NASA
Technical
Paper
2165

c.2

June 1984

STABCAR—A Program for Finding Characteristic Roots of Systems Having Transcendental Stability Matrices

William M. Adams, Jr.,
Sherwood H. Tiffany,
Jerry R. Newsom,
and Ellwood L. Peele

NASA

NASA
Technical
Paper
2165

1984

STABCAR—A Program for Finding Characteristic Roots of Systems Having Transcendental Stability Matrices

William M. Adams, Jr.,
Sherwood H. Tiffany,
Jerry R. Newsom,
and Ellwood L. Peele

Langley Research Center
Hampton, Virginia

NASA

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

CONTENTS

SYMBOLS

A matrix whose eigenvalues are found by using matrix iteration approach (eqs. (18))

$A_{0_j},\ldots,A_{(n_{\ell j}+2)_j}$ column matrices in p-plane approximation to jth column of generalized aerodynamic forces (eq. (3))

$a_{Ad_{1i}},\ldots,a_{Ad_{(nAd_i)i}}$ denominator polynomial coefficients in transfer function for ith actuator (eq. (12))

$a_{An_{1i}},\ldots,a_{An_{(nAn_i)i}}$ numerator polynomial coefficients in transfer function for ith actuator (eq. (12))

$a_{Fd_{1i}},\ldots,a_{Fd_{(nFd_i)i}}$ denominator polynomial coefficients in ith filter transfer function shown in equation (10)

$a_{Fn_{1i}},\ldots,a_{Fn_{(nFn_i)i}}$ numerator polynomial coefficient in ith filter transfer function shown in equation (10)

$a_{Sd_{1j}},\ldots,a_{Sd_{(nSd_j)j}}$ denominator polynomial coefficients in transfer function for jth sensor (eq. (8))

$a_{Sn_{1j}},\ldots,a_{Sn_{(nSn_j)j}}$ numerator polynomial coefficients in transfer function for jth sensor (eq. (8))

B_n matrix defined by equation (A6)

b reference length used in computation of generalized aerodynamic forces

b_j column vector ($n_{\ell j} \times 1$) of constants in denominator terms of p-plane fit (eq. (3))

C,C_0 matrices defining constraining relationships imposed on p-plane coefficients (eq. (A4))

c mean aerodynamic chord of wing

D complex determinant of characteristic matrix of homogeneous equations (eq. (1) or (2))

D' modified determinant (eqs. (14))

d common denominator of nonscheduled elements of transfer matrix relating actuator outputs to sensor inputs (see eq. (C3))

$e_{ij},e_{c_{ij}},e_n$ error functions associated with least-squares p-plane fit to unsteady aerodynamic forces (eqs. (A5), (A7), and (A8))

F_D matrix of modal coefficients converting hinge moment outputs produced by actuator to generalized forces (eq. (1))

f_n vector of natural frequencies in vacuum, Hz

G gain matrix (eq. (9))

$g_i(s)$ $= \sqrt{-1}\, g_{s_i}$, $\frac{s}{|s|} g_{s_i}$, or $g_{s_i} \frac{s}{\omega_{n_i}}$

g_s vector of structural damping coefficients

H matrix relating sensor input to generalized coordinates (eq. (7))

H_E matrix relating actuator elongation to generalized coordinates (eq. (1))

h altitude

J_j integer; J_j = 0, 1, or 2 defines sensor j as position, rate, or acceleration sensor, respectively (eq. (8))

K generalized stiffness matrix

K_0, K_1, K_2 gain for integral, proportional, or derivative filter, respectively

k reduced frequency, $\omega b/U$

M generalized mass matrix

N N_{ij}/d is transfer function relating output of actuator i to input sensor j (see eq. (C4))

N_{Ma} Mach number

nAn_i, nAd_i number of numerator and denominator polynomial coefficients, respectively, in transfer function for ith actuator

nFn_i, nFd_i number of numerator and denominator polynomial coefficients, respectively, in transfer function associated with filter i (eq. (10))

nSn_i, nSd_i number of numerator and denominator polynomial coefficients, respectively, in transfer function for sensor i (nSn_i contains J_i in addition to basic sensor dynamics)

nkk_j number of values of k for which $Q_j(k)$ will be employed in determining least-squares fit for p-plane approximation to jth column of aerodynamic forces (eq. (A7))

$n_{\ell j}$ number of high pass terms in p-plane approximation for jth column of unsteady aerodynamic forces

$n_{\Delta V_c}, n_{\Delta V_s}$ number of subintervals to divide interval (V_i, V_{i+1}) into when having convergence difficulties or locating a point of change in stability, respectively

$n_\xi, n_\delta, n_r, n_s, n_k$ number of generalized coordinates, control surfaces, rigid-body degrees of freedom, sensors, and reduced frequencies for which aerodynamic forces have been computed, respectively

$\Delta P_j(x,y;N_{Ma},p)$ difference in pressure between upper and lower surface of lifting body at point (x,y) due to motion in jth mode

p nondimensional Laplace variable, sb/U

Q matrix of generalized aerodynamic forces

$\hat{Q}$ p-plane approximation of Q (eq. (3))

$Q(k_i)$ tabular value in unsteady aerodynamic force matrix at a reduced frequency of k_i

$Q'(k)$ obtained by interpolation given $Q'(k_i) = \text{Im}\,\frac{Q(k_i)}{k_i}$, $i = 1, \ldots, n_k$ (if $k_1 = 0$, $Q'(k_1) \equiv Q'(k_2)$)

q dynamic pressure, $\rho U^2/2$

$R_{x'},R_{y'}$ angular rotations about x' and y' axes, respectively (eqs. (4) and (6))

r column vector of sensor inputs

r' column vector of sensor outputs

r_D column vector of linear displacement sensor inputs

r_θ column vector of angular displacement sensor inputs

S surface area

s Laplace variable

$\hat{s}_{i_n}$ nth estimate of ith characteristic root

$\left[T_A\right]$ diagonal matrix of actuator transfer functions

$\left[T_{HE}\right]$ transfer matrix relating actuator hinge moment output to actuator elongation

T_{HS} transfer matrix relating hinge moments produced by actuator to input to sensors (s^{J_j} is also included in T_{HM}, see definition of J_j)

T_L transfer matrix relating actuator inputs to sensor outputs

T_L' T_L excluding gain, phase error, and scheduling components (eq. (9))

$\left(T_{LS}(s,N_{Ma},q)\right)_{ij}$ scheduled portion of compensation between jth sensor and ith control (eq. (9))

$\left[T_S\right]$ diagonal transfer matrix relating sensor outputs to sensor inputs (s^{J_j} is also included in T_{HM}, see definition of J_j)

T_δ transfer matrix relating control surface position to sensor inputs (s^{J_j} is also included in T_δ, see definition of J_j)

t time variable

U aircraft speed

U_D aircraft design dive speed

U_F open-loop flutter velocity

(x,y,z) position triple defining three-dimensional coordinate system

(x',y',z') coordinate system rotated and translated from (x,y,z) (eq. (5))

x_{cg} x coordinate of center of gravity

$(x_0,y_0,0)$ origin of (x',y',z') coordinate system with respect to (x,y,z)

$z_i(x,y)$ displacement in ith mode at $(x,y,0)$

α_n scalar defining step from s_{i_n} to $s_{i_{n+1}}$ for which $\left|D'_{i_{n+1}}\right| \leq \left|D'_{i_n}\right|$ (eq. (15))

δ column vector of control rotations, $n_\delta \times 1$

δ_A, δ_L actuator and control logic output, respectively, $n_\delta \times 1$

ε criterion for convergence upon characteristic root (eqs. (16))

ζ damping ratio associated with complex conjugate pair of roots

ζ_F damping ratio associated with notch filter

θ_n normalizing angle in definition of rigid-body pitch mode shape (see section entitled "Sensor input definition"

Λ sweep angle of elastic axis, positive clockwise (eq. (5))

λ_j jth characteristic root

$\tilde{\lambda}_j$ column vector of Lagrange multipliers associated with constraints imposed upon coefficients in p-plane approximation to jth column of aerodynamic forces (eq. (A8))

ξ column vector of generalized coordinates, $n_\xi \times 1$

ρ density

σ mass surface density of flat-plate idealization of airplane

τ_1, τ_2 time constants in first-order lead-lag (lag-lead) filter

ϕ_{ij} phase error introduced in compensation between jth sensor and ith actuator

ω frequency of oscillation

ω_{n_F} natural frequency associated with notch filter

ω_{n_i} natural frequency of ith mode in vacuum

Subscripts:

F filter

S sensor

Abbreviations:

freq. frequency

nos. numbers

Red. reduced

Special mathematical symbol:

$\lceil \ \rfloor$ diagonal matrix

SUMMARY

STABCAR is a computer program that can be used to determine the characteristic roots of flexible, actively controlled aircraft, including the effects of unsteady aerodynamics. A modal formulation is employed to characterize the aircraft. The control system representation is input as a matrix of transfer functions. Roots are determined through the use of determinant iteration utilizing either oscillatory or approximate fully unsteady aerodynamic forces. Provisions are included to allow automatic variation of velocity, density, altitude, or feedback gains.

The mathematical model employed for the aircraft is described, a flowchart is provided, detailed description is given of the function of key program elements and parameters that direct program operation, program inputs are defined, and sample inputs and outputs are presented.

STABCAR can be executed in either a batch or an interactive mode. It is written specifically for use on CDC® CYBER 175 equipment. Modification would be required for operation on other equipment.

INTRODUCTION

The equations of motion of flexible aircraft contain aerodynamic force terms which, when expressed in the Laplace domain, are transcendental functions of the Laplace variable. Determination of the characteristic roots of the system, therefore, requires the use of iterative or graphical procedures. Alternatively, one may approximate the aerodynamic force terms with rational functions of the Laplace variable and convert the equations of motion to a standard eigenvalue problem (refs. 1 and 2).

This paper describes a computer program which utilizes the method of determinant iteration to solve for the characteristic roots of flexible, actively controlled aircraft. References 3 through 6 describe other programs for iterative determination of characteristic roots. Reference 3, which was closely followed in the development of STABCAR, describes results obtained with a code developed by a major airframe manufacturer; reference 4 describes methods employed in England. Neither code is readily available and no reference is made to a user's guide for the programs. References 5 and 6, however, describe NASTRAN®, a publically available program. The NASTRAN program was developed for batch-type operations. In contrast, STABCAR, described herein, is tailored for interactive execution and contains a number of features which facilitate the interactive study of the effects upon stability of variations in control-system parameters.

STABCAR is a module of the ISAC (Interaction of Structures, Aerodynamics, and Controls) program (ref. 7). The present paper is the first which documents a major segment of the ISAC program. Future papers which document other segments may be found by employing ISAC as a key word in a library search.

The equations of motion are outlined by employing a modal representation of the aircraft. Aerodynamic forces for either purely oscillatory or approximate fully unsteady motion can be considered. General linear control laws can be analyzed either by direct input of the elements of a transfer matrix or by construction utilizing selectable built-in filter types.

Considerable effort has been directed toward the incorporation of features that allow the user to utilize his time efficiently. Interactive program execution is an option. Characteristic root variation with altitude, density, velocity, or control system gains can be performed in an automated manner with the results presented graphically.

The STABCAR program and all user options are described in detail in this report. The functions of major program elements are specified, and key parameters are identified that direct the input to, the computations in, and the results from each major subroutine. Flowcharts and a detailed description of input parameters are also provided.

Results are presented and discussed which illustrate major program options. Characteristic root variations are shown with and without feedback control as a function of parameters dependent on flight conditions. Root loci, as a function of variation in control parameters, are also shown; comparisons of characteristic roots obtained by using approximate fully unsteady and oscillatory aerodynamics are presented.

The inputs for a series of sample cases are presented. The cases correspond to the graphical results that are shown. These sample cases provide explicit examples of the inputs required to exercise the major capabilities of STABCAR.

STABCAR has been written specifically for efficient operation on CDC® CYBER 175 equipment. Revisions are required to run this program on other equipment.

OVERVIEW

In STABCAR, stability characteristics are investigated. The dynamic elements that may be included are shown in figure 1. Mathematical descriptions of each of the dynamic elements are presented in the next section. The element representing the aircraft includes rigid, elastic, and control degrees of freedom as well as unsteady aerodynamic forces. The unsteady aerodynamic forces cause the equations of motion in the Laplace domain to be transcendental functions of the Laplace variable. Multiple-input/multiple-output control laws can be studied; these can include dynamic compensation expressible in terms of a transfer matrix.

STABCAR has been developed to solve for characteristic roots of the stability matrix. Typically, a search is made for roots as a function of the variation of a parameter of interest such as altitude, density, velocity, or a feedback gain. This procedure is illustrated in figure 2. Since the stability matrix is a transcendental function of the Laplace variable, the characteristic roots are determined by an iterative procedure. Consequently, initial estimates of the characteristic roots are required. Methods employed in STABCAR for determining initial estimates are presented in a later section.

MATHEMATICAL REPRESENTATION AND SOLUTION TECHNIQUES

In this section, the mathematical description of STABCAR is outlined. The equations of motion are discussed as well as alternate options as to the form that can be chosen for the aerodynamic forces. A capability for interpolation of beam- or plate-type mode shapes to obtain sensor inputs is described. The manner in which the control system is included is described, and the techniques for characteristic root determination and major options for output of information are detailed.

Equations of Motion

The equations of motion consider only perturbations from a level equilibrium flight condition. The perturbed aircraft motion is represented in terms of a truncated set of rigid and elastic modes. The resulting equations, expressed in terms of generalized coordinates, masses, stiffnesses, dampings, and aerodynamic forces, are essentially the same as those developed in reference 8. The homogeneous part of these equations, expressed in the Laplace domain, are given in the following equation:

$$\left[Ms^2 + \left[g_i(s)\ K_{ii}\right] + K + qQ\left(N_{Ma},\frac{sb}{U}\right) - F_D\left(T_{HS}(s)\ H + T_{HE}(s)\ H_E\right)\right]\xi = \{0\} \quad (1)$$

where

M_{ij} element of generalized mass matrix, $\iint_S z_i(x,y)\ \sigma(x,y)\ z_j(x,y)\ dS$

K_{ij} element of generalized stiffness matrix; $K(x,y;u,v)$ is influence coefficient giving force at (x,y) due to unit displacement at (u,v), $\iint_{S_1} z_i(x,y) \iint_{S_2} K(x,y;u,v)\ z_j(u,v)\ dS_1\ dS_2$

$-Q_{ij}\left(N_{Ma},\frac{sb}{U}\right)$ element of generalized aerodynamic force matrix with $\Delta P_j(x,y;N_{Ma},p)$ being the pressure difference due to motion in jth mode, $\iint_S z_i(x,y)\frac{\Delta P_j(x,y;N_{Ma},p)}{q}\ dS$

$g_i(s)$ $= \sqrt{-1}\ g_{s_i}$, $\frac{s}{|s|}\ g_{s_i}$, or $g_{s_i}\frac{s}{\omega_{n_i}}$; user chooses desired form. If $\omega_{n_i} = 0$, g_i set to zero

g_{s_i} structural damping coefficient associated with ith mode

$[T_{HS}(s)]$ $n_\delta \times n_\xi$ matrix of transfer functions relating actuator hinge moment outputs to sensor inputs

$T_{HE}(s)$ $n_\delta \times n_\delta$ diagonal matrix of transfer functions relating actuator hinge moment outputs to actuator elongation

F_D $n_\xi \times n_\delta$ matrix of modal coefficients converting hinge moment outputs to generalized force

ξ vector, $n_\xi \times 1$, of rigid and elastic generalized coordinates

δ vector, $n_\delta \times 1$, of control-surface generalized coordinates

$z_i(x,y)$ ith mode shape, $z(x,y,t) = \sum_{i=1}^{n_\xi} z_i(x,y)\ \xi_i(t)$

$H_{ij}(x,y)$ linear or angular modal deflection contributing to input to sensor i resulting from unit deflection in jth generalized coordinate, $r = H\xi$ where H is $n_s \times n_\xi$

H_E $n_\delta \times n_\xi$ matrix of modal coefficients relating actuator elongation to generalized coordinates

An alternate form of the equations neglects aerodynamic hinge moments and hinge moments due to inertial coupling in which case the control surface deflections are

$$\delta = T_\delta(s)\ H\xi$$

The dimension of δ is $n_\delta \times 1$ and T_δ is a matrix of transfer functions relating control deflections to sensor inputs. This expression replaces the control rows of equation (1); the resulting expression is

$$\left[\begin{array}{c|c} M_{\xi\xi}s^2 + \left[g_i(s)\ K_{ii}\right] + K_{\xi\xi} + qQ_{\xi\xi}\left(N_{Ma},\frac{sb}{U}\right) & M_{\xi\delta}s^2 + qQ_{\xi\delta}\left(N_{Ma},\frac{sb}{U}\right) \\ \hline -T_\delta(s)\ H & I \end{array}\right]\begin{Bmatrix}\xi \\ \delta\end{Bmatrix} = \begin{Bmatrix}0 \\ 0\end{Bmatrix} \qquad (2)$$

Unsteady aerodynamic forces.- In equations (1) and (2), the aerodynamic forces are expressed as functions of Mach number and the complex variable $p = sb/U$. The programs currently available for production generation of aerodynamic forces can only compute these forces for $p = \sqrt{-1}\ b\omega/U$ (e.g., refs. 9 through 12). Two approaches are taken in STABCAR to circumvent this difficulty.

The first approach, known as the p-k approximation (refs. 3 through 5) is valid for lowly damped modes. Two options are implemented for this approximation:

(a) Form of reference 3

$$Q(p) = Q(sb/U) \approx Q(p = 0 + \sqrt{-1}\ k) = Q(k)$$

(b) British form (refs. 4 and 5)

$$Q(p) = Q(sb/U) \approx \mathrm{Re}(Q(k)) + pQ'(k)$$

$Q'(k)$ is obtained by interpolation of the matrix of functions $\{Q'(k_1),\ Q'(k_2),\ \ldots,\ Q'(k_{n_k})\}$ where $Q'(k_i) = \mathrm{Im}\left(\frac{Q(k_i)}{k_i}\right)$. If $k_1 = 0$, $Q'(k_1) \equiv Q'(k_2)$. The approximation becomes exact at the flutter point. The second approach, known as the p-p method (ref. 3), requires the aerodynamic forces as a function of p. In STABCAR, use is made of the concept of analytic continuation to develop an approximation to the aerodynamic forces for arbitrary motion in terms of known aerodynamic forces for oscillatory motion ($p = \sqrt{-1}\,k$). The form assumed for the jth column of the aerodynamic force matrix is

$$\hat{Q}_j(p) = A_{0_j} + A_{1_j} p + A_{2_j} p^2 + \sum_{\ell=1}^{n_{\ell j}} \frac{A_{(\ell+2)_j} p}{p + b_{\ell j}} \tag{3}$$

This expression has the same form as that employed in references 1 and 2. Determination of the polynomial coefficients is initiated by computation of $Q_j(\sqrt{-1}\,k)$ for a number of values of reduced frequency. This computation must be done by an external program such as those described in references 9 through 12. The user specifies the vector b_j and may impose linear constraints upon the coefficients in equation (3). For example, the A_0 columns associated with rigid-body linear displacements should be zero. The coefficients of each element of $\hat{Q}_j(p)$ are determined, subject to any imposed constraints, such that the errors between the curve fit (eq. (3)) and the tabular data are minimized in a least-squares sense. The least-squares-fit procedure and allowable constraints are described in appendix A.

References 13 and 14 present a theoretical basis for the curve-fit procedure and describe the asymptotic form $\hat{Q}_j(p)$ should take. The asymptotic limits are not enforced in STABCAR because constraining the curves to fit high frequency limits tends to degrade the fit in the limited frequency band where tabular data are available. Reference 15 describes the application of a constrained optimization procedure to optimally choose b_j in equation (3).

References 16 and 17 describe a program now under development which will allow computation of aerodynamic forces for transient motion where p has a nonzero real part. STABCAR can be readily modified to utilize such data. The required modification is to provide for interpolation of tabular aerodynamic force arrays as a function of two independent variables (the real and imaginary parts of p) rather than as a function of only the imaginary part of p.

Sensor input definition.- Sensor inputs in STABCAR are assumed to be linear combinations of the generalized coordinates. For linear displacements,

$$r_{D_i}(x,y) = \sum_{j=1}^{n_\xi} z_j(x,y)\,\xi_j$$

For angular displacements,

$$r_{\theta_i}(x,y) = \sum_{j=1}^{n_\xi} \frac{\partial z_j(x,y)}{\partial x}\,\xi_j$$

These expressions would be multiplied by s for a velocity type sensor and by s^2 for an accelerometer. The multiplication by the appropriate power of s is included within the computation of the effects of sensor dynamics. (See eq. (8).) Thus, in computing sensor inputs, the assumption is made that inertial axes are employed. The user may override this assumption by appropriately modifying the rigid-body portions of the mass, damping, and stiffness matrices.

The mode shapes or their derivatives at the sensor locations must be specified before the sensor inputs can be determined. Two options are included for defining the modal data for input to sensors. The first option is by direct input of the modal data required at the sensor locations. The second option is to input all the modal data (mode shapes and node-point locations) and interpolate for the modal amplitudes at the sensor locations. This option is attractive when performing sensor-location studies.

The remainder of this section describes the interpolation procedure. Vibration analyses, which use modelings ranging from simple beams to finite-element representations, calculate modal data at a number of grid points. In order to accommodate output from a variety of vibration analyses, it is necessary, in STABCAR, to arrange the modal data into subset divisions called structural sections. For aircraft configurations, the structural sections follow naturally as the wing, fuselage, horizontal tail, and so forth.

There are two types of interpolation methods in STABCAR corresponding to the two types of structural sections available in the program. The first is a displacement-twist interpolation method which is most commonly used when the mode shapes are generated from an elastic-axis beam model. The second is a surface-spline interpolation method (ref. 18) which is used when the mode shapes are generated from a finite-element "plate-type" model. The entire modal representation may consist of a combination of both types of structural sections.

For a zero-thickness idealization of a surface and a beam structural representation, the displacement in the ith mode normal to the surface can be written as

$$z_i(x,y) = z_i(x',y') = z_i(y') - x'R_{y'_i}(y') \tag{4}$$

where

(x,y,z) right-hand coordinate system with x positive aft and z positive up

(x',y',z') right-hand coordinate system with z' parallel to z and x' and y' swept from (x,y) by angle Λ

$z_i(x,y)$ deflection in ith mode at (x,y) normal to surface

$z_i(y')$ vertical bending in ith mode at $(x' = 0,y')$

$R_{y'_i}(y')$ rotation about y' axis in ith mode at $(x' = 0,y')$

Figure 3 shows the geometrical relationships for a flat surface parallel to the xy plane. The origin of the swept, translated axis system is at (x_0, y_0); Λ is the sweep angle (positive clockwise); and positive z is out of the page. The transformation from (x,y) to (x',y') is given by

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos\Lambda & -\sin\Lambda \\ \sin\Lambda & \cos\Lambda \end{bmatrix} \begin{Bmatrix} x - x_0 \\ y - y_0 \end{Bmatrix} \tag{5}$$

For angular displacements, the partial derivative of equation (4) is required and is given as follows:

$$\frac{\partial z_i}{\partial x} = \left[\frac{dz_i(y')}{dy'} - x' \frac{dR_{y_i'}(y')}{dy'}\right] \sin\Lambda - R_{y_i'} \cos\Lambda$$

or

$$\frac{\partial z_i}{\partial x} = \left[R_{x_i'}(y') - x' \frac{dR_{y_i'}(y')}{dy'}\right] \sin\Lambda - R_{y_i'}(y') \cos\Lambda \tag{6}$$

where $R_{x_i'}$ is rotation about x' (positive counterclockwise).

For "plate" type modes the modal data from a structural analysis are defined at points distributed over the structural section. Thus, there are two independent variables (x,y) and the interpolation is done by using the surface spline methods described in reference 18. Slope information (e.g., $\partial z_i(x,y)/\partial x$) is generated by interpolating for the derivative of the surface spline.

Rigid-body mode shapes may be defined by input by the user. In this case, the rigid-modal data must be treated as if they are elastic modes (i.e., $n_r = 0$ where n_r is the number of rigid modes). Alternatively, if the symmetric rigid-body mode shapes of vertical translation and pitch about the aircraft center of mass are desired, the user may set $n_r = 2$. Then the following built-in expressions are employed to compute the rigid-body contributions to sensor inputs:

Linear displacements:

$$z_1(x,y) = 1; \quad z_2(x,y) = -(x - x_{cg})\theta_n$$

Angular displacements:

$$\frac{dz_1}{dx}(x,y) = 0; \quad \frac{dz_2}{dx} = -\theta_n$$

Control-System Representation

The control system is divided into three parts: sensor, compensation (filtering), and actuator dynamics. The input-output relationships between these dynamic elements and their coupling with the aircraft dynamics are indicated schematically in figure 1 and explicitly in equations (1) and (2). In a subsequent discussion of these dynamic elements, the term "block" is used interchangeably with "transfer matrix."

Sensor dynamics options.- Input to the sensor dynamics can be either linear or angular position information expressed in terms of the generalized coordinates as

$$r = H\xi \tag{7}$$

The dimension of r is $n_s \times 1$. Within the sensor dynamics block each r_i can be transformed to a position, velocity, or acceleration input by multiplication by the appropriate power of the Laplace variable s. A given sensor output will be of the form

$$r_j' = \frac{a_{Sn_{1j}} + a_{Sn_{2j}} s + \dots + a_{Sn_{(nSn_j)j}} s^{nSn_j - 1}}{a_{Sd_{1j}} + a_{Sd_{2j}} s + \dots + a_{Sd_{(nSd_j)j}} s^{nSd_j - 1}} s^{J_j} r_i = T_{Sj}(s)\, r_j \tag{8}$$

where $J_j = 0, 1,$ or 2 corresponds to a displacement, rate, or acceleration sensor, respectively. Default values are built into the program so that only J_j must be input if sensor dynamics are neglected.

Compensation options.- The transfer matrix T_L, of dimension $n_\delta \times n_s$, specifying compensation of the signals from the sensors prior to sending them to the actuators can be full with distinct dynamics in each element. The transfer matrix relationship between compensator outputs and compensator inputs (sensor outputs) is

$$\delta_L = T_L(s)\, r' \tag{9}$$

where

$$\left[T_L(s)\right]_{ij} = (G)_{ij}\left[T_L'(s)\right]_{ij}\left[T_{LS}(s,N_{Ma},q)\right]_{ij} e^{\sqrt{-1}\,\phi_{ij}}$$

Here $(G)_{ij}$ is a gain, each $[T_L(s)]_{ij}$ may contain dynamic compensation elements but has a default value of 1. Alternate values for $[T_L(s)]_{ij}$ may be defined either by direct input of polynomial coefficients or by utilization of the following built-in filter types:

1. Notch filter:

$$\frac{1 + \left(s^2/\omega_{n_F}^2\right)}{1 + 2s\left(\zeta_F/\omega_{n_F}\right) + \left(s^2/\omega_{n_F}^2\right)}$$

where ω_{n_F} and ζ_F are the natural frequency and damping ratio of the notch filter

2. Integral filter:

K_0/s

3. Proportional plus derivative filter:

$K_1 + K_2 s$

4. Lead-lag (lag-lead) filter:

$$\frac{1 + \tau_1 s}{1 + \tau_2 s}$$

5. Polynomial filter:

$$\frac{a_{Fn_{1i}} + a_{Fn_{2i}} s + \ldots + a_{Fn_{(nFn_i)i}} s^{nFn_i - 1}}{a_{Fd_{1i}} + a_{Fd_{2i}} s + \ldots + a_{Fd_{(nFd_i)i}} s^{nFd_i - 1}} \tag{10}$$

Any combination of these filter types may be utilized including repetitive use of the same filter type with the same or distinct constants. Additional filter options are as follows:

6. Scheduling:

The capability for providing scheduled compensation elements $T_{LS}(s,N_{Ma},q)_{ij}$ is also included. For each control-sensor pair, it may be desirable to allow the control law to vary as a function of Mach number and dynamic pressure. This variation is accomplished by providing a call to a subroutine which has available for use control-sensor pair identification, the value of the Laplace variable s, Mach number, and dynamic pressure. The subroutine, which defines the desired scheduling, must be supplied by the user. This procedure is illustrated in appendix B for a specific scheduling law.

7. Phase error:

An additional feature included is the following option of introducing a phase error ϕ_{ij} into the compensation for one control-sensor pair:

$$e^{\sqrt{-1}\,\phi_{ij}} \tag{11}$$

This option allows one to study sensitivity to phase errors.

Actuator dynamics options.- The transfer-function matrix relating inputs to and outputs from each actuator is diagonal. Its default value is the identity matrix. If actuator dynamics are to be considered, the following numerator and denominator polynomial coefficients are input:

$$(\delta_A)_i = \frac{a_{An_{1i}} + a_{An_{2i}} s + \ldots + a_{An_{(nAn_i)i}} s^{nAn_i - 1}}{a_{Ad_{1i}} + a_{Ad_{2i}} s + \ldots + a_{Ad_{(nAd_i)i}} s^{nAd_i - 1}} (\delta_L)_i \tag{12}$$

or

$$\left(\delta_A\right) = \left[T_A\right] \delta_L$$

where the dimension of δ_A is $n_\delta \times 1$.

Combination of sensor, compensator, and actuator matrices.- The relationship between δ_A and r is

$$\delta_A = \left[T_A\right]\left[T_L\right]\left[T_S\right] r = \frac{N(s)}{d(s)}\{r\} \tag{13}$$

If aerodynamic and mass coupling hinge moments are considered, the resulting transfer matrix corresponds to T_{HS} in equation (1); otherwise it corresponds to T_δ in equation (2). The polynomial $d(s)$ in equation (13) is the common denominator of $T_L'(s)$ (see eq. (9)). This factor can be cleared from the denominator in equation (1) or (2). Appendix C describes this option.

Characteristic Root Determination

The primary objective of STABCAR is to determine characteristic roots of an aircraft represented mathematically by the transcendental matrix equation (eq. (1) or (2)). Roots can be found as a function of variations in velocity, density, altitude, or control parameters.

Determinant iteration.- Characteristic root determination is accomplished mode by mode with determinant iteration given initial estimates for the roots. The roots to be traced are specified by input. Methods for obtaining the initial estimates are described after the following discussion of the procedure for determining a characteristic root.

An attempt is made to find the ith characteristic root in a manner which avoids convergence upon a previously determined root, $j < i$. This is accomplished by determining the zeros of

$$D'(s) = D(s) \qquad (i = 1) \tag{14a}$$

or

$$D'(s) = \frac{D(s)}{\prod_{j=1}^{i-1} (s - \lambda_j)} \qquad (i > 1) \tag{14b}$$

where

$D(s)$ value of complex determinant of stability matrix in equation (1) or (2)

λ_j previously determined characteristic root

Let $\hat{s}_{i_{n+1}}$ denote the desired $n + 1$ estimate of the zero of equations (14) corresponding to the ith characteristic root and $\hat{s}_{i_n}$ denote a previously determined estimate; therefore, $D'_{i_{n+1}}$ or D'_{i_n} is the value of $D'(s)$ at $\hat{s}_{i_{n+1}}$ or $\hat{s}_{i_n}$, respectively. There exists an $\alpha_n \geqslant 0$ such that, for

$$\hat{s}_{i_{n+1}} = \hat{s}_{i_n} - \frac{\alpha_n D'_{i_n}}{dD'_{i_n}/ds} \tag{15}$$

$$\left|D'_{i_{n+1}}\right| \leqslant \left|D'_{i_n}\right|$$

In the search for λ_i, α_n is constrained to be $\leqslant 0.1$ in order to reduce the likelihood of jumping to a point sufficiently far away from λ_i as to cause convergence upon another root. The general procedure is to solve equation (15) repetitively until $\hat{s}_{i_{n+1}}$ differs from $\hat{s}_{i_n}$ by less than a tolerence ε specified by the user.

That is, if

$$\left|\hat{s}_{i_{n+1}} - \hat{s}_{i_n}\right| < \varepsilon \qquad \left(\left|\hat{s}_{i_n}\right| \leqslant 1\right) \tag{16a}$$

or

$$\frac{\left|\hat{s}_{i_{n+1}} - \hat{s}_{i_n}\right|}{\left|\hat{s}_{i_n}\right|} < \varepsilon \qquad \left(\left|\hat{s}_{i_n}\right| > 1\right) \tag{16b}$$

convergence has occurred and $\lambda_i \equiv \hat{s}_{i_{n+1}}$.

Initial estimates for roots.- The algorithm for determining characteristic roots described in the previous section requires both function (eqs. (14)) and slope (eq. (15)) information. Thus, two initial estimates are required for each root that is to be traced. Three options are available for specifying the initial estimates:

1. Direct user input:

2. Computations based upon system natural frequencies of vibration in vacuum:

$$\left.\begin{aligned} \hat{s}_{i_1} &= 0 + \sqrt{-1}\ \omega_{n_i} \\ \hat{s}_{i_2} &= (d_1 + \sqrt{-1}\ d_2)\omega_{n_i} \end{aligned}\right\} \tag{17}$$

where d_1 and d_2 are input parameters having default values of 0.01 and 1.0, respectively. This approximation works well for the elastic degrees of freedom at sufficiently low dynamic pressures.

3. Initial estimates by matrix iteration: In this approach the matrix equation (1) or (2) is expressed in the form

$$s^2 \xi = \mathbf{A}(s)\ \xi \tag{18a}$$

or

$$s^2 \begin{Bmatrix} \xi \\ \delta \end{Bmatrix} = \mathbf{A}(s) \begin{Bmatrix} \xi \\ \delta \end{Bmatrix} \tag{18b}$$

Appendix C shows how $\mathbf{A}(s)$ is obtained, and appendix D describes the matrix iteration method.

Prediction of $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$ at a new value of independent variable.- Subsequent predictions for $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$ at each new value of the independent variable are obtained by using an extrapolation of either a linear fit to the two preceding converged roots (after second iteration), a quadratic fit to the preceding three converged roots (after third iteration), or a cubic fit to the preceding four converged roots (after the fourth iteration). The type of extrapolation to be performed (linear, quadratic, or cubic), after the requisite number of iterations, is specified by input.

Logic during convergence difficulties.- Failure to converge upon a characteristic root occurs periodically when the root is changing rapidly as a function of the parameter being varied. When such a failure occurs, the matrix iteration approach is employed in an attempt to recover. The matrix iteration approach is started with the last successful estimate for the root. If this attempt is successful, normal determinant iteration is then resumed. If the attempted recovery by using matrix iteration fails, the following scheme is utilized:

1. Let V_i denote the value of the parameter being varied (independent variable) at the point the root was last successfully obtained. Let V_{i+1} denote the value of the independent variable at which convergence failed.

2. Divide the interval (V_i, V_{i+1}) into a user specified number of increments $n_{\Delta V_c}$.

3. Solve for the root by using determinant iteration at each intermediate point and the end point V_{i+1}.

If convergence is achieved at V_{i+1}, normal execution is resumed. If not, that particular root is deleted from the set being sought and normal execution is resumed for the remaining roots.

Precise determination of point at which change in stability occurs.- When the first change in stability is noted, the value of the parameter being varied which corresponds to zero for the real part of the root is automatically determined. The determination is accomplished by

1. Dividing the interval between V_{i+1}, where a change in stability is noted, and V_i into a user specified number of increments $n_{\Delta V_s}$

2. Evaluating the characteristic root at all intermediate points

3. Performing an interpolation with these data to find the precise value of V at which the real part of the characteristic root is zero

Matched-point computations.- Two options for automatic matched-point computations are provided. Each seeks matched points at a particular input Mach number. A matched point is one for which the density, Mach number, and velocity are consistent with actual physical conditions (e.g., atmospheric conditions or tunnel operating conditions). Characteristic roots are computed and their corresponding root loci or damping ratios and natural frequencies can be plotted as a function of the parameter being varied. The point at which the real part of a root becomes zero denotes a change in system stability. For an elastic mode, this corresponds to a matched flutter point. The two automated types of matched-point computations are

1. Wind-tunnel matched point (density variation at fixed Mach number):

   (a) Characteristic roots are examined to determine stability characteristics with velocity held fixed.

   (b) Outputs for this type run include damping ratios and natural frequencies or root loci as a function of either density or dynamic pressure.

   Density variation analyses correspond to the method of operation of some wind tunnels and are useful for correlating analytical and experimental results for scaled flutter models.

2. Atmospheric matched point (altitude variation at fixed Mach number):

   (a) Beginning at a high-altitude (low-q) condition, characteristic roots are determined at specific altitudes with the data of reference 19 to determine the corresponding density and speed of sound. This type run yields a matched flutter point, if one exists, for the specified Mach number.

   (b) Outputs for this type of run include damping ratios and natural frequencies or root loci as a function of either altitude or dynamic pressure.

PROGRAM DESCRIPTION

In this section, program features are outlined and the primary functions of major program modules are identified.

Features for Enhancement of Resource and User Efficiency

Considerable effort was expended to develop a code that makes efficient use of resources and allows the user to efficiently study the stability characteristics of flexible aircraft. The following features are a result of the effort.

Overlay structure.- STABCAR is an overlay program with two levels of overlays. The overlay structure is indicated in figure 4. Overlaying allowed modularization of the code which facilitates modifications affecting only one overlay. It also reduced the amount of code required in core at one time.

Variable dimensions.- All large arrays are variably dimensioned, with the dimension specified by input. This enables the user to either study a new larger problem or make efficient use of central memory when a smaller problem is to be investigated without making program modifications.

Dynamic storage allocation.- Within each overlay, the required computer memory is computed and updated at critical points during job execution. This dynamic allocation of storage results in more efficient utilization of central memory. Further description of this technique is given in reference 20.

Free-field reads.- Free-field reads are employed to input most of the array data (e.g., mode shapes, generalized masses, stiffnesses and dampings, and unsteady aerodynamic forces).

Model selection.- A subset of the model that has been input can be selected for study. This option allows one to consider specified degrees of freedom while ignoring others that are represented in a given set of data. The appropriate generalized masses, stiffnesses, aerodynamic forces, and so forth are automatically extracted.

Automated parameter variations.- Stability characteristics are typically studied as a function of variation of dynamic pressure or control-law parameters. Automatic variation of dynamic pressure can be specified by the user in three ways: (1) altitude variation, (2) density variation, or (3) velocity variation. Gain variations can also be performed in an automated fashion. The effects of variations of other parameters - for example, sensor locations - upon system stability can be examined by running several cases, each having a different value of the chosen parameter.

Graphical display of output.- Stability characteristics can be displayed graphically in two forms. The damping ratio and natural frequency corresponding to each characteristic root can be plotted, and characteristic root loci can be plotted as a function of any parameter which can be varied in an automated manner. In addition, plots can be made which show the variation of the unsteady aerodynamic force with reduced frequency. The aerodynamic-force plots can exhibit the data points input into STABCAR, curves resulting from interpolation, and curves resulting from a p-plane approximation to the input data. The Langley Graphics package employed to obtain graphics output is not included in STABCAR. A description is given in appendix E of the function of the routines in this package to facilitate substitutions of equivalent software.

Interactive execution.- Interactive execution of STABCAR is an option. Some advantages of interactive execution are the more rapid receipt of output, the immediate visibility of input errors, and the capability of early termination.

Restart capability.- Data that define the mathematical model being employed in the current case are stored on TAPE1, a random access file. The data include selected modal information and corresponding aerodynamic forces including a p-plane fit if it has been generated. In addition, if the plot control parameter is nonzero, characteristic roots are stored as a function of the parameter being varied along with the user-supplied case title for identification. Consequently, if TAPE1 is saved, it may be used in a subsequent case if the model selected is a subset of the previous case. Thus, for example, the p-plane coefficients need not be recomputed. The new case might be to extend the range of variation of a parameter, to delete redundant or erroneous characteristic root data, or to combine sets of characteristic root data into one composite plot. Examples are given in the sample cases which illustrate several of these possibilities.

Functional Description of Overlays

The program is structured with two levels of overlays. The first level (overlay MAIN) is an executive. A brief summary of the major functions of each overlay will be presented here. A more detailed description including the function of major subroutines and definition of key parameters which direct the computations is given in appendix F.

MAIN: This executive overlay directs the computations by calling other overlays based upon input parameters. Namelist data are input which define the type of studies to be performed.

INTERP: INTERP performs geometry-related computations needed for a surface spline fit if mode-shape data are for a plate. The data required in these computations are input here. The results of the computations are subsequently used in SENSOR to generate surface spline coefficients that fit the plate modal data.

SENSOR: This overlay determines surface spline coefficients for fitting plate-type modes. Beam-type modal data are also input here. Interpolation of beam and plate modal data is performed to obtain modal coefficients defining either deflections or slopes that would be measured by sensors at specified locations. These modal coefficients are stored for subsequent use and define the matrix H in equation (1) or (2).

MATINPT: Large data arrays such as the aerodynamic data, generalized masses, generalized stiffnesses, and control-system dynamics definitions are input. In addition, the selection feature is contained, which enables one to choose a mathematical model that is a subset of the model defined by a given set of data.

