IOWA STATE UNIVERSITY Digital Repository

Graduate Theses and Dissertations

Graduate College

2011

Improved rate and angle estimation through higher accuracy planar motion models

Keegan Gartner Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/etd Part of the <u>Mechanical Engineering Commons</u>

Recommended Citation

Gartner, Keegan, "Improved rate and angle estimation through higher accuracy planar motion models" (2011). *Graduate Theses and Dissertations*. 10359. http://lib.dr.iastate.edu/etd/10359

This Thesis is brought to you for free and open access by the Graduate College at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Improved rate and angle estimation through higher accuracy planar motion models

by

Keegan Gartner

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Mechanical Engineering

Program of Study Committee: Greg Luecke, Major Professor Gloria Starns Peter Sherman

Iowa State University

Ames, Iowa

2011

Copyright © Keegan Gartner, 2011. All rights reserved.

TABLE OF CONTENTS

LIST (OF TABLES	iv
LIST (OF FIGURES	v
ACKN	NOWLEDGEMENTS	vii
ABST	RACT	viii
CHAP	TER 1. Introduction	1
CHAP	TER 2. Kalman Filter Development	3
2.1	Linear Kalman Filter Development	3
	2.1.1 Summarized Process	6
	2.1.2 Computation Considerations	6
2.2	Extended Kalman Filter Development	7
CHAP	TER 3. One Degree of Freedom Attitude Estimation Kalman Filter	9
CHAP 3.1	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations	9 10
CHAP 3.1 3.2	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations	9 10 12
CHAP 3.1 3.2 3.3	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations	9 10 12 14
CHAP 3.1 3.2 3.3 3.4	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations	9 10 12 14 14
CHAP 3.1 3.2 3.3 3.4	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations	9 10 12 14 14 15
CHAP 3.1 3.2 3.3 3.4	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations	9 10 12 14 14 15 18
CHAP 3.1 3.2 3.3 3.4 3.5	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations Measurement Equation Initial Conditions Simulation 3.4.1 Simulation data sets 3.4.2 Simulation results	9 10 12 14 14 15 18 20
 CHAP 3.1 3.2 3.3 3.4 	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations Measurement Equation Initial Conditions Simulation 3.4.1 Simulation data sets 3.4.2 Simulation results Experiment 3.5.1 Test Setup	 9 10 12 14 14 15 18 20 21
 CHAP 3.1 3.2 3.3 3.4 	PTER 3. One Degree of Freedom Attitude Estimation Kalman Filter Process Equations Measurement Equation Initial Conditions Simulation 3.4.1 Simulation data sets 3.4.2 Simulation results Simulation results 3.5.1 Test Setup 3.5.2 Sensor noise analysis	 9 10 12 14 14 15 18 20 21 21

CHAP	TER 4	4. Attitude Estimation with Radius Estimation	26
4.1	Measu	rement and Process Equations	27
4.2	Simula	ation	29
	4.2.1	Simulation results	31
4.3	Exper	iment	36
	4.3.1	Noise analysis	36
	4.3.2	Filter results	38
	4.3.3	Convergence	40
	4.3.4	Zero radius experiments with radius estimation filter	41
CHAP	TER 5	5. Conclusion	42

LIST OF TABLES

Table 3.1	Simulation motion parameters for basic Kalman filter	15
Table 3.2	Simulation RMS error results for basic Kalman filter	20
Table 3.3	Experimental motion parameters and computed noise variances for the	
	basick Kalman filter.	23
Table 3.4	Experiment RMS error results for expanded state Kalman filter	24
Table 4.1	Simulation motion parameters for the expanded state Kalman filter.	29
Table 4.2	Simulation RMS error results for expanded state Kalman filter	36
Table 4.3	Experimental motion parameters and computed noise variances for the	
	expanded state Kalman filter	37
Table 4.4	Experiment RMS error results for expanded state Kalman filter	38
Table 4.5	Experiment RMS error results for expanded state Kalman filter used	
	with inline motion profiles	41

LIST OF FIGURES

Figure 3.1	System definition diagram	9
Figure 3.2	Position computed from noisy rate gyro measurment	16
Figure 3.3	Position computed from noisy accelerometer measurement	16
Figure 3.4	Kalman filter angle result for simulation 1	18
Figure 3.5	Kalman filter rate result for simulation 1	18
Figure 3.6	Kalman filter angle error for simulation 1	19
Figure 3.7	Kalman filter rate error for simulation 1	20
Figure 3.8	IMU signal processing chain	21
Figure 3.9	Power spectrum density of the rate gyro error	22
Figure 3.10	Power spectrum density of the rate gyro error at f_{filter}	23
Figure 3.11	Angle and rate error for Experiment 1	24
Figure 3.12	Angle and Rate error for Experiment 1 with phase shifted sensor data.	25
Figure 4.1	System definition with normal and tangential acceleration components.	26
Figure 4.2	Reference angle vs angle computed directly from accelerometer mea-	
	surement.	29
Figure 4.3	Simulation 5 reference and estimated angle comparison	31
Figure 4.4	Simulation 5 reference and estimated rate comparison	31
Figure 4.5	Simulation 5 angle error	32
Figure 4.6	Simulation 5 rate error	32
Figure 4.7	Rate covariance comaprison for radius estimation filter and inline filter.	33
Figure 4.8	Radius estimate for all simulation.	34
Figure 4.9	Radius estimate covariance.	35

Figure 4.10	Spectral power response for rate gyro sampled at 204.8Hz	37
Figure 4.11	Experiment 5 angle error	38
Figure 4.12	Experiment 5 rate error	39
Figure 4.13	Radius estimate for experiments with radius	40

ACKNOWLEDGEMENTS

There are a few key individuals whom I would like to thank. First my advisor, Dr. Greg Luecke, who brought me on as a student and gave me countless opportunities to build, learn, and explore. Second my parents, Ambrey and Sharon, and family, who support me in everything I do.

Most importantly I would like to thank my wife Melissa, whose smile is contageous and gives everything I do meaning.

ABSTRACT

A large number of attitude estimation algorithms assume sensor placement along the axis of rotation. These models assume that all measured accelerations are due to gravity, while in reality this is not always the case. By expanding the system model to accommodate centrifugal and tangential accelerations the attitude can be more accurately estimated. This thesis introduces an approach to model accelerations not caused by gravity. This model is incorporated into a Kalman filter and then compared with the more conventional Kalman filter approach which assumes all system acceleration is due to gravity.

CHAPTER 1. Introduction

Research in the area of attitude determination is a maturing field with many established technologies and methodologies[1]. Strapdown and gimbaled attitude, heading, reference systems (AHRS) have historically relied on large, precise, and predominately mechanical, instruments to measure rate and acceleration [8]. Modern devices, such as laser ring and fiber optic gyros have largely replaced mechanical gyros to provide high accuracy measurements. While microelectromechanical (MEM) devices have appeared, providing a low cost, lower performance means to measure the same inertial parameters. These AHRS systems combine various inertial measurements from rate gyros, accelerometers, and sometimes magnetic compasses to compute complete roll, pitch, and heading information through Kalman filtering, an optimal method of combining multiple sensor measurements.

MEMs devices offer a low cost alternative to traditional mechanical components. Three rate gyros and three accelerometers can be combined to form an inertial measurement unit (IMU) which is capable of measuring roll and pitch. It is common to see this configuration used in research. Also, because MEMs type sensors are small and lightweight they are now being used in portable applications such as body tracking [5] and heads up displays [2]. For these applications it is common to see accuracy requirements compromised in favor of responsiveness. Other work focuses on the development of efficient and cost effective means of sensor fusion [7] [10] to maintain high accuracy while reducing the computational footprint.

