2015

Advances in MEMS IMU Cluster Technology for Small Satellite Applications

Drew E. Bittner

Recommended Citation

Bittner, Drew E., "Advances in MEMS IMU Cluster Technology for Small Satellite Applications" (2015). *Graduate Theses, Dissertations, and Problem Reports*. 5216.
https://researchrepository.wvu.edu/etd/5216

Advances in MEMS IMU Cluster Technology for Small Satellite Applications

Drew E. Bittner

Thesis submitted to

The Benjamin M. 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 Mechanical Engineering

Committee Members

John A. Christian, WVU, Chair Ph.D.
Mario Perhinschi, WVU, Ph.D.
Robert H. Bishop, USF, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia 2015

Keywords & Phrases: Inertial Navigation, Inertial Measurement Unit, IMU, IMU cluster, Redundant IMU, IMU Calibration, IMU Calibration Apparatus, MEMS, FDIR

ABSTRACT

Advances in MEMS IMU Cluster Technology for Small Satellite Applications

Drew E. Bittner

In recent years, there has been increased interest in Micro-Electro Mechanical Systems (MEMS) Inertial Measurement Units (IMUs) due to their relatively small volumetric footprint and low-cost. Although this advantage far outweighs the volumetric footprint and cost of traditional high-performance IMUs, MEMS technology has yet to match the performance of such devices. In spite of this, it has been shown in theory that a cluster of MEMS IMUs may significantly improve the performance over a single MEMS IMU. To further develop this theory, two prototype boards have been designed and constructed that include 16 MEMS accelerometers and gyroscopes affixed to a single Printed Circuit Board (PCB). To prove this technology, hardware and software has been developed for calibration and fault detection, which represents the majority of the body of this thesis. An apparatus has been designed to easily acquire three-axis measurements from the cluster prototype on a single-axis rate table. These measurements may then be placed into a Maximum Likelihood Estimation (MLE) algorithm in order to acquire the necessary error coefficients incorporated in IMU measurements. Once these error coefficients are accurately determined, future measurements may be calibrated. Finally, a fault detection, isolation, and recovery (FDIR) architecture was developed and simulated to determine faulty measurements in real-time, so that bad measurements may not be placed into downstream navigation filters. The hardware, software, and testing developed and performed in this thesis will be used in the verification process of an IMU cluster to help prove its worthiness in modern day small satellite applications.

Acknowledgments

This journey would not have been possible without the support of my friends, family, professors, and mentors. To my family, thank you for encouraging me to constantly pursue higher education and continued learning. Especially to my parents, without your emotional and financial support along the way, I would not be where I stand today. To all of my friends and roommates at PSU and WVU, thank you for listening, providing advice, and supporting me through the last six years. It has been a wild ride, but knowing that you were always there made it all the more comforting.

I am very grateful to all the professors and mentors during my time at Penn State Altoona and at West Virginia University. Specifically, I would like to thank Dr.Grant Risha of Penn State Altoona, for your knowledge, expertise, research, and teaching led me to where I am today. Early in my time at PSU you saw the potential and mentored me through my undergraduate academic career and for that I am forever grateful. To Dr.John Christian, thank you for providing me with the opportunity to work in aerospace navigation. You have greatly helped diversify my engineering background with inertial navigation and I am very appreciative of your continued support, advising, expertise, and effort through the completion of this thesis. I am also very appreciative to all the members of the ASEL team for all the support they provided along the way. I can only hope that I provided each of you with the same level of support during my term as a graduate student. In addition, I am very thankful to the support provided by Dr. Robert Bishop of the University of South Florida and Daniel Greenheck of Marquette University for their knowledge and expertise of the software and hardware on this project. Thank you for working diligently in the final months of my graduate career so that I had the necessary materials and data to present in this thesis. I am also thankful to thank Darryl May and David Fanelli of Johnson Space Center for their countless hours of support of testing and knowledgeable advice on this project. Special thanks to NASA cooperative agreement NNX13AQ79A for the funding in support of this research.

Most importantly, to my beautiful and intelligent fiancée Jordan, thank you for always being there for me and helping me through these last six years. You encouraged me to pursue this masters degree, and while it was a lot of work, you were always there to help in the most trying times. We work great together, not only professionally but at home as well, and that's why I can't wait to see what our future has in store.

Contents

List of Figures

List of Tables

Acronyms

MEMS	Micro Electro-Mechanical Systems
IMU	Inertial Measurement Unit
PCB	Printed Circuit Board
MLE	Maximum Likelihood Estimation
FDIR	Fault Detection, Isolation, and Recovery
JSC	Johnson Space Center
USIP	Undergraduate Student Implementation Project
FOG	Fiber Optic Gyroscope
ARW	Angle Random Walk
AVAR	Allan Variance
ADEV	Allan Deviation
FPGA	Field Programmable Gate Array
ADC	Analog to Digital Converter
CubeSat	Cube Satellite

Nomenclature

Swarm/Cluster	A multiple sensors system fused to create a common output
δ	Scale factor error
Γ	Misalignment error
Ξ	Non-orthogonality error
β	Bias error
w_s	A 3×1 synthetically generated XYZ sensor body rate truth
T_S^I	A 3×3 rotation matrix converting the inertial frame to the sensor frame
H	Measurement sensitivity matrix
R	Measurement covariance matrix
i	A variable specifying the sensor number in the swarm
k	A variable specifying the orientation and rate combination
γ	Angle of the rate table when it is rotating
θ	Outer rotation angle of the Patent-Pending calibration apparatus
α	Inner rotation angle of the Patent-Pending calibration apparatus
ϑ	Articulating bracket angle
φ	Manual rotation stage angle
$\tilde{y}$	A 3×1 body rate measurement
y_{cal}	A 3×1 calibrated body rate measurement

1 Introduction

1.1 Motivation

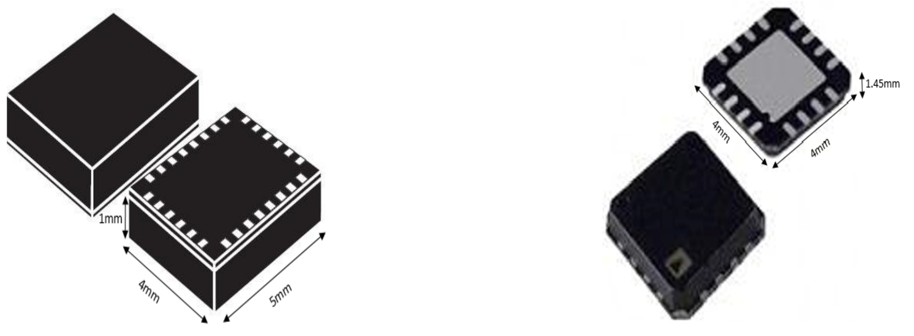
Over the past century, man has become entertained with the idea of navigation, which is mostly linked to the wonders of space travel, space exploration, and automation. Without navigation it is impossible to explore the deepest darkest corners of our Earth and Galaxy. One notable contribution to the history of navigation is the Global Positioning System (GPS). GPS has been used over the past several decades as the go-to navigation system for nearly all aerospace and ground related positioning needs and only in recent years has it been commercialized such that the public may now reap its benefits. Currently, a constellation of 24 GPS satellites broadcast navigation related data down to Earth with the use of antennas. To acquire this information for tracking and positioning, a user must obtain a GPS antenna and receiver of which can be found in most modern day store-bought car GPS's. As long as you are underneath the GPS constellation and have a clear view of the sky, then tracking is relatively straight-forward. But, what about tracking above the GPS constellation, or positioning below the surface of the earth? This brings to light one obvious yet critical disadvantage of GPS. Without a clear view of the GPS satellites in orbit with a user's GPS Antenna, it is not possible to track or position an object using GPS. Therefore, when a vehicle goes into a building or in through a tunnel no GPS data is available to help.

This disadvantage of such a system is solved by means of inertial navigation. Inertial navigation does not use external sources for positioning and tracking, but rather is packaged as a self-contained system. The principle behind an inertial navigation system is that a fixed mass is either rotated or accelerated, as long as the mass is known, the velocity and acceleration may be measured. Accelerometers are used to measure linear acceleration, and changes in inertial velocity may be estimated by integrating the accelerometer output. it is possible to obtain the velocity of the vehicle body. Gyroscopes (or Gyros) are used to measure angular velocity, and changes in inertial attitude may be estimated by integrating the gyro output. The combination of a three-axis accelerometer and a three-axis gyroscope into a single package is frequently called an Inertial Measurement Unit (IMU).

There are four tiers of modern IMU performance: Marine and Navigation, Tactical, Industrial, and Automotive & Consumer Grade [1]. The spectrum of prices between these grades can be anywhere from a few cents to hundreds of thousands of dollars. The performance often matches the price. For that reason, most of the demanding applications, such as space exploration, require the highest grade IMUs. These high grade IMUs offer the best performance and reliability, however they are bulky (volume of roughly $216in^2$)and they are costly (roughly $100,000) [1]. These characteristics of such high performance sensors make them non-feasible for low-budget, low-volume missions such as 1U CubeSats (10cm x 10cm x 10cm), 2U CubeSats (10cm x 10cm x 20cm), or 3U CubeSats (10cm x 10cm x 30cm). Meaning that, these missions must accept downgraded navigational performance because the only feasible sensor for these missions are MEMS IMUs. As an example, the drift-rate is about 1°/ min for MEMS gyros and about 1°/ year for ring laser gyros [2]. Thus, in the present time, single MEMS sensors cannot match the performance of a high grade IMU. To address this deficiency, recent studies have shown that a cluster of MEMS IMUs can deliver increased performance greater than any one MEMS IMU may deliver.[3, 4, 5, 6]

1.2 What is a MEMS IMU Cluster?

A MEMS IMU consists of miniaturized gyroscope and accelerometer to provide acceleration and rotation rate data to the user. As can be observed from Fig. 1-a from an STmicroelectronics LPY410AL analog MEMS gyroscope, these MEMS sensors are about the size of an Integrated Circuit (IC) chip with size of 4mm x 5mm x 1mm and mass of about 1 gram. Within the same figure, an ADXL335 MEMS accelerometer has a size of 4mm x 4mm x 1.45mm and has a mass of about 1 gram.



www.st.com/web/catalog/sense_power/FM89/SC1288/PF248616

(a) APY410AL MEMS Gyroscope

www.analog.com/en/products/mems/mems-accelerometers/adxl335.html

(b) ADXL335 MEMS Accelerometer

Figure 1: MEMS IMU Sensor Examples

The novel idea addressed in [3, 4, 5, 6], was to combine multiple MEMS sensors onto a single printed circuit board (PCB). This idea would theoretically increase the performance capabilities of a MEMS-based IMU system. As an example, Fig. 2 is chart showing how the performance of a cluster of MEMS units increases as the number of MEMS sensors is increased within the cluster. This chart was created assuming the simple averaging of the IMU outputs, which yields a performance improvement according to the following expression [3],

$$ARW_N = \frac{ARW_1}{\sqrt{N}} \tag{1}$$

where ARW_N is the angle random walk of N sensors in the cluster, ARW_1 is the angle random walk of a single MEMS unit, and N represents the number of MEMS units in the cluster. The smaller the angle random walk, the better the performance of the unit. In addition, Fig. 2 shows where this theoretical performance crosses the threshold into a tactical grade IMU. For this particular example, combining 19 MEMS gyroscopes that each have an angle random walk of $.3\frac{deg}{rt{-}hr}$ can yield performance that matches that of a typical tactical grade IMU. As MEMS sensors further mature, matching even higher-performing IMU grades (i.e. navigation grade) may someday be possible with this clustering technology.

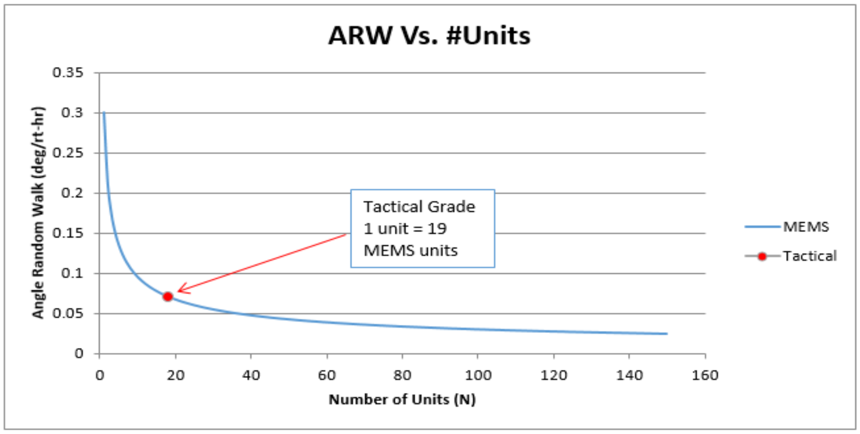
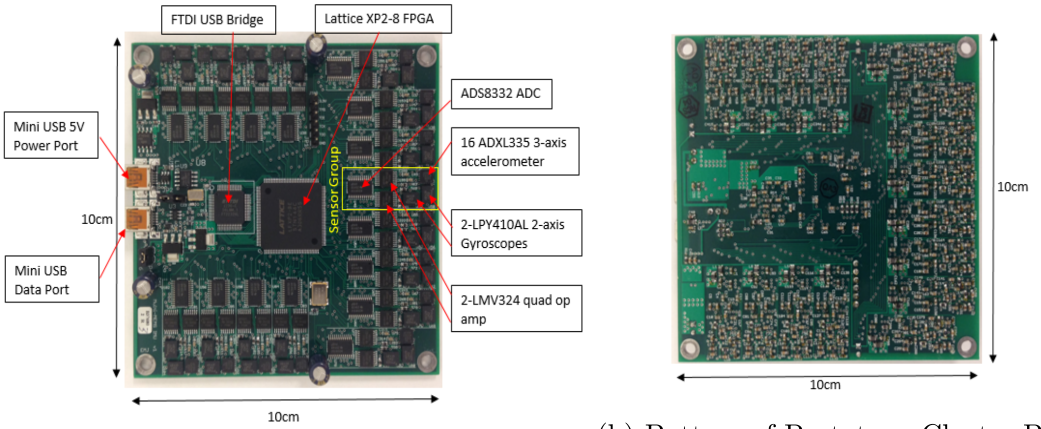


Figure 2: Theoretical ARW Vs. Number of MEMS sensors

This theoretical performance of MEMS IMU clusters is what lead researchers at West Virginia University and Marquette University to create a prototype cluster with 16+ MEMS gyroscopes and accelerometers. The first version of this systems used analog sensors. The board, which is shown in Fig. 3 a-b, is comprised of 16 ADXL335 3-axis accelerometers (Fig. 1-b), 32 LPY410AL 2-axis gyroscopes (Fig. 1-a), 16 LMV324 Op amps, 16 ADS8332 Analog to Digital Converters, 1 Lattice XP2-8 FPGA, and 1 FTDI USB Bridge. Since there are 32 2-axis gyroscopes, this will create 16 3-axis gyroscopes with one axis that has a redundant set of 16 measurements. There are 16 sensor groups on the board, one of the sensor groups is shown in yellow box in Fig. 3-a. Each sensor group is comprised of two 2-axis gyroscopes, one accelerometer, two op-amps, and one analog to digital converter. Within each sensor group, the analog to digital converter converts all the analog information from the accelerometers and gyroscopes and sends that information to the FPGA. The FPGA simultaneously collects the data from all 16 sensor groups and sends that data to an FTDI to USB bridge which converts all the FTDI digital data to USB digital data for ease of use. The USB data is then sent out through a mini USB connection and can be collected with a number of data acquisition devices. The output from this IMU cluster is comprised of; 16 x-axis accelerometer measurements, 16 y-axis accelerometer measurements, 16 z-axis accelerometer measurements, 16 x-axis gyroscope measurements, 16 y-axis measurements, and 32 z-axis measurements per time step (112 digital measurements per time instance). These 112 digital measurements are collected by a Raspberry Pi microcomputer (Fig. 4), which sends the data via TCPIP to a data acquisition computer. The data acquisition computer collects the measurements via Matlab and the measurements are currently post-processed. However, once the technology is further advanced, the measurements will be processed in real-time.



(a) Top of Prototype Cluster Board

(b) Bottom of Prototype Cluster Board

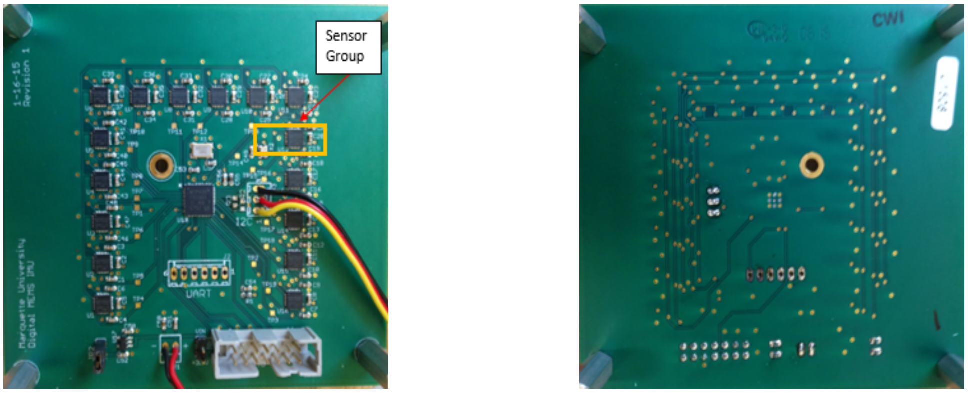
Figure 3: Layout of Prototype MEMS IMU Cluster Board

Observing Fig. 3, it is no secret that the analog board is software and hardware complex. This complexity leads to troubleshooting issues, a lack of board space, increased overall power, and most importantly high cost. After testing the analog board on a 3-axis rate table at Johnson Space Center, a fatal board flaw was discovered in which the output from the accelerometers was leaking into the gyroscope output via the analog to digital converter (ADC) and causing faulty measurements. Due to the complex design of the analog IMU sensor package, there was no easy fix

www.tenettech.com/product/2184/tenet-technetronicsraspberry-pi

Figure 4: Raspberry Pi

to the problem. As a result, this issues is what led to a redesigned IMU cluster with all digital sensors (Fig. 5). The advantages of this board are: the need for much fewer components (Digital=87, Analog=689), lower cost (Digital=$620/board, Analog=$1720/board), lower hardware complexity leading to easier troubleshooting, less overall sensor space on the board (Digital=44cm^2, Analog=100cm^2), and decreased total power (Digital=220mW, Analog=2.9W). One of the only advantages of the analog board is the ease of data time-tagging. The analog sensors could be sampled simultaneously, while the digital sensors must be sampled sequentially. Due to the flaw in the analog board, all results presented in this thesis are the product of the digital IMU cluster.

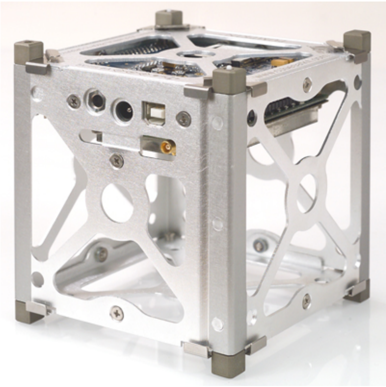


(a) Digital IMU Top View

(b) Digital IMU Bottom View

Figure 5: Digital IMU Cluster Layout

In addition to the overall layout of both boards, another important aspect of such designs is there overall dimension. For motivation, such sensor packages may make high performance inertial navigation possible for for small satellite applications (1U, 2U, and 3U CubeSats). Seen in Fig. 6 is 1U CubeSat chassis from Pumpkin CubeSats which has outer dimensions of 10cm x 10cm x 10cm. Purposefully, both analog and digital IMU clusters (Fig. 3 & Fig. 5) were designed to fit in such a chassis.



www.cubesatkit.com

Figure 6: 1U Pumpkin CubeSat Chassis

2 Thesis Overview

Although MEMS IMU clusters theoretically provide better performance than any one MEMS sensor alone, they must first be proven through rigorous testing and calibration practices before they can be used in practice. The work performed throughout this thesis aims to help make working with such a system much easier. Since an IMU measurement is only as good as its calibration, a major part of the work was the development of a new calibration technique to accurately and quickly estimate the systematic errors associated with a raw gyroscope measurement. Removing these systematic errors (e.g. biases, misalignments, etc.) is critical to achieving better performance.

The calibration device is also very important. Therefore, with the help of an

Aerotech ARMS-200 single-axis rate table, a WVU calibration apparatus (for which a provisional patent application has been filed) was designed so that three-axis measurements could be acquired using only a single-axis rate table. This apparatus allows the user to position the IMU cluster along all three axes into 576 unique, repeatable positions. Many of these positions are essential to the chosen calibration technique.

Lastly, as a final portion to this thesis, a Fault Detection, Isolation, and Recovery (FDIR) software architecture was designed to accurately determine bad measurements from good measurements. This architecture is essential to the IMU cluster so that bad measurements from any one IMU do not affect the overall performance. This thesis is organized as follows. An alignment technique for IMU clusters was designed, simulated and analyzed (Section 3). A survey of current calibration techniques is presented along with the design of our new calibration apparatus (Section 4). A full calibration technique using the digital IMU cluster and the new WVU calibration apparatus is presented along with the results (Section 5). Preparation to fly the IMU cluster on a sounding rocket flight (Section 6). Lastly, a discussion and the construction of a FDIR technique to recognize faulty measurements in an IMU cluster (Section 7).

3 Development of an Alignment Technique for a Large Number of Redundant Inertial Measurement Units [7]

In this chapter, an idea for estimating misalignment errors is presented. Two techniques, the MLE and the Wahba method were chosen to provide this estimation after a brief survey of current constructs. The results output from each method are compared so that a single architecture may be defined to provide the error estimates for the IMU cluster. The resulting method would then help determine misalignment and the additional errors in further work. Definitions of the gyroscope measurement model and each of the associated errors may also be found this chapter.

3.1 IMU Measurement Model

To estimate the errors associated with the output measurement of an IMU, it is important to develop a measurement model. The measurement model, as for example seen in Fang, et al.[8], is:

$$\tilde{\mathbf{w}} = [\mathbf{I}_{3\times3} + \boldsymbol{\delta} + \boldsymbol{\Gamma} + \boldsymbol{\Xi}]\mathbf{w}_S + \boldsymbol{\beta} + \boldsymbol{\nu} \tag{2}$$

where $\tilde{\mathbf{w}}$ is the 3×1 measurement in the sensor frame, $\mathbf{I}_{3\times3}$ is a 3×3 identity matrix, $\boldsymbol{\delta}$ is a diagonal scale factor error matrix, $\boldsymbol{\Gamma}$ is the misalignment matrix (a skew-symmetric cross product matrix), $\boldsymbol{\Xi}$ is a strictly lower triangular non-orthogonality matrix, $\boldsymbol{\beta}$ is a 3×1 bias error vector, and $\boldsymbol{\nu}$ is a 3×1 vector representing zero-mean Gaussian noise. For simplicity of the present discussion, the non-orthogonality matrix ($\boldsymbol{\Xi}$) will be neglected. To represent the true measurement in the sensor frame, $\mathbf{w}_S$ is defined as:

$$\mathbf{w}_S = \mathbf{T}_S^I \mathbf{w}_I \tag{3}$$

where $\mathbf{w}_I$ is the 3×1 true measurement vector in the inertial frame and $\mathbf{T}_S^I$ is a 3×3 rotation matrix from the inertial frame to the sensor frame. Since the IMU cluster involves many IMUs, each of these IMUs will have it's own unique set of errors. It is then the task of the error estimation algorithm to estimate the errors produced inside each of the various IMU measurement models. These estimation techniques are to follow in the IMU cluster calibration technique section.

3.2 Explanation of Errors

As defined in the measurement model there are four errors associated with a raw IMU measurement. In order to determine these, it is first essential to understand what these errors actually mean. Therefore, through this subsection, a background into the dynamics and construction of each of the errors will be given.

To begin, the misalignment error is a crucial error in an IMU measurement because non-compensation of this error likely leads to measurements that deviate far from the truth. As a visual representation of misalignment, Fig. 7 shows an example IMU cluster with 9 IMUs.

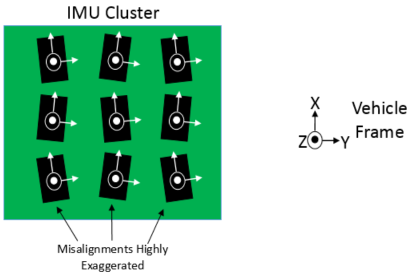
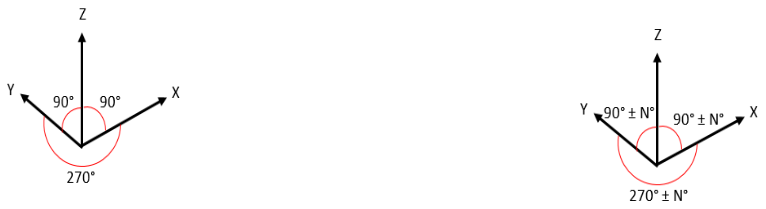


Figure 7: IMU Cluster Misalignment Example

In this figure, the axes of the IMUs (black objects in the figure) are not aligned with the axes of the vehicle frame. This 'misalignment' will cause skewed measurements from the cluster and therefore will lead to errors in actual versus measured rates. Misalignment is an angular error and is due to the manufacture of the board. The personnel or machine that solders the component to the printed circuit board is not aligned with the frame of the PCB and errors are introduced. There can also be misalignments introduced when the board is fixed to the vehicle body. Imperfect alignment with the PCB containing the cluster and the vehicle body frame will cause this misalignment. Figure 7 is an exaggerated version of what actually happens. To compensate for this error, the angle between the vehicle body axes and the IMU axes must be known which is found from a number of calibration techniques.

In addition to the misalignment, non-orthogonality is another angular error. Non-orthogonality is caused by the manufacture of the IMU itself. Orthogonality is defined as vectors or axes that are exactly 90° from one another. Within a 3-axis IMU this means the angle between x-y, x-z, and y-z are all 90°. Shown in Fig. 8-a, is the ideal case for non-orthogonality, also known as orthogonality. However, manufacturing an IMU with perfectly orthogonal axes is only possible down to a finite degree. Therefore, Fig. 8-b, is the actual case for all IMUs, where ± N° is some finite degree. This imperfection in the construction of the axes causes an error between axes and must be compensated for by the calibration.



(a) Ideal Non-Orthogonality (Orthogonality) Between Axes

(b) True Non-Orthogonality Between Axes

Figure 8: Non-Orthogonality Between Axes in a 3-axis IMU

The final two errors are less inherent than misalignment or non-orthogonality. The first, scale factor, is caused by conversion error from the units of which the IMU outputs (in this case digital numbers) to the useable engineering units (deg/sec or rad/sec) for gyroscopes. Although these conversions are usually very close, temperature, and rates can vary the scale factor ever so slightly. To add, gyroscope and accelerometer measurements are inherently non-linear and using a linear conversion will result in high errors. Therefore, finding a conversion equation or conversion factor that accurately portrays this non-linearity is key to having small scale factor errors. However even with the best non-linear fit equation, there is still error that must be recouped in the calibration.

Bias error or systematic error is the last measurable error included. A measurement bias is considered any offset from zero when the IMU is at rest. Ideally this value would be zero, however in nature all measurements posses some bias. In an IMU gyroscope measurement the bias could be nearly zero or it may be plus or minus a few deg/sec. Either way, in order to accurately reveal the truth, this error must also be discovered in the estimation to accurately calibrate the IMU cluster.

To conclude, because there are 16 gyroscopes in the IMU cluster, estimation of errors must be redundantly completed. These errors can change between power cycles of the cluster, which adds to the significance of having an efficient, practical method for calibration.

3.3 IMU Cluster Calibration Technique

Coupled with the output measurements of IMUs are several errors that corrupt the measurements and render them less useful. Some of the associated errors include scale factor, misalignment, bias, and readout errors as discussed above. The noisy IMU measurements contain the actual body accelerations and angular velocities. In order to extract this true measurement, the error terms must be accurately estimated and removed from the IMU output. Because we have a cluster of MEMS IMUs, the errors must be accurately estimated for each sensor. Estimations that inaccurately estimate the error sources can cause the estimated measurements to deviate far from the actual parameters of the vehicle. A build up of these inaccurate estimations over time can lead to a larger gap, which re-establishes the need for an accurate estimation algorithm. As a simplified version of these error estimation algorithms, refer to the flow chart in Fig. 9.

In previous work, the dynamic six-position method[9] was used as the calibration technique for the IMU cluster project[4] that motivated this research. This approach is relatively straight forward, however this technique relies on excellent sensor alignment (which can be quite time consuming) since it cannot estimate the misalignment term

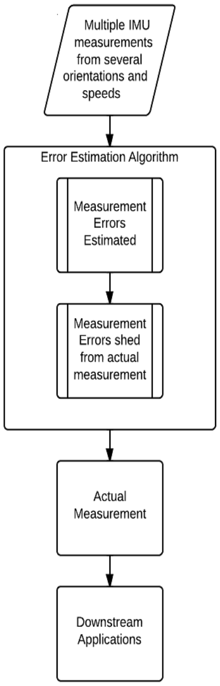

Multiple IMU measurements from several orientations and speeds
Error Estimation Algorithm
Measurement Errors Estimated
Measurement Errors shed from actual measurement
Actual Measurement
Downstream Applications

Figure 9: Simple Error Estimation Flow Chart.

or the non-orthogonality of the IMUs. Due to the disadvantages of this calibration approach, the author of this paper set out to determine a more appropriate method for this application.

3.4 Error Estimation Techniques

Several techniques have been developed to estimate the errors associated with the measurement model of an IMU. Each of these techniques offer their own unique set of advantages. Described below are several existing alignment techniques currently in use for calibration of IMUs.

