Programming Homework 3 - (Shane Billingsley)

Table of Contents

Task 1

(Nothing additional is needed for this task besides passing the tests.)

```
runtests('testLonAeroForcesAndMoments.m')
```

```
Running testLonAeroForcesAndMoments
...
Done testLonAeroForcesAndMoments
__________


ans =

  1×3 TestResult array with properties:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details

Totals:
   3 Passed, 0 Failed, 0 Incomplete.
   0.098573 seconds testing time.
```

Task 2

(Nothing additional is needed for this task besides passing the tests.)

```
runtests('testAircraftDynamics.m')
```

```
Running testAircraftDynamics
...
```

```
Done testAircraftDynamics
__________

ans =

  1×3 TestResult array with properties:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details

Totals:
   3 Passed, 0 Failed, 0 Incomplete.
   0.046312 seconds testing time.
```

Task 3

```
evaluate('shane.billingsley@colorado.edu') % Change the email to your own
```

```
Running evaluate
...
Done evaluate
__________

score =

   100

Results saved to submission.json
Please submit this to gradescope!

ans =

    'submission.json'
```

Task 4

```
x_trim = [0; 0; -1800; 0; 0.02780; 0; 20.99; 0; 0.5837; 0; 0; 0];
u_trim = [0.1079; 0; 0; 0.3182];
A_lon = estimateAlon(@aircraftDynamics, x_trim, u_trim, ttwistor);

% Calculate eigenvectors and indicate which corresponds to phugoid and short
period
[V1,D1] = eig(A_lon); %get eigenvalues/eigenvectors
values = diag(D1);
%find indices for short period and phugoid poles
```

```
sp_val_ind = find(real(values) == min(real(values)));
ph_val_ind = find(real(values) == max(real(values)));
%sort values and vectors into short period and phugoid
sp_values = values(sp_val_ind); ph_values = values(ph_val_ind);
sp_vectors = V1(:,sp_val_ind);  ph_vectors = V1(:,ph_val_ind);
disp("Short Period Eigenvectors");
disp(sp_vectors);
disp("Phugoid Eigenvectors")
disp(ph_vectors);
```

```
A_lon =

   -0.1211    0.5145   -0.5750   -9.8062
   -0.7291   -7.3539   19.7457   -0.2727
    0.0687   -2.4693   -3.8325         0
         0         0    1.0000         0

Short Period Eigenvectors
   0.0055 + 0.0279i   0.0055 - 0.0279i
  -0.9420 + 0.0000i  -0.9420 + 0.0000i
  -0.0837 - 0.3216i  -0.0837 + 0.3216i
  -0.0221 + 0.0307i  -0.0221 - 0.0307i

Phugoid Eigenvectors
   0.9978 + 0.0000i   0.9978 + 0.0000i
  -0.0186 - 0.0029i  -0.0186 + 0.0029i
   0.0299 - 0.0023i   0.0299 + 0.0023i
  -0.0097 - 0.0547i  -0.0097 + 0.0547i
```

