N70-10614
NASA CR-73383

RE-369

FURTHER DEVELOPMENT OF AN ALGORITHM FOR THE NONLINEAR STABILITY ANALYSIS OF THE ORBITING ASTRONOMICAL OBSERVATORY "PAIRED-TRACKER" CONTROL SYSTEM

August 1969

Grumman
RESEARCH DEPARTMENT

GRUMMAN AIRCRAFT ENGINEERING CORPORATION
BETHPAGE NEW YORK

Grumman Research Department Report RE-369

FURTHER DEVELOPMENT OF AN ALGORITHM FOR THE NONLINEAR STABILITY ANALYSIS OF THE ORBITING ASTRONOMICAL OBSERVATORY "PAIRED-TRACKER" CONTROL SYSTEM

by

G. Geiss
V. Cohen
R. D'heedene
D. Rothschild

Systems Research Section

and

A. Chomas

Guidance and Control Section
Product Engineering Department

August 1969

Final Contract Report on Extension to NAS2-4063.

Approved by: Charles E. Mack, Jr.
Charles E. Mack, Jr.
Director of Research

ACKNOWLEDGMENTS

This work was supported by NASA Ames Research Center. The authors wish to thank Mr. Brian F. Doolin, Asst. Chief, Theoretical Guidance and Control Branch, for his general encouragement and specific help in obtaining the use of the IBM 360/95 computer at the NASA Institute for Space Studies. We also wish to thank Dr. David L. Stonehill of the Computer Applications Inc., who cooperated in every possible way, and Dr. George Meyer and Mr. Robert D. Showman of the NASA Ames Research Center, who provided valuable support and technical criticism.

ABSTRACT

This report describes the results of a study directed toward development of an effective algorithm for estimating the domain of attraction of the equilibrium state of the OAO "paired-Tracker" coarse pointing mode attitude control system via "optimal" quadratic form Liapunov functions. The algorithm is developed by formulating the estimation problem as a min-max problem which is solved by random search techniques.

The model of the system is reviewed, and several approximations to it are developed. A Popov type stability analysis is carried out for a simplified model to determine if it is absolutely stable and to then formulate a Luré type Liapunov function to be used in the estimation procedure. The results of the analysis were negative due to a pole at the origin of the linear part of the system and because the dominant nonlinearity was saturation.

The algorithm is tested on both the full nine dimensional model and a six dimensional approximation. The results for both are similar, but disappointing by comparison to simulation results. However, the estimates obtained are well into the region where the nonlinearities are dominant, and this is encouraging.

Recommendations for further research are made in the areas of: better search techniques for problems of high dimension, methods for determining the fundamental limitations of "optimal" quadratic estimation of the domain of attraction, and more direct methods of estimating the domain of attraction.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

I. INTRODUCTION

This report describes work performed under an extension to Contract NAS2-4063, "Nonlinear Analysis of the Orbiting Astronomical Observatory (OAO) Control System." It deals with the development of a numerical algorithm for estimating the domain of attraction of the equilibrium state of the NASA-Ames "paired-Tracker" coarse pointing mode attitude control system. This is the control system devised by Doolin and Showman (Refs. 1 and 2).

In our previous work (Ref. 3) we formulated the system equations in the required quasi-linear form and developed the algorithm as a min-max problem via LaSalle's theorem (Ref. 4) on the extent of asymptotic stability, and proved the feasibility of this approach. The computational approach was to use a gradient search routine (MIN-ALL) and the penalty function approach to solve the minimum problem. Since, in fact, the problem was to find a particular local minimum under a constraint and in the vicinity of the global (trivial, in this case) solution, a great deal of difficulty was encountered. At that time a random search was used to verify the results of the gradient search. It was much more effective than the gradient search, but very costly (it took some 26 minutes of IBM 360-75 time to solve the minimum problem, and this would have to be done repetitively to solve the maximum problem).

The goal of the present study, which is described in this report, is to make the feasible approach practical. We began by trying to improve the gradient search; however, this was abandoned as soon as major success was achieved in making the random search more efficient. The improved efficiency was accomplished by taking advantage of the known geometry of the problem and incorporating a one dimensional search for the constraint surface. This was effected by judicious choice of the probability distribution of the random points and by incorporating appropriate logic to speed the search.

As soon as the algorithm for solving the minimization problem was functioning satisfactorily, attention was focused on developing an effective random search for the much higher dimensional maximization problem. Here a "creeping accelerated random search" was employed with some success. The dimensionality and complexity of this problem precluded consideration of any search based on gradients.

Thus, to summarize, the objective of this study was to attempt to make a practical tool of an algorithm whose feasibility had been proven and to investigate its effectiveness on a complex real problem. In the following (Section II), we describe briefly the derivation of the system equations and some useful approximations to these equations. Some of the approximations enable the performance of theoretical analyses; some reduce the computational complexity of the problem.

In Section III we describe a frequency domain stability analysis, based on Popov's theory, of the approximate system model and indicate why it does not work for more complex models. The objective was to establish the absolute stability of the approximate model and to derive from this a Luré-Liapunov function that might be used in the algorithm to obtain a better estimate than the quadratic form provided. This is followed (Section IV) by the details of two numerical algorithms that were constructed and tested, along with the experimental results obtained for the complete model and one approximate model and some two dimensional test problems.

Section V details the difficulties encountered in the study and the shortcomings of our approach, along with the outlines of some problems whose solutions might contribute to improving the state of the art in estimating the domain of attraction of physical nonlinear systems. The summary (Section VI) places in capsule form the salient features of the study. Details of the computer programs used in the study are found in the appendices.

II. THE SYSTEM MODEL AND ITS APPROXIMATIONS

Review of Original Model

The basic block diagram of the system is given in Fig. 1.

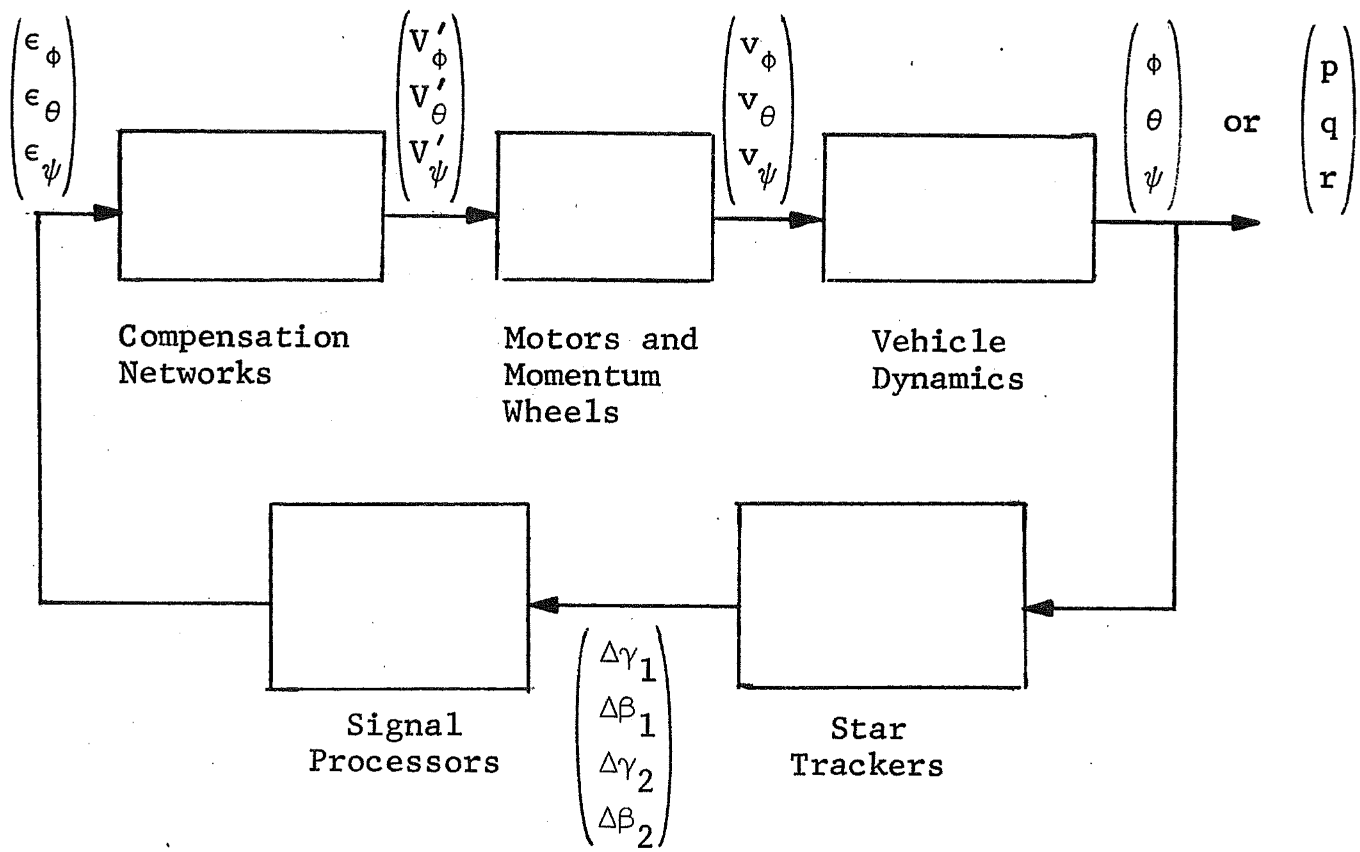


Fig. 1 Block Diagram of Basic Model

Star Tracker Model

The relationship of the inertial reference coordinates (x_r, y_r, z_r) and body coordinates (x_b, y_b, z_b) is given by a set of rotation transformations R_ϕ, R_θ, R_ψ,

$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = R_\psi R_\theta R_\phi \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} , \tag{1}$$

where the Euler angles ϕ, θ, ψ are, respectively, the roll, pitch, and yaw angles with respect to the reference coordinates, and R_ϕ, R_θ, R_ψ are given by

$$R_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{pmatrix} ,$$

$$R_\theta = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} , \tag{2}$$

and

$$R_\psi = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

For each Star Tracker, the relationship between the Tracker and Tracker reference coordinates is given by the rotation transformations $R_\alpha, R_\beta, R_\gamma$, viz.,

$$\begin{pmatrix} x_T \\ y_T \\ z_T \end{pmatrix} = R_\alpha R_\beta R_\gamma \begin{pmatrix} x_{TR} \\ y_{TR} \\ z_{TR} \end{pmatrix} , \tag{3}$$

where γ,β are the outer and inner gimbal angles, and the angle α is the rotation about the Tracker optical axis. In Ref. 3, two equations for each Star Tracker are derived by relating the actual and commanded values of β and γ with the corresponding error differences $\Delta\beta$ and $\Delta\gamma$. The equations are given below:

$$\Delta\beta_1 = \sin^{-1}(c\psi c\theta s\beta_{1c} + s\psi c\theta c\gamma_{1c} c\beta_{1c} + s\theta s\gamma_{1c} c\beta_{1c}) - \beta_{1c}$$

$$\Delta\gamma_1 = \tan^{-1}\left(\frac{-(s\psi s\phi + c\psi s\theta c\phi) s\beta_{1c} + (c\psi s\phi - s\psi s\theta c\phi) c\gamma_{1c} c\beta_{1c} + c\theta c\phi s\gamma_{1c} c\beta_{1c}}{-(s\psi c\phi - c\psi s\theta s\phi) s\beta_{1c} + (c\psi c\phi + s\psi s\theta s\phi) c\gamma_{1c} c\beta_{1c} - c\theta s\phi s\gamma_{1c} c\beta_{1c}}\right) - \gamma_{1c}$$

$$\Delta\beta_2 = \sin^{-1}(c\psi c\theta s\beta_{2c} - s\psi c\theta c\gamma_{2c} c\beta_{2c} - s\theta s\gamma_{2c} c\beta_{2c}) - \beta_{2c}$$

$$\Delta\gamma_2 = \tan^{-1}\left(\frac{(s\psi s\phi + c\psi s\theta c\phi) s\beta_{2c} + (c\psi s\phi - s\psi s\theta c\phi) c\gamma_{2c} c\beta_{2c} + c\theta c\phi s\gamma_{2c} c\beta_{2c}}{(s\psi c\phi - c\psi s\theta s\phi) s\beta_{2c} + (c\psi c\phi + s\psi s\theta s\phi) c\gamma_{2c} c\beta_{2c} - c\theta s\phi s\gamma_{2c} c\beta_{2c}}\right) - \gamma_{2c} \tag{4}$$

$$\Delta\beta_3 = \sin^{-1}(c\psi c\theta s\beta_{3c} + s\psi c\theta s\gamma_{3c} c\beta_{3c} - s\theta c\gamma_{3c} c\beta_{3c}) - \beta_{3c}$$

$$\Delta\gamma_3 = \tan^{-1}\left(\frac{-(s\psi c\phi - c\psi s\theta s\phi) s\beta_{3c} + (c\psi c\phi + s\psi s\theta s\phi) s\gamma_{3c} c\beta_{3c} + c\theta s\phi c\gamma_{3c} c\beta_{3c}}{(s\psi s\phi + c\psi s\theta c\phi) s\beta_{3c} - (c\psi s\phi - s\psi s\theta c\phi) s\gamma_{3c} c\beta_{3c} + c\theta c\phi c\gamma_{3c} c\beta_{3c}}\right) - \gamma_{3c}$$

$$\Delta\beta_4 = \sin^{-1}(c\psi c\theta s\beta_{4c} - s\psi c\theta s\gamma_{4c} c\beta_{4c} + s\theta c\gamma_{4c} c\beta_{4c}) - \beta_{4c}$$

$$\Delta\gamma_4 = \tan^{-1}\left(\frac{+(s\psi c\phi - c\psi s\theta s\phi) s\beta_{4c} + (c\psi c\phi + s\psi s\theta s\phi) s\gamma_{4c} c\beta_{4c} + c\theta s\phi c\gamma_{4c} c\beta_{4c}}{-(s\psi s\phi + c\psi s\theta c\phi) s\beta_{4c} - (c\psi s\phi - s\psi s\theta c\phi) s\gamma_{4c} c\beta_{4c} + c\theta c\phi c\gamma_{4c} c\beta_{4c}}\right) - \gamma_{4c}$$

Error Processor and Actuator

The equations for the "partial processor" used in our model for Trackers 1 and 2 are

$$\begin{pmatrix} \epsilon_\phi \\ \epsilon_\theta \\ \epsilon_\psi \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ d_{12}c(\gamma_{2c} + \Delta\gamma_2) & 0 & d_{12}c(\gamma_{1c} + \Delta\gamma_1) \\ -d_{12}s(\gamma_{2c} + \Delta\gamma_2) & 0 & -d_{12}s(\gamma_{1c} + \Delta\gamma_1) \end{pmatrix} \begin{pmatrix} \Delta\beta_1 \\ \Delta\gamma_1 \\ \Delta\beta_2 \end{pmatrix} . \tag{5}$$

Each of these signals passes through a lead-lag compensation network, thereby producing the set of typical equations:

$$\dot{\omega}_\phi + \frac{1}{\tau_2}\,\omega_\phi = -\,K_c\,\frac{\tau_1}{\tau_2^2}\,\epsilon_\phi$$

$$V'_\phi = \omega_\phi + K_c\left(1 + \frac{\tau_1}{\tau_2}\right)\epsilon_\phi \qquad (6)$$

with a similar set of equations for θ and ψ. Here V' denotes the output voltage of the compensation network.

The motor saturation is represented as

$$V'' = f(V') \ , \qquad (7)$$

where

$$f(V') = \begin{cases} 26 \ , & V' > 26 \text{ volts} \\ V' \ , & |V'| \leq 26 \text{ volts} \\ -26 \ , & V' < -26 \text{ volts} \ . \end{cases} \qquad (8)$$

Incorporating the dynamic equations for the motors, momentum wheels, and vehicle in state variable form, and neglecting gyroscopic torques due to the momentum wheels, the state equations for the momentum wheels and vehicle reduce to

$$\left.\begin{aligned}
\dot{v}_\phi + \frac{1}{\tau_m} v_\phi &= \frac{K_m}{\tau_m} V''_\phi \\
p &= -\frac{1}{I} v_\phi + \left(p(0) + \frac{1}{I} v_\phi(0)\right) \\
\dot{v}_\theta + \frac{1}{\tau_m} v_\theta &= \frac{K_m}{\tau_m} V''_o \\
q &= -\frac{1}{I} v_\theta + \left(q(0) + \frac{1}{I} v_\theta(0)\right) \\
\dot{v}_\psi + \frac{1}{\tau_m} v_\psi &= \frac{K_m}{\tau_m} V''_\psi \\
r &= -\frac{1}{I} v_\psi + \left(r(0) + \frac{1}{I} v_\psi(0)\right)
\end{aligned}\right\} , \qquad (9)$$

where p,q,r are the rotational rates about the body axes x_b, y_b, z_b and v_ϕ, v_θ, v_ψ are the wheel momentum variables. The equations for the Euler angles ϕ, θ, ψ are

$$\left.\begin{aligned}
\dot{\phi} &= p + (t\theta s\phi)q + (t\theta c\phi)r \\
\dot{\theta} &= (c\phi)q - (s\phi)r \\
\ddot{\psi} &= \frac{s\phi}{c\theta} q + \frac{c\phi}{c\theta} r
\end{aligned}\right\} \qquad (10)$$

The equations presented thus far are summarized in the block diagram of Fig. 2.

Reformulation of the State Equations

For the numerical portion of our stability analysis, we require the system state equations to take the form

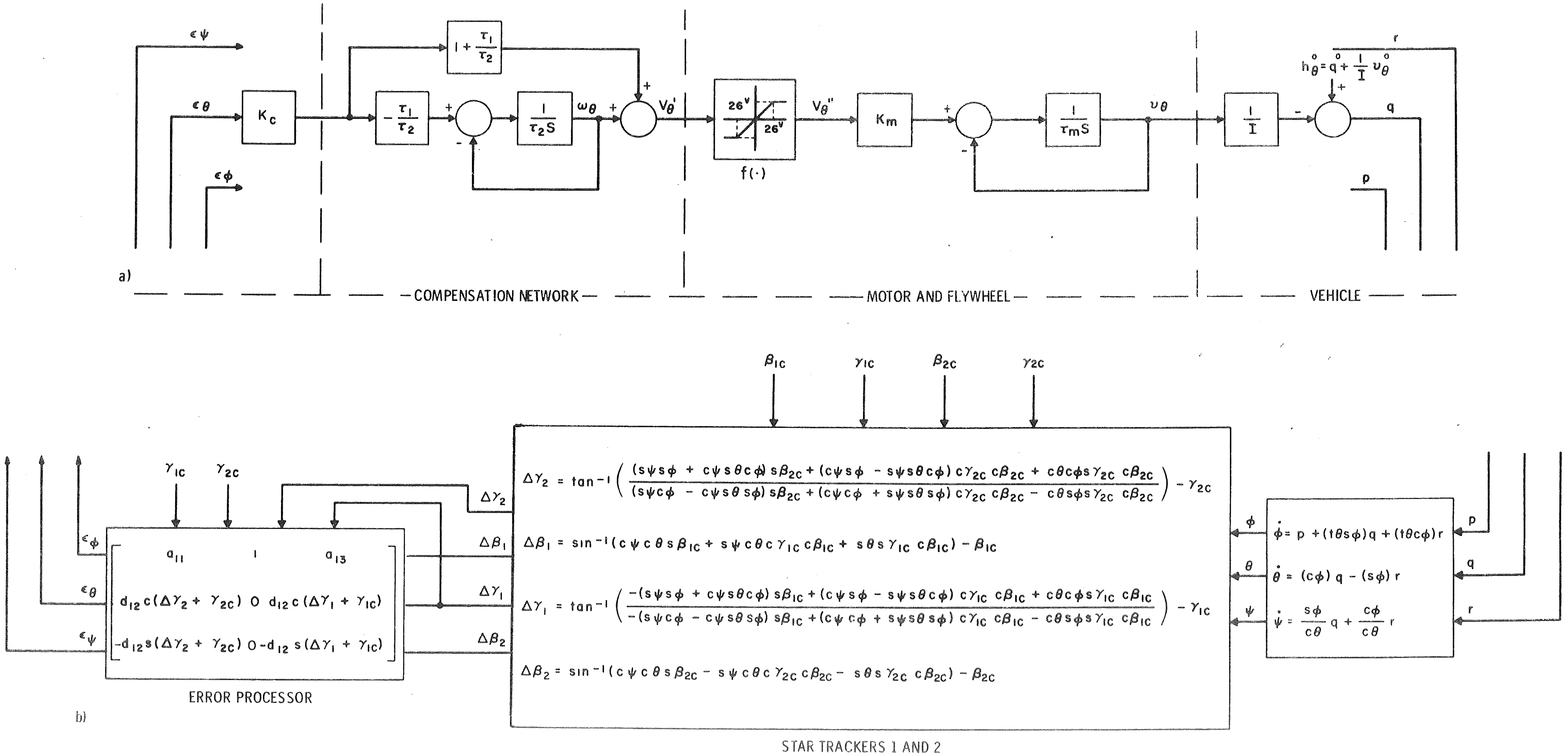


Fig. 2 Typical Forward Channel [a) Without Wheel Gyroscopic Torques] and Feedback Path [b) Based on Gimbal Angle Rate Equations]

$$\dot{x} = Ax + f(x) \; , \tag{11}$$

where x is the state vector of appropriate dimension, A is the matrix of the linear part, and $f(x)$ is the collection of non-linear terms that have no linear part, i.e.

$$\lim_{\|x\| \to 0} \frac{\|f(x)\|}{\|x\|} = 0 \; . \tag{12}$$

First we define a set of variables whose value at equilibrium $(\dot{x} = 0)$ will be zero, viz.,

$$\left.\begin{aligned}
\phi' &= \phi - \phi_e \\
\theta' &= \theta - \theta_e \\
\psi' &= \psi - \psi_e \\
v'_\phi &= v_\phi - Ih^o_\phi \\
v'_\theta &= v_\theta - Ih^o_\theta \\
v'_\psi &= v_\psi - Ih^o_\psi \\
\omega'_\phi &= \omega_\phi - \left(- \frac{\tau_1 I}{\tau_2 K_m} \cdot h^o_\phi\right) \\
\omega'_\theta &= \omega_\theta - \left(- \frac{\tau_1 I}{\tau_2 K_m} h^o_\theta\right) \\
\omega'_\psi &= \omega_\psi - \left(- \frac{\tau_1 I}{\tau_2 K_m} h^o_\psi\right)
\end{aligned}\right\} \; , \tag{13}$$

where

$$h_\phi^o = p(0) + \frac{1}{I} v_\phi(0)$$

$$h_\theta^o = q(0) + \frac{1}{I} v_\theta(0) \tag{14}$$

$$h_\psi^o = r(0) + \frac{1}{I} v_\psi(0) ,$$

and ϕ_e, θ_e, ψ_e are the offset angles that are complicated functions of the initial angular momentum $Ih_\phi^o, Ih_\theta^o, Ih_\psi^o$ and the commanded gimbal angles.

The equations using this newly-defined set of state variables become

$$\dot{\phi}' = - \frac{1}{I} v_\phi' - \frac{(t\theta s\phi)}{I} v_\theta' - \frac{(t\theta c\phi)}{I} v_\psi'$$

$$\dot{v}_\phi' = - \frac{1}{\tau_m} v_\phi' + \frac{K_m}{\tau_m} f \left(K_c \left(1 + \frac{\tau_1}{\tau_2}\right) \epsilon_\phi' + \omega_\phi' + \frac{I}{K_m} h_\phi^o \right) - \frac{I}{\tau_m} h_\phi^o$$

$$\dot{\omega}_\phi' = - \frac{1}{\tau_m} \omega_\phi' - \frac{K_c \tau_1}{\tau_2^2} \epsilon_\phi'$$

(15)

$$\dot{\theta}' = - \frac{(c\phi)}{I} v_\theta' + \frac{(s\phi)}{I} v_\psi'$$

$$\dot{v}_\theta' = - \frac{1}{\tau_m} v_\theta' + \frac{K_m}{\tau_m} f \left(K_c \left(1 + \frac{\tau_1}{\tau_2}\right) \epsilon_\theta' + \omega_\theta' + \frac{I}{K_m} h_\theta^o \right) - \frac{I}{\tau_m} h_\theta^o$$

$$\dot{\omega}_\theta' = - \frac{1}{\tau_2} \omega_\theta' - \frac{K_c \tau_1}{\tau_2^2} \epsilon_\theta'$$

$$\dot{\psi}' = -\frac{s\phi}{Ic\theta} v'_\theta - \frac{c\phi}{Ic\theta} v'_\psi$$

$$\dot{v}'_\psi = -\frac{1}{\tau_m} v'_\psi + \frac{K_m}{\tau_m} f \left(K_c \left(1 + \frac{\tau_1}{\tau_2}\right) \epsilon'_\psi + \omega'_\psi + \frac{I}{K_m} h^o_\psi \right) - \frac{I}{\tau_m} h^o_\psi$$

$$\dot{\omega}'_\psi = -\frac{1}{\tau_2} \omega'_\psi - \frac{K_c \tau_1}{\tau_2^2} \epsilon'_\psi$$

$$\epsilon'_\phi = a_{11}\Delta\beta_1 + \Delta\gamma_1 + a_{13}\Delta\beta_2 - \frac{I}{K_c K_m} h^o_\phi$$

$$\epsilon'_\theta = d_{12}c(\gamma_{2c} + \Delta\gamma_2) \cdot \Delta\beta_1 + d_{12}c(\gamma_{1c} + \Delta\gamma_1) \cdot \Delta\beta_2 - \frac{I}{K_c K_m} h^o_\theta \qquad \text{(15) (Cont.)}$$

$$\epsilon'_\psi = -d_{12}s(\gamma_{2c} + \Delta\gamma_2) \cdot \Delta\beta_1 - d_{12}s(\gamma_{1c} + \Delta\gamma_1) \cdot \Delta\beta_2 - \frac{I}{K_c K_m} h^o_\psi$$

$$\Delta\beta_1 = \sin^{-1}(c\psi c\theta s\beta_{1c} + s\psi c\theta c\gamma_{1c} c\beta_{1c} + s\theta s\gamma_{1c} c\beta_{1c}) - \beta_{1c}$$

$$\Delta\gamma_1 = \tan^{-1} \frac{1}{D_1} \Big(-(s\psi s\phi + c\psi s\theta c\phi) s\beta_{1c} + (c\psi s\phi - s\psi s\theta c\phi) c\gamma_{1c} c\beta_{1c}$$

$$+ c\theta c\phi s\gamma_{1c} c\beta_{1c} \Big) - \gamma_{1c}$$

$$D_1 = \Big(-(s\psi c\phi - c\psi s\theta s\phi) s\beta_{1c} + (c\psi c\phi + s\psi s\theta s\phi) c\gamma_{1c} c\beta_{1c}$$

$$- c\theta s\phi s\gamma_{1c} c\beta_{1c} \Big)$$

$$\Delta\beta_2 = \sin^{-1}(c\psi c\theta s\beta_{2c} - s\psi c\theta c\gamma_{2c} c\beta_{2c} - s\theta s\gamma_{2c} c\beta_{2c}) - \beta_{2c}$$

$$\Delta\gamma_2 = \tan^{-1} \frac{1}{D_2} \Big((s\psi s\phi + c\psi s\theta c\phi) s\beta_{2c} + (c\psi s\phi - s\psi s\theta c\phi) c\gamma_{2c} c\beta_{2c}$$

$$+ c\theta c\phi s\gamma_{2c} c\beta_{2c} \Big) - \gamma_{2c}$$

$$D_2 = \Big((s\psi c\phi - c\psi s\theta s\phi) s\beta_{2c} + (c\psi c\phi + s\psi s\theta s\phi) c\gamma_{2c} c\beta_{2c}$$

$$- c\theta s\phi s\gamma_{2c} c\beta_{2c} \Big)$$

The details of putting these equations into the required form, Eq. (11), are given in Ref. 5, and the general form of the result is illustrated in Fig. 3. The A_{ij} are complicated functions of the commanded gimbal angles and the initial total momenta, and these too are tabulated in Ref. 5. Their linearizations about $\phi_e = \theta_e = \psi_e = 0$ are given in Ref. 3.

If the effects of nonzero equilibrium are assumed to be negligibly small, then the equations of Fig. 3 become those of Fig. 4, where the variables v', ω' have been rescaled to be dimensionless, i.e.,

$$v''_\theta = \frac{1}{K_m K_c} v'_\theta \text{ , etc.} \quad , \quad \omega''_\theta = \frac{\tau_2}{K_c \tau_1} \omega'_\theta \text{ , etc.} \tag{16}$$

Note that nonzero initial momenta, $Ih^o_\phi, Ih^o_\theta, Ih^o_\psi$, have the effect of displacing the linear regions of the corresponding saturation terms for f_2, f_5, and f_7.

Effects of Offset on Linear Models

In the models used for our study, the assumption of zero angular offset at equilibrium was made in determining the

$$
\begin{bmatrix} \dot{\phi}' \\ \dot{\upsilon}_\phi' \\ \dot{\omega}_\phi' \\ \dot{\theta}' \\ \dot{\upsilon}_\theta' \\ \dot{\omega}_\theta' \\ \dot{\psi}' \\ \dot{\upsilon}_\psi' \\ \dot{\omega}_\psi' \end{bmatrix}
=
\begin{bmatrix}
0 & -\frac{1}{I} & 0 & 0 & A_{15} & 0 & 0 & A_{18} & 0 \\
+\frac{K_m K_c}{\tau_m}\left(1+\frac{\tau_1}{\tau_2}\right) & -\frac{1}{\tau_m} & +\frac{K_m}{\tau_m} & A_{24} & 0 & 0 & A_{27} & 0 & 0 \\
-K_c\frac{\tau_1}{\tau_2^2} & 0 & -\frac{1}{\tau_2} & A_{34} & 0 & 0 & A_{37} & 0 & 0 \\
0 & 0 & 0 & 0 & A_{45} & 0 & 0 & A_{48} & 0 \\
A_{51} & 0 & 0 & A_{54} & -\frac{1}{\tau_m} & +\frac{K_m}{\tau_m} & A_{57} & 0 & 0 \\
A_{61} & 0 & 0 & A_{64} & 0 & -\frac{1}{\tau_2} & A_{67} & 0 & 0 \\
0 & 0 & 0 & 0 & A_{75} & 0 & 0 & A_{78} & 0 \\
A_{81} & 0 & 0 & A_{84} & 0 & 0 & A_{87} & -\frac{1}{\tau_m} & +\frac{K_m}{\tau_m} \\
A_{91} & 0 & 0 & A_{94} & 0 & 0 & A_{97} & 0 & -\frac{1}{\tau_2}
\end{bmatrix}
\cdot
\begin{bmatrix} \phi' \\ \upsilon_\phi' \\ \omega_\phi' \\ \theta' \\ \upsilon_\theta' \\ \omega_\theta' \\ \psi' \\ \upsilon_\psi' \\ \omega_\psi' \end{bmatrix}
+
\begin{bmatrix}
-t(\theta'+\theta_e)\left[\frac{1}{I}\upsilon_\theta' s(\phi'+\phi_e) + \frac{1}{I}\upsilon_\psi' c(\phi'+\phi_e)\right] - A_{15}\upsilon_\theta - A_{18}\upsilon_\psi' \\
+\frac{K_m}{\tau_m} f\left|K_c\left(1+\frac{\tau_1}{\tau_2}\right)\left(a_{11}\Delta\beta_1 + \Delta\gamma_1 + a_{13}\Delta\beta_2\right) + \omega_\phi' - \frac{\tau_1 I}{\tau_2 K_m} h_\phi^\circ\right| \\
\quad -\frac{K_m}{\tau_m}K_c\left(1+\frac{\tau_1}{\tau_2}\right)\phi' - A_{24}\theta' - A_{27}\psi' - \frac{K_m}{\tau_m}\omega_\phi' - \frac{I}{\tau_m}h_\phi^\circ \\
-K_c\frac{\tau_1}{\tau_2^2}\left[\left(a_{11}\Delta\beta_1 + \Delta\gamma_1 + a_{13}\Delta\beta_2\right) - \phi'\right] - A_{34}\theta' - A_{37}\psi' + \frac{I\tau_1}{K_m\tau_2^2}h_\phi^\circ \\
-\frac{1}{I}\left[\upsilon_\theta'\left(c(\phi'+\phi_e)-1\right) + \upsilon_\psi' s(\phi'+\phi_e)\right] - A_{45}\upsilon_\theta' - A_{48}\upsilon_\psi' \\
+\frac{K_m}{\tau_m} f\left|K_c d_{12}\left(1+\frac{\tau_1}{\tau_2}\right)\left(c(\Delta\gamma_2+\gamma_{2c})\cdot\Delta\beta_1 + c(\Delta\gamma_1+\gamma_{1c})\cdot\Delta\beta_2\right) + \omega_\theta' - \frac{\tau_1 I}{\tau_2 K_m}h_\theta^\circ\right| \\
\quad -A_{51}\phi' - A_{54}\theta' - A_{57}\psi' - \frac{K_m}{\tau_m}\omega_\theta' - \frac{I}{\tau_m}h_\theta^\circ \\
-K_c\frac{\tau_1}{\tau_2^2}d_{12}\left(c(\Delta\gamma_2+\gamma_{2c})\cdot\Delta\beta_1 + c(\Delta\gamma_1+\gamma_{1c})\cdot\Delta\beta_2\right) \\
\quad -A_{61}\phi' - A_{64}\theta' - A_{67}\psi' + \frac{I\tau_1}{K_m\tau_2^2}h_\theta^\circ \\
-\frac{1}{Ic(\theta'+\theta_e)}\left[\upsilon_\theta' s(\phi'+\phi_e) + \upsilon_\psi'\left(c(\phi'+\phi_e)-1\right)\right] - A_{75}\upsilon_\theta' - A_{78}\upsilon_\psi' \\
+\frac{K_m}{\tau_m} f\left|-K_c\left(1+\frac{\tau_1}{\tau_2}\right)d_{12}\left(s(\Delta\gamma_2+\gamma_{2c})\cdot\Delta\beta_1 + s(\Delta\gamma_1+\gamma_{1c})\cdot\Delta\beta_2\right) + \omega_\psi' - \frac{\tau_1 I}{\tau_2 K_m}h_\psi^\circ\right| \\
\quad -A_{81}\phi' - A_{84}\theta' - A_{87}\psi' - \frac{K_m}{\tau_m}\omega_\psi' - \frac{I}{\tau_m}h_\psi^\circ \\
K_c\frac{\tau_1}{\tau_2^2}d_{12}\left(s(\Delta\gamma_2+\gamma_{2c})\cdot\Delta\beta_1 + s(\Delta\gamma_1+\gamma_{1c})\Delta\beta_2\right) \\
\quad -A_{91}\phi' - A_{94}\theta' - A_{97}\psi' + \frac{I\tau_1}{K_m\tau_2^2}h_\psi^\circ
\end{bmatrix}
$$

EQUATIONS LINEARIZED ABOUT

$\phi' = \theta' = \psi' = 0$ $\qquad$ $\theta' = \theta - \theta_e$, etc.

$\upsilon_\phi' = \upsilon_\theta' = \upsilon_\psi' = 0$ $\qquad$ $\upsilon_\theta' = \upsilon_\theta - I h_\theta^\circ$, etc.

$\omega_\phi' = \omega_\theta' = \omega_\psi' = 0$ $\qquad$ $\omega_\theta' = \omega_\theta + \frac{\tau_1 I}{\tau_2 K_m}h_\theta^\circ$, etc.

$\Delta\beta_1 = \sin^{-1}(c\psi c\theta s\beta_{1c} + s\psi c\theta c\gamma_{1c} c\beta_{1c} + s\theta s\gamma_{1c} c\beta_{1c}) - \beta_{1c}$, $\quad \Delta\beta_2 = \sin^{-1}(+ \quad - \quad - \quad) - \beta_{2c}$

$\Delta\gamma_1 = \tan^{-1}\left(\frac{-(s\psi s\phi + c\psi s\theta c\phi) s\beta_{1c} + (c\psi s\phi - s\psi s\theta c\phi) c\gamma_{1c} c\beta_{1c} + c\theta c\phi s\gamma_{1c} c\beta_{1c}}{-(s\psi c\phi - c\psi s\theta s\phi) s\beta_{1c} + (c\psi c\phi + s\psi s\theta s\phi) c\gamma_{1c} c\beta_{1c} - c\theta s\phi s\gamma_{1c} c\beta_{1c}}\right) - \gamma_{1c}$, $\quad \Delta\gamma_2 = \tan^{-1}\left(\frac{+ \quad + \quad +}{+ \quad + \quad -}\right) - \gamma_{2c}$

Fig. 3 State Equations Based on Tracker Angle Model

$$
\begin{pmatrix} \dot{\phi}' \\ \dot{\upsilon}''_{\phi} \\ \dot{\omega}''_{\phi} \\ \dot{\theta}' \\ \dot{\upsilon}''_{\theta} \\ \dot{\omega}''_{\theta} \\ \dot{\psi}' \\ \dot{\upsilon}''_{\psi} \\ \dot{\omega}''_{\psi} \end{pmatrix}
=
\left[\begin{array}{ccc|ccc|ccc}
0 & -\frac{K_m K_c}{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{(\tau_1+\tau_2)}{\tau_m \tau_2} & -\frac{1}{\tau_m} & \frac{\tau_1}{\tau_m \tau_2} & \begin{gathered}(a_{11}\, s\gamma_{1c} \\ -i\beta_{1c}\, c\gamma_{1c} \\ -a_{13}\, s\gamma_{2c}) \\ \cdot\frac{\tau_1+\tau_2}{\tau_m \tau_2}\end{gathered} & 0 & 0 & \begin{gathered}(a_{11}\, c\gamma_{1c} \\ +i\beta_{1c}\, s\gamma_{1c} \\ -a_{13}\, c\gamma_{2c}) \\ \cdot\frac{\tau_1+\tau_2}{\tau_m \tau_2}\end{gathered} & 0 & 0 \\
-\frac{1}{\tau_2} & 0 & -\frac{1}{\tau_2} & \begin{gathered}-(a_{11}\, s\gamma_{1c} \\ -i\beta_{1c}\, c\gamma_{1c} \\ -a_{13}\, s\gamma_{2c}) \\ \cdot\frac{1}{\tau_2}\end{gathered} & 0 & 0 & \begin{gathered}-(a_{11}\, c\gamma_{1c} \\ +i\beta_{1c}\, s\gamma_{1c} \\ -a_{13}\, c\gamma_{2c}) \\ \cdot\frac{1}{\tau_2}\end{gathered} & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & -\frac{K_m K_c}{I} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \begin{gathered}\frac{\tau_1+\tau_2}{\tau_m \tau_2} \\ \cdot d_{12}\, s(\gamma_{1c}-\gamma_{2c})\end{gathered} & -\frac{1}{\tau_m} & \frac{\tau_1}{\tau_m \tau_2} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{d_{12}}{\tau_2}\, s(\gamma_{1c}-\gamma_{2c}) & 0 & -\frac{1}{\tau_2} & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_m K_c}{I} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \begin{gathered}\frac{\tau_1+\tau_2}{\tau_m \tau_2} \\ \cdot d_{12}\, s(\gamma_{1c}-\gamma_{2c})\end{gathered} & -\frac{1}{\tau_m} & \frac{\tau_1}{\tau_m \tau_2} \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{d_{12}}{\tau_2}\, s(\gamma_{1c}-\gamma_{2c}) & 0 & -\frac{1}{\tau_2}
\end{array}\right]
\begin{pmatrix} \phi' \\ \upsilon''_{\phi} \\ \omega''_{\phi} \\ \theta' \\ \upsilon''_{\theta} \\ \omega''_{\theta} \\ \psi' \\ \upsilon''_{\psi} \\ \omega''_{\psi} \end{pmatrix}
+
\left[\begin{array}{c}
-i\theta\left[\frac{K_m K_c}{I}\,\upsilon''_{\theta}\, s\phi + \frac{K_m K_c}{I}\,\upsilon''_{\psi}\, c\phi\right] \\
\hline
\begin{gathered}
\frac{1}{K_c \tau_m}\, f\left\{K_c\left(1+\frac{\tau_1}{\tau_2}\right)\left(a_{11}\,\Delta\beta_1 + \Delta\gamma_1 + a_{13}\,\Delta\beta_2\right) + \frac{K_c \tau_1}{\tau_2}\,\omega''_{\phi} - \frac{\tau_1 I}{\tau_2 K_m}\, h^{\circ}_{\phi}\right\} \\
-\frac{1}{K_c \tau_m}\left[K_c\left(1+\frac{\tau_1}{\tau_2}\right)\left(\phi' + (a_{11}\, s\gamma_{1c} - i\beta_{1c}\, c\gamma_{1c} - a_{13}\, s\gamma_{2c})\,\theta'\right.\right. \\
\left.\left. + (a_{11}\, c\gamma_{1c} + i\beta_{1c}\, s\gamma_{1c} - a_{13}\, c\gamma_{2c})\,\psi'\right)\right] - \frac{\tau_1}{\tau_m \tau_2}\,\omega''_{\phi} - \frac{I}{K_m K_c \tau_m}\, h^{\circ}_{\phi}
\end{gathered} \\
\hline
\begin{gathered}
-\frac{1}{\tau_2}\left[\left(a_{11}\,\Delta\beta_1 + \Delta\gamma_1 + a_{13}\,\Delta\beta_2\right) - \left(\phi' + (a_{11}\, s\gamma_{1c} - i\beta_{1c}\, c\gamma_{1c} - a_{13}\, s\gamma_{2c})\,\theta'\right.\right. \\
\left.\left. + (a_{11}\, c\gamma_{1c} + i\beta_{1c}\, s\gamma_{1c} - a_{13}\, c\gamma_{2c})\,\psi'\right)\right] + \frac{I}{\tau_2 K_c K_m}\, h^{\circ}_{\phi}
\end{gathered} \\
\hline
-\left[\frac{K_m K_c}{I}\,\upsilon''_{\theta}\,(c\phi - 1) - \frac{K_m K_c}{I}\,\upsilon''_{\psi}\, s\phi\right] \\
\hline
\begin{gathered}
\frac{1}{K_c \tau_m}\, f\left\{K_c d_{12}\left(1+\frac{\tau_1}{\tau_2}\right)\left(c(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1 + c(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\right) + \frac{K_c \tau_1}{\tau_2}\,\omega''_{\theta} - \frac{\tau_1 I}{\tau_2 K_m}\, h^{\circ}_{\theta}\right\} \\
-\frac{1}{K_c \tau_m}\left[K_c d_{12}\left(1+\frac{\tau_1}{\tau_2}\right)\left(s(\gamma_{1c} - \gamma_{2c})\right)\theta + \frac{K_c \tau_1}{\tau_2}\,\omega''_{\theta}\right] - \frac{I}{K_m K_c \tau_m}\, h^{\circ}_{\theta}
\end{gathered} \\
\hline
\begin{gathered}
-\frac{d_{12}}{\tau_2}\left(c(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1 + c(\Delta\gamma_1 + \gamma_{1c})\,\Delta\beta_2\right) \\
+\frac{d_{12}}{\tau_2}\left(s(\gamma_{1c} - \gamma_{2c})\right)\theta' + \frac{I}{\tau_2 K_c K_m}\, h^{\circ}_{\theta}
\end{gathered} \\
\hline
-\frac{1}{I c\theta}\left[K_m K_c\,\upsilon''_{\theta}\, s\phi + K_m K_c\,\upsilon''_{\psi}\,(c\phi - c\theta)\right] \\
\hline
\begin{gathered}
\frac{1}{K_c \tau_m}\, f\left\{-K_c\left(1+\frac{\tau_1}{\tau_2}\right) d_{12}\left(s(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1 + s(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\right) + \frac{K_c \tau_1}{\tau_2}\,\omega''_{\psi} - \frac{\tau_1 I}{\tau_2 K_m}\, h^{\circ}_{\psi}\right\} \\
-\frac{1}{K_c \tau_m}\left[+K_c\left(1+\frac{\tau_1}{\tau_2}\right) d_{12}\, s(\gamma_{1c} - \gamma_{2c})\,\psi' + \frac{K_c \tau_1}{\tau_2}\,\omega''_{\psi}\right] - \frac{I}{K_m K_c \tau_m}\, h^{\circ}_{\psi}
\end{gathered} \\
\hline
\begin{gathered}
\frac{d_{12}}{\tau_2}\left(s(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1 + s(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\right) \\
+\frac{d_{12}}{\tau_2}\, s(\gamma_{1c} - \gamma_{2c})\,\psi' + \frac{I}{\tau_2 K_c K_m}\, h^{\circ}_{\psi}
\end{gathered}
\end{array}\right]
$$

EQUATIONS LINEARIZED ABOUT

$$\phi' = \theta' = \psi' = 0$$
$$\upsilon''_{\phi} = \upsilon''_{\theta} = \upsilon''_{\psi} = 0$$
$$\omega''_{\phi} = \omega''_{\theta} = \omega''_{\psi} = 0$$

$\theta' = \theta - \theta_e$, etc.

$$\upsilon''_{\theta} = \frac{1}{K_m K_c}\,\upsilon'_{\theta} = \frac{1}{K_m K_c}\,(\upsilon_{\theta} - I h^{\circ}_{\theta}), \text{ etc.}$$

$$\omega''_{\theta} = \frac{\tau_2}{K_c \tau_1}\,\omega'_{\theta} = \frac{\tau_2}{K_c \tau_1}\,\left(\omega_{\theta} + \frac{\tau_1 I}{\tau_2 K_m}\, h^{\circ}_{\theta}\right), \text{ etc.}$$

$$\text{sgn } d_{12} = \text{sgn}\,(\gamma_{1c} - \gamma_{2c})$$

$$\Delta\beta_1 = \sin^{-1}(c\psi\, c\theta\, s\beta_{1c} + s\psi\, c\theta\, c\gamma_{1c}\, c\beta_{1c} + s\theta\, s\gamma_{1c}\, c\beta_{1c}) - \beta_{1c} \quad , \Delta\beta_2 = \sin^{-1}(+ \quad - \quad - \quad) - \beta_{2c}$$

$$\Delta\gamma_1 = \tan^{-1}\left(\frac{-(s\psi\, s\phi + c\psi\, s\theta\, c\phi)\, s\beta_{1c} + (c\psi\, s\phi - s\psi\, s\theta\, c\phi)\, c\gamma_{1c}\, c\beta_{1c} + c\theta\, c\phi\, s\gamma_{1c}\, c\beta_{1c}}{-(s\psi\, c\phi - c\psi\, s\theta\, s\phi)\, s\beta_{1c} + (c\psi\, c\phi + s\psi\, s\theta\, s\phi)\, c\gamma_{1c}\, c\beta_{1c} - c\theta\, s\phi\, s\gamma_{1c}\, c\beta_{1c}}\right) - \gamma_{1c}, \quad \Delta\gamma_2 = \tan^{-1}\left(\frac{+ \quad + \quad +}{+ \quad + \quad -}\right) - \gamma_{2c}$$

Fig. 4 Nondimensional State Equations Based on Tracker Angle Model—Offset Neglected linearized part of the matrix differential equations. To be exact, however, the state equations should be linearized about the true equilibrium values of the roll, pitch, and yaw angles.

A computer program was written for the IBM 360/75 to compare eigenvalues of the matrices of the linear part which were derived by assuming both zero and nonzero offset, for various values of commanded gimbal angles and initial momenta. Results have shown that for a range of commanded gimbal angles and initial total momenta well beyond those expected, that the differences in both real and imaginary parts of the calculated eigenvalues occur in the fourth significant figure for the actual OAO system (Ref. 6) and in the fifth figure for the "paired-Tracker" system.

System Model with Motor Saturation Only

An analysis of the simplified system where the only nonlinearities considered are those of motor saturation has been carried out literally because the numerical computation was too sensitive. The simplified system is represented as

$$\dot{x} = Ax + Gf(v) \quad , \tag{17}$$

where G is a 9×3 matrix, $f(v)$ is a three vector of saturation functions obtained from $f(x)$ by deleting all nonlinear terms except saturation, and v is a three dimensional vector. These terms are:

$$
A = \begin{bmatrix}
0 & -\frac{K_m K_c}{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{1}{\tau_m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{\tau_2} & 0 & -\frac{1}{\tau_2} & 2t\beta_{1c}c\gamma_{1c} & 0 & 0 & -2t\beta_{1c}s\gamma_{1c} & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{K_m K_c}{I} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{1}{\tau_m} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{\tau_2} & 0 & -\frac{1}{\tau_2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_m K_c}{I} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_m} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_2} & 0 & -\frac{1}{\tau_2}
\end{bmatrix}
$$

$$
G = \begin{bmatrix}
0 & 0 & 0 \\
\frac{1}{K_c \tau_m} & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & \frac{1}{K_c \tau_m} & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & \frac{1}{K_c \tau_m}
\end{bmatrix}
$$

$$x' = \left[\phi', v''_\phi, \omega'''_\phi, \theta', v''_\theta, \omega'''_\theta, \psi', v''_\psi, \omega'''_\psi\right]$$

$$\omega'''_\theta = \frac{\omega''_\theta}{d_{12}s(\gamma_{1c} - \gamma_{2c})} \quad ; \quad \omega'''_\psi = \frac{\omega''_\psi}{d_{12}s(\gamma_{1c} - \gamma_{2c})}$$

$$f(v) = \begin{bmatrix} f\left\{K_c\left(1 + \frac{\tau_1}{\tau_2}\right)(\phi' - t\beta_{1c}c\gamma_{1c}\theta' + t\beta_{1c}s\gamma_{1c}\psi') + \frac{K_c\tau_1}{\tau_2} d_{12}s(\gamma_{1c} - \gamma_{2c})\omega'''_\phi + \frac{I}{K_m} h^o_\phi\right\} - \frac{I}{K_m} h^o_\phi \\ f\left\{K_c\left(1 + \frac{\tau_1}{\tau_2}\right)d_{12}\left[s(\gamma_{1c} - \gamma_{2c}) \cdot \theta'\right] + \frac{K_c\tau_1}{\tau_2} d_{12}s(\gamma_{1c} - \gamma_{2c})\omega'''_\theta + \frac{I}{K_m} h^o_\theta\right\} - \frac{I}{K_m} h^o_\theta \\ f\left\{K_c\left(1 + \frac{\tau_1}{\tau_2}\right)d_{12}\left[s(\gamma_{1c} - \gamma_{2c}) \cdot \psi'\right] + \frac{K_c\tau_1}{\tau_2} d_{12}s(\gamma_{1c} - \gamma_{2c})\omega'''_\psi + \frac{I}{K_m} h^o_\psi\right\} - \frac{I}{K_m} h^o_\psi \end{bmatrix}$$

If we define T to be a matrix whose columns are the eigenvectors of A, and relate y to x by

$$x = Ty \tag{18}$$

we obtain

$$\dot{y} = T^{-1}ATy + T^{-1}Gf(v) \ . \tag{19}$$

It can be shown that

$$T^{-1}AT = \text{diag}\left[0,\ 0,\ 0,\ -\frac{1}{\tau_m},\ -\frac{1}{\tau_m},\ -\frac{1}{\tau_m},\ -\frac{1}{\tau_2},\ -\frac{1}{\tau_2},\ -\frac{1}{\tau_2}\right] , \tag{20}$$

where T can be written as

$$
T = \begin{bmatrix}
0 & 0 & 0 & -\frac{1}{\tau_m} & -\frac{1}{\tau_m} & -\frac{1}{\tau_m} & -\frac{1}{\tau_2} & -\frac{1}{\tau_2} & -\frac{1}{\tau_2} \\
1 & 0 & (2\tau_2 t\beta_{1c} c\gamma_{1c}) & \frac{\tau_2}{\tau_m} - 1 & 0 & -2\tau_2 t\beta_{1c} s\gamma_{1c} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{I(\tau_2 - \tau_m)}{K_m K_c \tau_m^2} & 0 & \frac{-2I\tau_2 t\beta_{1c} s\gamma_{1c}}{\tau_m K_m K_c} & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & \frac{-2\tau_2 \tau_m t\beta_{1c} c\gamma_{1c}}{\tau_2 - \tau_m} & 0 & 1 & 0 & 0 \\
0 & \frac{-1}{c\gamma_{1c}} & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{I}{\tau_m K_m K_c} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{c\gamma_{1c}} & -1 & 0 & \frac{\tau_m}{\tau_2 - \tau_m} & 0 & 0 & 1 & 0 \\
0 & \frac{-1}{s\gamma_{1c}} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{I}{\tau_m K_m K_c} & 0 & 0 & 0 \\
0 & \frac{1}{s\gamma_{1c}} & 0 & 0 & 0 & -\frac{\tau_m}{\tau_m - \tau_2} & 0 & 0 & 1
\end{bmatrix}
$$

with

$$
T^{-1} = \begin{bmatrix}
1 & -\frac{K_m K_c \tau_m}{I} & 0 & -2\tau_2 t\beta_{1c} c\gamma_{1c} & \frac{2\tau_2 \tau_m K_m K_c t\beta_{1c} c\gamma_{1c}}{I} & 0 & 2\tau_2 t\beta_{1c} s\gamma_{1c} & -\frac{2\tau_2 \tau_m K_m K_c t\beta_{1c} s\gamma_{1c}}{I} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -s\gamma_{1c} & \frac{K_m K_c \tau_m s\gamma_{1c}}{I} & 0 \\
0 & 0 & 0 & 1 & -\frac{K_m K_c \tau_m}{I} & 0 & -t\gamma_{1c} & \frac{K_m K_c \tau_m s\gamma_{1c}}{I c\gamma_{1c}} & 0 \\
0 & \frac{K_m K_c \tau_m^2}{I(\tau_2 - \tau_m)} & 0 & 0 & 0 & 0 & 0 & \frac{2\tau_2 \tau_m^2 K_m K_c t\beta_{1c} s\gamma_{1c}}{I(\tau_2 - \tau_m)} & 0 \\
0 & 0 & 0 & 0 & \frac{K_m K_c \tau_m}{I} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_m K_c \tau_m}{I} & 0 \\
1 & \frac{\tau_m \tau_2 K_m K_c}{(\tau_m - \tau_2) I} & 1 & -2\tau_2 t\beta_{1c} c\gamma_{1c} & \frac{2K_m K_c \tau_2^2 t\beta_{1c} c\gamma_{1c} \tau_m}{I(\tau_2 - \tau_m)} & 0 & 2\tau_2 t\beta_{1c} s\gamma_{1c} & -\frac{2\tau_2^2 \tau_m K_m K_c t\beta_{1c} s\gamma_{1c}}{I(\tau_2 - \tau_m)} & 0 \\
0 & 0 & 0 & 1 & \frac{\tau_m K_m K_c \tau_2}{I(\tau_m - \tau_2)} & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -\frac{\tau_2 \tau_m K_m K_c}{I(\tau_2 - \tau_m)} & 1
\end{bmatrix}
$$

If we now proceed to "tear" the system by defining

$$z^T = [y_4, y_5, y_6, y_7, y_8, y_9]$$

where superscript T denotes transpose and consider v to be just the linear part of the argument in f(v), it can be shown that the system equations will reduce to the form

$$\dot{z} = A^* z + B^* f(v)$$

$$\dot{v} = H^* z + J^* f(v) \quad , \tag{21}$$

where

$$J^* \equiv 0$$

$$A^* = \text{diag}\left[-\frac{1}{\tau_m}, -\frac{1}{\tau_m}, -\frac{1}{\tau_m}, -\frac{1}{\tau_2}, -\frac{1}{\tau_2}, -\frac{1}{\tau_2}\right] \quad .$$

$$B^* = \begin{bmatrix} \frac{K_m \tau_m}{I(\tau_2 - \tau_m)} & 0 & \frac{2\tau_2 \tau_m K_m t\beta_{1c} s\gamma_{1c}}{I(\tau_2 - \tau_m)} \\ 0 & \frac{K_m}{I} & 0 \\ 0 & 0 & \frac{K_m}{I} \\ \frac{K_m \tau_2}{I(\tau_m - \tau_2)} & \frac{2K_m \tau_2^2 t\beta_{1c} c\gamma_{1c}}{I(\tau_2 - \tau_m)} & \frac{-2\tau_2^2 K_m t\beta_{1c} s\gamma_{1c}}{I(\tau_2 - \tau_m)} \\ 0 & \frac{K_m \tau_2}{I(\tau_m - \tau_2)} & 0 \\ 0 & 0 & \frac{K_m \tau_2}{I(\tau_m - \tau_2)} \end{bmatrix}$$

$$H^* = \begin{bmatrix} \frac{K_c}{\tau_m}\left[1 - \frac{\tau_2 + \tau_1}{\tau_m}\right] & \frac{K_c t\beta_{1c} c\gamma_{1c}}{\tau_m}\left[\frac{2\tau_1 \tau_m}{\tau_2 - \tau_m} + \frac{\tau_2 + \tau_1}{\tau_2}\right] & \frac{K_c}{\tau_m}\left(1 + \frac{\tau_1}{\tau_2}\right)(2\tau_2 - 1) t\beta_{1c} s\gamma_{1c} & -\frac{K_c \tau_1}{\tau_2^2} & 0 & 0 \\ 0 & -\frac{K_c}{\tau_m} d_{12} s(\gamma_{1c} - \gamma_{2c})\left[\frac{\tau_1 + \tau_2 - \tau_m}{\tau_2 - \tau_m}\right] & 0 & 0 & -\frac{K_c \tau_1}{\tau_2^2} d_{12} s(\gamma_{1c} - \gamma_{2c}) & 0 \\ 0 & 0 & -\frac{K_c}{\tau_m} d_{12} s(\gamma_{1c} - \gamma_{2c})\left[\frac{\tau_1 + \tau_2 - \tau_m}{\tau_2 - \tau_m}\right] & 0 & 0 & -\frac{K_c \tau_1}{\tau_2^2} d_{12} s(\gamma_{1c} - \gamma_{2c}) \end{bmatrix}$$

Elimination of Compensator Lag Dynamics

The transfer function of the lead-lag compensation network is described by

$$G_c(s) = K_c \frac{(\tau_1 + \tau_2) s + 1}{\tau_2 s + 1} , \qquad (22)$$

where

$K_c = 2.685 \times 10^5$ volt/rad

$\tau_1 = 4.5$ sec , $\tau_2 = .5$ sec .

