2015

Design of a force sensing system to assist robotic space servicing and exploration operations

Jason Brandon Battin

DESIGN OF A FORCE SENSING SYSTEM TO ASSIST ROBOTIC SPACE SERVICING AND EXPLORATION OPERATIONS

by

Jason Brandon Battin

Thesis submitted to the Statler College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

Master of Science
in
Engineering

Approved by

Dr. Powsiri Klinkhachorn, LCSEE, Committee Chair
Dr. Thomas Evans, MAE
Dr. Roy Nutter, LCSEE

Lane Department of Computer Science and Electrical Engineering

Morgantown, WV
2015

Keywords: strain gage, load cell, contact force, centroid, force reconstruction, Wheatstone bridge, spacecraft mating, rail sensor system

ABSTRACT

DESIGN OF A FORCE SENSING SYSTEM TO ASSIST ROBOTIC SPACE SERVICING AND EXPLORATION OPERATIONS

by Jason Brandon Battin

The focus of this research has been the design and fabrication of a rail sensor system (RSS) that employs an array of commercially available load cells to reconstruct contact forces by determining a centroid of force. The proposed RSS system can be divided into two coherent systems: a mechanical system and an electrical system. The mechanical system is composed of two load cells, two aluminum support structures, and a friction resistant shoulder screw. The electrical system consists of a commercially available USB interface board responsible for capturing and transmitting raw voltage values from each load cell to the data logging software. Computer simulations and ground based testing were conducted and compared to validate the proof of concept and a fuzzy logic control scheme was developed to simulate real-time angle and trajectory corrections based on the output of each load cell. Tests conducted with the Rail Sensor System (RSS) reinforce the concept of reconstructing contact forces using an array of strain gages and their calculated centroid of force. The raw voltage values reported by the load cells contain valuable information that can potentially provide teleoperators and autonomous algorithms the information necessary to determine nominal service vehicle approach angles.

ACKNOWLEDGMENTS

I would like to express my sincere thanks and appreciation to Dr. Powsiri Klinkhachorn for his endless support and guidance throughout this research. His wisdom both inside and outside of the classroom were invaluable to the success of my degree. I would also like to thank Dr. Thomas Evans for providing me with such an amazing opportunity and access to the West Virginia Robotic Technology center lab and equipment. A special thanks to Dr. Roy Nutter for serving on my examining committee and providing me with valuable input during my oral defense. Endless gratitude and love goes to my entire family for their support and encouragement throughout my research and education. If it were not for them this work would not have been possible.

This thesis is dedicated to my wife Nikki and our newborn son Colt -- thank you for your perpetual patience and support throughout this transition. I am truly blessed to have such an amazing family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 | INTRODUCTION

1.1 Robotic Assets in Space

The reliance on robotic assets for spacecraft maintenance has been a key element in the sustainability of the Hubble space telescope, International Space Station (ISS), and upcoming National Aeronautics and Space Administration (NASA) refueling and servicing missions [1]. An international investment in the development of robotic space operations has propelled a growing interest in commercial space exploration [2] [3]. Several successful commercial ISS resupply operations have been carried out by organizations such as SpaceX while Virgin Galactic works to develop the world's first "spaceline for earth" [4]. As robotic technologies continue forward safety and cost considerations drive the advancement of robotic tools to assist with teleoperated and autonomous space servicing operations such as satellite servicing.

On orbit satellite servicing encompasses a vast number of operations including the assembly of structures, spacecraft refueling and repair, and the delivery of cargo. Several challenges must be overcome before a service vehicle can rendezvous, capture, and service a wounded space vehicle as shown in Figure 1.1 [5]. Despite numerous advancements in technology, rendezvous and berthing of spacecraft is still not an exact science [6]. NASA's Satellite Servicing Capabilities Office (SSCO) deputy project manager Benjamin Reed expressed his concern during a 2013 NASA TV interview: "There are, as available on the web, something

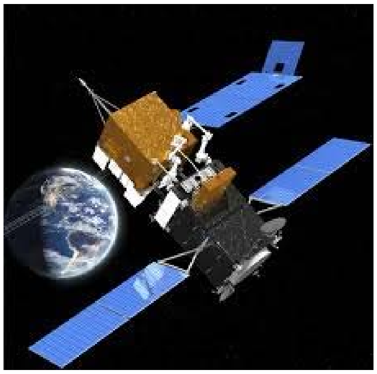


Figure 1.1 | Artist's concept of servicing vehicle servicing a client satellite

like 1000 operational satellites in orbit. Only two of the 1000 are serviceable." Satellites that are not designed to be serviced provide a unique set of challenges that require custom tools and unique operations [3] [6]. Space docking and space berthing are two procedures frequently used in the servicing of orbiting spacecraft. The details of each procedure is provided in the following sections.

1.1.1 Space Docking

Space docking or docking occurs when an incoming spacecraft mates with another spacecraft and the joined unit continues on a controlled trajectory [7]. The docking process typically consists of a soft capture phase followed by a load attenuation phase leading to a final hard docked position. Space docking is not a new concept. "In the late 1950s rendezvous and docking was recognized as necessary for building space stations and assembling vehicles in low Earth orbit to perform exploration missions." [6]. The first successful space dock was achieved during the Gemini program in 1966 [7] [8]. Astronauts Michael Collins and John Young successfully orbited the earth for 39 hours while docked to their Agena rocket [8]. Figure 1.2 illustrates a Russian Soyuz spacecraft preparing to dock with a Mini-Research Module (MRM) [9].

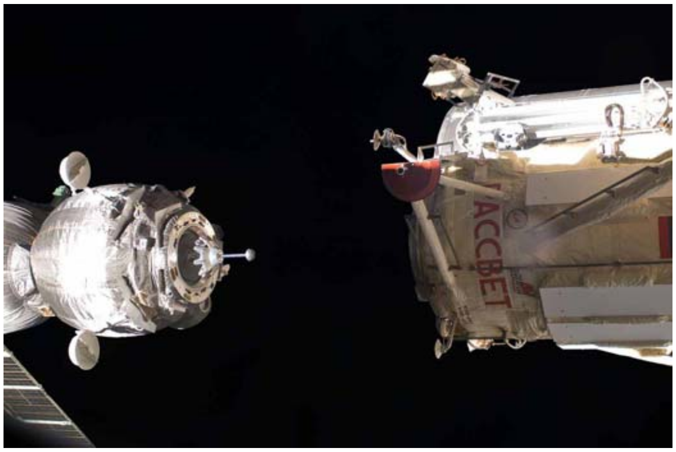


Figure 1.2 | A Soyuz TMA-03M docking with MRM-1 Rassvet

Docking has been used extensively throughout the construction and maintenance of the ISS. The ISS is composed of a series of modules that were fabricated on earth and assembled on-orbit using a combination of berthing and docking procedures involving the space shuttle, robotic manipulators such as the iconic Canadian Space Agencies (CSA) Canadarm, and human space walks. Today astronauts and cosmonauts aboard the ISS rely on successful spacecraft docking to receive supplies and scientific payloads from visiting spacecraft. It is critical that contact forces are monitored and controlled during the final docking procedure to prevent catastrophic damage to both spacecraft.

1.1.2 Space Berthing

Cook et al define space berthing as the capture of an incoming spacecraft using a robotic arm in close proximity of a stationary interface mechanism. The capture is typically followed by a variety of grapple and alignment stages [7]. The CSA most famous and technological achievement, Canadarm is well known for its ability to grapple and transport objects in space [10]. The Canadarm remote manipulator system made its first appearance on the Space Shuttle Columbia (STS-2) on November 13, 1981. Canadarm has since had its "hands" in several notable missions including the first satellite capture and repair involving the Solar Maximum Mission satellite in April of 1984, the capture of the 10.5-tonne Long Duration Exposure Facility (LDEF) in January of 1990, the Hubble telescope service missions, and the construction and maintenance of the ISS [11]. Figure 1.3 illustrates the berthing of the SpaceX Dragon via a robotic manipulator.

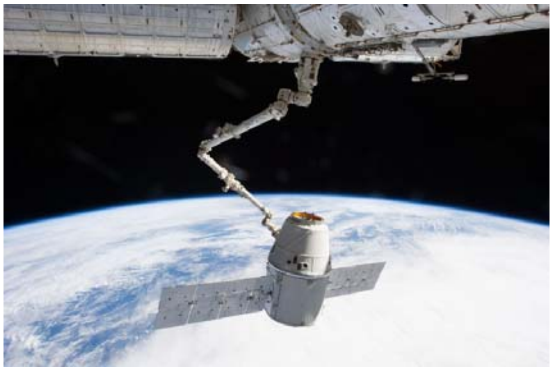


Figure 1.3 | Berthing of the SpaceX Dragon

1.1.3 Hubble Space Telescope

The Hubble Space Telescope (HST) was placed into orbit to capture high definition images of the distant universe. Placing a telescope in orbit eliminates image distortion caused by shifting air pockets in the atmosphere. The HST circles the earth at a height of 569km above the earth's surface and was the first telescope designed to be serviced in space [12]. Shortly after its deployment in 1990, scientist realized one of the telescopes primary mirrors had undergone spherical aberration resulting in fuzzy images requiring the first of five on orbit Hubble service missions [12].

The first Hubble service mission (SM1), was conducted in December of 1993. The Space Shuttle Endeavor's (STS-61) remote manipulator system (RMS) shown in the upper right corner of Figure 1.4 was used to grapple the telescope and secure it safely in the shuttle's payload bay. The RMS was developed by Spar Aerospace and consists of a manipulator arm, an RMS display and control panel, and a manipulator controller interface unit [13]. Once docking was achieved the RMS was used to transport and anchor astronauts throughout the remaining servicing tasks. Notable improvements to the HST during SM1 include the correction of the primary lens using a Corrective Optics Space Telescope Axial Replacement (COSTAR), the addition of the Wide Field Planetary Camera 2 (WFPC2), replacing the solar arrays and corresponding electronics, and two new gyroscope control units.

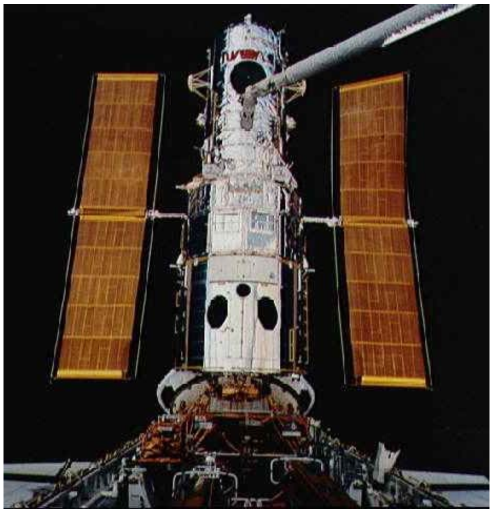


Figure 1.4 | Capture of Hubble telescope during SM1

Four successful servicing missions followed SM1. The second servicing mission (SM2) was executed in February 1997 and lasted ten days. During SM2 (STS-82) a Near Infrared Camera and Multi-object Spectrometer (NICMOS) was installed to properly view light from the most distant galaxies [14]. During this mission, the telescope also received a refurbished Fine Guidance Sensor, a Solid State Recorder, and a refurbished, spare Reaction Wheel Assembly.

SM3 (STS-103) was slated to be one of the busiest servicing missions to date. The telescope was scheduled to receive six new gyroscopes, one of three Fine Guidance Sensors, and a transmitter. Unfortunately, SM3 had to be split into two missions (SM3A & SM3B) after the third of six Hubble's gyroscopes failed [14]. SM3A took place at the end of December 1999 with a crew of seven astronauts. Luckily, the first part of the third mission was not a complete waste. Astronauts Steven Smith, and payload commander John Grunsfeld were able to perform servicing tasks with the help of the RMS as shown in Figure 1.5.

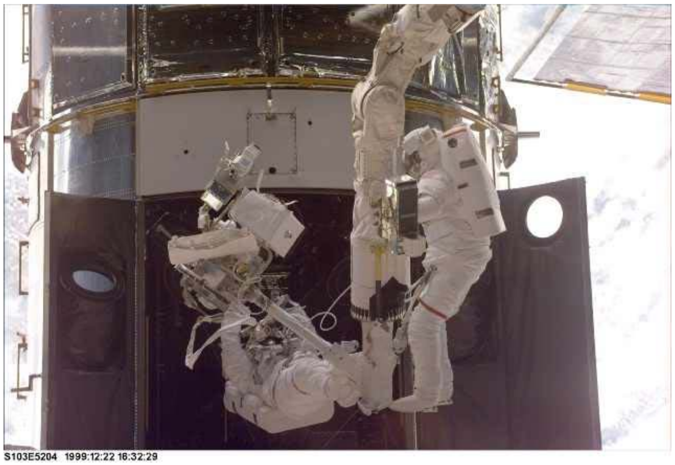


Figure 1.5 | Astronauts service Hubble during SM3A

The second part of SM3, SM3B was completed in March 2002. The primary task of SM3B was to install a new instrument called the Advanced Camera for Surveys (ACS). The ACS provided the HST with a sharper image, a wider field of view, greater capture speeds, and enhanced sensitivity. During SM3B the telescope was fitted with new solar panels yielding a 30 percent increase in power. The telescope was also fitted with a new cooling systems for the NICMOS and new reaction wheel assembly.

The final servicing mission SM4 (STS-125) was completed in May 2009 and is considered to be Hubble's most challenging service mission due to the mission's large to-do list and extensive space walks [14]. To kick off the mission two new instruments were scheduled to be installed: the Wide Field Camera 3 (WFC3) and the Cosmic Origins Spectrograph. A last minute malfunction on the telescope required engineers to develop and plan an additional task of replacing the data handling and science instrument command units two weeks before the scheduled launch date [14]. During the fourth and final Hubble service mission, Astronauts were also required to perform on-site repair of the ACS and the Space Telescope Imaging Spectrograph (STIS). Each HST servicing mission faced unique challenges that could not have been overcome without robotic assistance.

1.1.4 Satellites and Spacecraft On-Orbit

According to the Union of Concerned Scientists there are currently 1,235 civil, commercial, and military satellites in operation, on-orbit. There are three primary earth orbit regions: low Earth orbit (LEO), medium Earth orbit (MEO), and geostationary orbit (GEO). The orbit of a satellite depends on the satellites end application, i.e. communication, intelligence, positioning, etc. Satellites in LEO circle the planet at heights ranging from 100 to 1900km above the Earth's surface [15] [16].

MEO is the next available orbit (in terms of distance). MEO contains a small percentage of the total satellites in orbit [15]. The MEO is home to precision timing and navigation constellations including U.S. Global Positioning Systems (GPS) satellites [15] [16]. The GEO is the furthest orbit from Earth. GEO contains 458 orbiting satellites and is located approximately 22,300 miles [35,888km] from Earth's surface [17]. GEO provides satellites with an orbital period equal to one Earth rotational period. The syncing of orbit and rotational periods provides satellites with a *fixed* position relative to Earth [15]. Early-warning missile detection, nuclear testing detection, intelligence, commercial communication and direct broadcasting are among the types of satellites found in GEO [15].

Short orbit periods and minimal signal propagation delay makes LEO the perfect home for communication satellites. The LEO accounts for 655 of the orbiting satellites including the ISS which circles the earth 16 times a day at 28,000 km/h at an altitude of roughly 370km [18]. Figure 1.6 illustrates the state of the ISS shortly after space shuttle Discovery (STS-119) delivered and installed the final U.S. truss segment and final pair of solar panels. Since November 2, 2000 the ISS has served as a test bed for unique berthing and docking mechanisms including some of the most advanced manned and unmanned robotics ever created [7] [19].

Figure 1.6 | A view of the ISS as STS-119 departs

The construction and operation of the ISS relies on an international partnership including the United States, Russia, Europe, Japan, and Canada [20]. Thanks to the tests and research performed on the ISS a significate advancement in satellite servicing has been made over the last few years including NASA's SSCO Robotic Refueling Mission (RRM). RRM is a multiphase demonstration of technology, tools, and techniques necessary for servicing satellites on-orbit. The RRM hardware was delivered by the space shuttle (STS-135) and attached to the ISS Express Logistics Carrier 4 (ELC-4) circled in Figure 1.7.

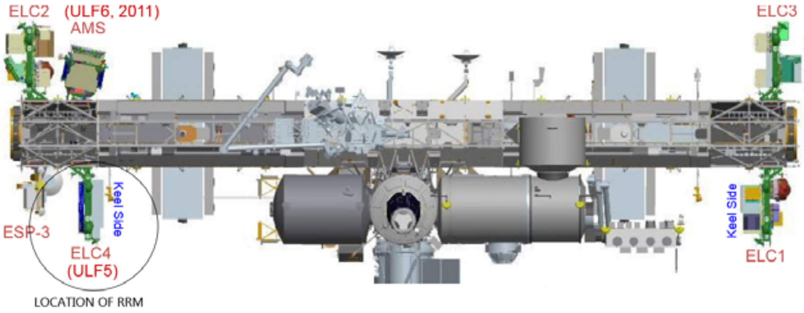


Figure 1.7 | Location of ELC-4 and RRM

The first phase of the mission demonstrated robotic operations such as launch lock and gas fitting removal, sub-miniature cap removal, screw removal and refueling. On January 25, 2013 the SSCO successfully demonstrated fluid transfer in space. The transfer of liquid fuel made its mark in history; ending the first phase of NASA's multiphase RRM [21]. Figure 1.8 depicts the SSCO Enhanced Vapor Recovery (EVR) nozzle about to mate with the fuel valve located on the RRM test bed.

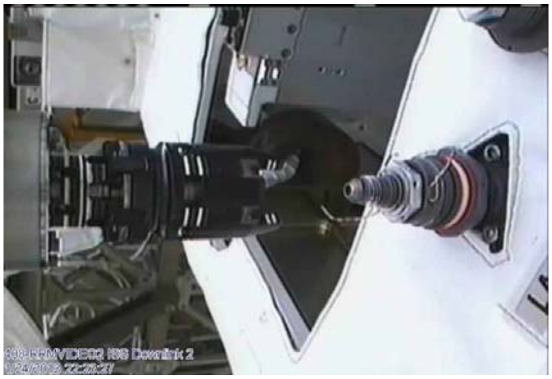
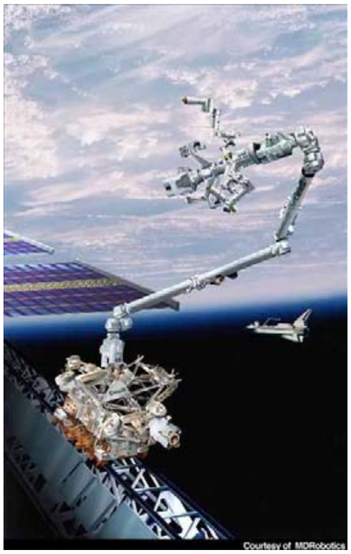


Figure 1.8 | EVR nozzle about to mate with RRM fuel valve

The achievements and success of the ISS and SSCO would not have been possible without the help of robotic assets such as CSA's Mobile Serving System (MSS). The MSS consists of three elements: a space station remote manipulator system (SSRMS) or Canadarm2, a mobile base system (MBS), and Dextre, the special purpose dexterous manipulator also known as the "Space Handyman" [22]. The MSS, illustrated in Figure 1.9, has been vital to the success of the recent robotic refueling missions conducted by NASA and the SSCO. On July 17, 2013 the



Courtesy of MDRobotics

Figure 1.9 | Canadian Space Agency Mobile Servicing System

SSCO Robotic Refueling Mission (RRM) was awarded a "Top Exploration Technology Application from the International Space Station" for its excellent research achievements [23].

1.2 Research Goals

Two of the major challenges involved with robotic docking and berthing include the measurement of dynamic contact forces during vehicle capture and monitoring static forces while spacecraft maintenance is being performed. The slightest nudge on-orbit can result in the catastrophic destruction of a single satellite potentially leading to a destructive chain reaction involving all satellites on-orbit [15]. The 2013 film "Gravity" starring Sandra Bullock and George Clooney illustrates the devastation that could occur after a Russian spy satellite is purposely destroyed on-orbit [24]. Although the film illustrates the results of intentional destruction of on-orbit spacecraft it is a clear representation of the dangers involved with space debris. Space debris could also be caused by minute, careless spacecraft collisions during servicing. It is important to understand and monitor the contact forces between two on-orbit spacecraft so teleoperators and/or computer algorithms can make the proper telemetry adjustments and minimize the chance of a catastrophic failure. The proposed RSS will have the ability to provide said systems with important force measurements and assist both manned and unmanned dexterous mating of spacecraft.

This thesis provides a method for monitoring contact forces by calculating a centroid of force using measurements taken from commercial-off-the-shelf (COTS) subminiature compression load cells. It will be shown that multiple RSS rails can be used to achieve both large and small scale force reconstruction. The harsh environment of space will require the system to be resistant to a full spectrum of electromagnetic interference (EMI) as well as extreme temperature fluctuations [25]. The tool's proof of concept will be verified using the results of a series of ground-based test scenarios involving high precision test equipment and a fuzzy logic simulator.