3.4.1 Static Six-Position Method

Although a relatively simple technique, the six position method can be quite accurate with the correct equipment and alignment of the system. Aggarwal, et al.,[9] states that an error calibration may be performed by placing the IMU in alternate up and down orientations along each axis. Using the up and down measurements along each axis while the IMU is statically positioned on a leveled surface, the bias (β) and scale factor error (δ) may be estimated. To do this, a measurement model must be defined for each up and down measurement along each IMU axis,

$$m^{up} = \beta + (1 + \delta)e \tag{4}$$

$$m^{down} = \beta - (1 + \delta)e \tag{5}$$

where m^{up} is a measurement from the IMU when an axis is pointed upwards, m^{down} is a measurement from the IMU when that same axis is pointed downward, and e is the component of Earth's rotation rate along the corresponding axis for gyroscopes and the corresponding gravity magnitude for accelerometer measurements. Equations 4 & 5 may be arranged to solve for the bias and scale factor errors, as

$$\beta = (m^{up} + m^{down})/2 \tag{6}$$

$$\delta = (m^{up} - m^{down} - 2e)/2e \tag{7}$$

Once the errors have been estimated for the first axis of interest, the procedure may be repeated for the other axes. The advantage of this method for error calibration is its simplicity. Although a leveled surface is needed, no specialized equipment (i.e. rate table) is required. The disadvantages to this method are mostly linked to it's simplicity. However, the accuracy of this technique is dependent on how level one can make the surface on which the IMU is positioned. If the surface is not appropriately leveled, the axis of the IMU and the axis of the Earth may be slightly misaligned, causing errors in the actual measurements. In addition, with this method, only one axis may be calibrated at a time. Therefore, this technique would have to be repeated multiple times. Additionally, for the case of gyroscopes, the input Earth rate signal e is very small compared to the amount of noise on typical MEMS sensors. This low signal-to-noise ratio (SNR) makes the estimate of δ relatively poor compared to some of the methods introduced later.

Lastly, this technique can only estimate bias and scale factor errors, therefore another technique must be implemented to estimate the misalignment error. Due to these disadvantages, the static six position method is less useful for the IMU cluster application.

3.4.2 Dynamic Six-Position Method

Similar to the static six position method, the dynamic six-position method[9] involves measurements using a single-axis rate table. The measurement model and procedure used to find the errors are the exact same as described in the previous method however, m^{up} is replaced with m^{CW} to represent clockwise rotation measurements of the IMU and m^{down} is replaced with m^{CCW} to represent counter-clockwise rotation measurements from the IMU. The advantage of using this method is that there is a known rate from the rate table, which can be referenced and used to provide a more accurate estimation of the scale factor error. Also, when taking clockwise and counter clockwise measurements, the IMU does not physically have to change position or be flipped as long as the rate table can produce the clockwise and counterclockwise rotations. Otherwise, the advantages and disadvantages of this technique are similar to the static six-positions method.

3.4.3 Multi-Position Calibration Method

Syed, et al.,[10] developed a technique for in field calibration of IMUs using several pre-planned orientations of the IMU sensor. This method uses Earth's rotation and the local gravity to create the references needed for calibration. Unlike the two methods prior to this, this technique does not need to be aligned with the local level frame because of these references. In this approach, a redundant number of IMU rotations is developed and the IMU errors are estimated using a least square (LS) adjustment. For the redundant number of IMU rotations, the authors of [10] define a set of 18 orientations and offer a rotation scheme to help calibrate the IMU. This method performs better than the six position method because there are more measurements to include into the estimation. In addition, an orientation rotation scheme is a useful tool, especially during in-field calibration. However, the Earth's rotation rate is a poor signal as a gyroscope reference for the calibration, which can lead to poor estimates for scale factor and non-orthogonality errors. To combat this problem, a modified multi-position calibration method is introduced that uses a rate table as the rotation reference. Because of this calibration technique's weak signal reference (which, does not work well with MEMS technology), the approach was not considered for the IMU cluster application.

3.4.4 Wahba's Method

Solutions to Wahba's problem[11] are currently in widespread use for attitude estimation by the aerospace community. Because of the characteristics, it can also be used to find the rotational errors (i.e. misalignment) of an IMU. This technique can be used to find the misalignment of an IMU to the sensor frame, or to find the misalignment between two IMUs in a cluster. The solution to Wahba's problem yields the optimal transformation describing the misalignment. Wahba's problem will seek to minimize the cost function:

$$J(\mathbf{T}_{S_j}^{B}) = 1/2 \sum_{i=1}^{n} w_i ||\tilde{\mathbf{u}}_{i_{S_j}} - \mathbf{T}_{S_j}^{B} \mathbf{u}_{i_I}||^2 \tag{8}$$

where w_i is a weighting term for the i-th measurement, $\tilde{\mathbf{u}}_{i_{S_j}}$ is the 3×1 measurement of the body rate from the sensor, and $\mathbf{u}_{i_I}$ is the 3×1 vector of the actual body rate. Since all sensors are the same on the IMU cluster, the weighting term is the same for each IMU. The matrix, $\mathbf{T}_{S_j}^{B}$, is the rotation matrix that transforms the angular velocity from the ideal sensor frame to the j-th IMU's sensor frame.

There are many ways to solve for the rotation matrix $\mathbf{T}_{S_j}^{I}$ such as Davenport's q-method[12], Quarternion Estimator(QUEST)[13], and Singular Value Decomposition (SVD)[14]. The SVD method is used here. In order to solve for $\mathbf{T}_{S_j}^{B}$ using the SVD, the attitude profile matrix $\mathbf{B}$ must first be formed, as

$$\mathbf{B} = \sum_{i=1}^{n} \frac{1}{\sigma^2} \tilde{\mathbf{u}}_{i_{S_j}} \mathbf{u}_{i_I}^{T} \tag{9}$$

The SVD of $\mathbf{B}$ is then found, so that:

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^{T} \tag{10}$$

The optimal rotation matrix may then be found using:

$$\mathbf{T}_{S_j}^{B} = \mathbf{U}\mathbf{M}\mathbf{V}^{T} \tag{11}$$

where,

$$\mathbf{M} = diag\{[1 \quad 1 \quad det(\mathbf{U})det(\mathbf{V})]\} \tag{12}$$

The advantage of this method is that the rotation matrix is relatively easy to compute. However, this algorithm can only find rotational errors, such as misalignment. As can be seen in the measurement model, IMU measurements suffer from other error terms that must also be estimated in order to reveal the true value. Therefore, an additional estimator/method must be used to determine the bias and scale factor error.

3.4.5 Maximum Likelihood Estimate (MLE)

The MLE[15, 16, 17] algorithm is widely used in many applications due to its use of the measurement model to create a state equation. By using this state equation this method can estimate multiple error sources (e.g. bias, misalignment and scale factor) providing they are all observable. Because of these algorithmic advantages, this approach was selected as the focus of this paper. In order to begin this approach, it is crucial to understand the measurement from the system to be estimated. Since this is a redundant system, there are redundant measurements along each axis, which the MLE and all other methods must be designed to accommodate to create an efficient and complete estimation. The optimal solution to this batch estimation problem is given by,

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\tilde{\mathbf{w}} - \hat{\mathbf{w}}) \tag{13}$$

where $\hat{\mathbf{x}}$ represents the $n \times 1$ state vector, $\mathbf{H}$ is the $3 \times n$ measurement sensitivity matrix, $\tilde{\mathbf{w}}$ is the 3×1 measurement seen from the sensor, and $\hat{\mathbf{w}}$ is the expected measurement. Assuming errors on each axis are uncorrelated, the measurement covariance, $\mathbf{R}$, is computed as:

$$\mathbf{R} = \sigma^2\mathbf{I}_{3\times3} \tag{14}$$

where σ is the sensor measurement standard deviation. Once each of these parameters is defined, a solution to the MLE may be found from Eq. 13. The primary advantage of this approach is that all simultaneously observable error sources may be estimated. Because of this, this method was chosen to provide the error estimation for the present IMU cluster application. A further explanation of how such a technique was implemented for this application is described below along with a comparison of the performance to other approaches.

3.4.6 Summary of Estimation Techniques

As a comparison between each of the methods described in this this section, (see Table 1), which shows the errors that each technique is capable of estimating within an IMU measurement.

Estimation Technique	Scale Factor	Misalignment	Bias	Non-Orthogonality
Six-Position Static	✓		✓	
Six-Position Dynamic	✓		✓	
Multi-Position	✓		✓	✓
Wahba's Problem		✓		
MLE	✓	✓	✓	✓

Table 1: Technique Capability Matrix

3.5 Monte Carlo Cluster Simulation

A Monte Carlo simulation was constructed to verify the estimation methods chosen for the IMU cluster application. The two methods of interest are the Wahba Problem method and the MLE approach. In order to compare the performance of these methods, misalignment was the only error placed into the measurement model, since the Wahba Problem can only estimate rotational errors. To begin the construction of the simulation, a measurement model was defined using only the misalignment error (simplification of Eq. 2),

$$\tilde{\mathbf{w}} = [\mathbf{I}_{3\times 3} + \mathbf{\Gamma}]\mathbf{w}_S + \boldsymbol{\nu} \tag{15}$$

The ideal measurement vector $\mathbf{w}_S$ is constructed using Eq. 3. Because the author was using a single-axis rate table to complete the experiment, only the Z-axis of the table will spin in the inertial frame. Thus, $\mathbf{w}_I$ is constructed as follows,

$$\mathbf{w}_I = w_{RT} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \mathbf{w}_e \tag{16}$$

where w_{RT} is the scalar value of the rate table speed and $\mathbf{w}_e$ is a 3×1 vector of Earth's rotation rate.

Since the IMU cluster may be oriented differently, a rotation matrix ($\mathbf{T}_S^I$) must be constructed that will transfer the measurements from the inertial frame ($\mathbf{w}_I$) into the sensor frame ($\mathbf{w}_S$). To construct the rotation matrix it is first necessary to describe the custom made articulating bracket designed for the rotation rate tests (this Monte Carlo is designed to simulate actual measurements within these tests). This articulating bracket allows the IMU cluster to be placed into different positions so that measurements may be taken at different orientations. Figure 10 is a representation of how the articulating bracket works.

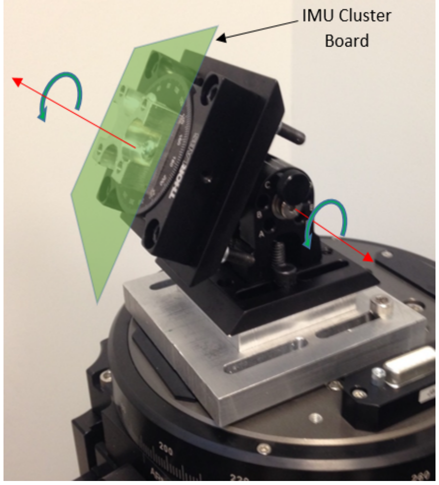


Figure 10: Custom Bracket for Table Testing

When the IMU is oriented flat/level on the articulating bracket, measurements can be taken along the IMU's Z-axis. When the articulating bracket is rotated so that the plane of the IMU and the plane of the rate table are no longer parallel, then measurements can be taken along the Y-axis. Finally, if the top rotation mount is rotated by any angle while the IMU cluster board is non-parallel with the table, then measurements may be taken along the X-axis. In addition, the IMU may be oriented in a position that is a combination of each of these rotations so that measurements may be taken along each axis simultaneously. The simulation was created to mimic measurements that one might see from this bracket while mounted on a single-axis rate table.

Given the construction of this bracket a 3-1-3 Euler angle sequence may be used to create a rotation matrix $\mathbf{T}_S^I$. This rotation matrix is given by,

$$\mathbf{T}_S^I = \mathbf{D}_\varphi \mathbf{D}_\vartheta \mathbf{D}_\phi \tag{17}$$

where,

$$\mathbf{D}_\varphi = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$\mathbf{D}_\vartheta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\vartheta & \sin\vartheta \\ 0 & -\sin\vartheta & \cos\vartheta \end{bmatrix} \tag{19}$$

$$\mathbf{D}_\phi = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

and ϕ is the angle of the rate table while the table is spinning (time is factored into the angle), ϑ is the angle of the articulating bracket, and φ is the angle of the manual rotation stage mounted on the top of the articulating bracket. Oriented measurements in the sensor frame may then be created by,

$$\mathbf{w}_S = \mathbf{T}_S^I \mathbf{w}_I \tag{21}$$

After $\mathbf{w}_S$ has been created, the simulated measurement may be constructed as,

$$\tilde{\mathbf{w}} = [\mathbf{I}_{3\times3} + \mathbf{\Gamma}]\mathbf{w}_S + \boldsymbol{\nu} \tag{22}$$

This process may be repeated several times to create the multiple measurements within the IMU cluster. In order to estimate the error source(s) using the methods described above, many orientations and rotation rate speeds must be constructed. Estimating errors using measurements at only one rate and orientation can create a poor estimate and may not be possible with these methods. For this particular application, four different table rates and four different orientations of the IMU were chosen, as illustrated in Table 2 and 3.

Rate 1	Rate 2	Rate 3	Rate 4
$90°/s$	$60°/s$	$45°/s$	$30°/s$

Table 2: Simulated IMU Cluster Rates

Orientation	φ	ϑ	ϕ
1	0°	0°	0°
2	0°	90°	0°
3	0°	45°	0°
4	45°	45°	0°

Table 3: Simulated IMU Cluster Orientations

Since there are 4 table rates and 4 orientations per run, there will be 16 unique 3-axis measurements per IMU per run. Whichever method is chosen for the error estimation, will have 16 unique 3-axis angular velocities per IMU to work with in order to estimate the misalignment errors. The misalignment input vector was held constant per each run of the 10,000 run Monte Carlo. In the interest of comparison, results from one of the many IMUs on the cluster board is shown in the subsequent section. The program can easily be switched to incorporate all IMUs, but all comparison data between each method that is collected from the additional IMUs would be redundant.

3.6 Cluster Estimation Techniques

Two methods were constructed to provide the alignment estimation for the IMU cluster application. These methods are the MLE approach and the Wahba method. An approach to constructing and solving for these techniques is given in relation to solving or estimating the misalignment error in an IMU cluster application. Results and a comparison of the methods are provided in the sections following.

3.6.1 MLE Solution

The output from the Monte Carlo simulation consists of 16 unique 3-axis body rate measurements that were constructed by using 4 different orientations of the IMU swarm and 4 different rates from the simulated rate table. These rates have a misalignment error and noise term included in their measurement,

$$\tilde{\mathbf{y}}_i = [\mathbf{I}_{3\times3} + \mathbf{\Gamma}]\mathbf{w}_{S_i} + \boldsymbol{\nu} \tag{23}$$

where $\tilde{\mathbf{y}}_i$ is the 3×1 body rate measurement from the simulation, $\mathbf{I}_{3\times3}$ is a 3×3 identity matrix, $\mathbf{w}_{S_i}$ is the actual body rate in the sensor frame, and $\boldsymbol{\nu}$ is zero-mean Gaussian noise. Additionally, $\mathbf{\Gamma}$ is a skew symmetric cross product matrix containing the simulated misalignment vector, given by

$$\mathbf{\Gamma} = [\boldsymbol{\gamma}\times] = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix} \tag{24}$$

So that the measurement equation may be rewritten as

$$\tilde{\mathbf{y}} = \big[\mathbf{I}_{3\times3} - [\mathbf{w}_S\times]\big]\boldsymbol{\gamma} + \boldsymbol{\nu} \tag{25}$$

Therefore, the **H** matrix may be constructed, as

$$\mathbf{H}_i = \frac{\partial y_i}{\partial x} = \frac{\partial y_i}{\partial \gamma} = [-\mathbf{w}_{S_i}\times] \tag{26}$$

Once the **H** matrix is constructed and assuming the sequential measurement errors are uncorrelated ($E[\boldsymbol{\nu}_i\boldsymbol{\nu}_j^T] = 0$), the MLE solution for $\hat{\boldsymbol{\gamma}}$ is given by,

$$\hat{\boldsymbol{\gamma}} = \left(\sum_{i=1}^{n}\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\right)^{-1}\sum_{i=1}^{n}\mathbf{H}_i^T\mathbf{R}_i^{-1}(\tilde{\mathbf{w}}_i - \hat{\mathbf{w}}_{S_i}) \tag{27}$$

where $\hat{\boldsymbol{\gamma}}$ is a 3×1 angle vector describing the estimated IMU misalignment. This estimated misalignment vector ($\hat{\boldsymbol{\gamma}}$) is used to form the rotation matrix ($\mathbf{T}_{MLE}$) via a transformation algorithm. This rotation matrix is then compared to the actual rotation matrix placed into the simulation ($\mathbf{T}_{Actual}$) by a rotation error matrix ($\delta\mathbf{T}_{MLE}$):

$$\delta\mathbf{T}_{MLE} = (\mathbf{T}_{MLE})(\mathbf{T}_{Actual})^{-1} \tag{28}$$

The matrix $\delta\mathbf{T}_{MLE}$ can then be transformed into an angle error vector via a transformation algorithm. This angle error vector is computed and stored for each of the 10,000 Monte Carlo runs.

3.6.2 Wahba's Solution Method

An additional method to solving this alignment problem is by using Wahba's problem and SVD. Like the MLE approach, this method will only be able to use 16 unique 3-axis measurements to compute the solution for misalignment. The technique begins much like the MLE approach by entering a loop designed to compute the attitude profile matrix (**B**) in Eq. 9 for each of the 16 3-axis measurements. As a result of the loop an additive **B** matrix is created. It is possible to create an additive matrix like this, because the misalignment does not change through each run of the Monte Carlo. Next, **U**, **S**, and **V** are found through the SVD of **B**. The optimal rotation is then found as in Eq. 11,

$$\mathbf{T}_{wahba} = \mathbf{UMV}^T \tag{29}$$

This rotation matrix may then be compared to the actual rotation matrix ($\mathbf{T}_{actual}$) formed from the true misalignment vector placed into the simulation by:

$$\delta\mathbf{T}_{wahba} = (\mathbf{T}_{wahba})(\mathbf{T}_{actual})^{-1} \tag{30}$$

$\delta\mathbf{T}_{wahba}$ can then be transformed into an angle error vector via a transformation algorithm. As before, results are kept for all 10,000 Monte Carlo runs.

3.7 Misalignment Estimation Comparison Results

A 10,000 run Monte Carlo was performed to verify each of the methods of interest for error estimation of the IMU cluster application. The error from these measurements (i.e. misalignment) was estimated using Maximum Likelihood Estimation and Wahba's method. The measurement standard deviation was assumed to be 1.43 deg/sec. The results are plotted in Fig. 11 and Fig. 12 .

In Fig. 11, the estimation errors from the MLE are plotted in separate histograms. The performance characteristics of the MLE method are presented in Table 4. It is noted that the mean of each histogram in Fig. 3 is nearly zero.

Similarly, in Fig. 12, the estimation errors from the Wahba problem approach are plotted in separate histograms. When observing Table 4, it can be noted that the mean is slightly worse than the MLE solution along each axis. This difference is most

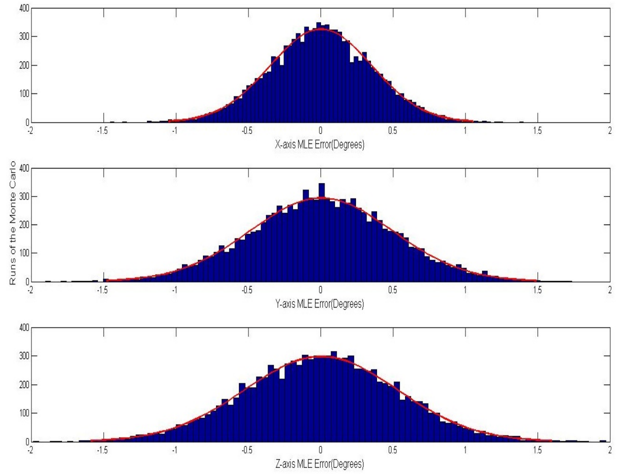


Figure 11: MLE Error Histogram

likely due to precision issues in the SVD. In other simulations not shown here, the Wahba solution outperforms the MLE solution, in terms of mean and standard deviation, when the misalignment and noise vectors are increased dramatically. However, the author believes that this state (misalignments $> 20°$ and sigma > 57.30 deg/sec) does not reflect the conditions of the real-world IMU cluster.

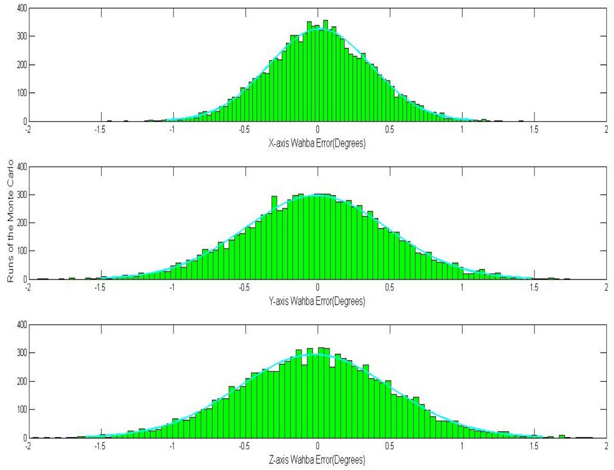


Figure 12: Wahba Error Histogram

Performance Parameter	**MLE**	**Wahba**
Mean X	0.0030°	0.0149°
Mean Y	0.0054°	−0.0102°
Mean Z	0.0038°	−0.0225°
Std. Dev. X	0.3506°	0.3535°
Std. Dev. Y	0.4951°	0.4966°
Std. Dev. Z	0.5314°	0.5270°

Table 4: Implemented Method Comparison Table

4 A Survey of Present Calibration Apparatuses and Techniques

Complete calibration of three-axis IMUs involves estimating the systematic errors associated with an IMU measurement which includes misalignment, bias, non-orthogonality, and scale factor errors. Some of these quantities may be temperature dependent, but we will not be considering that effect here. IMU calibration is usually completed by spinning the IMU on a precision rate table and using analytic methods to extract the calibration parameters. The number of axes within the sensor determines how in depth the calibration must be. For instance, calibrating one single axis IMU only involves calibrating about 1 axis and therefore does not include non-orthogonality. However, when the sensor or system consists of multiple axes a specialized calibration must be performed. For the IMU cluster application, since there are three axes, using a single axis rate table to complete a calibration is not ideal because there are less axes spinning then the IMU can sense. To combat this disadvantage, traditional methods of calibration either 1.) Use specialized equipment (i.e. two axis and three axis rate tables) which allows all axis to spin at different rates simultaneously or 2.) Use precision machined angled blocks affixed to a single axis rate table so that the IMU may be oriented such that all axes receive a portion of the rate commanded by the table. These methods come at a drawback of cost and/or repeatability however. As a result, a new apparatus was invented by both the Author (Drew E. Bittner) and his Advisor (John A. Christian) that would allow tri-axis calibration using only a single axis table. This apparatus is described in further detail in the sections following.

4.1 Current Apparatuses & Calibration Techniques

There are a number of techniques used to calibrate IMUs. Discussed in this section are current technologies addressing different methods to which that task may be performed. In addition, the advantages and disadvantages of each method are discussed to help further invoke the necessity of the patent-pending apparatus.

4.1.1 Six Position Calibration With Manual Maneuvering

Aggarwal et al. first proposed the six position method [9], which can be performed in a static or dynamic environment. However, since Earth's rotation rate is a relatively weak signal in the static calibration, it is suggested that the dynamic calibration be used with a rate table. Due to this techniques simplicity and ease of use, it was chosen to provide the calibration for the IMU cluster application [4]. To help verbally describe this method, first, a 3-axis IMU is placed flat on the rate table with one of its axes pointing up or down. Next, The IMU is spun clockwise (measurements collected) and spun counter-clockwise (measurements collected). The IMU is then flipped so that the axis that was pointing up is pointing down and measurements are collected in the same fashion by spinning the rate table clockwise and counter-clockwise. This is completed for each of the 3 axes which makes six positions (2-positions per axis). Using the measurements at each of these positions one may easily find the calibration errors using the equations described in Aggarwal et al [9](Eq. 6 & Eq. 7). As a graphical representation of the method refer to Fig. 13 where the cluster board is shown in the 6 unique positions on top of a single-axis rate table.

This relatively straight-forward procedure makes it appealing to those who are looking to do a quick calibration of their IMU or system. However, because the IMU is always oriented with it's axis parallel to the rotating axis of the table, a disadvantage to this method is that non-orthogonality term cannot be determined for two or three axis sensors. Therefore, other methods must be used in conjunction to obtain a complete calibration. Secondly, the IMU must be manually maneuvered to each position. Meaning that the IMU must be physically handled during the calibration and it must be constantly fixed and unfixed from the rate table. When the IMU is re-fixed to the table (i.e. during a position change), the IMU may no longer be oriented exactly as it was in the previous positions leading to inaccuracies in the calibration. These inaccuracies cause issues in precise calibration which make it difficult to use this method.

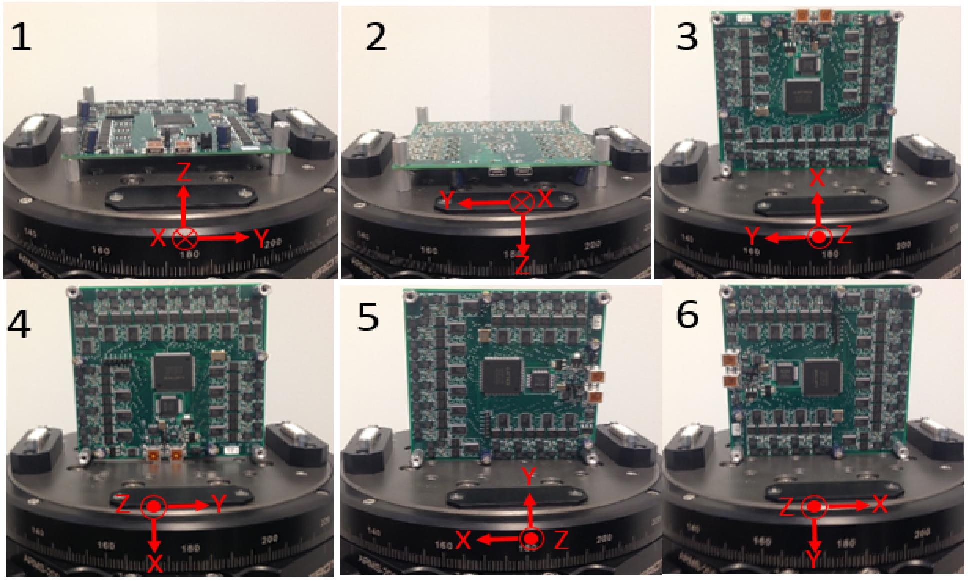


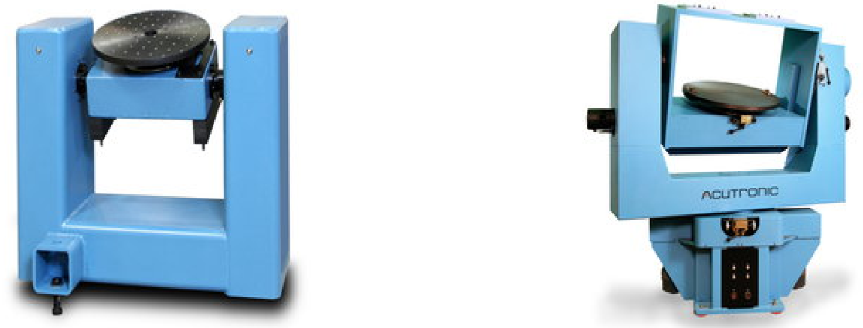
Figure 13: Six Position Calibration Technique

4.1.2 Multi-Axis Table Calibration

Another common way of calibrating multi-axis IMUs is with a multi-axis rate table (i.e. 2 and 3 axis rate tables). With rate tables like the ones shown in Fig. 14, a

2 or 3-axis IMU may be spun simultaneously on all axes. Therefore, errors between axes (i.e. non-orthogonality) may be retrieved. One advantage of such a system is that the axes of the rate table may be controlled separately and therefore different rates may be executed on separate axes of the IMU. Additionally, an IMU may be placed in any 3-axis orientation with a 3-axis rate table, where with the other methods described this would be impossible. To correctly calibrate a multi-axis IMU and to verify its performance a multi-axis rate table is the current standard to perform these duties.

Although this is the current standard for precision calibration, there are disadvantages to this technique. For small labs and small businesses looking for precise calibration a multi-axis table is often too expensive to consider. A 3-axis rate table system can cost over $100,000, which is not feasible for someone looking to calibrate a MEMS sensor a few times. Therefore, to perform a precise calibration someone in the situation must seek outside help or suffer estimation precision by using a less accurate calibration method. Another disadvantage of such a system is overall volumetric space. A 3-axis table can consume nearly $2m^2$ of lab space, which is unreasonable for most universities and businesses. Due to these disadvantages obtaining a multi-axis table was inconceivable for calibration of the MEMS IMU cluster.



www.acutronic.com/us/products/detail/view/
2-axis-motion-simulator-series-ac217

(a) 2-axis Rate Table

www.acutronic.com/us/products/3-axis/product-detail/view/
3-axis-motion-simulator-series-bd357

(b) 3-axis Rate Table

Figure 14: Two and Three Axis Rate Table Examples

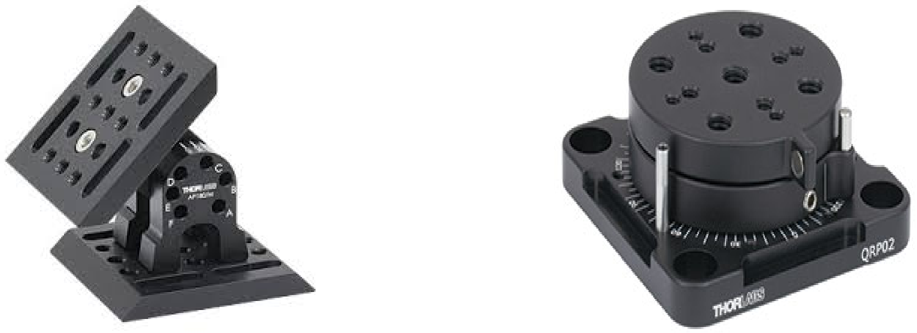
4.1.3 Angled Block Calibration

Similar to the six-position method, the calibration method using angled blocks, similar to what is shown in [18, 19], can help achieve complete calibration. This method fixes a machined angled block (machined to a precision angle) on top of a single-axis table. A multi-axis IMU is then fixed to the angled block. Then by spinning the rate table simultaneous measurements may be taken along all axes of the IMU because the IMU is oriented in such a way where all or most of the axes share in the controlled rate of the table. As mentioned, this method is similar to the six position method because it still uses those six positions, however more positions are added due to the angled orientations from the machined angled blocks.

