Table of Contents

Programming Homework 4 - (Shane Billingsley)

```
x_trim = [0; 0; -1800; 0; 0.02780; 0; 20.99; 0; 0.5837; 0; 0; 0];
u_trim = [0.1068; 0; 0; 0.2439];
```

Task 1

```
% Estimate the roll angle and calculate the integration time here
phi_r = atan2(21^2,(200*9.81));
t_max = (200/21)*(pi/2);
disp("Roll Angle required is "+string(phi_r)+" rad, or
"+string(rad2deg(phi_r))+" degrees.");
disp("Time required is "+string(t_max)+" seconds.");
```

```
Roll Angle required is 0.2211 rad, or 12.6679 degrees.
Time required is 14.96 seconds.
```

Task 2

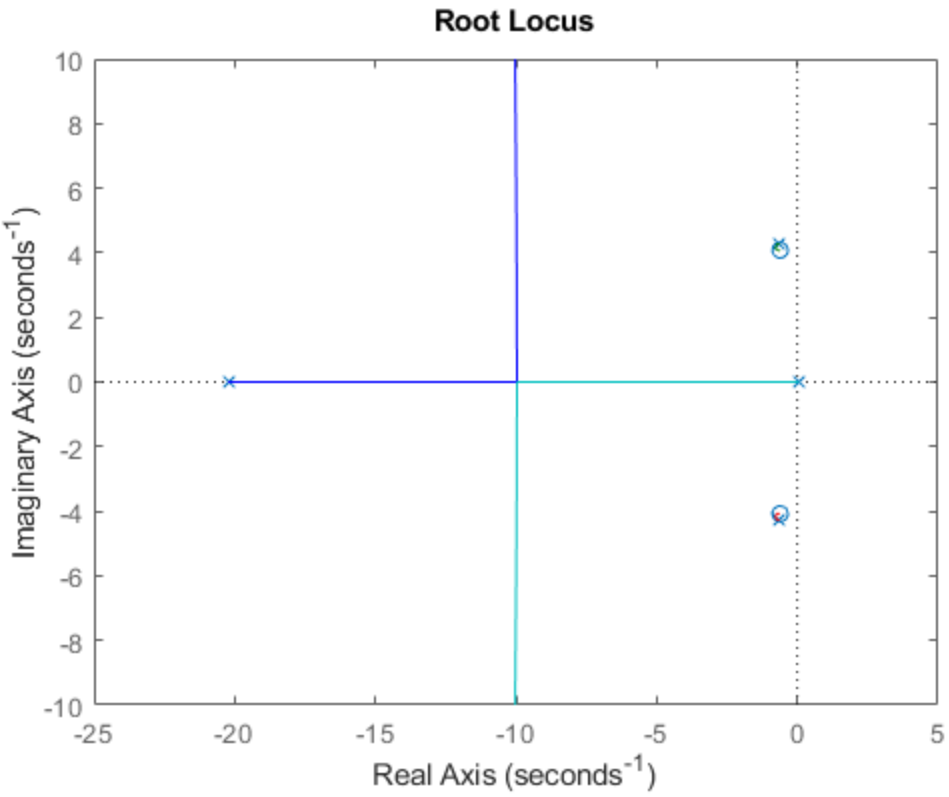
```
[Alat, Blat] = estimateLateralSS(@aircraftDynamics, x_trim, u_trim,
ttwistor);
Lp = Alat(2,2); L_delta_a = Blat(2,1);
tau_r = 1/Lp;
ka = -(1/9)*(Lp/L_delta_a);
tau_cl = 1/(Lp - (ka*L_delta_a));
```

