NASA/CR–2014-218275

Investigation and Development of Control Laws for the NASA Langley Research Center Cockpit Motion Facility

Craig R. Coon, Frank M. Cardullo, and Kirill B. Zaychik
State University of New York, Binghamton, New York

June 2014

NASA/CR–2014-218275

Investigation and Development of Control Laws for the NASA Langley Research Center Cockpit Motion Facility

Craig R. Coon, Frank M. Cardullo, and Kirill B. Zaychik
State University of New York, Binghamton, New York

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NNL11AA08C

June 2014

Acknowledgments

I want to first thank Professor Frank Cardullo for his support and guidance throughout this work. Without him this work would have never been developed.

I would also like to thank Victoria Chung from the NASA Langley Research Center for allowing us to perform this research. I want to extend my appreciation to Brian Hutchinson, Lawrence Gupton, Frederick Lallman, Brandon Cowen, Jacob Houck, Paul Davidson, Miguel Alvarez, and Robert Redman for taking time to share their knowledge about the CMF systems with me.

Finally I would like to thank my colleagues at Binghamton University who have helped me throughout this research including Luke Wentlent, Anthony Stanco, Robert Mess and Kirill Zaychik.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Abstract

The ability to develop highly advanced simulators is a critical need that has the ability to significantly impact the aerospace industry. The aerospace industry is advancing at an ever increasing pace and flight simulators must match this development with ever increasing urgency. In order to address both current problems and potential advancements with flight simulator techniques, several aspects of current control law technology of the National Aeronautics and Space Administration (NASA) Langley Research Center's Cockpit Motion Facility (CMF) motion base simulator were examined. Preliminary investigation of linear models based upon hardware data were examined to ensure that the most accurate models are used.

This research identified both system improvements in the bandwidth and more reliable linear models. Advancements in the compensator design were developed and verified through multiple techniques. The position error rate feedback, the acceleration feedback and the force feedback were all analyzed in the heave direction using the non-linear model of the hardware. Improvements were made using the position error rate feedback technique. The acceleration feedback compensator also provided noteworthy improvement, while attempts at implementing a force feedback compensator proved unsuccessful.

Contents

List of Tables

List of Figures

Nomenclature

Ψ - Yaw Euler Angle

Φ - Roll Euler Angle

Θ - Pitch Euler Angle

ζ - System Damping

ω_n - System Natural Frequency

ΔP - Change in Pressure

A_j - Coordinates of the upper bearing attachment point of the j-th actuator.

ARX - A system identification technique which utilizes an Auto-Regressive with an eXogenous input method.

B_j - Coordinates of the lower bearing attachment point of the j-th actuator.

c_{cg} - Vector length from center of platform in neutral to the static COM.

CC - Compensator Command

CL - Control Law

CMF - Cockpit Motion Facility

COM - Center of Mass

COTS - Commercial off the Shelf

CT - Continuous Time

d_i - Actuator Rod Diameter

d_o - Actuator Bore Diameter

dB - Decibels

DC - Demand Calculations

DCL - Digital Control Law

DCU - Digital Control Unit

DOF - Degree-of-Freedom

DT - Discrete Time

DVS - Data Visualization System (Hardware Interface)

EC - Error Calculations

F - Force

FB - Feedback

FC - Feedback Calculations

FFT - Fast Fourier Transform

GFD - Generic Flight Deck

Hz - Hertz

IFD - Integration Flight Deck

l_i - Actuator Length Vector

LaRC - Langley Research Center

LC - Limit Calculations

LQR - Linear Quadratic Regulator

mdl_Fact - CMF Model of Actuator Force

NASA - National Aeronautics and Space Administration

r_{cg} - Vector length from gimbaled centroid to COM

RFD - Research Flight Deck

VC - Valve Calculations

1 Introduction

With the continuous and rapid development of the aerospace industry, it is paramount that flight simulator technology progress along with it. One of the most significant reasons for this is that flight simulators allow for cheaper, safer alternatives to actual flight tests. While there are a number of different areas in which simulators can be improved, the one addressed during this investigation is control law technology. Much advancement has been made in control law technology including improved linear controllers and development of complex model reference approaches.

Control law development is an important part of flight simulation, especially in complex six degree-of-freedom systems, because the better the control law the more effective the simulator. A major hurdle to the development of an improved simulator is increasing the bandwidth while keeping the system well damped. David Carrelli previously had developed an operational controller for the Cockpit Motion Facility (CMF) located at the NASA Langley Research Center [1]. The development of the hardware model which can be used to run the controller on the SIMULINK platform was developed earlier as well [2].

This study sought to improve the performance of the previously developed controller. The majority of the research performed to achieve this was completed using the non-linear model of the CMF dynamics. The non-linear model was utilized because it provided a cost and time effective way to manipulate the CMF dynamics. In order to reduce the complexity of this analysis, most of the research was based in the heave direction because each leg can then be assumed to be identical. To achieve an increase in performance, new feedback techniques were employed such as the implementation of new rate feedback techniques, the addition of acceleration feedback and addition of force feedback.

During the development of the new feedback techniques, a steady-state position error problem was discovered. To solve this problem, integral control was added, which is known from basic control theory to remove such errors. After the steady-state error was removed, the bandwidth was increased through the use of new rate feedback techniques. Finally, the force and acceleration feedbacks were added which also provided system improvement.

The accuracy of various flight simulator models, including models of the simulated aircraft, the pilot and the simulator dynamics are important. These models allow research on the control laws and other governing software to be performed without taking time away from training programs or risking damage to the hardware. The second goal of this research effort was to examine the linear models of the CMF dynamics. Accurate linear models allow for faster and easier testing of controllers. It is also important to have an accurate linear model if the model reference control approach is to be implemented.

2 Background

2.1 Introduction

Flight simulation is important for many reasons, not the least of which is because of the time and resources that can be saved by implementing a simulator in place of actual flight time. With rising fuel costs and the risk of damage to high cost airplanes during training flights, a simulator is a cost effective alternative to performing landing approaches or to train dangerous critical failure modes of an in-flight situation. In addition, training in a simulator allows a student to perform repeated exercises while at the same time receiving immediate feedback from an instructor. There are many variations of flight simulators, from simple systems which have only screen projections, to complicated simulators involving motion in all six degrees-of-freedom with advanced visual displays. These more advanced simulators are often employed at state-of-the-art facilities such as the NASA Langley Research Center (LaRC) in Hampton, Virginia or major military, automotive or airline training centers. The CMF at LaRC was used to perform this research.

2.2 Motion Axis Systems

When discussing motion simulators, the topic of axes systems is very important. With a pilot axes system, motion platform axes system and actuator axes systems all working relative to the ground, it is important to keep each axes system correct. Throughout this work, the term actuator space will be used when a calculation is performed in actuator or leg space; meaning that the axes system is through the actuator as shown below in Figure 2-1.

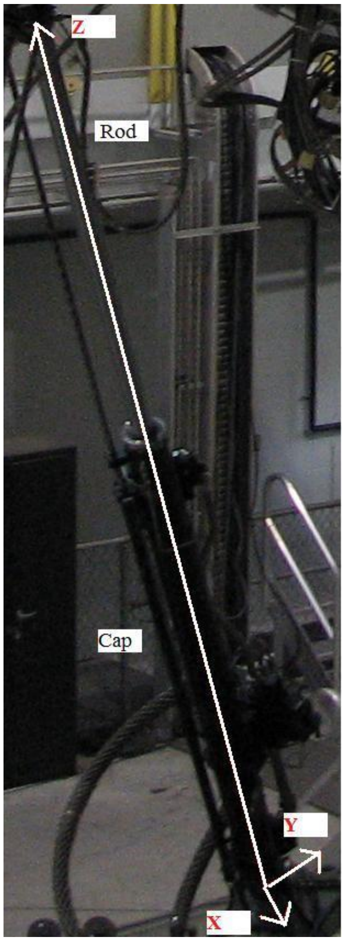

Figure 2-1 - Actuator Axes System

The origin of the actuator axes system is located at the lower gimbal connection, where the actuator is connected to the ground. Note also how the positive z-direction is up, out of the top of the rod. The internal calculations for the CMF motion system controller are performed in actuator space; however, the important performance characteristics are measured in DOF space. Platform space is oriented as shown in the following Figure 2-2.

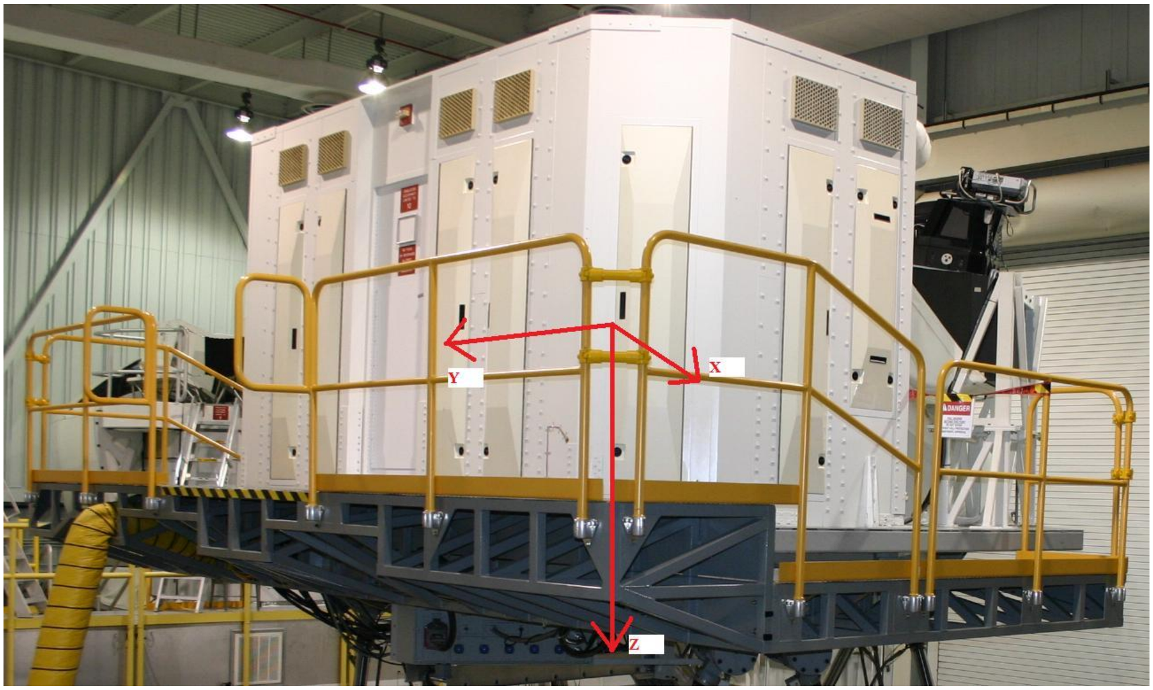

Figure 2-2 - Platform Axes System

As seen in Figure 2-2, the positive z-direction is down with respect to the platform. When reviewing results it is important to remember the difference between the positive z-direction for the actuators versus the platform or body reference frame. The coordinate transformation between coordinate systems associated with the base and the moving platform of the simulator are important to understand. Coordination between coordinate axes systems for the CMF can be seen in Figure 2-3. David Carrelli examined the axes system transformation and the following is a brief summary of his conclusions [2].

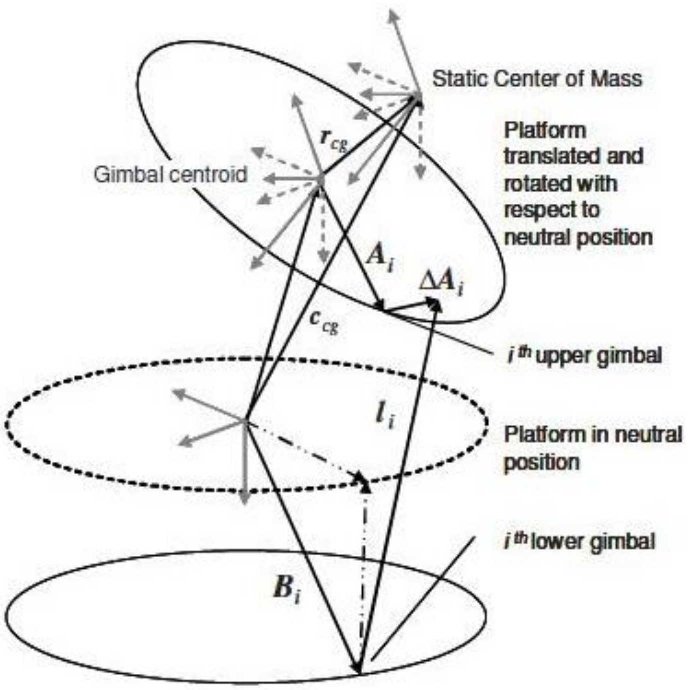

Figure 2-3 - Motion Base Architecture Description [2]

Through examination of Figure 2-3, the actuator position vectors can be derived including flexible body deformation. In the body frame:

$$l_i = A_i + \Delta A_i - r_{cg} + T^T\left(e_{cg}\right)\left(c_{cg} - B_i\right) \tag{2.1}$$

where $c_{cg} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the translational DOF movement of the static center of mass in the inertial frame and B_i is the i^{th} column vector of the matrix defining the lower gimbal locations in the inertial frame [2]. The matrix T is the orthogonal Euler transformation matrix mapping body frame coordinates (associated with the moving platform of the simulator) to inertial frame coordinates (associated with the base of the simulator) and is given by:

$$T\left(e_{cg}\right) = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2.2}$$

where $e_{cg} = \begin{bmatrix} \psi & \theta & \phi \end{bmatrix}^T$ and represents the Euler angles (yaw, pitch and roll respectively) of the body frame relative to the inertial frame [2]. The vectors c_{cg} and e_{cg} are calculated using rigid body dynamics [2]. The simulator platform includes rate gyros and accelerometers located on the platform, near the centroid, which enable the overall platform performance to be measured accurately without requiring the use of any transformation equations. The platform sensors were used to assist in verifying the transformation equations. The importance of understanding the coordinate systems can be clearly seen in Figure 2-4.

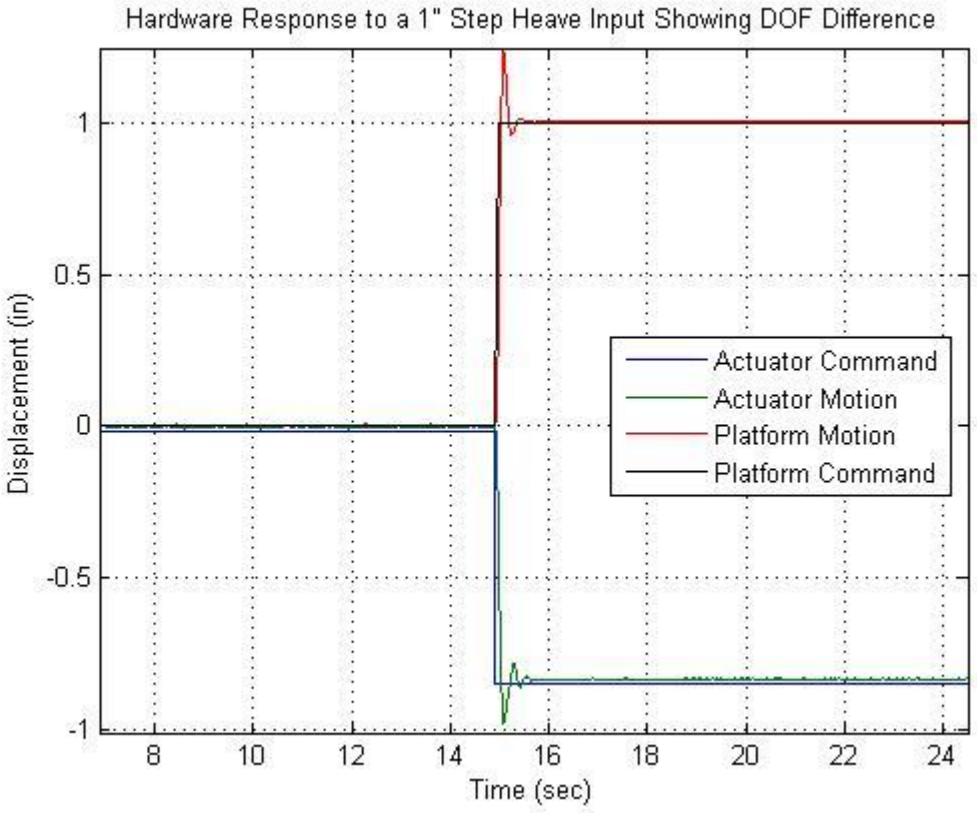

Figure 2-4 - Example of the Responses Depending on the Coordinate Systems

Figure 2-4 shows that the response in either coordinate system matches the input for that coordinate system well. The reason to be careful is that the steady-state value is different between the two coordinate systems.

It has been shown that motion cueing can help augment the visual effects when properly executed [3]. However, it has also been demonstrated that motion cues can have a negative impact during simulation, disrupting the visual cue [3]. The cueing algorithm is designed to compute the transformation between the desired real-world movement, the host input and into the platform DOF. Human motion perception is dominated by sight at low frequencies. Another element of human physiology which helps people perceive motion is the vestibular system. The vestibular system obtains motion cues through fluid movement in the inner ear canals [4]. An important part of flight simulation then is tricking the vestibular system [5]. While an airplane is capable of very large excursions in all degrees-of-freedom a flight simulator has limited excursions. Shown below in Table 2-1 are the mechanical and operational limits of the CMF motion base.

Table 2-1 - Mechanical and Operational Displacement Limits of Motion Base [6]

	Mechanical Limits	Operational Limits
Vertical	+/- 43 Inches	+/- 41 Inches
Lateral	+/- 58 Inches	+/- 55 Inches
Longitudinal	+70,-57 Inches	+67, -55 Inches
Pitch	+30, -27 Degrees	+28, -25 Degrees
Roll	+/- 30 Degrees	+/- 28 Degrees
Yaw	+/- 40 Degrees	+/- 38 Inches

In order to keep the sensation of acceleration throughout the simulation, other motions must be faded into the simulator body movement. The cueing algorithm calculates the transformation from desired motion cues to the proper motion platform movement. The commands begin as airplane commands which are then converted through the cueing algorithm to platform commands. The hardware input commands are in the platform degree-of-freedom. The platform commands are then converted to actuator commands. To obtain proper actuator commands the expected platform position, velocity and acceleration all need to be fed in from the host. Once all the commands are properly given, the system can perform the functions which are demanded of it.

2.3 Complete System Overview

The motion platform at the NASA Langley Research Center is a state-of-the-art six degree-of-freedom motion platform. The CMF is a unique facility and the operation of the system combines four major sub-systems. The organization of the overall system can be seen in Figure 2-5.

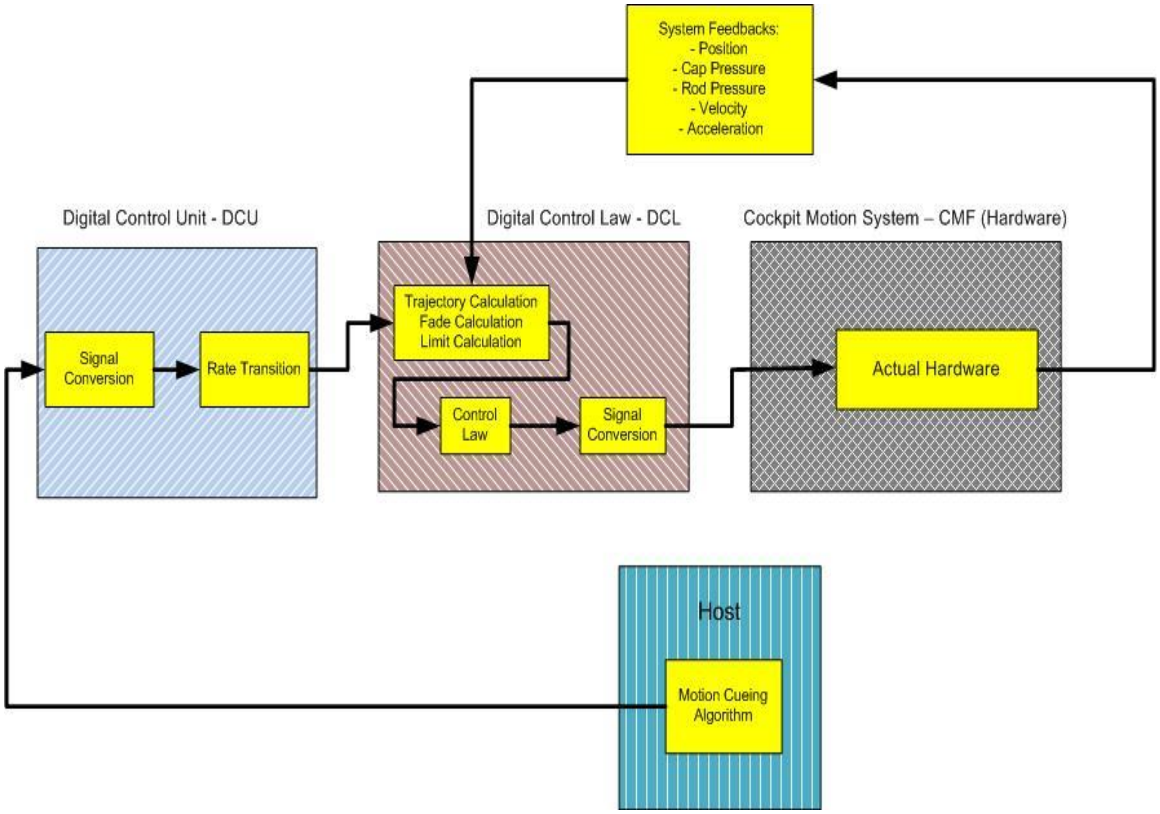

Figure 2-5 - CMF System Block Diagram

The block diagram shown in Figure 2-5 comprises four main components: the host, the Digital Control Unit (DCU), the Digital Control Law (DCL) and the Cockpit Motion System (CMF/Platform). The following is an overview of the DCU, the DCL and the CMF/Platform involved in the research.

2.3.1 DCU Architecture Overview

The DCU comprises the entire simulator control system architecture. The DCU is where the host inputs enter the motion control system. The host signal is derived from when the pilot exercises a control input and the aircraft dynamics simulation calculates the simulated aircraft state which is then processed through the cueing algorithm which determines an appropriate desired motion system state in inertial space. The host signal, which includes position, velocity and acceleration, is processed before being sent to the DCL. An important DCU operation is the system is sampled to a higher rate. The reason for this is that the host inputs are sampled at no higher than 80 Hz [6], while the control law, inside the DCL, operates at 2048 Hz. Therefore in order to perform the necessary computations, the host input needs to have data points interpolated for use. The signal is then passed through to the DCL.

2.3.2 Digital Control Law

The DCL is at the heart of the research performed in regard to the motion platform. The DCL is responsible for a number of different computations. The major areas the DCL covers can be seen in Figure 2-6.

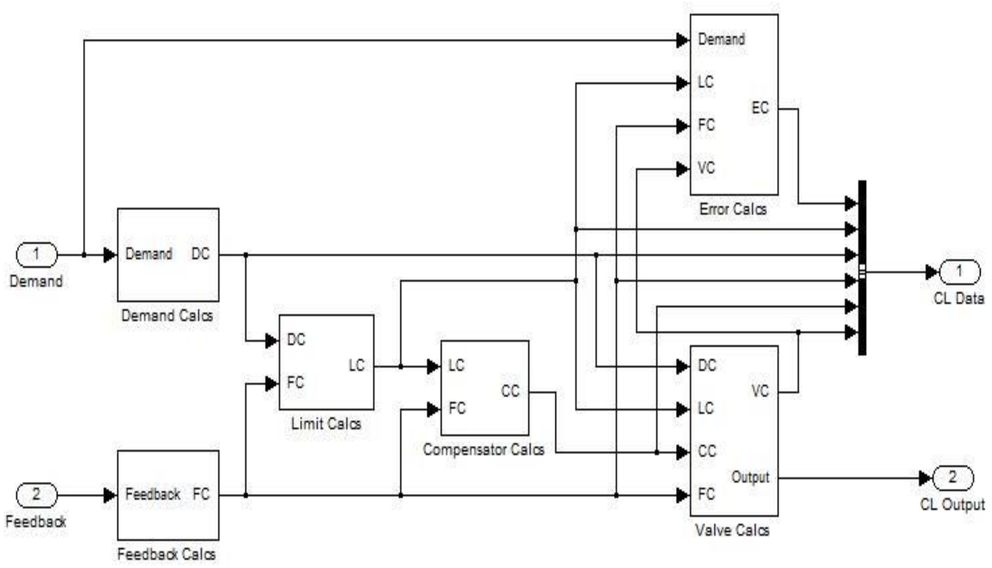

Figure 2-6 - Digital Control Law Block Diagram

The DCL is comprised of the demand calculations, feedback calculations, limit calculations, compensator calculations, error calculations and valve calculations. The signal arriving from the DCU, labeled as demand, has not been transformed to the actuator coordinate system. The DCL performs this conversion as one of the first computations and is performed within the demand calculations. The demand signal, after passing through the Demand Calcs block, is now in the proper coordinate system for computation throughout the remaining portions of the control law as well as for use in the system hardware.

The Demand Calcs (DC) signal is then fed into the limit portion of the control law. The demand is combined with the feedback signals from the sensors in the hardware. The feedback is converted from analog to digital within the feedback block and only the necessary signals are passed on for the remaining control law calculations.

The limit calculation block performs the important role of maintaining safe operation of the hardware. It is important to monitor the movements of the platform because an out of control leg could lead to expensive damage or even possible harm to the human operator(s). The hardware limits of the system include leg position limit switches, a velocity clamp, implemented as part of the NASA designed Safety Electronics Module, and a rate limiter, included in a COTS Servo Drive card as seen in Figure 2-7. These functions are also mirrored in the limit calculation block to allow the control system the ability to grace fully respond to commanded trajectories outside the system limits [1]. The legs are therefore limited in position, velocity and acceleration.

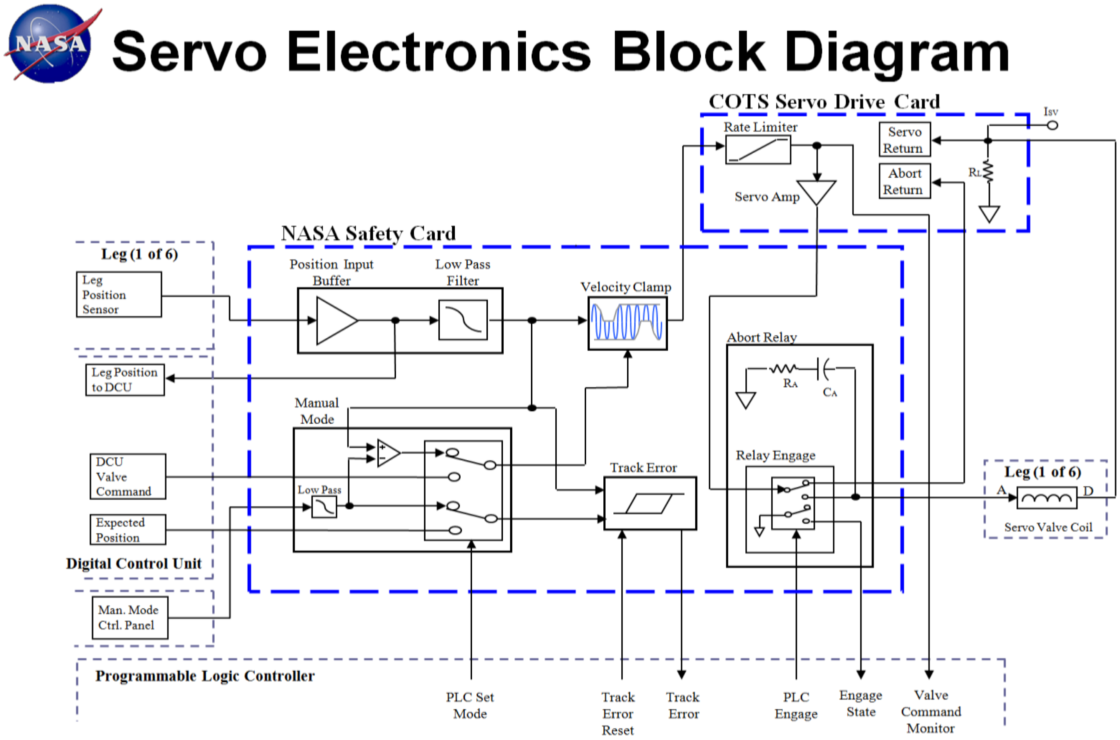

Figure 2-7 - Servo Electronics Block Diagram [7]

After the feedback signals and demand signals are compared in the limit calculations and the safety stops have not been triggered then the limited demand signals and the feedback signals are passed to the compensator. The compensator calculations block is the location of the compensator.

2.3.3 Compensator

The compensator is the part of the control system which governs the platform hardware, the original design of which can be seen in Figure 2-8.

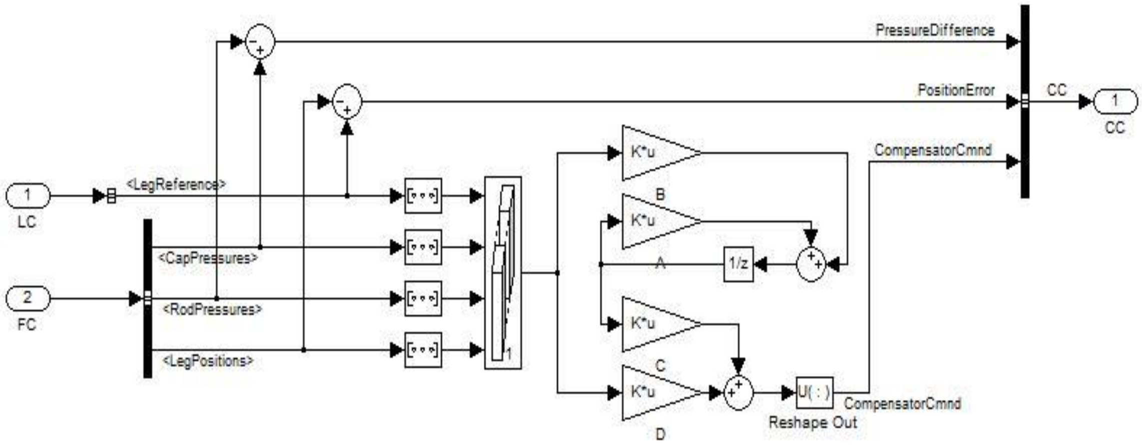

Figure 2-8 - Compensator SIMULINK Block Diagram

The compensator design shown in Figure 2-8 represents the original compensator [8]. The design was picked for optimizing the block diagram and is not in fact true state-space; however it is of quasi state space form, as shown in Figure 2-9.

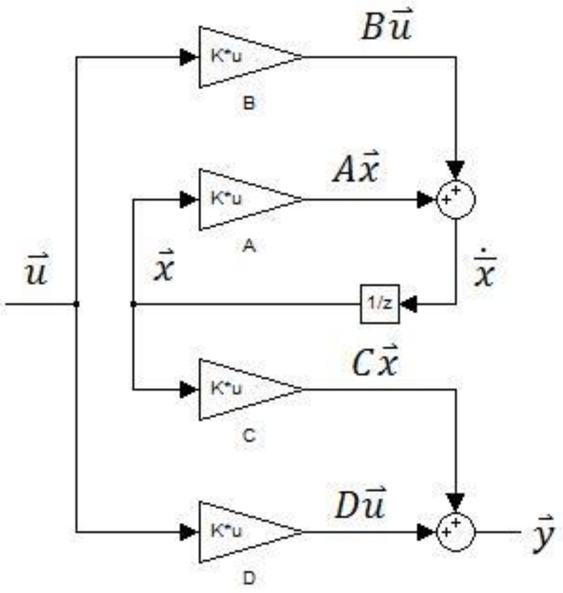

Figure 2-9 - Block Diagram of the Linear, Continuous Time Control System Represented in State Space [9]

The linearized state equation and output equation can be represented as shown in Equation 2.3 and Equation 2.4 which are based upon Figure 2-9.

$$\dot{x}=A\vec{x}+B\vec{u} \tag{2.3}$$

$$y = C\bar{x} + D\bar{u} \tag{2.4}$$

The variables available at that time included rod and cap pressures and axial leg extension. During the original compensator design there were no axial leg velocities available in real-time from actuator sensors; therefore the leg velocity was approximated over expected frequencies using a lead-lag compensator. The compensated velocity was then combined with the pressures, leg position feedback and leg reference to form the final compensator. The leg reference signal is combined with the leg position feedback, the cap, and the rod pressures in the concatenation block. The concatenated signal is then passed into gain blocks B and D simultaneously. To better understand the SIMULINK compensator design including the combination of an estimated velocity with the other sensor feedbacks consider the block diagram in Figure 2-10.

