DEVELOPMENT OF A POWER SYSTEM AND ANALYSIS OF INERTIAL SYSTEM CALIBRATION FOR A SMALL AUTONOMOUS UNDERWATER VEHICLE

by

WILLIAM FORRESTER SEELY

A thesis submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Dr. Daniel Stilwell, Chair
Dr. William Baumann
Dr. Peter King

June 28, 2004
Blacksburg, VA

DEVELOPMENT OF A POWER SYSTEM AND ANALYSIS OF INERTIAL SYSTEM CALIBRATION FOR A SMALL AUTONOMOUS UNDERWATER VEHICLE

William Forrester Seely

ABSTRACT

Compared to large vehicles acting individually, platoons of small, inexpensive autonomous underwater vehicles have the potential to perform some missions that are commonly conducted by larger vehicles faster, more efficiently, and at a reduced operational cost. This thesis describes the power system of a small, inexpensive autonomous underwater vehicle developed by the Autonomous Systems Controls Laboratory at Virginia Tech.

Reduction in vehicle size and cost reduces the accuracy of navigational sensors, leading to the need for autonomous calibration. Several models of navigational sensors are discussed, and the extended Kalman filter is used to form an observer for each, which are simulated and analyzed.

Acknowledgments

I would not be where I am today without the love and support of my future wife, Melissa. I know that it is a daunting and often thankless task to keep me down to earth and focused, but she accomplishes it with patience and compassion, for which she has my deepest thanks.

I would like to gratefully thank my committee chair, Dr. Stilwell, and committee members Dr. Baumann and Dr. King for their guidance and advice throughout my graduate career.

Thanks are also due to the Autonomous Systems and Controls Laboratory for making the long hours spent on research and classes enjoyable, and for providing sanity checks when it was apparent that I had been staring at the same equation for too long.

Finally to the Turbo Lab, my home away from home, thank you for being there to help me relax whenever I needed it.

Contents

List of Figures

List of Tables

List of Symbols

Chapter 1

Introduction

Autonomous underwater vehicles (AUVs) are unmanned, self-propelled submersibles that can perform missions without the need for human supervision. AUVs are used in a wide variety of missions including mine-detection, harbor mapping, underwater structure inspection, and water quality surveying [16, 17]. These missions benefit from automation with increased productivity, reduced human risk, and error reduction.

The disadvantage of using AUVs is that the state of the art vehicles are large and expensive. A typical small AUV, such as REMUS [10], displaces 80 kilograms of water and has a cost on the order of $250,000 for the vehicle alone, not including the specialized equipment required to deploy and retrieve it.

A cooperating platoon of small AUVs could perform the job of a single larger AUV quicker and more efficiently, as in the case of surveying a large body of water. Additionally, applications exist in which cooperating vehicles can perform tasks that a single vehicle fails at, such as finding the global maximum of a two dimensional gradient [3]. Small commercial AUVs such as Nekton's Ranger are appropriately sized, but are not designed with low-cost in mind [6]. To enable research of platoons of small AUVs, there exists the need for a small, inexpensive AUV.

Two problems immediately present themselves when attempting to develop a small and inexpensive AUV. Battery capacity varies with the cube of their volume, so halving the size of a vehicle in both length and hull diameter

will reduce its battery capacity by an eighth. Current advances in battery technology have done well to allow smaller vehicles, and the advent of fuel cell technology may very well enable the use of small AUVs in all but the most demanding missions. Vehicle sensors are the other primary problem; they are already one of the major limitations in developing an autonomous vehicle [17]. Keeping the vehicle inexpensive means using cheaper, lower-accuracy sensors, and reducing the size of the vehicle puts noisy devices like motors and transmitters close to these inaccurate sensors.

High-accuracy inertial measurement units that are used in aerospace applications are too costly to be used in many robotic applications [4], especially a small, inexpensive AUV. In most robots, a less accurate selection of sensors are used to replace the traditional IMU: a magnetic compass for heading, accelerometers for pitch and roll, and a rate gyroscope for angular velocity. This allows for navigation by dead-reckoning, a technique where the current location is derived from a previously known location by integrating vehicle attitude and velocity over time, supplemented by GPS readings to correct any errors. A system using continuous GPS readings is not feasible for an AUV, as GPS signals quickly attenuate underwater. The underwater parallel to GPS would be a network of buoys, each gathering GPS data above the surface and transmitting the data underwater by other means such as acoustic modem. This is not feasible either, given the low-cost restriction. One solution available for a small, inexpensive AUV is to navigate by dead reckoning for some period of time determined by the confidence in the navigational sensors, periodically surfacing for GPS fixes. This leads directly to the desire to get the highest possible accuracy from low-cost navigational sensors.

Manual calibration can accommodate for most significant sensor errors, but errors tend to drift over time, requiring recalibration. Some errors such as those found in low-cost rate gyroscopes drift fast enough that the system would need to be recalibrated several times per hour. One solution is to have the vehicles autonomously calibrate their sensors continuously during operation. This would free up precious man-hours, placing the burden of calibration on the vehicles, which can perform the routines in parallel. Ideally

they would complete initial calibration of the whole platoon in the time it would take to manually calibrate a single vehicle.

The goal of this thesis was to design and fabricate a power system for a small AUV that has been developed by the Autonomous Systems and Controls Laboratory (ASCL), and to research and analyze navigational system calibration routines for a platoon of these AUVs. Chapter 2 details the design and development of the Virginia Tech miniature autonomous underwater vehicle, pictured in Figure 1.1, in particular the battery and power systems which were the primary responsibility of the author. The navigational sensors used in the current vehicle is discussed in Chapter 3, along with the possible error sources, while models of an IMU are detailed in Chapter 4. Chapter 5 describes the Kalman filter and extended Kalman filter, and creates observers for various IMU models, which are simulated in Chapter 6.

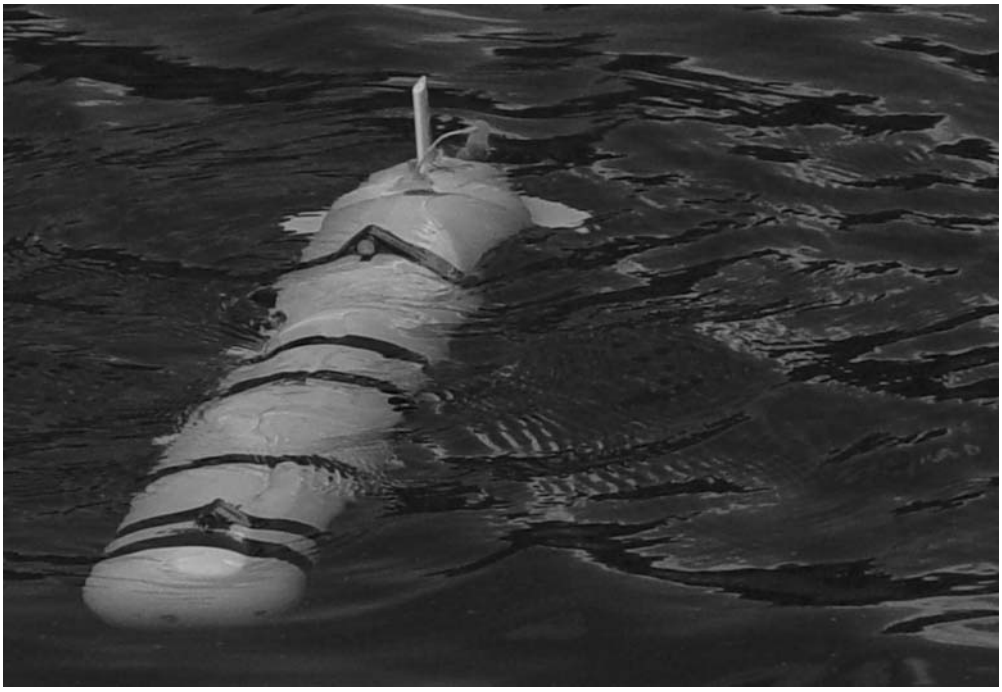

Figure 1.1: Virginia Tech miniature AUV being tested at Bluestone quarry

The algorithms developed in this thesis were successful at calibrating for a navigational system with an additive bias error. The system may encounter problems when considering scale factor and bias errors however, as proven in

Chapter 6 for a single degree of freedom case. Still, the algorithms developed herein may yet be experimentally applicable if a model is found which is observable for additional error sources, or the IMU is aided by additional sensors that allow the model used in this thesis to become observable. The development of a power system for a small AUV was a complete success, the power distribution board is currently being mass produced for an additional five AUVs, which will be used to conduct research of platoons of small, inexpensive AUVs.

Chapter 2

The Virginia Tech Miniature Autonomous Underwater Vehicle

An ongoing project of the Autonomous Systems and Controls Laboratory (ASCL) is the development of a platoon of small, inexpensive, field-deployable AUVs for research in scientific and military applications. Toward this goal, the ASCL has created a prototype AUV in its first year of operation, and with lessons learned from it, built a fully field-deployable vehicle the following year. Work is currently in progress for the creation of five more vehicles to allow experimental research on platoons of cooperating vehicles.

2.1 History

2.1.1 USNA-1

C.E. Wicks built the USNA-1, pictured in Figure 2.1, at the United States Naval Academy in 2001. The vehicle was constructed as a proof-of-concept for low-cost AUVs [13].

USNA-1 used small commercial off-the-shelf (COTS) sensors that trade accuracy for a smaller footprint, and did not incorporate mission specific sensors. These deviations from conventional autonomous vehicle design allowed

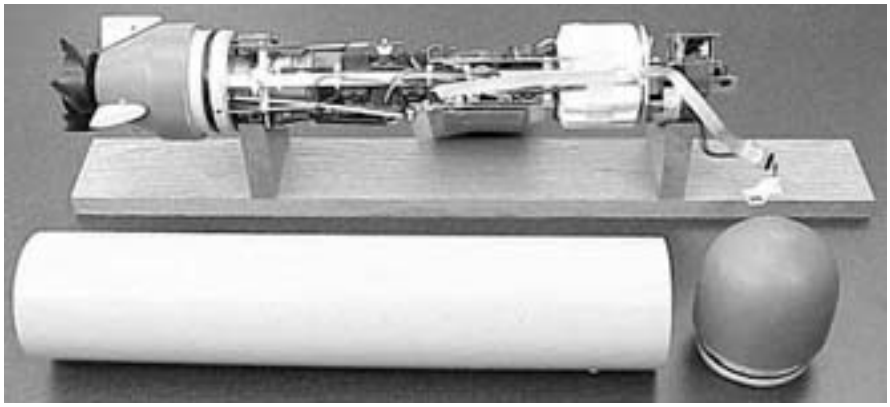

Figure 2.1: USNA-1 miniature autonomous underwater vehicle, courtesy of United States Naval Academy

for a vehicle which could be adapted to many different missions and was both small and inexpensive.

2.1.2 Prototype

The prototype vehicle, pictured in Figure 2.2, was built by the Autonomous Systems Control Lab at Virginia Tech. The vehicle iterated on the USNA-1 design in an attempt to focus on practical experimental use. The prototype saw an increase in navigation performance, communication, motion control, and modularity over USNA-1 [9]. The general specifications of the vehicle can be seen in table 2.1.

Figure 2.2: Prototype miniature autonomous underwater vehicle

Parameter	*Specification*
Length	91 cm
Hull Diameter	8.9 cm
Mass	∼5.3 kg
Stability	passively stable in pitch and roll
Motion Control, Propulsion	brushless DC motor with encoder feedback and custom driver to control shaft velocity
Motion Control, Steering	three independently actuated fins driven by R/C servo motors
Computer Control System	Rabbit 3000 microprocessor
Power	21.6 volt, 80 watt-hour NiMh battery stack
Communications	57Kbps bidirectional, 60 mile range in air with ideal antenna
Data storage	8MB Flash ROM
Range	17 km (anticipated)
Cruising speed	2-3 knots (1-1.5 m/s)
Max depth	200 meters (anticipated)

Table 2.1: Prototype general specifications

With construction of the prototype, several issues became apparent that would need to be addressed for a field-deployable vehicle. Foremost among the problems seen was modularity: the vehicle would have to be as modular as possible to allow for single components to be replaced or upgraded without sending the vehicle into dry-dock for a significant length of time.

2.2 Virginia Tech miniature autonomous underwater vehicle

The Virginia Tech miniature autonomous underwater vehicle (VT miniature AUV) was designed during the summer of 2003. A summary of the vehicle's specifications can be seen in Table 2.2. Figure 2.3 displays the vehicle before the shell was painted yellow, allowing a clear view of the internal electronics.

Parameter	*Specification*
Length	78 cm
Hull Diameter	9.5 cm
Mass	5.195 kg with mission sensor
Stability	passively stable in pitch and roll
Computer Control System	White Dwarf Linux, 133 MHz AMD Elan SC520 DIMM-PC
Power	22.2 volt, 88.8 watt-hour LiPoly battery stack
Communications	115Kbps bidirectional, 60 mile range in air with ideal antenna
Data storage	32M flash ROM, external expansion possible
Maximum speed	3 knots
Max depth	200 meters (anticipated)

Table 2.2: VT miniature AUV general specifications

2.2.1 Structural system

The prototype was designed with the goal of being slightly positively buoyant, but due to the inability to simulate the weight of every item in the

Figure 2.3: VT miniature AUV with clear shell

vehicle, it came very close to being too heavy. To rectify the issue on the VT miniature AUV, the outer diameter was widened so the vehicle would displace more water. The VT miniature AUV was designed to be very positively buoyant with extra room available for lead weights at the bottom of the vehicle to bring it back to slightly positive. This design had the added benefit of lowering the vehicle's center of gravity, adding to the passive roll stability.

The new new shell was constructed of acrylic instead of PVC, which allowed for a thinner shell, making the inner diameter wider as well. This allowed for a more tightly packed subsystem arrangement where twin rails at the top and bottom were used to secure three modules stacked vertically. This design change along with the selection of a new power system with greater energy density allowed for a 14 percent reduction in the vehicle's length. The completed vehicle is 78 centimeters long, 9.5 centimeters outer diameter, and weighs 5.195 kilograms with a YSI mission sensor attached, slightly less than an equal volume of water.

2.2.2 Propulsion system

The VT miniature AUV uses a brushless DC motor with encoder feedback to drive the propeller. A custom motor controller and driver board regulates propeller velocity. The controller board also ensures that the current draw from the motor board does not exceed two amperes. The vehicle's current propeller is a small model airplane propeller with a low pitch. The propeller

delivers a maximum thrust of 1.56 pounds at full draw in fresh water, which corresponds to three feet per second maximum speed. Research is currently underway for optimizing the motor revolutions per minute and propeller design to allow for greater vehicle endurance.

2.2.3 Instrumentation and navigation system

Instrumentation and navigation sensors are a magnetic compass, depth transducer, accelerometers, and rate-gyroscope [5]. The depth transducer measures pressure through an access hole in the bottom of the vehicle's nose. The accelerometers and rate gyroscope are part of the analog sensors board, a custom-engineered printed circuit board which also contains analog to digital converters connected to all sensor components. The VT miniature AUV uses smaller and higher precision accelerometers and gyros over the prototype. The full details of the accelerometers, compass, and rate-gyroscope are in Chapter 3.

2.2.4 Vehicle control system

To avoid the complications met with the previous vehicle's CPU, The VT miniature AUV uses a 133 megahertz Advanced Microdevices Elan SC520 DIMM-PC. The CPU is 5x86 PC/AT compatible, its operating system is White Dwarf Linux. The code base is POSIX compliant, which will facilitate portability if the CPU or operating system ever needs to be changed.

2.3 Power system

The author's primary responsibility with respect to the vehicle design and operation was the power distribution system.

Table 2.3 shows the power budget for the VT miniature AUV. In the idle state, the vehicle has an expected load of 5.5 watts; this number is referred to as the vehicle's hotel load. While surfaced and communicating, the load is 5.8 watts. The underwater operation load is not fully specified due to variances in the density of water and the effects on the DC motor load, but

with the motor running at full power, the underwater operation load will be 56 watts. Data is still being collected that will show the optimal motor speed for vehicle endurance.

Component	#	*Voltage* (V)	*Current* (A)	*Power* (W)	*State*		
					idle	*comm*	*oper*
GPS (idle)	1	5	0.18	0.9	x		x
GPS (active)	1	5	0.0001	0.0005		x	
Compass	1	5	0.016	0.08	x	x	x
Analog sensors	1	5	0.07	0.35	x	x	x
DIMM-PC	1	5	0.7	3.5	x	x	x
Serial expansion	1	5	0.125	0.625	x	x	x
DC motor	1	22.2	2	44.4			x
Servos (idle)	3	5	0.003	0.045	x	x	
Servos (active)	3	5	0.3	4.5			x
Sonar	2	12	0.07	1.68			x
RF modem	1	12	0.1	1.2		x	
Total Power (W)					5.5	5.8	56

Table 2.3: VT miniature AUV power budget

Assume that in a given mission the vehicle will spend 95% of its time in underwater operation at full speed, and the remainder of its time surfaced and either communicating with a base station or collecting GPS data. Given this distribution, the average load is expected to be 53.5 watts.

2.3.1 Batteries

The VT miniature AUV is powered by twelve Kokam KOK2000HCT lithium-polymer (LiPoly) batteries. A single cell, shown in Figure 2.4, weighs 46 grams, and is 96mm long, 64.5mm wide, and 4mm thick. Each cell has a capacity of 2 ampere-hours, a maximum voltage of 4.2 volts, and a nominal voltage of 3.7 volts.

The battery stack, shown in Figure 2.5, consists of two sub-stacks connected in parallel, each of which consists of six cells connected in series. The complete battery stack has a capacity of 4 ampere-hours, and a nominal voltage of 22.2 volts, totaling 88.8 watt-hours. Given the expected power requirements, this gives just over 100 minutes of operation between recharges.

