Ross Fischer
Fall 2023
Homework 7

Statistical Estimation for Dynamical Systems
ASEN 5044

Intro

Consider an aircraft moving in a plane with constant speed (i.e. magnitude of velocity) and turning with a constant angular rate. Such a model is often used by air traffic control tracking algorithms to describe an aircraft executing coordinated turns. Given 2D inertial position variables $\xi(t)$ (East position) and $\eta(t)$ (North position), the equations of motion are

$$\ddot{\xi} = -\Omega\dot{\eta}$$
$$\ddot{\eta} = \Omega\dot{\xi}$$

where Ω is the constant angular rate, such that $\Omega > 0$ implies a counterclockwise turn. Using the state representation $x(t) = \begin{bmatrix}\xi & \dot{\xi} & \eta & \dot{\eta}\end{bmatrix}^T$, it can be shown that

$$e^{A\Delta t} = \begin{bmatrix} 1 & \frac{sin(\Omega\Delta t)}{\Omega} & 0 & -\frac{1-cos(\Omega\Delta t)}{\Omega} \\ 0 & cos(\Omega\Delta t) & 0 & -sin(\Omega\Delta t) \\ 0 & \frac{1-cos(\Omega\Delta t)}{\Omega} & 1 & \frac{sin(\Omega\Delta t)}{\Omega} \\ 0 & sin(\Omega\Delta t) & 0 & cos(\Omega\Delta t) \end{bmatrix}$$

where A is the CT LTI state matrix for this system. Use this model for all the questions in this assignment.

Exercise 1

(1) Find A (ignore measurements y and disregard process noise for now). What is the DT LTI model for this system if we are ignoring the process noise?

Start with the CT LTI system $\dot{x} = Ax + Bu$

$$\begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \\ \dot{\eta} \\ \ddot{\eta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\Omega \\ 0 & 0 & 0 & 1 \\ 0 & \Omega & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix}, \quad \rightarrow \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\Omega \\ 0 & 0 & 0 & 1 \\ 0 & \Omega & 0 & 0 \end{bmatrix} + Bu$$

Now, F is simply taken from $F = e^{A\Delta t}$. Ignoring y and process noise will produce a DT LTI system of the form $x(k+1) = Fx(k) + G(k)u(k)$.

$$\begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix}_{(k+1)} = \begin{bmatrix} 1 & \frac{sin(\Omega\Delta t)}{\Omega} & 0 & -\frac{1-cos(\Omega\Delta t)}{\Omega} \\ 0 & cos(\Omega\Delta t) & 0 & -sin(\Omega\Delta t) \\ 0 & \frac{1-cos(\Omega\Delta t)}{\Omega} & 1 & \frac{sin(\Omega\Delta t)}{\Omega} \\ 0 & sin(\Omega\Delta t) & 0 & cos(\Omega\Delta t) \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix}_{(k)} + G_{(k)}u_{(k)}$$

Exercise 2

(2) For the following parts (a), (b), assume a sampling time $\Delta t = 0.5sec$, $\Omega = 0.045rad/s$, and an initial DT state uncertainty $x(0) \sim \mathcal{N}(\mu(0), P(0))$ at $k = 0$. Again, disregard process noise and measurements for now (we will revisit this problem in the final HW8.)

(a) Derive an expression for predicted DT state mean $\mu(k) = E[x(k)]$ and state covariance $P(k) = E[(x(k) - \mu(k))(x(k) - \mu(k))^T]$ for an arbitrary number of steps $k = N$ into the future, starting from $k = 0$. Be sure to express your answer in terms of N, $\mu(0)$, $P(0)$, and the DT state transition matrix F.

(a) For this problem, we are assuming there is no input $Gu_{(k)}$ or process noise. Note $\mu_{(0)}$ is referring to the mean of the distribution of x. First note the first few terms:

$$x_{(1)} = Fx_{(0)}, \qquad x_{(2)} = Fx_{(1)} = F^2x_{(0)}, \qquad x_{(3)} = F^3x_{(0)}$$
$$x_{(k)} = F^kx_{(0)}, \quad or \quad x_{(N)} = F^Nx_{(N)}$$

Since $x_{(k)}$ is a linear combination of $x_{(0)}$...Or in other words, propagate $\mu_{(0)}$ using F, after N steps...

$$\mu_{(N)} = F^N\mu_{(0)}$$

Note that the noise covariance matrix, R is usually propagated to a new time step in the Batch LLS equation $P_e = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$. Despite there being no noise, we still have a state covariance

$$\begin{aligned}
&P_{(k)} = E[(x_{(k)} - \mu_{(k)})(x_{(k)} - \mu_{(k)})^T] \\
and \quad &P_{(0)} = E[(x_{(0)} - \mu_{(0)})(x_{(0)} - \mu_{(0)})^T] \\
and \quad &P_{(N)} = E[(F^Nx_{(0)} - F^N\mu_{(0)})(F^Nx_{(0)} - F^N\mu_{(0)})^T] \\
&P_{(N)} = F^NE[(x_{(0)} - \mu_{(0)})(x_{(0)} - \mu_{(0)})^T](F^N)^T \\
&P_{(N)} = F^NP_{(0)}(F^N)^T
\end{aligned}$$

