# CALIBRATION AND UNCERTAINTY ANALYSIS METHOD FOR A PYRAMIDAL EXTERNAL BALANCE

A Thesis

by

JOHN WILLIAM STANFORD III

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

| | |
|---|---|
| Chair of Committee, | Edward White |
| Co-Chair of Committee, | Othon K. Rediniotis |
| Committee Member, | Sungyon Lee |
| Head of Department, | Rodney Bowersox |

August 2016

Major Subject: Aerospace Engineering

# ABSTRACT

The Oran W. Nicks Low Speed Wind Tunnel (LSWT) uses a large-scale pyramidal-type balance to measure aerodynamic loads. The balance was built before computers were widely available and, consequently, it was designed to have a fixed, linear calibration without interactions between components. The physical system would be periodically adjusted to match the mathematical model. However, modern data acquisition and data processing make it much easier to leave the physical system fixed and modify the calibration to match the physical system. This thesis details a new calibration and uncertainty analysis method based on this principle. The method of calibration involves using pneumatic cylinders to apply a load to the external balance and using a strain gauge to measure the applied load, then comparing this measurement to the raw output from the balance in the form of rotary encoder counts. The relationship between these encoder counts and the forces and moments constitutes the calibration. Results show that the calibration coefficients have changed less than 1% from their original nominal values. The relationship between the forces and encoder counts has also remained very linear with very little evidence of interactions between components. It is recommended that the LSWT retain the linear model with independent components, but allow the coefficients to vary independently of each other. This 6-degree of freedom calibration allows a reduction in uncertainty from the current 1-degree of freedom calibration, but remains simpler than the more general 36-degree of freedom calibration.

ACKNOWLEDGEMENTS

I would like to first thank my advisor, Dr. Ed White. His guidance and reassurance during this long process have made this thesis possible. I would also like to thanks the entire wind tunnel staff. John Kochan and Ric Warren were especially helpful in the early stages when I was learning to basics of how the balance works. Zahir Udovicic was extremely helpful during the fabrication phase, mentoring me on machine shop skills. I would also like to thanks my family, especially my wife, Michelle, for her support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

# 1. INTRODUCTION

## 1.1 Background and Objective

Wind tunnels of various designs have been the primary tool of experimental aerodynamics for more than 100 years. Wind tunnels are important because even the simplest fluid dynamics problems usually require significant simplifying assumptions for analytic solutions. The solutions resulting from these assumptions are often good enough to give the aerodynamicist a general idea of how a certain geometry should produce aerodynamic forces, but not good enough to be used for real engineering problems. For complicated shapes, analytic solutions are normally not possible, even with simplifying assumptions. Computational fluid dynamics (CFD) is a field that takes advantage of modern computing power to produce solutions closer to reality than analytic solutions. CFD can analyze complex shapes with fewer simplifying assumptions than analytic solutions. However, CFD is still significantly limited by the very small length and time scales involved in many aerodynamic problems.

Even after more than a century of development of the field, the best way to determine how an object will generate aerodynamic forces is still to simply apply wind to a real object and measure the forces. Wind tunnels are the primary tool used for this purpose. There are two basic types of wind tunnels: open circuit and closed return.[1] The low speed wind tunnel at Texas A&M University began as an open circuit design when it was originally built in the 1940s and was upgraded to a closed return design in 1958.[2] Open-circuit tunnels intake air from the surroundings through a contraction cone, then return the air back to the surroundings after the test section. With a closed-return design, air leaves the test section then follows a continuous loop back around to the test section. Each design has advantages and

disadvantages. The primary advantage of the open circuit design is cost. However, it requires more energy to reach the same dynamic pressure in the test section because the air must be continually accelerated from rest to the desired velocity. With a closed return design, the air retains some of its momentum as it goes around the loop and a portion of the momentum is "recycled." This makes it possible to achieve higher dynamic pressures in the test section with the same energy input.

Any model in the test section of a wind tunnel is subjected to three forces and three moments: drag, lift, and side forces and roll, yaw, and pitch moments. The objective of most wind tunnel tests is to determine these six components resulting from a variety of configurations and conditions. External balances of various designs are one of the most common and time-tested ways to measure these components.

Rae and Pope [1] detail the principles of operations as well as the advantages and disadvantages of three of the most common designs: platform, yoke, and pyramidal balances. The platform and pyramidal balances have the advantage of producing small deflections compared to the yoke balance when measuring forces and moments. Compared to the pyramidal balance, the yoke and platform designs have the disadvantage that moments are read as small differences in large forces. The yoke and pyramidal balance designs have the advantage that the resolving center is at the model, which is not the case with the platform design. Apart from listing the general advantages and disadvantages for comparison to the pyramidal balance, further details on the principles of operation of platform and yoke balances are outside the scope of this work.

To summarize the characteristics of the pyramidal design, it has the advantage of measuring forces and moments about the model, doing so with small deflections, and measuring each component independently. However, this comes at the cost of increased complexity and the requirement to ensure that the supporting struts

are aligned properly. Figure 1.1 illustrates the basic principle of operation of this type of balance.[1] Although an actual pyramidal balance is more complicated, this diagram illustrates the importance of having the supporting struts aligned properly. The intersection of the strut axes defines the resolving center at which an applied moment will create no reaction force. Any misalignment will create interactions among the measured forces at the nominal resolving center, which must be detected during calibration and either corrected through mechanical adjustment or accounted for in the mathematical model of the balance.

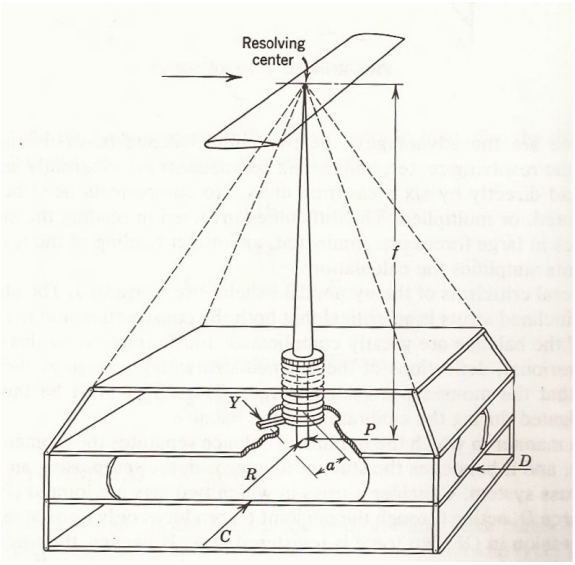

Figure 1.1: Diagram of a Pyramidal-Type External Balance

## 1.2 Usage Case Study

In April 2012 the Sierra Nevada Corporation issued a press release regarding a test they conducted on a model of their Dream Chaser Orbital Crew Vehicle in the Oran W. Nicks Low Speed Wind Tunnel[3]. This case study provides a useful illustration of how the external balance is typically used for aerodynamic testing. Figure 1.2 shows the scale model mounted on a single strut, which connects to the external balance.[3]

Figure 1.2: Scale Model Mounted on a Single Strut

In this type of configuration, the strut remains stationary and, if the model is required to pitch, an actuator is installed between the strut and the rest of the model. The figure shows a fairing below the model, which minimizes the aerodynamic effect of the wind on the strut.

The press release helps illustrate how wind tunnel testing fits into the typical aircraft design lifecycle. It quotes Mark Sirangelo, Corporate Vice President of SNC's

Space Systems, as saying "we are thankful for the opportunity to verify our computational data in such an advanced facility." This illustrates one of the most common objectives of these types of wind tunnel tests, which is to verify CFD calculations. In the press release Mark Sirangelo also says that this test "is an important step in preparing for the vehicle's first free flight." In the typical design lifecycle, wind tunnel testing comes after CFD simulation and before actual flight testing.

The press release doesn't include more details on the specific technical objectives of the test. A typical vehicle involves gathering data on the six forces and moments from the external balance resulting from a range of client-specified configurations. The configurations could include specified values of pitch angle, yaw angle, dynamic pressure, or variation in any degrees of freedom that the model might have. For example, at a given dynamic pressure, a client may have predicted a certain stall angle of attack using CFD. The wind tunnel test could verify this value by pitching up through that angle and looking for a significant increase in drag and decrease in lift read by the external balance.

To give a rough order of magnitude, the model could weigh somewhere between 100 and 200 lbf. Depending of dynamic pressure and angle of attack, the lift would be somewhere in the -300 to +300 lbf range and drag would be somewhere in the 100 to 200 lbf range. The external balance is a versatile tool used for a wide variety of wind tunnel tests. This case study shows just one of its many uses.

## 1.3 External Balance Design

As explained in the previous section, the LSWT uses a pyramidal external balance, which separates each force and moment into an independent measurement. The balance reads moments about the resolving center at the geometric center of the test section, 42 inches above the floor.

The best source of information for the details of the external balance as it was originally installed is the instruction manual [4], dated January 6th, 1958. The manual contains information on basic operation and maintenance as well as drawings showing how the balance resolves individual forces and moments. The drawings for the lift and pitch components are shown in Figure 1.3 and Figure 1.4, respectively.[4]

