Title: Design of a Stability Augmentation System (SAS) for Hypersonic Research Aircraft X-15
Date: May 13, 2019
Name: Jyot R. Buch

Abstract

The goal of this paper is to design a Stability Augmentation System (SAS) for X-15 hypersonic research aircraft. X-15 is a rocket powered airplane that could fly up to Mach 6 at altitudes of up to 300,000 ft above the sea level. Nonlinear dynamical equations of motion were linearized for steady level flight condition. Resulting Linear Time-Invariant (LTI) state-space representation of longitudinal and lateral dynamics were used to design SAS. Eigenvalues and initial condition response were used to analyze the natural behavior of an aircraft. In order to achieve the desired behavior by means of control forcing, SAS was designed using MATLAB computational tools. A pair of full-state feedback controller was designed for each LTI model. Closed-loop system performance was analyzed to meet the design requirements. Bryan's assumptions were enforced throughout the study. Such a design shows promising results in damping improvement due to longitudinal and lateral-directional wind gust perturbations. Desired performance and stability was obtained as a result of this study.

1 Introduction

The North American X-15 was a hypersonic rocket-powered aircraft operated by the United States Air Force and the National Aeronautics and Space Administration as part of the X-plane series of experimental aircraft. The X-15 set speed and altitude records in the 1960s, reaching the edge of outer space and returning with valuable data used in aircraft and spacecraft design. Advanced control systems were used to ensure various performance and safety objectives throughout the flight envelope. A particular system, which is the interest of this paper is a Stability Augmentation System, often abbreviated as SAS. It is one of the automatic flight control system which is used to augment the aircraft design to ensure stability in the presence of any exogenous disturbances such as wind gust or unintentional control surface movement. This is an important part of the flight computer, as it allows us to change the behavior of the aircraft without changing the geometry, aerodynamics or physical flight design parameters.

The design of SAS is typically approached from control theoretic formulations. Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems with inputs, and how their behavior is modified by feedback. The usual objective of control theory is to control a system, often called the plant, so its output follows a desired control signal, called the reference, which may be a fixed or changing value. Often a higher level complicated system is abstracted to a block diagram level to systematically approach the design. A generic control system block diagram is shown in Fig. 1, where reference is compared to the current state of the system which is being sensed by sensors. The control decisions are made by the flight computer and are fed to actuators to drive the plant (in this case an aircraft) to achieve the desired behavior.

X-15 aircraft's geometrical, aerodynamic, stability and mass properties are given in the NASA report [4]. For the purpose of this study, it is assumed that X-15 is flying at Mach 2.0 at an altitude of 60,000 ft in a trim (equilibrium) condition [1]. Nonlinear dynamical equations of motion were linearized about this trim condition and Linear Time-Invariant state-space model was obtained [1]. Bryan's formulation for such a problem [3] in 1911 established the conventional assumption that aerodynamic forces and moments depend only on instantaneous values of aircraft states and controls, which can be expressed in a linear polynomial expansion. Similar assumptions have been made in this work which are explicitly listed below.

(A1) Quasi-steady aerodynamics and time-invariant aerodynamic coefficients.

(A2) Decoupled longitudinal and lateral-directional dynamics.

(A3) All the aircraft states are available for feedback control design.

(A4) Sensor dynamics, inaccuracies and uncertainties are ignored.

(A5) Maximum control surface deflection is $\pm 10^o$ at the maximum rate of $1^o/s$.

As will be shown later in this report that natural response of an open-loop system does not have the required characteristics but have an unstable lateral-directional behavior. Typically, in addition to dynamic stability, we would like to have enough damping to suppress the effect of external disturbances for several reasons such as ride quality, maneuverability, etc. For this project, design specifications were given in terms of a desired time-domain performance requirements. The goal is to,

- Design a full-state feedback SAS that achieves $\zeta_p \geq 0.04$ and $0.35 \leq \zeta_s \leq 1.3$ for the closed-loop longitudinal dynamics.
- Deisgn a full-state feedback SAS that achieves $\zeta_{DR} \geq 0.4$ and $\omega_{DR} \geq 1$ rad/sec for the closed-loop lateral-directional dynamics.

Section 2 explains a systematic technical approach that was used to design the SAS that meets the above mentioned objectives. Results have been discussed in section 3 followed by summary and conclusions in section 4. Figures are attached at the end of the report for completeness.

2 Technical Approach

First, nonlinear equations of motion were linearized [1][2] at a given trim condition and the following LTI representation was obtained. This type of analysis is often refereed to as the small-perturbation analysis as the linearized states are perturbation around trim condition.

$$\dot{x} = Ax + B\eta \tag{1}$$

Where, x is a state vector of interest and η is a control input vector. For longitudinal dynamics using A_{lon} and B_{lon} the state-space is given by,

$$\begin{bmatrix}\Delta\dot{u}\\ \Delta\dot{w}\\ \Delta\dot{q}\\ \Delta\dot{\theta}\end{bmatrix} = \begin{bmatrix}-0.0087 & -0.019 & 0 & -32.1\\ -0.01169 & -0.311 & 1931 & -2.244\\ 0.0004705 & -0.00673 & -0.182 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\Delta u\\ \Delta w\\ \Delta q\\ \Delta\theta\end{bmatrix} + \begin{bmatrix}6.24\\ -89.2\\ -9.8\\ 0\end{bmatrix}\Delta\delta_e \tag{2}$$

For lateral-directional dynamics using A_{lat} and B_{lat} the state-space is given by,

$$\begin{bmatrix}\Delta\dot{\beta}\\ \Delta\dot{p}\\ \Delta\dot{r}\\ \Delta\dot{\phi}\end{bmatrix} = \begin{bmatrix}-0.1274 & 0 & -1 & 0.01662\\ -0.5658 & -1.021 & 0.07293 & 0\\ 11.08 & -0.01469 & -0.1853 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\beta\\ \Delta\phi\\ \Delta r\\ \Delta\phi\end{bmatrix} + \begin{bmatrix}0 & 2.2\times10^{-5}\\ 28.86 & 4.265\\ 1.2 & -6.861\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\delta_a\\ \Delta\delta_r\end{bmatrix} \tag{3}$$

The notations used here are consistent with [2] and are slightly different from those used in [4]. The control input for the longitudinal dynamics will be the elevator deflection δ_e and control inputs for the lateral-directional dynamics will be the aileron and rudder deflection δ_a and δ_r, respectively. This notations are prefixed by Δ to represent perturbation around trim condition.

Next, we analyze the open-loop characteristics by setting the control inputs to 0 i.e. unforced response. Eigenvalue analysis will be performed to asses the stability and performance, followed by the initial condition response and forced response to the wind gust disturbance. The natural response characteristics are not as desirable, so to change this behavior we will design a full-state feedback controller using pole-placement approach. In this approach, a designer can place the poles/eigenvalues of a closed loop system to achieve the design requirement. The desired eigenvalues are given by,

$$\lambda_d = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2} \tag{4}$$

Where, ζ and ω_n are desired damping ratio and desired natural frequency. Desired eigenvalues correspond to the roots of the closed-loop characteristic polynomial, as we want the closed-loop to behave accordingly. In this work the reference input from pilot will be considered as 0, thus the linearized control decisions for a regulator design will become,

$$\eta = -Kx \tag{5}$$

Where, K is a gain matrix which will determine what signals are to be send to actuators for control surface movement. MATLAB's `place` command will be used to design such a controller. More sophisticated designs can be achieved using robust and optimal control techniques such as LQR, loop-shaping, H_∞ synthesis which are out of the scope of this project. Finally, the closed loop performance will be assessed to verify if the desired performance is achieved. The overall design process is iterative as we choose the desired damping and natural frequency and check if control effort is reasonable and desired performance is achieved.

3 Results and Discussions

3.1 Open-Loop Response

To assess the open-loop stability characteristics for linearized X-15 model following eigenvalues of system matrix A were computed.

Open-loop Eigenvalues	
Longitudinal Dynamics	Lateral Dynamics
$-0.2459 + 3.6045i$	$-0.1556 + 3.3295i$
$-0.2459 - 3.6045i$	$-0.1556 - 3.3295i$
$-0.0050 + 0.0227i$	-1.0236
$-0.0050 - 0.0227i$	0.0010

All the open-loop eigenvalues are plotted in the single plot with named after their specific characteristic modes. The blue color correspond to the longitudinal dynamics eigenvalues and red color denotes lateral-directional eigenvalues. It is to be noted that lateral dynamics have one positive eigenvalue associated with spiral mode which is on the right half of the s-plane. Positive eigenvalue denotes a growth in the time domain solution to previously mentioned differential equation and thus an unstable behavior. This can also be seen from the lateral cross-wind perturbation response in attached figure that it grows unbounded. One of the design goal will be to stabilize the lateral dynamics plant. Longitudinal step response was plotted to illustrate the effect of unintentional elevator deflection. Further longitudinal response to wind gust was also plotted to analyze the natural behavior of the system. Various time-domain characteristics such as rise time, settling time, overshoot, undershoot, etc. were obtained using MATLAB function `stepinfo` which can be observed from attached code.

