KALMAN FILTER AND ITS APPLICATIONS IN NAVIGATION

A graduate project submitted in partial fulfillment of the requirements
For the degree of Master of Science
In Electrical Engineering

By
Dursun Tepeli

December 2012
The graduate project of Dursun Tepeli is approved by:

__________________________________________  _____________
Dr. Xiyi Hang  

__________________________________________  _____________
Dr. Ali Amini  

__________________________________________  _____________
Dr. Xiaojun (Ashley) Geng (Chair)  

California State University, Northridge
# TABLE OF CONTENTS

SIGNATURE PAGE ii
LIST OF FIGURES iv
LIST OF SYMBOLS vii
ABSTRACT viii

## CHAPTER 1: INTRODUCTION
1.1 Objectives 2
1.2 Motive for Kalman Filter in Navigation 2

## CHAPTER 2: INERTIAL NAVIGATION SYSTEM
2.1 INS History 5
2.2 INS Theory 5
2.3 Practical INS 5
2.4 INS Solution and Aiding Devices 7
   2.4.1 INS Velocity and Position Solution 7
   2.4.2 Doppler Radar 8
   2.4.3 Doppler Velocity Sensor 8

## CHAPTER 3: DISCRETE KALMAN FILTER
3.1 Kalman Filter Algorithm 9

## CHAPTER 4: KALMAN FILTER AND ITS APPLICATIONS IN NAVIGATION
4.1 Kalman filter and its application on altitude Damping 15
4.2 Baro-altitude filter Simulation 18
   4.2.1 Baro-altitude Simulation Results 20
4.3 Kalman filter and its application on external velocity aiding 38
4.4 INS/Doppler Radar Measurement Model Simulation 38
   4.4.1 Simulation Results 48

## CHAPTER 5: CONCLUSION AND FUTURE WORK
5.1 Conclusion Remarks 78
5.2 Future Considerations 79

REFERENCES 82

APPENDIX: SIMULATION MODEL AND CODE 83
LIST OF FIGURES

Figure 3.1.1 Kalman filter recursive process 13
Figure 4.1.1 Error models used for the vertical channel 16
Figure 4.2.1.1 Mean Square Error of Estimated State Altitude 22
Figure 4.2.1.2 True and Estimated Altitude 23
Figure 4.2.1.3 Residual z 24
Figure 4.2.1.4 True State minus Estimated State 25
Figure 4.2.1.5 True and estimated Baro Error RMS 26
Figure 4.2.1.6 True and estimated states, x1= 10 m, P0 = 0 m 27
Figure 4.2.1.7 Residual z, x1= 10 m, P0 = 0 m 28
Figure 4.2.1.8 True minus estimate, x1= 10 m, P0 = 0 m 29
Figure 4.2.1.9 True and estimated states, x1 = 10 m, P0 = 10 m 30
Figure 4.2.1.10 Residual z, x1= 10 m, P0 = 10 m 31
Figure 4.2.1.11 True minus estimate, x1= 10 m, P0 = 10 m 32
Figure 4.2.1.12 True and estimated states, x1 = 0 m, P0 = 0 m 33
Figure 4.2.1.13 Residual z, x1= 0 m, P0 = 0 m 34
Figure 4.2.1.14 True minus estimate, x1 = 0 m, P0 = 0 m 35
Figure 4.2.1.15 True and estimated states, x1 = 0 m, P0 = 10 m 36
Figure 4.2.1.16 Residual z, x1 = 0 m, P0 = 10 m 37
Figure 4.2.1.17 True minus estimate, x1 = 0 m, P0 = 10 m 38
Figure 4.4.1 Aircraft velocity vector convention 39
Figure 4.4.2 North Channel error model 40
Figure 4.4.3 East Channel error model 40
Figure 4.4.4 Altitude channel error model 41
Figure 4.4.1.1 True and estimated x-velocity, x2 = 500 m/s, P0 = 1000 m/s 50
Figure 4.4.1.2 Residual z1, x2 = 500 m/s, P0 = 1000 m/s 51
Figure 4.4.1.3 True minus estimated, x2 = 500 m/s, P0 = 1000 m/s 52
Figure 4.4.1.4 True and estimated y-velocity, x5 = 500 m/s, P0 = 1000 m/s 52
Figure 4.4.1.5 Residual z2, x5 = 500 m/s, P0 = 1000 m/s 53
Figure 4.4.1.6 True minus estimated, x5 = 500 m/s, P0 = 1000 m/s 53
Figure 4.4.1.7 True and estimated x-velocity, x2 = 500 m/s, P0 = 100 m/s 54
Figure 4.4.1.8 Residual z2, x2 = 500 m/s, P0 = 100 m/s 54
Figure 4.4.1.9 True minus estimated, x2 = 500 m/s, P0 = 100 m/s 55
Figure 4.4.1.10 True and estimated y-velocity, x5 = 500 m/s, P0 = 100 m/s 55
Figure 4.4.1.11 Residual z2, x5 = 500 m/s, P0 = 100 m/s 56
Figure 4.4.1.12 True minus estimated, x5 = 500 m/s, P0 = 100 m/s 56
Figure 4.4.1.13 True and estimated x-velocity,
x2′ =500 m/s, P0′ = 0 m/s 57
Figure 4.4.1.14 Residual z1, x2− =500 m/s, P0′ = 0 m/s 57
Figure 4.4.1.15 True minus estimated, x2− =500 m/s, P0′ = 0 m/s 58
Figure 4.4.1.16 True and estimated y-velocity,
x5′ =500 m/s, P0′ = 0 m/s 58
Figure 4.4.1.17 Residual z2, x5′ =500 m/s, P0′ = 0 m/s 59
Figure 4.4.1.18 True minus estimated, x5′ =500 m/s, P0′ = 0 m/s 59
Figure 4.4.1.19 True and estimated x-velocity, x2′ =100 m/s,
P0′ = 1000 m/s 60
Figure 4.4.1.20 Residual z1, x2− =100 m/s, P0′ = 1000 m/s 60
Figure 4.4.1.21 True minus estimated, x2− =100 m/s,
P0′ = 1000 m/s 61
Figure 4.4.1.22 True and estimated y-velocity,
x2′ =100 m/s, P0′ = 1000 m/s 61
Figure 4.4.1.23 Residual z2, x5′ =100 m/s, P0′ = 1000 m/s 62
Figure 4.4.1.24 True minus estimated, x5′ =100 m/s,
P0′ = 1000 m/s 62
Figure 4.4.1.25 True and estimated x-velocity,
x2′ =100 m/s, P0′ = 100 m/s 63
Figure 4.4.1.26 Residual z1, x2− =100 m/s, P0′ = 100 m/s 63
Figure 4.4.1.27 True minus estimated, x2− =100 m/s,
P0′ = 100 m/s 64
Figure 4.4.1.28 True and estimated y-velocity,
x5′ =100 m/s, P0′ = 100 m/s 64
Figure 4.4.1.29 Residual z2, x5′ =100 m/s, P0′ = 100 m/s 65
Figure 4.4.1.30 True minus estimated, x5′ =100 m/s,
P0′ = 100 m/s 65
Figure 4.4.1.31 True and estimated x-velocity,
x2′ =100 m/s, P0′ = 0 m/s 66
Figure 4.4.1.32 Residual z1, x2− =100 m/s, P0′ = 0 m/s 66
Figure 4.4.1.33 True minus estimated, x2− =100 m/s, P0′ = 0 m/s 67
Figure 4.4.1.34 True and estimated y-velocity, x5′ =100 m/s,
P0′ = 0 m/s 67
Figure 4.4.1.35 Residual z2, x5′ =100 m/s, P0′ = 0 m/s 68
Figure 4.4.1.36 True minus estimated, x5′ =100 m/s, P0′ = 0 m/s 68
Figure 4.4.1.37 True and estimated x-velocity, x2′ =0 m/s,
P0′ = 1000 m/s 69
Figure 4.4.1.38 Residual z1, x2− =0 m/s, P0′ = 1000 m/s 69
Figure 4.4.1.39 True minus estimated, x2− =0 m/s,
P0′ = 1000 m/s 70
Figure 4.4.1.40 True and estimated y-velocity, x5′ =0 m/s,
P0′ = 1000 m/s 70
Figure 4.4.1.41 Residual z2, x5′ =0 m/s, P0′ = 1000 m/s 71
Figure 4.4.1.42 True minus estimated, x5′ = 0 m/s, P0′ = 1000 m/s 71
Figure 4.4.1.43 True and estimated x-velocity, $x_2^* = 0$ m/s, $P_0^* = 100$ m/s 72
Figure 4.4.1.44 Residual $z_1$, $x_2^* = 0$ m/s, $P_0^* = 100$ m/s 72
Figure 4.4.1.45 True minus estimated, $x_2^* = 0$ m/s, $P_0^* = 100$ m/s 73
Figure 4.4.1.46 True and estimated y-velocity, $x_5^* = 0$ m/s, $P_0^* = 100$ m/s 73
Figure 4.4.1.47 Residual $z_2$, $x_5^* = 0$ m/s, $P_0^* = 100$ m/s 74
Figure 4.4.1.48 True minus estimated, $x_5^* = 0$ m/s, $P_0^* = 100$ m/s 74
Figure 4.4.1.49 True and estimated x-velocity, $x_2^* = 0$ m/s, $P_0^* = 0$ m/s 75
Figure 4.4.1.50 Residual $z_1$, $x_2^* = 0$ m/s, $P_0^* = 0$ m/s 75
Figure 4.4.1.51 True minus estimated, $x_5^* = 0$ m/s, $P_0^* = 0$ m/s 76
Figure 4.4.1.52 True and estimated y-velocity, $x_5^* = 0$ m/s, $P_0^* = 0$ m/s 76
Figure 4.4.1.53 Residual $z_2$, $x_5^* = 0$ m/s, $P_0^* = 0$ m/s 77
Figure 4.4.1.54 True minus estimated, $x_5^* = 0$ m/s, $P_0^* = 0$ m/s 77
Figure 5.1 External Velocity and Altitude Reference 80
LIST OF SYMBOLS

