Digital Rotating Unbalance Identification and Parametric Determination of Counterbalance Placement for Predictable Dynamic Behavior.

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Maura O’Neill

Graduate Program in Mechanical Engineering

The Ohio State University

2017

Master’s Examination Committee:

Professor Sandra Metzler, Advisor, and Professor Blaine Lilly
Abstract

There are increasingly many situations where the art and engineering worlds overlap. This is particularly true in the art installations created by Sarah Oppenheimer’s Folding Enterprises where engineering analyses were used to evaluate the structural integrity and behavior of a dynamic glass and aluminum rotational museum installation. This structure presented a particular challenge in that its rotational axis was only connected to two outer surfaces on either side of a hollow, 52 ft³ volume of glass and aluminum, introducing the risk of axis mis-alignment and unpredictable rotational behavior. The lack of predictability in the rotational behavior of these kinetic installations poses a danger to museum inhabitants and detracts from Oppenheimer’s design intent.

This study was performed to specifically address the dynamic behavior surrounding the rotational equilibrium of these kinetic installations. The dynamics of the system were described through Lagrangian mechanics and simulated numerically as a rotating machine with a static unbalance. The angular motion of the model was recorded with a 6-axis inertial measurement unit supported by an Arduino Board 101. A nonlinear least squares regression method was implemented within a grey-box system identification to estimate the parameters of static unbalance in the system. A numerical algorithm implemented in MATLAB determined the appropriate counterbalance sizes and locations.
to selectively alter the center of gravity of the system and, as a result, shift the rotational equilibrium positions of the system of interest.
Vita

June 2011 ............................................. Kettering Fairmont High School

December 2015 ............................................ B.S. Mechanical Engineering, The Ohio State University

January 2016 to present Graduate Teaching Associate, Department of Mechanical Engineering, The Ohio State University

Field of Study

Major Field: Mechanical Engineering
# Table of Contents

Abstract.............................................................................................................................................ii

Vita....................................................................................................................................................iv

Table of Contents ...............................................................................................................................v

List of Tables .......................................................................................................................................vii

List of Figures ......................................................................................................................................ix

I. Introduction ..................................................................................................................................1

   A. Purpose......................................................................................................................................1

   B. Relevant Terminology ...............................................................................................................7

II. Literature Review .........................................................................................................................10

   A. Rotating Unbalance ..................................................................................................................10

   B. Madgwick Signal Filtering .......................................................................................................16

   C. Parameter Estimation ..............................................................................................................18

III. Experimental Methods ...............................................................................................................20

   A. Preliminary Derivations, Etc. .................................................................................................20

   B. Measurement Instrumentation ...............................................................................................34

   C. Test Fixture Design ..................................................................................................................38

   D. Data Analysis ...........................................................................................................................47

   E. Equilibrium Adjustment ..........................................................................................................55

IV. Results ..........................................................................................................................................61

   A. Verification and Simulation.......................................................................................................61
List of Tables

Table 1. Types of system identification. .................................................................18

Table 2. Relevant variables. ......................................................................................21

Table 3. Needle roller bearing mechanical specifications. ........................................40

Table 4. Thrust bearing specifications. ........................................................................45

Table 5. Masses of frame and shaft sub-assembly components...............................51

Table 6. Frame and shaft components’ mass moments of inertia. ..............................52

Table 7. Grey-box estimation settings. .........................................................................54

Table 8. Simulation parameter values. ..........................................................................62

Table 9. Scenario-specific details of applied eccentric mass. ......................................68

Table 10. Estimated damping constants. ......................................................................71

Table 11. Nonlinear grey-box free parameter estimates with incorrect algorithm. ..77

Table 12. Nonlinear grey-box free parameter estimates with correct algorithm. ..78

Table 13. Counterbalance specifications for +2 [deg] equilibrium adjustment. ..........80

Table 14. Counterbalance specifications for +5 [deg] equilibrium adjustment. ..........82

Table 15. Analysis of nonlinear grey-box estimation free parameter. .......................85

Table 16. Percent error of parameter estimates..........................................................86
List of Figures

Fig. 1. S-337473, 2017. Installation view: Wexner Center for the Arts. USA. 2017.

Fig. 2. Internal hardware of S-337473. (a) Far view. (b) Close view. (c) Detailed view of internal hardware.

Fig. 3. Complete Solidworks assembly of experimental fixture.

Fig. 4. Frame sub-assembly.

Fig. 5. (a) Static unbalance. (b) Couple unbalance. (c) Quasi-static unbalance. (d) Dynamic unbalance.

Fig. 6. Static unbalance of rotor with concentrated mass unbalance.

Fig. 7. Static unbalance of eccentric rotor.

Fig. 8. Simplified balancing machine model.

Fig. 9. Diagram of frame with Cartesian coordinate system.

Fig. 10. Rotation of \{e_1,e_2,e_3\} about the e_1 axis.

Fig. 11. Rotation of \{e_1’,e_2’,e_3’\} about the e_3’ axis.

Fig. 12. Datum used for definition of potential energy.

Fig. 13. Rotation of eccentric mass about e_s axis.
Fig. 14. (a) InvenSense MPU-6050. (b) Assembled Data Logger Shield and Arduino Uno.

Fig. 15. (a) Arduino Board 101. (b) HC-05 Bluetooth module.

Fig. 16. Solidworks rendering of test fixture.

Fig. 17. Shaft sub-assembly.

Fig. 18. Frame sub-assembly.

Fig. 19. (a) Mounting sub-assembly. (b) Mounting sub-assembly on test fixture.

Fig. 20. Degrees of freedom of counterbalance placement in U-channels.

Fig. 21. Thrust bearings inserted between lower shaft and mounting hardware.

Fig. 22. Final test fixture assembly.

Fig. 23. Angular displacement and filtered angular velocity.

Fig. 24. Angular displacement curve maxima and damped frequency.

Fig. 25. Exponential curve determined by the logarithmic decrement method.

Fig. 26. Rotation of counterbalance mass about rotational axis.

Fig. 27. Gravity force magnitudes during simulated rotation.

Fig. 28. Angular displacement curves for simulated model.

Fig. 29. Final resting position of frame and eccentric mass with defined parameters.

Fig. 30. Side view of system’s final resting position.
Fig. 31. Simulation results for $\phi=\pi/2$ [rad]. (a) Visualization. (b) Angular displacement curves.

Fig. 32. Simulation results for $\phi=3\pi/2$ [rad]. (a) Visualization. (b) Angular displacement curves.

Fig. 33. Transient angular displacement of system in Variation A.

Fig. 34. Exponential damping curve for data set 1.

Fig. 35. Exponential damping curve for data set 2.

Fig. 36. Exponential damping curve for data set 3.

Fig. 37. Exponential damping curve for data set 4.

Fig. 38. Exponential damping curve for data set 5.

Fig. 39. Nonlinear grey-box estimation for data set 1 with incorrect algorithm.

Fig. 40. Nonlinear grey-box estimation for data set 1 with correct algorithm.

Fig. 41. Nonlinear grey-box estimation for data set 2 with incorrect algorithm.

Fig. 42. Nonlinear grey-box estimation for data set 2 with correct algorithm.

Fig. 43. Nonlinear grey-box estimation for data set 3 with incorrect algorithm.

Fig. 44. Nonlinear grey-box estimation for data set 3 with correct algorithm.

Fig. 45. Nonlinear grey-box estimation for data set 4 with incorrect algorithm.

Fig. 46. Nonlinear grey-box estimation for data set 4 with correct algorithm.
Fig. 47. Nonlinear grey-box estimation for data set 5 with incorrect algorithm. ............76

Fig. 48. Nonlinear grey-box estimation for data set 5 with correct algorithm. ............76

Fig. 49. Comparison of nonlinear grey-box estimations produced by incorrect algorithm for all data sets. ..........................................................................................................................79

Fig. 50. Comparison of nonlinear grey-box estimations produced by correct algorithm for all data sets. ..........................................................................................................................79

Fig. 51. +2 [deg] rotational equilibrium adjustment for data set 1. ...............................81

Fig. 52. +5 [deg] rotational equilibrium adjustment for data set 1. ...............................82

Fig. 53. Circuit schematic of HC-05 Bluetooth Module and Arduino Board 101. ...........104

Fig. 54. Dimensioned drawing of T-slotted framing. ......................................................106

Fig. 55. Dimensioned drawing of diagonal T-slotted frame piece. ...............................107

Fig. 56. Dimensioned drawing of T-slotted framing corner bracket. .............................108

Fig. 57. Dimensioned drawing of T-slotted framing attachment hardware. .....................109

Fig. 58. Dimensioned drawing of needle roller bearing. ..............................................110

Fig. 59. Dimensioned drawing of stationary tube. .......................................................111

Fig. 60. Dimensioned drawing of shaft. .......................................................................112

Fig. 61. Dimensioned drawing of shaft attachment piece. ............................................113

Fig. 62. Dimensioned drawing of glass. ......................................................................114
Fig. 63. Dimensioned drawing of U-channel.................................................................115

Fig. 64. Dimensioned drawing of T-slotted counterbalance mounting bar..............116

Fig. 65. Dimensioned drawing of battery and BLE holder........................................117

Fig. 66. Dimensioned drawing of electronics shaft mount........................................118

Fig. 67. Exploded view of test fixture........................................................................120

Fig. 68. Frame sub-assembly.....................................................................................157

Fig. 69. Dimensioned U-channel cross section. Dimensions in [in]........................158
I. Introduction

Visual arts is often perceived as a static art discipline, stationary in the physical sense and limited to classic mediums such as ceramics or canvas art. Sarah Oppenheimer’s Folding Enterprises is shattering this perception with exhibits that walk the line between sculpture and machine “just as Oppenheimer herself is constantly switching through a complex field of media formats [1].” These exhibits require collaborations between fine arts and technology that are prompting the application of classical engineering concepts in new and interesting ways.

A. Purpose

The installation of particular interest for this study, S-337473 at The Ohio State University's Wexner Center for the Arts, required not only Sarah’s creative vision and curiosity about the perception of spatial division but also a tremendous amount of engineering design and analyses to ensure a robust structure and machine. This installation consisted of two large glass and aluminum frames that rotated about discontinuous axes mounted at an angle of 45° from the museum floor. These frames were designed and adjusted as necessary to rest at their respective horizontal and vertical positions or, technically speaking, their rotational equilibrium positions. For clarity, Fig. 1 shows a photograph of the S-337473 installation in the Wexner Center [2].
Sarah often explores the division of space with her work and referred to these kinetic frames as “switches” that served as “…an intervention that fundamentally challenge[d] the conception of architecture as a solid and durable construction, proposing instead a conception of architecture as a controlled and controllable environment [1].”

Although any rotating machine is subject to some amount of unbalance due to manufacturing tolerances and assembly, this exhibit posed additional concern because of the lack of continuity in the rotational axes. Aesthetically speaking, this discontinuity was a visual treat for viewers such that “as the glass panels pivot[ed] through space, reflecting and refracting light, they seem[ed] to rotate around nothing at all [1].” Technically speaking, these discontinuities contributed to a lack of predictability in the equilibrium behavior of the kinetic frames. This study was pursued with the intent of identifying the
rotating unbalance in this installation and future installations in order to consequently shift the rotational equilibrium positions of the kinetic frames as needed.

Creating the desired equilibrium behavior for these kinetic frames proved to be a daunting task during their installation. A previous installation, S-281913, featured a similar design with two kinetic frames rotating on diagonal, discontinuous axes with preferred rotational positions of equilibrium. Numerous jigs were used to accurately align the rotational axes but a balancing system was not prepared for the S-281913 exhibit installation and difficulty adjusting the rotational equilibrium positions was extremely detrimental in completing the installation of the exhibit in a timely manner. Additionally, the installation team did not yet fully understand the cause of the unpredictable resting positions of the frames and, as such, struggled to modify its behavior in a strategic manner. Sarah and her team put in place more preparations to streamline the rotational equilibrium adjustment for the installation of S-337473 for the Wexner Center by integrating mounting brackets into the design of the exhibit to allow the placement of counterbalances as needed. Fig. 2 shows the internal hardware of the S-337473 frames with these mounting brackets and two of the mounted counterbalances.
Fig. 2. Internal hardware of S-337473. (a) Far view. (b) Close view. (c) Detailed view of internal hardware.
Although the inclusion of counterbalance mounting hardware in the design of this exhibit helped to simplify the rotational equilibrium adjustment, there remained no systematic approach to determine the appropriate counterbalance masses and mounting locations to control the rotational equilibrium positions.

The system developed during this study was implemented on a scale model of the S-337473 installation. An inertial measurement unit controlled by an Arduino 101 microcontroller collected the scale model’s rotational motion due to only static unbalance in the system in order to estimate the unknown parameters of eccentricity in the derived equation of motion for the system. A numerical algorithm used these estimated parameters and Newtonian mechanics to determine the appropriate sizes and locations of counterbalances to create the intended state of static unbalance in the rotational model. The scale model created for this research study mimicked the counterbalance mounting technique implemented in the S-337473 exhibit.

The same inertial measurement unit system measured the rotational motion of the system after the placement of these counterbalances and the resulting motion was objectively compared to the numerically predicted motion. This study evaluated the ability of this methodology to accurately estimate the eccentricity in the rotating assembly and to identify appropriate counterbalance placement in order to establish the desired rotational equilibrium positions. The balancing procedure tested in this study may aid Sarah Oppenheimer’s team during the installation of future projects and save valuable
installation time by implementing a more systematic procedure for balancing similar kinetic exhibits.

Keep in mind that, for the scope of this study, the phrase “balancing process” indicates the process of altering the rotational equilibrium positions, not eliminating unbalance.
B. Relevant Terminology

Important terminology specific to the experimental fixture utilized for collecting inertial measurements during this study is included in this section. Initial references to terms specific to this study that will be used throughout this document are indicated by italicized font. Fig. 3 shows a Solidworks rendition of the complete experimental fixture assembly.

![Complete Solidworks assembly of experimental fixture.](image)

Fig. 3. Complete Solidworks assembly of experimental fixture.

The sub-assembly outlined in red in Fig. 3 is referred to as the *frame* and the pieces outlined in green are referred to as the *shafts*. Fig. 4 shows a side-view of the frame sub-assembly to highlight the components involved in the attachment of counterbalances to the frame.
The frame sub-assembly consisted of two panels of glass and two aluminum alloy \textit{U-channels}, one of which is indicated by a red box in Fig. 4. Four aluminum 80/20 extrusions referred to as \textit{mounting bars} were also included in the frame sub-assembly. One of these extrusions is indicated by a green box in Fig. 4. The experimentally determined \textit{counterbalances} were connected to the mounting bars. More specific details related to the design of the testing fixture are discussed in a later section.

Section II of this document reviews the literature and concepts that were influential in modeling the dynamics of the system for this study and understanding numerical estimation techniques. Derivations related to the dynamics of the observed model and the experimental methodology are detailed in Section III. The collected measurements and numerical results are presented in Section IV and the accuracy and repeatability of the
developed balancing method is discussed in Section V. The implications of this study and future work is introduced in Section VI.
II. Literature Review

This study required the use of dynamics analysis and numerical methods techniques in order to accurately predict the motion of an unbalanced rotating rigid body and numerically estimate certain inertial quantities describing the rotating system. This section will review the concepts influencing the experimental methodology and analysis of this study including current machine balancing technology and numerical parameter estimation.

A. Rotating Unbalance

The presence of a rotating unbalance is typically concerned with rotating machines that are externally excited at high angular velocities. It can have serious consequences in an industrial environment due to the resulting undesirable mechanical vibrations such as excessive wear on mechanical components and audible noises during rotation, among other consequences. Unbalance in a rotating system can be the result of several different aspects of the machine material selection and fabrication methods.

Although the focus of this study, the dynamic behavior of the kinetic frames in the S-337473 exhibition, a rotating machine without an external excitation, some of the principles involved in the detection and compensation of industrial rotating unbalances
were useful in dynamic study of this exhibition. The consequences of rotating unbalance of most concern to the S-337473 exhibition are:

- Unfavorable rotational equilibrium positions for visual appeal and creative intent.
- Audible noises during rotation in the museum’s exhibition space.

As with any dynamic machine, several different aspects of the material fabrication techniques and assembly processes could contribute to the presence of a rotating unbalance. The discontinuous nature of the rotational axis in the S-337473 exhibition required particular consideration during assembly to minimize the possibility of axis misalignment.

There are four different types of rotating unbalance defined by the International Organization for Standardization that can be measured by current balancing technology. Each type of unbalance causes a variation in the “displacement of the principal axis of inertia from the shaft axis [3].” These types of unbalance include static unbalance, couple unbalance, quasi-static unbalance, and dynamic unbalance [3]. Combinations of these types of unbalance are also possible in rotating machines. Fig. 5 illustrates possible conditions leading to each of these types of unbalances with brief descriptions of these conditions to follow [3].
In the case of static unbalance, the principal axis of inertia is displaced from the rotational axis but is parallel to the rotational axis as the unbalance is in the plane of the rotor’s center of gravity. Couple unbalance results from two unbalances that are diametrically opposite, causing the principal axis of inertia to intersect the rotor’s center of gravity and the rotational axis. The principal axis of inertia of a quasi-static unbalance also intersects the rotational axis but does not intersect the rotor’s center of gravity. Finally, the principal axis of inertia in a dynamic unbalance does not intersect the rotational axis or the rotor’s center of gravity. The unbalance scenario of interest for the purpose of this study is that of...
a static unbalance. There are two methods for modeling a static unbalance in a rotating machine, either with the rotor’s center of gravity and, therefore, its principal axis of inertia offset from the rotational axis or with the rotor subject to a concentrated unbalance mass causing the resulting principal axis of inertia to be offset from the rotational axis. These two situations are shown in Figs. 6 and 7, respectively [3].

Fig. 6. Static unbalance of rotor with concentrated mass unbalance.

Fig. 7. Static unbalance of eccentric rotor.

The dynamic analysis of the S-337473 exhibition modeled the eccentricity in the kinetic frame as a concentrated mass, demonstrated by the unbalance scenario depicted in Fig. 6.
This selection of a dynamic model consisting of the kinetic frame and a concentrated eccentric mass influences the derivation of the system’s equation of motion discussed in a later section. Additionally, a core principle surrounding the behavior of rotating machines with static unbalances was considered in verifying the derived equation of motion discussed in a later section. As dictated by classical dynamics principles, the expected behavior of a rotating machine with a static unbalance consists of the “body free to rotate [seeking] a position where its center of gravity is lowest [4].”

A brief discussion of traditional balancing techniques is necessary to distinguish the experimental methods of this study discussed in a later section. A simplified model of traditional balancing machines evaluates static unbalance by placing the rotor in question on a rigid support that is mounted on a spring and damper, illustrated in Fig. 8 [5].

