Attitude Determination and Bias Estimation Using Kalman Filtering

Roni Yadlin[1]
United States Air Force Academy, Colorado Springs, Colorado, 80841

The US Air Force Academy has an advanced and largely self-sustaining small satellite program. Cadets are responsible for the design, testing, and construction of a series of small satellites. In anticipation of this program, the Astronautics Department selects cadets to participate in research internships. Research in one such internship focused on using low cost attitude sensors and gyroscopes to estimate the attitude of a satellite and determine the gyroscopes' bias. Kalman filtering was used to determine this bias and estimate the attitude in three axes, using measurements in only two axes. A computerized satellite dynamic model was used to test the bias estimation filter. Next, the code was tested using actual data collected on an air-bearing table at the University of Surrey's Surrey Space Centre. Finally, the code was connected directly to the air-bearing table to implement hardware-in-loop testing. This paper details the theory behind the estimator in addition to the implementation of the estimator into the hardware as well as the results of the hardware-in-loop testing.

Nomenclature

ϕ	=	Euler angle representing roll
θ	=	Euler angle representing pitch
ψ	=	Euler angle representing yaw
$\dot{\phi}$	=	Euler rate representing roll rate
$\dot{\theta}$	=	Euler rate representing pitch rate
$\dot{\psi}$	=	Euler rate representing yaw rate
$\vec{b}$	=	Bias of gyroscopes
$\vec{\omega}$	=	Angular ~~momentum~~ velocity
$\bar{X}$	=	State vector representing the outputs of the estimator
$\bar{F}$	=	F matrix relating the states to their derivatives
Φ	=	State Transition Matrix (STM)
$\bar{Q}$	=	System process noise matrix
$\bar{R}$	=	System measurement noise matrix
$\bar{P}$	=	Covariance matrix
$\hat{P}$	=	Updated covariance matrix
$\bar{Z}$	=	Measurement from attitude sensor
$\bar{H}$	=	Observation matrix relating measurements to desired states
$\bar{K}$	=	Kalman Gain
$\hat{X}$	=	Updated state vector
T_s	=	Sampling time for attitude sensor
τ_s	=	Sampling time for gyroscope

I. Introduction

While working at Surrey Space Centre at the University of Surrey, in conjunction with Surrey Satellite Technology Ltd. (SSTL), the research completed focused on the possibility of using a low cost, yet highly accurate, attitude sensors in combination with low cost gyroscopes of unknown or questionable accuracy. This research project entailed building an estimator that would read in the sensor data and in turn output the Euler angles and the bias of the gyros. Finally, the estimator was tested on an air-bearing table. Several methods were used to build different estimators that would most effectively provide the desired outputs.

---

[1] Software and Analysis Team Lead, FalconSAT-5, U.S. Air Force Academy Small Satellite Research Laboratory, P.O. Box 5228, USAF Academy, CO 80841, AIAA Student Member

II. Kalman Filtering

Developed by R.E. Kalman in 1960, the Kalman filter is an algorithm that processes noisy data and sensor measurements to accurately predict a satellite's attitude and motion.[i] The Kalman filter is a set of mathematical equations that provides a recursive means of estimating the state of a process while minimizing any error in the system.[ii] Because all gyroscopes have some inherent bias, they fail to provide accurate measurements. The Kalman filter allows for an accurate estimation of the states that eliminates the bias and noise of the gyroscopes.

The purpose of the Kalman filter is to force the measured and estimated values for the filter's states to converge and for the difference between the estimate and actual states, or covariance, to be minimized. Figure 1 shows a demonstration of the mechanics of a Kalman filter.

