Table of Contents

X15 Overview

The X-15 was a hypersonic research airplane First flown in 1960. The rocket powered airplane could fly up to Mach 6 at altitudes of up to 300,000 ft above sea level. Prior to launch, the X-15 would be mounted under a B-52 aircraft and carried up to an altitude of approximately 45,000 ft. Once at altitude, the X-15 would be launched with an initial speed of Mach 0.8, quickly accelerating to full speed and a higher altitude. Following the powered phase of fight, the vehicle would enter a glide for eventual landing and recovery. Further details about the X-15 can be found at http://z.umn.edu/x15fs

```
% Author: Jyot Buch
% Ph.D. Student in Aerospace Engineering and Mechanics
% Ref: NASA CR-2144 by Heffley and Jewell technical report.
clear;clc;close all;
```

Trim Flight Condition

Trim corresponds to equilibrium condition in which we are using small perturbation theory to obtain LTI model of an aircraft from nonlinear equations of motion.

```
% Mach Number
M0 = 2.0;

% Altitude
h0 = 60000; % [ft]

% Pitch Angle (Steady Level Flight but given in the table)
theta0 = deg2rad(4); % [rad]

% Trim Speed
u0 = 1936*cos(theta0); %[ft/s]
```

```
% Acceleration due to gravity on planet Earth
g = 32.17405; %[ft/s^2]
```

Aircraft Parameters

Aerodynamic, stability, and mass properties data provided in NASA CR-2144 by Heffley and Jewell [Starting Page 114 in PDF]

```
% Physical Parameters
W        = 15560;    % [lb]
Ix       = 3650;     % [slugs-ft^2]
Iy       = 80000;    % [slugs-ft^2]
Iz       = 82000;    % [slugs-ft^2]
Ixz      = 590;      % [slugs-ft^2]
S        = 200;      % [ft^2]
b        = 22.36;    % [ft]
cbar     = 10.27;    % [ft]
Xcg      = 0.22*cbar;% [ft]
Q        = 424;      % [PSF]
Qc       = 703;      % [PSF]
alpha    = 4;        % [deg]
Lxp      = 18.8;     % [ft]
Lzp      = -2.2;     % [ft]
m        = W;        % [lb]

% LONGITUDINAL PARAMETERS (Page: 131) (Notations Page: 330)
XuS      = -0.00871; %[1/sec]
ZuS      = -0.0117;  %[1/sec]
MuS      = 0.000471; %[1/(sec-ft)]
Xw       = -0.0190;  %[1/sec]
Zw       = -0.311;   %[1/sec]
Mw       = -0.00673; %[1/(sec-ft)]
Zwdot    = 0;        %[1/sec^2]
Zq       = 0;        %[1/sec]
Mwdot    = 0;        %[1/(sec-ft)]
Mq       = -0.182;   %[1/sec]
Xde      = 6.24;
Zde      = -89.2;
Mde      = -9.8;

% LATERAL-DIRECTIONAL PARAMETERS (Page: 135) (Notations Page: 331)
Yv       = -0.127;   %[1/sec]
Yb       = -246.0;
Yp       = 0;
Yr       = 0;
LbS      = -2.36;    %[1/sec^2]
NbS      = 11.1;     %[1/sec^2]
LpS      = -1.02;    %[1/sec]
NpS      = -0.00735; %[1/sec]
LrS      = 0.103;    %[1/sec]
NrS      = -0.186;   %[1/sec]
Yda      = -0.00498; %[1/sec]
LdaS     = 28.7;     %[1/sec^2]
```

```
NdaS    = 0.993;    %[1/sec^2]
YdrS    = 0.0426;
LdrS    = 5.38;
NdrS    = -6.9;

% Anonymous Functions for star to regular conversion and vice-versa
factor = 1 - Ixz^2/(Ix*Iz);
S2R = @(x) x*factor;

% Required values
Xu = S2R(XuS);
Zu = S2R(ZuS);
Mu = S2R(MuS);
Ydr = S2R(YdrS);
```

Linearized Longitudinal Equations of Motion

```
Alon = [...
    Xu          Xw          0           -g*cos(theta0);
    Zu          Zw          u0          -g*sin(theta0);
    Mu+Mwdot*Zu Mw+Mwdot*Zw Mq+Mwdot*u0 0;
    0           0           1           0];
Blon = [...
    Xde;
    Zde;
    Mde+Mwdot*Zde;
    0];
Glon = ss(Alon,Blon,eye(4),0)
Glon.InputName = {'\Delta\delta_e'};
Glon.StateName = {'\Deltau','\Deltaw','\Deltaq','\Delta\theta'};
Glon.OutputName = {'\Deltau','\Deltaw','\Deltaq','\Delta\theta'};
```

```
Glon =

  A =
                x1          x2          x3          x4
   x1     -0.0087      -0.019           0       -32.1
   x2    -0.01169      -0.311        1931      -2.244
   x3   0.0004705    -0.00673      -0.182           0
   x4           0           0           1           0

  B =
            u1
   x1     6.24
   x2    -89.2
   x3     -9.8
   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

Continuous-time state-space model.
```

