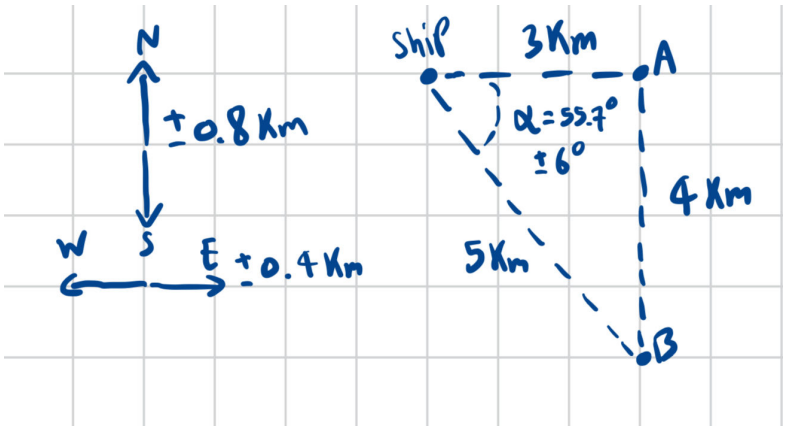
Assignment 7 / Seif Elkhashab / 04-26-2024

Static And Dynamic Estimation

Static Estimation:

This problem is taken from Applied Optimal Control by A. E. Bryson. A navigator has an estimated position of his ship $3\ km$ west of a landmark A and $5\ km$ from another landmark B, with an estimated accuracy of $\pm 0.8\ km$ in the north-south direction and $\pm 0.4\ km$ in the east-west direction. He then measures the angle α between A and B as $55.7\ deg$ and estimates the accuracy of this measurement to be $\pm 6\ deg$. Note that A is $4\ km$ directly north of B and that the angle measurement locates the ship on a circle that passes through A and B. Hint: Draw a picture.



(1) Read "StaticEstimation.pdf" which was posted in Canvas.

I affirm that I have read the StaticEstimation pdf posted to canvas.

(2) Write matlab code to estimate the ship's most likely position and a $1-sigma$ uncertainty ellipse for the estimate. Note that Matlab code for a very similar example is posted in "MeasurementUpdateExample.mlx" in Canvas.

- 1) The first step is I need to relate the angle α to the ship's x and y position. It took me a long time but I realized that while the ship is 3 km west of A, no north-south information was given, which means that the ship can be more north or south of point A. To get α we can split the coordinate frame at the ship so there are two right angles represented by the formula below:

$$\alpha = \tan^{-1}\left(\frac{4-y}{x}\right) + \tan^{-1}\left(\frac{y}{x}\right)$$

Alpha can be written as:

$$\alpha = \text{atan}\left(\frac{y}{x}\right) - \text{atan}\left(\frac{y-4}{x}\right)$$

- 2) Now that a relationship between α and the ship's position is defined, we can take the jacobian of α to see how changes in x and y affect the angle between points A and B, α.

The jacobian of alpha:

$$\left(\frac{y-4}{x^2+y^2-8\,y+16}-\frac{y}{x^2+y^2}\quad\frac{x}{x^2+y^2}-\frac{x}{x^2+y^2-8\,y+16}\right)$$

- 3) Now we can take the Taylor series around the nominal value we have

$$z(\bar{x}+\delta x)=z(\bar{x})+\frac{\partial h}{\partial x}\delta x$$

This allows us to define δz,

$$\delta z=z(\bar{x}+\delta x)-z(\bar{x}).$$

Therefore, we have the linear measurement equation

$$\delta z=C\delta x \text{ where } C=\frac{\partial h}{\partial x}(\bar{x})$$

The uncertainty matrix shows the uncertainity in the x and y direction:

$$\begin{pmatrix}0.16 & 0\\ 0 & 0.64\end{pmatrix}$$

z_{bar} is found to be:

$$\bar{z}=-53.13$$

- 4) The estimate can be made with the following equation, I used z_i + bar_z because subtracting bar_z seemed to place the ship very far.

$$\hat{x}\simeq[x_0;y_0]+\mathbb{E}[\delta X|\delta Z]$$
$$\mathbb{E}[\delta X|\delta Z]=(M^{-1}+C^TR^{-1}C)^{-1}C^TR^{-1}\delta z$$
$$\delta z_i=z_i-\bar{z}_i$$

