,11) w_n(1,11)*w_n(1,11)]);
19  G_flex_mode_12=tf([k(1,12) 0],[1 2*zeta(1,12)*w_n(1,12) w_n(1,12)*w_n(1,12)]);
20  G_flex_mode_13=tf([k(1,13) 0],[1 2*zeta(1,13)*w_n(1,13) w_n(1,13)*w_n(1,13)]);
```

```
21  G_flex_mode_14=tf([k(1,14) 0],[1 2*zeta(1,14)*w_n(1,14) w_n(1,14)*w_n(1,14)]);
22  G_flex_mode_15=tf([k(1,15) 0],[1 2*zeta(1,15)*w_n(1,15) w_n(1,15)*w_n(1,15)]);
23  G_flex_mode_16=tf([k(1,16) 0],[1 2*zeta(1,16)*w_n(1,16) w_n(1,16)*w_n(1,16)]);
24  G_flex_mode_17=tf([k(1,17) 0],[1 2*zeta(1,17)*w_n(1,17) w_n(1,17)*w_n(1,17)]);
25  G_flex_mode_18=tf([k(1,18) 0],[1 2*zeta(1,18)*w_n(1,18) w_n(1,18)*w_n(1,18)]);
26  G_flex_mode_19=tf([k(1,19) 0],[1 2*zeta(1,19)*w_n(1,19) w_n(1,19)*w_n(1,19)]);
27  G_flex_mode_20=tf([k(1,20) 0],[1 2*zeta(1,20)*w_n(1,20) w_n(1,20)*w_n(1,20)]);
28  sys1=parallel(G_flex_mode_1,G_flex_mode_2);
29  sys2=parallel(sys1,G_flex_mode_3);
30  sys3=parallel(sys2,G_flex_mode_4);
31  sys4=parallel(sys3,G_flex_mode_5);
32  sys5=parallel(sys4,G_flex_mode_6);
33  sys6=parallel(sys5,G_flex_mode_7);
34  sys7=parallel(sys6,G_flex_mode_8);
35  sys8=parallel(sys7,G_flex_mode_9);
36  sys9=parallel(sys8,G_flex_mode_10);
37  sys10=parallel(sys9,G_flex_mode_11);
38  sys11=parallel(sys10,G_flex_mode_12);
39  sys12=parallel(sys11,G_flex_mode_13);
40  sys13=parallel(sys12,G_flex_mode_14);
41  sys14=parallel(sys13,G_flex_mode_15);
42  sys15=parallel(sys14,G_flex_mode_16);
43  sys16=parallel(sys15,G_flex_mode_17);
44  sys17=parallel(sys16,G_flex_mode_18);
45  sys18=parallel(sys17,G_flex_mode_19);
46  sys19=parallel(sys18,G_flex_mode_20);
47  sysT=parallel(sys19,G_rigid_body);
48  sys=sysT
49  figure
50  bode(sys)
    P = bodeoptions; % Create plot with the options specified by P
    P.FreqUnits = 'Hz'; % Set phase visiblity to off and frequency units to Hz in options
    figure
    h = bodeplot(sys,P);
```

Figure 42. MATLAB Script for Transfer Function

Also in the code is the “Bode” command with a modification to show the frequency in Hz. The following three plots are generated by the same transfer function but differ by varying the value of *J*. The first Bode plot has *J*=39.46 oz·in·s^2; the second has *J*=394.6 oz·in·s^2; and the third is with *J*=3946 oz·in·s^2.

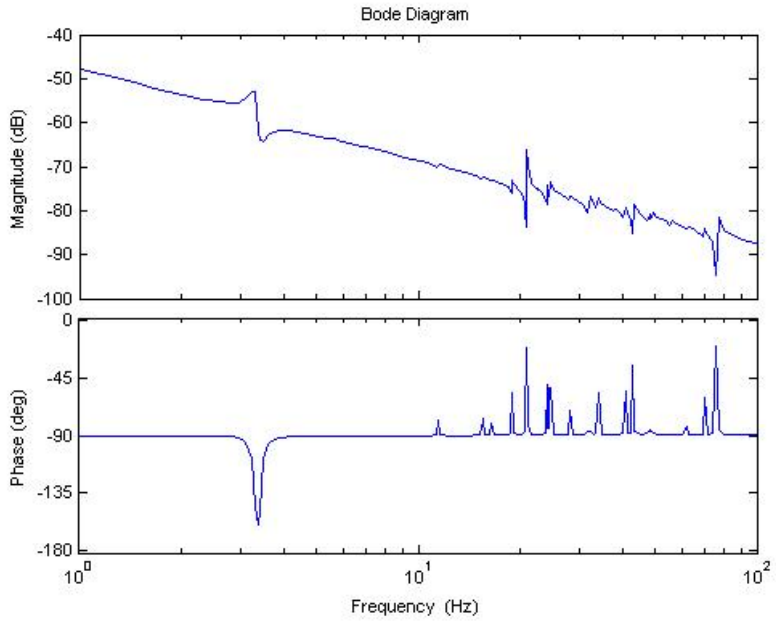


Figure 43. Bode Plot of Complete Control System, J= 39.46 oz·in·s^2

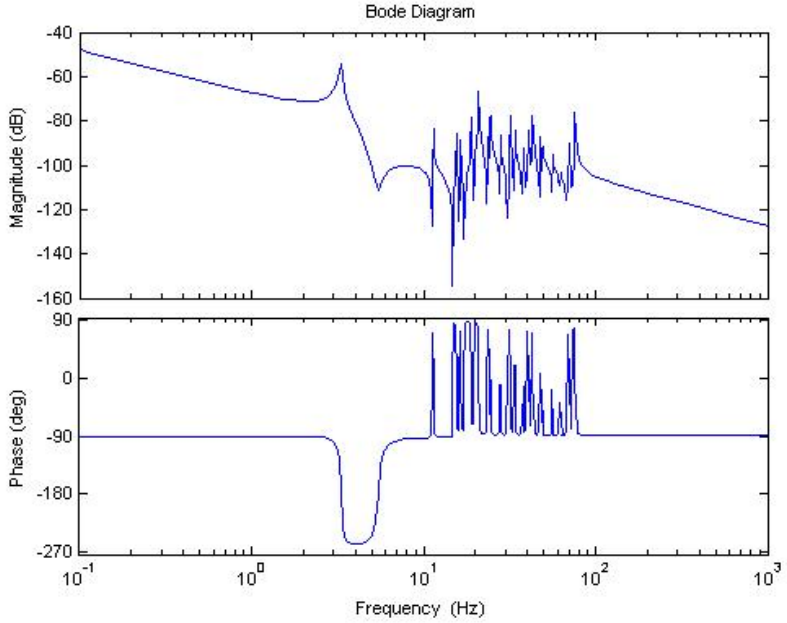


Figure 44. Bode Plot of Complete Control System, *J*= 394.6 oz·in·s^2

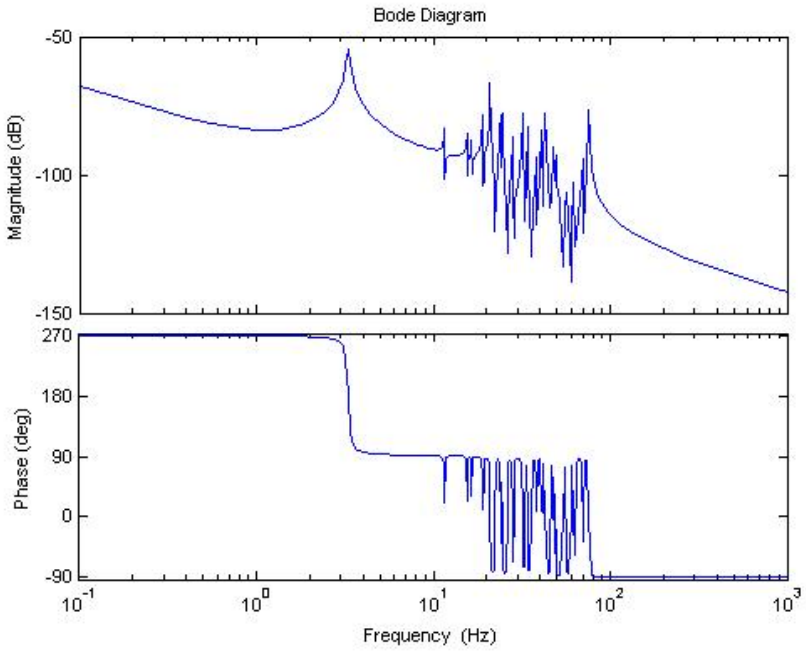


Figure 45. Bode Plot of Complete Control System, J= 3946 oz·in·s^2

The reason for varying the value of J is an attempt to systematically show different levels in the ability to reconstruct the desired Bode plot. A problem was encountered when working with a large value of $G_{rigid\ body}$. However, as the value of J was increased, the value of $G_{rigid\ body}$ decreased and became less influential. Referencing the above Bode plots, a noticeable similarity among all three plots is the level of output signal provided by the plant. In all cases, the ratio of output to input is less than -40dB which more clearly stated implies that a unity input at t=0$^+$ results in an output 1/100th the amplitude, and from there it only decreases! Also, in each plot there is an obvious region of response that falls between 3 and 80 Hertz. This is the basis for defining a window of observation. The reconstruction will focus on this region alone. By taking a closer look at the figure where J=394.6, the individual modes in the response are more easily recognized.

