MATHEMATICAL MODEL VALIDATION OF A CENTER OF GRAVITY MEASURING PLATFORM USING EXPERIMENTAL TESTS AND FEA

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mechanical Engineering

by

Michael Lashore

June 2015

COMMITTEE MEMBERSHIP

TITLE:	Mathematical Model Validation of a Center of Gravity Measuring Platform Using Experimental Tests and FEA
AUTHOR:	Michael Lashore
DATE SUBMITTED:	June 2015
COMMITTEE CHAIR:	Peter Schuster, Ph.D. Professor of Mechanical Engineering
COMMITTEE MEMBER:	John Ridgely, Ph.D. Professor of Mechanical Engineering
COMMITTEE MEMBER:	Colleen Kirk, Ph.D. Professor of Mathematics

ABSTRACT

Mathematical Model Validation of a Center of Gravity Measuring Platform Using Experimental Tests and FEA

Michael Lashore

This thesis sets out to derive an analytical model for a center of gravity (CG) measuring platform and examines its validity through experimental testing and Finite Element Modeling. The method uses a two-stage platform tilting process to first locate the planar CG coordinates and then find the third CG coordinate normal to the platform. An uncertainty model of the measuring platform was also developed, both CG and uncertainty models were implemented in the form of a MATLAB code. A load cell sizing task was also added to the code to assist the Integration Engineers at Jet Propulsion Laboratory in selecting load cells to design their own version of the CG Platform. The constructed CG Platform for this project used an array of six strain gauges, four C2A-06-062LT-120 Tee Rosettes and two C2A-06-031WW-120 Stacked Rosettes. They were bonded onto the legs of three truss shaped bipods. Results from the Platform Tilting Tests could not be used to validate the CG model as the measured CG and weight values found from the experimental tests contained a considerable amount of error. The errors in the Platform Tilting Tests are believed to stem from the initial errors observed during the bipod rod and strain gauge calibration tests. As an alternative, an FE model of the CG measuring platform was created as another means of validation. The math model of the CG measuring platform was successfully validated by showing that there was less than a 0.01% different between the bipod loads predicted from the MATLAB code and the FE model. Using the FEM generated loads as inputs into the CG code to calculate a CG matched the initial point mass or CG created in the FE model within a 0.01% difference. To validate the CG model even further, another test should be performed using

a CG Platform prototype instrumented with load cells to generate new experimental data and compare them with the results from the FE model.

Keywords: Strain Gauges, uncertainty model, center of gravity modeling, measuring CG

ACKNOWLEDGMENTS

I would first like to give thanks to God because with him all things are possible. Working on this thesis project has been a fulfilling learning experience in which I was blessed with the opportunity to work with two outstanding industry mentors. Sergio Valdez and Blair Frandeen, I would like to thank both of you for initially proposing this project and for always providing me with insightful guidance. I owe a special thanks to Dr. Schuster, I would not have been able to complete this thesis without your help. I appreciate how you helped me establish the good habit of documenting my work after every milestone. This is something I will always practice. Thank you to my committee members Dr. Kirk and Dr. Ridgely. I would also like to thank my family for supporting me from the beginning. Mom, Dad, Kemi, Shade and Roxanna, this accomplishment is as much yours as it is mine. Thank you to everyone in MEP, MESA and the entire Mechanical Engineering Department.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	=	Cross sectional area ……....	in^2
Θ_i	=	Bipod rod angle ……..	degs
W	=	Weight of payload ……..	lbs
ε_μ	=	Microstrain …….	-
α	=	Platform tilt angle ……..	degs
β	=	Platform rotation angle ……..	degs
γ	=	Bipod offset angle ……..	degs
κ	=	Arbitrary function with q dependent variables ……..	-
F_i	=	Load in bipod rod ……...	lbs
F_{BPi}	=	Loads at each bipod ……...	lbs
δ	=	Bipod out of plane error …….	degs
σ	=	Uncertainty ……	-
GF	=	Gauge factor ……..	-
G_{amp}	=	Amplifier gain ……	v/v
R_s	=	Shunt resistances ……	Ω
x_{meas}	=	Measured x-component of the CG …….	in
y_{meas}	=	Measured y-component of the CG …….	in
z_{meas}	=	Measured z-component of the CG …….	in
x_{act}	=	Actual x-component of the CG ……..	in
y_{act}	=	Actual y-component of the CG ……..	in
z_{act}	=	Actual z-component of the CG ……..	in
ξ	=	Stain gauge misalignment ……..	degs
z_{shelf}	=	Shelf height relative to the adaptor plate……	in

1. INTRODUCTION

In the aerospace field, it is critical that the center of mass and gravity (COM and CG) of a particular structure be well understood. Having knowledge of both of these properties allow engineers to calculate trajectory paths for satellites and other spacecraft. The COM is the average position of a body's mass. CG refers to a specific point on the body were the mean gravitational forces are acting. For objects near the surface of the earth or at altitudes that are much smaller than the radius of the earth, the CG and the COM coincide with each other. This assumption will be maintained throughout the rest of this paper and the term CG will be used to describe this location.

A small deviation of the calculated CG from their actual locations can lead to systematic errors that can result in a spacecraft missing its estimated landing target or worse, not making it into orbit. In rockets, the line of action of the thrust must be directed toward the CG to avoid induced moments. In rotary devices like turbines, the CG of the turbine blade/rod assembly must be coincident with the axis of rotation to minimize imbalances that can create unwanted loading or vibrations.

$$\boldsymbol{r}_{CG} = \frac{\int \boldsymbol{r} * dm}{\int dm} \approx \frac{\sum_{i=1}^{N} \boldsymbol{r}_i * m_i}{\sum_{i=1}^{N} m_i} \tag{1}$$

For simple objects and shapes, the analytical solution for the CG is relatively easy to ascertain. This can be done using the equation above [1]. Common Computer Aided Design (CAD) software packages such as SolidWorks and PTC Creo Parametric (formally known as Pro/ENGINEER) can be used to approximate the CG for a CAD assembly model with complex geometries. Any variations in the CAD model from the actual model can result in finite differences between the CG estimated using CAD and the actual CG. If the CG is critical to the

design or application, it is often considered good engineering practice to validate the CG predicted from CAD software by some other means.

1.1 Objective

Jet Propulsion Laboratory (JPL) has requested a validated analytical model of a CG measuring platform which should be implemented in the form of a MATLAB code. A scaled down prototype of the CG measuring platform should be designed and built to test the validity of the CG math model. A model of the uncertainty for the CG is also needed to demonstrate how the range of the predicted CG is affected by measurement errors. The written code will help to corroborate the selected approach for determining CG's. Additionally, the code must have the ability to compare expected LCP loads from known payload CG's and platform orientations to estimate the percent capacities of the load cells during testing. This portion of the code would aid in the load cell selection process. The user should have the ability to input capacities and error specifications provided by load cell manufactures to assure that the loads being measured by each load cell are operating within their capabilities. This thesis project acts as a proof of concept for JPL to test the validity of the CG model by comparing experimental results gather from load cell platform (LCP) tests to the actual CG measurements.

The proposed code would be used to develop a larger LCP that would serve as an adaptor base for spacecrafts during the assembly phase at JPL. With the code in place, integration engineers will be able to make quick sanity checks, with confidence, on the CG location of the spacecraft as new components are added or subtracted. A system like this would aid in detecting local assembly errors such as large components installed in incorrect locations. It may even help to find systemic design flaws before the spacecraft is approved to move on to the next phase. The following sections in this chapter discusses existing methods for locating CG's as well as the method chosen in this project to measure CG.

1.2 Literature Review

Before a CG model can be derived or a prototype built, research on the existing methods for measuring CG must first be investigated. Doing so provides a better understanding of the work that has already be done in this area of study. The following sections highlight a few selected methods for measuring CG and review how they work.

1.2.1 Plumb Blob

The Plumb Line Method is a quick and crude method for locating the CG of rigid objects experimentally. It uses the idea that the gravity vector is always pointing downwards, towards the center of the earth. All that is required is a small plumb blob, a few pieces of string and an anchoring system for suspending the object. To locate the CG, the object must first be suspended from a single point on the object. Next, the plumb blob should be attached to a piece of string and hung at the anchoring point of the suspended object. The string should then be tapped to the object in its current hanging location and the plumb blob detached at the opposite end of the anchoring point. This process can be repeated for different orientations of the suspended object until a clear intersection point of the strings is visible. The intersection point for the strings is the effective CG of the object. This relatively crude method works well for locating two planar CG coordinates but its elegance suffers when trying to locate the last CG component relative to the third principal axis. More work would be required to establish a 3-dimensional grid to project the intersection point of the planar CG location onto the third principal axis. The figure below shows a sequence of how this can be done to find a 2-dimensional CG location.

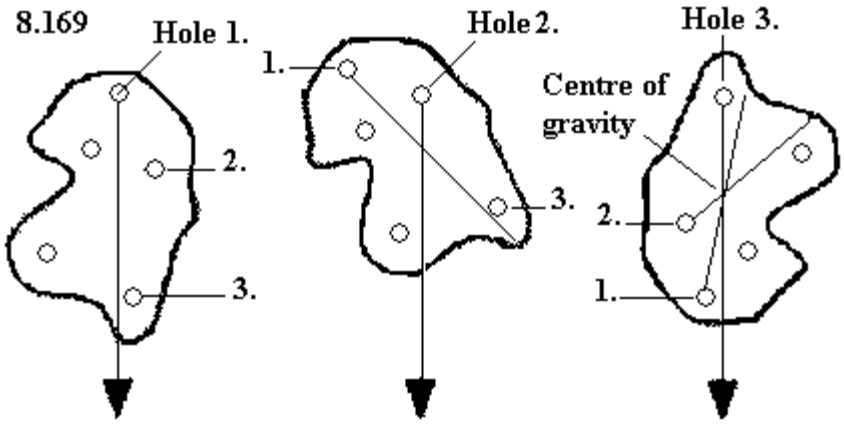

Figure 1: Plumb Blob method for locating the planar CG [2].

1.2.2 Tilting Method

Another commonly used method utilizes equilibrium concepts to determine CG. If an object is placed on the ground, one can use the reactional forces to calculate the CG using Newton's second law. The first step in using this method is finding the reaction forces. This can be done by using force transducers or load cells at the base of the object to measure the reaction loads at strategic points.

Figure 2: GM's method tilting method for measuring CG height [3].

A sponsored study by the Motor Vehicle Manufacturers Association sought out to assess the current practices for measuring the CG height of light truck vehicles [3]. Chrysler, Ford, General Motors (GM) and the National Highway Traffic Safety Administrations (NHTSA) all participated

in the study by conducting their own CG height tests on the same vehicles. Both Chrysler and GM utilized a tilting method to measure the CG height while Ford used a swing method. The NHTSA facilities had a special system which enabled them to measure the three principal inertia tensor as well as the CG height. More information on this design and its capabilities can be found in [3]. The study concluded that the results from the measured CG heights produced by GM's tilting tests were the closest to the true CG heights of the vehicles.

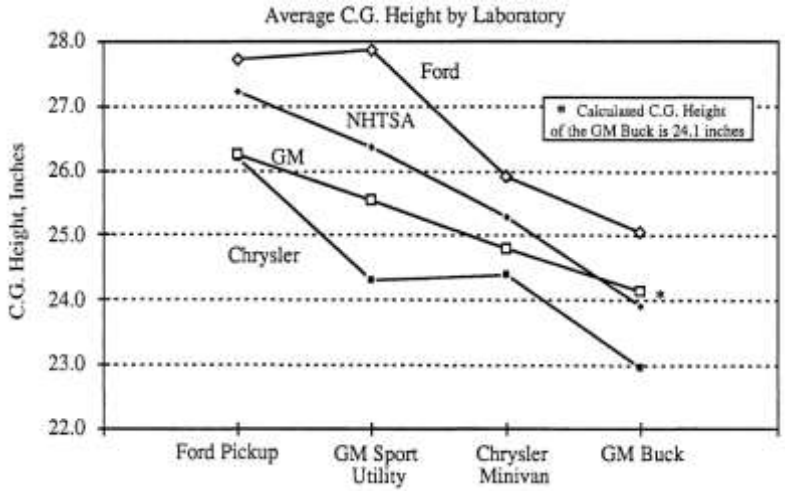

Figure 3: Average measured CG height comparison [3].

The report highlights the fact that measuring the CG height is not a simple matter. As it can be view in Figure 3, the different techniques used by the laboratories resulted in different CG height predictions. One key concept that can be taken way from this report was GM's care to keep the measured car as rigidity as possible. The paper noted that the other laboratories did not do that quite as well which could be one of the underlining causes for the difference between the CG results.

A technical paper written by Richard Boynton from Space Electronics [4] discusses a new method (Model: WCGT1000) for measuring the CG of a payload which minimizes the error seen in the CG results. Specifically, the error in the CG prediction is reduced by using a pivot point at

the center of the base while the load cells are place on the outer edge of one side of the base. Rather than using standard strain gauge based load cells, which rely on deflection, the WCGT1000 uses solenoid based load cells that prevent the load cells from deflecting. A closed loop system controlling the load cells applies current to restore the load cells to its unloaded position. This process is referred to as force restoration.

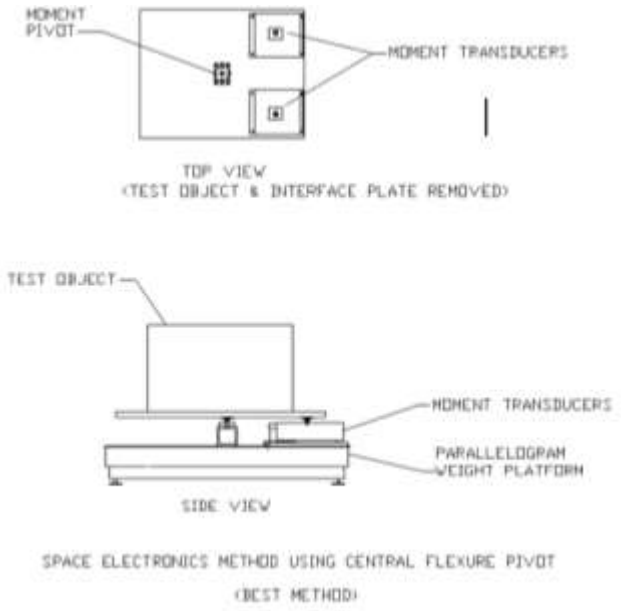

Figure 4: Pivot design for the WCGT1000.

In traditional CG platforms, CG prediction abilities are limited by the dynamic range of the strain based load cells. This range tends to have a large effect on the error in the CG prediction. When using standard load cells in this applications, each load cell must be capable of handling the entire weight of the payload. However, the required range of the load cell can be reduced by increasing the distance between the load cells and the center of the base. This lamentably decreases the accuracy of the CG outputs. The force restoration transducers tend to have much larger dynamic ranges with an excitation voltage of about 20 V. Standard strain gauge load cells have excitation voltages around 5V with poor noise to signal ratios in comparison.

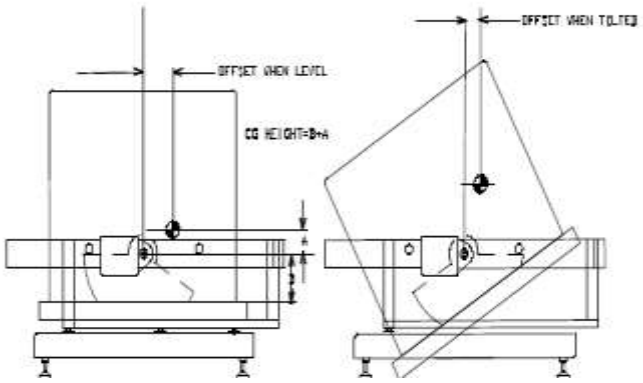

Figure 5: Tilted WCGT1000.

It was found that this system can reduce the total CG error by a factor of 30 [4]. The difference in this method is its geometric approach to calculating the vertical CG. For taller and softer payload, the CG also moves due to deflection of the payload. The force restoration method coupled with the pivot point work to compensate for CG movement due to the deflection of the payload body.

1.2.3 Uncertainty Methods

It is often considered good engineering practice to accompany a measured nominal value with its range of uncertainty. In actuality, the "true value" of a measurement is merely just a concept or a range in which the true value is believed to reside within. The width of the uncertainty range establishes what type of precision is needed for the measurement. The form in which the uncertainty is displayed usually depends on how the data was taken. Uncertainty can be divided into two groups. The first group, or group A, considers the uncertainty from data sampling. This group can be thought of as the uncertainty due to repeatability or random error. The second category, or groups B, are uncertainty quantifies based on measurement error or errors that can be recalled based on sound engineering knowledge. Group B can be referred as

Heuristic estimates. Operator bias, display resolution, computation and environmental factors are different forms of group B uncertainties [5]. More detail on how these types of uncertainties are calculated can be found in section 2.2.

1.3 Selected Method: Platform Tilting

The original design proposed by JPL for the CG measuring platform consisted of three bipods equally spaced apart, sandwiched between a payload adapter and base plate. Each bipod is comprised of two inline load cells that form a truss configuration. This particular design is refer to as the Load Cell Platform (LCP). The truss configuration was selected so that applied moments on the load cells could be avoided. Furthermore, each load cell would be anchored to the bipods using two ball endrods, one endrod at each end of the load cell. The ball endrods maintain a rigid connection between the load cells while the ball and socket portion of the endrods are connected to the bipods using shoulder screws. The three degree of freedom nature of the ball endrods aid in the assembly of the bipods and help widen the tolerance range during manufacturing. Once the platform is fully assembled the adapter plate cannot move without triggering the inline load cells. Figure 6 and Figure 7 show a CAD rendering of the proposed LCP and a single bipod, respectively.

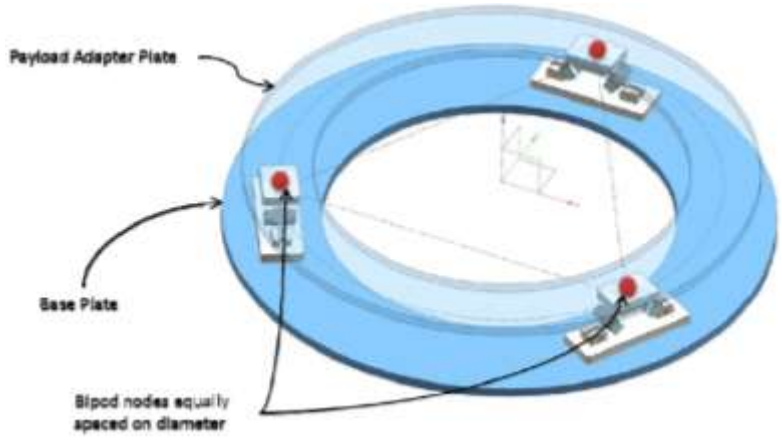

Figure 6: Initial CAD rendering of the LCP.

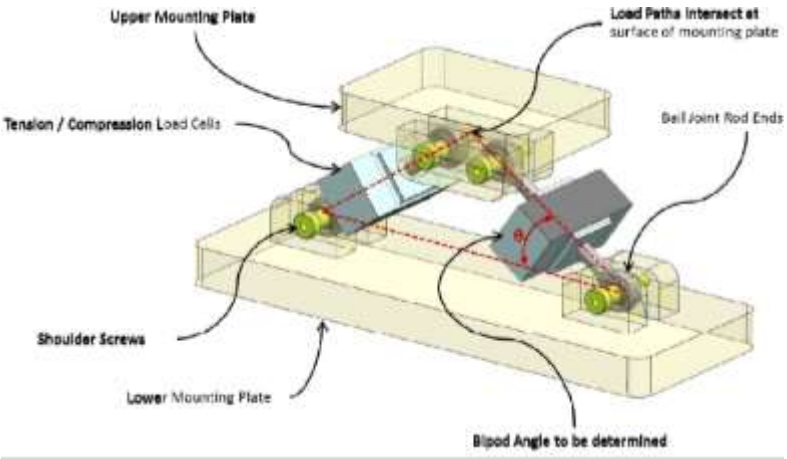

Figure 7: Proposed CAD model of a single bipod.

To effectively measure CG, the loads in the bipods can be recorded while the platform remains parallel to the horizontal to determine the planar CG coordinates. The platform can then be tilted to find the third CG coordinate along the vertical principal axis. Doing this would require a structure to rotate the platform and its payload around one axis. In both cases, statics can be

used to convert the bipod loads to a CG value. The CAD rendering below demonstrates this concept used on a rover.

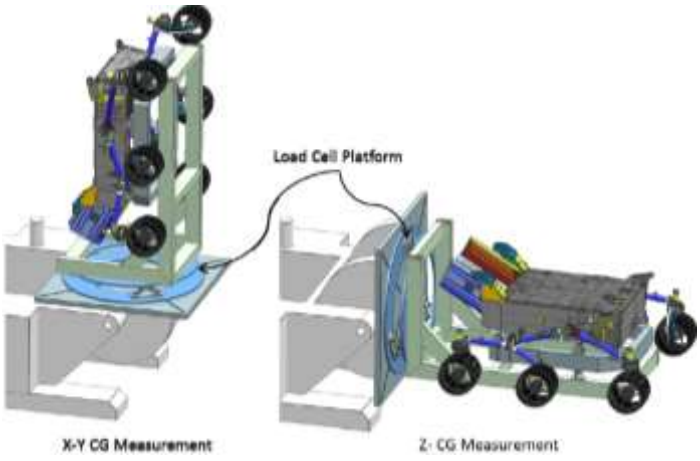

Figure 8: CAD concept rendering of a rover resting on a LCP in two different orientations [6].

The platform that was design for this project consisted of three bipods that were all 10.941 in away from the center of the platform and spaced 120^{o} away from each other. A solid hexagonal Base and Adaptor Plate was chosen instead of the proposed circular plate with a hollowed center. This decision was made to minimize the weight of the plates as much as possible without sacrificing its rigidity.

Rather than using load cells in the bipod legs to measure the loads in the platform, strain gauges can be used in their place to reduce the total cost of the prototype. The designed prototype utilizes strain gauges instrumented onto the bipod legs to measure the applied strain in the platform. The strain values can then be converted to loads to calculate the CG of a payload. A discussion of the uses of strain gauges and there selection can be reviewed in Chapter 4.

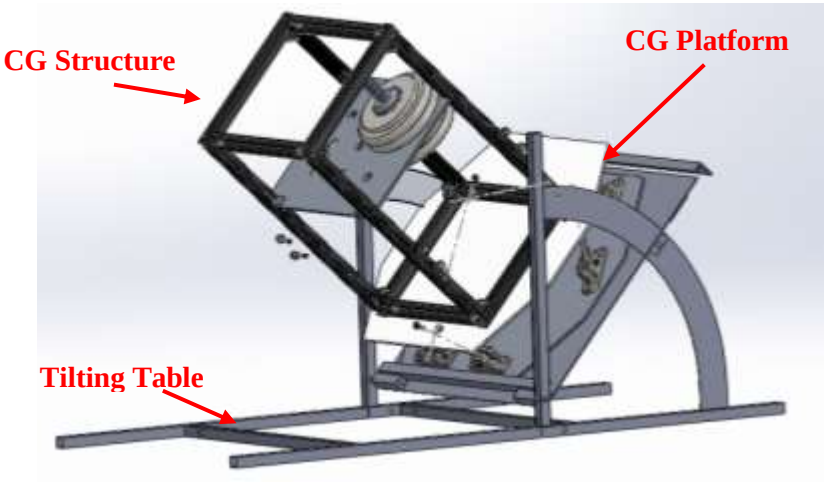

Figure 9: CAD model of CG Platform and CG Testing Structure being tilted by the Tilting Table.

A CG Testing Structure was designed to hold a payload in a fixed position while the platform is tilted at different angles. The main objective for this structure was to have the ability to accommodate various payload positions in order to mimic different CG locations. A shelf with an array of holes was used to change the position of the payloads to a finite number of discrete locations. Similarly, holes in the vertical members of the structure allow the shelf to be adjusted in the vertical direction. This was done by first constraining the vertical payload height to be no larger than 24 in and confining the payload to stay within a 15x14.5 in^2 area. The area constraint was derived from the fact that the payload CG must stay between the bipods to obtain viable load measurements to calculate CG [4]. The 15x14.5 in^2 area was the largest rectangular area that could fit within the bipod area constraint. Figure 9 shows the CAD model of the designed CG Platform with the CG Structure being tilted by Tilting Table.

2. CENTER OF GRAVITY MATHEMATICAL MODEL

This chapter relates the forces in the bipod rods to the CG of a payload resting on top the CG Platform. The process for determining the CG location and payload magnitude given the loads in the bipod rods and the orientation of the CG Platform will be referred to as the Forward Method. The process for calculating the loads in the bipod rods from a known CG location, payload magnitude and CG Platform orientation will be referred to as the Reverse direction. Please refer to the CAD model images as a visual aid in the derivations for both methods.

A model of the uncertainty in the Forward Method will be developed in section 2.3. Both models including the uncertainty in the Forward Method were then translated into a MATLAB algorithm as it will be described in Chapter 3. First, a body fixed coordinate system (BFCS or BCS for short) was chosen along with a naming convention for the bipods and bipod rods/forces. Figure 10 - Figure 12 show this respectively.

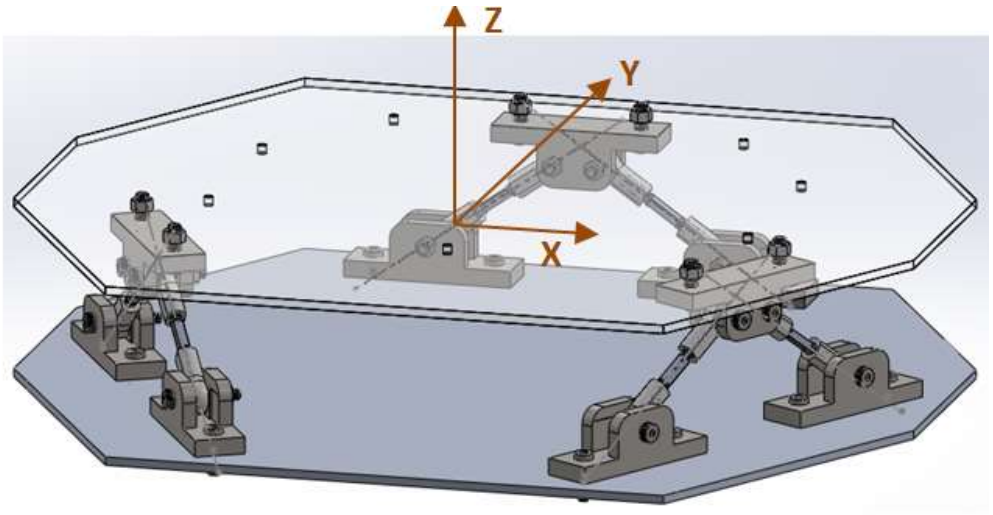

Figure 10: CAD drawing in an isometric view with the BCS at the center of the platform. The top plate is made transparent in this image.

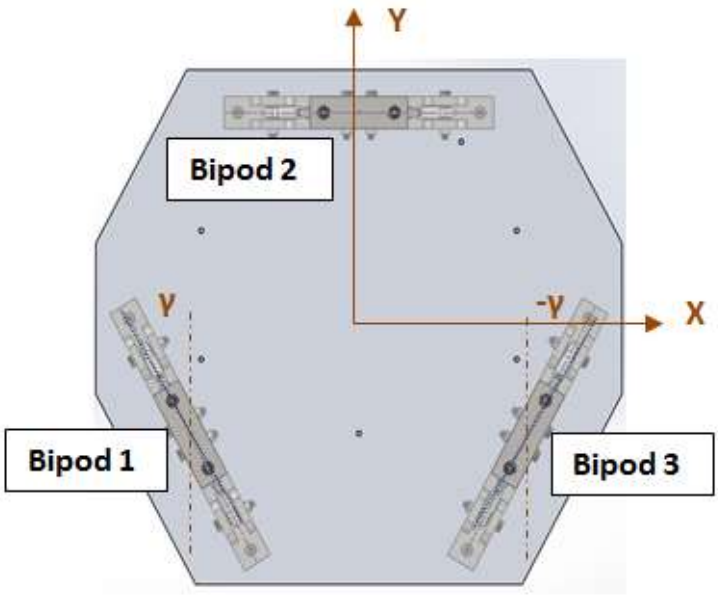

Figure 11: Top view of the CG Platform illustrating the Bipod naming convention.

Both bipods 1 and 3 have offset angles measured from an axis that is parallel to the y-axis and runs through their respective centers. The offset angle (γ) for bipod 1 (Bp 1) is positive since its rotation vector is in the positive z-direction. For Bp 3, γ is negative since the rotation vector points in the negative z-direction. By taking a sectional cut of the CG Platform and only looking at the top half of the platform, the naming convention for the bipod loads (F_i) can be defined. Figure 12 shows the sectional free body diagram with the weight of the platform neglected; tension loads were define as positive in this model. It is important to note that θ_i is the angle of elevation between each bipod rod and the Adaptor Plate. For these equations, the platform's bipods and bipod rods were all assumed to be one rigid body. The $F_{i,BPCS}$ terms below are the force vectors with no offset or when γ is zero, in the bipod coordinate system (BPCS).

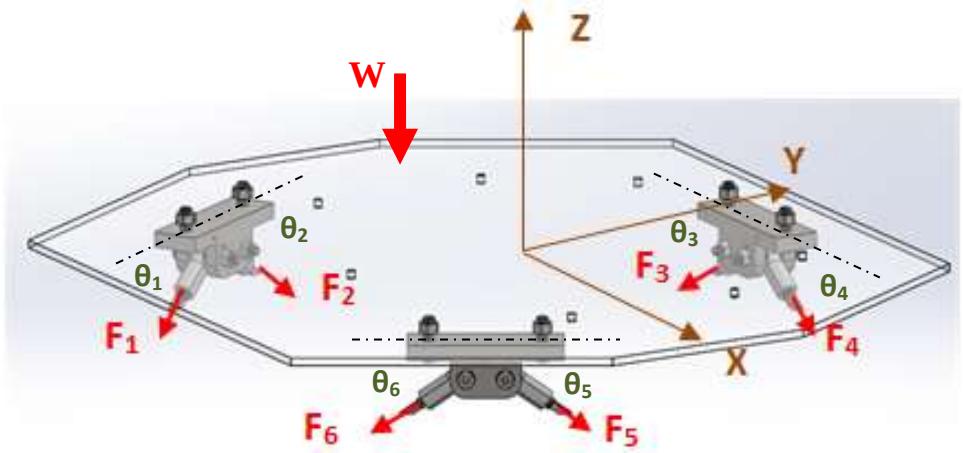

Figure 12: FBD of the top section of the CG Platform with the payload force omitted.

$$\boldsymbol{F_{1,BPCS}} = |F_1| \begin{bmatrix} 0 \\ -\cos(\theta_1) \\ -\sin(\theta_1) \end{bmatrix} \tag{2}$$

$$\boldsymbol{F_{2,BPCS}} = |F_2| \begin{bmatrix} 0 \\ \cos(\theta_2) \\ -\sin(\theta_2) \end{bmatrix} \tag{3}$$

$$\boldsymbol{F_3} = |F_3| \begin{bmatrix} -\cos(\theta_3) \\ 0 \\ -\sin(\theta_3) \end{bmatrix} \tag{4}$$

$$\boldsymbol{F_4} = |F_4| \begin{bmatrix} \cos(\theta_4) \\ 0 \\ -\sin(\theta_4) \end{bmatrix} \tag{5}$$

$$\boldsymbol{F_{5,BPCS}} = |F_5| \begin{bmatrix} 0 \\ -\cos(\theta_5) \\ -\sin(\theta_5) \end{bmatrix} \tag{6}$$

$$\boldsymbol{F_{6,BPCS}} = |F_6| \begin{bmatrix} 0 \\ \cos(\theta_6) \\ -\sin(\theta_6) \end{bmatrix} \tag{7}$$

The forces in the BPCS can be transformed to the BCS using a rotation transformation matrix, R_z, which rotates any vector an angle of ϕ about the z-axis. These transformations are applied to the F_1, F_2, F_5 and F_6 to compensate for their bipod offsets where $\phi = \gamma$. F_3 and F_4 are parallel with the x-axis and do not need to be transformed. (note: the bolded terms are vector quantities while the un-bolded are scalars)

$$\mathrm{R_z}(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\boldsymbol{F_1} = \mathrm{R_z}(\gamma) * \boldsymbol{F_{2,BPCS}} \quad ; \quad \boldsymbol{F_2} = \mathrm{R_z}(\gamma) * \boldsymbol{F_{2,BPCS}}$$

$$\boldsymbol{F_1} = |F_1| \begin{bmatrix} -\cos(\theta_1)\sin(\gamma) \\ \cos(\theta_1)\cos(\gamma) \\ -\sin(\theta_1) \end{bmatrix} \quad ; \quad \boldsymbol{F_2} = |F_2| \begin{bmatrix} \cos(\theta_2)\sin(\gamma) \\ -\cos(\theta_2)\cos(\gamma) \\ -\sin(\theta_2) \end{bmatrix} \tag{9}$$

$$\boldsymbol{F_5} = \mathrm{R_z}(-\gamma) * \boldsymbol{F_{5,BPCS}} \quad ; \quad \boldsymbol{F_6} = \mathrm{R_z}(-\gamma) * \boldsymbol{F_{6,BPCS}}$$

$$\boldsymbol{F_5} = |F_5| \begin{bmatrix} \cos(\theta_5)\sin(\gamma) \\ \cos(\theta_5)\cos(\gamma) \\ -\sin(\theta_5) \end{bmatrix} \quad ; \quad \boldsymbol{F_6} = |F_6| \begin{bmatrix} -\cos(\theta_6)\sin(\gamma) \\ -\cos(\theta_6)\cos(\gamma) \\ -\sin(\theta_6) \end{bmatrix} \tag{10}$$

Next, the weight vector is examined to model how its values change as the table is tilted and then rotated to different orientations. As it was stated in the previous chapter, the CG Platform is assumed to only function on the surface of the earth. A new approach would be needed to consider how the gravity vector changes as the CG Platform and the payload move to altitudes close to the magnitude of the radius of the earth. With the initial assumption of the COM of the payload being coincident with its CG, the weight vector will always point downwards in any platform orientation. Equation 11 below defines the weight vector as the table is tilted about the

y-axis a positive angle of α. Figure 13 shows the payload's weight vector components above the tilted CG Platform in the BCS.

