NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXPERIMENTAL TEST RIG FOR OPTIMAL CONTROL OF FLEXIBLE SPACE ROBOTIC ARMS

by

Martin J. Griggs

December 2016

Thesis Advisor: Mark Karpenko
Second Reader: I. Michael Ross

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE EXPERIMENTAL TEST RIG FOR OPTIMAL CONTROL OF FLEXIBLE SPACE ROBOTIC ARMS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Martin J. Griggs	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The goal of this thesis was to build an experimental test rig for demonstrations on flexible space systems control. Specifically, an air-bearing test bed incorporated a two-degree of freedom (2DOF) rigid robotic arm and an appendage with flexible joints to test the effects of movement of the robotic arm on the appendage. The two-link, 2DOF rigid robotic arm can be used to simulate a moving space antenna or other movable appendages. Optimal trajectories of the two-link arm to simulate a conventional antenna slewing maneuver were investigated, to illustrate the type of flexible motion that may be produced in the laboratory. An iterative process was used to refine the test bed design and the experimental workflow. Three concepts incorporated various strategies to design a robust flexible link. Inertia measurement units (IMU), a central processor for data analysis, power distribution, and robotics software, are all integrated as part of the test bed design. A single link arm with a torsional, helical spring at the base was finalized to investigate the effects of coupling due to movement of the rigid two-link arm. The torsional spring allowed the vibrating arm to displace sufficiently to have a high signal-to-noise ratio compared to earlier concepts in which IMU noise dominated the response. The test bed was designed to accommodate further testing that may require increased loading due to, for example, the incorporation of reaction wheels or additional instrumentation.

14. SUBJECT TERMS degree of freedom, optimal control, optimal trajectory, vibration analysis, satellite, flexible space systems, air-bearing, slew rates	**15. NUMBER OF PAGES** 141
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

EXPERIMENTAL TEST RIG FOR OPTIMAL CONTROL OF FLEXIBLE SPACE ROBOTIC ARMS

Martin J. Griggs
Commander, United States Navy
B.S., University of Florida, 1999

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2016

Approved by: Mark Karpenko
Thesis Advisor

I. Michael Ross
Second Reader

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The goal of this thesis was to build an experimental test rig for demonstrations on flexible space systems control. Specifically, an air-bearing test bed incorporated a two-degree of freedom (2DOF) rigid robotic arm and an appendage with flexible joints to test the effects of movement of the robotic arm on the appendage. The two-link, 2DOF rigid robotic arm can be used to simulate a moving space antenna or other movable appendages. Optimal trajectories of the two-link arm to simulate a conventional antenna slewing maneuver were investigated, to illustrate the type of flexible motion that may be produced in the laboratory. An iterative process was used to refine the test bed design and the experimental workflow. Three concepts incorporated various strategies to design a robust flexible link. Inertia measurement units (IMU), a central processor for data analysis, power distribution, and robotics software, are all integrated as part of the test bed design. A single link arm with a torsional, helical spring at the base was finalized to investigate the effects of coupling due to movement of the rigid two-link arm. The torsional spring allowed the vibrating arm to displace sufficiently to have a high signal-to-noise ratio compared to earlier concepts in which IMU noise dominated the response. The test bed was designed to accommodate further testing that may require increased loading due to, for example, the incorporation of reaction wheels or additional instrumentation.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

cRIO	Compact reconfigurable input/output real-time controller
DOF	Degree of freedom
HMC	Hamiltonian minimization condition
IC	Integrated Circuit
IMU	Inertial Measurement Unit
LEO	Low Earth Orbit
NI	National Instruments
NPS	Naval Postgraduate School
SSAG	Space Systems Academic Group
TDRS	Tracking and Data Relay Satellite
V&V	Validation and verification

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Mark Karpenko and Dr. I. Michael Ross, for their support, guidance, and trust. I couldn't have made it this far or learned so much without their help. Additionally, I'd like to thank Mr. Dan Sakoda and the Space Systems Academic Group (SSAG) for use of the 3D printer. Mr. Sakoda helped me through numerous iterations of designing and creating the vibrating arm components. On that note, I'd also like to thank Mr. John Mobley with the school's machine shop. As with Mr. Sakoda, Mr. Mobley's help was critical to the test bed components' development. Without them or the tools made available at this institution, this test bed would not be what it is today. Last but not least, I'd like to thank Midshipman Aaron Martinez. Mid. Martinez was at NPS as part of his summer internship. His help was instrumental in the research, development, and construction of the test rig data handling and processing used in the National Instruments cRIO embedded computer.

I especially want to thank for my wife, Taesha, whose tremendous support, patience, and strength during our time at the Naval Postgraduate School has allowed me the time to work toward the end product delivered. I'd also like to thank God for the opportunities we've been given and the strength to face the challenges ahead. And for my son, Broden, thank you for inspiring me to work hard to be the best husband I can be for your Mommy and the best daddy I can be for you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Vibration is defined as "periodic back-and-forth motion of the particles of an elastic body or medium, commonly resulting when almost any physical system is displaced from its equilibrium condition and allowed to respond to the forces that tend to restore equilibrium" [1]. Aircraft, automobiles, buildings, bridges, etc. all require significant consideration to vibration analysis in their design. To minimize vibrations in automobiles and improve driver comfort, shock absorbers are incorporated into the suspension. Skyscrapers such as Taipei 101 Tower use tuned mass dampers to reduce the amplitude of vibrations imposed by forces from wind to earthquakes by absorbing kinetic energy [2]. Simply making a system more rigid with more structural components, screws, bolts, welds, etc. is another way to reduce vibrations [3].

Vibration design consideration for spacecraft is no different. In the launch environment, the launch as well as the payload, undergoes significant stresses and vibrations. When the spacecraft is on orbit, any appendage movement, such as unfurling the payload, an antenna arm slew, camera slewing, or solar panel deployment can impart vibrations on the entire spacecraft. Vibrations on a spacecraft affect the components and structural integrity, but also, for example, the pointing accuracy of an imaging system or an antenna's communication link. On Earth, a system can generally be fixed or repaired in the event of a mishap or complication resulting from excess vibrations; however, after a spacecraft is on orbit, one does not have that luxury. Spacecraft and satellites are complex systems. Robust spacecraft design and testing must be conducted on the ground to ensure the system will continue to operate within specifications during all regimes it encounters in its design life. Figure 1 illustrates the complexity of satellite antenna arms in the Tracking and Data Relay Satellite (TDRS).

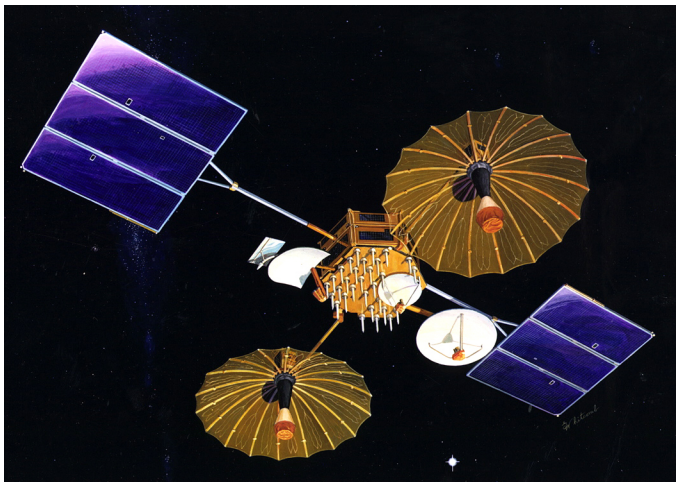

Figure 1. Tracking and Data Relay Satellite (TDRS). Source: [4].

A challenge with space systems is that it costs a lot of money to put a satellite or spacecraft into space. Estimates to send one kilogram of mass into low Earth orbit range from $2,720 for a SpaceEx Falcon 9 [5] to $20,200 for an Atlas V [6]. With this said, designing a rigid spacecraft with components used for vibration reduction introduces more mass to the system and therefore more cost as well. Alternately, a design choice to manage spacecraft vibrations is to control appendage deployment or antenna arm slew rates. However, what if a program required a satellite camera to take more pictures as it flew from horizon to horizon? What if the specification for the rate of an antenna to establish a link with a ground site or other satellite was twice that of previous designs?

These demands can be met if there is a way to control vibrations without adding mass. Furthermore improving vibration control methods for satellites that are already on orbit through software change, and thereby increasing their performance, allows that performance to be extended without the costs of designing and launching an entirely new system. Developing new motion control systems that optimize movement in a way that manages vibrations is indeed possible as evidenced by recent work at the Naval Postgraduate School [7], [8]. Numerous papers have been dedicated to the development of flexible space systems. In Steven Wojdakowski's 2015 thesis [7], he developed models for two degree of freedom flexible and double gimbal systems. However, there is little to

no evidence of work dedicated to building a two degree of freedom system with rigid links, yet a purposefully flexible joint in a laboratory setting.

In order to transition these ideas into practice they must be thoroughly vetted in the laboratory. Thus, the goal of this thesis is to develop a laboratory test bed for testing the new ideas.

B. OBJECTIVE AND SCOPE

This thesis focuses on the development of a simple laboratory test bed that can be used to test new ideas for control of flexible space systems. An understanding of the concepts is illustrated through the construction of a multi-body system involving a two-DOF robot arm mounted to a base plate with a flexible joint appendage also mounted to the base plate. In order to produce the forces necessary to observe vibration, three flexible joint concepts were developed while incorporating a previously developed two-link robotic arm with electric actuators mounted to an air-bearing test bed. The test bed, when applied with air pressure, rotates freely when excited. Figure 2 shows the test bed early in the development process that was developed, designed, and tested as part of this thesis.

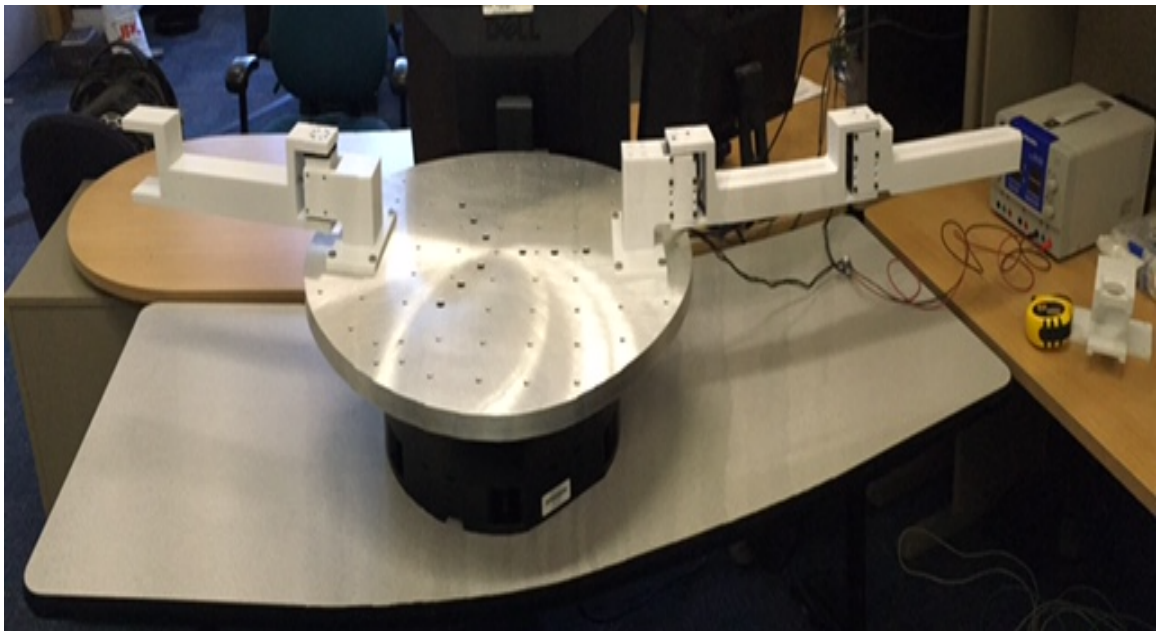

Figure 2. Laboratory Test Bed Early in the Development Process

The test bed provides researchers the capability of designing and testing concepts such as optimal control methods in a laboratory setting so they can be transitioned to present and future space systems. In all experiments and modeling, it is assumed that the moving links are rigid; that is, only the joints are considered flexible. Design of these joints forms a large part of the development work of the thesis. In all, three flexible joint concepts were designed to perform the experimentation. The first and second concepts use traditional elastic springs in varying configurations while a third uses a specially machined torsional spring. Additionally, inertia measurement units and an embedded computer were incorporated for system response measurement and data analysis. Through experimentation, a basis for future thesis work is developed to demonstrate how optimal control trajectories can be used to minimize vibrations in spacecraft slewing maneuvers.

Starting with the two-link arm, a standard maneuver will be designed using optimal control. The control torques used to move the arm impose a disturbance onto the flexible links. To assess this disturbance, a simple model is built. The model parameters will be adjusted to help bracket constants for design of a flexible joint suitable for laboratory demonstrations. Different approaches can be used to minimize vibrations on appendages. Simply reducing the slew rates on an antenna arm, which is common practice in industry today, will reduce vibrations. Although not tested in this thesis, the optimal control method can also be used with a model of the entire system to reduce vibration without reducing slew rates. The test bed is designed to enable experimentation to compare and contrast the various techniques.

C. THESIS OUTLINE

To demonstrate the background for the test bed Chapter II starts the thesis with a discussion of optimal control theory for a 2DOF robotic arm slew through 90 degrees. Chapter III uses the torque profile for the time-optimal trajectory found in Chapter II as the basis for a model to predict the amount of displacement that can be expected from the flexible system for such a maneuver. In Chapter IV, the details of the test bed design and assembly are discussed as well as the data processing procedures to measure the vibration

response. Finally in Chapter V, the experimental results for each concept are illustrated ending with a decision on the appropriate flexible joint and post-processing methods to be used in future work. Conclusion on future work is discussed in Chapter VI.

THIS PAGE INTENTIONALLY LEFT BLANK

II. OPTIMAL CONTROL OF A TWO-LINK ROBOT

Chapter II investigates time optimal control of a two-link 2DOF, planar arm. To begin, the equations of motion for the system are developed. The experimental hardware and software used to implement the trajectory is then detailed. Finally, with the equations of motion derived and scaled and parameters of the hardware and software incorporated, time optimal control methods are examined to produce an optimal trajectory for a 90-degree slewing maneuver. The motion profile obtained this chapter is a version of the industry standard where each appendage of a spacecraft is considered separate from the others. Interaction between the various appendages is not considered.

A. EQUATIONS OF MOTION

The equations of motion were developed for a two-degree of freedom (2DOF), two-link planar robotic arm to provide a basic model for the movement of a satellite antenna or other appendage. It was assumed at present that the links and joints are rigid; i.e., a rigid-body system. Figure 3 illustrates the setup.

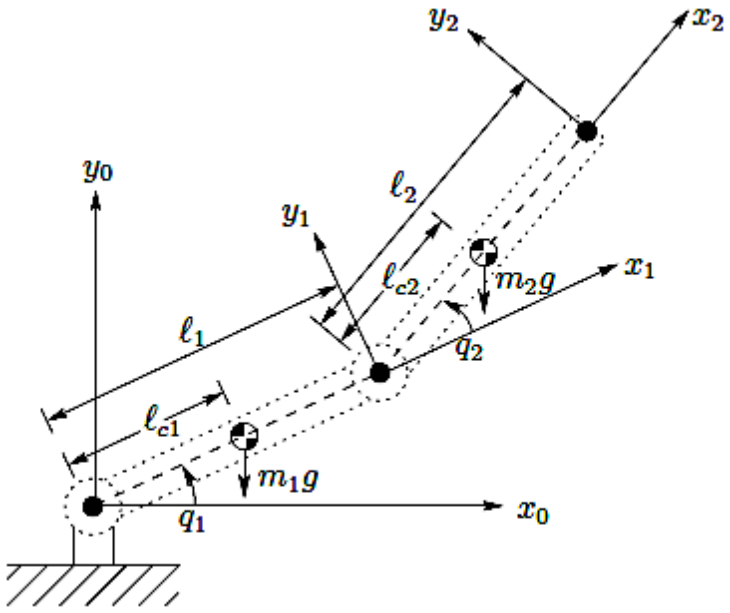

Figure 3. Rigid-Body 2DOF Planar Robot. Source: [9].

The kinematic and dynamic equations of motion for the movement of a 2DOF planar robotic arm were obtained from reference [9] and are shown in (2.1) and (2.2). It was assumed that the movement of the arms was in the horizontal plane, i.e., parallel to the ground. Therefore, in the laboratory setup, gravity terms will not appear in the equations of motion. By eliminating the gravity terms, this also provides a better proxy for space systems considering the lack of gravity in outer space. Additionally, the robot arm model is commanded in a comparable manner to a satellite antenna arm. That is, the robot arm, like the antenna arm must travel through a certain trajectory to perform a slew maneuver from point-to-point.

The dynamic model of the robot arm is given by [9]:

$$m_{11}\ddot{\theta}_1 + m_{12}\ddot{\theta}_2 - 2r\dot{\theta}_1\dot{\theta}_2 - r\dot{\theta}_2^2 = \tau_1 \tag{2.1}$$

$$m_{22}\ddot{\theta}_1 + m_{21}\ddot{\theta}_2 + r\dot{\theta}_1^2 = \tau_2 \tag{2.2}$$

where $\tau_1$and τ_2 are the torques applied at each joint, $\theta_1$and $\theta_2$are the angles displaced of each link $l_1$and l_2 are the lengths of the links, d_1 and d_2 are the distances from the each joint to the respective joints center of mass, and I_1 and I_2 are the mass moments of inertia of each link. Equations (2.3) – (2.5) illustrate the derivation of these values.

$$m_{11} = I_1 + I_2 + m_1 l_1^2 + m_2(d_1^2 + l_2^2 + 2d_1 d_2 \cos\theta_2) \tag{2.3}$$

$$m_{12} = m_{21} = I_2 + m_2(l_2^2 + d_1 l_2 \cos\theta_2) \tag{2.4}$$

$$m_{22} = I_2 + m_2 l_2^2 \tag{2.5}$$

The parameter '*r*' is formulated for convenience from the Christoffel symbols, which are used to derive the Coriolis matrix. The Christoffel symbols [8] are given as:

$$c_{111} = \frac{1}{2}\frac{\partial m_{11}}{\partial \theta_1} = 0 \tag{2.6}$$

$$c_{112} = c_{121} = \frac{1}{2}\frac{\partial m_{11}}{\partial \theta_2} = -m_2 l_1 l_2 \sin\theta_2 \tag{2.7}$$

$$c_{122} = \frac{\partial m_{12}}{\partial \theta_2} - \frac{1}{2}\frac{\partial m_{22}}{\partial \theta_1} = -m_2 l_1 l_2 \sin\theta_2 \tag{2.8}$$

$$c_{211} = \frac{\partial m_{21}}{\partial \theta_1} - \frac{1}{2}\frac{\partial m_{11}}{\partial \theta_2} = -m_2 l_1 l_2 \sin\theta_2 \tag{2.9}$$

$$c_{212} = c_{221} = \frac{1}{2}\frac{\partial m_{22}}{\partial \theta_1} = 0 \tag{2.10}$$

$$c_{222} = \frac{1}{2}\frac{\partial m_{22}}{\partial \theta_2} = 0 \tag{2.11}$$

The Coriolis matrix is then derived from the Christoffel symbols and defined as:

$$C = -m_2 l_1 l_2 \sin\theta_2 \begin{pmatrix} \dot{\theta}_2 & \dot{\theta}_1 + \dot{\theta}_2 \\ \dot{\theta}_1 & 0 \end{pmatrix} \tag{2.12}$$

Thus, the parameter '*r*' is obtained from the Coriolis matrix as:

$$r = m_2 l_1 l_2 \sin\theta_2 \tag{2.13}$$

B. LABORATORY PROTOTYPE ARM

The robotic arm linkage developed in this thesis is shown in Figure 4.

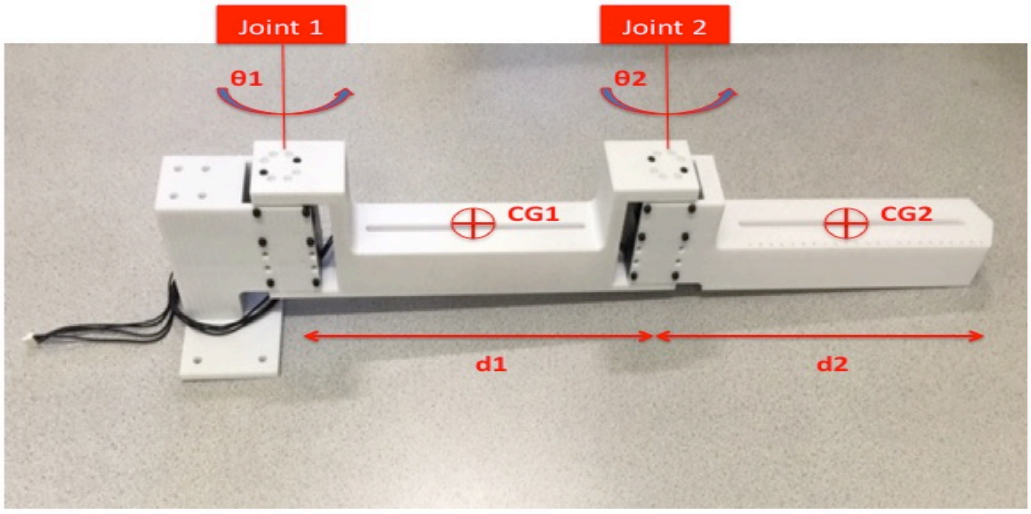

Figure 4. Prototype Robot Arm

The links were created in Solid Works and printed using the Naval Postgraduate School's (NPS) 3D printer of polycarbonate material. The joint actuators are Dynamixel Pro actuators; model M42-10-S260-R, from the company Robotis. Each actuator has a maximum torque rating of 5.6 Nm and a nominal load speed of 28 RPM [10]. The actuators are controlled by the host computer via Robotis Dynamixel Wizard software. In Dynamixel Wizard, multiple actuators can be commanded at once to set torque, velocity, and position. With these inputs, the user chooses the goal position, and the actuators perform the maneuver from point. Other operating modes are available (such as torque control); however, position control is the default and most appropriate mode to control the actuators. Important to note, there is not a mode that allows multiple commands at once, that is, a time series of inputs that would allow a robot arm to perform something other than a linear "point A to B" trajectory. To enable this type of commanding, custom software needs to be developed to drive the system. Figure 5 shows a screenshot of the Dynamixel Wizard user interface.

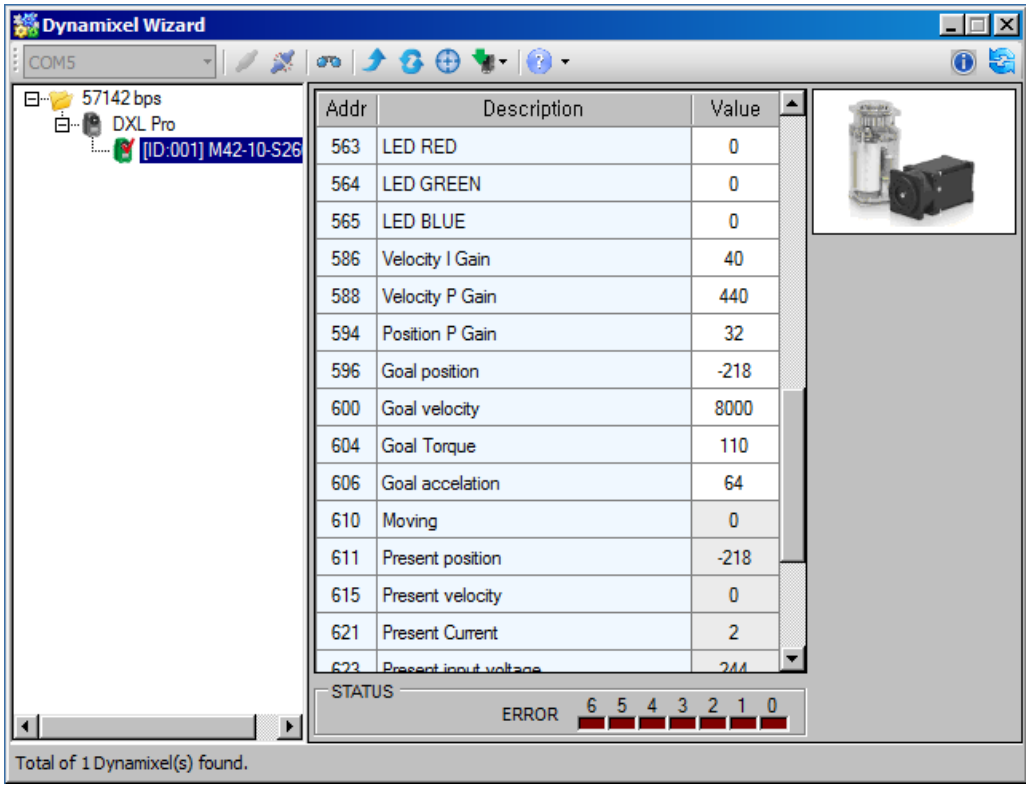

Figure 5. Dynamixel Wizard User Interface. Source: [10].

Measurements were made to determine the parameters of the robot. The length of the arms was measured to be 33 cm and each arm was estimated to be approximately 2.3 kilograms. The link parameters are summarized in Table 1.

The mass moment of inertia for each link is calculated for a simple beam as:

$$I_i = \frac{1}{12} * m_i * l_i^2 \tag{2.14}$$

where I_i denotes the moment of inertia of link *i* about an axis to the axis of joint rotation (see Figure 4) and passing through the center of mass of link *i*. Using the link measurements given above, *I* was estimated to $I = 0.021\ kg\text{-}m^2$.

Table 1. Estimated Link Parameters of a 2DOF Robot

	Length (m)	Mass (kg)	Inertia ($kg * m^2$)
Individual Link	0.33	2.3	0.021

C. TIME OPTIMAL CONTROL OF THE 2-DOF ARM

Optimal control can be applied for cases of minimum effort but the case of minimum time causes the largest disturbance to the spacecraft. This is because minimum time solutions are bang-bang in nature. Therefore, for the purposes of this thesis, time optimal control was considered. The problem formulation and subsequent derivation make use of the methods discussed in reference [11].

1. Problem Formulation

For the dynamic model described in section A, the state variables and control variables can be defined as:

$$x = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{bmatrix} \in \mathbb{R}^4 \qquad u = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \in \mathbb{R}^2 \tag{2.15}$$

Optimal control is applied for a minimum time problem by a cost functional as:

$$J\left[\bar{x}(\bullet),\bar{u}(\bullet),t_f\right]=t_f \tag{2.16}$$

The time optimal control problem uses to the dynamics in (2.1) and (2.2). To form the state-space model for each link's acceleration, (2.1) and (2.2) are solved for $\ddot{\theta}_1$ and $\ddot{\theta}_2$, respectively as:

$$\alpha_1=\frac{\tau_1}{m_{11}}-\frac{m_{12}}{m_{11}}\alpha_2+\frac{2r\omega_1\omega_2}{m_{11}}+\frac{r{\omega_2}^2}{m_{11}} \tag{2.17}$$

$$\alpha_2=\frac{\tau_2}{m_{21}}-\frac{m_{22}}{m_{21}}\alpha_1-\frac{r}{m_{21}}{\omega_1}^2 \tag{2.18}$$

where $\dot{\theta}_1=\omega_1$ and $\dot{\theta}_2=\omega_2$ and where $\ddot{\theta}_1=\dot{\omega}_1=\alpha_1$ and $\ddot{\theta}_2=\dot{\omega}_2=\alpha_2$.

By substituting (2.18) into (2.17), we arrive at:

$$\alpha_1=\frac{1}{m_{11}-m_{22}}\left(\tau_1-\tau_2+r\omega_1^2+2r\omega_1\omega_2+r\omega_2^2\right) \tag{2.19}$$

Similarly, by substituting (2.17) into (2.18), we arrive at:

$$\alpha_2=\frac{m_{11}}{m_{21}m_{11}-m_{22}m_{12}}\left(\tau_2-\frac{m_{22}}{m_{11}}\tau_1-\frac{m_{22}}{m_{11}}2r\omega_1\omega_2-\frac{m_{22}}{m_{11}}r\omega_2^2-r\omega_1^2\right) \tag{2.20}$$

2. Scaling the Dynamic Equations

Scaling is used to condition the problem numerically. Using the values assumed for link length and mass as a length unit and a mass unit, respectively, an inertia unit was determined as:

$$IU=MU*LU^2 \tag{2.21}$$

Since angular rate is a function of angle (referred to as a dimensionless unit of radians) an angular rate unit is derived as simply the angle displaced divided by time.

$$\omega U=\frac{\theta}{time}=\frac{\theta U}{TU} \tag{2.22}$$

Therefore, we may choose a time unit is used as simply unity. Consequently, a scaled torque unit simply becomes an *IU* divided by unity. Table 2 illustrates the canonical units derived from the robot arm problem.

Table 2. Canonical Units for a Robot Arm Problem

Length Unit (LU)	0.33 m
Mass Unit (MU)	2.3 kg
Inertia Unit (IU)	0.08 kg-m^2
Torque Unit (TQU)	0.08 Nm
Time Unit (TU)	1 sec
Theta Unit (θU)	θ rad

Consequently, the scaled parameters of the system become:

$$d_{1,2} = LU * \bar{d}_{1,2} \tag{2.23}$$

$$m_{1,2} = MU * \bar{m}_{1,2} \tag{2.24}$$

$$l_{1,2} = IU * \bar{I}_{1,2} \tag{2.25}$$

$$\tau_{1,2} = TQU * \bar{\tau}_{1,2} \tag{2.26}$$

$$\omega = \omega U * \bar{\omega} \tag{2.27}$$

where $\omega U = \dfrac{\theta U}{TU} = 1$.

The following illustrates the scaling of the dynamic equations:

For $\dot{\theta}_1$ and $\dot{\theta}_2$:

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{d\theta U\bar{\theta}}{dTUt} = \frac{\theta U}{TU}\frac{d\bar{\theta}}{d\bar{t}} = \frac{\theta U}{TU}\dot{\bar{\theta}} \tag{2.28}$$

$$\frac{\theta U}{TU}\dot{\bar{\theta}} = \omega U\bar{\omega} = \frac{\theta U}{TU}\bar{\omega} \tag{2.29}$$

$$\dot{\bar{\theta}} = \bar{\omega} \tag{2.30}$$

The LHS for $\dot{\omega}_1$ and $\dot{\omega}_2$is scaled as:

$$\ddot{\theta} = \dot{\omega} = \frac{d\omega}{dt} = \frac{d\omega U\bar{\omega}}{dTU\bar{t}} = \frac{\omega U}{TU}\frac{d\bar{\omega}}{d\bar{t}} = \frac{\omega U}{TU}\dot{\bar{\omega}} \tag{2.31}$$

The RHS for $\dot{\omega}_1 \; and \; \dot{\omega}_2$ is scaled as (note to reader, recall that terms m_{ij} are inertia terms, not mass terms):

$$\frac{\omega U}{TU}\dot{\bar{\omega}}_1 = \frac{1}{IU\bar{m}_{11} - IU\bar{m}_{22}}\begin{pmatrix} TQU\bar{\tau}_1 - TQU\bar{\tau}_2 \\ +MU\bar{m}_2 LU\bar{l}_1 LU\bar{l}_2 \sin\theta_2 \omega U^2 \bar{\omega}_1^2 \\ 2MU\bar{m}_2 LU\bar{l}_1 LU\bar{l}_2 LU\bar{l}_2 \sin\theta_2 \omega U^2 \bar{\omega}_1\bar{\omega}_2 \\ MU\bar{m}_2 LU\bar{l}_1 LU\bar{l}_2 LU\bar{l}_2 \sin\theta_2 \omega U^2 \bar{\omega}_2^2) \end{pmatrix} \tag{2.32}$$

The first term of (2.32) is scaled as:

$$\frac{TQU\bar{\tau}_1}{IU(\bar{m}_{11} - \bar{m}_{22})} = \frac{\frac{MULU^2}{TU^2}\bar{\tau}_1}{MULU^2(\bar{m}_{11} - \bar{m}_{22})} \tag{2.33}$$

Dividing through by $\frac{TU}{\omega U}$ gives:

$$\frac{\frac{MULU^2}{TU^2}\bar{\tau}_1}{MULU^2(\bar{m}_{11} - \bar{m}_{22})} \bullet \frac{TU}{\omega U} = \frac{\frac{MULU^2}{TU^2}\bar{\tau}_1}{MULU^2(\bar{m}_{11} - \bar{m}_{22})} \bullet \frac{TU}{\theta U \diagup TU} \tag{2.34}$$

All terms cancel leaving:

$$\frac{\bar{\tau}_1}{(\bar{m}_{11} - \bar{m}_{22})} \tag{2.35}$$

Similar results are derived for the second term of (2.32). For the third, fourth, and fifth terms of (2.32), the first term in the parenthetical expressions are considered only as the other terms in the parentheses have the same units. As well, as was done in the scaling of the first term of (2.32), the right hand side of (2.36) incorporates the division of $\frac{TU}{\omega U} = \frac{TU^2}{\theta U}$. With this said, (2.36) shows the scaling of the third term of (2.32):

$$\frac{TU^2}{\theta U}\left(\frac{1}{MULU^2(\bar{m}_{11} - \bar{m}_{22})}\right) MU\bar{m}_2 LU\bar{l}_1 LU\bar{l}_2 \sin\theta_1 \omega U^2 \bar{\omega}_1^2 \tag{2.36}$$

All terms cancel to give:

$$\frac{\bar{m}_2\overline{l_1 l_2}\sin\bar{\theta}_2}{(\bar{m}_{11}-\bar{m}_{22})}\omega_1^2=\frac{r\omega_1^2}{(\bar{m}_{11}-\bar{m}_{22})} \tag{2.37}$$

In all, the fourth and fifth terms are scaled in a similar fashion to arrive (finally) at the scaled equation for $\dot{\bar{\omega}}_1$:

$$\dot{\bar{\omega}}_1=\ddot{\bar{\theta}}_1=\frac{1}{\bar{m}_{11}-\bar{m}_{22}}(\bar{\tau}_1-\bar{\tau}_2+\bar{r}\bar{\omega}_1^2+2\bar{r}\bar{\omega}_1\bar{\omega}_2+\bar{r}\bar{\omega}_2^2) \tag{2.38}$$

Similarly for $\dot{\bar{\omega}}_2$:

$$\dot{\bar{\omega}}_2=\ddot{\bar{\theta}}_2=\frac{\bar{m}_{11}}{\bar{m}_{21}\bar{m}_{11}-\bar{m}_{22}\bar{m}_{12}}(\bar{\tau}_2-\frac{\bar{m}_{22}}{\bar{m}_{11}}\bar{\tau}_1-\frac{\bar{m}_{22}}{\bar{m}_{11}}2\bar{r}\bar{\omega}_1\bar{\omega}_2-\frac{\bar{m}_{22}}{\bar{m}_{11}}\bar{r}\bar{\omega}_2^2-\bar{r}\bar{\omega}_1^2) \tag{2.39}$$

where $\bar{r}$ is: $\bar{r}=\bar{m}_2\overline{l_1 l_2}\sin\bar{\theta}_2$.