Therefore,

$$G_c(s) = K_c\left[\frac{9s}{s+2} + 1\right] \quad . \tag{23}$$

Considering the fact that the rotational rates of the vehicle are much slower than 2 rad/sec, let us ignore for this treatment the effect of lag dynamics in $G_c(p)$. With this simplifying assumption, the resulting equations become

$$\frac{V'(s)}{\epsilon(s)} = K_c(4.5\ s + 1) \tag{24}$$

with a corresponding differential equation for each channel,

$$V' = K_c(4.5\ \dot{\epsilon} + \epsilon) \quad . \tag{25}$$

Proceeding to solve for $\epsilon, \dot{\epsilon}$ we obtain:

$$\left.\begin{aligned} \epsilon_\phi &= \Delta\gamma_1 \\ \epsilon_\theta &= d_{12}\left[c(\Delta\gamma_2 + \gamma_{2c})\Delta\beta_1 + c(\Delta\gamma_1 + \gamma_{1c})\Delta\beta_2\right] \\ \epsilon_\psi &= -\ d_{12}\left[s(\Delta\gamma_2 + \gamma_{2c})\ \Delta\beta_1 + s(\Delta\gamma_1 + \gamma_{1c})\Delta\beta_2\right] \end{aligned}\right\} \tag{26}$$

$$\begin{aligned} \dot{\epsilon}_\phi = \Delta\dot{\gamma}_1 &= p - t(\Delta\beta_1 + \beta_{1c}) \cdot c(\Delta\gamma_1 + \gamma_{1c}) \cdot q \\ &\quad + t(\Delta\beta_1 + \beta_{1c}) \cdot s(\Delta\gamma_1 + \gamma_{1c}) \cdot r \\ \dot{\epsilon}_\phi &= \left(h_\phi^o - \frac{v_\phi}{I}\right) - t(\Delta\beta_1 + \beta_{1c}) \cdot c(\Delta\gamma_1 + \gamma_{1c}) \cdot \left(h_\theta^o - \frac{v_\theta}{I}\right) \\ &\quad + t(\Delta\beta_1 + \beta_{1c}) \cdot s(\Delta\gamma_1 + \gamma_{1c}) \cdot \left(h_\psi^o - \frac{v_\psi}{I}\right) \end{aligned} \tag{27}$$

$$\dot{\epsilon}_\theta = d_{12} c(\Delta\gamma_2 + \gamma_{2c}) \cdot \Delta\dot{\beta}_1 - d_{12} s(\Delta\gamma_2 + \gamma_{2c}) \cdot \Delta\dot{\gamma}_2 \Delta\beta_1$$

$$+ d_{12} c(\Delta\gamma_1 + \gamma_{1c}) \cdot \Delta\dot{\beta}_2 - d_{12} s(\Delta\gamma_1 + \gamma_{1c}) \cdot \Delta\dot{\gamma}_1 \Delta\beta_2 \quad (27)\ (\text{Cont.})$$

By defining the following variables as:

$$PJ_1 \equiv s(\Delta\gamma_1 + \gamma_{1c}) \cdot \left(h_\theta^o - \frac{v_\theta}{I}\right) + c(\Delta\gamma_1 + \gamma_{1c}) \cdot \left(h_\psi^o - \frac{v_\psi}{I}\right)$$

$$PJ_2 \equiv - s(\Delta\gamma_2 + \gamma_{2c}) \cdot \left(h_\theta^o - \frac{v_\theta}{I}\right) - c(\Delta\gamma_2 + \gamma_{2c}) \cdot \left(h_\psi^o - \frac{v_\psi}{I}\right)$$

$$PJ_3 \equiv \left(h_\phi^o - \frac{v_\phi}{I}\right) + t(\Delta\beta_2 + \beta_{2c}) \cdot c(\Delta\gamma_2 + \gamma_{2c}) \cdot \left(h_\theta - \frac{v_\theta}{I}\right) \quad (28)$$

$$- t(\Delta\beta_2 + \beta_{2c}) \cdot s(\Delta\gamma_2 + \gamma_{2c}) \cdot \left(h_\psi^o - \frac{v_\psi}{I}\right)$$

$$PJ_4 \equiv \left(h_\phi^o - \frac{v_\phi}{I}\right) - t(\Delta\beta_1 + \beta_{1c}) \cdot c(\Delta\gamma_1 + \gamma_{1c}) \cdot \left(h_\theta^o - \frac{v_\theta}{I}\right)$$

$$+ t(\Delta\beta_1 + \beta_{1c}) \cdot s(\Delta\gamma_1 + \gamma_{1c}) \cdot \left(h_\psi^o - \frac{v_\psi}{I}\right)$$

$$\dot{\epsilon}_\phi = PJ_4$$

$$\dot{\epsilon}_\theta = d_{12}\Big[c(\Delta\gamma_2 + \gamma_{2c}) \cdot PJ_1 + c(\Delta\gamma_1 + \gamma_{1c}) \cdot PJ_2$$

$$- s(\Delta\gamma_2 + \gamma_{2c}) \cdot \Delta\beta_1 \cdot PJ_3 - s(\Delta\gamma_1 + \gamma_{1c}) \cdot \Delta\beta_2 \cdot PJ_4\Big]$$

$$\dot{\epsilon}_\psi = - d_{12}\Big[s(\Delta\gamma_2 + \gamma_{2c}) \cdot PJ_1 + s(\Delta\gamma_1 + \gamma_{1c}) \cdot PJ_2$$

$$+ c(\Delta\gamma_2 + \gamma_{2c}) \cdot \Delta\beta_1 \cdot PJ_3 + c(\Delta\gamma_1 + \gamma_{1c}) \cdot \Delta\beta_2 \cdot PJ_4\Big] \tag{28 (Cont.)}$$

the resulting state equations become:

$$\begin{aligned}
\dot{\phi} &= h^o_\phi - \frac{1}{I} v_\phi + (t\theta s\phi) \cdot \left(h^o_\theta - \frac{1}{I} v_\theta\right) + (t\theta c\phi) \cdot \left(h^o_\psi - \frac{1}{I} v_\psi\right) \\
\dot{v}_\phi &= - \frac{1}{\tau_m} v_\phi + \frac{K_m}{\tau_m} f\left(K_c(\epsilon_\phi + 4.5\ \dot{\epsilon}_\phi)\right) \\
\dot{\theta} &= c\phi \cdot \left(h^o_\theta - \frac{1}{I} v_\theta\right) - s\phi \cdot \left(h^o_\psi - \frac{1}{I} v_\psi\right) \\
\dot{v}_\theta &= - \frac{1}{\tau_m} v_\theta + \frac{K_m}{\tau_m} f\left(K_c(\epsilon_\theta + 4.5\ \dot{\epsilon}_\theta)\right) \\
\dot{\psi} &= \left(\frac{s\phi}{c\theta}\right)\left(h^o_\theta - \frac{1}{I} v_\theta\right) + \left(\frac{c\phi}{c\theta}\right)\left(h^o_\psi - \frac{1}{I} v_\psi\right) \\
\dot{v}_\psi &= - \frac{1}{\tau_m} v_\psi + \frac{K_m}{\tau_m} f\left(K_c(\epsilon_\psi + 4.5\ \dot{\epsilon}_\psi)\right) \quad .
\end{aligned} \tag{29}$$

Rescaling as before, we obtain

$$v'' = \frac{1}{K_m K_c} v' = \frac{1}{K_m K_c} (v - Ih^o)$$

$$\phi' = \phi - \phi_e \quad , \quad \text{etc.}$$

with Eq. (29) becoming:

$$\dot{\phi}' = -\frac{K_m K_c}{I}\left[v''_\phi + (t\theta s\phi)v''_\phi + (t\theta c\phi)v''_\psi\right]$$

$$\dot{v}''_\phi = -\frac{1}{\tau_m} v''_\phi + \frac{1}{K_c \tau_m} f\left(K_c(\epsilon_\phi + 4.5\,\dot{\epsilon}_\phi)\right) - \frac{I}{\tau_m K_m K_c} h^o_\phi$$

$$\dot{\theta}' = -\frac{K_m K_c}{I}\left[c\phi \cdot v''_\theta - s\phi \cdot v''_\psi\right]$$

$$\dot{v}_\theta = -\frac{1}{\tau_m} v''_\theta + \frac{1}{K_c \tau_m} f\left(K_c(\epsilon_\theta + 4.5\,\dot{\epsilon}_\theta)\right) - \frac{I}{\tau_m K_m K_c} h^o_\theta \qquad (30)$$

$$\dot{\psi}' = -\frac{K_m K_c}{I}\left(\frac{s\phi}{c\theta} \cdot v''_\theta + \frac{c\phi}{c\theta} \cdot v''_\psi\right)$$

$$\dot{v}''_\psi = -\frac{1}{\tau_m} v''_\psi + \frac{1}{K_c \tau_m} f\left(K_c(\epsilon_\psi + 4.5\,\dot{\epsilon}_\psi)\right) - \frac{I}{\tau_m K_m K_c} h^o_\psi \quad .$$

Separating the linear and nonlinear terms in Eq. (30), putting them into the required form Eq. (11), and assuming that $\Delta\beta_1^{e*}$ and $\Delta\beta_2^{e*}$ are negligible, it can be shown that the six dimensional state equations take the form shown in Fig. 5. If we wish to consider the six dimensional model, and include only the nonlinearity due to the motor saturation function, the form of Fig. 5 reduces to that of Fig. 6.

Comparison of Models via Simulation

Four models of the Ames system were simulated, namely: the basic nine dimensional nonlinear model, the basic model with a hard saturation of the error signals $(\epsilon_\phi, \epsilon_\theta, \epsilon_\psi)$, the model with motor saturation as the only nonlinearity, and the six dimensional version (lead-lag replaced by pure lead) of the basic model. The purpose of these simulations is two-fold: 1) to gain

* $\Delta\beta_1^e = \left[\Delta\beta_1(\phi_e, \theta_e, \psi_e)\right]$, etc.

$$
\begin{bmatrix} \dot{\phi}' \\ \dot{v}''_\phi \\ \dot{\theta}' \\ \dot{v}''_\theta \\ \dot{\psi}' \\ \dot{v}''_\psi \end{bmatrix}
=
\begin{bmatrix}
0 & -\frac{K_mK_c}{I} & 0 & 0 & 0 & 0 \\
\frac{1}{\tau_m} & -\frac{1}{\tau_m}\left(1 + 4.5\frac{K_mK_c}{I}\right) & -\frac{t\beta_{1c}c\gamma_{1c}}{\tau_m} & 4.5\frac{K_mK_c}{\tau_m I}\cdot t\beta_{1c}c\gamma_{1c} & \frac{t\beta_{1c}s\gamma_{1c}}{\tau_m} & -\frac{4.5\,K_mK_c}{\tau_m I}\cdot t\beta_{1c}s\gamma_{1c} \\
0 & 0 & 0 & -\frac{K_mK_c}{I} & 0 & 0 \\
0 & 0 & \frac{d_{12}}{\tau_m}\cdot s(\gamma_{1c}-\gamma_{2c}) & \frac{1}{\tau_m}\left[1 + \frac{4.5\,K_mK_cd_{12}}{I}\cdot s(\gamma_{1c}-\gamma_{2c})\right] & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{K_mK_c}{I} \\
0 & 0 & 0 & 0 & \frac{d_{12}}{\tau_m}\cdot s(\gamma_{1c}-\gamma_{2c}) & -\frac{1}{\tau_m}\left[1 + \frac{4.5\,K_mK_cd_{12}}{I}\cdot s(\gamma_{1c}-\gamma_{2c})\right]
\end{bmatrix}
\begin{bmatrix} \phi' \\ v''_\phi \\ \theta' \\ v''_\theta \\ \psi' \\ v''_\psi \end{bmatrix}
+
\begin{bmatrix}
-\frac{K_mK_c}{I}\left(t\theta s\phi v''_\theta + t\theta c\phi v''_\psi\right) \\
\frac{1}{K_c\tau_m} f\left(K_c(\epsilon_\phi + 4.5\,\dot{\epsilon}_\phi)\right) - \frac{Ih^o_\phi}{\tau_mK_mK_c} - \frac{1}{K_c\tau_m}\left\{\text{lin } f_2\right\} \\
-\frac{K_mK_c}{I}\left((c\phi - 1)v''_\theta - s\phi v''_\psi\right) \\
\frac{1}{K_c\tau_m} f\left(K_c(\epsilon_\theta + 4.5\,\dot{\epsilon}_\theta)\right) - \frac{Ih^o_\theta}{\tau_mK_mK_c} - \frac{1}{K_c\tau_m}\left\{\text{lin } f_4\right\} \\
-\frac{K_mK_c}{I}\left(\frac{s\phi}{c\theta}v''_\theta + \left(\frac{c\phi}{c\theta} - 1\right)v''_\psi\right) \\
\frac{1}{K_c\tau_m} f\left(K_c(\epsilon_\psi + 4.5\,\dot{\epsilon}_\psi)\right) - \frac{Ih^o_\psi}{\tau_mK_mK_c} - \frac{1}{K_c\tau_m}\left\{\text{lin } f_6\right\}
\end{bmatrix}
$$

$$\text{lin } f_2 = K_c\left[\left(\phi' - (t\beta_{1c}c\gamma_{1c})\theta' + (t\beta_{1c}s\gamma_{1c})\psi'\right) + \frac{I}{K_cK_m}h^o_\phi - \frac{4.5\,K_mK_c}{I}\left(v''_\phi - (t\beta_{1c}c\gamma_{1c})v''_\theta + t\beta_{1c}s\gamma_{1c})v''_\psi\right)\right]$$

$$\text{lin } f_4 = K_cd_{12}\left[s(\gamma_{1c} - \gamma_{2c})\cdot\theta + \frac{Ih^o_\theta}{K_cK_m} - \frac{4.5\,K_mK_c}{I}(c\gamma_{2c}s\gamma_{1c} - c\gamma_{1c}s\gamma_{2c})v''_\theta\right]$$

$$\text{lin } f_6 = -K_cd_{12}\left[-s(\gamma_{1c} - \gamma_{2c})\psi' + \frac{Ih^o_\psi}{K_cK_m} - \frac{4.5\,K_mK_c}{I}(s\gamma_{2c}c\gamma_{1c} - s\gamma_{1c}c\gamma_{2c})v''_\psi\right]$$

$$\epsilon_\phi = \Delta\gamma_1 \qquad \dot{\epsilon}_\phi = PJ_4$$

$$\epsilon_\theta = d_{12}\left[c(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1 + c(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\right]$$

$$\dot{\epsilon}_\theta = d_{12}\left[c(\Delta\gamma_2 + \gamma_{2c})\cdot PJ_1 + c(\Delta\gamma_1 + \gamma_{1c})\cdot PJ_2 - s(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1\cdot PJ_3 - s(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\cdot PJ_4\right]$$

$$\epsilon_\psi = -d_{12}\left[s(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1 + s(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\right]$$

$$\dot{\epsilon}_\psi = -d_{12}\left[s(\Delta\gamma_2 + \gamma_{2c})\cdot PJ_1 + s(\Delta\gamma_1 + \gamma_{1c})PJ_2 + c(\Delta\gamma_2 + \gamma_{2c})\cdot\Delta\beta_1\cdot PJ_3 + c(\Delta\gamma_1 + \gamma_{1c})\cdot\Delta\beta_2\cdot PJ_4\right]$$

Fig. 5 Six Dimensional Approximation of OAO "Paired-Tracker" System Model

$$
\begin{bmatrix} \dot{\phi}' \\ \dot{v}''_\phi \\ \dot{\theta}' \\ \dot{v}''_\theta \\ \dot{\psi}' \\ \dot{v}''_\psi \end{bmatrix}
=
\begin{bmatrix}
0 & -\frac{K_m K_c}{I} & 0 & 0 & 0 & 0 \\
\frac{1}{\tau_m} & -\frac{1}{\tau_m}\left(1 + 4.5\,\frac{K_m K_c}{I}\right) & -\frac{t\beta_{1c} c\gamma_{1c}}{\tau_m} & 4.5\,\frac{K_m K_c}{\tau_m I} \cdot t\beta_{1c} c\gamma_{1c} & \frac{t\beta_{1c} s\gamma_{1c}}{\tau_m} & -\frac{4.5\, K_m K_c}{\tau_m I} \cdot t\beta_{1c} s\gamma_{1c} \\
0 & 0 & 0 & -\frac{K_m K_c}{I} & 0 & 0 \\
0 & 0 & \frac{d_{12}}{\tau_m} \cdot s(\gamma_{1c} - \gamma_{2c}) & \frac{1}{\tau_m}\left[1 + \frac{4.5\, K_m K_c d_{12}}{I} \cdot s(\gamma_{1c} - \gamma_{2c})\right] & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{K_m K_c}{I} \\
0 & 0 & 0 & 0 & \frac{d_{12}}{\tau_m} \cdot s(\gamma_{1c} - \gamma_{2c}) & -\frac{1}{m}\left[1 + \frac{4.5\, K_m K_c d_{12}}{I} \cdot s(\gamma_{1c} - \gamma_{2c})\right]
\end{bmatrix}
\begin{bmatrix} \phi' \\ v''_\phi \\ \theta' \\ v''_\theta \\ \psi' \\ v''_\psi \end{bmatrix}
+
\begin{bmatrix}
0 \\
\frac{1}{K_c \tau_m} f(\text{lin } f_2) - \frac{I h^o_\phi}{\tau_m K_m K_c} - \frac{1}{K_c \tau_m}\{\text{lin } f_2\} \\
0 \\
\frac{1}{K_c \tau_m} f(\text{lin } f_4) - \frac{I h^o_\theta}{\tau_m K_m K_c} - \frac{1}{K_c \tau_m}\{\text{lin } f_4\} \\
0 \\
\frac{1}{K_c \tau_m} f(\text{lin } f_6) - \frac{I h^o_\psi}{\tau_m K_m K_c} - \frac{1}{K_c \tau_m}\{\text{lin } f_6\}
\end{bmatrix}
$$

$$\text{lin } f_2 = K_c\left[\left(\phi' - (t\beta_{1c} c\gamma_{1c})\theta' + (t\beta_{1c} s\gamma_{1c})\psi'\right) + \frac{I}{K_c K_m} h^o_\phi - \frac{4.5\, K_m K_c}{I}\left(v''_\phi - (t\beta_{1c} c\gamma_{1c}) v''_\theta + t\beta_{1c} s\gamma_{1c}) v''_\psi\right)\right]$$

$$\text{lin } f_4 = K_c d_{12}\left[s(\gamma_{1c} - \gamma_{2c}) \cdot \theta + \frac{I h^o_\theta}{K_c K_m} - \frac{4.5\, K_m K_c}{I}(c\gamma_{2c} s\gamma_{1c} - c\gamma_{1c} s\gamma_{2c}) v''_\theta\right]$$

$$\text{lin } f_6 = - K_c d_{12}\left[-s(\gamma_{1c} - \gamma_{2c})\psi' + \frac{I h^o_\psi}{K_c K_m} - \frac{4.5\, K_m K_c}{I}(s\gamma_{2c} c\gamma_{1c} - s\gamma_{1c} c\gamma_{2c}) v''_\psi\right]$$

Fig. 6 Six Dimensional Approximation of OAO "Paired-Tracker" System Model with Motor Saturation Only insight into the effect on performance of parameter (gimbal command, total momentum) variation, and 2) to compare the various models in order to evaluate the use of a simpler model for the determination of the estimate of the domain of attraction.

The system variables plotted are the roll, pitch, and yaw, euler angles and the wheel momenta. For the models presented with no gyro and inertia coupling, the wheel momenta are identical to the vehicle body rates within a constant scale factor and bias. One set of initial conditions (of the state) was used for all the runs. Variations were made on the commanded gimbal angles and the total system momentum. Figure 7 presents a run plan and a list of symbols that represent the four models. The criteria for comparison are settling time, number of overshoots, peak overshoot, and a general similarity of wave shape.

Some specific comparisons of the models are:

1) AN and MV are always very similar and MV always displays less coupling than AN (Figs. 8 through 12).

2) The pitch and yaw response are nearly identical to each other for AN, ANL and MV. The yaw response always has a longer settling time and more overshoots than pitch for 6D. This shows one radical difference between 6D and AN. The reason is unexplained to date, but it is unlikely to be a simulation flaw (Figs. 8 through 12).

3) ANL and AN are similar, but ANL displays less damping (Fig. 8).

4) ANL and 6D are effected more strongly by nonzero system momentum and $\beta_{1c} = 30°$. Figure 10 shows ANL

Run #	$\sin(\gamma_{1c} - \gamma_{2c})$	Inner Gimbal Commands Degrees β_{1c}	β_{2c}	Total Axis Momenta ft-lb sec Ih^o_ϕ	Ih^o_θ	Ih^o_ψ
1	0.1	0.0	-30.0	0.0	0.0	0.0
2	0.1	30.0	-30.0	0.0	0.0	0.0
3	0.1	30.0	-30.0	1.0	1.0	1.0
4	1.0	0.0	-30.0	0.0	0.0	0.0
5	1.0	0.0	-30.0	1.0	1.0	1.0

Initial Conditions: $\phi, \theta, \psi = 15.0^\circ$; $v_\phi, v_\theta, v_\psi = 1$ ft-lb$_f$ sec; $\omega_\phi, \omega_\theta, \omega_\psi = 100$ volts

System Parameters: $\tau_1 = 4.5$ sec; $\tau_2 = 0.5$ sec; $K_c = 2.685 \cdot 10^5$ volt/rad; $K_m = 1/13$ ft-lb$_f$-sec/volt; $I = 1500$ slug-ft^2

Basic Nonlinear Model - AN

Basic Model with Limiting - ANL

Motor Voltage Only Nonlinearity - MV

Six Dimensional Model - 6D

Fig. 7 Control System Simulation Run Plan and 6D undergoing roll motor voltage reversal significantly sooner than AN and MV. The pitch-yaw coupling into roll seems greater for the ANL and 6D pair under the above condition.

Thus it can be said that all four models have varying degrees of similarity with some radical differences appearing for the more severe initial conditions.

In addition to the runs described above two other situations were simulated and they are discussed separately because they produced unstable motions. The models used were AN, 6D, and MV with initial conditions $\phi(0) = \theta(0) = \psi(0) = 15°$, $v_\phi(0) = v_\theta(0) = v_\psi(0) = 1, 2$ ft-lb$_f$-sec, $\omega_\phi(0) = \omega_\theta(0) = \omega_\psi(0) = 100$ volts. The system parameters were $\gamma_{1c} = 10.7°$, $\gamma_{2c} = 5°\left(\sin(\gamma_{1c} - \gamma_{2c}) = 0.1\right)$, $\beta_{1c} = 30°$, $\beta_{2c} = 30°$, $Ih^o_\phi = Ih^o_\theta = Ih^o_\psi = 1$ ft-lb$_f$-sec. Note that these parameter values differ from those used previously, viz., $\beta_{1c} = 0, 30°$ and $\beta_{2c} = -30°$, and they do not satisfy the stability criterion given in Ref. 2, i.e., $|\gamma_{1c} - \gamma_{2c}| \gtrsim 10°$. Figure 13 presents the plotted results for the case $v_\phi(0) = v_\theta(0) = v_\psi(0) = 2$ ft-lb$_f$-sec, the other case (initial wheel momentum at half wheel capacity) yields similar results. As the figures show, unstable motions result when the AN and 6D models are used, but not when the MV model is used. Calculation of the error signals ϵ_ϕ, ϵ_θ, and ϵ_ψ at the initial point shows that ϵ_θ and ϵ_ψ have the wrong sign in AN and 6D, but they have the correct sign in MV. Thus, it is seen that although the motor saturation is the dominant nonlinearity for magnitude considerations ($\pm$ 20 arc secs of attitude error), the transcendental nonlinearities of the Tracker-error processor combination have an important bearing on the system stability; i.e., the use of the linearized error signal can produce erroneous stability conclusions. This emphasizes the need for the nonlinear analysis which was carried out in this study.

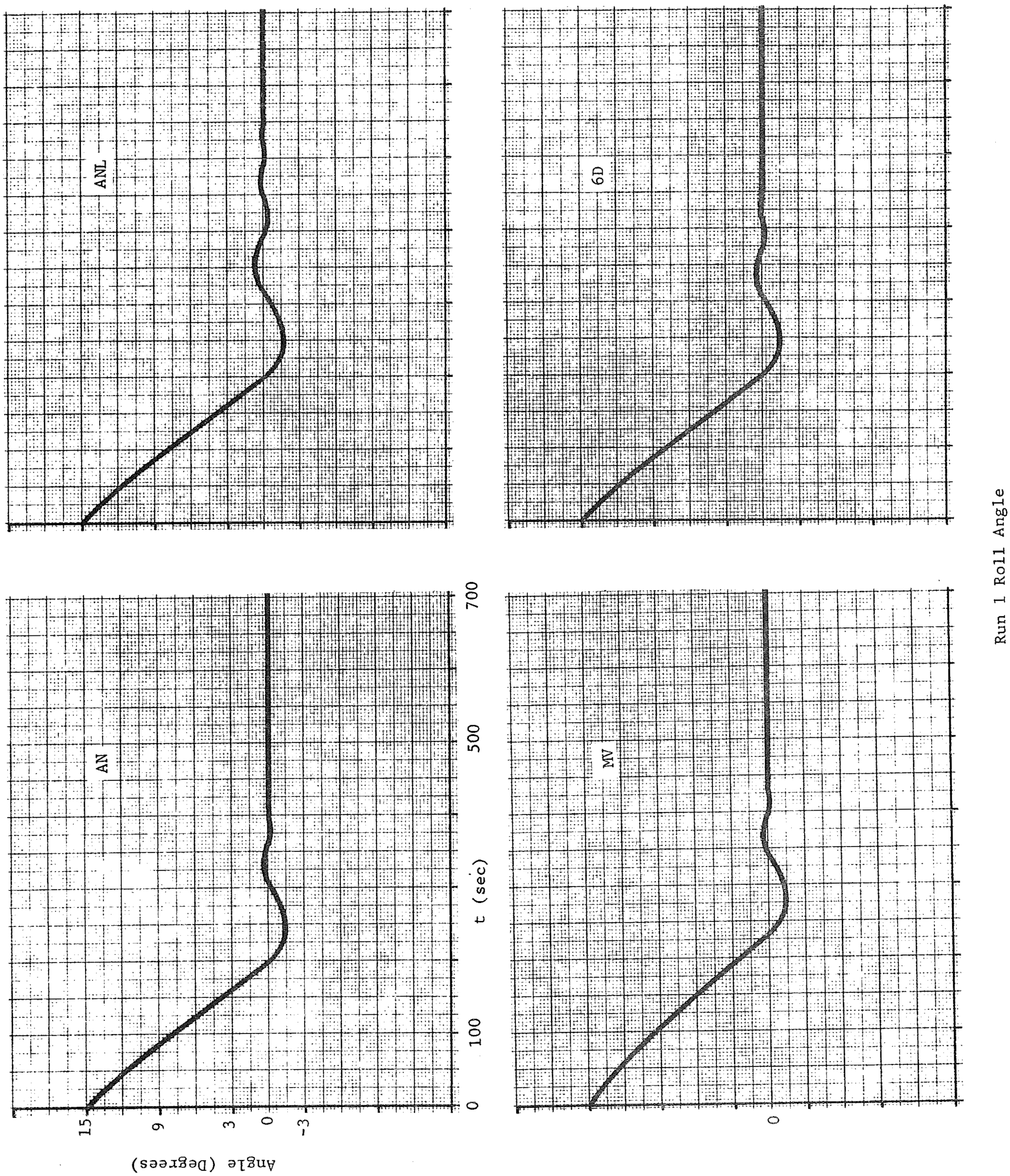


Fig. 8 Control System Simulation Run 1 (Sheet 1 of 6)

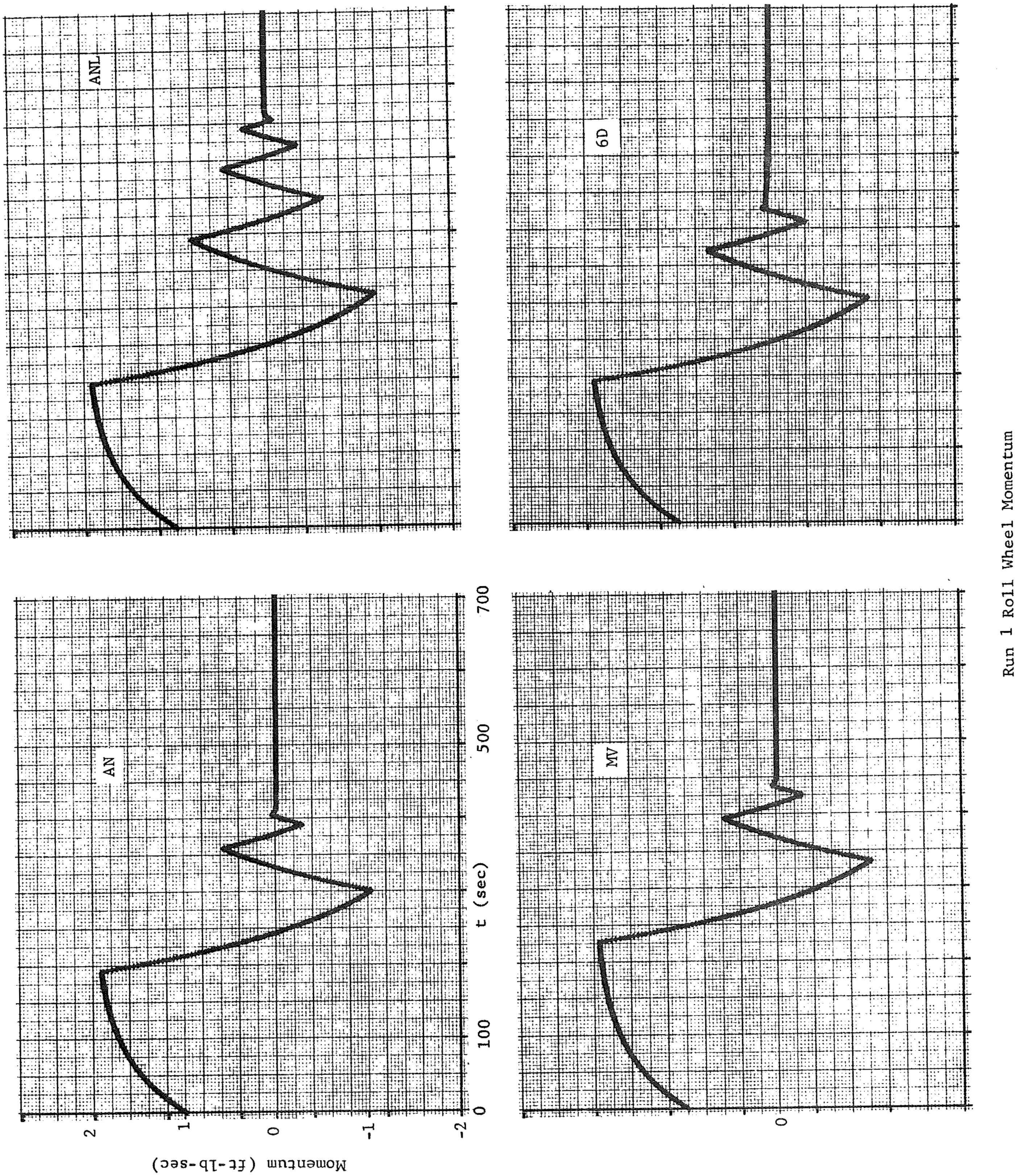


Fig. 8 Control System Simulation Run 1 (Sheet 2 of 6)

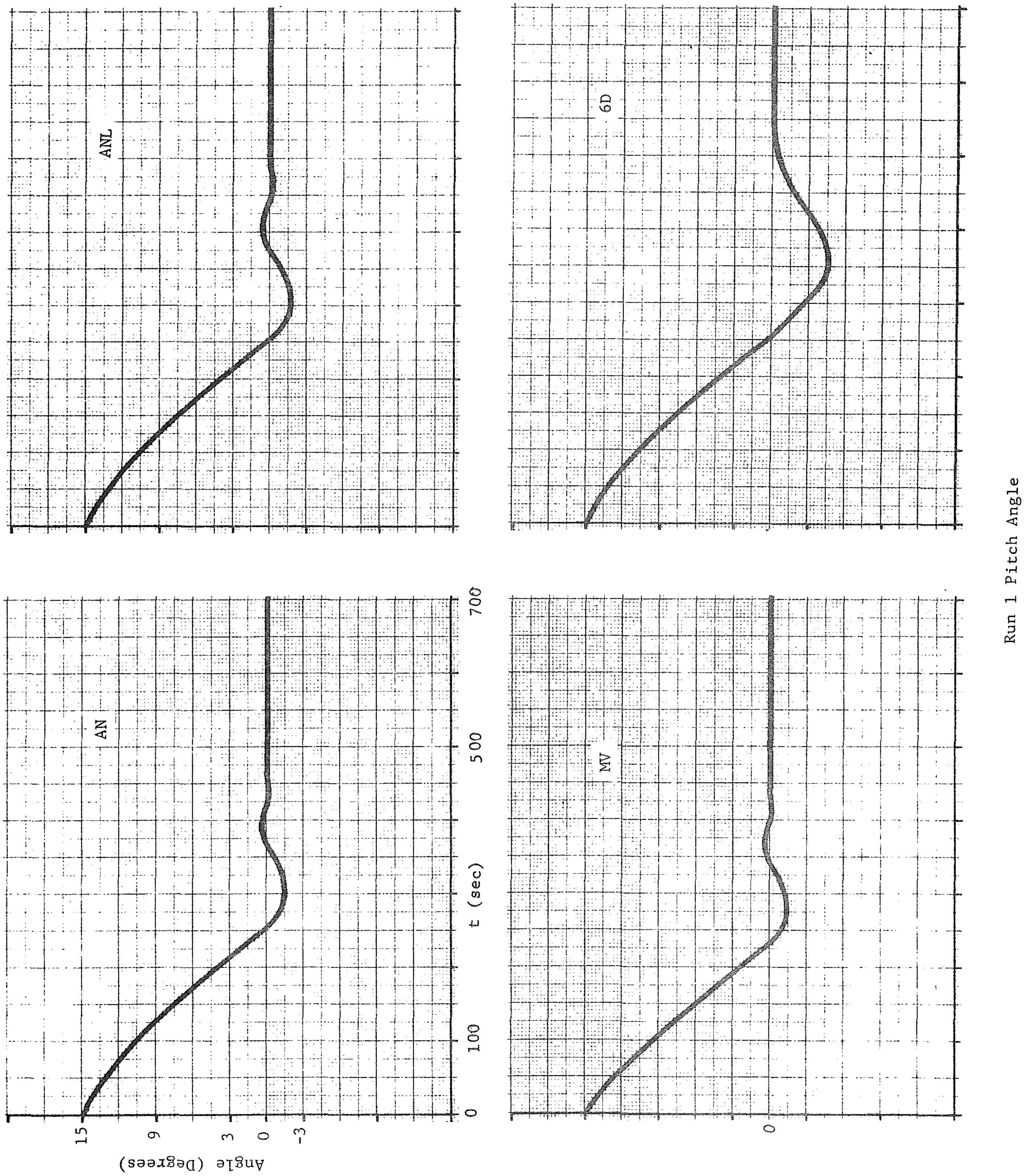


Fig. 8 Control System Simulation Run 1 (Sheet 3 of 6)

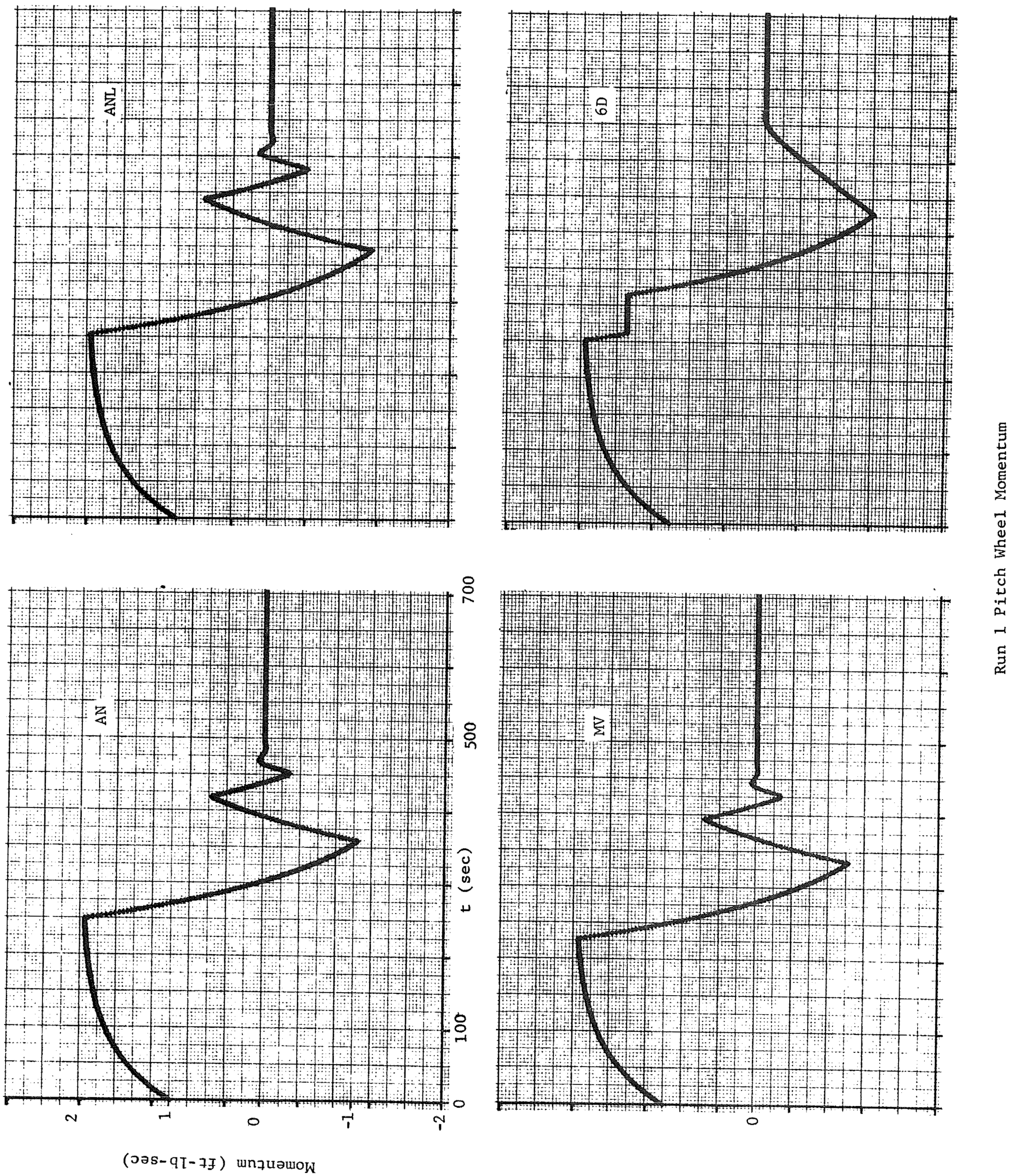


Fig. 8 Control System Simulation Run 1 (Sheet 4 of 6)

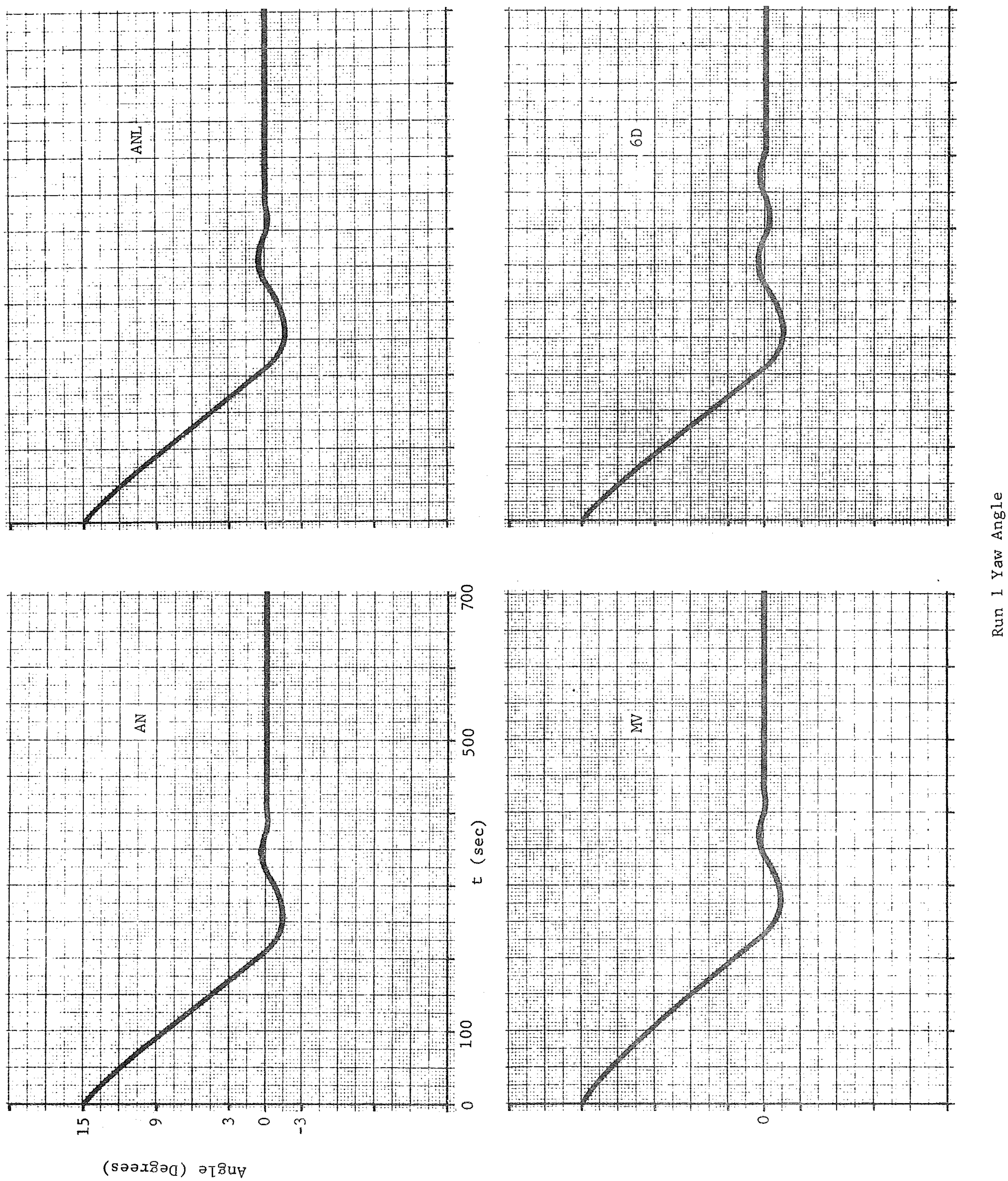


Fig. 8 Control System Simulation Run 1 (Sheet 5 of 6)

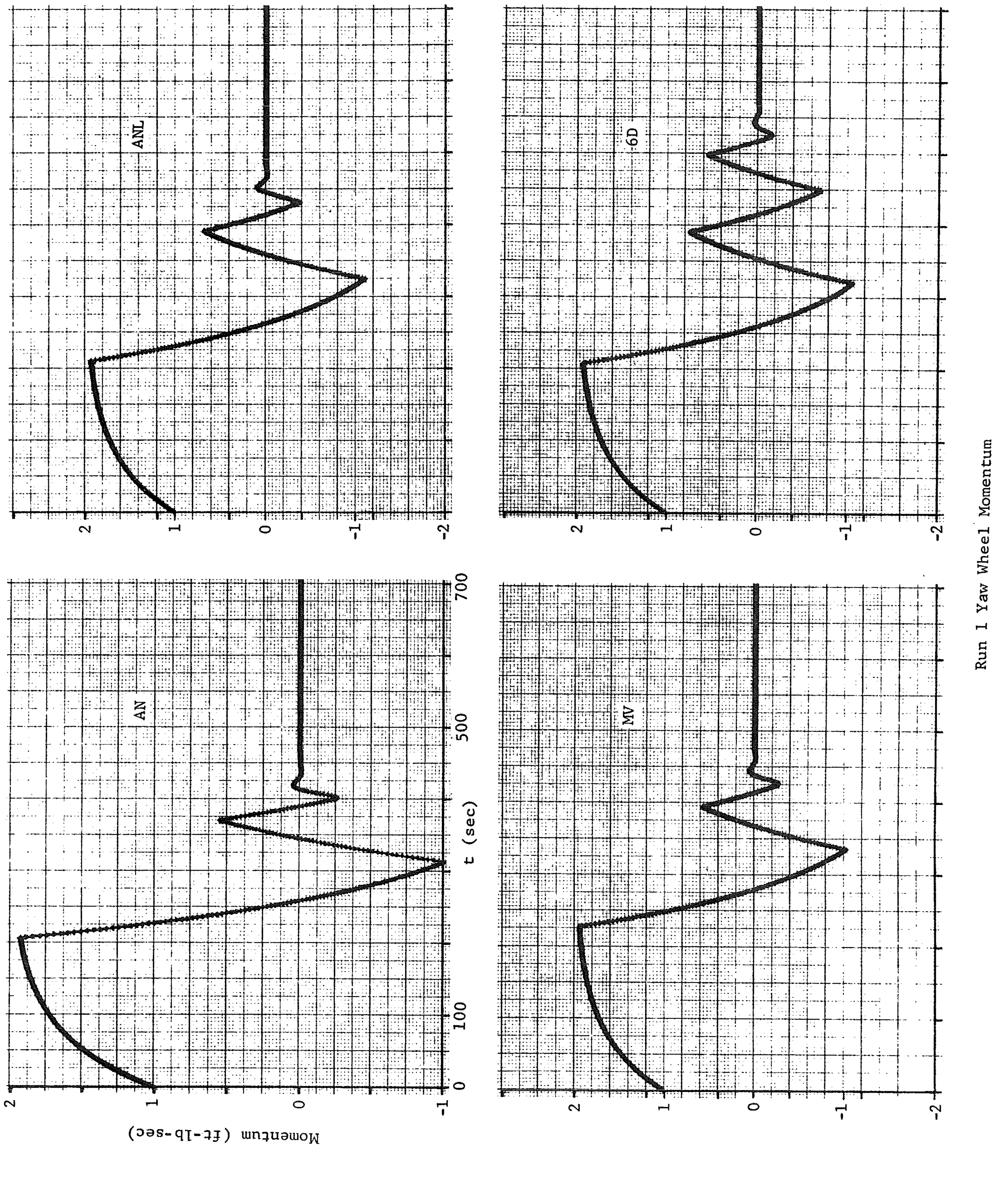


Fig. 8 Control System Simulation Run 1 (Sheet 6 of 6)

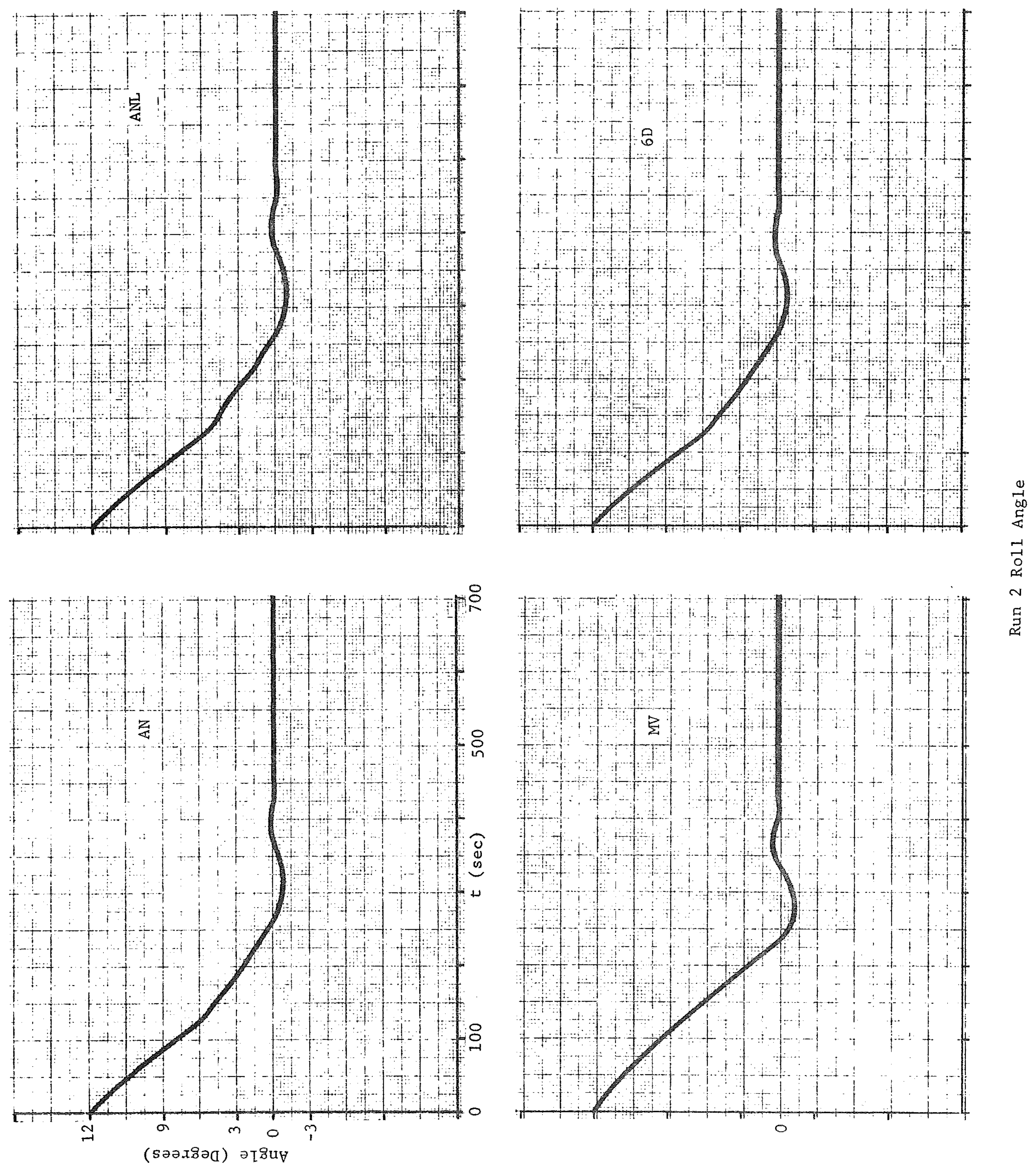


Fig. 9 Control System Simulation Run 2 (Sheet 1 of 6)

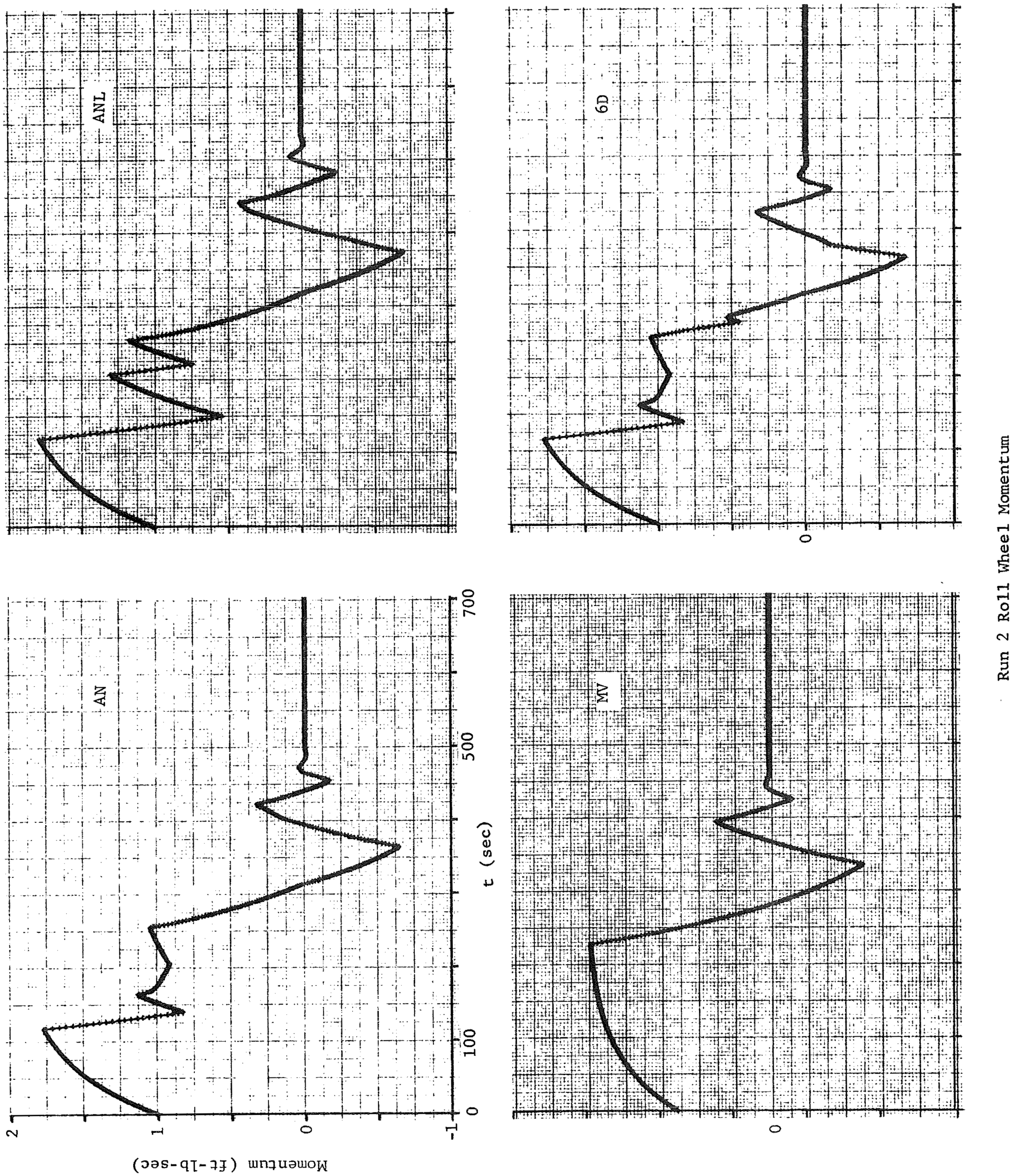


Fig. 9 Control System Simulation Run 2 (Sheet 2 of 6)

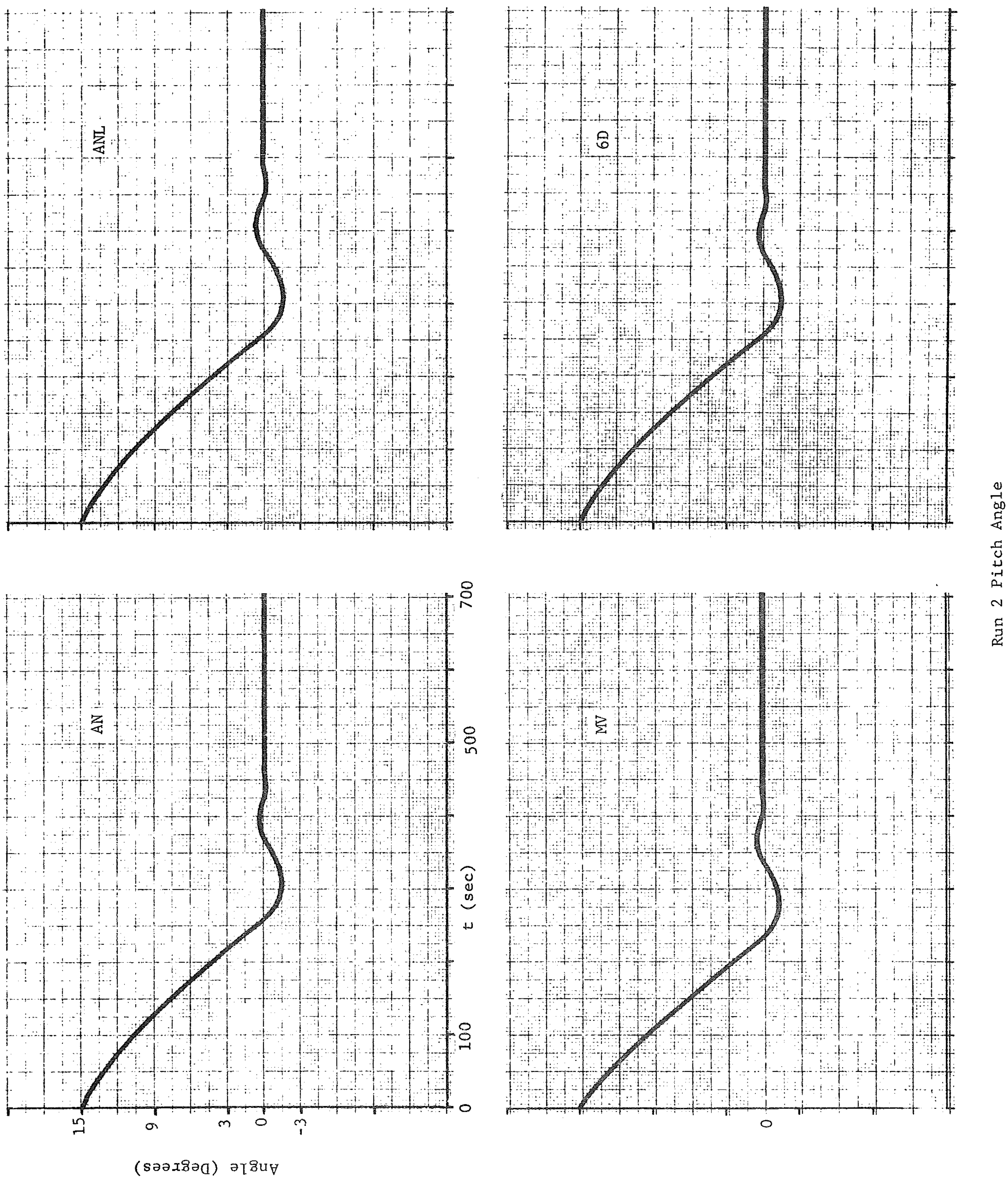
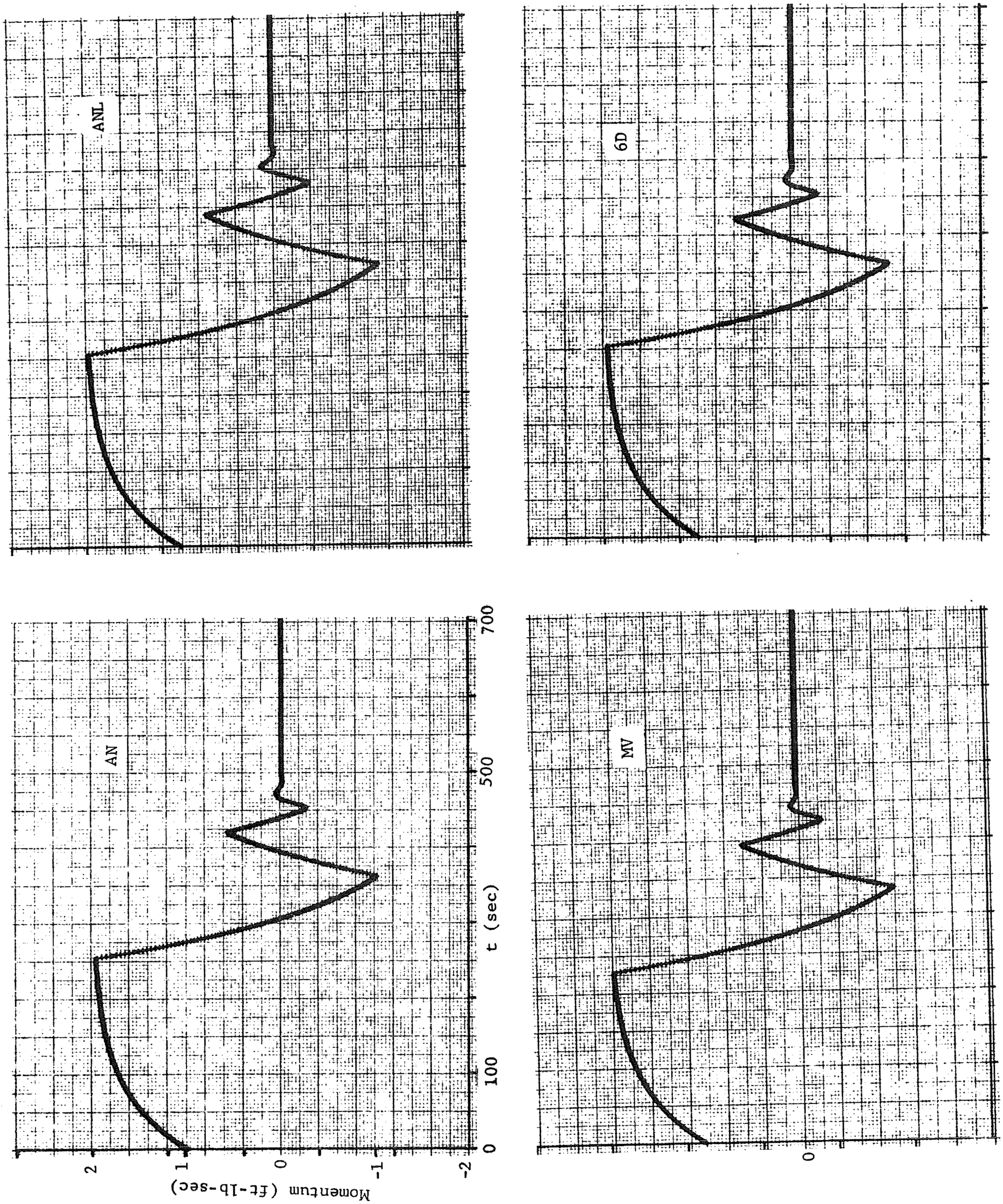