PPLANE: This overlay is called if a p-plane approximation to the aerodynamic forces is desired. The best coefficients are determined, in a least-squares sense, to employ in equation (3), given a set of aerodyamic force data defined over a range of reduced frequencies. These coefficients are stored for subsequent use in overlays PKFLUT and AEROPLT.

CONTROL: The matrix of transfer functions $T_A T'_L T_S$ is constructed. (See eqs. (9) and (13).) The $( )_{ij}$ element relates the output of actuator i to the input of sensor j excluding $(G)_{ij}$ and any scheduling or phase error. The polynomial coefficients for sensor, compensation, and actuator dynamics are constructed separately and then combined. Each element of the resulting matrix can then be multiplied by a distinct feedback gain and scheduling component in overlay PKFLUT.

PKFLUT: The matrix in equation (1) or (2) is formed. Characteristic roots are determined as a function of variation in altitude, density, velocity, or feedback gains and the resulting output is stored for possible subsequent plotting. By operating in a multicase mode, the effect of variation in other parameters such as actuator or compensator dynamics or sensor location can also be studied.

AEROPLT: Plots are constructed which show how the oscillatory aerodynamic forces vary with reduced frequency and depict how the p-plane approximation fits the data.

PKPLOT: Plots showing root loci resulting from variation in velocity, density, altitude, dynamic pressure, or feedback gains can be generated. Alternatively, damping ratios and natural frequencies can be plotted for specified modes as a function of any of these parameters. Namelist inputs are accepted which define the types of plots desired; scales; plot titles and which plot data files, among those available, should be combined to make a composite plot.

STABCAR Flowchart

A flowchart is presented in figure 5, which describes the computational flow in STABCAR. This flowchart, although relatively detailed, is not intended to describe either the effects of all input options or the function of all program subroutines.

INPUT REQUIREMENTS

Description of Input Files

In this section STABCAR input requirements are defined. The documentation of the input requirements is presented in tables I through V. Therein, the types of data are outlined; a detailed description of the inputs is given which defines the individual input variables along with case-dependent conditions that determine whether certain of the data are required; and the input files that contain the data are identified. The correspondence between the FORTRAN names and the algebraic names of the variables is also indicated in these tables.

TAPE2 contains data which define problem size, user options, starting estimates, control system definition, and so forth. These data, primarily namelist, are outlined in table I. Table II contains a detailed description of the TAPE2 inputs. If ICASE is positive, the run is to be an interactive one where changes in data are made on a remote terminal during program execution. In this case, all such changes in input are made interactively from file INPUT during program execution.

TAPE5 is a file containing the bulk of the array data required for program execution. All such data are read free field. The data contained on TAPE5 and the array sizes are shown in table III.

TAPE10 contains mode-shape data to be employed in determining sensor deflections for computing feedback signals as indicated in table IV.

TAPE1 is a random-access binary file created by the program. Data on TAPE1 have been either input or generated in a particular overlay of STABCAR. These data may be required in other sections of the program during the current case or may be needed in a subsequent case. For example, the user may have selected a subset of the mathematical model. This file could be saved by the user for future runs. The option of using the TAPE1 data as input is accomplished by setting ICASE = ±1 and the option of selecting a subset of the model data stored on TAPE 1 is exercised by setting ISELECT = -1 as illustrated in case 6 in appendix G. The contents of TAPE1 are indicated in table IV.

Input Aids

Many cases are run which employ a previously generated TAPE1 for the bulk of the input. The user may be in doubt as to precisely what values are stored on TAPE1. TAPE2 variables and selected sensor data stored on TAPE1 may be determined by executing STABCAR with IFLUT = 0, ICASE = ±1, ICONSYS = 1, and TAPE1 local. If IAEROWR is set equal to 1, the aerodynamic force tabular arrays will also be output. If IMODEWR is set equal to 1, generalized masses, structural dampings, and either natural frequencies or the generalized stiffness matrix will be output.

A local scratch file called STABIN is produced by STABCAR which reflects TAPE2 or INPUT file data generated during a series of batch or interactive STABCAR runs. When ICASE = ±2 (i.e., an initialization run), STABIN contains the values of all variables including default values, except for array variables where only the first value and any nonzero values are exhibited. When ICASE = ±1, STABIN contains the values actually input as well as any default values pertinent to the analysis being attempted. STABIN is not rewound by STABCAR; therefore, it can provide the user with a record of what was done during a multicase session.

When IPKLT = +1, the user has a number of options pertaining to any sets of plot data stored on TAPE1. If the user sets IDELPLT = +1, he may delete sets of bad plot data by specifying which of the stored plot sets he wishes to keep. The user can specify which sets he wishes to keep in any order. They will then be resequenced in the order selected. If IDELPLT = 0, the user may combine a series of plot sets to form a composite plot. (See appendix G, table GVII, for example). Here too, the plot sets may be chosen in any convenient order. The choice of order is generally made so as to maximize ISAME (table II), which minimizes the number of times namelist $PLOTP has to be input (table GVII).

PROGRAM CAPABILITY DEMONSTRATION

In this section some of the capabilities of STABCAR are demonstrated. A mathematical model of an aircraft is defined and STABCAR is employed to examine its stability characteristics as a function of several parameters. The inputs required to generate each sample case are shown and a graphical display of the results from each case is presented.

Aircraft Mathematical Model

Results obtained by using STABCAR are shown for a mathematical model of an early version of the DAST ARW-2 (Drones for Aerodynamic and Structural Testing, Aeroelastic Research Wing Number 2). The DAST ARW-2 has a high-aspect-ratio (10.3) supercritical wing having 25° sweep at midchord mounted on a Firebee drone fuselage. The structural design of the wing was performed by the Boeing Military Airplane Company (refs. 21 and 22) under the assumption that an active flutter suppression system would be operational. The wing will flutter within the flight envelope with the flutter suppression system turned off. The structural model was supplied to NASA by the Boeing Military Airplane Company. The structural model was input into the NASTRAN program (ref. 6) to obtain the modal characteristics of the model. Two rigid-body modes (plunge and pitch), 10 symmetric elastic modes, and 1 control mode were retained for computation of generalized aerodynamic forces. The resulting modal information together with geometric and paneling data were input into the ISAC (Interaction of Structures, Aerodynamics, and Controls) program (ref. 7) where unsteady aerodynamic forces were generated by using a doublet lattice method. The aerodynamic paneling employed, as well as the location of the control surface and sensors to be employed for a flutter suppression system, is indicated schematically in figure 6. The flutter suppression control law that is used is shown in figure 7. In this control law, the difference in signals from two accelerometers located at $(x_s,y_s)_1 = (7.144, 2.083)$ m and $(x_s,y_s)_2 = (7.336, 2.134)$ m is fed back to an actuator driving an outboard aileron. Note that there is a scheduled component in the filter which varies with Mach number and dynamic pressure to provide appropriate phasing between sensor output and actuator input at varying flight conditions. Appendix B shows how the scheduled component was incorporated into STABCAR.

Graphical Output From Sample Cases

The amount of information to be output by STABCAR is selectable by the user. The basic file for printed output is TAPE6 which can be sent to the printer for either batch or interactive runs. This lengthy output is well identified by Hollerith descriptors and is not shown here. When interactive runs are being made, the user will typically minimize the information displayed to the screen. The variables employed to designate the output desired have been defined in table II in namelist $INPUT. In the results which follow, only graphical outputs to the screen are presented. The inputs required to generate each figure are given in appendix G.

Case 1 - Aerodynamic forces and p-plane approximation.- In case 1 (aerodynamic forces and p-plane approximation), a p-plane approximation was made to the aerodynamic forces. Figure 8 shows the selected aerodynamic force data input into STABCAR, the curves resulting from a quadratic interpolation for points between the data, and a p-plane fit. The elements shown are those for the three elastic modes that are most critical in the flutter mechanism and those for the corresponding control surface effectiveness elements. The inputs required to generate these figures are presented in appendix G.

Case 2 - Open-loop characteristic roots as function of altitude.- Figure 9(a) shows the root loci of the aircraft without controls as the altitude is varied. Case 2 (open-loop characteristic roots as function of altitude) is for a Mach number of 0.86; each point is matched by solving for the atmospheric density at the altitude as well as the speed at that altitude corresponding to the 0.86 Mach number. The open-loop flutter occurs at $h = 6.705$ km which corresponds to a velocity of 269.6 m/sec and a density of 0.6101 kg/m^3. Note that the roots corresponding to modes 3, 5, and 8 are varying with altitude, whereas the other roots considered are relatively unaffected. It will be shown, in a subsequent plot, that essentially the same flutter point is predicted with a mathematical model that retains only the 3d, 5th, and 8th modes in the analysis. Figure 9(b) presents essentially the same information as figure 9(a) except damping ratios and natural frequencies are plotted as explicit functions of the altitude.

Case 3 - Open-loop characteristic roots as function of density.- Results obtained by running case 3 are shown in figure 10. Figures 10(a) and (b) show root loci and damping ratio and natural frequency variations, respectively, as a function of density for $N_{Ma} = 0.86$ and $U = 269.6$ m/sec for no controls. The flutter point occurred at $\rho = 0.6101$ kg/m^3 which is in perfect agreement with the altitude variation case.

Case 4 - Open-loop characteristic roots as function of dynamic pressure.- Figures 11(a) and (b) for case 4 (open-loop characteristic roots as function of dynamic pressure) correspond precisely to figures 10(a) and (b). They are density variation runs but the independent variable has been changed to dynamic pressure to show the characteristic roots as a function of q. Plots of root variations as a function of dynamic pressure can also be made in altitude and velocity variation runs.

Case 5 - Characteristic roots as function of velocity (truncated model).- Case 5 shows results from using a truncated model. Figure 12(a) is an open-loop plot of damping ratios and natural frequencies versus velocity. The aerodynamic forces for $N_{Ma} = 0.86$ and the matched-point density found in figures 9 and 10 are employed. Thus, only the point at which $U = 269.6$ is a matched point. In this run, the select feature has been employed to consider only the effects of the modes that are most important to the onset of flutter. These are modes 3, 5, and 8 which are the 1st, 3d, and 6th elastic modes. The predicted flutter speed of 271.2 m/sec is within 1 percent of that found earlier with two rigid-body and six elastic modes. In figure 12(b) damping ratios and natural frequencies are plotted as a function of velocity under the same conditions and assumptions as for figure 12(a) except that the control loop is closed. The control-sytem representation is that shown in figure 7. The aircraft does not flutter for the range of velocities shown.

Case 6 - Elastic mode gain root loci.- Figure 13 shows gain root loci (case 6) for elastic mode roots at $N_{Ma} = 0.86$ and $h = 4.572$ km which correspond to U_D. For this condition the open-loop ARW-2 configuration is unstable as indicated by the root in the right half-plane in figure 13. The root loci are shown for both a p-k (solid lines) and a p-p (dashed lines) analysis. The unstable mode is stabilized at a gain of approximately 0.7. The slight difference between the p-k and p-p analyses for this root at the $j\omega$-axis crossing is an indication that the least-squares fit of the aerodynamic forces is not precise at the flutter frequency. Note that the mode number identification does not correspond to the numbering of the earlier figures. The matrix iteration technique was employed here to obtain initial estimates for the open-loop roots. The roots are numbered in ascending order of magnitude of the open-loop roots. Since the two rigid-body roots are not shown, the lowest number is 3.

Case 7 - Effect of phase errors upon elastic mode gain root loci.- Figure 14 illustrates the use of STABCAR to determine whether specified phase and gain margins are met by a candidate control law. This figure presents the gain root loci of the elastic mode roots (case 7) at the same flight condition as those for figure 13 for the p-p analysis only. Additional root loci are shown where the control signal has been modified by phase errors of -45° and 45°. The figure indicates that the candidate control law has less than a -45° phase margin at the nominal gain and does not meet a -6 dB gain margin at nominal phase.

Case 8 - Gain root loci of control system poles.- Figure 15 shows the gain root loci for the control system poles (case 8) at $N_{Ma} = 0.86$ and $h = 6.705$ km, which is the open-loop flutter point. The dynamic pressure at this condition is 25 percent less than that at U_D (the flight condition of figs. 13 and 14). For this dynamic pressure, root loci are shown for both p-k (solid lines) and p-p (dashed lines) analyses. Loci on the negative real axis were determined for the p-k analysis only because of the complications occurring in the p-p analysis due to isolated singular points and poles introduced by the p-plane approximation to the aerodynamic forces. Note in particular the branch where two real poles merge to form a complex pair. This type of locus can best be handled in an interactive run by trying different initial estimates having nonzero imaginary parts when the branch point is approached. Another curve of interest is the locus of the filter root which has an open loop pole at (-50,196.5) rad/sec. Its locus is similar to that of the 3d elastic mode in figure 13 which emphasizes the variation in root loci patterns that can occur as a result of different zero and open-loop pole locations between two flight conditions.

Case 9 - Short-period altitude root locus.- A short-period root locus (case 9) with altitude is presented in figure 16 for p-k and p-p analyses. The rigid-body degrees of freedom considered in this study are plunge and pitch about the center of mass with respect to an inertial set of axes. Drag contributions to the perturbation equations are neglected. The resulting equations for the rigid-body roots correspond approximately to those for the short-period approximation with additional effects of unsteady aerodynamic force contributions due to both rigid and elastic motion.

CONCLUDING REMARKS

STABCAR is a computer program designed for the study of the stability characteristics of flexible, actively controlled aircraft. Analyses can be performed that include either oscillatory or approximate fully unsteady aerodynamic forces. Determinant iteration is employed for characteristic root determination. Consequently, sufficiently accurate initial estimates and sufficiently small steps in the quantities being varied are required to converge upon and follow the characteristic roots.

STABCAR can be executed interactively with a remote terminal which allows the user to rapidly assess the results he is obtaining and conveniently change the input data. Changes such as deleting modes from the model, altering transfer functions or sensor types, or executing different types of analyses may be performed quickly and easily.

In an effort to aid potential users, a description of the mathematical techniques employed has been outlined, a flowchart has been provided, key variables have been defined, and a description of the input requirements has been given. In addition, a series of sample cases have been presented which provide explicit examples of the inputs required to utilize the major program capabilities.

STABCAR has been written specifically for efficient operation on Control Data equipment. Revisions are required to run this program on other equipment.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
March 7, 1984

APPENDIX A

p-PLANE APPROXIMATION TO THE UNSTEADY AERODYNAMIC FORCES

Least-Squares Solution

It is desired to approximate the jth column of aerodynamic forces with the following function:

$$\hat{Q}_j(p) = A_{0_j} + A_{1_j}p + A_{2_j}p^2 + \sum_{\ell=1}^{n_{\ell j}} \frac{A_{(\ell+2)_j}p}{p + b_{\ell j}} \tag{A1}$$

The form of this expression is the same as that found in references 1 and 2. However, constraints would have to be imposed upon the coefficients to have full equivalence between the fit described here and that of references 1 and 2.

Assume tabular data for oscillatory motion have been computed:

$$\left\{Q_j(k_1=0),\ Q_j(k_2),\ \ldots,\ Q_j\left(k_{(n_k)}\right)\right\} \tag{A2}$$

The objective is to determine the coefficients in equation (A1) such that $\hat{Q}_j(p = \sqrt{-1}\ k)$ best fits the tabular data in a least-squares sense subject to a set of linear equality constraints which may be imposed upon the coefficients.

Consider a particular element $\hat{Q}_{ij}$ of $\hat{Q}_j$. Let

$$A_{ij}^T = \left(A_{0_{ij}},\ A_{1_{ij}},\ A_{2_{ij}},\ \ldots,\ A_{\left(n_{\ell j}+2\right)_{ij}}\right) \tag{A3}$$

denote the coefficients to be determined. Suppose that there are linear equality constraints which the coefficients must satisfy denoted by

$$C_j^T A_{ij} = C_{0_{ij}} \tag{A4}$$

The error in the fit of equation (A1) to the data at $k = k_n$ is

$$e_{n_{ij}} \equiv \begin{Bmatrix} \mathrm{Re}\left[\hat{Q}_{ij}(k_n)\right] \\ \mathrm{Im}\left[\hat{Q}_{ij}(k_n)\right] \end{Bmatrix} - \begin{Bmatrix} \mathrm{Re}\left[Q_{ij}(k_n)\right] \\ \mathrm{Im}\left[Q_{ij}(k_n)\right] \end{Bmatrix} \tag{A5}$$

or

$$e_{n_{ij}} = B_n A_{ij} - \tilde{q}_{n_{ij}}$$

where

$$\tilde{q}_{n_{ij}} = \begin{Bmatrix} \mathrm{Re}\left[Q_{ij}(k_n)\right] \\ \mathrm{Im}\left[Q_{ij}(k_n)\right] \end{Bmatrix}$$

and

$$B_n = \begin{bmatrix} 1 & 0 & -k_n^2 & \dfrac{k_n^2}{b_{1j}^2 + k_n^2} & \cdots & \dfrac{k_n^2}{b_{n_{\ell j}j}^2 + k_n^2} \\ 0 & k_n & 0 & \dfrac{b_{1j}k_n}{b_{1j}^2 + k_n^2} & \cdots & \dfrac{b_{n_{\ell j}j}k_n}{b_{n_{\ell j}j}^2 + k_n^2} \end{bmatrix} \tag{A6}$$

Define the square of the magnitude of the error over all data points included in the least-squares fit to be

$$\left|e_{ij}\right|^2 = \sum_{n=1}^{nkk_j} e_{n_{ij}}^T e_{n_{ij}} \tag{A7}$$

where $nkk_j \leqslant n_k$. Then the coefficients which provide the least-squares error subject to the imposed constraints are found by determining the A_{ij} and $\tilde{\lambda}_{ij}$, which minimizes

$$\left|e_{c_{ij}}\right|^2 = \left|e_{ij}\right|^2 + 2\tilde{\lambda}_{ij}^T\left(C_j^T A_{ij} - C_{0_{ij}}\right) \tag{A8}$$

Performing the indicated minimization yields

$$\begin{bmatrix} \sum_{n=1}^{nkk_j} B_n^T B_n & C_{ij} \\ C_{ij}^T & 0 \end{bmatrix} \begin{Bmatrix} A_{ij} \\ \lambda_{ij} \end{Bmatrix} = \begin{Bmatrix} \sum_{n=1}^{nkk_j} B_n^T \tilde{q}_{n_{ij}} \\ C_{0_{ij}} \end{Bmatrix} \qquad \text{(A9)}$$

Selectable Constraints

The user may select constraints to impose upon the p-plane coefficients for column j from the following list:

1. Force agreement with tabular value at $k = 0$ (asumes that $k_1 = 0$):

$$\hat{Q}_{ij}(0) = Q_{ij}(0) = A_{0_{ij}} = [1 \quad 0 \quad 0 \quad \dots \quad 0] A_{ij} \qquad \text{(all i)}$$

2. Constrain slope at $k = 0$ based upon tabular data for column j (assumes that $k_1 = 0$ and k_2 is small):

$$\left.\frac{\partial \hat{Q}_{ij}}{\partial p}(p)\right|_{p=0} = \left[0 \quad 1 \quad 0 \quad \frac{1}{b_{1j}} \quad \dots \quad \frac{1}{b_{n_{\ell j} j}}\right] A_{ij} = \frac{\mathrm{Im}\left[Q_{ij}(k_2)\right]}{k_2} \qquad \text{(all i)}$$

3. Constrain slope at $k = 0$ for column j to be related to the tabular data in the mth column at $k = 0$ (assumes that $k_1 = 0$):

$$\left.\frac{\partial \hat{Q}_{ij}}{\partial p}(p)\right|_{p=0} = \left[0 \quad 1 \quad 0 \quad \frac{1}{b_{1j}} \quad \dots \quad \frac{1}{b_{n_{\ell j} j}}\right] A_{ij} = \frac{-\mathrm{Re}\left[Q_{im}(k_1 = 0)\right]}{b\theta_n}$$

This constraint can be used to relate plunge and pitch columns at $k = 0$ (ref. 23)

4. Constrain slope at $k = 0$ to be zero:

$$\left.\frac{\partial \hat{Q}_{ij}}{\partial p}(p)\right|_{p=0} = \left[0 \quad 1 \quad 0 \quad \frac{1}{b_{1j}} \quad \dots \quad \frac{1}{b_{n_{\ell j} j}}\right] A_{ij} = 0$$

5. Null out A_{k_j}:

   For example, let $k = 2$,

   $$A_{2_{ij}} = [0 \quad 0 \quad 1 \quad \ldots \quad 0]A_{ij} = 0$$

   where 1 occurs in column $k + 1$.

6. Require a precise fit at a specific, nonzero frequency k which need not be a tabular point:

$$\mathrm{Re}[\hat{Q}_{ij}(k)] = \left[1 \quad 0 \quad -k^2 \quad \frac{k^2}{b_{1j}^2 + k^2} \quad \cdots \quad \frac{k^2}{b_{n_{\ell j}j}^2 + k^2}\right] A_{ij} = \mathrm{Re}[Q_{ij}(k)]$$

$$\mathrm{Im}[\hat{Q}_{ij}(k)] = \left[0 \quad k \quad 0 \quad \frac{b_{1j}k}{b_{1j}^2 + k^2} \quad \cdots \quad \frac{b_{n_{\ell j}j}k}{b_{n_{\ell j}j}^2 + k^2}\right] A_{ij} = \mathrm{Im}[Q_{ij}(k)]$$

   If this constraint is selected, the tabular data are interpolated to obtain $Q_{ij}(k)$.

Instructions for preparing program input to include one or more of these constraints are given in table II. See specifically the variables ICOF, THETAN, and SPKO.

APPENDIX B

EXAMPLE OF INCLUDING SCHEDULED COMPONENTS IN COMPENSATOR

STABCAR contains a provision for including compensator elements that vary as functions of Mach number and dynamic pressure. This capability is provided in function subprogram SCHEDUL where control/sensor pair identification, Mach number, and dynamic pressure are available. A phase error ϕ_{ij} introduced into the compensation for one control/sensor pair is also available. There are block data statements in SCHEDUL. The user inputs data required to define his scheduling laws by using the block data statements. He then provides code in SCHEDUL which utilizes the data and the previously mentioned parameters to define the desired scheduling laws. An example illustrating the procedure for a single-input/single-output control law follows. The scheduling law implemented is that shown in figure 7.

```
      COMPLEX FUNCTION SCHEDUL(IC,JC,XMACH,QBAR,S)
      COMMON/GAIN/IGAIN,DELGAIN(10,20),GN(10,20),IPH,JPH,PHASE,PHERR
     ,ISCHDUL(10,20)
      COMPLEX PHERR
      COMMON/AEROP/NMSAVE,NCSAVE,NRSAVE
C     ,NKSAVE,NMNCS
      COMMON /GNSCHC/ COF1(10,20),COF2(10,20),COF3(10,20),COF4(10,20)
     ,COF5(10,20),COF6(10,20),COF7(10,20),COF8(10,20),COF9(10,20)
     ,COF10(10,20)
      COMPLEX S
      COMMON/SELECT/NMODES,NOC(25),NKNEW,NCNEW,NRNEW
C                  ,NOK(15),ISELECT
      DATA COF1/0.,0.,.61,197*0/
C        ,COF2/0.,0.,2.3924807E+04,197*0./
C        ,COF3/0.,0.,3.147319E-05,197*0./
C        ,COF4/0.,0.,1034.21,197*0./
C        ,COF5/0.,0.,1.,197*0./
C        ,COF6/0.,0.,735.,197*0./
C        ,COF7/0.,0.,833.3,197*0./
C        ,COF8/0.,0.,.0033257154,197*0./
C        ,COF9/0.,0.,300.,197*0./
C        ,COF10/0.,0.,75.,197*0./
      NMNC=NMODES-NCNEW
      CF1=COF1(IC,JC)
      CF2=COF2(IC,JC)
      CF3=COF3(IC,JC)
      CF4=COF4(IC,JC)
      CF5=COF5(IC,JC)
      CF6=COF6(IC,JC)
      CF7=COF7(IC,JC)
      CF8=COF8(IC,JC)
      CF9=COF9(IC,JC)
      CF10=COF10(IC,JC)
C*--> PHASE ERROR CAN BE INTRODUCED BETWEEN CONTROL IPH AND SENSOR JPH
C*                    EXP(I*PHERR)
      SCHEDUL=(1.,0.)
      IF(IC.EQ.IPH.AND.JC.EQ.JPH)SCHEDUL=SCHEDUL*PHERR
      IF(ISCHDUL(IC,JC).NE.1)RETURN
C*--> THIS CODE IMPLEMENTS THE SCHEDULED COMPONENTS OF THE CONTROL LAW
C*                    DEFINED IN FIGURE 7.
      XM=XMACH
C*--> IF(M < 0.61) THEN M=.61
      IF(XM.LT.CF1)XM=CF1
      QB=QBAR
C*--> IF(Q < 2.3924+04) THEN Q = 2.3924+04
      IF(QB.LT.CF2)QB=CF2

C*-->   -1 <= KS = (3.1417-05)*(1.61-M)*(Q - 1034.21) <= 1
      XKS=CF3*(1+CF1-XM)*(QB-CF4)
      IF(XKS.GT.CF5)XKS=CF5
C*-->   -75 <= DS = 735.-833.33*M + 0.0033267*Q <= 300.
      DS=CF6-CF7*XM+CF8*QB
      IF(DS.GT.CF9)DS=CF9
      IF(DS.LT.CF10)DS=CF10
C*-->   (KS/(S+DS))
      SCHEDUL=CMPLX(XKS,0.)/(S+DS)*SCHEDUL
      RETURN
      END
```

APPENDIX C

OPTIONS FOR HANDLING DENOMINATOR DYNAMICS ARISING FROM ACTUATORS, FILTERS, AND SENSORS

This appendix describes options contained in STABCAR which allow denominator dynamics arising from actuators, filters, and sensors to be cleared from the denominators of the equations. This option is employed when control roots are to be found with gains near zero to avoid division by zero.

Consider the following transfer function between actuator i and sensor j (see eqs. (9) and (13)):

$$\frac{(\delta_A)_i}{r_j} = (T_A)_{ii} G_{ij} (T_{LS})_{ij} e^{\sqrt{-1}\,\phi_{ij}} (T_L')_{ij} (T_S)_{jj}$$

Let

$$\tilde{G}_{ij} = G_{ij} (T_{LS})_{ij}\, e^{\sqrt{-1}\,\phi_{ij}} \tag{C1}$$

and

$$\tilde{T}_{ij} = (T_A)_{ii} (T_L')_{ij} (T_S)_{jj} \tag{C2}$$

Define

$$d = d_0 + d_1 s + d_2 s^2 + \dots + d_{n_d} s^{n_d} \tag{C3}$$

as the common denominator of $\{\tilde{T}_{ij}\}$, $i = 1, \dots, n_\delta$, $j = 1, \dots, n_s$. Define

$$N_{ij} = \tilde{G}_{ij} \tilde{T}_{ij}\, d \tag{C4}$$

An option is provided to multiply the control rows of either equation (1) or (2) through by d. The multiplication is performed if the variable IDMULT is input as 1. If IDMULT is input as 0, the multiplication is not performed. If equation (1) is to be employed, the variable ICSACT is input as 0. If equation (2) is to be employed, ICSACT is input as 1.

When characteristic roots are determined with the determinant iteration approach, the modifications to equations (1) and (2) that occur for IDMULT = 1 are evident. The modifications employed for the matrix iteration option are defined below (d_r is the first nonzero coefficient in the polynomial $d(s)$ (see eq. (C3)):

1. Let ICSACT = 1. Partition the matrix (see eqs. (2) and (18))

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\xi\xi} & \mathbf{A}_{\xi\delta} \\ \mathbf{A}_{\delta\xi} & \mathbf{A}_{\delta\delta} \end{bmatrix} \tag{C5}$$

Then $\mathbf{A}_{\delta\xi}$ and $\mathbf{A}_{\delta\delta}$ have the following definitions:

(a) For IDMULT = 0,

$$\mathbf{A}_{\delta\xi} = s^2 \frac{N(s)H}{d(s)} \qquad \mathbf{A}_{\delta\delta} = 0$$

(b) For IDMULT = 1,

$$\mathbf{A}_{\delta\xi} = \frac{s^2 N(s)H}{d_r s^r} \qquad \mathbf{A}_{\delta\delta} = \frac{-s^2\left[d(s) - d_r s^r\right]}{d_r s^r}$$

2. Let ICSACT = 0 (see eqs. (1) and (18)).

(a) For IDMULT = 0,

$$\mathbf{A} = -\left[\frac{1}{M_{ii}}\right]\left\{\left(M - \left[M_{ii}\right]\right)s^2 + \left[g_i(s)\ K_{ii}\right] + K + qQ\left(N_{Ma}, \frac{sb}{U}\right) - F_D\left[\frac{N(s)H}{d(s)} + T_{HE}(s)\ H_E\right]\right\}$$

(b) For IDMULT = 1,

$$\mathbf{A} = \frac{-1}{d_r s^r}\left[\frac{1}{M_{ii}}\right]\left\{\left[\left(M - \left[M_{ii}\right]\right)s^2 + \left[g_i(s)\ K_{ii}\right] + K + qQ\left(N_{Ma}, \frac{sb}{U}\right) - F_D T_{HE}(s)\ H_{HE}\right] d(s) - F_D N(s)\ H\right\} - s^2 \frac{\left(d(s) - d_r s^r\right)}{d_r s^r}$$

APPENDIX D

ALGORITHM FOR MATRIX ITERATION FOR CHARACTERISTIC ROOTS

The matrix iteration algorithm for characteristic root determination employed in STABCAR during convergence difficulties and for obtaining initial estimates for the characteristic roots is presented in this appendix.

Let

O_p — option type for which matrix iteration routine is being employed. $O_p = 1$ during convergence difficulties; $O_p = 2$ if initial estimates are being sought and all $\hat{s}_{i_2} = 0$; $O_p = 3$ if initial estimates are being sought and some $\hat{s}_{i_2} \neq 0$.

n_λ — number of characteristic roots to be found by matrix iteration. $n_\lambda = 1$ if $O_p = 1$; $n_\lambda = n_\xi$ if $O_p = 2$ and ICSACT = 0; $n_\lambda = n_\xi + n_\delta$ if $O_p = 2$ and ICSACT = 1; n_λ = Number of nonzero $\hat{s}_{i_2}$ if $O_p = 3$

i — counter for number of roots being found. $i = 1, \ldots, n_\lambda$

i_n — root number in determinant iteration for which convergence difficulties were incurred

j — counter determining number of iterations for a root

ℓ — eigenvalue number $\left(\text{from set of eigenvalues of } \mathbf{A}\left(\hat{s}'_{i_1}\right)\right)$ which corresponds to $\hat{s}'_{i_1}$, the initial estimate of current root being determined (see step 2)

$\hat{s}_{i_2}$ — best initial estimate for ith characteristic root

$\hat{s}_{i_1}$ — second initial estimate for ith root (see step 6)

$\hat{s}_{(i_n)_2}$ — estimate from determinant iteration routine for root for which convergence difficulties were incurred

s'_i — ith root being determined

$\hat{s}'_{i_j}$ — jth estimate for s'_i

s'_{i_j} — square root of eigenvalue of $\mathbf{A}\left(\hat{s}'_{i_j}\right)$ which corresponds to s'_i, i.e., $s'_{i_j} = \sqrt{\lambda'_\ell}$

ω_r — frequency used in estimating s'_1 if $O_p = 2$

λ'_ℓ — see step 2

APPENDIX D

M^{-1} inverse of generalized mass matrix

$\sqrt{\lambda'_\ell}$ square root with positive imaginary part

Step 1: Initialize:

$i = 1$

$\hat{s}'_{i_1} = \hat{s}_{(i_n)_2}$ if $O_p = 1$

$= (-0.5\omega_r, \omega_r)$ if $O_p = 2$

$= \hat{s}_{i_2}$ if $O_p = 3$

Determine M^{-1}

Step 2: Set $j = 1$:

Determine eigenvalues $\{\lambda'_r\}$ of $\mathbf{A}\left(\hat{s}'_{i_1}\right)$ (see eqs. (18))

Determine index ℓ such that $s'_{i_1} = \sqrt{\lambda'_\ell}$

$\ell = n_\lambda - (i - 1)$ if $O_p = 2$

Or ℓ is defined such that s'_{i_1} is closest in magnitude to $\hat{s}'_{i_1}$ if $O_p \neq 2$

Step 3: Determine convergence:

If $\min\left(\frac{\left|s'_{i_j} - \hat{s}'_{i_j}\right|}{\left|\hat{s}'_{i_j}\right|}, \left|s'_{i_j} - \hat{s}'_{i_j}\right|\right) < 0.0001$ go to step 6 (convergence)

Step 4: Set $\hat{s}'_{i_{j+1}} = \left(\hat{s}'_{i_j} + s'_{i_j}\right)\Big/ 2$ if $j = 1$

$$\hat{s}'_{i_{j+1}} = \frac{s'_{i_j}\hat{s}'_{i_{j-1}} - \hat{s}'_{i_j}s'_{i_{j-1}}}{\left(s'_{i_j} - \hat{s}'_{i_j}\right) - \left(s'_{i_{j-1}} - \hat{s}'_{i_{j-1}}\right)} \quad \text{if} \quad (j > 1)$$

Step 5: Set $j = j + 1$

If $j > 30$, print out nonconvergence, and go to step 7

Determine eigenvalues $\{\lambda'_{r_j}\}$ of $\mathbf{A}\left(\hat{s}'_{i_j}\right)$ (see eqs. (18))

Set $s'_{i_j} = \sqrt{\lambda'_\ell}$

Repeat step 3

Step 6: Set $\hat{s}_{(i_n)_2} = s'_{i_j}$ and return if $O_p = 1$

Set $\hat{s}_{i_2} = s'_{i_j}$ and $\hat{s}_{i_1} = 0.99\hat{s}_{i_2}$ if $O_p \neq 2$

Step 7: Set $i = i + 1$

If $i > n_\lambda$ return

Set $\hat{s}'_{i_1} = \sqrt{\lambda'_{(\ell-1)}}$ if $O_p = 2$

$\hat{s}'_{i_1} = \hat{s}_{i_2}$ if $O_p = 3$

Repeat step 2

APPENDIX E

GRAPHICS ROUTINES NOT INCLUDED IN STABCAR

This appendix is devoted to the description of elements of the Langley Graphics package that have been employed by STABCAR but are not included as a part of STABCAR. The description is included to facilitate a substitution of equivalent routines. Alternatively, the Langley Graphics package can be purchased separately from Computer Software Management Information Center (COSMIC), Suite 112, Barrow Hall, University of Georgia, Athens, GA 30602, as LAR 12722 entitled LRCGOS (Langley Research Center Graphics Output System).

SUBROUTINE PSEUDO

PURPOSE: To create and write an appropriately named Plot Vector File. Through linkages set up by an initial call to PSEUDO, all subsequent graphics data generated by the user will be routed through one of the PSEUDO entry points and written on the Plot Vector File. The PSEUDO processor is designed for use with the frame dependent postprocessors described in Section 1.3, Volume IV, of the Computer Programing Manual.

USE: CALL PSEUDO

This will establish a Plot Vector File named SAVPLT.

CALL PSEUDO(6LMYFILE)

This will establish a Plot Vector File named MYFILE.

NOTE: The Plot Vector File (or Files) will usually be written to disk (as opposed to tape) and may be postprocessed following user program termination via appropriate specification of one or more PLOT control cards. (See Section 1.3, Volume IV, Computer Programing Manual).

RESTRICTIONS: An initializing call to PSEUDO (with or without a file name argument) must be made prior to any calls to CALPLT or any other graphics output routine.

OTHER CODING INFORMATION: CALPLT and NFRAME are entry points in PSEUDO. PSEUDO checks for indefinite and infinite value and provides a traceback.

ENTRY NFRAME

PURPOSE: To provide users specific means of executing frame advance movements on any plotter device via an appropriate frame-oriented device postprocessor. Frame advance distances are generally defined to be incremental from current frame origin. A frame is defined as a finitely bounded plotting area; it is an imaginary boundary around the usable plotting area. CALL NFRAME is intended to be used as a frame advance mechanism, not as a plot origin offset.

USE: CALL NFRAME

APPENDIX E

ENTRY CALPLT

PURPOSE: To move the plotter pen to a new location with pen up or down. This is the basic plotting routine that is used by all other graphics routines; therefore, a check is made on the parameters for indefinite and infinite. If any of the parameters are indefinite or infinite, the program will abort and a traceback to the routine is provided. (See "OTHER CODING INFORMATION.")

USE: CALL CALPLT(X,Y,IPEN)

where

X,Y are the floating-point values for pen movement.

IPEN = 2 pen down

= 3 pen up

Negative IPEN will assign X = 0, Y = 0 as the location of the pen after moving the X,Y (create a new reference point).

RESTRICTIONS: All X and Y coordinates must be expressed as floating-point inches (actual page dimensions) in deflection from the origin.

SUBROUTINE LEROY/BALLPT

PURPOSE: To set up the parameters necessary to accommodate plotting with the liquid ink pen. Once set, this mode will remain.

USE: CALL LEROY

RESTRICTIONS: The CALL LEROY is only recognized by the CalComp postprocessor; it is ignored by the other postprocessors. In addition to reducing the speed of the plotter for all plotting movements, the number of plot vectors in any annotation is considerably increased.

The CALL LEROY must be made prior to any plotting calls, but after the CALL PSEUDO.

APPENDIX E

SUBROUTINE NUMBER

PURPOSE: To convert a floating-point number to BCD (expressed in F format) and to draw the resulting alphanumeric characters.

USE: CALL NUMBER(X,Y,HEIGHT,FPN,THETA,NODIGIT)

where

X,Y are the coordinates in floating-point inches of the left lower corner of the first digit of output.

HEIGHT is the height of the plotted number in floating-point inches. (See NOTATE routine.)

FPN is the floating-point number to be drawn.

THETA is the angle in floating-point degrees at which the number is to be drawn. (See NOTATE routine.)

NODIGIT is the number of decimal digits to the right of the decimal point for output.

NODIGIT=-1 or NODIGIT=0 both specify no decimal places; however, -1 suppresses the decimal point.

APPENDIX E

SUBROUTINE NOTATE

PURPOSE: To draw alphanumeric information for annotation and labeling and provide special centered symbols for annotation of data points.

USE: CALL NOTATE(X,Y,HEIGHT,BCD,THETA,NOCHAR)

where

X,Y are the floating-point coordinates of the first character.

For alphanumeric characters, the coordinates of the lower left-hand corner of the characters are specified.

HEIGHT specifies character size and spacing in floating-point inches for a full-size character.

BCD is the string of characters to be drawn and is usually written in the form: nHXXXX--- (the same way an alpha message is written using FORTRAN format statements). Instead of specifying alpha information as above, the beginning storage location of an array containing alphanumeric information may be given.

THETA is the angle in floating-point degrees at which the information is to be drawn. Zero degrees will print horizontally reading from left to right, 90° will print the line vertically reading from bottom to top, 180° will print the line horizontally reading from right to left (i.e., upside down), and 270° will print vertically reading from top to bottom.

NOCHAR is the number of characters, including blanks, in the label. NOCHAR is limited to 400 characters in one call.

APPENDIX E

SUBROUTINE AXES

PURPOSE: To draw a line, to annotate the value of the variable at specified intervals with or without tic marks, and to provide an axis identification label.

USE: CALL AXES(X,Y,THETA,S,ORG,SPX,TMAJ,TMIN,BCD,HEIGHT,NOCHAR,NDEC)

where

X,Y are the coordinates in floating-point inches of the starting point of the axis with reference to the plotting area origin as established by CALPLT.

THETA is the angle of rotation measured counterclockwise from the X-axis in floating-point degrees. Normally, THETA is 0° for an X-axis and 90° for a Y-axis.

S is the length of the axis in floating-point inches. Should be a multiple of TMAJ.

+S will generate tic marks.
-S will eliminate tic marks.

ORG is the functional value to be assigned to the origin (i.e., the value of the first scale) in floating point.

SFX is the adjusted scale factor for the array to be plotted (change in value per inch).
NOTE: Values of ORG and SFX which will produce a reasonable scale may be calculated using subroutine ASCALE or BSCALE.

TMAJ is the distance in floating-point inches for major tic marks (0.25 inch high). If the values are integer multiples, the decimal point and decimal places are eliminated. A negative TMAJ will cause the actual value to be written instead of the adjusted value.

TMIN is the number of divisions per inch in floating point for minor tic marks (0.125 inch high). To eliminate minor tic marks the following may be used:

TMIN = 0.

BCD is the character label for the axis (see NOTATE routine).

HEIGHT is the height of the full-size characters in the BCD title. Numbers at the tic marks will be (0.75 *HEIGHT) high. HEIGHT is in floating-point inches.

If HEIGHT = 0., all annotation will be eliminated.

NOCHAR is an integer specifying the number of characters in BCD title. A negative NOCHAR places the annotation on the clockwise side of the axis and a positive NOCHAR places the annotation on the counterclockwise side of the axis. NOCHAR = 0 is not allowed. If it is desired to have no label, then the BCD parameter should be 1H, and NOCHAR either +1 or -1. Normally NOCHAR = + for Y-axis and NOCHAR = - for X-axis.