The dynamic equations can now be written in their scaled form as:

$$x=\begin{bmatrix}\dot{\bar{\theta}}_1\\ \dot{\bar{\theta}}_2\\ \dot{\bar{\omega}}_1\\ \dot{\bar{\omega}}_2\end{bmatrix}=\begin{bmatrix}\dot{\bar{\theta}}_1=\bar{\omega}_1\\ \dot{\bar{\theta}}_2=\bar{\omega}_2\\ \dot{\bar{\omega}}_1=\bar{\alpha}_1=\frac{1}{\bar{m}_{11}-\bar{m}_{22}}\left(\bar{\tau}_1-\bar{\tau}_2+\bar{r}\bar{\omega}_1^2+2\bar{r}\bar{\omega}_1\bar{\omega}_2+\bar{r}\bar{\omega}_2^2\right)\\ \dot{\bar{\omega}}_2=\bar{\alpha}_1=\frac{\bar{m}_{11}}{\bar{m}_{21}\bar{m}_{11}-\bar{m}_{22}\bar{m}_{12}}\left(\bar{\tau}_2-\frac{\bar{m}_{22}}{\bar{m}_{11}}\bar{\tau}_1-\frac{\bar{m}_{22}}{\bar{m}_{11}}2\bar{r}\bar{\omega}_1\bar{\omega}_2-\frac{\bar{m}_{22}}{\bar{m}_{11}}\bar{r}\bar{\omega}_2^2-\bar{r}\bar{\omega}_1^2\right)\end{bmatrix} \tag{2.40}$$

3. Boundary Conditions

The boundary conditions for the rest-to-rest motion are modeled as zero critical angular velocity and zero final angular velocity for each link. The boundary conditions for initial position x-y to final position x-y use the model kinematics based off of link length and angular displacement of each link. Equations (2.41) – (2.49) illustrate these conditions.

$$\bar{\omega}_{10}=0 \tag{2.41}$$

$$\bar{\omega}_{20}=0 \tag{2.42}$$

$$\bar{\omega}_{1f}=0 \tag{2.43}$$

$$\bar{\omega}_{2f} = 0 \tag{2.44}$$

$$\bar{d}_1 \cos\bar{\theta}_{10} + \bar{d}_2 \cos(\bar{\theta}_{10} + \bar{\theta}_{20}) = \bar{x}_0 \tag{2.45}$$

$$\bar{d}_1 \sin\bar{\theta}_{10} + \bar{d}_2 \sin(\bar{\theta}_{10} + \bar{\theta}_{20}) = \bar{y}_0 \tag{2.46}$$

$$\bar{d}_1 \cos\bar{\theta}_{1f} + \bar{d}_2 \cos(\bar{\theta}_{1f} + \bar{\theta}_{2f}) = \bar{x}_f \tag{2.47}$$

$$\bar{d}_1 \sin\bar{\theta}_{1f} + \bar{d}_2 \sin(\bar{\theta}_{1f} + \bar{\theta}_{2f}) = \bar{y}_f \tag{2.48}$$

$$t_0 = 0 \tag{2.49}$$

The values of x_0, x_f, y_0 and y_f were established to establish a joint space trajectory moving through 90-degrees.

In order to solve the optimal control problem, Pontryagin's principle is applied as outlined in [11]. A control Hamiltonian is created by relating any running cost function and the co-state vectors with the state dynamics. The Hamiltonian is then minimized with respect to the control variable to arrive at a relationship between the control variable to the co-state dynamics. An intermediate step to develop adjoint equations is then applied to determine the co-state dynamics. The concept of transversality is applied by taking the partial derivative of the end point cost function and end-point equation with respect to the end states to derive relationships that will serve to aid in solving the adjoint equations. The Hamiltonian minimization condition uses these results to solve for the ultimate goal of the control trajectory.

4. Formulation of the Hamiltonian

In formulating the Hamiltonian $H = F + \lambda^T f$, F is the running cost, λ is a vector of co-states, and f is a vector of the state dynamics equations. For the case of the time optimal control problem, the running cost is zero and the end-point cost, E, is simply t_f. The Hamiltonian is therefore:

$$
\begin{aligned}
H = & \quad \lambda_{\theta_1}\bar{\omega}_1 + \lambda_{\theta_2}\bar{\omega}_2 \\
& + \lambda_{\omega_1}\left[\frac{1}{\bar{m}_{11}-\bar{m}_{22}}(\bar{\tau}_1 - \bar{\tau}_2 + \bar{r}\bar{\omega}_1^2 + 2\bar{r}\bar{\omega}_1\bar{\omega}_2 + \bar{r}\bar{\omega}_2^2)\right] \\
& + \lambda_{\omega_2}\left[\frac{\bar{m}_{11}}{\bar{m}_{21}\bar{m}_{11}-\bar{m}_{22}\bar{m}_{12}}(\bar{\tau}_2 - \frac{\bar{m}_{22}}{\bar{m}_{11}}\bar{\tau}_1 - \frac{\bar{m}_{22}}{\bar{m}_{11}}2\bar{r}\bar{\omega}_1\bar{\omega}_2 - \frac{\bar{m}_{22}}{\bar{m}_{11}}\bar{r}\bar{\omega}_2^2 - \bar{r}\bar{\omega}_1^2)\right]
\end{aligned}
\tag{2.50}
$$

By observation, the appearance of control torque in the Hamiltonian is linear. As a consequence, when the Hamiltonian is minimized with respect to each control variable, i.e. the partial $\frac{\partial H}{\partial \tau_i}$ is evaluated, the control torques disappear. Therefore, to obtain an expression for the control torques, the Hamiltonian minimization condition (HMC) is interpreted as a switching function [12]. The torque is defined by the behavior of $\frac{\partial H}{\partial \bar{u}_i} = S_i$. As noted earlier, the maximum torque that can be applied by the two-link arm actuators is 5.6 Nm. We now evaluate the switching functions.

For $\bar{\tau}_1$ from (2.15) we have:

$$
\frac{\partial H}{\partial \bar{\tau}_1} = \frac{\lambda_{\omega_1}}{\bar{m}_{11}-\bar{m}_{22}} - \lambda_{\omega_2}\left(\frac{\bar{m}_{22}}{\bar{m}_{21}\bar{m}_{11}-\bar{m}_{22}\bar{m}_{12}}\right) = S_i \tag{2.51}
$$

which specifies the torque as follows:

$$S_1 > 0: \quad \bar{\tau}_1 = \tau_{min}$$

$$S_1 < 0: \quad \bar{\tau}_1 = \tau_{max}$$

Similarly, for $\bar{\tau}_2$ from (2.15):

$$
\frac{\partial H}{\partial \bar{\tau}_2} = \frac{-\lambda_{\omega_1}}{\bar{m}_{11}-\bar{m}_{22}} + \frac{\lambda_{\omega_2}\bar{m}_{11}}{\bar{m}_{21}(\bar{m}_{11}-\bar{m}_{22})} = S_i \tag{2.52}
$$

giving:

$$S_2 > 0: \quad \bar{\tau}_2 = \tau_{min}$$

$$S_2 < 0: \quad \bar{\tau}_2 = \tau_{max}$$

5. Adjoint Equations

With the Hamiltonian minimized the adjoint equations must be created to define the co-state dynamics equations that allow the switching functions to be evaluated and additionally to gain insight to the satisfaction of the conditions of Pontryagin's principle. The adjoint equations are developed as:

$$-\dot{\vec{\lambda}} = \frac{\partial H}{\partial \vec{x}} \tag{2.53}$$

Considering variable θ_1, we have $-\dot{\lambda}_{\theta_1} = \frac{\partial H}{\partial \theta_1}$, which gives:

$$\frac{\partial H}{\partial \theta_1} = 0 \tag{2.54}$$

Since the equations of motion do not have terms as a function of θ_1; therefore, we arrive at the value above. From (2.54), we can conclude that:

$$\lambda_{\theta_1} = constant \tag{2.55}$$

What will be found later is that the constancy of the Hamiltonian [11] and the fact that $\lambda_{\theta_1} = constant$ will be the key indicators for optimality regarding the satisfaction of Pontryagin's principle.

Considering θ_2, the adjoint equation $-\dot{\lambda}_{\theta_2} = \frac{\partial H}{\partial \theta_2}$, gives many terms due to the fact that $\overline{r} = \overline{m}_2 \overline{l_1} \overline{l_2} \sin\overline{\theta}_2$, which appears in the equations of motion, and subsequently in the Hamiltonian. With this said, the equation for $-\dot{\lambda}_{\theta_2}$ becomes very complicated. The equation does not give intuition as to the behavior of the co-state trajectory; that is, if the plot were that of an optimal or non-optimal solution. Due to the complicated nature of the dynamics, similar complicated equations are found for λ_{ω_1} and λ_{ω_2}; therefore, those adjoint equations also don't give a useful way to verify the optimality of the system. Additionally, it was illustrated earlier that the switching functions also include the terms λ_{ω_1} and λ_{ω_2}. With this said, the one could compute the values of S_i to check that the controls perform in the correct manner.

6. Transversality

The pencil to paper formulation of transversality is conducted for thoroughness, but in this case was not needed since the end conditions are known. However, if transversality results are not considered with this observation (i.e., that derivation of the transversality conditions are not useful) then we know the something may be wrong with our problem formulation on our subsequent analysis.

The steps to developing the transversality conditions are shown as follows.

$$\overline{\lambda}(t_f) = \frac{\partial \overline{E}}{\partial \overline{x}_f} \tag{2.56}$$

where $\bar{E} = E\left(x\left(t_f\right)\right) + \nu^T e\left(x\left(t_f\right)\right)$. Equation $e(x(t_f))$ provides the end-point conditions.

Since for the minimum time problem we have:

$$e = \begin{bmatrix} E = t_f \\ \omega_{1_f} - 0 \\ \omega_{2_f} - 0 \\ d_1\cos\theta_{1f} + d_2\cos(\theta_{1f}+\theta_{2f}) - x_f \\ d_1\sin\theta_{1f} + d_2\sin(\theta_{1f}+\theta_{2f}) - y_f \end{bmatrix} \tag{2.57}$$

We may write:

$$E = \begin{matrix} t_f + \nu_1\omega_{1f} + \nu_2\omega_{2f} \\ +\nu_3(d_1\cos\theta_{1f} + d_2\cos(\theta_{1f}+\theta_{2f}) - x_f) \\ +\nu_4(d_1\sin\theta_{1f} + d_2\sin(\theta_{1f}+\theta_{2f}) - y_f) \end{matrix} \tag{2.58}$$

Taking the partial of $\bar{E}$ with respect to the state variables at the final time gives:

$$\lambda_{\theta_1}(t_f) = \frac{\partial \overline{E}}{\partial \theta_{1f}} = \begin{matrix} \nu_3(-d_1\sin\theta_{1f}\dot{\theta}_{1f} + d_2\sin(\theta_{1f}+\theta_{2f})\dot{\theta}_{1f}) \\ +\nu_4(d_1\cos\theta_{1f}\dot{\theta}_{1f} + d_2\cos(\theta_{1f}+\theta_{2f})\dot{\theta}_{1f}) \end{matrix} \tag{2.59}$$

$$\lambda_{\theta 2}(t_f) = \frac{\partial \overline{E}}{\partial \theta_{2f}} = \nu_3(-d_2\sin(\theta_{1f}+\theta_{2f})\dot{\theta}_{2f}) + \nu_4(d_2\cos(\theta_{1f}+\theta_{2f})\dot{\theta}_{2f}) \tag{2.60}$$

$$\lambda_{\omega_1}(t_f) = \frac{\partial \bar{E}}{\partial \omega_{1f}} = \nu_1 \tag{2.61}$$

$$\lambda_{\omega_2}(t_f) = \frac{\partial \bar{E}}{\partial \omega_{2f}} = \nu_2 \tag{2.62}$$

As is seen in section 6, each partial derivative provides an unknown constant. Thus, the analysis does not provide any insight that can be used to validating an optimal solution. Moreover, since the transversality conditions provide no new information, we may conclude that the boundary conditions for our problem have been properly specified.

7. Solving the Optimal Control Problem

To perform the optimal control computation, the program DIDO was utilized [11], [13]. The model was coded to allow a trajectory that spanned 360 degrees; that is, there were no path constraints. With this said, the scenario illustrated in the following discussion addresses the arm moving from a position on the x-y plane of (2, 0) to (0, 2) where the distance traveled is in length units (LU). Figure 6 illustrates the trajectory.

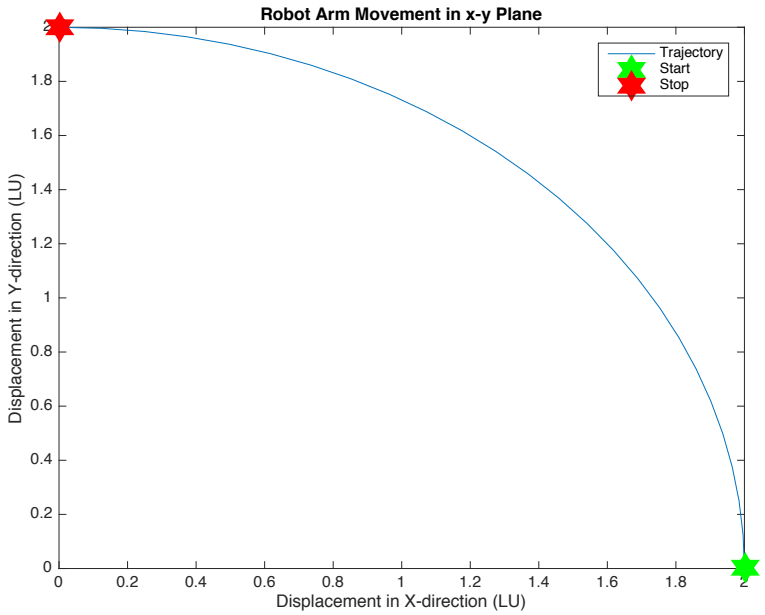

Figure 6. Robot Arm Movement in the x-y Plane

Additionally, the model was run using 20 nodes. The values gained from this simulation were then used as a "guess" for a simulation using 60 nodes. Using 60 nodes allows one to arrive at a more refined solution.

8. Results

Figures 7–11 represent the minimum time solution obtained from DIDO for the scenario depicted in Figure 6. Figure 7 shows that the maximum torque that could be applied to the model is saturated to 5.6 N-m. This saturation value is consistent with the bounding value entered into the DIDO code. It turns out that if a higher value of torque was allowed in the bounds, a non-optimal solution or non-feasible solution was returned. As noted in the boundary conditions section of this chapter, the problem needs reasonable bounds for DIDO to narrow its search for the optimal solution. If the range of torques, in this case, is too high, there are too many solutions and therefore an optimal solution can't be determined. This infeasibility is true in real life as well. Large amounts of torque would cause high stresses on system components leading to increased failure rates. Large torques could also lead to large amounts of jitter when attempting to stop an antenna slew therefore affecting pointing accuracy. Thus, realistic bounds must always be defined to ensure a reasonable solution.

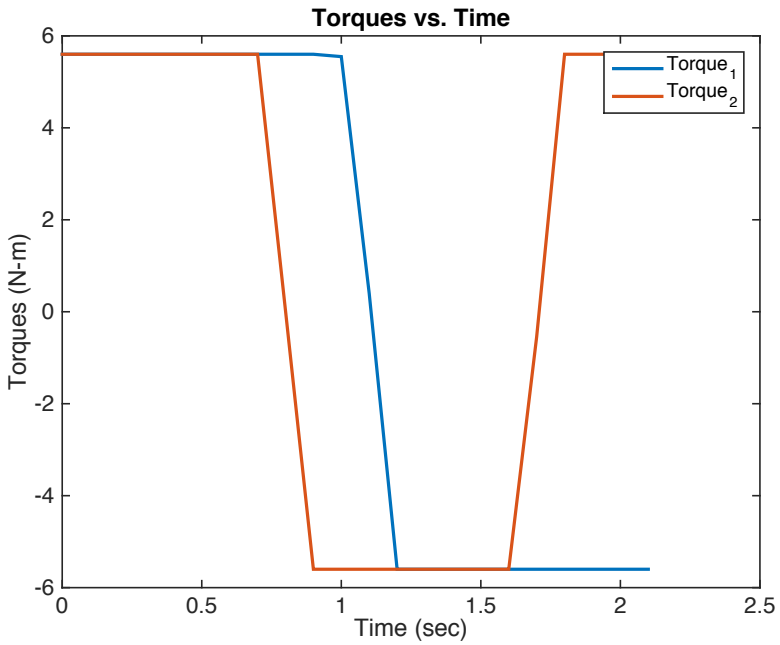

Figure 7. Minimum Time Torques versus Time for Trajectory (x,y) = (2,0) to (x,y) = (0, 2)

Figure 8 shows the state dynamics of the optimal control solution overlaid onto the validation and verification (V&V) solution. The V&V solution was obtained by running the trajectory obtained from DIDO through a numerical model [10]. For the V&V, initial values of the states were entered into an ode45 numerical integrator along with a new time vector that used smaller incremental time deltas compared to the nodal time vector determined by DIDO. This assisted in the interpolation of the DIDO time vector. Additionally, the V&V uses the optimal torque vectors (that is the control parameters) and propagates them through the state dynamics defined in the problem statement. A comparison of DIDO state trajectories and the V&V state trajectories can then be seen. Referring to Figure 8, one can see that the model indeed performs in a rest-to-rest manner. The variations between the DIDO solution and the V&V could be reduced with an increased number of nodes, or by using a feedback controller to implement the optimal joint trajectories.

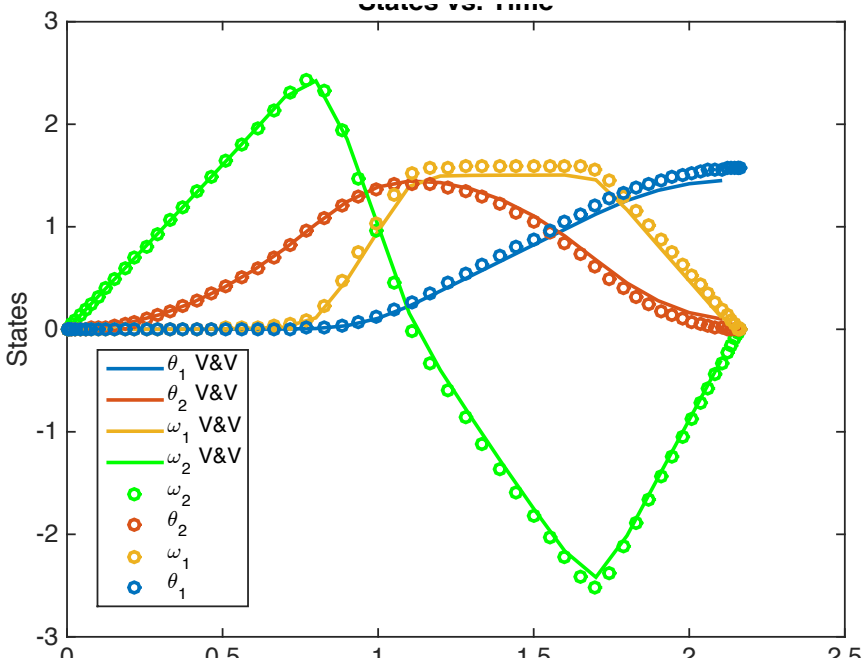

Figure 8. Minimum Time State Trajectories for (x,y) = (2, 0) to (x,y) = (0, 2)

One curious observation is the angular displacement, θ_1, and angular velocity, ω_1 of link one. From Figure 8, one can see that link one does not begin to move until approximately 0.75 seconds. This demonstrates an advantage of using computers to solve optimal control problems. The obvious solution where one would assume the links should always be moving is not always apparent, but with advanced computing power, unique solutions that could not otherwise be determined are discovered.

Figure 9 shows the co-states and that λ_{θ_1} is constant over the simulation time. This constancy is an indicator of optimality in accordance with Pontryagin's principle.

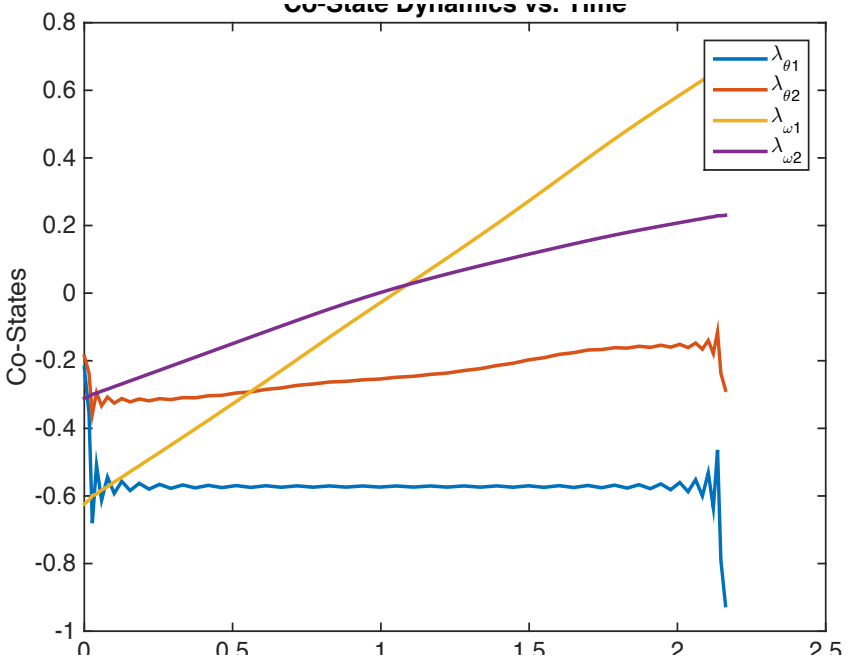

Figure 9. Minimum Time Co-states Trajectories for (x,y) = (2, 0) to (x,y) = (0, 2)

Lastly, Figure 10 illustrates a Hamiltonian that is relatively constant near a value of 'H = -1' as would be expected for a minimum time problem. The jump mid-way through the simulation indicates a non-ideal portion of the solution that typically is observed when a switch occurs between the nodes.

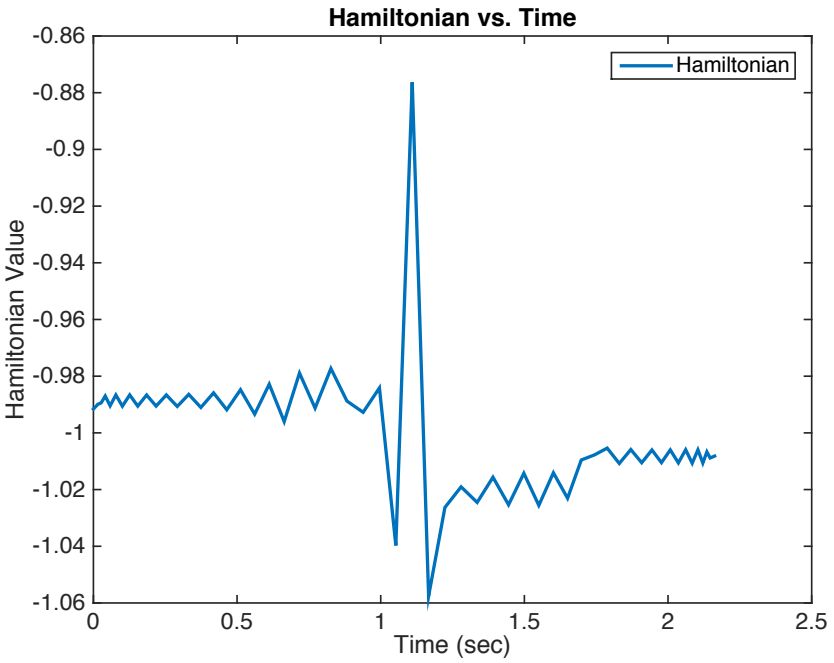

Figure 10. Minimum Time Hamiltonian for Trajectory (x,y) = (2,0 to (x,y) = (0, 2)

Figure 11 shows the optimal trajectory overlaid onto the V&V solution in the robot task space; i.e., the Cartesian plane. It is interesting to note that the trajectory between the start and end points is not an arc, as one would anticipate. From this plot, it is also apparent that a simple propagation of the open-loop controls doesn't move the robot's end-effect to exactly the right spot. Closing the loop (as is done in practice) or re-solving the optimal control problem with more stringent tolerances (e.g., a higher number of nodes) would resolve this issue.

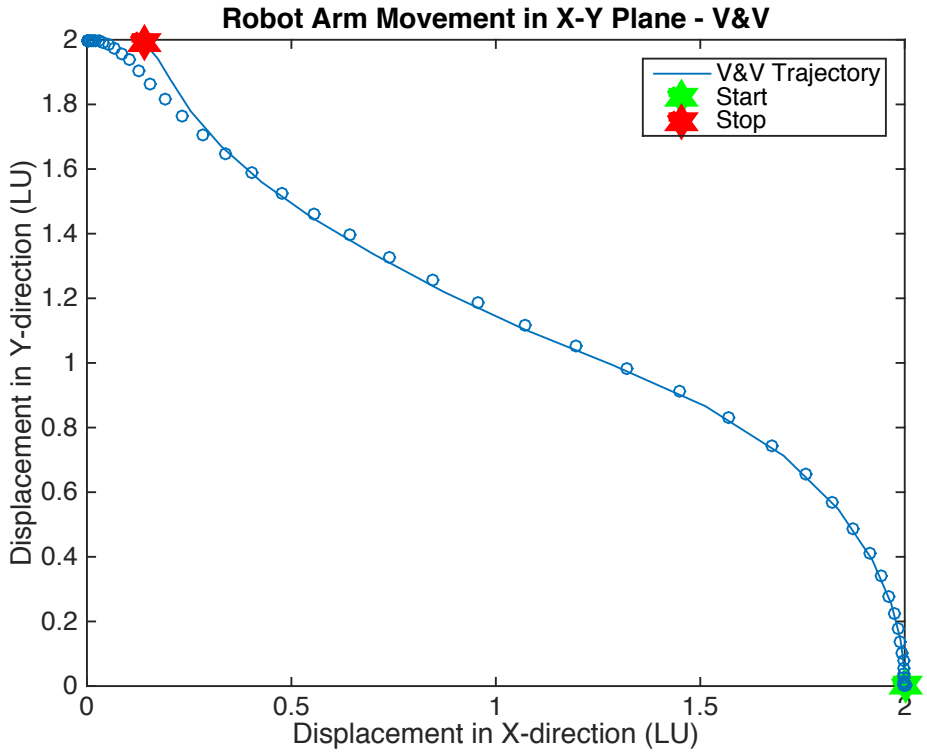

Figure 11. Minimum Time Robot Trajectory in the Task Space for Movement from (x,y) = (2,0) to (x,y) = (0,2)

D. SUMMARY

Through formulation of a time optimal slew for a 2DOF, planar robot arm, Chapter II established a solution around which to model the development of the experimental test bed. The optimal trajectory that was determined in this section is an example of the power of computers to find a better solution than what our intuition might tell us and as such is not exactly the same as the standard maneuver that industry would perform. That is, the time optimal path is not simply a great circle rotation, but in fact a more elegant path with varying torques at varying time steps to produce a better solution. With this said, the model presented in this chapter does not account for vibration analysis, but simply provides a basis to see how an optimal path performs as compared to a basic rotation of a two link robot arm. As stated in the Introduction to this thesis, one method to reduce vibrations is to reduce the applied torque and therefore the slew time for a given trajectory. With this said, the time optimal code was run a second time but with a torque input to each actuator of 2.8 Nm, or half of the 5.6 Nm maximum available torque from the Dynamixel Pro actuators. The resulting time to complete the maneuver was 3.0 seconds vice 2.1 seconds when 5.6 Nm was applied. Thus, vibrations can be reduced at the expense of an increase in maneuver time. In Chapter III, the results of this chapter are used to design a flexible joint with the consideration of the maximum torque. For comparison of other techniques aimed at minimizing vibrations, the reduced torque results will also be analyzed.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DEVELOPMENT OF A FLEXIBLE JOINT MECHANISM

Chapter III focuses on the development of the experimental system mechanical components, namely a flexible joint for the vibrating appendage. To form a basic understanding of the requirements for the flexible joint concepts, the chapter begins with a discussion of the test bed's air-bearing and base plate. From there, a simple model is developed to determine appropriate torsional spring rates required to produce a desired deflection. Chapter III continues by detailing the various flexible joint concepts that were tried before arriving at the concept used to build the final test bed configuration.

A. AIR-BEARING TEST BED

With the flexible joints developed for each case, simple simulations were conducted to (i) develop the process to analyze data from the test bed sensors, and (ii) to get a sense of the response that could be expected from the free-rotating system when excited from another robot arm's motion.

To begin, a rotary air bearing from Nelson Air Corporation was used to mount the support plate [14]. Figure 12 shows the air bearing. An air bearing provides for a near frictionless rotation in the horizontal plane. The rotary bearing originally incorporated a large magnet in the internal housing to motorize the motion; however, this was removed in order to ensure the top disk could rotate freely as in a gravity-free environment. A port with tubing is seen in Figure 12 illustrating how pneumatic pressure is applied to the test set. When air pressure is applied, between 4.1 – 4.8 kPa (60 – 70 psi) as specified in the test specification data sheet [14], the top disk rotates freely on a film of air between the two air bearing surfaces.

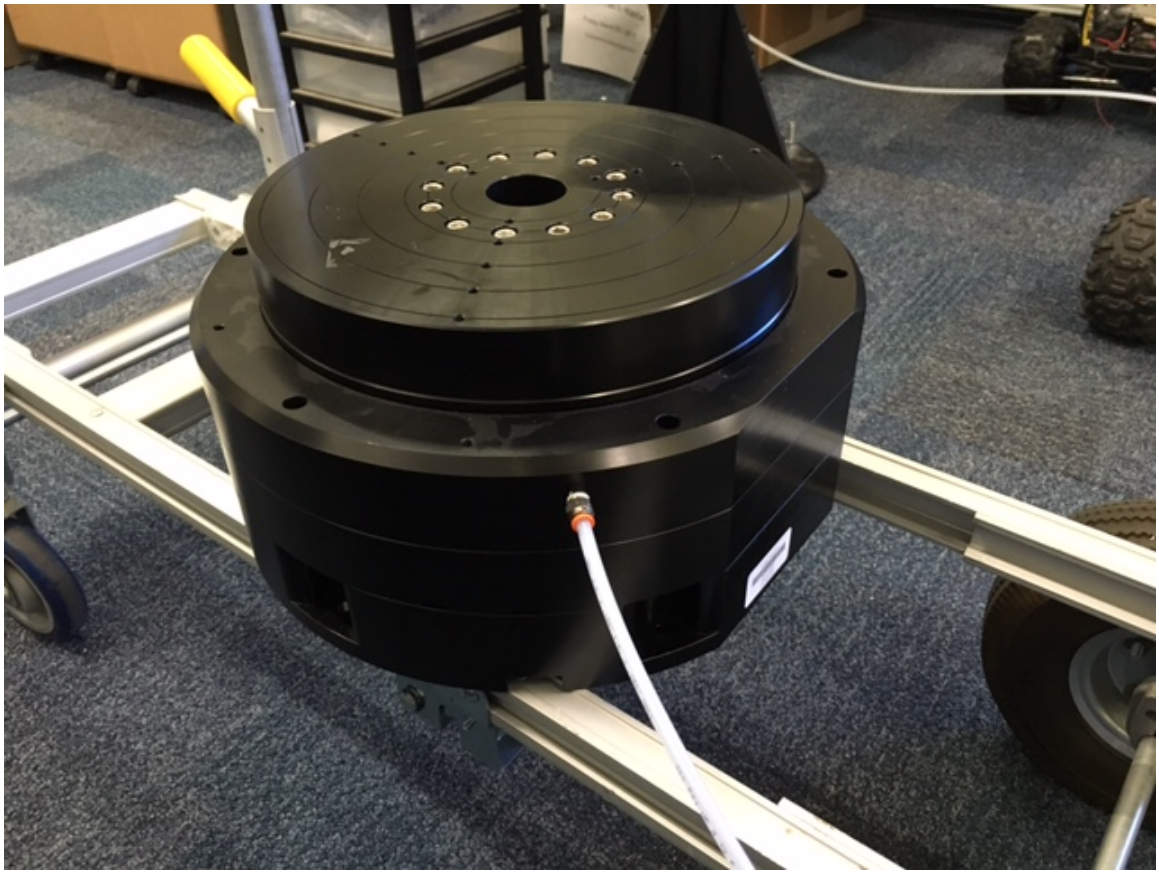

Figure 12. Single Axis Air-Bearing Test Set

B. BASE PLATE DEVELOPMENT

An aluminum plate was constructed to mount on top of the air bearing platter. The aluminum plate was designed to accommodate robotic linkages on opposite sides of the plate as well as to provide tapped holes to allow a means for integrating future modifications and experiments. The maximum overall loading of the air bearing is specified at 454 kg. The plate diameter design was based off of the air-bearing moment loading specification of 1730 kg-cm. From this specification, a plot was created to determine how far the test bed center of gravity could be from the center of the plate versus the amount of load that can be applied at the center of gravity. Figure 13 shows the results.