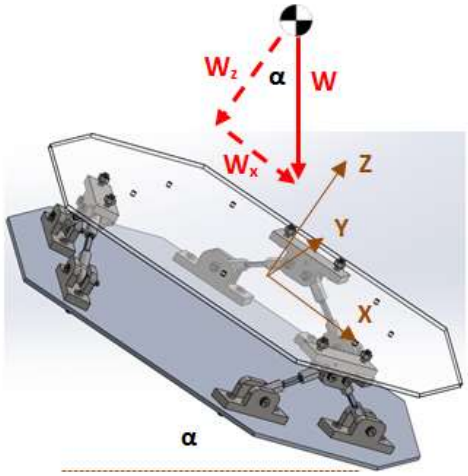

Figure 13: Isometric view of the tilted CG Platform rotated about the y-axis by an angle α.

$$\boldsymbol{W_{tilt}} = |W| \begin{bmatrix} \sin(\alpha) \\ 0 \\ -\cos(\alpha) \end{bmatrix} \tag{11}$$

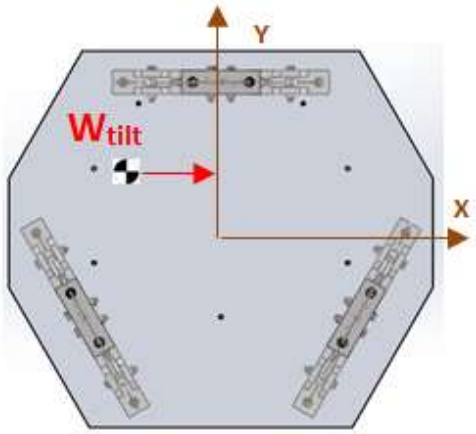

Figure 14: Top view of the platform about the y-axis in the BCS.

Any rotation of the platform about the z-axis of the BCS will be defined as β. A fixed local coordinate system (LCS) can placed at the CG of the payload where the z-axis of the LCS (z_{Local}) is normal to the platform and the y-axis of the LCS points in the same direction as the y-axis of the BCS. Any β rotation of the platform in the BCS is equivalent to a negative β rotation of the weight vector relative to the LCS. Figure 15 and Figure 16 shows a top view of the CG Platform with rotation applied to the platform in the BCS and rotation applied the weight vector in the LCS.

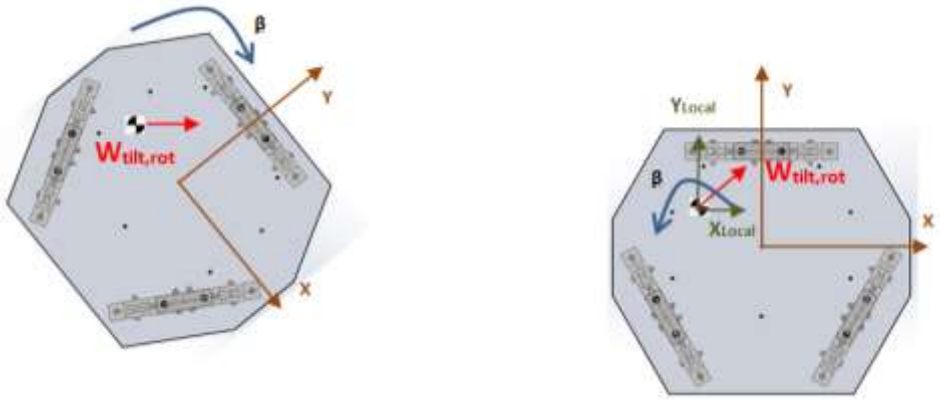

Figure 15: (Left) $0^o<\beta<90^o$, Rotation of the platform about z. (Right) $0^o<\beta<90^o$, Rotation of the weight vector about z_{Local}.

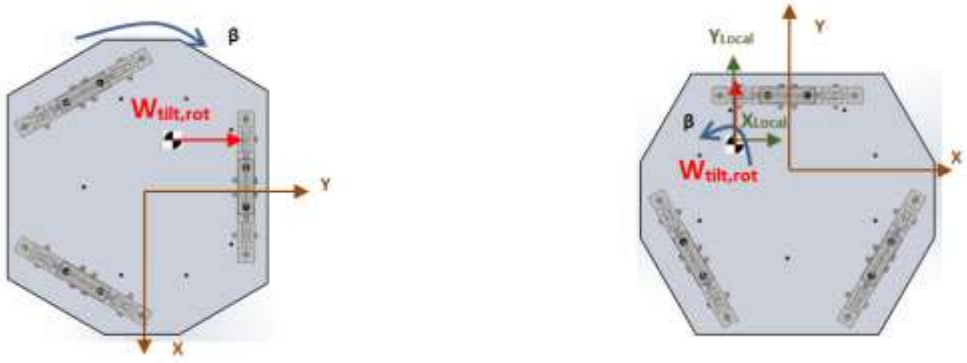

Figure 16: (Left) $\beta=90^o$, Rotation of the platform about z. (Right) 90^o Rotation of the weight vector about z_{Local}.

The same rotation transformation matrix can be applied to the payload to define its weight vector after the platform has experienced a rotation about z_{LOCAL}. To do this a negative β angle must be used to evaluate the transformation matrix. Applying this transformation matrix to the weight vector alone bypasses the need to transform all the bipod loads.

$$\boldsymbol{W_{tilt,rot}} = \mathrm{R_z}(-\beta) * \boldsymbol{W_{tilt}} \tag{12}$$

Before either the Forward or Reverse Method can be discussed, the intersection point of the load path of the forces in the three pairs of bipod rods must be defined in order to evaluate the moments created by the bipods. An expression for these intersection points relative to the center of the top of the Adaptor Plate or the origin of the BCS is defined as the following:

$$\boldsymbol{R_{b1}} = \left[x_{b1}, y_{b1}, z_{b1}\right] \qquad \boldsymbol{R_{b2}} = \left[x_{b2}, y_{b2}, z_{b2}\right] \qquad \boldsymbol{R_{b1}} = [x_{b3}, y_{b3}, z_{b3}] \tag{13}$$

In the following example, the intersection point relative to the origin of the BCS for Bp 1 or R_{b1}, will be considered. The same approach can be used for the other two bipod rod intersection points. R_2 is defined in the BCS as the distance from the origin to the center of rotation of the endrod holding bipod rod 2 in the lower bracket. Again, R_{b1} is the desired intersection point for Bp1. It is important to distinguish the difference between the two quantitates. Assuming that both the elevation angles of Bp rods 1 and 2 (θ_1 , θ_2) are known, as well as the distance between the two adjacent bottom bracket nodes (C_1), the Law of Sines can be used to find the length from the bottom bracket node 2 to the intersection point of two bipod rods. This term is referred to as $|\hat{r}_{rel}|_2$.

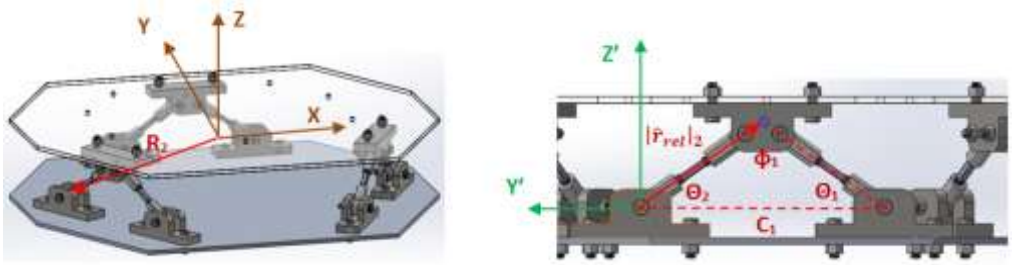

Figure 17: (Left) Defines R_2 in BCS . (Right) Shows rod angles, lower bracket nodes distance and $|\hat{r}_{rel}|_2$ in temporary coordinate frame.

A new temporary coordinate frame (TCF) is placed at bottom bracket node 2, at the tip of R_2. Noticing that the bipod rods form a triangle, the last interior angle (φ) of the bipod can be found by subtracting both θ_1 and θ_2 from 180°. $|\hat{r}_{rel}|_2$ can be found using one of the rod angles and φ using equation (14). This magnitude can be written as a vector in the TCF, $\hat{\boldsymbol{r}}_{\boldsymbol{rel_2}}$.

$$\frac{|\hat{r}_{rel}|_2}{\sin(\theta_1)} = \frac{|\hat{r}_{rel}|_1}{\sin(\theta_2)} = \frac{C_1}{\sin(\phi)} \tag{14}$$

$$\hat{\boldsymbol{r}}_{\boldsymbol{rel_2}} = C_1 * \frac{\sin(\theta_1)}{\sin(\phi)} * \begin{bmatrix} 0 \\ -\cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} \tag{15}$$

Care must be taken in selecting values for the bipod angles and the distance between the nodes of the bottom bracket. Some combinations can result in an intersection point that does not fall on the Adaptor Plate or an intersection point that does not exist. The latter scenario is possible if the bipod rods do not reside on the same plane. Out of plane error will be examined in more detail in the section 2.3.3. In this example, it is assumed that the bipod rods are coplanar.

Similar to the load vectors, $\hat{\boldsymbol{r}}_{\boldsymbol{rel_2}}$ can be written relative to the local coordinate system by multiplying the vector in the temporary coordinate frame by the z-rotation transformation matrix

R_z evaluated at the bipods offset angle γ. Now both $\hat{\boldsymbol{r}}_{\boldsymbol{rel_2}}$ and $\boldsymbol{R_2}$ can be summed together to find the intersection point of bipod 1 relative to the LCS.

$$\mathbf{r}_{\mathrm{rel},2} = \mathrm{R_z}(\gamma) * \hat{\boldsymbol{r}}_{\boldsymbol{rel,2}} \tag{16}$$

$$\boldsymbol{R_{b1}} = \boldsymbol{R_2} + \ \mathbf{r_{rel,2}} \tag{17}$$

2.1 Reverse Method

If the CG location and weight magnitude is known, a set of six linear equations can be derived using static equilibrium laws. The sum of the moments can be taken about the platform's origin where each bipod force is a summation of its two bipod rod forces and M_w is the moment induced by the payload's weight vector. Again summing the bipod loads at each individual bipod is only possible if the bipod rods are in the same plane. These moments can be used to generate three linear equations. The sum of all the bipod forces (F_{Bpi}) and the payloads weight vector can also be evaluated in an equilibrium state. The *i* in the equations below represents the bipod index. More moments and platform loads can be introduced to increase the overall sensitivity of the system by implementing more bipods, although, this was not investigated in this project.

$$\Sigma \boldsymbol{F} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{3}} \boldsymbol{F_{BPi}} + \ \boldsymbol{W_{tilt,rot}} = 0 \tag{18}$$

$$\Sigma \boldsymbol{M_o} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{3}} \boldsymbol{M_{bi}} + \ \boldsymbol{M_w} = 0 \tag{19}$$

The moments created by each bipod as well as the moment produced by the weight vector are defined as follows:

$$\boldsymbol{M_{bi}} = \boldsymbol{R_{bi}} \times \boldsymbol{F_{BPi}} \tag{20}$$

$$\boldsymbol{M_w} = [x_{cg}, y_{cg}, z_{cg}] \times \boldsymbol{W_{tilt,rot}} \tag{21}$$

A matrix filled with coefficients from the bipod loads and moments can be rearranged to form $[A]_{bp.}$ The load and moment components of the weight form a column vector $[B]_{load}$ while the unknown bipod forces can be factored to formed the column vector $\boldsymbol{X_{force}}$. These bipod forces can be found by using matrix inversion or Gaussian elimination techniques. In this case, the unknown bipod force column vector will be found using Gaussian elimination in MATLAB.

$$[A]_{bp}\boldsymbol{X_{force}} = [B]_{payload} \tag{22}$$

2.2 Forward Method

The previous procedure can be modified to find the CG coordinates when the bipod forces are known. In this case, the equilibrium matrix equation can be re-grouped in such a way that the matrix of coefficients, $[A]_{bp}$, becomes filled with the coefficients from the moment vectors produced by the payload weight, $[A]_{paylod}$. The solution vector, $\boldsymbol{X_{force}}$, then becomes a column vector of the CG coordinates (x_{cg}; y_{cg}; z_{cg}). All the known or measured forces are used to evaluate the moments of each bipod and are lumped in $[B]_{bp}$. Each row corresponds to a moment equilibrium equation along one of the three orthogonal axes, about the origin.

$$[A]_{payload}\boldsymbol{X_{cg}} = [B]_{bp} \tag{23}$$

$$mg * \begin{bmatrix} 0 & -cos(\alpha) & sin(\alpha) * sin(\beta) \\ cos(\alpha) & 0 & -\sin(\alpha) * \cos(\beta) \\ -sin(\theta_y) * sin(\beta) & \sin(\theta_y) * \cos(\beta) & 0 \end{bmatrix} \begin{bmatrix} x_{cg} \\ y_{cg} \\ z_{cg} \end{bmatrix} = \begin{bmatrix} \Sigma M_x \\ \Sigma M_y \\ \Sigma M_z \end{bmatrix} \tag{24}$$

The $[A]_{payload}$ matrix is singular and not positive definite. This characteristic causes conventional Gaussian or inversion methods to fail. One way to find the solution vector, $\boldsymbol{X_{cg}}$, and circumvent the singularity properties of $[A]_{payload}$ is to take a more systematic approach. The size of $[A]_{payload}$ and $\boldsymbol{X_{cg}}$ can be reduced by evaluating the moments about the origin when there is no tilt angle present in the platform ($\alpha = 0$). Doing this condenses $[A]_{payload}$ to only the upper left portion of the matrix and causes the off diagonal cosine terms to be 1 and -1. The two planar CG coordinates (x_{cg},y_{cg}) can be easily calculated when the platform is parallel to the horizontal. Once the two planar coordinates are found the platform can then be tilted to introduce moments about the third principal axis, with respect to the origin, so that z_{cg} can be evaluated.

An alternative method requires techniques for dealing with singular matrices. One way this can be done is to use the Least Squares Approximation to establish the normal equation and then use Singular Matrix Decomposition (SVD) to create a new invertible matrix. Details for this more robust method can be found in Appendix C. The code written for this project uses the two step systematic approach to calculate the full CG vector.

2.3 Uncertainty in Center of Gravity Algorithm

As it was mentioned in section 1.2.3, strict standards or an accepted range for which the true value should exist can be established. Four categories of error sources are considered in the development of the uncertainty model for the CG algorithm: measurement, repeatability, observed heuristic and actual errors. Each independent variable or constant parameter needed to calculate the CG vector contributes to the resulting *measurement* uncertainty. The measurement uncertainty was calculated using the uncertainty propagation technique. Along with the measurement uncertainty, the *repeatability* error was also examined by finding the standard deviation for a sample of results. *Observed Heuristic* uncertainty was formulated and focused on uncertainty discovered by observations. The combined uncertainty is then compared to the *actual*

error which is defined here as the difference between the measured values and the values that were expected. For example, the actual error in the payload weight would be the difference between the experimentally measured payload value using the CG Platform and the expected value of the payload using a digital scale. The uncertainty model was only developed for the Forward Method.

2.3.1 Measurement Uncertainty

One must first consider the resolution or margin of error of the measurement equipment to approximate how the measurement uncertainty builds after successive calculations. The resolution is defined as half of the precision capability of the measurement system. For example, if a load cell has the ability to measure loads to a precision of 0.001 lbs, then the resolution for this particular load cell is 0.0005 lbs. Four primary variable measurements were required in order to determine the values of the load and CG. These primary variables were angle, length, area and strain. Each variable's relative effect on the total uncertainty is captured in its sensitivity values. The sensitivity of a particular variable (S_{qi}), in a function, can be found by taking the partial derivative of the function (κ), with respect to the variable in question (q_i). Evaluating the result will produce a sensitivity value that can be compared with the sensitivities of the remaining variables.

$$S_{q_i} = \left|\frac{\partial \kappa}{\partial q_i}\right| \tag{25}$$

To compute the uncertainty (σ_k), the squared product of the sensitivity and the resolution error (ψ) for all the variables must be summed together to determine the covariance of κ. The square root of the covariance function can then be taken to find the uncertainty in κ due to the propagated errors in the variables [5]. This procedure will be used to find the measurement

uncertainty of the load values, the payload magnitude and the CG vector. This type of uncertainty is categorized as a B-Type uncertainty. However, an explicit expression for κ is needed in order to use the propagation of uncertainty method. An expression for each CG component is generated in the MATLAB code using the *CGsymbExpression* function so that the propagation method could be used. More information on how the function works will be discussed in Chapter 3. Equation (26) shows the expression that can be used to calculate the propagated uncertainty.

$$\sigma_F = \sqrt{\sum_{i=1}^{n} \left(S_{q_i}\right)^2 \psi_i^2} \tag{26}$$

2.3.2 Repeatability Uncertainty

It is often natural to assume that the repeatability error for a given population is normally distributed. Therefore, the uncertainty range due to repeatability error can be represented as the standard deviation away from the mean of a given sample [5]. The repeatability uncertainty becomes relevant during the testing of the CG Platform since multiple measurements are taken. The equation below shows the sample standard deviation; it is an A-type uncertainty estimate and it comes from the square root of the sample variance of the error distribution [5]. It is important to note that the sample variance of the error distribution is not the same as the variance from the measurement uncertainty. In this case, the sample variance refers to the sum of the squared difference between each sample and the sample mean divided by the degrees of freedom of the sample (*n-1*).

$$sd_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{27}$$

2.3.3 Observed Heuristic Uncertainty

The Observed Heuristic uncertainty refers to the inherent errors present in the CG model or the physical prototype which are discovered from observations. As it was mentioned earlier in this chapter, the steps demonstrated for locating the bipod load path intersection points, $\boldsymbol{R_{bi}}$, is only valid if the bipod rods in a single bipod are coplanar. Small offset angles between two adjacent bipod rods result in a violation of the assumption created to calculate $\boldsymbol{R_{bi}}$. To capture this heuristic uncertainty, the out of plane angle for a single set of bipod rods must be examined.

First, one must realize that all the bipods are 120^{o} away from each other and that the magnitude of $\boldsymbol{R_{bi}}$ in the x-y-plane of the BCS can be slightly larger or smaller depending on the direction of the offset angle (δ_i) of the bipod planes. Each δ is measured relative to the vertical plane that runs through the center of the bipod; an image of this can be view below. If $\delta_1 = \delta_2 = 0$ then the rods are coplanar and no additional heuristic uncertainty is needed. Although, if an offset angle does exist then only the worst case scenario needs to be considered. This occurs where the offset angles have the same magnitude. If a clearance between the endrods and brackets are small then a small angle approximation for δ is appropriate and bipod out of plane uncertainty can be modeled as the following:

$$\sigma_\delta = \pm\left|\hat{r}_{rel_2}\right| \sin(\delta) \approx \pm\left|\hat{r}_{rel_2}\right|\delta \tag{28}$$

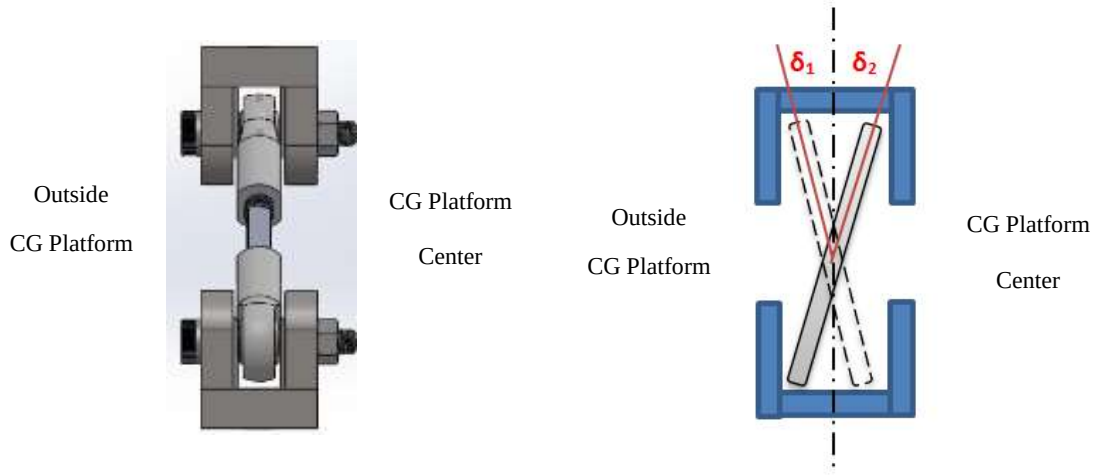

Figure 18: (Left) CAD Model right view of bipod 1. (Right) Exaggerated right view of bipod 1 showing δ for rod 1 and 2.

2.3.4 Combined Uncertainty

With all the individual sources of uncertainty defined, a combination of all the uncertainties can be found by taking the root-sum-squared (RMS) of the quantities. Any additional errors that can be identified can be accounted for in the total uncertainty by adding more terms in the RMS equation.

$$\sigma_{tot} = \sqrt{\sigma_{rep}^2 + \sigma_{meas}^2 + \sigma_H^2} \tag{29}$$

The implementation of the CG math model into a MATLAB code will be discussed in the next chapter. The methods used for coding the uncertainty analysis into MATLAB will also reviewed.

3. MODEL IMPLEMENTATION IN MATLAB

This section maps out the functionality and implementation of the CG model in MATLAB. The various inputs, outputs and coding concepts of the MATLAB code will be discussed in detail. The code has the ability to convert the bipod loads of six load cells into a CG location along with its uncertainty (Forward direction). This method also requires a platform tilt angle and the bipod loads at the corresponding angle to calculate all three CG components. The code can also work in the opposite direction, or in Reverse, to determine the values of the said load cells given the platforms orientation, CG location and weight of the payload. Since strain gauges were the transducer of choice in this project, extra functions were created to convert the measured strains into loads before they could be used as inputs to the main script file. The map/pseudo task diagram in Figure 19 should be reviewed before continuing.

Using the *MainCGscript* code, the user has the ability to switch between a CG measuring task and a load cell sizing task. Before running the main script file the user must first upload their inputs into the appropriate excel files. One file named **CG_platform_constants** contains the constants needed to run both the Forward and Reverse Method. In the second file, **ForceData**, the values of the bipod loads from each load cell are stored and used to generate outputs for the Forward Method.

The **CG_platform_constants** excel book contains two sheets in which the parameters for the platform and the resolution errors can be stored. The first sheet***, measured_const****,* is divided into two halves. On the left side are the various tables of parameters and on the right is a visual aid or legend for the parameter variables. A screen shot of the two halve can be seen in

Figure 20 and Figure 21. Both the Forward and Reverse Methods require that the values for the Bottom Bracket Parameters and Angles Input tables be filled out. The Angle Input table on

the left only needs to be filled out for the Forward method while either the left alone or both the left and right Angle tables can be filled for the Reverse Method. More information about the latter option will be reviewed in section 3.2.1. MATLAB calls on the Bottom Bracket Parameter cells to calculate the load path intersection point ($\boldsymbol{R_{bi}}$) for each bipod in the *bipodNodes* function. $\boldsymbol{R_{bi}}$ is calculated using the bottom bracket node locations with the bipod offset angle (γ) and rod angles (ϑ_i) rather than using the bottom and top bracket node locations to calculate $\boldsymbol{R_{bi}}$. This change can be done by reserving space for the top bracket node locations in **CG_platform_constants** and updating the $\boldsymbol{R_{bi}}$ calculation in the *bipodNodes* function. The added calculation in *bipodNods* would need to calculate ϑ_i from the top and bottom bracket nodes locations.

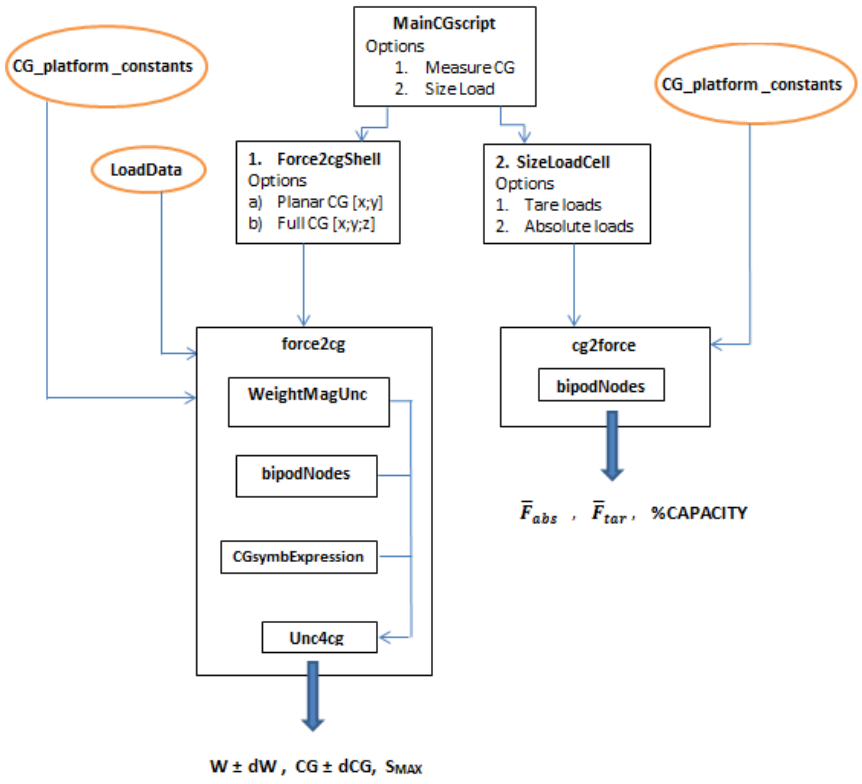

Figure 19: Map/pseudo-task diagram of CG code.

		Bottom Bracket Parameters					
		r1	r2	r3	r4	r5	r6
x	[in]	-6.2	-11.375	-4.375	4.5	11.438	6.25
y		-9.375	-1.75	10.125	10	-1.125	-9.3125
z		-3.5	-3.5	-3.5	-3.5	-3.5	-3.5
θ	[degs]	30	30	33	28	32	35
c	[in]	8.750		8.813		8.827	

Angle Inputs		
Rotation angle β	[degs]	0
Tilt angle α		30
γ		30

Angle Inputs After Tare		
Rotation angle β	[degs]	0
Tilt angle α		0
γ		30

Load Cell Sizing Inputs		
Load Cell Error	[%]	2
Capacity	[lbs]	300
Payload CG Location[x;y;z]	[in]	4.27
		-1.41
		12.72
payload magnitude	[lbs]	43.47

Load Cell After Tare		
Payload CG Location[x;y;z]	[in]	-4.5
		-1.14
		12.91
payload magnitude	[lbs]	43.42

Note:	Save file before running the MATLAB to accept any new changes

Figure 20: Table of parameters screen shot of the measured_const sheet in the CG_platform_constants excel workbook.

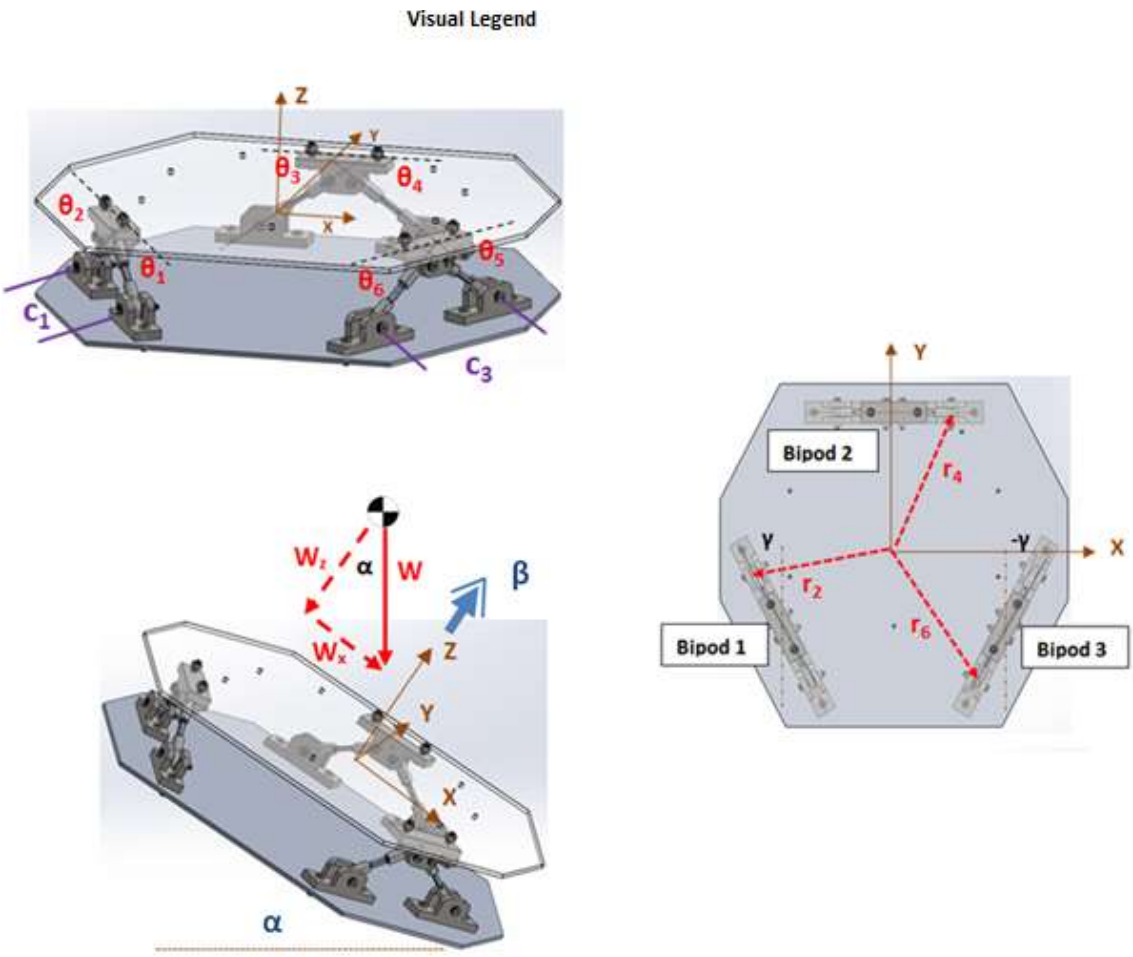

Figure 21: Visual legend screen shot of the measured_const sheet in the CG_platform_constants excel workbook.

The main code is comprised of two main functions, the *Force2cgShell* function that evaluates the Forward Method and the *SizeLoadCell* function that evaluates the Reverse Method. The following sections will be divided in a manner that discusses the Forward and Reverse Methods separately. Again, the pseudo-task diagram or map of the CG code in Figure 19 can be used as a reference.

3.1 Forward Direction

3.1.1 Inputs from Excel: CG Platform Constants & Load Data

In a second excel book, **ForceData**, the user is provided with two tables to enter the load values from six load cells. These values represent the load experienced by the bipod rods at some loading scenario. The first table is meant for loads (F_i) at $\alpha = 0^o$ while the second table is for loads (F'_i), at user defined α and β angles. Each table contains three columns; the first is for the bipod rod index and does not need to be changed. This column is not used in the MATLAB code, it is merely there to serve as a reference for the use. If the numbering is changed then the user must make the appropriate changes in the *Force2cgShell* function so that the loads being loaded into MATLAB match the numbering scheme in excel. Failure to do so may cause the code to output inaccurate results. The second and third columns are for the load values without a payload (F_o) and with a payload (F_{load}) respectively. These second column is needed to compensate for any preloads induced by the Adaptor Plate and CG structure to calculate the tare baseline for the load values. The table on the right serves as a legend for the user.

α =	0	
Load Cell Index	**F_o**	**F_{load}**
1	0	3.9840996
2	0	3.9840996
3	0	-26.16952
4	0	-26.16952
5	0	-15.70516
6	0	-15.70516

Legend	Description
F_o	Initial Load cell reading at α = 0
F_{load}	Load cell readings at α = 0
F'_o	Initial Load cell reading at α ≠ 0
F'_{load}	Load cell readings at α ≠ 0

α =	90	β =	60
Load Cell Index	**F'_o**	**F'_{load}**	
1	0	-36.00945	
2	0	-20.47501	
3	0	20.474563	
4	0	36.009002	
5	0	18.764228	
6	0	-18.76339	

Figure 22: LoadData file excel snapshot for the Forward Method.

The second sheet named ***uncertainty_const*** in **CG_uncertainty_constants** holds the measurement resolution uncertainties for all the inputs. The cells in this sheet are used to evaluate the uncertainty range and maximum sensitives for the CG and payload magnitude. Empty cells in this sheet could result in incorrect uncertainty values or create an error in MATLAB. The image in Figure 23 shows a snapshot of the uncertainty table in ***uncertainty_const***.

Variable	value	units
dθ	5	deg
dθy	5	
dθz	5	
dθzb	5	
dr1_x	0.03125	in
dr1_y	0.03125	
dr1_z	0.03125	
dr2_x	0.03125	
dr2_y	0.03125	
dr2_z	0.03125	
dr3_x	0.03125	
dr3_y	0.03125	
dr3_z	0.03125	
dF_1	22.3	%
dF_2	17.3	
dF_3	38.9	
dF_4	27.8	
dF_5	14.7	
dF_6	63.9	
dA	0.003906	in^2
dε	5E-07	
dE	0	lb/in^2
dGF	0.001	-
dV	0.000001	v

Figure 23: Screen shot of the uncertainty parameters in *uncertainty_const.*

3.1.2 Operation: Forward Method

One commonly used subroutine in the Forward Method is the *PropError* function. This function executes the uncertainty propagation method on a given symbolic expression. The user defined function was created by Brad Ridder to accept the following inputs: a symbolic expression, array of variables, array of variable values and an array of errors for each variable. With these inputs, the function could output a nominal value for the evaluated function as well as its uncertainty range and percent [7]. This subroutine was modified to give the user the option to

display and store the maximum sensitivity parameter. Below is the syntax of the modified function.