NDEC is the number of decimal places after the decimal point for the numbers placed on the axis at the major tic marks. NDEC may be omitted. If NDEC is omitted, the default value of 2 is set.

ENTRY ASCALE

PURPOSE: To compute a scaling factor for an array of numbers to be plotted over a certain area and find the minimum data value with the array.

USE: CALL ASCALE(ARRAY,S,N,K,DV)

where

ARRAY is the name of the array containing the floating-point values to be scaled. (See RESTRICTIONS.)

S is the length (floating-point inches) over which the data are to be plotted (usually the length of one of the axes).

N is the number of data values in ARRAY from which points are to be plotted in accordance with K.

K is the interleave factor which specifies the sequence in which data are stored.

= 1 indicates the values are stored sequentially.

= 2 indicates that values are stored in every other location in the array, etc.

DV is the number of divisions per inch of the plotting paper to be used (should be: 10.0 or 20.0).

RESTRICTIONS: The array must be dimensioned to include storage space for two extra elements per interleave factor. For example: N = 100, K = 1, DIMENSION ARRAY (102); N = 75, K = 3, DIMENSION ARRAY (231).

METHOD: This routine scans the elements starting at location L of the array to find the minimum and maximum. (If the scaling begins with first location of array, L = 1; if scaling begins with third location of array, L = 3.) ASCALE computes an adjusted minimum (origin) and stores it in ARRAY(L+(N*K)) and computes a scale factor and stores it in ARRAY(L+(N*K)+K). The scale factor will be (A*10**J) where A is 1, 2, 4, or 5 and J is integer power. The data in the array may be scaled to floating-point inches by:

SV = (AE - AMV)/SF

where

SV = scaled value
AE = present value of array element
AMV = adjusted minimum value computed by ASCALE
SF = scale factor computed by ASCALE

APPENDIX E

SUBROUTINE BSCALE

PURPOSE: To compute a scaling factor for an array of numbers to be plotted over a certain area and to find an acceptable minimum data value for the array.

USE: 1. Stores the adjusted minimum value (origin) in ARRAY(L+(N*K)) and scale factor in ARRAY(L+(N*K)+K). If the scaling begins with first location of array, L = 1; if scaling begins with second location of array, L = 2, etc.

CALL BSCALE(ARRAY,S,N,K,TMAJ,M,ORG)

where

ARRAY is the name of the array containing the floating-point values to be scaled. (See RESTRICTIONS.)

S is the length (floating-point inches) over which the data are to be plotted (usually the length of one of the axes).

N is the number of data values in ARRAY from which points are to be plotted in accordance with K.

K is the interleave factor which specifies the sequence in which data are stored.

= 1 indicates that values are stored sequentially.

= 2 indicates that values are stored in every other location in the array, etc.

TMAJ is the distance in floating-point inches for major tic marks along the axis (length of S).

= 0. the routine scales in a manner similar to routine ASCALE.

> 0. the routine will adjust the origin and scale factor in order to give regular multiples of the delta number that will be written at the tic marks by the AXES routine.

M is an integer that has multiple uses.

= 0 if M is zero and TMAJ is zero, the routine scales in a manner similar to routine ASCALE.

> 0 if M is positive and TMAJ is zero, the data will be scaled over the full range of S, with no attempt to adjust the scale factor for appearance.

< 0 if M is negative, the valid value contained in parameter ORG will be used for a forced origin.

ORG is a floating-point value which will be used for a forced origin if parameter M is negative. The value in ORG may not be exactly the same on return from subroutine.

ORG may be omitted if M is positive or zero.

RESTRICTIONS: The array must be dimensioned to include storage space for two extra elements per interleave factor. For example: N = 100, K = 1, DIMENSION ARRAY (102); N = 75, K = 3, DIMENSION ARRAY (231).

APPENDIX E

SUBROUTINE PNTPLT

PURPOSE: To draw NASA Standard Plot symbols centered on a given coordinate value.

USE: CALL PNTPLT(X,Y,ISYM,IS)

where

X is the X coordinate for the centered symbol in floating-point inches.

Y is the Y coordinate for the centered symbol in floating-point inches.

ISYM is an integer specifying the symbol to be used. (See figs. E1 and E2.)

= 21 for a point.

= 22 for a plus sign +.

IS is an integer value specifying the size symbol to be used.

= 1 small

= 2 medium

= 3 large

NASA STANDARD PLOT SYMBOLS

Figure E1.- Plot symbols.

APPENDIX E

SUBROUTINE LINPLT

PURPOSE: To draw a line between and/or draw a NASA standard symbol at each successive data point (stored in an array), or to draw a continuous line through the data points and draw a specified symbol at the end of the line.

USE: CALL LINPLT(XARRAY,YARRAY,N,K,J,ISYM,IS,NG)

where

XARRAY, YARRAY — are the names of arrays containing the X values and Y values, respectively, to be plotted. Values must be in floating point. (See RESTRICTIONS.)

N — is the number of points to be plotted.

K — is the interleave factor which specifies the sequence in which data are stored.

= 1 indicates that values are stored sequentially.

= 2 indicates that values are stored in every other location in the array, etc.

J — is positive for line and symbol plot, negative for symbol-only plot. The magnitude specifies the alternate number of data points at which to plot a symbol.

= 0 for line plot with a specified symbol at the end of the line.

= 1 for symbol for every data point.

= 2 for symbol for every other data point, etc.

ISYM — is an integer describing symbol to be used, see PNTPLT routine for list (fig. E1).

= 0 no symbol will be drawn.

IS — is an integer specifying the size of the symbol (see PNTPLT routine).

NG — is an integer used to specify data points which will be enclosed by a 1-inch square grid, 20 × 20 lines to the inch. NG may be omitted. If NG is omitted, the default value of 0 is set.

= 0 no grids.

= 1 for every data point.

= 2 for every other data point, etc.

RESTRICTIONS: LINPLT expects the adjusted minimums and scale factors as described in ASCALE.

APPENDIX E

SUBROUTINE DLINPLT

PURPOSE: To draw a dashed line between and/or draw a NASA standard symbol at each successive data point (stored in an array), or to draw a continuous dashed line through the data points and draw a specified symbol at the end of the line.

USE: CALL DLINPLT(XARRAY,YARRAY,N,K,J,ISYM,IS,LP)

where

XARRAY, YARRAY are the names of arrays containing the X values and Y values, respectively, to be plotted. Values must be in floating point. (See RESTRICTIONS.)

N is the number of points to be plotted.

K is the interleave factor which specifies the sequence in which data are stored.

= 1 indicates that values are stored sequentially.

= 2 indicates that values are stored in every other location in the array, etc.

J is positive for line and symbol plot, negative for symbol only plot. The magnitude specifies the alternate number of data points at which to plot a symbol.

= 0 for line plot with a specified symbol at the end of the line.

= 1 for symbol for every data point.

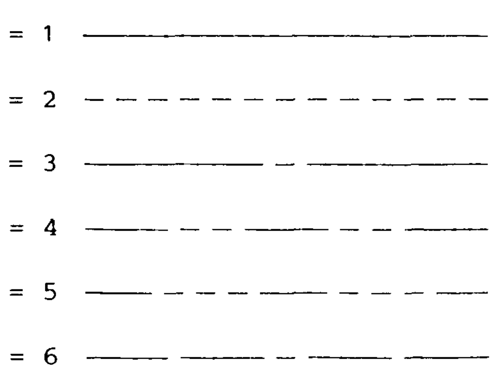
= 2 for symbol for every other data point, etc.

ISYM is an integer describing symbol to be used, see PNTPLT routine for list (fig. E1).

= 0 no symbol will be drawn.

IS is an integer specifying the size of the symbol (see PNTPLT routine).

LP is an integer used to specify the line pattern.



= 7

= 8

<u>RESTRICTIONS:</u> DLINPLT expects the adjusted minimums and scale factors as described in ASCALE.

APPENDIX F

DETAILED FUNCTIONAL DESCRIPTION OF OVERLAYS, MAJOR SUBROUTINES, AND IDENTIFICATION OF KEY PARAMETERS

In this appendix, a description of the functions of each overlay and its major subroutines is given including definitions of key input and internally generated parameters that control program flow.

Overlay MAIN

Overlay MAIN is the STABCAR executive. Data are input here which determine the type of case to run, where to find array data, and which overlays are required. Key parameters are defined as follows but are defined more completely in table II:

ICASE	Integer indicating type of case (interactive or batch) to run and where to get data.
ISPLANE	Integer indicating whether to make p-plane approximation to aerodynamic forces.
ISENSE	Integer indicating whether to perform interpolation for modal deflection or slopes at specified sensor locations.
INTERP	Integer indicating whether to compute geometric coefficients required for surface spline interpolation.
IPLATE(I)	Integer defining whether Ith structural section is modeled as beam or as plate.
ICSREAD	Integer defining whether sensor data are to be read from TAPE5.
ICONSYS	Integer defining whether control system is to be included.
ICHSEN, ICHFIL, and ICHACT	Integers defining whether changes to previously defined control system are to be made in current case; if not, overlay CONTROL is not needed.
ISELECT	Integer defining whether subset of current model is to be selected and where data for current model can be found.
IFLUT	Integer defining whether to perform stability analysis in current case.
IAPLT	Integer defining whether to make plots of aerodynamic forces.
IPKPLT	Integer defining whether to save data for making plots of stability characteristics.
IDELPLT	Integer defining whether to delete unwanted characteristic root data from data storage file TAPE1.

APPENDIX F

Overlay INTERP

Geometry-related, mode-independent coefficients which are needed for a surface spline fit for later interpolation purposes in overlay SENSOR for plate-type sections are computed by overlay INTERP. Primary subroutines called are INTEXT, SYMSURF, NONSURF, and MATINV. Key parameters are defined as follows:

NNODES (or NNODS) Total number of nodes at which modal data are available.

NPMX Maximum number of nodes in one structural section.

NSECTNS Number of structural sections vehicle is divided into.

IPLATE NSECTNS × 1 array.

IPLATE (I) { = 1 Ith section is modeled as a plate.
= 0 Ith section is modeled as a beam.

TAB NNODS × 4 array which, for plate-type sections, defines x,y coordinates (global) and z-deflection; fourth column is not employed for plate sections. For beam sections, array elements are distances along axis of rotation at which data are available, z-deflection, rotation about swept Y-axis (y') and rotation about swept X-axis (x'). (See fig. 3.) There is a TAB array for each elastic mode shape.

ISS 2 × NSECTNS array. ISS(1,I) is node number of first node of Ith structural section; ISS(2,I) is node number of last node of Ith structural section.

R0,X0, and Y0 NSECTN × 1 arrays. For plate-type modes, R0 = 1/B, where B is one-half root chord of Ith structural section; X0(I) is x-coordinate of Ith section root semichord; and Y0(I) is y-coordinate at root semichord. For beam-type sections, IPLATE = 0, (X0(I),Y0(I)) is elastic axis intercept with Ith structural section root chord; R0 is sweep angle of elastic axis of Ith structural section in radians.

H† Array of geometry coefficients.

Subroutine INTEXT.- Subroutine INTEXT is an extension of INTERP and it is here that the TAB arrays are actually read and geometry coefficients are obtained. Subroutines called are SYMSURF and NONSURF. Key parameters used are all those previously defined for overlay INTERP.

Subroutines SYMSURF and NONSURF.- Subroutines SYMSURF and NONSURF compute the mode-independent geometry coefficients for symmetric and antisymmetric or nonsymmetric plate sections, respectively. Subroutine called is MATINV. Key parameters used are TAB and H.

---

†Internally generated parameters.

Subroutine MATINV.- Subroutine MATINV performs the matrix inversion needed in computing H, the geometry coefficients.

Overlay SENSOR

Overlay SENSOR determines surface spline coefficients for fitting mode shapes over plate-type sections. Beam-type modal data are also input here. Interpolation of beam and plate modal data is performed to obtain modal coefficients defining either linear or angular displacements that would be measured by sensors at specified locations. These modal coefficients are stored for subsequent use and define the H matrix in equation (1) or (2). Primary subroutines are SENCTL, DEFLECT, CSIUNI, IUNI, SPLDER, ZFUN2, AFUN2, ZFUN, AFUN, ZDER, and ADER. Key parameters are defined as follows:

NS	Number of sensors.
XS and YS	NS × 1 array of sensor locations.
ITYPE	= 1 Linear measurement. = 2 Angular measurement.
NSS	NS × 1 array identifying structural section where each sensor is located.
ISS	2 × NSECTNS array defining beginning and ending node numbers for each structural section; see section "Overlay INTERP" for additional information.
IPLATE(I)	= 1 Ith structural section is modeled as plate. = 0 Ith structural section is modeled as beam.
TAB,RO,XO, and YO	Defined in section "Overlay INTERP."
NR	Number of rigid-body modes. In this overlay NR must be either 0, 1, or 2. If NR = 1, the rigid-body mode must be the negative of plunge; the user may input his own rigid-body modes by setting NR = 0 and treating his rigid modes as if they were elastic.
XCG	x-coordinate of airplane center of mass. Since first rigid-body mode is assumed to be the negative of plunge, and second, pitch about center of mass, this is only required if NR = 2.
U and V	Sensor location with respect to swept axis.
CS	Computed $n_s \times n_\xi$ array containing modal participation of each particular mode to input of each sensor.
W	Geometry coefficients computed in overlay INTERP for plate-type sections.

Subroutine SENCTL.- In subroutine SENCTL, the data required for modal interpolation are read in from file TAPE10, and after linear or angular displacements are calculated, resulting modal coefficients are written to data storage file TAPE1. Primary subroutine called is DEFLECT. Key parameters used are all those defined for overlay SENSOR.

Subroutine DEFLECT.- In subroutine DEFLECT, the actual modal coefficients defining linear or angular displacement inputs to sensor are obtained. Primary subroutines called are CSIUNI, IUNI, SPLDER, ZFUN2, AFUN2, ZFUN, AFUN, ZDER, and ADER. Key parameters used are all those defined for SENSOR except NS and NSS.

Subroutines IUNI and CSIUNI.- Subroutines IUNI and CSIUNI are used to perform interpolation of modal data used in calculating both linear and angular displacements for beam-type sections (IPLATE = 0). IUNI performs linear and quadratic interpolation when there are less than four tabular values and CSIUNI performs a cubic spline interpolation when there are four or more tabular values. Key parameters used are TAB and V.

Subroutine SPLDER.- Subroutine SPLDER interpolates for Ry' and estimates dRy'/dy' for beam-type sections (IPLATE = 0) used in computing angular displacements (ITYPE = 2). Key parameters used are TAB and V.

Subroutines ZFUN2 and AFUN2.- In subroutines ZFUN2 and AFUN2, spline coefficients are determined for symmetric (ISYM = 1), nonsymmetric or antisymmetric (ISYM $\neq$ 1) plate-type sections (IPLATE = 1). Key parameters used are TAB and W.

Subroutines ZFUN and AFUN.- In subroutines ZFUN and AFUN, modal coefficients for linear displacement-type sensors are determined for symmetric (ISYM = 1) and nonsymmetric or antisymmetric (ISYM $\neq$ 1) plate-type sections (ITYPE = 1 and IPLATE = 1). Key parameters used are TAB, W, XS, and YS.

Subroutines ZDER and ADER.- In subroutines ZDER and ADER, modal coefficients for angular displacement or rotational-type sensors are determined for symmetric (ISYM = 1) and nonsymmetric or antisymmetric (ISYM $\neq$ 1) plate-type sections (ITYPE = 2 and IPLATE = 1). Key parameters used are TAB, W, XS, YS, and RO.

Overlay MATINPT

Large data arrays such as the aerodynamic data, generalized masses, generalized stiffnesses, and control system dynamics definitions are input in overlay MATINPT. In addition, the selection feature is available which enables one to choose a mathematical model that is a subset of a previously defined model. Primary subroutines called are MATEXD, CSINPUT, and SELECTS. Key parameters are defined as follows:

ISPLANE
- = 1,2 Read p-plane coefficients.
- = 0,3 Read aerodynamic coefficients.

ISELECT

- < 0 Read modal data from data storage file TAPE1 (select from previously used data).
- > 0 Read modal data as formatted input from file TAPE5 (select from original data).
- = 0 No selection of array data; for initial case, data from file TAPE5 are used; in subsequent cases, array data from preceding case are used for current case.

ICONSYS

- = 1 Call subroutine CSINPUT to input sensor and control system data.
- = 0 No control system input.

KASE† Number of cases run during current program execution.

ICSREAD

- = 0 Read sensor data from data storage file TAPE1, record 16 as computed and stored in overlay sensor.
- = 1 Read sensor data as input from file TAPE5.

INTERAC

- = 1 Interactive run is being made (additional input after initialization case is requested from keyboard).
- = 0 Run is in batch mode (additional input after initialization case must also be on TAPE2.

ICHSEN

- = 1 Change in sensor dynamics.
- = 0 No change in sensor dynamics.

ICHFIL

- = 0 No change in filter dynamics.
- = 1 Change in filter numerator.
- = 2 Change in filter denominator.
- = 3 Change in both numerator and denominator.

ICHACT

- = 0 No change in actuator dynamics.
- = 1 Change in actuator numerator.
- = 2 Change in actuator denominator.
- = 3 Change in both numerator and denominator.

NM and NMODES Number of modes employed in model before selection and number to be selected (ISELECT $\neq$ 0).

---

†Internally generated parameter.

NR and NRNEW — Number of rigid-body modes employed in model before selection and number to be selected (ISELECT $\neq$ 0).

NC and NCNEW — Number of controls employed in model before selection and number to be selected (ISELECT $\neq$ 0).

NK and NKNEW — Number of reduced frequencies at which aerodynamic tabular values are employed in model before selection and number to be selected (ISELECT $\neq$ 0). If NKNEW = 0 (default), all will be used and no selection per frequency will be performed.

NOC — Array of mode numbers indicating which modes to select (ISELECT $\neq$ 0).

NOK — Array of frequency indices indicating which reduced frequency data to select (ISELECT $\neq$ 0 and NKNEW $\neq$ 0).

NS — Number of sensors.

Subroutine MATEXD.- All noncontrol array data are read in subroutine MATEXD and all array data are rewritten to data storage file, TAPE1, in form to be used by other overlays. Primary subroutines called are CSINPUT and SELECTS. Key parameters used are ISPLANE, ISELECT, ICONSYS, KASE, INTERAC, NM, NR, NC, NS, and all arrays defining aeroelastic model and control system.

Subroutine CSINPUT.- All control system data are read in subroutine CSINPUT, or default values set, including sensor deflection coefficients. Key parameters used are ICSREAD, ICHSEN, ICHFIL, ICHACT, NS, NC, and arrays defining control system.

Subroutine SELECTS.- All modal and frequency selection of data is performed in subroutine SELECTS. Key parameters used are ISPLANE, INTERAC, KASE, NM, NMODES, NR, NRNEW, NC, NCNEW, NK, NKNEW, NOC, NOK, NS, and all arrays defining basic aeroelastic model.

Overlay PPLANE

The p-plane curve approximations to the oscillatory aerodynamic forces are computed in overlay PPLANE corresponding to a specified set of denominator constants $\left(\{b_j\}\text{ (eq. (3))}\right)$. The best coefficients to employ in equation (3) for a given set of oscillatory aerodynamic force data at a number of reduced frequencies are determined in a least-squares sense. The resulting coefficients and the corresponding denominator constants are stored for subsequent use in overlays PKFLUT and AEROPLT. Primary subroutines called are SET and AERCOEF. Key parameters are defined as follows:

NK — Number of reduced frequencies at which aerodynamic data are available.

NKK(J) — Jth element of NM × 1 array indicating that only first NKK reduced frequencies of aerodynamic data are to be employed in least-squares fit for Jth column of aerodynamic forces.

NCOEF — Number of coefficients desired in p-plane fit, including polynomial and denominator (lag) terms (Max = 10).

NPOLYC(J) — The number of polynomial coefficients for column J.

Integer indicating whether to include nth constraint (as defined below) when fitting Jth column of aerodynamic forces. (See appendix A.) It is assumed that $k_1 = 0$ and k_2 is small.

ICOF(N,J)
- = 0 (Default) Nth constraint not employed in Jth column of aerodynamic forces.
- > 0 Employ Nth constraint.

ICOF(1,J) > 0 Force agreement with first tabular value at $k_1 = 0$; $\hat{Q}_{ij}(0) = Q_{ij}(0)$.

ICOF(2,J) > 0 For $k_1 = 0$, force slope,

$$\left.\frac{\partial \hat{Q}_{ij}}{\partial p}\right|_{p=0} = \mathrm{Im}\{Q_{ij}(\varepsilon)\}/\varepsilon, \text{ where } \varepsilon = k_2.$$

ICOF(3,J) > 0 For $k_1 = 0$, Force slope = $-\mathrm{Re}\{Q_{i2}(0)\}/(b\theta_n)$.

ICOF(4,J) > 0 For $k_1 = 0$, Force slope = 0.

ICOF(5,J) = $\ell + 1$ Force $A_{\ell_j} = 0$.

ICOF(6,J) > 0 Force $\hat{Q}_j(k_0) = Q(k_0)$ for some k_0, where $Q_j(k_0)$ is interpolated value of Jth aerodynamic force at k_0, obtained with a piecewise quadratic fit to $Q_j(k_i)$, $i = 1, \ldots, n_k$.

BN(I,J) Constant in Ith denominator term of form P/(BN + P) corresponding to Jth column (pressure mode); if same set of BN is used for several columns they need only be input once. (See NCOL.)

THETAN Normalizing factor employed when applying constraint number 3, ICOF(3,J) = 1; Default = 1.

SPKO Value of reduced frequency for which p-plane approximation must precisely fit data.

NM Number of modes being employed in model.

NC Number of controls being employed in model.

NOC Array of mode numbers indicating which modes are included in current model.

NCOL(J)
- = N Integer indicating preceding column number, if any, for which BN(I,J) = BN(I,N) (default is 1).
- = 0 BN terms for Jth column are to be input.

NBN(J)[†] Number of nonzero BN for column J.

C(I,J,K) Ith p-plane coefficient for (J,K) element of generalized aerodynamic force matrix.

IERPRT
- = 1 Write percent errors for p-plane fit at tabular values to TAPE6.
- = 0 Do not write.

Subroutine SET.- The number of lag terms is determined for each column of the aerodynamic forces based upon input parameters. The logic in subroutine SET was developed to reduce the input required of the user. The parameter NCOL(J) is employed to indicate whether lag terms for column J were read in (NCOL(J) = 0) or are the same as those for column JJ = NCOL(J). Key parameters used are NCOEF, BN, NM, NC, NOC, NCOL, and NBN.

Subroutine AERCOEF.- Subroutine AERCOEF solves for the coefficients that yield the minimum error in the least-squares sense in the p-plane fit to the aerodynamics. A library subroutine MATOPS is employed to solve the set of linear equations which result. (See appendix A.) Key parameters used are all for overlay PPLANE.

Overlay CONTROL

For overlay CONTROL, a matrix of transfer functions $T_A T_L' T_S$ is constructed. (See eq. (9).) The $(\ )_{ij}$ element relates the output of actuator i to the input of sensor j, excluding $(G)_{ij}$ and any scheduling or phase error. The polynomial coefficients for sensor, compensation, and actuator dynamics are constructed separately and then combined and a common denominator is found. Each element of the resulting matrix can then be multiplied by a distinct feedback gain and scheduling component in overlay PKFLUT. Primary subroutines called are CONTEXT, SENDYN, CLOGIC, FILDYN, ACTDYN, PMULT, NCOF, and EQUATEP. Key parameters are defined as follows:

NC Number of controls.

NS Number of sensors.

ISDYN(I)
- = 0 Perfect sensor (default).
- = 1 Sensor dynamics included.

IORD(I)
- = 0 Position sensor.
- = 1 Rate sensor.
- = 2 Acceleration sensor.

XKS(J,I) Jth coefficient of numerator polynomial for Ith sensor.

[†]Internally generated parameter.

ASN(J,I)[†]
- = 1 If J = IORD(I) + 1 and ISDYN(I) = 0.
- = 0 If J ≠ IORD(I) + 1 and ISDYN(I) = 0.
- = XKS(JJ,I) where J = JJ + IORD(I) + 1 if ISDYN(I) = 1.

ASD(J,I)
- = 1 If ISDYN(I) = 0 and J = 1.
- = 0 If ISDYN(I) = 0 and J > 1.
- = Jth coefficient of denominator polynomial for Ith sensor if ISDYN(I) = 1.

ISN(I) and ISD(I) — Number of coefficients in numerator and denominator, respectively, of transfer function modeling Ith sensor dynamics.

IACT(I)
- = 0 Ith actuator is modeled as perfect actuator.
- = 1 Actuator dynamics are included for Ith actuator.

AACTN(J,I) and AACTD(J,I) — Jth coefficient of numerator and denominator of polynomial, respectively, of transfer function used to model Ith actuator.

IAN(I) and IAD(I) — Number of coefficients in numerator and denominator of Ith actuator transfer function.

ACLN(K,I,J) and ACLD(K,I,J) — Kth coefficient of numerator and denominator polynomial, respectively, of transfer function which relates Ith actuator input to Jth sensor output excluding gain, scheduling, and phase error contributions.

ILN(I,J) and ILD(I,J) — Number of numerator and denominator coefficients in transfer function relating Ith actuator input to Jth sensor output.

$IFILTER_{ij}(K)$ — Indicates which types of filters are to be combined to define overall filter transfer function relating actuator i input to sensor j output. Maximum of 12 filters for each control-sensor pair. IFILTER(K) is Kth filter type (sequence must end with 1).

IFILTER(K)
- = 1 No filtering.
- = 2 Notch filter.
- = 3 Integral filter.
- = 4 Proportional plus derivative filter.
- = 5 Lead-lag (or lag-lead) filter.
- = 6 General filter comprised of numerator and denominator polynomials.

[†] Internally generated parameter.

WNI(K) and SNI(K) — Natural frequency and damping ratio of notch filter if IFILTER(K) = 2.

KIBDO(K) — Gain employed in integral filter if IFILTER(K) = 3.

KIBD1(K) and KIBD2(K) — Gains employed for proportional plus derivative filters, respectively, if IFILTER(K) = 4.

AN1(K) and AD1(K) — Time constants of zero and pole in lead-lag (lag-lead) compensator if IFILTER(K) = 5.

AFN(I,K) and AFD(I,K) — Numerator and denominator coefficients (constant term first) of general-type transfer function if IFILTER(K) = 6.

ACTLN(K,I,J)† and ACTLD(K,I,J)† — Kth computed coefficient of numerator and denominator polynomial, respectively, of overall transfer function $(T_A T_L' T_S)_{ij}$ relating actuator I output to sensor J input, excluding gain, scheduling, and phase error contributions.

ICLN(I,J)† and ICLD(I,J)† — Number of coefficients in numerator and denominator of overall transfer function $(T_A T_L' T_s)_{ij}$.

DCT† — Array of coefficients in common denominator d of all transfer functions $(T_A T_L' T_s)_{ij}$, i.e., common multiple of all ACTLD arrays.

NCT — Number of coefficients in DCT.

CNTLN(K,I,J)† — Kth coefficient of polynomial corresponding to product of ACTLN(I,J) and common denominator DCT = $(T_A T_L' T_s)_{ij} d$.

Subroutine CONTEXT.- In subroutine CONTEXT the overall matrix product of the sensor, control logic, and actuator matrices of transfer functions is formed and a common denominator is found. Primary subroutines called are SENDYN, CLOGIC, and ACTDYN. Key parameters used are all except IFILTER, WNI, SNI, KIBOO, KIBO1, KIBO2, AN1, AD1, AFN, and AFD.

Subroutine SENDYN.- Subroutine SENDYN determines the numerator and denominator polynomial coefficients and order of the transfer functions relating sensor output to sensor input. The numerator coefficients are adjusted for the appropriate power of s to define a position, velocity, or acceleration sensor. Subroutine called is NCOF. Key parameters used are NS, ISDYN, IORD, XKS, ASN, ASD, ISN, and ISD.

Subroutine CLOGIC.- Subroutine CLOGIC sets up the input for calls to FILDYN in order to obtain the numerator and denominator polynomial coefficients which make up the transfer functions relating sensor output to actuator input. Primary subroutine called is FILDYN. Key parameters used are ACLN, ACLD, ILN, and ILD.

Subroutine FILDYN.- Subroutine FILDYN contains the filter types that may be combined to construct control logic dynamics. The mathematical expressions that represent the various filter types are defined in the section "Control-System Representation." Any combination of the filter types may be utilized to generate an overall filter. For example, if IFILTER(1) = 3, IFILTER(2) = 6, IFILTER(3) = 6,

---

†Internally generated parameter.

and IFILTER(4) = 1, the overall filter is the product of one integral and two general polynomial filters. When IFILTER(K) = 1 is encountered, no further filter dynamics are computed for the particular control sensor pair under consideration. Therefore, the IFILTER array must end with 1. Subroutines called are NCOF and PMULT. Key parameters used are IFILTER, WN1, SNI, KIBD0, KIBD1, KIBD2, ANI, ADI, AFN, and AFD.

Subroutine ACTDYN.- Subroutine ACTDYN determines the order of the polynomials in the transfer function representing the actuator dynamics relating actuator output to actuator input. Subroutine called is NCOF. Key parameters used are IACT, AACTN, AACTD, IAN, and IAD.

Subroutine NCOF.- NCOF is a function subroutine which returns the number of coefficients in a polynomial; that is, it determines the order +1 of a polynomial.

Subroutine PMULT.- The utility subroutine PMULT multiplies two polynomials together. All polynomial arrays are assumed to have constant terms first.

Subroutine EQUATEP.- The utility subroutine EQUATEP sets one polynomial equal to another.

Overlay PKFLUT

The matrix in equation (1) or (2) is formed by overlay PKFLUT. A user-specified number of characteristic roots are determined as a function of variation in velocity, altitude, density, or feedback gains and the resulting output is stored for possible subsequent plotting (if IPKPLT $\neq$ 0). (See section "Characteristic Root Determination.") By operating in a multicase mode, the effect of variation in other parameters such as actuator or compensator dynamics or sensor location can also be studied. Primary subroutines called are PKEXT, MATIT, MNLOOP, DETIT, FINSCN, AT62, PREDICT, FUN, DEFINE, SCHEDUL, AERO, AEROSP, CXQR, MATINV, CXDEV, IUNI, and CSIUNI. Key parameters are defined as follows:

IOPT2
- = 1 Use matrix iteration to obtain initial estimates of roots.
- = 0 Do not calculate initial roots using matrix iteration.

ITRACE(I) Array of integers indicating whether to trace Ith root either from input or using IOPT1 or IOPT2 options. These parameters also indicate type of curve used in extrapolating for $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$. (See section "Determinant Iteration.")

- = 1 Linear.
- = 2 Quadratic.
- = 3 Cubic.

KVAR — Integer indicating type of variation to be performed. If negative, any corresponding plot output will be with respect to dynamic pressure. Can be reset interactively at plot time of IPKPLT > 0.

= ±1 Velocity variation.

= ±2 Altitude variation.

= ±3 Density variation.

IPKPLT — = 0 No plots.

= ±1 Root loci.

= ±2 ζ, ω_n plots.

= ±3 Both root loci and ζ, ω_n plots.

= + Allows alteration of titles, scales, curve deletion, etc., at plot time.

NV — Number of increments in parameter being varied.

IGAIN — { = 0 No gain variation. = 1 Gain variation.

DV — Increment in parameter variation

V0, H0, and RHO0 — Initial values of velocity, altitude, and density

DVEL†, DRHO†, and DH† — Incremental change in velocity, density, and altitude. Only one is set to DV, depending upon KVAR. Others are zero.

GN(I,J) — Current value of gain on transfer function relating actuator I output to sensor J input.

DELGAIN(I,J) — Increment in gain GN(I,J) for gain variation.

VV, H, and DENS — Current values of velocity, altitude, or density during parameter variation.

XVC and XV — Current value of independent parameter being varied.

S1I and S2I — Array of initial estimates, $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$, of roots to be traced. These need to correspond to initial value of parameter (velocity, altitude, density, or gain) variation. S2I, only, is used initially in MATIT if IOPT2 = 1 as an estimate (not required) of roots to be traced. MATIT will return values for both S1I and S2I in this case. S2I is always assumed to be the better estimate in determinant iteration.

---

†Internally generated parameter.

NF† Number of characteristic roots for which to solve.

ITEST† Integer indicating whether there is a convergence problem in tracing a specified root.

= 1 Iteration procedure converged.

= 2 Iteration procedure failed to converge.

CONFAC1 Factor for converting input velocity to feet per second for use in subroutine AT62.

CONFAC2 Factor for converting input altitude to feet for use in subroutine AT62.

CONFAC3 Factor for converting density output by subroutine AT62 in slugs per cubic foot into units corresponding to input model.

IPH and JPH Control (IPH) and sensor (JPH) pair for which phase error is to be introduced.

PHASE Phase error, in degrees, to be introduced in IPH control, JPH sensor transfer function.

IPRINT Option indicating how often to print characteristic root data to screen (or to file OUTPUT) during interactive session; IPRINT = N, prints every Nth iteration. All data are written to file TAPE6, regardless of IPRINT.

XMACH Mach number at which aerodynamic data were generated.

QBAR† Dynamic pressure corresponding to current velocity and density.

KFIT Type of interpolation to be performed to obtain aerodynamic force at particular frequency when k aerodynamic tabular data are being used rather than p-plane aerodynamic approximations.

If KFIT is negative, the interpolation will be for $\mathrm{Re}(Q(k))$ and $Q'(k)$ (see unsteady aerodynamic force subsection)

= ±1 Linear.

= ±2 Quadratic.

= ±3 Cubic spline.

P1† and P2† Nondimensionalized estimates, $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$ of characteristic root prior to determinant iteration at each value of independent variable. Upon returning from determinant iteration, P2 is current converged root, if found, and P1 is preceding converged root.

---

†Internally generated parameter.

P† — Value of estimate for nondimensionalized root in MATIT corresponding to root in determinant iteration procedure for which convergence difficulty was incurred. Upon return from MATIT, P is replaced by actual root, if found.

P11(I)† and P22(I)† — P22(I) is value of converged Ith root λ_i for current value of independent variable, and P11(I) is value of converged Ith root for preceding value of independent variable. P22(I) is employed to prevent iterations for Ith root from converging upon previously determined root by defining determinant

$$DET = DET \Big/ \prod_{i=1}^{I-1} (p - \lambda_i)$$

PP(I,IN)† (I=1,4) and PPF(I) — Array of up to four preceding values of converged INth root. P11(IN) is equivalent to PP(4,IN). Array is used for predicting new root estimates P1 and P2.

NORD† — Order of extrapolating curve currently being used to predict estimates P1 and P2.

STEPMX† — Maximum step increase allowed in amplitude of characteristic root in determinant iteration.

SIGN† —
- < 0 Indicates change in stability has occurred in current root being traced.
- > 0 No change in stability has occurred.

NCHECK† — NF × 1 vector recording successive number of times noncovergence has occurred in roots being traced. If NCHECK(IN) ⩾ 5, the INth root is no longer traced.

ITRACK† — NF × 1 vector initially having 1 in all elements. If NCHECK(IN) ⩾ 5, ITRACK(IN) is set to 0, and computations for INth root are bypassed

NFINE — Number of subintervals into which increment in independent variable is divided in performing finer scan for particular root. Number of values at which INth root is determined in finer scan. N = NFINE + 1.

IDEF† —
- = 1 Make precise determination of point where first change in stability occurs.
- = 0 Bypass precise determination.

---

†Internally generated parameter.

Y† — Vector array giving frequency, flutter speed, altitude, density, and damping obtained by interpolating for value of independent variable at which damping in INth mode is zero; i.e., at which change in stability occurred.

IT† and NIT — Number of iterations employed in seeking characteristic root, and maximum number allowed.

ICUT† and NCUT — If predicted step increases magnitude of determinant, step size is halved up to NCUT times in attempt to improve estimate of characteristic root. ICUT is number of times halving has occurred.

EPSI — If either relative or absolute change in estimate for characteristic root is less than EPSI, convergence is assumed.

PROOT† — Estimate of characteristic root at which determinant is being evaluated.

SCHEDUL — Complex number, for each control sensor pair, returned from SCHEDUL function subprogram giving contributions of scheduling component and any phase error.

IDAMP
- = 1 $g_i = i g_{s_i}$
- = 2 $g_i = \left(s/|s|\right) g_{s_i}$
- = 3 $g_i = s/\omega_{n_i} \, g_{s_i}$

ISPLANE
- = 0 Aerodynamic forces are computed as function of $k = \text{Im}(p)$.
- = 1 Aerodynamic forces are obtained from p-plane approximation by using coefficients previously generated.
- = 2 p-plane approximations to unselected set of modes is determined.
- = 3 p-plane approximations to selected subset of modes is determined.

ICONSYS
- = 1 Control system is included.
- = 0 No control system.

IDMULT
- = 1 Control system denominator is cleared.
- = 0 Control system denominator polynomial is not cleared.

---

†Internally generated parameter.

ICSACT $\begin{cases} = 1 \text{ Aerodynamic and inertial hinge moments are neglected in control rows of equations of motion} \\ = 0 \text{ Control surface is included as full degree of freedom, and effects of inertial and aerodynamic hinge moments are considered.} \end{cases}$

Subroutine PKEXT.- In subroutine PKEXT, modal, sensor, aerodynamic, and control system data are read-in from data storage file TAPE1; the parameter variation loop is controlled; and characteristic root data are stored on data storage file TAPE1 for plotting later if IPKPLT ≠ 0. Primary subroutines called are MATIT and MNLOOP. Key parameters used are IOPT2, ITRACE, IPKPLT, ISPLANE, ICONSYS, KVAR, IGAIN, NV, DV, DVEL, DH, DRHO, DELGAIN, S1I, S2I, NF, and IPRINT.

Subroutines MATIT and MATITX.- Subroutines MATIT and MATITX, together, perform a matrix iteration to determine the characteristic roots. They are used to obtain initial estimates $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$ of all the roots at the beginning value of the parameter (velocity, altitude, density, or gain) being varied if IOPT2 = 1 and/or to aid in obtaining a specific root when the determinant iteration process has convergence difficulties. (See appendix D.) Primary subroutines called are AERO, AEROSP, SCHEDUL, CXQR, and MATINV. Key parameters used are VV, H, DENS, S2I, P, and ITEST.

Subroutine MNLOOP.- In subroutine MNLOOP, the independent variable is incremented, initial estimates, of characteristic roots being traced are obtained, and (for each increment), the characteristic roots are determined using the determinant iteration method, if possible, or the matrix iteration method if the determinant iteration process fails to converge. Primary subroutines called are AT62, PREDICT, DETIT, MATIT, and FINSCN. Key parameters used are ITRACE, KVAR, IPKPLT, IGAIN, DVEL, DH, DRHO, DELGAIN, VV, H, DENS, NF, ITEST, P1, P2, P11, P22, PP, NORD, STEPMX, SIGN, NCHECK, ITRACK, NFINE, IDEF, IT, NIT, ICUT, NCUT, XMACH, CONFAC1, CONFAC2, CONFAC3, XVC, and IPRINT.

Subroutine DETIT.- Subroutine DETIT performs the actual determinant iteration to obtain λ_{IN}, the INth characteristic root corresponding to P2. See section "Determinant iteration." Subroutines called are FUN1 AND FUN. Key parameters used are ITEST, P1, P2, STEPMX, IT, NIT, ICUT, NCUT, EPSI, and PROOT.

Subroutine FINSCN.- Subroutine FINSCN traces the IN characteristic root for smaller increments in the parameter being varied (velocity, altitude, density, or gain) in the same manner as MNLOOP in case of a change in stability or convergence difficulties. A new value is returned for $P2 = \lambda_{IN}$. The first time a change in stability occurs, the precise point where it occurs is obtained. Subroutines called are AT62, DETIT, MATIT, DEFINE, and PREDICT. Key parameters used are KVAR, IGAIN, DEVEL, DH, DRHO, DELGAIN, W, H, DENS, ITEST, P1, P2, PP, PPF, NORD, NFINE, IDEF, XMACH, CONFAC1, CONFAC2, CONFAC3, XV, and IPRINT.

Subroutine AT62.- Subroutine AT62 determines the density and speed of sound at a specified altitude. Key parameters used are H*CONFAC2, DENS/CONFAC3, and V*CONFAC1/XMACH.

Subroutine PREDICT.- Subroutine PREDICT fits a polynomial of order $n \leq 3$ to $n + 1$ points and evaluates the function at a specified value. It is used to extrapolate for $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$ at the current value of the independent parameter (velocity, altitude, density, or gain) based upon the values of the corresponding $n + 1$

converged roots. It is also used to interpolate for the precise point at which a change in stability occurs. Key parameters used are P1, P2, XVC, XV, PP, PPF, and NORD.

Subroutines FUN1 and FUN.- In subroutine FUN1 and FUN, the final matrix model is formed and the value of its determinant is obtained corresponding to the current estimate $\left(\hat{s}_{IN_{n+1}}\right)$ of the root λ_{IN}. Primary subroutines called are AERO, AEROSP, CXDEV, and SCHEDUL. Key parameters used are VV, DENS, PROOT, SCHEDUL, IDAMP, ISPLANE, ICONSYS, IDMULT, and ICSACT.