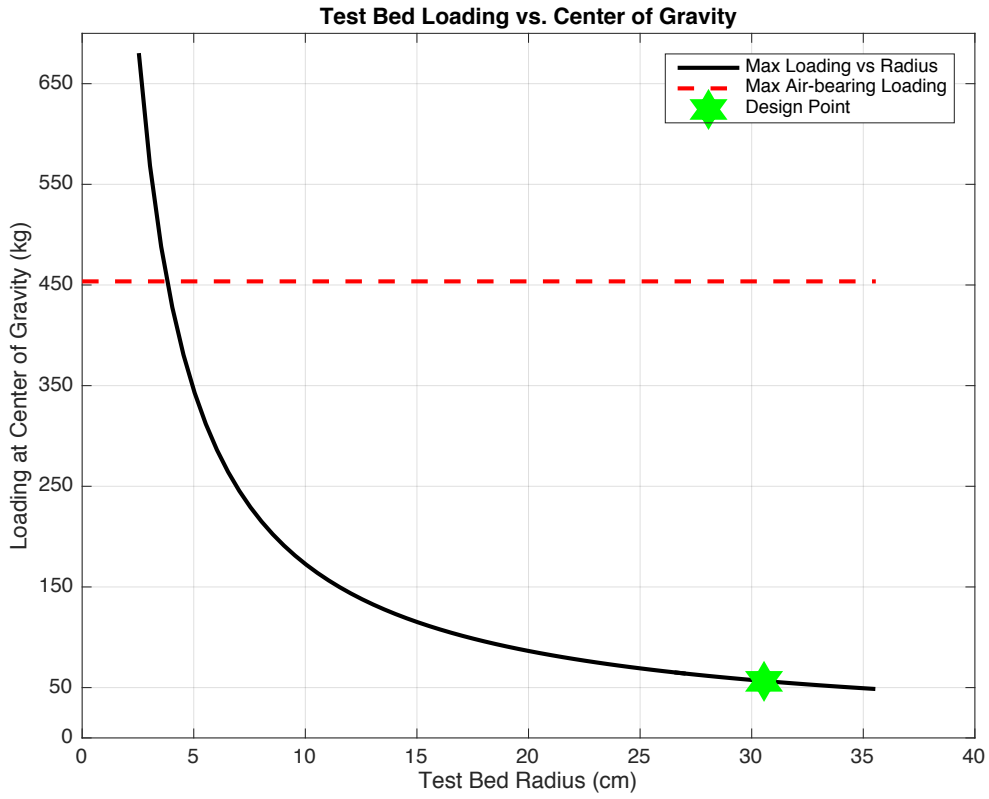

Figure 13. Test Bed Capacity versus Base Plate Radius

Figure 13 illustrates that if the total system center of gravity is 30.5 cm from the center of the air-bearing, up to 56.6 kg could be applied at that center of gravity before the resulting moment would cause the plate to be out of balance, and therefore, an interference in the free-floating rotation of the plate. The total weight of the current system is approximately 31 kg and the center of gravity is within 1 cm of the center of the plate. Therefore, even if the system center of gravity were more than 30.5 cm from the center of the plate, there would not be an imbalance. In the end, a 30.5 cm plate radius was chosen due to two reasons: the plot shown in Figure 13 and the amount of space that the system would encompass in the laboratory. Furthermore, Figure 13 shows that there is a significant margin between the current mass and the maximum bearing load to add a second level incorporating additional components.

Figure 14 illustrates a top view of the aluminum base plate when mounted on the air-bearing test set.

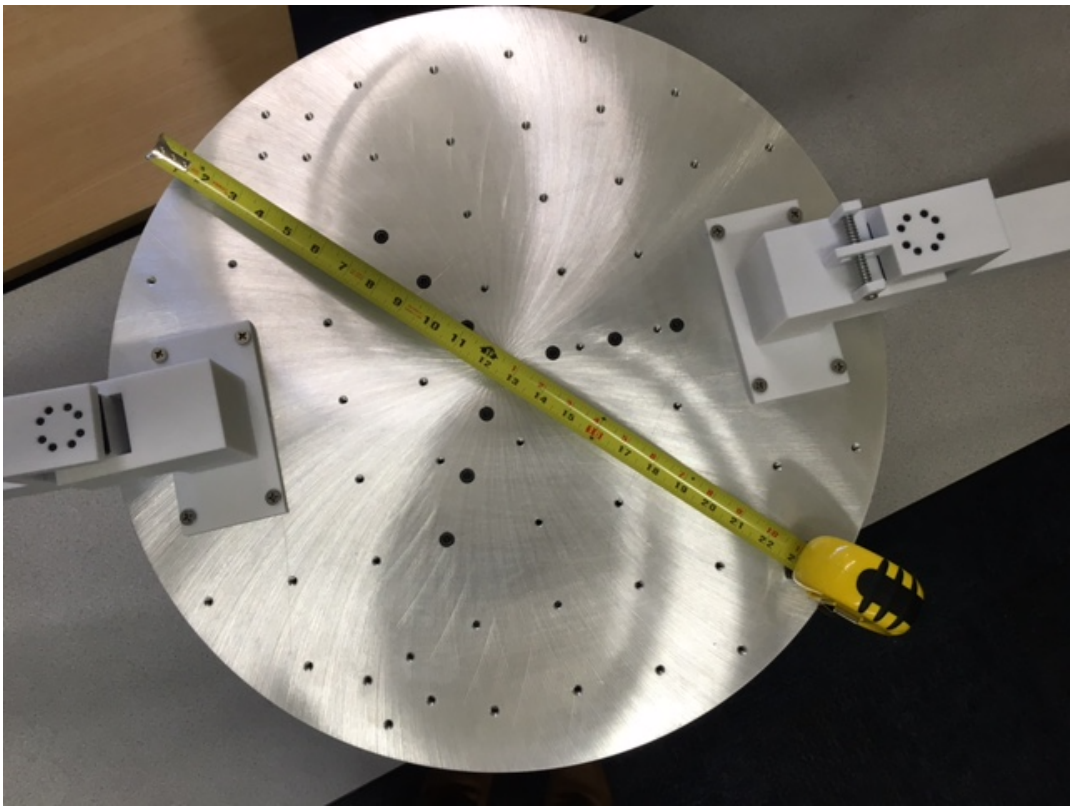

Figure 14. Aluminum Plate that Supports Robot Arm and Flexible Appendage

C. MODELLING APPENDAGE VIBRATIONS

The estimated mass-properties as outlined in Table 1 were used to determine the dynamics of the robot arm. Calculations were based on a simple beam model and do not account for actual design intricacies. The same geometric model was used to determine an appropriate value for a torsional spring rate so a one-arm link, with the flexible joint, excited by the two-link arm with electrically actuated joints would produce an easy to visualize vibration of approximately 19 degrees +/- 5 degrees. For the initial cut, a constant torque was applied to get a sense of how the system may perform. Subsequently, the optimal control profile was run to evaluate how that trajectory excites the vibration. The main point behind the large range of motion is to visually show the motion of the one-arm link flex. Figure 15 illustrates the simple torsional mass-spring model formed by a rigid plate attached to a spring that can be used to estimate the vibration amplitude.

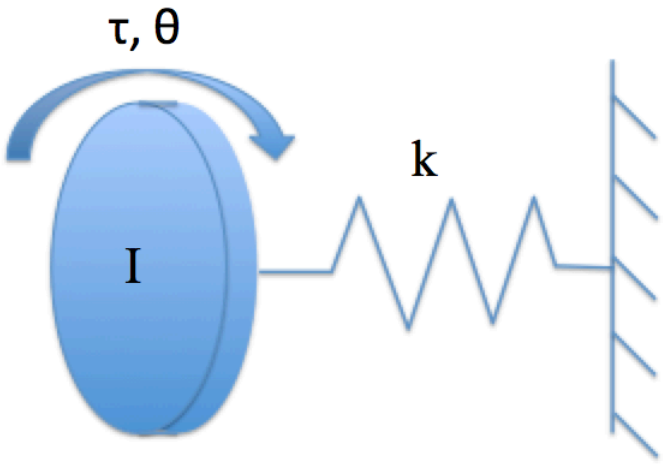

Figure 15. Simple Torsional Mass-Spring System

The equation of motion is simply derived as:

$$\tau = I\ddot{\theta} + k\theta \tag{3.1}$$

and, transformed into the Laplace domain becomes:

$$\tau(s) = (Is^2 + k)\theta(s) \tag{3.2}$$

The open-loop transfer function from input to output is:

$$G(s) = \frac{\theta(s)}{F(s)} = \frac{1}{Is^2 + k} \tag{3.3}$$

A Simulink implementation of this model is illustrated in Figure 16.

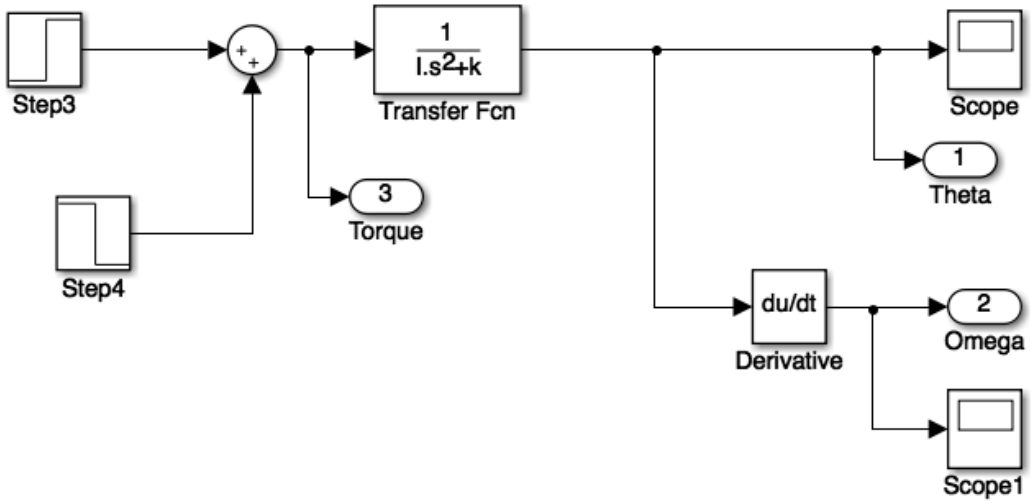

Figure 16. Simulink Model for Simple Mass-Spring System

In this model a constant input of 5.6 Nm was applied to system for 2.1 seconds then the applied torque is commanded to zero. The time frame of 2.1 seconds is the optimal time determined in Chapter II for the robotic arm maneuver. By turning the torque off after 2.1 seconds, the system is allowed to settle. For the initial analysis, each simulation was run for 15 seconds. The torque value of 5.6 Nm is the maximum torque capable of the Dynamixel Pro actuator. Figure 17 shows the response of a flexible link with k = 3.39 *deg/Nm* and I = 0.21 *kg-m*^{2}. The torsion spring rate was determined by simply dividing the desired deflection by the applied torque.

$$k = \frac{19\ deg}{5.6\ Nm} = 3.39\ deg/Nm$$

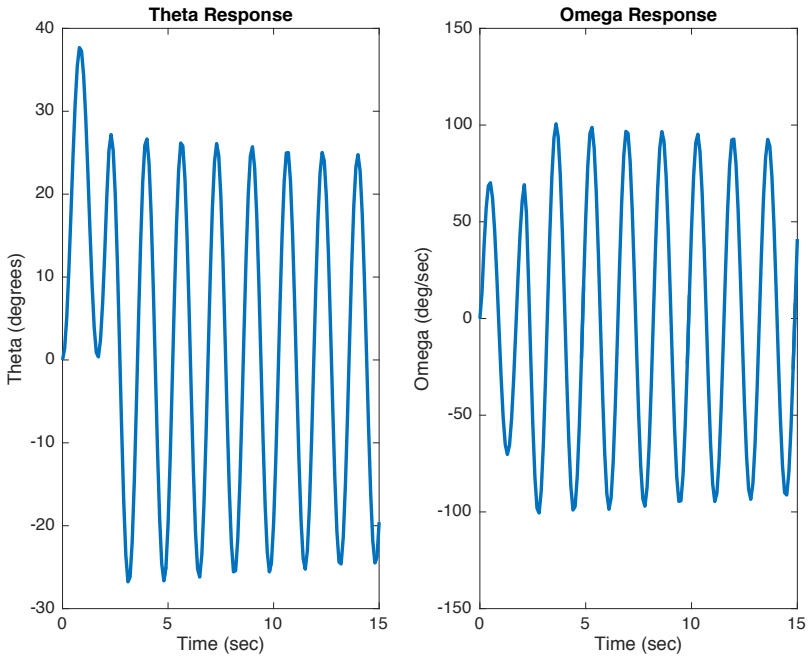

Figure 17. Angle and Rate Response of Flexible System Model when Subjected to 5.6 Nm of Torque Using Torsion Spring Rate of 3.39 deg/Nm

Referring to Figure 17, one can clearly see that for the constant input used, the approximated displacement would be over 35 degrees. This response would be too large compared to what was desired. Moreover, only one actuator on the two-link arm was considered in Figure 17 so the flexible motion would be much larger if the time optimal solution was implemented.

As a comparison to other techniques used in industry to minimize vibrations of flexible space systems, the optimal control model presented in Chapter II was run with an applied torque of half of the maximum torque available from the Dynamixel Pro actuators. The time to complete the maneuver of the planar, 2DOF robot arm increases from 2.1 seconds to 3.0 seconds with torque reduced from 5.6 Nm to 2.8 Nm. The reduced torque was entered into the same model used to produce Figure 17. Figure 18 illustrates the results.

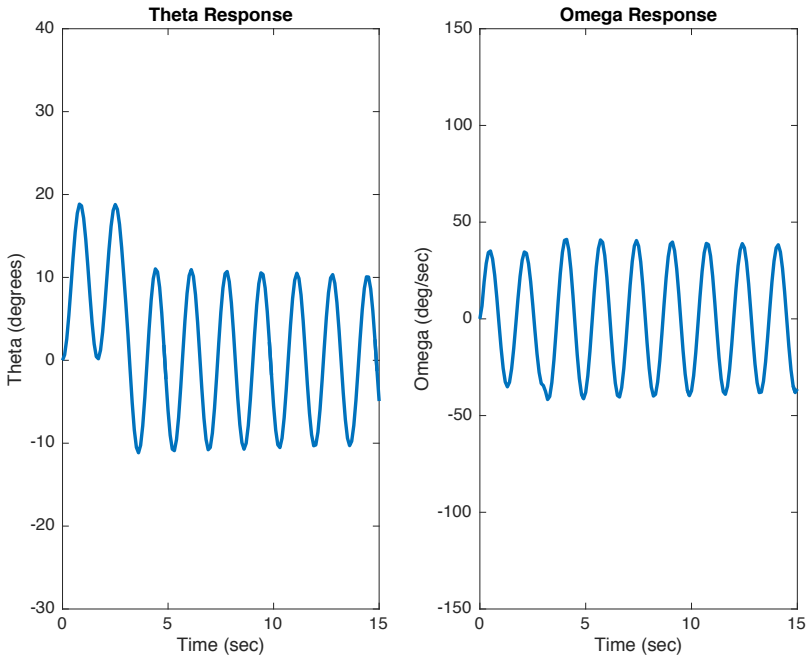

Figure 18. Response of Flexible System Model with Torque Reduced to 2.8 Nm

By comparing Figures 17 and 18, one can see that the torque reduction (with its unavoidable increase in slew time) does reduce the vibration displacement as well as the magnitude of angular rate. This technique is satisfactory to reduce vibration, but for our interests, faster slew times coupled with minimized vibrations are desired.

To find an appropriate spring rate for the flexible joint, the time optimal control trajectory developed earlier was approximated using three steps. The torque of each actuator from the time-optimal control trajectory was added together to give the approximated optimal control torque profile illustrated in Figure 19.

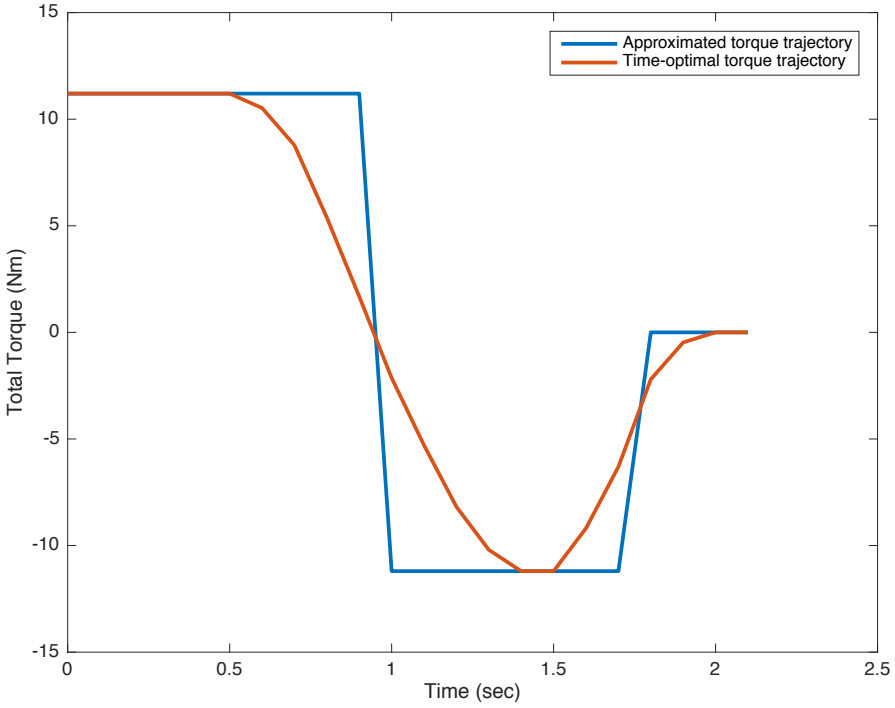

Figure 19. Approximate Torque Trajectory Compared to Total Time Optimal Control Torque Trajectory

This step-wise approximation can be seen to be a sufficient model of the system and to allow spring rates to be adjusted to obtain the desired deflection. Figure 20 illustrates the results for a torsion spring rate of 1.7 deg/Nm. The torsion spring rate was determined by simply dividing the desired deflection, 19 deg, by the combined applied torque of two actuators, 11.2 Nm.

$$k = \frac{19\ deg}{11.2\ Nm} = 1.7\ deg/Nm$$

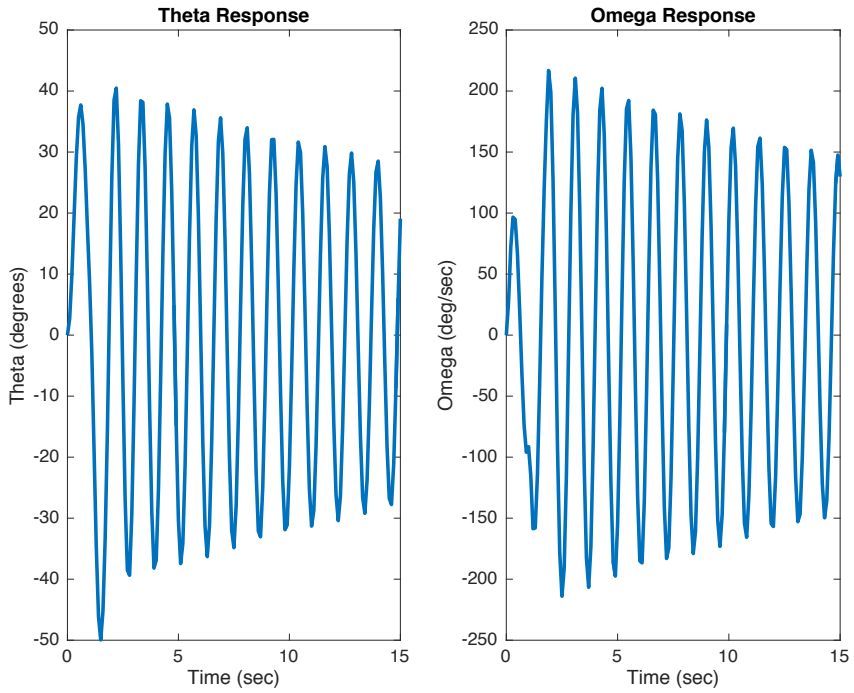

Figure 20. Angle and Rate Response of Flexible System Model when Subjected to the Torque Profile of Figure 19 Using a Torsion Spring Rate of 1.7 deg/Nm

The overall angle response in Figure 20 still was larger than required: over 35 degrees. The model was run again, halving the 1.7 deg/Nm spring rate to 0.85 deg/Nm. Figure 21 illustrates the results.

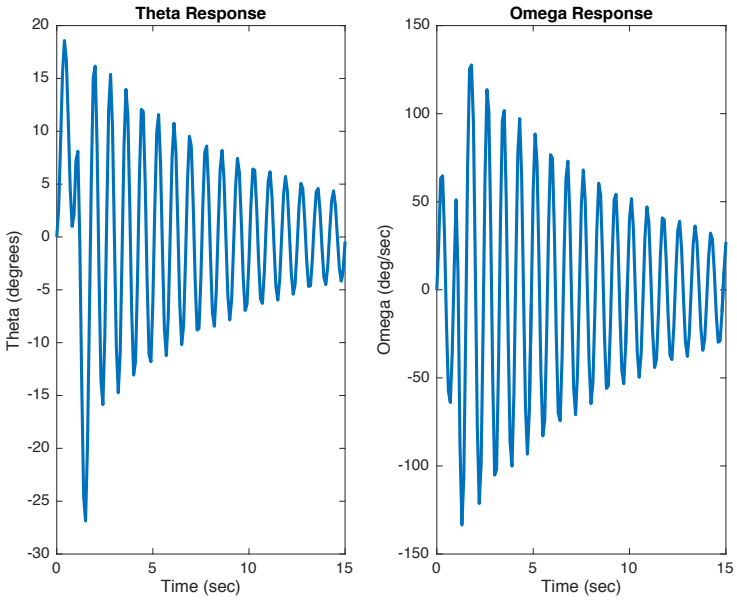

Figure 21. Angle and Rate Response of Flexible System when Subjected to the Torque Profile of Figure 19 Using a Torsion Spring Rate of 0.85 deg/Nm

In Figure 21 one can see that the desired initial response of approximately 19 degrees is achieved. The model does show the angle trajectory displacing over 25 degrees in the opposite direction. However, the simple model of the system with the spring rate of 0.85 deg/Nm was considered sufficient to proceed with the joint design and sizing of various options for the flexible link. These options are described in section D.

D. CONCEPT A—JOINT WITH TWO COUNTER SPRINGS

Concept A was inspired by a YouTube video describing an experiment using a flexible joint [15]. The experiment in the video uses control methods to minimize the vibrations of an appendage. Exact methods to control the appendage vibrations were not described, but the experimental set up provided an initial concept from which to design a flexible joint. Figure 22 is a simple sketch of the set up.

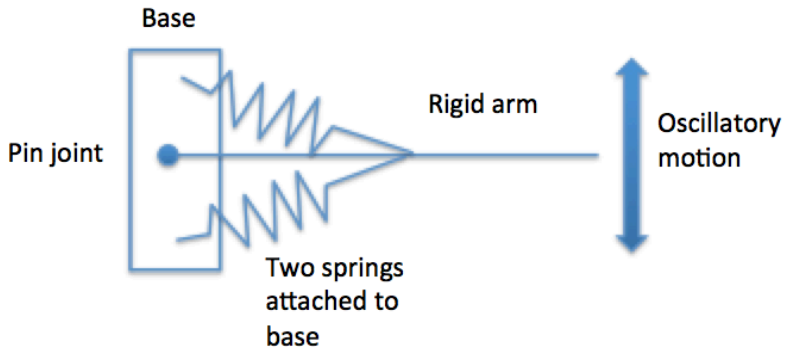

Figure 22. Simple Sketch Illustrating Flexible Joint for Concept A

This concept was developed as one possible means to evaluate the response imparted on the one-link arm from the two-link electrically actuated arm. A key feature to this design, and the two other concepts described in sections E and F, is this use of bearings as part of the joint. The bearings shown in Figure 23 are typically used with the actuators that are implemented on the two-link robot arm to allow appendage rotation. The repurposing of these bearings allows for smooth rotation to the flexible joint concepts [16].

Figure 23. Photograph of Dynamixel Pro Bearing Used for Flexible Joint Concept Designs. Source: [16].

It should be noted that, though the bearing does rotate freely, there is still friction in its mechanism. The ball bearings that allow the bearing to rotate may become damaged or dirt may become stuck in the rotating component introducing non-linearities to the system. At the very least, they introduce drag that will reduce the amplitude of the vibration over time. Figures 24 and 25 illustrate the developed prototype of concept A.

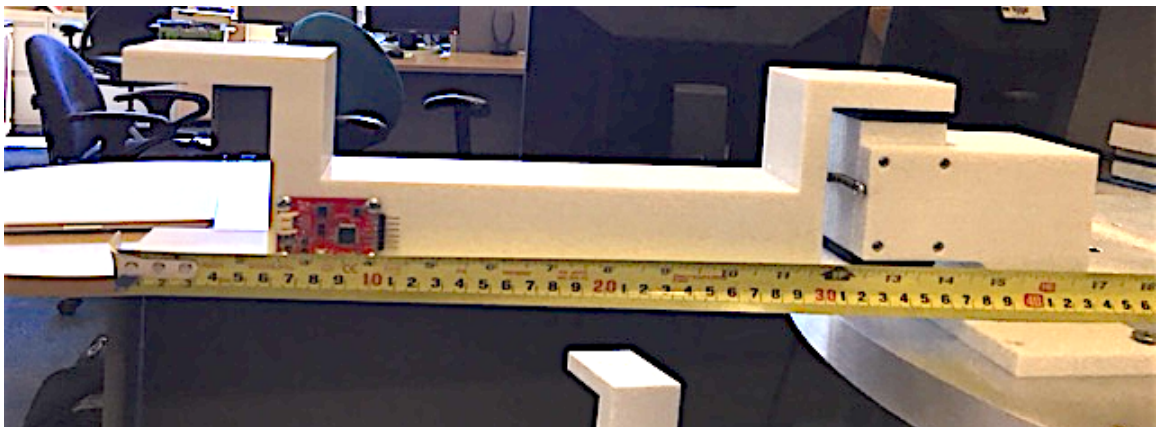

Figure 24. Concept A—Side View

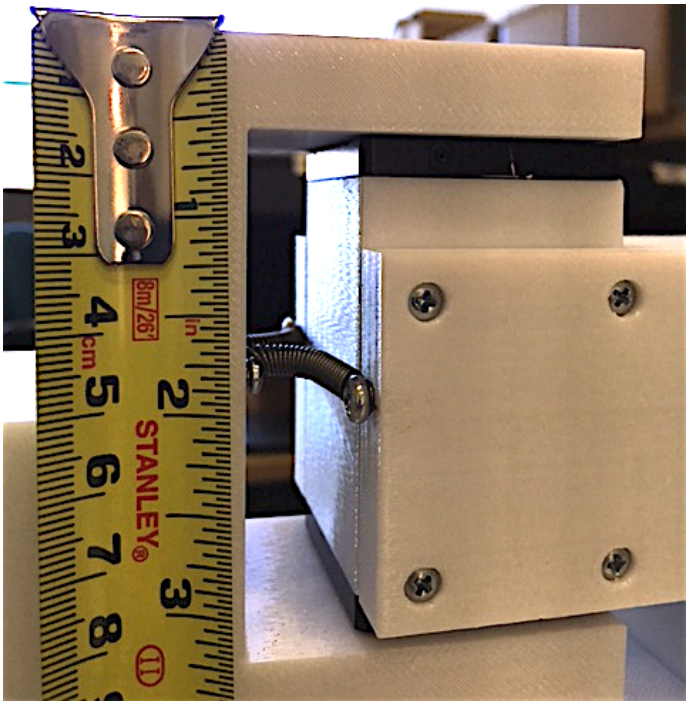

Figure 25. Concept A—Side View Close-up

Concept A seemed to be simple enough in the sketching phase. The main idea was to attach two extension springs on the outboard sides of the mount and arm. The concept may be simple on paper, but the actual build proved to be more difficult. As one can see from Figure 25, in order to change the spring to investigate varying spring constants and therefore different responses, the mount and arm have to be taken apart. This is a time consuming process. As well, in early experiments to test the basic functioning of the concept, the springs were observed to buckle in compression; that is, become concave with respect to the outside edge of the mount and arm. When the springs buckled in such a fashion, it was assumed that non-idealities would influence the response. As will be discussed later, the actual response appeared more linear than expected. Table 3 lists the springs that were utilized for initial experimentation.

Table 3. Springs Considered for Concept A

Length (cm)	Outside diameter (cm)	Wire diameter (cm)	Spring rate (kg/m)
1	0.478	0.0406	8.95
1	0.478	0.0508	29.2
1	0.478	0.0635	98.5

To get a sense of the response using the spring rates noted in Table 3, the torsional mass spring system was adjusted to account for the fact that the extension springs are being used. With this said, the force transmitted across the plate replaced the torque term by simply dividing the torque trajectory by the distance between the bases of the two-link arm and the one-link flexible joint arm. As well, the inertia term was replaced by mass. Figure 26 illustrates the response for a spring rate of 29.2 kg/m with a maximum deflection of 3 cm.

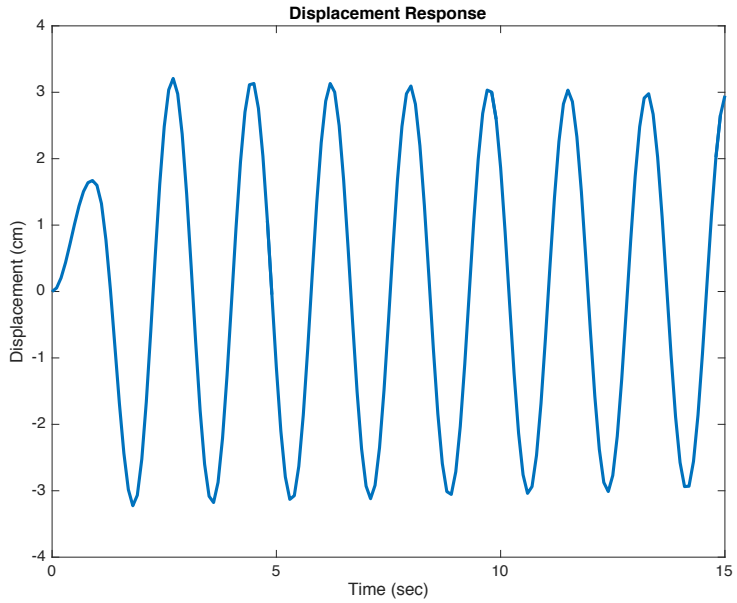

Figure 26. Simulated Response of Concept A Using a Spring Rate of 29.2 kg/m

The amount of linear deflection corresponds to approximately +/-5.2 deg when the deflection is considered at the end of the 33 cm vibrating arm as shown is Figure 27. The displacement in Figure 26 corresponded visually to initial experimentation of the system leading to an initial conclusion that the simple model provided a good approximation of the actual system response to robotic arm movement.

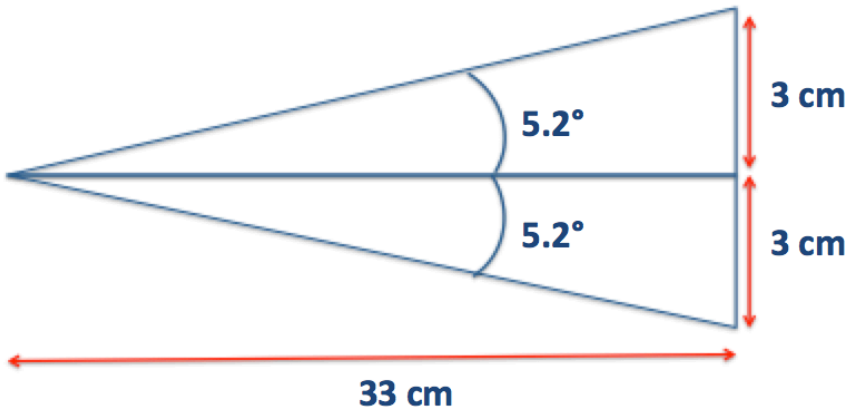

Figure 27. Displacement of Vibrating Arm

Due to the difficulty of assembling concept A, only one spring set was tested. In the end, the main purpose of this prototype was to provide an initial test of the concept. Although it did not produce a large amount of deflection when perturbed, it proved to be useful in the overall laboratory build as will be seen later.

E. CONCEPT B—SPRING FLEXION PIN

Concept B built upon the idea of concept A as a more rigid, interchangeable design. For this concept, a pin is routed through posts on both sides of the mount and in the middle through a tongue formed via 3D modeling into the arm. Grooves were created on the inside portion of the post and both sides of the tongue to provide a base for springs to rest, and therefore not slip and rub against the pin during the experiment. It should be noted; rubbing of the spring against the pin could not be avoided since in compression, the spring will want to naturally bend. With this said, the bending is far less then what is allowed for in concept A and should help to achieve a more linear response. Figures 28–31 illustrate the prototype of the improved setup.