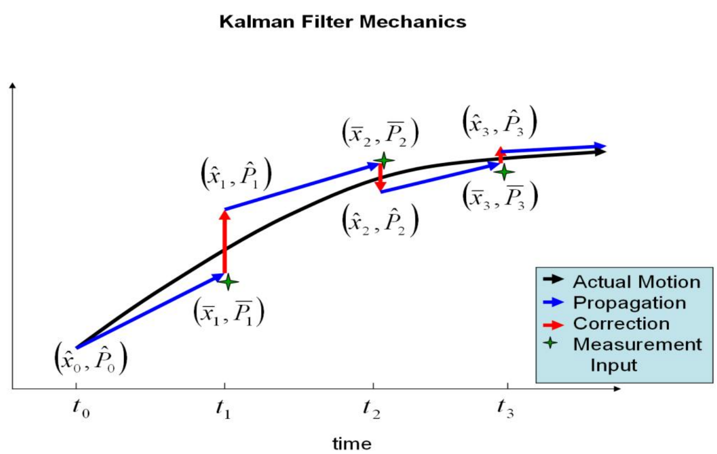

Figure 1: Kalman Filter process for converging on a true state vector

III. Attitude Estimator

The original plan for this research was to design an estimator that was able to input a measurement of two Euler angles (roll and pitch) from the attitude sensor and a measurement of angular velocity (with the inherent bias) from the gyroscopes and then output the desired states of all three Euler angles (roll, pitch, and yaw) as well as the bias of the gyros. In order to develop this estimator, a simplifying assumption was made. It was originally assumed that the attitude sensor measured, and therefore inputted to the estimator, all three Euler angles.

A series of computations were necessary prior to implementation of the Kalman filter. The following equation was used to find the Euler rates based on the inputted Euler angles and angular velocity (equation 1).

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \dfrac{\sin(\psi)}{\cos(\phi)} & \dfrac{\cos(\psi)}{\cos(\phi)} & 0 \\ \sin(\psi)\tan(\phi) & \cos(\psi)\tan(\phi) & 1 \end{bmatrix} \begin{bmatrix} \omega_1 - b_1 \\ \omega_2 - b_1 \\ \omega_2 - b_1 \end{bmatrix} \quad (1)$$

A simple, linear integration scheme was used to propagate the states forward in time (equation 2)

$$\phi = \phi + \dot{\phi}dt \qquad \theta = \theta + \dot{\theta}dt \qquad \psi = \psi + \dot{\psi}dt \quad (2)$$

The sampling rate of the gyroscope was much faster than that of the attitude sensor, and therefore this update of the angles occurred many times in between the various attitude sensor readings. After each attitude sensor measurement, the reading from the sensor was compared to the propagated angle to determine an error, which was then later used in the Kalman Filter.

$$\Delta\phi = \phi_m - \phi_c \qquad \Delta\theta = \theta_m - \theta_c \qquad \Delta\psi = \psi_m - \psi_c \quad (3)$$

The State Space representation of the system is shown in equation 4.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{b}_1 \\ \dot{b}_2 \\ \dot{b}_3 \end{bmatrix} = F \begin{bmatrix} \phi \\ \theta \\ \psi \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = F\bar{X} \qquad \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = H\bar{X} \quad (4)$$

The F matrix, which relates the states to their derivatives, was found by taking a series of partial derivatives as shown in equation 5.

$$F = \frac{\delta(\dot{\phi}, \dot{\theta}, \dot{\psi}, \dot{b_1}\dot{b_2}, \dot{b_3})}{\delta(\phi, \theta, \psi, b_1, b_2, b_3)} \quad (5)$$

For the simulation, it was assumed that the over the period of 4 seconds that the angles were propagated forward, bias was constant and therefore $\dot{\vec{b}} = 0.$ The F matrix, in combination with an identity matrix, was used to determine the state transition matrix (equation 6).

$$\Phi = I + FT_s \quad (6)$$

T_s is the sampling time of the attitude sensor. This is an approximation, using a first order Taylor Series.

The observation matrix, H, relates the states to the measurements taken by the attitude sensor (equation 7).

$$H = \frac{\delta(\phi, \theta, \phi)}{\delta(\phi, \theta, \phi, b_1, b_2, b_3)} \quad (7)$$

To begin testing, an initial covariance matrix (P) was defined. This matrix was based on the standard deviations of the model and measurement errors. To simplify, it was assumed there was no coupling and therefore the covariance matrix was diagonalized. Q represents the system process noise matrix and was used to propagate the covariance matrix (equation 8). R, the system measurement noise matrix, and K, the Kalman Gain matrix (equation 9), were used to propagate the states (equation 11). Similar to the covariance matrix, the noise matrices can also be modified to affect the filter mechanics. Modifying these matrices is known as tuning the Kalman filter.

$$\bar{P} = \Phi\bar{P}\Phi^T + Q \qquad (8)$$

$$K = PH^T[HPH^T + R]^{-1} \quad (9)$$

$$\hat{P} = [I - KH]\bar{P} \qquad (10)$$

$$\hat{X} = \bar{X} + K(\Delta\bar{X}) \qquad (11)$$

Once the states and the covariance matrix were updated, the process began again until the covariance was minimized and the estimated and measured values converged.

IV. Validation of 3-axes Filter

In order to validate the estimator, a set of data was generated. This data included Euler angles and angular velocity readings, with bias and error added in the form of white Gaussian noise. Figures 2, 3, and 4 show the results of the estimator relative to the truth data.

