

AD-A078 672

AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OH
DIGITAL COMPUTER SOLUTION OF AIRCRAFT LONGITUDINAL AND LATERAL --ETC(U)
JUL 79 J M GRIFFIN , R B YEAGER , L B JORDAN

F/G 1/3

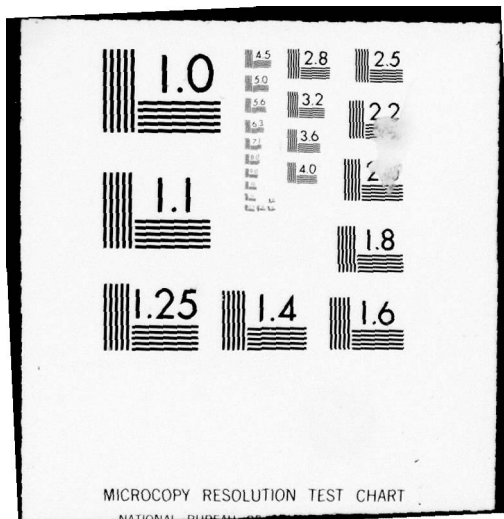
UNCLASSIFIED

AFFDL-TR-78-203

NL

1 OF 2
ADA
078672





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFFDL-TR-78-203

LEVEL III

22

ADA 078672

**DIGITAL COMPUTER SOLUTION OF AIRCRAFT
LONGITUDINAL AND LATERAL - DIRECTIONAL
DYNAMIC CHARACTERISTICS**

Control Dynamics Branch
Flight Control Division

DDC
RECEIVED
DEC 26 1979
E

July 1979

TECHNICAL REPORT AFFDL-TR-78-203

Final Report for Period

DDC FILE COPY

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

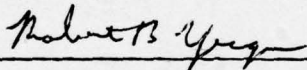
79 12 17 108

NOTICE

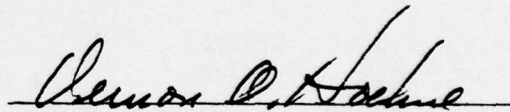
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

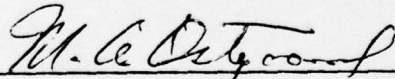


ROBERT B. YEAGER, Lt Colonel, USAFR
Project Engineer



VERNON O. HOEHNE, Acting Chief
Control Dynamics Branch

FOR THE COMMANDER



MORRIS A. OSTGAARD, Acting Chief
Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify _____, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
14 1. REPORT NUMBER AFFDL-TR-78-203	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9	
6 4. TITLE (and Subtitle) DIGITAL COMPUTER SOLUTION OF AIRCRAFT LONGITUDINAL AND LATERAL-DIRECTIONAL DYNAMIC CHARACTERISTICS.		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report.	
10 7. AUTHOR(s) John M. Griffin (see block 18) Robert B. Yeager, Larry B. Jordan, David A. Ratino		8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Flight Dynamics Laboratory (AFFDL/FGC) Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 2403-05-07	11. REPORT DATE July 1979	12. NUMBER OF PAGES 171
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Wright Aeronautical Laboratories (AFSC) Wright-Patterson Air Force Base, Ohio 45433	14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES Originally published as SEG-TR-66-52, December 1967, AD # Revised by Lt Col Robert B. Yeager, Maj Larry B. Jordan, and Maj David A. Ratino, USAFR.			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Stability and control Handling qualities Aircraft Digital computer programs			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Two Fortran IV computer programs are presented for the solution of aircraft longitudinal and lateral-directional transfer function factors and dynamic characteristics. The longitudinal program solves for the three-degree-of-freedom dynamic characteristics (phugoid damping ratio and natural frequency, short period damping ratio and natural frequency, etc.) and the numerator factors of the alpha, u, theta, h, and vertical acceleration transfer functions. The lateral-directional program solves for the three-degree-of-freedom			

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

012 070

PC

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

characteristics (Dutch roll damping ratio and natural frequency, roll and spiral mode time constants, etc.) and the numerator factors of the beta, phi, γ , and lateral acceleration transfer functions. In addition, some time histories and specialized handling qualities parameters can be computed and printed out. The equations and their underlying assumptions are discussed. The two complete computer programs are shown, and the input, output, and program functions are discussed.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

FOREWORD

This work was accomplished in-house by personnel of the Stability and Control Branch, Aeromechanics Division, Directorate of Airframe Subsystems Engineering, Systems Engineering Group, Research and Technology Division, which has become the Flight Stability and Control Branch, Flight Technology Division, Directorate of Flight Systems Engineering, Deputy for Engineering, Aeronautical Systems Division, ASD/ENFTC. It is applicable to aerospace systems. The initial part of the work was done between 1 January and 15 February 1965; since then, the computer programs have undergone several major revisions to reach their present status. Earlier versions were supplied to Lockheed-Georgia, Martin-Baltimore, NASA-Langley and AFFTC, Edwards Air Force Base. The digital work was done at the open shop facilities of the Systems Engineering Group.

The efforts of Mr. Paul Pietrzak in laying the basic foundation for this work are greatly appreciated, as well as the efforts of Miss Carol Scherer for her aid in digital programming and mathematics, and of Mr. Herbert Hickey for his aid in selecting handling qualities parameters.

This report, SEG-TR-66-52, was submitted by the original author, John H. Griffin, during October 1966 and was reviewed and approved by Richard H. Klepinger, Chief, Aeromechanics Division, Directorate of Airframe, Subsystems Engineering.

Report SEG-TR-66-52 was revised by members of the ASD Reserves for AFFDL/FGC to reflect numerous changes that have occurred in the computer program since the original report was written.

Accession For	
NTIS GMA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

TABLE OF CONTENTS

SECTION	PAGE
I INTRODUCTION	1
II DISCUSSION OF EQUATIONS OF MOTION	2
1. Longitudinal Motion	3
2. Lateral-Directional Motion	9
3. Assumptions for the Equations of Motion	9
III DISCUSSION OF THE COMPUTER PROGRAMS	11
1. Longitudinal Program	11
2. Lateral-Directional Program	27
IV CONCLUDING REMARKS	38
APPENDIX A TRANSFER EQUATIONS	39
APPENDIX B LONGITUDINAL EQUATIONS OF MOTION AND TRANSFER FUNCTIONS	41
APPENDIX C LATERAL-DIRECTIONAL EQUATIONS	53
APPENDIX D TIME TO n^{th} AMPLITUDE	77
APPENDIX E COMPUTER PROGRAM LISTING	81
REFERENCES	152

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1	Reference Inertial Axis and Associated Airframe Motion	3
2	Definition of ξ	16
3	Sample Longitudinal, Stability Axis, Per Radian Input Data. (No Plot)	19
4	Conventional Notation	30
5	Program Notation	30

LIST OF TABLES

TABLE		PAGE
1	Longitudinal Input Data	12
2	Longitudinal Input Formats	20
3	Lateral-Directional Input Data	31
3A	Lateral-Directional Options	32
4	Lateral-Directional Input Formats	33

LIST OF SYMBOLS

1. DEFINITION OF ALPHABETICAL SYMBOLS

- A = amplitude, degrees, degrees/sec, radians, radians/sec
- A = coefficient of an equation
- a_i = linear acceleration along the i th axis at the C.G., ft/sec²
- a_2 = a_i at a distance l_x from the C.G.
- a_y = a_y at a distance l_x from the C.G.
- a = speed of sound, ft/sec
- B = coefficient of an equation
- b = span, ft
- C = coefficient of an equation
- CG = center of gravity
- \bar{c} = mean aerodynamic chord, ft
- C_i = aerodynamic coefficient, per radian or per degree
($i = L, D, l, m, n, \dots$)
- $C_{i,j}$ = derivative of an aerodynamic coefficient C_i with respect to a function of a variable j
- D = drag, lbs
- D = coefficient of an equation
- E = coefficient of an equation
- e. = 2.71828
- F = force, lbs
- f = frequency, $\omega/2\pi$, cycles per second
- $f(i)$ = function of i
- g = acceleration of gravity, ft/sec²
- gs = $(g/U_0) \sin \Gamma_0$
- gc = $(g/U_0) \cos \Gamma_0$

LIST OF SYMBOLS (CONTINUED)

h	=	altitude, ft
h	=	moment of momentum, ft-lbs-sec
i	=	summation index
I	=	moment of inertia, slugs-ft ²
I_x	=	moment of inertia about the x-axis, slug ft ²
I_y	=	moment of inertia about the y-axis, slug ft ²
I_z	=	moment of inertia about the z-axis, slug ft ²
I_{xz}	=	product of inertia about the xz-axes, slug ft ²
I_{i_I}	=	moment of inertia about the i^{th} input axis
i_w	=	wing incidence angle, degrees
j	=	summation index
j	=	$\sqrt{-1}$
K	=	gain
K_d/K_{ss}	=	Dutch roll excitation parameter
k	=	constant
L_i	=	dimensional stability derivatives, roll axis
L'_i	=	primed dimensional stability derivative, roll axis
L	=	lift, lbs
L_α	=	change in lift due to change in angle of attack, lbs/deg
l_x	=	distance from CG to point at which acceleration transfer function will be measured, positive forward, ft
ℓ, l	=	rolling moment, ft-lbs
m	=	pitching moment, ft-lbs
M_i	=	dimensional stability derivative, pitch axis
M	=	Mach number
m	=	mass, slugs

LIST OF SYMBOLS (CONTINUED)

mil	= 1 mil = .0573 deg
N	= normal force, positive toward top of aircraft, lbs
n	= load factor
n	= any positive integer
n	= yawing moment, ft-lbs
$n_{z\alpha}$	= load factor response to change in angle of attack
N_i	= dimensional stability derivative, yaw axis
N'_i	= primed dimensional stability derivative, yaw axis
P	= period of an oscillation, sec.
p	= roll rate, radians/second or degrees/second
$\frac{pb}{2V}$	= roll helix angle, radians
P_1	= the first maximum value of roll rate in response to a control step input
P_2	= the first minimum in roll rate following the first maximum in roll rate in response to a control step input
q	= pitch rate, radians/second or degrees/second
\bar{q}	= dynamic pressure, lbs/ft ²
r	= yaw rate, radians or degrees per second
S	= reference area, ft ²
s	= Laplacian operator
T	= thrust, lbs
T_{DR}	= undamped Dutch roll mode period, sec
T_{dDR}	= damped Dutch roll mode period, sec
T_ϕ°	= time to bank to ϕ° of bank angle, sec

LIST OF SYMBOLS (CONTINUED)

t	=	time
U_o	=	initial longitudinal velocity along the axis of the stability axes, ft/sec
u	=	perturbation longitudinal velocity, ft/sec
V	=	total velocity, ft/sec
v	=	perturbation side velocity, ft/sec
W	=	gross weight, lbs
W	=	total vertical velocity along Z axis of the stability axes, ft/sec
w	=	perturbation vertical velocity, ft/sec
X	=	axial force, positive forward, lbs
X_i	=	dimensional stability derivative
x	=	reference axis or direction
y	=	side force, positive to pilot's right, lbs
Y_i	=	dimensional stability derivative
y	=	reference axis or direction
Z	=	-N
Z_i	=	dimensional stability derivative
z	=	reference axis or direction
z_t	=	perpendicular distance in the X-Z plane from the CG to the thrust line, positive down, ft
α	=	angle of attack, positive nose up, degrees
$\alpha_A, \alpha_I, \alpha_X$	=	reference axis angles, (A=aero, I=inertial, X=output)
α_w	=	wing angle of attack, degrees
β	=	angle of sideslip, positive nose left, degrees
$\Delta\beta_{MAX}$	=	maximum sideslip excursion occurring in 2 seconds or one-half the Dutch roll period, whichever is greater, for a step aileron input, degrees

LIST OF SYMBOLS (CONTINUED)

Γ	= flight path inclination angle, positive up, degrees
Δ	= denominator of a transfer function
γ	= perturbation flight path angle, degrees
δ	= control deflection, radians
δ_a	= roll control deflection, positive when producing right wing down rolling moment
δ_r	= directional control deflection, positive when producing positive side force and nose right rotation
ζ	= damping ratio
ζ_ϕ	= damping ratio of the ϕ/δ_a transfer function numerator quadratic
θ	= pitch attitude, positive up, degrees
ξ	= angle between body and thrust axes, positive for thrust component up, degrees
π	= 3.1416
ρ	= air density, slugs/ft ³
σ	= real part of complex root, 1/sec
τ	= time constant of the i^{th} mode of motion, time to 0.63 amplitude, seconds ($i = R, S, \text{etc.}$)
ϕ	= bank angle, positive right wing down, degrees
$ \phi / \beta $	= magnitude of the ratio of the free Dutch roll oscillation in bank angle to the free Dutch roll oscillation in sideslip
ψ	= heading angle, positive nose right, degrees
ψ_β	= phase angle of the Dutch roll oscillation in sideslip, degrees
ψ_p	= phase angle of the Dutch roll oscillation in roll rate, degrees
$\psi_{p/\beta}$	= phase angle between the free Dutch roll oscillations in roll rate and sideslip, degrees
ω	= frequency, $2\pi f$, radians per second
ω	= imaginary part of complex root, radians/sec

LIST OF SYMBOLS (CONTINUED)

- $\omega_{DR}, \omega_{nDR}, \omega_D$ = undamped natural frequency of the Dutch roll mode, radians/sec.
- ω_{dDR} = damped natural frequency of the Dutch roll mode, radians per second.
- $\omega_{SP}, \omega_{nSP}$ = undamped natural frequency of the short period mode, radians per second.
- ω_{ϕ} = undamped natural frequency of the $\phi/\delta a$ transfer function numerator quadratic, 1/sec.

Subscripts

- o = initial condition
- 1, 2 = sequence of sum variable
- 1/2 = one half
- 2 = double
- 1/10 = one tenth
- 10 = ten times
- A = aileron
- a = acceleration
- CL = closed loop
- D = Dutch roll mode (also DR)
- D = denominator
- e = elevator
- e = equivalent (as in V_e)
- h = altitude
- i = any independent variable
- j = any independent variable
- N = numerator
- n = natural
- n_d = natural damped

Subscripts (Concluded)

- p = phugoid mode
 p = roll rate
 q = pitch rate
 r = yaw rate ($1/\tau_r$ as in yaw rate transfer function)
 R = rudder (as in δ_R)
 R = roll mode (as in τ_R)
RPM = revolutions per minute (engine speed)
 S = spiral mode (as in τ_S)
SB = speed brake
 sp = short period mode
 T = thrust
 u = longitudinal velocity
 v = side velocity
 w = vertical velocity
 $x, y,$ and z = reference axes
 δ = control deflection
osc = oscillatory portion of component of an airplane response to a step control input
av = average response of an airplane to a step control input

Superscripts

- $(\dot{\quad})$ = time rate of change
 $(\quad)'$ = prime
 $(\hat{\quad})$ = caret - $(\quad)/u_0$

Other nomenclature is defined at the point of use.

SYMBOLS (CONTINUED)

2. DEFINITION OF AERODYNAMIC COEFFICIENTS

$$\begin{aligned}
 c_D &= \frac{D}{qS} & c_L &= \frac{L}{qS} \\
 c_{D_u} &= \frac{M}{2} c_{D_M} = \frac{U}{2} \frac{\partial c_D}{\partial u} & c_{L_u} &= \frac{M}{2} c_{L_M} = \frac{U}{2} \frac{\partial c_L}{\partial u} \\
 c_{D_M} &= \frac{\partial c_D}{\partial M} & c_{L_M} &= \frac{\partial c_L}{\partial M} \\
 c_{D_\alpha} &= \frac{\partial c_D}{\partial \alpha} & c_{L_\alpha} &= \frac{\partial c_L}{\partial \alpha} \\
 c_{D_{\dot{\alpha}}} &= \frac{\partial c_D}{\partial \left(\frac{\dot{\alpha} \bar{c}}{2U_0} \right)} & c_{L_{\dot{\alpha}}} &= \frac{\partial c_L}{\partial \left(\frac{\dot{\alpha} \bar{c}}{2U_0} \right)} \\
 c_{D_q} &= \frac{\partial c_D}{\partial \left(\frac{q \bar{c}}{2U_0} \right)} & c_{L_q} &= \frac{\partial c_L}{\partial \left(\frac{q \bar{c}}{2U_0} \right)} \\
 c_{D_{\delta_e}} &= \frac{\partial c_D}{\partial \delta_e} & c_{L_{\delta_e}} &= \frac{\partial c_L}{\partial \delta_e} \\
 c_{m_T} &= \frac{z_T \cdot T}{qS \bar{c}} & c_N &= \frac{N}{qS} \\
 c_{m_u} &= \frac{M}{2} c_{m_M} = \frac{U_0}{2} \frac{\partial c_m}{\partial u} & c_x &= \frac{-X}{qS}, \text{ positive aft} \\
 c_{m_M} &= \frac{\partial c_m}{\partial M} & c_y &= \frac{Y}{qS} \\
 c_{m_\alpha} &= \frac{\partial c_m}{\partial \alpha} & c_{y\beta} &= \frac{\partial c_y}{\partial \beta} \\
 c_{m_{\dot{\alpha}}} &= \frac{\partial c_m}{\partial \left(\frac{\dot{\alpha} \bar{c}}{2U_0} \right)} & c_{y\dot{\beta}} &= \frac{\partial c_y}{\partial \left(\frac{\dot{\beta} b}{2U_0} \right)} \\
 c_{m_q} &= \frac{\partial c_m}{\partial \left(\frac{q \bar{c}}{2U_0} \right)} & c_{y_r} &= \frac{\partial c_y}{\partial \left(\frac{r b}{2U_0} \right)} \\
 c_{m_{\delta_e}} &= \frac{\partial c_m}{\partial \delta_e} & c_{y_p} &= \frac{\partial c_y}{\partial \left(\frac{p b}{2U_0} \right)} \\
 & & c_{y_\delta} &= \frac{\partial c_y}{\partial \delta}
 \end{aligned}$$

SYMBOLS (CONTINUED)

$$\begin{aligned}
 C_n &= \frac{n}{qSb} & C_{\ell} &= \frac{\ell}{qSb} \\
 C_{n\beta} &= \frac{\partial C_n}{\partial \beta} & C_{\ell\beta} &= \frac{\partial C_{\ell}}{\partial \beta} \\
 C_{n\dot{\beta}} &= \frac{\partial C_n}{\partial \left(\frac{\beta b}{2U_0}\right)} & C_{\ell\dot{\beta}} &= \frac{\partial C_{\ell}}{\partial \left(\frac{\beta b}{2U_0}\right)} \\
 C_{nr} &= \frac{\partial C_n}{\partial \left(\frac{rb}{2U_0}\right)} & C_{\ell r} &= \frac{\partial C_{\ell}}{\partial \left(\frac{rb}{2U_0}\right)} \\
 C_{np} &= \frac{\partial C_n}{\partial \left(\frac{pb}{2U_0}\right)} & C_{\ell p} &= \frac{\partial C_{\ell}}{\partial \left(\frac{pb}{2U_0}\right)} \\
 C_{n\delta} &= \frac{\partial C_n}{\partial \delta} & C_{\ell\delta} &= \frac{\partial C_{\ell}}{\partial \delta}
 \end{aligned}$$

$$\begin{aligned}
 x_u &= \frac{-\rho S U_0}{m} (C_D + \frac{M}{2} C_{D_M}) & z_{\delta_e} &= \frac{-\rho S U_0^2}{2m} (C_{L\delta_e}) \\
 x_w &= \frac{\rho S U_0}{2m} (C_L - C_{D_a}) & M_u &= \frac{\rho S U_0 \bar{c}}{I_{yy}} (C_m + \frac{M}{2} C_{m_M}) \\
 x_{\dot{w}} &= \frac{-\rho S \bar{c}}{4m} (C_{D_a}) & M_w &= \frac{\rho S U_0 \bar{c}}{2I_{yy}} (C_{m_a}) \\
 x_q &= \frac{-\rho S U_0 \bar{c}}{4m} (C_{D_q}) & M_{\dot{w}} &= \frac{\rho S \bar{c}^2}{4I_{yy}} C_{m_{\dot{a}}} \\
 x_{\delta_e} &= \frac{-\rho S U_0^2}{2m} (C_{D\delta_e}) & M_q &= \frac{\rho S U_0 \bar{c}^2}{4I_{yy}} C_{m_q} \\
 z_u &= \frac{-\rho S U_0}{m} (C_L + \frac{M}{2} C_{L_M}) & M_{\delta_e} &= \frac{\rho S U_0 \bar{c}}{2I_{yy}} C_{m_{\delta_e}} \\
 z_w &= \frac{-\rho S U_0}{2m} (C_{L_a} + C_D) \\
 z_{\dot{w}} &= \frac{-\rho S \bar{c}}{4m} (C_{L_a}) \\
 z_q &= \frac{-\rho S U_0 \bar{c}}{4m} (C_{L_q})
 \end{aligned}$$

(CONTINUED) SYMBOLS

$$\begin{aligned}
 Y_v &= \frac{\rho S U_0}{2m} C_{y\beta} \\
 Y_{\dot{\beta}} &= \frac{\rho S b}{4m} C_{y\dot{\beta}} \\
 Y_r &= \frac{\rho S U_0 b}{4m} C_{y_r} \\
 Y_p &= \frac{\rho S U_0 b}{4m} C_{y_p} \\
 Y_{\delta} &= \frac{\rho U_0^2 S}{2m} C_{y\delta} \\
 N_{\beta} &= \frac{\rho S U_0^2 b}{2I_{zz}} C_{n\beta} \\
 N_{\dot{\beta}} &= \frac{\rho S U_0 b^2}{4I_{zz}} C_{n\dot{\beta}} \\
 N_r &= \frac{\rho S U_0 b^2}{4I_{zz}} C_{n_r} \\
 N_p &= \frac{\rho S U_0 b^2}{4I_{zz}} C_{n_p} \\
 N_{\delta} &= \frac{\rho S U_0^2 b}{2I_{zz}} C_{n\delta} \\
 L_{\beta} &= \frac{\rho S U_0^2 b}{2I_{xx}} C_{l\beta} \\
 L_{\dot{\beta}} &= \frac{\rho S U_0 b^2}{4I_{xx}} C_{l\dot{\beta}} \\
 L_r &= \frac{\rho S U_0 b^2}{4I_{xx}} C_{l_r} \\
 L_p &= \frac{\rho S U_0 b^2}{4I_{xx}} C_{l_p} \\
 L_{\delta} &= \frac{\rho S U_0^2 b}{2I_{xx}} C_{l\delta} \\
 L_a &= \frac{\rho S U_0}{2m} C_{L_a}
 \end{aligned}$$

$$\begin{aligned}
 N_i &= \frac{N_i + \frac{I_{xz}}{I_{zz}} L_i}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \\
 L_i &= \frac{L_i + \frac{I_{xz}}{I_{xx}} N_i}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \\
 \hat{Y}_r &= \frac{y_r}{U_0} \\
 \hat{Y}_p &= \frac{y_p}{U_0} \\
 \hat{Y}_{\delta} &= \frac{y_{\delta}}{U_0}
 \end{aligned}$$

SYMBOLS (CONTINUED)

3. CONVERSION OF COMPUTER SYMBOLS TO ENGINEERING SYMBOLS

A	= A_1 coefficient of an equation	CAYP	= $C_{a'y}$
AAYP	= $A_{a'y}$	CB	= C_β
AB	= A_β	CL	= C_L
AH	= A_h	CLA	= $C_{L\alpha}$
ALFAA	= α_A	CLAD	= $C_{l\dot{\alpha}}$
ALFAI	= α_I	CLB	= $C_{l\beta}$
ALFAX	= α_X	CLBD	= $C_{l\dot{\beta}}$
ALPHA	= α , angle of attack	CLDA	= $C_{l\delta\alpha}$
ANGLE P/B	= δ P/B	CLDE	= $C_{L\delta e}$
AP	= A_ϕ	CLDR	= $C_{l\delta r}$
AR	= A_r	CLM	= C_{LM}
AT	= A_θ	CLP	= C_{lp}
AU	= A_u	CLQ	= C_{Lq}
AW	= A_w	CLR	= C_{lr}
AZ	= a_z	CMT	= $C_{m\text{thrust}}$
B	= B_i , coefficient	CNB	= $C_{n\beta}$
B	= b, span	CNBD	= $C_{n\dot{\beta}}$
BAYP	= $B_{a'y}$		
BB	= B_β		
BP	= B_p		
BR	= B_r		
B(T)	= $\beta(t)$		
C	= C, coefficient		

SYMBOLS (CONTINUED)

CNDA = $C_{n\delta_a}$	G = g, acceleration of gravity
CNDR = $C_{n\delta_r}$	GAMA = Γ
CNP = C_{n_p}	GWT = W, gross weight
CNR = C_{n_r}	IX = I_x
CX = C_x	IXB = (I_x) body axis
CYB = C_{y_p}	IXI = I_{XI}
CYBD = $C_{y\dot{\beta}}$	IXS = (I_x) stability axis
CYDA = $C_{y\delta_a}$	IXZ = I_{XZ}
CYDR = $C_{y\delta_r}$	IXZI = I_{XZI}
CYP = C_{y_p}	IZ = I_z
CYR = C_{y_r}	IZI = I_{ZI}
D = D, coefficient	KB = K_β
DAYP = $D_{a'y}$	KBR = K_{β_R}
DB = D_β	KBS = K_{β_S}
DBMAX = $\Delta\beta_{MAX}/UNIT STEP$	KD/KSS = K_d/K_{ss}
DP = D_p	KP = K_p
DR = D_r	KPR = K_{p_R}
E = E, coefficient	KPS = K_{p_S}
EAYP = $E_{a'y}$	LA = L_α
FAYP = $F_{a'y}$	LB = L_β
	LBD = $L_{\dot{\beta}}$
	LBDP = $L'_{\dot{\beta}}$
	LBP = L'_{β}

SYMBOLS (CONTINUED)

LDA = L_{δ_a}	NBP = N'_{β}
LDAP = L'_{δ_a}	NDA = N_{δ_a}
LDR = L_{δ_r}	NDAP = N'_{δ_a}
LDRP = L'_{δ_r}	NDR = N_{δ_r}
LP = L_p	NDRP = N_{δ_r}
LPP = L'_p	NP = N_p
LR = L_r	NPP = N'_p
LRP = L'_r	NR = N_r
LX = l_x	NRP = N'_r
MAC = \bar{c}	P2/P1 = p_2/p_1
MACH = Mach number	PHIA = $\phi(t_A)$
MD = M_{δ}	PHI OSC/PHI AV = ϕ_{osc}/ϕ_{AV}
MKBPDR = $ K'_{\beta DR} $	POSC/PAV = P_{osc}/P_{ave}
MKPPDR = $ K'_{p DR} $	PSIB = ψ_{β}
MU = M_u	PSIBP = ψ'_{β}
MWD = M_w	PSIP = ψ_p
NB = N_{β}	P(T) = $p(t)$
NBD = N_{β}	RHO = ρ
NBDP = N'_{β}	S = S_w (reference area)
	SPAN = b

SYMBOLS (CONCLUDED)

TDDR = τ_{dDR}	YDA = Y_{δ_a}
TDR = τ_{DR}	YDR = Y_{δ_r}
TDT = $T_{\delta_{RPM}}$	YP = Y_p
TR = τ_R	YR = Y_r
TS = τ_S	ZD = Z_{δ}
1/TR = $1/\tau_r$	ZDR = ζ_{DR}
1/TAYI = $(1/\tau_{ay})_1$	ZP = ζ_p
U = U_0	ZSP = ζ_{sp}
V = V	ZT = z_t
VE = $V_{\text{equivalent}}$	ZW = Z_w
WDDR = ω_{dDR}	
WDR = ω_{DR}	
WP = ω_p	
WPHI/WDR = $\omega_{\phi}/\omega_{DR}$	
WSP = ω_{nSP}	
XQ = X_q	
XU = X_u	
YB = Y_{β}	
YBD = $Y_{\dot{\beta}}$	

SECTION I

INTRODUCTION

During the initial design phases of an aircraft or missile system, the aerodynamic characteristics of the airframe can be estimated to determine whether or not the approach being taken to meet the design objectives is correct. As the design progresses, the data must be more refined with more accurate airframe characteristics. The preliminary estimation methods are no longer acceptable. The methods for calculating the airframe characteristics used in defining the handling-qualities parameters for the final design are long and complex. In fact, they are so much so that a computer analysis is a necessity for today's systems. Therefore, these computer programs have been prepared for the solution of the longitudinal and lateral-directional equations of motion, each a separate entity and each consisting of three degrees of freedom. These computer programs are presented in this report. The longitudinal and lateral-directional modes are assumed to be uncoupled and the equations are linearized.

Handling-qualities information was a prime requirement for this study. When the equations were solved and programmed, therefore, considerable effort was devoted toward decreasing the amount of time spent in calculating such parameters as ω_n/L_α , n_{z_α} , ϕ/v_e , and ω_ϕ/ω_D . Many handling-qualities parameters are presented, but many others had to be excluded because a tremendous amount of input data would be required to define all the parameters. The two computer programs presented herein are complete Fortran IV programs.

SECTION II

DISCUSSION OF EQUATIONS OF MOTION

The derivation of the equations of motion is based on the classical method -- Newton's laws of motion referenced to an axis fixed in space. Newton's laws state that the force acting on a body is equal to the time rate of change of momentum, and the torque applied to the body is equal to the time rate of change of the moment of momentum. This can be stated mathematically for the reference system shown in Figure 1 as follows:

(1)

$$\Sigma F_x = \frac{d}{dt} (mU) \quad (2)$$

$$\Sigma F_y = \frac{d}{dt} (mV) \quad (3)$$

$$\Sigma F_z = \frac{d}{dt} (mW) \quad (4)$$

$$\Sigma L = \frac{dh_x}{dt} \quad (5)$$

$$\Sigma M = \frac{dh_y}{dt} \quad (6)$$

$$\Sigma N = \frac{dh_z}{dt} \quad (6)$$

This report will proceed no further with the fundamental derivation of the equations of motion; numerous reports have treated this subject, such as Reference 1. Further discussion in the use of these equations is broken into two sections, longitudinal and lateral-directional.

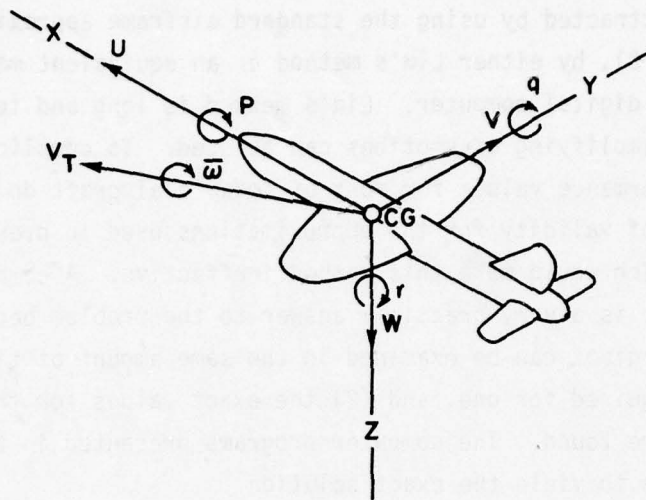


Figure 1. Reference Inertial Axis and Associated Airframe Motion

1. LONGITUDINAL MOTION

The linearized longitudinal equations of motion are ΣF_x , ΣF_y , and ΣM (Appendix A, Equations A-1, A-2, and A-3). The equations apply to an operating point in steady unaccelerated flight. To define the basic airframe characteristics in terms of mode damping and frequency, etc., the characteristic equation is derived (see Appendix A) with the final form

$$As^4 + Bs^3 + Cs^2 + Ds + E = 0 \quad (7)$$

The solution to this equation yields four roots. For the most common case, the solution is in the form

$$(s^2 + 2\zeta\omega_n s + \omega_n^2)_p (s^2 + 2\zeta\omega_n s + \omega_n^2)_{sp} \quad (8)$$

where the subscripts p and sp represent phugoid and short period modes, respectively. The characteristics (ζ and ω) specify the controls-fixed motion when the airframe is subjected to a unit impulse at $t = 0$.

Once the coefficients of Equation 7 are calculated, the roots of the equation can be extracted by using the standard airframe approximation (References 1 and 2), by either Lin's method or an equivalent method of factoring or by a digital computer. Lin's method is long and tedious, even if certain simplifying assumptions can be used. To complicate matters, the performance values for most of today's aircraft do not lie within the range of validity for the approximations used in previous studies (Reference 2), which would make this method ineffective. A computer solution, however, is a very practical answer to the problem because: (1) many flight regimes can be examined in the same amount of time that previously was required for one, and (2) the exact values for the roots and characteristics are found. The computer programs presented in this report are written to yield the exact solution.

The solution to the characteristic equation yields much information about the airframe, but more information is provided if specific control inputs are used by solving for the transfer functions of the airframe. Three basic transfer functions are derived in Appendix B. These are $\alpha(s)/\delta_e(s)$, $\theta(s)/\delta_e(s)$, and $u(s)/\delta_e(s)$. These transfer functions not only provide valuable information for design and optimization of the automatic flight control system but are a source of handling-qualities information. As an example, several reports (References 3 and 4) discuss the importance of the time constant in the numerator of the pitch attitude to elevator deflection transfer function.

From the three basic transfer functions α , θ , and u , many others can be derived. For example, rate of climb, altitude, and vertical acceleration responses can easily be derived by combining these three basic transfer functions. The altitude per delta elevator transfer function is included in the program for the longitudinal transfer functions; this program can be used as an example for deriving the others.

Assuming that $u(o^+) = \theta(o^+) = w(o^+) = 0$,

the equation for rate of climb is, for $\sin \Gamma \approx \Gamma$:

$$\dot{h} = U\Gamma \quad (9)$$

But $U = U_0 + u$ and $\Gamma = \Gamma_0 + \gamma$ so*

$$\dot{h} = (U_0 + u)(\Gamma_0 + \gamma) = U_0\Gamma_0 + \Gamma_0 u + U_0\gamma \quad (10)^*$$

However, with $\phi = 0$, $\gamma = \theta - \alpha$, so

$$\dot{h} = U_0\Gamma_0 + \Gamma_0 u - U_0\alpha + U_0\theta \quad (11)$$

Letting $\alpha = w/U_0$, Equation 11 becomes

$$\dot{h} = U_0\Gamma_0 + \Gamma_0 u - w + U_0\theta \quad (12)$$

Taking the Laplace transform yields

$$sh(s) = \frac{U_0\Gamma_0}{s} + \Gamma_0 u(s) - w(s) + U_0\theta(s) \quad (13)$$

The conditions for the altitude transfer function presented in the computer program are $\Gamma_0 = 0$, and initial steady flight at the operating point. Thus, Equation 13 can be expressed as

$$sh(s) = \left(U_0 \frac{\theta(s)}{\delta(s)} - \frac{w(s)}{\delta(s)} \right) \delta(s) \quad (14)$$

Now, expressing $\frac{\theta(s)}{\delta(s)}$ and $\frac{w(s)}{\delta(s)}$ in the general form

$$\frac{A_j s^m + B_j s^{m-1} + \dots}{A s^n + B s^{n-1} + \dots} \quad (15)$$

one can write (note the free s in the denominator)

$$\frac{h(s)}{\delta(s)} = \frac{A_h s^3 + B_h s^2 + C_h s + D_h}{s(A s^4 + B s^3 + C s^2 + D s + E)} \quad (16)$$

*The term $u\gamma$ is neglected because it is the product of small perturbations.

where the numerator coefficients are combinations of the $\theta(s)/\delta(s)$ and $w(s)/\delta(s)$ transfer functions and the denominator coefficients are from the longitudinal characteristic equation.

The numerator coefficients are

$$A_h = -Z_\delta \quad (17)$$

$$B_h = X_\delta Z_u + Z_\delta (X_u + M_q + U_0 M_{\dot{w}}) - M_\delta (U_0 Z_{\dot{w}} + Z_q) \quad (18)$$

$$\begin{aligned} C_h = & X_\delta (M_q Z_u - M_u Z_q + U_0 Z_u M_{\dot{w}} - M_u U_0 Z_{\dot{w}}) \\ & + Z_\delta (M_u X_q - X_u M_q + M_u U_0 X_{\dot{w}} + U_0 M_{\dot{w}} - X_u U_0 M_{\dot{w}}) \\ & + M_\delta (X_u Z_q - Z_u X_q + X_u U_0 Z_{\dot{w}} - Z_u U_0 X_{\dot{w}} - U_0 Z_{\dot{w}}) \end{aligned} \quad (19)$$

$$\begin{aligned} D_h = & X_\delta (Z_u U_0 M_w - M_u U_0 Z_w) \\ & + Z_\delta (-g M_u - X_u U_0 M_w + M_u U_0 X_w) \\ & + M_\delta (g Z_u + X_u U_0 Z_w - Z_u U_0 X_w) \end{aligned} \quad (20)$$

and are valid only when $\Gamma_0 = 0$.

The coefficients of the denominator, or characteristic equation, are as follows:

$$A = 1 - Z_{\dot{w}} \quad (21)$$

$$B = -A(X_u + M_q) - Z_w - M_{\dot{w}}(U_0 + Z_q) - Z_u X_{\dot{w}} \quad (22)$$

$$\begin{aligned} C = & X_u [M_q A + Z_w + M_{\dot{w}}(U_0 + Z_q)] - M_u [X_{\dot{w}}(U_0 + Z_q) + X_q A] \\ & + M_q Z_w + Z_u (X_{\dot{w}} M_q - X_w - M_{\dot{w}} X_q) + M_{\dot{w}} g \sin \Gamma_0 - M_w (U_0 + Z_q) \end{aligned} \quad (23)$$

$$\begin{aligned}
 D &= g \sin \Gamma_0 (X_w M_u + M_w - X_u M_w) + g \cos \Gamma_0 (Z_u M_w + M_u A) \\
 &+ M_u [X_q Z_w - X_w (U_0 + Z_q)] + Z_u (M_q X_w - M_w X_q) \\
 &+ X_u [M_w (U_0 + Z_q) - M_q Z_w]
 \end{aligned} \tag{24}$$

$$E = g \cos \Gamma_0 (Z_u M_w - M_u Z_w) + g \sin \Gamma_0 (M_u X_w - M_w X_u) \tag{25}$$

The rate of climb and the acceleration transfer functions can be found from the attitude transfer function, by successive differentiation

$$\dot{h}(t) = \frac{d}{dt} (h); \quad \frac{\dot{h}(s)}{\delta(s)} = s \frac{h(s)}{\delta(s)} = \frac{N_h}{\Delta} \tag{26}$$

The result is to remove a root of zero. For acceleration, there are two additional poles at zero in the transfer function for acceleration at the center of gravity (CG), but for the case where acceleration is desired at a specific point on the aircraft, the a_z transfer function becomes different from an s multiple of N_h . For acceleration at some point different from the CG where $a_{z_{CG}} = -\ddot{h}$

$$a_z = a_{z_{CG}} - \ell_x \dot{q} = a_{z_{CG}} - \ell_x \ddot{\theta} \tag{27}$$

(a_z is positive downward).

so

$$\frac{a_z(s)}{\delta(s)} = \frac{-s^2 h(s)}{\delta(s)} - \frac{s^2 \ell_x \theta(s)}{\delta(s)} \tag{28}$$

or

$$\frac{a_z(s)}{\delta(s)} = s \left(\frac{w(s)}{\delta(s)} - U_0 \frac{\theta(s)}{\delta(s)} - \ell_x \frac{\theta(s)}{\delta(s)} \right) \tag{29}$$

This transfer function is programmed but is printed out only when ℓ_x is different from zero.

Some of the more subtle characteristics of the equations are:

1) They cannot be used to obtain the basic airframe characteristics while the aircraft is in a steady pull-up and the load factor is greater than 1.0 because the terms involving Q_0 have been deleted. While it would be desirable to determine the airframe's characteristics under load, even if the equations could accept the necessary inputs, the aerodynamic coefficients would have to be corrected for aeroelasticity under load.

2) Initial conditions of any angle greater than 15 degrees inject errors of greater than 1%. For the sine error at 15°

$$\% \text{ error} = \frac{15/57.3 - \sin 15^\circ}{15/57.3} = 1.13\% \quad (30)$$

For the cosine error at 15°

$$\% \text{ error} = \frac{1 - \cos 15^\circ}{\cos 15^\circ} = 3.53\% \quad (31)$$

The tangent error is -2.36%. Thus the small angle assumption injects as much as 3.5% error at 15° of α_0 or Γ_0 , which should be the maximum error in any of the airframe characteristics. This is not considered an unacceptable level of error since aircraft flight angles are generally less than 15° and the basic aerodynamic data is seldom accurate within 3%.

3) The equation cannot be used for time and motion studies involving large angles because both small angles and small perturbations were assumed and these may not be small during a dynamic simulation.

Programming for the longitudinal equations is discussed further in Section III.

2. LATERAL-DIRECTIONAL MOTION

The lateral-directional equations of motion are derived from ΣF_y , ΣL , and ΣN and are presented in Appendix C as Equation C-1, C-2, and C-3. The characteristic equation, which is a quintic, (with a root at $s = 0$) and the transfer functions are derived in the same manner as the longitudinal equations of motion.

The three basic transfer functions are $\beta(s)/\delta(s)$, $\phi(s)/\delta(s)$, and $r(s)/\delta(s)$, where $r(s)/\delta(s)$ is $s\psi(s)/\delta(s)$. A fourth transfer function $a'_y(s)/\delta(s)$ is also included and is similar to $a_z(s)/\delta_e(s)$ in that it is derived from the three basic transfer functions (see Appendix II). The transfer functions are presented for both control deflections, i.e., aileron and rudder.

One primary handling-qualities parameter, the ω_ϕ/ω_D ratio, is calculated from the Dutch roll frequency and the frequency of the numerator of the roll angle transfer function. No approximations are used (see Appendix C).

Two of the three equations are selected and solved simultaneously for the ϕ to β ratio. Since this is a complex vector (or phasor), the magnitude $|\phi|/|\beta|$ is the square root of the sum of the squares of the real and imaginary parts of the numerator and denominator. This is shown in detail in Appendix C).

Time to 1/2 amplitude and time to double amplitude for the roll and spiral modes are not calculated. These calculations could be inserted at the expense of time and effort, but they are straightforward and are easily calculated. For the value of $T_{1/2}$ or T_2 , for an aperiodic mode, simply multiply the time constant by 0.693. The derivations are given in Appendix D.

3. ASSUMPTIONS FOR THE EQUATIONS OF MOTION

1) The airframe is assumed to be a rigid body at constant mass and inertias.

2) The earth is planar and fixed in space, and the earth's atmosphere is fixed with respect to the earth.

- 3) Rate of change of mass with respect to time is zero.
- 4) The XZ plane is a plane of symmetry.
- 5) The disturbances from the steady flight condition are sufficiently small to neglect products and squares of the changes in velocities when compared to the total values. Also, changes in air density during a disturbance are zero.
- 6) The airframe is initially wings level, and the only nonzero initial velocity is U_0 . ($V_0 = W_0 = 0$ defines stability axes; but in some lateral-directional options, output is provided in any desired symmetrical body areas.
- 7) Vehicle motions are slow enough that unsteady aerodynamic effects can be ignored.
- 8) Longitudinal motion does not induce lateral-directional motion.
- 9) The change in thrust with respect to velocity is linear.
- 10) No atmospheric disturbances occur. In the presence of a steady wind, motion is calculated with respect to the air mass.

SECTION III

DISCUSSION OF THE COMPUTER PROGRAMS

Two separate computer programs are shown, one for the three-degree-of-freedom longitudinal characteristic equation and five transfer functions, and one for the three-degree-of-freedom lateral-directional characteristic equation and four transfer functions. Both programs are written in Fortran IV language for the CDC 6600/CYBER 76 computers. The programs contain the Fortran subroutine DMULR (double-precision MULER) which is used to calculate the roots of the equations. In addition, the longitudinal program contains a Fortran subroutine called FRQCK (Frequency Check). The forms for the inputs and outputs of the two programs are similar and the same basic programming method was used.

1. LONGITUDINAL PROGRAM

a. General

The longitudinal program accepts data in several forms, and outputs in the form of airframe characteristics and transfer functions. The roots of the equations, associated mode time constants, damping, and frequency, and the coefficients of the equations are also printed on output. An example of the output is shown on pages 103 through 110.

The following step-by-step explanation of what the program does will help to explain the program's operation, input, and output.

(1) To run the program, prepare a set of aerodynamic data of the type shown in Table 1. Column 4 of Table 1 lists the data identification numbers associated with each data type; the identification number for the specific data type must appear in Columns 1, 2, and 3 of the first data card for each run. For further explanation of the input data card, see Figure 3.

TABLE 1
LONGITUDINAL INPUT DATA

Data Type	Units	Axis	Data Identification Number	Option
Dimensional	ft, sec, radian	stability	0 0 0	
Nondimensional	all per radian	stability	1 0 0	
	all per degree	stability	1 0 1	
	α, δ per degree $\dot{\alpha}, q$ per radian	stability	1 0 2	Derivative mix
	all per radian	stability	1 0 5	Namelist
	all per degree	stability	1 0 6	Namelist
	α, δ per degree $\dot{\alpha}, q$ per radian	stability	1 0 7	Namelist
	all per radian	body	1 1 0	
	all per degree	body	1 1 1	
	α, δ per degree $\dot{\alpha}, q$ per radian	body	1 1 2	Derivative mix
	all per radian	body	1 1 5	Namelist
	all per degree	body	1 1 6	Namelist
	α, δ per degree $\dot{\alpha}, q$ per radian	body	1 1 7	Namelist

Coupling numerators are obtained by adding 5 to the first digit of the data identification number; for example, 500 is the new data identification number for dimensional stability data in radians.

(2) The data are read and converted, if necessary, to dimensional stability axis data. The input data and the dimensional data are printed on output to allow a rapid check for errors in the input data. Printing the input data and converting it to the proper form takes the first 170 cards (see the program listing).

(3) The next operation is to calculate the coefficients of the denominator (characteristic equation) and then to call the subroutine DMULR to calculate the roots of an n^{th} order equation.

A feature of this subroutine is that the actual location of the root in the complex plane is found for both the first and second order factor. For example, a first order factor has the form

$$(s + \frac{1}{\tau}) = 0 \quad (32)$$

and the solution or root is

$$s = -\frac{1}{\tau} \quad (33)$$

It is in the latter form that DMULR calculates the solutions. Complex pairs are in the form

$$s = -\zeta\omega_n \pm \omega_n \sqrt{1-\zeta^2} j \quad (34)$$

when the values for the roots are printed on the output sheet.

The first order factor root will be printed as seen in Equation 33. Thus, negative roots are stable because they lie in the left half of the complex s-plane.

(4) After the roots are printed out, the program must choose the proper flow sequence in which to print the characteristics (ζ , ω , $1/\tau$, $C_{1/2}$, etc.) in the proper place with the proper labeling. Once the proper flow has been chosen, the program calculates the basic airframe characteristics for the output. Choosing the proper flow sequence and calculating the values on the first page of the output uses cards 243 through 344 plus the subroutine frequency check, FRQCK. FRQCK is used for the case in which one of the normally second-order modes of the denominator (short period or phugoid) combines into two real time constants instead of the classical complex conjugates. Frequency check then compares the frequency of the one remaining second order mode with that of the normal velocity transfer function numerator. The theory herein is based on the knowledge that the short-period-mode variables are primarily α and θ , while the phugoid mode variables are u plus θ or Γ . During a longitudinal oscillation, normal velocity will vary because of the phugoid contribution of $U\Gamma$ and the short period contribution of $U\theta$, plus $C_{L\alpha}$ effects. The contribution of $U\Gamma$ is usually more significant than any short period effects, so the frequency of the normal velocity numerator should be somewhere in the neighborhood of the phugoid frequency. Thus, the complex conjugate frequency of the characteristic equation is compared with that of the $w(s)/\delta_e(s)$ transfer function numerator, and if it lies within 40% of the $w(s)/\delta_e(s)$ frequency, it is assumed to be the phugoid mode. Once this information is known, the proper write sequence can be chosen.

(5) After the denominator characteristics are calculated and printed on output, the transfer functions are calculated in much the same way. Once the program has finished with one set of data, it goes back to the beginning of the program, reads the next set of data, and starts all over again for this next run. The second and any successive runs need not be the same type of data as any other run because each set of data is identified as shown in Table 1.

b. Input Parameters

The longitudinal input parameters are straightforward and are based on the definitions in Reference 1. An explanation of some of the parameters however, will aid in their use.

Note from Table 2 that the acceleration of gravity is a required input. This parameter varies with altitude to an extent that at very high altitudes its variation should be taken into account. At 100,000 feet, an altitude no longer considered unattainable, the error resulting from using the sea level value is 9.45%.

The distance from the CG to the thrust line, z_t , is included; it affects the characteristic equation only through its influence on M_δ , and it also affects the numerator characteristics if T_{δ_T} is specified. The parameter z_t is seen in the equation for M_δ :

$$M_\delta = \frac{\rho S U_0 \bar{c}}{2 I_{yy}} C_{m_\delta} + \frac{z_t T_{\delta_T}}{I_{yy}} \quad (35)$$

The parameter T_{δ_T} , or the change in thrust with throttle deflection (or RPM), affects the terms X_δ , Z_δ , and M_δ :

$$X_\delta = - \frac{\rho S U_0^2}{2m} C_{D_\delta} + T_{\delta_T} \frac{\cos(\xi + \alpha)}{m} \quad (36)$$

$$Z_\delta = - \frac{\rho S U_0^2}{2m} C_{L_\delta} - T_{\delta_T} \frac{\sin(\xi + \alpha)}{m} \quad (37)$$

Thus, if T_{δ_T} , ξ , and z_t are specified, the transfer functions are not totally elevator terms but include the thrust effects. If only the elevator terms are desired, specify $T_{\delta_t} = z_t = 0$ and input C_{L_δ} , C_{D_δ} , and C_{m_δ} . If the transfer functions with respect to thrust are desired, set $C_{L_\delta} = C_{D_\delta} = C_{m_\delta} = 0$ and define T_{δ_T} , ξ , and z_t . Note that it doesn't matter what dimensions are used for T_{δ_T} because the transfer function that results is a ratio; as long as the ratio is multiplied by the correct units, the equality is not destroyed. Thus

$$\frac{\theta(s)}{\delta_T(s)} \times \delta_T(s) = \theta(s) \quad (38)$$

and as long as the two $\delta_T(s)$'s have the same units, continuity is assured.

The term ξ is the angle of inclination of the thrust axis with respect to the body axis and is defined by Figure 2.

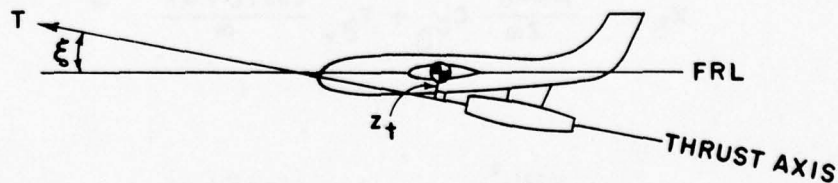


Figure 2. Definition of ξ

C_L and C_D are the trim values and C_{m_T} is defined as

$$C_{m_T} = \frac{z_T T}{\bar{q} s \bar{c}} \quad (39)$$

$$= \frac{z_T}{\bar{c}} \left[C_D / \cos(\xi + \alpha) + C_L \tan(\xi + \alpha) \right] \quad (40)$$

where T , the thrust, at trim in rectilinear flight is

$$T = C_D / \cos(\xi + \alpha) + C_L \tan(\xi + \alpha) \quad (41)$$

The Mach number derivatives are used in the program as opposed to the u derivatives. By definition in Reference 1

$$C_{L_u} = \frac{U_0}{2} \frac{\partial C_L}{\partial u} = \frac{M}{2} C_{L_M} \quad (42)$$

Thus, when C_{L_M} (or C_{D_M} or C_{m_M}) are set in the program, they are multiplied by $\frac{M}{2}$ to evaluate the u derivatives before the calculations proceed. If no Mach derivatives are used, the value for M can be zero.

The angle of attack input is used only in the calculation of X_{δ} and M_{δ} as seen in Equations 36 and 37; α can be zero if T_{δ_T} is zero. A flight path angle of more than 15° should not be specified because of the small-angle assumption.

The variable l_x is included in case the acceleration transfer function is desired at some point other than the CG. The sign on l_x is positive for points forward of the CG and its magnitude is measured in feet.