So we have,

$$\mu_{(N)} = F^N\mu_{(0)}, \qquad and \qquad P_{(N)} = F^NP_{(0)}(F^N)^T$$

(b) Assume

$$\mu(0) = \begin{bmatrix} 0m \\ 85cos(\pi/4)\frac{m}{s} \\ 0m \\ -85sin(\pi/4)\frac{m}{s} \end{bmatrix} \qquad P_a(0) = \begin{bmatrix} 10m^2 & 0 & 0 & 0 \\ 0 & 2(\frac{m}{s})^2 & 0 & 0 \\ 0 & 0 & 10m^2 & 0 \\ 0 & 0 & 0 & 2(\frac{m}{s})^2 \end{bmatrix}$$

Perform a 300 step calculation of $\mu(k)$ and $P(k)$ for $k = 1, 2, \ldots, 300$ starting from these initial conditions. Plot the following in separate plots:

- each state element of $\mu(k)$ versus time, along with $\pm 2\sigma$ (2 standard deviations) upper/lower bounds showing the uncertainty for each state at each time k (these can be obtained by looking at the appropriate multiples of the square root of the corresponding diagonal entry of $P(k)$, since each diagonal entry is the variance of some state)
- only the positive 2σ values versus time for each state at time k.

Be sure to label all axes with appropriate units, and comment on the results.

