Extended Kalman Filter Report

Saurabh K Pitre
N13097069

Part I) Study:

The extended Kalman Filter is a Nonlinear version of the Kalman Filter. It works principally by linearizing of the states about a given operating point and then calculating all the parameters about that point. This is done using Taylor series expansion of the nonlinear function, ignoring the higher order terms of the series and then evaluating it at the operating point.

$$A = \left.\frac{\partial f}{\partial x}\right|_{\overline{x},\overline{u},0} \quad ; \quad B = \left.\frac{\partial f}{\partial u}\right|_{\overline{x},\overline{u},0} \quad ; \quad F = \left.\frac{\partial f}{\partial v}\right|_{\overline{x},\overline{u},0}$$

$$C = \left.\frac{\partial h}{\partial x}\right|_{\overline{x},0} \quad ; \quad H = \left.\frac{\partial h}{\partial w}\right|_{\overline{x},0}$$

Where $\overline{x}, \overline{u}$ are the values of x and u at time t. After finding the values of A, B, C, F, H at time t we use the Filtering Ricatti Equation with a finite time horizon to calculate the value of the optimal gain G that will be used to minimize the covariance and track the mean of the states.

$$P' = \widehat{A}P + P\widehat{A}^T - PC^TW^{-1}CP + F\widehat{V}F^T$$

Using the P calculated above the G is calculated using:

$$G = PC^TW^{-1}$$

The gain thus calculated in substituted into the following equation for finding the state estimates

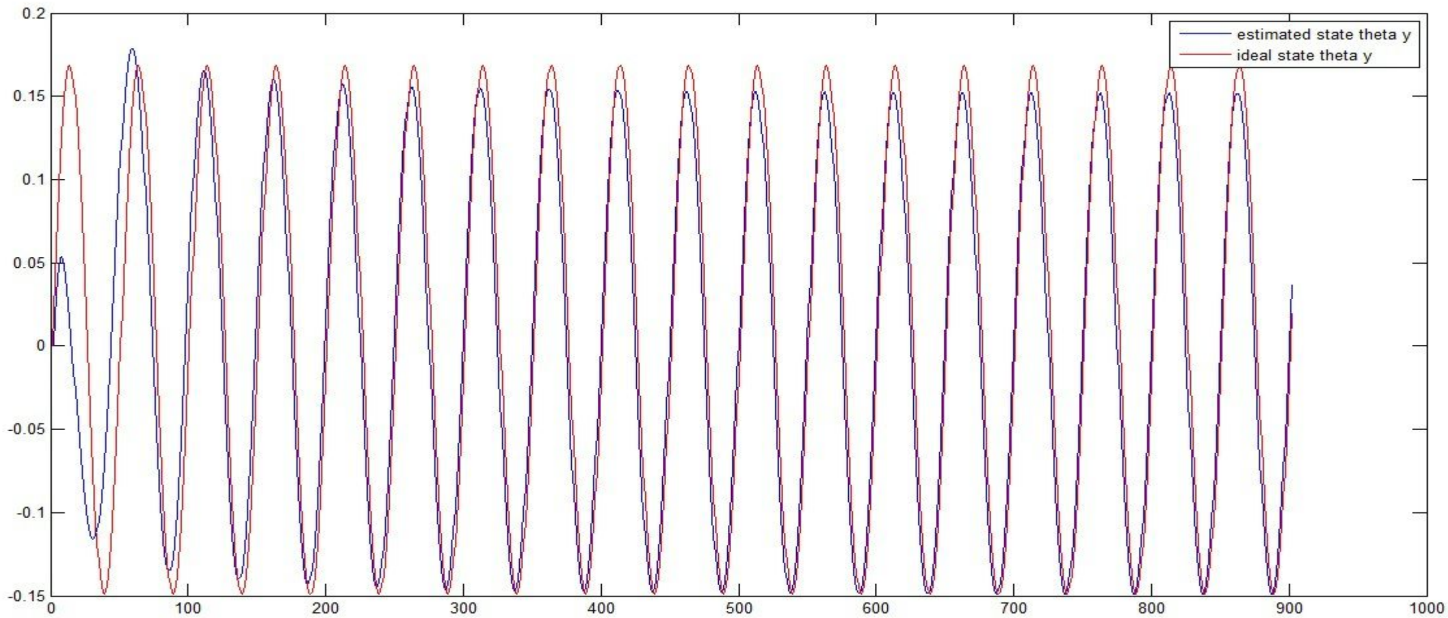
$$\widehat{x}' = f(\widehat{x}, u, 0) + G(y - h(\widehat{x}, 0))$$

This process is reiterated till the estimated state trajectory follows the actual state trajectory