For body axis derivatives, the angle of attack is necessary as the programs convert all data to the stability axis. Other than that, the previous discussion is valid.

For the dimensional input data, the last three values, V_e , L_α , and n_{z_α} are not needed to obtain the denominator and numerator solutions. The values will be printed out if non-dimensional data are used, since all the values necessary for these calculations are available.

c. Input Data

The method for inserting input data is similar for both programs. In the example, Figure 3, longitudinal, nondimensional stability axis derivatives are given in units of 1/radian. From Table 1, the data identification number is 100, and this number must appear in Columns 1, 2, and 3, respectively, of the first card of each data set (See Figure 3). To get longitudinal coupling numerators, make the number in column 1, card 1, 5 greater - use 5 instead of 0, or 6 instead of 1. Columns 4, 5, and 6 are reserved for the run number; this number may be in any alphanumeric format desired. In this example the number is 15A. Columns 7 through 72 inclusive are used to write anything required to identify the run, such as the altitude, date, or aircraft. Columns 73 through 80 are used for sequencing the cards; these columns are not read by the machine and are used only to identify the card and run number. In the example, the first card is labelled LONG15A1, which means that this is the first card of run 15A, and presents longitudinal data. Card 1 is not included in Table 2. The format of card 1 is the same for all data types. It must be present and contain the data identification numbers in columns 1 through 3. Cards 2 through 7 present the data, and each number shown in Figure 3 corresponds to the parameter included in Table 2 for data type 100. Each datum must appear somewhere in the assigned 12 spaces; therefore, the value for C_{L_q} must appear on the fourth card and must be entirely contained within Columns 37 through 48. Thus, the value for C_{L_q} of run number 15A in Figure 2 is 6.3 per radian.

Column Numbers	1	2	3	4	5	6	7	8	9	0	Run Number	Identification of Run	Data Card Identification
	10	0	5	A	TRANSPORT AIRCRAFT	H=10,000FT	CG=25C	M=.6			TMG/9/1766	LØNGI5A1	Card 1
Data Identification Numbers	4900				.77	745.	.0005873	32.051			LØNGI5A2	Card 2	
	350000				2.0	30.					LØNGI5A3	Card 3	
	.437				6.	6.3	.251				LØNGI5A4	Card 4	
	.025				.03		.0031				LØNGI5A5	Card 5	
					-2.	-5.1	-20.3	-1.04			LØNGI5A6	Card 6	
	1.3										LØNGI5A7	Card 7	

Figure 3. Sample Longitudinal, Stability Axis, Per Radian Input Data. (No Plot)

TABLE 2

LONGITUDINAL INPUT FORMATS

Stability axis, nondimensional

100 = per radian

101 = per degree

1 S(ft ²)	13 $\frac{I_y}{c}$ (ft)	25 M	37 U(ft/sec)	49 ρ (slugs/ft ³)	61 g	73 Card 2
W(lbs)	I_y (slug-ft ²)	z_t (ft)	l_x (ft)	$T_{\delta_T} = \frac{\delta T}{\delta_{Throttle}}$	ξ (deg)	Card 3
C_L	$C_{L\alpha}$	$C_{L\dot{\alpha}}$	C_{Lq}	$C_{L\delta_e}$	C_{LM}	Card 4
C_D	$C_{D\alpha}$	$C_{D\dot{\alpha}}$	C_{Dq}	$C_{D\delta_e}$	C_{DM}	Card 5
$C_{m_T} = \frac{z_t T}{qSc}$	$C_{m\alpha}$	$C_{m\dot{\alpha}}$	C_{mq}	$C_{m\delta_e}$	C_{mM}	Card 6
α (deg)	Γ_o (deg)	Plot Option*				Card 7

Body axis, nondimensional

110 = per radian

111 = per degree

1 S(ft ²)	13 $\frac{I_y}{c}$ (ft)	25 M	37 U(ft/sec)	49 ρ (slugs/ft ³)	61 g	73 Card 2
W(lbs)	I_y (slug-ft ²)	z_t (ft)	l_x (ft)	$T_{\delta_T} = \frac{\delta T}{\delta_{Throttle}}$	ξ (deg)	Card 3
C_N	$C_{N\alpha}$	$C_{N\dot{\alpha}}$	C_{Nq}	$C_{N\delta_e}$	C_{NM}	Card 4
C_x	$C_{x\alpha}$	$C_{x\dot{\alpha}}$	C_{xq}	$C_{x\delta_e}$	C_{xM}	Card 5
$C_{m_T} = \frac{z_t T}{qSc}$	$C_{m\alpha}$	$C_{m\dot{\alpha}}$	C_{mq}	$C_{m\delta_e}$	C_{mM}	Card 6
α (deg)	Γ_o (deg)					Card 7

Stability axis, dimensional = 000

1 X_u	13 Z_u	25 M_u	37 X_w	49 Z_w	61 M_w	73 Card 2
X_w	Z_w	M_w	X_q	Z_q	M_q	Card 3
X_{δ_e}	Z_{δ_e}	M_{δ_e}	U(ft/sec)	g	Γ_o (deg)	Card 4
V_e	L_α	nZ_α	X_{δ_T}	Z_{S_T}	M_{δ_T}	Card 5

*See Table 2 continuation for plot option codes

The format of the input data can be written in only one way. The number 6,753,000, for example, must be written as 6753000. and can appear anywhere in the allowable field. The number -.00745 is written as -.00745 in the allowable field.

The aerodynamic data must all be in consistent units or as indicated in Table 1. All angle inputs are in degrees.

Namelist input is obtained as shown in Table 1. The variable names in the namelist are exactly as printed on the output of the program; that is, flight path angle is called "GAMA", pitch inertia is listed as " I_y ," C_{L_α} is "CLAD" etc. All input options available to the user are available in the namelist form.

The namelist for the longitudinal program is titled "Change" and is used in the following manner:

(1) The first card of each run is written in the usual manner with Column 3 keyed for the namelist input.

(2) The next card must have a blank in Column 1 followed by the characters "\$CHANGE" followed by at least one blank space.

(3) On the same card, the parameters to be changed are written separated by commas. Parameters not entered will remain the same as on the previous run. The namelist is then closed by a dollar sign "\$". There is no restriction on the order in which the parameters being changed must be entered.

(4) Avoid writing in Columns 73-80. If more space is needed, go to another card but leave a blank in Column 1. Do not number cards if more than one is needed for the namelist. Numbering is permitted after the closing "\$".

Namelist Example (Longitudinal):

```
107 SAME CONDITIONS AS ABOVE BUT REDUCED CLQ
$CHANGE CLQ = 6.0$
```

(4) Nondimensional and dimensional (primed and nonprimed) data can be switched from run to run as desired, but the "per radian/per degree" option cannot be switched nor can stability and body axis data be interchanged.

For successive runs using either "LONG." or "LATE.", merely add seven-card sets to the data deck. An end-of-record card inserted between the two kinds of data sets will allow both "LONG." and "LATE." to be run together.

d. Output

The complete longitudinal program and a sample output are presented in Appendix E. The output data is explained in relation to the sample output data sheet. In the example, the first item printed out is: ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS. This title is part of the program and will always appear, followed by the run number (which, in this case, is 15A). The third line contains the exact information that appeared in Columns 7 through 72 on the first card of this data package (see Input Data). Following this run identification is the type of input data and the data itself. The output format is the same as the input format, i.e., the numerical values for s , \bar{c} , M , U_0 , ρ , and g all appear as on the second card of this run.

The dimensional derivatives are then calculated and shown. Note that the values for V_e , L_α , and n_{z_α} are also presented here. The program calculates the coefficients of the denominator and solves for the roots of the quartic equation. The roots of the equation are then printed in the form of $s_1, s_2 = \sigma \pm j\omega$. For the case where the roots are a complex pair, the form is

$$s_1, s_2 = -\zeta\omega_n \pm \omega_n \sqrt{1 - \zeta^2} j \quad (43)$$

and for the case of a real root with zero imaginary part, the form is

$$s = -\frac{1}{T} \quad (44)$$

Comparing these forms with the numbers printed under ROOTS (COMPLEX FORM), it can be seen that the roots to the equation* are

$$s_1, s_2 = -.008943 \pm .1852 j \quad (45)$$

$$s_3, s_4 = -.4507 \pm 2.657 j \quad (46)$$

Now the program must choose which complex pair is the phugoid and which is the short period. This decision is made by comparing the frequencies of the modes. The frequencies are calculated by taking the square root of the sum of the squares or

$$\sqrt{(\zeta\omega_n)^2 + \omega_n^2 (\sqrt{1-\zeta^2})^2} = \omega_n \quad (47)$$

The larger frequency is assumed to be that of the short period. The calculated values are then printed in their proper places, which yields the data seen immediately below the values of the roots. Note here that $ZP = \zeta_{\text{phugoid}}$ and $WP = \omega_{\text{phugoid}}$, etc.

The characteristics of each mode are then calculated and printed. The values are calculated as follows:

$$\text{Period} = P = \frac{2\pi}{\omega_n \sqrt{1-\zeta^2}} \quad [\text{seconds}] \quad (48)$$

$$\text{Time to half amplitude} = 0.69315 / \zeta\omega_n \quad [\text{seconds}] \quad (49)$$

$$\text{Time to one tenth amplitude} = 2.30259 / \zeta\omega_n \quad [\text{seconds}] \quad (50)$$

$$\text{Cycles to half amplitude} = \frac{T_{1/2}}{P} \quad [\text{cycles}] \quad (51)$$

$$\text{Cycles to one tenth amplitude} = \frac{T_{1/10}}{P} \quad [\text{cycles}] \quad (52)$$

*The signed digits following E or D in a number on output specifies the power of 10 by which the number must be multiplied.

where the bracketed quantity shows the dimension. For the case of an unstable oscillatory mode, the program will print the time to reach 2 or 10 times the amplitude. Finally the coefficients of the denominator quartic $As^4 + Bs^3 + Cs^2 + Ds + E$ are printed.

The transfer function numerator calculations are printed on the next page as follows:

Numerator of θ/δ , "THETA PER CONTROL DEFLECTION"
 Numerator of u/δ , "LONGITUDINAL VELOCITY PER CONTROL DEFLECTION"
 Numerator of w/δ , "NORMAL VELOCITY PER CONTROL DEFLECTION"
 Numerator of \dot{h}/δ , "ALTITUDE RATE PER CONTROL DEFLECTION"
 Numerator of a_z/δ , "VERTICAL ACCELERATION PER CONTROL DEFLECTION"
 (the free s in the a_z/δ numerator is not printed.)

Each numerator is labelled and the roots, time constants (or ζ and ω_n), and coefficients are printed. A non-zero value of l_x will cause the normal acceleration numerator terms to be printed; this is for a_z at a distance l_x from the CG.

There is an interesting point to be brought out in regard to the normal velocity per delta elevator transfer function. The values of the roots (complex form) show that in Run No. 111 the third root has an imaginary part of $.8787 \times 10^{-45}$. This, of course, is impossible because a complex root must have a conjugate as another solution (the first two roots do form a complex pair). The imaginary part of the third root is spurious and is stored unintentionally in this location by the subroutine DMULR. Care must be taken to eliminate such erroneous values for the roots. When these values appear, the program will usually ignore them; however, the printed values should always be checked by considering the coefficients of the transfer function. Notice here that the form of the numerator is

$$A_w s^3 + B_w s^2 + C_w s + D_w = 0 \quad (53)$$

and, since all the coefficients are nonzero, three roots will appear either as a real root and a complex pair or as three real roots; anything else is in error. Erroneous values can be spotted easily. Roots with values greater than 10^5 are probably the result of division by one of these very small "noise" numbers.

Another feature of the transfer function print-out is that poles and zeros (roots) with zero real and imaginary parts are not shown. For example, an inherent pole or zero of $s = 0$ is not printed out on either page of the output.

Note that the first set of sample data shows a characteristic equation consisting of an oscillatory mode and two aperiodic modes. The program, by use of FRQCK, has determined that the oscillatory mode is the short period. This interpretation should be treated with caution.

The output symbols are defined as follows:

$$\text{ZSP} = \zeta \text{ short period} \quad (54)$$

$$\text{WSP} = \omega \text{ short period (undamped actual frequency)} \quad (55)$$

$$1/\text{TP1} = (1/\tau_{\text{phugoid}})_1 \quad (56)$$

$$1/\text{TP2} = (1/\tau_{\text{phugoid}})_2 \quad (57)$$

e. Coupling Numerators

The coupling numerators $N_{\delta_e \delta_T}^{\theta_u}$, $N_{\delta_e \delta_T}^{w_u}$, $N_{\delta_e \delta_T}^{\theta_w}$ are obtained by straightforward substitution of columns in the characteristic determinant Δ .

For the coupling numerators involving h , consider the equation

$$h + \frac{\cos \Gamma_0}{s} w - \frac{\sin \Gamma_0}{s} u - u_0 \frac{\cos \Gamma_0}{s} \theta = 0 \quad (58)$$

which is more rigorous than Equations 13 and 14.

An augmented matrix can be formed from this equation and Δ :

$$\begin{bmatrix}
 s - X_u & -(s X_{\dot{w}} + X_w) & g \cos \Gamma_0 - s X_q \\
 -Z_u & s(1 - Z_{\dot{w}}) - Z_w & g \sin \Gamma_0 - s(U_0 + Z_q) \\
 -M_u & -(s M_{\dot{w}} + M_w) & s(s - M_q) \\
 -\frac{\sin \Gamma_0}{s} & \frac{\cos \Gamma_0}{s} & \frac{U_0 \cos \Gamma_0}{s}
 \end{bmatrix}
 \begin{bmatrix}
 u \\
 w \\
 \theta \\
 h
 \end{bmatrix}
 =
 \begin{bmatrix}
 X_{\delta_e} \\
 Z_{\delta_e} \\
 M_{\delta_e} \\
 0
 \end{bmatrix}
 \delta_e + \dots \quad (59)$$

The coupling numerators $N_{\delta_e \delta_T}^{\theta h}$, $N_{\delta_T \delta_e}^{uh}$, $N_{\delta_T \delta_e}^{wh}$ are formed by replacing columns of this 4 x 4 matrix with the indicated control columns, then expanding the resulting matrices in terms of minors of elements of the bottom row. It is seen that 1/s multiplies each coupling numerator in its entirety. In order to indicate that, the printout legends read

"S TIMES THETA TO ELEVATOR, ALTITUDE TO THRUST"

"S TIMES LONGITUDINAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR"

"S TIMES NORMAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR"

One given coupling numerator involves normal acceleration, $N_{\delta_T \delta_e}^{ua_z}$. Now, inertial acceleration is

$$a_z' = sw - U_0 s \theta - l_x s^2 \theta \quad (60)$$

Use this equation to augment Δ , giving

$$N_{\delta_T \delta_e}^{ua_z'} = \begin{vmatrix}
 X_{\delta_T} & -(s X_{\dot{w}} + X_w) & g \cos \Gamma_0 - s X_q & X_{\delta_e} \\
 Z_{\delta_T} & -s(1 - Z_{\dot{w}}) - Z_w & g \sin \Gamma_0 - s(U_0 + Z_q) & Z_{\delta_e} \\
 M_{\delta_T} & -(s M_{\dot{w}} + M_w) & s(s - M_q) & M_{\delta_e} \\
 0 & -s & l_x s^2 + U_0 s & 0
 \end{vmatrix} \quad (61)$$

which is expanded to obtain

$$N_{\delta_T} \frac{u_0 a_z'}{\delta_e} = s \left[N_{\delta_e}^{wu} \delta_T - (l_x s + u_0) N_{\delta_e}^{\theta_u} \delta_T \right] \quad (62)$$

This numerator is labeled

"S TIMES LONGITUDINAL VELOCITY TO THRUST, ACCELERATION TO ELEVATOR"

Again the $s = 0$ root (here a zero instead of a pole) is not given.

Note that here a_z is inertial acceleration. It does not include gravity, as sensed acceleration does. However, account is taken of sensor location forward or aft of the CG.

2. LATERAL-DIRECTIONAL PROGRAM

a. General

This lateral-directional program calculates the coefficients of the three-degree-of-freedom, small-perturbation, lateral-directional equations of motion. These coefficients are then used to calculate the coefficients of the characteristic equation and the numerators of the airplane transfer functions for aileron and rudder inputs. The characteristic equation and the transfer function numerators are factored, and the factors are used to compute several of the more pertinent lateral-directional flying qualities parameters (see Appendix C).

The main portion of the program is limited to computing the characteristic equation and the numerators for the ϕ , β , and ψ transfer functions. The numerator calculations will be bypassed if the control deflection derivatives are all zero. The lateral-directional program was modified extensively to agree with Reference 5.

b. Input Parameters

The lateral-directional program accepts the moments and products of inertia in the body axes, and it converts the inertias to the stability axes with the use of α . This is the only function of the α input; thus, body axes inertias will result if $\alpha = 0$. Use $\alpha = 0$ if inertias are in stability axes. Setting $\alpha \neq 0$ will convert body-axes inertias to stability axes for the calculations.

The parameter l_x is used when the side acceleration transfer function is desired at some point other than the CG.

An interesting point can be brought to light here concerning the use of the δ_A derivatives. Today's aircraft usually employ more than one roll axis control, such as aileron and spoilers. In this case, using only one of the control derivatives as the input is unrealistic because this is not the way the aircraft will behave. The method that has been employed successfully is to convert the control power to the wheel throw or $C_{l\delta_w}$ etc., as follows:

$$C_{l\delta_w} = C_{l\delta_A} \frac{\delta_A}{\delta_w} + C_{l\delta_s} \frac{\delta_s}{\delta_w} + \dots \quad (63)$$

and

$$C_{n\delta_w} = C_{n\delta_A} \frac{\delta_A}{\delta_w} + C_{n\delta_s} \frac{\delta_s}{\delta_w} + \dots \quad (64)$$

and

$$C_{y\delta_w} = C_{y\delta_A} \frac{\delta_A}{\delta_w} + C_{y\delta_s} \frac{\delta_s}{\delta_w} + \dots \quad (65)$$

and then enter these values for $C_{l\delta_a}$, $C_{n\delta_a}$, and $C_{y\delta_a}$.

Aerodynamic Data (See Figures 4, 5, and Table 3)

Using option 000,010 or 100, the aerodynamic data may be input in stability axes or body axes as follows: (See Figures 4 and 5 and Table 3).

$$\text{Input in Stability Axis System} \quad \alpha_A = 0 \quad (66)$$

$$\text{Input in Body Axis System} \quad \alpha_A = \alpha_{\text{TRIM}} - i_w \quad (67)$$

Inertial Data (See Figures 4 and 5)

Using Option 100, the inertia data may be input in body axes or an arbitrary axis system as follows:

$$\text{Input in Stability Axis System} \quad \alpha_I = 0 \quad (68)$$

$$\text{Input in Body Axis System} \quad \alpha_I = \alpha_{\text{TRIM}} - i_w \quad (69)$$

$$\text{Input in Arbitrary Axis System} \quad \alpha_I = \alpha_I \quad (70)$$

Output Axes System (See Figures 4 and 5)

Using Option 000,010 or 100, the output may be referred to the stability, body, or an arbitrary axis system as follows:

$$\text{Output in Stability Axis System} \quad \alpha_x = 0 \quad (71)$$

$$\text{Output in Body Axis System} \quad \alpha_x = \alpha_{w\text{TRIM}} - i_w \quad (72)$$

$$\text{Output in Arbitrary Axis System} \quad \alpha_x = \alpha_x \quad (73)$$

The remainder of the derivatives seen in Table 4 should be self-explanatory.

c. Input Data

The method of entering lateral-directional data is similar to that of the longitudinal program. However, Columns 7, 8, and 9 on Card 1 are also used for program control.

The lateral-directional computer program can provide time histories for a rudder or aileron step plus the MIL-F-8785B parameters. Angle of attack selections for body, stability, inertia, and arbitrary axes are included, as is a plot option and the ability to input the attitude and control derivatives as per degree and rate derivatives as per radian. (See Tables 3 and 3A.)

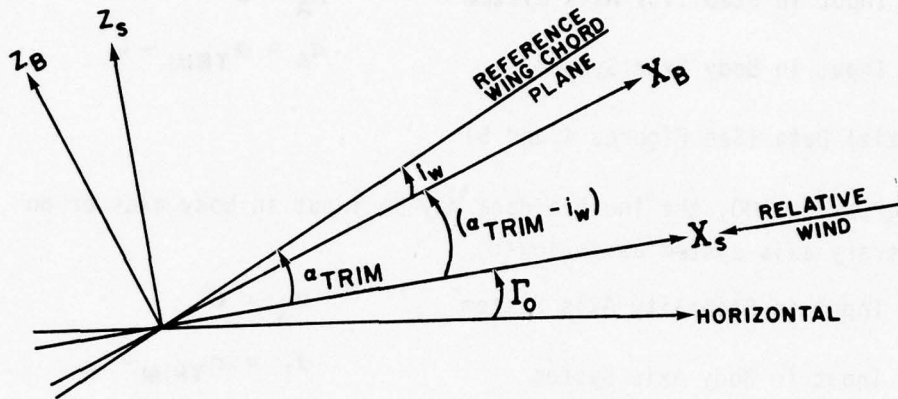


Figure 4. Conventional Notation

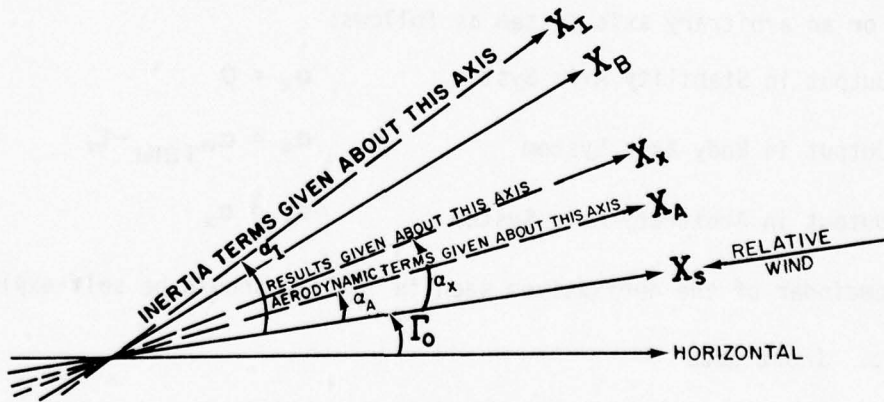


Figure 5. Program Notation

TABLE 3
LATERAL-DIRECTIONAL INPUT DATA

Data Type	Units	Axis	Data Identification Number	Options
Dimensional, primed		stability	101	
		stability	100	
Nondimensional	per radian	stability	000	
	per degree	stability	010	
	$\delta_A, \delta_R, \beta$ per degree p, r, β per radian	stability	020	
	β per degree, all others per radian	stability	030	
	control derivatives per degree, all others per radian	stability	040	
	per radian	stability	050	Namelist
	per degree	stability	060	Namelist
	$\delta_A, \delta_R, \beta$ per degree p, r, β per radian	stability	070	Namelist
	β per degree, all others per radian	stability	080	Namelist
	control derivatives per degree, all others per radian	stability	090	Namelist

Note: To get lateral-directional coupling numerators, add 5 to the first digit of the Data Identification Number.

Table 3 shows that all data must be in the stability axes; this is not entirely true since the primary difference is in the angle of attack. The lateral-directional program transfers the input body-axis inertias to stability-axis inertias; therefore, specifying an $\alpha = 0$ will yield an effective set of derivatives for body axis.

TABLE 3A
LATERAL-DIRECTIONAL OPTIONS

OPTIONS	OUTPUT
001	Basic calculations (roots of characteristic equation), ϕ , β , and ψ transfer function numerators, modal characteristics including $ \phi/\beta $, $ \phi / v_e $ and $\omega_{DR}^2 \phi/\beta _{DR}$
002	Basic calculations plus p and β modal response coefficients, p_{osc}/p_{av} , p_2/p_1 , ψ_β , $\Delta\beta_{MAX}$, and K_d/K_{SS} for a unit step lateral control input, ϕ_{osc}/ϕ_{AV} , ψ'_β for a unit impulse lateral control input and $\frac{1}{2} p/\beta$ for the free Dutch roll oscillation.
-02	Options 001 and 002 plus time histories of β , ϕ , and p for an aileron and for a rudder step.
003	Options 001 with the acceleration transfer function at the l_x distance from the CG.

NOTE: The option codes shown in this Table are placed in Columns 7, 8, and 9 of Card Number 1.

TABLE 2 (CONT'D)

Plot Options - Card 7

Code (Col. 25)	Options
0 (Blank)	No plot
1.	Tabulation of time history (lateral-directional only)
PLT	Namelist option

TABLE 4

LATERAL-DIRECTIONAL INPUT FORMATS

Nondimensional, unprimed

000 = per radian

010 = per degree

1 ρ (slugs/ft ³)	13 U(ft/sec)	25 S(ft ²)	37 W(lbs)	49 b(ft)	61 I_x (slug-ft ²)	73 Card 2
I_z (slug-ft ²)	I_{xz} (slug-ft ²)	g	α_I (deg)	Γ_o (deg)	ℓ_x (ft)	Card 3
C_{y_β}	C_{y_β}	C_{y_p}	C_{y_r}	$C_{y_{\delta A}}$	$C_{y_{\delta R}}$	Card 4
C_{ℓ_β}	C_{ℓ_β}	C_{ℓ_p}	C_{ℓ_r}	$C_{\ell_{\delta A}}$	$C_{\ell_{\delta R}}$	Card 5
C_{n_β}	C_{n_β}	C_{n_p}	C_{n_r}	$C_{n_{\delta A}}$	$C_{n_{\delta R}}$	Card 6
α_A	α_x	PLT*				Card 7

Dimensional

100 = unprimed

1 U(ft/sec)	13 g	25 α_I (deg)	37 Γ_o (deg)	49 ℓ_x (ft)	61 I_x (slug-ft ²)	73 Card 2
I_z (slug-ft ²)	I_{xz} (slug-ft ²)	Y_β	Y_β	Y_p	Y_r	Card 3
$Y_{\delta A}$	$Y_{\delta R}$	L_β	L_β	L_p	L_r	Card 4
$L_{\delta A}$	$L_{\delta R}$	N_β	N_β	N_p	N_r	Card 5
$N_{\delta A}$	$N_{\delta R}$	α_A	α_x	PLT*		Card 6

Stability axis, dimensional

101 = primed

1 U(ft/sec)	13 g	25 Γ_o (deg)	37 ℓ_x (ft)	49 Y_β	61 Y_β	73 Card 2
Y_p	Y_r	$Y_{\delta A}$	$Y_{\delta R}$	L'_β	L'_β	Card 3
L'_p	L'_r	$L'_{\delta A}$	$L'_{\delta R}$	N'_β	N'_β	Card 4
N'_p	N'_r	$N'_{\delta A}$	$N'_{\delta R}$	PLT*		Card 5

*See Table 3A for PLT option codes

To obtain a plot of the time histories provided by option -02, the PLT space on the input data cards is used. PLT has a value of one for all sets of data (runs). For all other options or, if no plot is desired with option -02, PLT is zero or left blank.

When using the namelist option the variable names are exactly as printed on the output of the program; that is, flight path angle is called "GAMA", roll inertia is listed as "IXB", $C_{\alpha\beta}$ is "CLBD", etc. all input options available to the user are given in the namelist form.

The namelist for the lateral-directional program is titled "Change" and is used in the following manner:

- 1) The first card of each run is written in the usual manner with Column 2 (lateral-directional) keyed for the namelist input.
- 2) The next card must have a blank in Column 1 followed by the characters "\$CHANGE" followed by at least one blank space.
- 3) On the same card, the values of the parameters to be changed are written, separated by commas. Parameters not entered will remain the same value as on the previous run. The namelist is then closed by a dollar sign "\$". There is no restriction on the order in which the parameters being changed must be entered on the change card.

Namelist Example: (Lateral-Directional)

```
070 -02 SAME CONDITIONS AS ABOVE BUT REDUCED CNB AND CNR
$CHANGE CNB = .0009, CNR = -.22$
```

- 4) Nondimensional and dimensional (primed and nonprimed) can be switched from run to run at will (however, the per radian/per degree option cannot be switched). This may be of use in studies if the data are presented in nondimensional form and the effects of a variation of dimensional parameters are to be considered.

d. Output

The complete lateral-directional program and a sample output are presented in Appendix E. The output data is explained in relation to the sample output data sheet (pages 140 through 151). In the example the first item printed out is: ROOTS OF A/C LATERAL DIRECTIONAL TRANSFER FUNCTIONS. This title is part of the program and will always appear, followed by the run number. The third line contains the exact information that appeared in Columns 7 through 72 on the first card of this data package (see Input Data). Following this run identification is the type of input data and the data itself. The output format is the same as the input format, i.e., the numerical values for p , U_0 , S , W , b , and I_x all appear as on the second card of the input data for this run.

The input data is read and converted to dimensional primed data, if necessary, and the primed and unprimed data are then printed. Then the denominator characteristics are calculated and printed. The five roots listed include the one which always occurs at $s = 0$ (Equation C-18). The program does not contain a frequency check because if one complex pair appears it is assumed to be the Dutch roll mode. The roll-spiral mode may couple and the Dutch roll may split up into two real roots; when this occurs, the output sheet will print the ζ and ω and label them Dutch roll. Thus, care must be taken when values indicate that this has occurred. Examining the characteristics (ζ and ω) and the complex forms of the roots should indicate which mode is coupled.

The case of two sets of complex conjugates is not covered because it occurs infrequently. When it does occur, an analogy will exist between the Dutch roll and the longitudinal short period, and between the coupled roll-spiral and the phugoid. Thus the mode can be identified by inspection.

When the solution to the stability quintic contains two real roots, the program assumes that the smaller of the two corresponding time constants (absolute value) is the roll subsidence mode. Thus, if the spiral mode has a smaller time constant than the roll mode, its value will appear as a TR on output. This does not occur often and is immediately recognized. Also, as in the longitudinal deck, roots with negative real parts are stable.

Among the denominator characteristics listed are:

TS τ_s , spiral-mode time constant
 TR τ_R , roll-mode time constant
 WDR undamped natural frequency of Dutch roll mode
 WDDR damped frequency of Dutch roll mode.

A few other Dutch roll modal parameters (not dependent upon the input) are also printed with the denominator characteristics:

$|\phi|/|\beta|$ "PHI TO BETA RATIO"
 $|\phi|/|v_e|$ "PHI TO EQUIV VEL"
 $\omega_d^2 |\phi|/|\beta|$ "FREQ SQUARED TIMES PHI TO BETA RATIO"

After the denominator characteristics are printed, the transfer function numerators are calculated. For example, the yaw rate to control deflection transfer function has a variable that is labelled $1/TR$, where the R stands for the yaw rate (r), and not the roll time constant.

The ω_ϕ/ω_D calculation needs ω_{DR} from the denominator characteristics and the ϕ/β calculation is based entirely on denominator characteristics.

COUPLING NUMERATORS:

The following are printed when the first digit of Card 1 has been increased by 5 from the value in Table 3:

$$N \frac{\phi \beta}{\delta_A \delta_R}$$

$$N \frac{\phi \psi}{\delta_A \delta_R}$$

$$N \frac{\psi \beta}{\delta_A \delta_R}$$

$$N \frac{\phi a'_y}{\delta_A \delta_R}$$

$$N \frac{\psi a'_y}{\delta_A \delta_R}$$

$$N \frac{a'_y \beta}{\delta_A \delta_R}$$

Here a'_y is sensed lateral acceleration:

$$a'_y = u_0 \dot{\beta} + u_0 r + l_x \dot{r} - (g \cos \Gamma_0) \dot{\phi} - (g \sin \Gamma_0) \dot{\psi} \quad (74)$$

If the first coefficient of the phi to aileron, acceleration to rudder; psi to aileron, acceleration to rudder; or acceleration to aileron, beta to rudder polynomial is equal to zero, the same value will appear in the printout for both the C and D coefficients. In this case D coefficient should be disregarded and it should be recognized that a second order polynomial is being evaluated. The phi to aileron, acceleration to rudder and the psi to aileron, acceleration to rudder numerators are third order in s, but the acceleration to aileron, beta to rudder is fourth order with the last coefficient equal to zero.

SECTION IV

CONCLUDING REMARKS

These three-degree-of-freedom programs (with the exception of the coupling numerator options) have been operational for many years and provide an easy method of obtaining uncoupled aircraft dynamic characteristics from physical and stability and control parameters. The coupling numerator options have been present in the program for many years but the codes to access them were not documented. Consequently, this portion of programs has not been as well checked out.

APPENDIX A

TRANSFER EQUATIONS

Equations for Transfer of Interim Data From (I) to (X) Axis System:

$$\alpha_1 = \alpha_I - \alpha_X \quad (A-1)$$

$$I_X = I_{X_I} \cos^2 \alpha_1 + I_{Z_I} \sin^2 \alpha_1 - I_{XZ_I} \sin(2\alpha_1) \quad (A-2)$$

$$I_Z = I_{Z_I} \cos^2 \alpha_1 + I_{X_I} \sin^2 \alpha_1 + I_{XZ_I} \sin(2\alpha_1) \quad (A-3)$$

$$I_{XZ} = I_{XZ_I} \cos(2\alpha_1) + \frac{1}{2} (I_{X_I} - I_{Z_I}) \sin(2\alpha_1) \quad (A-4)$$

Equations for Transfer of Aerodynamic Data from (A) to (X) Axis System:

$$\alpha_2 = (\alpha_X - \alpha_A) \quad (A-5)$$

$$C_{L_P} = C_{L_{P_A}} \cos^2 \alpha_2 + C_{n_{r_A}} \sin^2 \alpha_2 - (C_{L_{r_A}} + C_{n_{p_A}}) \sin \alpha_2 \cos \alpha_2 \quad (A-6)$$

$$C_{L_r} = C_{L_{r_A}} \cos^2 \alpha_2 - C_{n_{p_A}} \sin^2 \alpha_2 + (C_{L_{p_A}} - C_{n_{r_A}}) \sin \alpha_2 \cos \alpha_2 \quad (A-7)$$

$$C_{L_\beta} = C_{L_{\beta_A}} \cos \alpha_2 - C_{n_{\beta_A}} \sin \alpha_2 \quad (A-8)$$

$$C_{L_{\dot{\beta}}} = C_{L_{\dot{\beta}_A}} \cos \alpha_2 - C_{n_{\dot{\beta}_A}} \sin \alpha_2 \quad (A-9)$$

$$C_{L_\delta} = C_{L_{\delta_A}} \cos \alpha_2 - C_{n_{\delta_A}} \sin \alpha_2 \quad (A-10)$$

$$C_{n_p} = C_{n_{p_A}} \cos^2 \alpha_2 - C_{L_{r_A}} \sin^2 \alpha_2 + (C_{L_{p_A}} - C_{n_{r_A}}) \sin \alpha_2 \cos \alpha_2 \quad (A-11)$$

$$C_{n_r} = C_{n_{r_A}} \cos^2 \alpha_2 + C_{L_{p_A}} \sin^2 \alpha_2 + (C_{L_{r_A}} + C_{n_{p_A}}) \sin \alpha_2 \cos \alpha_2 \quad (A-12)$$

$$C_{n_\beta} = C_{n_{\beta_A}} \cos \alpha_2 + C_{L_{\beta_A}} \sin \alpha_2 \quad (A-13)$$

$$C_{n\dot{\beta}} = C_{n\dot{\beta}_A} \cos \alpha_2 + C_{\ell\dot{\beta}_A} \sin \alpha_2 \quad (\text{A-14})$$

$$C_{n\delta} = C_{n\delta_A} \cos \alpha_2 + C_{\ell\delta_A} \sin \alpha_2 \quad (\text{A-15})$$

$$C_{y_p} = C_{y_{p_A}} \cos \alpha_2 - C_{y_{r_A}} \sin \alpha_2 \quad (\text{A-16})$$

$$C_{y_r} = C_{y_{r_A}} \cos \alpha_2 + C_{y_{p_A}} \sin \alpha_2 \quad (\text{A-17})$$

$$C_{y\dot{\beta}} = C_{y\dot{\beta}_A} \quad (\text{A-18})$$

$$C_{y\beta} = C_{y\beta_A} \quad (\text{A-19})$$

$$C_{y\delta} = C_{y\delta_A} \quad (\text{A-20})$$

APPENDIX B

LONGITUDINAL EQUATIONS OF MOTION AND TRANSFER FUNCTIONS

 ΣF_x

$$\dot{u} + g\theta \cos \Gamma_0 = \frac{1}{m} T_{\delta_r} \delta_{RPM} \cos \xi + X_u u + X_q q + X_\alpha \alpha + X_{\dot{\alpha}} \dot{\alpha} + X_{\delta_e} \delta_e \quad (B-1)$$

 ΣF_z

$$U_0 \dot{\alpha} - U_0 q + g\theta \sin \Gamma_0 = -\frac{1}{m} T_{\delta_r} \delta_{RPM} \sin \xi + Z_u u + Z_q q + Z_{\delta_e} \delta_e \quad (B-2)$$

 ΣM

$$\dot{\alpha} = \frac{Z_\uparrow}{I_{yy}} T_{\delta_r} \delta_{RPM} + M_u u + M_q q + M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_{\delta_e} \delta_e \quad (B-3)$$

Taking the Laplace transform of 1, 2, and 3 and assembling in matrix notation yields (see Reference 6) for a single control input

$$\begin{bmatrix} s - X_u & -(sX_{\dot{\alpha}} + X_\alpha) & g \cos \Gamma_0 - sX_q \\ -Z_u & s(U_0 - Z_{\dot{\alpha}}) - Z_\alpha & g \sin \Gamma_0 - s(U_0 + Z_q) \\ -M_u & -(sM_{\dot{\alpha}} + M_\alpha) & s(s - M_q) \end{bmatrix} \begin{bmatrix} u(s) \\ \alpha(s) \\ \theta(s) \end{bmatrix} = \begin{bmatrix} X_\delta \delta(s) \\ Z_\delta \delta(s) \\ M_\delta \delta(s) \end{bmatrix} \quad (B-4)$$

where s is the Laplacian operator and $X_\delta \delta(s)$, etc., symbolizes any unit impulse forcing function such as $X_{\delta_e} \delta_e(s)$, $T_\delta \cos \zeta \delta(s)$, or $X_{\delta_{SB}} \delta_{SB}(s)$, etc.

The characteristic equation of motion is the determinate solution of the matrix.

$$\Delta = \begin{bmatrix} s - X_u & -sX_{\dot{\alpha}} - X_\alpha & g \cos \Gamma_0 - sX_q \\ -Z_u & sU_0 - sZ_{\dot{\alpha}} - Z_\alpha & g \sin \Gamma_0 - sU_0 - sZ_q \\ -M_u & -sM_{\dot{\alpha}} - M_\alpha & s^2 - sM_q \end{bmatrix} \quad (B-5)$$

$$\begin{aligned}
\Delta = & (s - X_u) \{ (sU_o - sZ_{\dot{\alpha}} - Z_{\alpha}) (s^2 - sM_q) - (g \sin \Gamma_o - sU_o - sZ_q) (-sM_{\dot{\alpha}} - M_{\alpha}) \} \\
& - Z_u \{ (g \cos \Gamma_o - sX_q) (-sM_{\dot{\alpha}} - M_{\alpha}) - (-sX_{\dot{\alpha}} - X_{\alpha}) (s^2 - sM_q) \} \\
& - M_u \{ (-sX_{\dot{\alpha}} - X_{\alpha}) (g \sin \Gamma_o - sU_o - sZ_q) - (sU_o - sZ_{\dot{\alpha}} - Z_{\alpha}) (g \cos \Gamma_o - sX_q) \} \quad (B-6)
\end{aligned}$$

Expanding,

$$\begin{aligned}
\Delta = & (s - X_u) \{ s^3 U_o - s^3 Z_{\dot{\alpha}} - s^2 Z_{\alpha} - s^2 U_o M_q + s^2 Z_{\dot{\alpha}} M_q + s Z_{\alpha} M_q + s M_{\dot{\alpha}} g \sin \Gamma_o \\
& + M_{\alpha} g \sin \Gamma_o - s^2 U_o M_{\dot{\alpha}} - s U_o M_{\alpha} - s^2 Z_q M_{\dot{\alpha}} - s Z_q M_{\alpha} \} \\
& - Z_u \{ -s M_{\dot{\alpha}} g \cos \Gamma_o - M_{\alpha} g \cos \Gamma_o + s^2 X_q M_{\dot{\alpha}} + s X_q M_{\alpha} + s^3 X_{\dot{\alpha}} \\
& - s^2 X_{\dot{\alpha}} M_q + s^2 X_{\alpha} - s X_{\alpha} M_q \} \\
& - M_u \{ -s X_{\dot{\alpha}} g \sin \Gamma_o + s^2 X_{\dot{\alpha}} U_o + s^2 Z_q X_{\dot{\alpha}} - X_{\alpha} g \sin \Gamma_o + s X_{\alpha} U_o + s Z_q X_{\alpha} \\
& - s U_o g \cos \Gamma_o + s^2 X_q U_o + s Z_{\dot{\alpha}} g \cos \Gamma_o - s^2 Z_{\dot{\alpha}} X_q + Z_{\alpha} g \cos \Gamma_o - s Z_{\alpha} X_q \} \quad (B-7)
\end{aligned}$$

$$\begin{aligned}
\Delta = & s^4 U_o - s^4 Z_{\dot{\alpha}} - s^3 Z_{\alpha} - s^3 U_o M_q + s^3 Z_{\dot{\alpha}} M_q + s^2 Z_{\alpha} M_q + s^2 M_{\dot{\alpha}} g \sin \Gamma_o \\
& + s M_{\alpha} g \sin \Gamma_o - s^3 U_o M_{\dot{\alpha}} - s^2 U_o M_{\alpha} - s^3 Z_q M_{\dot{\alpha}} - s^2 Z_q M_{\alpha} \\
& - s^3 X_u U_o + s^3 Z_{\dot{\alpha}} X_u + s^2 Z_{\alpha} X_u + s^2 X_u U_o M_q - s^2 Z_{\dot{\alpha}} X_u M_q - s Z_{\alpha} X_u M_q \\
& - s X_u M_{\dot{\alpha}} g \sin \Gamma_o - X_u M_{\alpha} g \sin \Gamma_o + s^2 X_u U_o M_{\dot{\alpha}} + s X_u U_o M_{\alpha} + s^2 Z_q X_u M_{\dot{\alpha}} \\
& + s Z_q X_u M_{\alpha} + s Z_u M_{\dot{\alpha}} g \cos \Gamma_o + Z_u M_{\alpha} g \cos \Gamma_o - s^2 Z_u X_q M_{\dot{\alpha}} \\
& - s Z_u X_q M_{\alpha} - s^3 Z_u X_{\dot{\alpha}} + s^2 Z_u X_{\dot{\alpha}} M_q - s^2 Z_u X_{\alpha} + s Z_u X_{\alpha} M_q \\
& + s X_{\dot{\alpha}} M_u g \sin \Gamma_o - s^2 X_{\dot{\alpha}} U_o M_u - s^2 Z_q X_{\dot{\alpha}} M_u + X_{\alpha} M_u g \sin \Gamma_o - s X_{\alpha} U_o M_u \\
& - s Z_q X_{\alpha} M_u + s U_o M_u g \cos \Gamma_o - s^2 X_q U_o M_u - s Z_{\dot{\alpha}} M_u g \cos \Gamma_o + s^2 Z_{\dot{\alpha}} X_q M_u \\
& - Z_{\alpha} M_u g \cos \Gamma_o + s Z_{\alpha} X_q M_u \quad (B-8)
\end{aligned}$$

This simplified to:

$$\Delta = As^4 + Bs^3 + Cs^2 + Ds + E \quad (B-9)$$

when

$$A = U_0 - Z_{\dot{\alpha}} \quad (B-10)$$

$$B = -Z_{\alpha} - U_0 M_q + Z_{\dot{\alpha}} M_q - U_0 M_{\dot{\alpha}} - Z_q M_{\dot{\alpha}} - X_u U_0 + Z_{\dot{\alpha}} X_u - Z_u X_{\dot{\alpha}} \quad (B-11)$$

$$C = Z_{\alpha} M_q + M_{\dot{\alpha}} g \sin \Gamma_0 - Z_q M_{\alpha} + Z_{\alpha} X_u + X_u U_0 M_q - Z_{\dot{\alpha}} X_u M_q \\ + X_u U_0 M_{\dot{\alpha}} + Z_q X_u M_{\dot{\alpha}} - U_0 M_{\alpha} - Z_u X_q M_{\dot{\alpha}} - Z_u X_{\alpha} \\ - X_{\dot{\alpha}} U_0 M_u - Z_q X_{\dot{\alpha}} M_u - X_q U_0 M_u + Z_{\dot{\alpha}} X_q M_u + Z_u X_{\dot{\alpha}} M_q \quad (B-12)$$

$$D = M_{\alpha} g \sin \Gamma_0 - Z_{\alpha} X_u M_q - X_u M_{\dot{\alpha}} g \sin \Gamma_0 + X_u U_0 M_{\dot{\alpha}} + Z_q X_u M_{\alpha} \\ + Z_u M_{\dot{\alpha}} g \cos \Gamma_0 - Z_u X_q M_{\alpha} + Z_u X_{\alpha} M_q + X_{\dot{\alpha}} M_u g \sin \Gamma_0 \\ - X_{\alpha} U_0 M_u - Z_q X_{\alpha} M_u + U_0 M_u g \cos \Gamma_0 - Z_{\dot{\alpha}} M_u g \cos \Gamma_0 + Z_{\alpha} X_q M_u \quad (B-13)$$

$$E = -X_u M_{\alpha} g \sin \Gamma_0 + Z_u M_{\alpha} g \cos \Gamma_0 + X_{\alpha} M_u g \sin \Gamma_0 - Z_{\alpha} M_u g \cos \Gamma_0 \quad (B-14)$$

Note that in Section II-1 and the computer printouts, Δ and the longitudinal numerator polynomials of this appendix have been divided by U_0 . That gives a consistent set of transfer functions for which the leading coefficient A of Δ is $1 - Z_w$ (or, when Z_w is zero, just 1). But the printout gives the transfer function of normal velocity (w) rather than angle of attack ($\alpha = w/U_0$) per control deflection; so the w/δ numerator printed out is the α/δ numerator of this appendix.

From the matrix (Equation B-4), three basic transfer functions can be derived

$$\frac{u(s)}{\delta_e(s)} = \frac{\begin{vmatrix} X_\delta & -(sX_{\dot{\alpha}} + X_\alpha) & g \cos \Gamma_0 - sX_q \\ Z_\delta & s(U_0 - Z_{\dot{\alpha}}) - Z_\alpha & g \sin \Gamma_0 - s(U_0 + Z_q) \\ M_\delta & -(sM_{\dot{\alpha}} + M_\alpha) & s(s - M_q) \end{vmatrix}}{\Delta} \quad (B-15)$$

The numerator is expanded as follows:

$$\begin{aligned} \text{NUM} = & X_\delta \left\{ [s(U_0 - Z_{\dot{\alpha}}) - Z_\alpha] [s(s - M_q)] + [g \sin \Gamma_0 - s(U_0 + Z_q)] [sM_{\dot{\alpha}} + M_\alpha] \right\} \\ & - Z_\delta \left\{ -[sX_{\dot{\alpha}} + X_\alpha] [s(s - M_q)] + [g \cos \Gamma_0 - sX_q] [sM_{\dot{\alpha}} + M_\alpha] \right\} \\ & - M_\delta \left\{ [sX_{\dot{\alpha}} + X_\alpha] [g \sin \Gamma_0 - s(U_0 + Z_q)] + [g \cos \Gamma_0 - sX_q] [s(U_0 - Z_{\dot{\alpha}}) - Z_\alpha] \right\} \quad (B-16) \end{aligned}$$

Expanding,

$$\begin{aligned} \text{NUM} = & X_\delta \left\{ (sU_0 - sZ_{\dot{\alpha}} - Z_\alpha)(s^2 - sM_q) + (g \sin \Gamma_0 - sU_0 - sZ_q)(sM_{\dot{\alpha}} + M_\alpha) \right\} \\ & - Z_\delta \left\{ (-sX_{\dot{\alpha}} - X_\alpha)Xs^2 - sM_q + (g \cos \Gamma_0 - sX_q)XsM_{\dot{\alpha}} + M_\alpha \right\} \\ & - M_\delta \left\{ (sX_{\dot{\alpha}} + X_\alpha)Xg \sin \Gamma_0 - sU_0 - sZ_q + (g \cos \Gamma_0 - sX_q)XsU_0 - sZ_{\dot{\alpha}} - Z_\alpha \right\} \quad (B-17) \end{aligned}$$

$$\begin{aligned} \text{NUM} = & X_\delta \left\{ s^3 U_0 - s^2 U_0 M_q - s^3 Z_{\dot{\alpha}} + s^2 Z_{\dot{\alpha}} M_q - s^2 Z_\alpha + s Z_\alpha M_q + s M_{\dot{\alpha}} g \sin \Gamma_0 \right. \\ & \left. + M_\alpha g \sin \Gamma_0 - s^2 U_0 M_{\dot{\alpha}} - s U_0 M_\alpha - s^2 Z_q M_{\dot{\alpha}} - s Z_q M_\alpha \right\} \\ & - Z_\delta \left\{ -s^3 X_{\dot{\alpha}} + s^2 X_{\dot{\alpha}} M_q - s^2 X_\alpha + s X_\alpha M_q + s M_{\dot{\alpha}} g \cos \Gamma_0 + M_\alpha g \cos \Gamma_0 \right. \\ & \left. - s^2 X_q M_{\dot{\alpha}} - s X_q M_\alpha \right\} \\ & - M_\delta \left\{ s X_{\dot{\alpha}} g \sin \Gamma_0 - s^2 X_{\dot{\alpha}} U_0 - s^2 Z_q X_{\dot{\alpha}} + X_\alpha g \sin \Gamma_0 - s X_\alpha U_0 - s Z_q X_\alpha \right. \\ & \left. + s U_0 g \cos \Gamma_0 - s Z_{\dot{\alpha}} g \cos \Gamma_0 - Z_\alpha g \cos \Gamma_0 - s^2 X_q U_0 + s^2 Z_{\dot{\alpha}} X_q + s Z_\alpha X_q \right\} \quad (B-18) \end{aligned}$$

$$\begin{aligned}
 \text{NUM} = & s^3 X_{\delta} U_0 - s^2 X_{\delta} U_0 M_q - s^3 Z_{\alpha} X_{\delta} + s^2 Z_{\alpha} X_{\delta} M_q - s^2 Z_{\alpha} X_{\delta} + s Z_{\alpha} X_{\delta} M_q \\
 & + s X_{\delta} M_{\alpha} g \sin \Gamma_0 + X_{\delta} M_{\alpha} g \sin \Gamma_0 - s^2 X_{\delta} U_0 M_{\alpha} - s X_{\delta} U_0 M_{\alpha} + s^2 Z_{\alpha} X_{\delta} M_{\alpha} \\
 & + s Z_{\alpha} X_{\delta} M_{\alpha} + s^3 Z_{\delta} X_{\alpha} - s^2 Z_{\delta} X_{\alpha} M_q + s^2 Z_{\delta} X_{\alpha} - s Z_{\delta} X_{\alpha} M_q - s Z_{\delta} M_{\alpha} g \cos \Gamma_0 \\
 & - Z_{\delta} M_{\alpha} g \cos \Gamma_0 + s^2 Z_{\delta} X_{\alpha} M_{\alpha} + s Z_{\delta} X_{\alpha} M_{\alpha} - s X_{\alpha} M_{\delta} g \sin \Gamma_0 + s^2 X_{\alpha} U_0 M_{\delta} \\
 & + s^2 Z_{\alpha} X_{\alpha} M_{\delta} - X_{\alpha} M_{\delta} g \sin \Gamma_0 + s X_{\alpha} U_0 M_{\delta} + s Z_{\alpha} X_{\alpha} M_{\delta} - s U_0 M_{\delta} g \cos \Gamma_0 \\
 & + s Z_{\alpha} M_{\delta} g \cos \Gamma_0 + Z_{\alpha} M_{\delta} g \cos \Gamma_0 + s^2 X_{\alpha} U_0 M_{\delta} - s^2 Z_{\alpha} X_{\alpha} M_{\delta} - s Z_{\alpha} X_{\alpha} M_{\delta} \quad (B-19)
 \end{aligned}$$

This simplifies to:

$$\text{NUM} = A s^3 + B s^2 + C s + D \quad (B-20)$$

when

$$A_u = X_{\delta} U_0 - Z_{\alpha} X_{\delta} + Z_{\delta} X_{\alpha} \quad (B-21)$$

$$\begin{aligned}
 B_u = & -X_{\delta} U_0 M_q + Z_{\alpha} X_{\delta} M_q - Z_{\alpha} X_{\delta} - X_{\delta} U_0 M_{\alpha} - Z_{\alpha} X_{\delta} M_{\alpha} - Z_{\delta} X_{\alpha} M_q + Z_{\delta} X_{\alpha} \\
 & + Z_{\delta} X_{\alpha} M_{\alpha} + X_{\alpha} U_0 M_{\delta} + Z_{\alpha} X_{\alpha} M_{\delta} + X_{\alpha} U_0 M_{\delta} - Z_{\alpha} X_{\alpha} M_{\delta} \quad (B-22)
 \end{aligned}$$

$$\begin{aligned}
 C_u = & Z_{\alpha} X_{\delta} M_q + X_{\delta} M_{\alpha} g \sin \Gamma_0 - X_{\delta} U_0 M_{\alpha} - Z_{\alpha} X_{\delta} M_{\alpha} - Z_{\delta} X_{\alpha} M_q - Z_{\delta} M_{\alpha} g \cos \Gamma_0 + Z_{\delta} X_{\alpha} M_{\alpha} \\
 & - X_{\alpha} M_{\delta} g \sin \Gamma_0 + X_{\alpha} U_0 M_{\delta} + Z_{\alpha} X_{\alpha} M_{\delta} - U_0 M_{\delta} g \cos \Gamma_0 + Z_{\alpha} M_{\delta} g \cos \Gamma_0 - Z_{\alpha} X_{\alpha} M_{\delta} \quad (B-23)
 \end{aligned}$$

$$D_u = X_{\delta} M_{\alpha} g \sin \Gamma_0 - Z_{\delta} M_{\alpha} g \cos \Gamma_0 - X_{\alpha} M_{\delta} g \sin \Gamma_0 + Z_{\alpha} M_{\delta} g \cos \Gamma_0 \quad (B-24)$$

The transfer functions for $\alpha(s)/\delta_e(s)$ is derived in a similar manner.

$$\frac{\alpha(s)}{\delta_e(s)} = \frac{\begin{vmatrix} s - X_u & X_q & g \cos \Gamma_0 - s \Gamma_q \\ -Z_u & Z_{\delta} & g \sin \Gamma_0 - s(U_0 + Z_q) \\ -M_u & M_{\delta} & s(s - M_q) \end{vmatrix}}{\Delta} \quad (B-25)$$

$$\begin{aligned} \text{NUM} = & (s - X_u) \{ Z_\delta [s(s - M_q)] - M_\delta [g \sin \Gamma_0 - s(U_0 + Z_q)] \} \\ & + Z_u \{ M_\delta (g \cos \Gamma_0 - sX_q) - X_\delta [s(s - M_q)] \} \\ & - M_u \{ X_\delta [g \sin \Gamma_0 - s(U_0 + Z_q)] - Z_\delta (g \cos \Gamma_0 - sX_q) \} \end{aligned} \quad (\text{B-26})$$

$$\begin{aligned} \text{NUM} = & (s - X_u) \{ s^2 Z_\delta - sZ_\delta M_q - M_\delta g \sin \Gamma_0 + sU_0 M_\delta + sZ_q M_\delta \} \\ & + Z_u \{ M_\delta g \cos \Gamma_0 - sX_q M_\delta - s^2 X_\delta + sX_\delta M_q \} \\ & - M_u \{ X_\delta g \sin \Gamma_0 - sX_\delta U_0 - sZ_q X_\delta - Z_\delta g \cos \Gamma_0 + sZ_\delta X_q \} \end{aligned} \quad (\text{B-27})$$

$$\begin{aligned} \text{NUM} = & s^3 Z_\delta - s^2 Z_\delta M_q - sM_\delta g \sin \Gamma_0 + s^2 U_0 M_\delta + s^2 Z_q M_\delta - s^2 Z_\delta X_u + sZ_\delta X_u M_q \\ & + X_u M_\delta g \sin \Gamma_0 - sX_u U_0 M_\delta - sZ_q X_u M_\delta + Z_u M_\delta g \cos \Gamma_0 - sZ_u X_q M_\delta - s^2 Z_u X_\delta \\ & + sZ_u X_\delta M_q - X_\delta M_u g \sin \Gamma_0 + sX_\delta U_0 M_u + sZ_q X_\delta M_u + Z_\delta M_u g \cos \Gamma_0 - sZ_\delta X_q M_u \end{aligned} \quad (\text{B-28})$$

This simplifies to:

$$\text{NUM} = A_\alpha s^3 + B_\alpha s^2 + C_\alpha s + D_\alpha \quad (\text{B-29})$$

when

$$A_\alpha = Z_\delta \quad (\text{B-30})$$

$$B_\alpha = -Z_\delta M_q + U_0 M_\delta + Z_q M_\delta - Z_\delta X_u - Z_u X_\delta \quad (\text{B-31})$$

$$\begin{aligned} C_\alpha = & -M_\delta g \sin \Gamma_0 + Z_\delta X_u M_q - X_u U_0 M_\delta - Z_q X_u M_\delta - Z_u X_q M_\delta \\ & + Z_u X_\delta M_q + X_\delta U_0 M_u + Z_q X_\delta M_u - Z_\delta X_q M_u \end{aligned} \quad (\text{B-32})$$

$$D_\alpha = X_u M_\delta g \sin \Gamma_0 + Z_u M_\delta g \cos \Gamma_0 - X_\delta M_u g \sin \Gamma_0 + Z_\delta M_u g \cos \Gamma_0 \quad (\text{B-33})$$

It should be pointed out here that the angle of attack transfer function differs from the vertical velocity transfer function by only a gain of U_0 .