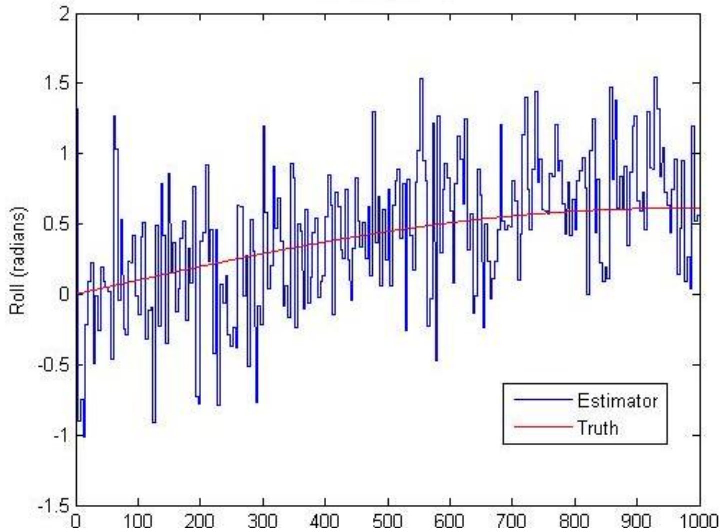

Figure 2: Roll versus Time for the estimator and truth

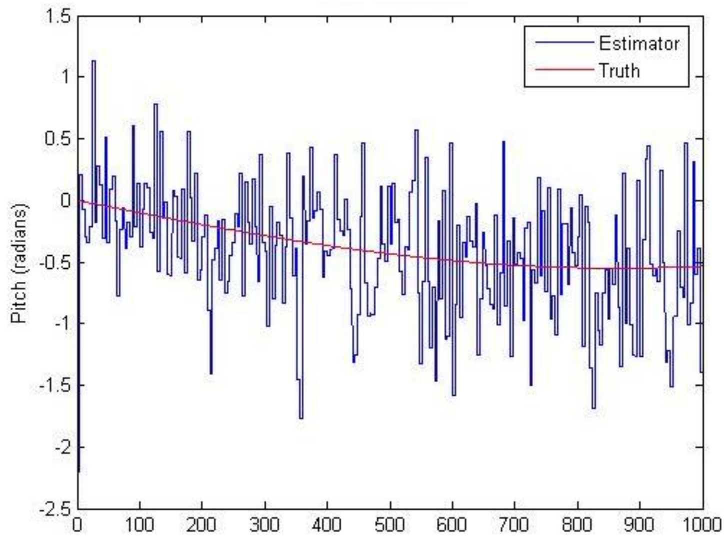

Figure 3: Pitch versus Time for the estimator and truth

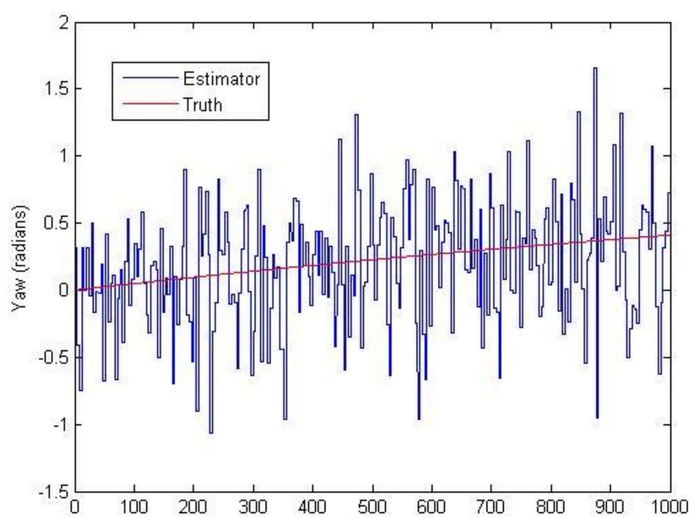

Figure 4: Yaw versus Time for the estimator and truth

The bias added to the system had a magnitude of .229 radians per second. The inputted data was made up of angles ranging from 1 to .001, which made the added bias fairly large relative to the actual values of the data. The noise added was also very large compared to the data itself. Because of this, the data that was inputted into the filter was too noisy for it to converge in the time provided. The estimator is, however, working appropriately because the estimated values for the states do follow the general trend of the truth data.

