Table of Contents

Clean

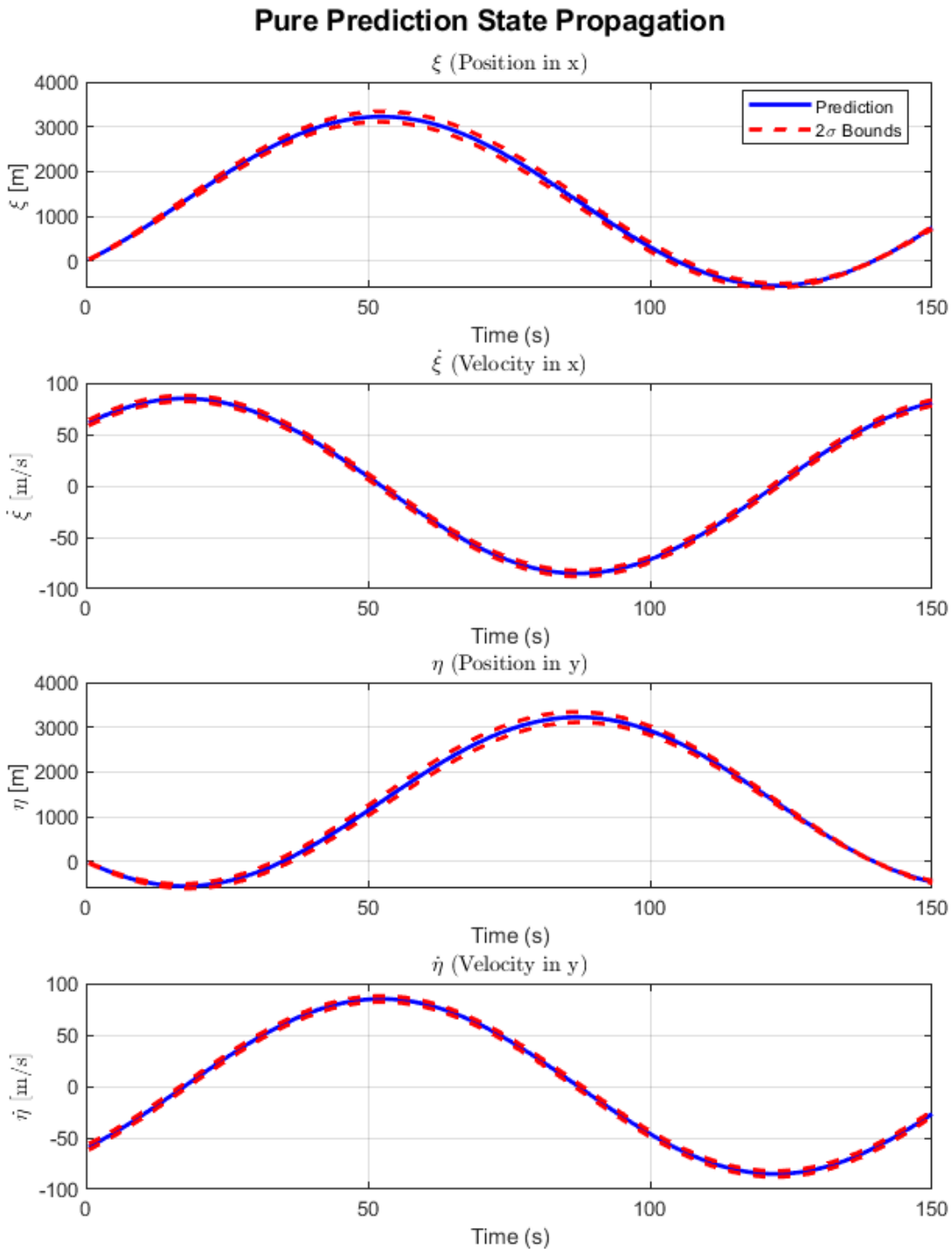
```
close all; clear; clc;
```

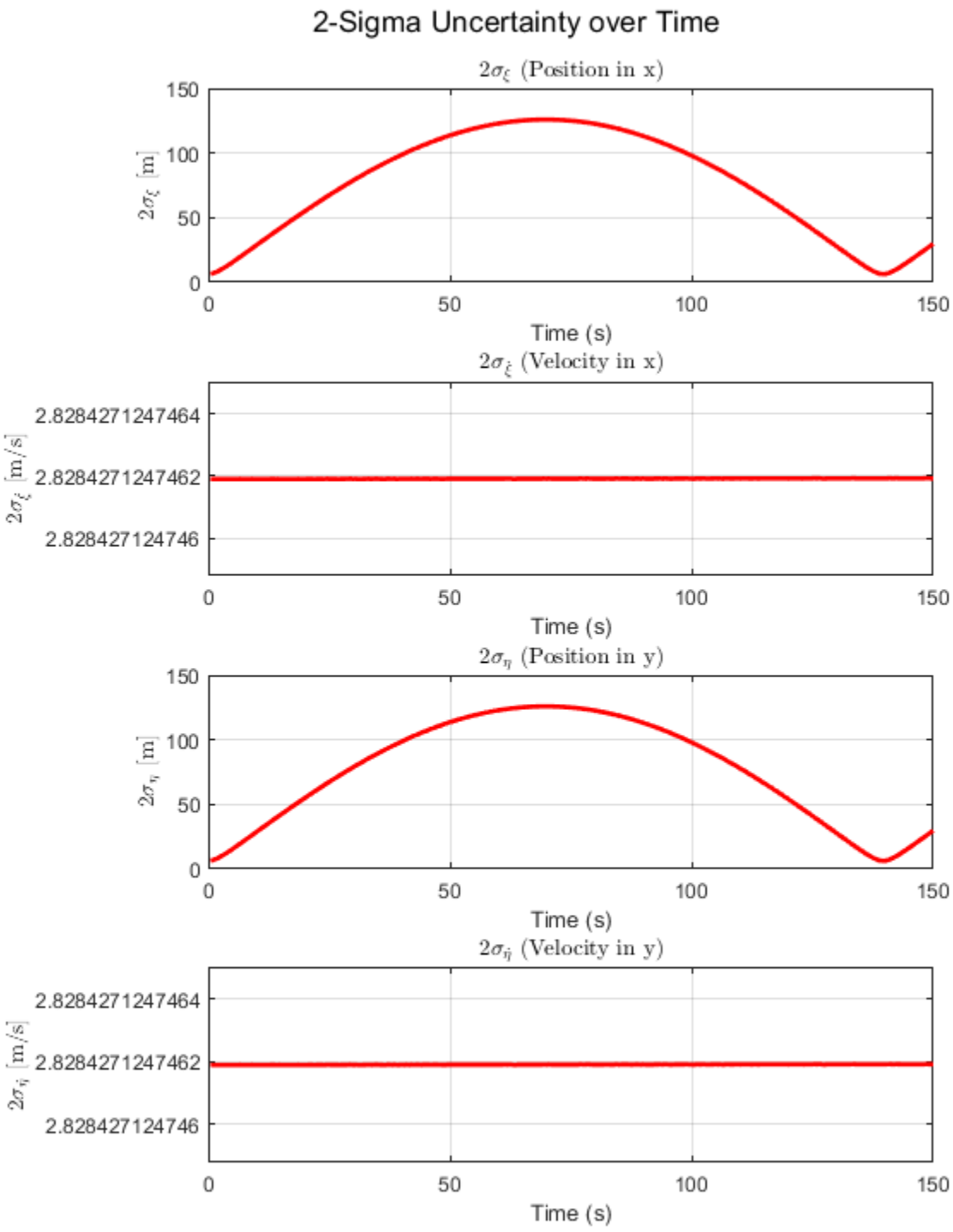
Problem 1

```
mu_0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
P_0 = [10, 0, 0, 0; 0, 2, 0 ,0; 0, 0, 10, 0; 0, 0, 0, 2];
sigma = 0.045;
dt = 0.5;
F = [1, sin(sigma*dt)/sigma, 0, -(1-cos(sigma*dt))/sigma;
    0, cos(sigma*dt), 0, -sin(sigma*dt);
    0, (1-cos(sigma*dt))/sigma, 1, sin(sigma*dt)/sigma;
    0, sin(sigma*dt), 0, cos(sigma*dt)];

% 300 step simulation
iterations = 1:300;
mu_mat = zeros(length(iterations), 4);
unc_mat = zeros(length(iterations), 4);

for i = iterations
    % Propogate average
    mu_N = F^i * mu_0;
    mu_mat(i,:) = mu_N';

    % Propogate covariance
    P_N = F^i * P_0 * (F^i)';
    unc_mat(i,:) = [sqrt(P_N(1,1)), sqrt(P_N(2,2)), sqrt(P_N(3,3)),
sqrt(P_N(4,4))];
end

time_vec = iterations .* 0.5;


%%%% +- 2sigma values
figure();

% Set line properties for clarity
mainLineColor = 'b';
boundLineColor = 'r';
lineWidth = 2;

% Main title for the figure
sgtitle('Pure Prediction State Propagation', 'FontSize', 14, 'FontWeight',
```

```
'bold');

% xi
subplot(4,1,1);
plot(time_vec, mu_mat(:,1), 'LineWidth', lineWidth, 'Color', mainLineColor);
hold on;
grid on;
plot(time_vec, mu_mat(:,1) + 2 .* unc_mat(:,1), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