3.2 SAS Design and Closed-Loop Response

Short period (Phugoid) and long period modes are clearly identifiable from the longitudinal dynamical response. Surprisingly the long period mode has a period of 276 seconds, which is really high value and not desirable. Moreover, Phugoid mode has a damping ratio of 0.21 which satisfies the requirement but short period mode has a damping ratio of 0.06 which does not satisfy the design requirement. Typically in the control design we would like to obtain a damping of at least $\zeta \geq 0.707$. Which means no peak in the bode magnitude (frequency response) plot, as it has better H_∞ norm and so good robustness to disturbance in the worst-case sense.

For lateral directional dynamics in order to stabilize the aircraft, we first need to make sure that closed-loop system has all the eigenvalues with negative real part. Dutch roll mode has a period of 1.887 seconds with natural frequency of 3 rad/s and damping ratio of 0.046. It is clearly seen that we meet the natural frequency requirement but damping ratio is not sufficient. We need to design a controller which provides a adequate closed-loop damping. It was observed that if we attempt to aim for only larger damping at the current natural frequency it produces the control input to be such that maximum rudder deflection requirement becomes $> 10^o$. Which means the actuator will saturate in order to respond to this requirement. Actuator saturation will reduce the life and performance of the actuator. Thus natural frequency of 1 rad/s was chosen to achieve the reasonable controller effort. Overall, a summary of desired closed-loop poles is provided in the following table. Cyan color coded eigenvalues were changed using the SAS design. A plot of this modified eigenvalues is also attached for reader's reference.

Desired Closed-loop Eigenvalues	
Longitudinal Dynamics	Lateral Dynamics
$-0.0165 + 0.0165i$	$-0.7070 + 0.7072i$
$-0.0165 - 0.0165i$	$-0.7070 - 0.7072i$
$-2.5543 + 2.5551i$	-1.0236
$-2.5543 + 2.5551i$	-0.4

Finally through some design iterations following controller gains were finalized for both lateral and longitudinal SAS.

$$K_{\text{lon}} = \begin{bmatrix} 0 & 10^{-4} & -0.4746 & -0.1287 \end{bmatrix} \tag{6}$$

$$K_{\text{lat}} = \begin{bmatrix} 0.1967 & 0.0140 & 0.0248 & 0.0139 \\ -1.4592 & -0.0002 & -0.1562 & 0.0020 \end{bmatrix} \tag{7}$$

After plugging in these controller gains, the closed-loop performance to external disturbance was analyzed. Moreover controller effort was plotted to make sure the control signals are reasonable. Maximum commanded elevator deflection was 2.96^o with rate of $0.34^o/s$. Similarly for lateral-directional SAS design, maximum commanded ailerons deflection was 0.02^o with rate of $0.9^o/s$ and rudder deflection was 8.06^o with rate of $0.27^o/s$. By means of lateral full-state feedback controller the effect of external disturbance was nullified within 12 seconds. Due to inherently large long period longitudinal modes, it was observed that it takes around 250 seconds to nullify the effect of disturbance. An undershoot of ≈ -45 ft/s in forward velocity was observed which may be handled by means of forward thrust.

4 Summary and Conclusions

In this work, Stability Augmentation System (SAS) for X-15 hypersonic research aircraft was designed. Linearized equations of motion for steady level flight condition were used to design longitudinal and lateral-directional Stability Augmentation System (SAS). A method called pole-placement was used to achieve the desired closed-loop performance using full state feedback controller. This approach allowed us to alter the behavior of the aircraft using control surfaces without changing the physical aircraft design parameters. Few trade-offs were made along the way to ensure the main objective of the project is satisfied. Further improvement can be made using more systematic approaches to place the poles optimally to achieve good stability margins and performance requirements.

References

[1] AEM 4303 Course Final Project Guide. 2019.

[2] Robert C Nelson et al. *Flight stability and automatic control*, volume 2. WCB/McGraw Hill New York, 1998.

[3] George Hartley Bryan. *Stability in aviation: an introduction to dynamical stability as applied to the motions of aeroplanes*. Macmillan and Co., limited, 1911.

[4] Robert K Heffley and Wayne F Jewell. Aircraft handling qualities data. 1972.