V. Simulation

The next step was to modify the Matlab code to be used in a Simulation built with Simulink. The model was first built with the three axes attitude measurements and then modified to reflect a sensor that only reads in measurements in two axes. Figure 5 shows the result for the roll Euler angle, whereas Figure 6 shows the estimated bias.

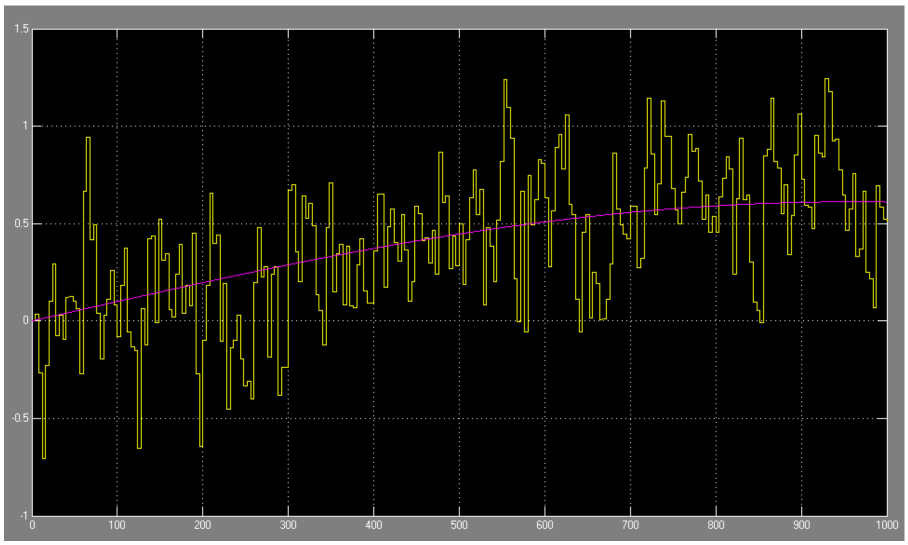

Figure 5: Roll versus time for simulation and truth

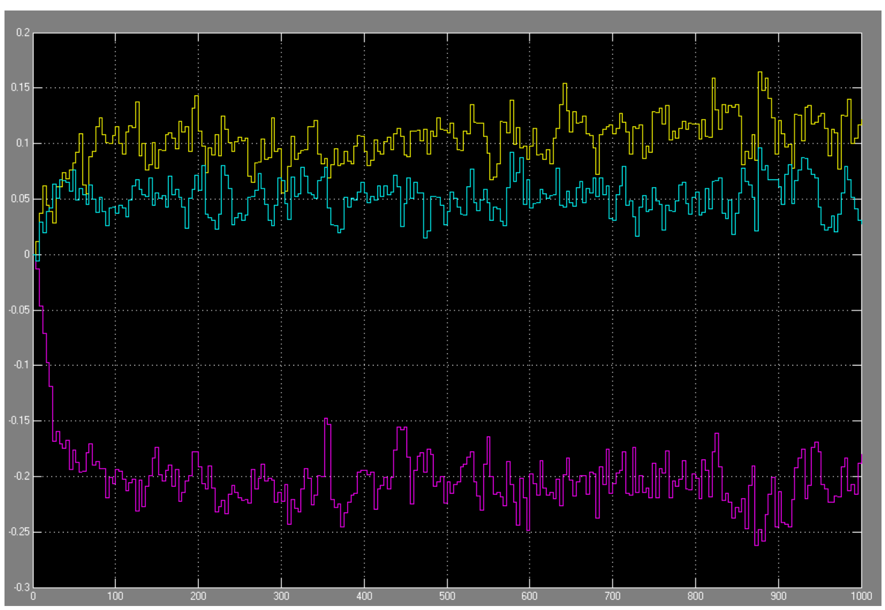

Figure 6: Bias versus time

VI. Two-axes Filter

In order to modify the code and simulation to reflect attitude measurements in only 2 axes, several changes were made to the Filter. The first step was determining if the system was observable. A system is observable if and only if

$$\text{rank}\begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} = n \quad (12)$$

For this system, the rank of the observation matrix was only 4, as opposed to the necessary 6 that would imply observability.

This system, however, was also not observable, and at this point, the focus of the project shifted towards simple bias estimation, independent of attitude estimation. A filter was designed that would estimate only the biases. The State Space representation of this system is shown in equation 13.

$$\begin{bmatrix} \dot{b}_1 \\ \dot{b}_2 \\ \dot{b}_3 \end{bmatrix} = F \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad \begin{bmatrix} \phi \\ \theta \end{bmatrix} = H \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (13)$$

With this form, the estimator was able to adequately estimate bias in all three axes.