\( x_k \)  
Process State Vector

\( \varphi_k \)  
State Transition Matrix

\( w_k \)  
Process Noise Vector

\( z_k \)  
Measurement Vector

\( H_k \)  
Measurement to state Mapping matrix

\( v_k \)  
Measurement Noise Vector

\( Q_k \)  
Process Noise Covariance matrix

\( R_k \)  
Measurement Error Covariance matrix

\( e_k \)  
Estimation Error

\( \hat{x}_k \)  
State Estimation before Measurement

\( P_k \)  
Previous Covariance Matrix

\( K_k \)  
Kalman Gain Matrix

\( P_{k+1} \)  
State Estimate Error Covariance Matrix

\( e_{k+1} \)  
Updated estimation error

\( I \)  
Identity Matrix

\( \hat{x}_{k+1} \)  
State Estimate after Measurement

\( e_{k+1} \)  
Estimation Error after Measurement

\( P_{k+1} \)  
Covariance matrix after Measurement

\( x_k \)  
State Estimate

\( x_3 \)  
True Process Baro Estimation Error

\( x_3^- \)  
Estimated State Baro Error

\( x_1 \)  
True Process Altitude Error

\( x_1^- \)  
Estimated State Altitude Error

\( x_2 \)  
True Process East Velocity Error

\( x_2^- \)  
Estimated State East Velocity Error

\( x_5 \)  
True Process North Velocity Error

\( x_5^- \)  
Estimated State North Velocity Error
Abstract

KALMAN FILTER AND ITS APPLICATIONS IN NAVIGATION

By

Dursun Tepeli

Master of Science in Electrical Engineering

In the early 1960’s Rudolf Emil Kalman published a recursive predictive filter based on the use of state space methods and recursive algorithms. Since its discovery, it has been involved in widespread research and practical applications. The primary focus of a Kalman Filter used in navigation applications is to integrate on-board navigation sensor data to obtain the best overall system performance. The host vehicle under consideration is an aircraft. The filters function is to estimate the state variables of the system such as position or velocity using statistical methods to provide the best solution. The filter supports estimations of past, present, and even future states.

Encompassing this study was to implement the Kalman filter algorithm in MATLAB. The intent was to use this algorithm to simulate practical applications in the field of navigation.

The description of the Kalman filter and its algorithm are presented herein. The main components of the algorithm including the prediction and correction steps are applied and demonstrated. The conclusion of the study consists of two applied examples of how Kalman filtering is used to estimate Inertial Navigation System (INS) position and velocity measurements using external aiding sources.
1 Introduction

In modern aircraft navigation, it is common for the integration of the Inertial Navigation System (INS) to provide the primary aircraft navigation solution. The INS sustains attitude, position and velocity accuracy. However, the INS solution alone is inherently unstable [1]. This is true due to integration drift. Integration drift is the integration of acceleration and angular velocity along with its errors, which over time are compounded into greater errors in position. Since acceleration and angular velocity are calculated based on previous values, these errors accumulate and increase at a rate proportional to the time the previous values were calculated [2]. To stabilize, the INS solution is integrated with external sensors such as the Global Positioning System (GPS). In addition to GPS, other auxiliary sensors exist which can be used to help keep INS navigation errors bounded [2]. These auxiliary sensors may include any of the following:

1. Celestial navigation such as
   (a) A sextant to obtain independent position measurements.
   (b) An INS-mounted star tracker to detect and correct tilts and heading errors.
2. A barometric altimeter for altitude measurements.
3. A water pressure sensor for measuring the depth of a submarine.
4. Doppler Radar (onboard or ground-based).
5. Radio navigation systems, such as
   (a) LORAN (long range aid to navigation).
   (b) VOR (VHF omni-directional range).
   (c) DME (distance measuring equipment).
   (d) GNSS (global navigation satellite systems), including GPS.
Each of the above sensors provides different means to update the INS solution. This study investigates the effectiveness of the integration of the INS solution with the Doppler Velocity Sensor (DVS) solution which provides velocity measurements. An additional example will be provided to demonstrate the effectiveness of updating the INS solution with an independent source providing position measurements.

1.1 Objectives

The main objective of this study was to analyze the performance of integrated INS and DVS data. In order to integrate the INS and DVS data, a Kalman Filter algorithm was simulated in MATLAB.

To better understand the analysis of integrated INS and DVS data, a comprehensive explanation of the two systems and the essential theory are presented. The prominence is placed on INS. The DVS will also be described. Next, the mathematical algorithm and the essential theory of the Kalman Filter will be presented. Finally, the INS error model used in the development of the Kalman Filter will be presented.

In summary, the main objectives of this study are:

1. to present the theory of INS and DVS.
2. to present the theory and the mathematical model of the Kalman Filter.
3. to develop the Kalman Filter algorithm to perform the integration.
4. to present the INS Error Model used in Kalman Filter Development.
5. to analyze the performance of Kalman algorithm in terms of position and velocity errors.

1.2 Motive for Kalman Filtering in Navigation

In modern navigation, an air vehicle may have multiple on-board sensors to provide state information. This information may consist of system position, velocity, and acceleration.
Moreover, multiple sensors on-board can provide all or some of this information simultaneously. This makes the system a multiple input multiple output system.

The Kalman Filter was chosen in navigation applications due to its optimality aspect. The way we adopt its optimality here is that it is able to process all information that it is given. It has the capability to provide the best estimate of a given state, or states, provided the current value, system and sensor dynamics, the statistical description of the system and sensor measurement noise and any information about initial conditions. In the applications we wish to discover in this study, we are faced with processing data from multiple sensors. Each of these sensors all contain data corrupted to some degree by noise, biases, and any device inaccuracies. The filter is able to process this type of data by combining all of this information and any prior system information to produce the best estimate of the system while minimizing errors. In addition to being able to process large sets of information, the filter algorithm is simple and can be easily programmed using a computer. The filter algorithm consists of very few steps and does not require all previous data to be kept in storage for re-processing.

We have described why the Kalman filter is useful for applications in navigation until now. However, the filter can be used in several other non-navigation related applications. As a simple example, we can consider the application of measuring the temperature in a room. We can start by assuming the temperature in a room to be 72 degrees Fahrenheit. We also expect our assumption to be off by plus or minus two degrees. In addition to our assumption, we have a thermometer which gives us the measurement of temperature uniformly to within plus or minus 5 degrees of the true temperature. We take a single measurement with the thermometer and it reads 75 degrees. The filter simply uses a
weighted average approach to pick a point between our assumption and the thermometer reading. Therefore the filter will first calculate the weight. If our assumption variance is very large compared to the thermometer variance, we would ignore our guess, and just use the thermometer measured value. Similarly, if the thermometer variance dominated, we would not trust our thermometer readings. Let’s say it turned out we were to select our assumption since our guess is good to plus or minus two degrees compared to plus or minus 5 degrees. Next, the filter will compute the weighted average based on the calculated weight value from the prior step. The next thing the filter will calculate is the confidence in the weighted average estimate. The filter will repeat this process as long as there is measurement data. In the next step however, it will use the previous confidence variance to do the next weight, weighted average estimate, and confidence computation. This allows us not to have the entire history of the measurements and estimates and only keep track of the most recent estimate and confidence variance computations.
2 Inertial Navigation

This chapter presents the philosophy of the INS. First it talks about the overview and the history behind its development. Next, the practical realization of the system and the system components are presented. Finally, the inertial navigation solution is introduced.

2.1 INS History

Inertial navigation had its beginnings in World War II with German Cruise missiles and short-range ballistic missiles. It was the first successful long-range autonomous navigation method, requiring no operator actions once it was underway. It does not rely on any external infrastructure, it cannot be jammed or spoofed, and it is inherently stealthy. It would become the most common navigation for military and commercial aviation. It has remained the navigation method of choice for applications that cannot rely on GNSS [2].