Subroutine DEFINE.- Subroutine DEFINE uses PREDICT to interpolate for the precise value of the independent variable (velocity, altitude, density, or gain) at which a change in stability occurs. (See section "Precise determination of point at which change in stability occurs.") Subroutine called is PREDICT. Key parameter used is V.

Subroutine SCHEDUL.- Subroutine SCHEDUL provides for scheduled compensation of the signals from the sensors prior to sending them to the actuators. It also provides for the inclusion of a phase error into the compensation for one control-sensor pair. See section "Compensation Options" and appendix B. Key parameters used are PROOT, PHASE, IPH, JPH, XMACH, and QBAR.

Subroutine AERO.- For ISPLANE = 0, aerodynamic forces as a function of k = Im(p) are determined in subroutine AERO. Subroutines called are IUNI and CSIUNI. Key parameters used are IM(PROOT) and KFIT.

Subroutine AEROSP.- For ISPLANE ≠ 0, the aerodynamic forces are evaluated at p in subroutine AEROSP by using the p-plane approximations generated in overlay PPLANE. Key parameter used is PROOT.

Subroutine CXQR.- The eigenvalues of a complex matrix are determined by subroutine CXQR along with selected eigenvectors.

Subroutine MATINV.- Subroutine MATINV determines the inverse of a real matrix.

Subroutine CXDEV.- Subroutine CXDEV computes the determinant of a complex matrix.

Subroutine IUNI.- Subroutine IUNI performs linear or quadratic interpolation needed to obtain the aerodynamic forces at a particular value of k, depending upon KFIT = ±1 or ±2, respectively.

Subroutine CSIUNI.- Subroutine CSIUNI performs cubic spline interpolation needed to obtain the aerodynamic forces at a particular value of p if KFIT = ±3.

Overlay AEROPLT

Plots are constructed which show how the oscillatory aerodynamic forces vary with reduced frequency and depict how the p-plane approximation fits the data.

Primary subroutines called are APLOT, MULTPT, IUNI, and CSIUNI. Key parameters are defined as follows:

ISPLANE
- = 0 Plot of particular element of aerodynamic force matrix shows input data and curve defined by interpolating for points between these data.
- ≠ 0 Additional curve showing the p-plane approximation is displayed.

MODEA(I,J)
- = 1 Plot of (I,J) aerodynamic force element is made.
- = 0 (default) no plot is made for (I,J)th element.

KMIN†, KMAX† Minimum and maximum reduced frequencies for which aerodynamic forces are available. These set the range of reduced frequencies over which the data are plotted.

IAPLT Indicates not only that aerodynamic data are to be plotted but also indicates which type of curves used to fit data are to be plotted.

= 1 Plot interpolated curves, depending upon KFIT, and p-plane approximation if ISPLANE ≠ 0.

= -1 Plot only interpolated curves regardless of ISPLANE.

KFIT Indicates type of interpolation to be performed in obtaining aerodynamic forces at a particular frequency. When KFIT is negative, $kQ'(k)$ will be plotted for the imaginary part of the aerodynamic force (see the unsteady aerodynamic force subsection).

= ±1 Linear.

= ±2 Quadratic.

= ±3 Cubic spline.

Subroutine APLOT.- KMIN and KMAX are determined in subroutine APLOT. Aerodynamic forces at 50 intermediate values of k between KMIN and KMAX are also determined by using interpolation and PPLANE approximating functions (if ISPLANE ≠ 0 and IAPLT > 0). Plot routine MULTPT is called. Subroutines called are IUNI, CSIUNI, and MULTPT. Key parameters used are all those defined for overlay AEROPLT.

Subroutine MULTPT.- Subroutine MULTPT actually calls the plot labeling routines, line plot routines, axes routines, etc., and creates the plot vector file which defines the actual plots. These plot vector files can either be sent to the screen immediately for interactive plotting or simply saved and plotted after execution via any plotting device available. Subroutines called are library plotting routines for generating axes, drawing lines, labeling, etc.

---

†Internally generated parameter.

Subroutine IUNI.- Subroutine IUNI performs linear or quadratic interpolation for the aerodynamic forces.

Subroutine CSIUNI.- Subroutine CSIUNI performs cubic spline interpolation for the aerodynamic forces if KFIT = 3.

Overlay PKPLOT

Plots are constructed which portray the stability characteristics of the aircraft by using data generated in overlay PKFLUT if IPKPLT ≠ 0 during its execution. All available sets of data are listed if IPKPLT > 0. Also in overlay PKPLOT, the option of removing unwanted sets of characteristic root data is available. Primary subroutines called are PKPLT, PLOT, RLPLT, SCALE, and DELPLOT. Key parameters are defined as follows:

INTERAC† Indicates whether PKPLOT is being executed in interative mode. INTERAC is true if ICASE > 0.

- .T. All input is from file INPUT and output is to file OUTPUT. These are usually assigned to the keyboard and screen, respectively.
- .F. All input is from file TAPE2, and output except plots is to file TAPE6.

IPKPLT Integer indicating which type of characteristic root plots to generate and whether modifications to labels, scaling, and other plot parameters are to be allowed on a plot-per-plot basis.

- >0 Modifications are allowed.
- <0 Only the last set of data is plotted and no changes to plot parameters are allowed during generation of plots.
- = ±1 Generate plots of root loci versus velocity, altitude, or density as determined by KVAR = 1, 2, or 3, respectively, or versus corresponding dynamic pressure if KVAR < 0, or versus gain if IGAIN = 1 during root determination.
- = ±2 Generate plots of ζ and ω_n versus velocity, altitude, or density depending upon KVAR during root determination (or corresponding to dynamic pressure if KVAR < 0) or versus gain if IGAIN = 1 during root determination.
- = ±3 Generate both (ζ, ω_n) and root loci plots.

IDELPLT

- = 1 Delete unwanted characteristic root data from data storage file TAPE1.
- = 0 Do not delete any data.

---

† Internally generated parameter.

ICOUNT — Number of sets of characteristic root data to be plotted concurrently out of total number of sets of data generated during different cases.

IPLT and IPLT2 — Array of numbers indicating which sets of characteristic root data are to be plotted or saved, respectively.

ICOMBIN — Integer indicating whether to combine plots of different sets of characteristic root data into one plot (all on same page) or to generate plots for separate pages.

= 1 Combine plots.

= 0 Do not combine plots.

PTITLE — 80-character description of data being plotted. When plots are being combined, this will appear on plot being generated in lieu of the original plot titles.

IDASH — Indicates type of line used to connect points in data currently being plotted. This may be changed for each set of data being plotted.

= 0 ______________________ (solid line)

= 1 _ _ _ _ _ _ _ _ _ _ _

= 2 ________ _ ________

= 3 _______ _ _ _______

= 4 _______ _ _ _ _______

= 5 ____ ____ _ ____ ____

= 6 ____ ____ _ _ ____ ____

= 7 ____ ____ _ _ _ ____ ____

ISAME — Integer indicating how many of next sets of characteristic root data are to be plotted by using same PLOTP namelist data as currently being read-in, including current set. (Default = 1). Program will not request another PLOTP namelist to be input until n = ISAME sets of data have been plotted.

KVAR — Sign of KVAR indicates whether data are to be plotted versus velocity, altitude, or density ($\text{KVAR} > 0$) or versus dynamic pressure ($\text{KVAR} < 0$).

MODEPK(I) — Integer indicating whether to generate a plot showing Ith characteristic root variation.

= 1 Generate plot of data (Default)

= 0 Bypass Ith root information.

XRLORG, XRLEND, XRLDEL, and NDXRL — Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for abscissa in root-locus plots. (Optional)

YRLORG, YRLEND, YRLDEL, and NDYRL — Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in root-locus plots. (Optional)

VORG, VEND, VDEL, and NDV — Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for abscissa in ζ and ω_n plots. (Optional)

FORG, FEND, FDEL, and NDF — Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in ω_n plots. (Optional)

GORG, GEND, GDEL, and NDG — Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in ζ plots.

IOP — Integer indicating whether to continue plotting. This option allows user to generate additional plots with characteristic root data that either were not included before or for which changes in plots are desired.

= 1 Continue plotting.

= 0 Return to main program and continue with new case or terminate execution.

NFR† — Total number of roots traced in current data set being plotted.

NVR† — Total number of increments in independent variable for current data set

IDYNAM† — Indicates whether to convert to dynamic pressure rather than plot original independent variable.

= 1 Convert to dynamic pressure.

= 0 Do not convert.

---

†Internally generated parameter.

Subroutine PKPLT.- Subroutine PKPLT actually reads in the plot data, converts the independent variable to dynamic pressure, if desired, and calls plotting routines to generate desired plots. Subroutines called are PLOT and RLPLT. Key parameters used are IPKPLT, NFR, NVR, MODEPK, and IDYNAM.

Subroutine PLOT - Entry points PLOT1 and PLOT2.- Subroutine PLOT generates the plots of ζ and ω_n versus the corresponding independent variable desired (velocity, altitude, density, dynamic pressure, or gain). Subroutines called are SCALE and library plotting routines for generating axes, drawing lines, labeling, etc. Key parameters used are ICOMBIN, IDASH, MODEPK, NFR, NVR, VORD, VEND, VDEL, NDV, FORG, FEND, FDEL, NDF, GORG, GEND, GDEL, and NDG.

Subroutine RLPLT.- Subroutine RLPLT generates the root locus plot. Initial points are identified with the mode number. Every 10th point is identified with a + for nongain variation plots. Gain points are identified on the gain variation plots, which are 0, .5, 1., and 2. times the nominal gain incorporated by input into $(T'_L)_{i,j}$. Subroutines called are IUNI, SCALE, and library plotting routines for generating axes, drawing lines, labeling, etc. Key parameters used are ICOMBIN, IDASH, MODEPK, NFR, NVR, XRLORG, XRLEND, XRLDEL, NDXRL, YRLDRG, YRLEND, YRLDEL, and NDYRL.

Subroutine SCALE.- Subroutine SCALE determines an appropriate power of 10 to be used for tic marks on axes which are 2., 4., 5., or 10. in. in length and returns the corresponding scale factor required to scale the data.

Subroutine DELPLOT.- Subroutine DELPLOT deletes unwanted sets of characteristic root data, generated by using overlay PKFLUT, from data storage file, TAPE1. Key parameter used is IPLT2.

INPUTS FOR SAMPLE CASES

The TAPE2 inputs required to generate the plots shown in figures 8 to 16 are presented in this appendix. These inputs exercise the major options of STABCAR. They are presented to serve as a guide for users to follow in preparing input data. Reference should be made to table II for the definitions of the input variables.

Cases 1 to 4 (Figs. 8 to 11)

The inputs for sample cases 1 to 4 (figs. 8 to 11) are presented as a unit in table GI. These cases are run in a batch mode (ICASE < 0). Case 1, which generates the data for figure 8, has ICASE = -2. This means that TAPE5 (table III) must be local in order to make the airplane generalized aerodynamic forces available. In cases 2 to 4, ICASE = -1 means that a batch mode of operation is in effect and all data must be obtained from TAPE1 (table V), which was generated in the previous case and from TAPE2. Note that in case 3 the array S2I ($\hat{s}_{i_2}$) has been set to zero. This was done because, with IPS = 0 (the default), S2I contains the roots found at the end of the last case having IFLUT = 1. That set of roots is for h = 0 (high dynamic pressure), whereas case 3 begins at a low dynamic pressure. Setting S2I to zero restarts the estimation process; thereby, convergence difficulties are avoided that would have occurred since S2I is used in starting the matrix iteration (IOPT2 = 1) search for initial estimates. (See appendix D.) These cases could have been run interactively with the plots being returned during job execution by making ICASE > 0 and loading LIBFTEK and Techtronix interface plotting routines, which support interactive graphics, or their equivalent.

Case 5 (Fig. 12)

Tables GII, GIII, and GIV contain the inputs required to generate figure 12.

The modal deflections at two accelerometer locations are needed to implement the flutter control law. TAPE10 (table IV) which contains the mode shape data, TAPE5, and TAPE2 are made local. For clarity a new TAPE2 is generated; it is shown in table GII. Alternatively, one could have modified the initial TAPE2 corresponding to case 1.

The modal deflections determined in this run are differenced and this difference is read onto TAPE5. None of these operations are shown. Next a run is made which defines the control law. The inputs are shown in table GIII.

At this point the user saves TAPE1 which now contains the control-law dynamics and selected sensor data. This TAPE1 will be employed not only to complete case 5 but also as part of the input for cases 6, 8, and 9. Table GIV presents TAPE2 input required to complete case 5. These runs could have been done interactively by setting ICASE = 1. Note that ISCHEDUL (3,1) = 1 which means that the scheduled control law factor defined in appendix B and figure 7 will be employed.

Case 6 (Fig. 13)

The input required to generate figure 13 is now shown. The TAPE1 employed is as it existed prior to making the last two runs in case 5. Thus, the aircraft is modeled including two rigid-body, six elastic, and one control mode, the corresponding sensor deflections, and the definition of the flutter suppression system control law. ICASE is set to +1 in any convenient TAPE2. For ICASE = +1 none of the rest of TAPE2 is employed. The values of all parameters are contained on the TAPE1 which must be local. The interactive session then proceeds as shown in table GV.

Case 7 (Fig. 14)

The interative session during which figure 14 was generated is shown in table GVI. This figure shows the effect of phase errors in the control system upon the elastic mode gain root loci. The TAPE1 created in case 6 is employed as the primary source of input. Three runs are made. The first two of these runs determine stability characteristics by using the p-p method after phase errors have been introduced. The third run combines the root loci plots from these two runs with phase errors with the corresponding plot from case 6 which had no phase errors.

Case 8 (Fig. 15)

Figure 15 presents gain root loci for the control-law poles (see fig. 7) and one of the open-loop actuator poles (s = -401 rad/sec). The loci are shown at the open-loop flutter point for $N_{Ma} = 0.86$. Table GVII shows the interactive session which created the figure. The TAPE1 saved prior to running the last two segments of case 5 is the primary source of input. Modifications are made with the TAPE2 shown at the top of the table.

The initial segment of this session traces one elastic root, mode 8, one actuator root, mode 13, and three control-law roots from zero gain to a gain of 0.07. The scheduled pole is not traced because this denominator is not cleared and, for a gain of zero, would result in division by zero in equation (2). Note that IDMULT = 1 in TAPE2 which means that the nonscheduled control-law denominator, actuator denominator, and sensor denominators are cleared in equation (2). The card image containing the three zeros tells the program, when IPKPLT = 1, that no plots are to be generated in this initial batch (ICASE = -1) run. This first segment is generated knowing in advance that a series of runs will be required with the resulting plots combined into one composite plot. The mode number identifiers (fig. 15) are only placed on the first plot set. Consequently, one plot set - they can be plotted in any order - should contain all the modes starting from the initial value of the independent variable. The +1 ending TAPE2 makes ICASE > 0 so that the interactive mode of operation is invoked.

A series of interactive runs are made to generate the gain loci by using the p-k method. Note the first interactive run where IPS = 1. This value for IPS means that the initial estimates at the start of the last execution with IFLUT = 1 will be employed as initial estimates in the current run. Note also the runs where root 12 is being determined as a function of gain. This is an illustration of the case where two real roots combine to form a complex pair. The next to last run is a batch run where roots 11 and 14 are found with the p-p method. Finally, the last case combines the plot sets to form figure 15.

Case 9 (Fig. 16)

The inputs required to generate figure 16 are shown in table GVIII. Figure 16 is an open-loop locus of the short period root with altitude. Three interactive runs are made. The TAPE1 generated in case 2 is made local and ICASE is set to +1 in a TAPE2. The first two runs obtain the root locus for the p-k and the p-p method. The third run combines the two plot sets to produce figure 16.

TABLE GI.- INPUT FOR SAMPLE CASES 1 TO 4

```
-2
CASE 1.  P-PLANE FIT AND AERODYNAMIC FORCES PLOTS. (FIG 8)
$INPUT NM=16,NR=2,NC=4,ICSACT=1,NK=12,KFIT=2,
ISPLANE=3,NCOEF=7,NKK(1)=2*5,ICOF(1,1)=0,0,1,0,1,ICOF(1,2)=1,1,
NCOL(1)=0,BN(1,1)=0.0939,0.188,0.2816,0.376,
CSURFEF(3)=1.,ISELECT=1,IAPLT=1,C=.596138,
MODEA(3,3)=1,MODEA(5,3)=1,MODEA(8,3)=1,
MODEA(3,5)=1,MODEA(5,5)=1,MODEA(8,5)=1,
MODEA(3,8)=1,MODEA(5,8)=1,MODEA(8,8)=1,
MODEA(3,15)=1,MODEA(5,15)=1,MODEA(8,15)=1$
$SELECT NMODES=9,NRNEW=2,NCNEW=1,NOC(1)=1,2,3,4,5,6,7,8,15$
-1
CASE 2.  OPEN LOOP STABILITY CHARACTERISTICS VERSUS ALTITUDE,DV=0.5 (FIG 9)
$INPUT IAPLT=0,ISPLANE=0,IFLUT=1,NV=60,ICSACT=1,
ISELECT=-1,
DV=0.5          ,HO=30.           ,KVAR=2           ,EPSI=1.E-6,
XMACH=0.86      ,CONFAC1=3.28084  ,CONFAC2=3280.84  ,CONFAC3=515.3787,
NIT=26          ,NCUT=25          ,NFINE=4          ,IDAMP=2         ,
IOPT2=1         ,IPRINT=5         ,IPKPLT=-3        ,MODEPK(1)=8*1   ,
VORG=0.         ,VEND=30.         ,VDEL=10          ,NDV=10          ,
FORG=0          ,FEND=60          ,FDEL=20          ,NDF=4           ,
GORG=-1.        ,GEND=1.0         ,GDEL=0.5         ,NDG=5           ,
XRLORG=-300.    ,XRLEND=100.      ,XRLDEL=100.      ,NDXRL=5         ,
YRLORG=0.       ,YRLEND=400.      ,YRLDEL=100.      ,NDYRL=4         ,
 ITRACE(1)=8*1   $
$SELECT NMODES=8,NRNEW=2,NCNEW=0,NOC(1)=1,2,3,4,5,6,7,8$
-1
CASE 3.  OPEN LOOP CHARACTERISTIC ROOTS VERSUS DENSITY,DV=0.025 (FIG 10)
$INPUT NV=28      ,DV=0.025  ,RHO0=0.05 ,V0=269.64 , S2I=25*(0,0),
KVAR=3            ,VORG=0.0  ,VEND=0.80 ,VDEL=0.20 ,
NDV=10            ,GORG=-0.2 ,GEND=0.6  ,GDEL=0.2  ,
NDG=10            ,XRLORG=-120,XRLEND=20,XRLDEL=20 ,
NDXRL=4           $
-1
CASE 4.  OPEN LOOP STABILITY CHARACTERISTICS VERSUS DYNAMIC PRESSURE. (FIG 11)
$INPUT IFLUT=0,KVAR=-3    ,VORG=0.    ,VEND=30000., VDEL=5000.,  NDV=10 $
0
```

TABLE GII.- MODAL DEFLECTIONS AT ACCELEROMETER LOCATIONS

```
-2
CASE 5A.  MODAL DEFLECTIONS AT ACCELEROMETER LOCATIONS
$INPUT NM=16,NR=2,NC=4,NS=2,IFLUT=0,IPKPLT=0,
ISENSE=1,INTERP=1,ISYM=1,NPMX=60,IPLATE(1)=1,
NSECTNS=3,NNODES=100,ISS(1,1)=1,60,61,92,93,100,
XCG=6.71576,X0(1)=6.4470,0.0,9.1328,Y0(3)=.2515,
R0(1)=1.7799,1.570796,.5175  $
$SENLOC XS(1)=7.1438,7.3363,YS(1)=2.083,2.134,
ITYPE(1)=1,1,NSS(1)=1,1   $
0
```

TABLE GIII.- SELECTION OF SENSOR DATA AND DEFINITION OF CONTROL LAW

```
-2
CASE 5B.  SELECTION OF SENSOR DATA, AND DEFINITION OF CONTROL LAW.
$INPUT NM=16,NR=2,NC=4,ICSACT=1,NK=12,KFIT=2,NS=1,
ISPLANE=3,NCOEF=7,NKK(1)=2*5,ICOF(1,1)=0,0,1,0,1,ICOF(1,2)=1,1,
NCOL(1)=0,BN(1,1)=0.0939,0.188,0.2816,0.376,
CSURFEF(3)=1.,ISELECT=1,IAPLT=0,C=.596138,
MODEA(3,3)=1,MODEA(5,3)=1,MODEA(8,3)=1,
MODEA(3,5)=1,MODEA(5,5)=1,MODEA(8,5)=1,
MODEA(3,8)=1,MODEA(5,8)=1,MODEA(8,8)=1,
MODEA(3,15)=1,MODEA(5,15)=1,MODEA(8,15)=1,
DV=0.5 ,NV=1      ,HO=4.572          ,KVAR=2            ,EPSI=1.E-6,
XMACH=0.86       ,CONFAC1=3.28084  ,CONFAC2=3280.84   ,CONFAC3=515.3787,
NIT=26           ,NCUT=25          ,NFINE=4           ,IDAMP=2          ,
IOPT2=1          ,IPRINT=5         ,IPKPLT=-3         ,MODEPK(1)=8*1    ,
VORG=0.          ,VEND=30.         ,VDEL=10           ,NDV=10           ,
FORG=0           ,FEND=60          ,FDEL=20           ,NDF=4            ,
GORG=-1.         ,GEND=1.0         ,GDEL=0.5          ,NDG=5            ,
XRLORG=-300.     ,XRLEND=100.      ,XRLDEL=100.       ,NDXRL=5          ,
YRLORG=0.        ,YRLEND=400.      ,YRLDEL=100.       ,NDYRL=4           ,
ITRACE(1)=8*1   ,
ICONSYS=1  ,GN(3,1)=0.  ,    ISCHDUL(3,1)=1,
ISENSE=0     ,IFLUT=0 ,IPKPLT=0 ,INTERP=0 , ISYM=1 ,NPMX=60 ,ICSREAD=1,
NSECTNS=3 ,NNODES=100 , ISS(1,1)=1,60,61,92,93,100,IPLATE(1)=1,
XCG=6.71576,XO(1)=6.4470,0.0,9.1328,
YO(3)=0.2515,RO(1)=1.7799,1.570796,.5175 $
$SENLOC XS(1)=7.1438,7.3363,YS(1)=2.083,2.134,
ITYPE(1)=1,1  ,NSS(1)=1,1 $
$SENINP ISDYN=0  ,IORD(1)=2  $
3 1
$FILTIN IFILTER=6,6,6,6,1,
AFN(1,1)=250.,AFD(1,1)=250.,1.,
AFN(1,2)=2.079E5,220.,1.  ,AFD(1,2)=2.704E5,400.,1.,
AFN(1,3)=3.8416E4,40.,1.  ,AFD(1,3)=3.8612E4,100.,1.,
AFN(2,4)=-1.1554  ,AFD(1,4)=2.,1.  $
0 0
$ACTINP IACT(3)=1,AACTN(1,3)=6.583E14,
AACTD(1,3)=6.583E14,2.348E12,2.9939E9,3.607E6,1.7373E3,1.  $
$SELECT NMODES=9,NRNEW=2,NCNEW=1,NOC(1)=1,2,3,4,5,6,7,8,15$
0
```

TABLE GIV.- INPUT FOR SAMPLE CASE 5

```
 -1
 CASE 5C.  OPEN LOOP STABILITY VERSUS VELOCITY (REDUCED ORDER MODEL, FIGURE 12A)
 $INPUT ISELECT=-1,ICSREAD=0,KVAR=1,IFLUT=1,ISPLANE=0,
 RHO0=0.6101,V0=50.,NV=75,DV=4,IAPLT=0,IPS=0,S2I=25*(0,0),
 IPKPLT=-3,VORG=0.,VEND=350,VDEL=50.,
 NDV=10,GORG=-1.,GEND=1.0,GDEL=0.5,NDG=5  $
 $SELECT NMODES=4,NRNEW=0,NCNEW=1,NOC(1)=3,5,8,15  $
-1
CLOSED LOOP STABILITY VS VELOCITY (REDUCED ORDER MODEL, FIGURE 12B)
 $INPUT GN(3,1)=1.,V0=50.,IPS=0,S2I=25*(0,0) $
 0
```

TABLE GV.- INTERACTIVE EXECUTION OF SAMPLE CASE 6

```
/GET,TAPE2
/GET,TAPE1=T1CNTRL
/GET,STABCAR
/STABCAR
  TYPE IN TITLE
? CASE 6. ELASTIC MODE GAIN ROOT LOCI (0 TO 2.5 BY .025) P-K METHOD (FIG 13)
  TYPE IN /INPUT/
?  $INPUT KVAR=1,RHO0=.771087,V0=277.1625,IGAIN=1,GN(3,1)=0,IFLUT=1,
?  DELGAIN(3,1)=.025,NV=100,IPRINT=0,IPKPLT=-1,ITRACE=2*0,6*1,IOPT2=1,
?  IPS=0,S2I=25*(0.0,0.0),ISPLANE=0$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/08/18. TIME:  08.20.21.
 CASE 6. ELASTIC MODE GAIN ROOT LOCI (0 TO 2.5 BY .025) P-K METHOD (FIG 13)
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105350           P-K FLUTTER ANALYSIS
  INITIAL ROOTS DETERMINED BY MATRIX ITERATION
 !! ZERO ROOT, NOT USED !!
                               UNSTABLE ROOT



 FIELD LENGTH INCREASED TO 107664
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .7710870E+00
          AN INITIAL VELOCITY OF    .2771625E+03
          AN INITIAL ALTITUDE OF    .4572000E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2961709E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL      S1                                                    S2
 MODE
    3   (     -.20604579E+01,      .10519848E+03)   (     -.20812706E+01,      .10626109E+03)
    4   (      .23192948E+02,      .12943256E+03)   (      .23427220E+02,      .13073996E+03)
    5   (     -.31871159E+01,      .19215485E+03)   (     -.32193090E+01,      .19409580E+03)
    6   (     -.10549623E+03,      .17415705E+03)   (     -.10656184E+03,      .17591622E+03)
    7   (     -.28774737E+02,      .22162978E+03)   (     -.29065391E+02,      .22386846E+03)
    8   (     -.43581485E+01,      .24541669E+03)   (     -.44021702E+01,      .24789564E+03)
 A CHANGE STABILIY OCCURRED IN ROOT      4 SCAN FOLLOWS
  FLUTTER VALUES
     SPEED            277.16
     ALTITUDE         4.5720
     DENSITY          .77109
     GAIN             .68787
         MODE                         FREQUENCY          DAMPING
          4                           144.26            0.
```

TABLE GV.- Continued

```
  CURRENTPLOT FILE TITLE IS:
CASE 6. ELASTIC MODE GAIN ROOT LOCI (O TO 2.5 BY .025) P-K METHOD (FIG 13)

 TYPE IN NEW TITLE?
? SAME

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? CASE 6. ELASTIC MODE GAIN ROOT LOCI (O TO 2.5 BY .025) P-P METHOD FIG 13)
  TYPE IN /INPUT/
?  $INPUT ISPLANE=1,GN(3,1)=0,S2I=25*(0.0,0.0)$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/08/18. TIME:  08.31.26.
 CASE 6. ELASTIC MODE GAIN ROOT LOCI (O TO 2.5 BY .025) P-P METHOD FIG 13)
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 102703           P-K FLUTTER ANALYSIS
  INITIAL ROOTS DETERMINED BY MATRIX ITERATION
 !! ZERO ROOT, NOT USED !!
                             UNSTABLE ROOT

                             UNSTABLE ROOT

 FIELD LENGTH INCREASED TO 105217
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .7710870E+00
          AN INITIAL VELOCITY OF    .2771625E+03
          AN INITIAL ALTITUDE OF    .4572000E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2961709E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL     S1                                              S2
 MODE
    3   (     -.23686653E+01,      .10529351E+03)   (     -.23925912E+01,      .10635708E+03)
    4   (      .24710514E+02,      .13753500E+03)   (      .24960116E+02,      .13892424E+03)
    5   (     -.31783179E+01,      .19215062E+03)   (     -.32104221E+01,      .19409154E+03)
    6   (     -.97411393E+02,      .16584347E+03)   (     -.98395346E+02,      .16751866E+03)
    7   (     -.27301601E+02,      .21998447E+03)   (     -.27577375E+02,      .22220653E+03)


    8   (     -.43669954E+01,      .24544068E+03)   (     -.44111065E+01,      .24791988E+03)
 A CHANGE STABILIY OCCURRED IN ROOT      4 SCAN FOLLOWS
  FLUTTER VALUES
     SPEED          277.16
     ALTITUDE       4.5720
     DENSITY        .77109
     GAIN           .71540
```

TABLE GV.- Concluded

```
  CURRENT PLOT FILE TITLE IS:
CASE 6. ELASTIC MODE GAIN ROOT LOCI (O TO 2.5 BY .025) P-P METHOD FIG 13)

 TYPE IN NEW TITLE?
? SAME

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? FIGURE 13. ELASTIC MODE GAIN ROOT LOCI AT VD (M=.86,H=4.572)
  TYPE IN /INPUT/
?  $INPUT IFLUT=0,IPKPLT=1$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/08/18. TIME:  08.39.30.
 FIGURE 13. ELASTIC MODE GAIN ROOT LOCI AT VD (M=.86,H=4.572)
 MACH #=    .8600

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 1
 PLOT SET  TYPE OF VARIATION      # MODES                   PLOT TITLE


    1        GAIN     G$D 3, 1      6      CASE 6. ELASTIC MODE GAIN ROOT LOCI (O TO 2.5 BY .025) P-K METHOD (FIG 13)
    2        GAIN     G$D 3, 1      6      CASE 6. ELASTIC MODE GAIN ROOT LOCI (O TO 2.5 BY .025) P-P METHOD (FIG 13)
 " TYPE IN NUMBER OF PLOT SETS TO BE PLOTTED,"
 " CORRESPONDING PLOT SET #'S, AND WHETHER TO COMBINE THEM"
? 2 1 2 1
 " TYPE IN NEW PLOT TITLE FOR COMBINED PLOTS"
? SAME
 " TYPE IN /PLOTP/ -KVAR, IDASH, MODEPK, ETC."
?  $PLOTP MODEPK=6*1,XRLORG=-200,XRLEND=50,XRLDEL=50,NDXRL=1,
?  YRLORG=0.,YRLEND=500,YRLDEL=100,NDYRL=2$


 FIELD LENGTH OVERLAY(5,0) IS 101105          PK FLUTTER PLOTS
 " TYPE IN /PLOTP/ -KVAR, IDASH, MODEPK, ETC."
?  $PLOTP IDASH=1$


 FIELD LENGTH OVERLAY(5,0) IS 101105          PK FLUTTER PLOTS
 " DO YOU WISH TO CONTINUE PLOTTING? (1/0)"
? 0

 CONTINUE
? 0
```

TABLE GVI.- INTERACTIVE EXECUTION OF SAMPLE CASE 7

```
/GET,TAPE2
/GET,TAPE1=f1CASE6
/GET,STABCAR
/STABCAR
  TYPE IN TITLE
? FIG 14. EMODE GN ROOT LOCI PHASE=45 (0 TO 2.5 BY .025) P-P
  TYPE IN /INPUT/
?  $INPUT IFLUT=1,ISPLANE=1,GN(3,1)=0,DELGAIN(3,1)=.025,NV=100,
?  IOPT2=1,IPH=3,JPH=1,PHASE=45.,IPKPLT=-1,ITRACE(1)=2*0,6*1,0,
?  S2I=25*(0.0,0.0)$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/08/18. TIME:  12.44.27.
 FIG 14. EMODE GN ROOT LOCI PHASE=45 (0 TO 2.5 BY .025) P-P
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 102703           P-K FLUTTER ANALYSIS
  INITIAL ROOTS DETERMINED BY MATRIX ITERATION
 !! ZERO ROOT, NOT USED !!
                             UNSTABLE ROOT

                             UNSTABLE ROOT



 FIELD LENGTH INCREASED TO 105217
1THIS CASE IS FOR
          AN INITIAL DENSITY   OF    .7710870E+00
          AN INITIAL VELOCITY OF    .2771625E+03
          AN INITIAL ALTITUDE OF    .4572000E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2961709E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL      S1                                               S2
 MODE
    3   (     -.23686653E+01,      .10529351E+03)   (     -.23925912E+01,      .10635708E+03)
    4   (      .24710514E+02,      .13753500E+03)   (      .24960116E+02,      .13892424E+03)
    5   (     -.31783179E+01,      .19215062E+03)   (     -.32104221E+01,      .19409154E+03)
    6   (     -.97411393E+02,      .16584347E+03)   (     -.98395346E+02,      .16751866E+03)
    7   (     -.27301601E+02,      .21998447E+03)   (     -.27577375E+02,      .22220653E+03)
    8   (     -.43669954E+01,      .24544068E+03)   (     -.44111065E+01,      .24791988E+03)
 A CHANGE STABILIY OCCURRED IN ROOT     4 SCAN FOLLOWS
  FLUTTER VALUES
     SPEED           277.16
     ALTITUDE        4.5720
     DENSITY         .77109


     GAIN            .76085
        MODE                           FREQUENCY           DAMPING
         4                             127.08              0.
```

TABLE GVI.- Continued

```
  CURRENT PLOT FILE TITLE IS:
FIG 14. EMODE GN ROOT LOCI PHASE=45 (0 TO 2.5 BY .025) P-P

 TYPE IN NEW TITLE?
? SAME

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? FIG 14. EMODE GN ROOT LOCI PHASE=-45 (0 TO 2.5 BY .025) P-P
  TYPE IN /INPUT/
?  $INPUT PHASE=-45,GN(3,1)=0.,IPKPLT=1,S2I=25*(0.0,0.0)$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/08/18. TIME:  13.02.05.
 FIG 14. EMODE GN ROOT LOCI PHASE=-45 (0 TO 2.5 BY .025) P-P
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 102703           P-K FLUTTER ANALYSIS
  INITIAL ROOTS DETERMINED BY MATRIX ITERATION
 !! ZERO ROOT, NOT USED !!
                              UNSTABLE ROOT

                              UNSTABLE ROOT



 FIELD LENGTH INCREASED TO 105217
1THIS CASE IS FOR
          AN INITIAL DENSITY   OF    .7710870E+00
          AN INITIAL VELOCITY OF    .2771625E+03
          AN INITIAL ALTITUDE OF    .4572000E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2961709E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL      S1                                           S2
 MODE
    3   (    -.23686653E+01,     .10529351E+03)   (     -.23925912E+01,     .10635708E+03)
    4   (     .24710514E+02,     .13753500E+03)   (      .24960116E+02,     .13892424E+03)
    5   (    -.31783179E+01,     .19215062E+03)   (     -.32104221E+01,     .19409154E+03)
    6   (    -.97411393E+02,     .16584347E+03)   (     -.98395346E+02,     .16751866E+03)
    7   (    -.27301601E+02,     .21998447E+03)   (     -.27577375E+02,     .22220653E+03)
    8   (    -.43669954E+01,     .24544068E+03)   (     -.44111065E+01,     .24791988E+03)
 A CHANGE STABILIY OCCURRED IN ROOT     4 SCAN FOLLOWS
  FLUTTER VALUES
     SPEED          277.16
     ALTITUDE       4.5720
     DENSITY        .77109
     GAIN           1.5883
         MODE                          FREQUENCY         DAMPING
          4                            193.21           0.
```

TABLE GVI.- Concluded

```
 CURRENT PLOT FILE TITLE IS:
FIG 14. EMODE GN ROOT LOCI PHASE=-45 (0 TO 2.5 BY .025) P-P

 TYPE IN NEW TITLE?
? SAME

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 1
 PLOT SET  TYPE OF VARIATION      # MODES                 PLOT TITLE


    1       GAIN     G$D 3, 1      6     CASE 6. ELASTIC MODE GAIN ROOT LOCI (0 TO 2.5 BY .025) P-K METHOD (FIG 13)
    2       GAIN     G$D 3, 1      6     CASE 6. ELASTIC MODE GAIN ROOT LOCI (0 TO 2.5 BY .025) P-P METHOD (FIG 13)
    3       GAIN     G$D 3, 1      6     FIG 14. EMODE GN ROOT LOCI PHASE=45 (0 TO 2.5 BY .025) P-P
    4       GAIN     G$D 3, 1      6     FIG 14. EMODE GN ROOT LOCI PHASE=-45 (0 TO 2.5 BY .025) P-P
 " TYPE IN NUMBER OF PLOT SETS TO BE PLOTTED,"
 " CORRESPONDING PLOT SET #'S, AND WHETHER TO COMBINE THEM"
? 3 2 3 4 1
 " TYPE IN NEW PLOT TITLE FOR COMBINED PLOTS"
? FIG 14. EFFECT OF PHASE ERRORS IN CONTROL SIGNAL UPON ELASTIC AXIS GAIN ROOT LOCI AT VD (P-P ANALYSIS)
 " TYPE IN /PLOTP/ -KVAR, IDASH, MODEPK, ETC."
?  $PLOTP IDASH=0,MODEPK=6*1$


 FIELD LENGTH OVERLAY(5,0) IS 101105          PK FLUTTER PLOTS
 " TYPE IN /PLOTP/ -KVAR, IDASH, MODEPK, ETC."
?  $PLOTP IDASH=2$


 FIELD LENGTH OVERLAY(5,0) IS 101105          PK FLUTTER PLOTS
 " TYPE IN /PLOTP/ -KVAR, IDASH, MODEPK, ETC."
?  $PLOTP IDASH=1$


 FIELD LENGTH OVERLAY(5,0) IS 101105          PK FLUTTER PLOTS
 " DO YOU WISH TO CONTINUE PLOTTING? (1/0)"
? 0

 CONTINUE
? 0
```

TABLE GVII.- INTERACTIVE EXECUTION OF SAMPLE CASE 8

```
/COPY,TAPE2
-1
TRACE 8 + CNTR - SCH 0 TO .07 BY 01 PK
 $INPUT RHOO=.6101,V0=269.64,H0=6.7045,IDMULT=1,ISPLANE=0,IFLUT=1,
 ITRACE=25*0,ITRACE(8)=1,1,0,1,1,1,1,
 IGAIN=1,DELGAIN(3,1)=.01,GN(3,1)=0,NV=7,IOPT2=0,
 S2I=25*(0,0),S1I=25*(0,0),
 S2I(9)=(-1.999,.0001),S2I(13)=(-400,0.),S2I(11)=(-50,196),S2I(14)=(-200,480),
 S1I(9)=(-1.990,.0002),S1I(13)=(-398,0.),S1I(11)=(-49,195),S1I(14)=(-200,482),
 S2I(12)=(-250,0),S1I(12)=(-252,0),S2I(8)=(-41.99,257.8),
 S1I(8)=(-41.57,255.2),
 IPKPLT=+1,IPRINT=7$
0 0 0
+1
 EOI ENCOUNTERED.
/REWIND,TAPE2
$REWIND,TAPE2.
/GET,TAPE1=T1CNTRL
/GET,STABCAR
/STABCAR
1       DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  12.50.44.
 TRACE 8 + CNTR - SCH 0 TO .07 BY 01 PK
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105331           P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 105511
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0         INITIAL     S1                                             S2
 MODE
    8   (     -.41570000E+02,     .25520000E+03)   (     -.41990000E+02,      .25780000E+03)
    9   (     -.19900000E+01,     .20000000E-03)   (     -.19990000E+01,      .10000000E-03)
   11   (     -.49000000E+02,     .19500000E+03)   (     -.50000000E+02,      .19600000E+03)
   12   (     -.25200000E+03,    0.                )   (     -.25000000E+03,    0.                )
   13   (     -.39800000E+03,    0.                )   (     -.40000000E+03,    0.                )
   14   (     -.20000000E+03,     .48200000E+03)   (     -.20000000E+03,      .48000000E+03)
  GAIN FOR CONTROL    3  AND SENSOR    1  IS  0.
      #-ITER     ROOT NBR      ALTITUDE          ZETA           OMEGA-N                  ROOT                          PRESSURE
         3          8       .6704500E+01     .1607713E+00     .4156597E+02    (  -.4198812E+02,   .2577693E+03)      .2217926E+05
         3          9       .6704500E+01     .1000000E+01     .3183099E+00    (  -.2000000E+01,   .4539394E-13)      .2217926E+05
         6         11       .6704500E+01     .2544538E+00     .3127385E+02    (  -.5000000E+02,   .1900316E+03)      .2217926E+05
         1         12       .6704500E+01     .1000000E+01     .3978874E+02    (  -.2500000E+03,   .1004338E-12)      .2217926E+05
         4         13       .6704500E+01     .1000000E+01     .6383778E+02    (  -.4011046E+03,  -.3045945E-08)      .2217926E+05
         1         14       .6704500E+01     .3846154E+00     .8276057E+02    (  -.2000000E+03,   .4800000E+03)      .2217926E+05
```