Fig. 9 Control System Simulation Run 2 (Sheet 3 of 6)



Run 2 Pitch Wheel Momentum

Fig. 9 Control System Simulation Run 2 (Sheet 4 of 6)

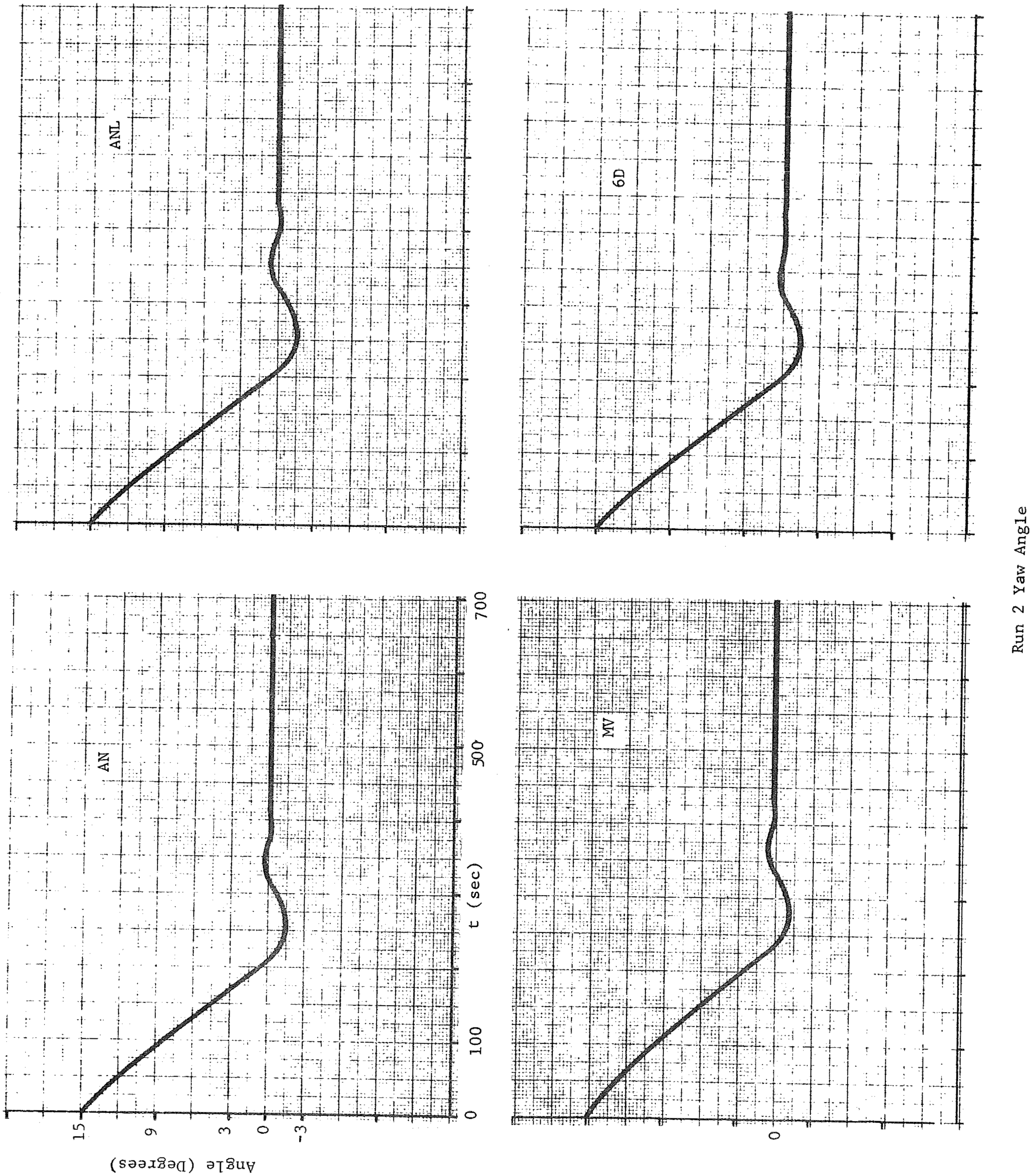


Fig. 9 Control System Simulation Run 2 (Sheet 5 of 6)

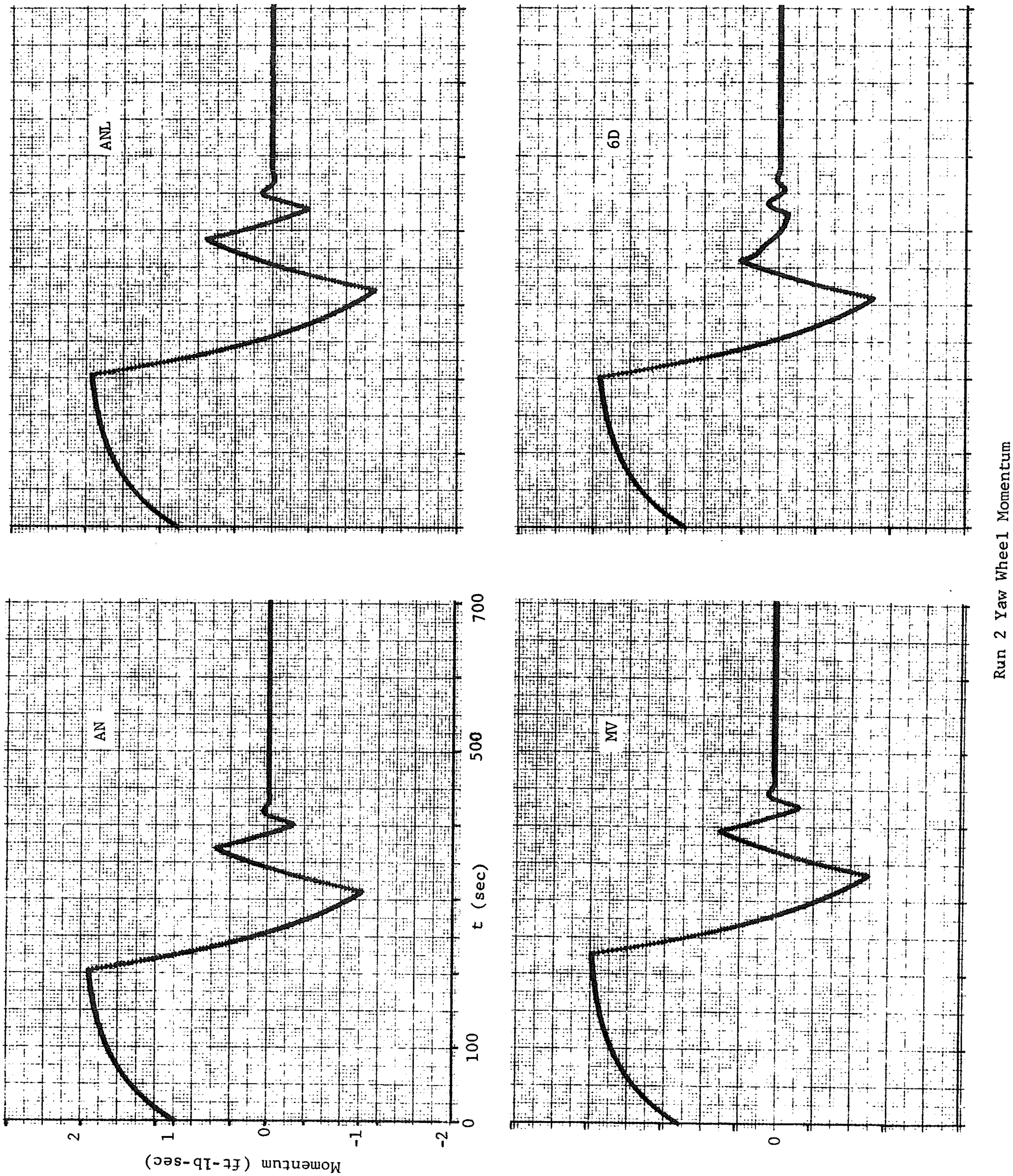


Fig. 9 Control System Simulation Run 2 (Sheet 6 of 6)

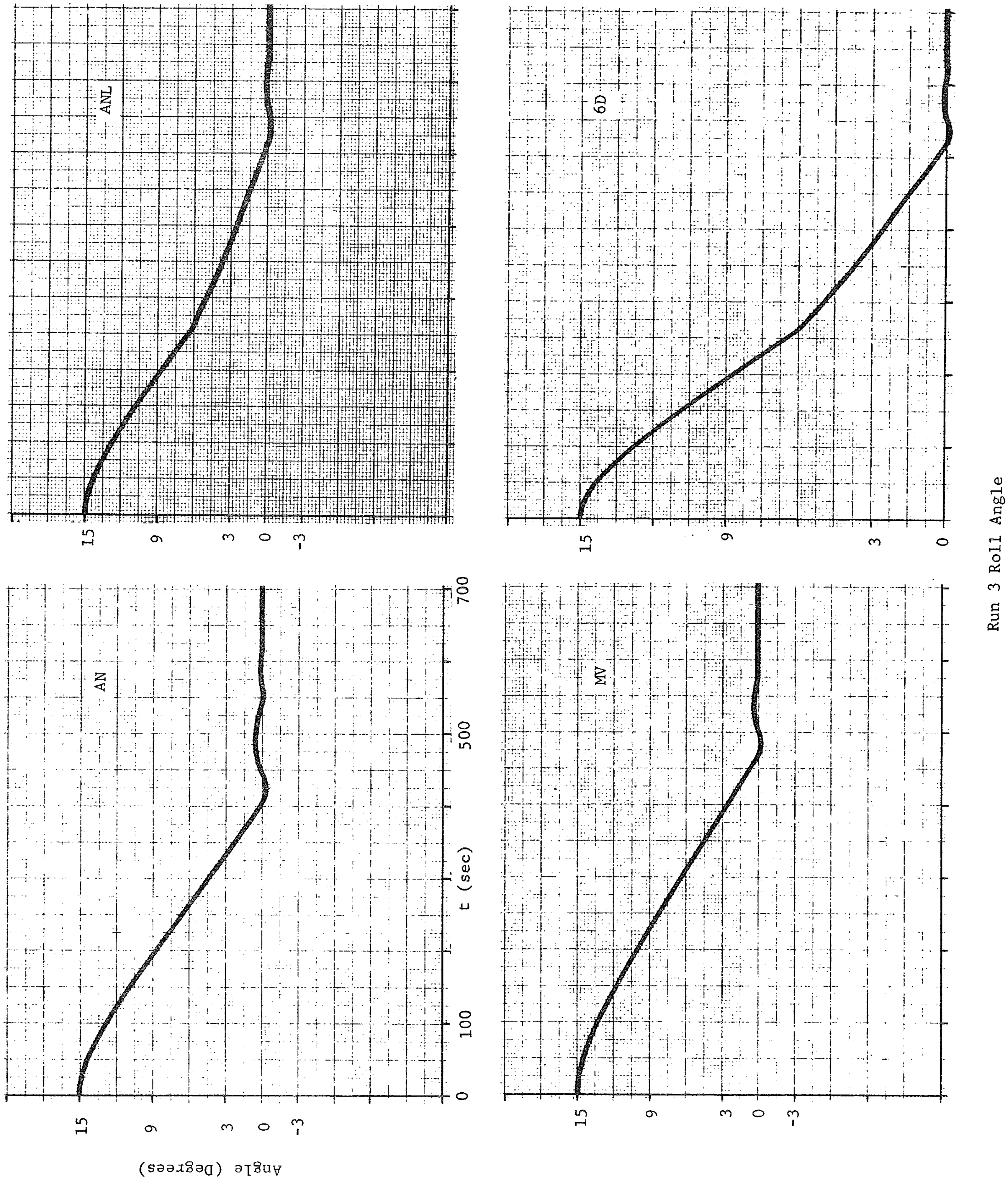


Fig. 10 Control System Simulation Run 3 (Sheet 1 of 6)

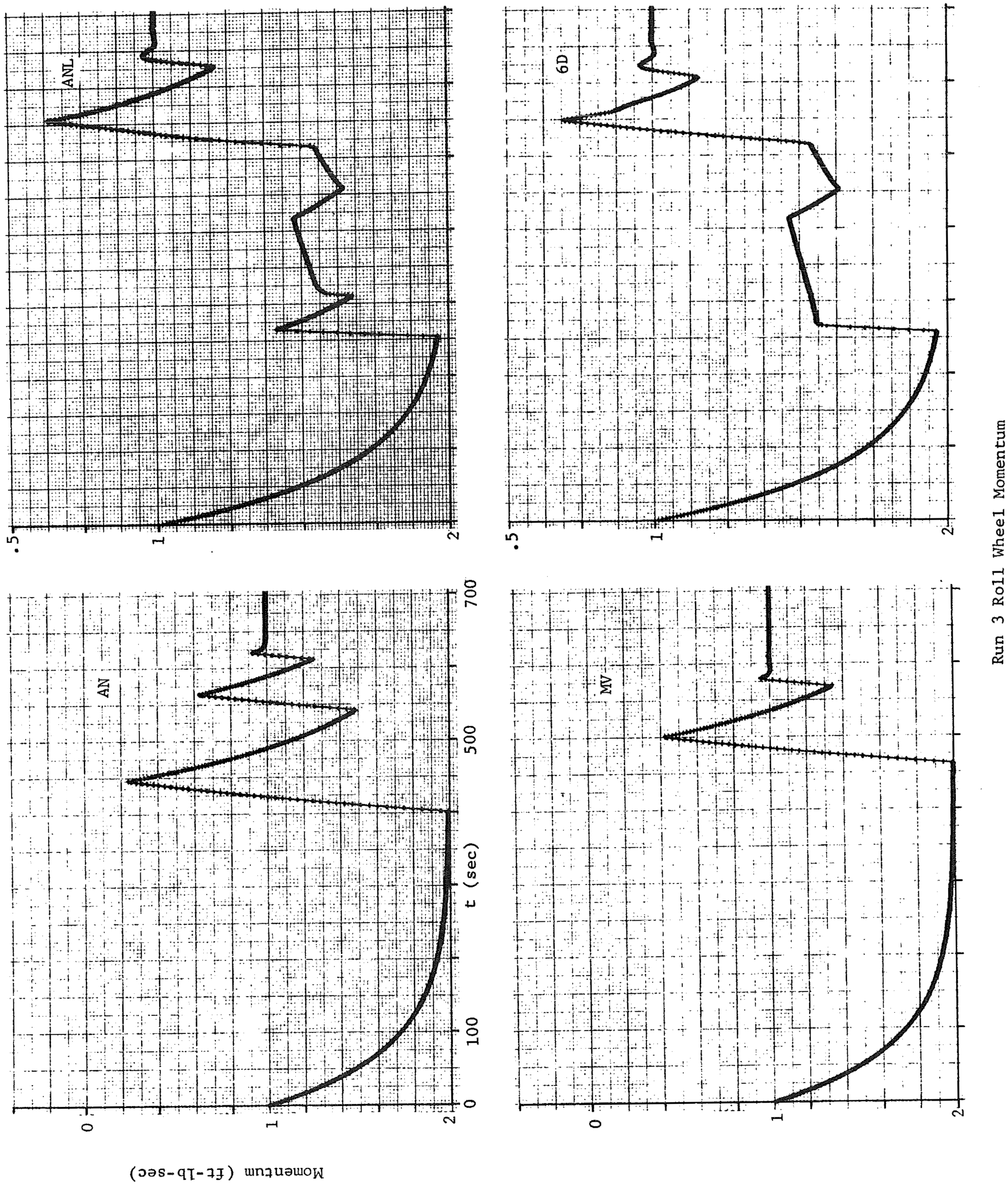


Fig. 10 Control System Simulation Run 3 (Sheet 2 of 6)

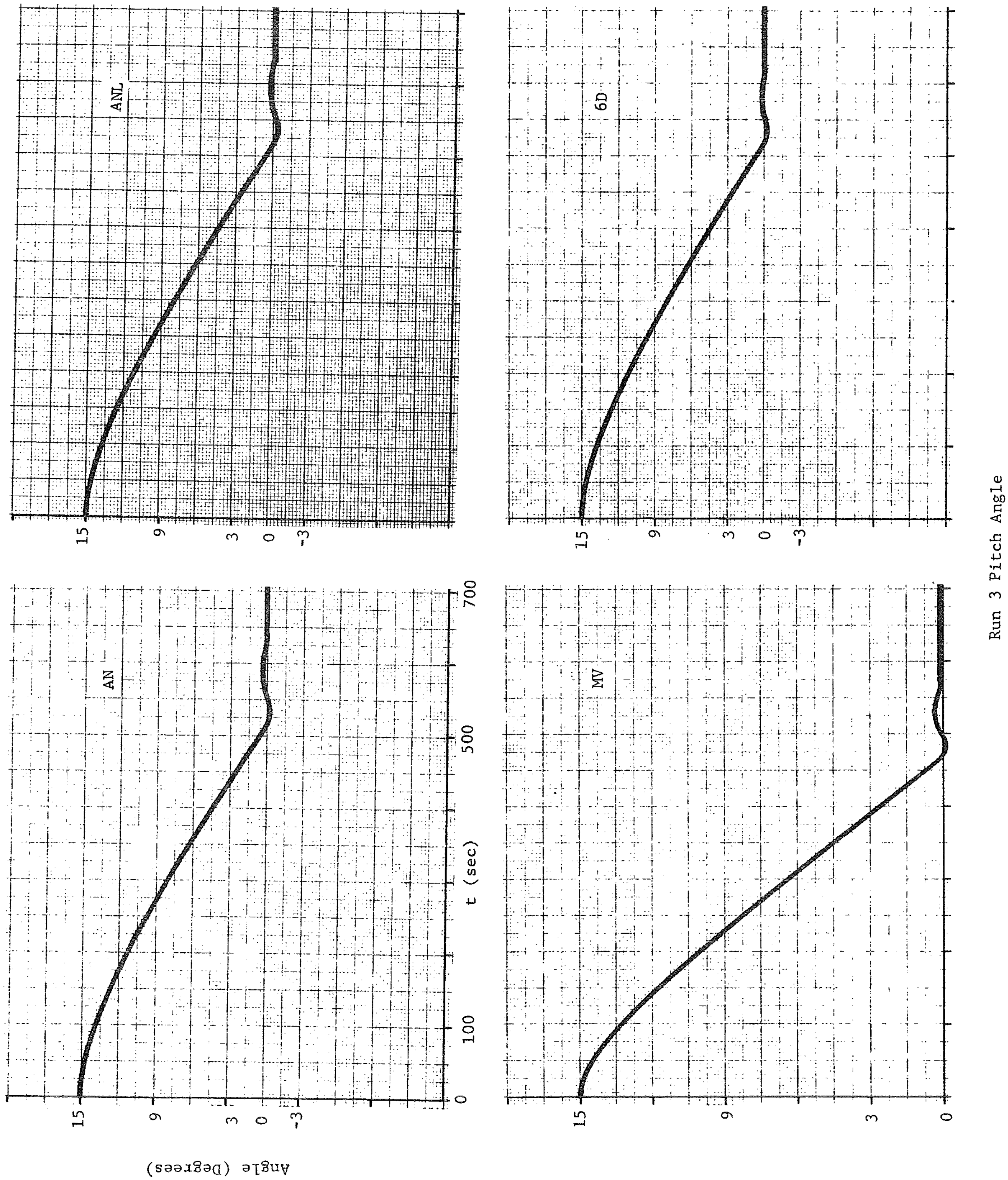


Fig. 10 Control System Simulation Run 3 (Sheet 3 of 6)

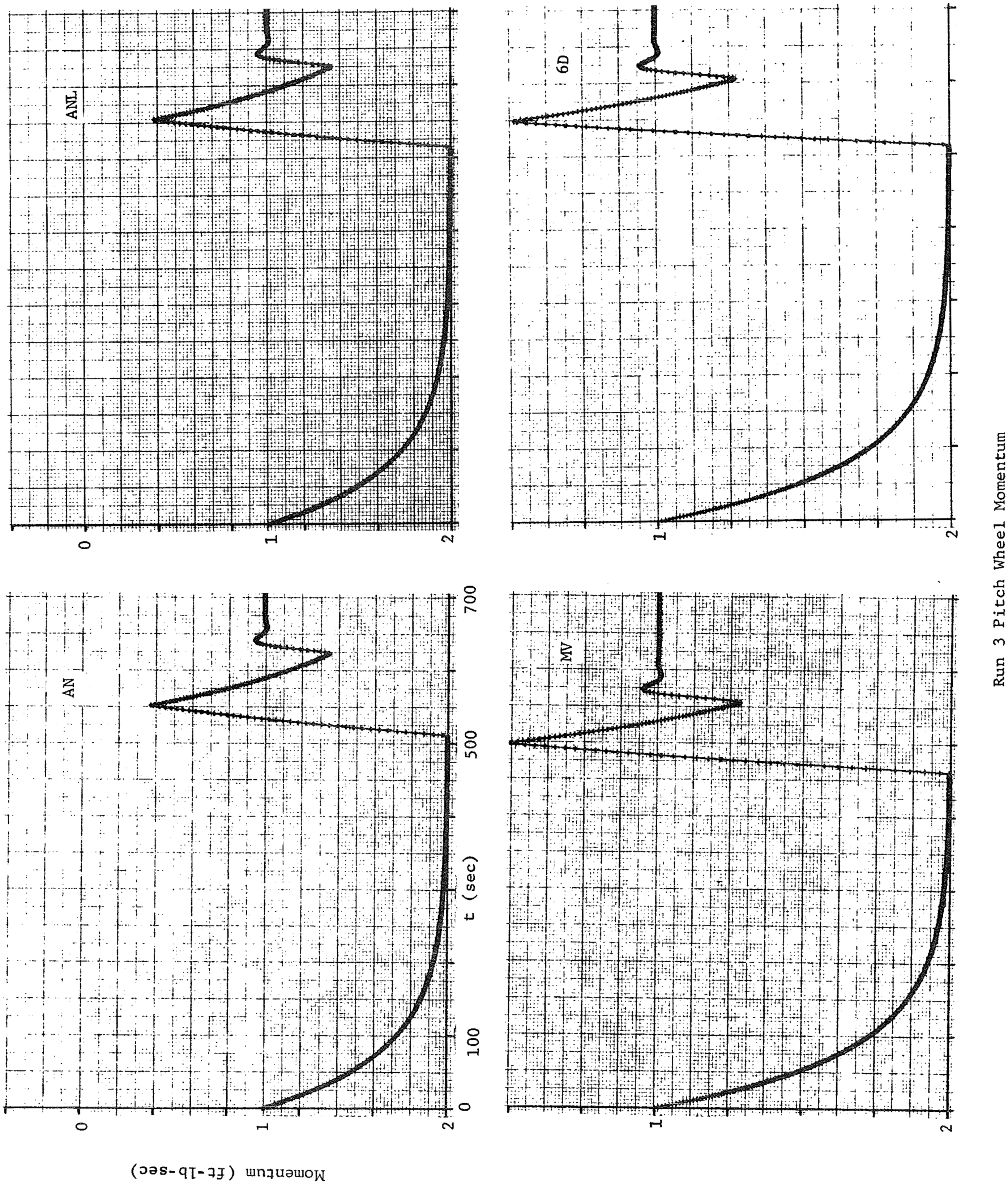


Fig. 10 Control System Simulation Run 3 (Sheet 4 of 6)

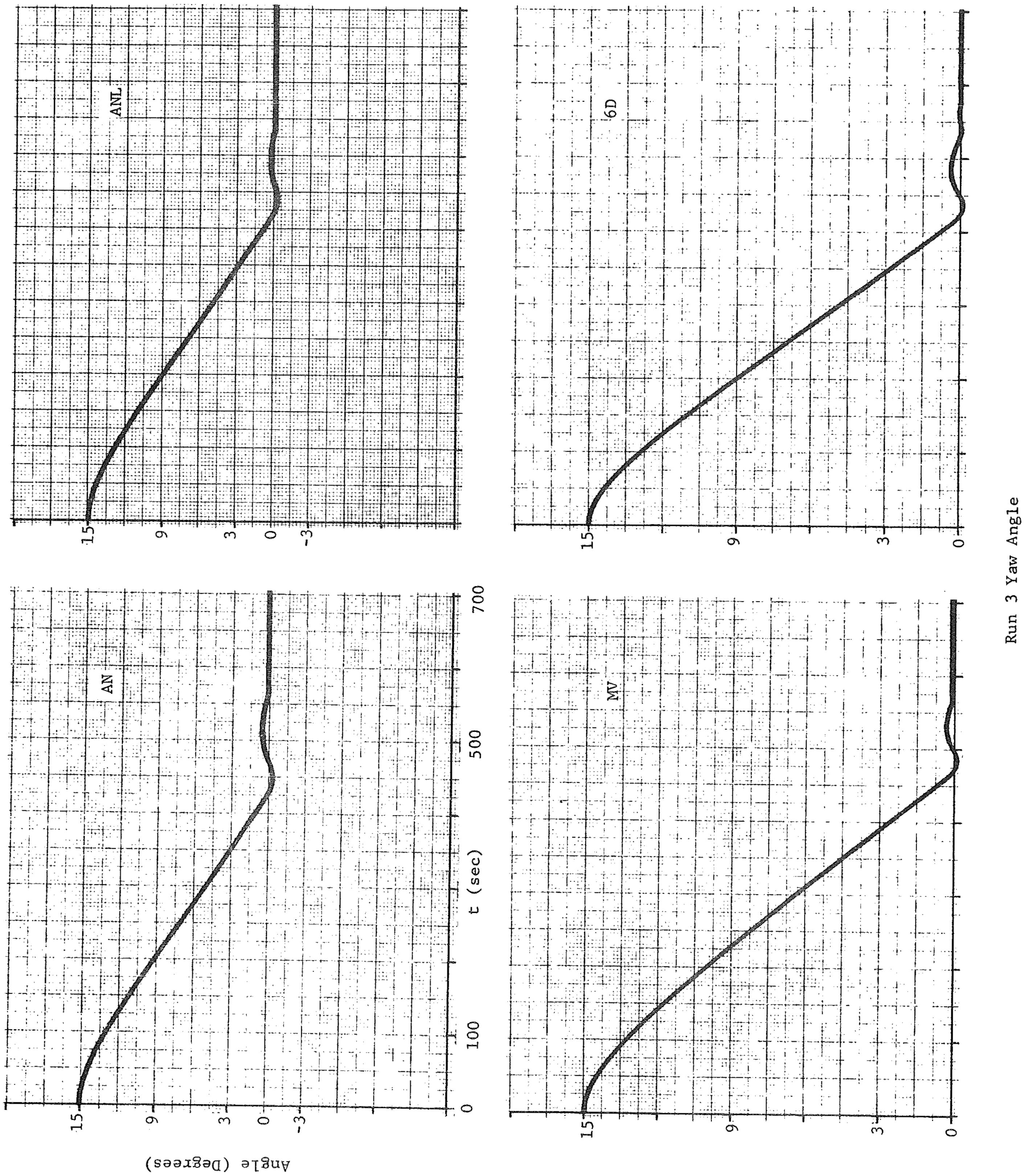


Fig. 10 Control System Simulation Run 3 (Sheet 5 of 6)

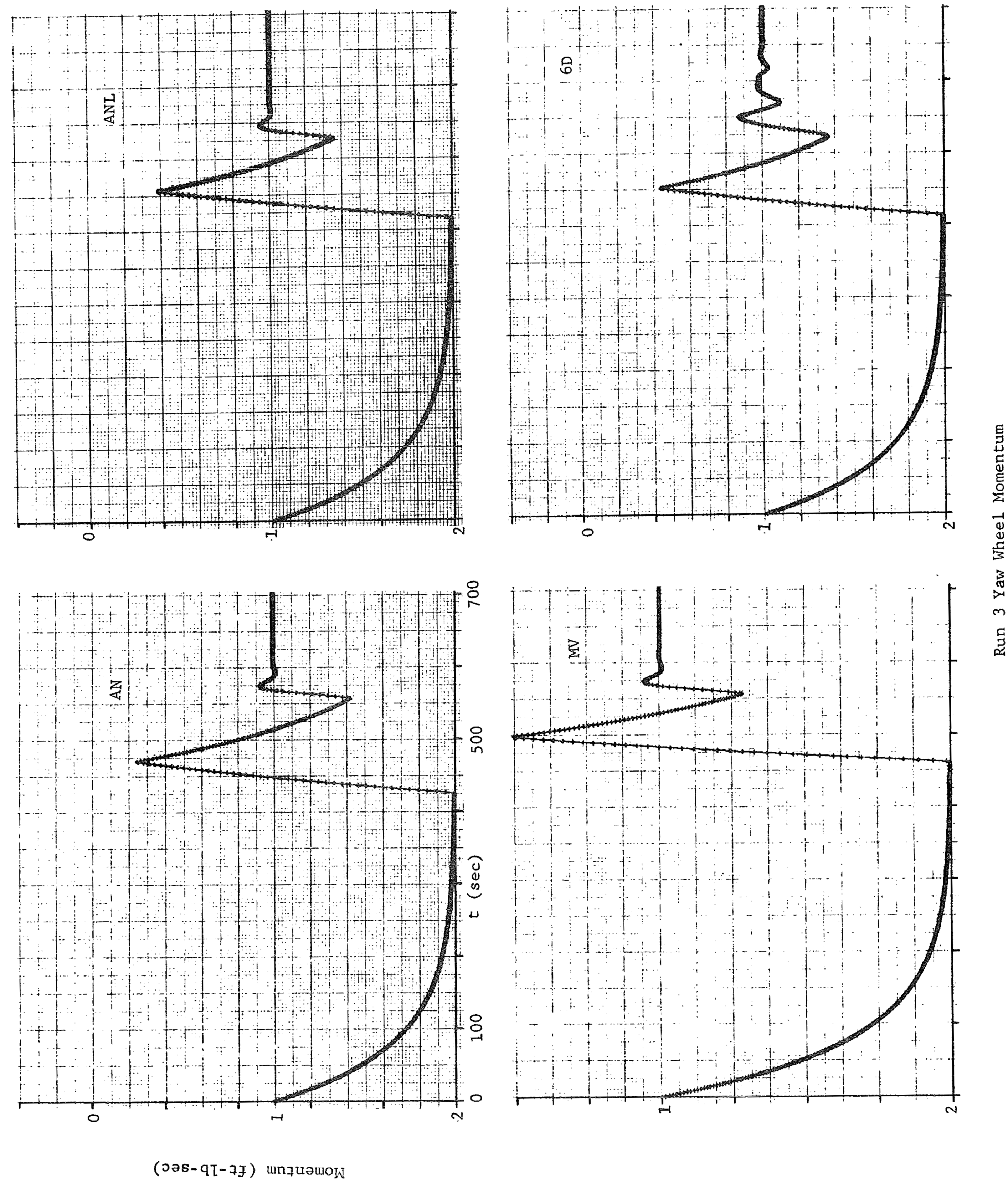


Fig. 10 Control System Simulation Run 3 (Sheet 6 of 6)

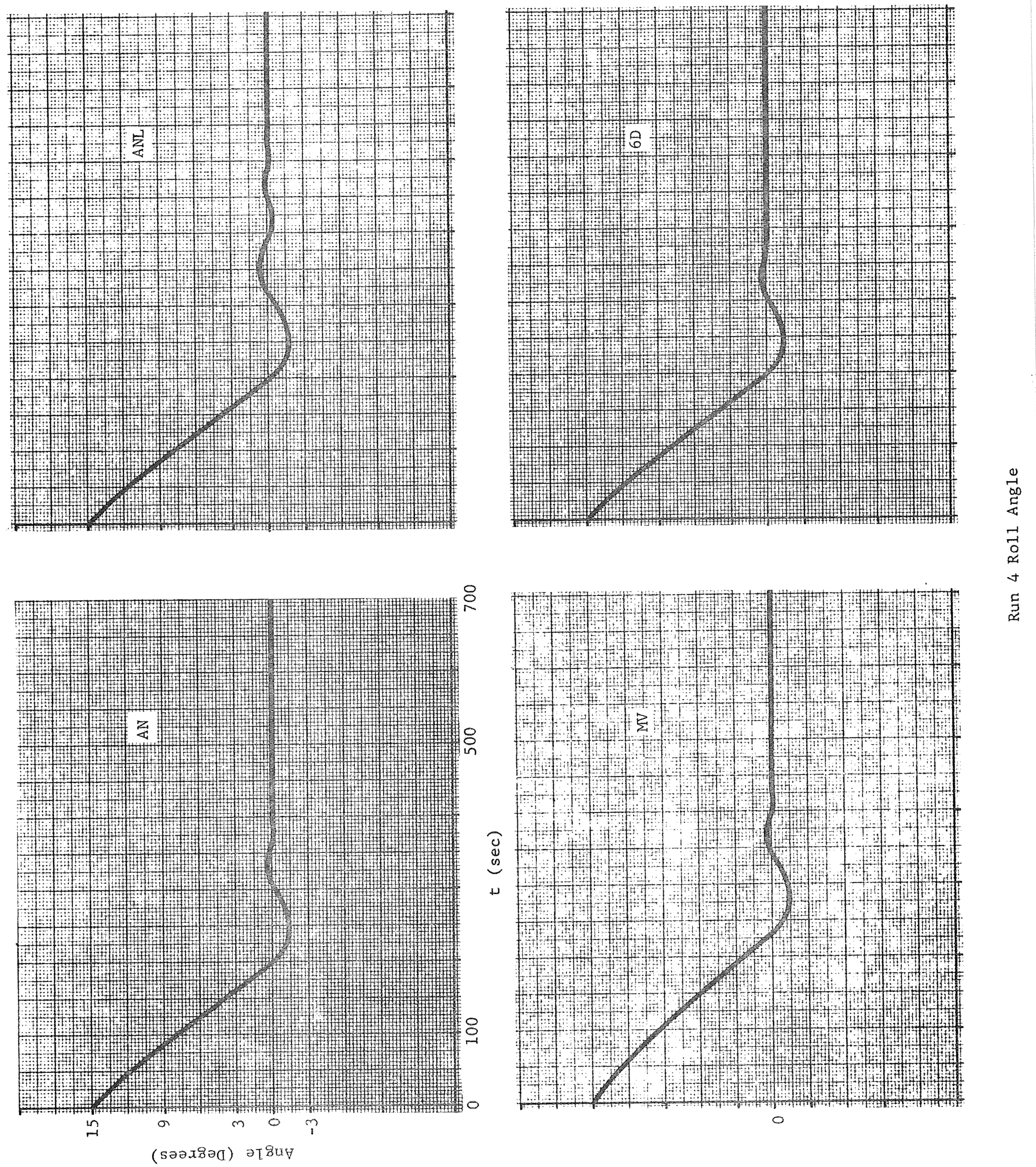


Fig. 11 Control System Simulation Run 4 (Sheet 1 of 6)

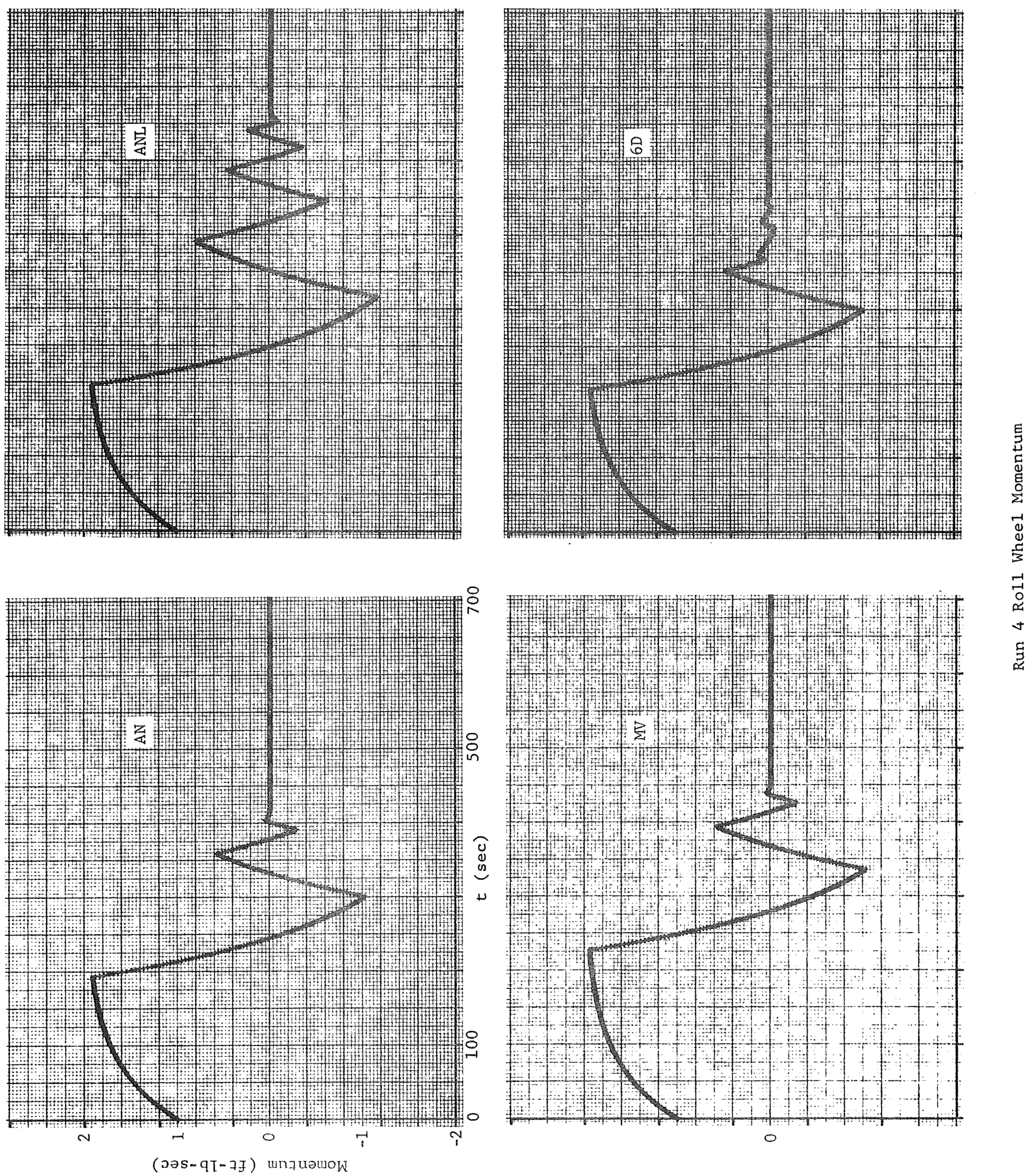
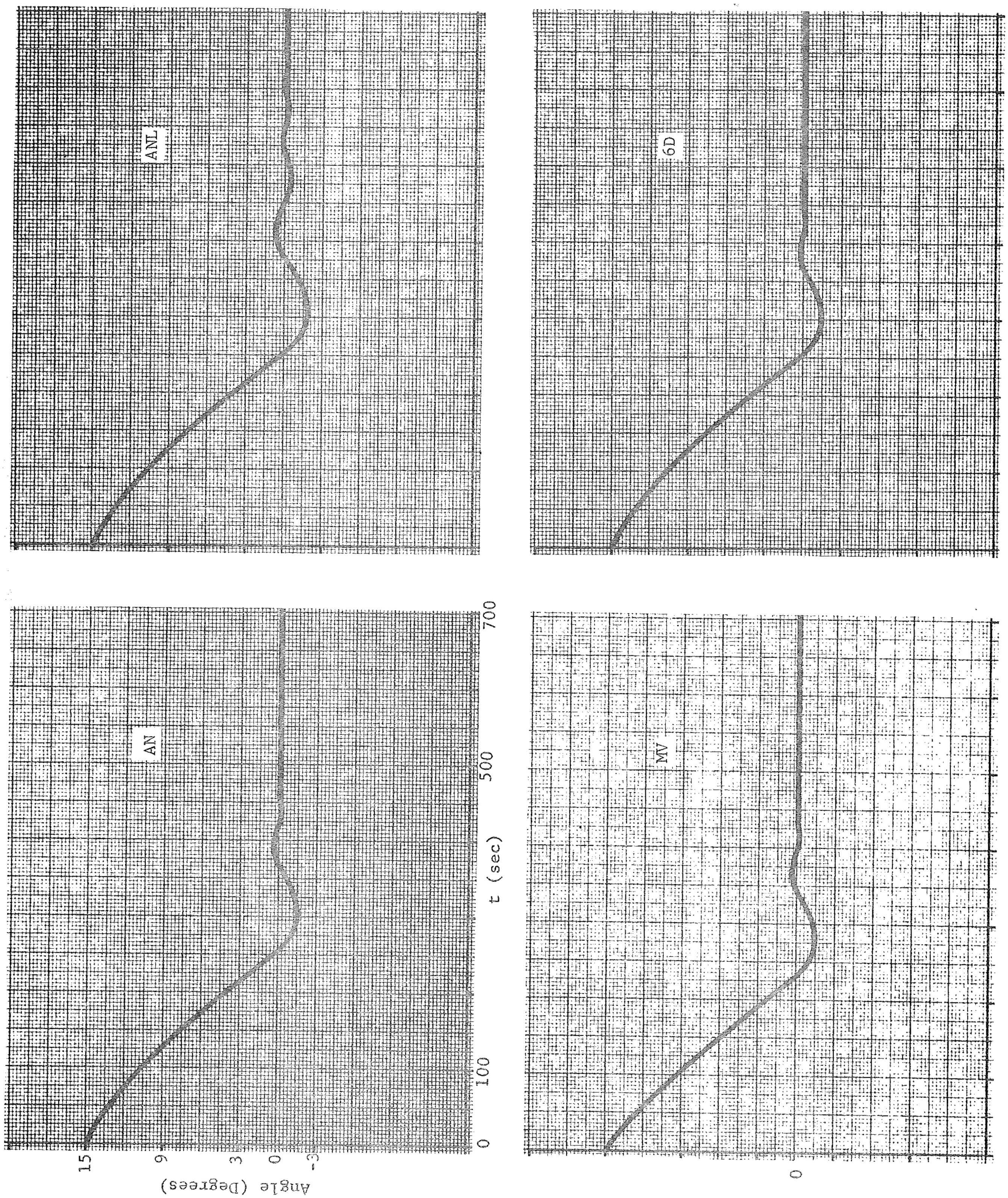


Fig. 11 Control System Simulation Run 4 (Sheet 2 of 6)



Run 4 Pitch Angle

Fig. 11 Control System Simulation Run 4 (Sheet 3 of 6)

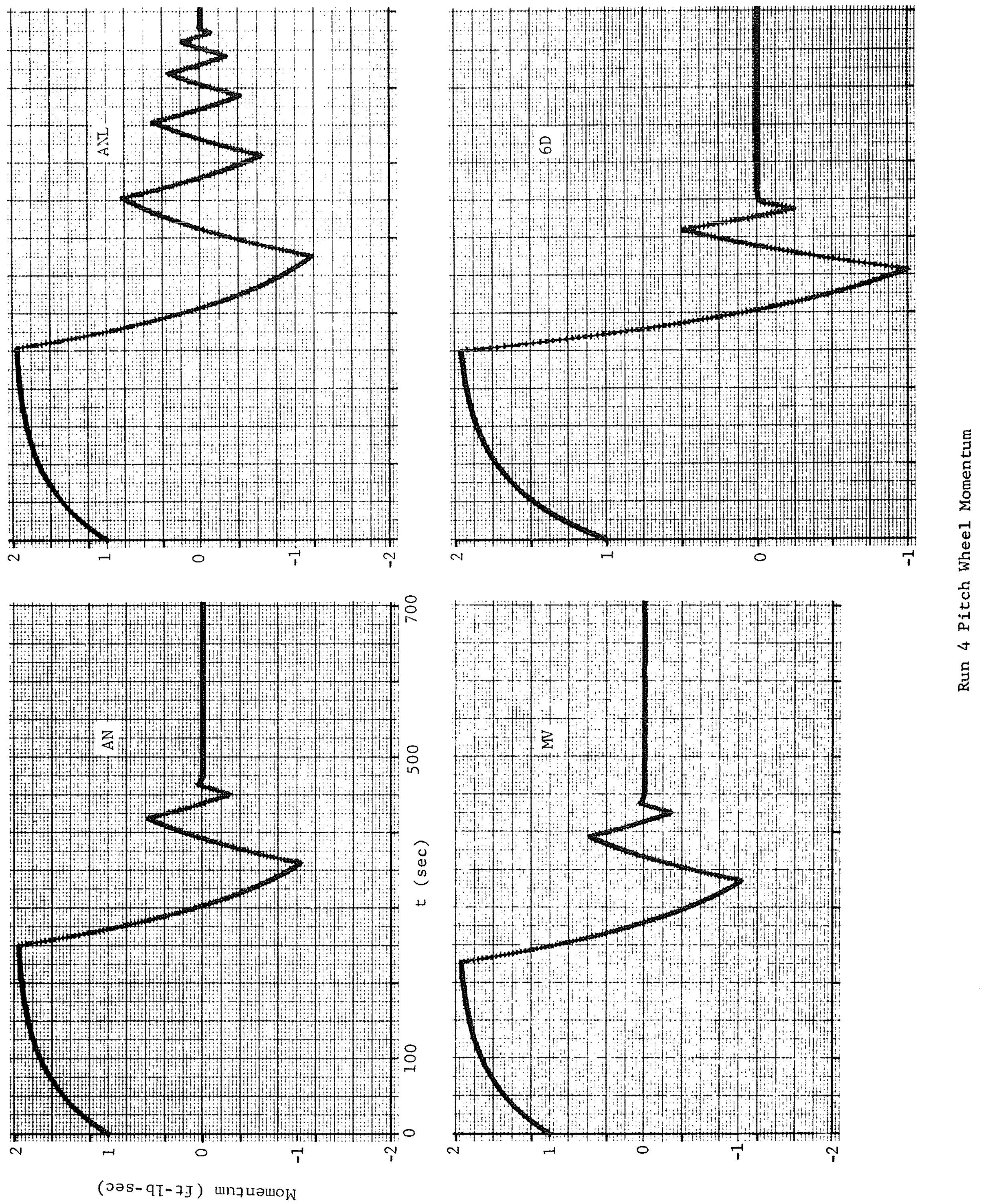


Fig. 11 Control System Simulation Run 4 (Sheet 4 of 6)

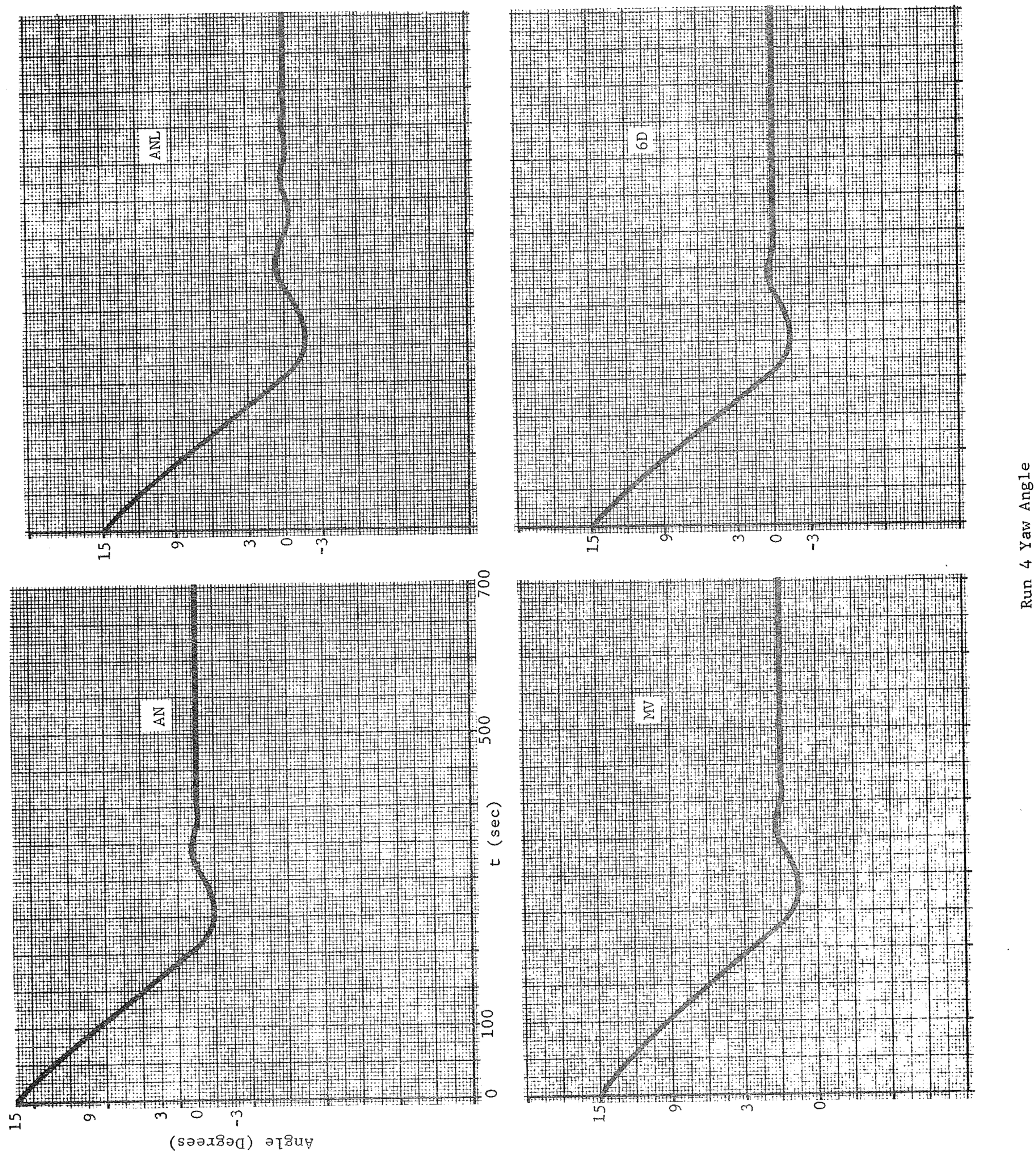


Fig. 11 Control System Simulation Run 4 (Sheet 5 of 6)

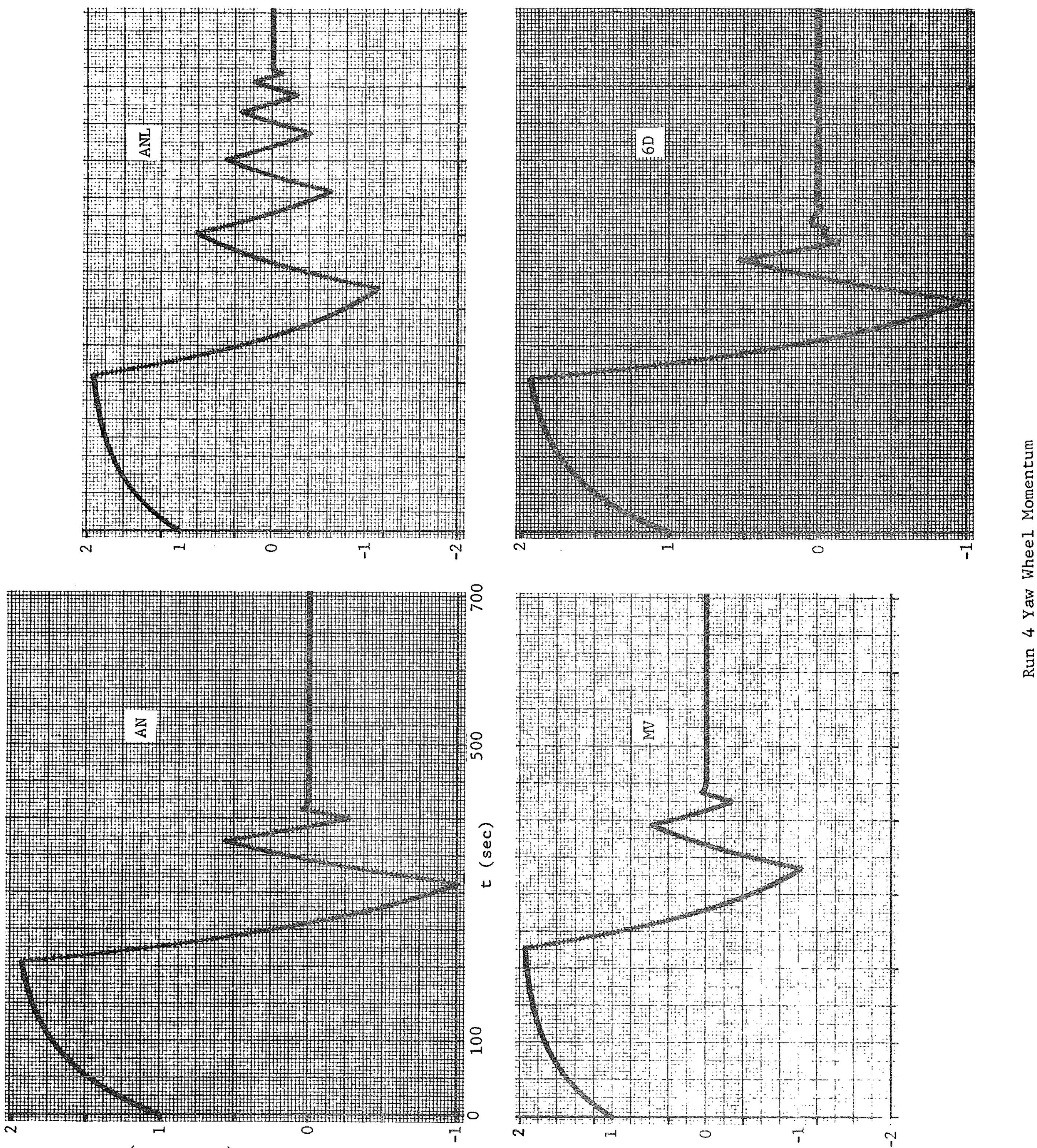


Fig. 11 Control System Simulation Run 4 (Sheet 6 of 6)

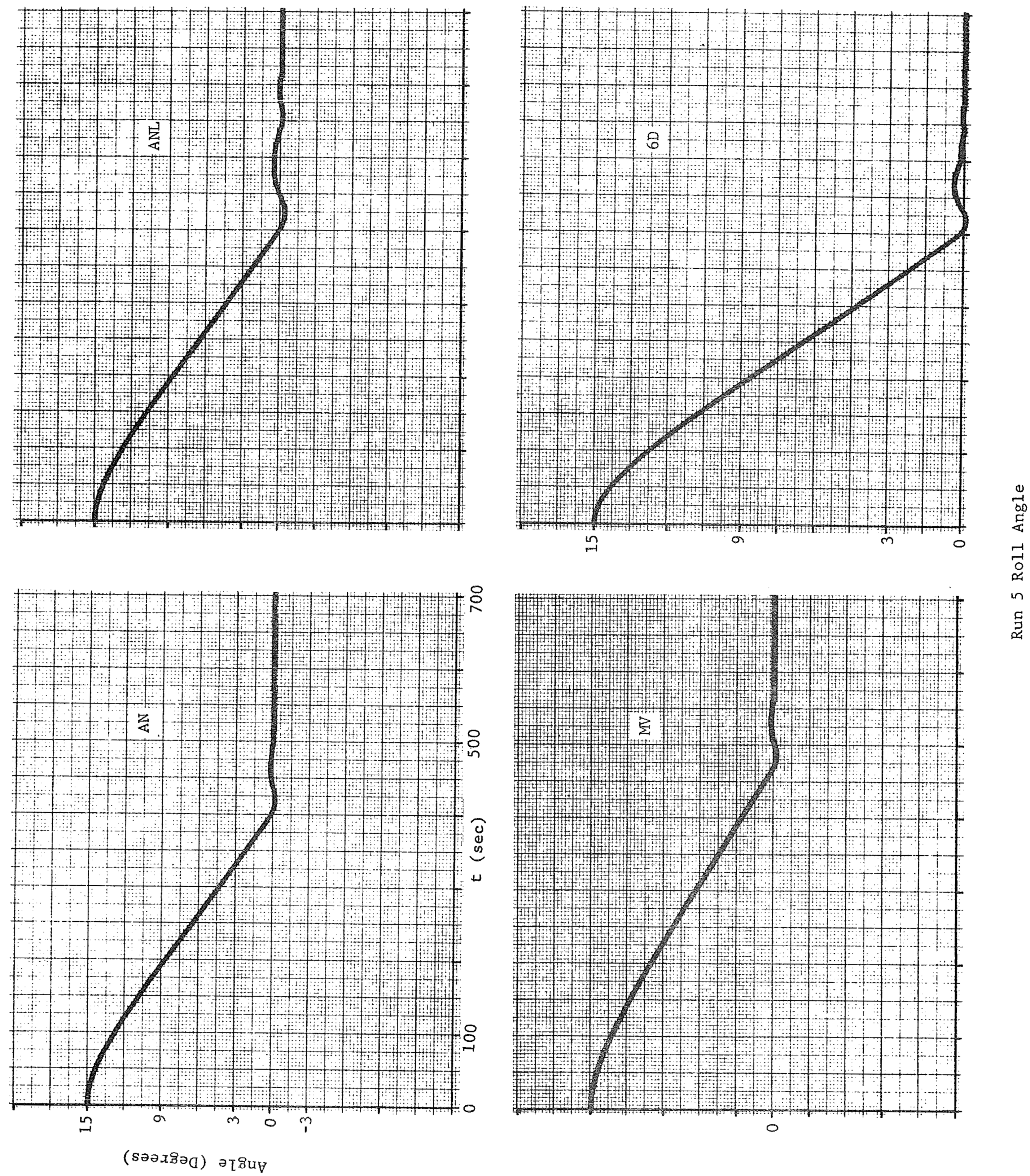


Fig. 12 Control System Simulation Run 5 (Sheet 1 of 6)

