Project 5

Ben Grimsley

Due 5/12/2016

Longitudinal Model with states and outputs [V q α θ] and input [δ e]

```
syslon =

  a =
                x1          x2          x3          x4
   x1     -0.01313      -4.515     -0.1884      -32.18
   x2   -0.0001625     -0.8462      0.9951           0
   x3   -3.773e-11      -3.327     -0.4353           0
   x4            0           0           1           0

  b =
            u1
   x1    3.892
   x2  -0.1289
   x3   -9.199
   x4        0

  c =
       x1  x2  x3  x4
   y1   1   0   0   0
   y2   0   1   0   0
   y3   0   0   1   0
   y4   0   0   0   1

  d =
       u1
   y1   0
   y2   0
   y3   0
   y4   0
```

Lateral/Directional Model with states and outputs [β ϕ p r] and inputs [δa δr]

```
syslat =

  a =
              x1         x2         x3         x4
   x1    -0.2317    0.06331    -0.9956    0.05123
   x2     -29.49     -3.017    0.02006          0
   x3      6.235   -0.02742    -0.4169          0
   x4          0          1     0.0631          0

  b =
              u1         u2
   x1   0.005215    0.03096
   x2     -36.49      8.109
   x3    -0.4916     -2.827
   x4          0          0

  c =
       x1  x2  x3  x4
   y1   1   0   0   0
   y2   0   1   0   0
   y3   0   0   1   0
   y4   0   0   0   1

  d =
       u1  u2
   y1   0   0
   y2   0   0
   y3   0   0
   y4   0   0
```

Short Period Eigenvalues, Damping Ratio and Natural Frequencies

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-5.72e-03 + 6.85e-02i	8.31e-02	6.88e-02	1.75e+02
-5.72e-03 - 6.85e-02i	8.31e-02	6.88e-02	1.75e+02

Long Period Eigenvalues, Damping Ratio and Natural Frequencies

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-6.42e-01 + 1.81e+00i	3.35e-01	1.92e+00	1.56e+00
-6.42e-01 - 1.81e+00i	3.35e-01	1.92e+00	1.56e+00

The Aircraft is longitudinally stable since all the Longitudinal roots lie in the left half plane (negative real components)

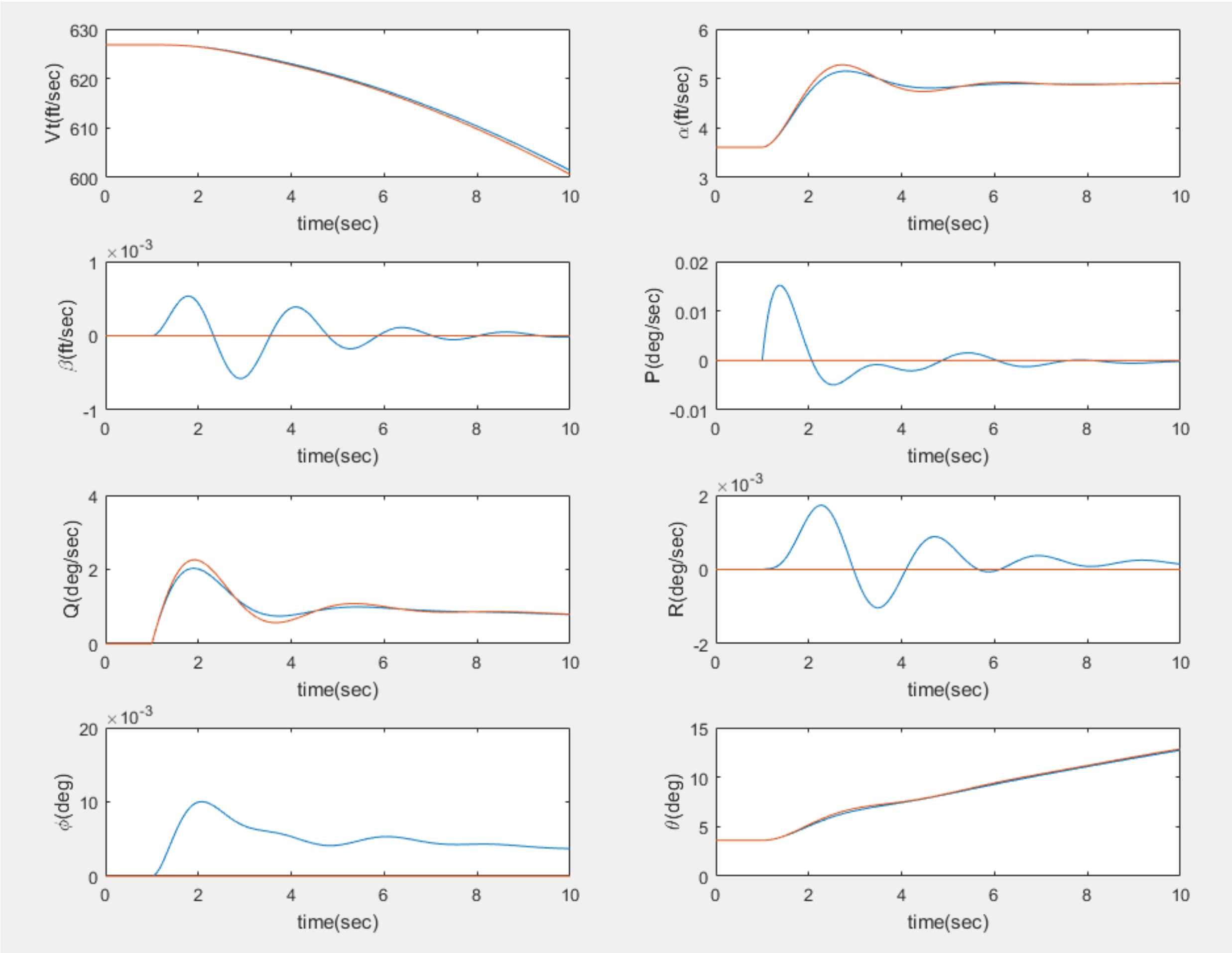
pitch-rate due to elevator transfer function

```
             -9.199 s^3 - 7.477 s^2 - 0.08776 s
 3:  -------------------------------------------------
     s^4 + 1.295 s^3 + 3.696 s^2 + 0.04811 s + 0.01739
```

