THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AEROSPACE ENGINEERING

DYNAMIC CHARACTERIZATION OF FORCE TRANSDUCERS FOR WIND TUNNEL MEASUREMENTS

GARRETT LEO TROWBRIDGE
FALL 2024

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Aerospace Engineering and Engineering Science
with honors in Aerospace Engineering

Reviewed and approved* by the following:

Dr. Nicholas Vlajic
Professor of Mechanical Engineering
Thesis Supervisor

Dr. Sven Schmitz
Professor of Aerospace Engineering
Honors Adviser

* Electronic approvals are on file.

Abstract

Dynamic forces are increasingly present as wind tunnel speed increases. However, problems arise with the calibration of dynamic forces. Dynamic forces that are calibrated like the traditional static forces will not read accurately. The current investigation explores two new methods to calibrate and measure dynamic forces by means of adding accelerometers in addition to static transducers. The approaches involve determining inertial, damping, and static coefficients from acceleration, velocity, and position data. The first method simultaneously solves for all three coefficients from a known force and respective data. The second method involves conducting a static calibration. The static force is then subtracted from the dynamic force and the other coefficients can be solved for. A computational simulation yields errors less than 0.1 percent for both methods, and around 1 percent error when noise is added. A two degree-of-freedom balance was designed to model a lift-pitch system and test the methods experimentally. A static calibration was conducted by hanging weights, measuring voltages reported by strain gauges, and computing calibration coefficients. Dynamic forces were simulated with a modal impact hammer. Velocity data was found by numerically integrating the acceleration data. The dynamic force reported by the modal impact hammer along with the position, velocity, and acceleration data was used to determine the calibration coefficients. These coefficients were used to reconstruct a static force, which was simulated by cutting string attached to static weights.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to acknowledge Dr. Devin E. Burns, my NASA intern mentor. He was the liaison between NASA and my thesis research. Dr. Burns is extremely knowledgeable on all matters related to balance design, and his expertise continues to amaze me over the year I have worked with him. On multiple occasions, Dr. Burns answered a question or solved a problem I had before I even finished asking for his help. I did not know how complex and difficult balances can be until I started reading the literature. There are hundreds of pages on mechanics to dive into the 'simple' designs. Despite the complexity, Dr. Burns was always able to massively simplify this process so that I could design my own balance. He is one of the most patient teachers I have ever had. I can, unfortunately, still vividly remember some obvious questions I've asked him. He always managed to find a way to take a step back and explain an already simple concept in an even simpler manner for me to understand. I have been extremely fortunate to have him as a mentor, and I hope to one day be as knowledgeable as him while being as patient to those who aren't.

I would like to acknowledge Dr. Nicholas A. Vlajic, my thesis supervisor. Dr. Vlajic worked with Dr. Burns and me, and facilitated my lab work in Penn State's Applied Research Laboratory. Dr. Vlajic has an extensive understanding of dynamics concepts that felt a little bit like magic to me. He has a clear passion, and talent, for explaining these concepts in the perfect amount of detail. If I ever had a question related to dynamics, whether it was directly thesis related, homework related, or just for curiosity sake, Dr. Vlajic would take time out of his day to enthusiastically break concepts down for my benefit. He has gone to incredible lengths to help me learn, set me up with writing and experiments, and help me succeed. On days that I could tell he was busy, or stressed, he would still devote time to help me with whatever it was I needed. Dr. Vlajic cares immensely for all of the students that he advises, and is the type of advisor/teacher that every student hopes to get. I was lucky to have the opportunity to work with him on this project, and I hope to be at a similar knowledge level and be as generous of a teacher as him someday.

I would like to acknowledge Mr. Keith Rice for creating a software that simultaneously read the strain data from the balance, the accelerometers, and the hammer impulse.

Federal funds from NASA were used to commission Ahmic Aerospace to create a balance. The findings and conclusions of this paper do not necessarily reflect the views of NASA.

Chapter 1

Background and Motivation

1.1 Research Motivation

Accurately measuring dynamic forces is of interest to ground testing cases. Some examples include measuring aerodynamic loads in short-duration facilities, such as shock and impulse wind tunnels, measuring aerodynamic loads in a wind tunnel when simulating a wind gust, and measuring aerodynamic loads acting on a model with oscillating lifting surfaces [1]. These examples can occur in a wind tunnel, which will need to be calibrated to measure dynamic forces instead of just static. Another application to dynamic forces has come forward with the advancement of hypersonic wind tunnels. Low speed wind tunnels can have long measurement periods which allow for any oscillations to be damped, meaning that modeling these forces as static is sufficient [2]. Hypersonic wind tunnels must conduct tests in short, high-speed bursts which do not allow for the damping of the oscillations of attached structures [3]. The force F is dynamic and changes with time. The measurement system is modelled as a mass-spring-damper where x is the displacement. The voltage output of the system, which indicates the force, is directly proportional to x. The commonly accepted equation for a one degree-of-freedom mass-spring-damper system is stated as:

$$F = m\ddot{x} + c\dot{x} + kx \tag{1.1}$$

where F is the force being measured, m is the mass of the model, c is the model's damping coefficient, k is the stiffness coefficient, and x is the position of the model. This one degree-of-freedom mass-spring-damper system can be used to demonstrate the importance of dynamic effects to force measurement systems [4]. When measuring a static force, the time derivative components can be assumed to be zero. This is done by using static force transducers, which only use the stiffness coefficient to reconstruct a force. When a dynamic force is applied, and only statically calibrated force transducers are used to measure it, the inertial and damping components introduce errors. This leads to an improperly reconstructed force [5]. An example of a static reconstruction of a step force on a two degree-of-freedom system can be seen in Figure 1.1.

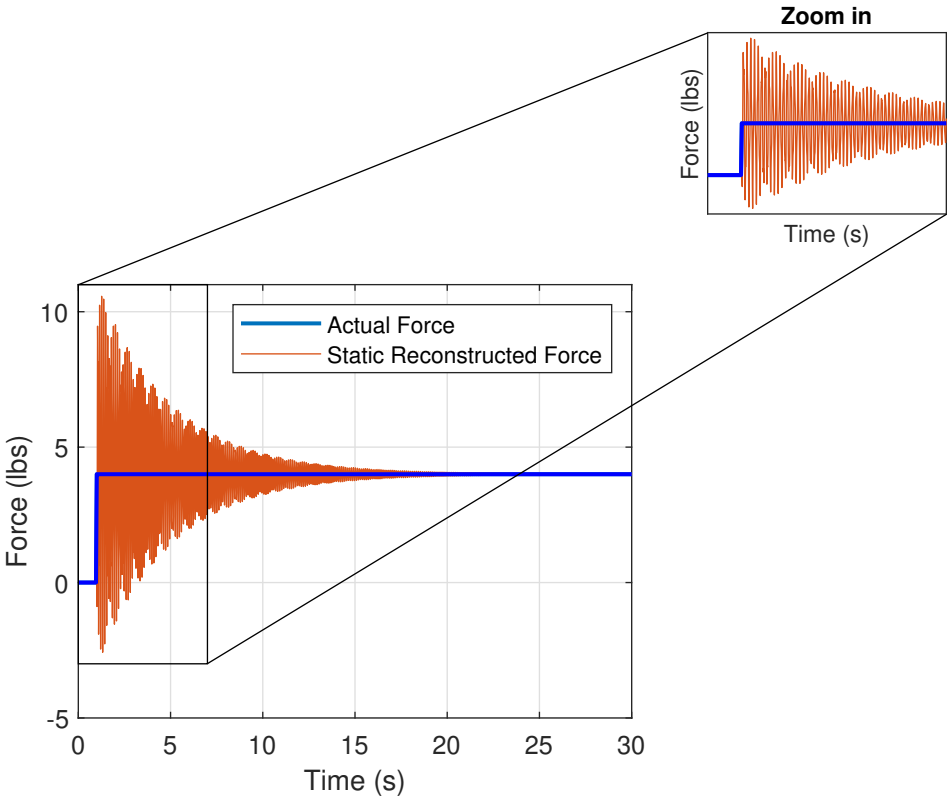

Figure 1.1: Static reconstruction of a two degree-of-freedom dynamic force

This simple example illustrates that using static force transducers to measure dynamic forces can yield inaccurate results. Dynamic forces can be present in wind tunnels, so the measurement methods need to be calibrated accordingly. The purpose of this research is to discuss methods of dynamic force reconstruction and present techniques to account for inertial and damping components.

1.2 Background Information

1.2.1 Static Calibration

Static forces are the simplest to measure, and can be modeled with first dimension Hooke's law:

$$F = kx \tag{1.2}$$

Where F is the force, k is the stiffness coefficient, and x is the displacement. This equation is used to model the wind tunnel measurement system by using k as a calibration coefficient and x as the voltage reported by static force transducers [6]. In order to reconstruct forces, the calibration matrix must be known. To determine the calibration matrix, objects with a known weight are hung from the test article. The force transducers report a voltage for each weight hung, and the calibration

matrix can be calculated. When a new object with an unknown weight is hung, the force can be reconstructed with the force transducer's voltages and the previously determined static calibration matrix.

1.2.2 Two Degree-of-Freedom Model

This paper uses a two degree-of-freedom representation of a wind tunnel model system proposed by Beuhrle [7]. This can be seen in Figure 1.2.

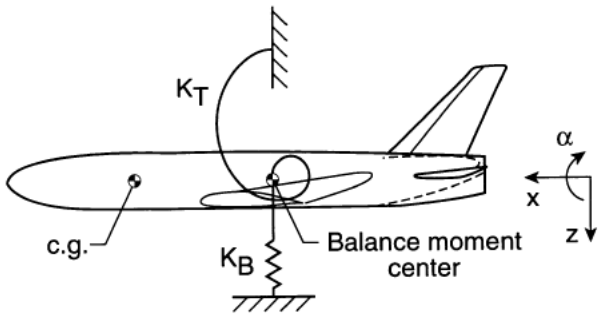

Figure 1.2: Two degree-of-freedom model [7].

where the z and α dimensions are the degrees of freedom. K_T represents the stiffness of the torsional spring in the α direction, and K_B represents the stiffness of the spring in the z direction. The full equations of motion are:

$$\begin{bmatrix} m_m & -m_m d_{cg/bc} \\ -m_m d_{cg/bc} & I_{yBC} \end{bmatrix} \begin{bmatrix} \ddot{z} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} K_B & 0 \\ 0 & K_T \end{bmatrix} \begin{bmatrix} z \\ \alpha \end{bmatrix} = \begin{bmatrix} -1 \\ d_{cg/bc} \end{bmatrix} F(t) \tag{1.3}$$

where $d_{cg/bc}$ is the distance from model mass center of gravity to model balance center, I_{yBC} is the inertia about the y-axis for a reference at the model balance center, m_m is the model mass, and $F(t)$ is the applied dynamic force.

For the purposes of this paper, the matrices are referred to as the inertial calibration matrix and the static calibration matrix. The equations of motion can be rewritten as:

$$C_a \begin{bmatrix} \ddot{z} \\ \ddot{\alpha} \end{bmatrix} + C_x \begin{bmatrix} z \\ \alpha \end{bmatrix} = \begin{bmatrix} -1 \\ d_{cg/bc} \end{bmatrix} F(t) \tag{1.4}$$

where C_a is the inertial calibration matrix and C_x is the static calibration matrix. The model does not assume any damping, but it can be added in:

$$C_a \begin{bmatrix} \ddot{z} \\ \ddot{\alpha} \end{bmatrix} + C_v \begin{bmatrix} \dot{z} \\ \dot{\alpha} \end{bmatrix} + C_x \begin{bmatrix} z \\ \alpha \end{bmatrix} = \begin{bmatrix} -1 \\ d_{cg/bc} \end{bmatrix} F(t) \tag{1.5}$$

$$C_v = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \tag{1.6}$$

where d_1 and d_2 are chosen values for the damping coefficients. Representative values for a realistic wind tunnel model system are presented in Table 1.1.

m_m	0.3313 pound-second^2/inch
I_{yBC}	19.51 inch-pound-second^2
$d_{cg/bc}$	5 inches
K_B	1308 pound/inch
K_T	277089 inch-pound
d_1	0.2 pound-second/inch
d_2	1 pound-second/inch

Table 1.1: Values used in calibration model

The model presented assumes values for all three coefficient matrices. The purpose of the values is to generate data that would otherwise be found experimentally, and then to use that data to test the methods theoretically.

1.3 Previous Work

The Sum of Weighted Accelerations Technique, abbreviated SWAT, uses acceleration data to compensate for the inaccuracies caused by dynamic forces [8]. The technique begins with a static calibration under static conditions, and then uses acceleration measurements and modal characteristics of the structure. An updated method to the SWAT technique by Draper [9] uses accelerometers to simplify the dynamic calibration process. This is accomplished by attaching accelerometers to the model. It is done for a multi-component balance, but for the sake of simplicity, the method will be summarized for one dimension. A known dynamic load is applied to the test article and the data is processed in MATLAB. The damping component is assumed to be zero, with the force being a sum of the inertial and static components:

$$F_{inertial} + F_{static} = m\ddot{x} + kx \tag{1.7}$$

The detrend function is used to filter out the static components. The force without the static portion is now known:

$$F_{inertial} = m\ddot{x} \tag{1.8}$$

This leaves the inertial force since damping is assumed to be zero. The inertial calibration matrix can then be found using the force and accelerometer data:

$$\frac{F_{inertial}}{\ddot{x}} = m \tag{1.9}$$

where $F_{inertial}$ represents the force with the static component filtered out, m represents the inertial calibration matrix, and $\ddot{x}$ represents the acceleration. The static k matrix can be found by using the same model, but subjecting it to a purely static load.

