

Project Thetis
A Low-Cost, Low-Profile Inertial Datalogger

by
Braidan Duffy

Bachelor of Science
Ocean Engineering
Florida Institute of Technology
2021

A thesis
submitted to the College of Engineering and Sciences
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Masters of Science
in
Ocean Engineering

Melbourne, Florida
December, 2023

© Copyright 2023 Braidan Duffy
All Rights Reserved

The author grants permission to make single copies.

We the undersigned committee
hereby approve the attached thesis

Project Thetis
A Low-Cost, Low-Profile Inertial Datalogger by Braidan Duffy

Dr. Stephen Wood, Ph.D., P.E.
Program Chair
Ocean Engineering and Marine Sciences
Committee Chair

Dr. Hector Gutierrez, Ph. D., P.E.
Professor
Mechanical Engineering
Outside Committee Member

Dr. Robert Weaver, Ph. D
Associate Professor
Ocean Engineering and Marine Sciences
Committee Member

Dr. Marius Silaghi, Ph.D.
Professor
Computer Engineering and Sciences
Committee Member

Dr. Richard Aronson, Ph.D.
Department Head
Ocean Engineering and Marine Sciences

Abstract

Title:

Project Thetis

A Low-Cost, Low-Profile Inertial Datalogger

Author:

Braidan Duffy

Major Advisor:

Dr. Stephen Wood, Ph.D., P.E.

Lorem ipsum dolor sit amet, idque apeirian mnesarchum vix ea, usu no mandamus constitutam comprehensam, cum cu vero definitionem. Fabulas probatus ne pro, eleifend hendrerit adipiscing at vim. Usu eu facilisis persecuti. Ne has vidit principes efficiantur, has liber euripidis no. Vide sapientem salutandi per at. Eu probo democritum referrentur pri, an vidit simul reprimique nam. Omnia eruditi ex per, elit regione reprimique te eos. Dicam timeam no nam. Tacimates indoctum inciderint ad his, sea fastidii appetere oporteat an. Ad cum movet efficiantur. Sale labitur no vis, complectitur consequuntur usu te. Mel te putant timeam denique. Ne mel vidit ancillae. Ius saperet facilisi an. Eligendi scaevola sea ex, ad recteque cotidieque quo. Te sumo brute eum.

Table of Contents

Abstract	iii
List of Figures	viii
List of Tables	ix
Abbreviations	x
Acknowledgments	xi
Dedication	xii
1 Introduction	1
2 How It Works	2
2.1 Orientation and Rotation	2
2.1.1 Rotation Matrices	3
2.1.2 Quaternions	6
2.2 Sensing	8
2.2.1 Magnetometer	8
2.2.2 Gyroscope	8
2.2.3 Accelerometer	9
2.2.4 MARG Arrays and Inertial Measurement Units	9
2.2.5 Global Positioning System	12

2.3	Sensor Fusion	12
2.3.1	Attitude and Heading Reference System	13
2.3.2	Inertial Calibration	14
2.3.3	Magnetic Calibration	18
2.3.4	Fast Complimentary Filter	20
2.3.5	Kalman Filter	22
2.3.6	Mahony Filter	23
2.3.7	Madgwick Filter	24
2.4	Project Management Techniques	25
2.4.1	Version Control	25
2.4.2	Systems Engineering	26
3	How I Built It	29
3.1	Stakeholders	29
3.1.1	Hands-On Users	29
3.2	Design Rationale	30
3.2.1	Problem Description	30
3.2.2	Mission Statement	31
3.2.3	Stakeholder Requirements	32
3.2.4	Risk Identification	33
3.2.5	Quality Functional Deployment	44
3.3	Concept of Operations	48
3.4	capabilities	48
3.5	Conceptual Design	50
3.5.1	System Requirements	50
3.5.2	Failure Mode, Effect, and Criticality Analysis	55
3.5.3	Functional Block Diagram	61
3.5.4	Functional Flow Diagram	61
3.6	Brief Revision History	63

3.7	Final Design	65
3.7.1	Schematic	66
3.7.2	Assembly Technique	67
References	70
A	Voltage Divider	71
A.1	Example: Battery Monitor	71
B	Wheatstone Bridge	73
C	Analog Measurement	75
C.1	Analog to Digital Conversion	76
D	Digital Communication Methods	78
D.1	UART Explained	78
D.2	I2C Explained	79
D.3	SPI Explained	80
E	Rotation Matrices	81
E.1	Three Dimensional Rotation Matrix	83
F	Kalman Filter	85
F.1	Introduction	85
F.2	Kalman Filter in One Dimension	86
F.2.1	The Kalman Gain	86
F.2.2	The Estimate Uncertainty Update Equation	87
F.2.3	The Estimate Uncertainty Extrapolation Equation	87
F.2.4	Putting it all Together	87
F.2.5	Process Noise	91

G Traceability Matrix	92
G.1 Requirement Lifecycle	95

List of Figures

2.1	Body rotations	3
2.2	Illustration of the difference between the global and local reference frames .	4
2.3	Illustration of the rotation from a base coordinate frame to a local coordinate frame about the axis, \mathbf{r}	4
2.4	Gimbal lock demonstration	6
2.5	Magnetometer block diagram	8
2.6	Gyroscope block diagram	9
2.7	Accelerometer block diagram	10
2.8	ST124-M Outline	11
2.9	GPS diagram	13
2.10	AHRS block diagram	14
2.11	Three simple graphs	20
2.12	Fast Complimentary Filter	21
2.13	Kalman Filter Process	22
2.14	The system engineering V-method showing the flow of the project. Courtesy of Allouis et. al [CITE].	27
3.1	Thetis RevF5 Block Diagram	61
3.2	Base Firmware Flow Diagram	62
F.1	Kalman Filter Diagram	88
G.1	Lifecycle of a requirement	96

List of Tables

2.1	Common definitions for IMUs of varying degrees of freedom.	10
3.1	A summary of stakeholders of the Thetis device	30
3.2	Stakeholder Traceability Matrix	32
3.12	House of Quality	47
G.1	Description of requirement statuses	95

List of Symbols, Nomenclature or Abbreviations

Start writing here. This is optional.

\sum \sum
 \cap \bigcap

Acknowledgements

Here are your acknowledgments! Don't forget to thank your sponsor

Dedication

Here is your dedication! Lorem ipsum dolor sit amet, idque apeirian mnesarchum vix ea, usu no mandamus constituam comprehensam, cum cu vero definitionem. Fabulas probatus ne pro, eleifend hendrerit adipiscing at vim. Usu eu facilisis persecuti. Ne has vidit principes efficiantur, has liber euripidis no. Vide sapientem salutandi per at. Eu probo democritum referrentur pri, an vidit simul reprimique nam. Omnium eruditi ex per, elit regione reprimique te eos. Dicam timeam no nam. Tacimates indoctum inciderint ad his, sea fastidii appetere oporteat an. Ad cum movet efficiantur. Sale labitur no vis, complectitur consequuntur usu te. Mel te putant timeam denique. Ne mel vidit ancillae. Ius saperet facilisi an. Eligendi scaevola sea ex, ad recteque cotidieque quo. Te sumo brute eum.

Chapter 1

Introduction

We all view the world through different lenses. Biologists look at plants and animals and seeks to understand their interactions; similarly, a scientist considers the universe around us and ponders what drives and constrains it; a psychologist examines human behavior and analyzes its origin and meaning. Humanity has always tried to study the way things work and figure out why they happen. That is the natural product of our curiosity and one of the many things that distinguishes us from other animals.

The fundamental problem that is considered throughout this thesis is sensing the inertial characteristics of a body and determining its orientation. The solutions to this problem are not new. It has been well-studied, well-documented, thoroughly proven out, and is even available to the masses at every moment of every day - even though they may not be aware they are using it. The purpose of the content herewithin is not to unveil something new or revolutionary. Rather it is the culmination of years of viewing this problem through the engineer's lense and pondering the questions, "how does this work?" and "how can I make it work for me?" This is a guide into how an engineer can look at a problem and tinker with it and research it and develop something for it that enhances their understanding. While the final product may not be itself an innovation, the skills and knowledge required to get there can always lead to greater things.

Chapter 2

How It Works

Measuring the movement of bodies in three dimensional space is not a new field of study. Especially in our current age of electronics, one can simply grab an accelerometer and microcontroller off of the Internet, have it delivered in less than 96 hours, and know the accelerations felt on a roller coaster or be able to orient a 3D rendering of a bunny¹. What makes today exciting is that improvements in manufacturing and computational density have drastically increased the capabilities of small embedded systems. Now, every cell phone has a sophisticated suite of sensors built into them that have the simple task of detecting whether the phone is in a landscape or portrait mode. Some applications, like Google Maps, have gone further and fused the sensor data, along with GPS information and computer vision, into a robust pedestrian navigation system².

2.1 Orientation and Rotation

A body can be rotated in 3D space along the x-, y-, and z-axis. **Roll** (ϕ) defines rotation about the x-axis; **pitch** (θ) defines rotation about the y-axis; and **yaw** (ψ) defines rotation

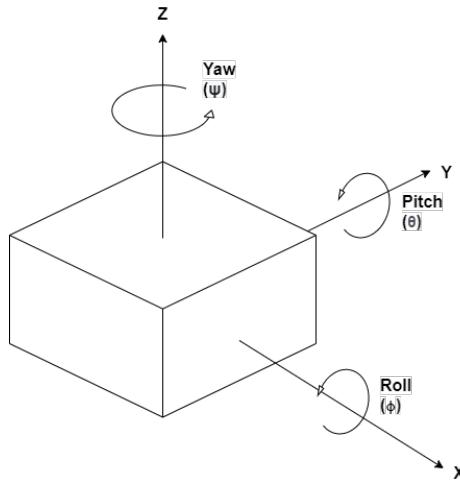
¹<https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview>

²<https://gizmodo.com/i-tried-google-maps-experimental-walking-directions-of-1833225629>

about the z-axis. These three rotations are known as Euler angles.

A body's orientation is always relative to another coordinate frame. Consider a person standing straight and still on the Earth's surface. Each of their appendages could be considered at some orientation relative to their body. At rest and in their local coordinate frame, the person's body may not be considered moving or rotating. But, when the scope is expanded to encompass the entire body and the Earth's frame, that person is now rotating about the Earth coordinate frame at almost 17,000 MPH.

Figure 2.1: Basic look at a 3D body and the axes of rotation in the body reference frame.



2.1.1 Rotation Matrices

A rotation matrix is a mathematical model for translating one body's inertial reference frame to another, e.g. local body to the global body. These are used for Eulerian transformations of vectors. The matrix is an $N \times N$ orthogonal matrix where N is the number of dimensions in the vector; for a three dimensional vector, the rotation matrix to go from one coordinate frame to another would be a 3×3 matrix.

In Figure 2.3, we can see a base coordinate frame, A , and a different coordinate frame, B that is rotated relative to the base frame about the axis, \mathbf{r} . To express the rotation from the base to the local frame, we can borrow notation from Craig [CITE - Introduction to Robotics]. The "from" or "base" frame is the preceding superscript while the "to" or "local"

Aside: Notation and Nomenclature for Rotations

Bodies on their own typically use the "X, Y, Z" notation that is ubiquitous for the Cartesian coordinate system. However, when referring to planetary bodies like the Earth, this nomenclature typically changes to North, East, and Down (NED). This occurs because on the surface of a large spherical body, we can assume the local area is a plane tangential to the surface. To keep the broader scope in mind, we arbitrarily associate the X-axis with East, the Y-axis with North, and the Z-axis with Down towards the Earth's core (perpendicular to the surface).

Figure 2.2: Illustration of the difference between the global and local reference frames

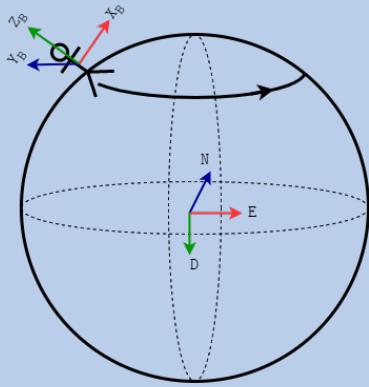
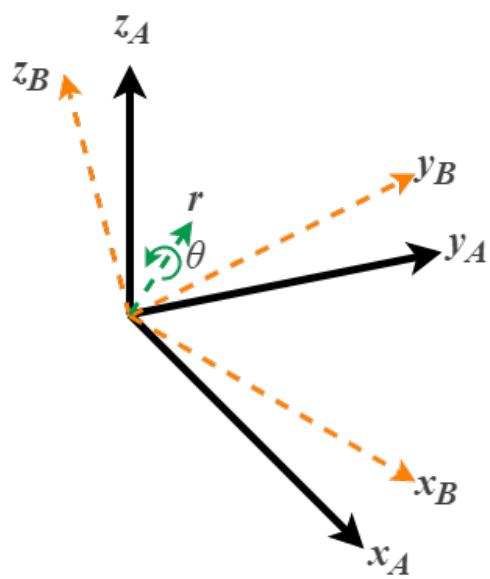


Figure 2.3: Illustration of the rotation from a base coordinate frame to a local coordinate frame about the axis, r



frame is the preceding subscript, as shown below:

$${}^A_B R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} {}^A x_b & {}^A y_B & {}^A z_B \end{bmatrix}$$

When we want to rotate a vector between frames, we must choose an order in which to do so. For example, if we wanted to rotate from frame A to frame B using a yaw-pitch-roll rotation order, we can construct the following rotation matrix from Appendix E.1:

$$\begin{aligned} {}^A_B R &= R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \theta & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \end{aligned}$$

Then, we can rotate the vector from A to B via:

$${}^A \vec{v}_B = {}^A_B R \cdot \vec{v}_A$$

Gimbal Lock When two axes become parallel, Eulerian changes to either axis become irrelevant, ergo, the system loses a degree of freedom. This condition is called "gimbal lock". When in gimbal lock, rotations yield discontinuities that need to be mitigated by changing the rotation order, or moving two or three axes at once. This can create strange pathways for the body and cause unexpected behavior. Monitoring for and breaking gimbal lock are also computationally expensive as the program must perform multiple matrix multiplications and then have logic to determine if a) it is in a gimbal lock condition and b) what rotation

order would be required to break the condition.

The figure below shows a body's rotation while in gimbal lock. The rotation order is roll-pitch-yaw. First, it is pitched 90 degrees upwards to align the roll and yaw axes, $\langle 0, 90, 0 \rangle$ (Figure 2.4b). If we then want its orientation to be $\langle 90, 0, 0 \rangle$ (Figure 2.4e), the body would be rolled left 90 degrees and pitched down 90 degrees. However, due to gimbal lock, the body ended up in the orientation $\langle 0, 0, 90 \rangle$ (Figure 2.4d)! To break the gimbal lock and get to the correct orientation, the rotation order would have to be changed to pitch-yaw-roll, or the roll and yaw axes driven simultaneously.

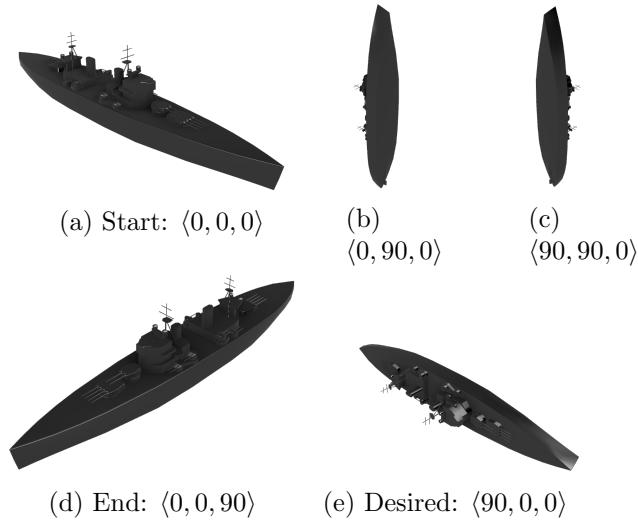


Figure 2.4: Demonstration of gimbal lock on a rotating 3D body. Despite only pitching and rolling the body, the end result of the transformation is an effective yaw. This is because the first pitch aligns the roll and yaw axes, effectively making them the same. In the given rotation order, any roll rotation would be equivalent to a yaw rotation when the object is pitched 90°. *3D model courtesy of "printable models" from Free3D.com.*

2.1.2 Quaternions

In the 19th century, an Irish mathematician was contemplating the problem of Euler angles and analyzing three dimensional geometry. William Hamilton considered the problem from within the complex or imaginary space and found that a three dimensional vector could not be expressed with a complex triplet. Instead, Hamilton discovered that the problem could be solved with a complex quadruplet, called a quaternion [CITE - On Quaternions], which

could be used as a more mathematically pure tool to understand three dimensional geometry. For more detailed notes, see Kuipers [CITE - QUATERNION TEXTBOOK]

A quaternion exists in four dimensional Hamiltonian space and has three complex and one real component in its vector. Like a rotation matrix, it describes the rotation from one reference frame to another about some axis, \vec{r} , as shown in Figure 2.3. But, the quaternion is not susceptible to gimbal lock as when two of its axes align, it still retains three degrees of freedom. Additionally, it is not limited to the range of 0 to 360 (or -180 to 180) like Euler angles and therefore has no discontinuities. We can express the quaternion from frame A to frame B as:

$${}^A_B \vec{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} = \begin{bmatrix} w & x & y & z \end{bmatrix} = w + x\hat{i} + y\hat{j} + z\hat{k} \quad (2.1)$$

Since the quaternion is just a vector, this drastically simplifies any rotation calculations and improves computational performance. In order to rotate a vector using a quaternion, we need to set the quaternion to be a unit quaternion by:

$$q = \frac{\vec{q}}{|q|} \quad (2.2)$$

Then, we can calculate the inverse quaternion, q^{-1} (or conjugate $q^* = q^{-1}$, if $|q| = 1$), using:

$$q^{-1} = \frac{q_0 - q_1\hat{i} - q_2\hat{j} - q_3\hat{k}}{|q|^2} \quad (2.3)$$

Then, we can rotate the vector using the equation:

$${}^A_B \vec{v} = {}^A_B q \vec{v} {}^A_B q^{-1} \quad (2.4)$$

In an inertial measurement unit or attitude and heading reference system, we will need to rotate freely between readings taken in the body's local frame to that in the global frame, such as determining linear acceleration. Using the quaternion method above we can also

relate body measurements to the global frame, which can be useful for other analyses.

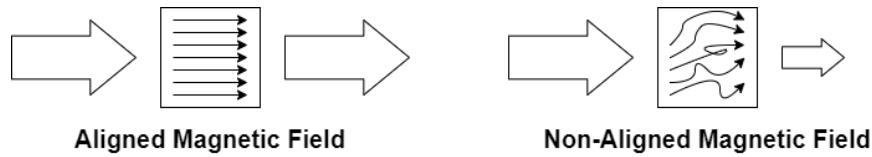
2.2 Sensing

Now that we have a basic understanding of some of the mathematical relationships present in inertial tracking, we need to start sensing our environment. So, how can we compute the position, velocity, acceleration, and attitude of a body in space? In order to answer this question, we must first examine the different sensors that are available.

2.2.1 Magnetometer

Magnetometers use magnetoresistive elements that change their effective resistance in the presence of a magnetic field [CITE - Robotics, Vision and Control]. Atoms within a magnetoresistive element change their orientations with the magnetic field. The new orientation can hinder or aid the path of free electrons moving through the element, thus changing the resistance. By measuring this value and correlating it to a measurement scale, the local magnetic field can be determined.

Figure 2.5: Basic block diagram of a single-axis magnetometer where electrons are flowing through the magnetoresistive material. The left diagram shows the condition when the magnetic field is aligned (minimal resistance); the right shows the non-aligned magnetic field condition which increases the resistance the electrons face passing through the material.