2.2 INS Theory

The fundamental idea behind inertial navigation is that the second integral of acceleration is position. Inertial navigation begins with known initial values for position and velocity. Thereafter, acceleration is measured and integrated twice to obtain the current position [2].

2.3 Practical INS

An inertial navigation system commonly includes [2]:

1. An inertial sensor assembly (ISA) for measuring translational accelerations (three components) and rotation rates (three components). It includes
accelerometers (acceleration sensor) and gyroscopes (attitude rate sensors), all rigidly affixed to a common base.

2. Digital processing systems (one or more) for
   a. Processing sensed rotation rates to maintain knowledge of the attitude of the inertial sensors relative to a reference coordinate frame. The choice of coordinates is usually driven by the application. For navigation relative to the earth it is usually earth-fixed. For surface navigation, it may have locally level coordinates (e.g., east-north-up).
   b. Calculating and compensating for any rotation of the coordinate frame (earth rotation, for example).
   c. Calculating the local gravitational acceleration, which is not detectable by the accelerometers, and adding it to the sensed acceleration to determine net acceleration.
   d. Integrating accelerations twice—first, to determine changes in vehicle velocity, and again, to determine changes in position.
   e. Compensating the sensor outputs for known deviations in sensor scale factors, biases (output offsets), input axis misalignments, etc.
   f. Performing other operational functions such as initial alignments and processing inputs and outputs.

3. Supporting electromechanical subsystems for such functions such as
   a. Operator interfaces.
   b. Electrical power conversion and conditioning.
c. Temperature regulation.

d. Signal interfaces to other onboard systems.

4. Mechanical isolation from shock and vibration transmitted though the vehicle mounting location. In what is known as a strapdown configuration, this is the only form of mechanical isolation also includes isolation of the sensors from rotations of the host vehicle.

*Gimbaled INS* is the first standalone INS. Their attitudes location is either from a spherical bearing or a set of gimbal rings. Gyroscopes in feedback control servo loops generate torques to null out any sensed attitude changes in the inertial sensor assembly.

*Strapdown INS* use software to replace gimbals. They compute the coordinate transformations needed to resolve accelerations and attitude rates from sensor-fixed coordinates to the coordinates of the navigation problem [2].

### 2.4 INS Solution and Aiding Devices

The navigation solution for INS includes the position, velocity, and orientation (or attitude) of the host vehicle. Outputs may also include accelerations and angular rates, which are useful for automatic control and guidance of the host vehicle [2].

#### 2.4.1 INS Velocity and Position Solution

Accelerations sensed in the ISA are resolved into navigation coordinates, where they are integrated once to compute velocity increments and again to compute position increments. Depending on the type of sensors used in an INS, each will have its own error characteristics. These errors, during the integration process will propagate over
time, and will cause long term instability in the INS solution. For this reason, the INS can use external aiding source(s) such as the ones previously mentioned as a means of bounding, or damping the errors [3].

2.4.2 Doppler Radar

Doppler radar is a radar transmitter-receiver that radiates electromagnetic energy from the aircraft downward to the earth’s surface with several beams. Some of the energy is backscattered by the surface and received again by the radar receiver. The difference in the frequencies between the transmitted beam and the received reflected energy, known as the Doppler shift, correlates with the velocity of the aircraft, relative to the earth’s surface [4].

2.4.3 DVS

The DVS is a velocity-measuring system based on Doppler shift in electromagnetic wave that is reflected back from the ground (or water) after being transmitted by an aircraft. Although there are a variety of Doppler radar equipments, we shall simply consider the Doppler radar system to measure and output the aircraft horizontal velocity in terms of two orthogonal components: $V_H$ in the direction of the aircraft heading, and $V_D$ that is perpendicular to $V_H$. Therefore, the Doppler radar supplies a 2-tuple velocity measurement that may be used to aid an INS [1].
3 The Discrete Kalman Filter (KF)

This section will provide an understanding of the Kalman Filter and its recursive equations used. By recursive, it is meant that new measurements can be processed as they arrive and the result of the filter is updated accordingly. The primary role of the Kalman filter is to estimate the state of a dynamic system from a series of incomplete and noisy measurements. We will see how this is done in the forth coming equations.

3.1 KF Algorithm

The random process to be estimated is modeled in the form [1]

\[ x_k = \Phi_k x_k + w_k \]  
\[ z_k = H_k x_k + v_k \]  

Below is the notation associated with equations 3.4.1 and 3.4.2 [1]:

- \( x_k \) = (nx1) Process State Vector at time \( t_k \).
- \( \Phi_k \) = (nxn) State Transition Matrix.
- \( w_k \) = (nx1) Process white sequence noise with known covariance.
- \( z_k \) = (mx1) Measurement vector at time \( t_k \).
- \( H_k \) = (mxn) Matrix relating measurement vector and state vector.
- \( v_k \) = (mx1) measurement noise, assumed to be white sequence with known covariance.

The covariance matrices for the \( w_k \) and \( v_k \) vectors are given by [1]

\[
E[w_k w_i^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases}
\]
\[ E[v_k v_i^T] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \quad (3.4.4) \]

\[ E[w_k v_i^T] = 0 \text{ for all } k \text{ and } i \quad (3.4.5) \]

Assume we have an initial estimate of the process at \( t_k \). This prior (or a priori) estimate is denoted by \( \hat{x}_k^- \) [1]. Then, the estimation error is [1]

\[ e_k^- = x_k - \hat{x}_k^- \quad (3.4.6) \]

And the associated error covariance matrix is

\[ P_k^- = E[e_k^- e_k^-^T] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \quad (3.4.7) \]

The goal is to use the measurement to improve the prior estimate with the following equation [1]:

\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \quad (3.4.8) \]

where

\[ \hat{x}_k = \text{updated estimate} \]

\[ K_k = \text{blending factor} \]

It is worth noting that the new estimate of equation 3.4.8 is a linear blending of the noisy measurement and the prior estimate. Then, to obtain the best estimate, the minimum mean-square error is used as similar to the Wiener solution [1]. The updated error covariance matrix for the updated estimate (a posteriori) is

\[ P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \quad (3.4.9) \]

Then we substituting equation 3.4.2 into 3.4.8 yields

\[ \hat{x}_k = \hat{x}_k^- + K_k (H_k x_k + v_k - H_k \hat{x}_k^-) \quad (3.4.10) \]

Then we substitute 3.4.10 into 3.4.9 and we obtain

\[ P_k = E[e_k e_k^T] \]
\[ E[(x_k - \hat{x}_k^- + K_k(H_kx_k + v_k - H_k\hat{x}_k^-))(x_k - \hat{x}_k^- + K_k(H_kx_k + v_k - H_k\hat{x}_k^-))^T] \]

(3.4.11)

After simplifying 3.4.11 and taking the expected value of the products and realizing that \( x_k - \hat{x}_k \) is uncorrelated with the measurement error \( v_k \), we get

\[ P_k = E[(I - K_kH_k)(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T(I - H_k^T K_k^T)] + E[K_k v_k v_k^T K_k^T] \]

Using Definition 3.4.4 and equation 3.4.7 we have the updated error covariance

\[ P_k = (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \]  

(3.4.12)

Next, we must minimize the terms along the major diagonal of \( P_k \). This is important since those terms represent the estimation error variances for the elements of the state vector being estimated. The trace of \( P \) is the sum of all the mean-square errors in the estimates of all the elements of the state vector and we need to take the derivative of the trace of \( P \) with respect to \( K \). The three equations required for this procedure are as follows [1]:

a.) Given a scalar quantity, the derivative of it with respect to a matrix is defined as

\[
\frac{ds}{dA} = \begin{bmatrix}
\frac{ds}{da_{11}} & \frac{ds}{da_{12}} & \cdots & \frac{ds}{da_{1n}} \\
\frac{ds}{da_{21}} & \frac{ds}{da_{22}} & \cdots & \frac{ds}{da_{2n}} \\
\frac{ds}{da_{31}} & \frac{ds}{da_{32}} & \cdots & \frac{ds}{da_{mn}}
\end{bmatrix}
\]

b.)

\[
\frac{d[\text{trace} (AB)]}{dA} = B^T
\]

c.)

\[
\frac{d[\text{trace} (ACA^T)]}{dA} = 2AC
\]

We will simplify equation 3.4.12 and rewrite it as

\[ P_k = P_k^- - K_kH_kP_k^- - P_k^- H_k^T K_k^T + K_k(H_kP_k^- H_k^T + R_k)K_k^T \]  

(3.4.13)
Now, using equation 3.4.13 and equation b of matrix differentiation, we get $K_k$ which minimizes the mean-square estimation error known as the Kalman Gain.

$$K_k = P_k^{-1}H_k^T(H_kP_k^{-1}H_k^T + R_k)^{-1}$$

(3.4.14)