1.4 Goals and Objectives

The primary goal of this paper will be to derive the two degree-of-freedom formulation for the acceleration compensation techniques presented in Section 1.3. The two degree-of-freedom system will be pitch and lift. Damping will be included in this formulation and will be compensated for. The methodology will be applied on a lab setup to explore practical considerations when implementing experimentally.

Chapter 2

Dynamic Force Reconstructions

Two weighted acceleration methodologies were explored to utilize accelerometers in dynamic force calibration, and to account for the effects of damping. The methodologies are explained mathematically with the two degree-of-freedom wind tunnel model and applied dynamic forces. The first method involves finding displacement, velocity, and acceleration data for a known dynamic force and using that data to find a calibration matrix. The second method involves doing a static calibration, removing the static component of a dynamic force, and then solving for the dynamic calibration matrices [10].

2.1 Simultaneously identify the static, velocity, and inertial weighting coefficient for lift-pitch

All three calibration matrices can be found with accelerometers and strain gauges attached to the model. Data should be taken for a force applied at some distance from the point at which the forces cause rotation, otherwise known as the moment center. A second set of data should be taken at a different distance from the moment center. The accelerometer data can be numerically integrated to find velocity data. An S matrix can be made for each of the two test cases:

$$S_1 = \begin{bmatrix} x_{1,1} & v_{1,1} & a_{1,1} & x_{2,1} & v_{2,1} & a_{2,1} \\ x_{1,2} & v_{1,2} & a_{1,2} & x_{2,2} & v_{2,2} & a_{2,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1,n} & v_{1,n} & a_{1,n} & x_{2,n} & v_{2,n} & a_{2,n} \end{bmatrix} \tag{2.1}$$

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{2.2}$$

where the first subscript denotes the degree-of-freedom, and the second subscript denotes the data point number.

The Q matrix can be assembled from the force in both degrees-of-freedom for both cases:

$$Q_1 = \begin{bmatrix} -F_1 \\ F_1 d_1 \end{bmatrix} = \begin{bmatrix} Normal\ Force \\ Pitching\ Moment \end{bmatrix} \tag{2.3}$$

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{2.4}$$

where F is the known applied force, and d is the distance between the force and the moment center. Now, the calibration matrix can be calculated:

$$C = (S^T S)^{-1} S^T Q \tag{2.5}$$

The known force can be recalculated by using the developed calibration matrix and one of the previous S matrices:

$$Q_1 = S_1 C \tag{2.6}$$

This can be seen in Figure 2.1 by using a Gaussian pulse force to develop a calibration matrix, and then using the same calibration matrix to redevelop the same force.

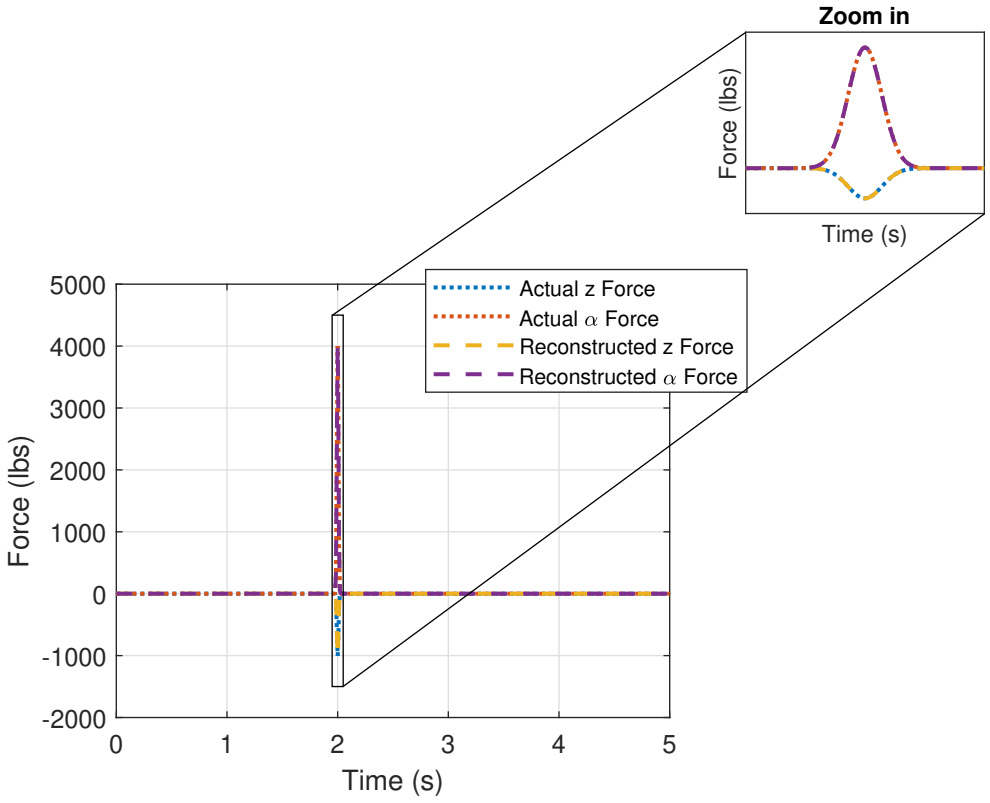

Figure 2.1: Reconstruction of a Gaussian force by simultaneous identification

The Gaussian force was used to generate a calibration matrix, and that same calibration matrix was used to reconstruct the same Gaussian force. Now, an unknown force generates displacement, velocity, and acceleration data. The same calibration matrix can be used with a different S matrix to identify the unknown force. Using the calibration matrix developed from the Gaussian pulse force, a step force can be reconstructed as seen in Figure 2.2.

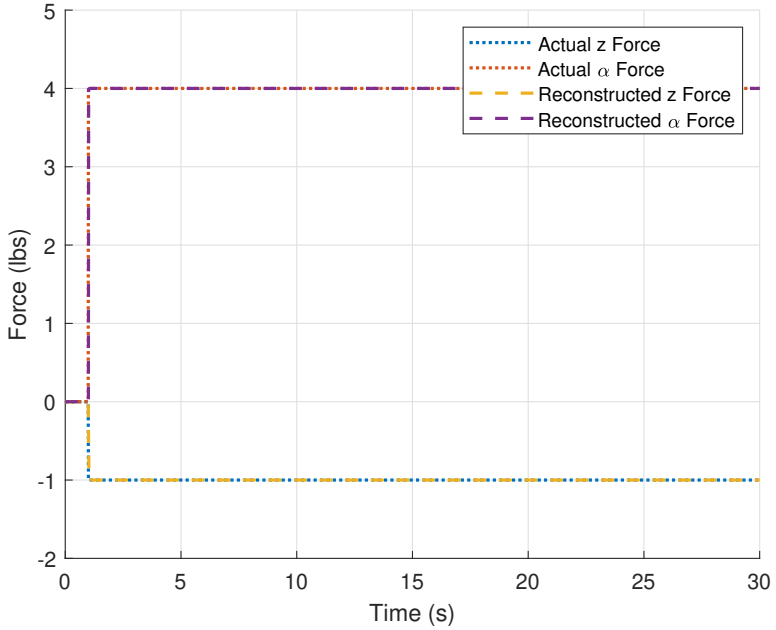

Figure 2.2: Reconstruction of a step force by simultaneous identification

The error between the reconstructed force and the actual force can be seen in Figure 2.3.

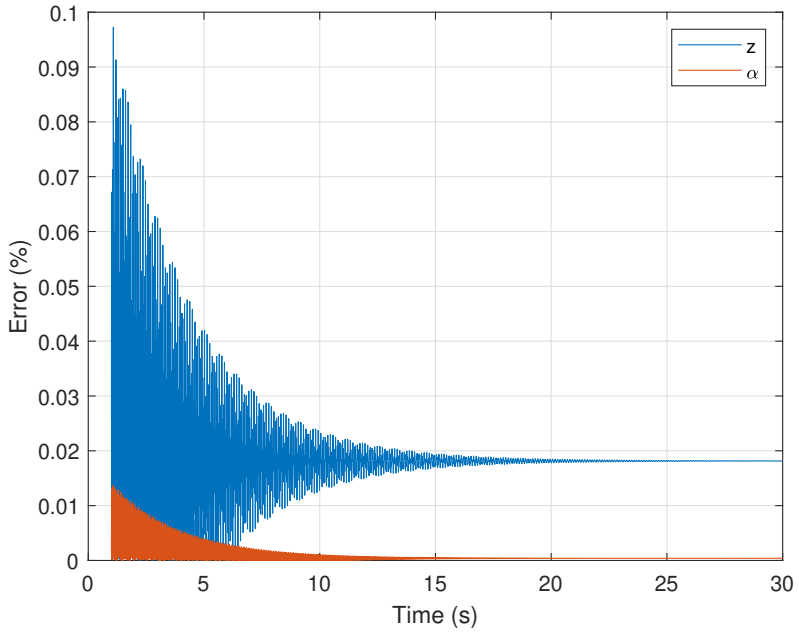

Figure 2.3: Step force reconstruction error from simultaneous identification

2.2 Identify coefficients for lift-pitch in a two-step process

The two step process starts with a static force and a static calibration. When the the static calibration matrix is known, a dynamic force can be applied to determine the velocity and acceleration coefficients. The accelerometer data can be numerically integrated to find the velocity data. The S matrix can be assembled without the static component:

$$S_1 = \begin{bmatrix} v_{1,1} & a_{1,1} & v_{2,1} & a_{2,1} \\ v_{1,2} & a_{1,2} & v_{2,2} & a_{2,2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1,n} & a_{1,n} & v_{2,n} & a_{2,n} \end{bmatrix} \tag{2.7}$$

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{2.8}$$

The Q matrix can be assembled similarly to the previous method:

$$Q_1 = \begin{bmatrix} -F_1 \\ F_1 d_1 \end{bmatrix} = \begin{bmatrix} Normal\ Force \\ Pitching\ Moment \end{bmatrix} \tag{2.9}$$

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{2.10}$$

The calibration can be computed by subtracting the static force from the total force, which leaves the velocity and acceleration components:

$$C_{va} = (S^T S)^{-1} S^T (Q - C_x x) \tag{2.11}$$

where x is the position data for the dynamic force case, C_{va} is the calibration matrix for the velocity and acceleration terms, and C_x is the calibration matrix from the static calibration. The same Gaussian pulse force can be recalculated with one of the previous S matrices:

$$Q_1 = S_1 C_{va} + x_1 C_x \tag{2.12}$$

This will reconstruct the same force, seen in Figure 2.4.

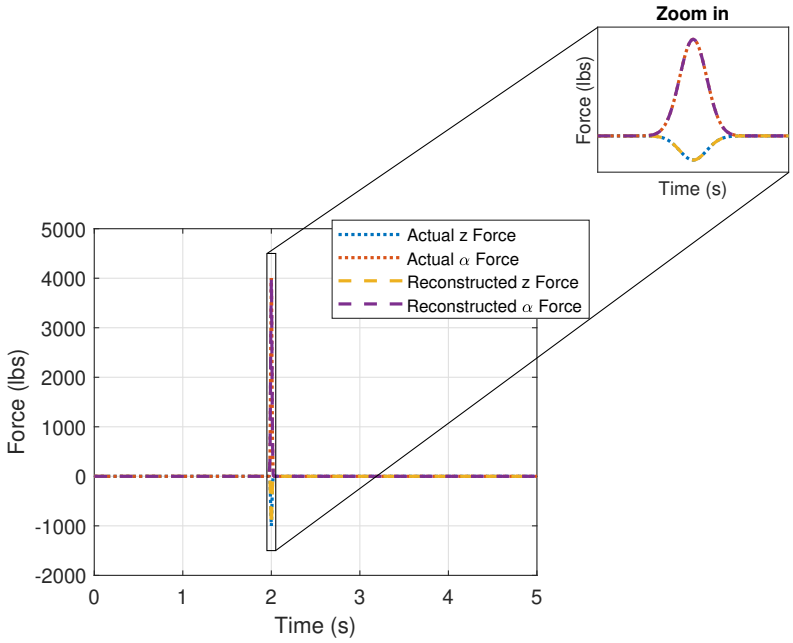

Figure 2.4: Reconstruction of a Gaussian force by two step method

The same calibration matrix can be used with a different S matrix, along with a new set of position data, to reconstruct a different step force seen in Figure 2.5.

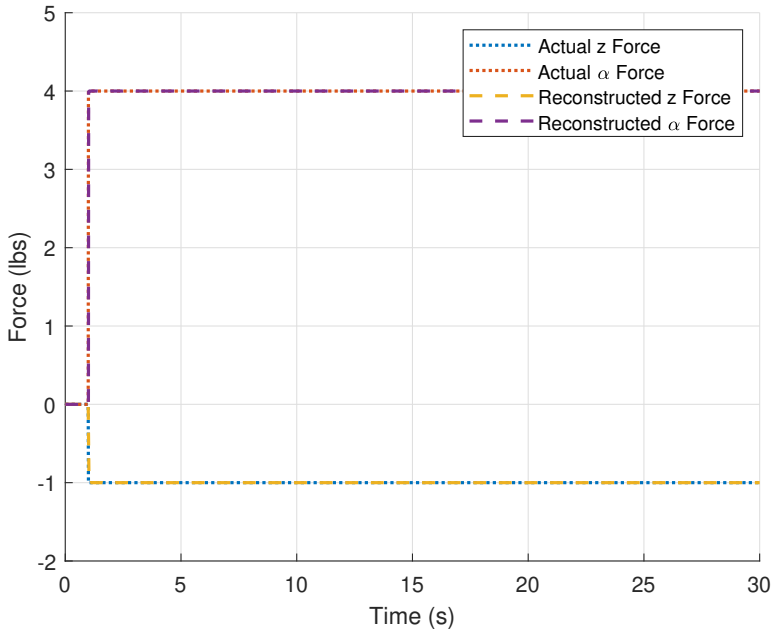

Figure 2.5: Reconstruction of a step force by two step method

The error between the reconstructed force and the actual force can be seen in Figure 2.6.