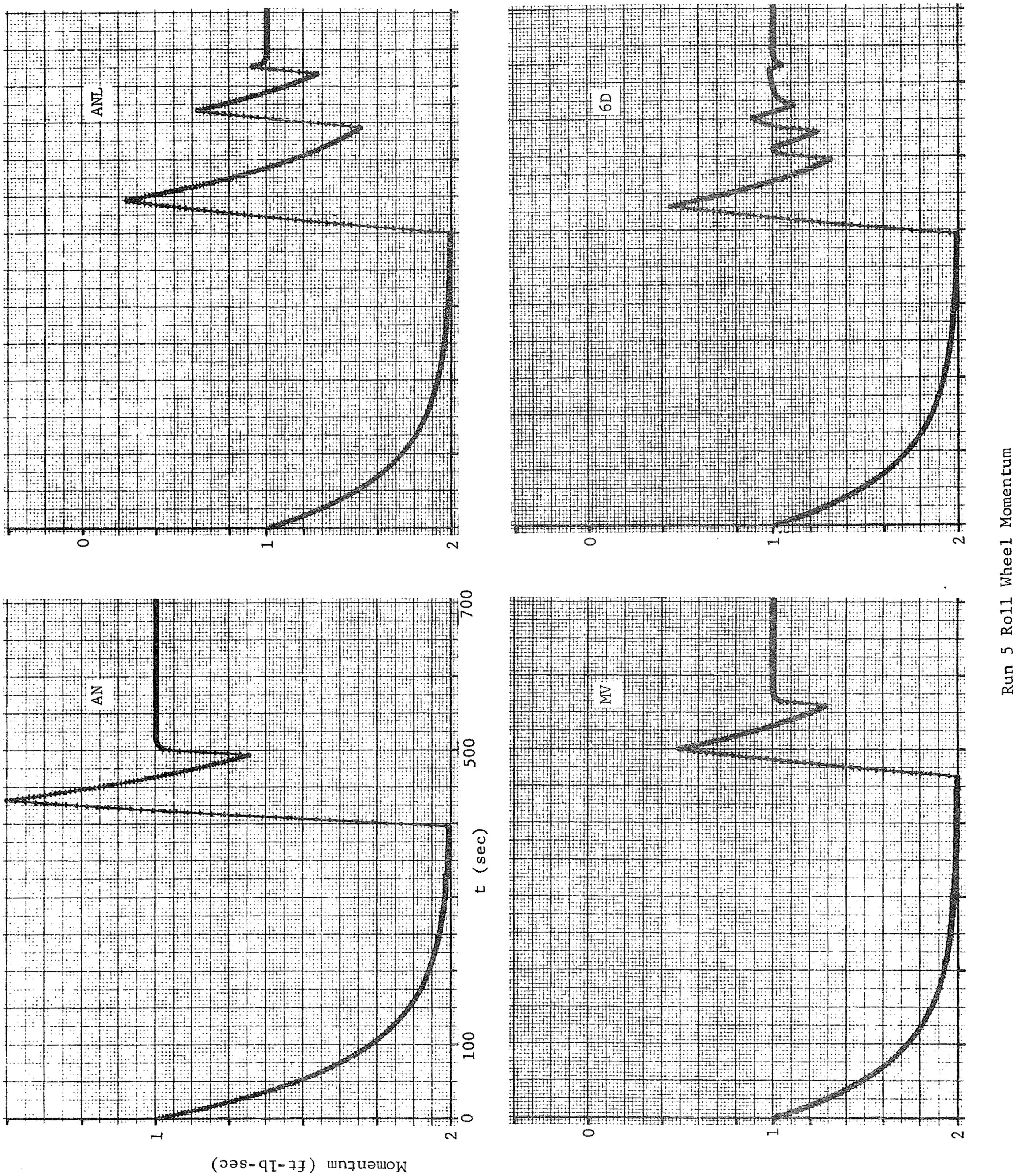


Fig. 12 Control System Simulation Run 5 (Sheet 2 of 6)

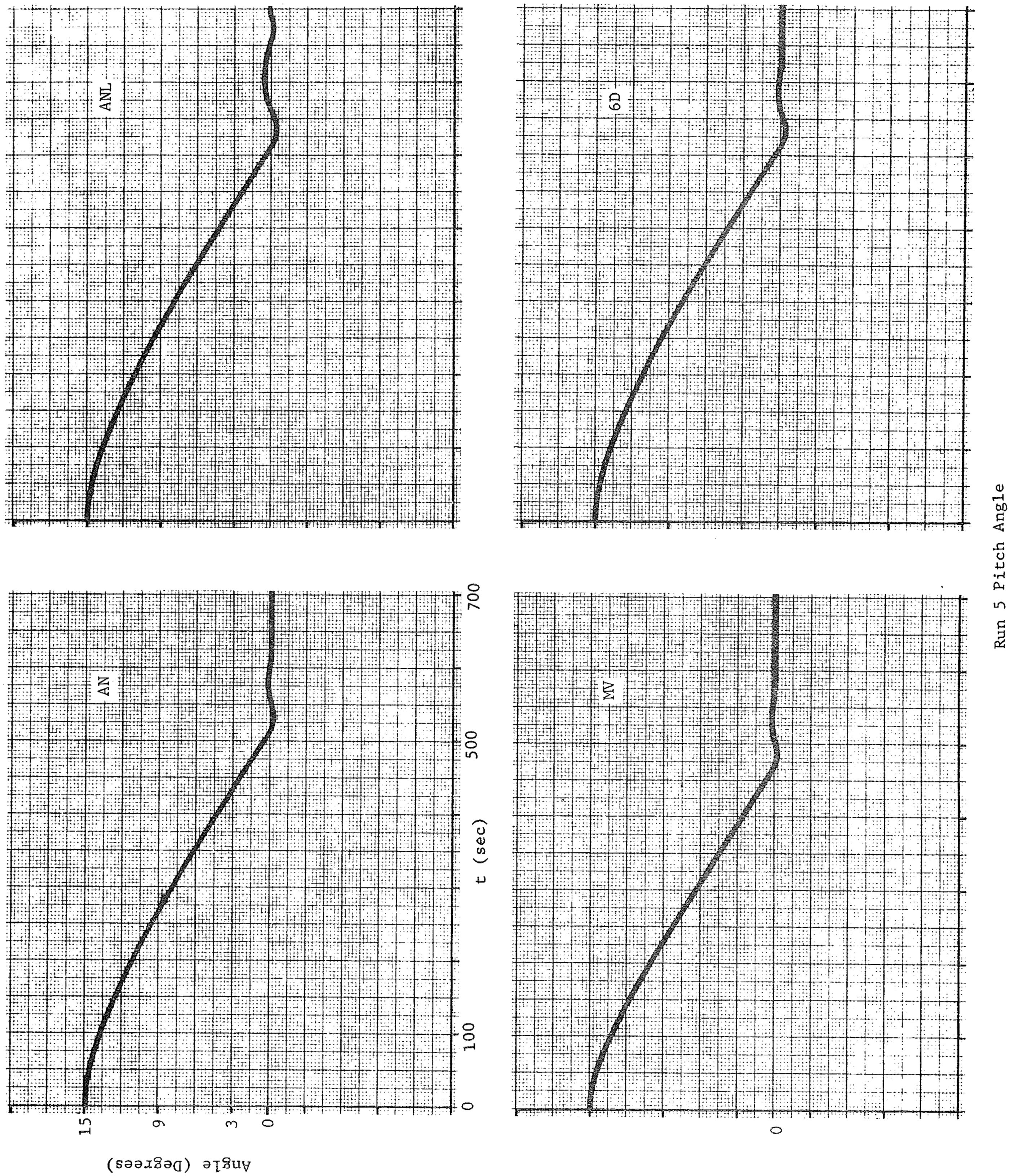


Fig. 12 Control System Simulation Run 5 (Sheet 3 of 6)

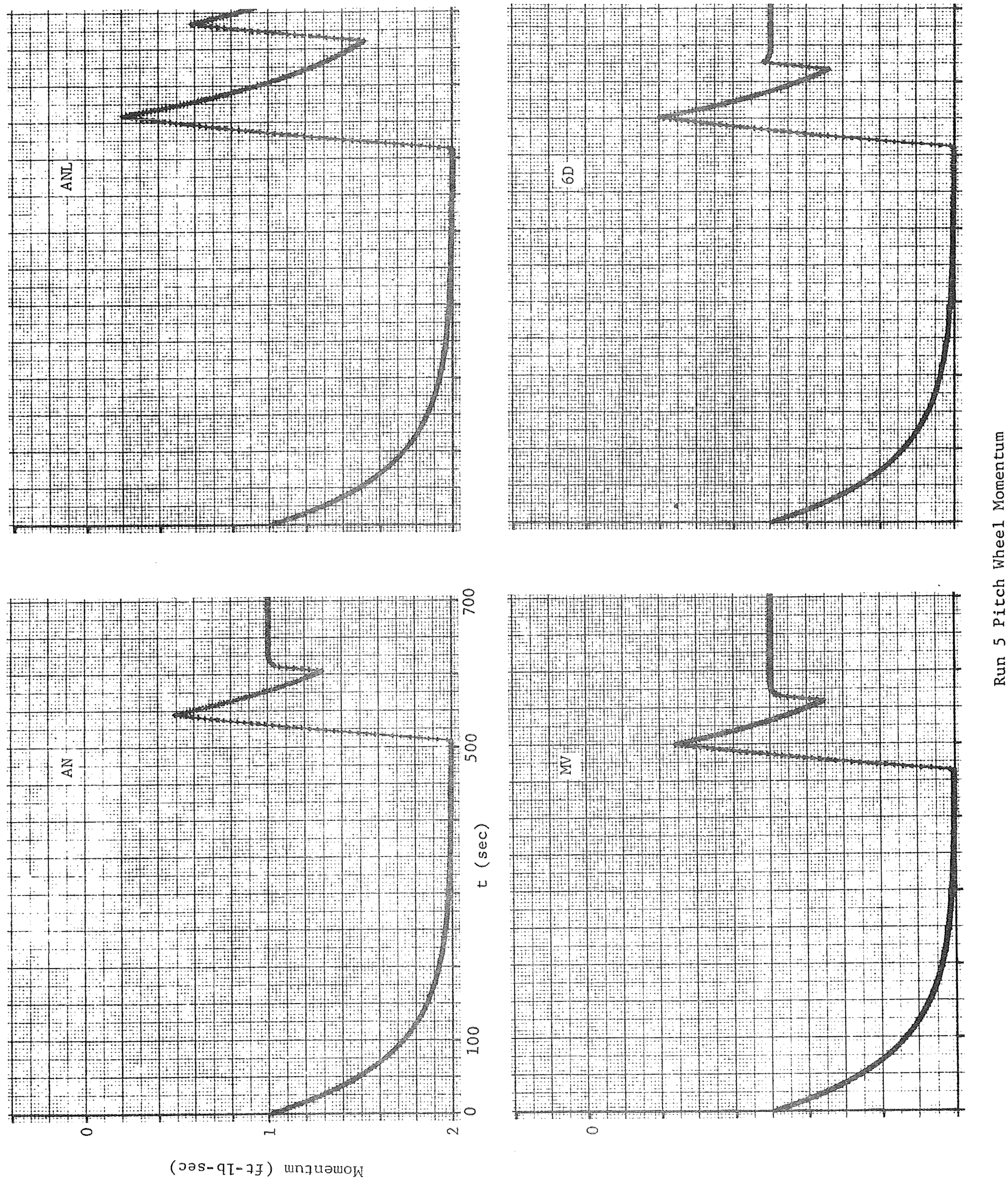


Fig. 12 Control System Simulation Run 5 (Sheet 4 of 6)

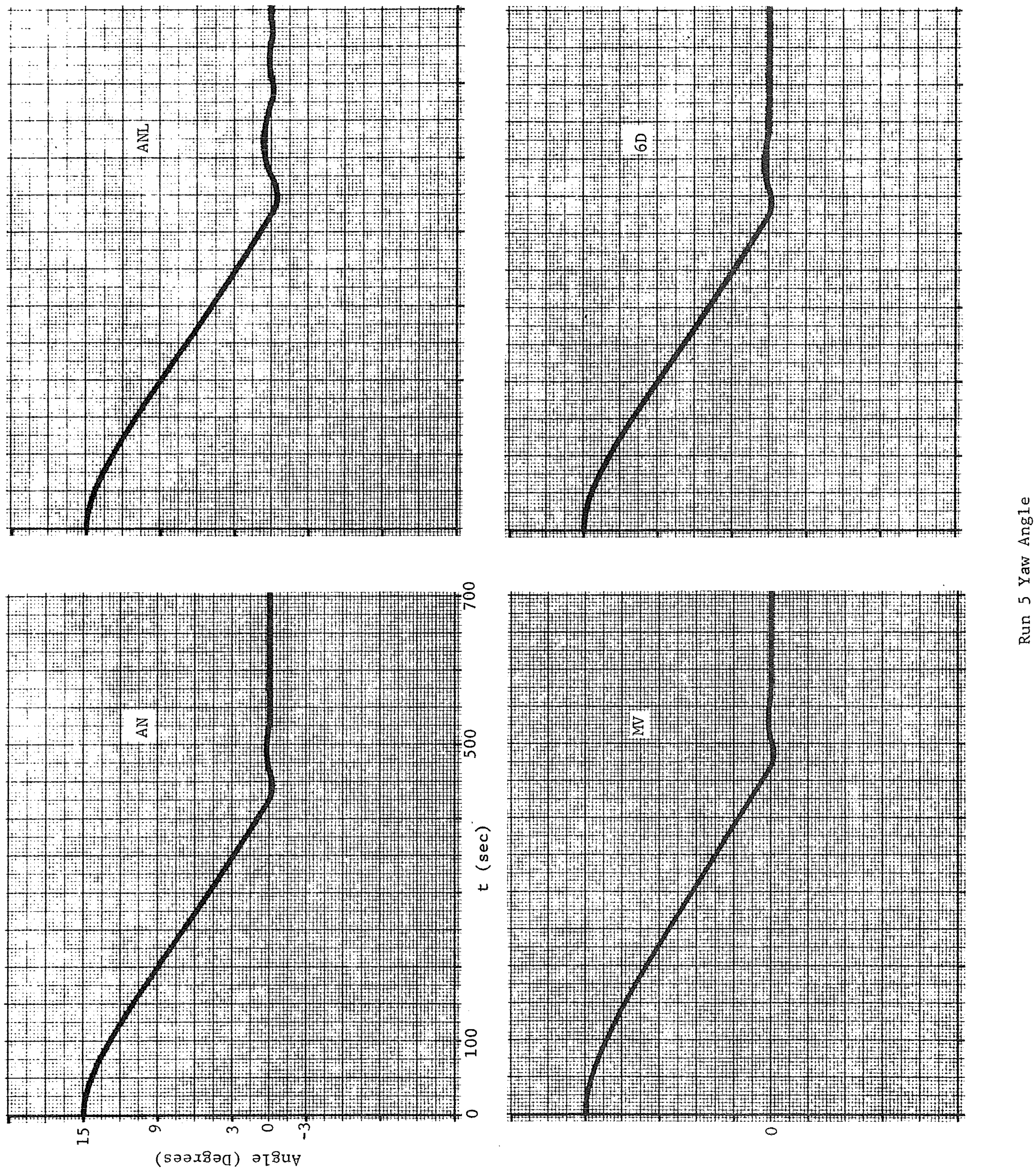


Fig. 12 Control System Simulation Run 5 (Sheet 5 of 6)

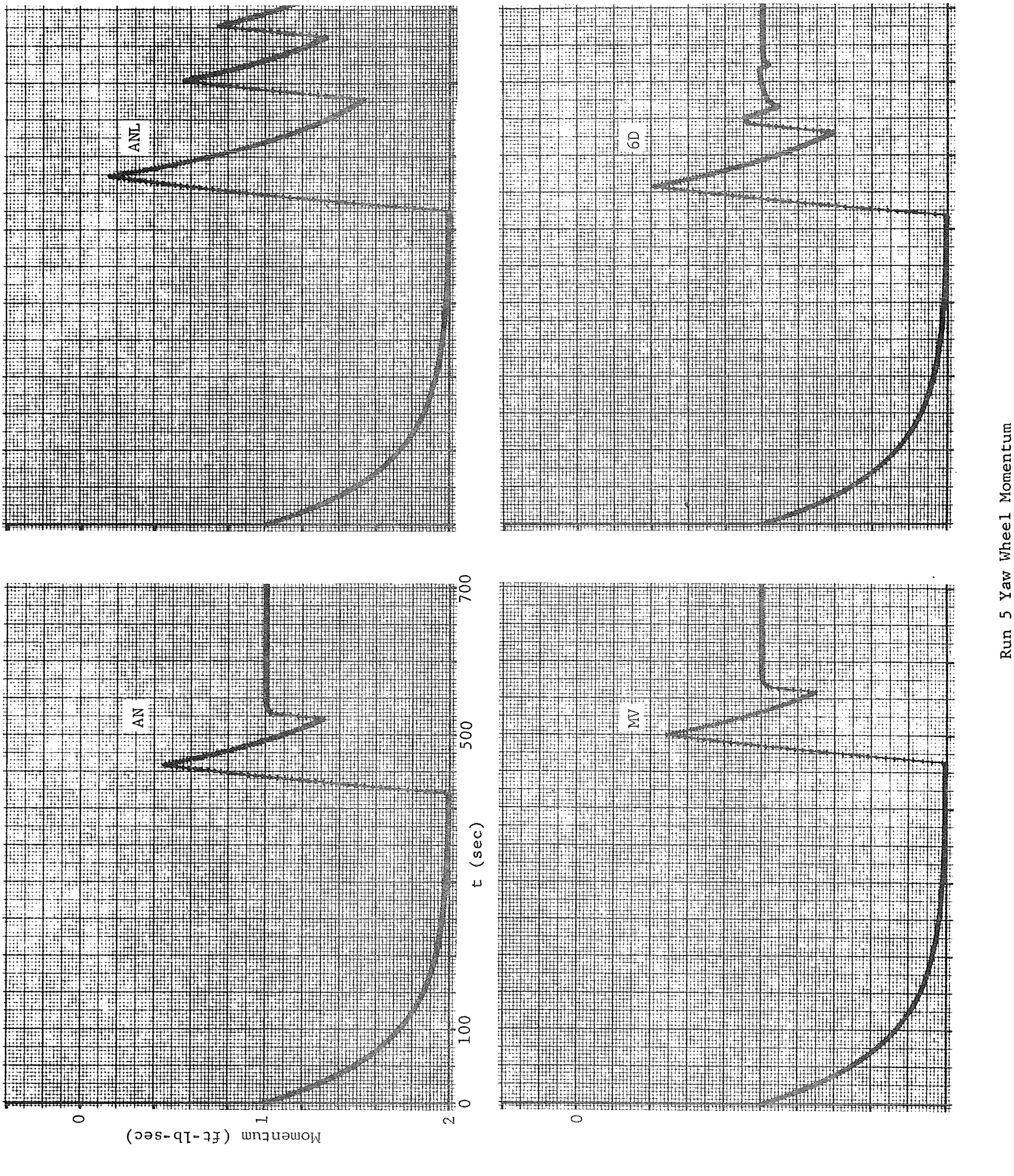


Fig. 12 Control System Simulation Run 5 (Sheet 6 of 6)

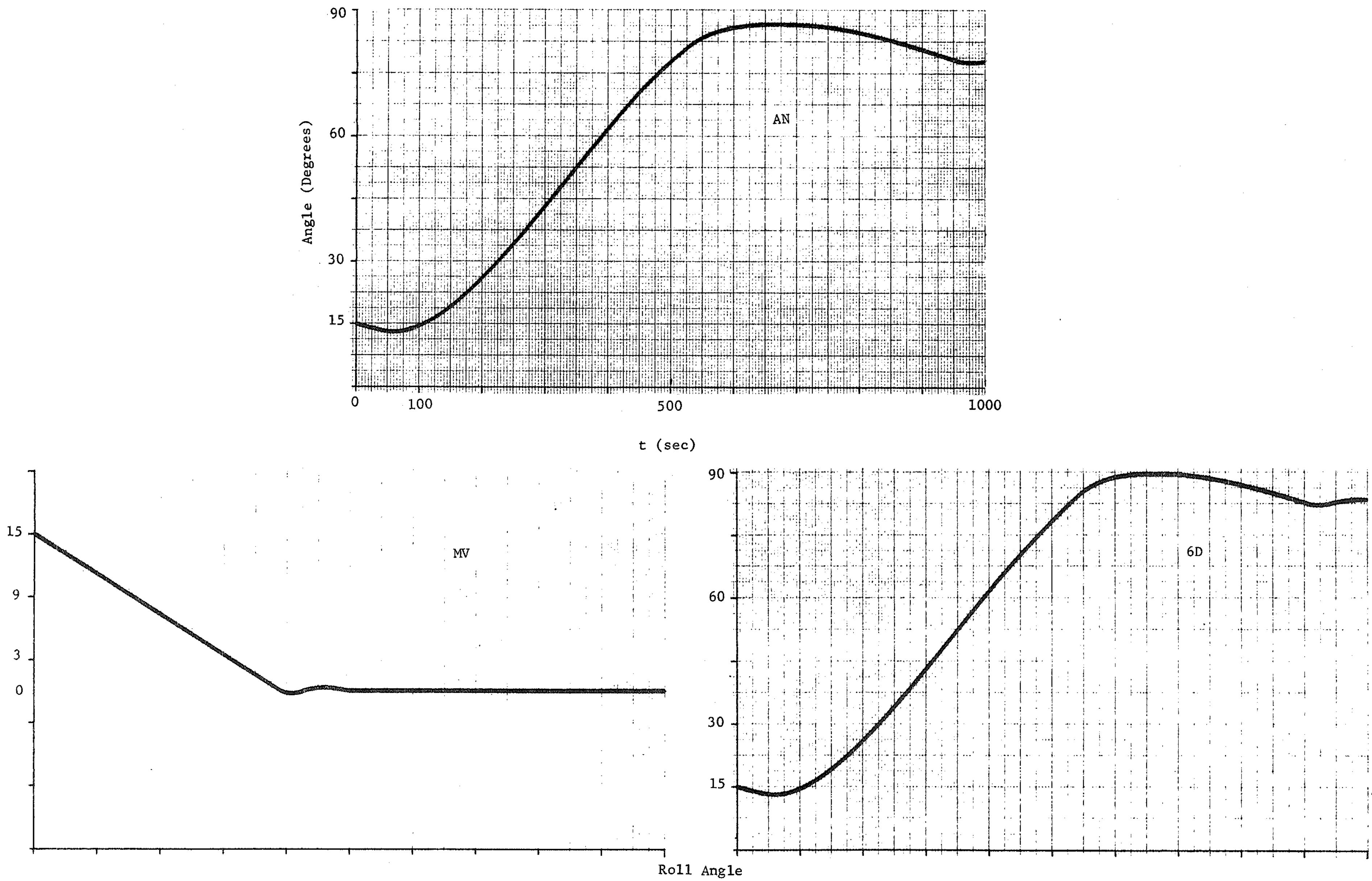
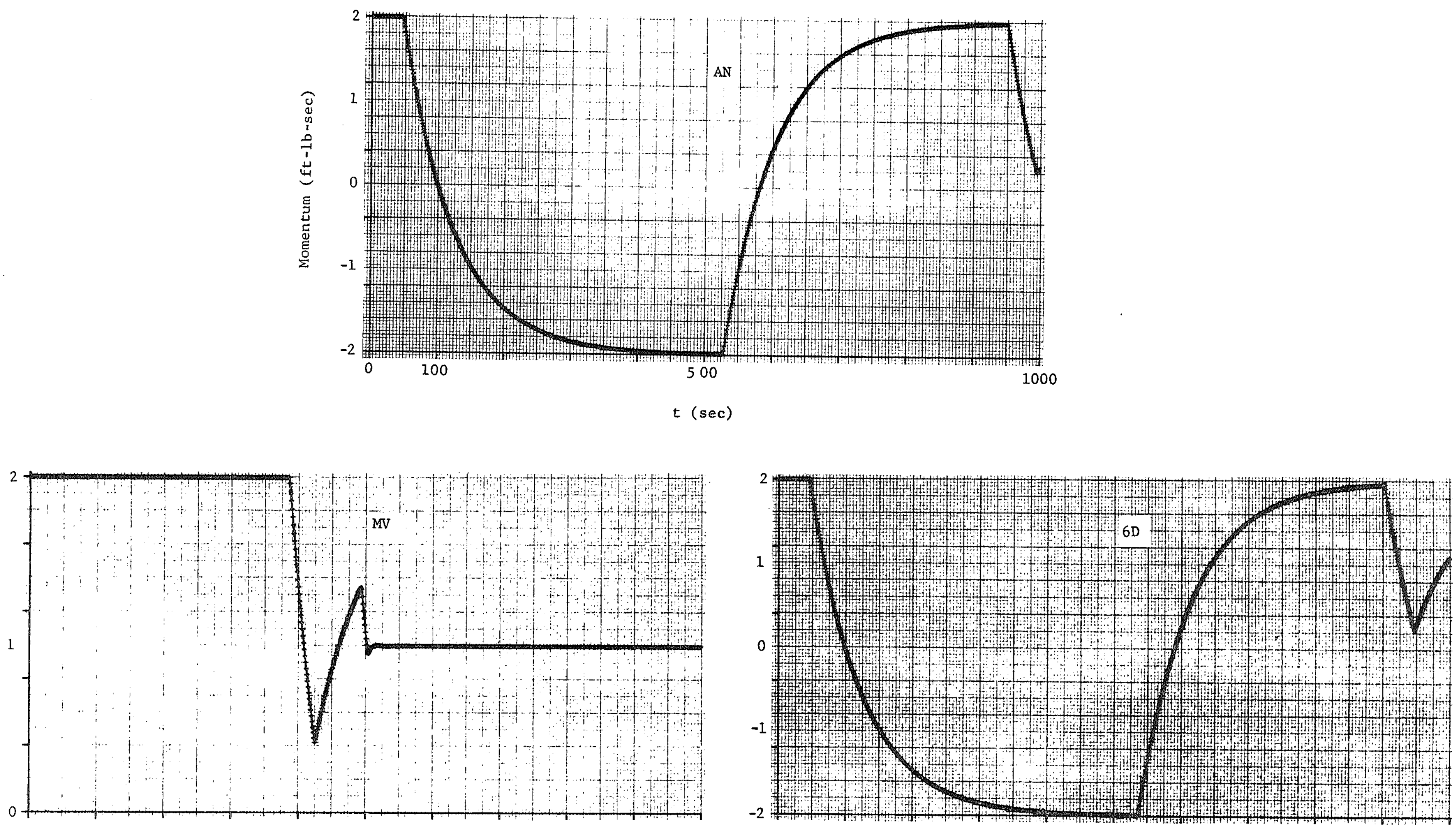
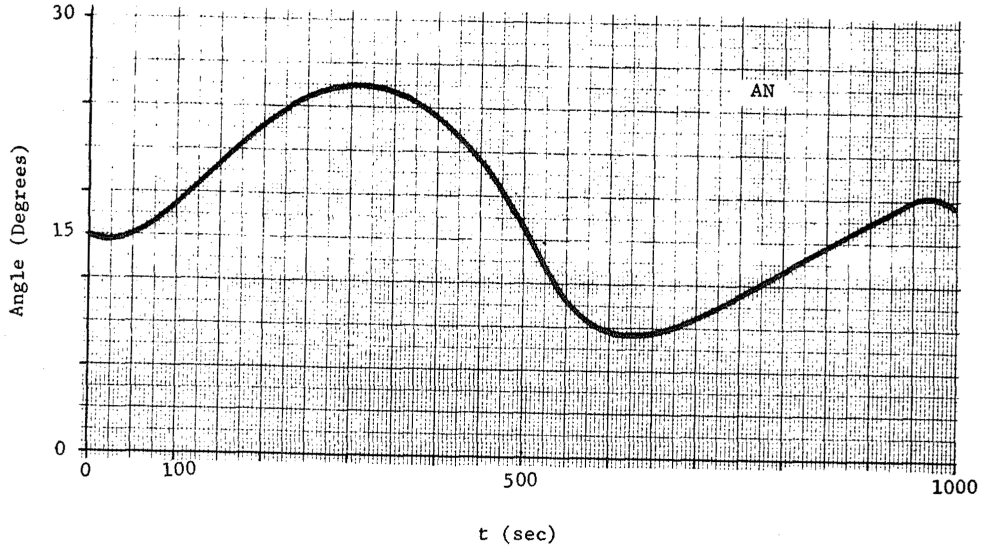


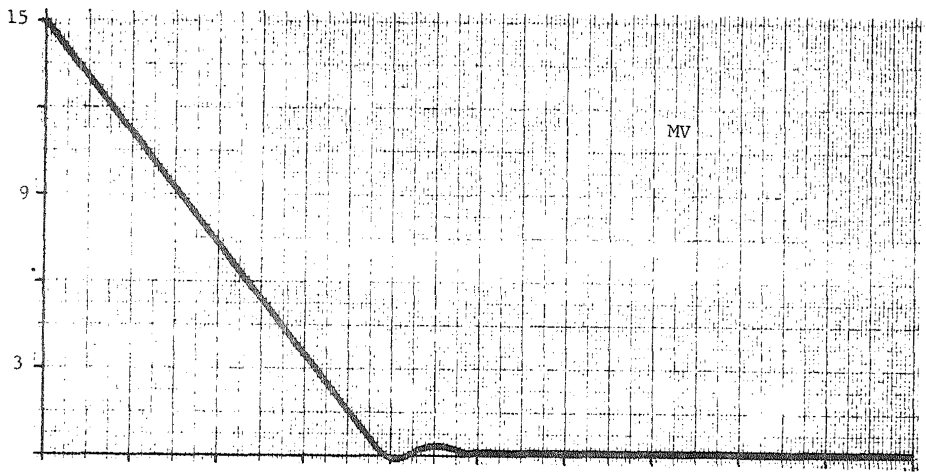
Fig. 13 Control System Simulation Unstable Case (Sheet 1 of 6)

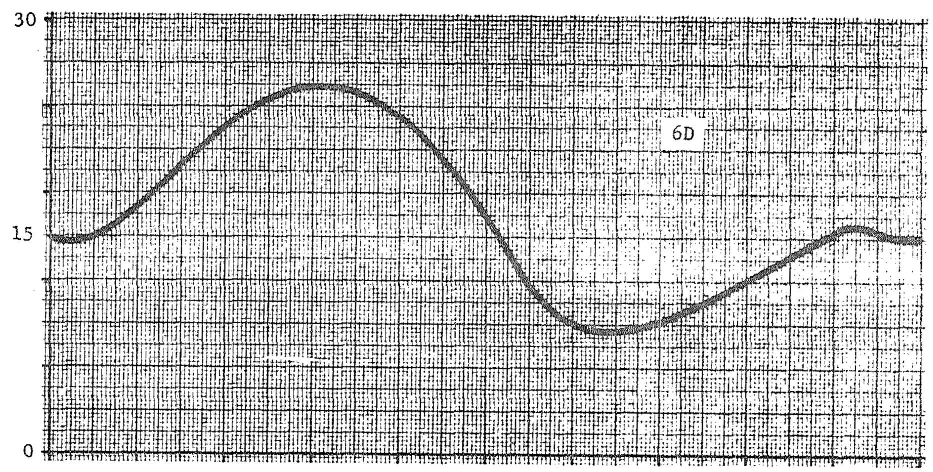


Roll Wheel

Fig. 13 Control System Simulation Unstable Case (Sheet 2 of 6)







Pitch Angle

Fig. 13 Control System Simulation Unstable Case (Sheet 3 of 6)

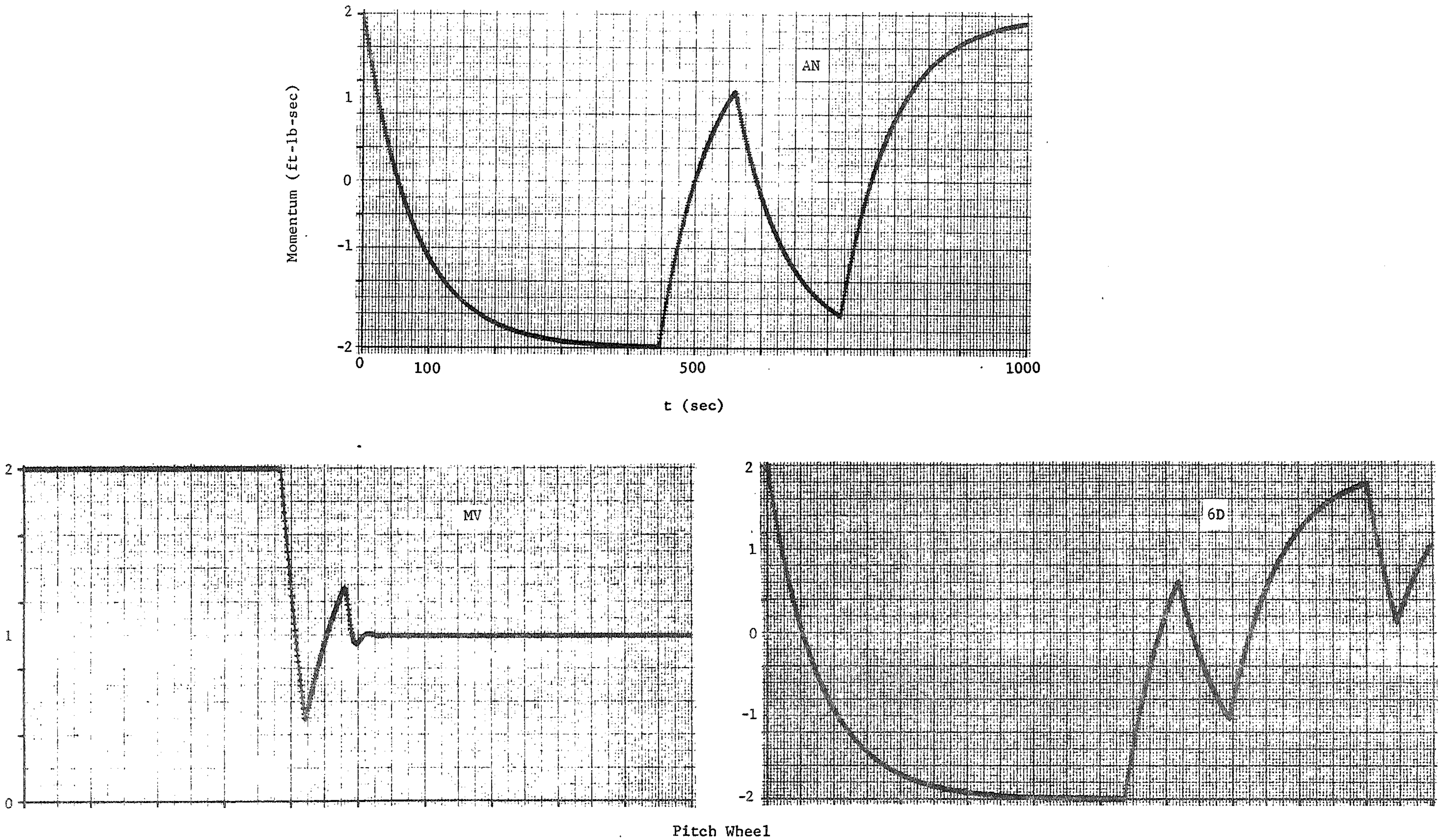


Fig. 13 Control System Simulation Unstable Case (Sheet 4 of 6)

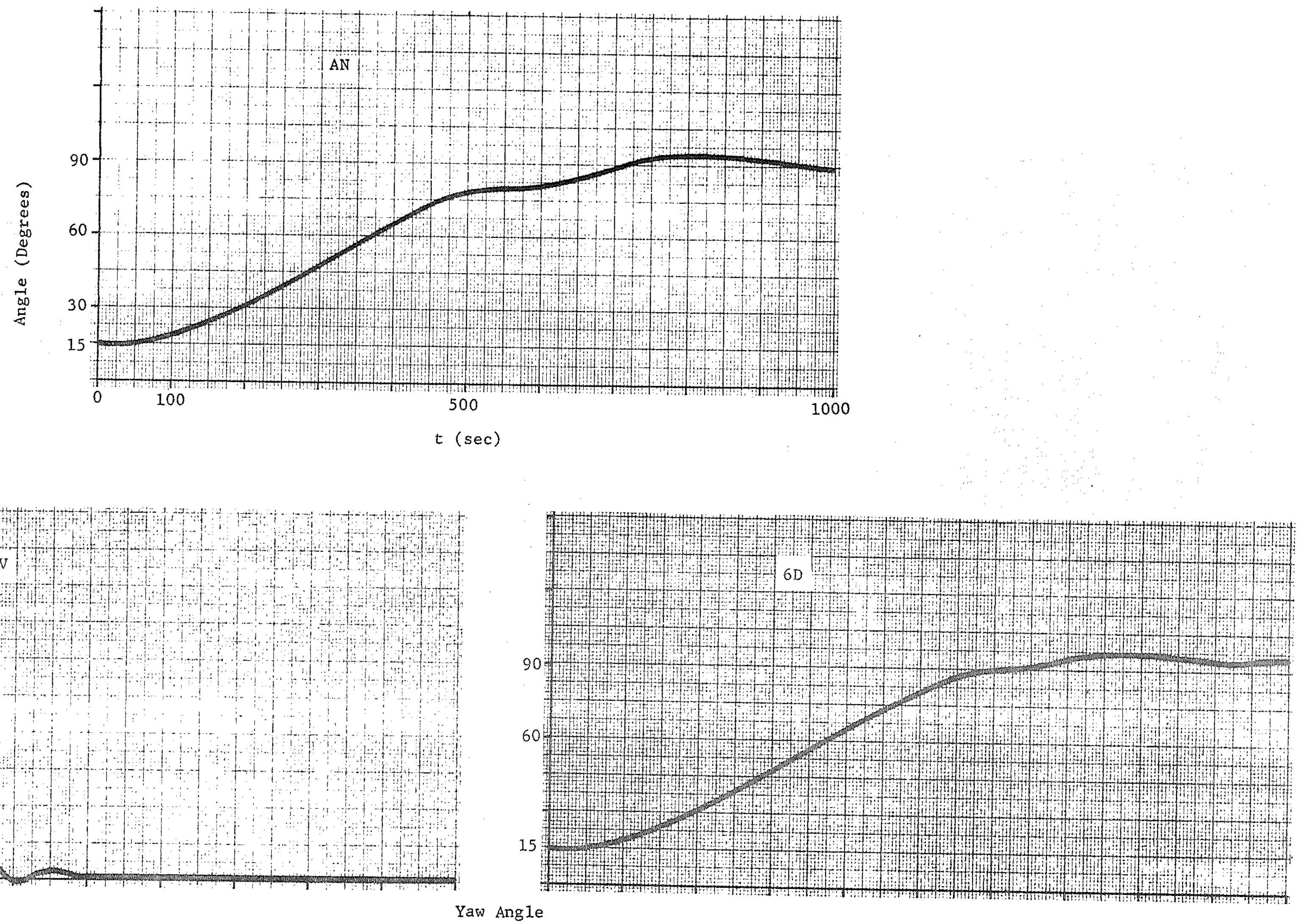
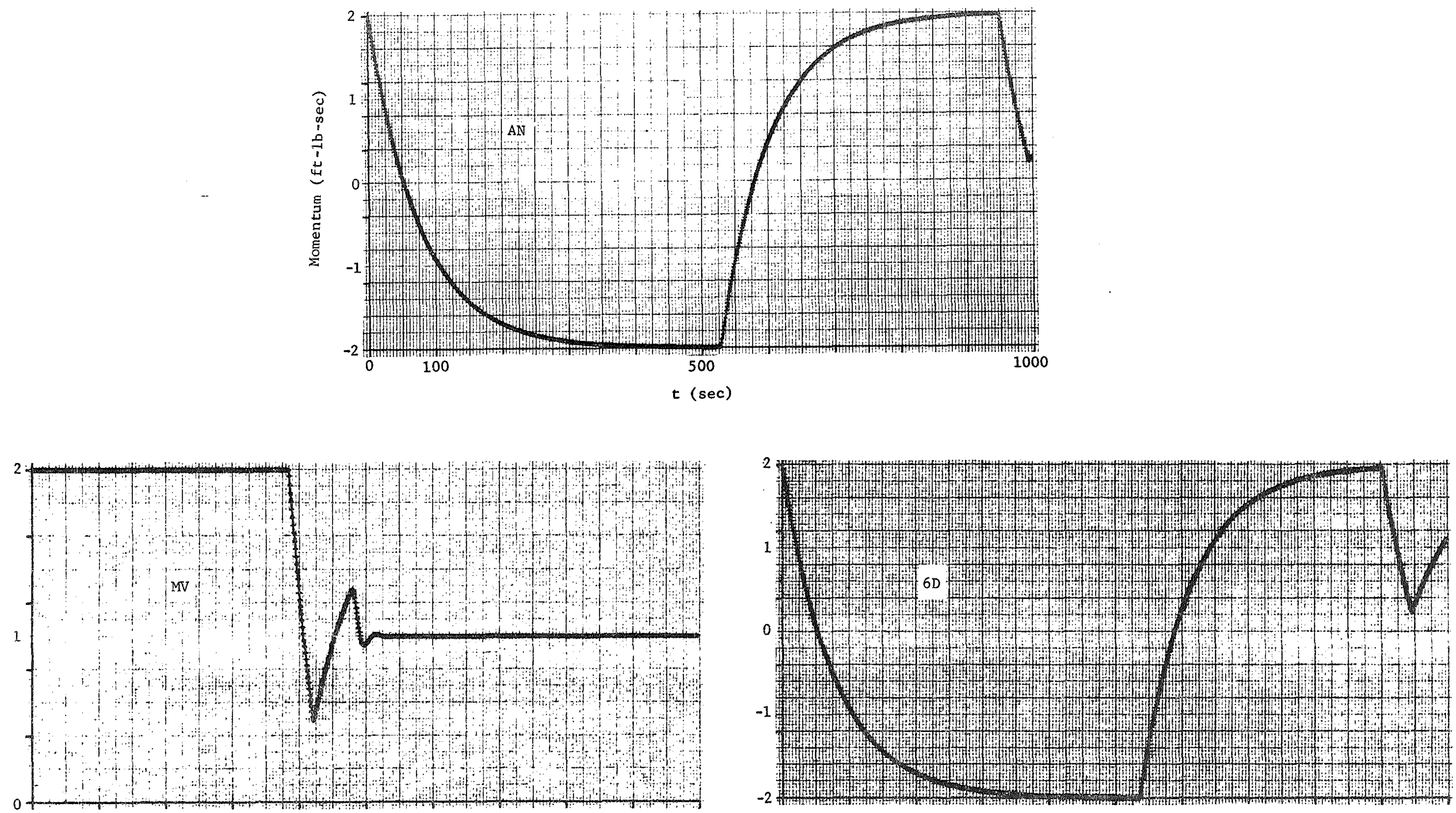


Fig. 13 Control System Simulation Unstable Case (Sheet 5 of 6)



Yaw Wheel

Fig. 13 Control System Simulation Unstable Case (Sheet 6 of 6)

III. STABILITY ANALYSIS OF A SIMPLIFIED MODEL

The stability analysis of simplified models (motor saturation only) of the "paired-Tracker" control system serves two purposes in this study. First, if the simplified model can be proven to be globally asymptotically stable, then the domain of attraction is the whole space and, the algorithm could be tested on this model to determine its effectiveness by comparing the estimate to the known domain. The algorithm could then be tested with the more exact model to determine the effects of the other nonlinearities. Secondly, if the simplified model is globally asymptotically stable, the existence of a Luré-Liapunov function (quadratic form plus integral of the nonlinear terms) is guaranteed, and this could be used in the algorithm with the more exact model to see if it produces a better estimate of the domain of attraction than the optimum quadratic form. It is for these reasons that we carry out the stability analysis of the simplified model.

In Section II, the models of the "paired-Tracker" coarse pointing mode system were derived for the case in which all nonlinearities except the momentum wheel motor saturation are linearized. This resulted in the system of equations

$$\dot{z} = A^* z + B^* f(v)$$
$$\dot{v} = H^* z + J^* f(v) \qquad (21)$$
$$J^* \equiv 0 \quad .$$

By applying the Laplace transform Eqs. (21) can be written as

$$\underset{\sim}{z}(s) = (sI - A^*)^{-1} B^* \underset{\sim}{f}(\underset{\sim}{v})$$
$$\underset{\sim}{v}(s) = \frac{1}{s} H^* \underset{\sim}{z}(s) \qquad (31)$$

where $\underset{\sim}{z}(s)$ is the Laplace transform of $z(t)$, etc., and thus,

$$\underset{\sim}{v}(s) = \frac{1}{s} H^*(sI - A^*)^{-1} B^* \underset{\sim}{f}(\underset{\sim}{v}) \quad . \tag{32}$$

By defining the transfer function from $f(v)$ to $(-v)$ as $\underset{\sim}{W}(s)$, viz.,

$$\underset{\sim}{W}(s) = \frac{-1}{s} H^*(sI - A^*)^{-1} B^* \quad , \tag{33}$$

we obtain the block diagram of Fig. 14.

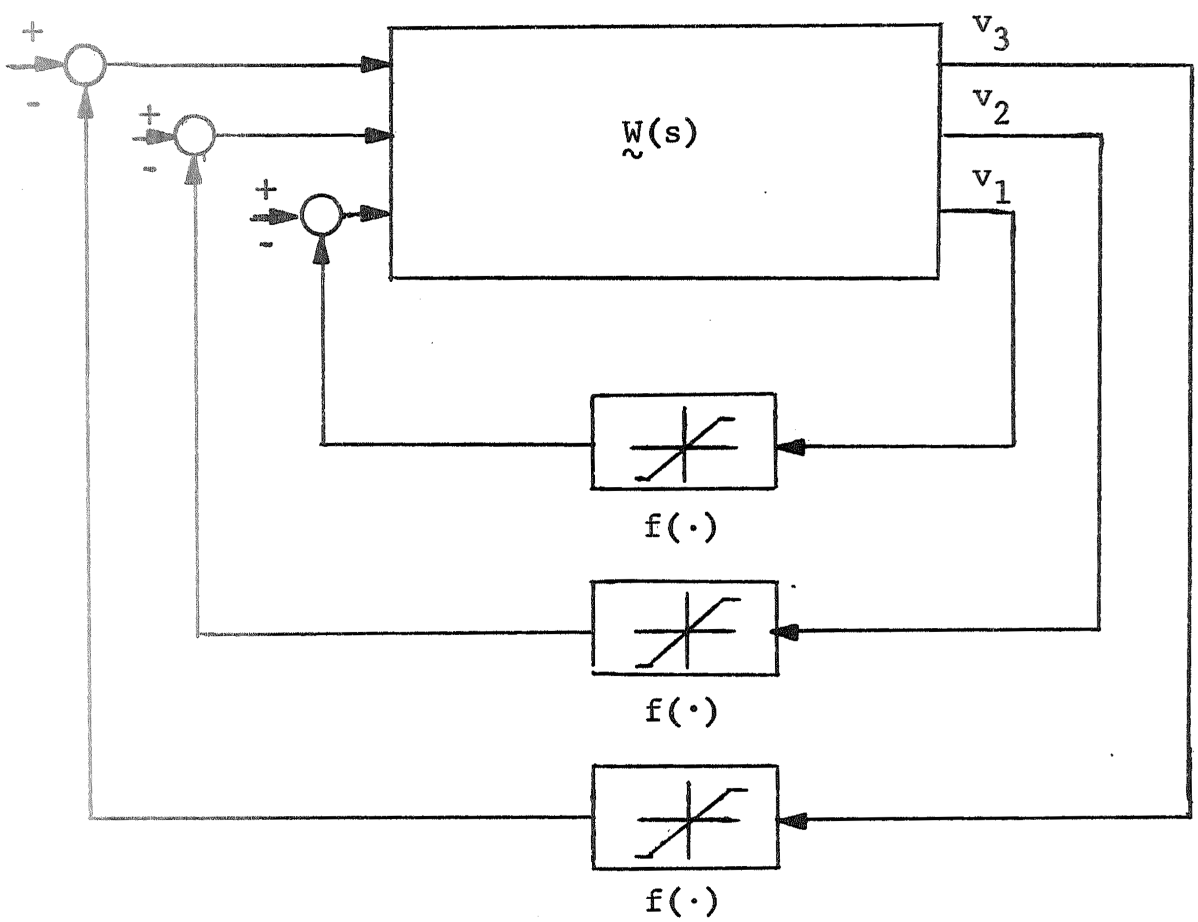


Fig. 14 Simplified System Model for Popov Analysis

A somewhat tedious calculation results in the exact form of $\underset{\sim}{W}(s)$, viz.,

$$\underset{\sim}{W}(s) = \underset{\sim}{w}(s)W \tag{34}$$

where

$$\underset{\sim}{w}(s) = +\frac{K_m K_c}{I}\left[\frac{(\tau_1 + \tau_2)s + 1}{s(\tau_m s + 1)(\tau_2 s + 1)}\right]$$

and (35)

$$W = \begin{bmatrix} 1 & -t\beta_{1c}c\gamma_{1c} & t\beta_{1c}s\gamma_{1c} \\ 0 & d_{12}s(\gamma_{1c} - \gamma_{2c}) & 0 \\ 0 & 0 & d_{12}s(\gamma_{1c} - \gamma_{2c}) \end{bmatrix} .$$

Thus, in the case $t\beta_{1c} = 0$, it is clear that the problem reduces to three single channel problems of the Popov type. In any event, the pitch and yaw channels are always completely decoupled in this model and can always be treated as separate Popov problems. Unfortunately, in each case there is a pole at the origin and this requires using a special form of the Popov theorem (Ref. 7).

<u>THEOREM</u> (Popov)

For the particular case of a system to be absolutely stable in the sector $[\epsilon, K]$ (where $\epsilon > 0$ is an arbitrarily small number), it is sufficient that there exist a finite real number q such that for all $\omega \geq 0$

$$\mathrm{Re}(1 + i\omega q)W(i\omega) + \frac{1}{K} > 0 \quad , \tag{36}$$

and that the conditions for stability in the limit (i.e., if there is a single pole at the origin then $\lim_{\omega \to 0^+} \mathrm{Im}\, W(i\omega) = -\infty$) are satisfied.

Note that the sector $[\epsilon, K]$ means that $\epsilon v^2 \leq vf(v) \leq Kv^2$, $v \neq 0$, which does not hold for saturation functions when $|v| \to \infty$. One could argue that since in reality the nonlinearity is only known for finite limits on v, it could be continued in an arbitrary fashion beyond the limits within which it is known and thus fit into the sector $[\epsilon, K]$. Although it would seem that the analysis is upset by just a fine mathematical point, the argument above is still not physically satisfying. In any event, let us continue to prove that $\underset{\sim}{W}(s)$ as given in Eqs. (34 and 35), satisfies the theorem.

Let us assume $t\beta_{1c} = 0$, then we have three separate problems, two of which are identical. In particular, if we define

$$\underset{\sim}{W}^{\dagger}(s) = \frac{K_m K_c}{I} d_{12} s(\gamma_{1c} - \gamma_{2c}) \left[\frac{(\tau_1 + \tau_2)s + 1}{s(\tau_m s + 1)(\tau_2 s + 1)} \right] , \tag{37}$$

we can solve all three problems simultaneously by recognizing that setting $d_{12} = \left(s(\gamma_{1c} - \gamma_{2c}) \right)^{-1}$ results in the pitch and yaw channels being identical to the roll channel. Note that for the system treated here $K = 1$.

Let us first examine the stability in the limit requirement, viz.,

$$\lim_{\omega \to 0^+} \operatorname{Im} \underset{\sim}{W}^{*}(i\omega) = \lim_{\omega \to 0^+} \frac{K_m K_c}{I} d_{12} s(\gamma_{1c} - \gamma_{2c}) \frac{1}{i\omega} = -\infty , \tag{38}$$

if

$$\frac{K_m K_c}{I} d_{12} s(\gamma_{1c} - \gamma_{2c}) > 0 .$$

Since K_m, K_c, and I are all positive, we have stability in the limit if $d_{12}s(\gamma_{1c} - \gamma_{2c}) > 0$, which is a design requirement, in fact $d_{12} = 2.0 \operatorname{sgn}(\gamma_{1c} - \gamma_{2c})$.

Now let us examine Eq. (36), i.e.,

$$\operatorname{Re}(1 + i\omega q)\left[\frac{K_m K_c}{I} d_{12}s(\gamma_{1c} - \gamma_{2c})\left(\frac{(\tau_1 + \tau_2)i\omega + 1}{i\omega(\tau_m i\omega + 1)(\tau_2 i\omega + 1)}\right)\right] + \frac{1}{K} > 0, \tag{39}$$

which can be rewritten as

$$\frac{K_m K_c}{I} d_{12}s(\gamma_{1c} - \gamma_{2c})\operatorname{Re}\left[\frac{(i\omega q + 1)\left((\tau_1 + \tau_2)i\omega + 1\right)}{i\omega(\tau_m i\omega + 1)(\tau_2 i\omega + 1)}\right] + \frac{1}{K} > 0 \quad . \tag{40}$$

Recall that a ratio of two polynomials whose roots are negative, real, and interlace, and whose numerator degree is one less than the denominator degree, is a positive real function. Thus, if we choose $q > \tau_m$, since $\tau_2 < (\tau_1 + \tau_2) < \tau_m$, the real part of the transfer function in brackets will be positive for all ω, and the system will be absolutely stable for all $K > 0$.

We have proven that each channel of the "paired-Tracker" system is absolutely stable for all $K > 0$ and all $f(v)$ if $d_{12}s(\gamma_{1c} - \gamma_{2c}) > 0$ and $\epsilon v^2 \leq vf(v) \leq Kv^2$. Unfortunately, this does not admit the saturation function. Note that

$$f(v) = \operatorname{sat}\left(v + \frac{Ih^o}{K_m}\right) - \frac{Ih^o}{K_m} \quad .$$

Thus the $[\epsilon, K]$ sector requirement is satisfied for finite v only when Ih^o/K_m is less than the saturation level, or the initial momentum Ih^o in that channel is less than the wheel capacity. We have also proven that the entire system is absolutely stable under the same conditions if $t\beta_{1c} = 0$ since the three loops are uncoupled.

Attempts were made to prove the absolute stability of the coupled system via the methods of Moore and Anderson (Ref. 8), Sandberg (Ref. 9) and the very inclusive results of Yakubovich (Ref. 10). In all cases the pole at the origin caused the required conditions to be violated. The reason for the difficulty becomes apparent in the uncoupled case. In order to have absolute stability in the sector $[0,K]$, it is necessary that the system be asymptotically stable for all $f(v) = c_1 v$, where $0 \le c_1 \le K$; however, because of the pole at the origin, the system is only stable for $c_1 = 0$.

The conclusions of this analysis are that: 1) for the uncoupled simplified system ($t\beta_{1c} = 0$) the analysis implies that the system model with saturation only is asymptotically stable for all finite values of the initial conditions, when the initial total vehicle momentum is less than the wheel capacity; 2) for the coupled simplified system, the theory is not sufficiently developed to treat this system successfully. As a result, our plan to use the simplified system as a test of the algorithm's effectiveness and to use the resulting Luré-Liapunov function to obtain improved estimates of the domain of attraction are not fulfilled because of the pole at the origin in the transfer function and the present state of analysis techniques for systems with multiple nonlinearities.

IV. NUMERICAL TECHNIQUE FOR ESTIMATING THE DOMAIN OF ATTRACTION

Picking a Q-Matrix

Theory

A review of the theory of how and why a Q-matrix is chosen so that it results in a volume estimate of the domain of attraction is presented in this section.

Given the set of differential equations partitioned to be of the form

$$\dot{x} = Ax + f(x) \quad , \tag{41}$$

where A is a stable matrix and $f(x)$ contains terms of $O(x^2)$ and higher, we define a Liapunov function

$$V = x^T P x \quad , \tag{42}$$

such that the matrix P is a positive definite matrix which is insured by solving the Liapunov equation

$$A^T P + PA = - Q \tag{43}$$

given a positive definite Q-matrix. In general, if Q is of order n, then there are $n(n + 1)/2$ independent variables that describe Q and are bounded as follows (see Ref. 11):

$$0 < \lambda_i < \infty \qquad i = 1, 2, \ldots, n$$

$$-\frac{\pi}{2} \leq \theta_j \leq \frac{\pi}{2} \qquad j = 1, 2, \ldots, (n-1)(n-2)/2$$

$$-\pi \leq \phi_k < \pi \qquad k = 1, 2, \ldots, (n-1) \quad .$$

The λ's are the eigenvalues of Q, while the θ's and ϕ's are rotation components of the Q-matrix and essentially orient the

Q-matrix in the $n(n + 1)/2$ space from which it is generated. Thus, $n(n + 1)/2$ arbitrary variables are chosen within prescribed limits and by proper manipulation by QGEN subroutine of the search program, there results a positive definite matrix designated Q.

The Liapunov equation is then solved for the P-matrix, which is now assuredly positive definite. The search of the n-dimensional state space is then begun in a random fashion until a maximum V and a $\dot{V}$† close to zero is achieved, where

$$\dot{V} = - x^T Qx + 2x^T Pf(x) \quad . \tag{44}$$

A point at which $\dot{V}$ is approximately zero is achieved by a deterministic sectioning of a line that is struck from the origin to the point where $V > 0$ and $\dot{V} > 0$. The line is then searched by consecutively halving it, to most closely approximate the point where $\dot{V}$ changes sign, i.e., the $\dot{V} = 0$ constraint curve — thus the approximation of $\dot{V} = 0$.

The result of the search is a Liapunov function V from which the volume estimate of the domain of attraction can be computed directly as proportional to

$$\text{vol} \propto \left(\frac{V^n}{|P|}\right)^{\frac{1}{2}} \quad . \tag{45}$$

The general idea of the method is to maximize the volume estimate to obtain the largest estimate of the domain of attraction that can then be translated into physical constraints on each of the state variables.

---

†$(\dot{\cdot})$ indicates differentiation with respect to time.

Experimental Results - Nine Dimensional Problem

Prior to presenting the numerical experimental results, a presentation of the development of the algorithm from the theory is thought to be not only of casual interest, but of a great degree of relevence to those persons concerned with search techniques in general. Although all known search techniques in multidimensional spaces fall into the local minimum trap and no assurance can be had that you have a global minimum since you have not examined every point or every realizable trajectory in the space, it was originally felt that gradient methods or modified gradient methods could eventually (if pursued for many trials from many initial points), yield an answer that would be satisfactory. In the case presented here, any estimate of the angular variables larger than that obtained from the limits of the linear system's angular variables which in turn were finally set by thousands of computer trials of the equations of motion would be called satisfactory.

The problem of searching for a largest volume estimate of the domain of attraction in an n-space is two fold. First a technique must be developed to search the state space of order n, while the second aspect requires that a space of order $n(n+1)/2$ be searched optimally to acquire a best point from which the state space search can proceed.

The initial development began by arbitrarily choosing a point in the $n(n+1)/2$ space such that a positive definite Q-matrix was forthcoming, and the Liapunov equation could be solved for a positive definite P. Thus the Liapunov function $V = x^T P x$ could be evaluated and a search of the state space could begin. The solution of the Liapunov equation entailed an $n^2 \times n^2$ inversion, which for this case was 81×81. Double precision was deemed necessary after a check of the resulting single precision P-matrix showed it to be in error when substituted back into the Liapunov equation. The search of the nine space was initially begun by a straight gradient technique (MIN-ALL), which was touted to be an omnipotent tool in the analyses of a multidimensional (up to one-hundred independent variables) space for the minimum of a given function in that space. Much time was spent in arriving at an analytic gradient needed to facilitate operation of the program. Debugging of this analytic gradient was tedious and a numerical gradient was employed as a check.

In the numerical gradient, step size was critical in that a single fixed step size was not applicable in all coordinate directions; zero gradients in certain directions were obtained while very large gradients were obtained in other directions. An adaptive step size was tried, but the function was too varied in each direction, and we subsequently resorted to the analytic gradient after assurance of its accuracy by different persons checking the calculations.