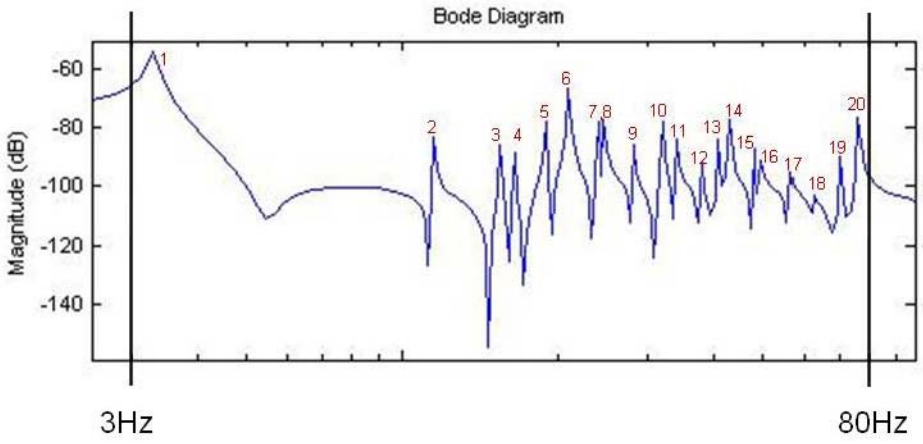


Figure 46. Window of Observation for Magnitude

Each of the 20 modes match that which has been given by the sponsor and will serve as a benchmark with which to compare future system identification results. The phase shift plot also shows a response at each mode.

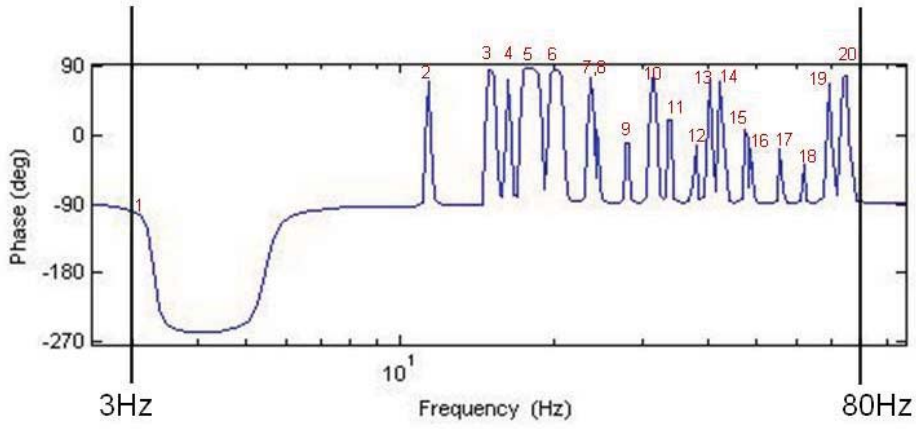


Figure 47. Window of Observation for Phase Shift

As was discussed in the previous chapter in connection with the simple transfer function, a period needs to be established to ensure a balance of capturing the response through its complete return to steady state while limiting any additional data that might serve to dilute the statistical analysis of the response. The following MATLAB code details the process of exciting the plant with a decreasing square wave and then finding the settling time based on a final value. For a quick look, an impulse response plot is made and the settling time is determined using the following script. (An arbitrary time of 50 seconds is chosen to start.) Of particular note, it must be emphasized that the standard value for the settling time of ±2% of the final value was reduced by a factor of 100 due to the extremely small final value.

```
1   %% Impulse Response
2   figure
3   impulse(sys,50)
4   [I,t]=impulse(sys,50);
5   %% Settling time
6   disp('Settling time');
7   disp('--------------------');
8   % figure %create new figure
9   final = I(length(I));% final value - careful!! check t
10  text_out = sprintf (' Final Value = %12.5f', final);
11  disp(text_out); % display output to screen
12  s=length(I);
13  while I(s)<(1.0002*final) & I(s)>.9998*final;s=s-1;end %Compute Settling Time
14  text_out = sprintf('Settling Time = %12.5f', t(s));
15  disp(text_out); % display output to screen
```

Figure 48. MATLAB Script for Impulse Response and Settling Time Computation

The result here is “Settling time=16.03sec”. Another run of the same script but altering the time in line 3 from 50 seconds to 20 seconds verifies the same settling time for a value of steady state ±0.02%.