Some prior work in the area of inertial attitude determination focuses on novel ways to utilize IMU data, some fusing accelerometer and rate gyro data [11], while other work focuses on compensating for rate gyro bias error [12] [9] for heading estimations. Other authors present methods to derive a complete solution for roll, pitch, and yaw by fusing accelerometer, rate

1

gyro, and magnetometer data, as is done by Sabatini [7]. In some cases rate gyro data is fused with other sensor readings, such as sun trackers for spacecraft applications. This fusion is performed by Kim [4] who compares sensor fusion methods.

The higher accuracy model presented in Chapter 4 accounts for off axis rotations by estimating tangential and normal accelerations sensed at the IMU. This is useful for applications where the sensor is not co-located with the center of rotation. For example, an aircraft performing a turn rotates about a point not on the aircraft, making it impossible to mount the sensor at the center of rotation. When estimating the angle of a thigh or calf in prosthetic applications, [6] it may simply not be practical to place the sensor on the hip or knee, so some compensation must be made to accommodate the off axis sensor location. Many models assume all accelerations sensed are due to gravity, but when a sensor is not co-located with the axis of rotation additional acceleration components are present and contribute errors to this incomplete model. Depending on the speed of the motion this error can be significant.

The work presented here begins with a derivation of the Kalman filter equations and then moves into the extended Kalman filter equation set. These generalized equations are then applied to a simple one degree of freedom attitude estimating filter. This basic filter assumes the sensor is inline with the axis of rotation. Simulation and experimental data are then used to test the performance of the filter under given test conditions. Results are presented and discussed that show the successful convergence of the filter along with performance data. Next a higher accuracy model is presented, which accounts for off axis rotation The Kalman filter equations are once again applied and more simulation and experimental data is used to compare the performance of the new model with the old model.

The simulation and experimental data provide a comparison of the performance of the two filter designs over a range of operating conditions, showing that it is usually only beneficial to use the higher accuracy model for 'fast' systems.

CHAPTER 2. Kalman Filter Development

Before the discussion on different implementations of the Kalman filter begins, some background is presented. Simply put, the Kalman filter is an algorithm that combines the model of a system with known inputs and measurements to obtain an optimal estimate of the system states. Often times, these states represent physical quantities. An error term is generated when comparing the values of measured terms to the expected values of those measured terms based on filter estimates. These error terms are used to adjust the model states for more accurate prediction.

The most basic version of the filter is presented first, it is then expanded to accommodate non linearities.

2.1 Linear Kalman Filter Development

Equation 2.1 introduces the linear process which is used in the Kalman filter. The derivation begins with the assumption that the system being filtered can be modeled in this form. This equation propagates the state vector, \mathbf{x} , from one time step to the next via $\boldsymbol{\Phi}$, the state transition matrix. Inputs into the system are represented by the vector \mathbf{u} . \mathbf{w} is a vector of white noise processes which accommodate error in the process model.

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k \tag{2.1}$$

Similarly the measurement process is presented. The vector \mathbf{z} holds the measured values and \mathbf{H} is a matrix which relates the measured values to the system states. Like \mathbf{w} , \mathbf{v} is a vector of white noise processes which represent the noise of the measurement. The subscript k indicates

that their respective symbols are for the current time step, k.

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.2}$$

The process and measurement equations aim to model the system perfectly. However, due to modeling error and measurement noise the states of the ideal system are generally not known exactly. The filter model instead makes an estimate of the states. The model of the system is represented by placing a hat above the appropriate vector, as in $\hat{\mathbf{x}}_k$.

At each time step the process model generates a state estimate based on the prior time step information. This is the a priori estimate, $\hat{\mathbf{x}}_k^-$. If the process model were perfect and there was no random component this would be the state vector for the new time step. However, the model is not perfect and there is random noise introduced, so a compensation must be made based on the difference between the measurement and model output. This difference, \mathbf{y} , known as the innovation, is scaled by the Kalman gain, \mathbf{K}_k . The formation of \mathbf{K}_k will be shown in more detail shortly. The relation is described mathematically in Equation 2.3.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y} \tag{2.3}$$

where the innovation is computed by

$$\mathbf{y} = \mathbf{z}_k - \mathbf{z}_k^- = \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^- \tag{2.4}$$

The goal of the filter is to minimize the error between the true state vector and the estimated state vector, or state vector error. This is done by minimizing the variance of the state vector error. This error term is defined in Equation 2.5.

$$\mathbf{e}_k \triangleq \mathbf{x}_k - \mathbf{\hat{x}}_k \tag{2.5}$$

Equation 2.5 is then used to define \mathbf{P}_k , the error covariance matrix in Equation 2.6.

$$P_k = E[\mathbf{e}_k \mathbf{e}_k^T] \tag{2.6}$$

The elements along the diagonal of \mathbf{P}_k represent the variances of the system states. Minimizing these variances also minimizes the error term, \mathbf{e}_k . The development of the optimal \mathbf{K}_k matrix

for Equation 2.3 minimizes the variance and error terms based on the statistics of the noise vectors, \mathbf{w} and \mathbf{v} .

By expanding Equation 2.6, combining the result with Equation 2.3, and setting its derivative to zero, the optimal \mathbf{K}_k value can be obtained by using Equation 2.7:

$$\mathbf{K}_{k} = \mathbf{P}_{k}^{-} \mathbf{H}_{k}^{T} (\mathbf{H}_{k} \mathbf{P}_{k}^{-} \mathbf{H}_{k}^{T} + \mathbf{R}_{k})^{-1}$$
(2.7)

Where \mathbf{R}_k is the measurement covariance matrix and \mathbf{P}_k^- is the a prior error covariance matrix.

 \mathbf{R}_k is defined in Equation 2.8.

$$\mathbf{R}_k = E[\mathbf{v}_k \mathbf{v}_k^T] \tag{2.8}$$

In a system where the noise for each measurement is independent, \mathbf{R}_k is a constant diagonal matrix with the variances of the processes along the diagonal.

The a priori error covariance matrix is defined in Equation 2.9. It provides the statistical variance of the prior estimate, \mathbf{x}_k^- , which has already been discussed. The definition of \mathbf{P}_k^- is similar to \mathbf{P}_k , but is shown for completeness.

$$\mathbf{P}_{k}^{-} = E[\mathbf{e}_{k}^{-}\mathbf{e}_{k}^{-T}]$$
(2.9)

Where \mathbf{e}_k^- is defined as

$$\mathbf{e}_k^- \triangleq \mathbf{x}_k^- - \hat{\mathbf{x}}_k^- \tag{2.10}$$

 \mathbf{P}_k^- is determined in a similar manner to \mathbf{x}_k^- . It is based on the previous time step's error covariance matrix and is projected ahead via the system's process equation. Its derivation starts by rewriting \mathbf{e}_k^- in terms of \mathbf{e}_{k-1} :

$$\mathbf{e}_{k}^{-} = \mathbf{\Phi}_{k-1}\mathbf{e}_{k-1} + \mathbf{w}_{k-1} \tag{2.11}$$

By substituting \mathbf{e}_k^- into Equation 2.9 one obtains:

$$\mathbf{P}_{k}^{-} = \mathbf{\Phi}_{k-1} \mathbf{P}_{k-1} \mathbf{\Phi}_{k-1}^{T} + \mathbf{Q}_{k}$$
(2.12)

Where \mathbf{Q}_k is the process equation equivalent of \mathbf{R}_k , defined by:

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] \tag{2.13}$$

2.1.1 Summarized Process

A summary is provided as an overview of the process. The algorithm can be divided into two phases, the Predict phase and the Update phase. The Predict phase happens first, using data from the last time step to compute the a priori estimates. Once computed, the a priori estimates are used in the Update phase to compute \mathbf{K}_k . Next the optimal estimates, \mathbf{x}_k and \mathbf{P}_k are computed, completing the cycle.