Figures

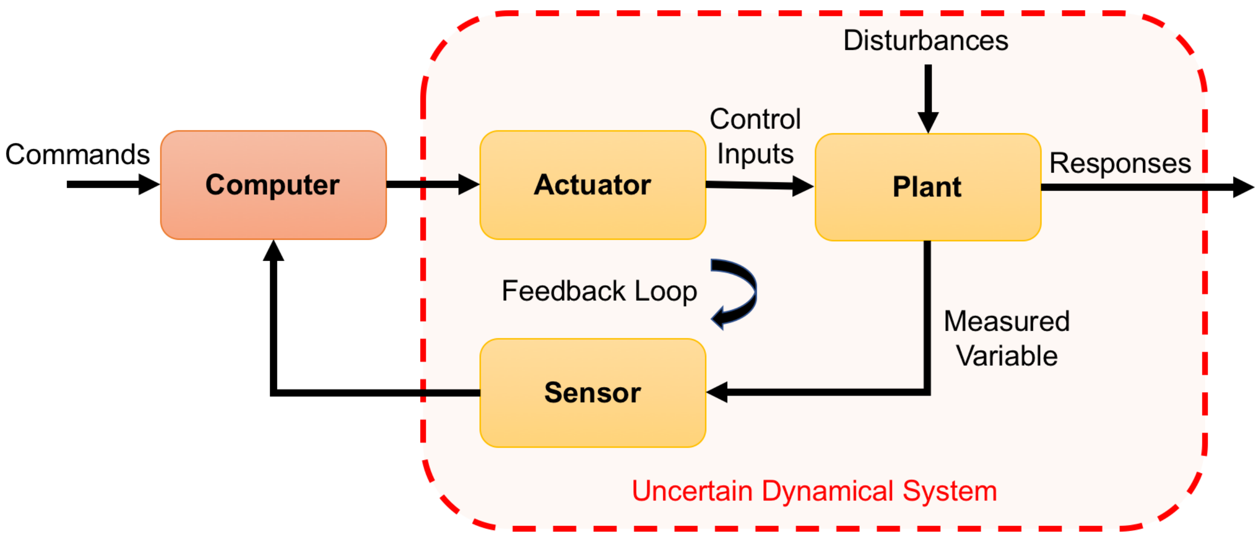
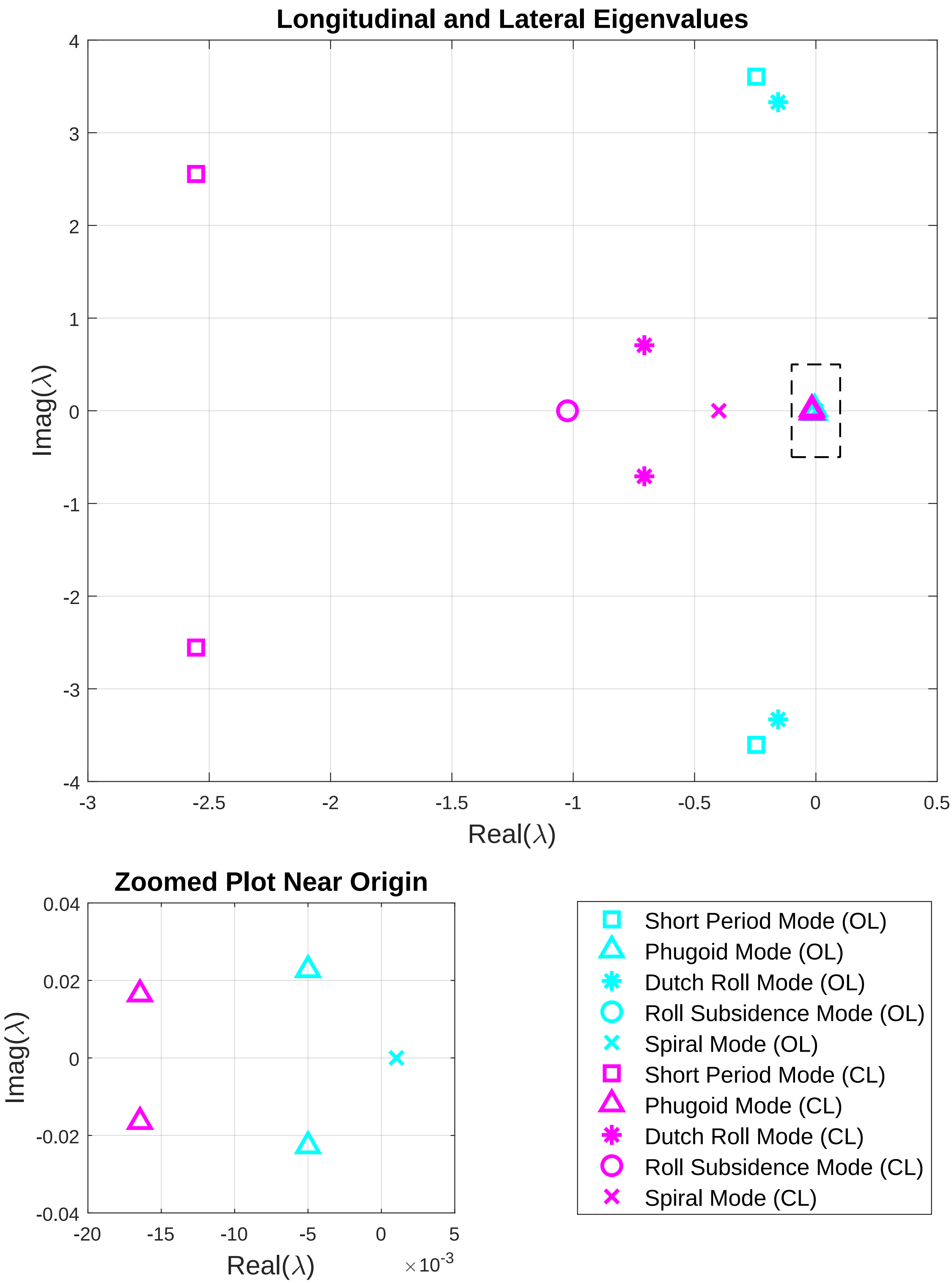
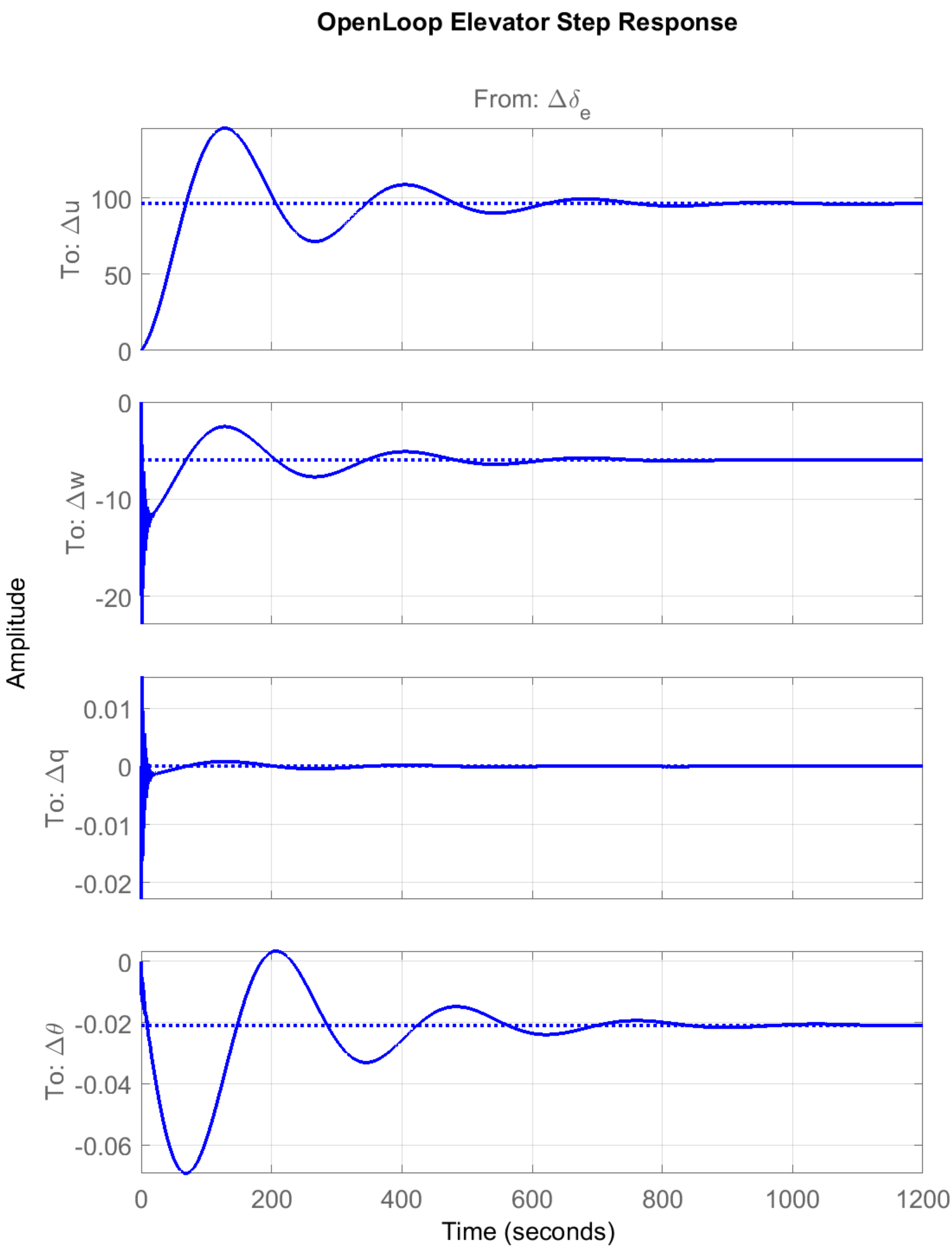
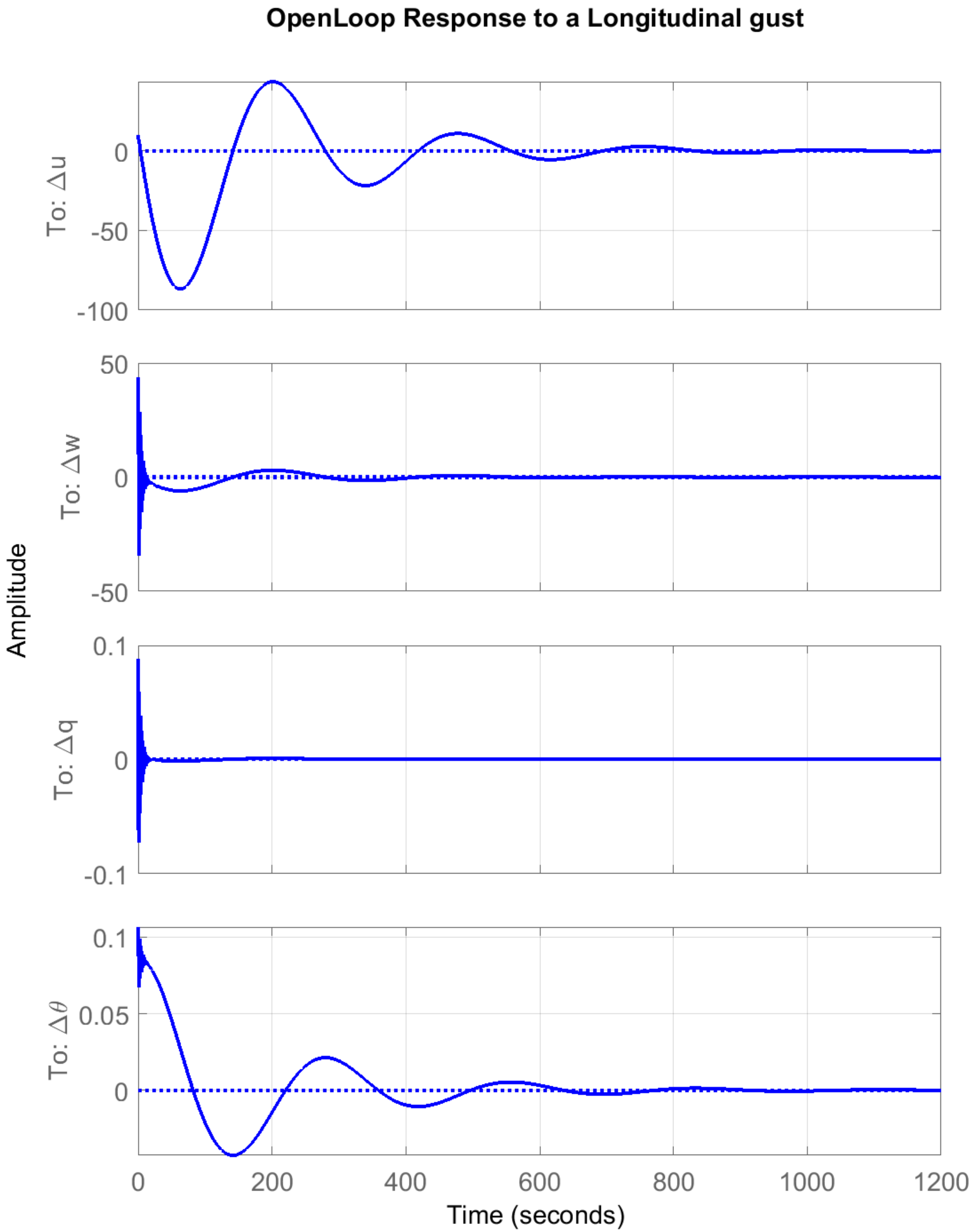