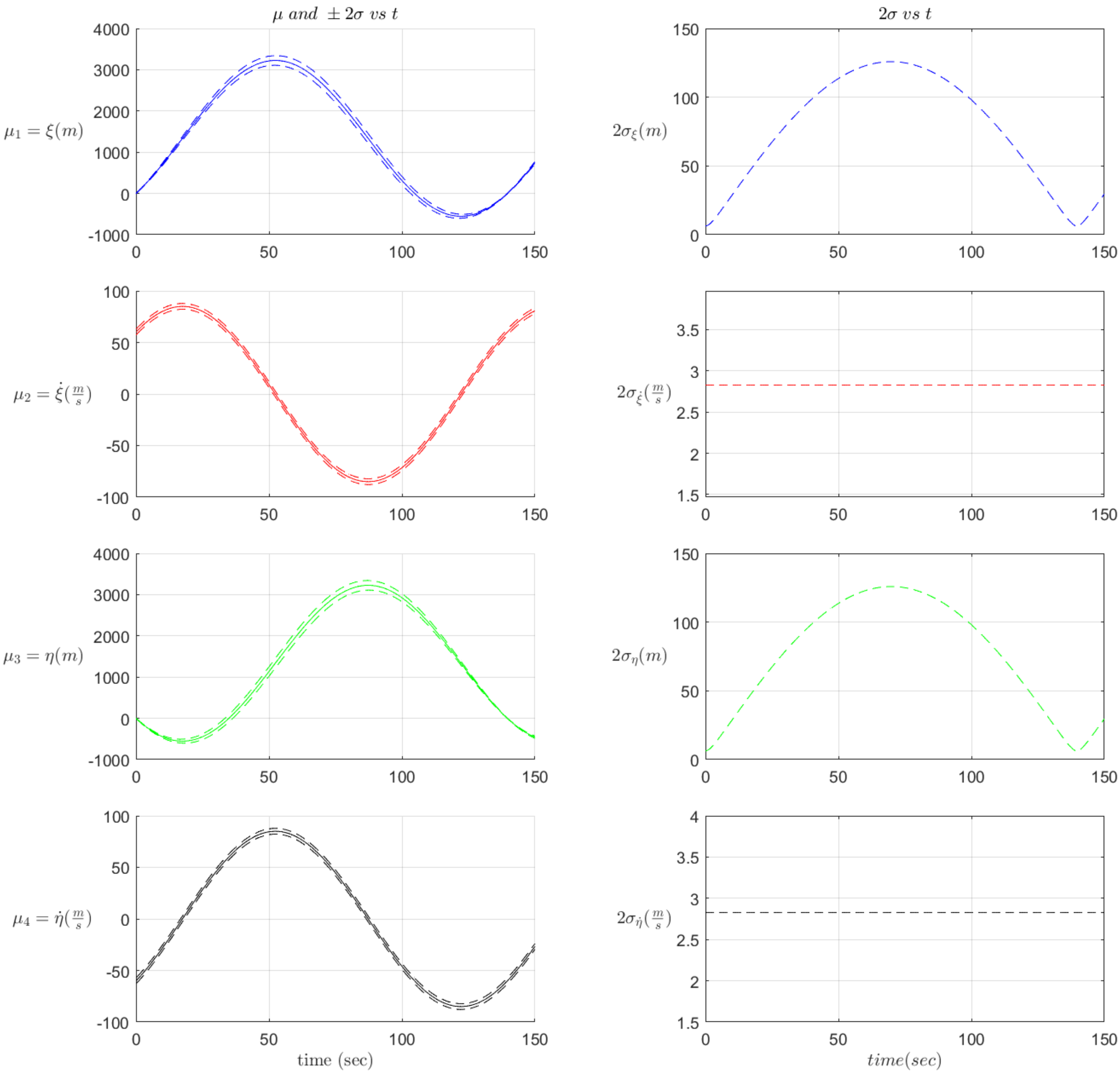


Figure 1: Problem 2)b)

We see that easting and northing position follows sinusoidal paths, suggesting a circular motion. Similarly, the respective velocities follow a sinusoidal motion. If the velocities were combined into a magnitude, we would see a constant velocity magnitude. The uncertainty of the position states grow as the aircraft flies further away, but interestingly it decreases as it returns to it's original position. The uncertainty of the velocities are constant. .

.

.

Exercise 3

3. Consider now computing the probability of collision for two aircraft turning independently at the same altitude at the same time, where the initial position and velocity of the aircraft are unknown. Let x_a denote the state of aircraft a and let x_b denote the state of aircraft b. A 'collision' will simply be modeled as an event where both aircraft occupy a bounded rectangular region of airspace at the same time. That is, if $r_a(k) = [\xi_a(k), \eta_a(k)]^T$ and $r_b(k) = [\xi_b(k), \eta_b(k)]^T$, a collision occurs at time k whenever $r_c(k) \equiv r_a(k) - r_b(k)$ lies inside the region R defined by $\Delta\xi(k) \in [-\xi_R, \xi_R]$ and $\Delta\eta(k) \in [-\eta_R, \eta_R]$, where $\Delta\xi(k) \equiv \xi_a(k) - \xi_b(k)$, $\Delta\eta(k) \equiv \eta_a(k) - \eta_b(k)$, and ξ_R and η_R are known constants.

At time $k = 0$, suppose both aircraft states are independent and normally distributed, with $x_a(0) \sim \mathcal{N}(\mu_a(0), P_a(0)$ and $x_b(0) \sim \mathcal{N}(\mu_b(0), P_b(0))$. Also assume that Δt=0.5 s, $\Omega_a = 0.045$ rad/s and $\Omega_b = -0.045$ rad/s (i.e. the aircraft turn in opposite directions at the same rate). **Again, ignore process noise and measurements.**

Figure 2: Problem 3)

(a) Derive the expression for the mean and covariance parameters of the pdf $p(r_c(k))$ at any time k, given that $r_c(k) \sim \mathcal{N}(\mu_{r_c}(k), P_{r_c}(k))$ (i.e. the pdf for $r_c(k)$ must be a multivariate Gaussian). Your answers should be in terms of components of F_a and F_b (the STMs for each aircraft), as well as components of $\mu_a(0), \mu_b(0)$ and $P_a(0), P_b(0)$. (**Hint:** The concept of marginalizing multivariate Gaussian pdfs was covered earlier in the course, and will be very useful to apply here.)

Figure 3: Problem 3)a)

Start by using the same approach from exercise 2 to get the mean of the state, x_c...

$$x_{c(k)} = x_{a(k)} - x_{b(k)} \qquad \mu_{b(k)} = F_a^k \mu_{a(0)}$$
$$E[x_c] = E[x_a] - E[x_b] \qquad \mu_{x_b(k)} = F_{x_b}^k \mu_{x_b(0)}$$
$$\mu_{x_c(k)} = \mu_{x_a(k)} - \mu_{x_b(k)}$$

$$\mu_{x_c(k)} = F_{x_a}^k \mu_{x_a(0)} - F_{x_b}^k \mu_{x_b(0)}$$

The variance of a difference of two independent variables is the sum of their variances...

$$\begin{aligned} P_{x_c(k)} &= P_{x_a(k)} + P_{x_b(k)} \\ and \quad P_{x_a(k)} &= F_{x_a}^k P_{x_a(0)} (F_{x_a}^k)^T \\ and \quad P_{x_b(k)} &= F_{x_b}^k P_{x_b(0)} (F_{x_b}^k)^T \\ so \quad P_{x_c(k)} &= F_{x_a}^k P_{x_a(0)} (F_{x_a}^k)^T + F_{x_b}^k P_{x_b(0)} (F_{x_b}^k)^T \end{aligned}$$