Predict

$$\hat{\mathbf{x}}_{k}^{-} = \boldsymbol{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}$$
(2.14)

$$\mathbf{P}_{k}^{-} = \mathbf{\Phi} \mathbf{P}_{k-1} \mathbf{\Phi}_{k-1}^{T} + \mathbf{Q}_{k-1}$$
(2.15)

Update

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \tag{2.16}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{2.17}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^-)$$
(2.18)

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \tag{2.19}$$

In a linear time invariant (LTI) system Φ , G, H, Q, and R are constant matrices and their k indicies are dropped. Φ , G, and H are dependent on the system dynamics while Q and R are based on the covariance of the process and measurement noise respectively. The equations also imply two required initial conditions, $\hat{\mathbf{x}}_{k-1}$ and \mathbf{P}_{k-1} . The linear filter will converge for any initial estimates, but values chosen close to true initial state vector provide less initial error and converge faster. These values must be chosen by the designer.

2.1.2 Computation Considerations

The implementation of the algorithm is typically not trivial. Matrix math is time intensive, and the use of floating point numbers is sometimes required for numerical stability. The most computationally expensive step in the algorithm is computing the Kalman gain which requires computing an mxm matrix inverse, where m is the length of the measurement vector. For LTI systems the Kalman gain converges as the filter runs. The converged Kalman gain, **K**, can then be used instead of \mathbf{K}_k . Making this substitution eliminates the need to recompute the Kalman gain at each time step which greatly reduces the computational effort required to run the filter. Unfortunately for nonlinear systems this type of substitution does not work as the Kalman gain never converges to a static matrix.

2.2 Extended Kalman Filter Development

Equations 2.14 through 2.19 represent the algorithm for a LTI system. More specifically, the $\boldsymbol{\Phi}$ and \mathbf{H} matrices are required to be linear and constant. The extended Kalman filter provides a means to minimize the MSE for nonlinear systems.

The extended Kalman filter still operates as a linear filter, but accommodates nonlinear systems by linearizing the system around some operating point, \mathbf{x}^* . The system process and measurement equations must now be modeled in the following form:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{u}$$

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v}$$
(2.20)

These equations are then linearized using a Taylor series expansion about the operating point, \mathbf{x}^* :

$$\dot{\mathbf{x}} \approx f(\mathbf{x}^*) + \frac{\delta \mathbf{f}}{\delta \mathbf{x}} \Delta \mathbf{x} + \mathbf{u}$$

$$\mathbf{z} \approx h(\mathbf{x}^*) + \frac{\delta \mathbf{h}}{\delta \mathbf{x}} \Delta \mathbf{x} + \mathbf{v}$$
(2.21)

Where $\Delta \mathbf{x}$ and $\dot{\Delta \mathbf{x}}$ are:

$$\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}^*$$

$$\dot{\Delta \mathbf{x}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}^*$$
(2.22)

Combining Equations 2.20, 2.21, and 2.22 yields the linearized system in terms of $\Delta \mathbf{x}$:

$$\dot{\mathbf{x}} = \frac{\delta \mathbf{f}}{\delta \mathbf{x}} \Delta \mathbf{x} + \mathbf{u}$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}^*) + \frac{\delta \mathbf{h}}{\delta \mathbf{x}} \Delta \mathbf{x} + \mathbf{v}$$
(2.23)

These equations do not simply replace the system in Equations 2.1 and 2.2. Instead of directly computing the best estimate, $\hat{\mathbf{x}}$, these equations compute the difference between the best estimate and the chosen linearizion point. There are different strategies as to how to accommodate this. If the operating point, \mathbf{x}^* , is always zero than these terms simply drop

out; $\Delta \mathbf{x}$ becomes \mathbf{x} and $\mathbf{h}(\mathbf{x}^*)$ goes to zero. This lets Equation 2.23 replace the linear process and measurement equations. When the linearizion point is not zero, these equations can still be updated as they would be normally. Here, however, the states no longer represent the value of the best estimate but rather the deviation from the current linearizion point. If the system can be linearized about a single operating point $\Delta \mathbf{x}$ can be computed and added to the operating point each time step. The last strategy presented, and the one used for this thesis, dictates that \mathbf{x}^* should not be left constant. By setting the operating point equal to the current best estimate of the state vector a more accurate estimation can be obtained. The following equations overview this process by returning to the state update estimate, Equation 2.3, and substituting the linearized system:

$$\Delta \hat{\mathbf{x}}_k = \Delta \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k^*) - \mathbf{H}_k \Delta \hat{\mathbf{x}}_k^-]$$
(2.24)

Adding \mathbf{x}_k^* to each side yields:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k^*) - \mathbf{H}_k \Delta \hat{\mathbf{x}}_k^-]$$
(2.25)

Finally making the substitution

$$\hat{\mathbf{z}}_{k}^{-} = \mathbf{h}(\mathbf{x}_{k}^{*}) + \mathbf{H}_{k} \Delta \hat{\mathbf{x}}_{k}^{-}$$
(2.26)

yields

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \hat{\mathbf{z}}_k^-]$$
(2.27)

This equation can now be substituted into the linear Kalman filter equations. One important difference here is that $\mathbf{H}_k \mathbf{x}_k$ is no longer used to predict the measurements, but rather, Equation 2.26 is used. At each time step the estimated operating point is set to the current estimation so $\Delta \hat{\mathbf{x}}_k^-$ is by definition zero and its associated term drops out.

While the extended Kalman filter provides a means to estimate non linear equations of motion, it does not always provide an optimal estimate as the linear filter does. The Q and R matrices are still assumed to have a linear relationship to the state and measurement vectors, which is not always true. Care should be taken to assure that this mis-modeling does not have adverse effects on the system performance.

CHAPTER 3. One Degree of Freedom Attitude Estimation Kalman Filter

As stated in the introduction the goal of this thesis is to present and analyze a Kalman filter which is capable of accurately estimating sensor orientation when rotated and simultaneously being moved in an arc. Before this filter is discussed a simpler model is presented to establish a performance reference point. The filter discussed in Chapter 3 assumes the sensor is rotated about its own axis and undergoes no linear accelerations. These assumptions minimize sensed accelerations which are not caused by gravity.

The angular position can be determined by using two orthogonal accelerometers to measure the local gravity vector. The two sensors are placed in the plane normal to the axis of rotation and as the sensors are rotated the component of gravity measured changes, as shown in Figure 3.1.



Figure 3.1 System definition diagram.

The relationship between the measurements of the accelerometers and the local gravity vector provides a means to compute absolute orientation relative to gravity. The accuracy of this type of system is limited by sensor noise. A low pass filter provides a simple way to reduce noise, but introduces a phase delay. For low speed applications this approach may be adequate, but high speed systems are more sensitive to phase delays. Additionally, any linear acceleration of the sensors (movement that is not rotational) introduces error into the system. This is a problem in aircraft when performing a coordinated turn. Due to the g loading generated by the turn, the acceleration seen by the accelerometers no longer points straight down. For this reason it is common that aircraft measure their attitude using rate gyros which are not susceptible to linear accelerations.

