Linear Dynamic Analysis:

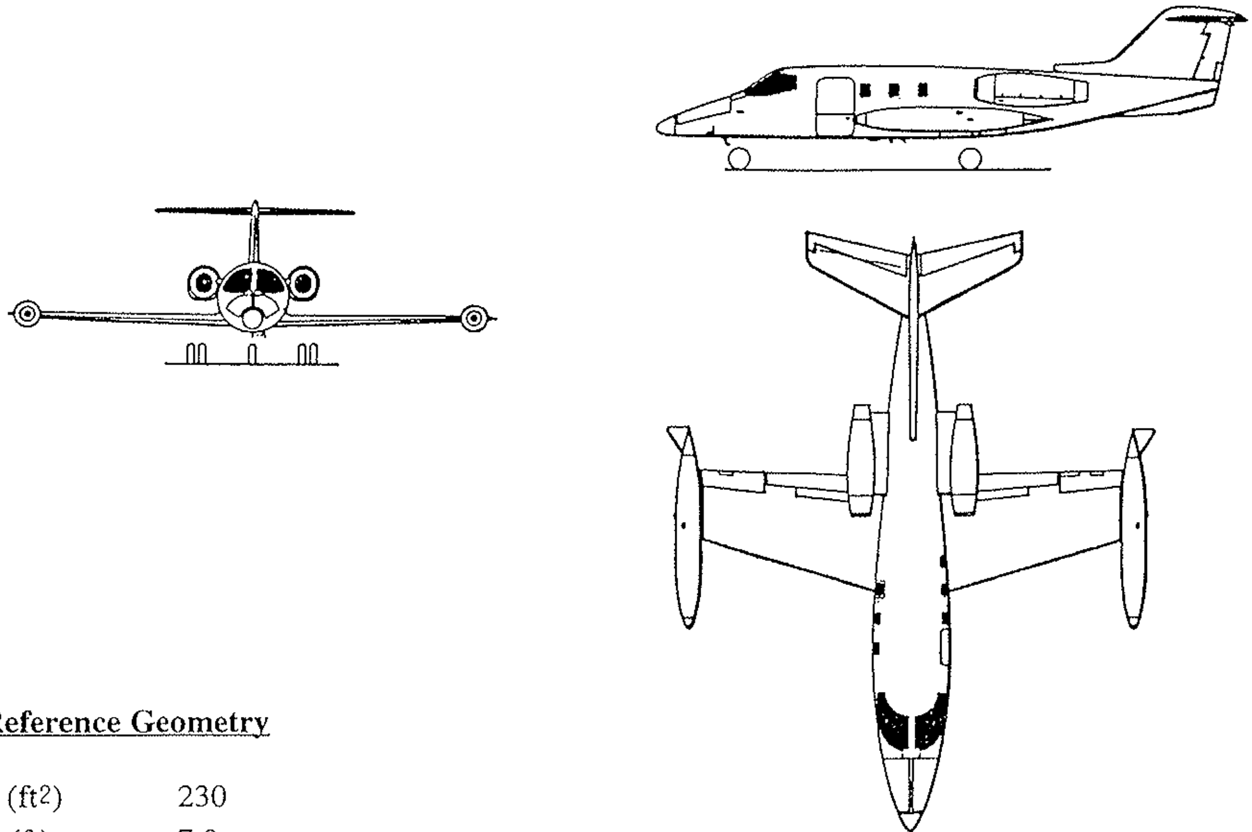
Longitudinal Fight Dynamics and Control:

Lear Jet Model 24

AAE 421

Table B7 Stability and Control Derivatives for Airplane G (Pages 522–528)

Three–view



Reference Geometry

S (ft^2)	230
$\bar{c}$ (ft)	7.0
b (ft)	34.0

Flight Condition Data	Approach	Cruise (Max Wht)	Cruise(Low Wht)
Altitude, h (ft)	0	40,000	40,000
Mach Number, M	0.152	0.7	0.7
TAS, U_1(ft/sec)	170	677	677
Dynamic pressure, $\bar{q}$ (lbs/ft^2)	34.3	134.6	134.6
C.G. location, fraction $\bar{c}$	0.32	0.32	0.32
Angle of attack, α_1 (deg)	5.0	2.7	1.5

Mass Data

W (lbs)	13,000	13,000	9,000
I_{xx_B} (slugft^2)	28,000	28,000	6,000
I_{yy_B} (slugft^2)	18,800	18,800	17,800
I_{zz_B} (slugft^2)	47,000	47,000	25,000
I_{xz_B} (slugft^2)	1,300	1,300	1,400

Table B7 (Continued) Stability and Control Derivatives for Airplane G (Pages 522–528)

Flight Condition	Approach	Cruise (Max Wht)	Cruise(Low Wht)
Steady State Coefficients			
C_{L_1}	1.64	0.41	0.28
C_{D_1}	0.2560	0.0335	0.0279
$C_{T_{x_1}}$	0.2560	0.0335	0.0279
C_{m_1}	0	0	0
$C_{m_{T_1}}$	0	0	0
Longitudinal Coefficients and Stability Derivatives (Stability Axes, Dimensionless)			
C_{D_0}	0.0431	0.0216	0.0216
C_{D_u}	0	0.104	0.104
C_{D_α}	1.06	0.30	0.22
$C_{T_{x_u}}$	–0.60	–0.07	–0.07
C_{L_0}	1.2	0.13	0.13
C_{L_u}	0.04	0.40	0.28
C_{L_α}	5.04	5.84	5.84
C_{L_α}	1.6	2.2	2.2
C_{L_q}	4.1	4.7	4.7
C_{m_0}	0.047	0.050	0.050
C_{m_u}	–0.01	0.050	0.070
C_{m_α}	–0.66	–0.64	–0.64
$C_{m_{\dot{\alpha}}}$	–5.0	–6.7	–6.7
C_{m_q}	–13.5	–15.5	–15.5
$C_{m_{T_u}}$	0.006	–0.003	–0.003
$C_{m_{T_\alpha}}$	0	0	0
Longitudinal Control and Hinge Moment Derivatives (Stability Axes, 1/rad)			
$C_{D_{\delta_e}}$	0	0	0
$C_{L_{\delta_e}}$	0.40	0.46	0.46
$C_{m_{\delta_e}}$	–0.98	–1.24	–1.24
$C_{D_{i_h}}$	0	0	0
$C_{L_{i_h}}$	0.85	0.94	0.94
$C_{m_{i_h}}$	–2.1	–2.5	–2.5

Table B7 (Continued) Stability and Control Derivatives for Airplane G (Pages 522–528)

Flight Condition	Approach	Cruise (Max Wht)	Cruise(Low Wht)
Longitudinal Control and Hinge Moment Derivatives: Cont'd (Stability Axes, 1/rad)			
C_{h_α}	–0.105	–0.132	–0.132
$C_{h_{\delta_e}}$	–0.378	–0.476	–0.476
Lateral–Directional Stability Derivatives (Stability Axes, Dimensionless)			
C_{l_β}	–0.173	–0.110	–0.100
C_{l_p}	–0.390	–0.450	–0.450
C_{l_r}	0.450	0.160	0.140
C_{y_β}	–0.730	–0.730	–0.730
C_{y_p}	0	0	0
C_{y_r}	0.400	0.400	0.400
C_{n_β}	0.150	0.127	0.124
$C_{n_{T_\beta}}$	0	0	0
C_{n_p}	–0.130	–0.008	–0.022
C_{n_r}	–0.260	–0.200	–0.200
Lateral–Directional Control and Hinge Moment Derivatives (Stability Axes, Dimensionless)			
$C_{l_{\delta_a}}$	0.149	0.178	0.178
$C_{l_{\delta_r}}$	0.014	0.019	0.021
$C_{y_{\delta_a}}$	0	0	0
$C_{y_{\delta_r}}$	0.140	0.140	0.140
$C_{n_{\delta_a}}$	–0.050	–0.020	–0.020
$C_{n_{\delta_r}}$	–0.074	–0.074	–0.074
$C_{h_{\alpha_a}}$	???	???	???
$C_{h_{\delta_a}}$	???	???	???
$C_{h_{\beta_r}}$	???	???	???
$C_{h_{\delta_r}}$	???	???	???

4

Table 5.1 Definition of Longitudinal, Dimensional Stability Derivatives

Derivative	Units	Derivative	Units
$X_u = \frac{-\bar{q}_1 S(C_{D_u} + 2C_{D_1})}{mU_1}$	$\frac{ft/sec^2}{ft/sec}$	$M_u = \frac{\bar{q}_1 S\bar{c}(C_{m_u} + 2C_{m_1})}{I_{yy}U_1}$	$\frac{rad/sec^2}{ft/sec}$
$X_{T_u} = \frac{\bar{q}_1 S(C_{T_{x_u}} + 2C_{T_{x_1}})}{mU_1}$	$\frac{ft/sec^2}{ft/sec}$	$M_{T_u} = \frac{\bar{q}_1 S\bar{c}(C_{m_{T_u}} + 2C_{m_{T_1}})}{I_{yy}U_1}$	$\frac{rad/sec^2}{ft/sec}$
$X_\alpha = \frac{-\bar{q}_1 S(C_{D_\alpha} - C_{L_1})}{m}$	$\frac{ft/sec^2}{rad}$	$M_\alpha = \frac{\bar{q}_1 S\bar{c}C_{m_\alpha}}{I_{yy}}$	$\frac{rad/sec^2}{rad}$
$X_{\delta_e} = \frac{-\bar{q}_1 SC_{D_{\delta_e}}}{m}$	$\frac{ft/sec^2}{rad}$	$M_{T_\alpha} = \frac{\bar{q}_1 S\bar{c}C_{m_{T_\alpha}}}{I_{yy}}$	$\frac{rad/sec^2}{rad/sec}$
$Z_u = \frac{-\bar{q}_1 S(C_{L_u} + 2C_{L_1})}{mU_1}$	$\frac{ft/sec^2}{ft/sec}$	$M_{\dot{\alpha}} = \frac{\bar{q}_1 S\bar{c}^2 C_{m_{\dot{\alpha}}}}{2I_{yy}U_1}$	$\frac{rad/sec^2}{rad/sec}$
$Z_\alpha = \frac{-\bar{q}_1 S(C_{L_\alpha} + C_{D_1})}{m}$	$\frac{ft/sec^2}{rad}$	$M_q = \frac{\bar{q}_1 S\bar{c}^2 C_{m_q}}{2I_{yy}U_1}$	$\frac{rad/sec^2}{rad/sec}$
$Z_{\dot{\alpha}} = \frac{-\bar{q}_1 S\bar{c}C_{L_{\dot{\alpha}}}}{2mU_1}$	$\frac{ft/sec^2}{rad/sec}$	$M_{\delta_e} = \frac{\bar{q}_1 S\bar{c}C_{m_{\delta_e}}}{I_{yy}}$	$\frac{rad/sec^2}{rad}$
$Z_q = \frac{-\bar{q}_1 S\bar{c}C_{L_q}}{2mU_1}$	$\frac{ft/sec^2}{rad/sec}$		
$Z_{\delta_e} = \frac{-\bar{q}_1 SC_{L_{\delta_e}}}{m}$	$\frac{ft/sec^2}{rad}$		