The disadvantages to this method are similar to the six position method because of the manual maneuvering of the sensor during the implementation of the technique. The sensor still must be fixed, unfixed, and then re-fixed to the table every time there is a change in position. This can cause repeatability problems in the calibration and there is also danger in handling the sensor during operation. Due to the potential repeatability issues this method was not chosen as a suitable method for the IMU cluster calibration even though it offers more advantages then the six position method described above.

4.1.4 WVU Designed Custom Bracket

To counteract the disadvantages of each of the techniques previously described, an apparatus was constructed out of two products commercially available from Thorlabs. In Fig. 15-a is an articulating bracket which can be positioned every 15 degrees for a total freedom of 180 degrees. Next, Fig. 15-b is a manual rotation stage which can be positioned-accurately every 30 degrees for 360 degrees. To construct this apparatus, the manual rotation stage is fixed to the top of the articulating bracket. The bottom of the articulating bracket is then fixed to a single axis rate table via coupling mounts. Then, the IMU cluster PCB is fixed to the top of the manual rotation stage via custom coupling mounts. The constructed bracket can be seen in Fig. 16. By changing angles on the articulating bracket, simultaneous measurements may be taken along two of the axes (Shown in Fig. 17 measurements may be taken along Z and Y axes). Then by additionally changing the angle of the rotation stage by which the IMU cluster is fixed, simultaneous measurements along all three axes may be obtained because all three axes will share in the rotation of the rate table (Shown in Fig. 18). Although arguably the best method so far due to cost, effectiveness, and ease of operation, this apparatus lacks the ability to completely flip a sensor upside down (a 180° rotation of the articulating bracket) without unmounting the sensor from the platform. This vertical flip is necessary to observe and compensate for g-sensitivity and it is therefore crucial to be able to take measurements parallel and anti-parallel to Earth's gravity field. Secondly, not having to unmount and remount the sensor or handle the sensor during calibration is of utmost importance for repeatability and sensor damage purposes, which is why this method was not chosen. Although this apparatus can not provide all necessary orientations it did aid in the understanding and design of it's successor.



www.thorlabs.com/newgrouppage9.cfm?objectgroup_ID=209

(a) Articulating Bracket

www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1859

(b) Manual Rotation Stage

Figure 15: Thorlabs Products used to Create First Calibration Apparatus

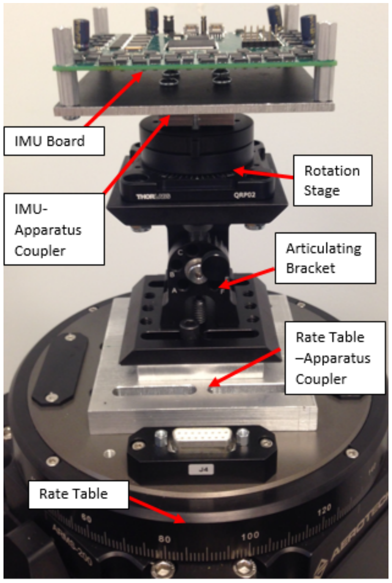


Figure 16: Constructed Apparatus

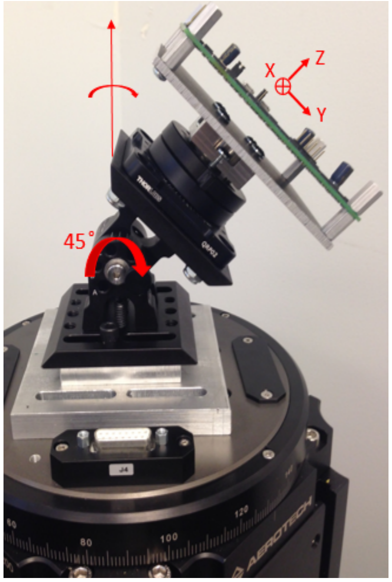


Figure 17: 45 Degree Tilt on Articulating Bracket

4.2 New WVU Calibration Apparatus for A Single Axis Rate Table

To address the disadvantages of each of the methods explained above, an apparatus was designed and constructed at WVU that will allow a user to easily obtain 3-axis measurements using only a single-axis rate table. The first apparatus (Fig. 16) came

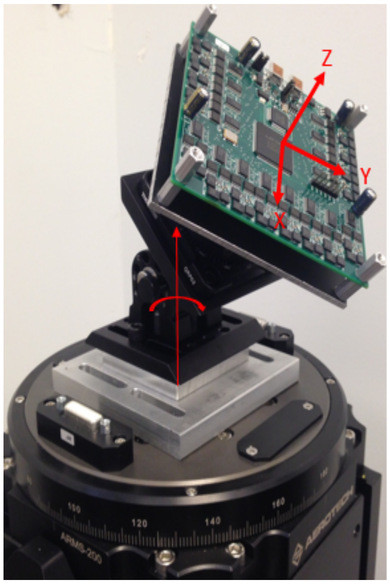


Figure 18: 45 Degree Tilt on Articulating Bracket w/ 45 Degree Rotation on Stage

close to achieving the orientation requirements needed for the calibration, however one position was impossible to obtain without unmounting the board from the apparatus. Therefore, the apparatus shown in a Computer Aided Drawing (CAD) in Fig. 19 was custom machined and a provisional patent has been filed. What follows is explanation of how the apparatus works.

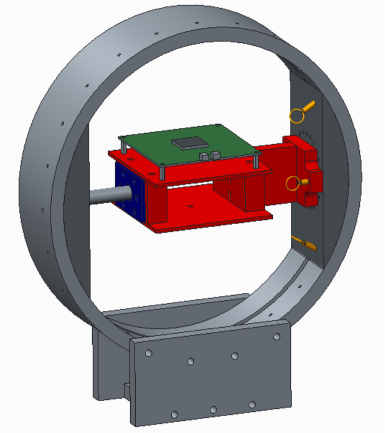


Figure 19: CAD Representation of Patent-Pending Calibration Apparatus

4.2.1 Explanation of Patent-Pending Calibration Apparatus

As is common for many designs, the inventors came up with a list of the necessary requirements that the apparatus must fulfill. The requirements are:

- Single axis rate table mountable
- Volumetric area not to exceed 1.5ft^2
- Overall apparatus weight not to exceed 20lb as to not affect rate table functionality
- Apparatus and single-axis table combination cost should be sufficiently lower than the cost of a multi-axis rate table
- Enable a sufficient number of 1,2,and 3-axis orientations
- Operational ease of use
- Sufficiently rigid for repeatability

With these requirements in mind, the inventors designed the apparatus shown in Fig. 19. To confront the first requirement, a sturdy base was designed to easily fix the apparatus to an Aerotech ARMS-200 single axis rate table. The rate table described here was readily available in the inventors lab, however this mounting plate can be easily retrofitted to include other mounting surfaces of different rate tables. A descriptive picture of the CAD model for the mounting structure is shown in Fig. 20. This mount was machined out of general purpose $\frac{1}{2}$in thick 6061 aluminum for lightweight rigidity.

Next, to observe two axis measurements, a circular outer structure was designed. This circular structure will allow complete 360 degree rotation about the x-axis as

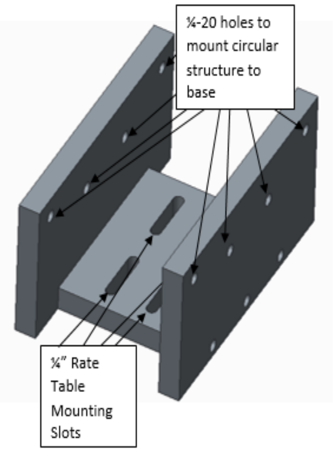

¼-20 holes to mount circular structure to base
¼" Rate Table Mounting Slots

Figure 20: CAD Representation of Rate Table Mount

shown in Fig. 21. This rotation is guided on the inner circle by a slot and two semicircular slides (Fig. 22a-b) which stay 180 degrees from one another at all times. These slides are guided by the inner teflon greased female slot of the circular structure with a male guide. The custom slides along with the circular outer structure were machined by Wilson Works, Inc of Morgantown, WV, while rest of the device was machined by me in the Mechanical and Aerospace Engineering machine shop at WVU. With a rotation about the X-axis of the apparatus, the Y and Z-axis of the 3-axis gyroscopes may be simultaneously spun by the rate table, yielding 2-axis measurements.

Finally to excite the X-axis of the gyroscope an inner rotation had to be designed. Shown in Fig. 23, the non-transparent portion of the CAD model spins the Y-axis of the apparatus. This rotation is locked at every 15 degrees with a spring pull pin shown in the figure and can be rotated for a full 360 degrees. This rotation in the Y-axis of the apparatus causes the X-axis of the board to turn into axis of rotation, therefore leading to rate measurements along the X-axis.

Figures 28, 30, & 31 are images of the new calibration apparatus in configurations which will provide single, double, and tri-axis measurements respectively from an IMU

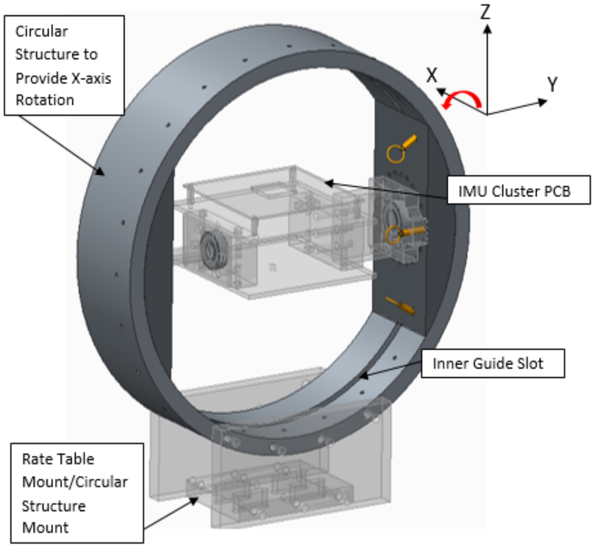
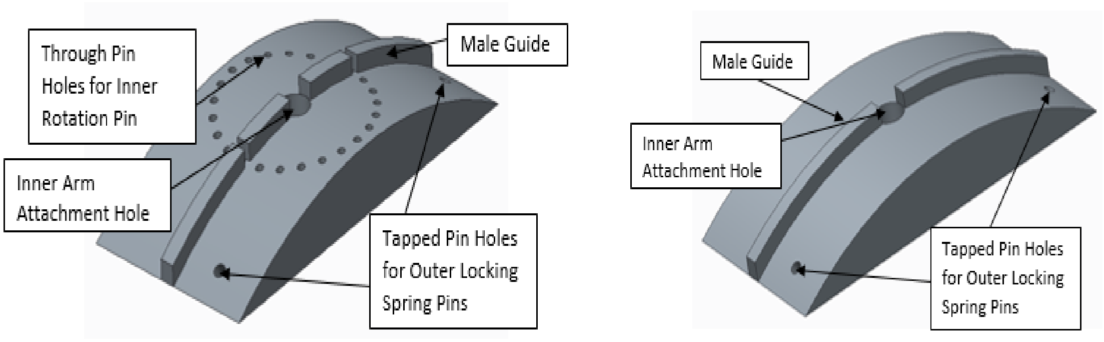

Circular Structure to Provide X-axis Rotation
Z
X
Y
IMU Cluster PCB
Inner Guide Slot
Rate Table Mount/Circular Structure Mount

Figure 21: X-axis Rotation Circular Structure


Through Pin Holes for Inner Rotation Pin
Male Guide
Inner Arm Attachment Hole
Tapped Pin Holes for Outer Locking Spring Pins
Male Guide
Inner Arm Attachment Hole
Tapped Pin Holes for Outer Locking Spring Pins

(a) Semi-Circular Slide 1

(b) Semi-Circular Slide 2

Figure 22: Inner Guide Slides for X-Axis Rotation of the Apparatus

mounted to its inner surface. The images are shown with the apparatus mounted to an Aerotech ARMS-200 rate table and the analog IMU cluster is mounted to the mounting surface of the apparatus via standoffs. Detailed drawings of the apparatus are available in the appendix of this thesis.

By combining the rotations of the axis from the apparatus, 576 unique orientations may be created. As an example, Fig. 28 shows an apparatus X-rotation of 45 degrees and a Y-rotation of 45 degrees. An apparatus orientation such as this will yield measurements from the IMU along all three of its axes. By rotating each of these axes simultaneously, errors between axes (i.e. non-orthogonality) can be estimated

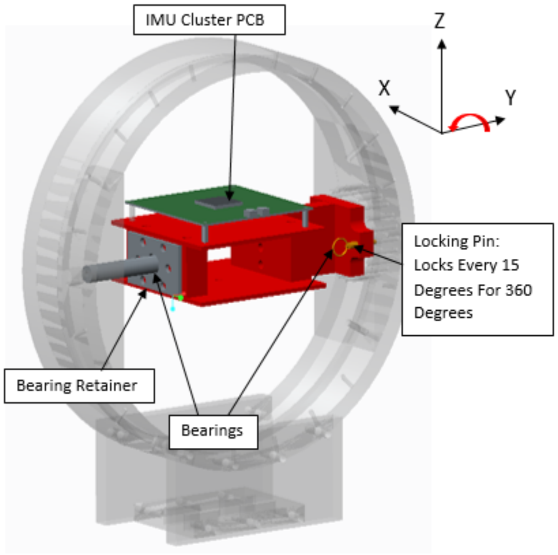


Figure 23: Inner Rotation Structure

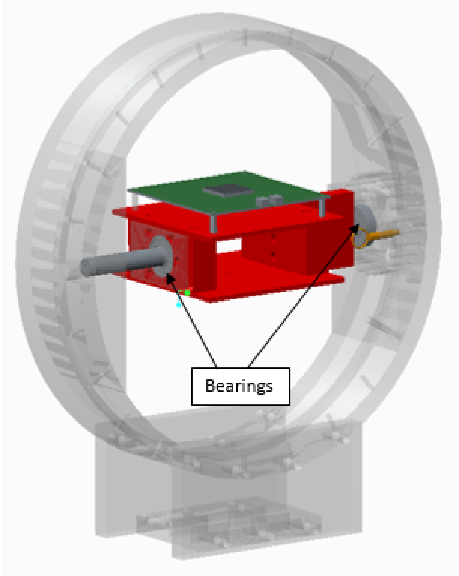


Figure 24: Inner Rotation Bearing Placement

and obtained, which is crucial to complete IMU calibration.

The advantages of such an apparatus are significant compared to that of the other mentioned techniques. To show the significance of the apparatus and rate table combination, Fig. 29-b is a simulated test using a specific orientation of the apparatus and a table rate from a single axis table. As can be observed, a constant rate three-

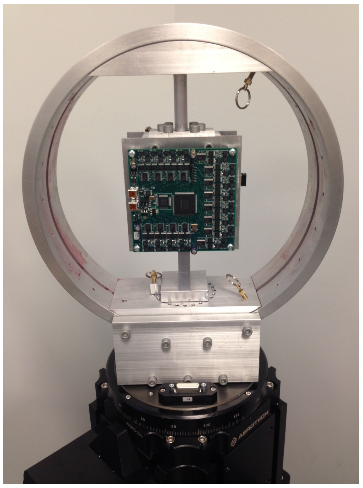


Figure 25: Single Axis IMU Testing

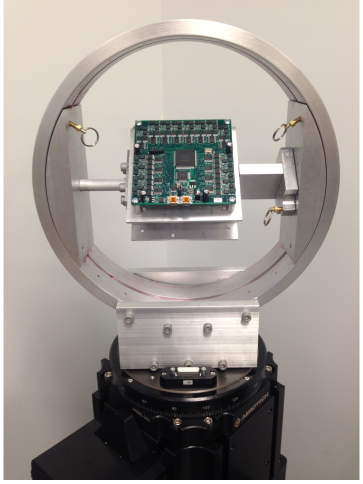


Figure 26: 2-axis IMU Testing

axis test can be achieved. First, the cost to have this apparatus custom machined was roughly $3000. This is significantly less than the other methods described that provide accurate calibration. Eventually, as machining methods mature, the price is expected to decrease. Second, the apparatus is designed with tight tolerances so positions will be highly repeatable. To add, once the IMU is mounted, the user would not have to unmount the IMU until the calibration is complete, which also makes repeatability easier. Lastly, and probably the most beneficial from this design, is the operational ease of use. All of the preceding single-axis methods and apparatuses require increased user handling of the sensor. This apparatus only requires initial

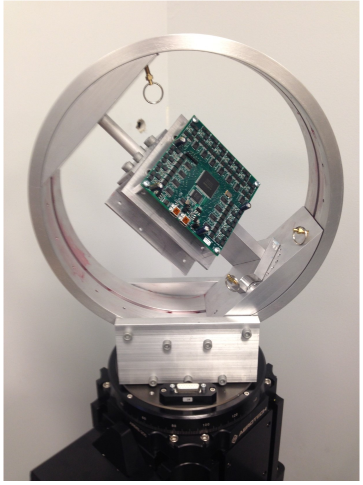


Figure 27: 3-Axis IMU Testing

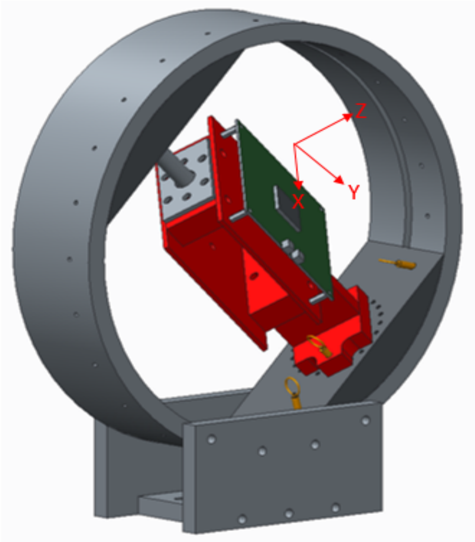


Figure 28: 45-45 Angled Orientation for Angled Measurements

mounting and, moving to different orientations is as simple as pulling a few locking spring pins.

This new calibration apparatus is patent-pending by West Virginia University. Using this calibration apparatus along with the Maximum Likelihood Parameter Estimation technique results in complete and accurate IMU cluster calibration on all three sensor axes using only a single-axis rate table.

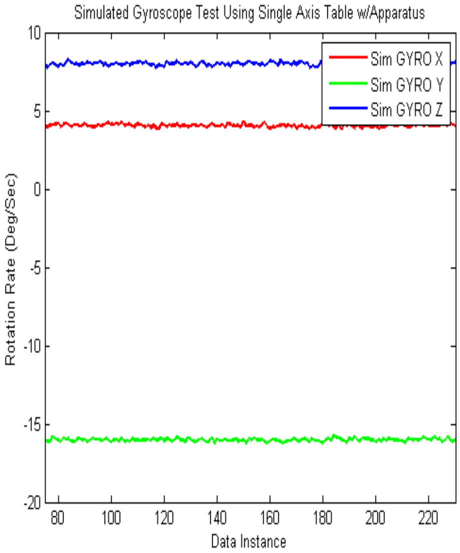


Figure 29: Simulated Single Axis Tests w/ Apparatus

4.3 Revised Calibration Apparatus

After examining a constructed prototype of the apparatus (Fig. 31), the inventors decided to make a few design updates to the outer rotation. While the original prototype functioned as desired, the outer circular ring and the pin slides were difficult to machine and were therefore quite costly. As a result, the outer rotation mechanism has been redesigned. This redesign should make machining much easier, and therefore reduce the cost of future prototypes or systems. In addition, the outer rotation is much more user-friendly since the pins now face the user and not the IMU board. A CAD image of the redesigned calibration apparatus may be seen in Fig. 30.

Figure. 31 shows how the second revision locks at every 15 degree increments. The pin slides are translucent in the image to reveal where the pins lock. Two outer pins connect to the inner guide through holes on the inner guide.

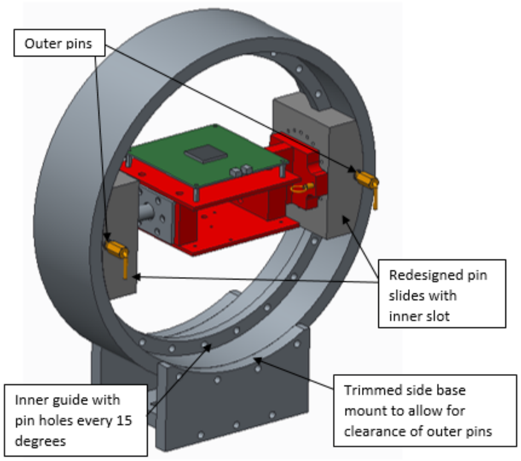

Outer pins
Redesigned pin slides with inner slot
Inner guide with pin holes every 15 degrees
Trimmed side base mount to allow for clearance of outer pins

Figure 30: Revision 2 of the Calibration Apparatus

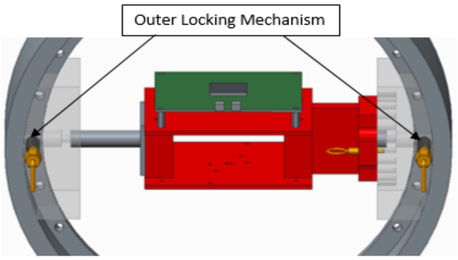


Figure 31: Revision 2 Outer Locking Mechanism

5 Complete IMU Cluster Calibration

The maximum likelihood estimation technique has shown promising advantages over other architectures due to its ability to estimate all errors within the IMU measurement model. In particular, its promising advantages for IMU clusters due to its relatively quick computation time. In earlier work [7], it was explained how this technique is capable of finding the misalignment of each IMU within the swarm. To carry this work further, a complete calibration was performed to show the capabilities of finding misalignment, scale-factor, non-orthogonality, and bias errors within the IMU measurement model. The new digital board was characterized in this calibration. Additionally, the WVU calibration apparatus for single axis rate tables was used to retrieve data from the necessary orientations needed for the MLE. The advantages of this apparatus will be highlighted in this calibration.

To begin, the gyroscope measurement model is,

$$\tilde{\mathbf{w}} = [\mathbf{I}_{3\times3} + \boldsymbol{\delta} + \boldsymbol{\Gamma} + \boldsymbol{\Xi}]\mathbf{w}_S + \boldsymbol{\beta} + \boldsymbol{\nu} \tag{31}$$

where, $\mathbf{I}_{3\times3}$ is a 3×3 identity matrix, $\boldsymbol{\delta}$ is a 3×3 diagonal matrix containing the XYZ scale factor error vector, $\boldsymbol{\Gamma}$ is a 3×3 skew-symmetric cross product matrix containing the XYZ misalignment error vector, $\boldsymbol{\Xi}$ is a 3×3 matrix containing the XYZ non-orthogonality error vector, $\boldsymbol{\beta}$ is the XYZ bias vector, and $\boldsymbol{\nu}$ is zero-mean Gaussian noise. In addition, $\mathbf{w}_S$, is the actual body rate vector in the sensor frame,

$$\mathbf{w}_S = \mathbf{T}_S^I \mathbf{w}_I \tag{32}$$

where, $\mathbf{w}_I$ is the 3×1 body rate vector in the inertial frame (the body rate vector of the rate table). Thus, the goal of the calibration procedure is to estimate $\boldsymbol{\delta}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\Xi}$, and $\boldsymbol{\beta}$.

Since only a single axis table is being used, only the Z-axis is rotating in the inertial frame, and therefore $\mathbf{w}_I$ may be constructed as,

$$\mathbf{w}_I = w_{RT} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \mathbf{w}_e \tag{33}$$

where, w_{RT} is the scalar value of the rate table speed and $\mathbf{w}_e$ is a 3×1 vector of Earth's rotation rate.

$\mathbf{T}_S^I$ is a 3×3 rotation matrix that transforms the body rate vector from the inertial frame to the sensor frame. This matrix is considered a 3-1-2 Euler angle sequence that is generated from the orientation of the WVU custom designed apparatus. This sequence is slightly different from revision 1 of the apparatus (Fig. 18), which used a 3-1-3 Euler angle sequence, because of the construction of the apparatus. The rotation for the new calibration apparatus is given by (3-1-2 Euler angle sequence),

$$\mathbf{T}_S^I = \begin{bmatrix} -\sin\alpha\sin\theta\sin\gamma+\cos\alpha\cos\gamma & \sin\alpha\sin\theta\cos\gamma+\cos\alpha\sin\gamma & -\sin\alpha\cos\theta \\ -\cos\theta\sin\gamma & \cos\theta\cos\gamma & \sin\theta \\ \cos\alpha\sin\theta\sin\gamma+\sin\alpha\cos\gamma & -\cos\alpha\sin\theta\cos\gamma+\sin\alpha\sin\gamma & \cos\alpha\cos\theta \end{bmatrix} \tag{34}$$

where γ is the angle of the rate table while the table is spinning (time is factored into this), θ is the angle of the outer rotation of the apparatus, and α is the angle of the inner rotation of the apparatus. These angles are defined in Fig. 32.

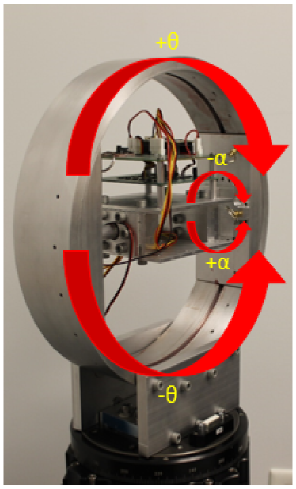


Figure 32: Apparatus Angle Rotation Definitions

5.1 Complete Error Extraction Using Maximum Likelihood Estimation

5.1.1 Acquiring True and Measured Data

To begin, several orientations of the calibration apparatus and several controlled rates of the rate table are selected for use. For the purposes of this thesis, a total number of four orientations and four rates were considered satisfactory. The orientations and rates chosen for this calibration are illustrated in Table 5 & 6 below.

Rate 1	Rate 2	Rate 3	Rate 4
$5°/s$	$10°/s$	$15°/s$	$20°/s$

Table 5: Controlled IMU Cluster Rates

Orientation	α	θ	γ
1	$-45°$	$45°$	$0°$
2	$60°$	$-135°$	$0°$
3	$-60°$	$15°$	$0°$
4	$135°$	$-30°$	$0°$

Table 6: Controlled IMU Cluster Orientations

This will provide 16 unique measurements from each sensor in the cluster (4 orientations × 4 rates = 16 combinations) . Next, 16 ”truth” body rate measurements ($\mathbf{w}_S$) are individually created using Eq. 32 based on the 16 orientation and rate combinations that were chosen. It is assumed $\mathbf{w}_S$ is the same for each sensor for any specific orientation. Next, the digital IMU cluster is placed on the apparatus and exposed to each of the 16 orientation and rate combinations. Measurements from the cluster are taken at each orientation-rate combination and the measurements may be compared with the truth body rates to estimate the error terms in the IMU model.

5.1.2 Algorithmic Layout to Provide MLE Error Estimation

Recall the gyro measurement model introduced earlier,

$$\tilde{\mathbf{y}} = [\mathbf{I}_{3\times3} + \boldsymbol{\delta}_i + \boldsymbol{\Gamma}_i + \boldsymbol{\Xi}_i]\mathbf{w}_{Sk} + \boldsymbol{\beta}_i + \boldsymbol{\nu} \tag{35}$$

where $\tilde{\mathbf{y}}$ is the 3×1 XYZ body rate measurements from the i-th sensor in the k-th orientation and rate combination, $\mathbf{I}_{3\times3}$ is a 3×3 identity matrix, $\mathbf{w}_{Sk}$ is the actual 3×1 body rate vector in the sensor frame of the k-th orientation and rate combination, and $\boldsymbol{\nu}$ is zero mean Gaussian noise. Additionally, $\boldsymbol{\delta}_i$ is a diagonal matrix containing the XYZ scale factor error,

$$\boldsymbol{\delta}_i = \begin{bmatrix} \delta_{ix} & 0 & 0 \\ 0 & \delta_{iy} & 0 \\ 0 & 0 & \delta_{iz} \end{bmatrix} \tag{36}$$

$\boldsymbol{\Gamma}_i$ is a skew symmetric cross product matrix containing the XYZ misalignment error,

$$\boldsymbol{\Gamma}_i = [\boldsymbol{\gamma}_i \times] = \begin{bmatrix} 0 & -\gamma_{iz} & \gamma_{iy} \\ \gamma_{iz} & 0 & -\gamma_{ix} \\ -\gamma_{iy} & \gamma_{ix} & 0 \end{bmatrix} \tag{37}$$

$\boldsymbol{\Xi}_i$ is a strictly lower triangular matrix containing the XYZ non-orthogonality,

$$\boldsymbol{\Xi}_i = \begin{bmatrix} 0 & 0 & 0 \\ \xi_{iyz} & 0 & 0 \\ -\xi_{izy} & \xi_{izx} & 0 \end{bmatrix} \tag{38}$$

and β_i is the 3x1 XYZ bias error vector.