Fig. 8. Simplified balancing machine model.
This system consists of the eccentric mass, \( m_u \), at a radial location \( e \) from the rotor’s principle axis of inertia and the rotor of mass \( m \). The spring has a stiffness of \( k \) and the damping constant is \( c \). The vertical displacement of the rigid support, \( x \), during the rotation of the rotor at an angular velocity \( \Omega \) is measured. The equation of motion with respect to this vertical displacement and the solution for \( x(t) \) are listed in Eqs. 1 and 2 where \( \zeta \) is the damping ratio, \( \phi \) is phase angle, and \( \omega \) is the natural frequency of the system.

\[
 m \ddot{x} + c \dot{x} + k x = m_u \Omega^2 e \sin(\Omega t) \quad (1)
\]

\[
 x(t) = \left( \frac{\omega^2 m_u \Omega^2 e}{k} \right) \frac{e}{\sqrt{\left( \omega^2 - \Omega^2 \right)^2 + (2\zeta \omega \Omega)^2}} \sin(\Omega t - \phi) \quad (2)
\]

After defining the \( r \) as the frequency ratio of \( \Omega \) to \( \omega \), the solution can be written as follows in Eq. 3.

\[
 x(t) = \left( \frac{m_u r^2 e}{m} \right) \frac{e}{\sqrt{(1 - r^2)^2 + (2\zeta r)^2}} \sin(\Omega t - \phi) \quad (3)
\]

This substitution allows the the eccentricity, \( em_u \), to be determined by measuring the frequency response of the magnitude of \( x(t) \). An opposing mass can then be added to the system to eliminate the static unbalance. As opposed to this traditional method, this study utilizes only the rotation of the model to determine and address the eccentricity in the rotating system.
B. Madgwick Signal Filtering

The digital measurement system utilized during this study implemented the Madgwick signal orientation and filtering algorithm originally developed for the tracking of human motion in rehabilitation applications [6]. This orientation filter boasts advantages over other orientation filters such as the Kalman filter in its ability to efficiently capture sensor orientation at low sampling rates [6].

The Madgwick algorithm determines the orientation of the inertial measurement unit by first collecting the rates of rotation about the $x$, $y$, and $z$ axes with respect to the sensor. These measurements are arranged into the vector listed in Eq. 4 and used to calculate the quaternion derivative listed in Eq. 5 [6].

$$S_{\omega} = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix}$$

(4)

$$S\dot{q} = \frac{1}{2} S_{\hat{q}} \otimes S_{\omega}$$

(5)

The result of the calculation in Eq. 5 describes the rate of change of the earth frame with respect to the sensor frame [6]. At any time $t$ the orientation of the earth frame with respect to the sensor frame can be described with the result of Eq. 5 and the previous orientation estimate, shown in Eq. 6.

$$S_{E}q_{\omega,t} = S_{E}\hat{q}_{est,t-1} + S_{E}q_{\omega,t} \Delta t$$

(6)

The orientation algorithm requires the optimization of the objective function listed in Eq. 7.
The predefined direction of the earth’s magnetic field is represented by $E\hat{d}$ while the measured magnetic field in the frame of the sensor is represented by $S\hat{s}$. The Madgwick algorithm proceeds to use a gradient descent algorithm to optimize the objective function listed in Eq. 7. The consequently estimated orientation $S\hat{q}_{\text{v},t}$ is calculated with the previous estimation and the objective function error, listed in Eq. 8.

$$f(S\hat{q}, E\hat{d}, S\hat{s}) = S\hat{q}^* \otimes E\hat{d} \otimes S\hat{q} - S\hat{s}$$

(7)

The term $\mu_t$ in Eq. 8 describes the convergence rate of the estimated orientation. Finally, the estimated orientation at time $t$ is filtered with the weight $\gamma_t$ in Eq. 9.

$$S\hat{q}_{\text{v},t} = S\hat{q}_{\text{est},t-1} - \mu_t \frac{\nabla f}{||\nabla f||}$$

(8)

$$S\hat{q}_{\text{est},t} = S\hat{q}_{\text{v},t} + (1 - \gamma_t)S\hat{q}_{\text{o},t} , \quad 0 \leq \gamma_t \leq 1$$

(9)

For the purpose of this study, the Madgwick algorithm was applied to the measurements of an inertial measurement unit with a three-axis gyrometer and accelerometer. The sampling frequency for the digital measurements collected during this study, 25 [Hz], was chosen based on the experimental results presented in [6]. Based on these results, a sampling frequency of 25 [Hz] produced a root mean square error in rotation measurements of less than 2.5° [6].
C. Parameter Estimation

This study required a method to compare the theoretical dynamic behavior of the system of interest to its measured rotational motion. Several numerical model estimation algorithms are available in the MATLAB program package for this purpose, one of which is a grey-box estimation for modeling nonlinear dynamic behavior. This section explores the variations of a grey-box estimation to explain why the grey-box estimation was pursued for this study as well as the error minimization procedure implemented in the estimation.

Grey-box estimation is one of three types of system identification. The alternative types of system identification are white-box estimations and black-box estimations [7]. All three of these system identification protocols can estimate the parameters in a physical model, but they vary in how confident the user is in the theoretical model created to describe the physical model. Table 1 lists these distinctions.

<table>
<thead>
<tr>
<th></th>
<th>Theoretical Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>White-box estimation</td>
<td>Known</td>
</tr>
<tr>
<td>Grey-box estimation</td>
<td>Predicted</td>
</tr>
<tr>
<td>Black-box estimation</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

For the purpose of this study, the theoretical dynamics of the system were derived based on a rigid-body physical model. However, the parameters of the derived physical model,
the parameters of eccentricity in the rotation, were unknown values. Because the
dynamics of the system were predicted for this study, a black-box estimation model was
not applicable. The grey-box estimation for system identification was the most applicable
for the purpose of this study. As discussed later in this document, the theoretical model
describing the dynamic behavior of the system in question was derived in order to
estimate the unknown coefficients in its equation of motion. While this derivation
eliminated the selection of a black-box estimation, the uncertainty of the physical
parameters in this system made a white-box estimation impractical. The details of the
grey-box estimation performed in MATLAB are discussed in a later section with the
experimental methods of this study.
III. Experimental Methods

This study incorporated many different aspects of traditional engineering principles including rigid body dynamics, digital instrumentation, mechanical design, programming algorithm development, and numerical parameter estimation. This section details the theoretical and experimental methods involved in this dynamic study of the S-337473 exhibition.

A. Preliminary Derivations, Etc.

The equation of motion for the frame and an eccentric mass rotating about the rotational axis was initially derived to determine the necessary parameters for estimation and analysis. This section begins with the derivation of the coordinate systems necessary to describe the locations of the frame and the eccentric mass in the system followed by the derivation of the equation of motion for the system. Table 2 lists the relevant variables, units, and variable descriptions in the derivations detailed in this section. Vector quantities are denoted by bold font beginning in Table 2 and in any following utilization throughout this document.
Table 2. Relevant variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>${e_1,e_2,e_3}$</td>
<td>-</td>
<td>Cartesian unit coordinate vectors</td>
</tr>
<tr>
<td>${e'_1,e'_2,e'_3}$</td>
<td>-</td>
<td>Exhibit unit coordinate vectors</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>-</td>
<td>Transformation matrix of ${e_1,e_2,e_3}$ to ${e'_1,e'_2,e'_3}$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>[rad]</td>
<td>Initial angular position of eccentric mass</td>
</tr>
<tr>
<td>${e_p,e_\Phi,e_s}$</td>
<td>-</td>
<td>Exhibit cylindrical unit coordinate vectors</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>-</td>
<td>Transformation matrix of ${e'_1,e'_2,e'<em>3}$ to ${e_p,e</em>\Phi,e_s}$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>-</td>
<td>Transformation matrix of ${e_1,e_2,e_3}$ to ${e_p,e_\Phi,e_s}$</td>
</tr>
<tr>
<td>$t$</td>
<td>[s]</td>
<td>Independent time variable</td>
</tr>
<tr>
<td>$\theta$</td>
<td>[rad]</td>
<td>Angular displacement of system</td>
</tr>
<tr>
<td>$T$</td>
<td>[in-lb]</td>
<td>Total kinetic energy</td>
</tr>
<tr>
<td>$R$</td>
<td>[in-lb]</td>
<td>Total dissipative energy</td>
</tr>
<tr>
<td>$V$</td>
<td>[in-lb]</td>
<td>Total potential energy</td>
</tr>
<tr>
<td>$r_f$</td>
<td>[in]</td>
<td>Location of center of gravity of frame</td>
</tr>
<tr>
<td>$r_e$</td>
<td>[in]</td>
<td>Location of center of gravity of eccentric mass</td>
</tr>
<tr>
<td>$p$</td>
<td>[in]</td>
<td>Radial distance of eccentric mass from rotational axis</td>
</tr>
<tr>
<td>$z$</td>
<td>[in]</td>
<td>Axial distance of eccentric mass from origin</td>
</tr>
<tr>
<td>$T_f$</td>
<td>[in-lb]</td>
<td>Kinetic energy of frame</td>
</tr>
<tr>
<td>$T_e$</td>
<td>[in-lb]</td>
<td>Kinetic energy of eccentric mass</td>
</tr>
<tr>
<td>$T_0$</td>
<td>[in-lb]</td>
<td>Generic kinetic energy term</td>
</tr>
<tr>
<td>$J$</td>
<td>[lb-in²]</td>
<td>Total mass moment of inertia of a rotating system</td>
</tr>
<tr>
<td>$J_e$</td>
<td>[lb-in²]</td>
<td>Mass moment of inertia of eccentric mass about $e_s$ axis</td>
</tr>
<tr>
<td>$m_e$</td>
<td>[lb]</td>
<td>Eccentric mass</td>
</tr>
</tbody>
</table>

continued
<table>
<thead>
<tr>
<th>Variable</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_f$</td>
<td>[lb-in²]</td>
<td>Mass moment of inertia of frame about $e_3$ axis</td>
</tr>
<tr>
<td>$c_t$</td>
<td>[in-lb-s/rad]</td>
<td>Damping constant</td>
</tr>
<tr>
<td>$V_f$</td>
<td>[in-lb]</td>
<td>Potential energy of frame</td>
</tr>
<tr>
<td>$V_e$</td>
<td>[in-lb]</td>
<td>Potential energy of eccentric mass</td>
</tr>
<tr>
<td>$m$</td>
<td>[lb]</td>
<td>Generic mass</td>
</tr>
<tr>
<td>$g$</td>
<td>[in/s²]</td>
<td>Acceleration due to gravity</td>
</tr>
<tr>
<td>$h$</td>
<td>[in]</td>
<td>Generic vertical height from datum</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>-</td>
<td>Exponential filter weight</td>
</tr>
<tr>
<td>$T_d$</td>
<td>[s]</td>
<td>Damped period</td>
</tr>
<tr>
<td>$\delta$</td>
<td>-</td>
<td>Logarithmic decrement value</td>
</tr>
<tr>
<td>$\theta_1$</td>
<td>[rad]</td>
<td>First angular displacement curve maxima</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>[rad]</td>
<td>Second angular displacement curve maxima</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>-</td>
<td>Damping ratio</td>
</tr>
<tr>
<td>$\omega_n$</td>
<td>[rad/s]</td>
<td>Natural frequency</td>
</tr>
<tr>
<td>$\omega_d$</td>
<td>[rad/s]</td>
<td>Damped frequency</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>[rad]</td>
<td>Initial value of logarithmic decrement curve</td>
</tr>
<tr>
<td>$x$</td>
<td>-</td>
<td>State vector</td>
</tr>
<tr>
<td>$x_1$</td>
<td>[rad]</td>
<td>Angular displacement state</td>
</tr>
<tr>
<td>$x_2$</td>
<td>[rad/s]</td>
<td>Angular velocity state</td>
</tr>
<tr>
<td>$r_e$</td>
<td>[in]</td>
<td>Position vector of counterbalance mass</td>
</tr>
<tr>
<td>$c_{1e}$</td>
<td>[in]</td>
<td>Position of counterbalance mass in $e_1$ direction</td>
</tr>
<tr>
<td>$c_{2e}$</td>
<td>[in]</td>
<td>Position of counterbalance mass in $e_2$ direction</td>
</tr>
<tr>
<td>$c_{3e}$</td>
<td>[in]</td>
<td>Position of counterbalance mass in $e_3$ direction</td>
</tr>
</tbody>
</table>
The equation of motion for this system was derived with the assumption that the frame’s center of mass was the origin of the initial Cartesian coordinate system, \{e_1,e_2,e_3\}. Two transformations of the \{e_1,e_2,e_3\} coordinate system were performed in order to define a cylindrical coordinate system aligned with the rotational axis for the purpose of locating the eccentric mass in the system. Fig. 9 shows a diagram of the frame as well as its center of mass and the \{e_1,e_2,e_3\} coordinate system.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_c)</td>
<td>[in]</td>
<td>Radial distance of counterbalance mass from rotational axis</td>
</tr>
<tr>
<td>(T_c)</td>
<td>[in-lb]</td>
<td>Kinetic energy of counterbalance mass</td>
</tr>
<tr>
<td>(h_c)</td>
<td>[in]</td>
<td>Vertical displacement of counterbalance mass from datum</td>
</tr>
<tr>
<td>(V_c)</td>
<td>[in-lb]</td>
<td>Potential energy of counterbalance mass</td>
</tr>
<tr>
<td>(F_g)</td>
<td>[lb]</td>
<td>Gravitational force</td>
</tr>
<tr>
<td>(m_c)</td>
<td>[lb]</td>
<td>Mass of counterbalance mass</td>
</tr>
</tbody>
</table>

Fig. 9. Diagram of frame with Cartesian coordinate system.
A rotation of the Cartesian coordinate system, \{e_1,e_2,e_3\}, of 45° about the e_1 axis was performed to create the \{e_1',e_2',e_3'\} coordinate system with the e_3' axis collinear to the rotational axis. Refer to Fig. 10. for a depiction of this rotation.

Fig. 10. Rotation of \{e_1,e_2,e_3\} about the e_1 axis.

The matrix transformation reflecting this rotation and the corresponding transformation matrix, \(\beta_1\), are listed in Eqs. 10 and 11, respectively.

\[
\begin{bmatrix}
e_1' \\
e_2' \\
e_3'
\end{bmatrix} = 
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos 45° & \sin 45° \\
0 & -\sin 45° & \cos 45°
\end{bmatrix}
\begin{bmatrix}
e_1 \\
e_2 \\
e_3
\end{bmatrix}
\]  \hspace{1cm} (10)

\[
\beta_1 = 
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos 45° & \sin 45° \\
0 & -\sin 45° & \cos 45°
\end{bmatrix}
\]  \hspace{1cm} (11)

The \{e_1',e_2',e_3'\} coordinate system was transformed by an additional rotation of the system by an angle \(\phi\) about the e_3' axis to create the cylindrical coordinate system,
\{e_p,e_\Phi,e_s\}, utilized to locate the initial position of the eccentric mass in the system. Refer to Fig. 11 for an illustration of this rotation.

![Rotation of \{e_1',e_2',e_3'\} about the e_3' axis.](image)

Eqs. 12 and 13, respectively, lists this matrix transformation as well as the corresponding transformation matrix, \(\beta_2\).

\[
\begin{bmatrix}
e_p \\
e_\Phi \\
e_s
\end{bmatrix} =
\begin{bmatrix}
cos \phi & sin \phi & 0 \\
-sin \phi & cos \phi & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
e_1' \\
e_2' \\
e_3'
\end{bmatrix}
\]

(12)

\[
\beta_2 =
\begin{bmatrix}
cos \phi & sin \phi & 0 \\
-sin \phi & cos \phi & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(13)

The following relationships between the \{e_p,e_\Phi,e_s\}, \{e_1',e_2',e_3'\}, and \{e_1,e_2,e_3\} coordinate systems were established in Eq. 14-16 with the previously derived transformation matrices.
The product of matrices $\beta_2$ and $\beta_1$ was defined as the transformation matrix $\beta$. This relationship and the transformation matrix $\beta$ is listed in Eqs. 17 and 18.

$$\beta = \beta_2 \beta_1$$

Finally, the transpose of the transformation matrix $\beta$ was taken in order to derive the transformation matrix necessary to transform the $\{e_p, e_\phi, e_s\}$ coordinate system to the $\{e_1, e_2, e_3\}$ coordinate system. This operation is listed in Eqs. 19 and 20.

$$
\begin{bmatrix}
    e_1' \\
    e_2' \\
    e_3'
\end{bmatrix} = \beta_1
\begin{bmatrix}
    e_1 \\
    e_2 \\
    e_3
\end{bmatrix}
$$

$$
\begin{bmatrix}
    e_p \\
    e_\phi \\
    e_s
\end{bmatrix} = \beta_2
\begin{bmatrix}
    e_1' \\
    e_2' \\
    e_3'
\end{bmatrix}
$$

$$
\begin{bmatrix}
    e_p \\
    e_\phi \\
    e_s
\end{bmatrix} = \beta_2 \beta_1
\begin{bmatrix}
    e_1 \\
    e_2 \\
    e_3
\end{bmatrix}
$$

$$
\begin{bmatrix}
    e_p \\
    e_\phi \\
    e_s
\end{bmatrix} =
\begin{bmatrix}
    \cos \phi & \sin \phi & 0 \\
    -\sin \phi & \cos \phi & 0 \\
    0 & -\sin 45^\circ & \cos 45^\circ
\end{bmatrix}
\begin{bmatrix}
    e_1 \\
    e_2 \\
    e_3
\end{bmatrix}
$$

$$
\begin{bmatrix}
    e_1 \\
    e_2 \\
    e_3
\end{bmatrix} =
\begin{bmatrix}
    \cos \phi & -\sin \phi & 0 \\
    \sin \phi & \cos \phi & 0 \\
    \sin 45^\circ & \cos 45^\circ & 0
\end{bmatrix}
\begin{bmatrix}
    e_p \\
    e_\phi \\
    e_s
\end{bmatrix}
$$
These transformation matrices were used in the following derivation of the equation of motion of the frame and an eccentric mass about the rotational axis, $e_s$. Lagrange’s equation, shown in Eq. 21, was used to derive the equation of motion for this system.

