2-7-2011

Modeling of a gimbal azimuth drive and simulation of control techniques

Frank Matt Broilo

Frank Mathew Broilo
_____
Candidate

Electrical and Computer Engineering
_____
Department

This thesis is approved, and it is acceptable in quality
and form for publication:

Approved by the Thesis Committee:

_____ ,Chairperson

_____

_____

_____

_____

_____

_____

_____

Modeling of a Gimbal Azimuth Drive and Simulation of Control Techniques

by

Frank Mathew Broilo

B.S., Electrical Engineering, New Mexico Institute of Mining and Technology, 2001

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2010

Dedication

To my wonderful sons Joshua and Julian, and my father, Frank J. Broilo, who have all contributed to this text in ways they may not know.

"If there is no struggle, there is no progress" – Frederick Douglas

Acknowledgments

I would like to thank my advisor, Professor Chaouki Abdallah, for his support, tutelage and time. I would also like to thank Dr. Jerry Alcone for his introduction to modeling and advanced controls.

I would like to thank my wife, Maria, for her encouragement and love. I also need to thank my parents, Carol and Charles Dunsworth. My mom set expectations for my siblings and myself to go to college. My step-father, Charles Dunsworth, gave me advice at a critical time which led to me attending the New Mexico Institute of Mining and Technology.

I also thank the RIEtech Global and Sagebrush Technology management teams who supported this effort from beginning to end. Finally, I would like to thank Dr. Joe Zmuda - without whom this wouldn't be possible.

Modeling of a Gimbal Azimuth Drive and Simulation of Control Techniques

by

Frank Mathew Broilo

ABSTRACT OF THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2010

Modeling of a Gimbal Azimuth Drive and Simulation of Control Techniques

by

Frank Mathew Broilo

B.S., Electrical Engineering, New Mexico Institute of Mining and Technology, 2001

M.S., Electrical Engineering, University of New Mexico, 2010

Abstract

A common problem in developing new gimbal products is predicting performance. At the beginning of the design stage, typically the proposal writing stage, it is critical to be able to anticipate the performance of a design, which may still be very roughly defined. Static performance metrics such as pointing accuracy are easier to predict and can be related to position sensor resolution and compliance in the structure and drive. Dynamic performance metrics such as rate tracking and bandwidth are much more difficult to estimate. These dynamic parameters will serve as the results of this paper.

A gimbal model is usually developed in this early stage of design. A model is built for each axis as cross-coupling is not usually significant. The development of each model is inherently dependent on the integrity of the parameters used. Some

parameters may be easily accessible and clearly defined, such as manufacturer specifications for commercial-off-the-shelf (COTS) components. An example of these would be motors. A motor datasheet will usually include specifications such as winding resistance, torque constant, etc. Some parameters have to estimated such as drive friction, structural rigidity and drive parameters. There is also much less certainty in these estimates due to their dependence on the integrated, final system. Building an accurate and sufficient model of the system is a challenging task.

This thesis developments and validates a single axis gimbal model. Following this is a recursion on the model parameters based upon empirical data. Finally, application of different control laws are evaluated in simulation on the model. First, a classical output feedback law is implemented. Secondly, a state observer is implemented with state feedback. State feedback coefficients are found using both pole placement (PP) algorithms and linear quadratic regulation (LQR) optimal control formulas.

Contents

Contents

List of Figures

List of Figures

List of Figures

List of Tables

Glossary

Terms

RTG RIEtech Global®, LLC

$DAYS$ Dual Axis, Yoke, Stabilized

FRF frequency response function

DSP digital signal processor

dB decibels, $20log_{10}(ratio)$

$COTS$ a commercial-off-the-shelf item

EMF electromagnetic field

FEA finite element analysis, utilized in structural rigidity analysis

LPF low-pass filter

PP pole placement

LQR linear quadratic regulator

ARE algebraic Ricatti equation

Bode Plot   frequency response of a system depicted in both magnitude (dB) and phase (degrees)

Glossary

LMS	least-mean squares performance estimate
TDL	tapped delay line

Notation

$\vec{x}$	vector x
e_r	rate error
$\dot{x}$	derivative of x, with respect to time
$\mathbf{A}$	matrix
$\dot{\vec{x}}$	vector holding time derivatives of x
$\mathbf{A}^n$	matrix raised to the nth power
$\mathbf{H}^{-1}(0)$	inverse of transfer function evaluated at 0 Hz
$\mathbf{A} \in \mathbb{R}^{nxm}$	matrix resides in the set of real numbers, of dimension n by m
$\hat{x}$	vector holding estimates of vector x
$\dot{\hat{x}}$	vector holding time derivative estimates of vector x
x'	transpose of vector, x
x_1	first entry in a vector, x
$\hat{\mathbf{B}}$	augmented matrix for state estimator

Symbols

w	angular velocity
α	angular acceleration

Glossary

θ	angular position
τ	torque
J	moment of inertia
Hz	cycles per second
ζ	damping of a complex conjugate pole or zero pair
f	frequency, in Hz

Chapter 1

Introduction

1.1 Overview

Gimbals are pointing devices used in many diverse industries. RIEtech Global®
(RTG) produces gimbals for medical devices, military applications, research labo-
ratories and commercial products. The payloads are typically sensors but can also
include any item which needs to be positioned in a highly accurate manner. Gimbals
can have one to several axes of rotation. Some gimbals have inertial stabilization
which tracks a command with respect to an inertial frame of reference and some
actuate to a very accurate vector with respect to the base of the gimbal.

A newly prototyped gimbal is utilized for this thesis experiment. This gimbal has
been given the moniker DAYS, or Dual Axis, Yoke, Stabilized. This gimbal is shown
in Figure: 1.1. Although the gimbal is designed for stabilization, which involves
high accelerations and bandwidth, the feedback sensors utilized on this gimbal are
actually not inertial sensors, but encoders. Encoders measure axis angular position
with respect to the base.

Figure 1.1: RIEtech Global DAYS gimbal

The architecture of the gimbal's azimuth axis system is shown in Figure: 1.2. This illustrates the components of the system and their interaction. This gimbal azimuth axis serves as the foundation for this thesis investigation.
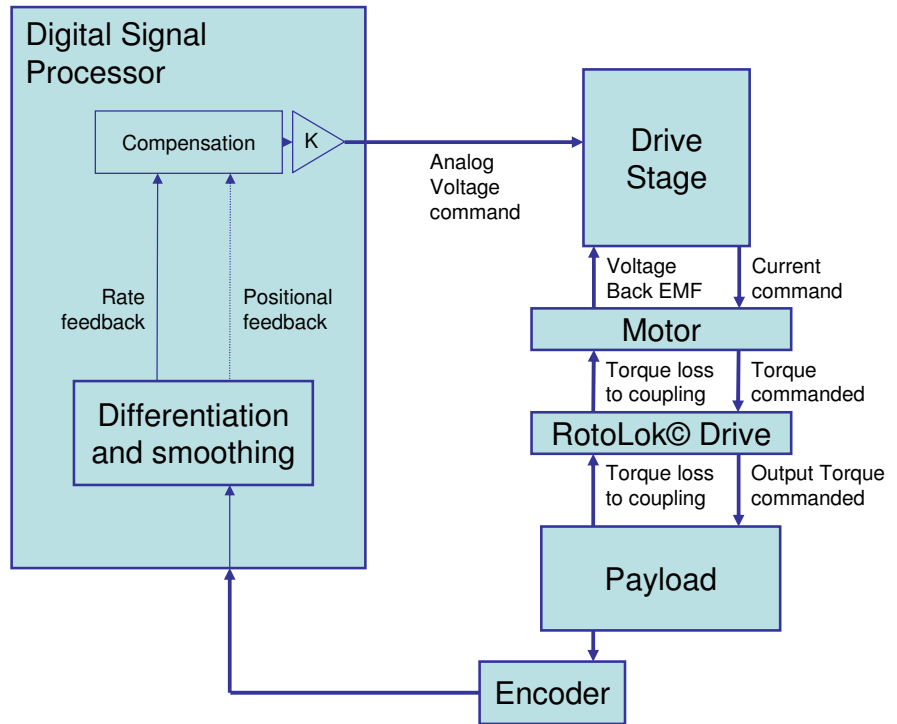

Figure 1.2: Block diagram of single axis system

This investigation addresses the problem statement introduced in the abstract. The problem scope encompasses several aspects of predicting a gimbal's dynamic response. These aspects include: physics-based modeling of the system, incorporating empirical data, and simulating controllers.

In Chapter 2, a physics-based model is derived from the block diagram presented in Figure: 1.2. This chapter also compares the frequency response of the model to the empirical Bode plot of the gimbal azimuth axis. The model is then adjusted to more closely approximate the gimbal and the potential sources of the discrepancies are discussed. In Chapter 3, controllers are applied to the model and compared with the performance of the gimbal. The gimbal software architecture is currently limited to classical output-feedback control. A similar output feedback scheme is simulated on the model, for direct comparison, as well as more advanced controls techniques including state feedback using pole placement and linear quadratic control.

A sampling of relevant papers is presented next. These papers represent contemporary research in the field of gimbal modeling, simulation and controls. The literature review provides both a survey of current similar research and a delineation between current work and the work presented in this thesis.

1.2   Literature Review

Work presented by Skoglar in [1], represents recent and similar research. This research includes the simulation and implementation of an aircraft mounted gimbal which is used to track regions on the ground. This is identical to the intended application of the DAYS system. This research includes development of a gimbal model and simulation of application of LQR control theory.

The gimbal axis modeling portion of this work [1] utilizes a simplified model to

predict the output of each axis. The system modeled is similar to the DAYS axis system, with the exception that a DC brushed motor is used for the drive in [1] and the DAYS system uses brushless torquer motors. The dynamics are fairly similar as well. This work [1] omits the motor inductance, a practice also used in this thesis. Skoglar also omits static friction but uses linear viscous friction. Note that in this thesis static friction and linear viscous friction are modeled and used. The modeling choices made for the DAYS gimbal are discussed further in Chapter 2: Physics-based Mathematical Model. It is difficult to directly compare results between [1] and the DAYS axis simulation as the results provided in [1] are of the final vector tracking and not of the individual axis frequency response or step response.

This work also addresses the application, in simulation, of classical control PID filters and LQR state feedback. Models of the link rigidity are used to indicate compliance in the drive structure. Because of this, Skoglar is able to predict when the actuator is tracking well and when it cannot perform. First, PI, PD and PID control techniques are simulated and are not able to compensate for the compliance in his simulations. The LQR simulation includes the linearization of the gimbal model and the application of state feedback. The model is linearized about a value for each state using the derivative of the state at that value. This model is then placed in a state space representation to facilitate derivation of LQR state feedback coefficients. This is a step which is accomplished using the Matlab function "linmod.m" in this thesis. The results for the LQR tracking are impressive.

In summary, work in [1] envelopes the area of research described in this thesis. However, the scope is applied on a larger scale and does not include as detailed of a model of the axis drive mechanism.

Another source of similar research is [2]. This research includes modeling and simulation of a two axis, stabilized gimbal. This paper focuses primarily on the development of the model, but it does produce some limited results in the form of a

Bode plot. The model development portion is thorough and worthy of discussion.

Two methods of modeling are discussed in [2]: rigid body models and flexible body models. The authors describe rigid body models as having many simplifications. These include assuming that the axes rotate about the same point always and that they are always orthogonal. A notable limitation is that structural modes can limit the bandwidth of a system and should be modeled if possible. This is a key distinction between the work presented in [2] and work presented in this thesis: the RTG DAYS gimbal dynamics which are not accounted for in the physics-based model are input from empirical Bode data. Work in [2] utilizes finite element analysis and some simplifications to feed the model.

Results from [2] reveal a Bode plot of the azimuth transfer function, from applied torque to output position. This transfer function has a typical -40 $\frac{dB}{decade}$ roll off, which represents the two free integrators present in a torque (acceleration) to position transfer function. The structural resonant modes are present at 50 and 80 Hz and do have a large impact on the potential bandwidth. These results are similar to to the results presented in this thesis, but it would be beneficial to see hardware validation of this model.

There is also applicable research in a paper by Maghami, in [3]. This includes advanced techniques for refining FEA analysis of structures. These techniques incorporate eigensystem assignment after system identification has taken place. The system identification process is followed and explained in this paper in the "Motor subsystem" section of the modeling chapter. However, the dynamics found in the DAYS system are either caused by the structure or the drive dynamics. As presented in the modeling chapter, neither subsystem could be directly correlated to the dynamics witnessed. Further application of the techniques shown in [3] may clarify this issue.

A final example of pertinent research is Fu's paper [4] on the topic of LQR systems with input saturation. This paper first illustrates the issue of input saturation in LQR control systems, then current techniques for compensating are shown and finally, a new, analytical, approach is presented. It should be readily apparent that any closed loop control system will saturate the input authority if challenged sufficiently. This will cause a non linearity in the system which negates all performance expectations.

The current approach to dealing with input saturation is anti-windup. This technique was utilized in [1] and involves detuning the optimal controller as the input approaches saturation. This process involves the art of defining invariant sets containing the unsaturated controller and the saturated version. Typically, a sector bound is used to relate the two controllers. Fu's paper presents this and then provides analytical tools for defining the invariant sets and sector bounds.

Input saturation was not checked for the purposes of this thesis paper. It was considered "out of scope." However, research in [4] does constitute a next step in development and would be critical to hardware implementation of LQR in the DAYS system.

1.3   Conclusion

This chapter presented an introduction to gimbals. This includes a functional block diagram which illustrates how the components of the gimbal interact. This was followed by a description of the problem defined in the abstract. Following this is a road map which shows how the problem is addressed in this paper. Finally, a literature review was provided to show relative published work. This chapter should serve as a primer to the work following.

Chapter 2

Physics-based Mathematical Model

2.1 Introduction

A physics-based system model is a powerful tool in many ways. It provides predictions of how the system will act under varying conditions. It allows access to states within the system which may not be directly measurable in the real implementation. This access can be used for troubleshooting or optimization. For example, a state may be saturating because of physical or power supply limitations. If you could witness where the limitation was, then you could make adjustments to alleviate this limitation. Another benefit to having an accurate system model is that it can be used to evaluate controls techniques-as in this thesis. This benefit will be explained further and evaluated in the next chapter. This chapter will develop the system model and qualify the values and relationships used.

