Gain Scheduling of an Extended Kalman Filter for Use in an Attitude/Heading Estimation System

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Donald Patrick Horkheimer

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Master of Science

Under the supervision of Dr. Demoz Gebre-Egziabher

March, 2012

Gain Scheduling of an Extended Kalman Filter
for Use in an
Attitude/Heading Estimation System

by Donald Patrick Horkheimer

ABSTRACT

Even with recent advances in computing power the development of smaller Unmanned Aerial Vehicles (UAVs) and sophisticated sensor payloads with high data rates can still challenge on-board computer resources. In response to this challenge, gain scheduling is investigated as a means to reduce the computational burden associated with a nonlinear attitude estimator. The attitude/heading filter used to validate the gain scheduling approach is based on an Euler angle parameterization. Its process dynamics and measurement updates are provided by nonlinear rate kinematic equations and absolute attitude measurement updates, respectively. The gain scheduling approach is intended to be instrumentation independent for the attitude parameterization used. Validation of the gain scheduling attitude/heading estimation filter utilized process dynamics driven by a low-cost Micro-Electromechanical System (MEMS) based Inertial Measurement Unit (IMU). Measurement updates are provided by an external machine-vision infrared tracking system. The gain scheduling approach should be applicable to other sensor types such as GPS, magnetometers, and other aides. Gain scheduling filter development has been tested using simulated trajectories and real data collected from a remote control helicopter flown indoors and processed off-line.

Acknowledgements

I would like to thank Jon Andersh, Navid Dadkhah, and Sukhada Palav for providing me with a variety of raw data sets that allowed me to test the attitude/heading estimation filter. Thanks also goes to Dr. Bérénice Mettler for providing contemporary practical motivation for using gain scheduling and access to her lab. Finally, I would like to thank my adviser Dr. Demoz Gebre-Egziabher for his advice on Kalman Filtering throughout my research and for being flexible with the schedule of a full-time professional and a part-time engineering student.

Dedication

To my mother and father for stressing the importance of an education.

Contents

List of Tables

List of Figures

Nomenclature

η_{SFx} Independent zero-mean Gaussian white noise process for X-axis gyro sensitivity

η_{SFy} Independent zero-mean Gaussian white noise process for Y-axis gyro sensitivity

η_{SFz} Independent zero-mean Gaussian white noise process for Z-axis gyro sensitivity

η_x X-axis independent zero-mean Gaussian white noise process

η_y Y-axis independent zero-mean Gaussian white noise process

η_z Z-axis independent zero-mean Gaussian white noise process

$\hat{\Phi}$ Estimated bank/roll angle

$\hat{\Psi}$ Estimated heading/yaw angle

$\hat{\Theta}$ Estimated elevation/pitch angle

$\hat{b}_x$ Estimate of X-axis gyro sensor bias

$\hat{b}_y$ Estimate of Y-axis gyro sensor bias

$\hat{b}_z$ Estimate of X-axis gyro sensor bias

$\hat{SF}_x$ Estimate of X-axis gyro sensitivity error

$\hat{SF}_y$ Estimate of Y-axis gyro sensitivity error

$\hat{SF}_z$ Estimate of Z-axis gyro sensitivity error

$\mathbf{\Phi}$ Discrete state transition matrix, (n×n)

$\mathbf{F}$	Continuous state transition matrix, (n×n)
$\mathbf{f}(\hat{x}(t),(t))$	Nonlinear process differential equation
$\mathbf{G}$	Process/dynamic system noise sensitivity matrix, (n×s)
$\mathbf{H}$	Measurement sensitivity matrix, (r×n)
$\mathbf{h}(\hat{x}(t),(t))$	Nonlinear measurement differential equation
$\mathbf{J}$	Jacobian matrix
$\mathbf{L}$	Kalman gain matrix, (n×r)
$\mathbf{P}$	State estimation error covariance matrix, (n×n)
$\mathbf{Q}$	Process noise covariance or spectral density matrix, (s×s)
$\mathbf{R}$	Discrete measurement noise covariance matrix, (r×r)
$\mathbf{V}$	Continuous measurement noise covariance matrix, (r×r)
$\mathbf{W}$	Continuous process noise matrix or vector
$\mathcal{N}$	Zero-mean Gaussian random variate
μ	Performance index
ν_x	X-axis independent zero-mean Gaussian white noise process for Gauss-Markov model
ν_y	Y-axis independent zero-mean Gaussian white noise process for Gauss-Markov model
ν_z	Z-axis independent zero-mean Gaussian white noise process for Gauss-Markov model
Φ	Bank/roll angle
Ψ	Heading/yaw angle
τ_{IMU}	Time constant of Gauss-Markov noise process

Θ	Elevation/pitch angle
$\underline{\beta}$	Discrete process noise matrix or vector
$\underline{\dot{x}}$	Derivative of system state vector with respect to t, (n×1)
$\underline{\eta}$	Vector of independent zero-mean Gaussian white noise process
$\underline{\hat{x}}$	Estimate of system state vector, (n×1)
$\underline{\hat{z}}$	Noise corrupted measurement vector, (r×1)
$\underline{\nu}$	Vector of independent zero-mean Gaussian white noise process for Gauss-Markov model
$\underline{\Omega}_{IMU}$	Vector of corrupted vehicle body angular rates
$\underline{\Omega}_{true}$	Vector of uncorrupted vehicle body angular rates
$\underline{b}$	Vector of gyro sensor bias
$\underline{x}$	System state vector, (n×1)
$\underline{z}$	Noise-free measurement vector, (r×1)
a_n	Regression coefficient, $n = 1, 2, 3, ...7$
J_t^s	Trace of the error covariance of the suboptimal filter
J_t	Trace of the error covariance of the optimal filter
k	Discrete time
n	Integer number of state variables
p	Vehicle body angular rate of roll
q	Vehicle body angular rate of pitch
r	Integer number of measurements/observations
r	Vehicle body angular rate of yaw

s Integer number of process disturbances

t Continuous time

Note: wherever possible matrices are denoted by **BOLD** capital letters and vectors are denoted by underlined lower case letters.

Chapter 1

Introduction

Attitude/heading estimation systems, sometimes known as Attitude/Heading Reference Systems (AHRS), are multi-sensor systems that are used to calculate the orientation of a vehicle or other physical system with respect to space. Multi-sensor systems are favored over single sensor systems as they tend to be cheaper, lighter-weight, and more robust. These multi-sensor systems are typically built around two or more types of sensors. Past multi-sensor attitude/heading estimation systems have been demonstrated with combinations of rate gyros, multi-antenna GPS, magnetometers, and accelerometers [1].

The output of each individual sensor in a generic attitude/heading estimation system is combined in an estimation algorithm, typically an Extended Kalman Filter (EKF), and the output of the algorithm is a description of the orientation of the vehicle. The attitude/heading estimation algorithms play an important role in the overall system. Their careful implementation can allow the system designer more freedom in sensor selection, meaning improvements in cost, weight, and robustness for the end user. However, if the algorithm is complicated it may require computational resources that are not available on the embedded system used to implement the filter.

The exponential growth in computing power is often seen as the solution for problems encountered when implementing relatively complex engineering algorithms in real-time. If the embedded computing hardware does not presently offer enough performance to do a successful implementation, then it seems the solution is to wait for the next generation of hardware. Gain scheduling is another means in which the computational burden of an EKF can be dealt with.

A more challenging problem with complicated algorithms (which cannot always be solved by powerful computers) is verification and validation of the algorithms. That is, will complicated algorithms always perform as intended. Can one certify that the output errors are always going to be less than some threshold? In attitude determination algorithms which are based on an EKF, one of the sources of complexity is the Kalman gain calculation. Some of complexity can be removed if these gains are computed ahead of time and saved in a look up table or defined by the output of a simple function. This is, in a loose sense, gain scheduling. This thesis examines gain scheduling of an attitude/heading estimating EKF which uses a multi-sensor system built around rate gyro and optical measurements.

A new potential application area for these gain schedule filters is in the navigation and control of small, unmanned vehicles [2], such as what the Defense Advanced Research Projects Agency (DARPA) is calling Nano Air Vehicle (NAV) and Micro Air Vehicle (MAV) range of aircraft [3]. With these applications in mind, a gain scheduled attitude/heading estimating EKF was developed and tested using both simulated attitude/heading time history trajectories and real measurements recorded from the flight of a remote control helicopter. Helicopter flight data recordings used to test the algorithms included both angular body rate measurements from a Microelectromechanical System (MEMS) Inertial Measurement Unit (IMU) and absolute attitude/heading measurements using an infrared machine vision tracking system. The approach taken in validating the gain scheduled EKF could be applied to other instrumentation configurations and other EKF problems.

1.1 Thesis Problem Statement

This thesis examines one method of gain scheduling an attitude/heading-estimating Extended Kalman Filter (EKF). This type of filter is an efficient estimator that is derived for use with nonlinear systems and historically have found a great deal of application in attitude/heading estimating systems. Gain scheduling simplifies the attitude/heading estimating algorithm in order to allow its use on-board a small, remote control helicopter.

Briefly, gain scheduling goes beyond the concept of a conventional, suboptimal EKF. The conventional EKF is commonly thought of as a suboptimal filter, because the filter uses a linearization of the system's true nonlinear dynamic process model. The linearized dynamic equations are then used to calculate a suboptimal EKF gain sequence in the conventional implementation. The gains are then used to sub-optimally weigh a combination of process model predictions and measurements to come up with the best suboptimal estimate of the system's state variables. Additional theory and definitions of terminology are presented later.

Gain scheduling eliminates the step of calculating the gain sequence in real-time and instead through off-line analysis and simulation reduces their real-time calculation to look-up tables, constants, or simple functions where the independent variable(s) is often a state variable and the dependent variable a Kalman gain value. Gain scheduling, therefore, reduces the computational burden and complexity, at the cost of greater state variable estimate uncertainty.

Suboptimal filtering should not be burdened with a negative connotation as all real and implemented EKFs contain certain model simplifications. It is just that gain scheduling goes further and simplifies the calculation of the gains. For some applications, the trade-off between cost and performance is likely worth it.

1.2 Literature Review on Gain Scheduling Kalman Filters

A literature survey was conducted of past work in the field of gain scheduling both linear and extended (nonlinear) Kalman Filters. In the late 1960's, the early days of Kalman filtering and digital computers, most works emphasized optimizing a fixed number of gains and their value and/or schedule. The finer the gain schedule approximation, the better the results in approximating the Kalman gain that would normally be calculated on-line in the conventional algorithm. That is assuming the scheduling variable is an appropriate match to the system.

One of the earliest and most practical examples of gain scheduling a 15-state linear Kalman Filter comes from work done at the Sperry Corporation in the late 1960's and published in the Sperry Engineering Review [4]. The article is based on work the company did for creating a hybrid LORAN/Inertial Navigation System or what the authors called a LINS system. LORAN was an early, ground-based position fixing radio navigation system similar in nature to modern GPS. The system has recently been decommissioned.

The article also has interesting commentary on the benefits of integrating radio navigation position fixing systems with inertial navigation systems. These are very familiar to many modern developers of hybrid GPS/INS systems. For example, the authors discuss how inertial aiding in the system can be used to tighten LORAN tracking loops to improve jamming performance and noise rejection. At a system level the accuracy of the navigation system is dominated by the accuracy of LORAN and that, consequently, less expensive and lower performing inertial sensors can be used and still achieve similar navigation performance relative to a much more expensive stand alone INS.

Kleinman et al. [5, 6] addressed the mathematical concept of how to schedule gains, but in this case for the linear regulator problem rather than a state estimation problem. This more theoretical study looked at gain scheduling by representing the gains as piecewise constant gains in time. The number, magnitude, and duration of the constant gain levels were determined by minimizing a quadratic cost function which represented a trade-off between mathematical optimality and engineering usefulness. The authors were able to demonstrate that under suitable assumptions the optimal control gains could be approximated by the suboptimal control arbitrarily closely.

Crotteau [7] upon observing that the Kalman gain histories generally followed an exponential time history demonstrated a different approach, arriving at a closed form expression for gain scheduling. The exponential response of the Kalman gain was linearized then fit with a line using the method of least squares. The fitted function was then used to allocate gains versus time. An error covariance update equation intended for use with suboptimal filters was presented. Another work used a similar equation when presenting a brief overview of the implications of assuming constant gains in a Kalman Filter [8].

More recently, gain scheduled EKFs have found use in predicting in-flight performance parameters of jet aircraft engines [9] as well as estimating the air consumption of turbocharged spark ignition engines in real-time [10]. Modern literature has referred to these filters as Constant Gain Extended Kalman Filters (CGEKF). The gains used in these EKF's are not constant over all operating conditions, but are only constant over small intervals. For example, in the case of the engine the constant gain values would populate a matrix or look-up table. The look-up table may be indexed by engine throttle position and RPM and different gains are assigned depending on where the engine is operating in the table.

In the area of attitude and heading estimation Yoo et al. [11] developed a gain scheduled complementary filter for use in vehicles. The AHRS system was MEMS based and utilized accelerometers, gyros, and magnetometers as sensors. The complementary filter sensor fusion algorithm uses frequency domain filtering techniques to blend the various sensor outputs in a way that can minimize the errors specific to individual types of sensors. The gains of the complementary filter were switched based on measurements made by the accelerometers. This approach gave improved performance over the fixed gain complementary filter without much additional complexity.

1.2.1 Past Work on Comparing Performance of Gain Scheduled Filters

Methods for comparing performance differences between optimal and suboptimal filters are important when assessing the performance of gain scheduled filters. One method given in [12] defines the performance index, μ, by evaluating the trace of the error covariances at time, t, of the optimal filter, $\mathbf{P}_t$, and suboptimal filter, $\mathbf{P}_t^s$, where s, denotes suboptimal. The performance index, μ, is defined in Equation 1.1.

$$\begin{aligned} J_t &= \text{Trace}[\mathbf{P}_t] \\ J_t^s &= \text{Trace}[\mathbf{P}_t^s] \\ \mu &= \frac{J_t^s - J_t}{J_t} \end{aligned} \tag{1.1}$$

When μ is equal to zero the suboptimal filter is actually optimal. And values greater than zero indicate to what degree the filter is suboptimal. Another approach is to simply take the square root of diagonal elements, on a per element basis, of the error covariance matrix to determine the root-mean-square value of the uncertainty of estimation error associated with individual elements and compared their magnitudes between filter designs [13].

The performance index of Equation 1.1 is utilized in this thesis to judge performance differences between gain scheduled and non-gain scheduled EKF implementations. It was felt that this was an efficient solution to the problem of judging filter performance. Both approaches provide a means to estimate filter performance without requiring access to an absolute truth reference.

1.3 Thesis Outline

Accordingly, the remainder of this thesis is organized as follows:

- Chapter 2 provides an overview of the generic continuous-time EKF. This chapter also provides background details on gain scheduling the EKF.

- Chapter 3 introduces the simplest discrete-time implementation of the attitude/heading estimating EKF with gyro bias error states. This is the filter evaluated in this work.

- Chapter 4 presents results of using the attitude/heading estimating EKF with and without gain scheduling on simulated trajectories and experimental data. The experimental results are from data collected with an instrumented Remote Control (RC) helicopter flown inside the Interactive Guidance and Control Lab (IGCL). From these results it became clear that incorporating additional EKF error states would likely improve the estimated attitude/heading solution. Thus a way to incorporate additional error states to improve the attitude/heading estimation accuracy of the EKF when processing real data is also presented.

- Chapter 5 presents concluding remarks of the work presented in the thesis.

In addition, the appendices provide supplementary material. Appendix A provides a glossary of terms. Appendix B provides the MATLAB code of the EKF used in this study. Appendix C lists the Kalman gain schedules for the various trajectories analyzed as part of this work.

Chapter 2

Extended Kalman Filtering and Gain Scheduling

The purpose of this chapter is to present the theory for a generic continuous EKF and background on how the EKF can be gain scheduled. Some details on gain scheduling the EKF specific to the problem of attitude and heading determination is also presented in order to illustrate the motivation for using gain scheduling given its computational advantages. The following chapter presents the attitude/heading estimator formulated as a discrete EKF. The basics of Kalman Filtering can be found in [14], [15], [16], [17], and [18]. Some basic terms and acronyms are introduced in Appendix A.

The basic Kalman Filter is a minimum variance, unbiased estimator for linear systems. The EKF is a technique that uses the basic Kalman filter on nonlinear systems with either nonlinear process dynamics or nonlinear measurements. This is accomplished by using the latest state estimate as the operating point for linearizing the nonlinear equations A generic continuous-time nonlinear system or dynamic model is defined by Equation 2.1. The formulation of the EKF presented in this thesis does not include deterministic control inputs.

$$\begin{aligned} \dot{\underline{x}}(t) &= \mathbf{f}(x(t),(t)) + \mathbf{W}(t) \\ \mathbf{W}(t) &\sim \mathcal{N}(0, \mathbf{Q}(t)) \end{aligned} \tag{2.1}$$

The true state variables are represented by $x(t)$ and the variables are corrupted by a Gaussian white noise process given by $\mathbf{W}(t)$. The expression $\mathcal{N}(0, \mathbf{Q}(t))$ refers to the

Gaussian random variate, $\mathcal{N}$, with a zero-mean value and a covariance defined by $\mathbf{Q}(t)$.

Similarly, a nonlinear measurement model can be defined as shown in Equation 2.2.

$$\begin{aligned} \underline{z}(t) &= \mathbf{h}(x(t),(t)) + \mathbf{V}(t) \\ \mathbf{V}(t) &\sim \mathcal{N}(0,\mathbf{R}(t)) \end{aligned} \tag{2.2}$$

The measurement variables are represented by $x(t)$ and the variables are corrupted by a Gaussian white noise process given by $\mathbf{V}(t)$. $\mathbf{R}(t)$ is the measurement noise covariance which is modeled as a stochastic process.

Next, the EKF algorithm linearizes the nonlinear dynamic and measurement models, defined by $\mathbf{F}(t)$ and $\mathbf{H}(t)$ respectively, by calculating their Jacobian's at their present respective operating condition, or in other words their present state estimate, $\hat{x}(t)$, and thereby calculating a linear approximate of the nonlinear functions as illustrated in Equation 2.3.

$$\begin{aligned} \mathbf{F}(t) &\approx \left.\frac{\partial \mathbf{f}(x,t)}{\partial x}\right|_{x=\hat{x}(t)} \\ \mathbf{H}(t) &\approx \left.\frac{\partial \mathbf{h}(x,t)}{\partial x}\right|_{x=\hat{x}(t)} \end{aligned} \tag{2.3}$$

Next, the Kalman gain, $\mathbf{L}(t)$, is calculated as shown in Equation 2.4. The Kalman gain represents the weight placed on the measurement relative to the process dynamics prediction in the state estimate update stage. Accurate, low-noise measurements result in a greater weighting of the measurement when updating the filter state estimates. Noisy measurements result in more weight being placed on the dynamic model prediction of the future state estimates.

$$\mathbf{L}(t) = \mathbf{P}(t)\mathbf{H}^T(t)\mathbf{R}^{-1}(t) \tag{2.4}$$

Now, the state estimation error covariance matrix update, $\mathbf{P}(t)$, can occur as shown in Equation 2.5.