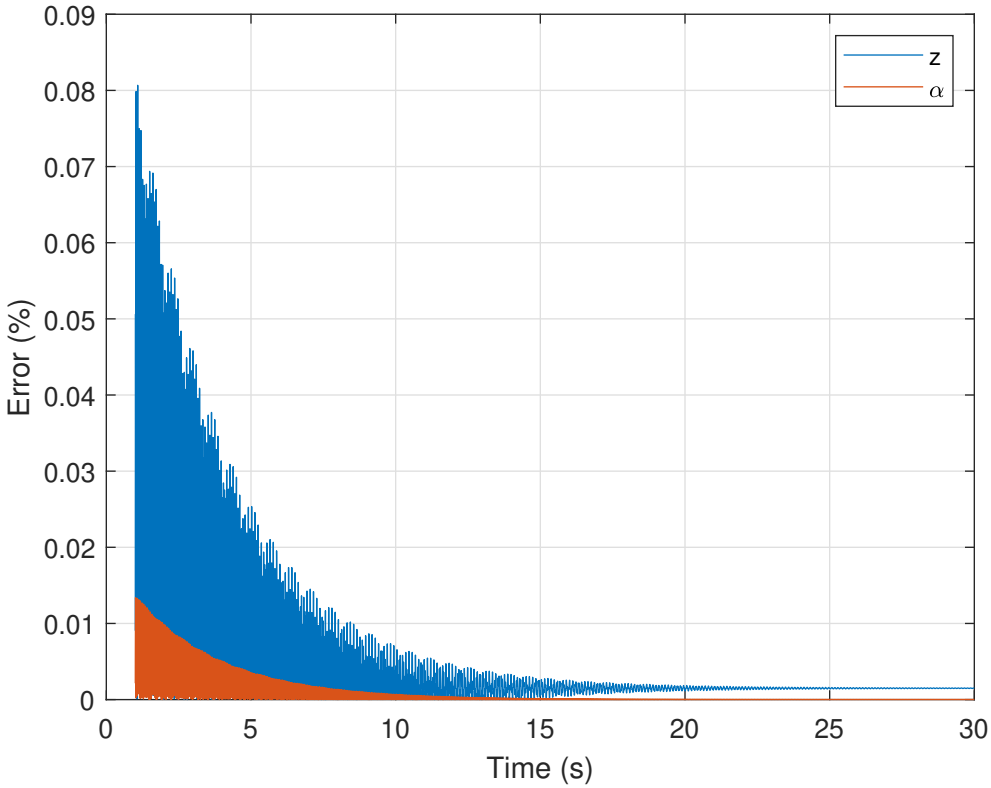

Figure 2.6: Step force reconstruction error from two step method

2.3 Adding noise to the two-step process

While both methods can yield near-zero errors, the current model assumes perfect data points. By using MATLAB's rand function, noise can be added to every data point. The process is repeated with a random number added to position and acceleration. After reconstructing the same Gaussian force applied, the following can be seen in Figure 2.7.

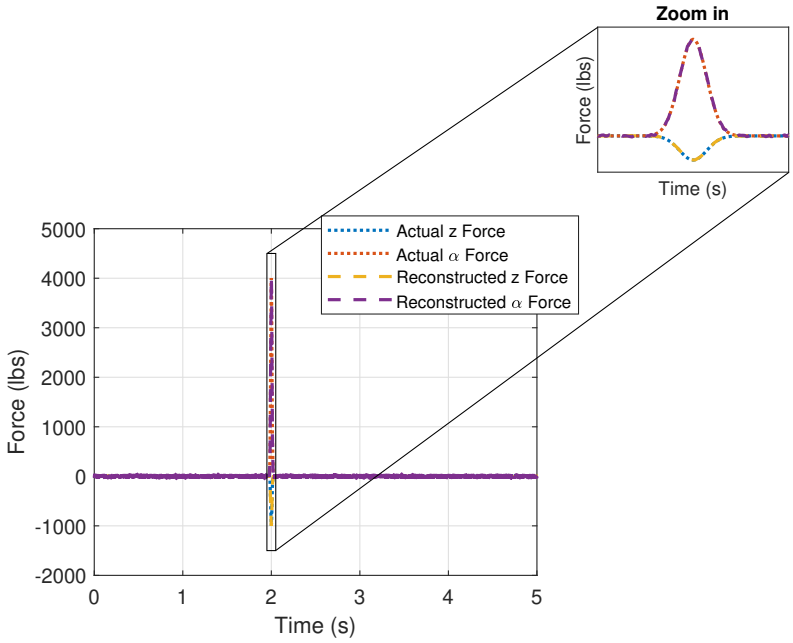

Figure 2.7: Noisy reconstruction of a Gaussian force by two step method

When the calibration matrix is used to calculate a different force, the following can be seen in Figure 2.8.

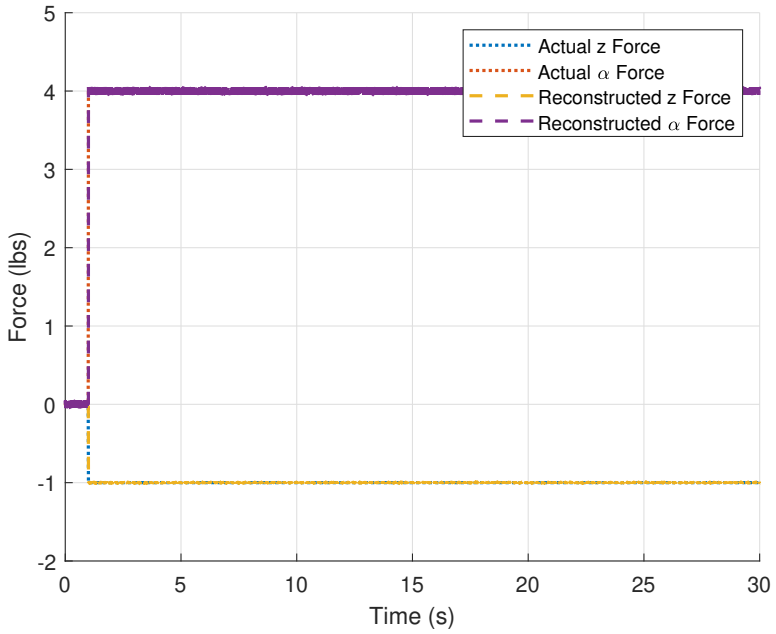

Figure 2.8: Noisy reconstruction of a step force by two step method

The error can be seen in Figure 2.9.

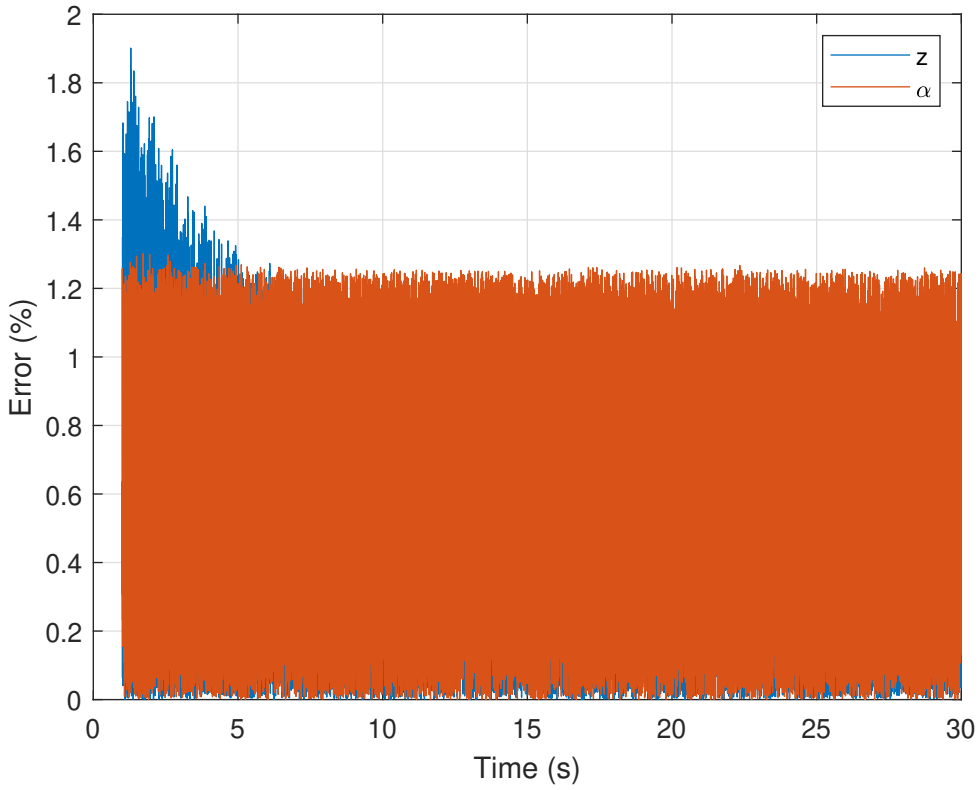

Figure 2.9: Noisy step force reconstruction error from two step method

The calculations done within computation precision gave negligible error. After introducing error to the data points, the error is no longer negligible.

Chapter 3

Balance Design Considerations

A two component (normal force and pitching moment) strain gauge balance was designed to test the dynamic force reconstruction methodologies. The balance was designed so that it could mount to a rigid mounting block on the non-metric end to minimize additional sources of vibration beyond the balance. The mounting block chosen was a Thorlabs AP90RL. It was also designed to support a block, a Thorlabs TS240 tombstone, which is where accelerometers and weights were attached. Two lands on the balance were designed to have maximum stress, which is where the strain gauges were placed. In order for the strain gauges to be able to measure accurate strain values, the balance was designed to exhibit 250 microstrain, not exceed 750 microstrain, and have a strain difference of approximately 500-1000 microstrain with both 45 N of normal force and 2.8 N-m of pitching moment applied [11]. The lower limit was established to ensure a noticeable value can be produced, and the upper limit was introduced to ensure that the strain gauges won't be damaged. These conditions must hold true for strains caused by pure bending. The strain can be computed as:

$$\epsilon = \frac{My}{IE} \tag{3.1}$$

where ϵ is the strain, M is the internal bending moment around the neutral axis, y is the perpendicular distance from the neutral axis, I is the moment of inertia, and E is the modulus of elasticity. In order to achieve the desired values, the applied force and length, cross sectional geometry, and modulus of elasticity were modified. Originally, the balance and tombstone were going to be cantilevered in series, but it was found that it is mathematically impossible to reach the desired strain values no matter how the constraints are changed. The solution was to drill a hole in the tombstone and have the balance cantilevered while inside the tombstone. This can be seen in Figure 3.1.

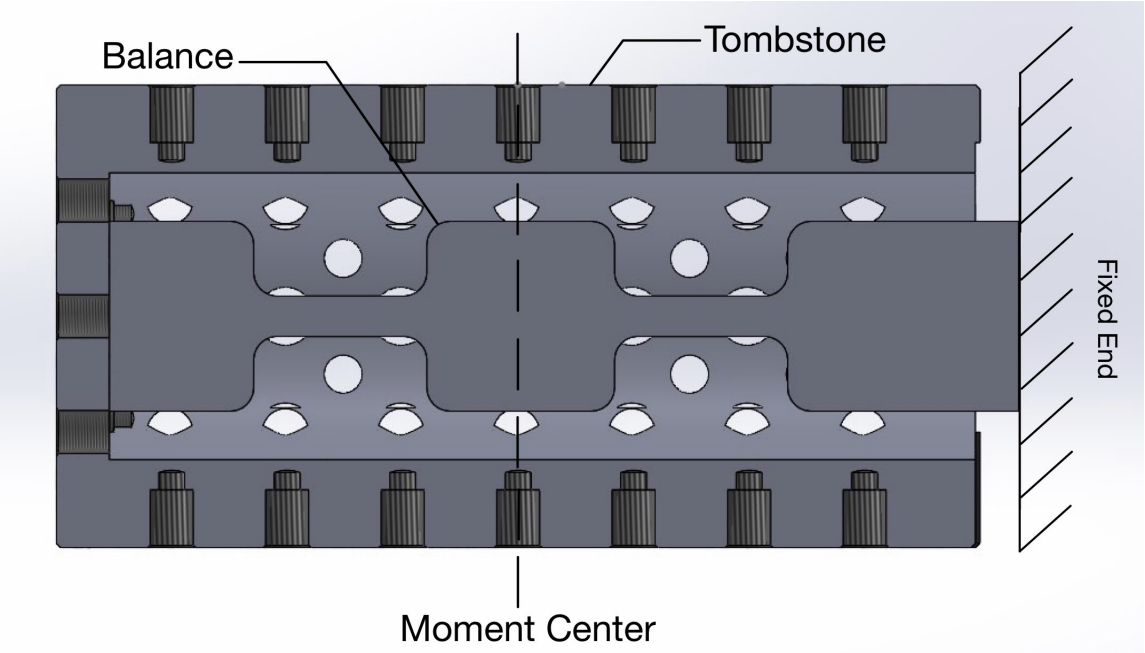

Figure 3.1: Cut view of balance inside tombstone

With the balance inside the tombstone, the force can now be applied to the moment center. Under these conditions, aluminum 7075-T6 was chosen for a modulus of elasticity of $10.4 * 10^6$ psi. The cross section was designed to be 0.75 by 0.175 inches. An image of the balance can be seen in Figure 3.2.