For the $\theta(s)/\delta_e(s)$ transfer function

$$\frac{\theta(s)}{\delta_e(s)} = \frac{\begin{vmatrix} s - X_u & -(sX_{\dot{\alpha}} + X_{\alpha}) & X_{\delta} \\ -Z_u & s(U_0 - Z_{\dot{\alpha}}) - Z_{\alpha} & Z_{\delta} \\ -M_u & -(sM_{\dot{\alpha}} + M_{\alpha}) & M_{\delta} \end{vmatrix}}{\Delta} \quad (\text{B-34})$$

$$\begin{aligned} \text{NUM} &= X_{\delta} \left\{ -Z_u (-sM_{\dot{\alpha}} - M_{\alpha}) + M_u \left[s(U_0 - Z_{\dot{\alpha}}) - Z_{\alpha} \right] \right\} \\ &+ Z_{\delta} \left\{ M_u (sX_{\dot{\alpha}} + X_{\alpha}) + (s - X_u)(sM_{\dot{\alpha}} + M_{\alpha}) \right\} \\ &+ M_{\delta} \left\{ (s - X_u) \left[s(U_0 - Z_{\dot{\alpha}}) - Z_{\alpha} \right] - Z_u (sX_{\dot{\alpha}} + X_{\alpha}) \right\} \end{aligned} \quad (\text{B-35})$$

$$\begin{aligned} \text{NUM} &= X_{\delta} \left[+sZ_u M_{\dot{\alpha}} + Z_u M_{\alpha} + sU_0 M_u - sZ_{\dot{\alpha}} M_u - Z_{\alpha} M_u \right] \\ &+ Z_{\delta} \left[+sX_{\dot{\alpha}} M_u + X_{\alpha} M_u + s^2 M_{\dot{\alpha}} + sM_{\alpha} - sX_u M_{\dot{\alpha}} - X_u M_{\alpha} \right] \\ &+ M_{\delta} \left[s^2 U_0 - s^2 Z_{\dot{\alpha}} - sZ_{\alpha} - sX_u U_0 + sZ_{\dot{\alpha}} X_u + Z_{\alpha} X_u - sX_{\dot{\alpha}} Z_u - X_{\alpha} Z_u \right] \end{aligned} \quad (\text{B-36})$$

$$\begin{aligned} \text{NUM} &= +sZ_u X_{\delta} M_{\dot{\alpha}} + Z_u X_{\delta} M_{\alpha} + sX_{\delta} U_0 M_u - sZ_{\dot{\alpha}} X_{\delta} M_u - Z_{\alpha} X_{\delta} M_u + sZ_{\delta} X_{\dot{\alpha}} M_u \\ &+ Z_{\delta} X_{\alpha} M_u + s^2 Z_{\delta} M_{\dot{\alpha}} + Z_{\delta} M_{\alpha} - sZ_{\delta} X_u M_{\dot{\alpha}} - Z_{\delta} X_u M_{\alpha} + s^2 U_0 M_{\delta} \\ &- s^2 Z_{\dot{\alpha}} M_{\delta} - sZ_{\alpha} M_{\delta} - sX_u U_0 M_{\delta} + sZ_{\dot{\alpha}} X_u M_{\delta} + Z_{\alpha} X_u M_{\delta} - sZ_u X_{\dot{\alpha}} M_{\delta} - Z_u X_{\alpha} M_{\delta} \end{aligned} \quad (\text{B-37})$$

This simplifies to

$$\text{NUM} = A_{\theta} S^2 + B_{\theta} S + C_{\theta} \quad (\text{B-38})$$

when

$$A_{\theta} = +Z_{\delta} M_{\dot{\alpha}} + U_0 M_{\delta} - Z_{\dot{\alpha}} M_{\delta} \quad (\text{B-39})$$

$$\begin{aligned} B_{\theta} &= +Z_u X_{\delta} M_{\dot{\alpha}} + X_{\delta} U_0 M_u - Z_{\dot{\alpha}} X_{\delta} M_u + Z_{\delta} X_{\alpha} M_u + Z_{\delta} M_{\alpha} - Z_{\delta} X_u M_{\dot{\alpha}} \\ &- Z_{\alpha} M_{\delta} - X_u U_0 M_{\delta} + Z_{\dot{\alpha}} X_u M_{\delta} - Z_u X_{\dot{\alpha}} M_{\delta} \end{aligned} \quad (\text{B-40})$$

$$C_{\theta} = +Z_u X_{\delta} M_{\alpha} - Z_{\alpha} X_{\delta} M_u + Z_{\delta} X_{\alpha} M_u - Z_{\delta} X_u M_{\alpha} + Z_{\alpha} X_u M_{\delta} - Z_u X_{\alpha} M_{\delta} \quad (\text{B-41})$$

Again, note that in the body of this report and the computer printout the polynomials of this appendix have been divided by U_0 .

Coupling Numerators

Coupling numerators were devised by McRuer, Ashkenas, and Pass to aid in analysis and synthesis of multiloop control systems. The method is detailed in Reference 7, Section 3-5, with longitudinal applications given in Sections 5-10.

Consider, for example, regulation of pitch attitude and airspeed with elevator, altitude rate with thrust. A simplified representation of elevator control is, with α_{ij} 's representing polynomials in s ,

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 0 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ \dot{h} \end{bmatrix} = \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{bmatrix} (\delta e_i - \gamma_{\theta} \theta - \gamma_u u) + \begin{bmatrix} X_{\delta T} \\ Z_{\delta T} \\ M_{\delta T} \\ 0 \end{bmatrix} (-\gamma_{\dot{h}} \dot{h}) \quad (B-42)$$

where δe_c is the command elevator deflection, say $\gamma_u u_c$. Then

$$\begin{bmatrix} \alpha_{11} + \gamma_u X_{\delta e} & \alpha_{12} & \alpha_{13} + \gamma_{\theta} X_{\delta e} & \gamma_{\dot{h}} X_{\delta T} \\ \alpha_{21} + \gamma_u Z_{\delta e} & \alpha_{22} & \alpha_{23} + \gamma_{\theta} Z_{\delta e} & \gamma_{\dot{h}} Z_{\delta T} \\ \alpha_{31} + \gamma_u M_{\delta e} & \alpha_{32} & \alpha_{33} + \gamma_{\theta} M_{\delta e} & \gamma_{\dot{h}} M_{\delta T} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ \dot{h} \end{bmatrix} = \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{bmatrix} \gamma_u u_c \quad (B-43)$$

The characteristic determinant, Δ_{CL} , and the transfer-function numerator determinants as well, can be expanded in such a way as to retain explicitly the vehicle-alone characteristics, which is a powerful advantage. Also, the resulting expressions can be made amenable to the conventional servo-analysis techniques. There can also be coupling effects between gust inputs and control inputs, and among more than two inputs, control, or disturbance, so that the possible variations are quite numerous. However, the coupling numerators are always easily computed and factored..., generally by being simpler and of lower order than $\Delta(s)$.

Define a notation $N_{\delta_j}^{x_i}$, $N_{\delta_j \delta_l}^{x_i x_h}$ to indicate determinants formed from the characteristic determinant Δ of the unaugmented aircraft in the manner of Cramer's rule (Reference 6). The column of coefficients of x_i is replaced by the column of coefficients of δ_j , and the x_k column is replaced by the δ_l column. For example, the $\theta \rightarrow \delta_e$, $\dot{h} \rightarrow \delta_T$ coupling numerator is

$$N_{\delta_e \delta_T}^{\theta \dot{h}} = \begin{vmatrix} a_{11} & a_{12} & X_{\delta_e} & X_{\delta_T} \\ a_{21} & a_{22} & Z_{\delta_e} & Z_{\delta_T} \\ a_{31} & a_{32} & M_{\delta_e} & M_{\delta_T} \\ a_{41} & a_{42} & 0 & 0 \end{vmatrix} = (a_{11} a_{42} - a_{12} a_{41}) (Z_{\delta_e} M_{\delta_T} - M_{\delta_e} Z_{\delta_T}) + (a_{21} a_{42} - a_{22} a_{41}) (X_{\delta_e} M_{\delta_T} - M_{\delta_e} X_{\delta_T}) + (a_{31} a_{42} - a_{32} a_{41}) (X_{\delta_e} Z_{\delta_T} - Z_{\delta_e} X_{\delta_T}) \quad (B-44)$$

The augmented aircraft denominator then can be expressed

$$\Delta_{CL} = \Delta + Y_u N_{\delta_e}^u + Y_\theta N_{\delta_e}^\theta + Y_h N_{\delta_e}^h + Y_u Y_h N_{\delta_e \delta_T}^{uh} + Y_\theta Y_h N_{\delta_e \delta_T}^{\theta h} \quad (B-45)$$

Note that $N_{\delta_e \delta_e}^{u \theta} \equiv N_{\delta_e \delta_e}^{u \theta h} \equiv 0$ since in every case two identical columns make a determinant zero.

Similarly, the closed-loop transfer-function numerators can be expressed

$$N_{u_c}^u = Y_u \begin{vmatrix} X_{\delta_e} & a_{12} & a_{13} + Y_\theta X_{\delta_e} & Y_h X_{\delta_T} \\ Z_{\delta_e} & a_{22} & a_{23} + Y_\theta Z_{\delta_e} & Y_h Z_{\delta_T} \\ M_{\delta_e} & a_{32} & a_{33} + Y_\theta M_{\delta_e} & Y_h M_{\delta_T} \\ 0 & a_{42} & a_{43} & 1 \end{vmatrix} \quad (B-46)$$

$$= Y_u (N_{\delta_e}^u + Y_h N_{\delta_e \delta_T}^{uh}) \quad (B-47)$$

$$N_{u_c}^\alpha = Y_u \begin{vmatrix} a_{11} + Y_u X_{\delta_e} & X_{\delta_e} & a_{13} + Y_\theta X_{\delta_e} & Y_h X_{\delta_T} \\ a_{21} + Y_u Z_{\delta_e} & Z_{\delta_e} & a_{23} + Y_\theta Z_{\delta_e} & Y_h Z_{\delta_T} \\ a_{31} + Y_u M_{\delta_e} & M_{\delta_e} & a_{33} + Y_\theta M_{\delta_e} & Y_h M_{\delta_T} \\ a_{41} & 0 & a_{43} & 1 \end{vmatrix} \quad (B-48)$$

$$N_{uc}^{\alpha} = Y_u \left(N_{\delta e}^{\alpha} + Y_h N_{\delta e \delta T}^{\alpha h} \right) \quad (B-49)$$

$$N_{uc}^{\theta} = Y_u \begin{vmatrix} a_{11} + Y_u X_{\delta e} & a_{12} & X_{\delta e} & Y_h X_{\delta T} \\ a_{21} + Y_u Z_{\delta e} & a_{22} & Z_{\delta e} & Y_h Z_{\delta T} \\ a_{31} + Y_u M_{\delta e} & a_{32} & M_{\delta e} & Y_h M_{\delta T} \\ a_{41} & a_{42} & 0 & 1 \end{vmatrix} \quad (B-50)$$

$$= Y_u \left(N_{\delta e}^{\theta} + K_h N_{\delta e \delta T}^{\theta h} \right) \quad (B-51)$$

$$N_{uc}^h = Y_u \begin{vmatrix} a_{11} + Y_u X_{\delta e} & a_{12} & a_{13} + Y_{\theta} X_{\delta e} & X_{\delta e} \\ a_{21} + Y_u Z_{\delta e} & a_{22} & a_{23} + Y_{\theta} Z_{\delta e} & Z_{\delta e} \\ a_{31} + Y_u M_{\delta e} & a_{32} & a_{33} + Y_{\theta} M_{\delta e} & M_{\delta e} \\ a_{41} & a_{42} & a_{43} & 0 \end{vmatrix} \quad (B-52)$$

$$= Y_u N_{\delta e}^h \left(\frac{sh(s)}{u_c(s)} = \frac{N_{uc}^h}{\Delta_{CL}} \right) \quad (B-53)$$

In these equations the fourth-degree-of-freedom, h, is a linear combination of the other three. From Equation B-44 it is apparent now that, using functional notation to represent quantities derived from only the three independent equations of motion in u, α , θ ,

$$N_{\delta \delta T}^{\theta h} = a_{42} N_{\delta e \delta T}^{\alpha \theta} (u, \alpha, \theta) + a_{41} N_{\delta e \delta T}^{u \theta} (u, \alpha, \theta) \quad (B-54)$$

Note the rules that follow from the properties of determinants:

$$N_{\delta_j \delta_j}^{X_i X_k} = 0 \quad (B-55)$$

$$N_{\delta_j \delta_l}^{X_i X_k} = - N_{\delta_l \delta_j}^{X_i X_k} = N_{\delta_l \delta_j}^{X_k X_i} \quad (B-56)$$

$$N_{\delta_j \delta_l}^{X_i X_k} = \frac{1}{\Delta} \left(N_{\delta_j}^{X_i} N_{\delta_l}^{X_k} - N_{\delta_l}^{X_i} N_{\delta_j}^{X_k} \right) \quad (B-57)$$

Feedback of bank angle and roll rate to aileron, yaw rate, and (crossfeed of) aileron deflection to rudder results in (if $p = \dot{\phi}$):

$$\begin{bmatrix} a_{11} & a_{12} + (k_p s + k_\phi) Y_{\delta a} & a_{13} + k_r Y_{\delta r} \\ a_{21} & a_{22} + (k_p s + k_\phi) L'_{\delta a} & a_{23} + k_r L'_{\delta r} \\ a_{31} & a_{32} + (k_p s + k_\phi) N'_{\delta a} & a_{33} + k_r N'_{\delta r} \end{bmatrix} \begin{pmatrix} \beta \\ \phi \\ r \end{pmatrix} \quad (B-58)$$

$$= \begin{pmatrix} Y_{\delta a} + k_{\delta a} Y_{\delta r} \\ L'_{\delta a} + k_{\delta a} L'_{\delta r} \\ N'_{\delta a} + k_{\delta a} N'_{\delta r} \end{pmatrix} \delta a_c + \begin{pmatrix} Y_{\delta r} \\ L'_{\delta r} \\ N'_{\delta r} \end{pmatrix} \delta r_c$$

from which lateral-directional closed-loop transfer functions can be expressed in terms of coupling numerators formed solely from feedback/cross-feed gains and the matrix equations of motion of the unaugmented vehicle. The closed-loop denominator is

$$\Delta_{CL} = \Delta + (k_p s + k_\phi) N'_{\delta a} \phi + k_r N'_{\delta r} r + (k_p s + k_\phi) k_r N'_{\delta a} \phi r \quad (B-59)$$

For aileron control inputs

$$N'_{\delta a_c} \beta = N'_{\delta a} \beta + k_{\delta a} N'_{\delta r} \beta + k_r N'_{\delta a} \beta r + k_{\delta a} (k_p s + k_\phi) N'_{\delta r} \beta \phi \quad (B-60)$$

$$N'_{\delta a_c} \phi = N'_{\delta a} \phi + k_{\delta a} N'_{\delta r} \phi + k_r N'_{\delta a} \phi r \quad (B-61)$$

$$N'_{\delta a_c} r = N'_{\delta a} r + k_{\delta a} N'_{\delta r} r + k_{\delta a} (k_p s + k_\phi) N'_{\delta a} \phi r \quad (B-62)$$

while for rudder control inputs

$$N'_{\delta r_c} \beta = N'_{\delta r} \beta + (k_p s + k_\phi) N'_{\delta r} \beta \phi \quad (B-63)$$

$$N'_{\delta r_c} \phi = N'_{\delta r} \phi \quad (B-64)$$

$$N'_{\delta r_c} r = N'_{\delta r} r + (k_p s + k_\phi) N'_{\delta a} \phi r \quad (B-65)$$

AFFDL-TR-78-203

Note that the properties of determinants eliminate a number of the coupling numerators.

Other multiloop control problems may be worked by analogy to these examples. For more detail see Reference 8, which in Section 3-5 goes on to show the use of this concept in multiloop analysis.

APPENDIX C

LATERAL-DIRECTIONAL EQUATIONS

Option 1

The programmed lateral-directional equations of motion are in the stability axis system, i.e., with α_x , the angle of attack of the x output axis, equal to zero. However, it is a simple matter to compensate for nonzero α_x . For body axes we have from pp. 256-258 of Reference 8

$$\left[\left(1 - \gamma_v\right)s - \gamma_v \right] \beta - \left[\left(\gamma_p + \alpha_x\right)s + \frac{q}{U_0} \cos(\Gamma_0 + \alpha_x) \right] \frac{p}{s} + \left[\left(1 - \frac{\gamma_r}{U_0}\right)s - \frac{q}{U_0} \sin(\Gamma_0 + \alpha_x) \right] \frac{r}{s} = \gamma_\delta \delta \quad (C-1)$$

where $\beta = v/U_0$ and $\alpha_x = W_0/U_0$. Note the presence of p/s rather than ϕ and r/s rather than ψ . These differences indicate a minor flaw in the notation. In the output, "bank angle" is really the integral of roll rate. The two terms are identical when θ_0 is zero, differing slightly for small θ_0 .

For the rolling and yawing moment equations the only change needed to accommodate nonzero α_I , α_A , and α_x is to transform the stability derivatives, moments and product of inertia into the output axis system (Appendix A). The program does this for dimensional inertias and non-dimensional stability derivatives, for all three lateral-directional equations. It is seen that in the side force equation, additional factors must be taken into account. The program substitutes

$$(\Gamma_0)_x = \Gamma_0 + \alpha_x \quad (C-2)$$

and

$$(C_{Y_p})_x = (C_{Y_p})_A \cos(\alpha_x - \alpha_A) + (C_{Y_r})_A \sin(\alpha_x - \alpha_A) + \alpha_x \quad (C-3)$$

The rolling and yawing moment equations contain the product of inertia term I_{xz} . To delete this term, define

$$L'_1 = \frac{L_1 + \frac{I_{xz}}{I_{xx}} N_1}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \quad N'_1 = \frac{N_1 + \frac{I_{xz}}{I_{zz}} L_1}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \quad (C-4)$$

As shown for example in Reference 2, recasting the stability derivatives in this form removes the explicit appearance of I_{xz} in the equations. This yields:

$$\dot{p} = L'_v v + L'_\dot{v} \dot{v} + L'_p p + L'_r r + L'_{\delta_A} \delta_A + L'_{\delta_R} \delta_R \quad (C-5)$$

$$\dot{r} = N'_v v + N'_\dot{v} \dot{v} + N'_p p + N'_r r + N'_{\delta_A} \delta_A + N'_{\delta_R} \delta_R \quad (C-6)$$

Taking the Laplace transforms of Equation C-1, C-5, and C-6 and assembling the result into a matrix yields*

$$\begin{bmatrix} s(1-Y'_v) - Y'_v & -\left(\frac{sY'_p}{U_0} + \frac{g}{U_0} \cos \Gamma_0\right) & s\left(1 - \frac{Y'_r}{U_0}\right) - \frac{g}{U_0} \sin \Gamma_0 \\ -sL'_{\dot{\beta}} - L'_{\beta} & s^2 - sL'_p & -sL'_r \\ -sN'_{\dot{\beta}} - N'_{\beta} & -N'_p s & s^2 - sN'_r \end{bmatrix} \begin{bmatrix} \beta(s) \\ \phi(s) \\ \psi(s) \end{bmatrix} = \begin{bmatrix} \frac{Y'_{\delta}}{U_0} \delta(s) \\ L'_{\delta} \delta(s) \\ N'_{\delta} \delta(s) \end{bmatrix} \quad (C-7)$$

or

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \beta(s) \\ \phi(s) \\ \psi(s) \end{bmatrix} = \begin{bmatrix} \frac{Y'_{\delta}}{U_0} \delta(s) \\ L'_{\delta} \delta(s) \\ N'_{\delta} \delta(s) \end{bmatrix}$$

Equation C-1 was divided by U_0 .

Let

$$\hat{Y}_i = Y_i / U_0$$

$$g_s = g \sin \Gamma_0 / U_0$$

and

$$g_c = g \cos \Gamma_0 / U_0$$

* - Strictly (Reference 2) the variable ϕ should be p/s and ψ should be r/s - with $\psi = \dot{\psi} - \dot{\psi} \sin \Gamma_0$ and $r = \dot{\psi} \cos \Gamma_0$ for lateral-directional motion only. The difference should be minor for small flight path inclination.

and solve the characteristic equation of motion

$$\Delta = \begin{vmatrix} s - s\hat{Y}_v - Y_v & -s\hat{Y}_p - gc & s - s\hat{Y}_r - gs \\ -sL'_\beta - L'_\beta & s^2 - sL'_p & -sL'_r \\ -sN'_\beta - N'_\beta & -sN'_p & s^2 - sN'_r \end{vmatrix} = 0 \quad (C-8)$$

$$\begin{aligned} \Delta &= (s - s\hat{Y}_v - Y_v) [(s^2 - sL'_p)(s^2 - sN'_r) - (-sL'_r)(-sN'_p)] \\ &\quad - (-sL'_\beta - L'_\beta) [(-sN'_p)(s - s\hat{Y}_r - gs) + (-s\hat{Y}_p - gc)(s^2 - sN'_r)] \\ &\quad - (+sN'_\beta + N'_\beta) [(-sL'_r)(s - s\hat{Y}_p - gc) - (s^2 - sL'_p)(s - s\hat{Y}_r - gs)] \end{aligned} \quad (C-9)$$

$$\begin{aligned} \Delta &= (s - s\hat{Y}_v - Y_v) [s^4 - s^3 N'_r - s^3 L'_p + s^2 N'_r L'_p - s^2 N'_p L'_r] \\ &\quad - (sL'_\beta + L'_\beta) [-s^2 N'_p + s^2 \hat{Y}_r N'_p + sN'_p gs + s^3 \hat{Y}_p - s^2 \hat{Y}_v N'_r + s^2 gc - sN'_r gc] \\ &\quad + (sN'_\beta + N'_\beta) [-s^2 \hat{Y}_p L'_r - sL'_r gc + s^3 - s^3 \hat{Y}_r - s^2 gs - s^2 L'_p + s^2 \hat{Y}_r L'_p + sL'_p gs] \end{aligned} \quad (C-10)$$

$$\begin{aligned} \Delta &= s^5 - s^4 N'_r - s^4 L'_p + s^3 N'_r L'_p - s^3 N'_p L'_r - s^3 Y_v + s^4 Y_v N'_r + s^4 Y_v L'_p - s^3 Y_v N'_r L'_p \\ &\quad + s^3 Y_v N'_p L'_r - s^4 Y_v + s^3 Y_v N'_r + s^3 Y_v L'_p - s^2 Y_v N'_r L'_p + s^2 Y_v N'_p L'_r \\ &\quad - (s^3 N'_p L'_\beta + s^3 \hat{Y}_r N'_p L'_\beta + s^2 N'_p L'_\beta gs + s^4 \hat{Y}_p L'_\beta - s^3 \hat{Y}_p N'_r L'_\beta + s^3 L'_\beta gc - s^2 N'_r L'_\beta gc \\ &\quad - s^2 N'_p L'_\beta + s^2 \hat{Y}_r N'_p L'_\beta + sN'_p L'_\beta gs + s^3 \hat{Y}_p L'_\beta - s^2 \hat{Y}_p N'_r L'_\beta + s^2 L'_\beta gc - sN'_r L'_\beta gc) \end{aligned} \quad (C-11)$$

$$\begin{aligned} &-s^3 \hat{Y}_p N'_\beta L'_r - s^2 N'_\beta L'_r gc + s^4 N'_\beta - s^4 \hat{Y}_r N'_\beta - s^3 N'_\beta gs - s^3 N'_\beta L'_p + s^3 \hat{Y}_r N'_\beta L'_p + s^2 N'_\beta L'_p gs \\ &-s^2 \hat{Y}_p N'_\beta L'_r - sN'_\beta L'_r gc + s^3 N'_\beta - s^3 \hat{Y}_r N'_\beta - s^2 N'_\beta gs - s^2 N'_\beta L'_p + s^2 \hat{Y}_r N'_\beta L'_p + sN'_\beta L'_p gs \end{aligned}$$

This simplifies to:

$$\Delta = As^5 + Bs^4 + Cs^3 + Ds^2 + Es \quad (C-12)$$

where

$$A = 1 - Y_v \quad (C-13)$$

$$B = -N_r' - L_p' + Y_v N_r' + Y_v L_p' - Y_v + \hat{Y}_p L_\beta' + N_\beta' - \hat{Y}_r N_\beta' \quad (C-14)$$

$$C = N_r' L_p' - N_p' L_r' - Y_v N_r' L_p' + Y_v N_p' L_r' + Y_v N_r' + Y_v L_p' + N_p' L_\beta' \\ + \hat{Y}_r N_p' L_\beta' - \hat{Y}_p N_r' L_\beta' + L_\beta' gc + \hat{Y}_p L_\beta' - \hat{Y}_p N_\beta' L_r' \\ - N_\beta' gs - N_\beta' L_p' + \hat{Y}_r N_\beta' L_p' + N_\beta' - \hat{Y}_r N_\beta' \quad (C-15)$$

$$D = -Y_v N_r' L_p' + Y_v N_p' L_r' + N_p' L_\beta' gs - N_r' L_\beta' gc - N_p' L_\beta' + \hat{Y}_r N_p' L_\beta' \\ - \hat{Y}_p N_r' L_\beta' - L_\beta' gc - N_\beta' L_r' gc - N_\beta' L_p' gs - \hat{Y}_p N_\beta' L_r' \\ - N_\beta' gs - N_\beta' L_p' + \hat{Y}_r N_\beta' L_p' \quad (C-16)$$

$$E = +N_p' L_\beta' gs + N_r' L_\beta' gc - N_\beta' L_r' gc + N_\beta' L_p' gs \quad (C-17)$$

The normal form of the characteristic equation roots is

$$\Delta = As \left(s + \frac{1}{\tau_S} \right) \left(s + \frac{1}{\tau_R} \right) (s^2 + 2 \zeta_{DR} \omega_{DR} s + \omega_{DR}^2) = 0 \quad (C-18)$$

For the Dutch roll mode:

$$s^2 + 2 \zeta_{DR} \omega_{DR} s + \omega_{DR}^2 = (s_1 - \sigma_{DR} + j\omega_{dDR})(s_2 - \sigma_{DR} - j\omega_{dDR}) \quad (C-19)$$

Damping ratio

$$\zeta_{DR} = - \frac{\sigma_{DR}}{\omega_{DR}}$$

Undamped natural frequency

$$\omega_{DR} = \omega_{dDR} / \sqrt{1 - \zeta_{DR}^2}$$

AFFDL-TR-78-203

$$\text{Undamped period } T_{DR} = \frac{2\pi}{\omega_{DR}}$$

$$\text{Damped period } T_{dDR} = \frac{2\pi}{\omega_{dDR}}$$

Cycles to half amplitude

$$C_{1/2DR} = T_{1/2DR} / T_{dDR}$$

Time to half amplitude

$$T_{1/2DR} = -0.69315 / \sigma_{DR}$$

Cycles to 1/10 amplitude

$$C_{1/10DR} = T_{1/10DR} / T_{dDR}$$

Time to 1/10 amplitude

$$T_{1/10DR} = -2.30259 / \sigma_{DR}$$

From the matrix on page 55, these basic transfer functions can be derived:

$$\frac{\beta(s)}{\delta(s)} = \frac{\begin{vmatrix} \hat{Y}_\delta & -s\hat{Y}_p - gc & s - s\hat{Y}_r - gs \\ L'_\delta & s^2 - sL'_p & -sL'_r \\ N'_\delta & -sN'_p & s^2 - sN'_r \end{vmatrix}}{\Delta} \quad (C-20)$$

$$\begin{aligned}
 \text{NUM} = & \hat{Y}_\delta \left[(s^2 - sL'_p)(s^2 - sN'_r) - (-sL'_r)(-sN'_p) \right] \\
 & - L'_\delta \left[(+sN'_p)(s - s\hat{Y}_r - gs) + (s^2 - sN'_r)(-s\hat{Y}_p - gc) \right] \\
 & + N'_\delta \left[(-sL'_r)(-s\hat{Y}_p - gc) - (s^2 - sL'_p)(s - s\hat{Y}_r - gs) \right] \quad (C-21)
 \end{aligned}$$

$$\begin{aligned}
 \text{NUM} = & \hat{Y}_\delta \left[s^4 - s^3 N'_r - s^3 L'_p + s^2 N'_r L'_p - s^2 N'_p L'_r \right] \\
 & - L'_\delta \left[s^2 N'_p - s^2 \hat{Y}_r N'_p - s N'_p gs - s^3 \hat{Y}_p - s^2 gc + s^2 \hat{Y}_p N'_r + s N'_r gc \right] \\
 & + N'_\delta \left[s^2 \hat{Y}_p L'_r + s L'_r gc - s^3 + s^3 \hat{Y}_r + s^2 gs + s^2 L'_p - s^2 \hat{Y}_r L'_p - s L'_p gs \right] \quad (C-22)
 \end{aligned}$$

$$\begin{aligned}
 \text{NUM} = & s^4 \hat{Y}_\delta - s^3 \hat{Y}_\delta N'_r - s^3 \hat{Y}_\delta L'_p + s^2 \hat{Y}_\delta N'_r L'_p - s^2 \hat{Y}_\delta N'_p L'_r - (s^2 N'_p L'_\delta - s^2 \hat{Y}_r N'_p L'_\delta \\
 & - s N'_p L'_\delta gs - s^3 \hat{Y}_p L'_\delta - s^2 L'_\delta gc + s^2 \hat{Y}_p N'_r L'_\delta + s N'_r L'_\delta gc) + s^2 \hat{Y}_p N'_\delta L'_r \\
 & + s N'_\delta L'_r gc - s^3 N'_\delta + s^3 \hat{Y}_r N'_\delta + s^2 N'_\delta gs + s^2 N'_\delta L'_p - s^2 \hat{Y}_r N'_\delta L'_p - s N'_\delta L'_p gs \quad (C-23)
 \end{aligned}$$

This simplifies to:

$$\text{NUM} = A_\beta s^4 + B_\beta s^3 + C_\beta s^2 + D_\beta s \quad (C-24)$$

where

$$A_\beta = \hat{Y}_\delta \quad (C-25)$$

$$B_\beta = \hat{Y}_\delta N'_r - \hat{Y}_\delta L'_p + \hat{Y}_p L'_\delta - N'_\delta + \hat{Y}_r N'_\delta \quad (C-26)$$

$$\begin{aligned}
 C_\beta = & \hat{Y}_\delta N'_r L'_p - \hat{Y}_\delta N'_p L'_r - N'_p L'_\delta + \hat{Y}_r N'_p L'_\delta + L'_\delta gc - \hat{Y}_p N'_r L'_\delta \\
 & + \hat{Y}_p N'_\delta L'_r + N'_\delta gs + N'_\delta L'_p - \hat{Y}_r N'_\delta L'_p \quad (C-27)
 \end{aligned}$$

$$D_\beta = +N'_p L'_\delta gs - N'_r L'_\delta gc + N'_\delta L'_r gc - N'_\delta L'_p gs \quad (C-28)$$

$$\frac{\phi(s)}{\delta(s)} = \frac{\begin{vmatrix} s - sY_v - Y_v & \hat{Y}_\delta & s - s\hat{Y}_r - qs \\ -sL'_\beta - L'_\beta & L'_\delta & -sL'_r \\ -sN'_\beta - N'_\beta & N'_\delta & s^2 - sN'_r \end{vmatrix}}{\Delta} \quad (C-29)$$

$$\begin{aligned} \text{NUM} = & -\hat{Y}_\delta [(-sL'_r X + sN'_\beta + N'_\beta) + (-sL'_\beta - L'_\beta X s^2 - N'_r)] \\ & + L'_\delta [(s - sY_v - Y_v X s^2 - sN'_r) - (s - s\hat{Y}_r - qs)(-sN'_\beta - N'_\beta)] \\ & - N'_\delta [(+sL'_\beta + L'_\beta)(s - s\hat{Y}_r - qs) + (-sL'_r)(s - sY_v - Y_v)] \end{aligned} \quad (C-30)$$

$$\begin{aligned} \text{NUM} = & +\hat{Y}_\delta [s^2 N'_\beta L'_r + s N'_\beta L'_r + s^3 L'_\beta - s N'_r L'_\beta + s^2 L'_\beta - s N'_r L'_\beta] \\ & + L'_\delta [s^3 - s^2 N'_r - s^3 Y_v + s^2 Y_v N'_r - s^2 Y_v + s Y_v N'_r + s^2 N'_\beta + s N'_\beta \\ & - s^2 \hat{Y}_r N'_\beta - s \hat{Y}_r N'_\beta - s N'_\beta qs - N'_\beta qs] \\ & + N'_\delta [-s^2 L'_\beta + s^2 \hat{Y}_r L'_\beta + s L'_\beta qs - s L'_\beta + s \hat{Y}_r L'_\beta + L'_\beta qs + s^2 L'_r - s^2 Y_v L'_r - s Y_v L'_r] \end{aligned} \quad (C-31)$$

$$\begin{aligned} \text{NUM} = & +s^2 \hat{Y}_\delta N'_\beta L'_r + s \hat{Y}_\delta N'_\beta L'_r + s^3 \hat{Y}_\delta L'_\beta - s^2 \hat{Y}_\delta N'_r L'_\beta + s^2 \hat{Y}_\delta L'_\beta - s \hat{Y}_\delta N'_r L'_\beta \\ & + s^3 L'_\delta - s^2 N'_r L'_\delta - s^3 Y_v L'_\delta + s^2 Y_v N'_r L'_\delta - s^2 Y_v L'_\delta + s Y_v N'_r L'_\delta + s^2 N'_\beta L'_\delta \\ & + s N'_\beta L'_\delta - s^2 \hat{Y}_r N'_\beta L'_\delta - s \hat{Y}_r N'_\beta L'_\delta - s N'_\beta L'_\delta qs - N'_\beta L'_\delta qs - s^2 N'_\delta L'_\beta \\ & + s^2 \hat{Y}_r N'_\delta L'_\beta + s N'_\delta L'_\beta qs - s N'_\delta L'_\beta + s \hat{Y}_r N'_\delta L'_\beta + N'_\delta L'_\beta qs + s^2 N'_\delta L'_r \\ & - s^2 Y_v N'_\delta L'_r - s Y_v N'_\delta L'_r \end{aligned} \quad (C-32)$$

This simplifies to:

$$\text{NUM} = A\phi s^3 + B\phi s^2 + C\phi s + D\phi \quad (C-33)$$

where

$$A_{\phi} = -\hat{Y}_{\delta} L'_{\beta} + L'_{\delta} - Y_v L'_{\delta} \quad (C-34)$$

$$B_{\phi} = +\hat{Y}_{\delta} N'_{\beta} L'_r - \hat{Y}_{\delta} N'_r L'_{\beta} + \hat{Y}_{\delta} L'_{\beta} - N'_r L'_{\delta} + Y_v N'_r L'_{\delta} - Y_v L'_{\delta} + N'_{\beta} L'_{\delta} \\ - \hat{Y}_r N'_{\beta} L'_{\delta} - N'_{\delta} L'_{\beta} + \hat{Y}_r N'_{\delta} L'_{\beta} + N'_{\delta} L'_r - Y_v N'_{\delta} L'_r \quad (C-35)$$

$$C_{\phi} = +\hat{Y}_{\delta} N'_{\beta} L'_r - \hat{Y}_{\delta} N'_r L'_{\beta} + Y_v N'_r L'_{\delta} + N'_{\beta} L'_{\delta} - \hat{Y}_r N'_{\beta} L'_{\delta} - N'_{\beta} L'_{\delta} g_s \quad (C-36)$$

$$+ N'_{\delta} L'_{\beta} g_s - N'_{\delta} L'_{\beta} + \hat{Y}_r N'_{\delta} L'_{\beta} - Y_v N'_{\delta} L'_r \quad (C-37)$$

$$D_{\phi} = -N'_{\beta} L'_{\delta} g_s + N'_{\delta} L'_{\beta} g_s$$

$$\frac{r(s)}{s\delta(s)} = \frac{\begin{vmatrix} s - sY_v - Y_v & -\hat{sY}_p - gc & \hat{Y}_{\delta} \\ -sL'_{\beta} - L'_{\beta} & s^2 - sL'_p & L'_{\delta} \\ -sN'_{\beta} - N'_{\beta} & -sN'_p & N'_{\delta} \end{vmatrix}}{\Delta} \quad (C-38)$$

$$\text{NUM} = \hat{Y}_{\delta} [(-sN'_p)(-sL'_{\beta} - L'_{\beta}) - (s^2 - sL'_p)(-sN'_{\beta} - N'_{\beta})] \\ - L'_{\delta} [(+s\hat{Y}_p + gc)(-sN'_{\beta} - N'_{\beta}) + (-sN'_p)(s - sY_v - Y_v)] \\ + N'_{\delta} [(s - sY_v - Y_v)(s^2 - sL'_p) + (-sL'_{\beta} - L'_{\beta})(s + s\hat{Y}_p + gc)] \quad (C-39)$$

$$\text{NUM} = \hat{Y}_{\delta} [s^2 N'_p L'_{\beta} + s N'_p L'_{\beta} + s^3 N'_{\beta} + s^2 N'_{\beta} - s^2 N'_{\beta} L'_p - s N'_{\beta} L'_p] \\ + L'_{\delta} [s^2 \hat{Y}_p N'_{\beta} + s \hat{Y}_p N'_{\beta} + s N'_{\beta} gc + N'_{\beta} gc + s^2 N'_p - s^2 Y_v N'_p - s Y_v N'_p] \\ + N'_{\delta} [s^3 - s^2 L'_p - s^3 Y_v + s^2 Y_v L'_p - s^2 Y_v + s Y_v L'_p - s^2 \hat{Y}_p L'_{\beta} - s L'_{\beta} gc - s \hat{Y}_p L'_{\beta} - L'_{\beta} gc] \quad (C-40)$$

$$\begin{aligned}
\text{NUM} = & s^2 \hat{Y}_\delta N'_p L'_\beta + s \hat{Y}_\delta N'_p L'_\beta + s^3 \hat{Y}_\delta N'_\beta + s^2 \hat{Y}_\delta N'_\beta - s^2 \hat{Y}_\delta N'_\beta L'_p - s \hat{Y}_\delta N'_\beta L'_p \\
& + s^2 \hat{Y}_p N'_\beta L'_\delta + s \hat{Y}_p N'_\beta L'_\delta + s N'_\beta L'_\delta g_c + N'_\beta L'_\delta g_c + s^2 N'_p L'_\delta - s^2 Y_v N'_p L'_\delta - s Y_v N'_p L'_\delta \\
& + s^3 N'_\delta - s^2 N'_\delta L'_p - s^3 Y_v N'_\delta + s^2 Y_v N'_\delta L'_p - s^2 Y_v N'_\delta + s Y_v N'_\delta L'_p - s^2 \hat{Y}_p N'_\delta L'_\beta \\
& - s N'_\delta L'_\beta g_c - s \hat{Y}_p N'_\delta L'_\beta - N'_\delta L'_\beta g_c
\end{aligned} \tag{C-41}$$

This simplifies to

$$\text{NUM} = A_r s^3 + B_r s^2 + C_r s + D_r \tag{C-42}$$

$$A_r = \hat{Y}_\delta N'_\beta + N'_\delta - Y_v N'_\delta \tag{C-43}$$

$$\begin{aligned}
B_r = & \hat{Y}_\delta N'_p L'_\beta + \hat{Y}_\delta N'_\beta - \hat{Y}_\delta N'_\beta L'_p + \hat{Y}_p N'_\beta L'_\delta + N'_p L'_\delta - Y_v N'_p L'_\delta \\
& - N'_\delta L'_p + Y_v N'_\delta L'_p - Y_v N'_\delta - \hat{Y}_p N'_\delta L'_\beta
\end{aligned} \tag{C-44}$$

$$\begin{aligned}
C_r = & \hat{Y}_\delta N'_p L'_\beta - \hat{Y}_\delta N'_\beta L'_p + s \hat{Y}_p N'_\beta L'_\delta + N'_\beta L'_\delta g_c - Y_v N'_p L'_\delta \\
& + Y_v N'_\delta L'_p - N'_\delta L'_\beta g_c - \hat{Y}_p N'_\delta L'_\beta
\end{aligned} \tag{C-45}$$

$$D_r = +N'_\beta L'_\delta g_c - N'_\delta L'_\beta g_c \tag{C-46}$$

ω_ϕ/ω_d

The ω_ϕ/ω_d ratio is the undamped natural frequency of the roll angle per delta aileron numerator divided by that of the Dutch roll. The $\phi(s)/\delta_A(s)$ numerator has the form

$$A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi = 0 \tag{C-47}$$

which has the usual form of solution

$$\left(s + \frac{1}{\tau_\phi} \right) \times s + 2 \zeta_\phi \omega_\phi s + \omega_\phi^2 \tag{C-48}$$

ϕ/β or ϕ/v_e

The parameter ϕ/β is the ratio of roll to sideslip in the Dutch roll mode. As a modal parameter, it is independent of the form or kind of input. The programmed expression may be developed by forming the ratio of transfer-function numerators for a pure yawing moment input:

$$\left. \frac{\phi}{\beta} \right|_{DR} = \frac{\begin{vmatrix} s(1-\gamma_v) - \gamma_v & 0 & s(1 - \frac{\gamma_r}{U_0}) \frac{g}{U_0} \sin \Gamma_0 \\ -sL'_\beta - L'_\beta & 0 & -sL'_r \\ -sN'_\beta - N'_\beta & N & s^2 - sN'_r \end{vmatrix}}{\begin{vmatrix} 0 & -(s \frac{\gamma_p}{U_0} + \frac{g}{U_0} \cos \Gamma_0) & s(1 - \frac{\gamma_r}{U_0}) - \frac{g}{U_0} \sin \Gamma_0 \\ 0 & s^2 - L'_p s & -sL'_r \\ N & -N'_p s & s^2 - sN'_r \end{vmatrix}} \quad \left. s = -\zeta_{DR} \omega_{DR} + j\omega_{DR} \sqrt{1 - \zeta_{DR}^2} \right|_{s = -\zeta_{DR} \omega_{DR} + j\omega_{DR} \sqrt{1 - \zeta_{DR}^2}} \quad (C-49)$$

from which

$$\left. \frac{\phi}{\beta} \right|_{DR} = \frac{[L'_\beta (\frac{\gamma_r}{U_0} - 1) + L'_r (1 - \gamma_v)] s^2 + [(L'_r - 1) L'_\beta + L'_\beta \frac{g}{U_0} \sin \Gamma_0 - L'_r \gamma_v] s + L'_\beta \frac{g}{U_0} \sin \Gamma_0}{(\frac{\gamma_r}{U_0} - 1) s^3 + [L'_r \frac{\gamma_r}{U_0} + \frac{g}{U_0} \sin \Gamma_0 - L'_p (\frac{\gamma_r}{U_0} - 1)] s^2 + (L'_r \frac{g}{U_0} \cos \Gamma_0 - L'_p \frac{g}{U_0} \sin \Gamma_0) s} \quad \left. s = -\zeta_{DR} \omega_{DR} + j\omega_{DR} \sqrt{1 - \zeta_{DR}^2} \right|_{s =} \quad (C-50)$$

To evaluate $\phi(s)/\beta(s)$ for the Dutch roll mode let $s = \sigma_{DR} + j\omega_{dDR}$.

This results in an equation of the form:

$$\left. \frac{\phi(s)}{\beta(s)} \right| = \frac{\sigma_N + j\omega_{dN}}{\sigma_D + j\omega_{dD}} \quad (C-51)$$

Now ϕ/β as defined above is a complex vector (or phasor) in s . For the Dutch roll,

$$s = -\zeta_{DR} \omega_{DR} + \omega_{DR} \sqrt{1 - \zeta_{DR}^2} \quad (C-52)$$

$$s = -\zeta \omega_n + \omega_n i \quad (C-53)$$

$$\omega_{nd} = \omega_n \sqrt{1 - \zeta^2} \quad (C-54)$$

This is substituted into Equation C-50, thus the magnitude of the phasor is

$$\frac{|\phi|}{|\beta|} = \left[\frac{\sigma_N^2 + \omega_N^2}{\sigma_D^2 + \omega_D^2} \right]^{\frac{1}{2}} \quad (C-55)$$

$$\frac{|\phi|}{v_e} = \frac{57.2958}{U_0(\sigma)^{\frac{1}{2}}} \frac{|\phi|}{|\beta|} \frac{\text{deg}}{\text{ft/sec}} \text{ where } \sigma = \frac{\rho}{.0025769} \quad (C-56)$$

Sideslip to Control Deflection:

$$\text{NUM} \frac{\beta(s)}{\delta_a(s)} = \begin{vmatrix} \frac{Y_{\delta_a}}{U_0} & C_{12} & C_{13} \\ L'_{\delta_a} & C_{22} & C_{23} \\ N'_{\delta_a} & C_{32} & C_{33} \end{vmatrix} \quad (C-57)$$

This is of the form:

$$\text{NUM} \frac{\beta(s)}{\delta_a(s)} = s(A_{\beta} s^3 + B_{\beta} s^2 + C_{\beta} s + D_{\beta}) = A_{\beta} s \left(s + \frac{1}{\tau_{\beta_1}} \right) \left(s + \frac{1}{\tau_{\beta_2}} \right) \left(s + \frac{1}{\tau_{\beta_3}} \right) \quad (C-58)$$

When Y_{δ} is zero, the order of this numerator is reduced by one.

Roll Angle to Control Deflection

$$\text{NUM } \frac{\phi(s)}{\delta_a(s)} = \begin{vmatrix} C_{11} & \frac{Y_{\delta_a}}{U_0} & C_{13} \\ C_{21} & L'_{\delta_a} & C_{23} \\ C_{31} & N'_{\delta_a} & C_{33} \end{vmatrix} \quad (\text{C-59})$$

This is of the form

$$\text{NUM } \frac{\phi(s)}{\delta_a(s)} = A_{\phi} s^3 + B_{\phi} s^2 + C_{\phi} s + D_{\phi} = A_{\phi} \left(s + \frac{1}{\tau_{\phi}} \right) (s^2 + 2\zeta_{\phi} \omega_{\phi} s + \omega_{\phi}^2) \quad (\text{C-60})$$

The above equation normally factors into a real root and a complex pair of roots. As already noted, the real root is zero when Γ_0 is zero. The damping ratio, ζ_{ϕ} , and natural frequency, ω_{ϕ} , of the complex pair are calculated in the same manner as the comparable Dutch roll parameters in Equation C-19. Strictly interpreted, this is the numerator of $(1/s) \rho(s)/\delta_a(s)$ rather than $\phi(s)/\delta_a(s)$.