$[F_{NOMINAL}, dF, F_{\%ERROR}, S_{MAX}] = PropError(f(x_{1,\ldots}x_n), VarArray, Values, Errors, Option)$

From the *MainCGscript*, the user can use the inputs previously discussed to calculate CG by selecting the "Measure CG" option after running the script file. Before continuing, the code reminds the user to update the appropriate excel files and save them. Next, the user is asked to determine whether they wish to calculate the planar CG ($[cg_x; cg_y]$) or all three CG components ($[cg_x; cg_y; cg_z]$) at once. Only six load values are needed for the former but a total of 12 loads and a tilt angle, α, are needed to calculate the latter. A β angle is not necessary to find the full CG but can still be used. In both cases the initial load values of the load cells can also be recoreded.

After either a partial or full CG is requested, the code enters the *Force2cgShell* function where the user's CG decision modifies the inputs to the *force2cg* subroutine to calculate the CG and weight, along with their uncertainties. The syntax for the function is as follows:

$[cg_{val}, dcg, mg, dmg] = force2cg(F_0 , F_1 , \alpha , UNC_{F0} , UNC_{F1} , \beta)$

F_0 and F_1 correspond to the bipod loads at $\alpha = 0$ and $\alpha \neq 0$ respectively. The *force2cg* function was designed to accept different combinations of inputs by using a technique analogous to Bit Masking used in embedded systems [8]. An "isempty" command was used to convert the input array of *force2cg* into a bit array of ones and zeros. A (1) is returned if the entry is empty ([]) otherwise the entry is set to (0). Next, the subroutine defines a series of combinations that prompt the code to calculate either partial or full CG using logical if-statements. For example, if the first five of the six inputs of *force2cg* is non empty then the input checking Bit array would read [0 0 0 0 0 1]. This combination initiates the code to calculate the full CG. The table below summarizes the input combinations and their results.

Table 1: Input combinations of the *force2cg* function and their results.

Inputs	Bit Array	Result
All input are non-empty	[0 0 0 0 0 0]	full CG and uncertainty.
Only β is excluded	[0 0 0 0 0 1]	assumes β = 0, calculate full CG and uncertainty.
12 included load values but α is empty	[0 0 1 0 0 1]	Error message, Include α
6 load values and their uncertainties	[0 1 1 0 1 0]	partial CG (i.e [x_{CG}; y_{CG}])

Once the Bit array has been evaluated, *force2cg* calls on three other subroutines to generate an output for the CG and payload magnitude. Below is the sequence of steps in which in the function takes to output the results for the Forward Method (Refer to Figure 19 as an aid to visualize the nested function hierarchy):

- Inside *force2cg*, the following nested functions are evaluated: The outputs of *bipodNodes*, *weightMagUnc*, and *CGsymbExpression* are all feed into the *unc4cg* function. All this happens inside *force2cg*. *Unc*4cg receives the outputs from the previously name functions to evaluate the CG and its uncertainty.
    - *bipodNodes* - outputs $\boldsymbol{R_{bi}}$ and $\boldsymbol{dR_{bi}}$ from uncertainty propagation
    - *weightMagUnc* - outputs the payload magnitude (*mg*) and it's uncertainty (*dmg*)
    - *CGsymbExpression* - outputs a symbolic expression for the individual CG components using the symbolic toolbox in MATLAB.
- Next, the CG vector, payload magnitude and their uncertainties are displayed in the command window from *force2cg* while the outputs are feed through to the *MainCGscript.* The angle

parameters from ***CG_platform_constants*** are also displayed. The screen shot in Figure 24 shows the MATLAB inputs and results in the command window.

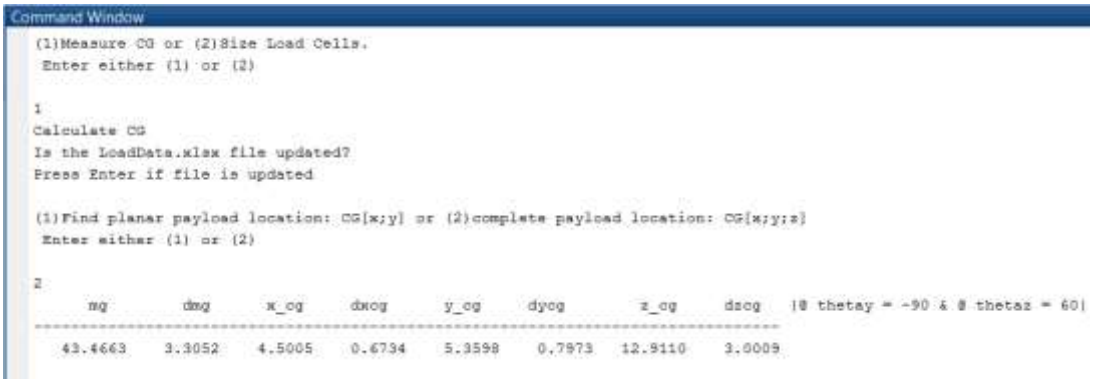

Figure 24: MATLAB command window screen shot of the outputs from the Forward Method.

Here are the steps require to run the *MainCGcode* to output the Forward results:

1. Updated the tables in the following excel files and sheets, then save them:
    a. Bottom Bracket and Angles Input tables in ***measured_const***
    b. Uncertainty table in ***uncertainty_const***
    c. Either the bipod load table for F_i or both the F_i and F_i' tables in **LoadData**
2. Run *MainCGcode*
3. Enter (1) to select the ”Measure CG task” and enter again to verify that the excel tables are saved
4. Choose whether a full or partial CG is to be calculated

3.2 Reverse Direction

3.2.1 Inputs for Excel: CG Platform Constants

Similar to the “Measure CG” option, the Reverse Method uses the **CG_Platform_Constants** excel file to summon the needed inputs. The same Bottom Bracket and

Angle Input tables need to be filled in order for the Reverse direction to work. Additionally, payload information such as the CG location and weight should be entered in the Load Cell Sizing table. Cells for the capacity and error for the load cells are available in this table. This code assumes that the load cells used in the CG Platform will all be the same model and as a result, they all share the same capacity and error. The excel file also reserves space to input new platform angles and payload parameters for a single case when the load cells are to be tared.

3.2.2 Operation: Reverse Method

The Reverse direction is initiated by choosing the ”Size Load Cell” option in *MainCGscript*. Once the sizing path is chosen, the user is asked whether they wish to calculate absolute bipod loads or tared loads. This decision is then fed into the *SizeLoadCells* subroutine where it determines the inputs used for the *cg2force* function. The *cg2force* function calculates the bipod load outputs based on the inputs entered in the excel file. In this subroutine, *bipodNodes* is again called upon to generate the intersection points of the bipod rods of each bipod, $\boldsymbol{R_{bi}}$. The load outputs are then processed in *SizeLoadCells* to calculate the absolute percent capacity and absolute error for each load reading. It should be noted that this function does not distinguish the difference between the various forms of load cell error such as non-linearity or hysteresis. The code simply accepts an error percentage and determines the corresponding absolute error ranges of the particular load values. It is up to the user to decide the type of error that is to be uploaded.

The load values generated from *cg2force* are equivalent to the absolute loads of each test. The term “absolute loads” in this context refers to the estimated reading from each load cell. To determine the tare loads, the second set of CG parameters is called upon to calculate a new set of absolute loads. The differences between the initial and new absolute loads are the tare load values. Again, the absolute error and percent capacities for the tare loads are calculated and displayed in the command window.

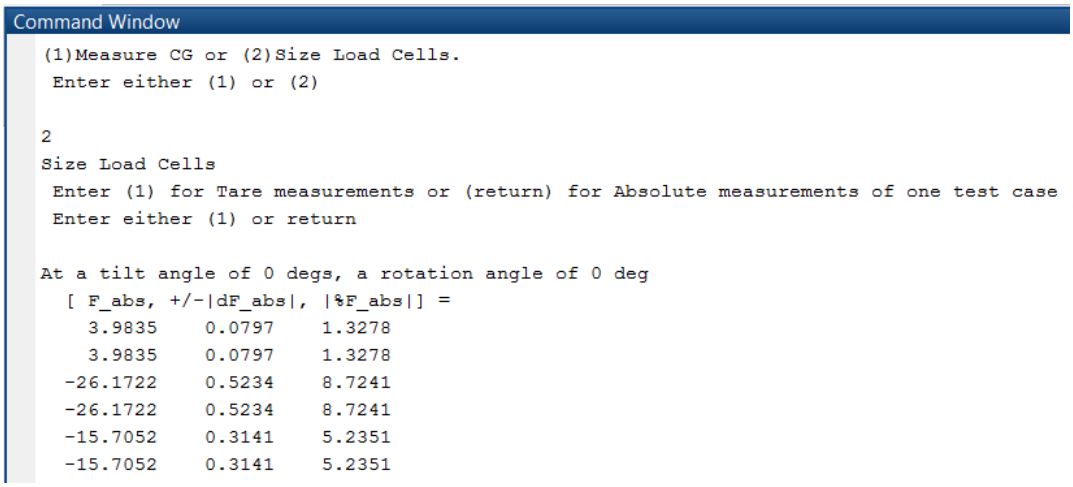

Figure 25: MATLAB command window snapshot of the result from the Reverse direction.

Below, the steps needed to run the *MainCGcode* to generate the results for the Reverse Method are outlined.

1. Updated the tables in the following excel files and sheets, then save them:
    a. Bottom Bracket, Angles Input and Load Cell Sizing Input tables in ***measured_const***. If tared values are desired then the Angle and Load Cell After Tare tables should be completed.
2. Run *MainCGcode*
3. Enter (2) to select the ”Size Load Cell Task”
4. Choose whether the Absolute or Tare loads are to be calculated

4. PLATFORM PROTOTYPE

Now that the code has been established, its validity can be tested by building a prototype of the CG Platform and preforming a series of test on the platform. The constructed prototype for the CG Platform Assembly (CGA) consists of two major components, the CG Platform containing the bipods instrumented with strain gauges and the adjustable CG Structure. A Tilt Table was also built to hold the CGA fixed at specified α angles during platform tilting tests. Together these components were used to conduct the CG measuring experiment to verify the math model.

4.1 Force Transducer

The original proposed design from JPL included load cells in the bipods of the platform. The estimated unit price of one inline load cell is roughly $350. This would mean that the total estimated cost for the six load cells is $2100. With a budget of $3000, the cost of the load cells would take up nearly 70% of the budget. Strain gauges instrumented to axial threaded rods could be used in place of the load cells as a low cost alternative. Instead of measuring load directly, the gauges would be used to record the strain in the bipod legs, these strains would then be converted to loads. Using strain gauges rather than loads cells free up space in the budget to purchase material needed to construct the Tilting Table, CG Structure and DAQ system. The constructed CG Platform Assembly for this project utilizes an array of strain gauges to perform the load measuring tasks.

4.1.1 Strain Gauge Selection

The following section reviews the process taken to select the strain gauges that were used in place of the load cells. Axial rods were used to provide a boding surface for the gauges while maintaining the truss configuration of the bipods.

Preliminary calculations to determine the expected strain in the bipod rods at both a maximum and minimum loading scenarios were needed to provide insight on reasonable operating ranges for the strain gauges. Only one rod (or bipod rod) needs to be considered to derive an expression for the strain in the bipod rod. First, a conservative minimum and maximum payload of 10 and 150 lbs was defined. For this analysis the weight of the Adaptor Plate and CG Structure were neglected.

To simplify these initial calculations, the platform is assumed to have a zero degrees tilt angle while the payload is placed at the center of the platform. Only bipod rod 1 on Bp 1 will be considered for this analysis. The cross sectional areas of all the rods are assumed to be the same and $\theta_1 = \theta_2 = \theta_i$ is established to model ideal conditions. The material of the axial rods are assumed to be aluminum, meaning that the rod exhibits isotropic characteristics. The free body diagram (FBD) in Figure 26 illustrates the symmetric loading on the two bipod rods. Since the payload is place at the center of the platform, each bipod holds a third of the actual payload weight ($W/3$). Due to symmetry, the reactional forces on each bottom bracket are a sixth of the applied load ($W/6$). Next, a cut of Bp 1 can be made and a new FBD is drawn to derive an expression for the strain the bipod rod 1. The sectional FBD is shown in Figure 27.

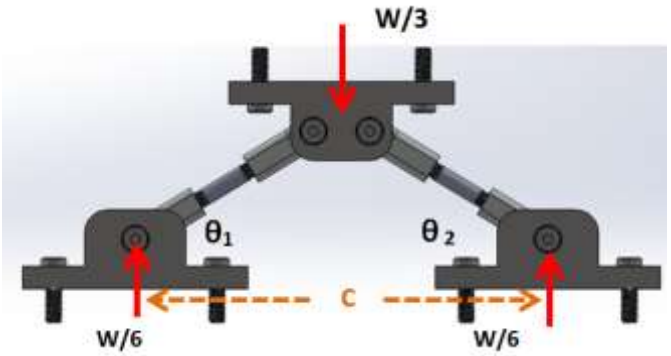

Figure 26: Free body diagram of a single bipod assembly with *W/3* applied.

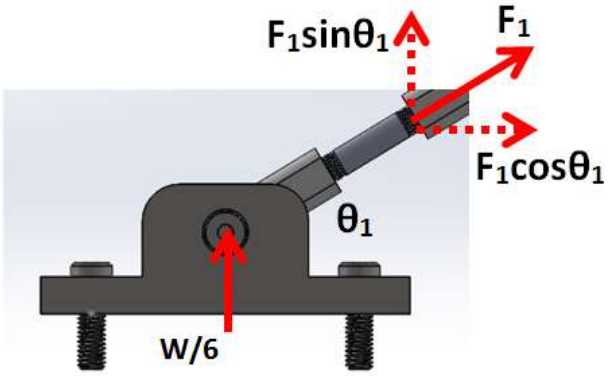

Figure 27: Section cut FBD of the bipod assembly.

Summing the forces in the vertical direction where up is positive leads to:

$$\sum F_y = F_1 sin\theta + \frac{W_{max}}{6} = 0 \tag{30}$$

The stress in the rod can be found by the dividing the bipod rod load, F_1, by A

$$\sigma_1 = \frac{F_1}{A} = \frac{W_{max}}{6 * sin\theta * A} \tag{31}$$

Now the strain can be extracted by dividing the stress by the Modulus of Elasticity.

$$\varepsilon_1 = \frac{\sigma_1}{E} = \frac{W_1}{6 * sin\theta * A * E} \tag{32}$$

Using a combination of static and linear elastic stress-strain relationships, the approximate strain in a single axial rod can be computed using equation (32). Measuring axial strains in an object experiencing uniaxial loading requires gauges to be bonded along the maximum and minimum principal axes. Since it was assumed that the rod is isotropic and will only undergo axial loading, the principal axes are coincident with the axial and transverse directions of the rod.

Vishay Micro-Measurements (a leading strain gage manufacturer) states that Tee Rosettes should be used when the principal directions of the specimen is well known [9]. Doing so will minimize errors due to gauge misalignment by eliminating the need to align an extra gauge. Using Tee Rosettes does not guarantee that there will be no error caused by misalignment, the Rosettes must still be aligned with one principal axis; alignment error can still arise. Conversely, when the direction of the principal strains are unknown it is recommended that three element rectangular or delta rosettes be used [9]. Errors caused by misaligned gauges will be discussed further in Chapter 5. For now, it is assumed that the gauges are perfectly aligned 90^o from each other, along the longitudinal and transverse axes. Together, the two gauges use Poisson's effect to track the axial strains in the rod.

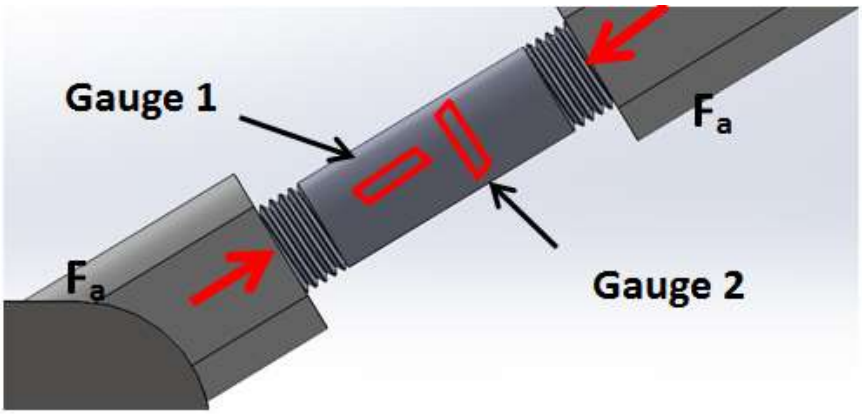

Figure 28: Desired strain gauge configuration and placement. Gauge 1 will be placed axially and Gauge 2 placed transversely.

To correctly measure strain from the Rosette gauges, the two strain gauges terminals must be wired to a wheatstone bridge to create a voltage differential between the gauge elements and dummy resistors. A circuit diagram of the wheatstone bridge can be viewed in Figure 29. Arms (1) and (2) in the bridge will be strain gauges while arms (4) and (3) are dummy gauges with the same nominal resistance of the strain gauges. Any transverse strains measured by the transverse gauge will be subtracted from the signal while only the axial strain will produce positive voltages. This configuration is called a half bridge circuit and knowledge of this is necessary to convert the voltage output of the bridge to strain.

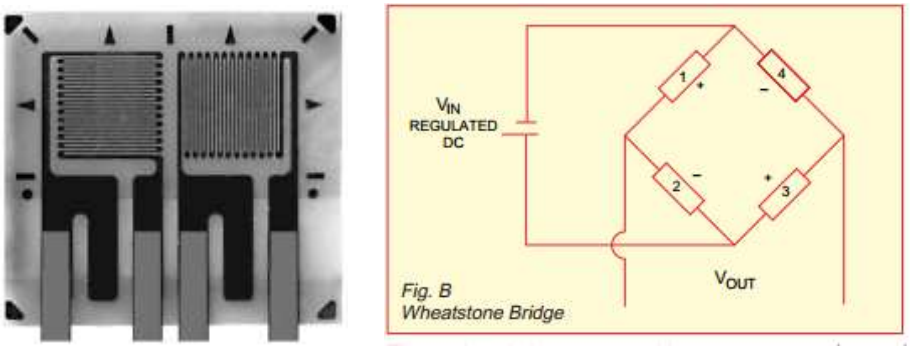

Figure 29: Tee Rosette to the left and a Wheatstone bridge diagram to the right.

Now that the Rosette gauge configuration has been established, the footprint of the gauge must be sized so that the gauges fit on the bipod rods. A 3 in long, partially threaded aluminum rod, with a 3/8 in diameter was chosen as the rod element in the bipods. The threaded portions of the rods only extend $1\frac{3}{16}$ in from both ends of the rod to the center leaving roughly 0.625 in of space along the length of a strain gauge. To make the strain gauge bonding process less strenuous, the middle portion of the rods were machined to a rectangular area. Bonding gauges to arched objects is possible but bonding them to flat surfaces is much easier. Machining the rod requires removing material so the available bonding area decreases. A nominal rectangular bonding area of 1/4 x 3/8 in^2 was selected and used to determine the maximum length and width of the gauge while a nominal cross sectional area of 1/4 x 1/4 in^2 was used to carry out the maximum and minimum strain calculations.

Details for this calculation can be found in Appendix B. These calculations together with the bonding area constraints were used in selecting the strain gauges that will be used on the bipods. Results from the analysis show that the minimum and maximum loads of 10 and 100 lbs produce strains of 46.5 and 464.9με respectively. This correlates to a minimum voltage of 0.0773 mV and a maximum voltage of 0.051 mV. The results from these calculations show that caution should be used when using light weights on the platform as the resulting voltage may be close to the resolution limits of the instrument and my not be measurable.

Gauges C2A-06-062LT-120 and C2A-06-031WW-120 from Vishay Micro-Measurements have strain ranges of 3%, gauge widths of 0.062 in or less and overall lengths smaller than 0.206 in. There should be no concern about the maximum limits of the strain gauges, instead small loads may prove to be difficult to read depending on the precision and resolution of the DAQ system [9]. All the other the parameters for these gauges fall within the established constraints. More information on the chosen strain gauges can be found in [10] and [11].

4.2 Center of Gravity Platform

The CG Platform houses three bipods, each of which contain two 3 in long, 3/8-24 threaded male end, 6061-T6 aluminum rods. As it was previously mentioned in the *Strain Gauge Selection section*, each rod was machined down to have a rectangular cross section to provide a flat surface for the strain gauges to be bonded to. Looking at the strain-load relationship, one will notice that the strain is inversely proportional to the cross sectional area of the rod. Therefore, a reduction in the cross section acts to increase the rod's strain.

$$\varepsilon_{rod} = \frac{F}{EA} \tag{33}$$

A nominal cross-section of 0.065 in^2 was chosen for the bipod rods. A strain magnification number (N) can be found by taking a ratio of the original circular cross section of 0.110 in^2 to the new cross section. The N for the nominal cross section is approximately 1.7, meaning that the strain magnitude using the new cross section is 1.7 times larger than the strain using the original cross section. The width and height of the cross section of each rod was measured using digital calipers with a 5.0E-4 in resolution. These dimensions were then used to determine the cross sectional area for all the rods after machining. Table 2 contains the cross sectional areas and magnification numbers for all the rods.

Figure 30: Two bipod rods during the strain gauges bonding process.

Six C2A-06-062LT-120 Tee Rosettes had initially been installed on the bipod rods, however, the gauges on rods 4 and 6 were damaged during calibration and were replaced with two C2A-06-031WW-120 Stacked Rosettes. The effect of using different strain gauges will be discussed in Chapter 5: *Bipod Rode Calibrations.*

Table 2: Rod cross-sectional area (A) and strain magnification number (N).

	A	N
Rod Index	[in^2]	-
1	0.0644	1.715
2	0.0687	1.608
3	0.0599	1.844
4	0.0694	1.592
5	0.0627	1.761
6	0.0645	1.712

Aluminum was selected as the material of choice due to its weight to strength ratio and ease of machinability. The nine brackets that make up the three bipods were individually machined from aluminum bar stock. The adaptor and base plates were also made of aluminum. Both plates were originally 27x24x1/4 in^3 in volume but were trimmed to octahedral shapes to reduce the net weight of the platform without sacrificing the rigidity.

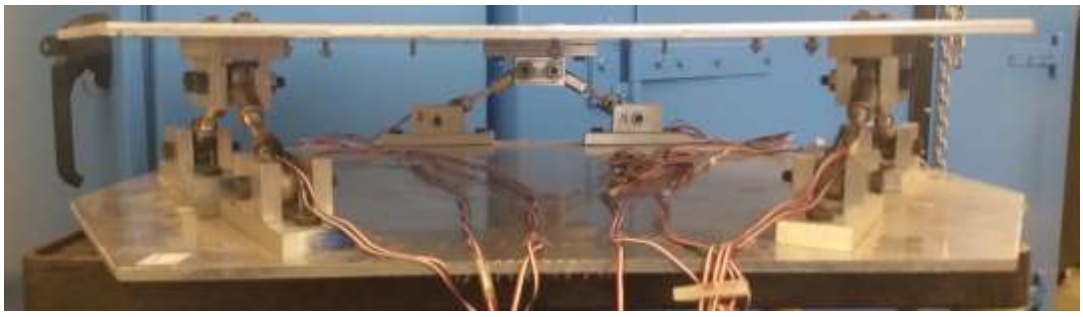

Figure 31: CG Platform with bipod instrumented with strain gauges.

After the CG Platform was completely assembled, the locations of each bottom bracket node were measured relative to the origin of the BCS using a tape measure with a 1/32 in

resolution. The θ_i angles of each bipod rod along with the γ angles were measured using a protractor with a 5^o resolution. These measurements and their instrument resolutions were recorded in the **CG_Platform_constants** spreadsheet.

4.3 Center of Gravity Structure

Figure 32: CG Structure with a payload.

The CG structure was constructed using aluminum T-slot framing. It provides rigid support for the shelf and payload assembly. The structure has a 17 3/8 x 15 7/8 in^2 footprint and stands 26 3/8 in tall. Four sets of holes were drilled along the vertical T-slot frames about 5 1/2 in apart, beginning at 5 3/8 in from the base of the framing. These holes were made so the payloads vertical position could be changed to four different discrete locations by adjusting the shelf height.

Nine 1 in diameter holes were drilled in a 1/4 in plate of aluminum so that the planar position of the payload could be adjusted to discrete locations relative to the x-y axes of the platforms BCS. This plate acted as a shelf which could slide vertically along the T-slot framing structure. A threaded portion of a 1 in diameter dumbbell bar was cut and used to keep weights at desired hole locations. The cut dumbbell bar, together with two dumbbell nuts, was used to secure the weights to the shelf during testing.

Table 3: Shelf hole and planar CG locations.

`	Left		Middle		Right	
	X	y	x	y	X	y
	[in]	[in]	[in]	[in]	[in]	[in]
Top	-4.23	5.091	0.02	5.091	4.27	5.091
Center	-4.23	1.841	0.02	1.841	4.27	1.841
Bottom	-4.23	-1.409	0.02	-1.409	4.27	-1.409

Table 4: Shelf height (z_{shelf}) settings.

	Setting 1	Setting 2	Setting 3	Setting 4
z [in]	5 3/8	10 7/8	16 7/16	22

The nine hole locations on the aluminum shelf in the BCS of the CG Platform was found by first measuring the hole locations relative to the bottom left corner of the CG Structure. These values were then added to the corner's measured position in the BCS. The vertical location of the top of the shelf at the four height settings were also measured relative to the BCS. Since the same tape measure used to measure the bracket nodes was used to measure the hole and shelf heights, the resolution of these parameters are all the same. Tables 3 and 4 show the measured planar hole locations and shelf height settings for the CG Structure. If weight is fastened to a certain hole, the weights planar CG coincides with the centers of the hole. However, the vertical CG of the

weights deviates away from the top of the shelf as more weights are stacked on top of each other while resting on the shelf. The actual CG of the payload can be found by adding the height of the shelf with the weighted average location of the payload's CG relative to the shelf; the actual CG can be expressed as:

$$Z_{cg} = Z_{shelf} + \frac{\sum_{i=1}^{n} w_i z_i}{\sum_{i=1}^{n} w_i} \tag{34}$$

where w_i is the magnitude of an individual weight and z_i is the vertical CG of the corresponding weight relative to the surface of the shelf. Further discussion on the assumptions made in order to use the above expression can reviewed in the Chapter 6: *Platform Loading Test section*.

4.4 Tilt Table

The Tilt Table was manufactured out of steel rectangular tubes and plates; it is comprised of a rectangular base frame and a tabletop. A piece of angle iron that runs across the width of the tabletop prevents the CGA from sliding off the tabletop while it is set to α angles other than zero degrees. Two C-clamps on the posterior end of the Tilt Table fixes the CGA to the tabletop. A small bushing was created by boring out a 3/16 in hole in a 1/2 in piece of steel bar stock and cutting it into four pieces. Two of the bored out pieces of steel were welded onto each vertical post, near the base. The other two were welded underneath the tabletop, near one of the edges. Two 3/16 in bolts were used to form pinned joints between the posts of the base frame and the tabletop. A factor of safety (F.S.) for the bushing welds was calculated before any payload tilt tests were performed to assure that the welds possessed ample strength and would not fail; a similar study was done for the welds at the base of the post. Using the Minimum Weld-Metal

Properties tables in *Shigley's Mechanical Design* book [9], a yield and tensile strength of 50 and 62 ksi for a E6010 AWS electrode was used to evaluate the Factor of Safety. A F.S. of 5.7 and 4.1 was calculated for the pinned joints on the post and at the base, respectively. These calculations can be found in Appendix E.

Figure 33: (Left) Rear view of the tilt table at 30^o. (Right) Front view of the tilt table at 0^o.

The overall length of the frame was determined by calculating the maximum moment that would be applied to the Tilt Table if the CGA, including the payload, was tilted to 90^o. A maximum net weight of 140 lbs and maximum shelf height (z_H) of 24 in was used in this study. The estimated maximum net weight refers to the sum of maximum payload of 100 lbs and the tabletop which weighs approximately 40 lbs. The goal of this study was to size the front member of the Tilt Table in such way that the CGA could be set to 90^o and not cause the entire system to topple over. Results from the calculation showed that the member must be at least 31 in long to not tip over in the extreme moment case. By observation, one can realize that a counter moment will prevent tipping if the front member is made longer than the z_{cg}. Due to a shortage of material, the front member was made 25.5 in long. At this length, a note was made to not set z_{cg} higher than 15 in to avoid tipping if a maximum payload of 100 lbs was used.

In the original design, the thin quarter circular plates on the base frame was intended to support the tabletop at 30^o and 60^o. However, misalignment between the holes on the bottom of the tabletop and the holes on the quarter circular plates prevented this from being done. As an alternative, two 14 and 24 in T-slot members were attached to the back end of the base frame by a pin connection. The two sizes correspond to the sine of 30 and 60 degrees multiplied by the distance between the post and the rear perpendicular member.

4.5 Hardware/Software

A DAQ system was designed to take semi-continuous measurements of the strain gauges during the platform loading tests. The DAQ system includes six bread-boarded wheatstone bridges, two Bridge Amplifiers, two NI USB A/D converters and a Strain Gauge Measuring VI file created in LabView. Data was also taken using a P3 Strain Indicator box from Micro-Measurements coupled with a Switch Terminal to produce test benchmarks for the proposed DAQ system. The preceding section discusses the function and usage of both DAQ methods.

4.5.1 Quad Bridge Amplifier and USB DAQ

Each wheatstone bridge services two strain gauges and uses two 120 Ω completion resistors to make a single half bridge circuit. The lateral nodes from each wheatstone bridge were wired to open input terminals on the Quad Bridge Amplifiers (QBA). A 3.3 v excitation voltage is supplied to each wheatstone by the QBA. The circuit diagram on the right in Figure 34 illustrates how the gauges are wired to the wheatstone bridge. The QBA receives the small differential voltage from two opposite nodes of the wheatstone bridge and amplifies the signal by a factor of 105. More details on the set up of the QBA can be found in *Quad Bridge Amplifier: Quick Setup Guide* [9].

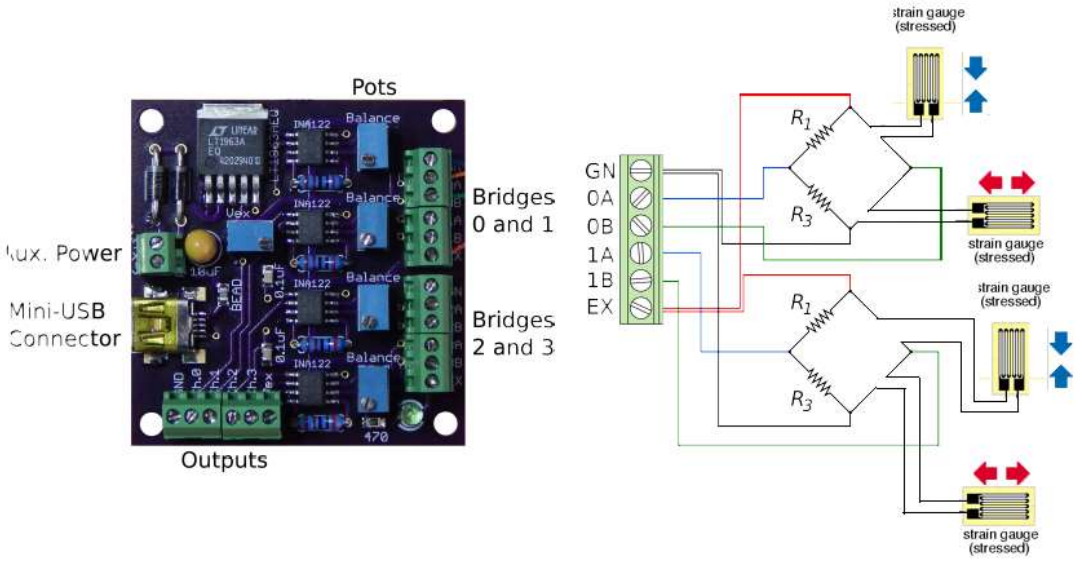

Figure 34: (Left) Quad Bridge Amplifier . (Right) Two wheatstone bridge circuits.

Outputs from each bridge was then feed into the NI USB A/D converter so that the digitized data could be read using the LabView software on a laptop. A VI program in LabView was created to convert the voltage output from the gauges, to strain and write relevant data to an excel file at the push of a button. Since all the planned CG test were static, any relatively large frequencies in the strain signal was assumed to be noise and a low pass filter with a cutoff frequency of 5 hz was applied to the signal. An initial sampling rate of 50 hz was selected but a much larger sampling rate could be used since the maximum sampling rate of the A/D is 10khz.

4.5.2 P3 Strain Indicator and Switch Terminal

The P3 Strain Indicator and Switch Terminal was the first method used for data acquisition in the CG Platform loading tests. Using the portable, battery powered, precision instrument prior to using the designed DAQ system was done to first generate benchmark results to compare the operation of the DAQ system. The P3 model contains built in noise rejection, proprietary scaling and nonlinearity algorithms [14]. It is also capable of taking measurement from other strain-gauge based transduces such as load cells and linear actuators.

Figure 35: P3 Box and Switch Terminal.