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) - \mathbf{L}(t)\mathbf{R}(t)\mathbf{L}^T(t) \tag{2.5}$$

Finally, the state estimate update can be defined as shown in Equation 2.6.

$$\underline{\hat{\dot{x}}}(t) \quad = \quad \mathbf{f}(\hat{x}(t),(t)) + \mathbf{L}(t)[z(t) - \hat{z}(t)] \tag{2.6}$$

Where $\hat{z}(t)$ is defined as the predicted noisy measurement as given in Equation 2.7.

$$\hat{z}(t) \quad = \quad \mathbf{h}(\hat{x}(t),(t)) \tag{2.7}$$

2.1 Gain Scheduling as Applied to the EKF

The motivation for gain scheduling the EKF is to reduce the computational burden associated with calculating the Kalman gain on every iteration. Gain scheduling replaces Equation 2.4, the conventional Kalman gain, with simpler forms. The form could be a constant value, a look-up table indexed by an auxiliary variable, or an evaluation of a simple function relating the auxiliary variable to the filter gain. Some researchers call the auxiliary variable the scheduling variable as the scheduling of gains are dependent on this variable.

2.1.1 Gain Scheduling Approaches

This work evaluates three gain scheduling approaches:

1. The first approach is a constant gain computed from the time-series average gain as predicted by the filter equations for the calculation of optimal gains. After disregarding an initial gain transient on start-up. Here, a scheduling variable is not used.

2. The second approach is a gain function based on a multi-nonlinear regression of the Kalman gain versus Euler angles. Each Kalman gain parameter is fitted with the same gain function. The regression is performed on specific optimal Kalman gain trajectories. In this approach the scheduling variables are individual Euler angles. Once again the initial gain transient is disregarded.

3. The third approach is a tailored gain function for each Kalman gain parameter based on a multi-nonlinear regression of the Kalman gain versus Euler angle

for a series of trajectories. The series of trajectories are characterized by ±45° magnitude sinusoidal oscillations 90° out of phase in pitch and yaw and constant magnitude roll angle for each trajectory. The magnitude of the roll angles tested are from -45° to 45°. The gain functions have anywhere from 2 to 6 terms in their regression equation. Again, in this approach the scheduling variables are individual Euler angles. Once again the initial gain transient is disregarded.

Computing Equation 2.4 on a digital computer requires mathematical operations of multiplication and addition. Using notations introduced in the Nomenclature section, the number of multiplication operations are $n^2r + 2nr^2 + r^3$ and the number of addition operations are $n^2r + 2nr^2 - 2nr + r^3$. These computational requirements were calculated by Mendel [19]. Gain scheduling can reduce the computational burden of the operations associated with Equation 2.4.

If steady-state filter values were used for the Kalman gains, the gains would be equivalent to the discrete time Wiener filter which is suboptimal until the filter achieves steady-state operation. But, in this application the gains did not appear to reach steady-state. This is believed to be due to the optimal gains being dependent on the attitude/heading of the helicopter and the nonlinear nature of the system dynamics.

The distinction between optimal and suboptimal filtering ties into gain scheduling in the following way. When Equation 2.4 is used to solve for the Kalman gains in the EKF, this is considered the optimal filter implementation. The implementation is only optimal for propagation of second order statistics associated with the system. The conventional EKF is suboptimal in the propagation of system statistics greater then second order. The EKF represents an algorithm to bridge the gap between the linear model assumptions behind the basic linear optimal Kalman Filter equations and the suboptimal linearization of the nonlinear model equations of the EKF. In other words, at its best, even without gain scheduling, the EKF is a suboptimal algorithm because of the higher order non-linearities it operates on in practice. It is perhaps better to state that the EKF is locally optimal given that the EKF linearization of the nonlinear equations is accurate. The simple linear Kalman filter is the only strict optimal estimator.

In this thesis the gain scheduled filter is suboptimal, it does not mean outside this thesis that the converse is true: that all suboptimal filters are gain scheduled filters. In actuality, there are many different flavors of suboptimal filters. Sometimes a filter is suboptimal due to the use of intentional, reduced order models, unknown bias errors in the process or measurement dynamics, or other implementation short-cuts or errors.

Chapter 3

A Multi-sensor For Attitude/Heading Estimation

Using the generic continuous EKF theory presented previously, a specific discrete filter was developed for the application of providing attitude/heading estimates for small, unmanned vehicles and that would also serve the purpose of validating the gain scheduling approach. The basic filter implemented for this research into gain scheduling is based on prior work [20, 21, 22].

One point of terminology here that can lead to confusion when discussing this specific implementation of an EKF is the difference between a system dynamics model and a measurement model. In a loose sense, both the IMU and the Vicon vision tracking system are making measurements that are used by the filter. In the specific terminology of Kalman filtering in this instance, the IMU is driving the process model and the VICON system is the measurement model or, perhaps in a less confusing approach, what some would call the observation model for the EKF. The gain scheduling approach taken in this thesis is not tied to the instrumentation selected to validate the algorithms. Other configurations are possible.

The instrumentation required for the EKF implementation presented could be done with generic, angular rate sensors and a generic, absolute attitude/heading sensors. But to tie in with the empirical testing performed in this thesis, it is suggested that the filter only utilizes an IMU and Vicon vision tracking system. Test cases utilizing simulated data are with generic sensors.

A high-level overview of the system used to validate the gain scheduled EKF is shown in Figure 3.1. More details on each component of the system is presented in the following chapter. This section provides more details on the filter implemented in MATLAB.

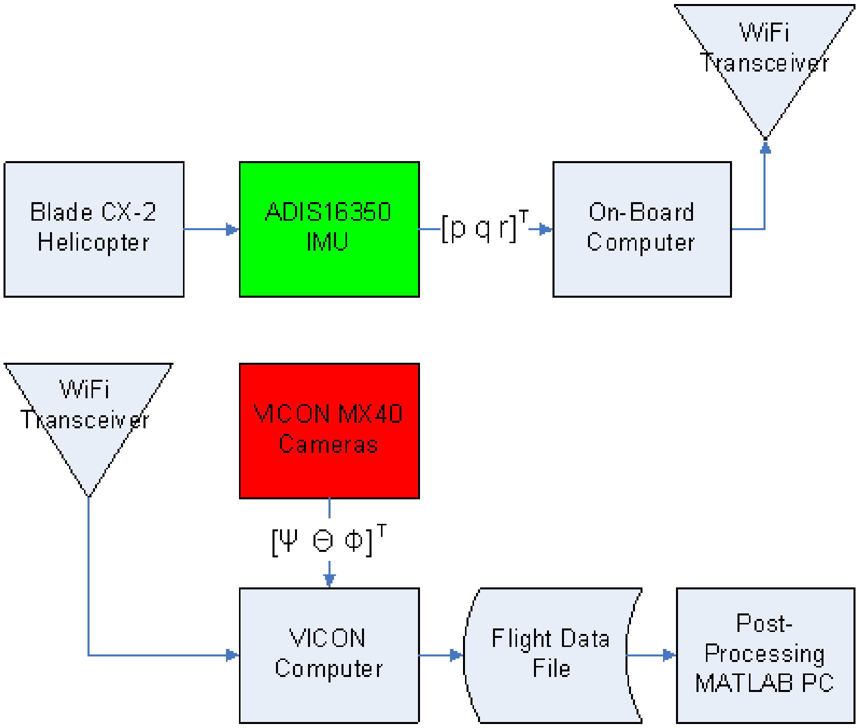

Figure 3.1: Block Diagram of Test Setup Used to Validate Specific EKF Implementation

The filter utilizes the Euler angle kinematic rate equations as the basis for its non-linear dynamics model as shown in Equation 3.1. The equation is driven by the IMU measurements.

$$\begin{bmatrix} \dot{\Psi} \\ \dot{\Theta} \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sin\Phi}{\cos\Theta} & \frac{\cos\Phi}{\cos\Theta} \\ 0 & \cos\Phi & -\sin\Phi \\ 1 & \frac{\sin\Phi\sin\Theta}{\cos\Theta} & \frac{\cos\Phi\sin\Theta}{\cos\Theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.1}$$

The parameterization of vehicle attitude/heading in terms of Euler angles typically presents a problem with the known mathematical singularity that occurs when the attitude of the vehicle reaches a pitch angle of $\Theta = 90^\circ$. This issue was neglected in favor of the simplicity of using Euler angles over quaternions or Direction Cosine Matrices (DCM), and because the vehicular application envisioned for this filter would not typically see pitch maneuvers this large.

The p, q, r terms in the equation covered in Equation 3.1 represent the vehicle body axis angular rates as measured by the gyros within the IMU, and these drive the process model for the EKF. In the application of the filter, they are the raw gyro angular rate measurements. The parameters Ψ, Θ, Φ are the Euler angles, sometimes referred to as the Tait-Bryan angles, of absolute attitude/heading in the standard 3-2-1 sequence. As rotations are non-commutative, the sequence of rotation matters. The parameters Ψ, Θ, Φ, can be thought of as the yaw, pitch, and roll of the vehicle, respectively.

For the purposes of the filter, the process update can be thought of as a vector containing the IMU gyro measurements. The update has the following process error model:

$$\begin{aligned} \underline{\Omega}_{IMU} &= \underline{\Omega}_{true} + \underline{b} + \underline{\eta} \\ \dot{\underline{b}} &= -\frac{1}{\tau_{IMU}}\underline{b} + \underline{\nu} \end{aligned} \tag{3.2}$$

The variables in Equation 3.2 represent the following terms: $\underline{\Omega}_{IMU}$ is the vector of the imperfect measurements of body angular rates, otherwise known as p, q, r. In this application these are the measurements provided by the MEMS IMU. The IMU contains three independent, orthogonal gyro sensors, this is a three element vector. $\underline{\Omega}_{true}$ is the uncorrupted or perfect measurement of body angular rates. $\underline{b}$ is the vector representing the gyro sensor bias errors, sometimes this type of error is called the gyro's null-shift error, in other words, the deterministic rate indicated by the gyro when it is not rotating. $\underline{\eta}$ and $\underline{\nu}$ are vectors of independent zero-mean Gaussian white-noise processes. The second equality in Equation 3.2 represents a model of the gyro bias as a first order Gauss-Markov process, where τ_{IMU} is the correlation time constant of the model.

Next, in developing the process model for the nonlinear dynamics, processed by the EKF, it is necessary to linearize about the state vector estimate, $\underline{\hat{x}}$. In this case this is the filter estimated attitude/heading of the helicopter. The result is a linear differential state equation in Equation 3.3:

$$\dot{\underline{x}}(t) \quad = \quad \mathbf{F}(t)\hat{\underline{x}}(t) + \mathbf{G}(t)\underline{\beta}(t) \tag{3.3}$$

The vector $\hat{\underline{x}}$ is the state vector containing the following variables in Equation 3.4. The $\hat{b}$ terms are the respective gyro bias estimate terms.

$$\hat{\underline{x}} = \begin{bmatrix} \hat{\Psi} & \hat{\Theta} & \hat{\Phi} & \hat{b}_x & \hat{b}_y & \hat{b}_z \end{bmatrix}^T \tag{3.4}$$

The matrix $\mathbf{F}$ represents the dynamic coefficient matrix, also known as the continuous state transition matrix of the helicopter and contains the following variables given in Equation 3.5:

$$\mathbf{F} = \begin{bmatrix} 0 & \mathbf{F}(1,2) & \frac{q\cos(\Phi)-r\sin(\Phi)}{cos(\Theta)} & 0 & \frac{\sin(\Phi)}{\cos(\Theta)} & \frac{\cos(\Phi)}{\cos(\Theta)} \\ 0 & 0 & -q\cos(\Phi)+r\sin(\Phi) & 0 & \cos(\Phi) & -\sin(\Phi) \\ 0 & \frac{q\cos(\Phi)-r\sin(\Phi)}{cos^2(\Theta)} & \mathbf{F}(3,3) & 1 & \frac{\sin(\Phi)\sin(\Theta)}{\cos(\Theta)} & \frac{\cos(\Phi)\sin(\Theta)}{\cos(\Theta)} \\ 0 & 0 & 0 & \frac{-1}{\tau} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau} \end{bmatrix} \tag{3.5}$$

Where $\mathbf{F}$(1,2) is defined by Equation 3.6:

$$\mathbf{F}(1,2) \quad = \quad -q\sin(\Phi)\frac{\sin(\Theta)}{\cos^2(\Theta)} - r\cos(\Phi)\frac{\sin(\Theta)}{\cos^2(\Theta)} \tag{3.6}$$

And $\mathbf{F}$(3,3) is defined by Equation 3.7:

$$\mathbf{F}(3,3) \quad = \quad q\cos(\Phi)\frac{\sin(\Theta)}{\cos(\Theta)} - r\sin(\Phi)\frac{\sin(\Theta)}{\cos(\Theta)} \tag{3.7}$$

Matrix entries $\mathbf{F}$(1:3,1:3) represent the calculation of the linearization of the right side of Equation 3.1. This is found by calculating the Jacobian of that matrix. The Jacobian is defined in Equation 3.8. This linearization represents the incremental change in the Euler angles from one iteration of the filter to the next.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial\dot{\Psi}}{\partial\Psi} & \frac{\partial\dot{\Psi}}{\partial\Theta} & \frac{\partial\dot{\Psi}}{\partial\Phi} \\ \frac{\partial\dot{\Theta}}{\partial\Psi} & \frac{\partial\dot{\Theta}}{\partial\Theta} & \frac{\partial\dot{\Theta}}{\partial\Phi} \\ \frac{\partial\dot{\Phi}}{\partial\Psi} & \frac{\partial\dot{\Phi}}{\partial\Theta} & \frac{\partial\dot{\Phi}}{\partial\Phi} \end{bmatrix} \tag{3.8}$$

Matrix entries $\mathbf{F}(1{:}3,4{:}6)$ represents the transformation matrix in Equation 3.1. The entries $\mathbf{F}(4{:}6,4{:}6)$ only model the incremental change in the gyro bias due to the Gauss-Markov model which is assumed to represent the gyro bias instability/in-run gyro bias.

The matrix $\mathbf{G}$ represents the continuous coupling matrix between random noise and the attitude/heading of the helicopter state estimate and contains the following variables in Equation 3.9.

$$\mathbf{G} = \begin{bmatrix} 0 & \frac{\sin(\Phi)}{\cos(\Theta)} & \frac{\cos(\Phi)}{\cos(\Theta)} & 0 & 0 & 0 \\ 0 & \cos(\Phi) & -\sin(\Phi) & 0 & 0 & 0 \\ 1 & \frac{\sin(\Phi)\sin(\Theta)}{\cos(\Theta)} & \frac{\cos(\Phi)\sin(\Theta)}{\cos(\Theta)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

The matrix entries $\mathbf{G}(1{:}3,1{:}3)$ are the transformation matrix of Equation 3.1. The entries $\mathbf{G}(4{:}6,4{:}6)$ models the Gaussian white noise process associated with the output of the IMU gyros.

The continuous matrices $\mathbf{F}$ and $\mathbf{G}$ are the Jacobians of the nonlinear dynamic model evaluated at the present state estimates. For a solution on digital computers these matrices must be discretized. And given that the discrete sampling time is small, the higher order terms of the Jacobians are neglected such that this is a first order approximation.

The vector $\underline{\beta}$ represents the characteristics and magnitude of the process noise from the IMU measurements corrupting the attitude/heading state estimates and is defined in Equation 3.10.

$$\underline{\beta} = \begin{bmatrix} \eta_z & \eta_y & \eta_x & \nu_x & \nu_y & \nu_z \end{bmatrix}^T \tag{3.10}$$

Next, the time update to the discrete state estimation error covariance matrix, $\mathbf{P}$, is determined in Equation 3.11.

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k^{+} \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \tag{3.11}$$

The initial setting of the discrete state estimation error covariance matrix, $\mathbf{P}$, is that of a 6-by-6 diagonal matrix. The first three diagonal elements are set to the variance of the attitude/heading measurement noise, and the last three diagonal elements are set to the variance of the gyro IMU signal noise. The remainder of the elements are set to zero.

The symbol $\mathbf{\Phi}$ in this context represents the discrete state transition matrix rather than one of the Euler angles. The discrete state transition matrix is a discrete approximation of the continuous state transition matrix, $\mathbf{F}$, using Van Loan's method [23]. Note too low of a sampling rate can adversely impact the quality of the state transition matrix approximation and the stability of the filter.

The matrix $\mathbf{Q}$ is process noise covariance matrix that is related to the column vector $\underline{\beta}$ by $\mathbf{Q} = \mathbb{E}[\underline{\beta \beta^T}]$ where $\mathbb{E}$ is the expectation operator. The superscripts $^-$ and $^+$ refer to variables that have different values before and after a measurement update to the filter occurs. In other words, once the filter processes a measurement these variables have to be updated before the next iteration of the filter.

Next, the measurement update of the EKF is performed. The true, discrete measurement vector $\underline{z}$ is defined in Equation 3.12.

$$\underline{z} = \begin{bmatrix} \Psi & \Theta & \Phi \end{bmatrix}^T \tag{3.12}$$

The actual measurements themselves are made by the Vicon MX optical motion capture system. The actual discrete measurement vector, $\hat{\underline{z}}$, is assumed to be corrupted versions of the true measurement vector, $\underline{z}$ by a linear relationship defined in Equation 3.13.

$$\hat{\underline{z}}_k = \mathbf{H}\underline{z}_k + \mathbf{R} \tag{3.13}$$

With the measurements corrupted by a zero-mean white Gaussian noise defined by measurement noise covariance $\mathbf{R}$, where $\mathbf{R}$ is defined in Equation 3.14, the noise is defined by the variance σ^2 and is assumed constant and equal for each measured Euler angle.

$$\mathbf{R} = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \tag{3.14}$$

The measurement sensitivity matrix $\mathbf{H}$ is defined in Equation 3.15.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.15}$$

Finally, the discrete Kalman Filter gain, $\mathbf{L}_k$, defined in Equation 3.16 is calculated using *a priori* information. Once a measurement/observation update is available to the filter, the discrete state estimation error covariance matrix, $\mathbf{P}_k^+$, defined in Equation 3.17, and the discrete state estimate, $\underline{\hat{x}}^+$, defined in Equation 3.18 are updated and the whole process is repeated.