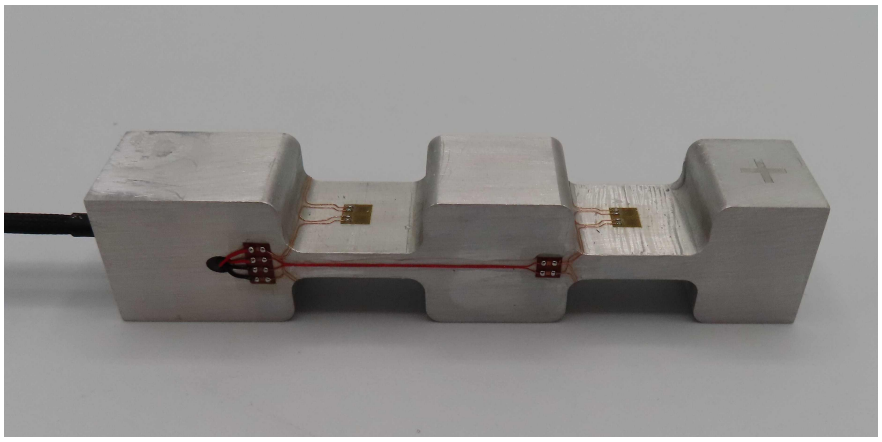

Figure 3.2: Photo of completed balance

The strain gauges selected were N5K-S5054K 5000 ohm transducer class. The balance was designed to be longer on one end so that it would extrude out of the tombstone. This is shown in Figure 3.3.

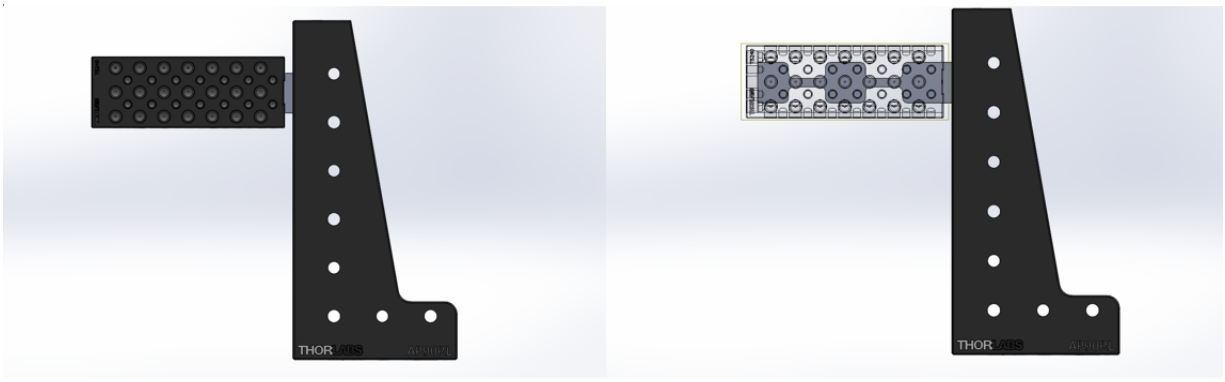

Figure 3.3: Balance extrusion (left) and balance extrusion with transparent tombstone (right)

With the balance inside the tombstone, referred to as the test article, static weights can be hung off the test article to conduct a static calibration. When the static calibration is concluded, the static weights should be removed. Accelerometers can be attached to the tombstone for the dynamic calibration. A picture of the setup can be seen in Figure 3.4.

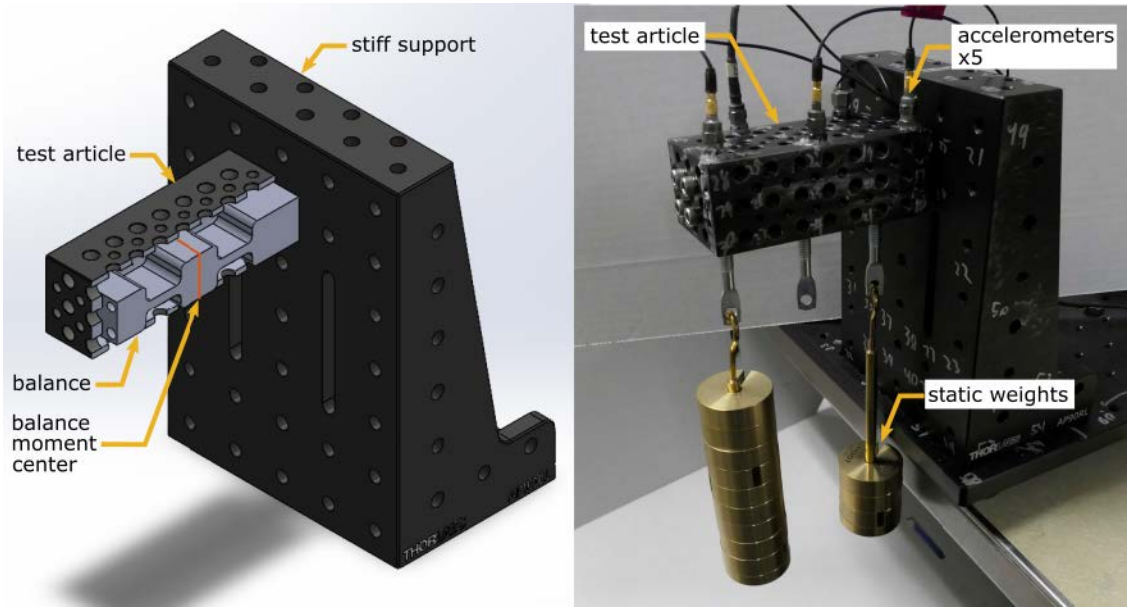

Figure 3.4: Model of the test article with cut view of tombstone (left) and lab view of full tombstone with static weights (right)

Experimental test points were mapped out on the test article. These were points for the static calibration, points for the accelerometers to be mounted, and points to tap with a modal impact hammer. These points, and a computational mesh used to map a modal analysis, can be seen in Figure 3.5.

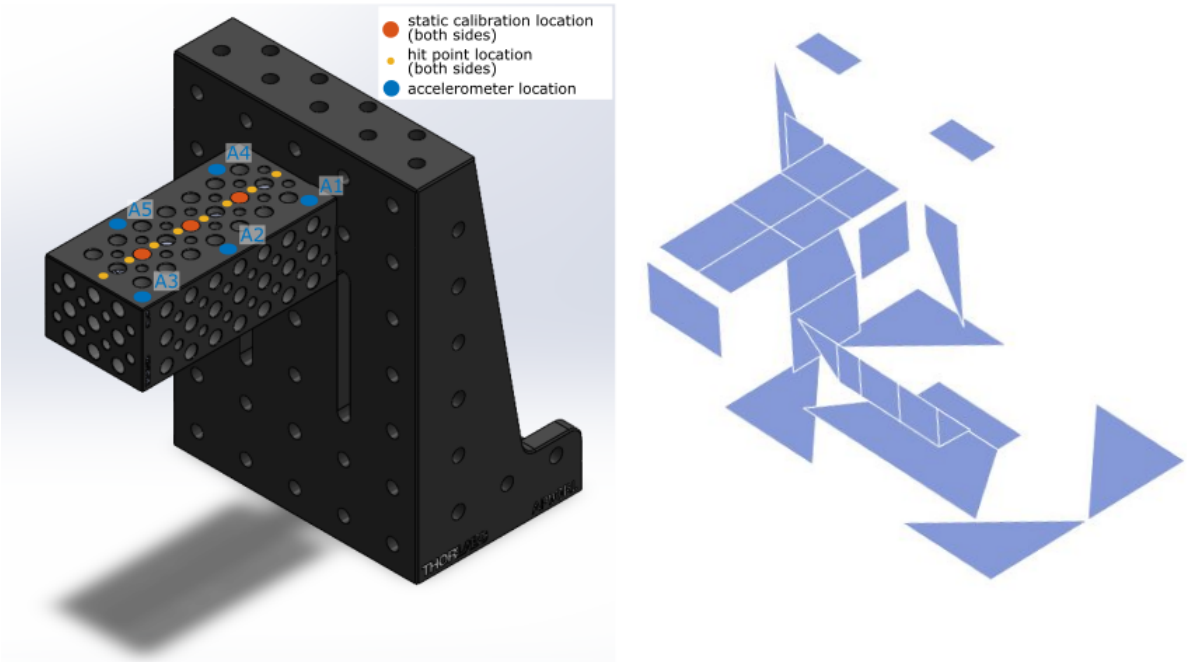

Figure 3.5: Test article map of important points (left) and mesh of modal analysis (right)

Chapter 4

Experimental Results

4.1 Modal Analysis

A modal analysis was performed on the test article and the fixture to determine response modes and natural frequencies. Analyzing the fixture allowed for modes of the supporting structure to be determined. Before analyzing the fixture and test article, the balance and tombstone were individually analyzed to verify rigidity. When the fixture and text article were tested, the mesh from Figure 3.5 had 68 hit points. A roving hammer modal analysis was used. A roving hammer modal analysis uses accelerometers as drive points and hammer hit locations as the response points, and it was used because the test article is relatively small and it helps to cut down on the number of sensors used. The modes of interest occurred at frequencies 95 Hz, 132 Hz, 202 Hz, 293 Hz, and 367 Hz. The 95 Hz frequency corresponds to the motion of the support fixture. The 132 Hz frequency relates to the pitching motion of the test article. The 202 Hz frequency is the lift motion of the test article. Since the modal analysis considered points that were not symmetric, the 293 Hz frequency corresponds to roll and the 367 Hz frequency corresponds to yaw. This can be seen in Figure 4.1.

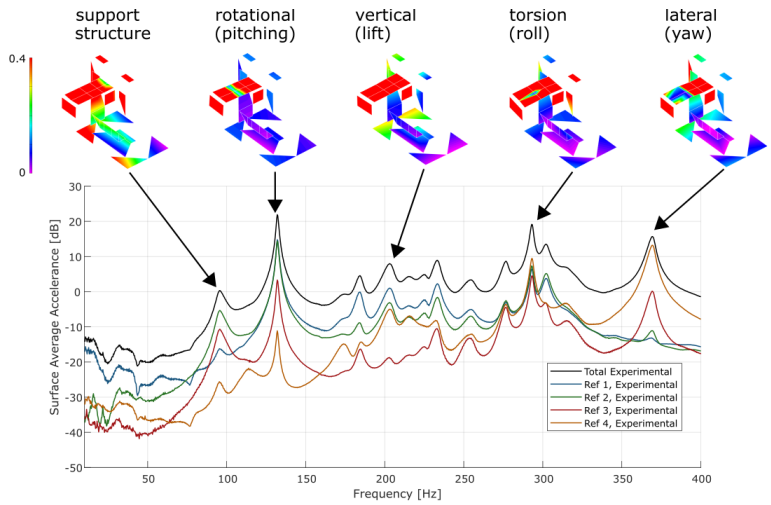

Figure 4.1: Surface average acceleration frequency response

4.2 Static Calibration

A static calibration was conducted by hanging weights from the tombstone at three different positions. The first position was located on the moment center, and the other two were located one inch forward and backward from the moment center, as defined in Figure 3.5. To apply a pure normal force, static weights were hung from the point lying on the moment center as seen in Figure 4.2.

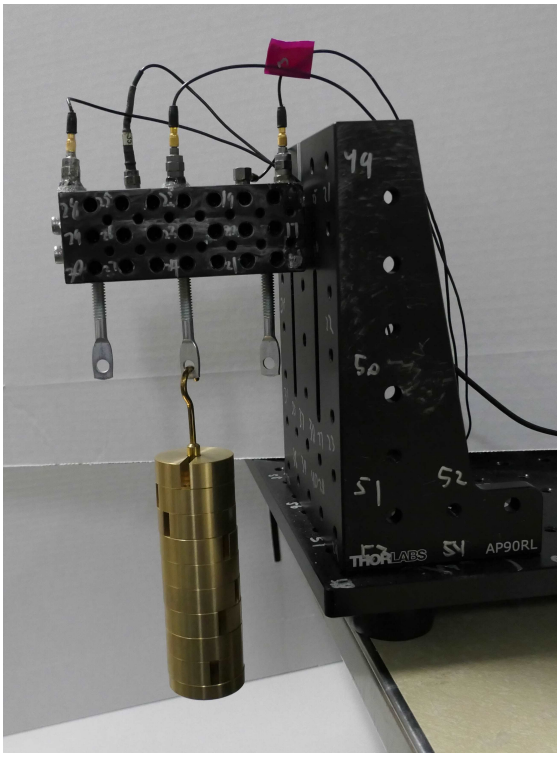

Figure 4.2: Experimental setup of test article in normal force configuration

Similarly, to apply a pitching moment combined with a normal force, weights were hung from the points offset from the moment center as seen in Figure 4.3.