Linearized Lateral Equations of Motion

```
Alat = [...
    Yb/u0               Yp/u0               -(1-Yr/u0)
 g*cos(theta0)/u0;
    LbS+(Ixz/Ix)*NbS  LpS+(Ixz/Ix)*NpS  LrS+(Ixz/Ix)*NrS  0;
    NbS+(Ixz/Iz)*LbS  NpS+(Ixz/Iz)*LpS  NrS+(Ixz/Iz)*LrS  0;
    0                 1                 0                 0];
Blat = [...
    0                   Ydr/u0;
    LdaS+(Ixz/Ix)*NdaS  LdrS+(Ixz/Ix)*NdrS;
    NdaS+(Ixz/Iz)*LdaS  NdrS+(Ixz/Iz)*LdrS;
    0                   0];
Glat = ss(Alat,Blat,eye(4),0)
Glat.InputName = {'\Delta\delta_a','\Delta\delta_r'};
Glat.StateName = {'\Delta\beta','\Deltap','\Deltar','\Delta\phi'};
Glat.OutputName = {'\Delta\beta','\Deltap','\Deltar','\Delta\phi'};

% NOTE: It is assumed that the longitudinal and lateral-directional
% dynamics are fully decoupled.
```

```
Glat =

  A =
             x1        x2        x3        x4
   x1   -0.1274         0        -1   0.01662
   x2   -0.5658    -1.021   0.07293         0
   x3     11.08  -0.01469   -0.1853         0
   x4         0         1         0         0

  B =
              u1         u2
   x1          0  2.203e-05
   x2      28.86      4.265
   x3        1.2     -6.861
   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

Continuous-time state-space model.
```

Longitudinal OpenLoop Eigenvalues

```
fprintf('===========================\n');
fprintf('Longitudinal Eigen Analysis\n');
fprintf('===========================\n');
[Vlon,Dlon] = eig(Alon);Dlon = diag(Dlon);
Eigenvalues = Dlon %#ok<*NOPTS,*NASGU>

% Normalize evecs by theta direction and non-dimensionalize
for ii = 1:4
    Vlon(:,ii) = Vlon(:,ii)./(Vlon(4,ii));
end
Vlon(1,:) = Vlon(1,:)/u0;
Vlon(2,:) = Vlon(2,:)/u0;
Vlon(3,:) = Vlon(3,:)*(cbar/2/u0);
EigenvectorsMagnitude = abs(Vlon)
EigenvectorsPhase = angle(Vlon)

lamsp = Dlon(1:2);
disp('### Short Period Mode:')
Tsp = 2*pi/imag(lamsp(1));
wnsp = sqrt(lamsp(1)*lamsp(2));
zetasp = -(lamsp(1)+lamsp(2))/2/wnsp;
disp(['Period: ' num2str(Tsp) ' seconds'])
disp(['Natural Frequency: ' num2str(wnsp) ' rad/s'])
disp(['Damping Ratio: ' num2str(zetasp)])

lamp = Dlon(3:4);
disp('### Long Period (Phugoid) Mode:')
Tp = 2*pi/imag(lamp(1));
wnp = sqrt(lamp(1)*lamp(2));
zetap = -(lamp(1)+lamp(2))/2/wnp;
disp(['Period: ' num2str(Tp) ' seconds'])
disp(['Natural Frequency :' num2str(wnp) ' rad/s'])
disp(['Damping Ratio: ' num2str(zetap)]);
fprintf(newline);
```

```
===========================
```

```
Longitudinal Eigen Analysis
===========================
Eigenvalues =
  -0.2459 + 3.6045i
  -0.2459 - 3.6045i
  -0.0050 + 0.0227i
  -0.0050 - 0.0227i
EigenvectorsMagnitude =
    0.0099    0.0099    0.7140    0.7140
    1.0021    1.0021    0.0496    0.0496
    0.0096    0.0096    0.0001    0.0001
    1.0000    1.0000    1.0000    1.0000
EigenvectorsPhase =
    1.5513   -1.5513    1.7882   -1.7882
    0.0865   -0.0865    1.7874   -1.7874
    1.6389   -1.6389    1.7873   -1.7873
         0         0         0         0
### Short Period Mode:
Period: 1.7432 seconds
Natural Frequency: 3.6129 rad/s
Damping Ratio: 0.068049
### Long Period (Phugoid) Mode:
Period: 276.4526 seconds
Natural Frequency :0.023271 rad/s
Damping Ratio: 0.21485
```