Figure 2.4: Kokam lithium-polymer cell

The average speed of the vehicle during a mission is expected to be 6 kilometers per hour, giving the vehicle a range of 10 kilometers at maximum speed. The primary factor governing motor load is expected to be hydrodynamic drag, which varies with the square of speed, so once an optimal speed is found the vehicle's endurance and range are expected to show considerable improvement.

Figure 2.5: Constructed battery stack

LiPoly batteries have a much higher energy density than other rechargeable battery technologies currently available, such as nickel metal hydride,

lead acid, and lithium ion. The full stack of 12 batteries weighs 552 grams. For energy density comparisons with other chemistries, this is 1.22 pounds. With a capacity of 88.8 watt-hours, the energy density of the battery stack is 73 watt-hours per pound. A comparison of energy densities can be seen in Table 2.4 [17]. LiPoly batteries also fit our size constraints more easily, given that LiPoly manufacturers shape the batteries into thin cells with rectangular footprints.

Chemistry	*Energy density (Wh/lb)*
Lead-Acid	10-18
Nickel-Cadmium	12-20
Nickel-Iron	20-25
Silver-Cadmium	18-45
Silver-Zinc	40-48
Lithium-Polymer	73

Table 2.4: Energy densities of various battery chemistries

Charge/Discharge characteristics

The battery stack is charged by an Orbit Microlader V6.3. The charger has the ability to automatically detect the number of batteries in series and adjust the voltage accordingly, and has an RS-232 port which allows a PC to collect data on the current state of the batteries being charged.

LiPoly batteries require special charging hardware to avoid damaging the batteries. At the start of a charging session the current is held at battery stack's full capacity (4 amperes in this case) while the voltage is gradually raised until it reaches the maximum charge, after which the current is gradually reduced to zero. The charge cycle as specified in the manufacturer data sheet for a generic Kokam LiPoly battery can be seen in Figure 2.6 [8].

The Orbit Microlader can be put into discharge mode, allowing it to discharge a set of batteries at a desired current. With the use of Orbit PC software, a plot of the battery stack's discharge characteristics can be obtained. Figure 2.7 shows the discharge characteristic of an example Kokam LiPoly battery under various currents, as depicted in the manufacturer data sheet.

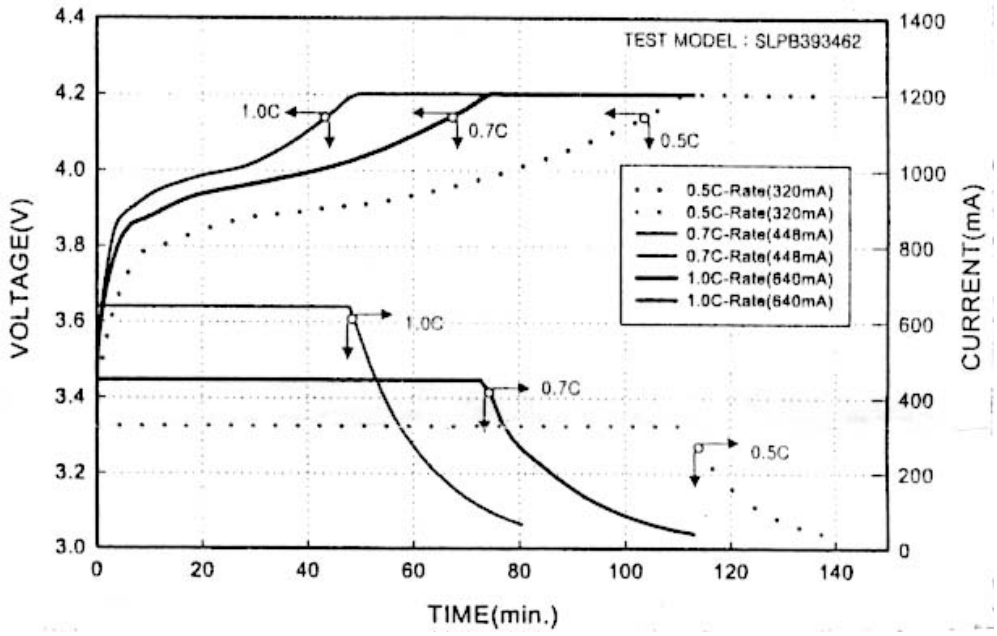

Figure 2.6: Charge characteristic of Kokam LiPoly batteries

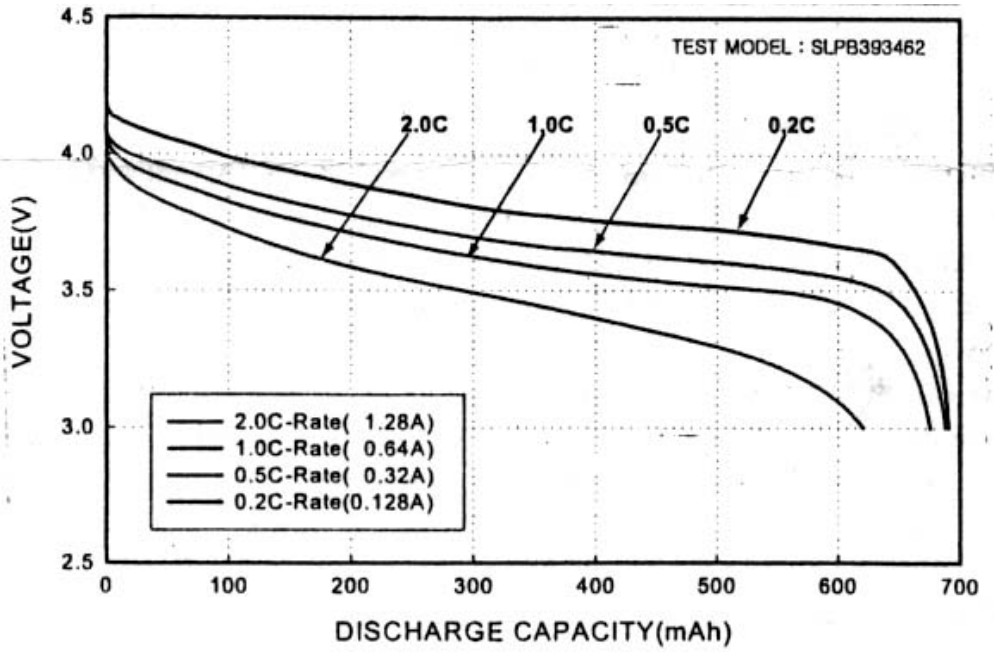

Figure 2.7: Discharge characteristic of Kokam LiPoly batteries

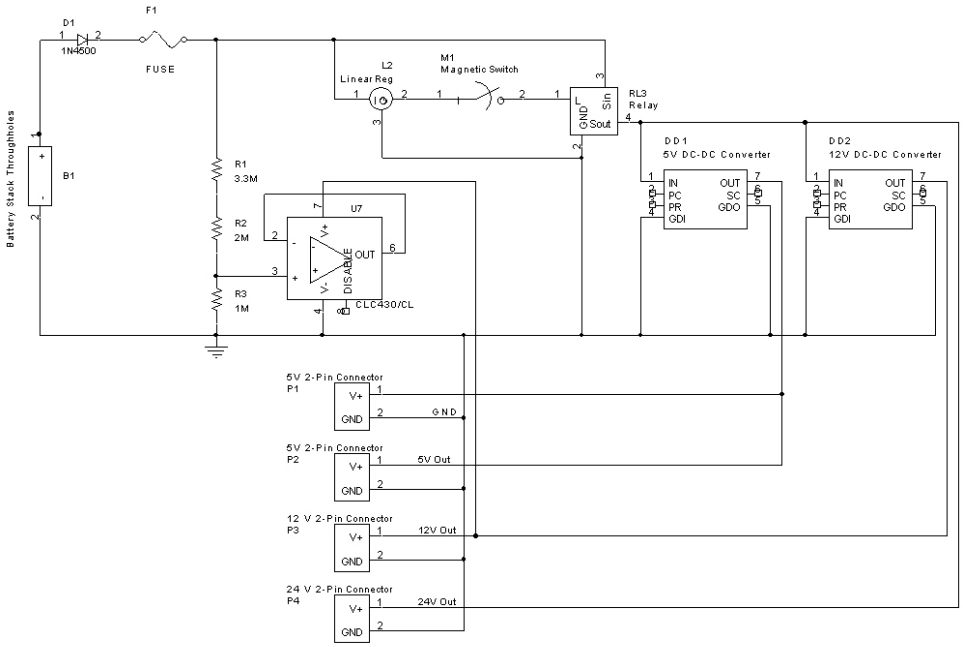

Figure 2.8: Circuit diagram of power distribution board

The discharge characteristic shows one of the disadvantages of LiPoly batteries, the voltage drop is fairly linear until it reaches 18 volts, after which there is a rapid loss of voltage in a short time. Allowing the stack to fall below 15 volts can cause permanent damage to the batteries, significantly reducing their capacity. If the batteries continue to discharge below 15 volts for extended periods of time, they have the potential to ignite.

2.3.2 Power distribution board

The VT miniature AUV's DC motor operates off the full battery stack voltage available, but all other components require regulated twelve volt or five volt lines. To accommodate this, the Power Distribution Board (PDB) sits above the battery stack to split the stack voltage into the three necessary voltages and offer several safety measures. A circuit diagram of the PDB can be seen in Figure 2.8.

The original PDB for the vehicle used switching regulators, but even

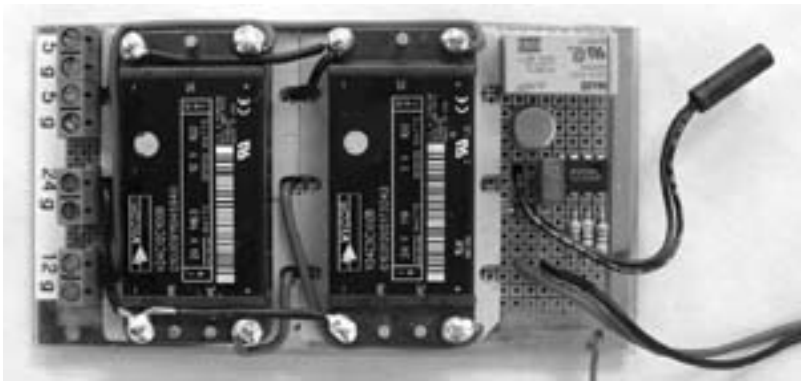

Figure 2.9: Power distribution board prototype

with bypass capacitors, the five volt line was too noisy. Whenever the motor ramped up under load, there would be a small change in the five volt line, causing the DIMM-PC to reset. At first the problem was rectified by using a single half-size DC-DC converter from Vicor, but this caused the PDB to take up precious real-estate in the nose area. Eventually Vicor began production of a quarter-size DC-DC converter, and the PDB was redesigned using them in place of the switching regulators. The final component layout of the PDB can be seen in Figure 2.9. The PDB and battery stack's location on the vehicle can be seen in Figure 2.10.

The PDB contains safety measures to protect the battery stack, the vehicle components, and the user.

The power line first passes through a diode and a five ampere fuse to protect against shorts; this current was selected because it is far in excess of the highest expected current requirements, even when considering future vehicle upgrades to include sonar transducers. The power line then passes through a relay and a five volt linear regulator. The output of the linear regulator is connected to a magnetic reed switch which controls the relay. When the reed is magnetized, the relay is closed and the power line continues on to the two DC-DC regulators and the stack voltage terminal. The magnetic switch allows the vehicle power to by cycled without having to open the shell.

Typically to prevent over-discharge of batteries, the relay and linear regulator would be selected such that the minimum voltage to keep the relay

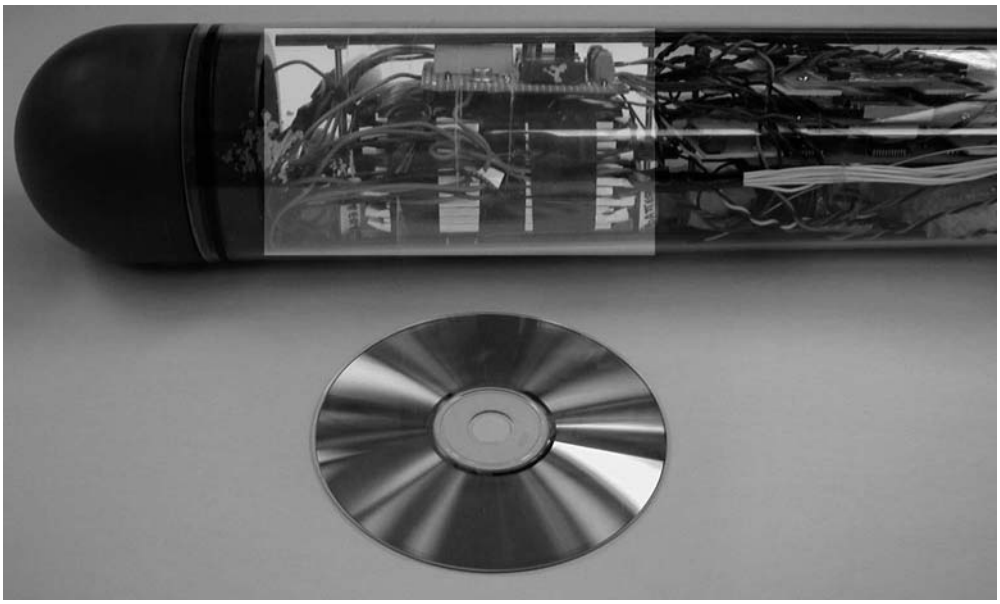

Figure 2.10: Power board and batteries highlighted in VT miniature AUV internals

active corresponds to the cut-off voltage of the battery stack. For the VT miniature AUV the decision was made that in extreme circumstances the vehicle may need to operate below the suggested minimum for short periods of time in the interest of saving the vehicle as a whole at the expense of lowering the capacity of the battery stack. Thus the battery stack is monitored by a voltage divider and voltage follower that outputs five volts at full capacity, and the job of deciding when to shut down the system is left to the software.

Chapter 3

Inertial Measurement Unit

The inertial measurement unit (IMU) is a common component of a navigation system. On the VT miniature AUV, a traditional IMU is replaced by a dual-axis accelerometer, a magnetic compass, and a three-axis rate gyroscope. With these sensors the IMU is able to measure body frame angular velocity and local-level frame angular displacement.

3.1 Dual-axis accelerometer

An Analog Devices ADXL311 dual-axis accelerometer is mounted on the vehicle such that when the vehicle is at rest the accelerometer axes are parallel to the earth's surface. Given the relatively slow dynamics of the vehicle, gravity is the dominant acceleration felt, and so the accelerometer can be used to determine pitch (θ) and roll (ϕ) from the accelerations felt in the x-axis (A_x) and y-axis (A_y) by

$$\theta = -\arcsin\left(\frac{A_x}{g}\right)$$
$$\phi = \arcsin\left(\frac{A_y}{g\cos(\theta)}\right)$$

As stated in the manufacturer data sheet[1], the ADXL311 has a sensitivity of nearly 17.5 milli-gees per degree of tilt for low tilt angles, lowering to 12.2 milli-gees per degree at the 45 degree mark and declining further. At

narrow output bandwidths the accelerometer can resolve a 2 milli-gee change, corresponding to a 0.1° resolution.

3.2 Magnetic compass

The True North Technologies OEM magnetic compass contains a three-axis magnetometer and a dual-axis electrolytic inclinometer. It is capable of generating raw or conditioned heading readings up to 20 times per second and sending them via RS-232 serial interface. The compass includes internal compensation for hard and soft iron, and the electrolytic tilt sensor allows the compass to operate despite misalignment or vehicle rotations.

According to the manufacturer data sheet [15], the compass is accurate to half a degree within ±35 degrees of roll and ±60 degrees of pitch. Accuracy deteriorates past these points, and the compass will continue to report heading regardless of roll and up to ±80 degrees of pitch. Response time for heading measurements is 75 milliseconds.

The compass is also capable of measuring pitch and roll using the dual-axis inclinometer, but the resolution is inferior to the accelerometers (0.2 degrees), the signal takes up to half a second to settle, and the pitch and roll cannot be determined past ±40 degrees of roll, as compared to the ±90 degree range on the accelerometers. For these reasons the compass inclinometer does not replace the accelerometer.

3.3 Rate gyroscope

The Analog Devices ADXRS150 rate gyroscope operates by sensing the Coriolis force due to angular rate acting upon two orthogonally placed vibrating structures. The manufacturer data sheet [2] states the range of operation for the gyroscope as $\pm 150^\circ/s$, and the output as an analog signal from zero to five volts. A limiting factor in the gyroscope is the substantial bias drift the sensor experiences.

3.4 IMU Error sources

The IMU components are susceptible to many of the same errors seen in other measurement systems: temperature changes, analog to digital converter resolution, and electromagnetic interference. These errors are of concern during experimental application, but are difficult to accurately account for in a simulation.

The magnetic compass, in particular, is prone to additional errors in the form of misalignment and hard and soft iron fields. The calibration routines necessary to accommodate for these errors were researched by J.J. Mach [9], and the OEM Compass currently in use contains calibration routines to correct them.

The main source of error for the rate gyroscope are in the form of scale and bias error [14]. The true angular rate, ω, relates to the angular rate measured by the rate gyroscope, ω_m, by

$$\omega_m = (1 + a)\omega + b \tag{3.1}$$

where a is the scale factor error, and b is the bias error. It will be more convenient for calibration to express (3.1) as

$$\omega = \bar{a}\omega_m + \bar{b} \tag{3.2}$$

where

$$\bar{a} = \frac{1}{1 + a}, \quad \bar{b} = \frac{-b}{1 + a}$$

Chapter 4

Kinematic Models

In order to create a low cost miniature AUV, every component must be inexpensive and compact. As a result the navigational sensors are poor quality.