And so we have $x_{c(k)} \sim \mathcal{N}(\mu_{x_c(k)}, P_{x_c(k)})$.
Now, "extract" the relevant information from μ_{x_c} and P_{x_c} by setting up the following equation...

$$r_{(k)} = \mathcal{J} x_{(k)}$$

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix} \qquad \rightarrow \quad \mathcal{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Now solve for $\mu_{r_{c(k)}}$ and $P_{r_c(k)}$.

$$\begin{aligned}\mu_{r_{c(k)}} &= E[r_{c(k)}] = E[r_{a(k)} - r_{b(k)}] \\ E[r_{c(k)}] &= \mathcal{J}F_a^k E[r_{a,(0)}] - \mathcal{J}F_b^k E[r_{b,(0)}]\end{aligned}$$

$$P_{r_c(k)} = E[(r_{c(k)} - \mu_{r_{c(k)}})(r_{c(k)} - \mu_{r_{c(k)}})^T]$$

Instead of fully deriving $P_{r_c(k)}$... we know which elements of $P_{x_c(k)}$ correspond to the elements of the states x_c. Simply pass $P_{x_c(k)}$ through a "$\mathcal{J}$" filter...
The full equation for $\mu_{r_c(k)}$ and $P_{r_c(k)}$ is

$$\begin{aligned}\mu_{r_c(k)} &= \mathcal{J}(F_{x_a}^k \mu_{x_a(0)} - F_{x_b}^k \mu_{x_b(0)}) \\ P_{r_c(k)} &= \mathcal{J}(F_{x_a}^k P_{x_a(0)}(F_{x_a}^k)^T + F_{x_b}^k P_{x_b(0)}(F_{x_b}^k)^T)\mathcal{J}^T\end{aligned}$$

(b) Use the result from part a above to derive an integral expression for the probability of collision at any time k.

Figure 4: Problem 3)b)

$$Where \quad r_c = r_a - r_b = \begin{bmatrix}\xi_c \\ \eta_c\end{bmatrix}$$

$$p(r_{c(k)}) \sim \mathcal{N}(u_{c(k)}, P_{c(k)})$$

$$p(r_{c(k)}) = \frac{1}{(2\pi)^{2/2}\sqrt{|P_{c(k)}|}} \cdot e^{-\frac{1}{2}[(r_{c(k)} - \mu_{r_c(k)})^T P_{c(k)}^{-1}(r_{c(k)} - \mu_{r_c(k)})]}$$

$$P(collision!) = \int_{-\xi_R}^{\xi_R}\int_{-\eta_R}^{\eta_R} p(r_{c(k)}) d\eta_c d\xi_c = P([-\xi_R < \xi_c < \xi_R] \cap [-\eta_R < \eta_c < \eta_R])$$

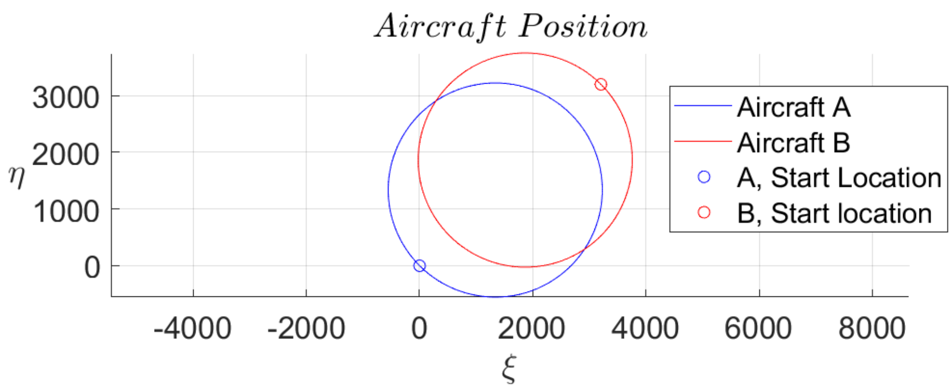
$$P(collision!) = \frac{1}{(2\pi)^{2/2}\sqrt{|P_{c(k)}|}} \int_{-\xi_R}^{\xi_R}\int_{-\eta_R}^{\eta_R} exp(-\tfrac{1}{2}[(r_{c(k)} - \mu_{r_c(k)})^T P_{c(k)}^{-1}(r_{c(k)} - \mu_{r_c(k)})])\ d\eta_c\ d\xi_c$$