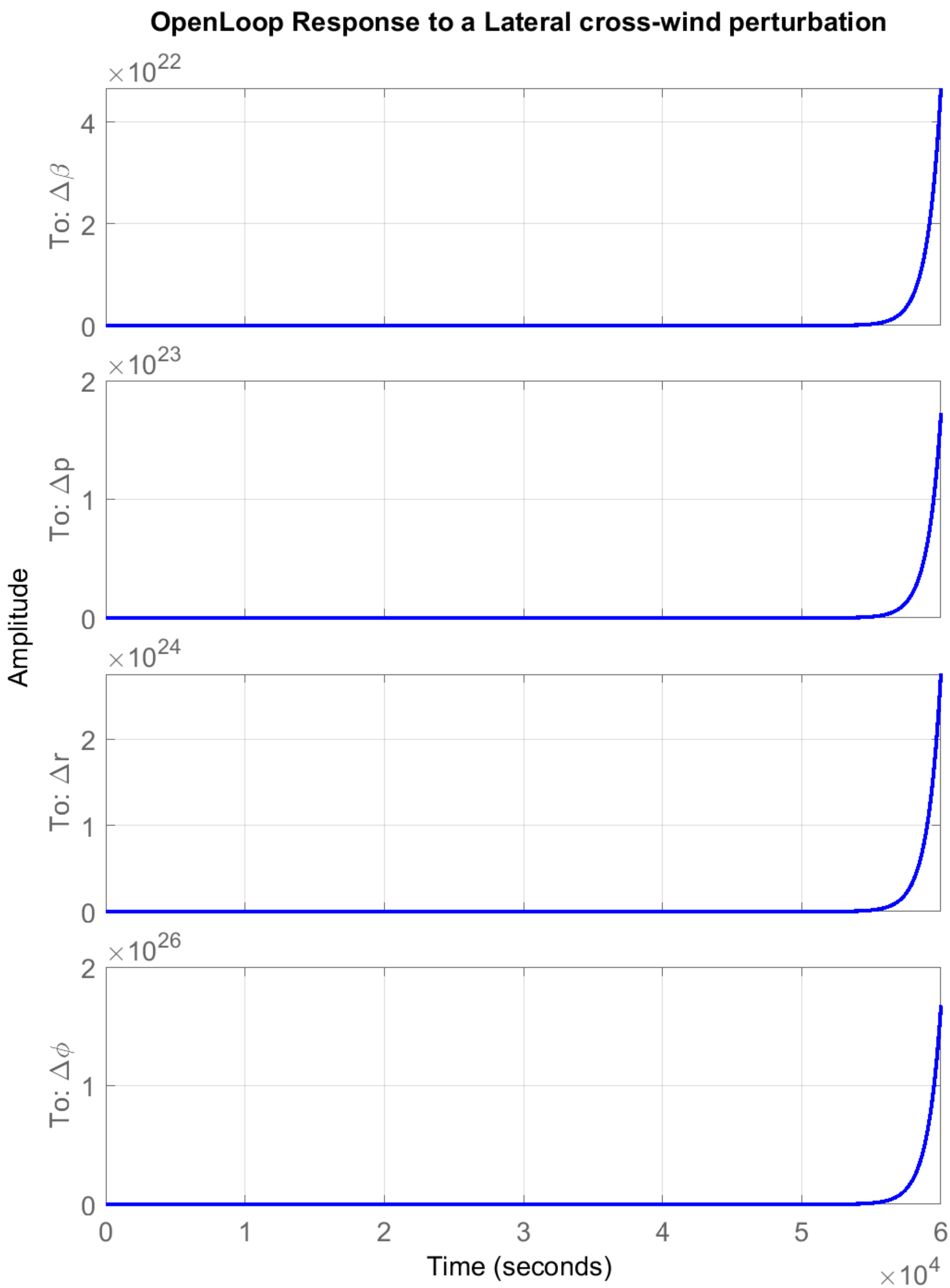
Figure 1: Generic Control System Abstraction

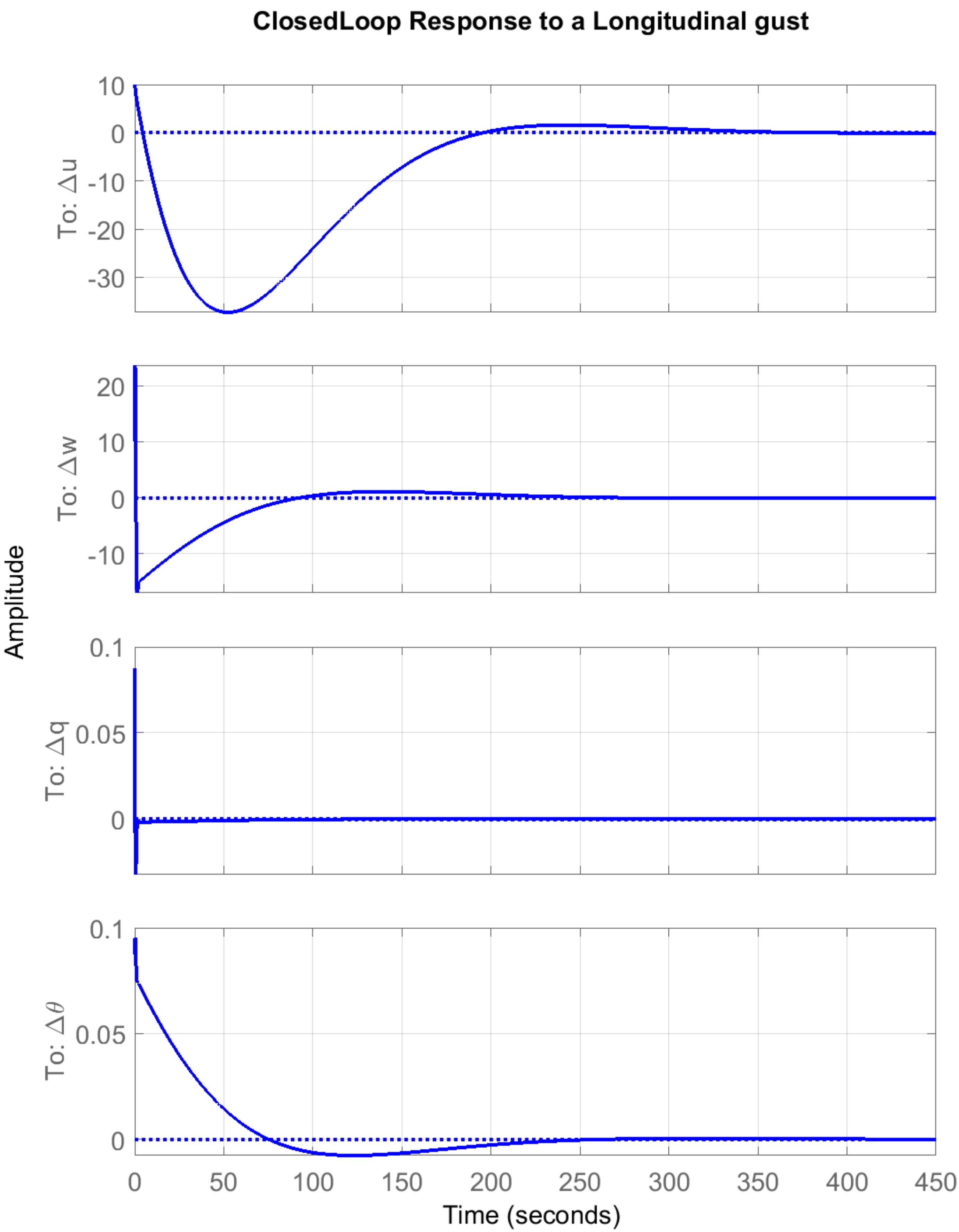

Longitudinal and Lateral Eigenvalues
Imag(λ)
Real(λ)
Zoomed Plot Near Origin
Imag(λ)
Real(λ)
×10^-3
Short Period Mode (OL)
Phugoid Mode (OL)
Dutch Roll Mode (OL)
Roll Subsidence Mode (OL)
Spiral Mode (OL)
Short Period Mode (CL)
Phugoid Mode (CL)
Dutch Roll Mode (CL)
Roll Subsidence Mode (CL)
Spiral Mode (CL)