The errors discussed in Section 3.4 can be accommodated through manual calibration, but problems occur when considering the tendency for the sensor errors to drift rapidly, requiring recalibration. This is the motivation for a vehicle to autonomously calibrate itself.

One approach toward auto-calibration is to make a system model and form an observer to determine how the true system deviates from the ideal. Such a system could continuously calibrate the sensors as the vehicle operated, accommodating for sensor drift. The constant bias error in the rate gyroscope is a prominent and common error source, making it a prime candidate for testing observers.

4.1 Notation

Before discussing the details of a rotational system, it is important to clarify the notation. Because the IMU is part of an AUV, the notations standard to the vehicle dynamics field are appropriate, as detailed below.

Angular displacement is described by roll(ϕ), pitch(θ), and yaw(ψ) with respect to a local north-east-down coordinate frame. This set of angles is referred to as Euler angles, the vector of which is denoted $\eta = [\phi, \theta, \psi]^T$.

This is in contrast to the robotics definition of Euler angles: a rotation about the local z-axis, followed by a rotation about the local y-axis, followed by a rotation about the (new) local z-axis.

Angular velocity with respect to the body-fixed coordinate frame is denoted $\omega = [\omega_x, \omega_y, \omega_z]^T$ [11].

The rotation matrix transforming vectors from the body-fixed frame to the local-level frame is $R(\eta)$, or simply R, defined

$$R(\phi, \theta, \psi) = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} \tag{4.1}$$

$$\begin{aligned}
R_{1,1} &= \cos(\theta)\cos(\psi) \\
R_{1,2} &= \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) \\
R_{1,3} &= \cos(\phi)\sin(\theta)\cos(\psi) - \sin(\phi)\sin(\psi) \\
R_{2,1} &= \cos(\theta)\sin(\psi) \\
R_{2,2} &= \sin(\phi)\sin(\theta)\sin(\psi) - \cos(\phi)\cos(\psi) \\
R_{2,3} &= \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\
R_{3,1} &= -\sin(\theta) \\
R_{3,2} &= \sin(\phi)\cos(\theta) \\
R_{3,3} &= \cos(\phi)\cos(\theta)
\end{aligned}$$

The rotation matrix is an element of the special orthogonal group of order three, or SO(3), that satisfies

$$RR^T = I$$

and

$$\det(R) = 1$$

4.2 Euler angle model

The attitude of the vehicle satisfies

$$\dot{\eta}(t) = Q(\eta(t))\omega(t) \tag{4.2}$$

where

$$Q(\eta) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix}$$

and ω is the body angular rate. Note that the above system has singularities at $\theta = \pm 90°$. With the addition of bias error, (4.2) becomes

$$\dot{\eta}(t) = Q(\eta(t))(\omega(t) + b) \tag{4.3}$$

where $b \in \mathbb{R}^3$.

4.3 Rotation matrix model

To avoid these singularities, the vehicle can instead be modeled by direct integration of (4.1). This will be referred to as the rotation matrix model, and can be written

$$\dot{R}(t) = R(t)S(\omega(t)) \tag{4.4}$$

where

$$S(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$S(\omega)$ represents the cross product. For any $a, b \in \mathbb{R}^3$

$$S(a)b = a \times b$$

Internally the system does not contain the Euler angle singularities, but we are ultimately interested in the Euler angles, which are retrieved from (4.4) by

$$\begin{aligned} \phi &= \arctan\left(\frac{R_{3,2}}{R_{3,3}}\right) \\ \theta &= \arcsin\left(-R_{3,1}\right) \\ \psi &= \arctan\left(\frac{R_{2,1}}{R_{1,1}}\right) \end{aligned} \tag{4.5}$$

Note that there are singularities in the roll and yaw at 180 degrees, and in the pitch at 90 degrees. These constraints are acceptable as they are within the boundary of the vehicle's expected behavior.

With the bias error added, (4.4) is written

$$\dot{R}(t) = R(t)S(\omega(t) + b) \tag{4.6}$$

4.4 Quaternion model

Quaternion-based models have distinct advantages over models based on Euler angle and rotation matrix representations. Quaternions are more compact than rotation matrices, and thus the calculations are inherently faster, and they do not contain the singularities present in Euler angle models.

Unit quaternions

Unit quaternions are a four element vector representation of SO(3), containing one real dimension and three imaginary dimensions. Quaternions are written

$$q = \begin{bmatrix} v_o \\ v \end{bmatrix}$$

where $v_o \in \mathbb{R}$ and $v \in \mathbb{R}^3$.

The imaginary dimensions are denoted i, j, k and satisfy

$$
\begin{aligned}
i^2 = j^2 = k^2 = ijk = -1 \\
ij = -ji = k \\
jk = -kj = i \\
ki = -ik = j
\end{aligned}
$$

A unit quaternion is distinguished from other quaternions in that it satisfies $\|q\| = \sqrt{v_o^2 + v^T v} = 1$ where v_o is the real component, and $v = [i, j, k]^T$. Like the matrices they represent, quaternions are associative but not commutative. Given two unit quaternions,

$$
q_x = \begin{bmatrix} v_{xo} \\ v_x \end{bmatrix}, \quad \text{and} \quad q_y = \begin{bmatrix} v_{yo} \\ v_y \end{bmatrix},
$$

the quaternion product is defined as

$$
q_1 \cdot q_2 = \begin{bmatrix} v_{xo} & -v_x^T \\ v_x & S(v_x) + I v_{xo} \end{bmatrix} \begin{bmatrix} v_{yo} \\ v_y \end{bmatrix}
$$

where S is the skew-symmetric matrix defined for (4.4). The transformations between $q(t)$ and $R(t)$ where

$$
q = \begin{bmatrix} v_o \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad \textit{and} \quad R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix}
$$

are defined as

$$
\begin{aligned}
R_{1,1} &= v_o^2 + v_1^2 - v_2^2 - v_3^2 \\
R_{1,2} &= 2(v_1 v_2 - v_o v_3) \\
R_{1,3} &= 2(v_1 v_3 + v_o v_2) \\
R_{2,1} &= 2(v_1 v_2 + v_o v_3) \\
R_{2,2} &= v_o^2 - v_1^2 + v_2^2 - v_3^2 \\
R_{2,3} &= 2(v_2 v_3 - v_o v_1) \\
R_{3,1} &= 2(v_1 v_3 - v_o v_2) \\
R_{3,2} &= 2(v_2 v_3 + v_o v_1) \\
R_{3,3} &= v_o^2 - v_1^2 - v_2^2 + v_3^2
\end{aligned}
\tag{4.7}
$$

and

$$
\begin{aligned}
v_o &= \frac{1}{2}\sqrt{1 + R_{1,1} + R_{2,2} + R_{3,3}} \\
v_1 &= \frac{1}{4v_o}(R_{3,2} - R_{2,3}) \\
v_2 &= \frac{1}{4v_o}(R_{1,3} - R_{3,1}) \\
v_3 &= \frac{1}{4v_o}(R_{2,1} - R_{1,2})
\end{aligned}
$$

The unit quaternion representation of (4.4) is

$$
\dot{q}(t) = \frac{1}{2}T(q(t))w(t) \tag{4.8}
$$

where

$$
T(q) = \begin{bmatrix} -v^T \\ S(v) + Iv_o \end{bmatrix}
$$

Chapter 5

The Kalman Filter

Several kinematic models have been developed in Chapter 4, and it remains to develop an observer for the system. Given the uncertainty inherent to the navigational sensors, a natural choice for an observer is the Kalman filter.

5.1 History

In 1960, R.E. Kalman published his paper on a recursive solution to the linear filtering problem [7]. Today the solution, which has come to be known as the Kalman filter, is widely used in autonomous navigation research and many other fields.

The Kalman filter minimizes mean square error in the state estimation, and most importantly, it allows for some degree of model uncertainty within certain constraints.

5.2 Continuous-time Kalman filter

Suppose a plant is given as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Lv(t), \quad x(t_o) = x_o, \quad t \geq t_o \\ y(t) &= Cx(t) + w(t) \end{aligned} \tag{5.1}$$

The random variables v and w are Gaussian, uncorrelated zero mean with covariance

$$Q = E\{vv^T\}$$
$$R = E\{ww^T\}$$

The covariance of the estimation error satisfies the Riccati equation

$$\dot{P} = AP + PA^T + LQL^T - PC^T R^{-1} CP$$
$$P(t_o) = P_o$$

The Kalman gain is determined by

$$K = PC^T R^{-1}$$

and the observer is

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y}) \tag{5.2}$$
$$\hat{y} = C\hat{x}$$

Typically the Kalman filter is discretized at this point and executed in two repeating steps. In step one the Kalman gain is updated using the Riccati equation, and in step two the new Kalman gain is used to update the error estimate. For simulation the discretization step is not necessary, and would only add quantization to the list of possible error sources when attempting to compare the Kalman filter with other methods, so the filter is left in its continuous form with the understanding that the performance is the maximum possible, and an experimental implementation may not be able to reach the full potential.

5.3 Extended Kalman filter

Since (5.2) assumes a linear plant, it can not be applied directly to nonlinear plants such as

$$\begin{aligned} \dot{x} &= f(x, u, v) \\ y &= h(x, w) \end{aligned} \tag{5.3}$$

The extended Kalman filter (EKF) overcomes this problem by adding a third step to the discrete Kalman filter where the nonlinear system is linearized around the current state. The primary drawback to this method is that the properties of the Kalman filter such as stability of the estimation error are only applied locally. The EKF also contains additional computational complexity, requiring additional CPU bandwidth. This drawback is not a problem for the simulation even when considering that it will be run continuously, as our plant is not operating in real time. However, this is something to keep in mind when considering experimental applications.

The EKF observer, linearized from (5.2) is

$$\begin{aligned} \dot{\hat{x}} &= \widetilde{A}\hat{x} + \widetilde{B}u + K(y - \hat{y}) \\ \hat{y} &= \widetilde{C}\hat{x} \end{aligned} \tag{5.4}$$

where

$$\begin{aligned} \widetilde{A} &= \frac{\partial f}{\partial x} \\ \widetilde{B} &= \frac{\partial f}{\partial u} \\ \widetilde{C} &= \frac{\partial h}{\partial x} \end{aligned}$$

are evaluated at the operating point $\tilde{x}$, u.

Chapter 6

Simulations and Results

The observer discussed in Chapter 5 can now be applied to the navigational sensors. The rotation matrix model does not lend itself to use in an observer given the fact that the Euler angles are not readily available as a vector in the system. However, it is useful for the vehicle model, as it is not susceptible to the singularities found in the Euler angle model.

The Euler angle and quaternion models are used to form observers, and the estimation from them is then compared to the vehicle rotation matrix model.

6.1 Euler angle extended Kalman filter

A common method for estimating unknown constants in a dynamic system is to augment the state vector to include the unknown constant. Recall the Euler angle model

$$\dot{\eta}(t) = Q(\eta(t))\omega(t) \tag{6.1}$$

Adding the bias error and process noise, (6.1) becomes

$$\dot{\eta}(t) = Q(\eta(t))(\omega(t) + b(t) + v(t))$$

Augmenting the state vector to include the bias error yields

$$\begin{bmatrix} \dot{\eta}(t) \\ \dot{b}(t) \end{bmatrix} = \begin{bmatrix} Q(\eta(t))(\omega(t) + b(t) + v(t)) \\ 0 \end{bmatrix} \tag{6.2}$$

In order to apply the EKF, the plant is linearized about an operating point that now includes the bias error as part of the state vector:

$$\begin{bmatrix} \dot{\eta}(t) \\ \dot{b}(t) \end{bmatrix} \approx \widetilde{A} \begin{bmatrix} \eta(t) \\ b(t) \end{bmatrix} + \widetilde{B}\omega(t) + \widetilde{L}v(t) \tag{6.3}$$
$$y(t) = \eta(t)$$

$\widetilde{A}$, $\widetilde{B}$, and $\widetilde{L}$ are matrices of Jacobians defined as

$$\widetilde{A} = \begin{bmatrix} \frac{\partial Q(\eta(t))b(t)}{\partial \eta(t)} & \frac{\partial Q(\eta(t))b(t)}{\partial b(t)} \\ 0 & 0 \end{bmatrix}$$
$$\widetilde{B} = \begin{bmatrix} \frac{\partial Q(\eta(t))\omega(t)}{\partial \omega(t)} \\ 0 \end{bmatrix}$$
$$\widetilde{L} = \begin{bmatrix} \frac{\partial Q(\eta(t))v(t)}{\partial v(t)} \\ 0 \end{bmatrix}$$

Computing the derivatives yields

$$\frac{\partial Q(\eta(t))b(t)}{\partial b(t)} = \frac{\partial Q(\eta(t))v(t)}{\partial v(t)} = \frac{\partial Q(\eta(t))\omega(t)}{\partial \omega(t)} = Q(\eta(t))$$
$$\frac{\partial Q(\eta(t))b(t)}{\partial \eta(t)} = \begin{bmatrix} \widetilde{A}_{1,1} & \widetilde{A}_{1,2} & 0 \\ \widetilde{A}_{2,1} & 0 & 0 \\ \widetilde{A}_{3,1} & \widetilde{A}_{3,2} & 0 \end{bmatrix}$$
$$\widetilde{A}_{1,1} = b_y \cos(\phi)\tan(\theta) - b_z \sin(\phi)\tan(\theta)$$
$$\widetilde{A}_{1,2} = b_z \cos(\phi)\sec^2(\theta) + b_y \sin(\phi)\sec^2(\theta)$$
$$\widetilde{A}_{2,1} = b_z \cos(\phi) - b_y \sin(\phi)$$
$$\widetilde{A}_{3,1} = b_y \cos(\phi)\sec(\theta) - b_z \sin(\phi)\sec(\theta)$$
$$\widetilde{A}_{3,2} = b_z \cos(\phi)\sec(\theta)\tan(\theta) + b_y \sin(\phi)\sec(\theta)\tan(\theta)$$

Applying the generalized EKF observer from (5.4) to the linearized Euler angle model in (6.3) yields

$$\begin{aligned}
\begin{bmatrix} \dot{\hat{\eta}}(t) \\ \dot{\hat{b}}(t) \end{bmatrix} &= \widetilde{A} \begin{bmatrix} \hat{\eta}(t) \\ \hat{b}(t) \end{bmatrix} + \widetilde{B}\omega(t) + K(y - \hat{y}) \qquad (6.4) \\
\hat{y} &= \hat{\eta}(t) \\
K &= PC^T R^{-1} \\
\dot{P} &= \widetilde{A}P + P\widetilde{A}^T + \widetilde{L}Q\widetilde{L}^T - PC^T R^{-1} CP
\end{aligned}$$

where $\widetilde{A}$ and $\widetilde{B}$ are linearized around $\hat{\eta}$, $\hat{b}$, and ω.

For convenience in simulation, the covariance matrices are chosen as

$$P_o = Q = R = I$$

An outline of the Simulink implementation of this system can be seen in Figure 6.1. The detailed Simulink model can be seen in Appendix A.2.

The vehicle model is driven by a vector input of sine waves

$$u(t) = \begin{bmatrix} a_\phi \sin(f_\phi \cdot t) \\ a_\theta \sin(f_\theta \cdot t) \\ a_\psi \sin(f_\psi \cdot t) \end{bmatrix}$$

where $[a_\phi, f_\phi, a_\theta, f_\theta, a_\psi, f_\psi]^T$ is a simulation parameter, along with the internal bias error $b = [b_x, b_y, b_z]^T$.

The observer contains a vector parameter corresponding to the initial conditions

$$\begin{bmatrix} \hat{\eta}_o \\ \hat{b}_o \end{bmatrix} = \begin{bmatrix} \hat{\phi}_i \\ \hat{\theta}_i \\ \hat{\psi}_i \\ \hat{b}_{xi} \\ \hat{b}_{yi} \\ \hat{b}_{zi} \end{bmatrix}$$

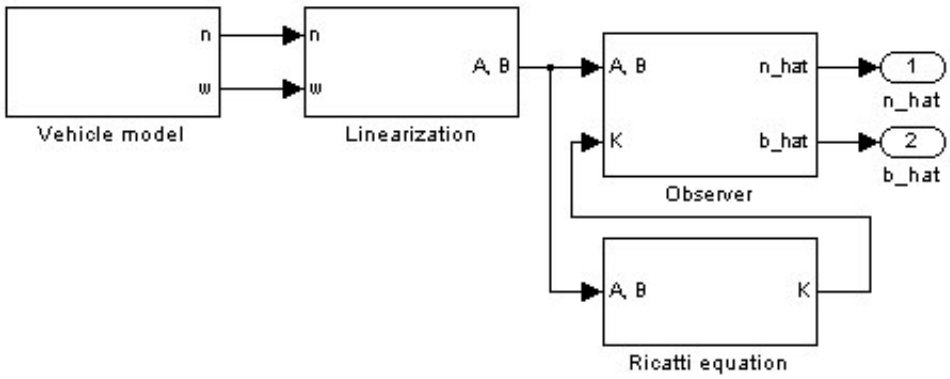

Figure 6.1: Outline of Euler angle EKF simulation

6.1.1 Analysis metrics

For analysis of the output, the error metric used is root-sum-square of the difference between the true η and estimated $\hat{\eta}$. The same algorithm is used to compare bias estimates by taking the root-sum-square of the difference between the estimated bias $\hat{b}$ and the known model bias b. Given the vector $\eta = [\phi, \theta, \psi]^T$ that is estimated by $\hat{\eta} = [\hat{\phi}, \hat{\theta}, \hat{\psi}]^T$, the root-sum-square (RSS) error is

$$e_{RSS} = \sqrt{(\phi - \hat{\phi})^2 + (\theta - \hat{\theta})^2 + (\psi - \hat{\psi})^2}$$

Many simulation trials used a bias error of $b = [0.01, 0.02, 0.03]^T$; if the observer assumed a constant zero bias error for these trials, the root-sum-square error would be

$$\sqrt{0.01^2 + 0.02^2 + 0.03^2} = 0.037\,\text{rad/s}$$

Estimates of bias error will be compared by how fast the error converges below certain percentages of this value.