1.3 Thesis Organization

A literature review of basic engineering mechanics, fundamental strain gage applications, and Wheatstone bridge circuits are given in Chapter 2. The design of RSS mechanical, electrical, and test software components are broken down into the appropriate disciplines in Chapter 3, including a detailed inspection of the RSS concept of operation. Chapter 4 reports the experimental results measured during ground based testing. Important design considerations and system modifications for flight ready space applications are considered in Chapter 5. Finally, a conclusion, possible future work, and lessons learned are provided in Chapter 6.

CHAPTER 2 | LITERATURE REVIEW

The RSS concept of operation as well as basic engineering mechanics is presented in this chapter. A brief review of strain gage fundamentals and the Wheatstone bridge are provided as a basis for the fundamental operation of the system. The chapter will conclude with the objective of this thesis.

2.1 Concept of Operation

The interaction between two objects can be described quantitatively using their mutual contact force [26]. Force is defined as a vector quantity that contains both magnitude and direction [26]. Typically the output of an electro-mechanical transducer such as a strain gage will provide a magnitude of force without direction [27]. Consider the force *(F)* shown in Figure 2.1.

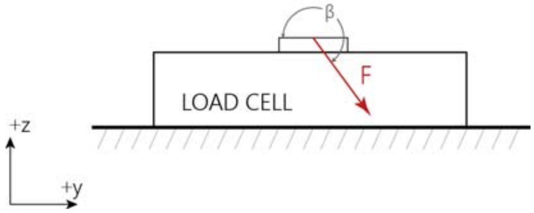


Figure 2.1 | Magnitude of force on individual load cell

Infinite combinations exist where the force magnitude may be identical regardless of the angle of contact. Therefore the magnitude of force measured at the load cell is independent of the angular offset β. A constellation of load cells is required if the exact location of force is to be determined. Consider the ideal single dimension bi-cell constellation shown in Figure 2.2. A *fixed* rigid body distributes the appropriate magnitude of force to each load cell. The mass of the rigid body and frictional forces are omitted in this example. Additionally, the angular offset of

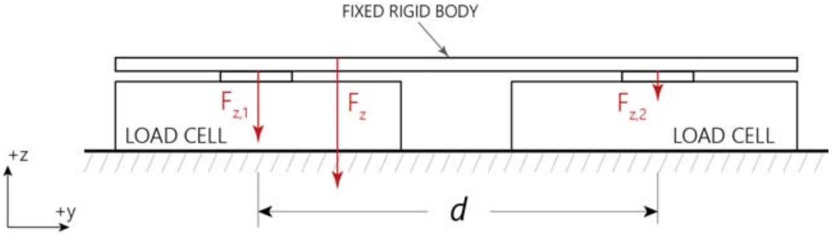


Figure 2.2 | Bi-cell constellation used to determine location of force in one dimension

the applied force is assumed to be zero and therefor the contact force is orthogonal to the surface of the plate. The described system is statically determinate and the problem can be solved using the equilibrium conditions in Eq. (2.1) [28].

$$\begin{aligned} \sum F_i &= 0 \\ \sum M_i &= 0 \end{aligned} \tag{2.1}$$

Further investigation of Eq. (2.1) shows that the location of the contact force can be obtained by calculating the centroid of force acting on the metal plate [28]. The *y* coordinate of the contact force can be calculated using Eq. (2.2). If the location of $F_{z,1}$ is mapped to the origin Eq. (2.2) simplifies to Eq. (2.3). For the ideal case the magnitude of force at *y* equates to the sum of forces present at each load cell.

$$y = \frac{|\boldsymbol{F}_{z,1}|y_1 + |\boldsymbol{F}_{z,2}|(y_1 + d)}{|\boldsymbol{F}_{z,1}| + |\boldsymbol{F}_{z,2}|} \tag{2.2}$$

$$y = \frac{|\boldsymbol{F}_{z,2}|(y_1 + d)}{|\boldsymbol{F}_{z,1}| + |\boldsymbol{F}_{z,2}|} \tag{2.3}$$

The centroid of force equations can be extended into two dimensions by arranging four load cells in a rectangular constellation as shown in Figure 2.3. The spatial rigid body is omitted to show the location of the load cells.

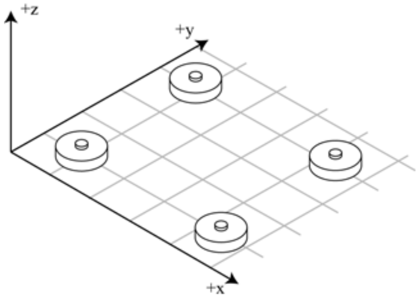


Figure 2.3 | 4-cell constellation used to determine location of force in two dimensions

The process of determining the coordinates of a centroid force in two dimension is identical to the method of determining the location in a single dimension.

$$x = \frac{|\boldsymbol{F}_{z,1}|x_1 + |\boldsymbol{F}_{z,2}|(x_1 + d)}{|\boldsymbol{F}_{z,1}| + |\boldsymbol{F}_{z,2}|} \tag{2.4}$$

As before, Eq. (2.4) can be simplified to (2.5) by mapping $F_{z,1}$ to the origin.

$$x = \frac{|\boldsymbol{F}_{z,2}|(x_1 + d)}{|\boldsymbol{F}_{z,1}| + |\boldsymbol{F}_{z,2}|} \tag{2.5}$$

Simplifying Eq. (2.3) and Eq. (2.5) yields the equations required to calculate the centroid location on a spatial structure, Eq. (2.6) and Eq. (2.7) respectively.

$$x = \sum_{i=1}^{2} \frac{|\boldsymbol{F}_{z,i}|(x_i)}{|\boldsymbol{F}_{z,i}|} \tag{2.6}$$

$$y = \sum_{j=1}^{2} \frac{|\boldsymbol{F}_{z,j}|(y_j)}{|\boldsymbol{F}_{z,j}|} \tag{2.7}$$

2.2 Fundamentals of the Strain Gage

2.2.1 Strain

In engineering, strain refers to the change in any linear deformation of a body due to the application of external forces [29]. Engineering strain is defined mathematically in Eq. (2.8), where L_1 is the final length and L_0 is the initial length of a given body. Since the early to mid-20th century, several techniques and instruments have been developed to measure and characterize material strain. The strain gage has been a fundamental tool for the experimental analysis of stress since the late 1930's [27]. Some of the most notable methods of measuring strain include: mechanical strain measurement, optical strain measurement, photoelastic strain gages, and electric strain gages.

$$\varepsilon = \frac{(L_1 - L_0)}{L_0} = \frac{\Delta L}{L} \tag{2.8}$$

The focus of this research has been on the bonded electrical resistance metal foil strain gage. The bonded electrical resistance metal foil strain gage consists of a grid made from thin (2.5μm) strain-sensitive foil, a thin (12.5-50μm) insulating backing, and an adhesive. A typical metal foil strain gage pattern is illustrated in Figure 2.4. The metal foil strain gage offers several advantages over similar wire strain gages. For example, sensitivity to transverse strains can be greatly reduced by increasing the size of the end loops [29]. Other advantages include increased surface area compared to cross-sectional area and improved heat dissipation characteristics.

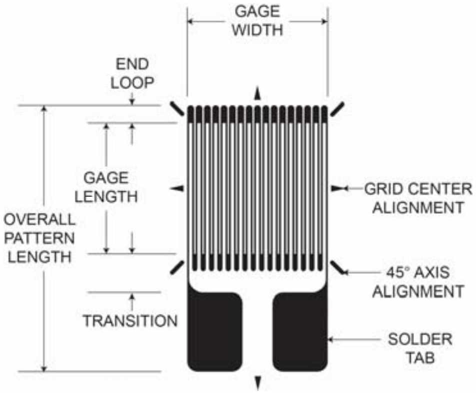


Figure 2.4 | Typical metal foil strain gage and nomenclature

The bonded electrical resistance strain gage is governed by the resistivity ρ of the strain gage material and the gage length. Resistivity is defined as the ratio of the magnitudes of electric field and current density where the current density J is inversely proportional to the cross-sectional area and proportional to the current I through the conductor. Current density can be defined mathematically using Eq. (2.9). Resistivity can be calculated using Eq. (2.10)

$$J = \frac{I}{A} \tag{2.9}$$

$$\rho = \frac{E}{J} \tag{2.10}$$

Inspection of Eq. (2.10) shows that an increase in cross-sectional area causes a decrease in current density, resulting in an increase in material resistivity. The resistance R of a conductor is proportional to the resistivity of the material as shown in Eq. (2.11).

$$R = \frac{\rho L}{A} \tag{2.11}$$

Rearranging Eq. (2.9), Eq. (2.10), and Eq. (2.11) and substituting variables leads to Eq. (2.12) the strain sensitivity or gage factor of the strain gage:

$$GF = \frac{\Delta R/_{R}}{\Delta L/_{L}} \tag{2.12}$$

where ΔR represents the resistance change in ohms and ΔL represents the change in conductor length. Strain gages with large gage factors provide excellent strain sensitivity resulting in greater electrical output for the data recording instrumentation [29].

2.2.2 Strain Gage Material

Typical COTS strain gage resistance values are: 120, 350, 700, and 1000Ω. Larger resistance values can decrease error caused by the resistance in connecting wires and other unwelcome phenomenon [30]. Table 2-1 lists some of the popular strain gage materials and their gage factors. The values in Table 2-1 are not exact and material composition, manufacturing techniques, and ambient temperature will ultimately decide the final gage factor.

Table 2-1 | Gage Factors for possible strain gage conductors

Metal or Alloy	**Trade Name**	**Gage Factor (GF)**
Aluminum	-	+0.85
Constantan	Advance, Cupron, Copel, etc.	+2.1
Copper	-	+2.6
Karma	Evanohm, Chromel R, etc.	+2.1
Nickel	-	-12.0
Silver	-	+3.0
Titanium (Commercially Pure)	-	-1.1

Constantan and Karma alloys are most commonly used in the fabrication of metal foil strain gages [31]. Constantan is the most frequently used alloy. It is composed of 55% copper and 45% nickel. Constantan provides moderate resistivity and has a low temperature coefficient of resistance yielding less sensitivity to temperature [29] [32]. Karma is a nickel chromium alloy that offers high electrical resistivity and a low temperature coefficient. Karma offers several benefits over Constantan such as improved fatigue life, excellent stability over a wide temperature range (-270 to 270ºC), a much flatter thermal output curve, and greater resistivity [33]. Table 2-2 compares material properties for Constantan and Copper.

Table 2-2 | Properties of common strain gage foils

Metal or Alloy	**Electrical resistivity** (μOhmcm)	**Temperature coefficient** (K^{-1})	**Modulus of elasticity** (GPa)	**Density** (g cm^{-3})	**Coefficient of thermal expansion** ($x10^{-6}K^{-1}$)
Constantan	52.0	+/-0.00002	162	8.9	14.9
Copper	1.69	.0043	129.8	8.96	17.0

2.2.3 Strain Gage Cement

The insulated backing material and cement used to adhere the metal foil to its carrier assumes a critical role in the success or failure of the strain gage. The adhesive layer must employ characteristics that permit maximum strain translation from the carrier surface to the metal foil. Perry and Lissner list three considerations that dictate the choice of adhesive cement [29]:

1. Material used in gage construction
2. Environmental conditions within which the gage must perform
3. Time available for making the gage installation

The most commonly used cements for strain gage applications below 350ºF are epoxy, bakelite, acrylic and nitrocellulose [29]. Polyimide and glass-fiber-reinforced epoxy phenolic are two common backing materials used in strain gage bonding [30]. In summary, the type of foil, backing, and adhesive used is specific to the strain gage application.

2.2.4 Strain Gage Temperature

Ambient temperature plays a major role in the operation of the strain gage. Temperature will be examined closely in the following sections. Temperature compensation circuits are often used in conjunction with strain gages to eliminate unwanted electrical noise at the output. Hannah and Reed provide three basic mechanisms directly related to temperature change: 1) the resistance of a conductive material changes, 2) the physical dimensions of a material changes, and 3) the thermo-electric effects of a circuit are modified [27]. These concerns can be suppressed by implementing temperature compensation techniques.

The majority of temperature compensation techniques can be divided into two basic categories: self-temperature compensation and electrical temperature compensation [27]. End users have little control over the effects of self-temperature compensated strain gages. Self-compensation relies solely on the material of the strain gage and fabrication techniques used to create the gage. Electrical temperature compensation can be implemented after the strain gage is bonded to a surface. Two types of post fabrication temperature compensation techniques are dummy-gage compensation and modulus compensation [27]. The details of these post-fabrication compensation techniques will be covered in the following section. Applications specific to rail sensor system are covered in Section 3.2 and Chapter 5.

2.3 The Wheatstone Bridge

There are generally two types of circuits used to condition voltage differences caused by resistance variations in the strain gage: potentiometric (half-bridge) or the more popular Wheatstone (full-bridge) shown in Figure 2.5 [27]. A Wheatstone bridge is a divided bridge circuit used for the measurement of static or dynamic electrical resistance [34]. Developed by Samuel Hunter Christie in 1833 and later improved and promoted by Sir Charles Wheatstone in 1843 [30] the Wheatstone bridge can operate in two distinct modes of operation: a null-balance system or a direct-reading "deflection" instrument [29]. The RSS employs the latter. By selecting suitable resistance values for R_1, R_2, R_3, and R_4 it is possible to balance the bridge so that no current flows in the galvanometer branch. An unbalanced bridge exists when current flows through the galvanometer branch resulting in a change in voltage V_G, shown in Figure 2.6.

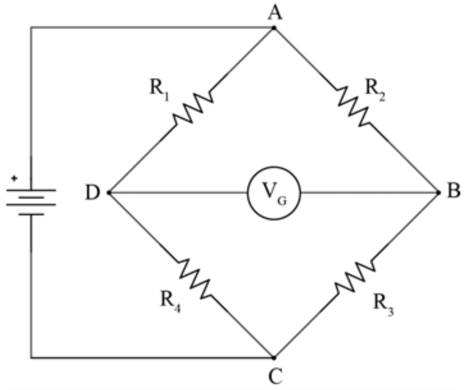


Figure 2.5 | Wheatstone bridge circuit

The temperature compensation techniques introduced in the previous section can be used in conjunction with a Wheatstone bridge to combat unwanted electrical noise at the output of the strain gage. *Dummy-gage compensation* involves the connection of a second gage in an adjacent bridge arm to the measuring gage [27]. This technique is commonly used when single measuring gage is used [27]. *Modulus compensation* is achieved by placing resistors in series with the excitation leads. Specific resistance values are selected to offset the decrease in modulus brought on by changes in temperature. The manufacturer of the load cells employed in the RSS prototype incorporated *modulus temperature compensation.* Applications specific to RSS are covered in Section 3.2

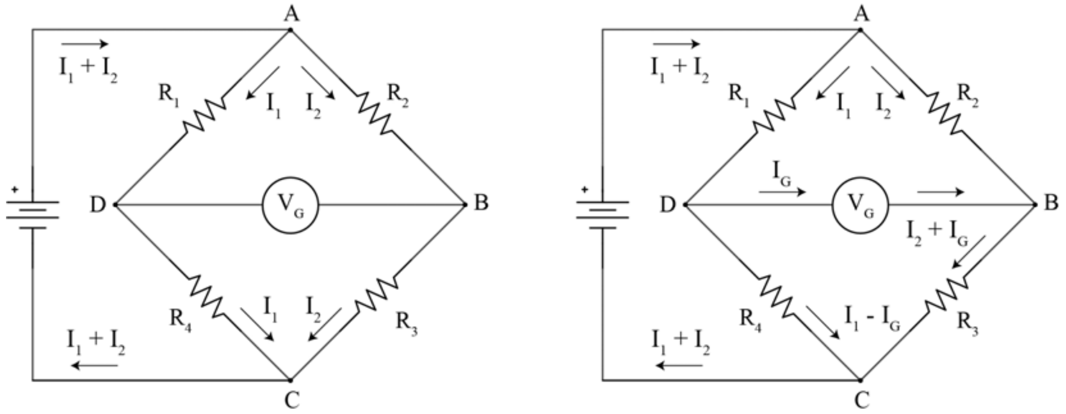


Figure 2.6 | Wheatstone bridge balanced (left), unbalanced (right)

2.4 Thesis Objective

The underlying objective of this thesis is to reconstruct contact forces using electrical responses generated by strain gages fixed to the inside of load cells. Commercially available load cells and electrical hardware will be used to construct an inexpensive preliminary prototype for ground based testing. Computer simulations will be performed to validate the RSS proof of concept. The use of computer numerical controlled (CNC) machinery will be employed to fabricate structural components and ensure mechanical precision. Software will be created to visualize and log voltage changes at each load cell and high precision universal test equipment will be used to test the physical prototype. The proof of concept will be reinforced by comparing test data with simulated data and a fuzzy logic simulator will be used to demonstrate the tools ability to provide a service vehicle with telemetry correction. RSS space considerations will also be discussed for implementation into future work.

CHAPTER 3 | RSS PROTOTYPE DESIGN

3.1 RSS Design Overview

RSS feedback becomes necessary once contact between a service vehicle and client vehicle is achieved. Forces between on-orbit service and client spacecraft must be carefully monitored during docking and berthing procedures. Insufficient berthing forces or careless collisions can lead to a catastrophic cascade of destruction [15]. Measurements taken from the RSS can potentially provide teleoperators and autonomous algorithms information equivalent to an astronaut's sense of touch.

The proposed RSS can be divided into two coherent systems: a mechanical system and an electrical system. The mechanical system is composed of two electro-mechanical transducers sandwiched between two aluminum support structures connected by a single screw. Commercial-off-the-shelf (COTS) Omega LCKD subminiature compression load cells serve as transducers [35]. The aforementioned mechanical system represents a single RSS rail. Two rails are required per RSS. The electrical system consists of a Phidgets 1046 USB interface board and data logger. The Phidgets USB interface board serves two purposes: provide the load cells with a +5Vdc excitation voltage and serve the data logger with raw voltage measurements. A computer and custom software is used to log and display data sent by the interface board via USB. The most recent RSS rail prototype RSSv6 shown in Figure 3.1 was tailored specifically for a geostationary operational environmental satellite (GOES) interface ring and a Schunk PG70 2-Finger parallel gripper [36]. A 3D printed bracket surrounds the gripper and provides

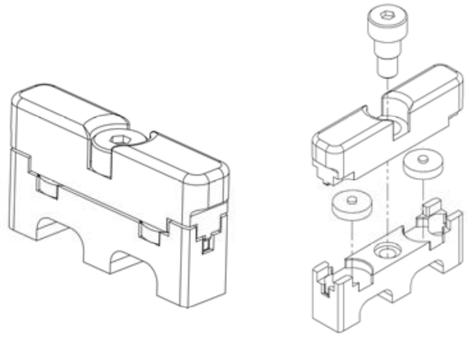


Figure 3.1 | Assembled RSSv6 rail (left)
exploded view (right)

mounting locations for two RSS rails resulting in the 4-cell constellation discussed in the previous section. A complete RSS assembly prototype is revealed in Figure 3.2.

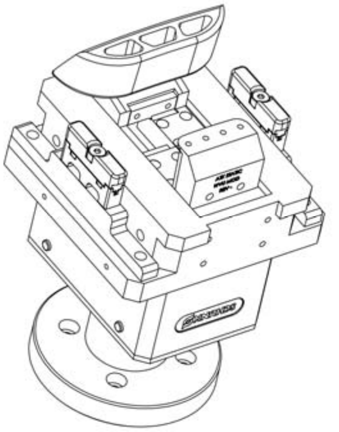


Figure 3.2 | RSS rails attached to the Schunk gripper

Throughout the course of this research several variations of the RSS have been fabricated and tested. Early RSS prototypes utilized thin beam load cells, bulky force plates, and complex mounting configurations. Previous RSS variations have provided sufficient results supporting the concept of force reconstruction but they have lacked the form factor necessary to mount to existing gripper units. Figure 3.3 illustrates a slice of the GOES interface ring referenced during the design of the RSS. The angular offset between the interface ring and capture mechanism during berthing can be as large as fifteen degrees [37]. The hatched region illustrated in Figure 3.3 represents the relatively small contact surface available for contact force between the interface ring and RSS.

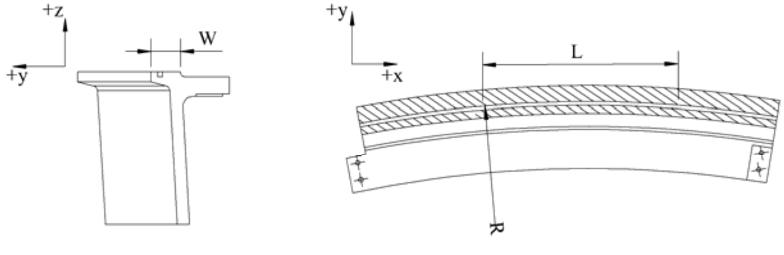


Figure 3.3 | Typical satellite interface ring profile (left) ring slice as seen by RSS (right)

The geometry of the interface ring, gripper unit, and approach conditions permit four general contact scenarios: *two rails - full response, two rails - partial response, single rail - full response,* and *single rail - partial response*. A full response occurs when all four load cells report a change in electric potential. Typically, a full response is the result of a nominal approach angle or grapple. However this not always the case. Additionally, it will be shown that a partial response is without doubt the result of an unsatisfactory approach angle. The approach of the gripper can be characterized by its *local coordinate system, roll, pitch, and yaw*. The coordinate system and rotation terminology used in the following discussions are defined in Figure 3.4.