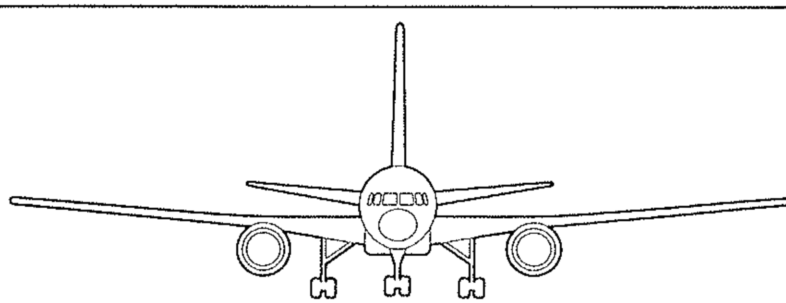


Table 5.2 Development of the Perturbed Longitudinal Equations of Motion with Dimensional Stability Derivatives in Matrix Format

$\gamma_1 = \theta_1 - \alpha_1$ page 56

$$\dot{u} = -g\theta\cos\theta_1 + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_e}\delta_e \qquad (5.30a)$$

$$U_1\dot{\alpha} - U_1\dot{\theta} = -g\theta\sin\theta_1 + Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q\dot{\theta} + Z_{\delta_e}\delta_e \qquad (5.30b)$$

$$\ddot{\theta} = M_u u + M_{T_u} u + M_\alpha \alpha + M_{T_\alpha}\alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q\dot{\theta} + M_{\delta_e}\delta_e \qquad (5.30c)$$

Laplace transforming Eqns (5.30) for zero initial conditions:

$$(s - X_u - X_{T_u})u(s) \quad - X_\alpha \alpha(s) \quad g\cos\theta_1\theta(s) \quad = X_{\delta_e}\delta_e(s) \qquad (5.31a)$$

$$- Z_u u(s) \quad \{s(U_1 - Z_{\dot{\alpha}}) - Z_\alpha\}\alpha(s) \quad \{-(Z_q + U_1)s + g\sin\theta_1\}\theta(s) \quad = Z_{\delta_e}\delta_e(s) \qquad (5.31b)$$

$$-(M_u + M_{T_u})u(s) \quad -\{M_{\dot{\alpha}}s + M_\alpha + M_{T_\alpha}\}\alpha(s) \quad (s^2 - M_q s)\theta(s) \quad = M_{\delta_e}\delta_e(s) \qquad (5.31c)$$

Writing Eqns (5.31) in matrix and transfer function format:

$$\underbrace{\begin{bmatrix} (s - X_u - X_{T_u}) & -X_\alpha & g\cos\theta_1 \\ -Z_u & \{s(U_1 - Z_{\dot{\alpha}}) - Z_\alpha\} & \{-(Z_q + U_1)s + g\sin\theta_1\} \\ -(M_u + M_{T_u}) & -(M_{\dot{\alpha}}s + M_\alpha + M_{T_\alpha}) & (s^2 - M_q s) \end{bmatrix}}_{\text{System Matrix}} \underbrace{\begin{Bmatrix} \dfrac{u(s)}{\delta_e(s)} \\ \dfrac{\alpha(s)}{\delta_e(s)} \\ \dfrac{\theta(s)}{\delta_e(s)} \end{Bmatrix}}_{\text{Transfer Function Matrix}} = \underbrace{\begin{Bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \end{Bmatrix}}_{\text{Control Power Matrix}} \qquad (5.32)$$

MATLAB SCRIPT LongSC.m
(see also LongSC2.m that plots time histories)

(http://cobweb.ecn.purdue.edu/~andrisan/Courses/AAE421 Fall 2006 /AAE421 Buffer F06/Dynamic%20Analysis%20of%20LearJet%2024/)

```
disp('*** Start Here ***')
echo on
%
% Longitudinal Airplane Stability and Control Analysis
%
% Ref: Airplane Flight Dyanmics and Automatic Flight Controls
%      Part 1, 1994
%      by Jan Roskam
%
% See Data input format of Appendix B
%
% Sample data for LearJet, approach configuration, pages 522-
523.
%
% Aircraft and configuration
aircraft='Learjet 24';
configuration='approach';
%
% Reference Geometry
S=230;       % Wing area, ft*ft
cbar=7;      % wing mean geometric chord, ft
span=34;     % Wing span, ft
%
% Steady State (Trim) Flight Condition
h=0;         % Altitude, ft
M=.152;      % Mach number, nondimensional
U1=170;      % True airspeed, ft/sec
qbar=34.3;   % Dynamic pressure, lbf/ft*ft
Xcg=.32;     % X-location of center of gravity, fraction of cbar
alpha1=5;    % Angle of attack, degree
%
% Mass data
W=13000;     % Weight, pounds (lbf)
Ixxb=28000; % X-direction moment of inertia, slug*ft*ft
Iyyb=18800; % Y-direction moment of inertia, slug*ft*ft
Izzb=47000; % Z-direction moment of inertia, slug*ft*ft
Ixzb=1300;  % XZ-direction product of inertia, slug*ft*ft
%
% Steady State Coefficients
```

```
CL1=1.64;    % Lift coefficient, nondimensional
CD1=.256;    % Drag coefficient, nondimensional
CTx1=.256;   % Thrust coefficient, nondimensional
Cm1=0;       % Aerodynamic Pitching Moment coefficient,
nondimensional
CmT1=0;      % Thrust Pitching Moment coefficient, nondimensional
%
%Longitudinal Coefficients and Stability Derivatives
CD0=.043;
CDu=0;
CDalpha=1.06;
CTxu=-.6;
CL0=1.2;
CLu=.04;
CLalpha=5.04;   % Lift curve slope
CLalphadot=1.6;
CLq=4.1;
CM0=.047;
Cmu=-.01;
Cmalpha=-.66;   % Static longitudianl stabiity derivative
Cmalphadot=-5;  % Lag of downwash stability derivative
Cmq=-13.5;      % Swish effect in pitch
CmTu=.006;
CmTalpha=0;
%
%Longitudinal Control Derivatives
CDdeltae=0;     % Drag from elevator
CLdeltae=.4;    % Lift from elevator
Cmdeltae=-.98;  % Pitching moment from elevator
CDih=0;         % Drag from incidence of the horizontal tail
CLih=.85;       % Lift from incidence of the horizontal tail
Cmih=-2.1;      % Pitlching moment from incidence of the
horizontal tail
%
% Miscellaneous inputs
g=32.17;        % Acceleration of gravity, ft/sec*sec
gamma1=0        % Steady state flight path angle, deg, see
figure page 66
%                 angle between horizontal and Xs axis (Wind
at trim)
%
% Computation of Dimensional Stability and Control Derivatives
%
% Preliminary calculations
% Stability axis inertia data, see page 346, eqn 5.94
Iyys=Iyyb;  % These are the same since that rotation is angle
alpha1
%               about the Y-axis
```

```
mass=W/g    % mass, slugs
theta1=gamma1+alpha1;   % Steady state
d2r=pi/180;
r2d=180/pi;
gamma1rad=gamma1*d2r;

% See page 319
Xu=-qbar*S*(CDu+2*CD1)/(mass*U1)     % 1/sec
XTu=qbar*S*(CTxu+2*CTx1)/(mass*U1)   % 1/sec
Xalpha=-qbar*S*(CDalpha-CL1)/mass    % ft/sec*sec
Xdeltae=-qbar*S*CDdeltae/mass        % ft/sec*sec
%
Zu=-qbar*S*(CLu+2*CL1)/(mass*U1)     % 1/sec
Zalpha=-qbar*S*(CLalpha+CD1)/mass    % ft/sec*sec
Zalphadot=-qbar*S*cbar*CLalphadot/(2*mass*U1)   % ft/sec
Zq=-qbar*S*cbar*CLq/(2*mass*U1)      % ft/sec
Zdeltae=-qbar*S*CLdeltae/mass        % ft/sec*sec
%
Mu=qbar*S*cbar*(Cmu+2*Cm1)/(Iyys*U1)     % 1/ft/sec
MTu=qbar*S*cbar*(CmTu+2*CmT1)/(Iyys*U1) % 1/ft/sec
Malpha=qbar*S*cbar*Cmalpha/Iyys          % 1/sec*sec
MTalpha=qbar*S*cbar*CmTalpha/Iyys        % 1/sec*sec
Malphadot=qbar*S*cbar*cbar*Cmalphadot/(2*Iyys*U1)   % 1/sec
Mq=qbar*S*cbar*cbar*Cmq/(2*Iyys*U1)      % 1/sec
Mdeltae=qbar*S*cbar*Cmdeltae/Iyys        % 1/sec*sec

%
% Computation of Primed Dimensional Stability and Control
Derivatives
%
XuP=Xu+XTu;
XalphaP=Xalpha;
XthetaP=-g*cos(gamma1rad);
XdeltaeP=Xdeltae;
%
ZuP=Zu/(U1-Zalphadot);
ZalphaP=Zalpha/(U1-Zalphadot);
ZqP=(Zq+U1)/(U1-Zalphadot);
ZthetaP=-g*sin(gamma1rad)/(U1-Zalphadot);
ZdeltaeP=Zdeltae/(U1-Zalphadot);
%
MuP=Mu+MTu+Malphadot*Zu/(U1-Zalphadot)
MalphaP=Malpha+MTalpha+Malphadot*Zalpha/(U1-Zalphadot)
MqP=Mq+Malphadot*(Zq+U1)/(U1-Zalphadot)
MthetaP=-Malphadot*g*sin(gamma1rad)/(U1-Zalphadot)
MdeltaeP=Mdeltae+Malphadot*Zdeltae/(U1-Zalphadot)
%
% Assemble the A and B matrices
```

```
% X=[u(ft/sec) alpha(rad) q(rad/sec) theta(rad)]'
%
A=[XuP XalphaP   0 XthetaP;
   ZuP ZalphaP ZqP ZthetaP;
   MuP MalphaP MqP MthetaP;
     0       0   1       0];
B=[XdeltaeP ZdeltaeP MdeltaeP 0]';
C=[1  0     0   0;
   0  r2d   0   0;
   0  0    r2d  0;
   0  0     0  r2d];
D=[0 0 0 0]';
%
% Analyze the linear equations of motion
%
sys=ss(A,B,C,D);
set(sys,'statename',{'u(f/s)' 'alpha(r)' 'q(r/s)' 'theta(r)'})
set(sys,'inputname','deltae(r)')
set(sys,'outputname',{'u(f/s)' 'alpha(d)' 'q(d/s)' 'theta(d)'})
sys
[Wn,Z,Poles]=damp(sys)
tfsys=tf(sys)
zpksys=zpk(sys)

echo off
```