The P3 box only has the ability to read strain from four different bridges simultaneously and cannot be used to measure the six half bridge arrangements that the platform possessed by itself. To accommodate for all six half bridges, three bridges were wired normally to three channels on the P3. The remaining three bridges were then wired to a Strain Gauge Switch Terminal. Using spare leads, the output of the Switch Terminal was then wired to the last open channel on the P3. The P3 was adjusted to a half bridge setting and the gauge factor for the Tee Rosettes, 2.055, was input for the first three strain gauges. The last three gauges on the last P3 channel were set to 2.175. It was important to note that the bipod strains measured from gauges 4 and 6 will have slightly lower strains since the correct gauge factor is 2.055. This is true since $\varepsilon \propto 1/[\text{gauge factor}]$. In the following chapter, the process taken to calibrate the strain gauges and the bipod rods is expanded.

5. BIPOD ROD CALIBRATION

Before any tests could be run on the CGA, the precision and accuracy of the bipod rod's ability to measure loads had to be investigated. A qualitative loading study was conducted by hanging a series of weights from the bipod/endrod assemblies to test out the designed DAQ system. Results from this study were used to formulate parameters needed to conduct a more comprehensive loading study using an Instron machine.

5.1 Weight Hanging Trial Test

Endrod assemblies 1-6 were removed from the CG Platform and wired to the designed DAQ system. A single endrod assembly includes the bipod rod and the strain gauge bonded to its surface along with the two endrods screwed onto both ends of the bipod rod. All six assemblies were then hung from a wooden frame using steel welding filler rods as hooks. Zip-ties were tied around the dangling end of the assemblies and the individual weights. Another hook was used to hang the weights on the assemblies. Figure 36 shows all six assemblies hanging from the wooden frame.

Figure 36: Hanging weights set up.

Combinations of two 7.5 lb and two 10 lb weights were used to load the individual endrod assemblies in tension while the LabView VI program read and recorded the desired data to comma-separated text files. At smaller loads, the measured voltages from the endrod assemblies showed large differences from the expected voltages. This trend was true for all the assemblies and occurred for loads equal to or less than 10 lbs. One possible explanation for the error is that moment induced bending appears when the rod goes from no load to 10 lbs and as a result, corrupts the axial load readings. Another plausible explanation is that loads under 10 lbs are exciting strain voltages that are too small for the DAQ system to measure. The NI DAQ has a 12-bit analog input resolution and a 5 V range. This translates into a voltage resolution of roughly 1.22 mV

$$V_{resolution} = \frac{V_{range}}{2^n - 1} \tag{35}$$

According to the theoretical calculations in B. 2 of the Appendix, the voltages in the endrods become too small to measure at applied smaller than 3.3 lbs. To avoid this small range of loads, a feature was added in the VI program which stores a reference voltage. In the VI program, the reference voltage is subtracted from preceding voltage measurements and acts as a new datum. As an effect, the endrod assemblies could be calibrated to read zero volts, or zero strain, at any starting applied load. This is analogous to the tare feature in load cell DAQ systems. The image in Figure 37 shows endrod assembly no. 5 loaded with a 10 lb preload weight and a 10 lb payload weight.

Figure 37: Endrod assembly no. 5 loaded with a 10 lb preload and 10 lb payload weight.

The weight hanging trial tests conducted on the endrod assemblies helped to shed light on large errors that could arise in low load scenarios. These tests also helped verify that the designed DAQ system was functioning properly. Some downsides to the weight hanging test was its inability to inflict compressional loads to the rods and the limited amount of available weights. A more comprehensive and analytical approach to calibrating the bipod rods was taken using data generated using an Instron.

5.2 Calibration Using the Instron

The Instron machine was utilized to apply a larger range of loads onto the endrod assemblies. A custom fixture was welded together so that a pin-pin loading condition could be simulated while the assembly was installed in the Instron. Using position control on the Instron, a total of about 10-12 different load magnitudes in both tension and compression were applied to the six endrod assemblies. The lab coordinator recommended that the position control feature on

the Instron be used rather than the load control feature. In this mode, small displacements on bottom clamp of the Instron were controlled to elicit loads on the endrod assembly. After each desired load was achieved, the load value from the digital read out (DRO) on the controller of the Instron was recorded while the voltage output from the strain gauge was recorded using LabView. If the position of the Instron actuator remained fixed while applying a load to the endrod assembly, the measured load magnitude from the DRO seemed to drop at what appeared to be a slow constant rate, as time elapsed. It was thought that the load cell of the Instron might be detecting small amounts of relaxation in the assemblies. Logging the data immediately after each loading scenario helped to avoid recording drifted load values from the DRO. The data was then post processed in Excel and MATLAB. Figure 38 shows the test set up for one of the endrod assemblies.

Prior to taking any strain measurements from the strain gauges, the wheatstone bridge arms were first balanced to read 0 V between the opposite nodes. Next, the initial output voltages of the QBA were adjusted to approximately half the excitation voltage (1.65 V). This was done to avoid any systematic errors that could arise if the output voltage were to drift to a negative value [9]. If a negative voltage is used as an input to the QBA while the initial unstrained QBA voltage is set to 0 V then the resulting output will also be 0 V. Any preceding strains calculations will be incorrect. On the other hand, changing the initial output voltage of the QBA unknowingly introduced nonlinearity errors into the system since the initial output voltage was changed by unbalancing the wheatstone bridge [10]. As it was mention in the section *Weight Hanging Trial Test* section, a gain was also added to the Labview VI file to bring the no-load output voltage down to 0 V. Doing this unfortunately helped to mask potential nonlinearity error rather than fix it. A solution for dealing with nonlinearity errors will be discussed shortly.

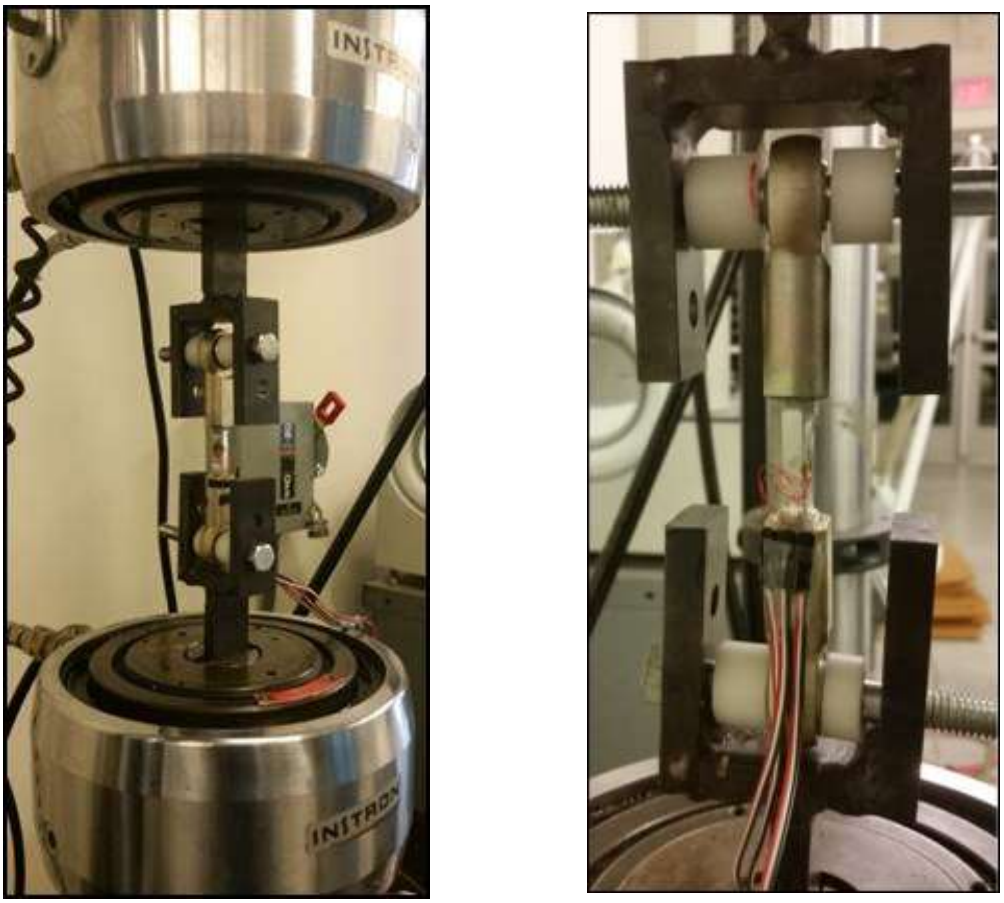

Figure 38: (Left) Wide view of the endrod assembly installed into the Instron. (Right) Plastic spacers were placed in both sides of the endrods to minimize the movements of the assembly along the bolts.

5.2.1 Results

The voltages gathered from the Instron test were converted into microstrain using equation (36), the derivation for the microstrain equation can be view in Appendix B. Next, these microstrains were changed to load values by utilizing the stress-strain relationship for materials (i.e. $E=\sigma/\varepsilon$). The microstrains produced from the loads applied by the Instron were multiplied by the corresponding cross sectional area of the rod and Elastic Modulus for aluminum. Typically, the Elastic Modulus is about 2% greater in compression than it is in tension for aluminum, however, in for these test the Modulus was assumed constant in both tension and compression [11].

$$\varepsilon_\mu = \frac{4\left(\frac{v_i}{v_{exc}}\right)}{(1+\nu)GF} * 10^6 \tag{36}$$

$$F_{compression} = (1.02)EA\varepsilon_\mu * 10^{-6} \tag{37}$$

Misalignment between the strain gauges and the longitudinal axis of the rods does exist and can create error in the strain readings. Figure 39 shows how the percent error in the strain values for a half bridge set up varies as the angle of gauge misalignment (ξ) increases. The plot was generated using the following equation:

$$n_{\sigma_p} = \frac{(R_\varepsilon - 1)(1-\nu)(1-\cos 2\xi)}{2(1+\nu R_\varepsilon)} \tag{38}$$

where R_ε is the ratio of principle strains and ν is the Poisson's ratio [12]. All the gauges have misalignments smaller than 3° except for gauge no. 5 which has a misalignment of about 4°. From Figure 39, a 4° gauge misalignment only produces a -0.5% error in the strain values. It appears that gauge misalignment only has a small contribution to the total error seen in the strain gauges. The bipod rods with gauge misalignments equal to 4° or small will be neglected since the actual strain error observed from the Instron testing results were a few magnitudes larger. (The figure is only valid for compressional loads, however, tension loads have similar error magnitudes).

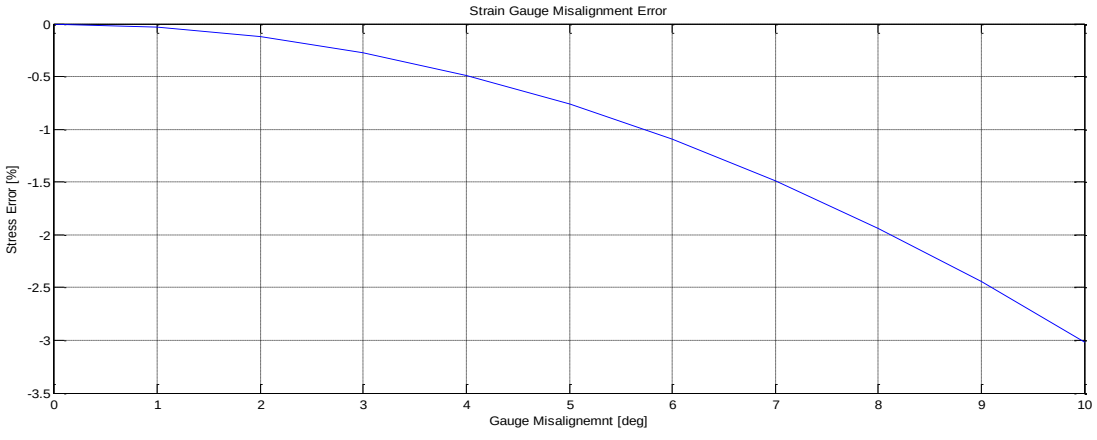

Figure 39: Expected error in rectangular rosette strain gauges as the gauge misalignment increases [12].

The plots and tables for the results from the Instron testing can be viewed in Appendix F, the plots show that errors exist in the endrod assemblies. Assemblies 1-5 show that the strain seen in the gauges do, in fact, increase linearly with an increase in load magnitudes, however, the difference between the measured and expected strain vary between ±23.2 % for applied loads magnitudes larger than 10 lbs. In most cases the endrod assemblies showed the smallest differences at the higher tested load magnitudes and the largest differences at smaller load magnitudes. This same trend was observed in the preliminary weight hanging loading tests. Endrod assemblies 2, 3 and 5 illustrated this trend best. Assembly 6 on the other hand showed slightly opposite results with the strain difference increasing at higher load magnitudes. Both assemblies 1 and 4 appear to overestimate the strain. Two primary methods can be used to reduce the errors seen in the endrod assemblies but the method that is to be implemented depends on the type of error that is present in the assemblies. These methods include:

- Shunt Calibrations
- Nonlinearity correction

Ideally, shunt calibration should done before any strain measurements are recorded to minimize the presents of nonlinearity error seen in strain gauges readings [13]. Nonetheless, in these particular tests, shunting did not reduce the errors as was expected. One shunt resistor was placed in parallel with the terminals of a single strain gauge on the breadboard to simulate a strain across the gauge by modifying its resistance. The potentiometer of the QBA was then adjusted to achieve the proper voltage that corresponds to the shunted strain gauge. A 599 kΩ resistor and a 119 kΩ resistor were used separately as shunts on the bridges. These resistances were selected to simulate 73.18 and 365.88 ε_{μ} respectively. The proper amplified voltages for the two resistors are 0.0173 V and 0.0867 V, they were calculated using the equations below.

$$\varepsilon_{\mu} = \frac{120\Omega}{(120 + R_s)(1+\nu)GF} * 10^{6} \tag{39}$$

$$V_s = \frac{(GF)\varepsilon_{\mu}}{\left[4 + 2GF * \varepsilon_{\mu}(1-\nu)\right]} * G_{amp} * 10^{-6} \tag{40}$$

The downside to this method was that the starting output voltage after shunt was too close to the voltage floor of the QBA and therefore shortened the range of relative negative voltages that could be read. If compressive loads larger than 50 lbs were applied after shunting, the magnitude of the negative bridge voltage would be larger than the allowable negative voltage range of the QBA; the QBA would then read an output voltage of 0 V [9]. To accommodate for a wider of negative voltages or compressive loads, the initial output voltage was adjusted to 0.2 V instead of the small initial shunting voltages.

In the case of endrod assemblies 1 and 4, a nonlinearity correction was applied to decrease the magnitudes of the strain readings from the gauges. From Figure 40 and Figure 41, the measured strain magnitudes recorded from the gauges are larger than the expected or theoretical strains making the slopes of the measured data steeper than that of the theoretical slops. Using the wheatstone bridge nonlinearity relationships provided by Vishay [10], the error was minimized.

This was done by applying the nonlinearity expression to the sum of the initial and loading strains. The correction was then applied to the initial strain and then the difference between the summed correction and the initial corrected is taken to find that relative nonlinearity corrected relative strains. The correction equations are listed below where $\varepsilon_{nonlin,cor}$ is the correction expression , $\varepsilon_{i,cor}$ is the corrected initial strain, $\varepsilon_{load,cor}$ is the corrected total strain and $\varepsilon_{corrected}$ is the nonlinearity corrected relative strain.

$$\varepsilon_{nonlin,cor} = \frac{2\varepsilon}{[2(1+\nu) - 2GF\varepsilon(1-\nu) * 10^{-6}]} \tag{41}$$

$$\varepsilon_{corrected} = \varepsilon_{load,cor} - \varepsilon_{i,cor} \tag{42}$$

Doing this produces corrected strain values relative to the initial strains. After applying the corrections, a drop in the strain differences for the measured strain readings from endrod assemblies 1 and 4 was observed. These steps were coded into a MATLAB function to accept initial measured strains and correct the loaded strains. This function can be viewed in Appendix D with the rest of the CG code.

Table 5: Strain and load values for strain gauge no. 1 before and after nonlinearity corrections were made.

		Before Correction			After correction		
Instron Load	ε_{actual}	ε_{meas}	ε_{DIFF}	F_{meas}	$\varepsilon_{corrected}$	ε_{DIFF}	F_{cor}
[lbs]	[με]	[με]	[με]	[lbs]	[με]	[με]	[lbs]
5.0	7.8	17.4	9.6	11.2	2.4	-5.3	1.6
13.6	21.1	23.4	2.2	15.0	24.4	3.3	15.7
35.8	55.6	82.2	26.6	52.9	48.7	-6.9	31.4
80.0	124.2	168.2	44.0	108.3	120.9	-3.4	77.8
106.6	165.5	209.2	43.7	134.7	172.1	6.6	110.9
-11.7	-17.9	-18.1	-0.2	-11.9	-22.2	-4.3	-14.3
-20.4	-31.0	-42.6	-11.6	-28.0	-30.4	0.6	-19.6
-34.6	-52.7	-52.3	0.4	-34.3	-66.5	-13.9	-42.8
-65.2	-99.3	-128.7	-29.4	-84.6	-102.8	-3.5	-66.2
-101.1	-153.9	-171.6	-17.7	-112.7	-180.1	-26.2	-116.0

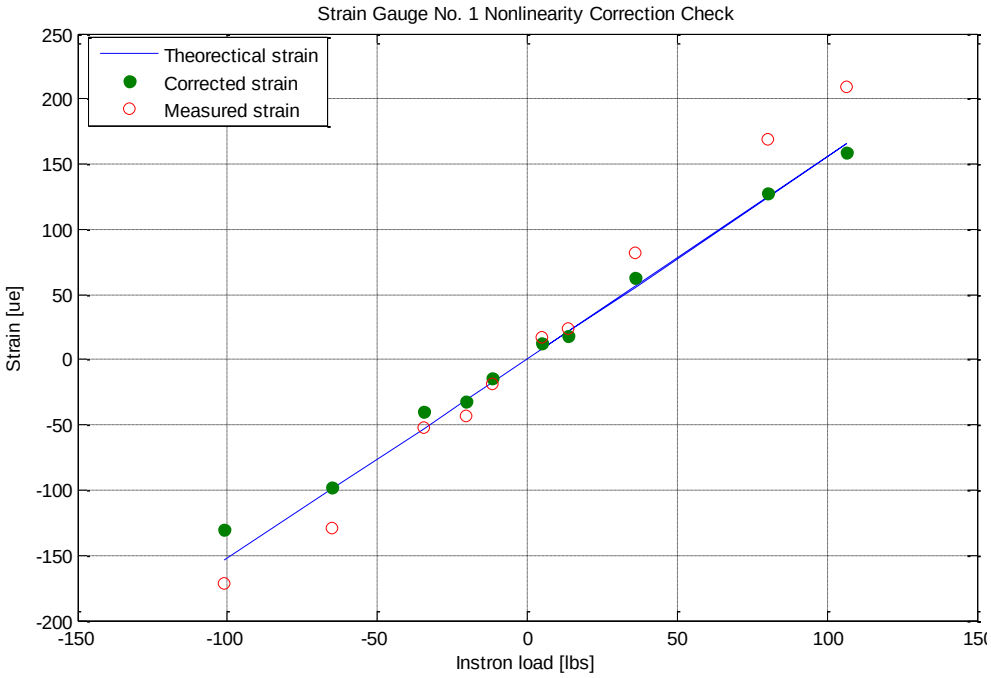

Figure 40: Measured strain nonlinearity correction for strain gauge no. 1.

Table 6: Strain and load values for strain gauge no. 4 before and after nonlinearity corrections are made.

		Before Correction			After correction		
Instron Load	ε_{actual}	ε_{meas}	ε_{DIFF}	F_{meas}	$\varepsilon_{corrected}$	ε_{DIFF}	$F_{corrected}$
[lbs]	[με]	[με]	[με]	[lbs]	[με]	[με]	[lbs]
2.8	4.0	0.0	-4.0	-69.2	8.1	4.0	5.6
18.5	26.7	39.3	12.6	32.8	24.0	-2.6	16.7
49.5	71.3	81.1	9.7	9.5	82.2	10.8	57.0
50.5	72.8	81.1	8.3	7.9	85.1	12.3	59.0
66	95.1	121.6	26.5	19.3	99.5	4.4	69.0
92	132.6	162.5	29.9	15.6	143.9	11.3	99.8
115	165.7	203.1	37.4	15.7	180.0	14.3	124.9
0.3	0.4	0.0	-0.4	-69.4	0.9	0.4	0.6
-4	-5.8	-5.5	0.2	-2.7	-7.5	-1.7	-5.2
-19.47	-28.1	-41.9	-13.8	34.2	-25.0	3.1	-17.3
-69.5	-100.2	-151.9	-51.7	35.8	-87.2	12.9	-60.5

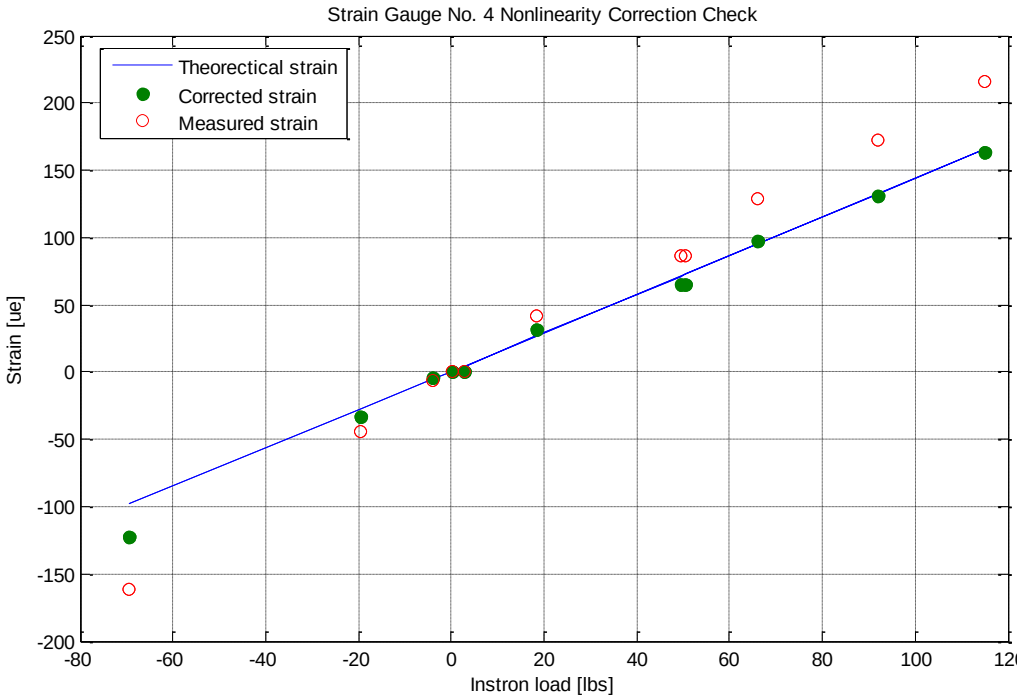

Figure 41: Nonlinearity correction for strain gauge no. 4.

This method was also applied on strain readings from assembly 6; however, it did not work as well as it did for assemblies 1 and 4. In Figure 42, notice how the measured strain values appear to deviate from the expected (theoretical) values as the magnitude of the applied Instron load increases. Implementing the correction only benefited endrod assembly 6 when it was loaded in compression. While in tension, the corrected strains were under estimated with strain values smaller than the original measured strains and the theoretical values. This large nonlinearity in gauge no. 6 was thought to be a sign showing that the gauge is damaged. Strain gauge no. 6 was then replaced and retest to verify that it's strain-load curve was indeed linear. The plots and table of results for the new gauge on endrod assembly 6 can be view in Table 21 and Figure 72 in Appendix F.

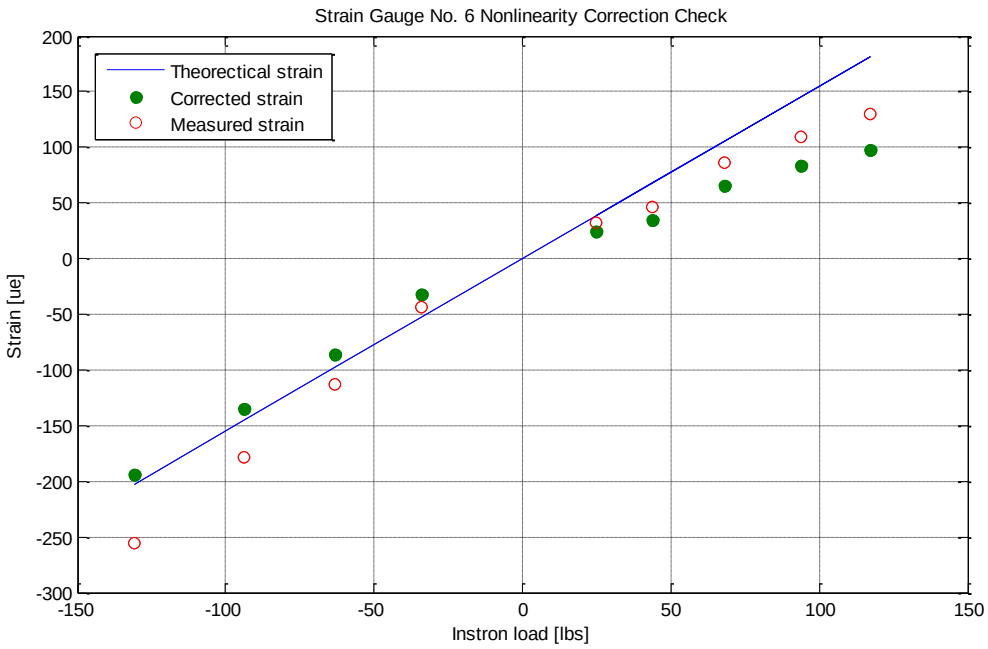

Figure 42: Nonlinearity correction check for strain gauge no. 6.

With the measured weight of the entire CG Structure being roughly 42 lbs, the endrod assemblies will be preloaded with a net weight of 42 lbs in compression. This is true when the tilt table has no tilt angle (i.e. $\alpha = 0^{o}$) and is parallel with the horizontal. This means that the endrod assemblies will initially be in compression before any payload is applied. Strain gauges 2, 3 and 5 all show less than 15 % difference in measured strain when compared to their theoretical counter parts if compressive loads greater than 75 lbs are applied. Using equation (1) in the *Strain Gauge Selection* section, the minimum net weight applied to a single bipod rod can be calculated. In this case the minimum net applied weight is 258 lbs. Therefore, to maintain strain errors smaller than or equal to 15 % error, payloads larger than 215 lbs must be used (Note: The minimum payload weight is the difference between the net weight and the preload weight of the CG Structure). Applying this amount of weight to achieve the desired error ranges is not possible for this project. A decision was made to use weights smaller than 100 lbs to conduct the platform loading test safely. The CGA and Tilting Table would need to designed in order to accommodate for such

large loads. Errors larger than 15% are expected in the CG measurements if payloads smaller 215 lbs are used. It is important to note that the CGA can still be used to validate the CG math model. The math model could be considered semi validated if the uncertainty range in the CG measurements include the actual CG data points.

Strain gauges 1, 4 and 6 work slightly better at smaller loads after the nonlinearity correction is applied. The average error observed in the calibration tests for each endrod assembly was catalogued in the ***uncertainty_const*** sheet of the **CG_platform_contants** excel book as an heuristic error. The *MainCGcode* uses these error values to evaluate more approximate uncertainty ranges for the measured CG vector. With a better understanding of the functionality of the bipod rods and the Designed DAQ system, the CG Platform and Structure were reassembled to perform The Platform Loading experiments to investigate if CG mathematical model could be validated.

6. PLATFORM LOADING TEST

Figure 43: Test set up using the P3 strain indicator (Blue box) and the Switch Terminal (Yellow box) as a DAQ.

The effectiveness of the MATLAB algorithm to calculate a CG was tested using the CG Assembly (CGA) and Tilting Table. A series of weights were placed on the CG shelf at various locations and shelf heights. Strain data from the bipod rods at these weight locations (or CG's) were then recorded at different platform tilt angles (α). The P3 Strain Indicator box coupled with the Switch Terminal was used first to display the strain outputs from the strain gauges during testing. CG results obtained from strain gauge data recorded by the P3 box will serve as a benchmark for the designed DAQ system to validate the CG model. If the results from the benchmark tests show poor results then the tests using the designed DAQ system will be abandoned and a FE model will be needed for code validation. A total of 88 data points were recorded from the platform loading tests. Each data point included the following entries:

- All six strain readings from each bipod rod
- Payload location on the shelf
- Platform tilt angle (α)
- Payload weight magnitude
- Shelf height (z_{shelf})

Five different planar shelf locations, two shelf heights and four α angles were tested for two weight magnitudes. The in Figure 44 shows the tested planar CG locations relative to the BCS. Both 35 and 42.5 lb payloads were tested at 0, 30, 60 and 90 α angles for a z_{shelf} of 5.5 in. These test were then repeated with the shelf height set at 10.875 in. The payloads or weights were measured on a digital scale with a 0.005 lb resolution; the approximate magnitudes for each weight were 35.77 and 43.47 lb. A table showing all the test combinations can be found in G. 1 in Appendix G.

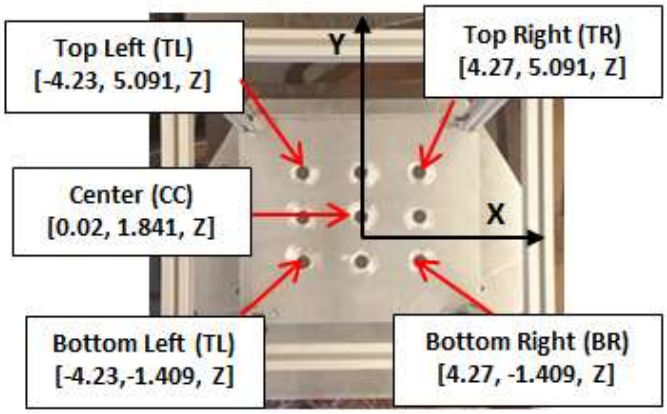

Figure 44: (Left) Data recording structure in excel. (Right) Desired test locations on shelf.

The initial strain or no-load readings of the strain gauges were recorded when the CG Assembly (CGA) was set at the four different α angles. This was done before any payloads were placed on the shelf. The no load strain readings were used to compensate for the preload in the

bipod rods created by the CG Structure and Adaptor Plate. This was done by calculating the nonlinearity corrected relative strains, $\varepsilon_{corrected}$. Figure 45 shows the no load tests for the CGA at $\alpha = 30^o$.

Figure 45: No load Test with $\alpha = 30^o$ tilt.

After the no load strains were recorded, payload weights were added to the shelf and the resulting strains in the bipod rods were measured. The bipod rod strains at a particular shelf location were measured with 35.77 lbs of weight and then with 43.47 lbs of weight at $\alpha = 0^o$. The 35.77 lbs of weight were fastened to the shelf using the dumbbell bar and nut. Rather than unfastening the nuts to add the extra 7.5 lb (nominal) to achieve a 43.47 lb payload, the extra weight was stacked on top of the already fastened weights. This was done to help speed up the testing process so more combinations can be tested. Figure 47 illustrates this as a 7.5 lb weight rests above the nut and weights while the platform is tilted to 90^o. The actual vertical CG (z_{act}) was calculated using the z_{cg} expression in equation (43). This equation accounts for the distance of each weight relative to the surface of the platform (z_i). In all cases, the weight of the aluminum shelf, dumbbell rod and nuts were neglected.

$$Z_{cg} = Z_{shelf} + \frac{\sum_{i=1}^{n} w_i z_i}{\sum_{i=1}^{n} w_i} \quad (43)$$

This process was then repeated for α = 30, 60 and 90 degrees at the same shelf location. Once all the stain data for the different α angles and payload magnitudes at was recorded, the entire process was repeated for a different payload shelf location. Again, a table of the test combinations can be found in G. 1 in Appendix G.

Figure 46: Test at α = 60^0 with 35 lbs payload (nominal) and the self-height set to 5.5in.

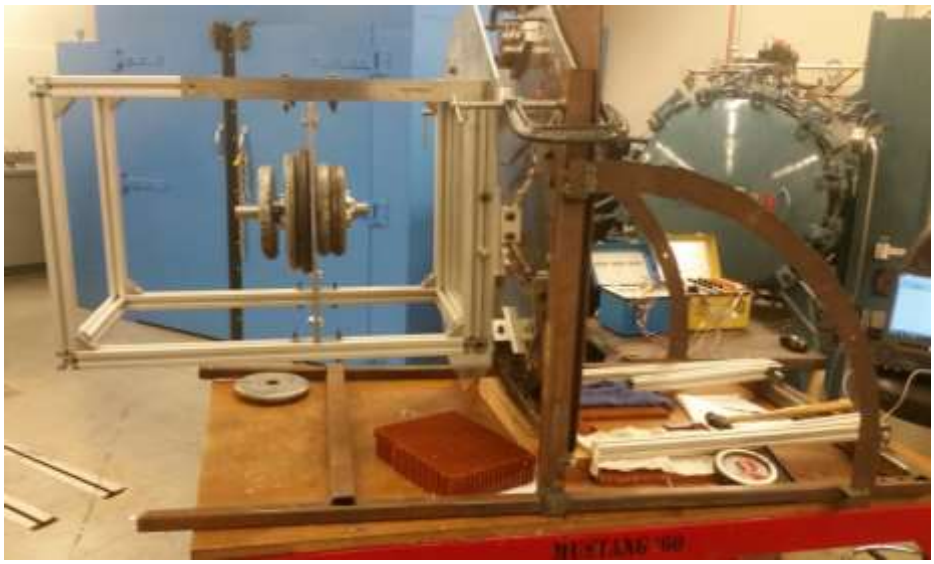

Figure 47: Test at $\alpha = 90^{o}$ with a 43.77 lb payload and a z_{shelf} set to 10.875 in.