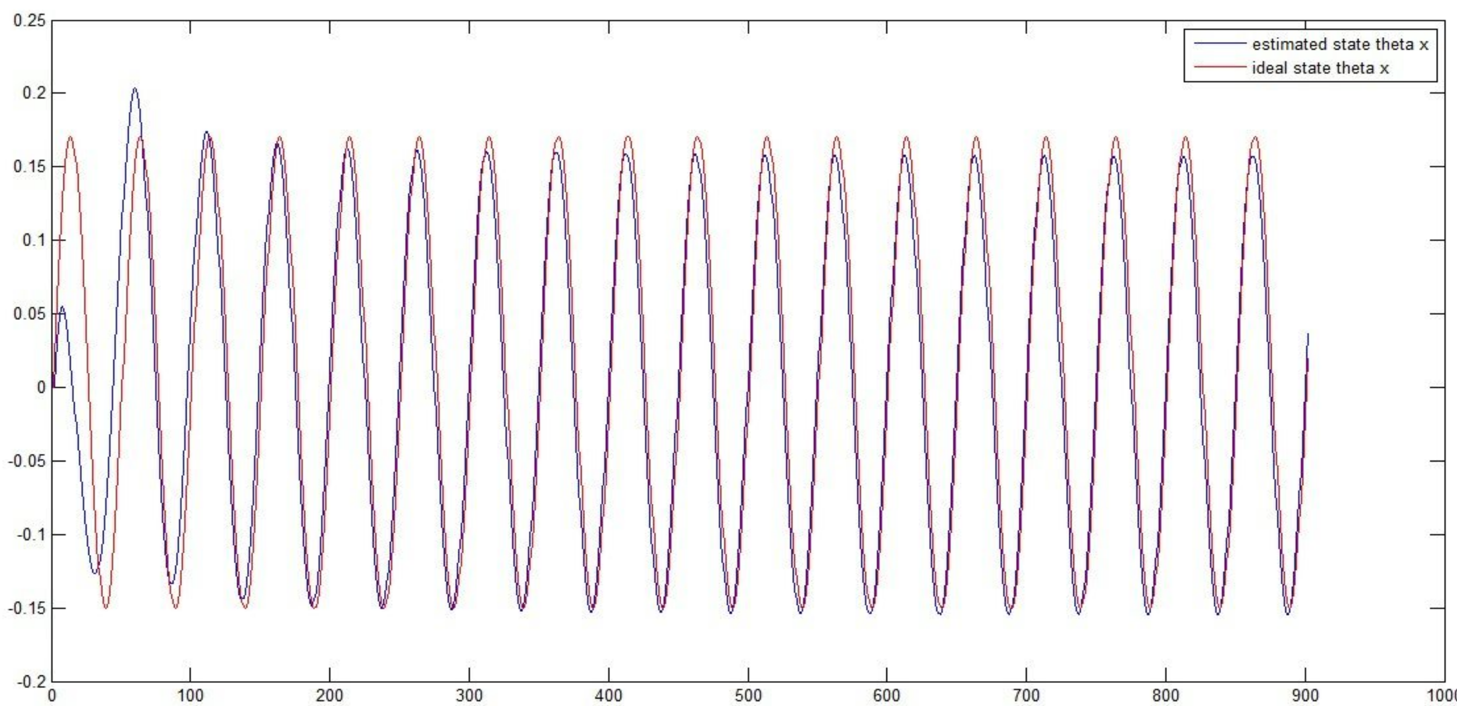
Part II: Simulation

The trajectories for different cases are shown below:

1) Pitching and rolling motion (oscillating motion along both the x and y axis):