OUTPUT FROM THE SCRIPT LongSC.m

```
*** Start Here ***
%
% Longitudinal Airplane Stability and Control Analysis
%
% Ref: Airplane Flight Dyanmics and Automatic Flight Controls
%      Part 1, 1994
%      by Jan Roskam
%
% See Data input format of Appendix B
%
% Sample data for LearJet, approach configuration, pages 522-
523.
%
% Aircraft and configuration
aircraft='Learjet 24';
configuration='approach';
%
% Reference Geometry
S=230;          % Wing area, ft*ft
cbar=7;         % wing mean geometric chord, ft
span=34;  % Wing span, ft
%
% Steady State (Trim) Flight Condition
h=0;      % Altitude, ft
M=.152;         % Mach number, nondimensional
U1=170;         % True airspeed, ft/sec
qbar=34.3;      % Dynamic pressure, lbf/ft*ft
Xcg=.32;  % X-location of center of gravity, fraction of cbar
alpha1=5; % Angle of attack, degree
%
% Mass data
W=13000;  % Weight, pounds (lbf)
Ixxb=28000;     % X-direction moment of inertia, slug*ft*ft
Iyyb=18800;     % Y-direction moment of inertia, slug*ft*ft
Izzb=47000;     % Z-direction moment of inertia, slug*ft*ft
Ixzb=1300;      % XZ-direction product of inertia, slug*ft*ft
%
% Steady State Coefficients
CL1=1.64; % Lift coefficient, nondimensional
CD1=.256; % Drag coefficient, nondimensional
CTx1=.256;      % Thrust coefficient, nondimensional
Cm1=0;          % Aerodynamic Pitching Moment coefficient,
nondimensional
CmT1=0;         % Thrust Pitching Moment coefficient,
nondimensional
```

```
%
%Longitudinal Coefficients and Stability Derivatives
CD0=.043;
CDu=0;
CDalpha=1.06;
CTxu=-.6;
CL0=1.2;
CLu=.04;
CLalpha=5.04;  % Lift curve slope
CLalphadot=1.6;
CLq=4.1;
CM0=.047;
Cmu=-.01;
Cmalpha=-.66;  % Static longitudianl stabiity derivative
Cmalphadot=-5; % Lag of downwash stability derivative
Cmq=-13.5;          % Swish effect in pitch
CmTu=.006;
CmTalpha=0;
%
%Longitudinal Control Derivatives
CDdeltae=0;         % Drag from elevator
CLdeltae=.4;   % Lift from elevator
Cmdeltae=-.98; % Pitching moment from elevator
CDih=0;             % Drag from incidence of the horizontal tail
CLih=.85;      % Lift from incidence of the horizontal tail
Cmih=-2.1;          % Pitlching moment from incidence of the
horizontal tail
%
% Miscellaneous inputs
g=32.17;       % Acceleration of gravity, ft/sec*sec
gamma1=0       % Steady state flight path angle, deg, see figure
page 66
gamma1 =0
%                        angle between horizontal and Xs axis
(Wind at trim)
%
% Computation of Dimensional Stability and Control Derivatives
%
% Preliminary calculations
% Stability axis inertia data, see page 346, eqn 5.94
Iyys=Iyyb;     % These are the same since that rotation is angle
alpha1
%                   about the Y-axis
mass=W/g  % mass, slugs
mass =        404.1
theta1=gamma1+alpha1;    % Steady state
d2r=pi/180;
r2d=180/pi;
```

```
gamma1rad=gamma1*d2r;
% See page 319
Xu=-qbar*S*(CDu+2*CD1)/(mass*U1)   % 1/sec
Xu =    -0.058796
XTu=qbar*S*(CTxu+2*CTx1)/(mass*U1) % 1/sec
XTu =    -0.010106
Xalpha=-qbar*S*(CDalpha-CL1)/mass  % ft/sec*sec
Xalpha =       11.323
Xdeltae=-qbar*S*CDdeltae/mass      % ft/sec*sec
Xdeltae =     0
%
Zu=-qbar*S*(CLu+2*CL1)/(mass*U1)   % 1/sec
Zu =     -0.38126
Zalpha=-qbar*S*(CLalpha+CD1)/mass  % ft/sec*sec
Zalpha =      -103.39
Zalphadot=-qbar*S*cbar*CLalphadot/(2*mass*U1)     % ft/sec
Zalphadot =     -0.64309
Zq=-qbar*S*cbar*CLq/(2*mass*U1)         % ft/sec
Zq =      -1.6479
Zdeltae=-qbar*S*CLdeltae/mass      % ft/sec*sec
Zdeltae =      -7.8089
%
Mu=qbar*S*cbar*(Cmu+2*Cm1)/(Iyys*U1)    % 1/ft/sec
Mu =  -0.00017279
MTu=qbar*S*cbar*(CmTu+2*CmT1)/(Iyys*U1) % 1/ft/sec
MTu =   0.00010367
Malpha=qbar*S*cbar*Cmalpha/Iyys                  % 1/sec*sec
Malpha =      -1.9387
MTalpha=qbar*S*cbar*CmTalpha/Iyys       % 1/sec*sec
MTalpha =     0
Malphadot=qbar*S*cbar*cbar*Cmalphadot/(2*Iyys*U1) % 1/sec
Malphadot =     -0.30238
Mq=qbar*S*cbar*cbar*Cmq/(2*Iyys*U1)              % 1/sec
Mq =     -0.81642
Mdeltae=qbar*S*cbar*Cmdeltae/Iyys       % 1/sec*sec
Mdeltae =      -2.8786

%
% Computation of Primed Dimensional Stability and Control
Derivatives
%
XuP=Xu+XTu;
XalphaP=Xalpha;
XthetaP=-g*cos(gamma1rad);
XdeltaeP=Xdeltae;
%
ZuP=Zu/(U1-Zalphadot);
ZalphaP=Zalpha/(U1-Zalphadot);
```

```
ZqP=(Zq+U1)/(U1-Zalphadot);
ZthetaP=-g*sin(gamma1rad)/(U1-Zalphadot);
ZdeltaeP=Zdeltae/(U1-Zalphadot);
%
MuP=Mu+MTu+Malphadot*Zu/(U1-Zalphadot)
MuP =   0.00060647
MalphaP=Malpha+MTalpha+Malphadot*Zalpha/(U1-Zalphadot)
MalphaP =      -1.7555
MqP=Mq+Malphadot*(Zq+U1)/(U1-Zalphadot)
MqP =      -1.1147
MthetaP=-Malphadot*g*sin(gamma1rad)/(U1-Zalphadot)
MthetaP =     0
MdeltaeP=Mdeltae+Malphadot*Zdeltae/(U1-Zalphadot)
MdeltaeP =      -2.8648
%
% Assemble the A and B matrices
% X=[u(ft/sec) alpha(rad) q(rad/sec) theta(rad)]'
%
A=[XuP XalphaP   0 XthetaP;
   ZuP ZalphaP ZqP ZthetaP;
   MuP MalphaP MqP MthetaP;
     0       0   1       0];
B=[XdeltaeP ZdeltaeP MdeltaeP 0]';
C=[1  0     0   0;
   0  r2d   0   0;
   0  0    r2d  0;
   0  0     0  r2d];
D=[0 0 0 0]';
%
% Analyze the linear equations of motion
%
sys=ss(A,B,C,D);
set(sys,'statename',{'u(f/s)' 'alpha(r)' 'q(r/s)' 'theta(r)'})
set(sys,'inputname','deltae(r)')
set(sys,'outputname',{'u(f/s)' 'alpha(d)' 'q(d/s)' 'theta(d)'})
sys
a =
                u(f/s)   alpha(r)     q(r/s)   theta(r)
   u(f/s)      -0.0689      11.32          0     -32.17
   alpha(r)  -0.002234    -0.6059     0.9866         -0
   q(r/s)    0.0006065     -1.755     -1.115          0
   theta(r)          0          0          1          0

b =
             deltae(r)
   u(f/s)           -0
   alpha(r)   -0.04576
   q(r/s)       -2.865
```

```
   theta(r)          0

c =
                 u(f/s)  alpha(r)    q(r/s)  theta(r)
   u(f/s)             1         0         0         0
   alpha(d)           0      57.3         0         0
   q(d/s)             0         0      57.3         0
   theta(d)           0         0         0      57.3

d =
               deltae(r)
   u(f/s)              0
   alpha(d)            0
   q(d/s)              0
   theta(d)            0

Continuous-time model.
[Wn,Z,Poles]=damp(sys)
Wn =       0.23896
     0.23896
       1.5546
       1.5546

Z =        0.092774
     0.092774
     0.56131
     0.56131

Poles =       -0.022169 +    0.23793i
    -0.022169 -    0.23793i
      -0.87259 +      1.2866i
      -0.87259 -      1.2866i
tfsys=tf(sys)

Transfer function from input "deltae(r)" to output...
                        -0.5182 s^2 + 59.58 s + 53.25
 u(f/s):  -------------------------------------------
          s^4 + 1.79 s^3 + 2.551 s^2 + 0.2068 s + 0.138

                    -2.622 s^3 - 165 s^2 - 11.36 s - 11.85
 alpha(d):  -------------------------------------------
            s^4 + 1.79 s^3 + 2.551 s^2 + 0.2068 s + 0.138

          -164.1 s^3 - 106.2 s^2 - 10.71 s + 1.109e-15
 q(d/s):  -------------------------------------------
          s^4 + 1.79 s^3 + 2.551 s^2 + 0.2068 s + 0.138

                     -164.1 s^2 - 106.2 s - 10.71
```