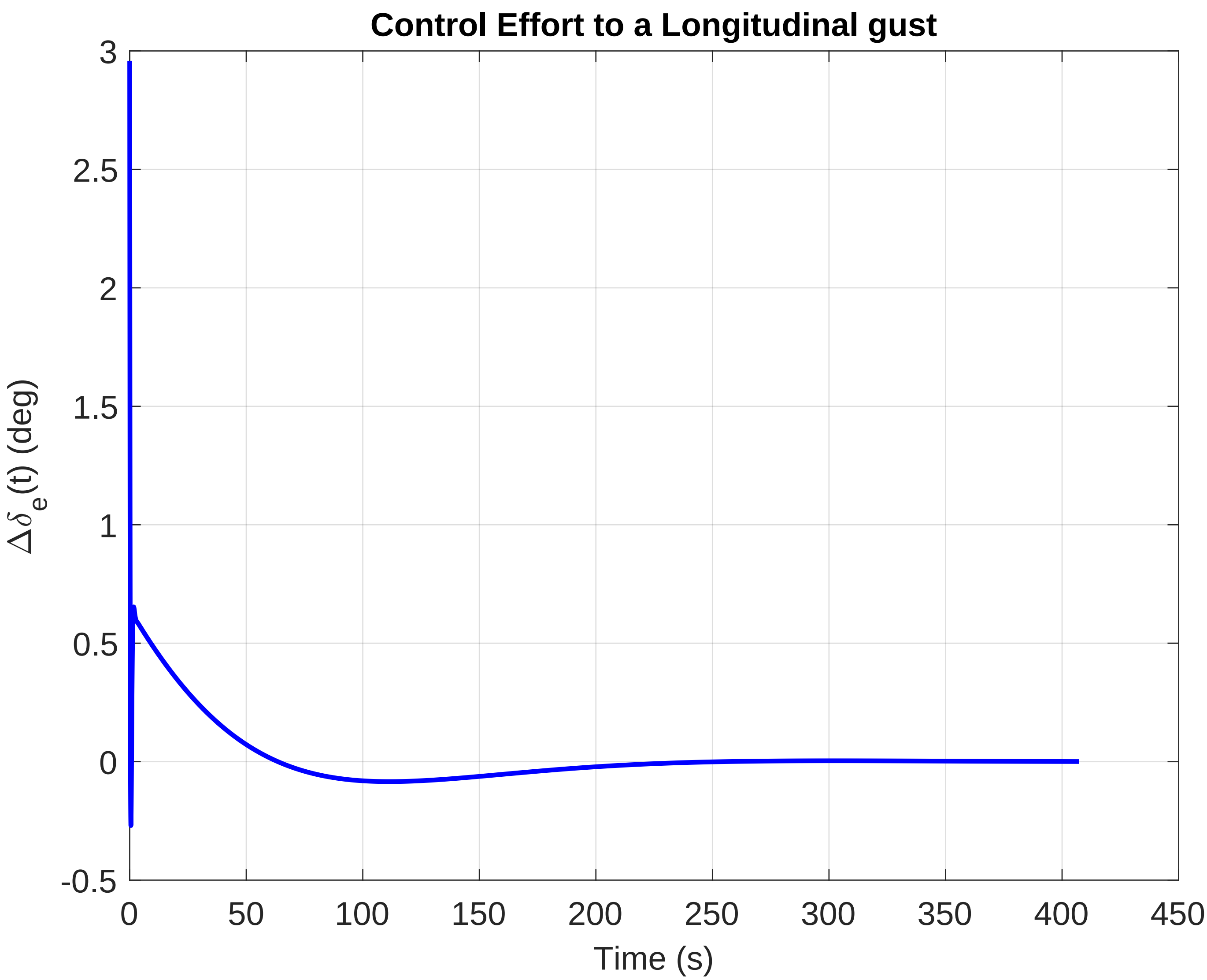
OpenLoop Elevator Step Response
From: Δδe
To: Δu
To: Δw
To: Δq
To: Δθ
Amplitude
Time (seconds)
0
50
100
-10
-20
0.01
-0.01
-0.02
-0.02
-0.04
-0.06
200
400
600
800
1000
1200

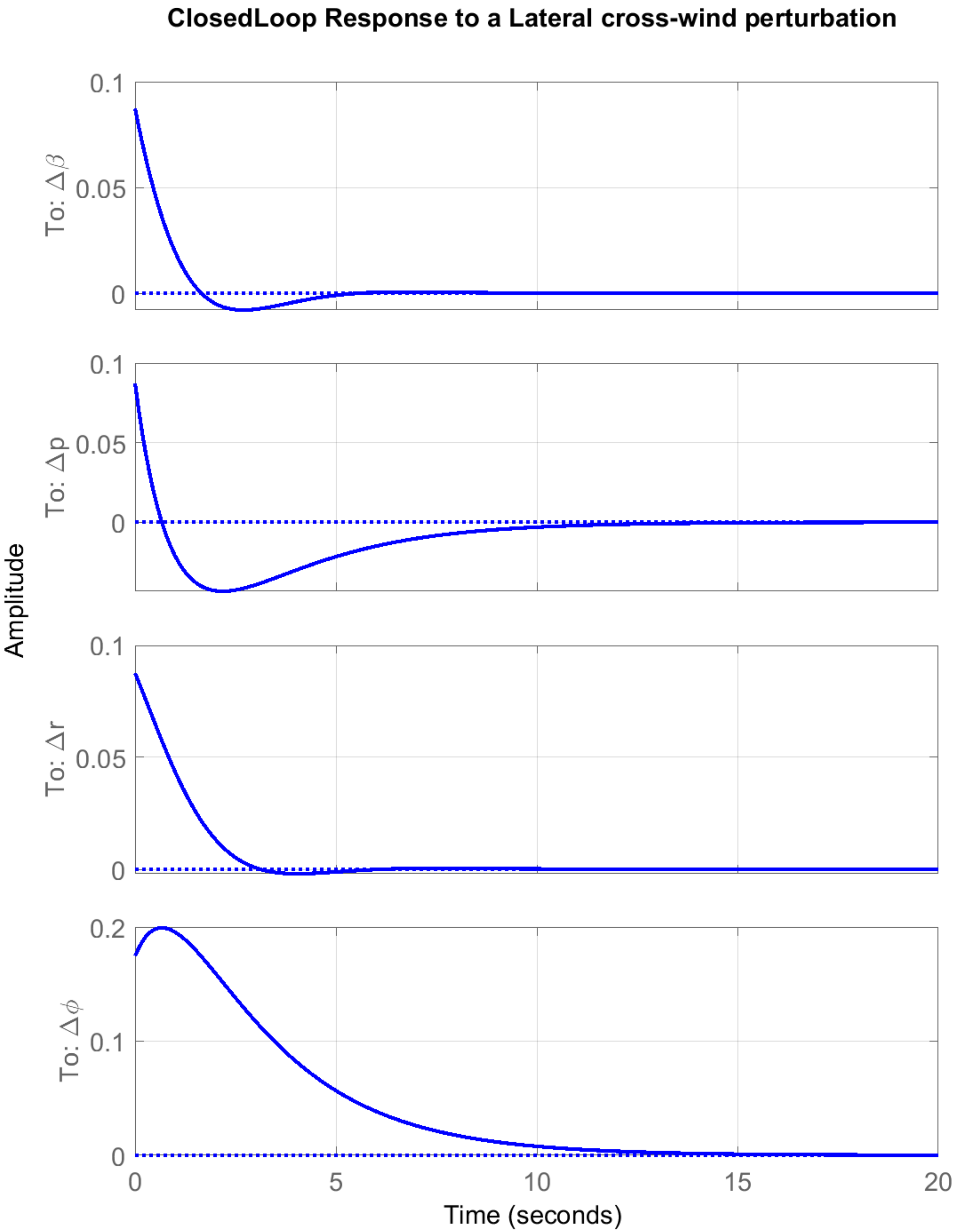

OpenLoop Response to a Longitudinal gust
To: Δu
0
-50
-100
50
To: Δw
0
-50
0.1
To: Δq
0
-0.1
0.1
To: Δθ
0.05
0
0
200
400
600
800
1000
1200
Time (seconds)
Amplitude