Task 5

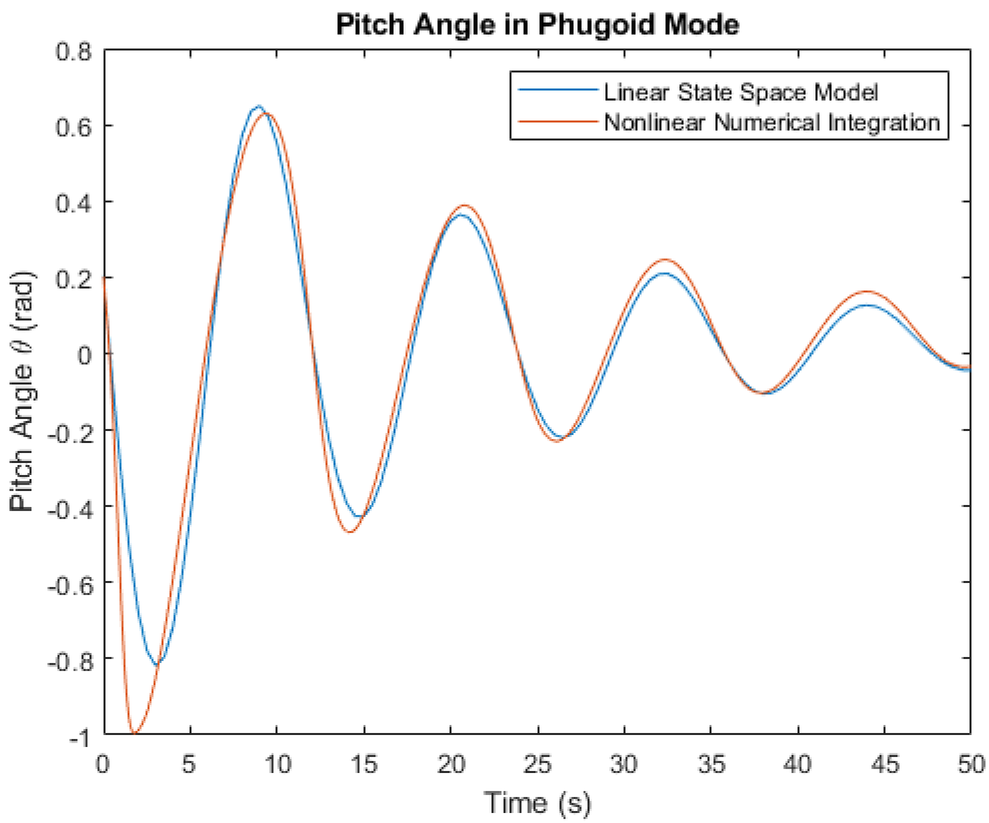
```
%to excite only phugoid mode we use a phugoid eigenvector, take the real part
% and normalize it such that delta_theta = 10 deg
factor = deg2rad(10);   %10 degrees in radians
vector = real(ph_vectors(:,1));   %pull out a vector
d_theta = vector(4);    %pull out delta_theta (according to estimateAlon)
x = factor/d_theta;     %find a normalization factor
vector = x.*vector;     %normalize vector such that delta_theta = 10 degrees

ss_lon = ss(A_lon, zeros(4,1), [0 0 0 1], [0]); % no control input, output
d_theta
dx_ph = vector; % put in normalized vector
[linear_theta, linear_time] = initial(ss_lon, dx_ph, 50);

vector_nonlinear = [0;0;0;0;vector(4);0;vector(1);0;vector(2);0;vector(3);0];
x_ph = x_trim + vector_nonlinear; %Add trim state to linearized deviations
vector
[nonlinear_time, nonlinear_x] = ode45(@(t, x) aircraftDynamics(x, u_trim,
ttwistor), [0, 50], x_ph);

figure(1)
```

```
plot(linear_time, linear_theta + x_trim(5), nonlinear_time, nonlinear_x(:,5))
title("Pitch Angle in Phugoid Mode");
xlabel("Time (s)");
ylabel("Pitch Angle \theta (rad)");
legend("Linear State Space Model","Nonlinear Numerical Integration");
```