Due to the modulo 2π nature of η, the maximum error between true η and estimated $\hat{\eta}$ is 360 degrees in each axis. The root-sum-square error

corresponding to this case would be

$$\sqrt{3(2\pi)^2} = 10.9\,\text{rad}$$

This number will be used for convergence comparisons of η.

6.1.2 Results of Euler angle EKF

The first step in testing the simulation was to verify operation over a range of possible inputs. Care needed to be taken to remain within certain bounds for the simulation, as the vehicle model shown in Appendix A.1 is sensitive to high frequencies, and any output that exceeds the bounds in place due to the arcsine and arctangent functions wraps around at their respective upper bounds of $\pi/2$ and π. The vehicle is not physically capable of maneuvers that approach these boundaries, so no effort was made to have the observer accommodate for them. As such, any inputs selected that approach these boundaries reflected poorly on the observer. A sample of the inputs used for verification can be seen in Appendix A.5.1. The input parameters listed in Table 6.1 were held constant for all trials after the input verification was completed.

Parameter	*All Trials*
a_ϕ	1.0
f_ϕ	0.02
a_θ	0.5
f_θ	0.015
a_ψ	1.0
f_ψ	0.02

Table 6.1: Constant input parameters for all trials

An example test case was set up with the parameters detailed in Table 6.2 in order to illustrate a typical simulation run. The vehicle model outputs the angular rate without the bias error, and the angular position generated from the rotation matrix kinematic model using the biased angular rate. Figure 6.2 compares the true and measured angular velocity about

Parameter	*Trial 1*
b_x	0.01
b_y	0.02
b_z	0.03
ϕ_i	0°
θ_i	0°
ψ_i	0°
b_{xi}	0.1
b_{yi}	0.2
b_{zi}	0.3

Table 6.2: Parameters for Trial 1

the x-axis during the first 20 minutes of simulation. The extended kalman filter uses the vehicle model outputs to generate the estimate $[\hat{\eta}, \hat{b}]^T$. Figure 6.3 compares the correct value of ϕ to the estimated value from the observer, while Figure 6.4 depicts the root-sum-square of the error between η and $\hat{\eta}$. The observer tracked η within one percent of the maximum error within 3.38 seconds. Figure 6.5 depicts the estimation error over the full simulation time. The steady state error is less than 0.12 percent of the maximum error.

The bias error tracking is shown in Figure 6.6 over the first 20 minutes of simulation. The root-sum-square error between the real and estimated bias errors over the first minute of the simulation can be seen in Figure 6.7. The simulation beats a constant zero estimate in 8.67 seconds. As seen in Figure 6.8, the estimation error falls below two percent of the constant zero estimate within 2.07×10^3 seconds (just over half an hour), and continues to decrease over time, but at a slow rate; after two hours the error falls just below one percent of the constant zero estimate.

The Kalman filter is known to be globally asymptotically stable, and linearization causes this property to apply only locally to the EKF. An interesting set of trials is to try various initial conditions in an attempt to find values that cause the observer to fail. The parameters used in each simulation run are shown in Table 6.3. Trial eight uses a bias error close to the highest possible allowed by the limitations of the simulation model, and

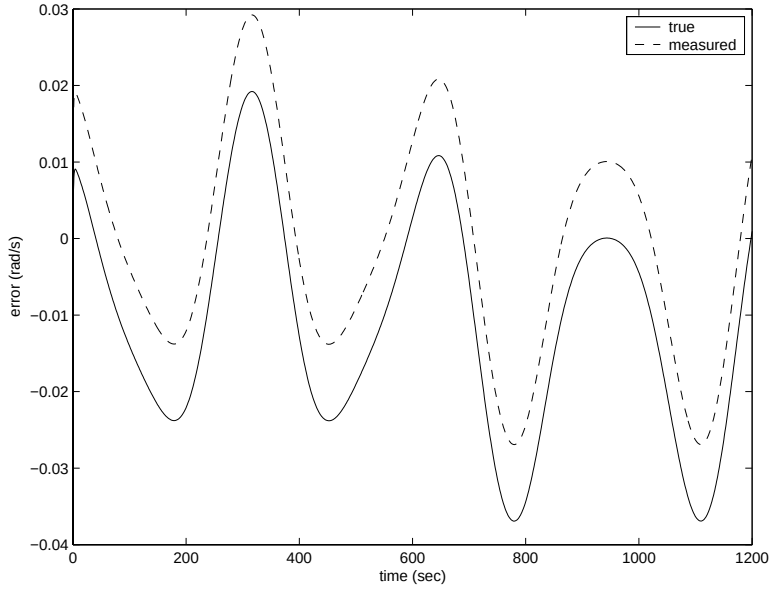

Figure 6.2: ω and ω_m about the x-axis in Trial 1

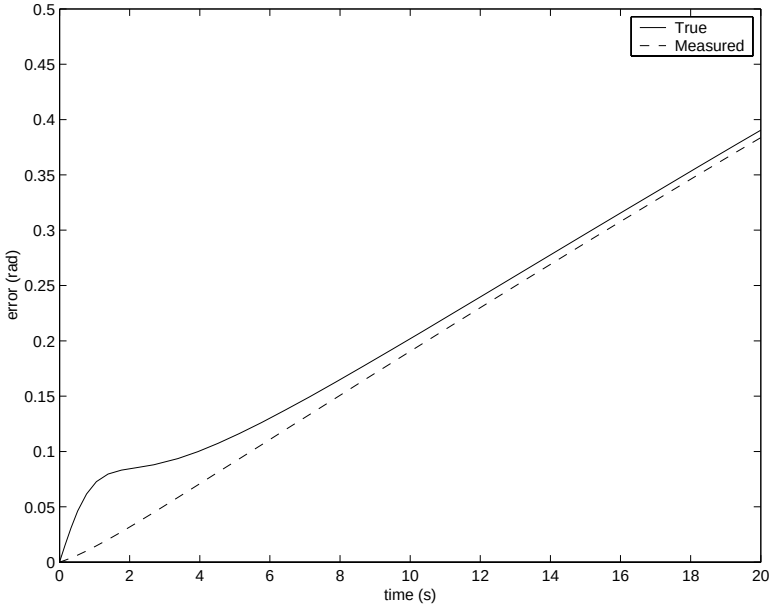

Figure 6.3: ϕ and $\hat{\phi}$ in Trial 1

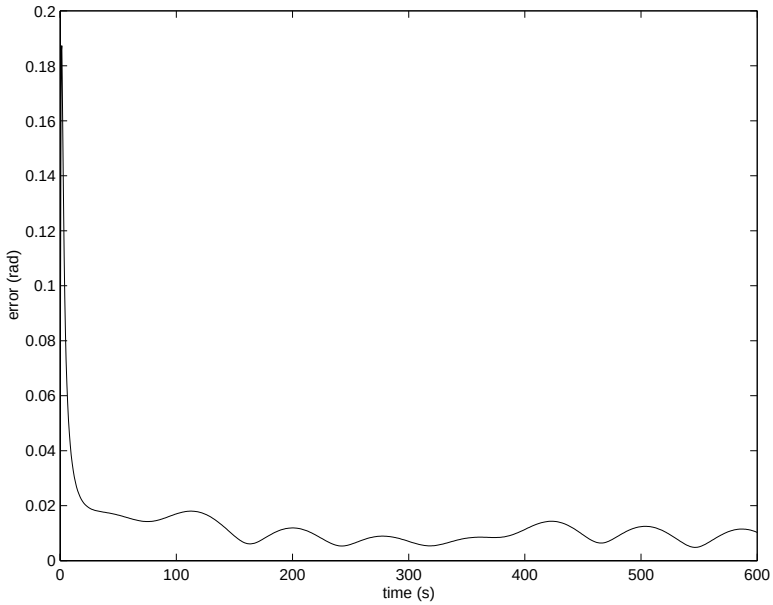

Figure 6.4: RSS estimation error for η in Trial 1

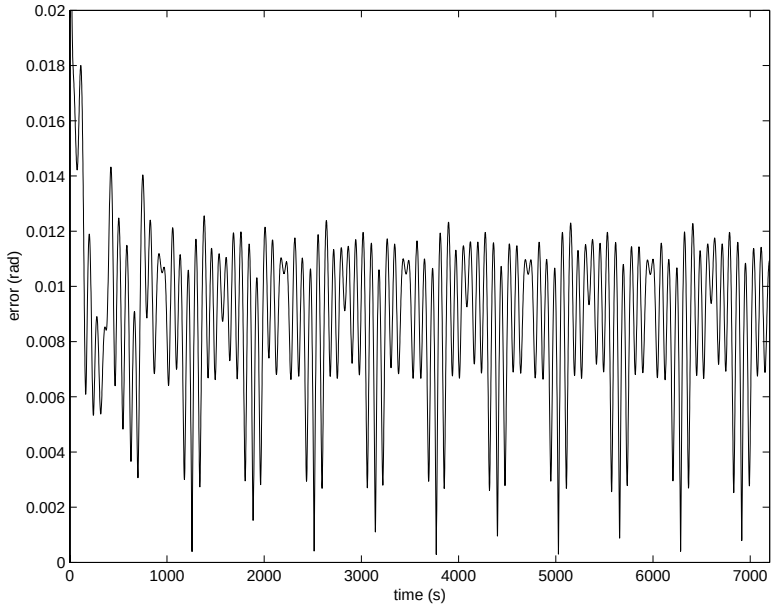

Figure 6.5: RSS steady state estimation error for η in Trial 1

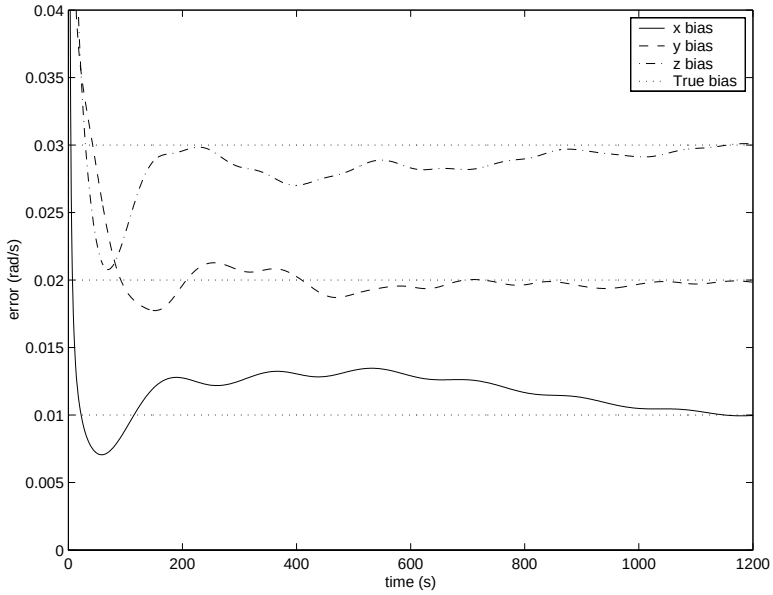

Figure 6.6: Bias estimate for Trial 1

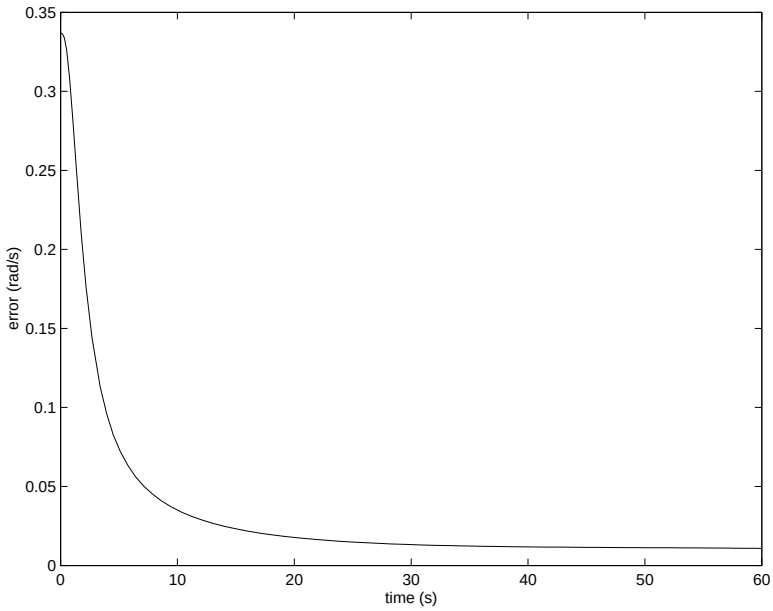

Figure 6.7: RSS estimation error for bias error in Trial 1

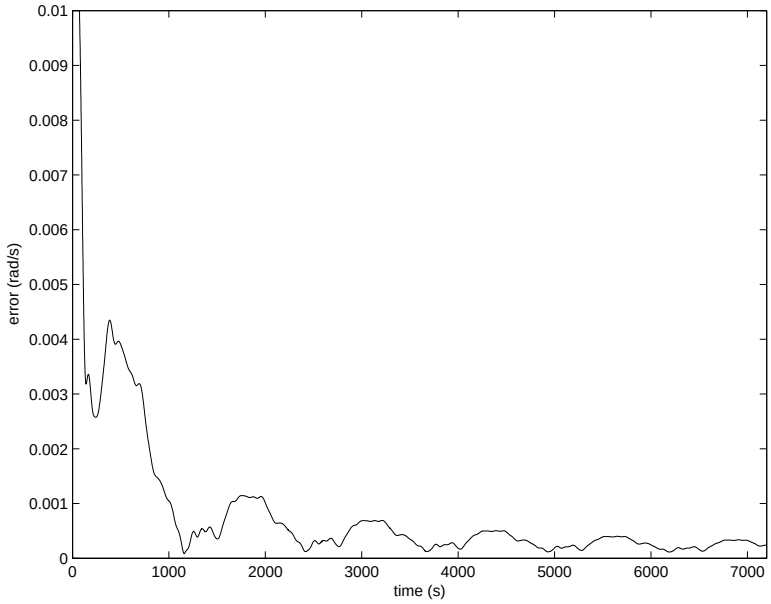

Figure 6.8: RSS steady state estimation error for bias error in Trial 1

Parameter	*Trial 8*	*Trial 11*	*Trial 12*	*Trial 13*	*Trial 14*
b_x	0.01	0.01	0.01	0.01	0.01
b_y	0.02	0.02	0.02	0.02	0.02
b_z	0.03	0.03	0.03	0.03	0.03
ϕ_i	0°	0°	15°	-80°	15°
θ_i	0°	0°	-5°	45°	-5°
ψ_i	0°	0°	1°	60°	1°
b_{xi}	100	0.01	0.01	0.01	1.0
b_{yi}	200	0.02	0.02	0.02	2.0
b_{zi}	300	0.03	0.03	0.03	3.0

Table 6.3: Parameters for Trials 8, 11, 12, 13, and 14

several orders of magnitude higher than any bias error expected from a real system. In this trial the bias estimate takes far longer to converge: 4.27×10^3 seconds to fall below the constant zero estimate, as seen in Figure 6.9. It will, however, eventually reach the values reached by guessing the bias error exactly.

The other option for incorrect initial conditions is the initial position of the vehicle. Three trials were set to have correct guesses for bias error, with trial 11 having a correct guess for vehicle starting position, trial 12 makes small errors, and trial 13 guesses far from the real position. Figure 6.10 depicts the bias error in the first five minutes of each trial. The estimate again converges, but more slowly the further from the truth the initial conditions are. Combining poor guesses in both bias error and starting position yields similar results, as shown in Figure 6.11, comparing this trial to the perfect guess of trial 11. Table 6.4 summarizes the convergence metrics of the trials used in initial condition testing.

Trial	*100%*	*10%*	*2%*
Trial 8	4272	>7200	>7200
Trial 11	0	551.1	2074
Trial 12	3.611	524.8	2065
Trial 13	28.90	777.0	4497
Trial 14	93.31	819.5	3349

Table 6.4: Bias estimation convergence in seconds for Trials 8, 11, 12, 13, and 14

The results of initial condition testing seem to indicate that the radius of convergence for the EKF is beyond the scope of the limits of the simulation, and thus the system should be asymptotically stable for any values that will be seen in experimental systems.