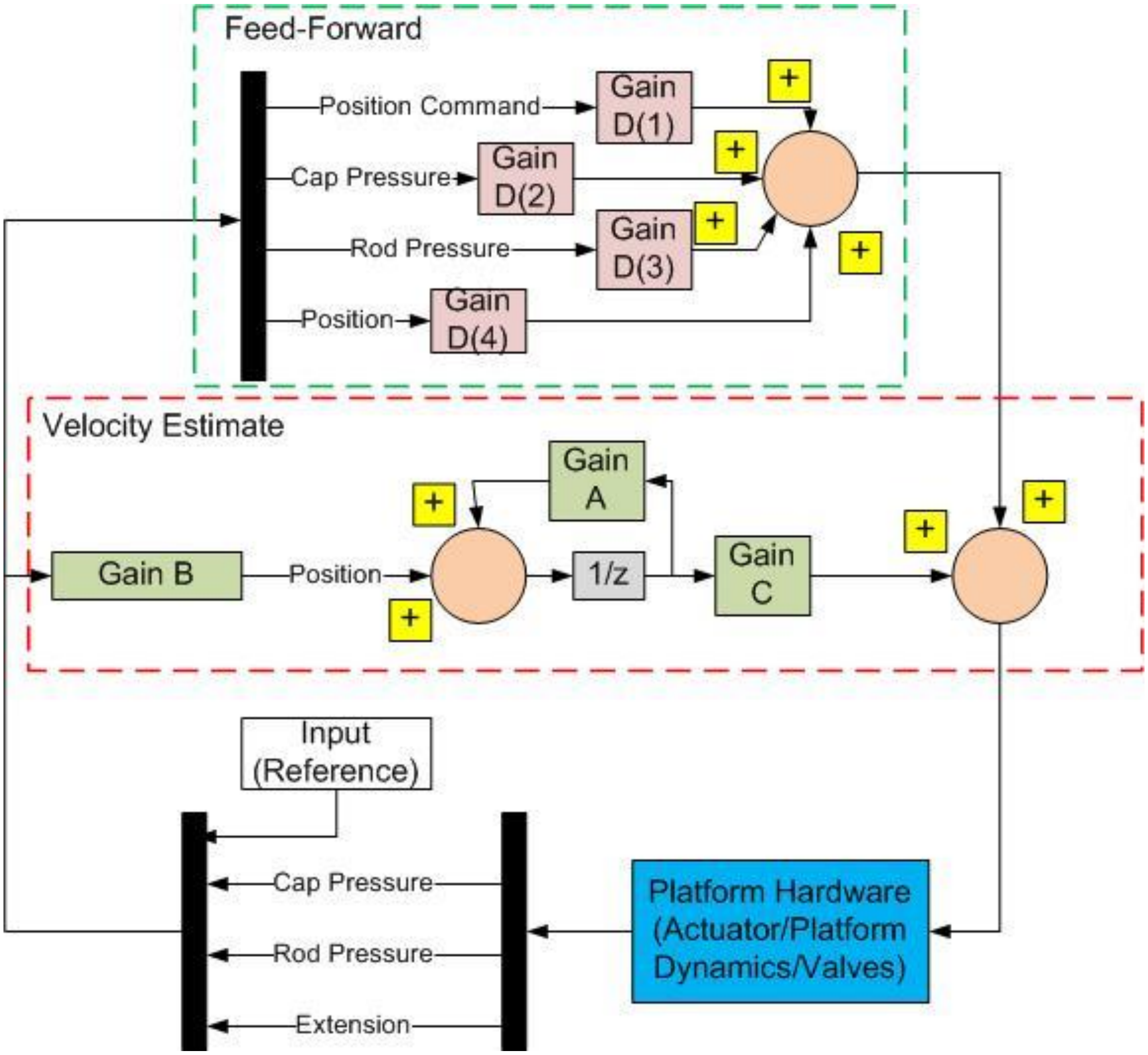

Figure 2-10 - Compensator Design Block Diagram

Figure 2-10 is an expansion of the SIMULINK design of Figure 2-8. The two block diagrams are mathematically equivalent, Figure 2-10 is meant to clarify how the matrices are applied. An example of the matrix gains are the original control law gains shown below in Table 2-2.

Table 2-2 - Original Compensator Gain Matrices

Gain A	0.914			
Gain B	0	0	0	0.1709
Gain C	-149.8			
Gain D	21.91	-1.949E-07	9.816E-08	166

It can be seen that gain matrix B acts to remove all incoming signals except position feedback. The velocity estimate loop then only has a single value being operated on. The signal in the velocity estimate loop, the position feedback, is then operated on by gains A and C. The result of the velocity estimation loop is combined with the result from matrix D which multiplies the concatenated output. The result of this combination is the compensator command, between plus and minus 100, which is then passed to the valve calculation block [3]. The valve calculation block, depending upon the operational state, applies any necessary fades, and outputs a digital percentage of valve command which equates to velocity. The digital signal count is then converted to an analog voltage by the digital to analog "DAC" converter for use in the hardware.

2.3.4 Cockpit Motion System

The CMF is a Stewart platform design, which is a synergetic system that operates in six degrees-of-freedom. An advantage of the Stewart or "hexapod" platform is that the configuration of the platform provides a geometric advantage. The platform's six degrees-of-freedom include three translational and three rotational degrees-of-freedom. The translational degrees-of-freedom include surge (movement forward and backward), sway (movement right and left) and heave (movement up and down). The rotational degrees-of-freedom include pitching (rotating up and down), rolling (rotating side to side) and yawing (rotating left and right).

The CMF has the ability to operate with three different cabs which can be changed when needed; the Generic Flight Deck (GFD), the Research Flight Deck (RFD) and the Integration Flight Deck (IFD). Each of the cabs is used in different research experiments, and has unique characteristics, such as different mass properties. The difference in flight deck mass property has implications for the dynamics of the system, which in turn impacts models of the hardware. As well as having different flight decks the hardware inside each cab is also reconfigurable and updateable [9]. The ability to change cabs and update deck hardware helps give the CMF a large operational window.

2.4 System Dynamic Model (CMF)

In order to perform off-line analysis of the motion system, a closed loop control system dynamic model was designed to operate with the DCL and DCU [2]. When the system was under initial control law improvement, or was being run at the laboratory in Binghamton, the non-linear model of the hardware dynamics was used. The platform and the associated hardware and dynamics were modeled in a non-linear fashion based somewhat off first principles. At the time Carrelli was designing his first compensator, the hardware had not yet been delivered from the manufacturer. However, an initial control law was required for when the hardware was installed. The lack of reliable hardware data as well as some of the hardware parameters, such as valve characteristics, is what drove Carrelli to design the non-linear model.

There are several major objectives of the non-linear model [2]:

- Calculate the basic hydraulic system variables including the fluid flows and pressures within the actuators.
- Calculate the axial leg loads during normal operation and failure events.
- Calculate the payloads during normal operation and failure events.
- Accurately predict plant dynamics, both linear and non-linear, for control law design and tuning.

To obtain these objectives, the motion base model was broken down into two subsections, the hydraulic system and the mechanics subsystem.

2.4.1 Hydraulic Subsystem

The hydraulic subsystem includes models of the accumulators, pumping unit, valves, actuators, flows and pressures. The accumulators and pumping unit were modeled to assess if there would be capacity problems during high demand motions [2]. The servo valves (two for each actuator) are the typical two-stage flapper-type design. It is assumed in the model that if the pressure across the valve is below the pressure required to maintain the spool opening then the flow is zero.

The actuator model leaves out a few aspects such as hysteresis and friction. The hysteresis of the actuators is believed to be caused by thermodynamic effects, such as how the temperature change of the oil during operation changes the viscosity of the oil. Another cause of hysteresis is assumed to be that the fluid is not actually incompressible. The hysteresis effects are very difficult to model and were therefore left out of the non-linear model. The actuator model did not include coulomb or dynamic friction because the hydro-static bearings were assumed to keep the amount of friction force at a negligible amount [2]. Without hysteresis and friction modeled there is cause for concern over how large a window the operation of the non-linear model is valid.

The system plumbing was designed using Ohm's law which is widely used in electrical circuits. Keeping with the electrical circuit analogy Kirchhoff's current law can be applied to the hydraulic circuitry if thermal expansion and contraction as well as piping distortion are ignored [2]. The equivalent hydraulic circuit can be seen below in Figure 2-11.

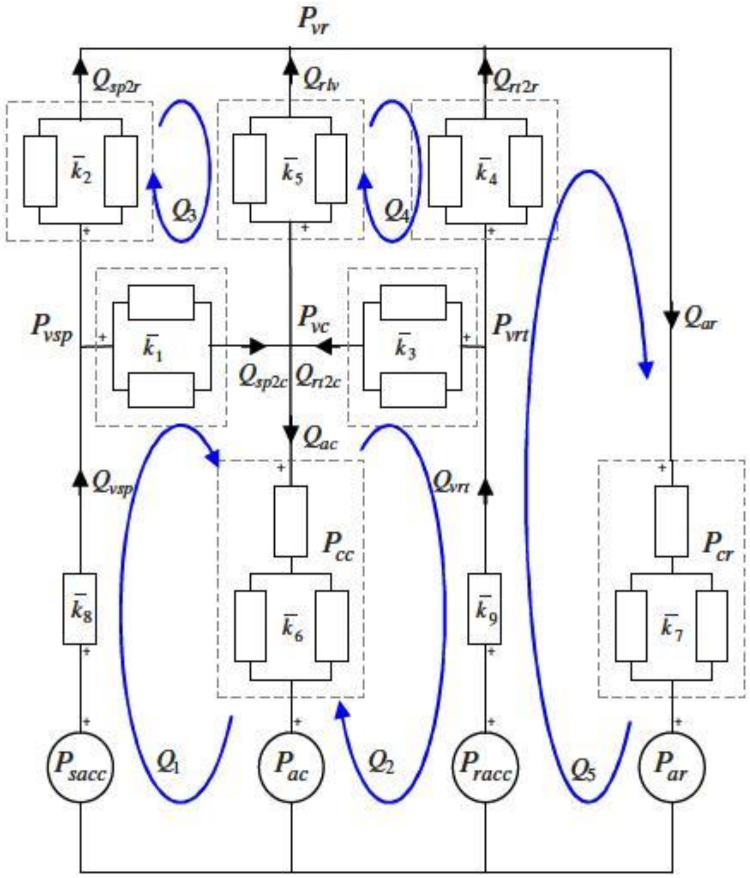

Figure 2-11 - Equivalent Circuit of Key Hydraulic Elements [2]

Finally it is important to note that the actuators are all modeled as if the properties of each actuator and valve are identical. Six actuator instances each with two valves need to be combined in the non-linear model to mimic the six actuators which are included in the actual hardware.

2.4.2 Mechanics Subsystem

The mechanics subsystem is where the payload dynamics and system kinematics are modeled. The current non-linear model only includes the use of static load and effective mass. The static load is a constant determined by knowing the cab which is desired to be modeled. The effective mass is the dynamic component imposed on each actuator, which is not necessarily the total system mass. However, the effective mass does not take into account the reflected mass. The reflected mass is the equivalent mass an actuator experiences when accounting for apparent inertial loading during accelerated motion. The reflected mass is defined as the ratio of the actuator’s axial force to the actuator’s axial acceleration, including gravitational effects, and is a complex function of motion base velocity, pose, geometry and payload mass properties [2].

2.5 Original Compensator

The compensator architecture, as discussed in detail in Section 2.3.2.1, was determined in the early stages of development. The SIMULINK implementation shown in Figure 2-8 was designed for optimal performance within the DCL. The next step was optimizing the compensator to meet the desired performance characteristics.

The performance characteristics of interest to operators are the damping of the system and the phase bandwidth. System damping is important because of the consequences associated with a very low under damped system. A very low under damped mechanical system will continue to have motion for an undesirable amount of time which leads to unintended or improper cues. The second characteristic which is of importance is the phase bandwidth. It is recognized that pilots, during a training exercise, can tolerate a response up to 90 degrees out of phase [5]. After the input and output are out of phase by 90 degrees or more, the cues deteriorate decreasing the effectiveness of the training exercise.

The original compensator design in Figure 2-8 with the values from Table 2-2 was tested using both the non-linear model and the system hardware. The inputs driving the system for the non-linear model and the hardware were not exactly identical. The non-linear model was evaluated using a chirp signal with frequency range from 0.01 Hz to 10 Hz and signal amplitude of 0.33 inches. The frequency range was chosen to have adequate data points past the frequencies of interest which are from one to five hertz. The signal amplitude was chosen because a full leg extension in heave from midpoint is 37.5 inches. 37.5 inches is used instead of the vertical operational limit of 41 inches because of the geometric advantage of the hexapod design. The final input signal was approximately one percent of the heave full leg extension which was 0.33 inches. A chirp signal input was not used to evaluate the system hardware. The Digital Visual System (DVS), which is the interface used to feed inputs to the hardware when the unit is not operating from the host, does not accept inputs such as chirps or white noise. The DVS is only capable of accepting a single, simple type of input per channel with seven channels per run available. To overcome the DVS input challenge, a range of independent single frequency sinusoids were used. The DVS has seven channels available for a signal input per motion base run. In order to obtain the range of frequencies necessary for an adequate frequency response function, five rounds of seven independent frequencies were executed on the motion base. The values used per round and in each channel can be seen in Table 2-3.

Table 2-3 - Frequency Response Hardware Frequencies

	Channel 1 (Hz)	Channel 2 (Hz)	Channel 3 (Hz)	Channel 4 (Hz)	Channel 5 (Hz)	Channel 6 (Hz)	Channel 7 (Hz)
Round 1	0.1	0.2	0.4	0.6	0.8	1.0	1.2
Round 2	1.4	1.6	1.8	2.0	2.4	2.6	3.0
Round 3	3.4	3.6	4.0	4.4	4.6	5.0	5.4
Round 4	5.6	6.0	6.4	6.6	7.0	7.4	7.6
Round 5	8.0	8.4	8.6	9.0	9.4	9.6	10.0

The hardware amplitude was the same as the non-linear input of 0.33 inches. In order to develop the frequency response curves, the hardware signals were stitched together before performing the Fast Fourier Transform (FFT) and determining the amplitude and frequency responses. The code for both the data stitching and performing the FFT responses, found in Appendix A, utilizes the inverse transformation discussed earlier. Figure 2-12 presents results of the non-linear model and hardware frequency responses.

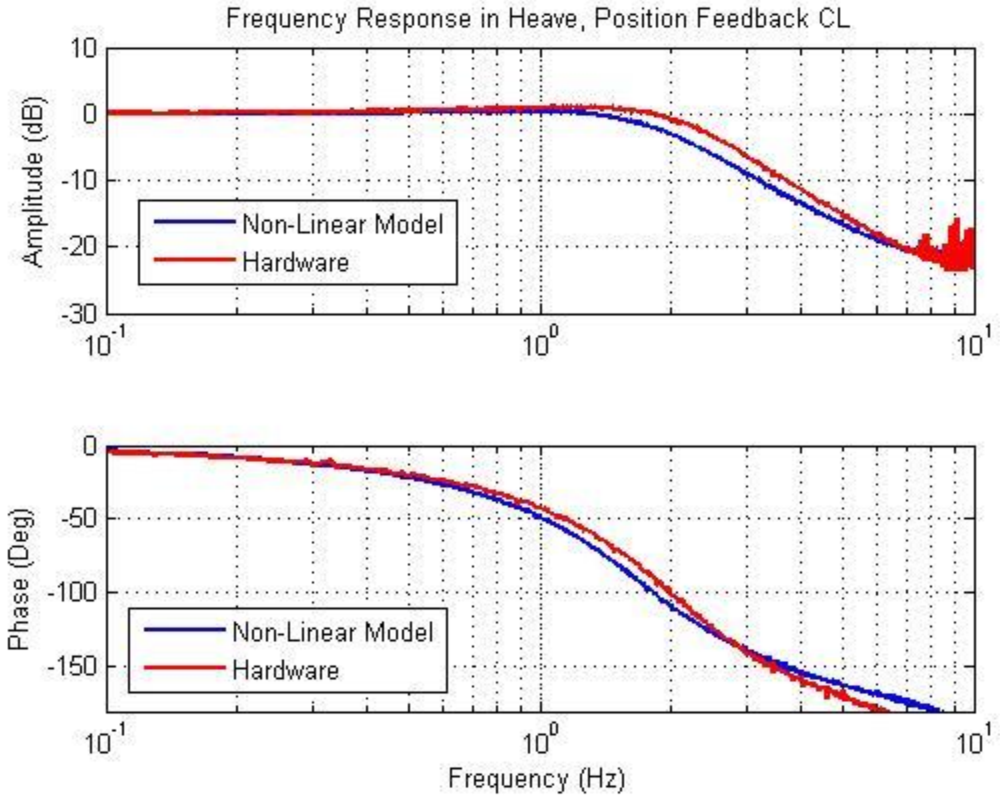

Figure 2-12 - Hardware & Non-linear Model Frequency Response Comparison

In order to show quantitatively how similar the key characteristics of the hardware and non-linear model are, the -90 degree phase bandwidth, -3 dB gain bandwidth, and the damping ratio are shown in Table 2-4.

Table 2-4 - Original Control Law Performance Characteristics

	-90° Phase Bandwidth	-3dB Bandwidth	Damping Ratio
Hardware	1.8 Hz	2.4 Hz	0.5
Non-Linear Model	1.6 Hz	2.0 Hz	0.6

The hardware and software system were also examined through the use of a simple step response. The step response easily shows the damping ratio, rise time and settling time, as well as the steady state error. Since, as discussed earlier, it is accelerations that trigger motion cues a steady-state position error less than 1% in the cab tracking will not cause problems for a pilot's cueing. Figure 2-13 shows the step response comparison between the non-linear model and actual hardware.

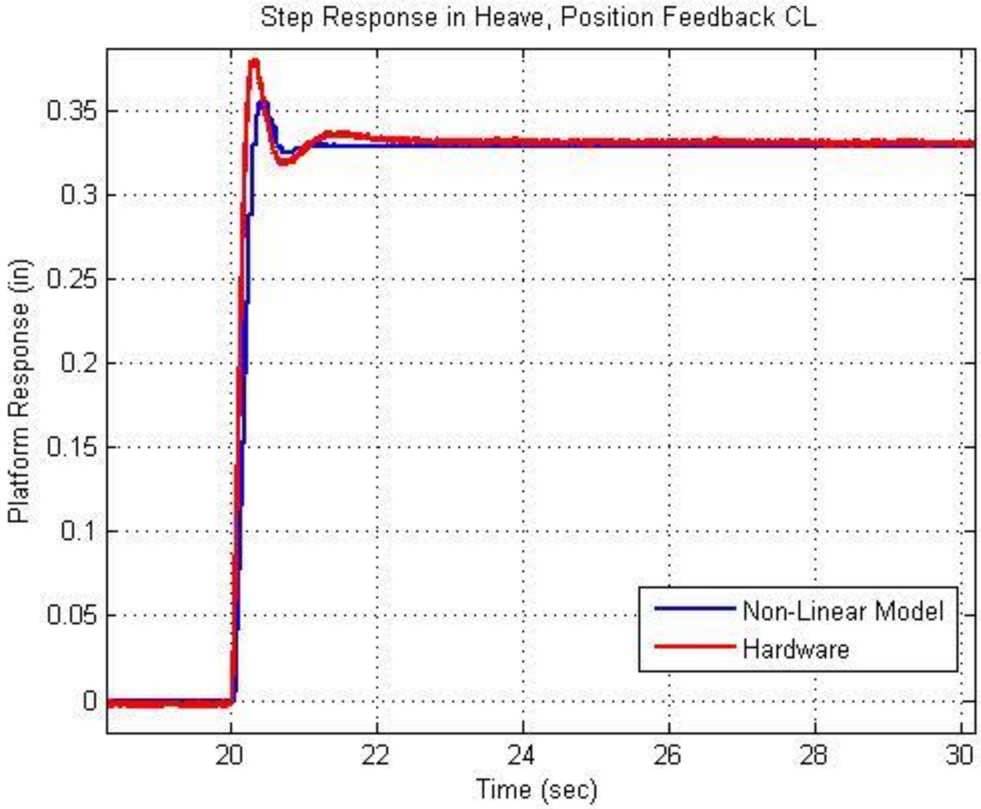

Figure 2-13 - Hardware and Software Step Response Comparison

The only major difference that can be seen is in the damping of the system. It can be seen that the hardware has a damping ratio on the order of 0.5, based on the 15% overshoot of the system. The non-linear model has a damping ratio on the order of 0.6, based on the overshoot of 7%. While characteristics from the original control law were good, the operators of the CMF desired better characteristics, primarily to increase the bandwidth.

The research was performed primarily in the heave DOF. While other degrees-of-freedom are important during flight exercises and even in design in some respects, the focus is always mainly in heave. The reason being heave is the simplest degree-of-freedom to analyze due to the fact that all of the legs experience the same movement, a result of the synergetic nature of the platform. When looking at hardware results any single leg should be identical to any other leg as long as the movement is pure heave. This makes modeling much simpler because only one leg needs to be modeled, however assumptions that every leg, including both valves per actuator, will act identically are not realistic. The error induced by this problem is small enough however that in the current development scheme the problem can be overlooked. Results for frequency hardware responses in the other five degrees-of-freedom for the original compensator, as well as the velocity feedback compensator, can be found in Appendix B.

2.6 Velocity Feedback Compensator

In order to increase the bandwidth of the CMF system an investigation and control law redesign was performed. The new compensator was designed by Zaychik, Cardullo and Hutchinson. The objective of the redesign was to increase the bandwidth while holding the damping ratio to a reasonable level. From classical control system design it is known that adding rate feedback will increase the stability of the system and will also help increase the damping [9]. With an increase in damping the position gain can be increased which is how the bandwidth of the system was increased.

Increasing the position gain helps the control law track the reference signal better; however, the effect also has a negative aspect, that of having an increased overshoot. To implement the rate feedback in the compensator of the CMF, the leg velocities were included as feedbacks and gained. The result was as expected, an increase in bandwidth with limited detriment of the damping ratio. The new velocity feedback compensator system operated in the same manner as the original compensator except for the addition of velocity feedback as seen in Figure 2-14.

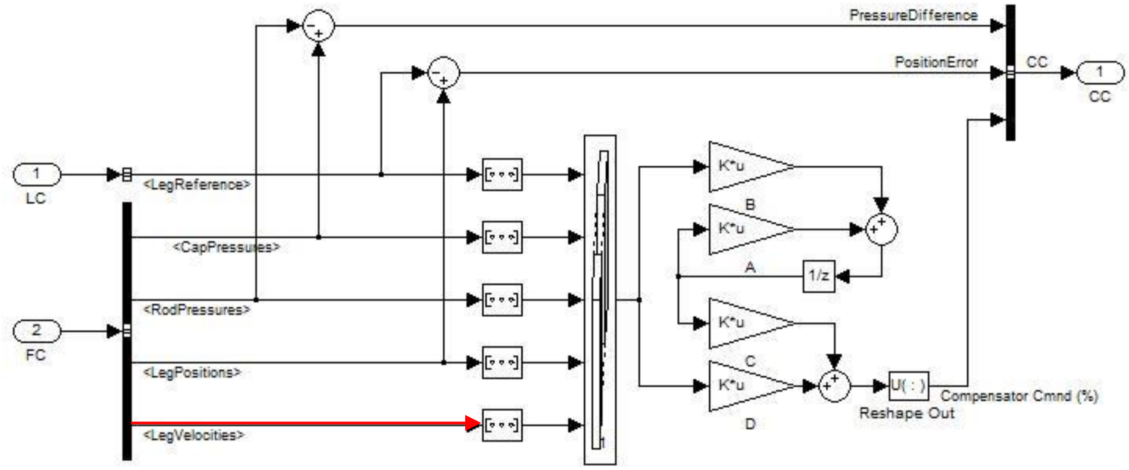

Figure 2-14 - Velocity Feedback Compensator Block Diagram

The velocity feedback compensator gains were modified as well through this new design. The final values used in the velocity feedback compensator design are shown in Table 2-5.

Table 2-5 - Velocity Feedback Compensator Gain Values

Gain A	0.914				
Gain B	0	0	0	1	0
Gain C	-37.1759				
Gain D	47	-1.949e-7	9.816e-8	381.8	-2.3

Again for comparison purposes the frequency response and step response of the hardware and non-linear model were operated and compared.

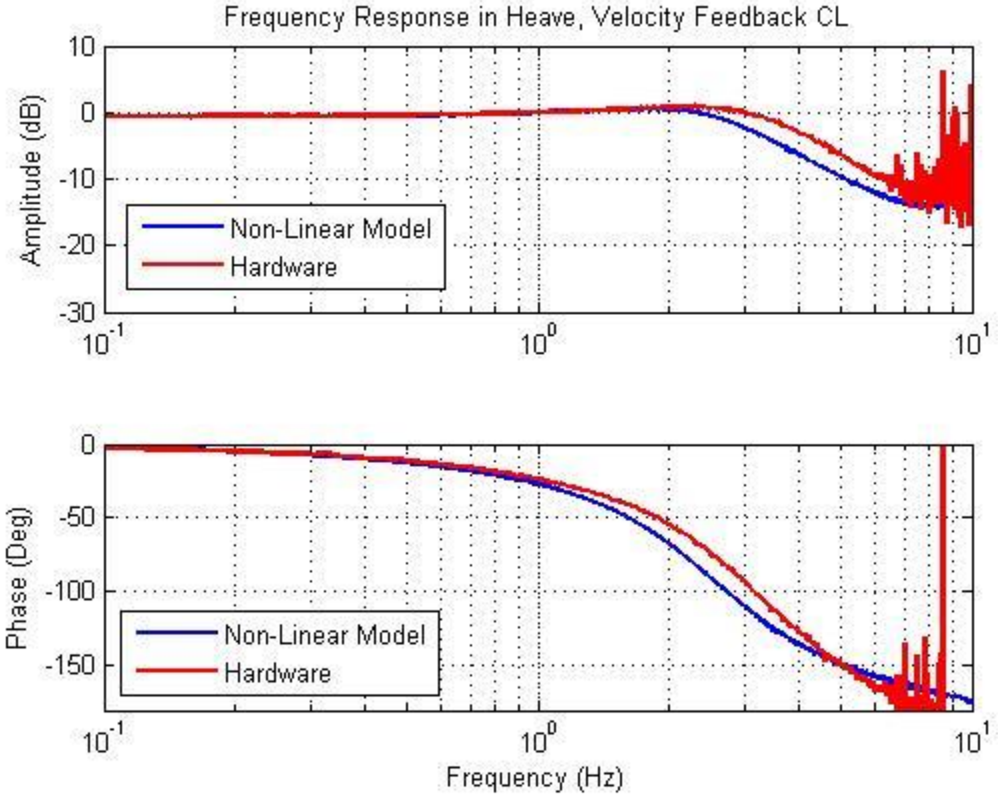

Figure 2-15 - Velocity Feedback Compensator Frequency Response

The frequency response in Figure 2-15 has noise in the response in the higher frequency regions which may cause some alarm. The reason for the noise however has more to do with signal processing and the amount of information density than the effects of the control law. The data used to formulate the frequency response were sampled at 128 Hz and with the final frequency tested being 10 Hz a couple of issues arise in terms of signal processing. The first is that the sampling rate is just sufficient for a frequency input of 10 Hz. The second problem is that there are not enough cycles run at the higher frequencies to generate the information density. The simulator shakes a great deal at the higher frequencies and the operators of the facility do not want to damage equipment on the simulator which is why the number of cycles is limited in high frequencies. A solution to this problem would be using white noise or a chirp input signal, both solutions are being actively pursued. The response to a step input in the heave direction was examined for the updated velocity feedback compensator as well, the results of which are in Figure 2-16.

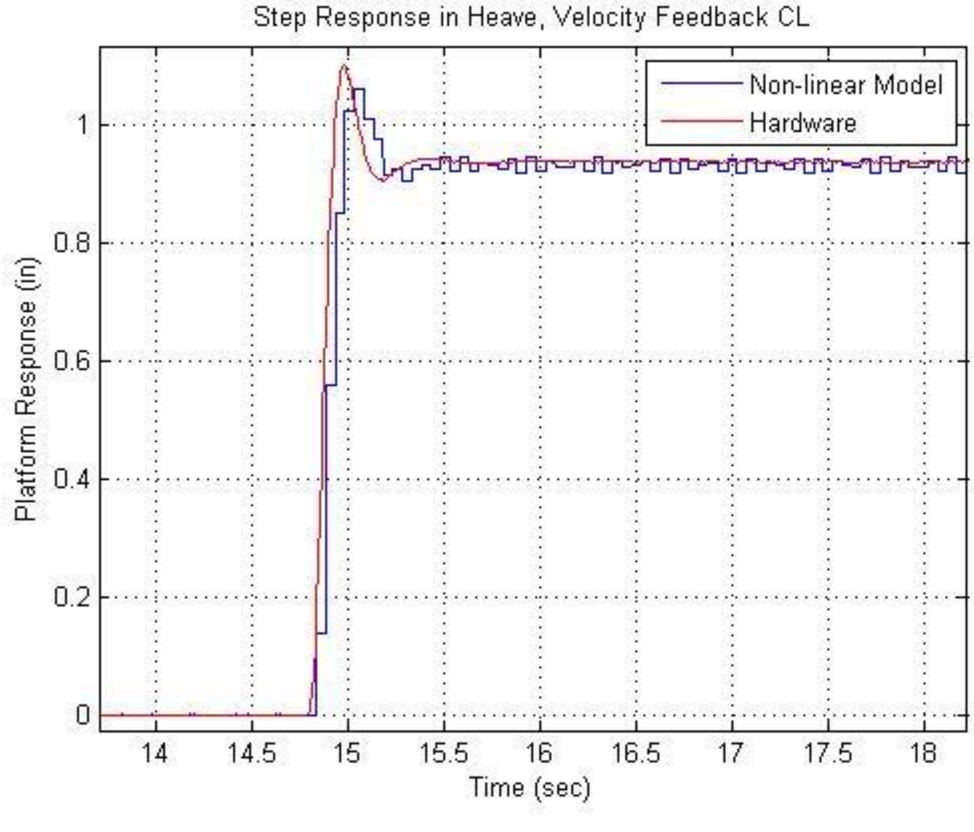

Figure 2-16 - Velocity Feedback Compensator Step Response

The hardware and non-linear model step qualities are very similar which shows that the non-linear model is still performing well as a model of the hardware. The results in terms of performance characteristics are shown in Table 2-6.

Table 2-6 - Velocity Feedback Compensator Performance Characteristics

	-90° Phase Bandwidth	-3 dB Gain Bandwidth	Damping Ratio
Hardware	2.94 Hz	4.0 Hz	0.4
Non-Linear Model	2.5 Hz	3.1 Hz	0.5

From Figure 2-16 it can be seen that a position offset is present. The position error had no impact on the projects performed on the simulator hardware. It was therefore ignored during initial implementation.

Further examination of Table 2-6 illustrates that the non-linear model did not accurately portray the important performance characteristics, underestimating the bandwidths by approximately three quarters of a hertz. The damping ratio with the new velocity feedback compensator was much closer between the non-linear model and the hardware than it was for the original compensator. The velocity compensator was a success and has been implemented on the hardware for normal operation.

2.7 Summary

The previous chapter highlighted many important facets of simulation technology as well as background information with regard to the work documented in this report. The basic axes systems were established to better understand results which are discussed throughout this work. The complete simulator system design was discussed as well including discussions on the DCL, DCU and the motion system. Simulator hardware properties and the non-linear model of the hardware dynamics were presented as well.

An important piece to take away from the DCL, DCU discussion is that the DCL and DCU are unchanged whether operating the hardware or the non-linear dynamic model. The only major difference between operating the system on the hardware versus using the non-linear model is that the non-linear model calculates the feedbacks while the hardware feedbacks are sensor data. Platform performance improvement through the compensator modification is the primary objective of this research and the best way to develop an improved design is utilizing all the tools available, which includes use of the linear models, non-linear models and the actual hardware.

The final discussion in the background section involved the designs of the first two compensators. The original compensator and the velocity feedback compensator were both developed previously. Table 2-6 shows the major results of each compensator. The highlight of which is the hardware result for the velocity feedback result with three hertz of phase bandwidth and a damping ratio of 0.4.

3 NEW COMPENSATOR DESIGN

3.1 Introduction

A major research objective was to design a control law with acceleration feedback, however during piloted runs with the velocity feedback control law, acceleration cross-talk was observed. It was initially thought that this was due to position error of the velocity feedback compensator. Subsequently this was shown not to be the case. The cross-talk was found to be due to the mismatch of actuator performance. A modification to the velocity feedback control law was undertaken to reduce the position error to prevent the possibility of problems being caused by the error.

While the velocity feedback compensator gives three hertz of phase bandwidth, it is desirable to increase the bandwidth beyond three hertz. Increasing the bandwidth is important because it enables more taxing simulations to be performed and thus widening the operational window. Correcting the steady-state error problem in the velocity feedback compensator was another research objective.

Improving the bandwidth of the motion system compensator and removing the steady-state error were treated as two independent tasks throughout this research. The solutions to each problem were resolved independently and are presented in separate sections in this report. The initial steady-state error problem was approached first with the use of integral control as a possible solution. The compensator performance increase, including raising the system bandwidth, was then addressed. The ideas for the compensator improvement were many including new ways of implementing derivative feedback as well as through the use of acceleration or force feedback.

3.2 Integration Compensator

A steady-state position error problem which arises in simulator compensator designs does not normally cause problems in terms of pilot cueing; however, it may cause potential mechanical problems. It is because of these possible mechanical problems that the steady-state error was considered a significant problem and the development of a solution necessary. The simplest method to use for removing steady-state error is using integral control, an approach that is known to typically be successful [9]. For the design the integral of the error signal was chosen. The leg position error signal was chosen because the goal was minimizing the leg position error. The integral controller was therefore designed and tested using the non-linear model before being implemented and tuned on the hardware.

3.2.1 Method

To add position error integral control, the discrete time integrator SIMULINK block was utilized with a gain value of 1, saturation limits of +/-50, and a trigger mechanism. The integrator needed a trigger because while the simulator is down in the stops an error is present before the platform is allowed to rise to the neutral position. This opening error value would cause the platform to lurch which caused a system shutdown. The trigger placement was set that when the platform leg becomes greater than a half inch the switch occurs. The integration input signal was picked off from the position error output which was already available. Figure 3-1 shows the new compensator, the new additions are highlighted with a red border.