The next step was minimization of V with $\dot{V} = 0$, along with the exclusion of the origin. Because the problem had a constrained minimum, the penalty function approach was chosen so that a constraint violation becomes penalized by virtue of the magnitude of the gain associated with the constraint; in this case, $\dot{V} = 0$ is the constraint. The function chosen to be minimized was

$$\ell = \min_{x \neq 0} \left[\left(V(x) \right)^2 + K_1^2 \left(\dot{V}(x) \right)^2 \right] , \tag{46}$$

where $x = 0$ must be avoided because it is a trivial solution to the problem. This was overcome by first minimizing an unconstrained problem

$$d = \min_{x} \left[\left(V(x) - r \right)^2 + K_1^2 \left(\dot{V}(x) \right)^2 \right] , \tag{47}$$

which drives $\dot{V}(x) \to 0$ and $V(x) \to r$. Adjustment of the gain K_1^2 during different phases of the search was of interest. Since $V(x) - r$ is the most important aspect in the d-minimization, $K_1^2 = 0.001$ was found to be satisfactory, whereas in the second phase (i.e., ℓ-minimization) $\dot{V}(x) \simeq 0$ was to be maintained, and hence $K_1^2 = 1000$ was found to be satisfactory to penalize the system from straying from this constraint. The choice of r was initially 0.1, but this was ultimately reduced to $r = 10^{-5}$ based on evidence produced by many runs.

The gradient method procedure was evaluated at this point. A great deal of time was required by the procedure for a search; this was due to the gradient computation at every point along the trajectory, and the local minimum problem which this method could not overcome. This, coupled with the fact that in the 45-space $\left(n(n+1)/2\right)$ search a gradient would be absurd, at least from an analytical point of view, and open to question concerning its validity in any numerical computation, dictated that another search method be found.

A random search was decided upon as a means of more effectively covering the nine-space with a better probability of avoiding the local minimum problem. How many points (nine-tuples) to evaluate for reasonable assurance of results was still a question, only to be answered by trial and error. Basically, a point (nine-tuple) was chosen arbitrarily (see Fig.15 for a two dimensional version of this technique) in the x-space along with an arbitrary Q and a large value of V, called V_o. ($V_o = 1$ is very large; V_o

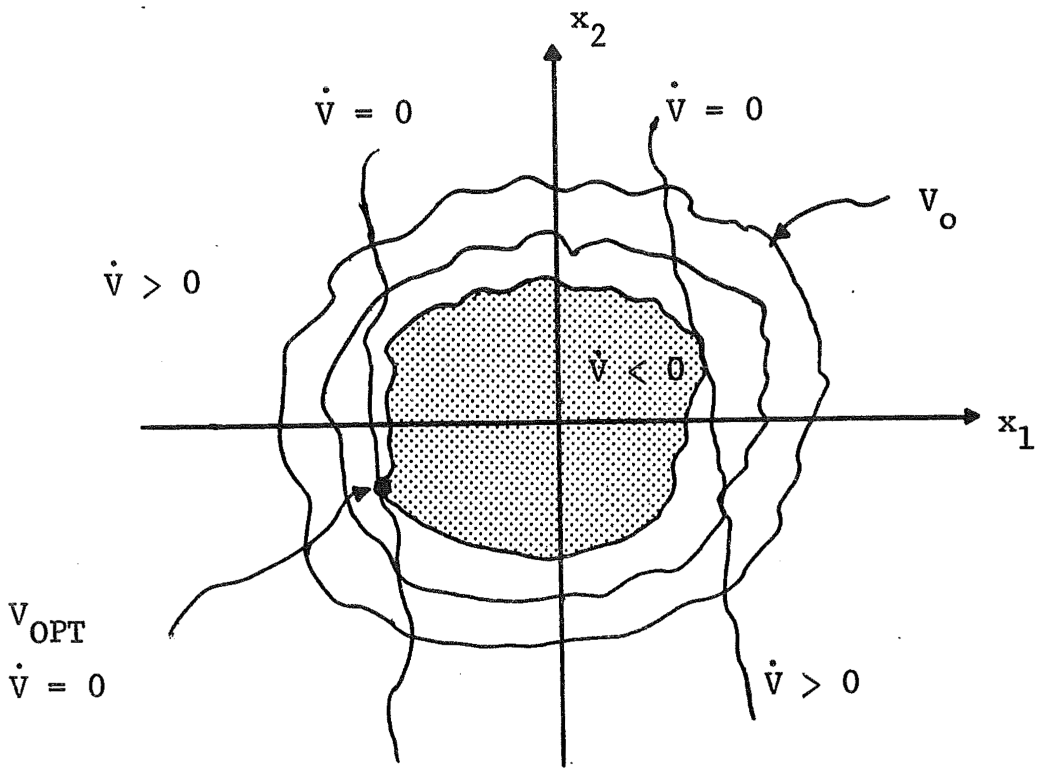


Fig. 15 Two Dimensional Representation of Search Region

corresponds to r in the gradient search discussed previously.) The Liapunov equation is still solved for P;

$$V = x^T P x \tag{48}$$

and

$$\dot{V} = - x^T Q x + 2x^T P f \quad . \tag{49}$$

It can be seen from the expression for V that the eigenvalues of P will determine the magnitude of the intercepts along the eigenvector directions by

$$y_{i_{INT}} = \sqrt{V/\lambda_i} \qquad i = 1, 2, \ldots, 9 \quad , \tag{50}$$

where λ_i are the eigenvalues of P and V is a value of the function V(x) at a particular point. A point in the x-space is related to a point in the y-space (eigenvector space) by a pure rotation given by

$$x = Cy \quad , \tag{51}$$

where C is the normalized matrix of eigenvectors of P (see Fig. 16). Thus the random points are determined by selecting a random number from a Gaussian distribution with zero mean and specified variance σ.

> [Experimental results indicated that when the random numbers were generated from a uniform distribution, a high percentage of the resulting V's were larger than the V calculated at the y-intercept point. In an attempt to compensate for this skewing effect, the eigenvector coordinates were generated such that the distribution near the boundaries would be attenuated. The Gaussian distribution model was tried and proved quite successful. Each scaled vector component is generated independently via the same Gaussian model (all zero mean, same variance). This gives a Rayleigh type distribution in the scaled radial envelope, i.e.,
>
> $$p(r) \approx \frac{r^{n-1} e^{-\frac{r^2}{2\sigma^2}}}{k(\sigma,n)} \quad ,$$
>
> where σ^2 is the variance for the Gaussian distribution and n is the dimension of the space. The tail of the radial distribution can be attenuated as desired by varying σ. A further modification was made by introducing a switching function which changes σ after a certain number of iterations: for less than 1000 iterations, $\sigma = 1/6$, while between 1000 and 5000 iterative points $\sigma = 1/3$.]

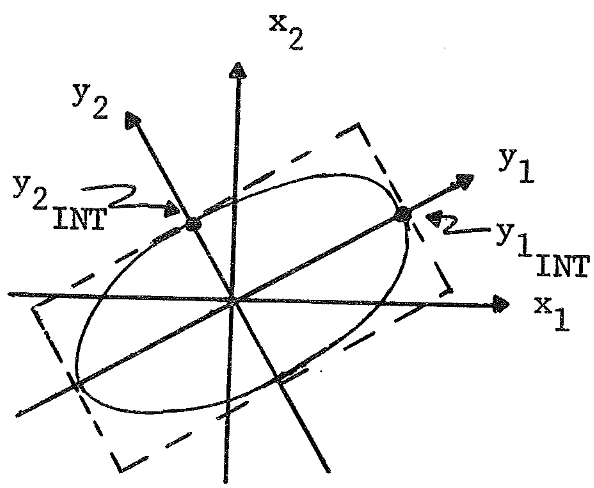


Fig. 16 Relationship of State Space (x) to its Associated Eigenvector Space (y) in Two Dimensions

Multiplying the random number by the intercept value $(y_{i_{INT}})$ yields a random point along y_i lying between $\pm y_{i_{INT}}$. Transformation to the x-space will allow a computation of V and $\dot{V}$ in the form expressed in Eqs. (48) and (49), respectively. If the Gaussian distribution produces a number outside the limits ± 1, this shows up in the calculation of V where the calculated V is greater than the last best V. If $V > V_{min}$, the point is discarded and a new random point is selected.

Since the problem has been formulated as a minimum problem by choice, a maximum V is being sought wherein all points interior to the contour $V = V_{max}$ have $\dot{V} < 0$ (stable trajectories). If $V_j < V_{j-1}$ $(j = 1, 2, \ldots, m,$ where m is the number of trials and j is the j^{th} trial) and $\dot{V} > 0$, then a line is struck from that point x to the origin, and this line of length ℓ is halved 15 (arbitrary number) times in order to best achieve the $\dot{V} = 0$ crossing. This portion of the search is termed the BISECTION PHASE and is deterministic (see Fig. 17). If $V_j > V_{j-1}$, a new random point is selected, since it has already been determined that, for that P, all trajectories have $\dot{V} > 0$ and are hence divergent. If $V_j < V_{j-1}$ and $\dot{V}_j < 0$ no new information is gained, since it only means that the point lies somewhere interior to the V_{j-1} contour, thus a new random point is selected. Figure 18 shows a flow diagram of the Random Search Technique.

The number of points in any search was still most uncertain. Experimentation showed that even if up to 300,000 nine-tuples were selected, the best estimate or max V, occurred usually before 5000 points were encountered. Thus 5000 points became the magical number as to how many trials were to be run during any one search.

The time to run was still excessive, so a reexamination of the steps was undertaken. The following steps reduced the running time measurably.

1. The generation of P from Q required an 81×81 inversion which was repeated each time a new Q was selected. This was wholly unnecessary since the inverted matrix was only a function of A which was constant. The revision was to perform the inversion once and store it.

2. Comparison of V_j to V_{j-1} was changed to compute the $\mathrm{vol}^{-1}(V_j)$ and compare it to the best vol^{-1*} gained since the beginning of

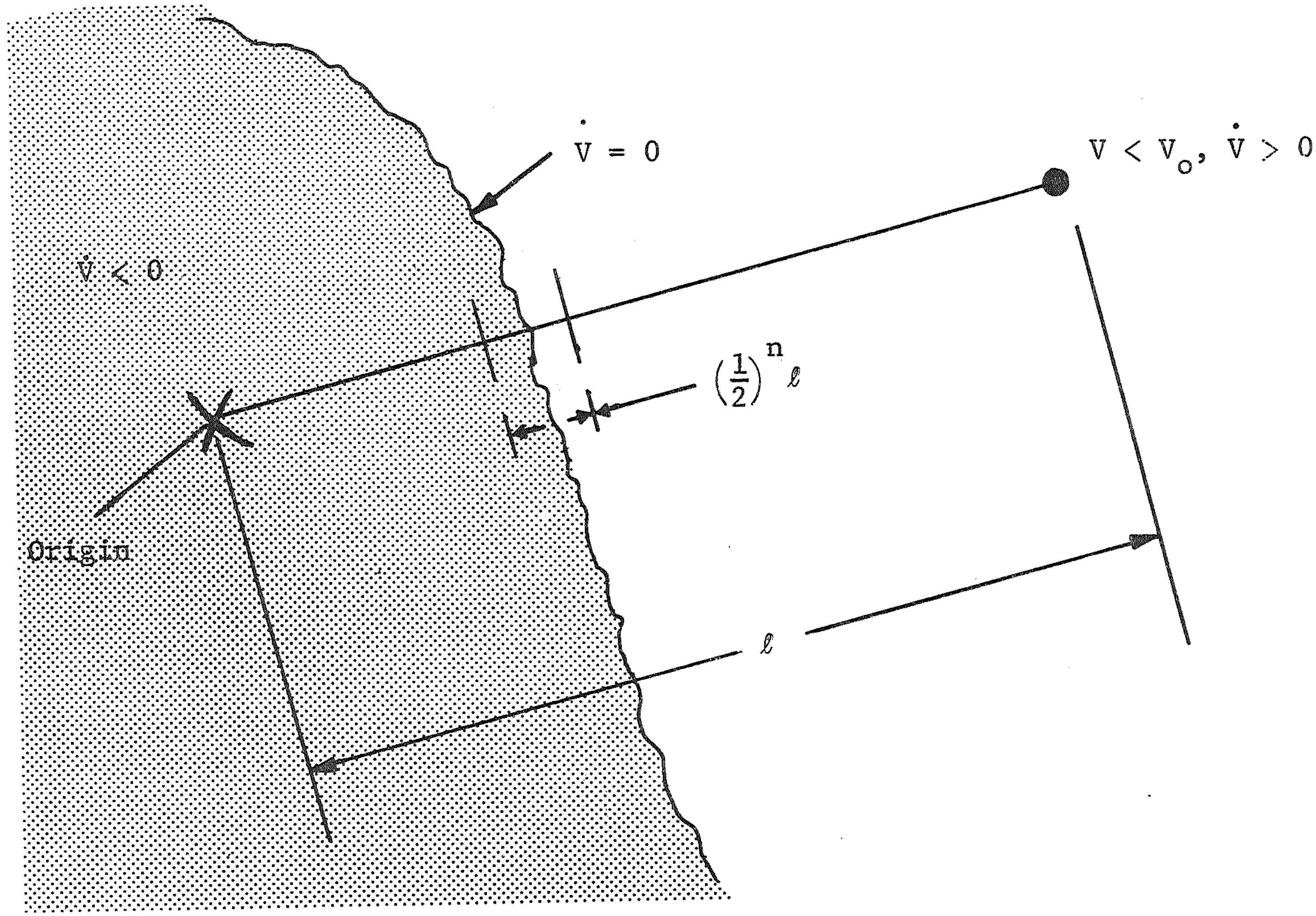


Fig. 17 Schematic Representation of BI-SECTION Deterministic Search

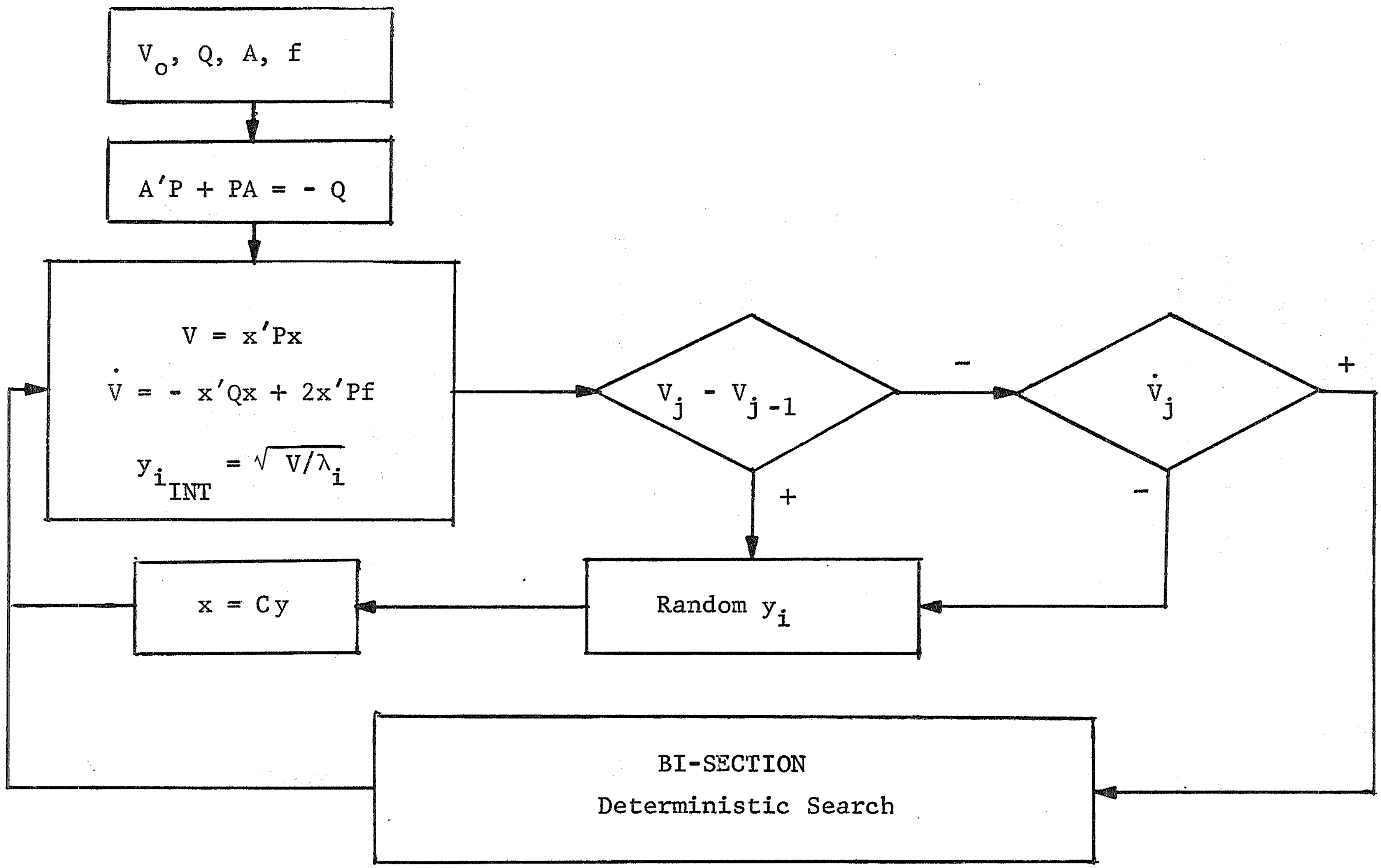


Fig. 18 Schematic Flow Chart of the Random Search of the State Space

the run and not just during that particular x-space search. If $vol^{-1}(V_j) > vol^{-1*}$ then the search was aborted immediately.

3. The search was also aborted for Liapunov functions $V < 10^{-12}$ since experience implied that V should be on the order of magnitude of 10^{-5} to 10^{-9}. There was an exponential overflow of the computer when $V \simeq 10^{-12}$ and $|P| \simeq 10^{30}$, both of which had occurred.

4. Another problem was alleviated by aborting the search if any of the eigenvalues of P were negative. This is possible due to numerical difficulties when the eigenvalues are small. The numerical technique yields small negative eigenvalues in enough cases to be annoying (because $y_i = \sqrt{V/\lambda_i}$, the computer yields incorrect results due to precision).

All of the details mentioned above were associated with the search of the state space (x-space). Those difficulties having been overcome, we next turned our attention to find a method of searching the associated parameter space (45 space) to yield a best Q, which in turn will yield a minimum inverse volume estimate. It could not be done in the same way as the random search of the state space, since the geometry of parameter space was not known as was the geometry of the state space. Gradient techniques were "OUT" based on our unrewarding experience with MIN-ALL.

Random search was deemed the way to proceed. Note here that the parameter space in this problem is of order $n(n+1)/2$, which is 45. The 45 variables are λ_i, $i = 1, 2, \ldots, 9$; θ_j, $j = 1, 2, \ldots, 28$, and ϕ_k, $k = 1, 2, \ldots, 8$.

The first search method was just on the lambdas (λ_i, $i = 1, 2, \ldots, 9$), which are the diagonal terms of the Q-matrix when the rotation components (θ_j, $j = 1, 2, \ldots, 28$ and ϕ_k, $k = 1, 2, \ldots, 8$) are zero. This method consisted of setting all θ's and ϕ's equal to zero and all λ's initially equal to unity. Since the Q-matrix can be scaled by any one of the λ's, λ_1 was chosen arbitrarily to remain at unity while the other λ's were varied by random choice in the following sequential manner:

1. Select, sequentially, ten random choices of λ_2 distributed uniformly from zero to one hundred with λ_3 through λ_9 equal to unity (the choice of the upper bound of one hundred is arbitrary; all that is necessary is that $\lambda_i > 0$).

2. Generate an inverse volume estimate for each λ and retain the minimum inverse volume estimate and its associated λ_2; call it $\lambda_{2_{min}}$.

3. Starting with $\lambda_1 = 1$ and $\lambda_2 = \lambda_{2_{min}}$, search ten random λ_3's, holding all λ_i's $= 1$, $i = 4, 5, \ldots, 9$.

4. Search the λ_4 through λ_9 variables in the same manner as above, retaining the minimum inverse volume with its associated λ.

The best inverse volume estimate using the above method was $vol^{-1} = .148 \times 10^{50}$ for $\lambda_1 = 1$, $\lambda_2 = 27.70$, and λ_3 through λ_9 equal to 1. This method was abandoned because other work indicated that there might be a greater sensitivity of the volume to changes in the rotation variables than to the hyperellipse axis scaling variables (λ_i, i = 1, ..., 9). Furthermore, the search technique employed above tends to restrict the search volume somewhat; a minimum along one axis will not necessarily lead to the minimum over the space in question.

The second technique was also random, and was based on a set of λ's, θ's and ϕ's that were thought to be optimum in the final report of last year's work (Ref. 3). The optimal set* was searched with $\gamma_{1c} = -\gamma_{2c} = 2.875°$, $\beta_{1c} = 0$, and $\beta_{2c} = -30°$. This resulted in our best inverse volume estimate at that time of $.111 \times 10^{41}$, which was nine orders of magnitude better than the best result of the λ search mentioned above. Holding the λ's at the values of the optimal set, the following θ's and ϕ's were chosen randomly (uniform distribution over $[-\pi/2, \pi/2]$): θ_j, j = 6, 7, 13, 21, 22, 25 with $\theta_{22} = \theta_{28}$, $\phi_7 = \theta_{21}$, and $\phi_8 = \theta_{25}$. All other angular parameters were set equal to zero. This choice of rotations restricts Q and P to consist of three nonzero 3 × 3 blocks on the diagonal. The best inverse volume estimate in this case was $vol^{-1} = .841 \times 10^{48}$, which is approximately eight orders of magnitude worse than that obtained with the optimal set.

---

*$\lambda_1 = 1.99$, $\lambda_2 = 50.0$, $\lambda_3 = 0.01$, $\lambda_4 = \lambda_7 = 5.1962$, $\lambda_5 = \lambda_8 = 10.0$, $\lambda_6 = \lambda_9 = 0.0038$, $\theta_6 = -\pi/4$, $\theta_{21} = \phi_7 = -1.373$, and $\theta_7 = \theta_{13} = \theta_{22} = \theta_{25} = \theta_{28} = \phi_8 = 0$.

Another series of runs were made with the optimal set and random perturbations, as before, on the θ and Φ variables, but with $\gamma_{1c} = \beta_{1c} = 30°$ and $\gamma_{2c} = \beta_{2c} = -30°$. The best value of the inverse volume estimate obtained was $vol^{-1} = .913 \times 10^{48}$, which again is not as good as the previous set of runs. Note that since $\beta_{1c} \neq 0$, P does not have the same form as Q because A is not of the same form.

Some interesting observations of the minimum inverse volume estimate $(.111 \times 10^{41})$ are that its calculation contained the smallest determinant of P $(|P| = 49.76)$ that has been observed to date. The determinant of P is usually on the order of from 10^9 to 10^{20}. Besides this, the angular state variables at the end of the search are approximately $\theta = 20$ sec, $\Phi = -12$ sec, and $\psi = 20$ sec of arc, which are also larger than usual. These values are at the point where the minimum V occurs on $\dot{V} = 0$. The maximum values obtained on the axes of the ellipsoidal estimate are $\Phi = 26.4$ sec, $\theta = 56.6$ sec, and $\psi = 56.5$ sec. The conjectured sensitivity to the rotation variables is borne out by these data, that is: for the case $\gamma_{1c} = -\gamma_{2c} = 2.875$, $\beta_{1c} = 0$ and $\beta_{2c} = -30°$, the least and largest inverse volume estimates differ by 14 orders of magnitude; for the case $\gamma_{1c} = \beta_{1c} = 30°$, and $\gamma_{2c} = \beta_{2c} = -30°$, the range is 5 orders of magnitude. The range of variation observed for the random search over the eigenvalues is only three orders of magnitude.

While the random searches were continuing, a paper given by Barron (Ref. 12) gave some insight into what was termed an accelerated random search, a search with an orderly way of selecting step size. The technique consists of picking a starting point, λ^o, θ^o, and Φ^o, for which a volume estimate is obtained. A performance measure is defined as:

$$\underline{P} = \log \left[\frac{(\mathrm{vol}^{-1})}{10^{30}} \right] ,$$

where 10^{30} is arbitrary and chosen to keep $\underline{P}$ positive and in the vicinity of unity. Since this first estimate is the best to date, set $\underline{P}^*$, the best estimate, equal to $\underline{P}$. A random step, consistent with the constraints on λ, θ, and ϕ (i.e., $\lambda > 0$, $|\theta| \leq \pi/2$, $|\phi| < \pi$), is chosen by first defining the variance as

$$\sigma = \log \underline{P}^* ,$$

picking a random number x, $-1 \leq x \leq +1$, and finally defining the step size as

$$\Delta u = (\mathrm{sgn}\ x) \frac{\ell}{d} e^{-x^2/\sigma^2} ,$$

where

$$\ell = \begin{cases} 1000 & \text{for } \lambda_i \quad i = 1,9 \\ \pi/2 & \text{for } \theta_i \quad i = 1,28 \\ \pi & \text{for } \phi_i \quad i = 1,8 \end{cases}$$

and

d = arbitrary divisor (taken as 4 initially).

Addition of this random step to the previous values of λ, θ, and ϕ results in another value of $\underline{P}$. If $\underline{P} > \underline{P}^*$, a step in the opposite direction is taken by setting $\Delta u = -\Delta u$, the consistency with the constraints is checked again, and $\underline{P}$ is recalculated. If $\underline{P}$ is still greater than $\underline{P}^*$, a new random step is chosen and added to the point associated with $\underline{P}^*$. As long as $\underline{P} < \underline{P}^*$, we set $\underline{P}^* = \underline{P}$ and the step size is doubled until $\underline{P} > \underline{P}^*$, then a new random step is instituted from the point (λ, θ, ϕ) associated with $\underline{P}^*$.

The accelerated random search is based upon the concept of randomly choosing a search direction and a step size, and searching in that direction until a minimum is found. At the minimum, a new random direction and step size are chosen and the process is repeated. As the search gets closer to the minimum, the variance of the random step is decreased to facilitate accurate determination of the minimum.

The best minimum inverse volume estimate gained during the initial phases with the above technique was $.585 \times 10^{38}$ whereas the best previous one was $.111 \times 10^{41}$. Therefore, the other two techniques were abandoned. The last change that was incorporated into the algorithm was a "creeping aspect" of the random search by which the random Δu's become either larger or smaller as required. In particular, if $d = 4$ in the expression for Δu, as prescribed above, and say 100 points are looked at with no improvement, then d is halved, thereby doubling Δu. One hundred trials with no improvement causes another halving of d, etc. If an improvement is obtained, d is set back to 4 and the expansion begins anew. If after say 500 trials where Δu is 16 times its original value and no improvement has been found, then d is set back to 4 and is consecutively doubled to decrease the step size in the same manner as the step size was increased above. Thus the creeping random search has an expanding and contracting facility, which has proven useful in determining the best vol^{-1} estimate to date.

The final results and conclusions of the nine dimensional search based on experimentation with the program utilizing an IBM 360/95 at the Institute for Space Studies-NASA (ISS) and the IBM 360/75 at Grumman are summarized in Table 1. Prior to running at the ISS, a 5000 point random search was being used in the state

Table 1

TABULATED RESULTS OF Q-MATRIX SEARCH PROGRAM AT THE INSTITUTE FOR SPACE STUDIES

Run #	Trials	Time (Min)	vol^{-1} *	$\underline{P}$ *	Comments
1*	~ 150	3	$.606 \times 10^{33}$	1.1217	5000-pt. search, quasi-diagonal Q (q-d-Q); job aborted - excessive output
2*	~ 2500	62	(not printed out)	1.1480	5000-pt. search, q-d-Q, reduced output run
3*	1019	105	$.599 \times 10^{35}$	1.1592	300,000-pt. search from the best point of run 2, q-d-Q (all runs from here on are 300,000 pts)
4*	1000	14	$.497 \times 10^{35}$	1.1566	Started from the best point of run 3, full-Q
5*	2000	16	$.892 \times 10^{35}$	1.1650	Continuation of run 4 in random # gen., full-Q
6*	2473	72	$.488 \times 10^{35}$	1.1563	q-d-Q from best point in run 3 (really an extension of 3)
7**	1022	118	$.187 \times 10^{42}$	1.3757	q-d-Q, start from $\lambda_i = 1$, $\theta_j = \phi_k = 0$
8**	1872	136	$.202 \times 10^{41}$	1.3435	Start from the best point in run 3, q-d-Q
9***	29	38	$.134 \times 10^{49}$	1.6042	Start from the best point in run 3, q-d-Q
10***	453	61	$.105 \times 10^{50}$	1.6340	Same as run 9 except full-Q
11****	1000	2	abort condition† 10^{51}	1.6666	q-d-Q, started from the best point in run 3
12*****	1000	3	abort condition† 10^{51}	1.6666	q-d-Q, started from the best point in run 3
Total Trials	14518				

* $h_i = 0$, $\gamma_{1c} = -\gamma_{2c} = .05017822$ rad, $\beta_{1c} = 0$, $\beta_{2c} = -\pi/6$

** $h_i = 0$, $\gamma_{1c} = -\gamma_{2c} = \pi/4$, $\beta_{1c} = 0$, $\beta_{2c} = -\pi/6$

*** $h_i = 0$, $\gamma_{1c} = -\gamma_{2c} = \pi/4$, $\beta_{1c} = -\beta_{2c} = \pi/6$

**** $h_i = 1/1500$ (half wheel speed)+,-,+, $\gamma_{1c} = -\gamma_{2c} = .05017822$, $\beta_{1c} = 0$, $\beta_{2c} = -\pi/6$

***** $h_i = 0.2/1500$ (1/10 wheel speed)+,+,+, $\gamma_{1c} = -\gamma_{2c} = .050178$, $\beta_{1c} = 0$, $\beta_{2c} = -\pi/6$

† if the best Liapunov function is $< 10^{-12}$, the vol^{-1} is set $= 10^{51}$ and P^* thus becomes $\sim 5/3$

space, and there were vol^{-1} estimates as small as $.224 \times 10^{34}$ at Grumman. Because the IBM 360/95 at the ISS has a core 10 times larger than the IBM 360/75, and is 5 to 8 times faster, a 300,000-point random state space search was instituted in lieu of the 5000-point search in order to gain greater assurance that a valid answer had been reached. The best vol^{-1} estimate obtained using the 300,000-random point search was $.488 \times 10^{35}$, which corresponds to physical variable limits of* $|\phi| = 2.48$ min, $|v_\phi| = 0.181$ ft-lb_f-sec, $|\omega_\phi| = 1980$ volts, $|\theta| = 5.91$ min, $|v_\theta| = 0.342$ ft-lb_f-sec, $|\omega_\theta| = 941$ volts, $|\psi| = 8.98$ min, $|v_\psi| = 0.452$ ft-lb_f-sec, $|\omega_\psi| = 1410$ volts. This result was obtained with a quasi-diagonal Q-matrix, zero initial momenta ($h_\phi^o = h_\theta^o = h_\psi^o = 0$), $\sin(\gamma_{1c} - \gamma_{2c}) = 0.1$, $\beta_{1c} = 0$, and $\beta_{2c} = -\pi/6$ radians (Run #6, Table 1). The inclusion of a full Q-matrix, addition of initial momenta, the inclusion of $\beta_{1c} \neq 0$, and increasing the $\sin(\gamma_{1c} - \gamma_{2c})$ to values > 0.1 causes degradation of the vol^{-1} as illustrated in Table 1.

The results indicate that the problem is quite sensitive to system parameter variation such as γ_{1c}, γ_{2c}, β_{1c}, and initial momenta. Further, the best Q-matrix for one set of parameters is certainly not the best for all sets of parameters. The best Q-matrix found was for $\beta_{1c} = 0$ and Q chosen as quasi-diagonal. When $\beta_{1c} = 0$, the A-matrix becomes quasi-diagonal thereby decoupling the system, at least in the linear part. The nonlinear part is still coupled through the roll, pitch, and yaw channels. The most severe degradation of the system came when initial momenta were introduced to even one tenth of wheel capacity.

The computer programs which perform the technique illustrated herein are found in Appendix I.

---

*It should be pointed out that the angle intercepts $|\phi|$, $|\theta|$, and $|\psi|$ were diminished by factors of approximately three from those obtained with the 5000-point random search.

Experimental Results - Six Dimensional Problem

The six dimensional approximation to the nine dimensional Ames OAO system has been programmed for domain of attraction investigation by using the stability analysis algorithm (Pick-a-Q). The Q-matrix parameter space is of dimension 21. After restricting the Q search to quasi-diagonal matrices, with a limit of 100,000 random points per inner loop search, the best results obtained for the case $\beta_{1c} = 0$, $\beta_{2c} = -30°$, $\gamma_{1c} = 2.875° = -\gamma_{2c}$, $s(\gamma_{1c} - \gamma_{2c}) = 0.1$, $h^o_\phi, h^o_\theta, h^o_\psi = 0$ are:

$$Q = \begin{bmatrix} 336.3 & -.2124 & 0 & 0 & 0 & 0 \\ -.2124 & .235 \times 10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 0 & .0131 & -1.387 & 0 & 0 \\ 0 & 0 & -1.387 & 147.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & .0203 & -2.036 \\ 0 & 0 & 0 & 0 & -2.036 & 217.1 \end{bmatrix}$$

$$P = \begin{bmatrix} 974.0 & -1.291 \times 10^4 & 0 & 0 & 0 & 0 \\ -1.291 \times 10^4 & 2.169 \times 10^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & .0488 & -2.512 & 0 & 0 \\ 0 & 0 & -2.512 & 621.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & .0774 & -3.895 \\ 0 & 0 & 0 & 0 & -3.895 & 928.6 \end{bmatrix}$$

$$\text{Inverse volume} = .562 \times 10^{25}$$

$$V_{final} = .353 \times 10^{-6}$$

$$\dot{V}_{final} = .117 \times 10^{-10}$$

$$X_{final} = \begin{bmatrix} .170 \times 10^{-4} \\ .210 \times 10^{-5} \\ -.735 \times 10^{-3} \\ -.654 \times 10^{-5} \\ -.432 \times 10^{-4} \\ .272 \times 10^{-5} \end{bmatrix}$$

The semiaxes of the ellipsoid estimating the domain of attraction in nondimensional variables are:

$$\begin{bmatrix} .75 \times 10^{-7} \\ -1.27 \times 10^{-6} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -.816 \times 10^{-7} \\ 1.95 \times 10^{-5} \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ .963 \times 10^{-7} \\ -2.38 \times 10^{-5} \\ 0 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 4.15 \times 10^{-5} \\ 2.48 \times 10^{-6} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2.4 \times 10^{-3} \\ -1.01 \times 10^{-5} \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -3.02 \times 10^{-3} \\ 1.22 \times 10^{-5} \\ 0 \\ 0 \end{bmatrix}$$

Note that the wide dispersion of eigenvalues in the P matrix gives very thin ellipsoidal projections. The maximum allowable values (not occurring simultaneously) of the physical variables are:* ϕ = .14 min, θ = 10.4 min, ψ = 8.30 min, v_ϕ = .050 lb-ft-sec, v_θ = 0.48 lb-ft-sec, v_ψ = 0.39 lb-ft-sec. The present computation rate is about 4000 Q matrices per hour.

Comparison of 6 and 9 Dimensional Results

Both programs exhibited best results for the case of the quasi-diagonal Q-matrix, zero initial momenta ($h_\phi^o = h_\theta^o = h_\psi^o$), $\sin(\gamma_{1c} - \gamma_{2c}) = .1$, $\beta_{1c} = 0$, and $\beta_{2c} = -\pi/6$ radians. The 6 dimensional results provided a better $|\theta|$ intercept estimate of 10.4 min than the 9 dimensional estimate of 5.91 min; however, the 9 dimensional program provided surprisingly better $|\phi|$ and $|\psi|$ estimates (2.48 min compared to .14 min for $|\phi|$, and 8.98 min compared to 8.30 min for $|\psi|$). Perhaps the failure of the 6 dimensional program to provide clearly

*It should be noted that a 100,000-point search in 6 dimensions is equivalent to a 32 million point search in 9 dimensions.

superior estimates in spite of the smaller dimension of its Q parameter search and greater number of trials is due to the fact that the effect of a 100,000-point search per trial in 6 dimensions is approximately equivalent to a 32 million-point search in 9 dimensions; the 32 million is very conservative compared with the 300,000-point search actually used in the 9 dimensional case. It is interesting to compare the P-matrix eigenvector projections corresponding to the above cases (see Fig. 19.

As can be seen by Tables 1 and 2 the volume estimates seem to be affected proportionately for both the 6 and 9 dimensional cases, as the commanded gimbal angles or initial total momentum values are changed.

Table 2

SUMMARY OF RUNS AT ISS FOR 6 DIMENSIONAL MODEL

Run #	Trials	Time (Min)	vol^{-1}	P^*	Comments
1^*	~ 8000	120	$.562 \times 10^{25}$	1.237	100,000 point search, $P^* = \log(vol)^{-1}/20$
2^{***}	~ 3000	60	$.261 \times 10^{30}$	1.471	" "
3^{**}	4200	60	$.225 \times 10^{27}$	1.318	" "
4^{****}	4000	60	----	---	vol^{-1} too small to compute without rescaling problem

* $h^o = 0$, $\gamma_{1c} = -\gamma_{2c} = .05017822$ rad, $B_{1c} = 0$, $B_{2c} = -\pi/6$ rad

*** $h^o = 0$, $\gamma_{1c} = -\gamma_{2c} = .05017822$ rad, $B_{1c} = \pi/6$, $B_{2c} = -\pi/6$ rad

** $h^o = 0$, $\gamma_{1c} = -\gamma_{2c} = \pi/4$, $B_{1c} = 0$, $B_{2c} = -0/6$ rad

**** $h^o = 1/1500\ (+,-,+)$, $\gamma_{1c} = -\gamma_{2c} = .05017822$ rad, $B_{1c} = 0$, $B_{2c} = -\pi/6$ rad

A. Six Dimensional Semiaxes $(\phi', v''_\phi, \theta', v''_\theta, \psi', v''_\psi)$

$$\begin{bmatrix} .75 \times 10^{-7} \\ -1.27 \times 10^{-6} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -.82 \times 10^{-7} \\ 1.95 \times 10^{-5} \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ .96 \times 10^{-7} \\ -2.4 \times 10^{-5} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4.2 \times 10^{-5} \\ 2.5 \times 10^{-6} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2.4 \times 10^{-3} \\ -1.0 \times 10^{-5} \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -3.0 \times 10^{-3} \\ 1.2 \times 10^{-5} \\ 0 \\ 0 \end{bmatrix}$$

B. Nine Dimensional Semiaxes $(\phi', v'_\phi, \omega''_\phi, \theta', v'_\theta, \omega''_\theta, \psi', v'_\psi, \omega'_\psi)$

$$\begin{bmatrix} .39 \times 10^{-8} \\ .68 \times 10^{-5} \\ .82 \times 10^{-7} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ .25 \times 10^{-7} \\ 1.7 \times 10^{-5} \\ .54 \times 10^{-6} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -.32 \times 10^{-7} \\ -2.3 \times 10^{-5} \\ .81 \times 10^{-6} \end{bmatrix}, \begin{bmatrix} .51 \times 10^{-4} \\ -.54 \times 10^{-6} \\ .45 \times 10^{-4} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -.33 \times 10^{-4} \\ .51 \times 10^{-5} \\ -1.5 \times 10^{-4} \end{bmatrix},$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ -.33 \times 10^{-4} \\ .47 \times 10^{-5} \\ -1.5 \times 10^{-4} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} .73 \times 10^{-3} \\ .91 \times 10^{-5} \\ -.83 \times 10^{-3} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.7 \times 10^{-3} \\ -1.0 \times 10^{-5} \\ 0.4 \times 10^{-3} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2.5 \times 10^{-3} \\ -1.7 \times 10^{-5} \\ .57 \times 10^{-3} \end{bmatrix}$$

Note: Both cases correspond to $B_{1c} = 0$, $B_{2c} = -\pi/6$ rad, $\gamma_{1c} = -\gamma_{2c} = 2.8750$, $s(\gamma_{1c} - \gamma_{2c}) = 0.1$, $h^o_\phi, h^o_\theta, h^o_\psi = 0$.

Fig. 19 Comparison of Semiaxes Projections Corresponding to Optimal Ellipsoids for 6 and 9 Dimension

Experimental Results for a Simple System

In the hope of determining whether there is a fundamental limitation inherent in estimating the domain of attraction of a system containing saturation and one or more zero eigenvalues, a two dimensional system with a single saturation and a zero eigenvalue has been formulated as a prototype of the OAO system. The random search algorithm has been programmed for use on the GE-235. A two dimensional stability problem was attempted, with the following system equations:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} \xi .$$

$$\xi = - \operatorname{sat}(\sigma) = \begin{cases} -.001 & \sigma > .001 \\ -\sigma & |\sigma| \leq .001 \\ .001 & \sigma < -.001 \end{cases} .$$

This describes a critical plant with high gain saturating characteristics. Note that our definition of ξ here is equivalent to defining a constant $K_c = 10^3$, and using a ξ defined as

$$\xi = - \frac{1}{K_c} \left(\operatorname{sat}\left(f(g) \right) \right) ,$$

where

$$g = K_c \sigma$$

$$- \operatorname{sat}(f) = \begin{cases} -1 & \sigma > 1 \\ -\sigma & \sigma \leq 1 \\ 1 & \sigma < -1 \end{cases}$$

$$\sigma = x_1 .$$

A change of coordinates produces:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} -\alpha & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} -\frac{K_2}{\alpha} \\ K_1 + \frac{K_2}{\alpha} \end{pmatrix} \xi$$

$$\xi = -\text{sat}(\sigma)$$

$$\sigma = y_1 + y_2 \quad ,$$

where

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 & -\alpha^{-1} \\ 1 & \alpha^{-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad .$$

In the format required for our analysis, the equations are:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} -\left(\alpha - \frac{K_2}{\alpha}\right) & \frac{K_2}{\alpha} \\ -\left(K_1 + \frac{K_2}{\alpha}\right) & -\left(K_1 + \frac{K_2}{\alpha}\right) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} \frac{K_2}{\alpha} \\ -\left(K_1 + \frac{K_2}{\alpha}\right) \end{pmatrix} g(\sigma)$$

$$\sigma = y_1 + y_2$$

$$g(\sigma) = \text{sat}(\sigma) - \sigma \quad .$$

The system characteristics exhibited here were chosen for their similarity to the system equations of the OAO stability problem.

Given a $[\lambda, \theta]$ pair, which parameterize the Q-matrix, the objective of the numerical experiments was to find answers to the following.

1. Given a particular $[\lambda,\theta]$ pair, how well does the estimate of $\min V$ on $\dot{V} = 0$ compare with the true value? The solution to this question can be found by amplifying a specific set of results. For the case where $\alpha, K_1, K_2 = 1$, $\lambda = .001$, and $\theta = -.40$, the algorithm provided $V = .097$ as a final answer, with coordinates $y_1 = .64$, $y_2 = -1.69$ corresponding to the intersection of the constraint $\dot{V} = 0$ with the ellipse $V = .097$. Neighboring V and $\dot{V}$ loci were plotted with the result that the $\dot{V} = 0$ constraint did not intersect any V curve below approximately $V = .090$. Thus the estimate is in error by less than 8 percent.

2. How large a region of stability can be predicted via the algorithm by searching over the $[\lambda,\theta]$ space? For this case we know that the system is absolutely stable in the $[\epsilon,K]$ sector. Therefore in the case of a saturation nonlinearity, the domain of attraction is nearly the whole space. In our numerical experimentation, most of the points searched in the $[\lambda,\theta]$ space resulted in stability estimates within the region of essential linearity $(|y_1 + y_2| < .001)$. There was, however, a small region in the $[\lambda,\theta]$ space that gave stability estimates well beyond that of essential linearity. Best results for about 100 points searched in the $[\lambda,\theta]$ space provide us with $[\lambda,\theta] = [.001, -.55]$, volume = 2.52, $V_{final} = .1739$, intersection with $\dot{V} = 0$ at $(y_1,y_2) = [-1.42, 2.46]$.

3. How sensitive is the "volume" of the estimate of the region of stability to $[\lambda,\theta]$ parameter changes?

The answer is dramatically evident in reviewing a small sample of results:

$$\left\{\begin{array}{l}\lambda = .001 \\ \theta = -.35\end{array}\right\} \longrightarrow \left\{\begin{array}{l}V_{final} = 3 \times 10^{-6} \\ \text{volume} = 3.5 \times 10^{-5} \\ y_1 = .0046 \\ y_2 = -.0036\end{array}\right.$$

$$\left\{\begin{array}{l}\lambda = .001 \\ \theta = -.40\end{array}\right\} \longrightarrow \left\{\begin{array}{l}V_{final} = .097 \\ \text{volume} = 1.24 \\ y_1 = .64 \\ y_2 = -1.69\end{array}\right.$$

$$\left\{\begin{array}{l}\lambda = .001 \\ \theta = -.55\end{array}\right\} \longrightarrow \left\{\begin{array}{l}V_{final} = .174 \\ \text{volume} = 2.52 \\ y_1 = -1.42 \\ y_2 = 2.46\end{array}\right.$$

$$\left\{\begin{array}{l}\lambda = .001 \\ \theta = -.60\end{array}\right\} \longrightarrow \left\{\begin{array}{l}V_{final} = 5.5 \times 10^{-6} \\ \text{volume} = 7.9 \times 10^{-5} \\ y_1 = -.0133 \\ y_2 = .0123\end{array}\right. .$$

The sensitivity to the rotation parameter θ is very apparent.

These results seem to indicate that the optimal quadratic estimation of the domain of attraction is a viable approach for the case of a critical plant with high gain saturating characteristics.

Picking a P-Matrix

Computer runs were made that used the technique of generating a positive definite P-matrix directly, without generating Q and then solving the Liapunov equation

$$PA + A^T P = - Q \tag{52}$$

for P. This work originated with the motive of speeding the routine so that a great many matrices could be examined with a minimum of machine time. The abort feature described previously was essential to this plan, because the fact that P is positive definite does not ensure the definiteness of Q. Use of the abort procedure without direct generation of P had reduced the time required to investigate a matrix from 50 seconds to an average of about 10 seconds (2 seconds to search nine dimensional space and 8 seconds to generate Q and solve the Liapunov equation for P). Inversion of the 81×81 matrix A_{mod} at each iteration was corrected as described previously. This reduced the time required to generate both Q and P, but the time saved was not the main advantage of direct P-matrix generation.

It soon became clear that direct generation of the P-matrix has other advantages. The eigenvalues and eigenvectors of P have direct physical interpretation in the nine dimensional state space, so that picking P directly follows Richard Hammings commandment, "the purpose of computing is insight, not numbers" (Ref. 13).

In addition, when the P-matrix is generated directly, the A-matrix is unnecessary, and the time derivative $\dot{x}$ can be efficiently computed directly from the nonlinear function. By setting $x_1 = \phi'$, $x_2 = v''_\phi$, $x_3 = \omega''_\phi$, $x_4 = \theta'$, $x_5 = v''_\theta$, $x_6 = \omega''_\theta$, $x_7 = \psi'$, $x_8 = v''_\psi$, $x_9 = \omega''_\psi$, the equations of Fig. 4 may be written in the form (for all initial momenta zero):

$$
\begin{aligned}
\dot{x}_1 &= -ax_2 - a(x_5 \sin x_1 + x_8 \cos x_1) \tan x_4 \\
\dot{x}_2 &= -bx_2 + \frac{b}{k} \text{sat}(10k\Delta\gamma_1 + 9kx_3) \\
\dot{x}_3 &= -2(x_3 + \Delta\gamma_1) \\
\dot{x}_4 &= -a(x_5 \cos x_1 - x_8 \sin x_1) \\
\dot{x}_5 &= -bx_5 + \frac{b}{k} \text{sat}\left(d_{12}k(\Delta\beta_1 \cos \Gamma_2 + \Delta\beta_2 \cos \Gamma_1 + \frac{9}{2} x_6)\right) \\
\dot{x}_6 &= -2x_6 - .2d_{12}(\cos \Gamma_2\Delta\beta_1 + \cos \Gamma_1\Delta\beta_2) \\
\dot{x}_7 &= -a(x_5 \sin x_1 + x_8 \cos x_1)/\cos x_4 \\
\dot{x}_8 &= -b\left(x_8 - \text{sat}\left(d_{12}k(-\Delta\beta_1 \sin \Gamma_2 - \Delta\beta_2 \sin \Gamma_1 + \frac{9}{2} x_9)\right)\right) \\
\dot{x}_9 &= -2x_9 + .2d_{12}(\sin \Gamma_2\Delta\beta_1 + \sin \Gamma_1\Delta\beta_2) \quad ,
\end{aligned}
\tag{53}
$$

where

$$
\begin{aligned}
a &= K_mK_c/I, \; b = \frac{1}{\tau_m}, \; k = K_c, \; d_{12} = 20 \,\text{sgn}(\gamma_{1c} - \gamma_{2c}), \\
h^o_\theta &= h^o_\phi = h^o_\psi = 0, \; \Gamma_1 = \Delta\gamma_1 + \gamma_{1c}, \; \Gamma_2 = \Delta\gamma_2 + \gamma_{2c}
\end{aligned}
\tag{54}
$$

relates the present notation to that of Fig. 4. (Note that d_{12} here is ten times d_{12} of Ref. 3.) See Appendix III for subroutine DER, which performs this calculation. The version in the search routine is more sophisticated than the earlier version in the simulation routine. The main improvement is in the calculation of $\Delta\gamma_i$ and $\Delta\beta_i$. Specifically, by using the upper sign for $i = 1$ and the lower for $i = 2$, in Eq. (55)

$$\Delta\gamma_i = \tan^{-1}\left(\frac{\epsilon_\alpha - \epsilon_\beta \tan \gamma_{ic}}{1 + \epsilon_\beta(\tan \gamma_{ic} + \epsilon_\alpha) \tan \gamma_{ic}}\right) , \tag{54}$$

where

$$\epsilon_\alpha = (\cos x_4 - 1) \tan \gamma_{ic} \mp \sin x_7 \tan x_1 \frac{\tan \beta_{ic}}{\cos \gamma_{ic}}$$
$$\mp \cos x_7 \sin x_4 \frac{\tan \beta_{ic}}{\cos \gamma_{ic}} + \cos x_7 \tan x_1 - \sin x_7 \sin x_4 ,$$

(55)

$$\epsilon_\beta = (\cos x_7 - 1) + \sin x_7 \sin x_4 \tan x_1 \mp \sin x_7 \frac{\tan \beta_{ic}}{\cos \gamma_{ic}}$$
$$\pm \tan x_1 \sin x_4 \cos x_7 \frac{\tan \beta_{ic}}{\cos \gamma_{ic}} - \tan x_1 \cos x_4 \tan \gamma_{ic} .$$

The formula in Fig. 4 is used to compute $\Delta\beta_i$ unless round-off error would be significant because $\Delta\beta_i < 0.1$, in which case $\Delta\beta = \sin^{-1} \mu$, where μ is the iteratively obtained solution of

$$\kappa = - a \tan \beta_{ic} + \mu , \tag{56}$$

in which

$$1 - a = \cos \Delta\beta_i = \sqrt{1 - \mu^2} = 1 - \tfrac{1}{2}\mu^2 - \frac{1}{2 \cdot 4} \mu^4 - \frac{1 \cdot 1 \cdot 3}{2 \cdot 4 \cdot 6} \mu^6 , \tag{57}$$

and letting $b_j = \cos x_j - 1,\ j = 4,7,$

$$\kappa = (b_7 + b_4 + b_7 b_4) \tan \beta_{ic} \pm \sin x_7 (1 + b_4) \cos \gamma_{ic} \pm \sin x_4 \sin \gamma_{ic} , \tag{58}$$

where the upper sign is used for $i = 1$ and the lower sign for $i = 2$. The procedure is fast, and does not require double precision arithmetic.

Initial experiments with direct generation of the P-matrix proceeded by using the classical Gershgorin Theorem (Ref. 14) and applying the results to the OAO. See the listing in Appendix III, where a positive definite matrix P is generated by simply requiring that each diagonal element is larger than the sum of the absolute values of the off diagonal elements in the same row. The computations on the PDP-10 time sharing service were so expensive that it was decided to refine the routine by using second order examples before returning to large order systems.

The results of this refinement at present are displayed in Appendix III. These routines have found quadratic Liapunov functions for the van der Pol equation

$$\ddot{x} + \epsilon(1 - x^2)\dot{x} + x = 0 \tag{59}$$

with $\epsilon = 0.1$, 1.0, and 5.0. For $\epsilon = 0.1$ the quadratic Liapunov function gives an estimate of the domain of attraction of area 4.8, while the actual area is approximately 12.5. A total of 1500 P-matrices were examined in about one minute on the PDP-10 computer, the best P-matrix found at trial number 89. Corresponding numbers for $\epsilon = 1.0$, $\epsilon = 5.0$; for the Faulkner differential equation (Ref. 15)

$$\dot{x} = 6y - 2y^2$$
$$\dot{y} = -10x - y + 4x^2 + 2xy + 4y^2 \qquad (60)$$

and the Frommer differential equation (Ref. 15)

$$\dot{x} = -y + x^2 + y^2$$
$$\dot{y} = x - 2xy \qquad (61)$$

are given in Table 3. The "actual areas" were obtained by counting squares in the figures in Davis's book (Ref. 15), and are intended for rough comparison only. Note that Frommer's equation, for which the quadratic estimate is worthless, has a linear part such that the origin is neutrally stable. The results for the Faulkner equation also appear to be discouraging, but represent a significant improvement over a previously published (Ref. 16) quadratic Liapunov function for this equation, derived through a computer study using Lie series.

Table 3

RESULTS FOR SAMPLE SECOND ORDER SYSTEMS

Equation		Area of Quadratic Estimate	Area of Actual Domain (Approx.)	Computer Time, min.	Total no. of P-matrices	Trial at which best was found
van der Pol	$\epsilon = 0.1$	4.8	12.5	1.05	1400	89
van der Pol	$\epsilon = 1.0$	3.7	13.4	2.78	4600	737
van der Pol	$\epsilon = 5.0$	4.9	26.4	0.78	1100	83
Faulkner		.03	3.14	7.62	22,000	3102
Faulkner		5×10^{-20}	0.15	1.94	1800	1

The main object of the second order studies described above was not to get answers for second order problems, but to develop a routine that would work efficiently on higher order systems. See the discussion in Appendix III of the third computer listing presented in that appendix.

The routines of Appendix III along with the methods shown in Eqs. (53) through (58) have been programmed for the IBM 360/75 and the combination shows promise of investigating over 50,000 P matrices per hour. The subroutined DER and PGGO are considerably more sophisticated than their predecessors AFX and QGEN, respectively. The former avoid double precision arithmetic and are over ten times faster than the latter. Additional compactness and speed is obtained by generation of points x by letting $x = Cy$ as in Eq. (51), where

$$y_i = \xi_i R \Big/ \left(\sum_{j=1}^{n} \xi_i^2 \right)^{\frac{1}{2}} ,$$

in which $R, \xi_1, \ldots, \xi_n$ are independent, each ξ_i is uniformly distributed on $(-1, +1)$, and R is so distributed that $\mathrm{Prob}(R < r) = r^n$. This procedure generates uniformly (by volume) distributed random points y without discarding points as mentioned following Fig. 16.

Some computer runs were also made by simply modifying the previously described PICK-A-Q routine so that it generated P matrices instead of Q matrices. There has been no real success from the PICK-A-P system.

There has been no real success in running the PICK-A-P system for the following reason. As a starting point, a particular selection of 45 parameters is undertaken in order to arrive at a p.d.

(positive-definite) P-matrix from which the search of the state space begins. Since $\dot{V}$ is a required variable in the search and is a function of Q, a p.d. Q is formed by

$$Q = -A^T P - PA \quad .$$

Initially, the p.d. P did not yield a p.d. $\dot{Q}$, and it is not necessary that it should. After a few hours of computer time in which 35,000 p.d. P-matrices were tried with no success (i.e., no p.d. Q-matrices), an alternative was undertaken.

The alternative was to take the best P from the best p.d. Q of the PICK-A-Q program, factor it into the required 45-input variables, and use these as input to the PICK-A-P program. This was done, but the reconstruction of the 45 variables to arrive at the p.d. P (QGEN-Subroutine) was deficient in that it did not yield the original p.d. P. This difficulty has not been resolved in terms of subroutine QGEN, and hence the negative results.

Another subroutine (PGGO), which utilizes the same factorization as mentioned above, to arrive at the 45 variables, yields the desired p.d. P. The exact reasons for the difference between PGGO and QGEN have not been ascertained to date.

V. FURTHER RESEARCH

To briefly summarize the research described here, an effort has been made to prove the stability of a ninth order system arising in engineering practice. Simulation, experience, and intuition give convincing evidence that the OAO is globally stable, within the momentum capacities of the wheels. Mathematical proofs of this stability have continually failed, but always, it seems, because of some subtle mathematical triviality rather than real physical attributes of the system or important characteristics of the system model. Brute force computer studies using quadratic Liapunov functions and random searches to estimate the domain of attraction have continually given estimates of the domain which extend well into the nonlinear region, but are disappointingly small compared to simulation studies. One consolation is that a conservative estimate of the volume of the domain has been found for a real ninth order system and this represents a first. However, future work based on the present foundations should be more definitive.

Many questions of interest have been opened by the research described herein. The most fundamental questions involve: 1) the basic effectiveness of the widely used quadratic Liapunov functions for estimating actual domains of attraction and 2) methods of random search that will succeed in high dimensional problems. Insight into both questions can be obtained by developing a numerical algorithm that efficiently generates quadratic Liapunov functions and simultaneously assures that the best quadratic estimate of the domain of attraction has been obtained. Experience in developing such an algorithm has provided information about random and deterministic search techniques. The algorithm itself gives experimental information on quadratic estimation of the domain of attraction. This information, it is hoped, will further stimulate more theoretical approaches, in particular, approaches that will lead to more sophisticated methods for directly describing the domain of attraction, a complicated set in Euclidian n-space.

Five very specific areas require immediate study: 1) methods of factoring a positive definite matrix into its diagonal eigenvalue matrix and its rotation matrices;[†] 2) the development of an algorithm to shrink the search in n-space to a set of points close to x^* and inside the estimated domain $x^T Px \leq \ell$ when a point x^* is discovered such that $\dot{V}(x^*) = 0$; 3) investigate more fully "almost diagonal" P-matrices (such matrices have provided the best estimates to date), 4) examine the possibility of describing the domain as a union of hyperellipses and possibly hyperannuli; and 5) construction of a neater algorithm for selecting random points inside a given domain in Euclidean n-space.

The above areas should not be regarded as main future research goals, but as representing ways in which further insight into the larger problem might be obtained, thereby leading to a better understanding of stable systems and models. The computer programs that have constituted the main emphasis in this report are regarded as tools leading to comprehension rather than as research goals.

One point should be emphasized about the viewpoint at Grumman. The most profound stability theorems presently available in the literature all relate to systems with a single nonlinearity or within a specific class of nonlinearities. The case under investigation at Grumman involves many nonlinearities that do not fall

---

[†]The converse construction problem is described in Ref. 3.

into these classes. The eventual hope is to develop theorems that identify stable systems with multiple nonlinearities, but our method is experimental rather than theoretical. It is based on the conviction that proving theorems in this area will become possible only after experimental evidence has presented a deeper understanding of the factors that cause a system to be stable.