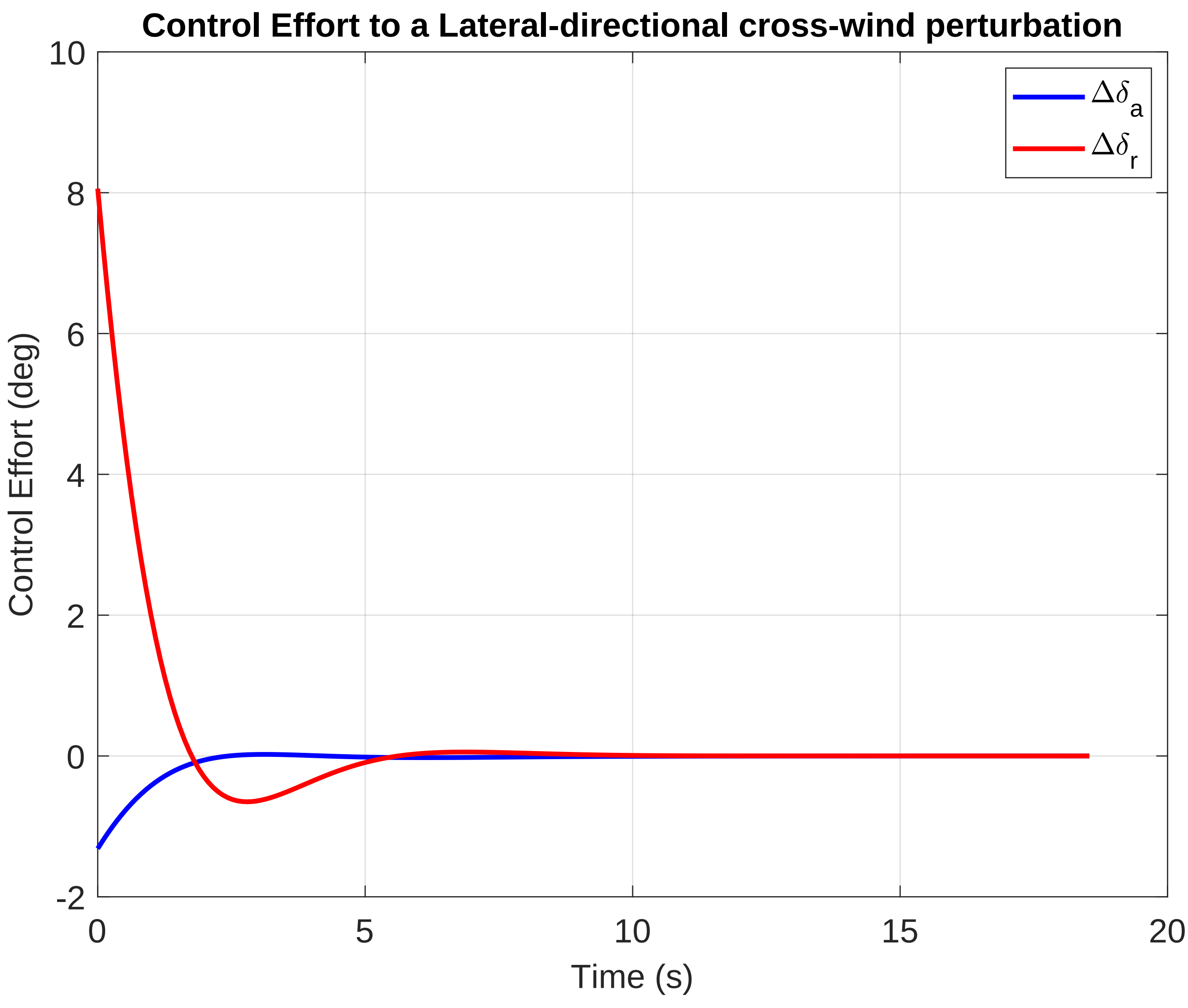

OpenLoop Response to a Lateral cross-wind perturbation
×10^22
To: Δβ
×10^23
To: Δp
×10^24
To: Δr
×10^26
To: Δϕ
Amplitude
Time (seconds)
×10^4



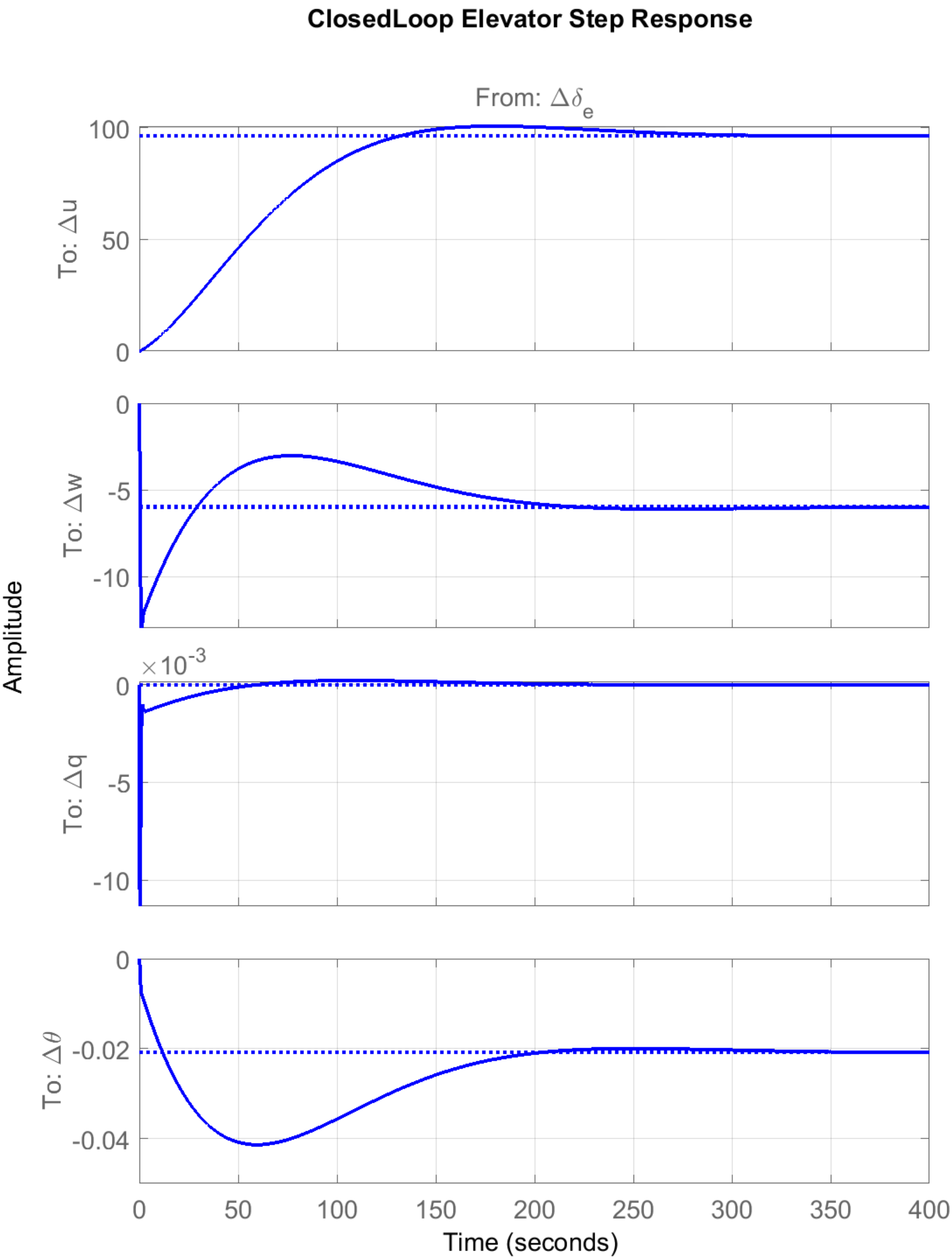
ClosedLoop Response to a Longitudinal gust
Amplitude
To: Δu
To: Δw
To: Δq
To: Δθ
10
0
-10
-20
-30
20
10
0
-10
0.1
0.05
0
0.1
0.05
0
0
50
100
150
200
250
300
350
400
450
Time (seconds)