Substituting 3.4.14 into 3.4.13, we get

$$P_k = P_k^{-} - P_k^{-}H_k^T(H_kP_k^{-1}H_k^T + R_k)^{-1}H_kP_k^{-} - P_k^{-}H_k^T P_k^{-}H_k^{-1}((H_kP_k^{-1}H_k^T + R_k)^{-1})^T + P_k^{-}H_k^T(H_kP_k^{-1}H_k^T + R_k)^{-1}(H_kP_k^{-1}H_k^T + R_k)(P_k^{-}H_k^T(H_kP_k^{-1}H_k^T + R_k)^{-1})^T$$

(3.4.15)

Simplifying and rewriting 3.4.15 can yield three different forms of it:

$$P_k = P_k^{-} - P_k^{-}H_k^T(H_kP_k^{-1}H_k^T + R_k)^{-1}H_kP_k^{-}$$

(3.4.15)

$$P_k = P_k^{-} - K_k(H_kP_k^{-1}H_k^T + R_k)K_k^T$$

(3.4.16)

$$P_k = (I - K_kH_k)P_k^{-}$$

(3.4.17)

It is important to note that equations 3.4.15 through 3.4.17 are identical. Therefore, equation 3.4.8 may be computed with the Kalman gain from 3.4.14. Next we must use a similar process to derive $\hat{x}_k^-$ and $P_k^-$ for the evaluation of $\hat{x}_k$. These parameters are required to make optimal use of the measurement $z_{k+1}$. The updated estimated $\hat{x}_k$ is easily projected ahead through the transition matrix. Then we have

$$\hat{x}_{k+1}^- = \varphi_k\hat{x}_k$$

(3.4.18)

In 3.4.18 we ignore $w_k$ because it has zero mean and not correlated with any of the previous $w$'s. Then the error covariance matrix associated with $\hat{x}_{k+1}^-$ is obtained by first forming the expression for the a priori error

$$= \varphi_k e_k + w_k = (\varphi_k x_k + w_k) - \varphi_k \hat{x}_k e_{k+1}^- = x_{k+1} - \hat{x}_{k+1}^-$$

(3.4.19)

Next, we note that $w_k$ and $e_k$ have no cross correlation. Hence, we can write the expression for $P_{k+1}^-$ as

$$P_{k+1}^- = E[e_{k+1}^-e_{k+1}^T] = E[(\varphi_k e_k + w_k)(\varphi_k e_k + w_k)^T]$$
\[ = \varphi_k P_k \varphi_k^T + Q_k \]  

(3.4.20)

In summary, the following set of equations comprises the Kalman filter recursive equations [1], [3]:

\[ \hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \]

\[ K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \]

\[ P_k = (I - K_k H_k) P_k^- \]

\[ \hat{x}_{k+1}^- = \varphi_k \hat{x}_k \]

\[ P_{k+1}^- = \varphi_k P_k \varphi_k^T + Q_k \]
To better display the process of the Kalman filter recursive equations, a block diagram
has been presented in Figure 3.1.1. In the figure, the process begins by entering the initial
estimate $\hat{x}_k$ and the error covariance $P_k$ it is associated with.
4 Applications of Kalman Filter in Navigation

4.1 Kalman filter and its application on altitude Damping

The first demonstration in this study focuses on using a barometric altimeter to stabilize the altitude of an INS. The simulation is run to show that the Kalman filter can be used in this type of application. It is important to note that the Kalman filter will only operate on system errors. The model and error propagation are demonstrated.

Figure 4.5.1 shows the error model used for the vertical channel of the INS and the barometric altimeter. The model used in this study was over simplified due to demonstration purposes, however the model allows for non minimal error propagation. The INS model alone presents a double pole at the origin of the s-plane due to double integration. This causes the INS vertical channel error to grow unstable.

To stabilize the INS vertical channel, the Markov model for the baro-altitude error was used. This model is used to bound the vertical channel error. This model allows the designer to vary the variance and inverse time constant of the Markov baro-altitude error model. Additionally, the user may also control the rate of growth of the INS by varying the spectral amplitude of the white-noise driving function.

The differential equations and state space model will be presented of the stabilizing model which is based on the block diagram of Figure 4.1.1. The three state variables necessary for the description:
\[ x_1 = \text{INS vertical position error (m)} \]
\[ x_2 = \text{INS vertical Velocity error (m/sec)} \]
\[ x_3 = \text{baro error (m)} \]

Figure 4.1.1 Error models used for the vertical channel

The appropriate differential equations are obtained directly from the block diagram, and they are

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f_1(t) \\
\dot{x}_3 &= -\beta x_3 + \frac{\sqrt{2} \sigma^2 \beta}{s + \beta} f_2(t)
\end{align*}
\]

These differential equations can be solved for a step size \( \Delta t \), and the result in matrix form is
\[
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{bmatrix}_{k+1} =
\begin{bmatrix}
    1 & \Delta t & 0 \\
    0 & 1 & 0 \\
    0 & 0 & e^{-\beta \Delta t}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{bmatrix}_k +
\begin{bmatrix}
    w_1 \\
    w_2 \\
    w_3
\end{bmatrix}_k
\]

The state transition matrix is solved for small $\Delta t$ as follows

\[
\varphi_k =
\begin{bmatrix}
    1 & \Delta t & 0 \\
    0 & 1 & 0 \\
    0 & 0 & e^{-\beta \Delta t}
\end{bmatrix}
\]

The covariance matrix is solved as

\[
Q_k =
\begin{bmatrix}
    \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\
    \frac{\Delta t^2}{2} & A\Delta t & 0 \\
    0 & 0 & \sigma^2(1 - e^{-2\beta \Delta t})
\end{bmatrix}
\]

$A$ is the power spectral density of $f_1$ and $\sigma^2$ and $\beta$ are the Markov parameters for the baro error process. The measurement relationship is also obtained from the block diagram in Figure 4.1.1. The measurement relationship is

\[
(\text{Measurement presented to the Kalman filter}) =
(\text{true altitude} + \text{baro error}) - (\text{true altitude} - \text{INS altitude error})
\]

The measurement relationship is written in state space form as follows

\[
z_k = [1 0 1] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v_k
\]
4.2 Baro-altitude filter Simulation

The Kalman filter model for integrating vertical acceleration and baro-altitude measurements were presented in section 4.1. We will now consider using the preceding numerical values to run the simulation.

\[ \Delta t \text{ step size} = 1 \text{ sec} \]

\[ \text{Accelerometer white – noise spectral density} = 0.13889 \left( \frac{m}{\text{sec}^2} \right) \left( \frac{\text{sec}}{\text{rad}} \right) \]

\[ \text{Markov baro error variance and inverse time constant:} \]

\[ \sigma^2 = (100 m)^2 \]

\[ \beta^{-1} = 300 \text{ sec} \]

\[ \text{White component of baro error} = (10 m)^2 \]

For this simulation we will assume that our system is turned on at \( t = 0 \) with initial estimate of altitude set to 0 meters, vertical velocity set to 0 m/sec, and barometer error set to 0 meters for the estimated state. Additionally, the initial conditions for estimated error covariance matrix were set equal to 0. Post simulation, the filter reached steady-state nearly around 1000 seconds. Figure 4.2.1.1 shows the resulting mean square error for the estimated state.

Filter outputs are presented. Figure 4.2.1.2 shows the true and estimated values of altitude, Figure x2 shows the baro error and its estimated filter value, Figure 4.2.1.3 shows the filter residual, which is defined as \((z_k - H_k x_k)\), Figure 4.2.1.4 shows the position error which is defined as the difference between true and filter estimated state. According to Brown, the residual reflects the discrepancy between the predicted measurement and the actual. He also asserts that a residual of 0 implies that the two are in
complete agreement. As shown in the resulting figures the discrepancy between the two are relatively small. Figure 4.2.1.5 shows the baro error RMS compared to the filter estimated baro error after a simulation of 5040 seconds (Schuler Cycle). In this Figure we can see that it takes nearly 836 seconds for the estimated value to reach the true value. However, we can see for the remainder of the simulation the RMS error of the filter estimate has been reduced compared to the truth. As expected, the filter does a really good job of tracking the true state.

For experimentation and to demonstrate the effectiveness of the model, different values of initial estimate and covariance values are used. The results are presented in tables 4.2.1 through 4.2.4.

Table 4.2.1 Mean Residual and Error $x_1= 10 \text{ m}, P_0^* = 0 \text{ m}$

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>500</td>
<td>0.07419</td>
</tr>
</tbody>
</table>

Table 4.2.2 Mean Residual and Error $x_1= 10 \text{ m}, P_0^* = 10 \text{ m}$

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>0</td>
<td>0.09932</td>
</tr>
</tbody>
</table>
Table 4.2.3 Mean Residual and Error $x_1= 0$ m, $P_0^\circ = 0$ m

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>0</td>
<td>0.08539</td>
</tr>
</tbody>
</table>

Table 4.2.4 Mean Residual and Error $x_1= 0$ m, $P_0^\circ = 10$ m

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>0</td>
<td>0.09608</td>
</tr>
</tbody>
</table>

As shown in the tables, the filter yields the best performance when the initial covariance diagonal values are set to 10. More specifically, the overall best performance is shown in table 4.2.4 where the mean error is the lowest, and the mean residual error is relatively low. Simulation results are plotted in the next section corresponding to the parameters chosen in these tables.