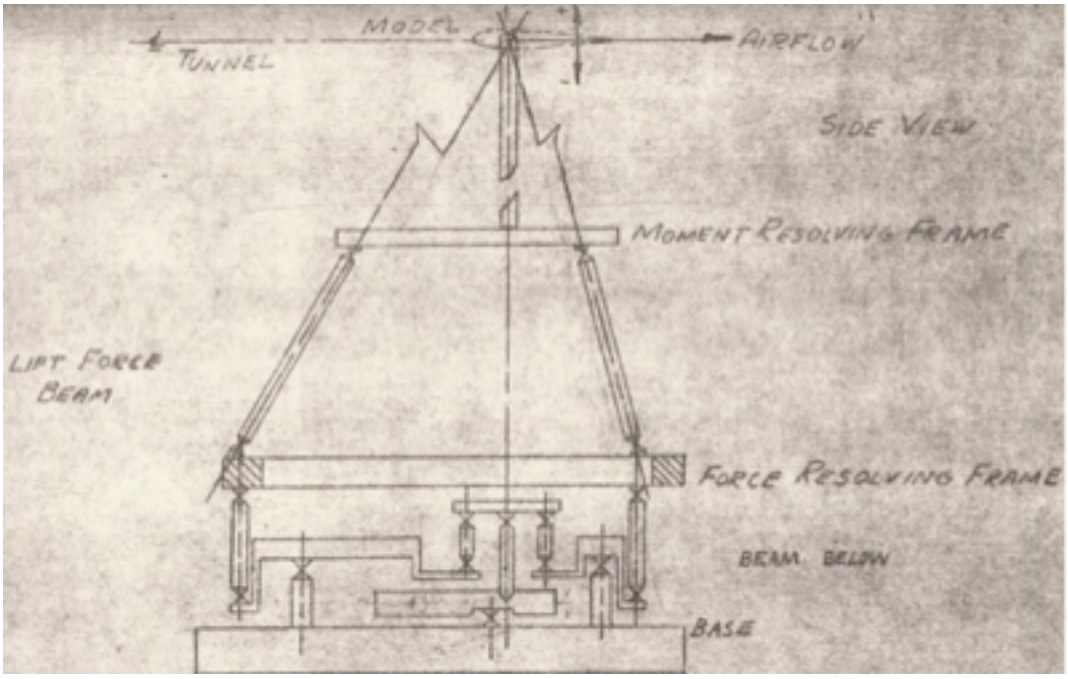

Figure 1.3: Lift Component of the External Balance

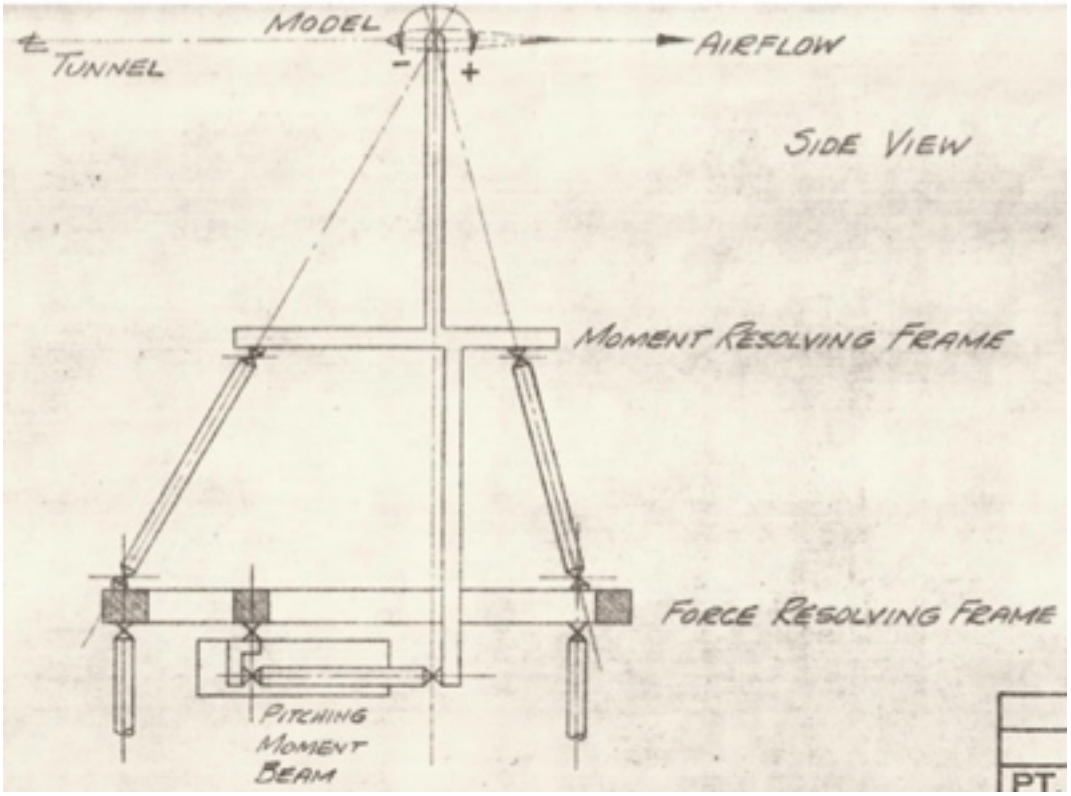

Figure 1.4: Pitch Component of the External Balance

From a conceptual standpoint, these drawings show that the balance has an upper moment resolving frame and a lower force resolving frame. These two are connected by the supporting struts. Since the supporting struts apply a force in line with the resolving center, they should transmit no moments to the force resolving frame when the balance is aligned properly. These two frames and the connection between them are shown in Figure 1.5. The connection between the two frames is at the far right of the photograph. The diagonal members going down toward the center of the balance do not transmit forces or moments to the force resolving frame. They simply transmit the moments to measuring beams.

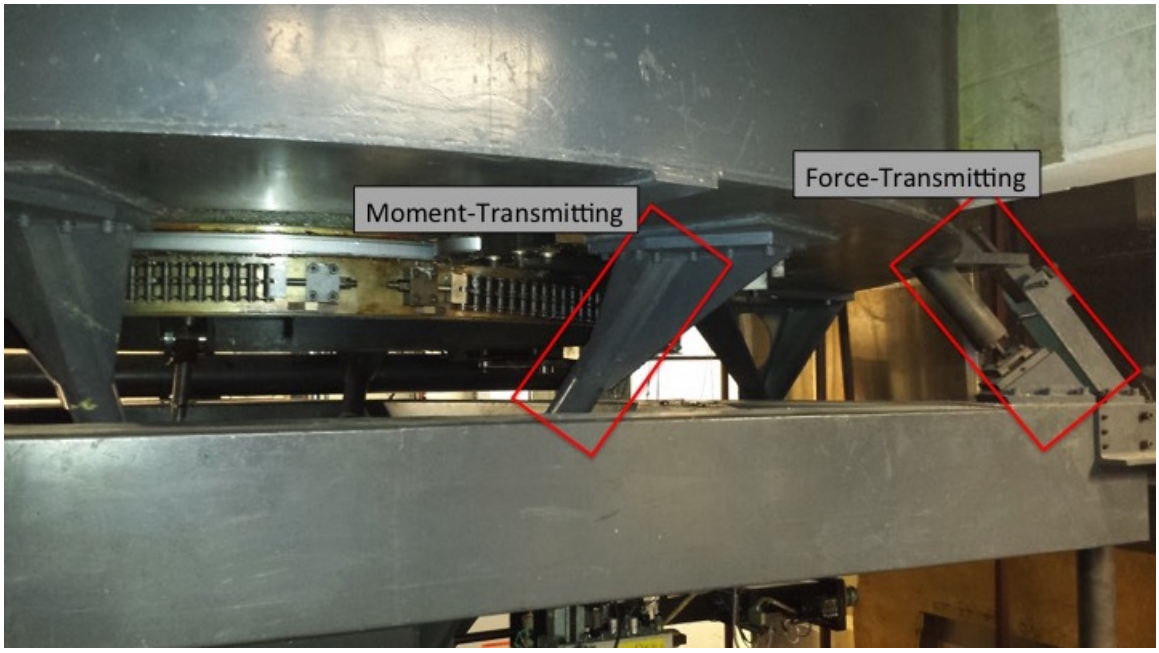

Figure 1.5: Upper Moment Resolving Frame and Lower Force Resolving Frame

One important part of the external balance not shown in the conceptual drawing is the lower turntable. This is a turntable on the moment resolving frame that can rotate independently of the rest of the balance, but still transmits its forces to the moment resolving frame. It is visible in Figure **??** as the part in the center of the moment resolving frame with the chain wrapped around it. Through this chain, the lower turntable is driven by an AC servomotor, allowing a model attached to the external balance to yaw. The servomotor assembly is shown in Figure 1.6.

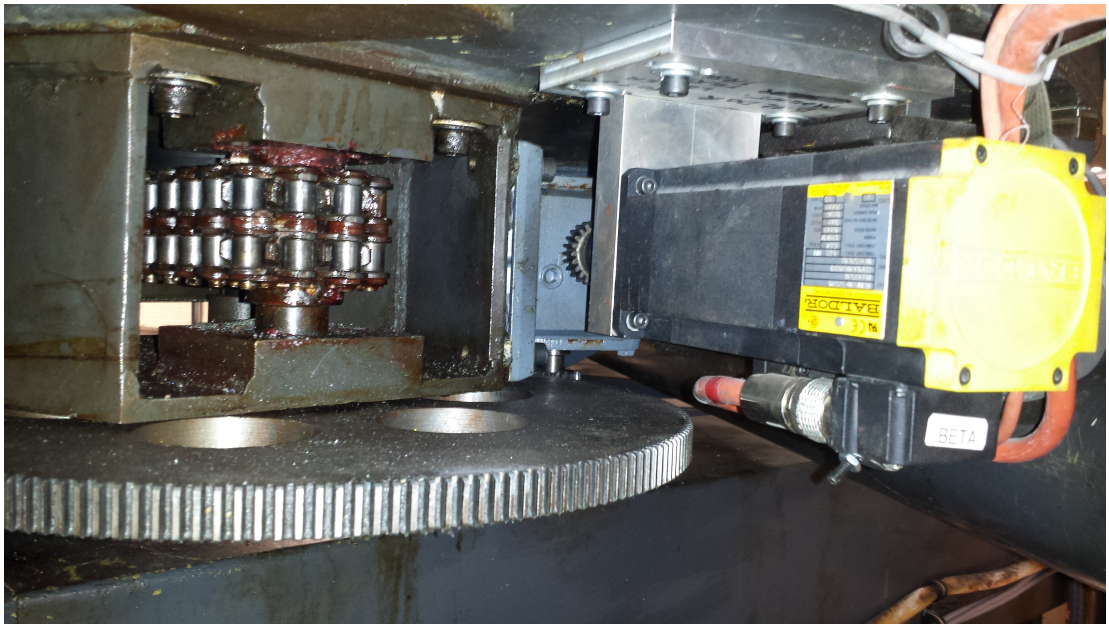

Figure 1.6: Servomotor Assembly that Drives the Lower Turntable

This becomes important when applying a large yawing moment to the balance because there is a small amount of slack in the chain that must be taken out by the servomotor every time the turntable changes direction. The chain slack direction can be set in the motion control program and by driving the turntable past the desired point, then coming back. It's important for the yawing moment to apply tension to the side of the chain that's already under tension. Otherwise the turntable will rotate in the direction of the yawing moment as the slack comes out.

For each of the six components, a force proportional to the force or moment on the model is applied to a measuring beam. A change in forces or moments on the model tips one or more of the beams. Each beam has a poise weight that moves along a leadscrew controlled by a servo motor that rebalances the beam. The servo motor is controlled by a signal from a linear variable differential transformer (LVDT), which measures linear displacement at the end of the beam. This signal travels through

an amplifier to the servo motor, which moves the poise weight in the appropriate direction to restore balance. Figure 1.7 shows all six measuring beams.

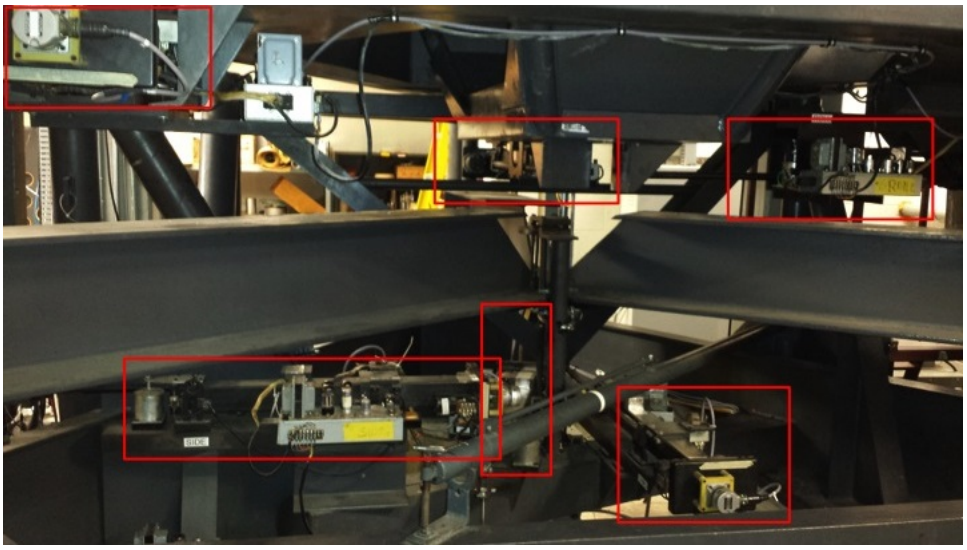

Figure 1.7: All Six Measuring Beams on the External Balance

The leadscrew controlled by the servo motor is also attached to a rotary encoder, which measures angular position of the leadscrew. This angular position corresponds to the linear position of the poise weight, which corresponds to the individual force or moment produced by the model and transmitter to the beam. Through this arrangement, the encoder counts from the measuring beams are related to the forces and moments on the model by a calibration model.