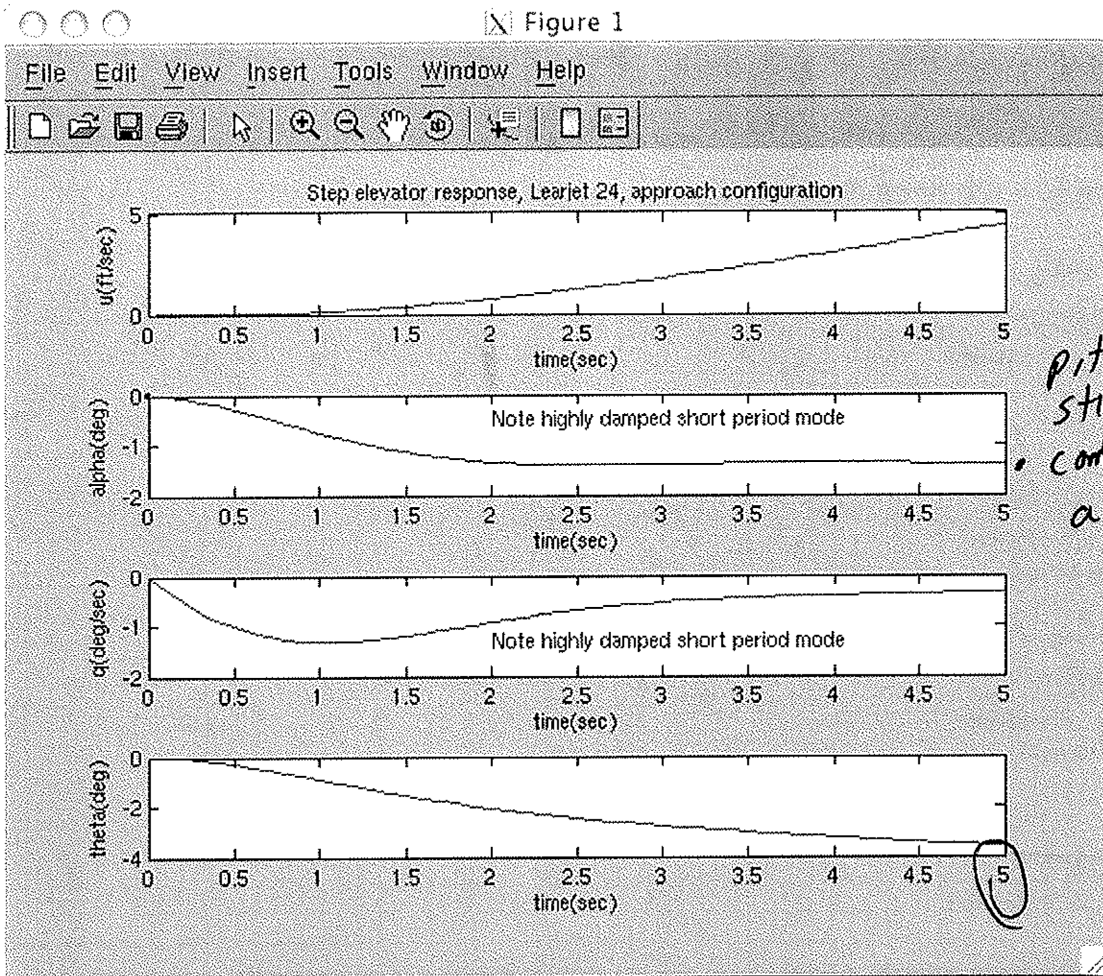
Handwritten annotations: Wn: 0.23896 (pair) } Phugoid; 1.5546 (pair) } S.P. — Z: 0.092774 (pair) } phugoid (long period); 0.56131 (pair) } S.P. — Poles: first pair } ; second pair } short period mode.

```
 theta(d):  ---------------------------------------------
            s^4 + 1.79 s^3 + 2.551 s^2 + 0.2068 s + 0.138

zpksys=zpk(sys)

Zero/pole/gain from input "deltae(r)" to output...
                        -0.51815 (s-115.9) (s+0.887)
 u(f/s):  --------------------------------------------------
          (s^2  + 0.04434s + 0.0571) (s^2  + 1.745s + 2.417)

                -2.6219 (s+62.88) (s^2  + 0.06776s + 0.07187)
 alpha(d):  --------------------------------------------------
            (s^2  + 0.04434s + 0.0571) (s^2  + 1.745s + 2.417)

                   -164.1414 s (s+0.5217) (s+0.125)
 q(d/s):  --------------------------------------------------
          (s^2  + 0.04434s + 0.0571) (s^2  + 1.745s + 2.417)

                      -164.1414 (s+0.125) (s+0.5217)
 theta(d):  --------------------------------------------------
            (s^2  + 0.04434s + 0.0571) (s^2  + 1.745s + 2.417)

echo off
>>
```

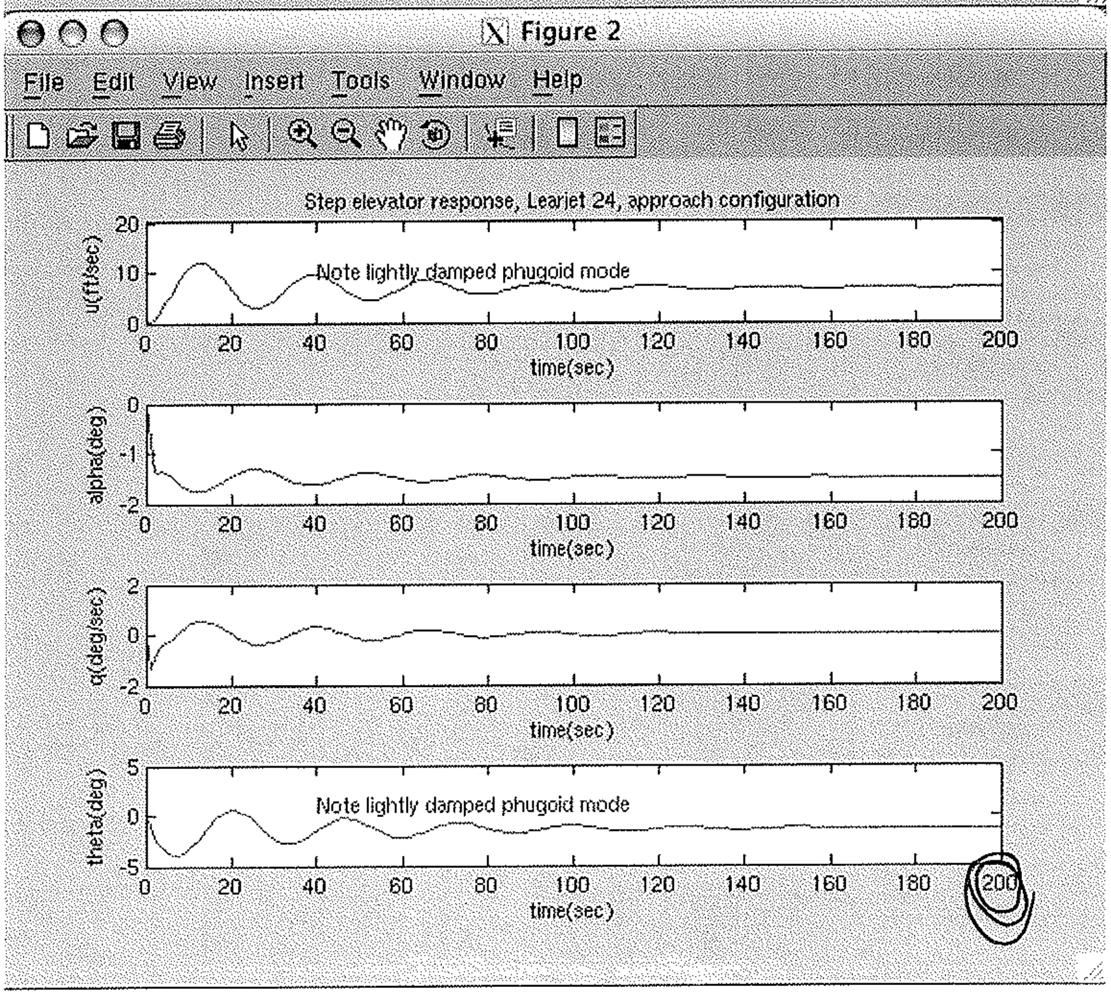
MATLAB SCRIPT LongSC2.m
(This script generates time responses and plots the results)

```
% You should run LongSC.m before running this script.
echo off
t=0:.1:5;                   % time scale to expose short period
mode
u=d2r*ones(length(t),1);    % deltae step of one degree
y=lsim(sys,u,t);
figure(1)
subplot(4,1,1)
plot(t,y(:,1))
xlabel('time(sec)')
ylabel('u(ft/sec)')
ts=['Step elevator response, ',aircraft,', ',configuration,'
configuration'];
title(ts)
```

```
subplot(4,1,2)
plot(t,y(:,2))
xlabel('time(sec)')
ylabel('alpha(deg)')
text(2,-.5,'Note highly damped short period mode')
subplot(4,1,3)
plot(t,y(:,3))
xlabel('time(sec)')
ylabel('q(deg/sec)')
text(2,-1.3,'Note highly damped short period mode')
subplot(4,1,4)
plot(t,y(:,4))
xlabel('time(sec)')
ylabel('theta(deg)')

t=0:1:200;                    % time scale to expose short period
mode
u=d2r*ones(length(t),1);      % deltae step of one degree
y=lsim(sys,u,t);
figure(2)
subplot(4,1,1)
plot(t,y(:,1))
xlabel('time(sec)')
ylabel('u(ft/sec)')
ts=['Step elevator response, ',aircraft,', ',configuration,'
configuration'];
title(ts)
text(40,10,'Note lightly damped phugoid mode')
subplot(4,1,2)
plot(t,y(:,2))
xlabel('time(sec)')
ylabel('alpha(deg)')
subplot(4,1,3)
plot(t,y(:,3))
xlabel('time(sec)')
ylabel('q(deg/sec)')
subplot(4,1,4)
plot(t,y(:,4))
xlabel('time(sec)')
ylabel('theta(deg)')
text(40,1.0,'Note lightly damped phugoid mode')
```



pitch stick
• commands angle of attack



Linear Dynamic Analysis: Part 2

Lateral-Directional Fight Dynamics and Control:

Lear Jet Model 24

AAE 421

Table 5.7 Definition of Lateral–Directional, Dimensional Stability Derivatives

$Y_\beta = \dfrac{\overline{q}_1 S C_{y_\beta}}{m}$	$\dfrac{ft/sec^2}{rad}$	$L_{\delta_r} = \dfrac{\overline{q}_1 S b C_{l_{\delta_r}}}{I_{xx}}$	$\dfrac{rad/sec^2}{rad}$
$Y_p = \dfrac{\overline{q}_1 S b C_{y_p}}{2mU_1}$	$\dfrac{ft/sec^2}{rad/sec}$	$N_\beta = \dfrac{\overline{q}_1 S b C_{n_\beta}}{I_{zz}}$	$\dfrac{rad/sec^2}{rad}$
$Y_r = \dfrac{\overline{q}_1 S b C_{y_r}}{2mU_1}$	$\dfrac{ft/sec^2}{rad/sec}$	$N_{T_\beta} = \dfrac{\overline{q}_1 S b C_{n_{T_\beta}}}{I_{zz}}$	$\dfrac{rad/sec^2}{rad}$
$Y_{\delta_a} = \dfrac{\overline{q}_1 S C_{y_{\delta_a}}}{m}$	$\dfrac{ft/sec^2}{rad}$	$N_p = \dfrac{\overline{q}_1 S b^2 C_{n_p}}{2I_{zz}U_1}$	$\dfrac{rad/sec^2}{rad/sec}$
$Y_{\delta_r} = \dfrac{\overline{q}_1 S C_{y_{\delta_r}}}{m}$	$\dfrac{ft/sec^2}{rad}$	$N_r = \dfrac{\overline{q}_1 S b^2 C_{n_r}}{2I_{zz}U_1}$	$\dfrac{rad/sec^2}{rad/sec}$
$L_\beta = \dfrac{\overline{q}_1 S b C_{l_\beta}}{I_{xx}}$	$\dfrac{rad/sec^2}{rad}$	$N_{\delta_a} = \dfrac{\overline{q}_1 S b C_{n_{\delta_a}}}{I_{zz}}$	$\dfrac{rad/sec^2}{rad}$
$L_p = \dfrac{\overline{q}_1 S b^2 C_{l_p}}{2I_{xx}U_1}$	$\dfrac{rad/sec^2}{rad/sec}$	$N_{\delta_r} = \dfrac{\overline{q}_1 S b C_{n_{\delta_r}}}{I_{zz}}$	$\dfrac{rad/sec^2}{rad}$
$L_r = \dfrac{\overline{q}_1 S b^2 C_{l_r}}{2I_{xx}U_1}$	$\dfrac{rad/sec^2}{rad/sec}$		
$L_{\delta_a} = \dfrac{\overline{q}_1 S b C_{l_{\delta_a}}}{I_{xx}}$	$\dfrac{rad/sec^2}{rad}$		

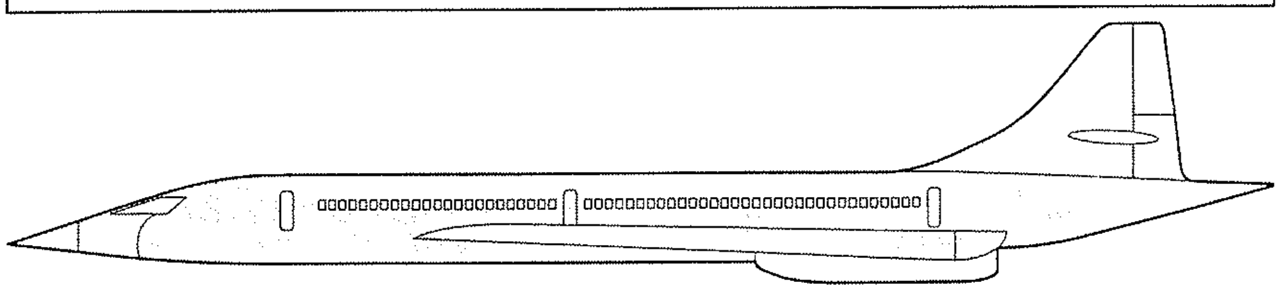


Table 5.8 Development of the Perturbed Lateral–Directional Equations of Motion with Dimensional Stability Derivatives in Matrix Format

$\gamma_1 = \theta_1 - \alpha_1$ page 66

$$U_1\dot{\beta} + U_1\dot{\psi} = g\phi\cos\theta_1 + Y_\beta\beta + Y_p\dot{\phi} + Y_r\dot{\psi} + Y_{\delta_a}\delta_a + Y_{\delta_r}\delta_r \tag{5.96a}$$

$$\ddot{\phi} - \overline{A}_1\ddot{\psi} = L_\beta\beta + L_p\dot{\phi} + L_r\dot{\psi} + L_{\delta_a}\delta_a + L_{\delta_r}\delta_r \tag{5.96b}$$

$$\ddot{\psi} - \overline{B}_1\ddot{\psi} = N_\beta\beta + N_{T_\beta}\beta + N_p\dot{\phi} + N_r\dot{\psi} + N_{\delta_a}\delta_a + N_{\delta_r}\delta_r \tag{5.96c}$$

NOTE: $\overline{A}_1 = \dfrac{I_{xz}}{I_{xx}}$ and $\overline{B}_1 = \dfrac{I_{xz}}{I_{zz}}$

Laplace transforming Eqns (5.96) for zero initial conditions:

$$(sU_1 - Y_\beta)\beta(s) \quad - (sY_p + g\cos\theta_1)\phi(s) \quad s(U_1 - Y_r)\psi(s) = Y_\delta\delta(s) \tag{5.97a}$$

$$-L_\beta\beta(s) \quad + (s^2 - L_p s)\phi(s) \quad - (s^2\overline{A}_1 + sL_r)\psi(s) = L_\delta\delta(s) \tag{5.97b}$$

$$-(N_\beta + N_{T_\beta})\beta(s) \quad - (s^2\overline{B}_1 + N_p s)\phi(s) \quad (s^2 - sN_r)\psi(s) = N_\delta\delta(s) \tag{5.97c}$$

Writing Eqns (5.97) in matrix and transfer function format:

$$\begin{bmatrix} (sU_1 - Y_\beta) & -(sY_p + g\cos\theta_1) & s(U_1 - Y_r) \\ -L_\beta & (s^2 - L_p s) & -(s^2\overline{A}_1 + sL_r) \\ -N_\beta - N_{T_\beta} & -(s^2\overline{B}_1 + N_p s) & (s^2 - sN_r) \end{bmatrix} \begin{Bmatrix} \dfrac{\beta(s)}{\delta(s)} \\ \dfrac{\phi(s)}{\delta(s)} \\ \dfrac{\psi(s)}{\delta(s)} \end{Bmatrix} = \begin{Bmatrix} Y_\delta \\ L_\delta \\ N_\delta \end{Bmatrix} \tag{5.98}$$

System Matrix ← (first matrix); Transfer Function Matrix → (column vector); Control Power Matrix → (right-hand side)

21

MATLAB SCRIPT LatSC.m
(see also LatSC2.m that plots time histories)

(http://cobweb.ecn.purdue.edu/~andrisan/Courses/AAE421 Fall 2006 /AAE421 Buffer F06/Dynamic%20Analysis%20of%20LearJet%2024/)

```
disp(' ')
disp(' ')
disp(' ')
disp(' Start Here')
echo on
%
% Lateral-Directional Airplane Stability and Control
Analysis
%
% Ref: Airplane Flight Dyanmics and Automatic Flight
Controls
%      Part 1, 1994
%      by Jan Roskam
%
% See Data input format of Appendix B
%
% Sample data for LearJet, approach configuration,
pages 522-523.
%
% Aircraft and configuration
echo off
aircraft='Learjet 24';
configuration='approach';
disp([aircraft '  ' configuration])
%
% Reference Geometry
S=230        % Wing area, ft*ft
cbar=7       % wing mean geometric chord, ft
span=34      % Wing span, ft
%
% Steady State (Trim) Flight Condition
h=0;         % Altitude, ft
M=.152       % Mach number, nondimensional
U1=170       % True airspeed, ft/sec
```

```
qbar=34.3    % Dynamic pressure, lbf/ft*ft
Xcg=.32      % X-location of center of gravity, fraction
of cbar
alpha1=5     % Trim angle of attack, degree
gamma1=0    % Trim flight path angle, degree

%
% Mass data
W=13000      % Weight, pounds (lbf)
Ixxb=28000  % X-direction moment of inertia, slug*ft*ft
Iyyb=18800  % Y-direction moment of inertia, slug*ft*ft
Izzb=47000  % Z-direction moment of inertia, slug*ft*ft
Ixzb=1300   % XZ-direction product of inertia,
slug*ft*ft
%
% Lateral-Directional Coefficients and Stability
Derivatives
Clb=-.173
Clp=-.39
Clr=.45
Cyb=-.73
Cyp=0
Cyr=.4
Cnb=.15
Cntb=0
Cnp=-.13
%Cnr=-.26
%Cnr=-.50
%Cnr=-.40
%Cnr=-.30
%Cnr=-.20
%Cnr=-.70
%Cnr=-.90
%Cnr=-1.50
Cnr=0
%
% Lateral-Directional Control Derivatives
Clda=.149
Cldr=.014
Cyda=0
Cydr=.14
Cnda=-.05
```

```
Cndr=-.074
% Miscellaneous inputs
g=32.17;        % Acceleration of gravity, ft/sec*sec
d2r=pi/180;
r2d=180/pi;
%
% Computation of Dimensional Stability and Control
Derivatives
%
% Preliminary calculations
%
% Stability axis inertia data, see page 346, eqn 5.94
% Rotate axes through an angle alpha1 (see fig 5.15
page 347)
alpha1rad=alpha1*d2r;
csq=cos(alpha1rad)^2;
ssq=sin(alpha1rad)^2;
c2a=cos(2.*alpha1rad);
s2a=sin(2.*alpha1rad);
Ixxs=Ixxb*csq+Izzb*ssq-Ixzb*s2a
Izzs=Izzb*csq+Ixxb*ssq+Ixzb*s2a
Ixzs=0.5*(Ixxb-Izzb)*s2a+Ixzb*c2a

Abar=Ixzs/Ixxs;   % See page 349
Bbar=Ixzs/Izzs;

mass=W/g    % mass, slugs

% See page 348
qsdm=qbar*S/mass;
qsbd2mu=qsdm*span/(2*U1);
qsbdixx=qbar*S*span/Ixxs;
qsbbd2ixxu=qsbdixx*span/(2*U1);
qsbdizz=qbar*S*span/Izzs;
qsbbd2izzu=qsbdizz*span/(2*U1);
theta1rad=(alpha1+gamma1)*d2r;


Yb =qsdm*Cyb
Yp =qsbd2mu*Cyp
Yr =qsbd2mu*Cyr
Yda=qsdm*Cyda
```

```
Ydr=qsdm*Cydr

Lb =qsbdixx*Clb
Lp =qsbbd2ixxu*Clp
Lr =qsbbd2ixxu*Clr
Lda=qsbdixx*Clda
Ldr=qsbdixx*Cldr

Nb =qsbdizz*Cnb
Ntb=qsbdizz*Cntb
Np =qsbbd2izzu*Cnp
Nr =qsbbd2izzu*Cnr
Nda=qsbdizz*Cnda
Ndr=qsbdizz*Cndr

%
% Computation of system matrices
%

%
% Assemble the A and B matrices. See page 349.
% x=[beta(rad) p(rad/sec) r(rad/sec) phi(rad)
psi(rad)]'
% u=[aileron(rad) rudder(rad)]'
% E*xdot=a*x + b*u
a=[Yb Yp Yr-U1 g*cos(theta1rad) 0
   Lb Lp Lr       0              0
   Nb Np Nr       0              0
    0  1  0       0              0
    0  0  1       0              0];
b=[Yda Ydr
   Lda Ldr
   Nda Ndr
    0   0
    0   0];
E=[U1    0      0   0  0
   0     1    -Abar 0  0
   0   -Bbar    1   0  0
   0     0      0   1  0
   0     0      0   0  1];

Einv=inv(E);
```

```
A=Einv*a;
B=Einv*b;
% Outputs y=[beta(deg) p(deg/sec) r(deg/sec) phi(deg)
psi(deg)]
C=r2d*eye(5);
D=zeros(5,2);


%
% Analyze the linear equations of motion
%
sys=ss(A,B,C,D);
set(sys,'statename',{'beta(r)' 'p(r/s)' 'r(r/s)'
'phi(r)' 'psi(r)'})
set(sys,'inputname',{'deltaA(r)' 'deltaR(r)'})
set(sys,'outputname',{'beta(d)' 'p(d/s)' 'r(d/s)'
'phi(d)' 'psi(d)'})
sys
[Wn,Z,Poles]=damp(sys)
tfsys=tf(sys)
zpksys=zpk(sys)
```