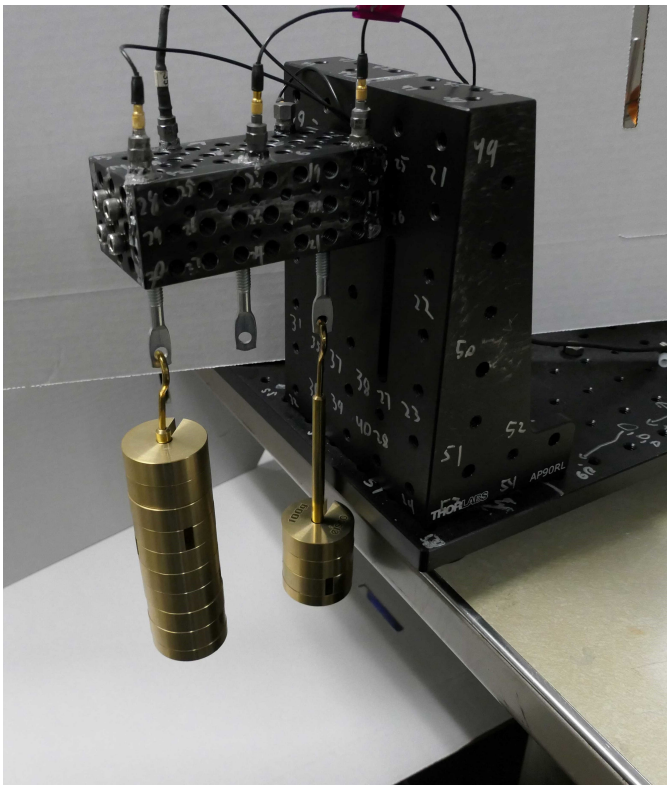

Figure 4.3: Experimental setup of test article in normal force and pitching moment configuration

For each test run, strain data was taken at the zero point. Then, the weight was increased by 200 grams until the maximum of 1000 grams was reached. Then, 200 grams were incrementally removed and a second zero data point was taken. Test runs were conducted for all three static calibration locations individually, and the first and third static calibration location simultaneously. The test article was removed from its fixture, flipped upside-down, and the test runs were repeated. The balance output values in units of mV/V at each land. To convert these to values of strain related to normal force and pitching moment for the balance design, the following equations can be used:

$$NF_i = V_{1i} + V_{2i} \tag{4.1}$$

$$PM_i = V_{1i} - V_{2i} \tag{4.2}$$

where i is the test run, V_1 and V_2 are the output values from balance land 1 and 2, respectively, and NF and PM are the mV/V outputs related to the normal force and pitching moment, respectively. Since the test article has an intrinsic mass, the balance does not read zero mV/V when there is no static weight applied. Therefore, the zero output must be subtracted from every data point to calculate the offset from zero weight. This will ensure that the data will correctly read zero output when no weight is applied, and only register the output due to the additional mass on the system:

$$\Delta NF_i = NF_i - NF_0 \tag{4.3}$$

$$\Delta PM_i = PM_i - PM_0 \tag{4.4}$$

where 0 indicates the output when no mass is applied. With the data that is zeroed out, the static calibration can be conducted by defining:

$$x = \begin{bmatrix} \Delta NF \\ \Delta PM \end{bmatrix} \tag{4.5}$$

and

$$F = \begin{bmatrix} Applied\ Normal\ Force \\ Applied\ Pitching\ Moment \end{bmatrix} \tag{4.6}$$

and then solving for the static coefficient matrix:

$$C_{static} = (x^T x)^{-1} x^T (F) \tag{4.7}$$

With the static coefficient matrix and the ΔNF and ΔPM outputs, the reconstructed force can be computed and seen in Figure 4.4.

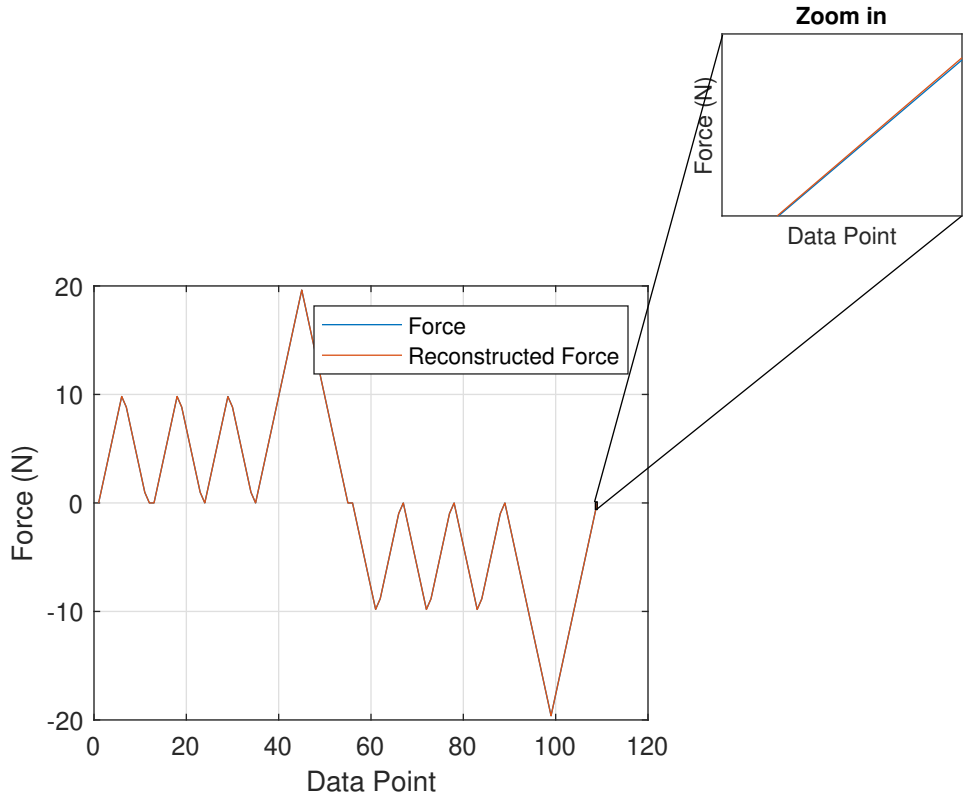

Figure 4.4: Experimental static calibration results

The difference between the measured force and the reconstructed force, as a percent of the maximum applied value, can be seen for the normal force in Figure 4.5.

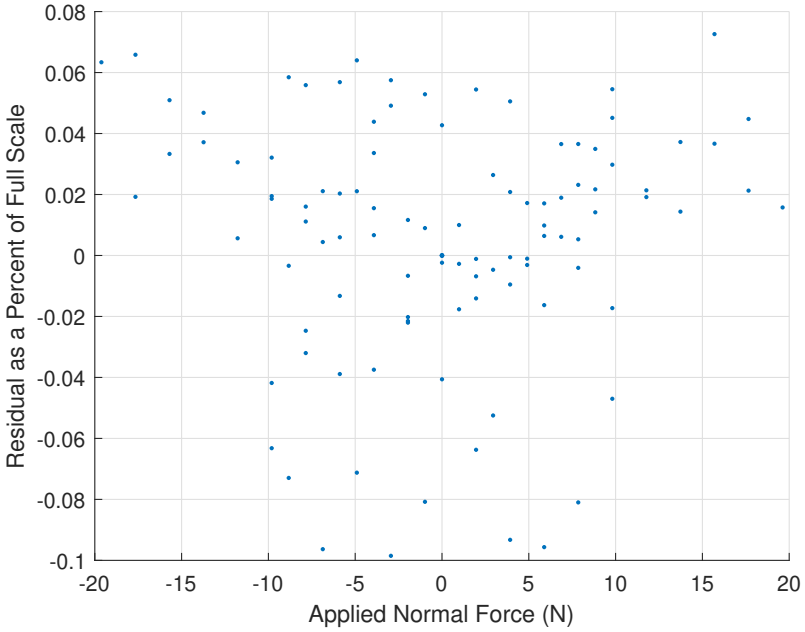

Figure 4.5: Residual force due to normal force

A similar graph can be seen for the pitching moment in Figure 4.6.

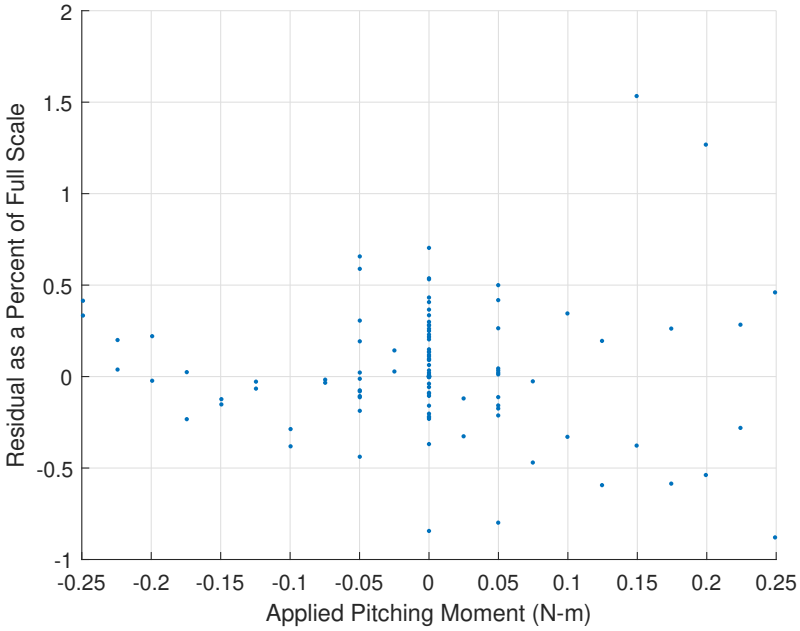

Figure 4.6: Residual force due to pitching moment

The normal force resulted in very high accuracy, while the pitching moment resulted in a

noticeably worse accuracy. The balance yielded accurate results for the normal force output and reconstruction, so the problem likely lies with how the actual pitching moment was measured. The normal force did not require a distance measurement, while the pitching moment did. The tolerances in the hole location and diameters of the tombstone could be the cause of the actual pitching moment being measured incorrectly. Using a longer test article with pitching moment being applied further away from the moment center would reduce this inaccuracy. To ensure the accuracy of the balance, the ΔNF and ΔPM outputs were plotted against the applied normal force and pitching moments. This can be seen in Figure 4.7.

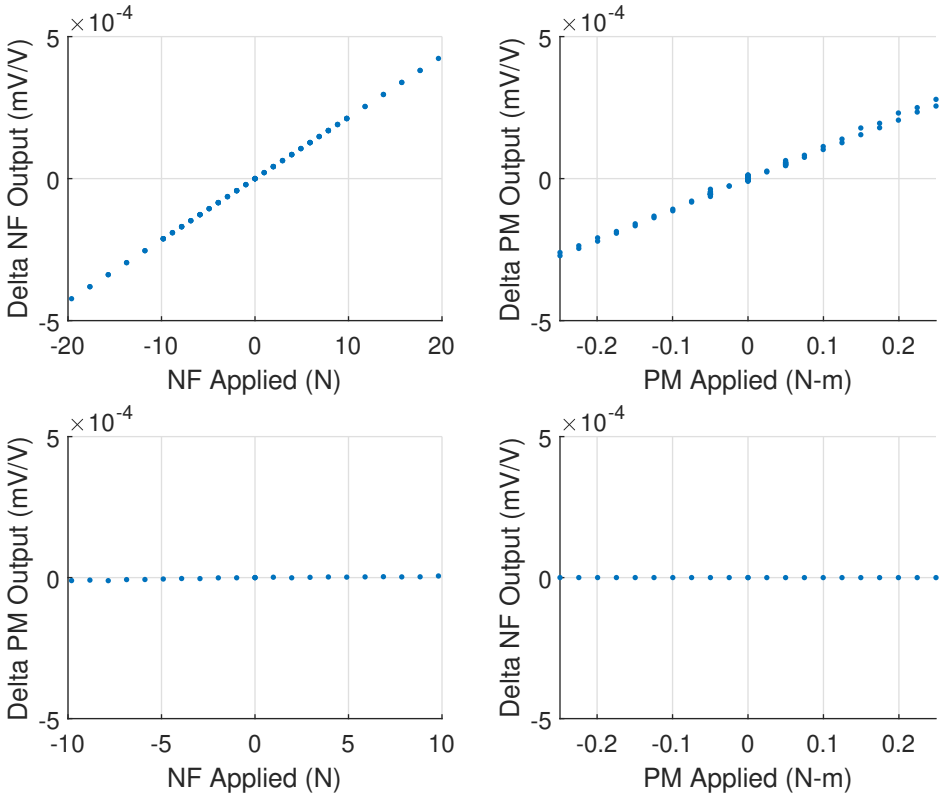

Figure 4.7: Different static responses from different inputs

The figure demonstrates that, as expected, applying a normal force will linearly increase the balance's normal force output while keeping the pitching moment output negligible. Likewise, applying a pitching moment will linearly increase the balance's pitching moment output while keeping the normal force output negligible.