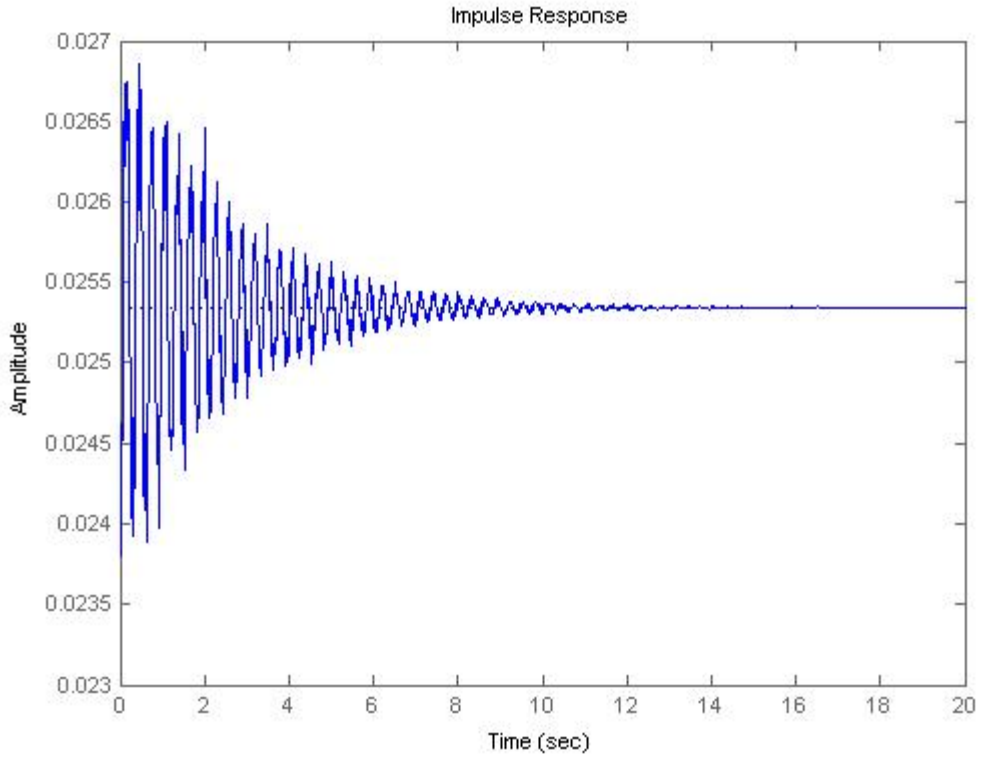


Figure 49. Output of Impulse Response of Complex System

However, this timing must be adjusted for the Decreasing Square Wave (DSW). Again, while it is true that the characteristics of a plant may be characterized solely by its response to a unit impulse as depicted above, the necessity for an alternate, effective input lies in the inability to transmit an "impulse" to control the antenna. The DSW is an input that can be controlled and utilized. When using the DSW, the time of the plant should be offset proportionately by the length of the input signal as depicted in Figure 50.

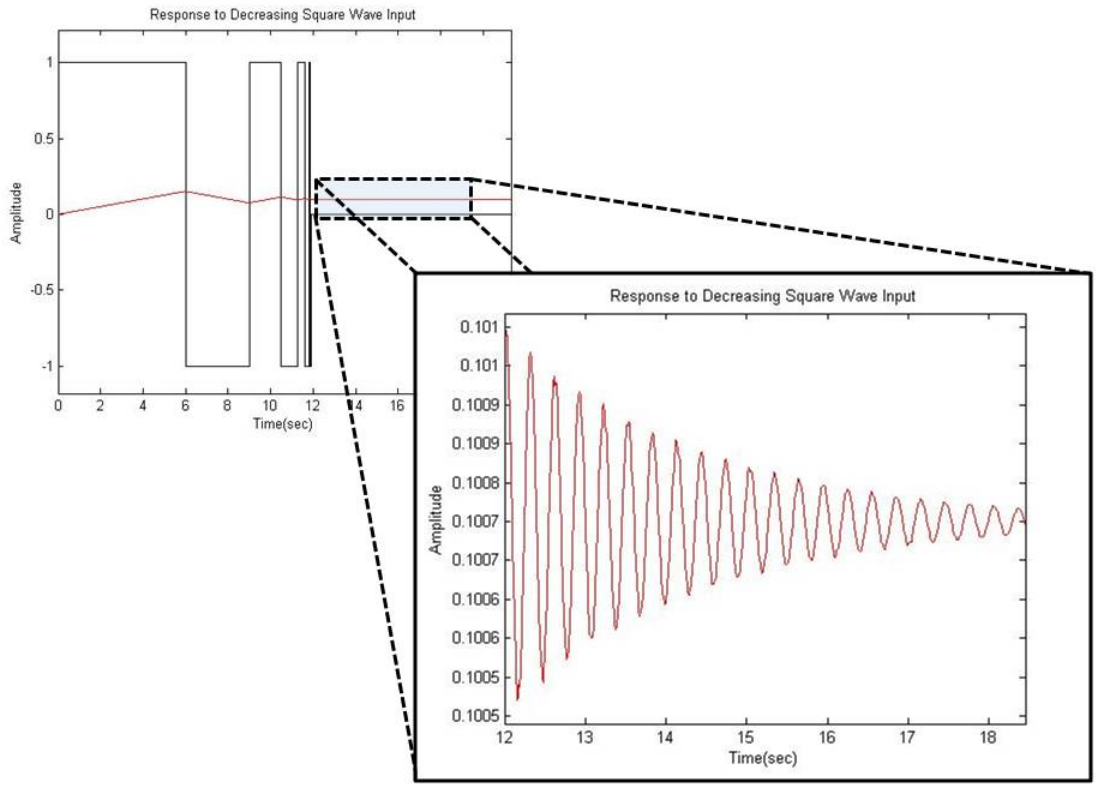


Figure 50. Complete Control System Response to Decreasing Square Wave

The settling time is determined to be 18.07sec. It's important to note that the "Lsim" plot from which this was determined was considered over Δt=60secs. It's logical to ask why, if after an input over a period of approximately 12 sec there is only an increase of 2 seconds to achieve a steady state as compared to an ideal impulse response. Note that while an ideal impulse will have an infinitely narrow width combined with infinite height, the DSW oscillates between -1 and +1 with increasing frequency. In the particular algorithm used to generate this input signal, the natural progression of the response to an input is interrupted by an opposing signal thereby acting somewhat as a damper. Instead of a final "ping" allowing the response to propagate and fade, the plant

is excited by two final opposing inputs, thus providing a controlled excitation and hence only a slight increase is settling time.