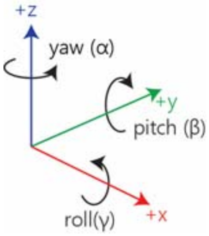


Figure 3.4 | Response coordinate system

3.1.1 Two Rails - Full Response

A full response exists when both rails come in contact with a foreign surface and all four load cells report a change in voltage. Unfortunately, this type of response does not always indicate a perfect approach angle. Further examination of each individual load cell will be necessary to determine the type of contact. Consider the alignment in Figure 3.5. This arrangement excites all four load cells but represents a less than nominal approach due to error in *pitch* and/or *roll*. The central location of the connecting screw eliminates the possibility of

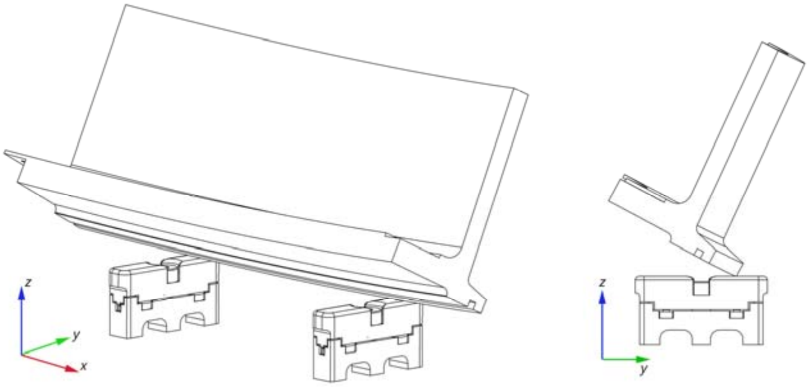


Figure 3.5 | Two rails - unsatisfactory full response caused by error in pitch or roll

determining nominal contact because the contact surface must be large enough to span the diameter of the counterbore.

A second type of full response generated by two rails is presented in Figure 3.6. This approach demonstrates perfect *pitch* and *roll* with error in *yaw*. This error can be caused by a less than nominal approach angle or uneven RSS contact surfaces. The free body diagram illustrated in Figure 3.7 and its centroid of force reinforce this phenomenon when $f_2 = f_4$ and $f_1 = f_3$. The length of the force vectors are equivalent to the magnitude of force at that location. Equal forces on a diagonal can lead to dangerous conclusions if individual sensor measurements are not considered. A perfect approach and grapple is the final type of general full response generated by two rails. The approach shown in Figure 3.8 exists when all load cells report *identical* force measurements.

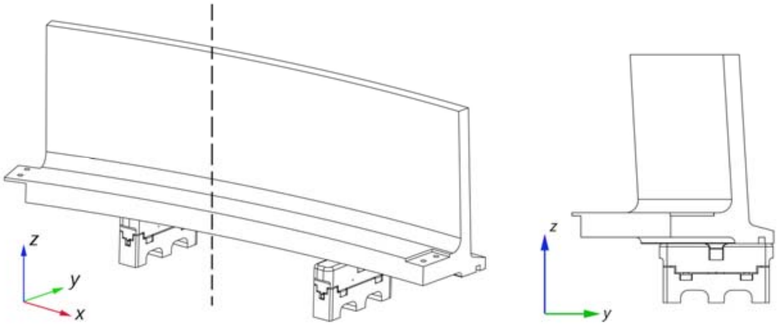


Figure 3.6 | Two rails - unsatisfactory full response caused by error in yaw

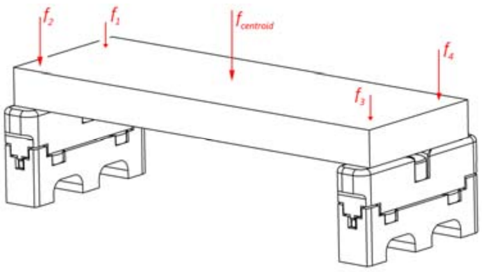


Figure 3.7 | Free body diagram illustrating equal diagonal forces

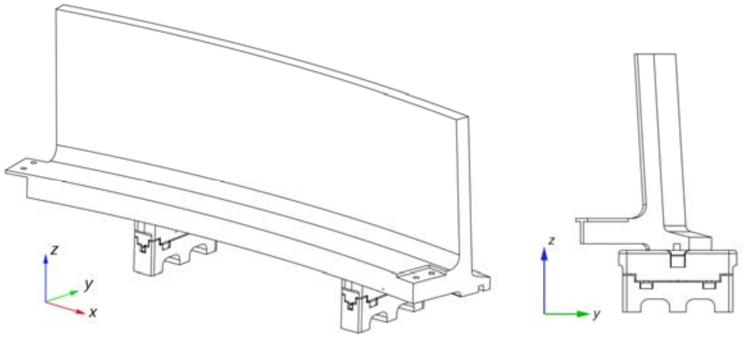


Figure 3.8 | Two rails - nominal full response

3.1.2 Two Rails – Partial Response

A partial response involving two rails indicates an unfavorable moment about one of the load cells. This phenomenon occurs when contact was made outside of the *hit box*. The *hit box* is an imaginary rectangle bound by the location of the four load cells. If a force occurs outside of the hit box it will cause the top rail to lift away from one of the load cells resulting in unwanted coupling as shown in Figure 3.9. The connecting screw and preloading contribute to the error by introducing a third contact location resulting in a statically indeterminate system [28]. A partial response can be detected using data from other on board sensors and individual load cell measurements.

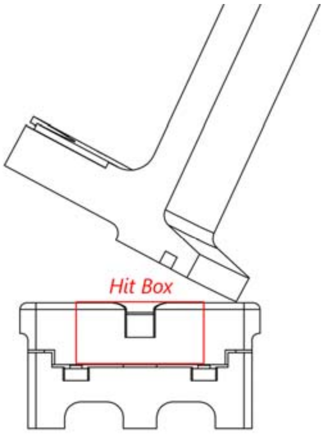


Figure 3.9 | Coupling caused by contact outside of hit box

3.1.3 Single Rail – Full Response

The single rail full response illustrated in Figure 3.10 exists when two load cells from the same rail report a change in voltage while the load cells from the adjacent rail remain unchanged. This type of contact provides a straightforward definitive conclusion that a poor approach has been made.

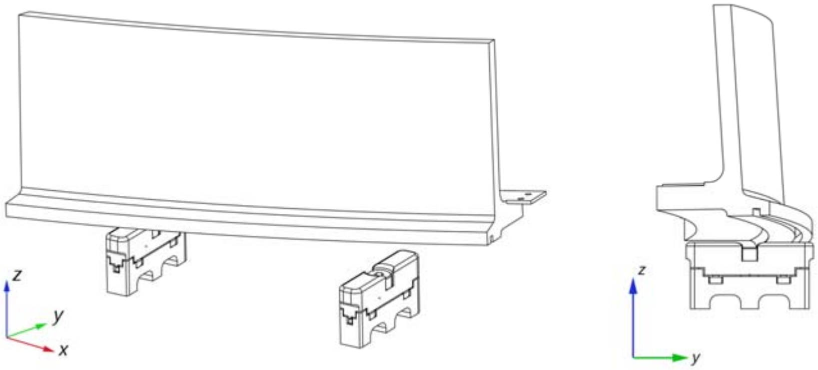


Figure 3.10 | Single rail full response caused by error in pitch and roll

3.1.4 Single Rail – Partial Response

A single rail partial response is a clear indication that major trajectory adjustments must be made. Recall that a partial response occurs when contact is made outside of the *hit box*. Unlike the two rail partial response, a partial response for a single rail is easier to identify, as shown in Figure 3.11.

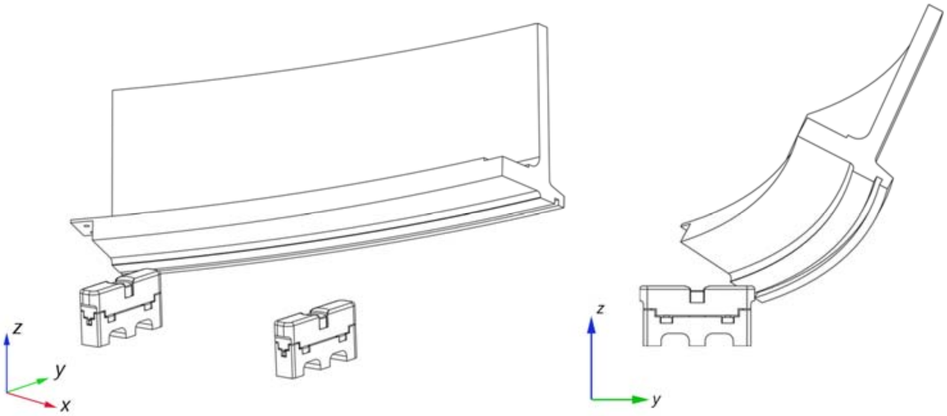


Figure 3.11 | Single rail full response caused by error in pitch, roll, and yaw

3.2 RSS Load Cells

Omega LCKD-50 subminiature compression load cells were used throughout the course of this work. Each LCKD-50 load cell consists of four metal foil strain gages, a cylindrical stainless steel shell, and a temperature compensation circuit board attached to the load cell cable. A load cell calibration sheet including a 5-point calibration was shipped with every Omega load cell. Two calibration points were used to form a line equation for each load cell. As an example consider the load cell calibration sheet in appendix A. The load cell calibration reports a response equivalent to .003mVdc when zero force is applied and a 9.885mVdc response when 50lbs of force is applied to the load cell. The linear nature of the strain gage suggest that the "point-slope" formula in Eq. (3.1) can be used to form a line equation that *fully* characterizes the load cell response. Substituting the calibration values into Eq. (3.1) yields a response equivalent to $y = 5.06x + 0.000$. A plot of the line equations used during testing are provided in Figure 3.12. Note that the slope was multiplied by the excitation voltage to achieve mV V lbs^{-1}.

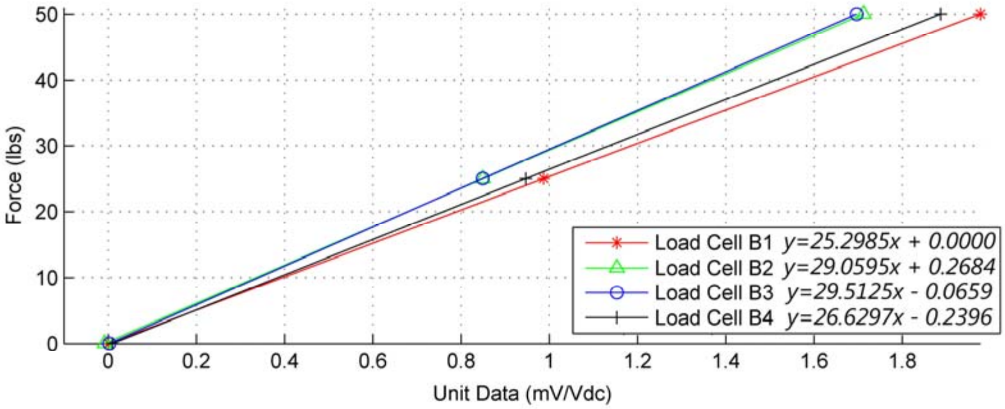


Figure 3.12 | Typical load cell calibration line equations

$$y = \frac{\Delta Force}{\Delta Response}(x - x_1) + Force(x_1) \tag{3.1}$$

The load cell strain gage pattern is illustrated in Figure 3.13. The solder points are labeled accordingly. The LCKD-50 requires a +5Vdc excitation voltage for proper operation. The nominal output for the LCKD-50 load cell is 2mV/V with a maximum load of 50lbs. The exact sensitivity for each load cell was included on the load cell calibration sheet.

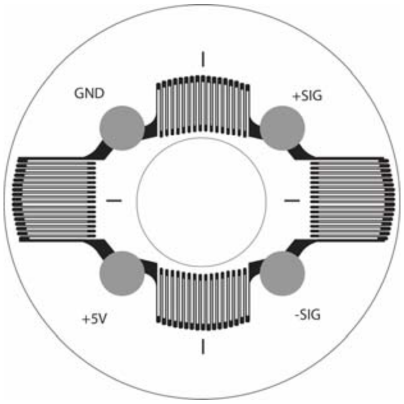


Figure 3.13 | Omega LCKD strain gage pattern

The load cell manufacturer also attached an inline temperature compensation circuit board between the load cell and connecting terminals. Recall from Section 2.3 that this type of temperature compensation is commonly referred to as modulus compensation.

3.3 RSS MATLAB Simulations

MATLAB is a high-level programming language that specializes in scientific computing and mathematics. MATLAB is a registered trademark of the MathWorks Company. Computer simulations provide an ideal system response that can be used to justify measured experiments. The location and magnitude of force were selected at random using the statistical discovery software JMP. JMP is a registered trademark of SAS. A screening design was chosen for the design of experiment (DOE). Specific information regarding the DOE will be covered in Chapter 4.

The conditions of equilibrium and centroid of force equations discussed in Section 2.1 were used to construct a mathematical model that fully defines the simulation of an ideal 4-cell RSS constellation. Matrix inversion was used to simplify the problem [38]. Eq. (3.2) represents the final 4-cell constellation model used to simulate forces at each load cell.

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -\sqrt{(x-x_1)^2+(y-y_1)^2} & \sqrt{(x-x_2)^2+(y-y_2)^2} & -\sqrt{(x-x_3)^2+(y-y_3)^2} & \sqrt{(x-x_4)^2+(y-y_4)^2} \\ \frac{x_1}{f_1+f_2+f_3+f_4} & \frac{x_2}{f_1+f_2+f_3+f_4} & \frac{x_3}{f_1+f_2+f_3+f_4} & \frac{x_4}{f_1+f_2+f_3+f_4} \\ \frac{y_1}{f_1+f_2+f_3+f_4} & \frac{y_2}{f_1+f_2+f_3+f_4} & \frac{y_3}{f_1+f_2+f_3+f_4} & \frac{y_4}{f_1+f_2+f_3+f_4} \end{pmatrix}^{-1} \begin{pmatrix} f \\ 0 \\ x \\ y \end{pmatrix} \tag{3.2}$$

The block diagram in Figure 3.14 illustrates the MATLAB simulation process. The simulation process begins by inputting randomly generated force data which includes location and magnitude. The force data is then passed through Eq. (3.2) and returns a matrix of individual load cell forces. Finally the theoretical data is verified by calculating the *centroid* of the theoretical forces and comparing with the original input data. A simulation is considered a success if the output matches the input.

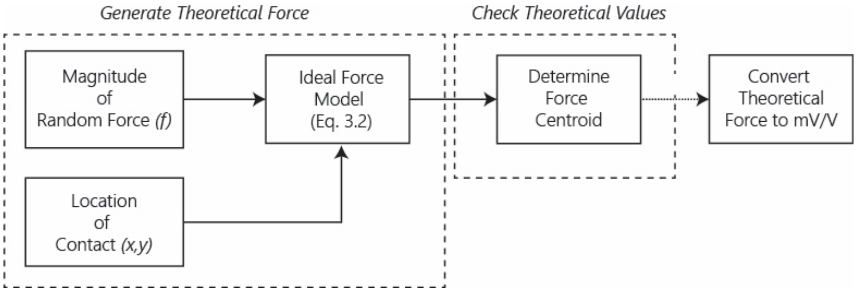

Generate Theoretical Force
Check Theoretical Values
Magnitude of Random Force (f)
Ideal Force Model (Eq. 3.2)
Location of Contact (x,y)
Determine Force Centroid
Convert Theoretical Force to mV/V

Figure 3.14 | Block diagram of MATLAB simulation

3.3.1 RSS Two Rail Centroid Reconstruction Simulation

To simplify the simulation the RSS gripper assembly was reduced to a single plate spanned across two *fixed* RSS rails as shown in Figure 3.15. Using nine contact locations and three forces, thirty six random force location and magnitude combinations were generated by JMP. The

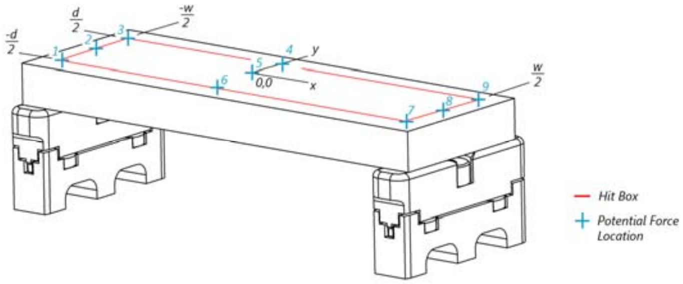


Figure 3.15| RSS two rail centroid simulation sensor configuration

location of the contact force was bound by the *hit box* formed by the location of the load cells. Partial responses were not considered in this simulation. Table 3-1 contains the locations and forces generated by JMP. A comparison of the original random data and theoretical data is illustrated in Figure 3.16 on the following page. The run number *n* is provided to distinguish between overlapping force locations.

Table 3-1 | RSS two rail centroid simulation trials

Run	Location of Force	Magnitude of Force (N)	Run	Location of Force	Magnitude of Force (N)	Run	Location of Force	Magnitude of Force (N)
1	8	40	**13**	8	40	**25**	3	120
2	9	80	**14**	4	40	**26**	1	40
3	5	120	**15**	4	40	**27**	5	120
4	4	40	**16**	9	80	**28**	7	120
5	9	80	**17**	1	40	**29**	2	80
6	6	80	**18**	8	40	**30**	9	80
7	5	120	**19**	2	80	**31**	1	40
8	3	120	**20**	6	80	**32**	5	120
9	7	120	**21**	3	120	**33**	7	120
10	8	80	**22**	4	40	**34**	7	120
11	2	80	**23**	3	120	**35**	2	80
12	1	40	**24**	6	80	**36**	8	40

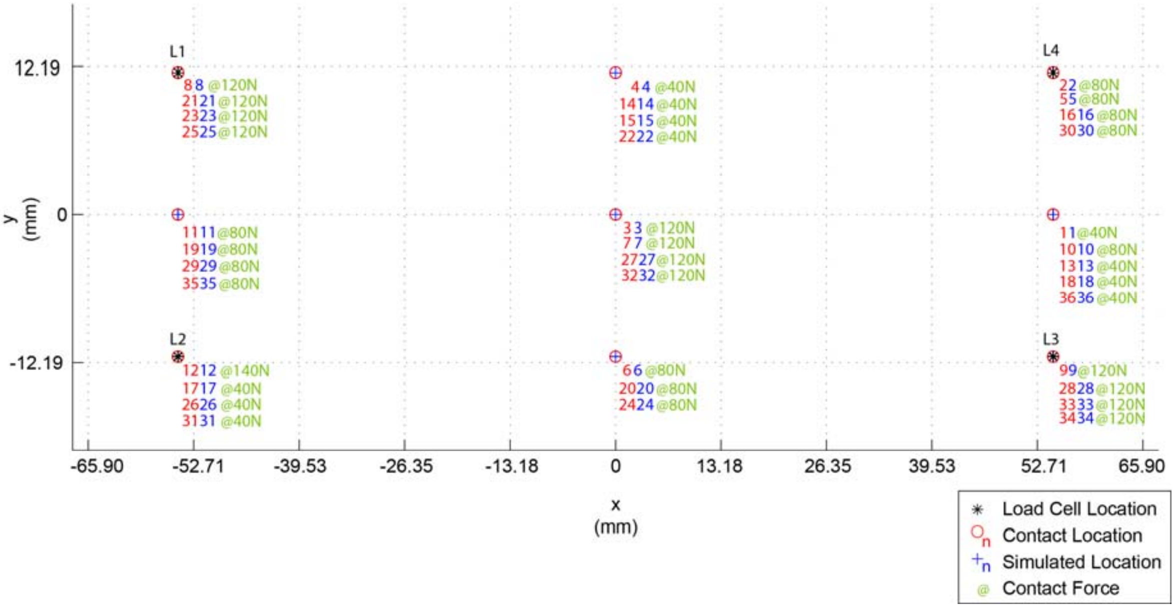


Figure 3.16 | 4-Cell RSS simulation

3.4 RSS Bridge Circuit and Hardware

A COTS Phidgets 1046 interface board was used to interface the sensors and computer via a universal serial bus (USB) cable. The 1046 offers four differential bridge inputs and provides each load cell with +5V of excitation via a computers USB port. The 1046 reports mV/V for each bridge via the device driver. The application programming interface (API) and interface device drivers were provided by the board manufacturer. The Phidgets interface board was selected for this research because of its outstanding specifications at a relatively low cost. Two of the most notable specifications include the ability to control bridge resolution versus gain and bridge data rate.

3.4.1 Phidgets 1046 Gain vs. Resolution

An onboard AD7193 24-Bit Sigma-Delta analog-to-digital converter (ADC) manufactured by Analog Devices provide each bridge with a 24 bit differential voltage resolution. Six predefined gain settings are available in the 1046 API. A gain setting of 128 was selected for most demonstrations and all testing. Selecting a larger gain resulted in a finer output resolution at the cost of a smaller ± 7.8125 mV/V output range [39]. This range covered the Omega LCKD series load cell nominal range (2mV/V) with room to spare. The complete listing of gain versus resolution values reported in Table 3-2**Error! Reference source not found.** were extracted from the Phidgets user manual.