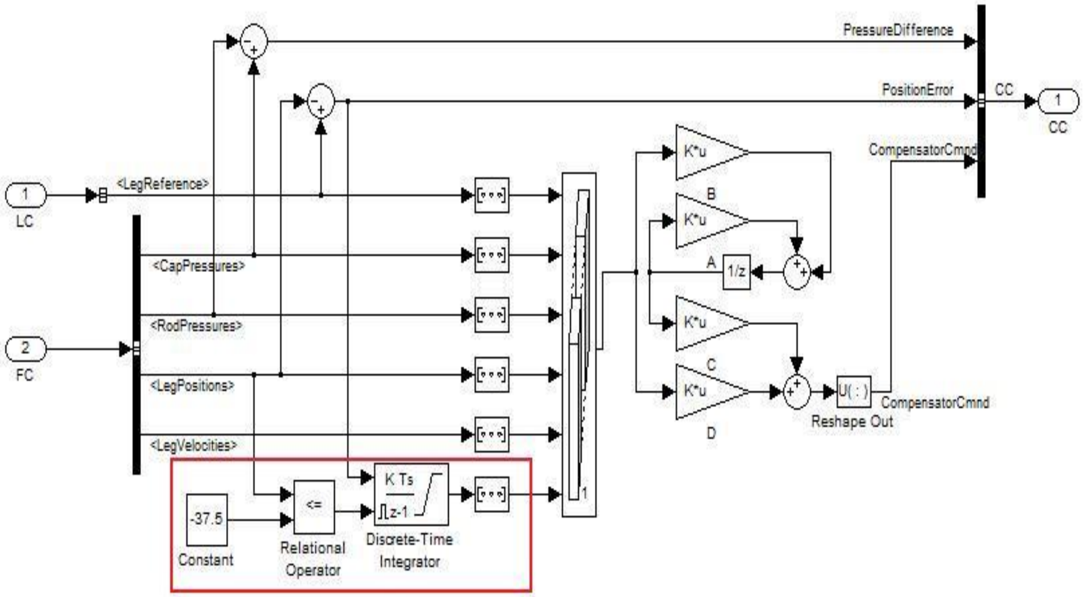

Figure 3-1 - Control Compensator with Integration Feedback

Notice the use of the relational operator which will switch the position output from the constant, -37.5, to the changing leg position once the leg passes by -37.5 inches. While Figure 3-1 shows one example of the discrete time integrator being implemented in the non-linear model, the hardware setup is slightly different. Each leg in the hardware model needs to have an individual discrete time integrator which can be easily setup using a de-multiplexor to separate the signals, apply the integrator, and then multiplex the signals back together. The reason for adding error signal integral control is that the velocity feedback compensator will not need to be adjusted, the only compensator gain change will come from the additional integral gain. To determine the appropriate integral gain, an iterative process was followed.

3.2.2 Results

The results of the iterative process to determine the optimal integral gain are shown in Figure 3-2.

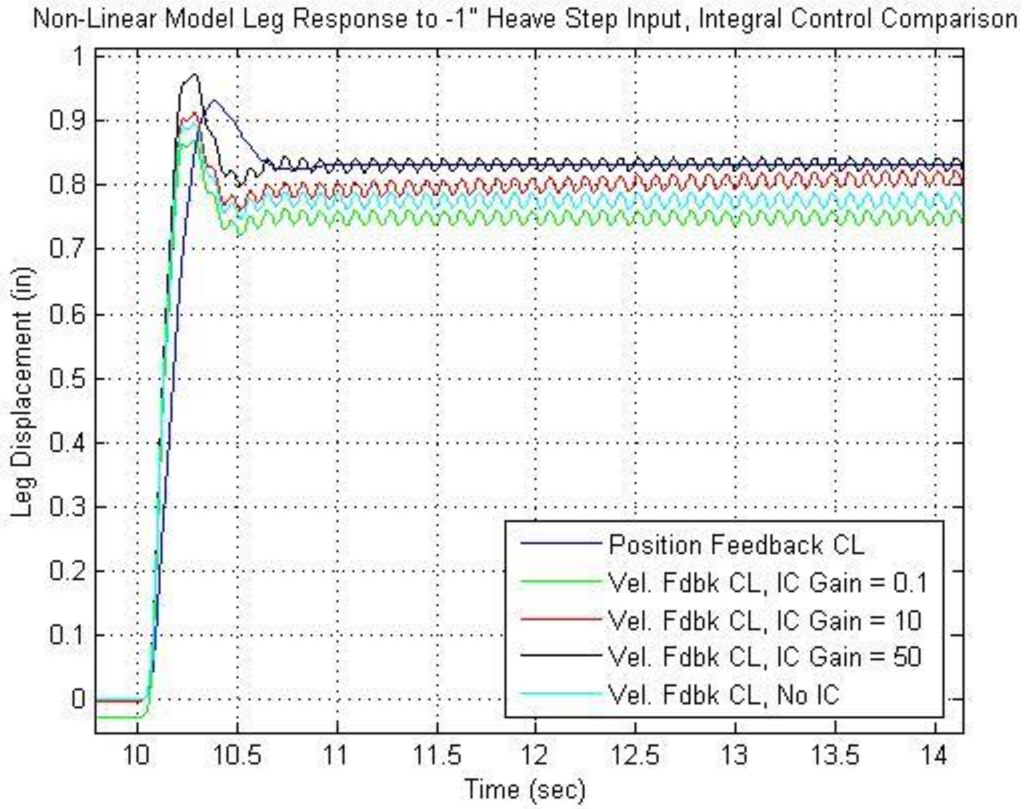

Figure 3-2 - Velocity Feedback with the Addition of Integral Feedback Comparison

The position feedback compensator was included in the analysis for comparison because there was no position error in the position feedback compensator. Figure 3-2 also shows how the addition of integral control does not increase the system overshoot or negatively impact the system in any major way. Figure 3-2 however does have one obvious negative looking characteristic which is the oscillations within the signal. The oscillations were ignored in this case because it was found that the accelerations associated with them were approximately 0.1g which is below the threshold for human detection [5]. The frequency response functions in Figure 3-3 are very similar as well.

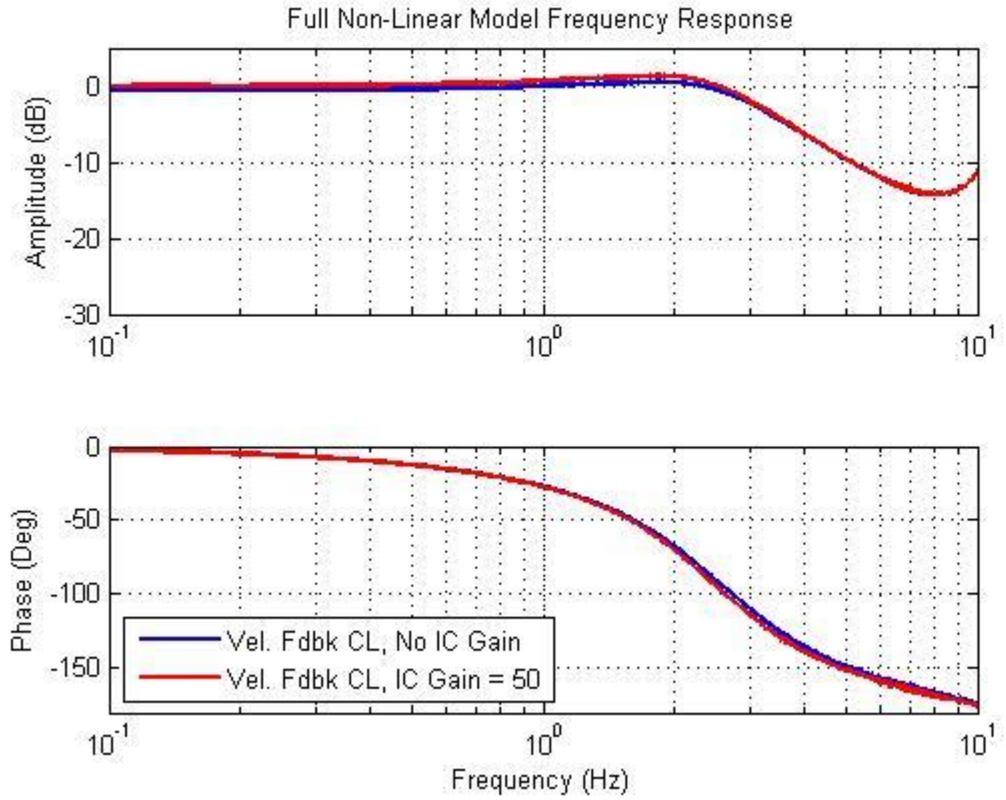

Figure 3-3 - Velocity Feedback and Velocity Feedback with Integral Feedback Frequency Response Comparison

The addition of integral control does not have significant negative effects on bandwidth. The -3 dB amplitude bandwidth was 3.1 Hz for both the original velocity feedback and the velocity feedback with integral feedback. The -90 degree phase bandwidth was 2.4 Hz for the velocity feedback with integral feedback. While the compensator with velocity feedback and no integral control had a phase bandwidth of 2.5 Hz. The steady-state error has all but been removed as well which was the objective of the addition of integral control.

The new integral compensator was then tested on the CMF hardware. The removal of any significant steady-state error was verified through the examination of a one inch step in the heave direction. Figure 3-4 shows the results of the hardware run in degree of freedom space.

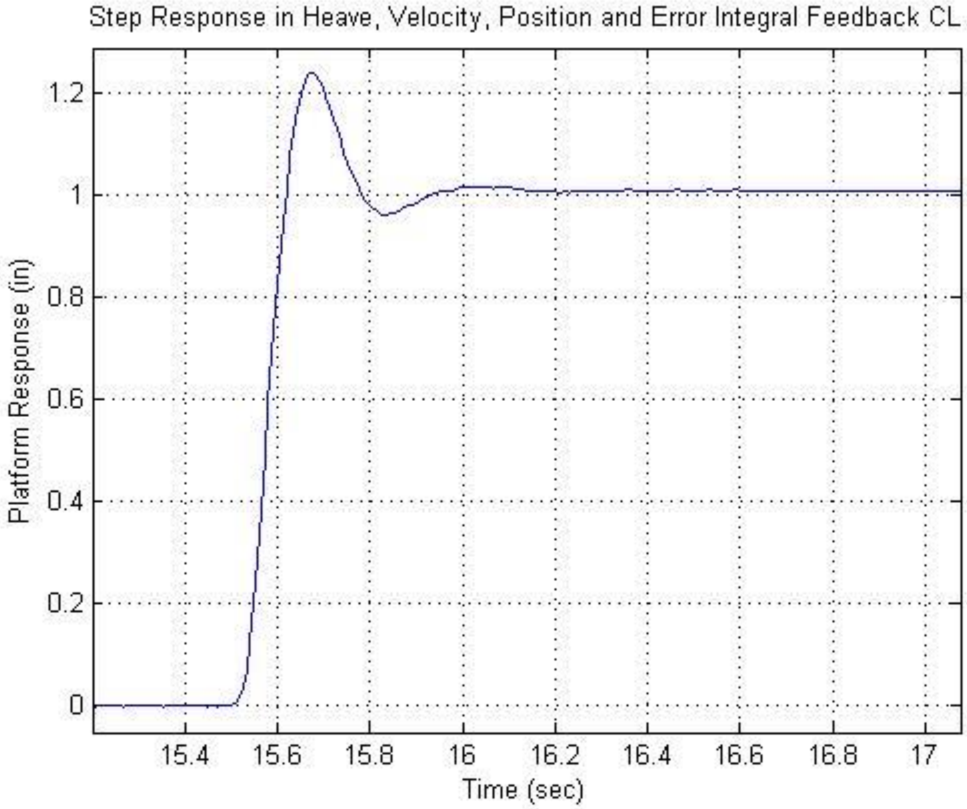

Figure 3-4 - Velocity, Position and Error Integral Feedback Compensator Response

As observed in Figure 3-4, not all of the steady-state position error has been removed; however, an error of less than 1% is considered acceptable. The system overshoots 23%, which corresponds to a damping ratio in the range of 0.4, a value very comparable to the velocity feedback compensator. Examination of the non-linear model results verifies that the use of position error integral control will eliminate steady-state error.

3.3 Position Error Derivative Feedback Compensator

After verifying that the addition of integral control would remove steady-state error, the next step to improve the system needed to be executed. The first step in the system improvement process was determining possible methods of improvement and the implementation on the non-linear model. It was determined that while true leg position velocity feedback is one type of rate feedback a few other rate feedback methods could be implemented and a combination of methods as well.

3.3.1 Method

Three new types of rate feedback were examined, position error derivative feedback, velocity error feedback, and a combination of velocity feedback and position error rate feedback. The position error derivative feedback technique was implemented using the position error signal which has the derivative taken of it and is then fed back into the compensator in place of the leg velocity feedback signal. Figure 3-5 shows the design at the compensator level and Figure 3-6 shows how the derivation was computed.

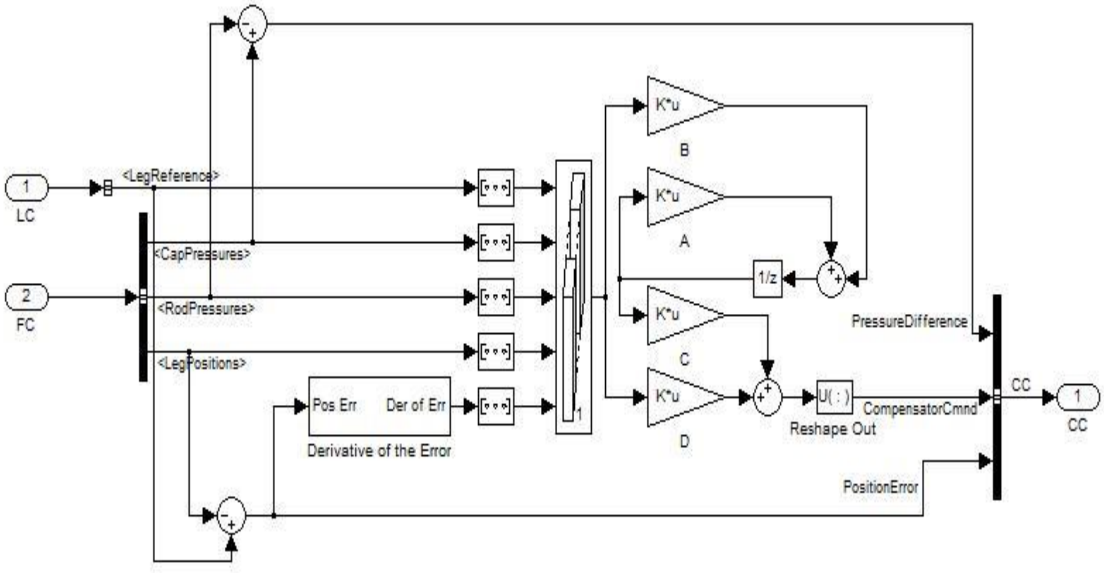

Figure 3-5 - Block Diagram of Error Derivative Feedback

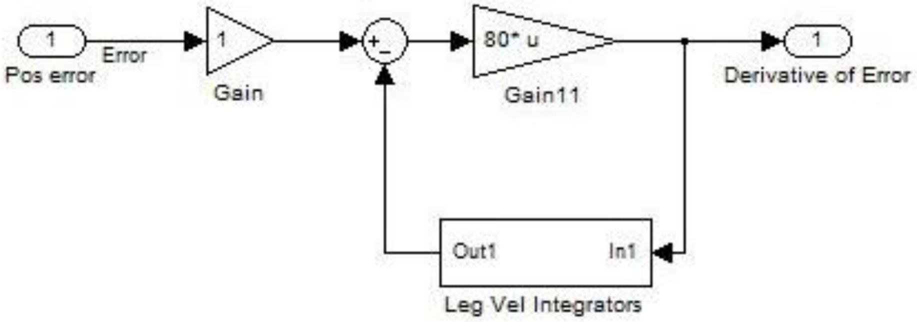

Figure 3-6 - Block Diagram of Derivative Implementation

The reason that discrete time integrators were used with the configuration shown in Figure 3-6 was that when taking the derivative noise was introduced into the system.

To mitigate the problem integrators were introduced and the integrated signals were compared to the input signal in the manner shown in Figure 3-6. This technique provided the desirable position error rate signal.

The second new rate implementation involved developing an error signal of velocity. The technique was implemented in a slightly different manner. This implementation, as shown in Figure 3-7, compared the feedback leg velocities to the leg reference signal derivative which is how the velocity error signal was developed.

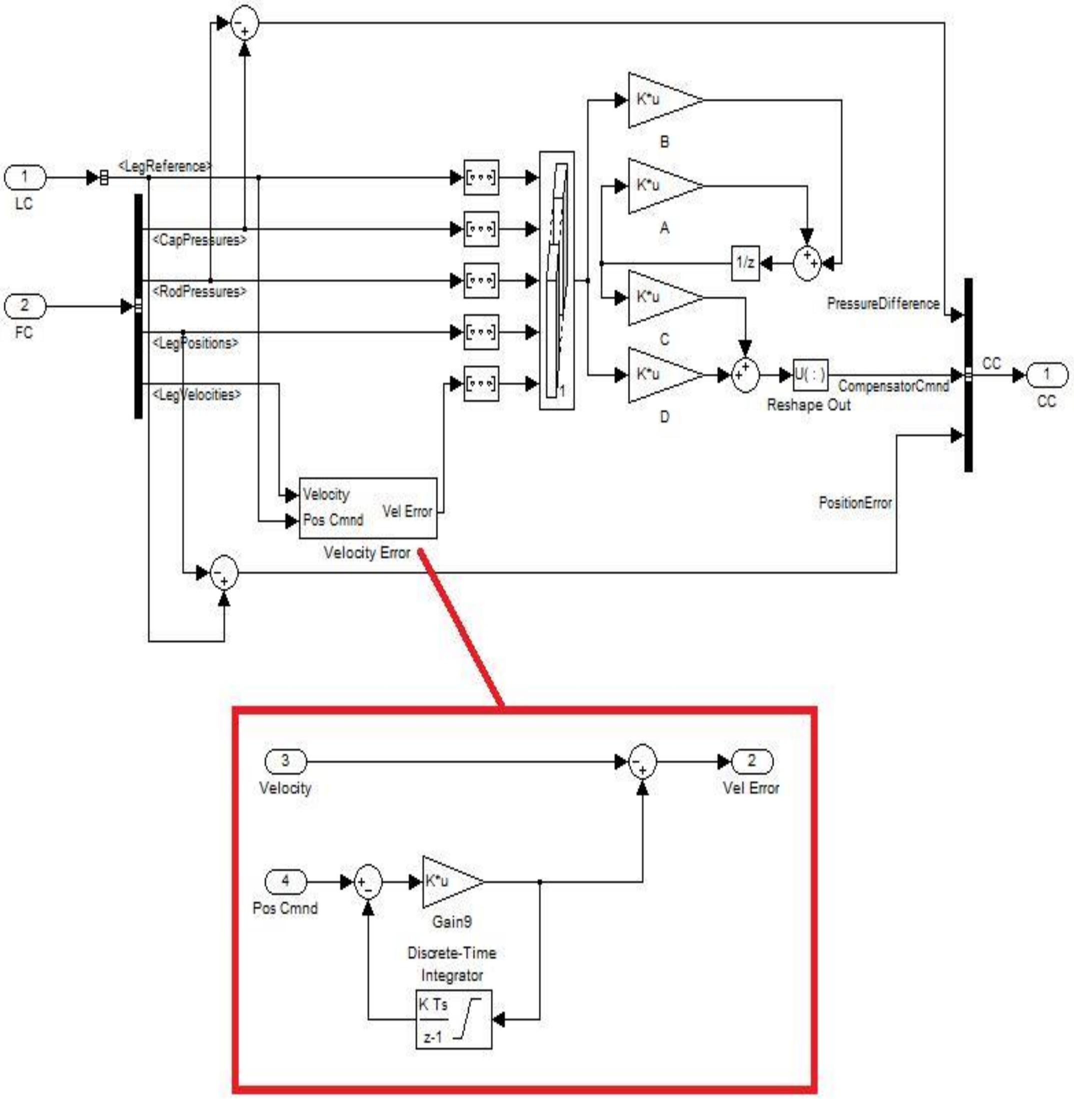

Figure 3-7 - Block Diagram of the Velocity Error Compensator Design

The velocity error feedback design was developed by creating a velocity error signal, obtained by comparing the actual leg feedback velocity signal and the derivative of the leg reference.

The final rate feedback method developed using the non-linear model was a combination of the position error derivative and the leg velocity feedback signals. This final method shown in Figure 3-8 was intended to investigate whether or not the advantages of the velocity feedback and position error rate feedback could combine for a larger improvement.

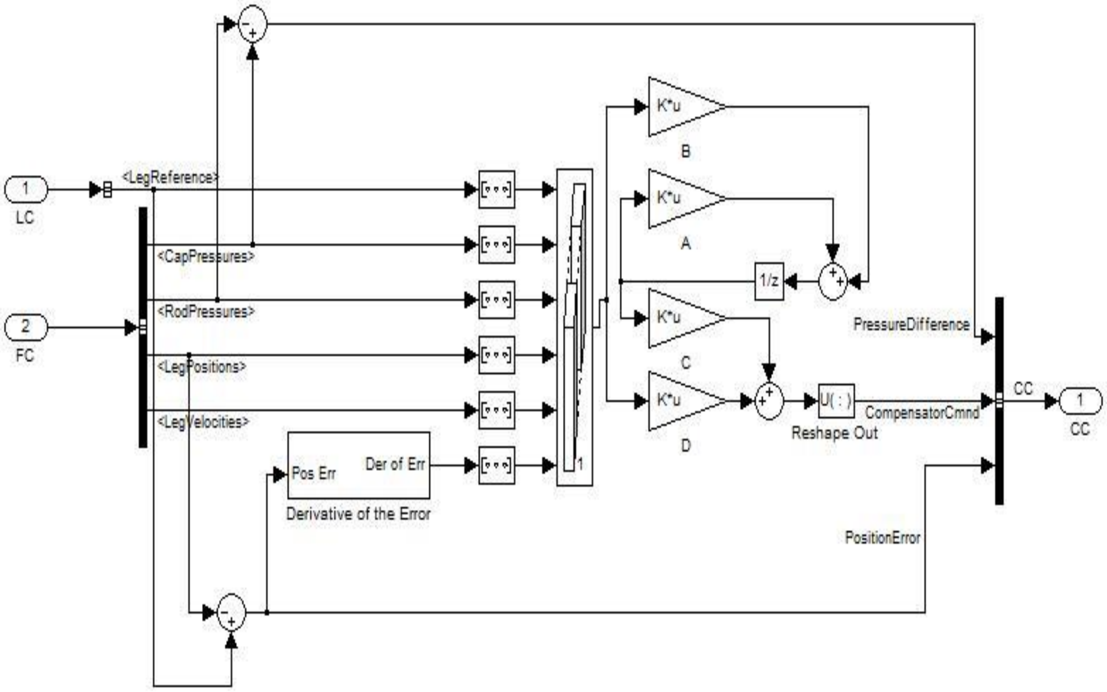

Figure 3-8 - Rate Feedback Combination Compensator Block Diagram

The testing of each technique involved considering responses to step inputs in heave as well as frequency sweeps in heave. The step responses are intended to show how well the new compensator can react to almost instantaneous inputs and give a clear indication of the system damping, while the frequency sweep data can be used to produce frequency response functions which will show the system bandwidth in phase and amplitude.

3.3.2 Results

The optimal gain sets for each of the rate feedback techniques are shown below in Table 3-1, the values of which were obtained through an iterative process.

Table 3-1 - Optimal Gain Sets for Rate Feedback Compensators

	Position Error Rate Feedback (PERFB) Optimal Gain Set					
Gain A	0.914					
Gain B	0	0	0	1	0	
Gain C	-37.1759					
Gain D	49	-1.949e-7	9.816e-8	381.8	2.2	
	Velocity Error Feedback (VEFB) Optimal Gain Set					
Gain A	0.914					
Gain B	0	0	0	1	0	
Gain C	-37.1759					
Gain D	49	-1.949e-7	9.816e-8	381.8	2.8	
	Position Error Rate Feedback and Leg Velocity Sensor Feedback Optimal Gain Set					
Gain A	0.914					
Gain B	0	0	0	1	0	0
Gain C	-37.1759					
Gain D	49	-1.949e-7	9.816e-8	381.8	2.2(PERFB)	-0.5(gain on Velocity feedback)

The first comparison between the rate feedback techniques is through the step input. Figure 3-9 shows the comparison of the three new rate feedback techniques as well as the original compensator and the velocity feedback compensator.

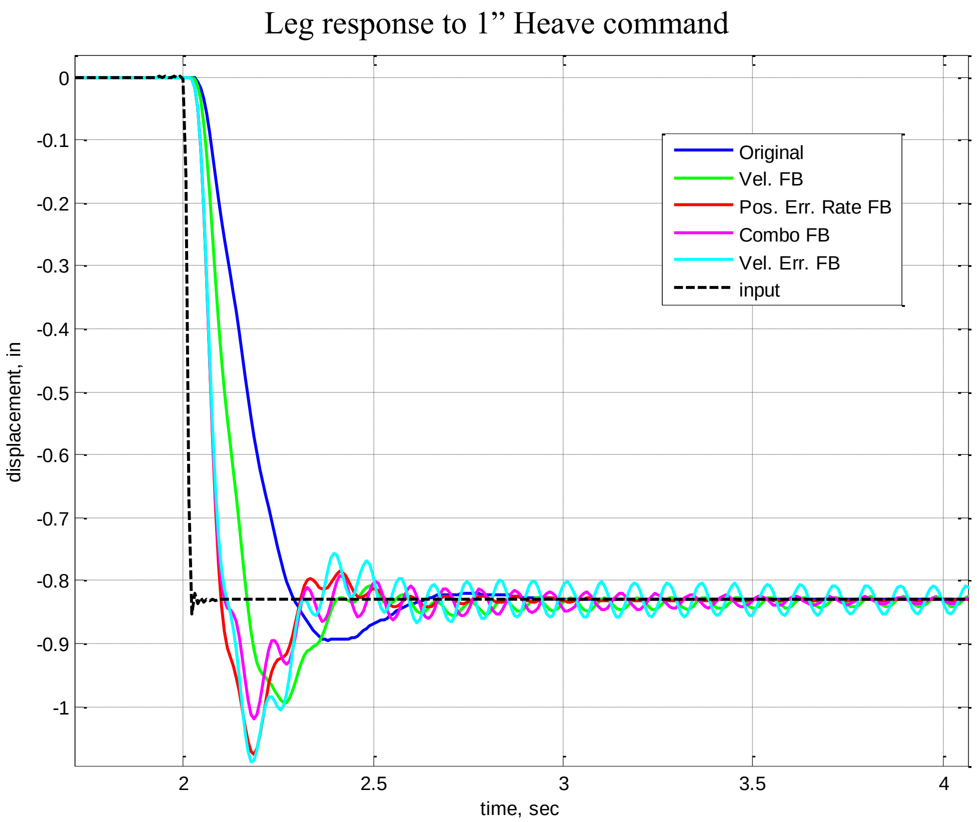

Figure 3-9 - Non-Linear Model Leg Response to 1 Inch Heave Step Input, Comparison of Rate Feedback Techniques

It can be easily noted that the newly designed control law feedback configurations do not result in steady-state error due to integral control loop described in 3.2. The position error rate feedback and the velocity error feedback compensators both improve the response time to the step input, while also having worse damping characteristics.

The next step in evaluating the quality of the new compensators is to examine a simple sinusoidal input. Figure 3-10 shows the leg responses for each of the rate feedback techniques including velocity feedback as well as the original compensator and indicates the benefits of each compensator to such a simple sinusoidal input.

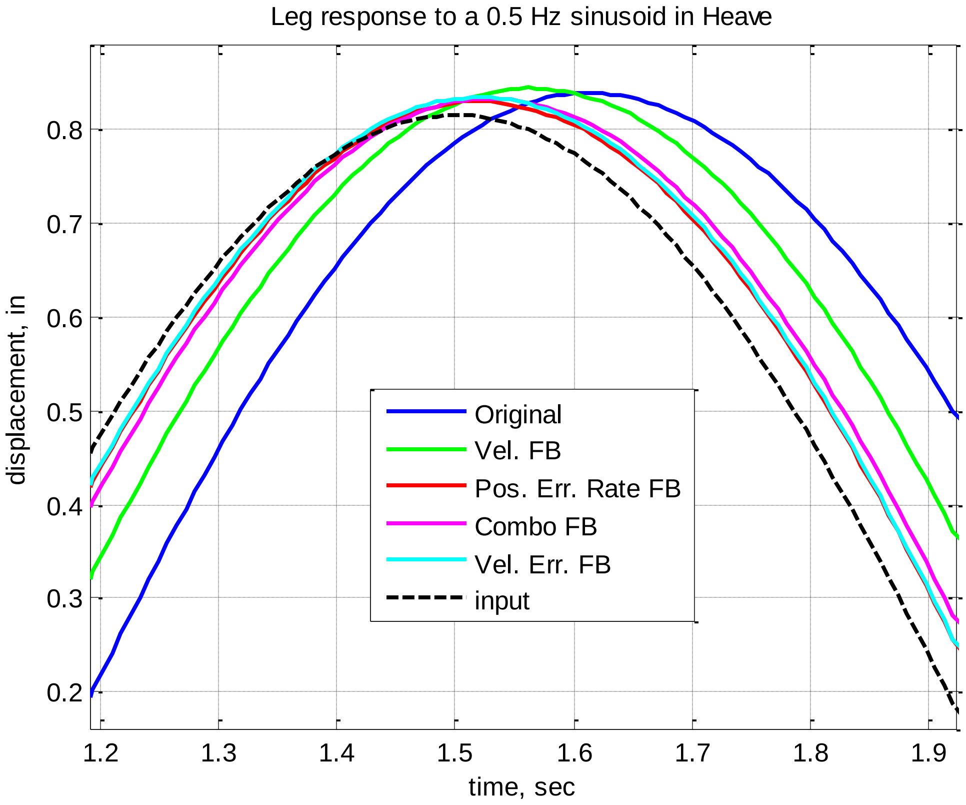

Figure 3-10 - Non-Linear Model Leg Responses to 1 Inch 0.5 Hertz Sinusoidal Input, Comparison of Rate Feedback Techniques

The position error rate feedback and velocity error feedback compensators give the best responses. The pure velocity feedback has more error and actually overshoots the input signal.

The frequency response comparison in Figure 3-11 of the different rate feedback compensators shows a much clearer picture of the quality of each compensator.

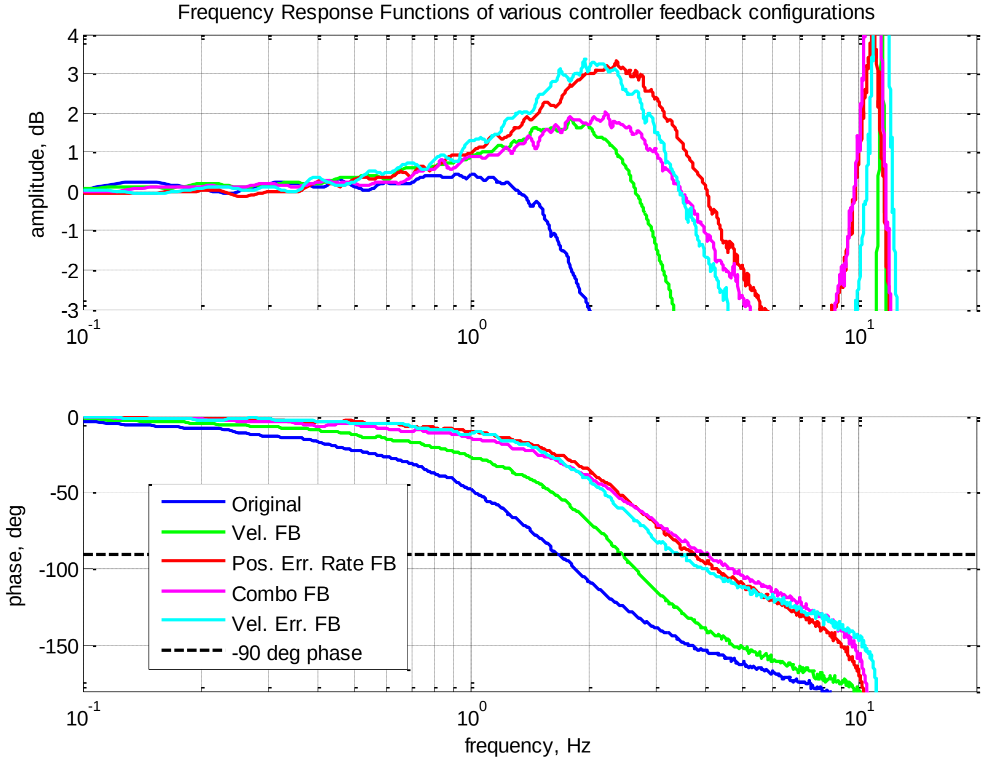

Figure 3-11 - Non-Linear Model Frequency Response, Comparison of Rate Feedback Techniques

From Figure 3-11 it can be seen that position error rate feedback or the combination feedback provide the performance edge both in terms of phase and amplitude bandwidths. Table 3-2 contains condensed performance characteristics of all feedback configurations being compared.

Table 3-2 - Rate Feedback Technique Non-Linear Model Performance Results

	Damping	Phase Bandwidth	Amplitude Bandwidth
Position Error Rate Feedback	0.4	3.8 Hz	5.5 Hz
Velocity Error Feedback	0.4	4.0 Hz	5.0 Hz
Combination of Rate Feedbacks	0.45	3.5 Hz	4.5 Hz

3.3.3 Summary

From Table 3-2 a number of conclusions with respect to the developed rate feedback models can be made. First, the combination of rate feedback forms, which shows equivalent performance to the position error rate feedback, must be evaluated.

The gain values used for each of the rate feedback inputs in the combination cancel out and therefore do not produce better results than the position error rate feedback independently. The second, more important result is that the position error rate feedback results, especially those in Figure 3-11, outperform the velocity feedback technique.