TABLE GVII.- Continued

```
 GAIN FOR CONTROL     3  AND SENSOR    1  IS  7.000E-02
        5         8      .6704500E+01     .1721935E+00       .4147449E+02      (  -.4487223E+02,   .2566995E+03)        .2217926E+05
        2         9      .6704500E+01     .1000000E+01       .3183012E+00      (  -.1999945E+01,   .1569397E-14)        .2217926E+05
        6        11      .6704500E+01     .3067763E+00       .3181999E+02      (  -.6133406E+02,   .1902906E+03)        .2217926E+05
        8        12      .6704500E+01     .1000000E+01       .2706461E+02      (  -.1700520E+03,  -.1270729E-06)        .2217926E+05
        5        13      .6704500E+01     .1000000E+01       .6930285E+02      (  -.4354426E+03,  -.2950560E-05)        .2217926E+05
        4        14      .6704500E+01     .3813816E+00       .8350224E+02      (  -.2000957E+03,   .4850051E+03)        .2217926E+05
 TYPE IN TITLE
? TRACE EXCEPT 8 11 12 SCH 0 TO 2.5 BY .025  P-K
 TYPE IN /INPUT/


?  $INPUT GN(3,1)=0,DELGAIN(3,1)=.025,NV=100,IPRINT=0,
?  ITRACE(8)=0,1,0,0,0,1,1,IPS=1$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  12.52.26.
 TRACE EXCEPT 8 11 12 SCH 0 TO 2.5 BY .025  P-K
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105466          P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 106636
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF   .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0         INITIAL     S1                                              S2
 MODE
    9   (     -.19900000E+01,      .20000000E-03)   (     -.19990000E+01,     .10000000E-03)
   13   (     -.39800000E+03,     0.               )   (     -.40000000E+03,    0.                )
   14   (     -.20000000E+03,      .48200000E+03)   (     -.20000000E+03,     .48000000E+03)
  CURRENT PLOT TITLE IS:
TRACE EXCEPT 8 11 12 SCH 0 TO 2.5 BY .025  P-K

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? TRACE 8 11 0 TO .2 BY .01  P-K
  TYPE IN /INPUT/
?  $INPUT GN(3,1)=0,DELGAIN(3,1)=.01,NV=20,IPRINT=20,IPS=0,
?  S2I=25*(0,0),S1I=25*(0,0),ITRACE=25*0,ITRACE(8)=1,0,0,1,
?  S2I(8)=(-41.99,255.78),S1I(8)=(-41.57,255.2),
?  S2I(11)=(-50,196),S1I(11)=(-51,196)$
```

TABLE GVII.- Continued

```
1       DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  12.57.59.
 TRACE 8 11 0 TO .2 BY .01  P-K
 MACH #=   .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105346            P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 105502
1THIS CASE IS FOR
          AN INITIAL DENSITY   OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL     S1                                          S2
 MODE
    8   (     -.41570000E+02,     .25520000E+03)   (     -.41990000E+02,     .25578000E+03)
   11   (     -.51000000E+02,     .19600000E+03)   (     -.50000000E+02,     .19600000E+03)
  GAIN FOR CONTROL    3  AND SENSOR    1  IS  0.
      #-ITER    ROOT NBR      ALTITUDE          ZETA            OMEGA-N                   ROOT                          PRESSURE


        6          8      .6704500E+01      .1607713E+00      .4156597E+02     (  -.4198811E+02,    .2577693E+03)      .2217926E+05
        7         11      .6704500E+01      .2544537E+00      .3127385E+02     (  -.5000000E+02,    .1900316E+03)      .2217926E+05
  GAIN FOR CONTROL    3  AND SENSOR    1  IS  2.000E-01
        7          8      .6704500E+01      .2042014E+00      .4107184E+02     (  -.5269661E+02,    .2526243E+03)      .2217926E+05
        6         11      .6704500E+01      .3333303E+00      .3625634E+02     (  -.7593442E+02,    .2147771E+03)      .2217926E+05
  CURRENT PLOT TITLE IS:
TRACE 8 11 0 TO .2 BY .01  P-K

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? TRACE 8 11 .2 TO .3 BY .0025 P-K
  TYPE IN /INPUT/
?  $INPUT DELGAIN(3,1)=.0025,NV=40,IPRINT=40$
1       DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  13.03.22.
 TRACE 8 11 .2 TO .3 BY .0025 P-K
 MACH #=   .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105372            P-K FLUTTER ANALYSIS
```

TABLE GVII.- Continued

```
 FIELD LENGTH INCREASED TO 105646
1THIS CASE IS FOR
           AN INITIAL DENSITY  OF    .6101046E+00
           AN INITIAL VELOCITY OF    .2696413E+03
           AN INITIAL ALTITUDE OF    .6704500E+01, AND
           AN INITIAL DYNAMIC PRESSURE OF    .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
          INITIAL      S1                                              S2
 MODE
   8   (     -.41570000E+02,      .25520000E+03)   (     -.52696606E+02,      .25262430E+03)
  11   (     -.51000000E+02,      .19600000E+03)   (     -.75934422E+02,      .21477715E+03)
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.000E-01
     #-ITER     ROOT NBR      ALTITUDE           ZETA            OMEGA-N                   ROOT                              PRESSURE
        1          8       .6704500E+01      .2042014E+00      .4107133E+02     (  -.5269661E+02,   .2526243E+03)      .2217926E+05
        1         11       .6704500E+01      .3333311E+00      .3625634E+02     (  -.7593459E+02,   .2147771E+03)      .2217926E+05
 !! NON-CONVERGENCE !! ==> # ITERATIONS =    27, MAX =   26

 MODE   1, VV =   .2696413E+03, H =    .6704500E+01, DENS =   .6101046E+00, QBAR =   .2217926E+05

 ROOT DETERMINED BY MATRIX ITERATION IS S=(  -.6217274E+02,   .2445987E+03)

                              ALTITUDE           ZETA            OMEGA-N                   ROOT                              PRESSURE
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.650E-01
                           .6704500E+01      .2425444E+00      .4027351E+02     (  -.6137487E+02,   .2454900E+03)      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.656E-01
                           .6704500E+01      .2434032E+00      .4024894E+02     (  -.6155461E+02,   .2452859E+03)      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.662E-01
                           .6704500E+01      .2443069E+00      .4022189E+02     (  -.6174163E+02,   .2450636E+03)      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.669E-01
                           .6704500E+01      .2452606E+00      .4019153E+02     (  -.6193585E+02,   .2448178E+03)      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.675E-01
                           .6704500E+01      .2462687E+00      .4015587E+02     (  -.6213526E+02,   .2445361E+03)      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  3.000E-01
        5          8       .6704500E+01      .2158801E+00      .3841113E+02     (  -.5210142E+02,   .2356533E+03)      .2217926E+05


        4         11       .6704500E+01      .2707937E+00      .4203308E+02     (  -.7151705E+02,   .2542341E+03)      .2217926E+05
  CURRENT PLOT TITLE IS:
 TRACE 8 11 .2 TO .3 BY .0025 P-K

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
 ?

 DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
 ? 0

 CONTINUE
 ? 1
   TYPE IN TITLE
 ? TRACE 8 11 .3 TO 2.5 BY .025 P-K
   TYPE IN /INPUT/
 ?  $INPUT DELGAIN(3,1)=.025,NV=88,IPRINT=88$
```

TABLE GVII.- Continued

```
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  14.43.36.
 TRACE 8 11 .3 TO 2.5 BY .025 PK
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105452           P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 106226
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL     S1                                       S2
 MODE
   8   (     -.41570000E+02,      .25520000E+03)   (     -.52101419E+02,      .23565332E+03)
  11   (     -.51000000E+02,      .19600000E+03)   (     -.71517054E+02,      .25423412E+03)
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  3.000E-01
     #-ITER     ROOT NBR      ALTITUDE             ZETA             OMEGA-N                     ROOT                              PRESSURE
        1          8       .6704500E+01       .2158801E+00       .3841113E+02     (  -.5210142E+02,    .2356533E+03)        .2217926E+05
        1         11       .6704500E+01       .2707932E+00       .4203309E+02     (  -.7151695E+02,    .2542342E+03)        .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.500E+00
        3          8       .6704500E+01       .7434102E-01       .3672153E+02     (  -.1715257E+02,    .2300897E+03)        .2217926E+05
        4         11       .6704500E+01       .1409972E+00       .6057015E+02     (  -.5365978E+02,    .3767716E+03)        .2217926E+05
  CURRENT PLOT TITLE IS:
TRACE 8 11 .3 TO 2.5 BY .025 PK

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? TRACE 12 .07 TO .079 BY .001 PK
  TYPE IN /INPUT/
?  $INPUT GN(3,1)=.07,DELGAIN(3,1)=.001,NV=9,ITRACE=25*0,
?  S2I=25*(0,0),S1I=25*(0,0),S2I(12)=(-170.05,0),S1I(12)=(-172,0),
?  ITRACE(12)=1,IFLUT=1,IPRINT=9$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  14.46.36.
 TRACE 12 .07 TO .079 BY .001 PK
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105333           P-K FLUTTER ANALYSIS
```

TABLE GVII.- Continued

```
 FIELD LENGTH INCREASED TO 105365
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF   .6101046E+00


          AN INITIAL VELOCITY OF   .2696413E+03
          AN INITIAL ALTITUDE OF   .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF   .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL     S1                                           S2
 MODE
  12   (     -.17200000E+03,     0.          )   (     -.17005000E+03,     0.            )
  GAIN FOR CONTROL     3  AND SENSOR    1  IS  7.000E-02
      #-ITER     ROOT NBR      ALTITUDE            ZETA            OMEGA-N                        ROOT                               PRESSURE
        3          12       .6704500E+01      .1000000E+01     .2706461E+02    (  -.1700520E+03,  0.              )      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR    1  IS  7.900E-02
        7          12       .6704500E+01      .1000000E+01     .2359645E+02    (  -.1482609E+03,  0.              )      .2217926E+05
  CURRENT PLOT TITLE IS:
TRACE 12 .07 TO .079 BY .001 PK

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? TRACE 12 .08 TO .2 BY .005
  TYPE IN /INPUT/
?  $INPUT GN(3,1)=.08,DELGAIN(3,1)=.005,NV=44,
?  S2I(12)=(-142.6,7.26),S1I(12)=(-142.9,4),IPRINT=2$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STARCAR<-- DATE:  82/09/01. TIME:  14.49.24.
 TRACE 12 .08 TO .2 BY .005
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105376          P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 105536
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF   .6101046E+00
          AN INITIAL VELOCITY OF   .2696413E+03
          AN INITIAL ALTITUDE OF   .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF   .2217926E+05
```

TABLE GVII.- Continued

```
 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL      S1                                           S2
 MODE
  12   (     -.14290000E+03,      .40000000E+01)    (      -.14260000E+03,      .72600000E+01)
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  8.000E-02
      #-ITER      ROOT NBR      ALTITUDE            ZETA             OMEGA-N                        ROOT                          PRESSURE
         6          12        .6704500E+01      .9989417E+00      .2273704E+02      (  -.1427098E+03,   .6570808E+01)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  9.000E-02
         8          12        .6704500E+01      .9845866E+00      .2236558E+02      (  -.1383611E+03,   .2457789E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.000E-01
         7          12        .6704500E+01      .9707994E+00      .2201844E+02      (  -.1343062E+03,   .3318817E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.100E-01
         6          12        .6704500E+01      .9574833E+00      .2167886E+02      (  -.1304210E+03,   .3929571E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.200E-01
         6          12        .6704500E+01      .9446496E+00      .2134528E+02      (  -.1266930E+03,   .4400101E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.300E-01
         6          12        .6704500E+01      .9323209E+00      .2101800E+02      (  -.1231223E+03,   .4775711E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.400E-01
         6          12        .6704500E+01      .9205177E+00      .2069785E+02      (  -.1197119E+03,   .5081004E+02)       .2217926E+05


  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.500E-01
         5          12        .6704500E+01      .9092538E+00      .2038581E+02      (  -.1164644E+03,   .5331557E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.600E-01
         5          12        .6704500E+01      .8985342E+00      .2008282E+02      (  -.1133807E+03,   .5538263E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.700E-01
         5          12        .6704500E+01      .8883551E+00      .1978965E+02      (  -.1104599E+03,   .5709253E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.800E-01
         5          12        .6704500E+01      .8787042E+00      .1950690E+02      (  -.1076988E+03,   .5850866E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.900E-01
         5          12        .6704500E+01      .8695623E+00      .1923492E+02      (  -.1050923E+03,   .5968177E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.000E-01
         5          12        .6704500E+01      .8609050E+00      .1897386E+02      (  -.1026339E+03,   .6065314E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.100E-01
         4          12        .6704500E+01      .8527046E+00      .1872368E+02      (  -.1003159E+03,   .6145662E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.200E-01
         4          12        .6704500E+01      .8449316E+00      .1848420E+02      (  -.9813006E+02,   .6212012E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.300E-01
         4          12        .6704500E+01      .8375559E+00      .1825511E+02      (  -.9606785E+02,   .6266665E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.400E-01
         4          12        .6704500E+01      .8305480E+00      .1803601E+02      (  -.9412069E+02,   .6311525E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.500E-01
         4          12        .6704500E+01      .8238794E+00      .1782647E+02      (  -.9228027E+02,   .6348166E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.600E-01
         4          12        .6704500E+01      .8175235E+00      .1762600E+02      (  -.9053864E+02,   .6377893E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.700E-01
         4          12        .6704500E+01      .8114553E+00      .1743413E+02      (  -.8888831E+02,   .6401785E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.800E-01
         4          12        .6704500E+01      .8056517E+00      .1725035E+02      (  -.8732229E+02,   .6420741E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.900E-01
         4          12        .6704500E+01      .8000918E+00      .1707420E+02      (  -.8583415E+02,   .6435508E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  3.000E-01
         4          12        .6704500E+01      .7947565E+00      .1690522E+02      (  -.8441793E+02,   .6446710E+02)       .2217926E+05
```

TABLE GVII.- Continued

```
  CURRENT PLOT TITLE IS:
TRACE 12 .08 TO .2 BY .005

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? TRACE 12 .2 TO 2.5 BY .025
  TYPE IN /INPUT/
?  $INPUT DELGAIN(3,1)=.025,NV=92,IPRINT=8$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  14.53.18.
 TRACE 12 .2 TO 2.5 BY .025
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105456          P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 105756
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF   .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.


0        INITIAL      S1                                   S2
 MODE
  12   (     -.14290000E+03,      .40000000E+01)   (      -.84417928E+02,      .64467097E+02)
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  3.000E-01
      #-ITER     ROOT NBR      ALTITUDE            ZETA            OMEGA-N                       ROOT                                PRESSURE
         1           12      .6704500E+01      .7947565E+00      .1690522E+02    (  -.8441793E+02,   .6446710E+02)    .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  5.000E-01
         5           12      .6704500E+01      .7171705E+00      .1452386E+02    (  -.6544617E+02,   .6359616E+02)    .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  7.000E-01
         5           12      .6704500E+01      .6666014E+00      .1311098E+02    (  -.5491376E+02,   .6140627E+02)    .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  9.000E-01
         4           12      .6704500E+01      .6281463E+00      .1211708E+02    (  -.4782318E+02,   .5923938E+02)    .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  1.100E+00
         4           12      .6704500E+01      .5969556E+00      .1135483E+02    (  -.4258950E+02,   .5723786E+02)    .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  1.300E+00
         4           12      .6704500E+01      .5707448E+00      .1074003E+02    (  -.3851477E+02,   .5541099E+02)    .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  1.500E+00
         4           12      .6704500E+01      .5481987E+00      .1022757E+02    (  -.3522818E+02,   .537451E+02)     .2217926E+05
  GAIN FOR CONTROL      3  AND SENSOR     1  IS  1.700E+00
         4           12      .6704500E+01      .5284690E+00      .9790322E+01    (  -.3250846E+02,   .5222281E+02)    .2217926E+05
```

TABLE GVII.- Continued

```
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  1.900E+00
        4          12        .6704500E+01      .5109694E+00       .9410643E+01      (  -.3021301E+02,    .5082707E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.100E+00
        4          12        .6704500E+01      .4952762E+00       .9076362E+01      (  -.2824484E+02,    .4954266E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.300E+00
        4          12        .6704500E+01      .4810733E+00       .8778729E+01      (  -.2653523E+02,    .4835627E+02)       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.500E+00
        4          12        .6704500E+01      .4681189E+00       .8511239E+01      (  -.2503392E+02,    .4725639E+02)       .2217926E+05
  CURRENT PLOT TITLE IS:
TRACE 12 .2 TO 2.5 BY .025

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? TRACE SCHED .079 TO .009 BY -.01 P-K
  TYPE IN /INPUT/
?  $INPUT GN(3,1)=.079,DELGAIN(3,1)=-.01,NV=7,
?  IPRINT=1,ITRACE=25*0,ITRACE(10)=2,S2I=25*(0,0),
?  S1I=25*(0,0),S2I(10)=(-143.1,0),S1I(10)=(-130,0)$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/01. TIME:  14.56.46.
 TRACE SCHED .079 TO .009 BY -.01 P-K
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 105331           P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 105357
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN      SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL      S1                                          S2
 MODE


   10    (     -.13000000E+03,     0.              )    (     -.14310000E+03,     0.              )
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  7.900E-02
      #-ITER     ROOT NBR       ALTITUDE            ZETA            OMEGA-N                       ROOT                         PRESSURE
        6          10        .6704500E+01      .1000000E+01      .2222950E+02     '(  -.1396721E+03,  0.             )       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  6.900E-02
        6          10        .6704500E+01      .1000000E+01      .1959648E+02      (  -.1231283E+03,  0.             )       .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  5.900E-02
        5          10        .6704500E+01      .1000000E+01      .1851127E+02      (  -.1163098E+03,  0.             )       .2217926E+05
```

TABLE GVII.- Continued

```
  GAIN FOR CONTROL     3   AND SENSOR     1   IS  4.900E-02
       5         10      .6704500E+01       .1000000E+01      .1775263E+02     (  -.1115431E+03,  0.            )      .2217926E+05
  GAIN FOR CONTROL     3   AND SENSOR     1   IS  3.900E-02
       4         10      .6704500E+01       .1000000E+01      .1715910E+02     (  -.1078138E+03,  0.            )      .2217926E+05
  GAIN FOR CONTROL     3   AND SENSOR     1   IS  2.900E-02
       4         10      .6704500E+01       .1000000E+01      .1666831E+02     (  -.1047301E+03,  0.            )      .2217926E+05
  GAIN FOR CONTROL     3   AND SENSOR     1   IS  1.900E-02
       4         10      .6704500E+01       .1000000E+01      .1624862E+02     (  -.1020931E+03,  0.            )      .2217926E+05
  GAIN FOR CONTROL     3   AND SENSOR     1   IS  9.000E-03
       4         10      .6704500E+01       .1000000E+01      .1588144E+02     (  -.9978601E+02,  0.            )      .2217926E+05
  CURRENT PLOT TITLE IS:
 TRACE SCHED .079 TO .009 BY -.01 F-K

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 0
   109.689 CP SECONDS EXECUTION TIME.
/GET,TAPE1=T1NEXT
/GET,TAPE2=T2FIG15
/COPY,TAPE2
-1
TRACE 11 14 0 TO 2.5 BY .025 PP
 $INPUT RHO0=.6101,V0=269.64,H0=6.7045,IDMULT=1,ISPLANE=1,IFLUT=1,
 ITRACE=25*0,ITRACE(8)=0,0,0,2,0,0,2,
 IGAIN=1,DELGAIN(3,1)=.025,GN(3,1)=0,NV=100,IOPT2=0,
 S2I=25*(0,0),S1I=25*(0,0),
 S2I(11)=(-50,196),S2I(14)=(-200,480),
 S1I(11)=(-49,195),S1I(14)=(-200,482),
 IPKPLT=+1,IPRINT=100$
0 0 0
+1
 EOI ENCOUNTERED.
/REWIND,*
  11 FILE(S) PROCESSED.
/GET,STABCAR
/STABCAR
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/02. TIME:  08.41.17.
 TRACE 11 14 0 TO 2.5 BY .025 PP
 MACH #=    .8600

 FIELD LENGTH-OVERLAY(4,0) IS 103021          P-K FLUTTER ANALYSIS


 FIELD LENGTH INCREASED TO 103655
```

TABLE GVII.- Continued

```
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .6101046E+00
          AN INITIAL VELOCITY OF    .2696413E+03
          AN INITIAL ALTITUDE OF    .6704500E+01, AND
          AN INITIAL DYNAMIC PRESSURE OF    .2217926E+05

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS FOR GAINS=0.
0        INITIAL      S1                                        S2
 MODE
   11   (     -.49000000E+02,      .19500000E+03)   (     -.50000000E+02,      .19600000E+03)
   14   (     -.20000000E+03,      .48200000E+03)   (     -.20000000E+03,      .48000000E+03)
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  0.
     #-ITER     ROOT NBR       ALTITUDE            ZETA             OMEGA-N                 ROOT                             PRESSURE
        6          11       .6704500E+01      .2544537E+00      .3127385E+02     (  -.5000000E+02,    .1900316E+03)      .2217926E+05
        1          14       .6704500E+01      .3846154E+00      .8276057E+02     (  -.2000000E+03,    .4800000E+03)      .2217926E+05
  GAIN FOR CONTROL     3  AND SENSOR     1  IS  2.500E+00
        1          11       .6704500E+01      .1516634E+00      .6094093E+02     (  -.5807239E+02,    .3784738E+03)      .2217926E+05
        1          14       .6704500E+01      .1791137E+00      .9010278E+02     (  -.1014021E+03,    .5569772E+03)      .2217926E+05
  TYPE IN TITLE
? FIGURE 15. CONTROL ROOT LOCI AT VF (M=.86,H=6.7045KM)
  TYPE IN /INPUT/
?  $INPUT IFLUT=0,IPKPLT=1$
1      DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  82/09/02. TIME:  08.43.24.
 FIGURE 15. CONTROL ROOT LOCI AT VF (M=.86,H=6.7045KM)
 MACH #=    .8600

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 1
 PLOT SET  TYPE OF VARIATION       # MODES                   PLOT TITLE


    1       GAIN     3, 1          6     TRACE 8 + CNTR - SCH 0 TO .07 BY 01 PK
    2       GAIN     3, 1          3     TRACE EXCEPT 8 11 12 SCH 0 TO 2.5 BY .025 PK



    3       GAIN     3, 1          2     TRACE 8 11 0 TO .2 BY .01 PK
    4       GAIN     3, 1          2     TRACE 8 11 .2 TO .3 BY .0025 PK
    5       GAIN     3, 1          2     TRACE 8 11 .3 TO 2.5 BY .025 PK
    6       GAIN     3, 1          1     TRACE 12 .07 TO .079 BY .001 PK
    7       GAIN     3, 1          1     TRACE 12 .08 TO .2 BY .005
    8       GAIN     3, 1          1     TRACE 12 .2 TO 2.5 BY .025
    9       GAIN     3, 1          1     TRACE SCHED .079 TO .009 BY -.01 P-K
   10       GAIN     3, 1          2     TRACE 11 14 0 TO 2.5 BY .025 PP
 ' TYPE IN NUMBER OF PLOT SETS TO BE PLOTTED,'
 ' CORRESPONDING PLOT SET #'S, AND WHETHER TO COMBINE THEM'
? 10 1 2 3 4 5 6 7 8 9 10 1
 CURRENT PLOT TITLE IS:
 FIGURE 15. CONTROL ROOT LOCI AT VF (M=.86,H=6.7045KM)
 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
?
 TYPE IN /PLOTP/ -KVAR,IDASH,MODEPK, ETC. FOR PLOT SET #  1
?  $PLOTP ISAME=9,XRLORG=-700,XRLEND=0,XRLDEL=100,NDXRL=5,
?  YRLORG=-100,YRLEND=600,YRLDEL=100,NDYRL=5,IDASH=0$
```

TABLE GVII.- Concluded

```
FIELD LENGTH OVERLAY(5,0) IS 077351          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 100776          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 077404          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 077620          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 100340          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 077243          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 077522          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 100102          PK FLUTTER PLOTS

FIELD LENGTH OVERLAY(5,0) IS 077231          PK FLUTTER PLOTS
 TYPE IN /PLOTP/ -KVAR,IDASH,MODEPK, ETC. FOR PLOT SET # 10
?  $PLOTPK ISAME=1,IDASH=1$
?  $PLOTPKISAME-1,IDASH=1$

 FIELD LENGTH OVERLAY(5,0) IS 100464          PK FLUTTER PLOTS
 ' DO YOU WISH TO CONTINUE PLOTTING? (1/0)'
? 0

 CONTINUE
? 0
    56.048 CP SECONDS EXECUTION TIME.
/
```

TABLE GVIII.- INTERACTIVE EXECUTION OF SAMPLE CASE 9

```
? 1
  TYPE IN TITLE
? CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (P-K, FIGURE 16)
  TYPE IN /INPUT/
?  $INPUT ISPLANE=0,IFLUT=1,IPKPLT=-1,KVAR=2,H0=16,DV=1,NV=10,IPRINT=10,
?  ITRACE=0,1,6*0,IOPT2=1$
1       DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  83/09/12.
 TIME:  11.33.04.
 CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (P-K, FIGURE 16)
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 104264          P-K FLUTTER ANALYSIS
  INITIAL ROOTS DETERMINED BY MATRIX ITERATION


 FIELD LENGTH INCREASED TO 104320
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .1664707E+00
          AN INITIAL VELOCITY OF    .2537597E+03
          AN INITIAL ALTITUDE OF    .1600000E+02, AND
          AN INITIAL DYNAMIC PRESSURE OF    .5359857E+04

 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN     SCHEDULING DENOMINATORS
 FOR GAINS=0.
0        INITIAL       S1                                            S2
 MODE
    2   (     -.33806443E+00,      .29525551E+01)  (     -.34147922E+00,      .2982
3789E+01)
      #-ITER     ROOT NBR      ALTITUDE             ZETA               OMEGA-N
             ROOT                              PRESSURE
         2          2        .1600000E+02       .1137569E+00       .4777617E+00      (
  -.3414830E+00,     .2982379E+01)       .5359857E+04
         4          2        .6000000E+01       .2078029E+00       .9194466E+00      (
  -.1200489E+01,     .5650944E+01)       .2444549E+05
  CURRENT PLOT TITLE IS:
CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (P-K, FIGURE 16)

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
? SAME

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (P-P,FIGURE 16)
  TYPE IN /INPUT/
?  $INPUT ISPLANE=1,IPS=0,S1I=25*(0,0),S2I=25*(0,0),IOPT2=1$
1       DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  83/09/12.
 TIME:  11.44.53.
 CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (P-P,FIGURE 16)
 MACH #=    .8600


 FIELD LENGTH-OVERLAY(4,0) IS 101750          P-K FLUTTER ANALYSIS
  INITIAL ROOTS DETERMINED BY MATRIX ITERATION
                              UNSTABLE ROOT


 FIELD LENGTH INCREASED TO 102004
1THIS CASE IS FOR
          AN INITIAL DENSITY  OF    .1664707E+00
          AN INITIAL VELOCITY OF    .2537597E+03
          AN INITIAL ALTITUDE OF    .1600000E+02, AND
          AN INITIAL DYNAMIC PRESSURE OF    .5359857E+04
```

TABLE GVIII.- Concluded

```
 DO NOT TRY TO DETERMINE ROOTS CORRESPONDING TO GAIN    SCHEDULING DENOMINATORS
 FOR GAINS=0
0         INITIAL      S1                                              S2
 MODE
    2   (     -.41972234E+00,       .27741750E+01)   (     -.42396196E+00,       .2802
1970E+01)
      #-ITER    ROOT NBR       ALTITUDE              ZETA               OMEGA-N
              ROOT                              PRESSURE
         1          2         .1600000E+02        .1495938E+00        .4510590E+00      (
  -.4239619E+00,      2802197E+01)       .5359857E+04
         3          2         .6000000E+01        .2985435E+00        .8603155E+00      (
  -.1613783E+01,     .5159008E+01)       .2444549E+05
  CURRENT PLOT TITLE IS:
CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (P-P,FIGURE 16)

 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
? SAME

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 0

 CONTINUE
? 1
  TYPE IN TITLE
? CASE9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (FIGURE 16)
  TYPE IN /INPUT/
?  $INPUT IFLUT=0,IPKPLT=1$
1       DETERMINATION OF STABILITY CHARACTERISTICS -->STABCAR<-- DATE:  83/09/12.
 TIME.  11 50 35
 CASE9   LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (FIGURE 16)
 MACH #=    .8600

DO YOU WISH TO GENERATE PLOTS OF STABILITY CHARACTERISTICS AT THIS TIME?
? 1
 PLOT SET  TYPE OF VARIATION       # MODES                 PLOT TITLE


   1        ALTITUDE               8      OPEN LOOP STABILITY CHARACTERISTICS VER
SUS ALTITUDE,DV=0.5 (FIG 9)
   2        ALTITUDE               1      CASE 9   LOCUS OF SHORT PERIOD ROOT WITH
 ALTITUDE (P-K, FIGURE 16)
   3        ALTITUDE               1      CASE 9.  LOCUS OF SHORT PERIOD ROOT WITH
 ALTITUDE (P-P,FIGURE 16)
 " TYPE IN NUMBER OF PLOT SETS TO BE PLOTTED,"
 " CORRESPONDING PLOT SET #'S, AND WHETHER TO COMBINE THEM"
? 2
2 3 1
 CURRENT PLOT TITLE IS:
 CASE9.  LOCUS OF SHORT PERIOD ROOT WITH ALTITUDE (FIGURE 16)
 HIT RETURN OR TYPE: SAME, IF OK; OTHERWISE TYPE IN NEW TITLE.
? SAME
 TYPE IN /PLOTP/ -KVAR,IDASH,MODEPK, ETC.  FOR PLOT SET #   2
?  $PLOTP IDASH=0,XRLORG=-2,XRLEND=0,XRLDEL=1,NDXRL=2,
?   YRLORG=0,YRLEND=6,YRLDEL=2,NDYRL=4$


 FIELD LENGTH OVERLAY(5,0) IS 077256          PK FLUTTER PLOTS
 TYPE IN /PLOTP/ -KVAR,IDASH,MODEPK, ETC.  FOR PLOT SET #   3
?  $PLOTP IDASH=1$


 FIELD LENGTH OVERLAY(5,0) IS 077256          PK FLUTTER PLOTS
 " DO YOU WISH TO CONTINUE PLOTTING? (1/0)"
? 0

 CONTINUE
? 0
```

REFERENCES

1. Severt, Francis D.: Development of Active Flutter Suppression Wind Tunnel Testing Technology. AFFDL-TR-74-126, U.S. Air Force, Jan. 1975.

2. Roger, Kenneth L.: Airplane Math Modeling Methods for Active Control Design. Structural Aspects of Active Controls, AGARD-CP-228, Aug. 1977, pp. 4-1 - 4-11.

3. Hassig, Hermann J.: An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration. J. Aircr., vol. 8, no. 11, Nov. 1971, pp. 885-889.

4. Woodcock, D. L.; and Lawrence, A. Jocelyn: Further Comparisons of Different Methods of Assessing the Free Oscillatory Characteristics of Aeroelastic Systems. Tech. Rep. 72188, British R.A.E., Dec. 1972.

5. Rodden William P.; Harder, Robert L.; and Bellinger, E. Dean: Aeroelastic Addition to NASTRAN. NASA CR-3094, 1979.

6. Field, E. I.; Herting, D. N.; and Morgan, M. J.: NASTRAN® User's Guide (Level 17.5). NASA CR-3146, 1979.

7. Peele, Ellwood L.; and Adams, William M., Jr.: A Digital Program for Calculating the Interaction Between Flexible Structures, Unsteady Aerodynamics and Active Controls. NASA TM-80040, 1979.

8. Bisplinghoff, Raymond L.; Ashley, Holt; and Halfman, Robert L.: Aeroelasticity. Addison-Wesley Pub. Co., Inc., c.1955.

9. Giesing, J. P.; Kalman, T. P.; and Rodden, W. P.: Subsonic Unsteady Aerodynamics for General Configurations. Part I, Vol I - Direct Application of the Nonplanar Doublet-Lattice Method. AFFDL-1R-71-5, Pt. I, Vol I, U.S. Air Force, Nov. 1971. (Available from DTIC as AD 891 403L.)

10. Giesing, J. P.; Kalman, T. P.; and Rodden, W. P.: Subsonic Unsteady Aerodynamics for General Configurations. Part I, Vol II - Computer Program H7WC. AFFDL-TR-71-5, Pt. I, Vol II, U.S. Air Force, Nov. 1971. (Available from DTIC as AD 892 535L.)

11. Cunningham, Atlee M., Jr.: A Steady and Oscillatory Kernel Function Method for Interfering Surfaces in Subsonic, Transonic and Supersonic Flow. NASA CR-144895, 1976.

12. Rowe, W. S.; Redman, M. C.; Ehlers, F. E.; and Sebastian, J. D.: Prediction of Unsteady Aerodynamic Loadings Caused by Leading Edge and Trailing Edge Control Surface Motions in Subsonic Compressible Flow - Analysis and Results. NASA CR-2543, 1975.

13. Vepa, Ranjan: Finite State Modeling of Aeroelastic Systems. NASA CR-2779, 1977.

14. Edwards, John William: Unsteady Aerodynamic Modeling and Active Aeroelastic Control. SUDAAR 504 (NASA Grant NGL-05-020-007), Stanford Univ., Feb. 1977. (Available as NASA CR-148019.)

15. Dunn, H. J.: An Analytical Technique for Approximating Unsteady Aerodynamics in the Time Domain. NASA TP-1738, 1980.

16. Morino, Luigi: A General Theory of Unsteady Compressible Potential Aerodynamics. NASA CR-2464, 1974.

17. Morino, Luigi: Steady, Oscillatory, and Unsteady Subsonic and Supersonic Aerodynamics - Production Version (SOUSSA-P 1.1), Volume I - Theoretical Manual. NASA CR-159130, 1980.

18. Harder, Robert L.; and Desmarais, Robert N.: Interpolation Using Surface Splines. J. Aircr., vol. 9, no. 2, Feb. 1972, pp. 189-191.

19. Anon.: U.S. Standard Atmosphere, 1962. NASA, U.S. Air Force, and U.S. Weather Bur., Dec. 1962.

20. Tiffany, Sherwood H.; and Newsom, Jerry R.: Some Programming Techniques for Increasing Program Versatility and Efficiency on CDC Equipment. NASA CR-3033, 1978.

21. Boeing Wichita Co.: Integrated Design of a High Aspect Ratio Research Wing With an Active Control System for Flight Tests on a BQM-34E/F Drone Vehicle, Volume I. Doc. No. D3-11535-1 Volume I (Contract NAS1-14665), Boeing Co., 1979. (Available as NASA CR-166108.)

22. Boeing Wichita Co.: Integrated Design of a High Aspect Ratio Research Wing With an Active Control System for Flight Tests on a BQM-34E/F Drone Vehicle, Volume I Appendices. Doc. No. D3-11535-1 Vol. I Appen. (Contract NAS1-14665), Boeing Co., 1979. (Available as NASA CR-166109.)

23. Boeing Commercial Airplane Co.: Integrated Application of Active Controls (IAAC) Technology to an Advanced Subsonic Transport Project - Wing Planform Study and Final Configuration Selection. NASA CR-165630, 1981.

TABLE I.- OUTLINE OF DATA ON TAPE2 AND/OR TAPE3 (INPUT)

Set	When required	Variable or namelist name	Data type
1	(a)	ICASE	Free-field integer
2	(a)	HEADER	Eight-word (A10) alphanumeric array
3	(a)	$INPUT...$	Namelist
4	ISENSE = 1	$SENLOC...$	Namelist
5	ICONSYS = 1 & ICASE = ±2 or ICHSEN = 1 & ICASE = ±1	$SENINP...$	Namelist
6	ICONSYS = 1 & ICASE = ±2 or ICHFIL = 1 & ICASE = ±1	I & J	Free-field integers
7	ICONSYS = 1 & ICASE = ±2 or ICHFIL = 1 & ICASE = ±1	$FILTIN...$	Namelist
Repeat sets 6 and 7 for each control-sensor pair; terminate with 0,0			
8	ICONSYS = 1 & ICASE = ±2 or ICHACT = 1 & ICASE = ±1	$ACTINP...$	Namelist
9	ISELECT ≠ 0	$SELECT...$	Namelist
10	IDELPLT ≠ 0	IPLT2	Free-field integer array
11	IPKPLT > 0	ICOUNT	Free-field integer
12	IPKPLT > 0	IPLT	Free-field integer array (Size = ICOUNT)
13	IPKPLT > 0	ICOMBIN	Free-field integer
14	IPKPLT > 0 & ICOMBIN = 1	PTITLE	Eight-word (A10) alphanumeric array
15	IPKPLT < 0	$PLOTP...$	Namelist
Repeat set 15 up to ICOUNT times (see definition of ISAME)			
16	IPKPLT > 0	IOP	Free-field integer
Repeat sets 10 to 16 as desired, terminate with IOP = 0 Repeat sets 1 to 16 as desired for each case, end with ICASE = 0			

[a]Required in all cases.