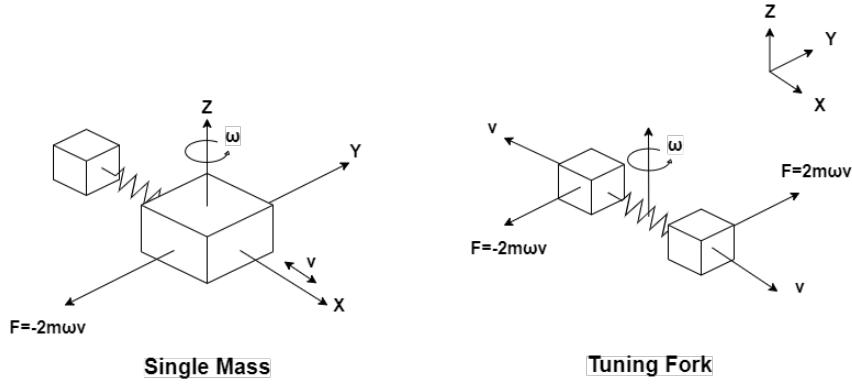


2.2.2 Gyroscope

A gyroscope is an inertial sensor that measures the angular velocity of a rotating body. MEMS-based gyroscopes measure this value by applying the Coriolis effect on a microscopic mass [CITE - Robotics, Vision and Control]. As shown in Figure 2.6, an oscillation is induced on the x-axis using a driving circuit. While oscillating, if an angular velocity (ω) is imparted

on the z-axis, the suspended mass will experience a force in the y-axis that is proportional to ω (Appendix [INSERT APPENDIX NUMBER]). Since the mass is suspended on springs, Newton's Second Law can be applied again to directly correlate the force experienced to a displacement and therefore a change in resistance or capacitance.

Figure 2.6: Basic block diagram of a three-axis gyroscope where the masses are suspended from springs. The left diagram shows a single mass configuration; the right shows a tuning fork configuration which is twice as sensitive as the singular mass.



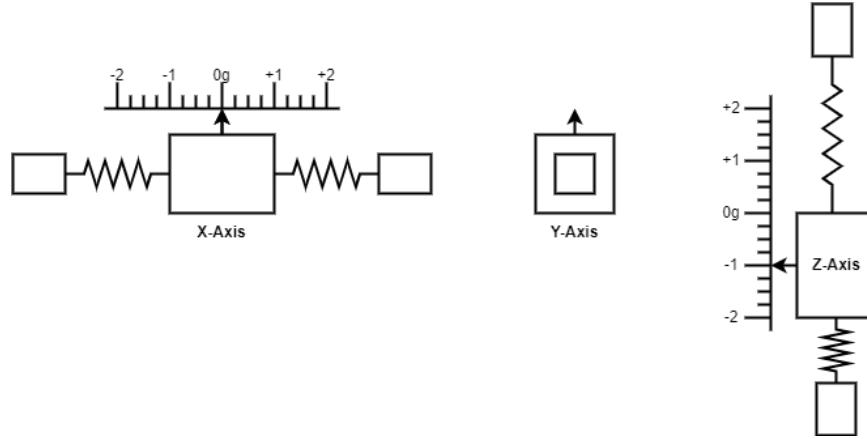
2.2.3 Accelerometer

Accelerometers measure a change in velocity over time (acceleration). An accelerometer is comprised of a mass suspended in an axis of motion by springs of a known K-constant [CITE - Robotics, Vision and Control]. From Newton's Second Law, as applied to a spring-mass system, $\frac{-K}{m}x = a$, we can directly correlate the displacement of the mass to the magnitude of the acceleration along the measurement axis. Typically, this scale is electrically resistive or capacitative and creates an analog change in a driving (exciting) voltage which can be measured in a circuit using a Wheatstone Bridge and microcontroller. A basic representation of a three-axis accelerometer is shown in Figure 2.7.

2.2.4 MARG Arrays and Inertial Measurement Units

Now, with the basics of each sensor in mind, we can combine them into a single package, called a Magnetic, Angular Rate, and Gravity (MARG) array. Each of the above diagrams

Figure 2.7: Basic block diagram of a three-axis accelerometer where the masses are suspended from springs.



represent a single sensing axis, or Degree of Freedom (DOF). In order for a MARG array to be useful in a 3D environment, it needs to have three sensing axes orthogonal to each other. Each DOF will measure the X-, Y-, and Z-axis, respectively with the positive sensing direction according to the right hand rule. By combining multiple tri-axial arrays, we can define different types of Inertial Measurement Units (IMUs), as shown in Table 2.1. Typically, the more sensing axes, the more accurate the array will be (depending on the performance of the sensor fusion algorithm). Note that a MARG array is an IMU with an integrated tri-axial magnetometer .

Table 2.1: Common definitions for IMUs of varying degrees of freedom.

DOFs	Accelerometer	Gyroscope	Magnetometer	Barometer
3-DOF	3-axis ²	3-axis ²	—	—
6-DOF	3-axis	3-axis	—	—
9-DOF	3-axis	3-axis	3-axis	—
10-DOF	3-axis	3-axis	3-axis	1-axis

According to VectorNav [CITE], a producer of industrial-grade IMUS, the cheapest, least precise, and least accurate IMUs are considered "consumer-grade". Every day smartphones, cheap commercial breakout boards, and even shipping crates have these devices on-board. Consumer-grade IMUs can be bought for cents, dollars, or tens of dollars per unit in bulk and are ideal for mass production spaces where quantity is king over quality. A step up from these

Aside: MEMS Technology

During the Apollo program, the ST124-M inertial measurement unit was developed that fed the Saturn V rocket's inertial characteristics to the main flight computer [CITE - ST124-M description]. The IMU consisted of a tri-axial gyroscope array, redundant tri-axial accelerometer arrays, pendulums, and other sensors. While it was a technological marvel at the time, it was the size of a basketball and weighed about 45-65 kilograms. Modern cellphones with the same capability are orders of magnitude smaller, lighter, and cheaper; so, what happened?

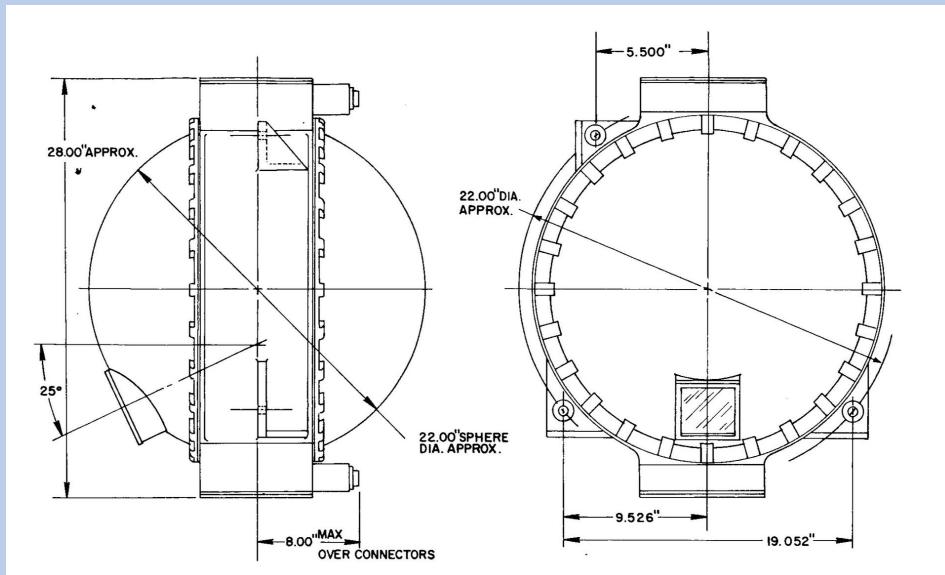


Figure 2.8: ST124-M inertial measurement unit courtesy of NASA via Wikimedia Commons [CITE - Wikimedia]

Modern manufacturing methods have enabled a new technology called Microelectromechanical Systems, or MEMS. MEMS are microscopic devices that perform an electromechanical function such as sensing acceleration, rotation rate, or magnetic fields [CITE - What is MEMS]. They are the most common technology in modern sensor development and revolutionized the technology space by shrinking components down to the micro- and nanometer scales. Due to the advent of MEMS technology, devices such as accelerometers, magnetometers, gyroscopes, barometers, hygrometers, etc. have been shrunk down from large mechanical masterpieces to mass-producible products that fit within a few square millimeters of epoxy. This dramatically cheapened these devices and has allowed more products to integrate smart sensors into their designs.

sensors are "industrial-grade" IMUs. These are tens to hundreds of dollars per unit but are an order of magnitude more accurate and precise than their consumer counter parts. This makes them desirable for the automotive and industrial sectors as they can assist in automation, control, and monitoring of expensive unmanned systems. The "tactical-grade" sensors are even more robust and accurate than the previous tiers and have an appropriate military-industrial complex price tag to match. These devices will typically be used in applications where war fighters need precise guidance for munitions, or need to navigate their way through hazardous and GNSS-denied terrain. Finally, the last major tier is the "navigation-grade" sensors. These sensors are extremely precise, extremely accurate, and cost more than some middle-class families will make in a year. Primarily, these will be used in survey missions or on underwater vehicles where absolute precision and knowledge of their location is necessary.

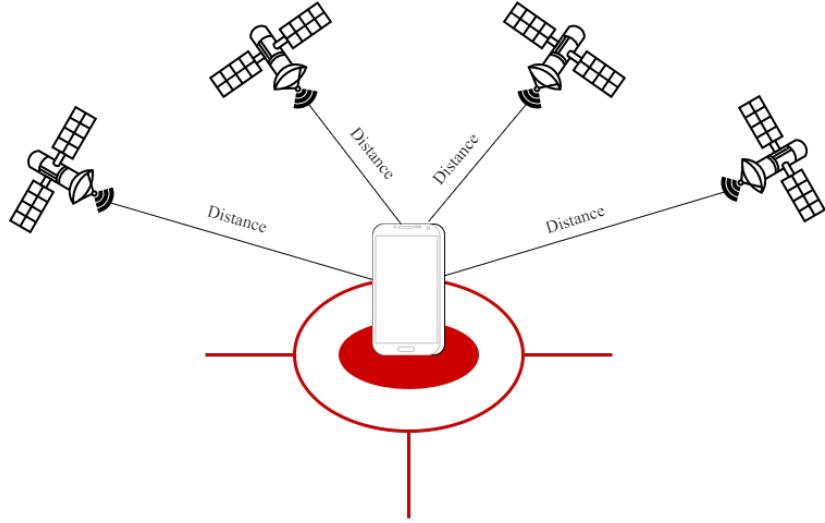
2.2.5 Global Positioning System

The Global Position System, or GPS, is a constellation of high-altitude satellites that service most of the globe [CITE]. First developed by the US military for large scale maneuvers on the battlefield, GPS is a ubiquitous technology that is available in almost every device from smartphones to cars. Each GPS satellite in orbit transmits the current time measured by their internal atomic clocks. A GPS receiver on the ground can synchronize its own internal clock to the GPS time and wait for a satellite's transmission. When the satellite time is received, the device can determine the difference between its clock and the satellite's report called the Time of Flight. Since the Time of Flight can be assumed to be the constant speed of light, the GPS receiver can determine its distance to a satellite in a known orbit. Repeat for at least three satellites, and a GPS receiver can triangulate its position to a reasonable circle of error of about 3-meters.

2.3 Sensor Fusion

While having each individual sensor will give some data, they offer an incomplete picture of a body's orientation and movement through space. An accelerometer can detect acceleration

Figure 2.9: Basic representation of how GPS communicates with a device to calculate its position on Earth. GPS satellites in orbit broadcast a time and using trigonometry and algebra, a device can calculate the distance to various satellites and triangulate its position.



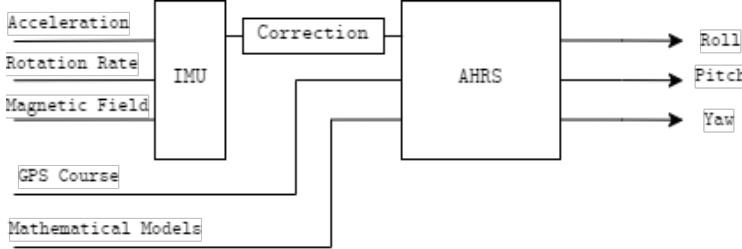
and determine pitch and roll using basic vector math, but it cannot determine yaw or heading. Data from a gyroscope can be integrated over time to determine the sensor's attitude, but this method will drift over time and accumulate errors, it also cannot detect movement. Magnetometers in a weak magnetic field, like Earth's, are not accurate enough to determine roll and pitch, but can easily provide heading. Finally, GPS readings will provide position and velocity vectors, but typically update slowly and have a large circle of error and cannot determine attitude. In order to make these sensors more effective for an inertial sensing application, we need to fuse the data feeds into a unified output that emphasizes the strength of each sensor, while mitigating their limitations.

2.3.1 Attitude and Heading Reference System

An Attitude and Heading Reference System (AHRS) is an IMU equipped with an accelerometer, gyroscope, and/or a magnetometer in all three axes. The data streams from the sensors can be fused together with external information like GPS data and mathematical models to estimate a body's inertial orientation in three dimensional space. The block diagram for this operation is shown in Figure 2.10. Many algorithms exist to do this sensor fusion such as the

the Fast Complimentary Filter [CITE], Kalman filter [CITE - Kalman], the Mahony filter [CITE], and the Madgwick filter [CITE].

Figure 2.10: Basic block diagram of an AHRS. The acceleration, rotation rates, and magnetic field readings are fused together in the IMU the filtered using calibration data. The AHRS can then apply a bias from the GPS course and corrections from a mathematical model of the system.



Before measurements are fed into a sensor fusion algorithm, the data should be pre-filtered based on known calibration data. This will improve the sensor fusion performance and accuracy of the resultant values.

2.3.2 Inertial Calibration

In order to calibrate an IMU, we first need to create a model of the data. For the accelerometer and gyroscope, we can create an inertial model defined as:

$$\vec{i}_c = MS(\vec{i}_u - \vec{b}) \quad (2.5)$$

where \vec{i}_c is the calibrated inertial measurements,

M is the misalignment matrix,

S is the sensitivity identity matrix,

\vec{i}_u is the uncalibrated inertial measurements, and

\vec{b} is the bias or offset vector

The three calibration values, M , S , b , represent a set of correction factors that make the measured values more accurate. The inertial measurement is generalized here to represent

either accelerometer or gyroscope data. Each sensor will have its own set of the calibration values.

Bias Vector The bias, or offset, vector is the average of the inertial readings while the sensor is in a stable, known orientation. For example, when the gyroscope is at rest, we would expect the measurement output to be $[0, 0, 0]$ [deg/sec]. However, we may measure an average of $[0.89, -0.66, 0.31]$ [deg/sec] instead. By subtracting this bias vector from the measurements, we eliminate any unintentional offset from our readings. The bias can be calculated using the equation below:

$$\vec{b}_g = \frac{1}{N} \sum_{i=0}^N \vec{g}_n \quad (2.6)$$

where \vec{b}_g is the bias vector,

N is the number of collected samples to average, and

\vec{g}_n is the n-th uncalibrated gyroscope measurement vector in the dataset

For accelerometers, the bias is calculated differently. Each measurement axis must be exposed to $\pm 1g$ of acceleration by placing the instrument vertical on each of the measurement axes in both the positive and negative directions. The bias for each axis can then be determined by taking the average value as shown below:

$$\vec{b}_a = \frac{1}{2} \left[\frac{1}{N} \sum_{n=0}^N \vec{a}_{n,+g} + \frac{1}{M} \sum_{m=0}^M \vec{a}_{m-g} \right] \quad (2.7)$$

where b_a is the bias vector,

N is the number of samples taken in the +1g orientation,

a_{+g} is the average axis measurement when exposed to +1g, and

M is the number of samples taken in the -1g orientation,

a_{-g} is the average axis measurement when exposed to -1g

Sensitivity Matrix The sensitivity matrix is a diagonal matrix that accounts for minor errors with variations in process and material. This can make each uniaxial sensor in a triaxial array sense the environment slightly differently. While bias covers this area when at rest or in specific orientations, the sensitivity error will change depending on the motion and orientation. To account for this, we need to expose each sensor to a known reference stimulus and calculate a sensitivity value based on the average magnitude of the measurement vector during that time. For a gyroscope, this value can be an arbitrary, but constant, rotation rate, ω . For an accelerometer, this value will be $\pm 1g$. The equations for each gyroscope and accelerometer axis are provided below:

$$s_{g,\omega} = \frac{\|g_{+\omega}\| + \|g_{-\omega}\|}{2\omega} \quad (2.8)$$

where $s_{g,\omega}$ is the sensitivity value for each gyroscope axis when exposed to ω ,

$g_{+\omega}$ is the average gyroscope axis reading whe exposed to $+\omega$,

$g_{-\omega}$ is the average gyroscope axis reading whe exposed to $-\omega$, and

ω is the reference rotation rate

$$s_{a,g} = \frac{\|a_{+g}\| + \|a_{-g}\|}{2} \quad (2.9)$$

where $s_{a,g}$ is the sensitivity value for each axis when exposed to 1g,

a_{+g} is the average axis measurement when exposed to +1g, and

a_{-g} is the average axis measurement when exposed to -1g

After calculating these values for each axis, we can form the sensitivity matrix for the gyroscope and accelerometer like so:

$$S_g = \begin{bmatrix} s_{g,x} & 0 & 0 \\ 0 & s_{g,y} & 0 \\ 0 & 0 & s_{g,z} \end{bmatrix} \quad (2.10)$$

$$S_a = \begin{bmatrix} s_{a,x} & 0 & 0 \\ 0 & s_{a,y} & 0 \\ 0 & 0 & s_{a,z} \end{bmatrix} \quad (2.11)$$

While this calibration value accounts for some of the intrinsic sensor error, it does not account for axes misalignment or non-orthogonality within the sensor. To accomplish this, we must incorporate the misalignment matrix.

Misalignment Matrix The misalignment matrix is the final, but most complex, calibration value that we can calculate for the inertial sensors. It reduces error from multiple sources such as non-orthogonality between the measurement axes, the misalignment from the measurement axes to the actual sensor packaging, and misalignment from the package onto the application board. In order to calculate the misalignment matrix, we must consider it as the solution to a non-linear problem. First, we must define an objective function for the solution space. Since the goal of the misalignment matrix is to reduce error, then we can define the objective function as the Root Mean Square Error (RMSE) of the measurement vector, the

misalignment vector, and the reference vector:

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=0}^N \|\vec{i}_{n,u} \cdot M^* - \vec{i}_{n,ref}\|^2} \quad (2.12)$$

where N is the number of samples in the dataset,

$\vec{i}_{n,u}$ is the n-th uncalibrated measurement vector with the bias removed,

M^* is the guessed 3-by-3 matrix that corrects $i_{n,u}$ to be near i_{ref} , and

\vec{i}_{ref} is the expected reference stimulus vector for the measurement

Then, given a dataset of measurement vectors and their expected reference vectors, we can use a non-liner solver to determine the value of M^* . This can be done using toolboxes⁴. Assuming the sensitivity was not removed from the measurement signal, the M^* matrix happens to include it as its diagonal. By extracting the diagonal and normalizing the original matrix, we are left with the sensitivity and misalignment matrix calibration values.

$$S = \text{diag}(M^*)M = M^* \cdot S^{-1} \quad (2.13)$$

We can now apply the misalignment matrix, sensitivity matrix, and bias vector to the sensor readings according to the model above. This should yield outputs that are close to the real values, which is examined in the next section.