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{ix} \\ \beta_{iy} \\ \beta_{iz} \end{bmatrix} \tag{39}$$

Because a MLE is being performed, the errors to which we desire to estimate must be placed into the state matrix. For that reason, the measurement equation may be rewritten as,

$$\tilde{\mathbf{y}} - \mathbf{w}_{Sk} = \begin{bmatrix} [\mathbf{w}_{Sk}\backslash] & -[\mathbf{w}_{Sk}\times] & \boldsymbol{\zeta}_{W_{s_k}} & \mathbf{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_i \\ \gamma_i \\ \boldsymbol{\Xi}_i \\ \boldsymbol{\beta}_i \end{bmatrix} + \boldsymbol{\nu} \tag{40}$$

where $[\mathbf{w}_{Sk}\backslash]$ is a diagonal matrix containing the $\mathbf{w}_S$ XYZ vector of the k-th combination, $-[\mathbf{w}_S\times]$ is a skew symmetric cross product matrix containing the XYZ $\mathbf{w}_S$ vector of the k-th combination, and $\boldsymbol{\zeta}_{W_{s_k}}$ is constructed as,

$$\boldsymbol{\zeta}_{W_{s_k}} = \begin{bmatrix} 0 & 0 & 0 \\ W_{Skx} & 0 & 0 \\ 0 & -W_{Skx} & W_{Sky} \end{bmatrix} \tag{41}$$

Due to the way the measurement equation is now constructed, the measurement sensitivity matrix, $\mathbf{H}_k$, is

$$\mathbf{H}_k = \begin{bmatrix} [\mathbf{w}_{Sk}\backslash] & -[\mathbf{w}_{Sk}\times] & \boldsymbol{\zeta}_{W_{s_k}} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{42}$$

Since there are 16 combinations of rates and orientations there will be 16 $\mathbf{H}_k$, 16 XYZ rate measurements from the sensor ($\tilde{\mathbf{y}}_k$), and 16 XYZ synthetically generated truths ($\hat{\mathbf{w}}_{S_k}$). Assuming the sequential measurement errors are uncorrelated ($E[\boldsymbol{\nu}_i\boldsymbol{\nu}_j^T] = 0$), the 16 combinations may be summed together into one solution (n=16). The MLE solution ($\hat{\boldsymbol{x}}_i$) which includes the XYZ scale factor ($\hat{\boldsymbol{\delta}}_i$), misalignment($\hat{\boldsymbol{\gamma}}_i$), non-orthogonality ($\hat{\boldsymbol{\Xi}}_i$), and bias error ($\hat{\boldsymbol{\beta}}_i$) is given by,

$$\hat{\boldsymbol{x}}_i = \left(\sum_{k=1}^{n} \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \right)^{-1} \sum_{k=1}^{n} \mathbf{H}_k^T \mathbf{R}_k^{-1} (\tilde{\mathbf{y}}_k - \hat{\mathbf{w}}_{S_k}) \tag{43}$$

Assuming the errors on each axis are uncorrelated, the measurement covariance, $\mathbf{R}_k$, is computed as:

$$\mathbf{R}_k = \sigma^2\mathbf{I}_{3\times3} \tag{44}$$

where σ is the sensor measurement standard deviation. Since each sensor was manufactured the same and is delivered with the same specifications, σ is assumed the same for each of the sensors unless proven different through testing. The 12×1 output from the MLE solution is,

$$\hat{\boldsymbol{x}}_i = \begin{bmatrix}\hat{\delta}_{xi} & \hat{\delta}_{yi} & \hat{\delta}_{zi} & \hat{\Gamma}_{xi} & \hat{\Gamma}_{yi} & \hat{\Gamma}_{zi} & \hat{\Xi}_{xi} & \hat{\Xi}_{yi} & \hat{\Xi}_{zi} & \hat{\beta}_{xi} & \hat{\beta}_{yi} & \hat{\beta}_{zi}\end{bmatrix}^T \tag{45}$$

Because there are 16 gyroscopes in the cluster, there will be 16 maximum likelihood estimations, after which each error for each sensor is observed. These errors are then used to calibrate each of the sensor so that future measurements are not affected by them.

5.1.3 Calibrating the IMU

Now that the errors have been estimated, applying the calibration to future measurements is straightforward. Begin by solving the Eq. 35 for $\mathbf{w}_{Sk}$,

$$\mathbf{w}_{Sk} = [\mathbf{I}_{3\times3} + \boldsymbol{\delta} + \boldsymbol{\Gamma} + \boldsymbol{\Xi}]^{-1}(\tilde{\mathbf{y}} - \boldsymbol{\beta} - \boldsymbol{\nu}) \tag{46}$$

The best estimate of $\mathbf{w}_{Sk}$ is found by taking the expected value,

$$E[\mathbf{w}_{Sk}] = \hat{\mathbf{w}}_{Sk} = [\mathbf{I}_{3\times3} + \boldsymbol{\delta} + \boldsymbol{\Gamma} + \boldsymbol{\Xi}]^{-1}(\tilde{\mathbf{y}} - \boldsymbol{\beta}) \tag{47}$$

This is the calibrated output of any one MEMS IMU in the cluster. Each sensor in the cluster will have its own calibration equation containing the sensor's errors so that, for instance, errors from IMU 2 do not go towards calibrating IMU 4's measurements.

5.2 Results and Discussion

A MATLAB script was written to implement the calibration algorithms described above. Experimental data was collected using a single-axis rate table and the new calibration apparatus, and then the calibration parameters were computed. The results are shown in Table 7 and 8.

	IMU 1	IMU 2	IMU 3	IMU 4	IMU 5	IMU 6	IMU 7	IMU 8
δ_x	-0.002	0.001	-0.003	0.005	0.003	0.007	0.016	-0.002
δ_y	-0.003	-0.009	0.006	-0.004	0.004	0.002	0.007	0.004
δ_z	0.031	0.021	0.026	0.018	0.023	0.015	0.025	0.029
Γ_x (deg)	-0.930	-0.923	-1.043	-0.959	-1.117	-0.595	-0.973	-0.693
Γ_y (deg)	-0.499	-0.285	-0.045	-0.337	-0.865	-1.238	-1.041	0.177
Γ_z (deg)	0.131	-0.316	0.293	0.648	0.077	-0.189	0.151	0.210
Ξ_x (deg)	-0.175	-0.043	-0.081	-0.041	-0.094	0.094	-0.009	-0.219
Ξ_y (deg)	-0.527	-0.160	0.370	-0.543	-1.308	-0.889	-1.578	0.659
Ξ_z (deg)	0.257	0.507	1.081	0.352	1.053	-0.895	0.414	-0.642
β_x(deg/sec)	-1.636	-1.957	-0.340	0.788	-1.190	-1.108	-1.385	0.031
β_y (deg/sec)	1.321	1.185	0.796	1.448	0.910	0.638	1.498	0.858
β_z (deg/sec)	0.068	0.247	-0.449	0.088	-0.303	0.268	-0.094	-0.367

Table 7: Individual IMU Error Coefficients 1-8

	IMU 9	IMU 10	IMU 11	IMU 12	IMU 13	IMU 14	IMU 15	IMU 16
δ_x	0.008	0.009	0.012	0.003	0.003	-0.001	0.005	0.006
δ_y	-0.012	-0.008	0.004	0.000	-0.003	0.004	0.004	0.002
δ_z	0.036	0.021	0.029	0.028	0.033	0.022	0.014	0.015
Γ_x (deg)	-0.983	-0.790	-0.403	-0.901	-1.619	-0.871	-0.753	-0.389
Γ_y (deg)	-0.783	-1.516	-0.663	-0.861	-0.789	-1.432	-1.276	-0.602
Γ_z (deg)	0.094	0.425	-0.018	-0.024	-0.153	0.146	0.194	0.153
Ξ_x (deg)	-0.259	-0.104	-0.239	-0.193	-0.164	-0.118	-0.022	-0.006
Ξ_y (deg)	-0.901	-2.376	-0.039	-0.225	-0.203	-1.165	-0.958	0.225
Ξ_z (deg)	-0.813	-0.613	-1.921	-0.856	0.732	-0.408	-1.161	-1.284
β_x (deg/sec)	-1.206	-1.651	-0.638	-1.555	-1.247	-1.462	-0.411	-2.621
β_y(deg/sec)	-0.234	0.359	0.389	0.847	0.443	-0.501	0.210	1.410
β_z (deg/sec)	-0.540	-1.341	-0.172	-2.515	0.095	0.338	-0.435	0.874

Table 8: Individual IMU Error Coefficients 9-16

Each of these error coefficients are then used to calibrate the gyroscope according to Eq. 47. This must be performed 16 times (once for each of IMU in the cluster) per data instance because each sensor has different error coefficients.

To test the calibration a random orientation and rate were chosen. For the first test, the table was spun at 12 deg/sec in the counter-clockwise direction and the inner and outer rotations of the apparatus were set at 15 and 75 degrees respectively. The raw data from this test was collected for each sensor and each axis for 60 seconds (approx. 2,700 data points at a 115,200 baud rate). Two forms of the data were computed 1.) A calibrated version using the scale factor conversion and Eq. 47 with each of the error coefficients described above and 2.) an uncalibrated version in which only the manufacturers scale factor conversion is applied (converts digital number to deg/sec). The calibrated output should have a smaller systematic error than the uncalibrated output. Residuals are computed as follows;

$$\mathbf{r}_{cal} = \hat{\mathbf{w}}_{Sk} - \mathbf{w}_{Sk} \tag{48}$$

$$\mathbf{r}_{uncal} = \tilde{\mathbf{w}}_{Sk} - \mathbf{w}_{Sk} \tag{49}$$

where $\hat{\mathbf{w}}_{Sk}$ is from Eq. 47, $\tilde{\mathbf{w}}_{Sk}$ is the raw gyro measurement, and $\mathbf{w}_{Sk}$ is the truth from Eq. 32. Results are shown in Figs. 33, 34, & 35 for the X, Y and Z gyro axes, respectively.

In nearly every case the calibrated output has a smaller residual than the uncalibrated output. There is nearly zero error between the calibrated data and the truth, signifying that the calibration worked and that the error coefficients are very close to their actual values. On the other hand, the uncalibrated data is farther from the truth and therefore produces a larger residual. Because these residuals are for a random test case that was run after the calibration these results demonstrate that our calibration approach significantly improves the system's performance.

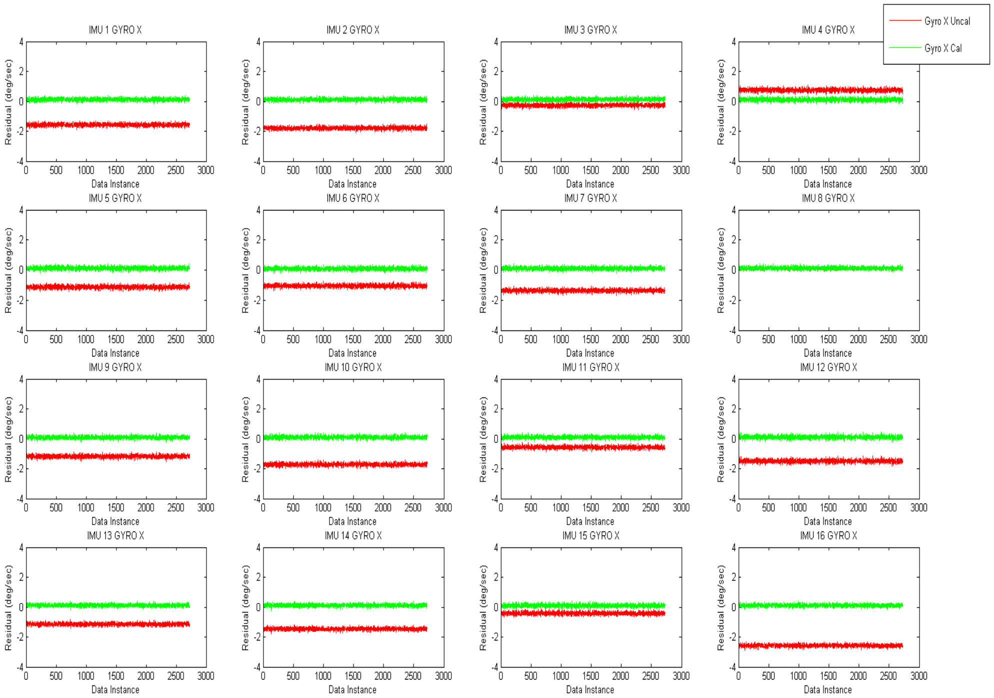


Figure 33: Gyro X Residuals From all 16 sensors

In Fig. 35 the difference between the calibrated and uncalibrated residuals of the Z-axis gyroscopes is not as great as the other two axes, but the calibrated measurements were still slightly more accurate. This indicates that there was less error in the scale factor conversion for the Z-axis outlined in the IMU specification sheet, because the measurements that had the conversion factor applied (uncalibrated measurements) nearly matched the calibrated measurements.

The high performing solution to the IMU cluster relies on averaging all of the 16 measurements from the X, Y, and Z gyroscopes. Therefore the calibrated data will all be averaged to provide 1-XYZ solution per data instance. Figure 36 is an averaged solution showing the averaged XYZ calibrated solution over 2700 data instances, the averaged XYZ uncalibrated solution over 2700 data instances, and the XYZ truth over 2700 data instances. It is evident through this figure, that the calibrated solution provides an estimate that nearly falls on top of the truth in each of the X, Y,

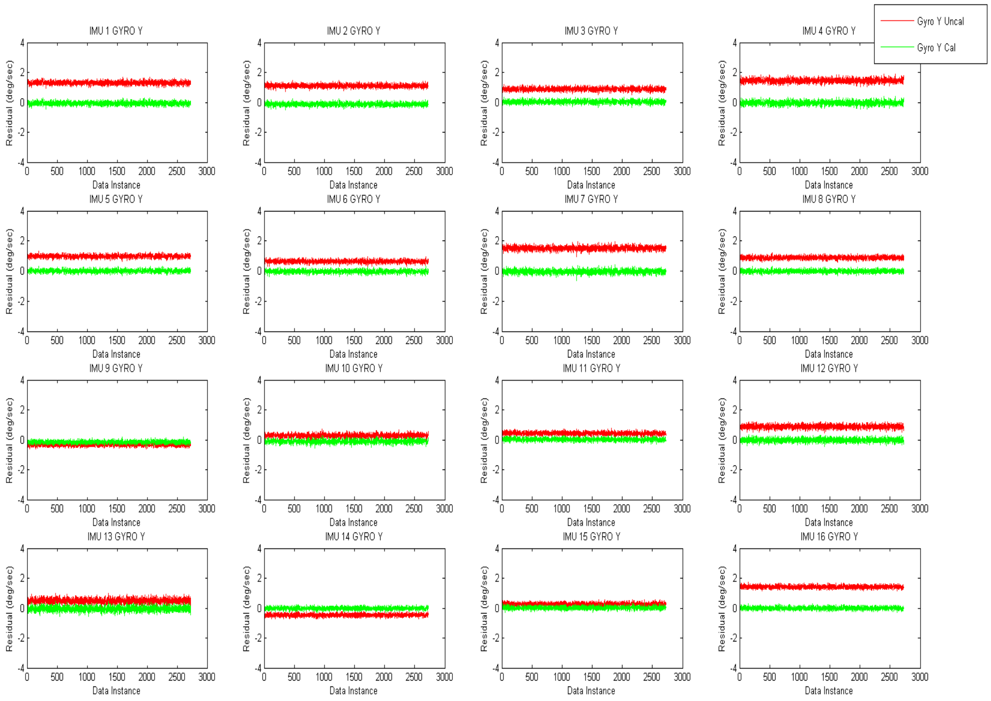


Figure 34: Gyro Y Residuals From all 16 sensors

and Z cases, further signifying that the calibration has worked.

Finally, in Fig. 37, XYZ histograms were constructed to show the combined calibrated solution residuals versus the combined uncalibrated residuals. In each case, the residuals of the combined calibrated measurements far outperforms the residuals of the combined uncalibrated measurements.

In addition to the results above, a 200 second X, Y, and Z three axis trajectory was performed and plotted in Fig. 38, Fig. 39, Fig. 40, and Fig. 41. In Fig. 40 is the full Z-axis trajectory showing the uncalibrated, calibrated, and truth of the XYZ gyroscopes in the IMU cluster. Because it is difficult to see the differences in trajectory rates, Fig. 41 shows a small highlighted version of that Z-axis trajectory. As can be observed, the calibrated falls very close to the truth, while the uncalibrated falls much farther from it. Having accurate gyroscopes rates is the key to understanding

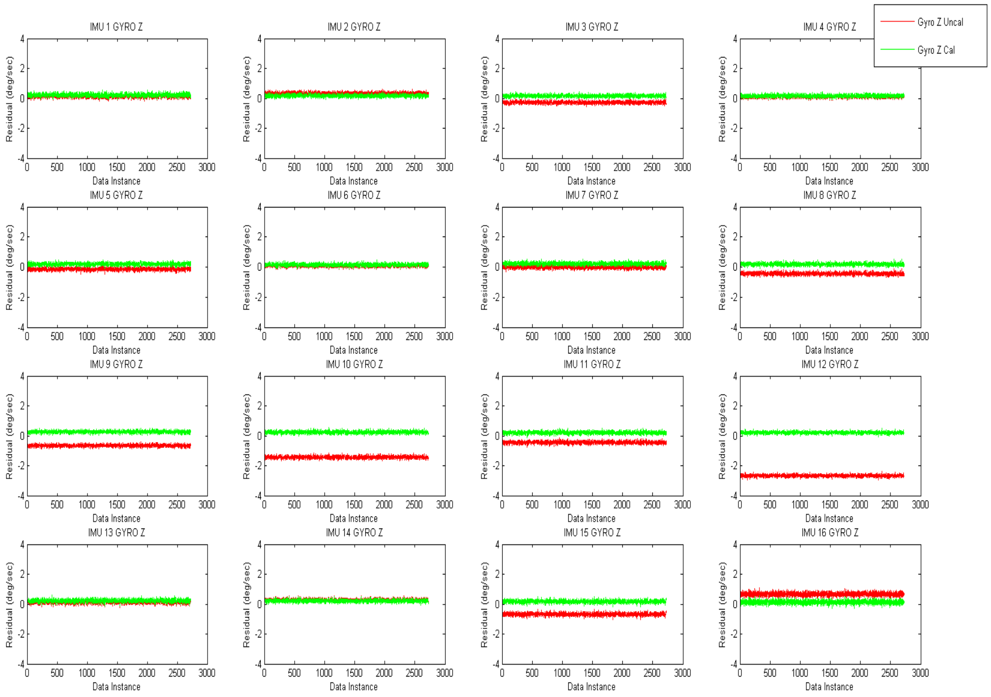


Figure 35: Gyro Z Residuals From all 16 sensors

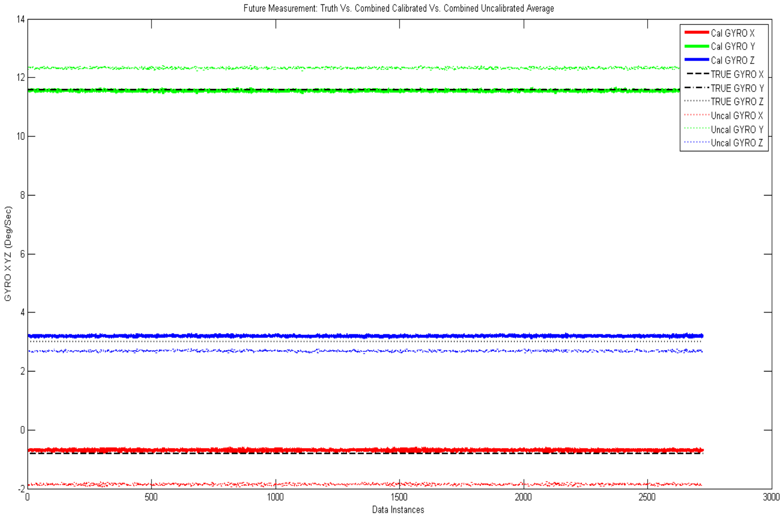


Figure 36: XYZ Truth, XYZ Calibrated, and XYZ Uncalibrated Test Data Set

a small satellites orientation on orbit. Integrating an uncalibrated rate will provide a poor estimate of position along the XYZ axes and therefore provide a false solution for orientation. Although it may not be as crucial on CubeSats, understanding ori-

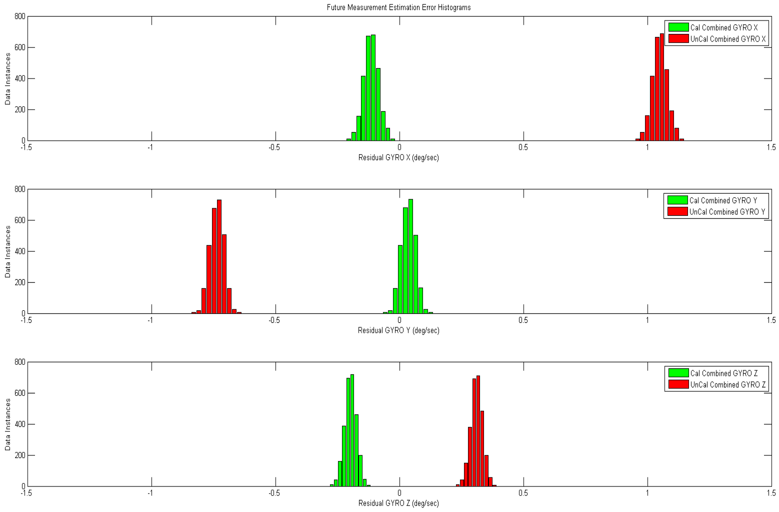


Figure 37: Calibrated Averaged and Uncalibrated Averaged Histograms of the XYZ residuals

entation when making maneuvers in space to 1.) keep or change orbit or 2.) orient the spacecraft to point certain directions for example, to capture images of the earth or point solar cells at the sun, is highly critical. Therefore the calibrated rates must provide a highly accurate understanding of the XYZ rates so that an accurate orientation may be obtained. As an example, Fig. 42 shows the integrated calibrated and uncalibrated Z-axis gyro rates of the trajectory in Fig. 40 to find an estimate of the angular position of the Z-axis of the IMU. The integrated calibrated position falls much closer to the the true position recorded from the feedback of the rate table while, the integrated uncalibrated position is quickly diverging farther from the truth.

It should be noted that with each power cycle of the IMU, the calibration bias parameter (β) may change slightly. Therefore, precision navigation systems should include gyro bias as states to estimate their navigation filter.

This calibration process should be optimized for the rates specific to an application. Calibrating at rates far under or far over the rates typically seen in the application can lead to large differences in error coefficients, which leads to larger differences in calibrated residuals. There are also orientations which reveal more in-

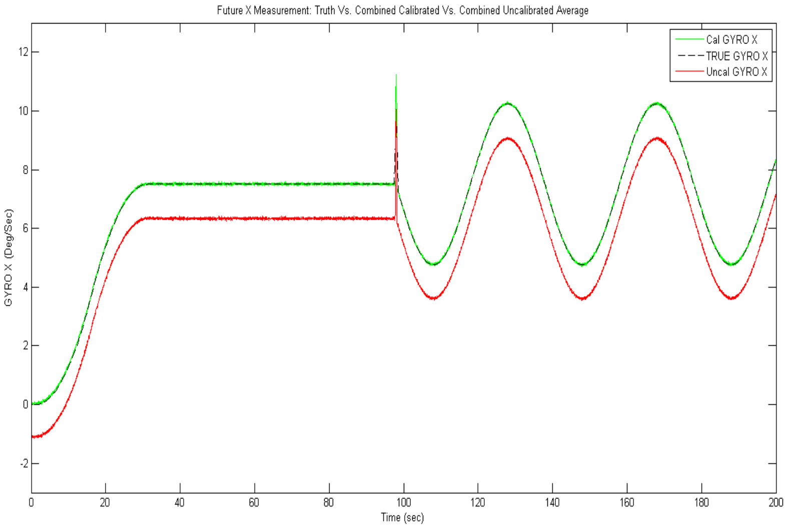


Figure 38: X-Axis Trajectory Test; Calibrated, Uncalibrated, and Truth

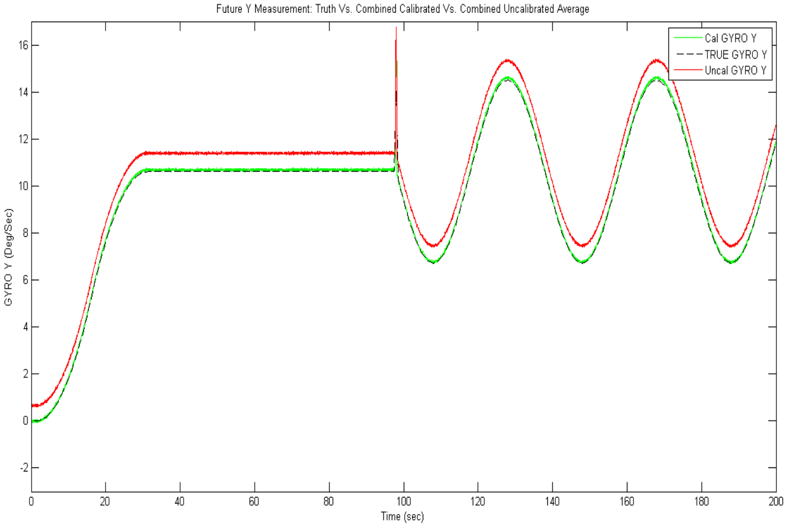


Figure 39: Y-Axis Axis Trajectory Test; Calibrated, Uncalibrated, and Truth

formation than others, and care should be taken to ensure that the combinations of the orientations and rates chosen are sufficient to make all the calibration terms observable. Lastly, bias can significantly change with sensor temperature, thus it is important to account for this in the calibration equation. For these tests, the IMU was turned on for 30 minutes prior so that each of the sensors on the PCB would have a chance to warm up and get to a constant temperature.

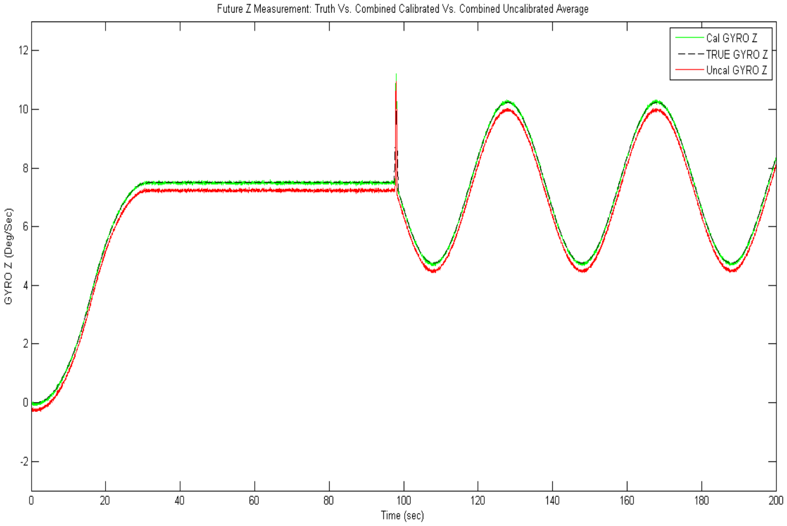


Figure 40: Z-Axis Trajectory Test; Calibrated, Uncalibrated, and Truth

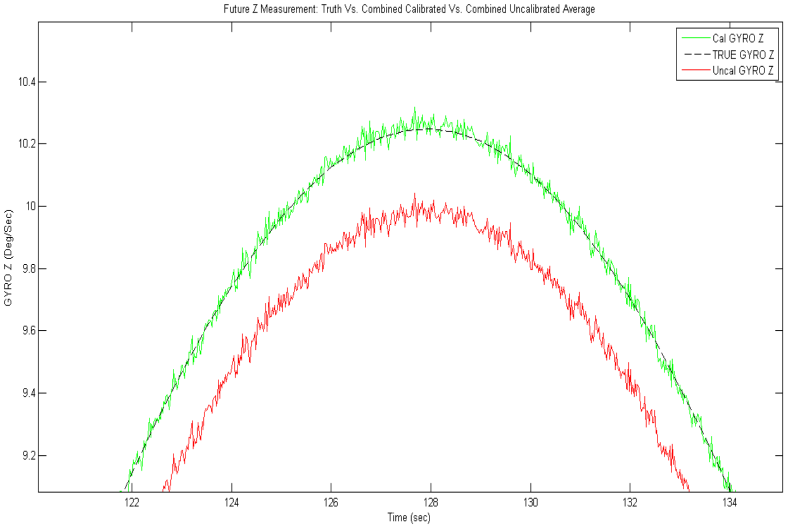


Figure 41: Zoomed Figure 40 from Z-axis Trajectory

To help verify the performance of the IMU cluster, the bias stability and angle random walk (ARW) were measured from a 2-hour static test. These two parameters may be acquired from an Allan variance plot. Allan variance is a characterization of the noise as a function of averaging time and may be computed as [20],

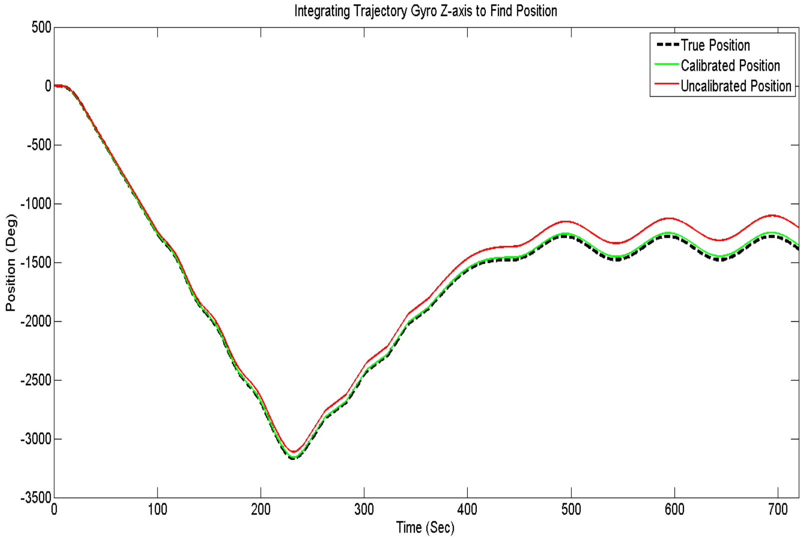


Figure 42: Z-axis Gyro Trajectory Integrated Angular Position: Truth, Calibrated, and Uncalibrated

$$\sigma_{\Omega}^{2}(\tau) = \frac{1}{2(\tau^{2})}\langle(\theta_{k+2M} - 2\theta_{k+M} + \theta_{k})^{2}\rangle \tag{50}$$

which is the same as,

$$\sigma_{\Omega}^{2}(\tau) = \frac{1}{2(\tau^{2})(N-2M)}\sum_{k-1}^{N-2M}(\theta_{k+2M} - 2\theta_{k+M} + \theta_{k})^{2} \tag{51}$$

where N is the number of samples, τ is the observation period, and M is the number of samples per observation period. The observation period or averaging time is then plotted against the square root of the Allan variance (Allan deviation) on a log-log scale, and charts similar to Fig. 43 may be produced. The absolute minimum of the Allan deviation curve is then bias stability. This is the deviation of the bias over a certain averaging time. The lower this bias stability, the better performing the gyroscope, because there is less deviation in the measurements over a longer averaging time. The angle random walk is found from the Allan deviation curve when the averaging time is 1. When the gyroscope rate is integrated over time to find an angle, the noise in the measurement will cause the angle to deviate from its actual angular position. If the measurements to find this ARW are taken during static tests, the angle should be zero if integrated without noise. However, integrating the gyro measurement with noise will cause the angle to "walk" from zero. This walk is characterized by the angle to which the integrated angular position walks from zero over time. Similar to bias stability, the lower the ARW, the better performing the gyro, because there is less noise in the gyro measurements (less noise in the gyro measurements causes better estimates of actual angular position when integrated)[20]. Figure 43 shows the Allan deviation plot for both the combined IMU cluster output as well as for each of the 16 individual MEMS IMUs. Results are plotted separately for each axis. As can be seen, by combining the output of all of the 16 MEMS sensors in the cluster the bias stability decreases significantly as compared to the single MEMS case,

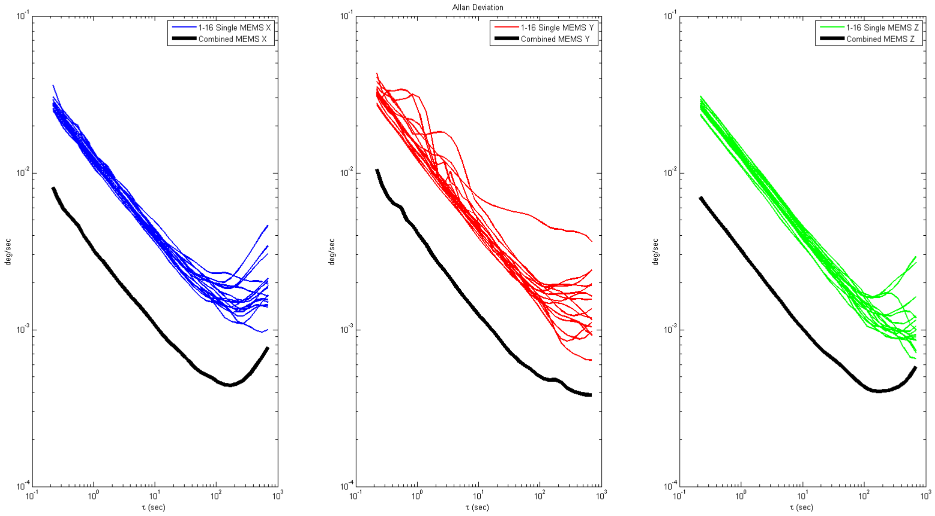


Figure 43: Digital IMU Cluster & Single MEMS XYZ Gyroscope Allan Deviation

The importance of averaging the IMU outputs is even more apparent in Fig. 44 where it is shown how bias stability is affected by the number of sensor outputs that are averaged. As the number of sensors in the cluster is increased, the performance increases, because the bias stability greatly decreases. As more sensor units are added (IMUs>16) it is expected that this performance will increase slightly further, although as shown in Fig. 44 and 45 the performance begins to plateau. Additionally, the theoretical curve was plotted in red using Eq. 1. The start of the measured curve was fed as the start of the theoretical. As a result, the theoretical performance closely matches that of the measured.