TABLE II.- DESCRIPTION OF TAPE2 INPUT DATA

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
1		Always		ICASE	Integer indicating type of case to be run (initialization or continuation) and which version of program (interactive or noninteractive) to invoke.
				= ±2	Initialization case. All data are (re)initialized by namelist and option parameters from TAPE2 and all array data (such as aerodynamics, generalized masses, and mode shapes) from TAPE5 and TAPE10.
				= ±1	Continuation case. Case being run will use data previously stored on TAPE1, data storage file created and used by program during execution. Changes only to namelist and option parameters are read-in from keyboard if ICASE = +1 and from TAPE2 if -1. Changes to array data are made according to ISELECT.
				= 0	Execution stopped.
2		Always		HEADER	Alphanumeric description of case being run.
3	†INPUT	ICASE = ±2		NM	Total number of modes: rigid, elastic, and control.
			n_r	NR	Number of rigid-body modes (see section on "Sensor input definitions").
			n_δ	NC	Number of control modes.
			n_k	NK	Number of reduced frequencies at which aerodynamic data have been computed.
				NG	Number of gust forces included as columns in aerodynamic array input. Provision has been made to read-in aerodynamic gust forces which may be included as additional (NG) columns in aerodynamic data. These data are not used by the program. (Default = 0)
			n_s	NS	Number of sensors. (Default = 0)
				IREW5	
				= 1	Rewind TAPE5 input file at beginning of case. This allows for reinitialization of original array data. (Default)
				= 0	Do not rewind TAPE5. This allows new aerodynamic and structural data to be read-in.
				IREW2	
				= 1	Rewind TAPE2 input file immediately after reading namelist INPUT, allowing reinitialization of sensor and control system data.
				= 0	Do not rewind. (Default)
				ISELECT	Indicates whether model definition arrays are to be altered for current case. Different modal combinations can be selected, different subsets of aerodynamic data in terms of frequency can be selected, etc. Refer to definition of namelist SELECT.
				= 1	Model definition arrays are read from TAPE5 and data are selected from this input. Refer to definition of IREW5 if reinitializing.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT			= -1	Model definition arrays are read-in from TAPE1 (a data storage file created and used by program during execution). These data represent model as it was at end of preceding case, incorporating any modal and/or frequency selections up to that point. New modal and/or frequency selections are from this input.
				= 0	Model definition arrays will not be altered for this case. (Default) If ICASE = ±1, the arrays will be precisely those in effect at the end of the preceding case.
		ICASE = ±2		CSURFEF(I)	Multiplier which modifies column of aerodynamic force matrices corresponding to ith control mode. The multiplication is only performed when ICASE = ±2 or ISELECT > 0. (Default = 1.0)
			2b or c	C	Reference chord length
			N_{Ma}	XMACH	Mach number at which aerodynamic forces were generated.
				KFIT	Type of fit to be employed when interpolating input aerodynamic force data to obtain aerodynamic forces at a particular frequency. Used when ISPLANE = 0. When KFIT < 0, the interpolation is for Re(Q(k)) and Q'(k) (see section "Unsteady aerodynamic forces").
				= ±1	Linear fit.
				= ±2	Quadratic fit. (Default)
				= ±3	Cubic fit.
				ISTIFF	Integer indicating how stiffness matrix is obtained.
				= 0	Compute diagonal stiffness matrix: $K_{ii} = M_{ii}\left(2\pi f_{N_i}\right)^2$. (Default)
				= 1	Read-in full stiffness matrix.
				ISPLANE	Integer indicating whether to employ p-plane approximations to aerodynamics. Functions used for this fit have the form: $\hat{Q}_j(p) = A_{0_j} + A_{1_j} p + A_{2_j} p^2 + \sum_{\ell=1}^{n_{\ell j}} \frac{A_{(\ell+2)_j} p}{p + b_{\ell j}}$
				= 0	Do not compute or use p-plane fit. (Default)
				= 1	Employ p-plane curve fit, using coefficients which have been previously computed and stored on TAPE1.
				= 2	Compute and employ p-plane approximations for full set of aerodynamics from TAPE5, rewinding TAPE5 automatically.
				= 3	Compute and employ p-plane approximations for selected set of aerodynamics from TAPE1.
				INTERP	
				= 1	Compute and store geometric coefficients, used when computing sensor deflections for plate-type sections.
				= 0	Do not compute.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT			ISENSE	
				= 1	Compute and store sensor deflections on TAPE1.
				= 0	Do not compute.
				ICONSYS	Integer indicating whether control system is to be included in analyses.
				= 1	Set up control system coefficients (if ICASE = 2 or ISELECT > 0) and perform analyses using a control system.
				= 0	Do not include control system. No quantities pertaining to control system need be input. (Default)
				IFLUT	
				= 1	Perform characteristic root determination.
				= 0	Do not determine roots for current case. (Default)
				IAEROWR	Integer indicating whether to write out aerodynamic forces to file TAPE6.
				= 1	Write out original aerodynamic forces if ICASE = ±2 and/or selected aerodynamic forces if ISELECT ≠ 0.
				= 0	Do not write out aerodynamic forces. (Default)
				IMODEWR	This parameter is employed primarily in noninitialization case to output array data from previous case.
				= 1	Write out generalized masses, structural dampings, and natural frequencies (generalized stiffness if ISTIFF = 1) to file TAPE6.
				= 0	Do not write these arrays on TAPE6 unless ISELECT ≠ 0 or ICASE = ±2. (Default)
				IAPLT	Integer indicating whether to plot aerodynamic forces and corresponding curve fits. Uses selected model aerodynamics (those on TAPE1).
				= -1	Plot only tabular points and interpolated curve (refer to KFIT).
				= 1	Plot tabular points, interpolated curves (refer to KFIT), and p-plane curves (if ISPLANE ≠ 0).
				0	Do not plot aerodynamic data. (Default)
				IPKPLT	Integer indicating whether to generate plots of characteristic root data, which type of plots to generate, and whether to allow modification if plot parameters during generation of plots on a plot-per-plot basis. If IPKPLT > 0, modification will be allowed. If IPKPLT < 0, only the last set of data will be plotted and no changes to plot parameters input in namelist INPUT will be allowed during generation of plots.
				= ±1	Generate plots of loci versus velocity, altitude, or density, variation determined by KVAR = 1, 2, or 3, respectively, during root generation, or versus corresponding dynamic pressure if KVAR < 0 or versus gain if IGAIN 1 during root determination.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT			= ±2	Generate plots of ζ and ω_n versus velocity, altitude, or density depending upon KVAR during root determination (or corresponding dynamic pressure if KVAR < 0) or versus gain if IGAIN = 1 during root determination.
				= ±3	Generate both (ζ,ω_n) and root loci plots.
				= 0	Do not generate any characteristic root plots. (Default)
				IDELPLT	
				= 1	Delete unwanted characteristic root data used for plotting from data storage file TAPE1.
				= 0	Do not delete any data.
		ISPLANE ≠ 0	$3 + \max\{n_{\ell j}\}$	NCOEF	Number of coefficients desired currently for p-plane fit, including rational terms $p/(p + b_{\ell j})$. Maximum = 10.
				NPOLYC(J)	Number of polynomial coefficients in p-plane fit of jth column of aerodynamic forces. (Default = 3)
			$b_{\ell j}$	BN(L,J)	Constant denominator coefficient in ℓth $p/(p + b_{\ell j})$ corresponding to jth column mode. If same values of $b_{\ell j}$ are used for more than one mode, they still need only be input once. See definition of NCOL(J).
			nkk_j	NKK(J)	Number of reduced frequencies to be used for jth pressure mode fit. Program uses first nkk_j of total number n_k. If 0, the program uses all n_k frequencies for the jth mode. (Default = 0)
				NCOL(J)	Integer indicating whether to use a preceding set of b_ℓ for jth mode.
				= N	Set each $b_{\ell j} = b_{\ell N}$. (Default = 1)
				= 0	Input new $b_{\ell j}$ coefficients for jth mode.
				ICOF(N,J)	Integer indicating whether to include Nth constraint (as defined below) when fitting the Jth column of aerodynamic forces. (See appendix A.)
				= 0	Nth constraint not employed. (Default)
				> 0	Employ Nth constraint.
				ICOF(1,J) = 1	Force agreement with tabular value for $k_1 = 0$. $\hat{Q}_{ij}(0) = Q_{ij}(0)$.
				ICOF(2,J) = 1	For $k_1 = 0$, Force slope = $\left.\frac{\partial \hat{Q}_{ij}}{\partial k}\right\|_{k_1=0} = \frac{\sqrt{-1}\, \mathrm{Im}[Q_{ij}(\varepsilon)]}{\varepsilon}$, $\varepsilon = k_2$
				ICOF(3,J) = 1	For $k_1 = 0$, Force slope = $-\sqrt{-1}\, \mathrm{Re}[Q_{i2}(0)]/b\theta_n$.
				ICOF(4,J) = 1	For $k_1 = 0$, Force slope = 0.
				ICOF(5,J) = $\ell + 1$	Force $A_{\ell j} = 0$.
				ICOF(6,J) = 1	Force $\hat{Q}_j(k_0) = Q_j(k_0)$ for some k_0, where $Q_j(k_0)$ is interpolated value of jth aerodynamic force at k_0, using a piecewise quadratic fit to $Q_j(k_i)$ $(i = 1, \ldots, n_k)$.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT	ISPLANE ≠ 0		IERPRT	
				= 1	Write percentage errors between p-plane fit and actual data.
				= 0	Do not write out errors.
		ISPLANE ≠ 0 & ICOF(2,J) = 1	θ_n	THETAN	Normalizing factor employed when applying constraint number 3 in p-plane fit. (Refer to appendix A.) (Default = 1)
		ISPLANE ≠ 0 & ICOF(6,J) = 1	k	SPKO	Value of reduced frequency for which p-plane approximation must precisely fit data. (See appendix A, constraint number 6.)
		INTERP = 1 or ISENSE = 1		ISYM	
				= 1	Modes symmetric with respect to x-axis.
				≠ 1	Modes either antisymmetric or nonsymmetric with respect to x-axis.
				NPMX	Maximum number of nodes in any one structural section.
				NSECTNS	Number of sections into which structure is divided.
				NNODES	Total number of nodes in all structural sections at which modal data are available.
				ISS(1,I)	Beginning node number of structural section I
				ISS(2,I)	Ending node number of structural section I
				IPLATE(I)	
				= 1	Indicates that structural section I is modeled as plate.
				= 0	Indicates that structural section I is modeled as beam.
				NDOF	Number of degrees of freedom in TAB array. If all sections are plate type, this can be set to 3 and TAB(I,4) array can be omitted. (Default = 4)
			x_{cg}	XCG	x-coordinate of aircraft center of mass.
			x_0	XO(I)	x-coordinate of translated axis of section I when IPLATE(I) = 0; x-coordinate of mean of root chord of structural section I when IPLATE(I) = 1.
			y_0	YO(I)	y-coordinate of translated axis of section I when IPLATE = 0; y-coordinate at root of section I for IPLATE(I) = 1.
			Λ	RO(I)	Sweep angle of elastic axis of section I if IPLATE(I) = 0; if IPLATE(I) = 1, RO(I) = $1/b_R$ where b_R is root semichord for section I.
		ICONSYS = 1		ICSREAD	Integer indicating from where to get sensor deflections needed in conjunction with control system.
				1	Sensor deflections input by user on TAPE5.
				0	Sensor deflections are read-in from data storage file TAPE1 (refer to definition of ISENSE). (Default)

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT	ICONSYS = 1		ICSACT	
				= 1	Actuator transfer function relates actual control surface position to commanded position. (Default)
				= 0	Control surfaces are treated as full degrees of freedom and actuator transfer functions can be considered to relate actuator-produced hinge moments to actuator inputs.
			G_{ij}	GN(I,J)	Gain for Ith control and Jth sensor pair. The user will find it convenient (virtually essential in plotting gain root loci) to nondimensionalize G_{ij} by including desired dimensional gain in $(T_L')_{ij}$. Then gain variations will be in terms of a multiple of a nominal or desired value. (Default = 0)
				PHASE	Phase error (degrees) in transfer function. (Default = 0)
				IPH & JPH	Control (IPH) and sensor (JPH) pair for which phase error is introduced.
				ISCHDUL(I,J)	
				= 0	No scheduling in control logic for control I, sensor J. (Default)
				= 1	Scheduling is included in control logic for control I, sensor J.
		ICONSYS = 1 & ICASE = ±1		ICHSEN	
				= 1	Indicates a change is to be made in a denominator coefficient for sensor dynamics. (See namelist SENINP.)
				= 0	Sensor dynamics are unchanged from preceding case. (Default)
				ICHFIL	
				≠ 0	Indicates a change is to be made in filter dynamics (see namelist FILTIN).
				= 0	No change is to be made in filter dynamics. (Default)
				= 1	Change only occurs in numerator coefficients.
				= 2	Change only occurs in denominator coefficients.
				= 3	Change occurs in both numerator and denominator.
				ICHACT	
				≠ 0	Indicates a change is to be made in actuator dynamics (see namelist ACTINP).
				= 0	No change is to be made in actuator dynamics.
				= 1	Change only occurs in numerator coefficients.
				= 2	Change only occurs in denominator coefficients.
				= 3	Change occurs in both numerator and denominator.
		IFLUT = 1		KVAR	Integer indicating type of variation to be performed in flutter analysis. If negative, any corresponding plot output will be with respect to dynamic pressure. Can be changed at plot time in interactive execution.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT	IFLUT = 1		KVAR	
				= ±1	Velocity is varied with respect to fixed density, using fixed Mach number aerodynamics. RHO0 and V0 are required inputs.
				= ±2	Altitude is varied, matching density and velocity at the input Mach number. (H0 and XMACH are necessary inputs, as are the unit conversion factors defined subsequently.
				= ±3	Density is varied with respect to fixed velocity and Mach number. (V0 and RHO0 are necessary inputs.) This option is usually run to correlate with model testing in a wind tunnel when only density is varied.
				IGAIN	
				= 1	Perform an automatic gain variation. (ICONSYS = 1) DV is set to 0. Velocity and density are fixed at V0 and RHO0 if KVAR = ±1 or ±3. Velocity and density are computed to correspond to altitude H0 if KVAR = ±2. Input also DELGAIN.
				= 0	Do not perform automatic gain variation. (Default)
				IDMULT	
				= 1	Multiply all matrix elements by control-system transfer-function common denominator (this must be done to avoid singularities when tracking control system roots beginning with zero gains)
				= 0	Do not clear control denominator. (Default)
				NV	Number of stepwise variations in quantity being varied - velocity, altitude, density, or gain. (Default = 1)
			ϵ	EPSI	Criterion for determining whether convergence upon sufficiently accurate estimate of characteristic root has been achieved. If s_{i_N} is Nth estimate of ith root and $s_{i_{N-1}}$ is $(N-1)^{th}$ estimate, then either $\lvert s_{i_N}\rvert \leq 0.1\epsilon$ or $\lvert s_{i_N} - s_{i_{N-1}}\rvert / \lvert s_{i_{N-1}}\rvert < \epsilon$ implies convergence. (Default = 10^{-10})
				NIT	Maximum number of iterations, without convergence, to determine roots using determinant iteration techniques before using matrix iteration techniques instead. (Default = 30)
				NCUT	Maximum number of times to halve step size when computed step size causes an increase in size of determinant. (Default = 5)
			$n_{\Delta V_c}$	NFINE	Number of subdivisions of DV for finer scan. This is used when convergence difficulties arise in obtaining a root using both determinant iteration and matrix iteration during normal step size scan. For change in stability, a finer scan is also performed. Number of subdivisions for this is min(max(NFINE,10),25). (Default = 25)
				IDAMP	Integer which indicates how damping is to be incorporated into equations of motion (eq. (1)).
				= 1	Damping is $\sqrt{-1}\, q_{s_i}$. (Default)
				= 2	Damping is $(s/\lvert s\rvert) q_{s_i}$ ($s \neq 0$) and $\sqrt{-1}\, q_{s_i}$ ($s = 0$).

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT	IFLUT = 1		IDAMP	
				= 3	Damping is $(s/\omega_{n_i})q_{s_i}$ (if $\omega_{n_i} \neq 0$) and 0 (if $\omega_{n_i} = 0$), corresponds to viscous damping 2ζ.
				ITRACE(I)	Integer indicating whether to trace Ith root, provided Ith initial guess in nonzero.
				= 1, 2, or 3	Trace root. Use linear (1), quadratic (2), or cubic (3) curve fit to extrapolate for estimate of Ith root at new value of independent variable. (Default = 1)
				= 0	Do not trace root.
				IPRINT	
				= N	Print characteristic root data to screen (file OUTPUT) every N times they are computed. (Default = 0)
		IFLUT = 1 & KVAR = ±1, ±3	U_o	VO	Initial velocity for velocity variation; constant velocity for density variation.
			ρ_o	RHOO	Initial density for density variation; constant density for velocity variation.
		IFLUT = 1 & KVAR = ±2	h_o	HO	Initial altitude for altitude variation.
				CONFAC1	Factor for converting input velocity to feet per second for use in subroutine AT62. (Default = 0.0833 to convert inches per second)
				CONFAC2	Factor for converting input altitude to feet for use in subroutine AT62. (Default = 1)
				CONFAC3	Factor for converting density, output by subroutine AT62 in slugs per foot^3, into type of units corresponding to input model. (Default = 0.48225E-04 for converting to inch units)
		IFLUT = 1 & IGAIN = 0		DV	Step size for each increment of independent variable: velocity, altitude, or density, corresponding to KVAR = ±1, ±2, or ±3, respectively.
		IFLUT = 1 & IGAIN = 1		DELGAIN(I,J)	Increment in GN(I,J) for proper root estimation. Each nonzero DELGAIN should be equal. (Default = 0)
		IFLUT = 1	$\hat{s}_{i_1}$ & $\hat{s}_{i_2}$	S1I(I) & S2I(I)	Two initial estimates (complex) for Ith characteristic root. S2I(I) should be best guess. In some cases, these can be determined automatically by program. See definition of IOPT1 and IOPT2.
				IOPT1	
				= 1	Compute initial estimates for elastic modes using natural frequencies and FS and GS (defined subsequently). (See eq. (17).)
				= 0	Do not compute initial estimates in this manner.
				IOPT2	
				= 1	Use matrix iteration to obtain initial estimate (appendix D, O_p = 2 and 3). This method attempts to converge on elastic and rigid-body characteristic roots (O_p = 2) unless some S2I(I) ≠ 0. (O_p = 3), in which case method will attempt to converge on nonzero roots nearest those input in array S2I(I). See also definition of OMR.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT	IFLUT = 1		IOPT2	
				= 0	Do not compute initial estimates in this manner.
		IFLUT = 1 & IOPT1 = 1	d_1 & d_2	FS & GS	Used in computing initial estimates for s_{i_1} and s_{i_2}, as in $\hat{s}_{i_2} = \left(-d_1\omega_{n_i}, d_2\omega_{n_i}\right)$ and $\hat{s}_{i_1} = \left(0, d_2\omega_{n_i}\right)$. (For FS, Default = 0.01; for GS, Default = 1.0)
		IOPT2 = 1 & all S2I(I) = (0,0)	ω_r	OMR	Initial frequency corresponding to imaginary part of characteristic root of smallest magnitude desired. (See O_p = 2 in appendix D.) (Default = 1.0)
		IFLUT = 1 & ICASE = ±1		IPS	Integer which indicates from where to obtain initial estimates, $\hat{s}_{i_1}$ and $\hat{s}_{i_2}$, for current case.
				= 0	(Default) S1I and S2I are final values of characteristic roots from most recent flutter case (IFLUT = 1) if run. If one has not been run, S1I and S2I will be same as those initially input or computed using IOPT1. In any case these can be overwritten by input.
				= 1	S1I and S2I are same as those at beginning of preceding case, prior to any overwrite using IOPT1 or IOPT2. These are not overwritten by input.
				-1	S1I and S2I are set to initial estimates in last or current initialization case (ICASE = ±2). If none were input at that time, S1I and S2I are set to 0. These can only be overwritten in current case by use of IOPT1 or IOPT2.
		IAPLT = 1		MODEA(I,J)	Integer indicating whether to generate a plot of (I,J)th aerodynamic element vs reduced frequency.
				= 1	Generate plot.
				= 0	Do not generate plot. (Default)
		IPKPLT ≠ 0		MODEPK(I)	Integer indicating whether to generate a plot showing Ith characteristic root variation.
				= 1	Generate plot of data. (Default)
				= 0	Bypass Ith root information.
		IPKPLT = ±1, ±3		XRLORG, XRLEND, XRLDEL, & NDXRL	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for abscissa in root-locus plots. (Optional)
				YRLORG, YRLEND, YRLDEL, & NDYRL	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in root-locus plots. (Optional)
		IPKPLT = ±2, ±3		VORD, VEND, VDEL, & NDV	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for abscissa in ζ and ω_n plots. (Optional)

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
3	†INPUT	IPKPLT = ±2, ±3		FORG, FEND, FDEL, & NDF	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in ω_n plots. (Optional)
				GORD, GEND, GDEL, & NDG	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in ζ plots.
4	†SENLOC	ISENSE = 1		XS(I) & YS(I)	Arrays defining sensor locations. These should be unrotated coordinates I = 1,NS.
				ITYPE(I)	Integer indicating whether sensors measure linear or angular motion.
				= 1	Linear.
				= 2	Angular.
				NSS(I)	Number of the structural section on which sensor is located.
5	†SENINP	ICASE = ±2 & ICONSYS = 1 or ICASE = ±1 & ICHSEN = 1 (Changes only)	a_{Sd}	ASD(I,J)	Ith coefficient of denominator polynomial of transfer function which models sensor dynamics for Jth sensor (constant term first). I = 1, ..., 11.
			a_{Sn}	XKS(I,J)	Ith coefficient of numerator polynomial of transfer function which models sensor dynamics for Jth sensor (constant term first). I = 1, ..., 10.
				IORD(J)	Integer indicating type of sensor that sensor J is.
				= 0	Position sensor. (Default)
				= 1	Rate sensor.
				= 2	Acceleration sensor.
			J_j	ISDYN(J)	Integer indicating whether to include sensor dynamics.
				= 1	Include sensor dynamics.
				= 0	Perfect sensor; do not include dynamics. (Default)
6		ICASE = ±2 & ICONSYS = 1 or ICASE = ±1 & ICHFIL = 1		I & J	Integers indicating control-sensor pair for which sensor data are to be filtered. I and J refer to identification prior to any selection. Only control-sensor pairs for which data are to be filtered need be input. End all filter input with 0,0.
7	†FILTIN	ICASE = ±2 & ICONSYS = 1 or ICASE = ±1 & ICHFIL = 1	ω_{n_F} & ζ	WN1 & SN1	Natural frequency and damping ratio in second-order notch. See section "Compensation options."
			K_0	K1BD0	Gain for integral feedback.
			K_1	K1BD1	Gain for proportional feedback.
			K_2	K1BD2	Gain for derivative feedback.
			τ_1 & τ_2	AN1 & AN2	Time constants of zero and pole in lead-lag filter.

†Individual parameters are optional if default values are acceptable.

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
7	†FILTIN	ICASE = ±2 & ICONSYS = 1 or ICASE = ±1 & ICHFIL = 1	a_{Fn} & a_{Fd}	AFN(K,L) & AFD(K,L)	Kth coefficient (constant term first) of numerator and denominator polynomials for Lth filter in transfer function relating control I to sensor J. (See set 6) K = 1, ..., 6.
				IFILTER	Array of integers indicating type of filters being combined to make up transfer function relating control I to sensor J. Example: IFILTER = 3, 4, 1 means transfer function will be composed of a type 3 and 4 filter. (Default = 1, no filter)
				= 1	No further filtering. (Default)
				= 2	Notch filter.
				= 3	Integral filter.
				= 4	Proportional plus derivative filter.
				= 5	Lead-lag or lag-lead filter.
				= 6	General rational function (quotient of two polynomials).
8	†ACTINP	ICASE = ±2 & ICONSYS = 1 or ICASE = ±1 & ICHACT = 1	a_{An} & a_{Ad}	AACTN(I,J) & AACTD(I,J)	Ith coefficient (constant term first) of numerator and denominator polynomials of transfer function which models actuator for Jth control.
				IACT(J)	Integer indicating whether to include actuator dynamics. J corresponds to identification prior to any selection.
				= 1	Include actuator dynamics for Jth control.
				= 0	Perfect actuator; do not include any dynamics. (Default)
9	†SELECT	ISELECT ≠ 0		NMODES	Total number of modes to be selected from available data. After selection, NM is set to NMODES.
				NRNEW	New number of rigid-body modes; i.e., new NR.
				NCNEW	New number of control modes; i.e., new NC.
				NOC	Ascending array of mode numbers indicating for which modes modal data are to be selected. (NMODES = Total number of nonzero NOC values) Example: NOC = 2, 7, and 9 and NMODES = 3 indicates that data for three modes, namely nodes 2, 7, and 9, will be selected for analysis. This option effects a truncation of the model.
				NKNEW	New number of reduced frequencies; i.e., new NK.
				= 0	Indicates no frequency selection; all will be used. (Default)
		ISELECT ≠ 0 & NKNEW ≠ 0		NOK	Ascending array of reduced frequency indices indicating for which frequencies aerodynamic force data are to be selected. Example: NOK = 1, 2, 5 and NKNEW = 3 indicate only aerodynamic force data corresponding to three reduced frequencies, namely 1st, 2nd, and 5th, will be used in analyses.

†Individual parameters are optional if default values are acceptable.

TABLE II.- Continued

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
10		IDELPLT ≠ 0		IPLT2	Array of numbers indicating which sets of characteristic root data are to be saved from file. If ICASE > 0 (interactive execution) then these can be modified before actual deletions occur.
11		IPKPLT > 0		ICOUNT	Number of sets of characteristic root data to be plotted, currently, out of total number of sets of data generated during different cases. Complete list available is printed out before this input is required.
12		IPKPLT > 0		IPLT	Array of numbers indicating which sets of characteristic root data are to be plotted.
13		IPKPLT > 0		ICOMBIN	Integer indicating whether to combine plots of different sets of characteristic root data into one plot (all on same frame) or generate plots on separate frames.
				= 1	Combine plots.
				= 0	Do not combine plots.
14		IPKPLT > 0 & ICOMBIN = 1		PTITLE	80-character description of data being plotted. This will appear on plot being generated in lieu of original plot titles when plots are being combined.
15	†PLOTP	IPKPLT > 0		IDASH	Indicates type of line used to connect points in data currently being plotted. This may be changed for each set of data plotted.
				= 0	____________________ (solid line)
				= 1	— — — — — — — — — — — — — —
				= 2	____________ __ ____________
				= 3	____________ __ __ __________
				= 4	____________ __ __ __ _______
				= 5	______ ______ __ ______ ______
				= 6	_____ _____ __ __ ______ ____ . .
				= 7	_____ ______ __ __ __ _____ _____
				ISAME	Integer indicating how many of next sets of characteristic root data can be plotted with same PLOTP namelist data currently being read-in, including current set. The program will not request another PLOTP namelist to be input until n = ISAME sets of data have been plotted. (Default = 1)
				KVAR	Input negative of value of KVAR that was used when determining roots if you desire to change the independent variable in the graphics output (see definition of KVAR on p. 101).
				MODEPK(I)	Integer indicating whether to generate plot showing Ith characteristic root variation.
				= 1	Generate plot of data. (Default)
				= 0	Bypass Ith root information.
		IPKPLT = +1 or +3		XRLORG, XRLEND, XRLDEL, & NDXRL	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for abscissa in root-locus plots. (Optional)

†Individual parameters are optional if default values are acceptable.

TABLE II.- Concluded

Set	Namelist	When required	Algebraic variable	FORTRAN variable	Description
15	†PLOTP	IPKPLT = +1 or +3		YRLORG, YRLEND, YRLDEL, & NDYRL	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in root-locus plots. (Optional)
		IPKPLT = +2 or +3		VORD, VEND, VDEL, & NDV	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for abscissa in ζ and ω_n plots. (Optional)
				FORG, FEND, FDEL, & NDF	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in ω_n plots. (Optional)
				GORG, GEND, GDEL, & NDG	Lower limit, upper limit, increment between major tic marks, and number of minor divisions between major tic marks for ordinate in ζ plots.
16		IPKPLT > 0		IOP	Integer indicating whether to continue plotting. This option allows user to generate additional plots of characteristic root data that either were not included before or for which changes in the plots are desired.
				= 1	Continue plotting.
				= 0	Return to main program and continue with a new case or terminate execution.

†Individual parameters are optional if default values are acceptable.

TABLE III.- DESCRIPTION OF TAPE5 INPUT DATA

[All data are read-in using free-field format]

Set	When required	Algebraic variable	FORTRAN array name	Description	READ statements
1	ICASE = ± 2 and/or ISELECT > 0	$\{k_i\}$	KK	Reduced frequencies at which aerodynamic data were computed	FOR K=1,NK READ*,KK(K) where KK(K)=k_K
2	ICASE = ± 2 and/or ISELECT > 0	Re{Q} & Im{Q}	VARDR & VARDI or AR & AI	Real and imaginary parts of aerodynamic motion forces; pairs are read-in by columns (per pressure mode)† per frequency and transposed internally	FOR K=1,NK FOR J=1,NM(+NG)† FOR I=1,NM READ*,(VARDR(K,J,I),VARDI(K,J,I) where VARDR(K,J,I)=$\mathrm{Re}\{Q(k_K)_{I,J}\}$ and VARDI(K,J,I)=$\mathrm{Im}\{Q(k_K)_{I,J}\}$
3	ICASE = ± 2 and/or ISELECT > 0	M	MAT1	Generalized masses; read-in by rows	FOR I=1,NM FOR J=1,NM READ*,MAT1(I,J) where MAT1(I,J)=$M_{I,J}$
4	ICASE = ± 2 and/or ISELECT > 0	$\{f_n\}$	FREQ	Natural frequencies (Hz)	FOR I=1,NM READ*,FREQ(I) where FREQ(I)=f_{n_I}
5	ICASE = ± 2 and/or ISELECT > 0 & ISTIFF = 1	K	MOMSQ	Generalized stiffness matrix; read-in by rows	FOR I=1,NM FOR J=1,NM READ*,MOMSQ(I,J) where MOMSQ(I,J)=KI,J
6	ICASE = ± 2 and/or ISELECT > 0	$\{g_s\}$	G	Structural damping coefficients	FOR I=1,NM READ*,G(I) where G(I)=g_{s_I}
7	ICASE = ± 2 and/or ISELECT > 0 & ICSREAD = 1	H	DS	Sensor deflections; read-in by columns per sensor	FOR J=1,NS FOR I=1,NM READ*,DS(I,J) where DS(I,J)=$H^T_{I,J}$
8	ICASE = ± 2 and/or ISELECT > 0 & ICSACT = 0	H_E	HE	Actuator elongation per unit generalized coordinate; read-in by rows per control	FOR I=1,NC FOR J=1,NM READ*,HE(I,J)
9	ICASE = ± 2 and/or ISELECT > 0 & ICSACT = 0	F_D	FD	Generalized force per unit actuator hinge moment; read-in by columns per control	FOR J=1,NC FOR I=1,NM READ*,FD(I,J)

†If gust forces are included as last NG columns of aerodynamic data, provision has been made to read these in (NG≠0), although they are not stored in these arrays.

TABLE IV.- DESCRIPTION OF TAPE10 INPUT DATA

[All data are read-in using free-field format]

Set	When required	Algebraic variable	FORTRAN array name	Description	READ statements
1	INTERP = 1 or ISENSE = 1	x or y'	TAB(I,1)	x-coordinates if IPLATE = 1 or points along y'-axis if IPLATE = 0 at which modal data are available	FOR I=1,NNODES READ*,TAB(I,1) where TAB(I,1)=x_I or y'_I
2	INTERP = 1 or ISENSE = 1	y or z(y')	TAB(I,2)	y-coordinates at which modal data are available if IPLATE = 1 or bending deflections along the y'-axis if IPLATE = 0	FOR I=1,NNODES READ*,TAB(I,2) where TAB(I,2)=y_I or $z(y'_I)$
3	INTERP = 1 or ISENSE = 1	z or $R'_y(y')$	TAB(I,3)	Modal deflections at (x,y) if IPLATE = 1 or rotation about y' at points along y'-axis if IPLATE = 0	FOR I=1,NNODES READ*,TAB(I,3) where TAB(I,3)=z_I or $R'_y(y'_I)$
4	INTERP = 1 or ISENSE = 1 and NDOF = 4	0 or $R'_x(y')$	TAB(I,4)	Zeros if IPLATE = 1, or rotation about x' at points along y'-axis if IPLATE = 0	FOR I=1,NNODES READ*,TAB(I,4) where TAB(I,4)=0 or $R'_x(y')$
Repeat sets 1 to 4 for each mode (1 to 3 if NDOF = 3)					

TABLE V.- DATA STORED ON DATA STORAGE FILE TAPE1 BY PROGRAM

Record	Algebraic variable	FORTRAN array name	Size	Description
1	M	MAT1	NMODES**2	Selected generalized masses
2	$M\{\omega_n^2\}$	MOMSQ	NMODES	Selected $M\omega_n^2$ array
3	f_n	FREQ	NMODES	Selected frequency (Hz) array
4	$\{\hat{Q}_{ij}\}$ & $[b_{\ell j}]$	SCOF & BN	NCOEF*NMODES**2+250	Selected p-plane coefficients and $b_{\ell j}$ array
5		IA1SUB	102	Subindex array for PKPLOT data
6	$\{k\}^T$ & $Re[Q(k_i),i=1,\dots,n_k]$	RFRQ & AR	NK+NK*NMODES**2	Selected reduced frequencies and real part of aerodynamic data
7	$Im[Q(k_i),i=1,\dots,n_k]$	AI	NK*NMODES**2	Selected imaginary part of aerodynamic data
8	H_E	HE	NCNEW*NMODES	Selected actuator elongation per unit generalized coordinate
9	F_D	FD	NMODES*NCNEW	Selected generalized force per unit actuator hinge moment
10	H	CS	NS*(NMODES-NCNEW)	Selected sensor deflections
11	$\{g_s\}$	G	NMODES	Selected structural dampings
12		DCT, CNTLN, & ICLN	NCTM+(MPM+1)*NS*NCNEW	Control logic common denominator (d(s)), numerator elements, and number of coefficients in each numerator element
14	$\{\hat{Q}_{ij}\}$ & $[b_{\ell j}]$	SCOF & BN	NCOEF*NM**2+250	Full set of p-plane coefficients and $b_{\ell j}$ array
15		INDEX4 for SPLINE	51	Subindex array for geometric coefficients used in surface spline interpolation
16	H	CS	NS*(NM-NC)	Full set of sensor deflections
17		INDEX for FILTIN	51	Subindex used in storing filter data
20	$a_{Sn}, a_{Fn}, \dots$	ASN,ACLN,...	Computed	All intermediate control data used in computing d and N_{ij}
50		All labeled commons		All labeled common data from previous case for restart capability

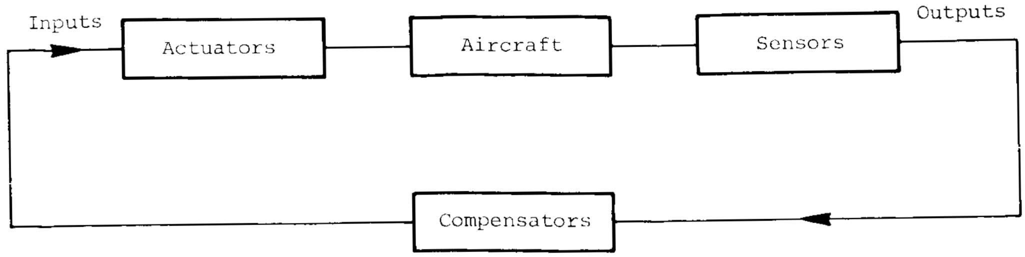


Figure 1.- STABCAR dynamic elements.

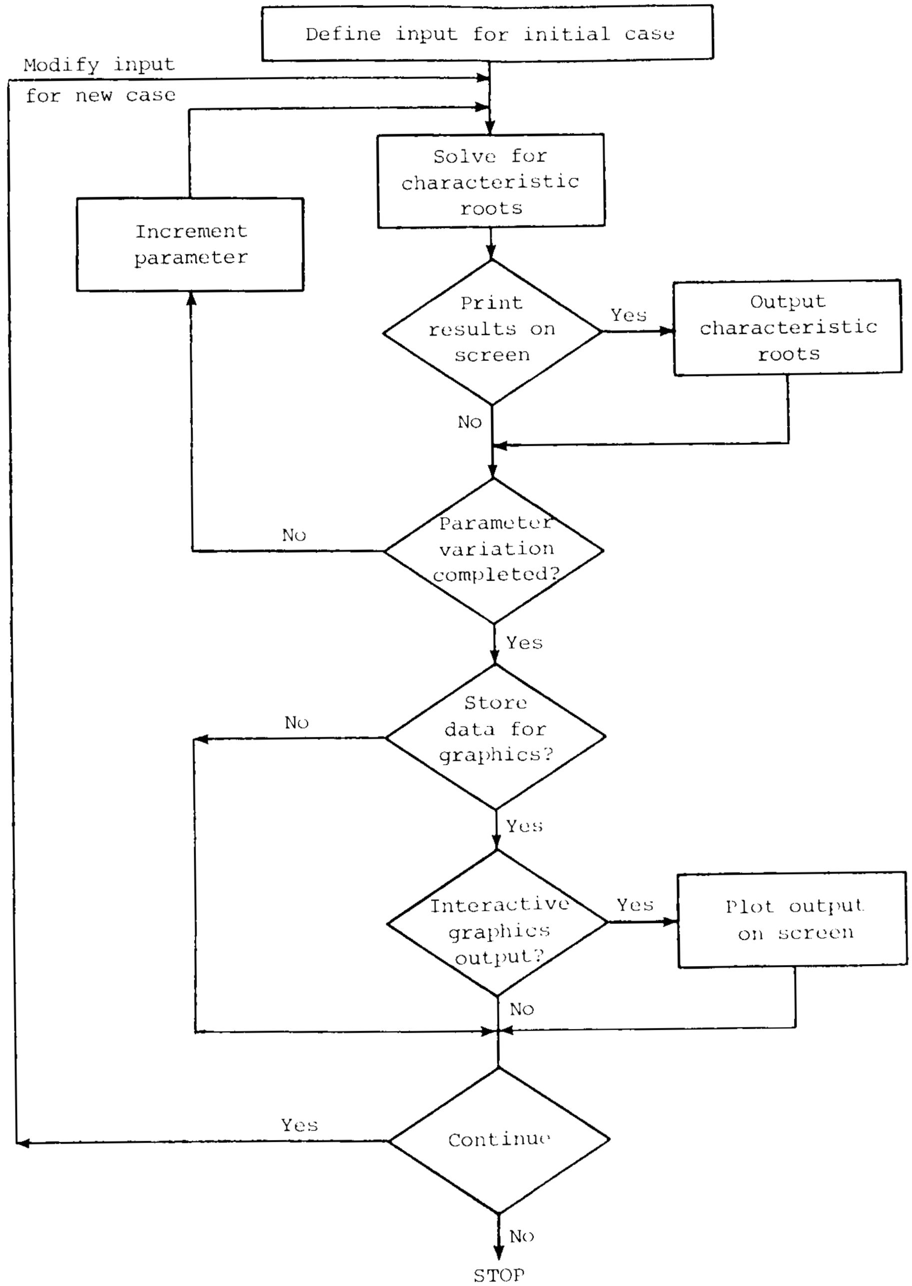
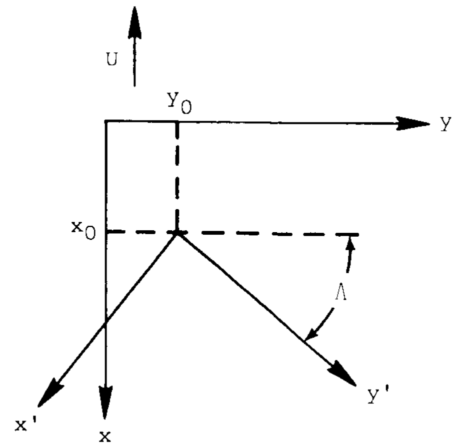


Figure 2.- Characteristic root determination with STABCAR.



Figures 3.- Geometrical relationship between (x,y) and (x',y').

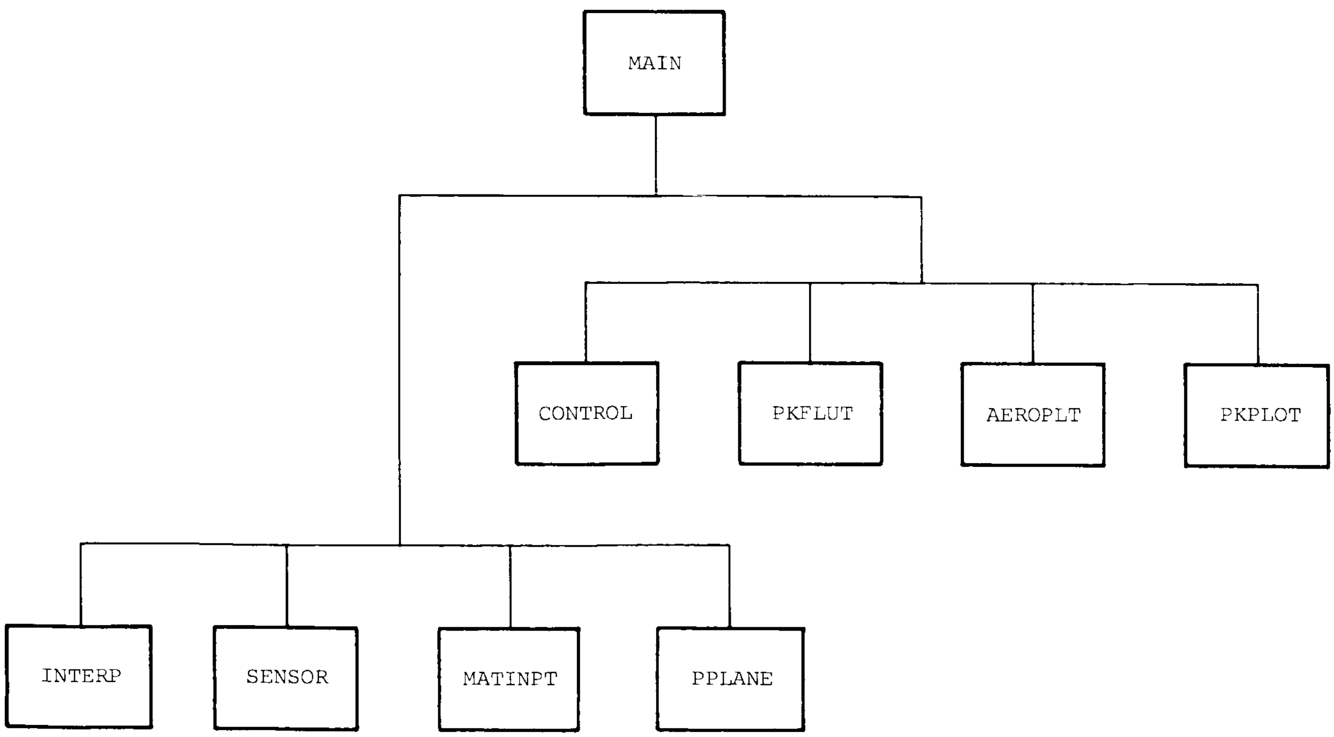


Figure 4.- STABCAR overlay structure.