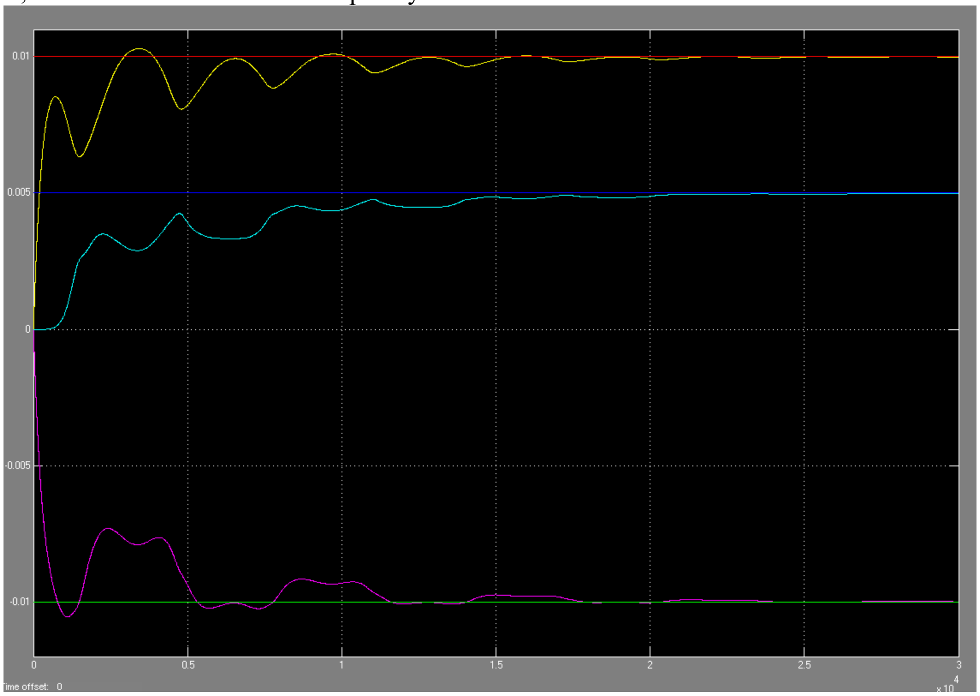

Figure 7: Bias versus time

VI. Hardware-in-loop Testing

The final step in this process was to implement the estimator to a hardware system. The Controls Group at Surrey Space Center recently installed an Air-Bearing Table that was used as the hardware component. The first step in the hardware implementation was to collect data from the table and use that data in the filter to determine if the estimator was capable of converging on the bias of the gyroscopes in a reasonable time.

Finally, the filter was attached directly to the air-bearing table in order to estimate the biases. The following figures show the results of the estimator both stationary and while moving. After 3 minutes, the table was allowed to move in the X and Y directions and the estimator was still able to adequately converge on the bias of the gyroscopes.