4.3 Dynamic Calibration

Five accelerometers were mounted on the test article. Their locations are shown in Figure 3.5. They have a nominal sensitivity of 100 mV/g. The data was collected using a NI-XXX with time synchronization of 10 kHz for 5 seconds. A modal impact hammer hit eight points on the

top of the test article, and the same eight points on the bottom of the test article as shown in Figure 3.5. This was repeated across five tests. The impact hammer simulated a Gaussian impulse, although there were double hits present. The double hits were not an issue with this experiment, since the dynamic calibration coefficients were able to take the second peaks into account. When the previously defined static calibration coefficient matrix was imported, the dynamic calibration coefficient matrix was generated five times, once for each test. The final matrix was an average of the five matrices. The normal force result of a Gaussian force reconstructed experimentally from the combination of the static and dynamic coefficient matrices can be seen in Figure 4.8.

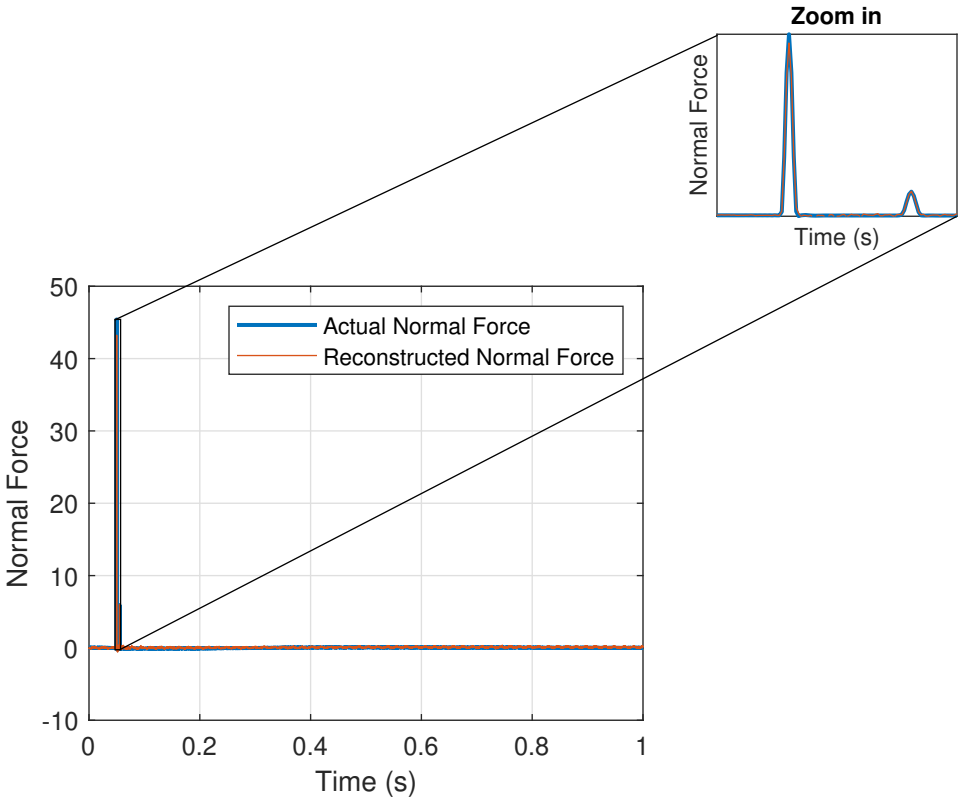

Figure 4.8: Experimental dynamic normal force reconstructed

Likewise, the pitching moment result of a Gaussian force reconstructed experimentally from the combination of the static and dynamic coefficient matrices can be seen in Figure 4.9.

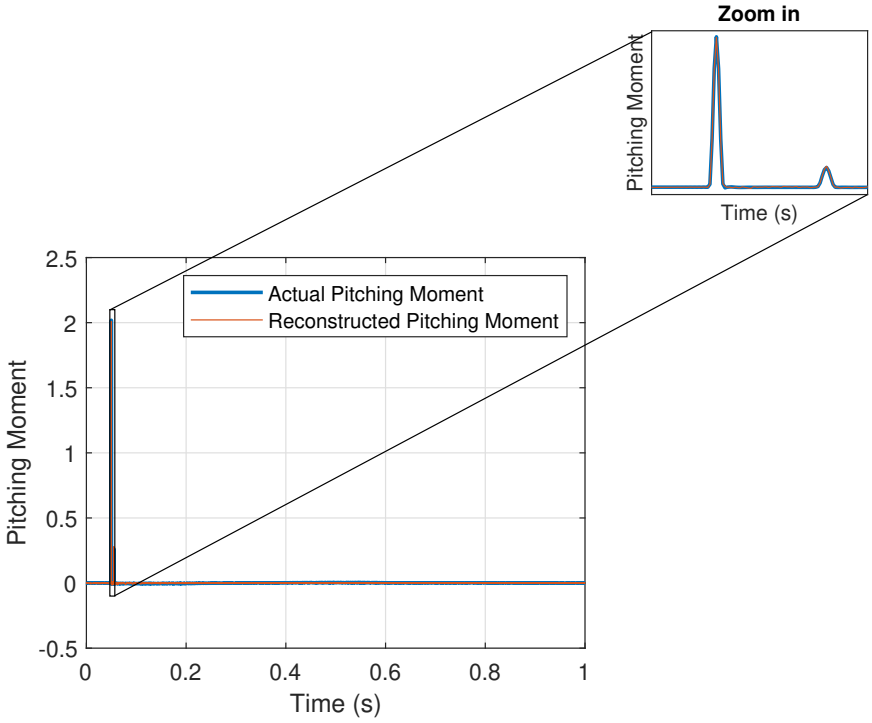

Figure 4.9: Experimental dynamic pitching moment reconstructed

The hammer's frequency domain reconstruction can be seen in Figure 4.10.

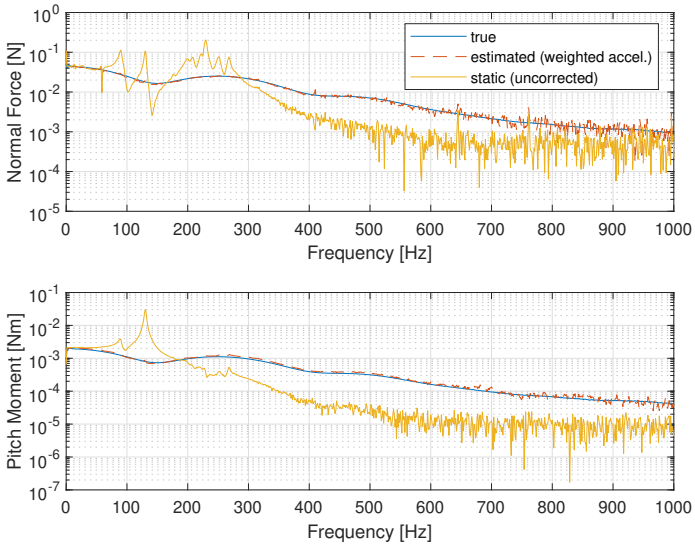

Figure 4.10: Frequency domain reconstruction

A comparison of the dynamic calibration conducted versus a static reconstruction, and an error

plot, can be seen in Figure 4.11.

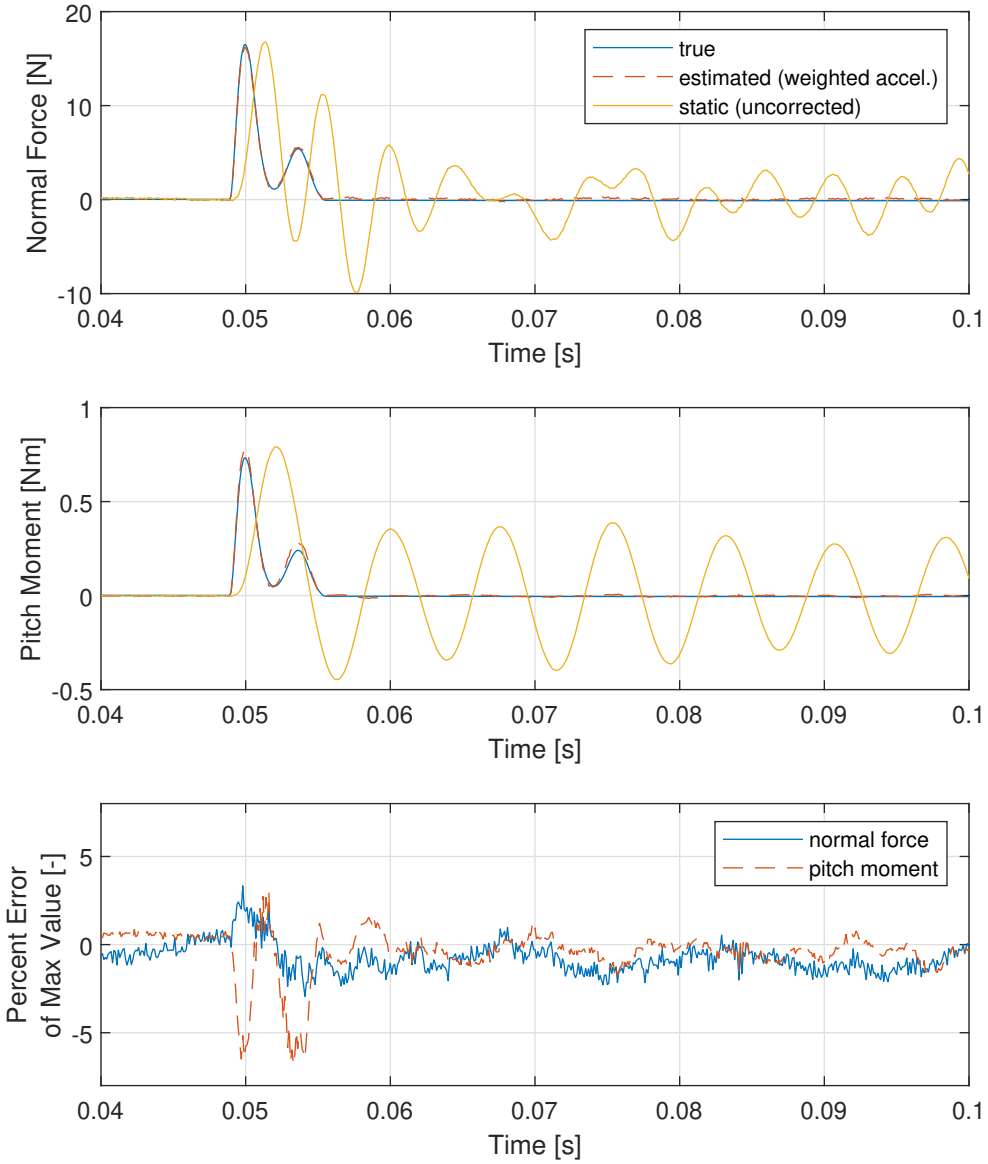

Figure 4.11: Comparison of experimental static calibration and experimental dynamic calibration

Using the generated dynamic calibration coefficient matrix, a new force can be reconstructed. 1000 grams of static weight were added one inch forward of the balance moment center. It was connected by a string. During the five seconds of test, the string was cut, which simulates a step force. The reconstruction of this experiment can be seen in Figure 4.12.

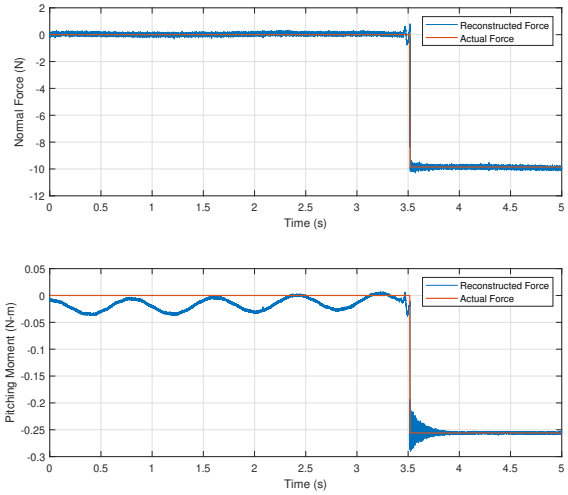

Figure 4.12: Reconstruction of a step force using the dynamic calibration coefficients

The step force reconstruction error can be seen in Figure 4.13.

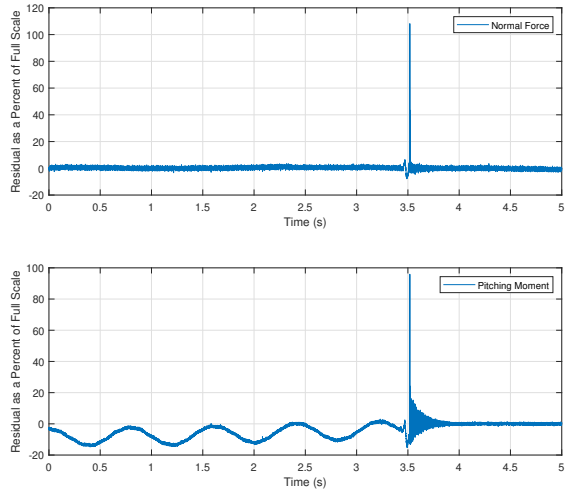

Figure 4.13: Reconstruction error of a step force using the dynamic calibration coefficients

The normal force maintains a small error until the string is cut, where it spikes and then goes back down. The pitching moment oscillates before the string is cut. This could be because of tolerances on the tombstone, or because of a swinging motion on the string. After it is cut there is a high error, and then it levels out.

Chapter 5

Conclusions

A two degree-of-freedom lift-pitch model was successfully able to mathematically and experimentally reconstruct dynamic forces. Theoretically, this can be applied on wind tunnels to vehicle models. However, a few complications arise. If the system is greater than two degrees-of-freedom, with the introduction of axial forces or yaw/roll moments, a greater complexity model will be needed. It should also be noted that the experiment was designed to prove the model. The tombstone was chosen so that the balance could fit inside with controlled forces applied. Fitting a balance and accelerometers may be difficult in a test setting, and conditions not covered by this experiment could add error to the method.