The estimated ship's location:

$$\bar{x}=\begin{pmatrix}-3.032760636260196364927922914778 5\\ 4.174723393387713652032289308135 8\end{pmatrix}$$

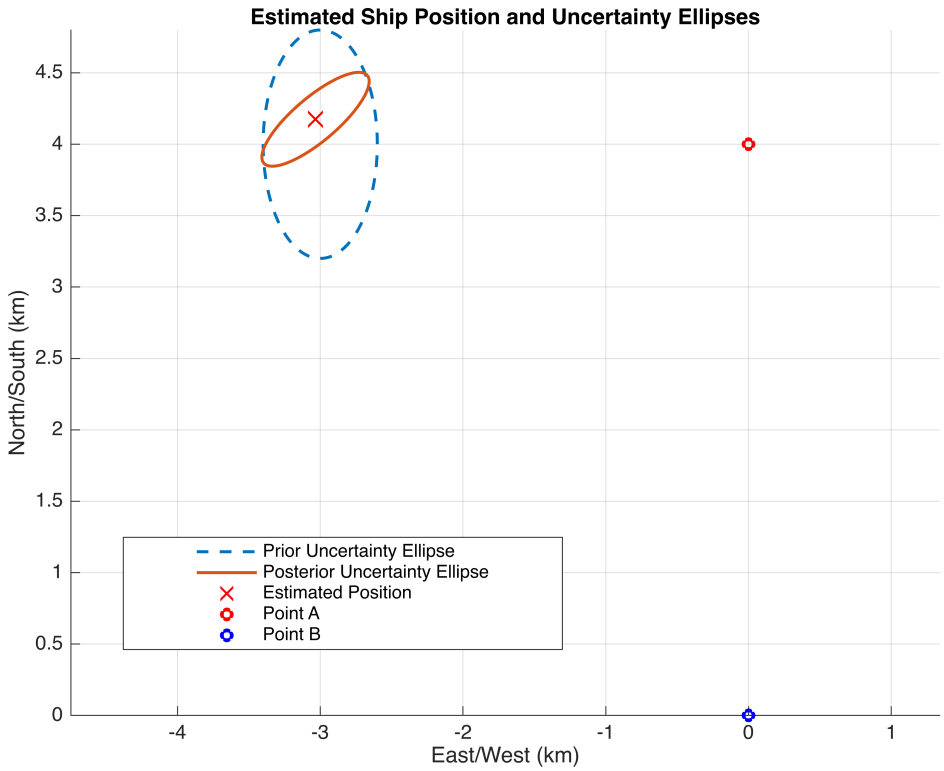
The eigenvalues of p are shown below:

$$\text{eig}(p)=\begin{pmatrix}0.225819300376471112561397688013 35 & 0\\ 0 & 0.023645404955866640343505443136357\end{pmatrix}$$

The eigenvectors of p are shown below:

$$\overrightarrow{\text{eig}(p)}=\begin{pmatrix}0.762863189661663750305765198887 64 & -0.646559938334592726251825313747 81\\ 0.646559938334592797831480560858 16 & 0.762863189661663810972655555249 72\end{pmatrix}$$

(3) Plot the prior and posterior uncertainty ellipse for the ship's position.



Dynamic Estimation:

Data taken from your drone's sensors has been provided in Canvas under the Data module. If you have any issues with this data set, please notify me. You will use a Kalman filter to estimate the drone's state for $roll$ and $pitch$. You will assume that the $roll$ and $pitch$ ϕ, θ are unknown. You should assume that the drone's airspeed V_a and angular velocities p, q, r are measurable from the drone's sensors. The drone's IMU also provides accelerometer (accels) measurements in the x,y ,z , body axis, A_x, A_y ,A_z . The estimation model you should assume are shown in the "book notes on state estimation" link between slides 39 and 42. The book solves this problem with an extended Kalman filter, but you will solve this problem with a linear Kalman filter.You should use a linear kalman filter block that is built-in to Matlab's simulink.