Lateral OpenLoop Eigenvalues

```
fprintf('======================\n');
fprintf('Lateral Eigen Analysis\n');
fprintf('======================\n');
[Vlat,Dlat] = eig(Alat);Dlat = diag(Dlat);
Eigenvalues = Dlat

% Normalize evecs by phi direction and non-dimensionalize
for ii = 1:4
    Vlat(:,ii) = Vlat(:,ii)./(Vlat(4,ii));
end
Vlon(1,:) = Vlon(1,:)*(b/2/u0);
Vlon(2,:) = Vlon(2,:)/u0;
Vlon(3,:) = Vlon(3,:)/u0;
EigenvectorsMagnitude = abs(Vlat)
EigenvectorsPhase = angle(Vlat)

dutchr = Dlat(1:2);
disp('### Dutch Roll Mode:')
Tp = 2*pi/imag(dutchr(1));
wnd = sqrt(dutchr(1)*dutchr(2));
zetad = -(dutchr(1)+dutchr(2))/2/wnd;
disp(['Period: ' num2str(Tp) ' seconds'])
disp(['Natural Frequency :' num2str(wnd) ' rad/s'])
disp(['Damping Ratio: ' num2str(zetad)]);
```

```
% Plot
figN = 1;
figN = figN+1;figure(figN);clf;
subplot(3,2,[1 2 3 4]);hold on;box on;grid on;
plt1 = plot(real(Dlon(1:2)),imag(Dlon(1:2)),'bs',...

 real(Dlon(3:4)),imag(Dlon(3:4)),'b^','MarkerSize',10,'LineWidth',2);
plt2 =
 plot(real(Dlat(1:2)),imag(Dlat(1:2)),'r*',real(Dlat(3)),imag(Dlat(3)),...

  'ro',real(Dlat(4)),imag(Dlat(4)),'rx','MarkerSize',10,'LineWidth',2);
title('Longitudinal and Lateral OpenLoop Eigenvalues','FontSize',14);
xlabel('Real(\lambda)','FontSize',14);
ylabel('Imag(\lambda)','FontSize',14);

% Put a Ractangle
r = rectangle('Position',[-0.1 -0.5 0.20
 1],'LineStyle','--','LineWidth',1);

% Put a new axis for zoomed plot
subplot(3,2,5);
box on;hold on;grid on;
plt3 =
 plot(real(Dlon(3:4)),imag(Dlon(3:4)),'b^','MarkerSize',10,'LineWidth',2);
plt4 =
 plot(real(Dlat(4)),imag(Dlat(4)),'rx','MarkerSize',10,'LineWidth',2);
title('Zoomed Plot Near Origin','FontSize',14);ylim([-0.04 0.04]);
xlabel('Real(\lambda)','FontSize',14);
ylabel('Imag(\lambda)','FontSize',14);

% Legend plot
sh=subplot(3,2,6);
p=get(sh,'position');
lh=legend(sh,[plt1; plt2],...
    'Short Period Mode','Long Period (Phugoid) Mode',...
    'Dutch Roll Mode','Roll Subsidence Mode','Spiral
 Mode','FontSize',12);
set(lh,'position',p);
axis(sh,'off');
print(gcf,'-dpdf','-fillpage','EigenValuesOL');

======================
Lateral Eigen Analysis
======================
Eigenvalues =
  -0.1556 + 3.3295i
  -0.1556 - 3.3295i
  -1.0236 + 0.0000i
   0.0010 + 0.0000i
EigenvectorsMagnitude =
   18.6838   18.6838    0.0024    0.0003
    3.3331    3.3331    1.0236    0.0010
   62.2004   62.2004    0.0144    0.0166
```

```
    1.0000    1.0000    1.0000    1.0000
EigenvectorsPhase =
   -0.6144    0.6144    3.1416         0
    1.6175   -1.6175    3.1416         0
   -2.1765    2.1765         0         0
         0         0         0         0
### Dutch Roll Mode:
Period: 1.8871 seconds
Natural Frequency :3.3331 rad/s
Damping Ratio: 0.046695
```

OpenLoop Elevator Step Response

```
fprintf('======================\n');
fprintf('Elevator Step Response\n');
fprintf('======================\n');
figN = figN+1;figure(figN);clf;
opt = stepDataOptions('StepAmplitude',deg2rad(0.5));
step(Glon,opt);grid on;title('OpenLoop Elevator Step Response')
[y,t] = step(Glon,opt);
S = stepinfo(y,t);
StateNames = {'Delta u','Delta w','Delta q','Delta theta'};
for i = 1:numel(S)
    fprintf('Stepinfo Data for Longitudinal State %s:
\n',StateNames{i});
    disp(S(i))
end
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LonStepRespOL');
```

```
======================
Elevator Step Response
======================
Stepinfo Data for Longitudinal State Delta u:
        RiseTime: 50.0514
    SettlingTime: 729.5855
     SettlingMin: 71.5114
     SettlingMax: 145.9100
       Overshoot: 51.9636
      Undershoot: 0
            Peak: 145.9100
        PeakTime: 128.2599
Stepinfo Data for Longitudinal State Delta w:
        RiseTime: 0.1897
    SettlingTime: 569.4075
     SettlingMin: -22.8587
     SettlingMax: -2.5304
       Overshoot: 281.2706
      Undershoot: 0
            Peak: 22.8587
        PeakTime: 0.8696
Stepinfo Data for Longitudinal State Delta q:
```

```
        RiseTime: 5.3282e-05
    SettlingTime: 174.9189
     SettlingMin: -0.0228
     SettlingMax: 0.0155
       Overshoot: 4.1183e+05
      Undershoot: 2.7973e+05
            Peak: 0.0228
        PeakTime: 0.4348
Stepinfo Data for Longitudinal State Delta theta:
        RiseTime: 7.3210
    SettlingTime: 802.3177
     SettlingMin: -0.0693
     SettlingMax: 0.0034
       Overshoot: 231.1629
      Undershoot: 16.1086
            Peak: 0.0693
        PeakTime: 68.6951
```

Response to a Longitudinal Gust

```
fprintf('================================\n');
fprintf('Response to a Longitudinal Gust\n');
fprintf('================================\n');
figN = figN+1;figure(figN);clf;
x0lon = [10;10;deg2rad(5);deg2rad(5)];
initial(Glon,x0lon);[y,t] = initial(Glon,x0lon);Ip = lsiminfo(y,t);
title('OpenLoop Response to a Longitudinal gust');
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LonGustOL');
for i = 1:numel(Ip)
    fprintf('I.C. Response to a Longitudinal gust %s:\n',...
        StateNames{i});
    disp(Ip(i))
end
```

```
================================
Response to a Longitudinal Gust
================================
I.C. Response to a Longitudinal gust Delta u:
    SettlingTime: 796.0854
             Min: -86.8492
         MinTime: 63.0430
             Max: 43.5138
         MaxTime: 201.2158
I.C. Response to a Longitudinal gust Delta w:
    SettlingTime: 383.5749
             Min: -34.4454
         MinTime: 1.2174
             Max: 43.8318
         MaxTime: 0.3478
I.C. Response to a Longitudinal gust Delta q:
    SettlingTime: 16.6815
```

```
           Min: -0.0722
       MinTime: 0.7826
           Max: 0.0873
       MaxTime: 0
I.C. Response to a Longitudinal gust Delta theta:
    SettlingTime: 733.0728
           Min: -0.0425
       MinTime: 141.7380
           Max: 0.1066
       MaxTime: 0.3478
```