2.2   Approach

It is first necessary to create a mathematical model of the system using the laws of physics and both known and measurable parameters. Known parameters and "rule of thumb" estimates are used unless empirical data suggests otherwise. For example, motor characteristics were gathered from the data sheet and used directly in the model. Friction values were both directly measured and estimated. All parameters are appropriately described and justifications are provided as needed. Any physics-based mathematical derivation is also provided in the appropriate subsystem description. The subsystem descriptions will follow a brief presentation of the overall system model.

The system model encompasses the azimuth drive of a RTG custom gimbal, from command generation to sensor feedback. This model is broken down into subsystems as in Figure 2.1 (block diagram) and Figure 2.2 (Simulink model).

2.3   Subsystem Decomposition

These subsystems include:

1) Command generation

2) Motor characteristics

3) Drive, Payload and Sensor parameters

2.3.1   Command Generation subsystem

The command generation subsystem includes compensation and scaling that would take place in the digital signal processor (DSP). Please reference Figure 2.3 for the
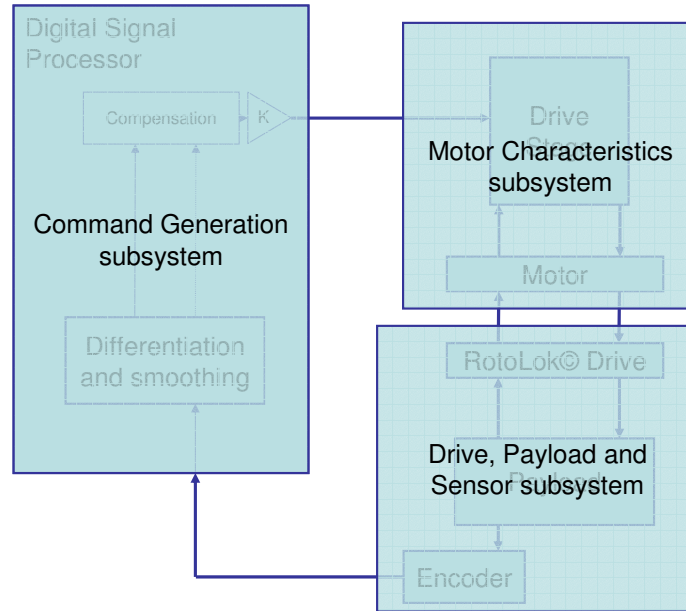
Figure 2.1: Block Diagram of System with Subsystems

Simulink subsystem representation. Included in this blockset is a software switch between a simple rate gain and the compensation filterset. The filterset is used in the simulation of classical controller compensation - which is applied in the next chapter. The rate loop gain parameter is set to unity gain for the modeling effort and used in a later chapter for evaluating the bandwidth of the open loop system. The final block in this subsystem is the "rate to Volts" gain. This block serves to convert the maximum rate commanded into the maximum analog voltage command that can be output. This value is set to $\frac{1.5}{2.5}$ $(\frac{V}{rad/sec})$ and corresponds to the maximum output analog voltage command possible divided by the maximum desired output
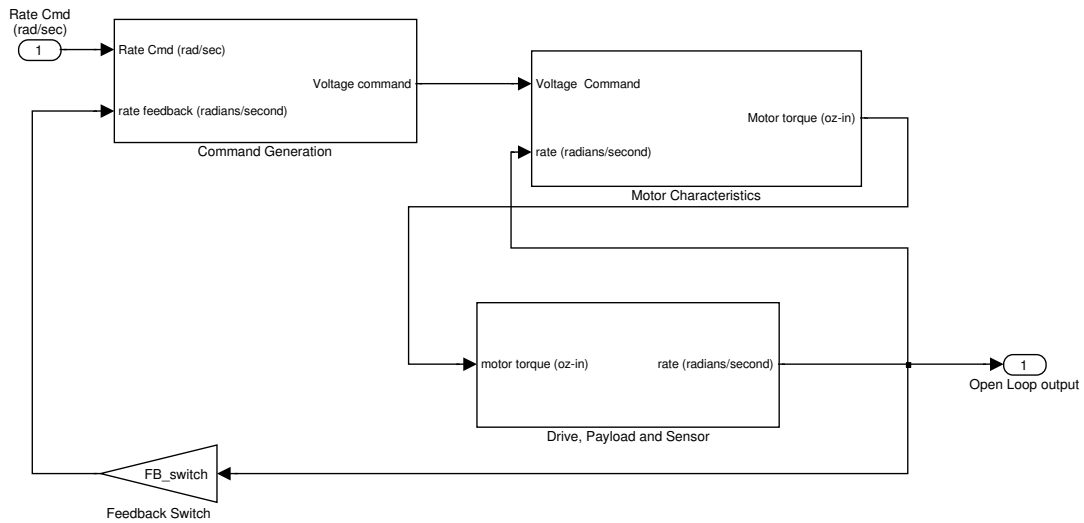
Figure 2.2: Single Axis Block Diagram with Subsystems

rate. This rate was determined based on limitations of the system. This is discussed
in more detail in subsection 2.3.3.

2.3.2   Motor subsystem

The next subsystem represents the drive stage and motor portion of the system. This
is depicted in Figure 2.4. This portion and the third subsystem together represent
the plant. The Servo Amp block is a simplified representation of a COTS motor
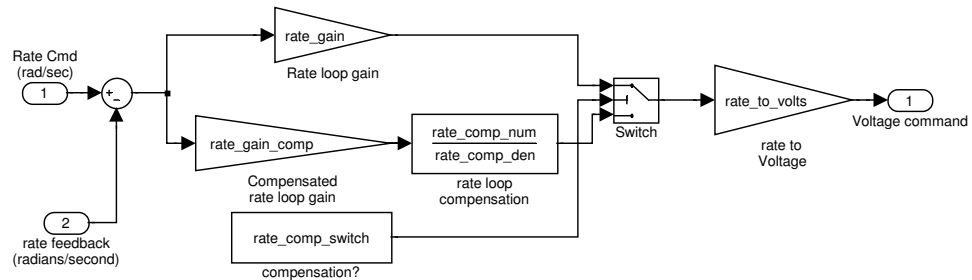driver produced by Advanced Motion Controls (part number DZRALTE-012L080).

Figure 2.3: Command Generation subsystem

This is a digital servo amplifier with internal current loop tuning. The dynamics of the servo amplifier and motor are purposefully omitted from this model. The dynamics of the servo amplifier are set to the static gain of $K_a = \frac{28}{1.5}$ $\left(\frac{V}{V}\right)$. This represents an input output relationship between voltage command from the DSP card and the resulting voltage applied across the motor windings. Following the Servo amplifier output is a saturation block which constrains the voltage output to the actual bus voltage.

The motor dynamics are also purposefully limited to the static gain of $\frac{K_t}{R_m} = \frac{44.5}{5.1}$ $\left(\frac{oz-in/A}{\Omega}\right)$ and a "Motor DC gain" block. This represents the overall static gain
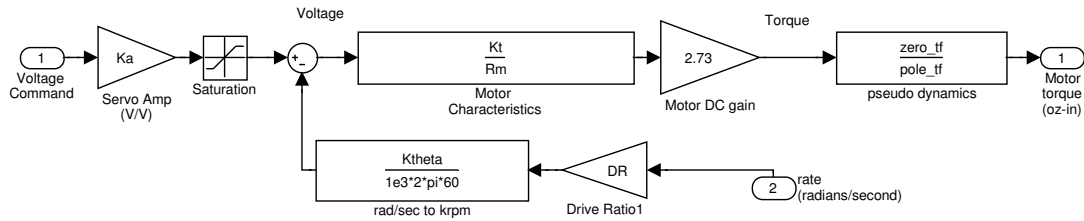
Figure 2.4: Motor Characteristics subsystem

of the motor, at 1 Hz. This can be viewed as a three step process in the plant subsystem. First, the voltage applied across the windings creates a current which is determined by the resistance. Secondly, the motor torque is a product of the motor's specified torque constant, K_t. The second block, "Motor DC gain", is the 1 Hz gain of the motor dynamics when the winding inductance and resulting electrical pole are included. The motor electrical pole location is a known parameter since the winding inductance is specified by the manufacturer (Kollmorgen part number RBE-01811-B00) and $s = \frac{R_m}{L_m}$ $\left(\frac{radians}{second}\right)$, which yields 823 $\frac{radians}{second}$, or 131 Hz, for this case.

13

The frequency domain effects of the omission of the motor pole are shown in Figure 2.5. It is important to note these dynamics, for purposes of being complete, however, the justification for leaving them out is two-fold: they have an insignificant impact on the bandwidth of a typical gimbal axis system, which is 6-15 Hz, and, the inclusion of these dynamics causes difficulties with the advanced controls techniques presented later in this paper. More explanation of this is presented in section 3.4.
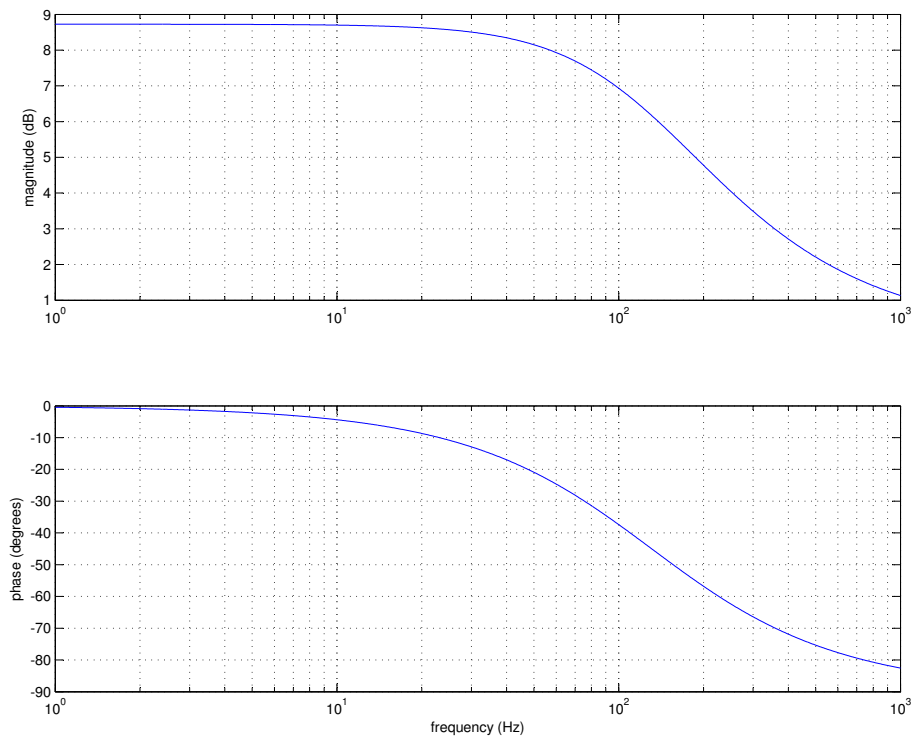

Figure 2.5: Frequency Response of Motor Pole

The remainder of this subsystem includes the back electromagnetic field (EMF) effects of the motor on the supply voltage and a block labeled "pseudo dynamics". The back EMF constant is cited in the motor specification sheet (RBE-01811-B00,

Kollmorgen). The back EMF effects are directly proportional to the rate of the motor output and is typically represented by K_θ ($\frac{V}{krpm}$). The second block, labeled "pseudo dynamics" is necessary to match the frequency response of the model with the recorded gimbal Bode plot. This is a transfer function block with the following frequency response function (FRF):

$$H_{pseudo} = \frac{s^2 + 2\zeta_z 8(2\pi)s + (8(2\pi))^2}{s^2 + 2\zeta_p 60(2\pi)s + (60(2\pi))^2} \tag{2.1}$$

with $\zeta_z = 0.4$ and $\zeta_p = 0.25$. These are the damping factors of each pair. This is a set of complex poles and zeros which are added to give a closer representation of the actual gimbal axis system over the potential bandwidth. Figures 2.6 and 2.7 show the frequency response of the model and gimbal without and with the added dynamics.

The source of these dynamics is theorized to be either in the drive system, a RTG Roto-Lok® drive, or in the structural modes of the payload. Ancillary analysis was done by the RTG mechanical engineering department manager, Santiago Castro (UNM, BSME '04). This analysis used two mechanical engineering tools: a Roto-Lok® analysis tool and Ansys CFX finite element analysis (FEA). First resonant mode of the drive may be responsible for the frequency response shown in the gimbal Bode plot. This is is depicted in a frequency versus torque curve in Figure 2.8. An active range was determined to be between zero and 15 lb-in. This is based upon the maximum current available (5.5 A) and the motor's torque constant, K_t (44.5 oz-in). It is then feasible that the first resonant mode of the RotoLok® is low-enough to be responsible for the added system dynamics. What is not available is the damping on this mode. The FEA analysis of the yoke is similar in nature in that natural frequencies predicted are within the band of interest, but the magnitude and damping is not given. The first four resonant frequencies, and their physical manifestations, are presented in Figure 2.9. The first three modes are within our
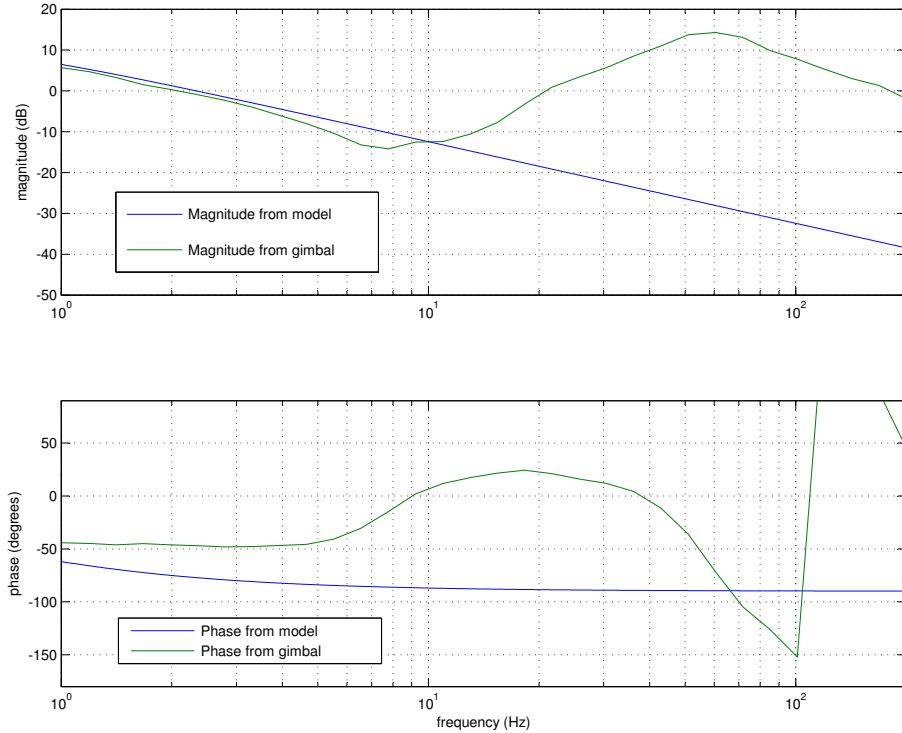
Figure 2.6: Frequency Response of Model and Gimbal without Pseudo Dynamics