2.3.3 Magnetic Calibration

The magnetometer in a MARG array needs a different sensor model than the inertial sensors because the ambient magnetic field introduces different errors. There are two types: soft iron, W , and hard iron, \vec{v} . A hard iron distortion is introduced by magnetic materials placed near the sensor. This can occur when the sensor is placed in a casing with magnetic fasteners or

⁴Like the `scipy.optimize.minimize` toolbox

a speaker, as shown in Tuupola [CITE - Tuupola]. Soft iron distortion is typically less severe than hard iron distortion and is caused by materials near the sensor that distort the local magnetic field. The soft iron distortion can also account for misalignment and sensitivity errors. The sensor model for a calibrated magnetometer is given below:

$$\vec{m}_c = W^{-1}(\vec{m}_u - \vec{v}) \quad (2.14)$$

where \vec{m}_c is the calibrated magnetometer vector,

W^{-1} is the inverse soft iron matrix,

\vec{m}_u is the uncalibrated magnetometer vector,

\vec{v} is the hard iron bias vector

With a calibrated sensor in an ideal environment, we would expect a sphere centered on the origin with a radius equal to the magnitude of the local magnetic field. Due to the hard and soft iron distortions, this will rarely be the case. Hard iron distortion can be removed from the signal by determining the average of the minimum and maximum values of each axis, as shown below:

$$\vec{v} = \frac{1}{2} \begin{bmatrix} \max(v_x) + \min(v_x) \\ \max(v_y) + \min(v_y) \\ \max(v_z) + \min(v_z) \end{bmatrix} \quad (2.15)$$

Once the hard iron distortion is removed from the signal, we should see the readings more closely adhere to the correct spherical shape. Then, we can apply Li's ellipsoid fitting algorithm[CITE - Li] to characterize the fit of the optimal ellipsoid for the data, expressed as the symmetrical matrix, A . Ozyagcilar [CITE - AN4246] builds a relationship between the ellipsoid fit and the soft iron matrix, summarized by:

$$W^{-1} = A^{1/2} \quad (2.16)$$

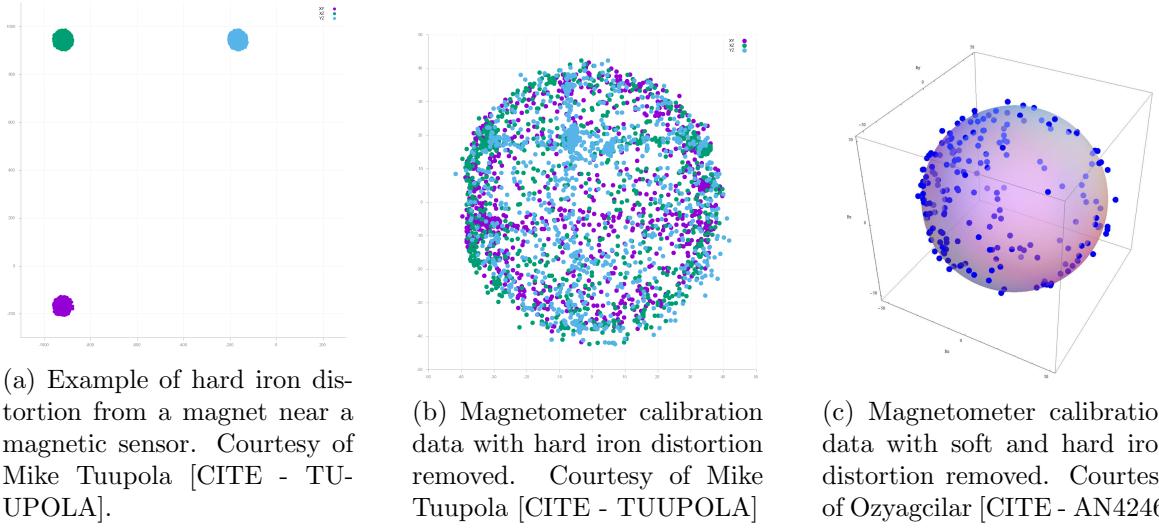


Figure 2.11: Three simple graphs

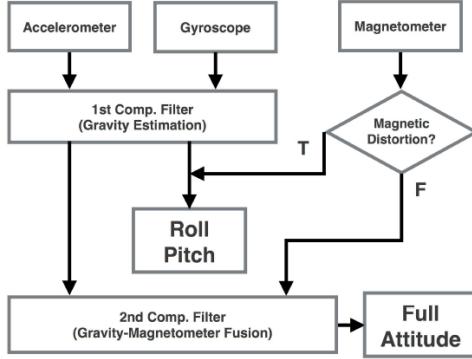
With these calibration parameters calculated, they can be used in a sensor fusion algorithm to increase the accuracy of the magnetometer. However, these parameters must be recalibrated whenever the sensor enters a new magnetic environment since either the hard or soft iron distortion could be different than previously calculated.

2.3.4 Fast Complimentary Filter

A complimentary filter take advantage of the problems inherent with the accelerometer and gyroscope measurements such they *compliment* each other. By integrating gyroscopic measurements, we can quickly obtain the body's attitude but the errors in the gyroscope sensor cause this value to quickly drift away from the "true" value. Accelerometer measurements on the other hand can use trigonometry on the gravitational acceleration vector to determine pitch and roll (tilt). These measurements are affected by the body's motion and can drift over time as the body experiences other accelerations than gravity. An accelerometer also cannot report the yaw degree, so must be coupled with a tri-axial magnetometer which can also be influenced by external magnetic fields.

In the frequency space, we can consider the gyroscope attitude estimate to drift with a high frequency, and the accelerometer-magnetometer signal to drift with a low frequency.

Figure 2.12: A simplified block diagram for the Fast Complimentary Filter. Courtesy of Wu et al. [CITE - FCF]



To compensate for these drifts, we can establish a complimentary or band-pass filter that attenuates the drift signal and outputs the best estimate of orientation. This filter typically uses a hyperparameter, γ , to control the attenuation of each of the drifts, i.e. the weight of each estimation like so:

$${}^G_B \mathbf{q}_{est} = (1 - \gamma) {}^G_B \mathbf{q}_{lf} + \gamma {}^G_B \mathbf{q}_{hf} \quad (2.17)$$

where ${}^G_B \mathbf{q}_{est}$ is the best estimate of the body quaternion relative to the global frame,

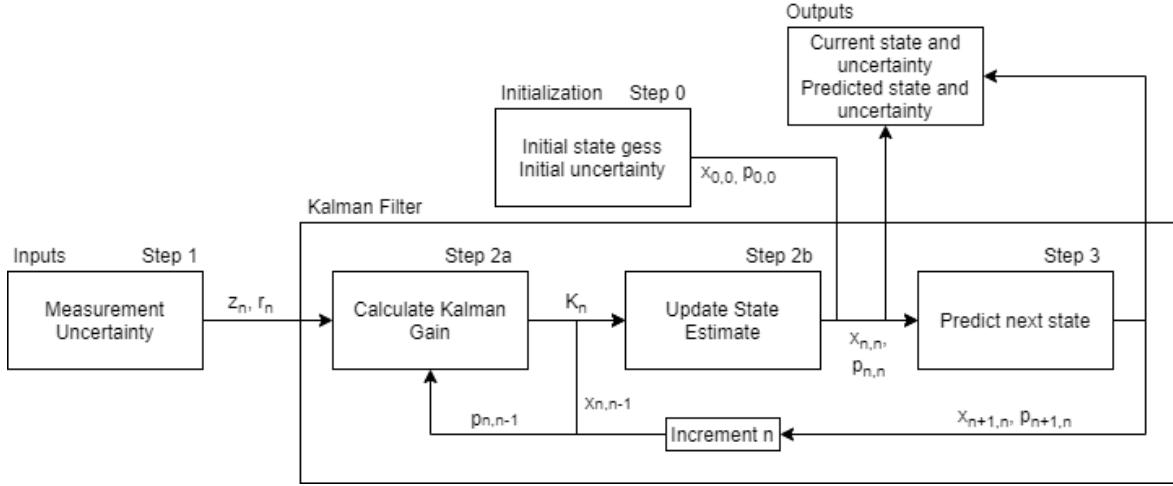
γ is the controlling hyperparameter,

${}^G_B \mathbf{q}_{lf}$ is the body quaternion estimate from the low frequency drift source, and

${}^G_B \mathbf{q}_{hf}$ is the body quaternion estimate from the high frequency drift source.

Wu et. al (CITE - FCF) propose a complimentary filter solution that reports being on par with more computationally expensive fusion algorithms like the Kalman Filter and Madgwick Filter. This algorithm computes the gravity vector using accelerometer and gyroscope measurements. By itself, this can yield the tilt attitude of the body. Then, if there is not severe magnetic distortion detected by the magnetometer, the full attitude can be acquired by fusing together the tilt attitude and that estimated from the magnetometer. The basic block diagram for the filter is in Figure 2.12.

Figure 2.13: A simplified process block diagram for the Kalman Filter that includes the steps required for each iteration.



2.3.5 Kalman Filter

The Kalman Filter [CITE - KF] was introduced in the 1960's as a method to estimate and track a series of explicit and hidden states within a system. In a sensor fusion/AHRS application, the Kalman Filter can be used to estimate the orientation state (quaternion) of the system based on the gyroscope, accelerometer, and magnetometer readings. The Kalman Filter also estimates the uncertainty and noise associated with attitude estimate to give more insightful feedback on the usefulness of the estimate. At each timestep, the filter uses a built-in mathematical model of the system to predict the current system state and its associated uncertainty. Then, the filter can predict the state and uncertainty in the next time step and uses that to adjust a correction factor, called the Kalman Gain. With each iteration of the filter, the Kalman Gain converges on a minimum value and the uncertainty and accuracy of the estimates increase. The convergence speed and accuracy can be tuned using various hyper parameters like process noise and sensor uncertainty. The process for the Kalman Filter is shown in Figure 2.13 and a more detailed explanation can be found in Appendix F.

2.3.6 Mahony Filter

The Mahony Filter [CITE - Mahony] is a deterministic kinematic observer on the Special Orthogonal Group 3 that is driven by instantaneous attitude and angular velocity measurements. That is, the filter exploits a special relationship between observers to decouple gyroscope measurements from attitude estimates and then does on-line gyroscope bias compensation to increase accuracy. The filter proposes a model for the gyroscope in the body frame as:

$$\mathbf{q} = \mathbf{g}_u + \mathbf{b}_\omega + \boldsymbol{\mu}_\omega \quad (2.18)$$

where \mathbf{g} is the true angular velocity,

\mathbf{g}_u is the uncalibrated measured angular velocity,

\mathbf{b}_ω is the gyroscope bias vector, and

$\boldsymbol{\mu}_\omega$ is the gyroscope measurement noise

Additionally, the filter defines the accelerometer measurement as the instantaneous linear acceleration, (i.e. accelerations without the gravitational acceleration vector), with corresponding bias and noise vectors. The model for the accelerometer is given below:

$$\mathbf{a} = {}_B^G R (\mathbf{a}_u - \mathbf{G}_0 + \mathbf{b}_a) + \boldsymbol{\mu}_a \quad (2.19)$$

where \mathbf{a} is the true linear acceleration in the body frame,

${}_B^G R$ is the rotation matrix from the global to the body frame,

\mathbf{G}_0 is the gravitational acceleration vector,

\mathbf{a}_u is the uncalibrated measured angular velocity,

\mathbf{b}_a is the accelerometer bias vector, and

$\boldsymbol{\mu}_a$ is the accelerometer measurement noise

These measurements are used to express the body kinematics in an Explicit Complementary Filter in quaternion form and then rearranged to solve for the change in the body attitude over time, $\dot{\mathbf{q}}$. This estimate can be integrated over the sampling time to get the best estimated attitude via:

$${}^G_B \mathbf{q}_t = {}^G_B \mathbf{q}_{t-1} + {}^G_B \dot{\mathbf{q}}_t \Delta t \quad (2.20)$$

where ${}^G_B \mathbf{q}_t$ is the body attitude from the global frame to the local frame at time, t ,

${}^G_B \dot{\mathbf{q}}_t$ is the time rate of change for the body attitude at time, t , and

Δt is the sample interval

2.3.7 Madgwick Filter

The Madgwick Filter [CITE - Madgwick] considers body attitude as an optimization problem that can be solved by a gradient descent algorithm (GDA). The filter tries to optimize the difference (deviation) between a predefined reference vector in the global frame, $\mathbf{v}_G = [0, v_x, v_y, v_z]$ with its corresponding measurement vector in the local frame, $\mathbf{s}_B = [0, s_x, s_y, s_z]$.

This deviation is expressed in the objective function:

$$f(\mathbf{q}, \mathbf{v}_G, \mathbf{s}_B) = \mathbf{q} \mathbf{v}_G \mathbf{q}^{-1} - \mathbf{s}_B \quad (2.21)$$

The GDA can compute the solution to the objective function by utilizing an initial guess, ${}^G_B \mathbf{q}_0$ and a step-size, μ . This yields the estimate for the quaternion gradient, $\mathbf{q}_{\nabla,t}$. The filter then calculates the time rate of change in the body quaternion, $\dot{\mathbf{q}}_t$, as measured from the gyroscopes, $\mathbf{q}_{\omega,t}$. By removing the gyroscope error, β from the estimated quaternion error, $\dot{\mathbf{q}}_{\epsilon,t}$ along the gradient and numerically integrating the value over the sampling interval, Δt , the filter yields a quaternion estimate via:

$$\mathbf{q}_t = \mathbf{q}_{t-1} + \dot{\mathbf{q}}_t \Delta t \quad (2.22)$$

$$= \mathbf{q}_{t-1} + (\dot{\mathbf{q}}_{\omega,t} - \beta \dot{\mathbf{q}}_{\epsilon,t}) \Delta t \quad (2.23)$$

$$= \mathbf{q}_{t-1} + \left(\dot{\mathbf{q}}_{\omega,t} - \beta \frac{\nabla f}{\|\nabla f\|} \right) \Delta t \quad (2.24)$$

While the driving principle behind this filter is straight forward, the GDA has a convergence time that can lead to a high initial error during the algorithm start. The convergence time can be tuned using the β value and the overall error reduced by implementing the sensor models in Equations 2.5 and 2.14. The Madgwick filter is also computational and memory intensive as it requires many matrix multiplication operations and Jacobian calculations. On low-power, embedded platforms this can cause performance issues that may not be present with simpler models like the Fast Complimentary Filter [CITE - FCF].

2.4 Project Management Techniques

Any engineering project requires careful consideration to the design methodology and management structure. The methodology and structure dictates how quickly a project will progress and how flexible it is to changes. Over the course of this thesis, several questions were considered: "how will the project adjust to design changes?" and "how can the project be structured such that it can be easily picked up in the future?"

2.4.1 Version Control

One of the first things considered was tracking changes in software and hardware using a version control system (VCS). Version control allows a team to track changes in source files and restore previous editions, or branch out to experiment with new features. Git ⁵ is the industry standard VCS and was used extensively for the software and hardware of

⁵<https://docs.github.com/en/get-started/using-git/about-git>

this project. Online services like GitHub allow developers to manage complex repositories of source code with version control, branches, issue tracking, and so much more. Typically, features could be introduced as an issue in the source repository, and then a specific branch created that implemented that feature. When completed and tested, the feature could then be re-introduced into the main code base⁶. The same practice could be followed for bugs in the software, ensuring that everything is tracked and fixes did not balloon in size and break the working code. GitHub also provided a project planning tool⁷ that aided in building out firmware releases and testing.

The hardware was designed in Autodesk Fusion360⁸ which has a built-in version control system more simplistic than Git. This was still utilized extensively and as a safety precaution, the final design revisions were also backed up to a GitHub repository.

With version control, the project was able to manage many different versions of firmware that were developed over years and various hardware platforms. The tools available made design changes easier to implement and track and allows for new team members to easily come on-board in the future and have ready access to the latest firmware and hardware revisions, and the entire design history.

2.4.2 Systems Engineering

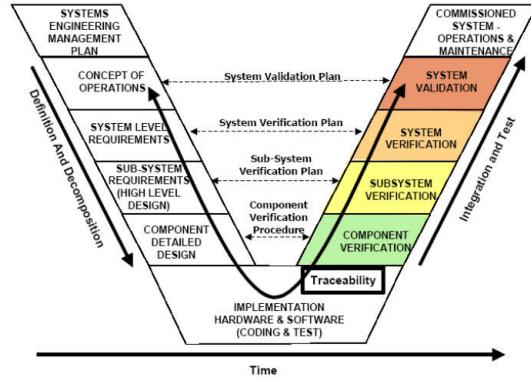
The project was designed using a hybrid systems engineering process that blended the V-methodology and the design spiral. The purpose of these two methods is to constantly iterate on a product until it has reached completion (i.e. the stakeholder requirements are met and the customer is satisfied). At the start of the process, the key problem has to be identified and then requirements drawn up for the system. At the beginning, these requirements start

⁶<https://docs.github.com/en/repositories/creating-and-managing-repositories/best-practices-for-repositories>

⁷<https://github.blog/2022-02-11-getting-started-with-project-planning-on-github/>

⁸<https://www.autodesk.com/products/fusion-360/overview?term=1-YEAR&tab=subscription>

Figure 2.14: The system engineering V-method showing the flow of the project. Courtesy of Allouis et. al [CITE].



out broad and are best identified with stakeholders (the people who will be directly impacted by the project). Then, it is up to a systems engineer to expand the overarching requirements into capabilities and then subsystem- and component-level requirements.

The V-Methodology details a process from design to integration to verification and validation (Figure 2.14). Each step on the left side of the "V" (the start of the project) is verified by the same steps on the right side (the end of the project). The idea is that by building good requirements at the start and verifying them at the end, the project will be "successful" with clear expectations outlined at the beginning and a traceability matrix (Appendix G) to track requirement progression and achievement. The engineering and management teams can follow the "V" from left to right and ensure that their product is on-track while mitigating "feature-creep". As the product is developed, the design volleys between the left and right sides, slowly iterating back up to the top of the V shape. This process is described in more detail in Allouis et al [CITE - FACILITY FOR V and V].

Agile Methodology

The Agile methodology is a relatively new concept introduced in the mid-1990's that provided a radical new method for doing software development. Instead of working in siloed teams based on a broad set of requirements and a large, expansive Gantt chart, programmers were

encouraged to start working together in smaller teams to tackle smaller issues and constantly test and present their progress. Team leaders would have an easier time managing the smaller teams and workloads and project managers would have a better way of tracking product progress. The scrum methodology was also introduced that radically changed the team dynamic and has evolved to improve efficiency in a wide variety of industries [CITE - scrum].

The Agile methodology was implemented on the project so that tasking loading remained light and features could be implemented sprint-by-sprint. The testing recommended by the Agile method meant that products could be more quickly verified and validated against, allowing for quicker iteration. This meant the project could adapt quickly and provide a framework for new people to start contributing to the project, if necessary.

Chapter 3

How I Built It

3.1 Stakeholders

As with any project, stakeholders are a crucial part of the design process. Stakeholders drive overarching requirements that the device must achieve in order to be accepted for operation. On an individual level, the committee overseeing this thesis are the major stakeholders as they have a vested interest in the success or failure of the project. However, some organizational-level stakeholders have also expressed interest in the project and provided some additional requirements. A summary of the stakeholders is provided in Table 3.1.

3.1.1 Hands-On Users

Thetis is intended to be used by college students that are at least sophomore-level and have a basic understanding of instrumentation and electronics. The device is simple to use, requiring the user to flick a switch to turn it on or off, and then press a physical or digital button to start or end logging. Thetis is also designed to accommodate advanced users who need to leverage different features of the device for their research or want to tinker with it. It can also be used by laboratory technicians and instrumentation engineers to aid with data-collection experiments for tracking three dimensional bodies

Table 3.1: A summary of stakeholders of the Thetis device

Stakeholder	Description
Committee members	Individual professors who have expressed interest in the project and have agreed to assist in its development. Specifically, Dr. Wood and Dr. Weaver would like to deploy Thetis on university projects; Dr. Gutierrez has provided many requirements on the performance of the instrumentation; and Dr. Silaghi is interested in its application to autonomous navigation.
Florida Institute of Technology	The university has several classes and projects where Thetis could be useful. The Instrumentation Design and Analysis class used Thetis as a demonstrator for designing PCBs and field experiments. Thetis was also designed with Surf Engineering Analysis in mind for students to have a new open source sensor to experiment with.
NSWC Carderock - Combatant Craft Division (CCD)	This three-letter agency of the government has expressed interest in using Thetis as a testing and evaluation tool for small unmanned crafts

3.2 Design Rationale

Thetis is envisioned as an open-source all-in-one inertial data logging solution for use in research projects. The device incorporates multiple sensors, GPS tracking, and a WiFi-capable microcontroller in order to enable as many features as can be envisioned by the end users. One of the driving considerations was the small footprint. Thetis Revision F is designed to fit into one of the smallest IP67-rated enclosures available on the market [LINK]. This tiny form factor allows it to be inconspicuously mounted to any floating body like surfboards, scale models, or wave buoys without upsetting their inertial characteristics or impeding nominal operation.