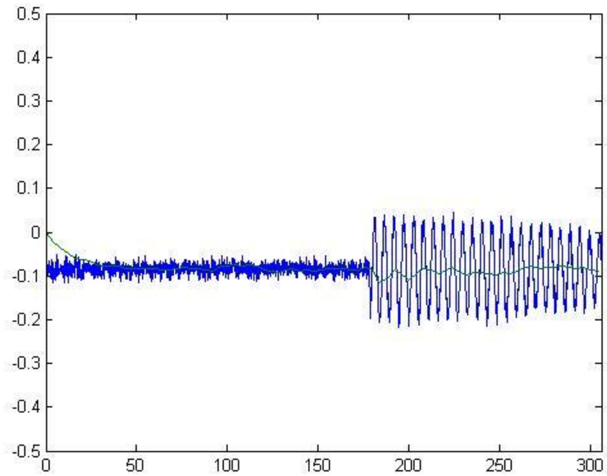

Figure 5: Bias in the first axis

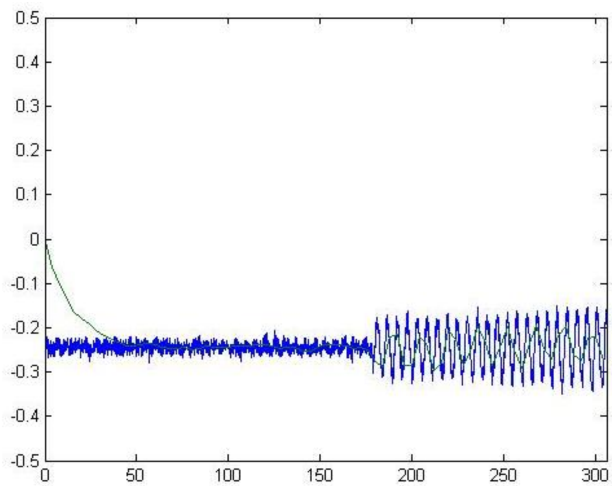

Figure 6: Bias in the second axis

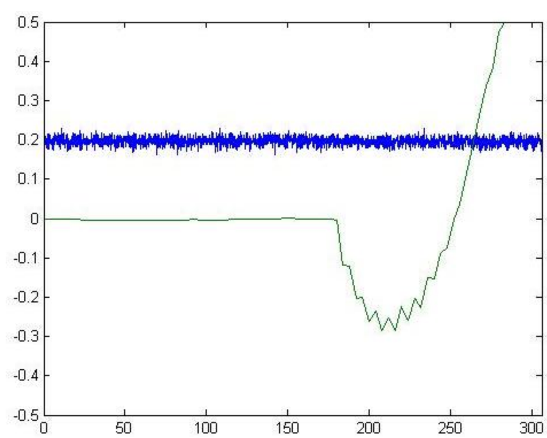

Figure 7: Bias in the third axis

As is obvious from Figure 21, the code was unable to estimate bias in the third axis. Because of the makeup of the H matrix, when the Euler angles were both zero, the estimator was unable to estimate bias in the third axis.

In order to estimate the bias in the third axis, another estimator was built in which the rotation scheme used to determine the angular rates in the Kalman filter was changed. The following rotation scheme and H matrix were used for the second estimator.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) & 1 \\ \sin(\phi) & \cos(\phi) & 0 \\ \dfrac{\sin(\phi)}{\cos(\theta)} & \dfrac{\cos(\phi)}{\cos(\theta)} & 0 \end{bmatrix} \begin{bmatrix} (\omega_1 - b_1) \\ (\omega_2 - b_2) \\ (\omega_3 - b_3) \end{bmatrix} \quad (14)$$

This estimator was run in parallel with the first estimator to estimate the bias in all three axes. The following figures show the results of the estimators in all three axes.