Tracking theta_y

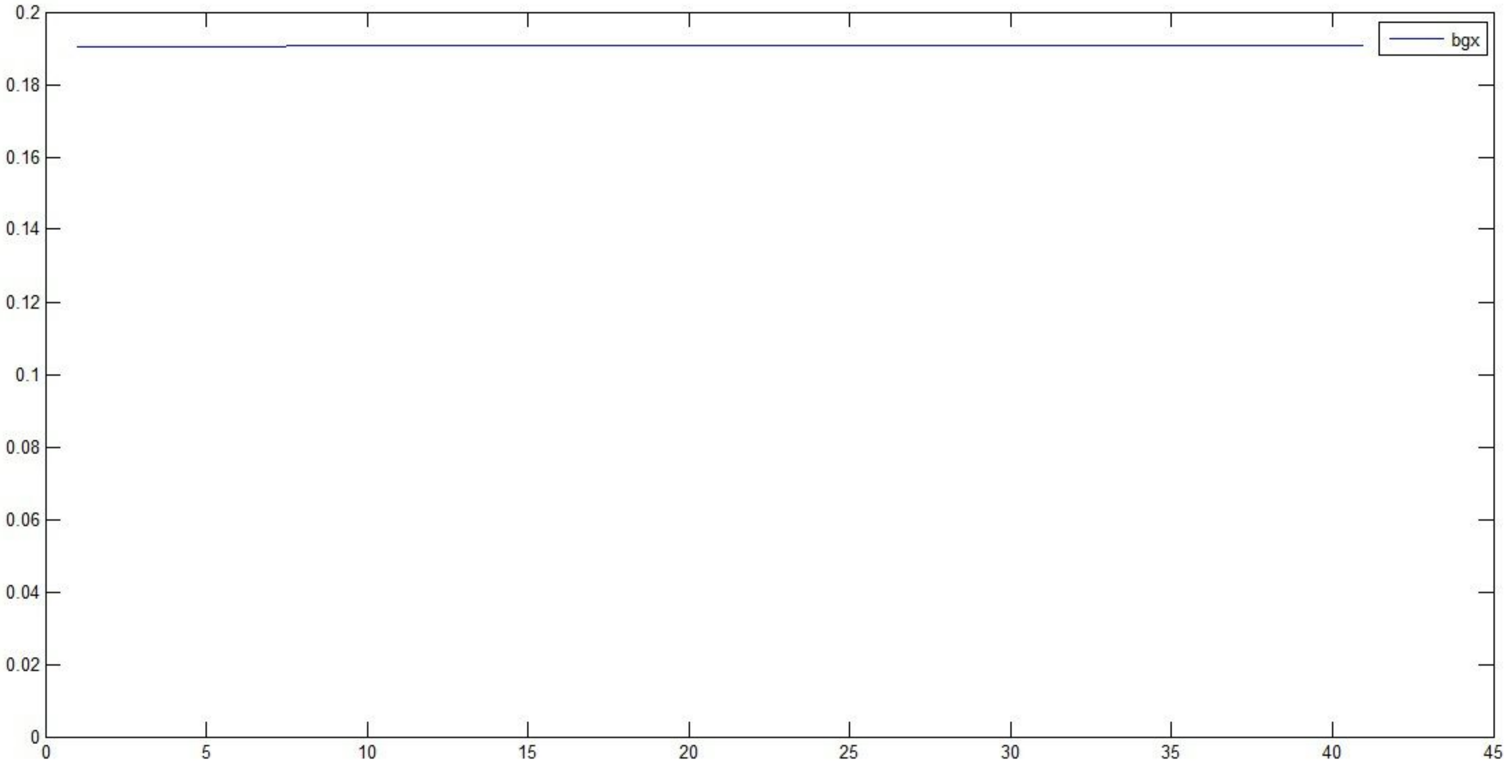


Tracking theta_x

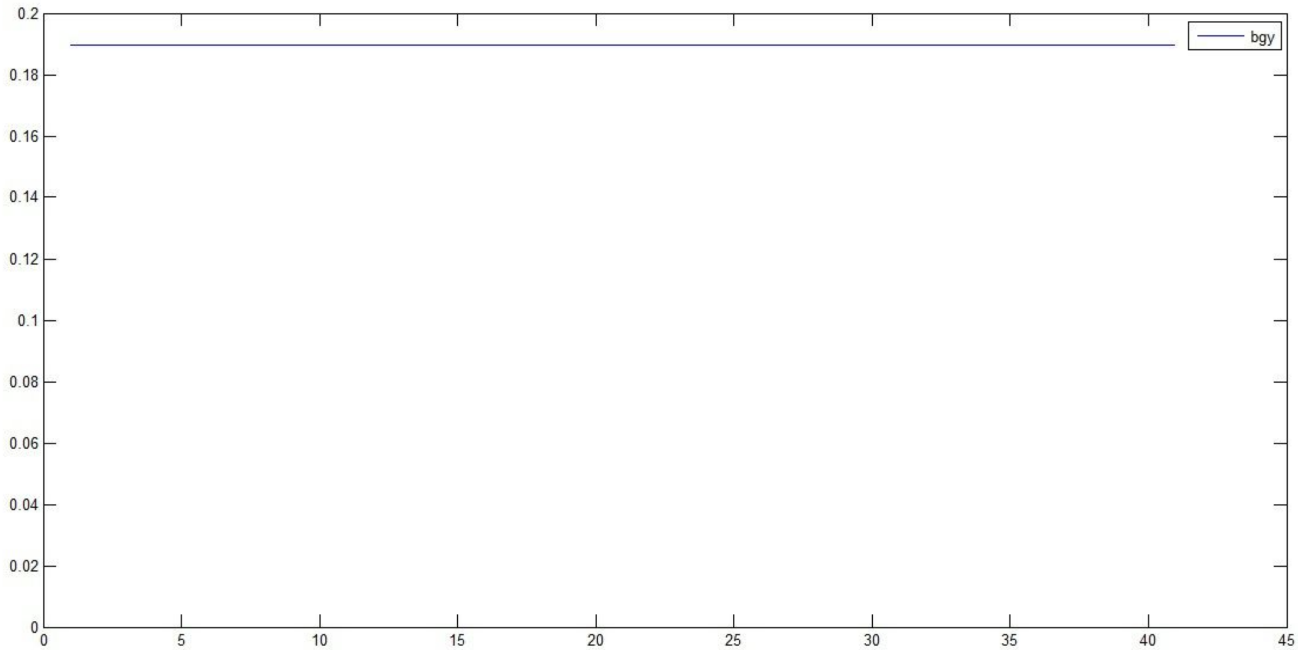
As it can be seen from the figure above the estimated states perfectly track the actual states