OUTPUT FROM THE SCRIPT LongSC.m

```
 Start Here
%
% Lateral-Directional Airplane Stability and Control Analysis
%
% Ref: Airplane Flight Dyanmics and Automatic Flight Controls
%      Part 1, 1994
%      by Jan Roskam
%
% See Data input format of Appendix B
%
% Sample data for LearJet, approach configuration, pages 522-
523.
%
% Aircraft and configuration
echo off
Learjet 24  approach

S =230
cbar =7
span =34
M =0.152
U1 =170
qbar =34.3
Xcg =0.32
alpha1 =5
gamma1 =0
W =13000
Ixxb =28000
Iyyb =18800
Izzb =47000
Ixzb =1300
Clb =-0.173
Clp =-0.39
Clr =0.45
Cyb =-0.73
Cyp =0
Cyr =0.4
Cnb =0.15
Cntb =0
Cnp =-0.13
Cnr =0
Clda =0.149
Cldr =0.014
Cyda =0
```

Cydr =0.14
Cnda =-0.05
Cndr =-0.074
Ixxs =27919
Izzs =47081
Ixzs =-369.41
mass =404.1
Yb =-14.251
Yp =0
Yr =0.78089
Yda =0
Ydr =2.7331
Lb =-1.6621
Lp =-0.37469
Lr =0.43233
Lda =1.4315
Ldr =0.1345
Nb =0.85456
Ntb =0
Np =-0.074062
Nr =0
Nda =-0.28485
Ndr =-0.42158

a =

	beta(r)	p(r/s)	r(r/s)	phi(r)	psi(r)
beta(r)	-0.08383	0	-0.9954	0.1885	0
p(r/s)	-1.674	-0.3737	0.4324	0	0
r(r/s)	0.8677	-0.07113	-0.003393	0	0
phi(r)	0	1	0	0	0
psi(r)	0	0	1	0	0

b =

	deltaA(r)	deltaR(r)
beta(r)	0	0.01608
p(r/s)	1.435	0.1401
r(r/s)	-0.2961	-0.4227
phi(r)	0	0
psi(r)	0	0

c =

	beta(r)	p(r/s)	r(r/s)	phi(r)	psi(r)
beta(d)	57.3	0	0	0	0
p(d/s)	0	57.3	0	0	0
r(d/s)	0	0	57.3	0	0
phi(d)	0	0	0	57.3	0
psi(d)	0	0	0	0	57.3

```
d =
             deltaA(r)  deltaR(r)
   beta(d)           0          0
   p(d/s)            0          0
   r(d/s)            0          0
   phi(d)            0          0
   psi(d)            0          0

Continuous-time model.

Wn =

            0
     0.082912
      0.74719
       1.0604
       1.0604


Z =

           -1
           -1
            1
    -0.095868
    -0.095868


Poles =

            0
     0.082912
     -0.74719
      0.10165 +     1.0555i
      0.10165 -     1.0555i


Transfer function from input "deltaA(r)" to output...
                      16.89 s^2 + 27.64 s - 1.33
 beta(d):  -------------------------------------------------
           s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

            82.24 s^3 - 0.1623 s^2 + 42.18 s + 2.151e-16
 p(d/s):  -------------------------------------------------
          s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966
```

```
           -16.97 s^3 - 13.61 s^2 - 1.022 s + 8.1
 r(d/s):  ------------------------------------------------
          s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

                82.24 s^2 - 0.1623 s + 42.18
 phi(d):  ------------------------------------------------
          s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

             -16.97 s^3 - 13.61 s^2 - 1.022 s + 8.1
 psi(d):  ----------------------------------------------------
          s^5 + 0.461 s^4 + 0.9273 s^3 + 0.7595 s^2 - 0.06966 s

Transfer function from input "deltaR(r)" to output...
           0.9212 s^3 + 24.45 s^2 + 11.12 s - 1.969
 beta(d):  ------------------------------------------------
           s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

            8.027 s^3 - 11.31 s^2 - 33.95 s + 1.916e-16
 p(d/s):  ------------------------------------------------
          s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

             -24.22 s^3 - 10.85 s^2 - 0.3983 s - 6.328
 r(d/s):  ------------------------------------------------
          s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

                 8.027 s^2 - 11.31 s - 33.95
 phi(d):  ------------------------------------------------
          s^4 + 0.461 s^3 + 0.9273 s^2 + 0.7595 s - 0.06966

               -24.22 s^3 - 10.85 s^2 - 0.3983 s - 6.328
 psi(d):  ----------------------------------------------------
          s^5 + 0.461 s^4 + 0.9273 s^3 + 0.7595 s^2 - 0.06966 s


Zero/pole/gain from input "deltaA(r)" to output...
                 16.8883 (s+1.683) (s-0.04679)
 beta(d):  ----------------------------------------------
           (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

           82.2439 s (s^2  - 0.001973s + 0.5129)
 p(d/s):  ----------------------------------------------
          (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

           -16.9662 (s-0.5686) (s^2  + 1.371s + 0.8397)
 r(d/s):  ----------------------------------------------
          (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

             82.2439 (s^2  - 0.001973s + 0.5129)
```

```
phi(d):  ---------------------------------------------
         (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

           -16.9662 (s-0.5686) (s^2  + 1.371s + 0.8397)
 psi(d):  -----------------------------------------------
         s (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

Zero/pole/gain from input "deltaR(r)" to output...
               0.92115 (s+26.08) (s+0.6021) (s-0.1361)
 beta(d):  ---------------------------------------------
          (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

                    8.027 s (s-2.879) (s+1.469)
 p(d/s):  ---------------------------------------------
         (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

          -24.2179 (s+0.8183) (s^2  - 0.3701s + 0.3193)
 r(d/s):  ---------------------------------------------
         (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

                     8.027 (s+1.469) (s-2.879)
 phi(d):  ---------------------------------------------
         (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

           -24.2179 (s+0.8183) (s^2  - 0.3701s + 0.3193)
 psi(d):  -----------------------------------------------
         s (s+0.7472) (s-0.08291) (s^2  - 0.2033s + 1.124)

>>
```