Rate gyros present their own challenges. Integration is required to compute the displaced angle, introducing two problems. First, the integration must be provided with an initial attitude, and any error here is carried through the integration process. The second problem is generated by the rate gyro measurement bias which is typically non-zero and not constant. Integrating with any bias will create accumulating error. When tracking attitude over a short amount of time a small bias may be manageable, but over long periods the bias will produce an unacceptable amount of drift.

An extended Kalman filter is used to combine data from the two sensor types. In the following sections the filter is derived and then used to make estimates of angle and rate for simulated and experimental data sets.

3.1 **Process Equations**

This section introduces the system model and develops the process equation which will be used in the Kalman filter formulation.

To begin, θ is defined as the absolute angle between the z axis and the measurement direction of the z axis accelerometer, as shown in Figure 3.1. The z axis is defined to point straight up, opposite of gravity. The Y axis points in the forward direction and positive rotation is defined by the right hand rule. $\dot{\theta}$ and $\ddot{\theta}$ are defined as the first and second time derivative of θ which is angular rate and angular acceleration, respectively.

The filter was designed to track arbitrary motion, modeled by a random walk process. $\hat{\theta}$

randomly walks according to w_{θ} , a zero mean white noise random process. The variance of w_{θ} is a function of the system motion.

The process equation formulation begins with Equation 3.1.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \vdots \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_{\theta} \end{bmatrix}$$
(3.1)

Here w_{θ} is white noise which drives the random walk process. Modeling the system in this way assumes that $\ddot{\theta}$ of the motion profile moves randomly. However, the filter was tested with sinusoidal motion profiles, which are clearly not white noise. Modeling the acceleration of the system as a random walk is an ignorance based model. There is no prior knowledge of system motion so this model is used as a guess. Given knowledge of how the system moves the model can be improved over the one given here.

Modeling the rate gyro bias error presents an interesting challenge as it is not white noise. Fast sampling of the bias shows that each sample is highly correlated to the samples before and after it. To model the bias error more accurately a new state ϵ has been appended to the state vector to estimate the rate gyro bias. This state is driven by another random process, w_{ϵ} , which is the rate at which the bias changes. The relationship is shown in Equation 3.2.

$$\dot{\epsilon} = w_{\dot{\epsilon}} \tag{3.2}$$

The addition of the ϵ to the previous state definition yields Equation 3.3.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \vdots \\ \dot{\theta} \\ \dot{\epsilon} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \\ \epsilon \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ w_{\theta} \\ w_{\dot{\epsilon}} \end{bmatrix}}_{\mathbf{w}}$$
(3.3)

Where \mathbf{F} , \mathbf{x} , and \mathbf{w} are redefinitions representing the continuous system matrix, the state vector, and the white noise driving functions respectively.

To develop the process equations \mathbf{F} is discretized via Equation 3.4.

$$\Phi = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \frac{(\mathbf{F}\Delta t)^2}{2!} + \cdots$$
(3.4)

Only the first three terms of Equation 3.4 result in non zero matrices, when summed together they produce the system update matrix, Φ :

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & \Delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(3.5)

The process covariance matrix, Q is constructed to produce the system equations necessary to run the Kalman filter. The process equations were derived with two sources of noise, $w_{\tilde{\theta}}$ and $w_{\hat{\epsilon}}$. Which are the variances of two independent white noise processes with zero mean. Noise is added to the acceleration term to create a random walk model. The subscript θ was chosen to indicate that this term is associated with the angular motion of the system. The bias, ϵ was also modeled as a random walk process in order to model the slowly time-varying rate gyro drift. These two processes are completely independent and there is no cross correlation, so the Q matrix is simply the diagonal matrix with the variances of the white noise processes along the diagonal. **Q** is then defined as:

The process equation for this filter is presented in it's final form in Equation 3.7.

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \mathbf{w} \tag{3.7}$$

Where the k index represents the current discrete timestep.

3.2 Measurement Equation

Unlike the process equation, the measurement equation for this system is nonlinear. The system was linearized about the current best estimate as described in Section 2.2. This allows the system to track the full range of motion for a single axis, between $(-180^{\circ} \text{ to } 180^{\circ})$.

Recalling Equation 2.20, the function, $h(\mathbf{x})$, was constructed by using the kinematic relationship between \mathbf{x} and the three measurements. These are the accelerations in the z and y directions along with the rate in the x direction.

The acceleration sensed by the two accelerometers are functions of the local gravity magnitude and θ , the angle between the z axis and the normal direction of the sensor, as defined in Figure 3.1. Gravity was considered constant for this application and acceleration was measured in G units, this allows the accelerations to be described as functions of θ only, as described in Equation 3.8:

$$A_z = -\cos(\theta) \tag{3.8}$$
$$A_u = -\sin(\theta)$$

Where A_z and A_y are the accelerations measured by the inertial measurement unit (IMU) in the normal and tangential directions respectively. The rate relation is then simply defined as:

$$\Omega_x = \dot{\theta} + \epsilon \tag{3.9}$$

Where Ω_x is the rate gyro reading in the x direction. $h(\mathbf{x})$ can now be defined:

$$\mathbf{z} = \begin{bmatrix} \Omega_x \\ A_y \\ A_z \end{bmatrix} = h(\mathbf{x}) = \begin{pmatrix} \dot{\theta} + \epsilon + v_\epsilon \\ -sin(\theta) + v_A \\ -cos(\theta) + v_A \end{cases}$$
(3.10)

Where v_A and v_{ϵ} are the accelerometer and rate gyro noise variances. Equation 2.23 is then used to define \mathbf{H}_k , the linearization of $h(\hat{\mathbf{x}}_k^-)$.

$$\mathbf{H}_{k} = \frac{\delta \mathbf{h}}{\delta \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ -\cos(\hat{\theta}_{k}^{-}) & 0 & 0 & 0 \\ \sin(\hat{\theta}_{k}^{-}) & 0 & 0 & 0 \end{bmatrix}$$
(3.11)

Where the - superscript indicates that the prior estimate should be used. For the extended Kalman filter \mathbf{H}_k must be recomputed at each time step, using the prior estimate of θ , which provides the closest approximation for use as a linearization point.

The measurement covariance matrix, \mathbf{R} is computed to complete the measurement equation. The variance terms, v, were introduced in Equation 3.10 and are used to formulate the the matrix, just as \mathbf{Q} was created. There is no cross correlation between measurements from different sensors so their variances are placed along the diagonal, resulting in:

$$\mathbf{R} = \begin{bmatrix} v_A & 0 & 0 \\ 0 & v_A & 0 \\ 0 & 0 & v_\epsilon \end{bmatrix}$$
(3.12)

3.3 Initial Conditions

Starting the filter requires the use of some initial conditions. The two parameters, $\hat{\mathbf{x}}_1^$ and \mathbf{P}_1^- , must be given initial values because there is not a k = -1 time step from which to derive these prior estimates. A linear Kalman filter will converge from any initial condition, but extended filters can exhibit divergence if the initial conditions are not chosen properly. Setting the initial prior state estimates close to the true values helps to ensure convergence, but does not always guarantee it. The filter presented requires initial angular position, rate, acceleration, and rate gyro bias. With no prior knowledge these initial values are all set to zero.

The initial error covariance matrix is the diagonal matrix of the initial state estimate variances. If one of the state values is known absolutely at filter startup the corresponding entry is set to zero. As confidence of this initial estimate decreases the entry increases in size.