Yaw Rate to Control Deflection

$$\text{NUM } \frac{r(s)}{\delta_a(s)} = s \text{NUM } \frac{\phi(s)}{\delta_a(s)} = s \begin{vmatrix} C_{11} & C_{12} & \frac{Y_{\delta_a}}{U_0} \\ C_{21} & C_{22} & L'_{\delta_a} \\ C_{31} & C_{32} & N'_{\delta_a} \end{vmatrix} \quad (\text{C-61})$$

This is of the form:

$$\text{NUM } \frac{r(s)}{\delta_a(s)} = s(A_r s^3 + B_r s^2 + C_r s + D_r) \quad (\text{C-62})$$

$$= s A_r \left(s + \frac{1}{\tau_R} \right) (s^2 + 2\zeta_R \omega_R s + \omega_R^2) \quad (\text{C-63})$$

Rudder Transfer Function Numerators

The rudder transfer function numerators are of the same form as the aileron transfer function numerators with Y_{δ_r} , L'_{δ_r} , and N'_{δ_r} substituted for Y_{δ_a} , L'_{δ_a} , and N'_{δ_a} , although they may factor differently.

Rudder Transfer Function Numerators, Option 2

The roll rate response to a unit step control input is shown to be of the form

$$p(t) \Big|_{\substack{\text{UNIT} \\ \text{STEP}}} = K_p + K_{pR} e^{-\frac{t}{\tau_R}} + K_{pS} e^{-\frac{t}{\tau_S}} + |K'_{pDR}| e^{-\zeta_{DR} \omega_{DR} t} \cos(\omega_{dDR} t + \psi_p) \quad (C-64)$$

The corresponding sideslip response is

$$\beta(t) \Big|_{\substack{\text{UNIT} \\ \text{STEP}}} = K_\beta + K_{\beta R} e^{-\frac{t}{\tau_R}} + K_{\beta S} e^{-\frac{t}{\tau_S}} + |K'_{\beta DR}| e^{-\zeta_{DR} \omega_{DR} t} \cos(\omega_{dDR} t + \psi_\beta) \quad (C-65)$$

For an aileron control input the parameters of these equations, as well as time histories of roll rate, bank angle and sideslip angle, will be printed:

K_p	KP	K_β	KB
K_{pR}	KPR	$K_{\beta R}$	KBR
K_{pS}	KPS	$K_{\beta S}$	KBS
$ K'_{pDR} $	MKPPDR	$ K'_{\beta DR} $	MKBPDR
ψ_p	PSIP	ψ_β	PSIB

If $1/\tau_S$ is zero, a message to that effect will be printed instead, since in that case the time history equation has a slightly different form.

Some parameters indicative of the amount of Dutch roll response in abrupt aileron rolling maneuvers are given. For a step control input:

$$p_{osc}/p_{av} = \begin{cases} \frac{p_1 + p_3 - 2p_2}{p_1 + p_3 + 2p_2} & , \zeta_{DR} > 0.2 \\ \frac{p_1 - p_2}{p_1 + p_2} & , \zeta_{DR} \leq 0.2 \end{cases} \quad (C-66)$$

where p_1 is the first peak, p_2 is the first minimum following p_1 , and p_3 is the next peak value of roll rate. In the same way ϕ_{osc}/ϕ_{av} is given for an impulse control input; it should be identical to the p_{osc}/p_{av} for a step input. Also given is:

$$K_d/K_{ss} = \left| K'_{p_{DR}} \right| / K_{p_s} \quad KD/KSS \quad (C-67)$$

The parameter $\Delta\beta_{max}$, as defined in MIL-F-8785B, "Flying Qualities of Piloted Airplanes," is a measure of the amount of sideslip in the response to a step roll control input. Over a time interval of half the Dutch roll period or two seconds, whichever is longer, $\Delta\beta_{max}$ is the magnitude of the difference between the largest positive and the largest negative values of sideslip angle:

$$\Delta\beta_{max} \quad \text{DBMAX}$$

For an aileron impulse the phase of the sideslip response in the Dutch roll mode is

$$\psi'_{\beta} \quad \text{PSIBP}$$

The phase angle between the ϕ and β Dutch roll responses is a model parameter, independent of the input:

$$\angle p/\beta \quad \text{ANGLE P/B}$$

This is the angle in the complex representation:

$$\left. \frac{p}{\beta} \right|_{DR} = \left| \frac{p}{\beta} \right|_{DR} e^{j \phi p / \beta} \quad (C-68)$$

Rudder Transfer Function Numerators, Option 3

With this option, the transfer function numerator of side acceleration at a distance l_x from the CG is given. This numerator is of fifth order:

$$\text{NUM} \frac{a'_y(s)}{\delta_a(s)} = A a'_y s^5 + B a'_y s^4 + C a'_y s^3 + D a'_y s^2 + E a'_y s + F a'_y \quad (C-69)$$

Account is taken only of longitudinal displacement from the CG:

$$a'_y = a_{yCG} + l_x \dot{\gamma} \quad (C-70)$$

Both the sensed a'_y (the sum of inertial and gravitational accelerations) and the inertial a_y are given.

Option 2 Equations

$$\frac{p_{osc}}{p_{av}}$$

Using equations C-12 and C-60, $\delta(s) = \frac{|\delta|}{s}$, and $p = s \phi$,

$$\frac{p(s)}{|\delta a|_{STEP}} = \frac{s(A \phi s^3 + B \phi s^2 + C \phi s + D \phi)}{s^2(As^4 + Bs^3 + Cs^2 + Ds + E)} \quad (C-71)$$

$$\frac{p(s)}{|\delta a|_{STEP}} = \frac{K_p}{s} + \frac{K_{pR}}{s + \frac{1}{\tau_R}} + \frac{K_{pS}}{s + \frac{1}{\tau_S}} + \frac{K_{pI}}{s - \sigma_{DR} - j\omega_{dDR}} + \frac{K_{p2}}{s - \sigma_{DR} + j\omega_{dDR}} \quad (C-72)$$

* - Since $p = \dot{\phi} - \dot{\psi} \sin \Gamma_0$, this implies a near-level flight path.

Taking the inverse Laplace transform,

$$p(t) \Big|_{\text{UNIT STEP}} = K_p + K_{p_R} e^{-\frac{t}{\tau_R}} + K_{p_S} e^{-\frac{t}{\tau_S}} + |K_{p_{DR}}| e^{-\zeta_{DR} \omega_{DR} t} \cos(\omega_{dDR} t + \psi_p) \quad (C-73)$$

Where

$$K_p = \frac{A\phi s^3 + B\phi s^2 + C\phi s + D\phi}{As^4 + Bs^3 + Cs^2 + Ds + E} \Big|_{s=0} = \frac{D\phi}{E} \quad (C-74)$$

$$K_{p_R} = \frac{\frac{1}{A} (A\phi s^3 + B\phi s^2 + C\phi s + D\phi)}{(s + \frac{1}{\tau_S})(s^2 + 2\zeta_{DR} \omega_{DR} s + \omega_{DR}^2)} \Big|_{s = -\frac{1}{\tau_R}} \quad (C-75)$$

$$K_{p_S} = \frac{\frac{1}{A} (A\phi s^3 + B\phi s^2 + C\phi s + D\phi)}{s(s + \frac{1}{\tau_R})(s^2 + 2\zeta_{DR} \omega_{DR} s + \omega_{DR}^2)} \Big|_{s = -\frac{1}{\tau_S}} \quad (C-76)$$

$$K_{p_i} = \frac{\frac{1}{A} (A\phi s^3 + B\phi s^2 + C\phi s + D\phi)}{s(s + \frac{1}{\tau_S})(s + \frac{1}{\tau_R})(s - \sigma_{DR} + j\omega_{dDR})} \Big|_{s = \sigma_{DR} + j\omega_{dDR}} \quad (C-77)$$

$$= \frac{\sigma_{p_NUM} + j\omega_{p_NUM}}{\sigma_{p_DENOM} + j\omega_{p_DENOM}} = |K_{p_i}| e^{j\psi_p} \quad (C-78)$$

$$|K_{p_i}| = \left[\frac{\sigma_{p_NUM}^2 + \omega_{p_NUM}^2}{\sigma_{p_DENOM}^2 + \omega_{p_DENOM}^2} \right]^{\frac{1}{2}} \quad (C-79)$$

$$|K'_{pDR}| = 2 |K_{p1}| \quad (C-80)$$

$$\psi_p = \tan^{-1} \left(\frac{\omega_p \text{ NUM}}{\sigma_p \text{ NUM}} \right) - \tan^{-1} \left(\frac{\omega_p \text{ DENOM}}{\sigma_p \text{ DENOM}} \right) \quad (C-81)$$

Now p_{osc}/p_{av} may be found by setting the derivative of Equation C-73 equal to zero and solving for the required peak values.

The values used to compute p_{osc}/p_{av} are also used to compute the peak ratio, p_2/p_1 .

ϕ_{osc}/ϕ_{av} and $\phi(t_x)$

$$\frac{\phi(s)}{|\delta_0|_{\text{IMPULSE}}} = \frac{A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi}{s(A s^4 + B s^3 + C s^2 + D s + E)} \quad (C-82)$$

Comparing Equations C-71 and C-82 it becomes obvious that they are identical. Thus,

$$\left. \frac{\phi_{osc}}{\phi_{av}} \right|_{\text{UNIT IMPULSE}} = \left. \frac{p_{osc}}{p_{av}} \right|_{\text{UNIT STEP}}$$

Equation C-73 may be integrated using initial conditions:

$$\left. \phi(t) \right|_{\text{UNIT STEP}} = 0 \text{ at } t = 0$$

This results in

$$\begin{aligned}
 \phi(t) \Big|_{\text{UNIT STEP}} &= K_p t + K_p \tau_R \left(1 - e^{-\frac{t}{\tau_R}}\right) + K_p \tau_S \left(1 - e^{-\frac{t}{\tau_S}}\right) \\
 &+ \frac{|K'_{DR}|}{(\zeta_{DR} \omega_{DR})^2 + (\omega_{dDR})^2} \left\{ e^{-\zeta_{DR} \omega_{DR} t} \left[-\zeta_{DR} \omega_{DR} \cos(\omega_{dDR} t + \psi_p) \right. \right. \\
 &+ \omega_{dDR} \sin(\omega_{dDR} t + \psi_p) \Big] + \zeta_{DR} \omega_{DR} \cos \psi_p \\
 &\left. - \omega_{dDR} \sin \psi_p \right\}
 \end{aligned} \tag{C-83}$$

Equation C-83 is solved for the input times t_A , t_B and t_C to give the bank angles required in the $\Delta\beta_{\max}$ requirements.

$\Delta\beta_{\max}$ AND ψ_β

Using equations C-12 and C-58:

$$\frac{\beta(s)}{|\delta a|_{\text{STEP}}} = \frac{s(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta)}{s^2(As^4 + Bs^3 + Cs^2 + Ds + E)} \tag{C-84}$$

$$\frac{\beta(s)}{|\delta a|_{\text{STEP}}} = \frac{K_\beta}{s} + \frac{K_{\beta_R}}{s + \frac{1}{\tau_R}} + \frac{K_{\beta_S}}{s + \frac{1}{\tau_S}} + \frac{K_{\beta_1}}{s - \sigma_{DR} - j\omega_{dDR}} + \frac{K_{\beta_2}}{s - \sigma_{DR} + j\omega_{dDR}} \tag{C-85}$$

Taking the inverse Laplace Transform with a unit step input:

$$\beta(t) \Big|_{\text{UNIT STEP}} = K_{\beta} + K_{\beta_R} e^{-\frac{t}{\tau_R}} + K_{\beta_S} e^{-\frac{t}{\tau_S}} + |K_{\beta_{DR}}| e^{-\zeta_{DR} \omega_{DR} t} \cos(\omega_{dDR} t + \psi_{\beta}) \quad (C-86)$$

where

$$K_{\beta} = \frac{A_{\beta} s^3 + B_{\beta} s^2 + C_{\beta} s + D_{\beta}}{A s^4 + B s^3 + C s^2 + D s + E} \Big|_{s=0} = \frac{D_{\beta}}{E} \quad (C-87)$$

$$K_{\beta_R} = \frac{\frac{1}{A} (A_{\beta} s^3 + B_{\beta} s^2 + C_{\beta} s + D_{\beta})}{s (s + \frac{1}{\tau_R}) (s^2 + 2 \zeta_{DR} \omega_{DR} s + \omega_{DR}^2)} \Big|_{s = -\frac{1}{\tau_R}} \quad (C-88)$$

$$K_{\beta_S} = \frac{\frac{1}{A} (A_{\beta} s^3 + B_{\beta} s^2 + C_{\beta} s + D_{\beta})}{s (s + \frac{1}{\tau_R}) (s^2 + 2 \zeta_{DR} \omega_{DR} s + \omega_{DR}^2)} \Big|_{s = -\frac{1}{\tau_S}} \quad (C-89)$$

$$K_{\beta_1} = \frac{\frac{1}{A} (A_{\beta} s^3 + B_{\beta} s^2 + C_{\beta} s + D_{\beta})}{s (s + \frac{1}{\tau_R}) (s + \frac{1}{\tau_S}) (s - \sigma_{DR} + j \omega_{dDR})} \Big|_{s = \sigma_{DR} + j \omega_{dDR}} \quad (C-90)$$

$$= \frac{\sigma_{\beta_{NUM}} + j \omega_{\beta_{NUM}}}{\sigma_{\beta_{DENOM}} + j \omega_{\beta_{DENOM}}} = |K_{\beta_1}| e^{j \psi_{\beta}} \quad (C-91)$$

$$|K_{\beta_1}| = \left[\frac{\sigma_{\beta_{NUM}}^2 + \omega_{\beta_{NUM}}^2}{\sigma_{\beta_{DENOM}}^2 + \omega_{\beta_{DENOM}}^2} \right]^{\frac{1}{2}} \quad (C-92)$$

$$|K'_{\beta_{DR}}| = 2 |K_{\beta_1}| \quad (C-93)$$

$$\psi_{\beta} = \tan^{-1} \left(\frac{\omega_{\beta \text{ NUM}}}{\sigma_{\beta \text{ NUM}}} \right) - \tan^{-1} \left(\frac{\omega_{\beta \text{ DENOM}}}{\sigma_{\beta \text{ DENOM}}} \right) \quad (\text{C-94})$$

Let $t_1 =$ largest of $\frac{T_{d \text{ DR}}}{2}$ or 2 seconds.

Compute:

$$\beta(t_1)$$

$$\beta(t_{\text{MAX}}) \rightarrow \text{MAXIMUM } \beta(t) \text{ for } t \leq t_1$$

$$\beta(t_{\text{MIN}}) \rightarrow \text{MINIMUM } \beta(t) \text{ for } t \leq t_1$$

$$\Delta \beta_{\text{MAX}} = \left| \text{LARGEST POSITIVE } \beta(t) - \text{LARGEST NEGATIVE } \beta(t) \right| \quad (\text{C-95})$$

where the largest positive and largest negative $\beta(t)$ refer to the β 's @ t , t_{max} and t_{min} .

ψ_{β}'

Using Equations C-12 and C-58

$$\frac{\beta(s)}{|\delta a|_{\text{IMPULSE}}} = \frac{s(A_{\beta} s^3 + B_{\beta} s^2 + C_{\beta} s + D_{\beta})}{s(As^4 + Bs^3 + Cs^2 + Ds + E)} \quad (\text{C-96})$$

$$\frac{\beta(s)}{|\delta a|_{\text{IMPULSE}}} = \frac{K'_{\beta R}}{s + \frac{1}{\tau_R}} + \frac{K'_{\beta S}}{s + \frac{1}{\tau_S}} + \frac{K'_{\beta 1}}{s - \sigma_{\text{DR}} - j\omega_{\text{DR}}} + \frac{K'_{\beta 2}}{s - \sigma_{\text{DR}} + j\omega_{\text{DR}}} \quad (\text{C-97})$$

Taking the inverse Laplace transform for a unit impulse input:

$$\beta(t) \Big|_{\text{UNIT IMPULSE}} = K'_{\beta R} e^{-\frac{1}{\tau_R} t} + K'_{\beta S} e^{-\frac{1}{\tau_S} t} + |K'_{\beta \text{DR}}| e^{-\zeta_{\text{DR}} \omega_{\text{DR}} t} \cos(\omega_{\text{DR}} t + \psi'_{\beta}) \quad (\text{C-98})$$

To compare K_{β_1} (in the step response) and K'_{β_1} (in the impulse response) with $\omega = \omega_{dDR}$, the complex coefficients may be written in the manner of Equations C-86 and C-87.

$$K_{\beta_1} = \frac{\sigma_{\beta_N} + j\omega_{\beta_N}}{(\sigma + j\omega)\left(\frac{1}{T_R} + \sigma + j\omega\right)\left(\frac{1}{T_S} + \sigma + j\omega\right)(j2\omega)} \quad (C-99)$$

$$K'_{\beta_1} = (\sigma + j\omega)K_{\beta_1} \quad (C-100)$$

Referring to Equation C-94, it is seen that the phase of the impulse response leads the phase of the step response by the angle $\tan^{-1}(\omega/\sigma)$; or

$$\psi'_{\beta} = \psi_{\beta} + \tan^{-1} \frac{\omega_{dDR}}{-\zeta_{DR} \omega_{DR}} \quad (C-101)$$

The coefficients of C-98 are not calculated, as $\beta(t)|_{\text{unit impulse}}$ is not required.

* ρ/β

Using Equation C-51:

$$\left. \frac{\phi(s)}{\beta(s)} \right|_{DR} = \frac{\sigma_N + j\omega_{dN}}{\sigma_D + j\omega_{dD}} \quad (C-102)$$

$$\left. \frac{\rho(s)}{\beta(s)} \right|_{DR} = \frac{s(\sigma_N + j\omega_{dN})}{\sigma_D + j\omega_{dD}} \Bigg|_{s = \sigma_{DR} + j\omega_{dDR}} \quad (C-103)$$

$$= \frac{(\sigma_{DR} + j\omega_{dDR})(\sigma + j\omega_{dN})}{\sigma_D + j\omega_{dD}} \quad (C-104)$$

$$\left. \frac{p(s)}{\beta(s)} \right|_{DR} = \frac{\sigma'_N + j\omega'_{dN}}{\sigma'_D + j\omega'_{dD}} = |K| e^{j\phi} \frac{p}{\beta} \quad (C-105)$$

$$\phi \frac{p}{\beta} = \tan^{-1} \left(\frac{\omega'_{dN}}{\sigma'_N} \right) - \tan^{-1} \left(\frac{\omega'_{dD}}{\sigma'_D} \right) \quad (C-106)$$

K_D/K_{SS}

$$K_D/K_{SS} = |K'_{pDR}|/K_{pS} \quad (C-107)$$

Option 3 Equations:

Sensed lateral acceleration is the sum of inertial and gravitational accelerations:

$$a'_y = U_0 \dot{\beta} + U_0 r + (g \cos \Gamma_0)(p/s) + (g \sin \Gamma_0)(r/s) + l_x i \quad (C-108)$$

The program solves the augmented determinant

$$N_{\delta}^{a'_y} = \begin{vmatrix} s(1 - Y_v) - Y_v - \frac{sY_p}{U_0} - \frac{g}{U_0} \cos \Gamma_0 & s(1 - \frac{Y_r}{\omega}) - \frac{g}{U_0} \sin \Gamma_0 & X_{\delta} \\ -sL'_\beta & -L'_\beta & s^2 + sL'_p & -sL'_r & Z_{\delta} \\ -sN'_\beta & -N'_\beta & -N'_p s & s^2 - N'_r s & M_{\delta} \\ -U_0 s & g \cos \Gamma_0 & & -(\ell s^2 + U_0 s - g \sin \Gamma_0) & 0 \end{vmatrix} \quad (C-109)$$

for sensed acceleration on the x axis at a distance l_x ahead of the CG to obtain

$$N_{\delta}^{a'_y} = U_0 s N_{\delta}^{\beta} - g \cos \Gamma_0 N_{\delta}^{\phi} + (l_x s^2 + U_0 s - g \sin \Gamma_0) N_{\delta}^{\psi} \quad (C-110)$$

The result is a fifth-order polynomial (for inertial acceleration one of the roots will always be $s = 0$):

$$N_{\delta}^{a'_y} = A_{a'_y} s^5 + B_{a'_y} s^4 + C_{a'_y} s^3 + D_{a'_y} s^2 + E_{a'_y} s + F_{a'_y} \quad (C-111)$$

but even for sensed acceleration the program ignores

$$F_{a'_y} = -g^2 \sin \Gamma_0 \cos \Gamma_0 (L'_{\delta} N'_{\beta} - N'_{\delta} L'_{\beta}) / U_0 \quad (C-112)$$

The fifth (zero) root is not printed.

Coupling Numerators:

Coupling numerators are detailed in Reference 7, Sections 3-5 and the lateral-directional case is explained in Sections 6-11. Coupling numerators for the lateral-directional case follow from an analysis similar to that presented for the longitudinal case in Appendix B. Feedback of bank angle and roll rate to aileron and yaw rate and (crossfeed of) aileron deflection to rudder results (if $p = \dot{\phi}$) in

$$\begin{bmatrix} a_{11} & a_{12} + (K_p s + K_{\phi}) Y_{\delta_a} & a_{13} + K_r Y_{\delta_r} \\ a_{21} & a_{22} + (K_p s + K_{\phi}) L'_{\delta_a} & a_{23} + K_r L'_{\delta_r} \\ a_{31} & a_{32} + (K_p s + K_{\phi}) N'_{\delta_a} & a_{33} + K_r N'_{\delta_r} \end{bmatrix} \begin{pmatrix} \beta \\ \phi \\ r \end{pmatrix} = \begin{pmatrix} Y_{\delta_a} + K_{\delta_a} Y_{\delta_r} \\ L'_{\delta_a} + K_{\delta_a} L'_{\delta_r} \\ N'_{\delta_a} + K_{\delta_a} N'_{\delta_r} \end{pmatrix} \quad (C-113)$$

$$\delta_{ac} + \begin{pmatrix} Y_{\delta_r} \\ L'_{\delta_r} \\ N'_{\delta_r} \end{pmatrix} \delta_{rc}$$

AD-A078 672

AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OH
DIGITAL COMPUTER SOLUTION OF AIRCRAFT LONGITUDINAL AND LATERAL --ETC(U)
JUL 79 J M GRIFFIN , R B YEAGER , L B JORDAN

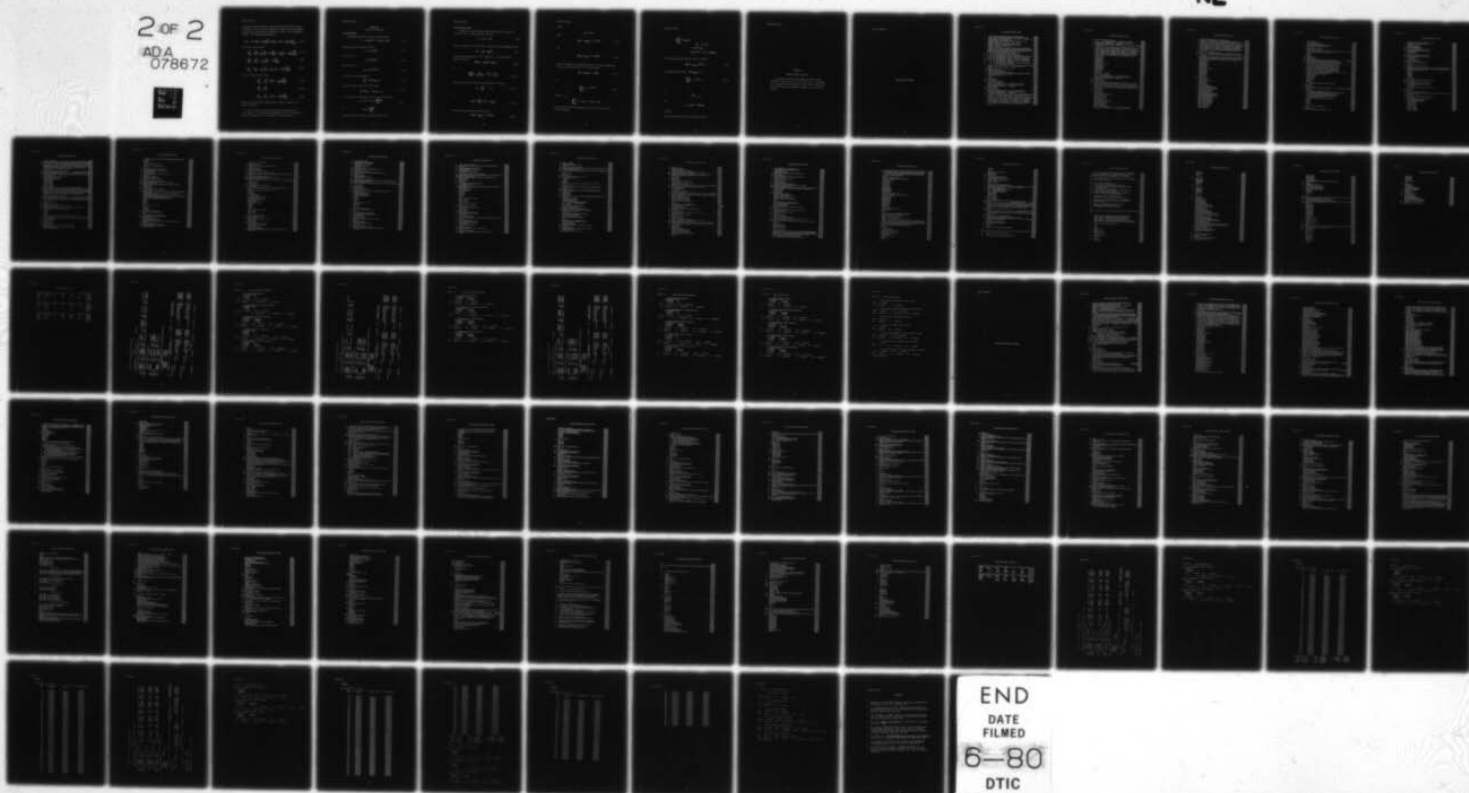
F/6 1/3

UNCLASSIFIED

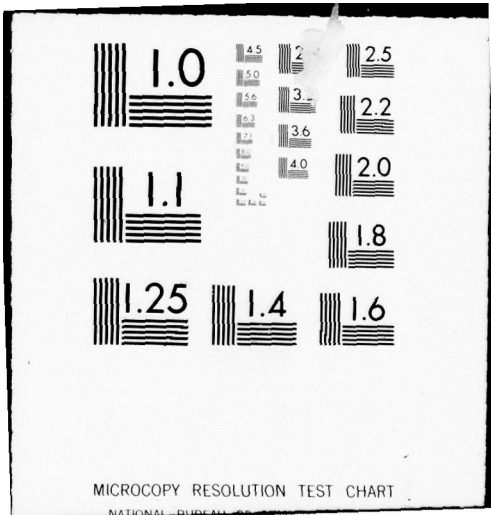
AFFDL-TR-78-203

NL

2 OF 2
ADA
078672



END
DATE
FILMED
6-80
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS

from which lateral-directional closed-loop transfer functions can be expressed in terms of coupling numerators formed solely from feedback/crossfeed gains and the matrix equations of motion of the unaugmented vehicle. The closed-loop denominator is

$$\Delta_{CL} = \Delta + (K_p s + K_\phi) N_{\delta_a}^\phi + K_r N_{\delta_r}^r + (K_p s + K_\phi) K_r N_{\delta_a \delta_r}^{\phi r} \quad (C-114)$$

For aileron control inputs

$$N_{\delta_{a_c}}^\beta = N_{\delta_a}^\beta + K_{\delta_a} N_{\delta_r}^\beta + K_r N_{\delta_a \delta_r}^{\beta r} + K_{\delta_a} (K_p s + K_\phi) N_{\delta_a \delta_r}^{\beta \phi} \quad (C-115)$$

$$N_{\delta_{a_c}}^\phi = N_{\delta_a}^\phi + K_{\delta_a} N_{\delta_r}^\phi + K_r N_{\delta_a \delta_r}^{\phi r} \quad (C-116)$$

$$N_{\delta_{a_c}}^r = N_{\delta_a}^r + K_{\delta_a} N_{\delta_r}^r + K_{\delta_a} (K_p s + K_\phi) N_{\delta_a \delta_r}^{r \phi} \quad (C-117)$$

while for rudder control inputs

$$N_{\delta_{r_c}}^\beta = N_{\delta_r}^\beta + (K_p s + K_\phi) N_{\delta_r \delta_a}^{\beta \phi} \quad (C-118)$$

$$N_{\delta_{r_c}}^\phi = N_{\delta_r}^\phi \quad (C-119)$$

$$N_{\delta_{r_c}}^r = N_{\delta_r}^r + (K_p s + K_\phi) N_{\delta_a \delta_r}^{r \phi} \quad (C-120)$$

Note that the properties of determinants eliminate a number of the coupling numerators.

Other multiloop control problems may be worked by analogy to these examples. For more detail see Reference 7, which in Sections 3-5 goes on to show the use of this concept in multiloop analysis.

APPENDIX D
TIME TO n^{th} AMPLITUDE

Oscillatory Mode

The governing equation for an oscillatory mode is

$$A = A_0 e^{-\zeta \omega_n t} \sin(\omega_n t + \phi) \quad (D-1)$$

The amplitude of this mode of motion

$$A = A_0 e^{-\zeta \omega_n t} \quad (D-2)$$

so, at time $T = 1$,

$$A_1 = A_0 e^{-\zeta \omega_n t_1} \quad (D-3)$$

and, at time $T = 2$,

$$A_2 = A_0 e^{-\zeta \omega_n t_2} \quad (D-4)$$

The ratio of these amplitudes is

$$\frac{A_2}{A_1} = e^{-\zeta \omega_n (t_2 - t_1)} \quad (D-5)$$

Taking the natural logarithm of both sides

$$\ln A_2/A_1 = -\zeta \omega_n (t_2 - t_1) \quad (D-6)$$

For a particular n^{th} amplitude, in this case 1/2 amplitude,

$$t_2 - t_1 = T_n = T_{1/2} = \frac{\ln(0.5)}{-\zeta \omega_n} \quad (D-7)$$

or

$$T_{1/2} = \frac{0.693}{\zeta \omega_n}$$

For time to double amplitude, the same equation holds.

Nonoscillatory Mode

Consider the case of a second order mode with one real root and one root of zero. The equation has the form

$$\ddot{x} + K_1 \dot{x} = f(t) \quad (D-8)$$

which is identical to the roll mode. Inserting the roll parameters yields

$$\ddot{\phi} - L_p \dot{\phi} = L_\delta \delta(t) \quad (D-9)$$

Let the forcing function be a unit impulse at $t = 0$ and taking the Laplace transform

$$s^2 \phi(s) - L_p s \phi(s) = L_\delta \delta(s) \quad (D-10)$$

or

$$\frac{\phi(s)}{\delta \alpha(s)} = \frac{L_\delta}{s(s - L_p)} = \frac{K_1}{s} + \frac{K_2}{s - L_p} \quad (D-11)$$

The method of partial fractions allows solution of K_1 and K_2 :

$$K_1 = \frac{L_\delta}{L_p} = -K_2 \quad (D-12)$$

so

$$\phi(s) = \frac{-L_\delta}{L_p} \left(\frac{1}{s} - \frac{1}{s - L_p} \right) \quad (D-13)$$

Taking the inverse Laplace transform yields

$$\phi(t) = L_\delta \tau_R (1 - e^{-t/\tau_R}) \quad (D-14)$$

where

$$\tau_R = 1/|L_p|$$

Now

$$\phi(t_1) = L_\delta \tau_R (1 - e^{-t_1/\tau_R}) \quad (D-15)$$

and

$$\phi(t_2) = L_\delta \tau_R (1 - e^{-t_2/\tau_R}) \quad (D-16)$$

The first problem is to determine what τ_R is in terms of amplitude. Since $L_\delta \tau_R$ is effectively $\phi(\infty)$ for a unit impulse

$$\phi(t) = \phi(\infty)(1 - e^{-t/\tau_R}) \quad (D-17)$$

so

$$\frac{\phi(t)}{\phi_{\max}} = 1 - e^{-t/\tau_R} \quad (D-18)$$

Letting $t = \tau_R$

$$\frac{\phi(t)}{\phi_{\max}} = 1 - \frac{1}{e} = 1 - 0.37 = 0.63$$

So the value of the time constant yields the time to 63% of the maximum amplitude.

$$\text{if } \frac{\phi(t)}{\phi_{\max}} = 0.5 \text{ then}$$

$$0.5 = 1 - e^{-t/\tau_R}$$

$$e^{-t/\tau_R} = 0.5$$

$$\tau_R \ln 0.5 = t = T_{1/2} = 0.693 \tau_R$$

For the case where the aperiodic mode is unstable

$$\phi(t) = L_{\delta} \tau_R (e^{t/\tau_R} - 1) \quad (\text{D-19})$$

First examine the case of $\phi(t_1)/L_{\delta} \tau_R = 1$

$$\frac{\phi(t_1)}{L_{\delta} \tau_R} = e^{t_1/\tau_R} - 1 = 1 \quad (\text{D-20})$$

so

$$e^{t_1/\tau_R} = 2$$

and

$$t_1 = \tau_R \ln 2 = 0.693 \tau_R$$

as before.

The same equations govern the single pole solution.

APPENDIX E

COMPUTER PROGRAM LISTING

This appendix lists the two programs along with their subroutines, the output, and a list of the input. The program can be keypunched from the listings shown, and the sample data can be used to check the program to ensure it is functioning properly.

LONGITUDINAL PROGRAM

This section lists the two programs along with their
descriptions, the output, and a list of the inputs.
The first program is the longitudinal program and the
second is the program to compute the longitudinal
derivatives.

LONGITUDINAL PROGRAM LISTING

```

PROGRAM LONG(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT)          000100
C   LONGITUDINAL TRANSFER FUNCTION INCLUDING THRUST AND GAMMA  000110
DOUBLE PRECISION ROOTRD, ROOTID, DL, TH, V, H, W, RTR, RTI, AZ  000120
DOUBLE PRECISION AMU, AUH, AMH, AUAZ                          000130
DIMENSION RR(5), RI(5), ROOTR(5), ROOTI(5), DL(5), TITLE(11)  000140
DIMENSION TH(3), V(4), H(4), AZ(4), W(4)                   000150
DIMENSION ROOTRD(5), ROOTID(5), RTR(5), RTI(5)             000160
DIMENSION IND(13,2), AMU(3), AUH(3), AMH(3), AUAZ(3)        000170
COMMON W, RR, RI, XKON, WMLA, ALAWN, LL                     000180
COMMON /A/RTR, RTI                                          000190
COMMON/B/XD, XU, XQ, ZD, ZU, ZQ, AMD, AMU, AMQ, U, GSG, GCG, AM, BM, CM, DM,  000200
1S, RHO, G, GWT, ZT, TDT, XI, CL, CLA, CLAD, CLQ, CLDE, CLM, CD, CDA, CDAD, CDQ,  000210
2CDE, CDM, CMA, CMAD, CMQ, CMDE, CMM, ALPHA, GAMA, CN, CNA, CNAD, CNQ, CNDE,  000220
3CNM, CX, CXA, CXAD, CXQ, CXDE, CXM, XM, ZM, XMD, ZMO, ALA , VE,  000230
4 ZMAC, AM, AIY, ALX, AMH, AMND, ALA1, ANZA, CMO             000240
C   FOR J=0, USE DIMENSIONAL STABILITY DERIVATIVES.          000250
C   J=1, USE NON-DIMENSIONAL STABILITY DERIVATIVES.         000260
C   FOR K=0, USE NON-DIMENSIONAL STABILITY-AXIS STABILITY DERIVATIVES.  000270
C   K=1, USE NON-DIMENSIONAL BODY-AXIS STABILITY DERIVATIVES.  000280
C   FOR M=0, USE NON-DIMENSIONAL STABILITY OR BODY AXIS DERIVATIVES  000290
C   K=1, USE NON-DIMENSIONAL BODY-AXIS STABILITY DERIVATIVES.  000280
C   FOR M=0, USE NON-DIMENSIONAL STABILITY OR BODY AXIS DERIVATIVES  000290
C   WITH UNITS OF 1 PER RADIAN                               000300
C   M=1, USE NON-DIMENSIONAL STABILITY OR BODY AXIS DERIVATIVES  000310
C   WITH UNITS OF 1 PER DEGREE                               000320
DATA(IND(I,1),I=1,12)/12*5H /,IND(1,2)/72H FOR ALPHA AND CONTR 000330
10L DERIVATIVES, AND PER RAD FOR AD AND Q DERIVATIVES/      000340
JJXX=0                                                         000350
100 READ (5,10)J,K,M, RUN, (TITLE(I),I=1,11)                 000360
IF(EOF(5).NE.0) STOP                                         000370
10  FORMAT(I1,I1,I1,A3,11A6)                                  000380
WRITE (6,11) RUN, (TITLE(I),I=1,11)                          000390
11  FORMAT(1H1,10X,45HROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS  000400
1 /1H0,27X8HRUN NO. ,A3/1H0, 7X,11A6)                       000410
IF(J.LT.2)GO TO 320                                          000420
JJXX=1                                                         000430
J=J-5                                                         000440
320 IF(J) 31, 32, 31                                         000450
31  IF(K) 34, 33, 34                                         000460
33  IF(M.GT.4)CALL CHNG(M)                                    000470
IF(M.GT.4)GO TO 1001                                         000480
READ (5,8)S,ZMAC,AM,U,RHO,G, GWT,AIY,ZT,ALX,TDT,XI,         000490
1CL,CLA,CLAD,CLQ,CLDE,CLM, CD,CDA,CDAD,CDQ,CDDE,CDM,        000500
2CMO,CMA,CMAD,CMQ,CMDE,CMM, ALPHA,GAMA                      000510
8  FORMAT(6E12.0)                                             000520
1001 IF(M.GT.4)M=K-5                                         000530
IF(M) 106, 37, 106                                           000540
37  WRITE (6,24)S,ZMAC,AM,U,RHO,G, GWT,AIY,ZT,ALX,TDT,XI,   000550
1CL,CLA,CLAD,CLQ,CLDE,CLM, CD,CDA,CDAD,CDQ,CDDE,CDM,        000560
2CMO,CMA,CMAD,CMQ,CMDE,CMM, ALPHA,GAMA                      000570
24  FORMAT(1H /10X48HINPUT DATA (STABILITY AXIS DERIVATIVES),PER RAD 000580
1 /1H0,4X3HS =1PE12.4,4X5HMAC =E12.4,3X6HMACH =E12.4,5X3HU =E12.4, 000590
2 4X5HRHO =E12.4,5X3HG =E12.4/3X5HGWT =E12.4,4X5HIYY =E12.4,  000600
3 5X4HZT =E12.4,4X4HLX =E12.4,4X5HTDT =E12.4,4X4HXI =E12.4/4X4HCL = 000610
4 E12.4,4X5HCLA =E12.4,3X6HCLAD =E12.4,3X5HCLQ =E12.4,3X6HCLDE = 000620
5 E12.4,3X5HCLM =E12.4/4X4HCD =E12.4,4X5HCDA =E12.4,3X6HCDAD =  000630
6 E12.4,3X5HCDD =E12.4,3X6HCDD =E12.4,3X5HCDM =E12.4/3X5HCMT =  000640
7 E12.4,4X5HCMA =E12.4,3X6HCMA =E12.4,3X5HCMQ =E12.4,3X6HCMDE =  000650
8 E12.4,3X5HCMH =E12.4/1H ,7HALPHA =E12.4,3X6HGAMA =E12.4)  000660
GO TO 101                                                     000670

```

LONGITUDINAL PROGRAM LISTING

```

106 WRITE(6,105) (IND(I,M),I=1,8)                                000680
A      S,ZMAC,AM,U,RHO,G,   GWT,AIY,ZT,ALX,TDT,XI,             000690
1CL,CLA,CLAD,CLQ,CLDE,CLM,  CD,CDA,CDAO,CDQ,CODE,CDM,         000700
2CMO,CMA,CMAO,CMQ,CMDE,CMH,  ALPHA,GAMA                       000710
105  FORMAT(1H0,10X,48HINPUT DATA (STABILITY AXIS DERIVATIVES), PER DEG000720
A      7A10,A2
1 /1H0,4X3MS =1PE12.4,4X5HMAC =E12.4,3X6HMACH =E12.4,5X3HU =E12.4, 000730
2 4X5HRHO =E12.4,5X3HG =E12.4/3X5HGWT =E12.4,4X5HIYY =E12.4,    000750
3 5X4HZT =E12.4,4X4HLX =E12.4,4X5HTDT =E12.4,4X4MXI =E12.4/4X4HCL =000760
4 E12.4,4X5HCLA =E12.4,3X6HCLAD =E12.4,3X5HCLQ =E12.4,3X6HCLDE = 000770
5 E12.4,3X5HCLM =E12.4/4X4HCD =E12.4,4X5HCDA =E12.4,3X6HCDAD = 000780
6 E12.4,3X5HCDO =E12.4,3X6HCDOE =E12.4,3X5HCDM =E12.4/3X5HCMT = 000790
7 E12.4,4X5HCMCA =E12.4,3X6HCMAD =E12.4,3X5HCMQ =E12.4,3X6HCMDE = 000800
8 E12.4,3X5HCMH =E12.4/1H ,7HALPHA =E12.4,3X6HGAMA =E12.4)    000810
DTR=57.295779                                                000820
CLA=CLA*DTR                                                    000830
CLDE=CLDE*DTR                                                  000840
CDA=CDA*DTR                                                    000850
CODE=CODE*DTR                                                  000860
CMA=CMA*DTR                                                    000870
CMDE=CMDE*DTR                                                  000880
IF(M.EQ.2) GO TO 101
CMQ=CMQ*DTR                                                    000890
CMAO=CMAO*DTR                                                  000900
CLAD=CLAD*DTR                                                  000910
CLQ=CLQ*DTR                                                    000920
CDAO=CDAO*DTR                                                  000930
CDQ=CDQ*DTR                                                    000940
GO TO 101
34 IF(M.GT.4) CALL CHNG(M)
IF (M.GT.4) GO TO 1003
READ (5,9)S,ZMAC,AM,U,RHO,G,   GWT,AIY,ZT,ALX,TDT,XI,       000990
1CN,CNA,CNAO,CNQ,CNDE,CNM,CX,CXA,CXAD,CXQ,CXDE,CXM,         001000
2CMO,CMA,CMAO,CMQ,CMDE,CMH,  ALPHA,GAMA                       001010
9  FORMAT(6E12.0)
1003 IF(M.GT.4)M=M-5
IF(M)107,36,107
36 WRITE (6,25)CN,CNA,CNAO,CNQ,CNDE,CNM,CX,CXA,CXAD,CXQ,CXDE,CXM 001050
25  FORMAT(1H0,10X43HINPUT DATA (BODY AXIS DERIVATIVES), PER RAD 001060
1 /1H0,2X4HCN =1PE12.4,4X5HCNA =E12.4,3X6HCNAO =E12.4,3X5HCNQ =E12.4 001070
2,3X6HCNOE =E12.4,3X5HCNM =E12.4/3X4HCX =E12.4,4X5HCXA =E12.4,    001080
3 3X6HCXOE =E12.4,3X5HCXQ =E12.4,3X6HCXDE =E12.4,3X5HCXM =E12.4) 001090
DTP=57.295779
108 ADD=ALPHA/DTR
SA=SIN(ADD)
CA=COS(ADD)
CL=CN*CA-CX*SA
CLA=(CNA-CX)*CA-(CN+CXA)*SA
CLAD=CNAO*CA-CXAD*SA
CLM=CNM*CA-CXM*SA
CLQ=CNQ*CA-CXQ*SA
CLDE=CNDE*CA-CXDE*SA
CD=CX*CA+CN*SA
CDA=(CXA+CN)*CA+(CNA-CX)*SA
CDAO=CXAD*CA+CNAO*SA
CDM=CXM*CA+CNM*SA
CDO=CXQ*CA+CNQ*SA
CDOE=CXDE*CA+CNDE*SA
WRITE (6,77)S,ZMAC,AM,U,RHO,G,   GWT,AIY,ZT,ALX,TDT,XI,    001260
1CL,CLA,CLAD,CLQ,CLDE,CLM,  CD,CDA,CDAO,CDQ,CODE,CDM,         001270

```

LONGITUDINAL PROGRAM LISTING

```

2CMO,CMA,CMAD,CMQ,CMDE,CMH, ALPHA,GAMA                                001280
77 FORMAT(1H0,10X,35HSTABILITY AXIS DERIVATIVES, PER RAD             001290
1 /1H0,4X3HS =1PE12.4,4X5HMAC =E12.4,3X6HMACH =E12.4,5X3HU =E12.4, 001300
2 4X5HRHO =E12.4,5X3HG =E12.4/3X5HGWT =E12.4,4X5HIYY =E12.4,      001310
3 5X4H7T =E12.4,4X4HLX =E12.4,4X5HTDT =E12.4,4X4HXI =E12.4/4X4HCL =001320
4 E12.4,4X5HCLA =E12.4,3X6HCLAD =E12.4,3X5HCLQ =E12.4,3X6HCLDE = 001330
5 E12.4,3X5HCLM =E12.4/4X4HCO =E12.4,4X5HCDA =E12.4,3X6HCDAD = 001340
6 E12.4,3X5HCDO =E12.4,3X6HCDOE =E12.4,3X5HCOM =E12.4/3X5HCOMT = 001350
7 E12.4,4X5HCMA =E12.4,3X6HCMAAD =E12.4,3X5HCMQ =E12.4,3X6HCMADE = 001360
8 E12.4,3X5HCMM =E12.4/1H ,7HALPHA =E12.4,3X6HGAMA =E12.4)         001370
GO TO 101                                                              001380
107 WRITE(6,78 )(IND(I,M),I=1,8)                                     001390
A , CN,CNA,CNAD,CNQ,CNDE,CNM,CX,CXA,CXAD,CXQ,CXDE,CXM              001400
78 FORMAT(1H0,10X,43HINPUT DATA (BODY AXIS DERIVATIVES), PER DEG  001410
A 7A10,A2                                                             001420
1 /1H0,2X4HCN =1PE12.4,4X5HCNA =E12.4,3X6HCNAD =E12.4,3X5HCNQ =E12. 001430
24,3X6HCNDE =E12.4,3X5HCNM =E12.4/3X4HCX =E12.4,4X5HCXA =E12.4, 001440
3 3X6HCXDE =E12.4,3X5HCXQ =E12.4,3X6HCXDE =E12.4,3X5HCXM =E12.4) 001450
DTR = 57.295779                                                       001460
CNA=CNA*DTR                                                            001470
CNDE=CNDE*DTR                                                         001480
CXA=CXA*DTR                                                            001490
CXDE=CXDE*DTR                                                         001500
CMA=CMA*DTR                                                            001510
CMDE=CMDE*DTR                                                         001520
IF (H.EQ.2) GO TO 108                                                001530
CMQ=CMQ*DTR                                                            001540
CMAD=CMAD*DTR                                                         001550
CXAD=CXAD*DTR                                                         001560
CXQ=CXQ*DTR                                                            001570
CNQ=CNQ*DTR                                                            001580
CNAD=CNAD*DTR                                                         001590
GO TO 108                                                             001600
101 DTR=57.295779                                                     001610
ZMASS=GWT/G                                                            001620
XIDD=(XI+ALPHA)/DTR                                                  001630
CIX=COS(XIDD)                                                         001640
SIX=SIN(XIDD)                                                         001650
RSU=RHO*S*U                                                            001660
RSUM=RSU/ZMASS                                                         001670
RSUIC=RSU*ZMAC/AIY                                                    001680
XU=-RSUM*((AM*COM/2.0)+CD)                                           001690
ZU=-RSUM*((AM*CLM/2.0)+CL)                                           001700
AMU=RSUIC*((AM*CMH/2.0)-CMO)                                          001710
XW=RSUM*(CL-CDA)/2.0                                                 001720
ZW=-RSUM*(CLA+CO)/2.0                                               001730
AMW=RSUIC*CMA/2.0                                                    001740
XWD=-RSUM*ZMAC*COAD/(4.0*U)                                          001750
ZWD=-RSUM*ZMAC*CLAD/(4.0*U)                                          001760
AMWD=RSUIC*ZMAC*CMAD/(4.0*U)                                         001770
XQ=-RSUM*ZMAC*COQ/4.0                                               001780
ZQ=-RSUM*ZMAC*CLQ/4.0                                               001790
AMQ=RSUIC*ZMAC*CMQ/4.0                                              001800
XDE=-RSUM*U*COOE/2.0                                                001810
ZDE=-RSUM*U*COOE/2.0                                                001820
AMDE=RSUIC*U*CMOE/2.0                                               001830
XDT=DTT*CIX/ZMASS                                                    001840
ZDT=-DTT*SIX/ZMASS                                                  001850
AMDT=ZT*DTT/AIY                                                      001860
ALA=RSUM*CLA/2.0                                                     001870

```

LONGITUDINAL PROGRAM LISTING

```

ANZA= ALA*U/G                                001880
AKX=SQRT(AIY/ZMASS)                          001890
VE =U*SQRT(RHO*420.716)                      001900
DEPGN=(CMA*CL+G*ZMAC*CMQ*CLA/(2.*U*U))      001910
DEPG=DEPGN/(CLA*CMDE-CMA*CLDE)              001920
GO TO 35                                       001930
32 IF(M.GT.4)CALL CHNG (M)                    001940
IF(M.GT.4)GO TO 1002                          001950
READ (5,12)XU,ZU,AMU,XW,ZW,AMW,XWD,ZWD,AMWD,XQ,ZQ,AMQ, XQ,ZQ,AMQ 001960
10,U,G,GAMA,VE,ALA,ANZA,XDT,ZDT,AMDT
AKX=0.                                         001980
XDE=XD                                         001990
ZDE=ZO                                         002000
AMDE=AMO                                       002010
1002 IF(M.GT.4)M=M-5                          002020
12 FORMAT(6E12.0)                             002030
35 WRITE (6,26)XU,ZU,AMU,XW,ZW,AMW,XWD,ZWD,AMWD, XQ,ZQ,AMQ,002040
1XDE,ZDE,AMDE,XDT,ZDT,AMDT,U,G,GAMA,VE,ALA,ANZA,AKX
26 FORMAT(1H0,10X,33H01MENSIONAL STABILITY DERIVATIVES 002060
1 /1H0,2X,4HXU =E12.4,5X,4HZU =E12.4,5X,4HMU =E12.4 002070
2 /3X,4HXW =E12.4,5X,4HZW =E12.4,5X,4HMW =E12.4 002080
3 /2X,5HXWD =E12.4,4X,5HZWD =E12.4,4X,5HMWD =E12.4 002090
4 /3X,4HXQ =E12.4,5X,4HZQ =E12.4,5X,4HMQ =E12.4 002100
5 /2X,5HXDE =E12.4,4X,5HZDE =E12.4,4X,5HMDE =E12.4 002110
A /2X,5HXDT =E12.4,4X,5HZDT =E12.4,4X,5HMDT =E12.4 002120
6 /4X,3HU =E12.4,6X,3HG =E12.4,3X,6HGAMA =E12.4/ 002130
7 3X4HVE =E12.4,5X4HLA =E12.4,4X5HNZA =E12.4/3X4HKY =E12.4) 002140
IF(J.EQ.1.AND.K.EQ.0)WRITE(6,69)DEPG        002150
69 FORMAT(1H+,21X,6HDE/G =E12.4)            002160
DTR=57.295779                                002170
XKON=2.*3.14159                              002180
GDD=GAMA/DTR                                  002190
SG=SIN(GDD)                                   002200
CG=COS(GDD)                                   002210
GSG=G*SG                                       002220
GCG=G*CG                                       002230
C LONGITUDINAL DENOMINATOR CHARACTERISTICS 002240
DO 128 I=1,4                                  002250
128 W(I)=0.0                                  002260
WRITE (6,16)                                  002270
16 FORMAT(1H0,20X,55HTHE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR 002280
1TOR ARE)
A=1.0-ZWD                                     002290
B=-A*(XU+AMQ)-ZW-AMWD*(U+ZQ)-ZU*XWD         002300
C=XU*(AMQ*A+ZW+AMWD*(U+ZQ))-AMU*(XWD*(U+ZQ)+XQ*A)+AMQ*ZW 002310
1 +ZU*(XWD*AMQ-XW-AMWD*XQ)+AMWD*GSG-AMW*(U+ZQ) 002320
D=GSG*(XWD*AMU+AMW-XU*AMWD)+GCG*(ZU*AMWD+AMU*A) 002330
1 +AMU*(XQ*ZW-XW*(U+ZQ))+ZU*(AMQ*XW-AMW*XQ) 002340
2 +XU*(AMW*(U+ZQ)-AMQ*ZW)                   002350
E=GCG*(ZU*AMW-AMU*ZW)+GSG*(AMU*XW-AMW*XU) 002360
DL(1)=A                                       002370
DL(2)=B                                       002380
DL(3)=C                                       002390
DL(4)=D                                       002400
DL(5)=E                                       002410
N=4                                           002420
CALL DMULR(DL,N,ROOTR,ROOTI)                 002430
M=1                                           002440
66 WRITE (6,401)                              002450
401 FORMAT(1H ,11X20HROOTS (COMPLEX FORM)) 002460