In the case of gain scheduling the EKF the calculation of the Kalman Gain in Equation 3.16 is greatly simplified. Gain scheduling reduces this equation to just a simple constant, look-up table, or a basic function for $\mathbf{L}_k$. In the present research, the case of a constant gain based on the average gain and gains calculated as a multi-nonlinear regression functions dependent on helicopter attitude/heading are explored. In both cases the scheduled gains are derived from post-processing gains predicted by the optimal EKF case. More sophisticated ways of allocating optimal gain schedules were considered but not implemented [5, 6, 7].

$$\mathbf{L}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \tag{3.16}$$

For the discrete estimation error covariance matrix update, it is necessary to use a more general form of the update equation that can be utilized when solving suboptimal filter equations [8]. This is by no means the only alternative form for solving the suboptimal filter equations [17]. This change in the update equation is due to the filter no longer being a minimum variance, unbiased estimator as in the optimal case. This general form can also used in the solution of the optimal equations. Below it is written as shown in the optimal case with time dependent gains.

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{L}_k\mathbf{H}]\mathbf{\Phi}_k\mathbf{P}_k^-\mathbf{\Phi}_k^T[\mathbf{I} - \mathbf{L}_k\mathbf{H}]^T + [\mathbf{I} - \mathbf{L}_k\mathbf{H}]\mathbf{Q}_k[\mathbf{I} - \mathbf{L}_k\mathbf{H}]^T + \mathbf{L}_k\mathbf{R}\mathbf{L}_k^T \tag{3.17}$$

Where $\mathbf{I}$ is a 6-by-6 identity matrix.

$$\underline{\hat{x}}_k^+ = \underline{\hat{x}}^- + \mathbf{L}_k[\underline{z}_k - \underline{\hat{z}}_k] \tag{3.18}$$

In the simplest approach that of constant value gain scheduling the terms $\mathbf{Q}_k$ and $\mathbf{L}_k$ become time-invariant.

A MATLAB implementation of the above digital algorithm used in this thesis can be seen in Appendix B.

Chapter 4

Results of Filter Performance Studies

4.1 Simulations Studies

A previously developed MATLAB script was used to generate simulated trajectories for testing the EKF algorithms developed in thesis. Although, the script can generate position, velocity, and magnetic field time histories anywhere on earth as long as the trajectories can be defined by continuous sine functions, the only motions of concern in this thesis are attitude/heading trajectories. The angular rate channels include earth rate components however the earth rate signal becomes undetectable when the generated signal is corrupted with noise.

As the script only generates perfect trajectories it is necessary to pass them through a second script that generates corresponding corrupted IMU measurements. The absolute attitude/heading measurements are for a generic sensor and not the VICON system used in empirical testing.

The second script allows for incorporating a range of IMU errors such as constant sensor bias, white noise, correlated noise, and scale factor errors. These error sources were set to values comparable to the class of IMU used in physical testing as shown in Table 4.1. A range of attitude/heading time histories were generated in order to serve as test cases for different methods of gain scheduling the EKF. An example of uncorrupted and corrupted simulated gyro data can be seen in Figure 4.1.

Gyro Sensor Error Terms	Error Term Value	Units
Bias/Offset	0.5	[deg/s]
White Noise	0.05	[deg/s] 1σ
Gauss Markov Bias	180	[deg/s]
Gauss Markov Time Constant	300	[s]

Table 4.1: Error Terms Applied to All Gyro Axes in All Simulated Test Cases

The following trajectory figures are plotted in unwrapped angles for clarity and are also unwrapped prior to use in the EKF to avoid introducing sudden jumps into the Euler angle parameters as some angles pass through 360 degrees. One characteristic observed in the simulated trajectories is apparent modulation of the IMU body rates in one channel due to attitude/heading changes in the other channels. This is due to inherent cross-coupling in the kinematic rate equation introduced in Equation 3.1. Thus several different trajectories are presented and tested to determine the effects these different errors have on the gain scheduled EKF. All trajectories have a duration of 1200 seconds. This duration was chosen arbitrarily. In some cases, the magnitude of integrated angular IMU drift is greater than the magnitude of the true oscillation, thus giving the oscillations the appearance of a straight line.

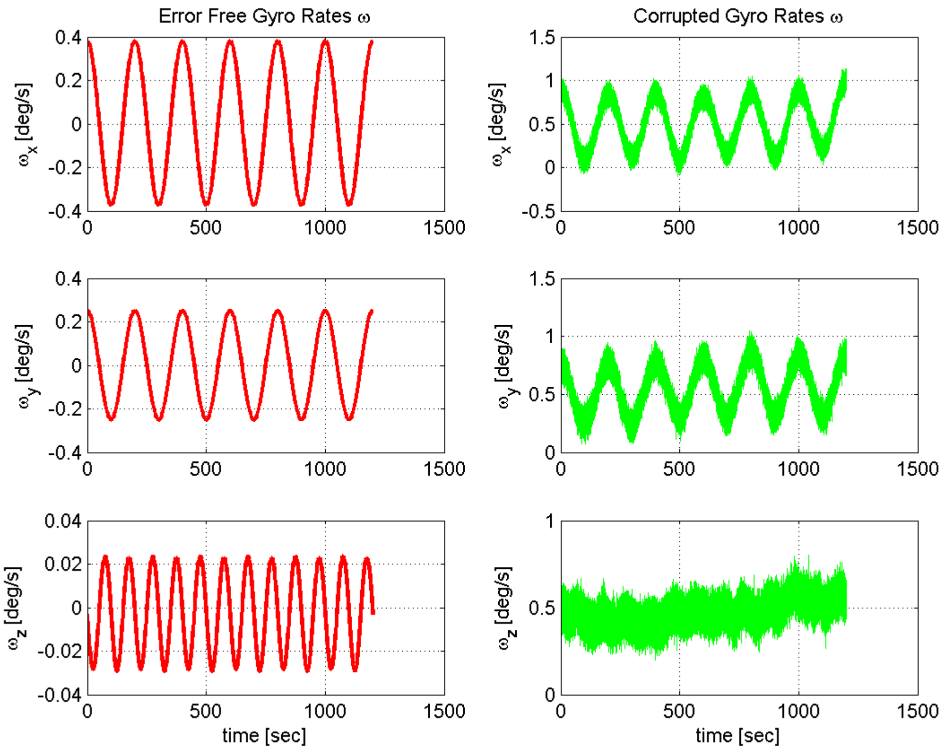

Figure 4.1: Comparison of Uncorrupted and Corrupted Simulated Gyro Measurements

4.1.1 Trajectory of Benign Pitch and Roll Dynamics

The benign pitch and roll trajectory is described by a pitch sinusoid with an amplitude of ±8° and a roll sinusoid with an amplitude of ±12°. Both sinusoids are in-phase with each other and have an oscillation frequency of 0.005 Hz. Yaw is a constant value of 0 and remains unchanged throughout the trajectory. The trajectory generated is illustrated in Figure 4.2.

4.1.2 Trajectory of Aggressive Pitch Dynamics

The aggressive pitch trajectory is described by a pitch sinusoid with amplitude of ±45° and an oscillation frequency of 0.005 Hz. Roll and yaw both have a constant value

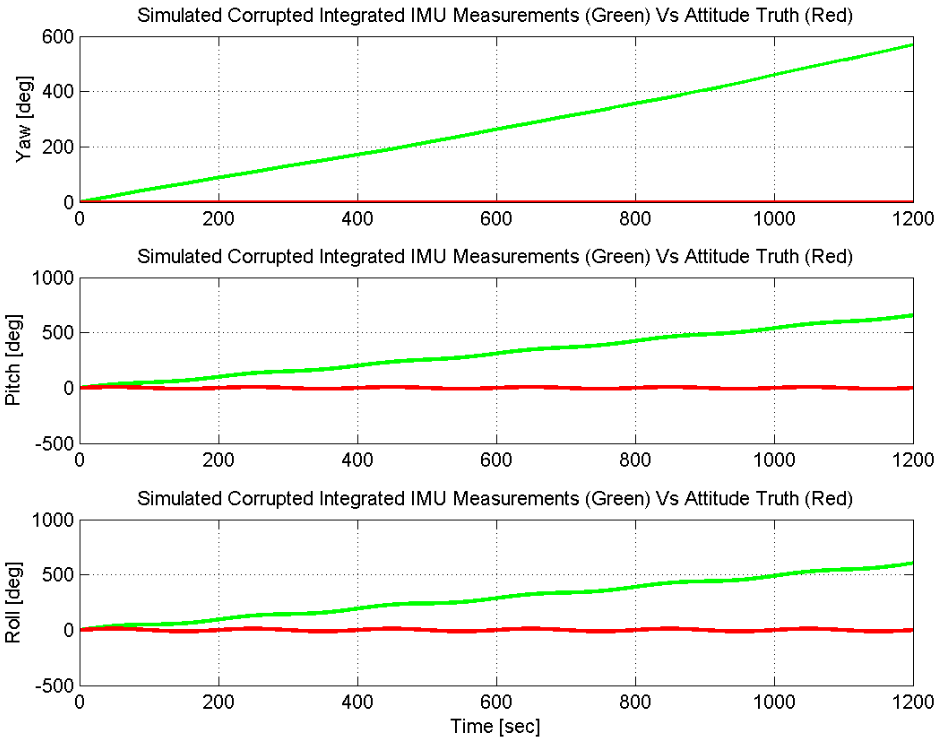

Figure 4.2: Simulated Trajectory of Benign Pitch and Roll Dynamics

of 0 and remain unchanged throughout the trajectory. The trajectory generated is illustrated in Figure 4.3.

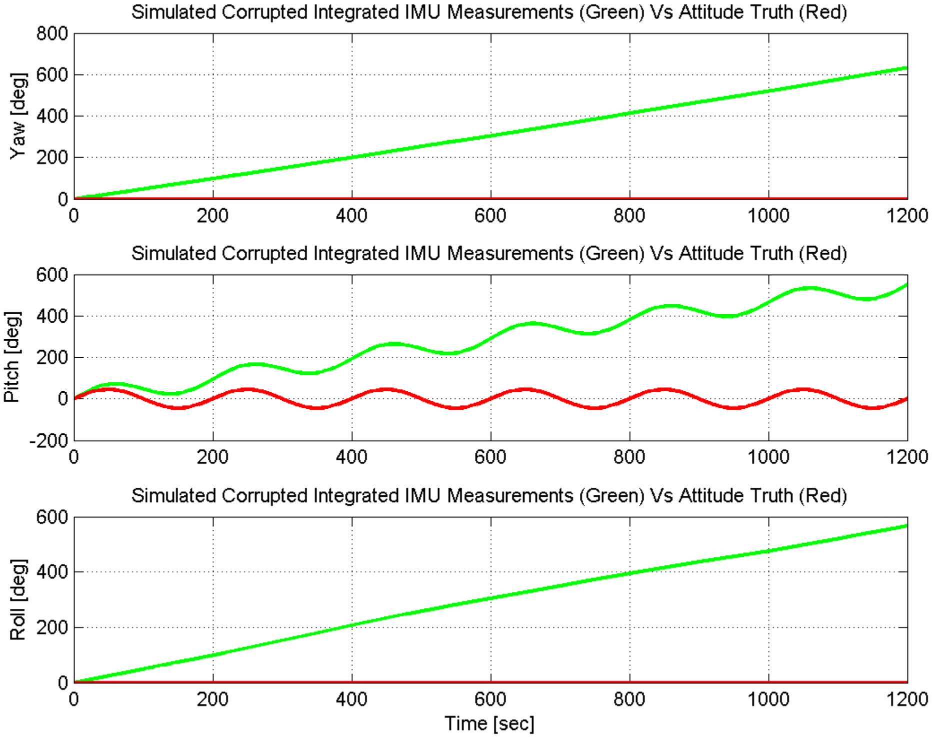

Figure 4.3: Simulated Trajectory of Aggressive Pitch Dynamics

4.1.3 Trajectory of Aggressive Out-of-Phase Pitch and Roll Dynamics

The aggressive pitch and roll trajectory is described by a pitch and roll sinusoid with an amplitude of ±45°. The pitch and roll sinusoids are 90° out of phase with each other and both have an oscillation frequency of 0.005 Hz. Yaw is a constant value of 0 and remains unchanged throughout the trajectory. The trajectory generated is illustrated in Figure 4.4.

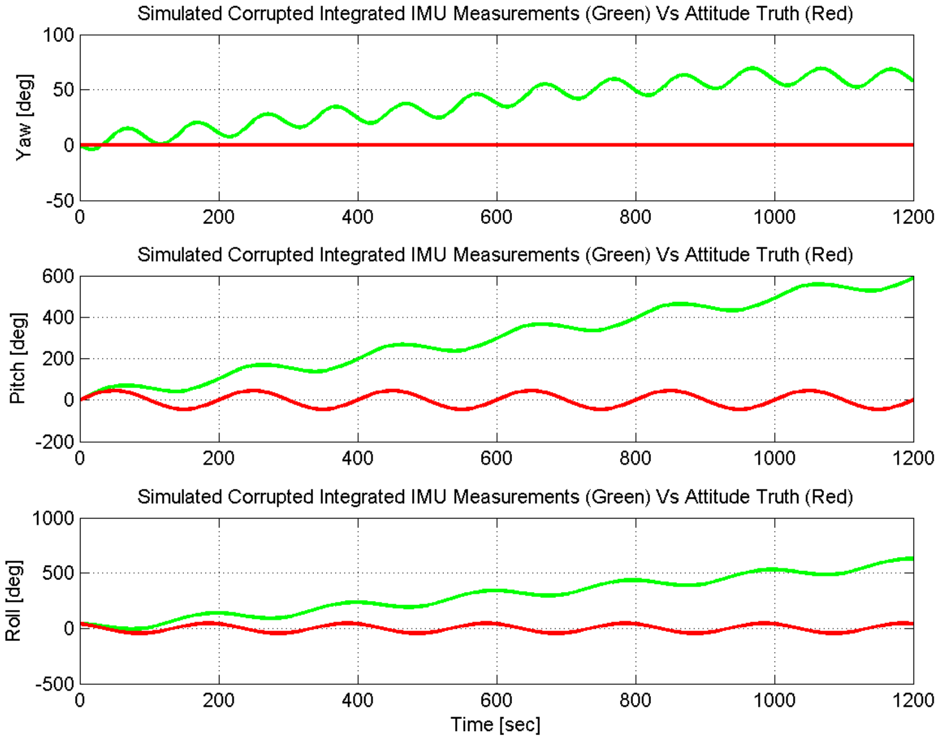

Figure 4.4: Simulated Trajectory of Aggressive Out of Phase Pitch and Roll Dynamics

4.1.4 Trajectory of Aggressive In-Phase Pitch, Roll, and Yaw Dynamics

The aggressive pitch, roll, and yaw trajectory is described by sinusoids with an amplitude of ±45°. All three sinusoids are in phase and all three sinusoids have an oscillation frequency of 0.005 Hz. The trajectory generated is illustrated in Figure 4.5.

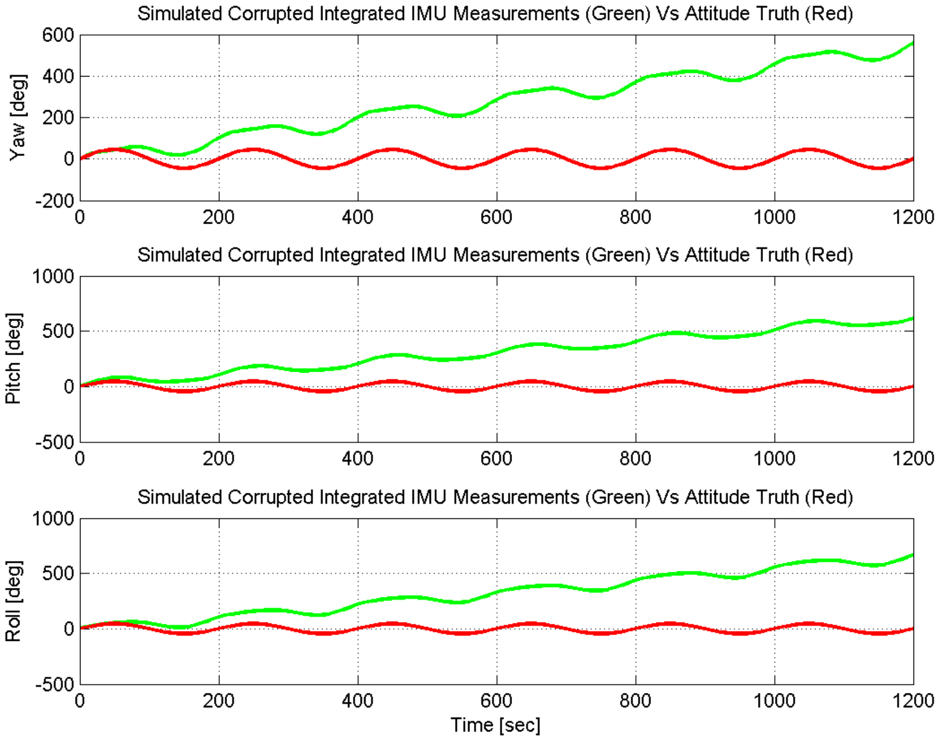

Figure 4.5: Simulated Trajectory of Aggressive In-Phase Pitch, Roll, and Yaw Dynamics

4.1.5 Trajectory of Aggressive Out-of-Phase Pitch, Roll, and Yaw Dynamics

The aggressive pitch, roll, and yaw trajectory is described by sinusoids with an amplitude of ±45°. The pitch and roll sinusoids are 90° out of phase with each other and pitch and yaw are 180° out of phase; and all three sinusoids have an oscillation frequency of 0.005 Hz. The trajectory generated is illustrated in Figure 4.6.