To facilitate convergence over a wide range of starting conditions, including the one examined here, the initial conditions were defined as:

$$\hat{\mathbf{x}}_{k}^{-} = \begin{bmatrix} 0\\0\\0\\0 \end{bmatrix} \quad \mathbf{P}_{1}^{-} = \begin{bmatrix} 1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 0.1 \end{bmatrix}$$
(3.13)

3.4 Simulation

All of the equations which are required for the attitude estimation problem have now been determined. These values were inserted into the predict and update phases previously discussed and executed sequentially for four different sets of simulated motion. Ideal data sets were generated based on a given angle and rate trajectory, these signals were then corrupted with a white noise with a given variance. The noise data was then fed through the filter algorithm and the estimated signals were then compared to the original trajectory information.

In order to evaluate the performance of each filter the root mean squared error (RMSE) was computed for each simulation run. This metric was also used to determine the performance of datasets derived from running experiments with an IMU sensor. In all cases the RMSE was only computed after the filter had converged.

3.4.1 Simulation data sets

Trajectory data is shown in Table 3.1. All simulations are based on these four sinusoidal motion profiles.

	Simulation 1	Simulation 2	Simulation 3	Simulation 4
Amplitude	10°	20°	40°	20°
Frequency $(\frac{radians}{s}, Hz)$	1, 0.159	1, 0.159	1, 0.159	2, 0.318
Accelerometer Noise Variance $(\frac{\mu g^2}{s^2})$	25	25	25	25
Rate Gyro Noise Variance $(\frac{radians^2}{s^4})$	0.0019	0.0019	0.0019	0.0019

 Table 3.1
 Simulation motion parameters for basic Kalman filter

All simulations were sampled at 819Hz, the same sampling frequency of the IMU which was used in experimental tests. A bias error was also added to the rate gyro. The bias has a random initial value, zero mean with a standard deviation of $3\frac{\circ}{s}$. Once initiated the bias error is modeled with a random walk process, driven by the white noise process $w_{\acute{e}}$ which has a standard deviation of $0.007\frac{\circ}{s}$.

The measurement variances used in the filter were chosen based on the noise variances specified in Table 3.1. The rate gyro bias variance was also selected according to noise process definition. The remaining variance parameter, w_{θ} , was selected to be $0.25 \frac{\delta^2}{s^9}$ to allow for a wide range of motion.

To demonstrate the need to filter noisy measurements for Simulation 1 IMU sensor measurements were directly converted to position data. Figure 3.2 shows the position derived from rate gyro integration and Figure 3.3 shows the position derived from the accelerometer measurement.



Figure 3.2 Position computed from noisy rate gyro measurment.



Figure 3.3 Position computed from noisy accelerometer measurement.

It is clear from the figures that both methods have their own shortcomings. The integrated rate signal drifts due to the bias error accumulating and the accelerometer derived position signal did not drift but was more noisy. Adhering to the metric, the RMSE for the position from the accelerometer data was computed to be 0.290°^2} . The error from the integrated rate data clearly grows very large in a matter of seconds.

For some applications this noise level may be adequate in which case there is no need for advanced filtering. In other applications some amount of phase delay may be tolerated in which case the noisy signal could be low pass filtered to obtain sufficient results. However, most applications desire better performance than this.

3.4.2 Simulation results

After the Kalman filter was applied to Simulation 1 data, the estimated angle and rate were plotted against the reference trajectories. These comparisons are shown in Figure 3.4 and Figure 3.5. The figures only plot the first two seconds of simulation to present the subtleties of the data.



True Angle vs Estimated Angle

Figure 3.4 Kalman filter angle result for simulation 1.



Figure 3.5 Kalman filter rate result for simulation 1.

Both signals converged in under a half second, though the rate estimate took longer to do

so. This was a result of the poor initial estimate. All initial states were guessed to be zero, for the angle state this was accurate, but not so for rate. The initial velocity was not zero and the estimate takes some time to converge to the true signal. To obtain an average RMSE value for each set of data the simulation was run fifty times, and the results were averaged together. For the first simulation case the RMSE for the angle and rate estimate are 0.022° and $0.055\frac{\circ}{s}$ respectively. Recalling the RMSE error for the direct conversion the reader can see that the Kalman filter provides nearly an order of magnitude improvement over the directly converted data.

Error signals were also computed, comparing the difference between the estimated values computed by the filter and uncorrupted motion data. These plots are presented in Figure 3.6 3.7.



Figure 3.6 Kalman filter angle error for simulation 1.

These plots show a more detailed view of the initial convergence and the error seen afterwards. The angle estimate converges all at once, in under a half second. It is also verified that the rate takes longer to reach steady state, but not all at once. A large burst for the first half second and then eliminating the remaining error over the next half second.



Figure 3.7 Kalman filter rate error for simulation 1.

The same simulation was then run for the remaining three simulation sets. The RMSE values were once again used as a means of comparison. The results are tabulated below in Table 3.2. The results indicated consistent performance over all simulation sets, consistent with expectations. All simulations used identical noise variances and fall within the bandwidth of the system, so it follows that the output noise was consistent.

	Simulation 1	Simulation 2	Simulation 3	Simulation 4
Amplitude RMS error (°)	0.0220	0.0221	0.0221	0.0223
Rate RMS error $(\frac{\circ}{s})$	0.0550	0.0551	0.0571	0.0645

Table 3.2 Simulation RMS error results for basic Kalman filter.

3.5 Experiment

To explore the accuracy of the model and the caveats that exist in real dynamic systems an experiment using real sensors was performed. Although much can be learned through modeling, there is no substitute for analyzing real data.

3.5.1 Test Setup

To collect the experimental data sets a test fixture was constructed using an Analog Devices ADIS16362 triaxial rate gyro and accelerometer IMU. The sensor was affixed to rotate about the axis of a large stepper motor. The stepper motor was then commanded to move in the profiles described in Table 3.1. The stepper motor allowed for a very accurate reference signal to be recorded alongside the measurement data. Both signals were sampled at $f_{IMU} = 819.2$ Hz, the maximum sampling rate of the IMU.

3.5.2 Sensor noise analysis

The noise used to corrupt the signal for the simulated data sets had equal power at all frequencies. While convenient for simulation, the sensors used did not exhibit this behavior. White noise was assumed for the derivation of the Kalman filter and proper implementation relies upon each time step having noise uncorrelated to any other time step.

The IMU used in this experiment digitally processes measurement data internally before passing it onto the end user. After being digitized, samples are passed through two averaging filters before being passed to the Kalman filter. The signal chain is shown for reference in Figure 3.8.



Figure 3.8 IMU signal processing chain.

The two averaging filters act to correlate data between time steps. Each averaging filter has four samples of memory which act to correlate data between time steps. This correlated sensor data, or colored sensor noise, violates the white noise assumption required by the Kalman filter. Figure 3.9 shows the power spectrum density (PSD) of the rate gyro noise as an example. The two accelerometer measurements used have similar PSD responses, but were more susceptible



to external vibrations so were not used for analysis.

Figure 3.9 Power spectrum density of the rate gyro error.

One method to account for colored noise is to add the filter response to the system model. The Kalman filter would require eight additional states to accurately model the digital filter response. If the exact filter method were not known an autoregresive (AR) model could also be used to model the digital filter. Both scenarios require the addition of system states, a process described by Gibson [3].

An alternative approach, and the one used in this experiment, limits the bandwidth of the Kalman filter so that the colored noise appears white. Power between zero and one hundred Hertz varies only by a few dB. Data can be re-sampled at twice this rate to limit the bandwidth and give it the appearance of being white. The spectral content of the simulation and experiments performed occurred at less than one Hertz, so limiting the system bandwidth did not interfere with tracking. The bandwidth of the system was limited by taking every fourth sample that came from the digital signal. This re-sampling process defined a new filter sampling rate, $f_{filter} = 204.8$ Hz. Because re-sampling was done at even intervals no new aliasing occurred.