```

LONGITUDINAL PROGRAM LISTING

```

WRITE (6,18) (ROOTRD(I),ROOTID(I),I=1,N)          002480
18  FORMAT(1H , 10XD12.4,5XD12.4)                002490
DO 700 I=1,N                                       002500
ROOTR(I)=-ROOTRD(I)                               002510
700  ROOTI(I)=-ROOTID(I)                          002520
GO TO (65,67,72,73),M                            002530
65  IF(1.E-4-ABS(ROOTI(1)))113,114,114           002540
113  W1=SQRT(ROOTR(1)**2+ROOTI(1)**2)            002550
Z1= ROOTR(1)/W1                                    002560
W3=W1/XKON                                         002570
L=1                                                002580
121  IF(1.E-4-ABS(ROOTI(3)))115,116,116          002590
115  W2=SQRT(ROOTR(3)**2+ROOTI(3)**2)            002600
Z2= ROOTR(3)/W2                                    002610
W4=W2/XKON                                         002620
GO TO (111,122),L                                 002630
111  IF(W1-W2)118,118,117                         002640
117  WRITE (6,14) Z2,W2,Z1,W1,W4,W3              002650
WSP= W1                                            002660
GO TO 81                                           002670
118  WRITE (6,14) Z1,W1,Z2,W2,W3,W4              002680
WSP= W2                                            002690
14  FORMAT(1H0,2X4HZP =E14.6,5X4HWP =E14.6,8H  002700
RAD/SEC,5X5HZSP =E14.6,
15X5HWSP =E14.6,8H RAD/SEC/26X4H =E14.6,11H  002710
CYCLES/SEC,26X5H
2E14.6,11H CYCLES/SEC)                            002720
DUMB=Z2                                            002730
Z2=Z1                                             002740
Z1=DUMB                                           002750
STUPE=W2                                          002760
W2=W1                                             002770
W1=STUPE                                         002780
GO TO 81                                           002790
116  GO TO (20,21),L                               002800
20  CALL FRQCK (Z1,W1,ROOTR(3),ROOTR(4),W3)      002810
GO TO 183                                          002820
114  IF(1.E-4-ABS(ROOTI(2)))119,120,120          002830
119  W1=SQRT(ROOTR(2)**2+ROOTI(2)**2)            002840
Z1= ROOTR(2)/W1                                    002850
W3=W1/XKON                                         002860
CALL FRQCK (Z1,W1,ROOTR(1),ROOTR(4),W3)          002870
GO TO 183                                          002880
120  L=2                                           002890
GO TO 121                                          002900
21  WRITE (6,19) (ROOTR(I),I=1,N)                002910
19  FORMAT(1H0,1X7H1/TD1 =E14.6,5X7H1/TD2 =E14.6, 002920
15X7H1/TD4 =E14.6)
GO TO 83                                           002930
122  CALL FRQCK (Z2,W2,ROOTR(1),ROOTR(2),W4)      002940
GO TO 183                                          002950
81  PER=XKON/(W1*SQRT(1.-ABS(Z1)**2))             002970
TT01=.69315/(ABS(Z1)*W1)                          002980
TT02=2.30259/(ABS(Z1)*W1)                        002990
CT01=TT01/PER                                      003000
CT02=TT02/PER                                      003010
CT03=1.0/CT01                                      003020
CT04=1.0/CT02                                      003030
WNLA =WSP/ALA                                      003040
ALAWN =1./WNLA                                     003050
T7W = 2.*Z1*WSP                                    003060
WNOS = (WSP)**2                                    003070

```


LONGITUDINAL PROGRAM LISTING

```

IF(Z1)1,0,110,402                                003080
402 WRITE (6,124)PER,TT01,TT02,CT01,CT02,CT03,CT04,TZW,MNOS,WNLA,ALAWN003090
124 FORMAT(1H0,1X17HSHORT PERIOD MODE/1H0,11X8HPERIOD =E13.5, 6X19HTIM003100
1E TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =E13.5/1H ,36X2003110
21HCYCLES TO HALF AMP. =E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5003120
3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO003130
4 ONE TENTH AMP. =E13.5/47X11H2*ZSP*WSP =E13.5,33X7HWSPSQ =E13.5 003140
5/1H ,50X7HWN/LA =E13.5,33X7HLA/WN =E13.5) 003150
GO TO 74 003160
110 WRITE (6,149)PER,TT01,TT02,CT01,CT02 003170
149 FORMAT(1H0,1X26HSHORT PERIOD MODE /1H0,11X8HPERIOD =E13.5,003180
1 4X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =E13003190
2.5/1H ,34X23HCYCLES TO DOUBLE AMP. =E13.5,14X26HCYCLES TO TEN TIME003200
3S AMP. =E13.5) 003210
74 PER=XKON/(W2*SQRT(1.-ABS(Z2)**2)) 003220
TT01=.69315/(ABS(Z2)*W2) 003230
TT02=2.30259/(ABS(Z2)*W2) 003240
CT01=TT01/PER 003250
CT02=TT02/PER 003260
CT03=1.0/CT01 003270
CT04=1.0/CT02 003280
PTZW = 2.*Z2*W2 003290
PWNOS = (W2)**2 003300
IF(Z2)76,76,79 003310
79 WRITE (6,138)PER,TT01,TT02,CT01,CT02,CT03,CT04,PTZW,PWNOS 003320
138 FORMAT(1H0,1X17HLONG PERIOD MODE /1H0,11X8HPERIOD =E13.5, 6X19HTIM003330
1E TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =E13.5/1H ,36X2003340
21HCYCLES TO HALF AMP. =E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5003350
3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO003360
4 ONE TENTH AMP. =E13.5/49X9H2*ZP*WP =E13.5,34X6HWPSQ =E13.5) 003370
GO TO 83 003380
76 WRITE (6,139)PER,TT01,TT02,CT01,CT02 003390
139 FORMAT(1H0,1X26HLONG PERIOD MODE /1H0,11X8HPERIOD =E13.5,003400
1 4X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =E13003410
2.5/1H ,34X23HCYCLES TO DOUBLE AMP. =E13.5,14X26HCYCLES TO TEN TIME003420
3S AMP. =E13.5) 003430
GO TO 83 003440
183 IF(LL.NE.1) GO TO 83 003450
WRITE(6,184) WNLA,ALAWN 003460
184 FORMAT(51X7HWN/LA =E13.5,33X7HLA/WN =E13.5) 003470
83 WRITE (6,17)A,B,C,D,E 003480
17 FORMAT(1H0,40X12HCOEFFICIENTS/1H0,2X3HA =E14.6,5X3HB =E14.6, 003490
15X3HC =E14.6,5X3HD =E14.6,5X3HE =E14.6) 003500
C ELEVATOR 003510
XD=XDE 003520
ZD=ZDE 003530
AMD=AMDE 003540
J1=0 003550
IF(XDE.EQ.0..AND.ZDE.EQ.0..AND.AMDE.EQ.0.) GO TO 38 003560
WRITE (6,301) RUN 003570
301 FORMAT(1H1,8HRUN NO. ,A3,10X34HELEVATOR NUMERATOR CHARACTERISTICS)003580
C THETA NUMERATOR 003590
44 DO 131 II=1,5 003600
ROOTR(II)=0.0 003610
131 ROOTI(II)=0.0 003620
WRITE (6,302) 003630
302 FORMAT(1H-,15X*THETA PER CONTROL DEFLECTION*) 003640
AT=ZD*AMWD+AMD*A 003650
BT=XD*(ZU*AMWD+AMU*A)+ZD*(AMU*XWD+AMH-XU*AMWD) 003660
1 -AMD*(XU*A+ZW+ZU*XWD) 003670

```

LONGITUDINAL PROGRAM LISTING

```

CT=XD*(ZU*AMW-AMU*ZW)+ZD*(AMU*XW-XU*AMW)+AMD*(XU*ZW-ZU*XW)      003680
TH(1)=AT                                                                003690
TH(2)=BT                                                                003700
TH(3)=CT                                                                003710
IF(TH(1).EQ.0.)GO TO 42                                              003720
N=2                                                                      003730
CALL DMULR(TH,N,ROOTRD,ROOTID)                                       003740
M=2                                                                      003750
GO TO 66                                                                003760
67 IF(1.E-2-ABS(ROOTI(1)))134,135,135                                003770
134 WT1=SQRT(ROOTR(1)**2+ROOTI(1)**2)                                  003780
Z= ROOTR(1)/WT1                                                         003790
WRITE (6,22)Z,WT1                                                      003800
22  FORMAT(1H0,3X4HZT =E14.6,5X4HWT =E14.6)                          003810
GO TO 90                                                                003820
42  ROOTR(1) = CT/BT                                                    003830
WRITE (6,161) ROOTR(1)                                                 003840
161 FORMAT(1H0,4X6H1/TT =E14.6)                                       003850
GO TO 90                                                                003860
135 WRITE (6,23)ROOTR(1),ROOTR(2)                                       003870
23  FORMAT(1H0,3X7H1/TT1 =E14.6,5X7H1/TT2 =E14.6)                   003880
90  WRITE (6,303)AT,BT,CT                                               003890
303 FORMAT(1H0,3X4HAT =E14.6,5X4HBT =E14.6,5X4HCT =E14.6)           003900
DO 132 II=1,5                                                           003910
ROOTR(II)=0.0                                                           003920
132 ROOTI(II)=0.0                                                       003930
C  HORIZONTAL VELOCITY NUMERATOR                                       003940
WRITE (6,27)                                                            003950
27  FORMAT(1H-,15X*LONGITUDINAL VELOCITY PER CONTROL DEFLECTION*)   003960
AU=XD*A+ZD*XWD                                                         003970
BU=-XD*(AMQ*A+ZW+AMW*(U+ZQ))+ZD*(AMWD*XQ+XW-XWD*AMQ)                003980
1  +AMD*(XWD*(U+ZQ)+XQ*A)                                              003990
CU=XD*(AMQ*ZW-AMW*(U+ZQ)+AMWD*GSG)+ZD*(XQ*AMW-AMWD*GCG-XW*AMQ)     004000
1  +AMD*(XW*(U+ZQ)-XWD*GSG-XQ*ZW-GCG*A)                               004010
DU=XD*AMW*GSG-ZD*AMW*GCG+AMD*(ZW*GCG-XW*GSG)                          004020
V(1)=AU                                                                004030
V(2)=BU                                                                004040
V(3)=CU                                                                004050
V(4)=DU                                                                004060
N = 3                                                                    004070
IF(V(1).NE.0.) GO TO 152                                              004080
N = 2                                                                    004090
V(1)=V(2)                                                                004100
V(2)=V(3)                                                                004110
V(3)=V(4)                                                                004120
IF(V(1).EQ.0.)GO TO 152                                              004130
152 CALL DMULR(V,N,ROOTRD,ROOTID)                                       004140
M=3                                                                      004150
GO TO 66                                                                004160
72 IF(1.E-2-ABS(ROOTI(1)))136,137,137                                004170
136 WV1=SQRT(ROOTR(1)**2+ROOTI(1)**2)                                  004180
Z= ROOTR(1)/WV1                                                         004190
IF(N.EQ.2) GO TO 39                                                    004200
WRITE (6,40)Z,WV1,ROOTR(3)                                             004210
40  FORMAT(1H0,2X4HZU =E14.6,5X4HWU =E14.6,5X6H1/TU =E14.6)         004220
GO TO 84                                                                004230
137 IF(N.EQ.2)GO TO 140                                               004240
IF(1.E-2-ABS(ROOTI(2)))141,41,41                                       004250
141 WV2=SQRT(ROOTR(2)**2+ROOTI(2)**2)                                  004260
Z= ROOTR(2)/WV2                                                         004270

```

LONGITUDINAL PROGRAM LISTING

```

WRITE (6,40)Z,MV2,ROOTR(1) 004280
GO TO 84 004290
39 WRITE (6,143)Z,MV1 004300
143 FORMAT(1H0,2X4HZU =E14.6,5X4HMM =E14.6) 004310
GO TO 84 004320
15 ROOTR(1) = DU/CU 004330
WRITE (6,30)ROOTR(1) 004340
30 FORMAT(1H0,2X6H1/TU =E14.6) 004350
GO TO 84 004360
41 WRITE (6,145)ROOTR(1),ROOTR(2),ROOTR(3) 004370
145 FORMAT(1H0,3X7H1/TU1 =E14.6,5X7H1/TU2 =E14.6,5X7H1/TU3 =E14.6) 004380
GO TO 84 004390
140 WRITE (6,112)ROOTR(1),ROOTR(2) 004400
112 FORMAT(1H0,3X7H1/TU1 =E14.6,5X7H1/TU2 =E14.6) 004410
84 WRITE (6,304)AU,BU,CU,DU 004420
304 FORMAT(1H0,2X4HAU =E14.6,5X4HBU =E14.6,5X4HCU =E14.6,5X4H DU =E14.6)004430
1) 004440
DO 148 II=1,5 004450
ROOTR(II)=0.0 004460
148 ROOTI(II)=0.0 004470
C VERTICAL VELOCITY NUMERATOR 004480
WRITE (6,306) 004490
306 FORMAT(1H-,15X*NORMAL VELOCITY PER CONTROL DEFLECTION*) 004500
N=3 004510
DO 130 II=1,4$IF(W(II).NE.0.0) GO TO 180 004520
130 CONTINUE 004530
AW=ZD 004540
BW=+XD*7U 004550
1 +ZD*(-AMQ-XU) 004560
2 +AMD*(U+ZD) 004570
CW=+XD*((U+ZD)*AMU-AMQ*ZU) 004580
1+ZD*(AMQ*XU-XD*AMU) 004590
2+AMD*(ZU*XQ-GSG-(U+ZD)*XU) 004600
DW=-XD*AMU*GSG 004610
1+ZD*AMU*GCG 004620
2+AMD*(XU*GSG-ZU*GCG) 004630
W(1)=AW 004640
W(2)=BW 004650
W(3)=CW 004660
W(4)=DW 004670
N = 3 004680
IF(W(1).NE.0.0) GO TO 156 004690
W(1) = W(2) 004700
W(2) = W(3) 004710
W(3) = W(4) 004720
N=2 004730
IF(W(1).EQ.0.0)GO TO 123 004740
156 CALL DMULR(W,N,RTR,RTI) 004750
180 WRITE (6,401) 004760
WRITE (6,18) (RTR(I),RTI(I),I=1,N) 004770
DO 600 I=1,N 004780
RR(I)=-RTR(I) 004790
600 RI(I)=-RTI(I) 004800
IF(1.E-2-ABS(RI(1)))54,55,55 004810
54 WW1=SQRT(RR(1)**2+RI(1)**2) 004820
Z= RR(1)/WW1 004830
IF(N.EQ.2)GO TO 163 004840
WRITE (6,56)Z,WW1,RR(3) 004850
56 FORMAT(1H0,2X4HZW =E14.6,7X4HWW =E14.6,7X7H1/TW =E14.6) 004860
GO TO 103 004870

```

LONGITUDINAL PROGRAM LISTING

```

55 IF(N.EQ.2)GO TO 157                                004880
IF(1.E-2-ABS(RI(2)))57,58,58                        004890
57 MW2=SQRT(RR(2)**2+RI(2)**2)                      004900
Z= RR(2)/MW2                                         004910
WRITE(6,56) Z,MW2,RR(1)                             004920
GO TO 103                                            004930
163 WRITE (6,59)Z,WH1                                004940
59 FORMAT(1H0,2X4HZW =E14.6,7X4HWH =E14.6)         004950
GO TO 103                                            004960
157 WRITE (6,129)RR(1),RR(2)                         004970
129 FORMAT(1H0,2X7H1/TW1 =E14.6,7X7H1/TW2 =E14.6)  004980
GO TO 103                                            004990
123 RR(1) = DH/CH                                     005000
WRITE(6,29)RR(1)                                     005010
29 FORMAT(1H0,2X6H1/TW =E14.6)                     005020
GO TO 103                                            005030
58 WRITE (6,60)RR(1),RR(2),RR(3)                   005040
60 FORMAT(1H0,2X7H1/TW1 =E14.6,7X7H1/TW2 =E14.6,7X7H1/TW3 =E14.6)  005050
103 WRITE (6,307)AW,BW,CW,DW                        005060
307 FORMAT(1H0,2X4HAW =E14.6,7X4HBW =E14.6,7X4HCW =E14.6,7X4HDW =E14.6)  005070
1)                                                    005080
DO 133 II=1,5                                       005090
ROOTR(II)=0.0                                       005100
133 ROOTI(II)=0.0                                    005110
C ALTITUDE NUMERATOR                                005120
WRITE (6,305)                                        005130
305 FORMAT(1H-,15X*ALTITUDE RATE PER CONTROL DEFLECTION*) 005140
AH=AU*SG-AW*CG                                       005150
BH=U*AT*CG+BU*SG-BW*CG                              005160
CH=U*BT*CG+CU*SG-CW*CG                              005170
DH=U*CT*CG+DU*SG-DW*CG                              005180
H(1)=AH                                              005190
H(2)=BH                                              005200
H(3)=CH                                              005210
H(4)=DH                                              005220
N = 3                                                005230
IF(H(1).NE.0.0) GO TO 127                           005240
H(1)=H(2)                                           005250
H(2)=H(3)                                           005260
H(3)=H(4)                                           005270
N=2                                                  005280
IF(H(1).EQ.0.)GO TO 75                              005290
127 CALL DMULR (H,N,ROOTR,ROOTI)                    005300
M=4                                                  005310
GO TO 66                                             005320
73 IF(1.E-2-ABS(ROOTI(1)))45,46,46                  005330
45 WH1=SQRT(ROOTR(1)**2+ROOTI(1)**2)                005340
Z= ROOTR(1)/WH1                                      005350
IF(N.EQ.3) GO TO 43                                  005360
WRITE (6,47)Z,WH1                                    005370
47 FORMAT(1H0,2X4H7H =E14.6,7X4HW7 =E14.6)         005380
GO TO 104                                            005390
75 ROOTR(1) = DH/CH                                  005400
WRITE (6,49)ROOTR(1)                                 005410
49 FORMAT(1H0,3X6H1/TH =E14.6)                     005420
GO TO 104                                            005430
46 IF(N.EQ.3)GO TO 50                                005440
WRITE (6,48)ROOTR(1),ROOTR(2)                       005450
48 FORMAT(1H0,2X7H1/TH1 =E14.6,7X7H1/TH2 =E14.6)  005460
GO TO 104                                            005470

```

LONGITUDINAL PROGRAM LISTING

```

43 WRITE (6,51)Z,WH1,ROOTR(3) 005480
51 FORMAT(1H0,2X4HZH =E14.6,7X4MWH =E14.6,7X7H1/TH3 =E14.6) 005490
GO TO 104 005500
50 IF(1.E-2-ABS(ROOTI(2)))52,53,53 005510
52 WH3=SQRT(ROOTR(2)**2+ROOTI(2)**2) 005520
Z= ROOTR(2)/WH3 005530
WRITE (6,51) Z,WH3,ROOTR(1) 005540
GO TO 104 005550
53 WRITE (6,155)ROOTR(1),ROOTR(2),ROOTR(3) 005560
155 FORMAT(1H0,2X7H1/TH1 =E14.6,7X7H1/TH2 =E14.6,7X7H1/TH3 =E14.6) 005570
104 WRITE (6,13)AH,BH,CH,DH 005580
13 FORMAT(1H0,2X4MAH =E14.6,5X4MBH =E14.6,5X4MCH =E14.6, 005590
15X4MDH =E14.6) 005600
DO 146 II=1,5 005610
RR(II)=0.0 005620
146 RI(II)=0.0 005630
C VERTICAL ACCELERATION NUMERATOR 005640
IF(ALX.EQ.0.0) GO TO 109 005650
WRITE (6,308) 005660
308 FORMAT(1H0,15X4DHVERTICAL ACCELERATION PER DELTA ELEVATOR) 005670
AA=-ALX*AT+AM 005680
AB=-ALX*BT+BM-U*AT 005690
AC=-ALX*CT+CM-U*BT 005700
AD=-U*CT+DW 005710
AZ(1)=AA 005720
AZ(2)=AB 005730
AZ(3)=AC 005740
AZ(4)=AD 005750
N=3 005760
IF(AZ(1).NE.0.0) GO TO 159 005770
AZ(1) = AZ(2) 005780
AZ(2) = AZ(3) 005790
AZ(3) = AZ(4) 005800
IF(AZ(1).EQ.0.0) GO TO 160 005810
N=2 005820
159 CALL DMULR (AZ,N,RTR,RTI) 005830
WRITE (6,401) 005840
WRITE (6,18) (RTR(I),RTI(I),I=1,N) 005850
DO 900 I=1,N 005860
RR(I)=-RTR(I) 005870
900 RI(I)=-RTI(I) 005880
IF(1.E-2-ABS(RI(1)))61,62,62 005890
61 WA1=SQRT(RR(1)**2+RI(1)**2) 005900
Z= RR(1)/WA1 005910
IF(N.EQ.2)GO TO 164 005920
WRITE (6,63)Z,WA1,RR(3) 005930
63 FORMAT(1H0,2X5HZAZ =E14.6,7X5HMAZ =E14.6,7X8H1/TAZ1 =E14.6) 005940
GO TO 86 005950
62 IF(N.EQ.2)GO TO 166 005960
IF(1.E-2-ABS(RI(2)))64,68,68 005970
64 WA2=SQRT(RR(2)**2+RI(2)**2) 005980
Z= RR(2)/WA2 005990
WRITE (6,63)Z,WA2,RR(1) 006000
GO TO 86 006010
164 WRITE(6,165)Z,WA1 006020
165 FORMAT(1H0,2X5HZAZ =E14.6,7X5HMAZ =E14.6) 006030
GO TO 86 006040
166 WRITE(6,167)RR(1),RR(2) 006050
167 FORMAT(1H0,2X8H1/TAZ1 =E14.6,7X8H1/TAZ2 =E14.6) 006060
GO TO 86 006070

```

LONGITUDINAL PROGRAM LISTING

```

160 RR(1) = AD/AC                                006080
    WRITE(6,168)RR(1)                            006090
168 FORMAT(1H0,2X7H1/TAZ =E14.6)                006100
    GO TO 86                                      006110
68  WRITE (6,71)RR(1),RR(2),RR(3)                006120
71  FORMAT(1H0,2X8H1/TAZ1 =E14.6,7X8H1/TAZ2 =E14.6,7X8H1/TAZ3 =E14.6) 006130
86  WRITE (6,144)AA,AB,AC,AD                     006140
144 FORMAT(1H0,2X4HAA =E14.6,7X4HBA =E14.6,7X4HCA =E14.6,7X4HDA =E14.6)006150
    1)
    DO 147 II=1,5                                006160
    RR(II)=0.0                                    006170
147 RI(II) = 0.0                                  006180
109 IF(J1.EQ.1.AND.JJXX.EQ.1)GO TO 321          006190
    IF(J1.EQ.1)GO TO 100                          006200
C      THRUST                                     006220
    XD=XDT                                         006230
    ZD=ZDT                                         006240
    AMD=AMDT                                        006250
38  IF(XDT.EQ.0..AND.ZDT.EQ.0..AND.AMDT.EQ.0.) GO TO 100 006260
    J1=1                                           006270
    WRITE(6,28)RUN                                 006280
28  FORMAT(1H1,2X,8HRUN NO. A3,5X22HTHRUST NUMERATOR ROOTS) 006290
    GO TO 44                                       006300
321 WRITE(6,322)RUN                               006310
322 FORMAT(1H1,2X,8HRUN NO. A3,5X*COUPLING NUMERATOR ROOTS*) 006320
    DO 323 I1=1,5                                  006330
    ROOTR(I1)=0.                                  006340
323 ROOTI(I1)=0.                                  006350
    WRITE(6,324)                                    006360
324 FORMAT(1H-,14X*THETA TO ELEVATOR, LONGITUDINAL VELOCITY TO*,
1* THRUST*)
    ZMZM=ZDT*AMDE-ZDE*AMDT                       006380
    XMXM=XDT*AMDE-XDE*AMDT                       006390
    XZXZ=XDT*ZDE-XDE*ZDT                         006400
    ATU = XMXM+XWD*ZMZM-ZWD*XMXM+AMWD*XZXZ      006410
    BTU = XW*ZMZM- ZN*XMXM+AMW*XZXZ              006430
    TTU =BTU/ATU                                  006440
    IF(ATU.EQ.0..OR.BTU.EQ.0.)TTU=0.              006450
    WRITE(6,325)TTU,ATU,BTU                       006460
325 FORMAT(1H0,3X*1/TTU =*E14.6//
14X5HATU =E14.6,5X5HBTU =E14.6///15X,
2*NORMAL VELOCITY TO ELEVATOR, LONGITUDINAL*
3* VELOCITY TO THRUST*)
    AWU(1)=XZXZ                                    006510
    AWU(2)=U*XMXM-XQ*ZMZM+ZQ*XMXM-AMQ*XZXZ      006520
    AWU(3)=GCG*ZMZM-GSG*XMXM                     006530
    IF(AWU(1).EQ.0.)GO TO 326                     006540
    CALL DMULR(AWU,2,ROOTRD,ROOTID)               006550
    MM=1                                            006560
    GO TO 1                                         006570
3  IF(ABS(ROOTI(1)).LT..0001)GO TO 327            006580
    WWU=SQRT(ROOTI(1)**2+ROOTR(1)**2)             006590
    ZWU=-ROOTR(1)/WWU                              006600
    WRITE(6,328)ZWU,WWU                            006610
328 FORMAT(1H0,3X,*ZWU =*E14.6,5X*WWU =*E14.6) 006620
    GO TO 329                                       006630
326 TWU=AWU(3)/AWU(2)                              006640
    IF(AWU(2).EQ.0.DD.OR.AWU(3).EQ.0.DD) TWU=0.  006650
    WRITE(6,330)TWU                                006660
330 FORMAT(1H0,3X*1/TWU =*E14.6)                006670

```

LONGITUDINAL PROGRAM LISTING

```

GO TO 329                                006680
327 ROOTR(1)=-ROOTR(1)                    006690
    ROOTR(2)=-ROOTR(2)                    006700
    WRITE(6,331)ROOTR(1),ROOTR(2)         006710
331 FORMAT(1H0,3X*1/TWU1=*E14.6,5X*1/TWU2=*E14.6) 006720
329 WRITE(6,332)AWU(1),AKU(2),AWU(3)     006730
332 FORMAT(1H0,3X*AWU=*D14.6,5X*BWU=*D14.6,5X*CWU=*D14.6/// 006740
    115X*THETA TO ELEVATOR, NORMAL VELOCITY TO THRUST*)
    DO 333 I1=1,5                          006750
    ROOTI(I1)=0.                            006760
333 ROOTR(I1)=0.                            006770
    ATW=ZMZW                                006780
    BTW=-XU*ZMZW+ZU*XXM-XAMU*XZXZ        006790
    TTH1=BTW/ATW                            006800
    IF(ATW.EQ.0..OR.BTW.EQ.0.)TTH1=0.     006810
    WRITE(6,334)TTH1,ATW,BTW              006820
334 FORMAT(1H0,3X*1/TTH=*E14.6//4X*ATW=*E14.6,5X*BTW=*E14.6///15X, 006830
    1*S TIMES THETA TO ELEVATOR, ALTITUDE TO THRUST*)
    ATH=+SG*ATU-CG*ATW                     006840
    BTH= SG*BTU-CG*BTW                     006850
    TTH =BTH/ATH                           006860
    IF(ATH.EQ.0..OR.BTH.EQ.0.) TTH =0.    006870
    WRITE(6,335)TTH,ATH,BTH              006880
335 FORMAT(1H0,3X*1/TTH=*E14.6//4X*ATH=*E14.6,5X*BTH=*E14.6///15X, 006890
    1*S TIMES LONGITUDINAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR*)
    AUH(1)=-AWU(1)*CG                       006900
    AUH(2)=-AWU(2)*CG+U*CG*ATU            006910
    AUH(3)=-AWU(3)*CG+U*CG*BTU           006920
    IF(AUH(1).EQ.0.)GO TO 336             006930
    CALL DMULR(AUH,2,ROOTR,ROOTI)         006940
    MM=2                                    006950
    GO TO 1                                 006960
4 IF(ABS(ROOTI(1)).LT..0001)GO TO 337    006970
    WUH=SQRT(ROOTI(1)**2+ROOTR(1)**2)     006980
    ZUH=-ROOTR(1)/WUH                     006990
    WRITE(6,338)ZUH,WUH                    007000
338 FORMAT(1H0,3X*ZUH=*E14.6,5X*WUH=*E14.6) 007010
    GO TO 339                               007020
336 TUH=AUH(3)/AUH(2)                     007030
    IF(AUH(2).EQ.0..OR.AUH(3).EQ.0..OR) TUH=0. 007040
    WRITE(6,370)TUH                         007050
370 FORMAT(1H0,3X*1/TUH=*E14.6)          007060
    GO TO 339                               007070
337 ROOTR(1)=-ROOTR(1)                    007080
    ROOTR(2)=-ROOTR(2)                    007090
    WRITE(6,340)ROOTR(1),ROOTR(2)         007100
340 FORMAT(1H0,3X*1/TUH1=*E14.6,5X*1/TUH2=*E14.6) 007110
339 WRITE(6,341)AUH(1),AUH(2),AUH(3)     007120
341 FORMAT(1H0,3X*AUH=*D14.6,5X*BUH=*D14.6,5X*CUH=*D14.6/// 007130
    115X*S TIMES NORMAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR*)
    DO 342 I1=1,5                          007140
    ROOTI(I1)=0.                            007150
342 ROOTI(I1)=0.                            007160
    AWH(1)=-AWU(1)*SG                       007170
    AWH(2)=-AWU(2)*SG+U*ATW*CG           007180
    AWH(3)=-AWU(3)*SG+U*BTW*CG           007190
    IF(AWH(1).EQ.0.)GO TO 343             007200
    CALL DMULR(AWH,2,ROOTR,ROOTI)         007210
    MM=3                                    007220
    GO TO 1                                 007230

```

LONGITUDINAL PROGRAM LISTING

```

5  IF(ABS(ROOTI(1)).LT..0001)GO TO 344          007280
    MMH =SQRT(ROOTI(1)**2+ROOTR(1)**2)          007290
    ZWH=-ROOTR(1)/MMH                          007300
    WRITE(6,345)ZWH,MMH                        007310
345  FORMAT(1H0,3X*ZWH =*E14.6,5X*MMH =*E14.6) 007320
    GO TO 346                                  007330
343  TWH =AMH(3)/AMH(2)                        007340
    IF(AMH(2).EQ.0.D0.OR.AMH(3).EQ.0.D0) TWH=0. 007350
    WRITE(6,347)TWH                            007360
347  FORMAT(1H0,3X*1/TWH =*E14.6)            007370
    GO TO 346                                  007380
344  ROOTR(1)=-ROOTR(1)                        007390
    ROOTR(2)=-ROOTR(2)                        007400
    WRITE(6,349)ROOTR(1),ROOTR(2)             007410
349  FORMAT(1H0,3X*1/TWH1 =*E14.6,5X*1/TWH2 =*E14.6) 007420
346  WRITE(6,348)(AMH(I),I=1,3)               007430
348  FORMAT(1H0,3X*AMH =*D14.6,5X*BWH =*D14.6,5X*CMH =*D14.6///15X, 007440
    1*S TIMES LONGITUDINAL VELOCITY TO THRUST, ACCELERATION *
    2*TO ELEVATOR*)                          007450
    DO 350 I=1,5                               007460
    ROOTR(I)=0.                                007470
350  ROOTI(I)=0.                              007480
    AUAZ(1)= AMU(1)-ATU*ALX                   007490
    AUAZ(2)= AMU(2)-BTU*ALX+U*ATU             007500
    AUAZ(3)= AMU(3)-U*BTU                    007510
    IF(AUAZ(1).EQ.0.)GO TO 351                 007520
    CALL DNULR(AUAZ,2,ROOTR,ROOTI)            007530
    MM=4                                       007540
    GO TO 1                                    007550
6  IF(ABS(ROOTI(1)).LT..0001)GO TO 352          007560
    WUAZ =SQRT(ROOTI(1)**2+ROOTR(1)**2)        007570
    ZUAZ =-ROOTI(1)/WUAZ                      007580
    WRITE(6,353)ZUAZ,WUAZ                    007590
353  FORMAT(1H0,3X*ZUAY =*E14.6,5X*WUAZ =*E14.6) 007600
    GO TO 354                                  007610
351  TUAZ =AUAZ(3)/AUAZ(2)                    007620
    IF(AUAZ(2).EQ.0.D0.OR.AUAZ(3).EQ.0.D0) TUAZ=0. 007630
    WRITE(6,355)TUAZ                          007640
355  FORMAT(1H0,3X*1/TUAZ =*E14.6)           007650
    GO TO 354                                  007660
352  ROOTR(1)=-ROOTR(1)                        007670
    ROOTR(2)=-ROOTR(2)                        007680
    WRITE(6,356)ROOTR(1),ROOTR(2)            007690
356  FORMAT(1H0,3X*1/TUAZ1 =*E14.6,5X*1/TUAZ2 =*E14.6) 007700
354  WRITE(6,357)(AUAZ(I),I=1,3)             007710
357  FORMAT(1H0,3X*AUAZ =*D14.6,5X*BUAZ =*D14.6,5X*CUAZ =*D14.6) 007720
    JJXX=0                                     007730
    GO TO 100                                  007740
1  DO 2 I=1,3                                  007750
    ROOTR(I)=ROOTPD(I)                        007760
    ROOTI(I)=ROOTID(I)                       007770
2  GO TO (3,4,5,6),MM                        007780
    END                                        007790
    SUBROUTINE CHNG(J)                         007800
    COMMON/B/XD,XU,XQ,ZD,ZU,ZQ,MD,MU,MQ,U,GS,GCG,AM,BW,CW,DW, 007810
    1S,RHO,G,GMT,ZT,TT,XI,CL,CLA,CLAD,CLG,CLOE,CLM,CD,CDA,CDAD,CDQ,
    2CODE,COM,CMA,CMAD,CMQ,CNDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE,
    3CNM,CX,CXA,CXAD,CXQ,CXOE,CXM,XW,ZN,XND,ZHO,LA,VE,
    4 MAC,MACH,IYY,LX,MW,MWD,ALA,NZA,CMT     007850
    REAL MD,MU,MQ,MAC,MACH,IYY,LX,MW,MWD,LA ,NZA 007870

```


LONGITUDINAL PROGRAM LISTING

```

NAMELIST/CHANGE/S,MAC, U,RHO,G,GMT,IYY,ZT,LX,TDT,XI,CLA,CLAD, 007880
A CLQ,CL,CLDE,CLM,CD,CDA,CDAD,CDQ,CDDE,CDM,CMT,CMA,CMAD,CMQ, 007890
B CMDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE,CNM,CX,CXA,CXAD, 007900
C CXQ,CXDE,CXM,XU,ZU,MU,XM,ZH,MH,XWD,ZWD,MWD,XQ,ZQ,MQ,XD,ZD, 007910
D MD,VE,LA,NZA,TEST,MACH,CMCL 007920
CMCL=99. 007930
IF(J.EQ.5) READ(5,CHANGE) 007940
IF(J.EQ.5.AND.CMCL.NE.99.)CMA=CLA*CMCL 007950
IF(J.EQ.5) RETURN 007960
DTR=57.295779 007970
CLA=CLA/DTR 007980
CDA=CDA/DTR 007990
CMA=CMA/DTR 008000
CXA=CXA/DTR 008010
CZA=CZA/DTR 008020
CLDE=CLDE/DTR 008030
CDDE=CDDE/DTR 008040
CMDE=CMDE/DTR 008050
CXDE=CXDE/DTR 008060
CZDE=CZDE/DTR 008070
IF(J.EQ.7) READ(5,CHANGE) 008080
IF(J.EQ.7.AND.CMCL.NE.99.)CMA=CLA*CMCL 008090
IF(J.EQ.7) RETURN 008100
CLAD=CLAD/DTR 008110
CDAD=CDAD/DTR 008120
CMAD=CMAD/DTR 008130
CXAD=CXAD/DTR 008140
CZAD=CZAD/DTR 008150
CLQ=CLQ/DTR 008160
CDQ=CDQ/DTR 008170
CMQ=CMQ/DTR 008180
CXQ=CXQ/DTR 008190
CZQ=CZQ/DTR 008200
IF(CMCL.NE.99.)CMA=CLA*CMCL 008210
READ(5,CHANGE) 008220
RETURN 008230
END 008240
SUBROUTINE FROCK (ZN,WN,ROOTR1,ROOTR2,WNC) 008250
THIS SUBROUTINE USES SUBROUTINE DMULR 008260
DOUBLE PRECISION RTR,RTI,W 008270
DIMENSION W(4),RR(5),RI(5) 008280
COMMON W,RR,RI,XKON,WNLA,ALAWN, LL 008290
COMMON /A/RTR(5),RTI(5) 008300
COMMON/B/XD,XU,XQ,ZD,ZU,ZQ,AMD,AMU,AMQ,U,GSG,GCG,AW,BW,CW,DW, 008310
1S,RHO,G,GMT,ZT,TDT,XI,CL,CLA,CLAD,CLQ,CLDE,CLM,CD,CDA,CDAD,CDQ, 008320
2CDDE,CDM,CMA,CMAD,CMQ,CMDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE, 008330
3CNM,CX,CXA,CXAD,CXQ,CXDE,CXM,XW,ZH,XWD,ZWD,ALA ,VE, 008340
4 ZMAC,AM,AIY,ALX,AMW,AMWD,ALA1,ANZA,CMO 008350
AW=+ZD 008360
BW=+XD*ZU 008370
1 +ZD*(-AMQ-XU) 008380
2 +AMD*(U+ZQ) 008390
CW=+XD*((U+ZQ)*AMU-AMQ*ZU) 008400
1*ZD*(AMQ*XU-XQ*AMU) 008410
2*AMD*(ZU*XQ-GSG-(U+ZQ)*XU) 008420
DW=-XD*AMU*GSG 008430
1*ZD*AMU*GCG 008440
2*AMD*(XU*GSG-ZU*GCG) 008450
W(1)=AW 008460
W(2)=BW 008470

```

LONGITUDINAL PROGRAM LISTING

```

      W(3)=CW                                008480
      W(4)=DW                                008490
      N=3                                    008500
      CALL DMULR (W,N,RTR,RTI)              008510
      DO 800 I=1,N                           008520
      RR(I)=RTR(I)                           008530
800  RI(I)=RTI(I)                            008540
      IF(1.E-4-ABS(RI(1)))54,55,55          008550
54  WW1=SQRT(RR(1)**2+RI(1)**2)            008560
26  IF(WW1+.4*WW1.LT.WN) GO TO 23          008570
      IF(WW1-.4*WW1.LT.WN) GO TO 20        008580
23  WNLA = WN/ALA                           008590
      ALAWN = 1./WNLA                       008600
      LL = 1                                 008610
      WRITE(6,21) ZN,WN,ROOTR1,ROOTR2,WNC   008620
21  FORMAT(1H0,2X5HZSP =E14.6,5X5HWSP =E14.6,8H RAD/SEC,5X7H1/TP1 =E14.008630
      1.6,5X7H1/TP2 =E14.6/27X 5H =E14.6,11H CYCLES/SEC,
      2/1H0,17HSHORT PERIOD MODE)          008640
25  PER = XKON/(WN*SQRT(1.-ABS(ZN)**2))     008650
      TTO1 = .69315/(ABS(ZN)*WN)           008670
      TTO2 = 2.30259/(ABS(ZN)*WN)         008680
      CTO1=TTO1/PER                         008690
      CTO2=TTO2/PER                         008700
      CTO3=1.0/CTO1                        008710
      CTO4=1.0/CTO2                        008720
      TZW = 2.*ZN*WN                       008730
      WNOS = (WN)**2                       008740
      IF(ZN) 110,110,402                   008750
402  WRITE (6,124)PER,TTO1,TTO2,CTO1,CTO2,CTO3,CTO4,TZW,WNOS 008760
124  FORMAT (1H0,11X8HPERIOD =E13.5, 6X19HTIM008770
      1E TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =E13.5/1H ,36X2008780
      21HCYCLES TO HALF AMP. =E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5008790
      3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO008800
      4 ONE TENTH AMP. =E13.5/50X8H2*Z*WN =E13.5,35X5HWSQ =E13.5) 008810
      RETURN                                008820
110  WRITE (6,149)PER,TTO1,TTO2,CTO1,CTO2  008830
149  FORMAT (1H0,11X8HPERIOD =E13.5,008840
      1 4X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =E13008850
      2.5/1H ,34X23HCYCLES TO DOUBLE AMP. =E13.5,14X26HCYCLES TO TEN TIME008860
      3S AMP. =E13.5)                      008870
      RETURN                                008880
20  WRITE(6,24) ZN,WN,ROOTR1,ROOTR2,WNC   008890
24  FORMAT(1H0,2X4HZP =E14.6,5X4HWP =E14.6,8H RAD/SEC,
      15X8H1/TSP1 =E14.6,5X8H1/TSP2 =E14.6/25X5H =E14.6,11H CYCLES/SEC008910
      2/1H0,16HLONG PERIOD MODE)          008920
      GO TO 25                              008930
55  IF(1.E-4-ABS(RI(2)))57,58,58          008940
57  WW1=SQRT(RR(2)**2+RI(2)**2)            008950
      GO TO 26                              008960
58  GO TO 23                              008970
      END                                    008980
      SUBROUTINE DMULR (COE,N1,ROOTR,ROOTI)  008990
C                                          009000
C                                          009010
C*****009020
C                                          009030
C POLYNOMIAL ROOT FINDER SUBROUTINE ....  009040
C                                          009050
C ITERATIVE METHOD FOR POLYNOMIAL EQUATIONS .... 009060
C                                          009070

```

LONGITUDINAL PROGRAM LISTING

```

C   THIS METHOD APPROXIMATES THE FUNCTION F(Z) BY A QUADRATIC          009080
C   WHICH MAY ,IN GENERAL, HAVE COMPLEX COEFFICIENTS AND DOES NOT    009090
C   REQUIRE A KNOWLEDGE OF THE DERIVATIVE OF F(Z) THOUGH            009100
C   THE FUNCTION F(Z) MUST BE EVALUATED ONCE PER ITERATION ....      009110
C                                                                           009120
C   THIS SUBROUTINE FINDS REAL AND COMPLEX ROOTS OF A POLYNOMIAL      009130
C   WITH REAL COEFFICIENTS ....                                       009140
C                                                                           009150
C   USE OF MULLER SUBROUTINE ....                                       009160
C   1. CALL DMULR (COE,N1,ROOTR,ROOTI) ....                             009170
C   2. COE IS THE TAG OF THE ARRAY OF COEFFICIENTS.                   009180
C   THE COEFFICIENTS MUST BE ORDERED FROM HIGHEST DEGREE            009190
C   TO LOWEST DEGREE .                                               009200
C   3. N1 IS DEGREE OF THE POLYNOMIAL .                                009210
C   4. ROOTR IS THE TAG OF THE ARRAY WHERE THE REAL PARTS            009220
C   OF THE COMPLEX ROOTS ARE STORED .                                 009230
C   5. ROOTI IS THE TAG OF THE ARRAY WHERE THE IMAGINARY              009240
C   PARTS OF THE COMPLEX ROOTS ARE STORED ....                       009250
C                                                                           009260
C   ALL ARITHMETIC IS IN THE COMPLEX MODE ....                       009270
C   THEREFORE UNDER-FLOW IS NORMAL FOR REAL ROOTS ....              009280
C                                                                           009290
C   MULTIPLE ROOTS DECREASES ACCURACY OF THIS SUBROUTINE .          009300
C   WHEN MULTIPLICITY IS FOUR THE ACCURACY DECREASES TO              009310
C   ABOUT TWO PLACES ....                                             009320
C                                                                           009330
C   RUNNING TIME IS APPROXIMATELY PROPORTIONAL TO                     009340
C   DEGREE SQUARED DIVIDED BY TWENTY ....                             009350
C   FOR DEGREE ELEVEN IT TAKES SIX SECONDS ....                       009360
C                                                                           009370
C                                                                           009380
C                                                                           009390
C                                                                           009400
C .....                                                                    009410
C                                                                           009420
C                                                                           009430
C                                                                           009440
C   DOUBLE PRECISION ROOTR,ROOTI,AXR,AXI,ALP1R,ALP1I,TEM             009450
C   DOUBLE PRECISION BET1R,BET1I,ALP2R,ALP2I,BET2R,BET2I             009460
C   DOUBLE PRECISION TEMR,TEMI,ALP3R,ALP3I,BET3R,BET3I               009470
C   DOUBLE PRECISION ALP4R,ALP4I,TE1,TE2,HELL,BELL                   009480
C   DOUBLE PRECISION TE1,TE2,TE3,TE4,TE5,TE6,TE7,TE8,TE9,TE10        009490
C   DOUBLE PRECISION TE11,TE12,TE13,TE14,TE15,TE16,DE15,DE16,COE     009500
C                                                                           009510
C   DIMENSION COE(1),ROOTR(1),ROOTI(1)                                009520
C                                                                           009530
C   N2=N1+1                                                            009540
C   N4=0                                                                009550
C   I=N1+1                                                              009560
19  IF(COE(I))9,7,9                                                    009570
7   N4=N4+1                                                            009580
   ROOTR(N4)=0.000                                                    009590
   ROOTI(N4)=0.000                                                    009600
   I=I-1                                                                009610
   IF(N4-N1)19,37,19                                                  009620
9   CONTINUE                                                            009630
C                                                                           009640
10  AXR=0.000                                                         009650
   AXI=0.000                                                         009660
   L=1                                                                009670

```

LONGITUDINAL PROGRAM LISTING

	N3=1	009680
	ALP1R=AXR	009690
	ALP1I=AXI	009700
	M=1	009710
	GO TO 99	009720
C		009730
11	BET1R=TEMR	009740
	BET1I=TEMI	009750
	AXR=0.8500	009760
	ALP2R=AXR	009770
	ALP2I=AXI	009780
	M=2	009790
	GO TO 99	009800
C		009810
12	BET2R=TEMR	009820
	BET2I=TEMI	009830
	AXR=0.900	009840
	ALP3R=AXR	009850
	ALP3I=AXI	009860
	M=3	009870
	GO TO 99	009880
C		009890
13	BET3R=TEMR	009900
	BET3I=TEMI	009910
14	TE1=ALP1R-ALP3R	009920
	TE2=ALP1I-ALP3I	009930
	TE5=ALP3R-ALP2R	009940
	TE6=ALP3I-ALP2I	009950
	TEM=TE5*TE5+TE6*TE6	009960
	TE3=(TE1*TE5+TE2*TE6)/TEM	009970
	TE4=(TE2*TE5-TE1*TE6)/TEM	009980
	TE7=TE3+1.000	009990
	TE9=TE3*TE3-TE4*TE4	010000
	TE10=2.000*TE3*TE4	010010
	DE15=TE7*BET3R-TE4*BET3I	010020
	DE16=TE7*BET3I+TE4*BET3R	010030
	TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15	010040
	TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16	010050
	TE7=TE9-1.000	010060
	TE1=TE9*BET2R-TE10*BET2I	010070
	TE2=TE9*BET2I+TE10*BET2R	010080
	TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I	010090
	TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R	010100
	TE15=DE15*TE3-DE16*TE4	010110
	TE16=DE15*TE4+DE16*TE3	010120
	TE1=TE13*TE13-TE14*TE14-4.000*(TE11*TE15-TE12*TE16)	010130
	TE2=2.000*TE13*TE14-4.000*(TE12*TE15+TE11*TE16)	010140
	TEM=DSORT(TE1*TE1+TE2*TE2)	010150
	IF(TE1) 113,113,112	010160
113	TE4=DSORT(0.500*(TEM-TE1))	010170
	IF(TE4.NE.0.00) TE3=0.500*TE2/TE4	010180
	IF(TE4.EQ.0.00) TE3=0.00	010190
	GO TO 111	010200
C		010210
112	TE3=DSORT(0.500*(TEM+TE1))	010220
	IF(TE2) 110,200,200	010230
110	TE3=-TE3	010240
200	IF(TE3.NE.0.00) TE4=0.500*TE2/TE3	010250
	IF(TE3.EQ.0.00) TE4=0.00	010260
111	TE7=TE13+TE3	010270

LONGITUDINAL PROGRAM LISTING

	TE8=TE14+TE4	010280
	TE9=TE13-TE3	010290
	TE10=TE14-TE4	010300
	TE1=2.000*TE15	010310
	TE2=2.000*TE16	010320
	IF (TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10) 204,204,205	010330
204	TE7=TE9	010340
	TE8=TE10	010350
205	TEM=TE7*TE7+TE8*TE8	010360
	TE3=(TE1*TE7+TE2*TE8)/TEM	010370
	TE4=(TE2*TE7-TE1*TE8)/TEM	010380
	AXR=ALP3R+TE3*TE5-TE4*TE6	010390
	AXI=ALP3I+TE3*TE6+TE4*TE5	010400
	ALP4R=AXR	010410
	ALP4I=AXI	010420
	M=4	010430
	GO TO 99	010440
C		010450
15	N6=1	010460
C	*****	010470
38	IF (DABS (HELL) +DABS (BELL) -1.00-20) 18,18,16	010480
16	TE7=DABS (ALP3R-AXR) +DABS (ALP3I-AXI)	010490
	IF (TE7/(DABS (AXR) +DABS (AXI)) -1.00-7) 18,18,17	010500
C	*****	010510
17	N3=N3+1	010520
	ALP1R=ALP2R	010530
	ALP1I=ALP2I	010540
	ALP2R=ALP3R	010550
	ALP2I=ALP3I	010560
	ALP3R=ALP4R	010570
	ALP3I=ALP4I	010580
	BET1R=BET2R	010590
	BET1I=BET2I	010600
	BET2R=BET3R	010610
	BET2I=BET3I	010620
	BET3R=TEMR	010630
	BET3I=TEMI	010640
	IF (N3-100) 14,18,18	010650
18	N4=N4+1	010660
	ROOTR (N4) =ALP4R	010670
	ROOTI (N4) =ALP4I	010680
	N3=0	010690
41	IF (N4-N1) 30,37,37	010700
37	RETURN	010710
C	*****	010720
30	IF (DABS (ROOTI (N4)) -1.00-5) 10,10,31	010730
31	GO TO (32,10),L	010740
32	AXR=ALP1R	010750
	AXI=-ALP1I	010760
	ALP1I=-ALP1I	010770
	M=5	010780
	GO TO 99	010790
33	BET1R=TEMR	010800
	BET1I=TEMI	010810
	AXR=ALP2R	010820
	AXI=-ALP2I	010830
	ALP2I=-ALP2I	010840
	M=6	010850
	GO TO 99	010860
C		010870

LONGITUDINAL PROGRAM LISTING

34	BET2R=TEMR	010880
	BET2I=TEMI	010890
	AXR=ALP3R	010900
	AXI=-ALP3I	010910
	ALP3I=-ALP3I	010920
	L=2	010930
	M=3	010940
99	TEMR=COE(1)	010950
	TEMI=0.000	010960
	DO 100 I=1,N1	010970
	TE1=TEMR*AXR-TEMI*AXI	010980
	TEMI=TEMI*AXR+TEMR*AXI	010990
100	TEMR=TE1+COE(I+1)	011000
	HELL=TEMR	011010
	BELL=TEMI	011020
42	IF(N4)102,103,102	011030
102	DO 101 I=1,N4	011040
	TEM1=AXR-ROOIR(I)	011050
	TEM2=AXI-ROOII(I)	011060
	TE1=TEM1*TEM1+TEM2*TEM2	011070
	TE2=(TEMR*TEM1+TEMI*TEM2)/TE1	011080
	TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1	011090
101	TEMR=TE2	011100
103	GO TO (11,12,13,15,33,34),M	011110
	END	011120

LONGITUDINAL PROGRAM DATA

10015A TRANSPORT AIRCRAFT H=10,000 FT CG=250 M=.6 TMG/9/17/66							LONG15A1
4900.	24.1	.77	745.	.0005873	32.051		LONG15A2
350000.	19000000.	2.0	30.				LONG15A3
.437	6.		6.3	.251			LONG15A4
.025	.03				.0031		LONG15A5
	-2.	-5.1	-20.3	-1.04	-.01		LONG15A6
1.3							LONG15A7
101111 MEDIUM FIGHTER, SEA LEVEL, FORWARD CG, FLAPS=30, 1.4VSTALL							LONG1111
250.	9.0	.224	250.	.002377	32.174		LONG1112
22000.	55000.						LONG1113
1.25	.064			.052			LONG1114
.06							LONG1115
	-.041	-.06	-.1	-.025			LONG1116
9.5	-3.						LONG1117
60011A TRANSPORT AIRCRAFT H=40,000 FT CG=250 M=.77 JMG 9/7/66							LONG15A1
4900.	24.1	.77	745.	.0005873	32.051		LONG15A2
350000.	19000000.	2.0	30.	10000.	2.		LONG15A3
.437	6.		6.3	.251			LONG15A4
.075	.03				.0031		LONG15A5
.00417	-2.	-5.1	-20.3	-1.04	-.01		LONG15A6
1.3							LONG15A7

ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS

RUN NO. 15A

TRANSPORT AIRCRAFT H=10,000FT CG=25C M=.6

TMG/9/17/66

INPUT DATA (STABILITY AXIS DERIVATIVES), PER RAD

S = 4.9000E+03 MAC = 2.4400E+01 MACH = 7.7000E-01 U = 7.4500E+02 RHO = 5.0730E-04 G = 3.2051E+01
 GMT = 3.5000E+05 IYY = 1.9000E+07 ZT = 2.0000E+00 LX = 3.8000E+01 TDY = 0. XI = 0.
 CL = 4.3700E-01 CLA = 6.0000E+00 CLAD = 0. CLQ = 6.3000E+00 CLDE = 2.5100E-01 CLM = 0.
 CD = 2.5000E-02 CDA = 3.0000E-02 CDAD = 0. CDD = 0. CDE = 0. CODE = 0. CDM = 3.1000E-03
 CMT = 0. CMA = -2.0000E+00 CMAD = -5.1000E+00 CMQ = -2.0300E+01 CROE = -1.0400E+00 CMH = -1.0000E-02
 ALPHA = 1.3000E+00 GAMA = 0.