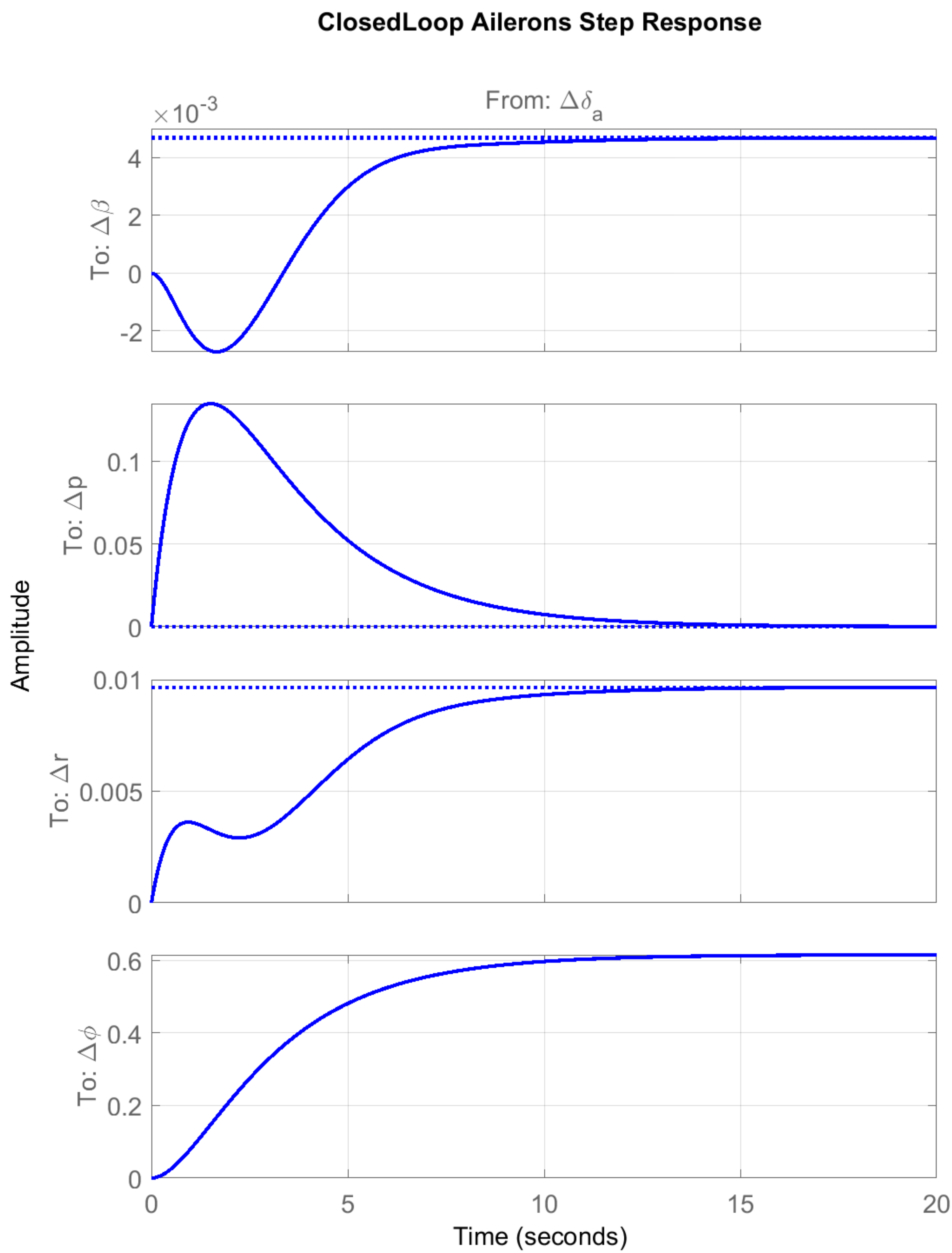

Control Effort to a Longitudinal gust
3
2.5
2
1.5
1
0.5
0
-0.5
0
50
100
150
200
250
300
350
400
450
Time (s)
$\Delta\delta_e$(t) (deg)

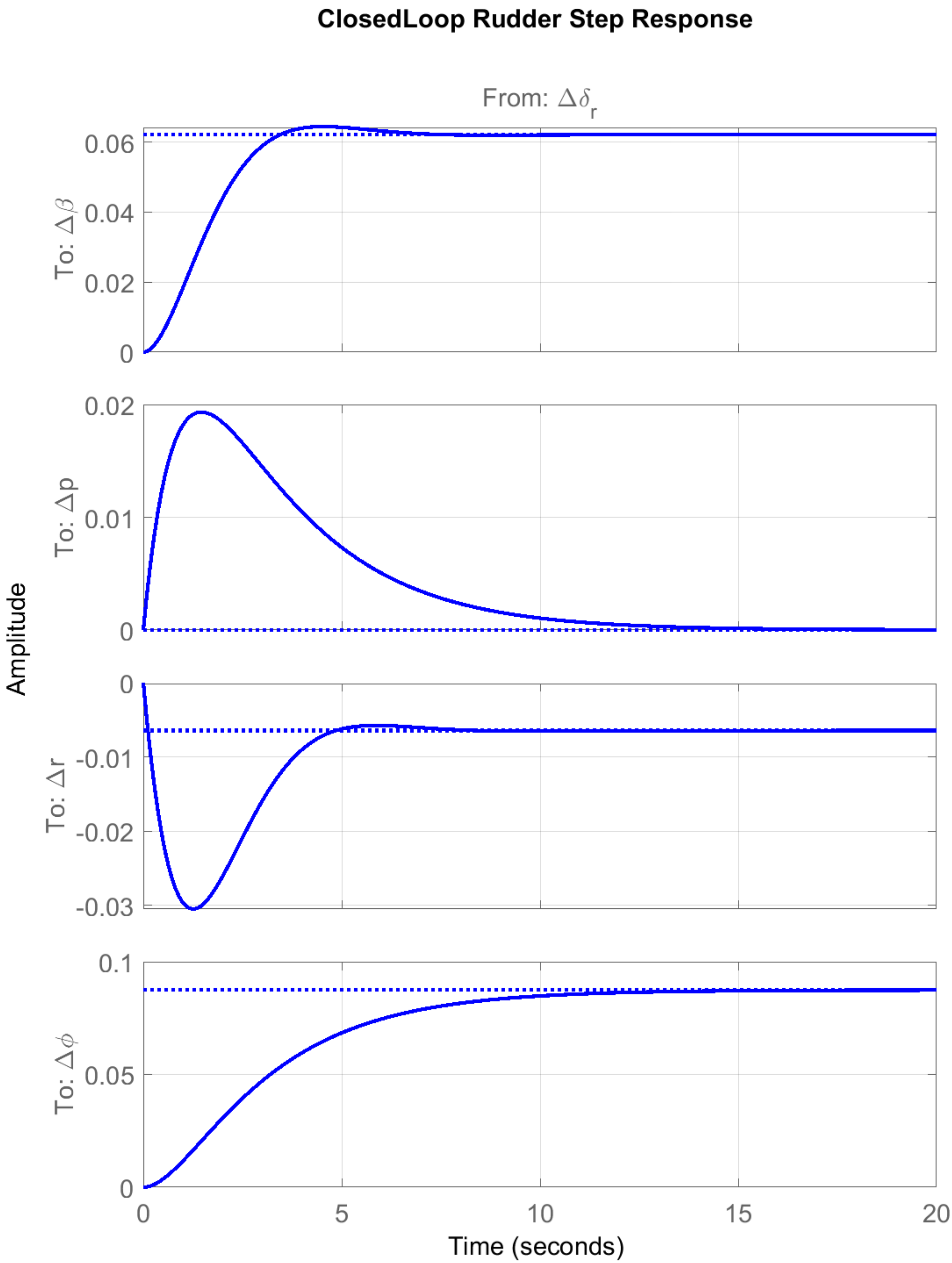

ClosedLoop Response to a Lateral cross-wind perturbation
Amplitude
To: Δβ
To: Δp
To: Δr
To: Δφ
0.1
0.05
0
0.2
5
10
15
20
Time (seconds)