Table 3-2 | Phidgets 1046 Gain vs. Resolution

Gain	Resolution	Range
1	119 nV/V	± 1000 mV/V
8	14.9 nV/V	± 125 mV/V
16	7.45 nV/V	± 62.5 mV/V
32	3.72 nV/V	± 31.25 mV/V
64	1.86 nV/V	± 15.625 mV/V
128	0.93 nV/V	± 7.8125 mV/V

3.4.2 Phidgets 1046 Data Rate

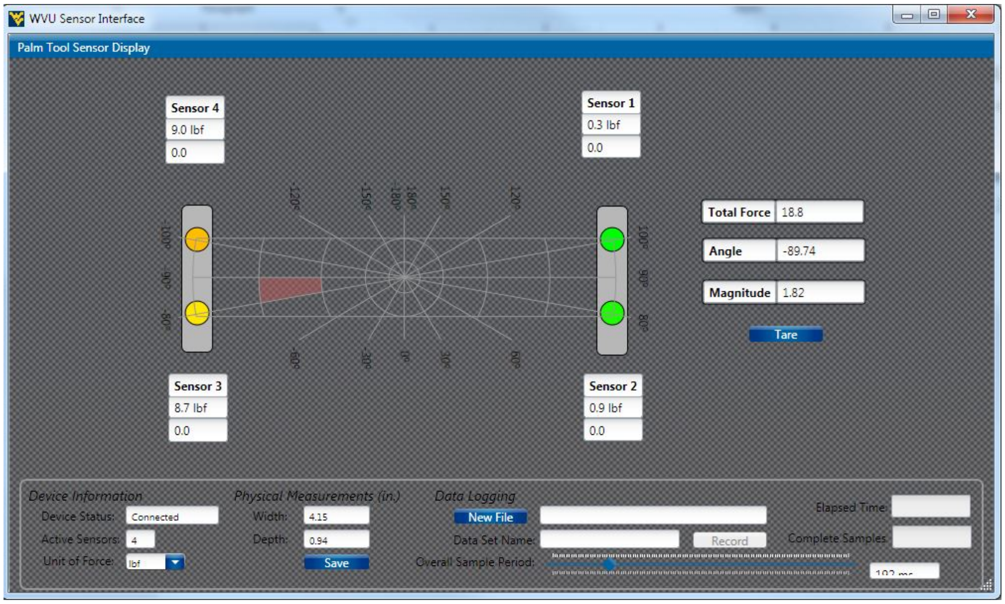
The Phidget 1046 interface board offers data rates between 8ms and 1000ms. An onboard CY7C64215-28PVXC USB controller manufactured by Cypress serves the sensor data to the device driver at 12 Mbps. Data can be retrieved at the software level via data events or by polling. The Phidget API provides functions for polling data on demand.

3.5 RSS Software

Two computer programs were created to interact with the RSS load cells via the Phidgets interface board device driver. A Windows Presentation Foundation (WPF) C# application was created for presentations and demonstrations while a far less graphic based C++ console application was created for RSS testing. The specifics of each application are discussed below.

3.5.1 C# Presentation Software

The Microsoft WPF model provides a vast number of tools for building rich graphical user interfaces (GUI) [40]. Multithreading techniques were employed for seamless data acquisition while maintaining smooth GUI operation. The presentation software screen capture shown in Figure 3.17 utilizes the data event functionality provided by the Phidgets API. Sampling at fast



Sensor 4: 9.0 lbf, 0.0

Sensor 1: 0.3 lbf, 0.0

Sensor 3: 8.7 lbf, 0.0

Sensor 2: 0.9 lbf, 0.0

Total Force: 18.8

Angle: -89.74

Magnitude: 1.82

Device Information

Device Status: Connected

Active Sensors: 4

Unit of Force: lbf

Physical Measurements (in.)

Width: 4.15

Depth: 0.94

Data Logging

Data Set Name:

Overall Sample Period:

Elapsed Time:

Complete Samples:

Figure 3.17 | Screenshot of C# presentation software data rates exposes excessive unwanted electrical noise and must be carefully monitored [39]. If not properly handled, fast data rates can also cause buffers to overflow resulting in incomplete data log files. Other features such as load cell calibration, gain selection, and force unit selection are also configurable at runtime. A tare function was included to remove preloaded load cell readings. The bottom text box in the sensor value box represents the value subtracted at the time of the tare.

Physical RSS dimensions can be inserted into the application via textboxes. A scaled graphical representation of the RSS system *as seen by the client* vehicle is represented at the center of the window. A polar grid overlay outlines a graphic representation of the real-time contact force direction and magnitude. Recall that the coordinates of force are determined using the centroid of force Eq. (2.6) and Eq. (2.7). The concentric circles represent different levels of force magnitude while the radiating lines represent possible directions of force. A detailed look at how the grid overlay was implemented is discussed in Section 4.3.

The red polygon in Figure 3.17 corresponds to the angle and magnitude of force at the moment the screen was captured. If the force is static the polygon will remain in that position. Change in force causes the polygon to move to another location. A nominal approach would be illustrated by a green polygon in the center-most circle of polygons. Figure 3.18 on the following page summarizes the applications sequence of events once the window is loaded.

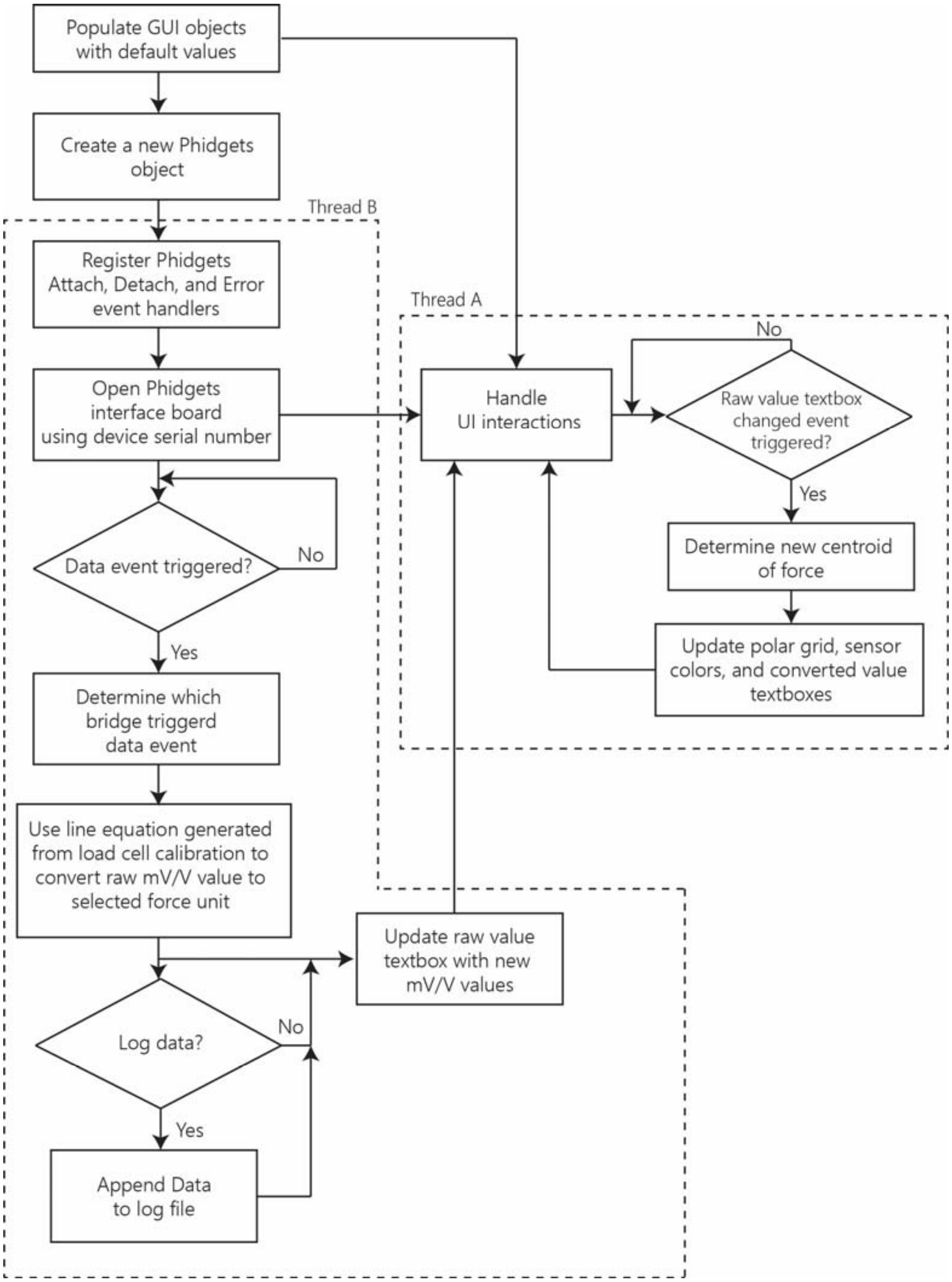

Populate GUI objects with default values
Create a new Phidgets object
Thread B
Register Phidgets Attach, Detach, and Error event handlers
Open Phidgets interface board using device serial number
Data event triggered?
No
Yes
Determine which bridge triggerd data event
Use line equation generated from load cell calibration to convert raw mV/V value to selected force unit
Update raw value textbox with new mV/V values
Log data?
No
Yes
Append Data to log file
Thread A
Handle UI interactions
Raw value textbox changed event triggered?
No
Yes
Determine new centroid of force
Update polar grid, sensor colors, and converted value textboxes

Figure 3.18 | Presentation software sequence of events

3.5.2 C++ Testing Software

The primary goal of the test software was to log raw sensor data generated by the interface board over several force locations. C++ was used to construct a lightweight console application that was not bogged down with flashy graphics and multithreading. The test software offers basic functionality consisting of an initial command line menu, raw sensor values printed to screen, and a set of commands that generate a sequence of log files. Unlike the presentation software, the test software uses polling techniques to acquire data from the interface board. A sampling rate was achieved by suspending thread execution for 100ms immediately after data acquisition. The main menu of the test application provides a user with five basic options:

1. Select Subsystem
2. Number of Configurations
3. Starting Configuration
4. Start Test
5. Exit Test

Recall that a subsystem is composed of two RSS rails, four load cells, and a Phidgets USB interface board. Three subsystems: A, B, and C were created over the course of this research. Recall from Figure 3.15 that a configuration represents a specific force location. Selecting option two allows the user to input the number of force locations being tested. A text file is created and populated at the start of each configuration test. The configuration number serves as the filename for the text file. For example, the data file for the third configuration would be titled "03.txt". In the event an error occurs before testing is complete the user can restart the program and selection option three to designate a new starting configuration.

The endpoint of a configuration is defined when the user hits the enter key. If the program has additional configurations in the queue it will wait for the user to enter 'g' to continue with the next test. A summary of the software's sequence of events is provided in Figure 3.19 on the following page. The complete C++ source code is available in appendix B.

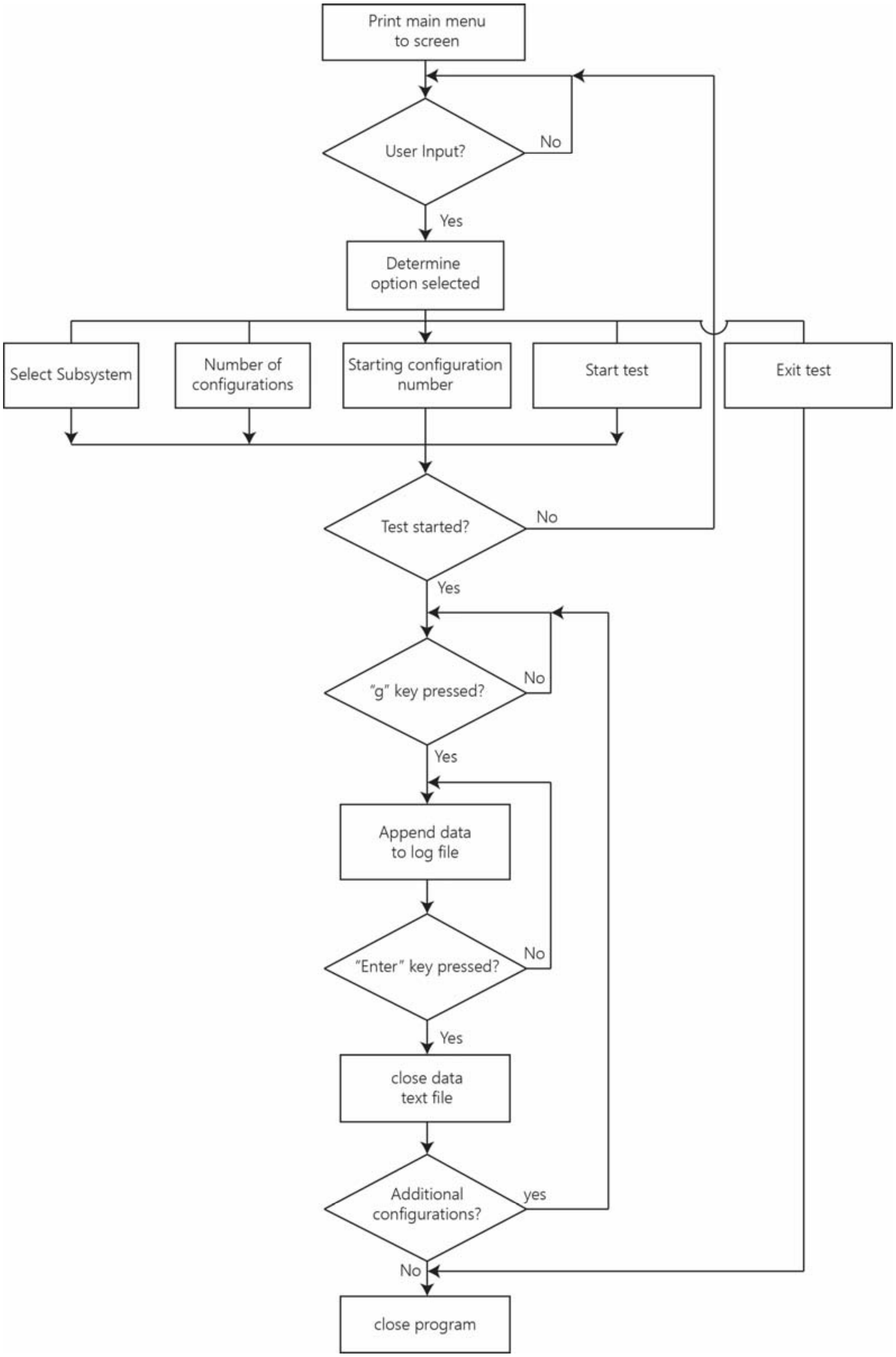


Figure 3.19 | Test software sequence of events

3.6 Summary of Operation

The complete RSS prototype used throughout this research consisted of two aluminum rails, a connecting screw, four LCKD-50 Omega load cells, one Phidget USB interface board, a USB cable, a laptop computer, and custom RSS software. A block diagram summarizing the RSS prototype setup is provided in Figure 3.20.

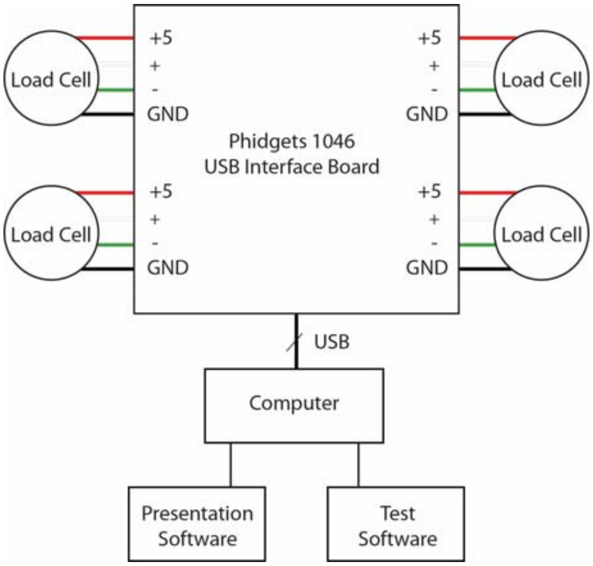


Figure 3.20 | RSS prototype setup

CHAPTER 4 | RSS GROUND BASED TESTING

The simulations performed using MATLAB reinforce the RSS ideal concept of operation perfectly. However, the ideal case is not always a sufficient statistic. Ground based testing was necessary to observe phenomena that would be difficult to simulate using a computer model. The configurations generated by JMP for the simulations in Section 3.2 were reused during ground based testing.

A screening design was used to compare theoretical simulated data with actual measured data. Screening designs consider several factors and identify areas with the greatest effect on the response [41]. The factors considered during testing were: location of force in terms of an *x* and *y* coordinate, and the magnitude of force applied at said location. The goal of each factor was set to obtain minimum values. This type of goal is ideal for detecting experimental impurities and defects [41]. Discrete numerical values were used to define the boundaries of the experiment. The experimental response was linked to the raw voltage measured at each load cell.

4.1 Testing Considerations

Testing was performed using an Autograph AGS-X series precision universal tester manufactured by Shimadzu. The AGS-X provided static and dynamic test forces ranging from 40N to 120N. The aluminum compression tool shown in Figure 4.1 was fabricated specifically for this testing. A round head square neck 3/8”-16 stainless steel bolt provided a *perfect* contact surface. Three potential sources of error were considered during testing: 1) Actual location of contact force vs desired location of contact force. 2) Friction forces generated by the connecting screw, and 3) Fabrication accuracy during RSS fabrication.

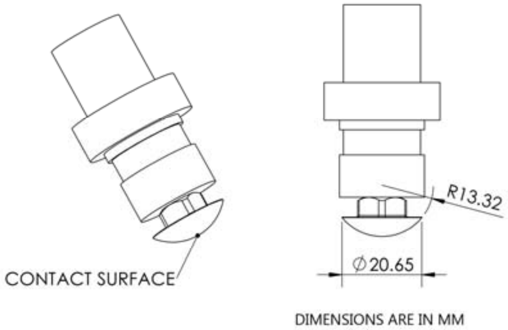


Figure 4.1 | Custom Shimadzu compression tool

4.1.1 Location of Contact

In order to obtain sufficient data, several force locations had to be tested. Recall from Section 3.2 that JMP generated 36 configurations by randomly pulling from a set of nine locations and a set of three forces. The configuration jig in Figure 4.2 was created to assist with quick configuration transitions between trials. The configuration jig was fabricated using 6061 aluminum plate and consists of strategically drilled holes that align with the underlying optical table.

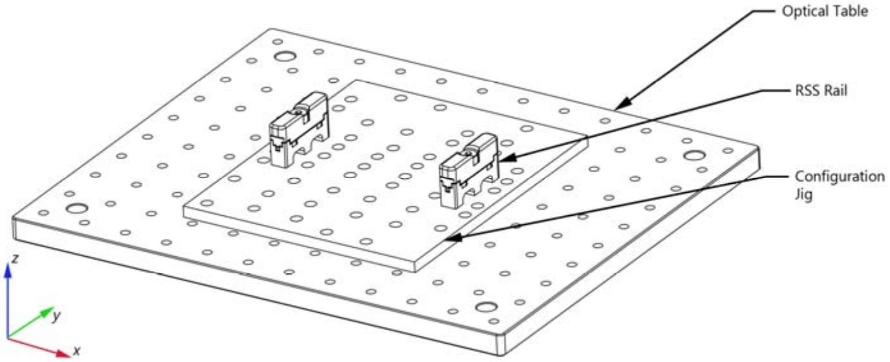


Figure 4.2 | Configuration transition jig

Each jig/optical table alignment corresponds to a predefined force location relative to the RSS rails. The optical table in was initially positioned and fixed to the AGS-X test platform to prevent accidental shifting during configuration transitions. Two hand tightened ¼”-20 screws were fastened in predetermined locations to secure the configuration jig in place during loading.

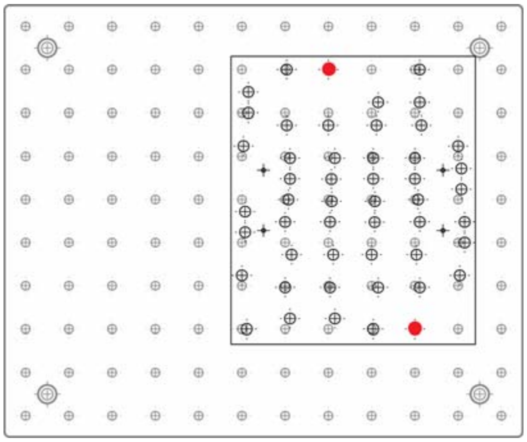


Figure 4.3 | Typical configuration jig alignment diagram

Diagrams like the one in Figure 4.3 were created for each possible test location. The red dots indicate the proper screw locations to achieve the desired contact position.