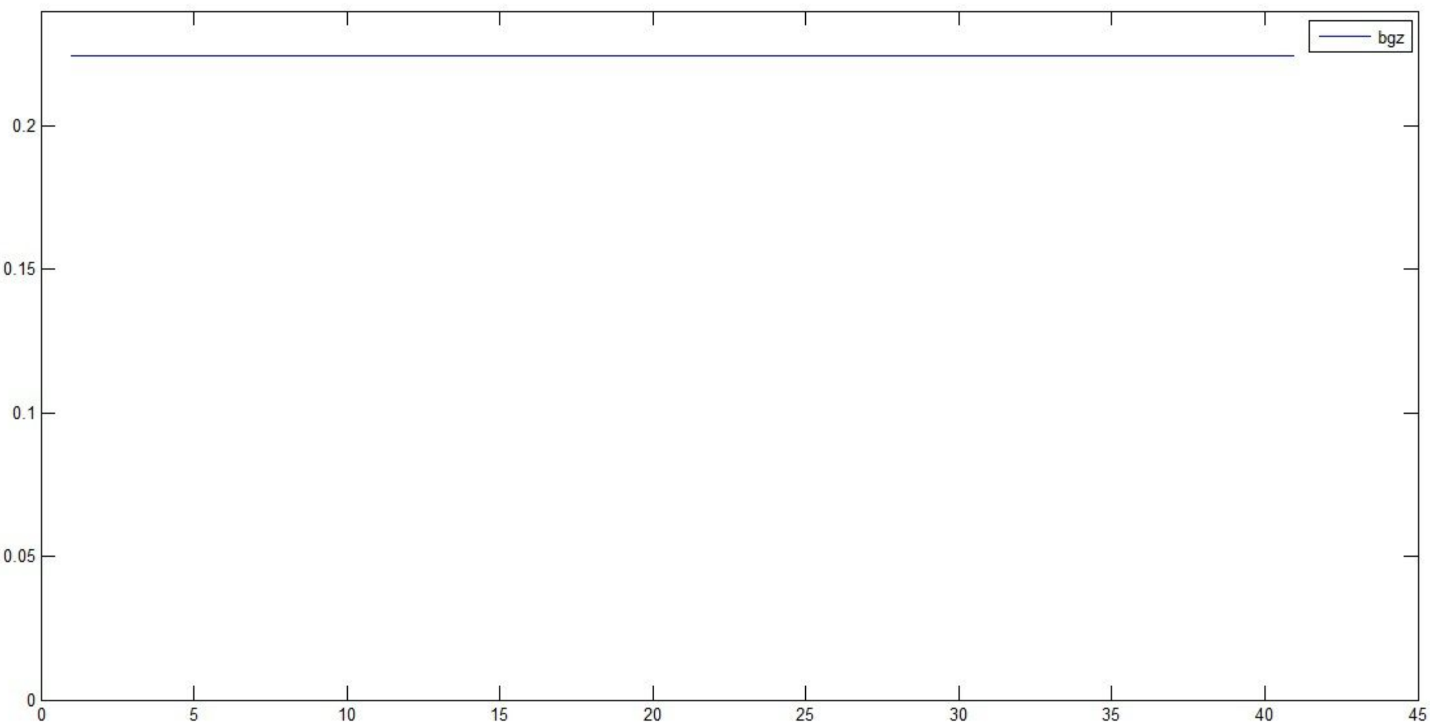
We also track the bias that is shown here:

Bgx, Bgy, Bgz are the bias from the gyro x-axis, y-axis and z-axis measurements. We have assumed a constant value of 0.2 here