This calibration model exists in the data acquisition software. The entire process of measuring a force or moment on a model transitions from hardware to software as the information flows through the rotary encoder. The rotary encoder converts the angular position of the leadscrew to a digital signal, which is read by a dual encoder to USB converter module. Each module reads the digital output from two encoders,

meaning that there are three modules for the balance. Each one converts the digital encoder signals to the serial port communication protocol through the USB port.

The data acquisition program that interfaces with the encoder converters through the serial port is programmed in Python. It continually collects encoder counts and converts them to forces and moments using a nominal calibration coefficient of 200 counts per lbf or ft-lbf. The same coefficient is used for all six components. It plots these components over time and runs a TCP/IP server that allows other computers on the testing network to connect and request data. The data is sent to another computer to be recorded. The program also shows the current sampling rate for the external balance, which is typically in the 70-90 Hz range. A screenshot of the external balance server's graphical user interface in shown in the Figure 1.8. For a typical test, one centralized computer will collect data and direct all the motion control.

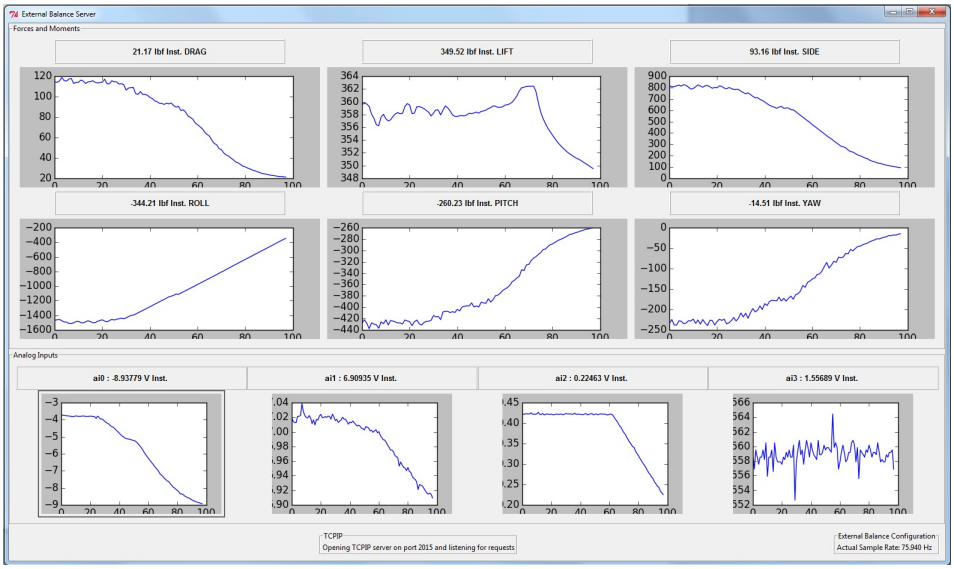

Figure 1.8: Graphical User Interface of the External Balance Server

The instruction manual for the balance indicates that anywhere from two to six months are required for calibration [4]. However, this estimate was made before the era of modern digital electronics. Many things that are now done digitally were done manually when the balance was built. For example, the instruction manual describes the procedure for manually zeroing the balance. The original balance had a console with analog indicators for the forces and moments. Once the model was installed in the test section, the test engineer would zero the balance by allowing the beams to reach steady state, turning off the power, and manually turning the indicators to zero. Today the balance is zeroed by sending a simple digital command to the rotary encoders.

The same general approach can be used to apply modern technology to calibrating the balance. In the 1950s, it was easier to change the physical model to match the mathematical model than vice versa. Today the reverse is true. Just as it is possible to zero the balance without mechanically changing anything, it is possible to recalibrate the balance without mechanically modifying it. This is extremely important because shutting down the wind tunnel for two to six months to do the calibration the old way is cost-prohibitive. The method developed in this project yields a new calibration in a matter of days. It does so by establishing a calibration corresponding to the as-found physical system rather than adjusting the physical system to match a nominal calibration.

# 2. CALIBRATION CONCEPT

Chapter 1 introduced the motivation to develop a new calibration method for the external balance and gave a basic description of its principle of operation. This chapter gives an overview of the basic research issues to be addressed by this project and the requirements that the calibration system must meet. It also covers the broad methodology used to approach the problem.

## 2.1 Research Issues

With the basic background information on the external balance and how it's typically used in mind, the next step is to formulate broad research issues to be investigated by this project.

The existing external balance calibration has just one degree of freedom: encoder counts per load on all channels. The first major research objective is to investigate the need for and uncertainty associated with using a 1-, 6-, or 36-degree-of-freedom calibration. In the current system, all six components use a coefficient of 200 encoder counts per lbf or ft-lbf. In the 1-degree-of-freedom case, this project will quantify the uncertainty associated with keeping a constant coefficient, but allowing it to deviate from 200 counts per lbf. In the 6-degree-of-freedom case, the coefficients will be allowed to vary independently of each other, but the system won't allow for interactions between components.

The 36-degree of freedom case will allow for linear interactions between components and quantify the significance of these interations. In theory, a pyramidal external balance separates each component into an independent measurement. However, there will be some nonzero interaction between the balance components due to manufacturing and strut alignment imperfections. This project will investigate

whether these interactions are significant. If they are, they will be included in the resulting calibration.

The second major research issue to be investigated is how this new calibration method should be implemented in the long term as part of routine operations at the LSWT. The calibration method implemented in the long term should be as simple and well-documented as possible so that the calibration can be performed with a minimum of time and effort. This is where the trade-off between complexity and uncertainty will be especially important. A 36-degree-of-freedom calibration will have less uncertainty than a 6-degree-of-freedom calibration. However, if it requires significantly more time and effort but yields a negligible improvement, it's not worth implementing.

## 2.2 Calibration Methodology

With the research issues set forth it is possible to outline the broad methodology used for the calibration. This section will explain the basic models and assumptions that will be used in approaching this problem.

First, this project will use a linear model to relate forces on the balance to encoder counts. The current calibration uses a linear model and this will be retained. However, the model used in this project will consider interactions between forces, which the current calibration does not. If nonlinear interactions are important, these will be apparent in the data and could be quantified as uncertainties. A future nonlinear calibration could be undertaken to address those effects.

Second, the model will specify that the resolving center of the balance is exactly at its nominal location 42 inches above the floor and centered on the turntable. It will also specify that the axes about which the balance measures forces and moments are all mutually perpendicular and intersecting at the resolving center. Any

imperfections in the external balance that deviate from these assumptions will manifest as off-axis terms in the 36-degree-of-freedom calibration matrix or parameter uncertainties.

Third, the forces applied to the balance should be enough to verify linearity over the range of forces and moments commonly applied to the balance during testing. According to the manual, the balance is rated for +/- 1000 lbf in the side, drag, and yaw components, +/- 2000 lbf in the pitch and roll components, and +3000/-1000 lbf in the lift component. The calibration will apply +1200 lbf to the lift component and +/- 1000 lbf to every other component. This is the full rated range of three of the components. Although it is less than the full rated range of the other three components, it is still enough to encompass the range used during most tests. It is also enough to verify linearity and justify extrapolating the linear model to the full rated range.

# 3. CALIBRATION HARDWARE AND INSTRUMENTATION

Chapter 2 gave an overview of the requirements that the calibration system must meet. With these requirements in mind, the first phase of the project involves creating and assembling the physical hardware needed for the project and setting up the instrumentation.

## 3.1 Hardware Overview

The hardware used to calibrate the external balance falls into two broad categories: the fixed attachment point framework and the load application assembly. The fixed attachment point framework is further divided into a strut assembly, two 5-foot frame assemblies, and two 7-foot frame assemblies. The hardware requiring machining was all fabricated at the Low Speed Wind Tunnel machine shop out of 17-4 precipitation hardened (PH) stainless steel.

## 3.2 Fixed Attachment Point Framework

The fixed attachment point framework consists of five major assemblies. There are two 5-foot frame assemblies used to provide fixed attachment points on either side of the resolving center and two 7-foot frame assemblies used to provide fixed attachment points upwind and downwind of the resolving center. In addition, there is a strut assembly used to provide attachment points on the strut, which attaches to the balance itself.

The strut assembly consists of three pieces: a base, vertical arm, and horizontal arm. The base attaches to the strut by three tapped holes with 1/2"-13 threading, which correspond to holes on the pre-existing strut. The strut can be moved up and down in the lower turntable to align the center attachment point with the resolving

center of the balance, 42 inches above the floor of the test section. Figure 3.1 and Figure 3.2 show the assembled and exploded models from SolidWorks.

Figure 3.1: SolidWorks Model of the Strut Assembly

Figure 3.2: Exploded SolidWorks Model of the Strut Assembly

The exploded SolidWorks view shows the eyebolts used to connect the strut assembly to the load application assembly which is in turn connected to the 5- or 7-foot frame assemblies. The eyebolts attach to the strut assembly by 3/8"-16 tapped holes located one foot in each direction from the resolving center. At the resolving center itself, a long eyebolt goes through the horizontal and vertical arms, connecting the two.

Based on the finite element analysis from SolidWorks, a 1000 lbf force at one of the outer attachment points of the horizontal arm is the worst case scenario for both von Mises stress and displacement resulting in changes to moment measurements.

This loading configuration creates a combined loading with 1000 lbf drag and 1000 ft-lbf yaw. This loading condition creates a total displacement at the attachement point of approximately 0.5 inches. However, the majority of that displacement is in the direction of the applied force and does not change the moment arm vector. The component of the displacement parallel to the moment arm is only about 0.03 in, meaning that the force goes from being applied at a distance of 12 in from resolving center to 11.97 in, a 0.25% change. Figure 3.3 shows the results of the displacement simulation.

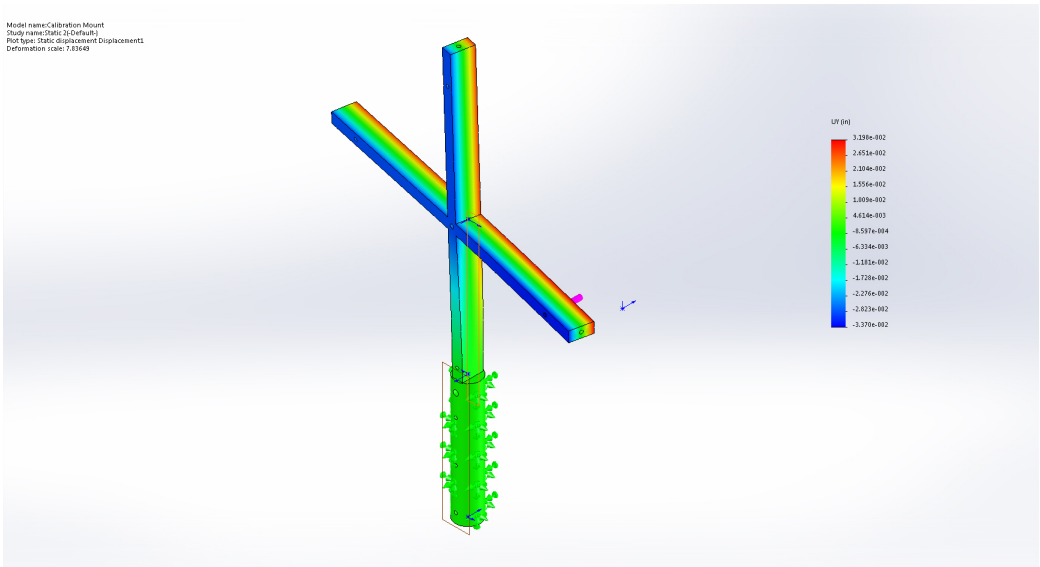


Figure 3.3: Deflection Parallel to the Moment Arm for a 1000 lbf Drag, 1000 ft-lbf Yaw Combined Loading

This same loading condition is also the worst case scenario for total von Mises stress and the analysis shows that the maximum stress is 75.3 ksi, which gives a factor of safety of approximately 2.4 with respect to yielding for 17-4 PH stainless steel. Figure 3.4 shows the results of the stress analysis. The majority of the lower

vertical arm shows a von Mises stress of only about 35 ksi. The maximum value of 75.3 ksi is difficult to see in the figure because it only occurs at a small stress concentration near the tapped hole at the bottom of the vertical arm.