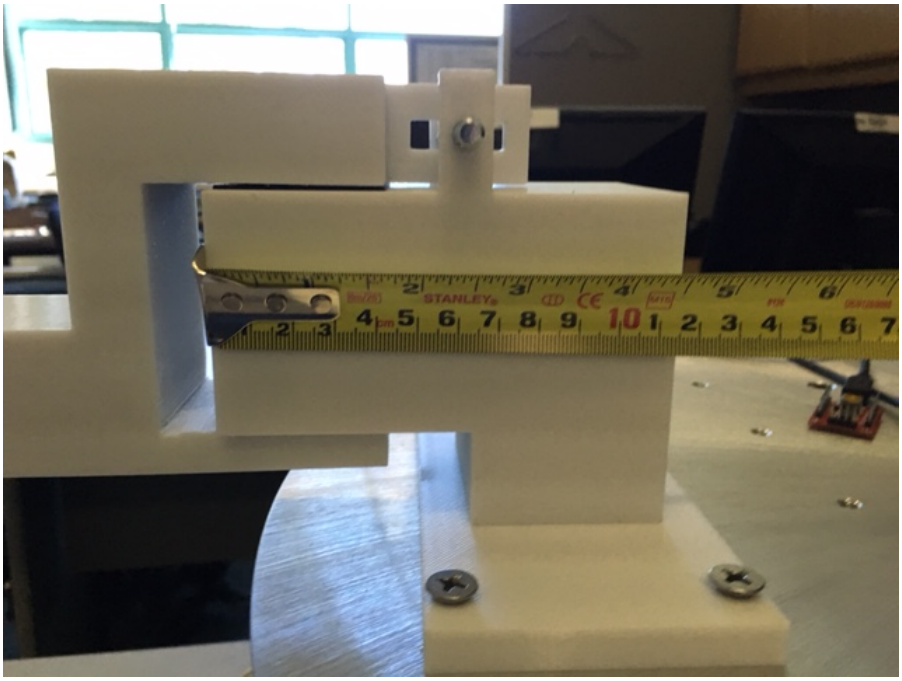

Figure 28. Concept B—Side View

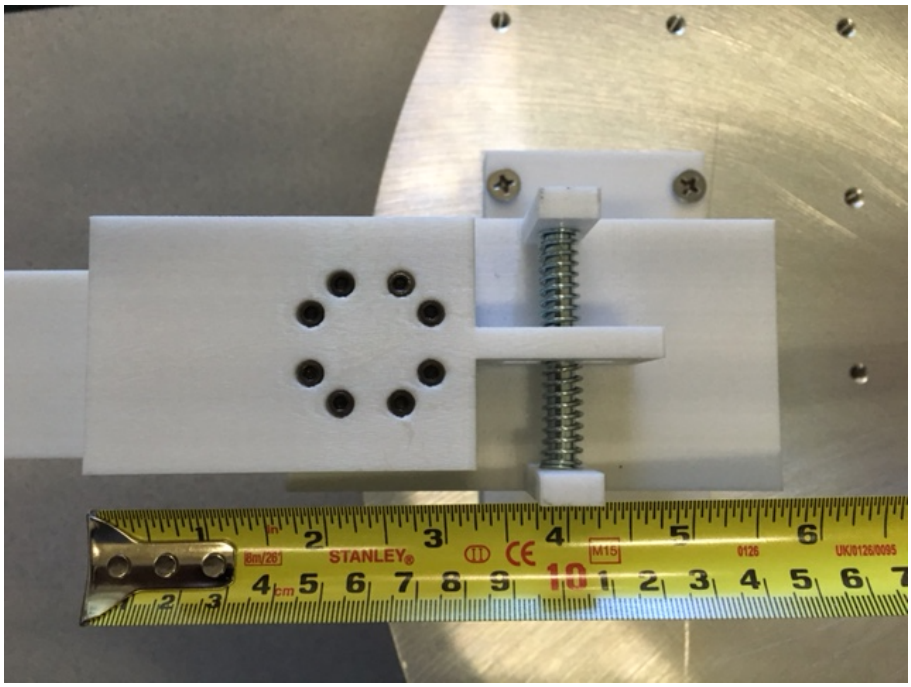

Figure 29. Concept B—Top View

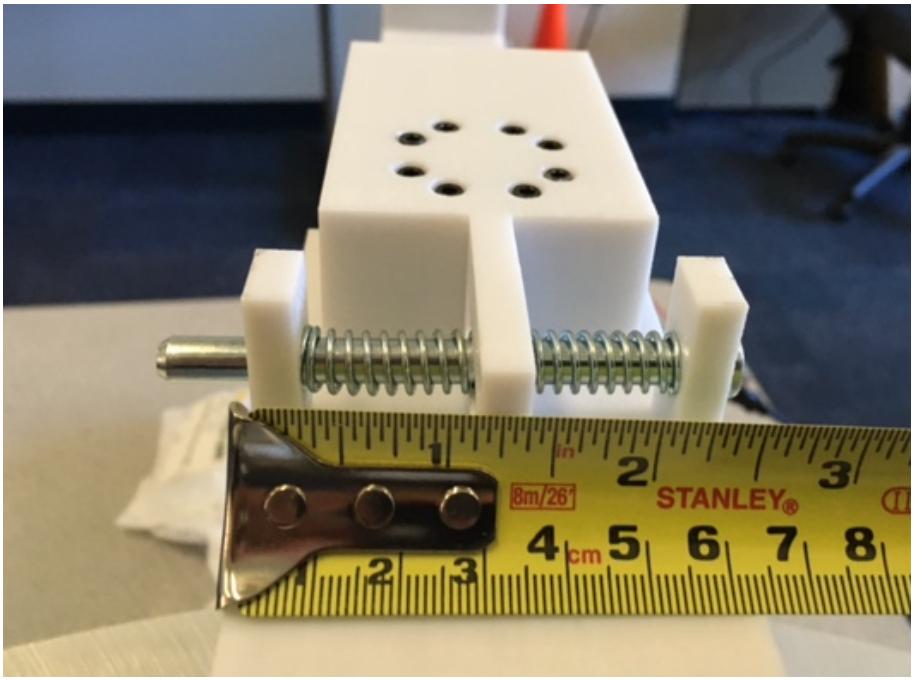

Figure 30. Concept B—Forward-Looking View

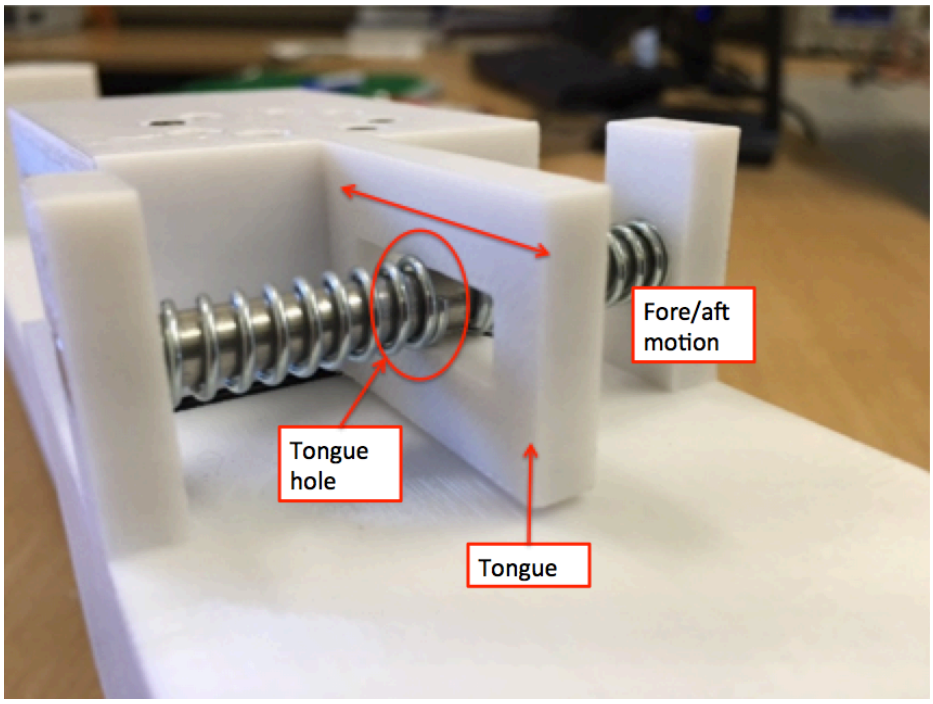

Figure 31. Concept B—Pin/Spring Arrangement Close-up

One key factor in the design of the tongue (see Figure 31) was to provide enough fore/aft room for the arm to displace under excitation and not hit the pin. To ensure this, a MATLAB script of basic trigonometry was coded to design the proper dimensions of the tongue hole. In this code, the goal is to ensure the hole is large enough so the outside edge of the tongue does not hit the pin when the arm is maximally displaced. The wider the tongue, the better chance that the tongue would impact the pin; therefore, the tongue was designed to be 1.27 cm in width. If the tongue were any thinner there was concern that the stresses from the spring compression may break the tongue or at best, bend it. The other piece of the design in considering hole size is the pin size. The pin diameter was chosen to be 0.64 cm based on the size of the height of the tongue coupled with spring's size that could fit well and provide an appropriate response. Considering this, the smaller the hole, the less allowance for displacement. Table 4 illustrates the point. The fore/aft displacement must be less than the hole diameter minus the pin diameter. In Table 4 this value is shown as the margin available for each hole diameter. The red values indicate when the end displacement will cause the margin for a particular hole diameter and the pin will be exceeded.

Table 4. Total Displacement of Clearance Hole (cm) for Concept B

	Varying Hole Diameter					
	0.76 cm	**1.012 cm**	**1.27 cm**	**1.52 cm**	**1.78 cm**	**2.03 cm**
Margin	0.064	0.19	0.32	0.45	0.57	0.70
End Displacement						
2.54 cm	0.046	0.046	0.048	0.048	0.048	0.048
5.08 cm	**0.130**	0.137	0.140	0.142	0.145	0.150
7.62 cm	**0.267**	**0.274**	0.279	0.287	0.292	0.30
10.2 cm	**0.442**	**0.455**	**0.465**	**0.478**	0.488	0.50
12.70 cm	**0.660**	**0.678**	**0.696**	**0.716**	**0.732**	**0.752**
15.24 cm	**0.922**	**0.947**	**0.975**	**1.001**	**1.026**	**1.054**

The shadowing in Table 4 illustrates for the end of the link to displace at least 10.2 cm (which equates to approximately 19 degrees) the hole diameter would need to be at least 1.78 cm in diameter (refer to the shaded row and column of Table 5). Considering the tongue was designed to be 2.54 cm in height, this was determined to be too large of a hole due to the strength of the material being used. That is, considering the amount of force the spring exerts on the tongue, by removing more material to create a bigger hole reduces the tongues ability to not bend or break during an experiment. In the end, the hole in the tongue was sized to be rectangular; 0.762 cm in height to accommodate the 0.635 cm pin diameter and 2.54 cm wide to allow for travel of the tongue without chance of interference with the pin. Table 5 lists the springs that were utilized for experimentation with the concept B prototype.

Table 5. Springs Considered for Concept B

Length (cm)	Outside diameter (cm)	Wire diameter (cm)	Spring rate (kg/m)
3.175	0.762	0.0813	154.0
3.175	1.156	0.991	146.8
3.175	0.914	0.104	248.8

As in concept A, to get a sense of the response of the spring rates noted in Table 6, the torsional mass spring system parameters of inertia and torque were replaced with mass and force, respectively, to account for the fact that extension springs are used. Figure 32 illustrates the response for a spring rate of 154.0 kg/m with a maximum deflection of just under 0.3 cm, which is less than a degree of deflection for the 33 cm vibrating arm. As in concept A, the displacement in Figure 32 corresponded visually to initial experimentation. Therefore, to use this design concept much lighter springs would need to be used.

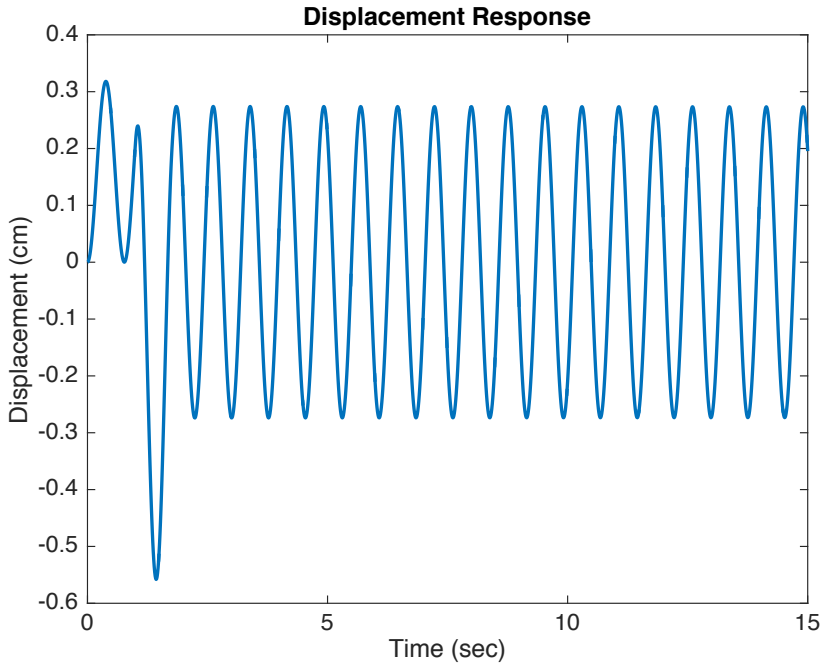

Figure 32. Displacement Modeling of Concept B Model with a Spring Rate of 154.0 kg/m

Again, the displacement from the model was far from the desired deflection of 19 degrees, but as will be seen in the next chapter, the motion induced on this link on the laboratory test-bed is sufficient to provide an understanding of the dynamics and vibration effects associated with spacecraft robot arm slews.

F. CONCEPT C—MACHINED TORSION SPRING

The concept of using a machined spring for the flexible joint was inspired by the website of Helical Products Co. based out of Santa Maria, California [16]. Helical specializes in engineering helical machined springs to produce desired amounts of torque and flexion. An advantage is that attachment points are (can be) integrated into the spring for mounting. These attachment points allow for simple assembly of the torsion spring joints by allowing one end to be fixed while the other end is mounted to a bearing or some device that allows the spring to be torqued. This concept enables a pure moment and linear rates to be transmitted to the vibrating arm. As such, this concept may prove to be the best option for a lab demonstrator. Figure 33 shows an example of the Helical product.

Figure 33. Example of a Helical Machined Spring. Source: [16], [17].

The full construction of concept C was delayed due to spring manufacturing lead-time and logistics; therefore, experimentation using this concept was delayed as well. With that said, initial parts for the concept C joint were created using the 3D printer and the prototype can be seen in Figures 34–36. Figure 37 illustrates the assembled design as created in the computer aided drawing program, Nx.

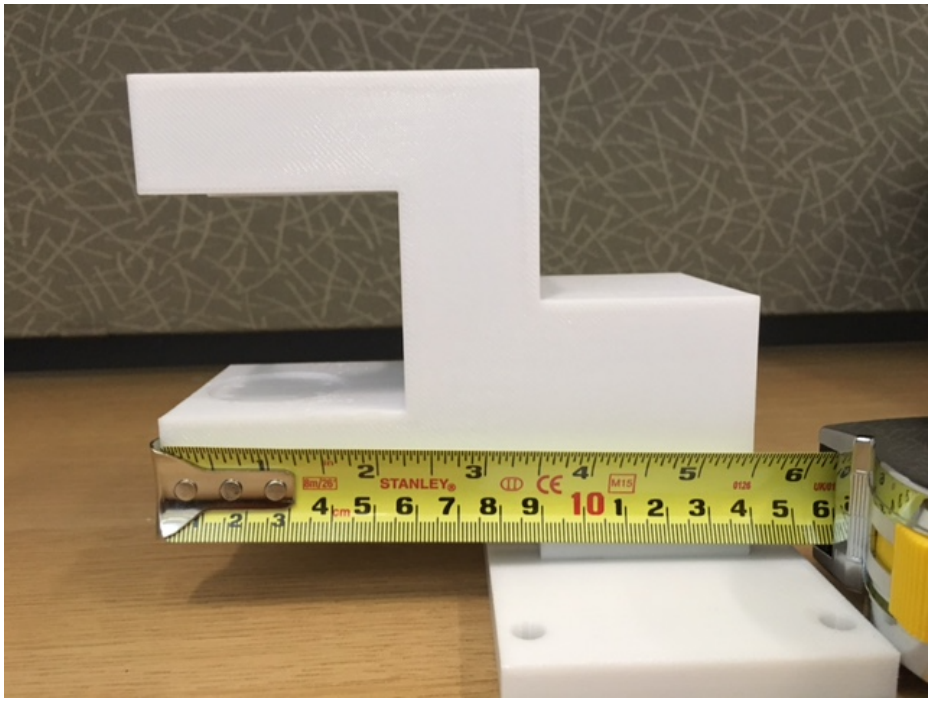

Figure 34. Concept C—Side View

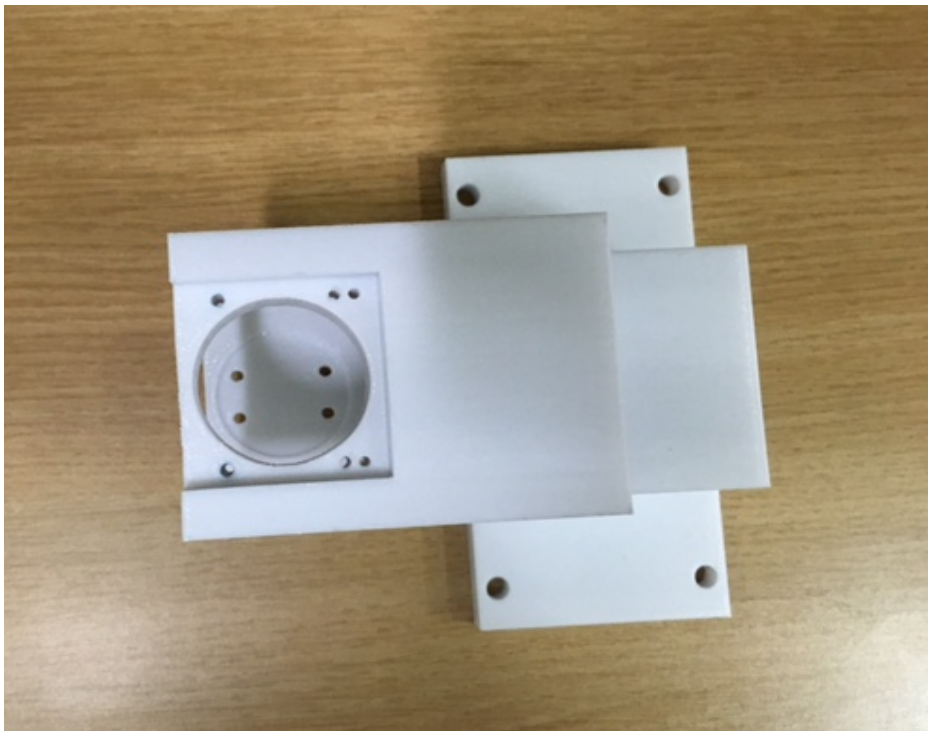

Figure 35. Concept C—Top View

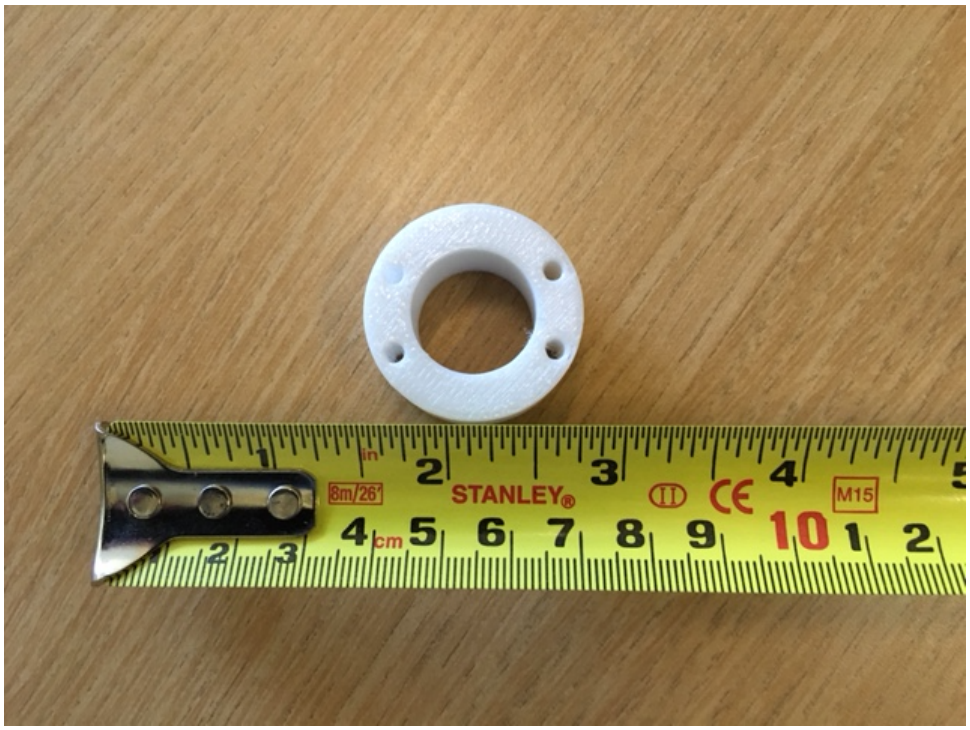

Figure 36. Concept C—Bearing Insert

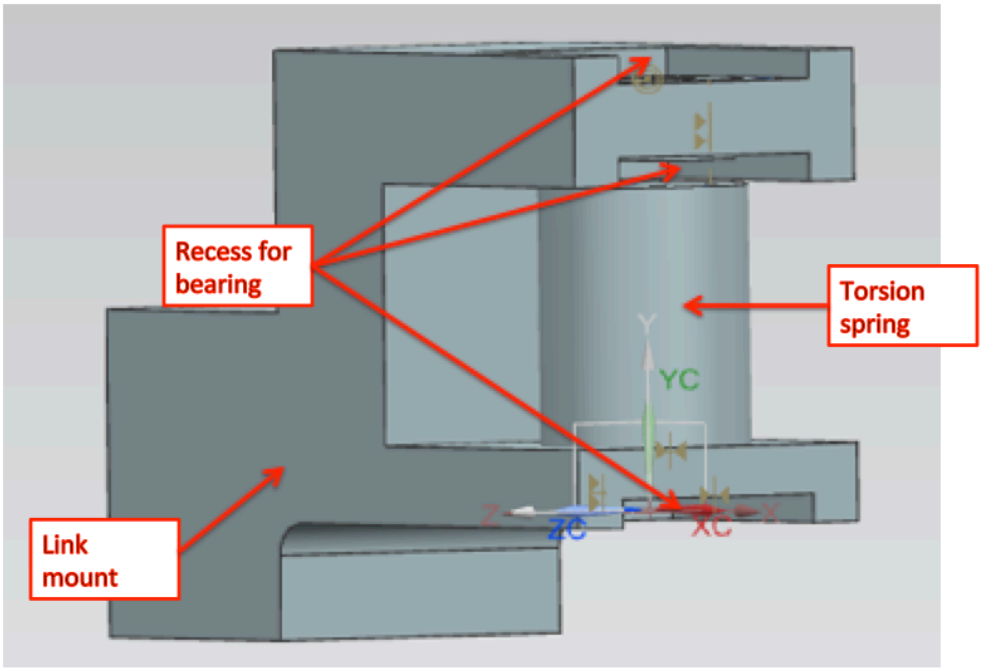

Figure 37. Concept C—Nx Model Front View of Link Mount with Simple Model of Machined Spring Inserted at the Joint

The parameters in Table 6 were provided to Helical engineers to provide a base estimate for performance capabilities and cost.

Table 6. Concept C Torsion Spring Requirements

Moment arm	30 cm
Total torque applied to two-link arm	11.2 Nm
Mass of one-link arm	11 kg
Deflection required	+/-10 cm
Spring Length	5.08 cm
Outside diameter	3.56 cm

When the design materialized, the torsional spring rate provided from Helical was 3.39 deg/Nm. The spring rate simply was derived by dividing the required deflection, in degrees, divided by the torque applied from one Dynamixel actuator. Figures 38 and 39 show a side and top view of the finished helical machined spring.

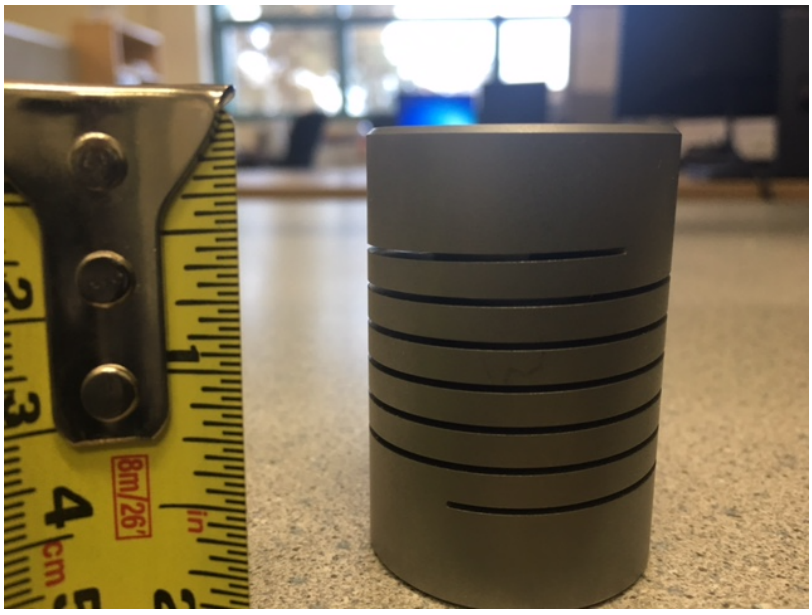

Figure 38. Side View of Helical Machined Spring Manufactured for Test Bed

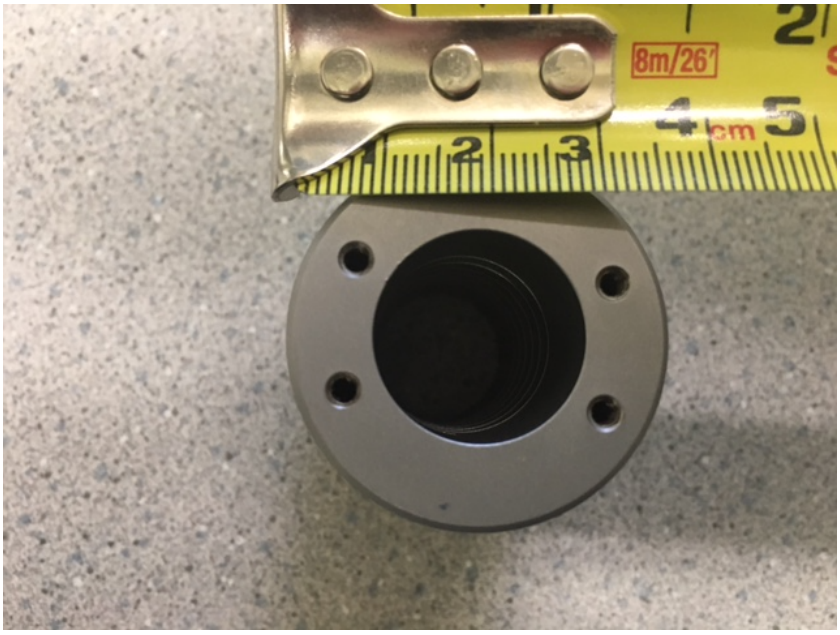

Figure 39. Top View of Helical Machined Spring Manufactured for Test Bed

When the spring arrived, it was apparent from handling it that it was too stiff to produce the desired deflection. The spring was assembled in the mount, but as expected, the amount of torque to produce any deflection, let alone the desired deflection, was not obtainable from the system. Unfortunately, the torsional spring rate determined from the simple modeling did not accurately represent the experimental system. The test bed is a complex, multi-body dynamical system. It is believed that the main error in the modeling was not accounting for the mass of the base plate. The base plate and components have a mass of approximately 31 kg compared to just 4.6 kg for the robot arm and 2.3 kg for the vibrating arm. In reality, the relatively small forcing function from the robot arm was simply not sufficient to overcome the significant plate inertia and induce the amount of deflection desired.

The spring was therefore taken to the NPS machine shop. The machinist was able to remove material from the outer diameter to create a "neck" in the spring. Figure 40 shows a before and after comparison of the machined spring. The fully assembled concept C prototype is shown in Figure 41.

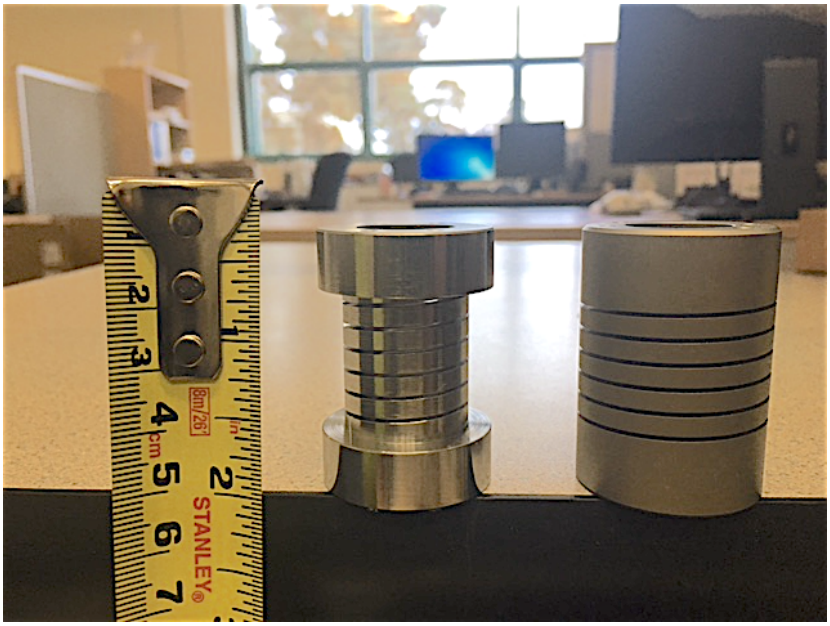

Figure 40. Comparison of Helical Machined Spring before and after Material Removed

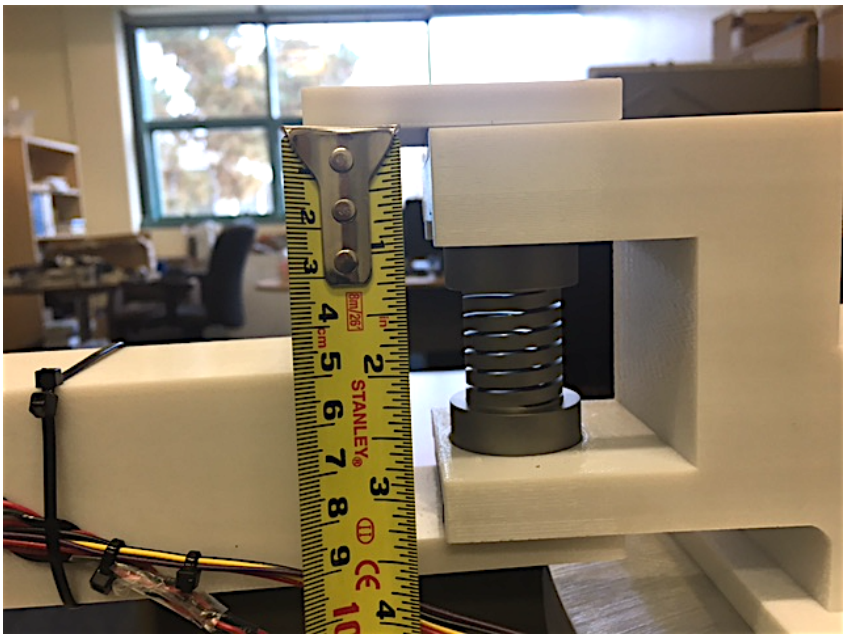

Figure 41. Concept C—Fully Assembled

When assembled the machined spring allows for deflection of approximately +/- 10 degrees. This result proves to be more than sufficient to visually appreciate the deflection of the vibrating arm as compared to concepts A and B. As well, since the machined spring now produces this larger deflection coupled with the pure moment and linear rates from a machined spring, there is confidence that concept C is the preferred choice for future experimentation.

G. SUMMARY

The main purpose of this chapter was to document the development of various concepts for realizing a flexible joint that could be used in experimentation of flexible space systems development for not just this but for many future theses. The development of each concept proved to lay the ground for the experimentation process and data analysis moving forward.

Concept A was a simple, initial prototype to conduct initial experiments of the system. Concept was a more elegant design that allowed for different spring rates to be tested easily. However, neither of these concepts produced the desired deflection of 19 deg. Concept C incorporated a helical machined spring that was benefited with pure moments and linear rates. The helical spring proved to be the best of the three concepts to produce the desired deflection for future test bed experimentation.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SENSOR INTERFACING AND DATA PROCESSING

In this chapter, the sensors and embedded computer implemented on the laboratory test bed are introduced in detail. Additionally, the post-processing procedures that were developed during the course of the thesis will be described including the difficulties encountered and the methods that were derived to mitigate these challenges and ultimately provide the most efficient and accurate means of analyzing the data after experimentation.

A. INERTIAL MEASUREMENT UNIT

In order to analyze the excitation imparted on the vibrating arm with the flexible joint, the robotic arm, and the plate, an inertial measurement unit (IMU) was mounted at the end, side face of each arm and near the outer edge of the plate, respectively. The IMU used is the nine-degree of freedom (9DOF) Razor Inertial Measurement Unit developed by the company "Sparkfun" [18]. In addition to an accelerometer in all three axes, the Razor has a magnetometer and gyroscope for each axis. The accelerometer in the Razor, the ADXL345 [19] is capable of sensing accelerations up to a maximum of 10,000 gs, far more than the accelerations to be observed in this test laboratory. The ITG-3200A gyroscope has a full-scale range of +/-2000 °/sec [20]. The Razor is shown in Figure 42. Figure 43 illustrates where each IMU is located in the final test bed configuration.

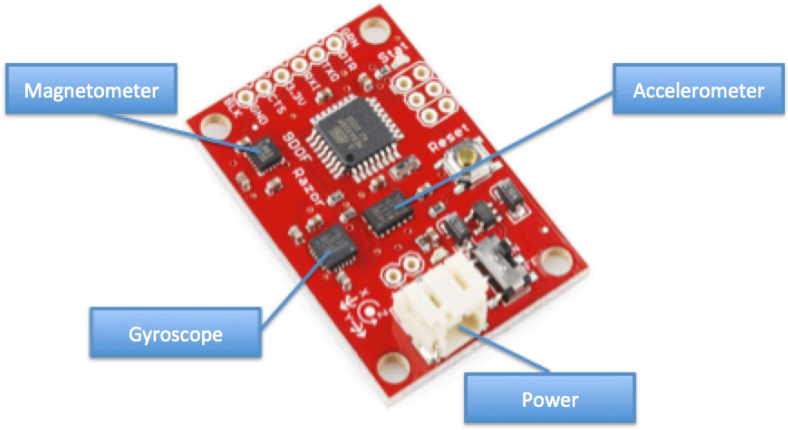

Figure 42. 9DOF Razor IMU. Source [18].

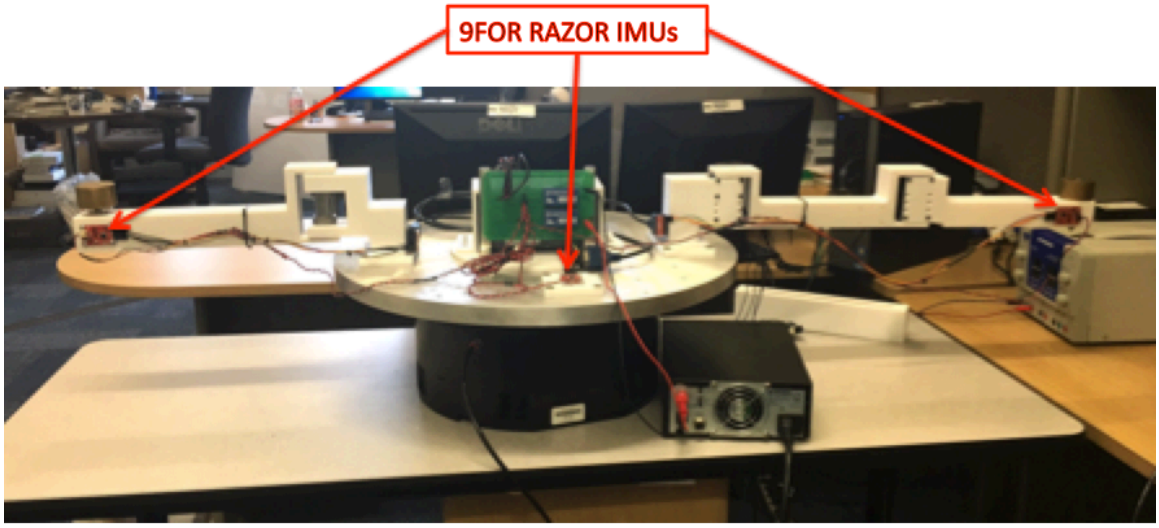

Figure 43. IMU Placement for Final Test Bed Configuration

B. CALIBRATION

Before the IMUs can be attached to their respective positions, each accelerometer and gyroscope must be calibrated. Sensor calibration is required to account for minor defects in the sensor due to manufacturing, assembly, etc. An IMU that is not calibrated can affect precision and responsiveness of the IMU such as drift in yaw when roll is applied to the board [18]. The calibration factors that are determined through the calibration process help to alleviate these minor defects that are particular to each IMU.