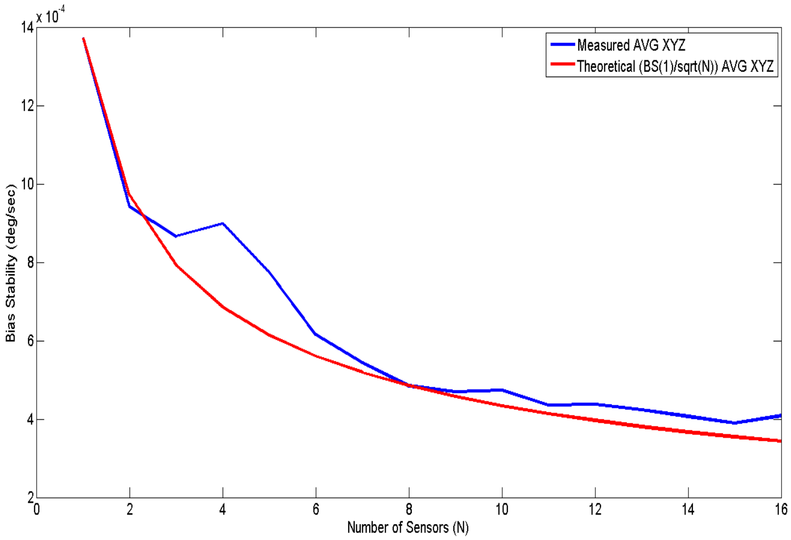


Figure 44: Bias Stability Vs. Number of Sensors in Digital Cluster

In addition, the angle random walk, which is another important performance parameter, was calculated from the Allan deviation plots. The angle random walk was plotted as function of the number of MEMS sensors within the cluster (Fig. 45). Similar to the bias stability chart (Fig. 44) this chart follows a decreasing trend as more sensors are averaged together in the cluster. Within this chart the theoretical curve is also plotted using Eq. 1 and the first data point from the measured curve. Once again, the theoretical closely matches the the measured curve, proving that the equation is representative of the performance increase with an increase in cluster size. This performance increase, once again, shows the importance of this clustering technology.

A performance comparison between the digital IMU cluster and a single MEMS

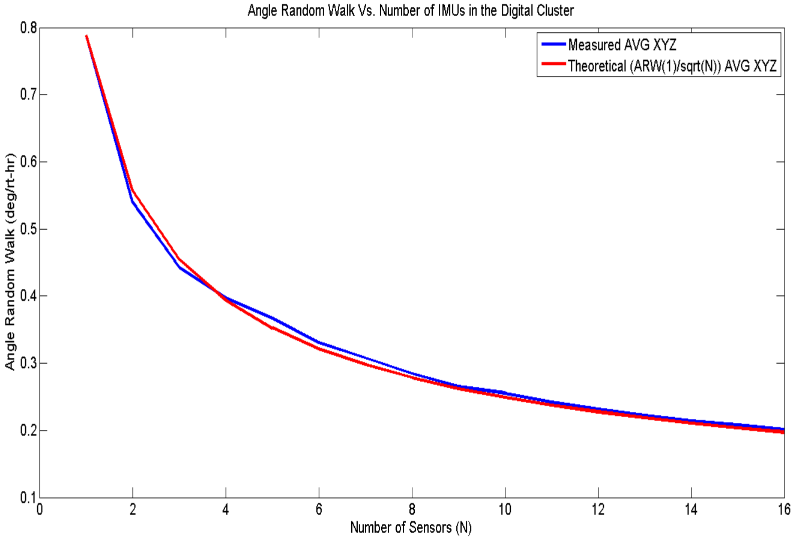


Figure 45: Angle Random Walk Vs. Number of Sensors in Digital Cluster

gyroscope is provided in Table 9 and Table 10. Table 9 shows the numeric values of the bias stability along each axis in deg/hr. Additionally, the angle random walk was calculated in each of the sensors and the data are shown in Table 10. From these observables it can be concluded that combining many IMUs in a cluster clearly improves performance, but significant forward work is still required to match the performance of a tactical grade IMU.

	Bias Stability		
	X (deg/hr)	Y (deg/hr)	Z (deg/hr)
Single MEMS	7.0358	4.2749	3.5102
Digital Cluster	1.5849	1.3807	1.4499

Table 9: Single MEMS and Digital Cluster Measured Bias Stability

	Angle Random Walk			
	X (deg/rt-hr)	Y (deg/rt-hr)	Z (deg/rt-hr)	Spec (deg/rt-hr)
Single MEMS	0.6919	0.8448	0.8269	0.3000
Digital Cluster	0.1835	0.2367	0.1825	- -

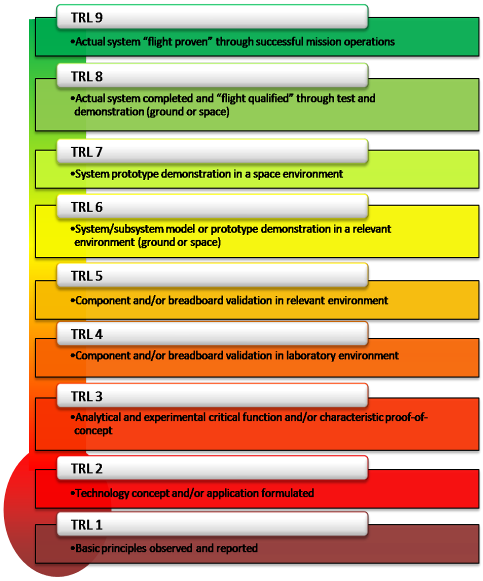
Table 10: Single MEMS and Digital Cluster Measured Angle Random Walk

To conclude, the digital board did not match the performance of a tactical grade sensor shown in the data provided above. However, the cluster did show a promising performance increase as compared the single MEMS output. Since the cluster currently takes up a small amount of space on the PCB, it may be possible to increase the number of sensors in the future. This increase in number of sensors should once again increase the performance, while still keeping cost, mass, power, and volume at a fraction of that of a typical tactical grade IMU. It should be noted however, that tactical grade IMU's have a bias stability of 0.1-1 deg/hr and an angle random walk of 0.1 deg/rt-hr [21]. Therefore, because the low-performing end of the grade is 1 deg/hr the IMU cluster nearly met the grade in terms of bias stability. In terms of angle random walk, the IMU cluster came close once again to meeting this grade. With this increased knowledge, the team at WVU and MU can design future revisions to the IMU cluster that may increase performance of such technology even further.

6 Maturing IMU Cluster Technology with a Sounding Rocket Flight

In order to help verify the digital IMU cluster technology in space and launch applications it will be flown on a future sounding rocket flight out of NASA Wallops Flight Facility (WFF) as part of an Undergraduate Student Implementation Project (USIP) payload at West Virginia University. This launch will help the IMU cluster achieve a technology readiness level (TRL) of 7 as seen in Fig. 46.

Preparing the board for the sounding rocket flight was part of the work done for this thesis. Up until this point the board has been proven in ground-based testing, thus the testing environment allowed the redundant data to be off-loaded to an acquisition computer and processed with a powerful computer. However, on board


TRL 9
•Actual system "flight proven" through successful mission operations
TRL 8
•Actual system completed and "flight qualified" through test and demonstration (ground or space)
TRL 7
•System prototype demonstration in a space environment
TRL 6
•System/subsystem model or prototype demonstration in a relevant environment (ground or space)
TRL 5
•Component and/or breadboard validation in relevant environment
TRL 4
•Component and/or breadboard validation in laboratory environment
TRL 3
•Analytical and experimental critical function and/or characteristic proof-of-concept
TRL 2
•Technology concept and/or application formulated
TRL 1
•Basic principles observed and reported

www.NASA.gov

Figure 46: Technology Readiness Levels

a sounding rocket, the data must be either stored at high rates on-board for post-processing or an on-board, low-powered computer must process the data in real-time. In addition, to collect valuable data, the board must survive the flight. For that reason, a structure was designed and is to be implemented such that any environmental effects related to a rocket flight will be survivable. Additionally, this technology will eventually fly in space, therefore it is necessary to prove that it can survive launch accelerations and vibrations.

The entire USIP payload measures 13 inches in diameter by 12 inches high and is the only university payload out of the 10 payloads on-board. Currently the IMU swarm is one of seven experiments that are planned for USIP's payload. A CAD model of USIP's payload may be seen in Fig. 47. Since the rocket will end it's journey in the Atlantic ocean, the payload has a sealed section where the IMU swarm will operate. In addition, an open section is available for all external experiments. In Fig. 47, the IMU structure containing the IMU cluster is shown to the left. The IMU cluster has its own structure so that in the event of a hard impact the structure will help the IMU maintain its integrity and also keep other objects inside the sealed section away from the PCB to which the IMU cluster is implemented. This IMU structure was created to simulate the outer structure of a 1U CubeSat which represents the application that this IMU was constructed for. In addition, the structure was designed to be the height of the inside of the sealed section to provide pillar like support. Should the top of the section encounter a hard impact, this structure will provide support as to not compromise the integrity of the o-ring seal to keep water out.

Data from this flight will help qualify the IMU in an applicable environment once it reaches apogee. The launch portion of the flight will mostly help verify survivability since motor launch vibrations will likely render the MEMS gyroscope data useless. When the flight reaches apogee, the motor will detach, and the rocket will slowly spin in a space environment for a few minutes, which closely resembles that of a CubeSat tumbling in space. Therefore, data taken during this time period will be the most beneficial to proving the technology. Data will be telemetered to the ground from the IMU cluster during the flight, to ensure that the team has a high probability of retrieving the data even if the payload is never recovered or if there is a water leak in the sealed section caused by a hard landing impact. Additionally, data will also be stored on-board at slightly higher rates than what is sent to the ground. The sounding rocket flight is scheduled for August 2015, after which the team looks forward to analyzing the data such that this IMU cluster technology can take another large step in the flight verification process.

7 Fault Detection, Isolation, and Recovery for Large Clusters of Inertial Measurement Units [22]

With such a large cluster of sensors, there is a need for a Fault Detection, Isolation, and Recovery (FDIR) system to identify failed IMUs and prevent them from

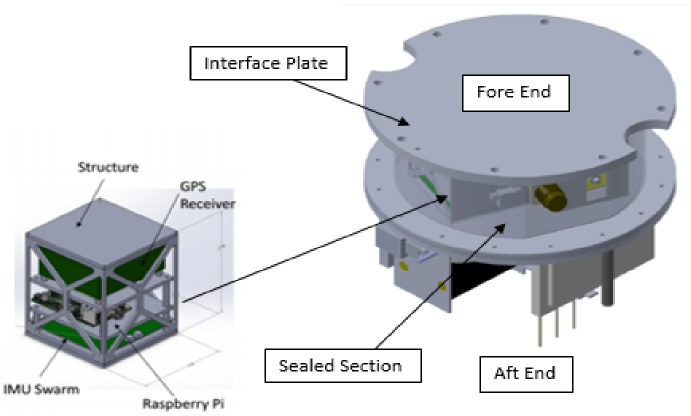


Figure 47: CAD representation of WVU USIP Payload

corrupting the output of the entire cluster. Therefore, the present work develops a FDIR architecture that can identify outlying or erroneous data outputs from large amounts of real-time parallel data, and then prevent erroneous outputs from being incorporated into the state estimation solution. This new work explores FDIR for large IMU clusters using a k-th nearest neighbor algorithm to identify failed IMUs. A Monte Carlo simulation is used to determine the reliability of the technique under random failures of various kinds/sizes. The result of this work is a robust FDIR architecture for use in processing large quantities of redundant IMU measurement information.

7.1 FDIR Introduction

Redundant information is often used by vehicle navigation systems to reduce the error in the estimate of the vehicle state. Measurements are typically fused in a Kalman Filter to produce an optimal state estimate [23]. For such a system to work, however, it is important that faulty measurements not be processed by the filter. Incorporation of faulty measurements may corrupt the state estimate and potentially endanger the vehicle.

The availability of redundant data allows for measurements to be cross-checked with one another before being passed along to the filter. Likewise, new measurements can be compared with what the filter expects and rejected if the difference is too great. The software system that performs these functions is typically referred to as the Fault Detection, Isolation, and Recovery (FDIR) system.

Many modern systems rely on a FDIR system to pre-screen measurements in applications where incorporation of erroneous data may significantly impact performance or safety. FDIR techniques are useful in a wide variety of applications, but have been found to be particularly important for the aerospace industry [24]. In the past few decades this industry has seen significant advancements in autonomous and robotic missions, such as Unmanned Aerial Vehicles (UAVs) and robotic space probes. These advancements in autonomous missions cultivates the need for autonomous FDIR architectures that can be used in a variety of on-board applications. In addition to these new robotic missions, robust FDIR architectures are also critical for crewed applications where faults must be detected and intercepted before they endanger the crew.

Therefore, a FDIR architecture will accept measurements from onboard sensors and then determine if the measurements are “good” or “bad”. If the measurement is good, it is passed on for use in downstream applications (e.g. a navigation filter). On the other hand, if the measurement is bad, the FDIR system will intercept the bad measurement and not allow it to corrupt other sensors or downstream applications. For more complex systems, it has been suggested that FDIR follows a hierarchical level approach [25]. Refer to the Fig. 48 as a general flow diagram of how FDIR is performed.

Over the years many architectures have been constructed that can perform FDIR, and the type of FDIR system that is chosen is highly dependent on the specific application. In the particular application that motivated the present work, FDIR will be

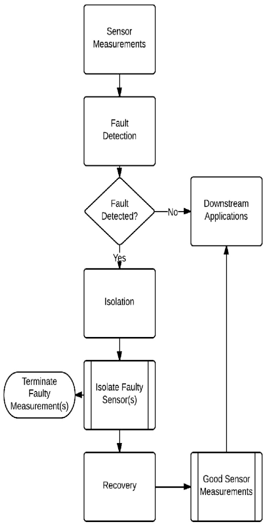


Figure 48: Simple FDIR Architecture

applied to a cluster (or “swarm”) of MEMS IMUs. The idea behind using a cluster of IMUs is that their output may be combined to achieve an effective performance greater than any one MEMS IMU can deliver [3]. Assuming simple averaging, the performance scales approximately with the inverse square-root of the number of IMUs. For this system to be useful, the data from a large number of IMUs must be fused in real-time. Thus, a real-time FDIR architecture is required to identify potentially failed IMUs within this cluster and isolate them before their erroneous measurements corrupt the fused IMU cluster output. Although many FDIR architectures are suitable for post processing applications in which all the data is available upfront, not many are equipped for real-time high frequency applications. This application requires a FDIR architecture with a fast computational time (so that measurements could be immediately processed as soon as they become available) and be robust (to detect a high percentage of faults above a specified size).

Additionally, the present work aims to create a flexible FDIR architecture that could easily be used in other applications where a sizeable amount of redundant and simultaneous measurement data is available. As was mentioned above, there are many applications for such techniques beyond just IMU clusters. For example, consider redundant pitot tubes affixed to commercial airliners. Redundancy is used throughout this application to ensure air speed measurements of the aircraft are valid and the sensors are performing as expected. Real-time FDIR is in place to ensure each pitot tube(s) are not outputting erroneous measurements. As a useful application, FDIR could have been used to prevent the catastrophic flight of Austral Lineas Aereas Flight 2553 [26], in which a pitot tube became clogged and output false readings to the flight computer. These false reading were interpreted as a low aircraft speed by the pilot, in which case was compensated by speeding up the aircraft even more. As a result, the aircraft pushed to a speed above the "never exceed speed" and caused catastrophic structural damage to the aircraft.

The motivation for this work is the use of a cluster of MEMS IMUs configured for use in a CubeSat platform. The relatively small dimensions onboard CubeSats make the mass, power, and volume constrained resources on a typical CubeSat. Further, it is these constraints that have led modern CubeSat designs to use MEMS-based IMUs. Although these IMUs are both small and cheap, their individual performance is substantially worse than tactical-grade or navigation-grade IMUs. Therefore, in order to use MEMS IMUs for CubeSat applications requiring accurate inertial navigation, the IMU cluster concept was developed to theoretically show how accuracy can be increased by adding multiple redundant IMUs [3]. Redundant systems are also important as back-up systems and for failure detection purposes [27]. Redundant (or partially redundant) measurements provide the data necessary to perform internal consistency checks and reject data from sensors that are not in agreement with the majority.

MEMS IMUs tend to have much larger noise than tactical-grade IMUs. Of particular note here, in some cases the noise can complicate the FDIR process. This issue has led the team to perform careful parametric studies of FDIR algorithm performance for IMU clusters.

Real-time IMU cluster applications require that the FDIR architecture have the following properties:

- Be computationally efficient with low overhead, so that IMU measurements produced at high frequency can be processed in real-time.
- Be capable of detecting a high percentage of faults.
- Be able to detect multiple faults simultaneously
- Isolate faulty sensors and prevent their outputs from being incorporated into the swarm solution.
- Be able to track past sensor faults to distinguish a single bad measurement from a permanently failed sensor. In event of a complete sensor failure, block all future measurements from that sensor.
- In the event of fault, be able to produce a valid overall measurement by using other simultaneous sensors in the clusters.
- In the event of total failure, supply the system with a back-up route or place into a safe mode.
- Be completely autonomous for use in a various number of applications.

With these requirements in mind, a FDIR approach was selected based on the k-th nearest neighbors algorithm. What follows is a discussion of the various FDIR techniques considered, a detailed description of the specific technique selected, and simulation results.

7.2 Survey of FDIR Architectures

There are multiple techniques used to perform FDIR in many modern applications. Current architectures such as classification, clustering, statistical, information theoretic, and spectral based approaches are used for the fault detection portion of FDIR [28]. For this particular application, the nearest neighbor based approach was chosen because of its ease of implementation in real-time. However it is important to understand each technique so that the proper approach can be chosen for any particular application. A brief explanation, along with advantages and disadvantages of each present fault detection technique, is presented here.

The techniques listed below are just few of the many existing architectures of FDIR currently in use. A few others, such as the Manual Technique and the Built in Test (BIT) technique [29] may be of interest depending on the speed of the application. These two constructs fall more into the isolation aspect of FDIR and often rely on human interaction to be implemented on high level systems. Implementing them on low level systems in a fast environment would be challenging because of their manual flow.

It is impossible to generically define a specific FDIR architecture as the "best" because they are specific to the type of application and each have their own set of advantages and disadvantages. Because of this, it is important to understand the requirements of the application, and choose several accordingly. It is then of interest to test each architecture (based on the application) and judge each based on performance (robustness, computational speed, detection percentage). Only then should one be chosen for the application.

7.2.1 Classification Based

The classification based technique [30, 31] is a learning based architecture in which a system will "learn" the measurements characteristics. This architecture, with the help of training data, will create a set of good and bad classes that each data point will lie within (first step). The classes would be determined by the user and specific to the particular application at hand. Then, each measurement from a sensor is fit to the set of classes (in our case “good” and “bad”) and the measurement is assigned an anomaly score (second step) [28]. Anomaly detection occurs by observing which measurements have bad anomaly scores (fall within bad classes) and only passing along the measurements with a good anomaly scores (fall within good classes).

As an advantage, this technique learns from data that is specific to a particular problem, and therefore classes can be closely calibrated if the test data accurately captures the difference between good and bad measurements. In addition, each measurement will have its own class, therefore it is easy to include or discard a measurement based on class.

There are two main disadvantages with this approach. First, the system must have access to a complete and representative data set up-front. Second, this approach is mainly targeted towards applications where the measurements are relatively consistent throughout the mission. This approach is not well suited for dynamic systems (such as IMUs) where the sensor output can vary over a wide range simply due to normal vehicle motion.

7.2.2 Clustering Based

Clustering based FDIR techniques cluster data into like data instances. Clustering [32, 30] relies on the assumption that good measurements will belong to a cluster, while bad measurements will not belong to any cluster [28]. Clustering in this way can be completed with many different clustering algorithms, such as Shared Nearest Neighbor (SNN) [33], Density-based spatial clustering of applications with noise (DBSCAN) [34], and Robust Clustering Algorithm for Categorical Attributes (ROCK) [35].

Of particular interest within this class of FDIR techniques, is the k-th nearest neighbor algorithm [36]. The k-th nearest neighbor algorithm is mostly used for classification purposes, however it is possible to use it in estimation and prediction applications [37]. For this application, data points will not be fit to a class as in historical classification methods, because saving the information to create classes would take up valuable memory space. Also, trying to fit highly variant data (as in this application, where speeds could change instantly) to a class could be detrimental to the operation. Therefore, classification in this instance, is a pass or fail type architecture based on a specified tolerance. In this technique, a particular data instance (Xn) is said to be the k-th nearest neighbor of another data instance (x), if (x-Xn) produces the k-th smallest distance in the set of data instances [38]. In this approach, the distance between the measurement of interest and its k-th nearest neighbor is computed. If this is less than the specified tolerance, then the measurement has at least k neighbors and is considered good. Otherwise it is considered bad. All good measurements are then allowed to pass to later algorithms and all bad measurements are discarded. There are at least two types of nearest neighbor based methods for detecting outliers or potentially faulty measurements; 1.) using the distance of a measurement to it's k-th nearest neighbor and 2.) computing the relative density of other measurements around the candidate measurement [28]. Either approach is valid, but determining a combination of k and the tolerance is selectable. It is known that choosing a value for k is not necessarily an obvious process. There can be consequences for choosing too small, too large, and in between values of k based on the application [37]. Therefore it is essential to simulate the data from an application and test different values of specific parameters such as (n) and (k). This type of approach was chosen as the most effective architecture and is explained in further detail in a later section.

Some disadvantages of clustering based fault detection is that the cluster must be predefined. In the case of the k-th nearest neighbor algorithm, the user must choose both k (the number of points that are considered “neighbors”) and N (the allowable tolerance for a neighbor). Choosing these values can be a challenge, especially with very noisy data, and is the motivation for some of the studies presented here.

7.2.3 Statistical Based

Statistical based FDIR techniques are based on the notion that good measurements lie within a high probability region and bad measurements (anomalies) lie in a low probability region in a distribution. This type of technique will fit a statistical model (frequently a normal distribution) to some known data set. Then a statistical inference test is applied to see if a particular measurement is expected to occur with a sufficiently high probability. If the measurement has a relatively low probability of fitting the model, then the measurement is considered bad. In general there are two types of statistical based anomaly detection; Parametric (knowledge of the underlying distribution)[39] and Non-Parametric (no known knowledge of the underlying distribution) [40]. Although statistical based methods are excellent for analyzing data, they are only of use in certain applications.

A disadvantage of this technique is that there must be prior knowledge or assumptions about the distribution to which the data is being fit. In reality, the data may not fit the chosen distribution and, therefore, knowing whether the data is good or bad can be difficult. As with the classification based FDIR techniques, the statistical based approach also suffers difficulty in handling measurements for some types of dynamic systems. Further, studies suggest that it is not a straightforward task in choosing the best test statistic to be applied that can detect anomalies [28]. Statistical based methods are of interest when learning about the distribution of a particular application, but using them to help find anomalies especially for this application could be quite challenging and was not selected.

7.2.4 Information Theoretic

Information theoretic FDIR techniques rely on the assumption that bad measurements will cause an irregularity or disruption in the content of the measurement data set. Many techniques have been proposed to find the disruption such as Kolomogorov Complexity, entropy, and relative entropy [41]. These techniques will find disruptions in a data set, and the disruption will (theoretically) coincide with the location of the bad measurement. Such algorithms often run in an unsupervised environment. The advantages of this type of architecture is that it may run in a non-supervised setting and it does not make assumptions about the distribution of the data. Therefore, trying to impose distributions that do not fit the actual distribution of the data set will not be an issue.

However, because this type of technique is computationally very complex, it can be quite slow. Given the high frequency at which IMU data is collected, this approach may not be practical. In addition, this technique may not be tolerant of applications where the data is highly variant. For example, if the vehicle performs a maneuver after a period of inactivity this technique may classify the change in sensor output as a fault.

7.2.5 (P-k)tet Technique

This (p-k)tet technique[27] identifies a failed sensor by observing a group of sensors within a cluster. The user is to determine a value of k (number of simultaneous failures to be identified) and a value of p (number of sensors in the cluster). The larger the cluster (value of p), the more sensor failures one can detect (value of k). From these two parameters a (p-k)tet of instruments may be formed. For example, if there are 6 sensors in a cluster(p), and you are looking to identify 1 failed instrument (k), then a quintet (set of five sensors) should be formed within the 6 sensors of the cluster. There will be six quintets within a set of 6 sensors that one must identify. Note that each (m)tet is made up of m (m-1)tets. Therefore, suppose we identify the minimum number of sensors we need to achieve redundancy – a value that varies depending on sensor configuration and degrees-of-freedom. Let this minimum number be n. For each possible (n)tet we can assess internal consistency and label each (n)tet as "clean" or "dirty". Once this is done, we label any (n+1)tet containing a dirty (n)tet as dirty, and (n+1)tets containing only clean (n)tets as clean. This procedure is repeated until "clean" and "dirty" assignments have been given to all possible (p-k)tets. The (p-k)tets that are labeled "clean" contain only good measurements. The main disadvantage of this technique is that it is only possible to discover k-number of simultaneous faulty measurements. If there are more than k simultaneous faulty measurements, then bad measurements could pass by this isolation and fault detection technique. However, some knowledge of system performance (i.e. knowledge of the maximum possible simultaneous failures) could address this disadvantage.

7.3 FDIR Design for Cluster Application

Although many of these techniques are of interest for data mining, they may not be appropriate for real-time systems with dynamic measurements. Since the motivating application intends to use measurements from a cluster of IMUs, the data must be processed in real-time. The FDIR solution used here will only be able to process the number of measurements that the cluster provides at that particular time. For example, if there are 16 IMUs in the cluster, then the FDIR architecture will only be able to work with the current 16 measurements to complete the tasks. The k-th nearest neighbor technique was chosen to accomplish the FDIR task for this application.

In the k-th nearest neighbor technique, the difference (or distance) is computed between all measurements in a time instance. This distance is calculated many times to determine how close a particular measurement is to its nearest neighbors. Depending on the tolerance that is set by the user, a measurement is identified as bad (far-away from all other measurements) or good (close-by other measurements).

In this approach, the user determines a tolerance (N) and k-value that will be used throughout the architecture. For a particular measurement to be considered “good,” at least k other measurements must have values within the tolerance N. If there are not at least k other data points within a distance of N (i.e. the present measurement does not have k neighbors), then that specific measurement is identified as “bad.” The difficulty with this technique is how the tuning parameters k and N are chosen.

This type of technique has many advantages. The most important being that it can be implemented in real-time. Although the computational overhead can be somewhat complex, it can be made quite fast. In addition, another advantage is that this type of architecture makes no assumption on the specific distribution of the data.

The main disadvantage of this technique is that there must be a large amount of data with a sufficient number of “good” measurements to identify the “bad” measurements within the cluster. In applications with a low number of measurements (e.g. three IMUs) this technique would be difficult to implement because knowledge of a “bad” measurement would only be based on the distance from two others, which is a poor evaluation of the measurement accuracy. Having more redundant measurements available makes it easier to reliably detect a potentially failed IMU.

In the nearest neighbor techniques, implementing isolation is straightforward. The structure of nearest neighbor algorithm is utilized in a such a way to specifically detect which sensor produced the fault. Therefore, fault detection and isolation are performed within the same step.

7.4 Simulation

In order to analyze the nearest neighbor and (p-k)tet techniques for the IMU cluster application, it is important to understand the performance and behavior of each approach. Two simulations were constructed to assess this performance. Due to the random nature of measurement errors and failures, a Monte Carlo (MC) framework was used to assess performance under an array of different conditions and failure scenarios for each technique. A detailed analysis of the performance of these FDIR techniques will be shown in the results section.