VI. SUMMARY

In this report we have reviewed the development of the state equations of the OAO "paired-Tracker" coarse pointing mode attitude control system and have displayed some simplifications of this model, viz., motor saturation only, and six dimensional approximation. These models were compared by simulation for relatively large initial conditions and were found to be essentially similar in behavior, with the exception of one unexplained difference in yaw response for the full model and the six dimensional approximation.

The motor saturation only model was analyzed using the Popov Theory, and it was shown that when the channels are linearly uncoupled, i.e., $\beta_{1c} = 0$, each channel is absolutely stable when the initial total momentum is within the wheel capacity, if the gain of the nonlinearity does not go to zero with large argument. Unfortunately, this is not the case for a saturation and so we can only conclude that the domain of attraction is large, but that the channels are not globally stable. This difficulty arises because of a pole at the origin in the transfer function of the linear part. This also prevents successful analysis of the coupled system via the methods of Moore and Anderson, Sandberg, and Yacubovich.

The algorithm (Pick a Q-Matrix) for estimating the domain of attraction was developed and random search techniques for solving the required minimization problem and maximization problem were developed. The former problem was solved very successfully by taking advantage of the known geometry of the problem. The latter problem was solved with some degree of success by using a "creeping accelerated random search." However, because of the unknown geometry of this latter problem and its higher dimensionality (45 versus 9), similar success was not achieved. The algorithm was tested on both the nine and six dimensional full nonlinear models with similar results. The results are disappointing by comparison to the simulation results; however, they are significantly beyond the region in which the nonlinear effects first become dominant.

The algorithm was also tried on a prototype two dimensional problem to demonstrate that the algorithm would work successfully on a saturated system with a critical linear part, but the results are very sensitive to the matrix rotation parameter. A second algorithm (Pick a P-Matrix) was formulated based upon a more heuristic approach and gave some promising results for two dimensional systems, but failed to produce any results for the nine dimensional problem. This failure was apparently due to our inability to obtain a good starting matrix by factoring the best P-matrix obtained via the first algorithm.

The review of required research shows that this approach (optimal quadratic estimation) is still a promising one, but illuminates some research problems of significance. Particularly, these are: the need for more effective search techniques for problems of high dimension, a method for determining the fundamental limitations of our approach, and a more direct procedure for estimating the domain of attraction.

VII. REFERENCES

1. Doolin, B. F. and Showman, R. D., "Attitude Stabilization of Satellites with Gimbaled Star Trackers," presented at Second IFAC Symposium on Automatic Control in Space, Vienna, Austria, September 1967.

2. Showman, R. D., "Simplified Processing of Star Tracker Commands for Satellite Attitude Control," presented at IEEE Intl. Conv., New York, N.Y., March 1967; also IEEE Trans. on A.C., Vol. AC-12, No. 4, August 1967.

3. Geiss, G., Cohen, V., and Rothschild, D., Development of an Algorithm for the Nonlinear Stability Analysis of the Orbiting Astronomical Observatory Control System, Grumman Research Department Report RE-307, November 1967.

4. LaSalle, J. P. and Lefschetz, S., Stability by Liapunov's Direct Method with Applications, Academic Press, New York, 1961.

5. Feldman, W. and Geiss, G., The Effects of Angular Offset at Equilibrium on the Linear Model of the OAO "Paired-Tracker" Control System, Grumman Research Department Memorandum RM-387, November 1967.

6. Chomas, A. and Geiss, G., Derivation of OAO Coarse Pointing Mode Model in State Variable Form, Grumman Research Department Memorandum RM-415, June 1968.

7. Aizerman, M. A. and Gantmacher, F. R., Absolute Stability of Regulator Systems, trans. E. Polak, Holden-Day, Inc., San Francisco, 1964, p. 52.

8. Moore, J. B. and Anderson, B. D. O., "A Generalization of the Popov Criterion," Franklin Institute Journal, Vol. 285, No. 6, June 1968.

9. Sandberg, I. W., "On the Boundedness of Solutions of Nonlinear Integral Equations," Bell System Technical Journal, Vol. 44, pp. 439-453, 1965.

10. Yakubovich, V. A., "Frequency Conditions for the Absolute Stability of Control Systems with Several Nonlinear or Linear Nonstationary Blocks," Automation and Remote Control, No. 6, June 1967, pp. 857-880.

11. Rothschild, D. and Geiss, G., Parameterization of the Set of Positive Definite Matrices and An Algorithm for Its Generation, Grumman Research Department Memorandum RM-386, November 1967.

12. Barron, R. L., "Inference of Vehicle and Atmosphere Parameters from Free Flight Motions," presented at AIAA Guidance, Control and Flight Dynamics Conference, Huntsville, Alabama, August 14-16, 1967, AIAA Paper 67-600.

13. Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Co., New York, 1962, p. 400.

14. Householder, A. S., "Norms and the Localization of Roots of Matrices," Bull. Amer. Math. Soc., Vol. 74, 1968, p. 822.

15. Davis, H. T., Introduction to Nonlinear Differential and Integral Equations, USAEC, 1960, p. 344 ff.

16. Burnand, G. and Sarlos, G., "Determination of the Domain of Stability," J. Math. Anal. and Appl., Vol. 23, pp. 714-722, 1968.

APPENDIX I

DETAILED DESCRIPTION OF ESTIMATION ALGORITHMS

This appendix contains a flow chart and listing of the stability analysis algorithms where a Q-matrix is initially selected and where a P-matrix is initially selected. Both algorithms are basically the same, the primary difference being that in the Q-matrix selection an inversion process is required to determine the associated P-matrix (which is assuredly positive definite), whereas in the P-matrix selection the resulting Q-matrix (not necessarily positive definite) found by a matrix multiplication must be tested to ascertain its character. All Q-matrices that result from picking a P-matrix must be discarded if they prove to be semidefinite or negative definite because of the theory being utilized.

A thumbnail sketch of each of the subroutines shown in the flow charts (Figs. I-1 and I-2) follows.

Subroutine AFX

This subroutine calculates the matrix A and the nonlinear vector $f(x)$ of the equations of motion $\dot{x} = Ax + f(x)$.

Subroutine QGEN

This subroutine generates a positive definite matrix Q given a set of $n(n+1)/2$ independent variables as given in Ref. 3.

Subroutine DSRCH

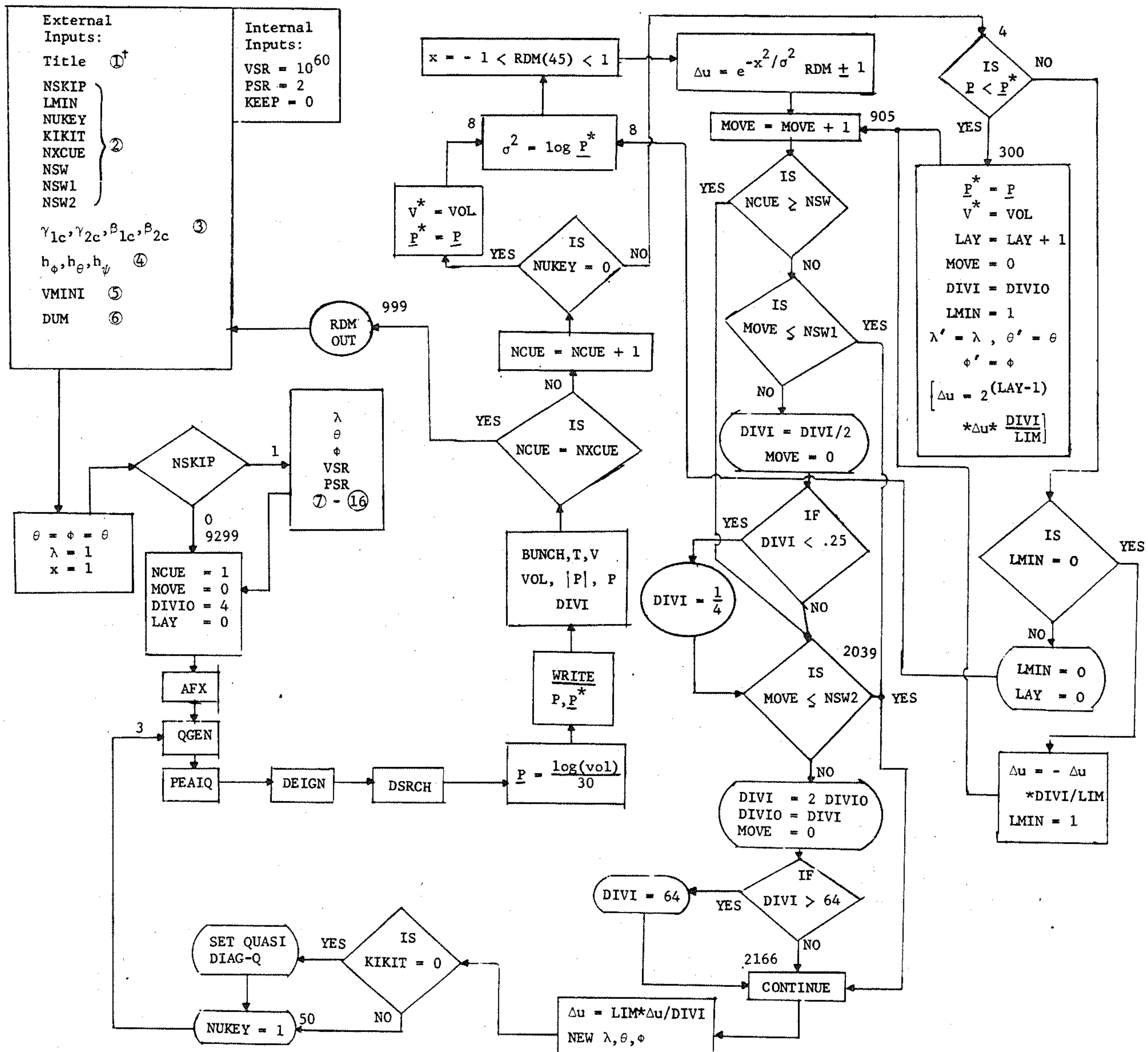
This subroutine is the search subroutine for the state space where a $\min V$ with $\dot{V} = 0$ is to be achieved.

Subroutine PEAIQ

This subroutine solves the equation $A^T P + PA = - Q$ for a positive definite P-matrix, given a stable A-matrix and a positive definite Q-matrix.

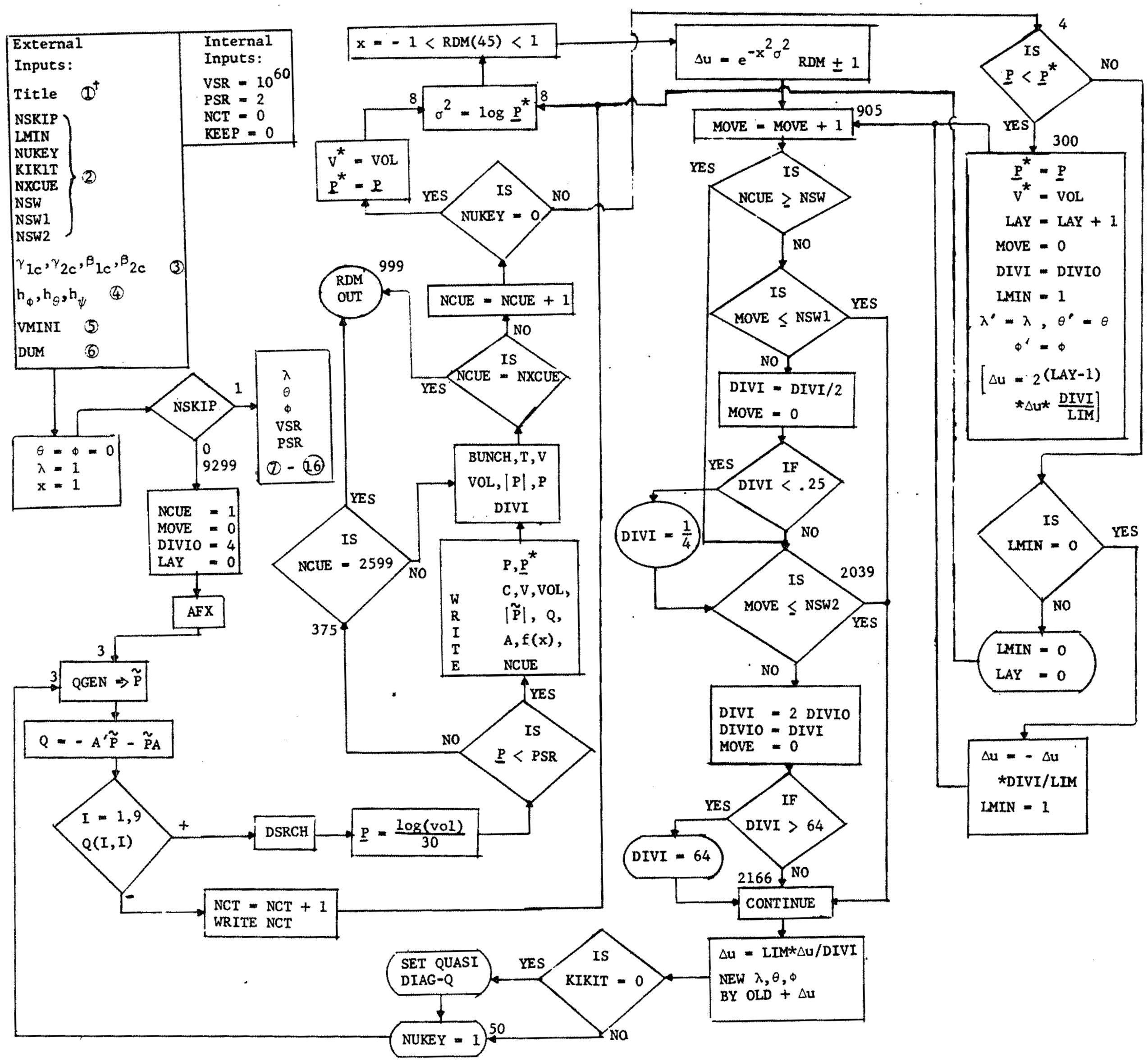
Subroutine DEIGN

This subroutine calculates the eigenvalues and eigenvectors of the positive definite P-matrix.



†ⓝ - input card numbers

Fig. I-1 Flow Chart for Algorithm Based on a Q-Matrix



†ⓝ - card numbers

Fig. I-2 Flow Chart for Algorithm Based on a P-Matrix

LISTINGS FOR PICK-A-Q

```
LEVEL 2 FEB 67                              OS/360  FORTRAN H                              DATE  69.199/15.46.49

       COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
 ISN 0002             IMPLICIT REAL*8 (A-H,O-Z)                                        10100010
           C          Q - OPTIMIZATION PROGRAM      8-20-68   INITIATE                 10100020
 ISN 0003             DIMENSION THETA(28),PHIV(8),XLAM(9),RHV(9),SLUF(9)               10100030
 ISN 0004             DIMENSION  XLMX(9)                                               10100040
 ISN 0005             DIMENSION PRM(9,9)                                               10100060
 ISN 0006             REAL*4  B,C,DUM,GUM                                              10100070
 ISN 0007             DIMENSION ALP(18)                                                10100080
 ISN 0008             DIMENSION DU(45),RX(45),NUTU(45),XLAMF(9),THETF(28),PHIVP(8)     10100090
 ISN 0009             DIMENSION EX(9),XINC(9)                                          10100100
 ISN 0010             DIMENSION DINC(9)                                                10100110
 ISN 0011             DIMENSION BUNCH(3001,6)                                          10100120
 ISN 0012             DIMENSION B(9),C(9,9)
 ISN 0013             COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI          10100130
 ISN 0014             COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM          10100140
 ISN 0015             COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C                            10100150
 ISN 0016             COMMON /BLK4/ HPHI,HTHT,HPSI                                     10100160
 ISN 0017             COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)                 10100170
 ISN 0018             COMMON/BLK 11/DB1,DB2,DG1,DG2                                    10100180
 ISN 0019              COMMON/BLK77/X(15)                                              10100190
 ISN 0020             COMMON/BLK78/X1E,X4E,X7E                                         10100200
 ISN 0021             COMMON/BLK68/ T                                                  10100210
 ISN 0022             COMMON/ABORT/ DET, VSR ,VOL                                      10100220
 ISN 0023             COMMON/BLK 70/ AAR(9,9),BM(9,9),PA(9,9),ATP(9,9)                 10100230
 ISN 0024             COMMON/GEORGE/INIT
 ISN 0025             COMMON/BLKDS/VDOTZ,XZERO(9),FZERO(9),NNZERO
 ISN 0026             CALL SETCLK
 ISN 0027        1001 FORMAT(5E14.7)                                                   10100240
 ISN 0028         102 READ(5,1101,END=1000) ALP                                        10100250
 ISN 0029        1101 FORMAT(18A4)                                                     10100260
 ISN 0030             WRITE(6,191)(ALP(J),J=1,18)                                      10100270
 ISN 0031         191 FORMAT(1H / 10X,18A4)                                            10100280
 ISN 0032             N=9                                                              10100290
 ISN 0033             KEEP = 0                                                         10100300
 ISN 0034         751 TM     = 76.8                                                    10100310
 ISN 0035             T1     =  4.5                                                    10100320
 ISN 0036             T2     =  0.5                                                    10100330
 ISN 0037             XKM    =  1.0/13.0                                               10100340
 ISN 0038             XKC    =  2.685 E+05                                             10100350
 ISN 0039             A13    =  0.0                                                    10100360
 ISN 0040             A11      = 0.0                                                   10100370
 ISN 0041             AITREN= 1500.0                                                   10100380
 ISN 0042             F2LIM = 26.0                                                     10100390
 ISN 0043             READ(5,1001)GAM1C,GAM2C,BET1C,BET2C                              10100400
 ISN 0044             READ(5,1001)HPHI,HTHT,HPSI                                       10100410
 ISN 0045             READ(5,1001)VMINI                                                10100420
 ISN 0046             READ(5,712) DUM                                                  10100430
 ISN 0047         712 FORMAT(Z8)                                                       10100440
 ISN 0048             READ(5,1104)NSKIP,LMIN,NUKEY,KIKIT,NXCUE,NSW,NSW1,NSW2
 ISN 0049        1104 FORMAT(18I4)
           C          NSKIP = 0 FOR NO P-MATRIX ELEMENT INPUT -- STARTS WITH UNIT MATRIX
           C          NSKIP = 1 INPUT 10 CARDS WITH XLAM,THETA,PHIV,PSR AND VSR
           C          IF VSR AND PSR ARE GIVEN  THEN  NUKEY=LMIN=1
           C          KIKIT = 0---YIELDS TRI-DIAGONAL Q
           C          NXCUE MUST BE AT LEAST 1 LESS THAN THE LARGE DIMENSION OF BUNCH
           C          NXCUE - NSW = SHRINKING PORTION = OF POINTS
```

```
                C      NSW = = OF POINTS IN THE EXPANDING PORTION OF SEARCH
                C      NSW1 = = OF PTS BEFORE DIVISOR SHRINKS IN EXPANDING SEARCH
                C      NSW2 = = OF PTS BEFORE DIVISOR DOUBLES IN CONTRACTING SEARCH
ISN 0050               IF(DUM)713,713,714                                            10100450
ISN 0051           714 CALL RDMIN(DUM)                                               10100460
ISN 0052           713 CONTINUE                                                      10100470
ISN 0053               WRITE(6,81)VMINI                                              10100480
ISN 0054            81 FORMAT ( 7H VMINT= E14.7)                                     10100490
ISN 0055               PI= 3.1415926                                                 10100500
ISN 0056               PI2 = PI/2.                                                   10100510
ISN 0057               DTR = PI/180.                                                 10100520
ISN 0058               GAM1C = GAM1C * DTR                                           10100530
ISN 0059               GAM2C = GAM2C * DTR                                           10100540
ISN 0060               BET1C = BET1C * DTR                                           10100550
ISN 0061               BET2C = BET2C * DTR                                           10100560
ISN 0062               DIF = GAM1C - GAM2C                                           10100570
ISN 0063               D12 =  DIF * 2.0 / DABS(DIF)                                  10100580
ISN 0064               SDIF=SIN(DIF)                                                 10100590
ISN 0065               DO 201 I=1,28                                                 10100600
ISN 0066           201 THETA(I) = 0.                                                 10100610
ISN 0067               DO 202 I=1,8                                                  10100620
ISN 0068           202 PHIV(I) = 0.0                                                 10100630
ISN 0069               DO 203 I=1,9                                                  10100640
ISN 0070               XLMX(I) = 1.0                                                 10100650
ISN 0071               X(I) = 1.0                                                    10100660
ISN 0072           203 XLAM(I)= 1.0                                                  10100670
ISN 0073               NCUE = 1                                                      10100920
ISN 0074               IF(NSKIP.EQ.0) GO TO 9299
ISN 0076               READ(5,1001) (XLAM(I),I=1,9),(THETA(J),J=1,28),(PHIV(K),K=1,8)
                      1  ,PSR,VSR
ISN 0077          9299 DO 444 I=1,N
ISN 0078               DO 444 J=1,N                                                  10100940
ISN 0079               ATP(I,J)=0.0                                                  10100950
ISN 0080           444 PA(I,J)=0.0                                                   10100960
ISN 0081               DO 6009 I=1,9                                                 10100970
ISN 0082          6009 EX(I) = DLOG(XLAM(I))                                         10100980
ISN 0083               DO 420 I=1,9                                                  10100990
ISN 0084           420 XLAMP(I) = XLAM(I)                                            10101000
ISN 0085               DO 430 I=1,28                                                 10101010
ISN 0086           430 THETP(I) = THETA(I)                                           10101020
ISN 0087               DO 440 I=1,8                                                  10101030
ISN 0088           440 PHIVP(I) = PHIV(I)                                            10101040
ISN 0089               WRITE(6,550)(XLAMP(I),I=1,9),(THETP(I),I=1,28),(PHIVP(I),I=1,8)   10101050
ISN 0090               SG1C = SIN(GAM1C)                                             10101060
ISN 0091               CG1C = COS(GAM1C)                                             10101070
ISN 0092               SG2C = SIN(GAM2C)                                             10101080
ISN 0093               CG2C = COS(GAM2C)                                             10101090
ISN 0094               TB1C = SIN(BET1C)/COS(BET1C)                                  10101100
ISN 0095               X1E = ( AITREN)/(XKM*XKC)*(HPHI-HTHT*(A11*SG1C-A13*SG2C-CG1C*TB1C 10101110
                      1 )/(D12*SDIF) +HPSI*(A11*CG1C-A13*CG2C+SG1C*TB1C)/(D12*SDIF))    10101120
ISN 0096               X4E = ( AITREN)/(XKM*XKC)*(HTHT/(D12*SDIF))                   10101130
ISN 0097               X7E = (-AITREN)/(XKM*XKC)*(HPSI/(D12*SDIF))                   10101140
ISN 0098               PHI = (X(1)+X1E)/DTR                                          10101150
ISN 0099               VPHI = X(2)*(XKM*XKC)+ HPHI*AITREN                            10101160
ISN 0100               WPHI = X(3)*XKC*T1/T2  - T1 *AITREN*HPHI/(T2*XKM)             10101170
ISN 0101               THT = (X(4)+X4E)/DTR                                          10101180
```

```
ISN 0102          VTHT = X(5)*XKM*XKC +  HTHT*AITREN                              10101190
ISN 0103          WTHT = X(6)*XKC*T1/T2  - T1* AITREN*HTHT/(T2*XKM)               10101200
ISN 0104          PSI = (X(7)+X7E)/DTR                                            10101210
ISN 0105          VPSI = X(8)*XKM*XKC + HPSI*AITREN                               10101220
ISN 0106          WPSI = X(9)*XKC*T1/T2  - T1*AITREN*HPSI/(T2*XKM)                10101230
ISN 0107          VMAX = 0.0                                                      10101240
ISN 0108          NPUSS = 2                                                       10101250
ISN 0109          JSW = 0                                                         10101260
ISN 0110          MOVE = 0                                                        10101270
ISN 0111          KARP = 1                                                        10101280
ISN 0112          XLLIM = 6.9                                                     10101290
ISN 0113          PHLIM = PI                                                      10101300
ISN 0114          THLIM = PI2                                                     10101310
ISN 0115          DIVIO = 4.                                                      10101320
ISN 0116          DIVI = DIVIO                                                    10101330
ISN 0117          PMIN  = 30.                                                     10101340
ISN 0118          LAY=0                                                           10101350
ISN 0119          NOPE = 0                                                        10101390
ISN 0120          CALL AFX(JSW)                                                   10101410
ISN 0121        3 CALL QGEN(THETA,PHIV,XLAM)                                      10101430
ISN 0122          DET = 1.0                                                       10101440
ISN 0123          IF(NOPE.EQ.1) GO TO 99                                          10101450
ISN 0125          CALL PEAIQ(KEEP)                                                10101470
ISN 0126          KEEP = 1                                                        10101480
            C                                                                     10101490
ISN 0127          DO 703 I=1,9                                                    10101510
ISN 0128          DO 703 J=1,9                                                    10101520
ISN 0129      703 PRM(I,J)=PM(I,J)                                                10101530
ISN 0130          CALL DEIGN(PM,B,C)                                              10101550
ISN 0131          DO 6 I=1,9                                                      10101560
ISN 0132        6 RHV(I) = 0.0                                                    10101570
ISN 0133          KDET = IMEQD(9,9,1,PRM,RHV,DET,SLUF)                            10101590
ISN 0134          CALL CLOCK
ISN 0135          WRITE(6,9061)NCUE,DET                                           10101610
ISN 0136     9061 FORMAT(' NCUE = ',I5 ,   ' DET-P = ',E14.7 )                    10101620
            C     NEGATIVE EIGEN VALUE ABORT                                      10101630
ISN 0137          DO 21 I=1,9                                                     10101640
ISN 0138       21 IF(B(I).LT.0.0) GO TO 22                                        10101650
ISN 0140          GO TO 445                                                       10101660
ISN 0141       99 CONTINUE                                                        10101670
ISN 0142          DO 704 I=1,9                                                    10101680
ISN 0143          B(I) = XLAM(I)                                                  10101690
ISN 0144          DET = DET * XLAM(I)                                             10101700
ISN 0145          DO 704 J=1,9                                                    10101710
ISN 0146          PM(I,J) = Q(I,J)                                                10101720
ISN 0147      704 C(I,J)  = AAR(I,J)                                              10101730
ISN 0148          KDET = 0                                                        10101740
ISN 0149          WRITE(6,9061)KDET,DET                                           10101750
            C     CALCULATION OF ATP (A TRANSPOSE P )                             10101760
            C     CALCULATION OF PA  (P-MATRIX X A )                              10101770
ISN 0150          DO 26 I=1,N                                                     10101780
ISN 0151          DO 26 J=1,N                                                     10101790
ISN 0152          DO 26 K=1,N                                                     10101800
ISN 0153          ATP(I,J) = ATP(I,J) + A(K,I) *PM(K,J)                           10101810
ISN 0154       26 PA(I,J)  = PA(I,J)  +PM(I,K) * A(K,J)                           10101820
ISN 0155          DO 27 I=1,N                                                     10101830
```

```
ISN 0156          DO 27 J=1,N                                                10101840
ISN 0157       27  Q(I,J) = -ATP(I,J) - PA(I,J)                              10101850
ISN 0158          WRITE(6,9765)                                              10101860
ISN 0159     9765 FORMAT(1H / 1X,36H  Q FROM PUTTING P INTO -ATP-PA = Q , //) 10101870
ISN 0160          DO941 I=1,N                                                10101880
ISN 0161      941 WRITE(6,2963) ( Q(I,J),J=1,N)                              10101890
ISN 0162     2963 FORMAT(1H /(1X,9E14.7) )                                   10101900
ISN 0163      445 VMIN = VMINI                                               10101910
ISN 0164          CALL DSRCH(VMIN,B,C,JSUE)                                  10101930
            C                                             EUSJ )6993.6(ETI   10101940
ISN 0165     3996 FORMAT(  ' JSUE = ',I5 //)                                 10101950
ISN 0166          VL= VMIN                                                   10101970
ISN 0167          P = DLOG10(VOL)/30.                                        10102010
ISN 0168      580 FORMAT( ' P =',E14.7,20X,'PSR=',E14.7//)                   10102020
ISN 0169          WRITE(6,580) P,PSR                                         10102030
ISN 0170          BUNCH(NCUE,1) = T                                          10102040
ISN 0171          BUNCH(NCUE,2) = VL                                         10102050
ISN 0172          BUNCH(NCUE,3) = VCL                                        10102060
ISN 0173          BUNCH(NCUE,4) = DET                                        10102070
ISN 0174          BUNCH(NCUE,5) = P                                          10102080
ISN 0175          BUNCH(NCUE,6) = DIVI                                       10102090
ISN 0176          IF(NCUE.EQ.NXCUE) GO TO 999                                10102100
ISN 0178          NCUE = NCUE + 1                                            10102110
ISN 0179          IF(NUKEY.NE.0) GO TO 4                                     10102120
ISN 0181          PSR = P                                                    10102130
ISN 0182          VSR = VOL                                                  10102140
ISN 0183          WRITE(6,570) PSR                                           10102150
ISN 0184      570 FORMAT(  ' P-STAR = ',E14.7/)                              10102160
ISN 0185        8 SIG2 = DLOG10(PSR)                                         10102170
ISN 0186       22 DO 9 I=1,45                                                10102200
ISN 0187          RX(I) = RDM(GUM)                                           10102210
ISN 0188          XLUV    = -1.+ 2.*RX(I)                                    10102220
ISN 0189          IF(XLUV.LE.0.0) GO TO 3415                                 10102230
ISN 0191          NUTU(I) = 1                                                10102240
ISN 0192          GO TO 9                                                    10102250
ISN 0193     3415 NUTU(I) =-1                                                10102260
ISN 0194        9 DU(I) =DEXP(-RX(I)**2/SIG2) * NUTU(I)                      10102270
            C     WRITE(6,560)(DU(I),I=1,45)                                 10102290
ISN 0195      560 FORMAT( ' DU = ' //(1X,9E13.6/))                           10102300
ISN 0196      905 CONTINUE                                                   10102310
ISN 0197          MOVE = MOVE + 1                                            10102320
ISN 0198          IF(NCUE.GE.NSW) GO TO 2039                                 10102330
ISN 0200          IF(MOVE.LE.NSW1)GO TO 2166                                 10102340
ISN 0202          DIVI = 0.5*DIVI                                            10102350
ISN 0203          MOVE = 0                                                   10102360
ISN 0204          IF (DIVI.LT.0.25) DIVI = 0.25                              10102370
ISN 0206          GO TO 2166                                                 10102380
ISN 0207     2039 IF(MOVE.LE.NSW2)GO TO 2166                                 10102390
ISN 0209          DIVI = 2.* DIVIO                                           10102400
ISN 0210          DIVIO = DIVI                                               10102410
ISN 0211          MOVE = 0                                                   10102420
ISN 0212          IF (DIVI.GT.64.) DIVI = 64.                                10102430
            C2166 WRITE(6,2332) DIVI                                         10102440
ISN 0214     2165 CONTINUE
ISN 0215     2332 FORMAT(  ' DIVI = ', E20.7 /)                              10102450
ISN 0216          DO 10 I=1,9                                                10102460
```

```
ISN 0217         10 DU(I) = XLLIM * DU(I)/DIVI                                  10102470
ISN 0218            DO 12 I=10,37                                               10102480
ISN 0219         12 DU(I) = THLIM * DU(I)/DIVI                                  10102490
ISN 0220            DO 14 I=38,45                                               10102500
ISN 0221         14 DU(I) = PHLIM * DU(I)/DIVI                                  10102510
             C      WRITE(6,560)(DU(I),I=1,45)                                  10102530
ISN 0222            DO 20 I=1,9                                                 10102550
ISN 0223        665 XINC(I) = EX(I) + DU(I)                                     10102560
ISN 0224            IF(XINC(I).LT.XLLIM) GO TO 209                              10102570
ISN 0226            DU(I) = 0.5 * DU(I)                                         10102580
ISN 0227            GO TO 665                                                   10102590
ISN 0228        209 IF(XINC(I).LT.-9.2) XINC(I)=-9.2                            10102600
ISN 0230         20 XLAM(I) =DEXP(XINC(I))                                      10102610
ISN 0231        666 DO 30 I=1,28                                                10102620
ISN 0232            IP = I+9                                                    10102630
ISN 0233         52 THETA(I)= THETP(I) + DU(IP)                                 10102640
ISN 0234            IF( DABS(THETA(I)).LT.PI2) GO TO 30                         10102650
ISN 0236            THETA(I) = THETA(I) - DU(IP)                                10102660
ISN 0237         44 DU(IP) = 0.5* DU(IP)                                        10102670
ISN 0238            GO TO 52                                                    10102680
ISN 0239         30 CONTINUE                                                    10102690
ISN 0240            DO 40 I=1,8                                                 10102700
ISN 0241            IP= 37+I                                                    10102710
ISN 0242         62 PHIV(I) = PHIVP(I) + DU(IP)                                 10102720
ISN 0243            IF(DABS(PHIV(I)).LT.PI) GO TO 40                            10102730
ISN 0245            PHIV(I) = PHIV(I) - DU(IP)                                  10102740
ISN 0246         54 DU(IP) =  0.5 * DU(IP)                                      10102750
ISN 0247            GO TO 62                                                    10102760
ISN 0248         40 CONTINUE                                                    10102770
             C      WRITE(6,560)(DU(I),I=1,45)                                  10102790
ISN 0249            IF(KIKIT.GT.0) GO TO 50                                     10102810
ISN 0251            DO 71 I=1,5                                                 10102820
ISN 0252            PHIV(I) = 0.0                                               10102830
ISN 0253         71 THETA(I) = 0.0                                              10102840
ISN 0254            DO 72 I=8,12                                                10102850
ISN 0255         72 THETA(I) = 0.0                                              10102860
ISN 0256            DO 73 I= 14,20                                              10102870
ISN 0257         73 THETA(I) = 0.0                                              10102880
ISN 0258            THETA(23) =0.0                                              10102890
ISN 0259            THETA(24) =0.0                                              10102900
ISN 0260            DO 74 I=26,27                                               10102910
ISN 0261         74 THETA(I) = 0.0                                              10102920
ISN 0262            PHIV(6) =0.0                                                10102930
ISN 0263            THETA(28) = THETA(22)                                       10102940
ISN 0264            PHIV(7) = THETA(21)                                         10102950
ISN 0265            PHIV(8) =THETA(25)                                          10102960
ISN 0266         50 NUKEY = 1                                                   10102980
ISN 0267            GO TO 3                                                     10102990
ISN 0268          4 IF(P.LT.PSR) GO TO 300                                      10103000
ISN 0270            IF(LMIN.EQ.0)    GO TO 305                                  10103010
ISN 0272            LMIN = 0                                                    10103020
ISN 0273            LAY= 0                                                      10103030
ISN 0274            GO TO 8                                                     10103050
ISN 0275        305 DO 840  I = 1,9                                             10103060
ISN 0276        840 DU(I) =-DU(I) * 1.* DIVI / XLLIM                            10103070
ISN 0277            DO 850  I = 10,37                                           10103080
```

```
ISN 0278     850 DU(I) =-DU(I) * 1.* DIVI / THLIM                                  10103090
ISN 0279         DO 860  I = 38,45                                                 10103100
ISN 0280     860 DU(I) =-DU(I) * 1.* DIVI / PHLIM                                  10103110
ISN 0281         LMIN = 1                                                          10103120
ISN 0282         GO TO 905                                                         10103140
ISN 0283     300 PSR = P                                                           10103150
ISN 0284         VSR = VOL                                                         10103160
ISN 0285         LAY = LAY+1                                                       10103170
ISN 0286         MOVE = 0                                                          10103180
ISN 0287         DIVI = DIVIO                                                      10103190
ISN 0288         LMIN = 1                                                          10103200
ISN 0289         DO 310 I=1,9                                                      10103220
ISN 0290         EX(I) = DLOG(XLAM(I))                                             10103230
ISN 0291     310 XLAMP(I) = XLAM(I)                                                10103240
ISN 0292         DO 320 I=1,28                                                     10103250
ISN 0293     320 THETP(I) = THETA(I)                                               10103260
ISN 0294         DO 330 I= 1,8                                                     10103270
ISN 0295     330 PHIVP(I) = PHIV(I)                                                10103280
ISN 0296         WRITE(6,550)(XLAMP(I),I=1,9),(THETP(I),I=1,28),(PHIVP(I),I=1,8)   10103300
ISN 0297     550 FORMAT( ' XLAM-PRIME  ,  THETA-PRIME  , FHI-PRIME'//(1X,9E13.6/))  10103310
ISN 0298         WRITE(6,2503) SIG2                                                10102180
ISN 0299    2503 FORMAT( ' SIGMA-SQUARED = ',E14.7//)                              10102190
ISN 0300         WRITE(6,100) VL,VOL                                               10101980
ISN 0301     100 FORMAT(1H /1X, 16H LIAPUNOV FCT  = E14.7 ,/                       10101990
               1          1X,23H INVERSE VOL ESTIMATE = E14.7,/ )                  10102000
ISN 0302    1020 WRITE(6,1030)(B(I),I=1,9)                                         10300200
ISN 0303    1030 FORMAT(13H1 EIGENVALUES//(1P6E20.7))                              10300210
ISN 0304         WRITE(6,700)((C(I,J),J=1,9),I=1,9)                                10300270
ISN 0305     700 FORMAT(13H EIGENVECTORS/(9E12.4))                                 10300280
ISN 0306         WRITE(6,93) (( Q(I,J),J=1,N),I=1,N)                               10901280
ISN 0307      93 FORMAT(1H /1X,7H Q(I,J)/(1X,9E14.7) )                             10901290
            C    WRITE(6,1200) (FZERO(I),I=1,9),(XZERO(J),J=1,9)                   10701080
ISN 0308    1200 FORMAT(10H FZERO(I)=,9E12.4/10H XZERO(I)=,9E12.4)                 10701090
            C    WRITE(6,1201)NNZERO,VDOTZ,VMIN                                    10701100
ISN 0309    1201 FORMAT(8H NNZERO=,I6/12H VDOTZ,VMIN=,2E14.7)                      10701110
ISN 0310         DO 810  I = 1,9                                                   10103330
ISN 0311     810 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /XLLIM                         10103340
ISN 0312         DO 820  I = 10,37                                                 10103350
ISN 0313     820 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /THLIM                         10103360
ISN 0314         DO 830  I = 38,45                                                 10103370
ISN 0315     830 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /PHLIM                         10103380
ISN 0316         GO TO 905                                                         10103400
ISN 0317     999 CALL RDMOUT(DUM)                                                  10103410
ISN 0318         WRITE(6,715) DUM                                                  10103420
ISN 0319     715 FORMAT(1H0 Z8)                                                    10103430
ISN 0320         GO TO 102                                                         10103440
ISN 0321    1000 CONTINUE                                                          10103450
ISN 0322         WRITE(6,191)(ALP(J),J=1,18)                                       10103460
ISN 0323         WRITE(6,8600)                                                     10103470
ISN 0324    8600 FORMAT(18X,'TIME',12X,'LIAPUNOV FCT',9X,'INVERSE VOL',11X,'DET(P)' 10103480
               1,11X,'PERFORMANCE',12X,'DIVISOR',/// )                             10103490
ISN 0325         WRITE(6,8605) ((BUNCH(I,J),J=1,6),I=1,NXCUE)                      10103500
ISN 0326    8605 FORMAT (7X,6(6X,E14.7)/)                                          10103510
ISN 0327         CALL EXIT                                                         10103520
ISN 0328         END                                                               10103530
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002           SUBROUTINE DSRCH(VMIN,B,C,JSUE)                              10700010
ISN 0003           IMPLICIT REAL*8 (A-H,O-Z)                                    10700020
ISN 0004           DIMENSION G(9),B(9),GG(9),XX(9),C(9,9),PX(9)                 10700030
                  1,PF(9),QX(9),DEL(9)                                          10700040
ISN 0005           REAL*4  B,C,R1,R2                                            10700060
ISN 0006           COMMON/BLKDS/VDOTZ,XZERC(9),FZERO(9),NNZERO
ISN 0007           COMMON/BLK 11/DB1,DB2,DG1,DG2                                10700070
ISN 0008            COMMON/BLK77/X(15)                                          10700080
ISN 0009           COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)             10700090
ISN 0010           COMMON/ABORT/ DET, VSR ,VCL                                  10700100
ISN 0011           SD = SQRT(DET)                                               10700110
ISN 0012           JSUE = 0                                                     10700120
ISN 0013           IFLAG = 0                                                    10700130
ISN 0014           KAM = 0                                                      10700140
ISN 0015           XK12 =100                                                    10700150
ISN 0016           NN = 0                                                       10700160
ISN 0017           NNN= 1                                                       10700170
             C                                                                  10700180
ISN 0018           KFAIL=0                                                      10700190
ISN 0019       100 DO 1050 I=1,9                                                10700200
ISN 0020           G(I)= VMIN/B(I)                                              10700210
ISN 0021      1050 GG(I)= SQRT(G(I))                                            10700220
ISN 0022       103 IF(NN .LT. 500*NNN)GC TC 800                                 10700230
ISN 0024       705 FORMAT(4H NN=,I6)                                            10700240
ISN 0025           NNN=NNN + 1                                                  10700250
ISN 0026       800 NN = NN + 1                                                  10700260
ISN 0027           IF(NN .GT. 3.0E+05) GO TO 311                                10700270
ISN 0029           DO 101 I=1,9                                                 10700280
ISN 0030           CALL BOXNO(R1,R2)                                            10700290
ISN 0031           XX(I) = R1                                                   10700300
ISN 0032           IF(NN .LT. 1000)XX(I)= GG(I)*XX(I)/6.                        10700310
ISN 0034           IF(NN .GE. 1000)XX(I)= GG(I)*XX(I)/3.                        10700320
ISN 0036       101 CONTINUE                                                     10700330
             C     XX(I) ARE THE EIGENVECTOR COORDINATES                        10700340
             C     NOW TRANSFORM TO X(I) COORDINATES                            10700350
ISN 0037           DO 930 I=1,9                                                 10700360
ISN 0038       930 X(I)=0.0                                                     10700370
ISN 0039           DO 935 I=1,9                                                 10700380
ISN 0040           DO 935 J=1,9                                                 10700390
ISN 0041       935 X(I)= X(I) + C(I,J)*XX(J)                                    10700400
             C     GENERATE VL                                                  10700410
ISN 0042       259 VL=0.0                                                       10700420
ISN 0043           DO 260 I=1,9                                                 10700430
ISN 0044       260 PX(I)=0.0                                                    10700440
ISN 0045           DO 261 I=1,9                                                 10700450
ISN 0046           DO 261 J=1,9                                                 10700460
ISN 0047       261 PX(I)=PX(I) + PM(I,J)*X(J)                                   10700470
ISN 0048           DO 262 I=1,9                                                 10700480
ISN 0049       262 VL=VL + X(I)*PX(I)                                           10700490
ISN 0050           IF(VL .GE. VMIN)GO TO 103                                    10700500
ISN 0052           JSW=1                                                        10700510
ISN 0053           CALL AFX(JSW)                                                10700520
             C     *********************************************************** 10700530
ISN 0054           VDOT =0.0                                                    10700540
ISN 0055           DO 250 I=1,9                                                 10700550
```

```
ISN 0056          PF(I)=0.0                                                     10700560
ISN 0057      250 QX(I)=0.0                                                     10700570
ISN 0058          DO 251 I=1,9                                                  10700580
ISN 0059          DO 251 J=1,9                                                  10700590
ISN 0060          QX(I)= QX(I) + Q(I,J)*X(J)                                    10700600
ISN 0061      251 PF(I)=PF(I) + PM(I,J)*F(J)                                    10700610
ISN 0062          DO 252 I=1,9                                                  10700620
ISN 0063      252 VDOT = VDOT - X(I)* (QX(I)-2.0 * PF(I))                       10700630
ISN 0064          IF(IFLAG.EQ.1)GO TC 520                                       10700640
ISN 0066          IF(VDOT .LT. 0)GO TC 103                                      10700650
ISN 0068          IF(VMIN .LT. VL)GO TC 103                                     10700660
ISN 0070          IFLAG = 1                                                     10700670
ISN 0071      500 DO 510 I=1,9                                                  10700680
ISN 0072      510 DEL(I)= 0.5*X(I)                                              10700690
             C                                    )9,1=I,)I(X()055,6(ETI 10700700
ISN 0073      512 DO 515I=1,9                                                   10700710
ISN 0074      515 X(I)= X(I)- DEL(I)                                            10700720
ISN 0075          GO TO 535                                                     10700730
ISN 0076      520 DO 530 I=1,9                                                  10700740
ISN 0077      530 DEL(I)= DEL(I)*.5                                             10700750
ISN 0078          IF(VDOT .GT.0.0)GO TO 512                                     10700760
ISN 0080          DO 540 I=1,9                                                  10700770
ISN 0081      540 X(I)= X(I) + DEL(I)                                           10700780
ISN 0082      535 KAM= KAM + 1                                                  10700790
             C                                       LV,TODV)065,6(ETI 10700800
ISN 0083      560 FORMAT(7H VDOT,V/(1X,2E14.7))                                 10700810
             C                                    )9,1=I,)I(X()055,6(ETI 10700820
ISN 0084      550 FORMAT(3H X=/(1X,9E13.4))                                     10700830
ISN 0085          IF(KAM .LT.15) GO TO 259                                      10700840
ISN 0087          VMIN = VL                                                     10700850
ISN 0088          DO 1100 I=1,9                                                 10700860
ISN 0089          FZERO(I)=F(I)                                                 10700870
ISN 0090     1100 XZERO(I)=X(I)                                                 10700880
ISN 0091          NNZERO= NN                                                    10700890
ISN 0092          VDOTZ=VDOT                                                    10700900
             C                                             NN)507,6(ETI 10700910
             C                  )9,1=I,)I(F(,2GD,1GD,2BD,1BD)213,6(ETI 10700920
ISN 0093      312 FORMAT(6H DB,DG,4E15.7/6H F(I)=,9E12.4 )                      10700930
ISN 0094          IFLAG = 0                                                     10700940
ISN 0095          KAM =0                                                        10700950
ISN 0096          KFAIL=1                                                       10700960
ISN 0097          IF(VL.LT.1.0E-12) GC TO 66                                    10700970
ISN 0099          VOL= SD/((SQRT(VL))**9)                                       10700980
ISN 0100          IF(VOL.GT.VSR) GO TC 69                                       10700990
ISN 0102          GO TO 100                                                     10701000
ISN 0103       66 CONTINUE
             C  66 WRITE(6,2220) VL,NN                                          10701010
ISN 0104     2220 FORMAT(' VL = ',E14.7,5X,'TOO SMALL',5X,' NN=',I5 )           10701020
ISN 0105          VOL = 1.0E+50                                                 10701030
             C                                                                  10701040
ISN 0106       69 JSUF = 1                                                      10701050
             C                                             NN)507,6(ETI 10701060
ISN 0107      311 IF(KFAIL.EQ.0)GO TC 315                                       10701070
ISN 0109          WRITE(6,1202)NNZERC
ISN 0110     1202 FCRMAT( ' NNZERO =',I6 )
ISN 0111          GO TO 316                                                     10701120
```

```
ISN 0112        315 WRITE(6,2503)                                          10701130
ISN 0113       2503 FORMAT( ' NO LINEAR SEARCH '/ )                        10701140
ISN 0114        316 WRITE(6,705)NN                                         10701150
ISN 0115            RETURN                                                 10701160
ISN 0116            END                                                    10701170
```

```
            COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
   ISN 0002              SUBROUTINE QGEN(THETA,PHIV,XLAM)                                    10900010
   ISN 0003              IMPLICIT REAL*8 (A-H,O-Z)                                           10900020
                   C                                                                         10900030
                   C     GENERATION OF POSITIVE DEFINITE Q MATRIX                            10900040
   ISN 0004              DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP                   10900050
   ISN 0005              COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),E(9),DF(9,9)                    10900060
   ISN 0006              COMMON/BLK 70/ AAR(9,9),BM(9,9),PA(9,9),ATP(9,9)                    10900070
   ISN 0007              N=9                                                                 10900080
                   C                                                                         10900090
   ISN 0008              DIMENSION THETA(28),PHIV(8),XLAM(9),ZTHETA(28),TTHETA(28),          10900100
                        1          BA(20,20),SS(20,20,20),CC(20,20),Z(20,20,20),             10900110
                        2      SM(9,9),G(9,9),GQ(9,9)                                        10900120
                   C          )9,1=K,)K(MALX(,)8,1=J,)J(VIHP(,)82,1=I,)I(ATEHT()3601,5(CA   10900130
   ISN 0009         1063 FORMAT(6E12.4)                                                      10900140
   ISN 0010              EXTERNAL DMOD
   ISN 0011              PI = 3.1415926                                                      10900150
   ISN 0012              PI2 = PI/2.                                                         10900160
                   C     WRITE(6,3) THETA,PHIV,XLAM                                          10900170
   ISN 0013            3 FORMAT(23H DATA-THETA,PHIV,XLAM     /(1X,9E14.7))                   10900180
   ISN 0014              NN=(N-1)*(N-2)/2                                                    10900190
   ISN 0015              DO 6 I=1,NN                                                         10900200
   ISN 0016              BAD=THETA(I)                                                        10900210
   ISN 0017            6 TTHETA(I) = DMOD(BAD,PI2)                                           10900220
                   C     WE HAVE NOW INDEXED THETA.                                          10900230
                   C     NOW WANT CONTINUED PRODUCT OF SS(I,J,L) FOR L=K+1,N .               10900240
                   C     FOR EACH K=1,N-1 OBTAIN Z(K,I,J).                                   10900250
   ISN 0018               NNI = N-1                                                          10900260
   ISN 0019           69 DO 20 K=1,NNI                                                       10900270
                   C                                                                         10900280
   ISN 0020              DO 8 I=1,N                                                          10900290
   ISN 0021              DO 8 J=1,N                                                          10900300
   ISN 0022            8 BA(I,J)=0.0                                                         10900310
   ISN 0023              DO 99 I=1,N                                                         10900320
   ISN 0024           99 BA(I,I)=1.0                                                         10900330
                   C                                                                         10900340
   ISN 0025              KK=K+1                                                              10900350
   ISN 0026              DO 10 L=KK,N                                                        10900360
                   C                                                                         10900370
   ISN 0027              DO 15 I=1,N                                                         10900380
   ISN 0028              DO 15 J=1,N                                                         10900390
   ISN 0029           15 SS(I,J,L)=0.0                                                       10900400
   ISN 0030              DO 98 I=1,N                                                         10900410
   ISN 0031           98 SS(I,I,L)=1.0                                                       10900420
                   C     WE DEVELOP SS(I,J,L) AS FUNCTION THETA(L,K,N) FOR L L.T. N          10900430
                   C     AND SS(I,J,L) FUNCTION OF PHIV(K) FOR L=N                           10900440
   ISN 0032              IF(L-N)25,23,23                                                     10900450
   ISN 0033           25 M=((2*N -K-2)*(K-1)/2)+N-L                                          10900460
   ISN 0034              SS(K,K,L)=COS(TTHETA(M))                                            10900470
   ISN 0035              SS(L,L,L)=COS(TTHETA(M))                                            10900480
   ISN 0036              SS(K,L,L)=-SIN(TTHETA(M))                                           10900490
   ISN 0037              SS(L,K,L)=SIN(TTHETA(M))                                            10900500
   ISN 0038              GO TO 35                                                            10900510
   ISN 0039           23 SS(K,K,L)=COS(PHIV(K))                                              10900520
   ISN 0040              SS(L,L,L)=COS(PHIV(K))                                              10900530
   ISN 0041              SS(K,L,L)=-SIN(PHIV(K))                                             10900540
```

```
ISN 0042          SS(L,K,L)=SIN(PHIV(K))
            C
ISN 0043       35 DO 70 I=1,N
ISN 0044          DO 70 J=1,N
ISN 0045       70 CC(I,J)=0.0
            C
ISN 0046          DO 50 M=1,N
ISN 0047          DO 50 J=1,N
ISN 0048          DO 50 I=1,N
ISN 0049       50 CC(M,J)=BA(M,I)*SS(I,J,L) +CC(M,J)
ISN 0050          DO110 I=1,N
ISN 0051          DO110 J=1,N
ISN 0052      110 BA(I,J)=CC(I,J)
ISN 0053       10 CONTINUE
ISN 0054          DO 20 I=1,N
ISN 0055          DO 20 J=1,N
ISN 0056       20 Z(K,I,J)=BA(I,J)
            C
ISN 0057          DO 7 I=1,N
ISN 0058          DO 7 J=1,N
ISN 0059        7 BM(I,J)=0.0
ISN 0060          DO 16 I=1,N
ISN 0061       16 BM(I,I)=1.0
            C
ISN 0062          DO 40 K=1,NNI
            C
ISN 0063          DO 75 I=1,N
ISN 0064          DO 75 J=1,N
ISN 0065       75 SM(I,J)=0.0
            C
ISN 0066          DO 55 M=1,N
ISN 0067          DO 55 J=1,N
ISN 0068          DO 55 I=1,N
ISN 0069       55 SM(M,J)=Z(K,M,I)*BM(I,J)+SM(M,J)
ISN 0070          DO 40 I=1,N
ISN 0071          DO 40 J=1,N
ISN 0072       40 BM(I,J)=SM(I,J)
            C
            C     BM(I,J) IS CONTINUED PRODUCT OF Z(K,I,J) FROM K=1 TO N-1
            C
ISN 0073          IF(PP)41,41,19
ISN 0074       19 CONTINUE
ISN 0075       18 FORMAT(8H BM(I,J)/(6E15.7))
ISN 0076       41 DO 78 I=1,N
ISN 0077          DO 78 J=1,N
ISN 0078       78 AAR(I,J)=BM(J,I)
            C
            C     AAR(I,J) IS TRANSPOSE BM(I,J)
            C
ISN 0079          DO 82 I=1,N
ISN 0080          DO 82 J=1,N
ISN 0081       82 G(I,J)=0.0
ISN 0082          DO 85 I=1,N
ISN 0083       85 G(I,I)=XLAM(I)
            C     G(I,J) IS THE LAMDA MATRIX
            C
```

```
ISN 0084          DO 86 I=1,N                                  10901110
ISN 0085          DO 86 J=1,N                                  10901120
ISN 0086       86 QQ(I,J)=0.0                                  10901130
ISN 0087          DO 88 I=1,N                                  10901140
ISN 0088          DO 88 J=1,N                                  10901150
ISN 0089          DO 88 M=1,N                                  10901160
ISN 0090       88 QQ(I,J)=G(I,M)*BM(M,J)+QQ(I,J)               10901170
            C                                                  10901180
            C     QQ(I,J)=LAMDA MATRIX *BM(I,J)                10901190
            C                                                  10901200
ISN 0091          DO 90 I=1,N                                  10901210
ISN 0092          DO 90 J=1,N                                  10901220
ISN 0093       90 Q(I,J)=0.0                                   10901230
ISN 0094          DO 95 I=1,N                                  10901240
ISN 0095          DO 95 J=1,N                                  10901250
ISN 0096          DO 95 M=1,N                                  10901260
ISN 0097       95 Q(I,J)=AAR(I,M)*QQ(M,J)+Q(I,J)               10901270
ISN 0098          RETURN                                       10901300
ISN 0099          END                                          10901310
```

```
            COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002              SUBROUTINE PEAIQ(KEEF)
ISN 0003              IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004              COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
                C
ISN 0005              N = 9
                C
                C
ISN 0006              DOUBLE PRECISION AAMOD,P,QV,A,E,AM,C,FM,ATP,FA,QP
ISN 0007              DIMENSION AAMOD(82,82),P(81),QV(81),E(9,9),AM(9,9)
                     1 ,ISTEP(82),QP(9,9)
ISN 0008              COMMON/BLK 70/ AAR(9,9),BM(9,9),PA(9,9),ATP(9,9)
                C
                C     N =  DIMENSION OF A - MATRIX (INPUT ON CARD NO.2 )
                C     A =  INPUT MATRIX
                C
                C
                C
                C
                C     INITIALIZE  PM , QP, ATP , PA , AM , E
                C
                C
                C
ISN 0009              NN = N*N
ISN 0010              DO 37 I=1,N
ISN 0011              DO 37 J=1,N
ISN 0012              PM(I,J) = 0.0
ISN 0013              QP(I,J) = 0.0
ISN 0014              ATP(I,J)= 0.0
ISN 0015              PA(I,J) = 0.0
ISN 0016              AM(I,J) = 0.0
ISN 0017           37 E(I,J)  = 0.0
                C                                           )N,1=I,)N,1=J,)J,I(A(()3692,6(ETI
ISN 0018         2963 FORMAT(1H /(1X,9E14.7) )
                C
                C
                C     MAKING Q PERFECTLY SYMMETRIC
ISN 0019               NK = 2
ISN 0020           42 DO 76 JJ = 1,8
ISN 0021               Q(NK,JJ) = Q(JJ,NK)
ISN 0022              NOOK = NK-1
ISN 0023               IF(NOOK  .EQ.JJ) GO TO 77
ISN 0025           76  CONTINUE
ISN 0026           77  IF (NK.EQ.9) GO TO 87
ISN 0028               NK = NK + 1
ISN 0029               GO TO 42
ISN 0030           87  CONTINUE
                C
                C     SETTING Q-MATRIX TO Q-VECTOR
                C
ISN 0031              IA=1
ISN 0032              DO 62 I=1,N
ISN 0033              DO 62 J=1,N
ISN 0034              QV(IA) = Q(I,J)
ISN 0035           62 IA=IA+1
                C     WRITE(6,206)
```

```
ISN 0036      206 FORMAT(1H /  1X, 12H  Q-VECTOR    // )
            C     WRITE(6,2963)(QV(IP),IP=1,NN)
            C
            C
            C
            C     THIS SECTION CALCULATES THE AAMOD MATRIX
            C     WHICH IS GIVEN BY
            C           XXXX                                 XXXX
            C           X                                       X
            C           X AT+ A11.I     A21.I       A31.I       X
            C           X                                       X
            C           X  A12.I       AT+ A22.I    A32.I       X
            C           X                                       X
            C           X  A13.I        A23.I     AT+ A33.I     X
            C           X                                       X
            C           XXXX                                 XXXX
            C
            C
            C     INITIALIZATION OF AAMOD
            C
            C
ISN 0037          IF(KEEP.GT.0) GO TO 66
ISN 0039          DO 10  L = 1,NN
ISN 0040          DO 10  LA= 1,NN
ISN 0041       10 AAMOD(L,LA) = 0.0
            C
            C     DEFINE UNIT MATRIX OF ORDER  N ,CALL IT E
            C
ISN 0042          DO 30 MAA = 1,N
ISN 0043       30 E(MAA,MAA) = 1.0
            C
            C
            C
            C
            C     THIS SECTION CALCULATES THE SECTION OF THE  AMOD
            C     MATRIX WHICH IS COMPRISED OF  A(J,I) * E
ISN 0044          IP =  -N
ISN 0045          DO 50  MR=1,N
ISN 0046          IP = IP + N
ISN 0047          IPP = -N
ISN 0048          DO 50  JR =1,N
ISN 0049          IPP = IPP +N
ISN 0050          DO 40  K= 1,N
ISN 0051          DO 40 KA= 1,N
ISN 0052          AM(K,KA) = A(JR,MR) * E(K,KA)
ISN 0053          KPIP = K+IP
ISN 0054          KAP  = KA+IPP
ISN 0055       40 AAMOD(KPIP,KAP)=AM(K,KA)
ISN 0056       50 CONTINUE
            C
            C
            C
            C
            C
            C     THIS SECTION ADDS THE A MATRIX TO THE DIAGONAL NXN
            C     ELEMENTS OF AAMOD
```

```
                C
                C
ISN 0057              DO 60   K = 1,N
ISN 0058              IP  = (K-1)* N
ISN 0059              DO 55  LT = 1,N
ISN 0060              DO 55  LM = 1,N
ISN 0061              ILT = IP+LT
ISN 0062              IPM = IP+LM
ISN 0063           55 AAMOD(ILT,IPM)= AAMOD(ILT,IPM)+ A(LM,LT)
ISN 0064           60 CONTINUE
                C
                C     THAT FINISHES THE CALCULATION OF AMOD ,NOW WE MUST
                C     PRINT IT OUT BECAUSE SREVNI WIPES  OUT AAMOD
                C
                C                                                   )002,6(ETI
ISN 0065          200 FORMAT(1H /    1X,16H AAMOD - MATRIX  // )
                C                               )NN,1=I,)NN,1=J,)J,I(DOMAA(()3692,6(ETI
                C     NOW FOR A-INVERSE
                C
ISN 0066              IDEM=NN+1
                C
ISN 0067              CALL MINVD(AAMOD,IDEM,NN,ISTEP,IERR)
                C
                C
                C     NOW AAMOD INVERSE HAS REPLACED AAMOD
                C
                C                                                   )192,6(ETI
ISN 0068          291 FORMAT(1H / 1X,22HAAMOD-INVERSE  MATRIX  , //)
                C                                                NN,1=I 371
                C                                     )NN,1=J,)J,I(DOMAA()3692,6(ETI
ISN 0069           66 DO 70  IB = 1,NN
ISN 0070              P(IB) =0.0
ISN 0071              DO 70  IC = 1,NN
ISN 0072           70 P(IB) =  P(IB) -AAMOD(IB,IC) * QV(IC)
                C     WRITE(6,205)
ISN 0073          205 FORMAT(1H / 1X,14H  P-MATRIX    ,//)
                C     WRITE(6,2963) (P(IR),IR=1,NN)
                C     SET P-VECTOR TO P-MATRIX TO GET Q-PRIME FROM-ATP-PA =QP
ISN 0074              K=1
ISN 0075              DO25 I=1,N
ISN 0076              DO25 J=1,N
ISN 0077              PM(I,J) = P(K)
ISN 0078           25 K=K+1
                C     CALCULATION OF ATP (A TRANSPOSE P )
                C     CALCULATION OF PA  (P-MATRIX X A )
ISN 0079              DO 26 I=1,N
ISN 0080              DO 26 J=1,N
ISN 0081              DO 26 K=1,N
ISN 0082              ATP(I,J) = ATP(I,J) + A(K,I) *PM(K,J)
ISN 0083           26 PA(I,J)  = PA(I,J)  +PM(I,K) * A(K,J)
ISN 0084              DO 27 I=1,N
ISN 0085              DO 27 J=1,N
ISN 0086           27 QP(I,J) = -ATP(I,J) - PA(I,J)
                C     WRITE(6,9765)
ISN 0087         9765 FORMAT(1H / 1X,36H  Q FROM PUTTING P INTO -ATP-PA = Q , //)
                C     DO941 I=1,N
```

```
          C 941 WRITE(6,2963) (QP(I,J),J=1,N)                    10401680
ISN 0088        RETURN                                           10401690
ISN 0089        END                                              10401700
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002           SUBROUTINE   CLOCK                                               10200010
ISN 0003           COMMON/BLK68/ T                                                  10200020
ISN 0004           DATA        I/1/                                                 10200030
ISN 0005           COMMON/GEORGE/INIT
ISN 0006           CALL CLOCKS(NEW)
ISN 0007           T = FLOAT(NEW-INIT) * .01
ISN 0008           WRITE (6,1) T                                                    10200050
ISN 0009         1 FORMAT('0',90X,'CLOCK TIME',F16.2,5X,'SECONDS')                  10200060
ISN 0010           I = 0                                                            10200070
ISN 0011           RETURN                                                           10200080
ISN 0012           END                                                              10200090
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE DEIGN(PM,B,C)                                      10300010
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)                                     10300020
         C        THIS SUBROUTINE FINDS THE EIGEN VALUES AND VECTORS OF PM      10300030
ISN 0004          REAL*4  AA,B,C                                                10300040
ISN 0005          DIMENSION AA(60),B(9),QV(9),U(9),V(9),C(9,9),W(9),PM(9,9)
ISN 0006          DOUBLE PRECISION P(9)                                         10300060
ISN 0007          IK=1                                                          10300070
ISN 0008          DO 1000 K= 1,9                                                10300080
ISN 0009          DO 1000 I= K,9                                                10300090
ISN 0010          AA(IK) = PM(I,K)                                              10300100
ISN 0011     1000 IK = IK + 1                                                   10300110
ISN 0012          N = 9                                                         10300120
ISN 0013          M = 45                                                        10300130
ISN 0014          LEAD = 1                                                      10300140
ISN 0015          CALL SYMBIG(AA,N,LEAD,N,M,B,P,QV,U,V,MISS)                    10300150
ISN 0016          IF(MISS)1010,1020,1010                                        10300160
ISN 0017     1010 WRITE(6,1040)                                                 10300170
ISN 0018     1040 FORMAT(17H ERROR IN EIGSYM )                                  10300180
ISN 0019          GO TO 60                                                      10300190
ISN 0020     1020 CONTINUE
         C        FIND EIGENVECTORS OF PM(I,J)                                  10300220
ISN 0021          LOW = 1                                                       10300230
ISN 0022          KOUNT = 9                                                     10300240
ISN 0023          MID = 9                                                       10300250
ISN 0024          CALL SECURE(C,LOW,KOUNT,MID,W)                                10300260
         C                                                                      10300290
ISN 0025       60 RETURN                                                        10300300
ISN 0026          END                                                           10300310
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002           FUNCTION IMEQD(MID,M,N,A,Y,D,SCALE)                            10500010
             C THIS FORTRAN 4 PROGRAM SOLVES AX = Y BY TRIANGULAR DECOMPOSITION.  10500020
             C THE ARGUMENTS HAVE THE SAME MEANING AS THOSE OF XSIMEQ.            10500030
ISN 0003           REAL*8 DOTPR                                                   10500040
ISN 0004           DOUBLE PRECISION SUM,A,Y,SCALE,D                               10500050
ISN 0005           DIMENSION A(MID,1),Y(MID,1),SCALE(1)                           10500060
ISN 0006           COMMON /INFO/ SUM,NUMBER,INCR,INCC                             10500070
ISN 0007           INTEGER SPILL                                                  10500080
             C SET OVERFLOW INDICATOR.                                            10500090
             C  TAMPER OVERRIDES STANDARD HANDLING OF SPILL INTERUPTIONS. ITS ARGU- 10500110
             C    MENT IS SET TO ZERO AND THEREAFTER THE VALUES 0,1,2,3 INDICATE NO 10500120
             C    SPILL, UNDERFLOW ONLY, OVERFLOW ONLY, AND BOTH, RESPECTIVELY.   10500130
ISN 0008           INCR = MID                                                     10500140
ISN 0009           INCC = 1                                                       10500150
ISN 0010           DO 120  I = 1,M                                                10500160
ISN 0011           X = 0.                                                         10500170
ISN 0012           DO 100  J = 1,M                                                10500180
ISN 0013           GETZ = ABS(A(I,J))                                             10500190
ISN 0014       100 X = AMAX1(X,GETZ)                                              10500200
ISN 0015           IF (X) 105,490,105                                             10500210
ISN 0016       105 X = POW16(X)                                                   10500220
             C  POW16(X) IS THE POWER OF 16 NEXT LARGER THAN ABS(X)               10500230
ISN 0017           D = D * X                                                      10500240
ISN 0018           X = 1./ X                                                      10500250
ISN 0019           DO 110  J = 1,M                                                10500260
ISN 0020       110 A(I,J) = A(I,J) * X                                            10500270
ISN 0021           DO 120  J = 1,N                                                10500280
ISN 0022       120 Y(I,J) = Y(I,J) * X                                            10500290
ISN 0023           DO 140  J = 1,M                                                10500300
ISN 0024           X = 0.                                                         10500310
ISN 0025           DO 130  I = 1,M                                                10500320
ISN 0026           GOTZ = ABS(A(I,J))                                             10500330
ISN 0027       130 X = AMAX1(X,GOTZ)                                              10500340
ISN 0028           IF (X) 135,490,135                                             10500350
ISN 0029       135 X = POW16(X)                                                   10500360
ISN 0030           D = D * X                                                      10500370
ISN 0031           SCALE(J) = X                                                   10500380
ISN 0032           X = 1./ X                                                      10500390
ISN 0033           DO 140  I = 1,M                                                10500400
ISN 0034       140 A(I,J) = A(I,J) * X                                            10500410
             C MAJOR LOOP. TRIANGULAR DECOMPOSITION WITH D.P. ACCUM OF INNER PRODUCTS 10500420
ISN 0035           DO 310  K = 1,M                                                10500430
ISN 0036           K1 = K - 1                                                     10500440
ISN 0037       150 NUMBER = K1                                                    10500450
ISN 0038           X = 0.                                                         10500460
ISN 0039           L = K                                                          10500470
ISN 0040           DO 180  I = K,M                                                10500480
ISN 0041           SUM = A(I,K)                                                   10500490
ISN 0042           A(I,K) = DOTPR(A(I,1),A(1,K))                                  10500500
             C     + X(NUMBER)*Y(NUMBER) WHERE X AND Y HAVE THE STORAGE INCREMENTS 10500510
             C  DOTPR(X,Y) GIVES THE (D.P. ACCUMULATED) VALUE SUM + X(1)*Y(1) +... 10500520
             C     INCR AND INCC. DOTPR USES COMMON AREA INFO                     10500530
ISN 0043       165 IF (X - ABS(SUM)) 170,180,180                                  10500540
ISN 0044       170 X = ABS(SUM)                                                   10500550
ISN 0045           L=I                                                            10500560
```

```
ISN 0046      180 CONTINUE                                                      10500570
ISN 0047          IF (L - K) 490,220,190                                        10500580
            C ROW INTERCHANGES TO INSURE LARGE PIVOTS                           10500590
ISN 0048      190 D = - D                                                       10500600
ISN 0049          DO 200 J=1,M                                                  10500610
ISN 0050          X = A(L,J)                                                    10500620
ISN 0051          A(L,J) = A(K,J)                                               10500630
ISN 0052      200 A(K,J) = X                                                    10500640
ISN 0053          DO 210 J=1,N                                                  10500650
ISN 0054          X = Y(L,J)                                                    10500660
ISN 0055          Y(L,J) = Y(K,J)                                               10500670
ISN 0056      210 Y(K,J) = X                                                    10500680
ISN 0057      220 X = -A(K,K)                                                   10500690
ISN 0058          IF (M-K) 490,275,230                                          10500700
ISN 0059      230 KD = K + 1                                                    10500710
ISN 0060          DO 240  I = KD,M                                              10500720
ISN 0061      240 A(I,K) = A(I,K) / X                                           10500730
ISN 0062      250 DO 270  L = KD,M                                              10500740
ISN 0063          SUM = A(K,L)                                                  10500750
ISN 0064      270 A(K,L) = DOTPR(A(K,1),A(1,L))                                 10500760
ISN 0065      275 DO 290 L=1,N                                                  10500770
ISN 0066          SUM = Y(K,L)                                                  10500780
ISN 0067      290 Y(K,L) = DOTPR(A(K,1),Y(1,L))                                 10500790
            C UNDULY SMALL PIVOT INDICATES A IS SINGULAR                        10500800
ISN 0068      300 IF (ABS(X) - 2.384186E-7) 490,490,310                         10500810
ISN 0069      310 CONTINUE                                                      10500820
            C BACK SOLUTION                                                     10500830
ISN 0070          I = M                                                         10500840
ISN 0071          DO 345 K=1,M                                                  10500850
ISN 0072          I1 = I + 1                                                    10500860
ISN 0073          NUMBER = M - I                                                10500870
ISN 0074          DO 340  L = 1,N                                               10500880
ISN 0075          SUM = -Y(I,L)                                                 10500890
ISN 0076      340 Y(I,L) = -DOTPR(A(I,I+1),Y(I+1,L)) / A(I,I)                   10500900
ISN 0077      345 I=I-1                                                         10500910
ISN 0078          DO 350  I = 1,M                                               10500920
ISN 0079          X = 1./ SCALE(I)                                              10500930
ISN 0080          D = D * A(I,I)                                                10500940
ISN 0081          DO 350  J = 1,N                                               10500950
ISN 0082      350 A(I,J) = Y(I,J) * X                                           10500960
ISN 0083          IMEQD = 1
ISN 0084      480 CONTINUE
            C     STNDRD ZEROS THE ARG OF TAMPER AND RESTORES STANDARD SPILL ACTION. 10500990
ISN 0085          RETURN                                                        10501000
ISN 0086      490 D = 0.                                                        10501010
ISN 0087          IMEQD = 3                                                     10501020
ISN 0088          GO TO 480                                                     10501030
ISN 0089          END                                                           10501040
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE AFX(JSW)                                                       10800010
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)                                                 10800020
            C     ***************************************************************** 10800030
ISN 0004          COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI                   10800040
ISN 0005          COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM                   10800050
ISN 0006          COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C                                     10800060
ISN 0007          COMMON /BLK4/ HPHI,HTHT,HPSI                                              10800070
ISN 0008          COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)                          10800080
ISN 0009          COMMON/BLK 11/DB1,DB2,DG1,DG2                                             10800090
ISN 0010           COMMON/BLK77/X(15)                                                       10800100
ISN 0011          COMMON/BLK78/X1E,X4E,X7E                                                  10800110
            C                                                                               10800120
            C     EXACT MODEL STATE EQUATIONS                                               10800130
            C                                                                               10800140
ISN 0012          IF(JSW.GT.0)GO TO 312                                                     10800150
            C      ( TRACKERS 3-4    (AMES 1-2)  )                                          10800160
            C                                                                               10800170
            C     EQUATIONS ARE IN THE FORM    X-DOT = A  X  + F(X)                         10800180
            C                                                                               10800190
            C     WHERE X IS A  NINE COMPONENT COLUMN VECTOR   AS  IS  F(X)                 10800200
            C                                                                               10800210
            C     AND  A  IS  9X9  MATRIX                                                   10800220
            C                                                                               10800230
            C     X-VECTOR  IS  ( PHI , VPHI , WPHI , THT , VTHT , WTHT , PSI ,             10800240
            C                                                                               10800250
            C          VPSI ,   WPSI )                                                      10800260
            C                                                                               10800270
            C     ***************************************************************** 10800280
            C                                                                               10800290
ISN 0014          WRITE(6,1070)PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI,TM,T1,T2,          10800300
                 1XKM,XKC,A11,A13,D12,GAM1C,GAM2C,BET1C,BET2C,AITREN,HTHT,HPHI,HPSI         10800310
ISN 0015     1070 FORMAT(1H /10X,14H INITIAL STATE / 9E13.6 / 10X,13H INPUT CONSTS /        10800320
                 110X, 29H TM,T1,T2,XKM,XKC,A11,A13,D12 / 8E14.7 / 10X, 30H GAM1C,GA        10800330
                 2M2C,BET1C,BET2C,(RAD) / 4E14.7 / 10X,10H INERTIA = E14.7 ,/10X,           10800340
                 317H HTHT,HPHI,HPSI = 3E14.7//)                                            10800350
            C     ***************************************************************** 10800360
ISN 0016          PI = 3.1415926                                                            10800370
ISN 0017          DTR  = PI/180.0                                                           10800380
ISN 0018          RTD  = 180.0/PI                                                           10800390
            C     ***************************************************************** 10800400
            C                                                                               10800410
ISN 0019          SB2C  = SIN(BET2C)                                                        10800420
ISN 0020          CB2C  = COS(BET2C)                                                        10800430
ISN 0021          TB2C = SB2C/CB2C                                                          10800440
ISN 0022          SG1C  = SIN(GAM1C)                                                        10800450
ISN 0023          CG1C  = COS(GAM1C)                                                        10800460
ISN 0024          SG2C  = SIN(GAM2C)                                                        10800470
ISN 0025          CG2C  = COS(GAM2C)                                                        10800480
ISN 0026          SB1C  = SIN(BET1C)                                                        10800490
ISN 0027          CB1C  = COS(BET1C)                                                        10800500
ISN 0028          TB1C  = SB1C/CB1C                                                         10800510
ISN 0029           SGAM1C = SG1C                                                            10800520
ISN 0030           SGAM2C = SG2C                                                            10800530
ISN 0031           CGAM1C = CG1C                                                            10800540
ISN 0032           CGAM2C = CG2C                                                            10800550
```

```
              C
              C      ****************************************************************
ISN 0033             DIF = GAM1C - GAM2C
ISN 0034             SDIF = SIN(DIF)
              C
              C      ****************************************************************
              C
              C      CALCULATION OF THE  A-MATRIX
              C
ISN 0035             DO 10 I=1,9
ISN 0036             DO 10 J=1,9
ISN 0037          10 A(I,J) = 0.0
              C
ISN 0038             A(1,2) = -XKM*XKC/AITREN
ISN 0039             A(7,8)=A(1,2)
ISN 0040             A(2,1) =  (T1+T2)/(TM*T2)
ISN 0041             A(2,2)=-1.0/TM
ISN 0042             A(5,5)=A(2,2)
ISN 0043             A(8,8)=A(5,5)
ISN 0044             A(3,3)=-1.0/T2
ISN 0045             A(6,6)=A(3,3)
ISN 0046             A(9,9)=-1/T2
ISN 0047             A(2,3) =  T1/(TM*T2)
ISN 0048             A(5,6)=A(2,3)
ISN 0049             A(8,9)=A(2,3)
ISN 0050             A(2,4) = A(2,1) *(-TE1C*CG1C)
ISN 0051             A(2,7)=A(2,1)*TB1C*SG1C
ISN 0052             A(3,1) = -1.0/T2
ISN 0053             A(3,4)=A(3,1)*A(2,4)/A(2,1)
ISN 0054             A(3,7)=A(3,1)*A(2,7)/A(2,1)
ISN 0055             A(4,5)=A(1,2)
ISN 0056             A(5,4) = A(2,1) * D12 * SDIF
ISN 0057             A(6,4)=A(3,1)*A(5,4)/A(2,1)
ISN 0058             A(8,7)=A(5,4)
ISN 0059             A(9,7)=A(6,4)
              C
ISN 0060             WRITE(6,5000) ( (A(I,J),J=1,9),I=1,9)
ISN 0061        5000 FORMAT(1H / 1X,11H  A-MATRIX  /(1X, 9E13.6 ))
              C
              C      ****************************************************************
              C
              C      CALCULATION  OF DB1,DB2,DG1,DG2
              C
              C      NEEDS ANGLES FROM STATE VECTOR AND COMMAND ANGLES AS INPUT
              C
              C
              C
ISN 0062         312 PHI= X(1) + X1E
ISN 0063             THT= X(4) + X4E
ISN 0064             PSI= X(7) + X7E
ISN 0065             SPHI = SIN(PHI)
ISN 0066             CPHI = COS(PHI)
ISN 0067             STHT = SIN(THT)
ISN 0068             CTHT = COS(THT)
ISN 0069             SPSI = SIN(PSI)
ISN 0070             CPSI = COS(PSI)
```

```
ISN 0071          TTHT = STHT/CTHT                                                  10801120
ISN 0072          XNORW = XKC*T1/T2                                                 10801130
ISN 0073          PUP = 1.0/(XKC*TM)                                                10801140
            C                                                                       10801150
ISN 0074          GR1= CPSI*CTHT*SB1C                                               10801160
ISN 0075          GR2= SPSI*CTHT*CG1C*CB1C                                          10801170
ISN 0076          GR3= STHT*SG1C*CB1C                                               10801180
ISN 0077          GR4= GR1 + GR2 + GR3                                              10801190
ISN 0078          DB1= ARSIN(GR4) - BET1C                                           10801200
            C                                                                       10801210
ISN 0079          DG1 =                                      ATAN(    (-SB1C*(SPSI*  10801220
                 1 SPHI + CPSI*STHT*CPHI) +CG1C*CB1C *(CPSI*SPHI -SPSI*STHT*CPHI)   10801230
                 2 +SG1C*CB1C*CTHT*CPHI ) /  (SB1C *(-SPSI*CPHI + CPSI*STHT*SPHI)   10801240
                 3 +CG1C*CB1C*(CPSI*CPHI + SPSI*STHT*SPHI) -CB1C*SG1C*CTHT*SPHI) )  10801250
                 4                                         - GAM1C                  10801260
            C                                                                       10801270
ISN 0080          GL1= CPSI*CTHT*SB2C-SPSI*CTHT*CG2C*CB2C-STHT*SG2C*CB2C            10801280
ISN 0081          DB2= ARSIN(GL1) - BET2C                                           10801290
            C                                                                       10801300
ISN 0082          DG2 = -GAM2C + ATAN(  (SB2C*(SPSI*SPHI +CPSI*STHT*CPHI) +CG2C*    10801310
                 1 CB2C*(CPSI*SPHI-SPSI*STHT*CPHI) +SG2C*CB2C*CTHT*CPHI )/ (SB2C*   10801320
                 2 (SPSI*CPHI-CPSI*STHT*SPHI)+CG2C*CB2C*(CPSI*CPHI+SPSI*STHT*SPHI)  10801330
                 3 -CTHT*SPHI*SG2C*CB2C ) )                                         10801340
ISN 0083          IF(BET1C.EQ. 0.0) GO TO 850                                       10801350
ISN 0085          APE1= ABS(DB1/BET1C)                                              10801360
ISN 0086          IF(APE1 .LT. 1.D-10)DB1= SG1C*THT + CG1C*PSI                      10801370
ISN 0088      850 IF(GAM1C .EQ. 0.0) GO TO 851                                      10801380
ISN 0090          APE2= ABS(DG1/GAM1C)                                              10801390
ISN 0091          IF(APE2 .LT. 1.D-10)DG1= PHI - TB1C*CG1C*THT + TB1C*SG1C*PSI      10801400
ISN 0093      851 IF(BET2C .EQ. 0.0) GO TO 852                                      10801410
ISN 0095          APE3= ABS(DB2/BET2C)                                              10801420
ISN 0096          IF(APE3 .LT. 1.D-10)DB2= -SG2C*THT - CG2C*PSI                     10801430
ISN 0098      852 IF(GAM2C .EQ. 0.0) GO TO 853                                      10801440
ISN 0100          APE4= ABS(DG2/GAM2C)                                              10801450
ISN 0101          IF(APE4 .LT. 1.D-10)DG2= PHI + TB2C*CG2C*THT - TB2C*SG2C*PSI      10801460
            C                                                                       10801470
            C     ************************************************************** 10801480
            C                                                                       10801490
            C     CALCULATION OF NONLINEAR F(X)                                     10801500
            C                                                                       10801510
            C                                                                       10801520
ISN 0103      853 SUM3 = A(2,4)/A(2,1)                                              10801530
ISN 0104          SUM4 = A(2,7)/A(2,1)                                              10801540
ISN 0105          SUM5 =(COS(DG2+GAM2C))* DB1 + (COS(DG1 + GAM1C)) * DB2            10801550
ISN 0106          SUM6 =(SIN(DG2+GAM2C))* DB1 + (SIN(DG1 + GAM1C)) * DB2            10801560
ISN 0107          GAIN = XKC * (T1+T2)/T2                                           10801570
ISN 0108          F(1) = (TTHT * A(1,2))*(X(5)*SPHI + X(8)*CPHI)                    10801580
            C                                                                       10801590
ISN 0109          ARGF2= -(T1/T2)*AITREN*HPHI/XKM + GAIN*DG1 + XNORW*X(3)           10801600
ISN 0110          IF (ABS(ARGF2).LE.F2LIM) F2 = ARGF2                               10801610
ISN 0112          IF (ARGF2.GT.F2LIM) F2 = F2LIM                                    10801620
ISN 0114          IF (ARGF2.LT.-F2LIM) F2= -F2LIM                                   10801630
ISN 0116          F(2)= -(-F2 +GAIN*(X(1)+SUM3*X(4) +SUM4*X(7))+AITREN*HPHI         10801640
                 1   /XKM ) * PUP   - A(2,3)*X(3)                                   10801650
            C                                                                       10801660
ISN 0117          F(3)=A(3,1)*(DG1-(X(1)+SUM3*X(4)+ SUM4*X(7)))-HPHI/(T2*A(1,2))    10801670
```

```
ISN 0118          F(4) = -A(1,2)* (X(5)*(1.0 - CPHI) + SPHI *X(8) )
              C
ISN 0119          ARGF5= GAIN*SUM5*D12 -(T1/T2)*AITREN*HTHT/XKM + XNORW*X(6)
ISN 0120          IF(ABS(ARGF5).LE.F2LIM) F2 =ARGF5
ISN 0122          IF(ARGF5.GT.F2LIM) F2=F2LIM
ISN 0124          IF(ARGF5 .LT. -F2LIM)F2 = -F2LIM
ISN 0126          F(5) =-(-F2 +GAIN*D12*SDIF*X(4) +XNORW*X(6) +AITREN*HTHT/XKM)
                 1     * PUP
              C
ISN 0127          F(6) = A(3,1)*D12*(SUM5 - SDIF*X(4)) -HTHT/(T2*A(1,2))
              C
ISN 0128          F(7) = A(1,2) * (X(5)*SPHI +X(8)*(CPHI-CTHT))/CTHT
              C
ISN 0129          ARGF8= -GAIN*D12*SUM6-(T1/T2)*AITREN*HPSI/XKM + XNORW*X(9)
ISN 0130          IF(ABS(ARGF8).LE.F2LIM) F2 = ARGF8
ISN 0132          IF(ARGF8.GT.F2LIM) F2 = F2LIM
ISN 0134          IF(ARGF8.LT.-F2LIM) F2=-F2LIM
ISN 0136          F(8) =-(-F2+GAIN*D12*SDIF*X(7) + XNORW*X(9) +AITREN*HPSI/XKM)*PUP
              C
ISN 0137          F(9)= -A(3,1)*D12*(SUM6+SDIF*X(7))-HPSI/(T2*A(1,2))
              C
              C
              C
              C   ***************************************************************
              C
ISN 0138          IF(JSW.GT.0)GO TO 602
ISN 0140          WRITE(6,5001) (F(I),I=1,9)
ISN 0141     5001 FORMAT( 1H / 1X,12H F(X)-VECTOR / 1X,9E13.6)
ISN 0142      602 CONTINUE
              C
              C
              C   ***************************************************************
ISN 0143          RETURN
ISN 0144          END
```

```
LEVEL 2 FEB 67                                   OS/360  FORTRAN H                                  DATE  69.199/15.48.07

         COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

   ISN 0002             FUNCTION SCAPR(X,Y,SUM,L,IX,IY)                                    11000010
   ISN 0003            REAL*8 X(IX,1),Y(IY,1)                                              11000020
   ISN 0004            REAL*8 SUM,SCAPR                                                    11000030
   ISN 0005            IF (L .EQ. 0) GO TO 110                                             11000040
   ISN 0007            DO 100  J = 1,L                                                     11000050
   ISN 0008        100 SUM = SUM + X(1,J)*Y(1,J)                                           11000060
   ISN 0009        110 SCAPR = SUM                                                         11000070
   ISN 0010            RETURN                                                              11000080
   ISN 0011             END                                                                11000090
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE MINVD(A,IDIM,N,ISTEP,IERR)                       10600010
         C                  MATRIX INVERSION          DOUBLE PRECISION        10600020
ISN 0003          DOUBLE PRECISION A,ABSAI,ABSAL,TEMP,FAC,ABSA,DABS           10600030
ISN 0004          DIMENSION A(IDIM,1),ISTEP(1)                                10600040
ISN 0005          K=1                                                         10600050
ISN 0006          IERR=0                                                      10600060
ISN 0007          NP1=N+1                                                     10600070
ISN 0008       30 LL=1                                                        10600080
ISN 0009          DO 35 J=1,N                                                 10600090
ISN 0010       35 A(J,NP1)=A(J,1)                                             10600100
ISN 0011       40 I=1                                                         10600110
ISN 0012          L=2                                                         10600120
ISN 0013       45 ABSAI=DABS(A(I,1))                                          10600130
ISN 0014          ABSAL=DABS(A(L,1))                                          10600140
ISN 0015          IF(ABSAI-ABSAL)50,55,55                                     10600150
ISN 0016       50 I=L                                                         10600160
ISN 0017       55 IF(L-N)60,56,56                                             10600170
ISN 0018       56 IF(A(I,1))65,85,65                                          10600180
ISN 0019       60 L=L+1                                                       10600190
ISN 0020          GO TO 45                                                    10600200
ISN 0021       65 IF(K-1)70,90,70                                             10600210
ISN 0022       70 M=1                                                         10600220
ISN 0023       75 IF(I-ISTEP(M))80,84,80                                      10600230
ISN 0024       80 IF(M-K+1)81,82,82                                           10600240
ISN 0025       81 M=M+1                                                       10600250
ISN 0026          GO TO 75                                                    10600260
ISN 0027       82 DO 83 J=1,N                                                 10600270
ISN 0028       83 A(J,1)=A(J,NP1)                                             10600280
ISN 0029          GO TO 90                                                    10600290
ISN 0030       84 IF(LL-N)86,85,85                                            10600300
ISN 0031       85 IERR=1                                                      10600310
ISN 0032          GO TO 610                                                   10600320
ISN 0033       86 LL=LL+1                                                     10600330
ISN 0034          A(I,1)=0.D0                                                 10600340
ISN 0035          GO TO 40                                                    10600350
ISN 0036       90 ISTEP(K)=I                                                  10600360
ISN 0037          J=1                                                         10600370
ISN 0038      100 IF(J-I)110,120,110                                          10600380
ISN 0039      110 A(J,NP1)=0.D0                                               10600390
ISN 0040          GO TO 130                                                   10600400
ISN 0041      120 A(J,NP1)=1.D0                                               10600410
ISN 0042      130 IF(J-N)140,150,150                                          10600420
ISN 0043      140 J=J+1                                                       10600430
ISN 0044          GO TO 100                                                   10600440
ISN 0045      150 J=1                                                         10600450
ISN 0046          TEMP=A(I,1)                                                 10600460
ISN 0047      160 A(I,J)=A(I,J)/TEMP                                          10600470
ISN 0048          IF(J-NP1)170,180,180                                        10600480
ISN 0049      170 J=J+1                                                       10600490
ISN 0050          GO TO 160                                                   10600500
ISN 0051      180 J=1                                                         10600510
ISN 0052      190 IF(J-I)200,290,200                                          10600520
ISN 0053      200 IF(A(J,1)-1.D0)230,210,230                                  10600530
ISN 0054      210 DO 220 M=1,NP1                                              10600540
ISN 0055          A(J,M)=A(J,M)-A(I,M)                                        10600550
```

```
220 CONTINUE
    GO TO 290
230 IF(A(J,1)+1.D0)26C,24C,26C
240 DO 250 M=1,NP1
    A(J,M)=A(J,M)+A(I,M)
250 CONTINUE
    GO TO 290
260 IF(A(J,1))270,29C,27C
270 FAC=A(J,1)
    DO 280 M=1,NP1
280 A(J,M)=A(J,M)-A(I,M)*FAC
290 IF(J-N)300,340,340
300 J=J+1
    GO TO 190
340 DO 350 J=1,N
    DO 350 M=1,N
    MP1=M+1
350 A(J,M)=A(J,MP1)
    IF(K-N)360,390,390
360 K=K+1
    GO TO 30
390 DO 400 J=1,N
400 A(NP1,J)=ISTEP(J)
    M=1
410 I=ISTEP(M)
    IF(I-M)420,470,42C
420 DC 430 J=1,N
    TEMP=A(M,J)
    A(M,J)=A(I,J)
430 A(I,J)=TEMP
    J=M
440 IF(M-ISTEP(J))450,46C,450
450 J=J+1
    GO TO 440
460 ISTEP(J)=I
470 IF(M-N)480,490,49C
480 M=M+1
    GO TO 410
490 DO 500 J=1,N
500 ISTEP(J)=A(NP1,J)
530 M=1
540 I=ISTEP(M)
    IF(I-M)550,570,550
550 DO 560 J=1,N
    TEMP=A(J,I)
    A(J,I)=A(J,M)
560 A(J,M)=TEMP
    J=ISTEP(M)
    ISTEP(M)=ISTEP(J)
    ISTEP(J)=J
    GO TO 540
570 IF(M-N)580,610,61C
580 M=M+1
    GO TO 540
610 RETURN
    END
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

   ISN 0002          FUNCTION DOTPR(X,Y)                                           11100010
   ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)                                     11100020
   ISN 0004          REAL*8 X(1),Y(1)                                              11100030
   ISN 0005          COMMON /INFO/ SUM,NUMBER,INCR,INCC                            11100040
   ISN 0006          DOTPR  = SCAPR(X,Y,SUM,NUMBER,INCR,INCC)                      11100050
   ISN 0007          RETURN                                                        11100060
   ISN 0008          END                                                           11100070
```