The position error rate feedback compensator has less damping than the velocity feedback compensator, but the amount of phase bandwidth is greater. Since the number one objective of this research is increased bandwidth, raising the bandwidth from 2.5 Hz to 3.8 Hz is worth noting. An improvement of more than one hertz of bandwidth, with only a loss of 0.1 in damping is why the position error rate feedback is considered the best type of rate feedback. Hardware results often show more improvement in bandwidth gains as well, but can under-predict damping. It is therefore recommended that the position error rate feedback compensator be evaluated on the hardware.

3.4 Acceleration and Force Feedback Compensator

It is expected that adding force or acceleration feedback should improve the system. The possibility of system improvement was determined through simple analysis of the transfer function's characteristic equation. To show this a general plant model was combined in a control system with proportional, rate and acceleration feedback as shown in Figure 3-12.

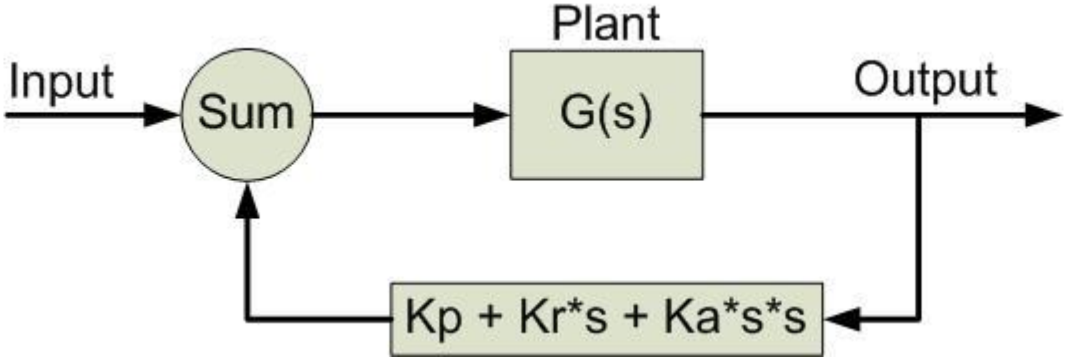

Figure 3-12 - Block Diagram of Control System with Proportional, Rate and Acceleration Feedback

The generic plant is of the following form:

$$G(s)=\frac{\omega_n^2}{s^2+2\omega_n\zeta s+\omega_n^2} \quad (3.1)$$

The natural frequency is ω_n, and ζ is the system's damping ratio. The general plant will help show the impact of adding position, velocity and acceleration feedback. The result of adding position feedback is an increase in natural frequency with an increase in K_p which can be seen in Equation 3.2.

$$G_1(s)=\frac{G(s)}{1+G(s)}=\frac{\omega_n^2}{s^2+2\omega_n\zeta s+\left(1+K_p\right)\omega_n^2} \tag{3.2}$$

Examining the characteristic equation of $G_1(s)$ it is apparent that the system bandwidth can be increased by increasing the proportional gain and thus increasing the natural frequency. The next step was to examine the addition of velocity feedback. From basic control theory it is expected that the addition of velocity feedback to a linear system will increase damping allowing for an increase in bandwidth. Equation 3.3 shows the new transfer function with the addition of velocity feedback.

$$G_2(s)=\frac{G_1(s)}{1+\left(K_r s\right)G_1(s)}=\frac{\omega_n^2}{s^2+\left(2\omega_n\zeta+K_r\omega_n^2\right)s+\left(1+K_p\right)\omega_n^2} \tag{3.3}$$

The damping can be increased through the addition of velocity feedback which is evident by examining the characteristic equation. The system damping is the term in front of the s in the characteristic equation. With the addition of K_r the damping can be increased.

The final addition, and for this forthcoming section, the most important addition to examine is the acceleration feedback. A new transfer function is formed through the addition of acceleration feedback and can be seen in Equation 3.4:

$$G_3(s)=\frac{G_2(s)}{1+\left(K_a s^2\right)G_2(s)}=\frac{\omega_n^2 \big/ \left(1+K_a\omega_n^2\right)}{s^2+\frac{\left(2\omega_n\zeta+K_r\omega_n^2\right)}{\left(1+K_a\omega_n^2\right)}s+\frac{\left(1+K_p\right)}{\left(1+K_a\omega_n^2\right)}\omega_n^2} \tag{3.4}$$

The benefit of adding acceleration feedback can be seen through examination of the characteristic equation. The acceleration feedback gain will impact the total system gain, the damping and the natural frequency. This proves that pursuing the addition of acceleration feedback may lead to a performance increase. The addition of acceleration feedback was further examined through tests using a simple linear model to show the frequency response comparison. The model, which included a position, velocity and acceleration component, is shown below in Figure 3-13.

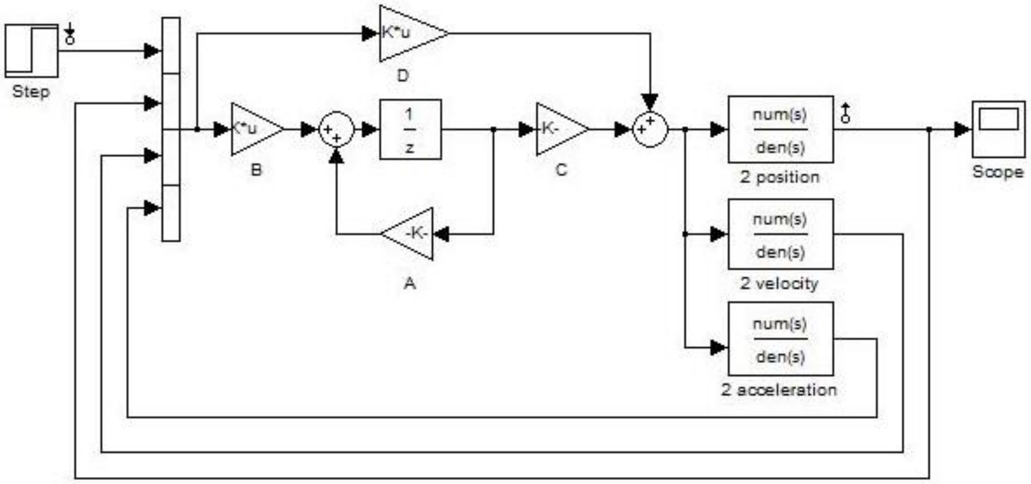

Figure 3-13 - Block Diagram of Simple Feedback Control Law Structure

The model in Figure 3-13 includes the compensator setup. Model linearization points have been added to enable a simple bode response to be found using internal SIMULINK tools. The bode magnitude response is shown below in Figure 3-14.

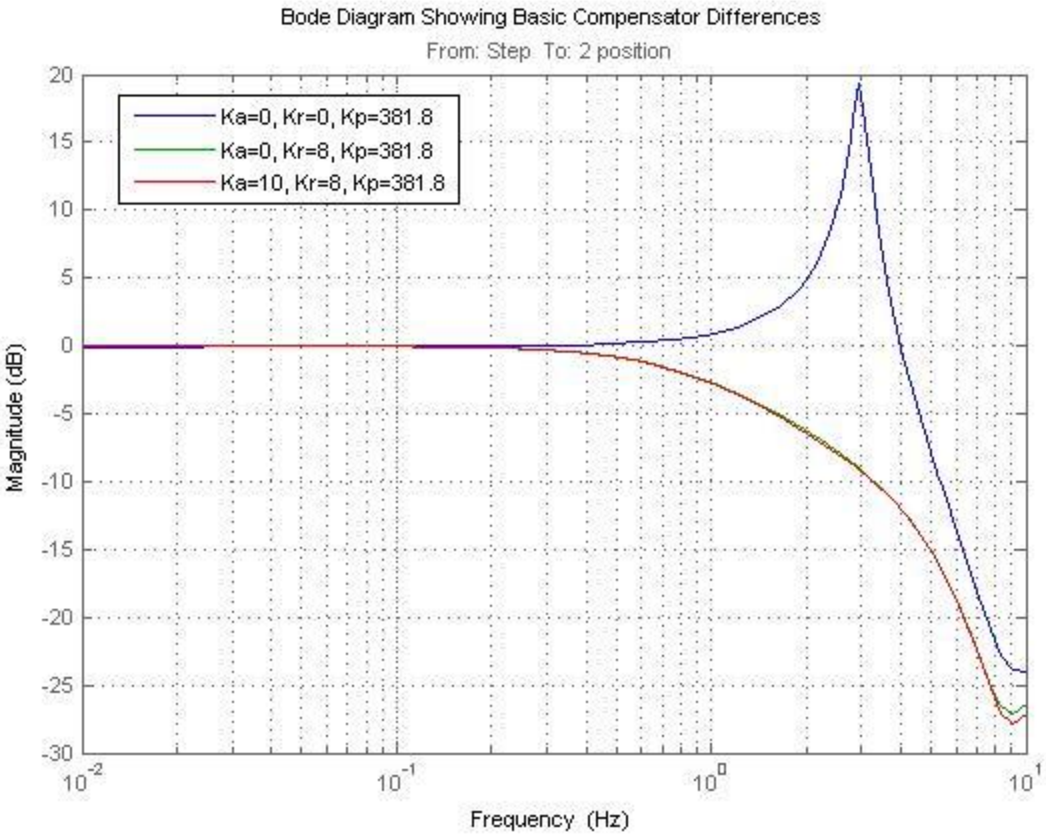

Figure 3-14 - Bode Diagram Comparison from a Simple Linear Model using Various Compensator Techniques

While it is a little difficult to see the difference between the rate and acceleration feedbacks in Figure 3-14 this can be considered another positive of adding acceleration feedback. It is clear that the addition of rate feedback is what improved the damping. Adding acceleration feedback however does not significantly deteriorate the damping or decrease the bandwidth. The important bandwidth characteristic in the -3 dB range is 1.08 Hz for position and velocity feedback while the addition of acceleration feedback is 1.06 Hz. What is not able to be shown in this simple implementation is that improvements are expected more in the rotational degrees of freedom due to the nature of force feedback. However, the following work will only be able to show the benefits in the heave direction adequately because the non-linear model does not account for the dynamic mass of the system. Therefore if the heave degree-of-freedom is improved on the non-linear model, then the acceleration and force feedback compensators should be tested on the hardware with a focus on the rotational degrees of freedom.

3.4.1 Method

The acceleration feedback method was developed using both the non-linear simulation as well as the hardware. Force accelerometers are inherently noisy which is why it was necessary to devise a filter to eliminate the high frequency noise. Figure 3-15 shows the noise in the hardware accelerometers while sitting at rest.

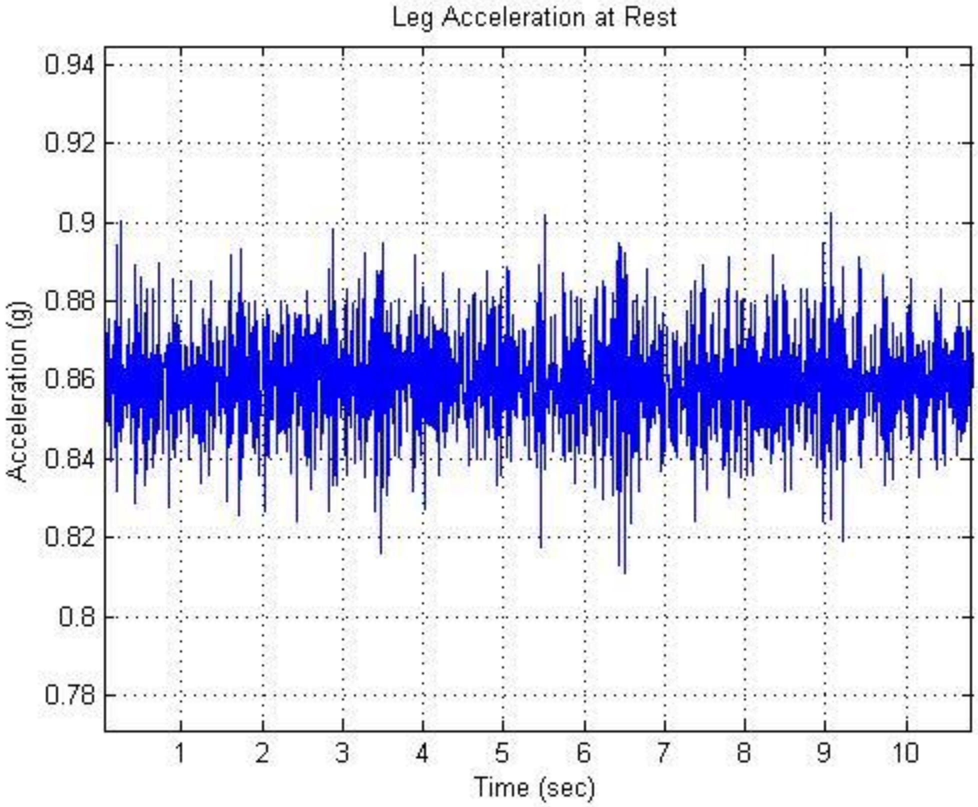

Figure 3-15 - Example of Noise in Acceleration Sensors

Figure 3-15 also shows the accelerometer signal bias. The signal has bias as a result of the accelerometer being fixed to the actuator and aligned along the axis of motion. The accelerometers will register one g of acceleration if positioned upright in the gravity plane. For the simulator's actuator the leg is at an angle relative to the gravity field which causes the bias and is a function of actuator extension and angle. To improve testing using the non-linear SIMULINK model an update was researched. The proper transformations of the gravity vector were computed and applied in the SIMULINK model through other research performed in the Man-Machine Systems Laboratory. The transformation, however, was added to the model after all of the results presented in this report were computed. Included in APPENDIX C are the derivation and application of the transformation to the non-linear SIMULINK model. Due to the large amount of noise in the accelerometer signal a filter was designed. Development of the filter was based off a simple two pole Butterworth filter. Recorded output acceleration signals were used as the development signals, this was done because the noise was real and it was easy to see the amount of lag. The filters were designed in continuous time; this made making the fine-tuning adjustments simpler. The filters were then transformed into discrete time using MATLAB through the zero-order-hold technique with a sampling time of 1/2048 seconds. The final filters, first in continuous-time followed by discrete-time are:

$$filter_{CT} = \frac{0.312}{0.2s + 0.312} \tag{3.5}$$

$$filter_{DT} = \frac{0.007588}{z - 0.9924} \tag{3.6}$$

It took a number of fine-tuning iterations to obtain the final filter, however as seen in Figure 3-16 the final filter design removes much of the noise while keeping signal attenuation to a minimum. To show the usefulness of the filter during operation a sinusoidal input with a frequency of 1.2 Hz and amplitude of 0.33 inches was used instead of just showing the platform at rest as shown in Figure 3-15.

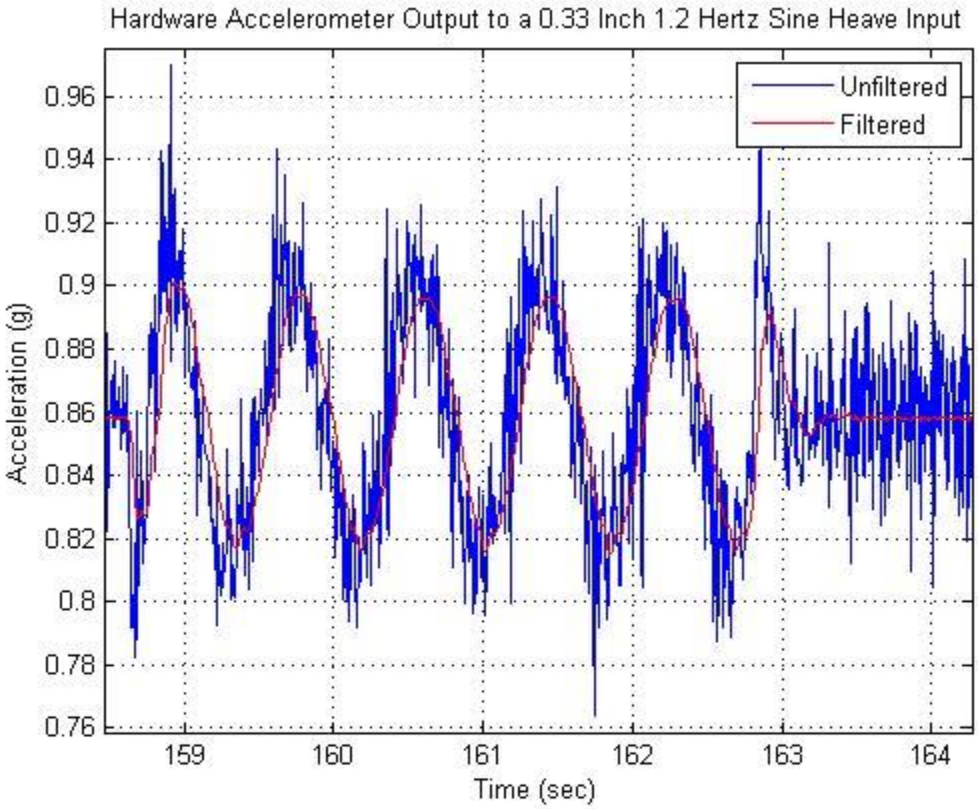

Figure 3-16 - Noise Cancellation Result for the Accelerometer Filter

The delay is kept to approximately one hundredth of a second while the signal maintains the general signal form while eliminating the noise which was determined to be adequate for use in the acceleration feedback compensator. With the filter properly designed it was then added to the compensator.

The next problem faced in developing the acceleration feedback compensator was the sensor offset. The hardware sensors report the effect of 0.86g of acceleration in actuator space. The acceleration signals in the non-linear model are from the actuator force plus the static load divided by the effective mass. Figure 3-17 shows how the acceleration, velocity and position signals are calculated and fed back in the non-linear model.

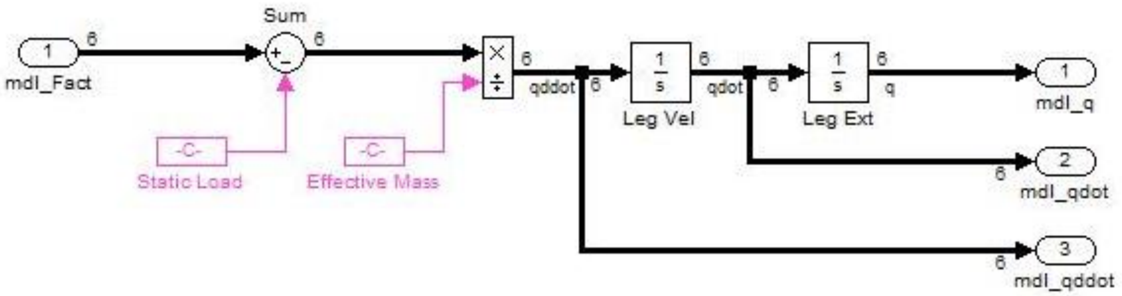

Figure 3-17 - Acceleration, Velocity and Position Feedback Signals Calculated in Non-Linear Model

This is important because the acceleration signal within the non-linear model will lack an offset. The offset will be impacted by the position of the platform which is not included in the non-linear model at this time and is why, for the initial design phase, the offset is ignored. When the acceleration feedback compensator is adequate in the non-linear model, then the values can be taken to the hardware and modified to compensate for the offset which occurs in the hardware.

The acceleration feedback compensator setup included not only acceleration feedback but also position feedback, commanded position, rod and cap pressure feedback as well as position error rate feedback. Figure 3-18 shows the system setup including the accelerometer filter.

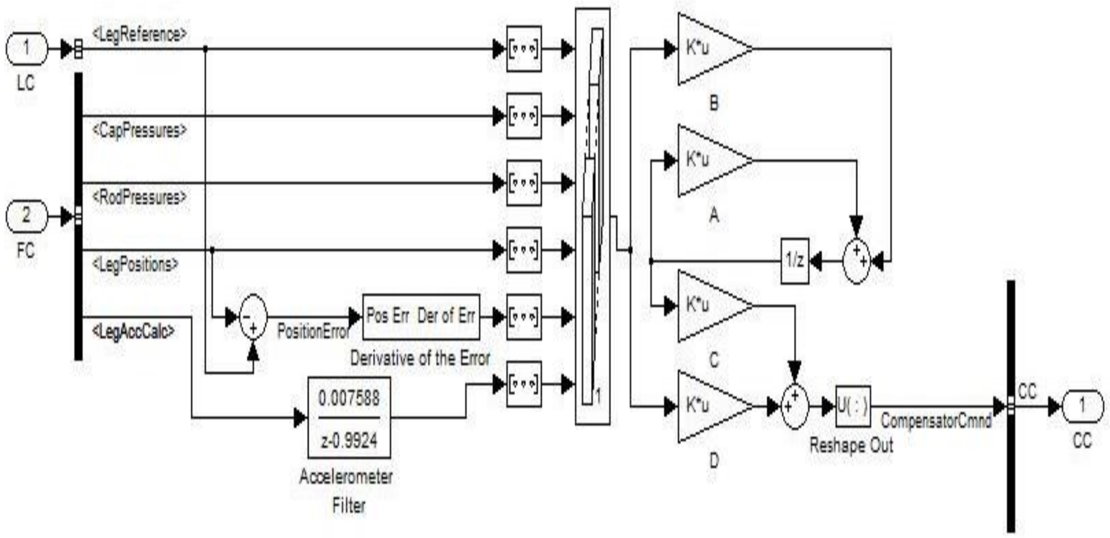

Figure 3-18 - Block Diagram of the Acceleration Feedback Compensator

As with the previous compensator updates the gain values on the feedback signals have been modified as well. Modification of the A, B, C and D gain blocks is necessary both in the non-linear model as well as when tuning the compensator on the hardware. Table 3-3 shows the gains for each of the matrices for the final acceleration feedback compensator as developed on the non-linear model.

Table 3-3 - Gain Table for Acceleration Feedback Compensator

Gain A	0.914					
Gain B	0	0	0	1	0	0
Gain C	-37.1759					
Gain D	49	-1.949e-7	9.816e-8	381.8	0.01	2.2

The implementation of force feedback faced a similar problem to that of the acceleration feedback. There is a static force difference between the non-linear model and the hardware. A second issue with force feedback is the variability of the pressures used in force feedback. Since the force is calculated using the cap and rod pressure values which change almost instantaneously it makes using force feedback a greater challenge.

The first step taken to design the force feedback compensator was developing a method to obtain the force feedback. From the actuator geometry and pressure values the force was determined using:

$$\begin{aligned} F &= (\Delta P) * A \\ A &= \frac{\pi}{4}\left(d_o^2 - d_i^2\right) \end{aligned} \tag{3.7}$$

Where F is the force, ΔP is the pressure difference between the cap and rod pressure, A is the area the pressure is acting over, d_o is the actuator bore diameter and d_i is the actuator rod diameter. Equation 3.7 can be implemented in the non-linear SIMULINK model and applied to obtain force feedback. The inside and outside actuator diameters are constants and the cap and rod pressures are available as feedbacks. The final force feedback compensator setup can be seen in Figure 3-19.

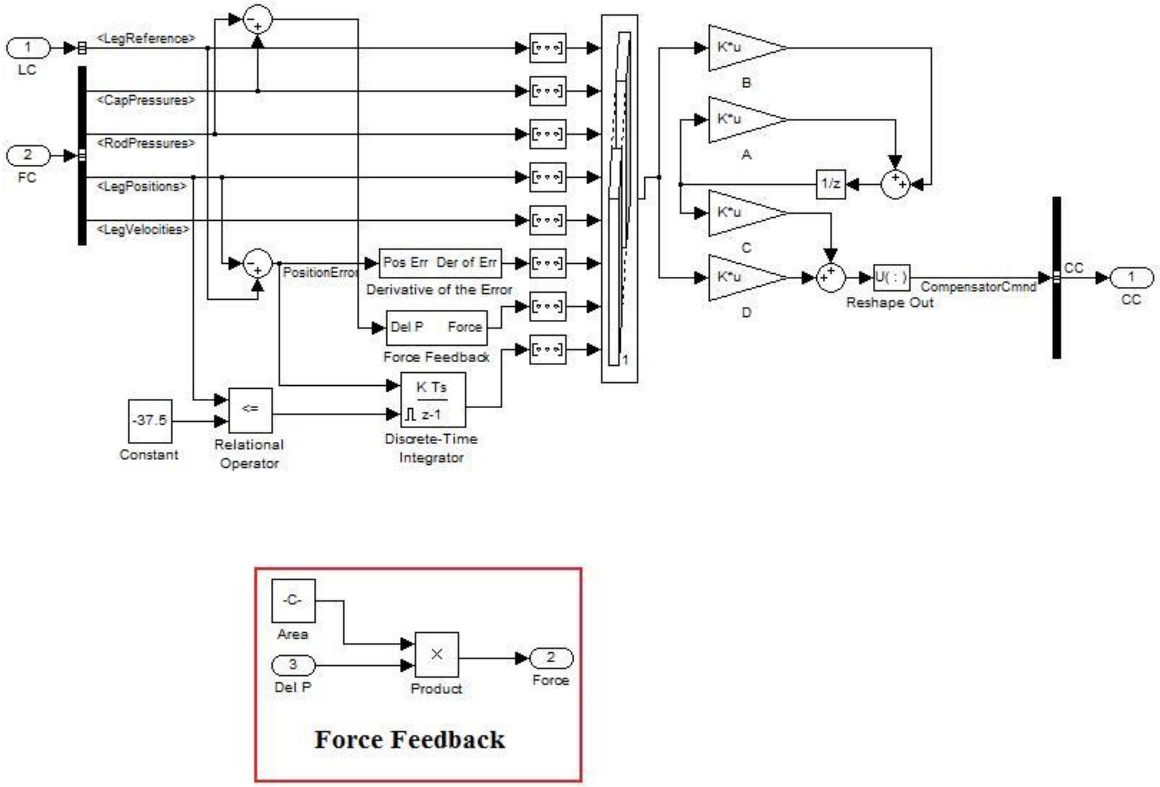

Figure 3-19 - Force Feedback Compensator Block Diagram

The area the pressure difference acted over was computed to be equal to 11.97 in^2. The error integration feedback was included in the force feedback model because there was a steady-state position error that could not be removed by modifying the gains. The force feedback compensator design gain sets are shown in Table 3-4.

Table 3-4 - Gain Set for the Force Feedback Compensator

Gain A	0.914						
Gain B	0	0	0	1	0	0	0
Gain C	-37.1759						
Gain D	48.8	-1.949e-7	9.816e-8	381.8	6	0.01	50

3.4.2 Results

The use of acceleration feedback in the compensator was evaluated in the non-linear model along with the force feedback setup. The following are the compensators compared in the final system improvement evaluation:

- Original Compensator, includes position feedback, commanded position, rod and cap pressure feedback (Original)
- Velocity Feedback Compensator, Original Compensator plus a gain on the velocity feedback (Vel. FB)
- Position Error Rate Feedback Compensator, Original Compensator plus position error derivative feedback (Pos. Err. Rate FB)
- Position Error Rate Feedback with Acceleration Feedback Compensator, the position error rate feedback compensator with acceleration feedback included (Acc. FB)
- Position Error Rate Feedback with Force Feedback Compensator, the position error rate feedback compensator with force feedback included (Force FB).

The first input used to compare the different compensator techniques was a one-inch heave step input. Figure 3-20 shows the response to the one-inch heave step input.

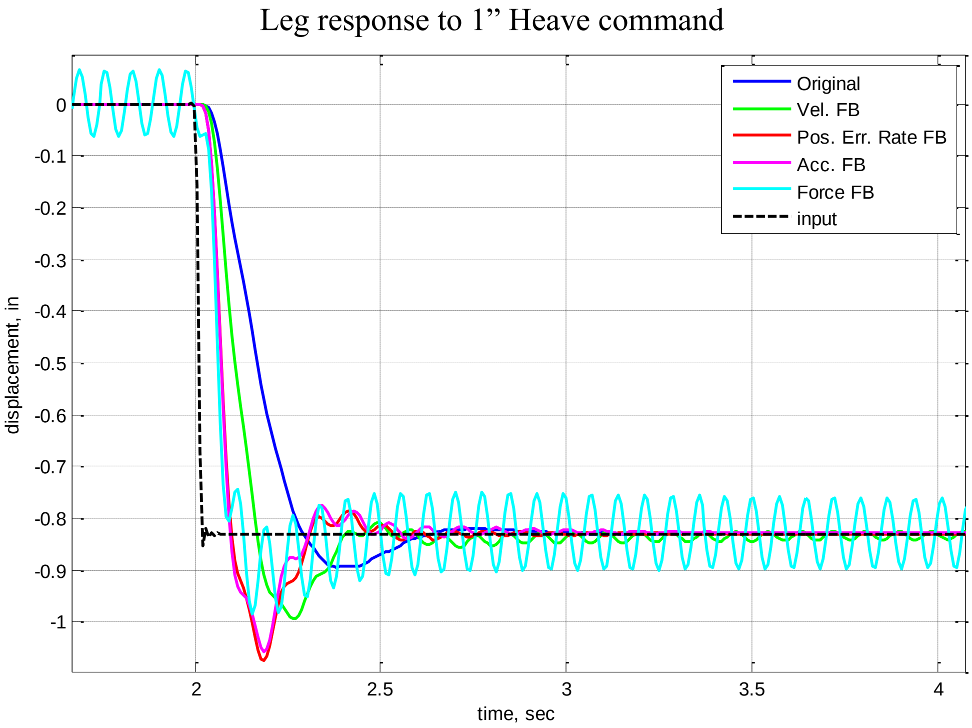

Figure 3-20 - Leg Response to a 1 Inch Heave Step Input, Compensator Comparison

The pure delay of approximately 0.03 seconds experienced by each of the responses can be attributed to the various rate transitions within the DCU and DCL. The delay is therefore inherent to the system and can be ignored throughout the analysis. From Figure 3-20 one benefit of the acceleration feedback is a quicker response, however, the overshoot is also increased. The force feedback has a slower response time, but also much better damping. The obvious major problem with the force feedback however is the oscillating response within the signal even when the response has reached steady state, which indicates the system is not divergent. It was found that for a heave input of one inch the force feedback will have accelerations upwards of three times the acceleration due to gravity. The oscillations within the response are caused by the amplification of the pressure feedback signal used in implementing the force feedback. Figure 3-21 shows the pressure feedback for the same input used in Figure 3-20.

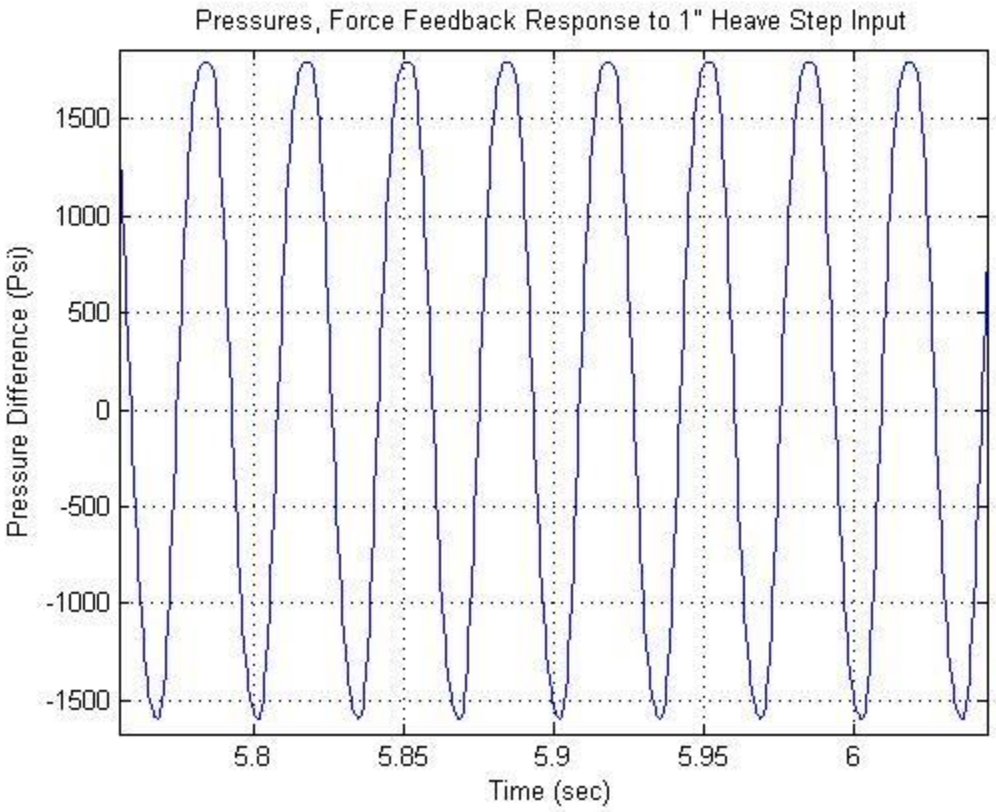

Figure 3-21 - Pressure Difference Feedback, Force Feedback Compensator

The high pressure which is being fed back into the force feedback controller where the pressure is gained to an even higher level can clearly be seen in Figure 3-21.

The frequency of the pressure feedback in Figure 3-21, approximately 25 Hz matches the frequency of the force feedback compensator response in Figure 3-20. The matching frequencies are why the pressure signal is recognized as being the problem.