To help you get started, here are some basic steps:

(1) Calculate the mean and covariance of the process and measurement noise for each of the provided measurements. The process noise and measurement noise covariances are Q and R. Are the process noise and measurement noise uncorrelated, zero mean, normal, and white?

Below the data structure found in the noise_data.txt file can be seen. The data file provides:

- t, the time
- V_a, the airspeed
- $\omega_x,\ \omega_y,\ \omega_z$, the angular accelerations
- $A_x,\ A_y,\ A_z$, the linear accelerations

Showing the first and last couple lines:

t	V_a	w_x	w_y	w_z	A_x	A_y	A_z
0.1	0.0067388	0.0011706	4.0904e-05	0.000276	0.00018357	-0.0026906	0.02
0.2	0.0078118	0.00086908	-6.005e-05	0.00014125	0.00017251	-0.0021615	0.01
0.3	0.0088125	0.00095793	1.632e-06	0.00023272	7.2496e-05	-0.0015583	0.0
0.4	0.0092137	0.00063063	-0.00022303	-0.00013007	0.00014666	-0.0011823	-0.002
0.5	0.0092068	0.0020131	0.00071479	0.00038832	0.00013041	-0.00068844	0.007
199.7	0.0080976	-0.0028285	-0.00049812	-8.6352e-05	-0.0011922	0.011642	-0.01
199.8	0.0074276	-0.0029914	-0.00056432	-0.00021851	-0.0009607	0.010228	-0.03
199.9	0.0067381	-0.0030697	-0.0003212	1.2196e-05	-0.00078742	0.0088264	-0.0
200	0.0063504	-0.0026733	2.7348e-05	-8.6193e-06	-0.00049991	0.0071026	-0.02
200.1	0.0062323	-0.0038624	-0.00060648	-7.6378e-05	-0.00046253	0.0056274	-0.01

The noise means from noise_data:

	V_a	w_x	w_y	w_z	A_x	A_y	A_z
Means	0.0088563	7.896e-06	1.6639e-07	-4.914e-06	4.9253e-06	5.8702e-05	-5.9788e-05

The Covariance Matrix from noise_data:

	V_a	w_x	w_y	w_z	A_x	A_y	A
V_a	3.1115e-05	8.5941e-07	5.2344e-08	2.2362e-08	3.53e-07	-1.1888e-06	-1.05
w_x	8.5941e-07	3.3488e-06	5.8066e-07	2.151e-07	-4.4061e-07	-1.8151e-07	2.82
w_y	5.2344e-08	5.8066e-07	4.8052e-07	1.5405e-07	5.7363e-08	-5.7392e-07	2.32
w_z	2.2362e-08	2.151e-07	1.5405e-07	1.0829e-07	-1.0802e-07	2.0543e-07	9.13
A_x	3.53e-07	-4.4061e-07	5.7363e-08	-1.0802e-07	1.9095e-06	-3.1973e-06	-1.00
A_y	-1.1888e-06	-1.8151e-07	-5.7392e-07	2.0543e-07	-3.1973e-06	4.338e-05	-1.39
A_z	-1.0557e-06	2.8208e-05	2.3287e-05	9.1327e-06	-1.0022e-05	-1.3925e-05	0.0

The noise means from maneuver_data:

	V_a	w_x	w_y	w_z	A_x	A_y	A_z
Means	0.009131	2.5776e-07	-2.2144e-05	4.4692e-06	-1.4786e-05	7.6272e-05	0.0021453

The Covariance Matrix from maneuver_data:

	V_a	w_x	w_y	w_z	A_x	A_y	A
V_a	4.2866e-05	5.7425e-07	-1.3262e-07	-1.8065e-07	4.3065e-07	1.2853e-06	-4.65
w_x	5.7425e-07	2.2469e-06	4.0727e-07	2.0799e-07	-5.8549e-07	1.161e-07	2.16
w_y	-1.3262e-07	4.0727e-07	4.2759e-07	1.0975e-07	9.0366e-08	-4.535e-07	1.90
w_z	-1.8065e-07	2.0799e-07	1.0975e-07	9.0772e-08	-1.2274e-07	1.5637e-07	7.46
A_x	4.3065e-07	-5.8549e-07	9.0366e-08	-1.2274e-07	2.0779e-06	-2.0519e-06	-1.35
A_y	1.2853e-06	1.161e-07	-4.535e-07	1.5637e-07	-2.0519e-06	2.8756e-05	-2.07
A_z	-4.6542e-05	2.1674e-05	1.9078e-05	7.4601e-06	-1.3561e-05	-2.0742e-05	0.0