3.2.1 Problem Description

I originally envisioned Thetis as the solution to a couple of problems I encountered during my undergraduate studies. For my fluid mechanics laboratory section, we conducted an experiment where we needed to track the rolling of a model pontoon boat as we adjusted its center of gravity. Because the lab lacked any way to directly measure the body's inertial

movement, angle gauges were placed on the hull and the students had to conduct a time motion analysis from a video recording to plot the roll angle frame-by-frame. This was inefficient and made me start thinking about how this could be improved.

The second problem occurred in a class the following semester where we deployed surf boards and pressure gauges into the ocean to monitor a surfer's ride along a wave. We used two different instruments, the Lowell MAT-1 data logger [CITE - LOWELL], and an iPhone 6 inside a life proof case. The Lowell instrument was not designed for this use case. First, the sensor board is housed within a cylindrical body, making properly alignment to a Cartesian (square) coordinate system almost impossible. Second, the sensor only had an accelerometer and magnetometer on board. It reported roll, pitch, and yaw but the roll and pitch were derived from tilt and gravitational acceleration and the magnetometer determined yaw. These values assumed the sensor would be in the vertical position, and we used it in a horizontal position. Suffice to say, we were not using the Lowell properly and could not get reliable or useful information out of it for our analysis.

The iPhones were much better suited as they had a full 9-DOF IMU sensor suite and a great data logger application to get the raw sensor values and sensor fusion results. However, since this was a class comprised of inexperienced undergraduates, we didn't often check that the seal on the phone cases were properly set before going into the water. Within a matter of months, both iPhones were lost because students removed the case and did not replace the water-tight gasket, costing a large sum of money.

Given my experience with these classes, it was clear to me that a low-cost inertial data logging solution could be valuable to the department as a teaching and laboratory instrument.

3.2.2 Mission Statement

Thetis aims to democratize the inertial measurement and tracking space for small scale experiments by implementing an open-source, feature-rich, all-in-one solution to monitoring the movements of bodies.

Table 3.2: A slimmed down traceability matrix for the stakeholder requirements. Verification and Validation status is not shown here. The priority of each requirement is the relative weight of requirement compared with all others.

ID	Description	Weight	Priority	Status
SR 01	The system shall be able to record acceleration, rotation rate, orientation, and position	1	14	TIP
SR 02	The system shall be able to fit into a small IP67-rated, or better, enclosure	1	14	D
SR 03	The system shall be able to be powered by battery for more than 4 hours continuously	1	14	A
SR 04	The system shall be cheaper than \$200 per unit	0.4	6	A
SR 05	The system shall use components that are readily available COTS	1	14	D
SR 06	The system designs shall be open source for modification by students	0.7	10	D
SR 07	The system shall allow users to change settings via interface and/or configuration file	0.7	10	D
SR 08	The system shall communicate extra-device using WiFi and USB	0.3	4	A
SR 09	The system shall have enough on-board storage for 4 hours of continuously logging at 64 Hz	1	14	TIP

3.2.3 Stakeholder Requirements

Interviews with the stakeholders occurred over several months and informed a set of requirements that they determined were necessary for the project's success. These overarching requirements were placed into a traceability matrix (Table 3.2) to track their priority and status throughout the project.

3.2.4 Risk Identification

The following tables explore some of the larger risks associated with each of the stakeholder requirements. Risks are categorized into several types: Technical, Cost, Schedule, Organizational, and Operational. Each risk has an associated consequence on the project's development and a severity rating from "Low" to "High". With every risk, we can apply a mitigation to it to reduce the severity of the consequence.

Technical Risks These are risks that pose a technical challenge to the project such as sensor accuracies, programming difficulty, or component specifications. These risks are likely to impede the development of Thetis by showing up during testing and forcing workarounds after the fact, rather than before.

Cost Risks These risks pose a threat to Thetis's development budget. They are more likely to manifest in the supply chain of Thetis, especially in the wake of the COVID-19 pandemic and the chaos of the chip shortage.

Schedule Risks By delaying the program's development, Schedule Risks are the most notorious and costly. These risks manifest as delays in development or the supply chain and can significantly impact the expected delivery time of the device. They also are prominent across all phases of the design process from planning to testing and are only mitigated, but never eliminated during the design process.

Organization Risks The stakeholders represent various organizations that have an interest in Thetis. This risk type poses a problem for the specific stakeholder organization and their interests. This can be viewed from the political, economical, or temporal points of view as these are important considerations for the stakeholders.

Operational Risks These risks challenge end users of the device while they are testing. Most of them can be mitigated in the design process by working directly with the end users and ensuring their concerns are met. However, the majority of them will only manifest during

testing and must be mitigated after the fact. This could mean substantial cost increases and schedule delays depending on the severity of the risk.

SR 01 - The system shall be able to record acceleration, rotation rate, orientation, and position

Stakeholder:	Thesis Committee				
Rationale:	The system needs to be able to record the inertial characteristics of a floating body				
Fit Criterion:	This will be accomplished using a 9-DOF IMU and GPS receiver with accuracies not to exceed one standard deviation of a reference source				
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	The IMU and/or GPS will report measurements that have a high margin of error and little consistency	Worthless data for analysis	Medium	Selection of sensors that have decent accuracy and low drift. Use a sensor calibration model to improve reported sensor accuracy	Low
Cost Assessment	Chip shortage as a result of the COVID-19 pandemic.	Unable to find appropriate components. Any found components are prohibitively expensive	Medium	Find components that are in stock and order in bulk. Rework design based on available components.	Medium
Schedule Assessment	Calibration is time consuming to implement, verify, and validate	Schedule overrun trying to tune the algorithms. Inaccuracies introduced through improper tuning.	Medium	Good programming practices to make tuning easier during testing. Good understanding of process and mathematics.	Low
Organizational Assessment	Lack of subject matter experts	Improper implementation and design of sensor and embedded systems.	Medium	Thorough background review and study of embedded system design Consult with external experts.	Low
Operational Assessment	Unreliable sensors.	Device fails and does not recover during testing; lost data	Medium	Reliability analysis and testing required. Consult and follow FMECA.	Low

SR 02 - The system shall be able to fit into a small IP67-rated, or better, enclosure

Stakeholder Information					
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	Unable to find a small enough IP-rated enclosure	Need to increase size that may violate size restrictions	Medium	Extensive searching of online retailers of catalogs	Low
Cost Assessment	Small IP-67 rated enclosures are a niche item that may have to be custom-ordered and therefore expensive.	Enclosures may be prohibitively expensive and drive up program costs	Low	Find reputable suppliers and order enclosures in bulk, if possible.	Low
Schedule Assessment	If the enclosures have to be custom ordered, there may be an extended lead time for delivery	Schedule overruns	Medium	Find reputable suppliers and order in bulk well ahead of time. maintain a stock of several enclosures as spares.	Low
Organizational Assessment	N/A	N/A		N/A	
Operational Assessment	IP-rating is not valid. Enclosure is not fully sealed. Operation exceeds rated limits	Chamber floods, damaging equipment, causing a potential fire hazard and data loss.	High	Reputable supplier and traceable IP-rating. End user vigilance to ensure enclosure is fully closed and sealed.	Medium

SR 03 - The system shall be able to be powered by battery for more than 4 hours continuously

Risk Assessment Summary					
Stakeholder:	Thesis Committee				
Rationale:	The system is expected to deploy for a minimum of 3 hours on a typical deployment. Need margin for battery chemistry and preparation time.				
Fit Criterion:	The battery shall have an appropriate capacity. Device will be tested with extended duration deployment				
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	Device is not turned off before battery voltage is critically low	Battery is damaged and needs to be replaced; fire hazard	Medium	Build the power regulator such that the device auto turns off before battery is damaged	Low
Cost Assessment	Larger capacity batteries are more expensive. Damaged batteries need to be replaced	Cost overruns	Low	Order batteries in bulk and ensure spares are available	Low
Schedule Assessment	If there are no spares, there may be delays in testing schedule	Any critical testing may have to be delayed, resulting in program delays	Low	Order batteries in bulk and ensure spares are available	Low
Organizational Assessment	N/A	N/A	Medium	N/A	Medium
Operational Assessment	Battery does not have enough capacity to support a 4 hour deployment	Data is lost	Medium	Conservative power analysis and choosing an oversized battery. Battery monitoring circuit to force save and shutdown at low voltage levels.	Low

SR 04 - The system shall be cheaper than \$200 per unit

Stakeholder: Thesis Committee					
Rationale: Thetis is designed to be a cheap, expendable unit so that, when damaged by students, it is financially feasible to replace it. It needs to be cheaper than comparable devices.					
Fit Criterion: The Bill of Materials for Thetis shall not exceed the required threshold.					
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	Cheaper components (sensors) that are not as precise or accurate.	Less accurate data that may not be useful for analysis	Medium	Use a sensor calibration model to improve reported sensor accuracy	Low
Cost Assessment	Chip shortage as a result of the COVID-19 pandemic. Small IP-67 rated enclosures are a niche item that may have to be custom-ordered	Parts significantly inflate the Bill of Materials cost	Meidum	Source from reputable suppliers and purchase in bulk	Medium
Schedule Assessment	N/A	N/A		N/A	
Organizational Assessment	Bill of Materials cost exceeds requirement limit. Development costs become too expensive	Invested stakeholders may pull support. Hardware development slows or stops	High	Purchase materials in bulk, when possible. Use proper development techniques and analysis to reduce the number of hardware iterations and cost	Medium
Operational Assessment	Cheaper, not as reliable parts.	Unexpected and unrecoverable failures during deployments	Medium	Reliability analysis and testing required. Consult and follow FMECA.	Low

SR 05 - The system shall use components that are readily available COTS

System Requirements Analysis					
Stakeholder:	Thesis Committee				
Rationale:	Using commercial off the shelf parts will reduce development time and costs. COTS parts will also make repair and maintenance easier.				
Fit Criterion:	All parts used on Thetis must be available from commercial suppliers and catalogs.				
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	N/A	N/A		N/A	
Cost Assessment	Chip shortage as a result of the COVID-19 pandemic. Small IP-67 rated enclosures are a niche item that may have to be custom-ordered	Parts significantly inflate the Bill of Materials cost	Medium	Source from reputable suppliers and purchase in bulk	Low
Schedule Assessment	Parts are not available. Small enclosure may have a long lead time	Development schedule should be delayed	Medium	Order parts in bulk and ensure spares are available	Low
Organizational Assessment	The small enclosure may no longer be available from a commercial supplier	Forces a new design iteration to fit a new enclosure. Forces end users to adapt to a new enclosure	Low	Work with supplier to ensure supply chain; order in bulk. 3D print older enclosure, if COTS is no longer available	Low
Operational Assessment	N/A	N/A		N/A	

SR 06 - The system designs shall be open source for modification by students

Stakeholder: Thesis Committee					
Rationale: Thetis will need to be maintained while students use it for class projects. Therefore, they will need to have the ability to directly modify the hardware and software.					
Fit Criterion: All of the code and hardware will be published open source in GitHub repositories under an MIT license.					
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	Designs are modified by students who unknowingly break functionality	Disabling or “bricking” Thetis, making it useless. Loss of development history/“good” designs	Low	GitHub tracks all commits and history. “Good” designs are published as release snapshots and cannot be lost.	Low
Cost Assessment	New design modifications require more investment	Program cost increases as students change the design	Low	Ensure that any design modifications go through a design review and adds substantial value to Thetis	Low
Schedule Assessment	New designs may require additional development time	New additions may not be ready in time for when students need them	Low	Ensure a ready supply of spare parts. Proper design practices to ensure students can easily add features	Low
Organizational Assessment	Thetis design may not be patentable and commercialized	Thetis cannot be patented since it is published open source	High	Thetis can still be commercialized by providing board assembly and support services	Low
Operational Assessment	Vulnerabilities are found and exploited by bad actors	User data can be compromised WiFi capability opens an attack vector to local network	Low	Avoid implementation of OTA software updates Use proper WiFi network security techniques.	Low

SR 07 - The system shall allow users to change settings via interface and/or configuration file.

Stakeholder Information					
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	Users can overwrite certain configurations and compromise sensor performance	Thetis will not provide accurate or useful data measurements	Medium	Implement read-only for critical settings and prevent users from overwriting them outside a “factory” mode.	Low
Cost Assessment	N/A	N/A	Medium	N/A	Medium
Schedule Assessment	Settings implementation and API will need to be custom	Development time will need to be spent on a custom API and implementation	Medium	Thorough background research to see if library solutions exist and can be adapted.	Low
Organizational Assessment	Selected ICs do not use a standard communication bus to transfer data to a microcontroller	Future developers will have to study the custom protocol layer in order to understand how to add features	Medium	Read device datasheets carefully to ensure they use standard protocols	Low
Operational Assessment	Configurations are lost or not saved between deployments	Users may lose custom settings and need to reload. Thetis may crash with corrupted configurations	Medium	Backup settings to data storage device. Have default settings config that loads on issue to prevent instability	Low

SR 08 - The system shall communicate extra-device using WiFi and USB

Stakeholder:	Thesis Committee				
Rationale:	A physical USB connection is crucial for programming and debugging Thetis. A WiFi connection would allow for wireless programming and data off-loading				
Fit Criterion:	The microcontroller shall have a USB-compatible PHY and WiFi capability.				
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	WiFi performance (speed/range) is too low and unacceptable	Data offloading is cumbersome. Unable to monitor device while deployed	Medium	Select microcontroller with known good WiFi-performance. Select an appropriate antenna for the situation	Low
Cost Assessment	WiFi is not useful and not fully utilized	Cost per microcontroller is needlessly increased; inflated bill of materials cost	Medium	Verify that WiFi functionality is desired by Stakeholders and end-users	Low
Schedule Assessment	Implementing WiFi functionality is difficult and increases complexity; feature creep.	Increased time to develop front end interface; schedule overruns	High	Agile management to limit feature creep	Medium
Organizational Assessment	Some stakeholders may have electromagnetic emissions restrictions	To comply, WiFi functionality will need to be able to be disabled	Low	Allow WiFi features to be disabled by uploading new firmware without those features. Allow users to toggle this function on/off in runtime using a configuration file.	Low
Operational Assessment	Operators may not have device that can connect to the device over WiFi	Data cannot be offloaded easily in-situ. Wireless monitoring not possible	Low	Ensure that there is always a physical backup for offloading data. Physical lights or other indicators to display system status	Low

SR 09 - The system shall have enough on-board storage for 4 hours of continuously logging at 64 Hz

Risk Assessment Summary					
Stakeholder:	Thesis Committee				
Rationale:	The system will be deployed for several hours and needs to continuously record data. Data rate should be as fast as possible to capture higher frequency oscillations.				
Fit Criterion:	Storage devices shall be large enough to support a full-length deployment and record all data as fast as possible (>64 Hz)				
Risk	Risk Issue	Risk Consequence	Initial Risk	Risk Mitigation	Risk After Mitigation
Technical Assessment	Logger is not able to consistently achieve 64 Hz rate	Significant data loss on higher frequency oscillations	Medium	Utilize fast storage mediums. Implement proper sensor interrupt practices	Low
Cost Assessment	Small flash chips will not be sufficient to store the amount of data	More expensive data storage devices (micro SD cards) will have to be used; increase bill of materials cost	Medium	Purchase storage devices in bulk and at the minimum capacity.	Low
Schedule Assessment	Optimizing the logging functionality will take significant development time to be consistently as fast as possible	Schedule overruns	High	Proper programming practices to optimize code wherever possible; reduce refactors	Medium
Organizational Assessment	N/A	N/A	Medium	N/A	Medium
Operational Assessment	Inertial movements and oscillations will occur at higher frequencies than can be reasonably measured (>16 Hz)	Significant decrease in data accuracy and correlation to real-world motions	High	Allow users to change sample rate in-situ using configuration settings. Validate that sensors can accurately record expected signals.	Medium

3.2.5 Quality Functional Deployment

In engineering, there are a variety of customers that need to have their needs met. These customers can be internal or external to the project and are described in Section 3.1. To ensure that Thetis is designed with the stakeholders in mind, a Quality Functional Deployment (QFD) was performed. A QFD weighs the stakeholder requirements (the “whats”) with the “hows”, or engineering methods, in a weighted design matrix. The QFD analysis is designed to inform engineers what the highest priority methods are based on customer feedback. This gives stakeholders a direct influence on the design in a compact matrix format. The product created by a QFD analysis is the House of Quality (HOQ), which can be seen in Appendix ??

The left side of the matrix defines the customer desires with an associated weight or priority for each. This data is collected by canvassing the stakeholders and getting their feedback on their priorities.

The middle of the matrix is dedicated with the engineering methods. Here, the design engineers add the methods they think are required to fulfill the projects requirements. The numbers throughout the matrix are the strength of the relationship between the requirement and method. The scale is the Fibonacci sequence from 0-8¹ with higher numbers indicating a stronger relationship. For example, SR-03 has a middling relationship with most of the engineering methods and a very strong (8) relationship with the ”Battery Capacity” method. However, it does not correlate at all (0) with the storage capacity method.

The “roof” or top of the matrix is the inter-relationship between the different engineering methods. The grading is slightly different with “-”, “0”, “+” indicating a detrimental, non-existent, or positive relation, respectively, with multiple of each denoting the magnitude. As an example, the cost of Thetis has varying degrees of negative association with each of the

¹0, 1, 2, 3, 5, 8

other engineering methods. The type of IMU, GPS, storage method, battery capacity, and enclosure size all add cost to the project and therefore are detrimental to this method.

The right side of the matrix is part of the competitive analysis where Thetis is compared with the other products in use by stakeholders. Each product is rated on a 1-5 scale of increasing quality for each stakeholder requirement. This rating is produced based on their capabilities and performance as determined by the stakeholders. From the matrix, we can see that Thetis has a strong overall rating with regard to all the requirements where the Lowell MAT-1 struggles with the cost and measurement requirements and the iPhone is strong overall except its price and access for students.

At the bottom of the matrix is the technical performance analysis. The first two rows rate each engineering method with an importance which is described by Equation 3.1 and 3.2. The importance of each method informs the design engineers what they should be focussing on during development and what is most important to the stakeholders. Beneath that is a technical comparison of Thetis and competing products across the engineering methods. This provides a quick breakdown of each product through the lense of the engineering methods, providing some insight to stakeholders of the benefit of Thetis over competitors.

$$I = \sum_{n=1}^N w_n \times R_n \quad (3.1)$$

where I is the engineering method's importance

N is the number of stakeholder requirements

w_n is the weight of a stakeholder requirement, n

R_n is the strength of the relationship between the requirement and method

$$RI = \frac{I_m}{\sum \mathbf{I}} \quad (3.2)$$

where RI is the engineering method's relative importance

I_m is the current method's importance

\mathbf{I} is the vector of all the method's importance

Table 3.12: The House of Quality matrix for the Thetis instrumentation package.

Stakeholder Requirements												
	Weight	IMU	GPS	Storage Capacity	Battery Capacity	Size	Sample Rate	Cost	Connectivity	Thetis	Lowell	iPhone
SR 01	1	8	8	0	0	0	2	0	0	5	3	5
SR 02	1	0	5	5	8	8	0	5	2	5	3	4
SR 03	1	3	5	0	8	5	3	2	3	5	5	5
SR 04	0.4	5	5	2	5	5	0	8	3	5	1	1
SR 05	1	5	5	3	3	5	0	8	5	5	3	1
SR 06	0.7	3	3	3	1	5	0	2	3	5	1	1
SR 07	0.7	8	8	8	0	0	1	0	3	5	5	5
SR 08	0.3	0	0	0	2	2	0	1	8	5	3	5
SR 09	1	3	1	8	0	0	8	1	0	5	5	5
Importance Weight	28.7	33.7	24.5	22.3	24.1	13.7	20.9	17.8				
Relative Importance	0.15	0.18	0.13	0.12	0.13	0.07	0.11	0.10				
Measurement Unit	DOF	Meters	GB	Wh	Cu. cm	Hertz	US Dollars	Ways				
Target Value	9	3	64	1.5	73.7	64	\$ 200.00	3				
Actual Value	9	3	64	1.5	73.7	100	\$ 150.00	3				
Lowell Instrument	6	-	64	1.5	122	64	\$ 900.00	1				
iPhone	9	3	128	6.7	63.8	100	\$ 649.00	2				

3.3 Concept of Operations

Thetis is envisioned to be an all-in-one data logging solution that can provide real-time calculations of body orientation and position in real-time at a low cost. In a normal operating case, Thetis will be placed near the center of gravity of a body and turned on to begin recording. The user can then connect Thetis to a larger wireless network or use its local WiFi access point to monitor the system and adjust configuration settings.