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \theta} \right) - \frac{\partial T}{\partial \theta} + \frac{\partial R}{\partial \theta} + \frac{\partial V}{\partial \theta} = 0$$

(21)

As seen in Eq. 21, this derivation implemented Lagrange’s equation with respect to the coordinate $\theta$, the angular displacement of the system about the $e_s$ axis. To begin, the location of the frame and the eccentric mass were defined as follows in Eqs. 22 and 23.

$$r_f = 0e_p + 0e_\phi + 0e_s$$

(22)

$$r_e = pe_p + 0e_\phi + ze_s$$

(23)

The first quantity necessary to utilize Eq. 21 was the total kinetic energy of the rotating system. The total kinetic energy of this system was defined as in Eq. 24 as the sum of the frame’s kinetic energy and the kinetic energy of the eccentric mass.

$$T = T_f + T_e$$

(24)

Recall that, in general, the rotational kinetic energy of a rotating object about a fixed axis is defined by Eq. 25.
The total mass moment of inertia, \( J \), in Eq. 25 needed to include both the mass moment of inertia of the entire frame and the mass moment of inertia of the eccentric mass. As the eccentric mass was treated as a point mass, its mass moment of inertia was determined with Eq. 26 using its mass and its radial distance from the rotational axis, \( p \).

\[
J_e = m_e p^2
\]  

(26)

With the mass moment of inertia of the eccentric mass defined, the kinetic energy of the eccentric mass was determined by Eqs. 27 and 28.

\[
T_e = \frac{1}{2} J_e \left( \frac{d\theta}{dt} \right)^2
\]  

(27)

\[
T_e = \frac{1}{2} m_e p^2 \left( \frac{d\theta}{dt} \right)^2
\]  

(28)

The mass moment of inertia of the frame about the \( e_s \) axis, \( J_f \), was the sum of the mass moments of inertia of each component in the frame assembly about the \( e_s \) axis. This total was referred to as \( J_f \) during this derivation. The mass moment of inertia of the frame was eventually estimated for experimental purposes and this estimation is discussed in a later section. For the purpose of this derivation, the kinetic energy of the frame was defined as follows in Eq. 29.
Finally, the total kinetic energy of the system was determined and is listed in Eq. 30.

\[ T = \frac{1}{2} m_e p^2 \left( \frac{d\theta}{dt} \right)^2 + \frac{1}{2} J_f \left( \frac{d\theta}{dt} \right)^2 \] (30)

The dissipative energy in the system, \( R \), was modeled as torsional damping opposing the rotation of the system, \( \theta \), about the \( e_s \) axis. The dissipative energy in the system was defined as follows in Eq. 31.

\[ R = \frac{1}{2} c_t \left( \frac{d\theta}{dt} \right)^2 \] (31)

Additionally, the total potential energy of the system was required to utilize Eq. 21 in determining the equation of motion. As with the total kinetic energy, the total potential energy of the system consisted of the potential energy of both the frame and the eccentric mass, shown in Eq. 32.

\[ V = V_f + V_e \] (32)

The general equation for the potential energy of an object is defined by the relationship listed in Eq. 33.

\[ V_0 = mgh \] (33)
For the purpose of determining the potential energies of the frame and the eccentric mass, the vertical height of each object was defined relative to a plane with the normal unit vector $e_3$ and containing the center of mass of the frame. Fig. 12 illustrates this datum as a horizontal line from a view normal to the plane of unit vectors $e_2$ and $e_3$.

![Figure 12: Datum used for definition of potential energy.]

One consequence of placing the origin at the center of mass of the frame was that the potential energy of the frame relative to the origin was eliminated, as declared in Eq. 34.

$$V_f = 0$$  \hspace{1cm} (34)

Alternately, the potential energy of the eccentric mass varied throughout the rotation about the $e_s$ axis. In order to fully define the potential of the eccentric mass, the displacement of the eccentric mass from the origin in the $e_3$ direction at any angle $\theta$ of rotation needed to be determined. Recall the position vector defined in Eq. 23 to locate the eccentric mass in the cylindrical coordinate system, $\{e_p, e_\phi, e_3\}$. Fig. 13 illustrates the rotation of the eccentric mass’ position vector in the plane of $e_p$ and $e_\phi$. 

30
The magnitude of the eccentric mass’ position vector, \( ||\mathbf{r}_e|| \), remained the same throughout this rotation but its angular displacement, \( \theta \), required that the cylindrical components of the position vector, \( \mathbf{r}_e \), be redefined as in Eq. 35.

\[
\mathbf{r}_e = p \cos \theta \mathbf{e}_p + p \sin \theta \mathbf{e}_\phi + z \mathbf{e}_s
\]  

(35)

This position vector could alternately be written in matrix form as listed below in Eq. 36.

\[
\mathbf{r}_e = \begin{bmatrix} p \cos \theta \\ p \sin \theta \\ z \end{bmatrix}_{p\phi s}
\]  

(36)

This position vector and its components were transformed from the cylindrical coordinate system \( \{ \mathbf{e}_p, \mathbf{e}_\phi, \mathbf{e}_s \} \) into the Cartesian coordinate system \( \{ \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \} \) by the relationship defined in Eq. 20. This transformation is listed in Eq. 37.

\[
\begin{bmatrix}
    p \cos \theta \cos \phi - p \sin \theta \sin \phi \\
    p \cos \theta \sin \phi \cos 45^\circ + p \sin \theta \cos \theta \cos 45^\circ - z \sin 45^\circ \\
    p \cos \theta \sin \phi \sin 45^\circ + p \sin \theta \cos \phi \sin 45^\circ + z \cos 45^\circ
\end{bmatrix}_{123} = \beta^T \begin{bmatrix} p \cos \theta \\ p \sin \theta \\ z \end{bmatrix}_{p\phi s}
\]  

(37)
Therefore, the displacement of the eccentric mass throughout the rotation $\theta$ in the vertical direction $e_3$ was defined as follows in Eq. 38 with the resulting potential energy of the eccentric mass listed in Eq. 39.

$$r_{e,3} = p \cos \theta \sin \phi \sin 45^\circ + p \sin \theta \cos \phi \sin 45^\circ + z \cos 45^\circ$$ (38)

$$V_e = m_e g (p \cos \theta \sin \phi \sin 45^\circ + p \sin \theta \cos \phi \sin 45^\circ + z \cos 45^\circ)$$ (39)

This potential energy was further simplified by implementing the trigonometric difference identity listed in Eq. 40.

$$\sin(\theta + \phi) = \cos \theta \sin \phi + \sin \theta \cos \phi$$ (40)

Including this trigonometric identity in the potential energy of the eccentric mass, the potential energy of the eccentric mass throughout the rotation $\theta$ about the $e_s$ axis was defined as Eq. 41.

$$V_e = m_e g [p \sin 45^\circ \sin(\theta + \phi) + z \cos 45^\circ]$$ (41)

Eq. 42 lists the total potential energy of the system with this substitution.

$$V = m_e g [p \sin 45^\circ \sin(\theta + \phi) + z \cos 45^\circ]$$ (42)

The equation of motion was now fully defined using the Lagrange equation listed initially in Eq. 21. The following equations, Eqs. 43-46, list the individual terms in the Lagrange equation and, finally, the differential equation describing the rotation $\theta$ about the $e_s$ axis is listed in Eq. 47.
This derived equation of motion fully describes the dynamic rotational behavior of the model created to study the S-337473 exhibition. The following sections will detail the equipment and methodology utilized to gather dynamic measurements of the model to eventually estimate the unknown parameters in this equation of motion and alter the physical model accordingly for the desired dynamic behavior.
B. Measurement Instrumentation

This study required a measurement system that would be adaptable to a variety of kinetic installations for future work. Micro-controllers and various sensors that are able to interact with them are an effective and portable tool for collecting information about their environment. This study specifically required the measurement of rotation about a fixed axis. For this purpose, two different Arduino micro-controllers were considered in addition to two possible sensors capable of measuring rotation angles and two data collection systems. This section details these measurement systems and their strengths and weaknesses.

Early measurement iterations during this study utilized an Arduino Uno micro-controller connected to an InvenSense MPU-6050 inertial measurement unit. The angular measurements collected with this particular inertial measurement unit took advantage of the Digital Motion Processor integrated into the sensor chip in order to convert the raw accelerometer and gyrometer measurements into rotation measurements about the desired axis. This system also utilized an Adafruit Data Logger Shield to capture the angular readings completed by the MPU-6050 inertial measurement unit. Fig. 14 shows the MPU-6050 inertial measurement unit in additional to a Data Logger Shield and Arduino Uno assembly [8],[9].
For reference, the Arduino sketch created to control the Arduino Uno with this initial measurement system is included in Appendix A. Although this measurement system provided accurate measurements of the model’s rotation, it required additional steps during experimentation to analyze the data saved to the MicroSD card connected to the Data Logger Shield. The MicroSD card had to be removed from the Data Logger Shield between readings in order to assess the collected data. The MPU-6050 inertial measurement unit’s Digital Motion Processor also required between 5 and 10 seconds to calibrate its angular readings before the model could be allowed to rotate. As a consequence, a button and a light-emitting diode were incorporated into this measurement system to indicate to the researcher when the measurements were settled at their correct initial values.

The second measurement system implemented during this study simplified the researcher’s interaction with the system by calculating the angular measurements with the Arduino library CurieIMU for on-board inertial measurement units and wirelessly...
transmitting the angular measurements to a CSV file on the researcher’s computer for data analysis. This measurement system incorporated an Arduino Board 101 with an on-board inertial measurement unit in addition to the JBtek HC-05 bluetooth module. The Arduino Board 101 and the HC-05 bluetooth module are shown in Fig. 15 [10],[11].

![Arduino Board 101 and HC-05 Bluetooth module](image)

(a)     (b)

Fig. 15. (a) Arduino Board 101. (b) HC-05 Bluetooth module.

The Arduino sketch for this final measurement system to control the Arduino Board 101 in conjunction with the inertial measurement unit and bluetooth module is listed in Appendix A. Appendix B also displays the circuit schematic for the connection of the HC-05 Bluetooth Module to the Arduino Board 101.

This Arduino sketch was adapted from the example sketch listed in [12] to collect the raw accelerometer and gyrometer measurements in the correct orientation and gather the values of roll, pitch, and heading for each reading. The Madgwick filtering algorithm enabled the Arduino to output a smooth signal of the angular displacement in the
orientation of the Arduino as mounted on the rotating model’s shaft. The sampling
frequency for the Arduino 101 inertial measurement unit, the HC-05 bluetooth module,
and the Madgwick filtering algorithm was 25 Hz. At each sampling time, the desired
measurement values were sent by the Arduino to the HC-05 bluetooth module and
subsequently the bluetooth module wirelessly delivered this information to the computer.
A graphical user interface, Wireless Serial, created by [13] displayed a live feed of the
measurements, eliminating the need for a visual indication of calibration, and allowed the
researcher to control the collection of this data to the appropriate Excel files for further
analysis.
C. Test Fixture Design

The test fixture for this study was designed to approximate a 1:4 scale model of one of the rotating glass and aluminum frames featured in the S-337473 exhibition as closely as possible in order to evaluate the effectiveness of this parametric system for future kinetic installations similar to that of the S-337473 exhibit. Fig. 16 contains a Solidworks rendering of the test fixture developed for this study.

![Fig. 16. Solidworks rendering of test fixture.](image)

The terminology specific to this test fixture was discussed in an earlier section including the sub-assemblies of the test fixture: the frame and the shafts. The outer structure of the test fixture was also an important aspect of this study with its modularity and adaptability. Additionally, a mounting assembly was designed to attach the Arduino Board 101 and the other electronic equipment to the shaft ensuring consistent axis alignment of these
electronic components throughout measurement iterations. This section details all of these components and their mechanical design.

The outermost portion of the outer structure consisted of four standard T-slotted aluminum bars assembled into a square with inner dimensions of 36 in. by 36 in. The diagonal components attached to this square were also standard aluminum T-slotted bars mounted at an angle of 45° from each of the outer bars. Appendix C contains dimensioned drawings of the individual components in this structure.

The shaft sub-assemblies, illustrated in Fig. 17, supported the frame by connecting to either of the U-channels and also served as the rotational axis for the model.

![Fig. 17. Shaft sub-assembly.](image)

These assemblies were attached to the outer structure’s T-slot components by a mounting hardware component specifically designed for attachment to the T-slot bars. One end of a
1 in. diameter tube was inserted into this mounting hardware while the opposite end of this tube was inserted into a needle roller bearing press fit into the shaft tube. The mechanical specifications of the McMaster-Carr needle roller bearings chosen for this design are listed in Table 3.

Table 3. Needle roller bearing mechanical specifications.

<table>
<thead>
<tr>
<th>Inner diameter [in.]</th>
<th>1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outer diameter [in.]</td>
<td>1.25</td>
</tr>
<tr>
<td>Width [in.]</td>
<td>0.875</td>
</tr>
<tr>
<td>Material</td>
<td>Steel</td>
</tr>
<tr>
<td>Lubricant type</td>
<td>Grease</td>
</tr>
<tr>
<td>Dynamic radial load capacity [lb.]</td>
<td>4,050</td>
</tr>
<tr>
<td>Static radial load capacity [lb.]</td>
<td>6,400</td>
</tr>
<tr>
<td>Maximum speed [rpm]</td>
<td>8,000</td>
</tr>
</tbody>
</table>

Finally, the shaft tube was press fit into a 3D-printed ABS plastic attachment piece. Appendix C contains dimensioned drawings of the individual pieces in the upper and lower shaft sub-assemblies.

The frame sub-assembly consisted of two rectangular plates of glass, two aluminum U-channels, four aluminum T-slot bars, and miscellaneous hardware. The plates of glass were attached to the aluminum U-channels with a Gorilla brand epoxy. This sub-assembly is depicted in Fig. 18.
An exploded view of the frame sub-assembly is also shown in Appendix C along with dimensioned drawings of the individual components in the assembly.

The final component of the test fixture was the mounting sub-assembly for the electronic measurement system incorporated into the study. This mounting sub-assembly was designed in Solidworks and subsequently 3D printed with ABS plastic material. This sub-assembly consisted of individual attachment pieces for the Arduino Board 101, the HC-05 bluetooth module, and an external power supply for the measurement system. The attachment pieces for the Arduino Board 101 and the HC-05 bluetooth module were adapted from designs by [14] and [15], respectively. Fig. 19 contains a rendering of the mounting sub-assembly as well as mounted in place on the test fixture for clarity.
Fig. 19. (a) Mounting sub-assembly. (b) Mounting sub-assembly on test fixture.
Refer to Appendix C for dimensioned drawings of this sub-assemblies individual components.

Although this experimental fixture maintains the basic structure of the exhibition for the Wexner Center, the scaled size of the test fixture allowed simplifications to be made to the exhibit’s design. Some design elements were crucial in mimicking and improving the balancing process perform on the S-337473 exhibit. These crucial design elements included the 45° angle of the rotational axis relative to the ground, the discontinuity of the rotational axis inside of the frame, and the degrees of freedom for counterbalance placement within the U-channels. The counterbalances needed to have the freedom to be placed anywhere along the length of the U-channels as well as anywhere in the width of the U-channels. These degrees of freedom are illustrated in Fig. 20.

![Fig. 20. Degrees of freedom of counterbalance placement in U-channels.](image-url)
The degrees of freedom in the placement of the counterbalances was crucial during the installation of the S-337473 exhibition and, as such, an important feature of the fixture design for this study.

Modifications had to be made to the test fixture upon construction in order to address strong torsional damping present in the test fixture due to friction. The first modification implemented in the test fixture was the replacement of the standard grease used to lubricate the needle roller bearings in the shaft sub-assemblies. McMaster-Carr did not provide specific information about the grease originally used to lubricate the needle roller bearings. This grease was replaced with a Silicone and Teflon based lubricant. Two thrust bearings were also incorporated into the lower shaft sub-assembly to prevent friction between the flat surface of the shaft and the corresponding face of the mounting hardware in the sub-assembly, labeled in Fig. 21 as Bearings A and B, respectively.

![Fig. 21. Thrust bearings inserted between lower shaft and mounting hardware.](image)
Additionally, Table 4 lists the mechanical specifications for these thrust bearings.

Table 4. Thrust bearing specifications.

<table>
<thead>
<tr>
<th></th>
<th>Thrust Bearing A</th>
<th>Thrust Bearing B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inner diameter [in.]</strong></td>
<td>1.005</td>
<td>1.003</td>
</tr>
<tr>
<td><strong>Outer diameter [in.]</strong></td>
<td>1.500</td>
<td>2.000</td>
</tr>
<tr>
<td><strong>Thickness [in.]</strong></td>
<td>0.125</td>
<td>0.1875</td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>SAE 841 Bronze</td>
<td>SAE 863 Iron-Copper Bronze</td>
</tr>
<tr>
<td><strong>Dynamic load capacity</strong></td>
<td>250 lbs. @ 90 rpm</td>
<td>700 lbs. @ 30 rpm</td>
</tr>
<tr>
<td><strong>Lubricant method</strong></td>
<td>Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td><strong>Lubricant</strong></td>
<td>SAE 80 Oil with PTFE</td>
<td>ISO 460 Oil</td>
</tr>
</tbody>
</table>

The final modification made to the test fixture upon construction was the removal of the top T-slotted aluminum bar. Inserting the two thrust bearings in the lower shaft sub-assembly shifted the frame’s position upward and, as a result, shifted the upper diagonal T-slotted aluminum bar. The removal of the top T-slotted aluminum bar resolved this shift in dimensions. Fig. 22 shows the the final test fixture fully assembled reflecting these modifications.
The complete bill of materials and an exploded view of the test fixture assembly are included in Appendix C.

The digital measurement system discussed previously along with the test fixture detailed in this section provided the rotational measurements that were analyzed and used to estimate the parameters of unbalance in the system.
D. Data Analysis

During the experimental procedure for this study, the frame was initially positioned at a small angular displacement from its vertical position and allowed to rotate to its natural resting position. The measurements collected by the Arduino Board 101 and transmitted to a Excel file during these periods of data collection were analyzed numerically in MATLAB software. The full MATLAB script used to analyze the collected data is listed in Appendix D. This section describes all of the relevant sections of this MATLAB script and their function in processing the data collected for the purpose of this study.

Initially, the angular velocity was calculated with a forward differentiation of the collected angular displacement measurements. For reference, this calculation is listed in Eq. 48.

\[ \dot{\theta}_i = \frac{\theta_{i+1} - \theta_i}{t_{i+1} - t_i} \]  

These angular velocities were subsequently filtered with an exponential filter of weight \( \alpha \) as seen in Eq. 49.

\[ \dot{\theta}_i = \alpha \dot{\theta}_{i-1} + (1 - \alpha)\dot{\theta}_i \]  

The weight of the exponential filter was declared as 0.9 for this study. To show an example of the results of data collection and filtering during this study, Fig. 23 depicts one of the data sets of angular displacement collected as well as the corresponding filtered angular velocity.
Fig. 23 will also aid in the explanation of the following data analysis. The main use of the angular velocity calculation was the determination of the initial angular velocity. This quantity was necessary to verify the results of the logarithmic decrement method performed on the collected data to estimate the torsional damping constant in the system. Estimating the damping ratio of the system based on the experimental measurements required several quantities collected from the angular displacement curves: the first two local maxima of the curve and the period of the damped signal.