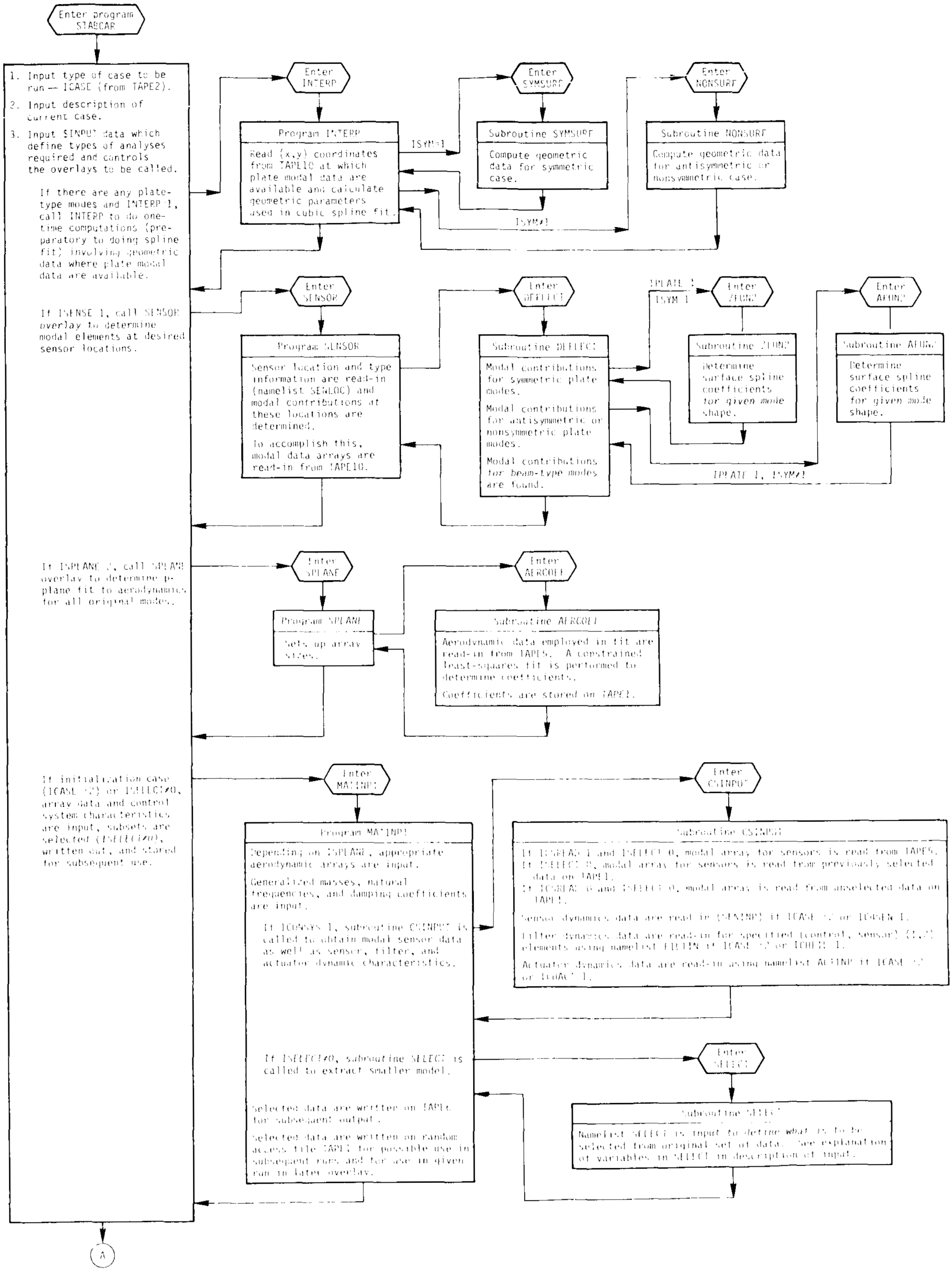


Figure 5.- Flow diagram of program STABCAR.

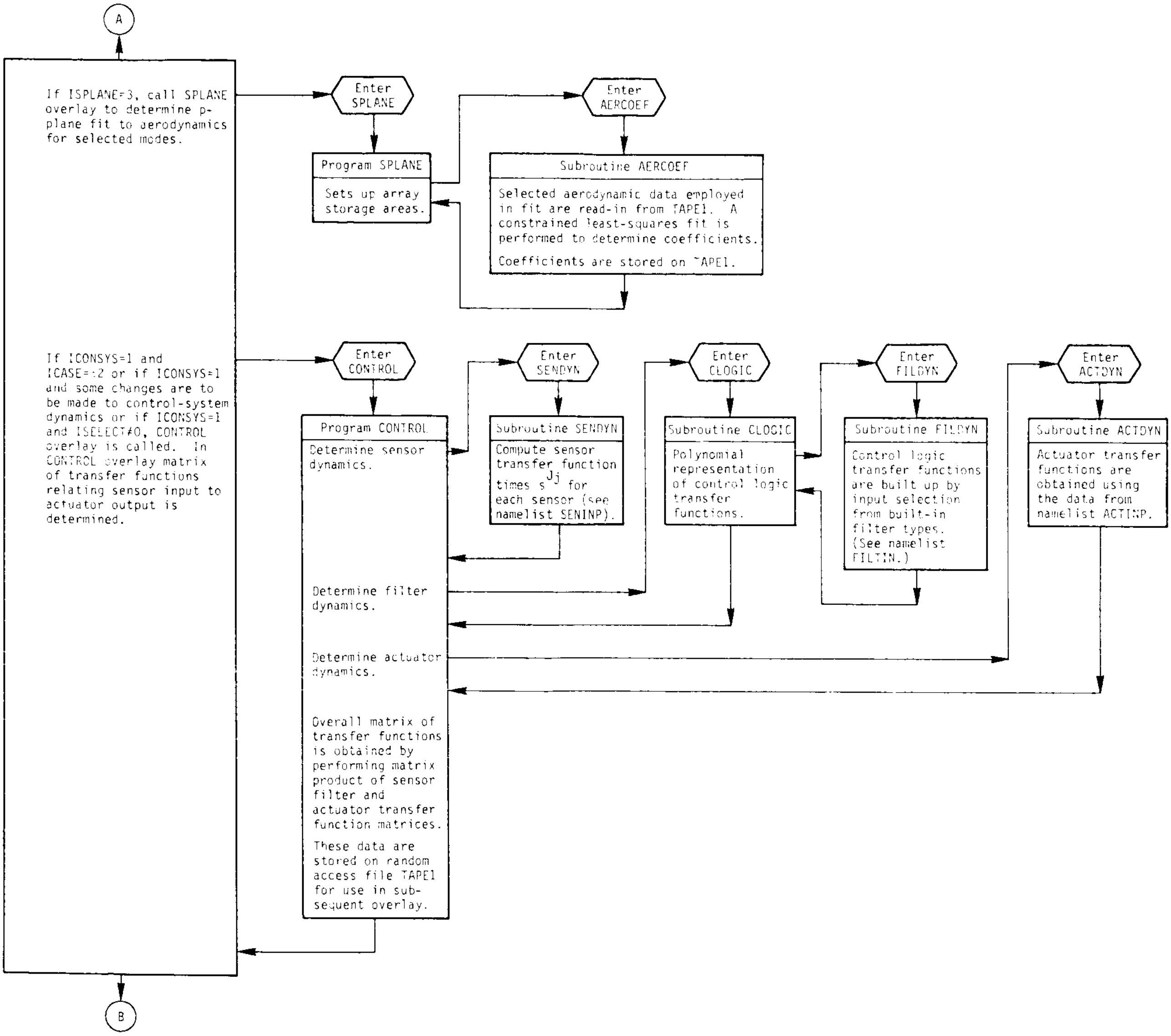


Figure 5.- Continued.

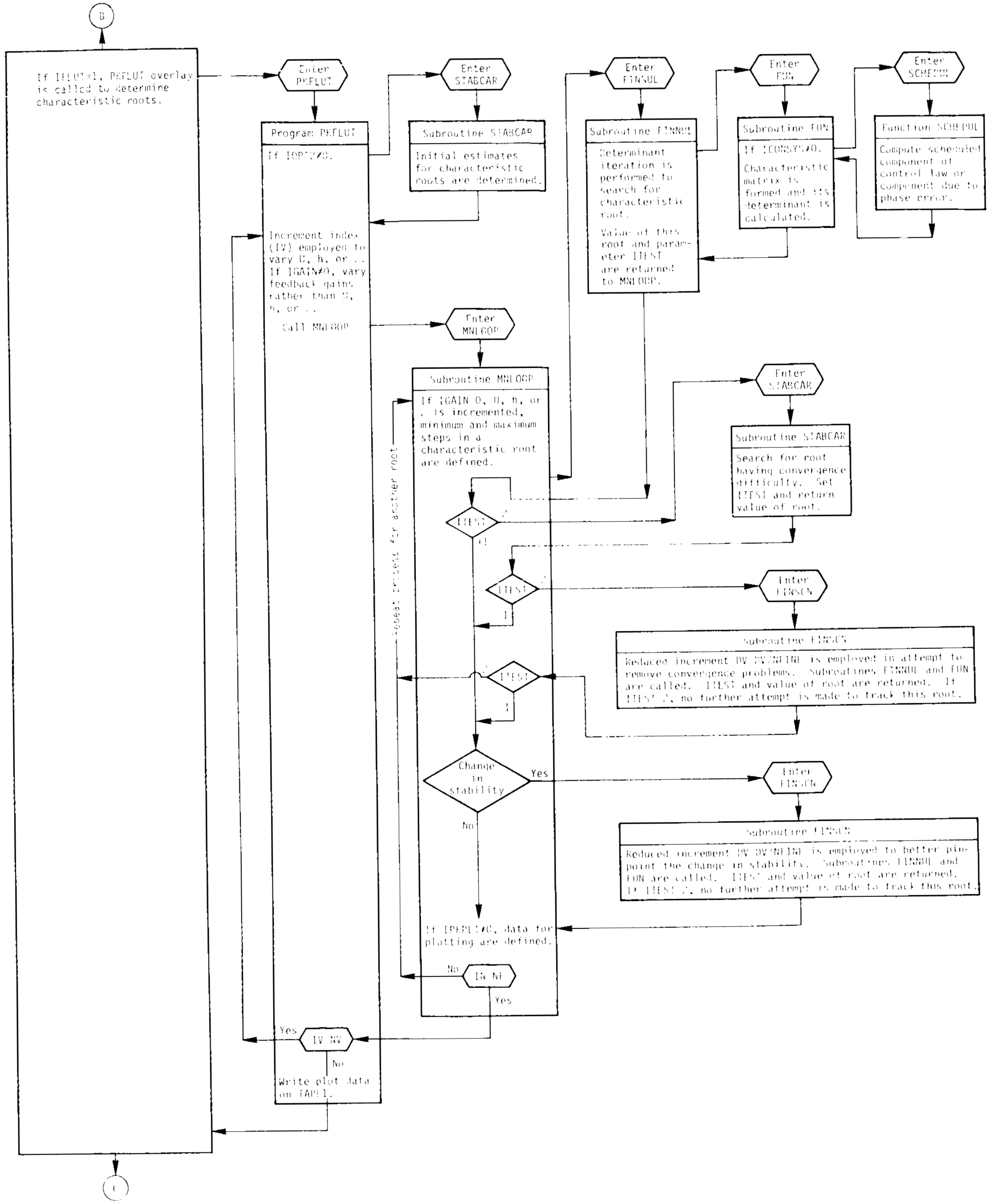


Figure 5.- Continued.

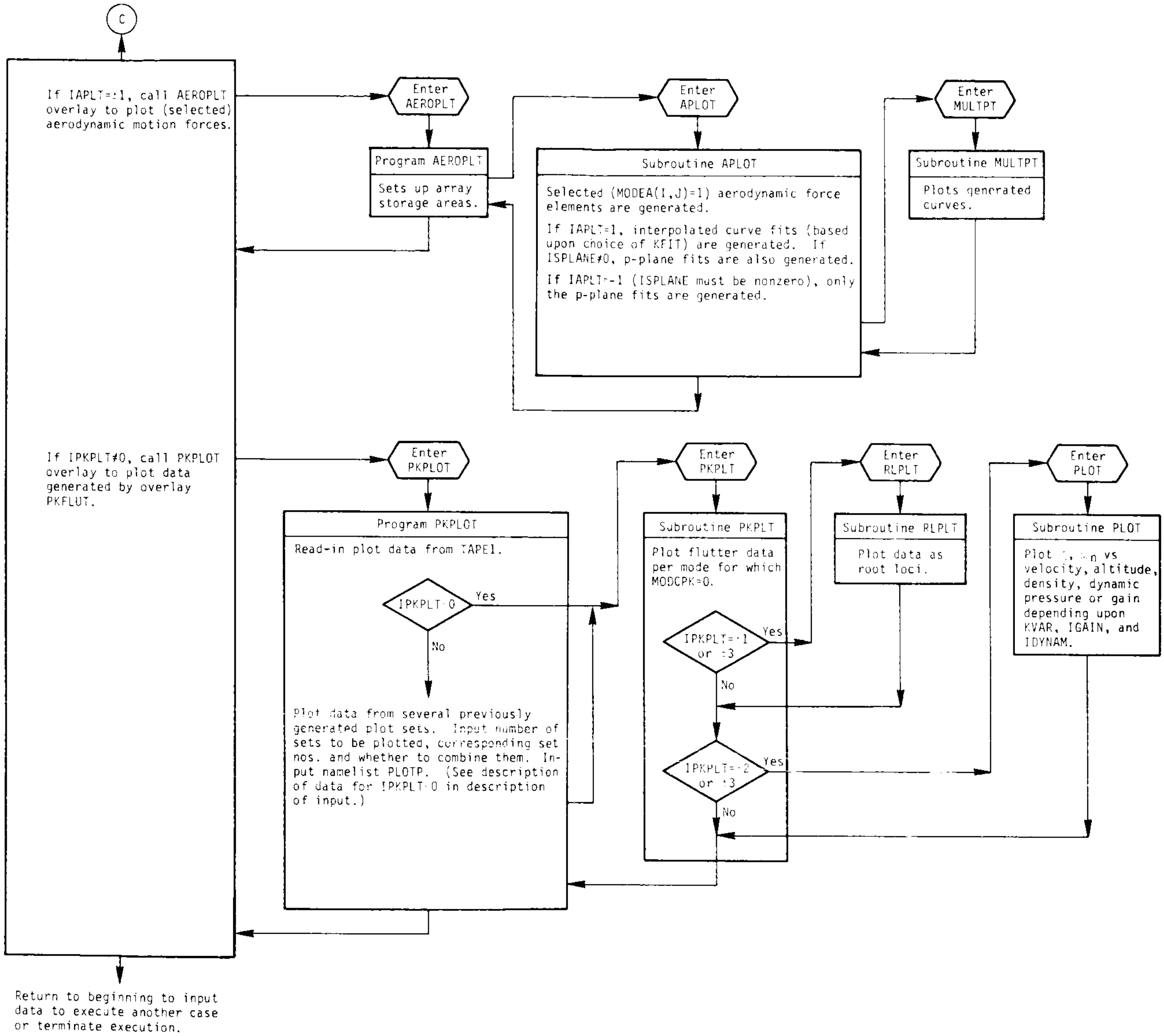


Figure 5.- Concluded.

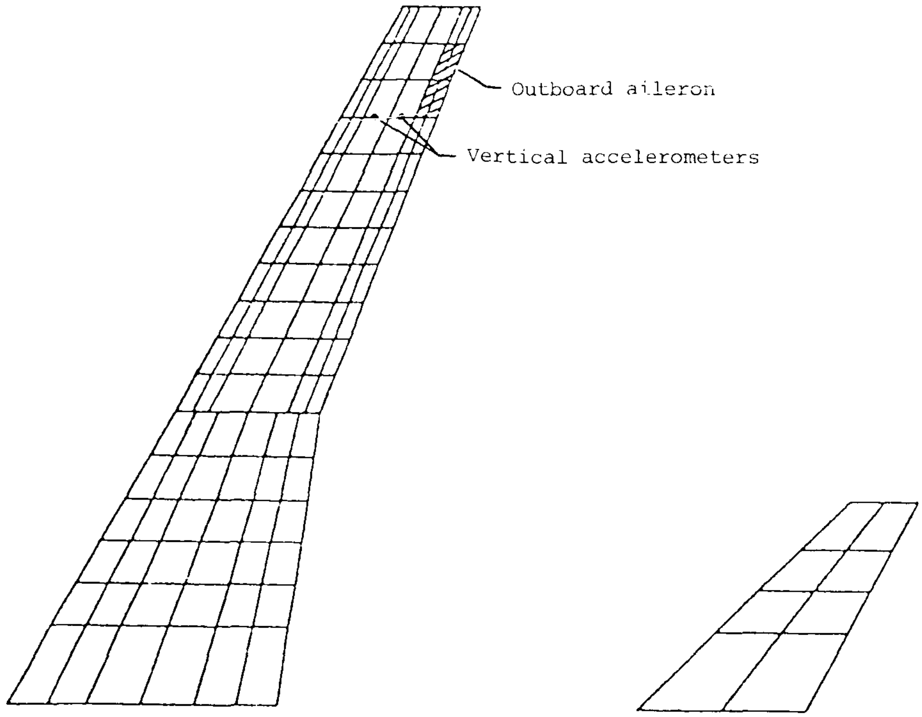


Figure 6.- Paneling of wing and horizontal tail for aerodynamic force computation.

Filter

$\left(\ddot{h}_2 - \ddot{h}_1\right)$, g units

$$649.2 \frac{s}{s + 2} \frac{G_{11}}{s + D_S} \frac{250}{s + 250} \frac{s^2 + 40s + 196^2}{s^2 + 100s + 196.5^2} \frac{s^2 + 220s + 456^2}{s^2 + 400s + 520^2}$$

δ_L, deg

$$-1 \leq G_{11} = \left(3.147 \times 10^{-5}\right)\left(1.61 - N_{Ma}\right)(q - 1034) \leq 1$$

$$75 \leq D_S = 735 - 833.3N_{Ma} + 0.0033267q \leq 300$$

$$\text{if } N_{Ma} < 0.61, \text{ set } N_{Ma} = 0.61$$

$$\text{if } q < 23.924 \text{ kN/m}^2, \text{ set } q = 23.9248 \text{ kN/m}^2$$

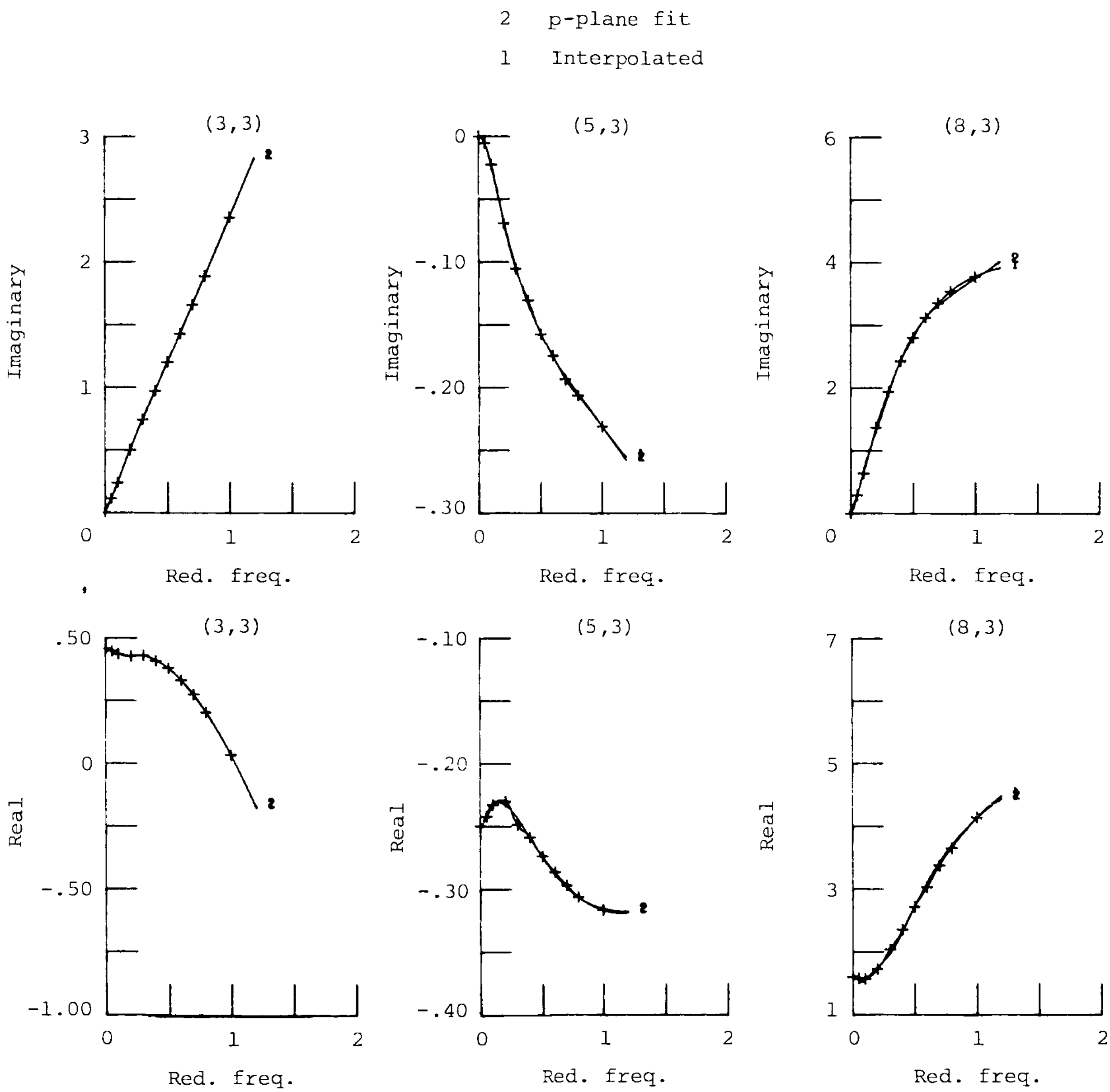
Actuator

δ_L, deg

$$\frac{6.583 \times 10^{14}}{(s + 401)\left(s^2 + 681.7s + 1.028 \times 10^6\right)\left(s^2 + 654.6s + 1.597 \times 10^6\right)}$$

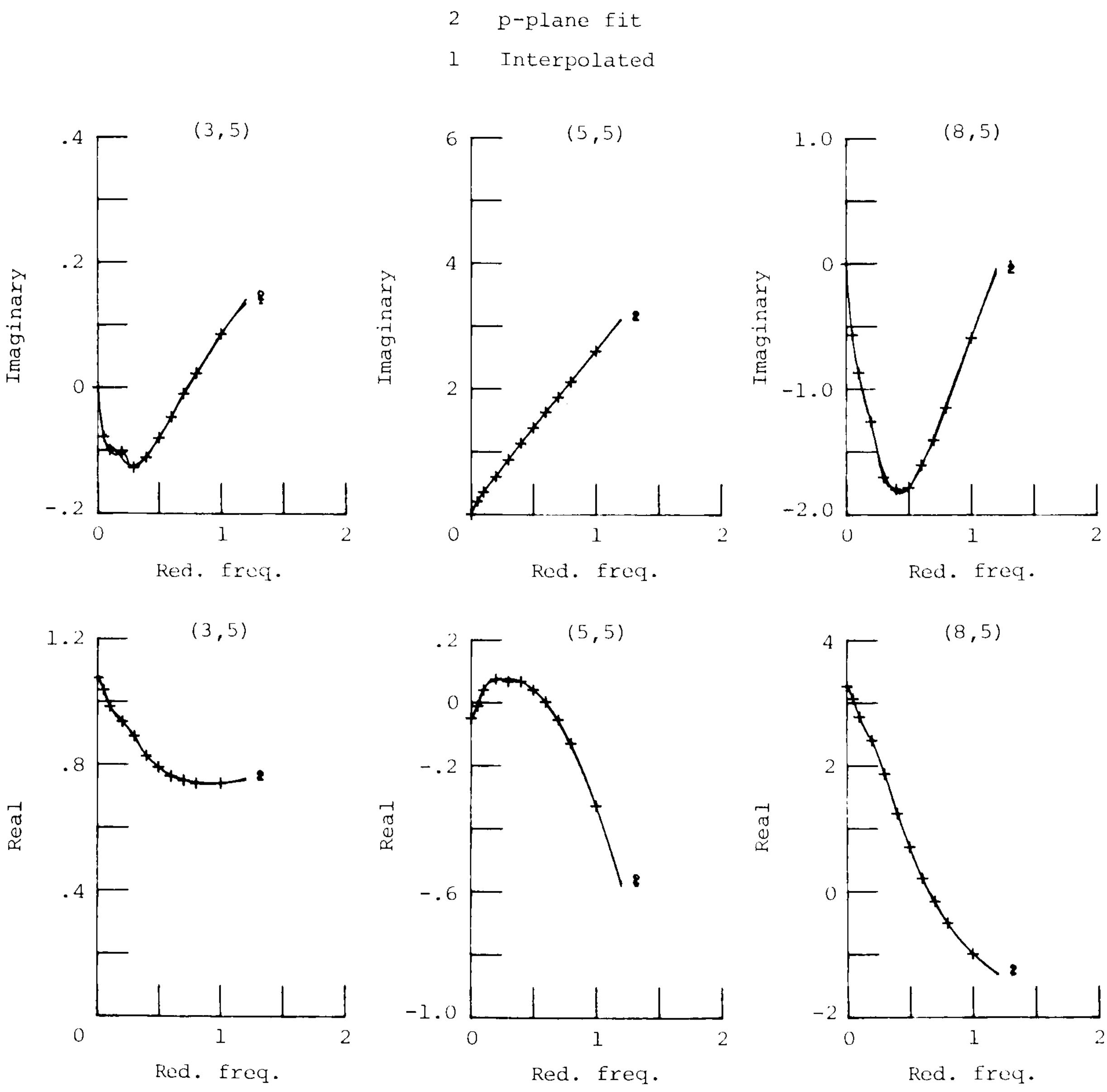
δ, deg

Figure 7.- Control law and actuator dynamics.



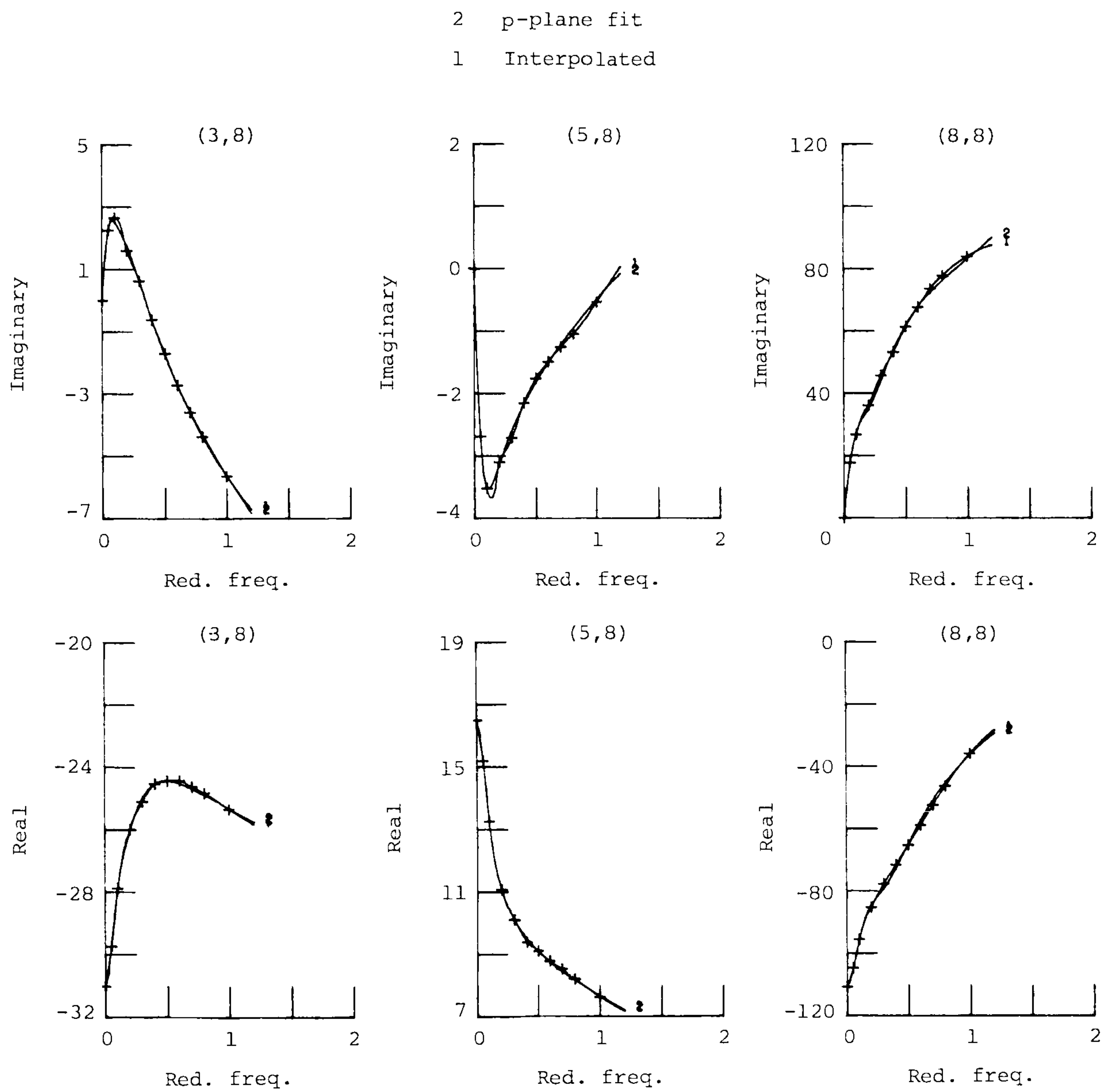
(a) Column 3.

Figure 8.- Selected generalized force matrix elements.



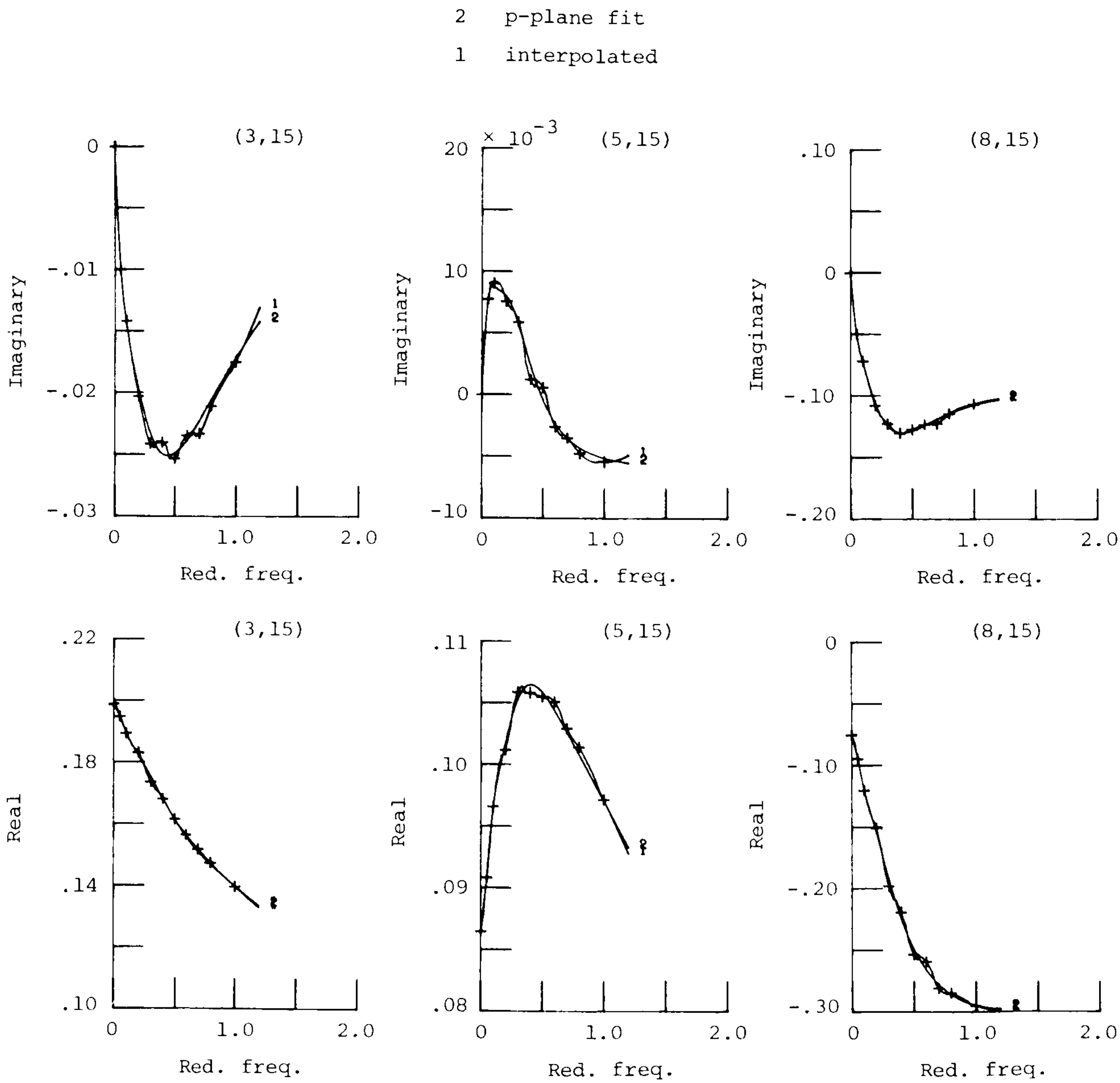
(b) Column 5.

Figure 8.- Continued.



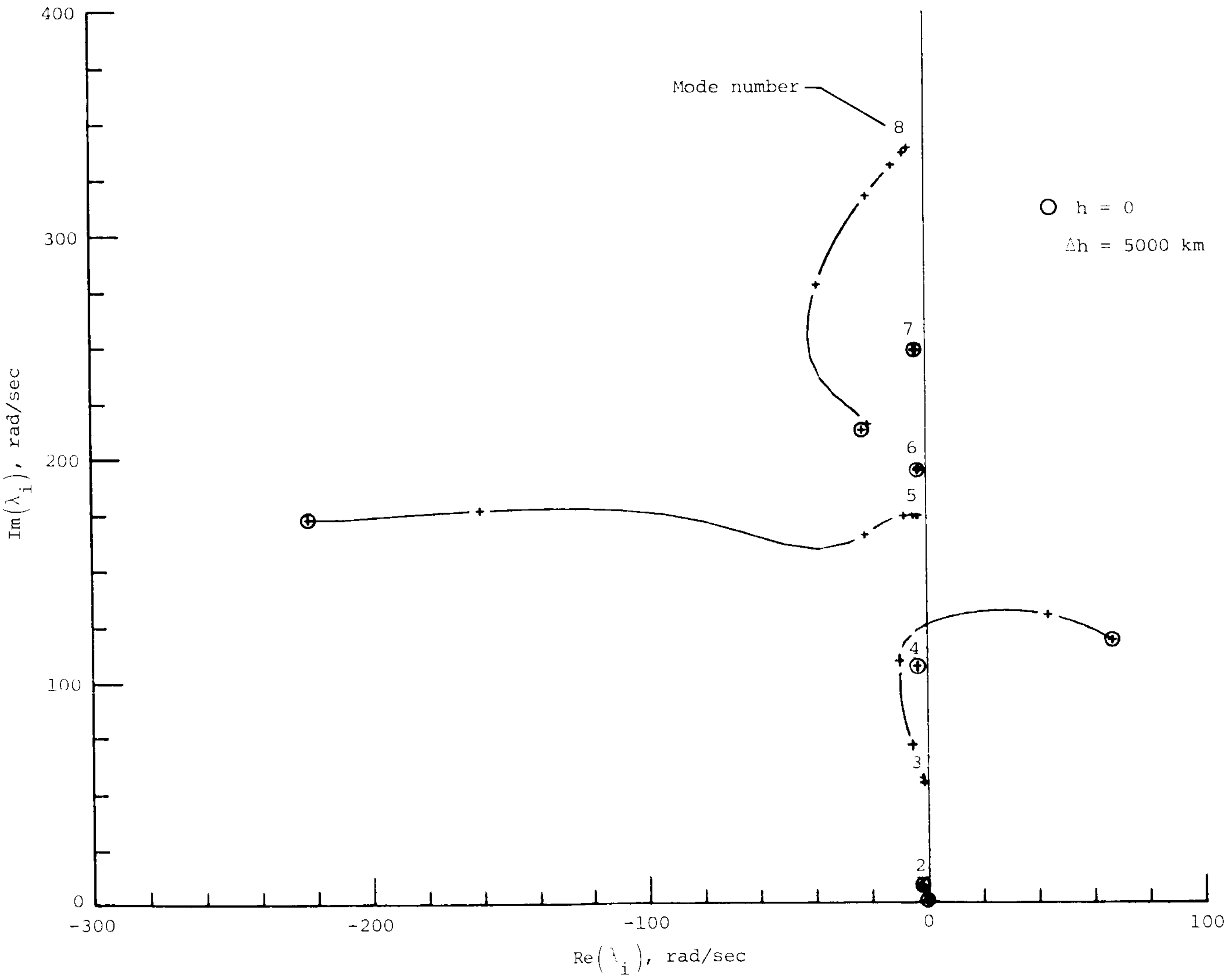
(c) Column 8.

Figure 8.- Continued.



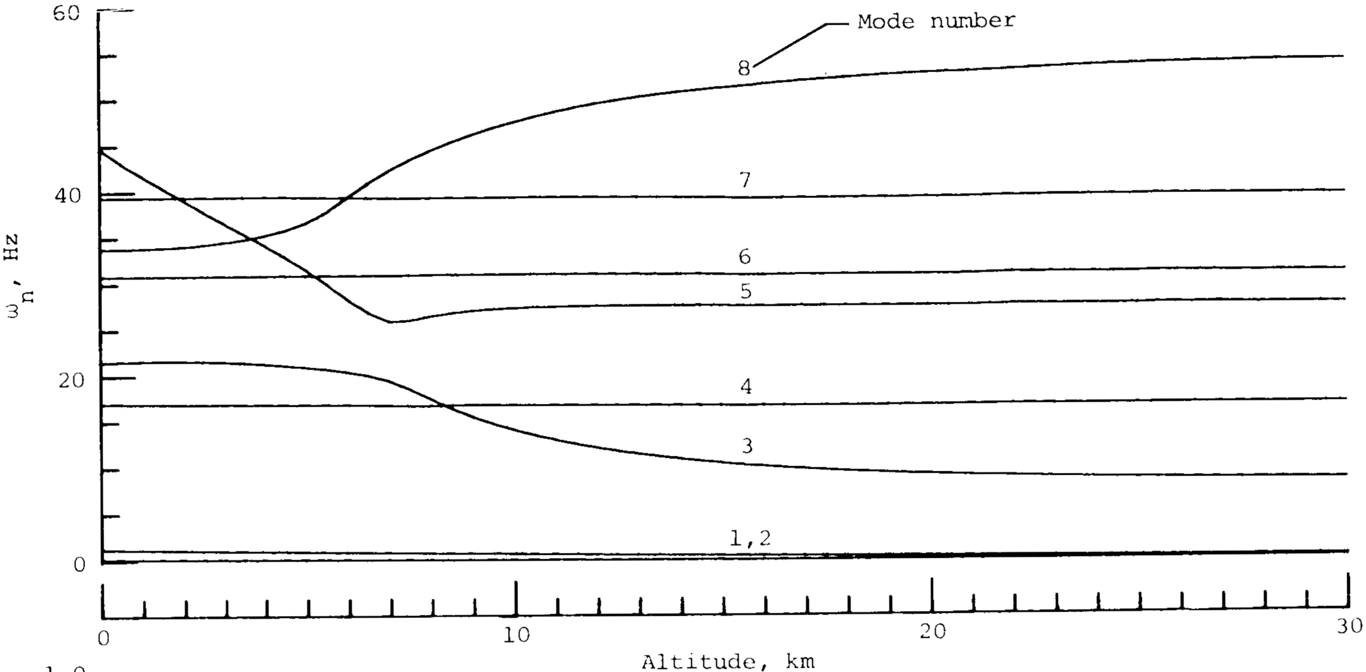
(d) Column 15.

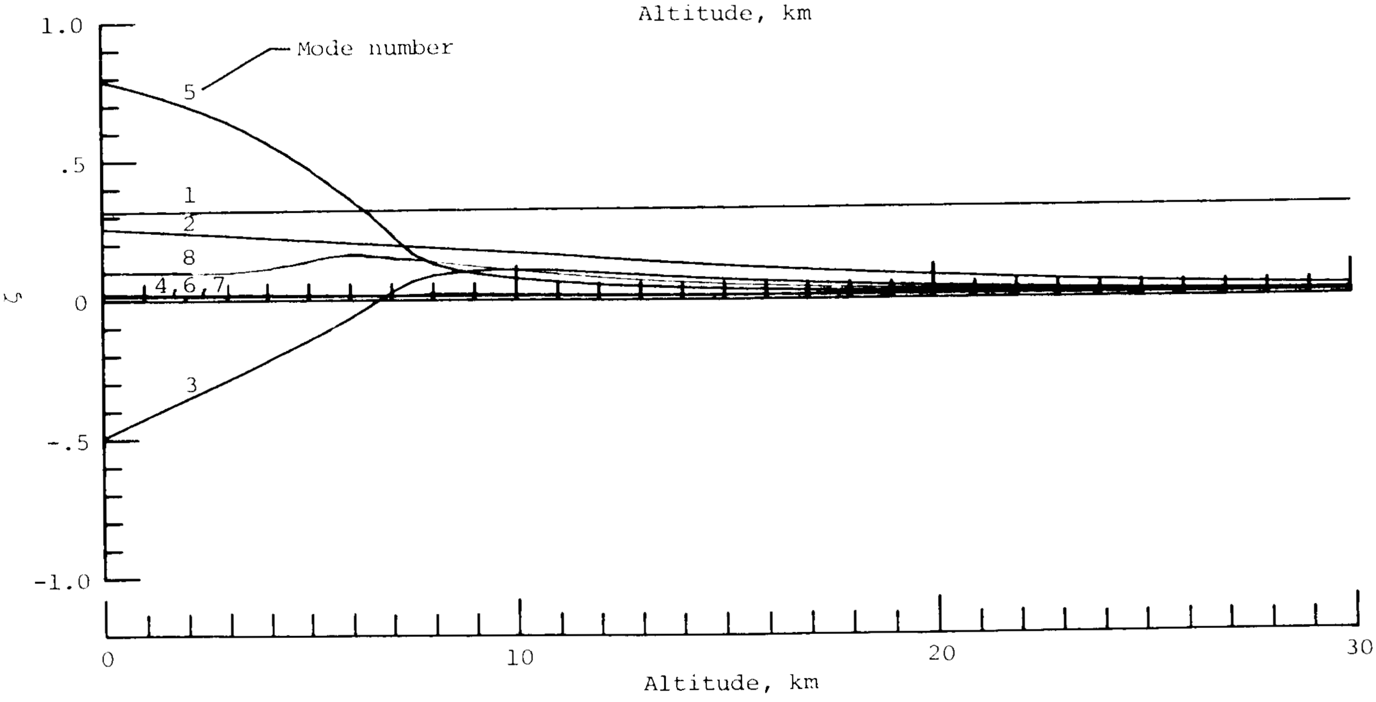
Figure 8.- Concluded.