3.4 capabilities

Based on the stakeholder requirements and concept of operations, a series of capabilities were derived that influenced the subsystem and component-level requirements. These capabilities were organized into three categories:

Threshold: The bare minimum features the system needs to have to be considered “successful” or “delivered”.

Reach: Features that enhance the product for the stakeholders and end users that require a moderate amount of effort above a threshold feature.

Stretch: Features that would distinguish the project from competitors and add immense value for stakeholders and end users, but come at a steep development cost in terms of time and money.

The capabilities were organized into a traceability matrix (more information in Appendix G), which is shown in Table ??.

ID	Description	Weight	Priority	Verified?	Validated?	Status
	Threshold					
101	The system must be housed in an IP67-rated enclosure	1	10	Y	Y	D
102	The system enclosure must fit within the volume of 8" x 5" x 1.25"	0.8	8	Y	Y	D
103	The system software and firmware will be fully open-source	0.9	9	Y	Y	D
104	The system shall be able to record inertial measurements of at least 64 Hz	1	10	Y	N	TIP
105	The system shall be able to store data locally for up to 4 hours continuously	0.8	8	Y	N	TIP
106	The system shall be able to operate for up to 4 hours continuously	0.8	8	Y	N	A
107	The system shall have simple human interface mechanism for status and logging	0.7	7	Y	Y	D
108	The system firmware shall be an open-architecture	0.5	5	Y	N	A
109	The system will be fully documented	1	10	N	N	A
110	The system shall use version control software to track changes	1	10	Y	Y	D
111	The operator shall be able to load data off the system	1	10	Y	Y	D
112	The system shall be capable of being assembled by hand using basic soldering tools	0.8	8	Y	Y	D
	Reach					
201	The system shall integrate a GPS for position tracking at least 1 Hz	0.7	6	Y	Y	D
202	The system will have a simple logging and status interface accessible via web terminal	0.3	2	Y	N	A
203	The system shall be able to monitor and report battery state of charge	0.2	2	Y	N	A
204	The system will have configurable settings that can be set by the user	0.4	3	Y	Y	D
205	The system can operate in a Wi-Fi access point or client mode	0.2	2	Y	N	A
206	The sensor measurements shall incorporate a calibration model	0.5	4	Y	N	TIP
	Stretch					
301	The system file storage shall be accessible via web interface	0.1	0.8	Y	N	A
302	The system shall have a backup storage option in case primary storage fails	0.1	0.8	N	N	R
303	The system will use a microcontroller capable of basic machine learning	0.05	0.4	N	N	UR
304	The system shall be able to log in a burst mode for up to 24 hours	0.05	0.4	N	N	UR

3.5 Conceptual Design

The general design of the Thetis instrumentation package was built around the principles of capability, manufacturability, and affordability. The parts selected for the design had to offer a suite of capabilities that would enable a wide range of projects. While Thetis was designed primarily for monitoring the movement of floating bodies, it can be adapted for use in general robotics, embedded AI research, and other datalogging tasks. Thetis is also envisioned to be continued by other endeavoring students, so it has to be easily manufacturable with a single-sided design that can be assembled by hand using tweezers and a hot-air gun. All of this had to be designed with low-cost readily-available components in mind. Thetis was originally designed at the start of the global chip shortage in 2020, so parts were limited in choice, especially the IMU. It is also assumed that Thetis will be lost at some point during testing. This is just an inevitability during field work, so keeping the cost down would also reduce the impact of such a loss on a project's budget.

3.5.1 System Requirements

To guide Thetis' design, a series of system requirements were generated that would need to be met in order to declare the design as "delivered". These requirements were derived from the House of Quality (Table 3.12) and the stakeholder requirements (Section 3.2.3). Each requirement is categorized into subsections and given a numerical designation. The requirement is then given as a clear statement of fact with only one consideration given per requirement. A comment may be associated with each requirement that provides additional context or relevant information. These requirements arranged within a traceability matrix, explained in-depth in Appendix G

Below is a table describing the system requirements for the Thetis instrumentation package.

Section	Design Considerations/Requirement	Capability	Comments	Verified?	Verification Method	Associated Test Number	Verification Comments	Validated?	Status
1	System								
1.1	Operation								
1.1.1	The system shall record inertial measurement data at least 64 Hz	104	Expected motion characteristic to be about 16 Hz; to gather accurate sinusoidal image requires at least 4x signal rate	Y	Demonstration	AT/GT/MT	2023-02-11: The current version of the software samples at the maximum possible rate (90-100 Hz). A future update will enable more uniform update rates	N	TIP
1.1.2	If GPS is present, the system shall record position, velocity, and course data at least 1 Hz, when fix is valid	201		Y	Demonstration	GPST		N	TIP
1.1.3	The system shall record all data onto onboard storage of at least 1 GB	105		Y	Inspection			Y	D
1.1.4	Data contained within the onboard storage shall be able to be retrieved either physically, over the air, or through a serial connection	111		Y	Inspection	Any	2023-02-11: As of writing, data has been retrieved physically (by removing the uSD card) and OTA. Serial not tested yet.	N	TIP
1.1.5	The system shall fit within a water-tight enclosure	101		Y	Inspection			Y	D
1.1.6	The system shall be capable of being mounted to a surface with bolts, screws, or zip ties			Y	Inspection			Y	D
1.1.7	The system shall be operable by a single person			Y	Demonstration			Y	D
1.1.8	The system shall be able to load user configurations from a file in the onboard storage unit when booted	204		Y	Test	Any		Y	D
1.1.9	The user shall be able to overwrite the user configurations through the serial console or over the air	204		Y	Test	Any		Y	TIP
1.1.10	The user shall be able to turn on and off the system using a physical switch	107	Power supply for the regulator should automatically be chosen when the switch is flipped. USB power has priority over the battery.	Y	Test	Any		Y	D
1.1.11	The system shall be able to power down peripherals (sensors, indicators, radios, etc.) when entering a sleep mode	304	This can be done either by setting each chip individually to their sleep or low power states, or by turning off a dedicated peripheral voltage regulator	N	Test		We can measure the power draw by supplying 5V through the test pads and monitoring current draw in the different modes	N	R
1.1.12	When plugged into a USB data cable the system shall enter a diagnostic mode		Need a sense line on the V_USB bus to detect USB is present	Y	Test	Any		Y	D
1.1.13	When plugged into a USB data cable, the system shall not start the main firmware loop until a diagnostic serial terminal is opened			Y	Test	Any		Y	D
1.1.14	The system shall have an RGB LED that can be used to provide operational feedback to the user without a diagnostic console	107	Different color codes for system states - error codes for failure modes	Y	Test	Any		Y	D
1.2	Sensors								
1.2.1	The system shall utilize an IMU with at least 6 DOF	104		Y	Inspection	AT/GT/MT		Y	D
1.2.2	If the IMU has only a gyroscope and accelerometer, the system shall integrate another sensor to determine heading	104	In GPS-enabled environments, the GPS sensor will suffice. Otherwise, a 3DOF magnetometer will need to be added	N	Inspection	MT		N	R
1.2.3	Accelerometer shall be capable of measuring accelerations up to $\pm 24g$	104		Y	Inspection			N	A

1.2.4	Accelerometer shall have a sensor resolution of at least 12 bits at $\pm 8g$ sample range	104		Y	Inspection			N	A
1.2.5	Gyroscope shall be capable of measuring rotation rates of up to 2000 deg/sec	104		Y	Inspection			N	A
1.2.6	Gyroscope shall have a sensor resolution of at least 12 bits at ± 500 deg/sec sample range	104		Y	Inspection			N	A
1.2.7	Magnetometer shall be capable of measuring magnetic fields up to $\pm 8G$	104		Y	Inspection			N	A
1.2.8	Magnetometer shall have a sensor resolution of at least 12 bits at $\pm 2G$ sample range	104		Y	Inspection			N	A
1.2.9	When in realistic conditions, the GPS shall report position data with an accuracy of at least $\pm 3m$	201		Y	Analysis			N	A
1.2.10	The GPS shall be capable of reporting NMEA-encoded data at least every 1 Hz	201		Y	Demonstration	GPST		N	A
1.2.11	The polling rate of the sensors shall be configurable by the user through the configuration file and must conform to the sensor's options	204		Y	Demonstration	AT/GT/MT		N	A
1.2.12	The sample rate of the sensors shall be configurable by the users through the configuration file	204	1, 2, 4, 8, 16, 32, 64 Hz options	Y	Demonstration	AT/GT/MT		N	A
1.3 Units									
1.3.1	Unless otherwise specified, internal processing units shall be metric (SI)			Y	Inspection			Y	D
1.3.2	Unless otherwise specified, measurements shall be reported in metric (SI) units			Y	Inspection			Y	D
1.3.3	Unless otherwise specified, internal timing shall be done with microsecond-precision		Arduino: micros()	Y	Inspection			Y	D
1.3.4	Unless otherwise specified, internal timestamping shall be done with POSIX epoch		Seconds since January 1, 1970 @ 00:00 UTC	Y	Inspection			Y	D
1.3.5	Unless otherwise specified, reported timestamps shall be in ISO8601 format with millisecond precision		YYYY-MM-DDTHH:mm:SS.sss	Y	Inspection			Y	D
1.3.7	Unless otherwise specified, reported timestamps shall be in the Universal Time Coordinated			Y	Inspection			Y	D
1.3.8	The user shall be able to override the timestamp format using the configuration file	204		N	Demonstration			N	UR
1.3.9	The user shall be able to override the recorded timezone using the configuration file	204		N	Demonstration			N	UR
1.3.10	Whenever possible the internal RTC shall be synchronized with external time sources like NTP and/or GPS	201		Y				N	TIP
1.3.11	In the absence of a synchronization source like NTP or GPS, the device shall report time in terms of time since power on			Y	Demonstration			Y	D
2 Mechanical									
2.1 Physical dimensions									
2.1.1	The system shall fit within a 8" x 5" x 1.25" space	102		Y	Inspection			Y	D
2.1.2	The system shall not weight more than 500 grams	102		Y	Inspection			Y	D
2.2 Enclosure									
2.2.1	The enclosure shall be rated to withstand at least submersion in 1 meter of water for up to 4 hours	101		Y	Test			Y	A
2.2.2	The enclosure shall not allow any dust to enter it	101		Y	Test			Y	A
2.2.3	The enclosure shall be sealed using a replaceable gasket or o-ring	101		Y	Inspection			Y	D
2.2.4	The enclosure shall not exceed the physical dimensions specified in Requirement 2.1.1	101		Y	Inspection			Y	D
2.2.5	The enclosure shall be able to be bolted, screwed, or zip tied to a surface with at least 2 points of contact			Y	Inspection			Y	D
2.2.6	The enclosure shall have external markings indicating the system's measurement axes and sensor location			Y	Inspection			N	A

2.2.7	The enclosure shall be made of a non-RF blocking material, unless an external antenna is available	205		Y	Inspection			N	A
2.2.8	The enclosure shall be made of a material resistant to continuous submersion in salt water (>25 ppt NaCl)	101		Y	Test			N	A
2.2.9	The enclosure shall be made of a material that can withstand constant exposure to sunlight (UV radiation)	101		Y	Test			N	A
2.2.10	The enclosure shall be capable of withstanding multiple drops without compromising its integrity	101		Y	Test			N	A
2.2.11	The enclosure shall have multiple points on which to mount the instrumentation board	101		Y	Inspection			Y	D
3 Electrical									
3.1 Power									
3.1.1	The system shall operate off a 1S (3.7V nom.) lithium polymer battery			Y	Inspection			Y	D
3.1.2	The system shall use appropriate onboard voltage busses, as necessary			Y	Inspection			Y	D
3.1.3	The system shall not exceed the current draw of the battery			Y	Test	PWT		N	A
3.1.4	The system shall be optionally powered from a USB or other external source			Y	Demonstration			Y	D
3.1.5	In accordance with Requirement 3.1.4, the system will not allow current to flow unregulated from the external source to the battery			Y	Analysis			N	A
3.1.6	The system shall not have multiple power sources being used at once			Y	Analysis	PWT		N	A
3.1.7	In accordance with Requirement 3.1.6, the system shall draw power from the external voltage source, before the battery			Y	Analysis	PWT		N	A
3.1.8	The system shall use low quiescent-current regulators, where feasible			Y	Analysis	PWT		N	A
3.1.9	The system shall provide a battery backup voltage to the GPS module, if supported			Y	Inspection			N	A
3.1.10	The system shall provide a battery backup voltage to the RTC, if supported			N	Inspection			N	R
3.1.11	The system shall be able to recharge the battery when plugged into USB power			Y	Demonstration			Y	D
3.2 Mechanical Connections									
3.2.1	Where possible, the system shall be assembled using lead-free solder that passes ASTM standards	112		Y	Inspection			N	A
3.2.2	Components shall be soldered to the PCB following IPC J-STD-001 standards for electrical soldering	112	More information: https://www.protoexpress.com/blog/IPC-J-STD-001-standard-soldering-requirements/	Y	Inspection			N	A
3.2.3	When possible, components shall be placed on a single side of the PCB	112		Y				Y	D
3.2.4	Any PCB designs will be made in accordance with the IPC-2221B standard	112	More information: https://www.protoexpress.com/blog/IPC-2221B-circuit-board-design/	Y	Inspection			N	A
3.2.5	Any board-to-board or board-to-cable connections shall use keyed receptacles that prevent connector reversal	112		Y	Inspection			N	A
3.2.6	Any board-to-board or board-to-cable connections shall use components that are rated for automotive use	112		Y	Inspection			N	A

3.2.7	Any board-to-antenna connectors shall use locking, friction-fit connectors with appropriate strain relief	112	This is meant to ensure that the antenna connectors do not dislodge during use	Y	Inspection			N	A
-------	---	-----	--	---	------------	--	--	---	---

3.5.2 Failure Mode, Effect, and Criticality Analysis

Since Thetis is intended to be used by end users who may not have a solid background in programming or electrical skills, it is important to identify potential flaws in the design and attempt to mitigate them. Each potential flaw is given a probability of occurrence and a weighting from 0 to 10 inclusive for its severity. By multiplying these two values together, a Risk Priority Number (RPN) is created that highlights the severity of a potential failure mode. The RPNs are categorized into three groupings:

LOW These are RPNs with a value below 2.5 and represent little risk to the system. These are denoted with green highlights in the FMECA chart below.

MEDIUM These are RPN's between 2.5 and 7.5 that are represented with yellow highlights below. Typically, these failure modes represent a noticeable threat to the system and the mitigations should be carefully implemented.

HIGH These are the highest risk RPNs between 7.5 and 10 and are represented with red highlights. They are the most severe and should be mitigated at all cost during and after the product development cycle.

The probabilities are generally conjured from past experience, the failure ratings of parts, and other factors that are carefully mapped out and documented over time. However, since Thetis is using hobby-grade materials and failures are not generally well-documented for the parts, the probabilities represented herein are my best guess based on experience. The same applies for the criticality score for each potential failure.

Reference Number	Related Requirement	Failure Mode	Type of Failure	Cause of Failure	Effect of Failure	Before			Mitigation of Failure	After		
						Likelihood	Criticality	RPN		Likelihood	Criticality	RPN
1a	1.1.3	uSD card not present	Operator Error	uSD card is not inserted into the receptacle	System will be unable to log data to a uSD card for storage and off-loading	0.50	9	4.50	Go into an error state when uSD is not detected. Prompt user to override by storing data on backup flash storage - notify them that doing so will dramatically limit their deployment time	0.50	7	3.50
1b	1.1.3	uSD card is broken	Hardware Failure	The uSD card is damaged and inoperable	System will be unable to log data to a uSD card for storage and off-loading.	0.10	9	0.90	Go into an error state when uSD is not detected. Prompt user to override by storing data on backup flash storage - notify them that doing so will dramatically limit their deployment time.	0.10	9	0.90
					Depending on the physical damage to the card, an electrical short may develop, leading to excess heating and power draw	0.10	9	0.90	Protect the SD card with a strong receptacle. Have fuses that can limit the current drawn in the event of a short circuit	0.05	9	0.45
2	1.1.10	The power switch is non-operable	Hardware Failure	The switch is melted and locked into a certain setting. Debris or corrosion jammed the switch into its current settings	The system will be unable to turn on	0.01	10	0.10	Ensure that components are protected during use by strong case.	0.01	10	0.10
					The system will be unable to turn off	0.01	1	0.01	Ensure that components are able to withstand intended soldering conditions (especially temperature)	0.01	1	0.01
3	3.1.11	The battery does not charge when plugged into USB power	Hardware Failure	Internal failure of the IC. Faulty connections	The system will be unable to charge the battery by itself	0.10	1	0.10	Ensure that the battery is easily removable and an external charger is available for it	0.10	1	0.10
					Physical damage to the charging IC	0.01	1	0.01	Have fuses on the board that can limit the current drawn in the event of a short circuit. Protect the IC from physical damage with a case	0.01	1	0.01
4a	1.1.2	The GPS does not receive a valid satellite fix	Radio Failure	The radio antenna does not have a clear LOS to the sky	The system will be unable to locate itself in geographic coordinates	0.90	5	4.50	Ensure that the GPS radio antenna has a clear line of sight to the sky. Ensure that antenna is clear of concrete or metal obstructions.	0.01	5	0.05
					The system will not be able to coordinate the system time to UTC reported by satellites	0.90	2	1.80	Allow the user to overwrite the current RTC setting via configuration settings.	0.90	1	0.90
4b	1.1.2	The GPS does not send or receive data to the microcontroller	Software Failure	The GPS firmware has crashed due to a buffer overflow error	The system will not be able locate itself in geographic coordinates	0.40	5	2.00	Integrate a software reset function for the GPS radio Adjust the GPS report rate and baud rate to be more stable	0.40	2	0.80
					The system will not be able to coordinate the system time to UTC reported by satellites	0.40	2	0.80	Allow the user to overwrite the current RTC setting via configuration settings.	0.40	1	0.40

5a	3.2.7	The microcontroller does not broadcast a WiFi Access Point station	Radio Failure	The radio antenna is not properly connected to the surface mount connector	The system will not be able to connect to outside devices and users will not be able to configure the device remotely	0.60	5	3.00	Ensure that the antenna connectors are securely mounted to any board connectors. Minimize cycle times of plugging/unplugging the connectors to maximize usable lifespan	0.20	5	1.00
	2.2.7			The radio antenna is mounted inside a RF-blocking material		0.10	5	0.50	Ensure that the enclosure is non-RF-blocking, or the antenna is mounted externally	0.10	5	0.50
			Hardware Failure	The WiFi PHY hardware inside the microcontroller fails to initialize		0.05	5	0.25	Ensure that the microcontroller is not damaged and does not exceed its maximum operating limits for power, temperature, humidity, etc.	0.05	5	0.25
			Software Failure	The WiFi PHY firmware inside the microcontroller crashes		0.05	5	0.25	Ensure that the microcontroller code does not crash due to run-time errors and the web server cannot be disrupted easily. Implement a software check that resets the system after a hanging failure.	0.01	5	0.05
				The WiFi AP is not initialized with the correct settings		0.30	5	1.50	Ensure that the configuration file is correct. Allow users to change settings in-situ	0.10	5	0.50
5b	3.2.7	The microcontroller does not connect to a WiFi network	Radio Failure	The radio antenna is not properly connected to the surface mount connector	The system will not be able to connect to outside devices and users will not be able to configure the device remotely	0.60	5	3.00	Ensure that the antenna connectors are securely mounted to any board connectors. Minimize cycle times of plugging/unplugging the connectors to maximize usable lifespan	0.20	5	1.00
	2.2.7			The radio antenna is mounted inside a RF-blocking material		0.10	5	0.50	Ensure that the enclosure is non-RF-blocking, or the antenna is mounted externally	0.10	5	0.50
			Hardware Failure	The WiFi PHY hardware inside the microcontroller fails to initialize		0.05	5	0.25	Ensure that the microcontroller is not damaged and does not exceed its maximum operating limits for power, temperature, humidity, etc.	0.05	5	0.25
			Software Failure	The WiFi PHY firmware inside the microcontroller crashes		0.05	5	0.25	Ensure that the microcontroller code does not crash due to run-time errors and the web server cannot be disrupted easily. Implement a software check that resets the system after a hanging failure.	0.01	5	0.05