These quantities were captured from the collected data, as labeled in Fig. 24, and utilized in the calculation of the decrement value, the damping ratio, and the natural frequency of the system, listed in Eqs. 50-56.
The final quantity necessary to verify the correct calculation of the damping ratio for the selected set of measurements was the initial value of the exponential curve corresponding to the logarithmic decrement, \( \Theta \). The equation used to calculate this quantity is listed in Eq. 53.
These calculations, performed in MATLAB, resulted in an estimate of the damping ratio for the respective set of measurements. To illustrate this, the same angular displacement measurements referred to in Figs. 23 and 24 are shown in Fig. 25 with the corresponding exponential curve calculated with the logarithmic decrement method.

\[
\Theta = \sqrt{\theta_0^2 + \left(\frac{\dot{\theta}_0 + \zeta \omega_n \theta_0}{\sqrt{1 - \zeta^2 \omega_n^2}}\right)^2}\]

\[d(t) = \Theta e^{(-\omega_n t)}\]  \hspace{1cm} (53)

Determining an accurate estimate of the damping in this system was crucial to the parameter estimation process implemented in MATLAB in order to minimize the number of unknown parameters in the derived equation of motion. Recall the derived equation of motion for the frame and the eccentric mass listed in Eq. 47. The parameters of interest in this dynamic model were \(m_e, p, J_f, c_t,\) and \(\phi\). The mass moment of inertia, \(J_f\), about the
rotational axis was calculated with the physically recorded masses of the frame and shaft sub-assembly components and the necessary transformation matrices. These masses are listed in Table 5.

Table 5. Masses of frame and shaft sub-assembly components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Mass [lb]</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-channel 1</td>
<td>8.1016</td>
</tr>
<tr>
<td>U-channel 2</td>
<td>8.1212</td>
</tr>
<tr>
<td>Mounting bar 1</td>
<td>1.4006</td>
</tr>
<tr>
<td>Mounting bar 2</td>
<td>1.4068</td>
</tr>
<tr>
<td>Mounting bar 3</td>
<td>1.4010</td>
</tr>
<tr>
<td>Mounting bar 4</td>
<td>1.3994</td>
</tr>
<tr>
<td>Shaft and bearing 1</td>
<td>0.9426</td>
</tr>
<tr>
<td>Shaft and bearing 2</td>
<td>0.9438</td>
</tr>
<tr>
<td>Shaft attachment piece 1</td>
<td>0.4194</td>
</tr>
<tr>
<td>Shaft attachment piece 2</td>
<td>0.4196</td>
</tr>
<tr>
<td>T-slot mounting hardware</td>
<td>0.0178</td>
</tr>
<tr>
<td>T-slot mounting hardware</td>
<td>0.0178</td>
</tr>
<tr>
<td>Glass piece 1</td>
<td>6.2536</td>
</tr>
<tr>
<td>Glass piece 2</td>
<td>6.2656</td>
</tr>
</tbody>
</table>

A sample calculation for the mass moment of inertia of one U-channel is included in Appendix E and all of the components' mass moments of inertia are listed in Table 6.
Table 6. Frame and shaft components’ mass moments of inertia.

<table>
<thead>
<tr>
<th>Component</th>
<th>Mass moment of inertia [lb-in$^2$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-channel 1</td>
<td>481.11</td>
</tr>
<tr>
<td>U-channel 2</td>
<td>481.11</td>
</tr>
<tr>
<td>Mounting bar 1</td>
<td>81.19</td>
</tr>
<tr>
<td>Mounting bar 2</td>
<td>81.19</td>
</tr>
<tr>
<td>Mounting bar 3</td>
<td>81.19</td>
</tr>
<tr>
<td>Mounting bar 4</td>
<td>81.19</td>
</tr>
<tr>
<td>Shaft and bearing 1</td>
<td>0.20</td>
</tr>
<tr>
<td>Shaft and bearing 2</td>
<td>0.20</td>
</tr>
<tr>
<td>Shaft attachment piece 1</td>
<td>0.67</td>
</tr>
<tr>
<td>Shaft attachment piece 2</td>
<td>0.67</td>
</tr>
<tr>
<td>Glass piece 1</td>
<td>392.35</td>
</tr>
<tr>
<td>Glass piece 2</td>
<td>392.35</td>
</tr>
<tr>
<td>Total, $J_f$</td>
<td>2073.42</td>
</tr>
</tbody>
</table>

The damping constant was estimated with the calculated mass moment of inertia $J_f$ in Eq. 54 and treated as a fixed parameter during the parameter estimation process.

$$c_t = 2\zeta J_f \omega_n$$ (54)

The unknown, or “free,” parameters in this equation of motion were $m_e$, $\rho$, and $\phi$. Initial guesses were declared for these free parameters and a nonlinear grey-box estimation was
used to estimate their values. This estimation process required the fixed parameter values, initial values for the free parameters, sample data with initial angular displacement and initial angular velocity, and a model file describing the dynamics of the system. This model file required by MATLAB’s nonlinear grey-box estimation method contains a state space model of the system used to solve the relevant ordinary differential equation for estimation. This model file is included in Appendix D. The state vector and the respective state equations used to describe the dynamic model in this study are listed in Eqs. 55-57.

\[
x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}
\]  
\(x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}\)  
(55)

\[
\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix}
\]  
\(\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix}\)  
(56)

\[
\ddot{x} = \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x}_2 \\ -c_t x_2 - m_e g p \sin(45^\circ) \cos(\phi + x_1) \\ J_f + m_e p^2 \end{bmatrix}
\]  
\(\ddot{x} = \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x}_2 \\ -c_t x_2 - m_e g p \sin(45^\circ) \cos(\phi + x_1) \\ J_f + m_e p^2 \end{bmatrix}\)  
(57)

Within MATLAB’s nonlinear grey-box estimation, certain parameter search specifications can be dictated. The estimation search specifications utilized during the grey-box estimation for this study are listed in Table 7.
An additional refinement of the grey-box estimation was performed under the same conditions by the MATLAB built-in function, `pem`, in an attempt to minimize the prediction error estimate of the model parameter values. The estimated parameters were consequently used in a developed algorithm to adjust the rotational equilibrium positions of the model with strategically placed counterbalance masses. This algorithm and its development are detailed in the following section.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimum, Maximum</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>[0, 10]</td>
<td></td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>[0, -]</td>
<td></td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>[0, -]</td>
<td></td>
</tr>
<tr>
<td>Maximum no. of iterations</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Search method</td>
<td>Nonlinear least squares method</td>
<td></td>
</tr>
</tbody>
</table>
E. Equilibrium Adjustment

The parameters of static unbalance in the rotational system estimated through the parameter estimation algorithms described in the previous section were input to the equilibrium adjustment algorithm described in this section. Energy methods were implemented in order to selectively alter the equilibrium positions of the rotating system with counterbalance masses placed in the system.

To begin, recall the equation of motion for this system derived previously listed in Eq. 47. In order to study the effect of placing counterbalance masses on this rotating system, the energy of any additional counterbalance mass was added to the dynamics of the system. Determining the kinetic and potential energies of a counterbalance mass began with describing the location of a counterbalance mass in the Cartesian coordinate system as listed below in Eq. 58.

\[ \mathbf{r}_c = c_1 \mathbf{e}_1 + c_2 \mathbf{e}_2 + c_3 \mathbf{e}_3 \]  

(58)

The derivation of the kinetic energy of a counterbalance mass required the squared value of the radial distance of the mass from the rotational axis. Consequently, the location listed in Eq. 58 was transformed into the \{\mathbf{e}_1^*, \mathbf{e}_2^*, \mathbf{e}_3^*\} coordinate system, shown in Eq. 59.

\[
\begin{bmatrix}
    c_1 \\
    c_2 \cos(45^\circ) + c_3 \sin(45^\circ) \\
    -c_2 \sin(45^\circ) + c_3 \cos(45^\circ)
\end{bmatrix} = \beta_1 \begin{bmatrix}
    c_1 \\
    c_2 \\
    c_3
\end{bmatrix}
\]  

(59)
The resulting radial distance of the counterbalance mass from the rotational axis squared is listed in Eq. 60.

\[ d_c^2 = c_1^2 + [c_2 \cos(45^\circ) + c_3 \sin(45^\circ)]^2 \] (60)

The kinetic energy of the counterbalance mass is shown in Eqs. 61 and 62.

\[ T_c = \frac{1}{2} m_c d_c^2 \left( \frac{d\theta}{dt} \right)^2 \] (61)

\[ T_c = \frac{1}{2} m_c \left[ c_1^2 + \sin^2(45^\circ)(c_2 + c_3)^2 \right] \left( \frac{d\theta}{dt} \right)^2 \] (62)

As with the location of the eccentric mass derived previously, the location of the counterbalance mass through the rotation \( \theta \), depicted in Fig. 26, had to be determined in order to derive its potential energy.

![Fig. 26. Rotation of counterbalance mass about rotational axis.](image)

The resulting position vector for the counterbalance mass in the \( \{e_1', e_2', e_3'\} \) coordinate system is listed in Eq. 63.
Transforming this position vector back into the \( \{e_1, e_2, e_3\} \) coordinate system, shown in Eq. 64, determined the counterbalance mass’ vertical distance from the previously declared datum for calculating its potential energy.

\[
\begin{align*}
    r_c &= \begin{bmatrix}
        c_1 \cos \theta - \sin \theta [c_2 \cos(45^\circ) + c_3 \sin(45^\circ)] \\
        c_1 \sin \theta + \cos \theta [c_2 \cos(45^\circ) + c_3 \sin(45^\circ)] \\
        -c_2 \sin(45^\circ) + c_3 \cos(45^\circ)
    \end{bmatrix}
\end{align*}
\]

(63)

Thus, the vertical displacement of the counterbalance mass at any rotation \( \theta \) about the rotational axis and the potential energy of the counterbalance mass are listed in Eqs. 65-67.

\[
\begin{align*}
    h_c &= \sin(45^\circ) [c_1 \sin \theta + (1 + \cos \theta) \sin(45^\circ) (c_2 + c_3)] \\
    V_c &= m_c g h_c \\
    V_c &= m_c g \sin(45^\circ) [c_1 \sin \theta + (1 + \cos \theta) \sin(45^\circ) (c_2 + c_3)]
\end{align*}
\]

(64)

(65)

(66)

(67)

Further, the Euler-Lagrange formulation of the equation of motion requires the two equations listed in Eqs. 68 and 69.
Finally, the equation of motion of the system with a counterbalance mass incorporated into the system is listed in Eq. 70.

\[
\begin{align*}
\frac{d}{dt} \left( \frac{\partial T_c}{\partial \dot{\theta}} \right) &= m_c \left[ c_1^2 + \sin^2(45^\circ) (c_2 + c_3)^2 \right] \ddot{\theta} \\
\frac{\partial V_c}{\partial \theta} &= c_1 m_c g \sin(45^\circ) \cos \theta - (c_2 + c_3) \sin(45^\circ) \sin \theta
\end{align*}
\]  

\begin{equation}
\tag{68}
( \dot{m} \ddot{p}^2 + J_f + m_c \left[ c_1^2 + \sin^2(45^\circ) (c_2 + c_3)^2 \right]) \frac{d^2 \theta}{dt^2} + c_1 \frac{d\theta}{dt} \nonumber \\
+ m_c g p \sin 45^\circ \cos(\theta + \phi) \nonumber \\
+ c_1 m_c g \sin(45^\circ) \cos \theta - (c_2 + c_3) \sin(45^\circ) \sin \theta \\
= 0
\end{equation}

In order to reach a stationary rotational equilibrium position, both the angular acceleration and the angular velocity had to be equal to zero. The result of this substitution is listed in Eq. 71.

\[
\begin{align*}
m_c g p \sin 45^\circ \cos(\theta + \phi) + c_1 m_c g \sin(45^\circ) \cos \theta - (c_2 + c_3) \sin(45^\circ) \sin \theta &= 0
\end{align*}
\]  

\begin{equation}
\tag{71}
\end{equation}

The incorporation of multiple counterbalance masses would simply require the superposition of each counterbalance mass’ kinetic and potential energy in the previous equations. The numerical algorithm implemented in MATLAB to determine the correct placement of counterbalance masses in the system utilized Eq. 71 to adjust the locations of the system’s stationary rotational equilibrium. The complete MATLAB code used to
determine the appropriate locations for placement of counterbalance masses is listed in Appendix D.

This algorithm begins by declaring some user-defined variables: the number of counterbalance masses that should be applied to the system, the mass of the counterbalances, and whether the counterbalance masses should be located in the upper or lower U-channel. The current algorithm only supports the determination of one counterbalance mass at a time. The desired rotational equilibrium position relative to the current rotational equilibrium position is also declared in the beginning of this algorithm.

As mentioned previously, the terms containing angular acceleration and angular velocity in the equation of motion are eliminated in considering instances of stationary rotational equilibrium. By declaring the vertical location of the counterbalance mass in either the upper or lower U-channel, \( c_2 \), and the mass of the counterbalance mass, \( m_c \), the remaining unknown parameters of the counterbalance became \( c_1 \) and \( c_3 \). The equilibrium adjustment algorithm generated the relationship between these parameters in order to achieve the desired rotational equilibrium position utilizing the relationship listed in Eq. 72.

\[
c_3 = \frac{1}{\sin(45^\circ)\sin \theta} [m_eg p \sin(45^\circ)\cos(\theta + \phi) + c_1m_eg \sin(45^\circ)\cos \theta] - c_2 \quad (72)
\]

Physical constraints were applied to the values for \( c_1 \) and \( c_3 \) based on where the counterbalance mass could physically be applied to the frame. These physical constraints for \( c_1 \) and \( c_3 \) were (-3,3) and (-17.25,17.25), respectively. Appropriate values for \( c_1 \) and \( c_3 \)
were selected from these results and a counterbalance mass was placed in the respective location. Measurement of the adjusted rotational motion of the system was recorded in order to compare the rotational equilibrium position after applying a counterbalance mass to the system to the original rotational equilibrium position.
The current results of this study include thorough verification and simulation of the derived dynamic behavior of this model as well as experimental results analyzed with the estimation methods described previously.

A. Verification and Simulation

A simulation was developed in MATLAB to illustrate the dynamic behavior of the frame and the eccentric mass. The free parameters describing the eccentric mass, $m_e$, $p$, and $\phi$, could be modified as desired to simulate various dynamic movements and equilibrium states. The MATLAB script for this simulation is listed in Appendix D.

An important aspect of this MATLAB simulation was its ability to numerically and visually verify the theoretical equation of motion derived previously. One verification completed with this simulation was that which transformed the gravitational force vector acting on the eccentric mass into the cylindrical coordinate system and confirmed that the magnitude of this vector in both coordinate systems remained equivalent throughout the frame’s rotation. This force vector is listed in the $\{e_1,e_2,e_3\}$ and in the $\{e_p,e_\phi,e_s\}$ coordinate systems in Eqs. 73 and 74, respectively.

$$F_g = -m_e g e_3$$  \hspace{1cm} (73)
The magnitudes of the initial gravity force vector and the transformed gravity force vector were simulated throughout the motion of the model with the free parameters defined as listed in Table 8.

\[
\mathbf{F}_g = \left[ -m_e g \sin(45^\circ)\sin(\phi)\cos(\theta) + m_e g \sin(45^\circ)\cos(\phi)\sin(\theta) \right] \mathbf{e}_p \\
\quad + \left[ -m_e g \sin(45^\circ)\sin(\phi)\sin(\theta) - m_e g \sin(45^\circ)\cos(\phi)\cos(\theta) \right] \mathbf{e}_\phi \\
\quad + \left[ -m_e g \cos(\theta) \right] \mathbf{e}_s
\]

(74)

The magnitudes of the initial gravity force vector and the transformed gravity force vector were both equal to 1,932 lb. throughout the motion of the simulated model. The gravity force vector magnitudes during this rotation are shown in Fig. 27 and the corresponding angular displacement curves for the frame and the eccentric mass are shown in Fig. 28.
As confirmed in Fig. 27, the initial gravity force vector and the transformed gravity force vector are equal in magnitude through rotation of the model about the rotational axis. This figure also displays the individual components of the transformed gravity force vector throughout the rotation of the model.

Fig. 28. Angular displacement curves for simulated model.
Recall from Section II regarding a static unbalance that a “body free to rotate will seek a position where its center of gravity is lowest [4].” Based on the coordinate systems defined during the derivation of the equation of motion in Section III, the initial angular displacement of the eccentric mass relative to the $e_1'$ coordinate axis was the angle $\phi$. The angular displacement curve of the eccentric mass in Fig. 28 reflects this initial position.

The dynamics of a static unbalance dictate that the eccentric mass should be in its lowest position, or its position of least potential, when a rotating object comes to rest. In this case, the eccentric mass should come to rest at at its lowest angular position, $3\pi/2$ [rad] or about 4.71 [rad]. Figs. 29 and 30 illustrate this behavior of the eccentric mass, confirming that the derived equation of motion accurately describes the predicted behavior of this system.

![Fig. 29. Final resting position of frame and eccentric mass with defined parameters.](image)
Fig. 30. Side view of system’s final resting position.

Fig. 30 highlights the eccentric mass in its final position by a red circle. As predicted, the eccentric mass comes to rest at its lowest potential regardless of the position of the frame.

The theoretical model developed for this study also accurately describes the expected equilibrium behavior of this system with the eccentric mass initially positioned at either $\phi = \pi/2 \text{ [rad]}$ or $\phi = 3\pi/2 \text{ [rad]}$, regardless of the mass of the eccentricity. These two simulated scenarios are depicted in Figs. 31 and 32 with the corresponding angular displacement curves of the frame and the eccentric mass.
Fig. 31. Simulation results for $\phi=\pi/2$ [rad]. (a) Visualization. (b) Angular displacement curves.

Fig. 32. Simulation results for $\phi=3\pi/2$ [rad]. (a) Visualization. (b) Angular displacement curves.
Based on these simulation results, the theoretical model derived to describe the physical model for this study provides an accurate prediction of its dynamic behavior according to classical dynamic principles.
B. Experimental Results

The results presented here are based on angular displacement measurements collected from the rotating system with an applied eccentric mass. A physical eccentric mass was applied to the system to eliminate the need for an externally applied force to begin rotation. The dynamic behavior of the system with this applied eccentric mass were recorded and analyzed with the parameter estimation method in order to produce appropriate counterbalance specifications for desired rotational equilibrium positions. The mass and location of the applied eccentric mass is listed in Table 9.