LISTINGS FOR PICK-A-P


```
        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
  ISN 0002              IMPLICIT REAL*8 (A-H,O-Z)
                C       Q - OPTIMIZATION PROGRAM     8-20-68   INITIATE
  ISN 0003              DIMENSION THETA(28),PHIV(8),XLAM(9),RHV(9),SLUF(9)
  ISN 0004              DIMENSION  XLMX(9)
  ISN 0005              DIMENSION B(9),C(9,9)
  ISN 0006              DIMENSION PRM(9,9)
  ISN 0007              REAL*4  B,C,DUM,GUM
  ISN 0008              DIMENSION ALP(18)
  ISN 0009              DIMENSION DU(45),RX(45),NUTU(45),XLAMP(9),THETP(28),PHIVP(8)        III
  ISN 0010              DIMENSION EX(9),XINC(9)                                             III
  ISN 0011              DIMENSION DINC(9)
  ISN 0012              DIMENSION BUNCH(260C,6)
  ISN 0013              COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI
  ISN 0014              COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM
  ISN 0015              COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C
  ISN 0016              COMMON /BLK4/ HPHI,HTHT,HPSI
  ISN 0017              COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
  ISN 0018              COMMON/BLK 11/DB1,DB2,DG1,DG2
  ISN 0019               COMMON/BLK77/X(15)
  ISN 0020              COMMON/BLK78/X1E,X4E,X7E
  ISN 0021              COMMON/BLK68/ T
  ISN 0022              COMMON/ABORT/ DET, VSR ,VOL                                         NP
  ISN 0023              COMMON/BLK 70/ AAR(9,9),BM(9,9),PA(9,9),ATP(9,9)
  ISN 0024              CALL SYSTRR(1)
                C       THIS PROGRAM GENERATES P-MATRIX IN QGEN FROM 45-INPUT NUMBERS
  ISN 0025         1001 FORMAT(5E14.7)
  ISN 0026          102 READ(5,1101,END=1000) ALP
  ISN 0027         1101 FORMAT(18A4)
  ISN 0028              WRITE(6,191)(ALP(J),J=1,18)
  ISN 0029          191 FORMAT(1H / 10X,18A4)
  ISN 0030              READ(5,1104)NSKIP,LMIN,NUKEY,KIKIT,NXCUE,NSW,NSW1,NSW2
  ISN 0031         1104 FORMAT(18I4)
                C       NSKIP = 0 FOR NO P-MATRIX ELEMENT INPUT -- STARTS WITH UNIT MATRIX
                C       NSKIP = 1 INPUT 10 CARDS WITH XLAM,THETA,PHIV,PSR AND VSR
                C       IF VSR AND PSR ARE GIVEN  THEN  NUKEY=LMIN=1
                C       KIKIT = 0---YIELDS TRI-DIAGONAL Q                                   III
                C       KIKIT = 1---VARIES ALL LAMDA , THETA AND PHIV                       III
                C       NXCUE MUST BE AT LEAST 1 LESS THAN THE LARGE DIMENSION OF BUNCH
                C       NXCUE - NSW = SHRINKING PORTION = OF POINTS
                C       NSW = = OF. POINTS IN THE EXPANDING PORTION OF SEARCH
                C       NSW1 = = OF PTS BEFORE DIVISOR SHRINKS IN EXPANDING SEARCH
                C       NSW2 = = OF PTS BEFORE DIVISOR DOUBLES IN CONTRACTING SEARCH
  ISN 0032              VSR = 1.0 E+60
  ISN 0033              PSR = 2.0
  ISN 0034              N=9
  ISN 0035              NCT = 0
  ISN 0036              KEEP = 0
  ISN 0037          751 TM     = 75.8
  ISN 0038              T1     =  4.5
  ISN 0039              T2     =  0.5
  ISN 0040              XKM    =  1.0/13.0
  ISN 0041              XKC    =  2.665 E+05
  ISN 0042              A13    =  0.0
  ISN 0043              A11     = 0.0
  ISN 0044              AITREN= 1500.0
```

```
ISN 0045          F2LIM = 25.0
ISN 0046          READ(5,1001)GAM1C,GAM2C,BET1C,BET2C
ISN 0047          READ(5,1001)HPHI,HTHT,HPSI
ISN 0048          READ(5,1001)VMINI
ISN 0049          READ(5,712) DUM                                              X
ISN 0050      712 FORMAT(Z8)                                                   X
ISN 0051          IF(DUM)713,713,714                                           X
ISN 0052      714 CALL RDMIN(DUM)                                              X
ISN 0053      713 CONTINUE                                                     X
            C                                                 INIMV)18,6(ETI RW
ISN 0054       81 FORMAT ( 7H VMINT= E14.7)
ISN 0055          PI= 3.1415926
ISN 0056          PI2 = PI/2.
ISN 0057          DTR = PI/180.
ISN 0058          GAM1C = GAM1C * DTR
ISN 0059          GAM2C = GAM2C * DTR
ISN 0060          BET1C = BET1C * DTR
ISN 0061          BET2C = BET2C * DTR
ISN 0062          DIF = GAM1C - GAM2C
ISN 0063          D12 =  DIF * 2.0 / DABS(DIF)
ISN 0064          SDIF=SIN(DIF)
ISN 0065          DO 201 I=1,28
ISN 0066      201 THETA(I) = 0.
ISN 0067          DO 202 I=1,8
ISN 0068      202 PHIV(I) = 0.0
ISN 0069          DO 203 I=1,9
ISN 0070          XLMX(I) = 1.0
ISN 0071          X(I) = 1.0
ISN 0072      203 XLAM(I)= 1.0
ISN 0073          IF(NSKIP.EQ.0) GO TO 9299
ISN 0075          READ(5,1001) (XLAM(I),I=1,9),(THETA(J),J=1,28),(PHIV(K),K=1,8)
                 1  ,PSR,VSR
ISN 0076     9299 DO 444 I=1,N
ISN 0077          DO 444 J=1,N
ISN 0078          ATP(I,J)=0.0
ISN 0079      444 PA(I,J)=0.0
ISN 0080          NCUE = 1
ISN 0081          DO 6009 I=1,9                                              III
ISN 0082     6009 EX(I) = DLOG(XLAM(I))                                      III
ISN 0083          DO 420 I=1,9                                               III
ISN 0084      420 XLAMP(I) = XLAM(I)                                         III
ISN 0085          DO 430 I=1,28                                              III
ISN 0086      430 THETP(I) = THETA(I)                                        III
ISN 0087          DO 440 I=1,8                                               III
ISN 0088      440 PHIVP(I) = PHIV(I)                                         III
            C          )8,1=I,)I(PVIHP(,)82,1=I,)I(PTEHT(,)9,1=I,)I(PMALX()055,6(ETI RW
ISN 0089          SG1C = SIN(GAM1C)
ISN 0090          CG1C = COS(GAM1C)
ISN 0091          SG2C = SIN(GAM2C)
ISN 0092          CG2C = COS(GAM2C)
ISN 0093          TB1C = SIN(BET1C)/COS(BET1C)
ISN 0094          X15 = ( AITHEN)/(XKM*XKC)*(HPHI-HTHT*(A11*SG1C-A13*SG2C-CG1C*TB1C
                 1  )/(D12*SDIF) &HPSI*(A11*CG1C-A13*CG2C&SG1C*TB1C)/(D12*SDIF))
ISN 0095          X45 = ( AITHEN)/(XKM*XKC)*(HTHT/(D12*SDIF))
ISN 0096          X75 = (-AITHEN)/(XKM*XKC)*(HPSI/(D12*SDIF))
ISN 0097          PHI = (X(1)+X15)/DTR
```

```
ISN 0098          VPHI = X(2)*(XKM*XKC)+ HPHI*AITREN
ISN 0099          WPHI = X(3)*XKC*T1/T2  - T1 *AITREN*HPHI/(T2*XKM)
ISN 0100          THT = (X(4)+X4E)/DTR
ISN 0101          VTHT = X(5)*XKM*XKC +  HTHT*AITREN
ISN 0102          WTHT = X(6)*XKC*T1/T2  - T1* AITREN*HTHT/(T2*XKM)
ISN 0103          PSI = (X(7)+X7E)/DTR
ISN 0104          VPSI = X(8)*XKM*XKC + HPSI*AITREN
ISN 0105          WPSI = X(9)*XKC*T1/T2  - T1*AITREN*HPSI/(T2*XKM)
ISN 0106          JSW = 0
ISN 0107          MOVE = 0                                                      XX
ISN 0108          XLLIM = 6.9                                                   III
ISN 0109          PHLIM = PI                                                    III
ISN 0110          THLIM = PI                                                    III
ISN 0111          DIVIO = 4.
ISN 0112          DIVI = DIVIO
ISN 0113          LAY=0
ISN 0114          CALL CLOCK
ISN 0115          CALL AFX(JSW)
ISN 0116        3 CALL QGEN(THETA,PHIV,XLAM)
ISN 0117          DET = 1.0
ISN 0118          DO 704 I=1,9
ISN 0119          B(I) = XLAM(I)
ISN 0120          DET = DET * XLAM(I)
ISN 0121          DO 704 J=1,9
ISN 0122          PM(I,J) = Q(I,J)
ISN 0123      704 C(I,J)  = AAR(I,J)
             C     CALCULATION OF ATP (A TRANSPOSE P )
             C     CALCULATION OF PA  (P-MATRIX X A )
ISN 0124          DO 26 I=1,N
ISN 0125          DO 26 J=1,N
ISN 0126          DO 26 K=1,N
ISN 0127          ATP(I,J) = ATP(I,J) & A(K,I) *PM(K,J)
ISN 0128       26 PA(I,J)  = PA(I,J) &PM(I,K) * A(K,J)
ISN 0129          DO 27 I=1,N
ISN 0130          DO 27 J=1,N
ISN 0131       27  Q(I,J) = -ATP(I,J) - PA(I,J)
ISN 0132          DO 705 I=1,9
ISN 0133      705 IF(Q(I,I) .LE. 0.0) GO TO 706
ISN 0135          WRITE(6,3344) THETA,PHIV,XLAM
ISN 0136     3344 FORMAT( ' PARAMETERS OF P.MATRIX ',/ ' DATA-THETA,PHIV,XLAM',/
                 1 / (1X,9E14.7) )
ISN 0137          GO TO 445
ISN 0138      706 NCT = NCT + 1
ISN 0139          WRITE(6,707)NCT
ISN 0140      707 FORMAT(' Q(I,I) NEG -- NO. OF NEG MATRICES = ',I5 )
ISN 0141          GO TO 3
ISN 0142      445 VMIN = VMINI
ISN 0143          CALL CLOCK
ISN 0144          CALL DSRCH(VMIN,B,C,JSUE)                                     NP
             C                                          EUSJ )6993,6(ETI PW
ISN 0145     3996 FORMAT(  ' JSUE = ',I5 //)                                    NP
ISN 0146          CALL CLOCK
ISN 0147          VL= VMIN
ISN 0148          P = DLOG10(VOL)/30.                                           X
ISN 0149          IF(P.GE.PSR) GO TO 375
ISN 0151          WRITE(6,2552) NCUC
```

```
ISN 0152      2552 FORMAT( ' WE HAVE AN IMPROVEMENT AT NCUE = ',I6/)
ISN 0153           WRITE(6,5000) ( (A(I,J),J=1,9),I=1,9)
ISN 0154      5000 FORMAT(1H / 1X,11H  A-MATRIX  /(1X, 9E13.6 ))
ISN 0155           WRITE(6,5001) (F(I),I=1,9)
ISN 0156      5001 FORMAT(1H / 1X,12H F(X)-VECTOR / 1X,9E13.6)
ISN 0157           WRITE(6,9061) DET
ISN 0158      9061 FORMAT(1H 10X,'DETERMINANT OF P-MATRIX =',E14.7/)
ISN 0159           WRITE(6,9765)
ISN 0160      9765 FORMAT(1H / 1X,36H  Q FROM PUTTING P INIT -ATP-PA = Q , //)
ISN 0161           DO941 I=1,N
ISN 0162       941 WRITE(6,2963) ( Q(I,J),J=1,N)
ISN 0163      2963 FORMAT(1H /(1X,9E14.7) )
ISN 0164           WRITE(6,491) ((C(I,J),J=1,9),I=1,9)
ISN 0165       491 FORMAT(' MATRIX OF EIGENVECTORS IN COLUMNS '/(1X,9E14.7) )
ISN 0166           WRITE(6,100) VL,VOL
ISN 0167       100 FORMAT(1H /1X, 16H LIAPUNOV FCT  = E14.7 ,/
              1            1X,23H INVERSE VOL ESTIMATE = E14.7,/ )
ISN 0168           WRITE(6,580) P,PSR                                            V
ISN 0169       580 FORMAT( ' P =',E14.7,20X,'PSR=',E14.7//)                      V
            C      2600 IS THE SIZE OF BUNCH
ISN 0170       375 IF(NCUE.EQ.2599) GO TO 999
ISN 0172           BUNCH(NCUE,1) = T
ISN 0173           BUNCH(NCUE,2) = VL
ISN 0174           BUNCH(NCUE,3) = VOL
ISN 0175           BUNCH(NCUE,4) = DET
ISN 0176           BUNCH(NCUE,5) = P
ISN 0177           BUNCH(NCUE,6) = DIVI
ISN 0178       376 IF(NCUE.EQ.NXCUE) GO TO 999
ISN 0180           NCUE = NCUE + 1
ISN 0181           IF(NUKEY.NE.0) GO TO 4                                           III
ISN 0183           PSR = P                                                          III
ISN 0184           VSP = VOL                                                         NP
ISN 0185           WRITE(6,570) PSR                                                 III
ISN 0186       570 FORMAT(  ' P-STAR  = ',E14.7/)                                   III
ISN 0187         8 SIG2 = DLOG10(PSR)                                               VII
ISN 0188           WRITE(6,2503) SIG2
ISN 0189      2503 FORMAT( ' SIGMA-SQUARED = ',E14.7//)
ISN 0190        22 DO 9 I=1,45                                                      III
ISN 0191           RX(I) = RDM(GUM)                                                 III
ISN 0192           XLUV    = -1.+ 2.*RX(I)                                          III
ISN 0193           IF(XLUV.LE.0.0) GO TO 3415                                       III
ISN 0195           NUTU(I) = 1                                                      III
ISN 0196           GO TO 9                                                          III
ISN 0197      3415 NUTU(I) =-1                                                      III
ISN 0198         9 DU(I) =DEXP(-RX(I)**2/SIG2) * NUTU(I)                            III
            C                                           )54,1=I,)I(UD()065,6(ETI RW
ISN 0199       560 FORMAT( ' DU = ' //(1X,9E13.6/))                                 III
ISN 0200       905 CONTINUE                                                          XX
ISN 0201           MOVE = MOVE + 1                                                   XX
ISN 0202           IF(NCUE.GE.NSW)GO TO 2039
ISN 0204           IF(MOVE.LE.NSW1)GO TO 2166
ISN 0206           DIVI = 0.5*DIVI                                                   XX
ISN 0207           MOVE = 0                                                          XX
ISN 0208           IF (DIVI.LT.0.25) DIVI = 0.25
ISN 0210           GO TO 2166
ISN 0211      2039 IF(MOVE.LE.NSW2) GO TO 2166
```

```
ISN 0213          DIVI = 2.* DIVIO
ISN 0214          DIVIO = DIVI
ISN 0215          MOVE = 0
ISN 0216          IF (DIVI.GT.64.) DIVI = 64.
ISN 0218     2166 CONTINUE
            C                                                    IVID )2332,6(ETI RW 6612
ISN 0219     2332 FORMAT(  ' DIVI = ', E20.7 /)                                        XX
ISN 0220          DO 10 I=1,9                                                          XX
ISN 0221       10 DU(I) = XLLIM * DU(I)/DIVI                                          III
ISN 0222          DO 12 I=10,37                                                       III
ISN 0223       12 DU(I) = THLIM * DU(I)/DIVI                                          III
ISN 0224          DO 14 I=38,45                                                       III
ISN 0225       14 DU(I) = PHLIM * DU(I)/DIVI                                          III
            C                                              )54,I=I,)I(UD()065,6(ETI RW
ISN 0226          DO 20 I=1,9                                                         III
ISN 0227      665 XINC(I) = EX(I) + DU(I)                                             III
ISN 0228          IF(XINC(I).LT.XLLIM) GO TO 209                                      III
ISN 0230          DU(I) = 0.5 * DU(I)                                                 III
ISN 0231          GO TO 665                                                           III
ISN 0232      209 IF(XINC(I).LT.-9.2) XINC(I)=-9.2
ISN 0234       20 XLAM(I) =DEXP(XINC(I))                                              III
ISN 0235      665 DO 30 I=1,28                                                        III
ISN 0236          IP = I+9                                                            III
ISN 0237       52 THETA(I)= THETP(I) + DU(IP)
ISN 0238          IF( DABS(THETA(I)).LT.PI ) GO TO 30
ISN 0240          THETA(I) = THETA(I) - DU(IP)
ISN 0241       44 DU(IP) = 0.5* DU(IP)                                                III
ISN 0242          GO TO 52                                                            III
ISN 0243       30 CONTINUE                                                            III
ISN 0244          DO 40 I=1,8                                                         III
ISN 0245          IP= 37+I                                                            III
ISN 0246       62 PHIV(I) = PHIVP(I) + DU(IP)                                         III
ISN 0247          IF(DABS(PHIV(I)).LT.PI) GO TO 40
ISN 0249          PHIV(I) = PHIV(I) - DU(IP)
ISN 0250       54 DU(IP) =  0.5 * DU(IP)                                              III
ISN 0251          GO TO 62                                                            III
ISN 0252       40 CONTINUE                                                            III
            C                                              )54,I=I,)I(UD()065,6(ETI RW
ISN 0253          IF(KIKIT.GT.0) GO TO 50                                             III
ISN 0255          DO 71 I=1,5                                                         III
ISN 0256          PHIV(I) = 0.0                                                       III
ISN 0257       71 THETA(I) = 0.0                                                      III
ISN 0258          DO 72 I=8,12                                                        III
ISN 0259       72 THETA(I) = 0.0                                                      III
ISN 0260          DO 73 I= 14,20                                                      III
ISN 0261       73 THETA(I) = 0.0                                                      III
ISN 0262          THETA(23) =0.0                                                      III
ISN 0263          THETA(24) =0.0                                                      III
ISN 0264          DO 74 I=26,27                                                       III
ISN 0265       74 THETA(I) = 0.0                                                      III
ISN 0266          PHIV(6) =0.0                                                        III
ISN 0267          THETA(28) = THETA(22)                                               III
ISN 0268          PHIV(7) = THETA(21)                                                 III
ISN 0269          PHIV(9) =THETA(25)                                                  III
ISN 0270       50 NUKEY = 1                                                           III
ISN 0271          GO TO 3                                                             III
```

```
ISN 0272        4 IF(P.LT.PSR) GO TO 300                                          III
ISN 0274          IF(LMIN.EQ.0)    GC TG 305                                      III
ISN 0276          LMIN = 0                                                        III
ISN 0277          LAY= 0                                                          III
ISN 0278          GO TO 8                                                         III
ISN 0279      305 DO 840  I = 1,9
ISN 0280      840 DU(I) =-DU(I) * 1.* DIVI / XLLIM
ISN 0281          DO 850  I = 10,37
ISN 0282      850 DU(I) =-DU(I) * 1.* DIVI / THLIM
ISN 0283          DO 860  I = 38,45
ISN 0284      860 DU(I) =-DU(I) * 1.* DIVI / PHLIM
ISN 0285          LMIN = 1                                                        III
ISN 0286          GO TO 905                                                       III
ISN 0287      300 PSP = P
ISN 0288          VSP = VOL                                                        NP
ISN 0289          WRITE(6,3443)VSR,PSR
ISN 0290     3443 FORMAT( ' VSR =',E14.7, ' PSR = ',E14.7 /)
ISN 0291          LAY = LAY+1                                                     III
ISN 0292          MOVE = 0                                                         XX
ISN 0293          DIVI = DIVIO
ISN 0294          LMIN = 1                                                        III
ISN 0295          DO 310 I=1,9                                                    III
ISN 0296          EX(I) = DLOG(XLAM(I))                                           III
ISN 0297      310 XLAMP(I) = XLAM(I)                                              III
ISN 0298          DO 320 I=1,28                                                   III
ISN 0299      320 THETP(I) = THETA(I)                                             III
ISN 0300          DO 330 I= 1,8                                                   III
ISN 0301      330 PHIVP(I) = PHIV(I)                                              III
ISN 0302          WRITE(6,550)(XLAMP(I),I=1,9),(THETP(I),I=1,28),(PHIVP(I),I=1,8) III
ISN 0303      550 FORMAT( ' XLAM-PRIME   .  THETA-PRIME   . PHI-PRIME'//(1X,9E13.6/)) III
ISN 0304          DO 810  I = 1,9
ISN 0305      810 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /XLLIM                       III
ISN 0306          DO 820  I = 10,37
ISN 0307      820 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /THLIM                       III
ISN 0308          DO 830  I = 38,45
ISN 0309      830 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /PHLIM                       III
ISN 0310          GO TO 905                                                       III
ISN 0311      999 CALL RDMOUT(DUM)                                                  X
ISN 0312          WRITE(6,715) DUM                                                  X
ISN 0313      715 FORMAT(1HO Z8)                                                    X
ISN 0314          GO TO 102                                                         X
ISN 0315     1000 CONTINUE
ISN 0316          WRITE(6,191)(ALP(J),J=1,18)
ISN 0317          WRITE(6,8600)
ISN 0318     8600 FORMAT(18X,'TIME',12X,'LIAPUNOV FCT',9X,'INVERSE VOL',11X,'DET(P)'
                 1,11X,'PERFORMANCE',12X,'DIVISOR',/// )
ISN 0319          WRITE(6,8605) ((BUNCH(I,J),J=1,6),I=1,NXCUE)
ISN 0320     8605 FORMAT(13X,E14.7,6X,E14.7,6X,E14.7,6X,E14.7,6X,E14.7,6X,E14.7/)
ISN 0321          CALL EXIT
ISN 0322          END
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
  ISN 0002          SUBROUTINE  CLOCK
  ISN 0003          COMMON/BLK68/ T
  ISN 0004          DATA        I/1/
  ISN 0005          T = FLOAT(ICHRON(I))*.01
  ISN 0006          WRITE (5,1) T
  ISN 0007        1 FORMAT('0',90X,'CLOCK TIME',F16.2,5X,'SECONDS')
  ISN 0008          I = 0
  ISN 0009          RETURN
  ISN 0010          END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE DSRCH(VMIN,B,C,JSUE)                                  NP
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004          DIMENSION G(9),B(9),GG(9),XX(9),C(9,9),PX(9)
                 1,PF(9),QX(9),DEL(9)
ISN 0005          DIMENSION XZERO(9),FZERO(9)
ISN 0006          REAL*4  B,C,R1,R2
ISN 0007          COMMON/BLK 11/DB1,DB2,DG1,DG2
ISN 0008           COMMON/BLK77/X(15)
ISN 0009          COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0010          COMMON/ABORT/ DET, VSR ,VOL                                      NP
ISN 0011      722 SD = SQRT(DET)
ISN 0012          JSUE = 0                                                         NP
ISN 0013          IFLAG = 0
ISN 0014          KAM = 0
ISN 0015          XK12 =100
ISN 0016          NN = 0
ISN 0017          NNN= 1
ISN 0018          KFAIL = 0
            C
ISN 0019      100 DO 1050 I=1,9
ISN 0020          G(I)= VMIN/B(I)
ISN 0021     1050 GG(I)= SQRT(G(I))
ISN 0022      103 IF(NN .LT. 500*NNN)GO TO 800
ISN 0024      705 FORMAT(4H NN=,I6)
ISN 0025          NNN=NNN & 1
ISN 0026      800 NN = NN & 1
ISN 0027          IF(NN .GT. 5000) GO TO 311
ISN 0029          DO 101 I=1,9
ISN 0030          CALL BOXNO(R1,R2)
ISN 0031          XX(I) = R1
ISN 0032          IF(NN .LT. 1000)XX(I)= GG(I)*XX(I)/6.
ISN 0034          IF(NN .GE. 1000)XX(I)= GG(I)*XX(I)/3.
ISN 0036      101 CONTINUE
            C     XX(I) ARE THE EIGENVECTOR COORDINATES
            C     NOW TRANSFORM TO X(I) COORDINATES
ISN 0037          DO 930 I=1,9
ISN 0038      930 X(I)=0.0
ISN 0039          DO 935 I=1,9
ISN 0040          DO 935 J=1,9
ISN 0041      935 X(I)= X(I) & C(I,J)*XX(J)
            C     GENERATE VL
ISN 0042      259 VL=0.0
ISN 0043          DO 260 I=1,9
ISN 0044      260 PX(I)=0.0
ISN 0045          DO 261 I=1,9
ISN 0046          DO 261 J=1,9
ISN 0047      261 PX(I)=PX(I) & PM(I,J)*X(J)
ISN 0048          DO 262 I=1,9
ISN 0049      262 VL=VL & X(I)*PX(I)
ISN 0050          IF(VL .GE. VMIN)GO TO 103
ISN 0052          JSW=1
ISN 0053          CALL AFX(JSW)
            C     ***************************************************************
ISN 0054          VDOT =0.0                                                        VDOT 40
ISN 0055          DO 250 I=1,9                                                     VDOT 50
```

```
ISN 0056          PF(I)=0.0                                                        VDOT 60
ISN 0057      250 QX(I)=0.0                                                        VDOT 70
ISN 0058          DO 251 I=1,9                                                     VDOT 80
ISN 0059          DO 251 J=1,9                                                     VDOT 90
ISN 0060          QX(I)= QX(I) & Q(I,J)*X(J)                                       VDOT100
ISN 0061      251 PF(I)=PF(I) & PM(I,J)*F(J)                                       VDOT110
ISN 0062          DO 252 I=1,9                                                     VDOT120
ISN 0063      252 VDOT = VDOT - X(I)* (QX(I)-2.0 * PF(I))
ISN 0064          IF(IFLAG.EQ.1)GO TO 520
ISN 0066          IF(VDOT .LT. 0)GO TO 103
ISN 0068          IF(VMIN .LT. VL)GO TO 103
ISN 0070          IFLAG = 1
ISN 0071      500 DO 510 I=1,9
ISN 0072      510 DEL(I)= 0.5*X(I)
            C                                              )9,1=I,)I(X()055,6(ETI RW
ISN 0073      512 DO 515I=1,9
ISN 0074      515 X(I)= X(I)- DEL(I)
ISN 0075          GO TO 535
ISN 0076      520 DO 530 I=1,9
ISN 0077      530 DEL(I)= DEL(I)*.5
ISN 0078          IF(VDOT .GT.0.0)GO TO 512
ISN 0080          DO 540 I=1,9
ISN 0081      540 X(I)= X(I) + DEL(I)
ISN 0082      535 KAM= KAM + 1
            C                                                 LV,TODV)065,6(ETI RW
ISN 0083      560 FORMAT(7H VDOT,V/(1X,2E14.7))
            C                                              )9,1=I,)I(X()055,6(ETI RW
ISN 0084      550 FORMAT(3H X=/(1X,9E13.4))
ISN 0085          IF(KAM .LT.15) GO TO 259
ISN 0087          VMIN = VL
ISN 0088          DO 1100 I=1,9
ISN 0089          FZERO(I)=F(I)
ISN 0090     1100 XZERO(I)=X(I)
ISN 0091          NNZERO= NN
ISN 0092          VDOTZ=VDOT
            C                                                         NN)507,6(ETI RW
            C                             )9,1=I,)I(F(,2GD,1GD,2BD,1BD)213,6(ETI RW
ISN 0093      312 FORMAT(6H DB,DG,4E15.7/6H F(I)=,9E12.4 )
ISN 0094          IFLAG = 0
ISN 0095          KAM =0
ISN 0096          KFAIL = 1
ISN 0097          IF(VL .LT.1.0E-12) GO TO 66
ISN 0099          VOL= SD/((SQRT(VL))**9)
ISN 0100          IF(VOL.GT.VSR) GO TO 69                                            NP
ISN 0102          GO TO 100
ISN 0103       66 WRITE(6,2220) VL,NN
ISN 0104     2220 FORMAT(' VL = ',E14.7,5X,'TOO SMALL',5X,' NN=',I5 )
ISN 0105          VOL = 1.0E+50
            C
ISN 0106       69 JSUE = 1
            C                                                     NN)507,6(ETI RW 113
ISN 0107      311 IF(KFAIL.EQ.0) GO TO 315
ISN 0109          WRITE(6,1200) (FZERO(I),I=1,9),(XZERO(J),J=1,9)
ISN 0110     1200 FORMAT(10H FZERO(I)=,9E12.4/10H XZERO(I)=,9E12.4)
ISN 0111          WRITE(6,1201)NNZERO,VDOTZ,VMIN
ISN 0112     1201 FORMAT(8H NNZERO=,I5/12H VDOTZ,VMIN=,2E14.7)
```

```
ISN 0113          GO TO 316
ISN 0114     315 WRITE(6,2503)VMIN
ISN 0115    2503 FORMAT(' NO LINEAR SEARCH '/ ,' VMIN=' E14.7/)
ISN 0116          VMIN = 10.0* VMIN
ISN 0117          GO TO 722
ISN 0118     316 WRITE(6,705)NN
ISN 0119          RETURN
ISN 0120          END
```