(c) Plot the probability of collision vs. time using a simulation run of 150 secs starting from the following initial conditions

$$\begin{aligned}\mu_a(0) &= [0\ m, 85\cos(\pi/4)\ m/s, 0\ m, -85\sin(\pi/4)\ m/s]^T \\ P_a(0) &= \text{diag}([10\ m^2, 4\ (m/s)^2, 10\ m^2, 4\ (m/s)^2], \\ \mu_b(0) &= [3200\ m, 85\cos(\pi/4)\ m/s, 3200\ m, -85\sin(\pi/4)\ m/s]^T \\ P_b(0) &= \text{diag}([11\ m^2, 3.5\ (m/s)^2, 11\ m^2, 3.5\ (m/s)^2],\end{aligned}$$

and using $\xi_R = 100$ m, $\eta_R = 100$ m. Be sure to label axes with appropriate units, and comment on the results. In particular, at what time steps are the vehicles in greatest danger of colliding? Also, what happens to the probability of collision as time increases? What explains this behavior? (Hint: Matlab's `mvncdf.m` command is useful here – read the documentation for this function to learn more about it).

Figure 5: Problem 3)c)



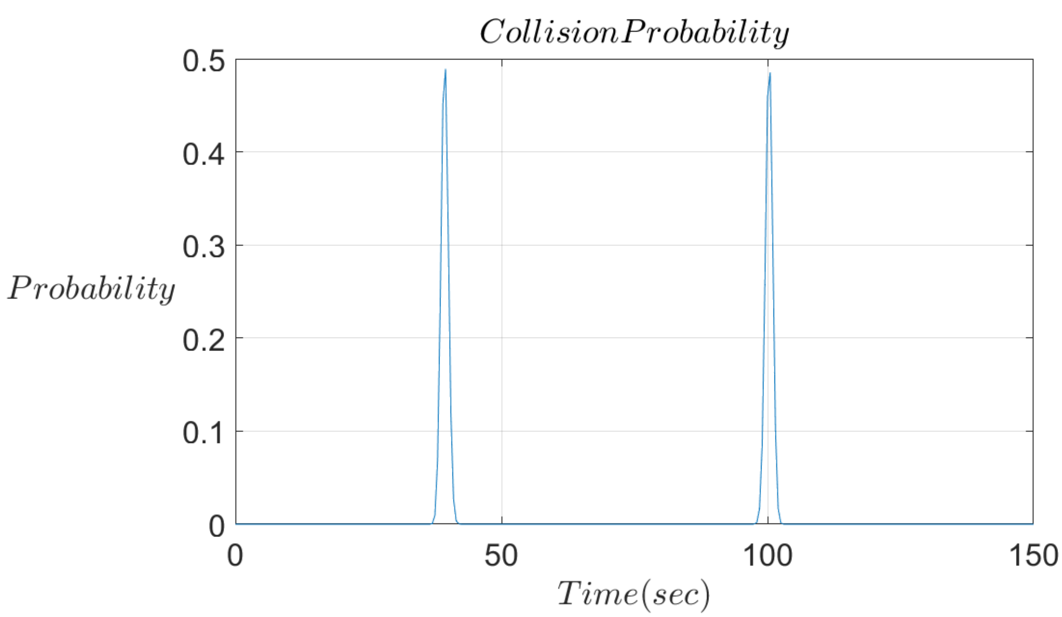


Figure 6: Problem 3)c)

We see (through the position plot) the paths of each aircraft. This could be expanded to show how close they are at each time step, but through analysis we see that there are two points where the aircraft are considerably closer than at other times. This is, of course, where the paths cross on the plot. At these points, the probability of a collision increases to almost 0.5 at around t 40 seconds and t 100 seconds. This is explained by the aforementioned phenomena of the aircraft approaching each other. The probability doesn't reach 100% because there is an uncertainty associated with each aircraft's position.

.

.

.

Advanced Question 7

Plot the predicted horizontal path of the aircraft in question 2b and show the uncertainty and simulated errors. That is, plot the predicted position state mean in the ξ vs. ν plane, but only show the location every 30 steps. Add the 2σ Gaussian covariance ellipses for the position states to the plot, with each ellipse centered on the predicted position state mean at the corresponding time step. On the same plot for each ellipse, also show the resulting position estimates of a sufficiently large number of Monte Carlo simulation runs of the aircraft motion sampled from the uncertain initial conditions to verify that your resulting ellipses are in fact correct. Explain how/why your samples provide verification.

To plot an ellipse, we first consider the equation for a non-rotated ellipse at (0,0), with radii r_x and r_y.

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 = 1$$

$$x(t) = r_x cos(t), \qquad y(t) = r_y sin(t), \qquad for \ \ t = 0 : 2\pi$$

This ellipse can be rotated by θ...

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} r_x cos(t) \\ r_y sin(t) \end{bmatrix}$$

Now, we can represent a covariance matrix as an ellipse by considering the square root of each eigenvalue of the covariance matrix as the radii of the ellipse, or $r_x = \sqrt{\lambda_1}$ and $r_y = \sqrt{\lambda_2}$. The square root is taken because λ^2 corrensponds to σ^2, and we are simply interested in σ. We then multiply $\sqrt{\sigma}$ by a given mahalanobis distance, s. In this case, $s\sigma = 2\sigma$.

