# Question #5

## Extended Kalman Filter (EKF)

The EKF starts off at time-step zero (0) with an estimate of the total state and covariance. At each time-step k, during the prediction step the EKF uses the non-linear equations to calculate the estimated state which in turn is used to approximate the predicted covariance matrix.

where

During the correction step the EKF calculates the innovation vector () using the measurement estimate derived from the non-linear measurement equation.

The filter then generates a new linearized DT System (H) matrix, linearized about the new state estimate, to calculate the Kalman gain, update the total state estimate and update the covariance matrix.

Matlab code for implementation and tuning of the EKF can be found in Appendix E.

## Tuning Approach

Tuning the filter involved generating randomized ground truth data from many Monte Carlo simulations and calculating the mean NEES and NIS values at each time step. The EKF’s process noise matrix Q was then tweaked until the NEES and NIS consistency tests were within the desired accuracy bounds (alpha = ±5%). The initial state was also randomized to demonstrate the filte wasn’t biased to a specific nominal trajectory.

## Typical Simulation Plots

The section shows plots from a typical simulation run. Figure how captures an example of the randomly generated UGV and UAV state data.



Figure Simulation Run - Ground Truth States

Figure 2 shows an example of the randomly generated noisy measurement data.



Figure Simulation Run - Simulated Measurement Data



Figure Estimated State Errors



Figure Estimated Measurement Errors

## Chi-Square Test Results

This section shows the final NEES and NIS test results.



Figure Final NEES results



Figure Final NIS Results