Some sources of error included the previously mentioned tolerances on the tombstone, and swaying of the string for the step force reconstruction. The numerical integration of the acceleration to yield velocity for the damping component is a source of error too. The damping component was small enough for this to not have a large effect, but a system with a significant damping component may need additional considerations. The simulation of a dynamic force by impact hammer created error because any tap that was offset would induce a twisting mode, or error induced by the hammer not hitting precisely where the tap point was recorded. The static calibration had error caused by the applied load. The balance was designed to reach optimal strain levels for a 45 N normal force or a 2.8 N-m pitching moment. Due to experimental constraints, the maximum normal force was 19.61 N lbs or 0.28 N-m. Since the maximum applied pitching moment was 10 percent of the design pitching moment, while the maximum normal force was 44 percent, the pitching moment static calibration showed a higher error.

Future work can be done on creating a six degree-of-freedom balance, creating a new model, and experimentally validating it. This work serves as a proof-of-concept that a higher order model could work. Future work could also include no-contact forces, particularly no contact dynamic forces. Electromagnetic induction could be used to accomplish this, which would simulate results more accurate to a wind tunnel. Distributed dynamic loads would be of interest to future research, because only point forces were applied while a distributed load would be more accurate to a wind tunnel simulation.

Bibliography

[1] A.R. Gorbushin., A.E. Kozik., and E.N. Anokhina. *Non-stationary load measurement using six-component strain-gauge balances*. ScienceDirect, Jan 2023.

[2] G. R. Duryea. *An Improved Piezoelectric Balance for Aerodynamic Force*. IEEE Transactions on Aerospace and Electronic Systems, Feb 2010.

[3] N. Sahoo. *Dynamic force balances for short-duration hypersonic testing facilities*. Experiments in Fluids, Mar 2005.

[4] Zachary T. Jones., Nicholas A. Vlajic., Peter A. Parker., and Devin E. Burns. *Reduced-Order Modeling and Parameter Identification of Wind Tunnel Measurement Systems*. AIAA, Jan 2023.

[5] Devin E. Burns., Nicholas Vlajic., Ako Chijioke., and Peter A. Parker. *Aerodynamic Metrologists Guide to Dynamic Force Measurement*. ARC, Oct 2022.

[6] Nicholas Vlajic. and Ako Chijioke. *An Introduction to Dynamic Force Metrology*. NCSLI, 2018.

[7] R. Buehrle. *SYSTEM DYNAMIC ANALYSIS OF A WIND TUNNEL MODEL WITH APPLICATIONS TO IMPROVE AERODYNAMIC DATA QUALITY*. NASA, Jan 2012.

[8] T G. Carne, R L. Mayes, and V I Bateman. *FORCE RECONSTRUCTION USING THE SUM OF WEIGHTED ACCELERATION TECHNIQUE MAX-FLAT PROCEDURE*. OSTI, Mar 1994.

[9] John Draper III., Sung Lee., and Eric C. Marineau. *Development and Implementation of a Hybrid Dynamic Force Measurement System at AEDC Tunnel 9*. AIAA, Jan 2017.

[10] John Draper III. *Dynamic Force Measurement in Hypersonic Wind Tunnels*. University of Maryland, May 2019.

[11] Devin E. Burns., Peter A. Parker., Benjamin D. Phillips., Thomas L. Webb III., and Drew Landman. *Wind Tunnel Balance Design: A NASA Langley Perspective*. NASA, Mar 2020.

Appendix

A Balance Design

A.1 Engineering Drawing of the Final Balance

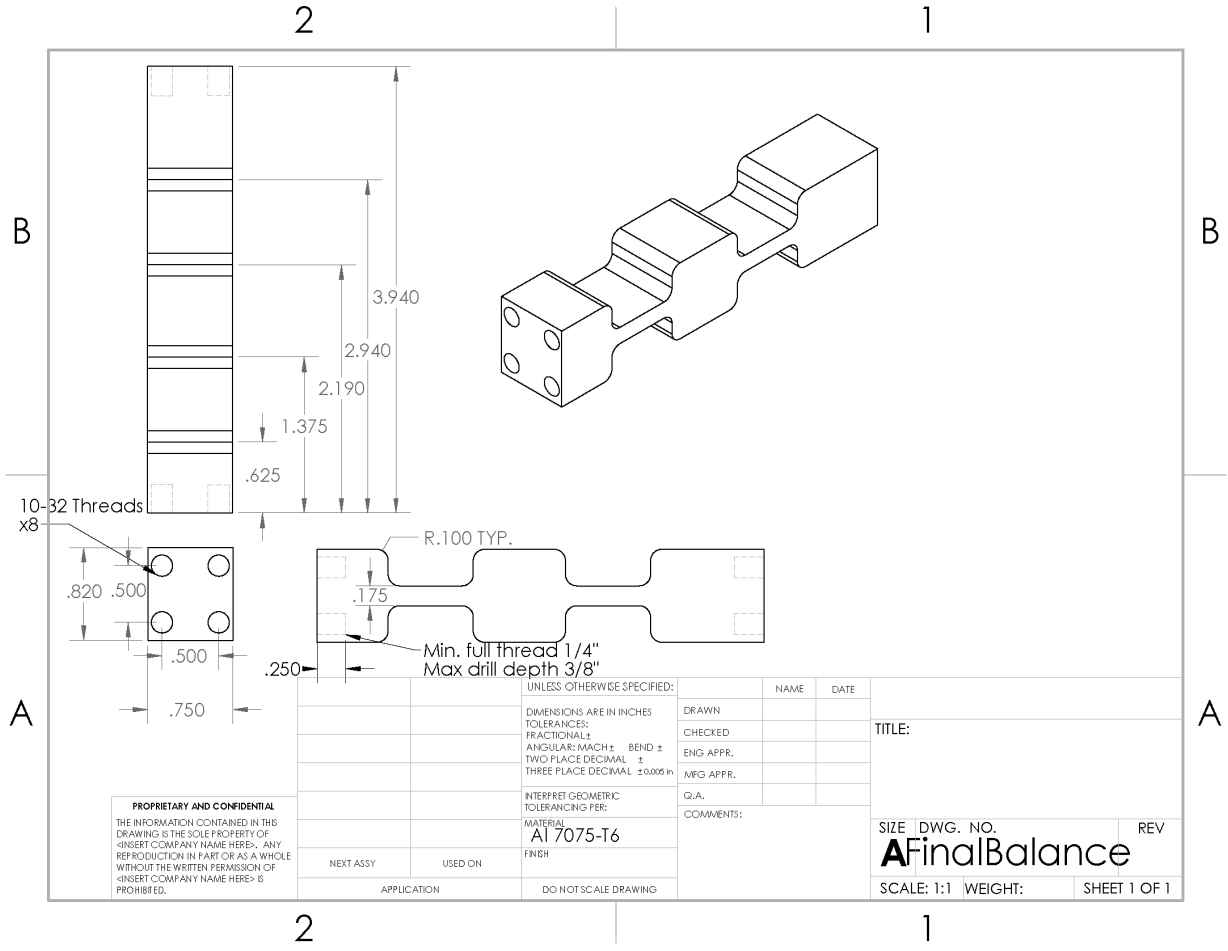

Figure 1: Engineering Drawing of the Final Balance

A.2 Wiring Diagram of the Strain Gauge Balance

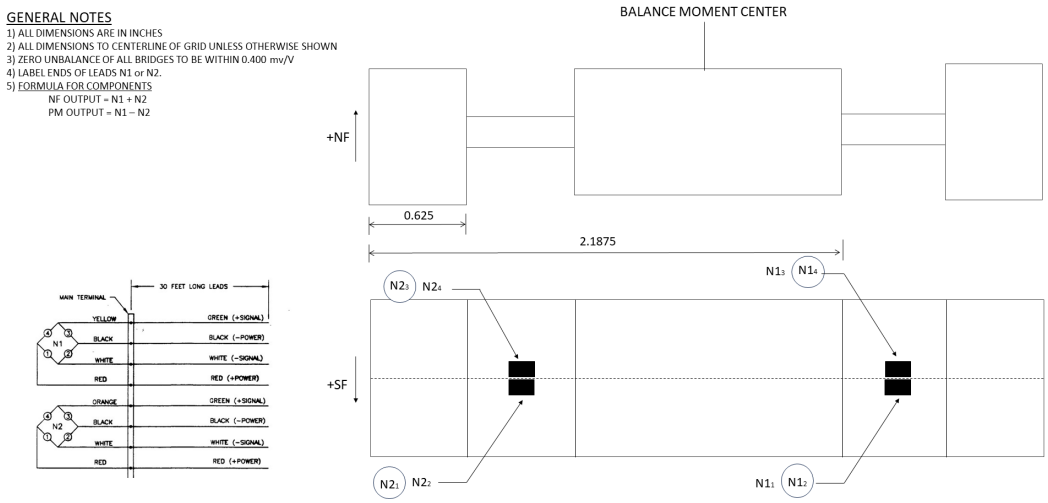

Figure 2: Wiring Diagram of the Strain Gauge Balance

B MATLAB Scripts

B.1 Simultaneously identify the static, velocity, and inertial weighting coefficient

```
    %% Simultaneously identify the static, velocity, and inertial weighting
clear
close all
clc

%% Script Setup

%Defined Variables for mass and moment of inertia
mm = 0.3313;
Iybc = 19.51;

%Applied distance to the moment center of two forces
dist1 = 4;
dist2 = 6;

%Moment distance
dcgbc = 5;

%Spring stiffnesses
```

```
kb = 1308;
kt = 277089;

%Damping
d1 = 0.2;
d2 = 1;

%ODE Setup
fs          = 1e3;
dt              = 1/fs;
tSpan       = 0:dt:30;
xo          = [0 0 0 0]'; %initial conditions (x0, v0)
K = [kb,0;0,kt];
M = [mm, -mm*dcgbc;-mm*dcgbc,Iybc];
D = [d1 0;
     0 d2];

% Rename the applied forces
options         = odeset('MaxStep',1e-3);
f_step          = @(t) double(t > 1); %Step Function
f_pulse         =  @(t) 1/1e-3*exp(-( 2 - t).^2/1e-4); %Gaussian Pulse

% Do the dynamic calibration with the same pulse force, but at two separate
% locations: dist1 and dist2. Also, changing the applied forces to be
% functions
Q1 = @(t)[-f_pulse(t);f_pulse(t).*dist1];
Q2 = @(t)[-f_pulse(t);f_pulse(t).*dist2];
%% Simultaneously identify the static, velocity, and inertial weighting coe

%Solving ODE for time and position of the same force applied at different
%spots
[time1,x1]    = ode45(@odeEqn2DOF,tSpan,xo,options,M,K,D,Q1);
[time2,x2]    = ode45(@odeEqn2DOF,tSpan,xo,options,M,K,D,Q2);

% Find accelerations for both scenarios by reinputting values into the ODE.
% This is to replace the data that would be found by accelerometers
acc1 = nan(length(time1),2); % 2 accelerations
for i1 = 1:length(time1)
    t   = time1(i1);
    xtemp   = x1(i1,:).';
    dx  = odeEqn2DOF(t,xtemp,M,K,D,Q1);

    acc1(i1,:)   = dx([3,4]);
end
```

```
acc2 = nan(length(time2),2); % 2 accelerations
for i1 = 1:length(time2)
    t    = time2(i1);
    xtemp    = x2(i1,:).';
    dx   = odeEqn2DOF(t,xtemp,M,K,D,Q2);

    acc2(i1,:)    = dx([3,4]);
end

% Numerically integrate acceleration to find velocity
vt = dt*cumsum(acc1);
vel1 = vt - mean(vt);

vt = dt*cumsum(acc1);
vel2 = vt - mean(vt);


%Relabel x position for convenience
pos1(:,1) = x1(:,1);
pos1(:,2) = x1(:,2);
pos2(:,1) = x2(:,1);
pos2(:,2) = x2(:,2);


% Assemble S matrix
S1 = [pos1(:,1),vel1(:,1),acc1(:,1),pos1(:,2),vel1(:,2),acc1(:,2)];
S2 = [pos2(:,1),vel2(:,1),acc2(:,1),pos2(:,2),vel2(:,2),acc2(:,2)];
S = [S1;S2];

%Define Q matrix
Q = [Q1(tSpan)';Q2(tSpan)'];

%Solve for inertial and damping coefficients
C = ((S'*S)\S')*Q;

%Generate graph comparing the reconstructed force with the actual force
figure()
hold on
subplot(3,3,[4 5 7 8])
plot(tSpan,Q1(tSpan),":','LineWidth',1.5)
hold off

hold on
subplot(3,3,[4 5 7 8])
plot(tSpan,S1*C,"--','LineWidth',1.5)
```

```
legend('Actual z Force','Actual \alpha Force','Reconstructed z Force','Reco
xlabel('Time (s)')
ylabel('Force (lbs)')
grid on
%title('Reconstructed Force vs Actual Force')
hold off
xlim([0,5])

ax = gca;
area = [1.95 -1500 2.05 4500];
inlarge = subplot(3,3,3);
panpos = inlarge.Position;
delete(inlarge);
inlarge = zoomin(ax,area,panpos);
title(inlarge,'Zoom in')
hold off

%% Using the Generated Calibration Constants to Reconstruct A Different Forc
%Redefining force and distance for a step force
Q1 = @(t)[-f_step(t);f_step(t).*dist1];

[time1,x1]     = ode45(@odeEqn2DOF,tSpan,xo,options,M,K,D,Q1);

% Finding accelerations for the new force
acc1 = nan(length(time1),2); % 2 accelerations
for i1 = 1:length(time1)
    t    = time1(i1);
    xtemp    = x1(i1,:).';
    dx   = odeEqn2DOF(t,xtemp,M,K,D,Q1);

    acc1(i1,:)    = dx([3,4]);
end

% Numerically integrate to find velocity
vt = dt*cumsum(acc1);
vel1 = vt - mean(vt);

%Redefine x to pos
pos1(:,1) = x1(:,1);
pos1(:,2) = x1(:,2);

% Assemble S matrix
S1 = [pos1(:,1),vel1(:,1),acc1(:,1),pos1(:,2),vel1(:,2),acc1(:,2)];

%Plot force reconstruction with previously constructed coefficients
```

```
figure()
hold on
plot(tSpan,Q1(tSpan),":",'LineWidth',1.5)
plot(tSpan,S1*C,"--",'LineWidth',1.5)
legend('Actual z Force','Actual \alpha Force','Reconstructed z Force','Reco
xlabel('Time (s)')
ylabel('Force (lbs)')
grid on
title('Reconstructed Force vs Actual Force with Previously Generated Coeffi
hold off

%Generate error
err = abs((S1*C)'-Q1(tSpan))./abs(Q1(tSpan)).*100;
figure()
plot(tSpan,err)
ylabel('Error (%)')
xlabel('Time (s)')
grid on
legend('z','\alpha')

function pan = zoomin(ax,areaToMagnify,panPosition)
% AX is a handle to the axes to magnify
% AREATOMAGNIFY is the area to magnify, given by a 4-element vector that de
%      lower-left and upper-right corners of a rectangle [x1 y1 x2 y2]
% PANPOSTION is the position of the magnifying pan in the figure, defined by
%        the normalized units of the figure [x y w h]
%

fig = ax.Parent;
pan = copyobj(ax,fig);
pan.Position = panPosition;
pan.XLim = areaToMagnify([1 3]);
pan.YLim = areaToMagnify([2 4]);
pan.XTick = [];
pan.YTick = [];
rectangle(ax,'Position',...
    [areaToMagnify(1:2) areaToMagnify(3:4)-areaToMagnify(1:2)])
xy = ax2annot(ax,areaToMagnify([1 4;3 2]));
annotation(fig,'line',[xy(1,1) panPosition(1)],...
    [xy(1,2) panPosition(2)+panPosition(4)],'Color','k')
annotation(fig,'line',[xy(2,1) panPosition(1)+panPosition(3)],...
    [xy(2,2) panPosition(2)],'Color','k')
end

function anxy = ax2annot(ax,xy)
```

```
% This function converts the axis unites to the figure normalized unites
% AX is a handle to the figure
% XY is a n-by-2 matrix, where the first column is the x values and the
% second is the y values
% ANXY is a matrix in the same size of XY, but with all the values
% converted to normalized units

pos = ax.Position;
%   white area * ((value - axis min) / axis length)   + gray area
normx = pos(3)*((xy(:,1)-ax.XLim(1))./range(ax.XLim))+ pos(1);
normy = pos(4)*((xy(:,2)-ax.YLim(1))./range(ax.YLim))+ pos(2);
anxy = [normx normy];
end

function dx = odeEqn2DOF(t,x,M,K,D,Q)

% SDOF
dx = [0 0 0 0]';

% state space representation x(1) = position, x(2) = velocity
% dx(1) = x(2);
% dx(2) = 1/m*(-d*x(2) - k*x(1) + f_fn(t));

% 2 DOF
F = Q(t);
A = [zeros(size(K)) eye(size(K));
     -M\K               -M\D];
B  =[zeros(length(M),1); M\F];

dx = A*x + B;
end
```