4.1.2 Unwelcome Forces

Preloading of the connecting screw and load cells was utilized to combat unwelcome friction and lateral force. Each RSS screw was tightened until the load cells reported a value of approximately 44.5N. The exact preload value was recorded for each load cell.

4.1.3 Exactitude of RSS Rails

It is absolutely critical that the contact surfaces of the RSS tool be coplanar. The *slightest* difference between rails can lead to false conclusions and serious consequences. Several factors can lead to uneven contact surfaces including poor fabrication techniques, material imperfections, or improper RSS mounting. The reader is encouraged to revisit Figure 3.7 in Section 3.1.1.

4.2 RSS Two Rail Centroid Reconstruction Testing

The RSS experimental results for the two rail case illustrated in Figure 4.4 are presented in this section. A Dell laptop with an Intel i5 processor and 8GB of RAM was used to run the test software. 36 individual log files were created over a period of 180 minutes. MATLAB was later used to process the string data into numerical vectors. Preloading forces and the weight of the aluminum plate were deducted from the measured data before the data was returned to JMP for

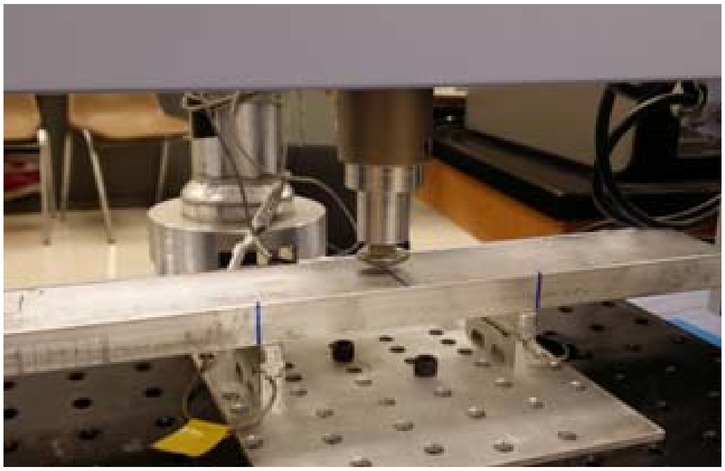


Figure 4.4 | RSS two rail centroid test setup

comparison with the simulated results. The trial configurations are provided on the following page. The experiment results are arranged in the following order: force location relative to RSS rail locations, combined force versus error, and individual load cell error versus applied force and position. Table 4-1 contains the order of configurations tested during this experiment. The data is a repeat of the simulation data provided in Section 3.3.1. It has been duplicated here for convenience.

Table 4-1 | RSS two rail centroid test trials

Run	Location of Force	Magnitude of Force (N)	Run	Location of Force	Magnitude of Force (N)	Run	Location of Force	Magnitude of Force (N)
1	8	40	**13**	8	40	**25**	3	120
2	9	80	**14**	4	40	**26**	1	40
3	5	120	**15**	4	40	**27**	5	120
4	4	40	**16**	9	80	**28**	7	120
5	9	80	**17**	1	40	**29**	2	80
6	6	80	**18**	8	40	**30**	9	80
7	5	120	**19**	2	80	**31**	1	40
8	3	120	**20**	6	80	**32**	5	120
9	7	120	**21**	3	120	**33**	7	120
10	8	80	**22**	4	40	**34**	7	120
11	2	80	**23**	3	120	**35**	2	80
12	1	40	**24**	6	80	**36**	8	40

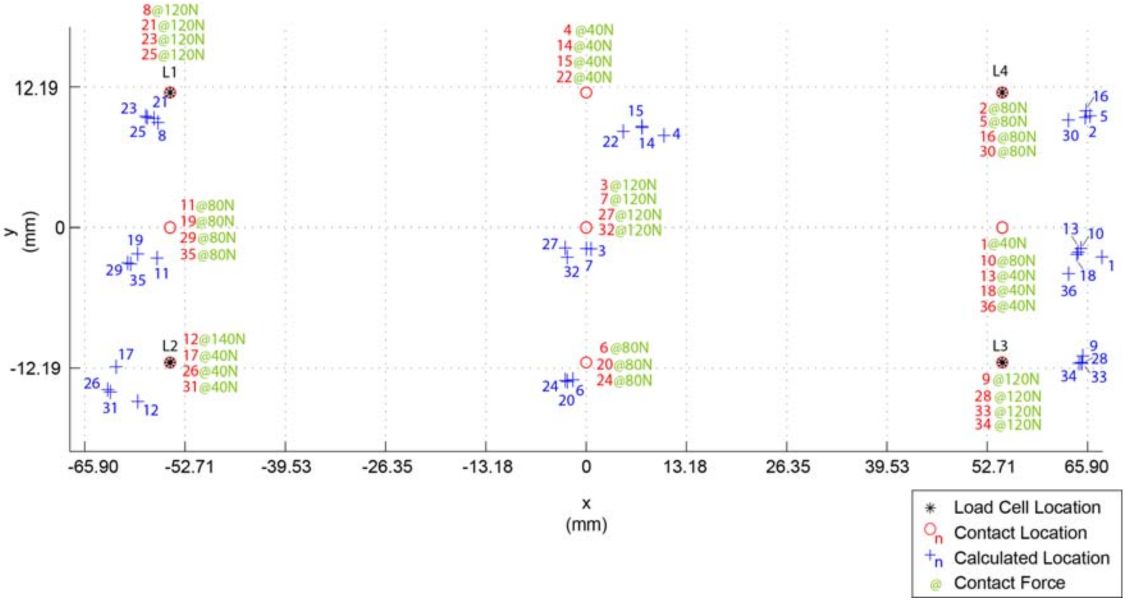


Figure 4.5 | 4-Cell RSS test locations

4.2.1 Summary

The plot in Figure 4.6 illustrates the magnitude of error between simulated load cell response data and the measured data recorded during ground based testing. Eq. (4.1) was used to determine the error.

$$Error = |Theoretical\ mV/V - Actual\ mV/V| \tag{4.1}$$

The data in Figure 4.6 is grouped by the forces applied during testing. F = 1 represents the 40N case, F = 2 represents the 80N case, and F = 3 represents the 120N case. The error plot given in Figure 4.6 suggests larger contact forces induce greater error. Load cell L2 contributes to the majority of error when a force is applied at $y = 0$. Load cells L1, L2, and L4 contribute most of the error when force is applied at the origin. The standard error is largest when 120N of force was applied. The error was smallest when force was applied directly above a load cell at positions 1, 3, 7, and 9. Figure 4.8 through Figure 4.10 illustrate individual load cell error given a location of contact. The raw data recorded during ground based testing is is provided in appendix C.

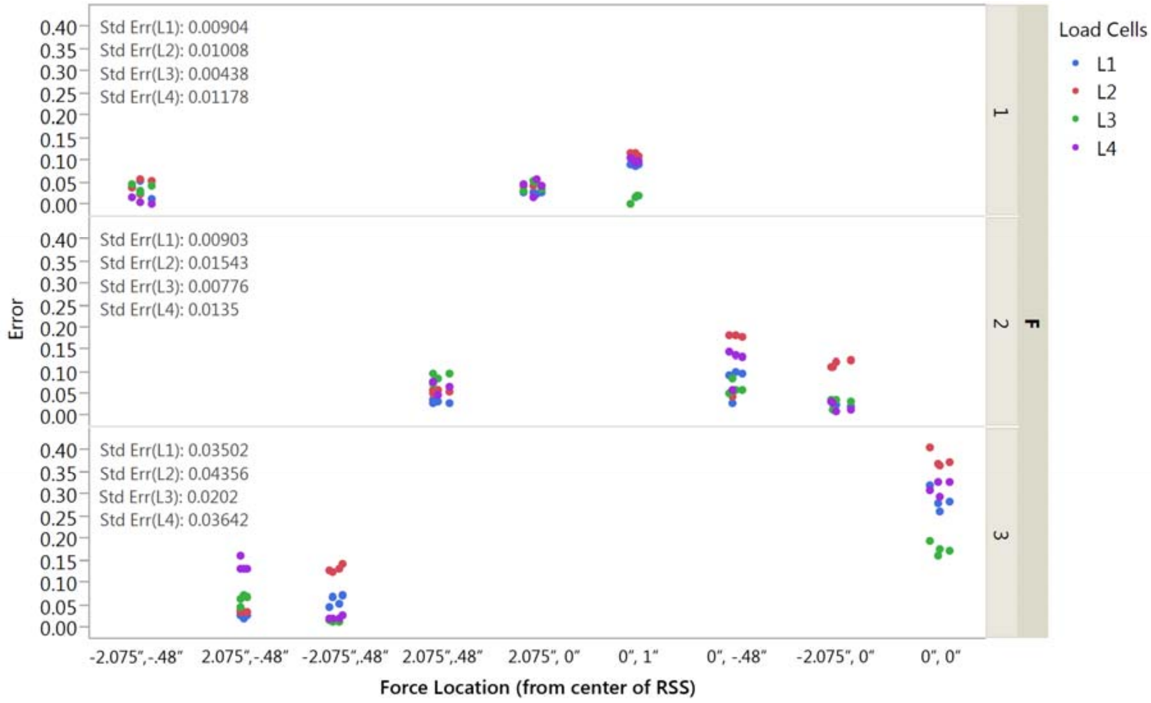


Figure 4.6 | 4-Cell RSS force location vs. mV/V error

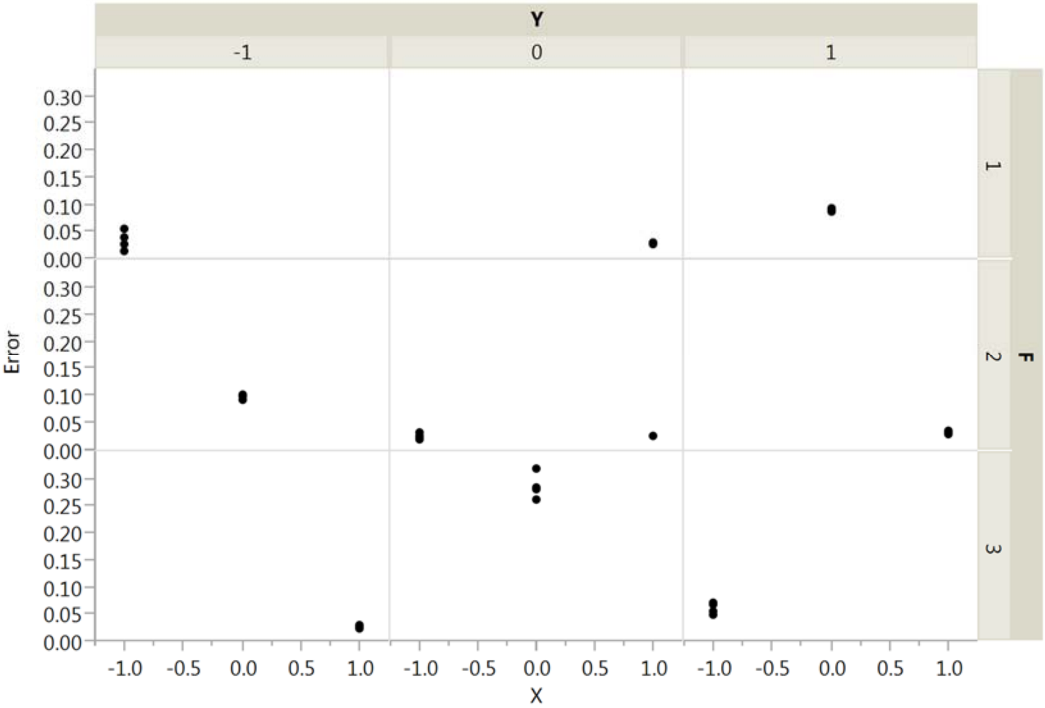


Figure 4.8 | Load cell #1 mV/V Error vs Position & Force

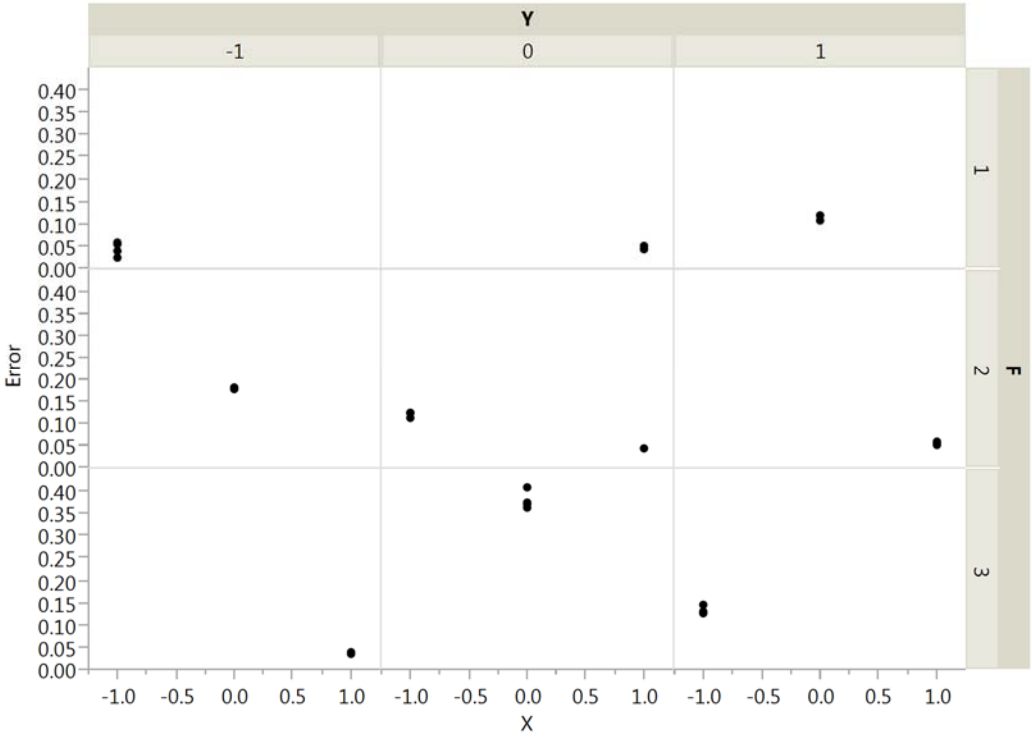


Figure 4.7 | Load cell #2 mV/V Error vs Position & Force

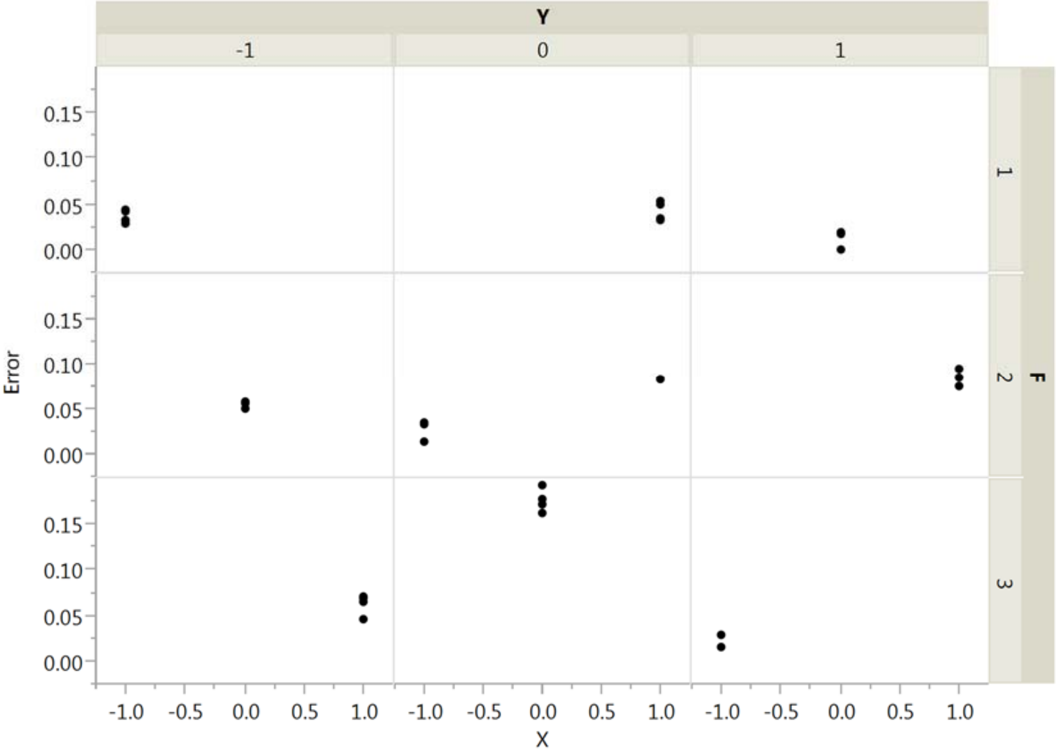


Figure 4.9 | Load cell #3 mV/V Error vs Position & Force

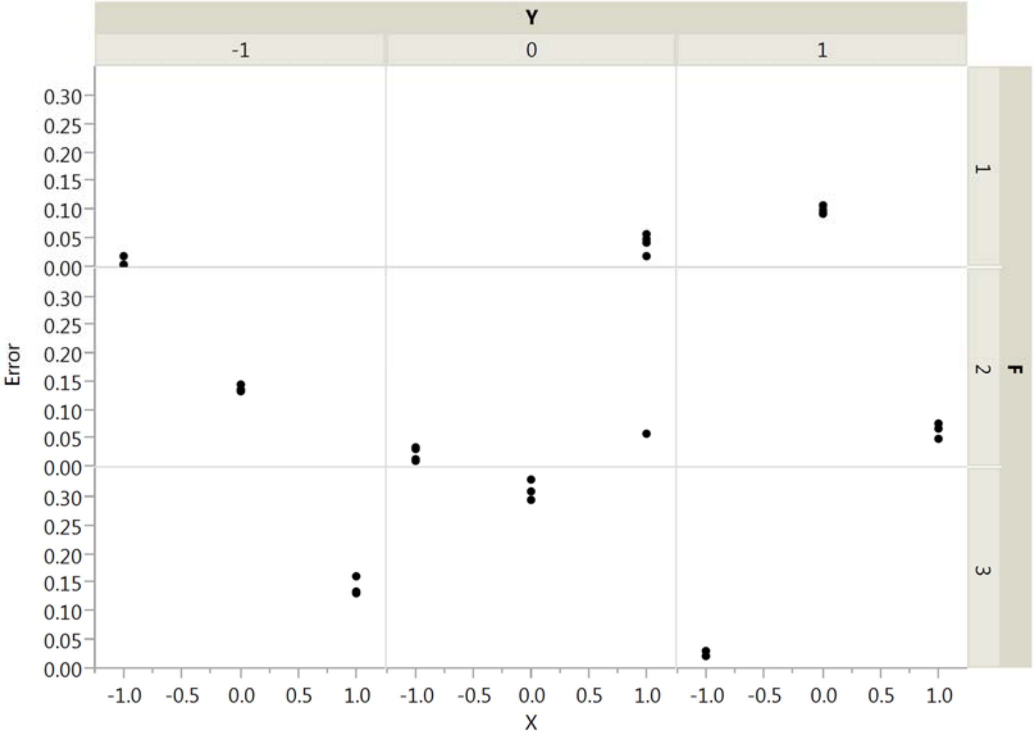


Figure 4.10 | Load cell #4 mV/V Error vs Position & Force

4.3 Fuzzy Logic Simulator

A virtual fuzzy logic controller based on the Mamdani architecture was developed to simulate the ability for an RSS to correct spacecraft trajectory. A desktop XR-1 robotic manipulator with six degrees of freedom and Mark III controller manufactured by RHINO Robots, Inc. was used to simulate the approach of a robotic space service vehicle. The Mark III motor controller was connected to a computer via a modified R232 serial to USB converter. The proper pinout for serial communication with the Mark III is given in Figure 4.11. The C# presentation software discussed in Section 3.5.1 was modified to implement Dmitry Zaluzhny's "Fuzzy Logic Library for Microsoft.NET" fuzzy logic library and manual robot controls [42].

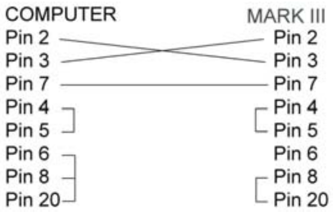


Figure 4.11 | Mark III controller to computer pinout

4.3.1 The RHINO XR-1 and Mark III Controller

The translation speed of the simulated service vehicle was limited by the Mark III controller and its ability to process multiple movements simultaneously. The Mark III motor controller consists of a series of registers that can send and process a single byte of data each transmission. A typical byte demonstrating a move command is provided in Figure 4.12 [43].

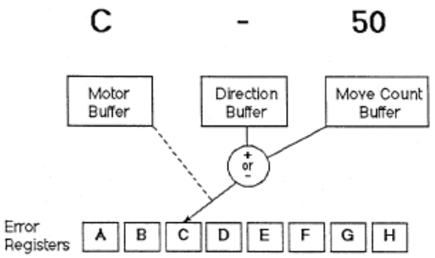


Figure 4.12 | Mark III serial command

When a command is received by the controller it is split up and stored in the corresponding buffer. The controller does not execute the move until a carriage return character is received. When a letter (A-F) is received it is stored in the Motor Buffer until a new motor is specified the Direction Buffer is also set. If a carriage return is received the error buffer is incremented or decremented the proper number of moves. Great care had to be taken at the software level to ensure the Move Count Buffer did not overflow. Sending a new move command while the XR1 is in the middle of an existing move command can cause the XR1 to switch directions if the sum of both moves was greater than 95 [43]. Table 4-2 contains the relevant XR-1 and Mark III control specifications used to determine the fuzzy controller output variables Universe of Discourse in Section 4.3.3.