6.1 Platform Tilting Results

The next two tables contain the planar CG and weight results from the Platform Loading Test when $\alpha = 0^{o}$. Table 7 displays the actual weight magnitudes and planar CG coordinates ($\underline{x_{act}}$ and y_{act}) for the different tested payload locations. The actual weight magnitudes (w_{act}) refer to the values of the weights recorded from measuring the weights on a digital scale. The uncertainty of the scale was used for the uncertainty in the uncertainty of the actual weight magnitude, ($\sigma_{w,a}$). The actual planar CG coordinates refer to the tested hole locations on the shelf. These were measured directly using a tape measure. The resolution of the ruler was used for the uncertainty in the actual planar CG locations, ($\sigma_{x,a}$ and $\sigma_{y,a}$). The right half of the table contains the measured planar CG coordinates and weight values along their uncertainties. Table 9 compares the actual and measured results for the planar CG coordinates and weight magnitudes for the different payload locations on the shelf. The right half of the table shows the actual values while the left half displays the differences between the between the measured and actual values.

Table 7: Actual and measured results for the planar CG components for z_{shelf} = 10.875 in.

	Actual						Measured					
Payload Shelf Location	W_{act}	$\sigma_{W,a}$	x_{act}	$\sigma_{x,a}$	y_{act}	$\sigma_{y,a}$	W_{meas}	$\sigma_{W,m}$	x_{meas}	$\sigma_{x,m}$	y_{meas}	$\sigma_{y,m}$
	[lb]		[in]				[lb]		[in]			
Center	35.77	0.005	0.02	0.0313	1.841	0.0313	31.539	5.517	0.279	0.998	2.889	1.649
	43.47	0.005					38.871	6.761	0.387	1.020	2.938	1.630
Bottom left	35.77	0.005	-4.23	0.0313	-1.409	0.0313	30.053	4.628	-2.923	1.260	-0.751	1.220
	43.47	0.005					37.175	5.529	-3.425	1.261	-0.762	1.158
Top Left	35.77	0.005	-4.23	0.0313	5.091	0.0313	29.614	5.572	-3.756	1.118	4.200	1.989
	43.47	0.005					37.481	6.897	-3.472	1.051	4.183	1.945
Top Right	35.77	0.005	4.27	0.0313	5.091	0.0313	33.956	7.315	3.213	1.330	6.488	2.463
	43.47	0.005					39.722	8.498	3.259	1.273	6.618	2.486
Bottom Right	35.77	0.005	4.27	0.0313	-1.409	0.0313	25.678	4.715	3.457	1.734	0.286	1.259
	43.47	0.005					29.160	5.146	3.405	1.648	0.009	1.204

Table 8: Difference between the measured and the actual planar CG results for z_{shelf} = 10.875in.

	Actual			Difference		
Payload Shelf Location	**W_{act}**	**x_{act}**	**y_{act}**	**ΔW**	**Δx**	**Δy**
	[lb]	**[in]**		**[lb]**	**[in]**	
Center	35.77	0.02	1.841	-4.23	0.26	1.05
	43.47			-4.60	0.37	1.10
Bottom left	35.77	-4.23	-1.41	-5.72	1.31	0.66
	43.47			-6.30	0.81	0.65
Top Left	35.77	-4.23	5.091	-6.16	0.47	-0.89
	43.47			-5.99	0.76	-0.91
Top Right	35.77	4.27	5.091	-1.81	-1.06	1.40
	43.47			-3.75	-1.83	1.53
Bottom Right	35.77	4.27	-1.41	-10.09	-0.81	1.69
	43.47			-14.31	-0.87	1.42

Figure 48 plots the measured weight results versus the location of the test weights relative to the shelf. The plot in Figure 48 shows the measured weight values using a 35.77 lb test weight while the lower plot shows results using a 43.47 lb test weight. A reference line is plotted in both figures so that the measured weight values can be compared to the actual weight. Each point on the plot contains an error bar associated with it. Each error bar has a width that is twice the size of the uncertainty of the measured weight (EB_w), at a particular shelf location and is plotted symmetrically about the corresponding measured weight value.

$$EB_w = 2\sigma_w \tag{44}$$

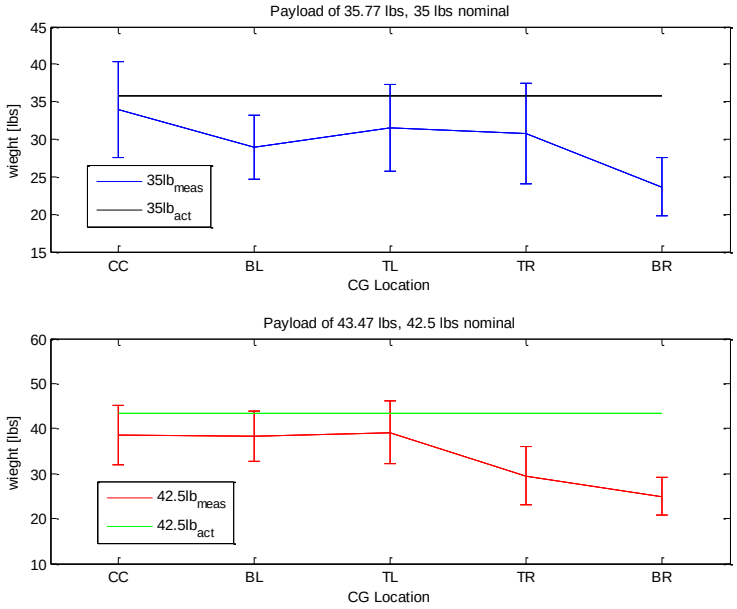

Figure 48: Measured payload weights at various CG locations for 35.77 and 43.47 lb test weights.

The top plot in Figure 49 plots the average of all the x_{meas} values when the payloads are on the left, center or right hand side of the shelf, versus the actual x-coordinates of x_{act} = {-4.23, 0.02, 4.27}in. The left and right most data points are averages of four samples comprised of data generated from two y-locations and two payload magnitudes. An average of their associated propagated uncertainty ($\overline{\sigma_x}$) was combined with the standard deviation of the nominal CG coordinates using equation (27) in section 2.3.2 to create error bars for the left and right most data points. The width of the symmetric errorbars (EB_x) were calculated using equation (45).

$$EB_{x,y} = 2\sqrt{\overline{\sigma_{x,y}}^2 + Sd_{x,y}{}^2} \tag{45}$$

The center data point at x_{act} = 0.02 in is an average of two samples of x_{meas} when the payloads were at the center of the shelf using two different payload magnitudes. A similar process was used to create the lower plot in Figure 49. The average of all the y_{meas} values when the payloads

are on the lower, center and top side of the shelf were plotted versus the actual y-coordinates of y_{act} = {-1.409, 1.841, 5.091}in. Equation (45) was also used to calculated EB_y.

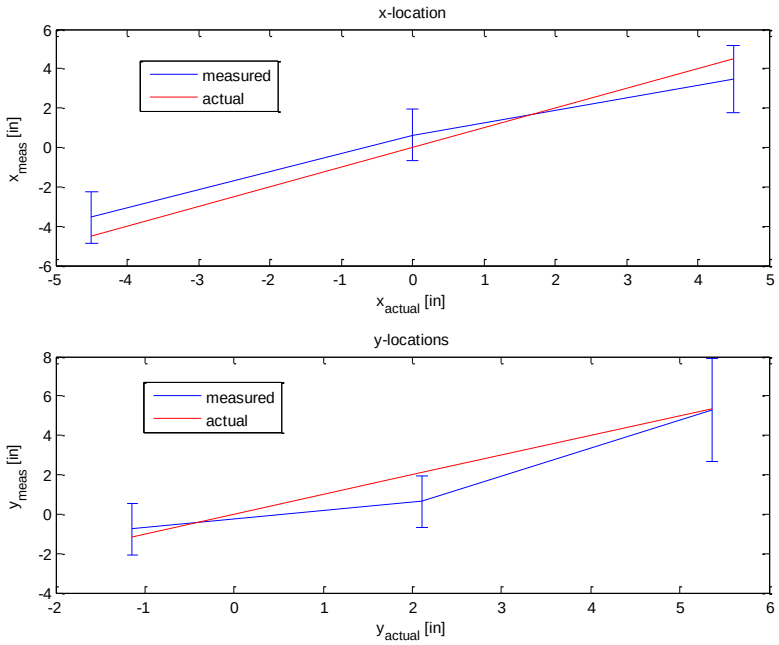

Figure 49: X and Y CG location accuracy plots.

Tabulated results for the measured vertical CG coordinates (z_{meas}) for different test weight locations are displayed in Table 9. The results for z_{meas} at α = 30, 60 and 90 degrees are also presented in the table, all the result where generated with z_{shelf} = 10.875in. The actual vertical CG (z_{act}) changes as more weight is added onto the shelf. It should be noted that z_{meas} is calculated using the bipod strain data at $\alpha \neq 0^o$ and the planar CG coordinates (x_{meas} and y_{meas}) when $\alpha = 0^o$. The uncertainty propagated values of the vertical CG coordinates ($\sigma_{z,\alpha}$) at different α angles are also tabulated in the table below. The difference between the measured and actual vertical CG coordinates for different α angles (Δz_α) are shown on the right of the table.

Table 9: Platform tilting test results for z_{meas} at different planar payload locations and α's at z_{shelf} = 10.875 in.

	Actual			Measured						Difference		
Payload Shelf Location	W_{act}	z_{act}	σ_{zact}	z_{30}	σ_{z30}	z_{60}	σ_{z60}	z_{90}	σ_{z90}	Δz_{30}	Δz_{60}	Δz_{90}
	[lb]	[in]		[in]						[in]		
Center	35.77	11.96	0.035	15.78	5.50	20.65	5.83	20.46	5.19	3.16	8.03	7.81
	43.47	12.72	0.036	14.72	5.20	20.09	5.69	19.71	4.99	2.10	7.47	7.06
Bottom left	35.77	11.96	0.035	17.43	5.31	21.78	5.38	21.55	5.10	4.81	9.16	8.90
	43.47	12.72	0.036	17.40	5.25	21.13	5.15	21.13	4.91	4.78	8.51	8.48
Top Left	35.77	11.96	0.035	15.19	4.40	19.86	4.88	22.43	5.83	2.57	7.24	9.78
	43.47	12.72	0.036	15.63	4.36	21.00	4.91	21.24	5.51	3.01	8.38	8.59
Top Right	35.77	11.96	0.035	26.06	11.46	18.17	6.15	19.90	5.68	13.44	5.55	7.25
	43.47	12.72	0.036	24.93	10.71	18.94	6.39	19.83	5.73	12.31	6.32	7.18
Bottom Right	35.77	11.96	0.035	32.08	13.20	28.24	9.36	24.58	6.19	19.46	15.62	11.93
	43.47	12.72	0.036	33.86	13.83	28.94	9.41	25.73	6.38	21.24	16.32	13.08

The figure below plots 30 samples of the z_{meas} vs z_{act} for all the tested weight locations, α angles and both shelf heights for a 43.47 lb test weight. The purpose of this plot is to show the spread of the z_{meas} results.

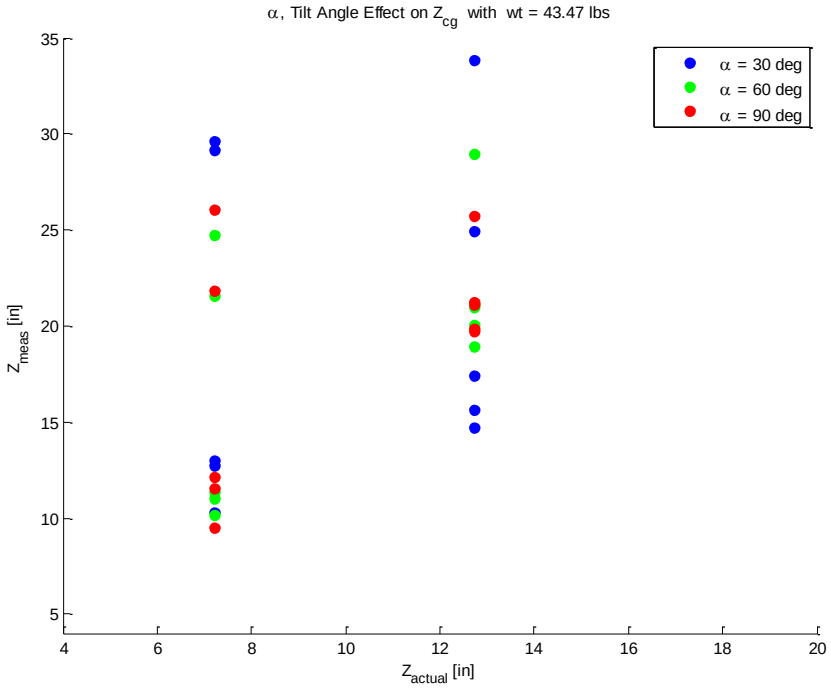

Figure 50: Scatter plot of the z_{meas} versus the z_{act} for different α.

The next two plots in Figure 51 and Figure 52 show α's influence on z_{meas} when a 43.47 lb test weight is used. Figure 51 shows a box plots of z_{meas} for the three tested α values at both z_{act} = 12.72 and 7.22 in. The z_{meas} values for top and lower plots in Figure 51 of was generated using z_{shelf} of 5.5 and 10.875 in respectively. Tabulated results for z_{shelf} = 5.5 in can be found in G. 1 in Appendix G. Figure 52 plot the z_{meas} values for the test weight at different shelf location for varying α's.

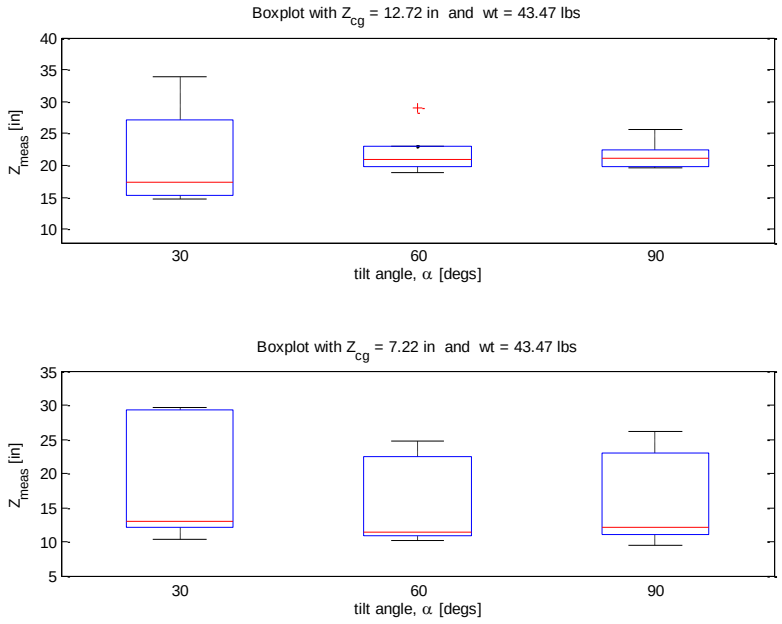

Figure 51: Boxplot of z_{meas} at various tilt angles when z_{act} was is 12.72 in and 7.22 in using a 43.47 lb payload.

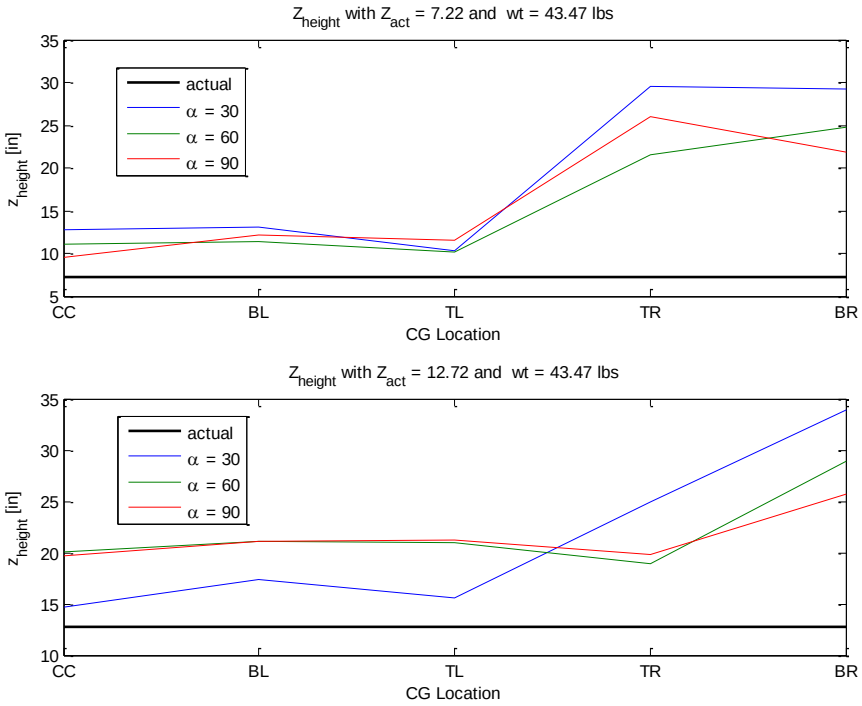

Figure 52: Compares z_{meas} with the payload planar location using a 43.47 lb payload.

6.2 Discussion

After examining the plots, we see that large differences between the actual CG and the measured CG exist. This result was expected since a sizable amount of heuristic error was observed from the bipod calibration tests. However, the errorbars failing to encompass all actual CG and payload values was not expected. It is very likely that another source of error in the experimental set up is not being accounted for, resulting in errorbar widths that do not capture the actual data points.

From Figure 48, all of the measured payload values undershot the actual payload weights. A large drop in the measured weights is observed when the actual payloads are moved to positions on the right side of the shelf. The predicted uncertainty range using a 43.47 lb payload at the center and left half of the shelf encompassed the actual magnitudes for the weight. However, using a lighter 35.77 lb payload, the predicted uncertainty range only includes the actual weight magnitude when the payload was at the center and top half of the shelf.

The largest error in the measured planar CG locations occurred when the payloads were placed on the right half of the shelf; this can be observed by looking at the bottom right (BR) and top right (TR) Δy columns in Table 8. In particular, the y_{meas} values when the test weight was placed in the BR location on the shelf, are nearly zero. These discrepancies in the measured CG of the y-component propagate through the system causing errors in the measured vertical CG coordinate, z_{meas}. This happens because z_{meas} is calculated using the measured planar CG values (x_{meas} and y_{meas}).

With large errors observed in the measured CG components, for payloads placed on the right half of the shelf, one would expect there to be large errors in z_{meas} as well. The net spread of the z_{meas} results for the two tested z_{act} cases, using 43.47 lbs of weight were both large and approximately the same. This trend can be viewed in Figure 50 . Results examining how the tilt

angles (α) influence z_{meas} are plotted in the Box plots in Figure 51. They show that the range of the z_{meas} decreases as α increases if z_{act} is 12.72 in and a test payload of 43.47 lbs is used. Looking at the top plot in the same figure, at $\alpha = 60^{o}$, one outlier appears at z_{meas} = 28.94 in. Using the smaller shelf height and with z_{act} = 7.22 in, the range between the 25^{th} and 75^{th} percentile remains relatively unchanged for different α values. The median lines for z_{meas} in all the Box plots were closer to the 25^{th} percentile range for the plots at 30 and 60 degree tilt angles. This trend was true for both heights. Both of the plots in Figure 52 show a spike in z_{meas} for payloads placed at the top and bottom right corners of the shelf. Generally speaking, z_{meas} always over estimated z_{act}. More results for the platform tests can be found in Appendix G. After examining these plots it is clear that theses experimental results cannot be used to validate the CG math model.

It is very likely that the heuristic errors observed in the bipod rods caused the large differences between the actual and measured CG vector as well as in the measured payload weights. The fact, the discussion at the end of Chapter 5 pointed out that payloads of about 200 lbs would be needed to achieve bipod load errors smaller than 15%. By using loads much smaller than 200 lbs, errors larger than 15% is expected in the CG measurements. A more interesting observation is noticing that the uncertainty ranges for the measured values were not large enough to include the actual values. It is possible that some other sources of error in the experimental set up are not being accounted.

Results from the Platform Loading Tests alone are not enough to validate the mathematical model of the CG measuring platform. If sufficient time was granted to rectify this issue, more Instron calibration tests would be conducted to investigate different endrod assembly error identification hypotheses. If the hidden errors in the endrod assemblies can be identified then the new errors can be incorporated into the CG uncertainty model. This would ultimately widen the width of the uncertainty range include the actual data points.

Another approach is needed to test the MATLAB code's ability to predict the location of a CG vector given the loads in the bipod rods. To investigate whether the code works correctly, an FE model of the CG Platform assembly was created to generate the expected bipod loads at a number of different loading scenarios. The following chapter outlines this task and the results.

7. VALIDATING MATH MODEL USING FEA

Since the results from the Platform Tilting Test produced CG uncertainty ranges that did not encompass the actual CG locations entirely, another independent method for validating the CG math model was needed. The Finite Element Method (FEM) was selected to perform this task for its ability to obtain numerical solutions for stress analysis problems. The FE method works by first discretizing a body into an equivalent system of smaller finite elements, also referred to as a mesh. Each element is assigned a displacement function along with compatibility strain/displacement laws and constitutive stress/strain relationships. Next, the elemental stiffness matrices can then be generated and used to calculate the strain and stress in the elements. More detail on the FEM theory can be found in [19].

FEM can be used as a means to validate the CG model because it directly expresses the strain and stress quantities in terms of the generalized displacements. These displacements can be calculated using iterative methods such as Gauss-Seidel or Gaussian matrix elimination. The method implement into the CG code for measuring CG uses a force or flexibility method to determine the internal forces in the bipod rods [19].

In this chapter, the process of developing and using a simplified FE model is described. The simplified model was created in ABAQUS 6.12 rather than exporting the complete CAD model from SOLIDWORKS. Once the Platform model was created, a point mass was placed at various locations relative to the BCS of the platform. Care was taken to assure that point mass locations matched locations inputted to the CG code. Simulations for a few cases were run to generate the expected loads in each bipod rod.

To validate the model, the bipod loads from ABAQUS were compared to the loads generated from the CG code set to solve the Reverse Method. The load values generated from

ABAQUS were aslo as inputs to the CG code to compute the CG using loads from ABAQUS. The same CG parameters that were used for the code were used to construct the Platform assembly in ABAQUS. The following sections will review how the FE model was created and how well it substantiates the CG model.

7.1 FE Model Construction

Only the components that have the most influence in the prediction of the CG needed to be modeled in ABAQUS:

- The bipod rods along with their load path intersection points
- Adaptor Plate of the platform
- The payload

The Adaptor Plate was modeled as a 3-D shell object and its original hexagonal geometry was simplified to a rectangular shape. A shell type element was chosen since the loading experienced by the plate was not the main focus. In particular, the shell element works best for structures in which the stress in the thickness direction is negligible or when one dimension is significantly smaller than the others [14]. Ideally, the Adaptor Plate should be as rigid as possible so that the most significant areas of stress occur in the bipod rods. The Elastic Modulus and density of 10E6 lb/in^2 and 0.098 lb/in^3 for aluminum was used for the plate in this FE model. A total of 288 linear quadrilateral elements were used to mesh the plate. A more refined mesh of the platform was not needed since stress within the platform was not the focus. The Adaptor Plate was portioned so that the bipod rods could be assembled to the proper location on the plate.

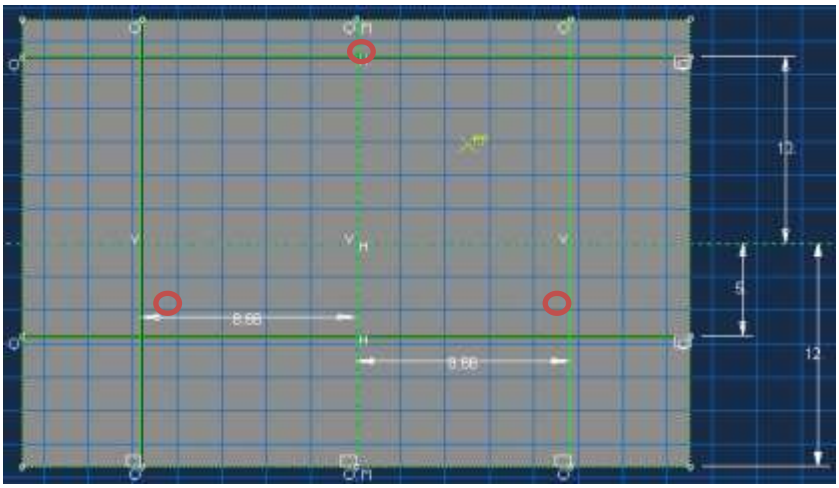

Figure 53: Top view of portioned Adaptor Plate.

The bipod rods were modeled as 1-D wire objects containing only one element. This was done to ensure that the rods would only resist axial loads. A truss section was assigned to the bipod rods to provide axial strength but no bending stiffness [15]. This is intended to model the combination of brackets and endrods in the physical system. For the bipod rod FE model, a cross sectional area of 0.0644 in^2 was used as well as the Elastic Modulus and density of aluminum. The CG code was modified to assign the same cross sectional area to all the bipod rods, to be compatible with the FE model. Each bipod rod spans the distance between the bottom nodes on the brackets bolted to the base plate of the platform to the load path intersection point of their respective bipods.

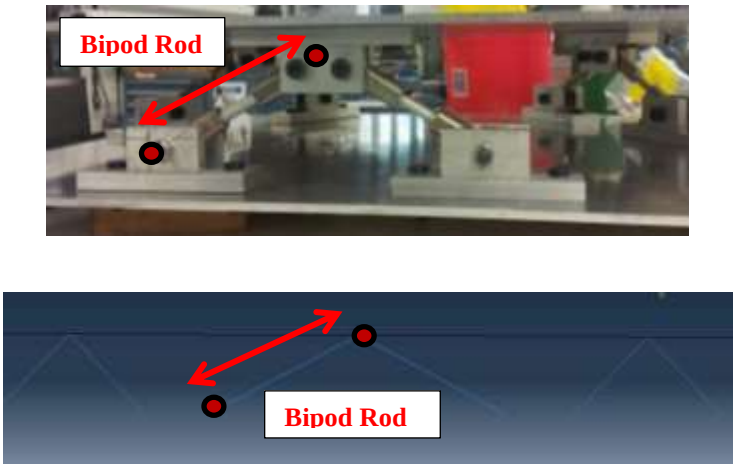

Figure 54: (Top) Bipod rod in physical model. (Bottom) FE model.

A tie constraint was used to constrain the upper nodes of the two bipod rods in a bipod set to the load path intersection point beneath the Adaptor Plate. Displacement boundary conditions were placed on the lower nodes of each bipod rod to model the revolute joints of the physical model.

The point mass of a payload was created as an inertial mass at a reference point above the platform. A payload weight of 43.47 lb and a z_{cg} of 12.91 in was used for all the simulations in ABAQUS. Different CG locations were tested by modifying the location of the reference point. To mimic the rigidity of the CG Structure, the reference point of the point mass was constrained to the Adaptor Plate using a coupling kinematic constraint. This feature constraints the motion of the point to the surface of the plate. Figure 55 shows the point mass constrained to the Adaptor Plate.

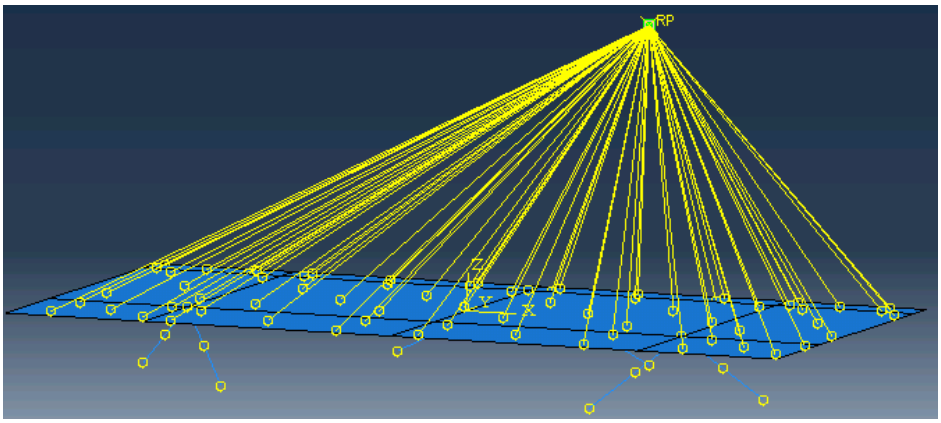

Figure 55: Simplified CG Platform Assembly in ABAQUS.

7.2 FE Model and CG Math Model Comparison

Five different loading cases were simulated in the FE model to generate the loads in each bipod rod. This process is equivalent to the Forward Method in the CG code which calculate the bipod rod load given the CG vector and platform tilt angle (α). Table 10 contains a list of all the simulated cases, these cases where chosen randomly. The image in Figure 56 shows a separate simulate so illustrate the Adaptor Plats and bipod rod deflections when $\alpha = 90^0$. (Note: - α simply refers a platform tilt angle who's rotation vector is point in the negative y-direction.) The bipod loads from the FE model were then used as inputs to the CG code set to solve the Forward Method to calculate the CG vector. The CG vector using the FEM loads were then compared to the expected CG or the point mass location used in ABAQUS.

Table 10: Simulated cases for FE model.

Case Number	Payload Shelf Location	Tilt Angle (α)	Platform Rotation (β)
		[Degs]	
1	Bottom Left	0	0
2	Bottom Left	-60	0
3	Top Right	0	0
4	Top Right	30	0
5	Top Right	-90	60

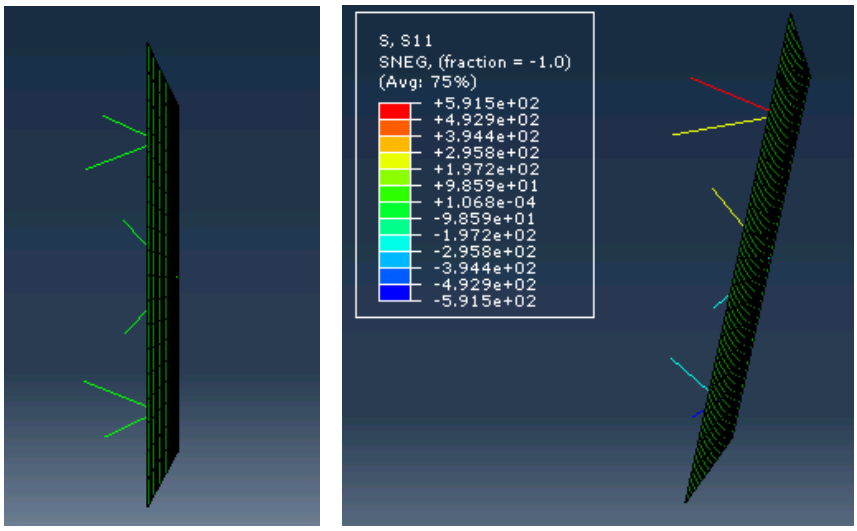

Figure 56: (Left) Platform Tilted at α = 90°. (Right) Simulation Contour of Platform at α = 90° and a payload weight of 43.47 lbs at BR.

Results from the FEA simulations generated bipod loads that matched extremely well with the loads calculated from the CG Code while it was set in the Reverse Direction. Table 11 shows that the percent difference between the bipod loads using the two methods are on the order of 0.01%. This was true for case 5 when α and β were set to -90 and 60 degrees respectively with the payload placed at the top right (TR) location of the shelf. These parameter we selected at random. Using the bipod rod loads generated from the FE model as inputs into the CG Code produced a measured CG and payload that were very close point mass location and weight used in ABAQUS.

Table 12 contains the results from using the bipod loads generated from the FE model as inputs to run the Forward Direction in the CG Code. Similar to the Forward comparison, the results show that both the predicted CG vector and the payload are approximately 0.01% different from the actual CG and payload.

Table 11: ABAQUS and MATLAB Bipod Rod load comparison with a payload on the top right corner of the shelf (TR).

	Reverse Comparison (TR)			
	σ_{ABAQUS}	F_{FEM}	F_{MATLAB}	Diff
Bipod Rod Index	**[lb/in^2]**	**[lb]**	**[lb]**	**[%]**
1	-559.153	-36.009	-36.012	-0.006
2	-317.935	-20.475	-20.476	-0.002
3	317.928	20.475	20.476	-0.008
4	559.146	36.009	36.012	-0.009
5	291.37	18.764	18.765	-0.003
6	-291.357	-18.763	-18.766	-0.014
α = -90 , β = 60				

Table 12: Forward results comparison between the actual values and the results using loads generated by FEM as inputs to the CG code. Payload set to the top right corner of the shelf (TR).

	Forward Comparison (TR)			
α = -90 , β = 60	**x**	**Y**	**z**	**mg**
Units	**[in]**	**[in]**	**[in]**	**[lbs]**
Actual Values	4.50	5.36	12.91	43.470
Using FEM	4.501	5.360	12.911	43.466
Percent Diff	-0.011	0.004	-0.001	0.009

Additional simulations with the FEM were performed with the payload at the bottom left (BL) shelf setting to confirm the small percent difference between the FEM and CG Code. Similar results were observed and the tables of results from these comparisons can be found in

Appendix H. The agreement of the CG Code with the FEM suggests that the derived Math model for the CG measuring platform is indeed valid

7.2.1 Platform Loading Results Revisited

The Reverse direction of the CG Code was then used to generate the expected bipod rod loads and their corresponding strains to determine how far off the measured experimental strains are from their expected values. Figure 57-58 and Table 13-13 compare the results of both of these cases when the actual payload of 43.47 lbs is on the BR corner of the platform and the z_{act} is 12.72in. This payload and CG location was selected for the comparison here since the measured values at this particular position showed the largest difference from the actual CG and the expected. CG. The corrected measured strains ($\varepsilon_{corrected}$) refers to the measures strain from the Platform Loading Tests which has been corrected for nonlinearity errors. The expected strains refer to the strain values that were generated using the CG verified MATLAB code.