4.2.1 Simulation Results

The simulation results associated with the parameters in Table 4.2.1 – 4.2.4 are shown in figures 4.2.1.6 through 4.2.18. Figure 4.2.1.1 displays the mean square error of the filter estimated altitude error. Figure 4.2.1.2 displays the true and filter estimated altitude error with initial altitude error and covariance diagonal values set to be 0. Figure 4.2.1.3 displays the residual $z$, with initial altitude error and covariance diagonal values set to be 0. Figure 4.2.1.4 displays the difference between the true state and the filter estimate of altitude error. Figure 4.2.1.5 displays the true state and estimated Baro altitude error.
RMS. The remainder of the figures displays various configurations of the filter with various initial state and covariance diagonal values. Figure 4.2.1.6 through Figure 4.2.1.8 displays true and filter estimated altitude error, residual z, and the difference between true and filter estimated altitude error for initial altitude set to 10 m and covariance diagonals set to 0 m respectively. Figure 4.2.1.9 through Figure 4.2.1.11 displays true and filter estimated altitude error, residual z, and the difference between true and filter estimated altitude error for initial altitude set to 10 m and covariance diagonals set to 10 m respectively. Figure 4.2.1.12 through Figure 4.2.1.14 displays true and filter estimated altitude error, residual z, and the difference between true and filter estimated altitude error for initial altitude set to 0 m and covariance diagonals set to 0 m respectively. Figure 4.2.1.15 through Figure 4.2.1.17 displays true and filter estimated altitude error, residual z, and the difference between true and filter estimated altitude error for initial altitude set to 0 m and covariance diagonals set to 10 m respectively.
Figure 4.2.1.1 Mean Square Error of Estimated State Altitude
Figure 4.2.1.2 True and Estimated Altitude
Figure 4.2.1.3 Residual $z_k = H_k x_k$
Figure 4.2.1.4 True State minus Estimated State
Figure 4.2.1.5 True and estimated Baro Error RMS
Figure 4.2.1.6 True and estimated states, $x_1=10\text{m}$, $P_0=0\text{m}$
Figure 4.2.1.7 Residual $z$, $x_1=10 \text{m}$, $P_0=0 \text{m}$
Figure 4.2.1.8 True minus estimate, \( x_1 = 10 \text{m}, P_0 = 0 \text{m} \)
Figure 4.2.1.9 True and estimated states, $x_1=10\text{m}$, $P_0=10\text{m}$
Figure 4.2.1.10 Residual $z_k$, $x_1=10m$, $P_0=10m$
Figure 4.2.1.11 True minus estimate, $x_1=10\text{m}$, $P_0=10\text{m}$
Figure 4.2.1.12 True and estimated states, $x_1=0\text{m}$, $P_0=0\text{m}$
Figure 4.2.1.13 Residual $z$, $x_1=0$ m, $P_0^*=0$ m
Figure 4.2.1.14 True minus estimate, $x_i=0$ m, $P_0=0$ m
Figure 4.2.1.15 True and estimated states, $x_1=0$ m, $P_0=10$ m
Figure 4.2.1.16 Residual z, $x_1=0m$, $P_0^- = 10m$
4.3 Kalman filter and its application on external velocity aiding

Per example 10.1 in Hwang, we will generalize the Schuler-Damping problem to the full 9-state INS model. Doppler radar is used to measure and output the aircraft horizontal velocity in terms of two orthogonal components: $V_H$ in the direction of the aircraft heading and $V_D$ that is perpendicular to $V_H$ (see Figure 4.4.1). The radar supplies a two state velocity measurement used to aid the INS. In this example, we will use a linearized $H_k$ matrix. The measurement noise is assumed to be white, and the INS errors are modeled as a 9-state system.

4.4 INS/Doppler Radar Measurement Model Simulation

It is important to note in this example that the INS must resolve its indicated horizontal velocity components $V_x$ and $V_y$ into the aircraft body frame of reference, which we will
call heading and drift. The coordinate directions are shown in Figure 4.4.1. We will also assume that the INS platform azimuth angle $\beta$ is used to resolve $V_x$ and $V_y$ into $V_H$ and $V_D$. This resolution yields the INS predicted values of the two radar measurements. The idealized noiseless relationships are as follows:

\[
\begin{align*}
V_H &= -V_x \sin \beta + V_y \cos \beta \\
V_D &= -V_x \cos \beta + V_y \sin \beta
\end{align*}
\]

Figure 4.4.1 Aircraft velocity vector convention

The above equations must be linearized. However, we will first introduce the 9 state INS error model first. Figure 4.4.2 shows the North Channel error model block diagram. A similar model is shown in Figure 4.4.3 for the east channel. The differential equations used from the transfer function of the two block diagrams are as follows:
Figure 4.4.2 North Channel error model (x is east, y is north, z is up, and $\omega_y = \text{platform angular rate about y-axis}$)

Figure 4.4.3 East Channel error model (x is east, y is north, z is up, and $\omega_x = \text{platform angular rate about x-axis}$)
The platform rotation rate is $\omega$ which is an angular velocity. The platform tilt rate error is $\Phi$. The true north and east channel models take into account earth curvature. The vertical channel is governed by a simpler model as shown in Figure 4.4.4. The basic 9-state dynamic model is used for an aided INS Kalman filter model. The nine state variables are

\begin{align*}
x_1 &= \text{east position error (m)} \\
x_2 &= \text{east velocity error} \left(\frac{m}{\text{sec}}\right) \\
x_3 &= \text{platform tilt about y axis (rad)} \\
x_4 &= \text{north position error (m)} \\
x_5 &= \text{north velocity error} \left(\frac{m}{\text{sec}}\right) \\
x_6 &= \text{platform tilt about (-x)axis (rad)} \\
x_7 &= \text{vertical position error (m)} \\
x_8 &= \text{vertical velocity error} \left(\frac{m}{\text{sec}}\right) \\
x_9 &= \text{platform azimuth error (rad)}
\end{align*}

The nine dimensional state space model is

**North Channel**
\[
\Delta \dot{y} = a_y - g(-\varphi_x) \\
-\dot{\varphi}_x = \frac{1}{R_e} \Delta \dot{y} + \omega_y \varphi_z - \varepsilon_x
\]

**East Channel**
\[
\Delta \ddot{x} = a_x - g(\varphi_y) \\
\dot{\varphi}_y = \frac{1}{R_e} \Delta \ddot{x} + \omega_x \varphi_z + \varepsilon_y
\]

**Vertical Channel**
\[
\Delta \ddot{z} = a_z
\]

**Platform Azimuth**
\[
\varphi_z = \varepsilon_z
\]
In the above expression, $R_e$ represents radius of the earth in meters and $g$ represents earth acceleration in meters. The white noise input forcing functions are: $u_{ax}$, $u_{gx}$, $u_{ay}$, $u_{gy}$, $u_{az}$ and $u_{gz}$, which are used for acceleration and gyroscope $x$, $y$ and $z$-axis inputs respectively. The state transition matrix $\Phi_{INS}$ and the covariance matrix $Q_k$ are calculated using the Loan procedure outlined in Hwang. We first begin by forming the $2n \times 2n$ matrix what we will call $A$, where $n$ is the dimension of $x$.

$$A = \begin{bmatrix} -F \Delta t & GWG^T \Delta t \\ \vdots & \vdots \\ 0 & F^T \Delta t \end{bmatrix} \Delta t$$

The matrix $W$ is the power spectral density associated with the input forcing functions. Next, we will use MATLABs built in function, expm, we will form $e^A$ and call it $B$. The function expm evaluates the exponential of a matrix.

$$B = expm(A) = \begin{bmatrix} \vdots & \varphi^{-1}Q_k \\ \vdots & \vdots \\ 0 & F^T \end{bmatrix}$$

To get $\Phi_{INS}$, we must transpose the lower-right partition of $B$.

$$\varphi = \text{transpose of lower-right partition of } B$$
Finally, $Q_k$ is obtained from the upper-right partition of $B$ as follows:

$$Q_k = \varphi \ast (\text{Upper } - \text{right part of } B)$$

We will next focus our attention to the measurement relationship, which in mathematical form is

$$(\text{Measurement presented to the Kalman filter})
= -(\text{true velocity} - \text{INS velocity error})
+ (\text{true velocity} + \text{reference velocity error})
= (\text{INS velocity error}) + (\text{reference velocity error})$$