Table 4-2 | XR-1 and Mark III specifications

Axis	**Motor Gear Ratio**	**Encoder Steps/Degree of Movement**	**Velocity (degrees/sec)**
Rotation	165.4/1	5.5	45
Writst Flex (Pitch)	66.1/1	8.8	45
Wrist (Yaw)	66.1/1	4.4	60

4.3.2 Defining a Coordinate System

Two of the six possible XR-1 degrees of freedom were implemented during the simulations. Pitch and Yaw were used to describe the position of the XR-1's "wrist flex" and "waist" as shown in Figure 4.13. Roll was fixed during the simulations by manually setting the "wrist rotation" of the XR-1 to an optimal approach position. Future ground based testing will most likely require the addition of an emergency retract axis to resist RSS wedging caused by excessive correction angles and rigid targets. This effect would not be as relevant in a space application where the client vehicle does not "push back".

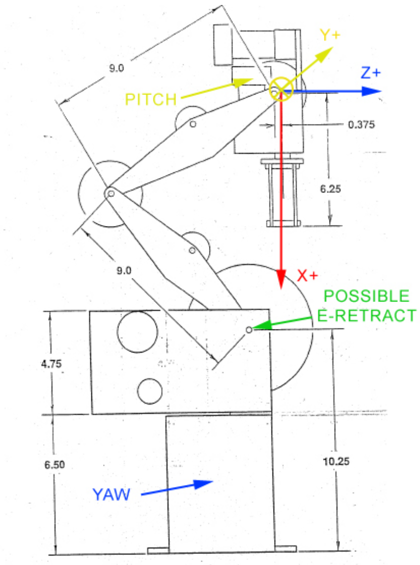


Figure 4.13 | RHINO XR-1 coordinate system

4.3.3 Incorporating Fuzzy Logic

The magnitude of force and angle calculated by the RSS presentation software were initially considered to be the inputs to the fuzzy controller but during initial testing it became clear that the exact magnitude of force at each load cell was not helpful in determining the final fuzzy output. Simply detecting a change in force would suffice. Consider the RSS approach depicted in Figure 4.14. This type of approach illustrates nominal roll and pitch with incorrect yaw. Assuming the load cells are functioning normally and the service vehicle has passed initial approach procedures a single rail response will always indicate a poor approach regardless of the rail magnitude. In fact, until both rails come in contact with the client vehicle the RSS will report

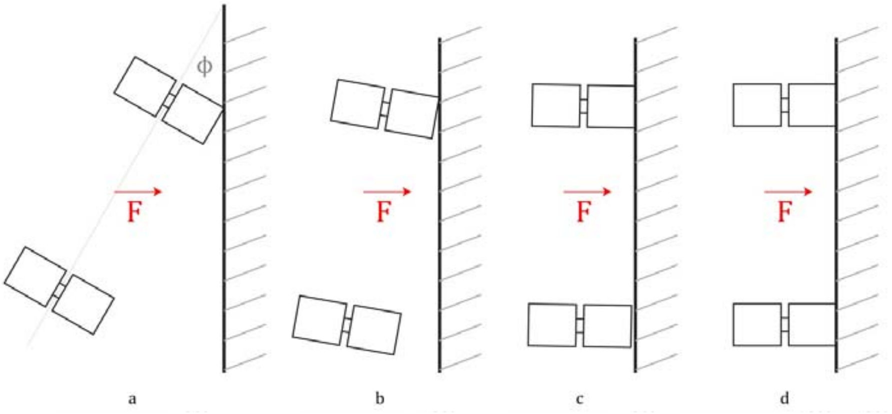


Figure 4.14 | Force present during RSS contact

an approach angle ϕ equal to 90 degrees regardless of the magnitude of force. It was this observation that lead to the final fuzzy input variables: current yaw and current pitch.

The Universe of Discourse for the input variable membership functions and the grid overlay illustrated in Figure 3.17 were confined to the C# *Math* class and the return value of the *Atan2* function. The return value for this specific function is determined using the Cartesian plane coordinates *x* and *y* where $\tan\theta = y / x$ [44]:

- For (x, y) in quadrant 1, 0 < θ < π/2.
- For (x, y) in quadrant 2, π/2 < θ ≤ π.
- For (x, y) in quadrant 3, -π < θ < -π/2.
- For (x, y) in quadrant 4, -π/2 < θ < 0.

and for points on the boundaries of the quadrants:

- If y is 0 and x is not negative, θ = 0.
- If y is 0 and x is negative, θ = π.
- If y is positive and x is 0, θ = π/2.
- If y is negative and x is 0, θ = -π/2.
- If y is 0 and x is 0, θ = 0.

The input values *x* and *y* were determined using the readings from each load cell and the resulting centroid of force calculated using Eq. 2.6 and Eq. 2.7. The value returned by the *Atan2* function was inserted into the fuzzy system and passed through a single combined input variable "Current Approach" illustrated in Figure 4.15. The first letter of each membership function represents the axis being evaluated and the second letter represents the sign of the angle. For example "yN1" represents the membership function related to negative yaw inputs and "pZ1" represents zero valued pitch inputs.

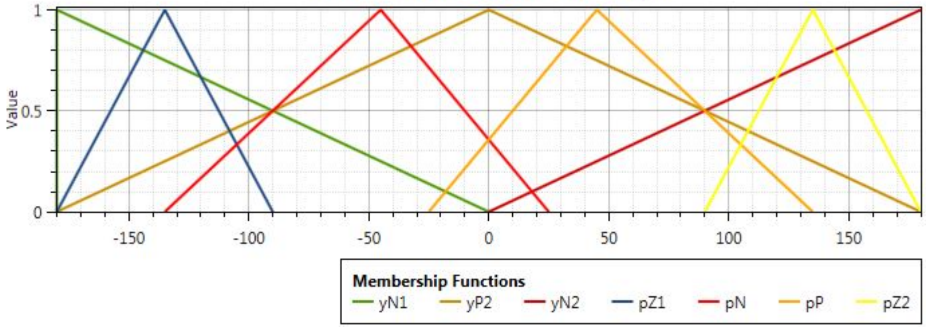


Figure 4.15 | Universe of Discourse for input variable "Current Approach"

The fuzzy output variables include "Corrected Pitch" and "Corrected Yaw" as shown in Figure 4.16 and Figure 4.17 respectively. The Universe of Discourse of each output variable was determined solely on the specifications of the XR-1 motors, encoders, and Mark III controller provided in Table 4-2.

The output of each output variable relates to the normalized number of encoder steps that the XR-1 needs to take to correct the trajectory. Before being sent to the motor controller the normalized output is multiplied by a scaling "step multiplier". The step multiplier controls the speed of the XR-1 motors. The default step multiplier was initially set to 1.6 but the presentation software GUI was modified to include a textbox and slider for multiplier modification at runtime. After initial testing the yaw axis was given an additional multiplier of eight to achieve smooth motions.

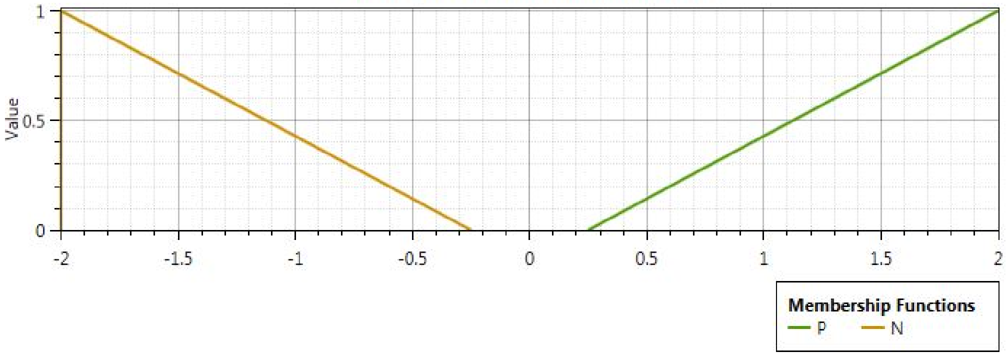


Figure 4.16 | Universe of Discourse for output variable "Corrected Pitch"

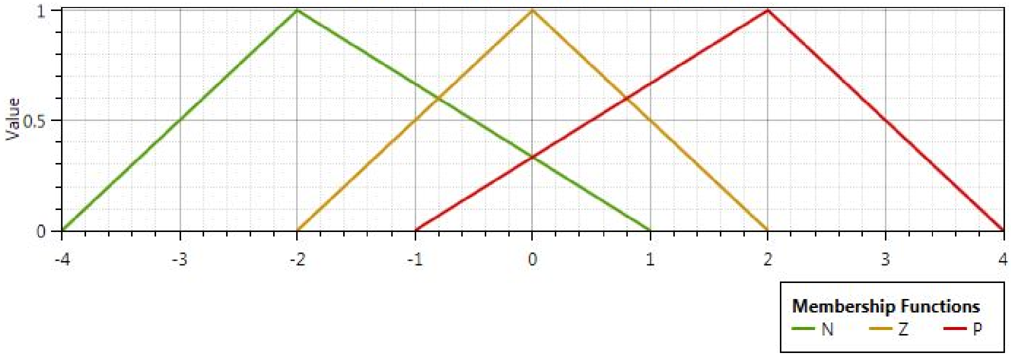


Figure 4.17 | Universe of Discourse for output variable "Corrected Yaw"

Recall from Section 3.1 that the largest deviation between the client and target vehicles should be less than 15 degrees. Referring to Table 4-2 it is clear that the yaw *(4.4 steps/degree)* is different than the pitch *(8.8 steps/degree)*. Their translation velocities are also different. Pitch translates at 45 degrees per second while Yaw travels at 60 degrees per second. Excessive steps could cause a 'floating' object to be pushed away. It was determined through trial and error that sending a large number of small steps was better than sending one giant step. A crisp output was determined using the centroid method and a series of non-additive *if-then* fuzzy rules. Each axis was considered independent of the other. For example pitch is described as:

If (Current Approach is pN) *then* (Corrected Pitch is N)
If (Current Approach is pP) *then* (Corrected Pitch is P)
If (Current Approach is pZ1) *then* (Corrected Pitch is Z)
If (Current Approach is pZ2) *then* (Corrected Pitch is Z)

and for yaw:

If (Current Approach is yP2) *then* (Corrected Yaw is P)
If (Current Approach is yN1) *then* (Corrected Yaw is N)
If (Current Approach is yN2) *then* (Corrected Yaw is N)

The MATLAB "Fuzzy Logic Toolbox" was used to visualize the output of the fuzzy system graphically. Figure 4.18 contains an example RSS fuzzy output for current yaw equal to 10.7

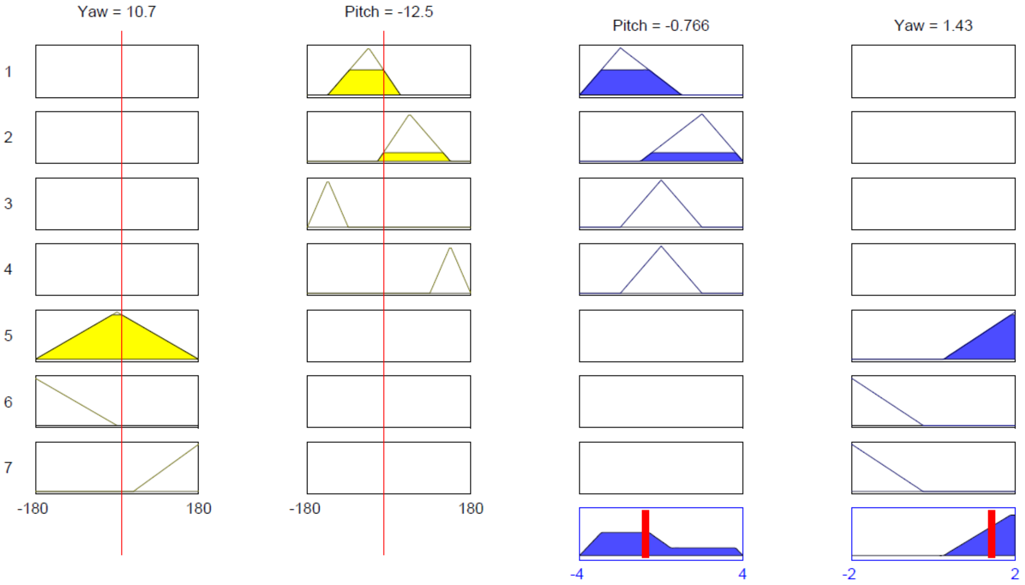


Figure 4.18 | Graphical representation of fuzzy output

degrees and current pitch equal to -12.5 degrees resulting in a corrected yaw output equal to 1.43 and corrected pitch output of -.766. Applying the appropriate multipliers requires the yaw motor move a total of 18 steps while the pitch motor only requires one step. This process repeats until a "lock" on the target is achieved. Applying a larger step multiplier would generate larger movements at the risk of excessive force applied to the client target. Figure 4.19 illustrates the hardware arrangement and setup used during fuzzy testing.

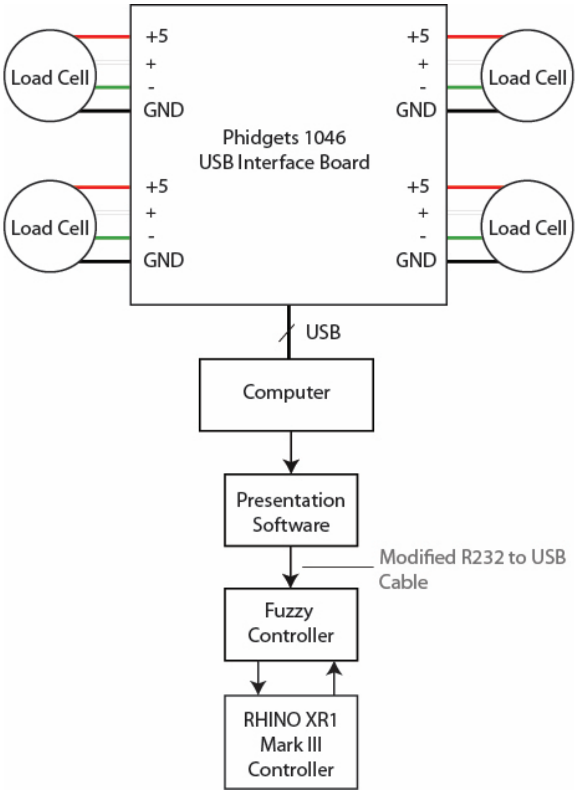


Figure 4.19 | Fuzzy Logic test setup

4.3.4 Summary

The fuzzy logic simulator functioned as expected. Incorporating a fuzzy logic controller into the system allowed real-time trajectory modifications to be quickly processed and executed to achieve nominal approach angles. The RHINO-XR1 and Mark III controller were the limiting factor during testing. The speed at which trajectory modifications could be made was limited by the buffer size and serial transmission of the Mark III motor controller.

CHAPTER 5 | RSS SPACE CONSIDERATIONS

The results and simulations provided in the previous sections do not take into account the harsh environment of space. Several factors must be considered before the RSS can be considered "flight ready". Radiation shielding, temperature, and stability over time are three of the primary concerns observed over the course of this research.

5.1 Space Radiation

There are three fundamental types of radiation that can directly effect a spacecraft including radiation from the sun, the Earth's atmosphere, and space itself [16]. Space radiation can wreak havoc on a number of mechanical and electrical components including: microelectronics, photonics, materials, epoxies, and biological systems [45]. NASA focuses its attention primarily on the natural space (atmospheric) radiation environment [45]. The effects of natural space radiation and ionizing can be divided into two parts: a total ionizing dose (TID) and single event effects (SEE) [46] [47]. TID is defined by the Radiation Effects and Analysis Group as a long-term degradation of electronics due to a cumulative energy deposited in a material. Excessive TID exposure arises from trapped electrons, trapped protons, or solar protons as shown in

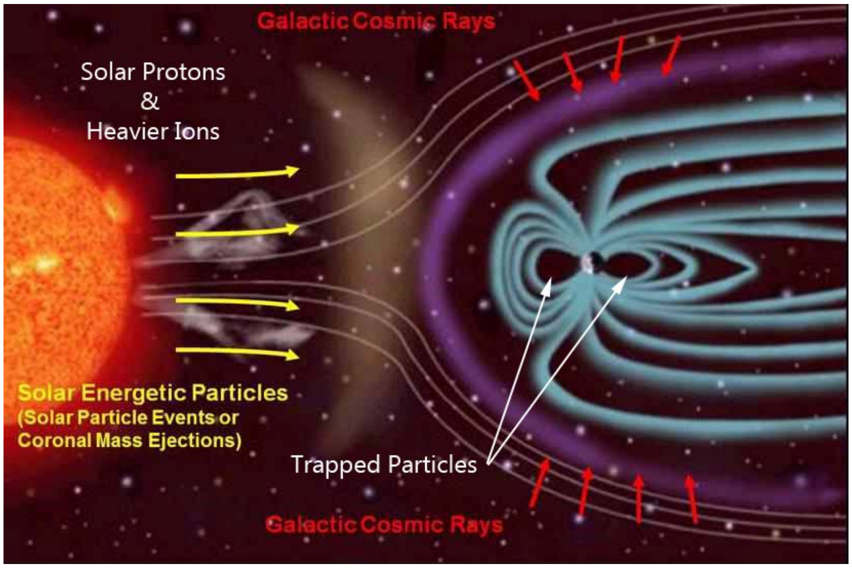


Figure 5.1 | Galactic cosmic rays and trapped particles

Figure 5.1 [46] [47] [48]. SEE can be further divided into two categories such as soft errors and hard errors. Soft errors typically result in an unexpected transient pulse or bit-flip (single event upset) in an electronic device and may or may not be recovered [46] [47].

Hard errors such as single hard errors (SHE) are most often physically destructive resulting in catastrophic failure or a permanent change in device operation [46] [47]. The presence of ionizing particle environments causes a unique concern on the reliability of commercially available hardware and electronics used in space applications [47]. The level of radiation protection or requirements is specific to the orbit for a particular mission [45]. LaBel et al provide the following rational approach when considering the effects of radiation on space systems [45]:

- Define the environment
- Evaluate the environment
- Define the requirements
- Evaluate the design/components
- "Engineer" with designers
- Iterate Process

Based on this process it is required that the electronics used to record strain gage data are shielded against ionized particles and radiation. The strain gage itself is not as susceptible to these effects. However, shielding from electromagnetic interference (EMI) should be considered.

5.2 Temperature

A thermal control system should be incorporated to ensure the RSS electronics and other service vehicle hardware operate in the proper temperature regions. On-orbit spacecraft are prone to extreme temperature fluctuations governed by the relative location of the sun. Typical satellite electronic equipment has an operating temperature of -20ºC to 55ºC [16]. **Without** *modulus compensation* the Omega load cells are rated for a temperature between -54 and 121ºC. This range is reduced to between 16 to 71ºC with the manufacturer supplied temperature compensation circuit. The Phidgets interface board operates in a similar temperature range. It is clear that the load cells and USB interface board used throughout ground based testing do not comply with typical satellite operating temperature ranges.

Two types of thermal control systems that are commonly used in satellite technology: passive and active could assist the RSS region of operation [16]. Passive applications employ multilayer insulation surfaces such as Kapton or MLI to control the propagation of radiation. This method could potentially provide a layer of protection for the strain gage, bonding cement, and solder points. Active systems are often employed to protect electronic equipment exposed to large extreme temperature fluctuations. Heat pipes, controlled refrigeration, and heaters make up the majority of active systems currently used in satellite technology [16]. RSS electronics should be incorporated into an all-encompassing active thermal control system.

5.3 Stability in Time

For a strain gage to remain useful over time one must consider its stability in time. Zero drift is an effect manifested over time and is specific to the gage alloy, the boding cement, and the carrier of the gage [30]. Mechanical hysteresis is experienced when a strain gage is repeatedly loaded and unloaded [30]. A similar phenomenon, electrical hysteresis, can be reproduced by pressing an electrical push button repeatedly at high frequencies. Hysteresis is a function of 1) maximum deformation applied and 2) number of applied cycles [30]. Systems that expect rapid dynamic forces should employ strain gages with low hysteresis.

CHAPTER 6 | CONCLUSION

6.1. Summary

Tests conducted with the Rail Sensor System (RSS) reinforce the concept of reconstructing contact forces using an array of strain gages and their calculated centroid of force. The raw voltage values reported by the load cells contain valuable information that can potentially provide teleoperators and autonomous algorithms the information necessary to determine nominal service vehicle approach angles. A data polling technique and large sample rates were used to combat the explicit effect of electrical noise in the fuzzy controller.