7.4.1 Monte Carlo Measurement Simulation for K-th Nearest Neighbor

A 15,000-run Monte Carlo analysis was performed on a cluster of 16 IMUs. Although three-axis IMUs offer a measurement for each axis, the simulation only considers 16 scalars to simulate the independent processing of measurements for a particular axis. This scalar technique could then be replicated on the other two axes. For the creation of the simulated measurements, the true value was randomly corrupted by zero-mean noise with a Gaussian distribution. The noise on the 16 simultaneous measurements are assumed to be uncorrelated. Once the cluster measurements are created, 1-3 measurements are tainted by adding a random failure of varying size. This cluster of 16 measurements are then evaluate by the FDIR technique.

The performance of the FDIR architecture is based on whether the system can detect the injected failure(s). Performance is assessed for various FDIR settings (choices for k and N), failure sizes, and number of failures. The MC analysis is performed for each of these configurations. For each case, the simulation records the percentage of times that all bad measurements were identified and the percentage of times that all good measurements were identified. If not all bad measurements are identified, then at least one bad measurement was mislabeled as good. By failing to intercept this bad measurement, downstream algorithms may be corrupted by using erroneous data. Alternatively, if not all good measurements are identified, then the FDIR system accidentally discarded potentially good data.

7.4.2 Fault Detection K-th Nearest Neighbors

In the k-th nearest neighbor technique, a distance is equated between two points relative to the measured value of the two measurements. Since the application involved a cluster of IMUs the difference between two gyro measurements or two accelerometer measurements would be taken, depending on which measurements are of interest. The first requirement of the k-th nearest neighbor is to determine a k-value (number of nearest neighbors needed to be considered a good measurement). The k-value chosen depends on the application but, for this instance 25% of the cluster is selected for the value of k (25% of 16 IMUs gives k=4). Meaning that for the simulation, a particular sensor data instance will need at least 4 neighbors closer than the tolerance, N, in order to be considered a good measurement. If it does not have the required number of nearest neighbors then it is likely too far from all other data. The k-value was also changed (from 3-5), to observe how performance and the tuning parameters are affected.

To start the algorithm, the distance is evaluated between each measurement in the IMU cluster. Since there are 16 IMUs in the cluster, there will need to be 120 distance measurements in order to fulfill each combination of two sensor measurements. Once each distance is computed, a tolerance is defined to determine whether a distance is satisfactory or not. Again, choosing this tolerance is application based. In this study, the tolerance was varied from 1σ to 6σ (where σ is the IMU measurement standard deviation) in order to show how the detection percentage contours change with an increase in tolerance. Any distance measurements inside of the tolerance, would be labeled as a good measurement. Each IMU that was involved to calculate that distance would then get 1 strike against it. Each distance is evaluated to see whether it falls within the tolerance or not, and the strikes per each IMU are totaled.

Once the strikes are totaled for each IMU, they are compared to the k-value determined at the beginning. If the number of strikes exceeds the k-value then measurement from that particular IMU is considered of value. If the number of strikes for an IMU falls below the k-value, then that measurement is likely not close to other data points in the cluster and should not be considered. In this simulation, the program looks for distance instances inside of the tolerance, out of preference. It is possible to determine distance instances outside a tolerance, in which case at least (#IMUs-k) strikes would be needed for a good measurement. All data from the IMUs that are considered of value are allowed to pass into downstream algorithms. All data from IMUs that are not considered of value will be rejected at that time instance. See Fig. 49 below for a flowchart of the program process.

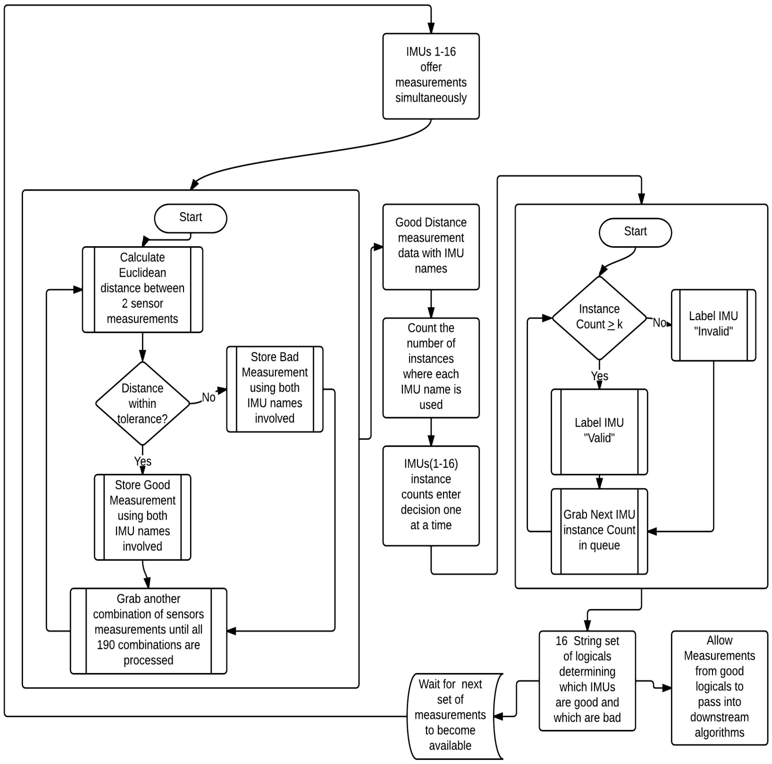

IMUs 1-16 offer measurements simultaneously
Start
Calculate Euclidean distance between 2 sensor measurements
Distance within tolerance?
No
Store Bad Measurement using both IMU names involved
Yes
Store Good Measurement using both IMU names involved
Grab another combination of sensors measurements until all 190 combinations are processed
Good Distance measurement data with IMU names
Count the number of instances where each IMU name is used
IMUs(1-16) instance counts enter decision one at a time
Start
Instance Count ≥ k
No
Label IMU "Invalid"
Yes
Label IMU "Valid"
Grab Next IMU instance Count in queue
16 String set of logicals determining which IMUs are good and which are bad
Allow Measurements from good logicals to pass into downstream algorithms
Wait for next set of measurements to become available

Figure 49: Flow Chart of FDIR Process

7.4.3 Monte Carlo Measurement Simulation for (p-k)tet

As a comparison, a (p-k)tet like simulation was constructed to evaluate the performance of this architecture against the k-th nearest neighbor. The same Monte Carlo framework is used to evaluate the (p-k)tet algorithm as was used for the k-th nearest neighbor technique. The value of k is the maximum number of simultaneous failures to occur in the array of measurements, and the value of n is a tolerance between only two measurements. The value of p is the number of simultaneous IMU measurements in the cluster. Using only these changes to the input parameters, a (p-k)tet architecture was constructed and placed into the simulation in the same manner as the k-th nearest neighbor technique.

7.4.4 Fault Detection (p-k)tet

To start this approach, all possible combinations of pairs within the 16 measurements is computed. The absolute difference between the pairs is then evaluated to find the distance between two specific measurements. For example, 16 simultaneous measurements will have 120 combinations of pairs, and therefore 120 distances are computed. These measured distances are then compared to the tolerance. If the distance that a pair makes is above the tolerance it is considered a bad pair and it is placed into a known bad pair matrix. On the other hand, if a pair creates a distance that is below the tolerance it is considered a good pair and it is placed into a good pair matrix.

With all of the good and bad pairs now acquired, the (p-k)tet combinations must be formed. The value of p and k are up to user and specific to the application. For this application the number of measurements (p) is 16 and number of simultaneous failures (k) is varied from 1 to 3. When the value of k is 1, then all combinations of (15)tets is formed. There are 16 different (15)tets with 16 sensor measurements. Once these (p-k)tets are created, they are then evaluated to determine if any bad pairs acquired in the previous step are a member of that specific (p-k)tet. If a bad pair is found within the p-k(tet) then the p-k(tet) is labeled as “dirty”. It is expected that all pair combinations that contain the IMU with the injected failure will be considered failed. Therefore any (p-k)tet containing a pair with that injected failure is considered “dirty” (all (p-k)tets except for 1). With one failure there will be one missing (p-k)tet from the “dirty” combinations of 15(tets). This missing (p-k)tet is the only good (p-k)tet because it will not contain the failure that was injected during the Monte Carlo Simulation. Therefore, the last procedure in the (p-k)tet architecture is finding the good or “clean” (p-k)tet. Once this is found, all measurements within the clean (p-k)tet are passed to downstream algorithms while all other IMUs measurements that are not included in the clean (p-k)tet are discarded. When there are 2 and 3 failures (k=2,k=3), then a different number of (p-k)tets is generated, but they are still evaluated in the same way as single failure case. The only difference is that now, there will be more “dirty” (p-k)tets to be analyzed, which could lead to a slower computational time.

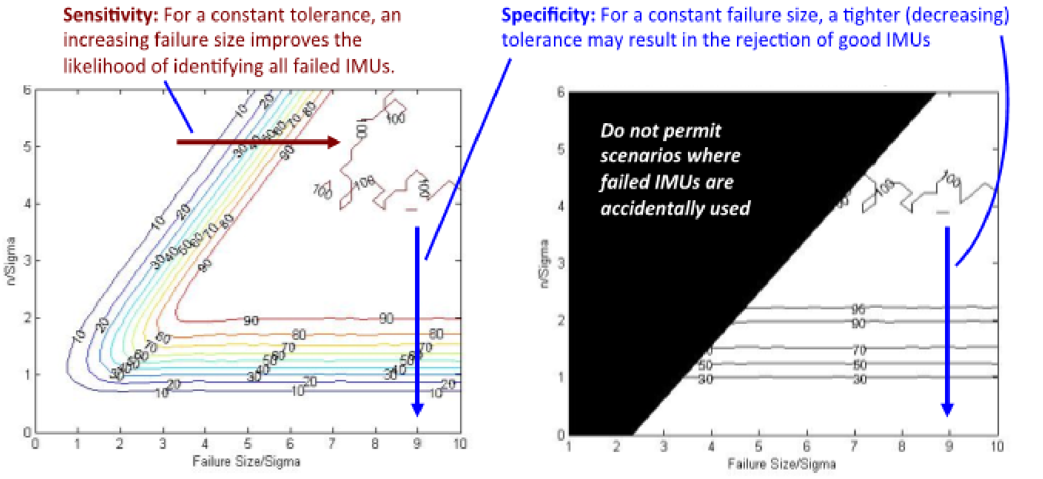
7.5 FDIR Results and Discussion

The results from the Monte Carlo simulation are presented in terms of detection percentages. This detected percentage is defined as the percentage of injected failures that were detected by the k-th nearest neighbor and (p-k)tet FDIR algorithms. The Monte Carlo outputs a detection percentage for each combination of tolerance and injected failure size.

All results within this section are presented using a common format. For each case, two contour plots are provided (with the first example being shown in Fig. 50). The left plot contains contours of the percentage of MC runs where perfect classification is achieved. Here we define perfect classification as identifying all good IMUs as good and all failed IMUs as bad. It should be noted, however, that perfect identification is not always required. What is most important is that no failed IMUs are accidentally allowed to pass their data to downstream algorithms where they may corrupt navigation solutions. Therefore, the plot on the right shades out the region where any failed

IMUs are accidentally used. What remains are percentage contours of the percent of good IMUs to send to downstream applications. A value of less than 100% means that some good IMUs were sometimes labeled as bad, and thus not all available valid data is always being used. Depending on the size of the IMU cluster, it may be acceptable in some situations to design a system with less than 100% specificity. Having a large number of redundant measurements means that a few valid measurements may be discarded and many good measurements will remain for filtering. Lower tolerances result in tighter data distributions. On the other hand, going too low on this known good contour axis (30%-70%) will result in lower performance from the IMU cluster. At these percentages, the algorithm will discard a large percentage of good data. It is especially important to have the maximum number of good measurements so that the performance of the cluster is at its peak.

For Fig. 50 the nearest neighbor construct was evaluated for three nearest neighbors (k=3) and only one failure was injected into the cluster of 16 measurements per time instance. Different values of k and different numbers of failures will change these contours.



Sensitivity: For a constant tolerance, an increasing failure size improves the likelihood of identifying all failed IMUs.

Specificity: For a constant failure size, a tighter (decreasing) tolerance may result in the rejection of good IMUs

Do not permit scenarios where failed IMUs are accidentally used

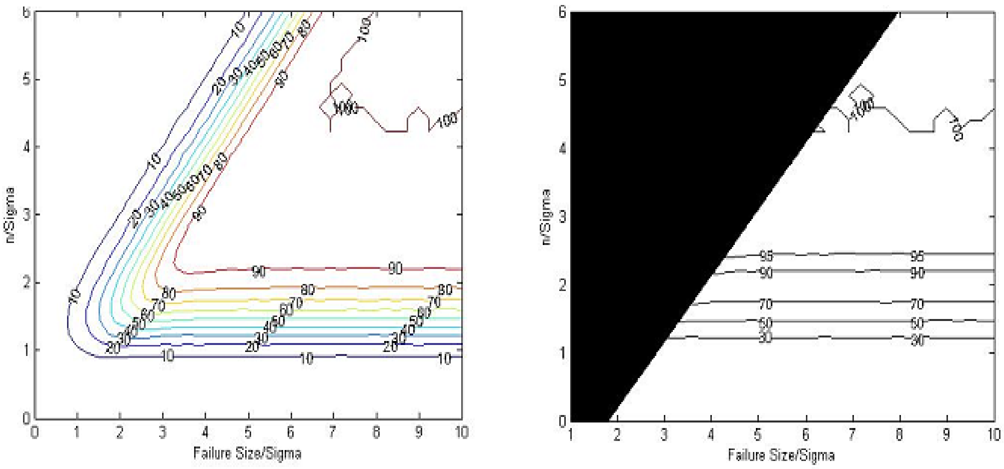
a) Contours of percentage of MC runs where perfect classification is achieved.

b) After blocking out the region where any failed IMUs are accidently used, the remaining contours show the percentage MC runs where all of the good IMUs are used.

Figure 50: MC Results for k-th nearest neighbor, k=3, 1 failure

In order to show how these contours change, Fig. 51 and Fig. 52 were created with different k-values. Figure 51 shows the charts that were created when four nearest neighbors is required (k=4) and Fig. 52 shows the charts that were created when five nearest neighbors are required (k=5). Although there is not a great difference between each set of charts, it is possible to see that the excluded region in the right-hand plots (black shaded area) moves towards the left as k increases. This indicates that the shaded region is decreasing in size, and the non-shaded region (area of acceptable scenarios exist) is increasing in size. It should also be observed that the contours of specificity appear to shift upwards in the plot. This indicates that in any given scenario increasing k will cause more good measurements to potentially be unnecessarily discarded. Increasing k improves the chances of detecting all the bad measurements, but also increases the chances of mistakenly discarding good measurements. Thus, there is a limit to the highest k-value that can reasonably be chosen. Having too many nearest neighbors will cause the k-th nearest neighbor algorithm to become too strict depending on the tolerance. A tight tolerance with a high number of nearest neighbors will most likely result in the entire cluster of measurements to be classified as “failed,” leaving no data for use in subsequent applications. Therefore, it is important to choose a k and n value that will satisfy the requirements of the application of interest, but not be so strict to where the program has “zero tolerance”. These graphs can help determine these two variables of interest.

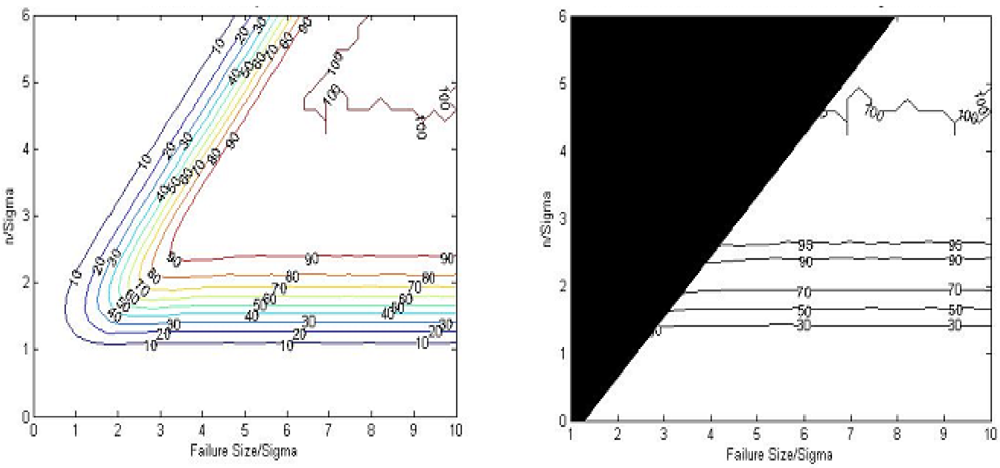
Another important variable to consider is the number of failures that are injected into the simulation. The scenarios discussed above (as shown in Fig. 50, Fig. 51, and Fig. 52) are all from simulations where only one failure was injected at each time instance. Thus, the next logical step is to consider how the algorithm responds to a varying numbers of simultaneous failures. This will help assess the robustness of the k-th nearest neighbor algorithm, as a robust algorithm will be capable of detecting a varying number of failures. To test this, two more simulations were performed that



a) Contours of percentage of MC runs where perfect classification is achieved.

b) After blocking out the region where any failed IMUs are accidently used, the remaining contours show the percentage MC runs where all of the good IMUs are used.

Figure 51: MC Results for k-th nearest neighbor, k=4, 1 failure



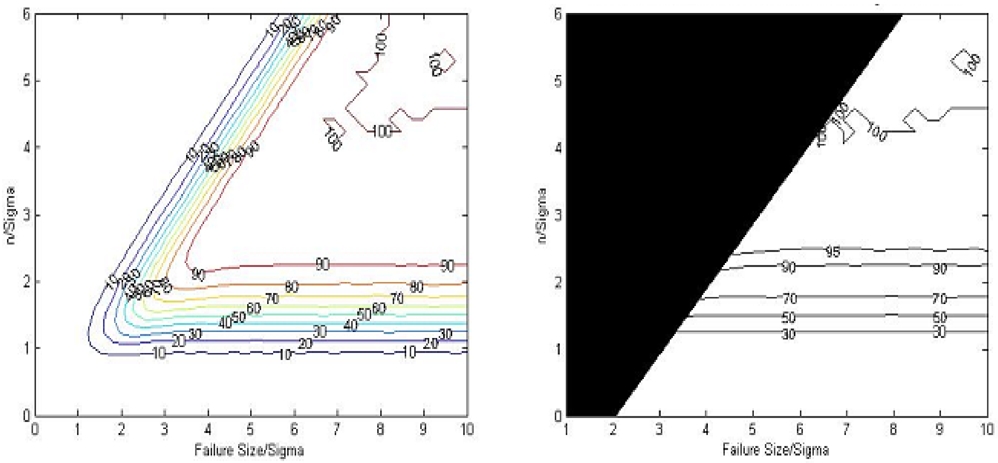
a) Contours of percentage of MC runs where perfect classification is achieved.

b) After blocking out the region where any failed IMUs are accidently used, the remaining contours show the percentage MC runs where all of the good IMUs are used.

Figure 52: MC Results for k-th nearest neighbor, k=5, 1 failure

create measurements in the same way, but injected two failures (Fig. 53) and three failures (Fig. 54) at each time instance. In observing the figures it can be seen that while the sensitivity contours on the left-hand plots expand with more failures, the specificity contours remain nearly identical. Thus, there are no significant changes in the right-hand plot (which is the most important plot from a practical implementation standpoint).

Using the results, one may choose an n and k value specific to any application. Being able to choose these variables will help the user get close to a desired or required detection percentage. In table 11, a general high-level truth table, defining the outputs for the given input scenarios of the k-th nearest neighbor is given. The first two scenarios are permitted according to the cluster application. The final two scenarios however, are not permitted because of the chance that some bad IMUs may go undetected.



a) Contours of percentage of MC runs where perfect classification is achieved.

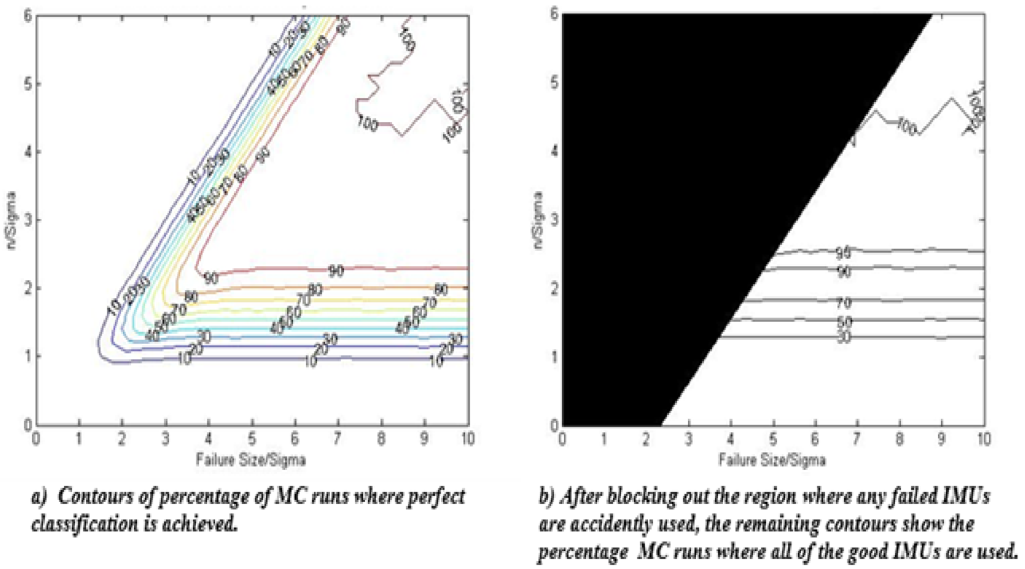
b) After blocking out the region where any failed IMUs are accidently used, the remaining contours show the percentage MC runs where all of the good IMUs are used.

Figure 53: MC Results for k-th nearest neighbor, k=4, 2 failures

Inputs		**FDIR Outcome**		
Tolerance	**Failure Size**	**All Bad IMUs Detected**	**All Good IMUs Used**	**Scenario**
High	High	TRUE	TRUE	**Acceptable**
Low-Moderate	Moderate-High	TRUE	FALSE	
High	Low-Moderate	FALSE	TRUE	**Do NOT Permit**
Low-Moderate	Low-Moderate	FALSE	FALSE	

Table 11: Input Scenario Truth Table

In addition to the k-th nearest neighbor, the performance of the (p-k)tet technique was also analyzed. The results from the MC analysis of this are shown using the same type of contour plots as before. The results for the (p-k)tet FDIR algorithm are shown in Fig. 55. For this particular situation, the (p-k)tet algorithm will attempt to detect



a) Contours of percentage of MC runs where perfect classification is achieved.

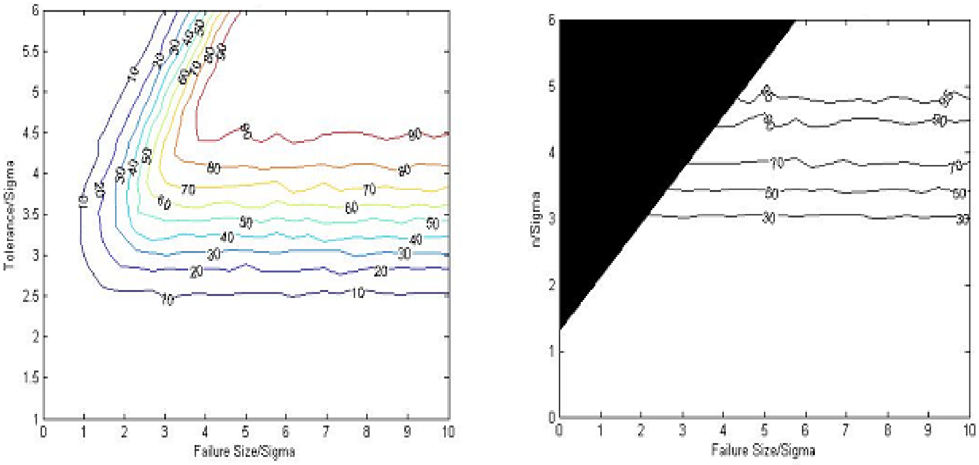
b) After blocking out the region where any failed IMUs are accidently used, the remaining contours show the percentage MC runs where all of the good IMUs are used.

Figure 54: MC Results for k-th nearest neighbor, k=4, 3 failures

one randomly injected failure in swarm of 16 measurements. From the figures, one can note that the black shaded area Fig. 55-b is much smaller than any of the k-th nearest neighbor plots, meaning that this type of technique was able to pick-up smaller failure sizes then the k-th nearest neighbor technique. However, it is also noted that there is no 100% contour line on the right-hand plot. Therefore, the search areas shown in the graphs for tolerance and failure size were not large enough for 100% detection of all good IMUs. In other words, there were always some MC runs where the data from a good IMU was inadvertently, undesirably discarded. With larger values for failure size and tolerance, a 100% contour may be found.

7.5.1 Comparison of FDIR Techniques

Both the k-th nearest neighbor and (p-k)tet techniques showed promise for real-time FDIR of a cluster of IMUs. However, the k-th nearest neighbor technique proved to be much faster and was more robust than the (p-k)tet algorithm. The (p-k)tet technique was considerably slower even when only needing to detect one failure. Further, the (p-k)tet technique was unable to produce a 100% detection contour in the



a) Contours of percentage of MC runs where perfect classification is achieved.

b) After blocking out the region where any failed IMUs are accidently used, the remaining contours show the percentage MC runs where all of the good IMUs are used.

Figure 55: MC Results for (p-k)tet algorithm, k=1, 1 failures

same search region as the k-th nearest neighbor. Lastly, in the (p-k)tet technique the user must assume a maximum number of allowable simultaneous failures. This may be difficult to know in advance. For example, if the algorithm is structured to detect 2 failures, but 3 or more failures occur in a time instance, all the (p-k)tets would be bad. In the k-th nearest neighbor technique, as long as there is a sufficient number of measurements, it can detect any reasonable number of failures. Therefore, the k-th nearest neighbor technique was chosen because of the fast computational time and robustness.

7.6 FDIR Conclusion

This chapter summarizes a FDIR architecture based on the k-th nearest neighbor and (p-k)tet techniques that are suitable for use with a cluster of MEMS IMUs. A review of other FDIR techniques is also presented to determine their viability in such an application. After careful consideration, the k-th nearest neighbor construct is determined the most appropriate technique for this application. The performance of such an algorithm under various noise and failure sizes is shown for different algorithm tuning parameters. Using these results, one may be able to select the tuning parameters (n and k) for the k-th nearest neighbor algorithm based on the desired detection percentage. As a result of this study, a tool was created that allows a user to input the parameters specific to the application of interest, and retrieve output parameters that will allow the user to reach a desired fault detection percentage for their application in real-time.

8 Conclusion

Presented in this thesis are software architectures for calibration and FDIR which will be used to help advance small satellite IMU cluster technology. Through simulation and testing, these constructs have proven to be quick and highly robust. The results help prove that technology such as this can be highly affective for high performance navigation for small satellites. While a cluster of 16 MEMS may not enter into the realm of the highest performing gyroscopes, as MEMS technology continues to mature and additional sensors are added to the PCB, performance may far exceed tactical grade. This would allow IMU clusters to enter into even more applications. Additionally, a new design for a calibration apparatus has been machined and implemented which will allow three axis measurements to be taken from only a single axis rate table. A provisional patent has been filed for this design. With this apparatus and the MLE calibration algorithm shown in the appendix, a full calibration can be performed which yields highly accurate error coefficients for each cluster sensor. Lastly, two FDIR designs were implemented on a simulated IMU cluster. The author has created a MATLAB script which creates plots that help determine the adequate input parameters for the k-th nearest neighbor and (p-k)tet techniques implemented within redundant applications. To conclude, advancements made throughout the work of this thesis have laid the ground work for future on-board calibration software and FDIR diagnostics which will help boost IMU cluster readiness for real-world applications.

References

[1] Bittner, D. E., Christian, J. A., and Bishop, R. H., "Development of An Alignment Technique for A Large Number of Redundant Inertial Measurement Units," *International Astronautical Congress 2014*, No. IAC-14-C1.5.8, West Virginia University, Toronto, Canada, October 2014.

[2] Bittner, D. E., Christian, J. A., Bishop, R. H., and May, D., "Fault Detection, Isolation, and Recovery Techniques for Large Clusters of Inertial Measurement Units," *Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION*, No. 10.1109/PLANS.2014.6851379, West Virginia University, Monterey, CA, May 2014, pp. 219–229.

[3] "The Inertial Sensor Market," VectorNAV Library, 2015, http://www.vectornav.com/support/library/imu-and-ins.

[4] "Gyro sensors - How they work and what's ahead," Gyro Sensor Portal Site, 2015, http://www5.epsondevice.com/en/sensing_device/gyroportal/about.html.

[5] Crain, T. P., Bishop, R. H., and Brady, T., "Shifting the Inertial Navigation Paradigm with MEMS Technology," No. AAS 10-043, Breckenridge, CO, February 2010.

[6] Greenheck, D. R., Bishop, R. H., Jonardi, E. M., and Christian, J. A., "Design and Testing of a Low-Cost MEMS IMU Cluster for SmallSat Applications," *28th Annual AIAA/USU Conference on Small Satellites*, No. SSC14-III-6, Logan, UT, August 2014.

[7] Skog, I., Nilsson, J.-O., and Händel, P., "An Open-source Multi Inertial Measurement Unit (MIMU) Platform," *2014 International Symposium on Inertial Sensors and Systems (ISISS)*, No. 10.1109/ISISS.2014.6782523, Leguna Beach, CA, February 2014.

[8] Chang, H., Xue, L., Qin, W., Yuan, G., and Yuan, W., "An Integrated MEMS Gyroscope Array with Higher Accuracy Output," *Sensors*, Vol. 8, No. 3, 2008, pp. 2886–2899.

[9] Fang, B., Chou, W., and Dang, L., "An Optimal Calibration Method for a MEMS Inertial Measurement Unit," *International Journal of Advanced Robotic Systems*, Vol. 14, 11 2014, pp. 1–14.

[10] Aggarwal, P., Syed, Z., Niu, X., and El-Sheimy, N., "A Standard Testing and Calibration Procedure for Low Cost MEMS Inertial Sensors and Units," *Journal of Navigation*, Vol. 61, 4 2008, pp. 323–336.