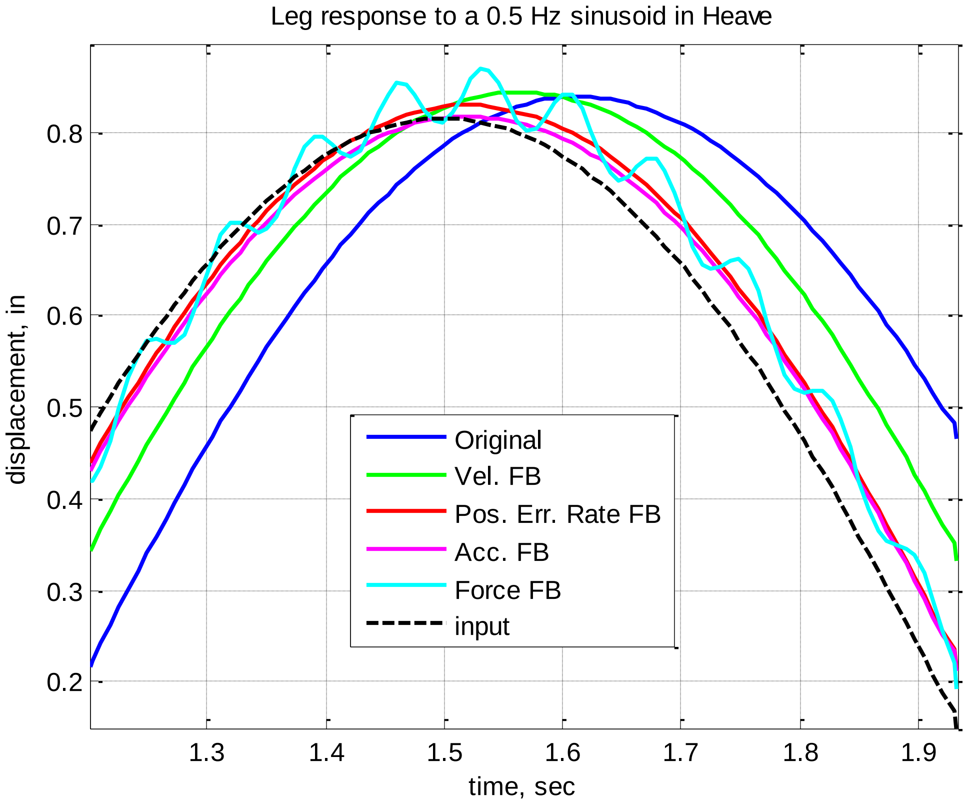

Figure 3-22 - Leg Responses to a 1 Inch 0.5 Hertz Sinusoidal Heave Input, Compensator Comparison

The responses in Figure 3-22 show how each of the compensators matches a simple sinusoidal hardware input signal. The force feedback still has the oscillations within the response. The acceleration feedback on the other hand looks to be a model with future benefits on the hardware. The acceleration feedback compensator has the closest match to the input signal in time response and does not have any significant damping issues. The acceleration feedback also has the least amount of delay. The next step is to examine each of the compensator designs' frequency response with respect to each other which is shown in Figure 3-23.

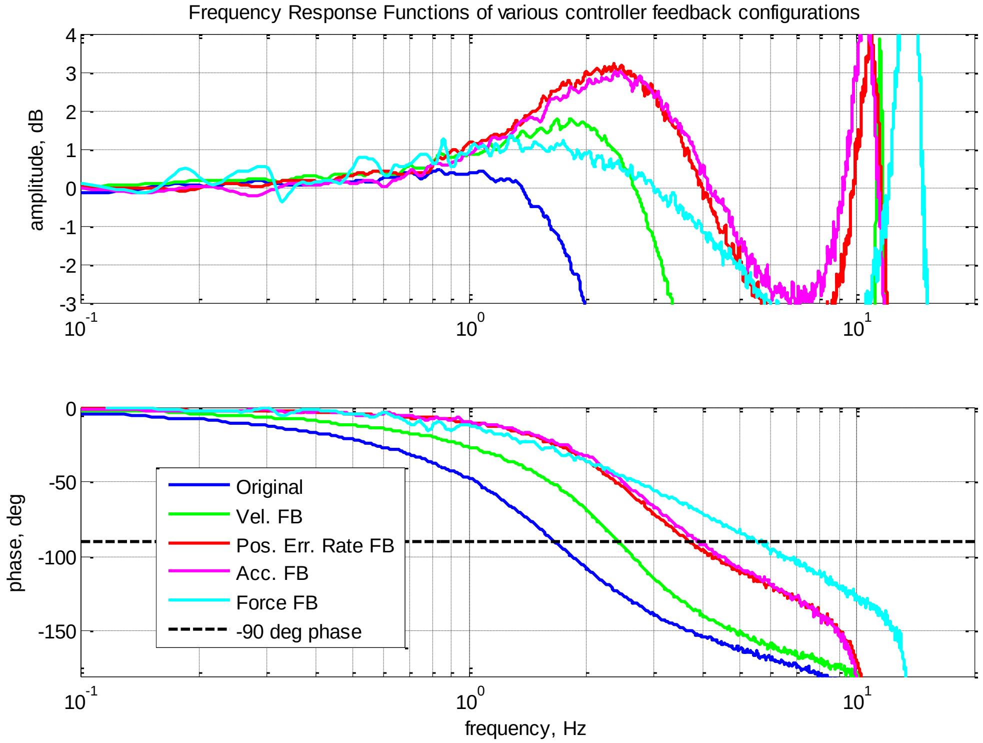

Figure 3-23 - Compensator Comparison in Frequency Response

The frequency responses in Figure 3-23 show the general progression of improvement from compensator to compensator. The most important areas to consider are the -3 dB and the -90^{o} phase bandwidth indicator lines. It can be seen that Position Error Rate feedback along with Acceleration and Force feedback configurations demonstrated highest amplitude bandwidth values falling into the range between 6 and 8 Hz. Force feedback, however, demonstrated the least amount of gain overshoot – only about 1 dB, which corresponds to the damping ratio of 0.5, which is considered acceptable. The force feedback also features the highest phase bandwidth at 5.5 Hz, of all the compensators. On the other hand, the force feedback shows comparatively better results in the frequency domain than with respect to the time responses.

3.4.3 Summary

The general results comparing each of the evaluated compensators can be seen in Table 3-5. Results for the remaining degrees of freedom for the position error rate feedback compensator and the acceleration feedback compensator are shown in APPENDIX D.

Table 3-5 - General Compensator Results, Evaluated from Non-Linear Model

Compensator	Damping	Phase Bandwidth	Amplitude Bandwidth
Original	0.55	1.5 Hz	2.0 Hz
Velocity FB	0.48	2.5 Hz	3.25 Hz
Position Error Rate FB	0.4	3.8 Hz	6.0 Hz
Acceleration FB	0.4	4.0 Hz	6.5 Hz
Force FB	0.5	5.5 Hz	6.0 Hz

From Table 3-5 as well as the results in Figure 3-20 and Figure 3-22 a few important conclusions can be drawn. The acceleration feedback improves performance to step inputs and tracks slow, low amplitude sinusoids better than any other compensator. The acceleration feedback also provides the best phase bandwidth at 4 Hz. The problem with the acceleration feedback is that the damping is poor; a damping ratio of 0.4 is acceptable but lower than desired.

The force feedback compensator on the other hand gives poor results for the step and sinusoidal inputs, having excessive fluttering within the signal. The force feedback however does provide the best damping. The fluttering of the force feedback compensator was found to be caused by the pressure difference feedback in the previous section. Keep in mind that the results produced for this section were performed using the non-linear model. This is important to remember because the modeling of the valves and the actuators may be a part of the fluttering pressures problem. It is evident then that as long as the high amplitude and high frequency pressure signal is used that the force feedback will not be stable enough for use without further filtering. Therefore it can be concluded that acceleration feedback should be tested further using the CMF hardware. Force feedback did not succeed on the non-linear model but may show improvements being operated on the hardware instead of the non-linear model.

Figure 3-3, Figure 3-11 and Figure 3-23 each show a possible divergence of the frequency response function at higher frequencies. While this can be viewed as a problem it is important to remember that the area of interest for this research was 3-5 Hz. A few possible causes of the increasing gain responses may be that a linear analysis is being performed of a non-linear system. Another possibility is that the motion platform has a resonance in the 7-10 Hz range, this hypothesis could be determined through testing on the hardware. The system is not often operated at high frequencies, mainly only for turbulence, which is why the problem was not addressed in this work. The issue was worth noting however because if the phase is greater than -180^o phase and the gain is greater than one then the system will become unstable.

4 CMF HARDWARE LINEAR MODEL IDENTIFICATION

4.1 Introduction

Operational mechanical hardware systems are very complex and contain a number of multifaceted relationships including linear relationships and time and space dependent relationships. When designing mechanical systems it is critical to properly model the actual hardware system. Accordingly, a number of models were developed during the initial system design. The non-linear model designed somewhat from first principles was discussed earlier in Section 2.4. It can also be useful to have a linear model of a hardware system, for although all system non-linearity is ignored and the system may be over simplified, a linear model can work very well as a model over a certain range of inputs. A model of the motion system is necessary for the planned model reference control law. A linear model is preferable over a non-linear model because the simple model takes less computing power to function which cuts down on adding delay to the system.

The first linear model was designed around a single design point, at the neutral point, the effective mass of the platform being 7200 lbm and a static load of 7200 lbf [8]. The linear model of the system was then able to be determined using classical SISO analysis techniques. Equations 4.1, 4.2 and 4.3 are the original transfer functions linearly modeling the CMF at the aforementioned design point [8].

$$\frac{Position}{CC}=\frac{9.129e10s+1.815e10}{s^6+491.9s^5+1.481e4s^4+2.488e6s^3+4.526e7s^2+8.755e6s+4.32e4} \tag{4.1}$$

$$\frac{Cap\,\mathrm{Pr}essure}{CC}=\frac{4.786e10s^3+2.983e10s^2+1.037e12s-2.943e11}{s^6+491.9s^5+1.481e4s^4+2.488e6s^3+4.526e7s^2+8.755e6s+4.32e4} \tag{4.2}$$

$$\frac{Rod\,\mathrm{Pr}essure}{CC}=\frac{-4.725e10s^3+2.759e10s^2+2.058e12s-5.844e11}{s^6+491.9s^5+1.481e4s^4+2.488e6s^3+4.526e7s^2+8.755e6s+4.32e4} \tag{4.3}$$

Note that CC is shorthand for the compensator command, the input command. Where the transfer functions in Equations 4.1, 4.2 and 4.3 are represented by input-output functions. The linear model design employed modeled each portion of the feedback. That is, the original linear model identified the leg extension, cap pressure and rod pressure feedbacks from the non-linear model as well as the compensator command input. Developing a model using a more sophisticated model is useful when working on control law improvements within a small boundary of the design point. However, the values used in this model were determined before the non-linear model was completely designed, leaving the gains and thus this first linear model of no use in the current improvement scheme.

A step to improve the original system linear model was to examine more closely the linear model gains. It was determined that the gains could be improved through the single leg linearization process if the non-linear model was used as the system to be identified. Another improvement to the original linear model is that the demand input to velocity output plant model would be calculated. With the addition of velocity feedback to the primary compensator, having a linear transfer function model including velocity feedback would be useful. The updated linearization was performed on the non-linear model and involved setting input/output points in the same areas as the original linearization. The new linear model, designated linear model A, utilized the control design, linear analysis tool from SIMULINK. This tool gave pole-zero maps for each of the four linearizations and from basic control theory the linear model transfer functions could be determined. The dominant poles and zeros were chosen for use in the linear transfer function. The linear model A transfer functions, shown in Equations 4.4, 4.5, 4.6 and 4.7, give a better representation of the CMF plant.

$$\frac{Position}{CC}=\frac{4e10s+3.6e9}{s^6+1438s^5+1.323e7s^4+6.258e9s^3+1.136e11s^2+6.795e9s} \quad (4.4)$$

$$\frac{Cap\,\mathrm{Pr}essure}{CC}=\frac{1e8s^2+2.1e7s+3.96e10}{s^5+1438s^4+1.323e7s^3+6.258e9s^2+1.136e11s^1+6.795e9} \quad (4.5)$$

$$\frac{Rod\,\mathrm{Pr}essure}{CC}=\frac{5000s^2-1.293e7s-5.596e10}{s^5+1438s^4+1.323e7s^3+6.258e9s^2+1.136e11s^1+6.795e9} \quad (4.6)$$

$$\frac{Velocity}{CC}=\frac{4e-10s+3.6e-11}{s^5+1438s^4+1.323e7s^3+6.258e9s^2+1.136e11s^1+6.795e9} \quad (4.7)$$

Linear model A represented the non-linear model well under certain conditions. As long as the velocity feedback control law is in use and simple single degree of freedom motions are used the linear model is a good match with the non-linear model as shown in Figure 4-1.

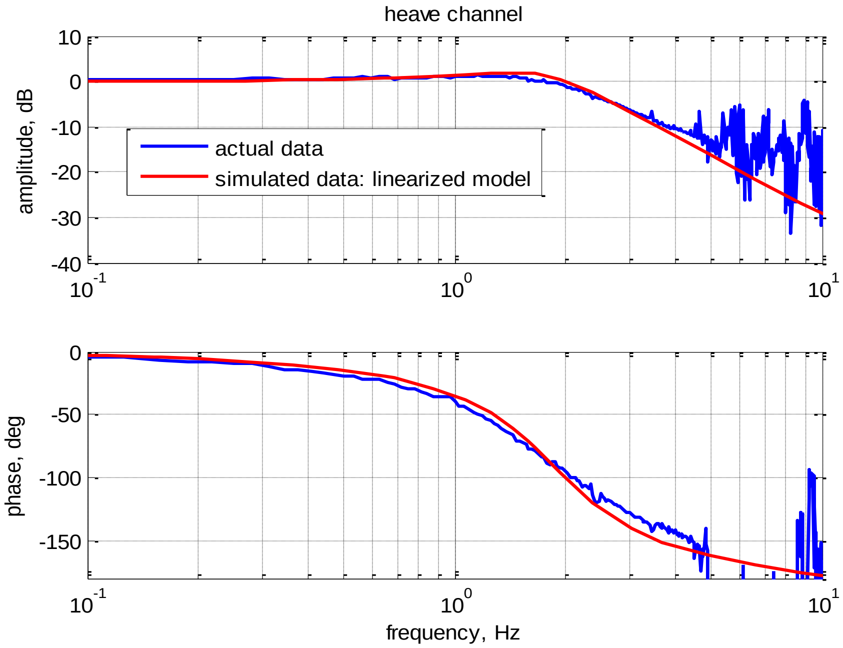

Figure 4-1 - Comparison of Linear Model A versus Actual Hardware Data

The problem with the linear model A was that the system representation was developed based upon the SIMULINK-based non-linear model which already has design flaws, such as not including moment of inertia effects. If control law modifications or model reference control approaches are to be used on the hardware, then the models need to be based upon the hardware to make them more robust.

4.2 Study Methodology

It is understood that the current linear model designs available were not adequate. Each of the previous linear model designs was developed without the use of hardware data. In order to develop the best linear model, the development should involve hardware information. Modeling each of the feedbacks independently will lead to problems in extended service when trying to implement a model reference control law approach. In order to obtain a linear model of the hardware system alone, it is necessary to carefully consider the design input and output points. To begin, a proper input signal which will be easily available from the hardware is desired. The signal which best fits this description is the compensator command. From the earlier description of the hardware and associated non-linear model, it is known that the control law output is the compensator command. The compensator command is also available as a DVS output which is important because it becomes very easy to record. After an input had been chosen the next step was to select an output. The output signals which were considered were the leg position, velocity and acceleration feedback values. The new plant model will not require having the rod and cap pressures available, the reason for this being that the gains in the compensator are so small that the contribution from the pressures are negligible. It was also shown earlier in the force feedback discussion that the pressure feedback is an unreliable signal. Therefore, the only required feedbacks which need to be available for this portion of the study are the acceleration, velocity and position feedbacks.

From an understanding of basic physics it is clear that there are a number of ways available to obtain the position and velocity feedback. The first method would be to obtain two separate transfer functions, one for each position and velocity. A second method would be to obtain the acceleration transfer function and perform integration once for velocity and again for position. A variation of this method would be to obtain the velocity transfer function and integrate the output signal to feedback position. The final method would be to obtain the position transfer function and take the derivative of the output to feedback the velocity signal.

Obtaining the position transfer function and taking the derivative of the output is an undesirable method because of the amount of noise which is known to enter a system through the use of numerical derivative techniques. The development of two separate position and velocity transfer functions from position and velocity feedbacks was not chosen either. The reason for this is due to the fact that one motivator for the new plant model was to have a single linear transfer function to describe the hardware. The chosen technique then is to perform the linearization on the velocity output signal with the compensator command as an input. The position feedback signal can then be obtained through a single integration, this being why the model is designed using velocity instead of acceleration.

After the linearization points in the model have been determined the next step is choosing the signals to use in the hardware to disturb the system and give meaning to the input-output relationship. When considering signals for use in model identification, the desired model breadth must be considered. For this system the most important area of concern, at this time, is the range of frequencies between two and five hertz. This range was chosen because the frequencies encompass the current phase bandwidth up to the desired bandwidth and are therefore the most relevant frequencies. A small range of frequencies was chosen in order to maximize the number of data points available for the linearization process. Furthermore, the signal amplitudes of concern are very small, approximately 1% of the total hardware displacement. A few different signal input techniques were tested and the results closest to the frequency response of the hardware were chosen to become the new linear model. The inputs to the plant were a pulse input, a sinusoidal input and a frequency sweep input.

4.3 Hardware Model Identification

Three different linear models were developed. Each of the linear models was based on hardware data as stated previously. The DVS recorded the compensator command, the system input, the velocity feedback and the system output command. The information from the recorded signals was used to predict a model using the autoregressive with an exogenous input method, ARX. The autoregressive model is known to have the ability to characterize the input-output behavior of a process which is the objective here. The autoregressive process is actually a special case of the Box-Jenkins model structure of the form:

$$y(k)=\frac{B\left(q^{-1}\right)}{A\left(q^{-1}\right)}u(k-d)+\frac{C\left(q^{-1}\right)}{D\left(q^{-1}\right)}e(k) \tag{4.8}$$

where { $y(k)$ } is a sequence of outputs, { $u(k)$ } is a sequence of process inputs, and { $e(k)$ } is a discrete white noise sequence [10]. The value d is the time delay between the process input and the output. For the ARX model structure, C=1 and D=A. The resulting system output is of the following form:

$$y(k+1)=B\left(q^{-1}\right)u(k-d+1)-A_1\left(q^{-1}\right)y(k)+e(k+1) \tag{4.9}$$

where $A=1+q^{-1}A_1$. The noise term $e(k+1)$ will act on the process in the future and can therefore be dropped; using the assumption the noise is included in the system in the future. The MATLAB system identification toolbox, which can utilize the ARX structure to determine a system model, was used in this part of the study and can utilize either time or frequency response inputs. The process used to determine the transfer function models began with a selection of the input and output signals. The desired model order was initially designed based on the original Carrelli linear model which was sixth order. Therefore the scalar setup becomes:

$$\hat{y}(k+1)=b_0u(k-d+1)+b_1u(k-d)+...+b_6u(k-d-5)-a_1y(k)-a_2y(k-1)-...-a_6y(k-5) \quad (4.10)$$

As the models progressed it became apparent that more zeros were necessary to obtain a quality model. The use of the MATLAB software allowed for much faster calculations of the transfer function models. This theoretical background was applied to the varying types of system input and resulting outputs to obtain the new linear models.

4.3.1 Linear Model #1 Identification and Results

The first linear model was developed using a one inch pulse signal. The recorded signals used for analysis include the compensator command, the position signal, velocity signal and acceleration signal. The position, velocity and acceleration feedback are measured data while the compensator command is a computed command signal. Figure 4-2 shows the pulse hardware data used for the first linear model design.

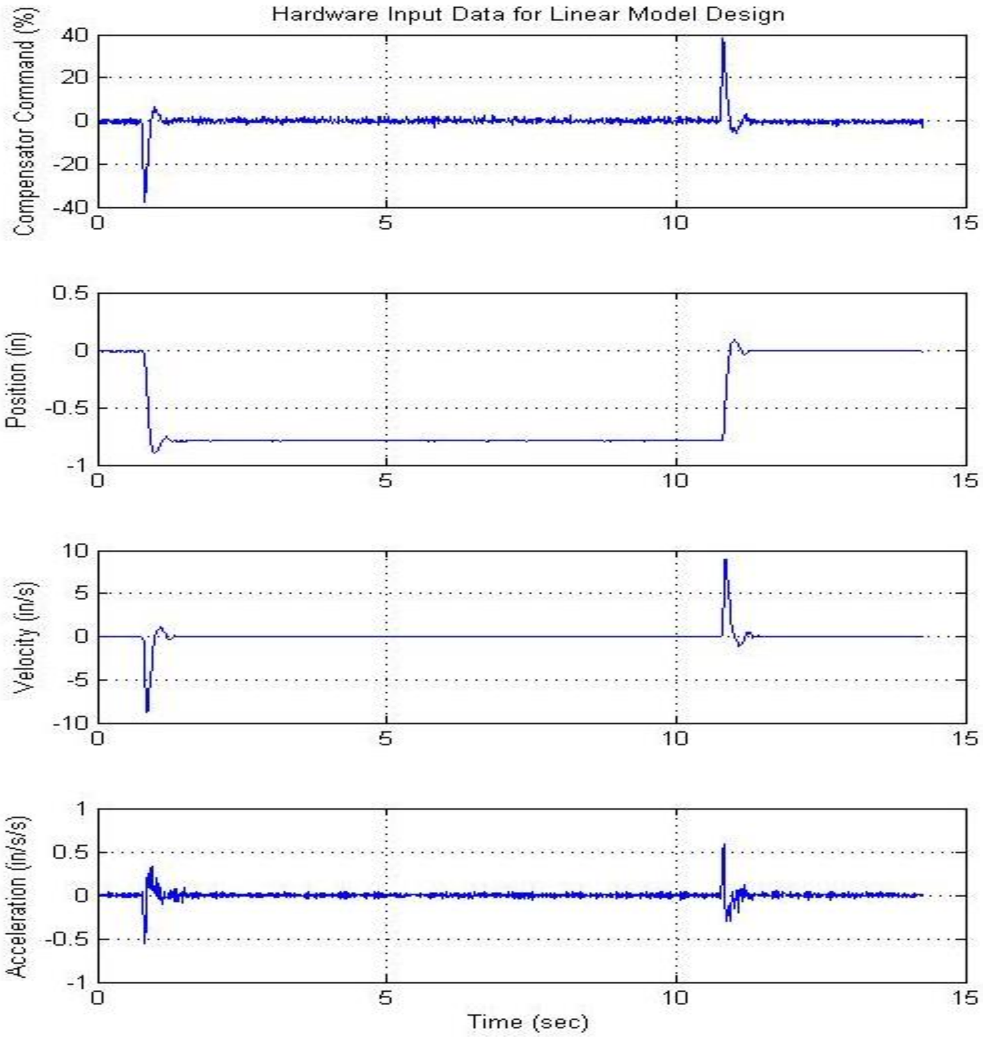

Figure 4-2 - Pulse Data for Linear Model Design 1

From the hardware data and use of the linearization technique discussed previously, a linear model was developed to describe the position, velocity and acceleration signals. While it was stated previously that only the velocity transfer function is necessary, the position and acceleration linear models were determined also in case they were needed for the future. Through the use of MATLAB internal ARX functions to expedite the design process, the following transfer functions were developed to represent the linear models.

$$\frac{Position}{CC}=\frac{0.002327s^6+2.595s^5+1064s^4+1.903e5s^3+1.312e7s^2+3.471e8s+4.912e10}{s^6+1091s^5+4.747e5s^4+8.331e7s^3+6.389e9s^2+1.573e11s-6.059e9} \quad (4.11)$$

$$\frac{Velocity}{CC}=\frac{0.002907s^{6}+1.357s^{5}+632.6s^{4}+1.597e5s^{3}+2.862e7s^{2}+3.596e9s+2.482e11}{s^{6}+199.5s^{5}+1.232e5s^{4}+1.19e7s^{3}+1.014e9s^{2}+5.103e10s+8.28e11} \tag{4.12}$$

$$\frac{Acceleration}{CC}=\frac{0.004442s^{7}+1.198s^{6}+1083s^{5}+3.264e5s^{4}1.685e8s^{3}+2.433e10s^{2}6.999e12s+3.016e13}{s^{7}565.4s^{6}+4.19e5s^{5}+1.159e8s^{4}+3.711e10s^{3}+4.551e12s^{2}+3.549e14s+1.361e16} \tag{4.13}$$

The position, velocity and acceleration linear models fit to the hardware can be seen in the response executed in open-loop fashion seen in Figure 4-3. The original data are shown in blue with the model data shown in red.

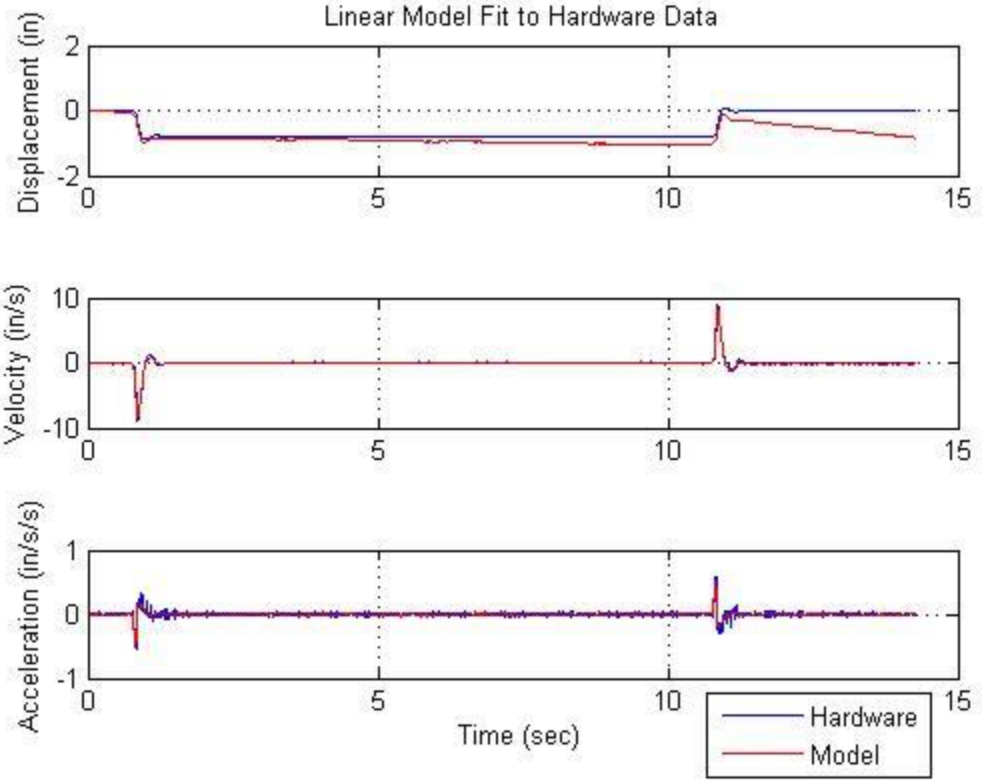

Figure 4-3 - Linear Model #1 Executed in Open Loop Fashion

The model velocity is almost equivalent to the hardware output, with a quality match during the dynamics and no errors during the steady-state portions. The position model has the largest difference between the linear model and the hardware results. The position model had difficulty maintaining a steady-state match as apparent in the model drift. The acceleration model is a strong fit considering the noise in the accelerometer hardware data, which can be clearly seen in Figure 4-4.

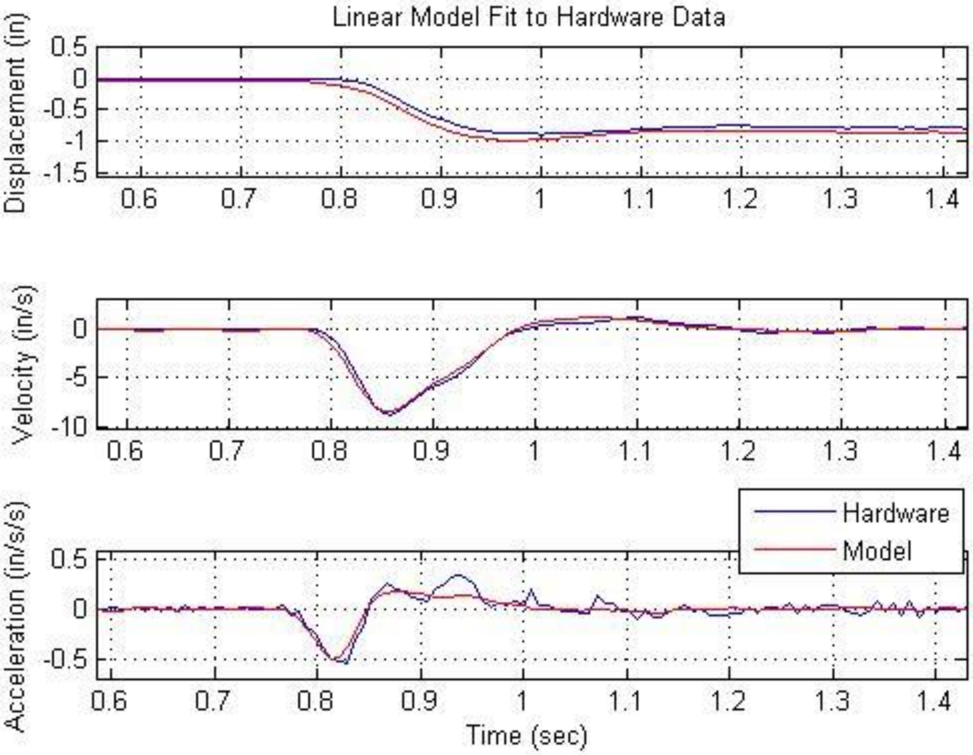

Figure 4-4 - Linear Model #1 Executed in Open Loop Fashion (Zoom)

With a good open loop model response, the next important step is to consider the closed loop response. The closed loop response will show how the model works in cooperation with the control law as well as how the model performs for different system inputs other than a step. The closed loop linear model, with block diagram in Figure 4-5, shows how the control law is implemented with the plant models.

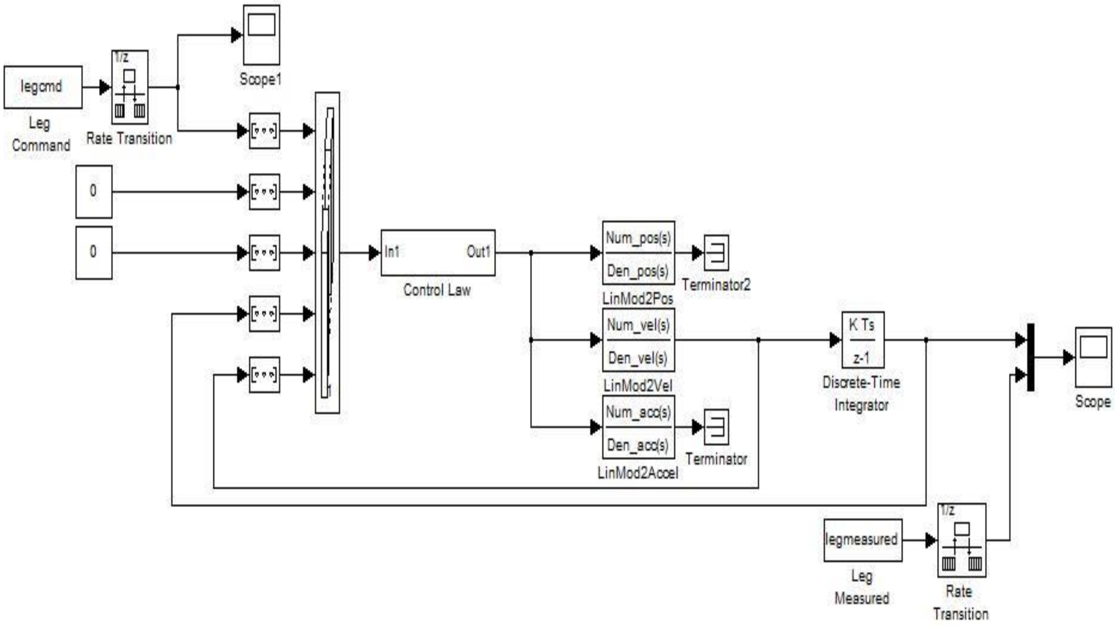

Figure 4-5 - Closed-Loop Linear Model Block Diagram

The pressure feedbacks, which were not modeled, have been set to zero. As explained previously the plant model which is used is the transfer function relating the output velocity to the command, with the position feedback calculated using a discrete-time integrator. Another important implementation in the closed loop model is that the commands are actual hardware input commands. This design was used because of the transformation which occurs in the non-linear model and the hardware between leg-space and platform space. The design is simplified significantly by using the actual hardware inputs which are already in leg-space. It is evident through the general behavior of the linear model compared to the hardware response in Figure 4-6 that the linear model is an appropriate model.