DIMENSIONAL STABILITY DERIVATIVES

XU = -.5143E-02 ZU = -.8500E-01 MU = -.1047E-04
 XM = .3995E-01 ZM = -.5914E+00 MM = -.2719E-02
 XW = 0. ZW = 0. MWD = -.1122E-03
 XO = 0. ZO = -.7452E+01 HQ = -.3326E+00
 XOE = 0. ZOE = -.1836E+02 MDE = -.1054E+01
 XOY = 0. ZOT = 0. MOT = 0.
 U = .7450E+03 G = .3205E+02 GAMA = 0.
 VE = .3783E+03 LA = .5890E+00 NZA = .1369E+02
 KY = .4171E+02 DE/G = .1671E+00

THE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR ARE

ROOTS (COMPLEX FORM)
 -.21370-02 .57440-01
 -.21370-02 -.57440-01
 -.50380+00 .13960+01
 -.50380+00 -.13960+01

ZP = .37183E-01 HP = .57478E-01 RAD/SEC ZSP = .339413E+00 MSP = .144439E+01 RAD/SEC
 .914796E-02 CYCLES/SEC .236246E+00 CYCLES/SEC

SHORT PERIOD MODE

PERIOD = .45000E+01 TIME TO HALF AMP. = .13758E+01
 CYCLES TO HALF AMP. = .30573E+00
 ONE OVER CYCLES TO HALF AMP. = .32708E+01
 2*ZSP*WSP = .10076E+01
 MN/LA = .25202E+01
 TIME TO ONE TENTH AMP. = .45703E+01
 CYCLES TO ONE TENTH AMP. = .10156E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .22034E+01
 WSPSQ = .39679E+00
 LA/MN =

LONG PERIOD MODE

PERIOD = .10939E+03 TIME TO HALF AMP. = .32432E+03
 CYCLES TO HALF AMP. = .29644E+01
 ONE OVER CYCLES TO HALF AMP. = .33729E+00
 2*ZP*WP = .42745E-02
 TIME TO ONE TENTH AMP. = .10774E+04
 CYCLES TO ONE TENTH AMP. = .98488E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .10154E+00
 MPSQ = .33037E-02

COEFFICIENTS

A = .100000E+01 B = .101192E+01 C = .221102E+01 D = .127476E-01 E = .727952E-02

RUN NO. 15A ELEVATOR NUMERATOR CHARACTERISTICS

THETA PER CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)
 -.11570-01 0.
 -.53870+00 0.

1/TT1 = .115659E-01 1/TT2 = .538702E+00
 AT = -.105144E+01 BT = -.578575E+00 CT = -.655109E-02

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)
 -.36240+01 0.
 .69130+01 0.

1/TU1 = .362369E+01 1/TU2 = -.691252E+01
 AU = 0. BU = -.733392E+00 CU = .241200E+01 DU = .183706E+02

NORMAL VELOCITY PER CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)
 -.25240-02 -.60700-01
 -.25240-02 .60700-01
 -.42660+02 .20600-45

ZW = .416085E-01 WW = .607571E-01 1/TW = .426619E+02
 AW = -.183563E+02 BW = -.783208E+03 CW = -.402721E+01 DW = -.289081E+01

ALTITUDE RATE PER CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)
 -.46600-02 0.
 .48290+01 0.
 -.48190+01 0.

1/TH1 = .465971E-02 1/TH2 = -.482863E+01 1/TH3 = .481758E+01
 AH = .183563E+02 BH = -.117211E+00 CH = -.427011E+03 DH = -.198975E+01

VERTICAL ACCELERATION PER DELTA ELEVATOR
 ROOTS (COMPLEX FORM)
 -.46580-02 0.
 -.66020+00 .56530+01
 -.66020+00 -.56530+01

ZA7 = .116010E+00 WAZ = .569123E+01 1/TAZ1 = .465845E-02
 AA = .131870E+02 BA = .174745E+02 CA = .427208E+03 DA = .198975E+01

ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS

RUN NO. 111

MEDIUM FIGHTER, SEA LEVEL, FORWARD CG, FLAPS=30, 1.4VSTALL

INPUT DATA (STABILITY AXIS DERIVATIVES), PER DEG

S = 2.5000E+02 MAC = 9.0000E+00 MACH = 2.2400E-01 U = 2.5000E+02 RHO = 2.3770E-03 C = 3.2174E+01
 GMT = 2.2000E+04 IYV = 5.5000E+04 ZY = 0. LX = 0. TOT = 0. XI = 0.
 CL = 1.2500E+00 CLA = 6.4000E-02 CLAD = 0. CLQ = 0. CLOE = 5.2000E-02 CLM = 0.
 CD = 6.0000E-02 CDA = 0. CDB = 0. CDD = 0. CODE = 0. CDM = 0.
 CMT = 0. CMA = -4.1000E-02 CMAD = -6.0000E-02 CMQ = -1.0000E-01 CMDE = -2.5000E-02 CMH = 0.
 ALPHA = 9.5000E+00 GAMA = -3.0000E+00

DIMENSIONAL STABILITY DERIVATIVES

XU = -.1304E-01 ZU = -.2716E+00 MU = 0.
 XM = .1356E+00 ZM = -.4009E+00 MM = -.2855E-01
 XW = 0. ZW = 0. MW = -.7522E-03
 XO = 0. ZO = 0. MO = -.3134E+00
 XOE = 0. XOE = -.6091E+02 MOE = -.4353E+01
 XOT = 0. ZOT = 0. MOT = 0.
 U = .2500E+03 G = .3217E+02 GAMA = -.3000E+01
 VE = .2500E+03 LA = .3983E+00 NZA = .3095E+01
 KY = .8969E+01 DE/G = -.1709E+01

THE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR ARE

ROOTS (COMPLEX FORM)
 -.8943D-02
 -.1852D+00
 -.1852D+00
 -.2657D+01
 -.2657D+01

ZP = .48227AE-01 MP = .185426E+00 RAD/SEC ZSP = .167225E+00 MSP = .269533E+01 RAD/SEC
 .295116E-01 CYCLES/SEC .428976E+00 CYCLES/SEC

SHORT PERIOD MODE

PERIOD = .23644E+01 TIME TO HALF AMP. = .15379E+01
 CYCLES TO HALF AMP. = .65044E+00
 ONE OVER CYCLES TO HALF AMP. = .15379E+01
 2*ZSP*MP = .90145E+00
 MN/LA = .67663E+01
 TIME TO ONE TENTH AMP. = .51086E+01
 CYCLES TO ONE TENTH AMP. = .21606E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .46283E+00
 MSPSQ = .72648E+01
 LA/MN = .14779E+00

LONG PERIOD MODE

PERIOD = .33925E+02 TIME TO HALF AMP. = .77510E+02
 CYCLES TO HALF AMP. = .22846E+01
 ONE OVER CYCLES TO HALF AMP. = .43768E+00
 2*ZP*MP = .17885E-01
 TIME TO ONE TENTH AMP. = .25748E+03
 CYCLES TO ONE TENTH AMP. = .75899E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .13175E+00
 MPSQ = .34383E-01

COEFFICIENTS

A = .100000E+01 B = .919330E+00 C = .731532E+01 D = .160929E+00 E = .249786E+00

RUN NO. 111 ELEVATOR NUMERATOR CHARACTERISTICS

THETA PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
.5734D-01 -.1801D+00
.5734D-01 .1801D+00

ZT = -.303328E+00 WT = .189041E+00
AT = -.429187E+01 BT = .492205E+00 CT = -.153376E+00

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
-.6058D+00 -.1152D+01
-.6058D+00 .1152D+01

ZU = .465490E+00 WU = .130135E+01
AU = 0. BU = -.109875E+02 CU = -.133117E+02 DU = -.186074E+02

NORMAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
-.8569D-02 -.1847D+00
-.8569D-02 .1847D+00
-.1376D+02 .8787D-45

ZW = .453316E-01 WW = .184945E+00 1/TW = .137578E+02
AW = -.809147E+02 BW = -.111460E+04 CW = -.218454E+02 DW = -.380771E+02

ALTITUDE RATE PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
-.4871D-02 .5095D-21
-.2584D+00 -.1315D+01
-.2584D+00 .1315D+01

ZH = .192738E+00 WH = .134046E+01 1/TH3 = .487080E-02
AH = .808038E+02 BH = .421462E+02 CH = .145395E+03 DH = .707197E+00

ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS

RUN NO. 11A

TRANSPORT AIRCRAFT H=40,000FT CG=25C M=.77

JMC 9/7/66

INPUT DATA (STABILITY AXIS DERIVATIVES), PER RAD

S = 4.9000E+03 MAC = 2.4000E+01 MACH = 7.7000E-01 U = 7.4500E+02 PHO = 5.8730E-04 G = 3.2051E+01
 GNT = 3.5000E+05 IYY = 1.9000E+07 ZT = 2.6000E+00 LX = 3.0000E+01 TDY = 1.0000E+04 XI = 2.0000E+00
 CL = 4.3700E-01 CLA = 6.0000E+00 CLAD = 0. CLD = 6.3000E+00 CLO = 2.5100E-01 CLM = 0.
 CD = 7.5000E-02 CDA = 3.0000E-02 CDD = 0. CDD = 0. CDE = -1.0400E+00 CDM = 3.1000E-03
 CMT = 4.1700E-03 CMA = -2.0000E+00 CMAO = -5.1000E+00 CMQ = -2.0300E+01 CMOE = -1.0400E+00 CMH = -1.0000E-02
 ALPHA = 1.3000E+00 GAMA = 0.

DIMENSIONAL STABILITY DERIVATIVES

XU = -.1496E-01 ZU = -.8540E-01 MU = -.2181E-04
 XM = .3995E-01 ZM = -.5964E+00 MM = -.2719E-02
 XQ = 0. ZQ = 0. MQ = -.1122E-03
 XQ = 0. ZQ = 0. MQ = -.3326E+00
 XDE = 0. ZDE = -.7452E+01 MDE = -.1054E+01
 XDT = .9142E+00 ZDT = -.1836E+02 MDT = -.1053E-02
 U = .7450E+03 G = .3205E+02 GAMA = 0.
 VE = .3703E+03 LA = .5890E+00 NZA = .1369E+02
 KY = .4171E+02 DE/G = .1671E+00

THE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR ARE

ROOTS (COMPLEX FORM)
 -.70560-02 .5615D-01
 -.70560-02 -.5615D-01
 -.5063D+00 .1396D+01
 -.5063D+00 -.1396D+01

ZP = .124699E+00 WP = .565875E-01 RAD/SEC ZSP = .340927E+00 MSP = .148496E+01 RAD/SEC
 .900620E-02 CYCLES/SEC .236340E+00 CYCLES/SEC

SHORT PERIOD MODE

PERIOD = .4500E+01 TIME TO HALF AMP. = .13691E+01
 CYCLES TO HALF AMP. = .30420E+00
 ONE OVER CYCLES TO HALF AMP. = .32873E+01
 2*ZSP*WSP = .10129E+01
 WN/LA = .25212E+01
 TIME TO ONE TENTH AMP. = .45482E+01
 CYCLES TO ONE TENTH AMP. = .10105E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .98959E+00
 MSPSQ = .20951E+01
 LA/MN = .39664E+00

LONG PERIOD MODE

PERIOD = .11191E+03 TIME TO HALF AMP. = .98230E+02
 CYCLES TO HALF AMP. = .87777E+00
 ONE OVER CYCLES TO HALF AMP. = .11392E+01
 2*ZP*WP = .14413E-01
 TIME TO ONE TENTH AMP. = .32631E+03
 CYCLES TO ONE TENTH AMP. = .29159E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .34295E+00
 MSPSQ = .32022E-02

COEFFICIENTS

A = .100000E+01 B = .102664E+01 C = .222261E+01 D = .343627E-01 E = .706112E-02

RUN NO. 11A ELEVATOR NUMERATOR CHARACTERISTICS

THETA PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.2143D-01	0.
-.5435D+00	0.

1/TT1 = .214275E-01 1/TT2 = .543574E+00

AT = -.105144E+01 BT = -.594067E+00 CT = -.122466E-01

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.3645D+01	0.
.6934D+01	0.

1/TU1 = .364509E+01 1/TU2 = -.693393E+01

AU = 0. BU = -.733392E+00 CU = .241200E+01 DU = .185363E+02

NORMAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.7436D-02	-.6023D-01
-.7436D-02	.6023D-01
-.4266D+02	.1128D-45

ZW = .122536E+00 WW = .606870E-01 1/TW = .426619E+02

AX = -.183563E+02 BX = -.783388E+03 CX = -.117146E+02 DX = -.288414E+01

ALTITUDE RATE PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.1448D-01	0.
.4853D+01	0.
-.4839D+01	0.

1/TH1 = .144816E-01 1/TH2 = -.485034E+01 1/TH3 = .483929E+01

AH = .183563E+02 BH = .629834E-01 CH = -.430866E+03 DH = -.623959E+01

VERTICAL ACCELERATION PER DELTA ELEVATOR
ROOTS (COMPLEX FORM)

-.1448D-01	0.
-.6661D+00	-.5678D+01
-.6661D+00	.5678D+01

ZAZ = .116519E+00 WAZ = .571683E+01 1/TAZ1 = .144777E-01

AA = .131870E+02 BA = .177590E+02 CA = .431233E+03 DA = .623959E+01

RUN NO. 11A THRUST NUMERATOR ROOTS

THETA PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
-.36640+00 .26520+00
-.36640+00 -.26520+00

ZT = .410076E+00 WT = .452351E+00
AT = .105854E-02 BT = .775783E-03 CT = .216601E-03

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
.12220-01 .17150-20
-.51090+00 -.13970+01
-.51090+00 .13970+01

ZU = .343399E+00 WU = .148748E+01 1/TU = -.122178E-01
AU = .914224E+00 BU = .922798E+00 CU = .201139E+01 DU = -.247142E-01

NORMAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
-.14480-01 0.
-.66610+00 -.56780+01
-.66610+00 .56780+01

ZW = .116519E+00 WW = .571683E+01 1/TW = .144777E-01
AW = -.183563E+02 BW = -.783398E+03 CW = -.117146E+02 DW = -.268414E+01

ALTITUDE RATE PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
-.77950-02 .61840-01
-.77950-02 -.61840-01
-.42700+02 .14920-47

ZH = .125063E+00 WH = .623307E-01 1/TH3 = .427041E+02
AH = .183563E+02 BH = .784177E+03 CH = .122926E+02 DH = .304551E+01

VERTICAL ACCELERATION PER DELTA ELEVATOR
ROOTS (COMPLEX FORM)
-.77990-02 -.61840-01
-.77990-02 .61840-01
-.42630+02 -.27480-45

7AZ = .125126E+00 WAZ = .623298E-01 1/TAZ1 = .426316E+02
AA = -.183881E+02 BA = -.784200E+03 CA = -.122991E+02 DA = -.304551E+01

RUN NO. 11A COUPLING NUMERATOR ROOTS

THETA TO ELEVATOR, LONGITUDINAL VELOCITY TO THRUST

1/TTU = .546932E+00

ATU = -.961255E+00 BTU = -.525741E+00

NORMAL VELOCITY TO ELEVATOR, LONGITUDINAL VELOCITY TO THRUST

1/TWU1 = -.335089E-02 1/TWU2 = .426651E+02

AWU = -.167818D+02 BWU = -.715942D+03 CWU = .239923D+01

THETA TO ELEVATOR, NORMAL VELOCITY TO THRUST

1/TTW = .111396E+01

ATW = .748566E-01 BTW = .833872E-01

S TIMES THETA TO ELEVATOR, ALTITUDE TO THRUST

1/TTH = .111396E+01

ATH = -.748566E-01 BTH = -.833872E-01

S TIMES LONGITUDINAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR

1/TUH1 = -.485163E+01 1/TUH2 = .484010E+01

AUH = .167818D+02 BUH = -.193459D+00 CUH = -.394076D+03

S TIMES NORMAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR

1/TWH = .111396E+01

AWH = 0. BWH = .557682D+02 CWH = .621234D+02

S TIMES LONGITUDINAL VELOCITY TO THRUST, ACCELERATION TO ELEVATOR

1/TUAZ1 = -.278905E+00 1/TUAYZ = -.117200E+03

AUAZ = .120558D+02 BUAZ = -.141630D+04 CUAZ = .394076D+03

PLEASE RETURN PAPER

CONFIDENTIAL - SECURITY INFORMATION

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

LATERAL-DIRECTIONAL PROGRAM

LATERAL-DIRECTIONAL PROGRAM LISTING

```

PROGRAM LATE(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,PLOT)          000100
C  LATERAL DIRECTIONAL TRANSFER FUNCTIONS                          000110
  DIMENSION ROOTR(5),ROOTI(5),B(4),P(4),RA(4),AYP(6),DEL(5)        000120
  DIMENSION ROOTRD(5), ROOTID(5),TITLE(21)                        000130
  DIMENSION APB(3),APSB(3),APAY(4),IND(8,4),DATA(438)             000140
  COMPLEX COM,COMA,COMB,ANUM,ADEN                                000150
  COMMON /AA/ CON,CONA,COM,ANUM,ADEN,DTR,IROOT,TR,TS,ZD,WD,E,PER,  000160
  1AP,BP,CP,DP,AB,BB,CB,DB,IOPT,A,TA,TB,TC,DATA,TITLE,PLT,IPLT  000170
  2,RUN,WDD                                                       000180
  COMMON/BB/RHO,U,S,GWT,BSPAN,ZIXB,G,ALFAI,GAMA,ALX,CX(18),ALFAA,  000190
  A ALFAX,PLO,YB,YBD,YP,YR,YDA,ALB,ALBD,ALP,ALR,ALDA,ALDR,ANB,ANBD,  000200
  B ANP,ANR,ANDA,ANDR,ALBP,ALBDP,ALPP,ALRP,ALDAP,ALDRP,ANBP,ANBDP,  000210
  C ANPP,ANRP,ANDAP,ANDRP,ZIZB,ZIXZB,YDR                          000220
  DOUBLE PRECISION ROOTRD, ROOTID, B, P, RA, AYP, DEL,APB,APSB,APAY 000230
  DATA(IND(I,1),I=1, 8)/ 8*6H /, IND(1,2) /75H FOR B, DA000240
  1, AND DR DERIVATIVES, AND PER RADIAN FOR BD, P, AND R DERIVATIVES/000250
  2,IND(1,3)/52H FOR SIDESLIP DERIVATIVES, PER RADIAN FOR ALL OTHERS/000260
  3,IND(1,4)/51H FOR CONTROL DERIVATIVES, PER RADIAN FOR ALL OTHERS/ 000270
  4,(IND(I,3),I= 6,8 )/3*6H /,(IND(I,4),I= 6,8 )/3*6H / 000280
  IPLT=0.$PLO=0.                                                 000290
  JJXX=0                                                         000300
250 READ(5,11) M,J,K,RUN,IOPT,(TITLE(I),I=1,11)                 000310
  IF(EOF(5).NE.0)STOP                                           000320
  11 FORMAT(I1,I1,I1,A3,I3,10A6,A3)                               000330
  WRITE(6,175)RUN,(TITLE(I),I=1,11)                              000340
  175 FORMAT(1H1,3X, 28H ROOTS OF A/C LATERAL ,                 000350
  * 30HDIRECTIONAL TRANSFER FUNCTIONS,/1H0,36X,                 000360
  * 8HRUN NO. ,A3/1H0,7X,10A6,A3)                               000370
C FOR M=1 USE DIMENSIONAL INPUT DATA (STAB AXIS)                000380
C M=0 USE NONDIMENSIONAL INPUT DATA (STAB AXIS)                000390
C FOR J=0, USE NON-DIMEN. STAB. DERIVATIVES WITH UNITS OF 1 PER RADIAN. 000400
C FOR J=1, USE NON-DIMEN. STAB. DERIVATIVES WITH UNITS OF 1 PER DEGREE. 000410
C FOR J=2 USE NON-DIMEN. STAB. DERIVATIVES WITH UNITS PER DG FOR B,DA 000420
C AND DR DERIV, AND PER RADIAN FOR BD,P, AND R DERIF.          000430
C FOR K=1 USE PRIMED DIMENSIONAL INPUT DATA (STAB AXIS)        000440
C K=0 USE UNPRIMED DIMENSIONAL INPUT DATA (STAB AXIS)          000450
  IF(M.LT.2)GO TO 1143                                           000460
  JJXX=1                                                         000470
  M=M-5                                                           000480
  1143 IF(M)143,144,143                                          000490
  143 IF(K)90,142,90                                             000500
  144 IF(J.GT.4)CALL CHNG(J)                                     000510
  IF(J.GT.4)PLT=PLO                                             000520
  IF(J.LE.4)READ(5,13)RHO,U,S,GWT,BSPAN,ZIXB,ZIZB,ZIXZB,G,ALFAI,  000530
  A GAMA,ALX,(CX(I1),I1=1,18),ALFAA,ALFAX,PLT                    000540
  CYB=CX(1)$CYBD=CX(2)$CYP=CX(3)$CYR=CX(4)$CYDA=CX(5)$CYDR=CX(6)  000550
  CLB=CX(7)$CLBD=CX(8)$CLP=CX(9)$CLR=CX(10)$CLDA=CX(11)$CLDR=CX(12)  000560
  CNB=CX(13)$CNBD=CX(14)$CNP=CX(15)$CNR=CX(16)$CNDA=CX(17)        000570
  CNDR=CX(18)                                                    000580
  IF(J.GT.4)J=J-5                                               000590
  13 FORMAT(6E12.0)                                             000600
  IF(J)168,167,168                                             000610
  167 WRITE (6,202)RHO,U,S,GWT ,BSPAN,ZIXB, ZIZB,ZIXZB000620
  1,C,ALFAI,GAMA,ALX,CYB,CYBD,CYP,CYR,CYDA,CYDR, CLB,CLBD,C000630
  2LP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA,CNDR 000640
  3 ,ALFAA,ALFAX 000650
  202 FORMAT(1H0,5X50HINPUT DATA (NON-DIMENSIONAL) PER RADIAN  000660
  1 /1H0,6H RHO =E12.4,6X3HU =E12.4,6X3HS =E12.4,4X5HGWT =E12.4,  000670
  2 2X7H SPAN =E12.4,4X5HIXB =E12.4,7H IZB =E12.4,3X6HIXZB =E12.4,  000680
  3 6X3HG =E12.4,2X7H ALFI =E12.4,3X6HGAMA =E12.4,5X4HLX =E12.4/ 000690

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

4 7H  CYB =E12.4,3X6HCYBD =E12.4,4X5HCYP =E12.4,4X5HCYR =E12.4,      000700
5 3X6HCYDA =E12.4,3X6HCYDR =E12.4,7H  CLB =E12.4,3X6HCLBD =E12.4,      000710
6 4X5HCLP =E12.4,4X5HCLR =E12.4,3X6HCLDA =E12.4,3X6HCLDR =E12.4,      000720
7 7H  CNB =E12.4,3X6HCNBD =E12.4,4X5HCNP =E12.4,4X5HCNR =E12.4,      000730
8 3X6HCNDA =E12.4,3X6HCNDR =E12.4,      000740
9 /1X,6HALFA =,E12.4,2X7H ALFX =,E12.4)      000750
GO TO 1000      000760
168 WRITE(6,166)(IND(I,J),I=1,8 ),RHO,U,S,GWT,BSPAN,ZIXB,ZIZB,ZIXZB      000770
1,G,ALFAI,GAMA,ALX,CYB,CYBD,CYP,CYR,CYDA,CYDR,      CLB,CLBD,C000780
2LP,CLR,CLOA,CLDR,CNB,CNBD,CNP,CNR,CNDA,CNDR,ALFAA,ALFAX      000790
166 FORMAT(1H0,6X39HINPUT DATA (NON-DIMENSIONAL) PER DEGREE,7A10,A5      000800
1 /1H0,6H RHO =E12.4,6X3HU =E12.4,6X3HS =E12.4,4X5HGWT =E12.4,      000810
2 2X7H SPAN =E12.4,4X5HIXB =E12.4,7H  IZB =E12.4,3X6HIXZB =E12.4,      000820
3 6X3HG =E12.4,2X7H ALFI =E12.4,3X6HGAMA =E12.4,5X4HLX =E12.4,      000830
4 7H  CYB =E12.4,3X6HCYBD =E12.4,4X5HCYP =E12.4,4X5HCYR =E12.4,      000840
5 3X6HCYDA =E12.4,3X6HCYDR =E12.4,7H  CLB =E12.4,3X6HCLBD =E12.4,      000850
6 4X5HCLP =E12.4,4X5HCLR =E12.4,3X6HCLDA =E12.4,3X6HCLDR =E12.4,      000860
7 7H  CNB =E12.4,3X6HCNBD =E12.4,4X5HCNP =E12.4,4X5HCNR =E12.4,      000870
8 3X6HCNDA =E12.4,3X6HCNDR =E12.4,      000880
9 /1X,6HALFA =,E12.4,2X7H ALFX =,E12.4)      000890
DTR=57.295779      000900
IF(J.EQ.4) GO TO 1104      000910
CYB=CYB*DTR      000920
CLB=CLB*DTR      000930
CNB=CNB*DTR      000940
IF(J.EQ.3) GOTO 1000      000950
1104 IF(J.EQ.4) J=2      000960
CYDA=CYDA*DTR      000970
CYDR=CYDR*DTR      000980
CLDA=CLDA*DTR      000990
CLDR=CLDR*DTR      001000
CNDA=CNDA*DTR      001010
CNDR=CNDR*DTR      001020
IF(J.EQ.2) GO TO 1000      001030
CYBD=CYBD*DTR      001040
CYP=CYP*DTR      001050
CYR=CYR*DTR      001060
CLBD=CLBD*DTR      001070
CLP=CLP*DTR      001080
CLR=CLR*DTR      001090
CNBD=CNBD*DTR      001100
CNP=CNP*DTR      001110
CNR=CNR*DTR      001120
1000 ALFA2=(ALFAX-ALFAA)/57.295779      001130
SINA=SIN(ALFA2)      001140
COSA=COS(ALFA2)      001150
SCLOA=CLDA*SINA      001160
CLOA=CLDA*COSA-CNDA*SINA      001170
CNDA=CNDA*COSA+SCLOA      001180
SCLDR=CLDR*SINA      001190
CLDR=CLDR*COSA-CNDR*SINA      001200
CNDR=CNDR*COSA+SCLDR      001210
SCLBD=CLBD*SINA      001220
CLBD=CLBD*COSA-CNBD*SINA      001230
CNBD=CNBD*COSA+SCLBD      001240
SCLB=CLB*SINA      001250
CLB=CLB*COSA-CNB*SINA      001260
CNB=CNB*COSA+SCLB      001270
SCYP=CYP*SINA      001280
CYP=CYP*COSA-CYR*SINA+ALFAX/57.295779      001290

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

CYP=CYP*COA+SCYP                                001300
SCLP=CLP*SINA**2                                001310
SCCLP=(CLP-CNR)*SINA*COA                        001320
SCLR=CLR*SINA**2                                001330
SCCLR=(CLR+CNP)*SINA*COA                        001340
CLP=CLP*COA**2+CNR*SINA**2-SCCLR              001350
CLR=CLR*COA**2-CNP*SINA**2+SCCLR              001360
CNP=CNP*COA**2-SCLR+SCCLR                      001370
CNR=CNR*COA**2+SCLP+SCCLR                      001380
GO TO 96                                         001390
146 RSU=RHO*S*U                                  001400
    ZMASS=GWT/32.174                             001410
    RSUM=RSU/ZMASS                               001420
    RSUX=RSU*BSPAN/ZIXS                          001430
    RSUZ=RSU*BSPAN/ZIZS                          001440
    YV=(RSUM/2.0)*CYB                            001450
    YB=U*YV                                       001460
    YVD=(RSUM*BSPAN/(4.0*U))*CYB                 001470
    YBD=U*YVD                                     001480
    YP=(RSUM*BSPAN/4.0)*CYP                      001490
    YR=(RSUM*BSPAN/4.0)*CYR                      001500
    YDA=(RSUM*U/2.0)*CYDA                       001510
    YDR=(RSUM*U/2.0)*CYDR                       001520
    ALB=(RSUX*U/2.0)*CLB                         001530
    ALBD=(RSUX*BSPAN/4.0)*CLBD                  001540
    ALP=(RSUX*BSPAN/4.0)*CLP                    001550
    ALR=(RSUX*BSPAN/4.0)*CLR                    001560
    ALDA=(RSUX*U/2.0)*CLDA                      001570
    ALDR=(RSUX*U/2.0)*CLDR                      001580
    ANB=(RSUZ*U/2.0)*CNB                        001590
    ANBD=(RSUZ*BSPAN/4.0)*CNBD                 001600
    ANP=(RSUZ*BSPAN/4.0)*CNP                    001610
    ANR=(RSUZ*BSPAN/4.0)*CNR                    001620
    ANDA=(RSUZ*U/2.0)*CNDA                      001630
    ANDR=(RSUZ*U/2.0)*CNDR                     001640
    WRITE(6,300)YB,YBD,YP,YR,YDA,YDR,ALB,ALBD,ALP,ALR,ALDA,ALDR,
    ANB,ANBD,ANP,ANR,ANDA,ANDR                  001650
300 FORMAT(1H0,5X33HDIMENSIONAL STABILITY DERIVATIVES 001660
1 /1H0,2X4HYB =E12.4,4X5HYBD =E12.4,5X4HYP =E12.4,5X4HYR =E12.4, 001680
2 4X5HYDA =E12.4,4X5HYDR =E12.4/3X4HLB =E12.4,4X5HLBD =E12.4, 001690
3 5X4HLP =E12.4,5X4HLR =E12.4,4X5HLDA =E12.4,4X5HLDR =E12.4/ 001700
4 3X4HNB =E12.4,4X5HNB =E12.4,5X4HNP =E12.4,5X4HNR =E12.4, 001710
5 4X5HNOA =E12.4,4X5HNDR =E12.4) 001720
GO TO 145                                       001730
142 IF(IJ.GT.4)CALL CHNG(IJ)                    001740
    IF(IJ.GT.4)PLT=PL0                           001750
    IF(IJ.GT.4)GO TO 1101                        001760
    READ(5,12)U,G,ALFAI,GAMA,ALX,ZIXB,ZIZB,ZIXZB,YB,YBD,
    1P,YR,YDA,YDR,ALB,ALBD,ALP,ALR,ALDA,ALDR,
    2,ANP,ANR,ANDA,ANDR
    3,ALFAA,ALFAX,PLT
    ANB,ANBD,ANP,ANR,ANDA,ANDR                  001770
1101 IF(IJ.GT.4)J=J-5                            001780
12  FORMAT(6E12.0)                               001790
    WRITE(6,203)U,G,ALFAI,GAMA,ALX,ZIXB,ZIZB,ZIXZB,YB,YBD,YP,
    1YR,YDA,YDR,ALB,ALBD,ALP,ALR,ALDA,ALDR,
    2NP,ANR,ANDA,ANDR
    3,ALFAA,ALFAX
    ANB,ANBD,ANP,ANR,ANDA,ANDR                  001800
203 FORMAT(1H0,5X39HINPUT DATA (DIMENSIONAL, UNPRIMED) 001810
1 /1H0,3X3HU =E12.4,6X3HC =E12.4,2X7H ALFI =E12.4,3X6HGAMA =E12.4, 001820
2 5X4HLX =E12.4,4X5HIXB =E12.4/2X5HIZB =E12.4,3X6HIXZB =E12.4, 001830

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

3 5X4HYB =E12.4,4X5HYBD =E12.4,5X4HYP =E12.4,5X4HYR =E12.4/      001900
4 2X5HYDA =E12.4,4X5HYDR =E12.4,5X4HLB =E12.4,4X5HLBD =E12.4,      001910
5 5X4HLP =E12.4,5X4HLR =E12.4/2X5HLDA =E12.4,4X5HLDR =E12.4,      001920
6 5X4HNB =E12.4,4X5HNB =E12.4,5X4HNP =E12.4,5X4HNR =E12.4/      001930
7 2X5HNDA =E12.4,4X5HNDR =E12.4,3X6HALFA =E12.4,3X6HALFX =E12.4)  001940
TV=YB/U
YVD=YBD/U
96 DTR=57.295779
ADD=(ALFAI-ALFAX)/DTR
SA=SIN(ADD)
CA=COS(ADD)
TAA=2.0*ADD
STA=SIN(TAA)
CTA=COS(TAA)
ZIXS=ZIXB*CA**2 +ZIZB*SA**2 -ZIXZB*STA
ZIZS=ZIZB*CA**2 +ZIXB*SA**2 +ZIXZB*STA
ZIXZS=ZIXZB*CTA+.5*(ZIXB-ZIZB)*STA
IF(M.NE.1)GO TO 146
145 XM=ZIXZS/ZIXS
ZM=ZIXZS/ZIZS
DXZ=1.0-((IABS(ZIXZS)**2)/(ZIXS*ZIZS))
ALBP=(ALB+XM*ANB)/DXZ
ALRBP=(ALBD+XM*ANBD)/DXZ
ANBP=(ANB+ZM*ALB)/DXZ
ANBDP=(ANBD+ZM*ALBD)/DXZ
ALPP=(ALP+XM*ANP)/DXZ
ANPP=(ANP+ZM*ALP)/DXZ
ALRP=(ALR+XM*ANR)/DXZ
ANRP=(ANR+ZM*ALR)/DXZ
ALDAP=(ALDA+XM*ANDA)/DXZ
ANDAP=(ANDA+ZM*ALDA)/DXZ
ALDRP=(ALDR+XM*ANDR)/DXZ
ANDRP=(ANDR+ZM*ALDR)/DXZ
YP=YB+U*SIN(ALFAX/DTR)
YR=YB+U*(1.-COS(ALFAX/DTR))
WRITE(6,301)ALFAI,ALFAA,ALFAX,ZIXS,ZIZS,ZIXZS,ALBP,ALBDP,
1 ALPP,ALRP,ALDAP,ALDRP,ANBP,ANBDP,ANPP,ANRP,ANDAP,ANDRP
301 FORMAT(1H0,5X4DH DIMENSIONAL STABILITY DERIVATIVES PRIMED
1 /1H0,6HALFI =E12.4,3X6HALFA =E12.4,3X6HALFX =E12.4,
2 5X4HIX =E12.4,5X4HIZ =E12.4,4X5HIXZ =E12.4,
X /7H LBP =E12.4,3X6HLBDP =E12.4,4X5HLPP =E12.4,
3 4X5HLRP =E12.4,3X6HLDAP =E12.4,3X6HLDRP =E12.4/7H NBP =E12.4,
4 3X6HNBDP =E12.4,4X5HNPP =E12.4,4X5HNRP =E12.4,3X6HNDAP =E12.4,
5 3X6HNDRP =E12.4)
GO TO 112
90 IF(J.GT.4)CALL CHNG(J)
IF(J.GT.4)PLT=PLO
IF(J.GT.4)GO TO 1100
READ(5,10)U,G,GAMA,ALX,YB,YBD,YP,YR,YDA,YDR,ALBP,ALBDP,ALPP,
1 ALRP,ALDAP,ALDRP,ANBP,ANBDP,ANPP,ANRP,ANDAP,ANDRP,PLT
ALFAX = 0.
ALFAA = 0.
ALFAI = 0.
1100 IF(J.GT.4)J=J-5
10 FORMAT(6E12.0)
WRITE(6,204)U,G,GAMA,ALX,YB,YBD,YP,YR,YDA,YDR,ALBP,ALBDP,
1 ALPP,ALRP,ALDAP,ALDRP,ANBP,ANBDP,ANPP,ANRP,ANDAP,ANDRP
204 FORMAT(1H0,5X4SH INPUT DATA (DIMENSIONAL, PRIMED)
1/1H03X3HU =E12.4,6X3HG =E12.4,3X6HGAMA =E12.4,5X4HLX =E12.4,
2 5X4HYB =E12.4,4X5HYBD =E12.4/3X4HYP =E12.4,5X4HYR =E12.4,

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

3 4X5HYDA =E12.4,4X5HYDR =E12.4,4X5HLBP =E12.4,3X6HLBDP =E12.4/ 002500
4 2X5HLPP =E12.4,4X5HLRP =E12.4,3X6HLDAP =E12.4,3X6HLDRP =E12.4, 002510
5 4X5HNBP =E12.4,3X6HNBDP =E12.4,2X5HNPP =E12.4,4X5HNRP =E12.4, 002520
6 3X6HNDAP =E12.4,3X6HNDRP =E12.4) 002530
YV=YB/U 002540
YVD=YBD/U 002550
112 DTR=57.295779 002560
XKON=2.0*3.14159 002570
GDD=(GAMA*ALFAK)/DTR 002580
SG=SIN(GDD) 002590
CG=COS(GDD) 002600
GSG=G*SG 002610
GCG=G*CG 002620
RZERO=0.0 002630
IROOT=1 002640
IF(PLT.GT.0. AND.IPLT.EQ.0)CALL PLOTS(DATA,438) 002650
C LATERAL-DIRECTIONAL DENOMINATOR 002660
A=1.0-YVD 002670
BD=-ALPP-ANRP-YV*ANBDP*(1.0-(YR/U))-ALBDP*(YP/U) 002680
1 +YVD*(ANRP+ALPP) 002690
C=ANRP*ALPP-ALRP*ANPP+ANBP*(1.0-(YR/U))+YV*(ALPP+ANRP) 002700
1 -(YP/U)*ALBP-ANBDP*(ALPP*(1.0-(YR/U))+YP/U)*ALRP+(GSG/U) 002710
2 +ALBDP*(ANPP*(1.0-(YR/U))+YP/U)*ANRP-(GCG/U) 002720
3 +YVD*(ALRP*ANPP-ANRP*ALPP) 002730
D=ALRP*ANPP*YV-ANRP*ALPP*YV+(YP/U)*(ANRP*ALBP-ALRP*ANBP) 002740
1 +(1.0-(YR/U))*(ALBP*ANPP-ALPP*ANBP)-(GCG/U)*ALBP 002750
2 -(GSG/U)*ANBP+ANBDP*(GSG/U)*ALPP-(GCG/U)*ALRP 002760
3 +ALBDP*(GCG/U)*ANRP-(GSG/U)*ANPP 002770
E=(GCG/U)*(ANRP*ALBP-ALRP*ANBP)+(GSG/U)*(ALPP*ANBP-ALBP*ANPP) 002780
WRITE(6,176) 002790
176 FORMAT(1H0,15X,37HLATERAL DIRECTIONAL DENOMINATOR ROOTS) 002800
DEL(1)=A 002810
DEL(2)=BD 002820
DEL(3)=C 002830
DEL(4)=D 002840
DEL(5)=E 002850
N=4 002860
CALL DMULR (DEL,N,ROOTRD,ROOTID) 002870
M=1 002880
66 WRITE(6,401) 002890
401 FORMAT(1H,20HROOTS (COMPLEX FORM)) 002900
IF(N.EQ.3) GO TO 1007 002910
IF(N.EQ.5.AND.JXY.EQ.1) GO TO 1007 002920
WRITE(6,403)RZERO,RZERO 002930
403 FORMAT(5X,F4.1,13X,F4.1) 002940
1007 DO 1002 I=1,N 002950
IF(DABS(ROOTID(I)).LT.1.0-5)GO TO 1001 002960
WRITE(6,21)ROOTRD(I),ROOTID(I) 002970
21 FORMAT(1H,3XC12.4,5XD12.4) 002980
GO TO 1002 002990
1001 WRITE(6,21)ROOTRD(I) 003000
1002 CONTINUE 003010
DO 800 I=1,N 003020
ROOTR(I)=ROOTRD(I) 003030
800 ROOTI(I)=ROOTID(I) 003040
GO TO (94,67,72,73,80),M 003050
94 IF(1.E-4-ABS(ROOTI(1)))119,120,120 003060
119 W1=SQRT (ROOTR(1)**2+ROOTI(1)**2) 003070
WD1=ABS(ROOTI(1)) 003080
Z1=-ROOTR(1)/W1 003090

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

W3=W1/XKON                                003100
WD3=WD1/XKON                                003110
115 IF(1.E-4-ABS(ROOTI(3)))116,117,117      003120
116 WD=SQRT(ROOTR(3)**2+ROOTI(3)**2)        003130
      WDD=ABS(ROOTI(3))                      003140
      ZD=-ROOTR(3)/WD                       003150
      W4=WD/XKON                            003160
      WD4=WDD/XKON                          003170
      IF(ABS(ROOTI(1)).LT..001)GO TO 91      003180
      IPOOT=2                                003190
      IF(WD-W1)173,173,174                  003200
174 WRITE(6,39)Z1,W1,ZD,WD,WDD,W3,W4,WD4    003210
      I1=1                                   003220
      GO TO 222                              003230
173 WRITE(6,39)ZD,WD,Z1,W1,WD1,W4,W3,WD3    003240
39  FORMAT(1H0, 04H21 =,E14.6,1X, 04HW1 =,E14.6,1X, 07HRAD/SEC,4X, 04003250
      *H22 =, E14.6,1X, 04HW2 =,E14.6,1X, 07HRAD/SEC,4X, 06HWDDR =,E1003260
      *4.6,1X,07HRAD/SEC,/24X,01H=,E14.6,1X,10HCYCLES/SEC,23X,01H=
      *,E14.6,1X, 10HCYCLES/SEC, 6X, 01H=,E14.6,1X, 10HCYCLES/SEC) 003280
      DUMB=Z1                                003290
      Z1=ZD                                  003300
      ZD=DUMB                                003310
      STUPE=W1                              003320
      W1=WD                                  003330
      WD=STUPE                              003340
      STUPIO=WD1                            003350
      WD1=WDD                               003360
      WDD=STUPIO                            003370
      I1=1                                   003380
      GO TO 222                              003390
120 TD1=-1./ROOTR(1)                        003400
      ROOTI(1)=0.0                          003410
      IF(1.E-4-ABS(ROOTI(2)))130,131,131    003420
130 WD=SQRT(ROOTR(2)**2+ROOTI(2)**2)        003430
      WDD=ABS(ROOTI(2))                    003440
      ZD=-ROOTR(2)/WD                      003450
      W4=WD/XKON                            003460
      WD4=WDD/XKON                          003470
      TD2=-1./ROOTP(4)                      003480
      GO TO 91                              003490
131 TD2=-1./ROOTR(2)                       003500
      GO TO 115                              003510
91  I1=2                                     003520
      IF(ABS(TD1).LT.ABS(TD2))GO TO 89      003530
      WRITE(6,170)TD1,TD2,ZD,WD,WDD,W4,WD4 003540
170 FORMAT(1H0,1X4HTS =E14.6,3X,4HTR =E14.6,3X,5HZDR =E14.6,
      1 3X,5HWDR =E14.6,8H RAD/SEC,6X,6HWDDR =E14.6,8H RAD/SEC,
      2 /1H ,69X, 01H=,E14.6,11H CYCLES/SEC,8X, 01H=,E14.6,11H CYCLES/SEC)003570
      *C)
      TS=TD1                                003580
      TP=TD2                                003590
      I1=2                                   003600
      GO TO 222                              003610
89  WRITE(6,170)TD2,TD1,ZD,WD,WDD,W4,WD4    003620
      TS=TD2                                003640
      TP=TD1                                003650
      GO TO 222                              003660
117 TD3=-1./ROOTR(3)                        003670
      TD4=-1./ROOTR(4)                      003680
      WD=W1                                  003690

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

W00=W01                                003700
ZD=Z1                                    003710
IF(ABS(ROOTI(1)).GT..001)GO TO 109      003720
WRITE (6,171)T01,T02,T03,T04           003730
IROOT=0                                  003740
171 FORMAT(1H0,7H T1 =E14.6,4X,7H T2 =E14.6,4X,7H T3 =E14.6, 003750
1 4X6H T4 =E14.6)                       003760
GO TO 221                                 003770
109 I1=2                                  003780
IF(ABS(T03).LT.ABS(T04))GO TO 124       003790
WRITE(6,170)T03,T04,Z1,W1,W01,W3,W03    003800
TS=T03                                    003810
TR=T04                                    003820
GO TO 222                                 003830
124 WRITE(6,170)T04,T03,Z1,W1,W01,W3,W03 003840
TS=T04                                    003850
TR=T03                                    003860
222 PER=XKON/(W0*SQRT(1.-ABS(ZD)**2))     003870
TDR=XKON/W0                               003880
TT01=.69315/(ABS(ZD)*W0)                 003890
TT02=2.30259/(ABS(ZD)*W0)               003900
CT01=TT01/PER                             003910
CT02=TT02/PER                             003920
CT03=1.0/CT01                             003930
CT04=1.0/CT02                             003940
IF(ZD)223,223,224                         003950
224 WRITE(6,114)TDR,TT01,TT02,PER,CT01,CT02,CT03,CT04 003960
114 FORMAT(1H0,1X17HDUTCH ROLL MODE /1H0,6X6HTDR =E13.5, 003970
1 13X19HTIME TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =, 003980
2 E13.5,/1H ,6X6HTDDR =E13.5,11X21HCYCLES TO HALF AMP. =, 003990
X E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5, 004000
3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO 004010
4 ONE TENTH AMP. =E13.5)                 004020
TZW=2.*Z0*W0                              004030
WNOSQ=W0*W0                                004040
WRITE(6,600)TZW,WNOSQ                     004050
600 FORMAT(48X,10H2*Z0*WDR =,E13.5,33X,7HWDRSO =,E13.5) 004060
GO TO 165                                  004070
223 WRITE(6,402)TDR,TT01,TT02,PER,CT01,CT02 004080
402 FORMAT(1H0,1X15HDUTCH ROLL MODE,/1H0,11X6HTDR =E13.5, 004090
1 6X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =, 004100
2 E13.5,/1H ,11X,6HTDDR =E13.5,4X,23HCYCLES TO DOUBLE AMP. =, 004110
3 E13.5,14X26HCYCLES TO TEN TIMES AMP. =E13.5) 004120
TZW=2.*Z0*W0                              004130
WNOSQ=W0*W0                                004140
WRITE(6,600)TZW,WNOSQ                     004150
GO TO(149,221),I1                          004160
165 PER=XKON/(W1*SQRT(1.-ABS(Z1)**2))     004170
TDR=XKON/W1                               004180
TT01=.69315/(ABS(Z1)*W1)                 004190
TT02=2.30259/(ABS(Z1)*W1)               004200
CT01=TT01/PER                             004210
CT02=TT02/PER                             004220
CT03=1.0/CT01                             004230
CT04=1.0/CT02                             004240
IF(Z1)164,164,169                         004250
169 WRITE(6,177)TDR,TT01,TT02,PER,CT01,CT02,CT03,CT04 004260
TZWLP=2.*W1*Z1                            004270
WNOSQL=W1*W1                              004280
WRITE(6,600)TZWLP,WNOSQL                 004290

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

177 FORMAT(1H0,1X17HLONG PERIOD MODE /1H0,6X6HTDR =E13.5,      004300
1 13X19HTIME TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =, 004310
2 E13.5,/1H ,6X6HTDDR =E13.5,11X21HCYCLES TO HALF AMP. =,      004320
X E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5,                  004330
3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO 004340
4 ONE TENTH AMP. =E13.5)
GO TO 221
004350
164 WRITE(6,178)TDR,TT01,TT02,PER,CT01,CT02                      004360
178 FORMAT(1H0,1X16HLONG PERIOD MODE,/1H0,11X6HTDR =E13.5,      004370
1 6X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =, 004380
2 E13.5,/1H ,11X,6HTDDR =E13.5,4X,23HCYCLES TO DOUBLE AMP. =, 004390
3 E13.5,14X26HCYCLES TO TEN TIMES AMP. =E13.5)                  004400
T7WLP=2.*W1*Z1
004410
WNOSQL=W1*W1
004420
221 WRITE (6,201)A,BD,C,D,E
004430
201 FORMAT(1H0,37X12HCOEFFICIENTS/1H0,4X3HA =E13.5,3X3HB =E13.5, 004440
1 3X3HC =E13.5,3X3HD =E13.5,3X3HE =E13.5)
004450
CON = -ZD*WD
004460
CONA = WD*SQRT(1.-ABS(ZD)**2)
004470
COM = CMPLX(CON,CONA)
004480
COMA = COM*COM
004490
COMB = COMA*COM
004500
ANUM = (ALBDP*(YR/U-1.)+ALRP-YVD*ALRP)*COMA
004510
1 +((YR/U-1.)*ALBP+ALBDP*GSG/U-ALRP*YV)*COM+ALBP*GSG/U
004520
ADEN = (YR/U-1.)*COMB+(ALRP*YV/U+GSG/U-ALPP*(YR/U-1.))*COMA
004530
1 +(ALRP*GCG/U-ALPP*GSG/U)*COM
004540
PTOB = SQRT((REAL(ANUM)**2+AIMAG(ANUM)**2)/
004550
1 (REAL(ADEN)**2+AIMAG(ADEN)**2))
004560
WRITE (6,500)PTOB
004570
500 FORMAT(/2X,19HPHI TO BETA RATIO =,E12.4)
004580
SIGMA=RHO/2,3769E-03
004590
PVMAG=DTR*PTOB/(U*SQRT(SIGMA))
004600
WRITE (6,502)PVMAG
004610
502 FORMAT(/2X,18HPHI TO EQUIV VEL =,E12.4)
004620
FSPTOB=WD**2*PTOB
004630
WRITE(6,504)FSPTOB
004640
504 FORMAT(/2X,38HFREQ SQUARED TIMES PHI TO BETA RATIO =,E12.4)
004650
C
004660
AILERON
004670
YD=YDA
004680
ALDP=ALDAP
004690
ANDP=ANDAP
004700
J1=0
004710
IF(YD,NE.0.0)GO TO 1003
004720
IF(ALDP,NE.0.0)GO TO 1003
004730
IF(ANDP,NE.0.0)GO TO 1003
004740
WRITE(6,1004)RUN
004750
1004 FORMAT(1H1,5X8HRUN NO. A3,/1H0,10X,
004760
1 60HTHE AILERON NUMERATOR ROOTS AND CHARACTERISTICS ARE ZERO.
004770
GO TO 113
004780
1003 WRITE(6,14)RUN
004790
14 FORMAT(1H1,2X8HRUN NO. A3,5X23HAILERON NUMERATOR ROOTS)
004800
C
004810
SIDESLIP TO CONTROL DEFLECTION NUMERATOR
004820
92 WRITE (6,302)
004830
302 FORMAT(1H0,15X30HSIDESLIP TO CONTROL DEFLECTION)
004840
DO 330 I1=1,5
004850
ROOTR(I1) = 0.0
004860
330 ROOTI(I1) = 0.0
004870
AB=YD/U
004880
BB=-A9*(ALPP+ANRP)+ANDP*((YR/U)-1.0)+ALDP*(YP/U)
004890
CB=AB*(ALPP*ANRP-ALRP*ANPP)+ANDP*((YP/U)*ALRP-(YR/U)
004900