frequency band of interest. The first mode, while lower in frequency than the gimbal resonance, could be envisioned as being in the direction that the azimuth drive would torque the yoke. This is a "tuning fork" deformation. Note that the azimuth drive would be on top in this illustration and the yoke would be hanging below it. A payload shelf (coupling) is present in between the arms of the yoke and is hidden in this figure. Further detail into the limitations of FEA structural analysis and a potential solution are provided in [3].
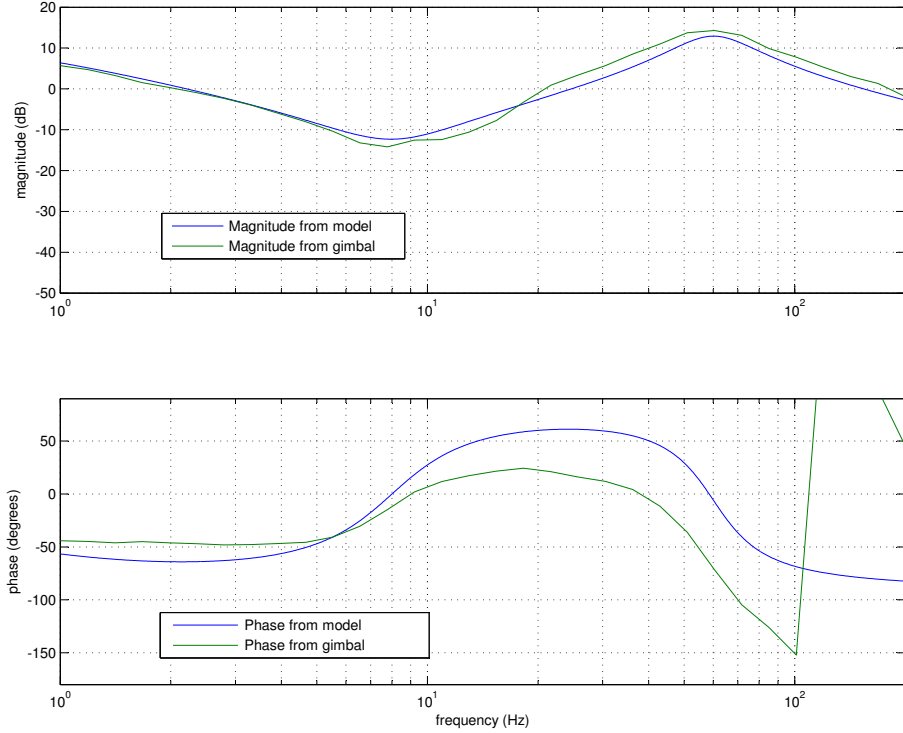
Figure 2.7: Frequency Response of Model and Gimbal with Pseudo Dynamics

2.3.3 Drive, Payload and Sensor subsystem

This subsystem includes frictional components of the drive along with payload inertia and sensor feedback. This is illustrated in Figure 2.10. The frictional model utilizes static friction, Coulomb friction and linear viscous friction. The definitions for these parameters are taken from "Nonlinear Systems", H. Khalil, page 10: [5]. Static friction is directly measured in the gimbal system via a data recording of current command and output position. This value is estimated as:

$$I_c = \frac{|I_{pos} - I_{neg}|}{2} \tag{2.2}$$
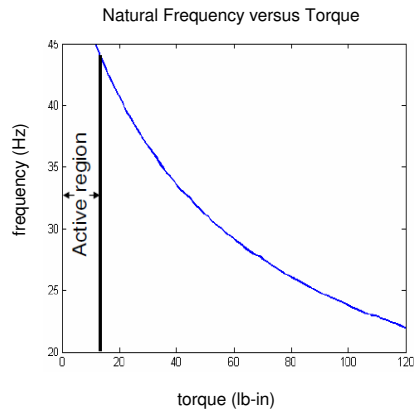
Figure 2.8: Predicted resonant mode of the RotoLok® versus torque.

In this relationship I_c is the percentage of the maximum current command that is required to overcome static friction in both the positive and negative directions. Notice that this doesn't have to be zero-biased. This is due to bias in the current loop which usually provides some low-frequency offset. This measurement for the gimbal azimuth drive is shown in Figure 2.11. Once I_c is established, the corresponding torque applied can be calculated as:

$$\tau = K_t I_c I_{max} \tag{2.3}$$
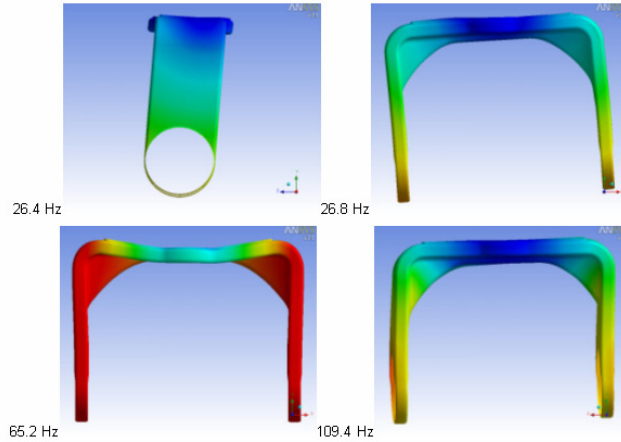
Figure 2.9: Predicted Resonant Modes of the Yoke Structure

in which K_t is the corresponding torque constant specified by the manufacturer and I_{max} is the maximum current allowed in the gimbal configuration. In this case, K_t is 44.5 $\frac{oz-in}{A}$ and

$$I_{max} = \frac{busVoltage}{R_{winding}} = \frac{28V}{5.1\Omega} = 5.5A \tag{2.4}$$

Then, applying 25 percent as I_c to equation 2.3 provides 61.2 oz-in.

Coulomb and linear viscous friction parameters are obtained from the measured static friction value and are strictly "rule of thumb" estimates. The value for Coulomb friction is approximated as $\frac{2}{3}$ of the static friction value. Linear viscous
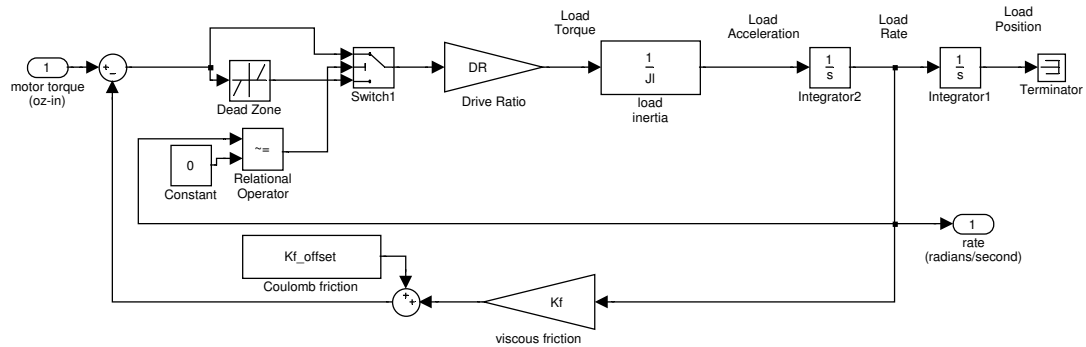
Figure 2.10: Drive, Payload and Sensor subsystem

friction, K_f is approximated as static friction. These three parameters are shown in a friction versus velocity plot in Figure 2.12. Note that static friction is modeled as a deadband block in the Simulink diagram. This value is switched on when $\omega = 0$ and off otherwise. The maximum desired output rate can now be determined based on the back emf effects and this friction model. This is established by commanding the gimbal model to a very large rate and observing the actual rate output. This is illustrated in Figure 2.13. The rate of $2.5(\frac{radians}{second})$ is then selected as a conservatively high value. This value is used in the Command Generation subsection.
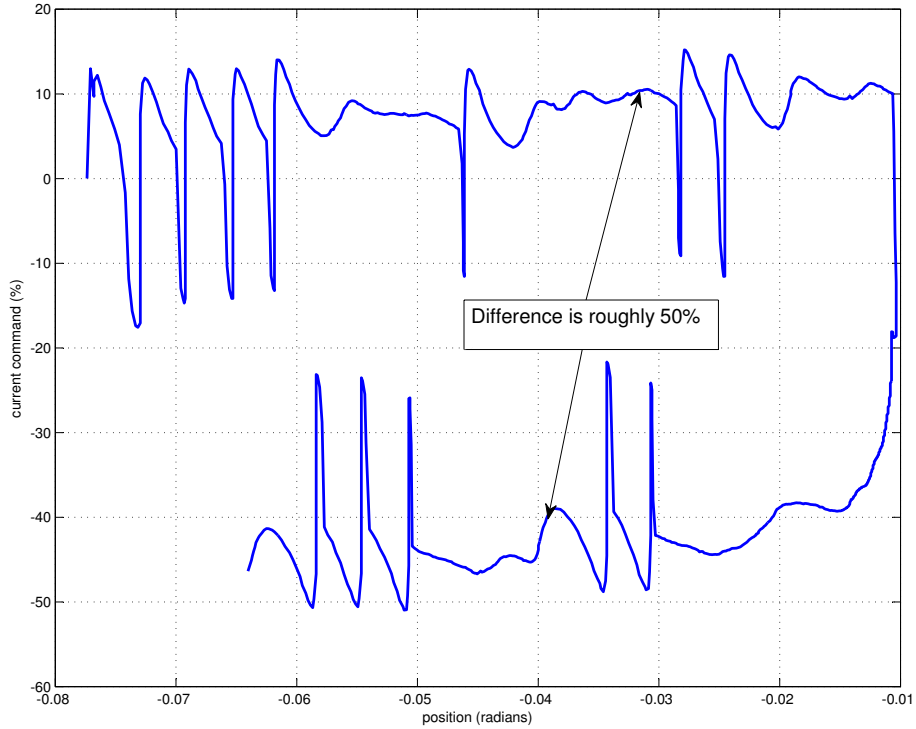
Figure 2.11: Static Friction Recorded in the Gimbal Azimuth Drive

The next portion of this system includes a Drive Ratio block, which merely gains up the torque output of the motor and conversion of torque to acceleration via the load inertia block. This utilizes the relationship:

$$\alpha = \frac{\tau}{J} \tag{2.5}$$

in which α is rotational acceleration, τ is torque and J is the moment of inertia. The moment of inertia estimated by mechanical modeling of the gimbal is 2200 $lb-in-s^2$. Using this number the maximum acceleration available in this axis is $\frac{15.3}{5.7}$ $\frac{lb-in}{lb-in-s^s}$ $= 2.7 \left(\frac{radians}{s^2}\right)$.

Figure 2.12: Friction Effects Modeled in System

It is then easy to see that rotational velocity and position then are:

$$\omega = \int \alpha dt \tag{2.6}$$

and

$$\theta = \int \omega dt \tag{2.7}$$

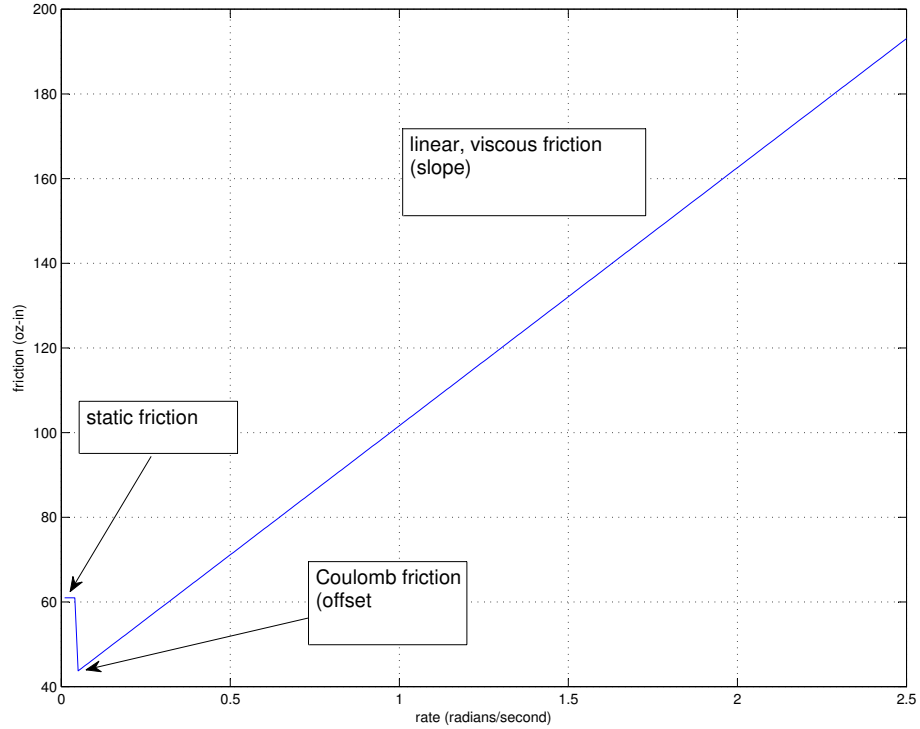in which ω is rotational velocity and θ is output angular position. One obvious omission on this portion of the model is payload to motor output coupling. Any
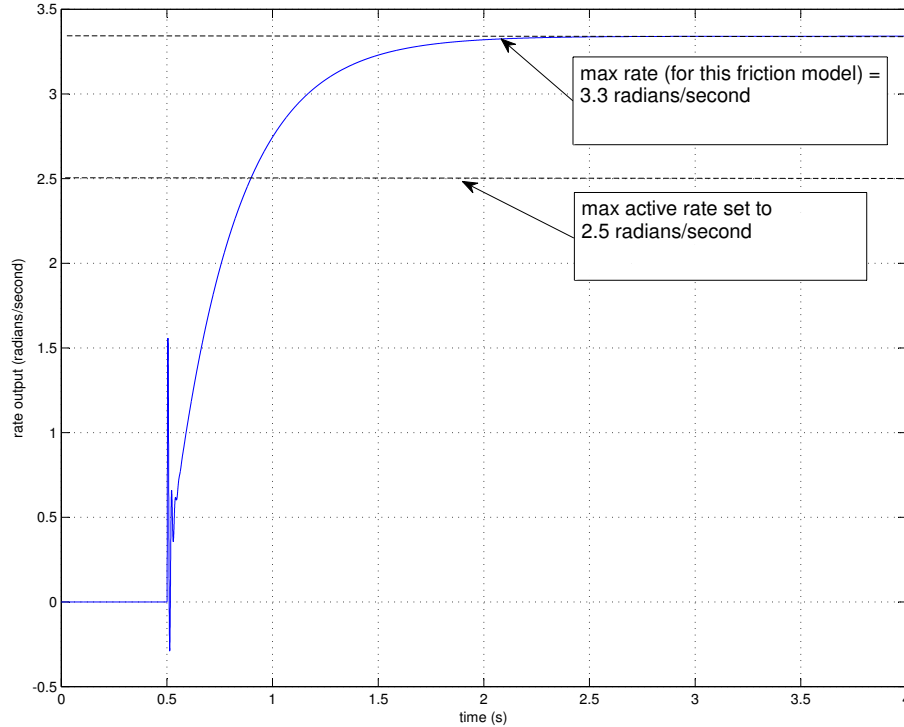
Figure 2.13: Maximum Rate of the Gimbal Drive based on Friction and Back EMF Limitations

compliance or dynamics in the coupling is assumed to be included in the pseudo dynamics.