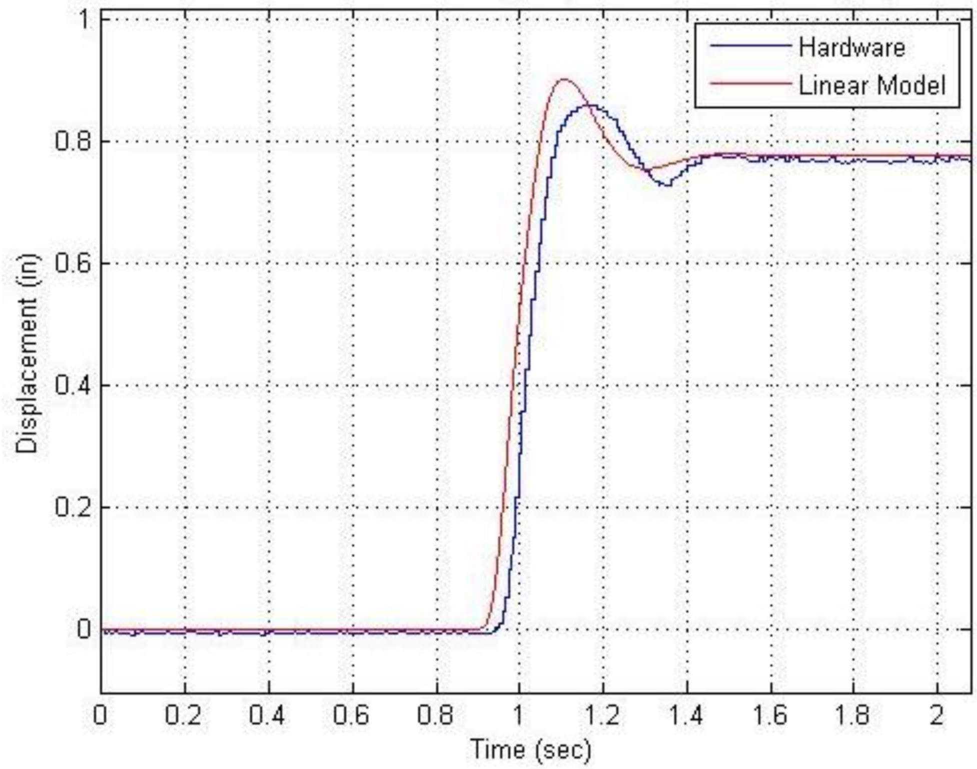

Figure 4-6 - Linear Model #1 and Hardware Leg Response to a -1 Inch Heave Step Input

As evidence in Figure 4-6, the overshoot difference is only around four percent, the steady-state values are essentially identical, and the rise times are very similar being only two-tenths of a second different.

4.3.2 Linear Model #2 Identification and Results

However, all linear models have limits and due to the design parameters of this first linear model, using a pulse input, matching high frequency sinusoids becomes more of a problem. With a high frequency input, for example a 5 Hz, 0.33 inch input shown in Figure 4-7, the linear model appears to begin to lose accuracy.

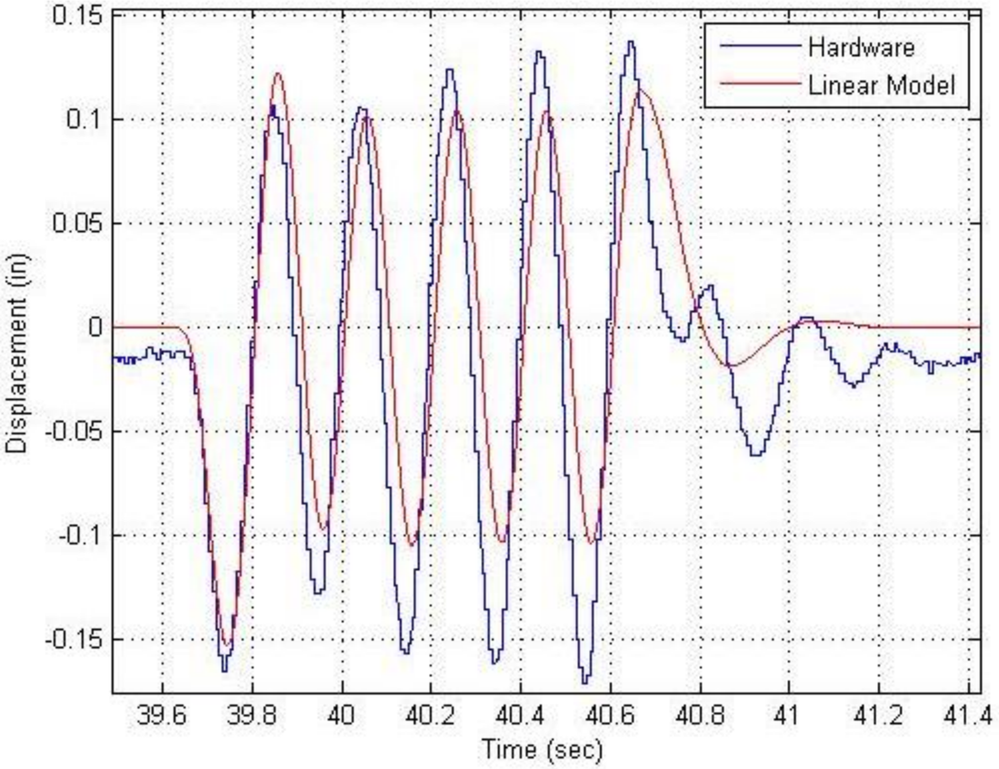

Figure 4-7 - Linear Model #1 and Hardware Leg Response to a 5 Hz, 0.33 Inch Heave Input

The failure of the linear model to follow the hardware response at a high frequency, including the degradation in displacement matching is shown in Figure 4-7.
A second model was designed based upon a simple sinusoid input. Through the use of the MATLAB code discussed earlier, the new linear models were based upon system inputs of one hertz and one inch heave sine.

$$\frac{Position}{CC}=\frac{0.002447s^6+3.595s^5+1805s^4+3.869e5s^3+3.285e7s^2+8.276e8s+6.144e10}{s^6+1465s^5+7.103e5s^4+1.522e8s^3+1.179e10s^2+2.004e11s-4.385e10} \quad (4.14)$$

$$\frac{Velocity}{CC}=\frac{0.004401s^6+3.003s^5+1277s^4+3.536e5s^3+5.963e7s^2+6.912e9s+4.759e11}{s^6+441.8s^5+1.611e5s^4+3.415e7s^3+2.303e9s^2+1.075e11s+1.621e12} \quad (4.15)$$

$$\frac{Acceleration}{CC}=\frac{0.008157s^6+1.957s^5+1458s^4+2.864e5s^3+5.245e7s^2+1.011e10s+5.375e9}{s^6+211.6s^5+1.893e5s^4+2.917e7s^3+7.353e9s^2+6.389e11s+1.509e13} \quad (4.16)$$

As before, the best matchup occurred with the velocity model. The acceleration signal had even greater noise with the sinusoidal input than the step input and the position linear model had a drift in it. Figure 4-8 presents hardware and linear model results for each of the displacement, velocity and acceleration models.

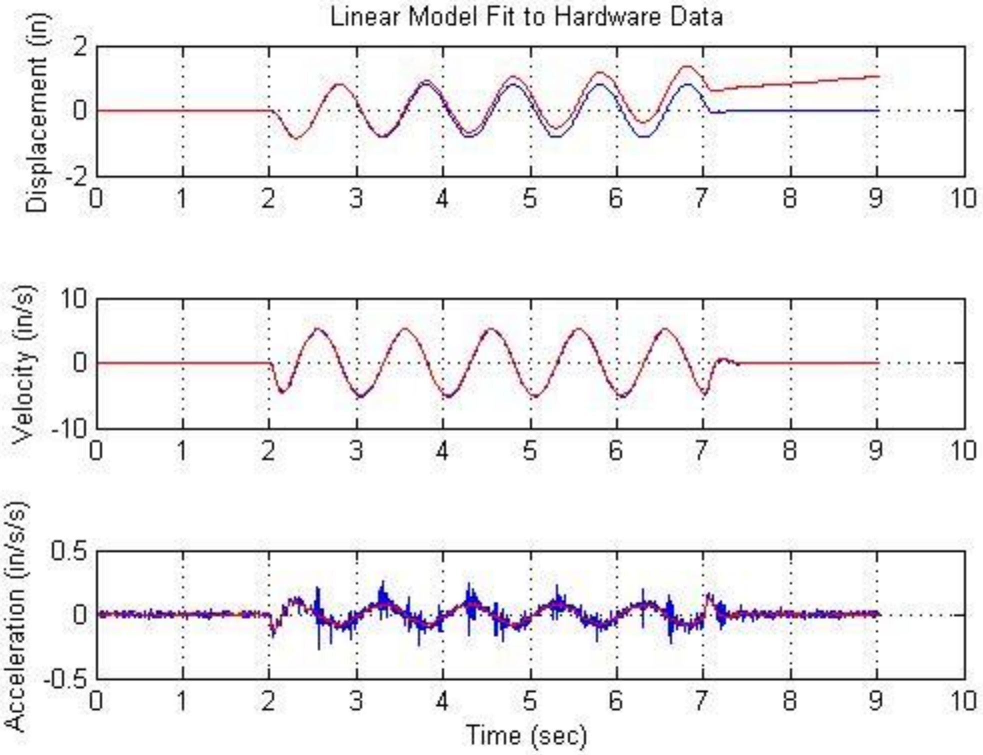

Figure 4-8 - Linear Model #2 Comparison to Hardware for Sinusoidal Input Data

With a strong velocity match, a step response was run with the velocity model to determine the effectiveness of the model in response to sharp changes in the commanded signal. When a one inch heave step input was used, the closed loop system performed exemplary. Figure 4-9 shows the hardware and linear model responses to the step input.

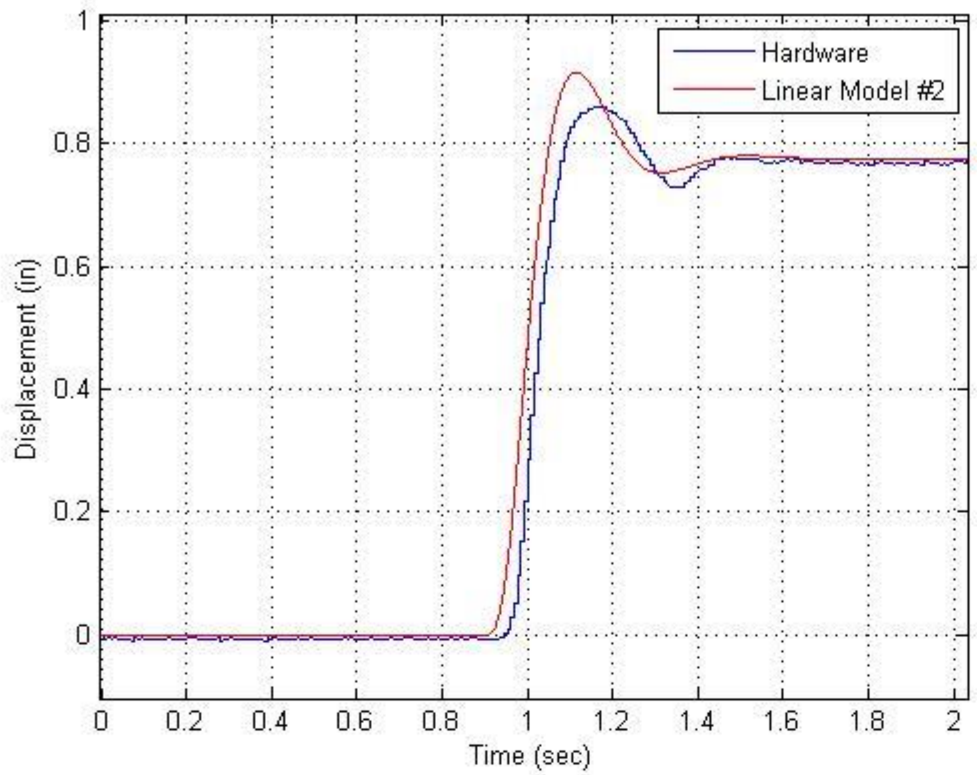

Figure 4-9 - Linear Model #2 and Hardware Leg Response to a -1 Inch Heave Step Input

When comparing with Figure 4-6, it can be seen that models will need to be compared in other ways because there are no distinguishable differences using a step command. The linear models both overshoot to 0.9 inches and settle 0.01 inches above the hardware model. The best way to compare the linear models at this point is to consider the responses to a chirp or a set of frequency sweeps. Using the results from a set of frequency sweeps will allow the frequency response functions of both linear models to be compared to the hardware frequency response function for the 3.0 Hz compensator design with the GFD platform. This exercise was again performed using the closed loop system in Figure 4-5. However, instead of a step or sinusoidal input, a chirp signal was used. The chirp signal was designed to cover the desired frequencies, in this case from 0.01 Hz up to 10 Hz. Since the range of frequencies is large, a time span of 100 seconds was chosen to enable an adequate amount of data points to be collected for each frequency. The chirp signal was then run for both linear model #1 and linear model #2. Figure 4-10 shows the comparison between the two linear models.

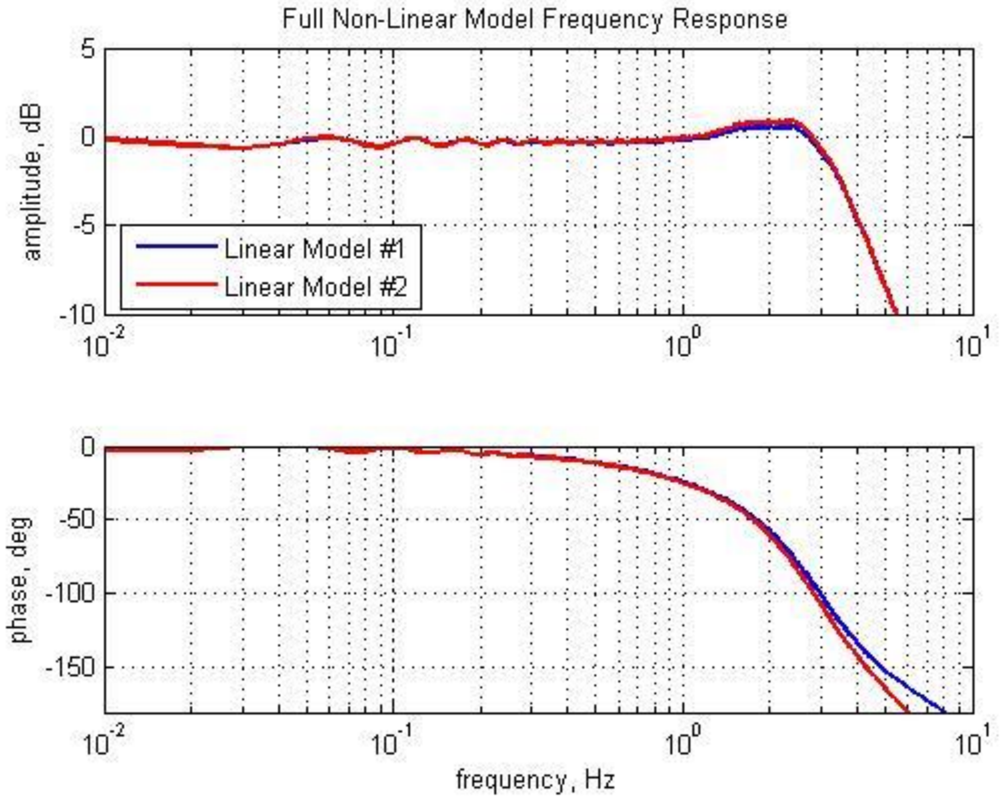

Figure 4-10 - Frequency Response Function Comparison between Linear Models #1 and #2

The frequency response function of Figure 4-10 should not come as a surprise; the two models are of the same order and are derived in similar ways. Both models were developed using MATLAB's internal ARX function. With both linear model #1 and linear model #2 showing approximately 2.7 Hz of phase bandwidth it is apparent that the models are strong fits for this compensator.

4.3.3 Linear Model #3 Identification and Results

After observing that the error in phase bandwidth compared to the velocity feedback compensator for the best linear model, linear model #1 was 6.5%, it was desired to determine if using frequency sweep data could improve the model. The final linear model was designed in a similar manner with the use of MATLAB's internal ARX command; however the input signal in this case was the frequency sweep. The frequency sweep data followed that of Table 2-3, with amplitudes of 0.3 inches. The resulting transfer function models for position, velocity and acceleration are shown below:

$$\frac{Position}{CC}=\frac{0.002093s^{6}+3.101s^{5}+1533s^{4}+3.165e5s^{3}+2.576e7s^{2}+1.042e9s+1.256e11}{s^{6}+1391s^{5}+7.049e5s^{4}+1.556e8s^{3}+1.775e10s^{2}+4.011e11s-2.932e11} \quad (4.17)$$

$$\frac{Velocity}{CC}=\frac{0.007454s^{7}+4.01s^{6}+2642s^{5}+7.91e5s^{4}+2.038e8s^{3}+3.111e10s^{2}+3.228e12s+1.862e14}{s^{7}+354.9s^{6}+2.753e5s^{5}+5.008e7s^{4}+1.381e10s^{3}+7.422e11s^{2}+4.226e13s+6.516e14} \quad (4.18)$$

$$\frac{Acceleration}{CC}=\frac{0.004772s^{6}+2.388s^{5}+962.8s^{4}+2.178e5s^{3}+4.368e7s^{2}+2.794e9s-8.217e10}{s^{6}+320s^{5}+1.925e5s^{4}+3.824e7s^{3}+6.746e9s^{2}+4.323e11s+1.36e11} \quad (4.19)$$

The frequency sweep linear model was built using the MATLAB ARX tool. The MATLAB ARX modeling tool forced the velocity model to increase in size to allow for a real negative root. This is why the transfer function order increased to the 7th order. The linear models in position and acceleration were very poor for the frequency sweep matching, as seen in Figure 4-11 below.

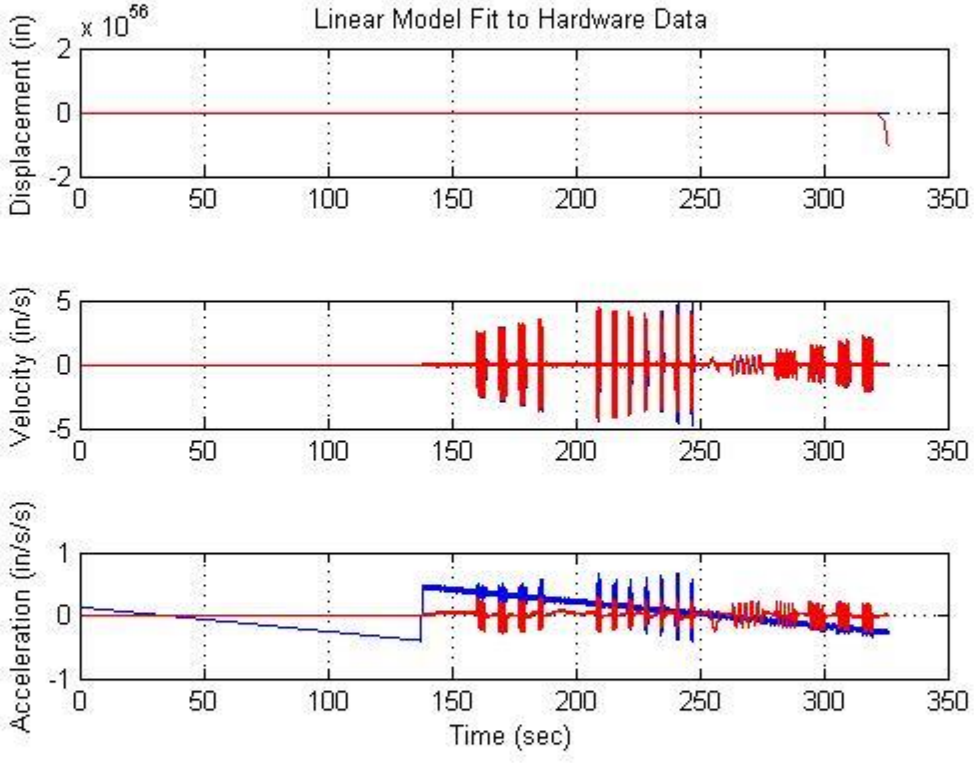

Figure 4-11 - Linear Model #3 Comparison to Hardware for Frequency Sweep Input

It is known that since the frequency sweeps needed to be stitched together that periods of no input occurred between sweeps. The stitching caused a lack of information throughout portions of the modeling data that deteriorated the model. In the future it is recommended to develop a technique to build models utilizing white noise with a constant power spectrum density or at a minimum a chirp type signal input. The white noise and chirp signal are in development at this time. In order to determine the quality of the velocity model, henceforth linear model #3, the model was run in the closed loop system for a step and the models frequency response function was determined as well. In Figure 4-12, the step response for linear model #3 can be seen.

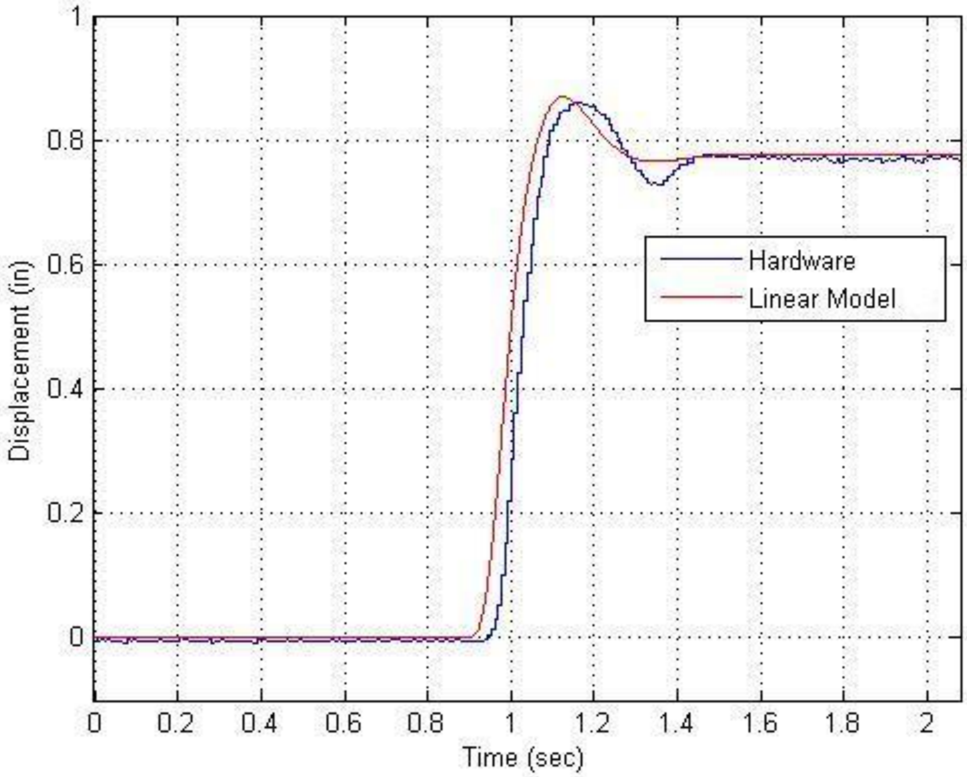

Figure 4-12 - Linear Model #3 and Hardware Leg Response to a -1" Heave Step Input

Linear model #3 has the best step response comparison yet, based upon an initial overshoot comparison. The peaks for Figure 4-12 are much closer in amplitude than for linear model #1 and #2 respectively.

4.4 Summary

The results for damping ratio, and phase and gain bandwidth from the three linear models as well as the velocity feedback compensator are shown in Table 4-1. Figure 2-15 and Figure 2-16 are the velocity feedback compensator results from the hardware. The results from the velocity feedback compensator have been included in Table 4-1 for comparison purposes.

Table 4-1 - Linear Model Performance Characteristics

	-90^o Phase Bandwidth	-3dB Bandwidth	Damping Ratio
Linear Model #1	2.75	3.59	0.5
Linear Model #2	2.62	3.62	0.5
Linear Model #3	2.72	3.32	0.7
Velocity FBK	2.94	4.0	0.4

The bode response in Figure 4-13, is Figure 4-10 with the addition of linear model #3. Figure 2-15 shows the frequency response for the velocity feedback compensator.

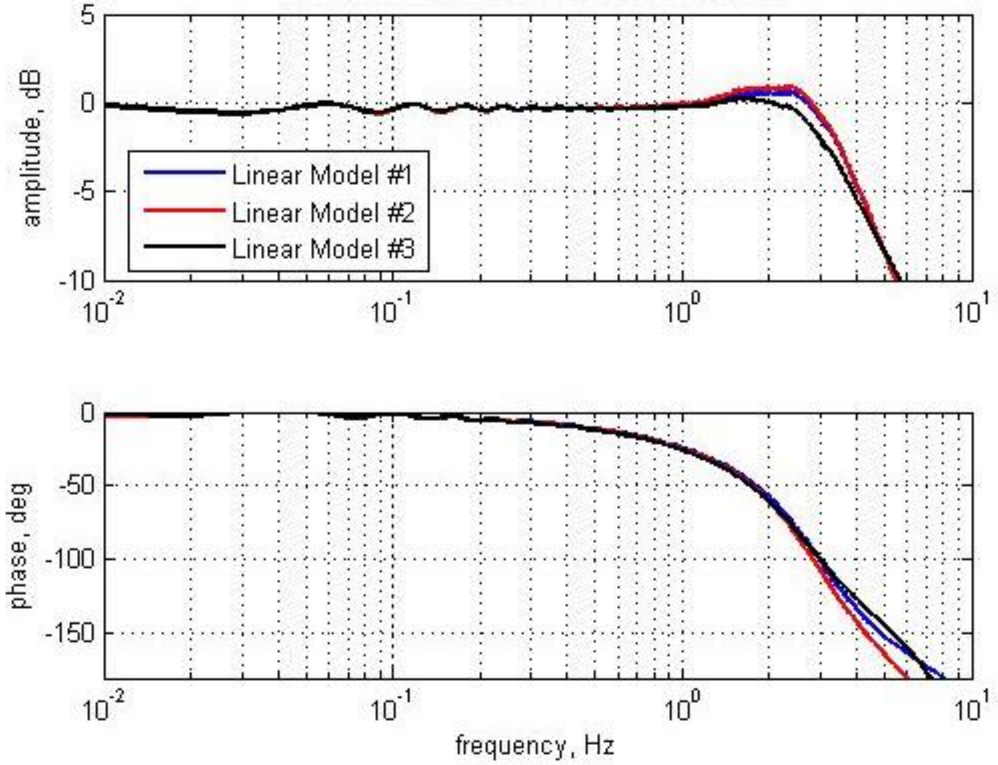

Figure 4-13 - Frequency Response Comparison between Linear Model #1, Linear Model #2 and Linear Model #3

From Figure 4-13 it can be concluded that the three linear models each give a slightly different response. Each of the linear models would be expected to be identical if the plant was linear. The discrepancies in the linear models are a result of the plant having non-linearity. The error for the -90^{o} phase bandwidth, the most important performance characteristic, is 6.5% for linear model #1 and 10.9% for linear model #2 and 7.5% for linear model #3 when comparing to the velocity feedback compensator results. The data driven linear models all have a better match than the theoretically driven non-linear model which has an error of 15.0% in phase bandwidth. The linear models outperform the non-linear model for small input commands, in a simple setup environment.

Linear model #3 appears to be a better match to the hardware for a step response however; linear model #1 is the best model in terms of frequency response when using Table 4-1. As mentioned previously it is important to note the linear model's limitations, most importantly that these results are only valid for small input commands and in the heave degree-of-freedom, a major reason the non-linear model is still an overall better model than any of the linear models.

It should be noted that this study did not examine large movements or movements in directions other than heave. Understanding the limitations of a model is important, especially that the linear model designed here is restricted by input signals and compatibility with the CMF-DCU-DCL system. The use of the relatively simple ARX modeling can lead to holes in a model especially biases [10]. However, the results indicate that there is potential to develop a linear model which may enable new control architectures such as a model reference approach to be developed.

5 CONCLUSION

5.1 Summary

Improvements in the overall compensator performance were first developed. In addition, the bandwidth of the system was increased maintaining or increasing the system damping. Finally, this work developed a more reliable linear model, the first step to providing a model reference control law.

The problem of a position offset was solved through the use of integral control on the error signal. It was proven that with a proper gain, in this case equal to 50, the position error could be removed. In addition to removing the position error, as expected the integral control was shown to not degrade the damping ratio or bandwidth. Therefore using integral control in the compensator is highly recommended in the future if position error becomes a problem.

Regarding improving the overall compensator performance, the first step in the process was to investigate rate feedback techniques. It was through the examination of different rate feedback techniques that the position error rate feedback compensator was developed. The position error rate feedback compensator provided the best results for a rate feedback technique. The position error rate feedback compensator provided an increase in phase bandwidth while keeping the damping ratio unchanged. The final position error rate feedback compensator phase bandwidth was 3.8 Hz and the damping ratio was approximately 0.3. After increasing the bandwidth through an improved rate feedback technique, the next step was to establish if the bandwidth or damping could be significantly improved through the addition of acceleration or force feedback.

The addition of acceleration feedback illustrated an improvement in motion system performance when analyzed using the non-linear SIMULINK model. The acceleration feedback provided a phase bandwidth of approximately 5 Hz and a damping ratio of 0.4. It was impressive to see another one hertz gain in phase bandwidth with the addition of acceleration feedback to the position error rate feedback compensator. The improvement in damping provided by the acceleration feedback compared to the position error rate feedback compensator is significant because it shows that the addition of acceleration feedback improves both of the most important performance characteristics.

While the improvements seen in the non-linear model are promising it is still just a model and should be tested on the CMF hardware.

Finally, an improved linear model of the CMF dynamics was developed through the use of system identification techniques. Hardware inputs and outputs were recorded and then through the use of MATLAB ARX an improved linear model was developed. Linear model #1, which was based upon a one inch pulse input, provided a very good match to the hardware response. The error in the frequency response domain was approximately 6.5% between the hardware bandwidth at -90^{o} Phase and linear model #1. Linear model #1, in Equation 4.17, is a model which should be used in a future model reference control law applications.

5.2 Future Work

A significant degree of the aerospace industry's development of control law technology is based upon models and theory. It is only natural then that engineering theory and hardware models do not always work as expected when applied to a realistic 'real-life' system. For these reasons, it is important to test the new compensator designs developed in this research on the hardware and determine if in fact they will provide the expected hardware results.

In addition, the linear model development process could be improved. Providing better linear models hinges on using better input signals and more sophisticated system identification techniques. The signal used in system identification should provide coverage of the most important areas of identification. The linear models designed through this research are only accurate for small signal amplitudes in the heave degree of freedom. For use as a model reference approach a signal which could provide more detail in the 0.01 Hz to 10 Hz range could provide large improvements to the linear model. Including movements in degrees of freedom other than heave would be important for future linear models as well. The second step to improving the linear models is using more sophisticated system identification techniques, methods such as fuzzy logic or another soft computing technique could provide an improved model.