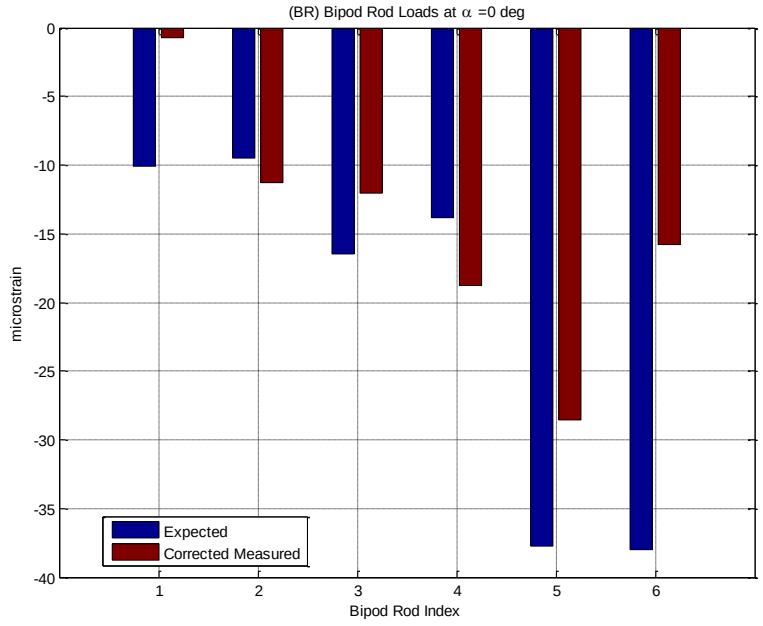

Figure 57: Measured and expected bipod rod strains at $\alpha = 0^{o}$ while the actual payload is at the BR corner of the shelf.

Table 13: Load and strain results for the expected and measured cases. Parameters: $\alpha = 0^o$, Payload at BR and $z_{act} = 12.72$ in.

BPR Index	F_{expect}	F_{meas}	F_{DIFF}	ε_{expect}	ε_{CM}	ε_{Diff}
	[lb]			[με]		
1	-6.51	-0.48	6.02	-10.10	-0.75	9.35
2	-6.51	-7.75	-1.24	-9.47	-11.28	-1.81
3	-9.88	-7.21	2.68	-16.50	-12.03	4.47
4	-9.57	-13.04	-3.47	-13.80	-18.80	-5.00
5	-23.67	-17.92	5.75	-37.74	-28.57	9.17
6	-24.51	-10.18	14.32	-37.99	-15.79	22.20
					$\|\|\varepsilon_{DIFF}\|\|$	52.00

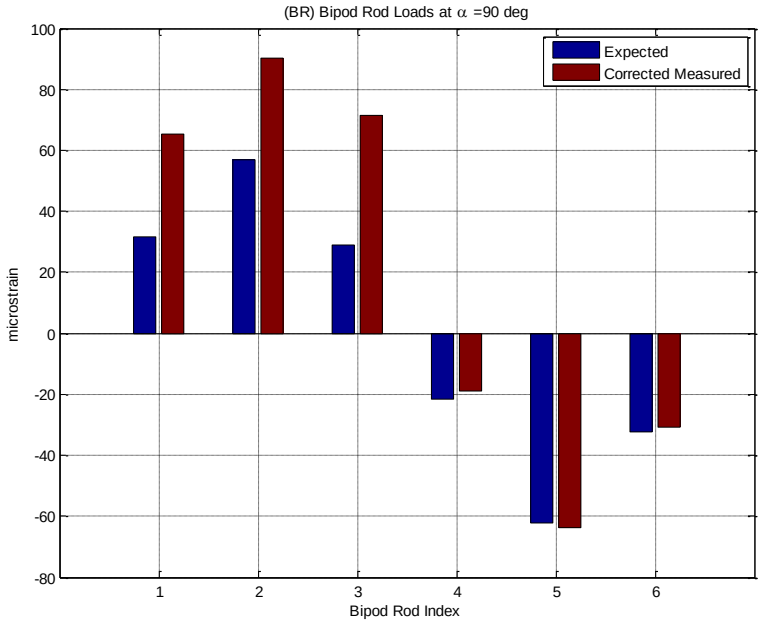

Figure 58: Measured and expected bipod rod strains at $\alpha = 90^o$ while the actual payload is at the BR corner of the shelf.

Table 14: Load and strain results for the expected and measured cases. Parameters: $\alpha = 90^o$, Payload at BR and $z_{act} = 12.72$ in.

BPR Index	F_{expect}	F_{meas}	F_{DIFF}	ε_{expect}	$\varepsilon_{cor,meas}$	ε_{Diff}
	[lb]			[με]		
1	20.47	42.13	21.66	31.79	65.42	33.63
2	39.02	61.99	22.97	56.80	90.24	33.44
3	17.37	42.79	25.42	28.99	71.44	42.44
4	-15.02	-13.04	1.98	-21.65	-18.80	2.85
5	-38.91	-40.08	-1.17	-62.04	-63.91	-1.87
6	-20.87	-19.88	0.99	-32.36	-30.83	1.53
					$\|\|\varepsilon_{DIFF}\|\|$	115.76

Looking at Figure 57 and Table 13, the results for the payload at BR with $\alpha = 0^o$, it can be noticed that the largest difference between the measured and expected strains occur in BPR 6. However, when the platform is tilted to 90^o the largest difference appears in BPR 3 while the smallest difference is at BPR 6. The results also show that when the platform is flat with the horizontal BPR 1 registers very little strain, in fact the strain is practically zero.

Although a sizable difference is observed between the measured and expected loads in the bipod rods, attention must not be everted from the fact that the CG model used to write the MATLAB code agrees with the FEM model in ABAQUS. Instead, the challenges experienced in the tests conducted for this project can be used to support the premise that better transducers will be needed to accurately measure CG. Using strain gauges to perform this task proved to be more difficult than expected. Now knowing that the CG model is indeed correct, it can be hypothesized that the observed errors in the Platform Tilting Test can either stem from the endrod assemblies and there strain gauges or the experimental set up. Judging from the original heuristic errors witnessed from the calibrations of the bipod rod in Chapter 5, it seems more likely that the measured CG errors come from the endrod assemblies.

8. CONCLUSIONS AND RECOMMENTATIONS

The focus of this thesis was to derive a mathematical model of a center of gravity measuring platform and to confirm its validity. This project was proposed by Jet Propulsion Laboratory (JPL). JPL requested that the center of gravity math model be implemented into a MATLAB code. The code would be utilized to perform quick center of gravity sanity checks for Integration Engineers while assembling satellites or other spacecraft. Measurement uncertainty was asked to be included in the model as it would provide a range in which the true center of gravity and payload values would reside. JPL expressed a need for the code to also function in the Reverse direction to back calculate the expected bipod loads given a CG vector and payload magnitude. The aim in doing this was to provide a tool that would assist Engineers in selecting load cells with sufficient capacity and error ranges to design their own CG Platform. A scaled down prototype of a CG Platform using relatively inexpensive strain gauge transducers was designed to conduct a series of loading experiments to test the validity of the math model and MATLAB code. In addition to the experimental tests, an Finite Element model was developed to serve as a secondary tool for validating the CG code.

8.1 Conclusions

Empirical results from the platform loading experiment were not accurate enough to validate the CG code, and as a result, the FE model became the primary source for validating the CG code. Looking back at section 5.2.1, results from the Bipod Rod Calibration Tests showed that heuristic errors observed in the rods were as large as 23.2% if load magnitudes larger than 10 lbs were applied to the endrod assemblies. With these error ranges in the bipod rods, it is easy to see why the nominal values of the measured CG and payload values from the Platform Loading Tests did not match well with the expected values.

The individual uncertainty of the measured parameters needed to model the CG measuring platform were recorded and used to develop a measurement uncertainty model using the uncertainty propagation principle. Combining the measurement uncertainty with the observed heuristic errors led to a wider uncertainty range for each measured/calculated CG and payload values.

It was somewhat surprising to see that only a few of the results from the measured CG and payload values possessed uncertainty ranges that were wide enough to encompass their actual CG and payload magnitude counterparts. The results from the Platform Loading Test in section 6.2 and Appendix G show that an inherent error exist in the experimental set up. Again, this result was traced back to the initial errors observed in the bipod rods after the CG math model was validated from the FEM. Results from the FEM and CG Model comparison show that there is less than a 0.01% difference between the loads predicted from the CG code to the loads generated from the FE model. These results accomplish the original goal to validate the math model for the CG measuring platform. Using the bipod loads from the FE model as inputs the CG code resulted in calculated CG vectors that matched the point mass coordinates used to create a payload in the FE model. The difference between these two results was also with a 0.01% difference.

8.2 Recommendations

Based on the conclusions presented, the following recommendations should be considered for future work:

- Run another set of tests to experimentally validate the CG model using precision load cells rather than strain gauges.

- A wider variety of payload locations should be tested as the tests conducted for this project was confined to a 17 3/8 x 15 7/8 x 26 3/8 in^3 volume. A wider range of payload magnitude should also be used.
- If strain gauges are to be used again as the transducers in a future CG platform, further bipod rod calibration loading tests should be conducted to identify the additional sources of error in the endrod assemblies.
- Modify the Code to receive the top bracket node locations to calculate R_{bi}. The current code uses the location of the bottom bracket nodes and ϑ_i to determine R_{bi}.
- Investigate the Singular Value Decomposition (SVD) method as a way to derive a more robust CG model to eliminate the need to have bipod load measurements at two α angles. The SVD method has the potential to extract CG solution vector using only one α.

REFERENCES

[1] F. P. Beer, Vector Mechanics for Engineers, New York: McGraw Hill, 2010.

[2] J. Elfick, "School Science Lessons," [Online]. Available: http://www.uq.edu.au/_School_Science_Lessons/8.169.GIF. [Accessed 2 June 2015].

[3] C. Winkler, "Center of Gravity Height: A Round-Robin Measurement Program," The Motor Vehicle Manufacturers Association, Detroit, 1991.

[4] R. Boynton, "Measuring weight and all three axes of the center of gravity of a rocket motor without having to re-position the motor," Space Electronic, Inc., Berlin, 2002.

[5] D. H. T. Castrup, Measurement Uncertainty Analysis Principles and Methods, Washington: NASA, 2010.

[6] B. Frandeen, "Statement of Work for Cal Poly San Luis Obispo ME Dept, Load Cell Platform," JPL, 2014.

[7] B. Ridder, "Propagation Uncertainty Matlab Function," 2007.

[8] S.-W. Seong, "Bitmask-Based Code Compression for Embedded Systems," *IEEE Transaction o Computer-Aided Design of Integrated Circuits and Systems ,* vol. 27, pp. 673-685, 2008.

[9] D. J. Ridgely, "Quad Bridge Amplifiers," 2014.

[10] Micro-Measurements, "Errors Due to Wheatstone Bridge Nonlinearity," Vishay Percision Group, 2010.

[11] ASM Aerospace Specification Metal Inc., "Aluminum 6061-T6," 2001. [Online].

[12] Micro-Measurements, "Errors Due to Misalignment of Strain Gages," Vishay Percision Group, 2010.

[13] Micro-Measurements, "Shunt Calibration of Strain Gage Instrumentation," Vishay Percision Group, 2013.

[14] Dassault Systemes, "Shell Features 11.9.2," in *Abaqus 6.12/CAE User's Manual*, 2012.

[15] Dassault Systemes, "Defining Sections 12.2.3," in *Abaqus 6.12/CAE User's Manual*, 2012.

[16] Micro-Measurements, "Strain Gage Rosettes: Selection, Application and Data Reduction," Vishay Precision Group, 2014.

[17] D. Price, "Estimation of Uncertain Vehicle Center of Gravity using Polynomial Chaos

Expansion," Virginia Polytechnic Institute and State University, Blacksburg, 2008.

[18] V. Micro-Measurements, "Model P3 Strain Indicator and Recorder," Raleigh, 2005.

[19] R. G. Budynas, "Welding, Bonding, and the Design of Permanent Joints: Chpt 9," in *Shigley's Mechanical Engineering Desgin*, McGraw Hill, 2011.

APPENDICES

APPENDIX A. HALF BRIDGE VOLTAGE TO STRAIN DERIVATION

ρ is the resistivity and R is the resistance of the wire

$$R = \frac{\rho L}{A}$$

A is a function of some constant C times the diameter squared

$$A = \frac{\pi}{4} * D^2 \ = CD^2$$

Taking the derivative of the resistance with $R = \frac{\rho L}{CD^2}$

$$dR = \frac{CD^2(\rho * dL + Ld\rho) - \ \rho L(2CD * dD)}{(CD^2)^2}$$

$$\frac{dR}{R} = \frac{dL}{L} - 2 * \frac{dD}{D} + \frac{d\rho}{\rho}$$

Divide by $\frac{\frac{dR}{R}}{\frac{dL}{L}}$

$$\frac{\frac{dR}{R}}{\frac{dL}{L}} = 1 - 2\frac{\left(\frac{dD}{D}\right)}{\frac{dL}{L}} + \frac{\frac{d\rho}{\rho}}{\frac{dL}{L}}$$

Where the following is true

$$\frac{dL}{L} = \varepsilon_a \qquad \frac{dD}{D} = \varepsilon_L \qquad \frac{\frac{dD}{D}}{\frac{dL}{L}} = -\nu$$

Making substitutions

$$\frac{\frac{dR}{R}}{\varepsilon_a} = 1 + 2\nu + \frac{\frac{d\rho}{\rho}}{\frac{dL}{L}}$$

Where GF is the Gauge factor for the strain gauge

$$F_g = \frac{\frac{dR}{R}}{\varepsilon_a}$$

$$\varepsilon_a = \frac{1}{GF}\frac{\Delta R}{R}$$

It is easier to measure changes in voltage than it is to measure changes in resistances

$$\frac{\Delta V}{V_{exc}} = \frac{V_{out}}{V_{exc}} = \frac{K}{4}\frac{\Delta R}{R} = \frac{KGF\varepsilon_a}{4}$$

Where K is the bridge factor and depends on the amount of gauges used in the system and there orientation (k=1, 2 , [1+v] or 4)

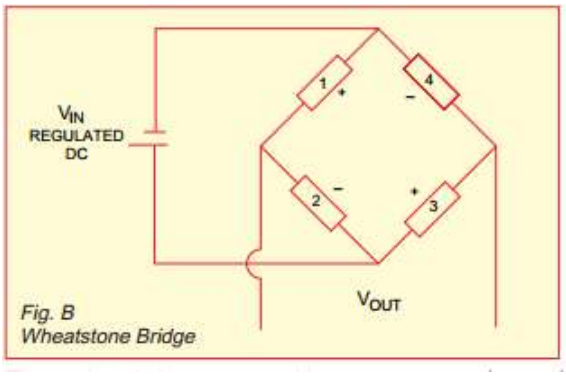

Figure 59: Wheatstone bridge diagram.

$$\varepsilon_a = \frac{4(V_i/V_{exp})}{\mathrm{K}(GF)}$$

For a half bridge arrangement to measure axial load, the bridge is equal to one plus the Possions ratio of the material.

$$\varepsilon_a = \frac{4(V_i/V_{exp})}{(1+\nu)(GF)}$$

The strain equation can then be rearranged to determine the voltage for the applied strain:

$$V_i = \frac{(1+\nu)}{4} GF * \varepsilon_i * V_{exc}$$

APPENDIX B. STRAIN GAUGE SELECTION CALCULATIONS

B. 1 Minimum and Maximum Strain Calcs

The maximum and minimum strains are calculated to see how large of a strain produce from the maximum and minimum payload constraint. Equation (33) is used to evaluate both strains

Given Values:
W_{max} = 150 lbs
W_{min} = 10 lbs
$\theta = 35^{o}$

$$\varepsilon_{max} = \frac{150/6\ lbs}{sin35 * (0.25 * 0.25)in^2 * 10E6lb/in^2} = 464.9\mu\varepsilon$$

$$\varepsilon_{min} = \frac{10/6\ lbs}{sin35 * (0.25 * 0.25)in^2 * 10E6lb/in^2} = 46.5\mu\varepsilon$$

The minimum voltage can be calculated using the voltage to strain equation derived in Appendix A.

Given Values:

K = 1.33	bridge factor
GF =2.095	Gauge factor of the strain gauge
V_{exc}= 3.3v	Excitation voltage of Wheatstone bridge

$$V_{min} = \frac{(1+\nu)}{4} GF * \varepsilon_{min} * V_{exc}$$

$$V_{min} = \frac{1.33}{4} * 2.092 * 46.5\mu\varepsilon * 3.3v$$

$$V_{min} = 0.051\ mV$$

B. 2 Load Limit for DAQ System

The smallest measurable load that can be applied to the endrod can be found be setting the voltage resolution of the DAQ System equal to the voltage expression for the strain gauge.

From equation (33) the strain in the endrod is all follows

$$\varepsilon_{limit} = \frac{F_{lim}}{EA}$$

Next the strain can be plugged into the voltage equation and set equal the voltage resolution

$$V_{resolution} = \frac{(1+\nu)}{4} GF\left(\frac{F_{lim}}{EA}\right) * V_{exc} * G_{amp}$$

Now the F_{lim} can be isolated and evaluated

$$F_{lim} = \frac{4}{GF(1+\nu)\mathrm{G_{amp}}}\left(\frac{V_{resolution}}{V_{exc}}\right) * EA$$

APPENDIX C. PROPOSED ROBUST METHODS

C. 1 Least Squares Method and Normal Equation

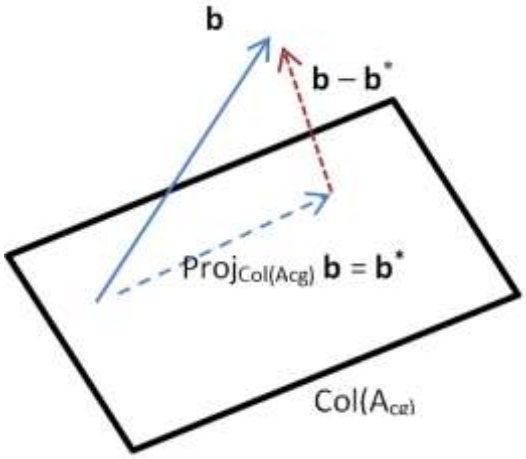

Figure 60: Illustration showing the orthogonal projection of b into the column space of A, $\mathbf{Proj_{col(Acg)}b = b^*}$.

$$\|A\boldsymbol{x}^* - \boldsymbol{b}\| \leq \|A\boldsymbol{x} - \boldsymbol{b}\|$$

$$A^T A\boldsymbol{x}^* = A^T\boldsymbol{b}$$

C. 2 Singular Value Decomposition

A singular matrix A

$$A = U\Sigma V^T$$

Where V is filled with the eigenvectors of A^TA, U is filled with the eigenvectors of AA^T and Σ is a diagonal matrix of the eigenvalues of A^TA.

APPENDIX D. MATLAB CODE

D. 1 Main CG Code

```
%% Main CG Code
%% ++++++++++++++++++++++++++++++++ %%
%%%==================================%%%
%%% CENTER OF GRAVITY PLATFORM CODE %%%
%%%==================================%%%
%%%++++++++++++++++++++++++++++++++++%%%

clc;clear all

% prompt user to select Measuring CG or Sizing Load Cell path
opt = '(1)Measure CG or (2)Size Load Cells. \n Enter either (1)
or (2) \n\n';
useropt = input(opt);


switch useropt
  case 1 % Measure CG
    disp('Calculate CG')
    disp('Is the LoadData.xlsx file updated?')
    disp('Press Enter if file is updated')
    disp(' ')
    pause

    opt2 = '(1)Find planar payload location: CG[x;y] or
(2)complete payload location: CG[x;y;z] \n Enter either (1) or
(2) \n\n';
    useropt2 = input(opt2);

    % evaluate payload location
    [cg,dcg,mg,dmg] = Force2cgShell(useropt2);

  case 2 % Size Load Cells
    disp('Size Load Cells')
    % Absolut readings or tare readings
    opt = ' Enter (1) for Tare measurements or (return) for
Absolute measurements of one test case \n Enter either (1) or
return\n';
    option = input(opt);

        if option == 1
            [F,Fcap,Fabs1,Fcap1,F_at,Fatcap] =
SizeLoadCells(option);
        else
            [F,Fcap,~,~,~,~] = SizeLoadCells(option);
```

```
        end

  otherwise
    warning('invalid input; Re-run and enter either (1) or (2)')
end

beep
pause(1)
beep
```

D. 2 Force2cgShell

```
function [cg,dcg,mg,dmg] = Force2cgShell(opt)
%% User Requested Variables
format
%%
% clc;clear all
% opt = 1
%% Inputs
filename = char('LoadData');     % load data from excel workbook,
filename indicates shelf height
LCindx0  = char('A2:D8');        % data range for platform at 0deg
tilt
LCindx1  = char('B12:D17');      % data range for platform at 0deg
tilt

% Store load cell data at alpha = 0 from excel to MATLAB
variables
LCData = xlsread(filename,1,LCindx0);
F0 = LCData(:,2);       % unloaded load cell data at alpha = 0
Fload = LCData(:,3);    % loaded load cell data at alpha = 0

% Store load cell data at alpha from excel to MATLAB variables
LCData1 = xlsread(filename,1,LCindx1);
F0_p = LCData1(:,1);              % unloaded load cell data at
alpha
Fload_p = LCData1(:,2);           % loaded load cell data at
alpha
alph = xlsread(filename,1,'B10'); % platform tilt angle
beta1 = xlsread(filename,1,'D10'); % platform rotation angle
%% Load cell error
dF_all = xlsread('CG_platform_constants',2,'B15:B20');

% error in force readings at alpha = 0
   dF0 = F0.*dF_all/100;
dFload = Fload.*dF_all/100;

% error in force readings at alpha ~= 0
   dF0_p = F0_p.*dF_all/100;
dFload_p = Fload_p.*dF_all/100;

%% Find CG
Fload = Fload - F0;
Fload_p = Fload_p - F0_p;

if  opt == 1
```

```
% calculate CG coordinates: cg_ang =[x,y,z], dcg_ang =[dx,dy,dz]
and mg_ang = [wt_0deg, wt_ang]
    [cg,dcg,mg,dmg] = force2cg(Fload,[],[],dFload,[],0);
elseif opt == 2
% calculate CG coordinates: cg_ang =[x,y,z], dcg_ang =[dx,dy,dz]
and mg_ang = [wt_0deg, wt_ang]
    [cg,dcg,mg,dmg] =
force2cg(Fload,Fload_p,alph,dF0,dF0_p,beta1);
else
    warning('Enter either (1)for CG[x;y] at alpha = 0 or (2)for
CG[x;y;z] at alpha ~= 0')
end
% end
```

D. 3 Force2cg

```
function [cg_val,dcg,mg_mag,dmg] =
force2cg(F0vec,F1vec,ang,dF0,dFi,thetaz_deg)
% -
['filename.xlsx','filename.xlsx',tiltangle,uncertainty,platform
rotation] -> [CG,wt,uncertainty]
% - The first file is load data at zero tilt (file0) and the
second file is load data
%   at the specified tilt angle (file1). The code will store the
CG coordinates in a
%   matrix or array.
% - dF0 and dFi are the uncertainty for the loads at 0 degs and
alpha repectively
% - ang is the tilt angle and thetaz_deg is the rotation angle,
both in
%   degree
%
% Example:
% [cg_val,dcg,mg_mag,dmg] =
force2cg(F_0deg.xlsx,F_60deg.xlsx,55,dF0,dFi,258)

%% Load inputs %%

% check for empty inputs, and empty input results in a 1 value.
Full input
% result in a 0 value
inputchk = [isempty(F0vec),isempty(F1vec),isempty(ang),...
            isempty(dF0),isempty(dFi),isempty(thetaz_deg)];

if  inputchk == [0 0 0 0 0 0]    % All inputs are accounted for
    F_measure(:,:,1) = F0vec;
    F_measure(:,:,2) = F1vec;
elseif inputchk == [0 0 0 0 0 1] % either F0,F1,ang and F_err are
known with empty beta
    F_measure(:,:,1) = F0vec;
    F_measure(:,:,2) = F1vec;
    thetaz_deg = 0;              % assumes
elseif inputchk == [0 0 1 0 0 1] % if only data at 0 deg tilt is
provided and or rotation angle
    error(message('Include platform tilt angle, alpha'))
elseif inputchk == [0 1 1 0 1 0] % Force at 0 tilt angle, load
errors and beta are non empty
    F_measure(:,:,1) = F0vec;    % data at 0 deg tilt angle
    F_measure(:,:,2) = zeros(size(F_measure(:,:,1))); % zero out
entries for load at tilt angle
    ang = 0;
    %warning('With three inputs, the tilt angle is assumed to be
0 degs')
else
```

```
    error(message('Invalid inputs, check excel file and make sure
the correct cell have values. Save File'))
end

%% Constants %%
% load angle constants and there uncertainty
[theta_deg,thetaz_deg,~,thetazb_deg,~,~,~,~,~,~,~,~,~,dtheta_deg,
~] = LoadConstants;

% bipod rod angle measure from the horizontal 6 angles for 6
bipod rods
% theta is [1,6]
theta = theta_deg*pi/180;                % angle in [rads]
dtheta = dtheta_deg(1)*ones(1,6)*pi/180; % 6 dtheta's for the 6
theta's

% bipod offset angle (see derivation for image)
thetazb = thetazb_deg*pi/180;  % angle in [rads]
dthetazb = dtheta_deg(4)*pi/180;

% Platform rotation about z-axis (vertical rotation) angle
thetaz = thetaz_deg*pi/180;    % angle in [rads]
dthetaz = dtheta_deg(3)*pi/180;

% Platform tilt angle. Rotation about the y-axis
thetay_deg = [0 ang];          % angle in [degs]
 thetay = thetay_deg*pi/180;    % angle in [rads]
dthetay = dtheta_deg(2)*pi/180;

% angle array for wt and cg calculations
angles0 = [theta thetay(1) thetaz thetazb];
angles1 = [theta thetay(2) thetaz thetazb];
dangles = [dtheta dthetay dthetaz dthetazb];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% load line intersection point for each bipod
[r1, r2, r3, dr1, dr2, dr3] = bipodNodes;

%% Calculations
syms xcg ycg zcg;
cg = [xcg; ycg; zcg];

for j=1:size(F_measure,2)       % weight location index
    for m = 1:size(F_measure,3) % index between 0 deg tilt and
some arbitrary angle
        F(:,j,m) =
[F_measure(1,j,m);F_measure(2,j,m);F_measure(3,j,m);...

F_measure(4,j,m);F_measure(5,j,m);F_measure(6,j,m)];

%% calculate wt magnitude and cg coordinate
```

```
        [mg_mag(1,j),dmg(1,j)] = WeightMagUnc([F(:,j,1)',
theta],[dF0(:,1,j)', dtheta]);
        [cgx(j),xerr(j),cgy(1,j),yerr(1,j)] =
unc4cg(F(:,j,1),dF0(:,1,j),mg_mag(1,j)',dmg(1,j)',...

angles0,dangles,r1,r2,r3,dr1,dr2,dr3);
    end
end
com = double([ mg_mag.' dmg.' cgx.' xerr.' cgy.' yerr.']);

% only data with 0 tilt is available
if (inputchk == [0 1 1 0 1 0])
%% Store Calculated X-Y coordinates into COM array to be
displayed
    cg_val = [cgx' cgy']; % x and y cg coordinates for output
       dcg = [xerr' yerr']; % x and y cg uncertainty for output
    disp(['  mg      dmg        x_cg     dxcg       y_cg     dycg
|@ thetay = ',num2str(ang),' & @ thetaz =
',num2str(thetaz_deg),'|']);
    disp(['-----------------------------------------------------
-']);
    disp([num2str(com,3)]);

% if load data and error at alpha = 0 & alpha for an empty beta
and user defined beta
elseif (inputchk == [0 0 0 0 0 0]) | (inputchk == [0 0 0 0 0 1])
% all 4 inputs are available
    ones_row=ones(1,size(com',2));
    com = [com';zcg*ones_row];

    for j=1:size(F_measure,2)    % weight location index
    %% calculate vertical cg
   [~,~,~,~,cgz(j),zerr(j)] =
unc4cg(F(:,j,2),dFi(:,1,j),mg_mag(1,j)',dmg(1,j)',angles1,dangles
,...

r1,r2,r3,dr1,dr2,dr3,cgx(j),xerr(j),cgy(1,j),yerr(1,j));
    end
    %% Display CG Location
    disp(['      mg        dmg      x_cg     dxcg      y_cg
dycg        z_cg     dzcg   |@ thetay = ',num2str(ang),' & @
thetaz = ',num2str(thetaz_deg),'|']);
    disp(['-----------------------------------------------------
-------------------------']);
    xx = double([com(1:6,:); cgz; zerr])';
    disp(xx);

    cg_val = [cgx' cgy' cgz']; % x and y cg coordinates for
output
```

```
        dcg = [xerr' yerr' zerr']; % x and y cg uncertainty for
output
end
end
```

D. 4 LoadConstants

```
function [theta, thetaz, thetay, thetazb, c1, c2, c3, phi1, phi2,
phi3,...
          rb2, rb4, rb6, dang, dr] = LoadConstants
% Extracts all the inputs from the CG_platform_constants.xlsx
excel file.
% The excel file must first be saved for any changes made in the
document
% to be loaded into MATLAB.
%
%  Procedure
%      1) change inputs in excel
%      2) save excel document
%      3) Run matlab scripte
%
%  Note:
% The first sheet are the constant parameters while the second
sheet are
% the resolution/measurement errors.
filename1 = char('CG_platform_constants');
%% Measured Parameters

thetaz  = xlsread(filename1,1,'C12'); % Rotation angle thetaz
thetay  = xlsread(filename1,1,'C13');  % tilt angle thetay
thetazb = xlsread(filename1,1,'C14'); % tilt angle thetazb

% all the bottom bracket vectors and angles
   rb  = xlsread(filename1,1,'C3:H5');  % Bottom bipod bracket
nodes at brackets
theta  = xlsread(filename1,1,'C6:H6'); % rode angles
    ci = xlsread(filename1,1,'C9:H9');  % horizontal node
distance

% Storing excel bracket parameters to appropriate variables
% for bipod 1
rb2 = rb(:,2);
phi1 = 180 - theta(1) - theta(2);
c1 = ci(1);

% from bipod 2
rb4 = rb(:,4);
phi2 = 180 - theta(3) - theta(4);
c2 = ci(3);

% for bipod 3
rb6 = rb(:,5);
phi3 = 180 - theta(5) - theta(6);
c3 = ci(5);
```

```
%% Load cell sizing inputs
cap   = xlsread(filename1,1,'C17');  % load cell capacity
lcErr  = xlsread(filename1,1,'C18'); % Load Cell Error
CGinput  = xlsread(filename1,1,'C19:C21'); % Payload Locations
mg  = xlsread(filename1,1,'C22'); % Payload Magnitude


%% Uncertaninties
% retrieve dtheta [4x1] uncertainty from Excel
dang  = xlsread(filename1,2,'B2:B5');

% retrieve dR uncertainty from Excel. dR = [dr1;dr2;dr3]
% where r = [rix; riy; riz]. dR is a 9x1 column vector
dr  = xlsread(filename1,2,'B6:B14');
end
```

D. 5 BipodNodes

```
function [R1, R2, R3, dR1, dR2, dR3] = bipodNodes
% Calculates the load intersection of the bipod rods for each
bipod (Rbi)
% as well as there uncertainty. The location of the bottom node
of each
% endrod are inputs (Ri) as column vectors in the
CG_platform_constants.xlsx
% - The outputs are all column vectors.
%% Load bipod constants
[theta,~,~,zeta,c1,c2,c3,phi1,phi2,phi3,rb2,rb4,rb6,dang,dr] =
LoadConstants;
clr = 0.075; % clearence between endrod and bipod brackets
syms A B C r ang rx ry rz     % symbolic variables
var = [A B C r ang rx ry rz]; % variable array
%% Bipod 1 node intersection using bottom bracket node 2
val = [theta(2)*pi/180 theta(1)*pi/180 phi1*pi/180 c1 zeta*pi/180
rb2(1) rb2(2) rb2(3)];
errs = [dang(1)*pi/180 dang(1)*pi/180 dang(1)*pi/180 dr(1)
dang(4)*pi/180 dr(1:3,1)'];

% symbolic expression for Rb1
rnrel = r*sin(B)/sin(C)*[0;-cos(A);sin(A)]; %  r_rel2 vector in
local coord
rpx2 = V_Rot(3,ang)*rnrel;                  %  r_rel2 vector in
global coord
Rex = [rx;ry;rz] + rpx2;                    %  Rb1 symbolic
expression

% calculates [Rb1x, Rb1y, Rb1z]
for i = 1:3
    [R1(i,1),dR1(i,1),~] = PropError(Rex(i),var,val,errs);
              rnrel1(i,1) =
double(subs(rnrel(i),var(1:4),val(1:4)));
end
%% Bipod 1 node intersection using bottom bracket node 4
 val = [theta(4)*pi/180 theta(3)*pi/180 phi2*pi/180 c2
zeta*pi/180 rb4(1) rb4(2) rb4(3)];
errs = [dang(1)*pi/180 dang(1)*pi/180 dang(1)*pi/180 dr(1)
dang(4)*pi/180 dr(1:3,1)'];

% symbolic expression for Rb2
rnrel = r*sin(B)/sin(C)*[-cos(A);0;sin(A)]; %  r_rel4 vector in
global coord
Rex = [rx;ry;rz] + rnrel;                   %  Rb2 symbolic
expression

% calculates [Rb2x; Rb2y; Rb2z]
for i = 1:3
```

```
    [R2(i,1),dR2(i,1),~] = PropError(Rex(i),var,val,errs);
             rnrel2(i,1) =
double(subs(rnrel(i),var(1:4),val(1:4)));
end
%% bottom bipod node/bracket info for rod 6
 val = [theta(6)*pi/180 theta(5)*pi/180 phi3*pi/180 c3 -
zeta*pi/180 rb6(1) rb6(2) rb6(3)];
errs = [dang(1)*pi/180 dang(1)*pi/180 dang(1)*pi/180 dr(1)
dang(4)*pi/180 dr(1:3,1)'];

% symbolic expression for Rb3
rnrel = r*sin(B)/sin(C)*[0;-cos(A);sin(A)]; % r_rel6 vector in
local coord
rpx2 = V_Rot(3,ang)*rnrel;                  % r_rel6 vector in
global coord
Rex = [rx;ry;rz] + rpx2;                    % Rb3 symbolic
expression

% calculates [Rb3x; Rb3y; Rb3z]
for i = 1:3
    [R3(i,1),dR3(i,1),~] = PropError(Rex(i),var,val,errs);
             rnrel3(i,1) =
double(subs(rnrel(i),var(1:4),val(1:4)));
end

% combined error in bipod intersection points
dR1 = ((R1*clr/2).^2+dR1.^2).^.5;
dR2 = ((R2*clr/2).^2+dR2.^2).^.5;
dR3 = ((R3*clr/2).^2+dR3.^2).^.5;

%% Ideal load line intersection point for each bipod
% R1=[-8.66; -5; 0];     % Intersection point for bipod 1
% R2=[0; 10; 0];         % Intersection point for bipod 2
% R3=[8.66; -5; 0];      % Intersection point for bipod 3
end
```