By only using every fourth sample the sensor data being fed into the Kalman filter is now sufficiently independent from time step to time step. As verification, Figure 3.10 presents the new spectral density plot at the lower sampling rate. The figure verifies the new noise is sufficiently white and that spectral content was not affected during the re-sampling process.



Figure 3.10 Power spectrum density of the rate gyro error at f_{filter} .

Although a single noise variance is defined by the manufacturer of the IMU, experimental data showed that noise varied with speed, test stand smoothness, and environmental vibrations for the different experiments. To provide the Kalman filter with more accurate noise variance information error signals were computed for each set of experimental data. The computed noise variances, along with nominal trajectory profiles, for each experiment is given in Table 3.3

	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Amplitude (°)	10	20	40	20
Frequency $(1rad/s)$	1	1	1	2
Y Accelerometer Noise Variance $\left(\frac{\mu g^2}{s^2}\right)$	108	164	139	143
Z Accelerometer Noise Variance $(\frac{\mu g^2}{s^2})$	55	86	78	62
Rate Gyro Noise Variance $\left(\frac{\mu radians^2}{s^4}\right)$	47	60	95	144

Table 3.3Experimental motion parameters and computed noise variances
for the basick Kalman filter.

3.5.3 Filter results

As described in the previous section, the filter was modified to utilize every fourth sample and run with variance parameters as defined in Table 3.3. The RMS errors for the four

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Angle RMS error (°)	0.08	0.13	0.23	0.27
Rate RMS error $\left(\frac{\circ}{s}\right)$	0.31	0.37	0.50	0.65

experiments are presented in Table 3.4.

Table 3.4 Experiment RMS error results for expanded state Kalman filter.

When compared to simulation counterparts, all experimental error values were found to be larger. This result was expected as the noise variance values were also higher. The error signals for rate and position are shown below in Figure 3.11. A large contributor of the total



Figure 3.11 Angle and rate error for Experiment 1.

RMS error is due to the slow sinusoid present in the error signal. This error is primarily due to a phase shift between the true position and the output of the filter, caused by the digital filter implemented in the IMU which imparts an eight sample delay. To verify the source of the error the reference angle and rate were phase shifted forward eight time steps to synchronize the data. Although this can not be done in real time it is useful for validation purposes.



The phase corrected data is shown in Figure 3.12.

Figure 3.12 Angle and Rate error for Experiment 1 with phase shifted sensor data.

The data shift clearly reduced angle and rate RMS error. For Experiment 1 the angle RMS error was reduced to 0.059° from 0.080° , while the rate RMS error was marginally improved to $0.30\frac{\circ}{s}$ from $0.31\frac{\circ}{s}$. Performing the phase shift reduced the sinusoidal component, an indication that the phase shift was eliminated. Shifting by more or less than eight time steps resulted in higher RMSE values, confirming the phase shift reduction. The remaining errors were likely due to minor imperfections in the test rig such as reference signal misalignment, IMU sensor orthogonality errors, and scale nonlinearities.

CHAPTER 4. Attitude Estimation with Radius Estimation

The system model presented in Section 3.1 assumed that all accelerations measured were due to gravity. This model is accurate when the IMU is placed along the axis of rotation. When this axis of rotation is offset from the sensor axis by some radius "R", tangential and normal acceleration components are also sensed by the IMU along the Z and Y axises. A diagram is presented in Figure 4.1 where the new accelerations and radius are denoted in red.



Figure 4.1 System definition with normal and tangential acceleration components.

The diagram shows that the two accelerometers now measure gravity along with normal and tangential accelerations. The model was expanded to account for the new acceleration components in an effort to reduce the error that would otherwise be seen due to the unmodeled acceleration. This expanded state filter is the focus of discussion for the remainder of this chapter.

4.1 Measurement and Process Equations

The normal and tangential accelerations presented are related to the radius of rotation, R, and the angular motion θ , as shown in Equation 4.1.

$$a_{normal} = \dot{\theta} \times (\dot{\theta} \times R)$$

$$a_{tangential} = \ddot{\theta} \times R$$
(4.1)

The IMU was positioned so that all normal acceleration was sensed by the Z axis and all tangential acceleration was sensed by the Y axis. Combining this relation with Equation 3.8 produces the final relation between acceleration and θ , presented below:

$$A_y = -\sin(\theta) + R\ddot{\theta}$$

$$A_z = -\cos(\theta) + R\dot{\theta}^2$$
(4.2)

These types of rate gyros are not significantly affected by linear accelerations so no new compensation modeling was required for rate estimation. This led to the redefinition of h(x):

$$\dot{\theta} + \epsilon + v_{\epsilon}$$

$$h(\mathbf{x}) = \begin{pmatrix} \dot{\theta} + \epsilon + v_{\epsilon} \\ -\sin(\theta) + R\ddot{\theta} + v_{A} \\ -\cos(\theta) + R\dot{\theta}^{2} + v_{A} \end{pmatrix}$$

$$(4.3)$$

Where the variable definitions remain consistent with the original filter definitions.

A new state was introduced in the new state vector, R the radius of rotation. The measurement equation was still nonlinear so it was again linearized to provide \mathbf{H}_k .

$$\mathbf{H}_{k} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ -\cos(\theta) & 0 & R & 0 & \ddot{\theta} \\ \sin(\theta) & 2\dot{\theta}R & 0 & 0 & \dot{\theta}^{2} \end{bmatrix}$$
(4.4)

The measurement covariance matrix used in the original Kalman filter can be reused exactly for the filter with radius estimation. The following were redefined and are shown below, \mathbf{F} , \mathbf{x} , and \mathbf{w} .

Using Equation 3.4, Φ is constructed from F:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
(4.6)

The formulation of the process error covariance matrix was similar to the original filter, with one main difference. The addition of the radius random walk process variance, ω_R . This parameter dictates how much the radius is expected to change between time steps. For a physical layout where the radius is known to be constant this value should be very small so that the estimate of the state will converge to the true value. For systems where it is known that the radius will change a larger ω_R should be chosen. Q was defined as:

The completed radius estimating process and measurement equations were, once again, used to filter simulated and experimental data.

4.2 Simulation

The simulation procedure for the filter with radius estimation was similar to the original filter. The new simulation sets which were used are described in Table 4.1. The primary change was the incorporation of the radius parameter for each data set.

	Simulation 5	Simulation 6	Simulation 7	Simulation 8
Amplitude (^{c}irc)	10°	20	40	20
Frequency $(1rad/s)$	1	1	1	2
Radius (m)	1.5	1.5	1.5	1.5
Accelerometer Noise Variance $(\frac{\mu g^2}{s^2})$	25	25	25	25
Rate Gyro Noise Variance $(\frac{radians^2}{s^4})$	0.0019	0.0019	0.0019	0.0019

Table 4.1Simulation motion parameters for the expanded state Kalmanfilter.

The noise parameters were defined based on the noise characteristics of the IMU sensor, as specified by the manufacturer. Unfiltered data was again used as a starting point of the simulation analysis. Figure 4.2 shows accelerometer data which was directly converted to position data in Figure 4.2.



Figure 4.2 Reference angle vs angle computed directly from accelerometer measurement.

Figure 4.2 shows that sensor noise is no longer the primary contributor to the error. Instead the additional acceleration components added the most error. The error when directly converted was in terms of degrees rather than tenths of a degree, as was seen in the filter with no radius. Although the rate gyros were unaffected by the additional acceleration they still suffer bias drift. Moving the sensor off of the axis of rotation caused large amounts of error when directly converted.