[11] Syed, Z. F., Aggarwal, P., Goodall, C., and El-Sheimy, N., "A new multi-position calibration method for MEMS inertial navigation systems," *Measurement and Science Technology*, Vol. 18, 7 2007, pp. 1897–1907.

[12] Wahba, G., "Problem 65-1: A Least Squares Estimate of Satellite Attitude," *SIAM Review*, Vol. 7, No. 3, July 1965, pp. 409.

[13] Keat, J., "Analysis of Least-Squares Attitude Determination Routine DOAOP," *Computer Sciences Corp.*, No. SCS/TM-77/6034, February 1977.

[14] Shuster, M. and Oh, S., "Three-Axis Attitude Determination from Vector Observations," *Journal of Guidance and Control*, Vol. 4, No. 1, 1 1981, pp. 70–77.

[15] Markley, F. L., "Attitude Determination Using Vector Observations and the Singular Value Decomposition," *Journal of the Astronautical Sciences*, Vol. 36, No. 3, July-September 1988, pp. 245–258.

[16] Fisher, R., "On the Mathematical Foundations of Theoretical Statistics," *Philosophical Transactions of the Royal Society Ser. A*, Vol. 1, No. 1, 1 1922, pp. 309–368.

[17] Tapley, B. D., Schutz, B. E., and Born, G. H., "Maximum Likelihood And Bayesian Estimation," *Statistical Orbit Determination*, Vol. 1, 2004, pp. 190–194.

[18] Crassidis, J. and Junkins, J., "Optimal Estimation of Dynamic Systems," *CRC Press, Boca Raton, FL* , 2004.

[19] Wang, L. and Wang, F., "Intelligent Calibration Method of Low Cost MEMS Inertial Measurement Unit for an FPGA-based Navigation System," *International Journal of Intelligent Engineering & Systems*, Vol. 4, No. 2, 2011, pp. 32–41.

[20] Lee, D., Lee, S., Park, S., and Ko, S., "Test and Error Parameter Estimation for MEMS-Based Low Cost IMU Calibration," *International Journal of Precision Engineering and Manufacturing*, Vol. 12, No. 4, 2011, pp. 597–603.

[21] "IEEE Recommended Practice for Inertial Sensor Test Equipment, Instrumentation, Data Acquisition, and Analysis," Institute of Electrical and Electronics Engineers, New York, NY, 2005, IEEE-STD-1554-2005(R2011).

[22] "Guide to Comparing Gyro and IMU Technologies - Micro-Electro-Mechanical Systems and Fiber Optic Gyros," KVH White Paper, 2014.

[23] Brown, R. and Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, Inc, New York, NY, 3rd ed., 1997.

[24] Srivastava, A. and Schwabacher, M., "Intelligent Data Understanding for Integrated Systems Heath Management (ISHM)," *CRC Press, Boca Raton, FL* , 2000.

[25] SalarKaleji, F. and Dayyani, A., "A Survey on Fault Detection, Isolation, and Recovery (FDIR) module in Satellite Onboard Software," *6th*

International Conference on Recent Advances in Space Technologies, No. 10.1109/RAST.2013.6581270, Istanbul, Turkey, June 2013.

[26] "Probe Eyes Pilot Heaters," *Aviation Week Space Technology*, Vol. 147, No. 18, 11 1997, pp. 41.

[27] Potter, J. and Deckert, J., "Gyro and Accelerometer Failure Detection and Identification in Redundant Sensor Systems," , No. E-2686, 1972.

[28] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly Detection: A Survey," *ACM Computing Surveys*, Vol. 41, No. 3, 7 2009.

[29] JSC, "Fault-Detection, Fault-Isolation, and Recovery (FDIR)," Technique dfe-7, NASA, Houston, TX, 2000.

[30] Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley, Boston, MA, 2009.

[31] Duda, R., Hart, P., and Stark, D., *Pattern Classification: 2nd Edition*, Wiley Interscience, Hoboken, NJ, 2000.

[32] Jain, A. and Dubes, R., *Algorithms for Clustering Data*, Prentice-Hall, Inc, Upper Saddle River, NJ, 1988.

[33] Ertoz, L., Steinbach, M., and Kumar, V., *Clustering and Information Retrieval*, Springer, New York, NY, 2004, pp. 83–103.

[34] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, OR, August 2014, pp. 226–231.

[35] Guha, S., Rastogi, R., and Shim, K., “ROCK: A Robust Clustering Algorithm for Categorical Attributes,” *15th International Conference on Data Engineering*, No. 10.1109/ICDE.1999.754967, Sydney, NSW, March 1999.

[36] Fix, E. and Hodges, J., “Discriminatory Analysis: Nonparametric Discrimination: Small Sample Performance,” *Technical Report Project 21-49-004*, No. Report Number 11, USAF School of Aviation Medicine, August 1952.

[37] Larose, D., *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley-Interscience, Hoboken, NJ, 2005.

[38] Devroye, L., Gyorfi, L., and Lugosi, G., *A Probabilistic Theory of Pattern Recognition*, Springer, New York, NY, 1996.

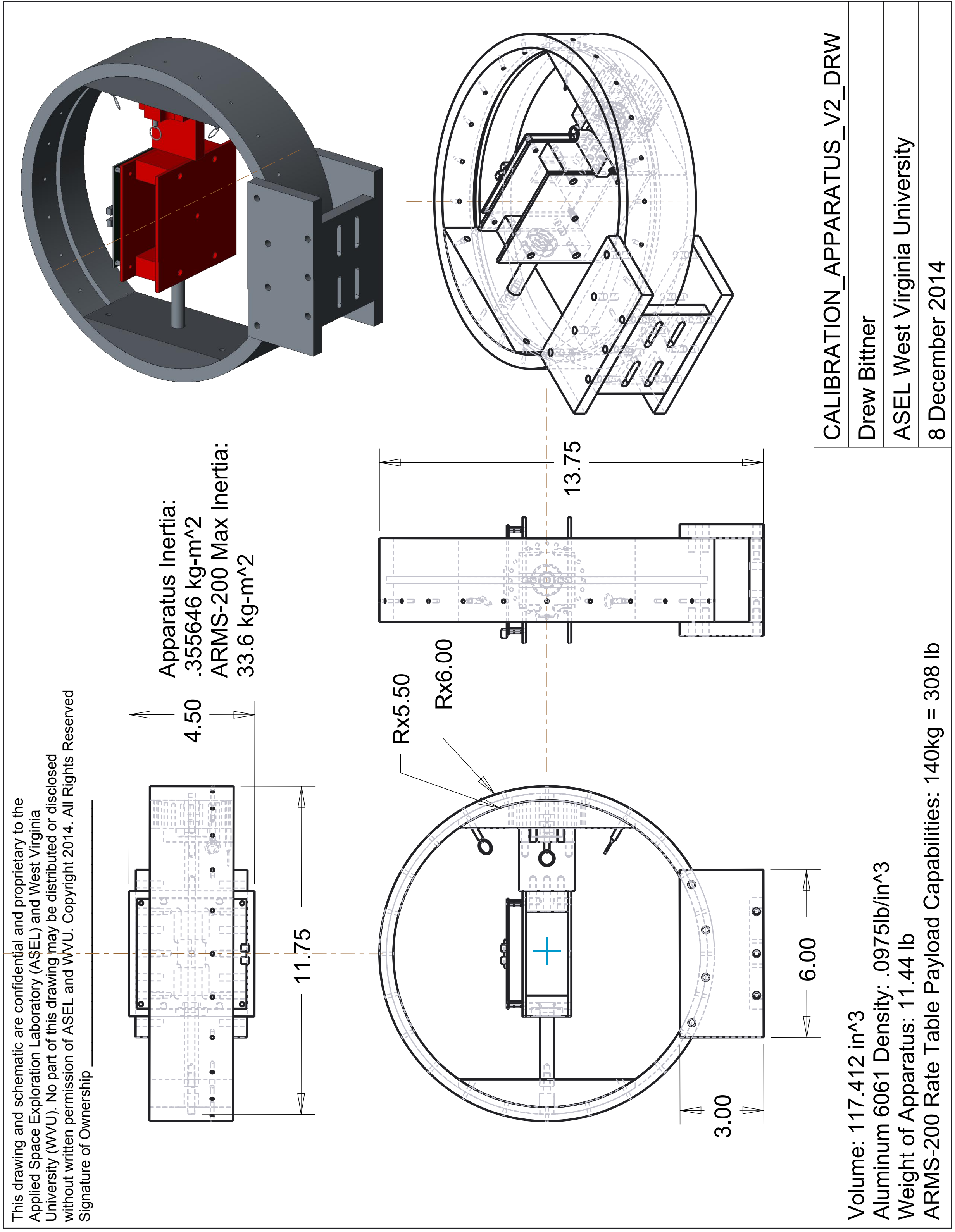
[39] Eskin, E., “Anomaly Detection over Noisy Data using Learned Probability Distributions,” *In Proceedings of the International Conference on Machine Learning*, Morgan Kaufmann, 2000, pp. 255–262.

[40] Desforges, M., Jacob, P., and Cooper, J., “Applications of Probability Density Estimation to the Detection of Abnormal Conditions in Engineering,” *Journal of Mechanical Engineering*, Vol. 212, 1998, pp. 687–703.

[41] Li, M. and Vitanyi, P., *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, Verlag, Berlin, 1993.

9 Appendix

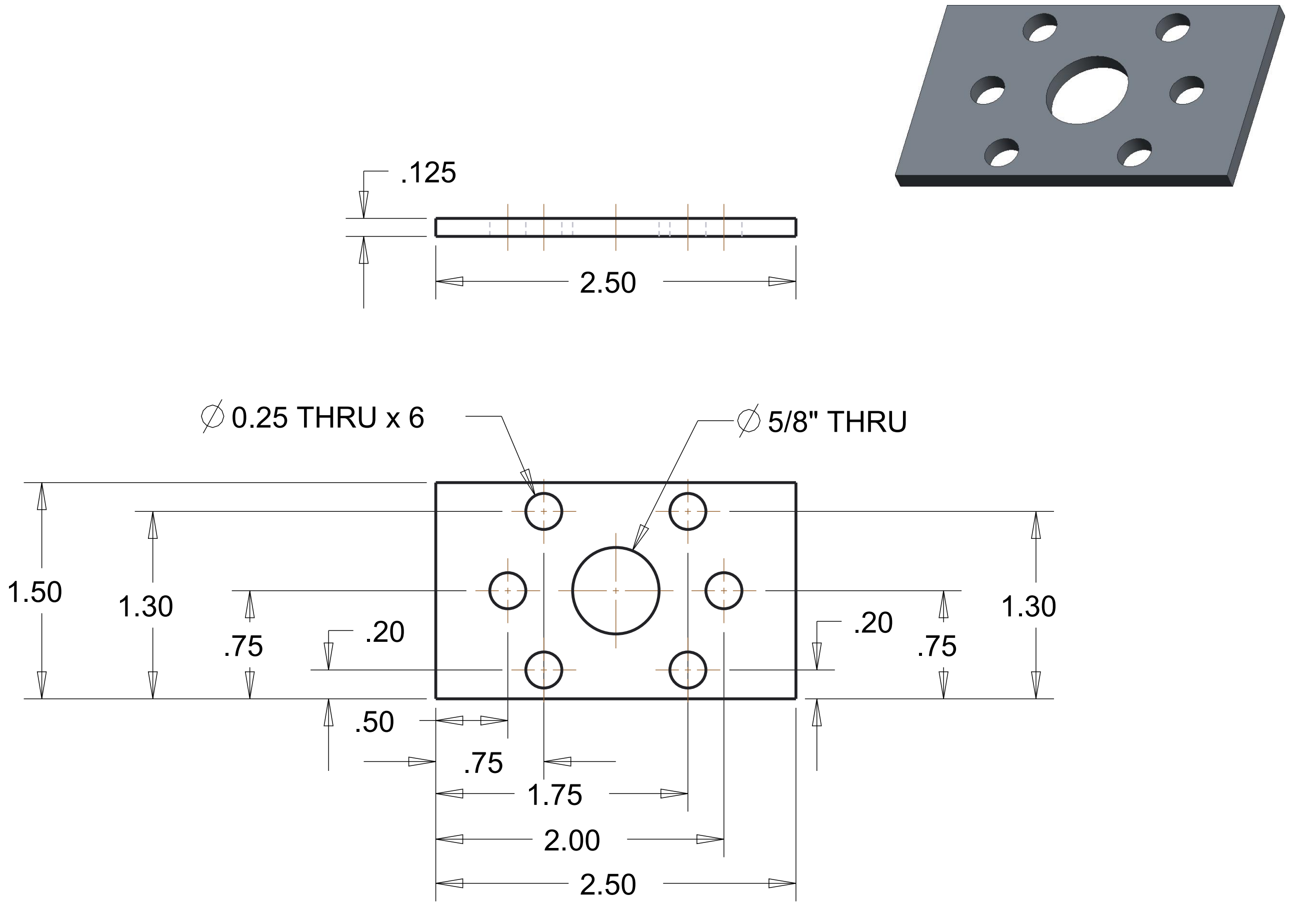
9.1 Patent-Pending Calibration Apparatus Drawings



Volume: 117.412 in^3
Aluminum 6061 Density: .0975lb/in^3
Weight of Apparatus: 11.44 lb
ARMS-200 Rate Table Payload Capabilities: 140kg = 308 lb

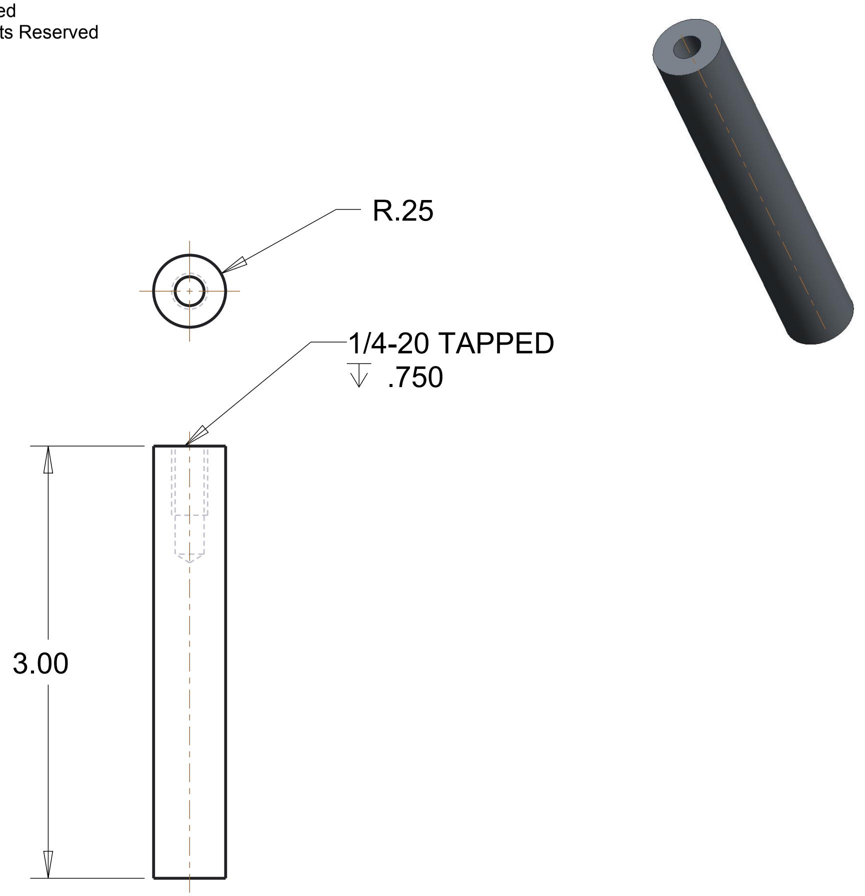
This drawing and schematic are confidential and proprietary to the Applied Space Exploration Laboratory (ASEL) and West Virginia University (WVU). No part of this drawing may be distributed or disclosed without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ____________

This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ________________________________



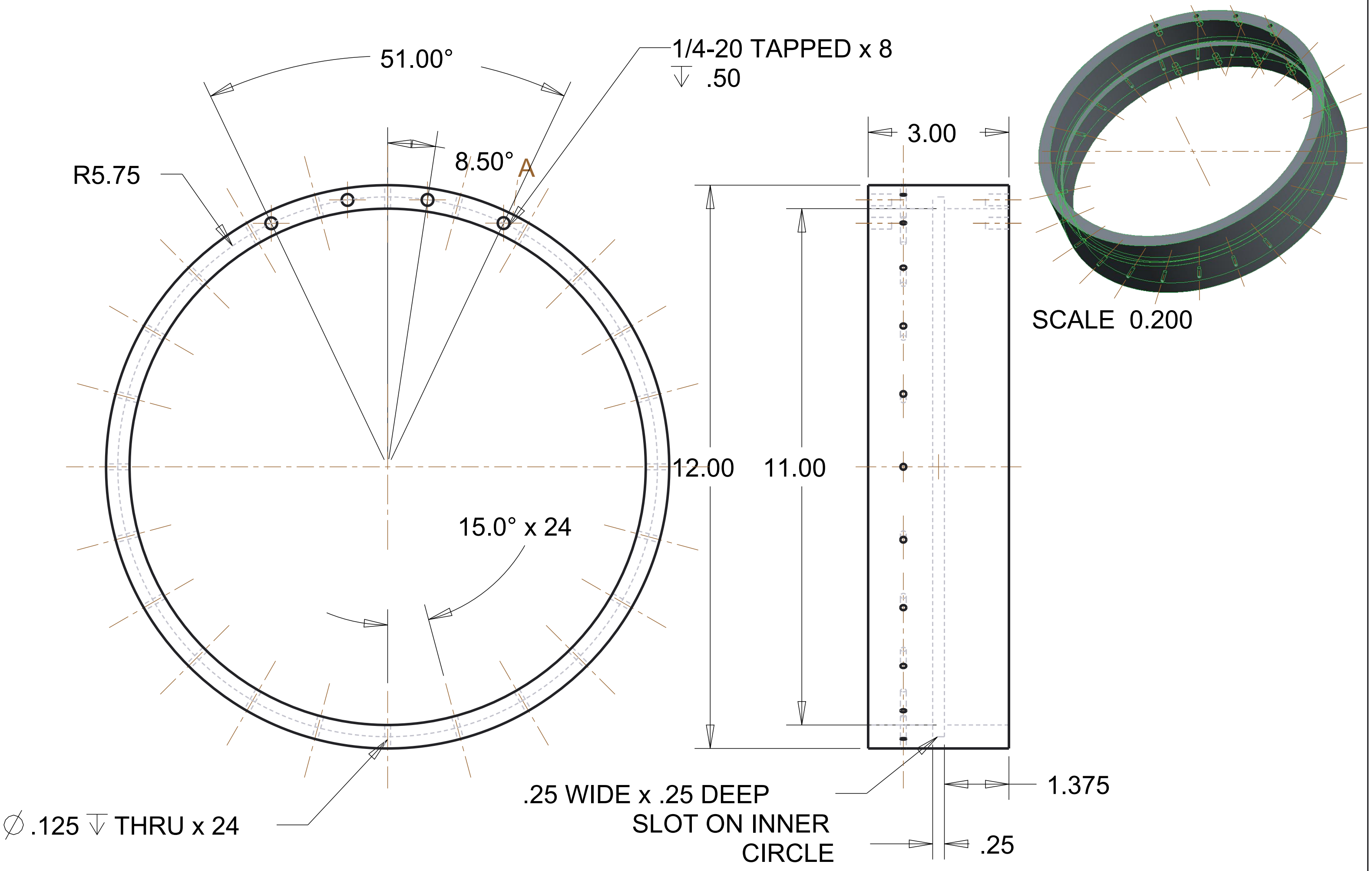
General Purpose 6061 Aluminum

BEARING_RETAINER_DRW
Drew Bittner
ASEL West Virginia University
23 JANUARY 2015



General Purpose Aluminum 6061

BEARING_SIDE_ROTATION_ARM_DRW
Drew Bittner
ASEL West Virginia University
2 December 2014

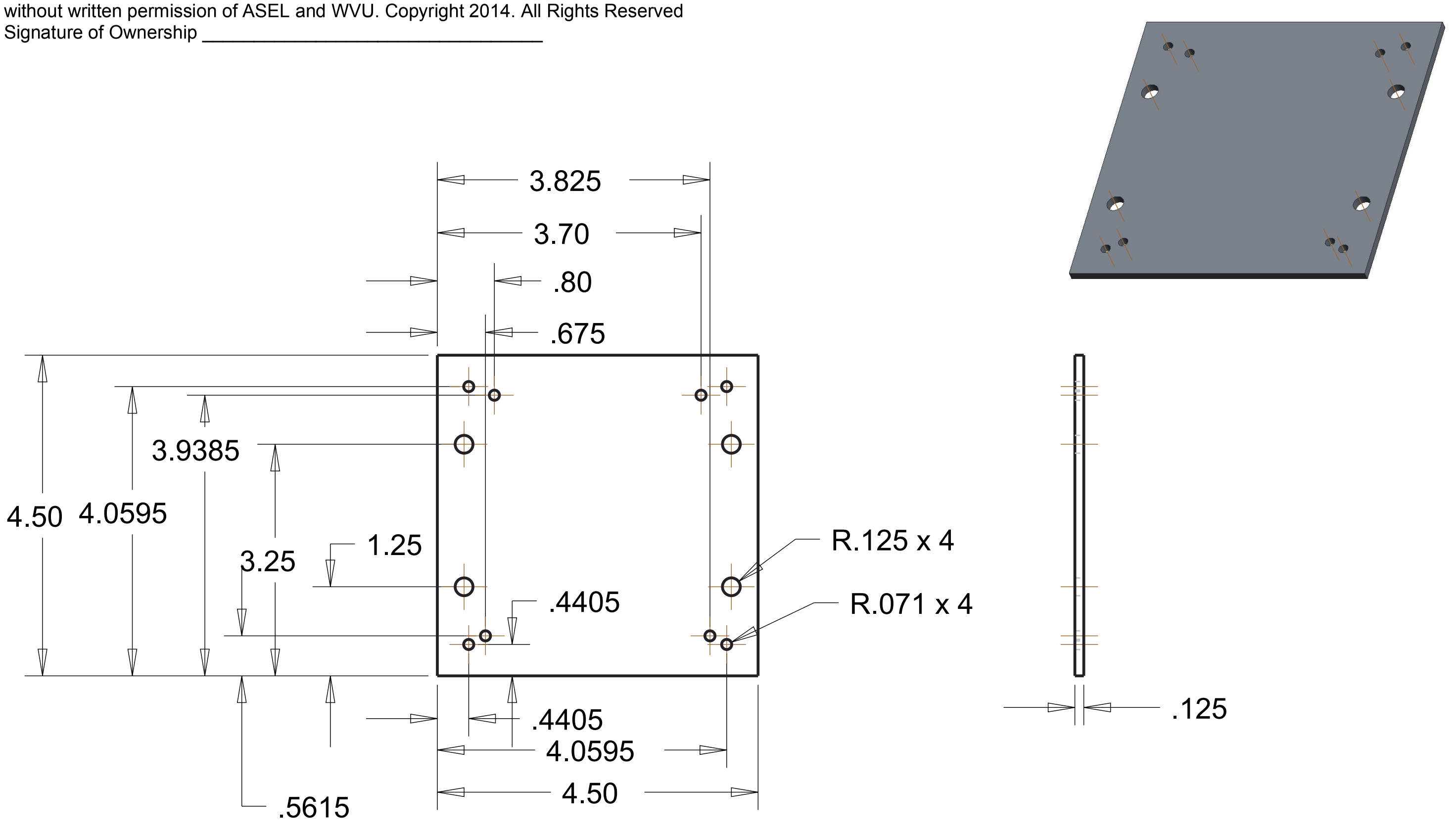


General Purpose Aluminum 6061

This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ________________________________

CIRCULAR_MOUNT_DRW
Drew Bittner
ASEL West Virginia University
2 December 2014

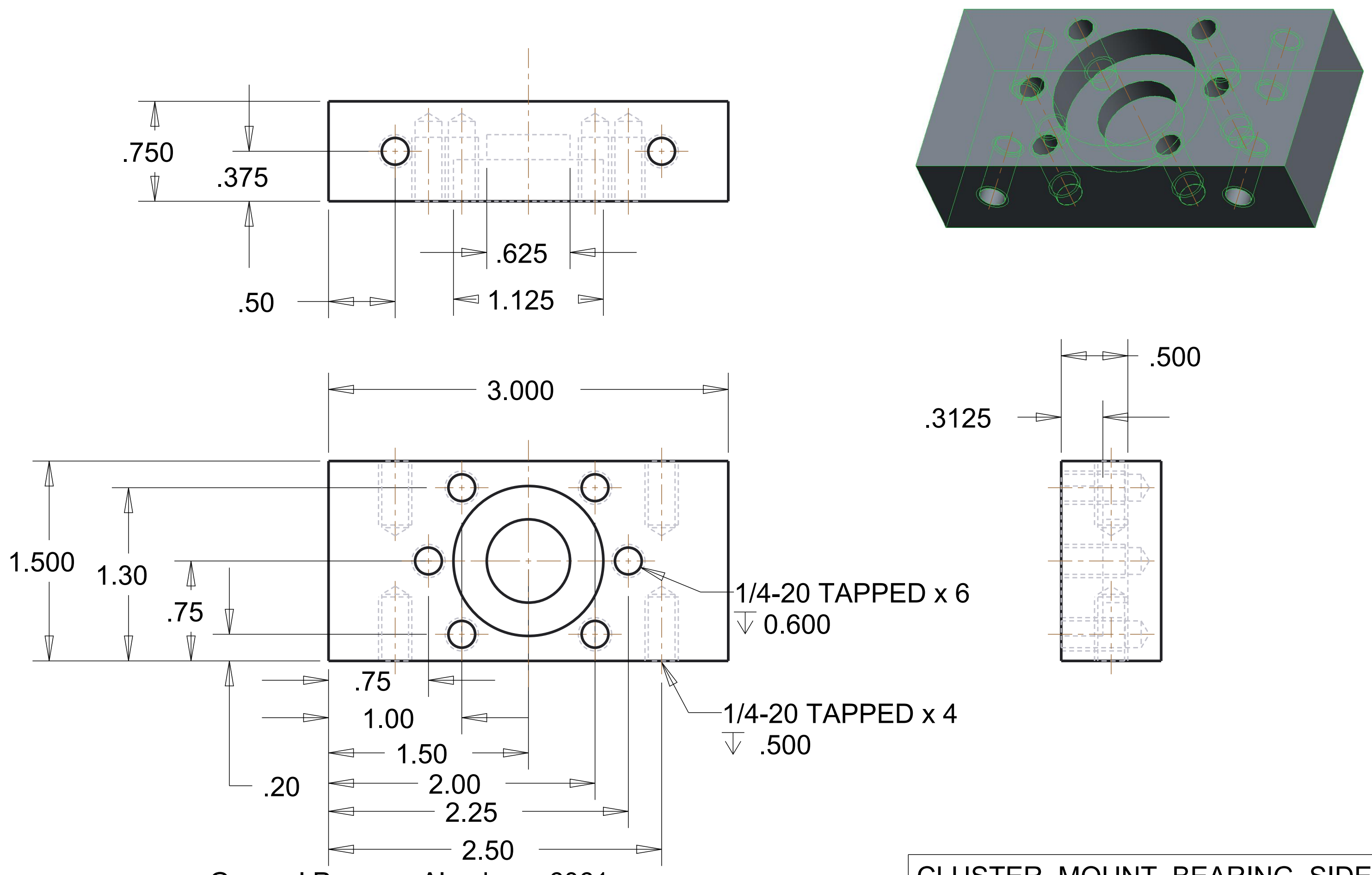
This drawing and schematic are confidential and proprietary to the Applied Space Exploration Laboratory (ASEL) and West Virginia University (WVU). No part of this drawing may be distributed or disclosed without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ______________________



General Purpose Aluminum 6061

CLUSTER_MOUNT_BASE
Drew Bittner
ASEL West Virginia University
2 December 2014

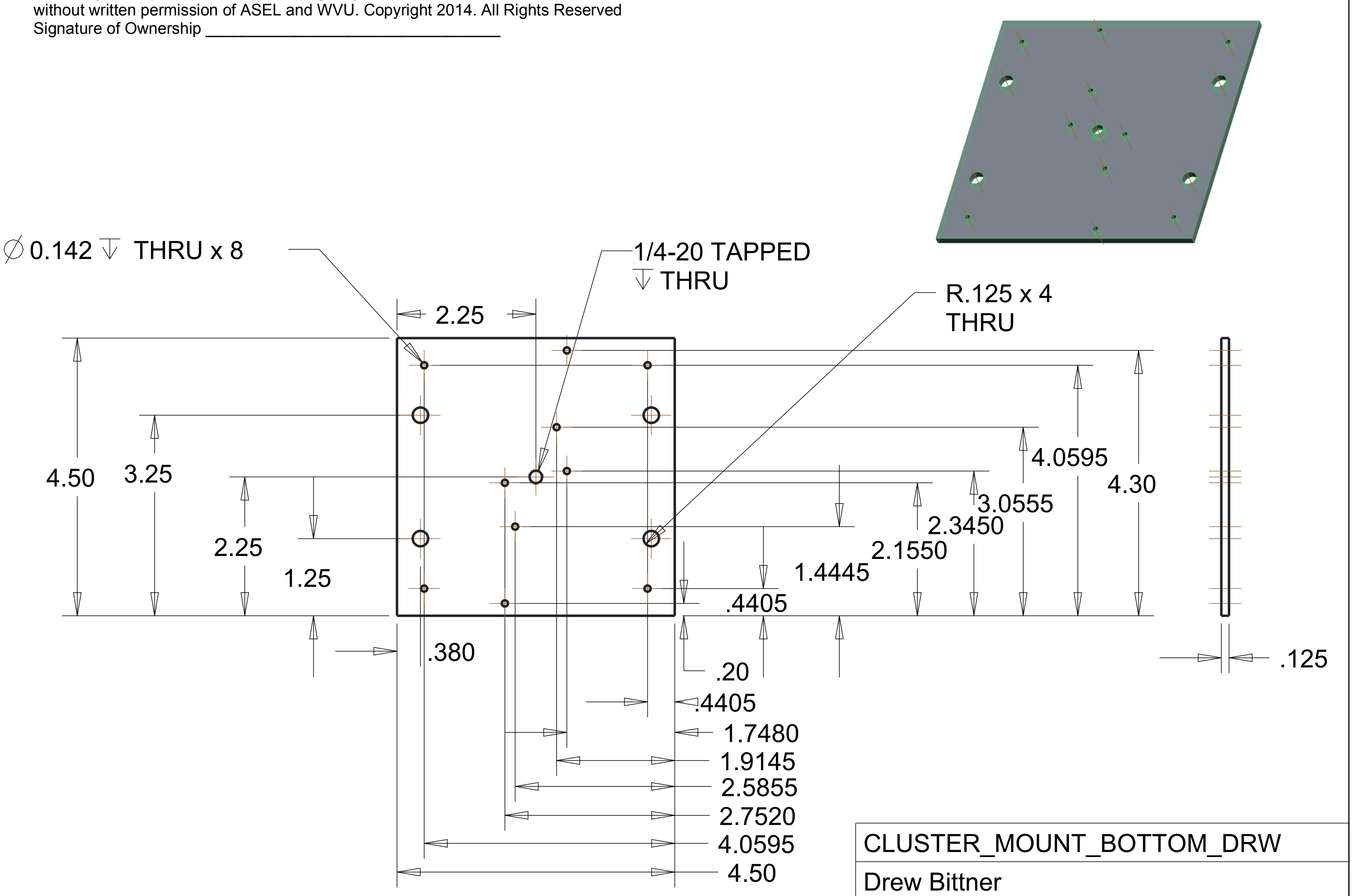
This drawing and schematic are confidential and proprietary to the Applied Space Exploration Laboratory (ASEL) and West Virginia University (WVU). No part of this drawing may be distributed or disclosed without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ________________________________