The number of samples (N) has been carried over from chapter 3 where N=2^15. In line with the following relationship,

$$N \bullet T_s = T_{total}(\text{sec}) \tag{45}$$

The sampling time required to observe a 60 sec window is 0.00183 seconds. And in accordance with the Nyquist sampling rate, the minimum T_s to reconstruct/capture a frequency of 80Hz is

$$T_s(\text{sec}) = \frac{1}{BW} = \frac{1}{2 f_c} = \frac{1}{2(80)} = 0.00625 \text{ seconds} \tag{46}$$

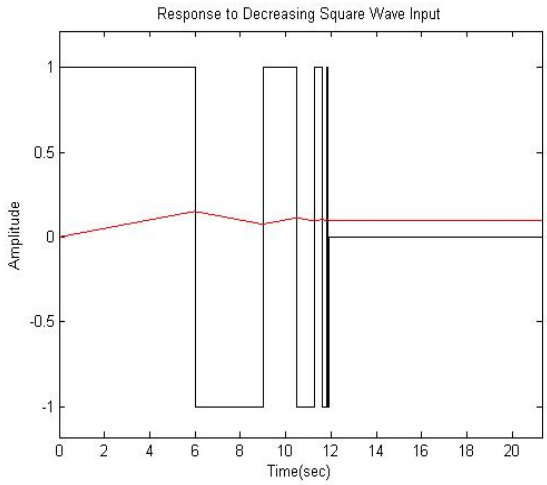
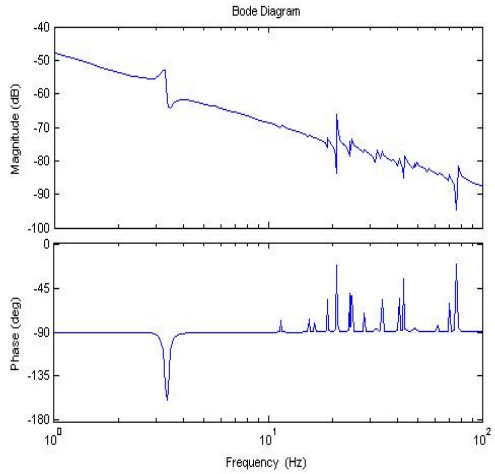
J (oz·in·s^2)	Settling time (sec)
39.46	18.07 sec
	

Figure 51. Settling Time = 18.07 sec

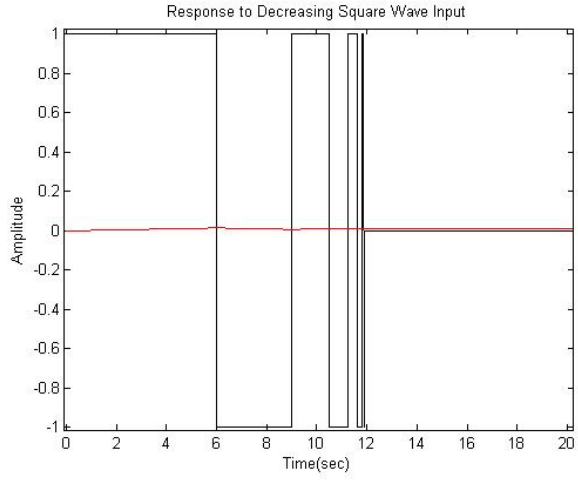
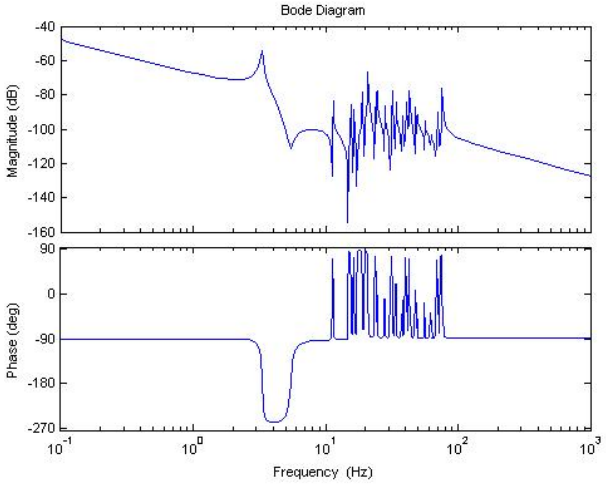
J (oz·in·s^2)	Settling time (sec)
394.6	23.67 sec
	

Figure 52. Settling Time = 23.67 sec

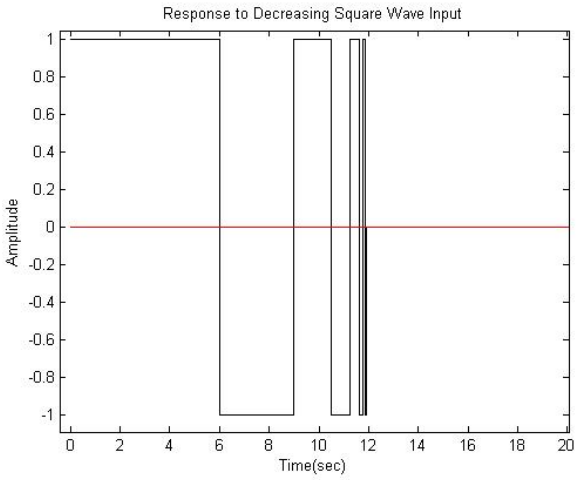
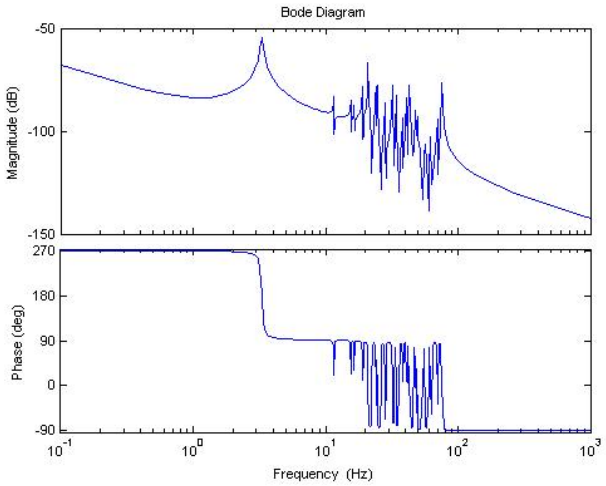
J (oz·in·s^2)	Settling time (sec)
3946	29.61 sec
	

Figure 53. Settling Time = 29.61 sec

Therefore, with the Nyquist minimums met and exceeded, the following variables are defined in MATLAB

```
1   N=2^15
2   fc=80    % the highest frequency (cutoff) we want to include in reconstruction
3   BW=2*fc  % bandwidth (Hz)
4   OS=3.413 % oversampling factor by which we will ensure greater than Nyquist min
5   fs=OS*BW % overall sampling frequency (Hz)
6   Ts=1/fs  % sampling interval (sec)