<table>
<thead>
<tr>
<th>Values</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>2.40625</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>13.5726</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>$\pi/2$</td>
</tr>
</tbody>
</table>

There were five iterations of this data collection performed. The transient dynamic behaviors of the system with this applied eccentric mass during each iteration are shown in Fig. 33.
The exponential curves generated by the data analysis algorithm to estimate the damping present in the system for each iteration are shown in Figs. 34-38.
Fig. 35. Exponential damping curve for data set 2.

Fig. 36. Exponential damping curve for data set 3.

Fig. 37. Exponential damping curve for data set 4.
Table 10 lists the damping constants estimated by the data analysis algorithm for each iteration of data collection.

Table 10. Estimated damping constants.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Damping constant [in-lb-s/rad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set 1</td>
<td>2,769.68</td>
</tr>
<tr>
<td>Data set 2</td>
<td>1,326.12</td>
</tr>
<tr>
<td>Data set 3</td>
<td>2,249.59</td>
</tr>
<tr>
<td>Data set 4</td>
<td>1,674.34</td>
</tr>
<tr>
<td>Data set 5</td>
<td>1,580.96</td>
</tr>
</tbody>
</table>

Initial parameter estimations with this collected data yielded inaccurate grey-box estimation model fits to the data. Many modifications were made to the MATLAB script in an effort to improve the numerical results through the course of this study. Unfortunately, one of the external functions utilized during this parameter estimation
contained a dynamic model describing the angular displacement of the system in degrees instead of radians. These initial results are included here for completeness.

Figs. 39-48 display the results of parameter estimation prior to and after resolving the discrepancy in units within the relevant MATLAB script. These results contain both the initial and refined nonlinear grey-box estimations compared to the collected measurements of angular displacement for each iteration of data collection.

![Graph](image)

**Fig. 39.** Nonlinear grey-box estimation for data set 1 with incorrect algorithm.

![Graph](image)

**Fig. 40.** Nonlinear grey-box estimation for data set 1 with correct algorithm.
Fig. 41. Nonlinear grey-box estimation for data set 2 with incorrect algorithm.

Fig. 42. Nonlinear grey-box estimation for data set 2 with correct algorithm.
Fig. 43. Nonlinear grey-box estimation for data set 3 with incorrect algorithm.

Fig. 44. Nonlinear grey-box estimation for data set 3 with correct algorithm.
Fig. 45. Nonlinear grey-box estimation for data set 4 with incorrect algorithm.

Fig. 46. Nonlinear grey-box estimation for data set 4 with correct algorithm.
The final estimates for the free parameters and their standard deviations during the nonlinear grey-box estimation process for each iteration of data collection with the incorrect algorithm and the correct algorithm are listed in Tables 11 and 12, respectively.
Table 11. Nonlinear grey-box free parameter estimates with incorrect algorithm.

<table>
<thead>
<tr>
<th>Data Set 1</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_e ) [lb]</td>
<td>9.9993</td>
<td>4.4047</td>
</tr>
<tr>
<td>( p ) [in]</td>
<td>22.2502</td>
<td>18.9554</td>
</tr>
<tr>
<td>( \phi ) [rad]</td>
<td>-94.7567</td>
<td>0.8002</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 2</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_e ) [lb]</td>
<td>9.9996</td>
<td>3.2931</td>
</tr>
<tr>
<td>( p ) [in]</td>
<td>15.1134</td>
<td>13.4490</td>
</tr>
<tr>
<td>( \phi ) [rad]</td>
<td>-94.5163</td>
<td>0.5882</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 3</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_e ) [lb]</td>
<td>9.9982</td>
<td>4.0100</td>
</tr>
<tr>
<td>( p ) [in]</td>
<td>19.3926</td>
<td>16.1102</td>
</tr>
<tr>
<td>( \phi ) [rad]</td>
<td>-94.7286</td>
<td>0.7114</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 4</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_e ) [lb]</td>
<td>9.9999</td>
<td>3.4591</td>
</tr>
<tr>
<td>( p ) [in]</td>
<td>14.1269</td>
<td>11.4180</td>
</tr>
<tr>
<td>( \phi ) [rad]</td>
<td>-94.9321</td>
<td>0.6690</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 5</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_e ) [lb]</td>
<td>10.0000</td>
<td>4.1309</td>
</tr>
<tr>
<td>( p ) [in]</td>
<td>17.9906</td>
<td>15.9315</td>
</tr>
<tr>
<td>( \phi ) [rad]</td>
<td>-94.3892</td>
<td>0.6141</td>
</tr>
</tbody>
</table>
Table 12. Nonlinear grey-box free parameter estimates with correct algorithm.

<table>
<thead>
<tr>
<th>Data Set 1</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>3.5349</td>
<td>1.0624</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>17.1035</td>
<td>6.6700</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.1249</td>
<td>0.0628</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 2</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>10.0000</td>
<td>24.6393</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>4.1692</td>
<td>11.2055</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0203</td>
<td>0.0590</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 3</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>7.0500</td>
<td>8.0290</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>6.5857</td>
<td>8.2811</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.1286</td>
<td>0.0629</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 4</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>10.0000</td>
<td>23.7123</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>4.3498</td>
<td>11.2254</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0471</td>
<td>0.0590</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set 5</th>
<th>Estimated Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>10.0000</td>
<td>26.8431</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>4.0365</td>
<td>11.6517</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0779</td>
<td>0.0621</td>
</tr>
</tbody>
</table>

Figs. 49 and 50 contain the nonlinear grey-box estimation curves for all of the data sets produced by the incorrect algorithm and the incorrect algorithm, respectively. The legends of these plots contain the percentage fit of each estimation curve to its respective data set.
Fig. 49. Comparison of nonlinear grey-box estimations produced by incorrect algorithm for all data sets.

Fig. 50. Comparison of nonlinear grey-box estimations produced by correct algorithm for all data sets.
Correcting the discrepancy in the parameter estimation algorithm made a large impact on the results of this estimation process and allowed the developed rotational equilibrium adjustment algorithm to be implemented with the collected data.

Recall that the original intent of this study was to develop a systematic approach for selectively modifying the stationary equilibrium positions of the rotating system of interest. A rotational equilibrium adjustment was performed using the damping constant and estimated free parameters from the first data set produced with the correct parameter estimation algorithm, listed respectively in Tables 10 and 12. To demonstrate the ability of this rotational equilibrium adjustment algorithm, two rotational equilibrium position adjustments, 2 [deg] (0.0873 [rad]) in the positive \( \theta \) direction and 5 [deg] (0.1222 [rad]) in the positive \( \theta \) direction were performed. Table 13 lists the counterbalance specifications determined by this algorithm for the positive 2 [deg] equilibrium adjustment and Fig. 51 illustrates this positive 2 [deg] rotational equilibrium adjustment.

Table 13. Counterbalance specifications for +2 [deg] equilibrium adjustment.

<table>
<thead>
<tr>
<th>( m_c ) [lb]</th>
<th>( c_1 ) [in]</th>
<th>( c_2 ) [in]</th>
<th>( c_3 ) [in]</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-2.70</td>
<td>-2.79</td>
<td>14.6315</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 51. +2 [deg] rotational equilibrium adjustment for data set 1.

Table 14 lists the counterbalance specifications determined by this algorithm for the positive 5 [deg] equilibrium adjustment and Fig. 52 illustrates this positive 5 [deg] rotational equilibrium adjustment.
Table 14. Counterbalance specifications for +5 [deg] equilibrium adjustment.

<table>
<thead>
<tr>
<th></th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_c$ [lb]</td>
<td>10</td>
</tr>
<tr>
<td>$c_1$ [in]</td>
<td>-2.52</td>
</tr>
<tr>
<td>$c_2$ [in]</td>
<td>2.79</td>
</tr>
<tr>
<td>$c_3$ [in]</td>
<td>9.4713</td>
</tr>
</tbody>
</table>

Fig. 52. +5 [deg] rotational equilibrium adjustment for data set 1.

This rotational equilibrium adjustment algorithm succeeds in producing appropriate counterbalance specifications to shift the stationary rotational equilibrium positions as desired. Although the experimental methodology detailed for the purpose of this study
addressed the relative shift of stations rotational equilibrium positions for the rotating system of interest, an ideal adjustment algorithm would produce a rotating system balanced in any rotational position. The physical constraints of this rotating system in where counterbalance masses could be placed on the rotating frame decreases the solution space available for completely balancing the rotational system. Complete equilibrium may be possible in the presence of an eccentricity by applying counterbalance masses, but if these counterbalance masses can not be placed on the physical frame only a shift in the rotational equilibrium positions is applicable.
VI. Discussion and Conclusions

This study showed promising results in the development of an adaptable, accurate, and portable rotational equilibrium adjustment system. However, several modifications to the experimental methodology could improve the precision and accuracy of this adjustment system.

The simulation results of the theoretical model developed to describe the rotating system of interest accurately portrayed the behavior of the rotating system and verified the derived dynamic model of the system for numerical estimation. Although the theoretical dynamic model was dependable, the physical test fixture needs improved stability at its base to prevent inaccurate angular displacement measurements during the data collection process.

The results of the parameter estimation method involving MATLAB’s nonlinear grey-box estimation model showed feasible but inconsistent results. This inconsistency could be seen in the free parameters estimated for each individual data set presented in the previous section. Not only were the estimated values of the free parameters inconsistent but the standard deviation of the values during the search method were also inconsistent. For instance, the estimated value of the eccentric mass ranged from 3.5349 [lb] to 10.0000 [lb] with standard deviations of 1.0624 [lb] and 26.8431 [lb], respectively.
Table 15 lists the average values for each free parameter estimated by the nonlinear grey-box model as well as their range.

Table 15. Analysis of nonlinear grey-box estimation free parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Average</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_e$ [lb]</td>
<td>8.1170</td>
<td>3.5349</td>
<td>10.0000</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>7.2490</td>
<td>4.0365</td>
<td>17.1035</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0798</td>
<td>-0.1286</td>
<td>-0.0203</td>
</tr>
</tbody>
</table>

Additionally, the percent errors of the parameter estimates for each data set from the applied eccentric mass are listed in Table 16.
Table 16. Percent error of parameter estimates.

<table>
<thead>
<tr>
<th>Parameter Estimate</th>
<th>Applied Eccentricity</th>
<th>Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_e$ [lb]</td>
<td>3.5349</td>
<td>1.06</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>17.1035</td>
<td>6.67</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.1249</td>
<td>0.0628</td>
</tr>
<tr>
<td>Data Set 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_e$ [lb]</td>
<td>10.0000</td>
<td>24.6</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>4.1692</td>
<td>11.2</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0203</td>
<td>0.0590</td>
</tr>
<tr>
<td>Data Set 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_e$ [lb]</td>
<td>7.0500</td>
<td>8.03</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>6.5857</td>
<td>8.28</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.1286</td>
<td>0.06</td>
</tr>
<tr>
<td>Data Set 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_e$ [lb]</td>
<td>10.0000</td>
<td>23.7</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>4.3498</td>
<td>11.2</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0471</td>
<td>0.0629</td>
</tr>
<tr>
<td>Data Set 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m_e$ [lb]</td>
<td>10.0000</td>
<td>26.8</td>
</tr>
<tr>
<td>$p$ [in]</td>
<td>4.0365</td>
<td>11.6</td>
</tr>
<tr>
<td>$\phi$ [rad]</td>
<td>-0.0779</td>
<td>0.0621</td>
</tr>
</tbody>
</table>

The root cause of these discrepancies needs to be further understood in order to accurately identify the eccentricity in the rotating system of interest. A local sensitivity analysis could assist in evaluating the best initial guesses for each free parameter in order to reduce the variation in parameter estimation. Creating a more stable test fixture would also aid in reducing these discrepancies by allowing the collection of more consistent data.
angular displacement measurements. With more time, this topic can be addressed and improved significantly.

The rotational equilibrium adjustment algorithm implemented for this study can also be improved and its results verified in further studies. A simple improvement that needs to be incorporated into this adjustment algorithm is its ability to consider multiple counterbalance masses as an option in altering the system’s equilibrium behavior. The algorithm currently only outputs the specifications of one counterbalance mass. The incorporation of multiple counterbalance masses could be completed in one iteration or in subsequent iterations of the algorithm.

As mentioned previously, the physical constraints on where counterbalance masses could be placed on the rotating frame limited the solution space for the rotational equilibrium adjustment algorithm. As a result, the adjustment algorithm could potentially return no counterbalance solution for the desired equilibrium adjustment. The user currently has to slightly adjust the initial values of counterbalance mass or location in order to produce a solution that is physically possible to place on the rotating frame. Ideally, this rotational equilibrium adjustment algorithm needs to recognize that the search for a feasible counterbalance mass must be completed again.

Finally, the counterbalance specifications output by the rotational equilibrium adjustment algorithm need to be verified. The results were collected and simulated during this study, but experimental verification of the results were not completed in the period of time this
study was completed. Although the simulated results indicated that the counterbalance mass adjusted the rotating system’s equilibrium behavior accordingly, other physical factors could contribute to the equilibrium behavior of the system, such as lubrication methods and axis alignment.

Evaluating the parameters of unbalance with the existing test fixture for this study poses a tedious problem as it is an imperfect rotational system. Its behavior is not as easily adaptable as the computational model developed to analyze and simulate the dynamics of the system. Several steps need to be taken in continuing this study in the coming weeks to inch closer to a parametric solution for controlling rotational behavior of kinetic systems such as that in the S-337473 exhibition.
References


Appendix A: Arduino Sketches
Arduino sketch for initial measurement system:

/*  Maura O'Neill
 *  6/22/17
 *
 *  Arduino sketch to control Arduino Uno, MPU-6050 inertial
 *  measurement unit, and Adafruit Data Logger Shield
 *
 *  This sketch is modified from an example sketch provided by:
 *
 *  FIGURE OUT WHICH SKETCH THIS IS SUPPOSED TO BE
 */

const int buttonPin = 6;
const int buttonPressed = LOW;
const int buttonNotPressed = HIGH;
int buttonState = 0;
const int ledPin = 7;
int count = 0;
float valueTwo = 0;
float valueOne = 0;
float diff = 1;
float averageDiff = 0;
int recording = 0;
float startTime = 0;
float endTime = 0;
unsigned long microsPerReading, microsPrevious;
float accelScale, gyroScale;
int counter = 1;
float time1, time2;
float roll, pitch, heading;

// include the SD library:
#include <SPI.h>
#include <SD.h>
#include <MadgwickAHRS.h>
#include <math.h>

Madgwick filter;

#define redLEDpin 3

// set up variables using the SD utility library functions:
Sd2Card card;
SdVolume volume;
SdFile root;

// change this to match your SD shield or module;
// Adafruit SD shields and modules: pin 10
const int chipSelect = 10;

File logfile;

// I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h files
// for both classes must be in the include path of your project
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"

// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for SparkFun breakout and InvenSense evaluation board)
// AD0 high = 0x69
//MPU6050 accelgyro;
MPU6050 accelgyro(0x69); // <-- use for AD0 high

int16_t aix, aiy, aiz;
int16_t gix, giy, giz;

#define LED_PIN 13 // (Arduino is 13, Teensy is 11, Teensy++ is 6)
bool blinkState = false;

void setup() {
  pinMode(4,OUTPUT);
  digitalWrite(4,HIGH);
  pinMode(buttonPin,INPUT);
  pinMode(ledPin, OUTPUT);
  filter.begin(25);

  // Open serial communications and wait for port to open:
  // initialize serial communication
  // (115200 chosen because it is required for Teapot Demo output,
  // but it's
  // really up to you depending on your project)
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  Serial.print("Initializing SD card...");

  // see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    setup();
    return;
}
Serial.println("card initialized.");

#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
#endif

// initialize device
Serial.println("Initializing I2C devices...");
accelgyro.initialize();

// verify connection
Serial.println("Testing device connections...");
Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");

// create a new file
char filename[] = "MOTION00.csv";
for (uint8_t i = 0; i < 100; i++) {
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (! SD.exists(filename)) {
        // only open a new file if it doesn't exist
        logfile = SD.open(filename, FILE_WRITE);
        break; // leave the loop!
    }
}
if (! logfile) {
    error(“couldn’t create file”);
}

Serial.print("Logging to: ");
Serial.println(filename);

// wait for ready
Serial.println(F("\nSend any character to begin data collection: "));
while (Serial.available() && Serial.read()); // empty buffer
while (!Serial.available()); // wait for data
while (Serial.available() && Serial.read()); // empty buffer again

pinMode(LED_PIN, OUTPUT);

microsPerReading = 1000000 / 25;
microsPrevious = micros();
time1 = millis();

void loop() {
    float ax, ay, az;
    float gx, gy, gz;
    unsigned long microsNow;

    microsNow = micros();
    if (microsNow - microsPrevious >= microsPerReading) {

        // read raw accel/gyro measurements from device
        accelgyro.getMotion6(&aix, &aix, &aiz, &gix, &giy, &giz);

        gx = convertRawGyro(gix);

        Serial.print("GX: "); Serial.println(gx);
        Serial.print("GY: "); Serial.println(gy);
        Serial.print("GZ: "); Serial.println(gz);
        Serial.print("AX: "); Serial.println(ax);
        Serial.print("AY: "); Serial.println(ay);
        Serial.print("AZ: "); Serial.println(az);

        microsPrevious = microsNow;
    }
}
gy = convertRawGyro(giy);
gz = convertRawGyro(giz);

Serial.print(gx);
Serial.print(", 	");
Serial.print(gy);
Serial.print(", 	");
Serial.print(gz);
Serial.println(", 	");

// Serial.println((String)((millis() - time1)/1000.00) + "," + (String)heading + "," + (String)pitch + "," + (String)roll);

microsPrevious = microsPrevious + microsPerReading;
counter++;
}

float convertRawAcceleration(int aRaw) {
    // since we are using 2G range
    // -2g maps to a raw value of -32768
    // +2g maps to a raw value of 32767

    float a = (aRaw * 2.0) / 32768.0;
    return a;
}

float convertRawGyro(int gRaw) {
    // since we are using 250 degrees/seconds range
    // -250 maps to a raw value of -32768
    // +250 maps to a raw value of 32767

    float g = (gRaw * 250.0) / 32768.0;
    return g;
}
void error(char *str)
{
    Serial.print("error: ");
    Serial.println(str);

    // red LED indicates error
    digitalWrite(redLEDpin, HIGH);

    while(1);}
Arduino sketch for final measurement system:

/*  Maura O'Neill  
   *  6/22/17  
   *  
   *  Arduino sketch to control Arduino Board 101 and on-board IMU  
   *  with an HC-05 bluetooth module for wireless data collection.  
   *  
   *  This sketch is modified from an example sketch provided by [11].  
   *  
   *  
   */

// Including all relevant libraries
#include <SoftwareSerial.h>
#include <BMI160.h>
#include <CurieIMU.h>
#include <MadgwickAHRS.h>

// Declaring all relevant variables
Madgwick filter;
unsigned long microsPerReading, microsPrevious;
float accelScale, gyroScale;
SoftwareSerial btSerial(10, 11);
int counter = 1;
float time1, time2;
int calc = 0;
float rollRate = 0;
float pitchRate = 0;
float headingRate = 0;
float rollPrevious = 0;
float pitchPrevious = 0;
float headingPrevious = 0;
void setup() {
    CurieIMU.begin();
    Serial.begin(9600);
    btSerial.begin(9600);

    CurieIMU.setGyroRate(25); // determines sampling frequency of the gyro in Hz
    CurieIMU.setAccelerometerRate(25); // determines sampling frequency of accelerometer in Hz
    filter.begin(25); // determines sampling frequency of Madgwick filter in Hz

    CurieIMU.setAccelerometerRange(2); // declares range of accelerometer to be +/-2g
    CurieIMU.setGyroRange(250); // declares range of gyrometer to be +/-250 degrees per second

    microsPerReading = 1000000 / 1000;
    microsPrevious = micros();
    time1 = millis();
}

void loop() {
    int aix, aiy, aiz;
    int gix, giy, giz;
    float ax, ay, az;
    float gx, gy, gz;
    float roll, pitch, heading;
    unsigned long microsNow;

    // check if it's time to read data and update the filter
    microsNow = micros();
    if (microsNow - microsPrevious >= microsPerReading) {

        // read raw data from inertial measurement unit
        CurieIMU.readMotionSensor(aix, aiy, aiz, gix, giy, giz);

        // process the data (e.g., applying filter)
    }
}
// convert from raw data to gravity and degrees/second units
ax = convertRawAcceleration(aix);
ay = convertRawAcceleration(aiy);
az = convertRawAcceleration(aiz);
gx = convertRawGyro(gix);
gy = convertRawGyro(giy);
gz = convertRawGyro(giz);

// update the filter, which computes orientation of the inertial measurement unit
filter.updateIMU(gx, gy, gz, ax, ay, az);

// collect the roll, pitch, and heading
roll = filter.getRoll(); // this block ensures that angles are measured between 0 and 360 degrees
if (roll < -100) {
    roll = roll + 360;
}
pitch = filter.getPitch();
heading = filter.getYaw();

// send inertial measurement unit data to bluetooth module
btSerial.println((String)((millis() - time1) / 1000.00) + "," + (String)heading + "," + (String)pitch + "," + (String)roll + "," + (String)gx + "," + (String)gy + "," + (String)gz);

// increment previous time to keep proper pace
microsPrevious = microsPrevious + microsPerReading;
counter++;
calc++;
}

float convertRawAcceleration(int aRaw) {
    // 2g range
float a = (aRaw * 2.0) / 32768.0;
return a;
}

float convertRawGyro(int gRaw) {
    // 250 degrees/seconds range
    // -250 maps to a raw value of -32768
    // +250 maps to a raw value of 32767

    float g = (gRaw * 250.0) / 32768.0;
    return g;
}
Appendix B: Circuit Schematics
Fig. 53. Circuit schematic of HC-05 Bluetooth Module and Arduino Board 101.
Appendix C: Test Fixture Drawings
Fig. 54. Dimensioned drawing of T-slotted framing.
Fig. 55. Dimensioned drawing of diagonal T-slotted frame piece.
Fig. 56. Dimensioned drawing of T-slotted framing corner bracket.
Fig. 57. Dimensioned drawing of T-slotted framing attachment hardware.
Fig. 58. Dimensioned drawing of needle roller bearing.
Fig. 59. Dimensioned drawing of stationary tube.
Fig. 60. Dimensioned drawing of shaft.

Note: Needle roller bearing press fit into one end of shaft.
Fig. 61: Dimensioned drawing of shaft attachment piece.
Fig. 62. Dimensioned drawing of glass.
Fig. 63. Dimensioned drawing of U-channel.
Fig. 64. Dimensioned drawing of T-slotted counterbalance mounting bar.
Fig. 65. Dimensioned drawing of battery and BLE holder.
Fig. 66. Dimensioned drawing of electronics shaft mount.
Table 17. Bill of materials for test fixture.

<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>DESCRIPTION</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T-slot mounting hardware</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>T-slot frame piece</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>T-slot corner bracket</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Trimmed T-slot frame piece</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Glass and U-channel assembly</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Needle Roller Bearing</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Stationary tube</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>T-slot mounting bar</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>Shaft attachment piece</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Shaft</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>Diagonal T-slot bar</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>T-slot screw and washer</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>Electronics mounting assembly</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>Thrust bearing B</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>Thrust bearing A</td>
<td>1</td>
</tr>
</tbody>
</table>
Fig. 67. Exploded view of test fixture.
Appendix D: MATLAB Code
Data analysis MATLAB script:

% Maura O'Neill

% MATLAB code to plot collected measurements, 
% estimate damping ratio and damping constant, 
% and fit the data to the derived state-space model 
% using a grey-box estimation and additional model refinement

% clear command window, all variables, all previous plots
clc
clear all
close all

%% plotting data from Excel file
kest = 1;
for k = 4:3:16
    % read measurement data from selected Excel file
    T = xlsread('Small Comparison.xlsx');
    
    % create vector of data collection times
    timeEst = T(:,1);
    time = timeEst(1:length(timeEst)); % use beginData to select first
    startTime = time(1);
    timeDiff = time(2)-startTime;
    
    % loop to create time vector with initial time t = 0
    for i = 1:length(time)
        timePos(i) = time(i)-startTime;
    end

    % correcting point that goes backward in time
for i = 2:1:length(timePos)
    if timePos(i)<timePos(i-1)
        timePos(i) = timePos(i-1)+timeDiff;
    end
end

% create vector of angular displacement values from collected data
rollEst1 = T(:,k); % units are [deg]
roll = rollEst1(1:length(rollEst1));
% eliminate initial stationary displacement data
% rollEst1 = rollEst1(30:length(rollEst1));

negative = 0;
for i = 1:1:length(roll)
    if roll(i)<0 || negative == 1
        negative = 1;
        %roll(i) = -roll(i);
        roll(i) = roll(i)+360;
    end
end

alpha = 0.9; % weight of exponential filter

% converting angular position into radians from degrees
for i = 1:1:length(roll)
    rollRad(i) = ((roll(i)-90)/360)*2*pi;
end

for i = 1:1:length(rollEst1)
    rollEst1(i) = (rollEst1(i)/360)*2*pi;
end
lastRoll = rollRad(length(rollRad));
% settling position of frame to use in calculations for
damping ratio
%
% calculating angular velocity about shaft [deg/s]
for i = 1:1:(length(roll)-1)
    velocityRad(i,1) = (rollRad(i+1)-rollRad(i))/(timePos(i+1)-timePos(i));
    timeVel(i,1) = timePos(i);
end
%
% filtering angular velocity with exponential filter
velocityRadFil(1,1) = velocityRad(1,1);
for i = 2:1:length(velocityRad)
    velocityRadFil(i,1) = alpha*velocityRadFil(i-1)+(1-alpha)*velocityRad(i);
end
%
% create new vectors for angular displacement, angular velocity, and time
% for the calculation of the damping ratio
decrementDisp = rollRad;
decrementVel = velocityRadFil;
decrementTime = timePos;
%
% plotting angular position and filtered angular velocity
figure('units','normalized','outerposition',[0 0 1 1])
subplot(3,2,1)
plot(timePos,rollRad,'r',timeVel,velocityRadFil,'k')
legend('Angular Displacement [rad]', 'Filtered Angular Velocity [rad/s]', 'Location', 'southeast')
title('Unbalanced Motion')
xlabel('Time [s]')
% using a logarithmic decrement to estimate the damping ratio

peakCount = 0; % number of local maxima in angular displacement
movementStarts = 0;
positiveSlope = 1; % variable to determine if for loop is on a positive or
% negative slope of the angular displacement curve

% for loop determines the local maxima in the angular displacement curve
for i = 2:1:length(decrementDisp)
    if decrementDisp(i) < decrementDisp(i-1) && positiveSlope == 1
        positiveSlope = 0;
        peakCount = peakCount + 1;
        peak(peakCount) = decrementDisp(i-1)-lastRoll; %-
lastRoll;
        timePeak(peakCount) = decrementTime(i-1);
    end
    if decrementDisp(i) > decrementDisp(i-1) && positiveSlope == 0
        positiveSlope = 1;
    end
    if peakCount == 2
        break;
    end
end

% for explicitly stating the peak locations/values
% timePeak = [1.69 5.41];
% peak = [(7.222-lastRoll) (6.2509-lastRoll)];
% timePeak(2) = 4.81;
% %peak(2) = -0.03281-lastRoll; %-lastRoll;
% peak(2) = 0.3;

decrement = log(peak(1)/peak(2)); % calculating logarithmic decrement
% calculating damping ratio
dampRatio = decrement/sqrt((2*pi)^2+decrement^2);
dampPeriod = timePeak(2)-timePeak(1); % calculating damped period
dampFreq = 2*pi/dampPeriod; % calculating damped frequency
% calculating natural frequency with damped frequency and damping ratio
natFreq = dampFreq/sqrt(1-dampRatio^2);
% determining initial value of exponential curve
X = sqrt(((decrementDisp(1)-lastRoll)^2+...
 (decrementVel(1)+dampRatio*natFreq*(decrementDisp(1)-lastRoll))/...
 (sqrt(1-dampRatio^2)*natFreq))^2);

% calculating exponential curve for underdamped system
for i = 1:1:length(decrementDisp)
    logDec(i) = X*exp(-dampRatio*natFreq*decrementTime(i)) +lastRoll;
end

% verifying correct logarithmic decrement was performed and plotting
% exponential curve on plot of angular displacement
subplot(3,2,2)
hold on
points = scatter([timePeak(1) timePeak(2)],...
 [peak(1)+lastRoll peak(2)+lastRoll],...
 'MarkerEdgeColor',[0 0 0],...
% estimating mass moment of inertia of model

Jtot = 0;

%m mass information for U-channel and glass using expected
dimensions and measured mass
mS = 0.099025464074912*(34.5*2.33*5.42); %[lb]
mL = 0.099025464074912*(34.5*2.5*6);
mG = 6.2536;
mSh = 0.4194;

% calculating inertia tensor for U-channels about x,y,z
IxxU = ((1/12)*mL*(34.5^2+2.33^2)+mL*0.4527^2)-
((1/12)*mS*(34.5^2+2.33^2)+mS*0.5377^2);
IyyU = ((1/12)*mL*(2.33^2+6^2)+mL*0.4527^2)-
((1/12)*mS*(2.5^2+6^2)+mS*0.5377^2);
IzzU = (1/12)*mL*(34.5^2+6^2)-(1/12)*mS*(34.5^2+5.42^2);
IU = [IxxU 0 0; 0 IyyU 0; 0 0 IzzU];
\[ R = \begin{bmatrix} 1 & 0 & 0; 0 & \cosd(45) & \sind(45); 0 & -\sind(45) & \cosd(45) \end{bmatrix} \]; % rotation matrix of x,y,z rotated around x by 45 degrees

\[ RT = R'; \]

\[ IU\_rot = R*IU*RT; \] % calculating new inertia tensor for bottom U-channel

\[ JU = IU\_rot(3,3)+(mL-mS)*(2.7973*sind(45))^2; \]

\[ Jtot = Jtot + 2*JU; \]

% calculating inertia tensor for glass about x,y,z
\[ IxxG = \frac{1}{12}*mG*(9^2+34.5^2); \]
\[ IyyG = \frac{1}{12}*mG*(9^2+0.25^2); \]
\[ IzzG = \frac{1}{12}*mG*(34.5^2+0.25^2); \]
\[ IG = [IxxG 0 0; 0 IyyG 0; 0 IzzG]; \]
\[ IG\_rot = R*IG*RT; \]
\[ JG = IG\_rot(3,3)+mG*3.125^2; \]
\[ Jtot = Jtot+2*JG; \]

% calculating mass moment of inertia of shaft
\[ JS = 0.5*mSh*((1.25/2)^2+(1.5/2)^2); \]

\[ Jtot = Jtot+2*JS; \]
\[ Jtot = Jtot + 4*81.19 + 2*0.67; \]

%% trying to estimate coefficients of ODE

% fixed parameters
\[ g = 32.26*12; \] % [in/s^2]
\[ c\_t = 2*dampRatio*(Jtot)*natFreq; \]
\[ %c\_t = 200; \]
\%c_{\text{min}} = 2 \ast \text{dampRatio} \ast (J_{\text{tot}}) \ast \text{natFreq};
init_{\text{pos}}(\text{kest}) = \text{decrementDisp}(1);
init_{\text{vel}}(\text{kest}) = \text{decrementVel}(1);
data_{\text{period}} = 0.04; \% \text{sampling period} \text{[s]}

\% \text{free parameters}
\text{m}_{E} = 2.40625; \% \text{[lb]}
p = 13.5726; \% \text{[in]}
\phi = \pi/2; \% \text{initial angular position of eccentric mass relative to}
\% e_{1}' \text{ unit vector} \text{[rad]}

\% \text{selecting data to be fit for grey-box estimation}
decrementDisp = \text{decrementDisp}';
data = \text{iddata}(\text{decrementDisp},[\text{},\text{data}_{\text{period}}]);
data.\text{OutputName} = '\text{Angular Displacement [rad]}';
data.\text{OutputUnit} = '\text{rad}';
data.\text{Tstart} = 0;

\text{order} = [1 \ 0 \ 2];
\text{parameters} = \{g, J_{\text{tot}}, c_t, m_{E}, p, \phi\}; \% \text{parameters for grey-box estimation}
\text{initial\_states} = [\text{init\_pos(kest)}; \text{init\_vel(kest)}]; \% \text{initial angular position}
\% \text{and initial angular velocity}
\text{Ts} = 0; \% \text{start time}
\% \text{create nonlinear grey-box model}
\text{nonlinear\_model} = \text{idnlgrey('NonlinearRotationWrong',order,parameters,...}
\% \text{initial\_states,Ts});

\% \text{parameter settings for grey-box model}
\% \text{set gravity, mass moment of inertia of frame, and damping as fixed}
% parameters
setpar(nonlinear_model,'Fixed',{true true true false false false});

% when allowing damping constant to be a free parameter, make sure it is no
% less than the damping constant calculated previously
% nonlinear_model.parameters(3).Minimum = 0;

% set minimum and maximum for mass of eccentricity
nonlinear_model.parameters(4).Minimum = 0;
nonlinear_model.parameters(4).Maximum = 10;

% set minimum for radial distance of eccentricity
nonlinear_model.parameters(5).Minimum = 0;

% set options for nonlinear grey-box estimation
opt = nlgreyestOptions;
opt.Display = 'on';
opt.SearchOption.MaxIter = 50;
opt.SearchMethod = 'lsqnonlin'; % specify that nonlinear
least squares
% method be used for fitting algorithm
nonlinear_model = nlgreyest(data,nonlinear_model,opt);

% collect parameters from grey-box estimation
%c_t = nonlinear_model.Parameters(3).Value; % uncomment this
when damping
% constant is free parameter
m_E = nonlinear_model.Parameters(4).Value;
p = nonlinear_model.Parameters(5).Value;
phi = nonlinear_model.Parameters(6).Value;

% print estimated parameters and standard deviation
[nonlinear_vars, dnonlinear_vars] =
getpvec(nonlinear_model,'free')

% plot collected data and initial grey-box estimation
subplot(3,2,3)
compare(decrementDisp,nonlinear_model)

% plot collected data and refined grey-box estimation
updated_model = pem(nonlinear_model,data)
subplot(3,2,4)
compare(decrementDisp,nonlinear_model,updated_model)

% print newly estimated parameters and standard deviation
[updated_vars, dupdated_vars] = getpvec(updated_model,'free')

% uncomment if damping ratio is free parameter
% c = updated_vars(1)
% m = updated_vars(2)
% pdist = updated_vars(3)
% pangle = updated_vars(4)

% comment if damping ratio is free parameter
 c(kest) = c_t
m(kest) = updated_vars(1)
pdist(kest) = updated_vars(2)
pangle(kest) = updated_vars(3)

% solve differential equation with ode45 with updated parameters

timeLength = timePos(length(timePos))+10;
tspan = 0:data_period:timeLength; % time interval for solution
s0_initial = [init_pos(kest) init_vel(kest)]; % initial conditions

options = odeset('reltol',1e-6,'abstol',1e-6);
[t,r] = ode45(@(t,r) TestingMotion(t,r,g,Jtot,c(kest),m(kest),pdist(kest),pangle(kest)), tspan, s0_initial, options);
estimate(:,kest) = r(:,1);

subplot(3,2,5)
plot(t,r(:,1),'r',timePos,decrementDisp,'k')
legend('ODE45 Solution','Measured Angular Displacement','Location','southeast')

kest = kest+1;
end

figure
plot(t,estimate(:,1),t,estimate(:,2),t,estimate(:,3),t,estimate(:,4),t,estimate(:,5))
legend('Run 1 - -4.46%','Run 2 - -17.98%','Run 3 - -1.04%','Run 4 - -24.65%','Run 5 - -5.28%','Location','southeast')
xlabel('Time [s]')
ylabel('Estimated Angular Displacement [rad]')
title('Grey-box Model Estimations for All Runs')

%% numerically locating counterbalance

numCountBal = 1; % number of counterbalance masses
mC = 5; % mass of counterbalance mass [lb]
% yC = 2.79; % for upper U-channel [in]
yC = -2.79; % for lower U-channel [in]
angleChange = 5; % [deg]
angleChangeRad = (2/360)*2*pi; % [rad]
dEq = estimate(length(estimate(:,1)),1)+angleChangeRad;

% calculate range of zC coordinate based on constraint of xC coordinate
xCrange = -3:0.01:3;
for i = 1:length(xCrange)
    zCrange(i) = (1/sind(45)/sin(dEq))*(m(1)*g*pdist(1)*sind(45)*cos(dEq+pangle(1))... +xCrange(i)*mC*g*sind(45)*cos(dEq))-yC;
end