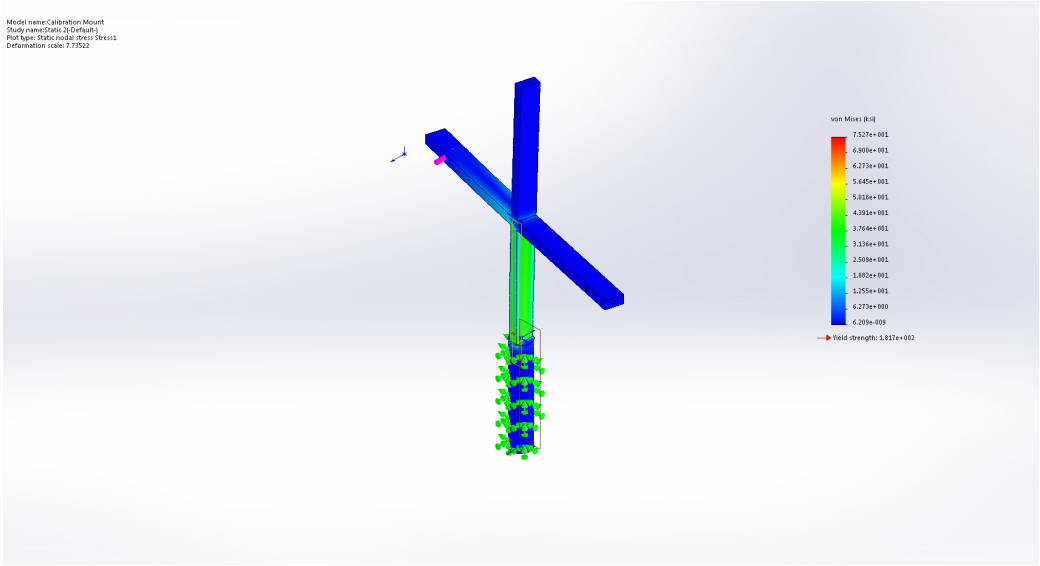


Figure 3.4: Von Mises Stress for a 1000 lbf Drag, 1000 ft-lbf Yaw Combined Loading

The raw materials for both the horizontal and vertical arms were 1" x 2" stainless steel bars. A notch was cut in the center of each bar to allow the two pieces to fit together. The bottom of the vertical arm was turned on the CNC lathe to a cylindrical shape matching a hole on the top of the base. Once the vertical arm was inserted into the base, it was held in place by a dowel pin inserted into a hole drilled through both pieces perpendicular to the vertical arm. The completed strut assembly mounted on the external balance is shown in Figure 3.5

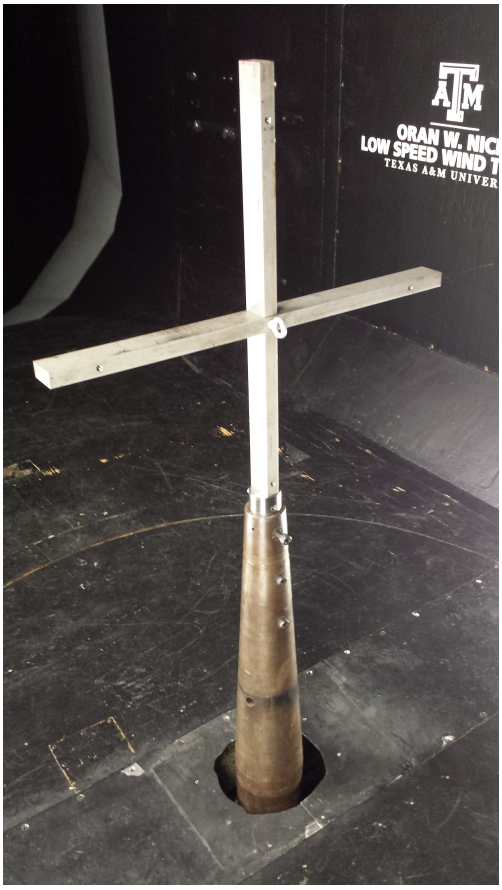

Figure 3.5: Completed Strut Assembly Mounted on the Balance

The 5-foot and 7-foot frame assemblies are used to provide fixed attachment points to the sides, upwind, and downwind of the resolving center. Each one consists of a center piece with three tapped holes and side pieces that connect to attachment points on the walls, roof, and floor of the wind tunnel. Figure 3.6 shows a SolidWorks model of the 7-foot assembly.

Figure 3.6: Assembled and Exploded Views of the 7-Foot Frame in SolidWorks

Apart from the length, the main difference between the 5-foot and 7-foot frames is that the 5-foot frames have one long continuous piece on each side, whereas the 7-foot frames are broken into two pieces. This is because the 5-foot frames can be inserted in place from the side, whereas the 7-foot frames can't. The ability to break the 7-foot frames into smaller pieces makes it easier to insert them into the fixed attachment points inside the LSWT test section. The 5-foot frame is shown in Figure 3.7.

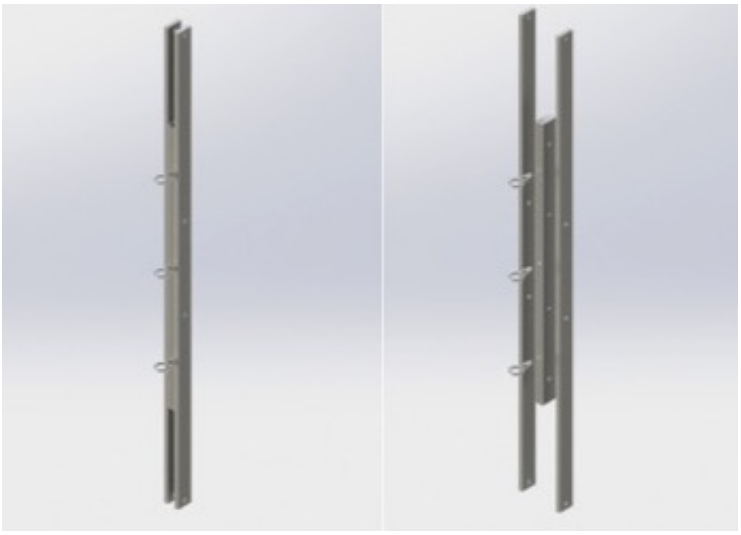

Figure 3.7: Assembled and Exploded Views of the 5-Foot Frame in SolidWorks

These frames attach to pre-existing attachment point on the wind tunnel upwind, downwind, and to the sides of the resolving center. Figure 3.8 shows one of the downwind attachment points.

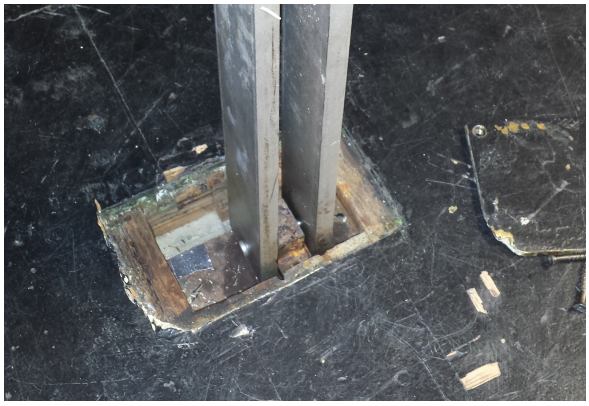

Figure 3.8: Pre-Existing Attachment Point on the Floor of the Test Section Downwind of the Resolving Center

## 3.3 Load Application Assembly

This section describes the load application assembly. The design changed from initial entries in the fall of 2015 to the final data collection in February 2016. The first part of this section describes the final design that was used and the last part describes the initial design before the modification. While only data from the final design are used in the calibraiton, the details on the early design and its shortcomings might be useful for anyone trying to troubleshoot a similar application of a load cell.

The load application assembly serves two major purposes: to apply a force between the strut assembly and 5- or 7-foot frames and to accurately measure the force applied. The first purpose is accomplished by two pneumatic cylinder. The second is accomplished by a strain gauge in line with the force applied. Figure 3.9 shows a SolidWorks rendering of the load application assembly. The only components of the assembly that require fabrication are the two plates on each each of the cylinders. All other components are simply ordered and assembled.

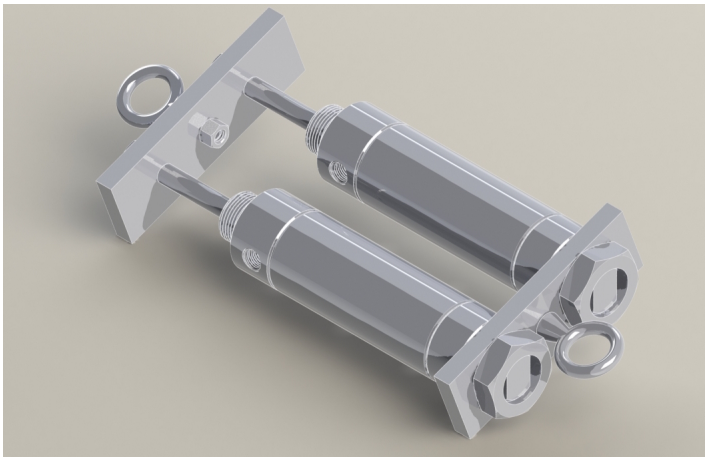

Figure 3.9: Load Application Assembly Rendering

The cylinders are air-retract, spring-extend pneumatic cylinders with a 2-inch diameter bore and 2-inch stroke length, part number 6498K316 on McMaster-Carr [5]. They are rated for a maximum pressure of 250 psi and deliver 620 lbf of force at 200 psi. With two cylinders in parallel, they will nominally deliver a force of 1240 lbf at 200 psi. The complete load application assembly is shown in Figure 3.10 with the strain gauge in series with the two cylinders.

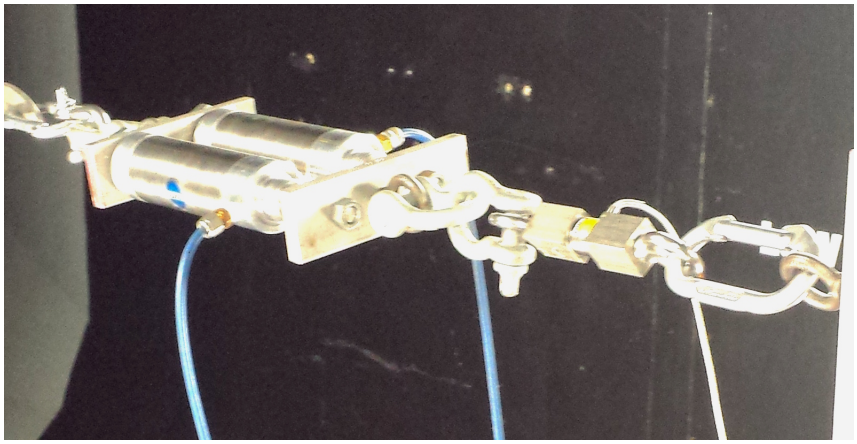

Figure 3.10: Final Load Application Assembly

The strain gauge is an LCM325 tension and compression load cell manufactured by Futek Advanced Sensor Technologies and rated for 2000 lbf [6]. The load cell excitation can be any voltage up to 18 V. The load cell nominally outputs a 1.3 mV/V signal at the full 2000 lbf rated capacity, but a more exact value varies from one unit to another and comes with the factory calibration. This means that with the full 18 V of excitation, the load cell will nominally output only 11.7 mV at 1000-lbf. This means that the resolution of the analog-to-digital converter and signal-to-noise ratio are significant considerations.