When Simulations five through eight were run through the Kalman filter discussed in Chapter 3 similar poor results were seen, with RMS errors greater than one degree seen. Increasing the accelerometer noise variance decreased the error some, but at the cost of less accuracy.

However, through the use of the improved model a much more accurate angle and rate estimate can be obtained. This was one of the primary motivations for using the radius estimating Kalman filter.

4.2.1 Simulation results

The simulations for the radius estimating filter were run in the same way as the first filter presented. The first few seconds of Simulation 5 are shown in Figure 4.3 and Figure 4.4. These demonstrate the convergence and tracking of the filter.



True Angle vs Estimated Angle

Figure 4.3 Simulation 5 reference and estimated angle comparison.



Figure 4.4 Simulation 5 reference and estimated rate comparison.

These results are similar in nature to Simulation 1. The angle estimate once again received a correct initial estimate and converged faster than the rate estimate which received an incorrect initial estimate. In absolute terms the new filter representation took about the same time to converge as the filter discussed in Chapter 3. By varying initial error covariance values, or **P** terms, it was observed that angle and rate convergence time was very dependent on radius convergence time. Given the coupled nature of these estimates this was not a surprise.

It is also useful to look at error plots for angle and rate, shown in Figures 4.5 and 4.6.



Figure 4.5 Simulation 5 angle error.



Figure 4.6 Simulation 5 rate error.

These plots again show the convergence time was less than one second. They clearly show the nature of the error and the style of convergence. The angle quickly approached the correct value but the remaining error decayed slowly. Meanwhile the error seen in the rate converged at a more steady pace.

Figure 4.7 plots the rate error covariance for the two filters, with and without radius estimation. These signals came from the diagonal parameters of P_k , which provides the covariance of the $\hat{\mathbf{X}}_k$ estimates. These variances were plotted over time to give a history of the rate confidence.



Figure 4.7 Rate covariance comaprison for radius estimation filter and inline filter.

It is clear that each filter ultimately converged to some final value, which was a function of the R and Q matrices along with sensor noise and system dynamics. Looking at the error covariance is typically a good indication of system convergence. This rule was used for Figure 4.7 to show that the radius estimating filter converged to an intermediate level in about 200ms before finally converging after about one second. This 200ms intermediate period was also observed in the rate estimate in Figure 4.6. The radius estimate is shown in Figure 4.8.



Figure 4.8 Radius estimate for all simulation.

It is clear that Simulation 5 took much longer to converge than the other test cases. This is due to the lower angular rate and acceleration components of the motion. Despite the half meter error at one second, Simulation 5 still produced respectable RMS error results. However the tabulated result section shows that RMS errors are lower for the last three simulations.

Fortunately it is easy to observe any of the state covariance signals to determine when the system has converged. The covariance history of the radius estimate for each simulation is shown in Figure 4.9.



Figure 4.9 Radius estimate covariance.

This confirms that the radius estimate was mostly converged within one second for all simulations. It also demonstrates the stair-step convergence demonstrated in Figure 4.7. Perhaps the most valuable comparison from Figure 4.9 was that as the speeds in the simulation increase the error covariance decreased faster and more steadily. Equation 3.8 presented a clear relation between the radius estimate and time derivatives of θ . In order for the filter to estimate radius the sensor must either be rotating or accelerating.

4.2.1.1 Tabulated simulation results

As was done in the first four simulations, simulations 5 through 8 were each run fifty times and the average RMSE for angle and rate were computed. The results are presented below.

Table 4.4 shows similar RMS errors for angle and rate across all simulations with Simulation 5 tending to have the most error. As discussed in Chapter 3, the sensor noise parameters for each simulation were the same and the different rate and acceleration profiles used were all well

	Simulation 5	Simulation 6	Simulation 7	Simulation 8
Angle RMS error (°)	0.029	0.027	0.024	0.028
Rate RMS error $\left(\frac{\circ}{s}\right)$	0.070	0.067	0.059	0.061

Table 4.2 Simulation RMS error results for expanded state Kalman filter.

within the bandwidth of the filter. It follows then that the output noise of the filter should be similar for all simulations.

Comparing simulations one through four with simulations five through eight show that the second set had slightly higher RMS error values overall. This was due to the error in radius estimation. As this error approaches zero, as in Simulation 8, the rate and angle errors approach Simulation 4. The greatest deviation is seen between Simulation 1 and Simulation 5, where the radius estimation error is the greatest.

4.3 Experiment

Once again an experiment was performed to corroborate the results found in the simulation section. Test data was taken in the same manner as it was for experiments one through four which have already been analyzed. The test stand was modified by moving the sensor onto the end of a 1.5 meter arm. This reflects the model presented in Figure 4.1.

4.3.1 Noise analysis

The same digital filter settings were used on the IMU, so the noise analysis performed earlier remains valid for this set of experiments. The filter was run at $f_{filter} = 204.8Hz$ in order to limit the bandwidth of the system and to preserve the white noise assumption. New noise variances for each experiment case were determined and are listed in Table 4.3.

The measurement variances were all larger than those for the first four experiments, due to the addition of vibrations on the test stand. Plotting the power spectrum in Figure 4.10 provides some insight into the sources of noise.

Experiments five through eight saw elevated power levels at 0.15 and 6 Hz. The low frequency power was at the same frequency as the system motion, one radian per second.

	Exp. 5	Exp. 6	Exp. 7	Exp. 8
Amplitude (°)	10	20	40	20
Frequency $(1rad/s)$	1	1	1	2
Radius (m)	1.5	1.5	1.5	1.5
Y Accelerometer Noise Variance $\left(\frac{\mu g}{\right)$	846	5212	6937	6431
Z Accelerometer Noise Variance $\left(\frac{\mu g}{\right)$	91	282	391	453
Rate Gyro Noise Variance $\left(\frac{\mu radians}{s}\right)$	88	326	367	442

Table 4.3Experimental motion parameters and computed noise variances
for the expanded state Kalman filter.



Figure 4.10 Spectral power response for rate gyro sampled at 204.8Hz.

The power at 6 Hz corresponds to the natural frequency of the test stand. This additional motion was not measured and not incorporated into the truth signal and therefore appears as error. In fact this motion actually occurred but cannot be accurately measured so was treated as error. This additional motion can be reduced through the use of a stiffer beam, but never completely eliminated. For more robust analysis a test stand should be designed with a natural frequency higher than the sampling rate of the system.

4.3.2 Filter results

Data from Experiments five through eight was processed by the Kalman filter with radius estimation. The results are listed in Table 4.4 and error signals from Experiment 5 are plotted in Figures 4.11 and 4.12.

	Experiment 5	Experiment 6	Experiment 7	Experiment 8
Angle RMS error (°)	0.11	0.24	0.31	0.26
Rate RMS error $\left(\frac{\circ}{s}\right)$	0.37	0.84	0.82	0.93

Table 4.4 Experiment RMS error results for expanded state Kalman filter.



Figure 4.11 Experiment 5 angle error.

Real sensors and mechanical vibration resulted in larger noise variances for all four experiments. This was the primary source of higher RMS errors. A portion of the error was due to uncertainties caused by flexure in the beam used to offset the sensor for the axis of rotation. The two sources of error occur due to the unavoidable resonance in the beam and flexing at the endpoints of the sensor movement. This is confirmed by the error signal plots, most notably Figure 4.12. The resonance manifests itself as the high frequency component of the error. Because these movements are within the bandwidth of the system the filter should track them reasonably well. However, with this test setup it was impossible to verify that this



Figure 4.12 Experiment 5 rate error.

was the case; an alternate method of angle measurement would be required to confirm this.