The procedures are detailed in a tutorial linked via the Sparkfun website [18]. To perform the calibration, each IMU was connected to an FTDI breakout board and then to the user's laptop via a USB cable. The FTDI breakout board receives the raw data from each sensor and routes that data through an embedded integrated circuit (IC). Now that the data is packaged neatly on the IC, the serial data is transferred to the host computer via a USB cable [21]. The program Arduino [22] was used to read the data off of the IMU as well as to perform the calibration. The process to calibrate the accelerometers was the most difficult and time consuming. The basic procedure was to find the maximum and minimum readings for each axis in a static test. The maximum / minimum values corresponded to a gravity measurement of 1g. To perform the calibration, the user held the IMU steady to begin the process and then slowly rotated the IMU to find the maximum and minimum values. Moving the IMU slowly so as to not induce excess accelerations was the difficult part. With this in mind, the process was performed three times to find an average maximum and minimum value for each axis. Figure 44 illustrates the captured sensor output during the acceleration calibration.

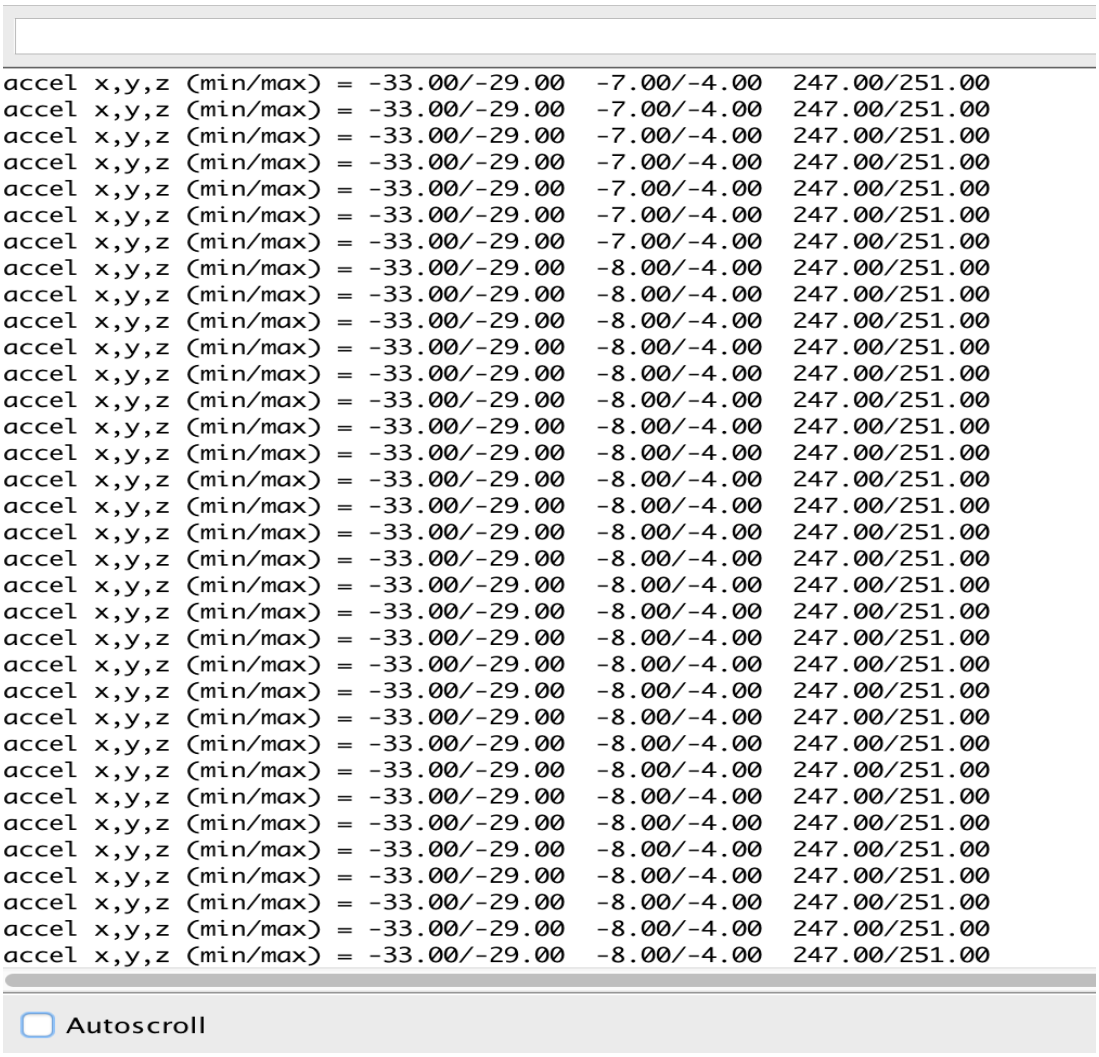

Figure 44. Example Data for Accelerometer Calibration. Source: [22].

The average values were then entered into the Arduino code by replacing the baseline values of +/-256 LSB. Figure 45 shows the location in the Arduino interface where the user enters the accelerometer calibration data. In Figure 45 the baseline accelerometer values are shown; that is +/- 256 LSB.

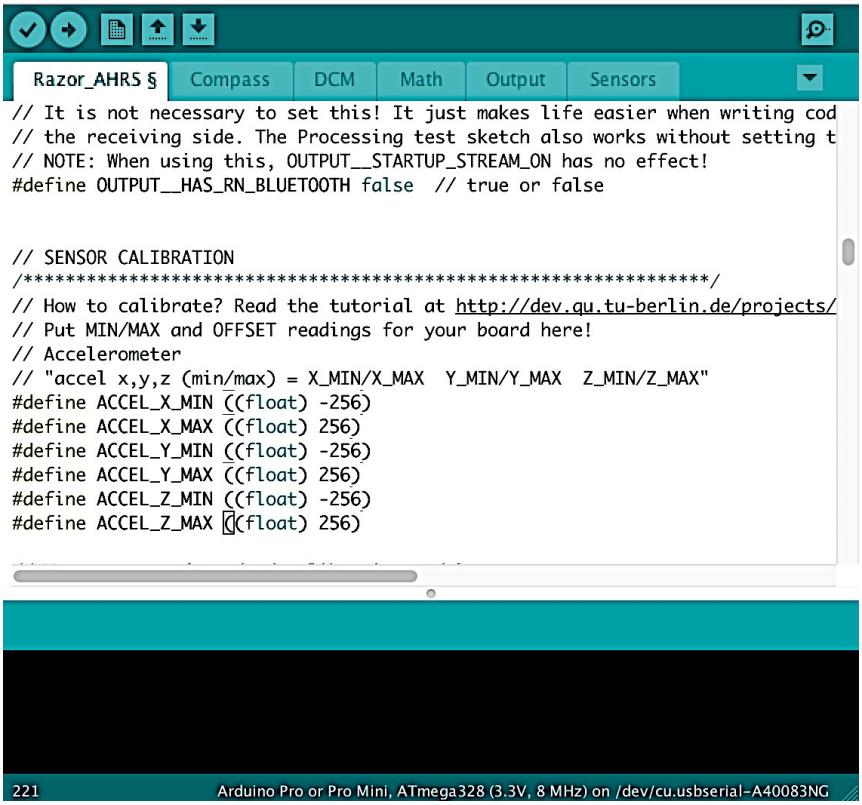

Figure 45. Coding the Accelerometer Arduino Interface. Source: [22].

Figure 46 illustrates how the calibration is changed in the code. Note the changes to each axes' minimum and maximum values.

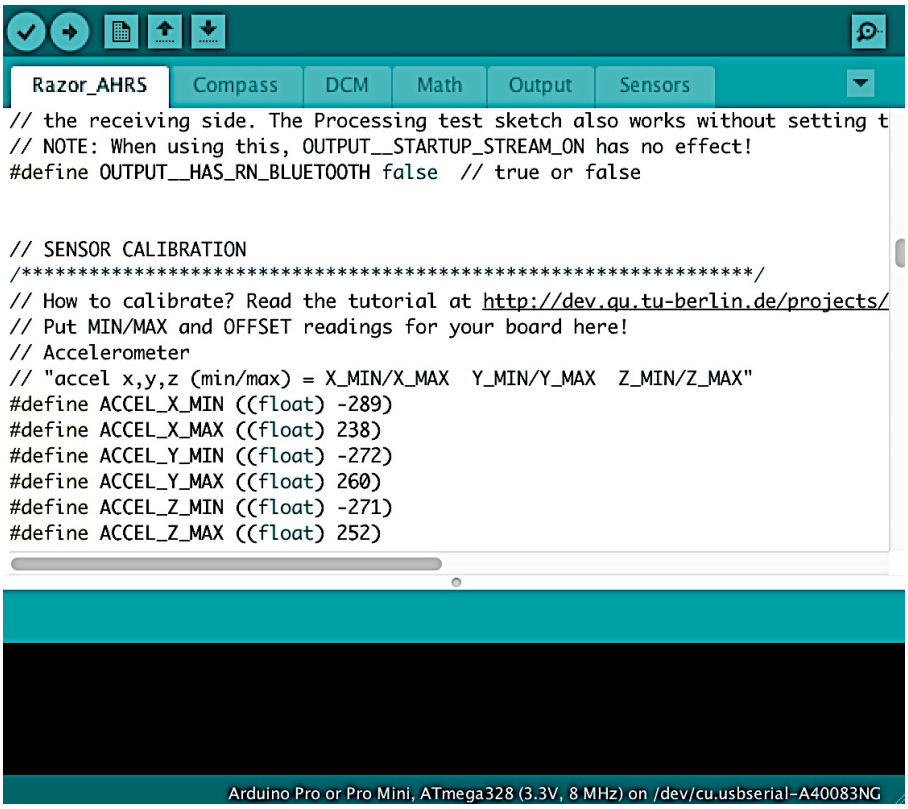

Figure 46. Coding the Accelerometer Calibration Factors. Source: [22].

After the accelerometer, the magnetometer is calibrated. To begin, the direction to North is determined. The x-axis of the IMU is pointed to North and then rotated about the East-West direction. Similar to calibrating the accelerometer, the IMU is then tilted in every direction until a maximum value is obtained, then the IMU is pointed in the opposite direction, and the IMU is titled to now find the minimum values. A similar process is performed for the y and z-axes. Each axes values are recorded and entered into the Arduino calibration code in place of the baseline values of 600 LSB.

After the magnetometer, the gyroscope is calibrated. This process is much simpler. The gyroscope is simply held steady for a minimum of ten seconds or until the values in each axis are steady. The steady values represent the gyroscope drift rate. Once

steady, the outputs are noted and entered in the Arduino code in place of gyroscope baseline values of 0.0 LSB. The gyroscopic data is in units of LSB. Tables 7, 8, and 9 show the average calibration results.

Table 7. Accelerometer Calibration Data

	X (LSB)	Y (LSB)	Z (LSB)
Plate (min/max)	-289 / 238	-272 / 260	-271 / 252
Vibrating Arm (min/max)	-269 / 266	-270 / 293	-282 / 246
Robot Arm (min/max)	-272 / 254	-254 / 265	-298 / 238

Table 8. Magnetometer Calibration Data

	X (LSB)	Y (LSB)	Z (LSB)
Plate (min/max)	-381 / 427	-571 / 394	-326 / 460
Vibrating Arm (min/max)	-399 / 459	-509 / 461	-295 / 602
Robot Arm (min/max)	-392 / 512	-439 / 422	-495 / 466

Table 9. Gyroscope Calibration Data

	X (LSB)	Y (LSB)	Z (LSB)
Plate	0.21	24.66	33.13
Vibrating Arm	-8.37	57.62	23.03
Robot Arm	-2.61	38.56	11.16

The sensitivities for each sensor, accelerometer, gyroscope, and magnetometer, were obtained from the respective data sheets. The accelerometer, gyroscope, and magnetometer data is given in LSBs. The data is then divided by the sensitivity factor for each sensor. Table 10 shows those factors.

Table 10. IMU Sensitivity Factors

	Sensor Sensitivity Factor	Converted Units for Post-Processing
Accelerometer [23]	256 +/- 30 (LSB/g)	Gravity
Gyroscope [20]	14.375 LSB/°/sec	°/sec
Magnetometer [24]	230 - 1370 LSB/gauss	Gauss

The magnetometer sensitivity factor is given for completeness. Ultimately the magnetometer was not used in the post-processing analysis, but was used only for the calibration process and to aid in evaluating the accuracy of the yaw, pitch, and roll measurements discussed in Chapter IV.

The IMUs were then attached to the system in their respective locations. A static test was the conducted to observe the accelerations along the tangential axis at the end of the arm and the outer edge of the plate. The vertical and normal axes were ignored since the primary motion of concern was in the horizontal plane, which is, vibrating motion. With this said, when the static tests were performed, it was expected that, after the calibration, the tangential axis accelerations would be near zero. If the values were not near zero, small refinements to the average calibrated values were entered until the static accelerations were observed as near zero. Even after this was completed, noise in the sensor was observed so as in a static condition the accelerations fluctuated +/- 2.0 LSB. When divided by 256 LSB/gravity, this gives a small fraction of gravity (0.0078 gs) imparted as the noise. What will be shown in the experiment results in Chapter V is that the noise was more apparent in concepts A and B due to their relatively small vibrating displacement compared to concept C.

After the calibration is complete, the user configures the Arduino output code so the displayed values only reflect what the user requires. Figure 47 illustrates the output code tab of Arduino. For example, if the Arduino output code were not changed, the yaw, pitch, roll angles and accelerations in all axes would be displayed. As will be discussed later in section E, the post-processing for concepts A and B were completed with the user's laptop computer whereas concept C incorporated a more robust system design incorporating the National Instruments cRIO 9024 embedded computer [25], discussed in

section D of this chapter. With that said tangential accelerations were only analyzed in concepts A and B for system initial development, and tangential and normal accelerations as well as gyroscopic data was captured for concept C. In Figure 47 the script colored orange represents the serial data that will be displayed. In this case all three axes of acceleration and all three axes of gyroscopic data are displayed.

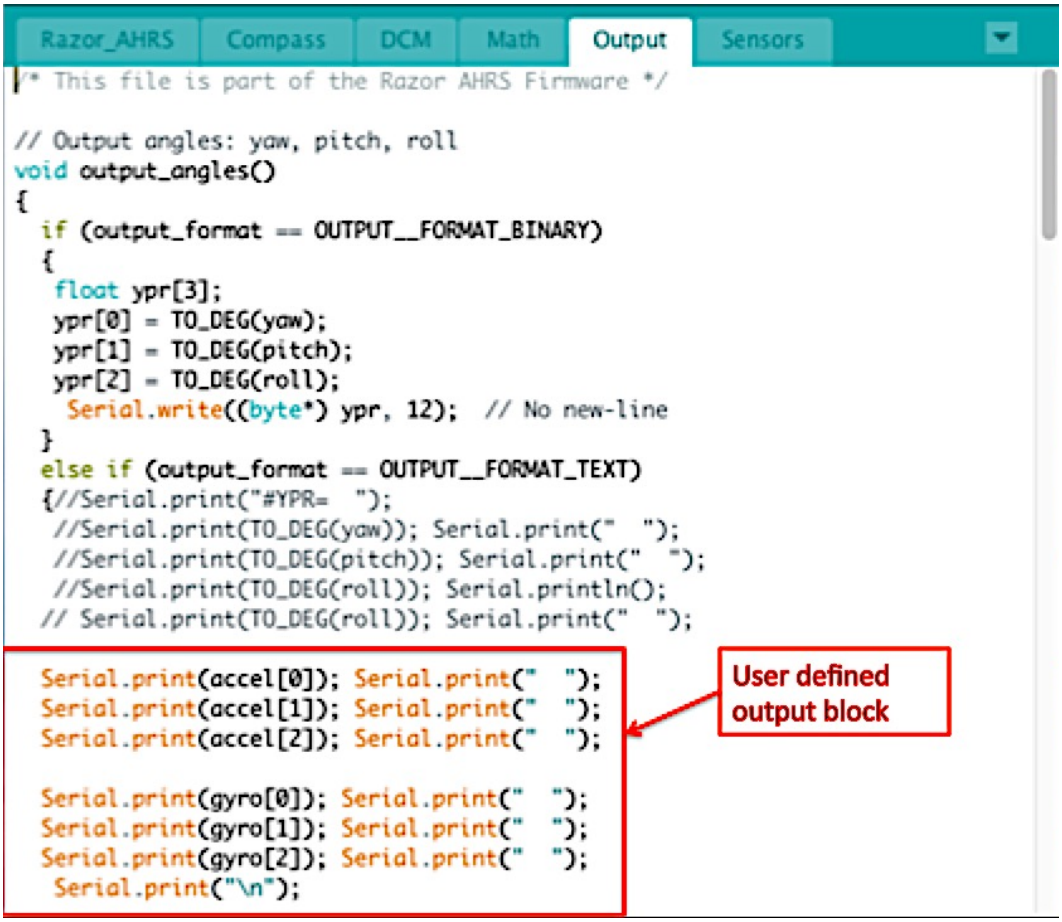

Figure 47. Arduino Output Code. Source: [21].

C. NATIONAL INSTRUMENTS CRIO CONTROLLER

In order to build a test bed interface that could monitor and control several devices at once, it was decided to incorporate an embedded computer onto the center of the plate to act as a central hub of information control. The device chosen to perform this task was the National Instruments cRIO model 9024 [25]. This model is a real-time

controller with specifications of 800 MHz, 512 MB DRAM, and 4 GB of storage. It is a real time embedded controller that runs the program LabVIEW "for real-time deterministic control, data logging, and analysis" [26]. Additionally, the chassis that holds the cRIO has four slots for serial interface modules. Three NI 9870 serial interface modules were used. Each module has four ports that connect to the IMUs via an RS232 connector. Additional ports allow future expansion of the test bed to incorporate up to 16 serial data components. Figures 48 and 49 illustrate the configuration of the embedded system.

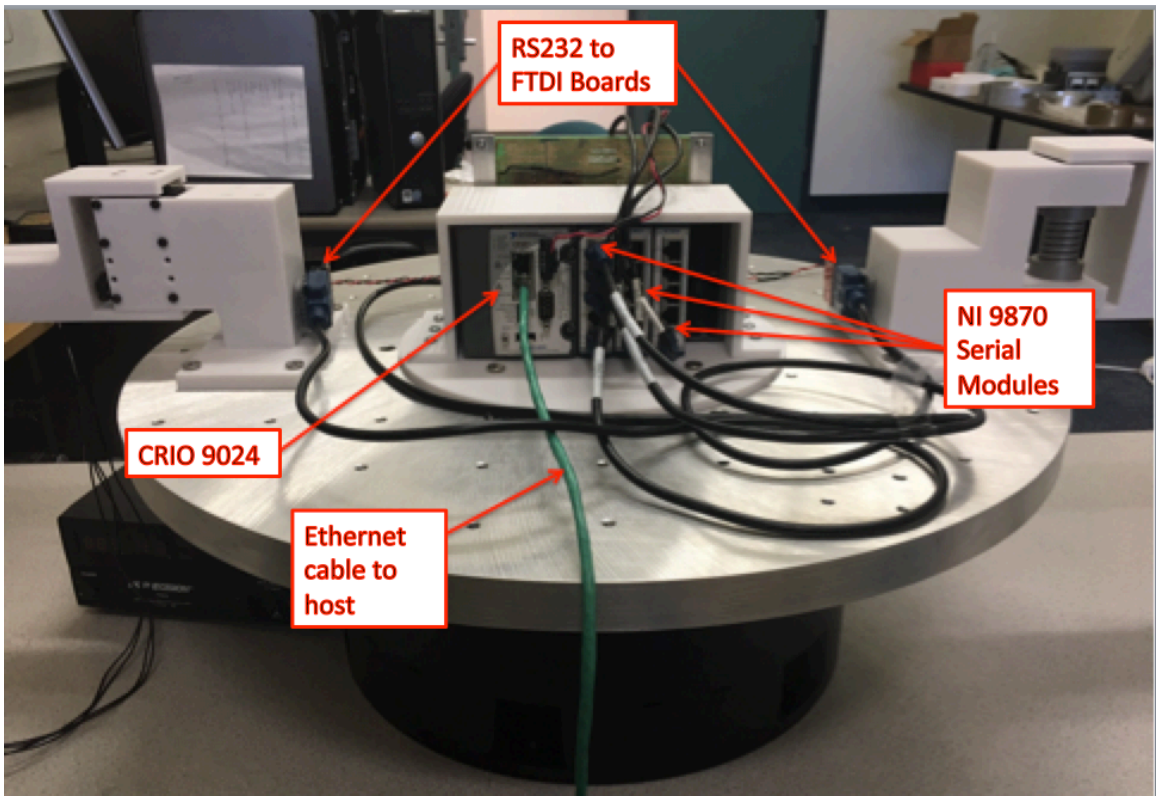

Figure 48. NI CRIO 9024 Configuration

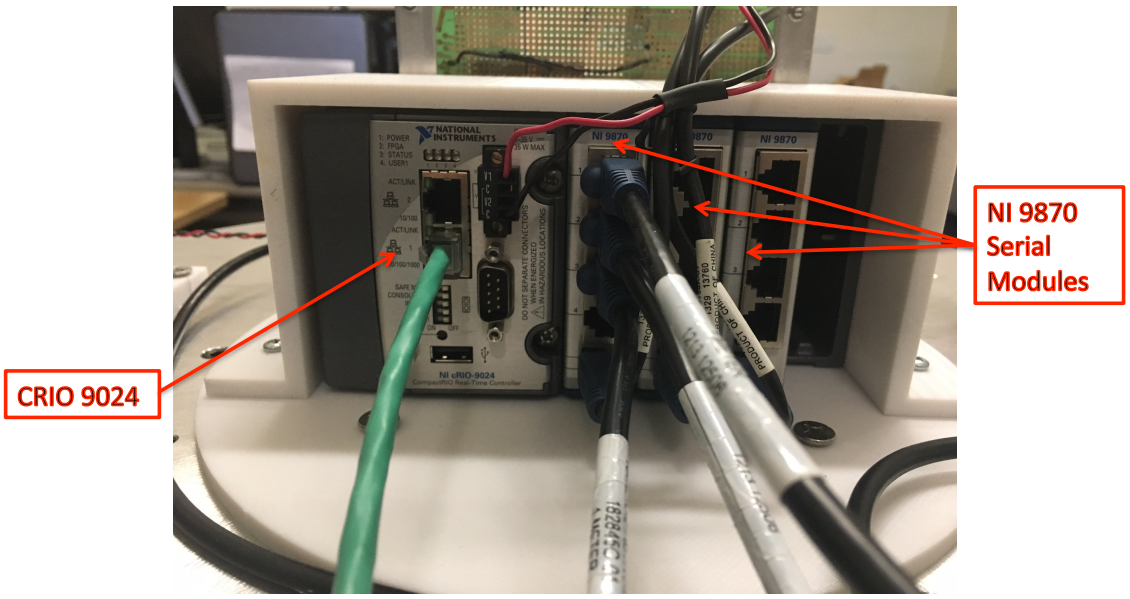

Figure 49. NI CRIO 9024 Close-up

The cRIO allows the user to monitor all IMU measurements at one time as well as providing a means to implement the optimal control trajectory to the robotic arm. This is a great advantage since the Arduino software allows the user to monitor just one IMU at a time. Moreover, the Robotis Dynamixel Wizard software does not provide a means to incorporate time-tagged control trajectories. In order to achieve this embedded functionality, the built in program LabVIEW will be used. LabVIEW is central to National Instruments approach to engineering and science applications. LabVIEW is a development environment with a graphical programming syntax to visualize, create, and code engineering systems. Additionally, LabVIEW is designed to interoperate with other software to ensure multiple tools are available to the user from a central platform [26].

The cRIO and IMUs were powered by a separate power supply providing 24 VDC. Power was routed through a distribution board to each IMU and the cRIO. Figure 50 illustrates the power distribution board. The two DC-DC converters provide outputs of +12VDC and +5VDC to power the various devices.

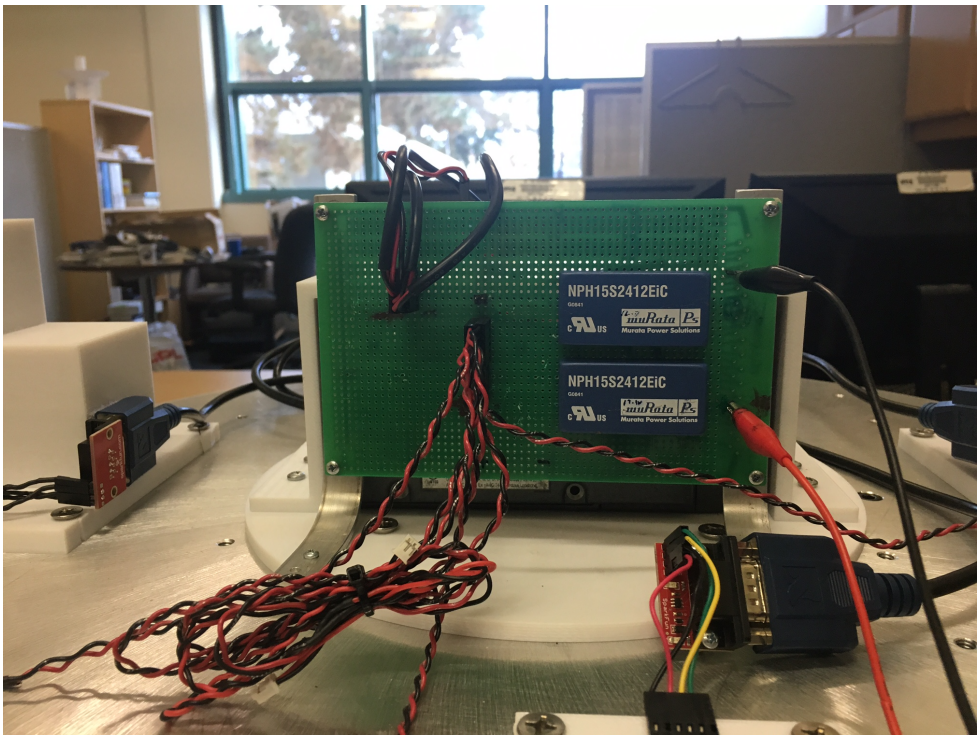

Figure 50. Power Distribution Board

Additionally, an Ethernet cable is connected from the cRIO to the host computer for monitoring and post-processing. Subsequent theses in support of this project will incorporate a housing for a 24 VDC battery as well as components to communicate wirelessly with the host computer. This will enable the test bed to be completely free of external cables and wires; therefore, no external torques from hanging cables will be applied to the overall system.

1. LabVIEW IMU Integration

In order to capture and display IMU output, the LabVIEW program was used. LabVIEW has a graphical coding environment similar to Simulink. A download from the National Instruments website support page was used to provide an example graphical block set that could be used to stream serial data [27]. Minor changes were made to the code in order to write the data from each IMU to a text file. Figure 51 illustrates the code block that was used while Figure 52 shows the front panel display for each port of the serial data module. For this test rig, three ports are used, one for each IMU.

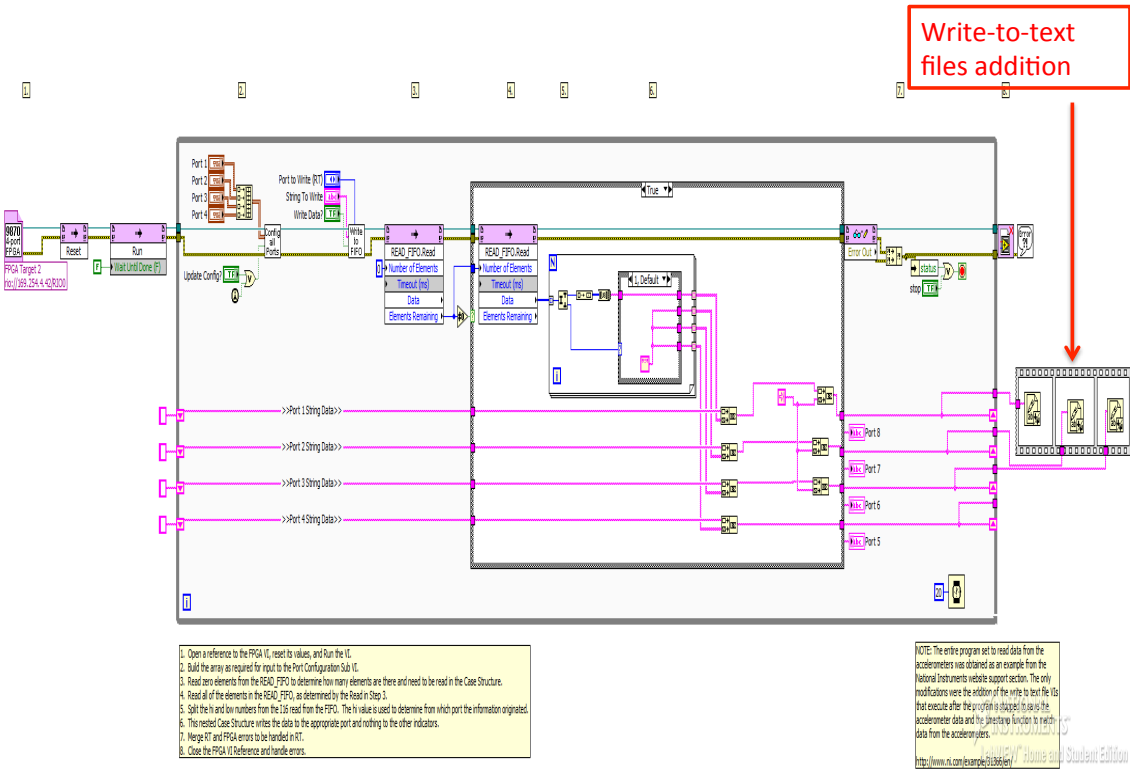

Figure 51. LabVIEW Continuous Serial Data code for IMU DATA. Source: [27].

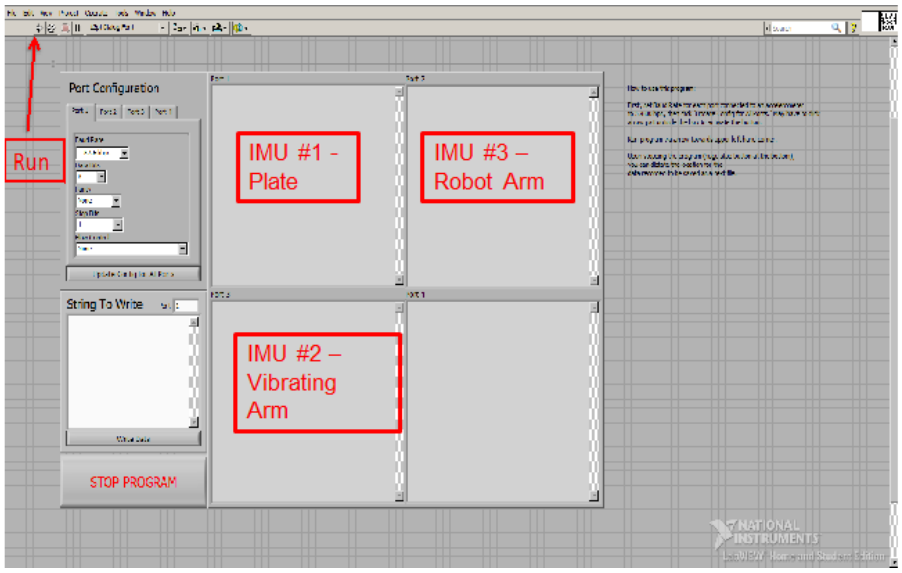

Figure 52. LabVIEW Continuous Serial Data Front Panel for IMU Data. Source: [27].

2. Collecting IMU Measurements

To collect and view the data, the "run" button on the front panel is clicked and serial data begins to stream. Once the experiment has completed, "stop program" is selected, and the first of three windows appear. Each window saves the serial data from the IMUs into separate text files for post-processing.

3. LabVIEW Robotic Arm Model Development

The utility and capability of LabVIEW was not just in the serial data processing, but also in the interface with robotic actuators. Code to read and display commands to the actuators was found in a National Instruments discussion forum. LabVIEW has compatibility with the Robotis MX-106 actuators. The LabVIEW code to implement the optimal control trajectory to the robotic actuators was created with the assistance of the National Instruments discussion forum and intern assistance [28]. These actuators are not as complex as the M42-10-S260-R actuators of the robot arm, but are useful in demonstrating the initial process and capability of LabVIEW to implement optimal control paths. Drivers for the more capable M42-10-S260-R actuators are needed for use in the LabVIEW environment. Not having the drivers restricted the ability to perform experimentation with these actuators. A note to the reader, in the process of developing the test rig, two of three M42-10-S260-R actuators that were to be used were damaged and therefore were not able to be used for experimentation. The actuator that was not damaged was mounted to the robot arm base to perform the experimentation discussed in Chapter IV. Due to the hardware issue only one DOF could be actuated for the tests. Figure 53 illustrates the graphical code that could be later extended to drive the M42-10-S260-R actuators.