The load cell has 3/8"-24 threading on both sides and the eyebolts used to connect

it to the balance and pressure cylinders have 3/8”-16 threading. The adapters in between the eyebolts and load cell are made of a small block of stainless steel and are machined so that the holes are perfectly centered on each other, which prevents a moment from being created in the adapter itself. The 3/8”-16 threads require a hole from a 5/16” drill bit and the 3/8”-24 threads require a hole from a slightly larger Q sized drill bit. To make sure the holes were centered, the material was placed on the mill and a 5/16” hole was drilled all the way through the adapter. Leaving everything clamped in place, the drill bit was changed from a 5/16” bit to a Q sized bit, then the larger hole was drilled halfway through the block. After this, each side was tapped with the appropriate threading, yielding an adapter with two concentric tapped holes. This is important because any moment on the load cell can distort the data, so it needs to be in pure tension. The load cell and adapters are shown in Figure 3.11.

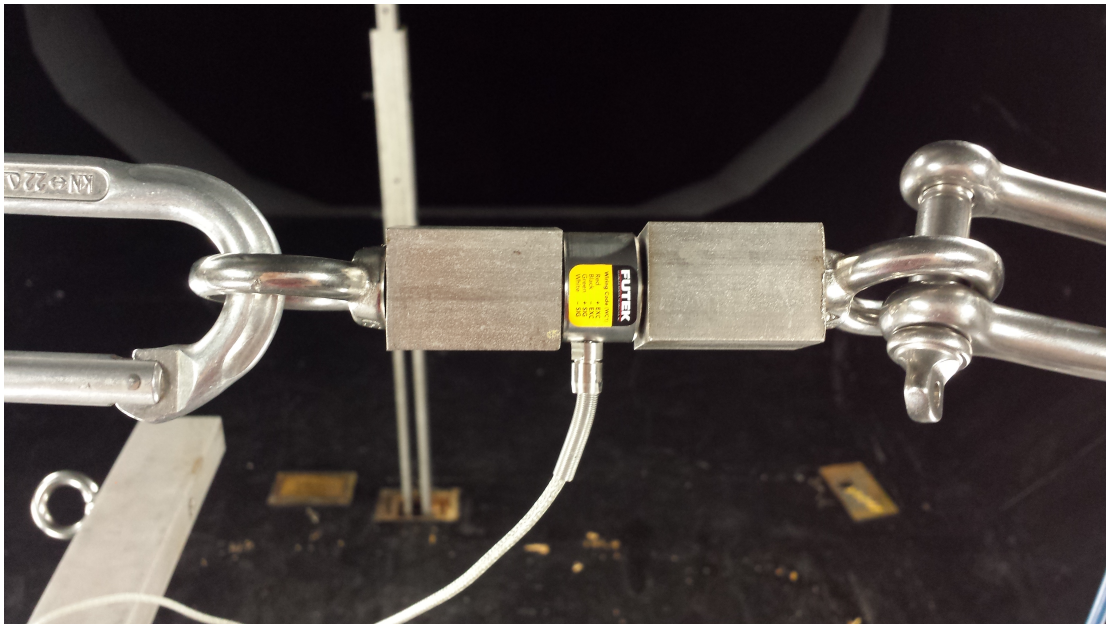

Figure 3.11: Load Cell with Adapters

The load cell is placed in series with the cylinders so that it reads the same force that is transmitted to the balance. The cylinders, load cell, and strut assembly are all connected with eyebolts so that none of the connections can transmit a moment. The mass of the load application assembly does not affect the readings of the external balance as long as it doesn't change during the run. At the beginning of the run, the balance is zeroed. The balance doesn't read absolute forces and moments, but changes in forces and moments since the last zero. The only change in force during the run is in line with the axis of the pneumatic cylinders and load cell. Steel wire, a turnbuckle, and an assortment of shackles and carabiners connect the rest of the span not covered by the cylinders and load cell.

The original load application assembly design that was found to be unsuitable because the load cell connected directly to the plate connecting the pneumatic cylinders. This inadvertently created a moment in the load cell, which distorted the data from some of the initial runs. A SolidWorks model of the original design is shown in Figure 3.12.

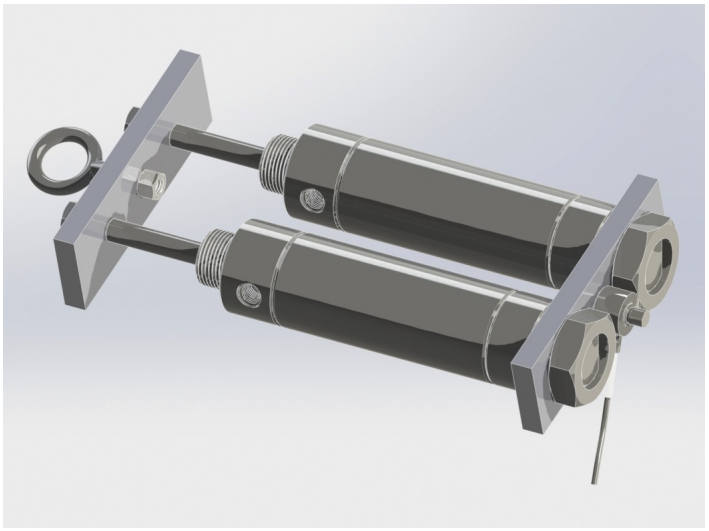

Figure 3.12: Original Load Application Assembly Rendering

Before the final test, the original load application assembly was compared to the moment-reducing design in an Instron tensile tester and the readings were found to have a difference of up to 3% due to moments generated between the cylinders and the load cell. The tensile test is shown in Figure 3.13.


Figure 3.13: Instron Tensile Tester Comparison Between Original Design (left) and Moment-Reducing Design (right)

## 3.4 Instrumentation

The instrumentation used for the project went through several major changes as the experimental procedure was refined. The first portion of this section will explain the final instrumentation set-up that was used to collect the data that is explained in further sections. The second portion of this section will give details on

instrumentation set-ups that were tried or considered, but found to be unsuitable.

The most important piece of instrumentation used was the analog-to-digital converter (ADC) and the specific model used for the final data collection was the ADS1115 from Texas Instruments. Two different ADCs were used, one for strain gauge excitation and one for signal, and this model was chosen primarily for its resolution. It offers 16-bit resolution over a 0 to 5 V range with an additional programmable gain amplifier that can amplify the incoming signal up to 16x. With no amplification, the ADS1115 can only resolve 0.0763 mV differences but the full 5 V range can be used. With maximum amplification, the ADS1115 can resolve 0.00477 mV differences in the signal but the range is limited to 0.3215 V. Using the manufacturer's calibration for the load cell, this voltage resolution gives 1.65 lbf of force resolution with 4.0 V excitation and 0.37 lbf of resolution with 18 V excitation. Each ADC has four input channels, but cannot use different amplification gains for different channels. This is why two different ADCs were used, one with high gain to resolve small differences in the signal and one with no gain to allow the excitation voltage to use the full range.

These ADC chips were controlled by an Arduino Uno using the I2C protocol, which uses two wires to supply power to the ADCs and two wires to send and receive data. The I2C protocol uses a master-slave architecture where the master devices addresses slave devices with an 8-bit address. The ADS1115 has an address pin and when this pin is connected to ground, the chip responds to a different address than when the address pin is connected to the positive power supply voltage. Figure 3.14 shows the Arduino Uno connected to the two ADCs, which are in turn connected to the excitation and signal wires of the load cell.

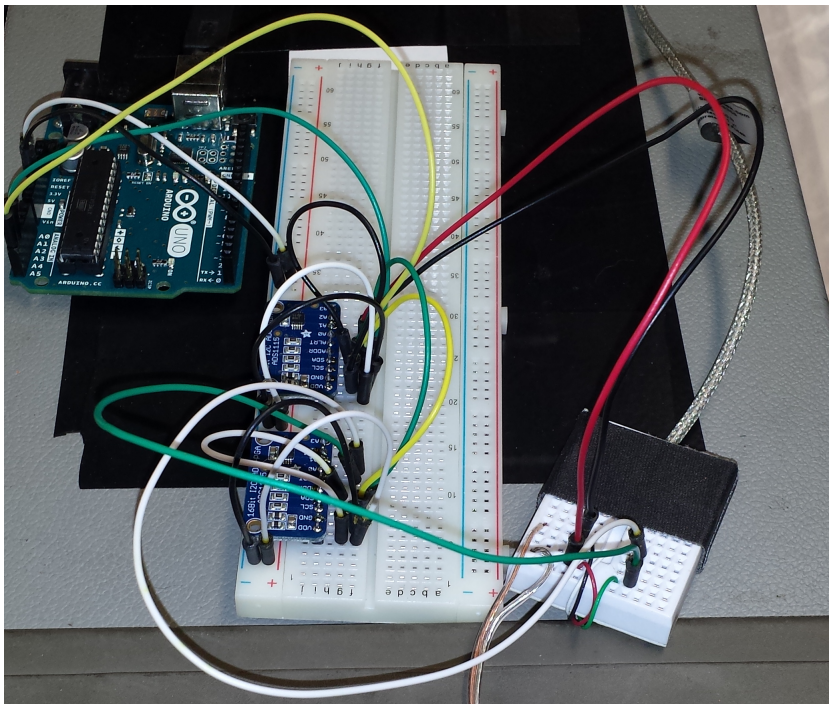

Figure 3.14: Analog-to-Digital Converters with Arduino Controller

For the connections from the Arduino to the ADCs, black wires are used for ground, white wires are used for +5 V, yellow wires are used for the SCL wire of the I2C protocol, and green wires are used for the SDA wire of the I2C protocol. The wires connecting the ADCs to the load cell match the colors of the load cell's wires. Negative and positive excitation voltage are black and red, respectively, and negative and positive signal voltage are white and green, respectively.

The excitation voltage to the load cell is supplied by a DC power supply, which is set to a nominal value. The excitation is read back in by an ADC to detect any differences between the actual and nominal excitation voltages. The Arduino Uno reads in the digital data from the ADCs as a 16-bit integer for every sample, converts it to a voltage, and sends the voltages along with the original raw integers through a USB serial port to a computer. A MacBook Air was used to read in the data from

the Arduino for the final data collection, but the serial protocol is very universal and any computer could be used.

# 4. CALIBRATION DATA ACQUISITION

## 4.1 Test Matrix Generation

Test matrix generation is the first major problem to be solved in the data acquisition phase of the project. The test matrix shows exactly how the test section will be configured for each run with enough specific detail to actually conduct the test.

The test matrices for most tests give details on the dynamic pressure, angle of attack, model configuration, etc. The test matrix for the calibration uses different information to describe the configuration, but serves the same purpose. For the calibration, the test matrix will provide the attachment points of the load application assembly, how the strut assembly will be oriented, and what excitation voltage will be used for the load cell.