% find the physically possible locations of counterbalance masses on frame
for i = 1:length(xCrange)
    if xCrange(i) < -3 || xCrange(i) > 3 || zCrange(i) < (-34.5/2) || zCrange(i) > 34.5/2
        xCrange(i)=0;
        zCrange(i)=0;
    end
end

j = 1;
for i = 1:length(xCrange)
    if xCrange(i) ~= 0
        xCrangeNew(j) = xCrange(i);
        zCrangeNew(j) = zCrange(i);
        j=j+1;
    end
end

figure
subplot(1,2,1)
plot(xCrange,zCrange)
axis([-3 3 -34.5/2 34.5/2])

xC = xCrangeNew(1);
zC = zCrangeNew(1);

% determine motion with counterbalance mass(es) applied to verify that the
% desired equilibrium position is achieved
timelength = timePos(length(timePos))+10;
tspan = 0:data_period:timelength;
init_cond = [init_pos(1) init_vel(1)];
options = odeset('reltol',1e-6,'abstol',1e-6);
[t,thetaCountBal] = ode45(@(t,s)
    TestingMotionWithCountBal(t,s,g,Jtot,c(1),m(1),pdist(1)...
        ,pangle(1),mC,xC,yC,zC),tspan,init_cond,options);

for i = 1:length(thetaCountBal)
    dEqGoal(i) = dEq;
end

subplot(1,2,2)
plot(t,estimate(:,1),t,thetaCountBal(:,1),t,dEqGoal)
%plot(t,estimate(:,1),t,dEqGoal)
legend('Box Angular Position','Angular Position with Counterbalance','Intended New Equilibrium Position')

%% functions

function result = TestingMotion(t,s,g,Jtot,c,m,pd,pa)
    result = zeros(2,1);
    result(1) = s(2);
result(2) = (-c*s(2)-m*g*pd*sind(45)*cos(pa+s(1)))/
(Jtot+m*pd^2);
end

function result =
TestingMotionWithCountBal(t,s,g,Jtot,ct,mE,pd,pa,mC,xC,yC,zC)
result = zeros(2,1);
result(1) = s(2);
result(2) = (-ct*s(2)-mE*g*pd*sind(45)*cos(pa+s(1))-
xC*mC*g*sind(45)*cos(s(1))...
+(yC+zC)*sind(45)*sin(s(1)))/(Jtot+mE*pd^2+mC*(xC^2+
(sind(45))^2*...
(yC+zC)^2));
end
Grey-box estimation model file:

% Maura O’Neill
% Model file used to describe output equations and
% state equations for nonlinear grey-box estimation
function [dx,y] = NonlinearRotation(t,x,u,g,Jtot,c_t,m_E,p,phi,varargin)

    % output equation
    y = x(1);
    % state equations
    dx =
    [x(2); ... % angular position
        (-c_t*x(2)-m_E*g*p*sind(45)*cos(phi+x(1)))/...
        (Jtot+m_E*p^2) ... % angular velocity
        ];
end
Gravity magnitude verification MATLAB script:

% Maura O'Neill

% MATLAB code to simulate the motion of the frame and the eccentric mass
% Parameters of unbalance can be modified accordingly

%% plotting data from Excel file

% clear command window, all variables, all previous plots
clc
clear all
close all

% read measurement data from selected Excel file
T = xlsread('TESTING3_8MAY17.xlsx');

% eliminate measurements of stationary frame before rotation
% by selecting beginning data point
beginData = 880;

% create vector of data collection times
time = T(:,1);
timeLength = length(time); % length of time vector
for i = 1:1:length(time)
    time(i) = time(i) - startTime;
end

% loop to determine time vector with initial time t = 0
for i = 1:1:length(time)
timePos(i) = time(i)-startTime;
end

% correcting point that goes backward in time
for i = 2:length(timePos)
    if timePos(i)<timePos(i-1)
        timePos(i) = timePos(i-1)+timeDiff;
    end
end
end

roll = T(:,4);
roll = roll(beginData:length(roll));

for i = 1:length(roll)
    if roll(i)<0
        roll(i) = -roll(i);
    end
end

% figure
% subplot(2,2,1)
% plot(timePos,roll,'k')

% simulation

n = 1000; % number of drawings
ct = 300; % damping constant
g = 32.2*12; % gravity in [in/s^2]
mE = 5; % mass of eccentricity
pd = 15; % normal distance of eccentricity from rotational axis
pa = 95; % initial angle of eccentricity
Jtot = 645.22; % total mass moment of inertial including U-channels, glass, etc.

z = 2;

% mass information for U-channel and glass using expected dimensions and measured mass
mS = 0.099025464074912*(34.5*2.33*5.42); %[lb]
mL = 0.099025464074912*(34.5*2.5*6);
mG = 6.2536;
mSh = 0.4194;

% calculating inertia tensor for U-channels about x,y,z
IxxU = ((1/12)*mL*(34.5^2+2.5^2)+mL*0.4527^2)-((1/12)*mS*(34.5^2+2.33^2)+mS*0.5377^2);
IyyU = ((1/12)*mL*(2.5^2+6^2)+mL*0.4527^2)-((1/12)*mS*(2.33^2+5.42^2)+mS*0.5377^2);
IzzU = (1/12)*mL*(34.5^2+6^2)-(1/12)*mS*(34.5^2+5.42^2);
IU = [IxxU 0 0;0 IyyU 0;0 0 IzzU];

R = [1 0 0;0 cosd(45) sind(45);0 -sind(45) cosd(45)]; % rotation matrix of x,y,z rotated around x by 45 degrees

RT = R';

IU_rot = R*IU*RT; % calculating new inertia tensor for bottom U-channel

JU = IU_rot(3,3)+(mL-mS)*(2.7973*sind(45))^2;

Jtot = Jtot + 2*JU;

% calculating inertia tensor for glass about x,y,z
IxxG = (1/12)*mG*(9^2+34.5^2);
IyyG = (1/12)*mG*(9^2+0.25^2);
IzzG = \((1/12)*mG*(34.5^2+0.25^2)\);
IG = [IxxG 0 0; 0 IyyG 0; 0 0 IzzG];
IG_rot = R*IG*R^T;
JG = IG_rot(3,3)+mG*3.125^2;
Jtot = Jtot+2*JG;

% calculating mass moment of inertia of shaft
JS = 0.5*mSh*((1.25/2)^2+(1.5/2)^2);

Jtot = Jtot+2*JS;
% if only doing half of model
Jtot = Jtot/2;

timelength = 50;
tspan = 0:(timelength/n):timelength;
init_cond = [0 0]; % [39.35 57.2822]
options = odeset('reltol',1e-6,'abstol',1e-6);
[t,theta] = ode45(@(t,s)
TestingMotion(t,s,ct,Jtot,mE,pd,pa),tspan,init_cond,options);

for i = 1:length(theta)
    egoal(i) = 270;
end

for i = 1:length(theta)
    Fg123(i) = -mE*g;
    %Fgrot(i,1) = -mE*g*sind(45)*sind(pa)*cosd(theta(i))-
    mE*g*sind(45)*cosd(pa)*sind(theta(i));
    %Fgrot(i,2) = mE*g*sind(45)*sind(pa)*sind(theta(i))-
    mE*g*sind(45)*cosd(pa)*cosd(theta(i));
    Fgrot(i,1) = -mE*g*sind(45)*sind(pa)*cosd(theta(i))
+mE*g*sind(45)*cosd(pa)*sind(theta(i));
    Fgrot(i,2) = -mE*g*sind(45)*sind(pa)*sind(theta(i))-
-mE*g*sind(45)*cosd(pa)*cosd(theta(i));
Fgrot(i,3) = -mE*g*cosd(45);
magFg123(i) = mE*g;
    magFgrot(i) = sqrt(Fgrot(i,1)^2 + Fgrot(i,2)^2 + Fgrot(i,3)^2);
end

subplot(2,1,1)
plot(t,theta(:,1),'k',t,(pa+theta(:,1)),'r',t,egoal,'--')
legend('Box Angular Position','Eccentricity Angular Position')

subplot(2,1,2)
plot(t,magFg123,'k',t,magFgrot,'k',...)
    t,abs(Fgrot(:,1)),'c',t,abs(Fgrot(:,2)),'b',t,abs(Fgrot(:,3)),'r')
legend('Force magnitude','Transformed force magnitude','Transformed force, comp. 1','Transformed force, comp. 2','Transformed force, comp. 3')
axis([0 50 -2000 2200])
xlabel('Time [s]')
ylabel('Force magnitude [lb]')

%% functions

function result = TestingMotion(t,s,ct,Jtot,mE,pd,pa)
    result = zeros(2,1);
    g = 32.2*12;
    result(1) = s(2);
    result(2) = (-ct*s(2)-mE*g*pd*sind(45)*cosd(pa+s(1)))/
    (Jtot+mE*pd^2);
end
Simulation MATLAB script:

% Maura O'Neill

% MATLAB code to simulate the motion of the frame and the eccentric mass
% Parameters of unbalance can be modified accordingly

%% plotting data from Excel file

% clear command window, all variables, all previous plots
clc
clear all
close all

% read measurement data from selected Excel file
T = xlsread('TESTING3_8MAY17.xlsx');

% eliminate measurements of stationary frame before rotation
% by selecting beginning data point
beginData = 880;

% create vector of data collection times
time = T(:,1);
timeLength = length(time); % length of time vector
time = time(beginData:timeLength); % use beginData to select first
% data point of interest
startTime = time(1);
timeDiff = time(2)-startTime;

% loop to determine time vector with initial time t = 0
for i = 1:1:length(time)

142
timePos(i) = time(i)-startTime;
end

% correcting point that goes backward in time
for i = 2:1:length(timePos)
    if timePos(i)<timePos(i-1)
        timePos(i) = timePos(i-1)+timeDiff;
    end
end

roll = T(:,4);
roll = roll(beginData:length(roll));

for i = 1:1:length(roll)
    if roll(i)<0
        roll(i) = -roll(i);
    end
end

figure
subplot(2,2,1)
plot(timePos,roll,'k')

% simulation

n = 1000; % number of drawings
ct = 300; % damping constant
g = 32.2*12; % gravity in [in/s^2]
mE = 3; % mass of eccentricity
pd = 15; % normal distance of eccentricity from rotational axis
pa = 210; % initial angle of eccentricity
Jtot = 0; % total mass moment of inertial including U-channels, glass, etc.

z = 2;

% mass information for U-channel and glass using expected dimensions and measured mass
mS = 0.099025464074912*(34.5*2.33*5.42); %[lb]

mL = 0.099025464074912*(34.5*2.5*6);

mG = 6.2536;

mSh = 0.4194;

% calculating inertia tensor for U-channels about x,y,z

IxxU = ((1/12)*mL*(34.5^2+2.5^2)+mL*0.4527^2)-((1/12)*mS*(34.5^2+2.33^2)+mS*0.5377^2);

IyyU = ((1/12)*mL*(2.5^2+6^2)+mL*0.4527^2)-((1/12)*mS*(2.33^2+5.42^2)+mS*0.5377^2);

IzzU = (1/12)*mL*(34.5^2+6^2)-(1/12)*mS*(34.5^2+5.42^2);

IU = [IxxU 0 0;0 IyyU 0;0 0 IzzU];

R = [1 0 0;0 cosd(45) sind(45);0 -sind(45) cosd(45)]; % rotation matrix of x,y,z rotated around x by 45 degrees

RT = R';

IU_rot = R*IU*RT; % calculating new inertia tensor for bottom U-channel

JU = IU_rot(3,3)+(mL-mS)*(2.7973*sind(45))^2;

Jtot = Jtot + 2*JU;

% calculating inertia tensor for glass about x,y,z

IxxG = (1/12)*mG*(9^2+34.5^2);

IyyG = (1/12)*mG*(9^2+0.25^2);
\[ I_{zzG} = \frac{1}{12}m_G(34.5^2 + 0.25^2); \]
\[ I_G = \begin{bmatrix} I_{xxG} & 0 & 0 \\ 0 & I_{yyG} & 0 \\ 0 & 0 & I_{zzG} \end{bmatrix}; \]
\[ I_{G\text{\_rot}} = R*I_G*R^T; \]
\[ J_G = I_{G\text{\_rot}}(3,3) + m_G*3.125^2; \]
\[ J_{\text{tot}} = J_{\text{tot}} + 2*J_G; \]

\% calculating mass moment of inertia of shaft
\[ J_S = 0.5*m_{Sh}((1.25/2)^2 + (1.5/2)^2); \]
\[ J_{\text{tot}} = J_{\text{tot}} + 2*J_S; \]
\% if only doing half of model
\[ J_{\text{tot}} = J_{\text{tot}}/2; \]

timelength = 50;
tspan = 0:(timelength/n):timelength;
init_cond = [0 0]; \% [39.35 57.2822]
options = odeset('reltol',1e-6,'abstol',1e-6);
[t,theta] = ode45(@(t,s) TestingMotion(t,s,ct,Jtot,mE,pd,pa),tspan,init_cond,options);

for i = 1:length(theta)
    egoal(i) = 270;
end

for i = 1:length(theta)
    Fgl23(i) = -mE*g;
    Fgrot(i,1) = -mE*g*sind(45)*sind(pa)*cosd(theta(i)) - mE*g*sind(45)*cosd(pa)*sind(theta(i));
    Fgrot(i,2) = mE*g*sind(45)*sind(pa)*sind(theta(i)) - mE*g*sind(45)*cosd(pa)*cosd(theta(i));
    Fgrot(i,3) = -mE*g*cosd(45);
    magFgl23(i) = mE*g;
    magFgrot(i) = sqrt(Fgrot(i,1)^2 + Fgrot(i,2)^2 + Fgrot(i,3)^2);
end

subplot(2,2,2)
plot(t,theta(:,1),t,(pa+theta(:,1)),t,egoal,'--')
legend('Box Angular Position','Eccentricity Angular Position')

subplot(2,2,3)
plot(t,theta(:,1),'k',t,magFg123,'b',t,magFgrot,'r')

%% confirming direction of gravitational force vector throughout motion
for i = 1:length(t)
    thetadd(i) = (-ct*theta(i,2)-mE*g*pd*sind(45)*cosd(pa+theta(i,1)))/(Jtot+mE*pd^2);
solvtorque(i) = thetadd(i)*(Jtot+mE*pd^2);
end

beta = [cosd(pa) -sind(pa) 0;sind(pa)*cosd(45) cosd(pa)*cosd(45)
-sind(45);sind(pa)*sind(45) cosd(pa)*sind(45) cosd(45)]
pivotpoint = [0 0 z];
pivotpoint = beta*pivotpoint';

for i = 1:length(t)
    rE(i,1) = pd*cosd(pa)*cosd(theta(i,1))-
pd*sind(pa)*sind(theta(i,1));
    rE(i,2) = cosd(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))-z*sind(45);
    rE(i,3) = sind(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))+z*cosd(45);
    Fg = [0 0 -mE*g*sind(45)];
predtorque(i,:) = cross(rE(i,:),Fg);
   ptorque(i) = predtorque(i,3);
end
%% beginning simulation

% beginning to plot
figure('units','normalized','outerposition',[0 0 1 1])
view(3);
axis([-30 30 -30 30 -30 30])
hold on
grid on
plot3([0 0],[0 18],[0 -18],'k','LineWidth',3);
plot3([0 0],[0 -18],[0 18],'k','LineWidth',3);
plot3([0 0],[-18 18],[-18 -18],'k','LineWidth',3);
plot3([0 0],[-18 18],[18 18],'k','LineWidth',3);
plot3([0 0],[-18 -18],[-18 18],'k','LineWidth',3);
plot3([0 0],[18 18],[-18 18],'k','LineWidth',3);
xlabel('x');
ylabel('y');
zlabel('z');
ax = gca;
c = ax.XDir;
ax.XDir = 'reverse'
ax.YDir = 'reverse'

% figure(2)
% plot(t,ptorque)

pause
for i = 1:length(t)
    if i>1
        delete(eccent)
        delete(backglass)
        delete(frontglass)
delete(toppiece)
delete(bottompiece)
%delete(eplane)
delete(predgravity)
delete(string)
delete(ptorquearrow)
delete(solvtorquearrow)
end

xE(i) = pd*cosd(pa)*cosd(theta(i,1))-
pd*sind(pa)*sind(theta(i,1));

yE(i) = cosd(45)*((pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1))))-z*sind(45);

zE(i) = sind(45)*((pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))+z*cosd(45));

eccent = scatter3(xE(i),yE(i),zE(i),'MarkerEdgeColor',[0 0
0],...
'MarkerFaceColor',[0.8 0.2 0.5],...
'LineWidth',1);

ptorqueend = 1+ptorque(i);

ptorquearrow = arrow3([25 0 -30],[25 ptorqueend -30],'r',
0.5,2)

solvtorqueend = 1+solvtorque(i);
solvtorquearrow = arrow3([20 0 -30],[20 solvtorqueend
-30],'',0.5,2)
%eccentdot = scatter3(xE(i),yE(i),zE(i),'.','k');
backcorner1 = [-3.25 17.25 4.5];
backcorner2 = [-3.25 -17.25 4.5];
backcorner3 = [-3.25 -17.25 -4.5];
backcorner4 = [-3.25 17.25 -4.5];
frontcorner1 = [3.25 17.25 4.5];
frontcorner2 = [3.25 -17.25 4.5];
frontcorner3 = [3.25 -17.25 -4.5];
frontcorner4 = [3.25 17.25 -4.5];

148
eplanesize = 20;
eplane1 = [pd*cosd(pa)*cosd(theta(i,1))-
pd*sind(pa)*sind(theta(i,1)) ...
    cosd(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))-z*sind(45) ...
    sind(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))+z*cosd(45)];
eplane2 = [pd*cosd(pa)*cosd(theta(i,1))-
pd*sind(pa)*sind(theta(i,1)) ...
    - (cosd(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))-z*sind(45)) ...
    sind(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))+z*cosd(45)];
eplane3 = [pd*cosd(pa)*cosd(theta(i,1))-
pd*sind(pa)*sind(theta(i,1)) ...
    - (sind(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))+z*cosd(45))];
eplane4 = [pd*cosd(pa)*cosd(theta(i,1))-
pd*sind(pa)*sind(theta(i,1)) ...
    cosd(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))-z*sind(45) ...
    -(sind(45)*(pd*cosd(pa)*sind(theta(i,1))
+pd*sind(pa)*cosd(theta(i,1)))+z*cosd(45))];