The final area of this subsystem is the sensor feedback. The gimbal used for this experiment has on-axis positional encoders. In a typical RTG gimbal system, these are fed back as position. The gimbal DSP can then utilize the position, in order to create a closed position loop, or it can differentiate the position to get output rates. For the purposes of this experiment, only the rate loop is being utilized. However, the position output is retained as it will serve as a more direct measurement for

observer input in section 3.4. The encoder-based rate is smoothed using a second order low-pass filter (LPF). The cutoff frequency of this filter is 60 Hz. The encoder position differentiation and smoothing is purposefully left out of the modeling effort as well. The reasons for this are two-fold: first, the encoder position differentiation causes a pole-zero cancellation, which causes issues with the application of advanced controls techniques, and second, the smoothing filter frequency effects are outside of the frequency band of interest. The pole-zero cancellation issue is described in more detail in Chapter 3. The frequency response of the smoothing filter is depicted in Figure 2.14.
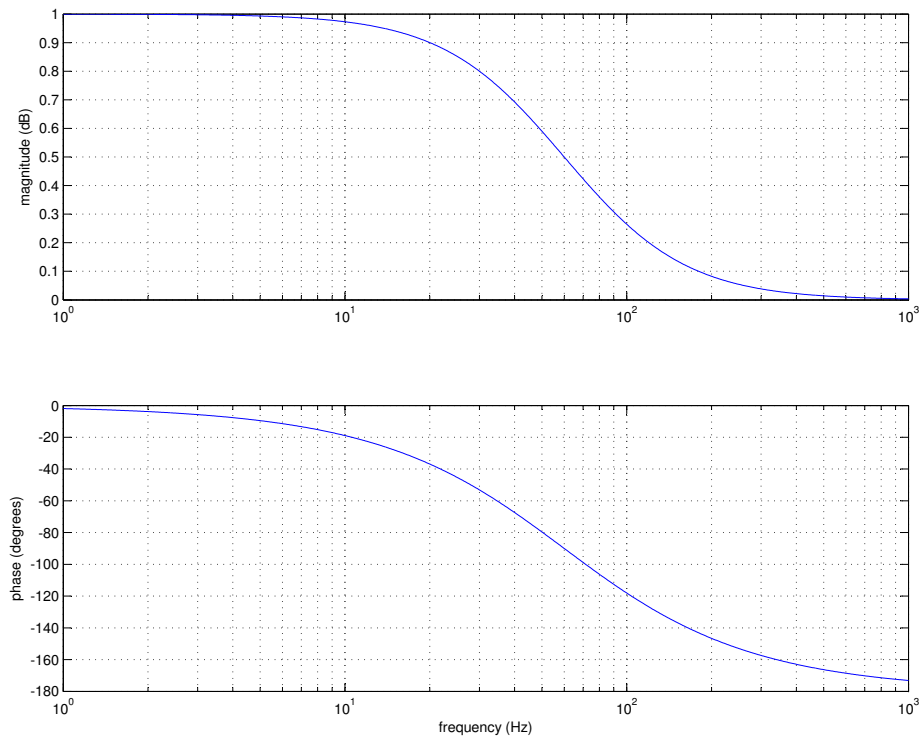
Figure 2.14: Frequency Response of Encoder Smoothing Filter

2.4   Summary

The modeling effort produced a very similar frequency response to the recorded Bode plot of the gimbal azimuth drive. This is presented again in Figure 2.15. The phase does not match as well as the magnitude. Perhaps a higher-order approximation of the pseudo dynamics could match this better. The phase discrepancy appears to be a factor of the damping on the complex conjugate zero pair in the pseudo dynamics block. This model will be used in the next chapter to evaluate different control laws.
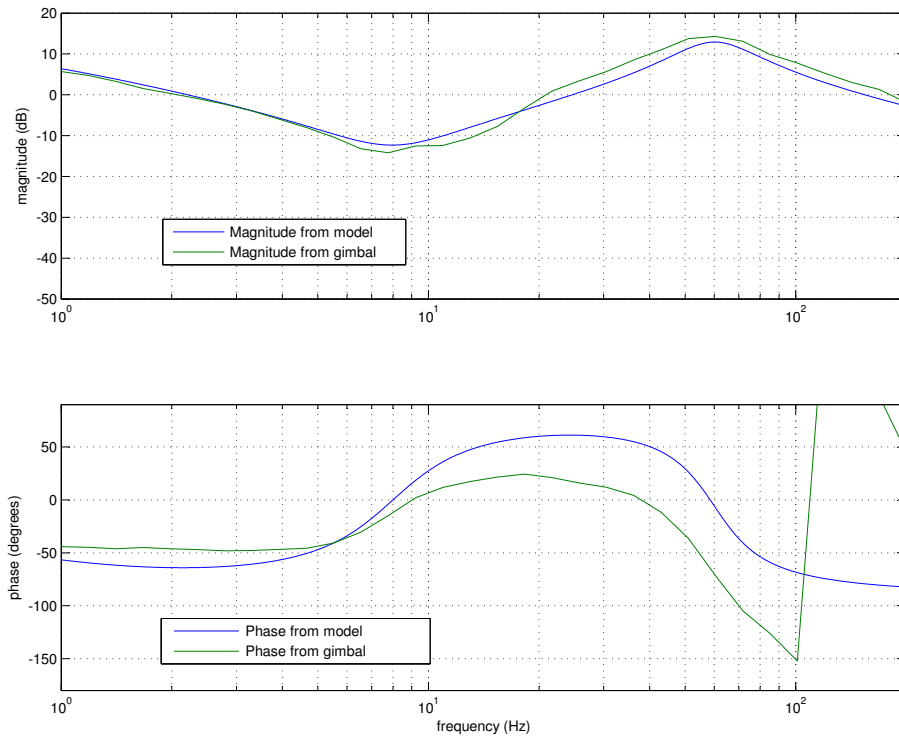
Figure 2.15: Frequency Response of Model and Gimbal with Pseudo Dynamics

Chapter 3

Control Techniques

3.1   Introduction

This chapter focuses on the application of established control algorithms to the gimbal model. The results are then compared to the actual gimbal performance. Performance can be measured in a variety of ways:

1) Bandwidth and phase margin

2) 10 to 90 percent rise time

3) Percent overshoot

4) Rate tracking

The expectation is that the simulated results will be better than that of the actual gimbal. This is based on the ideal nature of the model.

3.2   Overview

The model will now serve as a base for testing these different control laws. Two control laws are tested and compared against the actual gimbal closed rate loop performance. These include output feedback:

$$u = -ky \tag{3.1}$$

This is the law currently implemented in the gimbal. It is also referred to as "classical control". The second control law simulated is state feedback:

$$u = -\overrightarrow{k}\,\overrightarrow{x} \tag{3.2}$$

where $\overrightarrow{k}$ is a row vector and $\overrightarrow{x}$ is a column vector. This inner product then yields a scalar, u, for feedback.

As aforementioned, each of these are established and well documented control techniques. These will be developed as they are applied to the specific gimbal model in the following sections. For more background information, please reference [6] and [7].

3.3   Output Feedback Controller simulation

The gimbal architecture currently provides output feedback control. This is illustrated in Figure 1.2. Output position is directly measured and fed back through the control electronics. This is differentiated and smoothed and becomes an approximation of the output rate. This is then differenced with a rate command signal, v, which yields the rate error, e_r. This error is fed through a compensation filterset which

becomes the motor command. The filterset used on the actual gimbal is referenced in Table 3.1.

These filters are common to gimbal control. The proportional-integral filter adds low frequency response and low frequency disturbance rejection. The lead filter provides additional phase margin at the cross over frequency. The low pass filter drives down noise and any higher frequency modes which are present. The Bode diagram representing this combined filterset is presented in Figure 3.1 and the resulting open loop response is presented in Figure 3.2.
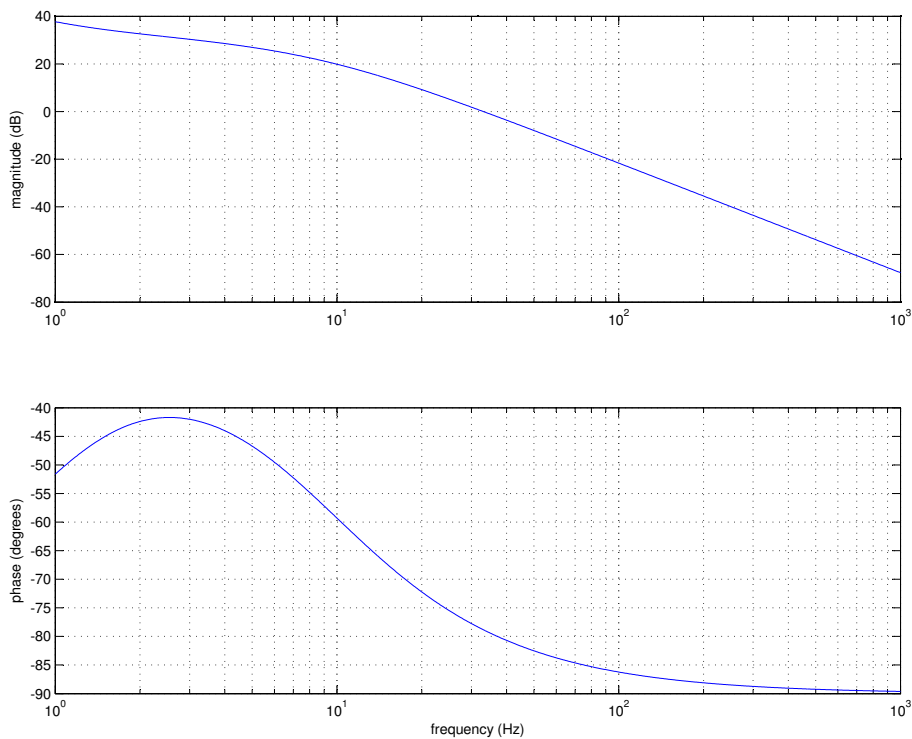
Figure 3.1: Frequency Response of Gimbal Filterset

All of the previously stated performance metrics were utilized to develop this
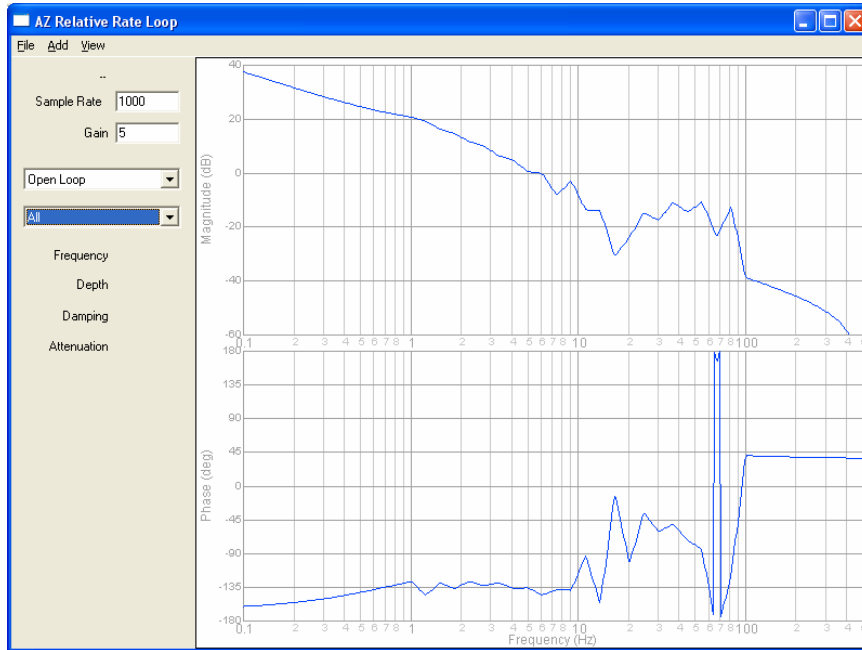
Figure 3.2: Open Loop Bode Plot of Compensated Gimbal Rate Loop

filterset. The open loop Bode plot indicates 6.5 Hz bandwidth and 30 degrees of phase margin. The gimbal step response is presented in Figure 3.3 and shows each of the time history metrics. Some frequency content is visible in the rate tracking inspection of this figure. A portion of this is at 4 Hz and is probably caused by lack of phase margin at cross over. This behavior makes it difficult to provide an accurate rise time. The ripple in the response causes two crossings of the 90 percent line. In these occurrences, the second crossing is used. The frequency content is quantified via Fourier analysis in Figure 3.4.