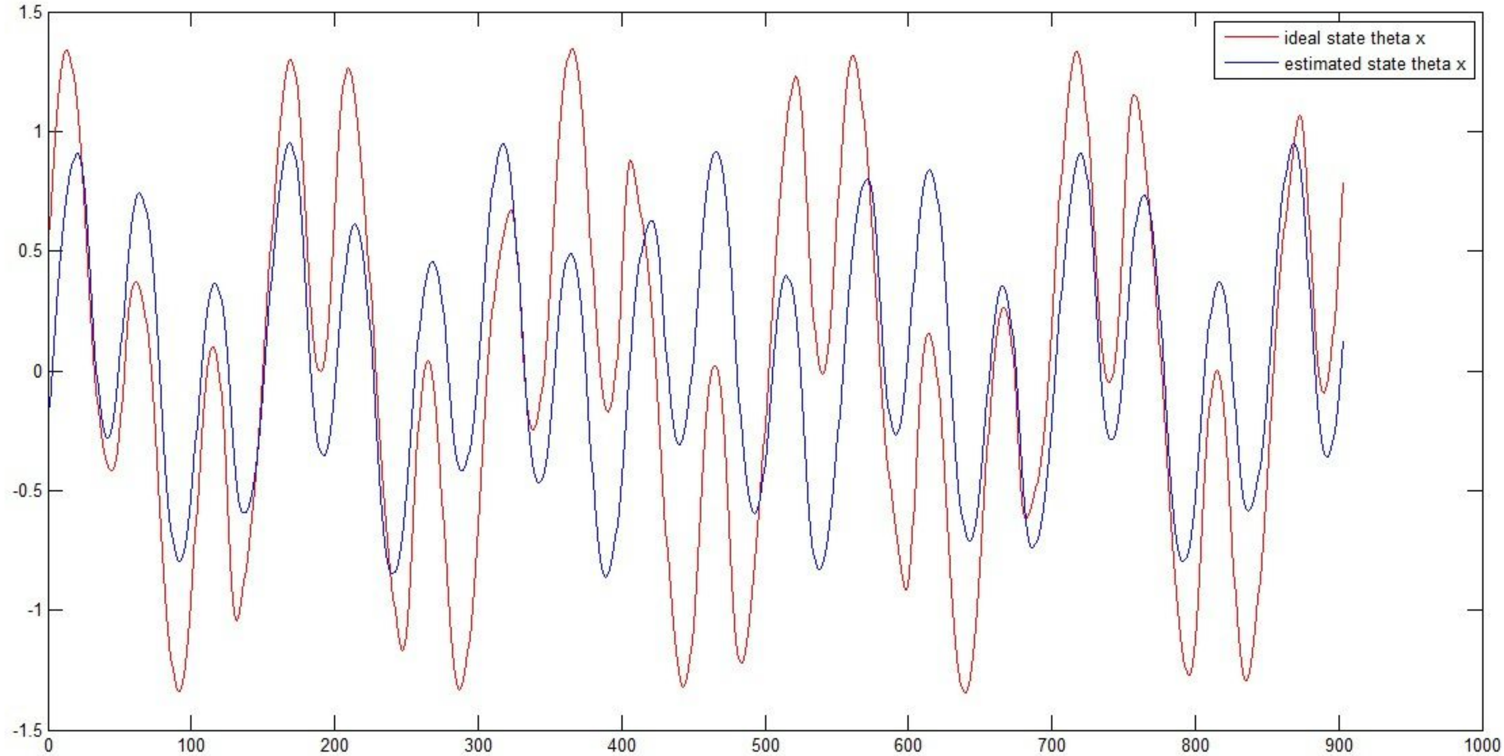


It can be seen from the figure above and below that the controller successfully tracks the bias bg. It must be noted that here the value of bg have been taken for the later half of the simulation as there were some technical errors in plotting the initial part properly.