Lateral-Directional Response to a Cross-Wind Perturbation

```
fprintf('=========================================================
\n');
fprintf('Lateral-directional response to a cross-wind perturbation
\n');
fprintf('=========================================================
\n');
figN = figN+1;figure(figN);clf;
x0lat = deg2rad([5;5;5;10]);
initial(Glat,x0lat);[y,t] = initial(Glat,x0lat);Ip = lsiminfo(y,t);
title('OpenLoop Response to a Lateral cross-wind perturbation');
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LatGustOL');
for i = 1:numel(Ip)
    fprintf('I.C. Response to a Lateral cross-wind perturbation %s:
\n',...
        StateNames{i});
    disp(Ip(i))
end
```

```
=========================================================
Lateral-directional response to a cross-wind perturbation
=========================================================
I.C. Response to a Lateral cross-wind perturbation Delta u:
    SettlingTime: 5.6309e+04
           Min: -0.0053
       MinTime: 17.9163
           Max: 1.0698e+21
       MaxTime: 5.6329e+04
I.C. Response to a Lateral cross-wind perturbation Delta w:
    SettlingTime: 5.6309e+04
           Min: 1.2757e-04
       MinTime: 26.8744
           Max: 3.9270e+21
       MaxTime: 5.6329e+04
I.C. Response to a Lateral cross-wind perturbation Delta q:
    SettlingTime: 5.6309e+04
           Min: -0.0675
```

```
         MinTime: 8.9581
             Max: 6.3337e+22
         MaxTime: 5.6329e+04
I.C. Response to a Lateral cross-wind perturbation Delta theta:
    SettlingTime: 5.6309e+04
             Min: 0.1745
         MinTime: 0
             Max: 3.8195e+24
         MaxTime: 5.6329e+04
```

Longitudinal Stability Augmentation System (SAS)

Longitudinal Dynamics full state-feedback SAS

```
dzetap = 0.707; % Desired phugoid damping
dzetasp = 0.707; % Desired short-period damping

% Desired Poles
dPlon = [...
    wnp*(-dzetap + 1i*sqrt(1-dzetap^2));
    wnp*(-dzetap - 1i*sqrt(1-dzetap^2));
    wnsp*(-dzetasp + 1i*sqrt(1-dzetasp^2));
    wnsp*(-dzetasp - 1i*sqrt(1-dzetasp^2));
    ]

% Pole Placement
Klon = place(Alon,Blon,dPlon)

% Obtain ClosedLoop
GlonCL = ss(Alon-Blon*Klon,Blon,eye(4),0);
GlonCL.StateName = {'\Deltau','\Deltaw','\Deltaq','\Delta\theta'};
GlonCL.OutputName = {'\Deltau','\Deltaw','\Deltaq','\Delta\theta'};
GlonCL.InputName = {'\Delta\delta_e'};

% Analysis
fprintf('=========================================\n');
fprintf('ClosedLoop Response to a Longitudinal Gust\n');
fprintf('=========================================\n');
figN = figN+1;figure(figN);clf;
x0lon = [10;10;deg2rad(5);deg2rad(5)];
initial(GlonCL,x0lon);[y,t] = initial(GlonCL,x0lon);Ip =
 lsiminfo(y,t);
title('ClosedLoop Response to a Longitudinal gust');
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LonGustCL');
for i = 1:numel(Ip)
    fprintf('ClosedLoop I.C. Response to a Longitudinal gust %s:
\n',...
        StateNames{i});
    disp(Ip(i))
```

```
end

% Control Effort
ulon = zeros(length(y),1);
for i = 1:length(y)
    ulon(i) = -Klon*y(i,:)';
end
ulon = rad2deg(ulon);
figN = figN+1;figure(figN);clf;
plot(t,ulon,'b');
title('Control Effort to a Longitudinal gust');xlabel('Time (s)');
ylabel('\Delta\delta_e(t) (deg)')
pp(gcf);print(gcf,'-dpdf','-bestfit','LonGustCLKEffort');

% Wrap into timeseries and obtain per second diff
tsulon = timeseries(ulon,t,'Name','Elevator Deflection');
tsulon = resample(tsulon,0:1:t(end));
ulonrate = max(diff(tsulon.Data)); % deg/s
fprintf('Maximum Elevator Deflection = %2.2f deg\n',max(ulon));
fprintf('Elevator Max Commanded Rate = %2.2f deg/s\n',ulonrate);
```

```
dPlon =
  -0.0165 + 0.0165i
  -0.0165 - 0.0165i
  -2.5543 + 2.5551i
  -2.5543 - 2.5551i
Klon =
   -0.0000    0.0001   -0.4746   -0.1287
==========================================
ClosedLoop Response to a Longitudinal Gust
==========================================
ClosedLoop I.C. Response to a Longitudinal gust Delta u:
    SettlingTime: 311.8141
             Min: -37.2775
         MinTime: 52.2124
             Max: 10
         MaxTime: 0
ClosedLoop I.C. Response to a Longitudinal gust Delta w:
    SettlingTime: 210.5334
             Min: -16.9671
         MinTime: 1.4063
             Max: 23.6782
         MaxTime: 0.1803
ClosedLoop I.C. Response to a Longitudinal gust Delta q:
    SettlingTime: 1.4467
             Min: -0.0320
         MinTime: 0.5409
             Max: 0.0873
         MaxTime: 0
ClosedLoop I.C. Response to a Longitudinal gust Delta theta:
    SettlingTime: 213.9027
             Min: -0.0077
         MinTime: 122.7424
             Max: 0.0953
```

```
         MaxTime: 0.2163
Maximum Elevator Deflection = 2.96 deg
Elevator Max Commanded Rate = 0.34 deg/s
```