These results indicate that there is some ring out at the cross over frequency,

29

but there is not any overshoot present. The 90 Hz is theorized to be a sharp high frequency mode. This mode is not apparent in the system Bode plot.
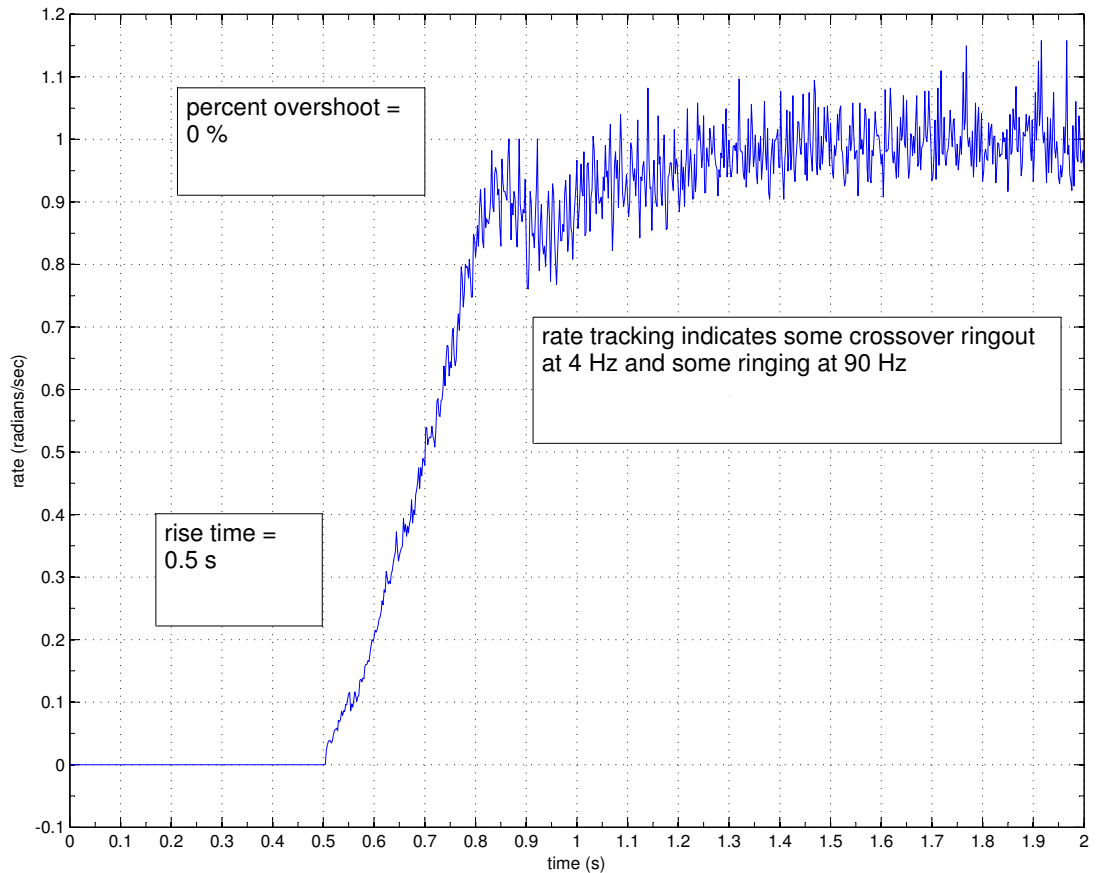
Figure 3.3: Gimbal Step Rate Response

A similar approach is then taken with the Matlab-generated model. These filter parameters are shown in Table 3.1. The gimbal filterset is presented also for comparison.
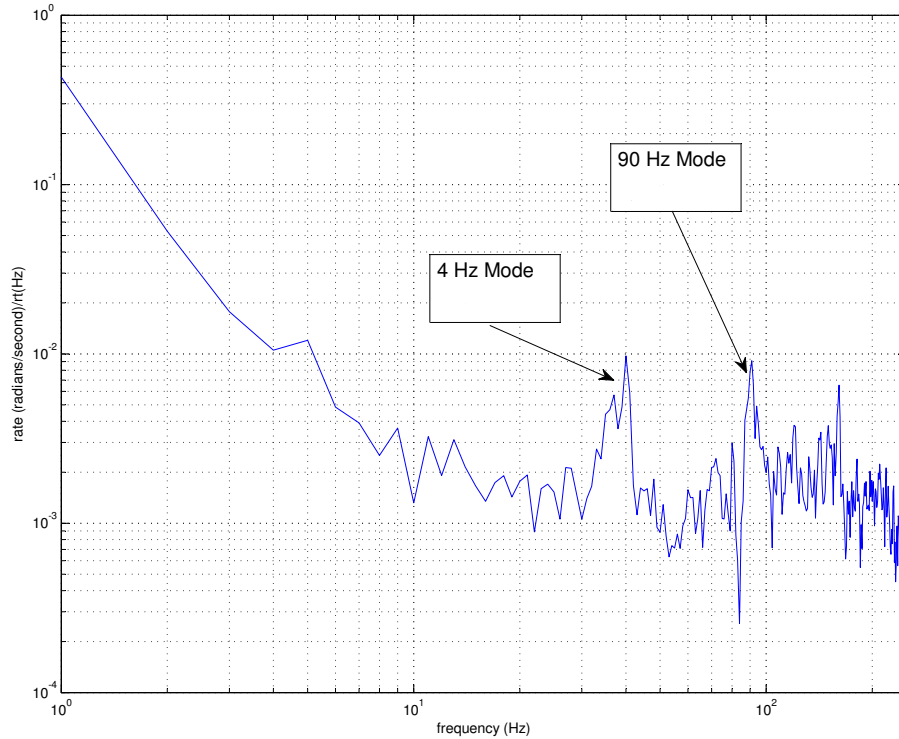
Figure 3.4: Gimbal step rate response frequency content.

These filters have a frequency response indicated in Figure 3.5. The application of these filters to the model yields an open loop Bode plot depicted in Figure 3.6. Note that the corresponding bandwidth is now 7.2 Hz and the phase margin is 60 degrees.

The resulting step response is presented in Figure 3.7. The corresponding metrics are presented in the figure and in a following Table 3.2 comparing the metrics of both model and gimbal.

Table 3.1: Gimbal versus Model Compensation Filters

Application	Filter Type	Value	Descriptor	Units
	Proportional-Integrator	0.9	cutoff f	Hz
Gimbal	Lead	4.1	zero f, 6 dB gain	Hz
	Low Pass Filter	3.4	cutoff f, first order	Hz
	Proportional-Integrator	1	cutoff f	Hz
	Lead	2.3	zero f, 20 dB gain	Hz
Model	Low Pass Filter	6	cutoff f, second order	Hz
	Low Pass Filter	12	cutoff f, second order	Hz

Table 3.2: Gimbal versus Model Performance Comparison

Application	Parameter	Value	Units
	Bandwidth	6.5	Hz
Gimbal	Phase Margin	30	degrees
	Rise Time	0.3	seconds
	Percent Overshoot	0	percent
	Bandwidth	8.1	Hz
Model	Phase Margin	60	degrees
	Rise Time	0.15	seconds
	Percent Overshoot	8	percent

3.4   State Feedback

As aforementioned, the gimbal architecture does not support full state direct feedback. Position and rate are both obtained via the on-axis encoder, but no direct measurement is taken of the structural or drive modes. This feedback does allow visibility of the other dynamics in the system, as was illustrated in the model development chapter. In this case, the swept sine test, which is used to determine the frequency response plot of the system, indicated dominant dynamics that were not anticipated. As discussed in the previous chapter, these dynamics are thought to be attributable to structural or drive compliance. These dynamics can be addressed by using a state observer. A state observer will provide full state estimates for feedback. The validity of these estimates is dependent on the accuracy of the system model.
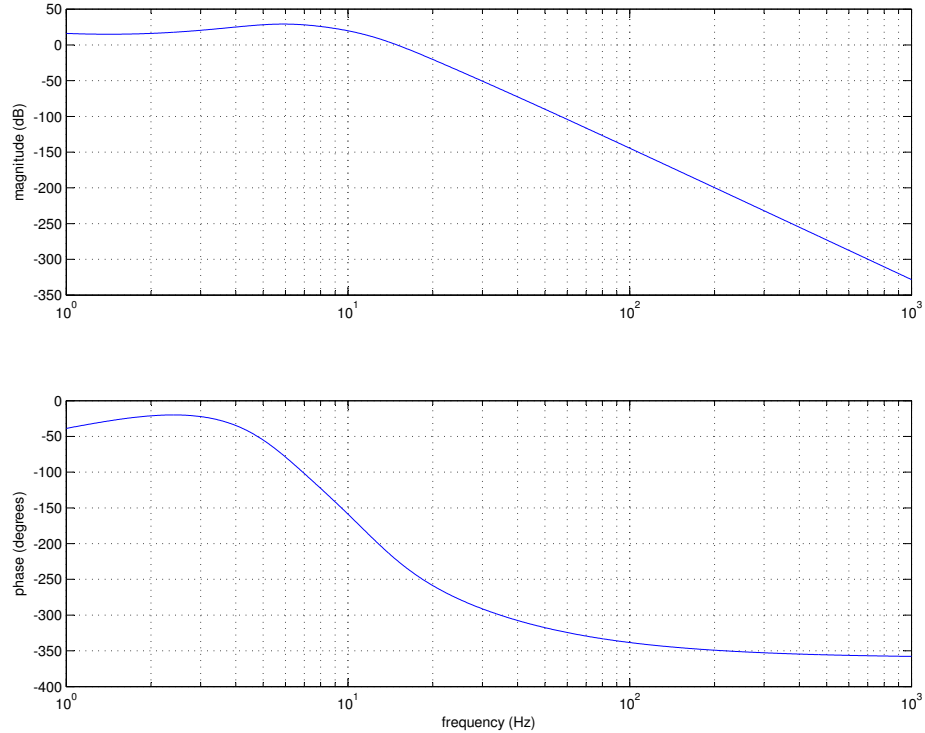
Figure 3.5: Model compensation frequency response.

Since a system model has been created based upon empirical data, a high confidence can be placed upon the validity of the parameters, especially within the band of interest. In order to create an observer-state feed back system, the model must be linearized and placed in the confines of a state space representation. State space is a standard which depicts the states in the form of a sequence of differential equations. These equations take the form:

$$\dot{\overrightarrow{x}} = \mathbf{A}\,\overrightarrow{x} + \mathbf{B}\,\overrightarrow{u} \tag{3.3}$$

$$\overrightarrow{y} = \mathbf{C}\,\overrightarrow{x} + \mathbf{D}\,\overrightarrow{u} \tag{3.4}$$

33
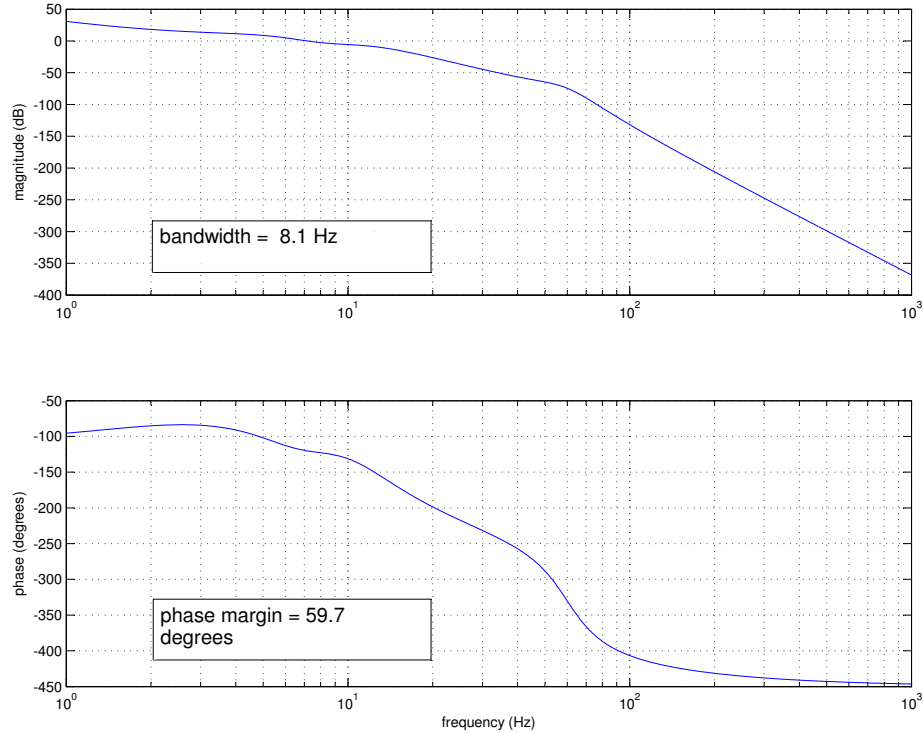
Figure 3.6: Open loop Bode plot of compensated model rate loop.

where:

$\overrightarrow{x}$ is a column vector containing the system states and has dimension n (number of states)

$\overrightarrow{u}$ is a column vector containing the system inputs and has dimension p (number of inputs)

$\overrightarrow{\dot{x}}$ is a column vector containing the derivatives of the states and has dimension n

$\overrightarrow{y}$ is a column vector containing the system outputs and has dimension q (number of outputs)
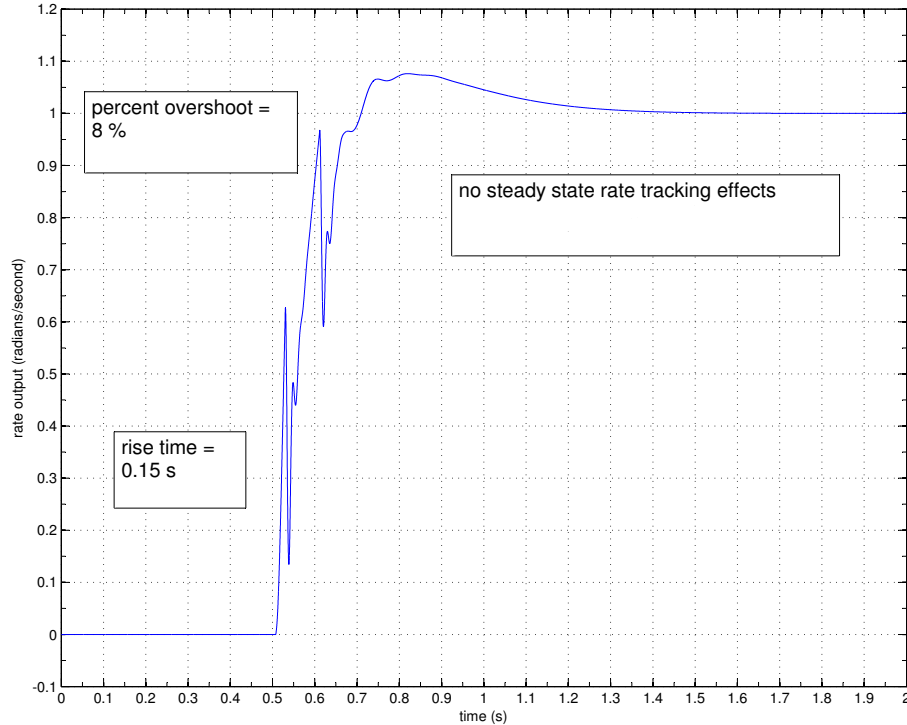
Figure 3.7: Model step rate response.

and

A is a matrix which indicates the state interrelation and has dimension $n \times n$

B is a matrix which indicates which states are directly accessible from the input and has dimension $n \times p$

C is a matrix which indicates which states are directly output and has dimension $q \times n$

D is a matrix which indicates which inputs are fed forward to the output and has dimension $q \times p$

Note that this format allows for multiple input and multiple output systems (MIMO). Further information on state space representation can be referenced in [6] or [7].