				The WiFi connection is not initialized with the correct settings	0.30	5	1.50	Ensure that the configuration file is correct. Allow users to change settings in-situ	0.10	5	0.50	
6	3.2.2	The I2C devices are not initialized properly	Hardware Failure	A solder joint is not connected properly to one or multiple devices	0.05	10	0.50	Ensure that the device is properly assembled and tested before deployment	0.05	10	0.50	
				One or more devices has physically broken	0.01	10	0.10	Ensure that the device is properly protected by a drop-resistant case Ensure that the device is stored in accordance with manufacturer recommendations	0.01	10	0.10	
			Software Failure	The I2C bus is not properly initialized on the device	0.30	10	3.00	Ensure that the SDA and SCL pins are properly set in the board configuration file. Ensure that "Wire.begin((int) SDA, (int) SCL);" is called before I2C sensor initialization begins	0.01	10	0.10	
				One or more devices have reached a soft failure state	0.05	10	0.50	Have the microcontroller be able to soft reset each device Notify the user of failure modes and advise them to hard restart the device	0.05	3	0.15	
	7	1.1.12	The device is not put into a diagnostic mode when a USB cable is inserted	Hardware Failure	One or more of the resistors in the sense divider are not the correct value	0.01	3	0.03	Ensure that the board is assembled with the appropriate parts and that the connections are secure	0.01	3	0.03
	8a	1.1.14	The NeoPixel LED does not come on during system start	Hardware Failure	The MOSFET enabling current flow to the component may not be open	0.20	1	0.20	Check that the NEOPixel_EN line is LOW (0V)	0.20	1	0.20
					A solder joint is not connected properly to the NeoPixel or related components	0.10	1	0.10	Ensure that the device is properly assembled and tested before deployment	0.05	1	0.05
					One or multiple LEDs within the NeoPixel are burned out	0.30	1	0.30	Ensure that the current-limiting resistor to the NeoPixel is greater than to equal to the value specified in the schematic	0.05	1	0.05
8b	1.1.14	The NeoPixel LED does not display the proper colors	Hardware Failure	One or multiple LEDs within the NeoPixel are burned out	0.30	1	0.30	Ensure that the current-limiting resistor to the NeoPixel is greater than to equal to the value specified in the schematic	0.05	1	0.05	
				Software Failure	0.05	1	0.05	Ensure that the colors specified in the firmware are in the proper color format	0.01	1	0.01	

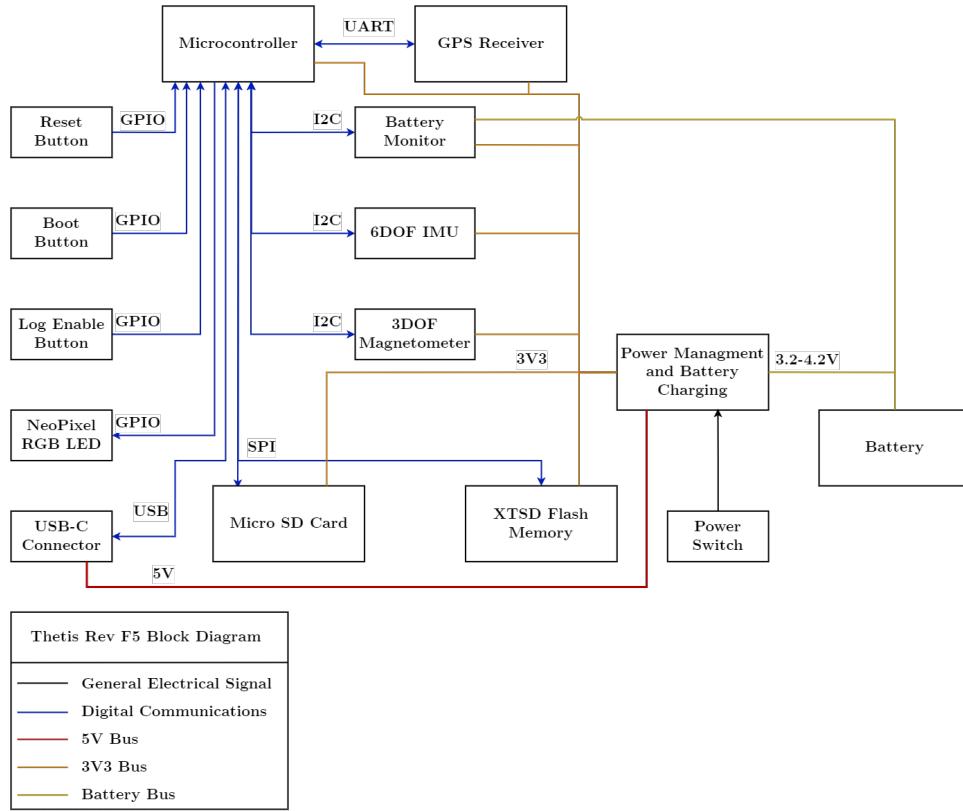
9	1.1.3	The diagnostic file logger does not initialize properly	Operator Error	uSD card is not inserted into the receptacle	System will be unable to log data to a uSD card for storage and off-loading	0.50	9	4.50	Go into an error state when uSD is not detected. Prompt user to override by storing data on backup flash storage - notify them that doing so will dramatically limit their deployment time	0.50	7	3.50
			Software Failure	The CS line is not properly selected in the firmware		0.01	9	0.09	Ensure that the SD_CS variable in the firmware is properly configured to the schematic.	0.01	7	0.07
10	1.1.8	Secondary flash storage (SPIFFS) is not initialized	Hardware Failure	The SPI bus is having electrical issues, or CS line is not set correctly. The microcontroller may be physically damaged	System will be unable to load settings from the on-board configuration file.	0.10	9	0.90	Warn user of a fault with secondary flash storage; enter failsafe state	0.10	9	0.90
					System will be unable to load webpage information for serving the dashboard and UI	0.10	1	0.10		0.10	1	0.10
			Software Failure	The SPIFFS memory registers may not be initialized	System will be unable to load settings from the on-board configuration file.	0.10	9	0.90	Warn user of a fault with secondary flash storage; enter failsafe state	0.10	9	0.90
					System will be unable to load webpage information for serving the dashboard and UI	0.10	1	0.10		0.10	1	0.10
12a	1.1.8	The configuration file cannot be loaded from secondary storage	Hardware Failure	The SPI bus is having electrical issues, or CS line is not set correctly. The microcontroller may be physically damaged	System will be unable to load settings from the on-board configuration file.	0.10	9	0.90	Warn user of a fault with secondary flash storage; enter failsafe state	0.10	5	0.50
						0.10	9	0.90		0.10	5	0.50
			Software Failure	The configuration file may not be loaded into the SPIFFS memory registers	The user configuration settings will be unable to be loaded	0.60	9	5.40	Ensure that the filesystem image is flashed to the device before deployment	0.50	9	4.50
						0.10	9	0.90		0.01	9	0.09
13	1.3.10	The microcontroller's internal RTC is not able to synchronize properly	Radio Failure	The GPS does not have a valid fix to the satellite constellation	The on-board report time will not be synchronized to UTC, rather it will be in terms of time since power-on	0.80	2	1.60	Attempt to synchronize the RTC every minute Advise the user of the synchronization status by using the diagnostic logs	0.80	1	0.80
			Software Failure	The RTC synchronizes to the GPS time, but incorrectly counts the milliseconds between updates	The on-board report time will not be correct. For instance, the time may be 2023-02-23T09:00:00.120000 where the clock is misreporting the 2 minutes that have passed.	0.90	9	8.10	Regulararly update the RTC milliseconds in the main loop execution	0.01	9	0.09

14	1.1.3	The data logger does not initialize	Operator Error	uSD card is not inserted into the receptacle	System will be unable to log data to a uSD card for storage and off-loading	0.50	9	4.50	Warn user of a fault with secondary flash storage; enter failsafe state	0.50	7	3.50
			Hardware Failure	The uSD card is damaged and inoperable	System will be unable to log data to a uSD card for storage and off-loading.	0.10	9	0.90		0.10	9	0.90
					Depending on the physical damage to the card, an electrical short may develop, leading to excess heating and power draw	0.10	9	0.90	Protect the SD card with a strong receptacle. Have fuses that can limit the current drawn in the event of a short circuit	0.05	9	0.45
			Software Failure	The CS line is not properly selected in the firmware	System will be unable to log data to a uSD card for storage and off-loading	0.01	9	0.09	Ensure that the SD_CS variable in the firmware is properly configured to the schematic.	0.10	7	0.70
				The maximum number of data logs have been created	System will be unable to log data to a uSD card for storage and off-loading	0.01	9	0.09	Warn user of a fault with secondary flash storage; enter failsafe state	0.01	9	0.09
15	2.2	The enclosure is not securely closed	Hardware Failure	The enclosure floods when exposed to water	System will become inundated with water and cease working. Board will be destroyed by galvanic corrosion	0.33	10	3.30	Ensure all screws for enclosure are present and properly torqued Do not over torque screws Inspect enclosure to ensure no cracking evident	0.10	10	1.00

3.5.3 Functional Block Diagram

Factoring in the design principles of capability, manufacturability, and affordability, along with the stakeholder requirements listed in Section 3.2.3, a block diagram was constructed that integrated the crucial parts and drew out their interactions. This block diagram shows the data sharing relationships in blue and the power relationships in warm colors. Labels on the specific lines indicate the specific communication protocol or voltage level being used by the line. Explanations for these protocols and acronyms can be found in the Appendix.

Figure 3.1: A block diagram representation of the Thetis RevF5 instrumentation board.

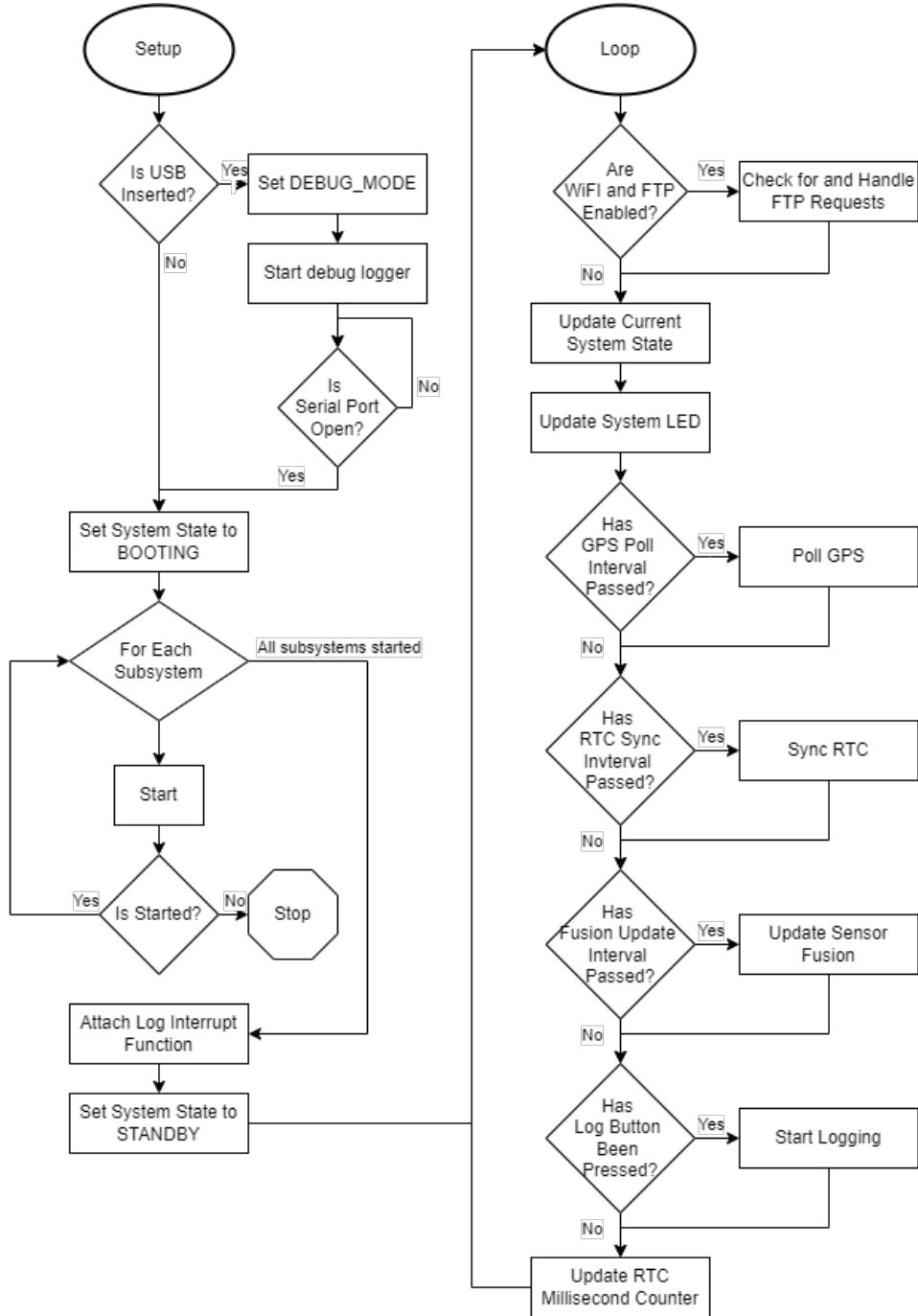


3.5.4 Functional Flow Diagram

Based off the block diagram and interviews with potential end users, a notional flow diagram for the firmware was created. The code was designed to be modular and allow for new features

to be quickly added or removed with separate libraries.

Figure 3.2: A flow diagram representation of the base firmware of the Thetis instrumentation board.



3.6 Brief Revision History

After carefully considering all the design criteria from the stakeholder requirements and system requirements, Thetis began several design revisions. The first design revision, Rev F1, was a functional proof of concept. It incorporated many of the base design features of later Thetis revisions, but relied on a breakout board for the IMU mounted to the bottom of the board and had the battery placed on top. This forced the battery to be smaller than the desired capacity and overfilled the enclosure, causing excessive compression and stress when the enclosure was tightened and secured.

[INSERT IMAGE OF REVISION F1]

The next revision, F2, changed to a new microcontroller version that simplified the circuit while providing the same capability of the previous revision. This version also switched from a Micro SD card receptacle to a soldered SMD flash memory chip. This chip maintained the same capacity requirements from the stakeholders, but took up a fraction of the space on the board. However, since this chip was directly soldered to the board, the only way to offload data was through the microcontroller either through an FTP server, or through the USB interface. This is not ideal - especially in the early testing phases, as those features are more complex to implement programmatically and test data cannot be easily offloaded if there is an issue with the microcontroller. There were also reliability issues which manifest in force on later revisions. This revision also kept the breakout board for the IMU on the back, giving it the same space constraint problem as its predecessor.

[INSERT IMAGE OF REVISION F2]

For Revision F3, some minor changes were made to the diagnostic LEDs and user inputs. Several discrete LEDs for logging, error notification, and low battery, were replaced by a single addressable RGB LED that could flash different colors and patterns to convey the system state or errors. The dedicated log enable button was also removed in favor of using the boot select button for both purposes. However, this method was entirely flawed as the log enable button relies on an interrupt input to trigger the system to begin recording. On the ESP32-S2 microcontroller, an interrupt tied to GPIO 0 (the boot select pin) would cause

the system to crash and force a reset. Therefore, the entire design had to be scrapped due to this design flaw.

[INSERT IMAGE OF REVISION F3]

Revision F4 was the most radical design change of the series. It marked a migration to all SMD components (no breakout board for the IMU) and incorporated all of the design lessons from the previous revision. However, the IMU that was used in previous revisions, the Bosch BNO055² was amongst the many IMUs affected by the Great Chip Shortage and was therefore unavailable. So, this design had to adapt by incorporating a new, previously untested IMU, the STdevices LSM6DSO32³. This new IMU does not have a magnetometer and at the time of its design, they were hard to find in stock at any distributors. So, in anticipation of the Chip Shortage ending, the BNO055 was incorporated to the design, but not populated, giving this board the unique opportunity of having either or both IMUs present - a useful feature for more accurate sensor fusion, but a terrible prospect for software integration. Additionally, this design reintroduced the micro SD card receptacle, but this time, the SMD flash chip was laid out beneath it. This provided the assembler to determine which storage system they wanted: easier to use micro SD, or more resilient SMD flash. Programmatically, these devices work the same, so the software cost was minimal while providing some flexibility. However, this design was also fundamentally flawed as the newly reintroduced log enable button was not integrated into the design properly and the chip select pin for the storage device was connected to an incorrect controller pin. This latter issue forced the microcontroller to crash and reset every time it tried to write to the external flash memory, necessitating a new board revision.

[INSERT IMAGE OF REVISION F4 (BOTH RENDERED AND ACTUAL)]

²<https://www.digikey.com/en/products/detail/bosch-sensortec/BNO055/6136301>

³<https://www.digikey.com/en/products/detail/stmicroelectronics/LSM6DSO32TR/11694177>

3.7 Final Design

Revision F5 is the latest version that realizes all the hardware features Thetis is meant to incorporate while being easy to assemble and use. The BNO055 was fully removed from the system in favor of adding a dedicated magnetometer, the STdevices LIS3MDL⁴. By reorganizing some components, the SMD flash ship and micro SD card receptacle were placed side-by-side enabling both primary and secondary storage services, enabling backups and improving reliability. Then, to improve the battery monitoring capabilities, a battery gauge IC was added. The MAX17048⁵ is a monitoring IC that runs a mathematical model of the battery's discharge curve given its amp-hour capacity and current operating voltage. This data is reported to the microcontroller as the current battery percentage or life remaining - making this a crucial part of monitoring performance and operations of Thetis when deployed. Component types and placements were also streamlined to improve manufacturability and allow the potential for the device to be more easily assembled by a pick and place machine. This also allows students who have never assembled PCBs before to more easily take up the task and build these boards themselves.