Lateral Stability Augmentation System (SAS)

Lateral Dynamics full state-feedback SAS

```
dzetaDR = 0.707; % Desired DutchRoll damping
dwnDR = 1; % Desired DutchRoll Natural Frequency

% Desired Poles
sprialP = -0.4;
dPlat = [...
    dwnDR*(-dzetaDR + 1i*sqrt(1-dzetaDR^2));
    dwnDR*(-dzetaDR - 1i*sqrt(1-dzetaDR^2));
    Dlat(3); % Keep the Roll-Subsidance as it is
    sprialP;
    ]

% Pole Placement
Klat = place(Alat,Blat,dPlat)

% Obtain ClosedLoop
GlatCL = ss(Alat-Blat*Klat,Blat,eye(4),0);
GlatCL.StateName = {'\Delta\beta','\Deltap','\Deltar','\Delta\phi'};
GlatCL.OutputName = {'\Delta\beta','\Deltap','\Deltar','\Delta\phi'};
GlatCL.InputName = {'\Delta\delta_a','\Delta\delta_r'};

% Analysis
fprintf('==========================================================================
\n');
fprintf('ClosedLoop Lateral-directional response to a cross-wind
 perturbation\n');
fprintf('==========================================================================
\n');
figN = figN+1;figure(figN);clf;
x0lat = deg2rad([5;5;5;10]);
initial(GlatCL,x0lat);[y,t] = initial(GlatCL,x0lat);Ip =
 lsiminfo(y,t);
title('ClosedLoop Response to a Lateral cross-wind perturbation');
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LatGustCL');
for i = 1:numel(Ip)
    fprintf('ClosedLoop I.C. Response to a Lateral cross-wind
 perturbation %s:\n',...
        StateNames{i});
    disp(Ip(i))
end

% Control Effort
ulatdeltaa = zeros(length(y),1);
```

```
ulatdeltar = zeros(length(y),1);
for i = 1:length(y)
    ulatdeltaa(i) = -Klat(1,:)*y(i,:)';
    ulatdeltar(i) = -Klat(2,:)*y(i,:)';
end
ulatdeltaa = rad2deg(ulatdeltaa);ulatdeltar = rad2deg(ulatdeltar);
figN = figN+1;figure(figN);clf;
plot(t,ulatdeltaa,'b',t,ulatdeltar,'r');
legend('\Delta\delta_a','\Delta\delta_r');
title('Control Effort to a Lateral-directional cross-wind
 perturbation');
xlabel('Time (s)');ylabel('Control Effort (deg)');
pp(gcf);print(gcf,'-dpdf','-bestfit','LatGustCLKEffort');

% Wrap into timeseries and obtain per second diff
tsulatdeltaa = timeseries(ulatdeltaa,t,'Name','Elevator Deflection');
tsulatdeltaa = resample(tsulatdeltaa,0:1:t(end));
ulatdeltaarate = max(diff(tsulatdeltaa.Data)); % deg/s
fprintf('Maximum Ailerons Deflection = %2.2f deg\n',max(ulatdeltaa));
fprintf('Ailerons Max Commanded Rate = %2.2f deg/s\n',ulatdeltaarate);

tsulatdeltar = timeseries(ulatdeltar,t,'Name','Elevator Deflection');
tsulatdeltar = resample(tsulatdeltar,0:1:t(end));
ulatdeltarrate = max(diff(tsulatdeltar.Data)); % deg/s
fprintf('Maximum Rudder Deflection = %2.2f deg\n',max(ulatdeltar));
fprintf('Rudder Max Commanded Rate = %2.2f deg/s\n',ulatdeltarrate);
```

```
dPlat =
  -0.7070 + 0.7072i
  -0.7070 - 0.7072i
  -1.0236 + 0.0000i
  -0.4000 + 0.0000i
Klat =
    0.1967    0.0140    0.0248    0.0139
   -1.4592   -0.0002   -0.1562    0.0020
====================================================================
ClosedLoop Lateral-directional response to a cross-wind perturbation
====================================================================
ClosedLoop I.C. Response to a Lateral cross-wind perturbation Delta u:
    SettlingTime: 4.6681
             Min: -0.0079
         MinTime: 2.6995
             Max: 0.0873
         MaxTime: 0
ClosedLoop I.C. Response to a Lateral cross-wind perturbation Delta w:
    SettlingTime: 11.3224
             Min: -0.0434
         MinTime: 2.1596
             Max: 0.0873
         MaxTime: 0
ClosedLoop I.C. Response to a Lateral cross-wind perturbation Delta q:
    SettlingTime: 4.4540
             Min: -0.0019
         MinTime: 4.0492
```

```
            Max: 0.0873
        MaxTime: 0
ClosedLoop I.C. Response to a Lateral cross-wind perturbation Delta
 theta:
    SettlingTime: 11.5438
             Min: 2.5871e-04
         MinTime: 18.5364
             Max: 0.1993
         MaxTime: 0.6299
Maximum Ailerons Deflection = 0.02 deg
Ailerons Max Commanded Rate = 0.90 deg/s
Maximum Rudder Deflection = 8.06 deg
Rudder Max Commanded Rate = 0.27 deg/s
```