7   MaxHz=fs/2 %max observable frequency expected in reconstruction (Hz)
8   MaxRads=fs*pi %max observable frequency expected in reconstruction (rad)
9   totaltime=N*Ts %approx 60 secs
```

Figure 54. Parameters assigned in MATLAB

With the variables set and the methodology established in Chapter 3, Figure 55 is generated assuming a clean input and clean output. The intent is to recreate a plot of magnitude and phase both with respect to frequency, similar in fashion to a Bode plot.

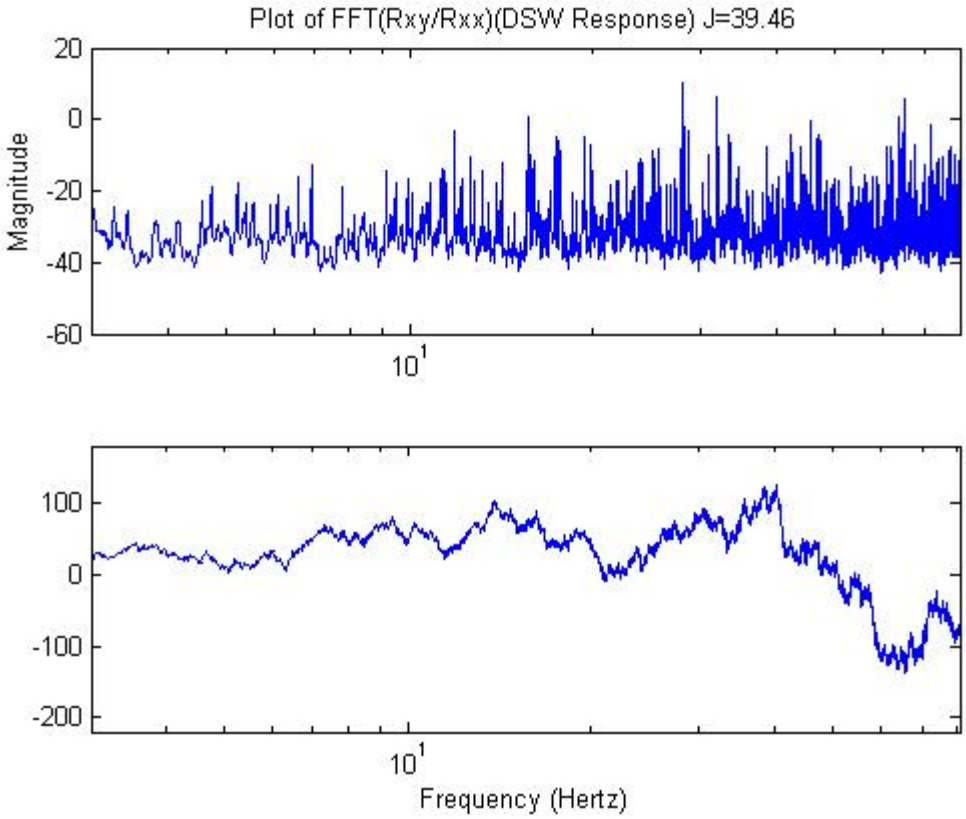


Figure 55. Attempt to Reconstruct Bode when J=39.46 oz·in·s^2

Admittedly, the plot is not too familiar looking; however by increasing J by a factor of 1000 the following plot is obtained:

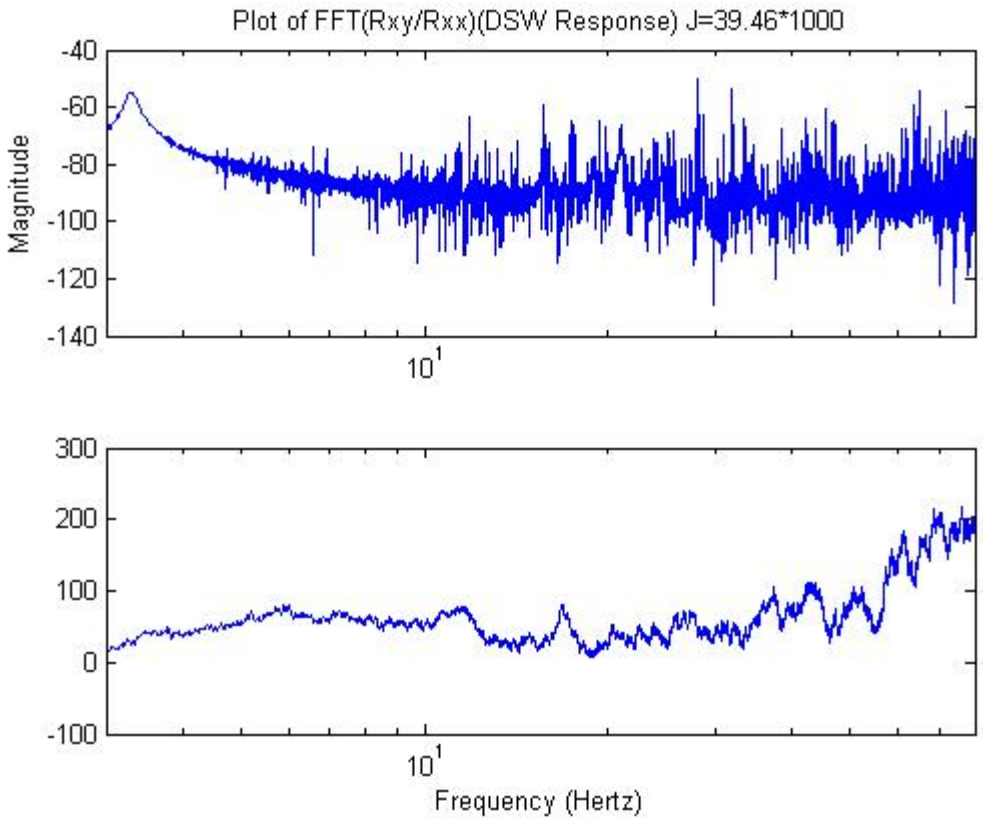


Figure 56. Attempt to Reconstruct Bode when J=39460 oz·in·s^2

The first mode is now recognizable, and by again increasing the J, but this time by an additional factor of 100 where J=39.46E5 the modes become clearer.

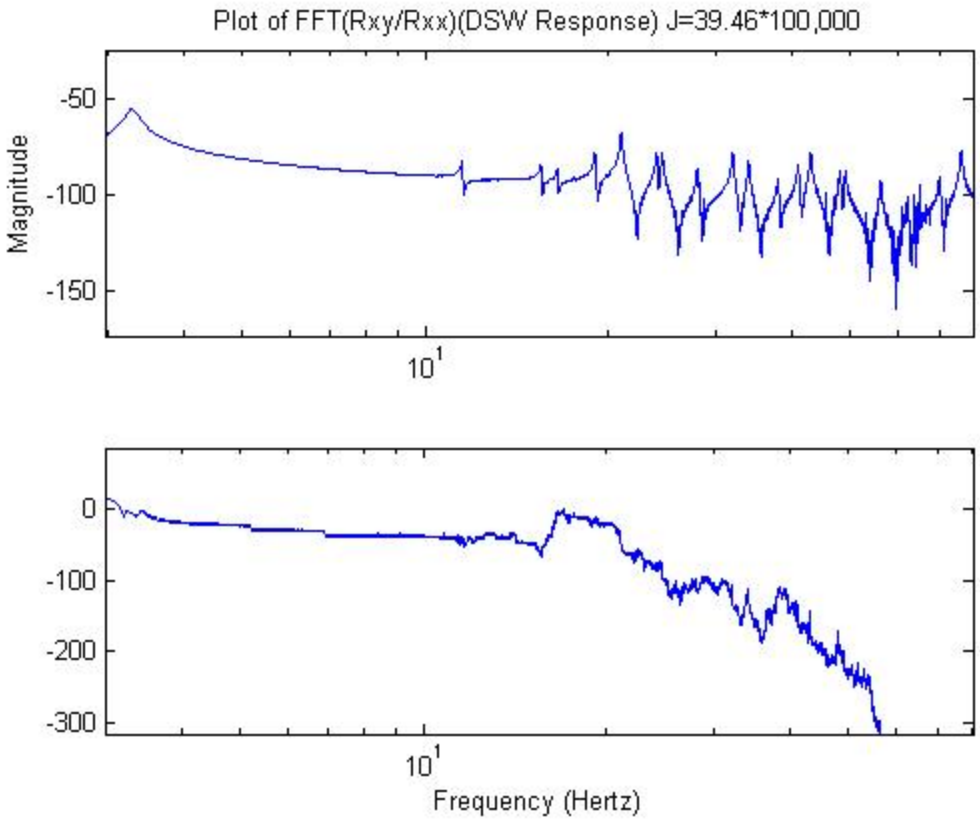


Figure 57. Attempt to Reconstruct Bode when J=39.46 x 10^5oz·in·s^2

Essentially, we have eliminated the $G_{rigid\ body}$ from the transfer function. Reemphasizing, the previous plots were products based on Power Spectral Densities. Figures 58 and 59 highlight the corresponding regions of interest in a plot generated via the direct FFT ratios versus a plot generated via the PSD ratios.