angle-of-attack due to elevator transfer function

```
       -0.1289 s^3 - 9.213 s^2 - 0.1215 s - 0.04809
 2:  -------------------------------------------------
     s^4 + 1.295 s^3 + 3.696 s^2 + 0.04811 s + 0.01739
```

Linear vs Nonlinear Elevator step response of -0.5 Deg:



Dutch Roll Eigenvalues, Damping Ratio and Natural Frequencies

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-4.08e-01 + 2.75e+00i	1.47e-01	2.78e+00	2.45e+00
-4.08e-01 - 2.75e+00i	1.47e-01	2.78e+00	2.45e+00

Roll Mode Eigenvalues, Damping Ratio and Natural Frequencies and Time Constant

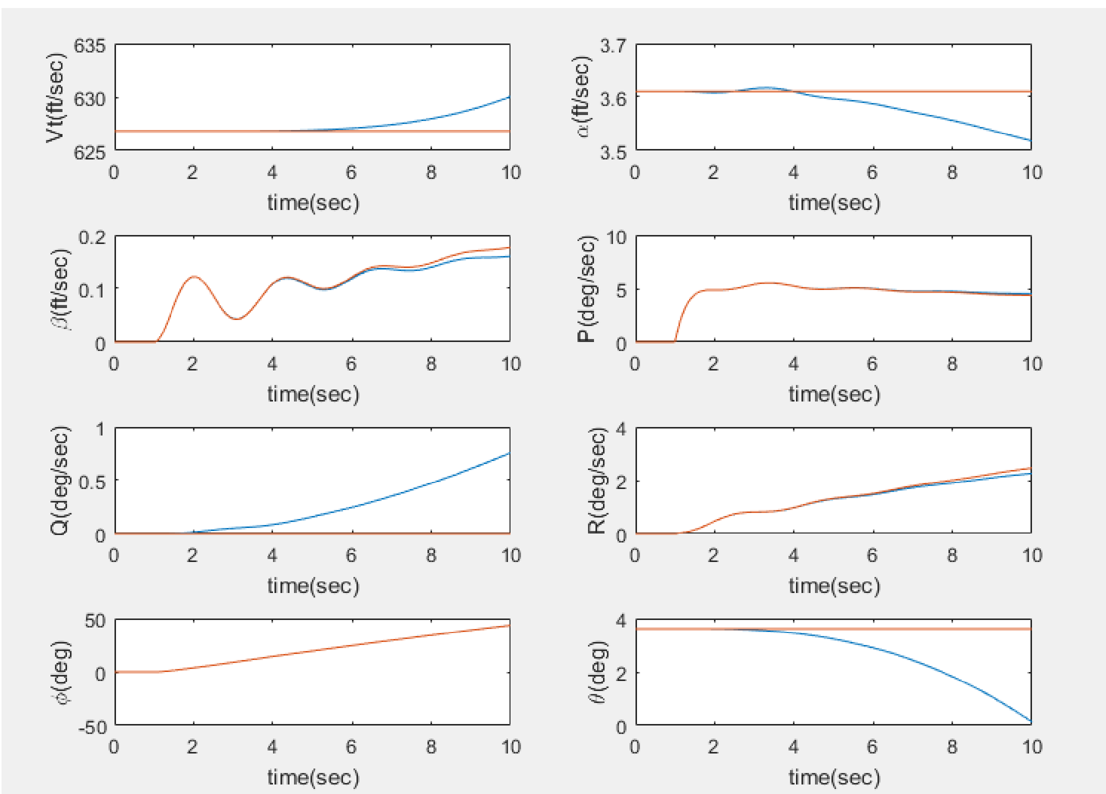
Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-2.82e+00	1.00e+00	2.82e+00	3.54e-01