```


LATERAL-DIRECTIONAL PROGRAM LISTING

```

1 *ALPP*ALPP*(GSG/U)+ALDP*((YR/U)*ANPP-(YP/U)*ANRP-ANPP      004900
2 *(GCG/U)                                                       004910
DB=(GCG/U)*(ALRP*ANDP-ANRP*ALDP)+(GSG/U)*(ALDP*ANPP-      004920
1 ALPP*ANDP)                                                    004930
B(1)=AB                                                         004940
B(2)=BB                                                         004950
B(3)=CB                                                         004960
B(4)=OB                                                         004970
IF(B(1))62,63,62                                              004980
63 N=2                                                           004990
B(1)=B(2)                                                       005000
B(2)=B(3)                                                       005010
B(3)=B(4)                                                       005020
GO TO 84                                                         005030
62 N=3                                                           005040
84 CALL DMULR (B,N,ROOTR,ROOTI)                                005050
M=2                                                             005060
GO TO 66                                                         005070
67 IF(N-2)64,65,64                                             005080
65 IF(1.E-2-ABS(ROOTI(1)))41,42,42                             005090
41 WB=SQRT(ROOTR(1)**2+ROOTI(1)**2)                            005100
ZB=-ROOTR(1)/WB                                               005110
WRITE (6,30)ZB,WB                                             005120
30 FORMAT(1H0,7X,4HZB =E14.6,7X,4HWB =E14.6)                 005130
GO TO 81                                                         005140
42 ROOTR(1)=-ROOTR(1)                                          005150
ROOTR(2)=-ROOTR(2)                                          005160
WRITE (6,29)ROOTR(1),ROOTR(2)                                005170
29 FORMAT(1H0,4X,7H1/TB1 =E14.6,4X,7H1/TB2 =E14.6)          005180
GO TO 81                                                         005190
64 IF(1.E-2-ABS(ROOTI(1)))43,44,44                             005200
43 WB1=SQRT(ROOTR(1)**2+ROOTI(1)**2)                          005210
ZB1=-ROOTR(1)/WB1                                           005220
ROOTR(3)=-ROOTR(3)                                          005230
WRITE (6,152)ZB1,WB1,ROOTR(3)                                005240
152 FORMAT(1H0,7X,5HZB =E14.6,5X,5HWB =E14.6,5X,7H1/TB1 =E14.6) 005250
GO TO 81                                                         005260
44 IF(1.E-2-ABS(ROOTI(2)))45,46,46                             005270
45 WB2=SQRT(ROOTR(2)**2+ROOTI(2)**2)                          005280
ZB2=-ROOTR(2)/WB2                                           005290
ROOTR(1)=-ROOTR(1)                                          005300
WRITE (6,151)ROOTR(1),ZB2,WB2                                005310
151 FORMAT(1H0,7X,7H1/TB =E14.6,5X,5HZB =E14.6,5X,5HWB =E14.6) 005320
GO TO 81                                                         005330
46 DO 47 I=1,3                                                 005340
47 ROOTR(I)=-ROOTR(I)                                         005350
WRITE (6,150)(ROOTR(I),I=1,3)                                005360
150 FORMAT(1H0,5X,7H1/TB1 =E14.6,5X,7H1/TB2 =E14.6,5X,7H1/TB3 =E14.6) 005370
81 WRITE (6,303)AB,BB,CB,OB                                   005380
303 FORMAT(1H0,3X4HAB =E12.4,3X4HBB =E12.4,3X4HCB =E12.4,   005390
1 3X4HOB =E12.4)                                             005400
C ROLL TO CONTROL DEFLECTION NUMERATOR                       005410
WRITE (6,304)                                                  005420
304 FORMAT(1H0,15X32HROLL ANGLE TO CONTROL DEFLECTION)      005430
DO 331 I1=1,5                                                 005440
ROOTR(I1) = 0.0                                              005450
331 ROOTI(I1) = 0.0                                           005460
AP=(YD/U)*ALBDP+ALDP*(1.0-YVD)                                005470
BP=(YD/U)*(ALBP-ANRP*ALBDP+ALRP*ANBDP)+ANDP*(ALRP-      005480
1 ALBDP*(1.0-(YR/U))-ALRP*YVD)+ALDP*(-ANRP-YV+ANRP*YVD)    005490

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

2 *ANBDP*(1.0-(YR/U))                                005500
CP=(YD/U)*(ALRP*ANBP-ANRP*ALBP)+ANDP*(-ALRP*YV-      005510
1 ALBP*(1.0-(YR/U))+ALBDP*(GSG/U))+ALDP*(ANRP*YV+ANBP* 005520
2 (1.0-(YR/U))-ANBDP*(GSG/U))                        005530
DP=(GSG/U)*(ANDP*ALBP-ALDP*ANBP)                    005540
P(1)=AP                                               005550
P(2)=BP                                               005560
P(3)=CP                                               005570
P(4)=DP                                               005580
IF(P(1))68,69,68                                     005590
69 N=2                                                 005600
P(1)=P(2)                                             005610
P(2)=P(3)                                             005620
P(3)=P(4)                                             005630
GO TO 125                                             005640
68 N=3                                                 005650
125 CALL DMULR (P,N,ROOTR,ROOTI)                      005660
M=3                                                    005670
GO TO 66                                             005680
72 IF(N-2)70,71,70                                    005690
71 IF(1.E-2-ABS(ROOTI(1)))48,49,49                  005700
48 WP=SQRT(ROOTR(1)**2+ROOTI(1)**2)                 005710
ZP=-ROOTR(1)/WP                                       005720
W=WP/WD                                               005730
WRITE (6,32) ZP,WP,W                                  005740
32 FORMAT(1H0,7X,4HZP =E14.6,7X,4HWP =E14.6,5X,*WPHI/WDR =*E14.6) 005750
GO TO 82                                             005760
49 ROOTR(1)=-ROOTR(1)                                 005770
ROOTR(2)=-ROOTR(2)                                 005780
WRITE (6,31)ROOTR(1),ROOTR(2)                       005790
31 FORMAT(1H0,4X,7H1/TP1 =E14.6,4X,7H1/TP2 =E14.6) 005800
GO TO 82                                             005810
70 IF(1.E-2-ABS(ROOTI(1)))50,51,51                  005820
50 WP=SQRT(ROOTR(1)**2+ROOTI(1)**2)                 005830
ZP=-ROOTR(1)/WP                                       005840
W=WP/WD                                               005850
ROOTR(3)=-ROOTR(3)                                 005860
WRITE (6,85)ZP,WP,ROOTR(3),W                         005870
85 FORMAT(1H0,4X4HZP =E14.6,7X4HWP =E14.6,7X7H1/TP =E14.6, 005880
1 5X10HWPHI/WDR =E14.6)                             005890
GO TO 82                                             005900
51 IF(1.E-2-ABS(ROOTI(2)))52,53,53                  005910
52 WP=SQRT(ROOTR(2)**2+ROOTI(2)**2)                 005920
ZP=-ROOTR(2)/WP                                       005930
W=WP/WD                                               005940
ROOTR(1)=-ROOTR(1)                                 005950
WRITE (6,25)ROOTR(1),ZP,WP,W                         005960
25 FORMAT(1H0,4X,7H1/TP1 =E14.6,7X,4HZP =E14.6,7X,4HWP =E14.6, 005970
1 5X10HWPHI/WDR =E14.6)                             005980
GO TO 82                                             005990
53 DO 40 I=1,3                                       006000
40 ROOTR(I)=-ROOTR(I)                                006010
WRITE (6,26) (ROOTR(I),I=1,3)                       006020
26 FORMAT(1H0,4X,7H1/TP1 =E14.6,4X,7H1/TP2 =E14.6,4X,7H1/TP3 =E14.6) 006030
82 WRITE (6,305) AP, BP, CP, DP                      006040
305 FORMAT(1H0,3X4HAP =E12.4,3X4HBP =E12.4,3X4HCP =E12.4, 006050
1 3X4HDP =E12.4)                                    006060
C YAW RATE TO CONTROL DEFLECTION NUMERATOR          006070
WRITE (6,306)                                         006080
306 FORMAT(1H0,15X3CHYAW RATE TO CONTROL DEFLECTION) 006090

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

DO 332 I1=1,5                                006100
ROOTR(I1) = 0.0                               006110
332 ROOTI(I1) = 0.0                            006120
AR=(YD/U)*ANBDP+ANDP*(1.0-YVD)                006130
BR=(YD/U)*(ANBP-ALPP*ANBDP+ANPP*ALBDP)        006140
1 +ANDP*(-YV-ALPP*(1.0-YVD)-ALBDP*(YP/U))     006150
2 +ALDP*(ANPP+ANBDP*(YP/U)-ANPP*YVD)          006160
CR=(YD/U)*(ALBP*ANPP-ANBP*ALPP)              006170
1 +ANDP*(YV*ALPP-ALBP*(YP/U)-ALBDP*(GCG/U))  006180
2 +ALDP*(ANBP*(YP/U)-ANPP*YV+ANBDP*(GCG/U))  006190
DR=(GCG/U)*(ALDP*ANBP-ANDP*ALBP)             006200
RA(1)=AR                                       006210
RA(2)=BR                                       006220
RA(3)=CR                                       006230
RA(4)=DR                                       006240
IF(RA(1))74,75,74                             006250
75 N=2                                         006260
RA(1)=RA(2)                                    006270
RA(2)=RA(3)                                    006280
RA(3)=RA(4)                                    006290
GO TO 126                                       006300
74 N=3                                         006310
126 CALL DMULR(RA,N,ROOTRD,ROOTID)            006320
M=4                                             006330
GO TO 66                                       006340
73 IF(N-2)76,77,76                             006350
77 IF(1.E-2-ABS(ROOTI(1)))55,56,56            006360
55 WR=SQRT(ROOTR(1)**2+ROOTI(1)**2)          006370
ZR=-ROOTR(1)/WR                               006380
WRITE(6,34) ZR,WR                              006390
34 FORMAT(1H0,7X4HZR =E14.6,7X4HWR =E14.6)    006400
GO TO 83                                       006410
56 ROOTR(1)=-ROOTR(1)                          006420
ROOTR(2)=-ROOTR(2)                          006430
WRITE(6,33) ROOTR(1),ROOTR(2)                006440
33 FORMAT(1H0,4X,7H1/TR1 =E14.6,4X,7H1/TR2 =E14.6) 006450
GO TO 83                                       006460
76 IF(1.E-2-ABS(ROOTI(1)))57,58,58            006470
57 WR=SQRT(ROOTR(1)**2+ROOTI(1)**2)          006480
ZR=-ROOTR(1)/WR                               006490
ROOTR(3)=-ROOTR(3)                          006500
WRITE(6,86) ZR,WR,ROOTR(3)                  006510
86 FORMAT(1H0,4X4HZR =E14.6,7X4HWR =E14.6,7X7H1/TR =E14.6) 006520
GO TO 83                                       006530
58 IF(1.E-2-ABS(ROOTI(2)))78,79,79            006540
78 WR=SQRT(ROOTR(2)**2+ROOTI(2)**2)          006550
ZR=-ROOTR(2)/WR                               006560
ROOTR(1)=-ROOTR(1)                          006570
WRITE(6,27) ROOTR(1),ZR,WR                  006580
27 FORMAT(1H0,4X,7H1/TR =E14.6,7X4HZR =E14.6,7X4HWR =E14.6) 006590
GO TO 83                                       006600
79 DO 88 I=1,3                                006610
88 ROOTR(I)=-ROOTR(I)                        006620
WRITE(6,28) ROOTR(1),ROOTR(2),ROOTR(3)      006630
28 FORMAT(1H0,4X,7H1/TR1 =E14.6,4X,7H1/TR2 =E14.6,4X,7H1/TR3 =E14.6) 006640
83 WRITE(6,307) AR,BR,CR,DR                  006650
307 FCRMAT(1H0,3X4HAR =E13.5,2X4HBR =E13.5,2X4HCR =E13.5, 006660
1 2X4HDR =E13.5)                               006670
IF(ABS(ALX).LT..001) GO TO 1005              006680
C ACCELERATION A Y PRIME TO CONTROL DEFLECTION NUMERATOR 006690

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

      JXY = 0                                006700
      WRITE(6,308)                           006710
308  FORMAT(1H0,15X*ACCLEROMETER SENSED SIDE ACCELERATION TO CONTROL DE 006720
      1FLECTION*)                             006730
      AAYP =AB*U+AR*ALX                       006740
      BAYP =BB*U+BR*ALX+U*AR                 006750
      CAYP =CB*U+CR*ALX+U*BR-GCG*AP-GSG*AR  006760
      DAYP =DB*U+DR*ALX+U*CR-GCG*BP-GSG*BR  006770
      EAYP =U*DR-GCG*CP-GSG*CR              006780
311  AYP(1)=AAYP                             006790
      AYP(2)=BAYP                             006800
      AYP(3)=CAYP                             006810
      AYP(4)=DAYP                             006820
      AYP(5)=EAYP                             006830
      DO 333 I1=1,5                          006840
      ROOTR(I1) =0.0                         006850
333  ROOTI(I1)=0.0                           006860
      IF(AYP(1))111,132,111                 006870
132  AYP(1)=AYP(2)                           006880
      AYP(2)=AYP(3)                           006890
      AYP(3)=AYP(4)                           006900
      AYP(4)=AYP(5)                           006910
      IF(AYP(1))121,122,121                 006920
121  N=3                                       006930
      GO TO 127                               006940
122  AYP(1)=AYP(2)                           006950
      AYP(2)=AYP(3)                           006960
      AYP(3)=AYP(4)                           006970
      N=2                                       006980
      GO TO 127                               006990
      111 N=4                                  007000
127  CALL DMULR(AYP,N,ROOTRD,ROOTID)         007010
      L=1                                       007020
      M=5                                       007030
      GO TO 66                               007040
80   IF(N-4)123,134,133                     007050
133  IF(1.E-2-ABS(ROOTI(1)))101,102,102     007060
102  IF(1.E-2-ABS(ROOTI(2)))103,104,104     007070
103  W1=SQRT(ROOTR(2)**2+ROOTI(2)**2)       007080
      Z1=-ROOTR(2)/W1                         007090
      GO TO (128,128,128,128,129),L         007100
128  IF(1.E-2-ABS(ROOTI(4)))105,106,106     007110
105  W2=SQRT(ROOTR(4)**2+ROOTI(4)**2)       007120
      Z2=-ROOTR(4)/W2                         007130
      ROOTR(1)=-ROOTR(1)                     007140
      WRITE(6,35)Z1,W1,Z2,W2,ROOTR(1)       007150
35   FORMAT(1H0,1X,6HZAY1 =E12.4,5X,6HWAY1 =E12.4,5X,6HZAY2 =E12.4,5X,
      1 6HWAY2 =E12.4,3X,8H1/TAY =E12.4)    007160
      GO TO 87                               007170
106  GO TO (15,16,17),L                     007180
15   DO 97 I=1,5                            007190
97   ROOTR(I)=-ROOTR(I)                     007200
      WRITE(6,93)ROOTR(1),Z1,W1,ROOTR(4),ROOTR(5) 007210
93   FORMAT(1H0,1X8H1/TAY1 =E12.4,5X6HZAY =E12.4,5X6HWAY =E12.4,
      1 5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4) 007220
      GO TO 87                               007230
104  GO TO (163,163,163,163,139),L         007240
163  IF(1.E-2-ABS(ROOTI(3)))107,108,108     007250
107  W3=SQRT(ROOTR(3)**2+ROOTI(3)**2)       007260
      Z3=-ROOTR(3)/W3                         007270

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

GO TO (155,156,154,153),L
155 DO 98 I=1,5 007300
98 ROOTR(I)=-ROOTR(I) 007310
WRITE (6,59) ROOTR(1),ROOTR(2),Z3,W3,ROOTR(5) 007320
59 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X6HZAY3 =E12.4, 007330
1 5X6HWAY3 =E12.4,5X8H1/TAY5 =E12.4) 007340
GO TO 87 007350
108 IF(1.E-2-ABS(ROOTI(4)))135,136,136 007360
135 W2=SQRT(ROOTR(4)**2+ROOTI(4)**2) 007370
Z2=ABS(ROOTR(4))/W2 007380
GO TO (157,158,16,16),L 007390
157 DO 99 I=1,3 007400
99 ROOTR(I)=-ROOTR(I) 007410
WRITE (6,60) ROOTR(1),ROOTR(2),ROOTR(3),Z2,W2 007420
60 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4, 007430
1 5X6HZAY =E12.4,5X6HWAY =E12.4) 007440
GO TO 87 007450
136 GO TO (159,160,161,162),L 007460
159 DO 100 I=1,N 007470
100 ROOTR(I)=-ROOTR(I) 007480
WRITE (6,37) (ROOTR(I),I=1,N) 007490
37 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4, 007500
1 5X8H1/TAY4 =E12.4,5X8H1/TAY5 =E12.4) 007510
GO TO 87 007520
101 W4=SQRT(ROOTR(1)**2+ROOTI(1)**2) 007530
Z4=-ROOTR(1)/W4 007540
GO TO (141,141,141,141,147),L 007550
141 N=N-3 007560
GO TO (23,24),N 007570
24 L=2 007580
GO TO 104 007590
23 L=4 007600
GO TO 104 007610
156 ROOTR(5)=-ROOTR(5) 007620
WRITE (6,36) Z4,W4,Z3,W3,ROOTR(5) 007630
36 FORMAT(1H0,1X,6HZAY1 =E12.4,5X,6HWAY1 =E12.4,5X,6HZAY2 =E12.4,5X, 007640
16HWAY2 =E12.4,3X,8H1/TAY1 =E12.4) 007650
GO TO 87 007660
158 ROOTR(3)=-ROOTR(3) 007670
WRITE (6,36) Z4,W4,Z2,W2,ROOTR(3) 007680
GO TO 87 007690
160 WRITE (6,59) ROOTR(3),ROOTR(4),Z4,W4,ROOTR(5) 007700
GO TO 87 007710
134 L=3 007720
GO TO 133 007730
17 ROOTR(1)=-ROOTR(1) 007740
ROOTR(4)=-ROOTR(4) 007750
WRITE (6,18) ROOTR(1),Z1,W1,ROOTR(4) 007760
18 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X6HZAY =E12.4,5X6HWAY =E12.4,5X 007770
18H1/TAY2 =E12.4) 007780
GO TO 87 007790
16 WRITE (6,19) 007800
19 FORMAT(1H0,1X*IF YOU GET TO THIS STATEMENT, YOU HAVE A SERIOUS* 007810
1 * PROGRAMMING OR LOGIC ERROR*) 007820
GO TO 87 007830
154 ROOTR(1)=-ROOTR(1) 007840
ROOTR(2)=-ROOTR(2) 007850
WRITE (6,20) ROOTR(1),ROOTR(2),Z3,W3 007860
20 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X6HZAY =E12.4, 007870
15X6HWAY =E12.4) 007880

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

GO TO 87
161 DO 179 I=1,N
179 ROOTR(I)=-ROOTR(I)
WRITE (6,22) (ROOTR(I),I=1,N)
22 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4,
15X8H1/TAY4 =E12.4)
GO TO 87
153 WRITE (6,54) Z4,W4,Z3,W3
54 FORMAT(1H0,1X,6HZAY4 =E12.4,5X,6HWAY4 =E12.4,5X,6HZAY3 =E12.4,5X6H
1WAY3 =E12.4)
GO TO 87
162 ROOTR(3)=-ROOTR(3)
ROOTR(4)=-ROOTR(4)
WRITE (6,61) Z4,W4,ROOTR(3),ROOTR(4)
61 FORMAT(1H0,1X6HZAY =E12.4,5X6HWAY =E12.4,5X8H1/TAY1 =E12.4,5X,8H
11/TAY2 =E12.4)
GO TO 87
123 L=5
GO TO 133
129 ROOTR(1)=-ROOTR(1)
WRITE (6,138) ROOTR(1),Z1,W1
138 FORMAT(1H0,2X,7H1/TAY =E14.6,5X5HZAY =E14.6,5X5HWAY =E14.6)
GO TO 87
139 DO 137 I=1,3
137 ROOTR(I)=-ROOTR(I)
WRITE (6,140) ROOTR(1),ROOTR(2),ROOTR(3)
140 FORMAT(1H0,2X8H1/TAY1 =E14.6,5X8H1/TAY2 =E14.6,5X8H1/TAY3 =E14.6)
GO TO 87
147 ROOTR(3)=-ROOTR(3)
WRITE (6,148) Z4,W4,ROOTR(3)
148 FORMAT(1H0,2X5HZAY =E14.6,5X5HWAY =E14.6,5X7H1/TAY =E14.6)
87 WRITE (6,309) AAYP,BAYP,CAYP,DAYP,EAYP
309 FORMAT(1H0,1X6HAYP =E13.5,2X6HBAYP =E13.5,2X6HCAYP =E13.5
1 /2X6HDAYP =E13.5,2X6HEAYP =E13.5)
IF (JXY.EQ.1) GO TO 1005
WRITE (6,310)
310 FORMAT(1H0,15X*INERTIAL SIDE ACCLERATION TO CONTROL DEFLECTION*)
AAYP =AB*U+AR*ALX
BAYP =BB*U+BR*ALX+U*AR
CAYP =CB*U+CR*ALX+U*BR
DAYP =DB*U+DR*ALX+U*CR
EAYP =U*DR
JXY=1
GO TO 311
1005 IF (IABS(IOPT).NE.2) GO TO 113
C
C OPTION 2
C
C CALL AOPT(J1)
C PREVIOUSLY CALCULATED - CON,CONA,COM,ANUM,ADEN,DTR
C
C RUDDER
C
113 IF (J1.EQ.1.AND.JJXX.EQ.1) GO TO 236
IF (J1.EQ.1) GO TO 250
YD=YDR
ALDP=ALDRP
ANDP=ANDRP
IF (YD) 205,206,205
206 IF (ALDP) 205,207,205
207 IF (ANDP) 205,208,205

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

205 J1=1                                008500
    WRITE (6,38) RUN                    008510
38  FORMAT(1H1,2X,8HRUN NO. A3,5X22HRUDDER NUMERATOR ROOTS) 008520
    GO TO 92                            008530
208 WRITE (6,209) RUN                  008540
209 FORMAT(1H1,5X8HRUN NO. A3,/1H0,10X60HTHE RUDDER NUMERATOR ROOTS AND 008550
    10 CHARACTERISTICS ARE ZERO. )
    GO TO 250                            008560
230 WRITE(6,231)RUN                    008570
231 FORMAT(1H1,5X8HRUN NO. A3,5X*COUPLING NUMERATOR ROOTS*) 008590
    JJXX=0                               008600
    DO 232 I1=1,5                       008610
    ROOTR(I1)=0.                        008620
232 ROOTI(I1)=0.                       008630
    WRITE(6,233)                         008640
233 FORMAT(1H-,15X*PHI TO AILERON, BETA TO RUDDER*) 008650
    ALNLN=ALDRP*ANDAP-ALDAP*ANDRP      008660
    YNYN=(YDR*ANDAP-YDA*ANDRP)/U      008670
    YLYL=(YDR*ALDAP-YDA*ALDRP)/U      008680
    APB(1)=YLYL                         008690
    APB(2)=ALNLN*(1.-YR/U)+YNYN*ALRP-YLYL*ANRP 008700
    APB(3)=-GSG*ALNLN/U                008710
    IF(APB(3).EQ.0.)GO TO 234          008720
    N=2                                  008730
    CALL DMULR(APB,N,ROOTR,ROOTI)      008740
    MM=1                                 008750
    GO TO 9                              008760
8  IF(ABS(ROOTI(1)).LT..0001)GO TO 236 008770
    WPB=SQRT(ROOTR(1)**2+ROOTI(1)**2)  008780
    ZPB=-ROOTR(1)/WPB                  008790
    WRITE(6,235)ZPB,WPB                008800
235 FORMAT(1H0,3X5HZPB =E14.6,5X5HWPB =E14.6) 008810
    GO TO 238                            008820
234 ROOTR(1)=APB(2)/APB(1)             008830
    IF(APB(2).EQ.0.D0.OR.APB(3).EQ.0.D0) ROOTR(1)=0. 008840
    WRITE(6,237)ROOTR(1)               008850
237 FORMAT(1H0,4X7H1/TPB =E14.6 )    008860
    GO TO 238                            008870
236 ROOTR(1)=-ROOTR(1)                 008880
    ROOTR(2)=-ROOTR(2)                 008890
    WRITE(6,239)ROOTR(1),ROOTR(2)      008900
239 FORMAT(1H0,3X*1/TPB1 =*E14.6,5X*1/TPB2 =*E14.6) 008910
238 WRITE(6,240)APB(1),APB(2),APB(3)  008920
240 FORMAT(1H0,3X*APB =*D14.6,5X*BPB =*D14.6,5X*CPB =*D14.6) 008930
    DO 241 I1=1,5                       008940
    ROOTR(I1)=0.                       008950
241 ROOTI(I1)=0.                      008960
C   PHI TO AILERON, PSI TO RUDDER     008970
    WRITE(6,242)                         008980
242 FORMAT(1H-,15X*PHI TO AILERON, PSI TO RUDDER*) 008990
    APP=ALNLN*(YVD-1.)-YNYN*ALBOP+YLYL*ANBOP 009000
    BPP=ALNLN*YV-YNYN*ALBP+YLYL*ANBP   009010
    ROT=BPP/APP                         009020
    IF(APP.EQ.0..OR.BPP.EQ.0.) ROT=0.  009030
    WRITE(6,243)ROT                     009040
243 FORMAT(1H0,4X7H1/TPP =E14.6)      009050
    WRITE(6,244)APP,BPP                 009060
244 FORMAT(1H0,3X*APP =*E14.6,5X*BPB =*E14.6///15X, 009070
    1 *PSI TO AILERON, BETA TO RUDDER*) 009080
C   PSI TO AILERON, BETA TO RUDDER   009090

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

      APSB(1)=YNYN                                009100
      APSB(2)=ALNLN*YP/U-YNYN*ALPP+YLYL*ANPP      009110
      APSB(3)=GCG*ALNLN/U                          009120
      N=2                                            009130
      CALL DMULR(APSBN,N,ROOTRD,ROOTID)            009140
      MM=2                                          009150
      GO TO 9                                       009160
6     IF(ABS(ROOTI(1)).LT..0001)GO TO 246          009170
      WPSB =SQRT(ROOTR(1)**2+ROOTI(1)**2)          009180
      ZPSB =-ROOTR(1)/WPSB                         009190
      WRITE(6,247)ZPSB,WPSB                        009200
247  FORMAT(1H0,3X6HZPSB =E14.6,4X6HWPSB =E14.6)  009210
      GO TO 248                                     009220
248  ROOTR(1)=-ROOTR(1)                            009230
      ROOTR(2)=-ROOTR(2)                            009240
      WRITE(6,249)ROOTR(1),ROOTR(2)               009250
249  FORMAT(1H0,3X*1/TPSB1 =*E14.6,5X*1/TPSB2 =*E14.6)  009260
248  WRITE(6,251)APSBN(1),APSBN(2),APSBN(3)       009270
251  FORMAT(1H0,3X*APSBN =*D14.6,5X*BPSB =*D14.6,5X*CPSB =*D14.6///15X, 009280
      1*PHI TO AILERON, ACCELERATION TO RUDDER*)  009290
      DO 252 I=1,5                                  009300
      ROOTR(I)=0.                                   009310
252  ROOTI(I)=0.                                    009320
      APAY(1)=U*APB(1)+ALX*APP                     009330
      APAY(2)=U*APB(2)+ALX*BPP+U*APP              009340
      APAY(3)=U*APB(3)+U*BPP-GSG*APP              009350
      APAY(4)=-GSG*BPP                             009360
      N=3                                            009370
      IF(APAY(4).EQ.000)N=2                         009380
      IF(APAY(1).NE.0.00)GO TO 254                 009390
      APAY(1)=APAY(2)                              009400
      APAY(2)=APAY(3)                              009410
      APAY(3)=APAY(4)                              009420
      IF(APAY(4).EQ.0.00)GO TO 255                 009430
      N=2                                            009440
254  CALL DMULR(APAY,N,ROOTRD,ROOTID)             009450
      MM=3                                          009460
      GO TO 9                                       009470
5     IF(ABS(ROOTI(1)).LT..0001)GO TO 257          009480
      WPAY =SQRT(ROOTI(1)**2+ROOTR(1)**2)          009490
      ZPAY =-ROOTR(1)/WPAY                         009500
      ROOTR(3)=-ROOTR(3)                           009510
      IF(N.EQ.2)ROOTR(3)=0.0                       009520
      WRITE(6,258)ZPAY,WPAY,ROOTR(3)               009530
258  FORMAT(1H0,3X*ZPAY =*E14.6,5X*WPAY =*E14.6,5X*1/TPAY =*E14.6)  009540
      GO TO 260                                     009550
257  IF(ABS(ROOTI(2)).LT..0001)GO TO 259          009560
      WPAY=SQRT(ROOTR(2)**2+ROOTI(2)**2)          009570
      ZPAY=-ROOTR(2)/WPAY                          009580
      ROOTR(1)=-ROOTR(1)                            009590
      WRITE(6,258)ZPAY,WPAY,ROOTR(1)               009600
      GO TO 260                                     009610
259  ROOTR(1)=-ROOTR(1)                            009620
      ROOTR(2)=-ROOTR(2)                            009630
      ROOTR(3)=-ROOTR(3)                            009640
      IF(N.EQ.2)ROOTR(3)=0.0                       009650
      WRITE(6,261)(ROOTR(I),I=1,3)                 009660
261  FORMAT(1H0,3X*1/TPAY1 =*E14.6,5X*1/TPAY2 =*E14.6,5X  009670
      1*1/TPAY3 =*E14.6)                           009680
      GO TO 260                                     009690

```


LATERAL-DIRECTIONAL PROGRAM LISTING

```

255 ROT=APAY(3)/APAY(2)                                009700
    IF(APAY(3).EQ.0.D0.OR.APAY(2).EQ.0.D0) ROT=0.      009710
    WRITE(6,261)ROT,RZERO,RZERO                        009720
260 WRITE(6,262)(APAY(I),I=1,4)                       009730
262 FORMAT(1H0,3X*APAY=*D14.6,5X*BPAY=*D14.6,5X*CPAY=*D14.6,
    15X*DPAY=*D14.6///15X*PSI TO AILERON, ACCELERATION*
    2* TO RUDDER*)                                     009740
    DO 263 I1=1,5                                       009750
    ROOTR(I1)=0.0                                       009760
263 ROOTI(I1)=0.0                                       009770
    APAY(1)=U*APSB(1)                                    009780
    APAY(2)=U*APSB(2)                                    009790
    APAY(3)=U*APSB(3)+GCG*APP                           009800
    APAY(4)=GCG*BPP                                      009810
    N=3                                                  009820
    IF(APAY(4).EQ.0.D0)N=2                              009830
    IF(APAY(1).NE.0.D0)GO TO 264                        009840
    APAY(1)=APAY(2)                                     009850
    APAY(2)=APAY(3)                                     009860
    APAY(3)=APAY(4)                                     009870
    IF(APAY(4).EQ.0.D0)GO TO 265                       009880
    N=2                                                  009890
264 CALL DMULR(APAY,N,ROOTR,ROOTI)                    009900
    MM=4                                                 009910
    GO TO 9                                              009920
4   IF(ABS(ROOTI(1)).LT..0001)GO TO 267               009930
    WSAY = SQRT(ROOTR(1)**2+ROOTI(1)**2)                009940
    ZSAY=-ROOTR(1)/WPSAY                                009950
    ROOTR(3)=-ROOTR(3)                                  009963
    IF(N.EQ.2)ROOTR(3)=0.0                              009970
    WRITE(6,268)ZSAY,WSAY,ROOTR(3)                    009980
    GO TO 270                                           009990
268 FORMAT(1H0,3X*ZPSAY=*E14.6,5X*WPSAY=*E14.6,5X*1/TPSAY=*E14.6)
    010000
267 IF(ABS(ROOTI(2)).LT..0001)GO TO 269               010010
    WSAY=SQRT(ROOTR(2)**2+ROOTI(2)**2)                 010020
    ZSAY=-ROOTR(2)/WSAY                                010030
    ROOTR(3)=-ROOTR(3)                                  010040
    WRITE(6,268)ZSAY,WSAY,ROOTR(3)                    010050
    GO TO 270                                           010060
269 ROOTR(1)=-ROOTR(1)                                  010070
    ROOTR(2)=-ROOTR(2)                                  010080
    ROOTR(3)=-ROOTR(3)                                  010090
    IF(N.EQ.2)ROOTR(3)=0.0                              010100
    WRITE(6,271)(ROOTR(I),I=1,3)                       010110
271 FORMAT(1H0,3X*1/TPSAY1=*E14.6,5X*1/TPSAY2=*E14.6,5X
    1*1/TPSAY3=*E14.6)                                  010120
    GO TO 270                                           010130
265 ROT=APAY(3)/APAY(2)                                010140
    IF(APAY(3).EQ.0.D0.OR.APAY(2).EQ.0.D0) ROT=0.      010150
    WRITE(6,271)ROT,RZERO,RZERO                        010160
270 WRITE(6,272)(APAY(I),I=1,4)                       010170
272 FORMAT(1H0,3X*APAY=*D14.6,5X*BPSAY=*D14.6,5X*CPAY=*D14.6,5X,
    1*DPAY=*D14.6///15X*ACCELERATION TO AILERON,*
    2* BETA TO RUDDER*)                                010180
    DO273 I1=1,5                                       010190
    ROOTR(I1)=0.0                                       010200
273 ROOTI(I1)=0.0                                       010210
    APAY(1)=ALX*APSB(1)                                   010220
    APAY(2)=ALX*APSB(2)+U*APSB(1)                     010230
    APAY(3)=ALX*APSB(3)+U*APSB(2)+GSG*APSB(1)+GCG*APB(1)
    010240
    010250
    010260
    010270
    010280
    010290

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

APAY(4)=U*APSB(3)+GSG*APSB(2)+GGG*APB(2) 010300
N=3 010310
IF (APAY(4).EQ.0.00)N=2 010320
IF (APAY(1).NE.0.00)GO TO 274 010330
APAY(1)=APAY(2) 010340
APAY(2)=APAY(3) 010350
APAY(3)=APAY(4) 010360
IF (APAY(4).EQ.0.00)GO TO 275 010370
N=2 010380
274 CALL DMULR (APAY,N,ROOTRD,ROOTID) 010390
MM=5 010400
GO TO 9 010410
3 IF (ABS (ROOTI(1)).LT..0001)GO TO 277 010420
WAYB=SQRT (ROOTR(1)**2+ROOTI(1)**2) 010430
ZAYB=-ROOTR(1)/WAYB 010440
ROOTR(3)=-ROOTR(3) 010450
IF (N.EQ.2)ROOTR(3)=0.0 010460
WRITE (6,278) ZAYB,WAYB,ROOTR(3) 010470
278 FORMAT (1H0,3X*ZAYB =*E14.6,5X*WAYB =*E14.6,5X*1/TAYB =*E14.6) 010480
GO TO 280 010490
277 IF (ABS (ROOTI(2)).LT..0001)GO TO 279 010500
WAYB =SQRT (ROOTR(2)**2+ROOTI(2)**2) 010510
ZAYB =-ROOTR(2)/WAYB 010520
ROOTR(1)=-ROOTR(1) 010530
WRITE (6,278) ZAYB,WAYB,ROOTR(1) 010540
GO TO 280 010550
279 ROOTR(1)=-ROOTR(1) 010560
ROOTR(2)=-ROOTR(2) 010570
ROOTR(3)=-ROOTR(3) 010580
IF (N.EQ.2)ROOTR(3)=0 010590
WRITE (6,281) (ROOTR(I),I=1,3) 010600
281 FORMAT (1H0,3X,*1/TAYB1 =*E14.6,5X*1/TAYB2 =*E14.6,5X 010610
1*1/TAYB3 =*E14.6) 010620
GO TO 280 010630
275 ROT=APAY(3)/APAY(2) 010640
IF (APAY(3).EQ.0.00.OR.APAY(2).EQ.0.00) ROT=0. 010650
WRITE (6,281) ROT,RZERO,RZERO 010660
280 WRITE (6,282) (APAY(I),I=1,4) 010670
282 FORMAT (1H0,3X*AYB =*D14.6,5X*BYB =*D14.6,5X*CYB =*D14.6, 010680
15X*DAYB =*D14.6) 010690
GO TO 250 010700
9 DO 7 I1=1,5 010710
ROOTI(I1)=ROOTID(I1) 010720
7 ROOTR(I1)=ROOTRD(I1) 010730
GO TO (8,6,5,4,3)MM 010740
END 010750
SUBROUTINE CHNG(J) 010760
COMMON/BB/RHO,U,S,GWT,SPAN,IXB,G,ALFI,GAMA,LX,CYB,CYBD,CYP,CYR, 010770
A CYDA,CYDR,CLB,CLBD,CLP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA,CNDR, 010780
B ALFA,ALFX,PLT,YB,YBD,YP,YR,YDA, LB,LBD,LP,LR,LDA,LDR,NB,NBD, 010790
C NP,NR,NDA,NDR,LBP,LBDP,LPP,LRP,LDAP,LDRP,NBP,NBDP,NPP,NRP,NDAP, 010800
D NDRP,IZB,IXZB,YDR 010810
REAL IXB,LX,LB,LBD,LP,LR,LDA,LDR,NB,NBD,NP,NR,NDA,NDR,LBP,LBDP, 010820
A LPP,LRP,LDAP,LDRP,NBP,NBDP,NPP,NRP,NDAP,NDRP,IZB,IXZB 010830
NAMELIST/CHANGE/RHO,U,S,GWT,SPAN,IXB,G,ALFI,GAMA,LX,CYB,CYBD,CYP, 010840
A CYR,CYDA,CYDR,CLB,CLBD,CLP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA, 010850
B CNDR,ALFA,ALFX,PLT,YB,YBD,YP,YR,YDA,YDR,LB,LBD,LP,LR,LDA,LDR, 010860
C NB,NBD,NP,NR,NDA,NDR,LBP,LBDP,LPP,LRP,LDAP,LDRP,NBP,NBDP,NPP, 010870
D NRP,NDAP,NDRP,IZB,IXZB,TEST 010880
PFAD(5,CHANGE) 010890

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

IF (TEST.EQ.1) WRITE(6,CHANGE)                                010900
TEST=0.                                                         010910
RETURN                                                         010920
END                                                             010930
SUBROUTINE PLTUP(DATA,T,P,PHI,B,N,PROG,RUN,WDS)                010940
DIMENSION DATA(438),T(122),P(122),PHI(122),B(122),PROG(11)  010950
DIMENSIONWDS(3)                                               010960
CALL SCALE(T,11.,N,1)                                         010970
CALL SCALE(P,8.,N,1)                                          010980
CALL SCALE(PHI,8.,N,1)                                        010990
CALL SCALE(B,8.,N,1)                                         011000
CALL PLOT( 0.,1.,-3)                                         011010
C                                                                011020
C          SET UP AXES                                       011030
C                                                                011040
CALL AXIS(0.,0.,19HROLL RATE - DEG/SEC,19,8.,90.,P(N+1),P(N+2)) 011050
CALL AXIS(-.5,0.,16HBANK ANGLE - DEG,16,8.,90.,PHI(N+1),PHI(N+2)) 011060
CALL AXIS(-1.,0.,20HSIDESLIP ANGLE - DEG,20,8.,90.,B(N+1),B(N+2)) 011080
CALL AXIS(0.,0.,14HTIME - SECONDS,-14,11.,0.,T(N+1),T(N+2))  011100
C                                                                011110
C          TITLE THE PLOT                                    011120
C                                                                011130
CALL SYMBOL (2.75,9.00,,2,16HTIME HISTORY FOR,0.,16)         011140
CALL SYMBOL (6.15,9.00,2,WDS,0.,18)                          011150
CALL SYMBOL (3.5,8.8,.1,PROG(1),0.,6)                        011160
DO 1 I=2,11                                                    011162
CALL SYMBOL (999.,8.8,.1,PROG(I),0.,6)                       011164
C                                                                011170
C          PLOT THE PLOT                                     011180
C                                                                011190
CALL LINE(T,P,N,1,N/4,1)                                       011200
CALL LINE(T,PHI,N,1,N/4,2)                                     011210
CALL LINE(T,B,N,1,N/4,5)                                       011220
C                                                                011230
C          IDENTIFY EACH PLOT                               011240
C                                                                011250
CALL SYMBOL (.2,8.,.1,1,0.,-1)                                011260
CALL SYMBOL (.3,8.,.1,1HP,0.,1)                               011270
CALL SYMBOL (.2,7.8.,.1,2,0.,-1)                              011280
CALL SYMBOL (.3,7.8.,.1,3PHI,0.,3)                           011290
CALL SYMBOL (.2,7.6.,.1,5,0.,-1)                              011300
CALL SYMBOL (.3,7.6.,.1,4HBETA,0.,4)                          011310
C                                                                011320
C          MOVE TO NEXT PLOT AND RETURN                     011330
C                                                                011340
CALL SYMBOL (11.35,7.00,.1,3HRUN,90.,3)                       011350
CALL SYMBOL (11.35,7.35,.1,RUN,90.,3)                         011360
CALL PLOT(11.5,-1.,3)                                         011370
CALL PLOT(11.5,9.,2)                                         011380
CALL PLOT(14.,-1.,-3)                                         011390
RETURN                                                         011400
END                                                             011410
SUBROUTINE AOPT(J1)                                           011420
COMMON /AA/ CON,CONA,COM,ANUM,ADEN,DTR,IROOT,TR,TS,ZD,WD,E,PER, 011430
1AP,BP,CP,DP,AB,BB,CB,DB,IOPT,A,TA,TB,TC,DATA,TITLE,PLT,IPLT
2,RUN,WDS                                                       011440
DIMENSION WORD1(3),WORD2(3)                                    011460
DIMENSION TM(3),PM(3),TIMEX(120),P3XX(120),PDAXX(120),B3XX(120) 011470
DIMENSION DATA (438),TITLE(21)                               011480
COMPLEX COM1,DEN,PNUM,BNUM,POBN                                011490

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

COMPLEX COM , ANUM , ADEN 011500
FUNP(X)=1./A*(AP*X**3+BP*X**2+CP*X+DP) 011510
FUNB(X)=1./A*(AB*X**3+BB*X**2+CB*X+DB) 011520
PT(T)=XKP+XKPR*EE**(XTR*T)+XKPS*EE**(XTS*T)+ 011530
1 XKPDR*EE**(CON*T)*COS(CONA*T+PSIP/DTR) 011540
BT(T)=XKB+XKBR*EE**(XTR*T)+XKBS*EE**(XTS*T)+ 011550
1 XKBDR*EE**(CON*T)*COS(CONA*T+PSIB/DTR) 011560
PDA(T)=XKP*T+XKPR*TR*(1.-EE**(XTR*T))+XKPS*TS* 011570
1 (1.-EE**(XTS*T))+CON2*(EE**(CON*T))*(CON*COS(CONA*T+PSIPR) 011580
2 +CONA*SIN(CONA*T+PSIPR))+CON3) 011590
DATA (WORD1(I),I=1,3)/21HAILERON STEP INPUT /,(WORD2(I),I=1,3) 011600
4/21HRUDDER STEP INPUT / 011610
IF(IOPT.GT.0.AND.J1.EQ.1) RETURN 011620
WRITE(6,1010) 011630
1010 FORMAT(1H1,2X,8HOPTION 2/2X,2(5H----)//) 011640
IF(IRCOT-1)1011,1015,1013 011650
1011 WRITE(6,1012) 011660
1012 FORMAT(/2X,43HNO COMPLEX ROOTS. REQUIREMENTS DO NOT APPLY) 011670
RETURN 011680
1013 WRITE(6,1014) 011690
1014 FORMAT(/2X,49HCOUPLED ROLL-SPIRAL MODE, YOU HAVE FAILED DYNAMIC 011700
* 12H STABILITY I) 011710
RETURN 011720
C 011730
C INITIALIZATION 011740
C 011750
1015 XTR=-1./TR 011760
XTS=-1./TS 011770
EF = 2.71828 011780
IF(ABS(XTS).NF.0.0)GO TO 1047 011790
WRITE(6,1048) 011800
1048 FORMAT(/2X,43HSPIRAL ROOT EQUALS ZERO, OPTION 2 EQUATIONS 011810
* 10H NOT VALID) 011820
RETURN 011830
C 011840
C BANK ANGLE RESPONSE FROM ROLL RATE EQUATION 011850
C 011860
1047 CON1=-CON 011870
COM1=CMPLX(CON1,CONA) 011880
DEN=COM*(COM-XTS)*(COM-XTR)*(COM+COM1) 011890
RDEN=REAL(DEN) 011900
AIDEN=AIMAG(DEN) 011910
PADEN=ATAN2(AIDEN,RDEN)*DTR 011920
IF(PADEN.LT.0.0)PADEN=PADEN+360. 011930
DENR=XTR*(XTR-XTS)*(XTR**2+2.*ZD*WD*XTR+WD**2) 011940
DENS=XTS*(XTS-XTR)*(XTS**2+2.*ZD*WD*XTS+WD**2) 011950
C 011960
C P(OSCILLATORY)/P(AVERAGE) 011970
C 011980
XKP=DP/E 011990
XKPR=FUNP(XTR)/DENR 012000
IF(OP.NE.0.0)GO TO 1050 012010
XKPS=1./A*(AP*XTS**2+BP*XTS+CP)/(1./XTS*DENS) 012020
GO TO 1052 012030
1050 XKPS=FUNP(XTS)/DENS 012040
1052 PNUM=1./A*(AP*COM**3+BP*COM**2+CP*COM+DP) 012050
RNUM=REAL(PNUM) 012060
AINUM=AIMAG(PNUM) 012070
XKP1=SQRT((RNUM**2+AINUM**2)/(RDEN**2+AIDEN**2)) 012080
XKPDR=2.*XKP1 012090

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

PANUM=ATAN2(AINUM,RNUM)*DTR                                012100
IF(PANUM.LT.0.0)PANUH=PANUM+360.                          012110
PSIP=PANUM-PADEN                                          012120
PSIPR=PSIP/DTR                                            012130
CON2=XKPR/(CON**2+CONA**2)                                012140
CON3=CON1*COS(PSIPR)-CONA*SIN(PSIPR)                     012150
TIME=0.0                                                  012160
P2=-999.                                                  012170
P3=PT(TIME)                                               012180
J = 1                                                      012190
IF(IOPT.GT.0) GO TO 1                                     012200
TIMEX(J) = TIME                                           012210
P3XX(J) = P3                                              012220
PDAXX(J) = PDA(TIME)                                      012230
1 DO 1025 I=1,3                                           012240
1018 P1=P2                                                 012250
      P2=P3                                                 012260
      TIME=TIME+.1                                         012270
      P3=PT(TIME)                                          012280
      J = J + 1                                           012290
      IF(IOPT.GT.0) GO TO 2                                 012300
      TIMEX(J) = TIME                                       012310
      P3XX(J) = P3                                         012320
      PDAXX(J) = PDA(TIME)                                 012330
2 IF(P1.NE.-999.)GO TO 1020                               012340
  IF(P3.GE.P2)GO TO 1018                                  012350
  WRITE(6,1019)                                           012360
1019 FORMAT(/2X,38HROLL RATE REVERSAL, TRY ANOTHER DESIGN) 012370
      GO TO 1027                                           012380
1020 IF(I.EQ.2)GO TO 1021                                  012390
      IF(P3.LT.P2)GO TO 1024                               012400
      GO TO 1022                                           012410
1021 IF(P3.GT.P2)GO TO 1024                               012420
1022 IF(TIME.LT.11.8) GO TO 1018                          012430
      WRITE(6,1023)                                        012440
1023 FORMAT(/2X,44HPEAK ROLL RATE OCCURS AFTER 12 SECONDS, TIME,
*          28H HISTORY LIMITATION EXCEEDED)               012450
      GO TO 1027                                           012460
1024 CALL PEAK(TIME-.2,TIME-.1,TIME,P1,P2,P3,TMAX,PHAX,1.) 012470
      TM(I)=TMAX                                           012480
      PH(I)=PHAX                                           012490
      IF(I.NE.2)GO TO 1025                                  012500
      IF(ZO.GT.2)GO TO 1026                                012510
1025 CONTINUE                                             012520
      POSPAV=(PM(1)+PM(3)-2.*PM(2))/(PM(1)+PM(3)+2.*PM(2)) 012530
      GO TO 1027                                           012540
1026 POSPAV=(PM(1)-PM(2))/(PM(1)+PM(2))                  012550
1027 P2OP1=PM(2)/PM(1)                                     012560
      TEND=TIME                                             012570
      JX = J                                               012580
C                                                         012590
C                                                         012600
C DELTA B(MAX)                                           012610
C                                                         012620
C                                                         012630
      XKB=DB/E                                             012640
      XKBR=FUNB(XTR)/DENR                                   012650
      XKBS=FUNB(XTS)/DENS                                   012660
      BNUM=1./A*(AB*COM**3+BB*COM**2+CB*COM+DB)           012670
      RNUM=REAL(BNUM)                                       012680
      AINUM=AIMAG(BNUM)                                       012690
      XKB1=SQRT((RNUM**2+AINUM**2)/(RDEN**2+AIDEN**2))

```

LATERAL-DIRECTIONAL PROGRAM LISTING

	XKBDR=2.*XKB1	012700
	BANUM=ATAN2(AINUM,RNUM)*DTR	012710
	IF(BANUM.LT.0.0)BANUM=BANUM+360.	012720
	PSIB=BANUM-PADEN	012730
	PSIBP=PSIB+ATAN2(WDD,CON)*DTR	012740
	TDDR2=PER/2.	012750
	TMAX1=AMAX1(TDDR2,2.)	012760
	BMAX1=BT(TMAX1)	012770
	BMAX=0.	012780
	BMIN=0.	012790
	TEST=0.	012800
	TIME=0.0	012810
	B2=-999.	012820
	J = 1	012830
	B2=BT(TIME)	012840
	IF(IOPT.LT.0)B3XX(J) = B3	012850
1031	B1=B2	012860
1032	B2=B3	012870
	TIME=TIME+.1	012880
	B3=BT(TIME)	012890
	J = J + 1	012900
	IF(IOPT.LT.0)B3XX(J) = B3	012910
	IF(B1.NE.-999.)GO TO 1036	012920
	IF(B3-B2)1033,1034,1035	012930
1033	ITEST=-1	012940
	GO TO 1038	012950
1034	B1=-999.	012960
	IF(TIME.LT.TMAX1)GO TO 1032	012970
	GO TO 1040	012980
1035	ITEST=1	012990
	GO TO 1038	013000
1036	IF(ITEST.GT.0)GO TO 1037	013010
	IF(B3.GT.B2)GO TO 1039	013020
	GO TO 1038	013030
1037	IF(B3.LT.B2)GO TO 1039	013040
1038	IF(TIME.LT.TMAX1)GO TO 1031	013050
	GO TO 1043	013060
1039	CALL PEAK(TIME-.2,TIME-.1,TIME,B1,B2,B3,TMAX,BM,1.)	013070
	IF(ITEST.LT.0)GO TO 1028	013080
	BMAX=BM	013090
	GO TO 1029	013100
1028	BMIN=BM	013110
1029	IF(TEST.EQ.1.)GO TO 1043	013120
	ITEST=-ITEST	013130
	TEST=1.	013140
	GO TO 1038	013150
1043	BNEG=0.	013160
	BPOS=0.	013170
	IF(BMAX1.GT.0.)GO TO 1055	013180
	BNEG=BMAX1	013190
	GO TO 1056	013200
1055	BPOS=BMAX1	013210
1056	IF(BMAX.GT.0.)GO TO 1057	013220
	BNEG=AMIN1(BNEG,BMAX)	013230
	GO TO 1058	013240
1057	BPOS=AMAX1(BPOS,BMAX)	013250
1058	IF(BMIN.GT.0.)GO TO 1059	013260
	BNEG=AMIN1(BNEG,BMIN)	013270
	GO TO 1060	013280
1059	BPOS=AMAX1(BPOS,BMIN)	013290