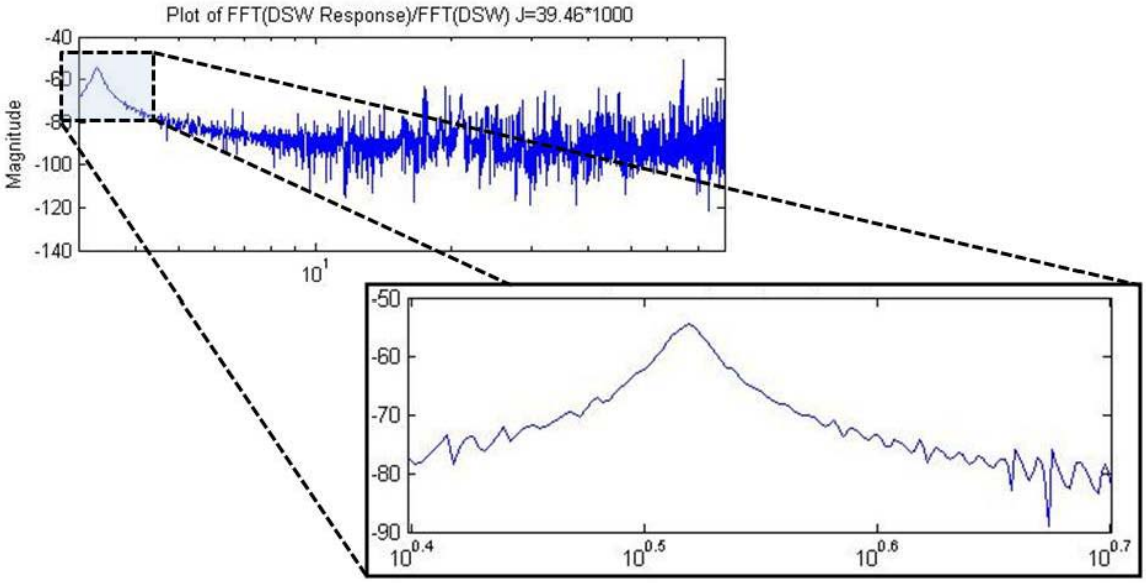


Figure 58. Region of Interest using FFT Ratios

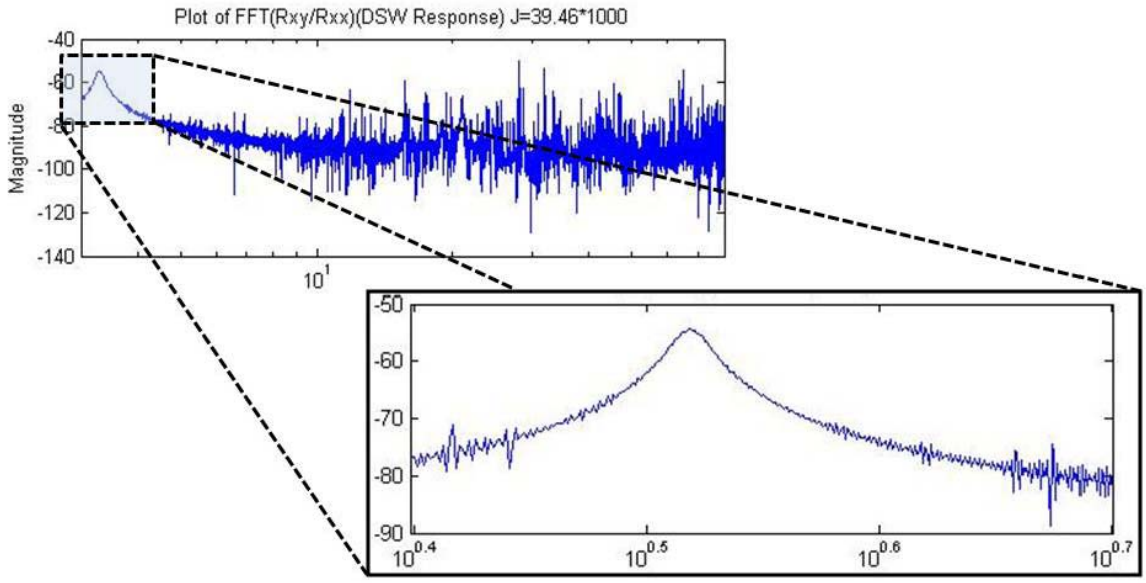


Figure 59. Region of Interest using PSD Ratios

The plot generated via the Power Spectral Density appears noisier. This is a direct result of a two-fold increase in the number of sample points. As denoted in the MATLAB help files, the "xcorr" algorithm will, with a given input vector M, yield a vector 2M as the

correlation process is similar to convolution without the time reversal. With more sampling points, an increase in the ability to reconstruct the Bode plot is anticipated. Thus far an improvement on previous assertions has been made by recognizing that in order to reconstruct the original Bode plot there needs to be a sufficient means to recover the information contained in $G_{\text{rigid body}}$. In this case, it is a function of 1/Js which proves difficult unless the fraction as a whole is very small.

The individual modes themselves prove to be no problem to recognize in a clean environment. The addition of noise will introduce a challenge. Not until the signal to noise ratio establishes itself at 40dB are each of the frequency response modes distinguishable. Based on a final steady state value of 0.00000178, the following reconstructed Bode plots increase in quality in line with an increase in the SNR.

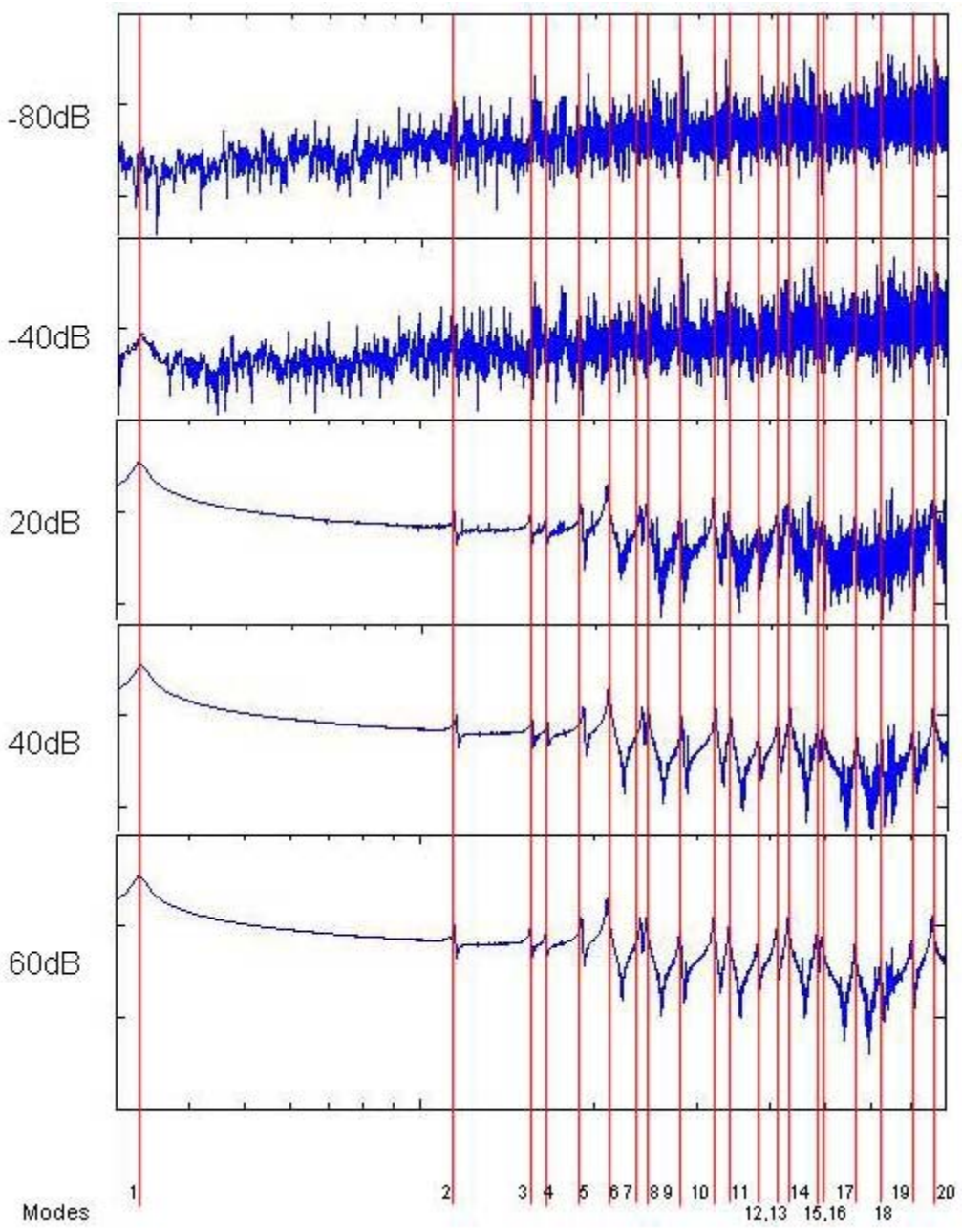


Figure 60. Increase in Recovered Peaks With Change in SNR

Via these series of simulations, a minimum SNR that must be ensured for complete recovery of all 20 modes is 40dB. However, should the tolerance be reduced as to the number of modes required, perhaps to the first three modes, the SNR may be as low at 10dB as shown here

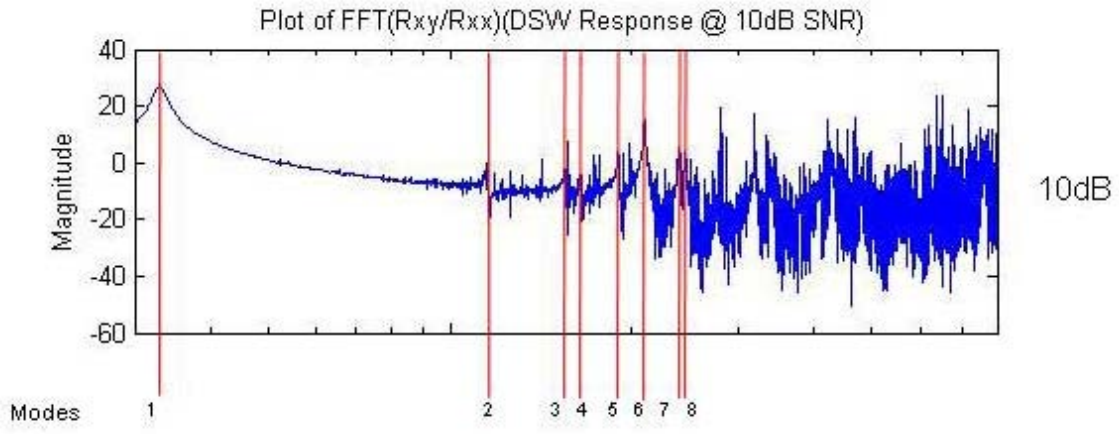


Figure 61. Attempt to Recover Modes 1-3 @ SNR 10dB

Due to the difference in magnitude, the point at which the first three modes are recognizable allows for a bonus inclusion of modes 4-8 as well.