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} v_{1,x} & v_{2,x} \\ v_{1,y} & v_{2,y} \end{bmatrix} \begin{bmatrix} s * \sqrt{\lambda_1} cos(t) \\ s * \sqrt{\lambda_2} sin(t) \end{bmatrix}$$

Now, perform position estimate via Monte Carlo simulations of 200 samples for each position. The mahalanobis distance s, tells us how many σ's a point is away from the mean. It is a sum of squares which corresponds to a chi squared distribution. I initially thought I could take random samples from a chi squared distribution, but I think I would also have to know the corresponding θ value, which I don't.

For the Monte Carlo simulations, simply use MATLAB's mvndrnd function to generate ξ and η samples. These samples provide verification that the ellipses are correct because, through calculation or observation, we can see that approximately 68 % of the samples fall within the 2σ bounds of the ellipse.

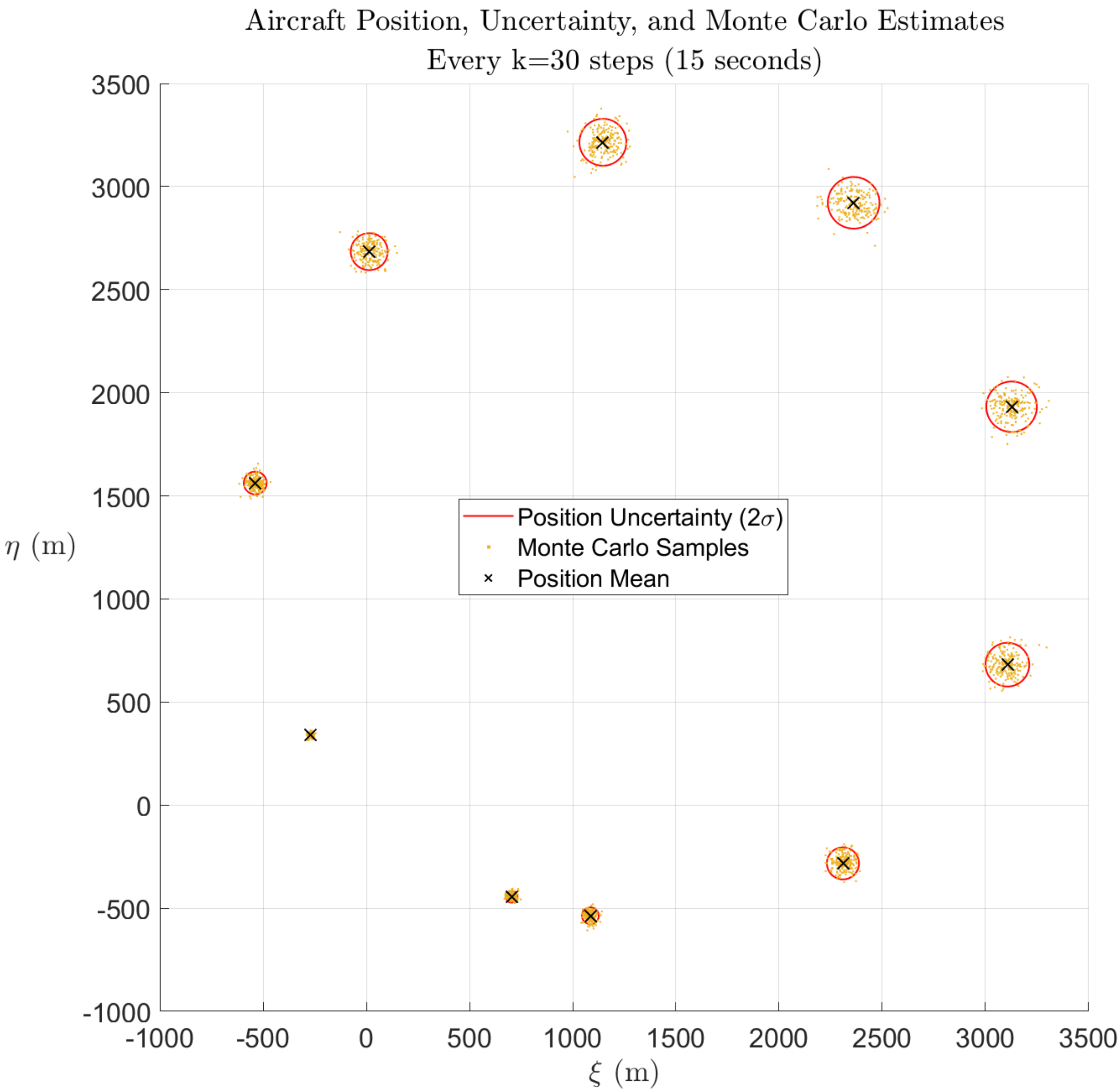


Figure 7: Problem 3)c)