[INSERT IMAGE OF REVISION F5 (BOTH RENDERED AND ACTUAL)]

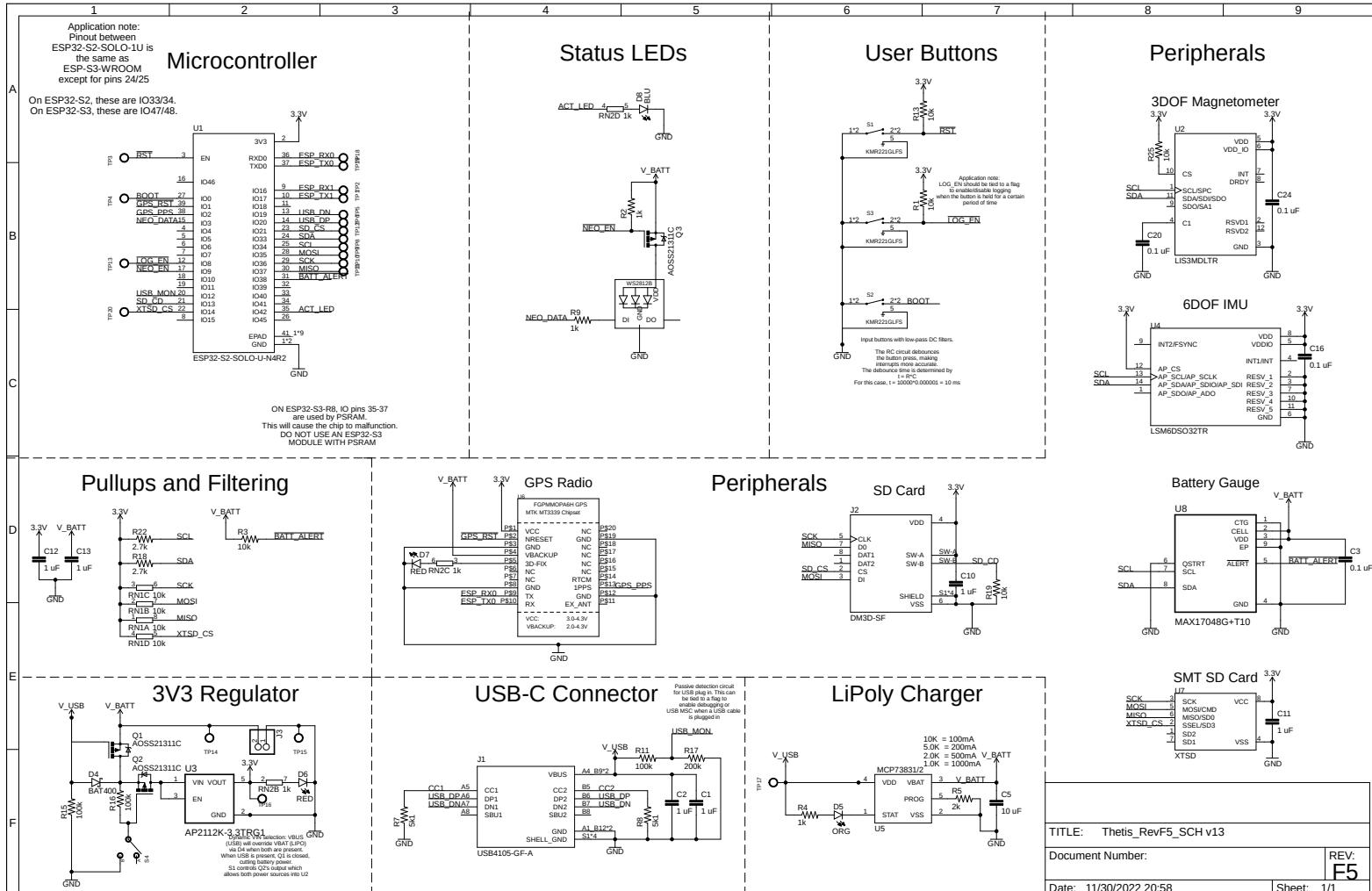
The next sections detail the assembly process in more detail and provide the schematics and PCB layout for Revision F5. The schematics for the previous revisions can be found in the appendix.

⁴<https://www.digikey.com/en/products/detail/stmicroelectronics/LIS3MDLTR/4309733>

⁵<https://www.digikey.com/en/products/detail/analog-devices-inc-maxim-integrated/MAX17048G-T10/3758921>

3.7.1 Schematic

66



3.7.2 Assembly Technique

Thetis was designed with surface mount soldering as the primary assembly technique. This technique differs from traditional soldering as the components are so small that a soldering iron is ineffective in mounting components to the board. Many of the ICs used on the final Thetis revision do not have exposed legs or pads that can be touched with an iron, so solder paste and hot air are required.

[INSERT COMPARISON SHOT OF THROUGH HOLE AND SURFACE MOUNT SOLDERING]

Solder Deposition First, the bare PCB is placed beneath a stencil and secured with tape. It is important that all of the pads on the PCB are aligned with the cutouts in the stencil as the solder paste must be evenly and thoroughly distributed on the board. When the PCB is in place, a line of paste is placed on one side and smoothly dragged along the cutouts with even pressure applied throughout. When the stencil is removed, there should be a healthy deposit of solder paste on each pad as shown in the rightmost picture of Figure ??

[INSERT PICTURE SERIES SHOWING STENCIL LAYING, SOLDER PASTE APPLICATION, AND RESULTANT DEPOSIT]

Component Placement From here, components can be placed onto the PCB carefully using tweezers. Some components measure in at tenths of an inch across and require placement tolerances of hundredths of an inch. Thankfully, in some areas, the surface tension of the solder when melting will forgive slightly misplaced components and drag them into the appropriate spots. However, it is important not to smear the solder as it may inadvertently bridge pads together and create shorts and undesired behavior. Throughout assembling these boards, this particular issue was the most common during the assembly process and occurred regularly on the USB-C receptacle, and IMU ICs where the pins are close together.

Reflow Soldering When all of the components are placed, the PCB can be placed into a reflow oven for baking. The oven used for this process uses infrared radiation to evenly

heat the board through a set profile. The temperature profile is specified by the solder paste manufacturer and gives ample time for the components to heat up and the paste to transform from a viscous mixture, to a solid metallic connection. The entire process from putting the board beneath the stencil to assembled and ready for inspection can take anywhere from 3 hours for a novice assembler, to 45 minutes for a more experienced one.

[INSERT PICTURE SERIES SHOWING COMPONENT PLACEMENT, SOLDERING OVEN, AND FINAL RESULT]

Inspection Following a successful reflow, the board must be inspected for defects. The most common are solder bridges between pads that create undesired shorts and can damage parts. The inspection will also check that parts are in the correct locations and orientations and that no shorts exist between power rails and ground. A microscope and good multimeter are crucial to this process.

If a soldering defect is discovered, the board can be placed onto a small heating plate to melt the solder and remove the part. Then, the part can be removed and the pads on both the component and board are cleaned up using a soldering iron to ensure the proper amount of solder is present. Once the pads are cleaned, either hot air or the heating pad can be used to heat the afflicted area and remelt the solder so the part can be replaced on the board. The board is inspected again and if the issue is cleared, the board is ready for testing.

[INSERT A PICTURE SERIES ON INSPECTING THE ASSEMBLED BOARD, REPLACING A COMPONENT, AND THEN REINSPECTING IT]

Testing When the board passes a visual inspection and there are no power shorts present, the board is plugged into a programming computer via the USB port. The microcontroller is placed into the bootloader mode by holding the boot button and pressing the reset button. This should cause the microcontroller to show up as a USB device on the computer. From here, the latest Thetis firmware can be flashed to the new board. By default, when plugged into USB, the board enters a diagnostic testing mode that prints out the boot and initialization status. By opening a serial console, the assembler can see this process in real time.

If there are any errors while initializing, the firmware will report which device is having a problem. The assembler will then need to do a deeper inspection to find the cause of the error and rectify it.

[INSERT PICTURE SERIES OF BOOTING AND TESTING THE NEW BOARD]

Once the new board boots without issue, the board is ready for deployment and can be used by operators and intended end users.

Bibliography

- [1] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete mathematics*. Addison-Wesley, Reading, MA, 1989.
- [2] D.E. Knuth. Two notes on notation. *Amer. Math. Monthly*, 99:403–422, 1992.

Appendix A

Voltage Divider

The most basic way to measure resistance is with a voltage divider. A voltage divider consists of two resistors in series that go between an input voltage and ground, as shown in the schematic below. The resistive sensor is placed at the R_2 position and we want to read the potential between the top of R_2 and ground, V_{out} to determine the sensor's resistance. This voltage can be determined by the equation:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (\text{A.1})$$

[TODO: INSERT VOLTAGE DIVIDER SCHEMATIC]

A.1 Example: Battery Monitor

Let's say the device you are bringing with you onto the roller coaster uses a 3.3V-logic microcontroller. It can only accept signals up to 3.3V before risking damage to its circuits. If you want your device to be portable, you can use a lithium polymer (LiPo) battery, but these must be carefully monitored to prevent over discharge. A single-cell LiPo battery has a peak voltage of 4.2V which is higher than what the microcontroller can handle. So, we must step down the voltage such that the peak battery voltage matches the peak input voltage of the microcontroller.

This is a perfect application for a voltage divider, but we must first determine what two resistors will be used. To linearize (simplify) the problem, we will arbitrarily set $R_2 = 10\text{k}\Omega$. Then, we can rearrange the problem to solve for R_1

$$\begin{aligned}
 V_{out} &= V_{in} \frac{R_2}{R_1 + R_2} \\
 \frac{R_2}{R_1 + R_2} &= \frac{V_{out}}{V_{in}} \\
 R_1 + R_2 &= \frac{V_{in} \cdot R_2}{V_{out}} \\
 R_1 &= \frac{V_{in} \cdot R_2}{V_{out}} - R_2 \\
 &= \frac{4.2 \cdot 10 \times 10^3}{3.3} - 10 \times 10^3 \\
 &= 2727\Omega
 \end{aligned}$$

To get a resistor that is exactly 2727Ω is very difficult, but we can approximate this value using the much more common $2.7\text{k}\Omega$ resistor without introducing much error into our measurements.

[TODO: INSERT BATTERY VOLTAGE MEASUREMENT CIRCUIT]

Appendix B

Wheatstone Bridge

The Wheatstone Bridge is a configuration of four resistors in two voltage dividers as shown in Figure ?? below that is sensitive to minute changes in resistance. Resistors R_1 and R_2 form the first divider and R_3 and R_4 form the second divider. The bridge has at least three different solutions, depending on the configuration and desired application. For applications where R_1 , R_2 , and R_3 are known to a high precision, then R_4 can be determined to an equally high precision by adjusting R_3 until the voltage potential between points B and D is near 0, i.e. the bridge is "balanced". However, for most embedded applications, like those described in further sections, R_4 cannot be determined by balancing the bridge, so either the general solution or linearization must be used. For simplicity, only the linearization solution will be examined here, the full derivation and explanation for all three solutions can be found in Appendix [APPENDIX].

[TODO: INSERT GENERAL WHEATSTONE BRIDGE SCHEMATIC]

For the linear solution, the circuit adds an operational amplifier between points D and B and assumes that $R_1 + R_2 = R_3 = R_0$ and $R_4 = R_0 + \Delta R$ and the op-amp is an ideal component ($R_{\text{amp}} = 0$). Because of this, the voltage potential at Point B will have a constant value of:

$$V_B = \frac{R_0 V_s}{R_0 + R_0} = \frac{V_s}{2}$$

[TODO: INSERT LINEARIZED WHEATSTONE BRIDGE SCHEMATIC]

Likewise, the op-amp will force the voltage at Point D to have the same voltage as B such that, $V_D = V_B = \frac{V_s}{2}$. This forces a constant current of $\frac{V_s}{2R_0}$ through R_3 and into the sensor. By Ohm's law, the voltage across the sensor will therefore be:

$$\begin{aligned} V_4 &= \frac{V_s}{2R_0} \cdot R_0(1 + \Delta R) \\ &= \frac{V_s}{2} + \frac{V_s}{2}\Delta R \end{aligned}$$

By applying Kirchoff's voltage law, the potential between the amplifier output and ground, V_{out} , is:

$$\begin{aligned} V_{out} &= -V_4 + V_D \\ &= -\left(\frac{V_s}{2} + \frac{V_s}{2}\Delta R\right) + V_D \\ V_{out} &= -\frac{V_s}{2}\Delta R \end{aligned}$$

The measured output voltage now linearly changes with the resistive sensor irregardless of if the sensor changes linearly itself. If the manufacturer of the sensor provides a table or equation that relates the sensor resistance to a real-world value, we can find the sensor resistance via:

$$\begin{aligned} R_4 &= R_0 + \Delta R \\ &= R_0 - \frac{2V_{out}}{V_s} \end{aligned}$$

Appendix C

Analog Measurement

In electronics, there are two different types of signals: digital and analog. Digital signals are either a "high" voltage, or a "low" voltage, representing either a logical 0 or logical 1. These signals are great for threshold measurements such as "is there a cart present at the start of the ride, yes or no?", but cannot express a range of values. Conversely, analog signals are better suited to expressing real-world values that naturally vary from a minimum to a maximum and can be any value in between. Thus, analog signals are prevalent for devices that measuring "real" things such as speed, acceleration, rotation rates, barometric pressures, etc.

[TODO: INSERT DIGITAL AND ANALOG WAVEFORMS]

Most analog sensors are resistive types where their electrical resistance changes to scale with a real-world measurement range. Ohm's Law defines the relation between voltage, (V) resistance (R), and current (I) as $V = IR$. Therefore, if we can measure the voltage drop across a resistive sensor and correlate it with a manufacturer-provided equation or table, we can quantify a real-world phenomenon. The resistance is best measured using a Wheatstone Bridge and an amplifier which can generate a voltage that is directly proportional to the change in resistance, δR

C.1 Analog to Digital Conversion

Now that we can get a sensor reading as an analog (changing) voltage, we need to find a way to translate it into a useful form. Decades ago, the analog voltage from a sensor was plotted onto a chart with respect to time and an analyst could run the conversion point by point. Following the roller coaster example from before, park security is very unlikely to allow you to bring a full chart plotter on the coaster with you, and its even less likely to survive the trip or print a good readout. We also want to work smarter and not harder, so we want a computer to perform the analysis for us. What can we do? Since we live in the digital age, we can convert the analog signal to a digital one and then we can store it on a digital media device such as an SD card then load it directly into a computer program like Excel to analyze it.

This process is done with an analog to digital converter circuit or ADC. The ADC samples an analog waveform at a given frequency and places the voltage level into bins for each measurement. There are 2^N bins in an ADC depending on its resolution, N . For each measurement, the ADC will also encode the voltage reading as a binary number and save it to a register.

This encoded binary number is expressed as "counts" which can be read by a microcontroller or other digital system. The number of counts recorded can be converted back to a voltage value at any time using the equation below. However, this conversion will lose some of the original precision of the analog signal, depending on the resolution of the ADC. A higher resolution will mean a more precise reading.

$$V = V_{cc} \frac{\text{counts}}{2^N}$$

[TODO: INSERT GRAPHIC ON ADC SHOWING BINS AND OUTPUTS, ETC.]

Aside: Registers

Digital values are stored in memory as 1's and 0's. Each bit of memory can be stored in a contiguous piece called a "register". A digital system can access registers of memory to grab values and perform calculations on them. For example, an 8-bit ADC will store the encoding of an input signal in an 8-bit register. A microcontroller can then read this value from the ADC and convert it to a real-world value, or save it to non-volatile memory for logging.

[TODO: INSERT GRAPHIC OF GATED LATCH MEMORY REGISTER]

Appendix D

Digital Communication Methods

Now that we have converted the analog sensor reading into the digital realm, we have to move the data around to use it. There are many communication methods that utilize a variety of protocols, but for simplicity we will be discussing three of the most common in small digital circuits. Each of these methods utilize "Transistor-Transistor-Logic" or TTL voltages, meaning the signals use 0V for a logical 0, and any other voltage above a threshold is considered a logical 1. Each method is also known as a serial communication method as each bit in the message is communicated in series.

D.1 Universal Asynchronous Receive and Transmit

The Universal Asynchronous Receive and Transmit (UART) bus is a full-duplex digital communication method where the receiver and transmitter of information can transmit asynchronously. This is one of the simplest communication interfaces as a controller only needs to pull the data bus to the "low" state at certain intervals while transmitting. There is no addressing, or synchronization required, so the transmitter can just "shout" the bits down the line and does not care if the receiver gets them or not. Since its asynchronous, both the receiver and transmitter must agree on the baudrate, or bits-per-second, beforehand and will set their own internal clocks to the appropriate speed. By timing the arrival of "low" or

"high" signals, the controllers can interpret data in the transmission packet. If the baudrates are not synchronized properly, the interval between the bits will be different and data will become corrupted at either end. Further reading can be done at Analog Dialogue

[TODO: INSERT GRAPHIC OF UART PROTOCOL]

D.2 Inter-Integrated Circuit

The I2C, or Inter-Integrated Circuit, communication bus is most commonly used for slower speed transmissions between integrated circuits on a circuit board. It is a synchronous, multi-nodal, half-duplex topology which means that multiple controllers and responders can exist on the same bus and communicate, but not simultaneously. I2C uses two pins for communication, one for data (SDA) and one for clock (SCK). Unlike UART, I2C devices do not need to agree on a clock speed before communication begins, the clock line speed is adjusted by the controller and all devices automatically synchronize to it. Additionally, I2C devices have a one-byte address that distinguish them and allow for a controller to request data from a specific responder. This limits the number of unique devices on an I2C bus to 255 and can create some problems if multiple I2C devices of the same address are present. For the latter problem, I2C multiplexing solutions exist to mitigate the issue ³.

To begin a message, the controller will initialize a start condition on the line then broadcast the desired device address. This lets all devices on the bus know that the controller wants to talk with a specific device. The controller will then follow-up with a one-byte data frame. Depending on the responder's programming, it will respond with another data frame that the controller will receive. A typical example of this is a controller querying a device for the value of a register in its memory. Further reading can be done at Analog Devices

[TODO: INSERT GRAPHIC OF I2C PROTOCOL]

³<https://www.adafruit.com/product/2717>

D.3 Serial Peripheral Interface

The Serial Peripheral Bus (SPI) is the slightly modified version of the UART protocol. SPI introduces clock synchronization so that a main and subnode do not have to agree on a communication rate beforehand and also allows multiple subnodes to be present on the same bus. By toggling an enable pin, the main node can tell a subnode to ignore or respond to a request on the data line. SPI uses four lines to communicate: Main Out Sub In (MOSI), Main In, Sub Out (MISO), Clock (SCK), and Chip Select (CS). A typical multi-subnode topology is shown in the figure below.

To begin a message, the main node will pull the CS pin for the desired subnode low and send out a message on the MOSI line. When the subnode receives the message, it can respond depending on its programming. In the case of a data storage unit like RAM or an SD card, the subnode will just write data to a specified register from the main node's message. If the subnode is a sensor, it may respond with the value from one or more registers in its memory.

[TODO: INSERT GRAPHIC OF SPI PROTOCOL]

Appendix E

Rotation Matrices

A rotation matrix is a mathematic model for translating one body's inertial reference frame to another, e.g. local frame to a global frame. These are used for Eulerian transformations of vectors (i.e. using roll, pitch, and yaw). To begin deriving these matrices, let's start with a two dimensional rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

If we rotate a vector, $\vec{v} = \langle x, y \rangle$ with a magnitude, r by θ degrees about the Z-axis (as shown in Figure ??), it will arrive at a new coordinate $\langle x', y' \rangle$. We can express \vec{v} in polar form as:

$$x = r \cos \phi \tag{E.1}$$

$$y = r \sin \phi \tag{E.2}$$

Similarly, the rotated vector, \vec{v}' in polar form is expressed as:

$$x' = r \cos(\phi + \theta)$$

$$y' = r \sin(\phi + \theta)$$

[TODO: INSERT FIGURE OF ROTATE VECTOR IN 2D]

Expanding and using trigonometric identities:

$$x' = r(\cos \phi \cos \theta + \sin \phi \sin \theta)$$

$$= r \cos \phi \cos \theta + r \sin \phi \sin \theta$$

$$= x \cos \theta + y \sin \theta$$

$$y' = r(\sin \phi \sin \theta + \cos \phi \sin \theta)$$

$$= r \sin \phi \cos \theta + r \cos \phi \sin \theta$$

$$= y \cos \theta + x \sin \theta$$

We can then express the resultant equations in the form of a 2×2 rotation matrix, R , yielding:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = R(\theta) \begin{bmatrix} x \\ y \end{bmatrix}$$

Example If we have a vessel moving in the \hat{x} direction at some velocity and it encounters a current, \vec{v} coming in from the starboard quarter, what is the current's influence on the vessel?

[TODO: INSERT FIGURE OF BOAT WITH OPPOSING CURRENT]

First, we can use the parallel axis theorem to determine that the current is impinging on the vessel at an angle, θ , relative to its longitudinal (y) axis. We can consider the current vector to be in the global frame and to find its influence on the vessel, we must rotate it to

the vessel's local frame and get the resultant vector, $[v_x, v_y]$. Using a rotation matrix, we can determine that:

$$\begin{bmatrix} v'_x \\ v'_y \end{bmatrix} = R(\theta) \begin{bmatrix} v_E \\ v_N \end{bmatrix}$$

E.1 Three Dimensional Rotation Matrix

In three dimensional space, rotation can be performed about the X-, Y-, or Z-axis. A basic rotation that occurs around a single axis is defined as an "elementary rotation" and given by the following rotation matrices:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To rotate a three dimensional vector from one frame to another, we must select the order of the axes to be rotated about, then multiply the rotation matrices and vector together.