MATLAB SCRIPT LatSC2.m

(This script generates time responses and plots the results)

```
% You should run LatSC.m before running this script.

d2r=pi/180;
echo off
ifig=0;
t1=0:.01:5;                     % time scale to expose roll
mode
mag=10*d2r;
uvector=doublet(t1,2,4,mag);    % deltaA doublet of ten
degree
zero=zeros(length(t1),1);
u=[uvector zero];
```

```
y=lsim(sys,u,t1);


ifig=ifig+1;
figure(ifig)
clf
subplot(3,1,1)
ts=['Aileron response, ',aircraft,', ',configuration,'
configuration'];
plot(t1,y(:,2))
title(ts)
xlabel('time(sec)')
ylabel('p(deg/sec)')
text2(.1,.15,'Note Da controls roll rate through the
roll mode.')
subplot(3,1,2)
plot(t1,y(:,3))
xlabel('time(sec)')
ylabel('r(deg/sec)')
subplot(3,1,3)
plot(t1,u(:,1))
xlabel('time(sec)')
ylabel('Da(rad)')
text2(.1,.15,'The roll mode involves primarily motion
variables p and phi.')


ifig=ifig+1;
figure(ifig)
clf
subplot(4,1,1)
plot(t1,y(:,1))
xlabel('time(sec)')
ylabel('beta(deg)')
ts=['Aileron response, ',aircraft,', ',configuration,'
configuration'];
title(ts)
subplot(4,1,2)
plot(t1,y(:,4))
xlabel('time(sec)')
```

```
ylabel('phi(deg)')
text2(.1,.15,'Note phi is the integral of roll rate.')
subplot(4,1,3)
plot(t1,y(:,5))
xlabel('time(sec)')
ylabel('psi(deg)')
subplot(4,1,4)
plot(t1,u(:,1))
xlabel('time(sec)')
ylabel('Da(rad)')
text2(.1,.15,'The roll mode involves primarily motion
variables p and phi.')
%%%%%%%%%%%

% Rudder doublet input
t2=0:.05:20;
mag=10*d2r;
uvector=doublet(t2,.2,.4,mag);  % deltaR doublet of ten
degree
zero=zeros(length(t2),1);
u=[zero uvector];
y=lsim(sys,u,t2);

ifig=ifig+1;
figure(ifig)
clf


subplot(3,1,1)
plot(t2,y(:,2))
ts=['Rudder doublet response, ',aircraft,',
',configuration,' configuration'];
title(ts)
xlabel('time(sec)')
ylabel('p(deg/sec)')
text2(.1,.15,'Note the slightly unstable dutch roll
mode.')
subplot(3,1,2)
plot(t2,y(:,3))
xlabel('time(sec)')
ylabel('r(deg/sec)')
text2(.1,.15,'The dutch roll mode involves all motion
```

```
variables.')
subplot(313)
plot(t2,u(:,2))
xlabel('time(sec)')
ylabel('Dr(rad)')
text2(.1,.15,'Rudder doublets excite the dutch roll
mode.')


ifig=ifig+1;
figure(ifig)
clf
subplot(4,1,1)
plot(t2,y(:,1))
xlabel('time(sec)')
ylabel('beta(deg)')
ts=['Rudder doublet response, ',aircraft,',
',configuration,' configuration'];
title(ts)
text2(.1,.15,'Note the slightly unstable dutch roll
mode.')
subplot(4,1,2)
plot(t2,y(:,4))
xlabel('time(sec)')
ylabel('phi(deg)')
text2(.1,.15,'The dutch rol mode involves all motion
variables.')
subplot(4,1,3)
plot(t2,y(:,5))
xlabel('time(sec)')
ylabel('psi(deg)')
subplot(414)
plot(t2,u(:,2))
xlabel('time(sec)')
ylabel('Dr(rad)')
text2(.1,.15,'Rudder doublets excite the dutch roll
mode.')

%%%%%%%%%%%%%%%


% initial condition response to expose spiral mode
```

```
t3=0:.1:100;
zero=zeros(length(t3),1);
u=[zero zero];
%[V,D] = EIG(A,B)
[V,D]=eig(A);
x0=V(:,3);
y=lsim(sys,u,t3,x0);
ifig=ifig+1;
figure(ifig)
clf
subplot(5,1,1)
ts=['Initial condition response to show spiral mode,
',aircraft,', ',configuration,' configuration'];
plot(t3,y(:,2))
title(ts)
xlabel('time(sec)')
ylabel('p(deg/sec)')
text2(.1,.15,'Note the spiral mode is unstable.')

subplot(5,1,2)
plot(t3,y(:,3))
xlabel('time(sec)')
ylabel('r(deg/sec)')
subplot(5,1,3)
plot(t3,y(:,1))
xlabel('time(sec)')
ylabel('beta(deg)')
subplot(5,1,4)
plot(t3,y(:,4))
xlabel('time(sec)')
ylabel('phi(deg)')
text2(.1,.15,'Note phi is the integral of roll rate.')
text2(.1,.35,'Note the spiraling motion, sometimes
called the death spiral.')
subplot(5,1,5)
plot(t3,y(:,5))
xlabel('time(sec)')
ylabel('psi(deg)')
%%%%%%%%%%%%
```

Linear Dynamic Analysis: Part 2 Continued

Lateral-Directional Fight Dynamics and Control:

Lear Jet Model 24

AAE 421

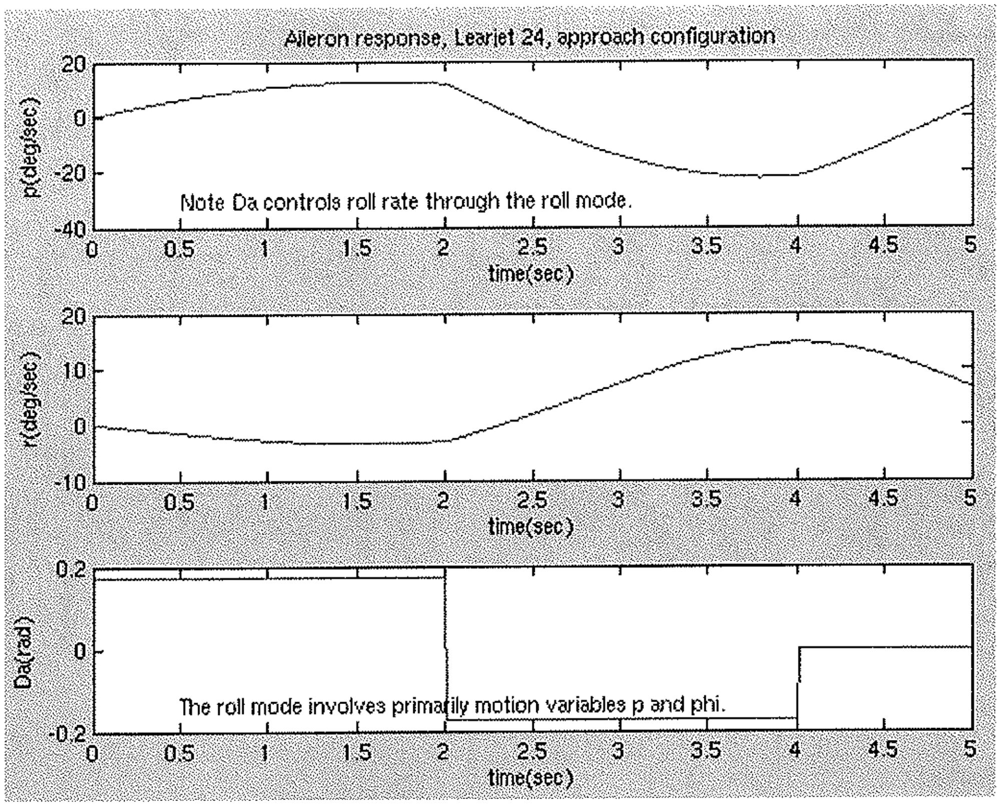
Learjet 24 Approach flight condition

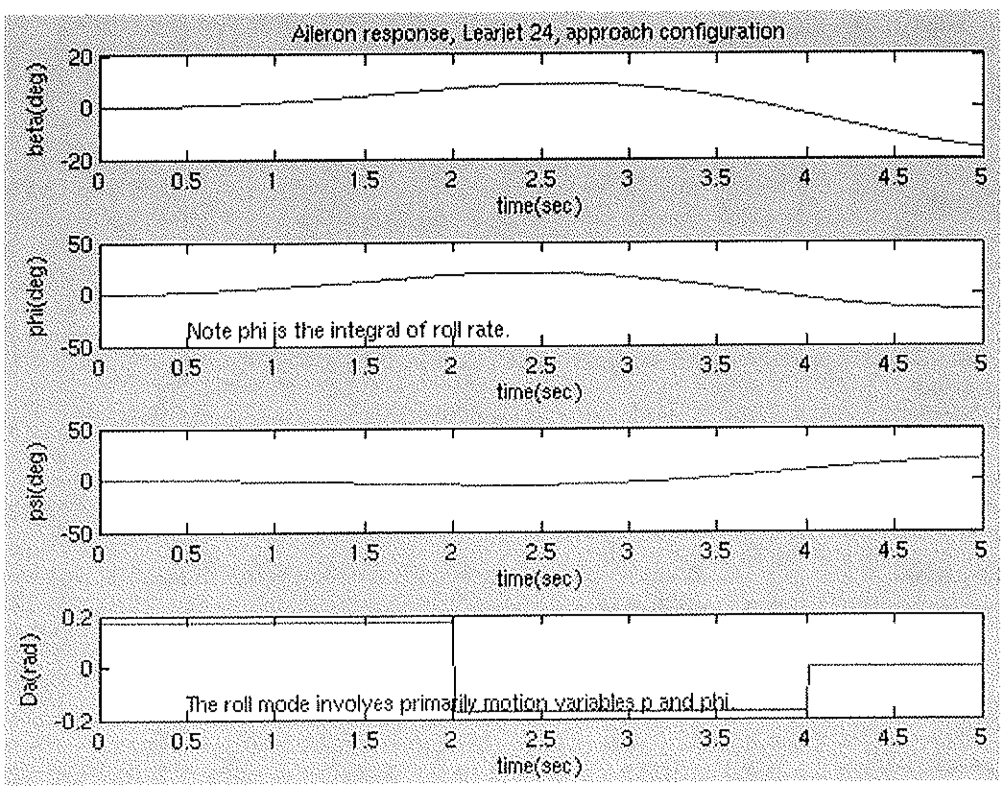
a =

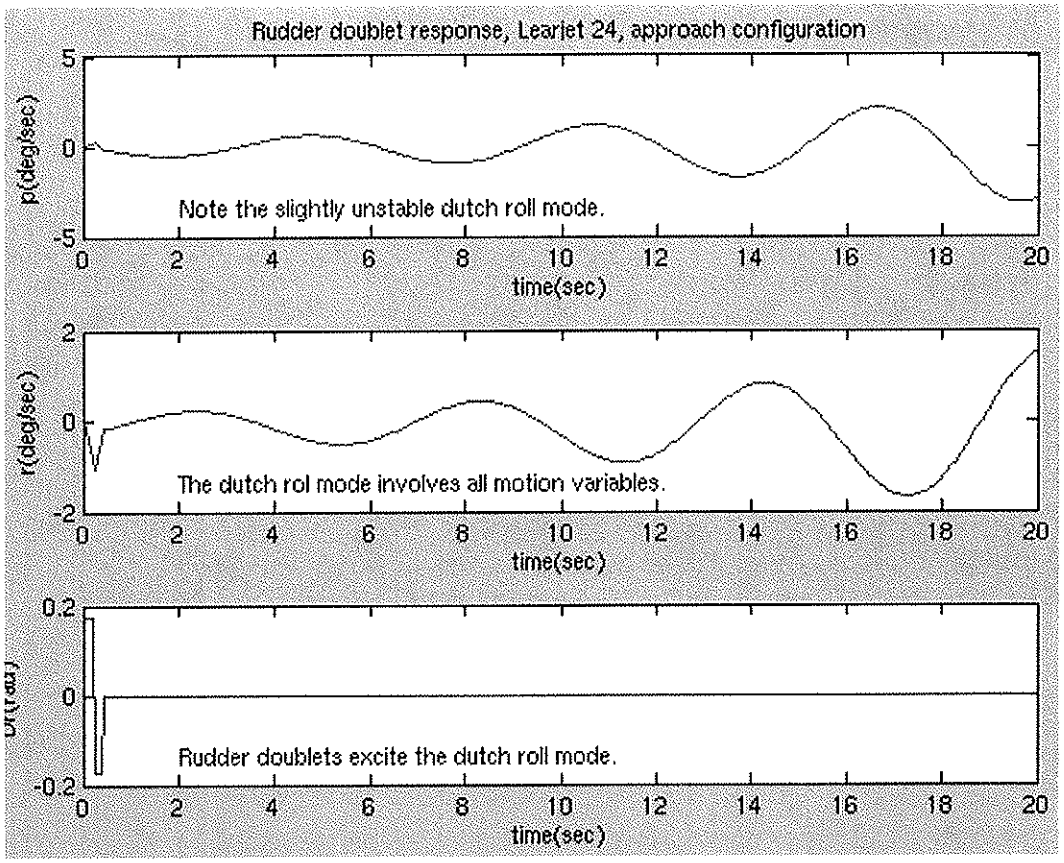
	beta(r)	p(r/s)	r(r/s)	phi(r)	psi(r)
beta(r)	-0.08383	0	-0.9954	0.1885	0
p(r/s)	-1.674	-0.3737	0.4324	0	0
r(r/s)	0.8677	-0.07113	-0.003393	0	0
phi(r)	0	1	0	0	0
psi(r)	0	0	1	0	0