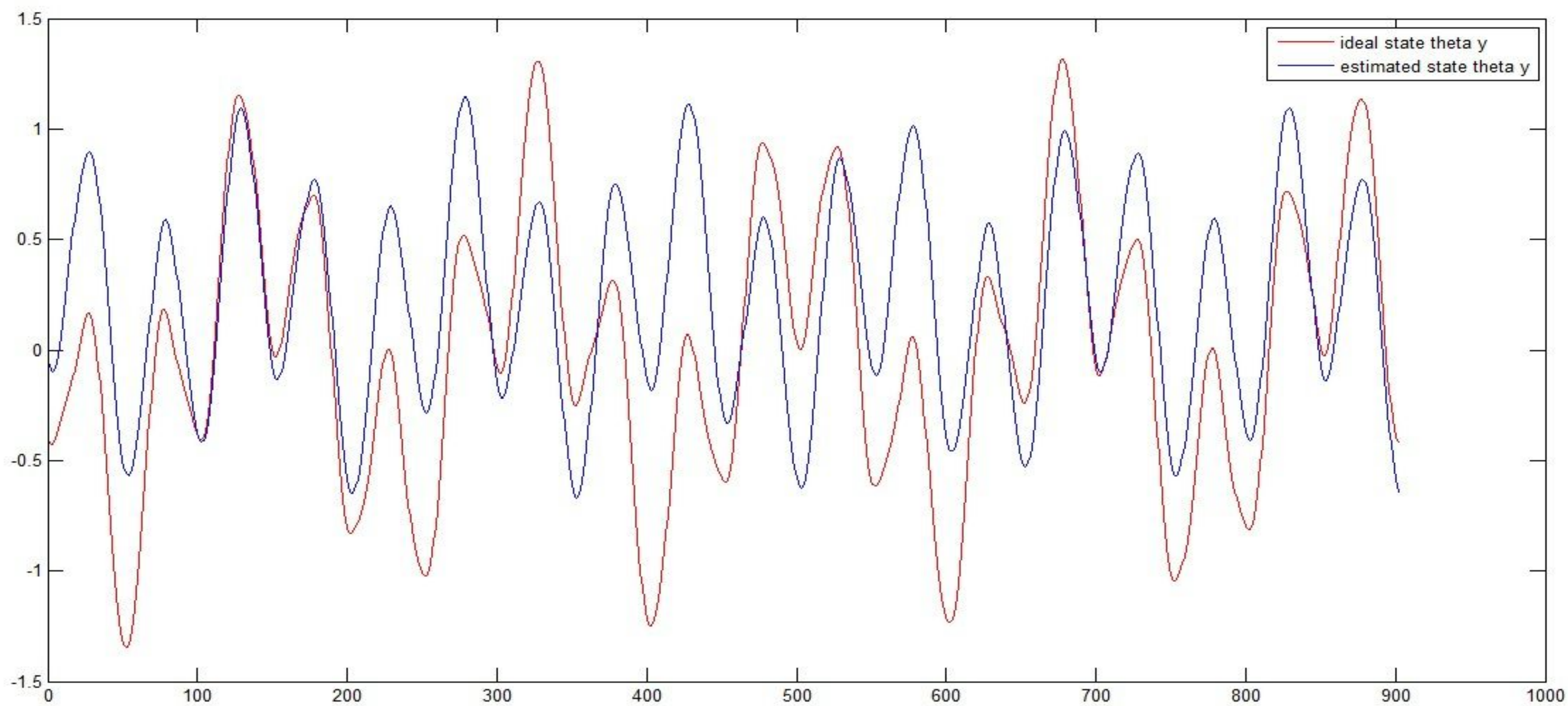




2) Complicated Motion: If our motion has harmonics, it is observed that the extended kalman filter can successfully track it, albeit with some deviation.