B.2 MATLAB Script: Identify coefficients for lift-pitch in a two-step process

```
%% Identify coefficients for lift-pitch in a two-step process
clear
close all
clc

%% Script Setup

%Defined Variables for mass and moment of inertia
mm = 0.3313;
Iybc = 19.51;
```

```
%Applied distance to the moment center of two forces
dist1 = 4;
dist2 = 6;

%Moment distance
dcgbc = 5;

%Spring stiffnesses
kb = 1308;
kt = 277089;

%Damping
d1 = 0.2;
d2 = 1;

%ODE Setup
fs          = 1e3;
dt              = 1/fs;
tSpan       = 0:dt:30;
xo          = [0 0 0 0]'; %initial conditions (x0, v0)
K = [kb,0;0,kt];
M = [mm, -mm*dcgbc;-mm*dcgbc,Iybc];
D = [d1 0;
     0 d2];

% Rename the applied forces
options         = odeset('MaxStep',1e-3);
f_step          = @(t) double(t > 1); %Step Function
f_pulse         =  @(t) 1/1e-3*exp(-( 2 - t).^2/1e-4); %Gaussian Pulse

% Do the dynamic calibration with the same pulse force, but at two separate
% locations: dist1 and dist2. Also, changing the applied forces to be
% functions
Q1 = @(t)[-f_pulse(t);f_pulse(t).*dist1];
Q2 = @(t)[-f_pulse(t);f_pulse(t).*dist2];

%% Static Calibration

%Apply known static force
F1 = [1;1];

%Generate displacement data that using an unknown k value. This is to
%replace data that would have come from force transducers.
x1 = K\F1;
```

```
%Apply known second force and repeat
F2 = [2;3];
x2 = K\F2;

%Calculate coefficient
SF = [x1,x2]';
F = [F1,F2]';
CF      = ((SF'*SF)\SF')*F;

%% Identify coefficients for lift-pitch in a two-step process:

%Solving ODE for time and position of the same force applied at different
%spots
[time1,x1]     = ode45(@odeEqn2DOF,tSpan,xo,options,M,K,D,Q1);
[time2,x2]     = ode45(@odeEqn2DOF,tSpan,xo,options,M,K,D,Q2);

% Find accelerations for both scenarios by reinputting values into the ODE.
% This is to replace the data that would be found by accelerometers
acc1 = nan(length(time1),2); % 2 accelerations
for i1 = 1:length(time1)
    t    = time1(i1);
    xtemp    = x1(i1,:).';
    dx   = odeEqn2DOF(t,xtemp,M,K,D,Q1);

    acc1(i1,:)    = dx([3,4]);
end

acc2 = nan(length(time2),2); % 2 accelerations
for i1 = 1:length(time2)
    t    = time2(i1);
    xtemp    = x2(i1,:).';
    dx   = odeEqn2DOF(t,xtemp,M,K,D,Q2);

    acc2(i1,:)    = dx([3,4]);
end

% Numerically integrate acceleration to find velocity
vt = dt*cumsum(acc1);
vel1 = vt - mean(vt);

vt = dt*cumsum(acc1);
vel2 = vt - mean(vt);
```

```
%Relabel x position for convenience
pos1(:,1) = x1(:,1);
pos1(:,2) = x1(:,2);
pos2(:,1) = x2(:,1);
pos2(:,2) = x2(:,2);


% Assemble S matrix
S1 = [vel1(:,1),acc1(:,1),vel1(:,2),acc1(:,2)];
S2 = [vel2(:,1),acc2(:,1),vel2(:,2),acc2(:,2)];
S = [S1;S2];

%Subtract static component from inertial and damping forces
Q1temp = Q1(tSpan)'-pos1*CF;
Q2temp = Q2(tSpan)'-pos2*CF;
Q = [Q1temp;Q2temp];

%Solve for inertial and damping coefficients
C = ((S'*S)\S')*Q;

%Generate graph comparing the reconstructed force with the actual force
figure()
hold on
subplot(3,3,[4 5 7 8])
plot(tSpan,Q1(tSpan),":','LineWidth',1.5)
hold off

hold on
subplot(3,3,[4 5 7 8])
plot(tSpan,S1*C+pos1*CF,"--','LineWidth',1.5)
legend('Actual z Force','Actual \alpha Force','Reconstructed z Force','Reco
xlabel('Time (s)')
ylabel('Force (lbs)')
grid on
%title('Reconstructed Force vs Actual Force')
hold off
xlim([0,5])

ax = gca;
area = [1.95 -1500 2.05 4500];
inlarge = subplot(3,3,3);
panpos = inlarge.Position;
delete(inlarge);
inlarge = zoomin(ax,area,panpos);
title(inlarge,'Zoom in')
```

```
hold off

%% Using the Generated Calibration Constants to Reconstruct A Different Fore
%Redefining force and distance for a step force
Q1 = @(t)[-f_step(t);f_step(t).*dist1];

[time1,x1]     = ode45(@odeEqn2DOF,tSpan,xo,options,M,K,D,Q1);

% Finding accelerations for the new force
acc1 = nan(length(time1),2); % 2 accelerations
for i1 = 1:length(time1)
    t    = time1(i1);
    xtemp    = x1(i1,:).';
    dx   = odeEqn2DOF(t,xtemp,M,K,D,Q1);

    acc1(i1,:)    = dx([3,4]);
end

% Numerically integrate to find velocity
vt = dt*cumsum(acc1);
vel1 = vt - mean(vt);


%Redefine x to pos
pos1(:,1) = x1(:,1);
pos1(:,2) = x1(:,2);


% Assemble S matrix
S1 = [vel1(:,1),acc1(:,1),vel1(:,2),acc1(:,2)];

%Plot force reconstruction with previously constructed coefficients
figure()
hold on
plot(tSpan,Q1(tSpan),":",'LineWidth',1.5)
plot(tSpan,S1*C+pos1*CF,"--",'LineWidth',1.5)

legend('Actual z Force','Actual \alpha Force','Reconstructed z Force','Reco
xlabel('Time (s)')
ylabel('Force (lbs)')
grid on
%title('Reconstructed Force vs Actual Force with Previously Generated Coeff
hold off

%Generate graph showing the statically reconstructed force with the actual
```

```
%force
stat = pos1*CF;
forc = Q1(tSpan);
figure()
hold on
subplot(3,3,[4 5 7 8])
plot(tSpan,forc(2,:),'LineWidth',1.5)
hold off

hold on
subplot(3,3,[4 5 7 8])
plot(tSpan,stat(:,2))
plot(tSpan, forc(2, :), 'b', 'LineWidth', 1.5);
%legend('Actual z Force','Actual \alpha Force','Reconstructed z Force','Rec
legend('Actual Force','Static Reconstructed Force')
xlabel('Time (s)')
ylabel('Force (lbs)')
grid on
%title('Reconstructed Force with Static Calibration vs Actual Force')

ax = gca;
area = [0 -3 7 11];
inlarge = subplot(3,3,3);
panpos = inlarge.Position;
delete(inlarge);
inlarge = zoomin(ax,area,panpos);
title(inlarge,'Zoom in')
hold off


%Generate error
err = abs((S1*C+pos1*CF)'-Q1(tSpan))./abs(Q1(tSpan)).*100;
figure()
plot(tSpan,err)
ylabel('Error (%)')
xlabel('Time (s)')
grid on
legend('z','\alpha')

function pan = zoomin(ax,areaToMagnify,panPosition)
% AX is a handle to the axes to magnify
% AREATOMAGNIFY is the area to magnify, given by a 4-element vector that de
%      lower-left and upper-right corners of a rectangle [x1 y1 x2 y2]
% PANPOSTION is the position of the magnifying pan in the figure, defined by
%        the normalized units of the figure [x y w h]
```

```
%

fig = ax.Parent;
pan = copyobj(ax,fig);
pan.Position = panPosition;
pan.XLim = areaToMagnify([1 3]);
pan.YLim = areaToMagnify([2 4]);
pan.XTick = [];
pan.YTick = [];
rectangle(ax,'Position',...
    [areaToMagnify(1:2) areaToMagnify(3:4)-areaToMagnify(1:2)])
xy = ax2annot(ax,areaToMagnify([1 4;3 2]));
annotation(fig,'line',[xy(1,1) panPosition(1)],...
    [xy(1,2) panPosition(2)+panPosition(4)],'Color','k')
annotation(fig,'line',[xy(2,1) panPosition(1)+panPosition(3)],...
    [xy(2,2) panPosition(2)],'Color','k')
end

function anxy = ax2annot(ax,xy)
% This function converts the axis unites to the figure normalized unites
% AX is a handle to the figure
% XY is a n-by-2 matrix, where the first column is the x values and the
% second is the y values
% ANXY is a matrix in the same size of XY, but with all the values
% converted to normalized units

pos = ax.Position;
%   white area * ((value - axis min) / axis length)   + gray area
normx = pos(3)*((xy(:,1)-ax.XLim(1))./range(ax.XLim))+ pos(1);
normy = pos(4)*((xy(:,2)-ax.YLim(1))./range(ax.YLim))+ pos(2);
anxy = [normx normy];
end

function dx = odeEqn2DOF(t,x,M,K,D,Q)

% SDOF
dx = [0 0 0 0]';

% state space representation x(1) = position, x(2) = velocity
% dx(1) = x(2);
% dx(2) = 1/m*(-d*x(2) - k*x(1) + f_fn(t));

% 2 DOF
F = Q(t);
A = [zeros(size(K)) eye(size(K));
```

```
        -M\K                    -M\D];
B    =[zeros(length(M),1); M\F];

dx = A*x + B;
end
```