In order to derive the above notation in state space form we must return to the linearization of the $H_k$ matrix. This is done by forming the appropriate matrix of partial derivatives of $V_H$ and $V_D$ with respect to $V_x$, $V_y$ and $\beta$. The resulting linearized $H_k$ matrix is then

$$H_k = \begin{bmatrix} 0 & -\sin \beta & 0 & 0 & \cos \beta & 0 & 0 & 0 & (-V_x \cos \beta - V_y \sin \beta) \\ 0 & \cos \beta & 0 & 0 & \sin \beta & 0 & 0 & 0 & (-V_x \sin \beta + V_y \cos \beta) \end{bmatrix}$$

The two radar measurements have been ordered with the heading component of velocity first and its drift component second. The terms in the $H_k$ matrix are time-varying and must be calculated with each step of the Kalman filter. The calculations are made using the most current measurements of the INS $V_x$, $V_y$ and $\beta$. We can now write the state space form of the measurement relationship as
\[ z_k = \begin{bmatrix} 0 & -\sin \beta & 0 & 0 & \cos \beta & 0 & 0 & 0 & (-V_x \cos \beta - V_y \sin \beta) \\ 0 & \cos \beta & 0 & 0 & \sin \beta & 0 & 0 & 0 & (-V_x \sin \beta - V_y \cos \beta) \end{bmatrix} x_k + \begin{bmatrix} v_{k1} \\ v_{k2} \end{bmatrix} \]

It is known that the Kalman filter operation is at its most optimum when it has an initial precise estimate of the aircrafts velocity. However, another benefit of the Kalman filter is that it can compensate for a not so accurate initial estimate. Through experimentation, the simulation has shown that for different values of the initial condition for estimated velocity the Kalman filter has been able to reduce the error to steady state. The experiment was run with three different values of initial velocity estimates: 500 m/s, 100 m/s, and 0. Additionally, the initial error covariance matrix diagonals were adjusted. Three different values were used corresponding to each value of the initial velocity estimate. So, for example, at initial east velocity estimate of 500 m/s, the covariance matrix was adjusted to 1000, 100, and 0. Similar cases were set up for the remaining two initial velocity estimate values. The plots for these experiments are shown in Figure 4.4.1.1 through 4.4.1.54 and the results are shown in tables 4.1 through 4.9.

Table 4.1 Mean Residual and Error \( x_2' = x_5' = 500 \text{ m/s}, P_0' = 1000 \text{ m/s} \)

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Velocity</td>
<td>4</td>
<td>-0.001545</td>
</tr>
<tr>
<td>North Velocity</td>
<td>6.5</td>
<td>0.002733</td>
</tr>
</tbody>
</table>
### Table 4.2 Mean Residual and Error $x_2^- = x_5^- = 500$ m/s, $P_0^- = 100$ m/s

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>East Velocity</strong></td>
<td>14.8</td>
<td>0.0003826</td>
</tr>
<tr>
<td><strong>North Velocity</strong></td>
<td>17</td>
<td>0.002489</td>
</tr>
</tbody>
</table>

### Table 4.3 Mean Residual and Error $x_2^- = x_5^- = 500$ m/s, $P_0^- = 0$ m/s

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>East Velocity</strong></td>
<td>5</td>
<td>-4.089</td>
</tr>
<tr>
<td><strong>North Velocity</strong></td>
<td>5</td>
<td>-5.786</td>
</tr>
</tbody>
</table>

### Table 4.4 Mean Residual and Error $x_2^- = x_5^- = 100$ m/s, $P_0^- = 1000$ m/s

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>East Velocity</strong></td>
<td>10</td>
<td>-0.001648</td>
</tr>
<tr>
<td><strong>North Velocity</strong></td>
<td>6</td>
<td>0.002701</td>
</tr>
</tbody>
</table>
### Table 4.5 Mean Residual and Error $x_2^* = x_5^* = 100 \text{ m/s}, P_0^* = 100 \text{ m/s}$

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Velocity</td>
<td>14</td>
<td>-0.001608</td>
</tr>
<tr>
<td>North Velocity</td>
<td>12</td>
<td>0.002647</td>
</tr>
</tbody>
</table>

### Table 4.6 Mean Residual and Error $x_2^* = x_5^* = 100 \text{ m/s}, P_0^* = 0 \text{ m/s}$

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Velocity</td>
<td>6</td>
<td>-0.7059</td>
</tr>
<tr>
<td>North Velocity</td>
<td>6</td>
<td>-1.025</td>
</tr>
</tbody>
</table>
Table 4.7 Mean Residual and Error $x_2^- = x_5^- = 0$ m/s, $P_0^- = 1000$ m/s

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Velocity</td>
<td>10</td>
<td>-0.001767</td>
</tr>
<tr>
<td>North Velocity</td>
<td>6</td>
<td>0.002699</td>
</tr>
</tbody>
</table>

Table 4.8 Mean Residual and Error $x_2^- = x_5^- = 0$ m/s, $P_0^- = 100$ m/s

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Velocity</td>
<td>14</td>
<td>-0.002105</td>
</tr>
<tr>
<td>North Velocity</td>
<td>7</td>
<td>0.002687</td>
</tr>
</tbody>
</table>

Table 4.9 Mean Residual and Error $x_2^- = x_5^- = 0$ m/s, $P_0^- = 0$ m/s

<table>
<thead>
<tr>
<th>Settle Time (sec)</th>
<th>Mean Residual (m/s)</th>
<th>Mean Error (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Velocity</td>
<td>4</td>
<td>0.1399</td>
</tr>
<tr>
<td>North Velocity</td>
<td>4</td>
<td>0.1653</td>
</tr>
</tbody>
</table>

Based on the results, the Kalman Filter yields the lowest error when the initial velocity and initial error covariance diagonals are set to 0. At this setting, the mean error, settling time, and residual errors are the lowest. The filter yields the worst performance in terms of residual and mean error when the initial velocity estimate and error covariance diagonals are set to 500 m/s and 0 m/s respectively. However, we have a relatively low
settling time. As stated by Hwang, the state variables are initial error quantities so the initial error estimates and error covariance terms are normally set to 0. This allows us to model a system representing initial uncertainty in the state estimates and was shown to have the best performance in the results tables.

### 4.4.1 Simulation Results

The figures in this section display simulation results. Various values have been used for initial x and y velocity and covariance diagonal values. The results of the various combinations are displayed in the following figures. Figure 4.4.1.1 through Figure 4.4.1.3 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 500 m/s and covariance diagonals set to 1000 m/s. Figure 4.4.1.4 through Figure 4.4.1.6 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 500 m/s and covariance diagonals set to 1000 m/s. Figure 4.4.1.7 through Figure 4.4.1.9 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 500 m/s and covariance diagonals set to 100 m/s. Figure 4.4.1.10 through Figure 4.4.1.12 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 500 m/s and covariance diagonals set to 100 m/s. Figure 4.4.1.13 through Figure 4.4.1.15 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 500 m/s and covariance diagonals set to 0 m/s. Figure 4.4.1.16 through Figure 4.4.1.18 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated
y-velocity with initial y-velocity set to 500 m/s and covariance diagonals set to 0 m/s. Figure 4.4.1.19 through Figure 4.4.1.21 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 100 m/s and covariance diagonals set to 1000 m/s. Figure 4.4.1.22 through Figure 4.4.1.24 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 100 m/s and covariance diagonals set to 1000 m/s. Figure 4.4.1.25 through Figure 4.4.1.27 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 100 m/s and covariance diagonals set to 100 m/s. Figure 4.4.1.28 through Figure 4.4.1.30 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 100 m/s and covariance diagonals set to 100 m/s. Figure 4.4.1.31 through Figure 4.4.1.33 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 100 m/s and covariance diagonals set to 0 m/s. Figure 4.4.1.34 through Figure 4.4.1.36 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 100 m/s and covariance diagonals set to 0 m/s. Figure 4.4.1.37 through Figure 4.4.1.39 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 0 m/s and covariance diagonals set to 1000 m/s. Figure 4.4.1.40 through Figure 4.4.1.42 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 0 m/s and covariance diagonals set to 1000 m/s.
Figure 4.4.1.43 through Figure 4.4.1.45 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 0 m/s and covariance diagonals set to 100 m/s. Figure 4.4.1.46 through Figure 4.4.1.48 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 0 m/s and covariance diagonals set to 100 m/s. Figure 4.4.1.49 through Figure 4.4.1.51 displays true and filter estimated x-velocity, the residual z, and the difference between true and filter estimated x-velocity with initial x-velocity set to 0 m/s and covariance diagonals set to 0 m/s. Figure 4.4.1.52 through Figure 4.4.1.54 displays true and filter estimated y-velocity, the residual z, and the difference between true and filter estimated y-velocity with initial y-velocity set to 0 m/s and covariance diagonals set to 0 m/s.