plot(time_vec, mu_mat(:,1) - 2 .* unc_mat(:,1), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('\xi [m]');
title('$\xi$ (Position in x)', 'Interpreter','latex');
legend('Prediction', '2\sigma Bounds', 'Location', 'best');

% xi dot
subplot(4,1,2);
plot(time_vec, mu_mat(:,2), 'LineWidth', lineWidth, 'Color', mainLineColor);
hold on;
grid on;
plot(time_vec, mu_mat(:,2) + 2 .* unc_mat(:,2), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
plot(time_vec, mu_mat(:,2) - 2 .* unc_mat(:,2), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('$\dot{\xi}$ [m/s]', 'Interpreter', 'latex');
title('$\dot{\xi}$ (Velocity in x)', 'Interpreter','latex');

% eta
subplot(4,1,3);
plot(time_vec, mu_mat(:,3), 'LineWidth', lineWidth, 'Color', mainLineColor);
hold on;
grid on;
plot(time_vec, mu_mat(:,3) + 2 .* unc_mat(:,3), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
plot(time_vec, mu_mat(:,3) - 2 .* unc_mat(:,3), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('\eta [m]');
title('$\eta$ (Position in y)', 'Interpreter','latex');

% eta dot
subplot(4,1,4);
plot(time_vec, mu_mat(:,4), 'LineWidth', lineWidth, 'Color', mainLineColor);
hold on;
grid on;
plot(time_vec, mu_mat(:,4) + 2 .* unc_mat(:,4), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