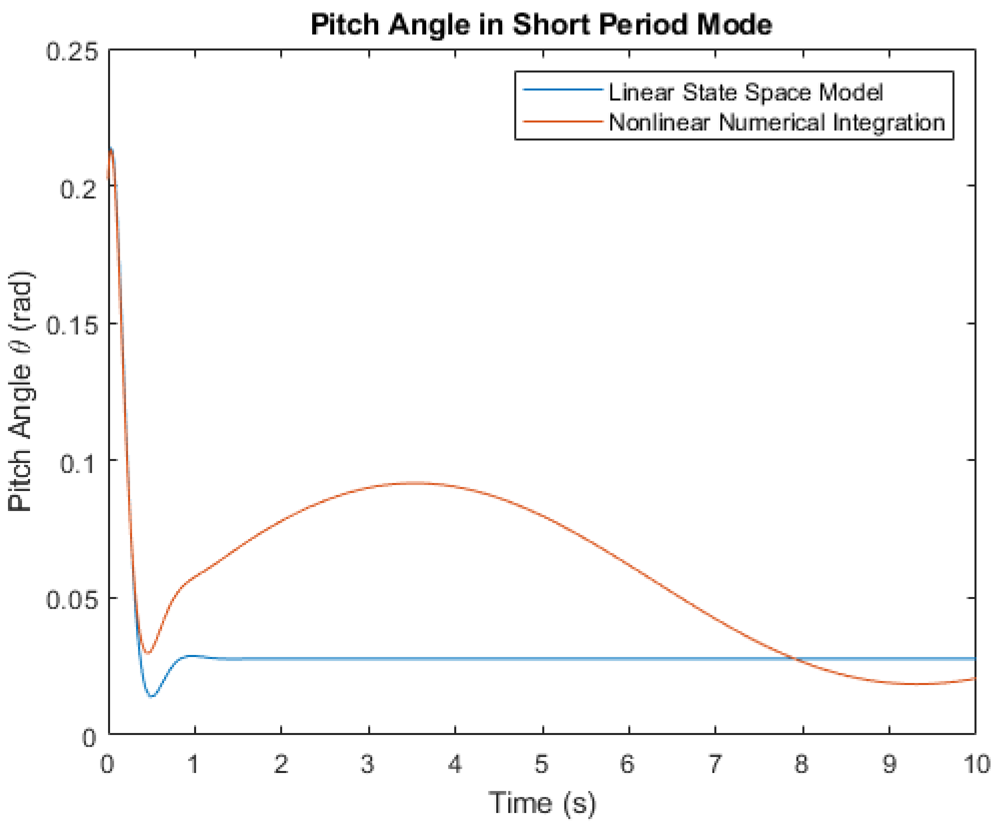
Task 6

```
%to excite only short mode we use a short period eigenvector, take the real
part
% and normalize it such that delta_theta = 10 deg
factor = deg2rad(10);   %10 degrees in radians
vector = real(sp_vectors(:,1));   %pull out a vector
d_theta = vector(4);     %pull out delta_theta (according to estimateAlon)
x = factor/d_theta;      %find a normalization factor
vector = x.*vector;      %normalize vector such that delta_theta = 10 degrees

dx_sp = vector; % Put in normalized vector
[linear_theta, linear_time] = initial(ss_lon, dx_sp, 10);

vector_nonlinear = [0;0;0;0;vector(4);0;vector(1);0;vector(2);0;vector(3);0];
x_sp = x_trim + vector_nonlinear; %Add trim state to linearized deviations
vector
[nonlinear_time, nonlinear_x] = ode45(@(t, x) aircraftDynamics(x, u_trim,
ttwistor), [0, 10], x_sp);
```

```
figure(2)
plot(linear_time, linear_theta + x_trim(5), nonlinear_time, nonlinear_x(:,5))
title("Pitch Angle in Short Period Mode");
xlabel("Time (s)");
ylabel("Pitch Angle \theta (rad)");
legend("Linear State Space Model","Nonlinear Numerical Integration");
```



Task 7

The short period mode showed a greater deviation between the linear and nonlinear models. This is because it is impossible to excite only one mode in the nonlinear model. The short period mode damps out very quickly, on the order of 1 second, so it has very little effect on the nonlinear phugoid mode. However, when we model the short period mode using nonlinear dynamics, we see a short period oscillation that quickly damps out, but then transitions to phugoid oscillations in pitch angle. However, the magnitudes of these oscillations are small compared to the pure phugoid oscillations, because the short period eigenvectors are smaller in magnitude. We are using a much smaller deviation in u when we excite the short period mode, so the resulting phugoid oscillations in the nonlinear model are small.

Published with MATLAB® R2023b