Figure 4.4.1.1 True and estimated x-velocity, $x_2^- = 500$ m/s, $P_0^- = 1000$ m/s
Figure 4.4.1.2 Residual $z_1, x_2^- = 500 \text{ m/s}, P_0^- = 1000 \text{ m/s}$
Figure 4.4.1.3 True minus estimated, $x_2^- = 500$ m/s, $P_0^- = 1000$ m/s

Figure 4.4.1.4 True and estimated y-velocity, $x_5^- = 500$ m/s, $P_0^- = 1000$ m/s
Figure 4.4.1.5 Residual $z_2$, $x_5^* = 500 \text{ m/s}$, $P_0^- = 1000 \text{ m/s}$

Figure 4.4.1.6 True minus estimated, $x_5^* = 500 \text{ m/s}$, $P_0^- = 1000 \text{ m/s}$
Figure 4.4.1.7 True and estimated x-velocity, $x_2^- = 500 \text{ m/s}$, $P_0^- = 100 \text{ m/s}$

Figure 4.4.1.8 Residual $z_2$, $x_2^- = 500 \text{ m/s}$, $P_0^- = 100 \text{ m/s}$
Figure 4.4.1.9 True minus estimated, $x_2^- = 500$ m/s, $P_0^- = 100$ m/s

Figure 4.4.1.10 True and estimated y-velocity, $x_5^- = 500$ m/s, $P_0^- = 100$ m/s
Figure 4.4.11 Residual $z_2$, $x_5^- = 500 \text{ m/s}$, $P_0^- = 100 \text{ m/s}$

Figure 4.4.12 True minus estimated, $x_5^- = 500 \text{ m/s}$, $P_0^- = 100 \text{ m/s}$
Figure 4.4.1.13 True and estimated x-velocity, $x_2^- = 500$ m/s, $P_0^- = 0$ m/s

Figure 4.4.1.14 Residual $z_1$, $x_2^- = 500$ m/s, $P_0^- = 0$ m/s
Figure 4.4.1.15 True minus estimated, $x_2^v = 500$ m/s, $P_0^v = 0$ m/s

Figure 4.4.1.16 True and estimated y-velocity, $x_5^v = 500$ m/s, $P_0^v = 0$ m/s
Figure 4.4.1.17 Residual $z_2$, $x_5^- = 500$ m/s, $P_0^- = 0$ m/s

Figure 4.4.1.18 True minus estimated, $x_5^- = 500$ m/s, $P_0^- = 0$ m/s
Figure 4.4.1.19 True and estimated x-velocity, $x_2^- = 100 \text{ m/s}, P_0^- = 1000 \text{ m/s}$

Figure 4.4.1.20 Residual $z_1$, $x_2^- = 100 \text{ m/s}, P_0^- = 1000 \text{ m/s}$
Figure 4.4.1.21 True minus estimated, $x_2^- = 100$ m/s, $P_0^- = 1000$ m/s

Figure 4.4.1.22 True and estimated y-velocity, $x_5^- = 100$ m/s, $P_0^- = 1000$ m/s
Figure 4.4.1.23 Residual $z_2$, $x_5^- = 100 \text{ m/s}$, $P_0^- = 1000 \text{ m/s}$

Figure 4.4.1.24 True minus estimated, $x_5^- = 100 \text{ m/s}$, $P_0^- = 1000 \text{ m/s}$
Figure 4.4.1.25 True and estimated x-velocity, $x_2 = 100$ m/s, $P_0 = 100$ m/s

Figure 4.4.1.26 Residual $z_1$, $x_2 = 100$ m/s, $P_0 = 100$ m/s
Figure 4.4.1.27 True minus estimated, $x_2^* = 100$ m/s, $P_0^* = 100$ m/s

Figure 4.4.1.28 True and estimated y-velocity, $x_5^* = 100$ m/s, $P_0^* = 100$ m/s
Figure 4.4.1.29 Residual $z_2$, $x_5^- = 100$ m/s, $P_0^- = 100$ m/s

Figure 4.4.1.30 True minus estimated, $x_5^- = 100$ m/s, $P_0^- = 100$ m/s
Figure 4.4.1.31 True and estimated x-velocity, $x_2^- = 100$ m/s, $P_0^- = 0$ m/s

Figure 4.4.1.32 Residual $z_1$, $x_2^- = 100$ m/s, $P_0^- = 0$ m/s
Figure 4.4.1.33 True minus estimated, $x_2^* = 100$ m/s, $P_0^* = 0$ m/s

Figure 4.4.1.34 True and estimated y-velocity, $x_5^* = 100$ m/s, $P_0^* = 0$ m/s
Figure 4.4.1.35 Residual $z_2$, $x_5^* = 100$ m/s, $P_0^* = 0$ m/s

Figure 4.4.1.36 True minus estimated, $x_5^* = 100$ m/s, $P_0^* = 0$ m/s
Figure 4.4.1.37 True and estimated x-velocity, $x_2^* = 0 \text{ m/s}, P_0^* = 1000 \text{ m/s}$

Figure 4.4.1.38 Residual $z_1$, $x_2^* = 0 \text{ m/s}, P_0^* = 1000 \text{ m/s}$
Figure 4.4.1.39 True minus estimated, $x_2^- = 0$ m/s, $P_0^- = 1000$ m/s

Figure 4.4.1.40 True and estimated y-velocity, $x_5^- = 0$ m/s, $P_0^- = 1000$ m/s
Figure 4.4.1.41 Residual $z_2$, $x_5^* = 0$ m/s, $P_0^* = 1000$ m/s

Figure 4.4.1.42 True minus estimated, $x_5^* = 0$ m/s, $P_0^* = 1000$ m/s
Figure 4.4.1.43 True and estimated x-velocity, $x_2^* = 0$ m/s, $P_0^- = 100$ m/s

Figure 4.4.1.44 Residual $z_1$, $x_2^* = 0$ m/s, $P_0^- = 100$ m/s
Figure 4.4.1.45 True minus estimated, $x_2^- = 0 \text{ m/s}$, $P_0^* = 100 \text{ m/s}$

Figure 4.4.1.46 True and estimated y-velocity, $x_5^- = 0 \text{ m/s}$, $P_0^* = 100 \text{ m/s}$
Figure 4.4.1.47 Residual $z_2$, $x_5^- = 0$ m/s, $P_0^- = 100$ m/s

Figure 4.4.1.48 True minus estimated, $x_5^- = 0$ m/s, $P_0^- = 100$ m/s
Figure 4.4.1.49 True and estimated x-velocity, $x_2^* = 0 \text{ m/s}, P_0^* = 0 \text{ m/s}$

Figure 4.4.1.50 Residual $z_1$, $x_2^* = 0 \text{ m/s}, P_0^* = 0 \text{ m/s}$
Figure 4.4.1.51 True minus estimated, $x_2^- = 0 \text{ m/s, } P_0^- = 0 \text{ m/s}$

Figure 4.4.1.52 True and estimated y-velocity, $x_5^- = 0 \text{ m/s, } P_0^- = 0 \text{ m/s}$
Figure 4.4.1.53 Residual $z_2$, $x_5^- = 0 \text{ m/s}$, $P_0^- = 0 \text{ m/s}$

Figure 4.4.1.54 True minus estimated, $x_5^- = 0 \text{ m/s}$, $P_0^- = 0 \text{ m/s}$
5 Conclusion and Future Work

5.1 Conclusion Remarks

In this project we were able to demonstrate the Kalman filter can be effectively used for aircraft navigation applications. It is possible to obtain an estimate of an aircraft velocity and altitude to within 1 m/s and 2 m/s with reasonable consistency for the east and north velocities respectively using the DVS aided system.

The barometric aided INS consists of a barometer installed in an aircraft which periodically measures altitude. These measurements are used to periodically update the INS vertical channel through the Kalman filter. For this particular application, the Kalman Filter algorithm may be implemented through a computer processing unit. Similarly, DVS is another sensor which is installed on an aircraft. Like the barometer, it also provides measurements to the INS through the Kalman filter. However, unlike the barometer, DVS provides measurements which need to be linearized. These measurements were successfully linearized through the measurements matrix of the Kalman filter algorithm.

Regardless of the nominal results from the simulation, there is room for improvement. It is important to emphasize that the model used in this project allows for very little complexity in errors associated with the accelerometers and gyros. This means that platform angular and torque rates are assumed to be small and change very slowly. More fidelity can be added to the model used here. This is done by adding more states to augment the state vector. The additional states consist of error time correlated errors for each axis of the acceleration and gyro rates.
The goal of this project was to create and develop a Kalman Filter to update the INS altitude and a Kalman Filter to update the INS velocity. This goal was successfully achieved as shown in the simulation results.

5.2 Future Considerations

The primary purpose of this project is to develop and implement Kalman filters which are used in altitude and velocity aiding of an INS in an aircraft navigation system. It is determined that for altitude aiding one can use a barometer and similarity for velocity aiding, a DVS. However, these systems are designed to have only one type of aiding source at a given moment. This type of system can be redesigned to integrate both aiding sensors into one navigation system.