Preliminary tests indicated that the bias error was on the order of 10^{-5} for the IMU used in the prototype [12]. The results of the bias error initial condition tests raise the question of how the magnitude of true bias value will effect the response time of the observer. To test this, trials one through five vary the true bias error while keeping the initial bias error guess constant; the

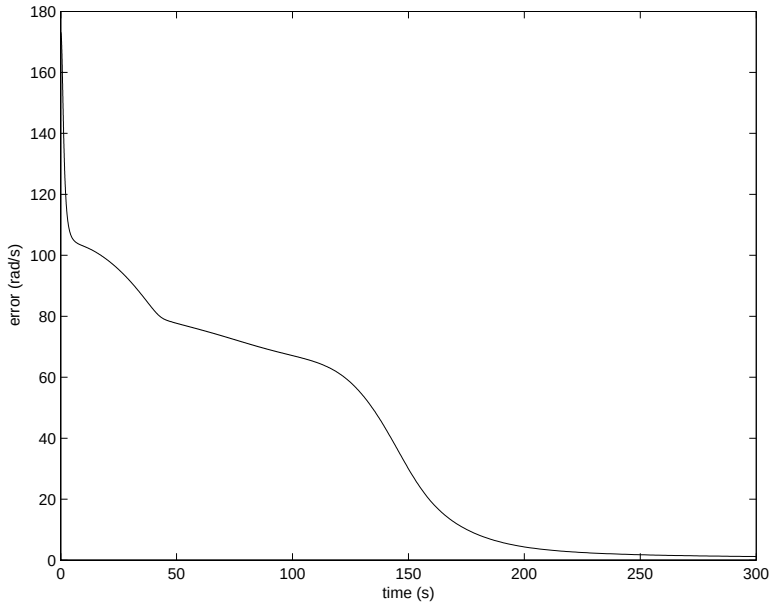

Figure 6.9: RSS estimation error for bias error in Trial 8

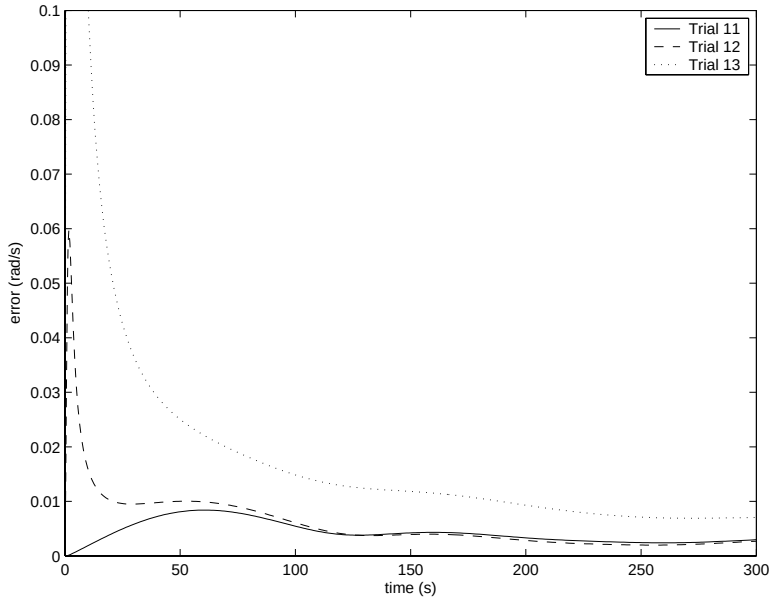

Figure 6.10: RSS estimation error for bias error in Trials 11, 12, and 13

parameters for these trials are detailed in Table 6.5. Figure 6.12 depicts the

Parameter	*Trial 1*	*Trial 2*	*Trial 3*	*Trial 4*	*Trial 5*
b_x	0.01	0.001	0.0001	0.00001	1×10^{-12}
b_y	0.02	0.002	0.0002	0.00002	2×10^{-12}
b_z	0.03	0.003	0.0003	0.00003	3×10^{-12}
ϕ_i	0°	0°	0°	0°	0°
θ_i	0°	0°	0°	0°	0°
ψ_i	0°	0°	0°	0°	0°
b_{xi}	1.0	1.0	1.0	1.0	1.0
b_{yi}	2.0	2.0	2.0	2.0	2.0
b_{zi}	3.0	3.0	3.0	3.0	3.0

Table 6.5: Parameters for Trials 1–5

first two minutes of simulation for trials one through four. As can be seen, an error of several orders of magnitude when the bias values are extremely small did not have the profound effect that a relatively small order of magnitude error had when the bias values were much larger. Another interesting result is that the estimation error seems to be approaching a maximum, as can be seen by zooming in on Figure 6.12, shown in Figure 6.13, with each order of magnitude difference having less impact than the previous.

Trial five was used to determine a practical limit on the estimation error. This bias error is on the order of machine errors that might be seen in a microprocessor used to implement the observer. Figure 6.14 shows the result of this minuscule bias error compared to the results of trials three and four to give a sense of scale. The Table 6.6 summarizes the effect of small bias magnitudes on time to converge to certain percentages of the constant zero estimate.

The results of the EKF applied to the Euler angle model indicate that the bias error can be autonomously calibrated for a system in which bias is the only significant error. This conclusion is reassuring, but the use of an Euler angle model prohibits comparison with other work in the field, which predominantly uses quaternion-based models. The results also indicate that the bias error does not scale linearly with the bias error it observes, meaning that two systems with several magnitudes of difference in their small

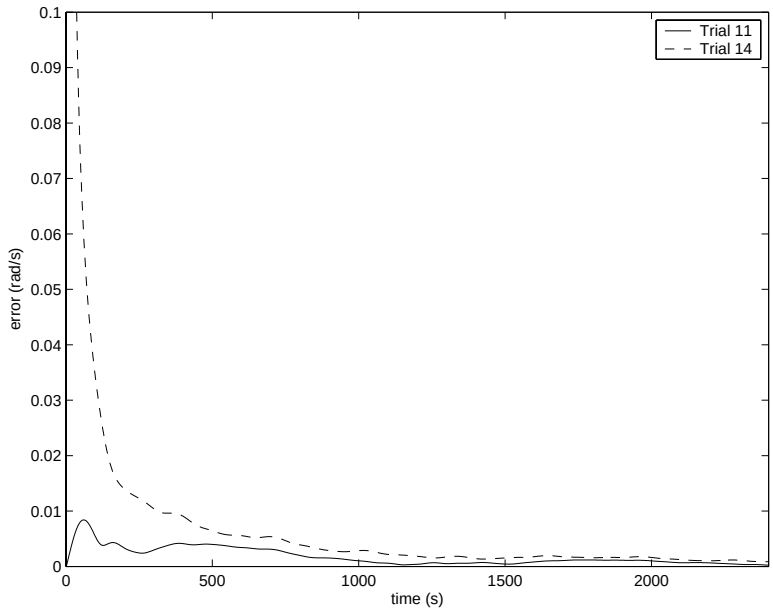

Figure 6.11: RSS estimation error for bias error in Trials 11 and 14

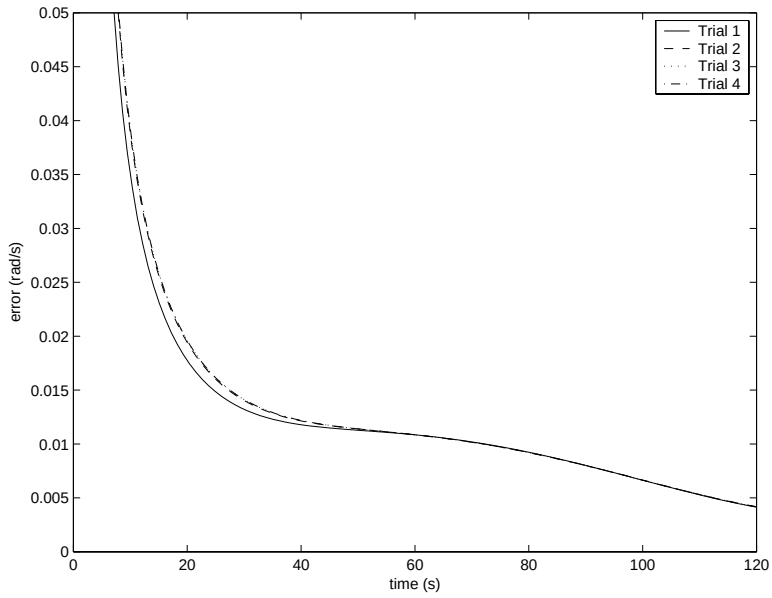

Figure 6.12: RSS estimation error for bias error in Trials 1, 2, 3, and 4

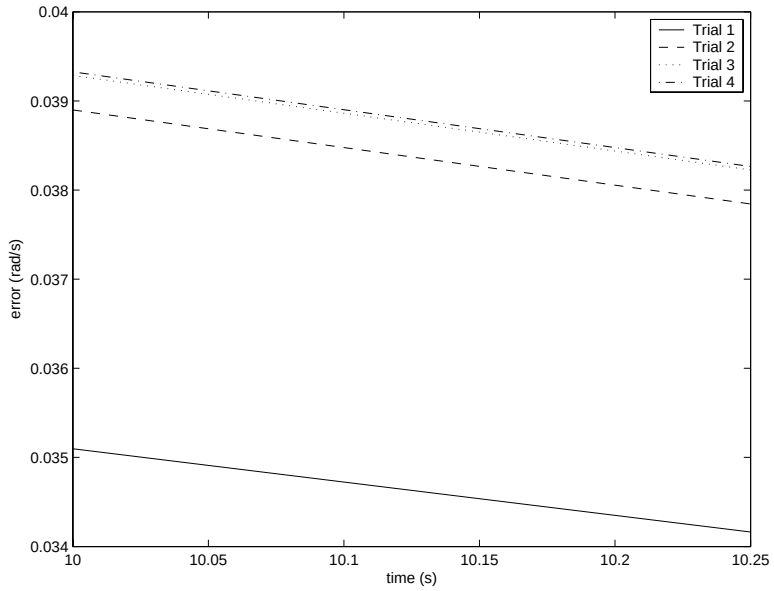

Figure 6.13: Closer look at RSS estimation error for bias error in Trials 1, 2, 3, and 4

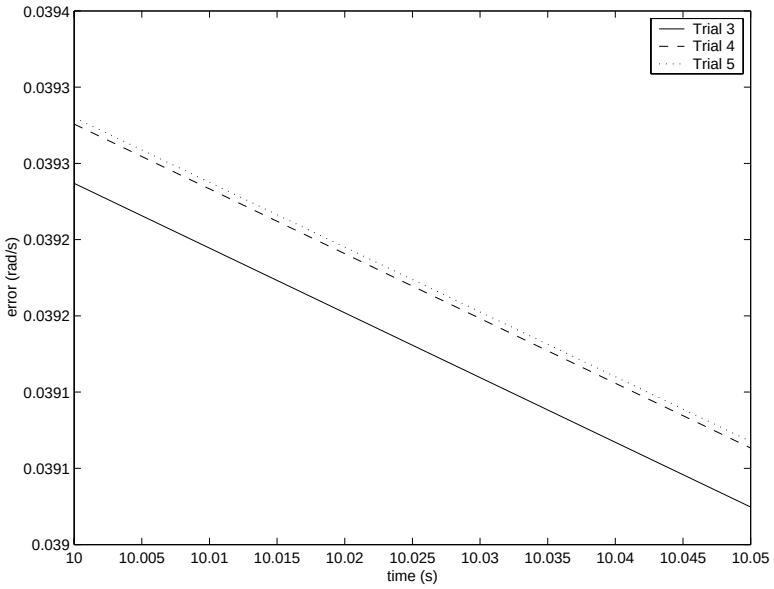

Figure 6.14: Very close look at RSS estimation error for bias error in Trials 3, 4, and 5

Trial	*100%*	*10%*	*2%*
Trial 1	8.666	538.9	2070
Trial 2	10.28	532.2	2068
Trial 3	10.30	532.2	2066
Trial 4	10.30	532.4	2067
Trial 5	10.30	531.7	2067

Table 6.6: Euler angle EKF convergence in seconds for Trials 1–5

magnitude bias error can expect similar estimation errors.

6.2 Quaternion extended kalman filter

The same state augmentation used in Section 6.1 can be used to add bias error to the quaternion model. Recall the quaternion kinematic model

$$\dot{q}(t) = \frac{1}{2}T(q(t))w(t) \tag{6.5}$$

After adding bias error and process noise to the equation, it can be augmented to

$$\begin{bmatrix} \dot{q}(t) \\ \dot{b}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}T(q(t))(w(t) + b(t) + v(t)) \\ 0 \end{bmatrix} \tag{6.6}$$

The system is linearized around the operating point $\hat{\eta}$, $\hat{b}$, u:

$$\begin{bmatrix} \dot{q}(t) \\ \dot{b}(t) \end{bmatrix} \approx \widetilde{A} \begin{bmatrix} q(t) \\ b(t) \end{bmatrix} + \widetilde{B}\omega(t) + \widetilde{L}v(t) \tag{6.7}$$

where

$$\widetilde{A} = \frac{1}{2}\begin{bmatrix} \frac{\partial T(q(t))b(t)}{\partial q(t)} & \frac{\partial T(q(t))b(t)}{\partial b(t)} \\ 0 & 0 \end{bmatrix}$$
$$\widetilde{B} = \frac{1}{2}\begin{bmatrix} \frac{\partial T(q(t))\omega(t)}{\partial \omega(t)} \\ 0 \end{bmatrix}$$
$$\widetilde{L} = \frac{1}{2}\begin{bmatrix} \frac{\partial T(q(t))v(t)}{\partial v(t)} \\ 0 \end{bmatrix}$$

Let

$$q = \begin{bmatrix} v_o \\ v_x \\ v_y \\ v_z \end{bmatrix}, \quad b = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

Computing the derivatives yields

$$\frac{\partial T(q)b(t)}{\partial b(t)} = \frac{\partial T(q)v(t)}{\partial v(t)} = \frac{\partial T(q(t))\omega(t)}{\partial \omega(t)} = T(q(t))$$
$$\frac{\partial T(q)b(t)}{\partial q(t)} = \frac{1}{2}\begin{bmatrix} 0 & -b_x & -b_y & -b_z \\ b_x & 0 & b_z & -b_y \\ b_y & -b_z & 0 & b_x \\ b_z & b_y & -b_x & 0 \end{bmatrix}$$

Applying the generalized EKF (5.2) to the linearized quaternion model (6.7) yields

$$\begin{bmatrix} \dot{\hat{q}}(t) \\ \dot{\hat{b}}(t) \end{bmatrix} = \widetilde{A}\begin{bmatrix} \hat{q}(t) \\ \hat{b}(t) \end{bmatrix} + \widetilde{B}\omega(t) + K(y - \hat{y}) \tag{6.8}$$
$$\hat{y} = \hat{q}(t)$$
$$K = PC^T R^{-1}$$
$$\dot{P} = AP + PA^T + LQL^T - PC^T R^{-1} CP$$

The simulation is set up identically to the Euler angle simulation, as

outlined in Figure 6.15 and seen in detail in Appendix A.3. The covariance matrices are again chosen as

$$P_o = Q = R = I$$

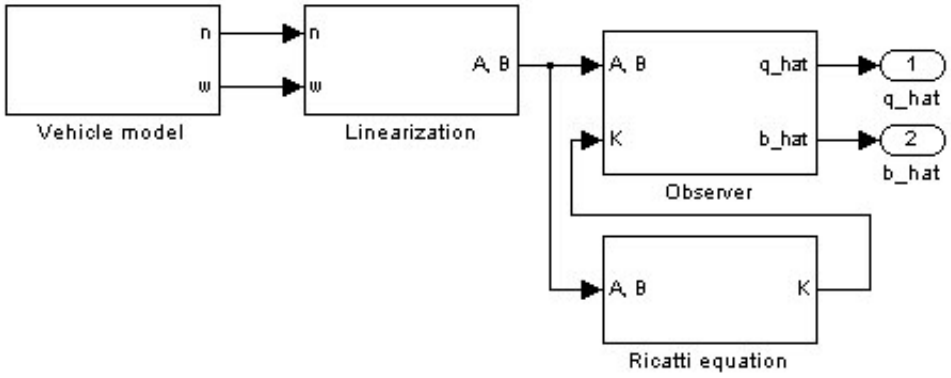

Figure 6.15: Outline of Euler angle EKF simulation

In order to allow the same metrics to be used, $\hat{q}(t)$ is converted to $\hat{\eta}$ by transforming $\hat{q}(t)$ into a rotation matrix by the algorithm listed in (4.7), then extracting $\hat{\eta}$ by the algorithm listed in (4.5). The direct transformation from $\hat{q}(t)$ to $\hat{\eta(t)}$ is thus

$$\begin{aligned}
\phi &= \arctan\left(\frac{2(v_2 v_3 + v_o v_1)}{v_o^2 - v_1^2 - v_2^2 + v_3^2}\right) \\
\theta &= \arcsin\left(2v_o v_2 - 2v_1 v_3\right) \\
\psi &= \arctan\left(\frac{2(v_1 v_2 + v_o v_3)}{v_o^2 + v_1^2 - v_2^2 - v_3^2}\right)
\end{aligned}$$

6.2.1 Results of quaternion EKF

The quaternion EKF was run through the same initial trials discussed in Section 6.1.2 in order to verify operation.

The example case used to demonstrate the Euler angle EKF, with parameters outlined in Tables 6.1 and 6.2, was repeated for the quaternion EKF

to allow a direct comparison. Figure 6.16 depicts the estimation error for η for both the quaternion and Euler angle EKF, showing that the systems are comparable, as expected. The quaternion appears to be performing worse than the Euler angle model, converging within one percent of the maximum error range within 6.44 seconds, compared to the 3.38 second convergence for the Euler angle EKF. As seen in Figure 6.17, the steady state error in η is worse in the quaternion EKF as well, at about 0.28 percent of the maximum error range compared to 0.12 percent for the Euler angle EKF. This apparent deficiency is not mirrored in the bias error estimation, however. As seen in Figure 6.18, the quaternion EKF converges below the constant zero estimate within 6.44 seconds, compared to 8.66 seconds for the Euler angle EKF. Figure 6.19 shows that the quaternion EKF bias error estimation error is slightly lower than the steady state error observed in the Euler angle EKF.