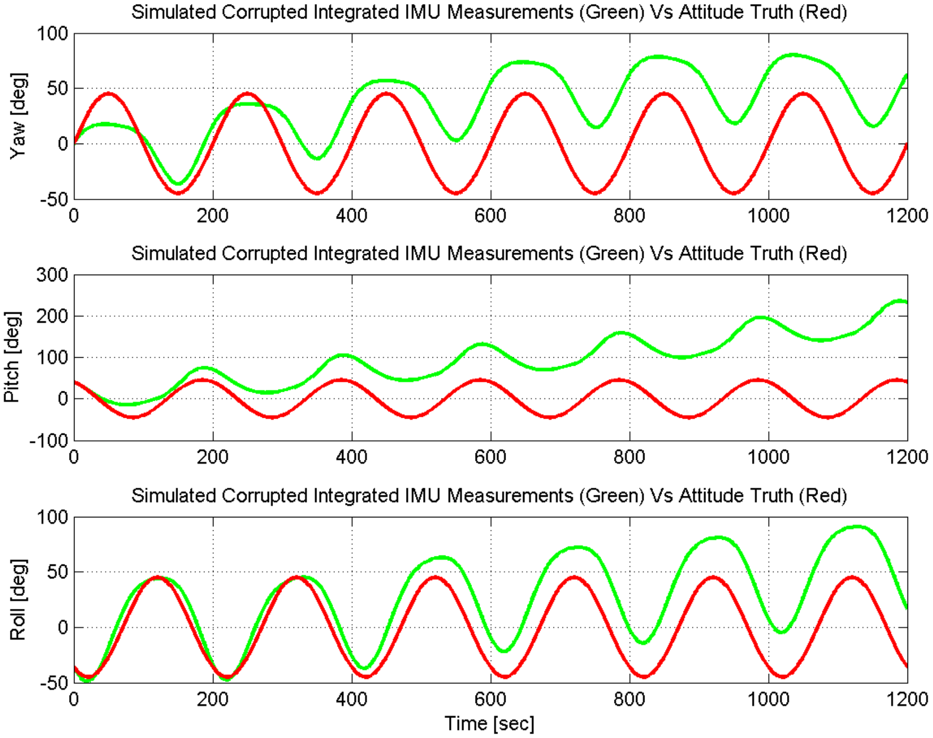

Figure 4.6: Simulated Trajectory of Aggressive Out-of-Phase Pitch, Roll, and Yaw Dynamics

4.1.6 Trajectory of Aggressive Out-of-Phase Pitch and Yaw Dynamics With Constant Roll Angle

The aggressive pitch and yaw trajectories are described by two sinusoids with an amplitude of ±45°. The pitch and yaw sinusoids are 90° out of phase with each other and both sinusoids have an oscillation frequency of 0.005 Hz. The roll angle for a given trajectory is held constant. Each trajectory has the roll angle set to a different constant. The roll angles tested were ±45°, ±35°, ±25°, ±15°, ±10°, ±5°, and 0°. An example of one of the trajectories generated is illustrated in Figure 4.7, the trajectory is of the constant -15° roll angle instance.

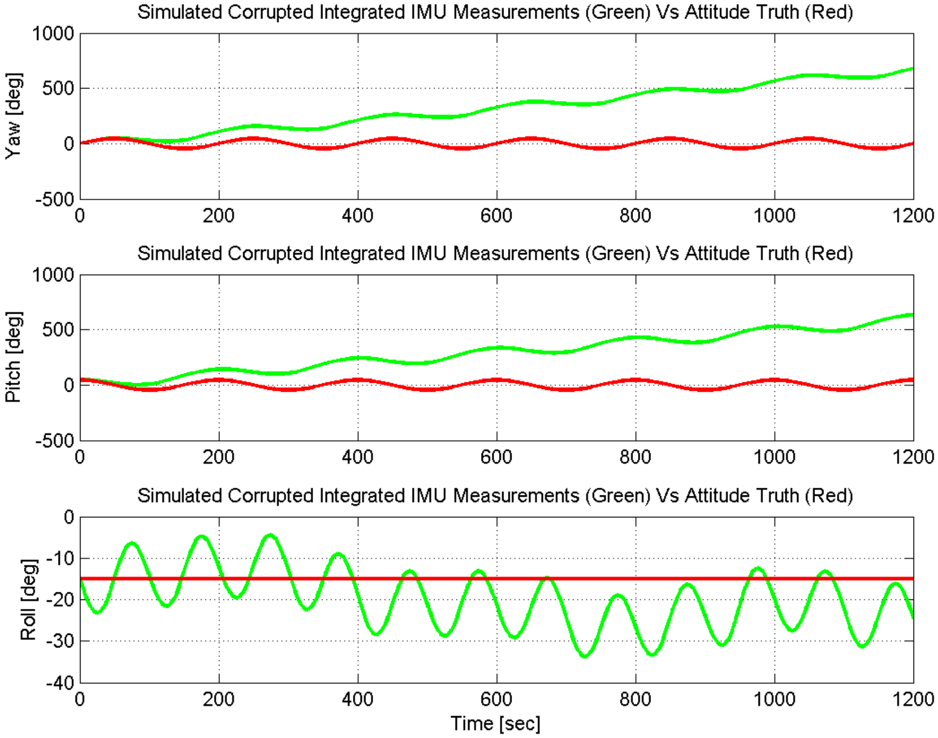

Figure 4.7: Example of Simulated Trajectory of Aggressive Out-of-Phase Pitch and Yaw Dynamics with Constant Roll Angle

4.2 Simulated Extended Kalman Filter Comparing Optimal to Gain Scheduling Results

The trajectories described in 4.1 were used to test out the different gain scheduling approaches to determine which approach would be suitable for future development. The following sections describe the results of testing the different simulated trajectories.

4.2.1 Description of Gain Scheduling Approaches

As previously noted section 2.1.1, three approaches were taken in gain scheduling the EKF:

1. Constant gains
2. Gains calculated from a multi-nonlinear regression of the optimal gains versus the attitude/heading state estimates
3. Gains calculated from tailored multi-nonlinear regression of the optimal gains form a series of trajectories versus the attitude/heading state estimates.

Simulation determined that specific gain elements in the Kalman gain matrix were significant; the remaining elements were not significant and consequently were set to zero. The significant elements for all three approaches for gain scheduling are $L(1,1)$, $L(2,2)$, $L(3,3)$, $L(4,3)$, $L(5,2)$, and $L(6,1)$.

The constant gain schedule is determined by taking the data set and arbitrarily discarding the first 20% of the data set in order to remove the effects of Kalman gain transients on startup of the EKF. The remaining Kalman gain time history is then averaged for the significant gain elements. The average values are then used as constant gains.

The second scheduling approach used a non-tailored multi-nonlinear regression of the optimal gain histories. Once again the first 20% of the data set was disgarded. Then a regression was performed on each significant gain element versus attitude/heading state estimates and included attitude/heading interaction terms. The gain elements were the response variables and the Euler angles were the scheduling variables. The form of the regression is seen in Equation 4.1 where a_n, with $n = 1, 2, 3, ..., 7$, are the coefficients used to fit the data. The form of the regression equation used in this case was not tailored to the optimal gain curves recorded.

$$L(row, column) = a_1 + a_2\Psi + a_3\Theta + a_4\Phi + a_5\Psi\Theta + a_6\Psi\Phi + a_7\Theta\Phi \tag{4.1}$$

If all the terms associated with the scheduling variables are set to zero except, a_1, then the regression equation reduces to finding the average constant value gain. Thus a_1 by itself is also used to signify the constant value gain.

The third scheduling approach took the optimal gains from a series of similar trajectories and found regression equations tailored to fit the curves of optimal gain associated with each specific and significant Kalman gain element. This was done to test the performance of a gain schedule that utilized a best sub-set style of regression approach and to see its performance on trajectories not used to generate the original gain schedule. As shown later, the performance of this approach on trajectories different then the family used to generate the gain schedule was poor.

The tailored gain scheduling approach utilized the regression equations seen in Equation 4.2 to 4.7.

$$\begin{aligned}
L(1,1) &= a_1\Psi^2 + a_2\Theta^2 && (4.2)\\
L(2,2) &= a_1\Psi + a_2\Psi^2 + a_3\Theta + a_4\Theta^2 + a_5\Phi + a_6\Phi^2 && (4.3)\\
L(3,3) &= a_1\Psi^2 + a_2\Theta^2 && (4.4)\\
L(4,3) &= a_1\Psi^2 + a_2\Theta^2 && (4.5)\\
L(5,2) &= a_1 + a_2\Phi + a_3\Phi^2 && (4.6)\\
L(6,1) &= a_1\Psi + a_2\Psi^2 + a_3\Theta + a_4\Theta^2 + a_5\Phi + a_6\Phi^2 && (4.7)
\end{aligned}$$

A summary listing of the coefficients for all of the gain schedules developed using both simulated and real trajectory data is available in Appendix C.

4.2.2 Performance Index and Results of EKF Gain Scheduling

The results of gain scheduling the EKF exhibited a performance penalty over the optimal non-gain scheduled approach. Generally, the gain schedule approach based on the non-tailored multi-nonlinear regression of the optimal attitude/heading state estimates gave superior results over the constant gain case, although this is somewhat expected as the trajectories tested are the same ones used to generate the regression. The tailored multi-nonlinear regression only gave superior results when the trajectory tested was from the same family of trajectories in which it was designed from. When tailored regression is applied to trajectories outside the ones used to generate the regression the results are significantly poorer. Summary performance index results can be seen in Table 4.2.

Trajectory Reference	Constant Gain	Nonlinear Function Gain	Tailored Nonlinear Function Gain
Figure 4.2	7.9058e-5	8.7770e-5	349.689
Figure 4.3	0.0051	0.0046	24.612
Figure 4.4	0.0283	0.0277	343.257
Figure 4.5	0.0076	0.0017	17.835
Figure 4.6	0.0694	0.0628	0.0989
Figure 4.7	0.0841	0.0824	0.0683

Table 4.2: Suboptimal EKF Performance Index (μ) Results for Simulated Trajectory Test Cases

4.2.3 Effects of the Frequency of Measurement Update On Suboptimal EKF Performance

The EKF filter implementation allows measurement updates to occur at a lower rate than the process dynamics, which is often the case in real systems. For example, Kalman Filtered INS/GPS systems typically operate at different rates. The GPS measurement update may only occur at 1 Hz, but INS updates can occur at several hundred Hz. The EKF can seamlessly blend these two different signal rates into optimal state estimates.

The presented results in this thesis are based on the measurement update occurring for every sample. For example, the simulated trajectory data is sampled at 100 Hz so that the measurement update occurs also at 100 Hz. Likewise, the experimental data is sampled at 50 Hz so in this case the measurement update also occurs at 50 Hz.

The EKF algorithm was modified to test the effect that decreased measurement update rate has on estimator performance. The simulated trajectory of benign pitch and roll dynamics was used in this test. As expected, the decreased measurement update rates resulted in poorer filter estimation performance as shown in Table 4.3. The multi-nonlinear regression results are not reported because the reduced measurement frequency resulted in the Φ matrix becoming singular.

The singular matrix leads to numerical problems that appear to be the result of erroneous attitude/heading state estimates creating a positive feedback loop with the multi-nonlinear regression equation whereby unreasonable gains are used in the state update equations which drives the attitude/heading estimates further off. Prior to MATLAB aborting the test case, the attitude/heading estimates are wildly off by many orders of magnitude. The initial trigger for this numerical problem was not determined.

The constant gain implementation avoids this issue as the gain is fixed, unlike the regressions which become meaningless when evaluated beyond its fitted initial dataset. These differences also suggest several possible fixes to the problem.

A means of implementing the multi-nonlinear function gain, and still maintaining stability, may be to exercise the fitted equation with attitude/heading measurements. These measurements should be bounded and decoupled from effects of positive feedback rather than the use of attitude/heading state estimates. Another option is to effectively have the gain saturate at a maximum or minimum value instead of evaluating the regression equation beyond its original dataset.

Measurement Update Frequency	Constant Gain Performance Index
1:10	8.4262
1:9	6.8096
1:8	5.5267
1:7	4.1785
1:6	3.2213
1:5	2.2826
1:4	1.4799
1:3	0.8101
1:2	0.2829
1:1	7.9058e-5 (Baseline)

Table 4.3: Impact EKF Measurement Update Frequency on Performance Index

4.3 Experimental Setup

In order to test the EKF algorithm outside the simulation domain, data was collected from a Remote Control (RC) helicopter. The helicopter was equipped with an IMU to provide the measurements for the process model and a vision based motion capture system in the Interactive Guidance and Control Lab was used to provide attitude/heading measurement updates to the filter.

4.3.1 RC Helicopter Test Vehicle

A RC helicopter was used to gather real flight data. The helicopter used is a Blade CX-2 purchased from E-Flite [24]. Figure 4.8 shows an example Blade CX-2. Figure 4.9 shows the modified Blade CX-2 used throughout testing, note the black cube to the right of the helicopter's main mast/shaft is the IMU. It was chosen because it is small enough to fly indoors, but large enough to carry a reasonable sized payload of electronics and a complete MEMS IMU. The helicopter features tandem counter-rotating coaxial main rotors and the upper rotor incorporates a free-floating Bell style stabilizer bar. Additionally, helicopter control is implemented through a conventional swash-plate mechanism acting on the lower rotor to generate cyclic blade pitch variations. Other details on the Blade CX-2 are summarized below in Table 4.4

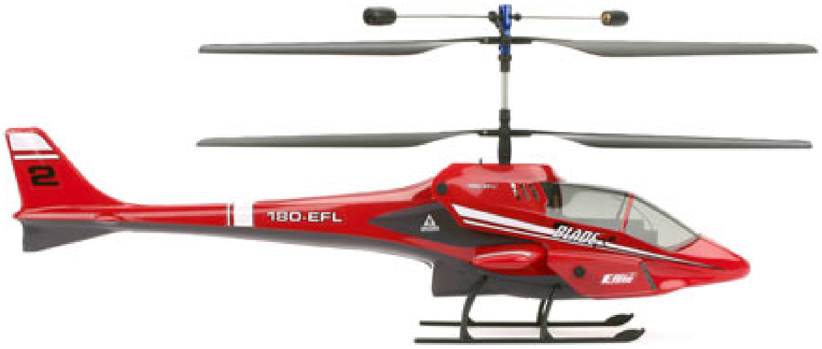

Figure 4.8: Example of the E-Flite Blade CX-2 Remote Control Helicopter

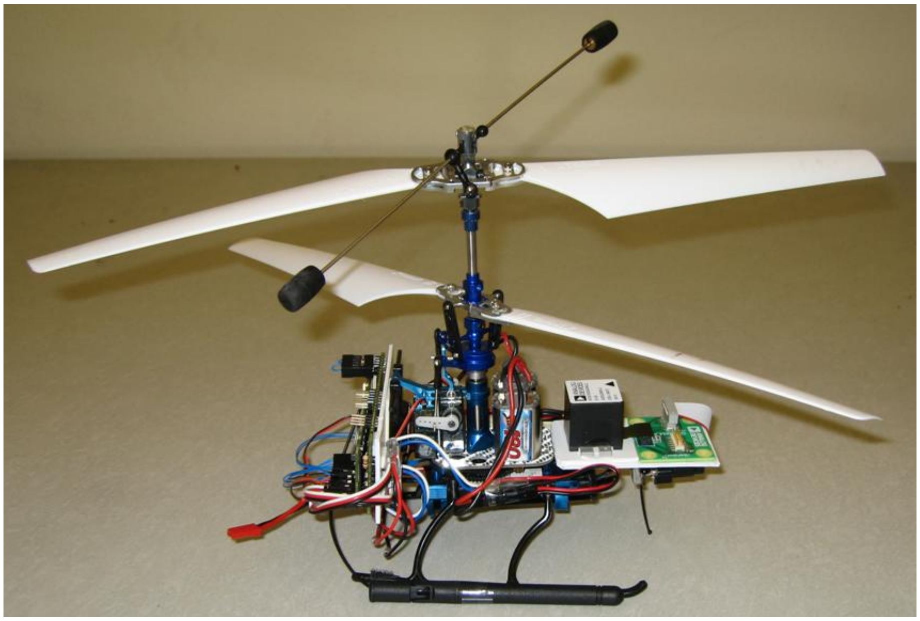

Figure 4.9: Modified E-Flite Blade CX-2 Test Helicopter - With IMU Installed

Specifications	Quantity	Units
Main Rotor Span	0.345	m
Mass Empty	200	g
Payload	50	g
Length	0.300	m
Height	0.175	m
Propulsion	Electric	N/A

Table 4.4: Blade CX2 Helicopter Specifications

4.3.2 Inertial Measurement Unit

The helicopter is equipped with a MEMS based digital IMU produced by Analog Devices [25]. The IMU contains three accelerometers, and three gyroscopes and an embedded controller for filtering and compensating the raw sensor signals. The Analog Devices part number for the IMU is ADIS16350. The output signals are time stamped and recorded at 50 Hz by the Vicon data collection system. IMU tri-axis accelerometer data was recorded during experimentation, but the information was not incorporated in the EKF. Additional IMU performance parameters evaluated at 25 °C are summarized in Table 4.5

Gyro Performance Specifications	Nominal Value	Units
Dynamic Range	±75	°/s
Initial Sensitivity Error (1σ)	±10,000	ppm
Sensitivity Error Due to Temperature	600	ppm/°C
Axis Nonorthogonality Error	±0.05	°
Axis Misalignment Error	±0.5	°
Sensitivity Nonlinearity (Best Fit Line)	0.1	% of Full Scale
In-Run Bias Stability	0.15	°/s
Output Noise (±75°/s, 32-tap FIR Filter)	0.17	°/s rms
Angular Random Walk	4.2	$^\circ/\sqrt{hr}$
Rate Noise Density (f=25Hz, ±300°/s, no filter)	0.05	$^\circ/s/\sqrt{Hz}$ rms
Bias Linear g-Sensitivity (1σ)	0.1	°/s/g
Bias Voltage Sensitivity (V_{cc}=4.75-5.25V)	0.25	°/s/V
IMU Dimensions	23x23x23	mm

Table 4.5: Analog Devices ADIS16350 MEMS IMU

The IMU is attached to the helicopter with improvised rubber isolators and is located directly ahead of the electric motors that drive the tandem coaxial main rotors. The effects of unintended IMU lever arms about the helicopter center of gravity and non-rigid helicopter body modes were ignored in both the mounting of the IMU and in data processing and in the EKF.

4.3.3 Infrared Motion Tracking Camera System

The laboratory (Interactive Guidance and Control Lab) where the helicopter testing was conducted is equipped with a Vicon MX optical motion capture system. The system is composed of 6 MX40 cameras located around the perimeter. Each camera has a built in infrared emitter. The camera is designed to detect the emitted infrared light reflected off special spherical targets attached to the helicopter structure.

The images collected by these cameras are streamed to a stand alone computer where proprietary Vicon software calculates in real-time the attitude/heading and body rates, expressed in Euler angles, and also calculates the helicopter's position and velocities This data is time-stamped and recorded at a 50 Hz rate in the same file as the IMU data. Previous work has identified that the attitude/heading estimated by the system is accurate to 0.06° 1σ [21].

4.4 Experimental RC Helicopter Test Trajectories

Data from two separate tests, using real measurements and hardware, demonstrated the ability to estimate attitude/heading using a gain scheduled EKF. The two test trajectories have been named the following way:

1. Unpowered Constrained Pendulum Trajectory Test

2. Powered Free Flight Trajectory Test

4.4.1 Unpowered Constrained Pendulum Trajectory Test

This data set from prior research was used because of its advantageous characteristics. Another research project[20] utilized an identical test setup, but instead hung the helicopter vertically from the ceiling to create a pendulum with the combined helicopter, IMU, and Vicon reflectors acting as the pendulum mass. The researcher then perturbed the setup in order to start oscillating the system back and forth.

This setup had several advantages over a pure free flight condition. The helicopter was non-operating during the test so concerns about operational vibrations or electromagnetic interference impacting the quality of the IMU measurements were eliminated.