There are many possible combinations of runs that can measure all six components of the balance. This calibration will use four sets of seven runs, with each set measuring all six balance components, plus four runs of dead-weight checks. After the initial 32 runs, additional runs will be conducted as needed. This test matrix is shown in Table 4.1.

| Run | Frame | Excitation | Components |
|---|---|---|---|
| 101 | Rear | 18.0V | +Pitch, -Drag |
| 102 | Front | 18.0V | -Pitch, +Drag |
| 103 | Left | 18.0V | -Roll, -Side |
| 104 | Right | 18.0V | +Roll, +Side |
| 105 | Left | 18.0V | +Drag, -Side, +Yaw |
| 106 | Left | 18.0V | -Drag, -Side, -Yaw |
| 107 | Upper | 18.0V | -Lift |
| 201 | Rear | 18.0V | -Pitch, -Drag |
| 202 | Front | 18.0V | +Pitch, +Drag |
| 203 | Left | 18.0V | +Roll, -Side |
| 204 | Right | 18.0V | -Roll, +Side |
| 205 | Right | 18.0V | -Drag, +Side, +Yaw |
| 206 | Right | 18.0V | +Drag, +Side, -Yaw |
| 207 | Upper | 18.0V | -Lift |
| 301 | Dead Weight | n/a | +Lift, -Pitch |
| 302 | Dead Weight | n/a | +Lift, +Pitch |
| 303 | Dead Weight | n/a | +Lift, +Roll |
| 304 | Dead Weight | n/a | +Lift, -Roll |
| 401 | Rear | 4.0V | +Pitch, -Drag |
| 402 | Front | 4.0V | -Pitch, +Drag |
| 403 | Left | 4.0V | -Roll, -Side |
| 404 | Right | 4.0V | +Roll, +Side |
| 405 | Left | 4.0V | +Drag, -Side, +Yaw |
| 406 | Left | 4.0V | -Drag, -Side, -Yaw |
| 407 | Upper | 4.0V | -Lift |
| 501 | Rear | 4.0V | -Pitch, -Drag |
| 502 | Front | 4.0V | +Pitch, +Drag |
| 503 | Left | 4.0V | +Roll, -Side |
| 504 | Right | 4.0V | -Roll, +Side |
| 505 | Right | 4.0V | -Drag, +Side, +Yaw |
| 506 | Right | 4.0V | +Drag, +Side, -Yaw |
| 507 | Upper | 4.0V | -Lift |

Table 4.1: External Balance Calibration Test Matrix

The 10x, 20x, 40x, and 50x series runs are four sets that each include seven runs and can independently calibrate the balance, but in different ways. The 10x and

40x runs use attachment points above the resolving center for the pitch, drag, roll, and side measurements and the 20x and 50x runs use attachment points below the resolving center. The 10x and 20x runs are the same physical configuration as the 40x and 50x runs, respectively, but with a different excitation voltage. The 18.0V excitation runs have the advantage of maximizing the signal of the load cell, which gives the best resolution and signal-to-noise ratio. The disadvantage is that since the excitation voltage is outside the range of the ADC, it must be read in manually with a multimeter and compared to the nominal value. The 4.0V excitation runs require less work by the person conducting the calibration because the excitation voltage can be read automatically by the ADC. However, the signal generated by the load cell will be much lower, decreasing the resolution. However, even with the lower excitation, the ADC should be able to resolve 1.65 lbf differences in force and using a time-averaged signal should reduce the noise. Both excitation voltages should give similar results and the test matrix is designed to confirm this.

The test matrix also includes the 30x series runs, which are dead weight runs. The usefulness of dead weight is very limited because with the hardware developed for this project, it can only be used to measure the lift, roll, and pitch components. It is also done over a very small range, only +/- 40 lbf. It isn't practical to use dead weight to conduct the full calibration. However, it's useful to confirm the results obtained using the pressure cylinders and load cell. It's extremely simple and doesn't have the difficulties associated with creating and reading an analog signal.

The x07 runs measure lift upwards from the zero point to the maximum force that the cylinders can apply. The cylinders are rated up to 250 psi and exert approximately 1200 lbf combined at this pressure. There is no run measuring negative lift from the zero point because there are no attachment points below the balance. However, it's important to note that the zero point is arbitrary because the balance

is always zeroed with the wind off every time a model is installed. That means that applying +/- 1000 lbf to the empty strut is physically the same as putting a 1000 lbf model on the strut, zeroing it, and applying 0 to 2000 lbf of upward force. The only difference is where the encoders start counting.

The arbitrary zero point is a fundamental aspect of the linear model used to convert encoder counts to forces and moments. If the model were nonlinear and the derivative of the function relating encoder counts to forces and moments were not constant, then the zero point would not be arbitrary. However, the design of the balance, especially its small deflections under load indicate that it should have a linear response. This project seeks to retain the linear model, but find the linear coefficients more accurately and efficiently. If the calibration should be nonlinear, this would be evident from the results.

Lift is measured over a smaller total range than the other components because the current equipment doesn't allow downward force. This might cause increased uncertainty compared to the other components that are measured over a larger range. If so, the linear regression results will account for this uncertainty.

## 4.2 Data Acquisition Software

The data acquisition software for the calibration is written in Python. There are three pieces of code used during data acquisition. The first is the module used to store and analyze the data, the second is used to collect data from the load cell and external balance, and the third is used to collect data from dead weight loads on the external balance.

The most important part of the data acquisition software is the pyBalCal module, which is essentially a data structure used for storing calibration data. The full code is show in Appendix A. This module is the link between the data acquisition and

data reduction software. It includes two classes, dataPoint and calibrationRun.

The dataPoint class stores individual data points and has several methods to perform some basic analysis of the raw data. It stores the excitation and signal voltages, encoder counts from all six encoders, nominal and hand-measured values for both the force and moment-arm vectors in inches, a string for the load cell serial number, the manufacturer's calibration coefficient relating the load cell signal [mV/V] to force [lbf], a signal gain adjustment coefficient based on the shunt calibration, a floating point number representing the time the data point was recorded, and a string variable called 'appendix' that is a placeholder for potential future modification of the class.

The dataPoint class includes a method to output the force magnitude based on the raw data and another to output the force vector. Another method returns the moment vector by simply taking the cross product of the moment-arm vector with the force vector. One method concatenates the force and moment vectors and another reorders the encoder counts to match the order of the concatenated force and moment vector. Two final methods involve exporting and importing data from the class. These methods use the JSON module, which is a standard module included with Python distributions. It essentially converts a given data structure to a string, which can easily be stored in a file, and converts a given string back into the original data structure.

The second class in the pyBalCal module is called calibrationRun and it consists primarily of an array of instances of the dataPoint class and a few methods for storing and analyzing them. The fit_linear method finds the linear fit for a given component of the balance. This gives the elements of the 6-degree of freedom calibration. Another method called fit_multilinear gives the full $6 \times 6$ matrix based on the run that relates the 6 forces and moments to the 6 encoder counts. This is the 36-degree of

freedom calibration based on a simple least squares fit. The class also includes two methods to export the whole run to a file or import it from a file. These methods use the JSON methods of the dataPoint instances belonging to the class.

The second part of the data acquisition software is called XBAL_CAL.py and is used to collect data from the load cell and external balance and store it in the dataPoint and calibrationRun classes. The main difference between this code and the pyBalCal module is that the pyBalCal module is much more general whereas this code is more specific to the instrumentation used for this calibration. For example, in this calibration, the signal from the load cell is read in through the serial port and the encoder counts are received over TCP/IP from the external balance computer. Someone conducting another external balance calibration in the future might prefer to run the code on the external balance computer as receive the encoder counts directly from the serial port, but collect the analog voltages on a different computer and receive them over TCP/IP. The pyBalCal module is more likely to be re-used in the future than the data acquisition code.

For this reason, the XBAL_CAL.py code is stored on the LSWT archive server but not included here as an appendix. In general terms, the code is a simple console program with three methods. There is a method to get the encoder counts from the external balance computer over TCP/IP, which returns an array of integers. Another method gets a 1000 sample time average of both excitation and signal voltage from the Arduino over the serial port and returns the voltages, raw integers, and an estimation of the force based on the manufacturer's calibration. The third method inputs the original force and moment-arm vectors along with a yield angle and returns new force and moment-arm vectors based on rotation about the yaw axis. Deflections due to yawing moment cause a more significant change in geometry than other forces and moments and the code allows the test engineer to account for them.

After defining these three methods, the code uses a loop to collect five data points per run. At the beginning of each iteration, it asks the user to adjust the pressure and allows them to input a yield angle and excitation signal if the excitation signal is outside the range of the ADC. Next, it begins to read encoder counts from the external balance and waits for all six components to reach steady state. Steady state is defined by a maximum rate of change of encoder counts, a tolerance which can be tuned by the engineer conducting the test. Once this threshold is reached, it considers the balance in steady state records the encoder counts. After the encoder counts are recorded, it records the signal from the ADC. The time-averaged ADC signal takes approximately 60 seconds to acquire with the equipment used for this calibration. After this, the code takes another encoder count reading and averages the first and second readings to get an estimate of the encoder count during the signal time-averaging.

After collecting the data, XBAL_CAL.py stores it in an instance of the dataPoint class. It also records the nominal and hand-measured geometry of the data point, along with the gain adjustment based on the shunt calibration. The measurements and shunt gain are measured before the run and stay constant for all five data points. The hand measurements are simply done with a tape measure and all dimensions are recorded in inches. The engineer measures the three components of the vector from the resolving center to the attachment point and a vector from the attachment point in the direction the force is applied. The code normalizes the force vector automatically, so the hand-measured vector doesn't necessarily have to span the entire length of the load application assembly to the other attachment point. The hand measurements only change during the run if the user inputs a yield angle due to yawing moment. The code also records the time at which the data point was collected. As the code collects points, it adds them to an instance of the calibrationRun class,

which is exported to a JSON file at the end of the run.

The third piece of code is like XBAL_CAL.py, but much simpler, and used to collect data for dead weight runs. Compared the XBAL_CAL.py, the method to receive encoder counts from the balance is exactly the same but the method to interface with the ADC is removed because it is not needed. The geometry of the force and moment-arm vectors are entered before the run. The program still iterates through five data points and simply prompts the user to enter the load applied. The program sets the excitation voltage to 1.0V and calculates the signal voltage equivalent to the entered load. It stores the data and outputs the force magnitude so that the user can verify it.

## 4.3 Data Acquisition

Only the final set of data collected during this project from February 18th to 23rd, 2016 is used for the calibration. Once all the hardware and instruments were configured, the first step in the actual data collection is the shunt calibration gain adjustment. This is a method of simulating a load on a load cell using a resistor of a given value which yields a correction factor used to adjust the gain of the measurement system [7]. The calibration sheet from the manufacturer gives the resistance of the shunt resistor, the terminals between which it should be connected, and the signal in mV/V that should be read. In the case of the load cell used for the external balance calibration, the shunt resistance is $100k\Omega$, which should be connected between the negative signal and negative excitation terminals, and the signal should be 0.8819 mV/V. During the shunt calibration, there should be no physical load on the load cell.