To increase the effectiveness of the RSS it is recommended that the number of components used to construct the tool be reduced. Currently, the RSS is composed of two aluminum rails, two load cells, and a connecting screw. Incorporating commercially available load cells and electronics provide a fast but solid test foundation that does not require extensive upfront design costs. However, custom strain gages should be considered for future RSS prototypes. The proposed RSS design provides a satisfactory fixture for mounting the load cells but the fabrication, assembly, and machining of the tool must be carefully monitored to ensure proper functionality. The contacting face located on the top of the RSS rails must remain coplanar for the tool to be effective. It was found that non coplanar rail faces lead to incorrect contact locations.

The addition of a fuzzy controller simplifies the processes of generating translation modifications to the current approach angle without the need for linear algebra and rotation matrices. A RHINO XR-1 robotic manipulator successfully demonstrated the role of the fuzzy controller.

6.2. Future Work

Although the current RSS tool provides acceptable results it will be necessary to perform a redesign of the RSS rail and how the tool is assembled. Pinching caused by the connecting screw leads to unwanted friction and can produce unfavorable results. The cost of incorporating a friction resistant shoulder screw does not provide a sufficient return on the results. The friction resistant connecting screw still poses potential for the top rail to stick. The addition of a spring has been considered but has been avoided due to the complexity it brings to the force calculations. Solutions to the space considerations introduced in Chapter 5 should also be incorporated into future RSS designs.

The methods for determining a centroid of force described in this research have the ability to be extended into larger arrays and systems. The client berthing system (CBS) shown in Figure 6.1 is a perfect example of said system. Future plans for a CBS play a crucial role in the grapple, berth, and servicing of a client spacecraft. The transparent ring depicted in the rendering represents the GOES interface ring discussed in Section 3.1 . The client satellite was omitted for visual clarity. The centroid of force method presented in earlier sections can be extended to the measurement of static and dynamic forces present on the CBS as presented in Figure 6.2. A CBS prototype is currently being developed for further testing.

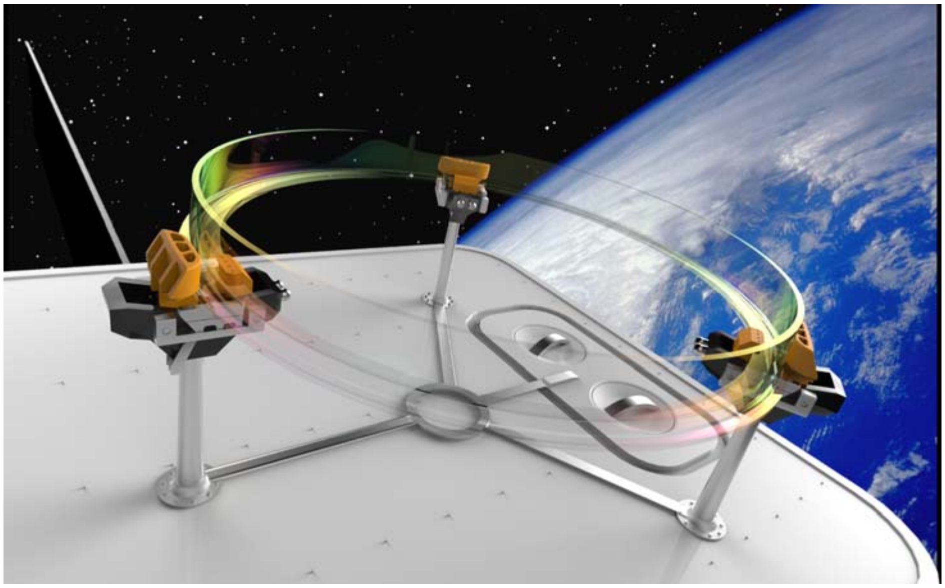


Figure 6.1 | CBS conceptual rendering

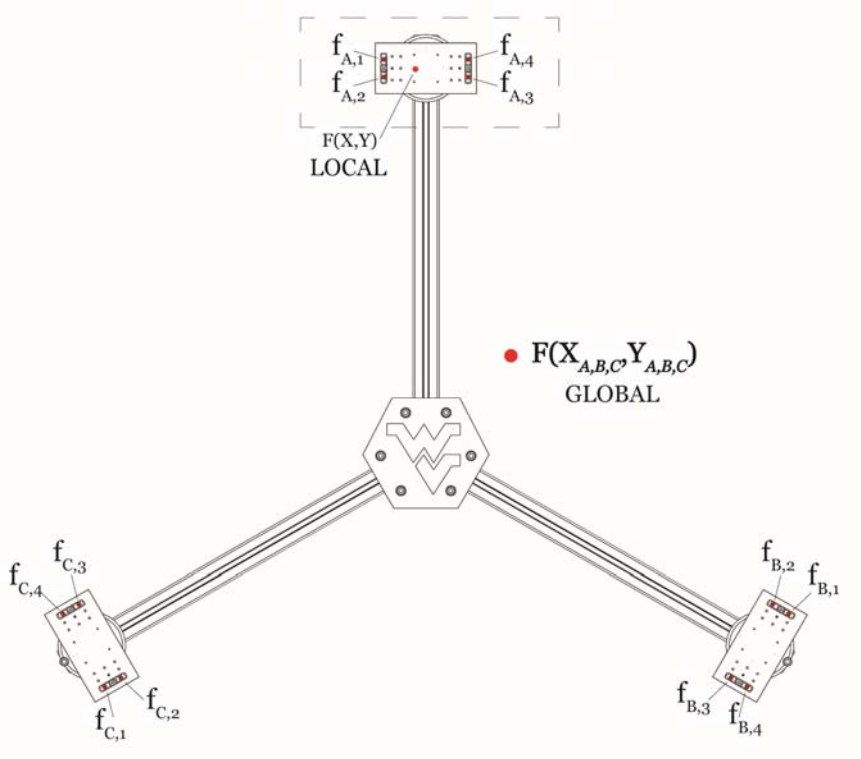


Figure 6.2 | CBS centroid of force

6.3. Lessons Learned

The location of the load cell relative to the mounting location is an area of concern. A slight cantilever of the load cell over the mounting screw counterbore may result in minor strain gage error. The load cell manufacturer recommends the entire bottom face of the load cell remain flush to a fixed surface. This effect could be eliminated by increasing the size of the RSS hitbox. Unfortunately the bolt pattern for the current tool was held to strict compliances relative to the gripping unit and could not be modified.

Imperfections in the aluminum RSS rails were still present despite the use of CNC machinery. A large portion of the error recorded during testing was directly related to these imperfections. The values returned by the RSS system will only be as accurate as the parts that make up the system.

BIBLIOGRAPHY

[1] E. Coleshill, L. Oshinowo, R. Rembala, B. Bina, D. Rey and S. Sindelar, "Dextre: Improving maintenance operations on the International Space Station," *Acta Astronautica,* vol. 64, no. 2009, pp. 869-874, 2008.

[2] C. J. Anders and C. H. Roy, "On-Orbit Spacecraft Servicing," *AIAA,* vol. 94, no. 1236, pp. 441-450, 1994.

[3] B. Reed, Interviewee, *Robotic Refeuling Mission.* [Interview]. 16 January 2013.

[4] Virgin Galactic, "Our Vision of the Future," Virgin Galactic, [Online]. Available: http://www.virgingalactic.com/our-vision-of-the-future/. [Accessed 20 03 2015].

[5] NASA, "Service Vehicle Servicing a Client," [Online]. Available: http://ssco.gsfc.nasa.gov/robotic_servicing_mission.html. [Accessed 19 December 2014].

[6] J. L. Goodman, "History of Space Shuttle Rendezvous," NASA, Houston, 2011.

[7] J. Cook, V. Aksamentov, T. Hoffman and W. Bruner, "ISS Interface Mechanisms and their Heritage," 2011. [Online]. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110010964.pdf. [Accessed 13 October 2014].

[8] G. Sparrow, J. John and C. McNab, The Illustrated Encyclopedia of Space & Space Exploration, New York: Sterling Publishing, 2014.

[9] NASA, "International Space Station Imagery," 30 October 2012. [Online]. Available: http://spaceflight.nasa.gov/gallery/images/station/crew-30/html/iss030e015599.html. [Accessed 19 December 2014].

[10] CSA, "Canadarm," CSA, 9 October 2014. [Online]. Available: http://www.asc-csa.gc.ca/eng/canadarm/default.asp. [Accessed 21 December 2014].

[11] CSA, "Flight History of Canadarm," CSA, 28 October 2011. [Online]. Available: http://www.asc-csa.gc.ca/eng/canadarm/flight.asp. [Accessed 20 December 2014].

[12] NASA, "Hubble Space Telescope Servicing Missions Overview," NASA, [Online]. Available: http://www.nasa.gov/mission_pages/hubble/servicing/. [Accessed 29 December 2014].

[13] NASA, "Remote Manipulator System," NASA, [Online]. Available: http://prime.jsc.nasa.gov/ROV/rms.html. [Accessed 24 December 2014].

[14] STScl, "Team Hubble: Servicing Missions," STScl, [Online]. Available: http://hubblesite.org/the_telescope/team_hubble/servicing_missions.php. [Accessed 29 December 2014].

[15] M. J. Clay, Crowded Orbits Conflict and Cooperation in Space, New York: Columbia University, 2014.

[16] A. K. Maini and V. Agrawal, Satellite Technology Principles and Applications, 3rd ed., Chichester: John Wiley & Sons Ltd, 2014.

[17] Union of Concerned Scientists, "UCS Satellite Database," 1 August 2014. [Online]. Available: http://www.ucsusa.org/nuclear_weapons_and_global_security/solutions/space-weapons/ucs-satellite-database.html. [Accessed 13 October 2014].

[18] CSA, "International Space Station," CSA, 5 March 2014. [Online]. Available: http://www.asc-csa.gc.ca/eng/iss/default.asp. [Accessed 22 December 2014].

[19] NASA, Reference guide to the International Space Station, Washington, DC: NASA, 2006.

[20] NASA, "International Space Station - International Cooperation," 30 October 2014. [Online]. Available:

http://www.nasa.gov/mission_pages/station/cooperation/index.html. [Accessed 21 December 2014].

[21] NASA, "Robotic Refueling Mission," 25 January 2013. [Online]. Available: http://ssco.gsfc.nasa.gov/rrm_refueling_task-blog.html. [Accessed 19 October 2014].

[22] CSA, "Image Gallery - Mobile Servicing System (MSS)," CSA, 10 February 2014. [Online]. Available: http://www.asc-csa.gc.ca/eng/search/images/watch.asp?id=394&search=MSS . [Accessed 21 Decemeber 2014].

[23] J. Nimon, "Top Space Station Research Awards of the Year - Technology," NASA, 17 July 2013. [Online]. Available: http://www.nasa.gov/content/top-space-station-research-awards-of-the-year-technology/stationresearch/. [Accessed 24 December 2014].

[24] A. Cuaron, Director, *Gravity.* [Film]. United States of America: Warner Bros. Pictures, 2013.

[25] M. Sznajder, "A short review of a degradation effects of materials under space conditions," in *Global Virtual Conference*, 2013.

[26] Y. Freedman, University Physics, vol. I, Boston: Pearson Custom Publishing, 2000.

[27] R. L. Hannah and S. E. Reed, Strain Gage Users' Handbook, New York: Elsevier Science Publishers Ltd., 1992.

[28] D. Gross, W. Hauger, J. Schroder, W. A. Wall and N. Rajapakse, Engineering Mechanics 1 Statics, 2nd ed., Dordrecht: Springer, 2013.

[29] C. C. Perry and H. R. Lissner, The Strain Gage Primer, 2nd ed., New York: McGraw-Hill Book Company, Inc., 1962.

[30] C. Sciammarella, "Strain Gages - Introduction to Electrical Strain Gages," in *Experimental Mechanics of Solids*, 1st ed., John Wiley & Sons, Ltd, 2012, pp. 65-97.

[31] R. B. Watson, "Bonded Electrical Resistance Strain Gages," in *Springer Handbook of Experimental Solid Mechanics*, W. N. Sharpe, Ed., New York, Springer, 2008, pp. 283-333.

[32] Goodfellow, "Constantan - Resistance Alloy Material Information," Goodfellow, 2014. [Online]. Available: http://www.goodfellow.com/E/Constantan-Resistance-Alloy.html. [Accessed 9 November 2014].

[33] MFL, "Strain Gages," MFL, [Online]. Available: http://www.mflstraingauges.com/index_gauge_types.html. [Accessed 9 November 2014].

[34] Omega, "The Strain Gage," [Online]. Available: http://www.omega.com/literature/transactions/volume3/strain2.html. [Accessed 6 August 2014].

[35] Omega, "Subminiature Load Cell," 2003-2014. [Online]. Available: http://www.omega.com/pptst/LCKD.html.

[36] Schunk, "PG70 Specifications," [Online]. Available: http://www.schunk.com/schunk_files/attachments/PG_70_EN.pdf. [Accessed 25 November 2014].

[37] T. Evans, Interviewee, *West Virginia Robotic Technology Center Research Program Manager*. [Interview]. 21 October 2014.

[38] W. H. Hayt, J. E. Kemmerly and S. M. Durbin, Engineering Circuit Analysis, 7th ed., New York: McGraw Hill, 2007.

[39] Phidgets, "1046 User Guide," 8 August 2013. [Online]. Available: http://www.phidgets.com/docs/1046_User_Guide.

[40] Microsoft, "Windows Presentation Foundation," Microsoft, 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms754130%28v=vs.110%29.aspx. [Accessed 11 11 2014].

[41] SAS, "JMP Design of Experiments Guide," SAS, Cary.

[42] D. Zaluzhny, "Fuzzy Logic Library for Microsoft.Net," 24 April 2013. [Online]. Available: http://sourceforge.net/projects/fuzzynet/. [Accessed 31 December 2014].

[43] RHINO Robots, Inc., XR Owner's Manual, Champaign: Rhino Robots, Inc..

[44] Microsoft, "Math.Atan2 Method," Microsoft, 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/system.math.atan2%28v=vs.110%29.aspx. [Accessed 31 December 2014].

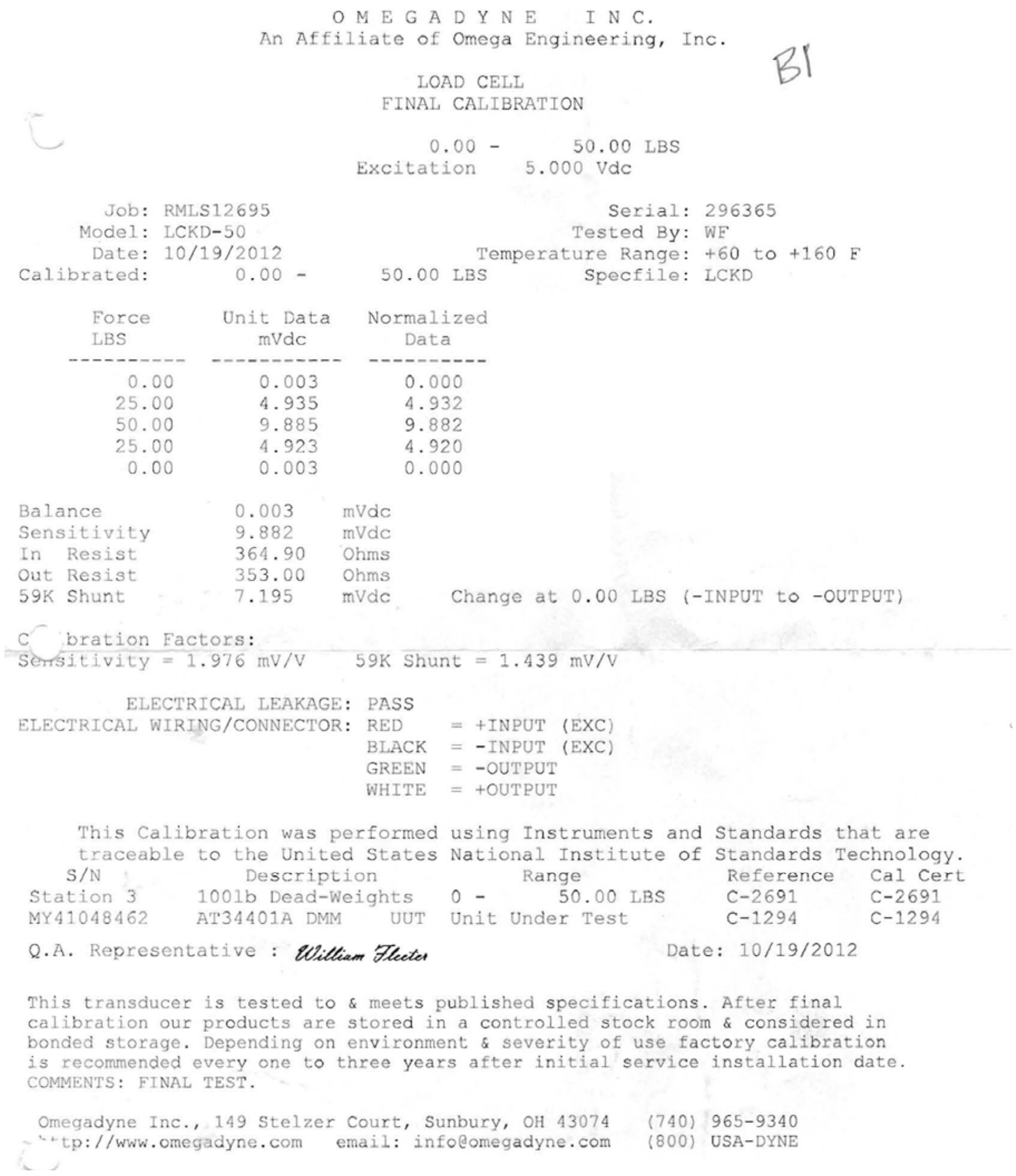
[45] K. LaBel and J. Barth, "An Insider's Guide to Designing Spacecraft Systems and Instuments for Operation in the Natural Space Radiation Environment," in *GSFC Systems Engineering Seminar*, 2001.

[46] M. O'Bryan, "Natural Space Radiation Effects on Technology," NASA Radiation Effects and Analysis Group, 15 November 2000. [Online]. Available: http://radhome.gsfc.nasa.gov/radhome/Nat_Space_Rad_Tech.htm.

[47] K. LaBel, M. Gates and A. Moran, "Commercial Microelectronics Technologies for Applications in the Satellite Radiation Environment," in *IEEE Aerospace Applications*, 1996.

[48] K. LaBel, "A NASA Perspective on Validation and Testing of Design Hardening for the Natural Space Radiation Enviornment," in *GOMAC*, Tampa, 2003.

APPENDIX A | TYPICAL OMEGA LOAD CELL CALIBRATION SHEET



OMEGADYNE INC.
An Affiliate of Omega Engineering, Inc.

B1

LOAD CELL
FINAL CALIBRATION

0.00 - 50.00 LBS
Excitation 5.000 Vdc

Job:	RMLS12695	Serial:	296365
Model:	LCKD-50	Tested By:	WF
Date:	10/19/2012	Temperature Range:	+60 to +160 F
Calibrated:	0.00 - 50.00 LBS	Specfile:	LCKD

Force LBS	Unit Data mVdc	Normalized Data
0.00	0.003	0.000
25.00	4.935	4.932
50.00	9.885	9.882
25.00	4.923	4.920
0.00	0.003	0.000

Balance	0.003	mVdc	
Sensitivity	9.882	mVdc	
In Resist	364.90	Ohms	
Out Resist	353.00	Ohms	
59K Shunt	7.195	mVdc	Change at 0.00 LBS (-INPUT to -OUTPUT)

Calibration Factors:
Sensitivity = 1.976 mV/V 59K Shunt = 1.439 mV/V

ELECTRICAL LEAKAGE: PASS
ELECTRICAL WIRING/CONNECTOR: RED = +INPUT (EXC)
BLACK = -INPUT (EXC)
GREEN = -OUTPUT
WHITE = +OUTPUT

This Calibration was performed using Instruments and Standards that are traceable to the United States National Institute of Standards Technology.

S/N	Description	Range	Reference	Cal Cert
Station 3	100lb Dead-Weights	0 - 50.00 LBS	C-2691	C-2691
MY41048462	AT34401A DMM UUT	Unit Under Test	C-1294	C-1294

Q.A. Representative : William Fleeter Date: 10/19/2012

This transducer is tested to & meets published specifications. After final calibration our products are stored in a controlled stock room & considered in bonded storage. Depending on environment & severity of use factory calibration is recommended every one to three years after initial service installation date.
COMMENTS: FINAL TEST.