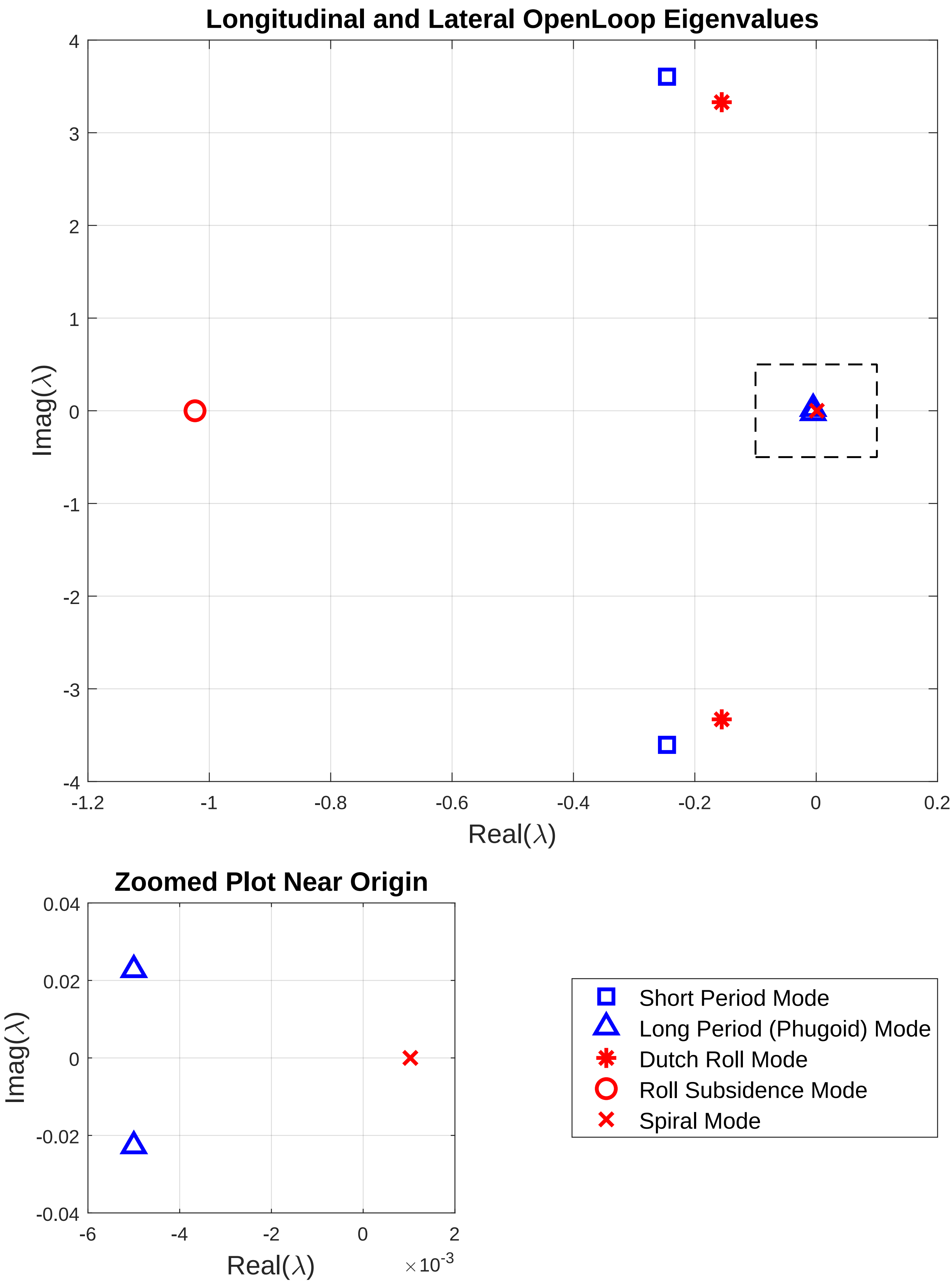


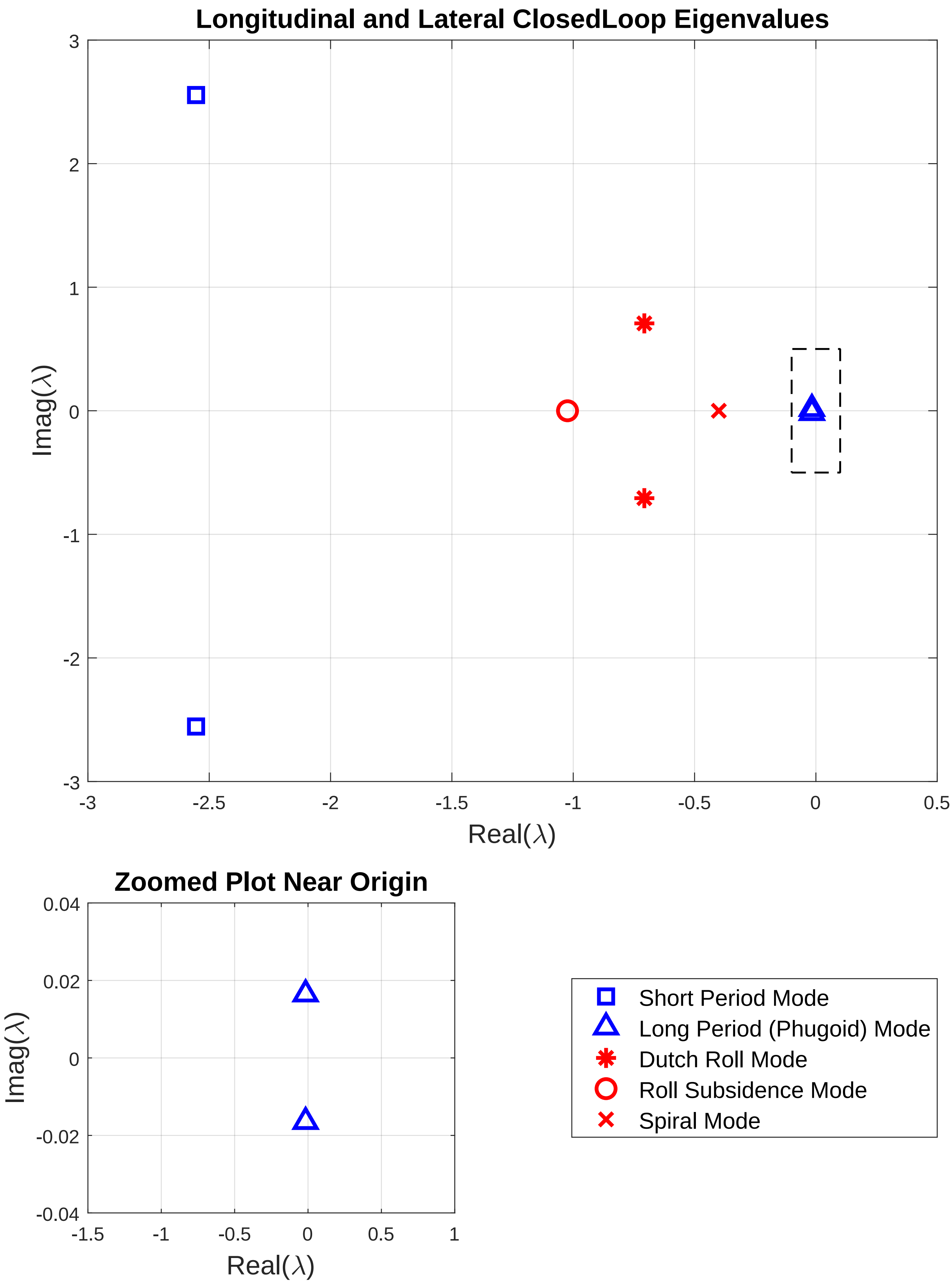
Control Effort to a Lateral-directional cross-wind perturbation
Control Effort (deg)
Time (s)
$\Delta\delta_a$
$\Delta\delta_r$
10
8
6
4
2
0
-2
0
5
10
15
20


ClosedLoop Elevator Step Response
From: $\Delta\delta_e$
To: Δu
To: Δw
To: Δq
To: $\Delta\theta$
$\times 10^{-3}$
Amplitude
Time (seconds)


ClosedLoop Ailerons Step Response
From: $\Delta\delta_a$
$\times 10^{-3}$
To: $\Delta\beta$
To: Δp
To: Δr
To: $\Delta\phi$
Amplitude
Time (seconds)


ClosedLoop Rudder Step Response
From: $\Delta\delta_r$
To: $\Delta\beta$
To: Δp
To: Δr
To: $\Delta\phi$
Amplitude
0.06
0.04
0.02
0
0.02
0.01
0
0
-0.01
-0.02
-0.03
0.1
0.05
0
0
5
10
15
20
Time (seconds)


Longitudinal and Lateral OpenLoop Eigenvalues
Imag(λ)
Real(λ)
Zoomed Plot Near Origin
Imag(λ)
Real(λ)
×10^-3
Short Period Mode
Long Period (Phugoid) Mode
Dutch Roll Mode
Roll Subsidence Mode
Spiral Mode



Longitudinal and Lateral ClosedLoop Eigenvalues
Imag(λ)
Real(λ)
Zoomed Plot Near Origin
Imag(λ)
Real(λ)
Short Period Mode
Long Period (Phugoid) Mode
Dutch Roll Mode
Roll Subsidence Mode
Spiral Mode

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 corrsponds 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