APPENDIX A

MATLAB Codes

Hardware Data Frequency Response Stitching and Calculation Tool:

Written By: Kirill Zaychik

```
tic
%%   %%%%%%%%%%%%%%%%  BLOCK 1 %%%%%%%%%%%%%%%%%%%%%%
     %----------- load CMF data files -----------%

% Use Array length + 1
% heave
% Data1=dlmread('XFreqSweep_Heave_0.33in_1of5_0011_2012-04-
11.csv',',','BO2..FW22705');
% Data2=dlmread('XFreqSweep_Heave_0.33in_2of5_0012_2012-04-
11.csv',',','BO2..FW9664');
% Data3=dlmread('XFreqSweep_Heave_0.33in_3of5_0013_2012-04-
11.csv',',','BO2..FW8450');
% Data4=dlmread('XFreqSweep_Heave_0.33in_4of5_0014_2012-04-
11.csv',',','BO2..FW8028');
%
Data5=dlmread('XFreqSweep_Heave_0p33in_5of5_0020_2011_12_02GFD.csv',','
,'BO2..FW8011');

% % surge
% Data1=dlmread('XFreqSweep_Surge_0.33in_1of5_0006_2012-04-
11.csv',',','BO2..FW21951');
% Data2=dlmread('XFreqSweep_Surge_0.33in_2of5_0007_2012-04-
11.csv',',','BO2..FW9654');
% Data3=dlmread('XFreqSweep_Surge_0.33in_3of5_0008_2012-04-
11.csv',',','BO2..FW8941');
% Data4=dlmread('XFreqSweep_Surge_0.33in_4of5_0009_2012-04-
11.csv',',','BO2..FW8675');
% Data5=dlmread('XFreqSweep_Surge_0.33in_5of5(partial)_0010_2012-04-
11.csv',',','BO2..FW8118');

% sway
% Data1=dlmread('XFreqSweep_Sway_0.33in_1of5_0018_2012-04-
11.csv',',','BO2..FW21810');
% Data2=dlmread('XFreqSweep_Sway_0.33in_2of5_0019_2012-04-
11.csv',',','BO2..FW10120');
% Data3=dlmread('XFreqSweep_Sway_0.33in_3of5_0020_2012-04-
11.csv',',','BO2..FW8438');
%Data4=dlmread('XtndFreqSweep_Sway_0.33in_4of5_0009_2010-07-
19.csv',',','BO2..FW8062');
%Data5=dlmread('XtndFreqSweep_Sway_0.33in_5of5_0010_2010-07-
19.csv',',','BO2..FW7764');

% roll
% Data1=dlmread('XFreqSweep_Roll_0.2deg_1of5_0021_2012-04-
11.csv',',','BO2..FW22629');
% Data2=dlmread('XFreqSweep_Roll_0.2deg_2of5_0022_2012-04-
11.csv',',','BO2..FW9503');
```

```
% Data3=dlmread('XFreqSweep_Roll_0.2deg_3of5_0023_2012-04-
11.csv',',','BO2..FW8470');
%Data4=dlmread('XtndFreqSweep_Roll_0.2deg_4of5_0019_2010-07-
19.csv',',','BO2..FW7871');
%Data5=dlmread('XtndFreqSweep_Roll_0.2deg_5of5_0020_2010-07-
19.csv',',','BO2..FW9280');

% pitch
% Data1=dlmread('XFreqSweep_Pitch_0.2deg_1of5_0024_2012-04-
11.csv',',','BO2..FW22485');
% Data2=dlmread('XFreqSweep_Pitch_0.2deg_2of5_0025_2012-04-
11.csv',',','BO2..FW9857');
% Data3=dlmread('XFreqSweep_Pitch_0.2deg_3of5_0026_2012-04-
11.csv',',','BO2..FW8669');
%Data4=dlmread('XtndFreqSweep_Pitch_0.2deg_4of5_0024_2010-07-
19.csv',',','BO2..FW7777');
%Data5=dlmread('XtndFreqSweep_Pitch_0.2deg_5of5_0025_2010-07-
19.csv',',','BO2..FW7918');

% yaw
% Data1=dlmread('XFreqSweep_Yaw_0.2deg_1of5_0027_2012-04-
11.csv',',','BO2..FW22135');
% Data2=dlmread('XFreqSweep_Yaw_0.2deg_2of5_0028_2012-04-
11.csv',',','BO2..FW10339');
% Data3=dlmread('XFreqSweep_Yaw_0.2deg_3of5_0029_2012-04-
11.csv',',','BO2..FW8900');
%Data4=dlmread('XtndFreqSweep_Yaw_0.2deg_4of5_0029_2010-07-
19.csv',',','BO2..FW7797');
%Data5=dlmread('XtndFreqSweep_Yaw_0.2deg_5of5_0030_2010-07-
19.csv',',','BO2..FW7713');

% WhiteNoise
Data=dlmread('WhiteNoise2_Heave_0031_2012-04-
13.csv',',','BO2..FW36739');

% Data=[Data1;Data2;Data3;Data4];      % stitch the low to high
frequency data segments together
fHz=128;                       % Sampling rate (Hz) at which data was
collected
Tsim=length(Data(:,1))/fHz;    % Total duration of the simulation
Time=0:1/fHz:Tsim-1/fHz;       % Time vector
Trec=Tsim;
Lwin=4096;                     % segment size used in FRF estimation
novlp=1024;                    % number of overlaping points
nfft=4096;                     % length of the FFT Xform
w=hanning(Lwin+1);             % type of the window applied
OptionsFRF = struct('NS',7,'overL',50,'wintype',1,'flagPorR',0);

dofid=3;  % DOF analysed: 1-surge, 2-sway, 3-heave, 4-yaw, 5-pitch, 6-
roll


%%   %%%%%%%%%%%%%%%%  BLOCK 2 %%%%%%%%%%%%%%%%%%%%%%
     %----------- Generate the demand and feedback leg extension
signals -----------%
```

```
dmd_actuator=detrend(Data(:,63:68),'constant');    %
DCL2DVS_DMD_LEG_POS
fdk_actuator=detrend(Data(:,1:6),'constant');      %
DCL2DVS_FDK_LEG_POS


%%   %%%%%%%%%%%%%%%%  BLOCK 3 %%%%%%%%%%%%%%%%%%%%%%
     %----------- Inverse transformation DOF to leg space -----------%

dmd_dof=invXform(dmd_actuator);   % the inverse transformation function
is called
fdk_dof=invXform(fdk_actuator);   % the inverse transformation function
is called

dmd=dmd_dof(dofid,:);   % only one (dofid) degree-of-freedom is
extracted
fdk=fdk_dof(dofid,:);   % only one (dofid) degree-of-freedom is
extracted

%%   %%%%%%%%%%%%%%%%  BLOCK 4 %%%%%%%%%%%%%%%%%%%%%%
     %----------- Compute FRF -----------%

% condition input/ouput signal and call act_frf_new to compute FR
function
input=detrend(dmd');     % remove mean from the signal
output=detrend(fdk');    % remove mean from the signal

%call FRF function act_frf_new
[F_Hz_d, F_rad_d, R_abs_d, R_db_d, Ph_d,
H,Coh]=FRFunction(input,output,Time,OptionsFRF);

F=F_Hz_d;   % frequency array
R=R_db_d;   % amplitude (dB)
Ph=Ph_d;  % phase (deg)

%%   %%%%%%%%%%%%%%%%  BLOCK 5 %%%%%%%%%%%%%%%%%%%%%%
     %----------- plotting and saving results -----------%
% plot time histories of the input/ouput signals
figure(1)
plot(Time,dmd,Time,fdk,'r'); grid on;
title('time histories: demand vs feedback dof trajectories');
xlabel('time, sec');
ylabel('displacement, inches');     % use for translational DOF
% ylabel('displacement, degrees');  % use for rotational DOF

legend('Dmd Heave','Fbk Heave');

% plot computed FRF
figure(2)
subplot(2,1,1); semilogx(F,R,'k'); grid on;
axis([0.1 10 -30 10]);
hold on
ylabel('Amplitude, dB');
```

```
title('Frequency Response, Position Feedback CL')
legend('Heave');
subplot(2,1,2); semilogx(F,Ph-180,'k'); grid on;
axis([0.1 10 -180 0]);
hold on
xlabel('Frequency, Hz'); ylabel('Phase, deg')
legend('Heave');

%save FRF_act_heave W   % save computed FRF
% save FRF_act_surge W   % save computed FRF
%save FRF_act_sway W   % save computed FRF
% save FRF_act_roll W   % save computed FRF
%save FRF_act_pitch W   % save computed FRF
%save FRF_act_yaw W   % save computed FRF
```

Inverse Transformation Function:

Written by David Carrelli

```
function y=invXform(r)

r=r+113.5;

% load platform geometry
            A = [91.30,-37.87,-53.43,-53.43,-37.87, 91.30;
                  8.98, 83.56, 74.58,-74.58,-83.56, -8.98;
                  0.05,  0.05,  0.05,  0.05,  0.05,  0.05]';
% Lower U-joint geometry matrix:
            B = [66.17,48.29,-114.46,-114.46,48.29,66.17;
                0.96,104.29,10.33,-10.33,-104.29,-93.96;
               133.08,133.08,133.08,133.08,133.08,133.08]';

term=length(r(:,1));

for j=1:term      % initiate pass through all data

    clear df f SV

    x=0; y=0; z=-113.5; psi=0; theta=0; phi=0;
    SV_old=[x y z phi theta psi]';

    for k=1:5    % initiate Newton-Raphson method

        T=dircos(phi, theta, psi);    % evaluate directional cosin
matrix

        % evaluate vector F and Jacobian Matrix
        for i=1:6

f(i,1)=A(i,1)^2+A(i,2)^2+A(i,3)^2+B(i,1)^2+B(i,2)^2+B(i,3)^2+x^2+y^2+z^
2-r(j,i)^2+...
                2*(x-
B(i,1))*(A(i,1)*T(1,1)+A(i,2)*T(2,1)+A(i,3)*T(3,1))+...
```

```
                2*(y-
B(i,2))*(A(i,1)*T(1,2)+A(i,2)*T(2,2)+A(i,3)*T(3,2))+...
                2*(z-
B(i,3))*(A(i,1)*T(1,3)+A(i,2)*T(2,3)+A(i,3)*T(3,3))+...
                -2*(x*B(i,1)+y*B(i,2)+z*B(i,3));

            df(i,1)=2*(x+A(i,1)*T(1,1)+A(i,2)*T(2,1)+A(i,3)*T(3,1)-
B(i,1));
            df(i,2)=2*(y+A(i,1)*T(1,2)+A(i,2)*T(2,2)+A(i,3)*T(3,2)-
B(i,2));
            df(i,3)=2*(z+A(i,1)*T(1,3)+A(i,2)*T(2,3)+A(i,3)*T(3,3)-
B(i,3));
            df(i,4)=-2*(x-
B(i,1))*(A(i,1)*T(1,2)+A(i,2)*T(2,2)+A(i,3)*T(3,2))+2*(y-
B(i,2))*(A(i,1)*T(1,1)+A(i,2)*T(2,1)+A(i,3)*T(3,1));
            df(i,5)=2*(x-
B(i,1))*(A(i,1)*sin(theta)*cos(psi)+A(i,2)*sin(phi)*cos(theta)*cos(psi)
+A(i,3))*cos(phi)*cos(theta)*cos(psi)+...
                    2*(y-B(i,2))*(-
A(i,1)*sin(theta)*sin(psi)+A(i,2)*sin(phi)*cos(theta)*sin(psi)+A(i,3)*c
os(phi)*cos(theta)*sin(psi))+...
                   -2*(z-
B(i,3))*(A(i,1)*cos(theta)+A(i,2)*sin(phi)*sin(theta)+A(i,3)*cos(phi)*s
in(theta));
            df(i,6)=2*(x-B(i,1))*(A(i,2)*T(3,1)-A(i,3)*T(2,1))+...
                    2*(y-B(i,2))*(A(i,2)*T(3,2)-A(i,3)*T(2,2))+...
                    2*(z-B(i,3))*(A(i,2)*T(3,3)-A(i,3)*T(2,3));
        end

        SV(:,k)=SV_old-inv(df)*f;
        SV_old=SV(:,k);
        x=SV(1,k); % surge
        y=SV(2,k); % sway
        z=SV(3,k); % heave
        phi=SV(4,k); % yaw
        theta=SV(5,k); % pitch
        psi=SV(6,k); % roll
    end

    DOF(:,j)=SV(:,end);

end

y=DOF;
% subplot(3,2,1);plot(Time,DOF(1,:)); title('surge')
% subplot(3,2,2);plot(Time,DOF(4,:)); title('yaw')
% subplot(3,2,3);plot(Time,DOF(2,:)); title('sway')
% subplot(3,2,4);plot(Time,DOF(5,:)); title('pitch')
% subplot(3,2,5);plot(Time,DOF(3,:)); title('heave')
% subplot(3,2,6);plot(Time,DOF(6,:)); title('roll')
```

Frequency Response Function Analysis Tool:

Written by Kirill Zaychik

```
%% %%%%%%%%%% Frequency Response Function Analysis Tool
%%%%%%%%%%%%%%%%%%%
%
% SYNTAX:   [F_Hz,F_rad,R_abs,R_db,Ph,H,
Coh]=FRFunction(x,y,time,Options)
%
% INPUTS:
% 1.    x       = input
% 2.    y       = output
% 3.    time    = time vector sampled at the same frequency
and of the
%                 same length as both the input and output.
% OUTPUTS:
% 1.    F_Hz    = frequency array       [Hz]
% 2.    F_rad   = frequency array       [rad]
% 3.    R_abs   = amplitude array       [absolute]
% 4.    R_db    = amplitude array       [dB]
% 5.    Ph      = phase array           [deg]
% 6.    H       = frequency response    [complex]
% 7.    Coh     = coherence function    []
%
% OPTIONS:
% 1.    NS                default = 7     number of segments
% 2.    overL    [%]      default = 50    [min 0, max 100]
this parameter
%       describes how much overlap is between two
consecutive segments of
%       the original signal
% 3.    wintype           default = 1     [min 1, max 3]
window type being
%       used. Default corresponds to the Hann window
% 4.    flagPorR          default = 0     [min 0, max 1]
flag for using the
%       "Periodic" or "Random Signals" approach for
computing the FRF. Default
%       corresponds to the "Random Signals" approach.
%
% NOTES:
% 1.    By default the algorithm will use the Hann window.
However, Hamming and
%       Blackman window are available as well. The choice
is dictated by an
%       optimal combination of the noise/performance of the
FIR of the window.
%       The code can easily be expanded to facilitate the
choice of different
%       windowing algorithms, such as:
```

```
%           -   bartlett
%           -   boxcar
%           -   chebwin
%           -   kaiser, etc.
%
% 2.    One should be aware of a delicate balance between
the length of
%       sub-segment of the input/output signal, the amount
of overlap and the
%       desired accuracy of the FRF estimate. "The rule of
thumb" here is as follows: the
%       larger the number of overlapping segments, the less
noisy the FRF
%       becomes. On the other hand, too much averaging may
result in the loss
%       of "important" information in the FRF estimate.
%
% EXAMPLE:
% OptionsFRF =
struct('NS',7,'overL',50,'wintype',1,'flagPorR',0);
% [F_Hz_d, F_rad_d, R_abs_d, R_db_d, Ph_d, H,
Coh]=FRFunction(In,Out,time,OptionsFRF);
%
% PROGRAMMER: Kirill Zaychik
%
% $ STATUS:  beta testing     $
% $ Revision: 1.5             $
% $ Date: 2011/06/29 12:50:41 $


function [F_Hz_d, F_rad_d, R_abs_d, R_db_d, Ph_d, H, Coh] =
FRFunction(In,Out,time,varargin)

warning('off','MATLAB:colon:nonIntegerIndex')
p = inputParser;

p.addParamValue('NS' ,      7 , @isnumeric);
p.addParamValue('overL' ,   50, @isnumeric);
p.addParamValue('wintype' ,  1, @isnumeric);
p.addParamValue('flagPorR' , 0, @isnumeric);
p.parse(varargin{:});

ParsedFields = fieldnames(p.Results);
for ii=1:length(ParsedFields)
        eval([ ParsedFields{ii},' =
p.Results.(ParsedFields{ii});']);
end
```

```
Time    = time;
dt      = (Time(50)-Time(49));  % define time vector
fHz     = ceil(1/dt);           % Sampling rate
nfft    = length(Time);         % length of the FFT is
equal to the length of the original signal
Trec    = Time(end);            % Total simulation time;
Nseg    = NS;                   % total number of segments
ovlp    = overL;                % amount of overlap between
consequtive segments (%)
winType = wintype;              % type of the window to use
Opt     = flagPorR;             % flag for using either
Periodic or Random Signals approach to compute the FRF

% Error check
sx      = length(In);
sy      = length(Out);
stime   = length(Time);
if sx ~= sy, error('Input and Output shall be the same
size'); end
if stime ~= sx, error('Time vector must be the same size as
both the Input and Output'); end
if stime ~= sy, error('Time vector must be the same size as
both the Input and Output'); end

%++++++++ transposes the input/output/time vectors if
needed +++++++++++++
sx      = size(In);
sy      = size(Out);
stime   = size(Time);

if sx(1) == 1
    input = In';
else
    input = In;
end

if sy(1) == 1
    output = Out';
else
    output = Out;
end

if stime(1) == 1
    Time = Time';
end
```

```
%++++++++++++++++ call FRF function builder
+++++++++++++++++++++++++
if Opt ~= 1
    Lwin = floor((length(input)-1)/((Nseg-1)*(1-
ovlp/100)+1));
    novlp = floor(Lwin*ovlp/100);

    %+++++++++++ segment both input and output signals
++++++++++++++
    dp=0;
    i=1;
    while dp<=(length(input)-1)
        sdp(i)=(i-1)*(Lwin-novlp)+1;
        edp(i)=(i-1)*(Lwin-novlp)+Lwin+1;
        if edp(i)>length(input)
            tail   = edp(i)-length(input);
            input  = [input;  zeros(tail,1)];
            output = [output; zeros(tail,1)];
        end
        dp=edp(i);
        i=i+1;
    end

    segment_t=Time(sdp(1):edp(1),1);

    for i=1:length(sdp)


        segment_i=input(sdp(i):edp(i));
        segment_o=output(sdp(i):edp(i));
        if winType == 1
            w = .5*(1-cos(2*pi*segment_t./segment_t(end)));
% HANN WINDOW
        elseif winType == 2
            w = .54-
.46*cos(2*pi*segment_t./segment_t(end));
% HAMMING WINDOW
        elseif winType == 3
            w = .42-
.5*cos(2*pi*segment_t./segment_t(end))+.08*cos(4*pi*segment
_t./segment_t(end));  % BLACKMAN WINDOW
        end

        segment_iw=segment_i.*w;
        segment_ow=segment_o.*w;

        Rf = fft(segment_iw,nfft);
        Yf = fft(segment_ow,nfft);

        Gryk1(:,i)  = 2/Trec*conj(Rf).*Yf;      %
crosscorrelation
        Gryk2(:,i)  = 2/Trec*conj(Yf).*Rf;      %
crosscorrelation
        Grrk(:,i)   = 2/Trec*(abs(Rf)).^2;      %
autocorrelation of the input
        Gyyk(:,i)   = 2/Trec*(abs(Yf)).^2;      %
autocorrelation of the output
    end
end

% averaging over the entire set of overlapping segments
if Opt ~= 1 %+++++++++++ Random Signals Approach
+++++++++++++++++++++++++
    Grrh    = mean(Grrk,2);
    Gyyh    = mean(Gyyk,2);
    Gryh1   = mean(Gryk1,2);
    Gryh2   = mean(Gryk2,2);

    % compute the FRF by taking the ratio:
    FRF1 = - Gryh1./(Grrh);
    FRF2 = - Gyyh./(Gryh2);

    Hf1  = FRF1(1:length(FRF1)/2+1);
    Hf2  = FRF2(1:length(FRF2)/2+1);
    H    = mean([Hf1 Hf2],2);
    F    = fHz/2*linspace(0,1,nfft/2+1);
    Coh  =
(abs(Gryh1(1:length(FRF1)/2+1)).^2)./(Grrh(1:length(FRF1)/2
+1).*Gyyh(1:length(FRF1)/2+1));
else %++++++++++++++++++ Periodic Signals Approach
+++++++++++++++++++++++
    Hf = - fft(output,nfft)./fft(input,nfft);
    H  = Hf(1:length(input)/2+1);
    F  = fHz/2*linspace(0,1,length(input)/2+1);
%     Coh = mscohere(input,output); % need license to use
this function
    Coh = [];
end

    R_abs_d     = abs(H);
% amplitude response (abs)
    R_db_d      = 20*log10(abs(H));
% amplitude response (dB)
%     Ph_rad      = pi/2*(1-
sign(real(H)))+unwrap(pi/180*atand(abs(imag(H)./real(H))),p
i).*sign(real(H)).*sign(imag(H));     % phase reponse (deg)
Explicit formula
%     Ph_d        = 180/pi*unwrap(Ph_rad,pi/2);
%     if sign(real(H(1))) == -1
%         Ph_d = Ph_d  - 180;
%     end
    Ph_d      = 180*(1/pi*unwrap(angle(H),pi/2) - 1);
% phase reponse (deg). As it is proposed by Matlab Tutorial
    F_Hz_d    = F';
% frequency array (Hz)
    F_rad_d   = F_Hz_d*2*pi;
% frequency array (rad/sec)
```

APPENDIX B

Hardware frequency response plots for the original and velocity feedback compensators.

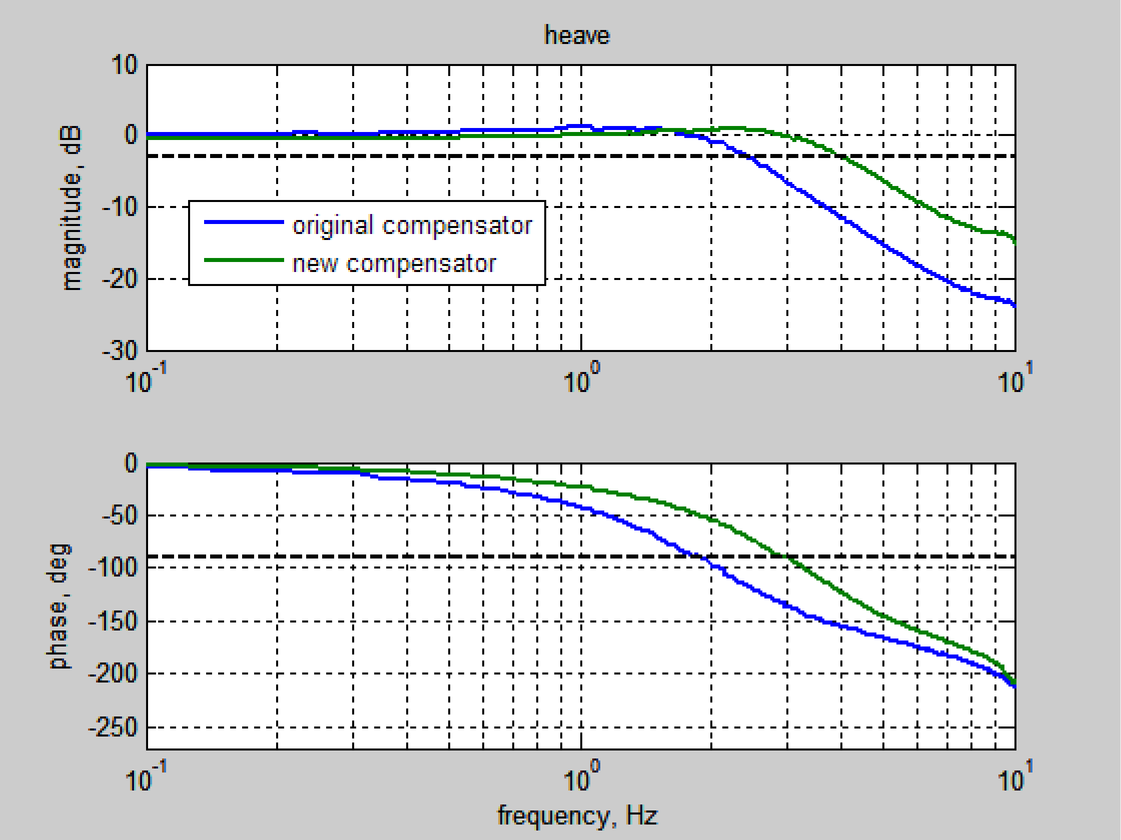

Figure B-1 - Comparison between Platform Responses with Original Compensator vs. Velocity Feedback Compensator. Translational DOF: Heave

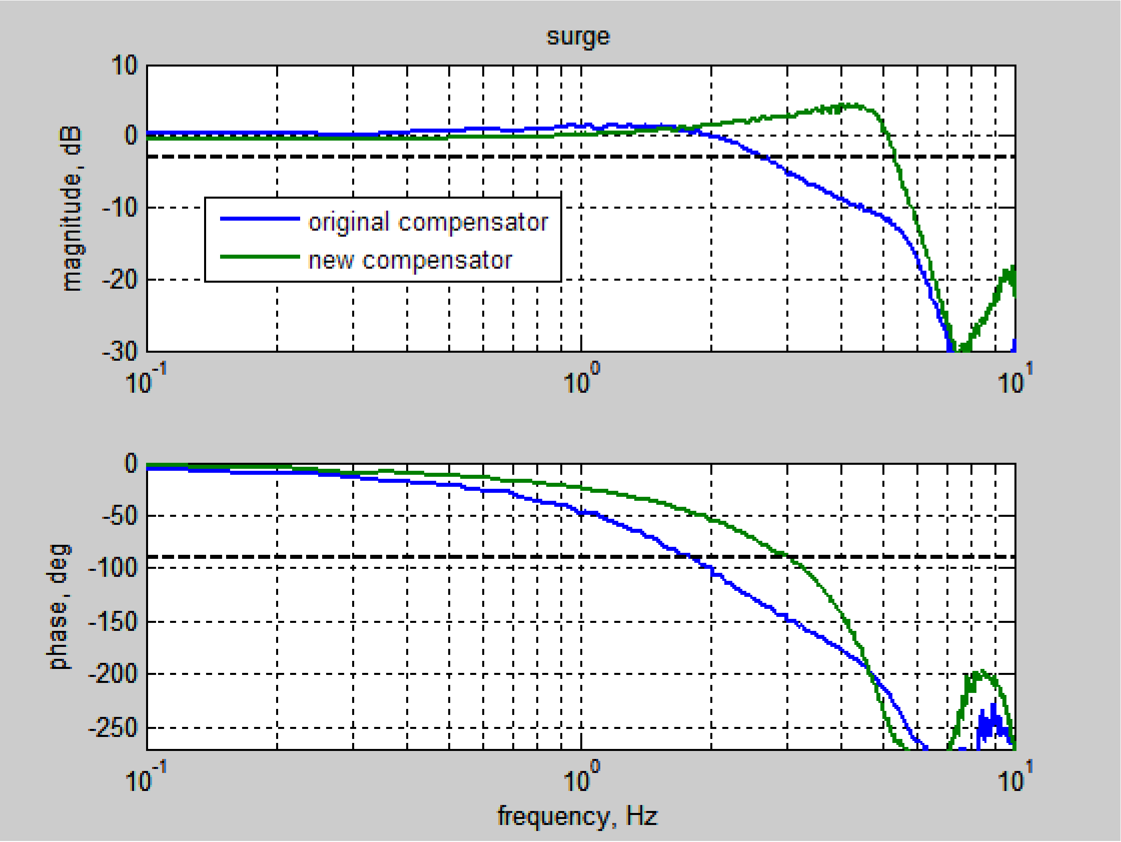

Figure B-2 - Comparison between Platform Responses with Original Compensator vs. Velocity Feedback Compensator. Translational DOF: Surge

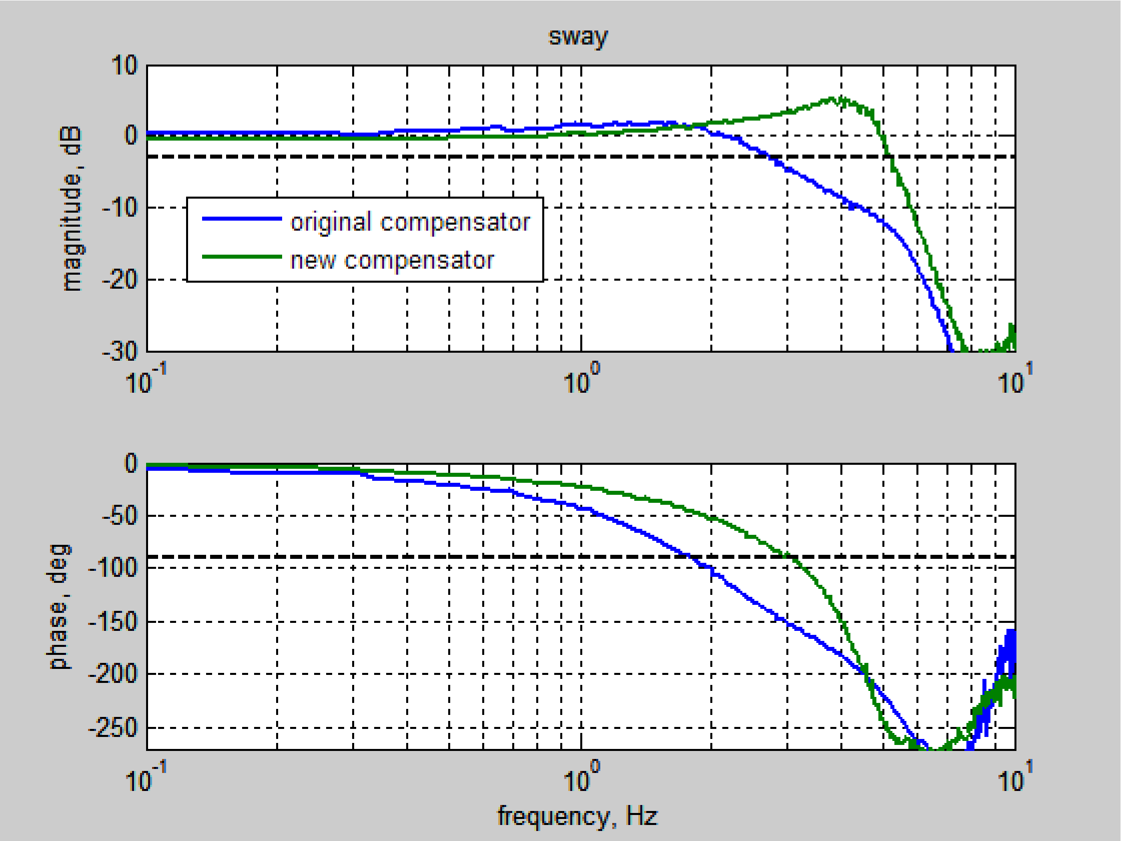

Figure B-3 - Comparison between Platform Responses with Original Compensator vs. Velocity Feedback Compensator. Translational DOF: Sway

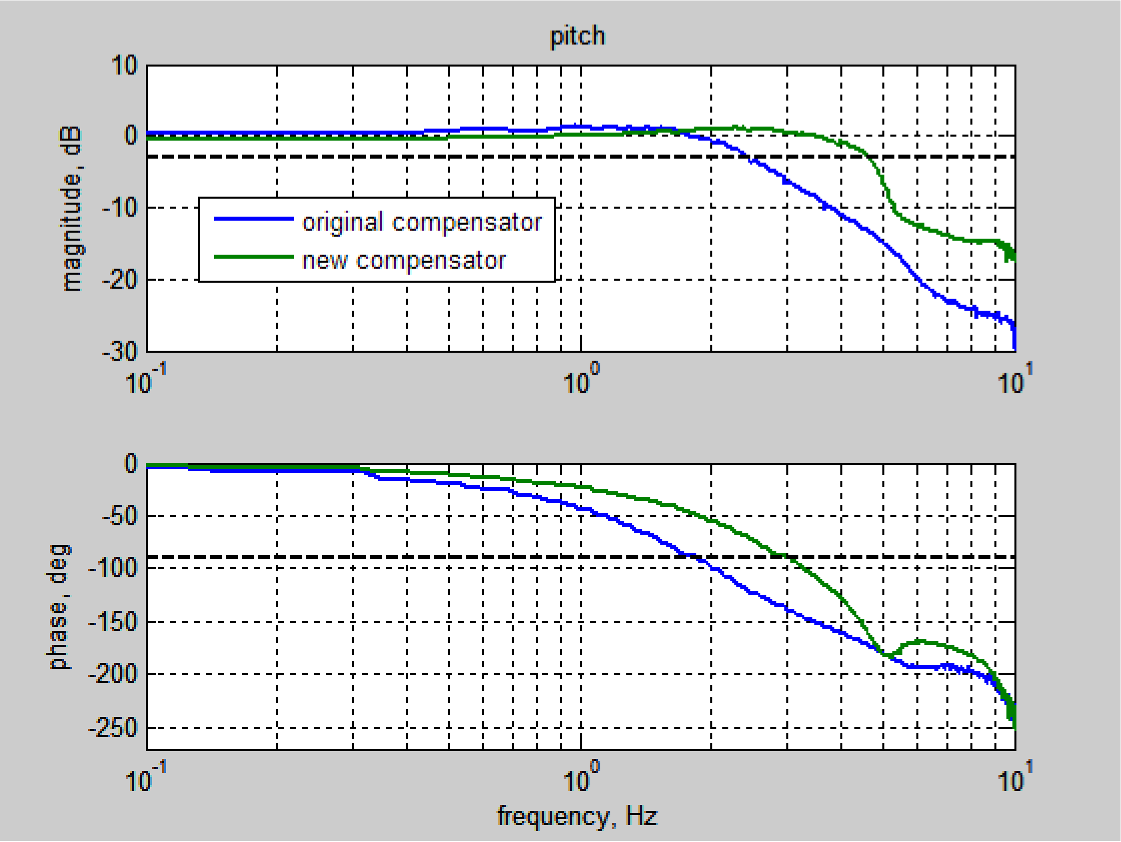

Figure B-4 - Comparison between Platform Responses with Original Compensator vs. Velocity Feedback Compensator. Rotational DOF: Pitch

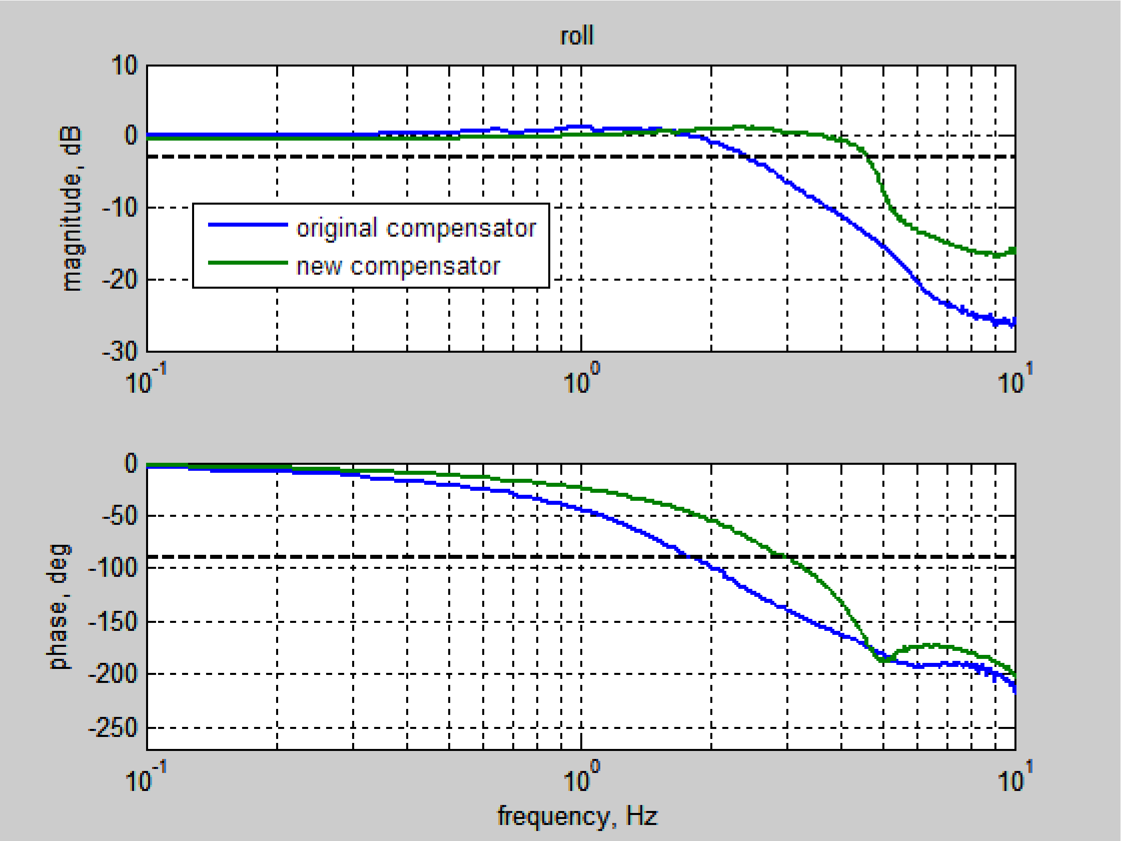

Figure B-5 - Comparison between Platform Responses with Original Compensator vs. Velocity Feedback Compensator. Rotational DOF: Roll

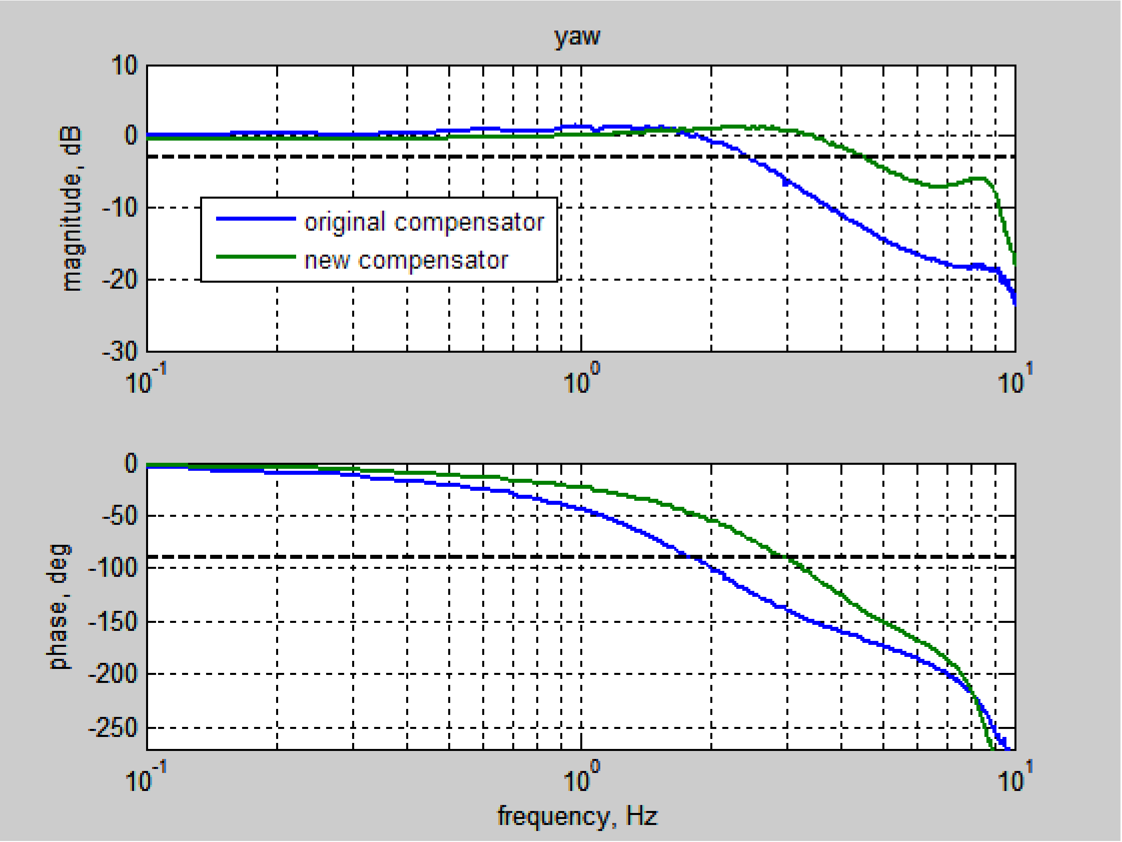

Figure B-6 - Comparison between Platform Responses with Original Compensator vs. Velocity Feedback Compensator. Rotational DOF: Yaw

APPENDIX C

C.1 Development of the Coordinate Transformation Matrix

APPENDIX C describes the development of the coordinate transformation matrix required to determine the component of the gravity vector along any given actuator of the six DOF motion platform.