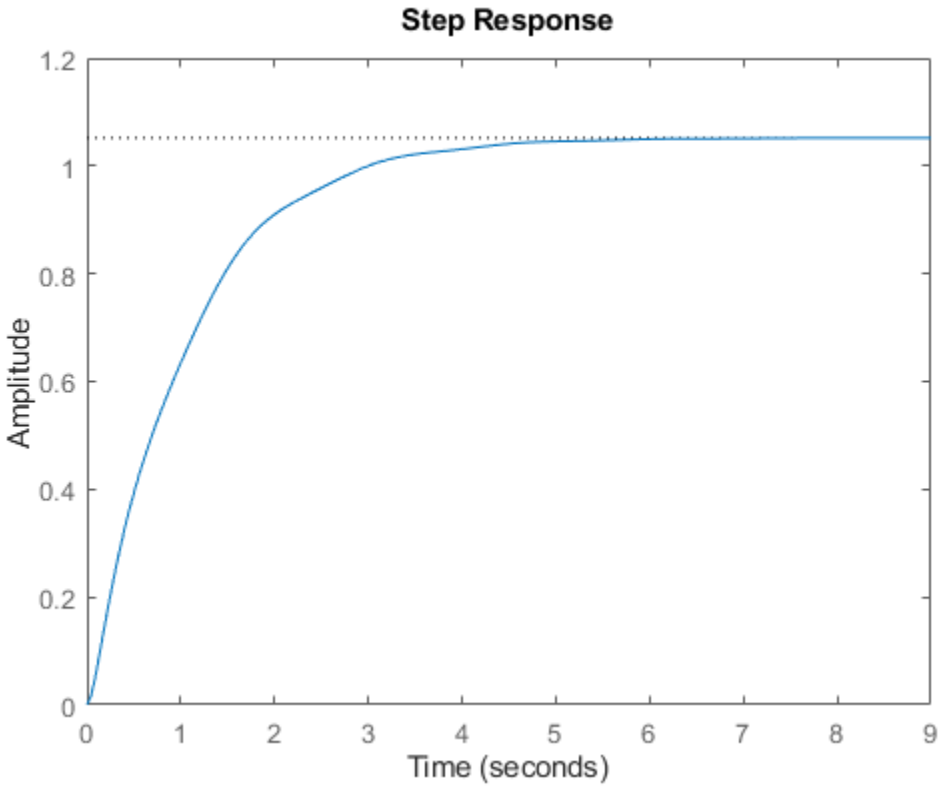
Task 3

```
aileron_sys = ss(Alat, Blat(:,1), eye(4), []);
p_fb_sys = feedback(ka*aileron_sys, 1, 1, 2);
phi_sys = p_fb_sys(4,1); % input roll rate; output roll angle

figure(1);
rlocus(phi_sys);

figure(2);
kp = 11
```

```
step(feedback(kp*phi_sys,1,1,1)); % Modify this to get the step response of
the feedback system
```

```
kp =

    11
```