As for the associated phase shifts, with the magnitude recovery established, a focus is placed on phase shift recovery for a SNR greater than 10dB. Figure 62 depicts the phase shifts corresponding to SNRs of 20dB, 40dB, and 60dB respectively each vertically aligned with the MATLAB generated Bode representation of the system's transfer function. The MATLAB plot details a momentary phase shift at each mode, but then returns to a baseline 90° shift while ultimately settling 360° out of phase. The plots that correspond with recovered magnitudes in Figure 60 do not show detail to the extent of a shift at each mode, but rather delineate overall trend. At a SNR=20dB, neither one of the modes can be recovered in magnitude nor the overall phase shift within the window of observation (3-80Hz). The plot shows a change of approximately 180°. An increase in SNR to 40dB yields a recovery of a 300° phase shift and the final plot is more appealing with a start at 0 degrees at 3Hz and 360° at 80Hz.

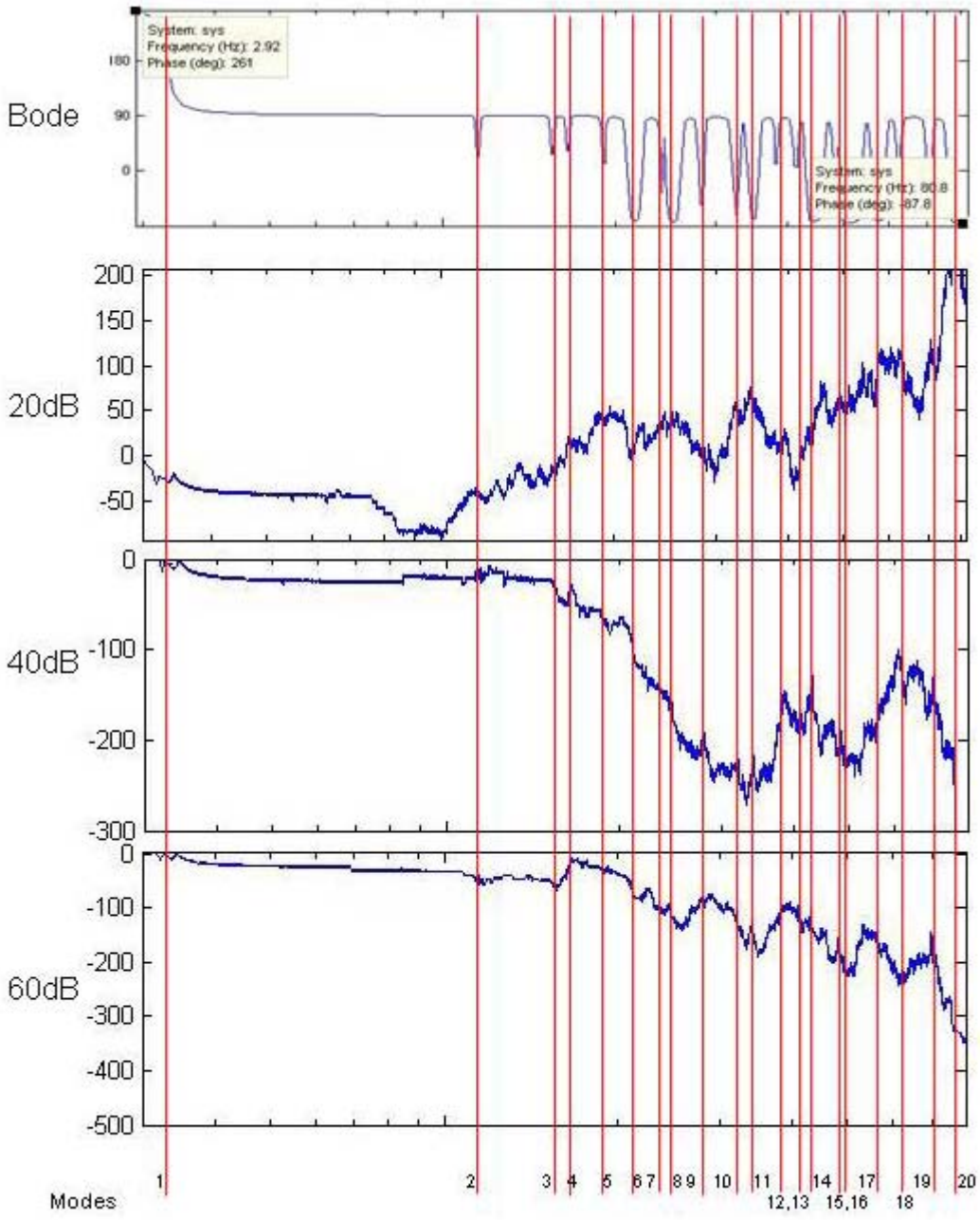


Figure 62. Recovery of Phase Shift With Change in SNR

V. Conclusion

This thesis documents study based on previous work and a set of data provided by the sponsor. Being only part of the end goal, the successful identification of a control system via signal analysis allows the operator to identify changes that have occurred in the system since last evaluated in a controlled environment. Moreover, understanding how the plant is currently behaving, namely, obtaining its transfer function, allows for more precise command input and opens the door for any required controller corrections and/or modifications. It is shown that using the PSD method is possible to identify at least 3 modes provided the SNR≥10dB, and all modes when the SNR ≥ 40dB. In this scenario, more precise plant knowledge will help to facilitate the concept of feed forward controller minimizing the reliance on feedback. While feedback will be used for increased robustness when countering the countless effects of the harsh space environment, the overall control efficiency will be increased by taking advantage of an accurately derived transfer function.