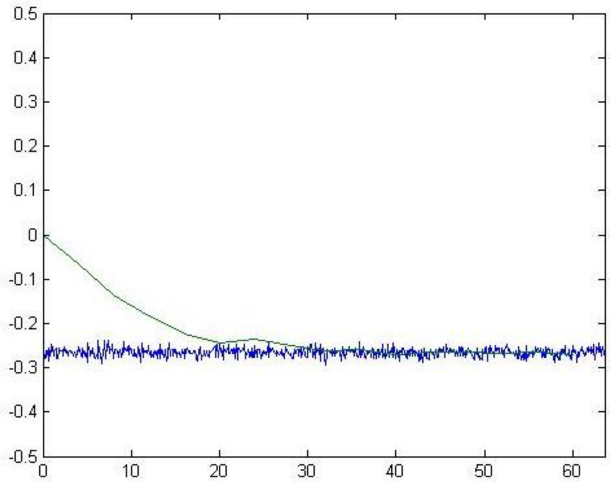

Figure 9: Bias estimation in first axis

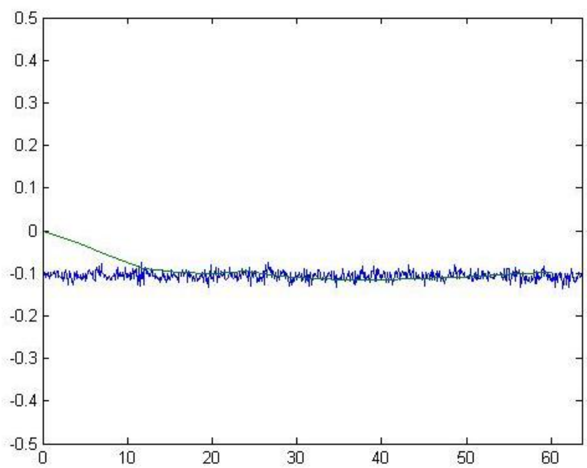

Figure 10: Bias estimation in the second axis

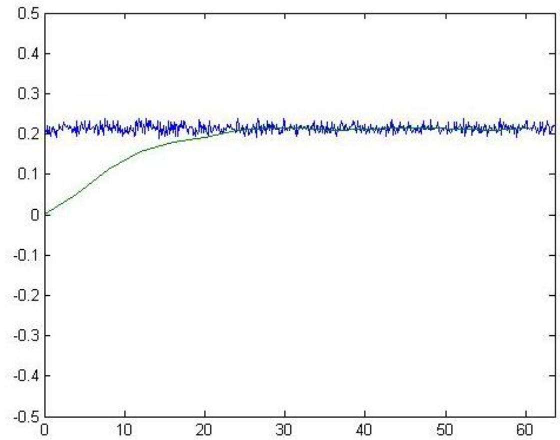

Figure 11: Bias estimation in the third axis

VII. Attitude and Bias

The final step in this project was to design an estimator that combined the bias estimation with the attitude estimation. This estimator ran simply as a bias estimator until it converged upon the true bias of the gyroscopes. Then, the estimator used bias free gyroscope data to determine the attitude.

The attitude estimator utilized the following rotation scheme to determine Euler rates from the angular velocity, which in this case was bias free.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \dfrac{\sin(\phi)}{\cos(\theta)} & \dfrac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (15)$$

The following figures show the attitude estimation for this final filter. The yellow line represents the estimate while the purple line is the truth data.