Example What are the components of gravitational acceleration, a_G that affect an airplane in flight at some arbitrary roll (ϕ), pitch (θ), and yaw (ψ)?

[TODO: INSERT PICTURE OF PLANE ROTATED]

First, we define the gravitational acceleration vector as existing in the global North-East-

Down (NED) reference frame, $a_G = \langle a_N, a_E, a_D \rangle = \langle 0, 0, 1 \rangle$. We will also assume that all of the rotations of the plane are relative to the same NED frame. We can then perform a yaw-pitch-roll rotation of a_G to get the gravitational acceleration in terms of the plane's coordinate frame, ${}^G_B a$:

$$\begin{aligned} {}^G_B a &= R_z(\psi) R_y(\theta) R_x(\phi) a_G \\ &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ {}^G_B a &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \theta & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

Appendix F

Kalman Filter

F.1 Introduction

The Kalman Filter is a recursive algorithm introduced in the 1960's as a method to track, estimate, and predict the state of a system and corresponding uncertainties. This filter integrates a dynamic (linear) model of the system, control inputs, measurements, and biases/uncertainties into a single algorithm. This effectively fuses together system inputs and responses and extrapolates what the system is currently doing and expected to do. One key advantage of this algorithm is that it only requires the guess of the previous state to estimate the current state. This massively decreases the memory and processing costs as the history of inputs, measurements, and uncertainties does not need to be remembered or analyzed. However, it does have some limitations when the sensor data is noisy or the control inputs cannot be linearly mapped to the system state. Random errors in the sensor data may cause the filter to behave unpredictably and non-linearity prevents proper fusion entirely. This problem can be solved using an Extended Kalman Filter, but that is out of scope for now.

This section will discuss the Kalman filter in a higher-order method. Many resources around with Internet can discuss the in-depth mathematics governing this filter and you are free to browse them at your leisure. For this guide, my primary concern is to get you acquainted with the Kalman filter and its broader practical applications.

F.2 Kalman Filter in One Dimension

The uni-dimensional Kalman filter is a special, idealized case for the Kalman filter. It is more convenient as a teaching tool as it does not include the complex matrix and vector operations the general multi-dimensional Kalman filter requires. However, this only makes this type useful for tracking and estimating a single variable.

Multiple uni-dimensional Kalman filters can be run in parallel or chained together to mimic a multi-dimensional filter, however, this will unnecessarily increase the computational resources required for the calculations and will reduce the quality of the filtering overall.

F.2.1 The Kalman Gain

In a Kalman filter, the $\alpha - \beta - \gamma$ parameters are calculated at every filter iteration. These parameters are combined together into the Kalman Gain, denoted as K_n , which is the third Kalman equation. The Kalman gain is bounded by: $0 \leq K_n \leq 1$.

$$\begin{aligned} K_n &= \frac{\text{Estimate Uncertainty}}{\text{Estimate Uncertainty} + \text{Measurement Uncertainty}} \\ &= \frac{p_{n,n-1}}{p_{n,n-1} + r_n} \end{aligned} \tag{F.1}$$

If we refresh the State Update Equation with the Kalman gain, we get:

$$\begin{aligned} \hat{x}_{n,n} &= \hat{x}_{n,n-1} + K_n(z_n - \hat{x}_{n,n-1}) \\ &= (1 - K_n)\hat{x}_{n,n-1} + K_n z_n \end{aligned} \tag{F.2}$$

In the above equation, K_n is the weight (importance) given to the measurement. Conversely, $(1 - K_n)$ is the weight given to the estimate.¹

¹**Important note:** When measurement uncertainty is very large, and the estimate uncertainty is small, $K_n \ll 1$, hence big weight to the estimate and small weight to the measurement. When the opposite is true, $K_n \rightarrow 1$, meaning a large weight to the measurement and a small weight to the estimate. This is how the Kalman filter can regulate and smooth out noisy data by knowing the uncertainties.

F.2.2 The Estimate Uncertainty Update Equation

The estimate uncertainty or covariance update equation predicts the uncertainty associated with the current estimate. The estimate uncertainty should approach (converge) to 0 with each filter iteration as the filter improves its guessing accuracy. However, if the measurement uncertainty is large ($K_n \ll 1$), the estimate uncertainty will converge more slowly. The opposite is true if the measurement uncertainty is small. Basically, the more precise your measurements are, the faster the Kalman filter will converge on the best estimate.

$$p_{n,n} = (1 - K_n)p_{n,n-1} \quad (\text{F.3})$$

where $p_{n,n}$ is the estimate uncertainty at the current state

K_n is the Kalman gain at the current state

$p_{n,n-1}$ is the estimate uncertainty of the previous state

F.2.3 The Estimate Uncertainty Extrapolation Equation

Another Kalman equation is how the filter predicts future uncertainties and is called the *Estimate Uncertainty Extrapolation Equation* or the *Estimate Covariance Extrapolation Equation*. Like with the State Extrapolation Equations, this is done with dynamic models and will be unique to every example. If we refer to the previous models from Equations ?? and ??, we can define the Estimate Uncertainty Extrapolation Equations as:

$$\begin{aligned} p_{n+1,n}^x &= p_{n,n}^x + \Delta t p_{n,n}^{\dot{x}} \\ p_{n+1,n}^{\dot{x}} &= p_{n,n}^{\dot{x}} \end{aligned} \quad (\text{F.4})$$

F.2.4 Putting it all Together

First, the filter is initialized with a first guess ($\hat{x}_{0,0}$) and an associated uncertainty ($p_{0,0}$). These values are passed into the dynamic model equations and Equation F.4 to predict the

state at the first measurement ($x_{1,0}$) and the associated uncertainty ($p_{1,0}$). This is shown as Step 0 in Figure ???. Then, we can take a measurement (z_n) and record its uncertainty (r_n), using the latter to determine the Kalman gain with Equation F.1. We can then estimate the current state value using Equation ?? using the recorded measurement and the Kalman gain (K_n). The current state estimate and its uncertainty can then be outputted from the filter and used in applications; this process is shown as Steps 1, 2a, and 2b in the figure below. These values are then fed into the dynamic model to predict the next state and estimate the associated uncertainty (Step 3 in the below diagram). Optionally, these predicted values can be outputted from the model as well, as the application requires. This process repeats for every measurement with n incrementing every time.

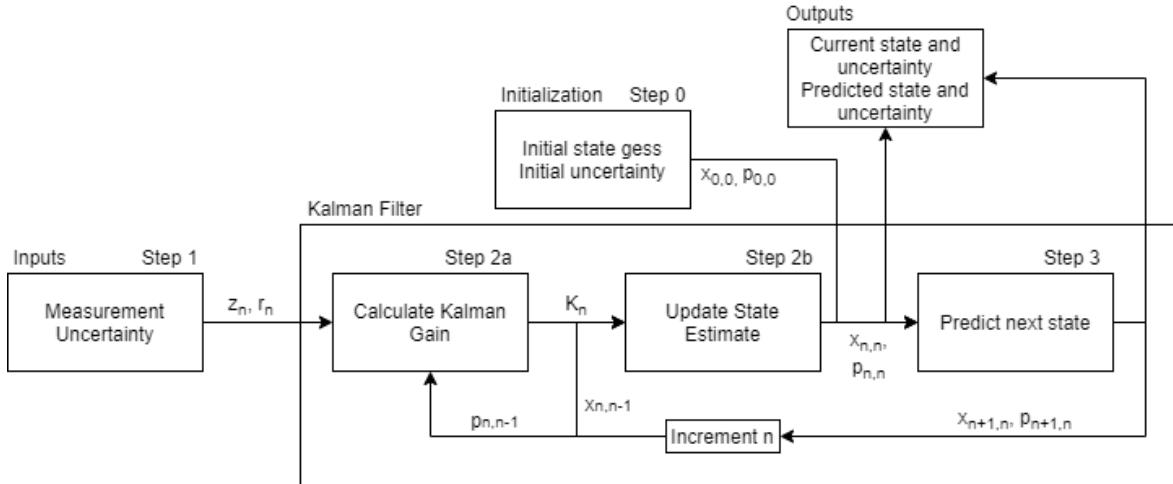


Figure F.1: Process diagram for a Kalman filter.

Example

In this example, we will be estimating the depth of an ROV using a 1D Kalman filter. We will be assuming the ROV depth remains constant and its location in space does not matter or change. For example's sake, we know for an absolute fact that the ROV is at a depth of 50-meters ($x = 50$ m). We will also assume that our depth sensor has a standard deviation of 5-meters ($\sigma = 5$ m), therefore we will have a measurement variance or uncertainty of 25-meters ($r = 25 \text{ m}^2$)

Iteration 0

- 0 : Estimate depth: $\hat{x}_{0,0} = \mathbf{60 \text{ m}}$
: Estimate uncertainty: $p_{0,0} = \mathbf{225 \text{ m}^2}$
3 : Predicted depth: $\hat{x}_{1,0} = \hat{x}_{0,0} = \mathbf{60 \text{ m}}$
: Prediction uncertainty: $p_{1,0} = p_{0,0} = \mathbf{225 \text{ m}^2}$

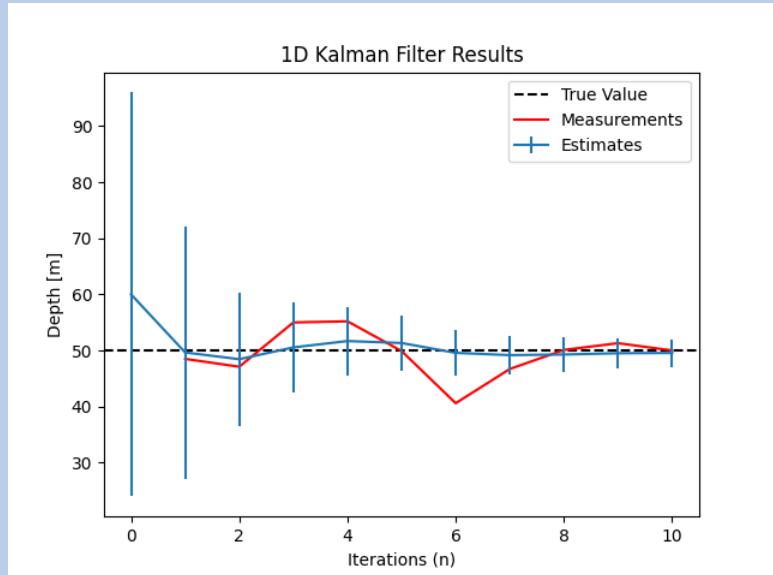
Iteration 1

- 1 : Measure depth: $z_1 = \mathbf{48.54 \text{ m}}$
: Measurement uncertainty: $r_1 = \mathbf{25 \text{ m}^2}$
2 : Kalman gain $K_1 = \frac{p_{1,0}}{p_{1,0} + r_1} = \frac{255}{255 + 25} = \mathbf{0.9}$
: Estimated depth: $\hat{x}_{1,1} = \hat{x}_{1,0} + K_1(z_1 - \hat{x}_{1,0}) = 60 + 0.9(48.54 - 60) = \mathbf{49.69 \text{ m}}$
: Estimate uncertainty: $p_{1,1} = (1 - K_1)p_{1,0} = (1 - 0.9)225 = \mathbf{22.5 \text{ m}^2}$
3 : Predicted depth: $\hat{x}_{2,1} = \hat{x}_{1,1} = \mathbf{49.69 \text{ m}}$
: Prediction uncertainty: $p_{1,0} = p_{1,1} = \mathbf{22.5 \text{ m}^2}$

Iteration 2

- 1 : Measure depth: $z_2 = \mathbf{47.11 \text{ m}}$
: Measurement uncertainty: $r_2 = \mathbf{25 \text{ m}^2}$
2 : Kalman gain $K_2 = \frac{p_{2,1}}{p_{2,1} + r_2} = \frac{22.5}{22.5 + 25} = \mathbf{0.47}$
: Estimated depth: $\hat{x}_{2,2} = \hat{x}_{2,1} + K_2(z_2 - \hat{x}_{2,1}) = 49.69 + 0.47(47.11 - 49.69) = \mathbf{48.47 \text{ m}}$
: Estimate uncertainty: $p_{2,2} = (1 - K_2)p_{2,1} = (1 - 0.47)22.5 = \mathbf{11.84 \text{ m}^2}$
3 : Predicted depth: $\hat{x}_{3,2} = \hat{x}_{2,2} = \mathbf{48.47 \text{ m}}$
: Prediction uncertainty: $p_{3,2} = p_{2,2} = \mathbf{11.84 \text{ m}^2}$

n	Measurements		Current Estimates		Predictions		
	z_n	r_n	K_n	$\hat{x}_{n,n}$	$p_{n,n}$	$\hat{x}_{n+1,n}$	$p_{n+1,n}$
0	—	—	—	60.0	225	60.0	225
1	48.5	25	0.90	49.7	22.5	49.7	22.5
2	47.1	25	0.47	48.4	11.8	48.4	11.8
3	55.0	25	0.32	50.6	8.03	50.6	8.03
4	55.2	25	0.24	51.7	6.08	51.7	6.08
5	49.9	25	0.20	51.3	4.89	51.3	4.89
6	40.6	25	0.16	49.6	4.09	49.6	4.09
7	46.7	25	0.14	49.2	3.52	49.2	3.52
8	50.1	25	0.12	49.3	3.08	49.3	3.08
9	51.3	25	0.11	49.5	2.74	49.5	2.74
10	50.0	25	0.10	49.6	2.47	49.6	2.47



We can see from the figure that the Kalman filter eventually converges on the true value and considerably smoothes out the noisy measurements. The error bars on the estimate

plot (blue) also show that the Kalman filter increases its confidence with every iteration as it converges to the true value. The oscillation shown in the estimates are a result of the gains being slightly off from their ideal value. Tuning the process noise variable (q) can cause the filter to converge quickly and confidently to the true value in fewer iterations.

F.2.5 Process Noise

In the real world, there are uncertainties in the system's dynamic model. Uncertainty is caused by unanticipated changes in the system due to external factors. This can be drift caused by ocean current, wind blowing a rocket to the side, drag, friction, even time dilation in extreme cases. Generally, these uncertainties can be combined into the Process Noise gain denoted by " q " To account for process noise, it must be included in the Covariance Extrapolation Equation (Equation F.4)

If the model is not known to be good or is very noisy, we can increase the process noise gain to reduce the lag error.

$$p_{n+1,n} = p_{n,n} + q \quad (\text{F.5})$$

Further Reading

The book *Kalman Filter: From the Ground Up* [CITE - Kalman Filter Textbook] provides an excellent resource on the intuitive understanding of the Kalman Filter and some of its more advanced applications like multi-variate Kalman Filters, Extended Kalman Filters, and applying Kalman Filters in sensor fusion applications.

Appendix G

Traceability Matrix

A traceability matrix is an artifact within Systems Engineering that tracks the stakeholder requirements, capabilities, system requirements, and testing methodologies. The primary purpose is to give a Lead Systems Engineer other other team member quick access to all of the driving information for the product and trace the requirement lifecycle. Each matrix has eight columns that contain information about the requirement:

ID This is the identification number for the requirement. Stakeholder requirements will start with the prefix: "SR". Capabilities will be triple-digit numbers with 100-level being "threshold", 200-level being "reach", and 300-level being "stretch" capabilities. Component-level requirements will be divided into sections and subsections in the format "X.X.X.X" depending on the section and depth of the requirement. For example, the first requirement in the first section, first subsection of the component-level matrix will have the ID "1.1.1".

Description Each requirement and capability needs to have a brief, single sentence description that details the feature. The description should follow the guidelines laid out by NASA¹. They should be unambiguous, testable, verifiable, and consistent in both terminol-

¹<https://www.nasa.gov/seh/appendix-c-how-to-write-a-good-requirement>

ogy and can either be functional or non-functional. Functional requirements directly drive a feature of the product, whereas a non-functional requirement drives an aesthetic or interface choice.

Weight The requirements and capabilities should all have a weight associated with them. These can be based on stakeholder preferences, developer capabilities, technology limitations, etc. The weight is a number greater than 0, but less than 1 and typically only expressed as a single decimal.

Priority Derived from the weight, the priority is the relative importance of a particular requirement or capability to others. It is calculated using Equation G.1. A higher priority means a requirement is more important to implement on the product. For capabilities, the priority is calculated slightly differently. The weight of a capability is compared to all of the other weights in the same category and the one(s) beneath it. For instance, a threshold capability's priority will only factor in the weights of other threshold capabilities. But, a stretch capability's priority will factor in the weights of all capabilities in the design.

$$P = \frac{w_c}{\sum w} \times 100 \quad (\text{G.1})$$

Owner The person who is responsible for a requirement or capability is called the owner. This individual is in charge of developing, implementing, verifying, testing, and validating the product feature. They do not have to be the only person working on that feature, but they are responsible for it. This individual should also be the one tracking and updating the requirement or capability status throughout the traceability matrix.

Verified? The verification of a requirement or capability requires an inspection to ensure that it is present. Component-level requirements will have a test number that should also be tracked in the traceability artifact to describe what test was performed that verified the requirement. Until a requirement or capability is successfully inspected, the answer to the

column header is, "No", but can be changed to "Yes" once verified. A requirement can be subject to a variety of methods to verify it is implemented:

Inspection: The nondestructive examination of a product or system using one or more of the five senses (visual, auditory, olfactory, tactile, taste). It may also include simple physical manipulation and measurements.

Demonstration: The manipulation of the product or system as it is intended to be used to verify that the results are as planned or expected.

Test: The verification of a product or system using a controlled and predefined series of inputs, data, or stimuli to ensure that the product or system will produce a very specific and predefined output as specified by the requirements.

Analysis: The verification of a product or system using models, calculations and testing equipment. Analysis allows someone to make predictive statements about the typical performance of a product or system based on the confirmed test results of a sample set or by combining the outcome of individual tests to conclude something new about the product or system. It is often used to predict the breaking point or failure of a product or system by using nondestructive tests to extrapolate the failure point.

Validated? Validation of a capability or requirement occurs when the feature is fully integrated to the system and tested. It is not enough that it is present, rather it must also contribute to the system as designed without impinging on another feature. This is another "Yes"/"No" column depending on if the feature has been fully integrated to the system.

Status The last major column to consider in the matrix is the status column. This lets the reader know if the capability or requirement has been reviewed or rejected, is undergoing testing, delivered, etc. The value of this column is pulled from Table G.1 based on the requirement's stage in the lifecycle (see Figure G.1).

Table G.1: Description of requirement statuses

UR	Under Review	Requirement has been submitted and is under review
A	Accepted	Requirement has been reviewed and accepted into the artifact
R	Rejected	Requirement has been reviewed and rejected from the artifact
TIP	Testing in Progress	Requirement is currently being tested
TS	Test Successful	Requirement has been successfully tested, but not yet delivered
TF	Test Failure	Requirement failed testing and needs further development
D	Delivered	Requirement has been fully realized and delivered

G.1 Requirement Lifecylce

Requirements are initially created at the start of the project and are expected to be monitored and have their status updated as the project continues. Once a requirement is written, it is up the Lead Systems Engineer, or other team members, to review the requirement and determine if it is suitable for the project. The major considerations are 1) "is the requirement relevant?", 2) "is the requirement unambiguous?", 3) "is the requirement testable?", and 4) "is the requirement complete?" If the requirement does not meet any of these criteria, it can be rejected for re-writing by its owner. Once the requirement is accepted, development for it can begin. It is up to the owner to ensure that the feature detailed by the requirement is properly designed and integrated into the larger system. When the owner verifies the feature and is ready to validate it with the system, they can begin the testing process ad update the status to "TIP" in the traceability matrix. If the test was successful, the requirement status should reflect this; same for a test failure. If the validation fails, the requirement may need to be re-written or re-tested in a different system configuration. Finally, when the feature is fully integrated with the product and released, the status can be set to "delivered" and the feature can be considered fully completed.

Figure G.1: Lifecycle of a requirement