It quickly became apparent during some trials that the quaternion EKF contained a limitation not present in the Euler angle EKF. For very high bias errors like those tested in trial eight, the observer overshoots the true η value by a large amount, then undershoots it again immediately. This effect has the same net effect as the Euler angle model's slow convergence time, but takes far longer to simulate. This behavior could possibly be rectified by adjustment of the Q and R matrices, but given the small order of magnitude expected for bias errors, any effort put into fine tuning the observer for erroneous guesses over five orders of magnitude too large is wasted. If a new IMU proves to have much larger bias errors further work would be warranted, but for the time being, the values of the initial bias guess for trials eight and fourteen were changed from $b = [100, 200, 300]^T$ to $b = [1.1, 1.1, 1.1]^T$, which was close to the upper limit of bias values possible before the observer took drastically longer to converge. Table 6.7 details the simulation parameters. Figure 6.20 depicts what remained of the model's limitation with the new values. It should be noted that this artifact lasted only six seconds in a two hour simulation, but required almost twenty percent of the simulation time.

The initial condition tests showed similar results to the Euler angle EKF,

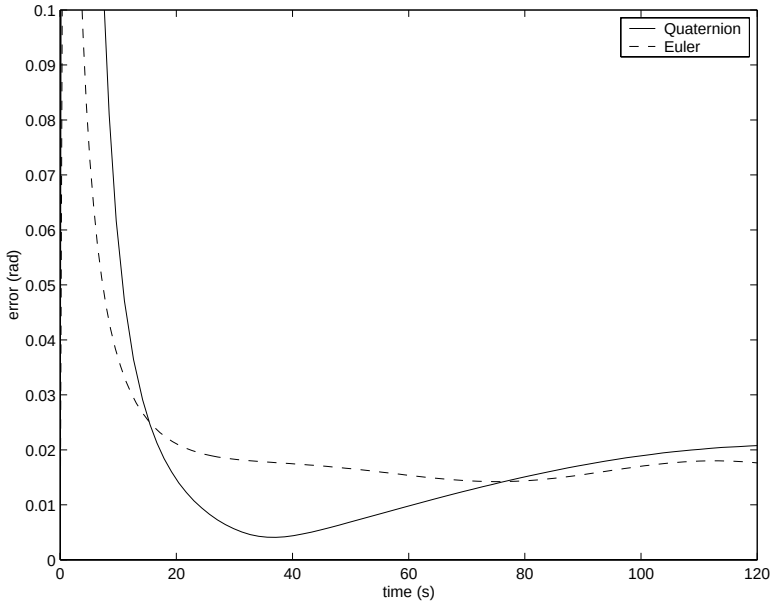

Figure 6.16: Comparison of RSS estimation error for η between quaternion and Euler angle EKF in Trial 1

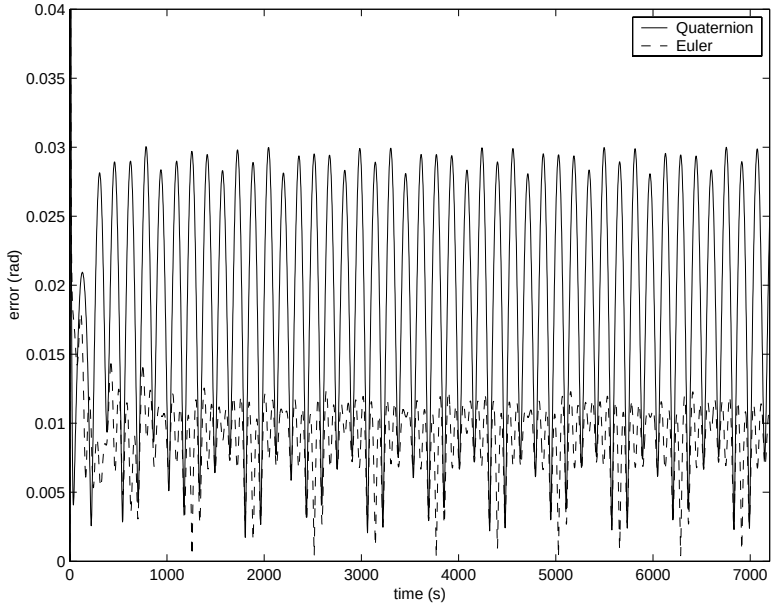

Figure 6.17: Comparison of steady state RSS estimation error for η between quaternion and Euler angle EKF in Trial 1

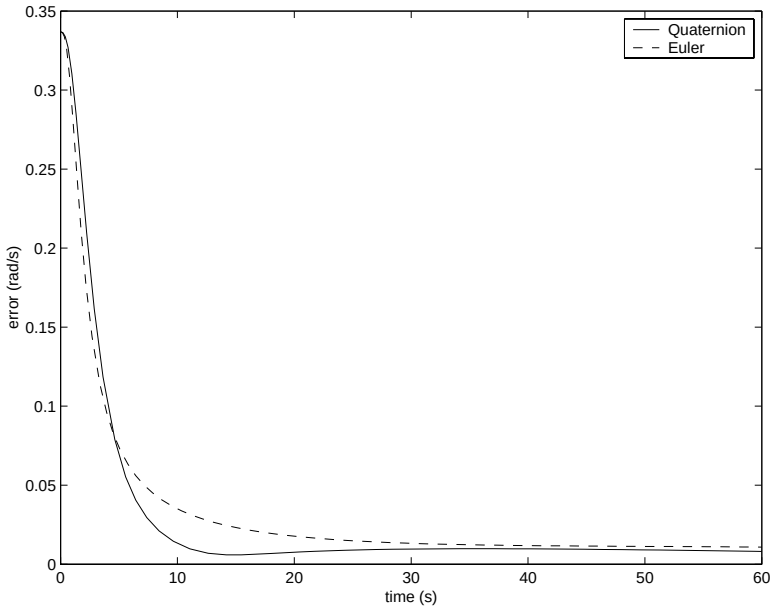

Figure 6.18: Comparison of RSS estimation error for bias error between quaternion and Euler angle EKF in Trial 1

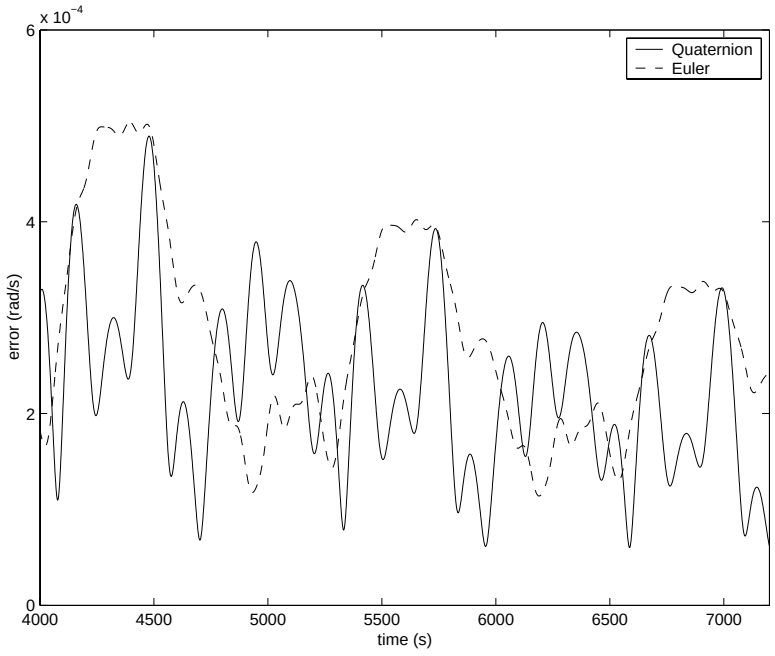

Figure 6.19: Comparison of RSS steady state estimation error for bias error between quaternion and Euler angle EKF in Trial 1

Parameter	*Trial 8*	*Trial 11*	*Trial 12*	*Trial 13*	*Trial 14*
b_x	0.01	0.01	0.01	0.01	0.01
b_y	0.02	0.02	0.02	0.02	0.02
b_z	0.03	0.03	0.03	0.03	0.03
ϕ_i	0°	0°	15°	-80°	15°
θ_i	0°	0°	-5°	45°	-5°
ψ_i	0°	0°	1°	60°	1°
b_{xi}	1.1	0.01	0.01	0.01	1.0
b_{yi}	1.1	0.02	0.02	0.02	2.0
b_{zi}	1.1	0.03	0.03	0.03	3.0

Table 6.7: Parameters for Trials 8, 11, 12, 13, and 14, adjusted for quaternion EKF

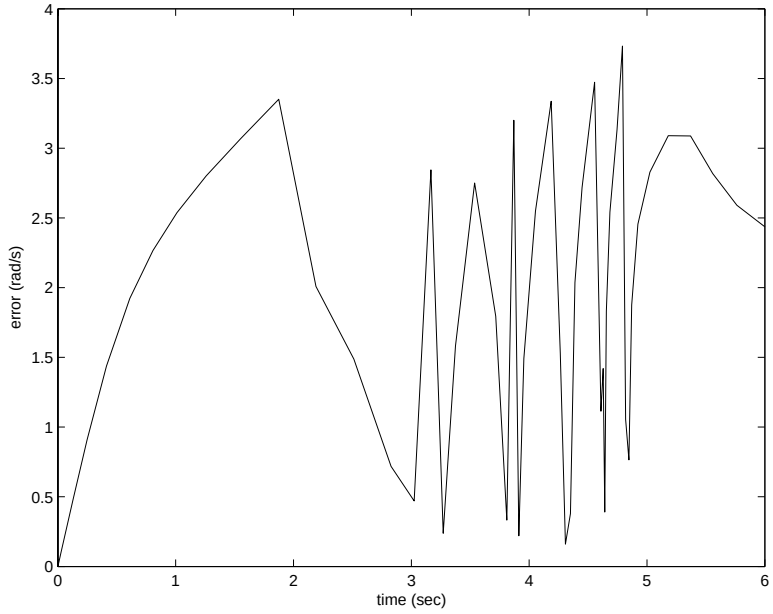

Figure 6.20: RSS estimation error for η in Trial 8

Table 6.8 summarizes the convergence metrics, and plots of the estimates can be seen in Appendix A.5.2. Figure 6.21 compares the quaternion and Euler

Trial	*100%*	*10%*	*2%*
Trial 8	23.45	414.8	2461
Trial 11	0	416.2	2460
Trial 12	0	415.8	2459
Trial 13	5.990	417.1	2459
Trial 14	22.97	414.2	2460

Table 6.8: Quaternion EKF bias estimation convergence in seconds for Trials 8, 11, 12, 13, and 14

angle EKF bias error estimation for trials one and five, using the same parameters detailed in Table 6.5. As can be seen, the quaternion EKF behaves similar to the Euler angle observer in the presence of small bias errors. Table 6.9 summarizes the time each trial takes to converge within certain percentages of the constant zero estimate. This table can be compared directly to Table 6.6.

Trial	*100%*	*10%*	*2%*
Trial 1	6.439	416.7	2459
Trial 2	6.418	416.7	2466
Trial 3	6.412	416.3	2468
Trial 4	6.412	416.2	2468
Trial 5	6.412	416.2	2468

Table 6.9: Quaternion EKF convergence in seconds for Trials 1–5

These results show that the quaternion-based EKF retains the important properties of the Euler angle EKF. A system in which bias error is the only significant error source can thus be autonomously calibrated, and because it is based upon quaternions, it can be compared to other work in the field of autonomous calibration.

6.3 Nonlinear observer

D.J. Stilwell developed a nonlinear observer proven to be globally asymptotically stable [12], the proof of which is reproduced in Appendix C. Global asymptotic stability is a large advantage over the kalman filter, which is proven only to be locally asymptotically stable.

For the system in (4.6), let $\hat{R}$ be an estimate of R, and let $\widetilde{R} = \hat{R}^T R$ be the estimation error. From the properties of rotation matrices, $\widetilde{R} = I$ if the estimate is perfect. The nonlinear observer is defined as

$$\begin{aligned} \dot{\hat{R}}(t) &= S\Big(\widetilde{R}^T\Big(\omega + \hat{b} + k_1 e(t)\mathrm{sgn}(e_o(t))\Big)\Big)\hat{R} \\ \dot{\hat{b}}(t) &= k_2 e(t)\,\mathrm{sgn}(e_o(t)) \end{aligned} \tag{6.9}$$

where $[e_o, e]^T$ is the quaternion representation of $\widetilde{R}$, k_1 and k_2 are positive scalars, and the signum function is defined as

$$\mathrm{sgn}(e_o) = \begin{cases} 1 & e_o \geq 0 \\ -1 & e_o < 0 \end{cases}$$

Results of quaternion nonlinear observer

The quaternion nonlinear observer was run through the same verification trials as the other two observers, the results of which can be seen in Appendix A.5.1. The difference between an observer based upon the true nonlinear model and one based upon a linearization is immediately apparent when viewing these results. The quaternion nonlinear observer converges faster than the quaternion EKF for all cases, as demonstrated in Figure 6.22, which depicts a comparison of the first four minutes of the nonlinear and EKF quaternion observers. The nonlinear observer converges within two percent of the constant zero estimate just after the quaternion EKF has converged within 100 percent, as shown in Table 6.10, which is directly comparable to Table 6.9.

The disadvantage of this observer is that it is incapable of dealing with unmodeled dynamics. The real system will always contain unmodeled dy-

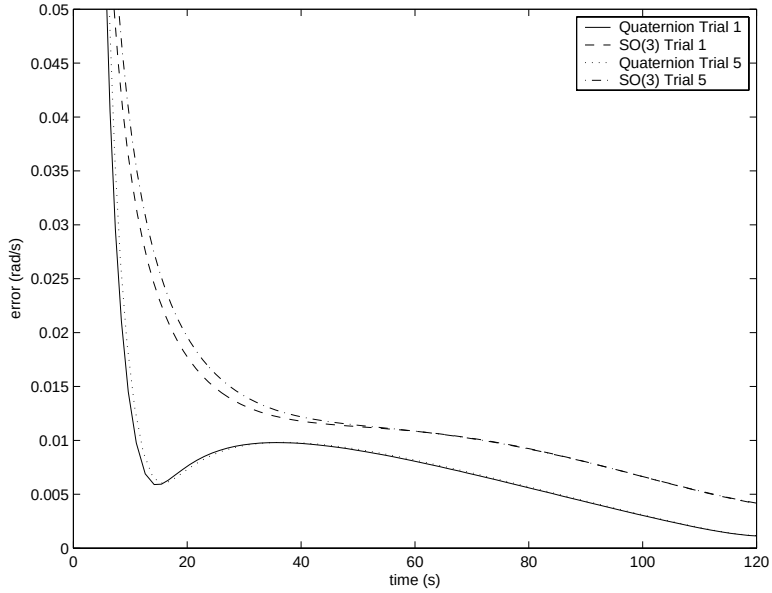

Figure 6.21: Comparison of RSS estimation for bias error between quaternion and Euler angle observers in Trials 1 and 5

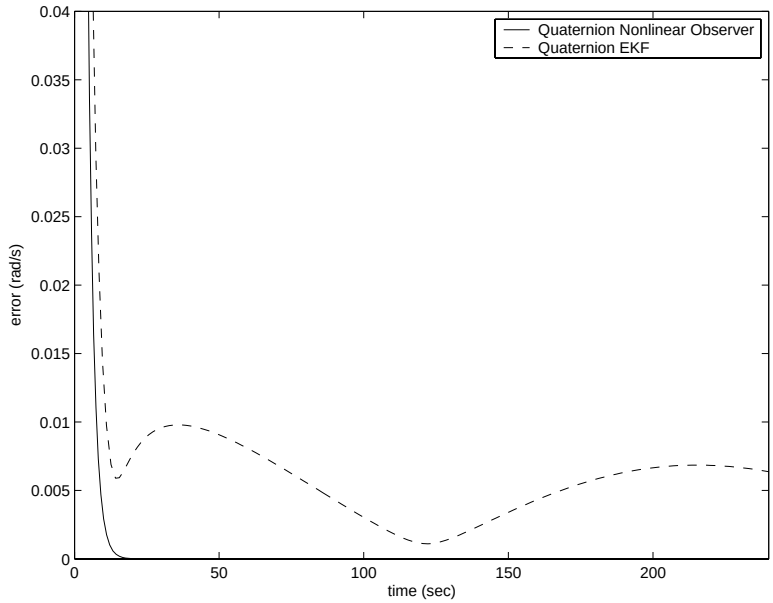

Figure 6.22: Comparison of RSS estimation error for bias error between nonlinear and EKF quaternion observers in Trial 1

Trial	*100%*	*10%*	*2%*
Trial 1	4.571	9.119	12.15
Trial 2	5.072	8.883	12.93
Trial 3	5.061	9.771	12.89
Trial 4	5.060	9.768	12.89
Trial 5	5.060	9.768	12.89

Table 6.10: Quaternion nonlinear observer convergence in seconds for Trials 1–5

namics, and it is not possible to have a perfectly accurate model of the system.

6.4 Toward experimental application

Real data contains many further error terms, as discussed in Section 3.4. Before applying any of the simulated observers to the vehicle, it must be determined if the results shown above will still work with additional error sources taken into account.