Longitudinal ClosedLoop Eigenvalues

```
fprintf('======================================\n');
fprintf('Longitudinal ClosedLoop Eigen Analysis\n');
fprintf('======================================\n');
[VlonCL,DlonCL] = eig(GlonCL.A);DlonCL = diag(DlonCL);
Eigenvalues = DlonCL %#ok<*NOPTS,*NASGU>

% Normalize evecs by theta direction and non-dimensionalize
for ii = 1:4
    VlonCL(:,ii) = VlonCL(:,ii)./(VlonCL(4,ii));
end
VlonCL(1,:) = VlonCL(1,:)/u0;
VlonCL(2,:) = VlonCL(2,:)/u0;
VlonCL(3,:) = VlonCL(3,:)*(cbar/2/u0);
EigenvectorsMagnitude = abs(VlonCL)
EigenvectorsPhase = angle(VlonCL)

lamsp = DlonCL(1:2);
disp('### Short Period Mode:')
Tsp = 2*pi/imag(lamsp(1));
wnsp = sqrt(lamsp(1)*lamsp(2));
zetasp = -(lamsp(1)+lamsp(2))/2/wnsp;
disp(['Period: ' num2str(Tsp) ' seconds'])
disp(['Natural Frequency: ' num2str(wnsp) ' rad/s'])
disp(['Damping Ratio: ' num2str(zetasp)])

lamp = DlonCL(3:4);
disp('### Long Period (Phugoid) Mode:')
Tp = 2*pi/imag(lamp(1));
wnp = sqrt(lamp(1)*lamp(2));
zetap = -(lamp(1)+lamp(2))/2/wnp;
disp(['Period: ' num2str(Tp) ' seconds'])
disp(['Natural Frequency :' num2str(wnp) ' rad/s'])
disp(['Damping Ratio: ' num2str(zetap)]);
fprintf(newline);
```

```
======================================
Longitudinal ClosedLoop Eigen Analysis
```

```
=======================================
Eigenvalues =
  -2.5543 + 2.5551i
  -2.5543 - 2.5551i
  -0.0165 + 0.0165i
  -0.0165 - 0.0165i
EigenvectorsMagnitude =
    0.0113    0.0113    0.7870    0.7870
    1.0386    1.0386    0.0991    0.0991
    0.0096    0.0096    0.0001    0.0001
    1.0000    1.0000    1.0000    1.0000
EigenvectorsPhase =
    0.7230   -0.7230    1.1609   -1.1609
    0.0638   -0.0638    2.8561   -2.8561
    2.3560   -2.3560    2.3560   -2.3560
         0         0         0         0
### Short Period Mode:
Period: 2.4591 seconds
Natural Frequency: 3.6129 rad/s
Damping Ratio: 0.707
### Long Period (Phugoid) Mode:
Period: 381.7756 seconds
Natural Frequency :0.023271 rad/s
Damping Ratio: 0.707
```

Lateral ClosedLoop Eigenvalues

```
fprintf('=================================\n');
fprintf('Lateral ClosedLoop Eigen Analysis\n');
fprintf('=================================\n');
[VlatCL,DlatCL] = eig(GlatCL.A);DlatCL = diag(DlatCL);
Eigenvalues = DlatCL

% Normalize evecs by phi direction and non-dimensionalize
for ii = 1:4
    VlatCL(:,ii) = VlatCL(:,ii)./(VlatCL(4,ii));
end
Vlon(1,:) = Vlon(1,:)*(b/2/u0);
Vlon(2,:) = Vlon(2,:)/u0;
Vlon(3,:) = Vlon(3,:)/u0;
EigenvectorsMagnitude = abs(Vlat)
EigenvectorsPhase = angle(Vlat)

dutchr = DlatCL(1:2);
disp('### Dutch Roll Mode:')
Tp = 2*pi/imag(dutchr(1));
wnd = sqrt(dutchr(1)*dutchr(2));
zetad = -(dutchr(1)+dutchr(2))/2/wnd;
disp(['Period: ' num2str(Tp) ' seconds'])
disp(['Natural Frequency :' num2str(wnd) ' rad/s'])
disp(['Damping Ratio: ' num2str(zetad)]);
```

```
% ClosedLoop Eigenvalue Plot
figN = figN+1;figure(figN);clf;
subplot(3,2,[1 2 3 4]);hold on;box on;grid on;
plt1 = plot(real(DlonCL(1:2)),imag(DlonCL(1:2)),'bs',...

 real(DlonCL(3:4)),imag(DlonCL(3:4)),'b^','MarkerSize',10,'LineWidth',2);
plt2 =
 plot(real(DlatCL(1:2)),imag(DlatCL(1:2)),'r*',real(DlatCL(3)),imag(DlatCL(3)),...

  'ro',real(DlatCL(4)),imag(DlatCL(4)),'rx','MarkerSize',10,'LineWidth',2);
title('Longitudinal and Lateral ClosedLoop
 Eigenvalues','FontSize',14);
xlabel('Real(\lambda)','FontSize',14);
ylabel('Imag(\lambda)','FontSize',14);

% Put a Ractangle
r = rectangle('Position',[-0.1 -0.5 0.20
 1],'LineStyle','--','LineWidth',1);

% Put a new axis for zoomed plot
subplot(3,2,5);
box on;hold on;grid on;
plt3 =
 plot(real(DlonCL(3:4)),imag(DlonCL(3:4)),'b^','MarkerSize',10,'LineWidth',2);
title('Zoomed Plot Near Origin','FontSize',14);ylim([-0.04 0.04]);
xlabel('Real(\lambda)','FontSize',14);
ylabel('Imag(\lambda)','FontSize',14);

% Legend plot
sh=subplot(3,2,6);
p=get(sh,'position');
lh=legend(sh,[plt1; plt2],...
    'Short Period Mode','Long Period (Phugoid) Mode',...
    'Dutch Roll Mode','Roll Subsidence Mode','Spiral
 Mode','FontSize',12);
set(lh,'position',p);
axis(sh,'off');
print(gcf,'-dpdf','-fillpage','EigenValuesCL');

=================================
Lateral ClosedLoop Eigen Analysis
=================================
Eigenvalues =
  -0.7070 + 0.7072i
  -0.7070 - 0.7072i
  -1.0236 + 0.0000i
  -0.4000 + 0.0000i
EigenvectorsMagnitude =
   18.6838   18.6838    0.0024    0.0003
    3.3331    3.3331    1.0236    0.0010
   62.2004   62.2004    0.0144    0.0166
    1.0000    1.0000    1.0000    1.0000
EigenvectorsPhase =
   -0.6144    0.6144    3.1416         0
```

```
    1.6175   -1.6175    3.1416         0
   -2.1765    2.1765         0         0
         0         0         0         0
### Dutch Roll Mode:
Period: 8.8844 seconds
Natural Frequency :1 rad/s
Damping Ratio: 0.707
```