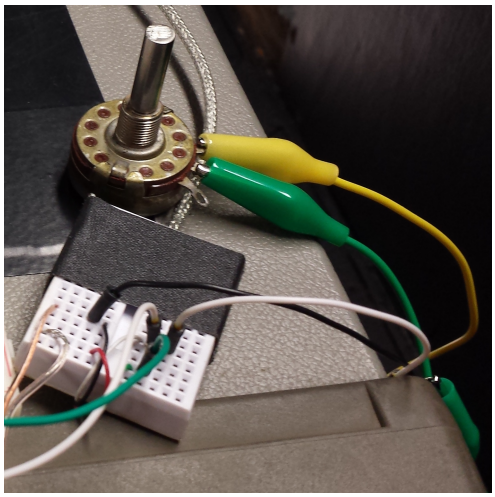

Figure 4.1: Shunt Resistor Connected Across Negative Power and Excitation Terminals

Figure 4.2: Load Cell with Physical Load Removed for Shunt Calibration

With the shunt calibration complete, the next step is to attach the load application assembly between the strut assembly and frames. The load cell is attached closest to the strut assembly. The pneumatic cylinders connect to the load cell and

the remaining distance is spanned with steel wire, carabiners, shackles, and a turnbuckle. Once everything is connected, the turnbuckle is tightened such that at full load, the clyinders won't use their full 2 inches of travel. Then, 70-80 psi is applied to put 300-400 lbf of tension in the assembly. The configuration for Run 101 is shown in Figure 4.3.

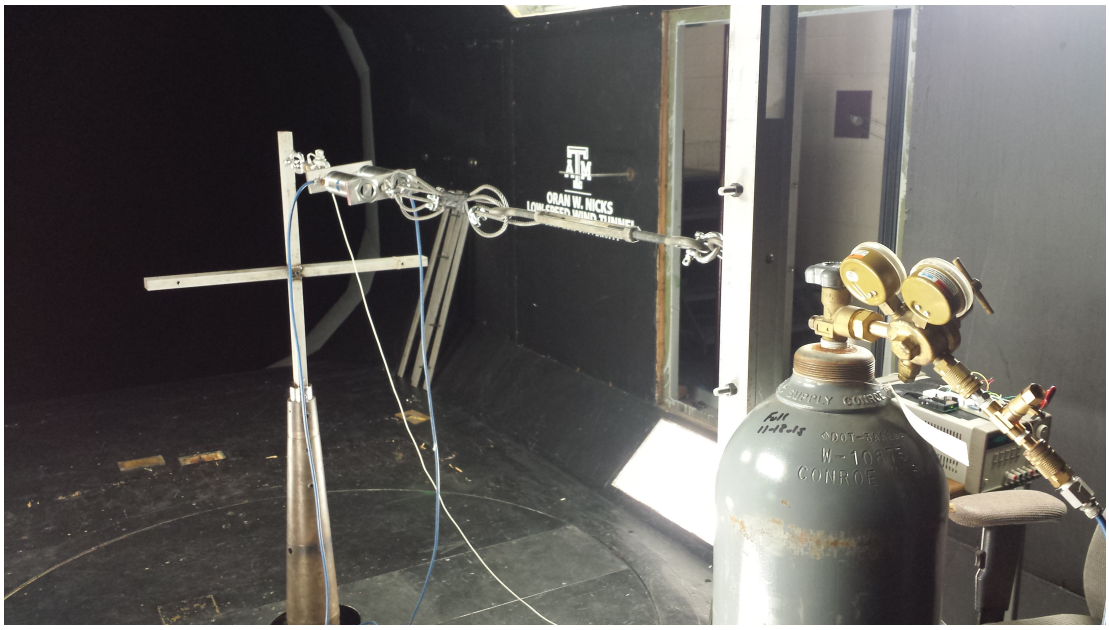

Figure 4.3: Configuration for Run 101, Combined Drag and Pitching Moment

For each run, five data points are collected. The exact pressure applied to the cylinders for each data point isn't important because the exact force is measured by the load cell. The data points simply need to be distributed evenly enough across the full force range to give a good representation of the linear relationship between forces and encoder counts. Runs 101 through 207 use 18.0 V of excitation for the load cell. Since this is outside the range of the ADCs used for this test, the excitation voltage is read in manually using a multimeter for each point. For many of the points, the

difference between the nominal and actual excitation voltage was indistinguishable by the multimeter.

After the first 14 runs with 18.0 V excitation, four runs of dead weight tests were conducted. Two plates, each weighing 20 lbf, were used to apply a force to the lift, roll, and pitch components. The plates were hung from the attachment points using a loop of steel wire. The wire loop without any weight was attached to the eyebolt to get a 0 lbf reading, the 20 lbf and 40 lbf readings were collected by adding plates. After these three points, 20 lbf and 0 lbf were collected again on the way down to give five total points per run just like with the pneumatically applied force. Figures 4.4 and 4.5 show dead weight configurations for Run 304.

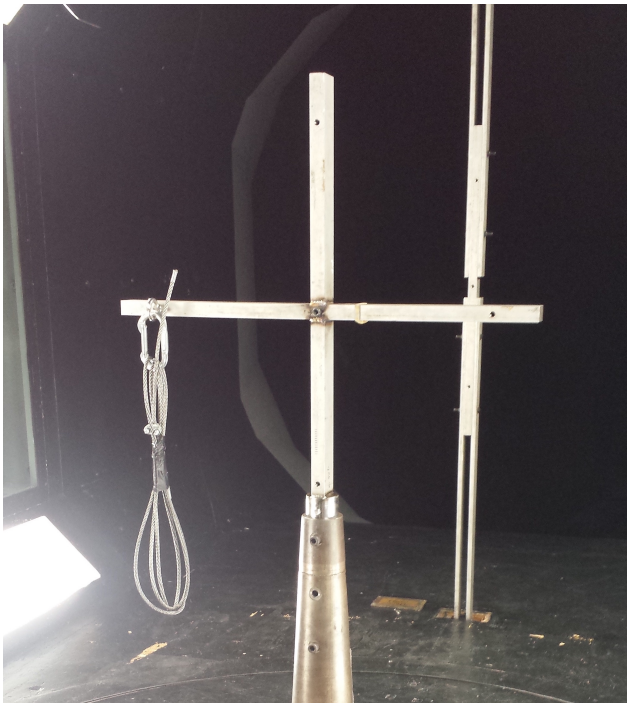

Figure 4.4: Dead Weight Configuration for Run 304, 0 lbf

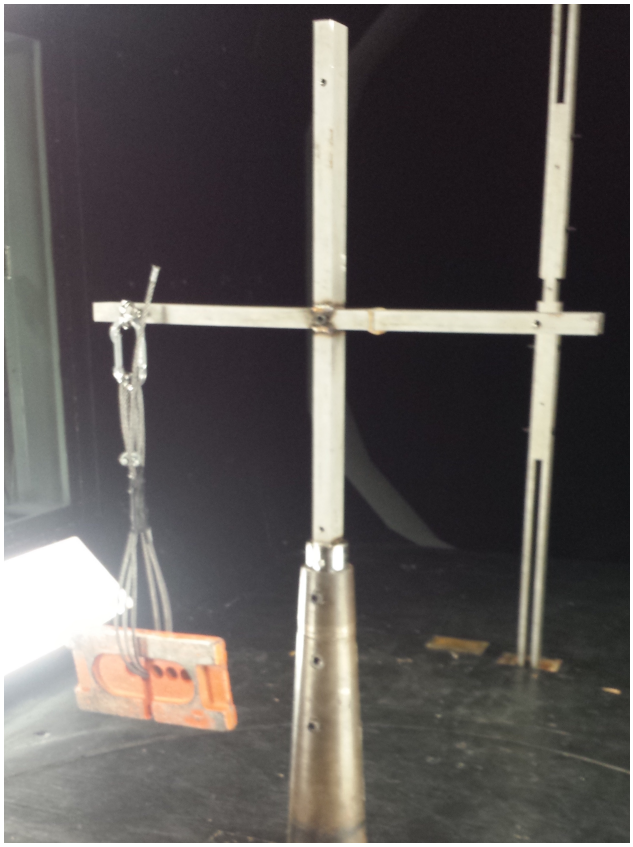

Figure 4.5: Dead Weight Configuration for Run 304, 40 lbf

After the dead weight runs, a new shunt calibration and Runs 401 through 507 were conducted using 4.0 V of excitation rather than 18.0 V. With these runs, the excitation signal is read directly by the second ADC without any need to manually check it with a multimeter. After the 4.0 V runs, the extra time available was used to conduct 8 additional 18.0 V excitation runs focused on drag-pitch and side-roll loading conditions.

# 5. CALIBRATION DATA REDUCTION AND ANALYSIS

## 5.1 Data Reduction Overview

The data reduction process has two key steps. First, the encoder counts from different runs must be shifted to a common zero point so that they can be compared to counts from other runs. Second, the runs must be combined into one large calibration data structure so that a linear regression can be performed on the entire data set.

The first step in this process is accomplished by the linear fit method mentioned in the data acquisition chapter. The code asks the user to select a directory with the runs to be analyzed. The associated files are then loaded into an instance of the calibrationRun class. The code then performs a linear regression between the force and counts for each component, then subtracts the y-intercept. This has the same effect as if the balance had been zeroed at a different point corresponding to no force on the strut assembly. The equivalent physical zero would be difficult to implement because if the balance is zeroed without the load application assembly attached, it will read a force in the lift component when it's attached even if the cylinders are applying no force in that direction. In the linear model, only the slope of the linear relationship is important. However, to compare any two points, the points must have a common zero point. This code shifts the encoder counts so that the condition of zero counts in each component corresponds to zero force.

Once the encoder counts are all adjusted to a common zero point, all the adjusted points are added to a calibration class. The calibration class inherits the calibrationRun class, but adds nothing to it. This means that the calibration class is essentially the same as the calibrationRun class. The only reason a new class is created is to differentiate based on their intended purposes. The fit_linear and

fit_multilinear methods of the this class can be used to perform a regression on the entire data set.

## 5.2 Results

The conclusions and recommendations of this project as based primarily on the results of the 18.0 V excitation runs. The dead weight and 4.0 V runs are used to show that the results remain consistent under difference testing conditions. One of the original objectives of this project was to compare the results of a 1, 6, and 36 degree-of-freedom calibration. The Figure 5.1 shows the results of the 1 degree-of-freedom calibration.

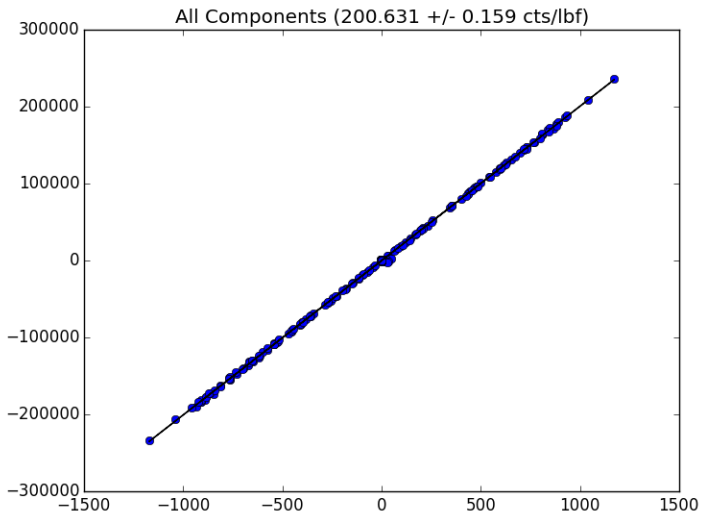


Figure 5.1: 1 Degree-of-Freedom Regression from Runs 101 through 207

For the 1 degree-of-freedom case, the calibration continues to use the same coefficient to relate encoder counts to forces and moments, but the coefficient is allowed to deviate from 200 counts/lbf. Figure 5.1 shows that the data set is very linear with

no evidence of nonlinear effects. Based on the regression, the original coefficient is approximately four times the standard error away from the new coefficient of 200.631 +/- 0.159. However, the standard deviation and difference from the original coefficient are both very small. If this calibration were implemented, it would only be a 0.3155% change from the current calibration.