D. 6 WeightMagUnc

```
function [wt_val,error,percent_err,Smax_wt] =
WeightMagUnc(ValVec,ErrVec)
% Accepts nominal bipod forces and there error/uncertainty
values.
% The bipod loads (ValVec) and there errors (ErrVec) must be a
% row vectors of the same length
%
% Example:
%    - Nominal array of bipod loads and the bipod angle
%      ValVec = [F1 F2 F3 F4 F5 F6...
%                theta1 theta2 theta3 theta4 theta5 theta6];
%    - Nominal error of bipod loads and bipod angle error
%      ErrVec = [dF1 dF2 dF3 dF4 dF5 dF6 dtheta*ones(1,6)];
%
% function [wt_val,error,percent_err] =
WeightMagUnc(ValVec,ErrVec)

% Symbolic Variables
syms F0_1 F0_2 F0_3 F0_4 F0_5 F0_6 theta1 theta2 theta3 theta4
theta5 theta6

% weight expression
wt = -sin(theta1)*F0_1 -sin(theta2)*F0_2 -sin(theta3)*F0_3...
     -sin(theta4)*F0_4 -sin(theta1)*F0_5 -sin(theta1)*F0_6;

% row vector of variables
varlist = [F0_1 F0_2 F0_3 F0_4 F0_5 F0_6 ...
           theta1 theta2 theta3 theta4 theta5 theta6];

% output nominal weight and uncertainty
[wt_val,error,percent_err,Smax_wt] =
PropError(wt,varlist,ValVec,ErrVec);
end
```

D. 7 Unc4cg

```
function
[x_val,x_err,Smax_x,y_val,y_err,Smax_y,z_val,z_err,Smax_z] =
unc4cg(F_bi,dF_bi,mg,dmg,ang,dang,...

r1,r2,r3,dr1,dr2,dr3,xi,dxcg,yi,dycg)
% Calculates the CG and the uncertainty
% -------------------------------------
% Accepts the following inputs:
%   - nominal bipod forces [F_bi] is 6x1 and there
error/uncertainties [dF_bi] is also 6x1
%   - nominl weight(mg) and its error (dmg)
%   - all the nominal angles:
%       + [theta1:theta6, alpha, beta, gamma] 1x9
%   - the angle error are the same length as the nominal angles.
%       + Note: angle error is updated and saved in the excel
document:
%         'CG_platform_constants' on the 2nd sheet
%   - nominal intersection nodes for bipods 1,2 and 3 where
%     r1,r2 and r3 are all 3x1
%
%% initializing outputs
x_val = 0;      y_val = 0;      z_val = 0;
x_err = 0;      y_err = 0;      z_err = 0;
per_err_x = 0; per_err_y = 0;  per_err_z = 0;
Smax_x = 0;     Smax_y =0;      Smax_z = 0;

% Symbolic Variables
syms F0_1 F0_2 F0_3 F0_4 F0_5 F0_6 mg_mag
syms theta1 theta2 theta3 theta4 theta5 theta6
syms alpha0 beta0 gam
syms r1x r1y r1z   r2x r2y r2z   r3x r3y r3z

% cg coordinate expressions
[mg_exps, cgx, cgy, cgz] = CGsymbExpression;

% Nominal values for bipod loads, weight, angles and nodal
intersection
% length relative to the origin (ri)
Val = [F_bi' mg ang r1' r2' r3'];

% Error array of the bipod loads, weight and angles
Err = [dF_bi' dmg dang dr1' dr2' dr3'];

% row vector of variables
varlist = [F0_1 F0_2 F0_3 F0_4 F0_5 F0_6 mg_mag...
           theta1 theta2 theta3 theta4 theta5 theta6...
           alpha0 beta0 gam r1x r1y r1z r2x r2y r2z r3x r3y r3z];
%% Evalute CG using PropError
```

```
if nargin == 12      % If tilt angle is 0 degs: xcg and ycg can
be calculated
    [x_val,x_err,per_err_x,Smax_x] =
PropError(cgx,varlist,Val,Err);
    [y_val,y_err,per_err_y,Smax_y] =
PropError(cgy,varlist,Val,Err);

elseif nargin == 16  % If tilt angle is NOT 0 degs: zcg can be
calculated from xcg and ycg
    if Val(14) ~= 0  % if the tilt angle is not 0 degs the zcg
can be evaluted
        syms xcg ycg
        varlist = [varlist xcg ycg];
        Val = [Val xi yi];
        Err = [Err dxcg dycg];
        [z_val,z_err,per_err_z,Smax_z] =
PropError(cgz,varlist,Val,Err);
    else
       error(message('tilt angle alpha must NOT be 0 degs'))
    end
end
end
```

D. 8 SizeLoadCells

```
function
[Fabs1,dFabs1,Fcap,Fabs2,dFabs2,Fcap2,Ftare,dFtare,Ftare_cap] =
SizeLoadCells(option)
% This function can be used to examine load cell preformance
% relative to different CG location, payload magnitudes and
% platform orientations

%% Initialize Parameters
Fabs1 = zeros(6,1);       Fcap = zeros(6,1);    dFabs1 = Fcap;
Fabs2 = zeros(6,1);      Fcap2 = zeros(6,1);    dFabs2 = Fcap;
Ftare = zeros(6,1);  Ftare_cap = zeros(6,1);    dFtare = Fcap;

%% Absolute Reading with one sample
filename1 = char('CG_platform_constants');
    CG = xlsread(filename1,1,'C19:C21'); % cg
    mg = xlsread(filename1,1,'C22'); % payload magnitude
 alpha = xlsread(filename1,1,'C13'); % alpha angle
  beta = xlsread(filename1,1,'C12'); % beta angle
   cap = xlsread(filename1,1,'C18'); % load cell capacity
lc_err = xlsread(filename1,1,'C17'); % load cell error

% inital bipod load ouputs before Tare
[Fabs1,~,~,~,Fcap] = cg2force(CG,mg,alpha,beta,cap);
dFabs1 = Fabs1*lc_err/100; % error range of load cell
dFabs1 = abs(dFabs1);      % absolute error
  Fcap = abs(Fcap);        % absolute percent capacity

% displaying loads and percent load
    disp(['At a tilt angle of ',num2str(alpha)...
         ,' degs, a rotation angle of ',num2str(beta),' deg'])
    disp('  [ F_abs, +/-|dF_abs|, |%F_abs|] =')
    disp([Fabs1,dFabs1,Fcap])

%% Tare Load Cell
if option == 1;
    % Tare Excel Parameters
     CG1   = xlsread(filename1,1,'H17:H19');
     mg1   = xlsread(filename1,1,'H20'); % node location
    alpha1 = xlsread(filename1,1,'H13'); % node location
    beta1  = xlsread(filename1,1,'H12'); % node location

    % Second bipod load ouputs
    [Fabs2,~,~,~,Fcap2] = cg2force(CG1,mg1,alpha1,beta1,cap);
    dFabs2 = Fabs2*lc_err/100; % LC error for 2nd set up

    dFabs2 = abs(dFabs2)       % load error for 2nd set up
     Fcap2 = abs(Fcap2)        % abolute percent capacity for 2nd
set up
```

```
 %% Tare values (Relative)
    Ftare = Fabs2 - Fabs1;      % Tare load values
    dFtare = Ftare*lc_err/100; % Error range in tare loads
    Ftare_cap = Ftare/cap*100; % percent capacity for tare loads

    dFtare = abs(dFtare)        % absolute tare error
    Ftare = abs(Ftare_cap)      % absolute tare capacity

        disp(['After Tare: At a tilt angle of ',num2str(alpha)...
         ,' degs  and a rotation angle of ',num2str(beta),'
deg'])
        disp('  [ F_abs2, +/-|dF_abs2| , |%F_abs2| =')
        disp([Fabs2,dFabs2,Fcap2])

        disp('  [ F_tare, +/-|dF_tare| , |%F_tare] =')
        disp([Ftare,dFtare,Ftare_cap])
end
end
```

D. 9 Cg2force

```
function [forces,eps,sum_M,sum_F,Fcap_per] =
cg2force(cg,mg_mag,thetay,thetaz,cap)
% [CG,wt,tiltangle,platform rotation] ->
[load,strain,voltage,uncertainty]
% The CG is input as a column vector as a single vector or a
matrix. All
% angles must be in degrees.
% Outputs
%   - Bipod forces [F1;F2...,F6] {lbs}
%   -   "   strain [e1;e2...,e6] {me}
%
% Example:
% [F,eps,~,~] = cg2force([2;5;3],75,60,0)
%
% F =  -11.6337     eps = -18.0648
%      -22.4535           -32.6834
%      -53.3429           -89.0532
%       2.9168             4.2040
%       10.1665           16.2094
%       -0.6533           -1.0128
%
%% Constants %% %%
% Array of cross sectional area of axial rods <A1,A2...,A6>
[in^2]
A_xx = [.0644; .0687; .0599; .06938; .06272; .0645];

%%%%%%%%%%
% angles %
%%%%%%%%%%
% load angles from excel file
[theta,~,~,thetazb,~,~,~,~,~,~,~,~,~,~]= LoadConstants;

% bipod rod angle measure from the horizontal
theta = theta*pi/180;     % angle in [rads]

% bipod offset angle (see derivation for image)
thetazb = thetazb*pi/180; % angle in [rads]

% Platform tilt angle. Rotation about the y-axis
thetay = thetay*pi/180;   % angle in [rads]

% Platform rotation about z-axis (vertical rotation) angle
thetaz = thetaz*pi/180;   % angle in [rads]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[r1, r2, r3, dR1, dR2, dR3] = bipodNodes;

E = 1.00E+07; % Modulus of Elasticity [psi]
```

```
%% Calculations
% CG and axial force variables
syms F_a_1 F_a_2 F_a_3 F_a_4 F_a_5 F_a_6
syms xcg ycg zcg
%% Force transformations
F = [F_a_1 F_a_2 F_a_3 F_a_4 F_a_5 F_a_6];
varCG = [xcg ycg zcg];

%% Forces and Moment created from Bipod 1
F1 = F_a_1*[0;-cos(theta(1));-sin(theta(1))];  % Force in Rod 1
without bipod offset
F_1 = V_Rot(3,thetazb)*F1;                % Force in Rod 1 after
the applied bipod offset

F2 = F_a_2*[0; cos(theta(2));-sin(theta(2))];  % Force in Rod 2
without bipod offset
F_2 = V_Rot(3,thetazb)*F2;                % Force in Rod 2 after
the applied bipod offset

FB1 = vpa(F_1 + F_2,5);                   % sum of the force
vectors from bipod 1 5th decimal place
MF1 =  vpa(cross(r1,FB1),5);              % Moment vector from
bipod 1 to the 5th decimal place

%% Force and moments from Bipod 2
F3 = F_a_3*[-cos(theta(3));0;-sin(theta(3))];   % Force in Rod 3
F_3 = F3;                                  % Force in Rod 3 with
consistant naming convention

F4 = F_a_4*[cos(theta(4));0;-sin(theta(4))];    % Force in Rod 4
F_4 = F4;                                  % Force in Rod 4 with
consistant naming convention

FB2 = vpa(F_3 + F_4,5);                    % sum of the force
vectors from bipod 2 5th decimal place
MF2 =  vpa(cross(r2,FB2),5);               % Moment vector from
bipod 2 to the 5th decimal place

%% Force and Moment from Bipod 3
F5 = F_a_5*[0; cos(theta(5)); -sin(theta(5))];   % Force in Rod 5
without bipod offset
F_5 = V_Rot(3,-thetazb)*F5;                % Force in Rod 5 after
the applied bipod offset

F6 = F_a_6*[0; -cos(theta(6)); -sin(theta(6))];  % Force in Rod 6
without bipod offset
F_6 = V_Rot(3,-thetazb)*F6;                % Force in Rod 6 after
the applied bipod offset
```

```
FB3 = vpa(F_5 + F_6,5);        % sum of the force vectors from
bipod 3 5th decimal place
MF3 = vpa(cross(r3,FB3),5);    % sum of the force vectors from
bipod 3 5th decimal place

sum_M = MF1 + MF2 + MF3;       % sum of the moment in the bipods
[sum_MF1x;sum_MF1y;sum_MF1z]
sum_F = FB1 + FB2 + FB3;       % net forces in bipods
[sum_Fx;sum_Fy;sum_Fz]

%% Initializing Matrices
A_f = zeros(3,6,5,4);
b_f = zeros(3,1,5,4);
A_M = zeros(3,6,5,4);        % Initializing the Moment coefficient
matrix with zeros
b_M = zeros(3,1,5,4);        % Initializing the B matrix with
zeros

for j = 1:size(cg,2)              % CG position index
    for k = 1:length(cg(1,:))     % indexing needed to store
equation of motion matrices
        for m = 1:size(thetay,2)  % tilt angle index

% Weight force and moment created by weight and CG
    mg(:,m) = mg_mag*[sin(thetay(m));0;-cos(thetay(m))];  %
weight vector after tilting the platform
     W(:,m) = V_Rot(3,thetaz)*mg(:,m);                     %
weight vector after applying a rotation about z-axis
 M_w(:,j,m) = cross(cg(:,j),W(:,m));                       %
moment created by weight vector and CG location

%%  EOM into Matrix form (i.e. Ax=b)
% stores the coefficients of the three bipod forces and weight
vector into Matrix
    [A_f(:,:,m),b_f(:,:,m)] =
equationsToMatrix(FB1+FB2+FB3+W(:,m),F);   % varying tilt angle

% stores the coefficients of the three bipod moment and weight
vector into Matrix
    [A_M(:,:,j,m),b_M(:,:,j,m)] =
equationsToMatrix(MF1+MF2+MF3+M_w(:,j,m),F);
     A(:,:,j,m) = [A_f(:,:,m);A_M(:,:,j,m)];      % Forms
coeffiecent matrix from sum of the forces and moments Eqs
     b(:,:,j,m) = [b_f(:,:,m);b_M(:,:,j,m)];      % Forms B
matrix from sum of the forces and moment Eqs

     forces(:,j,m) =  A(:,:,j,m)\b(:,:,j,m);       % Calculates
the 6 Forces from the system of eqs
     eps(:,j,m) = forces(:,j,m)./(E*A_xx)*10^6;    % Calculates
the 6 Strain values in microstrains
```

```
        end
    end
end
Fcap_per = forces./cap*100; % percent capacity
end
```

D. 10 V_Rot

```
function Rotation = V_Rot(axis_num,angle)
%rotation matrix
%   choose 1 for rotation about the x-axis
%   choose 2 for rotation about the y-axis
%   choose 3 for rotation about the z-axis
%
%   Note: angle must be in radians
%   Example:
%       syms ang
%       RotVec = V_Rot(3,pi/3)
%
%       RotVec =
%         [0.5000   -0.8660         0
%          0.8660    0.5000         0
%               0         0    1.0000]

%% x axis
if axis_num == 1
Rotation = [1    ,   0,   0;...
            0, cos(angle),-sin(angle);...
            0,sin(angle),cos(angle)];...

%Rotation = double(Rotation)

%% y axis
elseif axis_num == 2
Rotation = [cos(angle) 0 sin(angle);...
            0,      1,        0;...
            -sin(angle) 0 cos(angle)];

%Rotation = double(Rotation)

%% z axis
elseif axis_num == 3
Rotation = [cos(angle), -sin(angle),0;...
            sin(angle), cos(angle),0;...
            0,            0,     1];

%Rotation = double(Rotation)
end
end
```

APPENDIX E. TILT TABLE CALCULATIONS

E. 1 Sizing Front Member Length

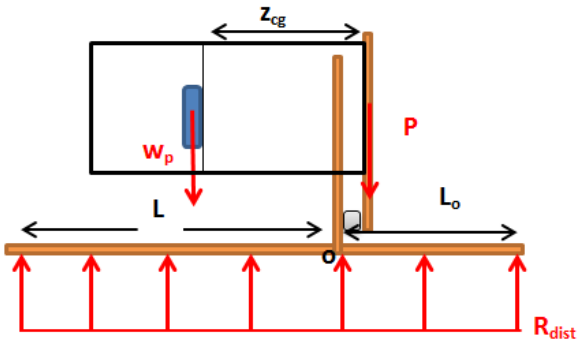

Figure 61: Free body diagram the CGA/Tilt Table assembly.

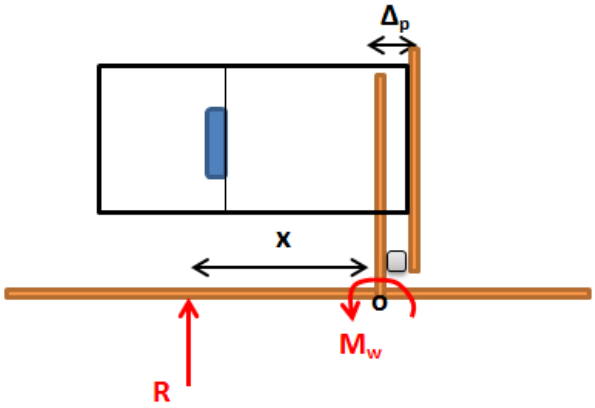

Figure 62: Equivalent FBD of CGA/Tilt Table assembly.

The location of the average reactional force can be calculated by summing the moments about point O.

$$\Sigma M_o = 0$$

$$R * x = (z_{cg} w_p)$$

$$x = \frac{z_{cg} w_p}{(w_p + P)}$$

The length of the front member L can be found by relating the location of the average distributed reactional force (x) to its location relative to point O.

$$x = \frac{(L + L_o)}{2} - L_o = L - \frac{L_o}{2}$$

$$L - \frac{L_o}{2} = \frac{z_{cg} w}{(w_p + P)}$$

$$L = \frac{z_{cg} w}{(w_p + P)} + \frac{L_o}{2}$$

E. 2 Weld Calculations

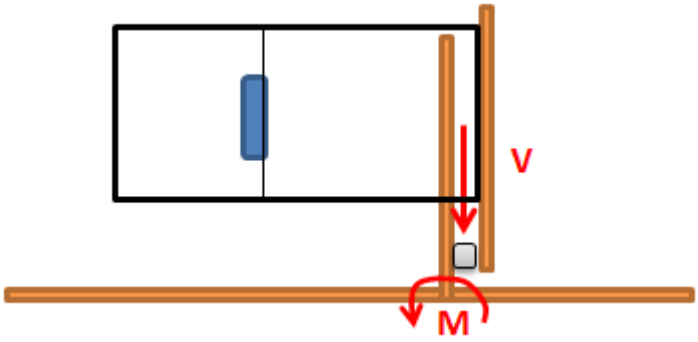

Figure 63: Shear and moment translated at the post.

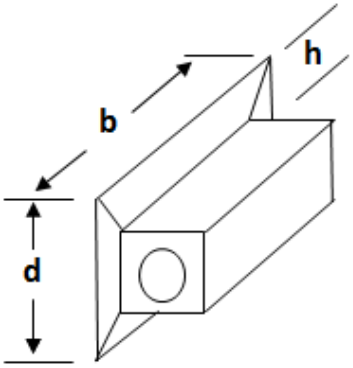

Figure 64: Illustration of the all-around weld joint of the pin connected on the posts.

Note: The following equations were pulled from [12].

Primary shear stress:

$$\tau' = \frac{V}{A_p}$$

Throat area of the pin joint where b and d are the height and width of the welded area respectively

$$A_p = 1.414h(b + h)$$

Unit Second moment of area of the pin joint

$$I_u = \frac{d^2}{6}(3b + d)$$

I is the second moment of area for the pin

$$I = 0.707hI_u$$

Secondary shear for the base

$$\tau'' = \frac{Mr}{I}$$

Shear magnitude from Pythagorean combination

$$\tau = (\tau'^2 + \tau''^2)^{\frac{1}{2}}$$

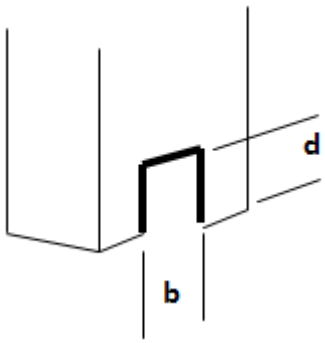

Figure 65: Illustration of the welds on the bottom of the posts.

The Primary shear is again defined by

$$\tau' = \frac{V}{A_b}$$

Where A_b is the throat area of the weld at the base.

$$A_b = 0.707h(b + 2d)$$

And I_u is the Unit second moment of area

$$I_u = \frac{d^2}{6}(3b + d)$$

I is the second moment of area

$$I = 0.707hI_u$$

Secondary shear stress:

$$\tau'' = \frac{Mr}{I}$$

Combined stress:

$$\tau = (\tau'^2 + \tau''^2)^{\frac{1}{2}}$$

Both welding calculations were translated into a MATLAB script file. The script is included on the next page.

The Factor of Safety for both joint is defined below:

$$n = \frac{S_{sy}}{\tau}$$

E. 3 Weld Strength MATLAB Script

```
%% weld strength calcs
F = 100;
Sy = 50e3;
Sut = 62e3;

% weld geometry of pin joint
b = 1;
h = 1/4;
d = .9;

A = 1.414*h*(b+d);  % area of fillet on pin join
Iu = d^2*(3*b+d)/6; % unit 2nd moment of inertia
I = 0.707*h*Iu;     % 2nd moment of inerita

% weld geometry of fixed base fillet joints
b1 = 1+1/8;
d1 = 1+1/8;
ybar = d1^2/(b1 + 2*d1);

A1 = 0.707*h*(b1 + 2*d1);    % area of fillet on fixed base
Iu1 = 2*d1^3/3 - 2*d1^2*ybar + (b1 + 2*d1)*ybar^2
I1 = 0.707*h*Iu1;

AA = [A A1];
dd = [d d1];
II = [I I1];

tou1 = F./AA;                    % primary shear
tou2 = (F*21).*(dd/2)./(II);     % secondary shear
tou = sqrt(tou1.^2 + tou2.^2);  % shear magnitude

n = .577*Sy./(tou.*.5)       % factor of safety
nut = .3*Sut./(tou.*.5)      % FOS using ultimate tensile strength
```

Table 15: Numerical results of the expected and measure data gathered from endrod assembly 1.

	Expected		Measured			
Instron Load	V_{out}	ε_{expect}	V_{out}	ε_{meas}	F_{meas}	ε_{DIFF}
[lbs]	[mv]	[με]	[mv]	[με]	[lbs]	[με]
5.0	0.018	7.8	0.039	17.4	11.2	9.6
13.6	0.048	21.1	0.053	23.4	15.0	2.2
35.8	0.126	55.6	0.186	82.2	52.9	26.6
80.0	0.281	124.2	0.380	168.2	108.3	44.0
106.6	0.374	165.5	0.473	209.2	134.7	43.7
-11.7	-0.041	-17.9	-0.041	-18.1	-11.9	-0.2
-20.4	-0.071	-31.0	-0.096	-42.6	-28.0	-11.6
-34.6	-0.121	-52.7	-0.118	-52.3	-34.3	0.4
-65.2	-0.229	-99.3	-0.291	-128.7	-84.6	-29.4
-101.1	-0.355	-153.9	-0.388	-171.6	-112.7	-17.7

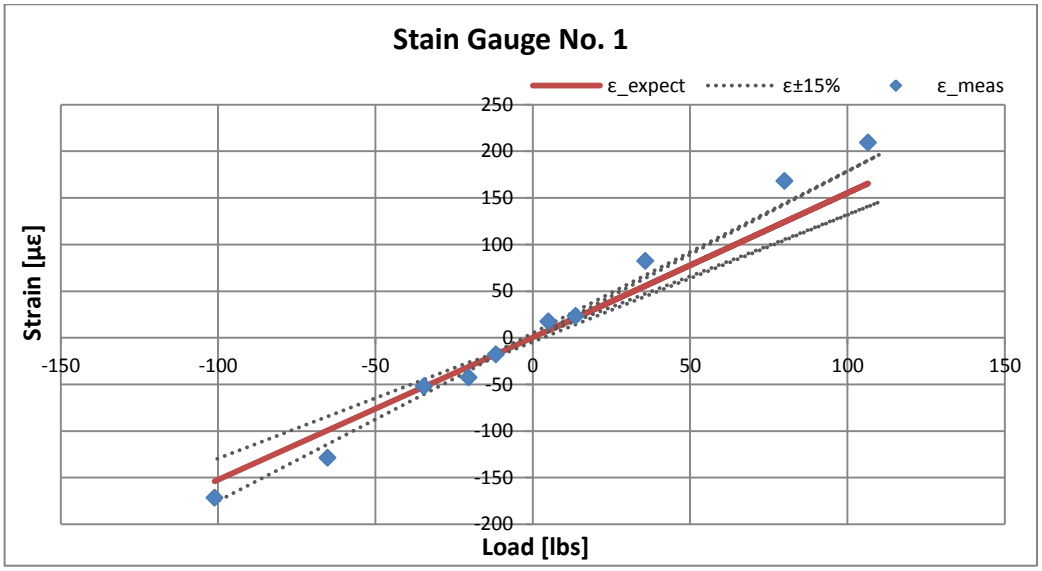

Figure 66: Applied Instron load versus the expected and measured strains for Endrod Assembly No. 1.

Table 16: Numerical results of the expected and measure data gathered from endrod assembly 2.

	Expected		Measured			
Instron Load	V_{out}	ε_{expect}	V_{out}	ε_{expec}	V_{out}	ε_{DIFF}
[lbs]	[mv]	[με]	[mv]	[με]	[lbs]	[με]
9.5	0.031	13.8	0.000	0.0	0.0	-13.8
37.6	0.124	54.7	0.097	43.0	29.5	-11.8
52.1	0.171	75.8	0.097	43.0	29.5	-32.8
72.5	0.238	105.5	0.194	85.8	59.0	-19.6
90.3	0.297	131.4	0.272	120.3	82.6	-11.2
111.4	0.367	162.2	0.319	141.1	96.9	-21.0
-10.5	-0.035	-15.0	-0.098	-43.2	-30.3	-28.2
-50.9	-0.168	-72.7	-0.126	-55.9	-39.2	16.8
-73.0	-0.240	-104.1	-0.201	-88.8	-62.2	15.4
-96.0	-0.316	-137.0	-0.292	-129.2	-90.5	7.8
-123.4	-0.406	-176.1	-0.370	-163.6	-114.7	12.5

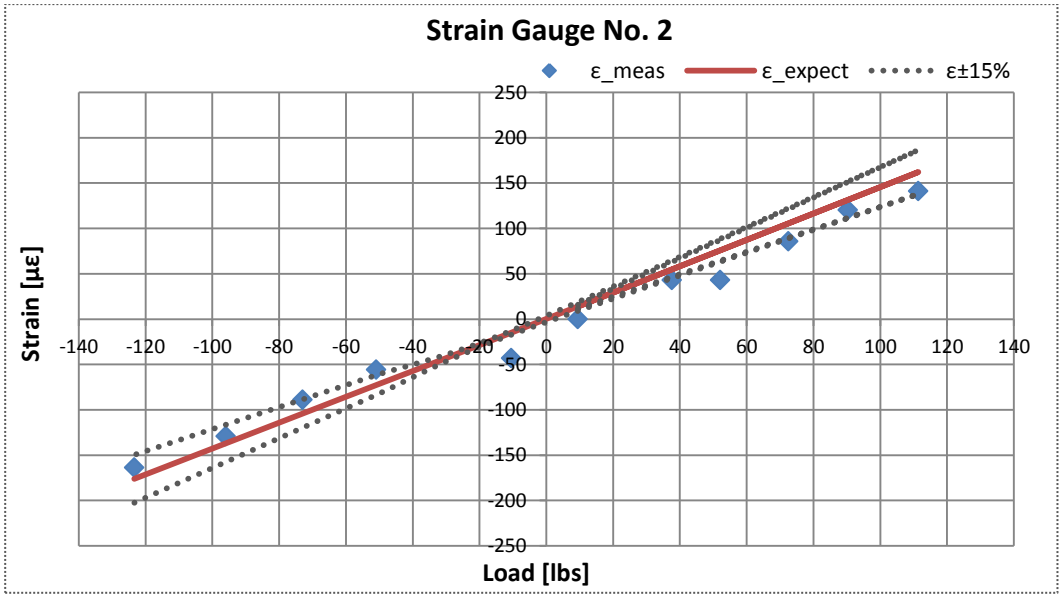

Figure 67: Applied Instron load versus the expected and measured strains for Endrod Assembly No. 2.

Table 17: Numerical results of the theoretical and experimental data gathered from endrod assembly 3.

	Expected		Measured			
Instron Load	V_{out}	ε_{expect}	V_{out}	ε_{meas}	F_{meas}	ε_{DIFF}
[lbs]	[mv]	[με]	[mv]	[με]	[lbs]	[με]
0.0	0.000	0.00	-0.084	-37.29	-22.3	-37.29
6.5	0.025	10.85	-0.038	-16.61	-10.0	-27.46
19.0	0.072	31.72	0.013	5.71	3.4	-26.01
35.0	0.132	58.43	0.097	42.74	25.6	-15.69
39	0.147	65.10	0.101	44.60	26.7	-20.51
60.0	0.226	100.16	0.185	82.03	49.1	-18.13
80.0	0.302	133.55	0.209	92.64	55.5	-40.90
-10.0	-0.038	-16.69	-0.084	-37.29	-22.8	-20.59
-23.0	-0.087	-38.39	-0.084	-37.17	-22.7	1.23
-39	-0.147	-65.10	-0.119	-52.62	-31.5	12.49
-42.0	-0.158	-70.11	-0.181	-80.28	-49.1	-10.17
-70.0	-0.264	-116.85	-0.247	-109.05	-66.6	7.80
-85	-0.321	-141.89	-0.285	-126.08	-75.5	15.82
-96.0	-0.362	-160.26	-0.313	-138.67	-84.7	21.59
-114	-0.430	-190.30	-0.383	-169.43	-101.5	20.87

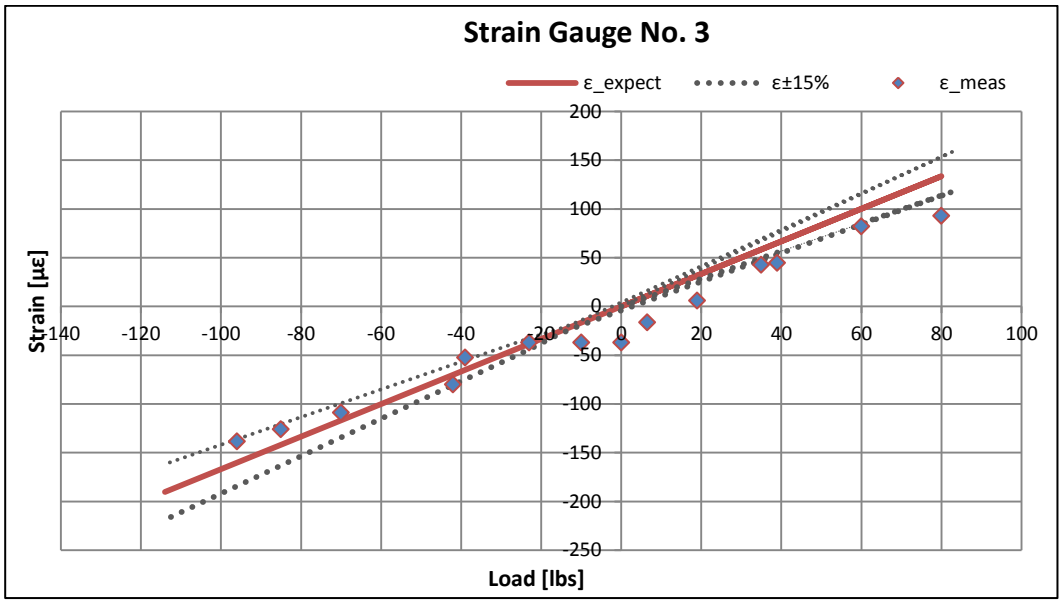

Figure 68: Applied Instron load versus the expected and measured strains for Endrod Assembly No. 3.

Table 18: Numerical results of the expected and measured data gathered from endrod assembly 4 before nonlinearity correction.