MATLAB Code

```
%% ASEN 5044 HW6
% Ross Fischer
% 10/24/2023
format long g
clc, clear all, close all
set(0,'defaulttextinterpreter','latex');
set(0,'defaultaxesfontsize',16);

%% 2b
```

```
%Find the DT LTI representation for this system, using...
dt=0.5; %sec
om=0.045;

A = [0 1 0 0;
    0 0 0 -om;
    0 0 0 1;
    0 om 0 0];
F = [1 (sin(om*dt))/om 0 -(1-cos(om*dt))/om;
    0 cos(om*dt) 0 -sin(om*dt);
    0 (1-cos(om*dt))/om 1 sin(om*dt)/om;
    0 sin(om*dt) 0 cos(om*dt)];
J = [1 0 0 0; 0 0 1 0];

xmu(:,1)=[0; 85*cos(pi/4); 0; -85*sin(pi/4)];
P(:,:,1) = diag([10, 2, 10, 2]);

k=300; %steps
times=(0:k)*dt;
for N=2:k+1
    Fk = F^(N-1);
    xmu(:,N) = Fk*xmu(:,1);
    P(:,:,N) = Fk*P(:,:,1)*Fk';
end

for plot1=1
    figure;
    tiledlayout(4,2);
    %Tile 1
    tile1=nexttile;
    hold on
    plot(times, xmu(1,:),'b-')
    plot(times, xmu(1,:) + 2*sqrt(reshape(P(1,1,:),1,[])), 'b--')
    plot(times, xmu(1,:) - 2*sqrt(reshape(P(1,1,:),1,[])), 'b--')
    title('$\mu \; and \; \pm 2 \sigma \; vs \; t$', 'interpreter','latex')
    ylabel('$\mu_1 = \xi (m)$', 'Rotation',0,'interpreter','latex')
    grid on

    tile2 = nexttile;
    plot(times,2*sqrt(reshape(P(1,1,:),1,[])), 'b--')
    title('$ 2 \sigma \; vs \; t $', 'interpreter','latex')
    ylabel('$ 2 \sigma_{\xi} (m)$', 'Rotation',0, 'interpreter', 'latex')
    grid on

    tile3 = nexttile;
    hold on
    plot(times, xmu(2,:),'r-')
    plot(times, xmu(2,:) + 2*sqrt(reshape(P(2,2,:),1,[])), 'r--')
    plot(times, xmu(2,:) - 2*sqrt(reshape(P(2,2,:),1,[])), 'r--')
    ylabel('$\mu_2 = \dot{\xi} (\frac{m}{s})$', 'Rotation',0,'interpreter','latex')
```

```
    grid on

    tile4 = nexttile;
    plot(times, 2*sqrt(reshape(P(2,2,:),1,[])), 'r--')
    ylabel('$2 \sigma_{\dot{\xi}} (\frac{m}{s}) $', 'Rotation',0,'interpreter','latex')
    grid on

    tile5 = nexttile;
    hold on
    plot(times,xmu(3,:),'g-')
    plot(times, xmu(3,:) + 2*sqrt(reshape(P(3,3,:),1,[])), 'g--')
    plot(times, xmu(3,:) - 2*sqrt(reshape(P(3,3,:),1,[])), 'g--')
    ylabel('$\mu_3 = \eta (m)$', 'Rotation',0,'interpreter','latex')
    grid on

    tile6 = nexttile;
    plot(times, 2*sqrt(reshape(P(3,3,:),1,[])), 'g--')
    ylabel('$2 \sigma_{\eta} (m)$', 'Rotation',0,'interpreter','latex')
    grid on

    tile7 = nexttile;
    hold on
    plot(times,xmu(4,:),'k-')
    plot(times, xmu(4,:) + 2*sqrt(reshape(P(4,4,:),1,[])), 'k--')
    plot(times, xmu(4,:) - 2*sqrt(reshape(P(4,4,:),1,[])), 'k--')
    ylabel('$\mu_4 = \dot{\eta} (\frac{m}{s})$', 'Rotation',0,'interpreter','latex')
    xlabel('$time (sec)$', 'interpreter', 'latex')
    grid on

    tile8 = nexttile;
    xlabel('time (sec)')
    plot(times, 2*sqrt(reshape(P(4,4,:),1,[])), 'k--')
    ylabel('$2 \sigma_{\dot{\eta}} (\frac{m}{s})$', 'Rotation',0,'interpreter','latex')
    xlabel('$time (sec)$', 'interpreter', 'latex')
    grid on
end

%% AQ7
%every 30 points
% xmu30 = xmu(:,1:30:end);