Another advantage of this setup is the pendulum trajectory itself is also generally constrained to a pitching motion and the equations of motion are known analytically if the helicopter is assumed to be a point mass. The actual data contained more roll motion than expected. Future pendulum tests should improve the test setup to reduce this. The author feels that a ceiling mounted hinge or bearing attached to a long slender rod which itself is rigidly connected to the frame of the helicopter would be a better setup. Experiments should also be done with the pendulum rod connected directly to the IMU without the helicopter involved in order to attempt to gather cleaner data.

The resulting Euler angle time-series of the pendulum test are shown in Figure 4.10. The Euler angles plotted are generated from the measurements collected from the IMU and from the Vicon system. The IMU measurements are integrated forward in time using a simple cumulative sum. The drift of the integrated IMU measurements is due to latent gyro bias and is visible in Figure 4.10.

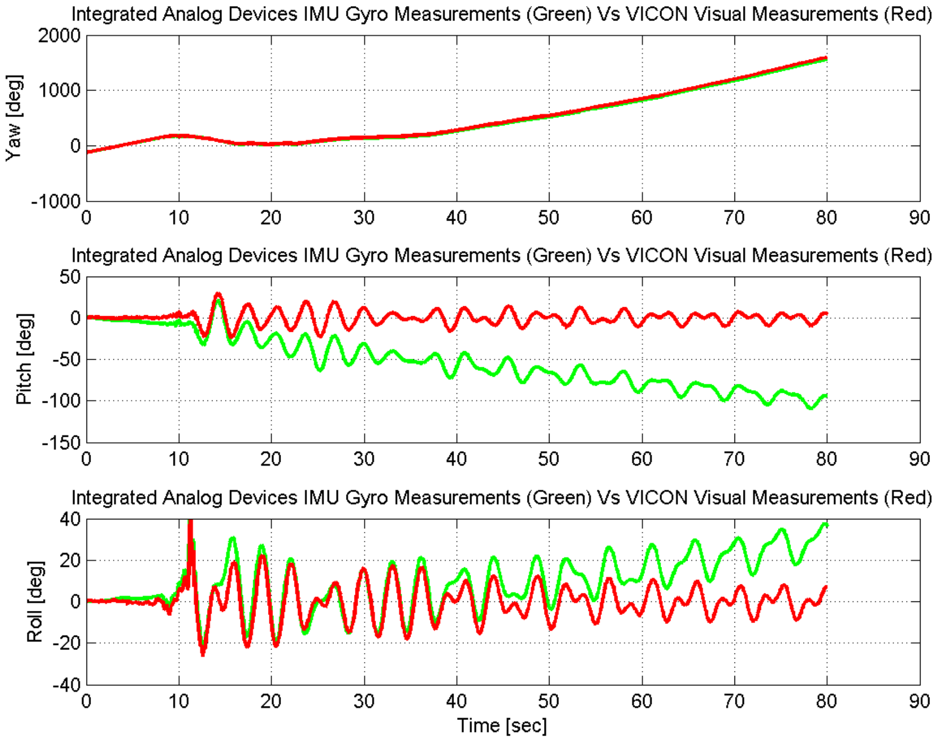

Figure 4.10: Unfiltered Pendulum Time-Series Test Data: Integrated IMU Vs. Vicon Measurements

Figure 4.11 illustrates the state estimates and Vicon measurements. The left-side series compare attitude/heading estimates against Vicon attitude/heading measurements. The right-side series illustrates the filter's estimate of the IMU gyro biases. In real test cases there is no independent measurement of gyro bias, consequently, there is no knowledge of the actual true gyro bias. An interesting observation here is that the gyro bias estimates change quite rapidly over the course of the data collected. It is known that in practice the gyro bias cannot change that quickly for this particular type of IMU. These plots are based on the optimal, non-gain scheduled EKF.

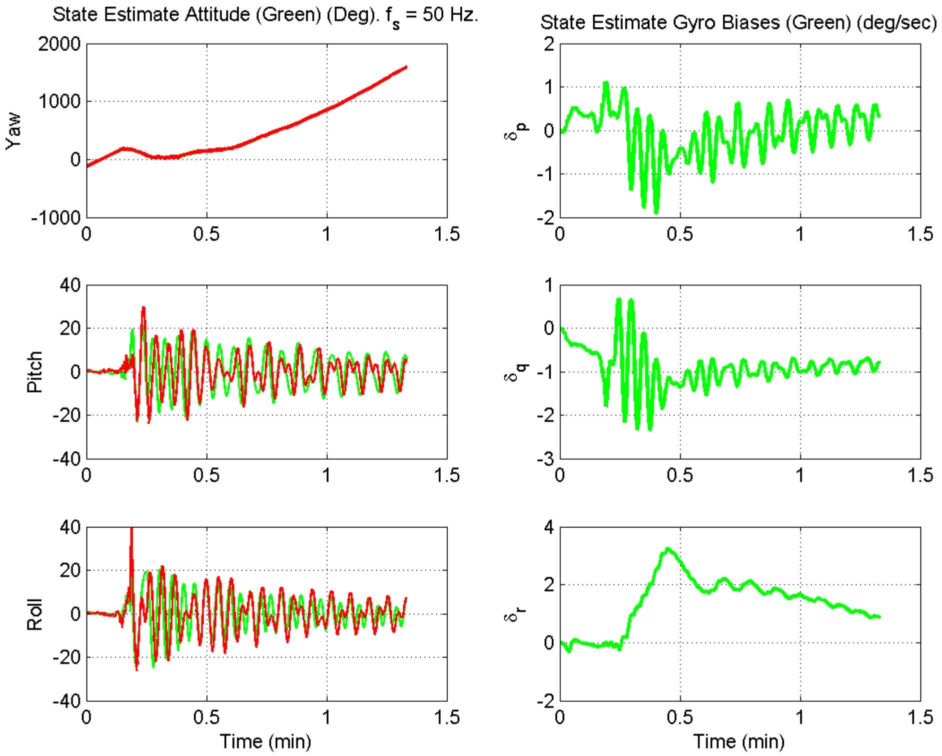

Figure 4.11: Optimal EKF Processed Pendulum Data: Filter State Estimates (Green) Vs. Vicon Measurements (Red) (Note oscillations of gyro bias estimates)

Figure 4.12 illustrates the same trajectory as Figure 4.11, but instead is based on the constant gain EKF. Note the differences between the two in terms of yaw bias estimates, δ_r, and what may indicate an initial convergence problem.

Constant Gain	Nonlinear Function Gain	Tailored Nonlinear Function Gain
0.0243	0.0271	NaN

Table 4.6: Suboptimal EKF Performance Index (μ) Results for Pendulum Test Data

The performance index for the three cases of the suboptimal EKF implementations are shown in Table 4.6.

In the case where the gains are scheduled as a multi-nonlinear function of the state estimates of the Euler angles the results appear similar to that seen in Figure 4.12.

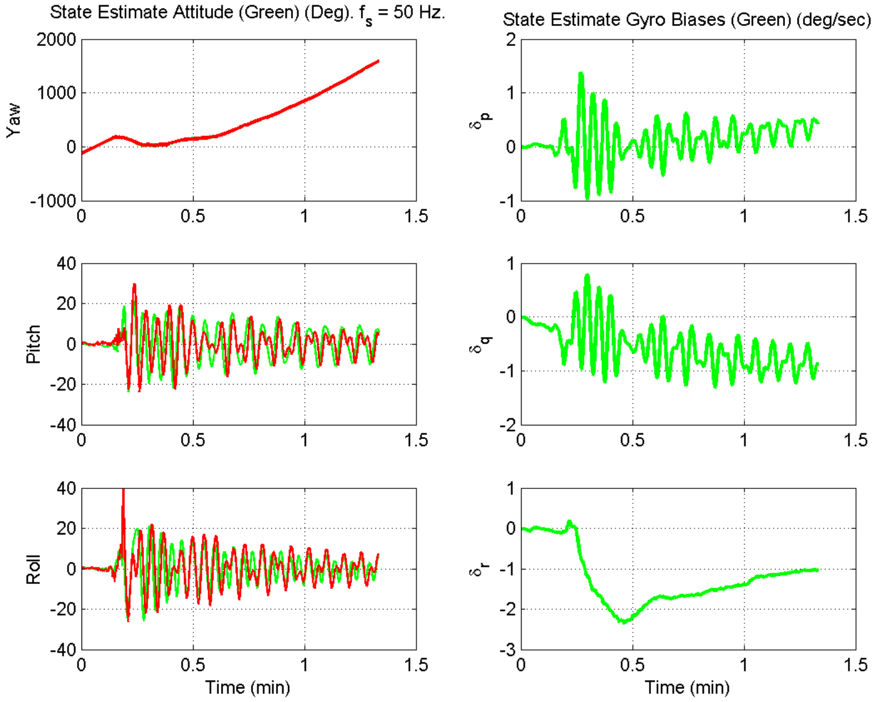

Figure 4.12: Constant Gain EKF Processed Pendulum: Suboptimal Filter State Estimates (Green) Vs. Vicon Measurements (Red)

4.4.2 Powered Free Flight Trajectory Test

In addition to the pendulum-like test data, additional data was also collected in free-flight. The helicopter was flown by a student pilot through a series of pitching and rolling motions

The resulting Euler angle time-series of the free-flight test is shown in Figure 4.13. As before, the Euler angles plotted are generated from the measurements collected.

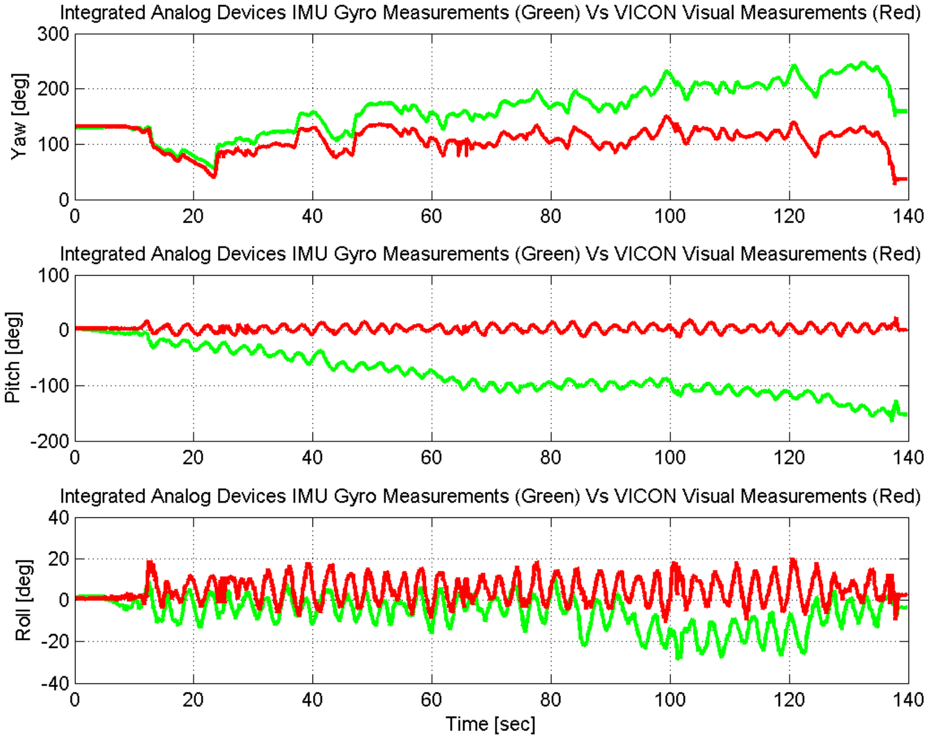

Figure 4.13: Unfiltered Helicopter Free Flight Time-Series Test Data: Integrated IMU Vs. Vicon Measurements

The optimal EKF state estimates is shown for comparison purposes in Figure 4.14.

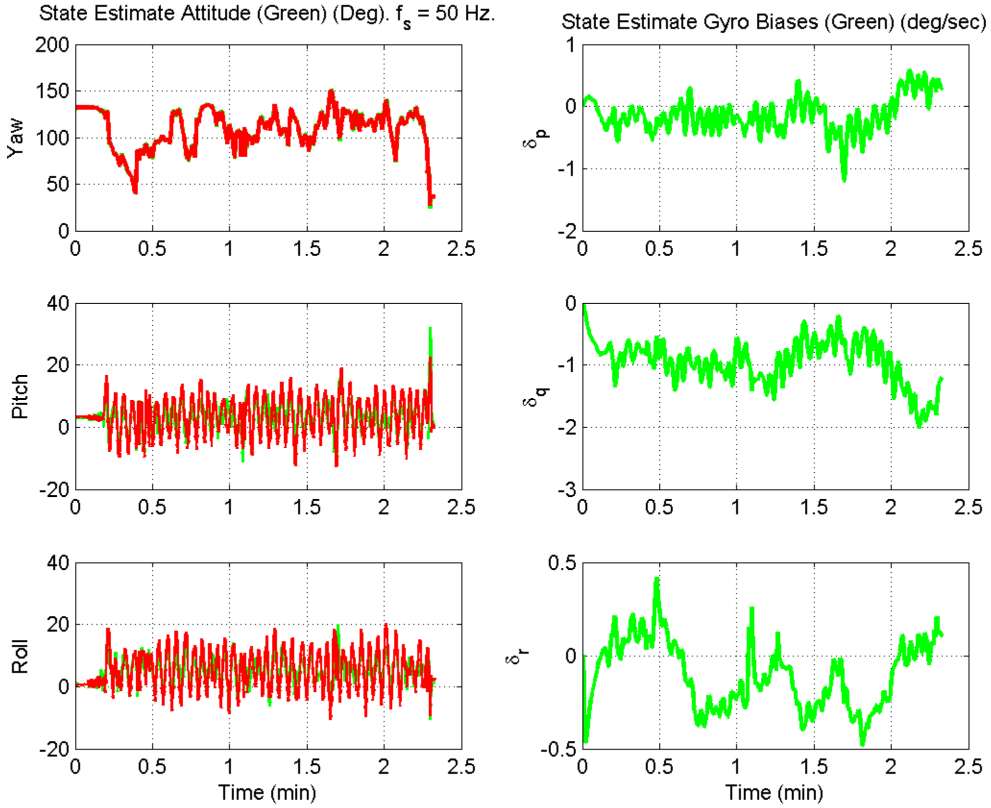

Figure 4.14: Optimal EKF Processed Helicopter Free Flight Data: Filter State Estimates (Green) Vs. Vicon Measurements (Red)

The results of the free flight test data is similar to the pendulum data collected. Table 4.7 shows the performance of the three suboptimal filter implementations.

Constant Gain	Nonlinear Function Gain	Tailored Nonlinear Function Gain
0.0061	0.0057	0.4464

Table 4.7: Suboptimal EKF Performance Index (μ) Results for Helicopter Free Flight Test Data

4.5 Experimental Results

The real gyro bias estimates from testing exhibit a higher variability than expected from simulation, as seen in Figure 4.11. Investigations into this observation led to the realization that the simulated trajectories and corresponding corrupted IMU signals did not include gyro sensitivity, gyro misalignments, and or other gyro error sources in the calculations used to corrupt the generated IMU signals. The original simulated trajectories only assumes gyro errors of noise and bias. The addition of these other gyro errors to the simulation introduced higher variability in the bias estimate, and is consistent with that seen when using the EKF on the real data which naturally contains a range of both deterministic and random error sources. This bias variability is due to differences or mismatches between the system dynamics modeled in the EKF and the dynamics of the actual experimental data. The modeled gyro error system dynamics in the EKF is a major simplification of the actual gyro error dynamics.

Another observation is that, on occasion, the measurement update provided by the Vicon was aphysical and there were measurement outages. These outages were corrected in post-processing, otherwise the EKF algorithm implemented would have swallowed these errors with likely adverse consequences for the filter. Online algorithms in practice would have to be setup to detect and reject wildly inaccurate measurements, and the system would then rely on dynamic model projections to coast through the outage.

4.5.1 Confirmation of Experimental Observations Through Simulation

Two attempts were made to replicate the bias estimate behavior seen in the flight data. In one case, misalignments were intentionally introduced in the simulated benign trajectory data. The magnitude of the misalignments was arbitrarily selected. In the other case, gyro sensitivity errors were introduced. Both resulted in unsteady gyro bias estimates being predicted by the EKF, although with slight differences between the two error sources.

Figure 4.15 shows the EKF predicted performance for the baseline benign pitch and roll trajectory introduced in section 4.1.1. This trajectory only contains gyro bias errors.

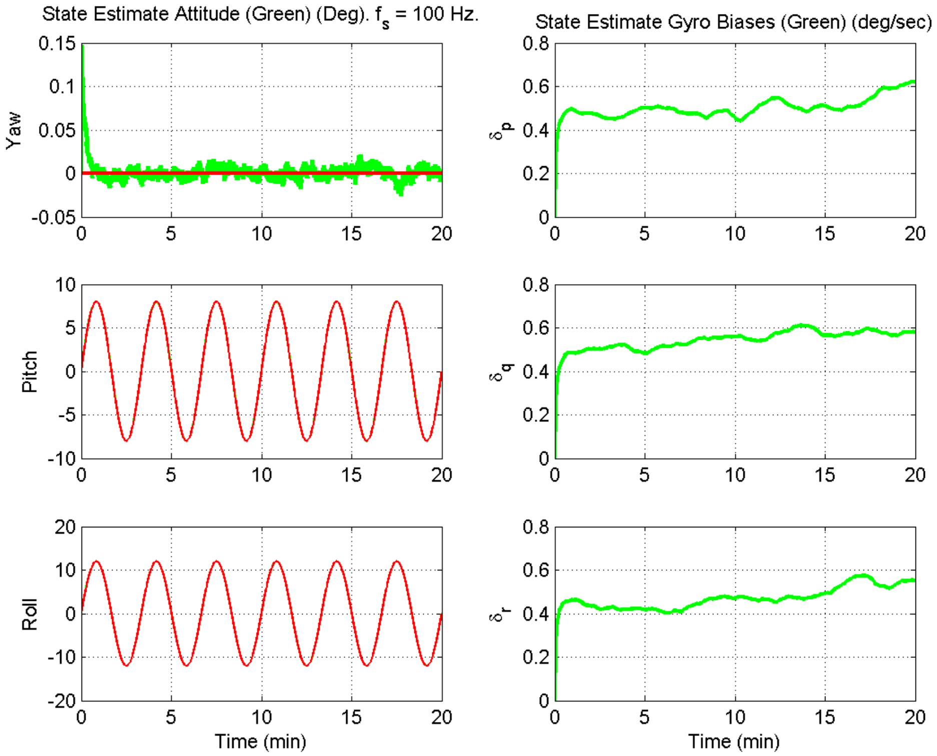

Figure 4.15: Optimal EKF Processing of the Benign Pitch and Roll Simulated Data - With Gyro Bias Errors and Without Misalignment or Gyro Sensitivity Errors (Red VICON measurements)

For comparison purposes, Figure 4.16 illustrates the case of introducing large gyro sensor axes misalignment errors into the simulated trajectory dynamics used in an optimal EKF that only expects gyro bias dynamics. Consequently, spurious gyro bias state estimate oscillations are observed. The misalignments applied are defined in Table 4.8 and expressed in the Direction Cosine Matrix (DCM) that is defined in Equation 4.8.