This system model is nonlinear in two places: the saturation block in the Command Generation subsystem and the friction model in the Drive, Payload and Sensor subsystem. Typically, when linearizing a system, an operating point must be chosen and the instantaneous rate of change at that point, for each state, is used as the linear equivalent. The Matlab command linmod.m was used with the Simulink block diagram to create the model. The default operating point for this algorithm is zero for each state. This default was used. Note also that the system is now linear and time-invariant.

This yielded the following state space representation:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -3.36 & -457.8 & -4.3098e5 \\ 0 & -0.0038 & -188.5 & -1.4212e5 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 844.8 \\ 273.63 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}$$

In order to build an observer and subsequent state feedback, observability and controllability must be checked with regards to this system.

Observability is the ability to monitor the internal state of a system through the output. Controllability is the ability to control internal states through the input. The check for observability and controllability of a time invariant system is as follows: compute the observability and controllability matrices and check for full rank. Full rank is defined as linear independence of all of the rows or columns of a matrix. If a matrix has more rows than columns, the columns are checked for linear independence, and vice-versa. The observability matrix is tall by definition (more rows than columns) and the controllability matrix is fat. This is illustrated in their time-invariant definitions below:

For a system consisting of state space matrices:

$\mathbf{A} \in \mathbb{R}^{n \times n}$

$\mathbf{B} \in \mathbb{R}^{n \times p}$

$\mathbf{C} \in \mathbb{R}^{q \times n}$

$\mathbf{D} \in \mathbb{R}^{q \times p}$

where n is the number of states, p is the number of inputs and q is the number of outputs.

The observability matrix is then defined as:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

and the controllability matrix is defined as:

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

Most systems are observable and controllable. The exception being systems where there is internal cancellation between poles and zeros. As mentioned in the previous

chapter on modeling, an approximation of the rate was used in the model. This was connected directly to the internal rate state, prior to the integrator block which creates position. This did circumvent the encoder-based position differentiation and smoothing. The reason for this bypass is to avoid the pole-zero cancellation that takes place when the integrator block between the rate and position and subsequent position differentiation are in the system. This would yield the system not fully observable and not fully controllable. The impact of removing this dynamic is illustrated in Figure 2.14.

Another reduction of system order is the removal of the motor pole. This dynamic was removed to reduce the eigenvalue spread in the $\mathbf{A}$ matrix. The motor pole resides at $s = \frac{R_m}{L_m} = 822.6\frac{radians}{second}, or 130.9Hz$. This eigenvalue caused the observability and controllability matrices to have very large entries - on the order of 1e18. The Matlab functions ctrb.m and obsv.m would return partial ranks on each of these calculations. The impact of removing the motor pole is illustrated in Figure 2.5. With these two dynamics removed, Matlab returned full rank for both matrices.

Once observability has been checked, an observer can be built to provide state estimates for feedback. The observer build requires an accurate model of the system. This is due to the observer being primarily a model of the system, with a recursion on the output error:

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + \mathbf{B}u + L(y - \hat{y}) \tag{3.5}$$

The hat nomenclature indicates an estimation. Note that $\dot{\hat{x}}$, $\hat{x}$, u, y and $\hat{y}$ are all vectors. The arrow nomenclature has been dropped for this portion to reduce clutter. $\mathbf{L}$ represents the observer feedback coefficients which can be found via Ackerman's

formula:

$$\mathbf{L} = \alpha_d(\mathbf{A})\mathcal{O}e_n \qquad\qquad (3.6)$$

where $e_n = [0\ 0\ 1]$ (in this case, generally $[0\ 0\ 0\ ...\ 1]^T$ of length n), $\mathcal{O}$ is the observability matrix and $\alpha_d(\mathbf{A})$ is the desired characteristic equation of the system, α_d, evaluated with the $\mathbf{A}$ matrix as the eigenvalues. Note that α_d is the desired characteristic equation, with observer closed-loop poles as the eigenvalues. For this simulation the more sophisticated Matlab function place.m was utilized. The choice of closed loop poles for the observer were chosen to be sufficiently higher frequency than the dominant eigenvalues of the system. These were chosen as 500, 520 and 530 Hz. All poles were assigned negative real values and no imaginary portion to ensure stability.

A separate Simulink model was used to simulate the system with the observer. Both the system and the observer can now be represented as state space systems. The $\hat{\mathbf{B}}$ matrix (observer) was augmented to be:

$$\hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} & L \end{bmatrix}$$

and both u and $(y - \hat{y})$ are input. This is depicted in Figure 3.8. The integrator from rate to position is now carried outside of the state space model as it is not part of the rate loop. State estimate error was checked to ensure convergence. This is shown in Figure 3.9. for this test, the state estimates were given an initial condition of one. The state feedback was turned off for this experiment.

Once the observer is shown to be performing adequately, which means convergence much faster than bandwidth, the state feedback coefficients can be calculated. These coefficients, the matrix $\mathbf{K}$, can be calculated in a few different manners: pole placement (PP) and Linear quadratic regulation (LQR).
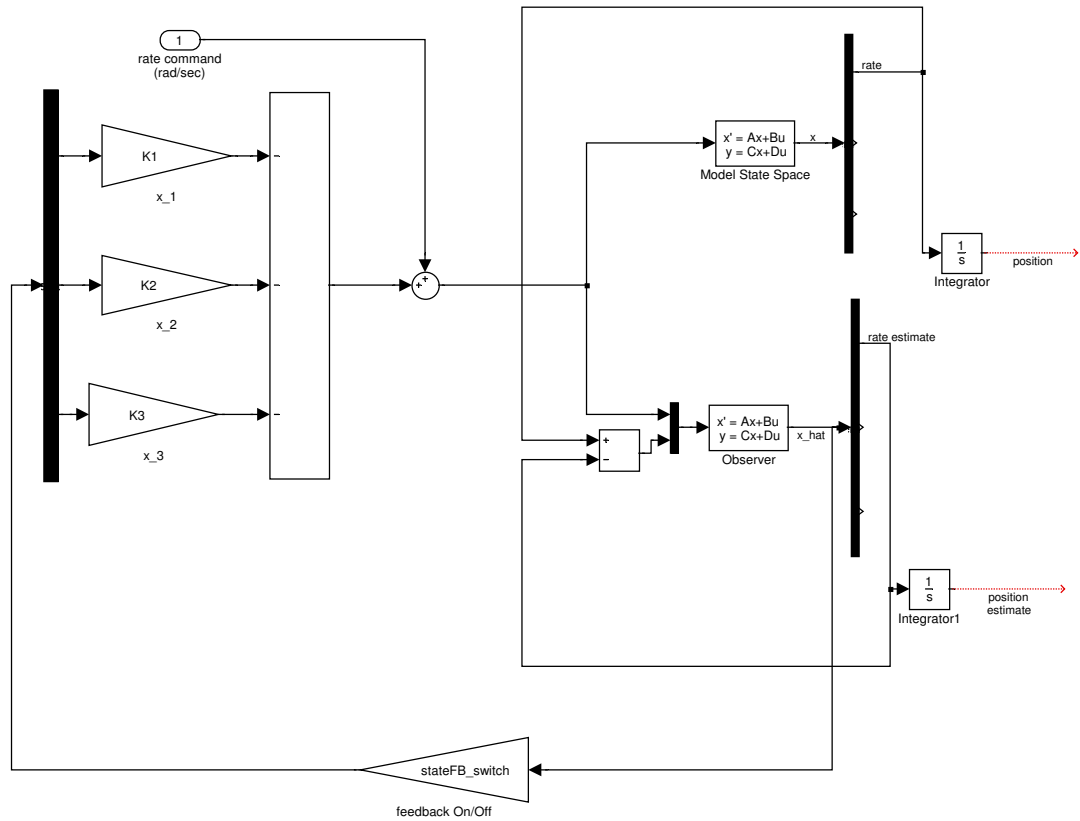
Figure 3.8: Observer Controller Model

Pole placement is utilized in the same manner as the observer closed-loop pole placements above. Another variant of Ackerman's formula can be used to place the closed-loop poles in desirable locations. Matlab's function place.m is used instead for this test. The poles were chosen to have all negative real values. An iterative test was done to check for the optimal placing of the poles with respect to each other. This iteration is accomplished via three Matlab for loops with each of the three poles being placed in range between -5 and -50 Hz. Each pole was incremented by steps of 5 and the rise time and overshoot was evaluated for each permutation. Place.m does not allow for identical values so the cases where any of the eigenvalues were identical
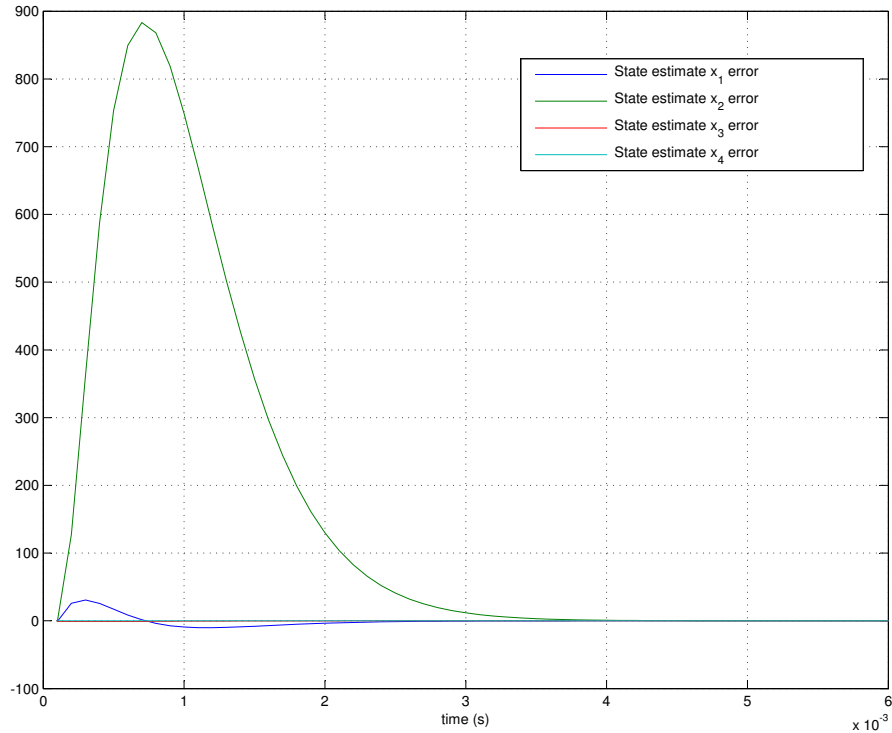
Figure 3.9: State Estimate Errors Converging to Zero

were omitted. It was noted that when the two pseudo dynamics poles were given larger negative real values the rate step response showed significant, high frequency overshoot. The optimal solution which allowed for this overshoot is presented first, followed by the optimal case in which no overshoot was allowed.

The optimal case with overshoot was produced using poles at:

$p_1 = -30 \ Hz$

$p_2 = -35 \ Hz$

$p_3 = -40 \ Hz$

which yields feedback coefficients:

41

$$\mathbf{K}_{pp} = \left[\begin{array}{ccc} 4.855 & -13.295 & -493.87 \end{array} \right]$$

Stabilization (for this feedback coefficient vector) is checked by simulating the model with the feedback coefficients and unity initial conditions for each state. A zero rate command was issued. This is shown in Figure 3.10. The states do converge to zero in a reasonable time, 0.04 seconds, which is above bandwidth ($\frac{1}{0.04s} = 25 Hz$). Note that the position state will not converge to zero as it is an integration of the rate state. The position state rate of change will reach zero.
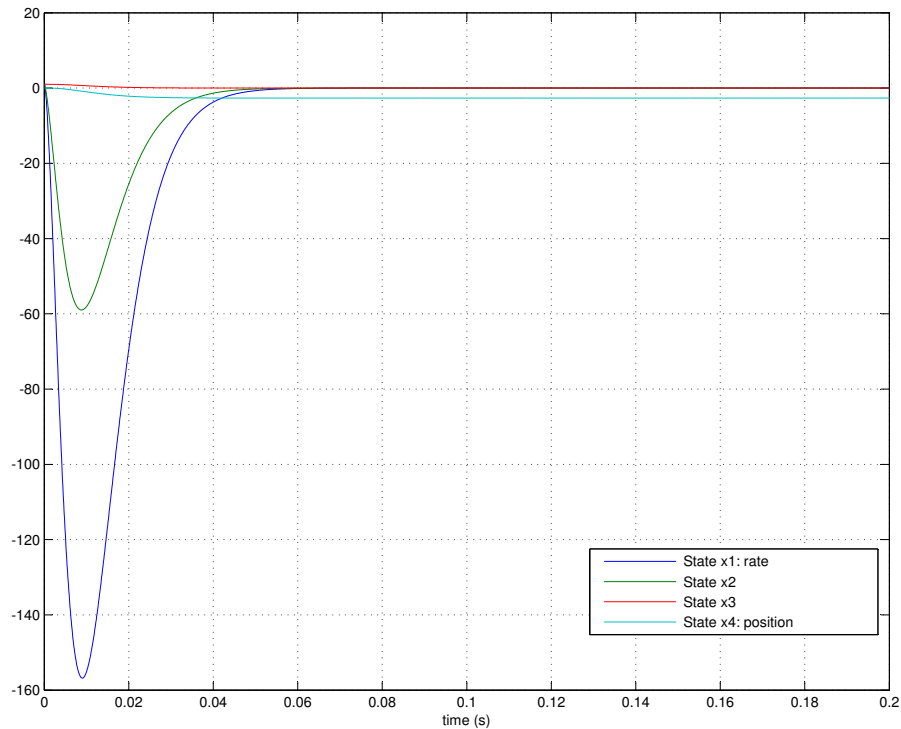

Figure 3.10: States Converging to Zero using Pole Placement with Overshoot allowed

Next, a rate command can be issued. The rate command is input to the system by the following relationship:

$$v_d = H_k^{-1}(0)y_d \tag{3.7}$$

where y_d is the desired output and H_k^{-1} is the inverse of the transfer function of the closed loop system evaluated at zero Hz. This formula is directly taken from reference [6], "Linear Systems", Kailath, page 207. The closed loop Bode diagram of this system is shown in Figure 3.11.