General Purpose Aluminum 6061

CLUSTER_MOUNT_BEARING_SIDE_DRW
Drew Bittner
ASEL West Virginia University
23 January 2015

This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ________________________________



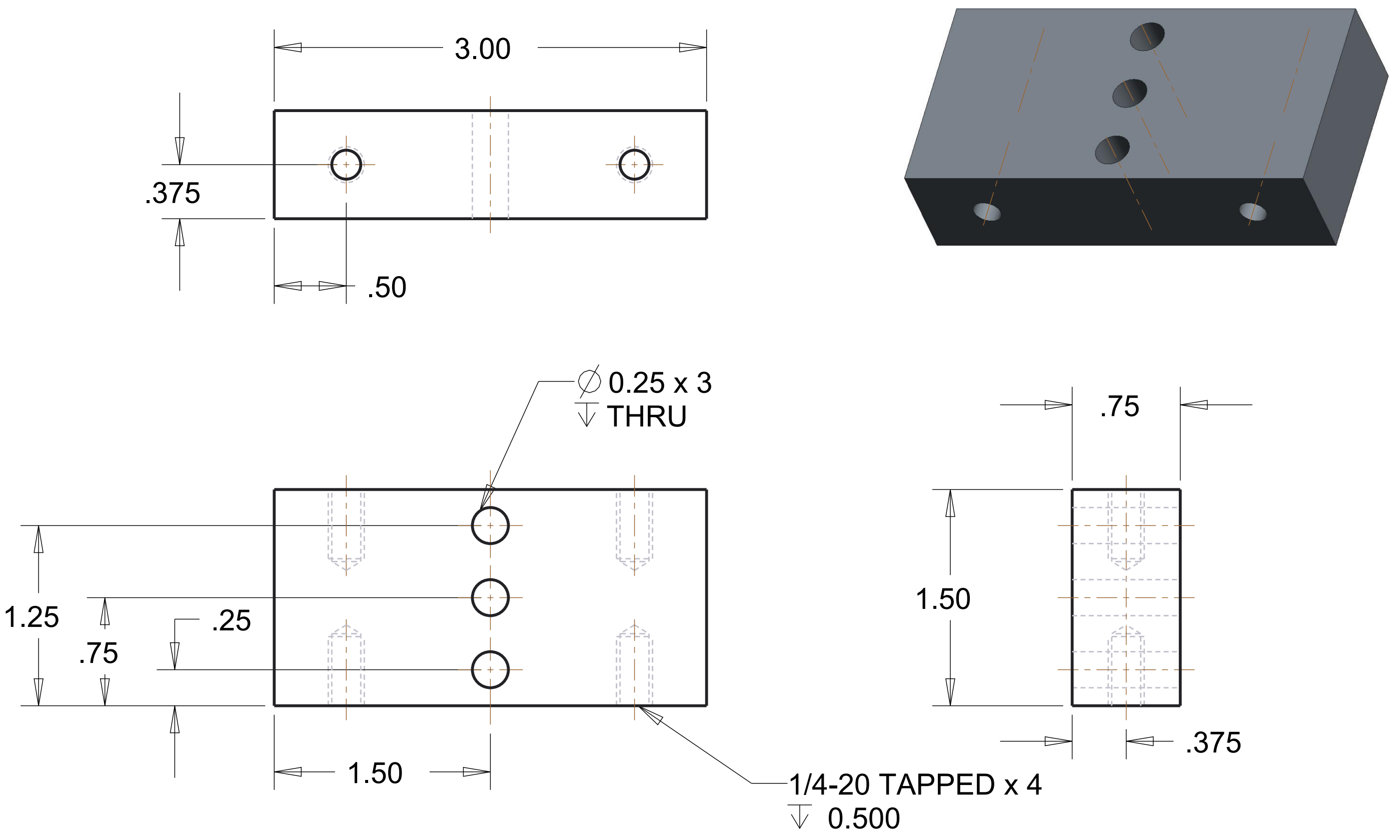
General Purpose Aluminum 6061

CLUSTER_MOUNT_BOTTOM_DRW

Drew Bittner

ASEL West Virginia University

2 December 2014



General Purpose Aluminum 6061

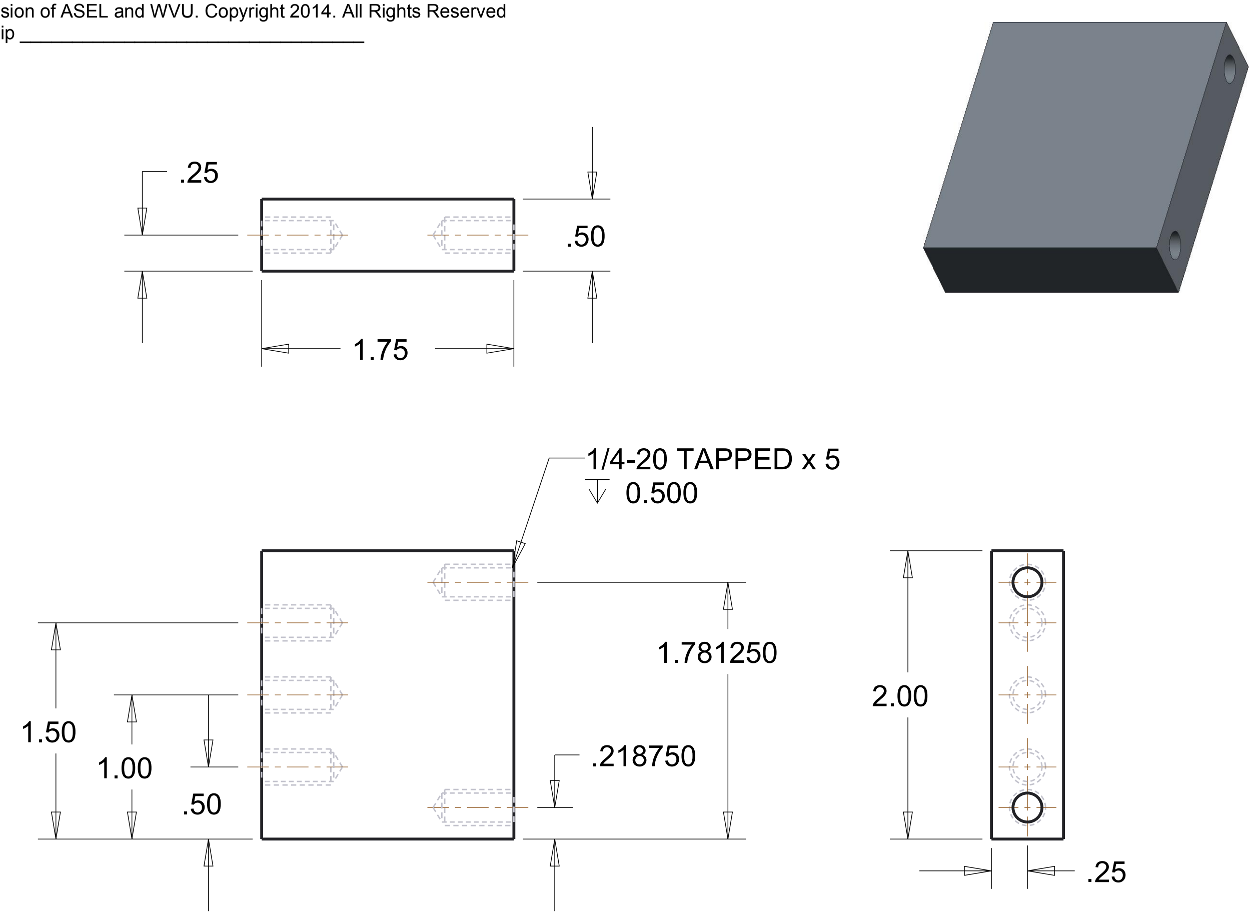
CLUSTER_MOUNT_SIDE_FIXED_DRW

Drew Bittner

ASEL West Virginia University

2 December 2014

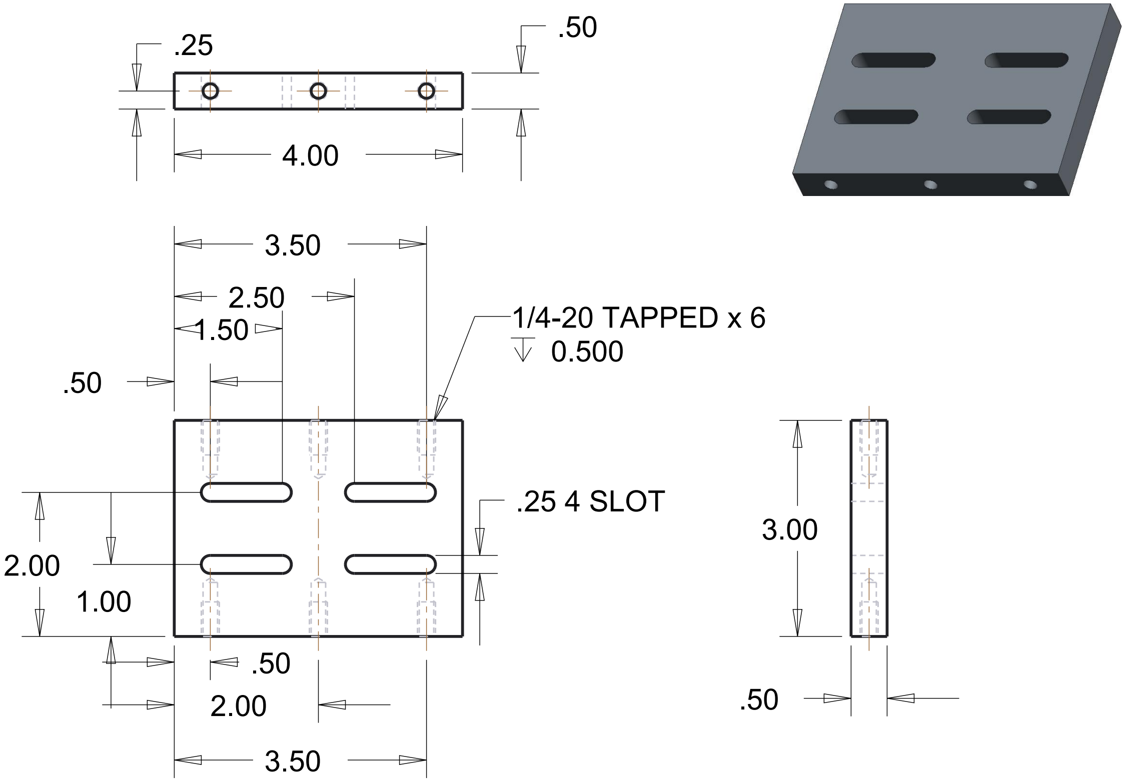
This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ______________________________



General Purpose Aluminum 6061

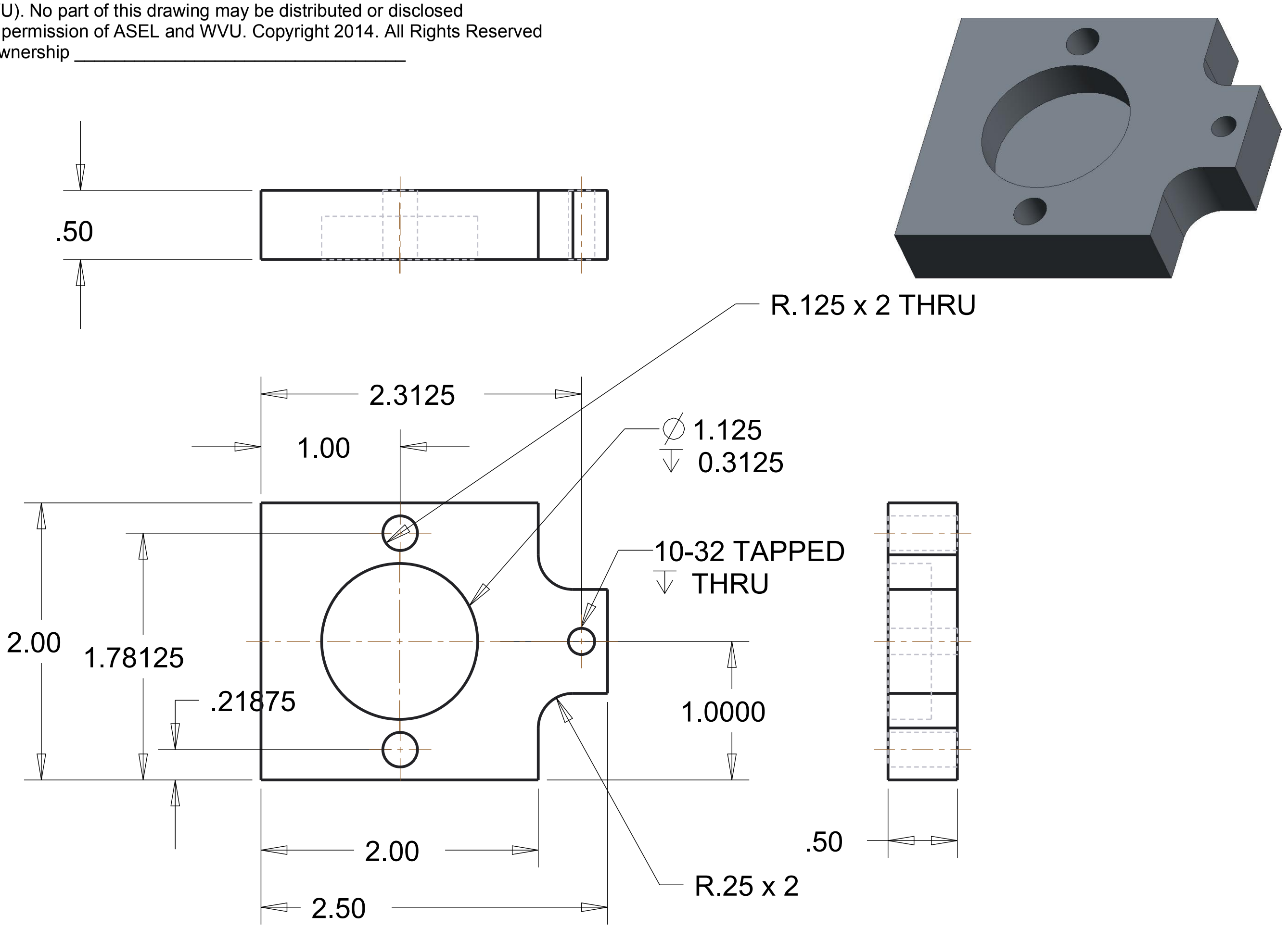
FIXED_ARM_DRW
Drew Bittner
ASEL West Virginia University
3 December 2014

This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ____________________



General Purpose Aluminum 6061

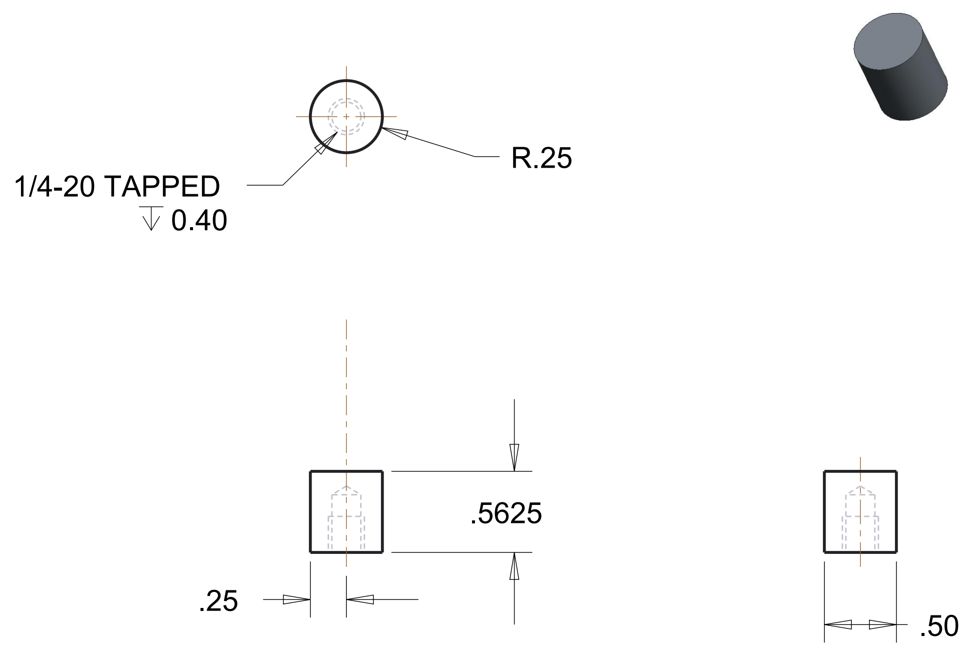
MOUNT_BASE_DRW
Drew Bittner
ASEL West Virginia University
3 December 2014



General Purpose Aluminum 6061

PIN_BEARING_MOUNT_DRW
Drew Bittner
ASEL West Virginia University
3 December 2014

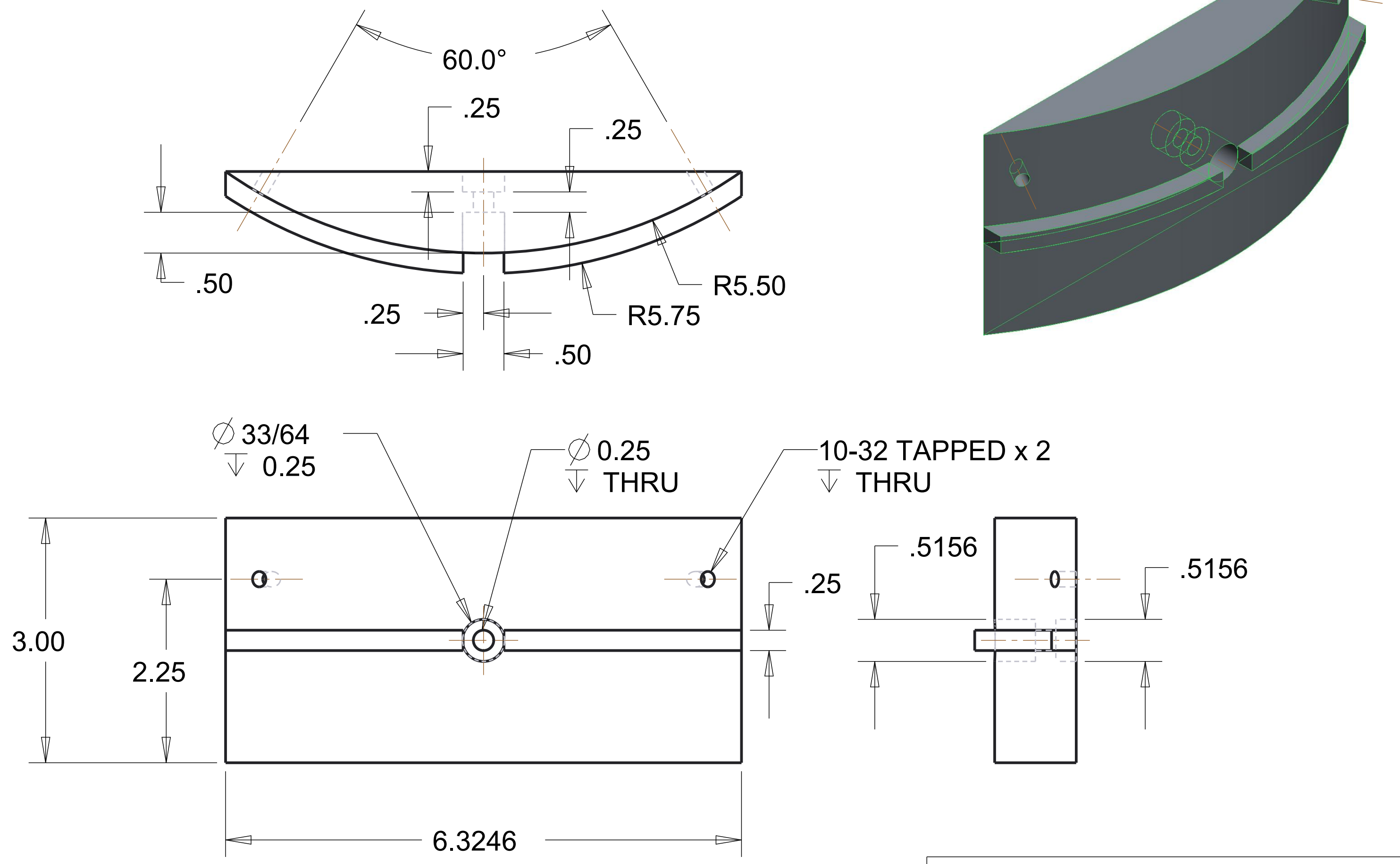
Signature of Ownership ____________________



General Purpose Aluminum 6061

PIN_SIDE_ROTATION_ARM_DRW
Drew Bittner
ASEL West Virginia University
3 December 2014

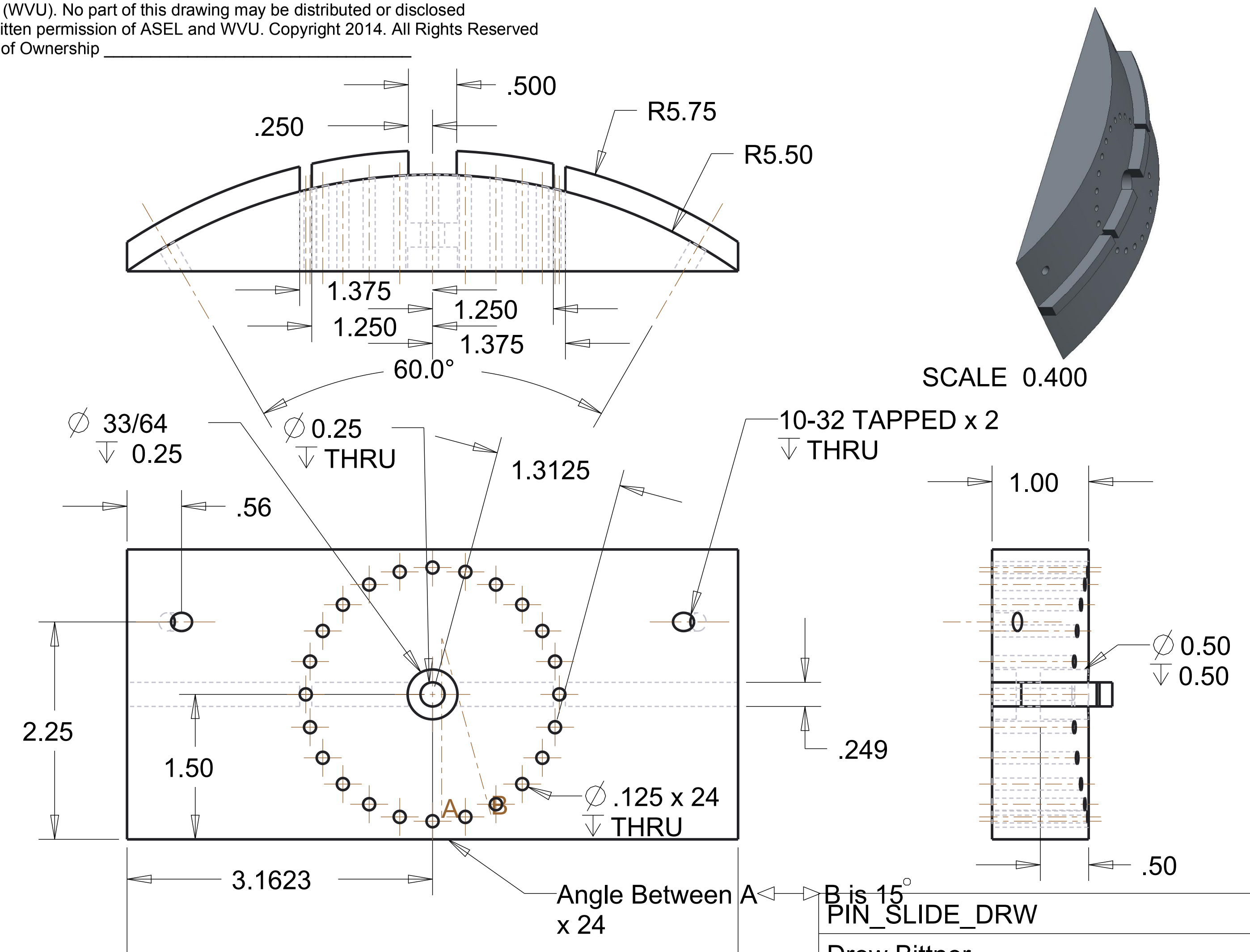
This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ______________________________



General Purpose Aluminum 6061

PIN_SLIDE_2_DRW
Drew Bittner
ASEL West Virginia University
8 December 2014

Signature of Ownership ______________________________



General Purpose Aluminum 6061

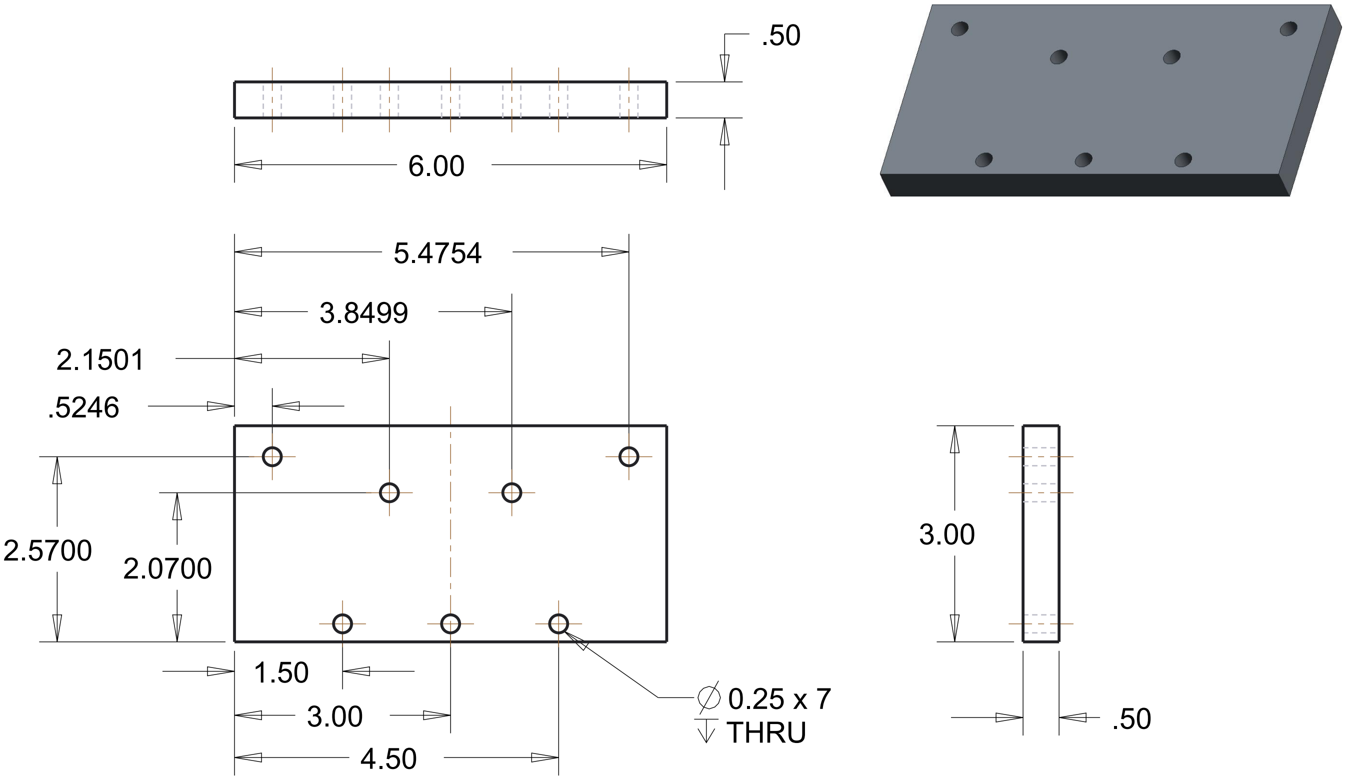
PIN_SLIDE_DRW

Drew Bittner

ASEL West Virginia University

3 December 2014

This drawing and schematic are confidential and proprietary to the
Applied Space Exploration Laboratory (ASEL) and West Virginia
University (WVU). No part of this drawing may be distributed or disclosed
without written permission of ASEL and WVU. Copyright 2014. All Rights Reserved
Signature of Ownership ________________________________



General Purpose Aluminum 6061

SIDE_MOUNT_DRW

Drew Bittner

ASEL West Virginia University

3 December 2014