The second objective of the project was to find the results of a 6 degree-of-freedom calibration, which is given in Figure 5.2 for the 18.0 V excitation runs.

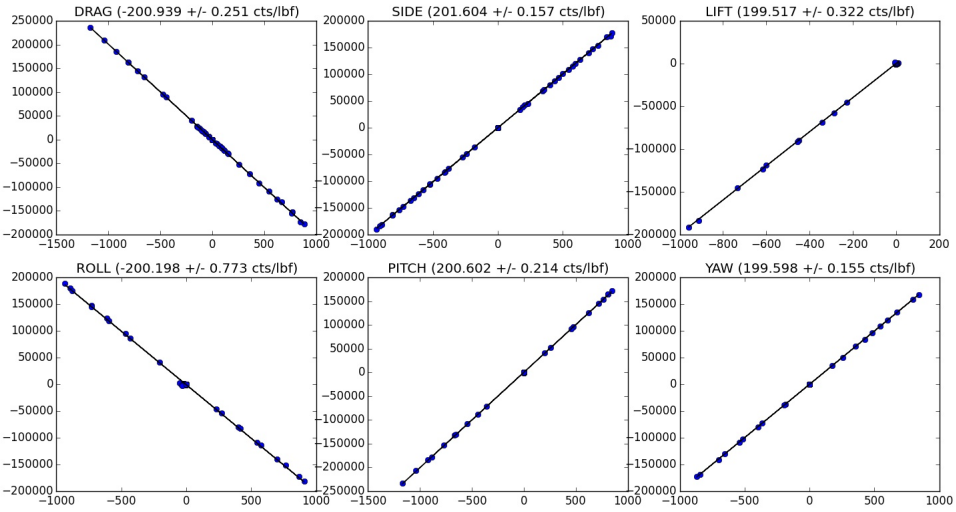


Figure 5.2: 6 Degree-of-Freedom Regression from Runs 101 through 207

In this calibration, each component is calibrated independently. The results, shown in Table 5.1, indicate larger deviations from the original calibration coefficients with the largest being in the side force component. However, all components have coefficients within 1% of the original calibration. All components are at least one standard deviation away from the original coefficients except for roll.

| Component | Coefficient [cts/lbf] | Standard Deviation [cts/lbf] |
|---|---|---|
| Drag | -200.939 | 0.251 |
| Side | 201.604 | 0.157 |
| Lift | 199.517 | 0.322 |
| Roll | -200.198 | 0.773 |
| Pitch | 200.602 | 0.214 |
| Yaw | 199.598 | 0.155 |

Table 5.1: 6 Degree-of-Freedom Coefficients

Finally, results of the 36 degree-of-freedom calibration based on the 18 V excitation runs are shown below.

$$\begin{bmatrix} -200.79 & 0.05 & 0.00 & -0.33 & -0.55 & -0.02 \\ 0.00 & 201.62 & 0.66 & -1.19 & 0.14 & 0.42 \\ 0.13 & -0.10 & 199.39 & 0.07 & -0.25 & -0.14 \\ 0.01 & -0.34 & -0.06 & -199.99 & -0.03 & 0.03 \\ -0.91 & -0.02 & -0.02 & 0.04 & 200.64 & 0.05 \\ 0.05 & -0.50 & -0.29 & -1.08 & 0.079 & 199.62 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} e_D \\ e_S \\ e_L \\ e_R \\ e_P \\ e_Y \end{bmatrix}$$

This model of the external balance allows for the possibility of interactions between components. This matrix relates forces and moments to encoder counts because it is much easier to visualize the results this way. For actual implementation, it would be necessary to invert the matrix so that it relates encoder counts to forces and moments.

The interaction terms of the 36 degree-of-freedom calibration are very small and likely to be caused by measurement and random error. The following matrix gives the standard errors of each element in the calibration matrix. Of the 30 off-diagonal terms, 17 are within a standard error of zero.

$$\begin{bmatrix} 0.164 & 0.146 & 0.350 & 0.172 & 0.165 & 0.267 \\ 0.159 & 0.141 & 0.339 & 0.166 & 0.160 & 0.258 \\ 0.104 & 0.092 & 0.222 & 0.109 & 0.105 & 0.169 \\ 0.478 & 0.424 & 1.018 & 0.499 & 0.480 & 0.774 \\ 0.150 & 0.133 & 0.320 & 0.157 & 0.151 & 0.244 \\ 0.078 & 0.070 & 0.167 & 0.082 & 0.079 & 0.127 \end{bmatrix}$$

# 6. CONCLUSIONS AND RECOMMENDATIONS

The goal of this project was to develop and demonstrate a faster and easier calibration method for the LSWT external balance by taking advantage of modern digital electronics. The recommendation resulting from the data is to implement a 6 degree-of-freedom calibration with the new coefficients given in the Table 5.1. The data from the 36 degree-of-freedom calibration shows that interactions between components are negligibly small and not needed to achieve good accuracy. The current model doesn't allow for interactions and there is not enough evidence to support changing this.

However, in the 6 degree-of-freedom calibration, the coefficients are all more than one standard error away from the nominal 200 counts-per-lbf value. In absolute terms, the new coefficients are still very close to the original ones and implementing the new calibration will represent less than a 1% change in all coefficients. In addition, the current calibration requires using a different sign for two of the components even though the magnitudes are the same. The negative contrasted to positive slopes can be seen in Figure 5.2. Since the calibration already has to treat each component differently, there would be very little improvement in simplicity of implementation forcing the magnitudes of all coefficients to be the same.

The calibration should be conducted with 18 V excitation for the strain gauge because this produces the highest signal and gives the best resolution. If a new ADC with a wide enough range to read 18 V is available, the actual excitation voltage should be read and compared to the nominal voltage. If not, the results of this calibration suggest that just using the nominal excitation voltage is acceptable. If the nominal voltage is used, occasional checks with a multimeter are recommended.

If time allows, a three-day calibration should be conducted once per year. This three-day calibration is estimated to include one day of set-up, one day of testing, a half day of break-down, and a half day of make-up or buffer time. If the results continue to stay consistent over several years, the frequency could be reduced as deemed appropriate. A dead weight test with the strut assembly (Runs 301 through 304) is a good way to quickly check the calibration if a client requests it. However, once a full day and a half are invested in set-up and break-down, it is generally worth conducting a full calibration.

# BIBLIOGRAPHY


[1] Rae, W. H. and Pope, A., *Low-Speed Wind Tunnel Testing*, John Wiley and Sons, Hoboken, NJ, 1984.

[2] Herring, A. K., *Dynamic Pressure Improvement to Closed-Circuit Wind Tunnels with Flow Quality Analysis*, Master's thesis, Texas A&M University, 2015.

[3] Sierra Nevada Corporation, "Sierra Nevada Corporation Completes Wind Tunnel Testing of The Dream Chaser Orbital Crew Vehicle at Texas A&M University," Retrieved from http://www.sncorp.com/AboutUs/NewsDetails/487, April 2012, Accessed: 2015-11-10.

[4] Brockington, P. H., "Instruction Manual for Six Component Mechanical Balance," January 1958.

[5] McMaster-Carr, "McMaster-Carr Online Catalog," Retrieved from http://www.mcmaster.com, Accessed: 2015-11-21.

[6] Futek Advanced Sensor Technologies, "Model LCM325 Specifications Sheet," Retrieved from http://www.futek.com/files/pdf/Product2015-11-21.

[7] Hoffmandd, K., *An Introduction to Stress Analysis and Transducer Design using Strain Gauges*, Hottinger Baldwin Messtechnik GmbH, Darmstadt, Germany.

# APPENDIX A

## PYTHON CODE FOR PYBALCAL MODULE

The pyBalCal module is used by both the data acquisition and data reduction software.

```
import json
import scipy.optimize as opt
import numpy as np

class dataPoint:
    def __init__(self):
        self.signal_raw = 0.0
        self.signal_mV = 0.0
        self.excitation_raw = 0.0
        self.excitation_V = 0.0

        self.counts = [0, 0, 0, 0, 0, 0]
        self.fVec_nom = [0.0, 0.0, 0.0]
        self.rVec_nom = [0.0, 0.0, 0.0]
        self.fVec_hand = [0.0, 0.0, 0.0]
        self.rVec_hand = [0.0, 0.0, 0.0]

        self.futekSerialNum = "unspecified"
        self.futekCal_lbf_mVpV = 1381.2
        self.shuntGainAdjust = 1.0

        self.timeRecorded = 0.0
```

```
        self.appendix = "unused"

    def forceMagnitude(self):
        return (self.signal_mV/self.excitation_V) * self.
            futekCal_lbf_mVpV * self.shuntGainAdjust

    def forceVector(self, geometry="hand"):
        if geometry == "hand":
            fVec_n = np.array(self.fVec_hand)
            fVec_n = fVec_n / np.linalg.norm(fVec_n)
            return fVec_n*self.forceMagnitude()
        else:
            fVec_n = np.array(self.fVec_nom)
            fVec_n = fVec_n / np.linalg.norm(fVec_n)
            return fVec_n*self.forceMagnitude()

    def momentVector(self, geometry="hand"):
        if geometry == "hand":
            return np.cross(np.array(self.rVec_hand)/12.0,
                self.forceVector(geometry="hand"))
        else:
            return np.cross(np.array(self.rVec_nom)/12.0, self
                .forceVector(geometry="nominal"))

    def forceMomentVector(self, geometry="hand"):
        return np.concatenate((self.forceVector(), self.
            momentVector()))
```

```
    def countVectorReordered(self):
        return [self.counts[i] for i in [4, 5, 0, 1, 2, 3]]

    def jsonDump(self):
        return json.dumps([self.signal_raw, self.signal_mV,
            self.excitation_raw, self.excitation_V, self.counts
            , self.fVec_nom, self.rVec_nom, self.fVec_hand,
            self.rVec_hand, self.futekSerialNum, self.
            futekCal_lbf_mVpV, self.shuntGainAdjust, self.
            timeRecorded, self.appendix])

    def jsonLoad(self, jsonStr):
        [self.signal_raw, self.signal_mV, self.excitation_raw,
             self.excitation_V, self.counts, self.fVec_nom,
            self.rVec_nom, self.fVec_hand, self.rVec_hand, self
            .futekSerialNum, self.futekCal_lbf_mVpV, self.
            shuntGainAdjust, self.timeRecorded, self.appendix]
            = json.loads(jsonStr)

class calibrationRun:
    def __init__(self):
        self.dataPoints = []

    def linear(self, x, A, B):
        return A*x+B
```

```
def fit_multilinear(self):
    self.counts = np.array([pt.countVectorReordered() for
       pt in self.dataPoints])
    self.components = np.array([pt.forceMomentVector() for
        pt in self.dataPoints])
    return np.linalg.lstsq(self.components, self.counts)

def fit_linear(self, component):
    i = {'DRAG':0, 'SIDE':1, 'LIFT':2, 'ROLL':3, 'PITCH'
       :4, 'YAW':5}
    counts = [pt.counts[i[component]] for pt in self.
       dataPoints]
    force = [pt.forceMomentVector()[i[component]] for pt
       in self.dataPoints]
    return opt.curve_fit(self.linear, np.array(force), np.
       array(counts))

def jsonDumpToFile(self, filename):
    f_out = open(filename, "w")
    for pt in self.dataPoints:
        f_out.write(pt.jsonDump())
        f_out.write("\n")
    f_out.close()

def jsonLoadFromFile(self, filename):
    f_in = open(filename, "r")
    self.dataPoints = []
```

```
for line in f_in:
    point = dataPoint()
    point.jsonLoad(line)
    self.dataPoints.append(point)
f_in.close()
```