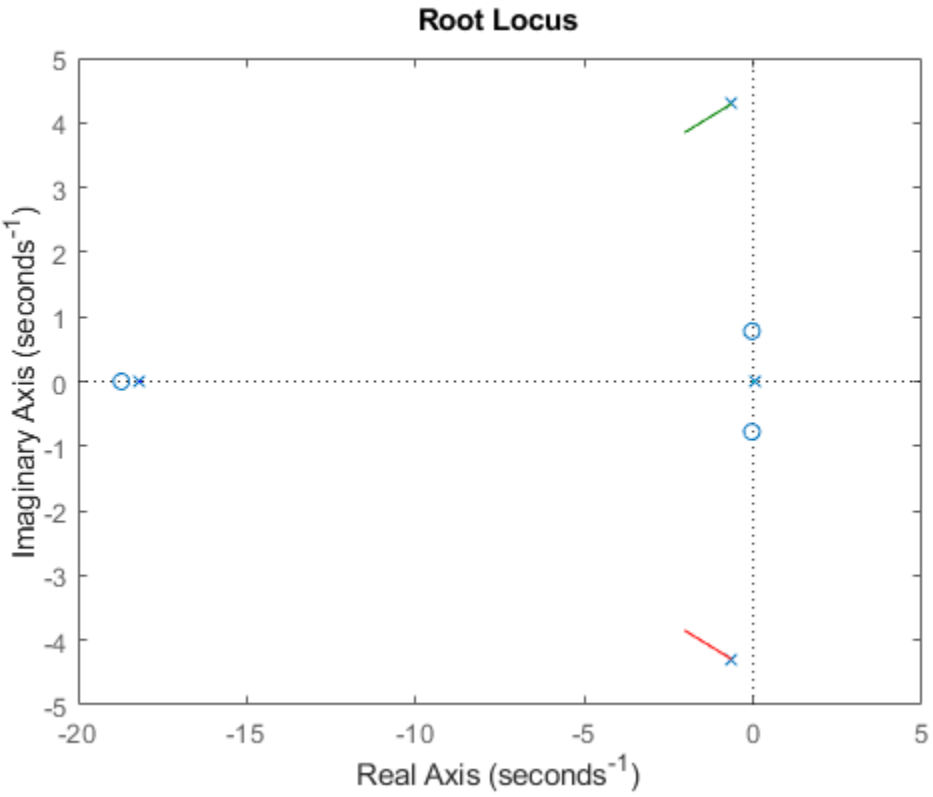




Task 4

```
r_sys = ss(Alat, Blat(:,2), [0 0 1 0], []);
figure(3);
rlocus(r_sys,[0,-15.7]);
kr = -7.6
```

```
kr =

   -7.6000
```