All OpenLoop and ClosedLoop Eigenvalue plot

```
figN = figN+1;figure(figN);clf;
subplot(3,2,[1 2 3 4]);hold on;box on;grid on;

% Plot OpenLoop
plt1 = plot(real(Dlon(1:2)),imag(Dlon(1:2)),'cs',...

 real(Dlon(3:4)),imag(Dlon(3:4)),'c^','MarkerSize',10,'LineWidth',2);
plt2 =
 plot(real(Dlat(1:2)),imag(Dlat(1:2)),'c*',real(Dlat(3)),imag(Dlat(3)),...

  'co',real(Dlat(4)),imag(Dlat(4)),'cx','MarkerSize',10,'LineWidth',2);

% Plot CloseLoop
plt3 = plot(real(DlonCL(1:2)),imag(DlonCL(1:2)),'ms',...

 real(DlonCL(3:4)),imag(DlonCL(3:4)),'m^','MarkerSize',10,'LineWidth',2);
plt4 =
 plot(real(DlatCL(1:2)),imag(DlatCL(1:2)),'m*',real(DlatCL(3)),imag(DlatCL(3)),...

  'mo',real(DlatCL(4)),imag(DlatCL(4)),'mx','MarkerSize',10,'LineWidth',2);
title('Longitudinal and Lateral Eigenvalues','FontSize',14);
xlabel('Real(\lambda)','FontSize',14);
ylabel('Imag(\lambda)','FontSize',14);

% Put a Ractangle
r = rectangle('Position',[-0.1 -0.5 0.20
 1],'LineStyle','--','LineWidth',1);

% Put a new axis for zoomed plot
subplot(3,2,5);
box on;hold on;grid on;
plt5 =
 plot(real(Dlon(3:4)),imag(Dlon(3:4)),'c^','MarkerSize',10,'LineWidth',2);
plt6 =
 plot(real(Dlat(4)),imag(Dlat(4)),'cx','MarkerSize',10,'LineWidth',2);
plt7 =
 plot(real(DlonCL(3:4)),imag(DlonCL(3:4)),'m^','MarkerSize',10,'LineWidth',2);
title('Zoomed Plot Near Origin','FontSize',14);ylim([-0.04 0.04]);
xlabel('Real(\lambda)','FontSize',14);
ylabel('Imag(\lambda)','FontSize',14);

% Legend plot
sh=subplot(3,2,6);
```

```
p=get(sh,'position');
lh=legend(sh,[plt1; plt2; plt3; plt4],...
    'Short Period Mode (OL)','Phugoid Mode (OL)',...
    'Dutch Roll Mode (OL)','Roll Subsidence Mode (OL)','Spiral Mode
 (OL)',...
    'Short Period Mode (CL)','Phugoid Mode (CL)',...
    'Dutch Roll Mode (CL)','Roll Subsidence Mode (CL)','Spiral Mode
 (CL)','FontSize',12);
set(lh,'position',p);
axis(sh,'off');
print(gcf,'-dpdf','-fillpage','EigenValuesCombined');
```