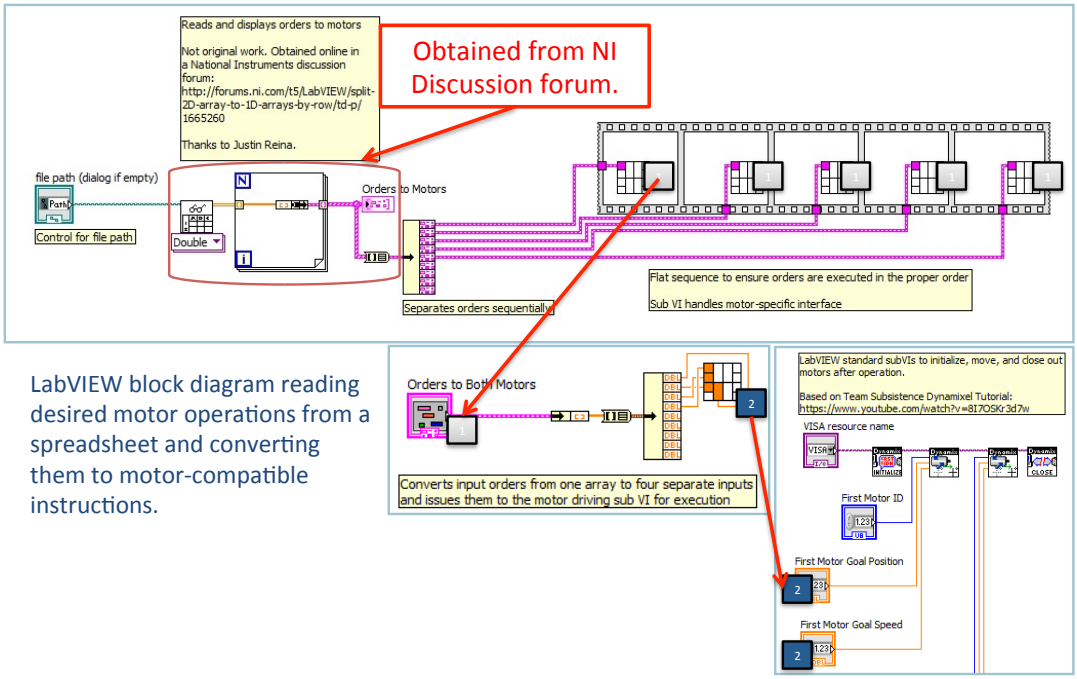

Figure 53. Robotic Motor Control LabVIEW Code. Source: [28].

D. DATA POST-PROCESSING PROCEDURES

1. Beginning Processes for Concepts A and B

To read the data from the sensors, software from Arduino was used to program Razor's microprocessor. The Arduino software provides a means to read accelerometer, magnetometer, and gyroscope data via a serial monitor [21]. Arduino itself does not have tools to plot the data received from the Razor. There is a means to view serial data graphically, but it is for real-time use only and not for post-processing needs. The user is required to develop a means to read the serial data into other programs such as MATLAB for analysis. Without editing the Arduino code, the serial monitor displays yaw, pitch and roll angles calculated from the nine inputs received from the Razor. In order to capture the data, the code for the serial monitor display was edited to display only raw acceleration measurements. The Arduino user interface provides a relative easy means to make such edits as discussed earlier.

Once the serial monitor displayed only accelerations for each axis, further edits were made such that only the axis excited by the movement from the two-link arm would be displayed. With the serial monitor now displaying the single, excited axis values, experiments could be conducted with the goal of measuring the motion induced on the flexible link due to movement of the robot arm. For the initial testing of each concept, the maximum torque possible, 5.6 Nm/actuator, from the Dynamixel M42-10-S260-R actuator was applied. The trajectory of the two-link arm displaced the linkage approximately 160 degrees; that is, the movement of the two-link arm was displaced from one side to the other and vice versa. Figure 54 shows the lab test bed where the two-link electrically actuated arm is on the right and the one-link arm with flexible joint is on the left.

Figure 54. Lab Test Bed—Early Stage of Completion

To plot the data, the following process was conducted. First the serial monitor displayed the data as discussed previously. With this said, it should be understood by the reader that the serial monitor continuously displays data from the Razor; that is, once initialized, the Razor will continue to produce data, moving or not, until the USB is disconnected from the test bed computer. Therefore, the basic method to capture the data was to send a command to the two-link arm via the Robotis software, allow the flexible joint arm time to settle and then disconnect the USB from FTDI breakout board. From this point, the serial monitor freezes. The data displayed on the serial monitor was then copied and pasted into MATLAB for analysis.

2. Refined Post-process Development

As mentioned previously, there was an accepted delay in the design and manufacturing of the machined helical spring. Once the beginning processes proved the test bed's basic functionality worked; that is, the robotic arm imparted a torque on the opposite flexible joint link arm, a more refined, robust procedure was developed to aid in the post-processing procedures for concept C, but more importantly to construct a means to evaluate several IMUs at once as well. The NI cRIO allows the user to do this. Figure 55 shows the final test bed construction.

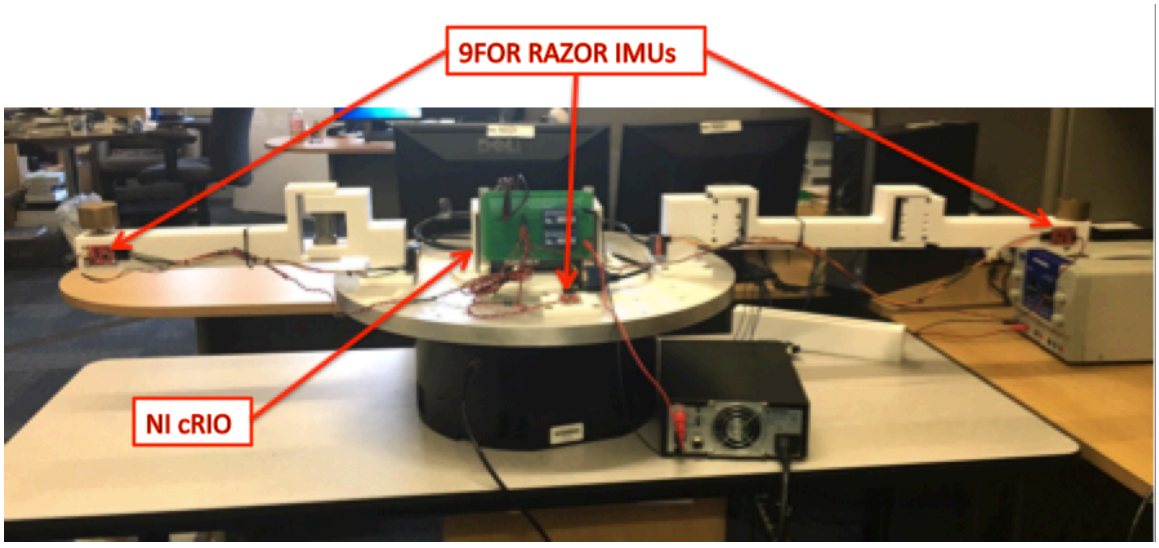

Figure 55. Lab Test Bed—Final Stage of Completion

The main limitation of the cRIO is that there is not a means to calibrate the IMUs nor upload the Arduino code to the IMU after the calibration and desired data for display has been entered. With this said, similar to the initial process, a separate laptop was used to perform this step of the process. Once the Arduino code was uploaded to each IMU, they were then secured to the test bed and connected to a RS232 connector, which was then connected to the NI cRIO.

The initial measurement techniques focused on monitoring the tangential acceleration of the vibrating arm to gain a sense of the system performance. However, it was determined that the analysis also required additional components of acceleration and gyroscope data to be captured. The actual angular displacement of the vibrating arm was desired as well to provide a visual sense of the experiments. This would confirm that, with simple time-series integration, the acceleration data could be processed to produce the angular motion that was being observed. Several iterations to obtain success had to be performed.

A typical experiment would be conducted by commanding the robotic arm through some trajectory. This imparts a torque and subsequent motion on the free-floating plate in the opposite direction of the robotic arm movement. This causes the vibrating arm to initially displace in the opposite direction of the plate. As the arm vibrates, its motion imparts subsequent torques on the plate until the motion is completely damped.

Initially, it was thought that the tangential acceleration could be sensed at the end of the vibrating arm and then mapped to the angular motion of the flexible joint by the rotational mechanics (4.1) and (4.2).

$$a_t = \alpha * r \tag{4.1}$$

$$\alpha = a_t / r \tag{4.2}$$

where α is in units of *rads/sec*^{2}. Then if the plate tangential accelerometer could be mapped to the center of the plate in a similar manner, the plate angular acceleration could be subtracted from the vibrating arm angular acceleration and then double integrated to

give the position angle of the flexible appendage. As it turns out, simply double integrating acceleration to obtain position produces invalid results due ignorance of the velocity and position initial conditions [28].

Figure 56 illustrates the erroneous results obtained from initial testing of concept C. The figure shows a comparison of integrated accelerations obtained by a numerical integration code written in Excel to MATLAB's cumtrapz [29].

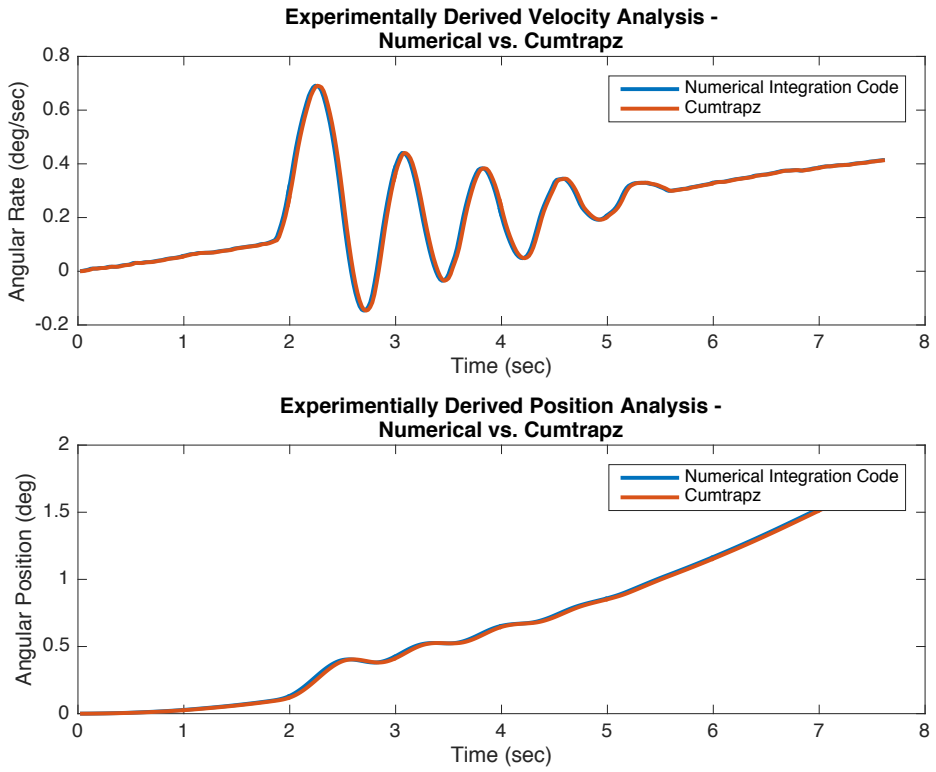

Figure 56. Experimental Data Analyzed by Double Integrating Raw Accelerometer Data

At first glance, the results of Figure 56 appear to make sense. When the experiment is conducted, the motion of the robot arm causes the plate to freely rotate while the flexible joint arm is vibrating. When the flexible joint arm stops vibrating, the plate continues to drift as is observed in Figure 56. However, the fact that velocity was "drifting" as well, that is increasing in this case, didn't make sense. To investigate this

"drift" phenomenon, a simple sine wave was double integrated via a Simulink model. Figure 57 depicts the model used.

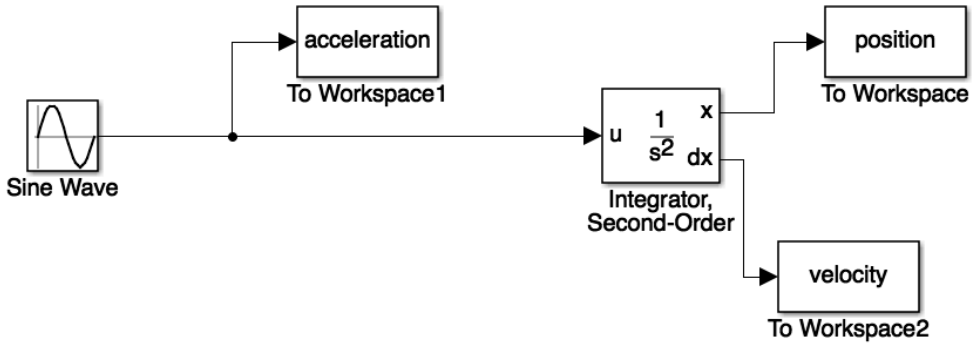

Figure 57. Simulink Model of a Double Integrator System

Figure 58 illustrates the results of the Simulink simulation assuming zero-initial conditions on velocity and position.

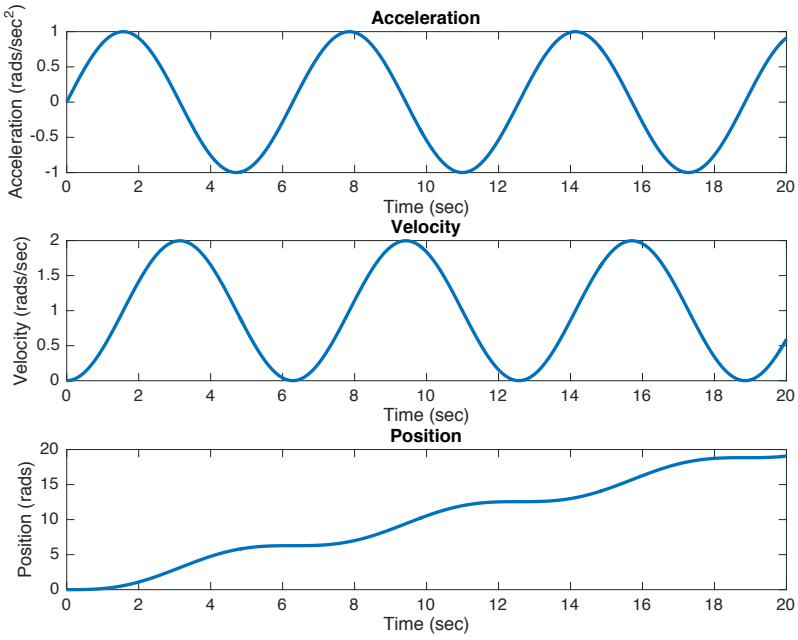

Figure 58. Double Integrating a Sine Wave without Accounting for Initial Conditions

Though velocity has the same sinusoidal shape of the acceleration plot, notice that its amplitude has shifted up introducing a drift error. The position plot in Figure 58 is the most apparent that something is wrong. Calculus tells us that the resultant plot should return a sinusoidal result as seen in the acceleration curve. Yet, the position drifts as seen in Figure 58. The same simple sine function was double integrated via numerical integration methods, a user created algorithm and MATLAB's cumtrapz function as a check on the Simulink results. Both simulations produced the same results as in Figure 58.

It turns out that this phenomenon is more common then realized and is discussed in detail in [30]. The problem is that when integrating acceleration data twice to produce position data, the exact initial conditions to derive the position from velocity are unknown. With time-series data, one does not know how much to shift the data or at what time-series data point to accurately account for the initial conditions of the second integration which leads to position determination. The resulting position plot provides erroneous data.

To verify the above arguments, another simulation was performed. In a Simulink model, the user can input the initial conditions as desired resulting in expected plots where velocity is out of phase with acceleration and furthermore position is out of phase with velocity, provided, as in Figure 59, that the correct initial conditions are selected.

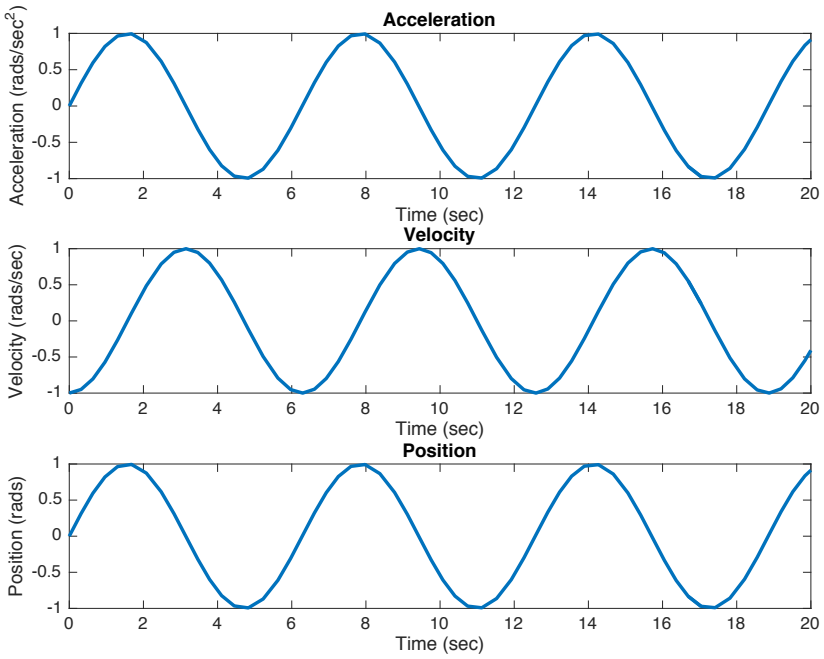

Figure 59. Double Integrating Sine Wave with Correct Initial Conditions

To overcome this challenge, it was determined that the gyroscopic data could additionally be used. The method that was finally settled on to accurately represent the motion of the system was to first differentiate the gyroscope data to compute the angular acceleration. This operation in MATLAB is performed as:

$$acceleration\ data = \frac{diff(gyro\ data)}{diff(time)}$$

Once each angular acceleration was obtained, the data was numerically integrated with cumtrapz to obtain angular rate, and then once more to obtain angular position data.

1. $rate = cumtrapz(acceleration)$
2. $position = cumtrapz(rate)$

The results of this process as applied to concept C are discussed in Chapter V. Frankly it is a peculiar outcome considering previous numerical methods gave erroneous results; however, the process above does produce the observed results. The explanation is that differentiating the gyro data to obtain an angular acceleration and then integrating to arrive back at angular rate allows drift to be corrected because of the mismatch between

finite differentiating and the trapezoidal rule for integration. To attempt to recreate this phenomenon manually, the vibrating arm acceleration data was numerically integrated to arrive at the plot in Figure 60.

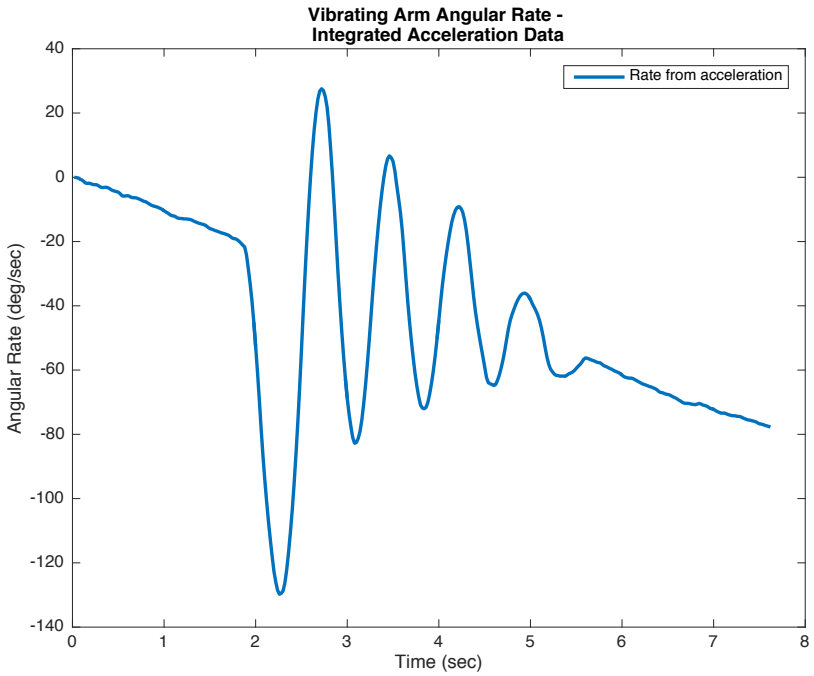

Figure 60. Vibrating Arm Angular Rate by Numerically Integrating Acceleration Data

Taking the initial and final y-axis values of the data in Figure 60 created the slope of the line. Figure 61 shows the slope transposed onto the plot in Figure 60.

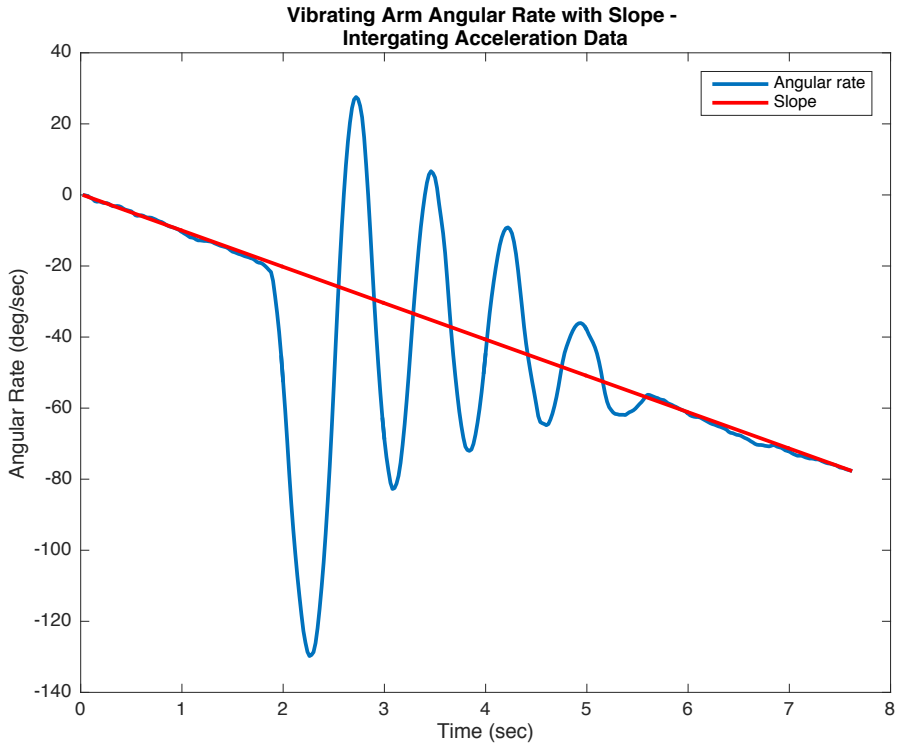

Figure 61. Vibrating Arm Angular Rate with Slope Overlay

The slope array was added to the vibrating arm angular rate data. This addition is the manual approach to removing the slope bias error incurred from the initial cumtrapz numerical integration. However, the problem is that the final value of the vibrating arm angular rate is accurate in MATLAB to four significant digits. To form a data series for the slope with its length equal to the time array, the time steps between each data point will have a small error of accuracy. Therefore, the slope data series end point will have a slightly different value than the original data's end point.

$$vib\ arm\ data\ corrected = \ vib\ arm\ data + slope$$

Figure 62 shows the results of the vibrating arm angular rate data compared to the process eventually used in the final experiment. One can see that the plot adjusted for slope is not smooth due to the error present in computing and then adding the slope adjustment.

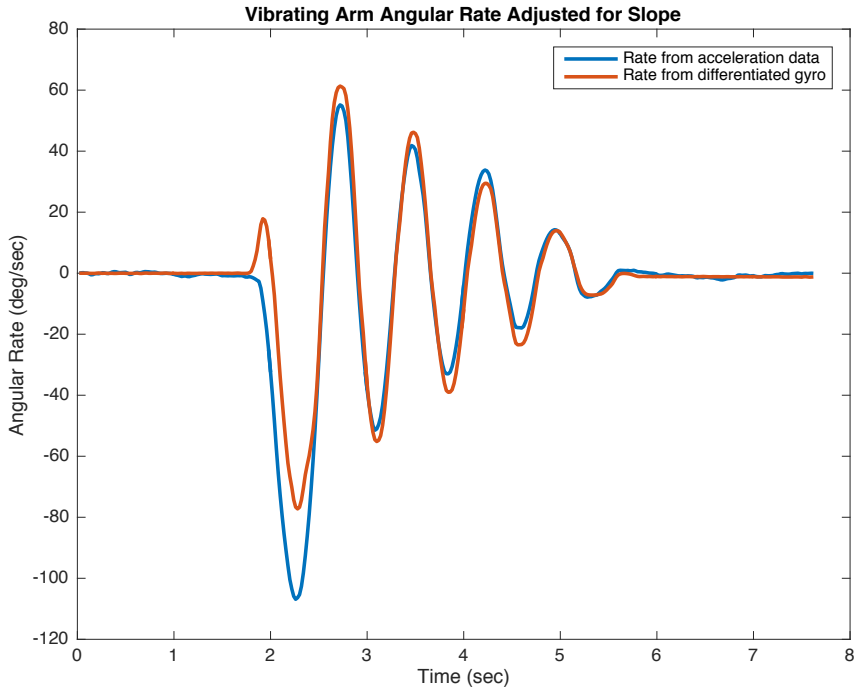

Figure 62. Vibrating Arm Angular Rate Adjusted for Slope Compared to Method of Determining Rate from Differentiating Gyro

With the plot corrected for the slope bias, the data was numerically integrated again with cumtrapz to arrive at the vibrating arm position data.

$$vib\ postion = cumtrapz(vib\ arm\ data\ corrected)$$

Figure 63 shows the results of the slope adjustment compared to the method that was ultimately used in the final experimental process. The small amount of error in the rate data contributes to the inaccuracies of the position data. While the results of manually accounting for the bias do not produce results as accurate as the real-world motion of the test bed compared to differentiating the gyro data, the plot is comparable and certainly better than not accounting for slope or initial conditions. This result also confirms the notion that by differentiating and then integrating the gyroscope data the bias slope is effectively canceled out. Additionally, the lower plot of Figure 63 adds a comparison of the increased error propagated through when simply double integrating the acceleration data and not accounting for the slope or initial conditions.

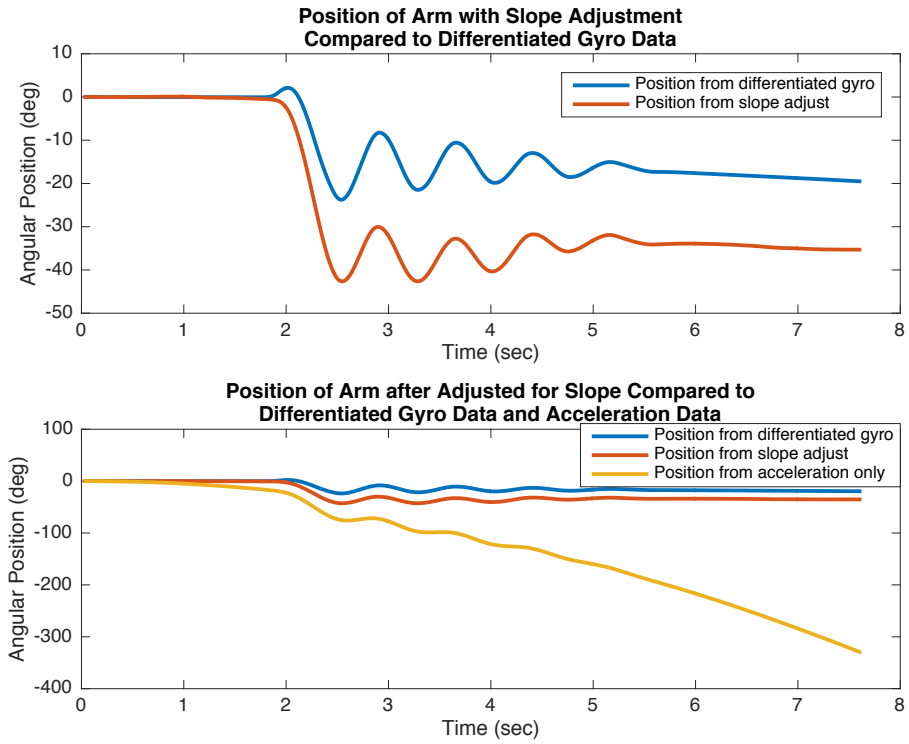

Figure 63. Position of Vibrating Arm when Adjusted for Slope Compared to Alternate Methods

For another comparison, the raw vibrating arm rate gyro data was numerically integrated to produce angular position. As discussed previously, when numerically integrated a plot with bias error is produced as seen in Figure 64. Additionally, Figure 64 illustrates the slope associated with the trace.

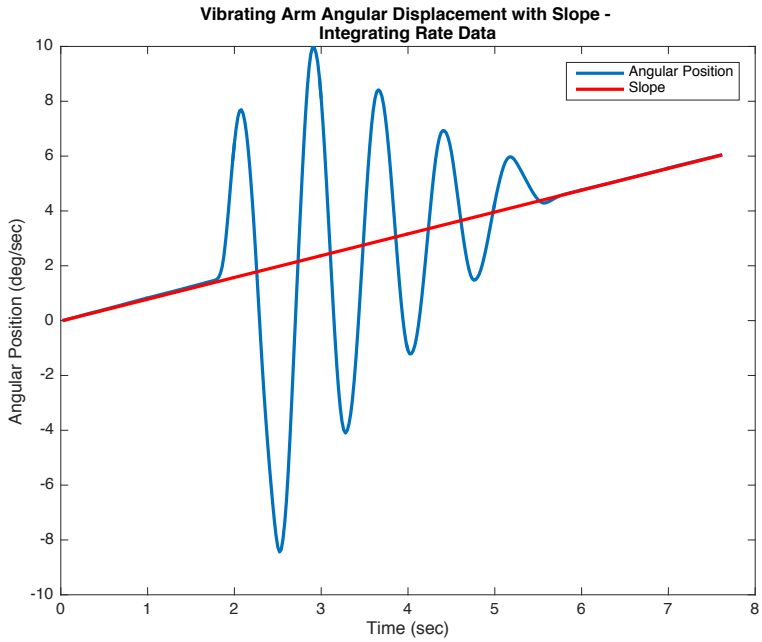

Figure 64. Vibrating Arm Angular Displacement with Slope Overlay

As in Figure 62, if the vibrating arm angular displacement plot, derived from raw rate gyro data, is transcribed to eliminate the drift produced from the bias error, a comparison can be made to the technique discussed previously of differentiating the rate gyro first. As one can see from Figure 65, the plots are nearly identical illustrating that both techniques can produce very similar results; however, performing the differentiating first eliminates the manual transcription approach.

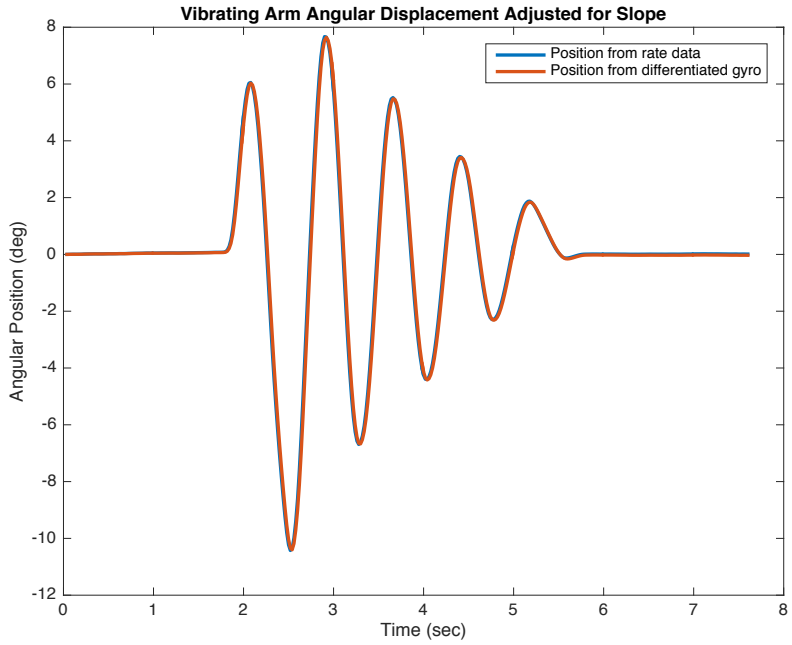

Figure 65. Vibrating Arm Angular Displacement Adjusted for Slope Compared to Method of Determining Position from Differentiating Gyro Data First

E. IMU ANGLE MEASUREMENTS

One last piece of data analysis was to compare yaw output from the Arduino code to the methods for determining system motion from acceleration and gyroscope data presented in this chapter. As discussed in the calibration chapter, the IMUs' magnetometers were calibrated in accordance with the tutorial in reference [23]. The magnetometer data coupled with accelerometer data was used in Arduino to calculate the yaw, pitch, and roll of the IMU. The IMUs output was recoded to output the yaw data from each IMU. With this said, lower-cost IMUs such as the 9DOF Razor tend to be more susceptible to magnetic influences that in turn affect the output of the angle, or yaw, measurements. Figures 66–68 show the results of yaw data of the vibrating arm, plate, and robot arm compared to the methods used earlier to derive the system displacements.