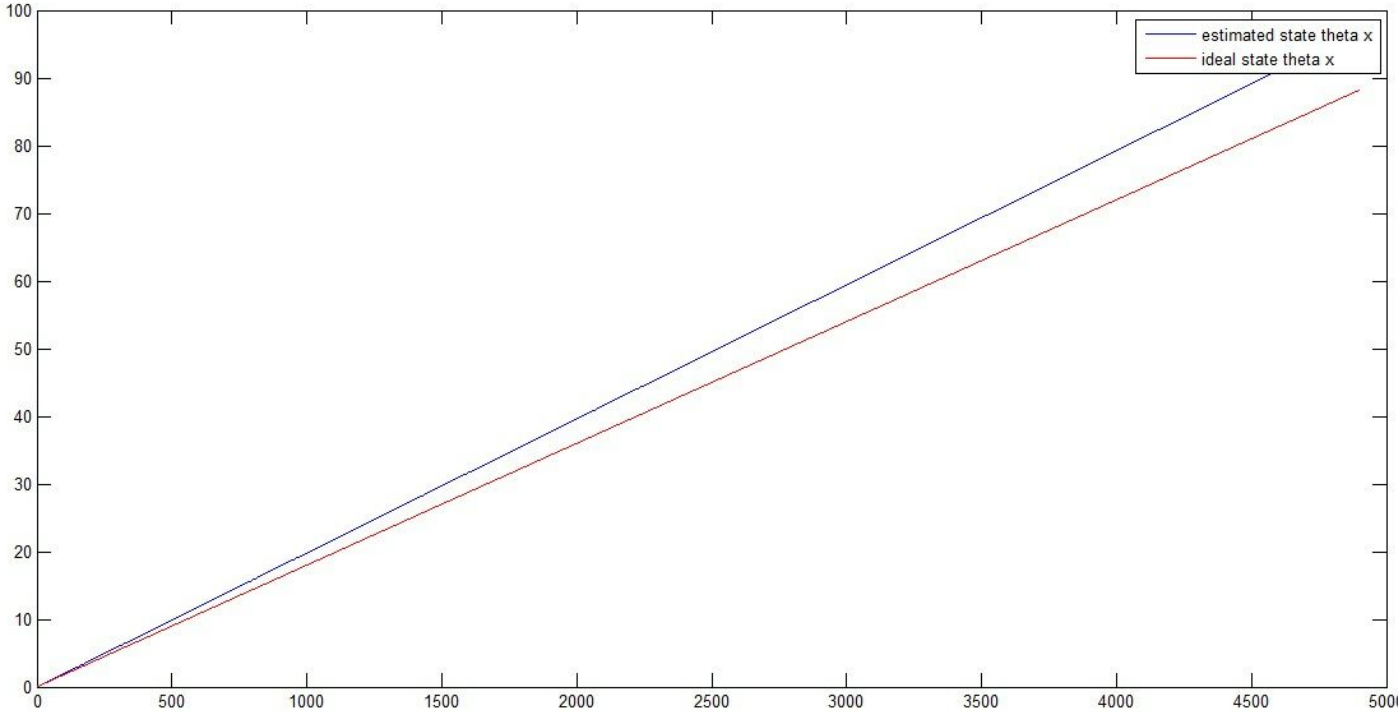


Tracking theta_x



Tracking Theta_y

3) Rolling case: In this case we simulated a case where the rigid object is rotation around its x axis.



Tracking theta_x

It can be seen that the Kalman Filter is not exactly tracking the constant motion and that there is an increasing deviation between the estimated theta and the actual value of theta

Part B: Magnetometer case: Here we also simulate pitching and rolling motion

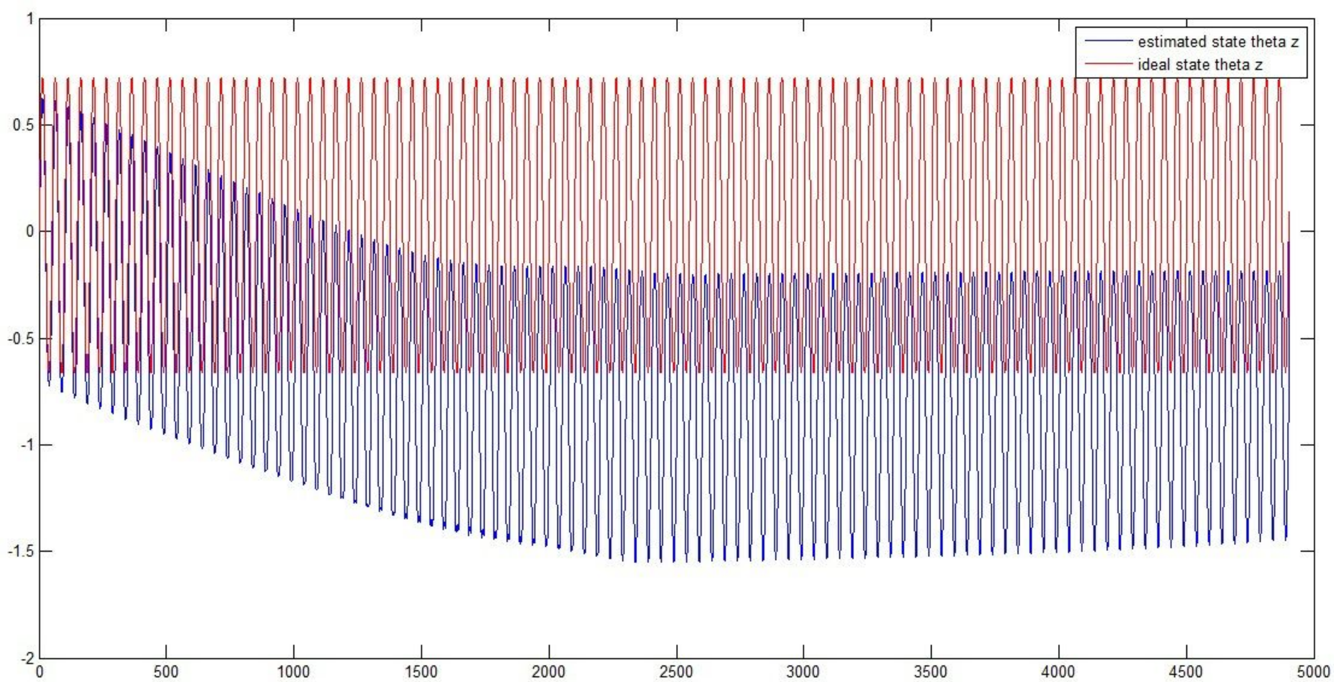
In the magnetometer case we need to change the state equation and the output equation to add theta_z
The resultant matrix function f is:

f(x,u,n) = [ux - nx - bgx - cos(tx)*tan(ty)*(bgz + nz - uz) - sin(tx)*tan(ty)*(bgy + ny - uy)
sin(tx)*(bgz + nz - uz) - cos(tx)*(bgy + ny - uy)
- (cos(tx)*(bgz + nz - uz))/cos(ty) - (sin(tx)*(bgy + ny - uy))/cos(ty)
0
0
0]