Spiral Mode Eigenvalues, Damping Ratio and Natural Frequencies and Time Constant

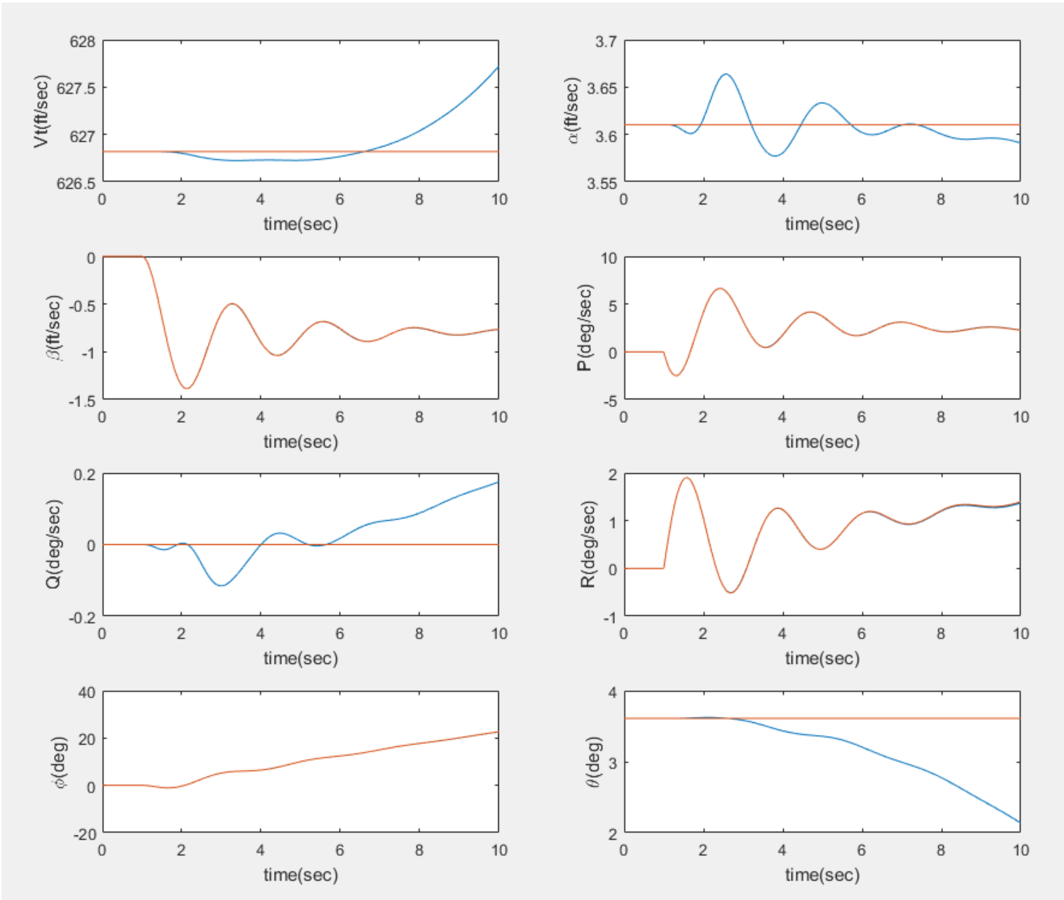
Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-2.57e-02	1.00e+00	2.57e-02	3.90e+01

The aircraft is laterally stable since all lateral roots are in the left half plane (negative real components)

Linear vs Nonlinear Aileron step response of -0.5 Deg:



Linear vs Nonlinear Rudder step response of -2.0 Deg:



MATLAB CODE

```
% Flight Dynamics Project 5
%Ben Grimsley
% 10 May 16
clc,close all,clear all
load_system('FD_5')

r2d = 180/pi;
d2r = pi/180;

Vt0 = 626.81863;
a0 = 3.6102915*d2r;
b0 = 0;
p0 = 0;
q0 = 0;
r0 = 0;
phi0 = 0;
theta0 = 3.6102915*d2r;
psi0=0;

Ic = [Vt0 a0 b0 p0 q0 r0 phi0 theta0 psi0];

% Inertia
Ixx = 8890.63;
Iyy = 71973.5;
Izz = 77141.1;
Ixz = 181.119;
Ixy = 0;
Iyz = 0;
I = [Ixx -Ixz -Ixz; -Ixy Iyy -Iyz; -Ixz -Iyz Izz];
Iinv = inv(I);

% Constants


S = 300;
cbar = 11.32;
b = 30;
mass = 762.8447;
T = 3146.482666;
rho = 0.0014962376;
g = 32.17561865;



CLo = 0.004608463;
CLa = 0.0794655*r2d;
CLq = 0.0508476*r2d;
CLadot = 0;
CLde = 0.0121988*r2d;
CLdf = 0.0144389*r2d;

CDo = 0.01192128;
```

```
CDa = 0.00550063*r2d;
CDq = 0.00315057*r2d;
CDadot = 0;
CDde = -0.000587647*r2d;
CDdf = 0.00136385*r2d;

CYo = 0;
CYb = -0.0219309*r2d;
CYp = 0.00133787*r2d;
CYr = 0.0094053*r2d;
CYda = 0.00049355*r2d;
CYdr = 0.00293048*r2d;

Clo = 0;
Clb = -0.00173748*r2d;
Clp = -0.00739342*r2d;
Clr = 0.0000699792*r2d;
Clda = -0.00213984*r2d;
Cldr = 0.000479021*r2d;

Cmo = -0.02092347;
Cma = -0.0041873*r2d;
Cmq = -0.060661*r2d;
Cmadot = -0.05*r2d;
Cmde = -0.0115767*r2d;
Cmdf = 0.000580220*r2d;

Cno = 0;
Cnb = 0.00320831*r2d;
Cnp = -0.000432575*r2d;
Cnr = -0.00886783*r2d;
Cnda = -0.000206591*r2d;
Cndr = -0.00144865*r2d;
%% Inputs
%% Trim:
% dei = -3.03804303*d2r;
% def = -3.03804303*d2r;
% dai = 0;
% daf = 0;
% dri = 0;
% drf = 0;
% df = 1.5*d2r;
%% Elevator:
dei = -3.03804303*d2r;
def = -3.53804303*d2r;
dai = 0;
daf = 0;
dri = 0;
drf = 0;
df = 1.5*d2r;
%% Aileron:
% dei = -3.03804303*d2r;
% def = -3.03804303*d2r;
% dai = 0;
% daf = -0.5*d2r;
% dri = 0;
```

```
% drf = 0;
% df = 1.5*d2r;
%% Rudder:
% dei = -3.03804303*d2r;
% def = -3.03804303*d2r;
% dai = 0;
% daf = 0;
% dri = 0;
% drf = -2*d2r;
% df = 1.5*d2r;


sim('FD_5')



[A,B,C,D] = linmod('FD_5');

Alon = A(1:4,1:4); Blon = B(1:4,1); Clon  = C(1:4,1:4); Dlon = D(1:4,1);
Alat = A(5:8,5:8); Blat = B(5:8,2:3); Clat = C(5:8,5:8); Dlat = D(5:8,2:3);
sim('FD_52')

figure
subplot (4,2,1);
plot(time,Vt,time,Vtlong+Vt0);
xlabel('time(sec)');ylabel('Vt(ft/sec)')
subplot (4,2,2);
plot(time,alpha,time,alphalong+a0*r2d);
xlabel('time(sec)');ylabel('\alpha(ft/sec)')
subplot (4,2,3);
plot(time,beta,time,Betalat+b0*r2d);
xlabel('time(sec)');ylabel('\beta(ft/sec)')
subplot (4,2,4);
plot(time,p,time,Plat);
xlabel('time(sec)');ylabel('P(deg/sec)')
subplot (4,2,5);
plot(time,q,time,Qlong);
xlabel('time(sec)');ylabel('Q(deg/sec)')
subplot (4,2,6);
plot(time,r,time,Rlat);
xlabel('time(sec)');ylabel('R(deg/sec)')
subplot (4,2,7);
plot(time,phi,time,philat);
xlabel('time(sec)');ylabel('\phi(deg)')
subplot (4,2,8);
plot(time,theta,time,Thetalong+theta0*r2d);
xlabel('time(sec)');ylabel('\theta(deg)')
% subplot (3,3,9);
% plot(time,psi);
% xlabel('time');ylabel('\psi(deg)')

size(ss(Alon,Blon,Clon,Dlon))
syslon = ss(Alon,Blon,Clon,Dlon)
syslat = ss(Alat,Blat,Clat,Dlat)
eig(Alon)
```

```
damp(syslon)
longtf = tf(syslon)
```