Stochastic Modeling

**Instructions:** Answer all Steps within a LATEX document. You may copy paste these questions into your Latex document – this does not need to be a report format. Insert your MATLAB code at the end of your document using an appropriate LATEX package to display the code. You may get help from others for this project but must write your own code and LATEX write-up. All plots should be well labeled, including legends. Ensemble plots should use the same color for each realization to make the plot more readable.

**Overview:** In this lab you will determine a stochastic model for the SpaceX rocket. You will test this stochastic model within the SpaceX MATLAB game. Finally, you will create a script which uses a set of winning inputs to generate a Monte Carlo simulation of the descending rocket.

**Problem Information:** In this lab you will be attempting to determine a full stochastic system model for the SpaceX rocket. This linear model will take the form

Your system model will have 3 states. The states are altitude, velocity, and radar altimeter error. Your measurement *z* is a radar altimeter measurement.

*System Dynamics:* The system dynamics matrix, described by the matrix *F,* captures the relationship between position and velocity as well as the dynamics of the radar altimeter error.

*System inputs:* The system has a single input *u* which describes the combined effect of gravity and vehicle thrust. This input is given as an acceleration, not a force. The actual inputs *u* will be provided by the MATLAB game based on your key presses. You do not need to determine values for *u*, only how these inputs effect the propagation of the system states.

*System Measurement:* The radar altimeter measurement *z* attempts to measure altitude. The radar altimeter measurement is corrupted by a random process (describing the altimeter error) as well as white Gaussian noise.

*System noises:* The SpaceX system model includes two stochastic processes which must be modeled. The first stochastic process captures the uncertainty of the vehicle thrust. Load the data called Project\_Data.

The thrust data contained in Project\_Data.mat is the measured error in the rocket thrust over time. The thrust data is a 200x1000 matrix of data. The columns represent individual realizations of thrust data over time, sampled at 10 Hz. There are 1000 realizations of the thrust data.

Project\_Data.mat also includes a single long data collect of radar altimeter error data, sampled at 10 Hz.

**Step 1:** Plot the ensemble of the thrust error data as well as the ensemble variance of the thrust error data

**Step 2:** Determine a model for a random process which describes the thrust error. Write out equations for the mean and variance of the random process. Include numerical values you empirically determine from the thrust data. Draw the block diagram for the random process which describes the thrust error. Write out a single-state equation which fits the form

**Step 3:** Determine a model for a random process which describes the radar altimeter error. Plot the autocorrelation of the radar altimeter error data. Write out equations for the mean and variance of the random process as well as the auto-correlation of the random process. Include numerical values you empirically determine from the radar altimeter data. Draw the block diagram for the random process which describes the radar altimeter error. Write out a single-state equation which fits the form

*HINT: The autocorrelation function will capture the effects of the white Gaussian noise and the unknown random process*

**Step 4:** Using your random process models, determine the dynamics noise matrix ***Q*** and *the* measurement noise matrix *R* for the complete system model*.*

**Step 5:** Write out the system dynamics matrix *F.*

**Step 6:** Write out the input matrix *B.*

**Step 7:** Write out the measurement matrix H.

**Step 8:** Discretize your system model. Find the Cholesky decompositions of the *R* and ***Q***  matrices. Open up

SpaceX\_Filter\_Structure.mat.

Overwrite the empty fields with your discrete system model.

**Note:** We desire the transpose of what MATLAB’s *chol* function returns. To address numerical stability issues with the Cholesky decomposition try adding a small amount of noise to the diagonal of the matrix, i.e. *chol(A+eps\*eye(size(A))*

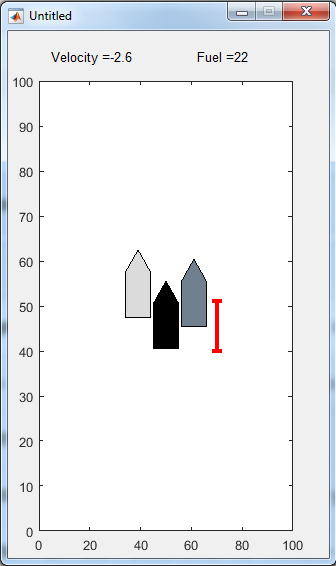
**Step 9:** Fill in the four empty functions

These functions are all a single line of code

1. SpaceX\_Prop\_Mean – This function propagates the mean of the joint PDF describing the SpaceX system forward in time.
2. SpaceX\_Prop\_Cov – This functions propagates the variance of the joint PDF describing the SpaceX system forward in time
3. SpaceX\_Prop\_States – This function propagates an actual realization of the SpaceX system forward in time. Random noise will be generated within the function in order to generate a specific realization.
4. SpaceX\_gen\_meas – This function generates an actual corrupted measurement of the SpaceX system at a given time based on the system state and randomly generated data.

**Step 10:** Verify your filter structure and functions with me before proceeding.

**Step 11:** Place the SpaceX game. There are three rockets on the screen when you play the game.



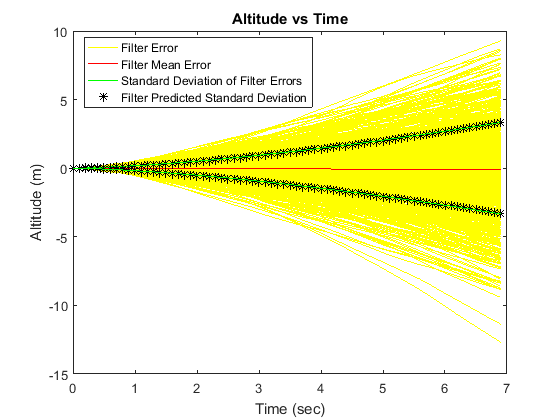
The dark grey rocket (right most) is the true rocket position. You normally do not have access to this position, and it will be what we are trying to actually estimate when we create a Kalman Filter

The light grey rocket (left-most) is what your radar altimeter is reporting. This is based on the true position (dark grey rocket) corrupted with various noise sources

The middle rocket is the mean of the rocket position state obtained by propagating the stochastic system forward in time. In a Kalman filter this would be your best guess of the position. The red bars show the associated covariance with your best guess of the rocket position (+/- 1 standard deviation)

Note that winning the game with the left two rocket positions is very difficult. Note the differences between the behavior and positions of the 3 rockets as the rocket descends, and note how the covariance bars behave. Write a paragraph detailing your observations. Finally, using the rightmost “truth” rocket, win the game to create a set of “Winning Inputs”

**Step 12:** Once you have a set of winning inputs generated by the SpaceX game, you will write a MATLAB script which performs a full Monte Carlo Simulation. This simulation will generate 1000 realizations of the rocket trajectory, given your set of winning inputs. Your “SpaceX\_Prop\_States” and “SpaceX\_gen\_meas” will create your Monte Carlo realizations. These realizations are “truth” and any difference between what your stochastic model predicts and these truth solutions will be a filter error. Note that you only have to propagate your stochastic model once, as the propagation is not effected in any way by the truth solution or generated measurements. Generate a set of three error plots, one for each system state, which show that the stochastic system correctly predicts the performance of the system as obtained in the Monte Carlo simulation. You should use your same filter structure *m* as well as the 4 functions you created. Error plots should look similar to the example plot below.



**Step 13:** The rocket lands successfully if the altitude is less than 1 meter and the velocity (absolute value) is less than 1 m/s. Of your 1000 Monte Carlo runs, how many have a successful rocket landing?