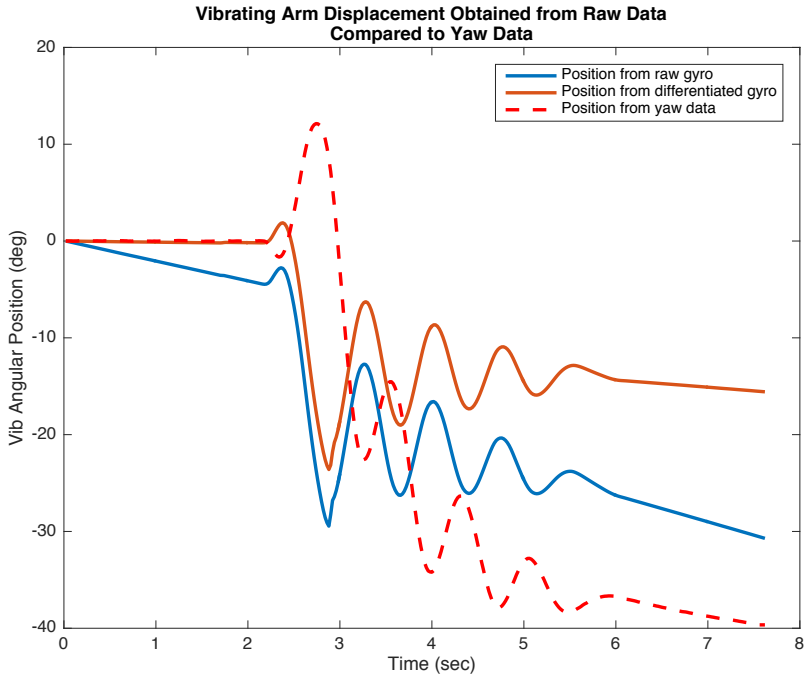

Figure 66. Vibrating Arm Displacement Compared to Yaw Measurements

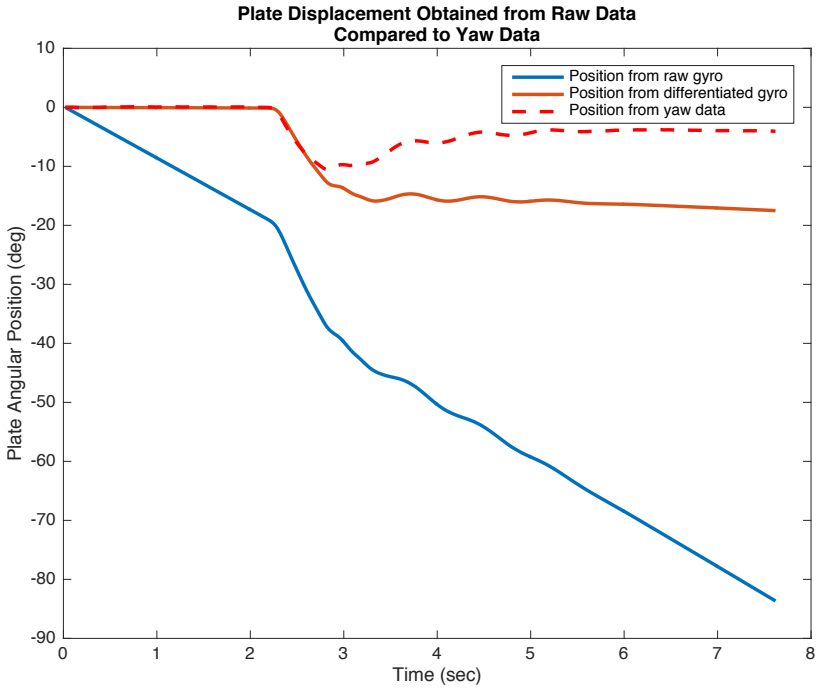

Figure 67. Plate Displacement Compared to Yaw Measurements

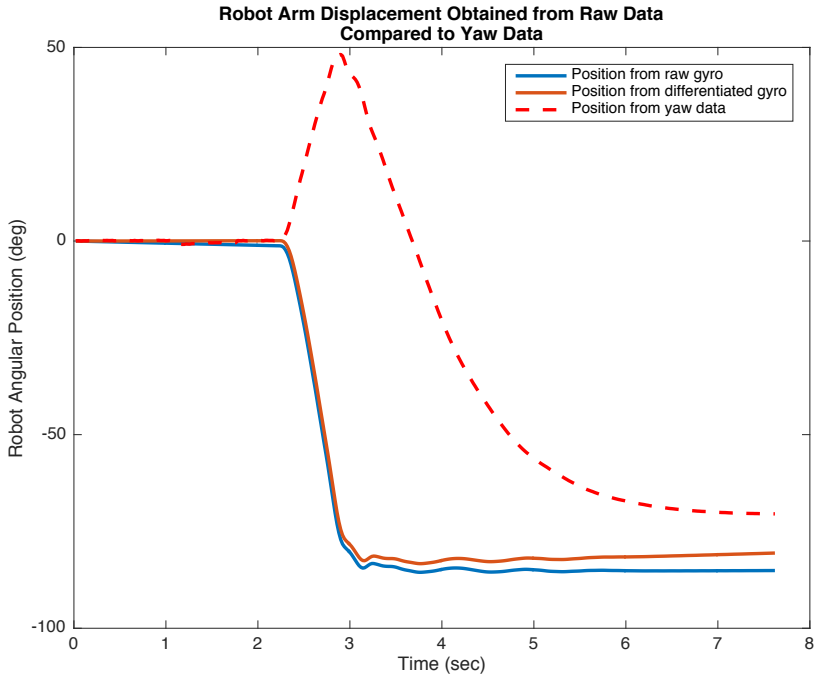

Figure 68. Robot Arm Displacement Compared to Yaw Measurements

While the yaw data is somewhat comparable, it is not as accurate as using the raw acceleration and gyroscope data to derive the system motion. The visually observed experimental displacement seems to be closest to the processing technique that uses the method of differentiating the rate gyro data first.

F. SUMMARY

In Chapter IV, test bed instrumentation was introduced by illustrating the sensors used to capture the motion of the test bed appendages and plate. It also demonstrated how these signals are processed via the NI cRIO. Additionally, post-processing procedures were developed highlighting the pitfalls one may encounter by naively double integrating angular acceleration data to arrive at angular position. Furthermore, it was shown that using raw accelerometer or rate gyro data to compute motion data, such as angular position, are more beneficial than using the IMUs yaw measurements.

The discussion to follow in Chapter V demonstrates the results from the initial processes of concept A and B, and finishes with the fully developed process for analyzing the data from concept C, the primary experimental method moving forward.

THIS PAGE INTENTIONALLY LEFT BLANK

V. LABORATORY EXPERIMENTS

The previous chapters demonstrated the purpose and method to designing and constructing the experimental test bed. Chapter V presents some example responses from each of the designed flexible joint prototypes and compares and contrasts the suitability of each design. Insight into a path ahead for future theses is also presented.

A. CONCEPT A—JOINT WITH TWO COUNTER SPRINGS

Before experiments could be conducted, the accelerometers were calibrated in accordance with the tutorial of reference [18]. Once calibrated, static testing was conducted to determine if the flexible joints responded appropriately, that is, produced a damped oscillatory motion in response to the movement of the two-link arm. For the purposes of the static experiment, the applied force was performed by simply manually displacing the link and followed by release of the link. Figure 69 is a simple schematic of a top down view of the system setup with the base plate is fixed. This approach was used for testing both for concepts A and B.

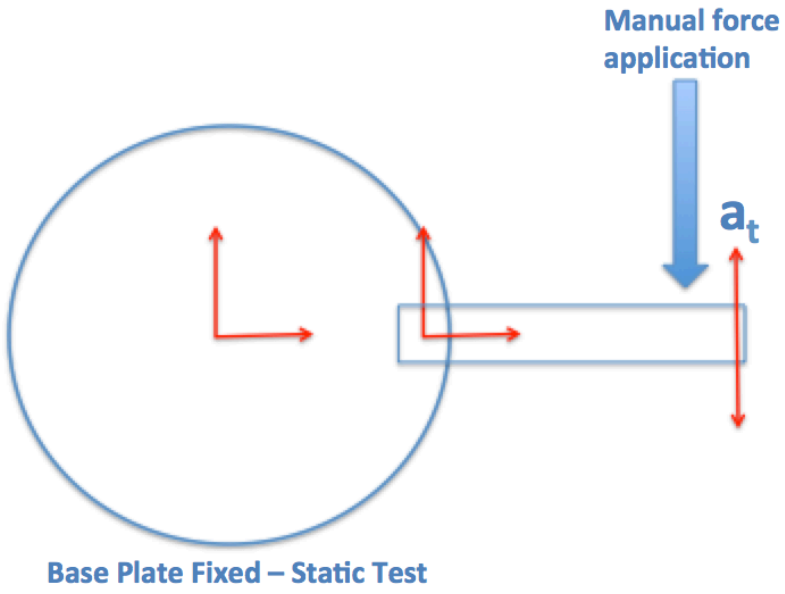

Figure 69. Schematic of Setup for Static Testing Concepts A and B

Figure 70 shows the acceleration response measured for concept A. One can see from Figure 70 that the response after the link is released damps out as expected with an exponential decay in the amplitude of the vibration. Additionally, in Figure 70, the disparity in the response prior to damping is simply a reflection of the user's input while displacing the link. The finding from this static experiment shows the flexible joint functions well enough to conduct experiments with air pressure applied to the base.

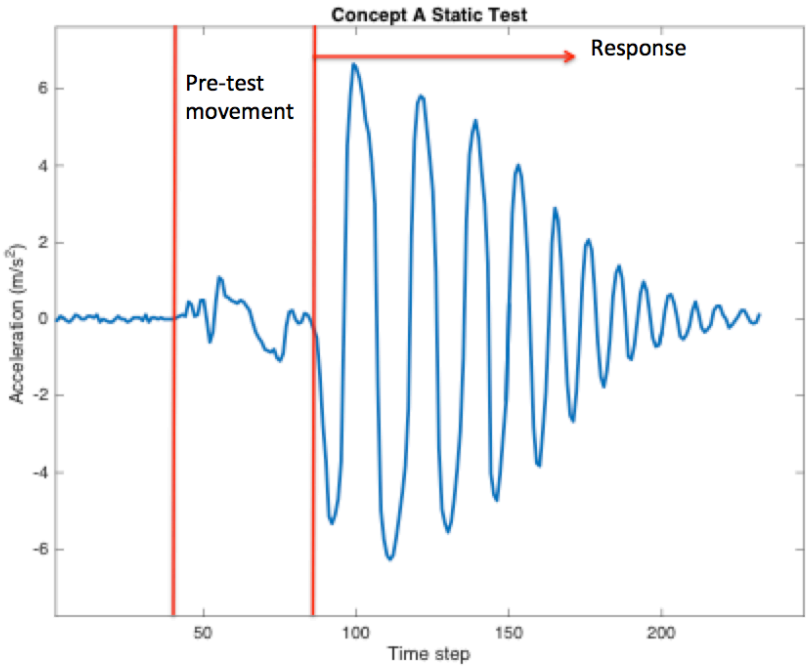

Figure 70. Concept A—Static Test Response

With the static test showing an appropriate response, experiments with pressure applied to the plate and input imparted by the two-link electrically actuated arm were conducted next. Figure 71 illustrates is a simple schematic of a top down view of the system when the base plate is floating and the input force to the system is applied via robot arm motion. This approach was used for testing concepts A and B.

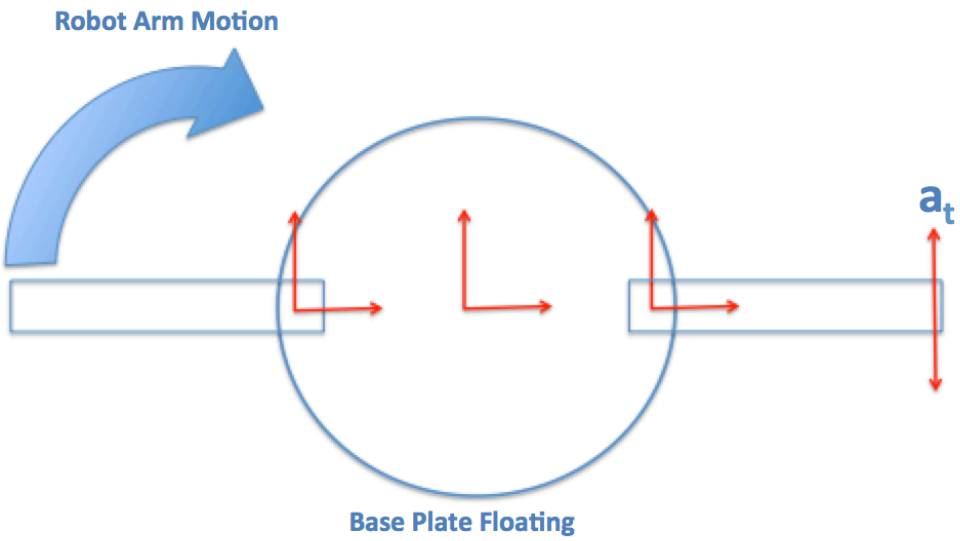

Figure 71. Schematic of Setup for Dynamic Testing Concepts A and B

Figure 72 illustrates the measured acceleration with a free-floating plate. From the plot, one can clearly see the initial increase in acceleration, coincident with the initial acceleration of the actuators, followed by a deceleration when the two-link arm stops, and then finally a settling of the flexible joint.

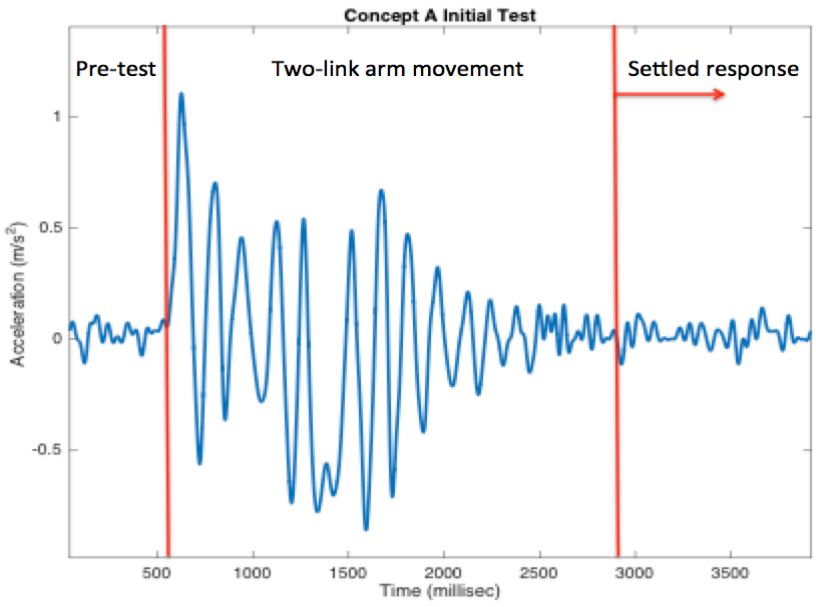

Figure 72. Concept A—Dynamic Test Response

Figure 72 demonstrates a reasonable result. When a satellite arm moves, counter forces are transcribed to not only other appendages but also the entire spacecraft. On this test set, when electric actuators displace the two-link arm, the entire free-floating plate moves in the opposite in accordance with Newton's 3rd law. Thus, the acceleration response includes a component due to the motion of the plate. In order to separate the two, an additional Razor IMU must be used to measure the motion of the plate as described in Chapter IV.

B. CONCEPT B—SPRING FLEXION PIN

Similar to concept A, the accelerometer for this concept was first calibrated, and then a static test was conducted. Figure 73 illustrates the results.

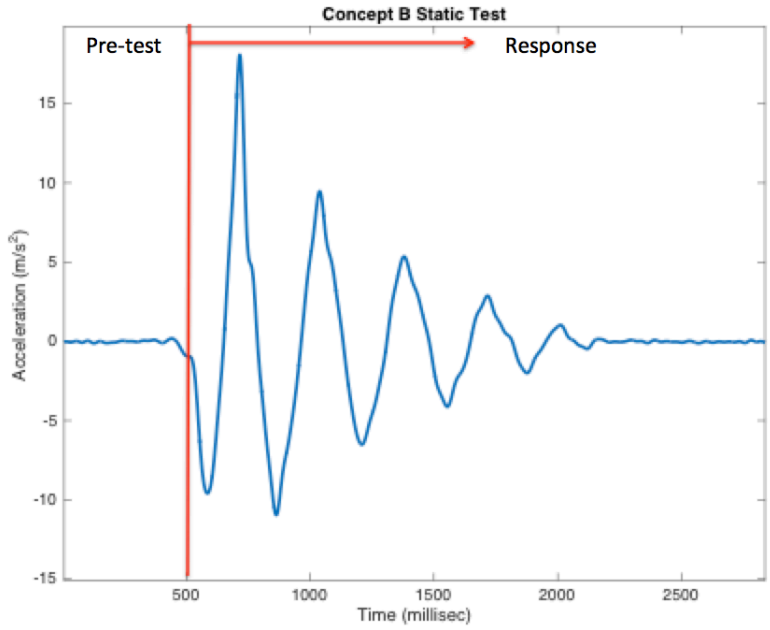

Figure 73. Concept B—Static Test Response

From the plot in Figure 73, one can see that the response for concept B damps the motion induced by the manually applied input in an expected fashion. In comparison to concept A, the response appears to die out more quickly indicating a more dominant viscous damping coefficient in the joint. This is a result of different spring coefficients used in the concept development as well the general design which affects the loading of

the joint bearings. Figure 74 illustrates the response when pressure is applied to the plate and the two-link arm is moved through its trajectory.

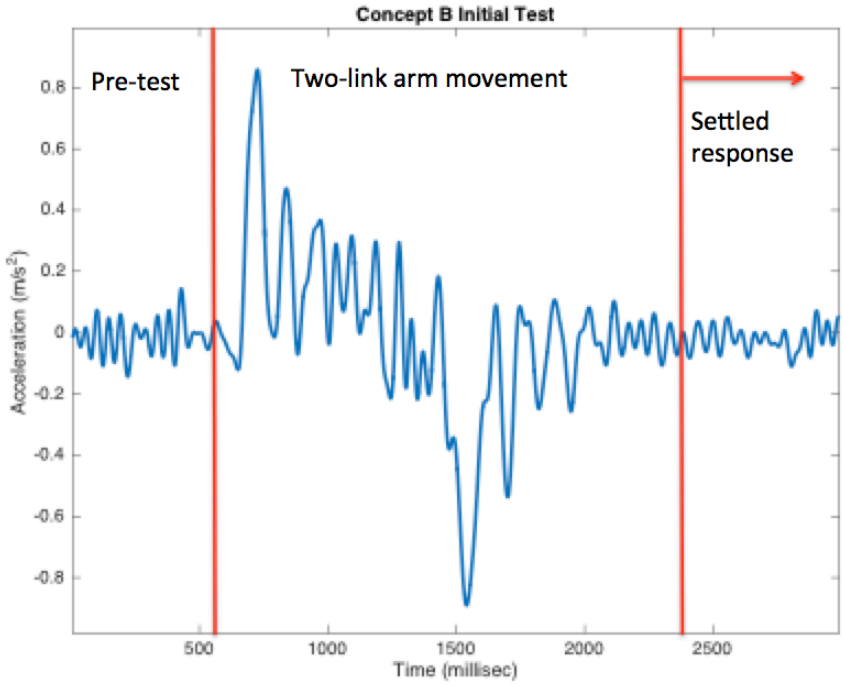

Figure 74. Concept B—Dynamic Test Response

Similar to the results for concept A, this plot demonstrates the initial acceleration of the two-link arm, followed by deceleration when the arm stops, and then a damped response. The damped response, similar to the static response, dies out faster compared to concept A, again due to the difference in springs used and the concept design.

One obvious difference between the results of the static tests compared to the tests where the plate is floating and the robotic arm movement causes the excitation is the observed signal to noise ratio. It was mentioned earlier that during the calibration process, it was not possible to calibrate the accelerometers to zero due to noise inherent in each IMU. This noise is also referred to as sensor drift caused by a small DC bias in the acceleration signal [30]. The amplitude of the static experiments is large in comparison to the floating plate tests; therefore, the noise is not as apparent in the results. The signal to noise ratio is large. When the signal to noise ratio is small on the other hand, it is difficult

to discern the vibrating arm motion from the noise output of the sensor and therefore one can't perform proper analysis of the results. With this said, concepts A and B seem to be inadequate for permanent implementation on the test bed.

C. CONCEPT C—MACHINED TORSION SPRING

Concept C incorporated the helical machined spring as the flexible joint. This concept allows for pure moments and linear rates. As noted in Chapter III.c, the manufacturers produced a torsion spring rate that resulted in a spring that was too stiff to provide the expected displacement. The spring was subsequently trimmed at the NPS machine shop to provide increased deflection. Additionally, 500-gram weights were added to the end of the robotic arm and then vibrating arm to add increased inertia to the system. Adding the weight contributed to an enhanced signal to noise ratio so the experimenter can observe the results more clearly by reducing the noise floor. Figures 75 and 76 show the test bed configured for the final experiments with added weights.

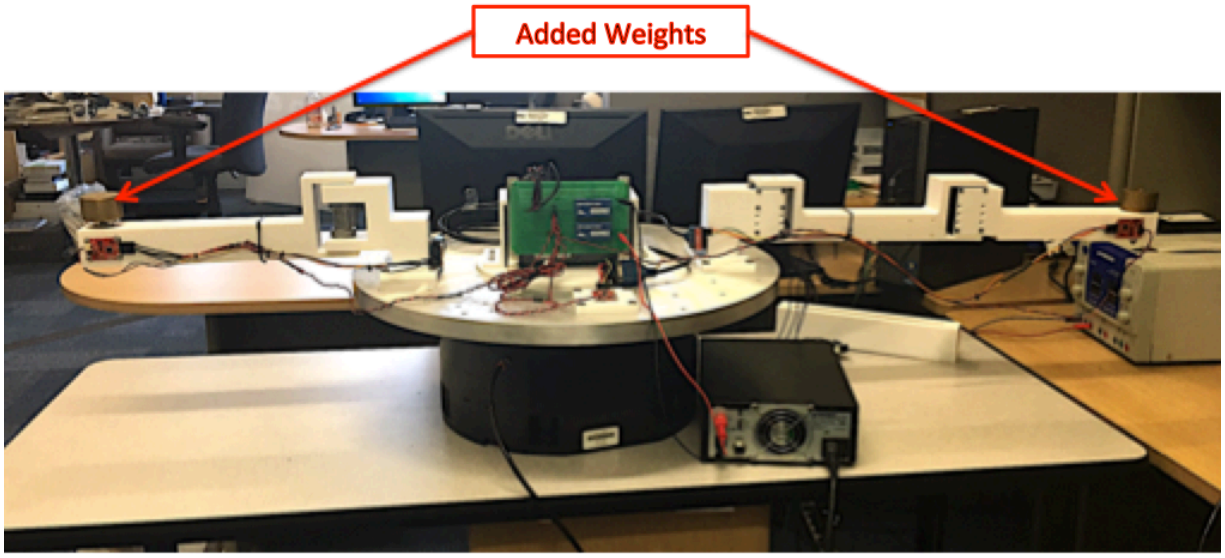

Figure 75. Test Bed Configuration for Final Experiments

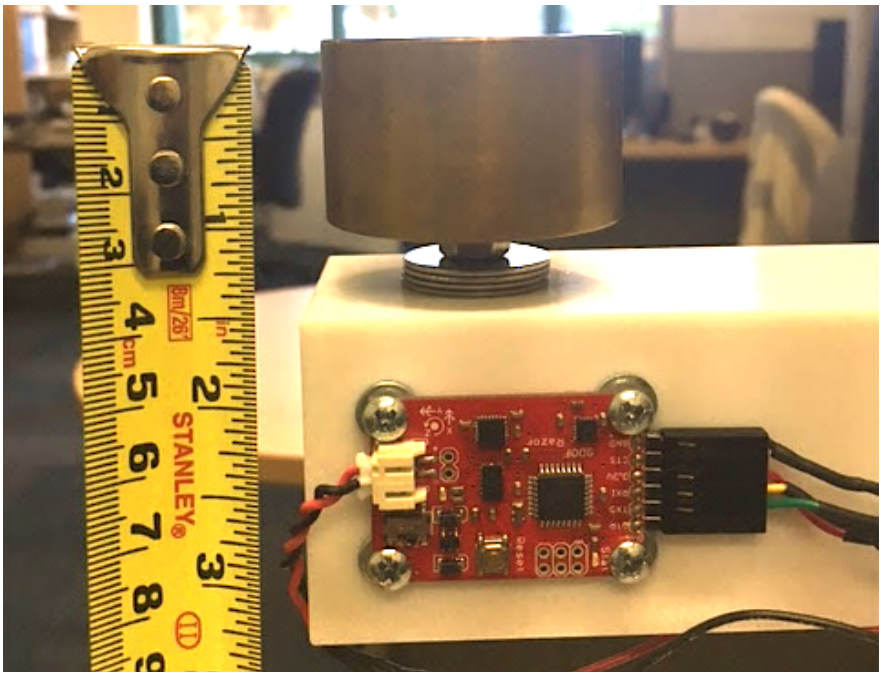

Figure 76. Close-up View of Weight Added to the End of Vibrating Arm

The first step for the final experiments was to upload the Arduino code onto each IMU to display all three components acceleration and gyroscope data. Ultimately only tangential data is needed. However, the plate IMU is laid flat while both arms' IMUs are vertical; therefore the output channels from each IMU are different depending on orientation. Additionally, post-processing the data to determine the gravity component can be used to verify the accuracy of the processed data.

An experiment was conducted by opening the front panel of the LabVIEW program as well as the Robotis Dynamixel Wizard software to control the Dynamixel Pro M42-10-S260-R. Again, just one actuator was functioning for the final experiment; therefore, the two-link robotic arm in effect functioned as a one-link arm. On the LabVIEW front panel, the "run" button was selected then serial data from each IMU began to stream onto the display. The serial data is separated by 20 milliseconds; with this said, in order to reduce the amount of data for post-processing, the robotic arm was quickly commanded to rotate 90 degrees. The robotic arm motion imparts rotational motion on the plate and therefore a vibration of the flexible-joint, vibrating arm. Once the vibrating arm settles, the LabVIEW program is commanded to stop. The data from each

IMU is subsequently written to text files, the data is copied and pasted into Excel, transposed, and finally copied into the MATLAB script detailed in the Appendix.

Experimental Results

Figures 77–79 illustrate the acceleration data from each axis of the respective IMUs. The vibrating arm and robotic arm data sets both illustrate that the axis normal to the plane of motion is the "a1" component and the plate normal acceleration component is "a2" since the data plotted corresponds to gravity in metric units. Additionally, the tangential motion component for each arm is "a2" whereas the plate tangential acceleration component is "a1" by inspection of the sinusoidal behavior of the plots.

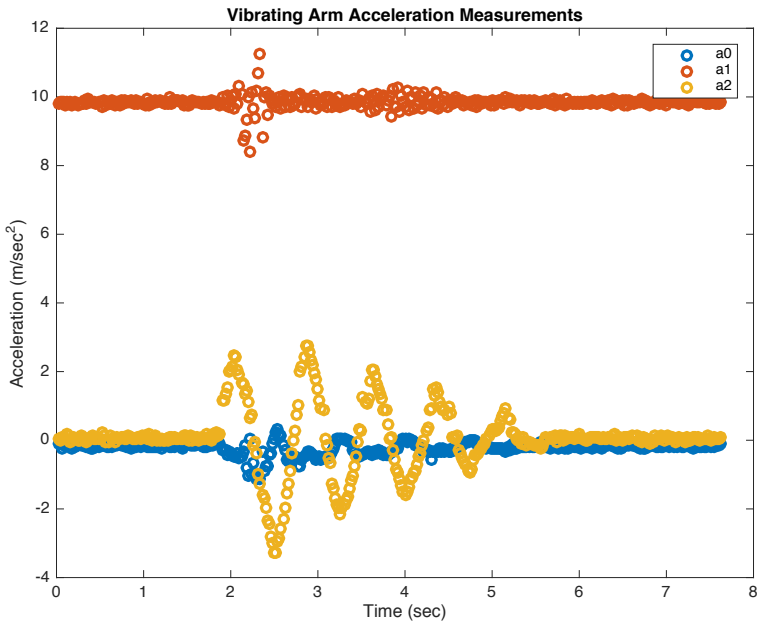

Figure 77. Vibrating Arm Acceleration Measurements

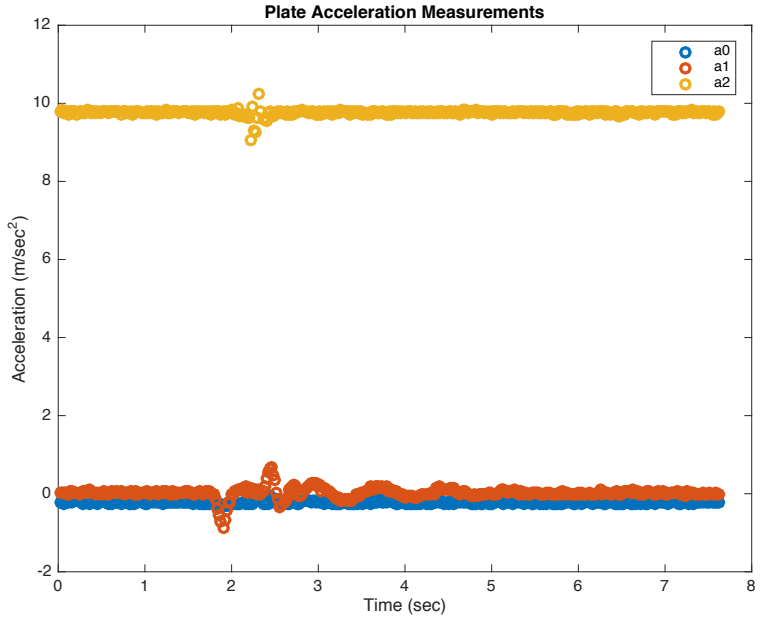

Figure 78. Plate Acceleration Measurements

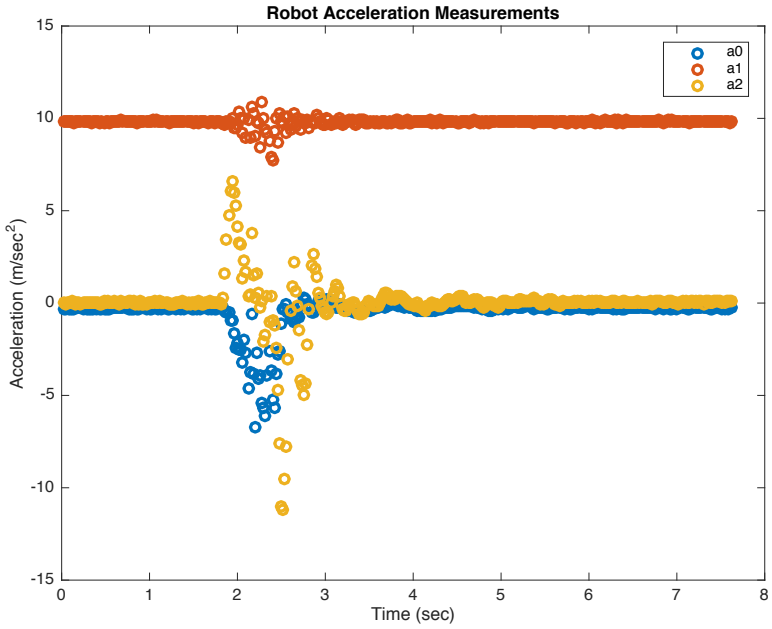

Figure 79. Robot Arm Acceleration Measurements

Figures 80–83 present the rate gyro data for the test. The vibrating arm and robotic arm results illustrates that the desired angular rate component is “g1” and for the plate, the desired angular rate component is “g2”. To note, in both the acceleration and rate plots, one can see how the robotic arm is commanded comparably to a “bang-bang” maneuver. That is, the responses show a rapid application of torque to the system followed by a rapid deceleration. This effect is translated to the plate, and then to the vibrating arm.

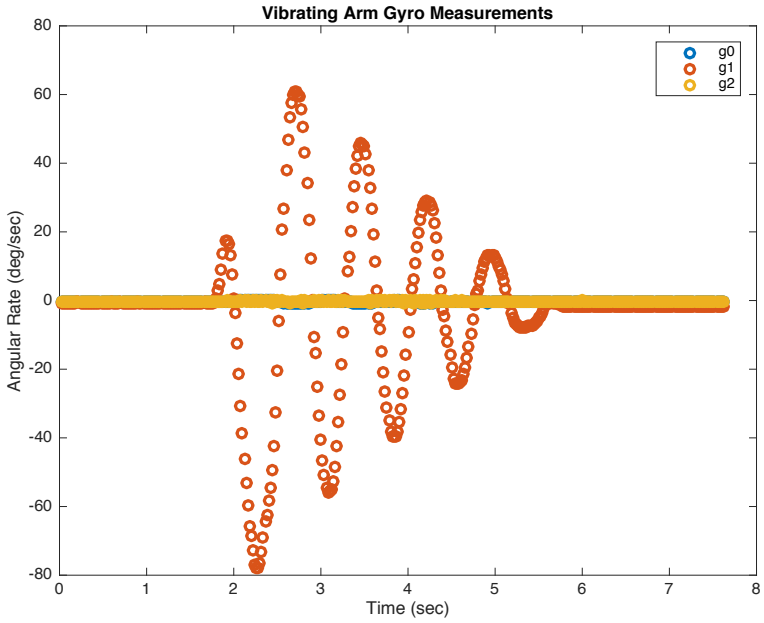

Figure 80. Vibrating Arm Rate Measurements for Each Axis

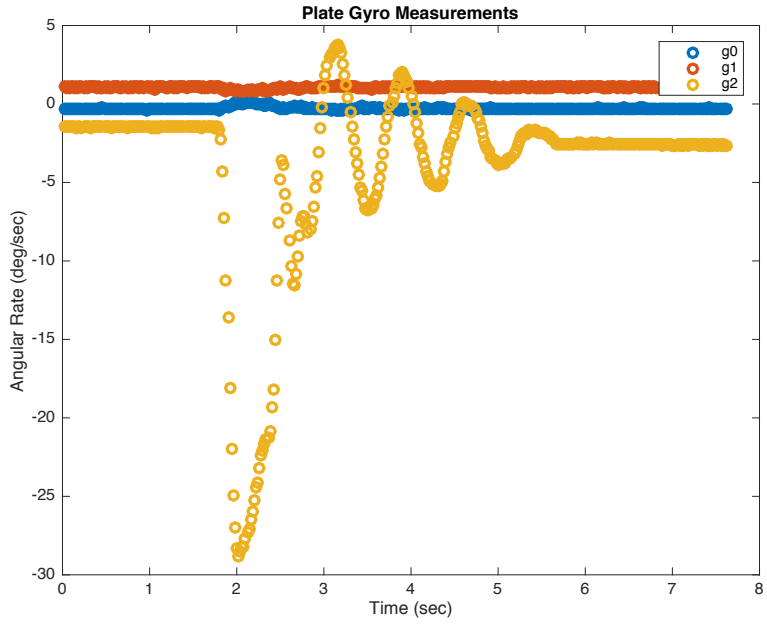

Figure 81. Plate Rate Measurements for Each Axis

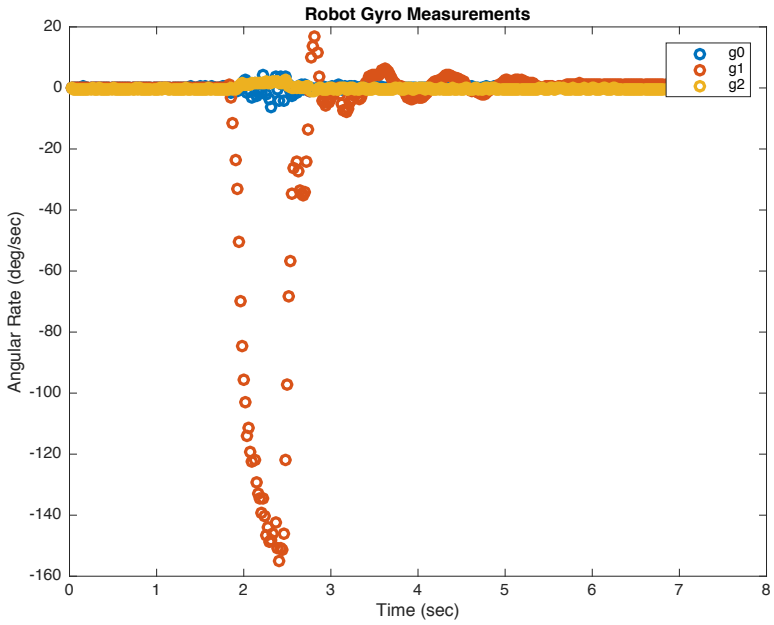

Figure 82. Robot Arm Rate Measurements for Each Axis

Figure 83 compares the vibrating arm angular acceleration response obtained by two methods. The "Alpha from differentiating gyro" is the angular acceleration derived by finite differencing the rate gyro data. The "Raw alpha from accelerometer" trace is simply the tangential acceleration mapped to the vibrating arm's flexible joint by dividing by the distance from the accelerometer to the center of the joint.