Another source of error comes from how accurately the radius of rotation is estimated. Section 4.2.1 discussed that larger $\dot{\theta}$ and $\ddot{\theta}$ values produced a better radius estimation and in turn a better angle and rate estimation. The experimental data collected had more noise for faster motion profiles. The filter then used larger variances for the measurements, which resulted in larger angle and rate estimation errors.





Figure 4.13 Radius estimate for experiments with radius.

This figure makes it clear that Experiment 8 converged quickly and accurately. With the other cases slowly approaching the true radius value of 1.5 meters. Despite the sensor noise variances being similar or higher, Experiments 8 maintained lower RMS error for the angle estimate. Experiment 8 converged faster because the motion profile had twice the frequency. The higher average angular rate and accelerations created more certainty in the radius estimate which in turn increased model accuracy and angle estimate performance.

4.3.3 Convergence

The extended Kalman filter typically requires some consideration regarding convergence, and the radius estimating filter was no exception. Typically a system diverges at filter start up when state estimate covariances are low, and initial state estimates are poor. Placing the initial estimates sufficiently close the the true starting locations allowed for proper filter convergence. Most important was selection of the initial radius estimate. For the experiments presented here this value was set to one meter. When set to negative or too near to zero the radius estimate would lock on to a negative estimate, obviously incorrect. The estimate could potentially jump a large amount in the first second of the filter and placing the initial radius estimate at some value larger than zero prevents it from reaching an equilibrium with a negative radius.

Divergence can also be caused by large sudden changes in the states of the system. These types of dramatic changes typically violate some part of the process model. So, for all the experiments smooth motion was generated in order to avoid this type of divergence.

4.3.4 Zero radius experiments with radius estimation filter

Given adequate motion, the radius estimation filter did a remarkable job of estimating the radius. Only non-zero radius experiments have been examined up to this point. It is valuable to ensure that the filter is still operates reasonably well for data sets where there is no radius of rotation.

To test the filter, data from experiments one through four were used with the radius estimating filter. The results are listed in Table 4.5.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Angle RMS error (°)	0.08	0.14	0.22	0.23
Rate RMS error $\left(\frac{\circ}{s}\right)$	0.30	0.43	0.49	0.63

Table 4.5Experiment RMS error results for expanded state Kalman filterused with inline motion profiles.

These RMS error values are very close to those seen in the Kalman filter without radius estimation in Table 3.4. This indicates that once properly converged the radius estimation filter performed as well as the filter without radius estimation. However, greater care must be taken to ensure that all estimates properly converge. If the radius converges to an incorrect value, a problem which can be exasperated by a slow moving system or by poorly selected variance terms, and the system performance can be greatly degraded.

CHAPTER 5. Conclusion

The formulation of the linear and extended Kalman filter equations have been presented, along with two methods for determining attitude with inertial sensors. It has been shown that for systems where the sensor can be placed on or near the axis of rotation the inline filter discussed in Chapter 3 provided more accurate results without the increased risk of divergence. Simulated filter runs produced RMS error values near 0.02 degrees.

For systems where the sensor could not be placed along the axis rotation, or the axis of rotation moved unpredictably the filter with radius estimation, presented in Chapter 4, was more appropriate. This filter provided similar RMS error values to the filter with no radius estimation while decreasing the assumptions made about sensor placement. The inaccuracies primary came from errors in the radius estimation. For systems moving faster than ten degrees per second and a frequency of one radian per second the radius converged adequately and little or now difference between the two filters was observed.

In contrast, for systems where the motion profile was slow, tangential and normal accelerations were also slow. For these systems the radius estimation filter was not appropriate. The slower motion produced inaccurate radius estimation which in turn created errors in the angle and rate estimations. Additionally low speed systems produced only small normal and tangential errors, providing little benefit for their reduction.

In some cases, the radius of rotation may be a known constant. If so, a modified version of the radius estimating filter could be used, where the radius state is eliminated and all instances of the "R" state are replaced with the known value.

Future work includes the incorporation of Coriolis terms for a changing radius and the extension to non-planar motion. Coriolis incorporation was not included here because all radii

42

were assumed to be constant. For a system where these radii are expected to move at a significant speed, there could be significant Coriolis terms that need to be accounted for. This would involve adding an additional state \dot{R} , the time derivative of R, which would be coupled to the rest of the system via the Coriolis term and normal acceleration.

To extend the system to two degrees of freedom requires more significant work. For simple motion where the system is not expected to roll or pitch more than ten or twenty degrees, angle and rate could be approximated by using two uncoupled planar models, this however, introduces another error component.

To properly model the system, roll and pitch models need to be combined to account for a common normal acceleration term. This will create a nine state filter. The process equation includes one state for radius, four states for roll, and four states for pitch, with similar construction to filters already presented. The measurement equation will be more complex, requiring the incorporation of the coupled dynamic model.

In all, care should be taken in choosing the type of filter to use for a given dynamic system. Sensor placement and expected motion profiles are two of the main concerns when investigating which filter type to be used.

Bibliography

- Crassidis, J. L., Markley, F. L., and Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance Control and Dynamics*, 30:12–28.
- [2] Foxlin, E., Harrington, M., and Altshuler, Y. (1998). Miniature 6-dof inertial system for tracking hmds. In In Proc. SPIE Helmet and Head-Mounted Displays III, pages 214–228.
- [3] Gibson, J., Koo, B., and Gray, S. (1991). Filtering of colored noise for speech enhancement and coding. Signal Processing, IEEE Transactions on, 39(8):1732-1742.
- [4] Kim, J.-W., Cristi, R., and Agrawal, B. N. (2004). Attitude determination for nps threeaxis spacecraft simulator.
- [5] Lee, J. K. and Park, E. J. (2009). A minimum-order kalman filter for ambulatory real-time human body orientation tracking. In *ICRA'09: Proceedings of the 2009 IEEE international* conference on Robotics and Automation, pages 2797–2802, Piscataway, NJ, USA. IEEE Press.
- [6] Moreno, J. C., de Lima, E. R., Ruz, A. F., Brunetti, F. J., and Pons, J. L. (2006). Design and implementation of an inertial measurement unit for control of artificial limbs: Application on leg orthoses. *Sensors and Actuators B: Chemical*, 118(1-2):333 – 337. Eurosensors XIX - Eurosensors XIX - The 19th European Conference on Solid-State Transducers.
- [7] Sabatini, A. (2006). Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *Biomedical Engineering, IEEE Transactions on*, 53(7):1346 -1356.
- [8] Savage, P. G. (2007). Introduction to Strapdown Inertial Navigation Systems.

- [9] Sun, F., Sun, W., Gao, W., and Ben, Y. (2009). Research on the technology of rotational motion for fog strapdown inertial navigation system. In *Mechatronics and Automation*, 2009. ICMA 2009. International Conference on, pages 4913–4918.
- [10] Tayebi, A., McGilvray, S., Roberts, A., and Moallem, M. (2007). Attitude estimation and stabilization of a rigid body using low-cost sensors. In *Decision and Control, 2007 46th IEEE Conference on*, pages 6424–6429.
- [11] William Kellar, Peter Roberts, O. Z. (2008). A self calibrating attitude determination system for precision farming using multiple low-cost complementary sensors. In *Proceedings* of the 1st International Conference on Machine Control and Guidance, Zurich, Switzerland.
- [12] Zha, F., ning Xu, J., qing Hu, B., and jun Qin, F. (2010). Error analysis for sins with different imu rotation scheme. In *Informatics in Control, Automation and Robotics (CAR)*, 2010 2nd International Asia Conference on, volume 1, pages 422 –425.