(a) Altitude root loci.

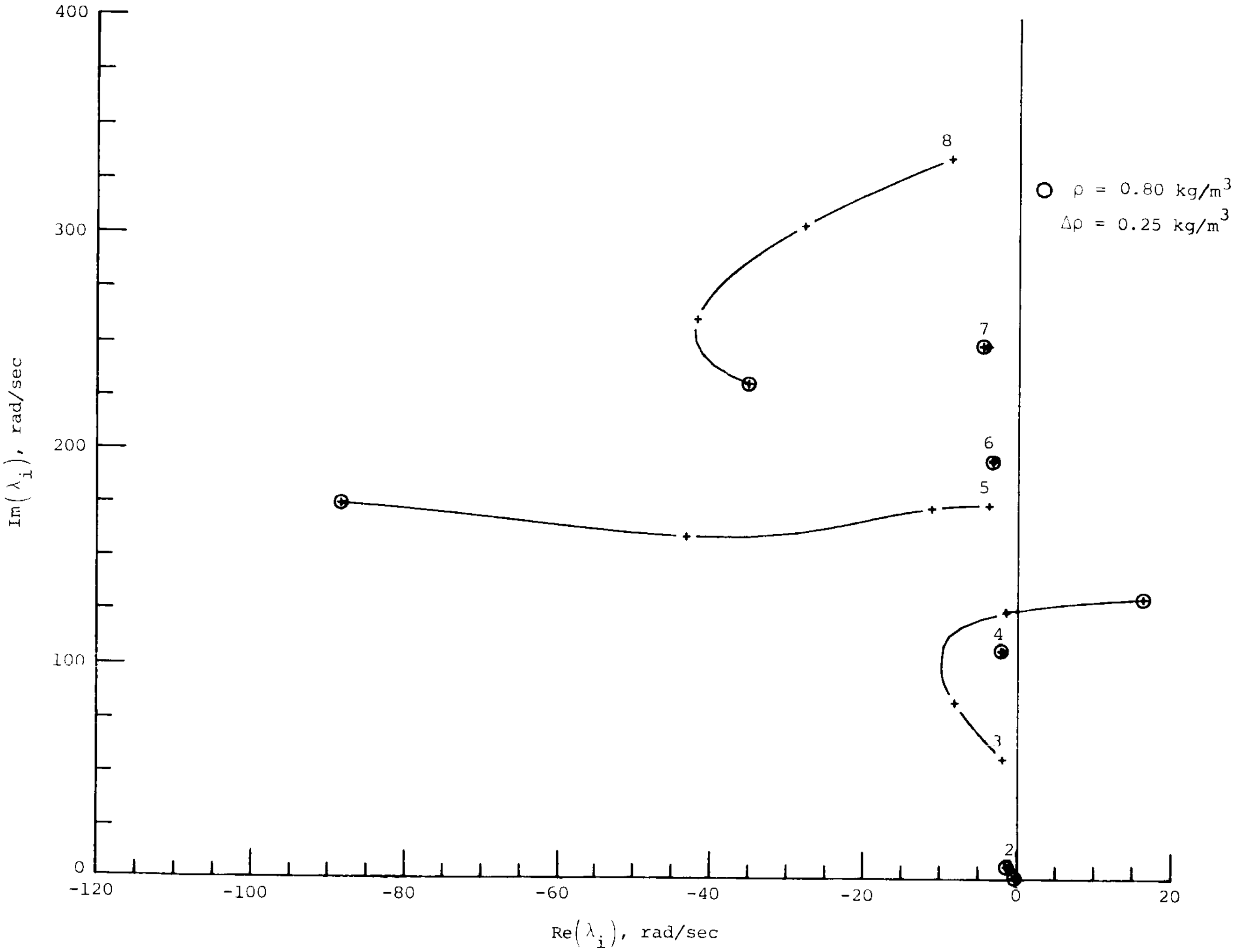
Figure 9.- Open-loop stability characteristic variation with altitude at $N_{Ma} = 0.86$.





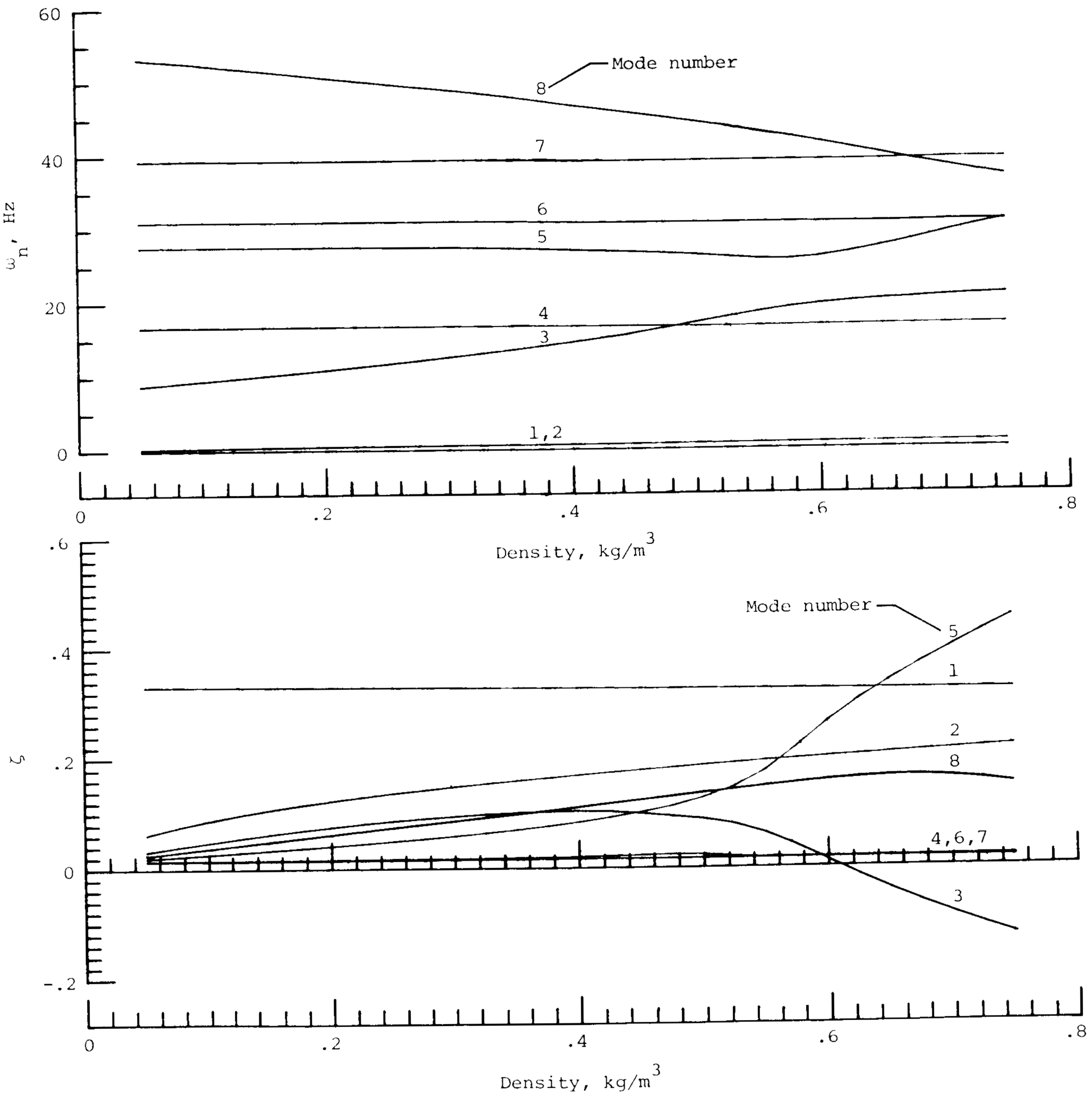
(b) ζ and ω_n versus altitude.

Figure 9.- Concluded.



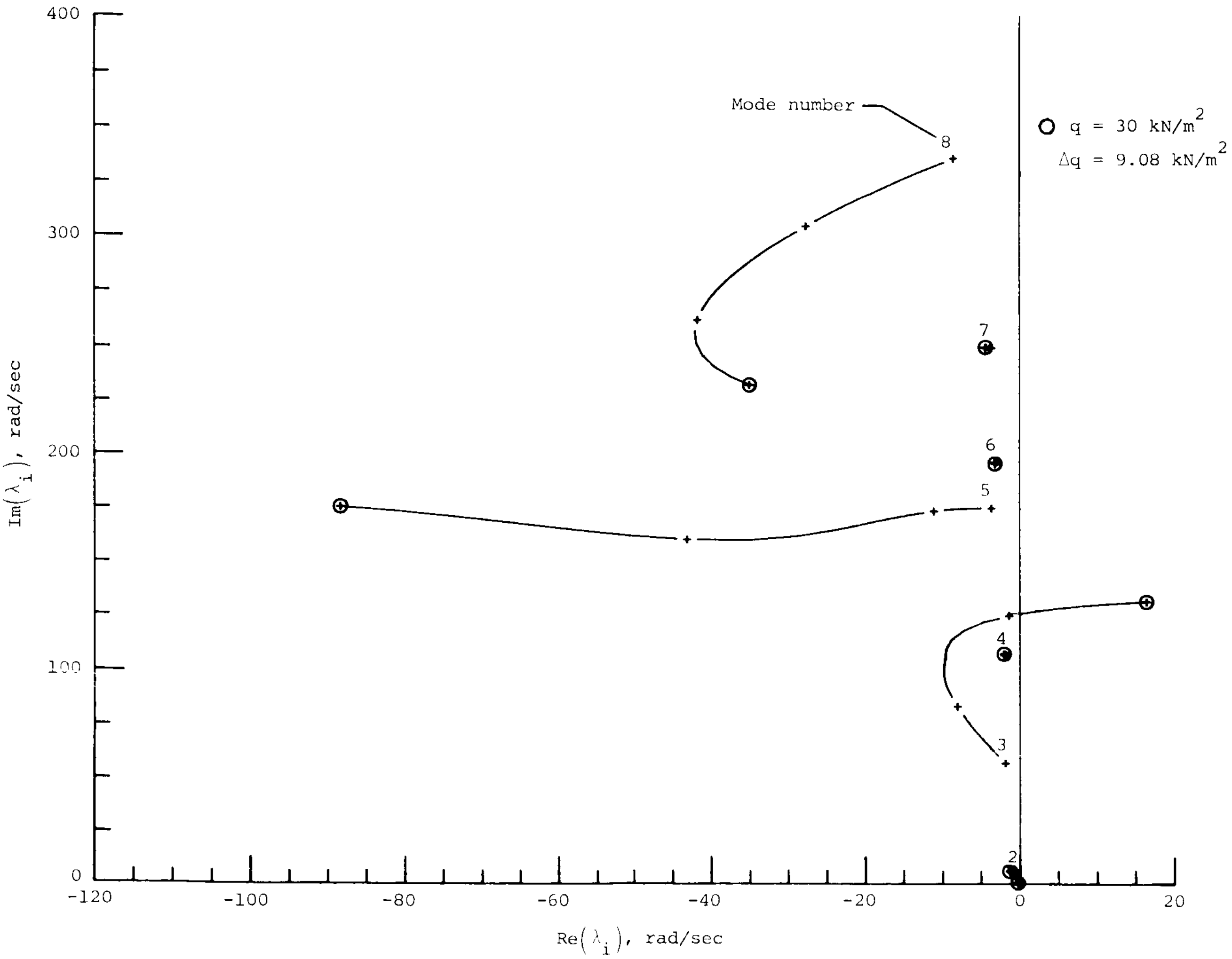
(a) Density root loci.

Figure 10.- Open-loop stability characteristic variation with density. $N_{Ma} = 0.86$; U = 269.6 m/sec.



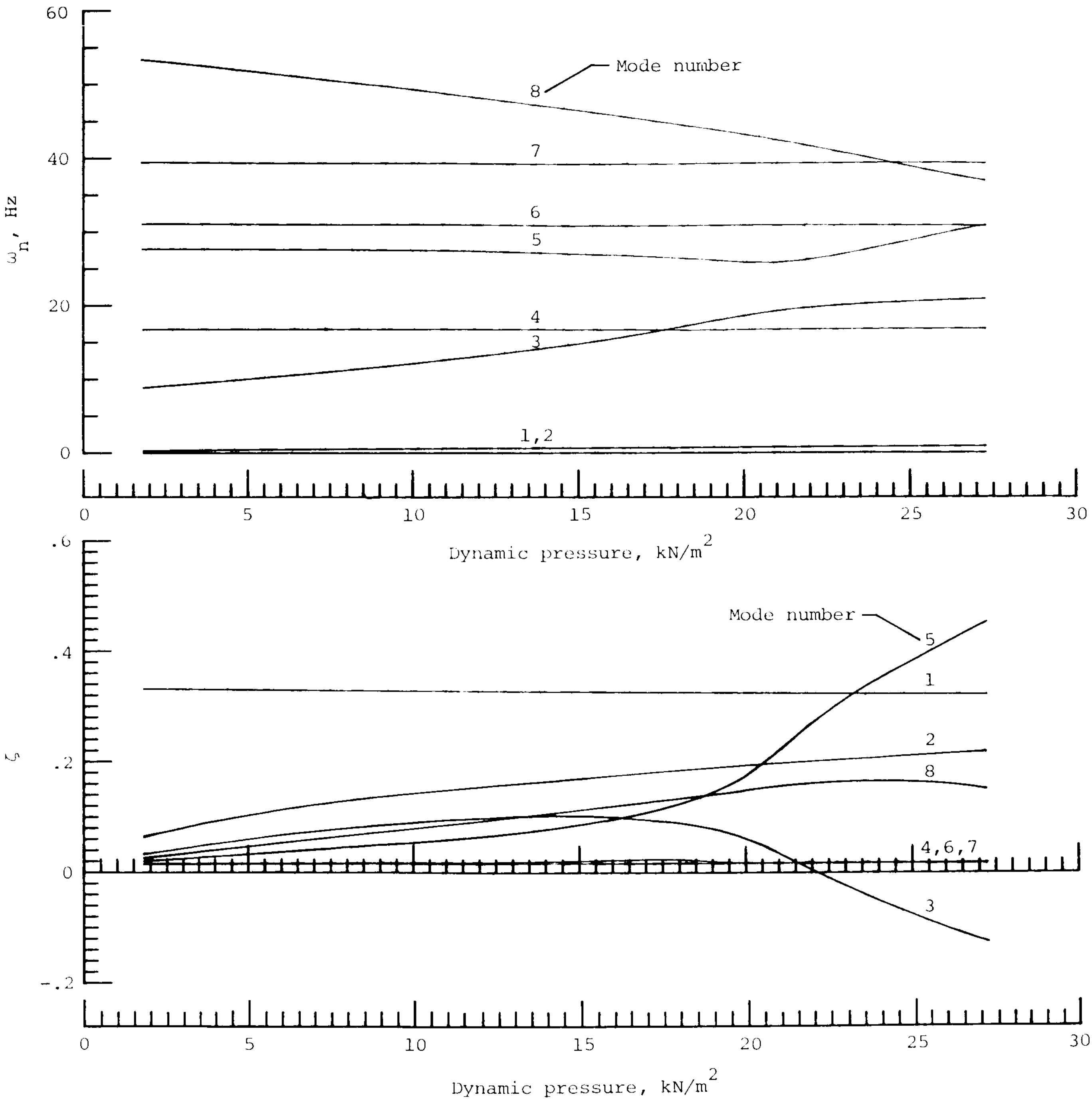
(b) ζ and ω_n versus density.

Figure 10.- Concluded.



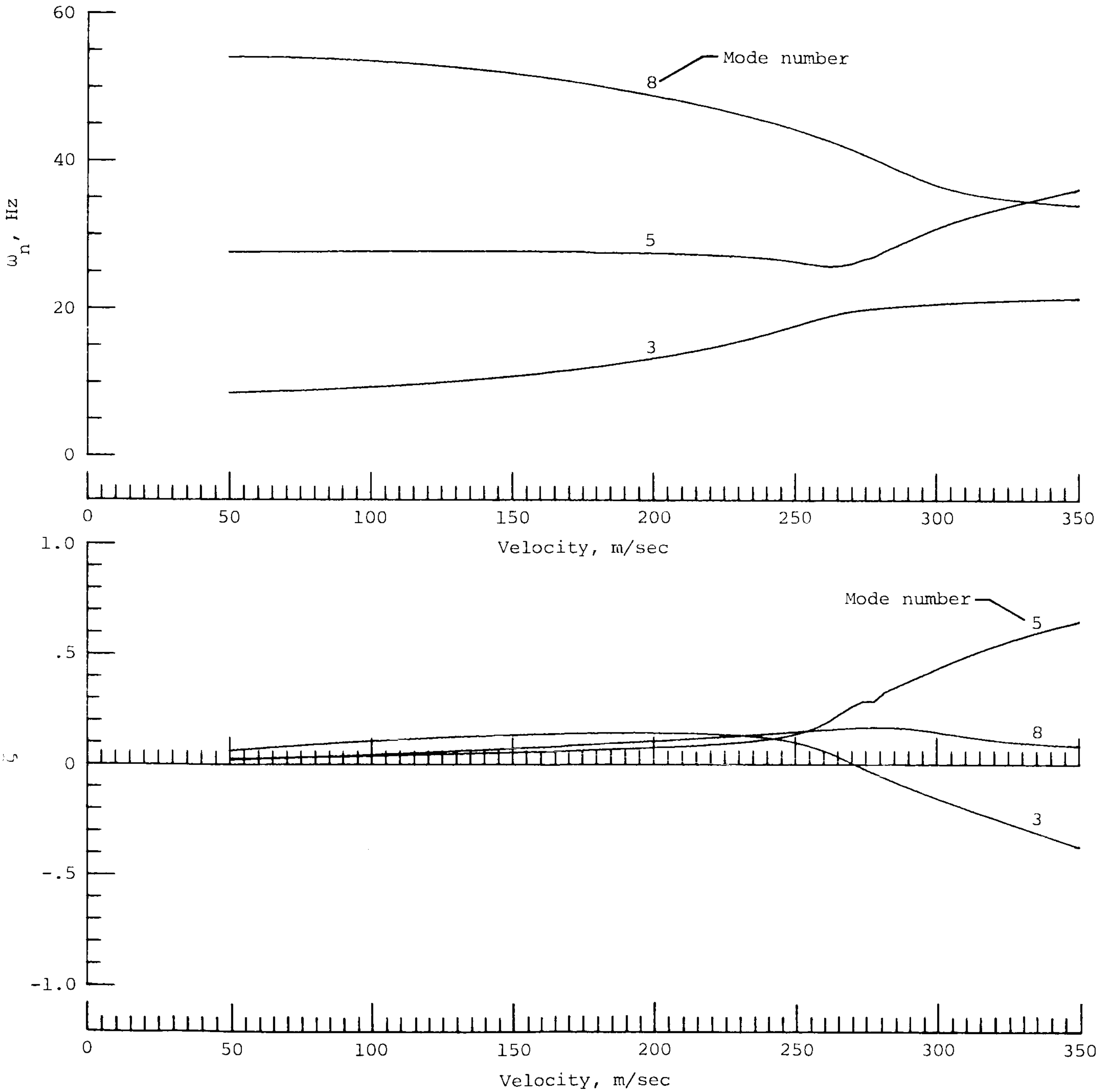
(a) Dynamic-pressure root loci.

Figure 11.- Open-loop stability characteristic variation with dynamic pressure. $N_{Ma} = 0.86$; $U = 269.6$ m/sec.



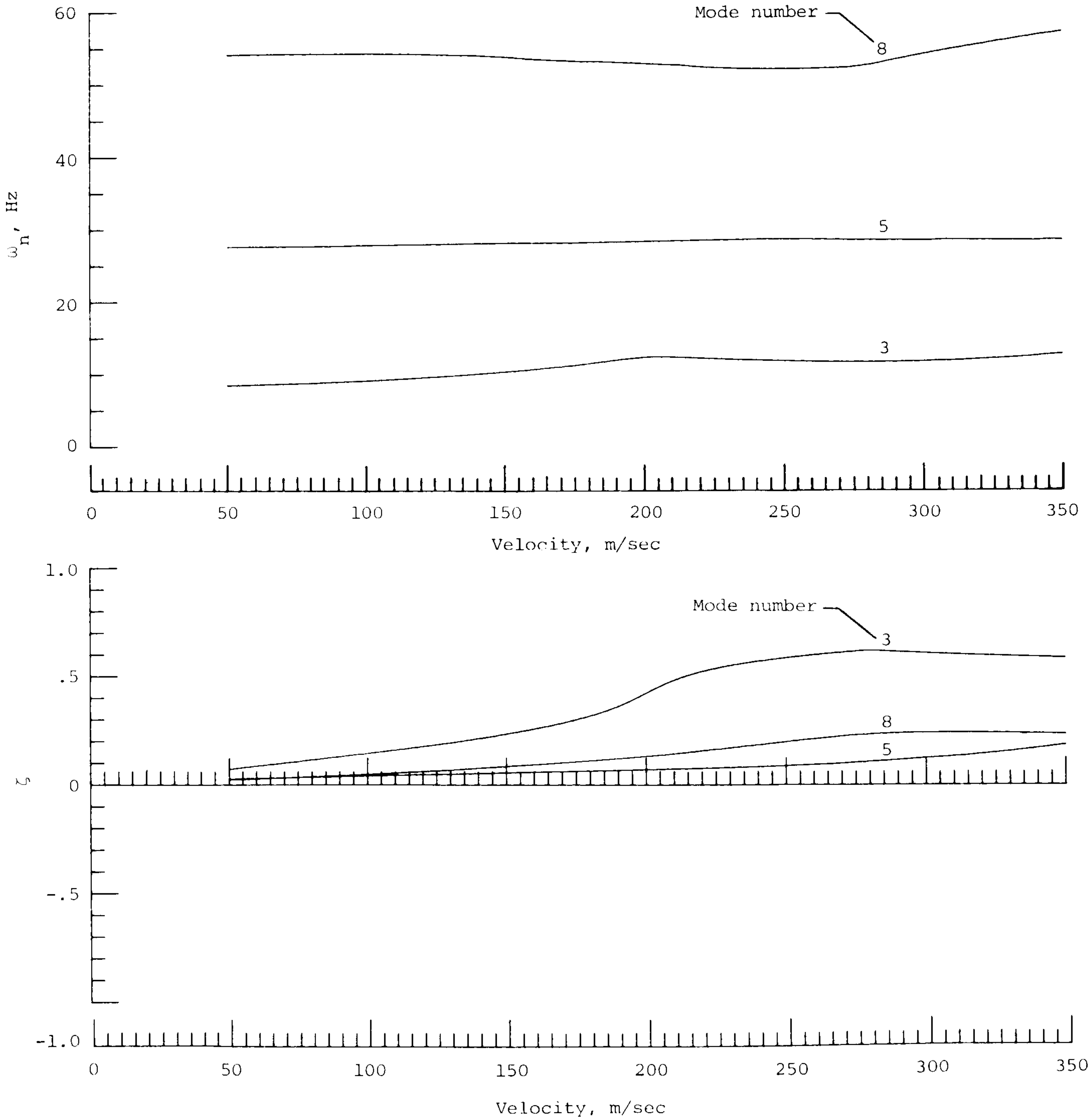
(b) ζ and ω_n versus dynamic pressure.

Figure 11.- Concluded.



(a) Open loop.

Figure 12.- Characteristic root variation with velocity (reduced-order model). $N_{Ma} = 0.86$; $\rho = 0.6101$ kg/m^3.



(b) Closed loop.

Figure 12.- Concluded.

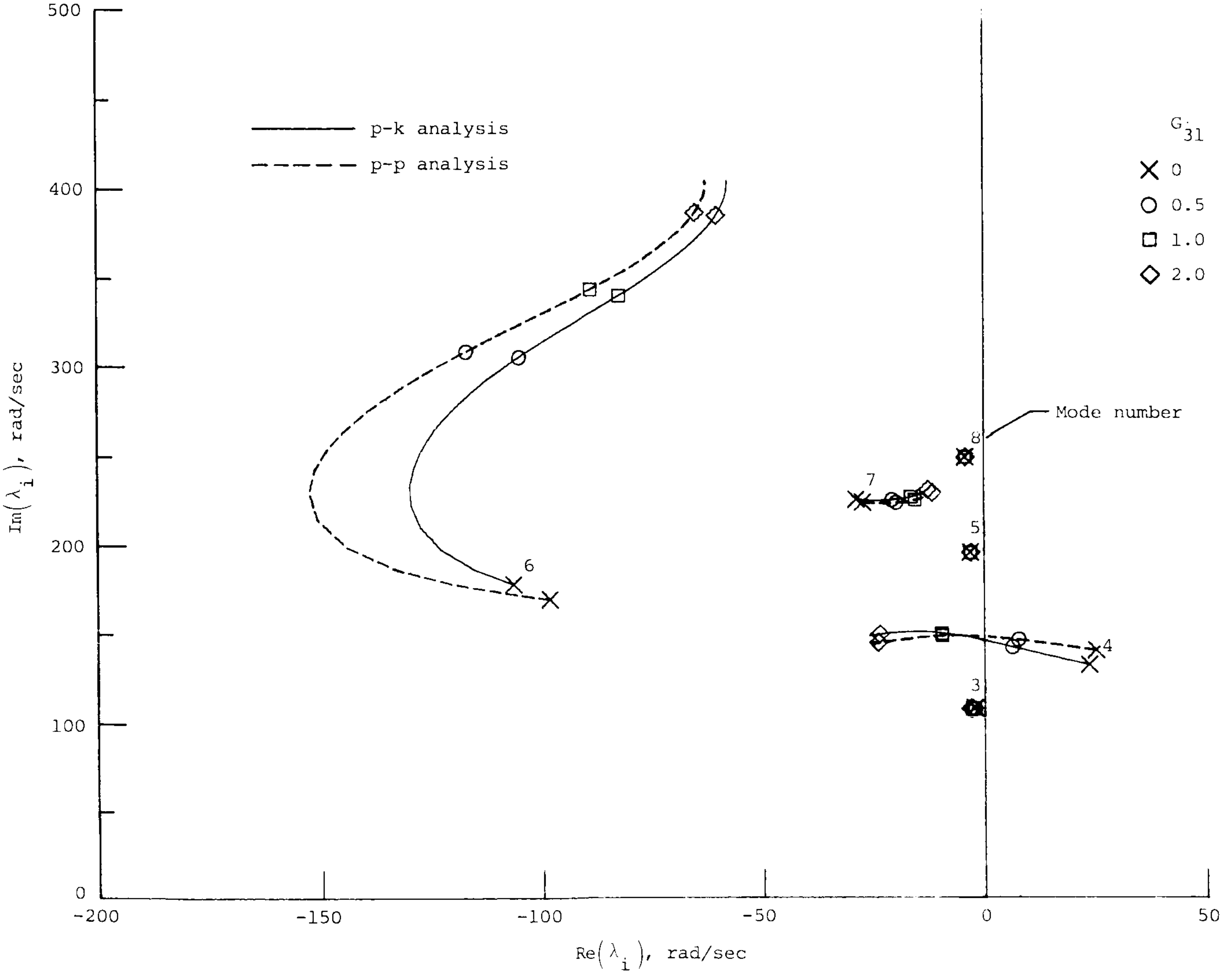


Figure 13.- Elastic mode gain root loci at U_D. $N_{Ma} = 0.86$; $h = 4.572$ km.

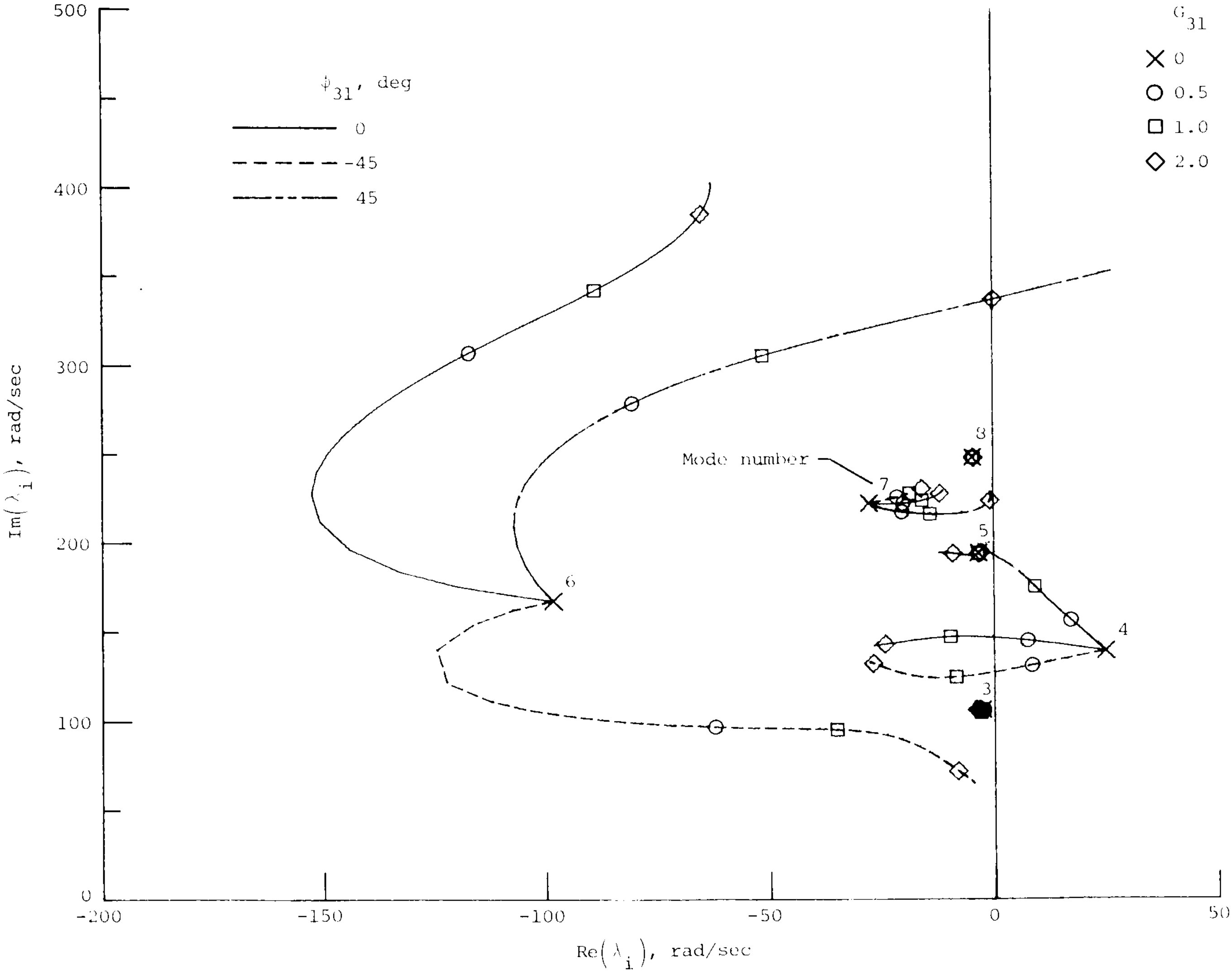


Figure 14.- Effect of phase errors in control signal elastic mode gain root loci at U_D (p-p analysis). $N_{Ma} = 0.86$; $h = 4.572$ km.

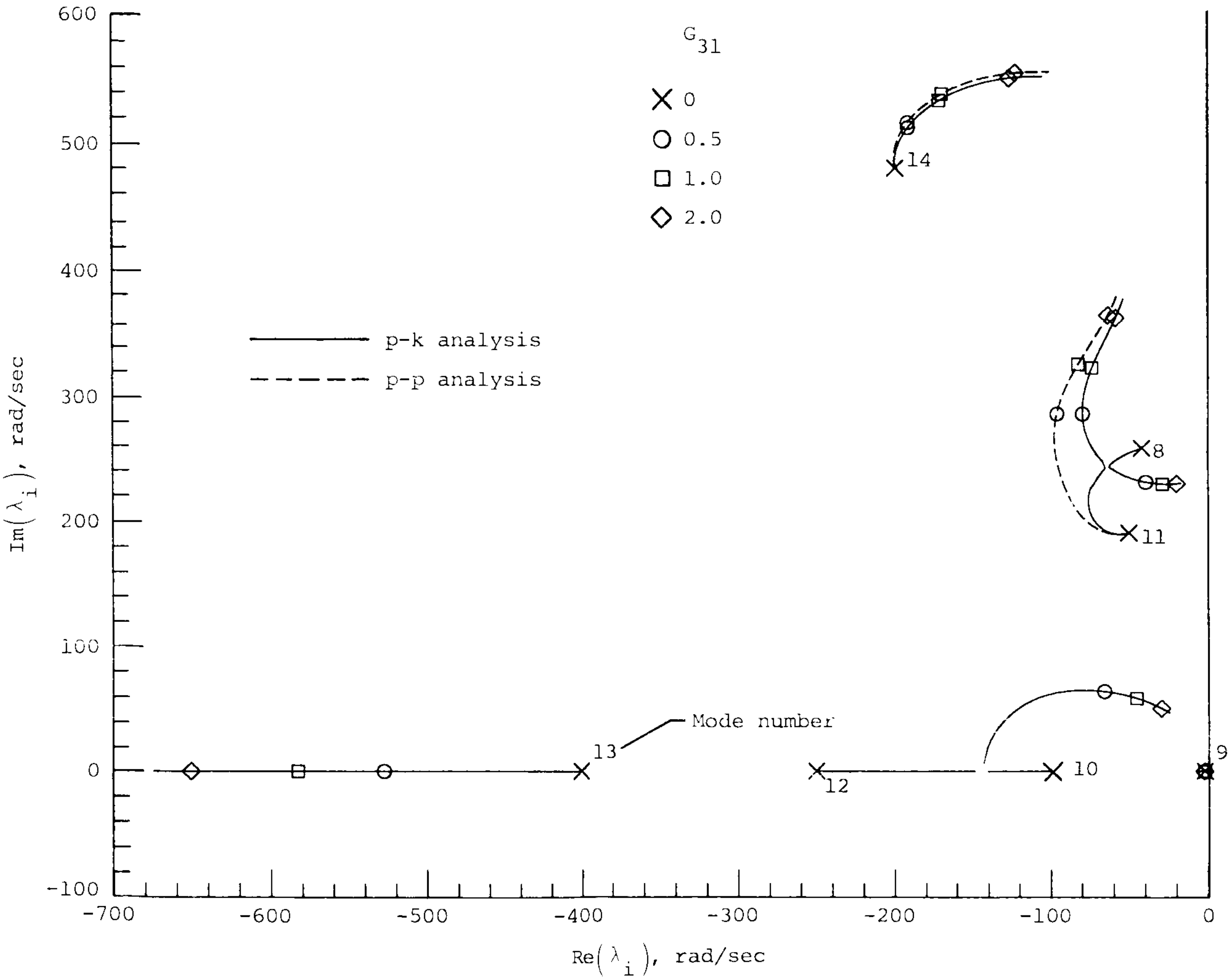


Figure 15.- Control root loci at U_F. N_{Ma} = 0.86; h = 6.705 km.

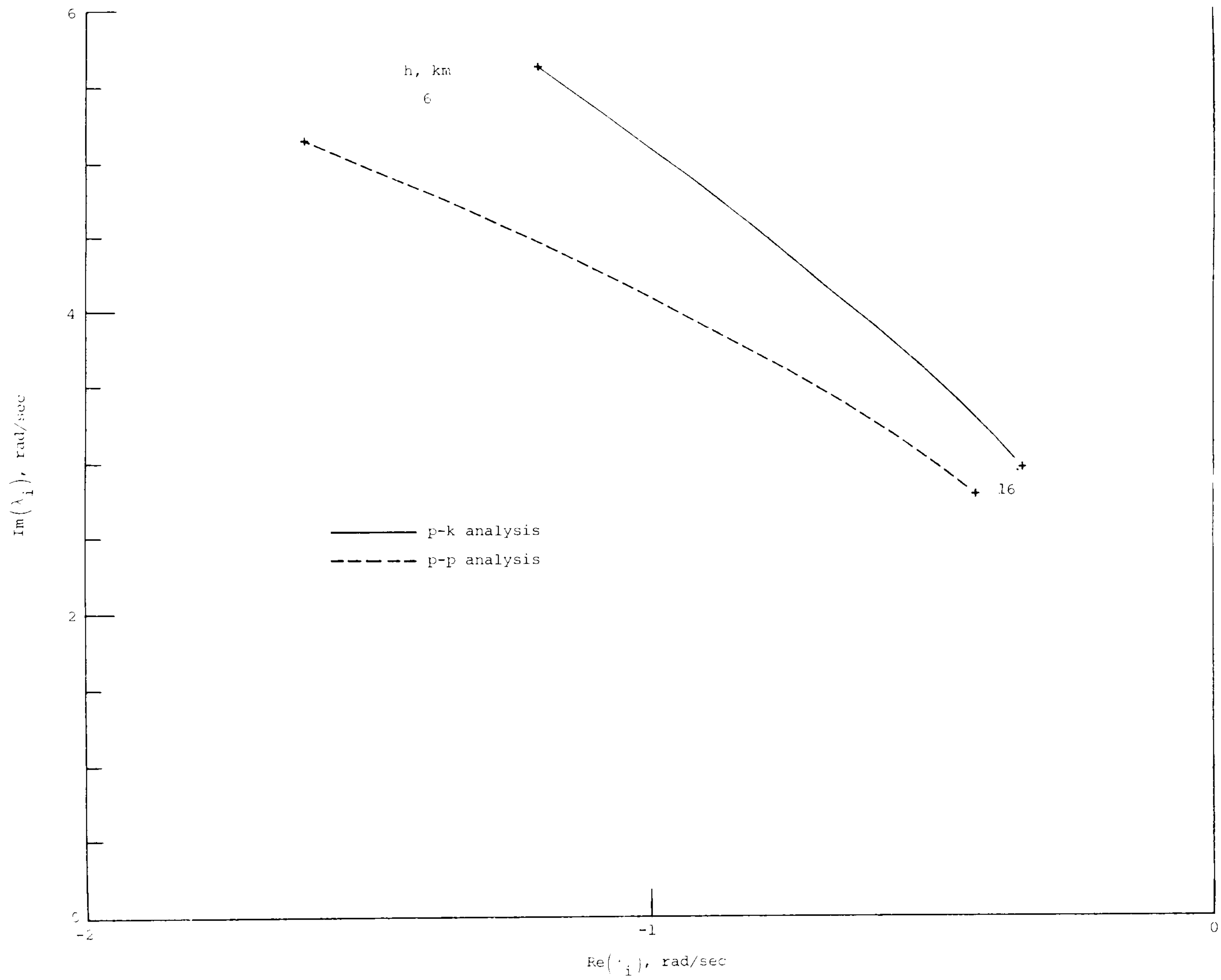


Figure 16.- Locus of short-period root with altitude. $N_{Ma} = 0.86$.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2165		

4. Title and Subtitle	5. Report Date
STABCAR - A PROGRAM FOR FINDING CHARACTERISTIC ROOTS OF SYSTEMS HAVING TRANSCENDENTAL STABILITY MATRICES	June 1984
	6. Performing Organization Code 505-34-33-05
7. Author(s) William M. Adams, Jr., Sherwood H. Tiffany, Jerry R. Newsom, and Ellwood L. Peele	8. Performing Organization Report No. L-14861
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665	10. Work Unit No.
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	13. Type of Report and Period Covered Technical Paper
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

STABCAR can be used to determine the characteristic roots of flexible, actively controlled aircraft, including the effects of unsteady aerodynamics. A modal formulation and a transfer-matrix representation of the control system are employed. Operable in either a batch or an interactive mode, STABCAR can provide graphical or tabular output of the variation of the roots with velocity, density, altitude, dynamic pressure or feedback gains. Herein the mathematical model, program structure, input requirements, output capabilities, and a series of sample cases are detailed. STABCAR was written for use on CDC® CYBER 175 equipment; modification would be required for operation on other machines.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
ISAC Flutter p-k method p-p method Active controls Stability Root loci	Unclassified - Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	143	A07

For sale by the National Technical Information Service, Springfield, Virginia 22161 NASA-Langley, 1984

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

1 2 1U,G, 840605 S00151DS
DEPT OF THE AIR FORCE
ARNOLD ENG DEVELOPMENT CENTER(AFSC)
ATTN: LIBRARY/DOCUMENTS
ARNOLD AF STA TN 37389

NASA

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return