This value was used to scale a unit step input and the results follow in Figure 3.12. Finally, the state error for the position and rate states is plotted in Figure 3.13.
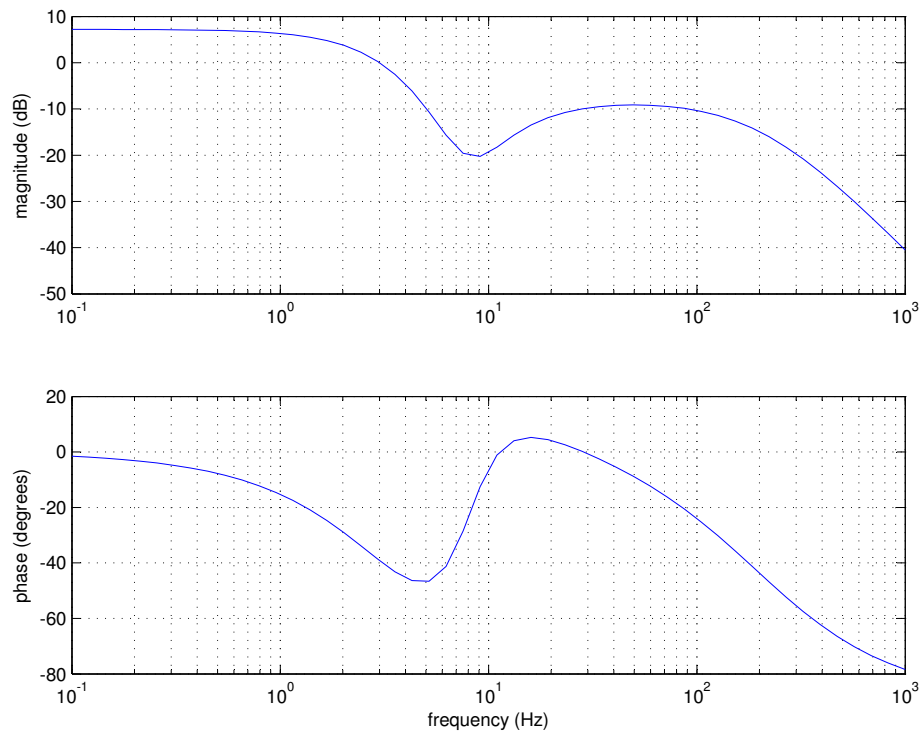
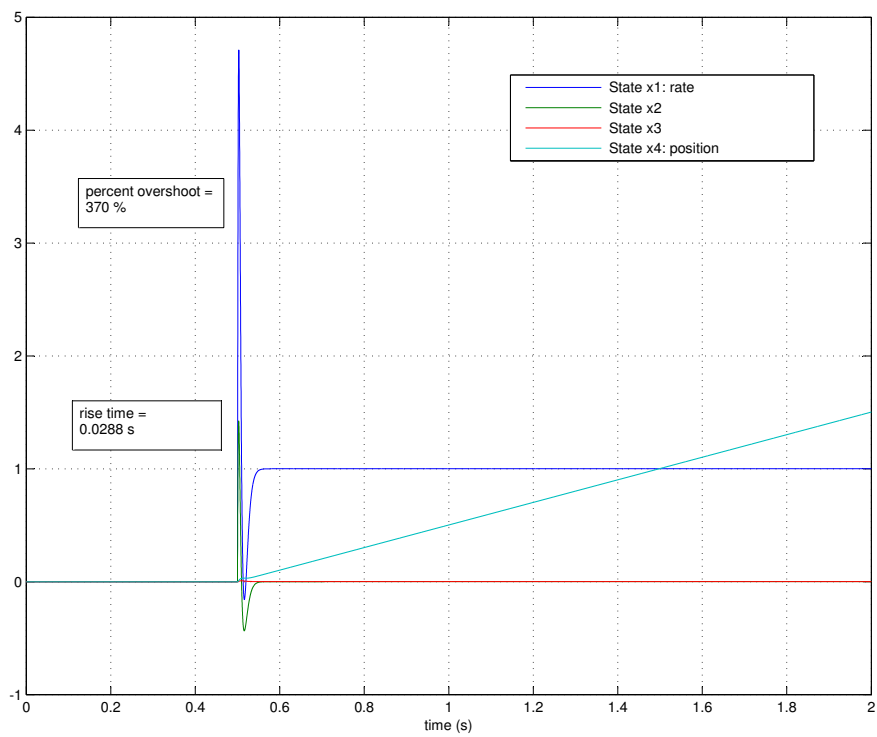Figure 3.11: Bode plot of Closed Loop Pole Placement System

Figure 3.12: States Responding to Rate Step Command using Pole Placement control with Overshoot allowed
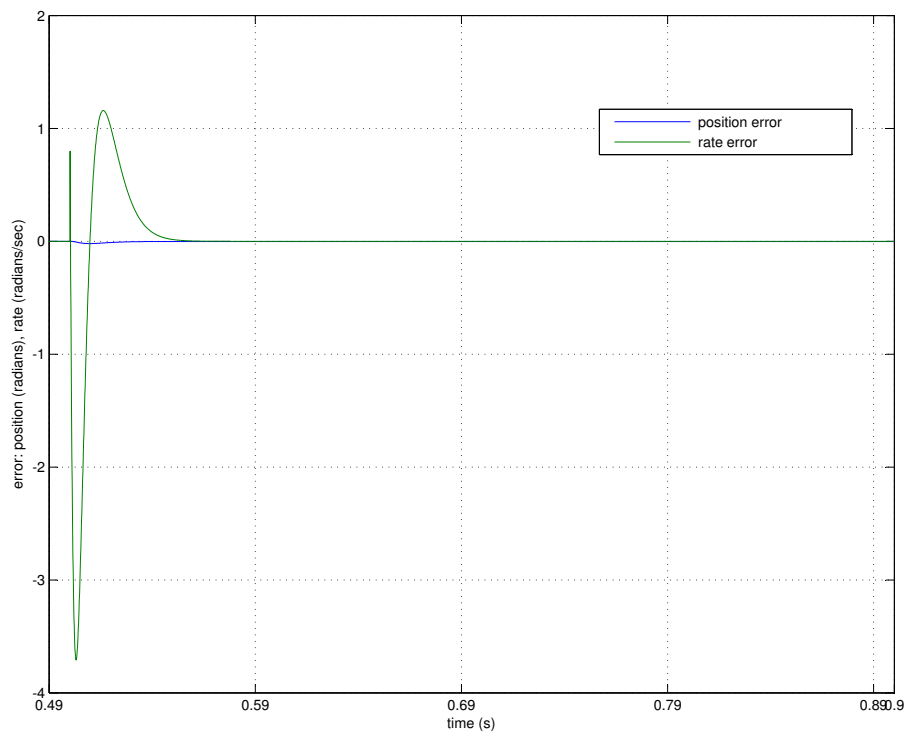
Figure 3.13: Position and Rate error using Pole Placement with Overshoot allowed

This level of overshoot is not acceptable and while the frequency response of the overshoot would probably be attenuated by the gimbal plant, it does show up in the ideal model. A second candidate is now presented which has no overshoot. It should be noted that the feedback coefficient set which provided the fastest rise time and no overshoot limited this high frequency response to 1 ($\frac{radian}{second}$). These poles were placed at:

$p_1 = -10 \ Hz$

$p_2 = -15 \ Hz$

$p_3 = -20 \ Hz$

which yielded feedback coefficients:

$$\mathbf{K}_{pp} = \begin{bmatrix} 0.2898 & -0.5798 & -462.47 \end{bmatrix}$$

The result of the simulation with these parameters is shown in the next three figures representing the stabilization test, the step response and the state error (Figures 3.14, 3.15 and 3.16). Note that the Closed Loop Bode plot is intentionally omitted.
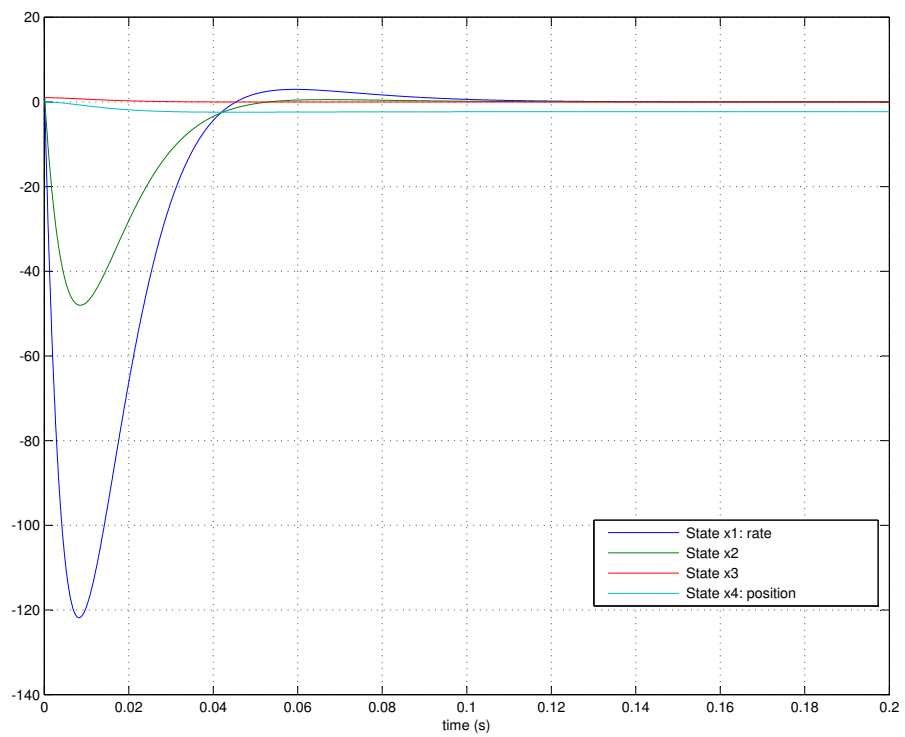
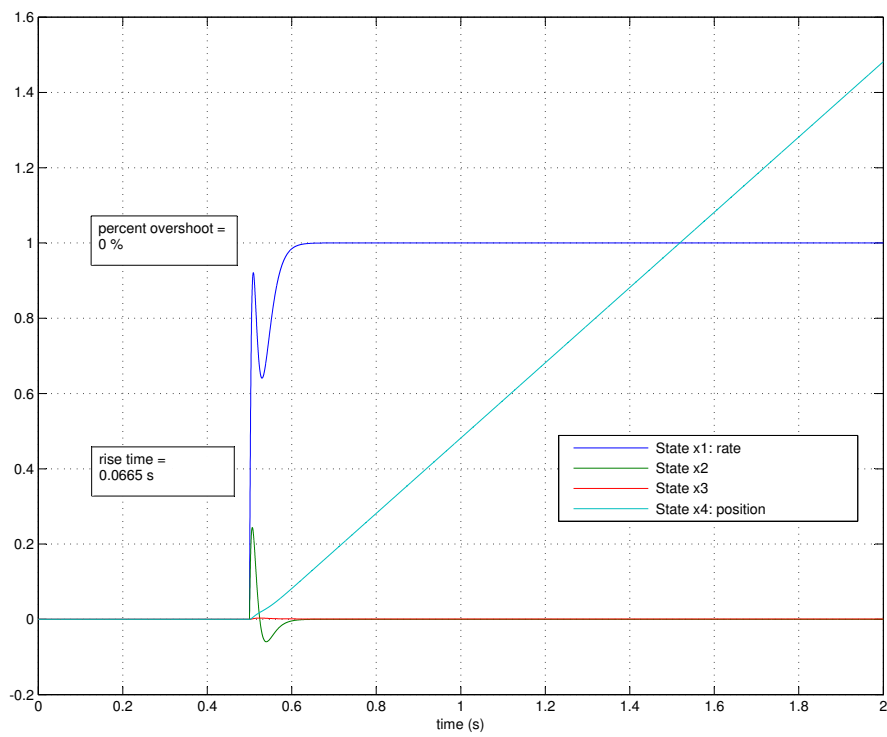Figure 3.14: States Converging to Zero using Pole Placement with no Overshoot allowed

Figure 3.15: States Responding to Rate Step Command using Pole Placement control with no Overshoot allowed
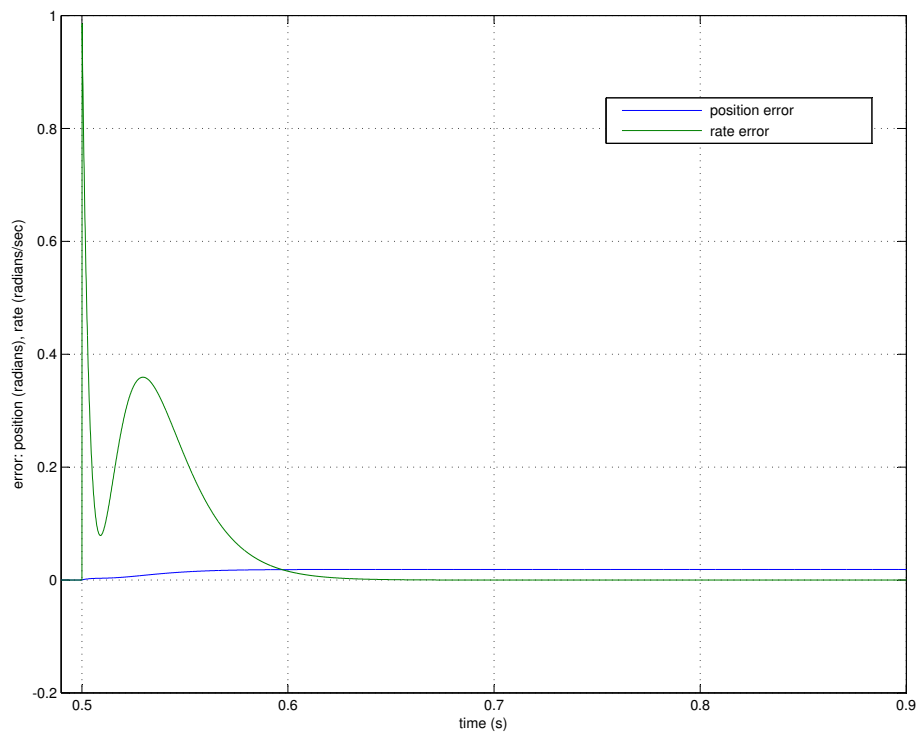
Figure 3.16: Position and Rate error using Pole Placement with no Overshoot allowed

The second technique for state feedback coefficient determination is LQR. To obtain the state feedback coefficients using LQR theory, a few parameters need to be established. LQR is used to minimize a cost function:

$$V = \int_0^T (x'\mathbf{Q}x + u'\mathbf{R}u)dt + x'(T)\mathbf{M}x(T) \tag{3.8}$$

This contains the squares of the x vector of states, the u vector of inputs and the terminal state vector: $x(T)$. These are minimized using the relative Frobenius norms of the $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{M}$ matrices. Relative state weighting is also performed by the distribution within each matrix. Once these parameters are set, the coefficients can be calculated using the algebraic Ricatti equation (ARE). For the purposes of this experiment, the Matlab function lqr.m is used.