C.2 Rationale for Updating the Model

It has been observed that the recorded leg acceleration signals demonstrate a bias due to the gravity component acting along the axis of a given actuator. This observation is supported by the fact that leg accelerometers are mounted not at (around) the actuator attachment point, but on the leg itself. The non-linear model, however, does not take into account such bias due to gravity. The computed leg accelerations have zero bias, which can potentially negatively impact the acceleration feedback design. To overcome such model deficiency it is proposed to add a coordinate transformation block to the model, which will compute the acceleration bias due to gravity for each leg of the motion platform.

C.3 Projection Matrix

Figure C-1 illustrates the concept of the projection matrix. The latter can be defined as the transformation matrix, which allows for determining the component of an arbitrary vector in the plane with the normalized vector $\bar{x}_n$, orthogonal to plane R.

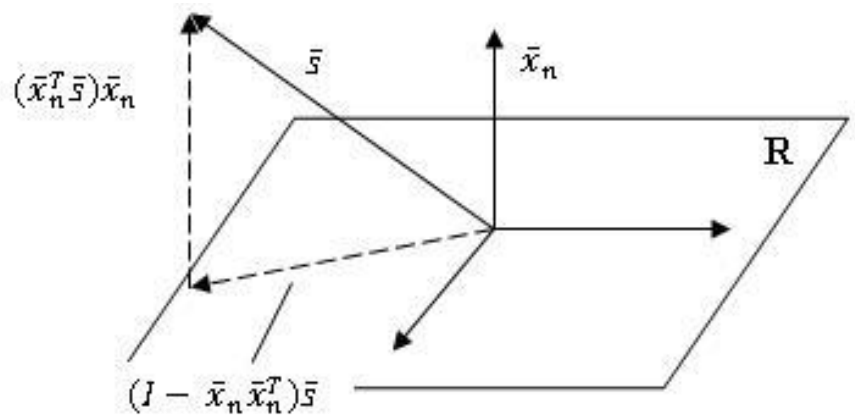
$(\bar{x}_n^T \bar{s})\bar{x}_n$
$\bar{s}$
$\bar{x}_n$
R
$(I - \bar{x}_n \bar{x}_n^T)\bar{s}$

Figure C-1 - Components of an arbitrary vector $\bar{s}$ projected onto the plane with normalized vector $\bar{x}_n$

Projection matrix is explicitly written in the equation below:

$$P_{x_n} = I - \bar{x}_n \bar{x}_n^T \tag{C.1}$$

Where $\bar{x}_n = \bar{x} \Big/ \sqrt{\bar{x}^T \bar{x}}$, and I is the identity matrix.

Koekebakker [11] discusses the impact of gravitational forces on the dynamics of the moving part of an actuator. Since such force is acting along the axis of a given actuator, it can be written as follows:

$$\bar{F}_G^a = m_a \left(I - \frac{r_a}{\left|\bar{l}\right|} P_{l_n} \right) \bar{g} \tag{C.2}$$

Where m_a is the mass of the moving part of the actuator and r_a is the distance between the upper gimbal and the center of gravity of the moving part of an actuator, and vector $\bar{l}$ is associated with the actuator itself. Since it is desired here to evaluate just the magnitude of the component of the gravity vector along the axis of an actuator, one can use the following formula:

$$\bar{g}^a = \left(I - \frac{r_a}{\left|\bar{l}\right|} P_{l_n} \right) \bar{g} \tag{C.3}$$

The following few paragraphs describe how it is possible to compute vector $\bar{l}$.

Figure C-2 illustrates relative orientation of two reference systems (RS). One associated with the centroid of the moving platform (simulator RS). The second one associated with centroid of the base of the motion system (inertial RS).

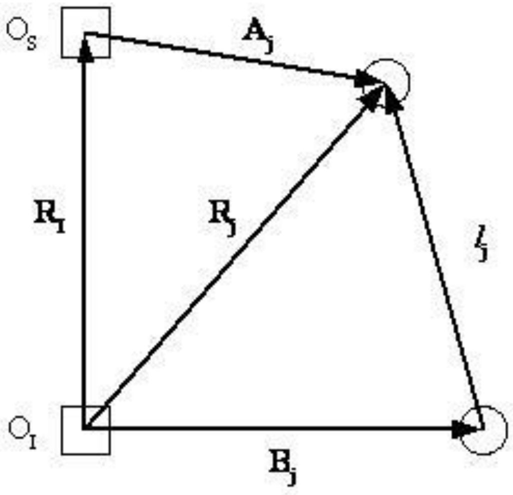

Figure C-2 - Vectors for the j-th Actuator

O_s and O_I are the centroids of the motion platform and fixed platform respectively, and are also respectively the origins for Fr_s and Fr_I. It can be seen that the relation among those vectors is:

$$R_I + A_j^I = R_j = B_j^I + l_j \tag{C.4}$$

The actuator length vector can then be found from:

$$l_j = A_j^I + R_I - B_j^I \tag{C.5}$$

The expression of l_j in the inertial reference frame Fr_I is desired:

$$\begin{aligned} l_j^I &= A_j^I + R_I - B_j^I \\ &= L_{SI} A_j^S + R_I - B_j^I \end{aligned} \tag{C.6}$$

where A_j^S are the coordinates of the upper bearing attachment point of the *j-th* actuator in Fr_S and B_j^I are the coordinates of the lower bearing attachment point of the *j-th* actuator in Fr_I.

Transformation matrix L_{SI} is a well-known directional cosine matrix:

$$L_{SI} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{C.7}$$

where ψ, θ and ϕ are Euler angles of the platform.

Therefore, Equations C.1, C.3, C.6 and C.7 are sufficient to compute the magnitude of the gravity vector along any given actuator. The required data are the position of the platform in DOF and the platform geometry.

Implementation

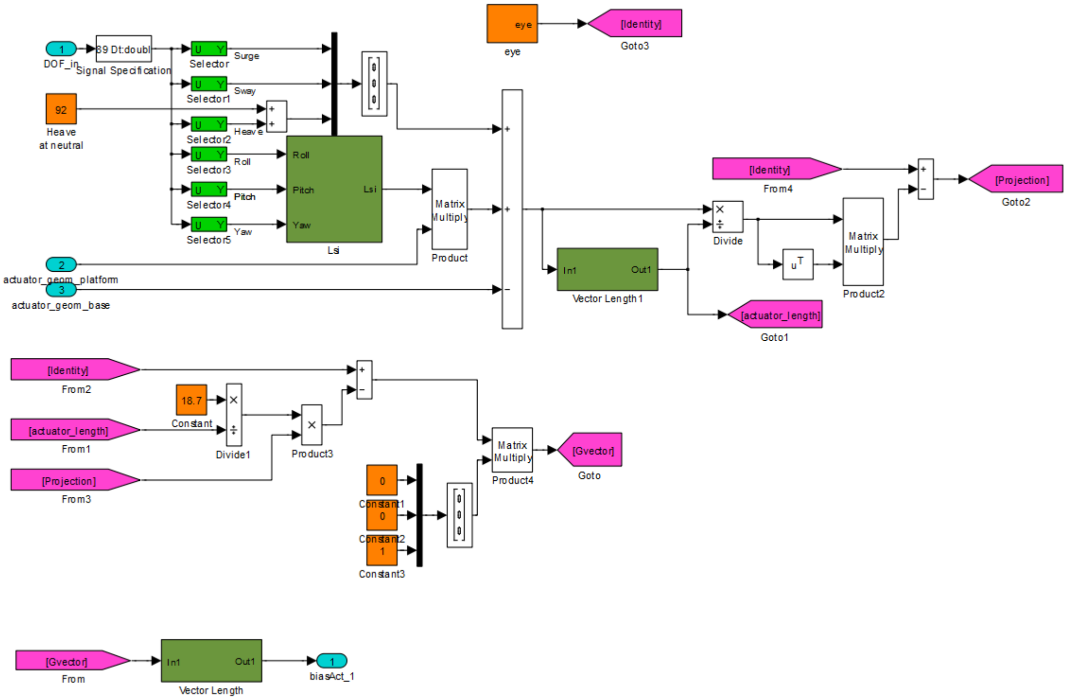

Figure C-3 - SIMULINK block diagram of the coordinate transformation

Figure C-3 contains a block diagram of the implementation of the discussed coordinate transformation in the SIMULINK environment. The system shown in Figure C-3 has been placed at the following address of the non-linear SIMULINK model of the CMF motion system: CMF_block\1 Six Single Leg Models\Six Single Leg Models\Payload\qddBiasCalc

The primary reason for such placement is the fact the CMF_block does not undergo auto code procedure. Moreover, to facilitate such model modification, the DCU2DCL signal had to be traced into and through the CMF_block.

APPENDIX D

The following are frequency responses for the position error rate feedback compensator and the acceleration feedback compensator in the degrees of freedom other than heave. The conclusions from the responses are summarized in Table D-1 and Table D-2 below. All translational degrees of freedom had input amplitudes of 0.33 inches while rotational degrees of freedom had an input of 0.2 degrees.

Table D-1 - Position Error Rate Feedback Compensator Performance Characteristics

Degree of Freedom	Phase (-90 Degree) Bandwidth (Hz)	Magnitude (-3 dB) Bandwidth (Hz)	Damping
Surge	4.0	6.2	0.4
Sway	3.8	6.0	0.4
Roll	4.0	6.2	0.4
Pitch	3.9	6.1	0.4
Yaw	3.7	5.9	0.4

Table D-2 - Acceleration Feedback Compensator Performance Characteristics

Degree of Freedom	Phase (-90 Degree) Bandwidth	Magnitude (-3 dB) Bandwidth	Damping
Surge	5.0	N/A	0.4
Sway	4.9	N/A	0.4
Roll	4.7	N/A	0.4
Pitch	4.7	N/A	0.4
Yaw	4.8	N/A	0.4

One of the main takeaways from reviewing the results in the other degrees of freedom after running them using the non-linear model is the similarity of each result. The reason for the similarity is due to a few of the assumptions made when developing the non-linear model. The lack of modeling the dynamic mass and the assumption that all of the valves are perfect allows for results in other degrees of freedom that become less accurate. This is the reason it is recommended that the new compensators are tested on the hardware before use in other research projects at the NASA LaRC facility.

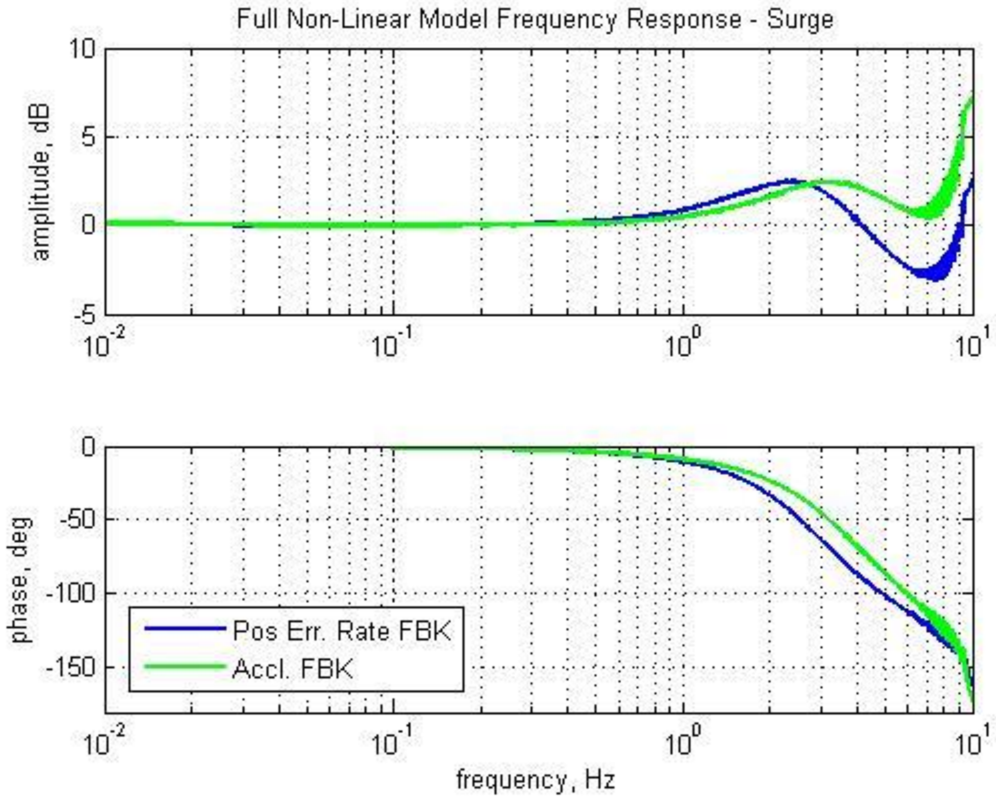

Figure D-1 - Comparison between the Position Error Rate Feedback and Acceleration Feedback in the Surge Degree of Freedom

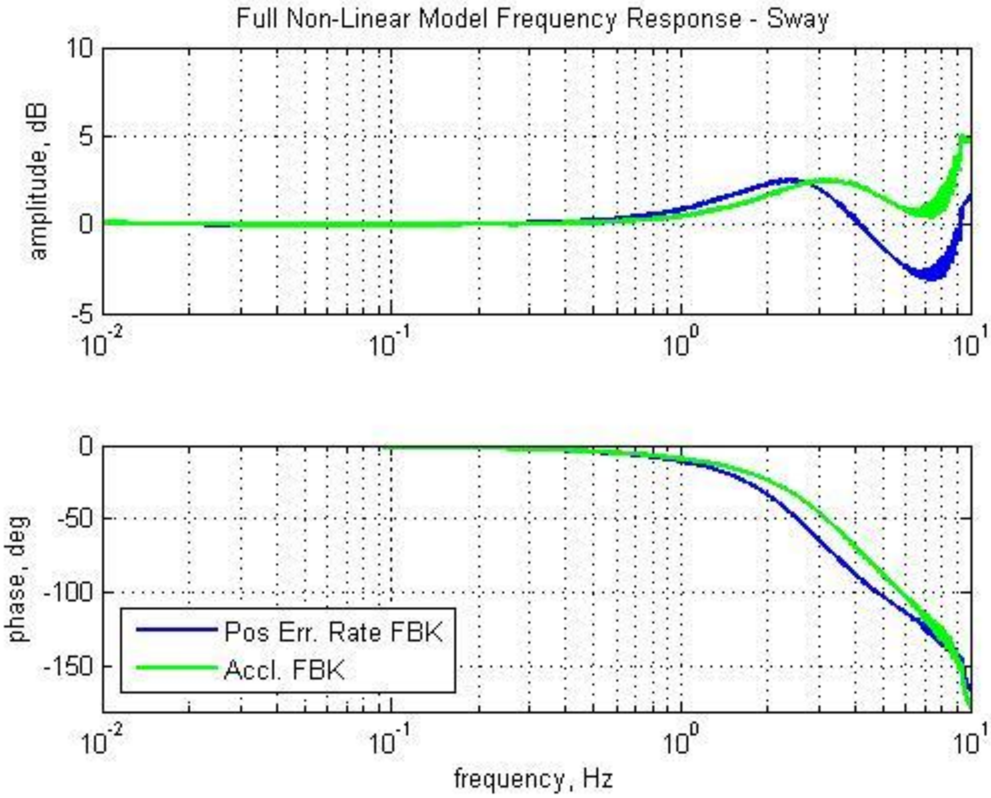

Figure D-2 - Comparison between the Position Error Rate Feedback and Acceleration Feedback in the Sway Degree of Freedom

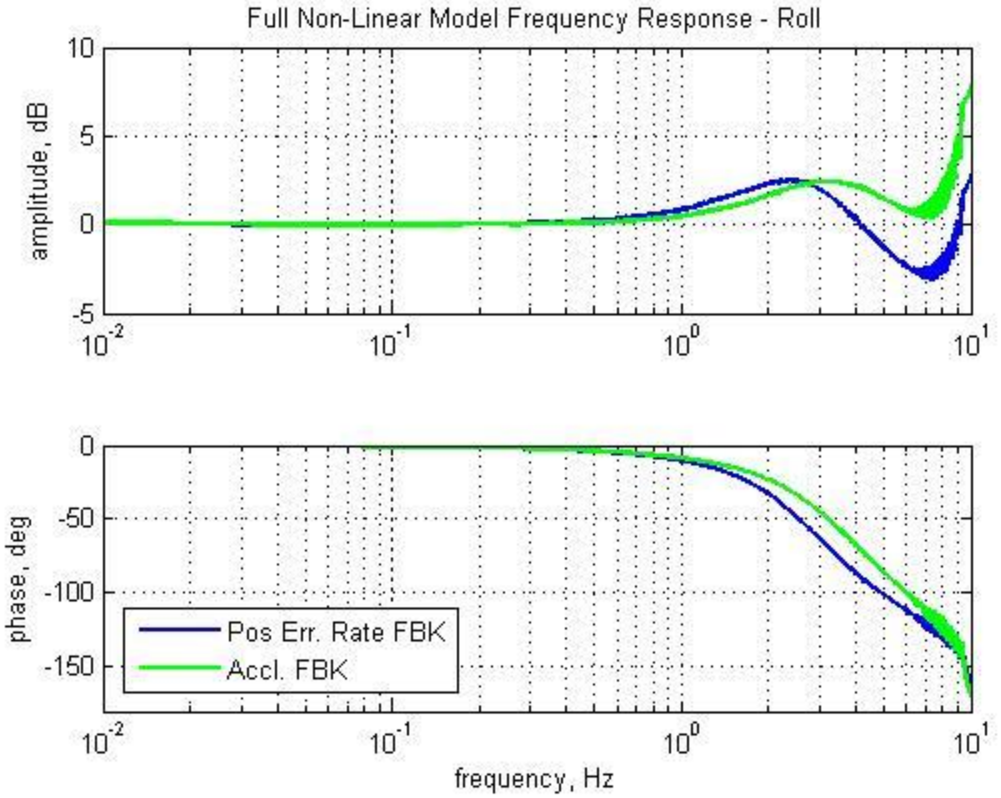

Figure D-3 - Comparison between the Position Error Rate Feedback and Acceleration Feedback in the Roll Degree of Freedom

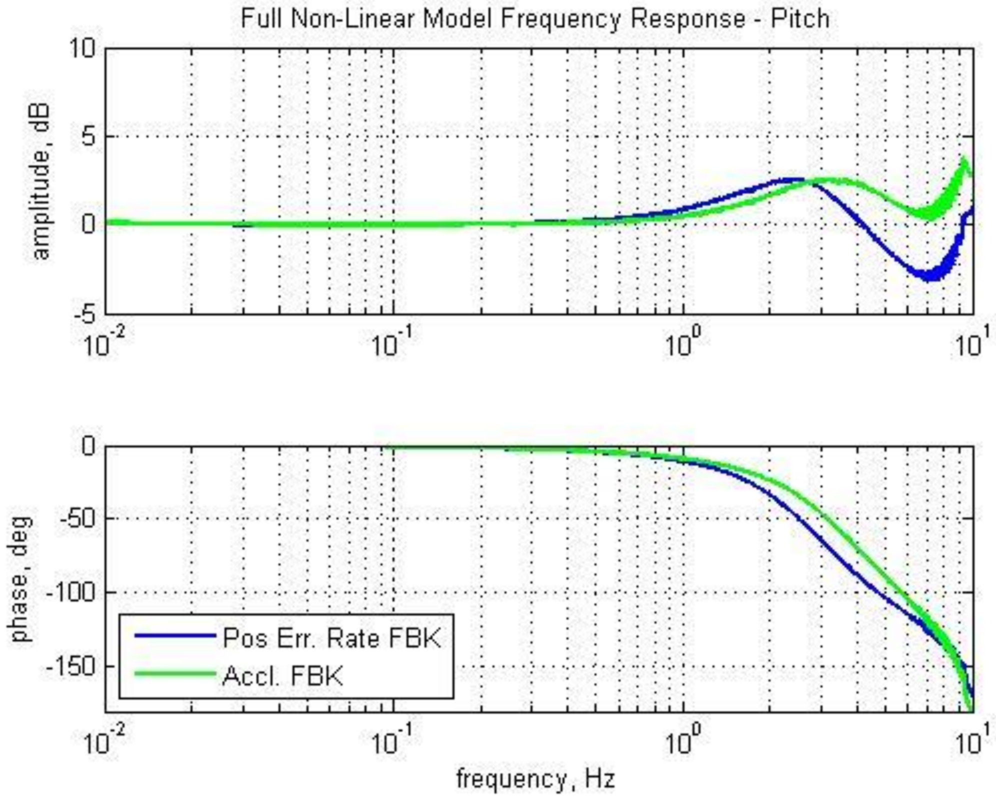

Figure D-4 - Comparison between the Position Error Rate Feedback and Acceleration Feedback in the Pitch Degree of Freedom

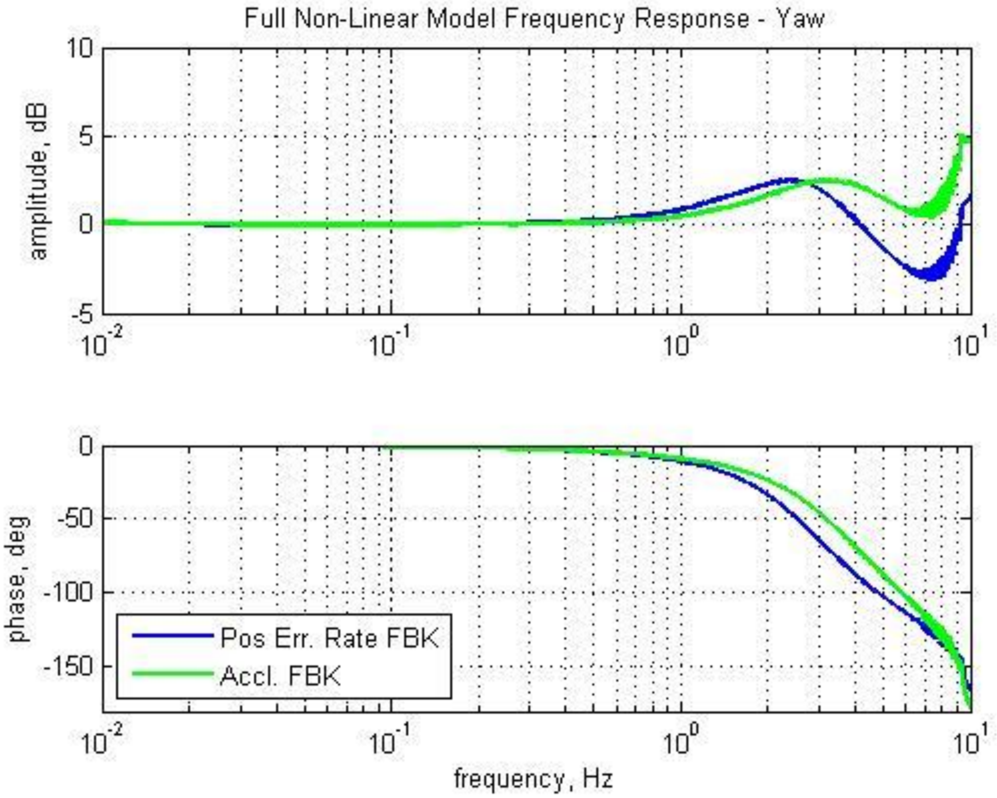

Figure D-5 - Comparison between the Position Error Rate Feedback and Acceleration Feedback in the Yaw Degree of Freedom

References

[1] Gundry, J.: "Thresholds to Roll Motion in a Flight Simulator", AIAA Visual and Motion Simulation Conference, Dayton, Ohio, April 1976

[2] Carrelli, D. J.: "Detailed Dynamic Modeling of the NASA LaRC CMF Motion Base", AIAA-2006-6362-CP, Keystone, CO, 2006

[3] Bryant, R. B.; Gupton, L. E.; Martinez, D.; and Carrelli, D. J.: "Mitigating Motion Base Safety Issues - The NASA LaRC CMF Implementation", AIAA-2005-6107-CP, San Francisco, CA, August 2005.

[4] Clark, B.; Stewart, J. D.: "Effects of Angular acceleration on Man: Thresholds for Perception of Rotation and the Oculogyral Illusion", Aerospace Medicine, September 1969.

[5] Telban, R. J.; Wu, W.; and Cardullo, F. M.: "Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms", NASA Langley Research Center, Hampton, VA, 2000.

[6] Bryant, R. B.: "CMF Motion Base Control System Derived Requirements, Version A", NASA Langley Research Center, Hampton, VA, February 5, 2001.

[7] Bryant, R. B.: "Servo Electronics Block Diagram", NASA Langley Research Center, Hampton, VA.

[8] Carrelli, D. J.: "Control Law Development", NASA Langley Research Center, Hampton, VA, 2001.

[9] Ogata, K.: "Modern Control Engineering, 5th Edition", Boston: Prentice Hall, 2010.

[10] Toffner-Clausen, S.: "System Identification and Robust Control: A Case Study Approach", London: Springer, 1996.

[11] Koekebakker, S.: "Model Based Control of a Flight Simulator Motion System", Ponsen & Looijen b.v., Delft, Netherlands, 2001

Bibliography

- Bryant, R. B.: “CMF Motion Base Control System Derived Requirements, Version A”, NASA Langley Research Center, Hampton, VA, February 5, 2001.
- Bryant, B.R.: “CMF Motion Base Control System. Version C”, NASA Langley Research Center, Hampton, VA, 2001.
- Bryant, R. B.: “Servo Electronics Block Diagram”, NASA Langley Research Center, Hampton, VA.
- Bryant, B. R.; and Carrelli, D. J.: “Software Tools for Developing and Simulating the NASA LaRC CMF Motion Base”, AIAA-2006-6363-CP, Keystone, CO, 2006
- Bryant, B. R.; Carrelli, D. J.; and Gupton, L. E.: “Evaluating the Performance of the NASA LaRC CMF Motion Base Safety Devices”, AIAA-2006-6364-CP, Keystone, CO, August 2006.
- Bryant, R. B.; Gupton, L. E.; Martinez, D.; and Carrelli, D. J.: “Mitigating Motion Base Safety Issues - The NASA LaRC CMF Implementation”, AIAA-2005-6107-CP, San Francisco, CA, August 2005.
- Carrelli, D. J.: “Control Law Development”, NASA Langley Research Center, Hampton, VA, 2001.
- Carrelli, D. J.: “Critical Design Review Control Law Development”, Hampton, VA: Swales Aerospace, 2001.
- Carrelli, D. J.: "Dynamic Modeling and Analysis," NASA Langley Research Center, Hampton, VA, 2004.
- Carrelli, D. J.: “Detailed Dynamic Modeling of the NASA LaRC CMF Motion Base”, AIAA-2006-6362-CP, Keystone, CO, 2006
- Chapra, S. C.: “Applied Numerical Methods with MATLAB for Engineers and Scientists”, Boston: McGraw-Hill, 2008.
- Clark, B.; and Stewart, J. D.: “Effects of Angular Acceleration on Man: Thresholds for Perception of Rotation and the Oculogyral Illusion”, Aerospace Medicine, September 1969.
- Goodwin, G. C.; Graebe, S. F.; and Salgado, M. E.: “Control System Design”, Upper Saddle River, New Jersey: Prentice Hall, 2001.
- Gundry, J.: “Thresholds to Roll Motion in a Flight Simulator”, AIAA Visual and Motion Simulation Conference, Dayton, Ohio, April 1976
- Hellendoorn, H.; and Driankov, D.: “Fuzzy Model Identification”, Berlin: Springer, 1997.
- Ikonen, E.; and Najim, K.: “Advanced Process Identification and Control”, New York: Marcel Dekker, Inc., 2002.

- Jategaonkar, V.: “Flight Vehicle System Identification: A Time Domain Approach”, Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2006.
- Kalouptsidis, N.; and Theodoridis, S.: “Apadtive System Identification and Signal Processing Algorithms”, New York: Prentice Hall, 1993.
- Koekebakker, S.: “Model Based Control of a Flight Simulator Motion System”, Ponsen & Looijen b.v., Delft, Netherlands, 2001.
- Milanese, M.; J. Norton, J.; Piet-Lahanier, H.; and Walter, É.: “Bounding Approaches to System Identification”, New York: Plenum Press, 1996.
- Ogata, K.: “Modern Control Engineering, 5th Edition”, Boston: Prentice Hall, 2010.
- Smith, R. M.: “A Description of the Cockpit Motion Facility and the Research Flight Deck Simulator”, NASA Langley Research Center, Hampton, VA, 2000.
- Telban, R. J.; and Cardullo, F. M.: “Motion Cueing Algorithm Development: Human-Centered Linear and Nonlinear Approaches”, NASA/CR-2005-213747, 2005.
- Telban, R. J.; Wu, W.; and Cardullo, F. M.: “Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms”, NASA Langley Research Center, Hampton, VA, 2000.
- Tischler, M. B.; and Remple, R. K.: “Aircraft and Rototcraft System Identification: Engineering Methods with Flight Test Examples”, Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2006.
- Toffner-Clausen, S.: “System Identification and Robust Control: A Case Study Approach”, London: Springer, 1996

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 01- 06 - 2014	**2. REPORT TYPE** Contractor Report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE Investigation and Development of Control Laws for the NASA Langley Research Center Cockpit Motion Facility	**5a. CONTRACT NUMBER** NNL11AA08C
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Coon, Craig R.; Cardullo, Frank M.; Zaychik, Kirill B.	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 160961.01.01.01
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, Virginia 23681	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	**10. SPONSOR/MONITOR'S ACRONYM(S)** NASA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/CR-2014-218275

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 54
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES
This report was prepared by State University of New York, Binghamton, NY, under NASA Contract NNL11AA08C with UNISYS Corporation, Reston, VA.
Langley Technical Monitor: Victoria I. Chung

14. ABSTRACT

The ability to develop highly advanced simulators is a critical need that has the ability to significantly impact the aerospace industry. The aerospace industry is advancing at an ever increasing pace and flight simulators must match this development with ever increasing urgency. In order to address both current problems and potential advancements with flight simulator techniques, several aspects of current control law technology of the National Aeronautics and Space Administration (NASA) Langley Research Center's Cockpit Motion Facility (CMF) motion base simulator were examined. Preliminary investigation of linear models based upon hardware data were examined to ensure that the most accurate models are used. This research identified both system improvements in the bandwidth and more reliable linear models. Advancements in the compensator design were developed and verified through multiple techniques. The position error rate feedback, the acceleration feedback and the force feedback were all analyzed in the heave direction using the non-linear model of the hardware. Improvements were made using the position error rate feedback technique. The acceleration feedback compensator also provided noteworthy improvement, while attempts at implementing a force feedback compensator proved unsuccessful.

15. SUBJECT TERMS
Compensator design; Control laws; Motion systems; Simulators

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER ***(Include area code)***
U	U	U	UU	121	(443) 757-5802