ClosedLoop Response to Pilot Step Command

```
fprintf('==========================================\n');
fprintf('ClosedLoop Response to Pilot Step Command\n');
fprintf('==========================================\n');

% Elevator
figN = figN+1;figure(figN);clf;
step(GlonCL,opt);grid on;title('ClosedLoop Elevator Step Response')
[y,t] = step(GlonCL,opt);
S = stepinfo(y,t);
StateNames = {'Delta u','Delta w','Delta q','Delta theta'};
for i = 1:numel(S)
    fprintf('Elevator Stepinfo Data for Longitudinal State %s:
\n',StateNames{i});
    disp(S(i))
end
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LonStepRespCL');

% Ailerons
figN = figN+1;figure(figN);clf;
step(GlatCL(:,1),opt);grid on;title('ClosedLoop Ailerons Step
 Response')
[y,t] = step(GlatCL(:,1),opt);
S = stepinfo(y,t);
StateNames = {'Delta beta','Delta p','Delta r','Delta phi'};
for i = 1:numel(S)
    fprintf('Ailerons Stepinfo Data for Lateral State %s:
\n',StateNames{i});
    disp(S(i))
end
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LatStepRespCL1');

% Rudder
figN = figN+1;figure(figN);clf;
step(GlatCL(:,2),opt);grid on;title('ClosedLoop Rudder Step Response')
[y,t] = step(GlatCL(:,2),opt);
```

```
S = stepinfo(y,t);
StateNames = {'Delta beta','Delta p','Delta r','Delta phi'};
for i = 1:numel(S)
    fprintf('Rudder Stepinfo Data for Longitudinal State %s:
\n',StateNames{i});
    disp(S(i))
end
L = findobj(gcf, 'Type', 'line');for i = 1:numel(L),L(i).Color
 = 'b';end
pp(gcf);print(gcf,'-dpdf','-fillpage','LatStepRespCL2');

==========================================
ClosedLoop Response to Pilot Step Command
==========================================
Elevator Stepinfo Data for Longitudinal State Delta u:
        RiseTime: 89.2959
    SettlingTime: 251.2917
     SettlingMin: 86.5507
     SettlingMax: 100.6451
       Overshoot: 4.6651
      Undershoot: 0
            Peak: 100.6451
        PeakTime: 179.8948
Elevator Stepinfo Data for Longitudinal State Delta w:
        RiseTime: 0.2593
    SettlingTime: 205.7061
     SettlingMin: -12.9552
     SettlingMax: -3.0246
       Overshoot: 116.1086
      Undershoot: 0
            Peak: 12.9552
        PeakTime: 1.1899
Elevator Stepinfo Data for Longitudinal State Delta q:
        RiseTime: 3.8149e-05
    SettlingTime: 43.4213
     SettlingMin: -0.0113
     SettlingMax: 2.1747e-04
       Overshoot: 2.9050e+05
      Undershoot: 5.5743e+03
            Peak: 0.0113
        PeakTime: 0.3245
Elevator Stepinfo Data for Longitudinal State Delta theta:
        RiseTime: 9.5398
    SettlingTime: 307.0416
     SettlingMin: -0.0414
     SettlingMax: -0.0187
       Overshoot: 98.8294
      Undershoot: 0
            Peak: 0.0414
        PeakTime: 59.5503
Ailerons Stepinfo Data for Lateral State Delta beta:
        RiseTime: 3.2823
    SettlingTime: 9.9187
     SettlingMin: 0.0042
```

```
     SettlingMax: 0.0047
       Overshoot: 0
      Undershoot: 58.1731
            Peak: 0.0047
        PeakTime: 19.6162
Ailerons Stepinfo Data for Lateral State Delta p:
        RiseTime: 5.3448e-04
    SettlingTime: 12.3828
     SettlingMin: 1.5785e-04
     SettlingMax: 0.1347
       Overshoot: 8.5245e+04
      Undershoot: 0
            Peak: 0.1347
        PeakTime: 1.5297
Ailerons Stepinfo Data for Lateral State Delta r:
        RiseTime: 7.3388
    SettlingTime: 11.3723
     SettlingMin: 0.0087
     SettlingMax: 0.0096
       Overshoot: 0
      Undershoot: 0
            Peak: 0.0096
        PeakTime: 19.6162
Ailerons Stepinfo Data for Lateral State Delta phi:
        RiseTime: 6.1196
    SettlingTime: 10.9392
     SettlingMin: 0.5545
     SettlingMax: 0.6147
       Overshoot: 0
      Undershoot: 0
            Peak: 0.6147
        PeakTime: 19.6162
Rudder Stepinfo Data for Longitudinal State Delta beta:
        RiseTime: 2.2067
    SettlingTime: 5.8368
     SettlingMin: 0.0567
     SettlingMax: 0.0644
       Overshoot: 3.7150
      Undershoot: 0
            Peak: 0.0644
        PeakTime: 4.4991
Rudder Stepinfo Data for Longitudinal State Delta p:
        RiseTime: 5.1486e-04
    SettlingTime: 12.3587
     SettlingMin: 2.2431e-05
     SettlingMax: 0.0193
       Overshoot: 8.6067e+04
      Undershoot: 0
            Peak: 0.0193
        PeakTime: 1.4397
Rudder Stepinfo Data for Longitudinal State Delta r:
        RiseTime: 0.0930
    SettlingTime: 6.7427
     SettlingMin: -0.0305
```

```
      SettlingMax: -0.0058
        Overshoot: 372.7876
       Undershoot: 0
             Peak: 0.0305
         PeakTime: 1.2598
Rudder Stepinfo Data for Longitudinal State Delta phi:
         RiseTime: 6.1355
     SettlingTime: 10.9376
      SettlingMin: 0.0788
      SettlingMax: 0.0874
        Overshoot: 0
       Undershoot: 0
             Peak: 0.0874
         PeakTime: 19.6162
```

Published with MATLAB® R2018b