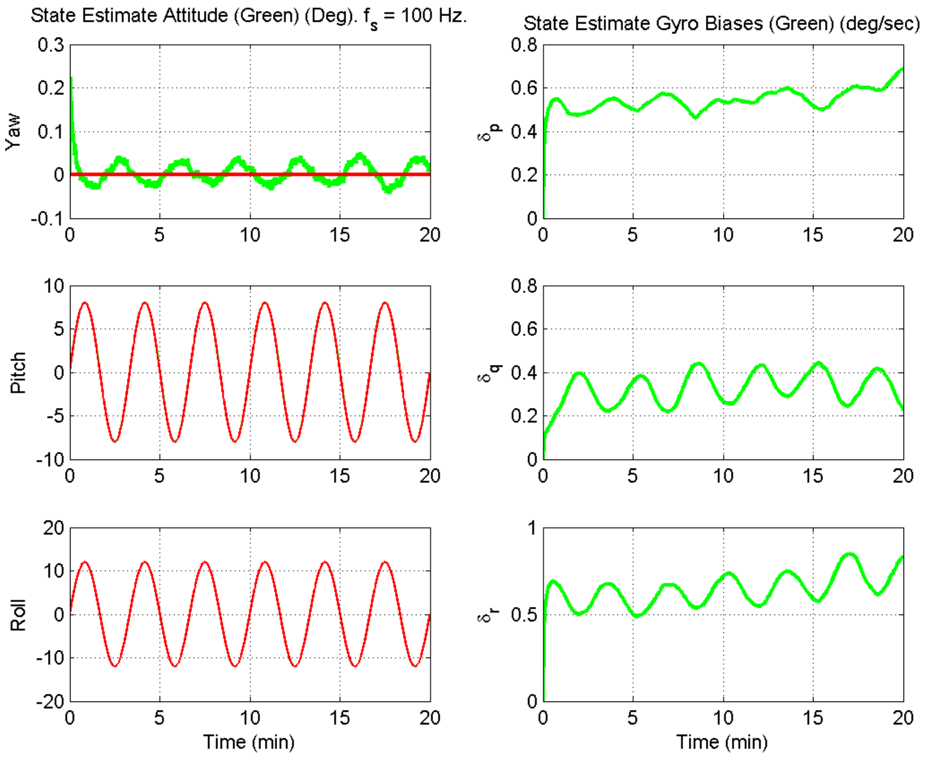

Figure 4.16: Optimal EKF Processing with Intentional Gyro Sense Axes Misalignment Error Introduced to the Benign Pitch and Roll Simulated Data (Red VICON measurements)

$$MisalignmentDCM = \begin{bmatrix} 0.9540 & 0.2556 & -0.1564 \\ -0.2753 & 0.9538 & -0.1204 \\ 0.1184 & 0.1579 & 0.9803 \end{bmatrix} \tag{4.8}$$

Yaw Error	Pitch Error	Roll Error
15°	9°	-7°

Table 4.8: Applied IMU Misalignments to Generate Gyro Bias State Estimate Oscillations

Additionally, Figure 4.17 illustrates the case of introducing large gyro sensitivity errors into the simulated trajectory dynamics and again processing the corrupted trajectory with an optimal EKF that only expects gyro bias dynamics. Similar gyro bias state estimate oscillations occur. The arbitrary gyro sensitivity errors applied are defined in Table 4.9.

Although, in theory it is possible to design an EKF to differentiate between misalignments and gyro sensitivity errors, in practice the quality of the measurements utilized and the maneuvers made by the vehicle can make it difficult to observe the differences between these two errors.

An EKF was designed with gyro sensitivity error states in order to attempt to illustrate how the oscillations in the gyro bias estimate could be reduced in certain conditions. By giving the filter additional error states the filter is able to capture the unmodeled process dynamics and prevent the gyro bias estimates from being impacted.

Yaw Gyro	Pitch Gyro	Roll Gyro
Sensitivity Error	Sensitivity Error	Sensitivity Error
1.10	1.30	1.29

Table 4.9: Applied Gyro Sensitivity Errors Used to Generate Gyro Bias State Estimate Oscillations

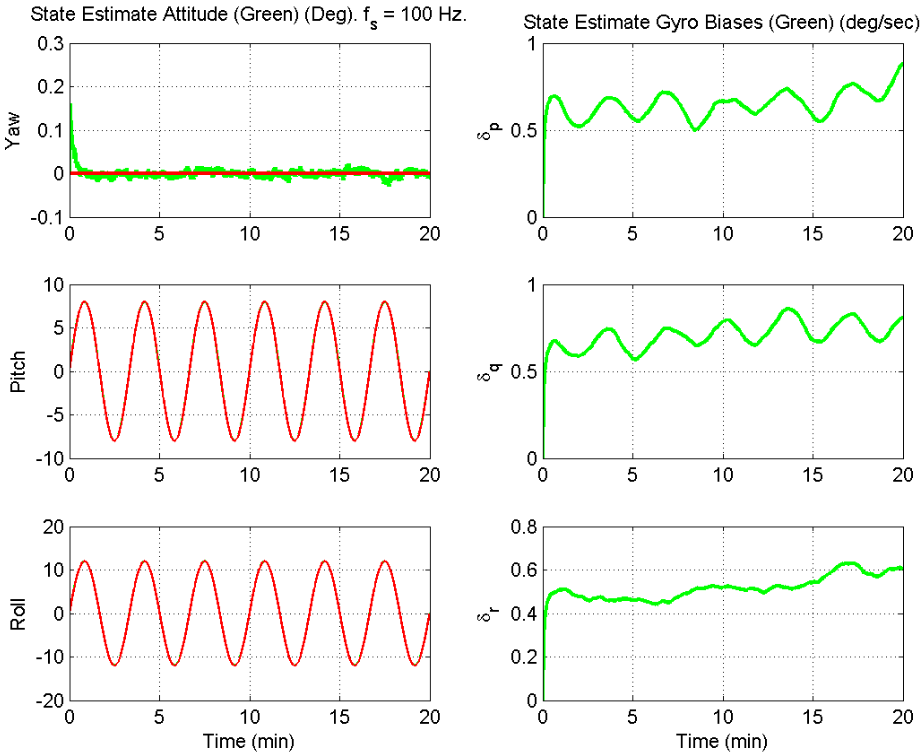

Figure 4.17: Optimal EKF Processing with Intentional Gyro Sensitivity Error Introduced to the Benign Pitch and Roll Simulated Data (Red VICON measurements)

4.6 Improving Filter Performance

As discussed in section 4.5.1 the use of real flight data highlighted a limitation of the simulation in that gyro sensitivity, misalignment, and nonorthogonality errors were not modeled in process dynamics of the filter and this resulted in oscillations in the estimate of the gyro biases due to the simplified process dynamics modeled.

In practice, this limits the usefulness of the basic EKF presented to situations in which measurement updates are always available. The situation where there is a loss of measurement updates, whether they be Vicon based or GPS based or some other form of aiding, would result in accelerated degradation of the IMU only navigation solution as the gyro bias estimates are less accurate due to the oscillations they exhibit.

The present filter implementation is essentially using the measurement update to calibrate the IMU on the fly. With the filter assigning unmodeled dynamics to gyro bias error states the IMU calibration is degraded, the bias oscillations are spurious errors added incorrectly to the state estimates. With more accurate bias estimates the attitude/heading drift under circumstances of lost aiding is diminished.

Several ways exist for eliminating these bias errors oscillation. A less then elegant option would be to increase the process noise or measurement noise modeled in the filter. This solution has the disadvantage of degrading the accuracy of the overall system.

Another option would be to improve the quality of the gyros in the case of gyro sensitivity errors or installation in the helicopter in the case of misalignment and non-orthogonality errors or both. The MEMS gyros used in the IMU are mid-range in terms of price and quality available on the commercial market. Better IMUs are available with improved gyro sensitivity error performance, but at a much greater price and often times in a larger form factor which makes mounting of the IMU on the helicopter difficult. The misalignment and non-orthogonality errors inherent in the improvised mounting of the IMU to the helicopter could certainly be improved with better mechanical design at a reasonable price.

The third option would be to incorporate additional error states in the EKF to deal with the unmodeled dynamics. It was discussed earlier that, in theory, an EKF could be set up to account for each specific error state, but in the case of the experimental setup utilized it would seem unlikely to have sufficient observability to accurately account for each error state. Another consequence of adding many additional error states would be the increased computational cost of such a filter, which is not in keeping with the spirit of the research.

Therefore, an EKF was created that only added gyro sensitivity error states to demonstrate where improvements could be made in the design of the basic filter used in this research. This was found to reduce the oscillations in the gyro bias estimates. This filter implementation also effectively rolled some of the misalignment and nonorthogonality errors into the gyro sensitivity states, as the distinction between these errors in terms of process dynamics is very subtle.

4.6.1 Addition of Gyro Sensitivity Error States

In order to add additional error states to the EKF the following equations had to be modified.

The process model presented in Equation 3.2 now includes gyro sensitivity error terms as seen in Equation 4.9.

$$\begin{aligned} \underline{\Omega}_{IMU} &= (1 + SensitivityError)\underline{\Omega}_{true} + \underline{b} + \underline{\eta} \\ \dot{\underline{b}} &= -\frac{1}{\tau_{IMU}}\underline{b} + \underline{\nu} \end{aligned} \tag{4.9}$$

The dynamic coefficient matrix of Equation 3.5 now becomes a 9-by-9 matrix with the elements defined in $\mathbf{F}(1:3, 7:9)$ equivalent to the evaluation of the right side of Equation 3.1. All other new elements in $\mathbf{F}$ are set to zero.

The coupling matrix shown in Equation 3.9 now takes on the form seen in Equation 4.10.

$$\mathbf{G} = \begin{bmatrix} 0 & \frac{\sin(\Phi)}{\cos(\Theta)} & \frac{\cos(\Phi)}{\cos(\Theta)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\Phi) & -\sin(\Phi) & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{\sin(\Phi)\sin(\Theta)}{\cos(\Theta)} & \frac{\cos(\Phi)\sin(\Theta)}{\cos(\Theta)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

The process noise vector shown in Equation 3.10 now takes on the form seen in Equation 4.11.

$$\underline{\beta} = \begin{bmatrix} \eta_z & \eta_y & \eta_x & \nu_x & \nu_y & \nu_z & \eta_{SFx} & \eta_{SFy} & \eta_{SFz} \end{bmatrix}^T \tag{4.11}$$

The initial setting of the discrete state estimation error covariance matrix, $\mathbf{P}$, is now a 9-by-9 diagonal matrix. The first three diagonal elements are set to the variance of the attitude/heading measurement noise. The next three diagonal elements are set to the variance of the gyro IMU signal noise and the last three diagonal elements are set to 10 times the variance of the gyro IMU signal noise. All off-diagonal elements are set to zero.

The greater initial uncertainty placed on the state estimation error associated with the gyro sensitivity states was found to speed up convergence of the filter in identifying the predicted gyro sensitivity errors.

The state vector shown in Equation 3.4 now takes on the form seen in Equation 4.12, where the $\hat{SF}$ terms are gyro sensitivity error terms.

$$\underline{\hat{x}} = \begin{bmatrix} \hat{\Psi} & \hat{\Theta} & \hat{\Phi} & \hat{b}_x & \hat{b}_y & \hat{b}_z & \hat{SF}_x & \hat{SF}_y & \hat{SF}_z \end{bmatrix}^T \tag{4.12}$$

The measurement sensitivity matrix shown in Equation 3.15 now takes on the form seen in Equation 4.13.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.13}$$

4.6.2 Experimental Results of Using Gyro Sensitivity Error States in the Base EKF Model Without Gain Scheduling

Using the free-flight helicopter data again for comparison purposes, the addition of gyro error sensitivity states is shown to only slightly improve the behavior of the gyro bias estimates. The estimated gyro sensitivity errors are also larger than expected for the IMU used in the testing. It is likely that misalignment errors rather than gyro sensitivity errors are the source of the bias oscillations.

More work needs to be done in determining the appropriate error model to use for in-flight calibration of the dominant IMU errors. The right mix of error process dynamics should improve system performance in the absence of the measurement update.

Additional work would need to be done to determine gain schedules for this more sophisticated filter implementation. Figure 4.18 illustrates the results of using the optimal EKF formulation with additional states for gyro sensitivity.

Figure 4.19 illustrates the estimated gyro sensitivity error states as a function of time.

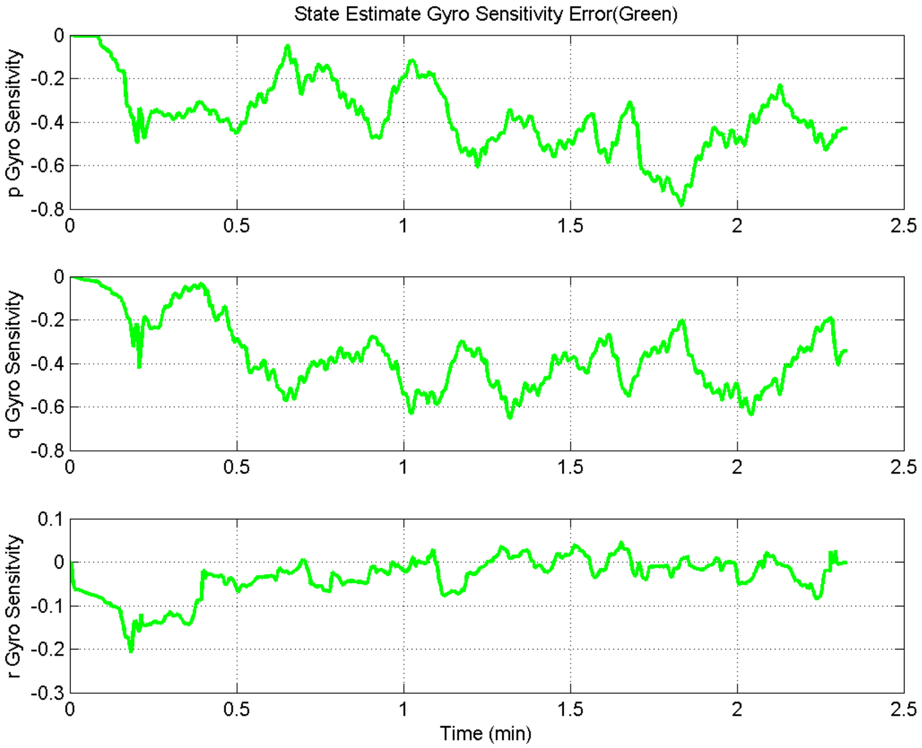

Figure 4.18: Optimal EKF Processed Helicopter Free Flight Data With Gyro Sensitivity Filter States

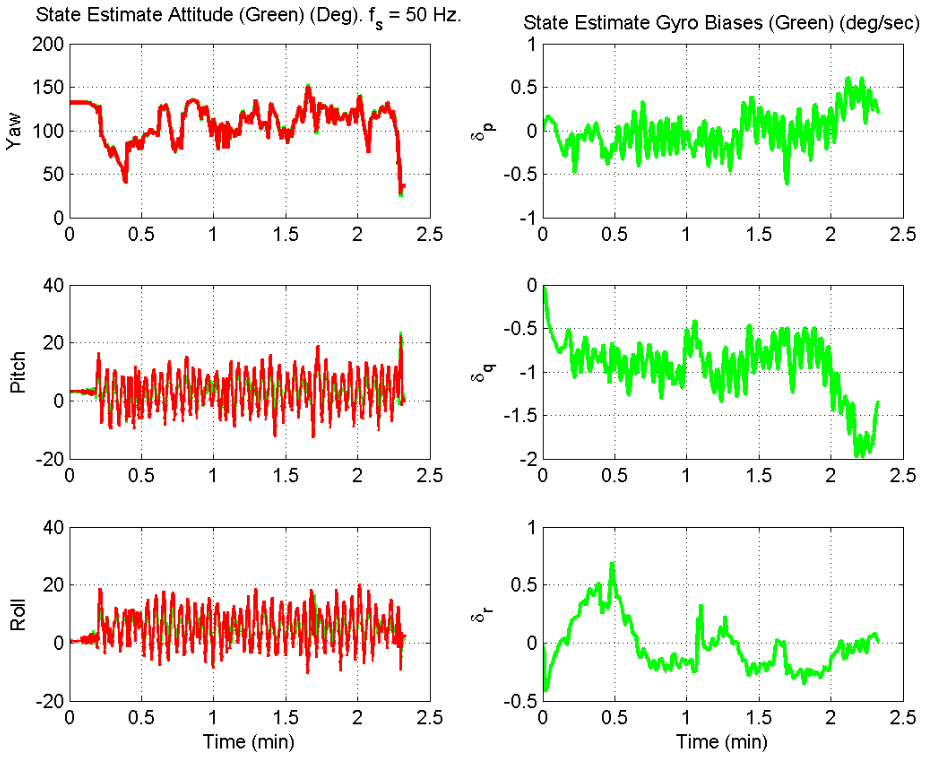

Figure 4.19: Optimal EKF Processed Helicopter Free Flight Gyro Sensitivity Error Estimates (Red VICON measurements)

Chapter 5

Conclusions

The concept of gain scheduling gains for an Extended Kalman Filter was explored. The motivation for this was to develop an attitude/heading estimation system for use in applications with limited computational resources. An attitude/heading estimation system utilizing a gain scheduled EKF was successfully demonstrated using both simulated and experimental test data.

Initially, the conventional EKF approach and implementation was presented. Next, the idea of gain scheduling was presented along with a review of historical work in the area. Three gain scheduling approaches of the attitude/heading estimating filter were developed. The first method was based on using the average optimal gain as a constant gain schedule value. The second method was a gain schedule based on fitting the same regression equation to each Kalman gain element in the matrix and using specific attitude/heading time histories to determine the coefficients of the regression. The third method was to tailor the regression equation for each specific Kalman gain element in the matrix and then fit the individual regressions to a family of similar attitude/heading time histories. In addition, a performance index was used to judge relative performance between the optimal and suboptimal filter implementations.

A series of simulated trajectories were generated to exercise the different filter approaches. The trajectories covered a range of vehicle dynamics and the simulated sensor measurements corresponding to the trajectories were corrupted by noise and gyro bias errors. The constant gain schedule approach generally gave poorer filter performance in all cases than either the non-tailored or the tailored multi-nonlinear regressions. The non-tailored regression generally gave better results than the tailored regression. Both multi-nonlinear regression approaches appeared to have numerical stability limitations. The effect of the filter's measurement update frequency on the performance index was determined in the case of the constant gain schedule approach. Less frequent measurement updates degraded the performance of the filter.