6.4.1 Scale and bias model

The next error source to consider is scale factor. With the scale factor and bias error added, the euler angle model becomes

$$\dot{\eta}(t) = Q(\eta(t))(A(t)\omega(t) + b(t))$$

where $b(t) \in \mathbb{R}^3$, $\omega(t) \in \mathbb{R}^3$, and

$$A(t) = \begin{bmatrix} a_1(t) & 0 & 0 \\ 0 & a_2(t) & 0 \\ 0 & 0 & a_3(t) \end{bmatrix} \tag{6.10}$$

If the system is assumed to move only about the z-axis, it can be simplified to $\dot{\psi} = a\omega + b$. Using the same augmented state technique used for finding

bias error, the system becomes

$$\begin{bmatrix} \dot{\psi}(t) \\ \dot{a} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} a\omega(t) + b \\ 0 \\ 0 \end{bmatrix} \tag{6.11}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ a \\ b \end{bmatrix} = \psi$$

In order to apply an EKF to this system, it must be linearized. Let $\tilde{\omega}(t)$ be an arbitrary time-varying trajectory, $\omega(t)$ can be defined as $\omega(t) = \tilde{\omega}(t) + \omega_\delta(t)$. Using similar notation for the state variables, the system can be written

$$\begin{bmatrix} \dot{\tilde{\psi}} + \dot{\psi}_\delta \\ \dot{\tilde{a}} + \dot{a}_\delta \\ \dot{\tilde{b}} + \dot{b}_\delta \end{bmatrix} = \begin{bmatrix} (\tilde{a} + a_\delta)(\tilde{\omega} + \omega_\delta) + \tilde{b} + b_\delta \\ 0 \\ 0 \end{bmatrix}$$

which linearizes to

$$\begin{bmatrix} \dot{\psi}_\delta(t) \\ \dot{a}_\delta(t) \\ \dot{b}_\delta(t) \end{bmatrix} = \widetilde{A}(t) \begin{bmatrix} \psi_\delta(t) \\ a_\delta(t) \\ b_\delta(t) \end{bmatrix} + \widetilde{B}(t)\omega_\delta(t)$$

$\widetilde{A}$ and $\widetilde{B}$ are the partial differential equations of $\dot{\psi} = a\omega + b$

$$\widetilde{A} = \begin{bmatrix} \frac{\partial \dot{\psi}}{\partial \psi} & \frac{\partial \dot{\psi}}{\partial a} & \frac{\partial \dot{\psi}}{\partial b} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Bigg|_{\omega=\tilde{\omega}} = \begin{bmatrix} 0 & \tilde{\omega} & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\widetilde{B} = \begin{bmatrix} \frac{\partial \dot{\psi}}{\partial \omega} \\ 0 \\ 0 \end{bmatrix} \Bigg|_{\omega=\tilde{\omega}} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$

So the full linearized system is written

$$\begin{bmatrix} \dot{\psi}_\delta(t) \\ \dot{a}_\delta(t) \\ \dot{b}_\delta(t) \end{bmatrix} = \begin{bmatrix} 0 & \tilde{\omega}(t) & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_\delta(t) \\ a_\delta(t) \\ b_\delta(t) \end{bmatrix} + \begin{bmatrix} a(t) \\ 0 \\ 0 \end{bmatrix} \omega_\delta(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_\delta(t) \\ a_\delta(t) \\ b_\delta(t) \end{bmatrix}$$

6.4.2 Observability

A system described by

$$\dot{x} = A(t)x(t), \quad x(t_o) = x_o, \quad t \geq 0$$
$$y(t) = C(t)x(t)$$

is observable over the interval $[t_o, t_f]$ if and only if the observability Gramian

$$M(t_o, t_f) = \int_{t_o}^{t_f} \Phi^T(t, t_o) C^T(t) C(t) \Phi(t, t_o) dt$$

is invertible. The transition matrix is calculated as

$$\Phi(t, t_o) = I + \int_{t_o}^{t} A(\sigma_1) d\sigma_1 + \int_{t_o}^{t} A(\sigma_1) \int_{t_o}^{\sigma_1} A(\sigma_2) d\sigma_2 \sigma_1 + \dots$$

$$\int_{t_o}^{t} A(\sigma_1) d\sigma_1 = \begin{bmatrix} 0 & \int_{t_o}^{t} \tilde{\omega}(\sigma) d\sigma & t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The first integral term results in a nilpotent matrix, so all further terms are zero. This leaves a resulting transition matrix of

$$\Phi(t, t_o) = \begin{bmatrix} 1 & \int_{t_o}^{t} \tilde{\omega}(\sigma) d\sigma & t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For clarity, let $s = \int_{t_o}^{t} \tilde{\omega}(\sigma) d\sigma$. The observability Gramian is then

$$M(t_o, t_f) = \int_{t_o}^{t_f} \begin{bmatrix} 1 & s & t \\ s & s^2 & ts \\ t & ts & t^2 \end{bmatrix} dt$$

To conclude that the system is not experimentally applicable in general, it is sufficient to show that the system is not observable for a specific trajectory. For example, consider the trajectory $\tilde{\omega}(t) = 2$ over the interval $[0, 1]$; the determinate of $M(0, 1)$ is zero, and thus the system is not observable.

This proves that there exists a trajectory such that the euler angle model is not observable when restricted to the z-axis. This implies that, in general, the system is not experimentally applicable. This result correlates with experimental results seen in [14] where the system described in (6.11) converged to different values depending on the initial conditions.

Observability for the arbitrary trajectory and the full three degree of freedom model proved to be computationally complex, and is left for future work.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

With the desire to develop small, inexpensive vehicles comes the motivation for autonomous calibration of sensor errors. A common method for determining unknown errors is to augment the state vector to include the error. The quaternion extended kalman filter was proven not to be experimentally applicable to the rate gyroscope when using this method for more than the bias error alone. The observer itself shows promise, however, as it reaches a respectable estimate within seconds and continues to improve over time. The quaternion nonlinear observer, on the other hand, has very desirable properties, such as global asymptotic stability, but it is not ready for experimental application, where the most prized property is the ability to deal with unmodeled dynamics.

The design of the VT miniature AUV was an outstanding success, and the vehicle is currently in the final phase of development. Some future work is necessary before the vehicle can be considered fully operational, but it is at a stage where regular untethered tests are performed. The power system developed for the vehicle has completed the final design phase and is currently ready to be mass produced for multiple vehicles.

7.2 Future Work

7.2.1 Auto-calibration

The auto-calibration techniques discussed are only the initial pass at the broader application of a fully auto-calibrated vehicle. The ideal small, inexpensive AUV can be deployed without any manual calibration at all. Upon being turned on, the vehicle should start from the base model of an AUV and determine any and all alterations for its circumstances or configurations. Placing a new mission sensor on the vehicle, deploying it in a more dense body of water, or even something as drastic as broken component could be automatically determined and, where possible, accommodated for.

An integral component to full auto-calibration is the autonomous calibration of the navigation sensors, as confidence in the these sensors is needed before the vehicle model can be refined. The augmented state method has been proven not to work for real navigational sensors, but redundant sensors or alternative models may yet find use for the quaternion extended Kalman filter.

7.2.2 Miniature AUV platoon

Concurrent with research into full auto-calibration of a single vehicle is the development of five identical AUVs in order to research decentralized control techniques for application to missions. This will allow the ASCL to be the first to research platoons of three or more cooperating vehicles.

Bibliography

[1] Analog Devices, Inc. *ADXL311 Low Cost Ultracompact ±2g Dual-Axis Accelerometer.*

[2] Analog Devices, Inc. *ADXRS150 ±150°/s Single Chip Yaw Rate Gyro with Signal Conditioning.*

[3] R. Bachmayer and N.E. Leonard. Experimental test-bed for multi-vehicle control, navigation, and communication. In *12th International Symposium on Unmanned Untethered Submersible Technology*, 2001.

[4] B. Barshan and H.F. Durrant-Whyte. Evaluation of a solid-state gyroscope for robotics applications. *IEEE Transactions of Instrumentation and Measurement*, 44(1):61–67, Feb 1994.

[5] A.S. Gadre, J.J. Mach, D.J. Stilwell, and C.E. Wick. Design of a prototype miniature autonomous underwater vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 842–846, Oct 2003.

[6] B. Hobson, B. Schulz, J. Janet, M. Kemp, R. Moody, C. Pell, and H. Pinnix. Development of a micro autonomous underwater vehicle for complex 3-D sensing. In *OCEANS 2001 MTS/IEEE Conference and Exhibition*, volume 4, pages 2043–2045, Nov 2001.

[7] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[8] Kokam Engineering Co., LTD. *General Applications Manual for FMA/Kokam Lithium Polymer Batteries.*

[9] J.J. Mach. Toward auto-calibration of navigation sensors for miniature autonomous underwater vehicles. Master's thesis, Virginia Tech, 2003.

[10] M. Purcell, C. von Alt, B. Allen, T. Austin, N. Forrester, R. Goldsborough, and R. Stokey. New capabilities of the REMUS autonomous underwater vehicle. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 1, pages 147–151, Sep 2000.

[11] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley & Sons, Inc., 1989.

[12] D.J. Stilwell. Personal correspondence. Virginia Tech, 2004.

[13] D.J. Stilwell and C.E. Wick. A miniature low-cost autonomous underwater vehicle. In *OCEANS 2001 MTS/IEEE Conference and Exhibition*, volume 1, pages 423–428, Nov 2001.

[14] D.J. Stilwell, C.E. Wick, and Bishop B.E. Small inertial sensors for a miniature autonomous underwater vehicle. In *IEEE International Conference on Control Applications*, volume 1, pages 841–846, Sep 2001.

[15] True North Technologies LLC. *OEM Electronic Compass.* http://www.tntc.com/Products/OEM.htm.

[16] L.L. Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *IEEE International Conference on Robotics & Automation*, volume 1, pages 709–716, Apr 2000.

[17] J. Yuh. Underwater robotics. In *IEEE International Conference on Robotics & Automation*, volume 1, pages 932–937, Apr 2000.

Appendix A

Simulink Models

A.1 Rotation matrix vehicle model

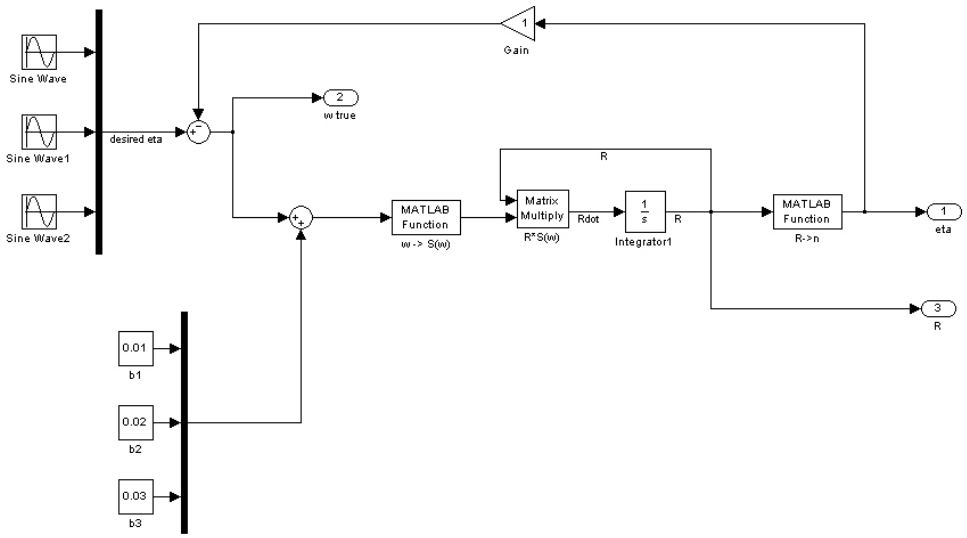

Figure A.1: Rotation matrix vehicle Simulink model

A.2 Euler angle extended Kalman filter

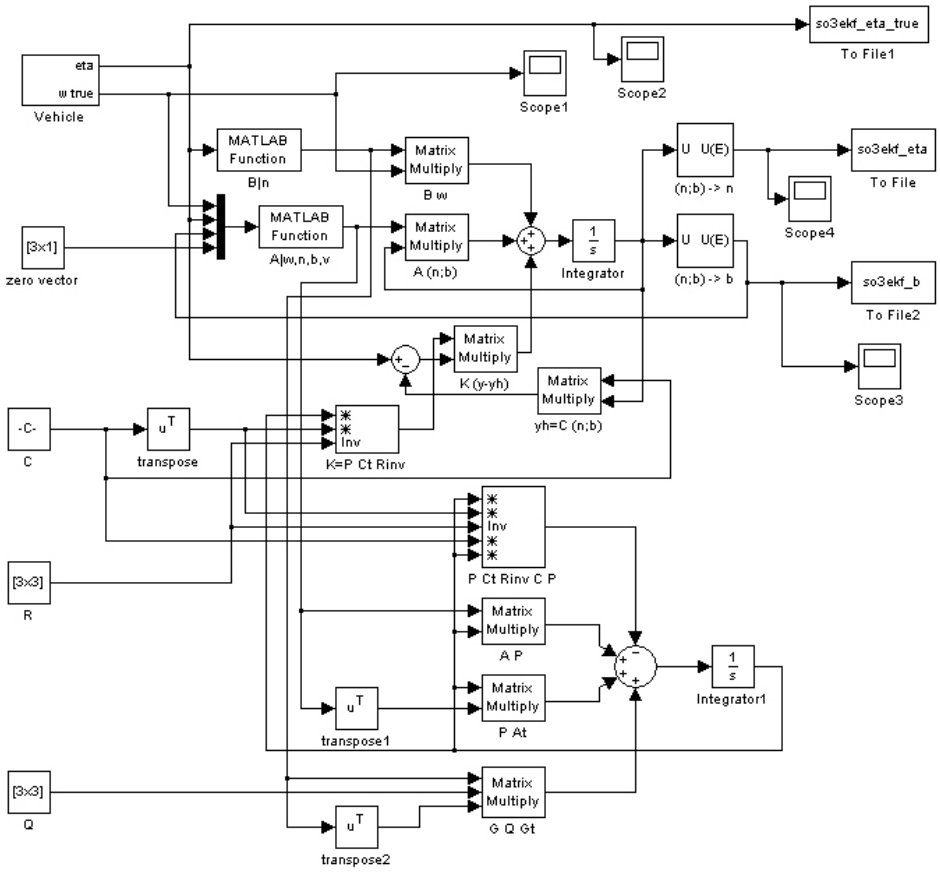

Figure A.2: Euler angle EKF Simulink model

A.3 Quaternion extended kalman filter

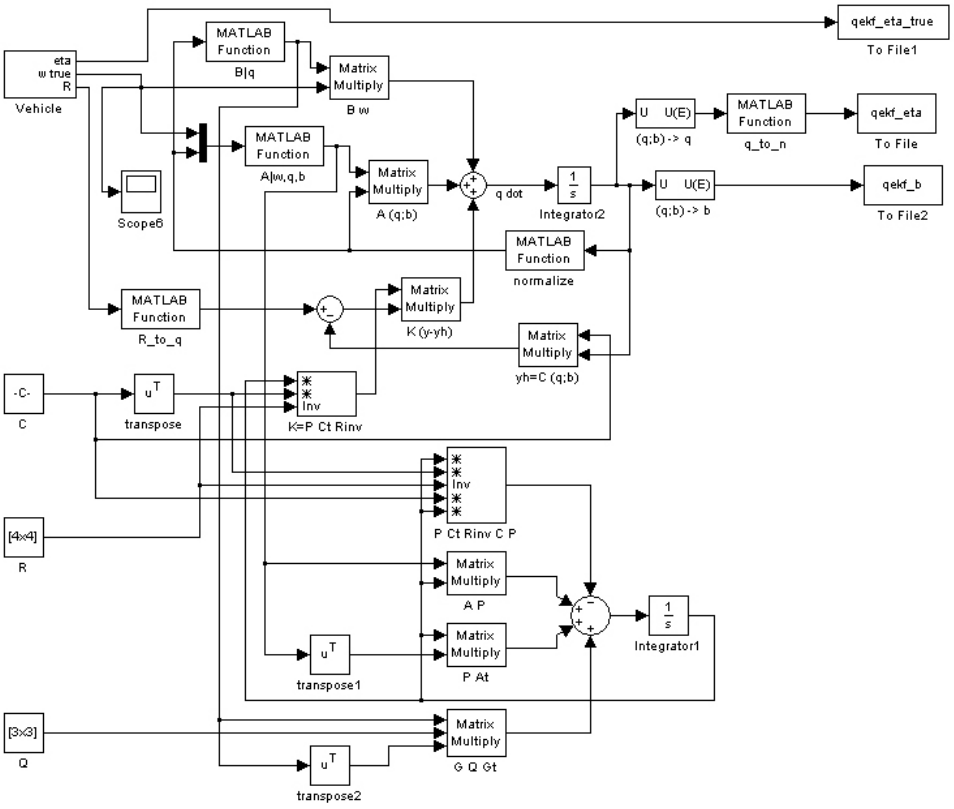

Figure A.3: Quaternion EKF Simulink model

A.4 Quaternion nonlinear observer

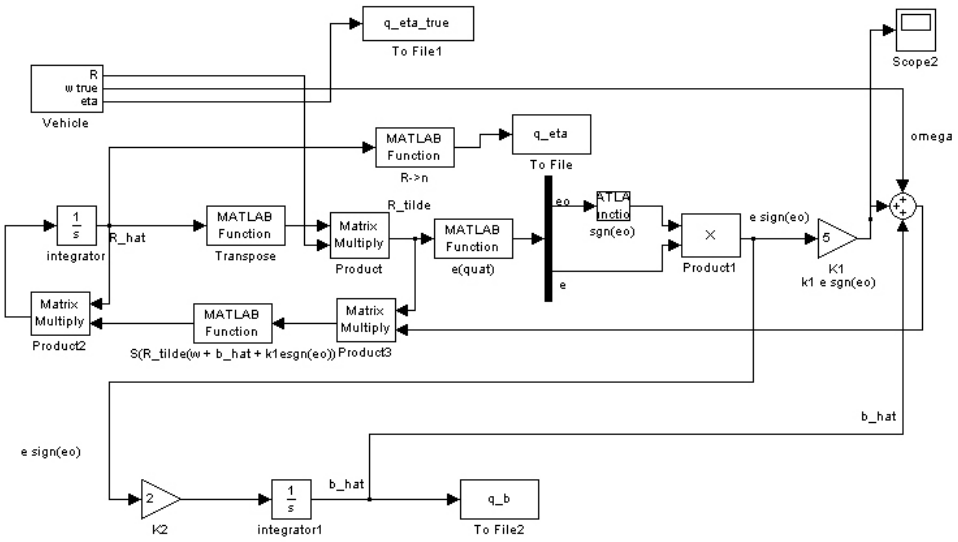

Figure A.4: Quaternion nonlinear observer Simulink model

A.5 Simulation trials

A.5.1 Input verification

Table A.1 details the inputs used for each verification trial. The initial conditions were all set to the values found in Table A.2.