(2) Derive the linearized dynamics of the drone around an operating point. This will yield state-space A,B ,C ,D matrices in continuous time

The plant A matrix:

$$A_{\text{plant}} = \begin{pmatrix} q\cos(\phi)\tan(\theta) - r\sin(\phi)\tan(\theta) & r\cos(\phi)\ (\tan(\theta)^2+1) + q\sin(\phi)\ (\tan(\theta)^2+1) \\ -r\cos(\phi) - q\sin(\phi) & 0 \end{pmatrix}$$

The linearized plant A matrix:

$$A_{\text{lin}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

The plant B matrix:

$$B_{\text{plant}} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \end{pmatrix}$$

The linearized plant B matrix:

$$B_{\text{lin}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The plant C matrix:

$$C_{\text{plant}} = \begin{pmatrix} 0 & \frac{981\cos(\theta)}{100} + q\,v\cos(\theta) \\ -\frac{981\cos(\phi)\cos(\theta)}{100} & \frac{981\sin(\phi)\sin(\theta)}{100} - p\,v\cos(\theta) - r\,v\sin(\theta) \\ \frac{981\cos(\theta)\sin(\phi)}{100} & \frac{981\cos(\phi)\sin(\theta)}{100} + q\,v\sin(\theta) \end{pmatrix}$$

The linearized plant C matrix:

$$C_{\text{lin}} = \begin{pmatrix} 0 & \frac{981}{100} \\ -\frac{981}{100} & 0 \\ 0 & 0 \end{pmatrix}$$

Noise Effect Matrix:

$$G(x) = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \end{pmatrix}$$

The linearized Noise Effect Matrix:

$$\text{Gx}_{\text{lin}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

(3) Data is provided to you at $10\ \ hz$. This defines the “sample time“ T_s you should use in your Kalman filter.

The sample time used is shown below, 10 hz:

$$T_s = \frac{1}{10}$$

(4) Discretize the continuous time dynamics. You may use the simplified formulas on slide 33 for A_d and B_d.

The Q matrix is shown below:

$$Q = \begin{pmatrix} 0.0000033488 & 0 & 0 & 0 \\ 0 & 0.0000004805 & 0 & 0 \\ 0 & 0 & 0.0000001083 & 0 \\ 0 & 0 & 0 & 0.0000311148 \end{pmatrix}$$

The R matrix is shown below:

$$R = \begin{pmatrix} 0.0000019095 & 0 & 0 \\ 0 & 0.00004338 & 0 \\ 0 & 0 & 0.0018451061 \end{pmatrix}$$

(5) Discretize the continuous process noise covariance Q by assuming the discrete process noise is $Q_D = Q * T_s$.

(6) Set the state estimate inital conditions and choose an initial estimation error covariance.

The initial state vector estimate:

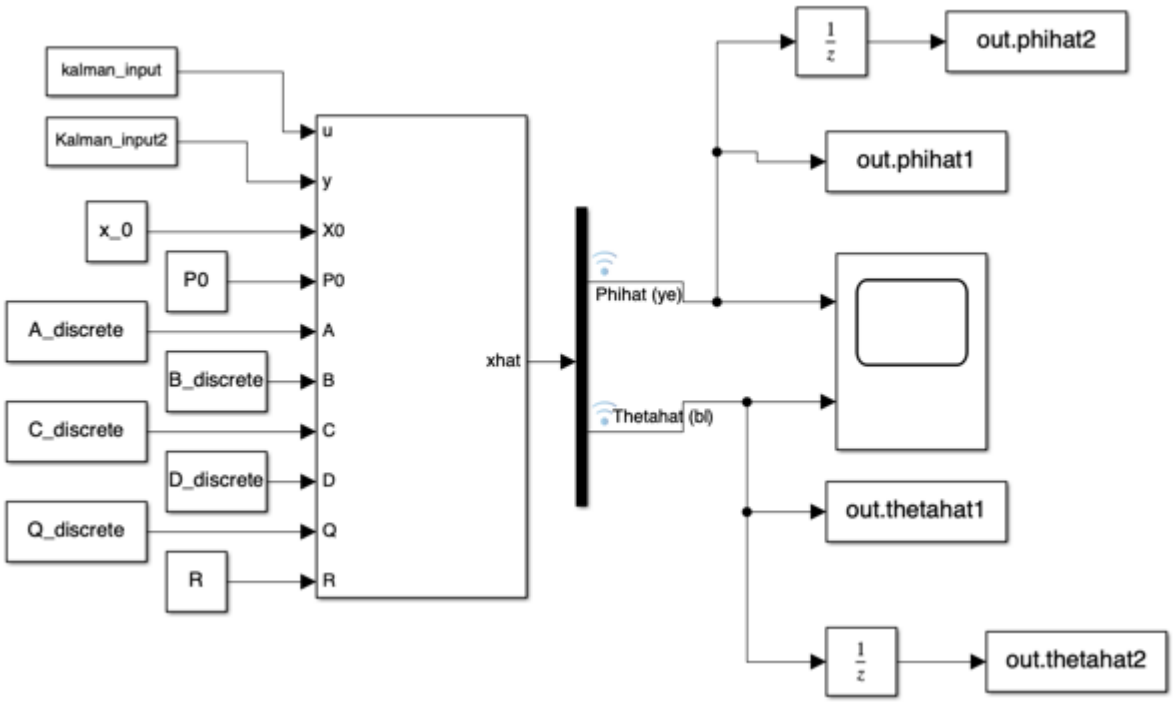
$$X_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

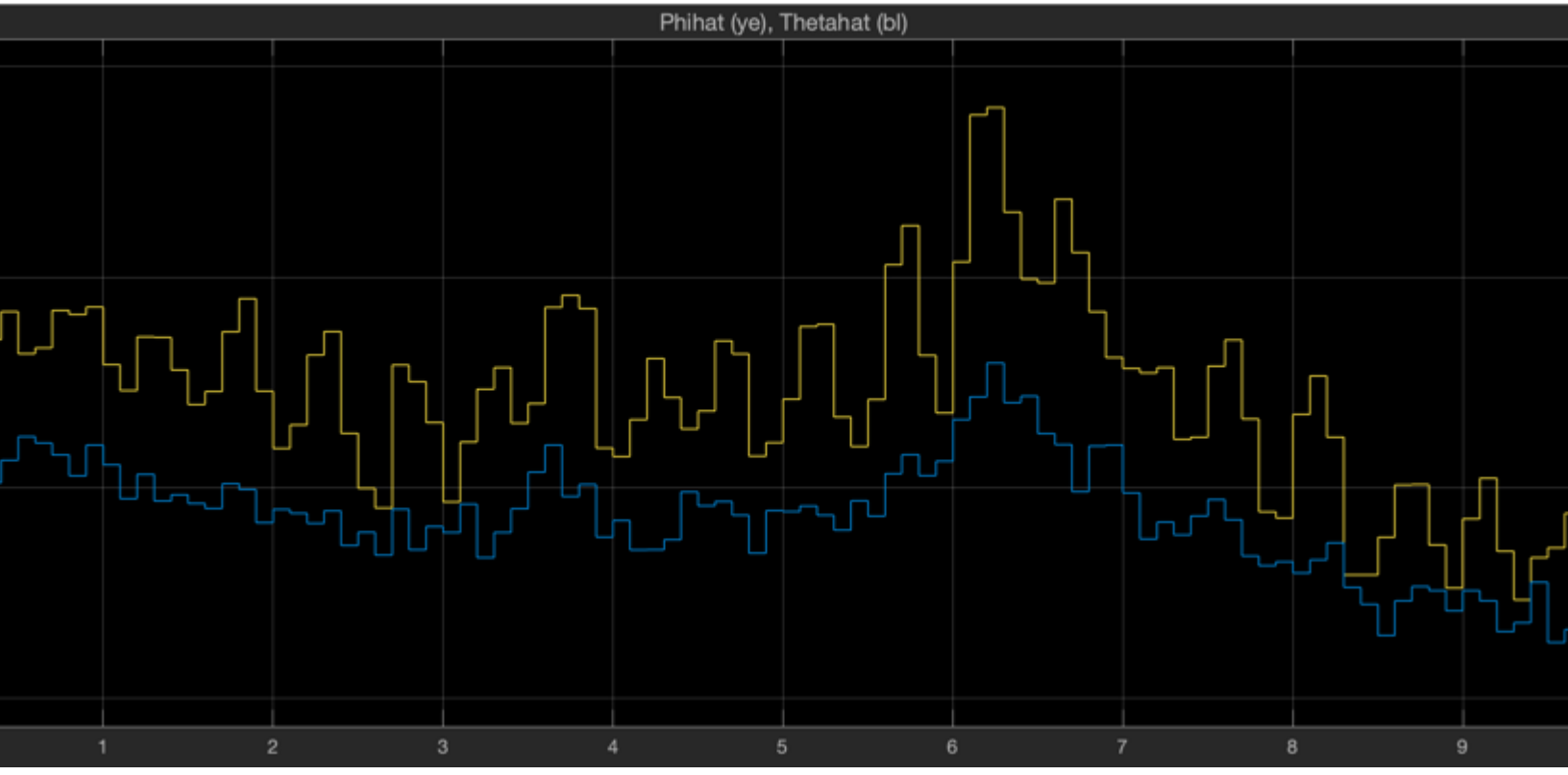
The initial coveriance matrix:

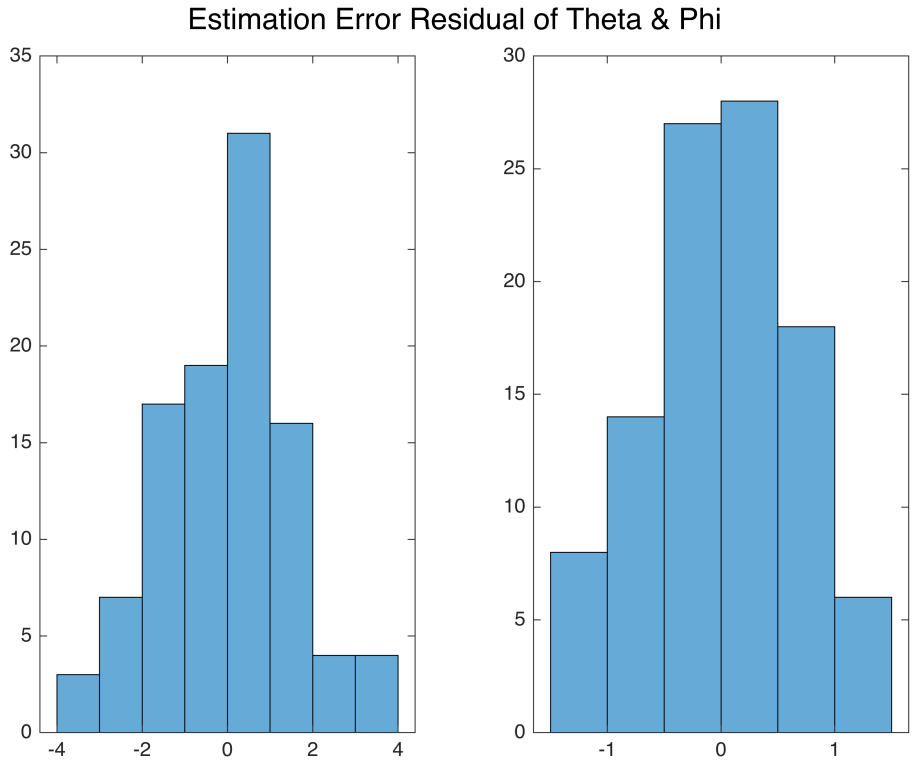
$$P_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

You should turn in a summary of your work, plots of the state estimates, and a histogram of the estimation error residuals. Recall that the estimation error residual is the difference between what you measure at the current time step $y[n]$ and your estimated output on the previous time step $\hat{y}[n-1] = C\hat{x}[n-1] + Du[n-1]$.