Utilizing data collected from indoor testing of a RC helicopter equipped with a MEMS IMU, and tracked by an infrared machine vision system, allowed the testing of different filter approaches using real data. Real data contained a range of MEMS gyro errors that were originally modeled in the corrupted simulated trajectories. The unmodeled errors resulted in unexpected oscillations in the gyro bias error state estimates. This connection between unmodeled gyro error dynamics and gyro bias error state estimates was confirmed through simulation. Additional simulations were conducted with intentional gyro misalignment and gyro sensitivity errors introduced into the trajectory data. These tests confirmed the source of gyro bias oscillations as likely to be caused by unmodeled process dynamics in the filter.

In an attempt to address these gyro bias oscillations, a filter was developed that included a model for gyro sensitivity error states and dynamics. This filter was able to pull out the gyro bias oscillation caused by the gyro sensitivity error dynamics in the simulated trajectories, but when used on the helicopter data it only had a minor impact. This suggests more work needs to be done in reducing these error terms in the IMU and improving IMU installation or in developing filters with alternative gyro error process dynamics.

To this end, the following areas of research are recommended:

- Develop and test alternative gyro error process models. Create a process model that allows for the identification of both gyro bias and gyro misalignment errors. Also investigate a process model the includes gyro bias, sensitivity error, and misalginment states in one model.

- Develop more sophisticated gain scheduling approaches and develop approaches to allow the filter to reject unreliable measurements. Gain scheduling approach should incorporate bounds on the computed gain used in the filter. The tailored multi-nonlinear regression's poorer performance showed the need to derive the gain schedule from a diverse range of trajectories.

- Implement a gain scheduled filter algorithm in a real-time embedded system for use in vehicles. Implementation will likely require the filter be implemented in fixed-point math to allow the algorithm to be supported on low cost, low power microcontrollers that do not include a floating-point capability.

References

[1] D. Gebre-Egziabher, R. C. Hayward, and J. D. Powell. Design of multi-sensor attitude determination systems. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(2):627–649, April 2004.

[2] T. J. Mueller and et. al. *Introduction to the Design of Fixed-Wing Micro Air Vehicles: Including Three Case Studies*. AIAA, 2007.

[3] S. Jameson and et. al. Samari nano air vehicle - a revolution in flight. Technical report, Lockheed Martin Advanced Technology Laboratories, September 2009.

[4] J. Downey, S. Hollander, and E. Rottach. A loran inertial navigation system. *Sperry Engineering Review*, 21(1):36–43, 1968.

[5] D. Kleinman and M. Athans. The design of suboptimal linear time-varying systems. *IEEE Transactions on Automatic Control*, AC-13(2):150–159, April 1968.

[6] D. Kleinman, T. Fortmann, and M. Athans. On the design of linear systems with piecewise-constant feedback gains. *IEEE Transactions on Automatic Control*, AC-13(4):354–361, August 1968.

[7] Robert A. Crotteau. Sub-optimal gain schedules for the discrete kalman filter. Master's thesis, Naval Post Graduate School, 1969.

[8] A. Stubberud and D. Wismer. Suboptimal kalman filtering techniques. In C. T. Leondes, editor, *Theory and Applications of Kalman Filtering*, volume 139, chapter 5, pages 105–118. AGARD, February 1970.

[9] T. Kobayashi, D. Simon, and Litt. J. Application of a constant gain extended kalman filter for in-flight estimation of aircraft engine performance parameters. Technical Report NASA TM-2005-213865, NASA, September 2005.

[10] P. Anderson. Air charge estimation in turbocharged spark ignition engines. Master's thesis, Linkoping University, 2005. Thesis No. 989.

[11] T. Yoo and et. al. Gain-scheduled complementary filter design for a mems based attitude and heading reference system. *Sensors*, 11:3816–3830, 2011.

[12] T. Cline and C. Triska. Suboptimization of a kalman filter with delayed-states as observables. Technical report, Engineering Research Institute, Iowa State University, October 1970.

[13] R. Asher and R. Reeves. Performance evaluation of sub-optimal filters. *IEEE Transactions on Aerospace and Electronic Systems*, 11(3):400–405, May 1975.

[14] R. M. du Plessis. *Poor Man's Explanation of Kalman Filtering or How I Stopped Worrying and Learned to Love Matrix Inversion.* Taygeta Scientific Incorporated, 1967.

[15] P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering: A Practical Approach.* AIAA, 2005.

[16] Arthur Gelb, editor. *Applied Optimal Estimation.* The MIT Press, 1974.

[17] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions.* John Wiley & Sons, Inc., 1997.

[18] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB.* Wiley-IEEE Press, 3rd edition, September 2008.

[19] J. Mendel. Computational requirements for a discrete kalman filter. *IEEE Transactions on Automatic Control*, 16(6):748–758, December 1971.

[20] Thomas Chambon. State estimation for a small-scale helicopter flying indoors. Internship report, University of Minnesota, Twin Cities, Minneapolis, MN, August 2009.

[21] N. Dadkhah, B. Mettler, and D. Gebre-Egziabher. A model-aided ahrs for micro aerial vehicle application. Technical report, University of Minnesota, Twin Cities, Minneapolis, MN, 2008.

[22] D. Gebre-Egziabher and G. H. Elkaim. Mav attitude determination from observation of earth's magnetic and gravity field vectors.

[23] Charles F. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, AC-23(3):395–404, June 1978.

[24] E-Flite. *Blade CX2*. Horizon Hobby.

[25] Analog Devices Inc. High precision tri-axis inertial sensor adis16350/adis16355, rev b., September 2009.

Appendix A

Glossary and Acronyms

Care has been taken in this thesis to minimize the use of subject matter specific terminology and acronyms, but this cannot always be achieved. This appendix defines the specific terminology used in a glossary and contains a table of acronyms and their meaning.

A.1 Glossary

- Auxiliary Variable: The variable that controls the scheduling of the Kalman filter gain. Kalman gains used by the filter change when this variable changes.
- Bias: A constant, deterministic, erroneous offset in the output of a sensor relative to its true error-free value.
- Covariance Matrix: The covariance matrix is a linear measure of how two or more random variables vary about their respective means and how they depend on each other. The diagonal elements of the matrix describe the variation of one random variable about itself. The off-diagonal elements of the matrix describe the variation between pairs of random variables.
- Estimation: Estimation attempts to determine the value of a parameter or state from measurements corrupted with noise.

- Euler angle: Euler angles are the parameters that describe the specific orientation of a rigid body through a composite sequence of rotations defined in a fixed reference frame or a frame co-moving with the rotating body.

- Extended Kalman Filter: The Extended Kalman Filter (EKF) is a modified version of the simple Kalman filter. The EKF capable of dealing with nonlinear systems and nonlinear measurements.

- Filtering: A filter is a mathematical means to reject noise from a corrupted signal containing information of interest.

- Kalman Filtering: Kalman filtering is a time domain, recursive filter that utilizes a linear measurement model and a linear dynamic mathematical model of the system being filtered to calculate an optimal estimate of the state of the system being filtered. The system and measurement are corrupted by Gaussian white noise. This is sometimes referred to as a simple Kalman Filter.

- Kalman Gain: Kalman Gain weights the difference between the current measurement of the state of the system and the current prediction of the state of the system such that in the case of an optimal estimator results in an optimal estimate of the state of the system in the future.

- Gauss-Markov noise: Gauss-Markov noise is a random signal where the value of the signal in a given instance is correlated in a finite amount time with another instance of the signal.

- Measurement: A measurement is a physical detection of a state variable of a system that is corrupted by noise. This is also referred to as an observation.

- Optimal: Optimal refers to an unbiased and minimum variance or minimum uncertainty estimate.

- Process: Process is synonymous with the equations/model that describes the dynamics of a system. An example of process is a random process which is discrete sequence of values generated from sampling a random variate.

- Recursive: A method of defining an equation such that the equation can be solved an unlimited number of times to arrive at the latest instance of the solution. For example, in business a running total plus the value of the latest transaction results in the calculation of the new total. This is the opposite of repeating all the arithmetic for every transaction that previously occurred every time a new transaction occurs. This allows the Kalman Filter to operate with a minimal amount of memory as system history does not need to be recorded, rather the current state estimate contains all the past information of the system. See Recursive.

- State variable: A state variable is mathematical parameter of a system that describes the system's dynamics in time. Knowing the state variables typically allows one to predict the future state or behavior of the system.

- System: A system is an object that can be described by a series of mathematical equations or functions. The system has the property that it is dynamic in that its state variables can be defined by mathematical equations that are time-dependent.

- White noise: A random signal where the value of the signal in a given instance is uncorrelated in time with any other instance of the signal.

A.2 Acronyms

Table A.1: Acronyms

Acronym	Meaning
AHRS	Attitude/Heading Reference System
CGEKF	Constant Gain Extended Kalman Filter
DARPA	Defense Advanced Research Projects Agency
DCM	Direction Cosine Matrix
EKF	Extended Kalman Filter
FIR	Finite Impulse Response
GPS	Global Positioning System
IGCL	Interactive Guidance and Control Lab
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LINS	LORAN Inertial Navigation System
LORAN	LOng RAnge Navigation
MAV	Micro Air Vehicle
MEMS	Micro-Electromechanical System
NAV	Nano Air Vehicle
RC	Remote Control
RMS	Root-Mean-Square
UAV	Unmanned Aerial Vehicles

Appendix B

Example Extended Kalman Filter MATLAB Code Used in Analysis (Excluding plotting functions)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Extend Kalman Filter Analysis Code Used In Thesis
%Code first calculate EKF with "optimal" varying Kalman Gains.
%Next, the Kalman gains are reduced to a constant schedule and the EKF run
%again.
%Finally, the Kalman gains are applied as a simple linear function schedule
%and the EKF run again.  A performance index is calculated
%for each suboptimal gain schedule.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
clc;
close all;

%Define some constants
d2r = pi/180; %Degress to radians conversion
```

```
r2d = 1/d2r; %Radians to degrees conversion
transientpercent=0.2;%percent of file to ignore due to transient gains

%% -----1.0  Load the Processed Data, Determine Processing  -------%
load uav_imu_data.mat;
[p n e] = fileparts(’uav_imu_data.mat’);

%% ------2.0   Define and Initialize Variables  -------------------%
imu = imu_corrupt; %Assign corrupted or real measurements to imu variable
imuerrors = 0;  %Switch if IMU errors are known (1 = on, 0 = off)
t = imu(:,1); %Extract time sequence from IMU variable
drl = length(t); %Identify the overall time duration of test data
drls = round(drl*transientpercent); %start of steady state
    %Initialize characteristics that define inertial
    %quality using external function.
insQual = ’CON’;  %Set IMU quality flag to Consumer grade
[gyro,accel] = getinsmodel(insQual);  %Get inertial sensor error model
tau_w = gyro(1);           %Markov Time Constant
sigma_wc = gyro(2);        %Markov Bias
sigma_wn = gyro(3);        %Wide Band White Noise
sigma_ns = gyro(4);        %Gyro bias/offset
L = zeros(6,3); % Initialize Kalman Gain matrix
I = eye(6); % Setup a 6 by 6 identify matrix
%Pre-allocate arrays for storing gain data over time (Row, Column)
L11 = zeros(drl, 1);
L12 = zeros(drl, 1);
L13 = zeros(drl, 1);
L21 = zeros(drl, 1);
L22 = zeros(drl, 1);
L23 = zeros(drl, 1);
L31 = zeros(drl, 1);
L32 = zeros(drl, 1);
```

```
L33 = zeros(drl, 1);
L41 = zeros(drl, 1);
L42 = zeros(drl, 1);
L43 = zeros(drl, 1);
L51 = zeros(drl, 1);
L52 = zeros(drl, 1);
L53 = zeros(drl, 1);
L61 = zeros(drl, 1);
L62 = zeros(drl, 1);
L63 = zeros(drl, 1);
P44 = zeros(drl,3);
P55 = zeros(drl,3);
P66 = zeros(drl,3);

%% -------3.0   Establish the EKF State Variable Initial Conditions --%
for j = 1:4
%Initialize state variables
eul = zeros(drl,3);  %Pre-allocate the Euler angle state variables
p_bias = zeros(drl,1); %Pre-allocate the roll gyro bias rate array
q_bias = zeros(drl,1); %Pre-allocate the pitch gyro bias rate array
r_bias = zeros(drl,1); %Pre-allocate the yaw gyro bias rate array
%Initialize Euler state variable using first attitude truth
%data-point
eul(1,:) = [psi(1) the(1) phi(1)]*d2r; %[yaw pitch roll]

%% -------3.1   Define EKF Filter Variables/Structure  ---------------%
loop_count = 0;
FILTER = 'ON';
update_limit = 0; %round(1/dt);
P = diag([0.06*d2r*ones(1,3) sigma_wc*ones(1,3)].^2);%eye(6);
H = [eye(3) zeros(3,3)];  %Define measurement matrix
F = zeros(6,6); %Initialize state transition matrix
```

```
Rwpsd = eye(6);    %Initialize process noise/spectral density matrix
Rwpsd(1,1) = 1*sigma_wn^2;    %Assign White Noise value
Rwpsd(2,2) = 1*sigma_wn^2;    %Assign White Noise value
Rwpsd(3,3) = 1*sigma_wn^2;    %Assign White Noise value
Rwpsd(4,4) = 1*sigma_wc^2/tau_w;     %Assign Gyro Makrov Bias
Rwpsd(5,5) = 1*sigma_wc^2/tau_w;     %Assign Gyro Makrov Bias
Rwpsd(6,6) = 1*sigma_wc^2/tau_w;     %Assign Gyro Makrov Bias
Rv = eye(3)*(0.06*d2r)^2;     %Initialize Measurement Noise Matrix
Fe2e = zeros(3,3);  %Initialize a portion of the state transition matrix
Fw2e = zeros(3,3);  %Initialize a portion of the state transition matrix
Fe2w = zeros(3,3); %Initialize a portion of the state transition matrix
Fw2w = -eye(3)/tau_w; %Initialize a portion of the state transition matrix
Gn2e = zeros(3,3);   % "n" = wide band noise; "c" = colored noise.
Gc2e=zeros(3,3);%Initialize a part of the process noise sensitivity matrix
Gn2w=zeros(3,3);%Initialize a part of the process noise sensitivity matrix
Gc2w = eye(3); %Initialize a part of the process noise sensitivity matrix

%% -------4.0   Begin the "Online" Attitude EKF Estimation Solution ----%
%Start of actual EKF
wB = waitbar(0,'Propagating Attitude Solution and Covariance ...');

for k = 2:drl
    waitbar(k/drl,wB);
    loop_count = loop_count + 1;

    p = imu(k-1,2); %roll_dot
    q = imu(k-1,3); %pitch_dot
    r = imu(k-1,4); %yaw_dot

    st = sin(eul(k-1,2));
    ct = cos(eul(k-1,2));
    sp = sin(eul(k-1,3));
    cp = cos(eul(k-1,3));
% ===== Fe2e:  Euler Angle Error to Euler Angle Error Block ==== %
    %sin(theta)/cos(theta)^2)*(pitch_dot*sin(phi)+yaw_dot*cos(phi))
    Fe2e(1,2) = -(st/ct^2)*(q*sp + r*cp);
    %(pitch_dot*cos(phi)-yaw_dot*cos(phi))/cos(theta)
    Fe2e(1,3) = (q*cp - r*sp)/ct;
    %-(pitch_dot*sin(phi)+yaw_dot*cos(phi))
    Fe2e(2,3) = -(q*sp + r*cp);
    %(pitch_dot*sin(phi)+yaw_dot*cos(phi))/(cos(theta)^2)
    Fe2e(3,2) = (q*sp + r*cp)/ct^2;
    %(sin(theta)/cos(theta))*(pitch_dot*cos(phi)-yaw_dot*sin(phi))
    Fe2e(3,3) = (st/ct)*(q*cp - r*sp);
% ===== Fw2e:  Gryo Bias to Euler Angle Error Block ==== %
    Fw2e(1,2) = sp/ct; %sin(phi)/cos(theta)
    Fw2e(1,3) = cp/ct; %cos(phi)/cos(theta)
    Fw2e(2,2) = cp; %cos(phi)
    Fw2e(2,3) = -sp; %-sin(phi)
    Fw2e(3,1) = 1;
    Fw2e(3,2) = sp*st/ct; %sin(phi)*sin(theta)/cos(theta)
    Fw2e(3,3) = cp*st/ct; %cos(phi)*sin(theta)/cos(theta)
% ===== Gn2e:  Gryo wide band noise to Euler Angle Error Block ==== %
    Gn2e = Fw2e;
% ===== Fw2e:  Assemble Dynamics Matrix ==== %
    F = [Fe2e    Fw2e;...
          Fe2w    Fw2w ];        %Dynamic Matirx at t = k

    G = [ Gn2e   Gc2e;...        %Process noise sensitivity matrix
           Gn2w   Gc2w];

    PHI = expm(dt*F); %Discrete Equivalent of F, State Transition matrix
    GQG = disrw(F,G,dt,Rwpsd); %Discrete Equivalent of G*Q*G', using
    %Van Loan algorithm for computing integral of a matrix exponential.
    %GQG is equivalent to discrete Q(k).
    P44(k,j)= sqrt(P(4,4));
    P55(k,j)= sqrt(P(5,5));
    P66(k,j)= sqrt(P(6,6));
    P = PHI*P*PHI' + GQG;         %State estimation error covariance
% =====  Propagare the actual Euler Angles  ==== %
     p_bias(k,1) = p_bias(k-1,1);
     q_bias(k,1) = q_bias(k-1,1);
     r_bias(k,1) = r_bias(k-1,1);

     p_use = imu_corrupt(k-1,2) + p_bias(k,1);
     q_use = imu_corrupt(k-1,3) + q_bias(k,1);
     r_use = imu_corrupt(k-1,4) + r_bias(k,1);

     sp_use = sin(eul(k-1,3));
     cp_use = cos(eul(k-1,3));
     st_use = sin(eul(k-1,2));
     ct_use = cos(eul(k-1,2));

 yaw_dot = (sp_use/ct_use)*q_use + (cp_use/ct_use)*r_use;
 pitch_dot = cp*q_use - sp_use*r_use;
 roll_dot=p_use+(sp_use*st_use/ct_use)*q_use+(cp_use*st_use/ct_use)*r_use;
 eul(k,:) = eul(k-1,:) + dt*[yaw_dot pitch_dot roll_dot];
        if (loop_count > update_limit & strcmp(FILTER,'ON'))
             loop_count = 0;
%% ----------4.1  "Online" Gain Calculations Inside Filter  -----------%
if j == 1
    L = P*H'*inv(H*P*H' + Rv); %Calculate Kalman Gain Matrix
    %%%Storing the Kalman Gain History
    L11(k) = L(1,1);
    L12(k) = L(1,2);
    L13(k) = L(1,3);
    L21(k) = L(2,1);
    L22(k) = L(2,2);
    L23(k) = L(2,3);
    L31(k) = L(3,1);
    L32(k) = L(3,2);
    L33(k) = L(3,3);
    L41(k) = L(4,1);
    L42(k) = L(4,2);
    L43(k) = L(4,3);
    L51(k) = L(5,1);
    L52(k) = L(5,2);
    L53(k) = L(5,3);
    L61(k) = L(6,1);
    L62(k) = L(6,2);
    L63(k) = L(6,3);
    %else: No action or Kalman gain computation occurs
    %for constant gain case.
    %Constant gain is calculated in 5.1 outside the loop.
elseif j == 3
    %Skip storing gains and instead solve for gains based on linear fits
L(1,1) = L1f(1)+L1f(2)*eul(k,1)+L1f(3)*eul(k,2)+L1f(4)*eul(k,3)+...
L1f(5)*eul(k,1)*eul(k,2)+L1f(6)*eul(k,2)*eul(k,3)+L1f(7)*eul(k,1)*eul(k,3);