Recommendations for follow-on work

Via the use of Cross Power Spectral Density and PSD, a rough recovery of the Bode plot was achieved but there is a great deal of improvement to be had. More consideration needs to be given to the statistical nature of signal processing. The method of signal reconstruction explored here is hindered by the presence of white noise of any significance and needs to be further considered. Moreover, this thesis considered interference only on the return trip of the signal to be evaluated. There is no reason whatsoever to assume that the command delivered from the controlling station will arrive at the plant clean and unaltered. And this in itself will introduce errors in the command

to which the controller is expected to respond. A deliberate effort needs to be made regarding noise in both directions of signal path travel.

A tandem effort with a laboratory model with actual rate sensor data fed into MATLAB and then further corrupted with various disturbances and noise would add a much needed realism to this study.

Appendix A. Parameters Provided by Sponsor

Table 3. Structure Transfer Function Properties

Flexible Modes	***k*(rad/oz-in x10^{-6})**	**ω_n(Hz)**	**ξ**
1	-1.5597	3.3	.02
2	.0242	11.5	.0025
3	.0255	15.5	.0025
4	.0189	16.5	.0025
5	.0720	19	.0025
6	.2990	21	.0025
7	.0909	24	.0025
8	.0980	24.5	.0025
9	.0419	28	.0025
10	.1288	32	.0025
11	.0808	34	.0025
12	.0287	38	.0025
13	.0784	41	.0025
14	.1793	43	.0025
15	.0642	48	.0025
16	.0617	49	.0025
17	.0408	56	.0025
18	.01367	62	.0025
19	.0676	70	.0025
20	-.3620	76	.0025

Bibliography

[1] Barba, Victor M., ENS, USN. (2007). Controller Design for Accurate Antenna Pointing Onboard a Spacecraft. Unpublished Masters in Aeronautical Engineering, Air Force Institute of Technology, Wright Patterson Air Force Base.

[2] Cooper, Brett J., Capt, USAF. (2009). Rigidizable Inflatable Get-Away-Special Experiment (RIGEX) Post Flight Analysis, Ground Testing, Modeling, and Future Applications. Unpublished Masters Astronautical Engineering, Air Force Institute of Technology, Wright Patterson AFB.

[3] Six Degrees of Freedom Inertial Sensor ADIs16360/ADIS16365I Analog Devices I. (Ed.), Analog Devices, Inc. Retrieved 9 Mar 2009, from http://www.analog.com/static/imported-files/data_sheets/ADIS16360_16365.pdf

[4] VanDyke, M. C., Schwartz, J. L., & Hall, C. D. (2004). Unscented Kalman Filtering for Spacecraft Attitude State and Parameter Estimation.

[5] Leon-Garcia, A. (1994). Probability and Random Processes for Electrical Engineering (Second ed.) Addison-Wesley Publishing Co.

[6] Julier, S., & Uhlmann, J. (2002). The Scaled Unscented Transformation. Proceedings American Control Conference, , 4555-4559.

[7] Wilson, E., Sutter, D. W., Berkovitz, D., Betts, B., J., DelMundo, R., Kong, E., et al. (2003). Motion-Based System Identification and Fault Detection and Isolation Technolgies for Thruster and Controlled Spacecraft NASA.

[8] Schoukens, J., & Pintelon, R. (1990). Measurement of Frequency Response Functions in Noisy Environment. IEE Transactions on Instrumentation and Measurement, 39(6), 905-909.

[9] Oh, S., Kim, J., & Sastry, S. (2004). A Sampling-Based Approach to Nonparametric Dynamic System Identification and Estimation. Proceeding of the 2004 American Control Conference, Boston, MA.

[10] Brown, James M., Capt, USAF. (1995). Optimal Inputs for System Identification. (PhD, EE) Doctoral Thesis.

[11] Chiu, E., Lin, J., McFerron, B., Petigara, N., & Seshasai, S. (2001). Mathematical Theory of Claude Shannon (A Study of the Style and Context of His Work Up to the Genesis of Information Theory). , 15-52.

[12] Guizzo, E. M. (2003). The Essential Message: Claude Shannon and the Making of Information Theory. Unpublished Masters in Science Writing, Massachusetts Institute of Technology.

[13] Bracewell, R. (2000). The Fourier Transform and its Applications (Third ed.) McGraw-Hill.

[14] Oppenheim, A. V., Schafer, R. W., & Buck, J. R. (1999). Discrete-Time Signal Processing. Upper Saddle River, N.J.: Prentice Hall.

[15] Lathi, B. P. (Bhagwandas Pannalal). (1998). Signal Processing and Linear Systems. New York: Oxford University Press.

Vita

LCDR Christopher Sylvester, USN is a native of Glendale, Arizona. He graduated from the United States Naval Academy with a Bachelor of Science degree in Physics in 1997. Following flight school and his designation as a Naval Aviator, he completed his first tour flying the venerable CH/UH/HH-46 “Sea Knight” with HC-5, Andersen Air Force Base, Guam. Missions were in support of underway vertical replenishment of naval vessels, other deployed operations and search and rescue for the Northern Marianas. Following 3 years in the Pacific, he was designated as a primary instructor pilot in the T-34C “Turbo Mentor”.

After redesignation as an Aerospace Engineering Duty Officer in 2005, LCDR Sylvester completed a staff tour at Headquarters, Defense Contract Management Agency, followed by AFIT. After graduation, he will report to NAS Patuxent River, Maryland for assignment to the MV/CV-22 Program Team.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
18-06-2009	Master's Thesis	June 2008 – June 2009

4. TITLE AND SUBTITLE

System Identification of an On Orbit Spacecraft's Antenna Dynamics

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sylvester, Christopher M., LCDR, USN

5d. PROJECT NUMBER
JON-119

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENG)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GA/ENG/09-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Reconnaissance Office
Attn: Lt Col. Mari Ross, USAF
14675 Lee Road
Chantilly, VA 20151-1715
703-808-2599 rossmari@nro.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
SECAF/ISC/T4

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research presented here is a follow on to previous efforts [Pachter, Barba, 2007] in which a tight control loop was designed to meet performance specifications while minimizing the control gains of a spacecraft mounted flexible antenna. Emphasis is now shifted to system identification in order to increase nominal plant knowledge, estimate plant uncertainty bounds, as well as determine the disturbance band.

Knowledge of the plant dynamics along with the corresponding uncertainty bounds will provide for the design of a control system which meets the specifications (tracking and disturbance rejection) while at the same time employing the lowest possible gain. This in turn is conducive to sensor noise disturbance rejection, avoidance of actuator saturation, and excitation of high frequency modes.

15. SUBJECT TERMS

SYSTEM IDENTIFICATION/POWER SPECTRAL DENSITY/FREQUENCY RESPONSE

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Meir Pachter
U	**U**	**U**	**UU**	97	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636x7247 meir.pachter@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18