plot(time_vec, mu_mat(:,4) - 2 .* unc_mat(:,4), 'Color', boundLineColor,
'LineStyle', '--', 'LineWidth', lineWidth);
xlabel('Time (s)');
ylabel('$\dot{\eta}$ [m/s]', 'Interpreter', 'latex');
title('$\dot{\eta}$ (Velocity in y)', 'Interpreter','latex');
```

```
% Adjust layout and size
set(gcf, 'Position', [100, 100, 700, 900]);



%%%% Positive 2 sigma bounds
figure();
sgtitle('2-Sigma Uncertainty over Time');

% Set color and line width for consistency
lineColor = 'r';
lineWidth = 2;

% xi
subplot(4,1,1);
plot(time_vec, 2 .* unc_mat(:,1), 'Color', lineColor, 'LineStyle', '-',
'LineWidth', lineWidth);
grid on;
xlabel('Time (s)');
ylabel('$2\sigma_{\xi}$ [m]', 'Interpreter', 'latex');
title('$2\sigma_{\xi}$ (Position in x)', 'Interpreter', 'latex');

% xi dot
subplot(4,1,2);
plot(time_vec, 2 .* unc_mat(:,2), 'Color', lineColor, 'LineStyle', '-',
'LineWidth', lineWidth);
grid on;
xlabel('Time (s)');
ylabel('2$\sigma_{\dot{\xi}}$ [m/s]', 'Interpreter', 'latex');
title('$2\sigma_{\dot{\xi}}$ (Velocity in x)', 'Interpreter', 'latex');

% eta
subplot(4,1,3);
plot(time_vec, 2 .* unc_mat(:,3), 'Color', lineColor, 'LineStyle', '-',
'LineWidth', lineWidth);
grid on;
xlabel('Time (s)');
ylabel('$2\sigma_\eta$ [m]', 'Interpreter', 'latex');
title('$2\sigma_{\eta}$ (Position in y)', 'Interpreter', 'latex');