Parameter	*Trial 6*	*Trial 7*	*Trial 9*	*Trial 10*
a_ϕ	1.0	0.1	1.0	0.45
f_ϕ	0.02	0.02	0.02	0.03
a_θ	0.5	0.1	0.1	0.82
f_θ	0.015	0.015	0.02	0.88
a_ψ	1.0	0.1	0.01	0.61
f_ψ	0.02	0.02	0.02	0.72

Table A.1: Parameters for Trials 6, 7, 9, 10

Parameter	*Trial 1*
b_x	0.01
b_y	0.01
b_z	0.01
ϕ_i	0°
θ_i	0°
ψ_i	0°
b_{xi}	0.01
b_{yi}	0.01
b_{zi}	0.01

Table A.2: Parameters for Trials 6, 7, 9, 10

A.5.2 Full trial data

Tables A.3 and A.4 give the full initialization data for all simulation trials. Table A.5 displays the results of each trial's root-sum-squared bias error convergence to various percentages of the constant zero estimate.

Parameter	*Trial 1*	*Trial 2*	*Trial 3*	*Trial 4*	*Trial 5*	*Trial 6*	*Trial 7*
a_ϕ	1.0	1.0	1.0	1.0	1.0	1.0	0.1
f_ϕ	0.02	0.02	0.02	0.02	0.02	0.02	0.02
a_θ	0.5	0.5	0.5	0.5	0.5	0.5	0.1
f_θ	0.015	0.015	0.015	0.015	0.015	0.015	0.015
a_ψ	1.0	1.0	1.0	1.0	1.0	1.0	0.1
f_ψ	0.02	0.02	0.02	0.02	0.02	0.02	0.02
b_x	0.01	0.001	0.0001	0.00001	1×10^{-12}	0.01	0.01
b_y	0.02	0.002	0.0002	0.00002	2×10^{-12}	0.01	0.01
b_z	0.03	0.003	0.0003	0.00003	3×10^{-12}	0.01	0.01
ϕ_i	0°	0°	0°	0°	0°	0°	0°
θ_i	0°	0°	0°	0°	0°	0°	0°
ψ_i	0°	0°	0°	0°	0°	0°	0°
b_{xi}	0.1	0.1	0.1	0.1	0.1	0.01	0.01
b_{yi}	0.2	0.2	0.2	0.2	0.2	0.01	0.01
b_{zi}	0.3	0.3	0.3	0.3	0.3	0.01	0.01

Table A.3: Parameters for Trials 1–7

Parameter	*Trial 8*	*Trial 9*	*Trial 10*	*Trial 11*	*Trial 12*	*Trial 13*	*Trial 14*
a_ϕ	1.0	1.0	0.45	1.0	1.0	1.0	1.0
f_ϕ	0.02	0.02	0.03	0.02	0.02	0.02	0.02
a_θ	0.5	0.1	0.82	0.5	0.5	0.5	0.5
f_θ	0.015	0.02	0.88	0.015	0.015	0.015	0.015
a_ψ	1.0	0.01	0.61	1.0	1.0	1.0	1.0
f_ψ	0.02	0.02	0.72	0.02	0.02	0.02	0.02
b_x	100	0.01	0.01	0.01	0.01	0.01	0.01
b_y	200	0.01	0.01	0.02	0.02	0.02	0.02
b_z	300	0.01	0.01	0.03	0.03	0.03	0.03
ϕ_i	0°	0°	0°	0°	15°	−80°	15°
θ_i	0°	0°	0°	0°	−5°	45°	−5°
ψ_i	0°	0°	0°	0°	1°	60°	1°
b_{xi}	0.01	0.01	0.01	0.01	0.01	0.01	0.1
b_{yi}	0.02	0.01	0.01	0.02	0.02	0.02	0.2
b_{zi}	0.03	0.01	0.01	0.03	0.03	0.03	0.3

Table A.4: Parameters for Trials 8–14

	Euler			*Quaternion*			*Nonlinear*		
Trial	*100%*	*10%*	*2%*	*100%*	*10%*	*2%*	*100%*	*10%*	*2%*
Trial 1	8.666	538.9	2070	6.439	416.7	2459	4.571	9.119	12.15
Trial 2	10.28	532.2	2068	6.418	416.7	2466	5.072	8.883	12.93
Trial 3	10.30	532.2	2066	6.412	416.3	2468	5.061	9.771	12.89
Trial 4	10.30	532.4	2067	6.412	416.2	2468	5.060	9.768	12.89
Trial 5	10.30	531.7	2067	6.412	416.2	2468	5.060	9.768	12.89
Trial 8	4272	>7200	>7200	23.45	414.8	2461	8.069	12.31	15.58
Trial 11	0	551.1	2074	0	416.2	2460	0	0	0
Trial 12	3.611	524.8	2065	0	415.8	2459	2.871	7.468	10.33
Trial 13	28.90	777.0	4497	5.990	417.1	2459	6.409	10.63	13.55
Trial 14	93.31	819.5	3349	22.97	414.2	2460	7.720	12.78	16.07

Table A.5: Bias estimation convergence in seconds for all performance trials

Appendix B

Matlab code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function A = Aq(u)
% linearize A matrix around an operating point
% for quaternion model
% u=[w;n;b;v]

wx=u(1);wy=u(2);wz=u(3);
v0=u(4);vx=u(5);vy=u(6);vz=u(7);
bx=u(8);by=u(9);bz=u(10);

A=[       0, -(wx+bx), -(wy+by), -(wz+bz), -vx, -vy, -vz;
    (bx+wx),        0,  (wz+bz), -(wy+by),  v0, -vz,  vy;
    (wy+by), -(wz+bz),        0,  (wx+bx),  vz,  v0, -vx;
    (wz+bz),  (wy+by),  -(wx+bx),       0, -vy,  vx,  v0;
    0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0]*(1/2);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function A = A_lin(u)
% linearize A matrix around an operating point
% for euler angle model
% u=[w;n;b;v]

wx=u(1);wy=u(2);wz=u(3);
x1=u(4);x2=u(5);x3=u(6);
bx=u(7);by=u(8);bz=u(9);
vx=u(10);vy=u(11);vz=u(12);

A=[ (by+wy+vy)*cos(x1)*tan(x2)-(bz+wz+vz)*sin(x1)*tan(x2),
     (bz+wz+vz)*cos(x1)*(sec(x2))^2+
      (by+wy+vy)*(sec(x2))^2*sin(x1),
     0, 1, sin(x1)*tan(x2), cos(x1)*tan(x2);
    -(bz+wz+vz)*cos(x1)-(by+wy+vy)*sin(x1),
     0, 0, 0, cos(x1), -sin(x1);
     (by+wy+vy)*cos(x1)*sec(x2)-(bz+wz+vz)*sec(x2)*sin(x1),
    (bz+wz+vz)*cos(x1)*sec(x2)*tan(x2)+
      (by+wy+vy)*sec(x2)*sin(x1)*tan(x2),
     0, 0, sec(x2)*sin(x1), cos(x1)*sec(x2);
    0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function B = Bq(u)
% linearize A matrix around an operating point
% for quaternion model

v0=u(1);vx=u(2);vy=u(3);vz=u(4);

B=[ -vx, -vy, -vz;
     v0, -vz,  vy;
     vz,  v0, -vx;
    -vy,  vx,  v0;
    0, 0, 0;
    0, 0, 0;
    0, 0, 0 ];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function B = B_lin(eta)
% linearize A matrix around an operating point
% for euler angle model

x1=eta(1);x2=eta(2);x3=eta(3);

B=[ 1, sin(x1)*tan(x2), cos(x1)*tan(x2);
    0, cos(x1), -sin(x1);
    0, sec(x2)*sin(x1), cos(x1)*sec(x2);
    0, 0, 0;
    0, 0, 0;
    0, 0, 0];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function qn = normalize_q(q)
% normalize a quaternion

qn=q/sqrt(q(1)^2+transpose(q(2:4))*q(2:4));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function qbn = normalize_qb(qb)
% normalize the quaternion q in [q;b]

qbn=[qb(1:4)/sqrt(qb(1)^2+transpose(qb(2:4))*qb(2:4));
    qb(5:7)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Q = Q(eta)
% create Q matrix for euler angle integration

x1=eta(1);x2=eta(2);x3=eta(3);

Q=[1,   sin(x1)*tan(x2),    cos(x1)*tan(x2);
    0,  cos(x1),            -sin(x1);
    0,  sin(x1)/cos(x2),    cos(x1)/cos(x2)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function R = q_to_R(q)
% convert quaternion to rotation matrix

v0=q(1);
v1=q(2);
v2=q(3);
v3=q(4);

R = [ v0^2+v1^2-v2^2-v3^2,
       2*(v1*v2-v0*v3), 2*(v1*v3+v0*v2);
      2*(v1*v2+v0*v3),
       v0^2-v1^2+v2^2-v3^2, 2*(v2*v3-v0*v1);
      2*(v1*v3-v0*v2),
       2*(v2*v3+v0*v1), v0^2-v1^2-v2^2+v3^2 ];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function T = q_to_T(q)
% create T matrix for quaternion integration

v0=q(1);
v=q(2:4);

T=[-transpose(v);S_omega(v)+eye(3,3)*v0];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function R = R(eta)
% create an R matrix from eta

x1=eta(1);x2=eta(2);x3=eta(3);

R=[cos(x2)*cos(x3),
    sin(x1)*sin(x2)*cos(x3)-cos(x1)*sin(x3),
    cos(x1)*sin(x2)*cos(x3)+sin(x1)*sin(x3);
   cos(x2)*sin(x3),
    sin(x1)*sin(x2)*sin(x3)+cos(x1)*cos(x3),
    cos(x1)*sin(x2)*sin(x3)-sin(x1)*cos(x3);
   -sin(x2),
    sin(x1)*cos(x2),
    cos(x1)*cos(x2)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function eta = R_eta(R)
% extract eta from a rotation matrix

eta(1,1)=atan2(R(3,2),R(3,3));
eta(2,1)=asin(-R(3,1));
eta(3,1)=atan2(R(2,1),R(1,1));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function q = R_to_q(R);
% convert a rotation matrix to a quaternion

a = (1/2)*sqrt( 1 + R(1,1) + R(2,2) + R(3,3));
b = (1/4/a)*(R(3,2) - R(2,3));
c = (1/4/a)*(R(1,3) - R(3,1));
d = (1/4/a)*(R(2,1) - R(1,2));

q = [a;b;c;d];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = skew(w)
% make a skew symmetric matrix out of vector w

S = [0, -w(3), w(2); w(3), 0, -w(1); -w(2), w(1), 0];
```

Appendix C

Proof of global asymptotic stability of quaternion nonlinear observer

The proof of global asymptotic stability for the quaternion nonlinear observer discussed in 6.3 was developed by D.J. Stilwell for a currently unpublished work [12], and is reproduced here for convenience.

A useful property of rotation matrices will be needed. Let rotation matrices $R_1, R_2 \in SO(3)$ satisfy

$$\dot{R}_1(t) = S(\omega_1(t))R_1(t)$$

and

$$\dot{R}_2(t) = S(\omega_2(t))R_2(t)$$

If $R_3 = R_2 R_1^T$ then R_3 satisfies

$$\dot{R}_3(t) = S(\omega_2(t) - R_3\omega_1(t))R_3(t) \tag{C.1}$$

Recall the quaternion nonlinear system (6.9)

$$\begin{aligned}\dot{\hat{R}}(t) &= S\Big(\widetilde{R}^T\Big(\omega + \hat{b} + k_1 e(t)\mathrm{sgn}(e_o(t))\Big)\Big)\hat{R} \\ \dot{\hat{b}}(t) &= k_2 e(t)\,\mathrm{sgn}(e_o(t))\end{aligned} \tag{C.2}$$

Using (C.1), the observer error $\widetilde{R} = R\hat{R}^T$ satisfies

$$\begin{aligned}\dot{\widetilde{R}}(t) &= S\Big(\tilde{b} - k_1 e(t)\,\mathrm{sgn}(e_o(t))\Big)\tilde{R}(t) \\ \dot{\tilde{b}}(t) &= -\frac{1}{2}k_2 e(t)\,\mathrm{sgn}(e_o(t))\end{aligned} \tag{C.3}$$

where $\tilde{b} = b - \hat{b}$ is bias estimate error. Expressing (C.3) using quaternion representation for $\widetilde{R}$ yields

$$\begin{aligned}\begin{bmatrix} \dot{e}_o(t) \\ \dot{e}(t) \end{bmatrix} &= \frac{1}{2}\begin{bmatrix} -e^T(t) \\ Ie_o(t) - S(e(t)) \end{bmatrix}\Big(\tilde{b} - k_1 e(t)\,\mathrm{sgn}(e_o(t))\Big) \\ \dot{\tilde{b}} &= -\frac{1}{2}k_2 e(t)\,\mathrm{sgn}(e_o(t))\end{aligned} \tag{C.4}$$

To compute equilibria for (C.4), note that since $[e_o, e^T]^T$ is a unit quaternion, $e_o^2 + e^T e = 1$. Thus (C.4) is written

$$\dot{e}_o(t) = -\frac{1}{2}e^T(t)\tilde{b}(t) + \frac{1}{2}k_1(1 - e_o^2(t))\,\mathrm{sgn}(e_o(t)) \tag{C.5}$$

$$\dot{e}(t) = \frac{1}{2}(e_o(t)I - S(e(t))(\tilde{b}(t) - k_1 e(t)\,\mathrm{sgn}(e_o(t))) \tag{C.6}$$

$$\dot{\tilde{b}}(t) = -\frac{1}{2}k_2 e(t)\,\mathrm{sgn}(e_o(t)) \tag{C.7}$$

Since $k_2 > 0$ and $\mathrm{sgn}(e_o) \neq 0$, (C.7) implies that $e(t) = 0$. Following similar arguments, the equilibria for the error dynamics are shown to be

$$\begin{bmatrix} e_o(t) \\ e(t) \\ \tilde{b} \end{bmatrix} = \begin{bmatrix} \pm 1 \\ 0 \\ 0 \end{bmatrix}$$

both of which correspond to the same rotation matrix $\widetilde{R} = I$. To ver-

ify asymptotic stability, define $\nu = [e_o, e^T, \tilde{b}^T]^T$ and consider the Lyapunov function candidate

$$V(\nu(t)) = \tilde{b}^T(t)\tilde{b}(t) + \begin{cases} k_2((e_o(t)-1)^2 + e^T(t)e(t)), & e_o(t) \geq 0 \\ k_2((e_o(t)+1)^2 + e^T(t)e(t)), & e_o(t) < 0 \end{cases}$$

Then using the fact that $e_o(t)\dot{e}_o(t) + e^T(t)\dot{e}(t) = 0$,

$$\dot{V}(\nu(t)) = 2\tilde{b}^T(t)\tilde{b}(t) + \begin{cases} -2\dot{e}_o(t), & e_o(t) \geq 0 \\ 2\dot{e}_o(t), & e_o(t) < 0 \end{cases}$$

and further simplifications yield

$$\dot{V}(\nu(t)) = -k_1 k_2 e^T(t)e(t) \tag{C.8}$$

Thus $\dot{V}(\nu(t)) \leq 0$ for all $\nu(t)$. Define $\Omega = \{\nu | \dot{V}(\nu(t)) = 0\}$. For asymptotic stability, it remains to apply LaSalle's Theorem and show that $\nu(t)$ can only remain in Ω for all future time if $\nu(t)$ is at an equilibrium. Note that $\nu(t) \in \Omega$ if and only if $e(t) = 0$, thus $\nu(t) \in \Omega$ implies that

$$\begin{aligned} \dot{e}_o(t) &= \frac{1}{2}k_1(1 - e_o^2(t))\,\mathrm{sgn}(e_o(t)) \\ \dot{e}(t) &= \frac{1}{2}e_o(t)\tilde{b}(t) \\ \dot{\tilde{b}} &= 0 \end{aligned}$$

If the error dynamics (C.4) are not at an equilibrium, then either $\dot{e}_o$ or $\dot{e}$ is nonzero. If $\dot{e}$ is nonzero, the proof is done. If $\dot{e}_o$ is nonzero, then $e_o^2 \neq 1$ and e cannot remain at zero since $e_o^2 + e^T e = 1$. Thus by LaSalle's Theorem, the error dynamics are asymptotically stable.

Appendix D

Power Distribution Board Parts List

Quantity	*Description*	*Part number*
1	5V Vicor DC-DC Converter	V24C5C100B
1	12V Vicor DC-DC Converter	V24C12C100B
1	5V Linear Regulator	LM340LAH-5.0-ND
1	5V 8A PC mount SPST Relay	DSP1a-DC5V
1	Fuse holder	WK0009-ND
1	3.15A Fuse	WK4324BK-ND
1	OpAmp	CLC430AJP-ND
1	Diode	1N4002
1	Magnetic Reed Switch, NO	374-1023-ND
1	Magnet	57025-000-ND
1	3MΩ resistor	
1	2.2MΩ resistor	
1	1MΩ resistor	
4	Terminal Block	
1	Receptacle Plug	

Table D.1: Parts list for power distribution board