To improve this design, one can begin with a single axis channel of the INU. We may provide the INS with an independent non-inertial source of velocity and altitude information. The block diagram of Figure 10.6 from Hwang can be used and modified to suit our new design requirement. The key portions of the diagram would be the components where the INS velocity and altitude are compared with the reference velocity and altitude respectively, and the difference fed back to through a scale factor to the accelerometer input. Similar to our previous models, the Kalman filter is to estimate the inertial system errors. This means that we must model random processes in state space form. The state space equations can be derived from the model in Figure 5.1.
Figure 5.1 External Velocity and Altitude Reference

Appropriate state equations are obtained by still choosing the three integrator outputs as state variables. They will be denoted as $x_1$, $x_2$, and $x_3$ which will be defined as follows:

$$x_1 = INS \text{ Vertical Position Error}$$

$$x_2 = INS \text{ Vertical Velocity Error}$$

$$x_3 = \text{platform tilt}$$

Then, the differential equations are derived from the block diagram.

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -gx_3 + f_1(t) - K(x_2 - f_2(t)) - K_1(x_1 - f_3(t))$$
\[ \dot{x}_3 = \frac{x_2}{R} \]

The measurement presented to the Kalman filter is the difference between INS velocity, aiding reference velocity, INS altitude, and aiding reference altitude. We now put the measurement relationship into a mathematical form then into state space.

\[(Measurement \ presented \ to \ the \ Kalman \ filter)\]
\[= -(true \ vertical \ velocity - INS \ vertical \ velocity \ error)\]
\[+ (true \ vertical \ velocity + reference \ velocity \ error)\]
\[- (true \ vertical \ position - INS \ vertical \ position \ error)\]
\[+ (true \ vertical \ position + reference \ vertical \ position \ error)\]
\[= (INS \ vertical \ velocity \ error) + (INS \ vertical \ position \ error)\]
\[+ (reference \ vertical \ velocity \ error)\]
\[+ (reference \ vertical \ position \ error)\]

\[z_k = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix}\]

The way we would expect this system to work is to continuously update the INS with vertical velocity and position data. To achieve even better results, one can have several aiding devices of the same sort. For example, a high accuracy Global Positioning System (GPS) device has the capability to provide position and velocity updates. Hence, one can obtain altitude updates from either the barometer or GPS. However, if there is ever a scenario where the GPS is not able to track its required satellites, position accuracy will degrade. Consequentially one can choose to use barometric altitude. Similarly, velocity can be obtained from a DVS as mentioned earlier in the report or a Synthetic Aperture Radar (SAR).
REFERENCES

[1] Introduction to Random Signals and Applied Kalman Filtering by Brown, R.G. and Hwang, Patrick
[5] Stochastic Models, Estimation, and Control, Volume 1 by Peter S. Maybeck
APPENDIX

PLOT GENERATION CODE

Plotting code for Barometer Model

```matlab
figure; plot(trueX.time,trueX.signals.values,'b+',xhat.time,xhat.signals.values(:,1),'g.'); %true and estimate
xlabel('Time (secs)'); grid;
ylabel('Altitude [m]'); title('Altitude'); legend('x_1','x_1^-')

figure; plot(trueBaroError.time,squeeze(trueBaroError.signals.values),xhat.time,xhat.signals.values(:,3),'g.'); %true and estimate
xlabel('Time (secs)'); grid;
ylabel('Baro Error'); title('Baro Error x3'); legend('True','Estimate')

figure; plot(Res.time,squeeze(Res.signals.values)); %res z1
xlabel('Time (secs)'); grid;
ylabel('Residual z'); title('z_k-H_kx_k^-');

figure; plot(trueX.time,trueX.signals.values-xhat.signals.values(:,1)); %true-est
xlabel('Time (secs)'); grid;
ylabel('m'); title('Error'); title('x_k - x_k^-')

figure; plot(xhatMSE.time,squeeze(xhatMSE.signals.values));
xlabel('Time (secs)'); grid;
ylabel('MSE Altitude [m]'); title('MSE of Estimated State')

figure; plot(xhatPredictedMSE.time,squeeze(xhatPredictedMSE.signals.values));
xlabel('Time (secs)'); grid;
ylabel('MSE'); title('MSE of Predicted State')

figure; plot(trueBaroError.time,squeeze(baroErrorRMS.signals.values),xhatRMS.time,xhatRMS.signals.values(:,3));
xlabel('Time (secs)'); grid;
ylabel('RMS'); title('RMS of x_3 and x_3^-'); legend('x_3 RMS','x_3^- RMS')
```
Plotting code for DVS model

```matlab
figure;plot(truexvel.time,squeeze(truexvel.signals.values),'b+',xhat.time,xhat.signals.values(:,2),'g+'); %true and estimate
xlabel('Time (secs)'); grid;
ylabel('Velocity (m/s)');title('x vel');legend('x_2','x_2^-')

figure;plot(res.time,res.signals.values(:,1)); %res z1
xlabel('Time (secs)'); grid;
ylabel('Residual z1'); title('z_k-H_kx_k^-')

figure;plot(truexvel.time,squeeze(truexvel.signals.values)-xhat.signals.values(:,2)); %true-est
xlabel('Time (secs)'); grid;
ylabel('m/s');title('x_2-x_2^-')

figure;plot(trueyvel.time,squeeze(trueyvel.signals.values),xhat.time,xhat.signals.values(:,5),'g+'); %true and estimate
xlabel('Time (secs)'); grid;
ylabel('Velocity (m/s)');title('y vel');legend('x_5','x_5^-')

figure;plot(res.time,res.signals.values(:,2)); %res z1
xlabel('Time (secs)'); grid;
ylabel('Residual z2'); title('z_k-H_kx_k^-')

figure;plot(trueyvel.time,squeeze(trueyvel.signals.values)-xhat.signals.values(:,5)); %true-est
xlabel('Time (secs)'); grid;
ylabel('m/s');title('x_5-x_5^-')
```
Simulink M-code for Baro model

dt=1;
nse=.13889;
beta=1/300;
F=[0 1 0;0 0 0;0 0 0 -beta]
GWGT=[0 0 0;0 nse 0;0 0 2*10000*beta]
A=[-dt*F dt*GWGT;zeros(3) dt*F']
B=expm(A);
PHIT=B(4:6,4:6);
PHI=PHIT'
Q=PHI*B(1:3,4:6)
Kalman Filter Input Dialog Box for Baro model

- **Parameters**
  - **Number of filters:** 1
  - **Enable filters:** Always
  - **Initial condition for estimated state:** [0;0;0]
  - **Initial condition for estimated error covariance:** 10^3*eye(3)
  - **State transition matrix:** PHI
  - **Process noise covariance:** diag([0 GWGT(2,2) GWGT(3,3)])
  - **Measurement matrix source:** Specify via dialog
  - **Measurement matrix:** [1 0 1]
  - **Measurement noise covariance:** 100^6*eye(1)

- **Outputs**
  - Output estimated measurement <Z_est>
  - Output estimated state <X_est>
  - Output MSE of estimated state <MSE_est>
  - Output predicted measurement <Z_prd>
  - Output predicted state <X_prd>
  - Output MSE of predicted state <MSE_prd>
Simulink M-Code for DVS model

dt=.1;
g=9.8;
re=6.37e6;
omegax=.001;
omegay=.001;
wacc=3e-6;
wgyro=5.23598776e-7;
F=zeros(9,9);
F(1,2)=1;
F(2,3)=-g;
F(3,2)=1/re;
F(3,9)=omegax;
F(4,5)=1;
F(5,6)=-g;
F(6,5)=1/re;
F(6,9)=omegay;
F(7,8)=1;
GWGT=zeros(9,9);
GWGT(2,2)=wacc;
GWGT(3,3)=wgyro;
GWGT(5,5)=wacc;
GWGT(6,6)=wgyro;
GWGT(8,8)=wacc;
GWGT(9,9)=wgyro;
A=[-dt*F dt*GWGT;zeros(9,9) dt*F'];
B=expm(A);
PHIT=B(10:18,10:18);
PHI=PHIT';
Q=PHI*B(1:9,10:18);
Kalman Filter Input Dialog Box for DVS model

Parameters

Number of filters: 1
Enable filters: Always

Initial condition for estimated state: [0;0;0;0;0;0;0;0;0]
Initial condition for estimated error covariance: 0*eye(9)
State transition matrix: PHI
Process noise covariance: diag([0 wacc wgyro 0 wacc wgyro 0 wacc wgyro])
Measurement matrix source: Input port <H>
Measurement noise covariance: diag([.001 .001])

Outputs

Output estimated measurement <Z_est>  Output predicted measurement <Z_prd>
Output estimated state <X_est>  Output predicted state <X_prd>
Output MSE of estimated state <MSE_est>  Output MSE of predicted state <MSE_prd>