% eta dot
subplot(4,1,4);
plot(time_vec, 2 .* unc_mat(:,4), 'Color', lineColor, 'LineStyle', '-',
'LineWidth', lineWidth);
grid on;
xlabel('Time (s)');
ylabel('$2\sigma_{\dot{\eta}}$ [m/s]', 'Interpreter', 'latex');  % Fixed
label with latex interpreter
title('$2\sigma_{\dot{\eta}}$ (Velocity in y)', 'Interpreter', 'latex');

% Main title and adjust layout
set(gcf, 'Position', [100, 100, 600, 800]);  % Adjust figure size
```

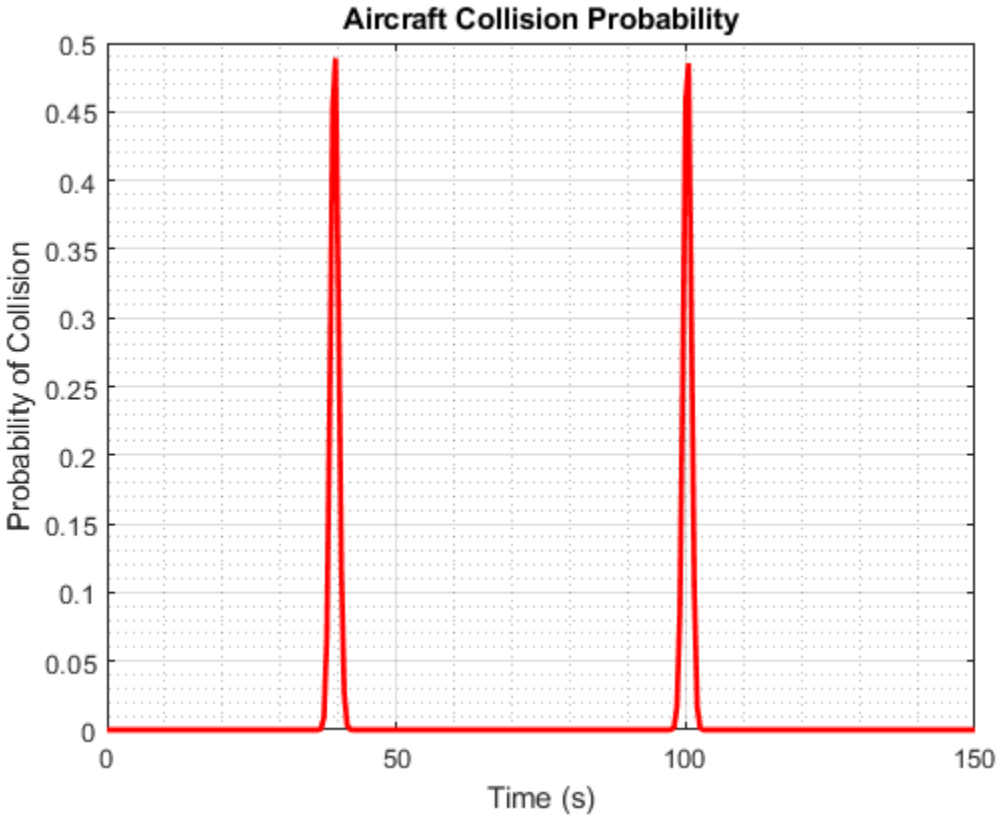

Pure Prediction State Propagation
ξ (Position in x)
ξ [m]
Prediction
2σ Bounds
Time (s)
ξ̇ (Velocity in x)
ξ̇ [m/s]
Time (s)
η (Position in y)
η [m]
Time (s)
η̇ (Velocity in y)
η̇ [m/s]
Time (s)


2-Sigma Uncertainty over Time
$2\sigma_\xi$ (Position in x)
$2\sigma_\xi$ [m]
150
100
50
0
0
50
100
150
Time (s)
$2\sigma_{\dot{\xi}}$ (Velocity in x)
$2\sigma_{\dot{\xi}}$ [m/s]
2.8284271247464
2.8284271247462
2.828427124746
0
50
100
150
Time (s)
$2\sigma_\eta$ (Position in y)
$2\sigma_\eta$ [m]
150
100
50
0
0
50
100
150
Time (s)
$2\sigma_{\dot{\eta}}$ (Velocity in y)
$2\sigma_{\dot{\eta}}$ [m/s]
2.8284271247464
2.8284271247462
2.828427124746
0
50
100
150
Time (s)

Problem 3

```
close all; clear; clc;

% Initial conditions
mua_0 = [0; 85*cos(pi/4); 0; -85*sin(pi/4)];
Pa_0 = diag([10; 4; 10; 4]);