	Expected		Measured			
Instron Load	V_{out}	E_{expect}	V_{out}	E_{meas}	F_{meas}	ε_{DIFF}
[lbs]	[mv]	[με]	[mv]	[με]	[lbs]	[με]
2.8	0.010	4.0	0.000	0.01	0.0	-4.0
18.5	0.064	26.7	0.094	39.26	27.2	12.6
49.5	0.171	71.3	0.194	81.09	56.3	9.7
50.5	0.174	72.8	0.194	81.09	56.3	8.3
66.0	0.228	95.1	0.291	121.63	84.4	26.5
92.0	0.317	132.6	0.389	162.48	112.7	29.9
115.0	0.397	165.7	0.486	203.13	140.9	37.4
0.3	0.001	0.4	0.000	0.00	0.0	-0.4
-4.0	-0.014	-5.7	-0.013	-5.54	-3.9	0.2
-19.5	-0.067	-27.5	-0.100	-41.90	-29.7	-13.8
-69.5	-0.240	-98.2	-0.363	-151.91	-107.5	-51.7

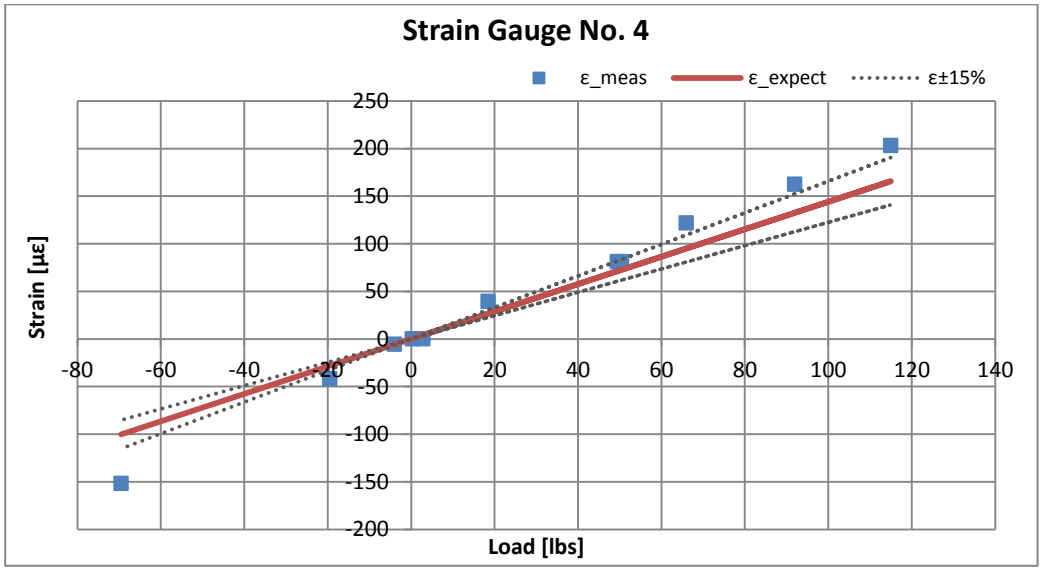

Figure 69: Applied Instron load versus the expected and measured strains for Endrod Assembly No. 4.

Table 19: Numerical results of the expected and measured data gathered from endrod assembly 5.

	Theoretical		Experimental			
Instron Load	V_{out}	E_{expect}	V_{out}	E_{meas}	F_{meas}	ε_{DIFF}
[lbs]	[mv]	[με]	[mv]	[με]	[lbs]	[με]
0.4	0.001	0.6	0.007	3.0	1.9	2.4
16.5	0.059	26.3	0.092	40.6	25.5	14.3
52.5	0.189	83.7	0.194	85.6	53.7	1.9
74.5	0.269	118.8	0.308	136.1	85.4	17.3
99.5	0.359	158.6	0.393	174.0	109.1	15.3
128.5	0.463	204.9	0.505	223.2	140.0	18.3
-14.5	-0.052	-22.7	-0.017	-7.6	-4.9	15.5
-50.5	-0.182	-78.9	-0.167	-73.8	-47.2	6.8
-70.0	-0.252	-109.4	-0.207	-91.5	-58.6	20.1
-65.0	-0.234	-101.6	-0.188	-83.2	-53.3	20.4
-92.5	-0.333	-144.6	-0.304	-134.3	-85.9	13.2
-113.5	-0.409	-177.4	-0.399	-176.5	-112.9	4.5

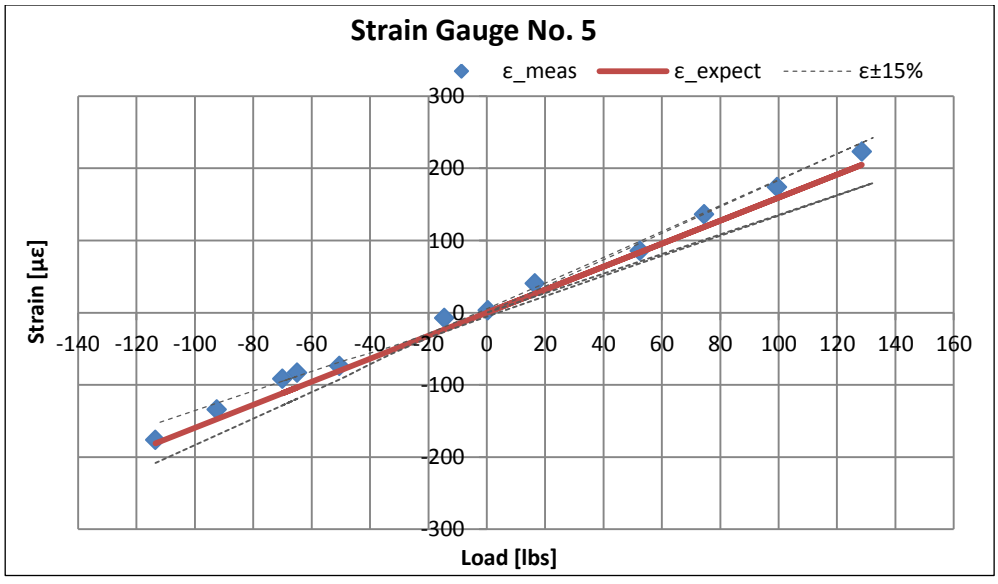

Figure 70: Applied Instron load versus the expected and measured strains for Endrod Assembly No. 5.

Table 20: Numerical results of the expected and measured data gathered from endrod assembly 6.

	Expected		Measured			
Instron Load	V_{out}	E_{expect}	V_{out}	E_{meas}	F_{meas}	ε_{DIFF}
[lbs]	[mv]	[με]	[mv]	[με]	[lbs]	[με]
25.0	0.093	38.7	0.072	30.23	19.508	-8.51
44.0	0.163	68.2	0.104	43.51	28.077	-24.68
68.0	0.252	105.4	0.194	81.31	52.471	-24.07
94.0	0.348	145.7	0.248	103.70	66.914	-41.98
117.0	0.433	181.3	0.292	122.16	78.828	-59.16
-34.0	-0.126	-52.7	-0.097	-40.66	-26.238	12.03
-63.0	-0.233	-97.6	-0.255	-106.58	-68.776	-8.95
-94.0	-0.348	-145.7	-0.403	-168.68	-108.847	-23.01
-131.0	-0.485	-203.0	-0.577	-241.22	-155.658	-38.21

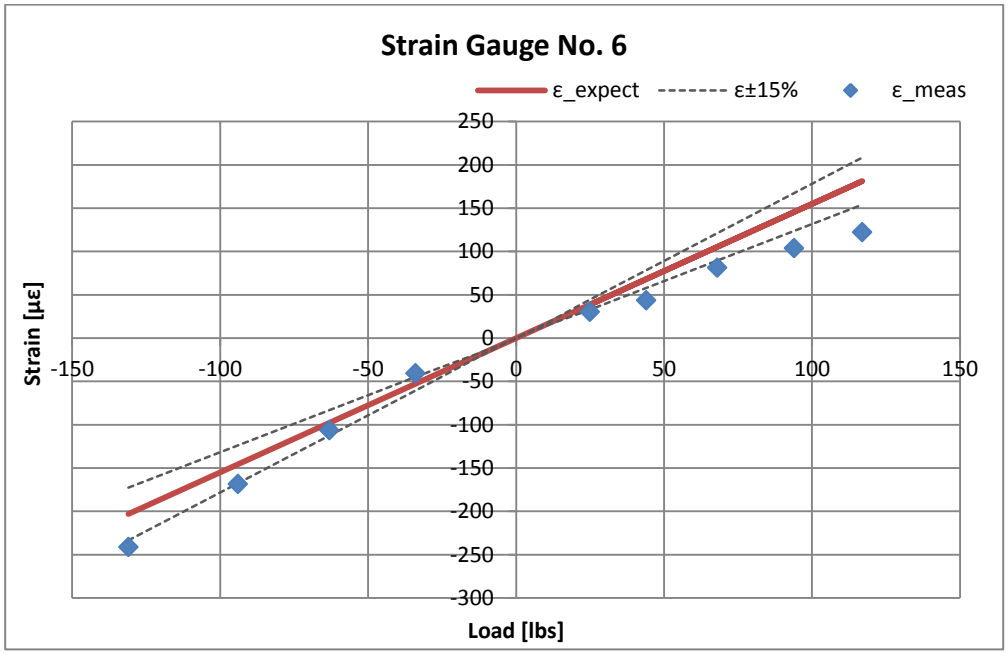

Figure 71: Applied Instron load versus the expected and measured strains for Endrod Assembly No. 6.

Table 21: Strain and load values for the new strain gauge on endrod assembly 6 before and after nonlinearity corrections are made.

		Before Correction			After correction		
Instron Load	ε_{actual}	ε_{meas}	ε_{DIFF}	F_{meas}	ε_{cor}	ε_{DIFF}	F_{cor}
[lbs]	[με]	[με]	[με]	[lbs]	[με]	[με]	[lbs]
0.13	0.2	27.6	27.4	17.8	20.8	20.6	13.43
-21.5	-33.3	-79.5	-46.2	-51.3	-59.9	-26.6	-38.66
-53.5	-82.9	-175.7	-92.8	-113.4	-132.5	-49.6	-85.45
-86	-133.3	-256.9	-123.6	-165.8	-193.6	-60.4	-124.90
-159	-246.4	-338.5	-92.0	-218.4	-255.1	-8.7	-164.55
-0.3	-0.5	13.4	13.9	8.7	10.1	10.6	6.51
9.5	14.7	27.6	12.9	17.8	20.8	6.1	13.43
29	44.9	66.9	22.0	43.2	50.5	5.5	32.56
68.5	106.2	133.0	26.9	85.8	100.3	-5.8	64.72
91	141.0	185.8	44.7	119.9	140.1	-0.9	90.37
91	141.0	189.1	48.1	122.0	142.6	1.6	92.00
134	207.7	270.7	63.0	174.7	204.2	-3.5	131.70

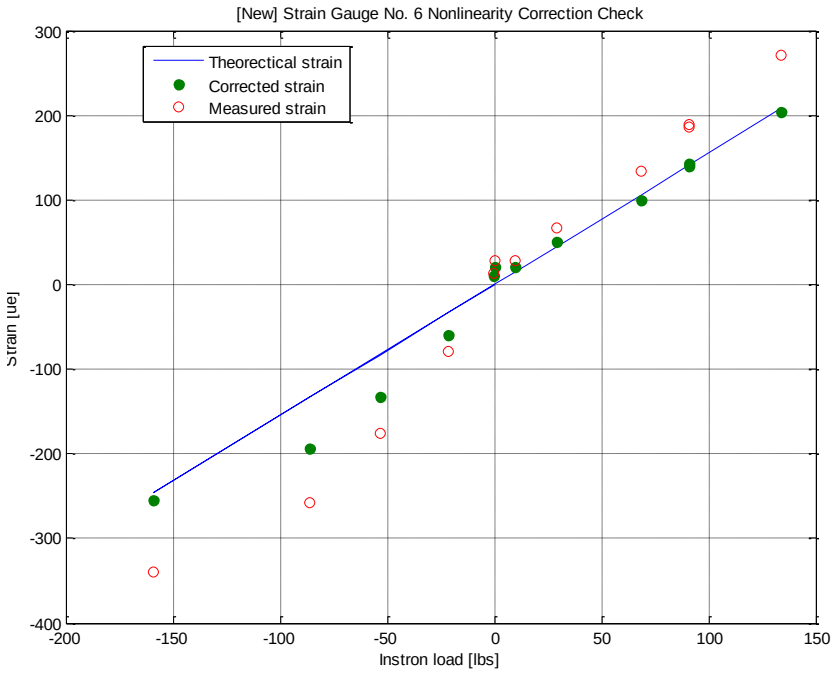

Figure 72: Nonlinearity correction plot for endrod assembly 6 with a new gauge.

APPENDIX G. PLATFORM TESTING TEST COMBINATIONS AND RESULTS

G. 1 Test Combination Tables

Table 22: Test combinations 1-28 for z_{shelf} = 5.5 in

data points with Z_{shelf} = 5.5 in			
Test No	CG shelf Location	Tilt Angle (α) [Degs]	Weight [lbs]
1	-	0	0
2	-	30	0
3	-	60	0
4	-	90	0
5	CC	0	35.77
6	CC	0	43.47
7	CC	30	35.77
8	CC	30	43.47
9	CC	60	35.77
10	CC	60	43.47
11	CC	90	35.77
12	CC	90	43.47
13	BL	0	35.77
14	BL	0	43.47
15	BL	30	35.77
16	BL	30	43.47
17	BL	60	35.77
18	BL	60	43.47
19	BL	90	35.77
20	BL	90	43.47
21	TL	0	35.77
22	TL	0	43.47
23	TL	30	35.77
24	TL	30	43.47
25	TL	60	35.77
26	TL	60	43.47
27	TL	90	35.77
28	TL	90	43.47

Table 23: Test combinations 28-44 for z_{shelf} = 5.5 in.

data points with Z_{shelf} = 5.5 in			
Test No	CG shelf Location	Tilt Angle (α) [Degs]	Weight [lbs]
29	TR	0	35.77
30	TR	0	43.47
31	TR	30	35.77
32	TR	30	43.47
33	TR	60	35.77
34	TR	60	43.47
35	TR	90	35.77
36	TR	90	43.47
37	BR	0	35.77
38	BR	0	43.47
39	BR	30	35.77
40	BR	30	43.47
41	BR	60	35.77
42	BR	60	43.47
43	BR	90	35.77
44	BR	90	43.47

Table 24: Test combinations 45-56 for z_{shelf} = 10.875 in.

data points with Z_{shelf} = 10.875 in			
Test No	CG shelf Location	Tilt Angle (α) [Degs]	Weight [lbs]
45	-	0	0
46	-	30	0
47	-	60	0
48	-	90	0
49	CC	0	35.77
50	CC	0	43.47
51	CC	30	35.77
52	CC	30	43.47
53	CC	60	35.77
54	CC	60	43.47
55	CC	90	35.77
56	CC	90	43.47

Table 25: Test combinations 57-56 for z_{shelf} = 10.875 in.

data points with Z_{shelf} = 10.875 in			
Test No	CG shelf Location	Tilt Angle (α) [Degs]	Weight [lbs]
57	BL	0	35.77
58	BL	0	43.47
59	BL	30	35.77
60	BL	30	43.47
61	BL	60	35.77
62	BL	60	43.47
63	BL	90	35.77
64	BL	90	43.47
65	TL	0	35.77
66	TL	0	43.47
67	TL	30	35.77
68	TL	30	43.47
69	TL	60	35.77
70	TL	60	43.47
71	TL	90	35.77
72	TL	90	43.47
73	TR	0	35.77
74	TR	0	43.47
75	TR	30	35.77
76	TR	30	43.47
77	TR	60	35.77
78	TR	60	43.47
79	TR	90	35.77
80	TR	90	43.47
81	BR	0	35.77
82	BR	0	43.47
83	BR	30	35.77
84	BR	30	43.47
85	BR	60	35.77
86	BR	60	43.47
87	BR	90	35.77
88	BR	90	43.47

G. 2 Platform Tilting Test Results

Table 26: Actual and measured results for the planar CG components for z_{shelf} = 5.5 in.

	Actual						Experimental					
Payload Shelf Location	W_{act}	$\sigma_{W,a}$	x_{act}	$\sigma_{x,a}$	y_{act}	$\sigma_{y,a}$	W_{meas}	$\sigma_{W,m}$	x_{meas}	$\sigma_{x,m}$	y_{meas}	$\sigma_{y,m}$
	[lb]		[in]				[lb]		[in]			
Center	35.77	0.005	0.02	0.0313	1.841	0.0313	36.007	5.581	1.255	0.339	1.723	0.454
	43.47	0.005					37.969	5.885	0.544	0.224	2.533	0.638
Bottom left	35.77	0.005	-4.23	0.0313	-1.409	0.0313	27.639	4.327	-4.180	0.956	-1.358	0.391
	43.47	0.005					39.059	6.052	-3.335	0.773	-1.387	0.373
Top Left	35.77	0.005	-4.23	0.0313	5.091	0.0313	32.969	5.146	-2.341	0.614	3.665	0.858
	43.47	0.005					40.293	6.271	-3.449	0.783	4.249	0.971
Top Right	35.77	0.005	4.27	0.0313	5.091	0.0313	27.889	4.406	3.788	0.917	5.858	1.351
	43.47	0.005					20.015	3.233	4.049	1.034	7.239	1.700
Bottom Right	35.77	0.005	4.27	0.0313	-1.409	0.0313	21.810	3.450	1.335	0.504	-0.769	0.360
	43.47	0.005					21.010	3.334	2.840	0.766	-0.953	0.385

Table 27: Difference between the measured and the actual planar CG results for z_{shelf}=5.5 in.

Payload Shelf Location	Actual	Difference		
	W_{act}	ΔW	Δx	Δy
	[lb]	[lb]	[in]	
Center	35.77	0.237	1.235	-0.387
	43.47	-5.501	0.524	0.423
Bottom left	35.77	-8.131	0.050	-0.218
	43.47	-4.411	0.896	-0.247
Top Left	35.77	-2.802	1.889	-1.695
	43.47	-3.177	0.781	-1.111
Top Right	35.77	-7.881	-0.482	0.498
	43.47	-23.455	-1.043	1.879
Bottom Right	35.77	-13.960	-2.935	-1.909
	43.47	-22.460	-1.430	-2.093

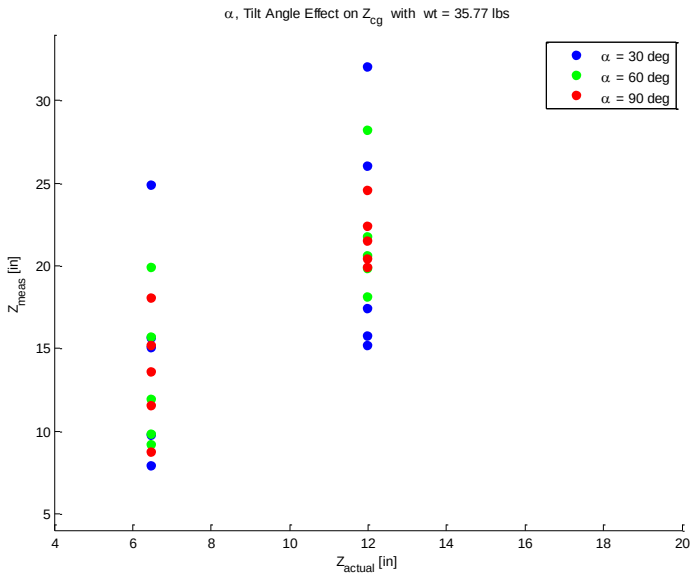

Figure 73: Scatter plot of the z_{meas} versus the z_{act} for a 35.77 lb payload.

Table 28: Platform tilting test results for z_{meas} at different planar payload locations and α's at z_{shelf} = 5.378 in.

	Actual			Measured						Difference		
Payload Shelf Location	W_{act}	z_{act}	σ_{zact}	z_{30}	σ_{z30}	z_{60}	σ_{z60}	z_{90}	σ_{z30}	Δz_{30}	Δz_{60}	Δz_{90}
	[lb]	[in]		[in]						[in]		
Center	35.77	6.59	0.035	7.862	2.759	7.253	2.162	7.350	1.850	1.272	0.663	0.760
	43.47	7.41	0.034	10.359	3.199	8.905	2.458	8.026	2.013	2.949	1.495	0.616
Bottom left	35.77	6.59	0.035	12.689	3.308	13.305	2.900	11.259	2.847	6.099	6.715	4.669
	43.47	7.41	0.034	10.629	2.741	9.395	2.103	10.023	2.496	3.219	1.985	2.613
Top Left	35.77	6.59	0.035	6.194	1.803	8.118	1.924	9.630	2.441	-0.396	1.528	3.040
	43.47	7.41	0.034	8.618	2.306	8.441	1.934	9.637	2.428	1.208	1.031	2.227
Top Right	35.77	6.59	0.035	13.058	5.100	9.815	3.122	13.959	3.069	6.468	3.225	7.369
	43.47	7.41	0.034	25.459	8.711	17.998	5.212	20.675	4.543	18.049	10.588	13.265
Bottom Right	35.77	6.59	0.035	21.309	6.522	16.908	4.369	14.920	3.743	14.719	10.318	8.330
	43.47	7.41	0.034	24.574	8.037	20.908	5.542	18.174	4.546	17.164	13.498	10.764

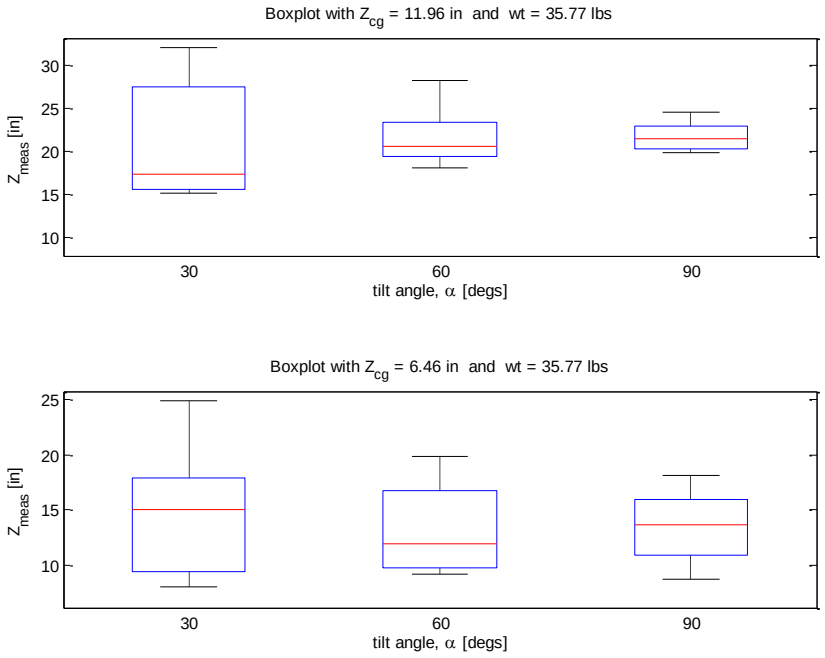

Figure 74: Boxplot of z_{meas} at various tilt angles using a 43.47 lb payload.

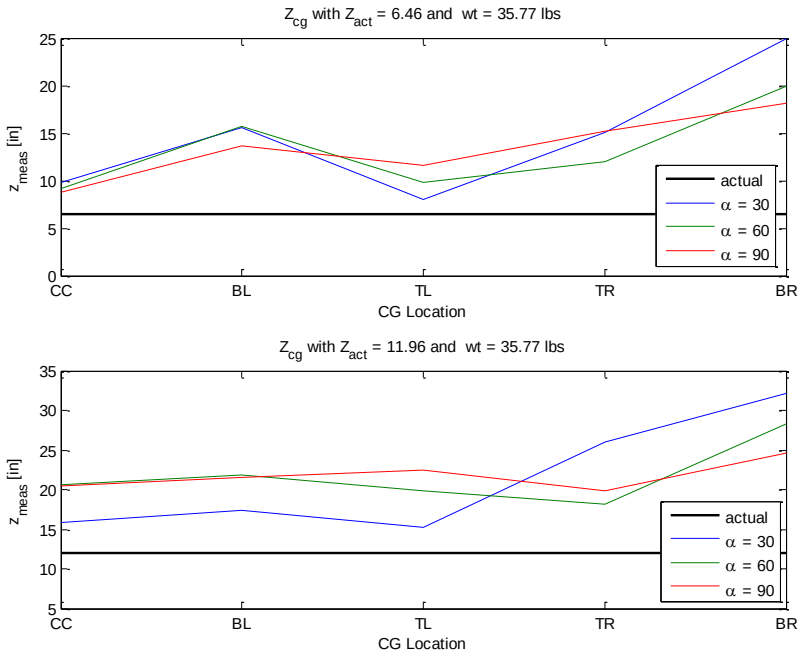

Figure 75: Compares z_{meas} with the payload planar location using a 35.77 lb payload.

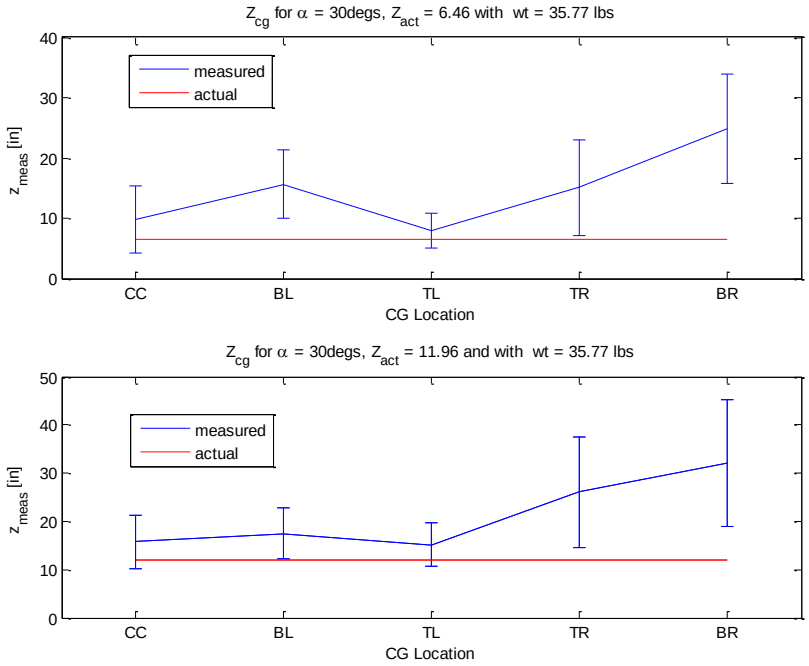

Figure 76: Compares z_{meas} with the payload planar location using a 43.47 lb payload at α=30^{o}and shows the uncertainty range.

Table 29: ABAQUS and MATLAB Bipod Rod load comparison with a payload on the bottom left corner of the shelf (BL).

	Reverse Comparison (BL)			
	σ_{ABAQUS}	F_{ABAQUS}	F_{MATLAB}	Diff
Bipod Rod Index	[lb/in^2]	[lb]	[lb]	[%]
1	-371.314	-23.913	-23.917	-0.017
2	-371.314	-23.913	-23.917	-0.017
3	-151.404	-9.750	-9.751	-0.009
4	-151.404	-9.750	-9.751	-0.009
5	-65.6125	-4.225	-4.226	-0.008
6	-65.6125	-4.225	-4.226	-0.008
α = 0 , β = 0				

Table 30: Forward results comparison between the actual values and the results generated using loads generated by FEM as inputs to the CG code. Payload set to the bottom left corner of the shelf (BL).

	Forward Comparison (BL)		
α = 0 , β = 0	x	y	mg
units	[in]	[in]	[lbs]
Actual Values	-4.50	-1.140	43.470
From ABAQUS Loads	-4.50	-1.139	43.464
Percent diff	0.004	0.088	0.014

Table 31: ABAQUS and MATLAB Bipod Rod load comparison with a payload on the bottom left corner of the shelf (BL) and α = -60 , β = 0.

	Reverse Comparison (BL)			
	σ_{ABAQUS}	F_{ABAQUS}	F_{MATLAB}	Diff
Bipod Rod Index	[lb/in^2]	[lb]	[lb]	[%]
1	-432.98	-27.884	-27.887	-0.010
2	-697.952	-44.948	-44.952	-0.010
3	-299.995	-19.320	-19.322	-0.009
4	148.591	9.569	9.570	-0.010
5	479.475	30.878	30.881	-0.009
6	214.504	13.814	13.815	-0.007
α = -60 , β = 0				

Table 32: Forward results comparison with the payload set to the bottom left corner of the shelf (BL) and α = -60 , β = 0.

	Forward Comparison (BL)			
α = -60 , β = 0	x	y	Z	mg
units	[in]	[in]	[in]	[lbs]
Actual Values	-4.5	-1.14	12.91	43.47
From ABAQUS Loads	-4.4998	-1.1398	12.9108	43.4639
Percent diff	0.0044	0.0175	-0.0062	0.0140

APPENDIX I. BIPOD ROD LOADS AND STRAIN COMPARISON

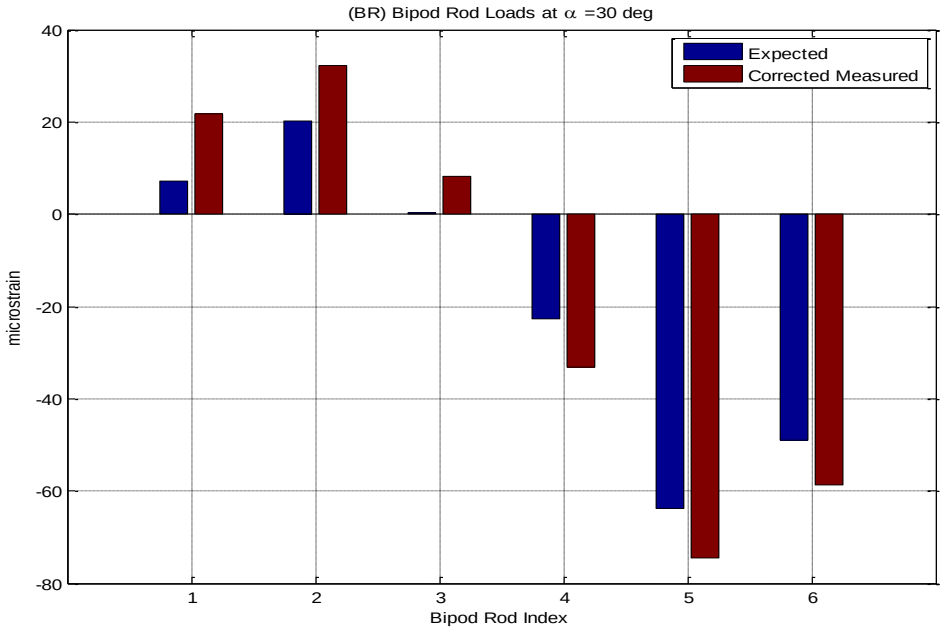

Figure 77: Measured and expected bipod rod strains at $\alpha = 30^{o}$ while the actual payload is at the BR corner of the shelf.

Table 33: Load and strain results for the expected and measured cases. Parameters: $\alpha = 30^{o}$, Payload at BR and $z_{act} = 12.72$ in.

BPR Index	F_{expect}	F_{meas}	F_{DIFF}	ε_{expect}	ε_{CM}	ε_{Diff}
	[lb]			[με]		
1	4.60	14.04	9.44	7.14	21.81	14.66
2	13.88	22.21	8.34	20.20	32.33	12.14
3	0.12	4.95	4.83	0.21	8.27	8.06
4	-15.80	-22.95	-7.15	-22.77	-33.08	-10.31
5	-39.95	-46.68	-6.73	-63.70	-74.43	-10.73
6	-31.66	-37.83	-6.17	-49.08	-58.64	-9.56
					$\|\|\varepsilon_{DIFF}\|\|$	65.46

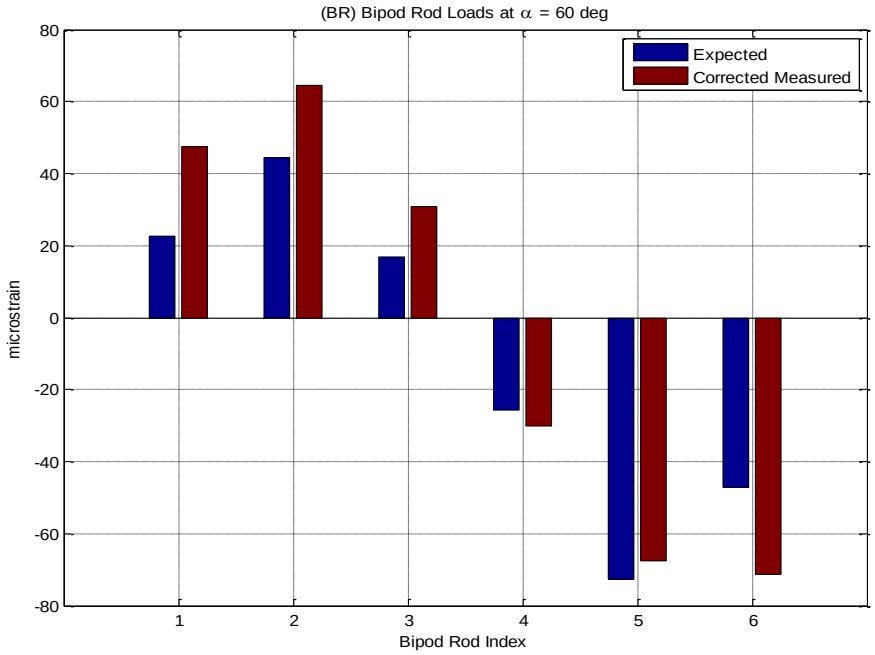

Figure 78: Measured and expected bipod rod strains at $\alpha = 60^{o}$ while the actual payload is at the BR corner of the shelf.

Table 34: Load and strain results for the expected and measured cases. Parameters: $\alpha = 60^{o}$, Payload at BR and $z_{act} = 12.72$ in.

BPR Index	F_{expect}	F_{meas}	F_{DIFF}	ε_{expect}	ε_{CM}	ε_{Diff}
	[lb]			[με]		
1	14.48	30.51	16.03	22.48	47.37	24.89
2	30.54	44.43	13.89	44.45	64.67	20.21
3	10.10	18.47	8.37	16.86	30.83	13.97
4	-17.79	-20.87	-3.07	-25.64	-30.07	-4.43
5	-45.53	-42.44	3.09	-72.59	-67.66	4.93
6	-30.33	-46.07	-15.74	-47.02	-71.43	-24.40
					$\|\|\varepsilon_{DIFF}\|\|$	92.84