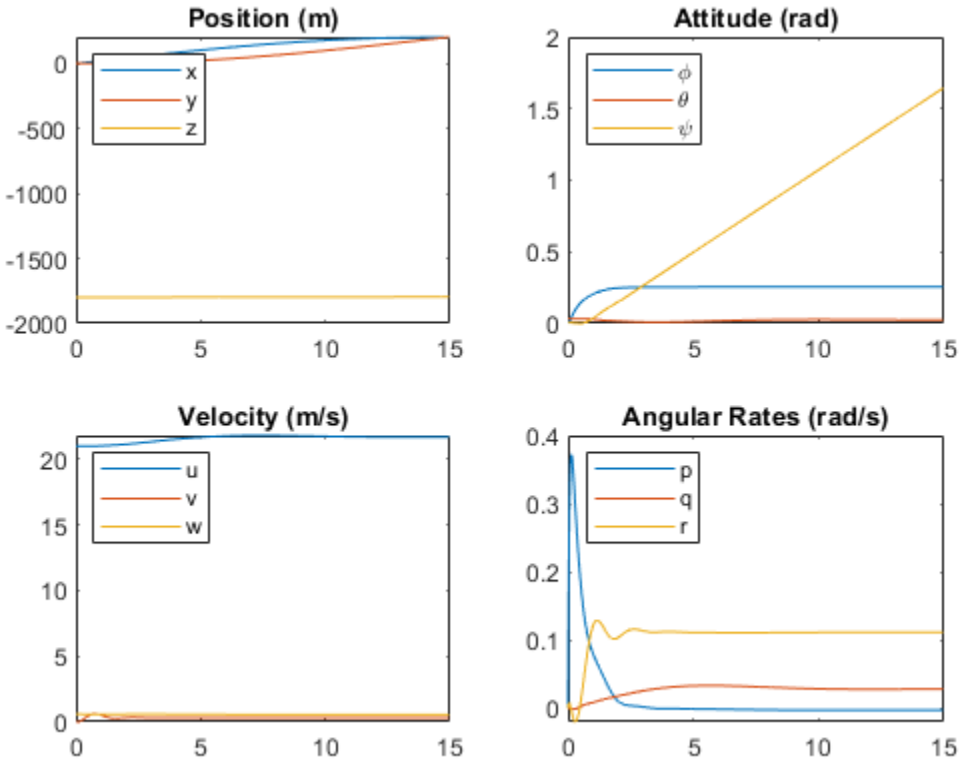


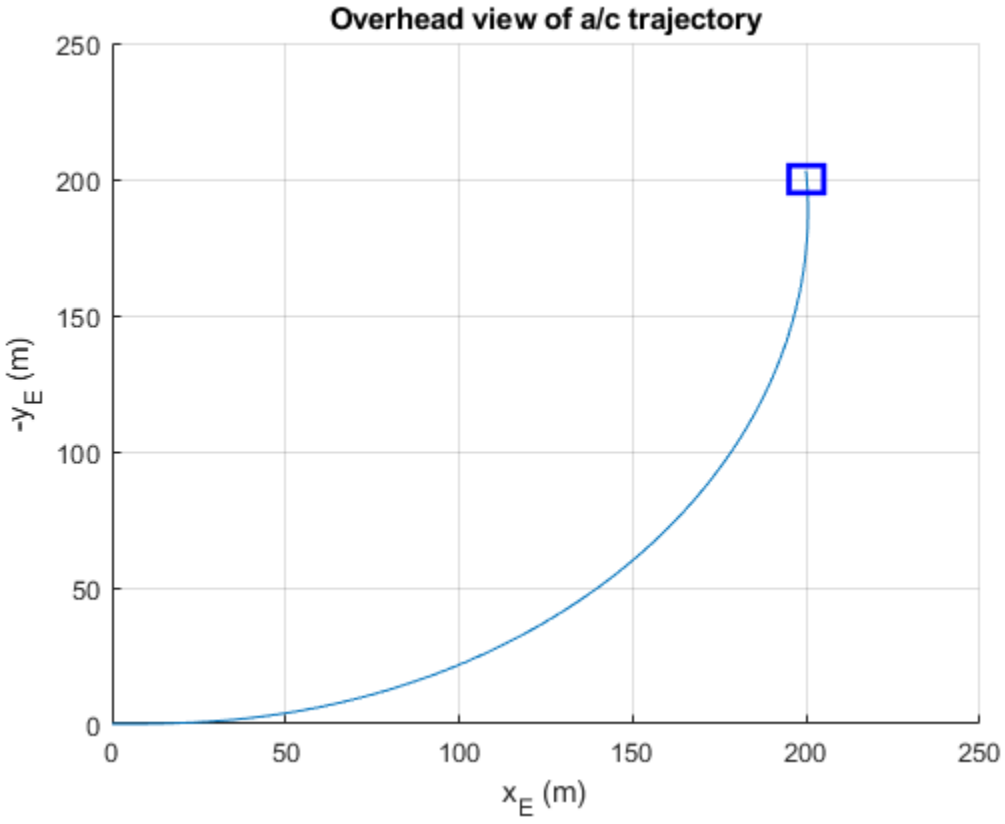
Task 5

Implement the control law in the controls.m file. No additional deliverables are needed for this task.

Task 6

```
[T, X] = ode45(@(t, x) aircraftDynamics(x, controls(t, x), ttwistor), [0,
t_max], x_trim);

figure(4);
plotStateHistory(T, X);

% Plot the trajectory in the x-y plane
x = X(:,1);
y = X(:,2);
figure(5); hold on;
plot(x, y);
rectangle('Position', [195, 195, 10, 10], 'EdgeColor', 'b', 'LineWidth', 2);
title('Overhead view of a/c trajectory');
xlabel('x_E (m)');
ylabel('-y_E (m)');
grid on;
```


Position (m)
x
y
z
0
-500
-1000
-1500
-2000
0
5
10
15
Attitude (rad)
ϕ
θ
ψ
2
1.5
1
0.5
0
Velocity (m/s)
u
v
w
20
15
10
5
0
Angular Rates (rad/s)
p
q
r
0.4
0.3
0.2
0.1


Overhead view of a/c trajectory
250
200
150
100
50
0
$-y_E$ (m)
x_E (m)

Task 7

```
evaluate(@controls, 'shane.billingsley@colorado.edu') % edit this with your
actual email address
```

```
Results saved to submission.json (score=100)
Please submit this to gradescope!

ans =

    'submission.json'
```

Task 8 (OPTIONAL)

```
% % add wind to the simulation like this to increase your score on the
leaderboard:
% wind = 1
% [json, T, X] = evaluate(@controls,
'REPLACE_WITH_YOUR_GRADESCOPE_EMAIL@colorado.edu', wind,
'wind_submission.json')
% % you can plot T and X to see the trajectory with wind
```

Published with MATLAB® R2024a