% P30 = J*P(:,:,1:30:end)*J';
times30 = times(1:30:end);

figure;

hold on
for N=1:length(xmu(1,30:30:end))
    xmu30(:,N) = J*xmu(:,N*30);
    P30(:,:,N) = J*P(:,:,N*30)*J';
```

```
    [eigvec, eigval] = eig(P30(:,:,N));

    ellipse(:,:,N) = eigvec * [2 * sqrt(eigval(1,1)) * cos(linspace(0,2*pi)); 2 * sqrt(eigval(
    plotting(1) = plot(ellipse(1,:,N) + xmu30(1,N) , ellipse(2,:,N) + xmu30(2,N), 'r-', 'linew

    MCsamples = mvnrnd(xmu30(:,N),P30(:,:,N),200)';
    plotting(2) = scatter(MCsamples(1,:), MCsamples(2,:), 10, [0.9290 0.6940 0.1250], ".")
end

plotting(3) = scatter(xmu30(1,:), xmu30(2,:), 100, 'kx', 'LineWidth',1)
legend(plotting,{'Position Uncertainty (2\sigma)','Monte Carlo Samples','Position Mean'})
title('Aircraft Position, Uncertainty, and Monte Carlo Estimates')
subtitle('Every k=30 steps (15 seconds)')
xlabel('$\xi$')
ylabel('$\eta$','rotation',0)
axis square
grid on


cdf_all=[cumsum(px)*dx; xval];
[C,ia,ic]=unique(cdf_all(1,:)); %ia will provide indices of unique values so we can later inte
cdf_unique=round(cdf_all(:,ia)*100000)/100000; %multiply, round, divide to give same results b
N=[100];

%% part c

oma = 0.045; %rad/s
omb = -0.045; %rad/s
Fa =  [1 (sin(oma*dt))/oma 0 -(1-cos(oma*dt))/oma;
    0 cos(oma*dt) 0 -sin(oma*dt);
    0 (1-cos(oma*dt))/oma 1 sin(oma*dt)/oma;
    0 sin(oma*dt) 0 cos(oma*dt)];
Fb =  [1 (sin(omb*dt))/omb 0 -(1-cos(omb*dt))/omb;
    0 cos(omb*dt) 0 -sin(omb*dt);
    0 (1-cos(omb*dt))/omb 1 sin(omb*dt)/omb;
    0 sin(omb*dt) 0 cos(omb*dt)];
xiR = 100;
etaR = 100;
lowerBound = [-xiR -etaR];
upperBound = [xiR etaR];

mua(:,1) = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
mub(:,1) = [3200; 85*cos(pi/4); 3200; -85*sin(pi/4)] ;
murc(:,1) = J*(mua(:,1) - mub(:,1));
Pa(:,:,1) = diag([10; 4; 10; 4]);
Pb(:,:,1) = diag([11; 3.5; 11; 3.5]);
Prc(:,:,1) = J * (Pa(:,:,1) + Pb(:,:,1)) * J';
Pcoll(1) = mvncdf(lowerBound,upperBound,murc(:,1)',Prc(:,:,1));

k=300; %steps
```

```
times=(0:k)*dt;
for N=2:k+1
    Fak = Fa^(N-1);
    Fbk = Fb^(N-1);
    mua(:,N) = Fak * mua(:,1);
    mub(:,N) = Fbk * mub(:,1);
    murc(:,N) = J*(mua(:,N) - mub(:,N));
    Pa(:,:,N) = Fak*Pa(:,:,1)*Fak';
    Pb(:,:,N) = Fbk*Pb(:,:,1)*Fbk';
    Prc(:,:,N) = J * (Pa(:,:,N) + Pb(:,:,N)) * J';
    Pcoll(N) = mvncdf(lowerBound,upperBound,murc(:,N)',Prc(:,:,N));
end

for plot3=1
    figure;
    tiledlayout(2,1);
    %Tile 1
    tile1=nexttile;
    hold on
    plot(mua(1,:), mua(3,:),'b-')
    plot(mub(1,:), mub(3,:),'r-')
    scatter(mua(1,1), mua(3,1), 'b')
    scatter(mub(1,1), mub(3,1), 'r')
    title('$Aircraft \; Position$')
    xlabel('$\xi$')
    ylabel('$\eta$', 'Rotation',0)
    legend('Aircraft A', 'Aircraft B', 'A, Start Location', 'B, Start location')
    axis equal
    grid on

    tile2 = nexttile;
    plot(times,Pcoll)
    title('$Collision Probability$', 'interpreter','latex')
    xlabel('$Time (sec)$')
    ylabel('$Probability$', 'Rotation',0)
    grid on
end
```