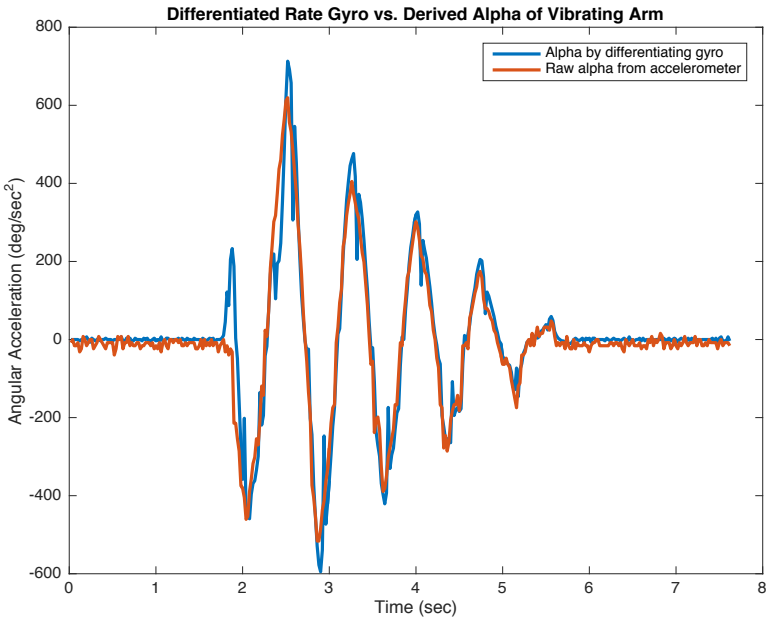

Figure 83. Vibrating Arm Acceleration Determined from Differentiated Rate Gyro Data versus Raw Acceleration Data Mapped to Joint

Figures 84 and 85 represent the angular acceleration for the plate and robot arm, respectively. The plots were derived in the same manner as for the vibrating arm. As noted previously, the rapid on/off application of the robotic arm is easily seen in the angular acceleration plots for both the plate and robot arm. Note also the comparably large angular acceleration magnitude of the robot arm in comparison with the other signals.

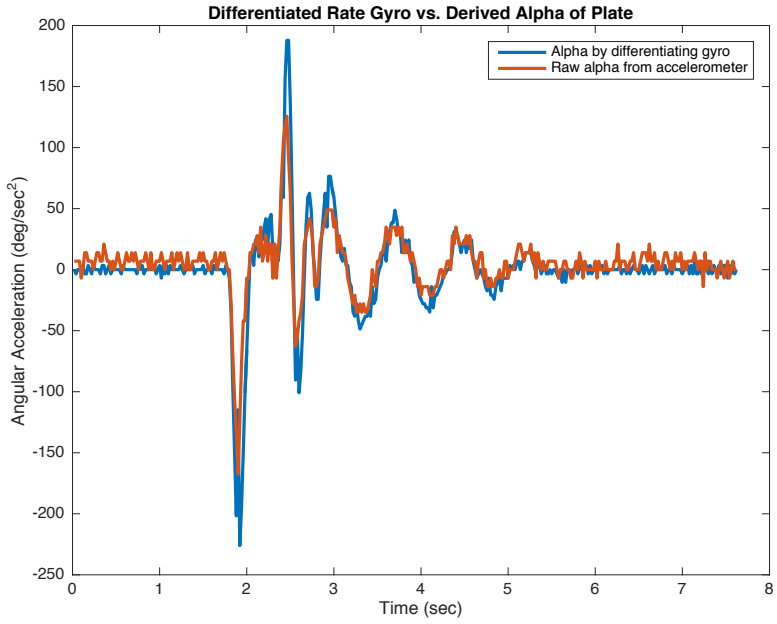

Figure 84. Plate Acceleration Determined from Differentiated Rate Gyro Data versus Raw Acceleration Data Mapped to Joint

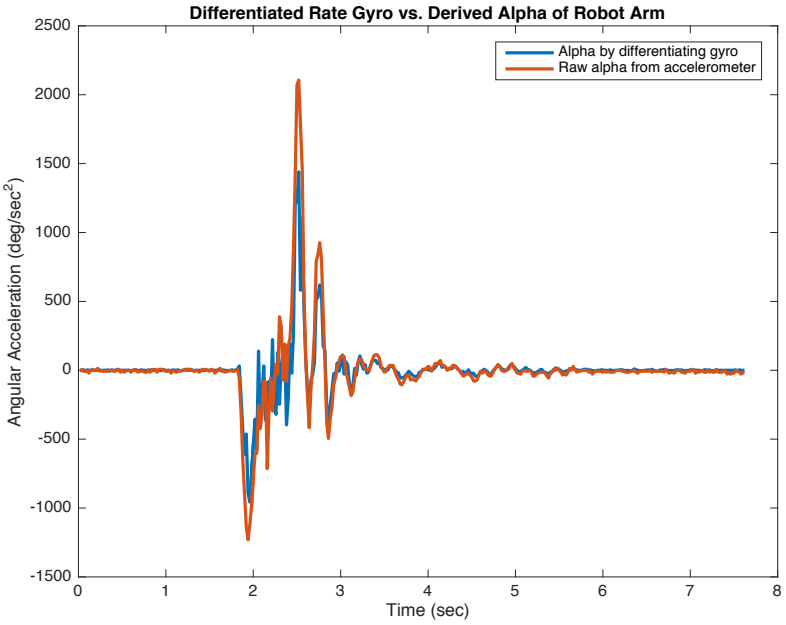

Figure 85. Robot Arm Acceleration Determined from Differentiated Rate Gyro Data versus Raw Acceleration Data Mapped to Joint

Figure 86 illustrates the angular velocity of the vibrating arm derived via three different approaches. The "raw gyro" plot is simply that, the raw data converted into units of °/sec. The "rate from diff gyro" was determined by numerically integrating the finite differenced gyro data noted in Figure 83. The "rate from accel" is the angular acceleration signal derived from the tangential acceleration component numerically integrated. The plot of raw gyro data compared to the "rate from diff gyro" confirms the process of differentiating to derive the angular acceleration and then integrating to derive the angular rate by canceling out the +/- errors of each step in the process. The "rate from accel" plot confirms the discussion in Chapter IV that simply numerically integrating the data without accounting for the inherent bias in the process due to unknown initial conditions will cause the resultant plot to drift.

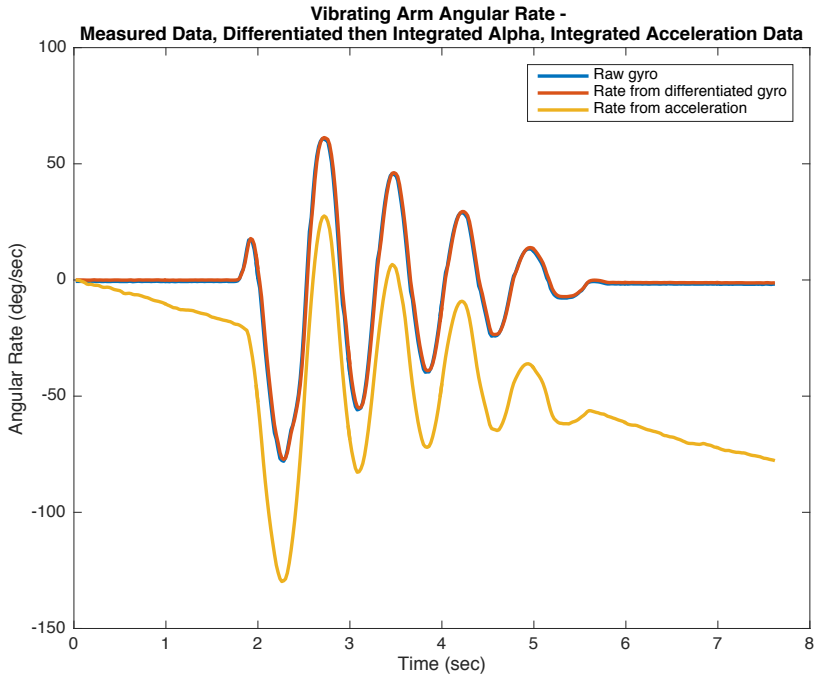

Figure 86. Processing the Vibrating Arm Angular Rate from the Measured Data

Figures 87 and 88 demonstrate similar results for the plate and robot arm. Again, the rapid onset of torque is easily observed in both figures.

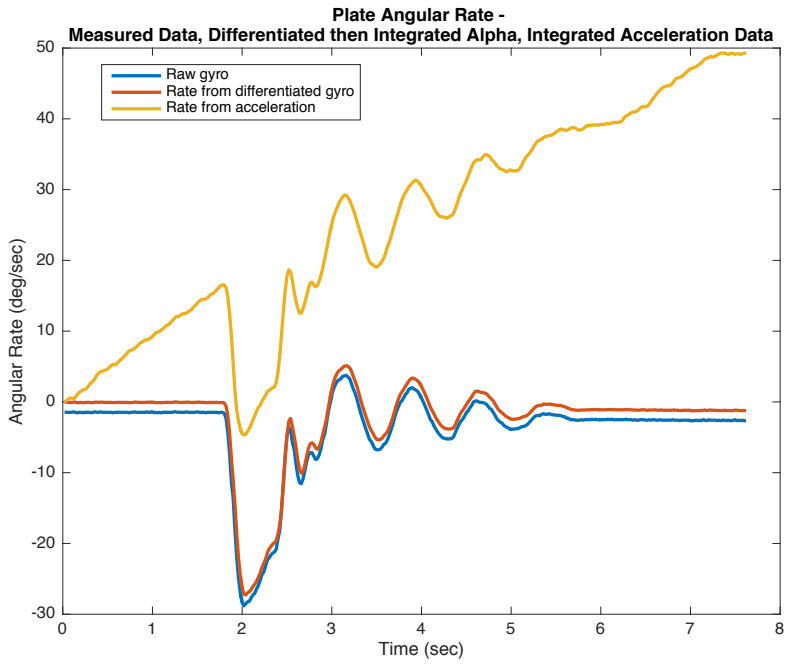

Figure 87. Processing the Plate Angular Rate from Measured Data

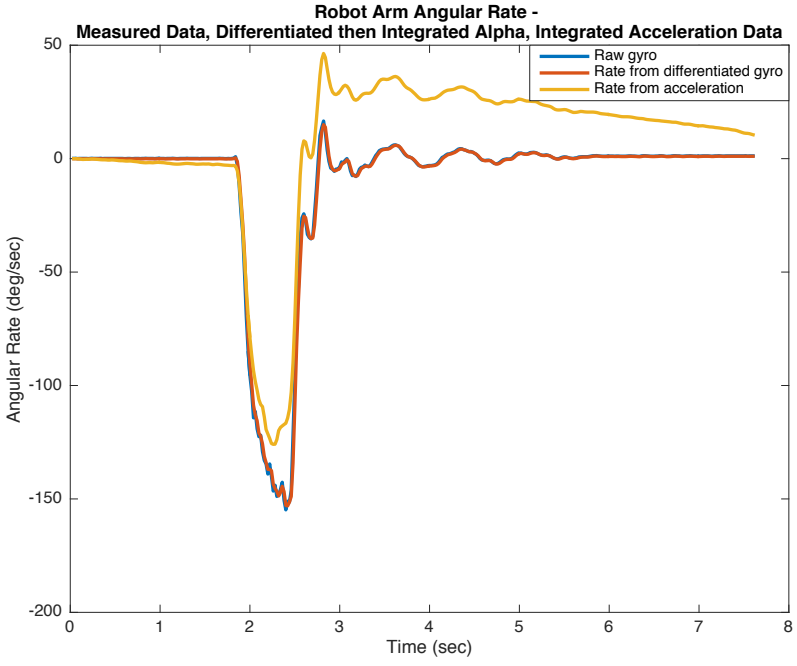

Figure 88. Processing the Robot Arm Angular Rate from Measured Data

Figure 89 illustrates the vibrating motion. The "Pos from raw gyro" plot was determined by numerically integrating the raw gyro data. The "Pos from diff gyro" was determined by numerically integrating the previously mentioned differentiated gyro data that was then integrated to derive the angular velocity. Both methods show comparable results to the actual motion of the system. The motion of the vibrating arm, as seen in Figure 89, represents not just the motion of the arm, but also the motion of the arm plus the motion of the plate. That is, after the robot arm stops, the vibrating arm continues to vibrate imparting a torque onto the plate. However, the plate also drifts while the arm is vibrating. The combined motions explain the drift observed in Figure 89 while the vibrating arm vibrates and subsequently damps out. This point is important to note, as the drift in the "Pos from diff gyro" is the plate actually drifting vice the "drift" as a result of numerical integration. With this said, as will be seen when the vibrating arm and plate motions are separated, the "Pos from raw gyro" plot represents components due to actual plate drift and numerical integration drift.

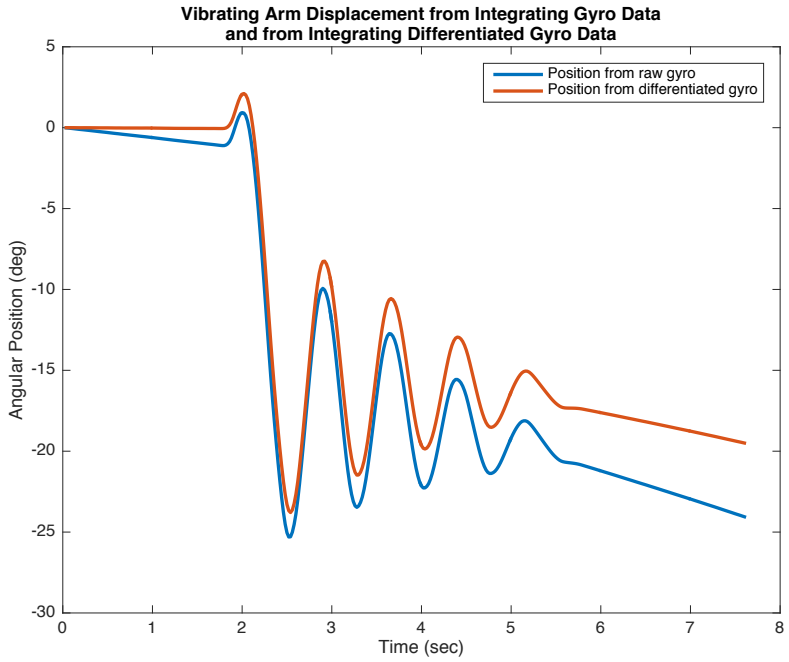

Figure 89. Vibrating Arm Displacement from Integrating Gyro Data and from Integrating Differentiated Gyro Data

Figures 90 and 91 illustrate the motion of the plate and the robot arm derived using the same processing as Figure 89. In Figure 90, the error from numerically integrating the raw gyro data is more apparent as seen in the initial slope of the "Pos from raw gyro" plot. To note, for completeness the acceleration data from each IMU was numerically integrated twice. The results skewed the plots of the previously discussed methods to such a degree that they were unable to be read. That is, the error from integrating the acceleration twice was so large that it overshadowed the other plots, demonstrating that twice integrating accelerometer data is unusable unless one can properly account for the slope and initial conditions associated with each integration.

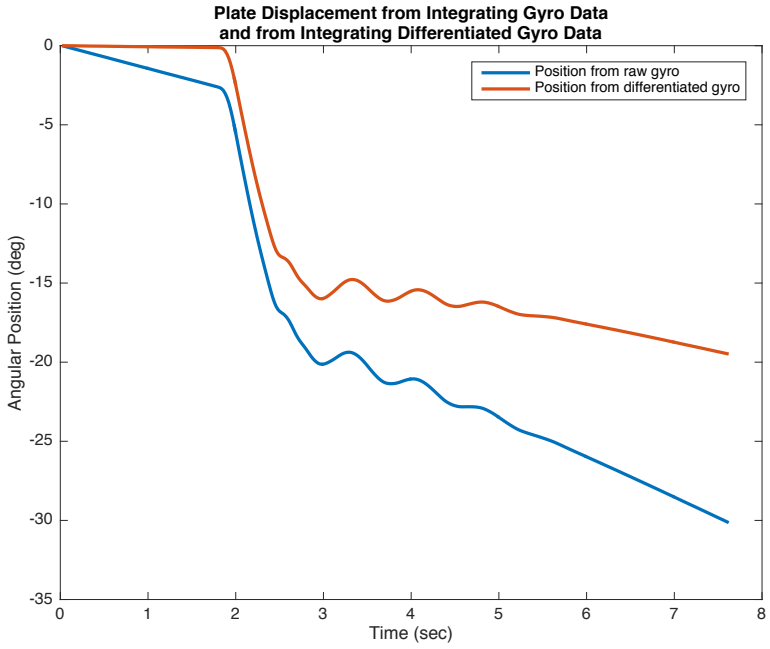

Figure 90. Plate Displacement from Integrating Gyro Data and from Integrating Differentiated Gyro Data

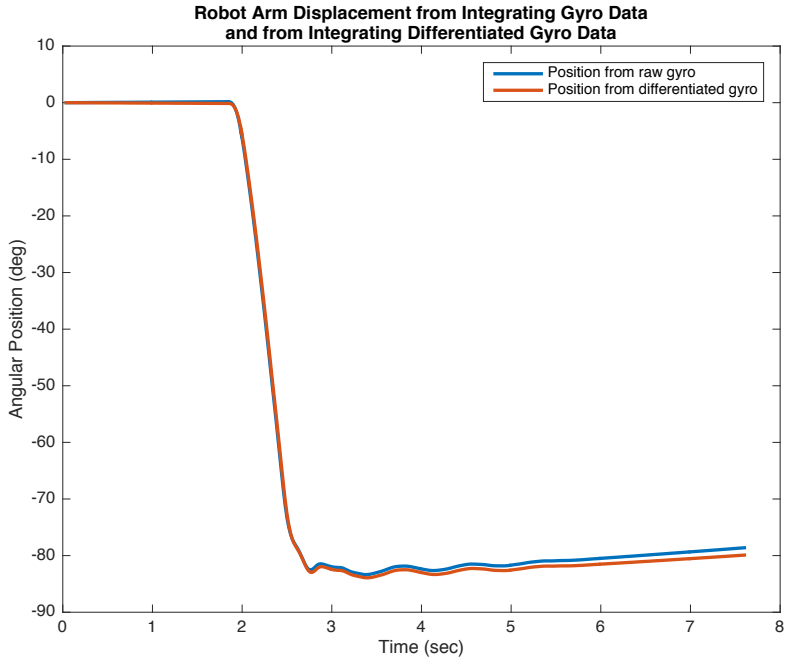

Figure 91. Robot Arm Displacement from Integrating Gyro Data and from Integrating Differentiated Gyro Data

Figure 92 illustrates the results of isolating the vibrating arm motion from the plate motion. The figure demonstrates that differentiating the raw gyro data first then integrating the result twice is clearly the best available method for eliminating the drift error and therefore to isolate the vibrating arm motion. The "Pos from raw gyro" approach still has a bias present after the plate motion is subtracted from the vibrating arm motion due to the differences in error present in this approach. The MATLAB code below demonstrates the simple code used to isolate the vibrating arm motion where "Subtract raw" subtracts the plate displacement determined by integrating the raw gyro data from the vibrating arm raw gyro data. "Subtract accel" performs a comparable operation for the method of differentiating then integrating the raw gyro data.

```
subtract_raw = vib_pos_gyro_raw - plate_pos_gyro_raw;

subtract_accel = vib_pos_rate_gyro_accel - plate_pos_rate_gyro_accel;
```

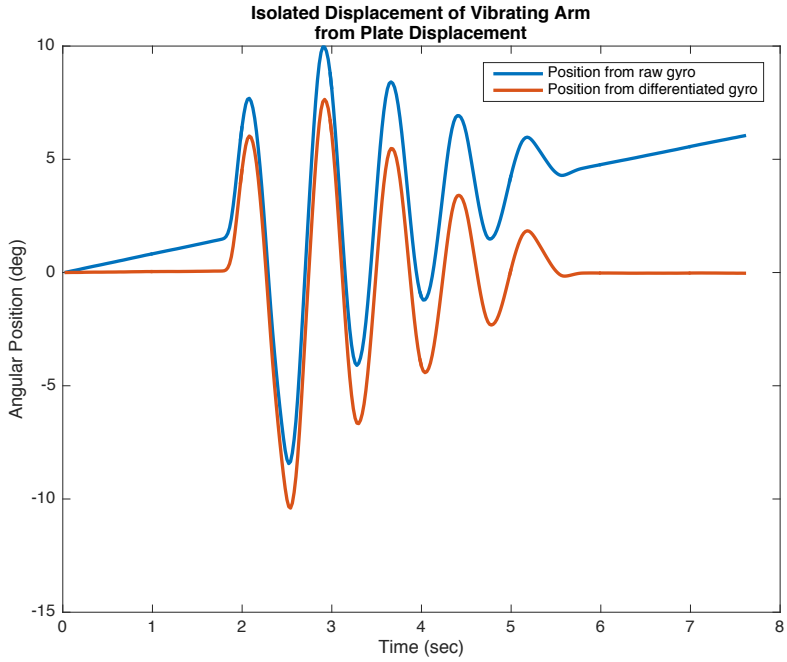

Figure 92. Isolated Displacement of Vibrating Arm from Plate Displacement

D. SUMMARY

This chapter illustrated that the three concepts developed for implementing a flexible joint built upon themselves in a crawl, walk, run fashion to produce a viable design and flexible joint mechanism to support future experiments. The helical torsional spring ultimately did produce the desired deflection. Additionally, the developed post-processing for the IMU data was able to accurately isolate and depict the actual motion of the system observed during the experiments.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

A. CONCLUSION

The goal of this thesis was to develop multiple concepts for a flexible joint motion simulator, from simple to more complex, to provide a basis for laboratory experimentation on advanced algorithms. As unique or clever a design concept may be, proper effort needs to be placed into the sensing processes such that useful data can be extracted from experiments. Ultimately three mechanism concepts were investigated. Concepts A and B were preliminary ideas developed to test the basic functionality of the air-bearing test bed, test for the expected motion of the system as a whole, and to develop initial post-processing data analysis work flows. The responses of concepts A and B in both the static and dynamic testing proved to be far from ideal. Because the deflection amplitude was small, the noise from the IMUs was more apparent. Concept C proved to be the best design. Concept C's flexible joint incorporated a helical torsion spring allowing linear rates and pure moments to be realized. Once concept C was assembled, it was apparent that good signal to noise ratios could be obtained. Experiments were performed by incorporating the NI cRIO and refined post-processing steps in order to measure link flexible effects. The responses in all cases were sufficient to provide an understanding of spacecraft dynamics and vibration analysis when the motion of one body excites other components of the spacecraft. Additionally, the efforts made to this point serve to provide the basis necessary to proceed with more complex thesis work incorporating optimal control theory. Future work will be aimed at finding optimal robot arm motions that can be accomplished in minimal time and/or with minimal effort, that also reduce the induced vibrations of other attached antenna linkages. To do this, a model of the system as a whole should be considered.

B. FUTURE WORK

The developed test bed provides opportunities for a variety of flexible space system experiments. Through optimal control applications, one can research the effects of robotic arm motion on the behavior of flexible appendages.

Further development of the sensing processes and software environment should be made to better understand the responses of the system. Furthermore, understanding of the Robotis software used to command the Dynamixel actuators would allow one to implement optimal control trajectories to the two-link robot with the goal to minimize the amount of vibration on the flexible joint link.

Additionally, the plate was created to allow for three stanchions to be assembled to the plate and then attach a second level for additional components. With pressure applied to the plate and when the two-link arm was displaced by a commanded trajectory, the base plate moved in the opposite direction of the arm displacement, as one would expect. This fact is noted as a complexity of spacecraft dynamics. As a satellite is floating through space, movements from attached bodies are felt throughout the spacecraft. The complexity of incorporating momentum exchange devices to counter this movement is not lost on the author. The laboratory test bed that has been built can provide a means to incorporate such devices on future theses. As well, the NI cRIO has 12 more ports for many other devices to be incorporated into the test bed. Furthermore, the laboratory setup can be evolved using Wi-Fi components. Wireless communication between the sensors and the working computer will eliminate the issue of hanging cables imparting torques on to the test bed.

Until that end, the efforts made so far in the build, experimentation, and understanding of flexible spacecraft systems have laid the groundwork for more complex and integrated experiments in this laboratory environment. In all, the possibilities that this test bed provides are very exciting and ready for the next student!

APPENDIX. EXAMPLE DATA PROCESSING CODE

In this appendix, numerical integration code was created to attempt to mimic the numerical integration function of Matlab's "cumtrapz" function. The displayed plots compare the results of the two methods.

Due to the size/length of the time, acceleration, and gyroscope data arrays, they were deleted from the appendix. A "*" has been added to highlight the change. This data changes with each experiment; therefore, the critical piece to the coding is still represented.

```
% Simple model to show problems with double integrating acceleration.
% This code performs numerical integration of experimental data and
% compares it to Matlab's cumtrapz function. As well, a simple Simulink
% model was created to show the results from passing a sine input
through a
% double integrator compared to passing the same simple sine wave
through
% Matlab's cumptrapz function.
% CDR Martin Griggs
% Thesis - 31 Oct 2016

clear all; close all;
%%% Run Simulink model  %%%
[tout, yout, output]=sim('test3');

%% Experiment Data analized using numerical integration code
vib_accel_0 = [*];
vib_accel_1 = [*];
vib_accel_2 = [*];

vib_accel_0 = vib_accel_0*1/225*9.81;   %for +/- 16g full scale, we
should divide by 256 LSB/g
                                        %by instead we divide by 225 which
gives 1g
                                        %in "z" direction per the "raw" data
                                        %(see also ADXL345 product
specification)
vib_accel_1 = vib_accel_1*1/225*9.81;
vib_accel_2 = vib_accel_2*1/225*9.81;

time = [*];
dt = 0.02;
time = time(1:381);
```

```
length_vib = length(vib_accel_2);

% Vibrating Arm Velocity
vib_vel(1) = 0.5*(vib_accel_2(1) + vib_accel_2(2))*dt;
for i = 1:length_vib-3;
    vib_vel(2) = 0.5*(vib_accel_2(2) + vib_accel_2(3))*dt + vib_vel(1);
    vib_vel(i+2) = 0.5*(vib_accel_2(i+2) + vib_accel_2(i+3))*dt +
vib_vel(i+1);

end
v_size = size(vib_vel);
vib_vel(v_size(2) + 1) = vib_vel(end);

% Vibrating Arm Displacement
vib_displ(1) = 0.5*(vib_vel(1) + vib_vel(2))*dt;
for j = 1:length_vib - 3;
    vib_displ(2) = 0.5*(vib_vel(2) + vib_vel(3))*dt + vib_displ(1);
    vib_displ(j+2) = 0.5*(vib_vel(j+2) + vib_vel(j+3))*dt +
vib_displ(j+1);

end

d_size = size(vib_displ);
vib_displ(d_size(2) + 1) = vib_displ(end);

%% Experimental Data using cumtrapz - only accounting for tangential
acceleration (accel [2])
vib_vel_cum = cumtrapz(time, vib_accel_2);
vib_pos_cum = cumtrapz(time, vib_vel_cum);

%% Plots for numerical versus cumtrapz
figure
subplot(2, 1, 1)
plot(time, vib_vel, time, vib_vel_cum, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Velocity (meters/sec)')
legend('Numerical Integration Code', 'Cumtrapz')
title({'Experimentally Derived Velocity Analysis -'; 'Numerical vs.
Cumtrapz'})

subplot(2, 1, 2)
plot(time, vib_displ, time, vib_pos_cum, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Position (meters)')
legend('Numerical Integration Code', 'Cumtrapz')
title({'Experimentially Derived Position Analysis -'; 'Numerical vs.
Cumtrapz'})
%% Cumtrapz simple sine wave
c = 0:0.1:20;
acceleration_cum = sin(c);
velocity_cum  = cumtrapz(c, acceleration_cum);
position_cum = cumtrapz(c, velocity_cum);

%% Plots of simple sine wave passed through simulink and corrected for
```

```
intital conditions compared to
% cumtrapz
figure
subplot(3, 1, 1)
plot(tout, acceleration, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Acceleration (rads/sec^2)')
title('Acceleration')
subplot(3, 1, 2)
plot(tout, velocity, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Velocity (rads/sec)')
title('Velocity')
subplot(3, 1, 3)
plot(tout, acceleration, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Position (rads)')
title('Position')

figure
subplot(3, 1, 1)
plot(c, acceleration_cum, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Acceleration (rads/sec^2)')
title('Acceleration')
subplot(3, 1, 2)
plot(c, velocity_cum, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Velocity (rads/sec)')
title('Velocity')
subplot(3, 1, 3)
plot(c, position_cum, 'Linewidth', 2)
xlabel('Time (sec)')
ylabel('Position (rads)')
title('Position')
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Vibration. (2016, May 10). Encyclopedia Britannica. [Online]. Available: https://www.britannica.com/science/vibration

[2] Tuned mass dampers in skyscrapers. (2016, Feb. 14). Practical Engineering. [Online]. Available: http://practical.engineering/blog/2016/2/14/tuned-mass-dampers-in-skyscrapers

[3] T. Harris. (n.d.). How skyscrapers work. [Online]. Available: http://science.howstuffworks.com/engineering/structural/skyscraper4.htm. Accessed Nov. 1, 2016.

[4] Tracking and Data Relay Satellite (TDRS). (n.d.). NASA. [Online]. Available: http://www.nasa.gov/directorates/heo/scan/services/networks/txt_tdrs.html

[5] Capabilities and services. (n.d.). SpaceEx. [Online]. Available: http://www.spacex.com/about/capabilities. Accessed Nov. 1, 2016.

[6] E. Johnson. (2015, Feb. 9). Rockets: What is cost of sending 1kg weight into space? [Online]. Available: https://www.quora.com/Rockets-What-is-cost-of-sending-1-kg-weight-into-space

[7] S.W. Wojdakowski, "Rapid slewing of flexible space structures," M.S. thesis, Dept. of Mech. and Aero. Eng., Naval Postgraduate School, Monterey, CA, 2015.

[8] A.G. Sears. "Design and experimental implementation of optimal spacecraft antenna slews," M.S. thesis, Dept. of Mech. and Aero. Eng., Naval Postgraduate School, Monterey, CA, 2013.

[9] Dynamics text, ch. 9, 13. (n.d.). [Online]. Available: http://robotics.itee.uq.edu.au/~metr4202/2012/T4_solutions_with_attachments.pdf. Accessed Dec. 1, 2015.

[10] *Quick Start for Dynamixel Pro*, Robotis, (n.d.), pp. 4–52. [Online]. Available: http://www.robotis.com/download/doc/Dynamixel_pro/Dynamixel-Pro%20Quick%Start_en.pdf. Accessed Jan. 5, 2016.

[11] I.M. Ross, *A Primer on Pontryagin's Principle in Optimal Control*, Carmel, CA: Collegiate Publishers, 2009.

[12] I.M. Ross and M. Karpenko. "Constrained dynamic optimization," class notes for AE4850: Astrodynamic Optimization, Dept. of Mech. and Aero. Eng., Naval Postgraduate School, Monterey, CA, 2015.

[13] Ellisar Global, Carmel, CA. (1998) *DIDO*. [Online]. Available: http://www.elissarglobal.com

[14] *Operating Manual for the NAC-R-301L-ME-01 12" Rotary Air Bearing Motorized with Encoder Manual Rev 00*, Nelson Air Corp., Milford, NH, p. 4.

[15] I. Akyuz. (2013, Feb. 19). Single link flexible joint robot manipulator. [YouTube video]. Available: https://www.youtube.com/watch?v=-3t2MV8p39A Accessed Jan. 5, 2016.

[16] FRP54-I110K Set. (n.d.) Robotis. [Online]. Available: http://www.robotis.us/frp54-i110k-set/. Accessed Nov.09, 2016.

[17] Helical Machined Springs. (n.d.). Helical Products Company. [Online]. Available: http://machinedsprings.com. Accessed Jan. 5, 2016.

[18] P. Bartz. (2016, April). Building an AHRS using the Sparkfun "9DOF Razor IMU" or "9DOF Sensor Stick." Sparkfun Electronics Corp. Niwot, CO. [Online]. Available: https://github.com/ptrbrtz/razor-9dof-ahrs-/wiki/Tutorial

[19] *Razor Accelerometer (ADXL345) Specifications,* Analog Devices, Inc., Norwood, MA, 2009. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf

[20] *ITG-3200 Product Specification Revision 1.4*, InvenSense, Boston, MA, 2010, p 7.

[21] What is a breakout board for Arduino? (n.d.). Programming Electronics Academy. [Online]. Available: https://programmingelectronics.com/what-is-a-breakout-board-for-arduino/. Accessed Nov. 8, 2016.

[22] Arduino Worldwide, 2016. (2016). Arduino Pro, 1.6.9. [Online]. Available: https://www.arduino.cc/en/Main/Software. Accessed Nov. 11, 2016.

[23] *3-Axis, +/- 2g/+/-4g / +/-8g/+/-16g Digital Accelerometer*, ADXL345, Analog Devices, Norwood, MA, , 2009, p 3.

[24] *3-Axis Digital Compass IC HMC5883L*, Honeywell, Plymouth, MN, 2011, p. 2.

[25] *NI cRIO-9024 User Manual and Specifications*, National Instruments Corporation, Austin, TX, 2015. [Online]. Available: http://www.ni.com/pdf/manuals/375233f.pdf

[26] LabVIEW System Design Software. (n.d.). National Instruments Incorporated. [Online]. Available: http://www.ni.com/labview/. Accessed Jul. 19, 2016.

[27] NI 9870 – Continuous serial read – On – Demand write (2011, Jun. 23). National Instruments Incorporated. [Online]. Available: http://www.ni.com/example/31366/en/

[28] A. Martinez, "Robotic arm movement optimization for satellites," slides 6, 9 LabVIEW Data Collection. Unpublished.

[29] *Cumtrapz – Cumulative Trapezoidal Numerical Integration.* MathWorks, Inc, Torrance, CA., 2016.

[30] L.D. Slifka, "An accelerometer based approach to measuring displacement of a vehicle body," M.S. thesis, Dept. Elect. Eng., Univ. of Michigan, Dearborn, MI, 2004.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California