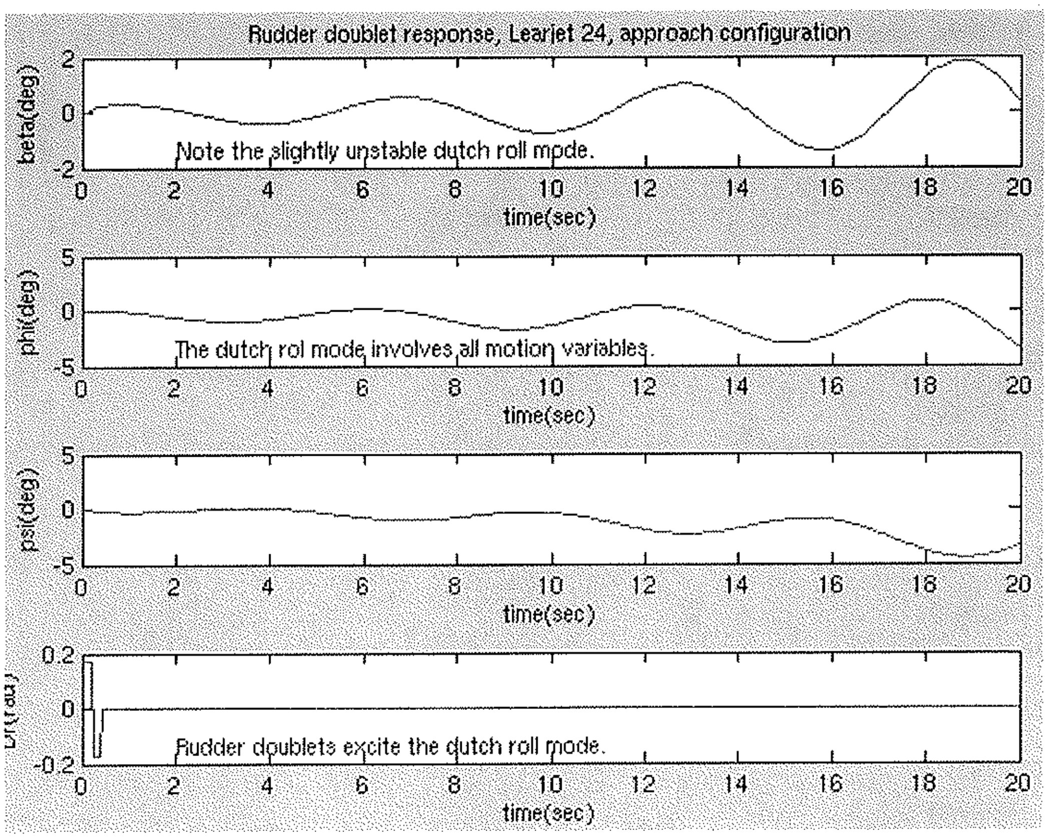
Continuous-time model.

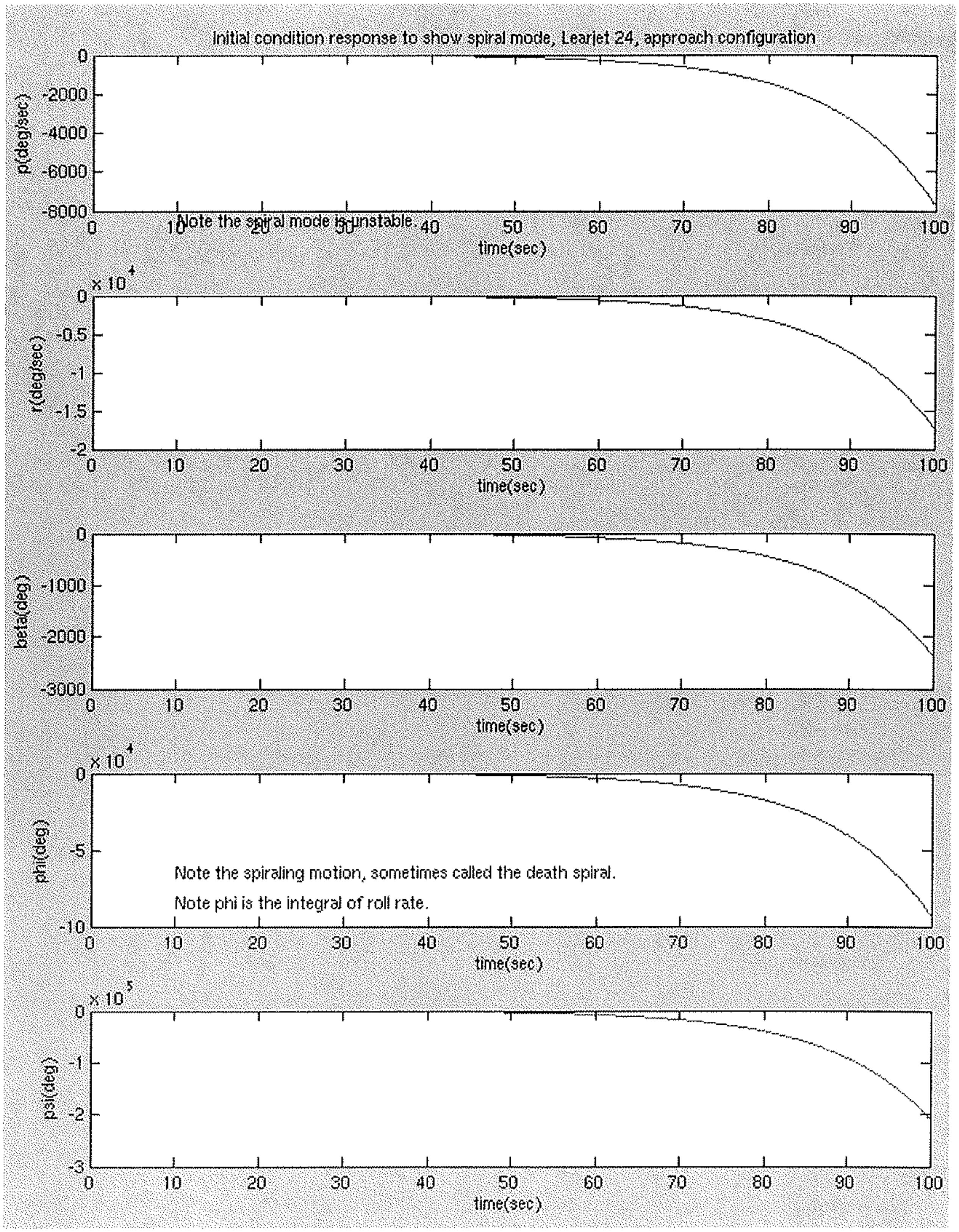
```
Wn =               <<<<<< Natural Frequency (rad/sec)

          0       <<<<< Heading Mode
   0.0829120      <<<<< Spiral Mode (a slow real mode, can be stable or
                                    unstable)
    0.74719       <<<<< Roll Mode (a fast real mode)
     1.0604       <<<<< Dutch Roll Mode (a complex mode)
     1.0604       <<<<< Dutch Roll Mode (a complex mode)


z =               <<<<< Damping Ratio

         -1       <<<<< Heading Mode
         -1       <<<<< Spiral Mode
          1       <<<<< Roll
  -0.095868       <<<<< Dutch Roll Mode (unstable in this case, usually
                                    lightly damped and stable)
  -0.095868       <<<<< Dutch Roll Mode


Poles =

          0                     <<<<< Heading Mode
   0.082912                     <<<<< Spiral Mode
   -0.74719                     <<<<< Roll Mode
    0.10165 +      1.0555i      <<<<< Dutch Roll Mode
    0.10165 -      1.0555i      <<<<< Dutch Roll Mode
```


Aileron response, Learjet 24, approach configuration
p(deg/sec)
Note Da controls roll rate through the roll mode.
time(sec)
r(deg/sec)
time(sec)
Da(rad)
The roll mode involves primarily motion variables p and phi.
time(sec)


Aileron response, Learjet 24, approach configuration
beta(deg)
time(sec)
phi(deg)
Note phi is the integral of roll rate.
time(sec)
psi(deg)
time(sec)
Da(rad)
The roll mode involves primarily motion variables p and phi.
time(sec)


Rudder doublet response, Learjet 24, approach configuration
p(deg/sec)
Note the slightly unstable dutch roll mode.
time(sec)
r(deg/sec)
The dutch rol mode involves all motion variables.
time(sec)
Rudder doublets excite the dutch roll mode.
time(sec)


Rudder doublet response, Learjet 24, approach configuration
beta(deg)
Note the slightly unstable dutch roll mode.
time(sec)
phi(deg)
The dutch rol mode involves all motion variables.
time(sec)
psi(deg)
time(sec)
Rudder doublets excite the dutch roll mode.
time(sec)


Initial condition response to show spiral mode, Learjet 24, approach configuration
p(deg/sec)
0
-2000
-4000
-6000
-8000
Note the spiral mode is unstable.
0
10
20
30
40
50
60
70
80
90
100
time(sec)
x 10^4
r(deg/sec)
0
-0.5
-1
-1.5
-2
time(sec)
beta(deg)
0
-1000
-2000
-3000
time(sec)
x 10^4
phi(deg)
0
-5
-10
Note the spiraling motion, sometimes called the death spiral.
Note phi is the integral of roll rate.
time(sec)
x 10^5
psi(deg)
0
-1
-2
-3
time(sec)