As can be seen below, the khalman filter was successfully implemented with good results. The error residuals showed a normal distribution as expected and is as expected.


kalman_input
Kalman_input2
x_0
P0
A_discrete
B_discrete
C_discrete
D_discrete
Q_discrete
R
u
y
X0
P0
A
B
C
D
Q
R
xhat
Phihat (ye)
Thetahat (bl)
1/z
out.phihat2
out.phihat1
out.thetahat1
1/z
out.thetahat2


Phihat (ye), Thetahat (bl)
1
2
3
4
5
6
7
8
9



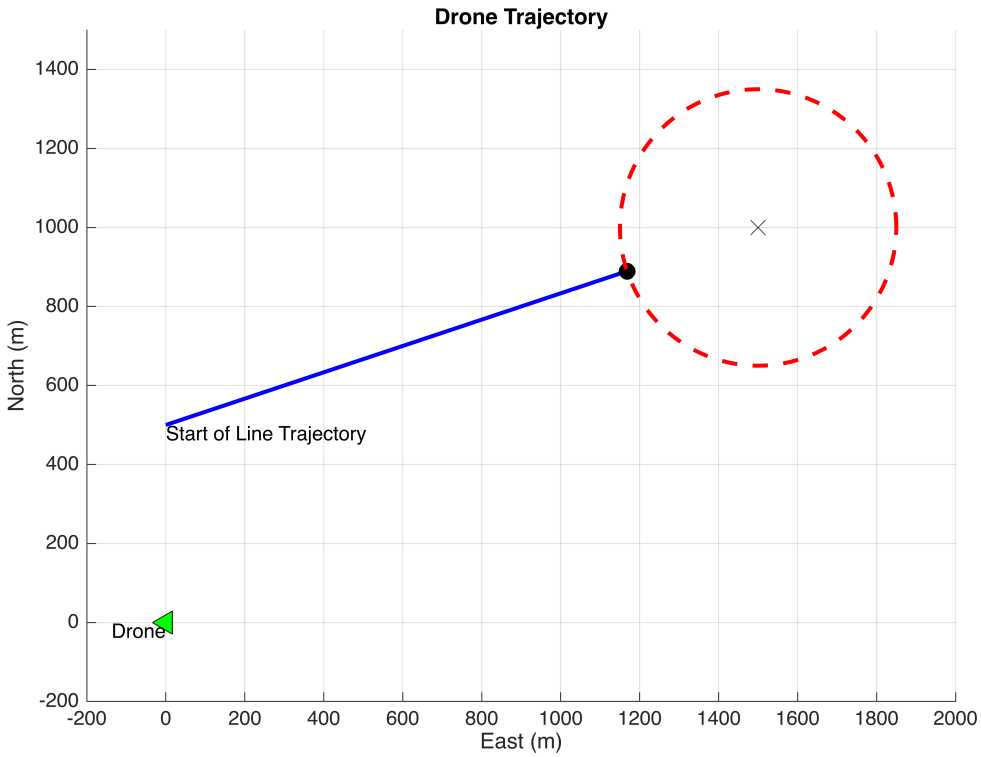
Steering:

Assume that the drone obeys Dubin's vehicle dynamics and uses bank-to-turn steering (the coordinated turn constraint). The vehicle has an airspeed controller that regulates airspeed to $27\ m/s$ and an altitude controller that manages its altitude. Assume that the drone's minimum turning radius is $160\ m$ (this specifies its maximum lateral acceleration). Assume that the drone may only bank up to $\pm 30\ degrees$, and that it perfectly tracks its airspeed and flight path angle. Assume also that there is no wind.

$$V_g = V_a$$
$$\dot{p}_n = V_g \cos\psi$$
$$\dot{p}_e = V_g \sin\psi$$
$$\dot{\psi} = \frac{g \tan\phi}{V_g}$$
$$\phi \leq 30^\circ \quad , \; a = v * \dot{\psi} = \frac{V^2}{R}, \; R = 160 \text{ m}$$

Visualizing the problem:

Below, the problem statement was plotted and will later be used to demonstrate how well the drone follows the requested trajectory. I encountered a lot of trouble trying to define the problem, but plan to complete it at a later time.



Stating the Dubins Dynamics:

Sources:

Bryson and Ho Chapter 12 Example 1

MeasurnmentUpdateExample.mlx