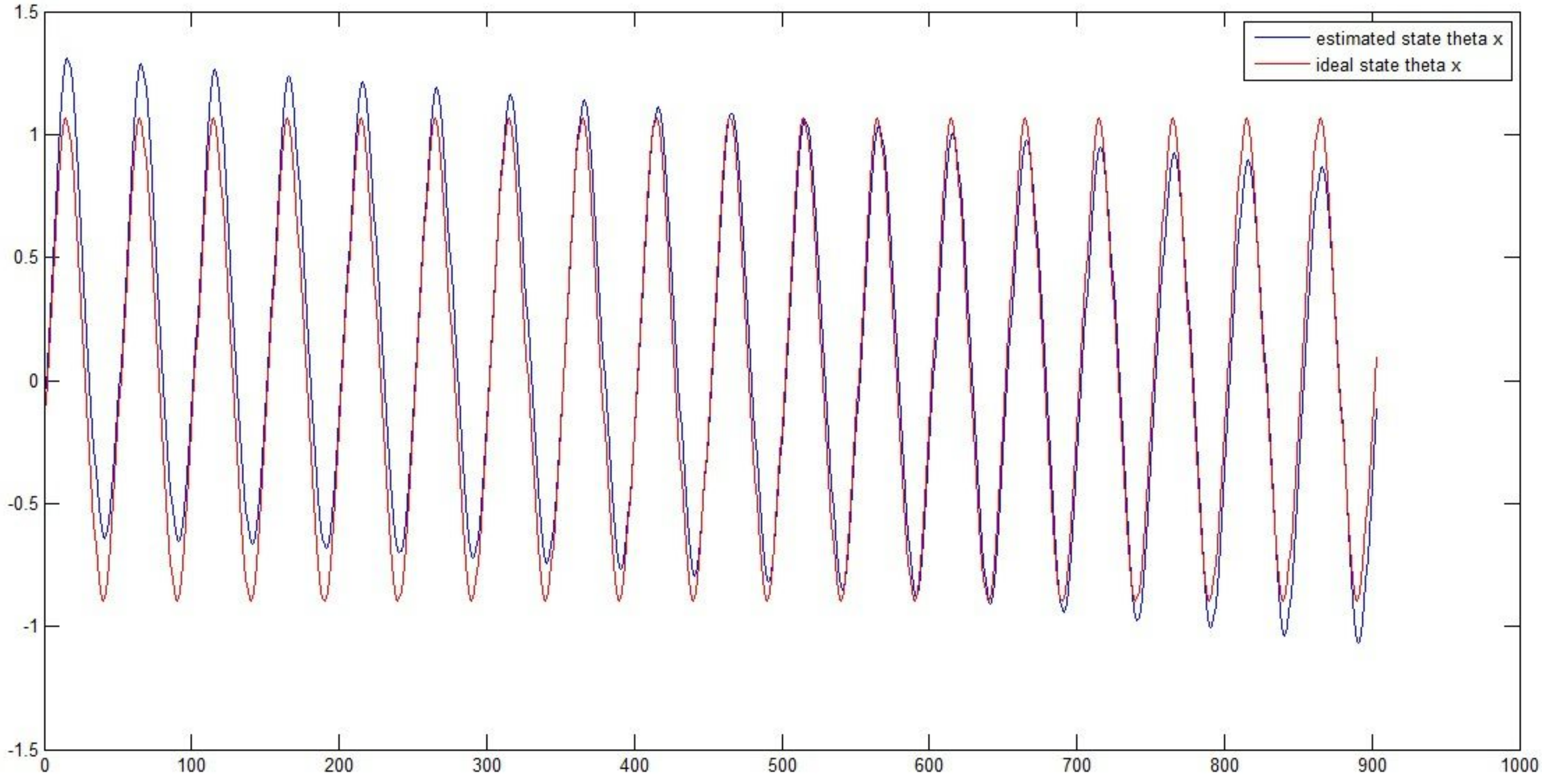
And the output equation g(x)=

[-(49*sin(ty))/5
(49*cos(ty)*sin(tx))/5
(49*cos(tx)*cos(ty))/5
tz]

The resultant states that are tracked are:



Tracking theta_z



Tracking theta_x

It can be seen that there is a significant error in tracking the state theta_z. However the error is relatively constant and the waveform is in phase