Omegadyne Inc., 149 Stelzer Court, Sunbury, OH 43074 (740) 965-9340
http://www.omegadyne.com email: info@omegadyne.com (800) USA-DYNE

APPENDIX B | C++ TEST SOFTWARE SOURCE CODE

```
// RSS.cpp
// RSS Test Software
// Jason Battin

#include <iostream>
#include "Phidget.h"
#include <phidget21.h>
#include <stdio.h>
#include <fstream>         //Log
#include <conio.h>         //keyboard hit
#include <vector>
#include <Windows.h>

using namespace std;

enum PossiblePhidgetDevices {
	PHIDGET_A = 140800, //Serial
	PHIDGET_B = 341104,
	PHIDGET_C = 341185
};

//Globals
Phidget _Phidget;

bool run;

int configs;
int configStart = 1;

//function declaration
void runSetup();

//Executes when Phidgets board is attached
int CCONV AttachHandler(CPhidgetHandle phid, void *userptr)
{
	CPhidgetBridgeHandle bridge = (CPhidgetBridgeHandle)phid;

	CPhidgetBridge_setEnabled(bridge, 0, PTRUE);
	CPhidgetBridge_setEnabled(bridge, 1, PTRUE);
	CPhidgetBridge_setEnabled(bridge, 2, PTRUE);
	CPhidgetBridge_setEnabled(bridge, 3, PTRUE);

	CPhidgetBridge_setGain(bridge, 0, PHIDGET_BRIDGE_GAIN_128);
	CPhidgetBridge_setGain(bridge, 1, PHIDGET_BRIDGE_GAIN_128);
	CPhidgetBridge_setGain(bridge, 2, PHIDGET_BRIDGE_GAIN_128);
	CPhidgetBridge_setGain(bridge, 3, PHIDGET_BRIDGE_GAIN_128);
	CPhidgetBridge_setDataRate(bridge, 1000);

	printf("Attach handler ran!\n");
	return 0;
}

//Executes when Phidgets board is detached
int CCONV DetachHandler(CPhidgetHandle phid, void *userptr)
{
	printf("Detach handler ran!\n");
	return 0;
}
```

```
//Executes when Phidgets board experiences an error
int CCONV ErrorHandler(CPhidgetHandle phid, void *userptr, int ErrorCode, const
char *errorStr)
{
	printf("Error event: %s\n", errorStr);
	return 0;
}

//Main program
int main(int argc, char* argv[])
{
	const char *err;
	int result;

	//Execute main menu
	runSetup();

	//Was the start test option selected?
	if (run)
	{
		//Construct Phidgets object
		CPhidgetBridgeHandle bridge;
		CPhidgetBridge_create(&bridge);

		//Register event handlers
		CPhidget_set_OnAttach_Handler((CPhidgetHandle)bridge, AttachHandler,
		NULL);
		CPhidget_set_OnDetach_Handler((CPhidgetHandle)bridge, DetachHandler,
		NULL);
		CPhidget_set_OnError_Handler((CPhidgetHandle)bridge, ErrorHandler,
		NULL);

		//Connect to Phidgets interface board
		CPhidget_open((CPhidgetHandle)bridge, -1);

		//Wait for 10 seconds, otherwise exit
		if (result = CPhidget_waitForAttachment((CPhidgetHandle)bridge,
		10000))
		{
			CPhidget_getErrorDescription(result, &err);
			printf("Problem waiting for attachment: %s\n", err);
			return 0;
		}

		//Containers for each Phidget value
		double val_0 = 0, val_1 = 0, val_2 = 0, val_3 = 0;

		//This is the configuration loop. Its length is set by the //user
		before the test is executed.

		for (int i = configStart; i <= configs; i++)
		{
			//True when ready to move to next test config
			bool ready = false;
			char userKey;

			//Pause exectution until the user is ready
			while (!ready)
			{
```

```
                cout << "\nEnter g key when ready to begin run " << i
                << endl;
                cin >> userKey;

                if  (userKey == 'g')
                    ready = true;

            }

            //Create a log file so that data can be analyzed at a later
            time.
            //ofstream logFile(buffer, ios::out);
            char buffer[15];
            int logPath = sprintf_s(buffer, "C:\\Logs\\%d.txt", i);

            //Increase buffer size to reduce risk of overflow
            std::vector<char> vec(4096);
            std::fstream fs(buffer, ios::out);
            fs.rdbuf()->pubsetbuf(&vec.front(), vec.size());

            //Output the run number to the log file
            fs << "Test Description" << endl;
            fs << "===========================" << endl;
            fs << "Run: #" << i  << "\n" << endl;

            //Collect and log data until user hits the Enter key
            while (!_kbhit())
            {

                CPhidgetBridge_getBridgeValue(bridge, 0, &val_0);
                CPhidgetBridge_getBridgeValue(bridge, 1, &val_1);
                CPhidgetBridge_getBridgeValue(bridge, 2, &val_2);
                CPhidgetBridge_getBridgeValue(bridge, 3, &val_3);

                cout << val_0 << " " << val_1 << " " << val_2 << " "
                << val_3 << "\n";
                fs << val_0 << " " << val_1 << " " << val_2 << " " <<
                val_3 << endl;

                //Wait 100ms before collecting next sample
                Sleep(100);

            }
            //Dump the enter key
            cout << _getch();
            //flush the text file buffer
            fs.flush();
            //Close the text file
            fs.close();
        }

        //Tear down Phidgets objects
        CPhidget_close((CPhidgetHandle)bridge);
        CPhidget_delete((CPhidgetHandle)bridge);
    }

    //Exit program
    return 0;
}
```

```
//Print main menu to screen
void runSetup()
{
    bool startTest = false;
    int userSelect = -1;
    bool configsSet = false;

    //Display menu until user starts a new test of exits current test
    while (!startTest)
    {
        cout << "RSS System Test Setup
        v1.0.0\n===============================\n";
        cout << "1) Select Subsystem\n";
        cout << "2) Configurations\n";
        cout << "3) Starting Configuration\n";
        cout << "4) Start Test\n";
        cout << "5) Exit Test\n";
        std::cin >> userSelect;

        switch (userSelect)
        {
        case 1:
            char userSel;
            cout << "Enter RSS Subsytem ID [A-C]: ";
            std::cin >> userSel;

            if (userSel == 'A' || userSel == 'a')
            {
                _Phidget.setSerial(PHIDGET_A);
                _Phidget.setID('A');
                _Phidget.setEnabled(true);
            }
            else if (userSel == 'B' || userSel == 'b')
            {
                _Phidget.setSerial(PHIDGET_B);
                _Phidget.setID('B');
                _Phidget.setEnabled(true);
            }
            else if (userSel == 'C' || userSel == 'c')
            {
                _Phidget.setSerial(PHIDGET_C);
                _Phidget.setID('C');
                _Phidget.setEnabled(true);
            }
            else {
                cout << "Invalid Input!\n";
            }
            break;
        case 2:
            cout << "How many configurations will be tested?\n";
            std::cin >> configs;
            configsSet = true;
            break;
        case 3:
            cout << "Enter a new starting point:\n";
            cin >> configStart;
            break;
        case 4:
            if (configsSet)
            {
                if (_Phidget.getEnabled())
                {
                    run = true;
                    startTest = true;
                }
                else
                {
                    cout << "Subsystem not selected!\n";
                }
            }
            else
            {
                cout << "Number of configurations not set!\n";
            }
            break;
        case 5:
            run = false;
            startTest = true;
            break;

        default:
            cout << "Invalid Input!\n";
        }
    }
}
```

```
// Phidget.hpp
// RSS Test Software
// Jason Battin

#include "Bridge.h"

/* Note: Using a preprocessor wrapper to prevent the code in the header
* from being included in the same source code file more than once.
*/
#ifndef PHIDGET_HPP_
#define PHIDGET_HPP_

class Phidget
{
public:
       Phidget();
       Phidget(int);
       //~Phidget();

       void setID(char);
       void setSerial(int);
       void setEnabled(bool);
       void setBridgeEnabled(int, bool);
       void setAllBridgesEnabled();
       void setBridgeGain(int, CPhidgetBridge_Gain);

       char getID();
       int getSerial();
       bool getEnabled();
       bool getBridgeEnabled(int);
       CPhidgetBridge_Gain getBridgeGain(int);


protected:
       int _serial;
       bool _enabled;
       char _id;
       Bridge _bridges[4];
};

#endif /* PHIDGET_HPP_ */
```

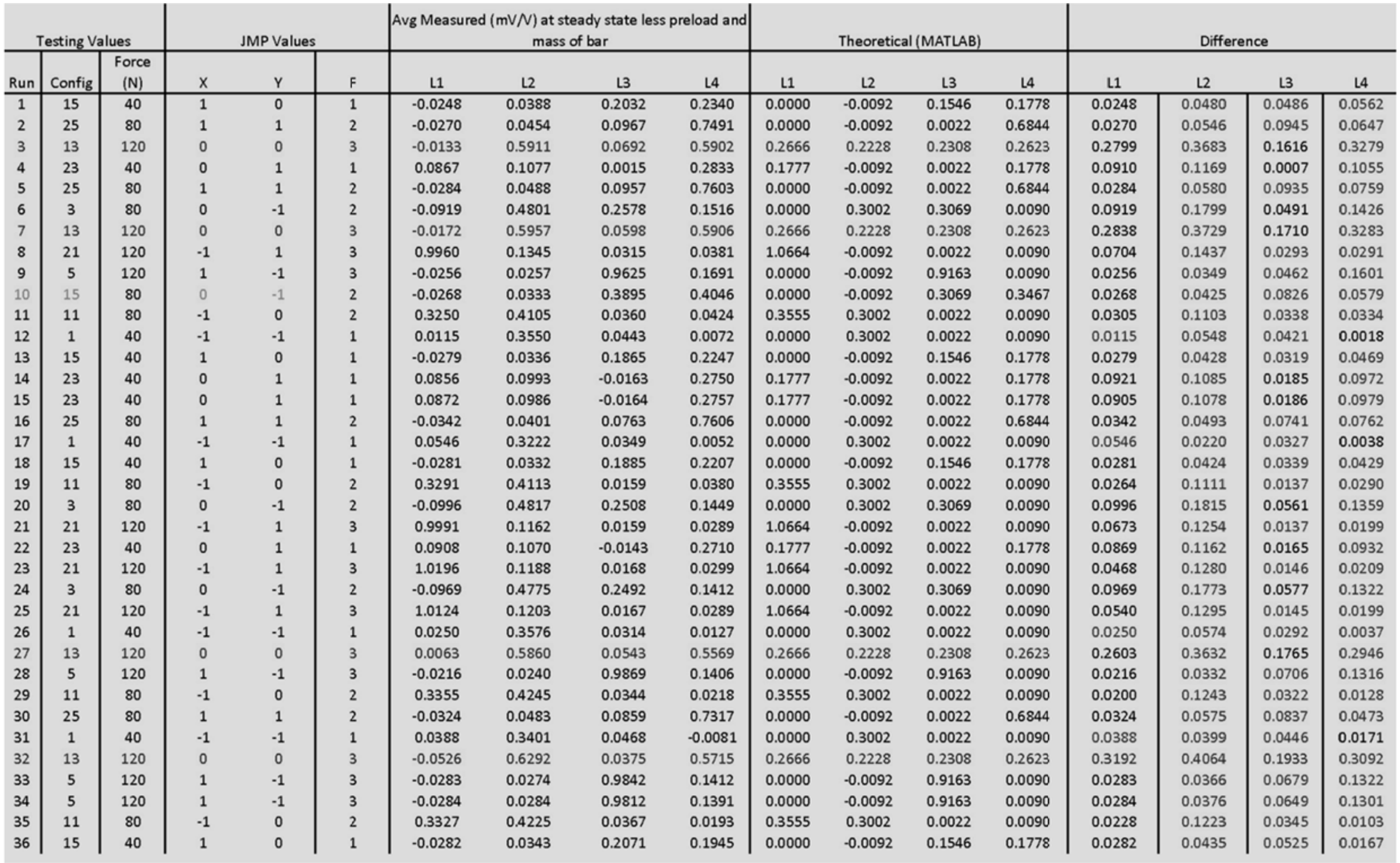
```
// Phidget.cpp
// RSS Test Software
// Jason Battin

#include <phidget21.h>
#include "Phidget.h"


/* NOTE: '::' represents the binary scope resolution operator.
* This ties each member function to the class definition which
* declares the class's member functions and data members. To
* create a global function exclude this operator.
*/
Phidget::Phidget()
{
       _id = 'E';
       _serial = -1;
       _enabled = false;
       _bridges[0] = Bridge(0);
       _bridges[1] = Bridge(1);
       _bridges[2] = Bridge(2);
       _bridges[3] = Bridge(3);

}

Phidget::Phidget(int serial)
{
       _id = 'E';
       _serial = serial;
       _enabled = false;
}

void Phidget::setID(char id)
{
       _id = id;
}

void Phidget::setSerial(int serial)
{
       _serial = serial;
}

void Phidget::setEnabled(bool enabled)
{
       _enabled = enabled;
}

void Phidget::setBridgeEnabled(int index, bool enabled)
{
       _bridges[index].setEnabled(enabled);
}

void Phidget::setAllBridgesEnabled()
{
       for (int i = 0; i < 4; i++)
       {
              _bridges[i].setEnabled(true);
       }
}
```

```
void Phidget::setBridgeGain(int index, CPhidgetBridge_Gain gain)
{
	_bridges[index].setGain(gain);
}

char Phidget::getID()
{
	return _id;
}

int Phidget::getSerial()
{
	return _serial;
}

bool Phidget::getEnabled()
{
	return _enabled;
}

bool Phidget::getBridgeEnabled(int index)
{
	return _bridges[index].getEnabled();
}

CPhidgetBridge_Gain Phidget::getBridgeGain(int index)
{
	return _bridges[index].getGain();
}
```

```
// Bridge.hpp
// RSS Test Software
// Jason Battin

#include <phidget21.h>

#ifndef BRIDGE_HPP_
#define BRIDGE_HPP_

class Bridge
{
public:
        Bridge();
        Bridge(int);

        void setEnabled(bool);
        void setIndex(int);
        void setGain(CPhidgetBridge_Gain);

        bool getEnabled();
        int getIndex();
        CPhidgetBridge_Gain getGain();



protected:
        bool _enabled;
        int _index;
        CPhidgetBridge_Gain _gain;

};



#endif /* BRIDGE_HPP_ */
```

```
// Bridge.cpp
// RSS Test Software
// Jason Battin

#include "Bridge.h"


Bridge::Bridge() :_index(-1), _enabled(false), _gain(PHIDGET_BRIDGE_GAIN_128)
{
}

Bridge::Bridge(int index)
{
       _gain = PHIDGET_BRIDGE_GAIN_128;
       _enabled = false;
       _index = index;
}

void Bridge::setEnabled(bool enabled)
{
       _enabled = enabled;
}

void Bridge::setIndex(int index)
{
       _index = index;
}

void Bridge::setGain(CPhidgetBridge_Gain gain)
{
       _gain = gain;
}

bool Bridge::getEnabled()
{
       return _enabled;
}

int Bridge::getIndex()
{
       return _index;
}

CPhidgetBridge_Gain Bridge::getGain()
{
       return _gain;
}
```

APPENDIX C | RAW SENSOR DATA FROM GROUND BASED TEST



Testing Values			JMP Values			Avg Measured (mV/V) at steady state less preload and mass of bar				Theoretical (MATLAB)				Difference			
Run	Config	Force (N)	X	Y	F	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4
1	15	40	1	0	1	-0.0248	0.0388	0.2032	0.2340	0.0000	-0.0092	0.1546	0.1778	0.0248	0.0480	0.0486	0.0562
2	25	80	1	1	2	-0.0270	0.0454	0.0967	0.7491	0.0000	-0.0092	0.0022	0.6844	0.0270	0.0546	0.0945	0.0647
3	13	120	0	0	3	-0.0133	0.5911	0.0692	0.5902	0.2666	0.2228	0.2308	0.2623	0.2799	0.3683	0.1616	0.3279
4	23	40	0	1	1	0.0867	0.1077	0.0015	0.2833	0.1777	-0.0092	0.0022	0.1778	0.0910	0.1169	0.0007	0.1055
5	25	80	1	1	2	-0.0284	0.0488	0.0957	0.7603	0.0000	-0.0092	0.0022	0.6844	0.0284	0.0580	0.0935	0.0759
6	3	80	0	-1	2	-0.0919	0.4801	0.2578	0.1516	0.0000	0.3002	0.3069	0.0090	0.0919	0.1799	0.0491	0.1426
7	13	120	0	0	3	-0.0172	0.5957	0.0598	0.5906	0.2666	0.2228	0.2308	0.2623	0.2838	0.3729	0.1710	0.3283
8	21	120	-1	1	3	0.9960	0.1345	0.0315	0.0381	1.0664	-0.0092	0.0022	0.0090	0.0704	0.1437	0.0293	0.0291
9	5	120	1	-1	3	-0.0256	0.0257	0.9625	0.1691	0.0000	-0.0092	0.9163	0.0090	0.0256	0.0349	0.0462	0.1601
10	15	80	0	-1	2	-0.0268	0.0333	0.3895	0.4046	0.0000	-0.0092	0.3069	0.3467	0.0268	0.0425	0.0826	0.0579
11	11	80	-1	0	2	0.3250	0.4105	0.0360	0.0424	0.3555	0.3002	0.0022	0.0090	0.0305	0.1103	0.0338	0.0334
12	1	40	-1	-1	1	0.0115	0.3550	0.0443	0.0072	0.0000	0.3002	0.0022	0.0090	0.0115	0.0548	0.0421	0.0018
13	15	40	1	0	1	-0.0279	0.0336	0.1865	0.2247	0.0000	-0.0092	0.1546	0.1778	0.0279	0.0428	0.0319	0.0469
14	23	40	0	1	1	0.0856	0.0993	-0.0163	0.2750	0.1777	-0.0092	0.0022	0.1778	0.0921	0.1085	0.0185	0.0972
15	23	40	0	1	1	0.0872	0.0986	-0.0164	0.2757	0.1777	-0.0092	0.0022	0.1778	0.0905	0.1078	0.0186	0.0979
16	25	80	1	1	2	-0.0342	0.0401	0.0763	0.7606	0.0000	-0.0092	0.0022	0.6844	0.0342	0.0493	0.0741	0.0762
17	1	40	-1	-1	1	0.0546	0.3222	0.0349	0.0052	0.0000	0.3002	0.0022	0.0090	0.0546	0.0220	0.0327	0.0038
18	15	40	1	0	1	-0.0281	0.0332	0.1885	0.2207	0.0000	-0.0092	0.1546	0.1778	0.0281	0.0424	0.0339	0.0429
19	11	80	-1	0	2	0.3291	0.4113	0.0159	0.0380	0.3555	0.3002	0.0022	0.0090	0.0264	0.1111	0.0137	0.0290
20	3	80	0	-1	2	-0.0996	0.4817	0.2508	0.1449	0.0000	0.3002	0.3069	0.0090	0.0996	0.1815	0.0561	0.1359
21	21	120	-1	1	3	0.9991	0.1162	0.0159	0.0289	1.0664	-0.0092	0.0022	0.0090	0.0673	0.1254	0.0137	0.0199
22	23	40	0	1	1	0.0908	0.1070	-0.0143	0.2710	0.1777	-0.0092	0.0022	0.1778	0.0869	0.1162	0.0165	0.0932
23	21	120	-1	1	3	1.0196	0.1188	0.0168	0.0299	1.0664	-0.0092	0.0022	0.0090	0.0468	0.1280	0.0146	0.0209
24	3	80	0	-1	2	-0.0969	0.4775	0.2492	0.1412	0.0000	0.3002	0.3069	0.0090	0.0969	0.1773	0.0577	0.1322
25	21	120	-1	1	3	1.0124	0.1203	0.0167	0.0289	1.0664	-0.0092	0.0022	0.0090	0.0540	0.1295	0.0145	0.0199
26	1	40	-1	-1	1	0.0250	0.3576	0.0314	0.0127	0.0000	0.3002	0.0022	0.0090	0.0250	0.0574	0.0292	0.0037
27	13	120	0	0	3	0.0063	0.5860	0.0543	0.5569	0.2666	0.2228	0.2308	0.2623	0.2603	0.3632	0.1765	0.2946
28	5	120	1	-1	3	-0.0216	0.0240	0.9869	0.1406	0.0000	-0.0092	0.9163	0.0090	0.0216	0.0332	0.0706	0.1316
29	11	80	-1	0	2	0.3355	0.4245	0.0344	0.0218	0.3555	0.3002	0.0022	0.0090	0.0200	0.1243	0.0322	0.0128
30	25	80	1	1	2	-0.0324	0.0483	0.0859	0.7317	0.0000	-0.0092	0.0022	0.6844	0.0324	0.0575	0.0837	0.0473
31	1	40	-1	-1	1	0.0388	0.3401	0.0468	-0.0081	0.0000	0.3002	0.0022	0.0090	0.0388	0.0399	0.0446	0.0171
32	13	120	0	0	3	-0.0526	0.6292	0.0375	0.5715	0.2666	0.2228	0.2308	0.2623	0.3192	0.4064	0.1933	0.3092
33	5	120	1	-1	3	-0.0283	0.0274	0.9842	0.1412	0.0000	-0.0092	0.9163	0.0090	0.0283	0.0366	0.0679	0.1322
34	5	120	1	-1	3	-0.0284	0.0284	0.9812	0.1391	0.0000	-0.0092	0.9163	0.0090	0.0284	0.0376	0.0649	0.1301
35	11	80	-1	0	2	0.3327	0.4225	0.0367	0.0193	0.3555	0.3002	0.0022	0.0090	0.0228	0.1223	0.0345	0.0103
36	15	40	1	0	1	-0.0282	0.0343	0.2071	0.1945	0.0000	-0.0092	0.1546	0.1778	0.0282	0.0435	0.0525	0.0167