For this test, the individual state weight values were scaled with respect to each other (within the $\mathbf{Q}$ matrix) and the $\mathbf{R}$ matrix was set to equal the Frobenius norm of the $\mathbf{Q}$ matrix. Again, the weights were iterated through using for loops in Matlab. The matrix $\mathbf{M}$ is omitted from this test as lqr.m doesn't make allocations for terminal state accuracy. The optimal solution with the fastest rise time is presented next.

The fastest rise time was obtained when the relative weighting within $\mathbf{Q}$ was set as follows:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

This yielded coefficients:

$$\mathbf{K}_{lqr} = \begin{bmatrix} 0.6655 & 0.4170 & -69.27 \end{bmatrix}$$

The same tests and results are produced. First, stabilization was tested using a

zero rate command and non-zero initial conditions for each state. This is shown in figure 3.17.
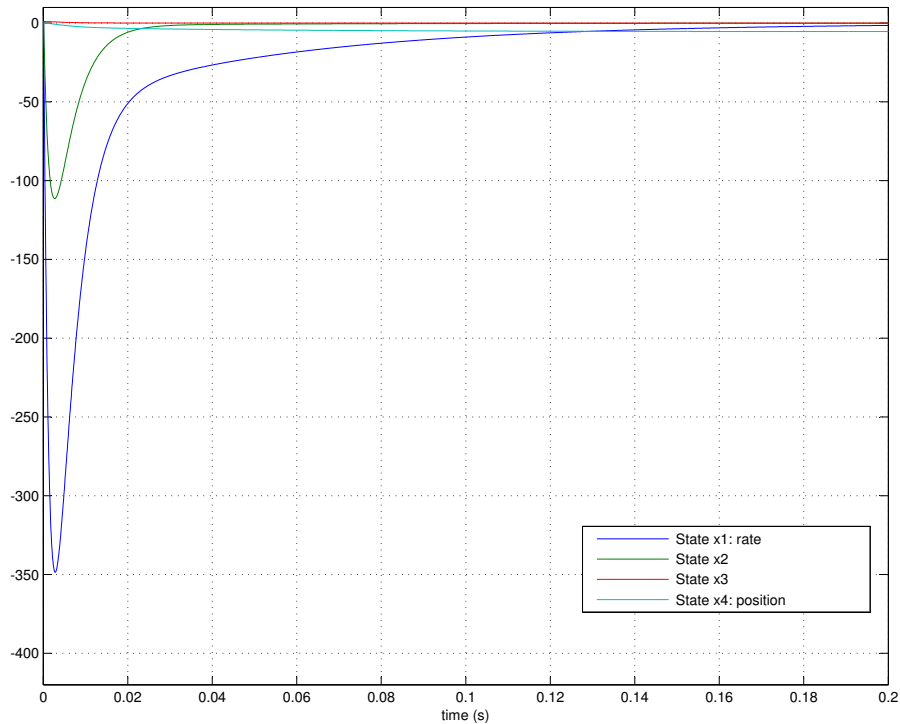
Figure 3.17: States Converging to Zero using LQR

The state feedback model can now be given a rate step command to test tracking. The closed loop transfer function was again evaluated at 0 Hz in order to find the proper command scaling. The simulation results of this test are graphed in Figure 3.18. The error for states x_1 (position) and x_2 (rate) are also generated and plotted in Figure 3.19.

Figure 3.18: States Responding to Rate Step Command using LQR

Figure 3.19: Position and Rate error using LQR

3.5 Results

The results of the three state feedback simulations are tabulated alongside the previous output feedback results and presented in Table 3.3. Only rise time and overshoot are presented for comparison. These metrics are chosen for comparison for two reasons: rate tracking in each of the model simulations doesn't include any ringing and bandwidth is an ambiguous measure when dealing with state feedback systems.
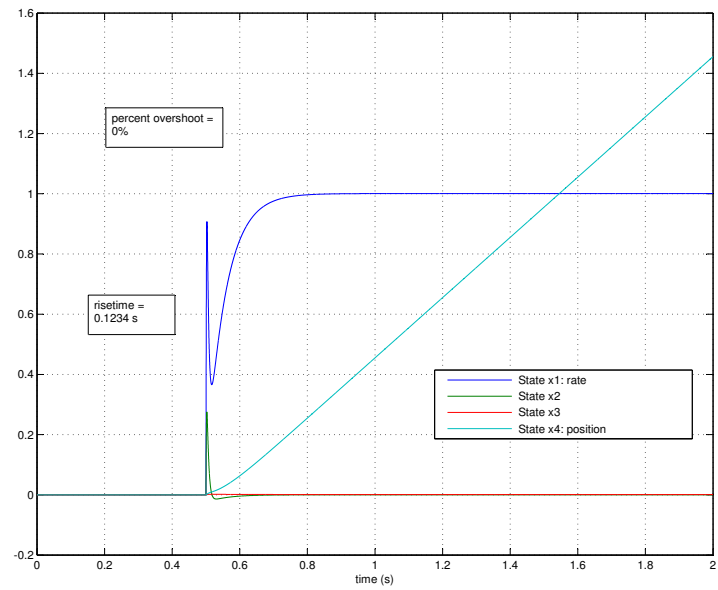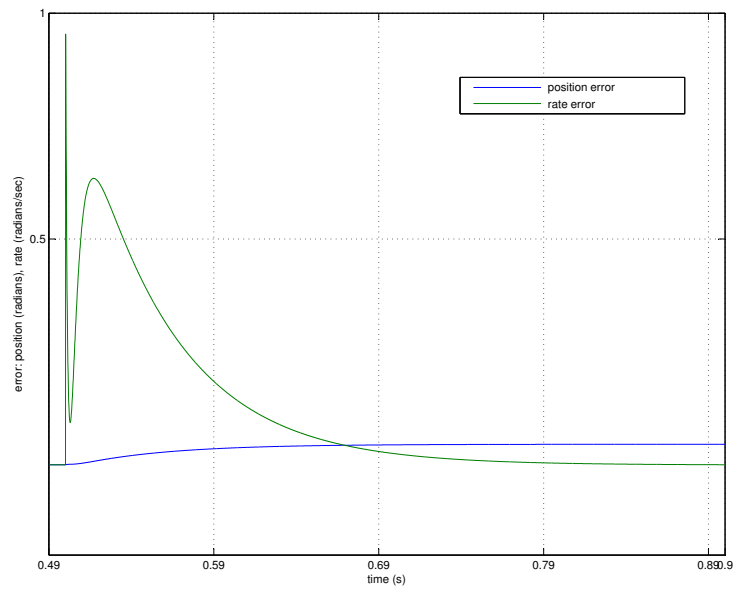
Table 3.3: Gimbal versus Output Feedback and State Feedback Metric Comparison

Application	Parameter	Value	Gains		
Gimbal with Classical Control	Rise Time	0.3 s	K		
	Percent Overshoot	0 %	5		
Model with Classical Control	Rise Time	0.03 s	K		
	Percent Overshoot	2 %	9		
PP with Overshoot	Rise Time	0.029 s	K1	K2	K3
	Percent Overshoot	370 %	4.855	-13.295	-493.87
PP with no Overshoot	Rise Time	0.067 s	K1	K2	K3
	Percent Overshoot	0 %	0.2898	-0.5798	-462.47
LQR	Rise Time	0.12 s	K1	K2	K3
	Percent Overshoot	0 %	0.5966	0.0349	-64.2074

These results indicate that the fastest rise times are present in the model with classical control and the pole placement with overshoot allowed. The PP with overshoot should only be considered because it represents an upper bound and an area of further study. This result is the fastest rise time achieved, and, as mentioned previously, the overshoot may not manifest itself in the actual gimbal rate response because the frequency content is above bandwidth. Of course, the reason for the decreased rise time is probably due to the internal state "ringing" and driving the system. Further investigations into this configuration might actually yield faster rise times in the gimbal.

The model rise time is almost as short in duration. The discrepancy between the model performance and gimbal performance may be attributable to a few factors.

These include: inaccuracies in the model versus the gimbal and a difference in the filter compensation set given to the gimbal and model. For example, the compensated model has a phase margin of 60 degrees, yet has overshoot and a much faster rise time. The gimbal, however, appears to be pushed to the extent of phase margin (30 degrees) and has ringing in the rate command response at a couple of frequencies.

The PP with no overshoot has a respectable rise time, relatively. This may be misleading as the internal state ringing, albeit restricted to unity amplitude ring, may be contributing to the response. As mentioned before, this may not occur in the actual gimbal. In other words, this may be a byproduct of the model being more idealistic than the actual gimbal plant.

Finally, the LQR step response yields a better rise time than the gimbal. While not as impressive as the PP simulation with overshoot allowed, it was roughly equal in rise time to the PP with no overshoot. The LQR control simulation was equal to the model response in rise time, but had no overshoot and may be the most realistic. LQR utilizes state feedback, in the same fashion as the pole placement technique, but LQR also guarantees robustness attributes. This is thought to be the reason for the LQR solution not performing as well as the PP variants. One of these attributes is a 60 degree phase margin in each channel. This can be further referenced in [8].

3.6   Summary

This chapter covered the application and comparison of different control laws on the empirically augmented gimbal model. Results were mixed. The output feedback controller performed better than the gimbal in all metrics. The two PP instances show fast rise times-potentially at the expense of internal stability. Finally, the LQR controller yielded a better response than the gimbal and should produce the most stable and robust controller. These results open the door for many avenues of further

investigation. These topics will be introduced in the next chapter.

Chapter 4

Future Work

The research performed and documented here provides a foundation for future gimbal work. This would include modeling efforts and expanded work in the arena of advanced controls. These two avenues of improvement are explained further next.

A baseline model has been established in this text and can be used in future simulations. This model and the resulting simulations illustrate which parameters are of the greatest value in predicting gimbal performance over a bandwidth. For example, the motor pole dynamics are deemed inconsequential to the gimbal performance over the band of interest. This is a judgment call, based on the margin of performance allowed, but reasonable in most applications. Secondly, the motor pole dynamics could not be allowed when attempting full state feedback. This is due to the inability to calculate observability and controllability (and therefore, use place.m or acker.m) in Matlab. The discoveries made here will aid in the development of future gimbal models.

This testing also indicates some areas of improvement that are necessary in order to make the modeling effort effective. The RTG gimbal used in this test was dominated by the dynamics of the structure or drive. The structure and drive dynamics

play a large role in every gimbal system, although they are not usually as dominant as in the DAYS system. This made it difficult to predict performance and forced me to utilize the empirically-derived pseudo dynamics. The outcome of this process exposed a weakness in the ability to predict both structural and drive dynamics. The drive dynamics were bounded by a plot of torque applied versus frequency. The structural dynamics were predicted via FEA software. The issue is that the damping of these modes and magnitude response is not predictable using the tools that are used currently. A proposed path to gain more accuracy and information in this region is presented in the Literature review section under [3].

The simulation and application of advanced controls is another area of future work. The work presented here establishes an applicable and beneficial control scheme for future gimbal designs. This thesis applied a limited solution in both PP and LQR state feedback. The PP solution was limited in that the pole locations were constrained to being all real and of a certain range. The LQR simulation parameters were limited as well in terms of the relative range of the weights in the $\mathbf{Q}$ matrix (from 0.1 to 1). Both of these techniques could be refined and expanded upon. The pole locations of the pole placement simulation could be strategically placed in order to interact with the existing system zeros. As well, the LQR parameters could be further skewed to look at maximizing the input authority. This would involve adjusting the ratio of the Frobenius norms of the $\mathbf{Q}$ and $\mathbf{R}$ matrices.

Finally, future research and development can be done in the field of Kalman filtering. The Kalman filter provides noise reduction in stochastic processes. This filter estimates the noise variance and uses this estimate to improve the signal to noise ratio (SNR). Applying this filter to analog, zero-mean signals can improve the mean square error significantly. This filter was not simulated in this experiment because the feedback sensor is an encoder, which is essentially digital. This can be implemented in the future on gyro stabilized gimbals as well as other analog sensors.

Appendices

Appendix A

Resources

Matlab and Simulink are registered trademarks of *The Mathworks, Inc.*

Ansys CFX is a registered trademark of *Autodesk*

The code used to produce these results is available. Please contact the author at **mattbroilo@RIEtechGlobal.com**

References

[1] P Skoglar. "Modelling and control of IR/EO-gimbal for UAV surveillance applications". *Linkopings University*, 2002.

[2] R Singh et al. "Modeling and Simulation of the Dynamics of a Large Size Gimbal Platform Assembly". *Asian International Journal of Science and Technology in Production and Manufacturing, Vol. 1, pp 111-119*, 2008.

[3] P Maghami. "Model Refinement Using Eigensystem Assignment". *American Institute of Aeronautics and Astronautics*, 1998.

[4] M Fu. "Linear Quadratic Control with Input Saturation".

[5] H Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, 2002.

[6] T. Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, NJ, 1980.

[7] P Antsaklis and A Michel. *Linear Systems*. Birkhauser, Boston, MA, 2006.

[8] P Dorato, C Abdallah, and V Cerone. *Linear Quadratic Control*. Krieger, Malabar, FL, 2000.