mub_0 = [3200; 85*cos(pi/4); 3200; -85*sin(pi/4)];
Pb_0 = diag([11; 3.5; 11; 3.5]);

% Given parameters
dt = 0.5;
sigma_a = 0.045;
sigma_b = -0.045;

% Collision parameters
xi_R = 100;
eta_R = 100;
xl = [-xi_R; -eta_R];
xu = [xi_R; eta_R];

% F matrices
Fa = [1, sin(sigma_a*dt)/sigma_a, 0, -(1-cos(sigma_a*dt))/sigma_a;
    0, cos(sigma_a*dt), 0, -sin(sigma_a*dt);
    0, (1-cos(sigma_a*dt))/sigma_a, 1, sin(sigma_a*dt)/sigma_a;
    0, sin(sigma_a*dt), 0, cos(sigma_a*dt)];

Fb = [1, sin(sigma_b*dt)/sigma_b, 0, -(1-cos(sigma_b*dt))/sigma_b;
    0, cos(sigma_b*dt), 0, -sin(sigma_b*dt);
    0, (1-cos(sigma_b*dt))/sigma_b, 1, sin(sigma_b*dt)/sigma_b;
    0, sin(sigma_b*dt), 0, cos(sigma_b*dt)];


% Time vector for simulation
times = 0:dt:150;

% Marginalize matrix
marg = [1, 0, 0, 0; 0, 0, 1, 0];

% Simulate
collision_prob = zeros(length(times), 1);
for i = 0:length(times)-1
    time = times(i+1);
    mu_rc = marg*(Fa^i * mua_0 - Fb^i * mub_0);
    Prc = marg*(Fa^i * Pa_0 * (Fa^i)' + Fb^i * Pb_0 * (Fb^i)')*marg';
    collision_prob(i+1) = mvncdf(xl, xu, mu_rc, Prc);
end



% Plotting
figure();
plot(times, collision_prob, 'linewidth', 2, 'color', 'r')
```

```
grid on
grid minor

xlabel('Time (s)')
ylabel('Probability of Collision')
title('Aircraft Collision Probability')
```



Published with MATLAB® R2024a