LATERAL-DIRECTIONAL PROGRAM LISTING

```

1060 DBMAX=8POS-BNEG                                013300
      GO TO 1041                                      013310
1040 DBMAX=BMAX1                                     013320
1041 IF(IOPT.GE.0) GO TO 1054                         013330
1053 IF(TIME.GE.TEND) GO TO 1054                    013340
      TIME=TIME+.1                                    013350
      J = J + 1                                       013360
      B3XX(J) = BT(TIME)                              013370
      GO TO 1053                                      013380
C                                                    013390
C ANGLE P/B                                          013400
C                                                    013410
1054 POBN=COM*ANUM                                    013420
      PBANUM=ATAN2(AIMAG(POBN),REAL(POBN))*DTR        013430
      IF(PBANUM.LT.0.0) PBANUM=PBANUM+360.          013440
      PBA DEN=ATAN2(AIMAG(ADEN),REAL(ADEN))*DTR      013450
      IF(PBA DEN.LT.0.0) PBA DEN=PBA DEN+360.        013460
      APOB=PBANUM-PBA DEN                             013470
C                                                    013480
C KD/KSS                                             013490
C                                                    013500
      XKDKSS=XKPDR/XKPS                               013510
C                                                    013520
C WRITE OUTPUT                                       013530
C                                                    013540
      IF(APOB.LT.0.0) APOB=APOB+360.                013550
      IF(PSIP.GT.0.0) PSIP=PSIP-360.                013560
      IF(PSIB.GT.0.0) PSIB=PSIB-360.                013570
      IF(IOPT.GT.0) GO TO 3                          013580
      WRITE(6,4)                                      013590
4      FORMAT(1H,31TIME HISTORIES FOR A STEP INPUT// 013600
1      10X4HTIME,5X15HP(T), ROLL RATE,5X18PHI(T), ROLL ANGLE, 013610
2      5X,17HBETA(T), SIDESLIP/10X,3HSEC,10X7HDEG/SEC,16X3HDEG,20X, 013620
3      3HDEG//)
      WRITE(6,5)(TIMEX(J),P3XX(J),PDAXX(J),B3XX(J), J=1,JX) 013640
5      FORMAT(8X,F6.1,6X,E11.4,10X,E11.4,12X,E11.4) 013650
      IF(PLT.GT.0..AND.J1.EQ.0) CALL PLTUP(DATA,TIMEX,P3XX,PDAXX, 013660
1      B3XX,JX,TITLE,RUN,WORD1)
      IF(PLT.GT.0..AND.J1.EQ.1) CALL PLTUP(DATA,TIMEX,P3XX,PDAXX, 013680
1      B3XX,JX,TITLE,RUN,WORD2)
      IF(PLT.GT.0.) IPLT=1                            013700
3      IF(J1.EQ.1) RETURN                             013710
      WRITE(6,1042) POSPAV,DBMAX,APOB,PSIP,PSIB,XKDKSS,XKP,XKB,POSPAV, 013720
1      XKPR,XKBR,PSIBP,XKPS,XKBS,P2OP1,XKPDR,XKBDR 013730
1042 FORMAT(/2X,10HPOSC/PAV =,E12.4,7X,07HDBMAX =,E12.4,11X, 11HANGLE 013740
      *P/B =,E12.4,/6X,06HPSIP =,E12.4,8X,06HPSIB =,E12.4,14X, 8HKD013750
      */KSS =,E12.4,/8X,04HKP =,E12.4,10X,04HKB =,E12.4,6X, 16HPHI OSC013760
      */PHI AV =,E12.4,/7X,05HKPR =,E12.4,9X,05HKBR =,E12.4,15X, 013770
      *07HPSIBP =,E12.4,/7X,05HKPS =,E12.4,9X,05HKBS =,E12.4,15X, 013780
      *07HP2/P1 =,E12.4,/4X,08HMKPDR =,E12.4,6X,08HMKBDR = 013790
      *,E12.4)
      RETURN                                          013800
      END                                            013820
      SUBROUTINE PEAK(Y1,Y2,Y3,X1,X2,X3,PIV,POV,PCTPK) 013830
      A=(((Y2-Y3)*(X1-X2))-((Y1-Y2)*(X2-X3)))/(((Y2-Y3)* 013840
1      (Y1**2-Y2**2))-((Y1-Y2)*(Y2**2-Y3**2)))
      B=(((X2-X3)-A*(Y2**2-Y3**2))/(Y2-Y3) 013860
      C=X1-B*Y1-A*Y1**2 013870
      PIV=-B/(2.*A) 013880
      POV=(4.*A*C-B**2)/(4.*A) 013890

```

LATERAL-DIRECTIONAL PROGRAM LISTING

```

IF(ABS(1.0-PCTPK)-.0001)1,1,2          013900
2  PDV=PDV*PCTPK                        013910
   PIV=PIV+SQRT( (PCTPK-1.0)*PDV/A)    013920
1  RETURN                               013930
   END                                  013940
SUBROUTINE PLCTS(N)                    013950
RETURN                                  013960
END                                      013970
SUBROUTINE DMULR(COE1,N,ROOT1,ROOTI1)  013980
DOUBLE PRECISION COE1,ROOT1,ROOTI1    013990
DIMENSION COE1(14),RCOT1(12),ROOTI1(12),COE(7),ROOTR(6),ROOTI(6) 014000
NN=N+1                                  014010
DO 1 I=1,NN                             014020
1  COE(I)=COE1(I)                        014030
   CALL SMULR(COE,N,ROOTR,ROOTI)        014040
   DO 2 I=1,N                             014050
   ROOT1(I)=ROOTR(I)                    014060
2  ROOTI1(I)=ROOTI(I)                  014070
RETURN                                  014080
END                                      014090
SUBROUTINE SMULR (COE,N1,ROOTR,ROOTI)  014100
C                                       014110
C                                       014120
C .....                                014130
C                                       014140
C POLYNOMIAL ROOT FINDER SUBROUTINE .... 014150
C                                       014160
C ITERATIVE METHOD FOR POLYNOMIAL EQUATIONS .... 014170
C                                       014180
C THIS METHOD APPROXIMATES THE FUNCTION F(Z) BY A QUADRATIC 014190
C WHICH MAY ,IN GENERAL, HAVE COMPLEX COEFFICIENTS AND DOES NOT 014200
C REQUIRE A KNOWLEDGE OF THE DERIVATIVE OF F(Z) THOUGH 014210
C THE FUNCTION F(Z) MUST BE EVALUATED ONCE PER ITERATION .... 014220
C                                       014230
C THIS SUBROUTINE FINDS REAL AND COMPLEX ROOTS OF A POLYNOMIAL 014240
C WITH REAL COEFFICIENTS .... 014250
C                                       014260
C                                       014270
C USE OF MULLER SUBROUTINE .... 014280
C 1. CALL SMULR (COE,N1,ROOTR,ROOTI) .... 014290
C 2. COE IS THE TAG OF THE ARRAY OF COEFFICIENTS. 014300
C THE COEFFICIENTS MUST BE ORDERED FROM HIGHEST DEGREE 014310
C TO LOWEST DEGREE . 014320
C 3. N1 IS DEGREE OF THE POLYNOMIAL . 014330
C 4. ROOTR IS THE TAG OF THE ARRAY WHERE THE REAL PARTS 014340
C OF THE COMPLEX ROOTS ARE STORED . 014350
C 5. RCOTI IS THE TAG OF THE ARRAY WHERE THE IMAGINARY 014360
C PARTS OF THE COMPLEX ROOTS ARE STORED .... 014370
C                                       014380
C ALL ARITHMETIC IS IN THE COMPLEX MODE .... 014390
C THEREFORE UNDER-FLOW IS NORMAL FOR REAL ROOTS .... 014400
C                                       014410
C MULTIPLE ROOTS DECREASES ACCURACY OF THIS SUBROUTINE . 014420
C WHEN MULTIPLICITY IS FOUR THE ACCURACY DECREASES TO 014430
C ABOUT TWO PLACES .... 014440
C                                       014450
C RUNNING TIME IS APPROXIMATELY PROPORTIONAL TO 014460
C DEGREE SQUARED DIVIDED BY TWENTY .... 014470
C FOR DEGREE ELEVEN IT TAKES SIX SECONDS .... 014480
C                                       014490

```


LATERAL-DIRECTIONAL PROGRAM LISTING

C		014500
C		014510
C	*****	014520
C		014530
C		014540
C		014550
C		014560
	DIMENSION COE(1),ROOTR(1),ROOTI(1)	014570
C		014580
	N2=N1+1	014590
	N4=0	014600
	I=N1+1	014610
19	IF(COE(I))9,7,9	014620
7	N4=N4+1	014630
	ROOTR(N4)=0.0	014640
	ROOTI(N4)=0.0	014650
	I=I-1	014660
	IF(N4-N1)19,37,19	014670
9	CONTINUE	014680
C		014690
10	AXR=0.8	014700
	AXI=0.0	014710
	L=1	014720
	N3=1	014730
	ALP1R=AXR	014740
	ALP1I=AXI	014750
	M=1	014760
	GO TO 99	014770
C		014780
11	BET1R=TEMR	014790
	BET1I=TEMI	014800
	AXR=0.85	014810
	ALP2R=AXR	014820
	ALP2I=AXI	014830
	M=2	014840
	GO TO 99	014850
C		014860
12	BET2R=TEMR	014870
	BET2I=TEMI	014880
	AXR=0.9	014890
	ALP3R=AXR	014900
	ALP3I=AXI	014910
	M=3	014920
	GO TO 99	014930
C		014940
13	BET3R=TEMR	014950
	BET3I=TEMI	014960
14	TE1=ALP1R-ALP3R	014970
	TE2=ALP1I-ALP3I	014980
	TE5=ALP3R-ALP2R	014990
	TE6=ALP3I-ALP2I	015000
	TEM=TE5*TE5+TE6*TE6	015010
	TE3=(TE1*TE5+TE2*TE6)/TEM	015020
	TE4=(TE2*TE5-TE1*TE6)/TEM	015030
	TE7=TE3+1.0	015040
	TE9=TE3*TE3-TE4*TE4	015050
	TE10=2.0 *TE3*TE4	015060
	DE15=TE7*BET3R-TE4*BET3I	015070
	DE16=TE7*BET3I+TE4*BET3R	015080
	TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15	015090

LATERAL-DIRECTIONAL PROGRAM LISTING

```

TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16      015100
TE7=TE9-1.0                                015110
TE1=TE9*BET2R-TE10*BET2I                  015120
TE2=TE9*BET2I+TE10*BET2R                  015130
TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I      015140
TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R      015150
TE15=DE15*TE3-DE16*TE4                    015160
TE16=DE15*TE4+DE16*TE3                    015170
TE1=TE13*TE13-TE14*TE14-4.0 *(TE11*TE15-TE12*TE16) 015180
TE2=2.0 *TE13*TE14-4.0 *(TE12*TE15+TE11*TE16) 015190
TEM= SQRT(TE1*TE1+TE2*TE2)                 015200
IF (TE1) 113, 113, 112                     015210
113 TE4= SQRT(0.5 *(TEM-TE1))               015220
TE3=0.5 *TE2/TE4                           015230
GO TO 111                                    015240
C                                             015250
112 TE3= SQRT(0.5 *(TEM+TE1))               015260
IF (TE2) 110, 200, 200                      015270
110 TE3=-TE3                                 015280
200 TE4=0.5 *TE2/TE3                         015290
111 TE7=TE13+TE3                             015300
TE8=TE14+TE4                                 015310
TE9=TE13-TE3                                 015320
TE10=TE14-TE4                               015330
TE1=2.0 *TE15                               015340
TE2=2.0 *TE16                               015350
IF (TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10) 204, 204, 205 015360
204 TE7=TE9                                  015370
TE8=TE10                                     015380
205 TEM=TE7*TE7+TE8*TE8                     015390
TE3=(TE1*TE7+TE2*TE8)/TEM                  015400
TE4=(TE2*TE7-TE1*TE8)/TEM                  015410
AXR=ALP3R+TE3*TE5-TE4*TE6                 015420
AXI=ALP3I+TE3*TE6+TE4*TE5                 015430
ALP4R=AXR                                   015440
ALP4I=AXI                                   015450
M=4                                          015460
GO TO 99                                     015470
C                                             015480
15 N6=1                                      015490
C*****015500
38 IF( ABS(HELL)+ ABS(BELL)-1.0E-20)18,18,16 015510
16 TE7= ABS(ALP3R-AXR)+ ABS(ALP3I-AXI)      015520
IF(TE7/( ABS(AXR)+ ABS(AXI))-1.0E-7)18,18,17 015530
C*****015540
17 N3=N3+1                                  015550
ALP1R=ALP2R                                 015560
ALP1I=ALP2I                                 015570
ALP2R=ALP3R                                 015580
ALP2I=ALP3I                                 015590
ALP3R=ALP4R                                 015600
ALP3I=ALP4I                                 015610
BET1R=BET2R                                 015620
BET1I=BET2I                                 015630
BET2R=BET3R                                 015640
BET2I=BET3I                                 015650
BET3R=TEMR                                  015660
BET3I=TEMI                                  015670
IF(N3-100)14,18,18                         015680
18 N4=N4+1                                  015690

```

LATERAL-DIRECTIONAL PROGRAM LISTING

	ROOTR(N4)=ALP4R	015700
	ROOTI(N4)=ALP4I	015710
	N3=0	015720
41	IF(N4-N1) 30,37,37	015730
37	RETURN	015740
	015750
30	IF(ABS(ROOTI(N4))-1.0E-5) 10,10,31	015760
31	GO TO (32,10),L	015770
32	AXR=ALP1R	015780
	AXI=-ALP1I	015790
	ALP1I=-ALP1I	015800
	M=5	015810
	GO TO 99	015820
33	BET1R=TEMR	015830
	BET1I=TEMI	015840
	AXR=ALP2R	015850
	AXI=-ALP2I	015860
	ALP2I=-ALP2I	015870
	M=6	015880
	GO TO 99	015890
	C	015900
34	BET2R=TEMR	015910
	BET2I=TEMI	015920
	AXR=ALP3R	015930
	AXI=-ALP3I	015940
	ALP3I=-ALP3I	015950
	L=2	015960
	M=3	015970
99	TEMR=COE(I)	015980
	TEMI=0.0	015990
	DO 100 I=1,N1	016000
	TE1=TEMR*AXR-TEMI*AXI	016010
	TEMI=TEMI*AXR+TEMR*AXI	016020
100	TEMR=TE1+COE(I+1)	016030
	HELL=TEMR	016040
	BELL=TEMI	016050
42	IF(N4) 102,103,102	016060
102	@@ 101 I=1,N4	016070
	TEM1=AXR-ROOTR(I)	016080
	TEM2=AXI-ROOTI(I)	016090
	TE1=TEM1*TEM1+TEM2*TEM2	016100
	TE2=(TEMR*TEM1+TEMI*TEM2)/TE1	016110
	TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1	016120
101	TEMR=TE2	016130
103	GO TO (11,12,13,15,33,34),M	016140
	END	016150

LATERAL-DIRECTIONAL PROGRAM DATA

010013-02MEDIUM FIGHTER, H=30,000FT, CG=20C, M=.9, PYLON TANKS						815166	LAT 13-1
.0008907	895.	220.	25000.	28.	38300.		LAT 13-2
75000.	-10000.	32.082					LAT 13-3
-.02			.0132		.025		LAT 13-4
-.0035		-.0085	.00463	.0098	.0051		LAT 13-5
.0051	-.0014	-.0057	-.0144	.003	.0025		LAT 13-6
							LAT 13-7
5102-8-02LARGE TRANSPORT, H=30000FT, CG=25, M=.745, START CRUISE 8/5/66							LAT2-B 1
.00089068	743.	4900.	350000.	200.	21000000.		LAT2-B 2
34000000.	1700000.	32.082					LAT2-B 3
-.0145			.007	-.0003	.0028		LAT2-B 4
-.0017		-.0096	.0035	.00071	.00031		LAT2-B 5
.0017		-.00061	-.0041	.000203	-.00132		LAT2-B 6
							LAT2-B 7

ROOTS OF A/2 LATERAL DIRECTIONAL TRANSFER FUNCTIONS

RUN NO. 013

MEDIUM FIGHTER, H=30,000FT, CG=200, M=.9, PYLON TANKS 815166

INPUT DATA (NON-DIMENSIONAL) PER DEGREE

RHO = .8907E+03 U = .8930E+03 S = .2200E+03 GWT = .2500E+05 SPAN = .2800E+02 IXB = .3830E+05
 IZB = .7500E+03 IZB = .8930E+03 G = .3208E+02 ALFI = 0.0 GANA = 0.0 LX = 0.0
 CYR = -.2000E-01 CYR = 0.0 CYP = .1320E-01 CYR = .4630E-02 CLDA = 0.0 CYDA = 0.0 CYDR = .2500E-01
 CLB = -.3900E-02 CLB = 0.0 CLP = -.8500E-02 CLR = .4630E-02 CLDA = 0.0 CYDA = 0.0 CYDR = .2500E-01
 CNB = .5100E-02 CNB = 0.0 CNP = -.5700E-02 CNR = -.1440E-01 CNDA = .3000E-02 CLDR = .5100E-02
 ALFA = 0.0 ALFA = 0.0 ALFX = 0.0

DIMENSIONAL STABILITY DERIVATIVES

YB = -.1157E+03 YBD = 0.0 YP = 0.0 YR = .1195E+01 YDA = 0.0 YDR = .1447E+03
 LB = -.1151E+02 LBD = 0.0 LP = -.4371E+00 LR = .2381E+00 LDA = 0.0 LDR = .1677E+02
 NB = .6562E+01 NBD = -.3676E-01 NP = -.1497E+00 NR = -.3781E+00 NDA = .5036E+01 NOR = .4197E+01

DIMENSIONAL STABILITY DERIVATIVES PRIMED

ALFI = 0.0 ALFA = 0.0 ALFX = 0.0 IX = .3830E+05 IZ = .7500E+05 IXZ = -.1000E+05
 LBP = -.1424E+02 LBDP = .9945E-02 LPP = -.4124E+00 LRP = .3490E+00 LQAP = .3202E+02 LDRP = .1624E+02
 NBP = .1046E+02 NBDP = -.3809E-01 NPP = -.9470E-01 NRP = -.4247E+00 NDAP = .7675E+00 NDRP = .2032E+01

LATERAL DIRECTIONAL DENOMINATOR ROOTS

ROOTS (COMPLEX FORM) 0.0
 -.57310+00 0.0
 -.14220-01
 -.17050+00
 -.17350+00
 -.32420+01
 .32420+01

TS = .703180E+02 TR = .174433E+01 ZDR = .525218E-01 MDR = .324607E+01 RAD/SEC WDR = .324159E+01 RAD/SEC
 .516629E+00 CYCLES/SEC WDDR = .519916E+00 CYCLES/SEC

OUTCH ROLL MODE

TDR = .13356E+01 TIME TO HALF AMP. = .40656E+01
 TDDR = .13383E+01 CYCLES TO HALF AMP. = .20975E+01
 ONE OVER CYCLES TO HALF AMP. = .47675E+00
 ONE OVER CYCLES TO ONE TENTH AMP. = .14352E+00
 ONE OVER CYCLES TO ONE TENTH AMP. = .10937E+00
 2*ZD*MDR = .34098E+00
 WDRSQ = .10937E+00

COEFFICIENTS

A = .10000E+01 B = .92832E+00 C = .10745E+02 D = .61915E+01 E = .85881E-01

PHI TO BETA RATIO = .1350E+01

PHI TO EQUIV VEL = .1412E+00

FREQ SQUARED TIMES PHI TO BETA RATIO = .1423E+02

AFFDL-TR-78-203

RUN NO. 013 AILERON NUMERATOR ROOTS

SLIP TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0 0.0
-.12560+00
.51610+01

1/TB1 = .125534E+00 1/TB2 = -.516108E+01

AB = 0. BB = -.7665E+00 CB = .3859E+01 DB = .4970E+00

ROLL ANGLE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0
-.26200+00 -.32820+01
-.26200+00 .32820+01

1/TP = 0. ZP = .795788E-01 WP = .329284E+01 WPHI/WCR = .101441E+01

AP = .3202E+02 BP = .1678E+02 CP = .3471E+03 DP = 0.

YAW RATE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0 0.0
.26090+01 -.14560+01
.26090+01 .14560+01
-.18090+01

ZR = -.873257E+00 WR = .298775E+01 1/TR = .180936E+01

AR = .76748E+00 BR = -.26162E+01 CR = -.39514E+00 DR = .12396E+02

OPTION 2

TIME HISTORIES FOR 4 STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.1201E-07	0.	.6827E-08
.1	.3139E+01	.1580E+00	-.3056E-02
.2	.6160E+01	.6239E+00	-.9066E-02
.3	.9066E+01	.1386E+01	-.1340E-01
.4	.1186E+02	.2433E+01	-.1192E-01
.5	.1453E+02	.3753E+01	-.1414E-02
.6	.1706E+02	.5334E+01	.2016E-01
.7	.1945E+02	.7161E+01	.5349E-01
.8	.2170E+02	.9220E+01	.9788E-01
.9	.2378E+02	.1149E+02	.1514E+00
1.0	.2570E+02	.1397E+02	.2111E+00
1.1	.2745E+02	.1663E+02	.2734E+00
1.2	.2906E+02	.1946E+02	.3345E+00
1.3	.3052E+02	.2243E+02	.3906E+00
1.4	.3185E+02	.2555E+02	.4336E+00
1.5	.3308E+02	.2880E+02	.4762E+00
1.6	.3422E+02	.3217E+02	.5021E+00
1.7	.3530E+02	.3564E+02	.5162E+00
1.8	.3633E+02	.3923E+02	.5196E+00
1.9	.3733E+02	.4291E+02	.5142E+00
2.0	.3830E+02	.4669E+02	.5029E+00
2.1	.3926E+02	.5057E+02	.4888E+00
2.2	.4021E+02	.5454E+02	.4754E+00
2.3	.4114E+02	.5861E+02	.4657E+00
2.4	.4204E+02	.6277E+02	.4623E+00
2.5	.4290E+02	.6702E+02	.4670E+00
2.6	.4373E+02	.7135E+02	.4806E+00
2.7	.4449E+02	.7576E+02	.5029E+00
2.8	.4519E+02	.8025E+02	.5328E+00
2.9	.4582E+02	.8480E+02	.5686E+00
3.0	.4638E+02	.8941E+02	.6077E+00
3.1	.4686E+02	.9407E+02	.6473E+00
3.2	.4728E+02	.9878E+02	.6848E+00
3.3	.4764E+02	.1035E+03	.7176E+00
3.4	.4795E+02	.1083E+03	.7437E+00
3.5	.4822E+02	.1131E+03	.7621E+00
3.6	.4847E+02	.1179E+03	.7722E+00
3.7	.4872E+02	.1228E+03	.7745E+00
3.8	.4896E+02	.1277E+03	.7701E+00
3.9	.4921E+02	.1326E+03	.7609E+00
4.0	.4947E+02	.1375E+03	.7491E+00
4.1	.4975E+02	.1425E+03	.7370E+00
4.2	.5003E+02	.1475E+03	.7270E+00
4.3	.5032E+02	.1525E+03	.7211E+00
4.4	.5060E+02	.1575E+03	.7207E+00
4.5	.5088E+02	.1626E+03	.7268E+00
4.6	.5113E+02	.1677E+03	.7395E+00
4.7	.5136E+02	.1728E+03	.7582E+00
4.8	.5155E+02	.1780E+03	.7817E+00
4.9	.5171E+02	.1832E+03	.8085E+00
5.0	.5182E+02	.1883E+03	.8366E+00
5.1	.5190E+02	.1935E+03	.8640E+00
5.2	.5195E+02	.1987E+03	.8889E+00
5.3	.5196E+02	.2039E+03	.9095E+00
5.4	.5197E+02	.2091E+03	.9249E+00
5.5	.5196E+02	.2143E+03	.9346E+00
5.6	.5195E+02	.2195E+03	.9385E+00
5.7	.5195E+02	.2247E+03	.9373E+00
5.8	.5196E+02	.2299E+03	.9322E+00
5.9	.5198E+02	.2351E+03	.9247E+00
6.0	.5203E+02	.2403E+03	.9164E+00
6.1	.5209E+02	.2455E+03	.9090E+00
6.2	.5216E+02	.2507E+03	.9042E+00
6.3	.5223E+02	.2559E+03	.9030E+00
6.4	.5231E+02	.2611E+03	.9063E+00
6.5	.5238E+02	.2664E+03	.9144E+00
6.6	.5244E+02	.2716E+03	.9272E+00
6.7	.5248E+02	.2769E+03	.9438E+00
6.8	.5250E+02	.2821E+03	.9633E+00
6.9	.5249E+02	.2874E+03	.9843E+00

POSC/PA/ = .2733E-02	DBMAX = .5332E+00	ANGLE P/B = .9869E+02
PSIP = -.2005E+03	PSIB = -.2992E+03	KD/KSS = .1034E-01
KP = 0.	KB = .5787E+01	PHI OSC/PHI AV = .2749E-02
KPR = -.5835E+02	KBR = -.5754E+00	PSIBP = -.2062E+03
KPS = .5833E+02	KBS = -.5279E+01	P2/P1 = .9996E+00
MKPPDX = .6033E+00	MKBPJR = .1390E+00	

AFFDL-TR-78-203

RUN NO. 013 RUDDER NUMERATOR ROOTS

SIJESLIP TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0 0.0
.9224D+00
-.1668D+00
.1096D+02

1/TB1 = -.322357E+00 1/TB2 = .166759E+00 1/TB3 = -.109624E+02

AB = .1616E+00 BB = -.189+E+01 CB = .1314E+01 DB = .2726E+00

ROLL ANGLE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.
-.2083D+00 -.3435D+01
-.2083D+00 .3495D+01

1/TP = 0. Z² = .594737E-01 WP = .350160E+01 WPHI/WDR = .107872E+01

AP = .1624E+02 BP = .6763E+01 CP = .1991E+03 DP = 0.

YAW RATE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0 0.0
.5202D+00 .1350D+01
.5202D+00 -.1360D+01
-.1658D+01

ZR = -.357135E+00 WR = .145628E+01 1/TR = .165807E+01

AR = .20261E+01 BR = .12515E+01 CR = .80185E+00 DR = .71244E+01

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.4488E-07	0.	.6897E-09
.1	.1587E+01	.7993E-01	.6270E-02
.2	.3121E+01	.3157E+00	-.5339E-02
.3	.4624E+01	.7031E+00	-.3078E-01
.4	.5112E+01	.1240E+01	-.6483E-01
.5	.7592E+01	.1925E+01	-.1016E+00
.6	.9062E+01	.2758E+01	-.1354E+00
.7	.1051E+02	.3737E+01	-.1608E+00
.8	.1193E+02	.4860E+01	-.1738E+00
.9	.1329E+02	.6121E+01	-.1716E+00
1.0	.1458E+02	.7516E+01	-.1535E+00
1.1	.1578E+02	.9035E+01	-.1202E+00
1.2	.1687E+02	.1067E+02	-.7406E-01
1.3	.1769E+02	.1241E+02	-.1887E-01
1.4	.1871E+02	.1423E+02	.4083E-01
1.5	.1945E+02	.1614E+02	.1001E+00
1.6	.2008E+02	.1812E+02	.1542E+00
1.7	.2063E+02	.2016E+02	.1989E+00
1.8	.2110E+02	.2224E+02	.2312E+00
1.9	.2153E+02	.2438E+02	.2493E+00
2.0	.2193E+02	.2655E+02	.2530E+00
2.1	.2233E+02	.2876E+02	.2435E+00
2.2	.2274E+02	.3101E+02	.2231E+00
2.3	.2317E+02	.3331E+02	.1954E+00
2.4	.2363E+02	.3565E+02	.1645E+00
2.5	.2412E+02	.3804E+02	.1345E+00
2.6	.2463E+02	.4048E+02	.1094E+00
2.7	.2515E+02	.4296E+02	.9253E-01
2.8	.2566E+02	.4551E+02	.8619E-01
2.9	.2615E+02	.4810E+02	.9151E-01
3.0	.2661E+02	.5073E+02	.1084E+00
3.1	.2702E+02	.5342E+02	.1354E+00
3.2	.2736E+02	.5614E+02	.1703E+00
3.3	.2764E+02	.5889E+02	.2100E+00
3.4	.2786E+02	.6166E+02	.2509E+00
3.5	.2802E+02	.6446E+02	.2895E+00
3.6	.2812E+02	.6726E+02	.3226E+00
3.7	.2819E+02	.7008E+02	.3478E+00
3.8	.2823E+02	.7290E+02	.3632E+00
3.9	.2826E+02	.7573E+02	.3684E+00
4.0	.2830E+02	.7855E+02	.3638E+00
4.1	.2835E+02	.8139E+02	.3507E+00
4.2	.2843E+02	.8423E+02	.3315E+00
4.3	.2855E+02	.8707E+02	.3089E+00
4.4	.2869E+02	.8994E+02	.2859E+00
4.5	.2886E+02	.9281E+02	.2655E+00
4.6	.2905E+02	.9571E+02	.2503E+00
4.7	.2925E+02	.9862E+02	.2422E+00
4.8	.2945E+02	.1016E+03	.2423E+00
4.9	.2964E+02	.1045E+03	.2510E+00
5.0	.2980E+02	.1075E+03	.2675E+00
5.1	.2993E+02	.1105E+03	.2905E+00
5.2	.3002E+02	.1135E+03	.3178E+00
5.3	.3007E+02	.1165E+03	.3471E+00
5.4	.3008E+02	.1195E+03	.3758E+00
5.5	.3005E+02	.1225E+03	.4014E+00
5.6	.3000E+02	.1255E+03	.4220E+00
5.7	.2994E+02	.1285E+03	.4360E+00
5.8	.2987E+02	.1315E+03	.4427E+00
5.9	.2980E+02	.1345E+03	.4423E+00
6.0	.2975E+02	.1374E+03	.4353E+00
6.1	.2971E+02	.1404E+03	.4233E+00
6.2	.2971E+02	.1434E+03	.4081E+00
6.3	.2973E+02	.1464E+03	.3919E+00
6.4	.2977E+02	.1493E+03	.3767E+00
6.5	.2983E+02	.1523E+03	.3646E+00
6.6	.2991E+02	.1553E+03	.3572E+00
6.7	.2999E+02	.1583E+03	.3555E+00
6.8	.3007E+02	.1613E+03	.3600E+00
6.9	.3014E+02	.1643E+03	.3703E+00
7.0	.3019E+02	.1673E+03	.3858E+00
7.1	.3021E+02	.1703E+03	.4051E+00
7.2	.3021E+02	.1734E+03	.4265E+00

ROOTS OF 4/0 LATERAL DIRECTIONAL TRANSFER FUNCTIONS

RUN NO. 2-9

LARGE TRANSPORT, H=3000FT, CG=25, M=.749, START CRUISE 8/5/66

INPUT DATA (NON-DIMENSIONAL) PER DEGREE

RHO = .8907E-03 U = .7430E+03 S = .4900E+04 GWT = .3500E+06 IXB = .2100E+08
 IZB = .3400E+03 IXZB = .1700E+07 G = .3200E+02 ALFI = 0. LX = 0.
 CYSB = -.1450E-01 CYB0 = 0. CYP = 0. CYR = .7600E-02 CYDR = .2800E-02
 CLB = -.1700E-02 CLB0 = 0. CLP = -.9000E-02 CLD = .3500E-02 CLDR = .3100E-03
 CNB = .1700E-02 CNB0 = 0. CNP = -.6100E-03 CNR = -.4400E-02 CNDR = -.1320E-02
 ALFA = 0. ALFX = 0.

DIMENSIONAL STABILITY DERIVATIVES

YB = -.9200E+02 YB0 = 0. YP = 0. YR = .5978E+01 YOR = .1777E+02
 LB = -.1118E+01 LB0 = 0. LP = -.6493E+00 LOR = .3097E+00 LDR = .2038E+00
 NB = .6902E+03 NBD = 0. NP = -.3333E-01 NRP = -.2240E+00 NDR = -.5539E+00

DIMENSIONAL STABILITY DERIVATIVES PRIME

ALFI = 0. ALFA = 0. ALFX = 0. IX = .2100E+08 IXZ = .1700E+07
 LBP = -.1060E+01 LBP0 = 0. LPP = -.6555E+00 LRP = .2927E+00 LDRP = .1610E+00
 NBP = .6369E+03 NBP0 = 0. NPP = -.7641E-01 NRP = -.2094E+00 NDRP = -.5279E+00

LATERAL DIRECTIONAL DENOMINATOR ROOTS

ROOTS (COMPLEX FORM)
 0.0
 0.0
 -.12220+00 .84690+00
 -.12220+00 -.84690+00
 -.23030-02
 -.9+210+00

TS = .434206E+03 TR = .165150E+01 ZOR = .142794E+00 WDR = .855656E+00 RAD/SEC
 WDR = .136182E+00 CYCLES/SEC WDDR = .846888E+00 RAD/SEC
 WDDR = .134786E+00 CYCLES/SEC

OUTCH ROLL MODE

TOR = .74431E+01 TIME TO HALF AMP. = .56731E+01
 TORR = .74191E+01 CYCLES TO HALF AMP. = .76405E+00
 ONE OVER CYCLES TO HALF AMP. = .13078E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .24437E+00
 ONE OVER CYCLES TO ONE TENTH AMP. = .18645E+02
 ONE OVER CYCLES TO ONE TENTH AMP. = .25401E+01
 ONE OVER CYCLES TO ONE TENTH AMP. = .39368E+00
 WDRSQ = .73215E+00

COEFFICIENTS

A = .10000E+01 B = .11387E+01 C = .95509E+00 D = .69195E+00 E = .15885E-02

PHI TO BETA RATIO = .1130E+01

PHI TO EQUIV VEL = .1423E+00

FREQ SQUARED TIMES PHI TO BETA RATIO = .6273E+00

RUN NO. 2-3 AILERON NUMERATOR ROOTS

SI/ESLIP TO CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)

0.0 0.0
 .11960+00
 -.43990+00
 -.41860+02

1/TB1 = -.119553E+00 1/TB2 = .439902E+00 1/TB3 = .418598E+02

AB = -.2562E-02 BB = -.1081E+00 CB = -.3422E-01 DB = .5640E-02

ROLL ANGLE TO CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)

0.
 -.20220+00 .92810+00
 -.20220+00 -.92810+00

1/TP = 0. ZP = .212856E+00 WP = .949864E+00 WPHI/WDR = .111010E+01

AP = .4753E+10 BP = .1922E+00 CP = .4289E+00 DP = 0.

YAW RATE TO CONTROL DEFLECTION
 ROOTS (COMPLEX FORM)

0.0 0.0
 .96900-01 .44450+00
 .96900-01 -.44450+00
 -.81710+00

ZR = -.21298+E+00 WR = .454967E+00 1/TR = .817079E+00

AR = .10619E+00 BR = .66183E-01 CR = .51653E-02 DR = .17959E-01

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.1319E-08	0.	-.3020E-13
.1	.4573E-01	.2316E-02	-.7658E-03
.2	.8813E-01	.9036E-02	-.2489E-02
.3	.1275E+00	.1984E-01	-.5076E-02
.4	.1643E+00	.3445E-01	-.8437E-02
.5	.1986E+00	.5261E-01	-.1248E-01
.6	.2307E+00	.7409E-01	-.1711E-01
.7	.2609E+00	.9869E-01	-.2225E-01
.8	.2894E+00	.1262E+00	-.2780E-01
.9	.3162E+00	.1565E+00	-.3368E-01
1.0	.3416E+00	.1894E+00	-.3981E-01
1.1	.3656E+00	.2248E+00	-.4610E-01
1.2	.3885E+00	.2625E+00	-.5249E-01
1.3	.4101E+00	.3024E+00	-.5889E-01
1.4	.4307E+00	.3445E+00	-.6524E-01
1.5	.4503E+00	.3886E+00	-.7146E-01
1.6	.4690E+00	.4345E+00	-.7750E-01
1.7	.4867E+00	.4823E+00	-.8329E-01
1.8	.5036E+00	.5318E+00	-.8879E-01
1.9	.5195E+00	.5830E+00	-.9394E-01
2.0	.5347E+00	.6357E+00	-.9869E-01
2.1	.5490E+00	.6899E+00	-.1030E+00
2.2	.5626E+00	.7455E+00	-.1069E+00
2.3	.5753E+00	.8024E+00	-.1102E+00
2.4	.5872E+00	.8605E+00	-.1130E+00
2.5	.5984E+00	.9198E+00	-.1153E+00
2.6	.6087E+00	.9802E+00	-.1171E+00
2.7	.6183E+00	.1042E+01	-.1182E+00
2.8	.6271E+00	.1104E+01	-.1188E+00
2.9	.6351E+00	.1167E+01	-.1188E+00
3.0	.6423E+00	.1231E+01	-.1182E+00
3.1	.6488E+00	.1295E+01	-.1171E+00
3.2	.6545E+00	.1361E+01	-.1154E+00
3.3	.6595E+00	.1426E+01	-.1132E+00
3.4	.6637E+00	.1492E+01	-.1105E+00
3.5	.6672E+00	.1559E+01	-.1072E+00
3.6	.6701E+00	.1626E+01	-.1035E+00
3.7	.6722E+00	.1693E+01	-.9943E-01
3.8	.6736E+00	.1760E+01	-.9491E-01
3.9	.6745E+00	.1828E+01	-.9002E-01
4.0	.6747E+00	.1895E+01	-.8480E-01
4.1	.6743E+00	.1963E+01	-.7928E-01
4.2	.6734E+00	.2030E+01	-.7352E-01
4.3	.6719E+00	.2097E+01	-.6754E-01
4.4	.6700E+00	.2164E+01	-.6140E-01
4.5	.6676E+00	.2231E+01	-.5512E-01
4.6	.6648E+00	.2298E+01	-.4876E-01
4.7	.6616E+00	.2364E+01	-.4235E-01
4.8	.6581E+00	.2430E+01	-.3594E-01
4.9	.6543E+00	.2496E+01	-.2956E-01
5.0	.6502E+00	.2561E+01	-.2325E-01
5.1	.6459E+00	.2626E+01	-.1706E-01
5.2	.6415E+00	.2690E+01	-.1101E-01
5.3	.6369E+00	.2754E+01	-.5145E-02
5.4	.6322E+00	.2818E+01	.5097E-03
5.5	.6274E+00	.2881E+01	.5924E-02
5.6	.6227E+00	.2943E+01	.1107E-01
5.7	.6179E+00	.3005E+01	.1592E-01
5.8	.6132E+00	.3067E+01	.2046E-01
5.9	.6086E+00	.3128E+01	.2466E-01
6.0	.6042E+00	.3188E+01	.2852E-01
6.1	.5999E+00	.3249E+01	.3201E-01
6.2	.5957E+00	.3308E+01	.3513E-01
6.3	.5918E+00	.3368E+01	.3786E-01
6.4	.5881E+00	.3427E+01	.4021E-01
6.5	.5847E+00	.3485E+01	.4218E-01
6.6	.5815E+00	.3544E+01	.4375E-01
6.7	.5787E+00	.3602E+01	.4495E-01
6.8	.5761E+00	.3659E+01	.4577E-01
6.9	.5739E+00	.3717E+01	.4623E-01

7.0	.5720E+00	.3774E+01	.4633E-01
7.1	.5704E+00	.3831E+01	.4609E-01
7.2	.5691E+00	.3888E+01	.4552E-01
7.3	.5682E+00	.3945E+01	.4465E-01
7.4	.5676E+00	.4002E+01	.4349E-01
7.5	.5674E+00	.4059E+01	.4206E-01
7.6	.5674E+00	.4115E+01	.4039E-01
7.7	.5678E+00	.4172E+01	.3851E-01
7.8	.5665E+00	.4229E+01	.3642E-01
7.9	.5695E+00	.4286E+01	.3417E-01
8.0	.5707E+00	.4343E+01	.3178E-01
8.1	.5722E+00	.4400E+01	.2927E-01
8.2	.5740E+00	.4457E+01	.2667E-01
8.3	.5760E+00	.4515E+01	.2401E-01
8.4	.5781E+00	.4573E+01	.2131E-01
8.5	.5805E+00	.4630E+01	.1861E-01
8.6	.5830E+00	.4689E+01	.1592E-01
8.7	.5857E+00	.4747E+01	.1327E-01
8.8	.5884E+00	.4806E+01	.1069E-01
8.9	.5913E+00	.4865E+01	.8196E-02
9.0	.5942E+00	.4924E+01	.5813E-02
9.1	.5972E+00	.4984E+01	.3560E-02
9.2	.6001E+00	.5043E+01	.1457E-02
9.3	.6031E+00	.5104E+01	-.4786E-03
9.4	.6060E+00	.5164E+01	-.2231E-02
9.5	.6089E+00	.5225E+01	-.3787E-02
9.6	.6118E+00	.5286E+01	-.5133E-02
9.7	.6145E+00	.5347E+01	-.6258E-02
9.8	.6171E+00	.5409E+01	-.7155E-02
9.9	.6196E+00	.5471E+01	-.7817E-02
10.0	.6220E+00	.5533E+01	-.8237E-02
10.1	.6242E+00	.5595E+01	-.8414E-02
10.2	.6263E+00	.5658E+01	-.8346E-02
10.3	.6281E+00	.5720E+01	-.8033E-02
10.4	.6298E+00	.5783E+01	-.7478E-02
10.5	.6313E+00	.5846E+01	-.6685E-02
10.6	.6325E+00	.5909E+01	-.5660E-02
10.7	.6336E+00	.5973E+01	-.4409E-02
10.8	.6344E+00	.6036E+01	-.2941E-02
10.9	.6350E+00	.6100E+01	-.1267E-02
11.0	.6354E+00	.6163E+01	.6037E-03
11.1	.6356E+00	.6227E+01	.2657E-02
11.2	.6356E+00	.6290E+01	.4680E-02

POSC/PAV = .7134E-01 DBMAX = .1189E+00 ANGLE P/B = .1437E+03
 PSIP = -.1920E+03 PSIB = -.3357E+03 KO/KSS = .1609E+00
 KP = 0. KB = .3550E+01 PHI OSC/PHI AV = .7184E-01
 KPR = -.5128E+00 KTR = -.4543E-01 PSIBP = .1225E+03
 KPS = .6231E+00 K3S = -.3611E+01 P2/P1 = .8409E+00
 MKPPDR = .1128E+00 MKBPJR = .1166E+00

RUN NO. 2-B RUDDER NUMERATOR ROOTS

SIDESLIP TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0 0.0
 .10910-01
 -.90960+00
 -.22070+02

1/TB1 = -.10913E-01 1/TB2 = .90960E+00 1/TB3 = .22098E+02

AB = .2391E-01 BB = .9491E+00 CB = -.719E+00 DB = -.5216E-02

ROLL ANGLE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.
 .21490+01
 -.13650+01

1/TP1 = 0. 1/TP2 = -.214901E+01 1/TP3 = .136454E+01

AP = .1610E+00 BP = -.1263E+00 CP = -.723E+00 DP = 0.

YAW RATE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0 0.0
 -.21500-01 -.19990+00
 -.21500-01 .19990+00
 -.93070+00

ZR = .106897E+00 WR = .201093E+00 1/TR = .930702E+00

AR = -.52783E+00 BR = -.51400E+00 CR = -.42469E-01 DR = -.19857E-01

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.7994E-14	0.	.2668E-08
.1	.1448E-01	.7513E-03	.4905E-02
.2	.2555E-01	.2782E-02	.1495E-01
.3	.3305E-01	.5743E-02	.2973E-01
.4	.3604E-01	.9269E-02	.4908E-01
.5	.3684E-01	.1299E-01	.7271E-01
.6	.3300E-01	.1651E-01	.1003E+00
.7	.2532E-01	.1946E-01	.1316E+00
.8	.1383E-01	.2145E-01	.1663E+00
.9	-.1391E-02	.2210E-01	.2040E+00
1.0	-.2023E-01	.2105E-01	.2443E+00
1.1	-.4256E-01	.1794E-01	.2869E+00
1.2	-.5820E-01	.1242E-01	.3313E+00
1.3	-.9696E-01	.4191E-02	.3773E+00
1.4	-.1286E+00	-.7065E-02	.4245E+00
1.5	-.1629E+00	-.2162E-01	.4724E+00
1.6	-.1996E+00	-.3973E-01	.5206E+00
1.7	-.2385E+00	-.6162E-01	.5689E+00
1.8	-.2731E+00	-.8749E-01	.6168E+00
1.9	-.3213E+00	-.1175E+00	.6641E+00
2.0	-.3648E+00	-.1518E+00	.7103E+00
2.1	-.4091E+00	-.1905E+00	.7551E+00
2.2	-.4540E+00	-.2336E+00	.7983E+00
2.3	-.4993E+00	-.2813E+00	.8396E+00
2.4	-.5445E+00	-.3335E+00	.8787E+00
2.5	-.5894E+00	-.3902E+00	.9153E+00
2.6	-.6336E+00	-.4513E+00	.9494E+00
2.7	-.6770E+00	-.5169E+00	.9805E+00
2.8	-.7191E+00	-.5867E+00	.1009E+01
2.9	-.7598E+00	-.6607E+00	.1034E+01
3.0	-.7989E+00	-.7386E+00	.1056E+01
3.1	-.8359E+00	-.8204E+00	.1074E+01
3.2	-.8709E+00	-.9057E+00	.1089E+01
3.3	-.9035E+00	-.9945E+00	.1101E+01
3.4	-.9337E+00	-.1086E+01	.1109E+01
3.5	-.9612E+00	-.1181E+01	.1113E+01
3.6	-.9859E+00	-.1278E+01	.1115E+01
3.7	-.1008E+01	-.1378E+01	.1113E+01
3.8	-.1027E+01	-.1480E+01	.1108E+01
3.9	-.1042E+01	-.1583E+01	.1099E+01
4.0	-.1056E+01	-.1688E+01	.1088E+01
4.1	-.1066E+01	-.1794E+01	.1074E+01
4.2	-.1072E+01	-.1901E+01	.1057E+01
4.3	-.1075E+01	-.2009E+01	.1038E+01
4.4	-.1076E+01	-.2116E+01	.1016E+01
4.5	-.1074E+01	-.2224E+01	.9923E+00
4.6	-.1068E+01	-.2331E+01	.9666E+00
4.7	-.1061E+01	-.2437E+01	.9393E+00
4.8	-.1050E+01	-.2543E+01	.9105E+00
4.9	-.1037E+01	-.2647E+01	.8806E+00
5.0	-.1022E+01	-.2750E+01	.8497E+00
5.1	-.1005E+01	-.2852E+01	.8180E+00
5.2	-.9854E+00	-.2951E+01	.7859E+00

5.3	-.9643E+00	-.3049E+01	.7535E+00
5.4	-.9416E+00	-.3144E+01	.7212E+00
5.5	-.917E+00	-.3237E+01	.6890E+00
5.6	-.8924E+00	-.3327E+01	.6573E+00
5.7	-.8662E+00	-.3415E+01	.6262E+00
5.8	-.8393E+00	-.3501E+01	.5960E+00
5.9	-.8118E+00	-.3583E+01	.5668E+00
6.0	-.7841E+00	-.3663E+01	.5389E+00
6.1	-.7562E+00	-.3740E+01	.5123E+00
6.2	-.7284E+00	-.3814E+01	.4873E+00
6.3	-.7009E+00	-.3886E+01	.4640E+00
6.4	-.6739E+00	-.3954E+01	.4424E+00
6.5	-.6475E+00	-.4020E+01	.4228E+00
6.6	-.6226E+00	-.4084E+01	.4051E+00
6.7	-.5974E+00	-.4145E+01	.3896E+00
6.8	-.5741E+00	-.4203E+01	.3761E+00
6.9	-.5520E+00	-.4260E+01	.3648E+00
7.0	-.5313E+00	-.4314E+01	.3557E+00
7.1	-.5122E+00	-.4366E+01	.3488E+00
7.2	-.4947E+00	-.4416E+01	.3441E+00
7.3	-.4789E+00	-.4465E+01	.3415E+00
7.4	-.4649E+00	-.4512E+01	.3410E+00
7.5	-.4528E+00	-.4558E+01	.3426E+00
7.6	-.4426E+00	-.4603E+01	.3462E+00
7.7	-.4342E+00	-.4647E+01	.3517E+00
7.8	-.4278E+00	-.4690E+01	.3591E+00
7.9	-.4234E+00	-.4732E+01	.3681E+00
8.0	-.4208E+00	-.4774E+01	.3788E+00
8.1	-.4202E+00	-.4816E+01	.3909E+00
8.2	-.4213E+00	-.4859E+01	.4044E+00

RUN NO. 2-B COUPLING NUMERATOR ROOTS

PHI TO AILERON, BETA TO RUDDER

1/TP3 = 0.

APB = .117778D-01 BPB = .268673D+00 CP3 = 0.

PHI TO AILERON, PSI TO RUDDER

1/TPP = .911149E-01

APP = -.268013E+00 BPP = -.244200E-01

PSI TO AILERON, BETA TO RUDDER

ZPSB = .160257E-01 WPSB = .312289E+01

APSB = .118653D-02 BPSB = .118773D-03 CPSB = .115725D-01

PHI TO AILERON, ACCELERATION TO RUDDER

1/TPAY1 = -.141223E+01 1/TPAY2 = .146811E+01 1/TPAY3 = 0.

APAY = .875030D+01 BPAY = .488484D+00 CPAY = -.181440D+02 DPAY = 0.

PSI TO AILERON, ACCELERATION TO RUDDER

ZPSAY = .525242E+00 WPSAY = .977926E+00 1/TPSAY = .514526E+00

APSA Y = .881560D+00 BPSAY = .882483D-01 CPSAY = .252570D-13 DPSAY = -.783441D+00

ACCELERATION TO AILERON, BETA TO RUDDER

ZAYB = .598150E-01 WAYB = .441914E+01 1/TAYB = 0.

AA YB = .881650D+00 BAYB = .466103D+00 CAYB = .172179D+02 DAYB = .172179D+02

PLEASE RETURN PAPER

REFERENCES

1. "Dynamics of the Airframe," Bureau of Aeronautics, Department of Navy Report No. AE-61-4II, September 1952.
2. I.L. Ashkenas and D.T. Mc Ruer, "Approximate Airframe Transfer Functions and Application to Single Sensor Control Systems," WADC Technical Report 58-82, June 1958.
3. D.T. Mc Ruer, I.L. Ashkenas, and C.L. Guerre, Systems Technology, Inc., "An Analysis View of Longitudinal Flying Qualities," WADC Technical Report 60-43, January 1960.
4. W.R. Kolk, Modern Flight Dynamics, Prentice-Hall, Inc., Englewood Cliff, N.J., 1961.
5. R.L. Sands, "Lateral-Directional Dynamic Stability Requirements of MIL-F-8785A Including a Stick Fixed, 3 Degree of Freedom, Lateral-Directional Dynamic Stability Digital Computer Program," Mc Donnell Engineering Note 682, May 1969.
6. C.R. Wylie, Jr., Advanced Mathematics for Engineers and Scientists, second edition, Mc Graw-Hill Book Company, Inc., New York, 1960.
7. D.T. Mc Ruer, I.L. Ashkenas, and D. Graham, Aircraft Dynamics and Automatic Control, Princeton University Press, 1973.
8. T.S. Durand and R.J. Wasicko, Systems Technology, Inc., "An Analysis of Carrier Landing," AIAA Paper No. 65-791, Aircraft Design and Technology Meeting, November 15-18, 1965, Los Angeles, California.