L(2,2) = L2f(1)+L2f(2)*eul(k,1)+L2f(3)*eul(k,2)+L2f(4)*eul(k,3)+...
L2f(5)*eul(k,1)*eul(k,2)+L2f(6)*eul(k,2)*eul(k,3)+L2f(7)*eul(k,1)*eul(k,3);

L(3,3) = L3f(1)+L3f(2)*eul(k,1)+L3f(3)*eul(k,2)+L3f(4)*eul(k,3)+...
L3f(5)*eul(k,1)*eul(k,2)+L3f(6)*eul(k,2)*eul(k,3)+L3f(7)*eul(k,1)*eul(k,3);

L(4,3) = L4f(1)+L4f(2)*eul(k,1)+L4f(3)*eul(k,2)+L4f(4)*eul(k,3)+...
L4f(5)*eul(k,1)*eul(k,2)+L4f(6)*eul(k,2)*eul(k,3)+L4f(7)*eul(k,1)*eul(k,3);
```

```
L(5,2) = L5f(1)+L5f(2)*eul(k,1)+L5f(3)*eul(k,2)+L5f(4)*eul(k,3)+...
L5f(5)*eul(k,1)*eul(k,2)+L5f(6)*eul(k,2)*eul(k,3)+L5f(7)*eul(k,1)*eul(k,3);

L(6,1) = L6f(1)+L6f(2)*eul(k,1)+L6f(3)*eul(k,2)+L6f(4)*eul(k,3)+...
L6f(5)*eul(k,1)*eul(k,2)+L6f(6)*eul(k,2)*eul(k,3)+L6f(7)*eul(k,1)*eul(k,3);

elseif j == 4
    L(1,1) = (1.01954760994007e-05*(r2d*eul(k,1))^2)+...
    (1.30447333578395e-05*(r2d*eul(k,2))^2);

    L(2,2) = ((-2.41420842380300e-09*(r2d*eul(k,1)))+...
    (1.02826109779989e-05*(r2d*eul(k,1))^2)+...
    (-9.58281344616228e-08*(r2d*eul(k,2)))+...
    (1.02785172978461e-05*(r2d*eul(k,2))^2)+...
    (-1.56699421187240e-08*(r2d*eul(k,3)))+...
    (-4.15974637516097e-11*(r2d*eul(k,3))^2));

    L(3,3) = (1.01933460519859e-05*(r2d*eul(k,1))^2)+...
    (1.30447689322578e-05*(r2d*eul(k,2))^2);

    L(4,3) = (5.30659348469686e-07*(r2d*eul(k,1))^2)+...
    (4.89126466558287e-07*(r2d*eul(k,2))^2);

    L(5,2) = (0.00107311676374309+(4.16898163844490e-08*(r2d*eul(k,3)))+...
    (-1.55713050059031e-07*(r2d*eul(k,3))^2));

    L(6,1) = ((7.80777838018693e-09*(r2d*eul(k,1)))+...
    (5.28011975305490e-07*(r2d*eul(k,1))^2)+...
    (6.41090951091655e-08*(r2d*eul(k,2)))+...
    (4.87522834123705e-07*(r2d*eul(k,2))^2)+...
    (1.26635097985323e-07*(r2d*eul(k,3)))+...
    (-1.49281589143412e-07*(r2d*eul(k,3))^2));
```

```
end

%% ---------4.2  Filter Update and "Online" Gain Usage -----------------%
            %State estimation error covariance update
            P=(I-L*H)*PHI*P*PHI’*(I-L*H)’+(I-L*H)*GQG*(I-L*H)’+(L*Rv*L’);
            %P = (I - L*H)*P;   %State estimation error covariance update
            state_update = L*(att(k,:) - eul(k,:))’;  %Kalman filter
            eul(k,:)=eul(k,:)+state_update(1:3,:)’;%update state estimate
            p_bias(k,:)=p_bias(k,:)+state_update(4,:); %roll rate bias
            q_bias(k,:)=q_bias(k,:)+state_update(5,:); %pitch rate bias
            r_bias(k,:)=r_bias(k,:)+state_update(6,:); %yaw rate bias
        end
        %End of actual EKF

%% ---------5.0  "Offline" Postprocessing of Filter Results  ------------%
%Find the trace of the final state estimation error covariance matrix
%Used to calculate the performance index, mu, for rating sub-optimal
%filters

Pf(j).final = P; %Save the final value of P to a structure for later use
Jt(j) = trace(P);  %assign scalar result to an array
% ===== Save the filter state estimate history for later use ===== %
eulhist(j).hist = eul;
end

% ===== Create the Kalman Gain schedule used in Step 5.1 ==== %
%This schedule provides variables for the above section
    if j == 1 %Constant Kalman Gain case
        L = zeros(6,3); % Overwrite Kalman Gain matrix and reset to zero
        X = ones(size(eul(drls:end,1))); %Setup up the predictor matrix
            %to determine the fit corresponding to the average value.
        %Calculate the mean gains.  Store coefficient, 95% conf.
        %interval for mean, store fit residuals, store 95% conf. interval
        %for residuals, and store fit statistics like R^2 value.
        [L1fc,bint11c,r11c,rint11c,stats11c] = regress(L11(drls:end),X);
        [L2fc,bint22c,r22c,rint22c,stats22c] = regress(L22(drls:end),X);
        [L3fc,bint33c,r33c,rint33c,stats33c] = regress(L33(drls:end),X);
        [L4fc,bint43c,r43c,rint43c,stats43c] = regress(L43(drls:end),X);
        [L5fc,bint52c,r52c,rint52c,stats52c] = regress(L52(drls:end),X);
        [L6fc,bint61c,r61c,rint61c,stats61c] = regress(L61(drls:end),X);
        L(1,1) = L1fc; %Populate Gain matrix with constant gains
        L(2,2) = L2fc;
        L(3,3) = L3fc;
        L(4,3) = L4fc;
        L(5,2) = L5fc;
        L(6,1) = L6fc;
    end
    if j == 2 %Kalman Gain linear function of attitude case
        L = zeros(6,3); % Overwrite Kalman Gain matrix and reset to zero
        %Restore the state vector history from the optimal run
        eulOG = eulhist(1).hist; %Write structure values to the variable
        %Calculate multi-nonlinear regression fits for gains based on attitude
        %state estimates.  Regression includes interaction terms between
        %attitudes.
        %Setup predictor matrix
        X = [ones(size(eulOG(drls:end,1))) eulOG(drls:end,1)...
            eulOG(drls:end,2)...
            eulOG((drls:end),3) eulOG(drls:end,1).*eulOG(drls:end,2)...
            eulOG(drls:end,2).*eulOG(drls:end,3)...
            eulOG(drls:end,1).*eulOG(drls:end,3)];
        %Perform regression.  Store coefficients, 95% conf. interval for
        %coefficients, store fit residuals, store 95% conf. interval
        %for residuals, and store fit statistics like R^2 value.
```

```
        %stats = [R^2, F statistic, p-value, and error variance]
        [L1f,bint11,r11,rint11,stats11] = regress(L11(drls:end),X);
        [L2f,bint22,r22,rint22,stats22] = regress(L22(drls:end),X);
        [L3f,bint33,r33,rint33,stats33] = regress(L33(drls:end),X);
        [L4f,bint43,r43,rint43,stats43] = regress(L43(drls:end),X);
        [L5f,bint52,r52,rint52,stats52] = regress(L52(drls:end),X);
        [L6f,bint61,r61,rint61,stats61] = regress(L61(drls:end),X);
    end
end
%% Calculate EKF Performance Index
% Index of performance for constant gain
mu2 = (Jt(2)-Jt(1))/Jt(1);
% Index of performance for linear function gain
mu3 = (Jt(3)-Jt(1))/Jt(1);
% Index of performance for
mu4 = (Jt(4)-Jt(1))/Jt(1);

save
```

Appendix C

Summary Gain Schedules Generated From Test Trajectories

Three gain schedules were analyzed in this thesis:

1. A constant gain, a_1, computed from the time-series average gain as predicted by the optimal filter equations. These gains are donated by subscript 1 in the table.

$$L(row, column)_1 = a_1 \tag{C.1}$$

2. A gain function based on a multi-nonlinear regression of the optimal Kalman gain versus Euler angles, computed using the following Equation C.2. These gains are donated by subscript 2 in the table.

$$L(row, column)_2 = a_1 + a_2\Psi + a_3\Theta + a_4\Phi + a_5\Psi\Theta + a_6\Psi\Phi + a_7\Theta\Phi \tag{C.2}$$

3. A series of gain functions based on a multi-nonlinear regression of the optimal Kalman gain versus Euler angle. The regression function for each Kalman gain term is tailored to fit the shape of a collection optimal Kalman gain time histories produced by a family of similar trajectories. Equations C.3 to C.8 give the specific equation fitted for each particular Kalman gain. These gains are donated by subscript 3 in the table.

$$L(1,1)_3 \quad = \quad a_1\Psi^2 + a_2\Theta^2 \tag{C.3}$$

$$L(2,2)_3 = a_1\Psi + a_2\Psi^2 + a_3\Theta + a_4\Theta^2 + a_5\Phi + a_6\Phi^2 \tag{C.4}$$

$$L(3,3)_3 = a_1\Psi^2 + a_2\Theta^2 \tag{C.5}$$

$$L(4,3)_3 = a_1\Psi^2 + a_2\Theta^2 \tag{C.6}$$

$$L(5,2)_3 = a_1 + a_2\Phi + a_3\Phi^2 \tag{C.7}$$

$$L(6,1)_3 = a_1\Psi + a_2\Psi^2 + a_3\Theta + a_4\Theta^2 + a_5\Phi + a_6\Phi^2 \tag{C.8}$$

What follows is a table containing the calculated gains for each trajectory simulated in this thesis as well as the goodness of fit measure R^2. This goodness of fit indicates how well the fitted equation explains the variation seen in the data. R^2 can take on a value from 0 to 1, with values close to 1 indicating a good match between the fitted regression equation and the data analyzed.

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	0.021	-1.786e-4	0.002	-0.001	-7.510	0.005	5.006	0.99
$L(2,2)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.021	-1.795e-4	2.36e-6	1.68e-6	0.0692	-4.905e-7	-0.046	0.80
$L(3,3)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	0.021	-4.425e-4	0.002	-0.002	-19.216	0.005	12.816	0.99
$L(4,3)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	0.001	-2.65e-5	2.44e-5	-1.48e-5	-0.276	-9.323e-5	0.185	0.99
$L(5,2)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	0.001	1.25e-4	2.57e-4	-1.71e-4	-1.712	-7.99e-4	1.145	0.99
$L(6,1)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	0.001	1.87e-4	3.43e-4	-2.27e-4	-2.6141	-8.93e-4	1.742	0.99

Table C.1: Gain Schedule For Simulated Trajectory of Benign Pitch and Roll Dynamics

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.023	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	0.023	-2.39	1.576e-4	-1.291	8.444	6.68	1.0e4	0.11
$L(2,2)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.021	3.99e-4	-5.31e-8	5.02e-4	-0.003	-0.002	-5.4	0.12
$L(3,3)_1$	0.023	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	0.023	-2.51	2.34e-4	-1.23	8.55	6.69	11	0.12
$L(4,3)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	0.001	0.03	3.83e-7	0.021	-0.118	-0.092	-135	0.10
$L(5,2)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	0.001	-4.87e-6	-2.316e-9	2.33e-5	-1.31e-4	-1.19e-4	-0.29	0.13
$L(6,1)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	0.001	0.03	2.78e-7	0.025	-0.125	-0.100	-155	0.10

Table C.2: Gain Schedule For Simulated Trajectory of Aggressive Pitch Dynamics

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.023	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	0.023	1.23	-2.06e-5	5.76e-5	2.33	-0.004	0.37	0.20
$L(2,2)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.021	-3.7e-4	6.22e-7	-4.24e-7	5.33e-4	1.26e-6	-3.14e-4	0.45
$L(3,3)_1$	0.024	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	0.023	1.25	5.40e-5	6.05e-5	2.38	-0.004	0.38	0.18
$L(4,3)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	0.001	-0.02	2.99e-6	-6.83e-7	-0.03	6.37e-5	-0.006	0.24
$L(5,2)_1$	9.12e-4	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	9.44e-4	0.06	-3.69e-6	2.44e-6	0.14	2.27e-4	0.028	0.18
$L(6,1)_1$	8.71e-4	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	9.12e-4	0.05	-1.43e-7	1.14-6	0.10	2.88e-4	0.017	0.41

Table C.3: Gain Schedule For Simulated Trajectory of Aggressive Out of Phase Pitch and Roll Dynamics

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.024	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	0.021	0.058	0.05	-0.108	-0.139	0.064	0.084	0.99
$L(2,2)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.021	1.40e-4	1.15e-4	-2.57e-4	-0.003	4.37e-4	0.002	0.30
$L(3,3)_1$	0.024	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	0.021	0.021	0.167	-0.187	-0.201	0.025	0.185	0.99
$L(4,3)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	0.001	-9.36e-4	0.0031	-0.002	-0.002	-0.001	0.003	0.99
$L(5,2)_1$	8.932e-4	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	0.001	-5.76e-4	0.005	-0.004	-0.0015	-0.002	0.003	0.99
$L(6,1)_1$	8.961e-4	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	0.001	-1.64e-4	0.009	-0.009	-0.002	-0.002	0.003	0.99

Table C.4: Gain Schedule For Simulated Trajectory of Aggressive In-Phase Pitch, Roll, and Yaw Dynamics

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.024	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	-0.062	0.055	0.049	0.055	-0.199	-0.199	-0.169	0.99
$L(2,2)_1$	0.021	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.021	-3.68e-4	-3.33e-4	-3.67e-4	-2.0e-4	-2.0-4	-1.8e-4	0.88
$L(3,3)_1$	0.024	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	-0.067	0.055	0.049	0.055	-0.208	-0.208	-0.177	0.99
$L(4,3)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	8.34e-4	3.53e-5	3.33e-5	3.53e-5	-4.0e-4	-3.9e-4	-4.9e-4	0.99
$L(5,2)_1$	9.05e-4	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	0.002	-9.55e-4	-8.57e-4	-9.54e-4	0.002	0.003	0.003	0.99
$L(6,1)_1$	8.96e-4	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	0.003	-0.001	-0.001	-0.001	0.004	0.005	0.004	0.99

Table C.5: Gain Schedule For Simulated Trajectory of Aggressive Out of Phase Pitch, Roll, and Yaw Dynamics

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.03	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	0.03	1.59e-4	0.017	0.001	-0.007	-0.033	5.29e-4	0.58
$L(2,2)_1$	0.03	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.03	3.67e-5	-0.002	1.37e-4	7.0e-4	0.007	-1.76e-4	0.41
$L(3,3)_1$	0.03	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	0.028	9.91e-4	0.016	0.011	-0.007	-0.028	-0.006	0.43
$L(4,3)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	0.001	4.48e-6	-0.002	-3.84e-4	5.28e-4	0.004	1.38e-4	0.31
$L(5,2)_1$	0.001	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	0.001	-1.64e-5	-0.001	-6.88-4	5.19e-4	0.003	2.89e-4	0.28
$L(6,1)_1$	0.002	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	0.002	1.27e-5	-3.93e-5	-6.85e-5	2.05e-5	7.51e-4	-2.09e-5	0.24

Table C.6: Gain Schedule For Recorded Pendulum Helicopter Trajectories

Gain	a_1	a_2	a_3	a_4	a_5	a_6	a_7	R^2
$L(1,1)_1$	0.031	-	-	-	-	-	-	≈ 0
$L(1,1)_2$	0.029	-4.95e-5	-0.001	8.65e-4	8.88e-6	-0.015	2.19e-5	0.21
$L(2,2)_1$	0.029	-	-	-	-	-	-	≈ 0
$L(2,2)_2$	0.029	-2.34e-5	-5.29-e5	1.0e-4	2.185e-5	-0.002	-1.62e-5	0.52
$L(3,3)_1$	0.030	-	-	-	-	-	-	≈ 0
$L(3,3)_2$	0.031	-4.41e-5	-7.0e-4	9.99e-4	-2.74e-5	-0.025	-1.72e-5	0.47
$L(4,3)_1$	0.002	-	-	-	-	-	-	≈ 0
$L(4,3)_2$	0.002	-1.76e-5	-4.59e-5	-5.87e-4	1.48e-5	-0.001	-3.73e-6	0.68
$L(5,2)_1$	0.002	-	-	-	-	-	-	≈ 0
$L(5,2)_2$	0.002	-1.76e-5	-2.68e-5	6.34e-5	1.23e-5	-7.56e-4	-5.58e-6	0.72
$L(6,1)_1$	0.002	-	-	-	-	-	-	≈ 0
$L(6,1)_2$	0.002	-4.09e-7	4.92e-5	-8.87e-6	-3.66e-6	-5.47e-4	1.64e-6	0.13

Table C.7: Gain Schedule For Recorded Free-Flight Helicopter Trajectories

Gain	a_1	a_2	a_3	a_4	a_5	a_6	R^2
$L(1,1)_3$	1.019e-5	1.304-5	-	-	-	-	99.51
$L(2,2)_3$	-2.414e-9	1.028e-5	-9.583-e8	1.028e-5	-1.567e-8	-0.002	79.78
$L(3,3)_3$	1.0193e-5	1.304e-5	-	-	-	-	99.23
$L(4,3)_3$	5.307e-7	4.891e-7	-	-	-	-	99.59
$L(5,2)_3$	1.073e-3	4.169e-8	-1.557e-7	-	-	-	99.98
$L(6,1)_3$	7.808e-9	5.280e-7	6.411e-8	4.875e-7	1.266e-7	1.493e-7	99.71

Table C.8: Tailored Gain Schedule Based Off Family Of Trajectories