%     eplane = patch([eplane1(1) 0 -eplane3(1) 0],...
%         [eplane1(2) 18 eplane3(2) -18],...
%         [eplane1(3) -18 eplane3(3) 18],'r')
%     eplane = patch([eplane1(1)*cosd(theta(i,1)+pa)-
%                  sind(theta(i,1)+pa)*(eplane1(2)*cosd(45)+eplane1(3)*sind(45))],...
%         [eplane1(2)*cosd(45)+eplane1(3)*sind(45)]*sin(45)) ...
\[
\begin{align*}
&[(\text{eplane1}(1)\cosd(45)\cdot\text{sind}(\theta(i,1)+\text{pa}) +\text{eplane1}(2)\cdot(\cosd(45))^2\cdot(1+\cosd(\theta(i,1)+\text{pa})) +\text{eplane1}(3)\cdot(\cosd(45))^2\cdot(\cosd(\theta(i,1)+\text{pa})-1)) \ldots \\
&\quad (\text{eplane2}(1)\cosd(45)\cdot\text{sind}(\theta(i,1)+\text{pa}) +\text{eplane2}(2)\cdot(\cosd(45))^2\cdot(1+\cosd(\theta(i,1)+\text{pa})) +\text{eplane2}(3)\cdot(\cosd(45))^2\cdot(\cosd(\theta(i,1)+\text{pa})-1)) \ldots \\
&\quad (\text{eplane3}(1)\cosd(45)\cdot\text{sind}(\theta(i,1)+\text{pa}) +\text{eplane3}(2)\cdot(\cosd(45))^2\cdot(1+\cosd(\theta(i,1)+\text{pa})) +\text{eplane3}(3)\cdot(\cosd(45))^2\cdot(\cosd(\theta(i,1)+\text{pa})-1)) \ldots \\
&\quad (\text{eplane4}(1)\cosd(45)\cdot\text{sind}(\theta(i,1)+\text{pa}) +\text{eplane4}(2)\cdot(\cosd(45))^2\cdot(1+\cosd(\theta(i,1)+\text{pa})) +\text{eplane4}(3)\cdot(\cosd(45))^2\cdot(\cosd(\theta(i,1)+\text{pa})-1))\ldots ,
\end{align*}
\]

\[
\begin{align*}
&\quad [\text{backglass} = \text{patch}([(\text{backcorner1}(1)\cosd(45)\cdot\text{sind}(\theta(i,1)) - \\
&\quad (\text{backcorner1}(2)\cosd(45)+\text{backcorner1}(3)\text{sind}(45))) \ldots \\
&\quad (\text{backcorner2}(1)\cosd(\theta(i,1))-\text{sind}(\theta(i,1))\cdot(\cosd(45)+\text{backcorner2}(3)\text{sind}(45))) \ldots \\
&\quad (\text{backcorner3}(1)\cosd(\theta(i,1))-\text{sind}(\theta(i,1))\cdot(\cosd(45)+\text{backcorner3}(3)\text{sind}(45))) \ldots \\
&\quad (\text{backcorner4}(1)\cosd(\theta(i,1))-\text{sind}(\theta(i,1))\cdot(\cosd(45)+\text{backcorner4}(3)\text{sind}(45)))\ldots ,
\end{align*}
\]
(backcorner2(1)*cosd(45)*sind(theta(i,1))
+backcorner2(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+backcorner2(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

(backcorner3(1)*cosd(45)*sind(theta(i,1))
+backcorner3(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+backcorner3(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

(backcorner4(1)*cosd(45)*sind(theta(i,1))
+backcorner4(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+backcorner4(3)*(cosd(45))^2*(cosd(theta(i,1))-1)]],...

[(backcorner1(1)*sind(45)*sind(theta(i,1))
+backcorner1(2)*(cosd(45))^2*(cosd(theta(i,1))-1)+backcorner1(3)*(cosd(45))^2*(cosd(theta(i,1))+1)) ...

(backcorner2(1)*sind(45)*sind(theta(i,1))
+backcorner2(2)*(cosd(45))^2*(cosd(theta(i,1))-1)+backcorner2(3)*(cosd(45))^2*(cosd(theta(i,1))+1)) ...

(backcorner3(1)*sind(45)*sind(theta(i,1))
+backcorner3(2)*(cosd(45))^2*(cosd(theta(i,1))-1)+backcorner3(3)*(cosd(45))^2*(cosd(theta(i,1))+1)) ...

(backcorner4(1)*sind(45)*sind(theta(i,1))
+backcorner4(2)*(cosd(45))^2*(cosd(theta(i,1))-1)+backcorner4(3)*(cosd(45))^2*(cosd(theta(i,1))+1))],...

'g')

frontglass = patch([(frontcorner1(1)*cosd(theta(i,1))-
                       sind(theta(i,1))*(frontcorner1(2)*cosd(45)+frontcorner1(3)*sind(45))) ...
                       (frontcorner2(1)*cosd(theta(i,1))-
                       sind(theta(i,1))*(frontcorner2(2)*cosd(45)+frontcorner2(3)*sind(45))) ...
                       (frontcorner3(1)*cosd(theta(i,1))-
                       sind(theta(i,1))*(frontcorner3(2)*cosd(45)+frontcorner3(3)*sind(45))) ...
                       (frontcorner4(1)*cosd(theta(i,1))-
                       sind(theta(i,1))*(frontcorner4(2)*cosd(45)+frontcorner4(3)*sind(45)))],...

[(frontcorner1(1)*cosd(45)*sind(theta(i,1))
+frontcorner1(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+frontcorner1(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

(frontcorner2(1)*cosd(45)*sind(theta(i,1))
+frontcorner2(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+frontcorner2(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

151
(frontcorner3(1)*cosd(45)*sind(theta(i,1))
+frontcorner3(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+frontcorner3(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

(frontcorner4(1)*cosd(45)*sind(theta(i,1))
+frontcorner4(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+frontcorner4(3)*(cosd(45))^2*(cosd(theta(i,1))-1)),...

[(frontcorner1(1)*sind(45)*sind(theta(i,1))
+frontcorner1(2)*(cosd(45))^2*(cosd(theta(i,1))
-1)+frontcorner1(3)*(cosd(45))^2*(cosd(theta(i,1))+1)) ...

(frontcorner2(1)*sind(45)*sind(theta(i,1))
+frontcorner2(2)*(cosd(45))^2*(cosd(theta(i,1))
-1)+frontcorner2(3)*(cosd(45))^2*(cosd(theta(i,1))+1)) ...

(frontcorner3(1)*sind(45)*sind(theta(i,1))
+frontcorner3(2)*(cosd(45))^2*(cosd(theta(i,1))
-1)+frontcorner3(3)*(cosd(45))^2*(cosd(theta(i,1))+1)) ...

(frontcorner4(1)*sind(45)*sind(theta(i,1))
+frontcorner4(2)*(cosd(45))^2*(cosd(theta(i,1))
-1)+frontcorner4(3)*(cosd(45))^2*(cosd(theta(i,1))+1))],...

'g')

toppiece = patch([(frontcorner1(1)*cosd(theta(i,1))-
-sind(theta(i,1))*(frontcorner1(2)*cosd(45)+frontcorner1(3)*sind(45))) ...

(backcorner1(1)*cosd(theta(i,1))-sind(theta(i,1))*(backcorner1(2)*cosd(45)+backcorner1(3)*sind(45))) ...

(backcorner2(1)*cosd(theta(i,1))-sind(theta(i,1))*(backcorner2(2)*cosd(45)+backcorner2(3)*sind(45))) ...

(frontcorner2(1)*cosd(theta(i,1))-sind(theta(i,1))*(frontcorner2(2)*cosd(45)+frontcorner2(3)*sind(45))],...

[(frontcorner1(1)*cosd(45)*sind(theta(i,1))
+frontcorner1(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+frontcorner1(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

(backcorner1(1)*cosd(45)*sind(theta(i,1))
+backcorner1(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+backcorner1(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...

(backcorner2(1)*cosd(45)*sind(theta(i,1))
+backcorner2(2)*(cosd(45))^2*(1+cosd(theta(i,1))))
+backcorner2(3)*(cosd(45))^2*(cosd(theta(i,1))-1)) ...
\[
\begin{align*}
&(\text{frontcorner2}(1)\cos(45)\sin(\theta(i,1)) \\
&+\text{frontcorner2}(2)\cos(45)^2(1+\cos(\theta(i,1)))) \\
&+\text{frontcorner2}(3)\cos(45)^2(\cos(\theta(i,1))-1)],... \\
&\text{[(frontcorner1}(1)\sin(45)\sin(\theta(i,1)) \\
&+\text{frontcorner1}(2)\cos(45)^2(\cos(\theta(i,1))-1)+\text{frontcorner1}(3)\cos(45)^2(\cos(\theta(i,1))+1)]}... \\
&(\text{backcorner1}(1)\sin(45)\sin(\theta(i,1)) \\
&+\text{backcorner1}(2)\cos(45)^2(\cos(\theta(i,1))-1)+\text{backcorner1}(3)\cos(45)^2(\cos(\theta(i,1))+1)]}... \\
&(\text{backcorner2}(1)\sin(45)\sin(\theta(i,1)) \\
&+\text{backcorner2}(2)\cos(45)^2(\cos(\theta(i,1))-1)+\text{backcorner2}(3)\cos(45)^2(\cos(\theta(i,1))+1)]},... \\
&'g') \\
\text{bottompiece = patch}([\text{(frontcorner4}(1)\cos(\theta(i,1)) \\
&-\sin(\theta(i,1))*(\text{frontcorner4}(2)\cos(45)+\text{frontcorner4}(3)\sin(45)))]... \\
&\text{(backcorner4}(1)\cos(\theta(i,1)) -\sin(\theta(i,1))*(\text{backcorner4}(2)\cos(45)+\text{backcorner4}(3)\sin(45)))]... \\
&\text{(backcorner3}(1)\cos(\theta(i,1)) -\sin(\theta(i,1))*(\text{backcorner3}(2)\cos(45)+\text{backcorner3}(3)\sin(45)))]... \\
&\text{(frontcorner3}(1)\cos(\theta(i,1)) -\sin(\theta(i,1))*(\text{frontcorner3}(2)\cos(45)+\text{frontcorner3}(3)\sin(45)))]],... \\
&\text{[(frontcorner4}(1)\cos(45)\sin(\theta(i,1)) \\
&+\text{frontcorner4}(2)\cos(45)^2(1+\cos(\theta(i,1)))) \\
&+\text{frontcorner4}(3)\cos(45)^2(\cos(\theta(i,1))-1)]}... \\
&(\text{backcorner4}(1)\cos(45)\sin(\theta(i,1)) \\
&+\text{backcorner4}(2)\cos(45)^2(1+\cos(\theta(i,1)))) \\
&+\text{backcorner4}(3)\cos(45)^2(\cos(\theta(i,1))-1)]}... \\
&(\text{backcorner3}(1)\cos(45)\sin(\theta(i,1)) \\
&+\text{backcorner3}(2)\cos(45)^2(1+\cos(\theta(i,1)))) \\
&+\text{backcorner3}(3)\cos(45)^2(\cos(\theta(i,1))-1)]},... \\
&(\text{frontcorner3}(1)\cos(45)\sin(\theta(i,1)) \\
&+\text{frontcorner3}(2)\cos(45)^2(1+\cos(\theta(i,1)))) \\
&+\text{frontcorner3}(3)\cos(45)^2(\cos(\theta(i,1))-1)])},... \\
\end{align*}
\]
\[
\begin{align*}
& (\text{frontcorner4}(1)\cdot \sin(45)\cdot \sin(\theta(i,1)) \\
& + \text{frontcorner4}(2)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))-1) + \text{frontcorner4}(3)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))+1)) \ldots \\
& (\text{backcorner4}(1)\cdot \sin(45)\cdot \sin(\theta(i,1)) \\
& + \text{backcorner4}(2)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))-1) + \text{backcorner4}(3)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))+1)) \ldots \\
& (\text{backcorner3}(1)\cdot \sin(45)\cdot \sin(\theta(i,1)) \\
& + \text{backcorner3}(2)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))-1) + \text{backcorner3}(3)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))+1)) \ldots \\
& (\text{frontcorner3}(1)\cdot \sin(45)\cdot \sin(\theta(i,1)) \\
& + \text{frontcorner3}(2)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))-1) + \text{frontcorner3}(3)\cdot (\cos(45))^2\cdot (\cos(\theta(i,1))+1))],
\end{align*}
\]

\text{predgravity} = \text{arrow3}([\text{xE}(i) \; \text{yE}(i) \; \text{zE}(i)], [\text{xE}(i) \; \text{yE}(i) \; (\text{zE}(i)-10)], 'k', 0.5, 1) \% mE\cdot g\cdot \sin(45)\cdot 0.01

\text{string} = \text{arrow3}([\text{pivotpoint}(1) \; \text{pivotpoint}(2) \; \text{pivotpoint}(3)], [\text{xE}(i) \; \text{yE}(i) \; \text{zE}(i)], 'k', 0.5, 2)

\text{alpha}(\text{backglass}, 0.5)
\text{alpha}(\text{frontglass}, 0.5)
\text{alpha}(\text{toppiece}, 0.5)
\text{alpha}(\text{bottompiece}, 0.5)
\% \alpha(eplane, 0.5)
\text{pause}(0.01)
\text{end}

\text{transform} = [1 \; 0 \; 0; 0 \; \cos(45) \; \sin(45); 0 \; -\sin(45) \; \cos(45)];

\text{for} \ i = 1: \text{length(xE)}
\quad \text{coordE}(i,1) = \text{xE}(i);
\quad \text{coordE}(i,2) = \text{yE}(i);
\quad \text{coordE}(i,3) = \text{zE}(i);
\text{end}
for i = 1:length(xE)
    tobetrans = coordE(i,:);
    trans = transform*tobetrans';
    newcoordE(i,1) = trans(1);
    newcoordE(i,2) = trans(2);
    newcoordE(i,3) = trans(3);
    radius(i) = sqrt((newcoordE(i,1))^2+(newcoordE(i,2))^2);
end

figure
subplot(1,2,1)
plot(t,radius)

subplot(1,2,2)
plot(t,newcoordE(:,3))

%% functions

function result = TestingMotion(t,s,ct,Jtot,mE,pd,pa)
    result = zeros(2,1);
    g = 32.2*12;
    result(1) = s(2);
    result(2) = (-ct*s(2)-mE*g*pdsaind(45)*cosd(pa+s(1)))/(Jtot+mE*pds^2);
end
Appendix E: Sample Calculations
Sample calculation of U-channel 1’s mass moment of inertia about the rotational axis:

This sample calculation is for the upper U-channel in Fig. 68, U-channel 1.

Table 18 lists the variables relevant to this sample calculation.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{U1}$</td>
<td>[lb-in$^2$]</td>
<td>U-channel 1 principle inertia tensor</td>
</tr>
<tr>
<td>$I_{11}, I_{22}, I_{33}$</td>
<td>[lb-in$^2$]</td>
<td>Principle mass moments of inertia in {e$_1$,e$_2$,e$_3$}</td>
</tr>
<tr>
<td>$x_{COM}$</td>
<td>[in]</td>
<td>Cross section center of mass horizontal coordinate</td>
</tr>
<tr>
<td>$y_{COM}$</td>
<td>[in]</td>
<td>Cross section center of mass vertical coordinate</td>
</tr>
<tr>
<td>$M$</td>
<td>[lb]</td>
<td>6.00”x2.50”x34.5” block mass</td>
</tr>
<tr>
<td>$m$</td>
<td>[lb]</td>
<td>5.42”x2.33”x34.5” block mass</td>
</tr>
<tr>
<td>$I_{U1}'$</td>
<td>[lb-in$^2$]</td>
<td>U-channel 1 transformed inertia tensor</td>
</tr>
<tr>
<td>$J_{U1}$</td>
<td>[lb-in$^2$]</td>
<td>U-channel mass moment of inertia about rotational axis</td>
</tr>
</tbody>
</table>
The inertia tensor, $I_{U1}$, of U-channel 1 about the Cartesian coordinate system was first calculated. This inertia tensor is listed in Eq. 75.

$$I_{U1} = \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix}$$ (75)

In order to calculate this inertia tensor, the center of mass of the U-channel was determined. This center of mass calculation involved splitting the cross-section of the U-channel into the following dimensioned sections in Fig. 69.

![Fig. 69. Dimensioned U-channel cross section. Dimensions in [in].](image)

The center of mass of this cross section relative to the point (-3,2) was calculated in the following Eqs. 76 and 77.

$$x_{COM} = \frac{(6)(0.17)(3) + (2.33)(0.29)(0.29/2) + (2.33)(0.29)(5.71 + 0.29/2)}{(6)(0.17) + (2.33)(0.29)(2)} = 3$$ (76)
The center of mass along the length of the U-channel was located at the halfway point of the length. Finally, the center of mass of the U-channel in the \{e_1, e_2, e_3\} coordinate system was \((0, 17.25, 4.7973)\).

Calculating the matrix entries of the inertia tensor for the U-channel involved treating the U-channel as a 6.00”x2.50”x34.5” block, mass \(M = 51.25 \text{ lb}\), with a 5.42”x2.33”x34.5” block, mass \(m = 43.14 \text{ lb}\), removed from it. The centers of mass of these blocks relative to the point (-3,2) in the cross section were (3,1.25) and (3,1.455), respectively. Eqs. 78 through 80 list the calculations of the non-zero matrix entries in the U-channel’s inertia tensor using the parallel axis theorem.

\[
I_{11} = \left[ \frac{1}{12} M(34.5^2 + 2.5^2) + M(0.4527)^2 \right] - \left[ \frac{1}{12} m(34.5^2 + 2.33^2) + m(0.6577)^2 \right] = 802.59
\]

\[
I_{22} = \left[ \frac{1}{12} M(6^2 + 2.5^2) + M(0.4527)^2 \right] - \left[ \frac{1}{12} m(5.42^2 + 2.33^2) + m(0.6577)^2 \right] = 47.13
\]

\[
I_{33} = \frac{1}{12} M(34.5^2 + 6^2) - \frac{1}{12} m(34.5^2 + 5.42^2) = 851.70
\]

This inertia tensor was then transformed into the \{e_1', e_2', e_3'\} coordinate system as seen in Eq. 81.
The mass moment of inertia of interest for this study was that about the rotational axis. Again implementing the parallel axis theorem, the mass moment of inertia of the U-channel about the rotational axis was calculated with Eq. 82.

\[
I_{U1} = \begin{bmatrix}
1 & 0 & 0 \\
0 \cos 45^\circ & \sin 45^\circ & 0 \\
0 & -\sin 45^\circ \cos 45^\circ & 0
\end{bmatrix}
\begin{bmatrix}
802.59 & 0 & 0 \\
47.13 & 0 & 851.70 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 \cos 45^\circ & -\sin 45^\circ \\
0 \sin 45^\circ \cos 45^\circ & 0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
802.59 & 0 \\
0 & 449.41 & 402.28 \\
0 & 402.28 & 449.41
\end{bmatrix}
\]

\( (81) \)

This calculation was subsequently performed for all of the components rotating about the rotational axis.