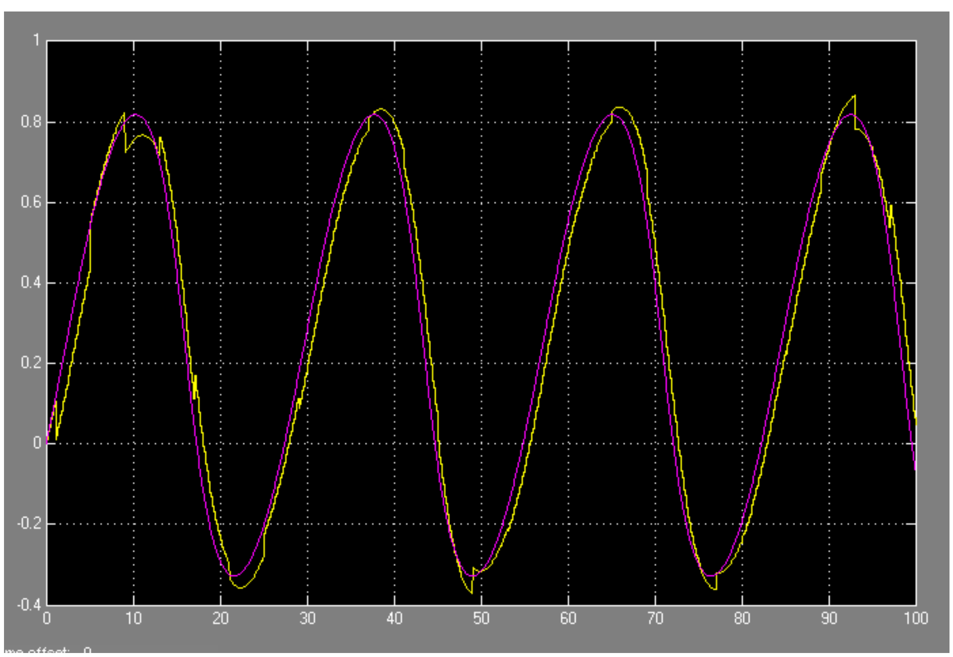

Figure 12: Roll Euler Angle relative to time

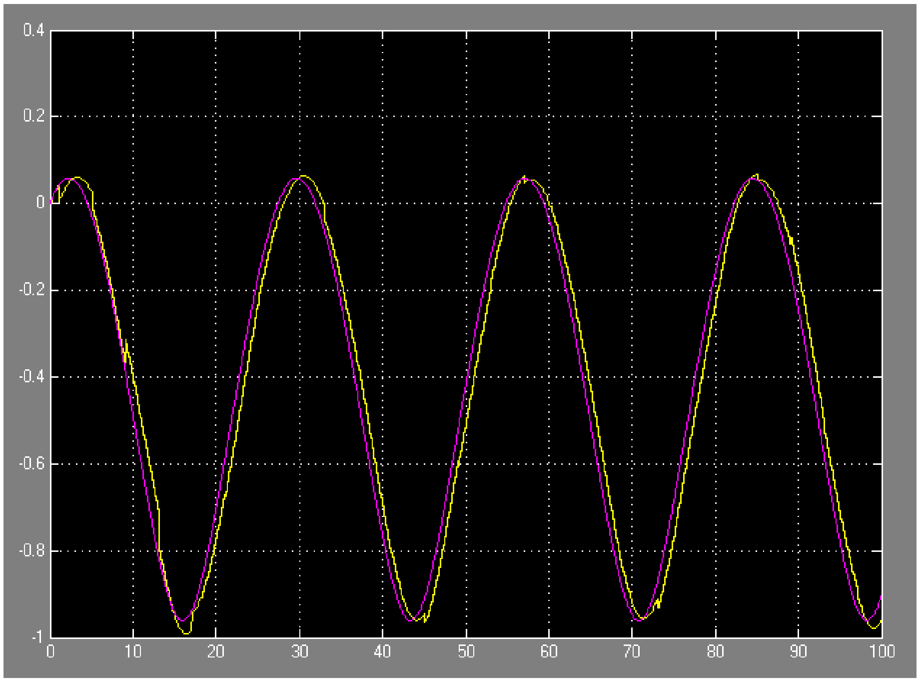

Figure 13: Pitch Euler Angle relative to time

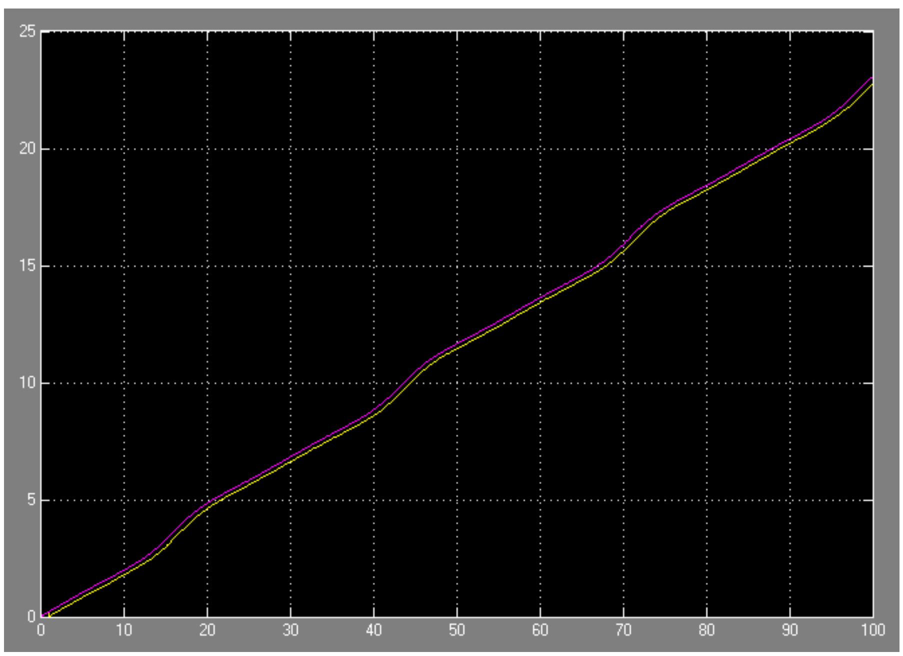

Figure 14: Yaw Euler Angle relative to time

The angle measurements were noisy, which explains the minor deviation from the true Euler angles. However, the error was minimal and deemed acceptable. These figures demonstrate the ability of the filter to estimate bias and attitude in three axes using three axis angular rate data and two axis angular position data.

VIII. Future Applications

This research can be adapted for use on a future generation of FalconSAT small satellites. The hardware used in this simulation is cheap and therefore is ideally suited to the budget constraints felt by the United States Air Force Academy's Astronautics department. While a lower cost brings inaccuracies in the measurements, filters such as the one designed in this project can easily be implemented to filter out the noise and error from the measurements and provide an accurate determination of a satellite's attitude.

IX. Acknowledgments

I would like to thank department of Astronautics at the United States Air Force Academy for preparing me for the summer research internship and for allowing me the opportunity to work at the Surrey Space Centre. In particular, for their help in the area of controls, I would like to thank Maj. Nick Hague, Dr. Scott Dahlke, and Prof. Bill Saylor. At the University of Surrey, I would like to thank my two supervisors, Dr Yoshi Hashida and Dr Alex Pechev. Without their help and support I would have never completed this project. I would like to thank Dr David Barnhart for all his help and support at the University. Also at Surrey, I would like to thank David Wokes and Chris Bridges for being so welcoming to and supportive of me during my time in England.

---

[i] Brown, Robert G. Introduction to Random Signal Analysis and Kalman Filtering. New York: John Wiley & Sons, 1983.
[ii] Welch, Greg and Gary Bishop. An Introduction to the Kalman Filter. Chapel Hill: University of North Carolina at Chapel Hill, 2006.