C

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE AFX(JSW)
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
         C        ****************************************************************
ISN 0004          COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI
ISN 0005          COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM
ISN 0006          COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C
ISN 0007          COMMON /BLK4/ HPHI,HTHT,HPSI
ISN 0008          COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0009          COMMON/BLK 11/DB1,DB2,DG1,DG2
ISN 0010           COMMON/BLK77/X(15)
ISN 0011          COMMON/BLK78/X1E,X4E,X7E
         C
         C        EXACT MODEL STATE EQUATIONS
         C
ISN 0012          IF(JSW.GT.0)GO TO 312
         C         ( TRACKERS 3-4   (AMES 1-2)  )
         C
         C        EQUATIONS ARE IN THE FORM   X-DOT = A  X  & F(X)
         C
         C        WHERE X IS A  NINE COMPONENT COLUMN VECTOR   AS  IS  F(X)
         C
         C        AND  A  IS  9X9  MATRIX
         C
         C        X-VECTOR  IS  ( PHI , VPHI , WPHI , THT , VTHT , WTHT , PSI ,
         C
         C               VPSI ,   WPSI )
         C
         C        ****************************************************************
         C
ISN 0014          WRITE(6,1070)PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI,TM,T1,T2,
                 1XKM,XKC,A11,A13,D12,GAM1C,GAM2C,BET1C,BET2C,AITREN,HTHT,HPHI,HPSI
ISN 0015     1070 FORMAT(1H /10X,14H INITIAL STATE / 9E13.6 / 10X,13H INPUT CONSTS /
                 110X, 29H TM,T1,T2,XKM,XKC,A11,A13,D12 / 8E14.7 / 10X, 30H GAM1C,GA
                 2M2C,BET1C,BET2C,(RAD) / 4E14.7 / 10X,10H INERTIA = E14.7 ,/10X,
                 317H HTHT,HPHI,HPSI = 3E14.7//)
         C        ****************************************************************
ISN 0016          PI = 3.1415926
ISN 0017          DTR  = PI/180.0
ISN 0018          RTD  = 180.0/PI
         C        ****************************************************************
         C
ISN 0019          SB2C  = SIN(BET2C)
ISN 0020          CB2C  = COS(BET2C)
ISN 0021          TB2C = SB2C/CB2C
ISN 0022          SG1C  = SIN(GAM1C)
ISN 0023          CG1C  = COS(GAM1C)
ISN 0024          SG2C  = SIN(GAM2C)
ISN 0025          CG2C  = COS(GAM2C)
ISN 0026          SB1C  = SIN(BET1C)
ISN 0027          CB1C  = COS(BET1C)
ISN 0028          TB1C  = SB1C/CB1C
ISN 0029           SGAM1C = SG1C
ISN 0030           SGAM2C = SG2C
ISN 0031           CGAM1C = CG1C
ISN 0032           CGAM2C = CG2C
```

```
                 C
                 C      ****************************************************************
ISN 0033                DIF = GAM1C - GAM2C
ISN 0034                SDIF = SIN(DIF)
                 C
                 C      ****************************************************************
                 C
                 C      CALCULATION OF THE  A-MATRIX
                 C
ISN 0035                DO 10 I=1,9
ISN 0036                DO 10 J=1,9
ISN 0037             10 A(I,J) = 0.0
                 C
ISN 0038                A(1,2) = -XKM*XKC/AITREN                                        AFX
ISN 0039                A(7,8)=A(1,2)
ISN 0040                A(2,1) =  (T1&T2)/(TM*T2)                                       AFX
ISN 0041                A(2,2)=-1.0/TM
ISN 0042                A(5,5)=A(2,2)
ISN 0043                A(8,8)=A(5,5)
ISN 0044                A(3,3)=-1.0/T2
ISN 0045                A(6,6)=A(3,3)
ISN 0046                A(9,9)=-1/T2
ISN 0047                A(2,3) =  T1/(TM*T2)                                            AFX
ISN 0048                A(5,6)=A(2,3)
ISN 0049                A(8,9)=A(2,3)
ISN 0050                A(2,4) = A(2,1) *(-TB1C*CG1C)
ISN 0051                A(2,7)=A(2,1)*TB1C*SG1C
ISN 0052                A(3,1) = -1.0/T2                                                AFX
ISN 0053                A(3,4)=A(3,1)*A(2,4)/A(2,1)
ISN 0054                A(3,7)=A(3,1)*A(2,7)/A(2,1)
ISN 0055                A(4,5)=A(1,2)
ISN 0056                A(5,4) = A(2,1) * D12 * SDIF                                    AFX
ISN 0057                A(6,4)=A(3,1)*A(5,4)/A(2,1)
ISN 0058                A(8,7)=A(5,4)
ISN 0059                A(9,7)=A(6,4)
                 C
                 C
                 C      ****************************************************************
                 C
                 C      CALCULATION  OF DB1,DB2,DG1,DG2
                 C
                 C      NEEDS ANGLES FROM STATE VECTOR AND COMMAND ANGLES AS INPUT
                 C
                 C
                 C
ISN 0060            312 PHI= X(1) + X1E
ISN 0061                THT= X(4) + X4E
ISN 0062                PSI= X(7) + X7E
ISN 0063                SPHI = SIN(PHI)
ISN 0064                CPHI = COS(PHI)
ISN 0065                STHT = SIN(THT)
ISN 0066                CTHT = COS(THT)
ISN 0067                SPSI = SIN(PSI)
ISN 0068                CPSI = COS(PSI)
ISN 0069                TTHT = STHT/CTHT                                                AFX
ISN 0070                XNCRW = XKC*T1/T2                                               AFX
```

```
ISN 0071          PUP = 1.0/(XKC*TM)                                                    AFX
             C
ISN 0072          GR1= CPSI*CTHT*SB1C
ISN 0073          GR2= SPSI*CTHT*CG1C*CB1C
ISN 0074          GR3= STHT*SG1C*CB1C
ISN 0075          GR4= GR1 + GR2 + GR3
ISN 0076          DB1= ARSIN(GR4) - BET1C
             C
ISN 0077          DG1 =                                         ATAN(   (-SB1C*(SPSI*
                 1 SPHI & CPSI*STHT*CPHI) &CG1C*CB1C *(CPSI*SPHI -SPSI*STHT*CPHI)
                 2 &SG1C*CB1C*CTHT*CPHI ) /  (SB1C *(-SPSI*CPHI & CPSI*STHT*SPHI)
                 3 &CG1C*CB1C*(CPSI*CPHI & SPSI*STHT*SPHI) -CB1C*SG1C*CTHT*SPHI) )
                 4                                              - GAM1C
             C
ISN 0078          GL1= CPSI*CTHT*SB2C-SPSI*CTHT*CG2C*CB2C-STHT*SG2C*CB2C
ISN 0079          DB2= ARSIN(GL1) - BET2C
             C
ISN 0080          DG2 = -GAM2C & ATAN(  (SB2C*(SPSI*SPHI &CPSI*STHT*CPHI) &CG2C*
                 1 CB2C*(CPSI*SPHI-SPSI*STHT*CPHI) &SG2C*CB2C*CTHT*CPHI )/ (SB2C*
                 2 (SPSI*CPHI-CPSI*STHT*SPHI)&CG2C*CB2C*(CPSI*CPHI&SPSI*STHT*SPHI)
                 3 -CTHT*SPHI*SG2C*CB2C ) )
ISN 0081          IF(BET1C.EQ. 0.0) GO TO 850
ISN 0083          APE1= ABS(DB1/BET1C)
ISN 0084          IF(APE1 .LT. 1.D-10)DB1= SG1C*THT + CG1C*PSI
ISN 0086      850 IF(GAM1C .EQ. 0.0) GO TO 851
ISN 0088          APE2= ABS(DG1/GAM1C)
ISN 0089          IF(APE2 .LT. 1.D-10)DG1= PHI - TB1C*CG1C*THT + TB1C*SG1C*PSI
ISN 0091      851 IF(BET2C .EQ. 0.0) GO TO 852
ISN 0093          APE3= ABS(DB2/BET2C)
ISN 0094          IF(APE3 .LT. 1.D-10)DB2= -SG2C*THT - CG2C*PSI
ISN 0096      852 IF(GAM2C .EQ. 0.0) GC TO 853
ISN 0098          APE4= ABS(DG2/GAM2C)
ISN 0099          IF(APE4 .LT. 1.D-10)DG2= PHI + TB2C*CG2C*THT - TB2C*SG2C*PSI
             C
             C    ************************************************************************
             C
             C    CALCULATION OF NONLINEAR F(X)
             C
             C
ISN 0101      853 SUM3 = A(2,4)/A(2,1)
ISN 0102          SUM4 = A(2,7)/A(2,1)
ISN 0103          SUM5 =(COS(DG2&GAM2C))* DB1 & (COS(DG1 & GAM1C)) * DB2
ISN 0104          SUM6 =(SIN(DG2&GAM2C))* DB1 & (SIN(DG1 & GAM1C)) * DB2
ISN 0105          GAIN = XKC * (T1&T2)/T2                                               AFX
ISN 0106          F(1) = (TTHT * A(1,2))*(X(5)*SPHI & X(8)*CPHI)                        AFX
             C
ISN 0107          ARGF2= -(T1/T2)*AITREN*HPHI/XKM + GAIN*DG1 + XNORW*X(3)
ISN 0108          IF (ABS(ARGF2).LE.F2LIM) F2 = ARGF2
ISN 0110          IF (ARGF2.GT.F2LIM) F2 = F2LIM
ISN 0112          IF (ARGF2.LT.-F2LIM) F2= -F2LIM
ISN 0114          F(2)= -(-F2 &GAIN*(X(1)&SUM3*X(4) &SUM4*X(7))&AITREN*HPHI
                 1    /XKM ) * PUP    - A(2,3)*X(3)                                     AFX
             C
ISN 0115          F(3)=A(3,1)*(DG1-(X(1)+SUM3*X(4)+ SUM4*X(7)))-HPHI/(T2*A(1,2))
ISN 0116          F(4) = -A(1,2)* (X(5)*(1.0 - CPHI) & SPHI *X(8) )                     AFX
             C
```

160

```
ISN 0117          ARGF5= GAIN*SUM5*D12 -(T1/T2)*AITREN*HTHT/XKM + XNORW*X(6)
ISN 0118          IF(ABS(ARGF5).LE.F2LIM) F2 =ARGF5
ISN 0120          IF(ARGF5.GT.F2LIM) F2=F2LIM
ISN 0122          IF(ARGF5 .LT. -F2LIM)F2 = -F2LIM
ISN 0124          F(5) =-(-F2 &GAIN*D12*SDIF*X(4) &XNORW*X(6) &AITREN*HTHT/XKM)      AFX
                 1     * PUP                                                        AFX
             C
ISN 0125          F(6) = A(3,1)*D12*(SUM5 - SDIF*X(4)) -HTHT/(T2*A(1,2))
             C
ISN 0126          F(7) = A(1,3) * (X(5)*SPHI &X(8)*(CPHI-CTHT))/CTHT
             C
ISN 0127          ARGF8= -GAIN*D12*SUM6-(T1/T2)*AITREN*HPSI/XKM + XNORW*X(9)
ISN 0128          IF(ABS(ARGF8).LE.F2LIM) F2 = ARGF8
ISN 0130          IF(ARGF8.GT.F2LIM) F2 = F2LIM
ISN 0132          IF(ARGF8.LT.-F2LIM) F2=-F2LIM
ISN 0134          F(8) =-(-F2&GAIN*D12*SDIF*X(7) & XNORW*X(9) &AITREN*HPSI/XKM)*PUP
             C
ISN 0135          F(9)= -A(3,1)*D12*(SUM6+SDIF*X(7))-HPSI/(T2*A(1,2))
             C
             C
             C
             C    ******************************************************************
             C
ISN 0136          IF(JSW.GT.0)GO TO 602
ISN 0138      602 CONTINUE
             C
             C
             C    ******************************************************************
ISN 0139          RETURN
ISN 0140          END
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID

ISN 0002            SUBROUTINE QGEN(THETA,PHIV,XLAM)
ISN 0003            IMPLICIT REAL*8 (A-H,O-Z)
              C
              C     GENERATION OF POSITIVE DEFINITE Q MATRIX
ISN 0004            DOUBLE PRECISION AAMCO,P,QV,A,E,AM,C,PM,ATP,PA,QP
ISN 0005            COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0006            COMMON/BLK 70/ AAR(9,9),BM(9,9),PA(9,9),ATP(9,9)
ISN 0007            N=9
              C
ISN 0008            DIMENSION THETA(28),PHIV(8),XLAM(9),ZTHETA(28),TTHETA(28),
                   1           BA(20,20),SS(20,20,20),CC(20,20),Z(20,20,20),
                   2     SM(9,9),G(9,9),QQ(9,9)
              C          )9,1=K,)K(MALX(,)8,1=J,)J(VIHP(,)82,1=I,)I(ATEHT()3601,5(DA ER
ISN 0009       1063 FORMAT(6E12.4)
ISN 0010            PI = 3.1415926
ISN 0011            PI2 = PI/2.
ISN 0012            NN=(N-1)*(N-2)/2
ISN 0013            DO 6 I=1,NN
ISN 0014            BAD=THETA(I)
ISN 0015          6 TTHETA(I)= AMOD(BAD,PI )
              C     WE HAVE NOW INDEXED THETA.
              C     NOW WANT CONTINUED PRODUCT OF SS(I,J,L) FOR L=K&1,N .
              C     FOR EACH K=1,N-1 OBTAIN Z(K,I,J).
ISN 0016            NNI = N-1
ISN 0017         69 DO 20 K=1,NNI
              C
ISN 0018            DO 8 I=1,N
ISN 0019            DO 9 J=1,N
ISN 0020          8 BA(I,J)=0.0
ISN 0021            DO 99 I=1,N
ISN 0022         99 BA(I,I)=1.0
              C
ISN 0023            KK=K&1
ISN 0024            DO 10 L=KK,N
              C
ISN 0025            DO 15 I=1,N
ISN 0026            DO 15 J=1,N
ISN 0027         15 SS(I,J,L)=0.0
ISN 0028            DO 98 I=1,N
ISN 0029         98 SS(I,I,L)=1.0
              C     WE DEVELOP SS(I,J,L) AS FUNCTION THETA(L,K,N) FOR L L.T. N
              C     AND SS(I,J,L) FUNCTION OF PHIV(K) FOR L=N
ISN 0030            IF(L-N)25,23,23
ISN 0031         25 M=((2*N -K-2)*(K-1)/2)&N-L
ISN 0032            SS(K,K,L)=COS(TTHETA(M))
ISN 0033            SS(L,L,L)=COS(TTHETA(M))
ISN 0034            SS(K,L,L)=-SIN(TTHETA(M))
ISN 0035            SS(L,K,L)=SIN(TTHETA(M))
ISN 0036            GO TO 35
ISN 0037         23 SS(K,K,L)=COS(PHIV(K))
ISN 0038            SS(L,L,L)=COS(PHIV(K))
ISN 0039            SS(K,L,L)=-SIN(PHIV(K))
ISN 0040            SS(L,K,L)=SIN(PHIV(K))
              C
ISN 0041         35 DO 70 I=1,N
```

```
ISN 0042          DO 70 J=1,N
ISN 0043       70 CC(I,J)=0.0
            C
ISN 0044          DO 50 M=1,N
ISN 0045          DO 50 J=1,N
ISN 0046          DO 50 I=1,N
ISN 0047       50 CC(M,J)=BA(M,I)*SS(I,J,L) &CC(M,J)
ISN 0048          DO110 I=1,N
ISN 0049          DO110 J=1,N
ISN 0050      110 BA(I,J)=CC(I,J)
ISN 0051       10 CONTINUE
ISN 0052          DO 20 I=1,N
ISN 0053          DO 20 J=1,N
ISN 0054       20 Z(K,I,J)=BA(I,J)
            C
ISN 0055          DO 7 I=1,N
ISN 0056          DO 7 J=1,N
ISN 0057        7 BM(I,J)=0.0
ISN 0058          DO 16 I=1,N
ISN 0059       16 BM(I,I)=1.0
            C
ISN 0060          DO 40 K=1,NNI
            C
ISN 0061          DO 75 I=1,N
ISN 0062          DO 75 J=1,N
ISN 0063       75 SM(I,J)=0.0
            C
ISN 0064          DO 55 M=1,N
ISN 0065          DO 55 J=1,N
ISN 0066          DO 55 I=1,N
ISN 0067       55 SM(M,J)=Z(K,M,I )*BM(I,J)&SM(M,J)
ISN 0068          DO 40 I=1,N
ISN 0069          DO 40 J=1,N
ISN 0070       40 BM(I,J)=SM(I,J)
            C
            C     BM(I,J) IS CONTINUED PRODUCT OF Z(K,I,J) FROM K=1 TO N-1
            C
ISN 0071          IF(PP)41,41,19
ISN 0072       19 CONTINUE
ISN 0073       18 FORMAT(8H BM(I,J)/(6E15.7))
ISN 0074       41 DO 78 I=1,N
ISN 0075          DO 78 J=1,N
ISN 0076       78 AAR(I,J)=BM(J,I)
            C
            C     AAR(I,J) IS TRANSPOSE BM(I,J)
            C
ISN 0077          DO 82 I=1,N
ISN 0078          DO 82 J=1,N
ISN 0079       82 G(I,J)=0.0
ISN 0080          DO 85 I=1,N
ISN 0081       85 G(I,I)=XLAM(I)
            C     G(I,J) IS THE LAMDA MATRIX
            C
ISN 0082          DO 86 I=1,N
ISN 0083          DO 86 J=1,N
ISN 0084       86 DD(I,J)=0.0
```

```
ISN 0085          DO 88  I=1,N
ISN 0086          DO 88  J=1,N
ISN 0087          DO 88  M=1,N
ISN 0088       88 QQ(I,J)=G(I,M)*BM(M,J)&QQ(I,J)
            C
            C     QQ(I,J)=LAMDA MATRIX *BM(I,J)
            C
ISN 0089          DO 90  I=1,N
ISN 0090          DO 90  J=1,N
ISN 0091       90 Q(I,J)=0.0
ISN 0092          DO 95  I=1,N
ISN 0093          DO 95  J=1,N
ISN 0094          DO 95  M=1,N
ISN 0095       95 Q(I,J)=AAR(I,M)*QQ(M,J)&Q(I,J)
            C                                     )N,1=I,)N,1=J,)J,I(Q (( )39,6(ETI RW
ISN 0096       93 FORMAT(1H /1X,7H Q(I,J)/(1X,9E14.7) )
ISN 0097          RETURN
ISN 0098          END
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=55,SOURCE,BCD,LIST,NODECK,LOAD,MAP,NOEDIT,ID
  ISN 0002          SUBROUTINE BOXNO (R1,R2)
  ISN 0003          T1 =  SQRT(-2.0 *ALOG(RDM (DUM)))
  ISN 0004          T2 =  6.2831853 * RDM (DUM)
  ISN 0005          R1 =  T1 * COS(T2)
  ISN 0006          R2 =  T1 * SIN(T2)
  ISN 0007          RETURN
  ISN 0008          END
```

```
LEVEL 2 FEB 67                          OS/360   FORTRAN H                          DATE  69.167/22.00.17

        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

ISN 0002             IMPLICIT REAL*8 (A-H,O-Z)                                          10100010
               C     Q - OPTIMIZATION PROGRAM      8-20-68   INITIATE                   10100020
ISN 0003             DIMENSION THETA(28),PHIV(8),XLAM(9),RHV(9),SLUF(9)                 10100030
ISN 0004             DIMENSION  XLMX(9)                                                 10100040
ISN 0005             DIMENSION B(6),C(6,6)                                              10100050
ISN 0006             DIMENSION ABC(6,6),PRM(6,6)                                        10100060
ISN 0007             REAL*4  B,C,DUM,GUM                                                10100070
ISN 0008             DIMENSION ALP(18)                                                  10100080
ISN 0009             DIMENSION DU(45),RX(45),NUTU(45),XLAMP(9),THETP(28),PHIVP(8)       10100090
ISN 0010             DIMENSION EX(9),XINC(9)                                            10100100
ISN 0011             DIMENSION DINC(6)                                                  10100110
ISN 0012             DIMENSION BUNCH(5001,6)                                            MAIN
ISN 0013             COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI            10100130
ISN 0014             COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM            10100140
ISN 0015             COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C                              10100150
ISN 0016             COMMON /BLK4/ HPHI,HTHT,HPSI                                       10100160
ISN 0017             COMMON/BLK5/A(6,6),Q(6,6),PM(6,6),F(6),DF(6,6)                     10100170
ISN 0018             COMMON/BLK 11/DB1,DB2,DG1,DG2                                      10100180
ISN 0019              CCMMON/BLK77/X(15)                                                10100190
ISN 0020             COMMON/BLK78/X1E,X3E,X5E                                           10100200
ISN 0021             COMMON/GEORGE/INIT                                                 MAIN
ISN 0022             COMMON/ ABORT/ DET, VSR, VOL                                       10100210
ISN 0023             COMMON/ BLK68/ T                                                   10100220
ISN 0024             COMMON/BLKDS/VDOTZ,XZERO(6),FZERO(6),NNZERO                        10100230
ISN 0025             CALL SETCLK                                                        MAIN
ISN 0026        1001 FORMAT(5E14.7)                                                     10100240
ISN 0027         102 READ(5,1101,END=1000)ALP                                           10100250
               C     KWITURBELYAKIN                                                     10100260
ISN 0028        1101 FORMAT(18A4)                                                       10100270
ISN 0029             WRITE(6,191)(ALP(J),J=1,18)                                        10100280
ISN 0030         191 FORMAT(1H / 10X,18A4)                                              10100290
ISN 0031             KEEP = 0                                                           10100300
ISN 0032         751 TM     = 76.8                                                      10100310
ISN 0033             N=6                                                                10100320
ISN 0034             T1     =  4.5                                                      10100330
ISN 0035             T2     =  0.5                                                      10100340
ISN 0036             XKM    =  1.0/13.0                                                 10100350
ISN 0037             XKC    =  2.685 E+05                                               10100360
ISN 0038             A13    =  0.0                                                      10100370
ISN 0039             A11      = 0.0                                                     10100380
ISN 0040             AITREN= 1500.0                                                     10100390
ISN 0041             F2LIM = 26.0                                                       10100400
ISN 0042             READ(5,1001)GAM1C,GAM2C,BET1C,BET2C                                10100410
ISN 0043             READ(5,1001)HPHI,HTHT,HPSI                                         10100420
ISN 0044             READ(5,1001)VMINI                                                  10100430
ISN 0045             READ(5,712) DUM                                                    10100440
ISN 0046         712 FORMAT(Z8)                                                         10100450
ISN 0047             READ(5,1104)NSKIP,LMIN,NUKEY,KIKIT,NXCUE,NSW,NSW1,NSW2
ISN 0048        1104 FORMAT(18I4)
ISN 0049             IF(DUM)713,713,714                                                 10100460
ISN 0050         714 CALL RDMIN(DUM)                                                    10100470
ISN 0051         713 CONTINUE                                                           10100480
ISN 0052             WRITE(6,81)VMINI                                                   10100490
ISN 0053          81 FORMAT ( 7H VMINI= E14.7)                                          10100500
ISN 0054             PI= 3.1415926                                                      10100510
```

```
ISN 0055          PI2 = PI/2.                                                        10100520
ISN 0056          DTR = PI/180.                                                      10100530
ISN 0057          GAM1C = GAM1C * DTR                                                10100540
ISN 0058          GAM2C = GAM2C * DTR                                                10100550
ISN 0059          BET1C = BET1C * DTR                                                10100560
ISN 0060          BET2C = BET2C * DTR                                                10100570
ISN 0061          DIF = GAM1C - GAM2C                                                10100580
ISN 0062          D12 =  DIF * 2.0 / DABS(DIF)                                       10100590
ISN 0063          SDIF=SIN(DIF)                                                      10100600
ISN 0064          DO 201 I=1,10                                                      10100610
ISN 0065      201 THETA(I) = 0.                                                      10100620
ISN 0066          DO 202 I=1,5                                                       10100630
ISN 0067      202 PHIV(I) = 0.0                                                      10100640
ISN 0068          DO 203 I=1,6                                                       10100650
ISN 0069          XLMX(I) = 1.0                                                      10100660
ISN 0070          X(I) = 1.0                                                         10100670
ISN 0071      203 XLAM(I)= 1.0                                                       10100680
ISN 0072          IF(NSKIP .EQ. 0) GO TO 9299
ISN 0074          READ(5,1001)(XLAM(I),I=1,6),(THETA(J),J=1,10),(PHIV(K),K=1,5),
                 1     PSR,VSR
ISN 0075     9299 DO 420 I=1,6
ISN 0076      420 XLAMP(I) = XLAM(I)                                                 10100810
ISN 0077          DO 6009 I=1,6                                                      10100750
ISN 0078     6009 EX(I) = DLOG(XLAM(I))                                              10100760
ISN 0079          DO 430 I=1,10                                                      10100820
ISN 0080      430 THETP(I) = THETA(I)                                                10100830
ISN 0081          DO 440 I=1,5                                                       10100840
ISN 0082      440 PHIVP(I) = PHIV(I)                                                 10100850
ISN 0083          WRITE(6,550)(XLAMP(I),I=1,6),(THETP(I),I=1,10),(PHIVP(I),I=1,5)    10100860
ISN 0084          SG1C = SIN(GAM1C)                                                  10100870
ISN 0085          CG1C = COS(GAM1C)                                                  10100880
ISN 0086          SG2C = SIN(GAM2C)                                                  10100890
ISN 0087          CG2C = COS(GAM2C)                                                  10100900
ISN 0088          TB1C = SIN(BET1C)/COS(BET1C)                                       10100910
ISN 0089          X1E = ( AITREN)/(XKM*XKC)*(HPHI-HTHT*(A11*SG1C-A13*SG2C-CG1C*TB1C   10100920
                 1  )/(D12*SDIF) +HPSI*(A11*CG1C-A13*CG2C+SG1C*TB1C)/(D12*SDIF))     10100930
ISN 0090          X3E = ( AITREN)/(XKM*XKC)*(HTHT/(D12*SDIF))                        10100940
ISN 0091          X5E = (-AITREN)/(XKM*XKC)*(HPSI/(D12*SDIF))                        10100950
ISN 0092          PHI = (X(1)+X1E)/DTR                                               10100960
ISN 0093          VPHI = X(2)*(XKM*XKC)+ HPHI*AITREN                                 10100970
ISN 0094          THT= (X(3)+X3E)/DTR                                                10100980
ISN 0095          VTHT= X(4)*XKM*XKC + HTHT*AITREN                                   10100990
ISN 0096          PSI= (X(5)+ X5E)/DTR                                               10101000
ISN 0097          VPSI = X(6)*XKM*XKC + HPSI*AITREN                                  10101010
ISN 0098          VMAX = 0.0                                                         10101020
ISN 0099          NPLSS = 2                                                          10101030
ISN 0100          JSW = 0                                                            10101040
ISN 0101          KARP = 1                                                           10101050
ISN 0102          MOVE = 0                                                           10101060
ISN 0103          XLLIM = 6.9                                                        10101070
ISN 0104          PHLIM = PI                                                         10101080
ISN 0105          THLIM = PI2                                                        10101090
ISN 0106          DIVIO = 4.                                                         10101100
ISN 0107          DIVI = DIVIO                                                       10101110
ISN 0108          PMIN  = 30.                                                        10101120
            C     IF VSR AND PSR ARE GIVEN THEN NUKEY=LMIN=1                         10101130
```

```
ISN 0109          NCUE = 1                                          10101190
ISN 0110          LAY=0                                             10101200
ISN 0111          CALL AFX(JSW)                                     10101250
ISN 0112        3 CALL QGEN(THETA,PHIV,XLAM)                        10101270
ISN 0113          CALL PEAIQ(KEEP)                                  10101290
ISN 0114          KEEP = 1                                          10101300
         C                                                          10101310
ISN 0115          DO 703 I=1,6                                      10101330
ISN 0116          DO 703 J=1,6                                      10101340
ISN 0117          ABC(I,J)=PM(I,J)                                  10101350
ISN 0118      703 PRM(I,J)=PM(I,J)                                  10101360
ISN 0119          CALL DEIGN(ABC,B,C)                               10101380
ISN 0120          DO 6 I=1,6                                        10101400
ISN 0121        6 RHV(I) = 0.0                                      10101410
ISN 0122          DET = 1.0                                         10101420
ISN 0123          KDET= IMEQD(6,6,1,PRM,RHV,DET,SLUF)               10101440
ISN 0124          WRITE(6,9061)NCUE,DET                             10101460
ISN 0125     9061 FORMAT(' NCUE=',I5 ,  'DET-P= ',E14.7 )           10101470
         C        NEGATIVE EIGENVALUE ABORT                         10101480
ISN 0126          DO 21 I=1,6                                       10101490
ISN 0127       21 IF(B(I).LT.0.0)GO TO 22                           10101500
ISN 0129          VMIN = VMINI                                      10101510
ISN 0130          CALL DSRCH(VMIN,B,C,JSUE)                         10101530
ISN 0131     3996 FORMAT('JSUE=',I5//)                              10101540
ISN 0132          VL= VMIN                                          10101550
ISN 0133          P= DLOG10(VOL)/20.                                10101560
ISN 0134      580 FORMAT( ' P =',E14.7,20X,'PSR=',E14.7//)          10101570
ISN 0135          WRITE(6,580) P,PSR                                10101580
ISN 0136          BUNCH(NCUE,1) = T                                 10101590
ISN 0137          BUNCH(NCUE,2)= VL                                 10101600
ISN 0138          BUNCH(NCUE,3)= VOL                                10101610
ISN 0139          BUNCH(NCUE,4)= DET                                10101620
ISN 0140          BUNCH(NCUE,5)= P                                  10101630
ISN 0141          BUNCH(NCUE,6)= DIVI                               10101640
ISN 0142          IF(NCUE.EQ.NXCUE) GO TO 999                       10101650
ISN 0144          NCUE = NCUE + 1                                   10101660
ISN 0145          IF(NUKEY.NE.0) GO TO 4                            10101670
ISN 0147          PSR = P                                           10101680
ISN 0148          VSR = VOL                                         10101690
ISN 0149          WRITE(6,570) PSR                                  10101700
ISN 0150      570 FORMAT(  ' P-STAR = ',E14.7/)                     10101710
ISN 0151        8 SIG2 = DLOG10(PSR)                                10101720
ISN 0152          WRITE(6,2503) SIG2                                10101730
ISN 0153     2503 FORMAT( ' SIGMA-SQUARED = ',E14.7//)              10101740
ISN 0154       22 DO 9 I=1,21                                       10101750
ISN 0155          RX(I) = RDM(GUM)                                  10101760
ISN 0156          XLUV    = -1.+ 2.*RX(I)                           10101770
ISN 0157          IF(XLUV.LE.0.0) GO TO 3415                        10101780
ISN 0159          NUTU(I) = 1                                       10101790
ISN 0160          GO TO 9                                           10101800
ISN 0161     3415 NUTU(I) =-1                                       10101810
ISN 0162        9 DU(I) =DEXP(-RX(I)**2/SIG2) * NUTU(I)             10101820
         C        WRITE(6,560)(DU(I),I=1,21)                        10102340
ISN 0163      560 FORMAT( ' DU = ' //(1X,9E13.6/))                  10101850
ISN 0164      905 CONTINUE                                          10101860
ISN 0165          MOVE = MOVE + 1                                   10101870
```

```
ISN 0166          IF(NCUE .GE. NSW)GO TO 2039
ISN 0168          IF(MOVE .LE. NSW1)GO TO 2166
ISN 0170          DIVI = 0.5 * DIVI                                          10101900
ISN 0171          MOVE = 0                                                   10101910
ISN 0172          IF(DIVI .LT..25) DIVI= .25                                 10101920
ISN 0174          GO TO 2166                                                 10101930
ISN 0175     2039 IF(MOVE.LE.NSW2)GC TO 2166
ISN 0177          DIVI = 2.* DIVIO                                           10101950
ISN 0178          DIVIO = DIVI                                               10101960
ISN 0179          MOVE = 0                                                   10101970
ISN 0180          IF(DIVI .GT. 64.)DIVI = 64.                                10101980
            C2166 WRITE(6,2332)DIVI                                          10101990
ISN 0182     2166 CONTINUE
ISN 0183     2332 FORMAT( 'DIVI=',E20.7/)                                    10102000
ISN 0184          DO 10 I=1,6                                                10102010
ISN 0185       10 DU(I) = XLLIM * DU(I)/DIVI                                 10102020
ISN 0186          DO 12 I=7,16                                               10102030
ISN 0187       12 DU(I) = THLIM * DU(I)/DIVI                                 10102040
ISN 0188          DO 14 I=17,21                                              10102050
ISN 0189       14 DU(I) = PHLIM * DU(I)/DIVI                                 10102060
            C     WRITE(6,560)(DU(I),I=1,21)                                 10101840
ISN 0190          DO 20 I=1,6                                                10102100
ISN 0191      665 XINC(I) = EX(I) + DU(I)                                    10102110
ISN 0192          IF(XINC(I) .LT. XLLIM) GO TO 209                           10102120
ISN 0194          DU(I) = 0.5 * DU(I)                                        10102130
ISN 0195          GO TO 665                                                  10102140
ISN 0196      209 IF(XINC(I) .LT. -9.2) XINC(I)= -9.2                        10102150
ISN 0198       20 XLAM(I) =DEXP(XINC(I))                                     10102160
ISN 0199      666 DO 30I=1,10                                                10102170
ISN 0200          IP=I + 6                                                   10102180
ISN 0201       52 THETA(I)= THETP(I) + DU(IP)                                10102190
ISN 0202          IF( DABS(THETA(I)).LT.PI2) GO TO 30                        10102200
ISN 0204          THETA(I) = THETA(I) - DU(IP)                               10102210
ISN 0205       44 DU(IP) = 0.5* DU(IP)                                       10102220
ISN 0206          GO TO 52                                                   10102230
ISN 0207       30 CONTINUE                                                   10102240
ISN 0208          DO 40I=1,5                                                 10102250
ISN 0209          IP=16 + I                                                  10102260
ISN 0210       62 PHIV(I) = PHIVP(I) + DU(IP)                                10102270
ISN 0211          IF(DABS(PHIV(I)).LT.PI) GO TO 40                           10102280
ISN 0213          PHIV(I) = PHIV(I) - DU(IP)                                 10102290
ISN 0214       54 DU(IP) =  0.5 * DU(IP)                                     10102300
ISN 0215          GO TO 62                                                   10102310
ISN 0216       40 CONTINUE                                                   10102320
            C     WRITE(6,560)(DU(I),I=1,21)                                 10102080
ISN 0217          IF(KIKIT.GT.0) GO TO 50                                    10102360
ISN 0219          DO 71 I=1,3                                                10102370
ISN 0220          PHIV(I)=0.0                                                10102380
ISN 0221       71 THETA(I)=0.0                                               10102390
ISN 0222          DO 72 I=5,8                                                10102400
ISN 0223       72 THETA(I)=0.0                                               10102410
ISN 0224          THETA(10)=0.0                                              10102420
ISN 0225          PHIV(4)=0.0                                                10102430
ISN 0226          PHIV(5)= THETA(9)                                          10102440
ISN 0227       50 NUKEY = 1                                                  10102460
ISN 0228          GO TO 3                                                    10102470
```

```
ISN 0229          4 IF(P.LT.PSR) GO TO 300                                                  10102480
ISN 0231            IF(LMIN.EQ.0)    GO TO 305                                              10102490
ISN 0233            LMIN = 0                                                                10102500
ISN 0234            LAY= 0                                                                  10102510
ISN 0235            CALL CLOCK                                                              10102520
ISN 0236            GO TO 8                                                                 10102530
ISN 0237        305 DO 840 I=1,6                                                            10102540
ISN 0238        840 DU(I) =-DU(I) * 1.* DIVI / XLLIM                                        10102550
ISN 0239            DO 850 I=7,16                                                           10102560
ISN 0240        850 DU(I) =-DU(I) * 1.* DIVI / THLIM                                        10102570
ISN 0241            DO 860 I=17,21                                                          10102580
ISN 0242        860 DU(I) =-DU(I) * 1.* DIVI / PHLIM                                        10102590
ISN 0243            LMIN = 1                                                                10102600
ISN 0244            CALL CLOCK                                                              10102610
ISN 0245            GO TO 905                                                               10102620
ISN 0246        300 PSR = P                                                                 10102630
ISN 0247            LAY = LAY+1                                                             10102640
ISN 0248            VSR = VOL                                                               10102650
ISN 0249            MOVE = 0                                                                10102660
ISN 0250            DIVI = DIVIO                                                            10102670
ISN 0251            LMIN = 1                                                                10102680
ISN 0252            WRITE(6,100) VL,VCL                                                     10102690
ISN 0253        100 FORMAT(1H /1X, 16H LIAPUNOV FCT  = E14.7 ,/                             10102700
                  1           1X,23H INVERSE VOL ESTIMATE = E14.7,/ )                       10102710
ISN 0254            WRITE(6,93) (( Q(I,J),J=1,N),I=1,N)                                     10102720
ISN 0255         93 FORMAT(1H /1X,7H Q(I,J)/(1X,6E14.7))                                    10102730
ISN 0256            WRITE(6,11)((PM(I,J),J=1,6),I=1,6)                                      10102740
ISN 0257         11 FORMAT('0 PM'/(1P6E18.7))                                               10102750
ISN 0258       1020 WRITE(6,1030)(B(I),I=1,6)                                               10102760
ISN 0259       1030 FORMAT(13H1 EIGENVALUES//(1P6E20.7))                                    10102770
ISN 0260            WRITE(6,700)((C(I,J),J=1,6),I=1,6)                                      10102780
ISN 0261        700 FORMAT(13H EIGENVECTORS/(6E12.4))                                       10102790
ISN 0262            WRITE(6,1200)(FZERO(I),I=1,6),(XZERO(J),J=1,6)                          10102800
ISN 0263       1200 FORMAT(10H FZERO(I)=,6E12.4/10H XZERO(I)=,6E12.4)                       10102810
ISN 0264            WRITE(6,1201)NNZERO,VDOTZ,VMIN                                          10102820
ISN 0265       1201 FORMAT(8H NNZERO=,I5/12H VDOTZ,VMIN=,2E14.7)                            10102830
ISN 0266            DO 310 I=1,6                                                            10102850
ISN 0267            EX(I) = DLOG(XLAM(I))                                                   10102860
ISN 0268        310 XLAMP(I) = XLAM(I)                                                      10102870
ISN 0269            DO 320 I=1,10                                                           10102880
ISN 0270        320 THETP(I) = THETA(I)                                                     10102890
ISN 0271            DO 330 I=1,5                                                            10102900
ISN 0272        330 PHIVP(I) = PHIV(I)                                                      10102910
ISN 0273            WRITE(6,550)(XLAMP(I),I=1,6),(THETP(I),I=1,10),(PHIVP(I),I=1,5)         10102930
ISN 0274        550 FORMAT( ' XLAM-PRIME   ,  THETA-PRIME   , PHI-PRIME'//(1X,9E13.6/))     10102940
ISN 0275            DO 810 I=1,6                                                            10102960
ISN 0276        810 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /XLLIM                               10102970
ISN 0277            DO 820 I=7,16                                                           10102980
ISN 0278        820 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /THLIM                               10102990
ISN 0279            DO 830 I=17,21                                                          10103000
ISN 0280        830 DU(I) = (2.**(LAY-1))*DU(I) * DIVI /PHLIM                               10103010
ISN 0281            CALL CLOCK                                                              10103020
ISN 0282            GO TO 905                                                               10103030
ISN 0283        999 CALL RDMOUT(DUM)                                                        10103040
ISN 0284            WRITE(6,715) DUM                                                        10103050
ISN 0285        715 FORMAT(1H0 Z8)                                                          10103060
```

```
ISN 0286          GO TO 102                                                         10103070
ISN 0287     1000 CONTINUE                                                          10103080
ISN 0288          WRITE(6,8600)                                                     10103090
ISN 0289     8600 FORMAT(18X,'TIME',12X,'LIAPUNOV FCT',9X,'INVERSE VOL',11X,'DET(P)' 10103100
                 1,11X,'PERFORMANCE',12X,'DIVISOR',///)                             10103110
ISN 0290          WRITE(6,8605) ((BUNCH(I,J),J=1,6),I=1,NXCUE)                      10103120
ISN 0291     8605 FORMAT(7X,6(6X,E14.7)/)                                           10103130
ISN 0292          CALL EXIT                                                         10103140
ISN 0293          END                                                               10103150
```

```
        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID
ISN 0002          SUBROUTINE DSRCH(VMIN,B,C,JSUE)                                 10300010
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)                                       10300020
ISN 0004          DIMENSION G(6),B(6),GG(6),XX(6),C(6,6),PX(6),PF(6),             10300030
                 1          GX(6),DEL(6)                                          10300040
ISN 0005          REAL*4  B,C,R1,R2                                               10300060
ISN 0006          COMMON/BLKDS/VDOTZ,XZERO(6),FZERO(6),NNZERO
ISN 0007          COMMON/BLK 11/DB1,DB2,DG1,DG2                                   10300070
ISN 0008          COMMON /BLK5/A(6,6),Q(6,6),FM(6,6),F(6),DF(6,6)                 10300080
ISN 0009          COMMON/BLK77/X(15)                                              10300090
ISN 0010          COMMON/ BLK68/ T                                                10300100
ISN 0011          COMMON/ABORT/ DET,VSR,VCL                                       10300110
ISN 0012      722 KFAIL = 0                                                       10300120
ISN 0013          SD = SQRT(DET)                                                  10300130
ISN 0014          JSUE = 0                                                        10300140
ISN 0015          IFLAG = 0                                                       10300150
ISN 0016          KAM = 0                                                         10300160
ISN 0017          XK12 =100                                                       10300170
ISN 0018          NN = 0                                                          10300180
ISN 0019          NNN= 1                                                          10300190
            C                                                                     10300200
ISN 0020      100 DO 1050 I=1,6                                                   10300210
ISN 0021          G(I)= VMIN/B(I)                                                 10300220
ISN 0022     1050 GG(I)= SQRT(G(I))                                               10300230
ISN 0023      103 IF(NN .LT. 50*NNN)GO TO 800                                     10300240
ISN 0025      705 FORMAT(4H NN=,I6)                                               10300250
ISN 0026          NNN=NNN + 1                                                     10300260
ISN 0027      800 NN = NN + 1                                                     10300270
ISN 0028          IF(NN .GT. 1.E+05) GO TO 311
ISN 0030          DO 101 I=1,6                                                    10300290
ISN 0031          CALL BCXNO(R1,R2)                                               10300300
ISN 0032          XX(I) = R1                                                      10300310
ISN 0033          IF(NN .LT. 100)XX(I)= GG(I)*XX(I)/6.                            10300320
ISN 0035          IF(NN .GE. 100)XX(I)= GG(I)*XX(I)/3.                            10300330
ISN 0037      101 CONTINUE                                                        10300340
            C     XX(I) ARE THE EIGENVECTOR COORDINATES                           10300350
            C     NOW TRANSFORM TO X(I) COORDINATES                               10300360
ISN 0038          DO 930 I=1,6                                                    10300370
ISN 0039      930 X(I)=0.0                                                        10300380
ISN 0040          DO 935 I=1,6                                                    10300390
ISN 0041          DO 935 J=1,6                                                    10300400
ISN 0042      935 X(I)= X(I) + C(I,J)*XX(J)                                       10300410
            C     GENERATE VL                                                     10300420
ISN 0043      259 VL=0.0                                                          10300430
ISN 0044          DO 260 I=1,6                                                    10300440
ISN 0045      260 PX(I)=0.0                                                       10300450
ISN 0046          DO 261 I=1,6                                                    10300460
ISN 0047          DO 251 J=1,5                                                    10300470
ISN 0048      251 PX(I)=PX(I) + PM(I,J)*X(J)                                      10300480
ISN 0049          DO 262 I=1,6                                                    10300490
ISN 0050      262 VL=VL + X(I)*PX(I)                                              10300500
ISN 0051          IF(VL .GE. VMIN)GO TO 103                                       10300510
ISN 0053          JSW=1                                                           10300520
ISN 0054          CALL AFX(JSW)                                                   10300530
            C     ************************************************************ 10300540
ISN 0055          VDOT =0.0                                                       10300550
```

```
ISN 0056             DO 250 I=1,6                                          10300560
ISN 0057             PF(I)=0.0                                             10300570
ISN 0058         250 QX(I)=0.0                                             10300580
ISN 0059             DO 251 I=1,6                                          10300590
ISN 0060             DO 251 J=1,6                                          10300600
ISN 0061             QX(I)= QX(I) + Q(I,J)*X(J)                            10300610
ISN 0062         251 PF(I)=PF(I) + PM(I,J)*F(J)                            10300620
ISN 0063             DO 252 I=1,6                                          10300630
ISN 0064         252 VDOT = VDOT - X(I)* (QX(I)-2.0 * PF(I))               10300640
ISN 0065             IF(IFLAG.EQ.1)GO TO 520                               10300650
ISN 0067             IF(VDOT .LT. 0)GO TO 103                              10300660
ISN 0069             IF(VMIN .LT. VL)GO TO 103                             10300670
ISN 0071             KFAIL=1                                               10300680
ISN 0072             IFLAG = 1                                             10300690
ISN 0073         500 DO 510 I=1,6                                          10300700
ISN 0074         510 DEL(I)= 0.5*X(I)                                      10300710
ISN 0075         512 DO 515 I=1,6                                          10300720
ISN 0076         515 X(I)= X(I)- DEL(I)                                    10300730
ISN 0077             GO TO 535                                             10300740
ISN 0078         520 DO 530 I=1,6                                          10300750
ISN 0079         530 DEL(I)= DEL(I)*.5                                     10300760
ISN 0080             IF(VDOT.GT.0) GO TO 512                               10300770
ISN 0082             DO 540 I=1,6                                          10300780
ISN 0083         540 X(I)= X(I) + DEL(I)                                   10300790
ISN 0084         535 KAM= KAM + 1                                          10300800
ISN 0085         560 FORMAT(7H VDOT,V/(1X,2E14.7))                         10300810
ISN 0086         550 FORMAT(3H X=/(1X,6E13.4))                             10300820
ISN 0087             IF(KAM .LT.15) GO TO 259                              10300830
ISN 0089             VMIN = VL                                             10300840
ISN 0090             DO 1100 I=1,6                                         10300850
ISN 0091             FZERO(I)= F(I)                                        10300860
ISN 0092        1100 XZERO(I)= X(I)                                        10300870
ISN 0093             NNZERO = NN                                           10300880
ISN 0094             VDOTZ = VDOT                                          10300890
ISN 0095         312 FORMAT(6H DB,DG,4E15.7/6H F(I)=,6E12.4)               10300900
ISN 0096             IFLAG = 0                                             10300910
ISN 0097             KAM =0                                                10300920
ISN 0098             IF(VL .LT. 1.0E-15)GO TO 66
ISN 0100             VOL = SD/((SQRT(VL))**5)                              10300930
ISN 0101             IF(VOL .GT. VSR) GO TO 69                             10300940
ISN 0103             GO TO 100                                             10300950
ISN 0104          66 CONTINUE
ISN 0105             VOL = 1.0E+60
                C                                                          10300960
ISN 0106          69 JSUE = 1                                              10300970
ISN 0107         311 IF(KFAIL .EQ. 0)GO TO 315                             10300980
ISN 0109             WRITE(6,1202)NNZERO
ISN 0110        1202 FORMAT(' NNZERO=', I6)
ISN 0111             GO TO 316                                             10301030
ISN 0112         315 WRITE(6,2503)VMIN                                     10301040
ISN 0113        2503 FORMAT(' NO LINEAR SEARCH'/ ,' VMIN=' E14.7/)         10301050
ISN 0114             VMIN = 10.*VMIN                                       10301060
ISN 0115             GO TO 722                                             10301070
ISN 0116         315 WRITE(6,705)NN                                        10301080
ISN 0117             RETURN                                                10301090
ISN 0118             END                                                   10301100
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

ISN 0002          FUNCTION IMEQD(MID,M,N,A,Y,D,SCALE)                               10800010
            C THIS FORTRAN 4 PROGRAM SOLVES AX = Y BY TRIANGULAR DECOMPOSITION.     10800020
            C THE ARGUMENTS HAVE THE SAME MEANING AS THOSE OF XSIMEQ.               10800030
ISN 0003          REAL*8 DOTPR                                                      10800040
ISN 0004          DOUBLE PRECISION SUM,A,Y,SCALE,D                                  10800050
ISN 0005          DIMENSION A(MID,1),Y(MID,1),SCALE(1)                              10800060
ISN 0006          COMMON /INFO/ SUM,NUMBER,INCR,INCC                                10800070
ISN 0007          INTEGER SPILL                                                     10800080
            C SET OVERFLOW INDICATOR.                                               10800090
            C  TAMPER OVERRIDES STANDARD HANDLING OF SPILL INTERUPTIONS. ITS ARGU-  10800110
            C     MENT IS SET TO ZERO AND THEREAFTER THE VALUES 0,1,2,3 INDICATE NO 10800120
            C     SPILL, UNDERFLOW ONLY, OVERFLOW ONLY, AND BOTH, RESPECTIVELY.     10800130
ISN 0008          INCR = MID                                                        10800140
ISN 0009          INCC = 1                                                          10800150
ISN 0010          DO 120  I = 1,M                                                   10800160
ISN 0011          X = 0.                                                            10800170
ISN 0012          DO 100  J = 1,M                                                   10800180
ISN 0013          GETZ = ABS(A(I,J))                                                10800190
ISN 0014      100 X = AMAX1(X,GETZ)                                                 10800200
ISN 0015          IF (X) 105,490,105                                                10800210
ISN 0016      105 X = POW16(X)                                                      10800220
            C  POW16(X) IS THE POWER OF 16 NEXT LARGER THAN ABS(X)                  10800230
ISN 0017          D = D * X                                                         10800240
ISN 0018          X = 1./ X                                                         10800250
ISN 0019          DO 110  J = 1,M                                                   10800260
ISN 0020      110 A(I,J) = A(I,J) * X                                               10800270
ISN 0021          DO 120  J = 1,N                                                   10800280
ISN 0022      120 Y(I,J) = Y(I,J) * X                                               10800290
ISN 0023          DO 140  J = 1,M                                                   10800300
ISN 0024          X = 0.                                                            10800310
ISN 0025          DO 130  I = 1,M                                                   10800320
ISN 0026          GOTZ = ABS(A(I,J))                                                10800330
ISN 0027      130 X = AMAX1(X,GOTZ)                                                 10800340
ISN 0028          IF (X) 135,490,135                                                10800350
ISN 0029      135 X = POW16(X)                                                      10800360
ISN 0030          D = D * X                                                         10800370
ISN 0031          SCALE(J) = X                                                      10800380
ISN 0032          X = 1./ X                                                         10800390
ISN 0033          DO 140  I = 1,M                                                   10800400
ISN 0034      140 A(I,J) = A(I,J) * X                                               10800410
            C MAJOR LOOP. TRIANGULAR DECOMPOSITION WITH D.P. ACCUM OF INNER PRODUCTS 10800420
ISN 0035          DO 310  K = 1,M                                                   10800430
ISN 0036          K1 = K - 1                                                        10800440
ISN 0037      150 NUMBER = K1                                                       10800450
ISN 0038          X = 0.                                                            10800460
ISN 0039          L = K                                                             10800470
ISN 0040          DO 180  I = K,M                                                   10800480
ISN 0041          SUM = A(I,K)                                                      10800490
ISN 0042          A(I,K) = DOTPR(A(I,1),A(1,K))                                     10800500
            C     + X(NUMBER)*Y(NUMBER) WHERE X AND Y HAVE THE STORAGE INCREMENTS   10800510
            C  DOTPR(X,Y) GIVES THE (D.P. ACCUMULATED) VALUE SUM + X(1)*Y(1) +...   10800520
            C     INCR AND INCC. DOTPR USES COMMON AREA INFO                        10800530
ISN 0043      165 IF (X - ABS(SUM)) 170,180,180                                     10800540
ISN 0044      170 X = ABS(SUM)                                                      10800550
ISN 0045          L=I                                                               10800560
```

```
ISN 0046        180 CONTINUE                                                            10800570
ISN 0047            IF (L - K) 490,220,190                                              10800580
              C ROW INTERCHANGES TO INSURE LARGE PIVOTS                                 10800590
ISN 0048        190 D = - D                                                             10800600
ISN 0049            DO 200 J=1,M                                                        10800610
ISN 0050            X = A(L,J)                                                          10800620
ISN 0051            A(L,J) = A(K,J)                                                     10800630
ISN 0052        200 A(K,J) = X                                                          10800640
ISN 0053            DO 210 J=1,N                                                        10800650
ISN 0054            X = Y(L,J)                                                          10800660
ISN 0055            Y(L,J) = Y(K,J)                                                     10800670
ISN 0056        210 Y(K,J) = X                                                          10800680
ISN 0057        220 X = -A(K,K)                                                         10800690
ISN 0058            IF (M-K) 490,275,230                                                10800700
ISN 0059        230 KD = K + 1                                                          10800710
ISN 0060            DO 240  I = KD,M                                                    10800720
ISN 0061        240 A(I,K) = A(I,K) / X                                                 10800730
ISN 0062        250 DO 270  L = KD,M                                                    10800740
ISN 0063            SUM = A(K,L)                                                        10800750
ISN 0064        270 A(K,L) = DOTPR(A(K,1),A(1,L))                                       10800760
ISN 0065        275 DO 290 L=1,N                                                        10800770
ISN 0066            SUM = Y(K,L)                                                        10800780
ISN 0067        290 Y(K,L) = DOTPR(A(K,1),Y(1,L))                                       10800790
              C UNDULY SMALL PIVOT INDICATES A IS SINGULAR                              10800800
ISN 0068        300 IF (ABS(X) - 2.384186E-7) 490,490,310                               10800810
ISN 0069        310 CONTINUE                                                            10800820
              C BACK SOLUTION                                                           10800830
ISN 0070            I = M                                                               10800840
ISN 0071            DO 345 K=1,M                                                        10800850
ISN 0072            I1 = I + 1                                                          10800860
ISN 0073            NUMBER = M - I                                                      10800870
ISN 0074            DO 340  L = 1,N                                                     10800880
ISN 0075            SUM = -Y(I,L)                                                       10800890
ISN 0076        340 Y(I,L) = -DOTPR(A(I,I+1),Y(I+1,L)) / A(I,I)                         10800900
ISN 0077        345 I=I-1                                                               10800910
ISN 0078            DO 350  I = 1,M                                                     10800920
ISN 0079            X = 1./ SCALE(I)                                                    10800930
ISN 0080            D = D * A(I,I)                                                      10800940
ISN 0081            DO 350  J = 1,N                                                     10800950
ISN 0082        350 A(I,J) = Y(I,J) * X                                                 10800960
ISN 0083            IMEQD = 1                                                           IMEQD
ISN 0084        480 CONTINUE                                                            IMEQD
              C     STNDRD ZEROS THE ARG OF TAMPER AND RESTORES STANDARD SPILL ACTION.  10800990
ISN 0085            RETURN                                                              10801000
ISN 0086        490 D = 0.                                                              10801010
ISN 0087            IMEQD = 3                                                           10801020
ISN 0088            GO TO 480                                                           10801030
ISN 0089            END                                                                 10801040
```

```
LEVEL 2 FEB 67                              OS/360   FORTRAN H                              DATE  69.167/22.00.53

          COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

   ISN 0002          SUBROUTINE  CLOCK                                                    10200010
   ISN 0003          COMMON/GEORGE/INIT
   ISN 0004          COMMON/BLK68/T                                                       CLOCK
   ISN 0005          DATA        I/1/                                                     10200020
   ISN 0006          CALL CLOCKS(NEW)                                                     CLOCK
   ISN 0007          T = FLOAT(NEW -INIT) * .01                                           CLOCK
   ISN 0008          WRITE (6,1) T                                                        10200040
   ISN 0009        1 FORMAT('0',90X,'CLOCK TIME',F16.2,5X,'SECONDS')                      10200050
   ISN 0010          I = 0                                                                10200060
   ISN 0011          RETURN                                                               10200070
   ISN 0012          END                                                                  10200080
```

```
          COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=55,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

ISN 0002              SUBROUTINE DEIGN(PM,B,C)                                               10400010
ISN 0003              IMPLICIT REAL*8 (A-H,O-Z)                                              10400020
                C     THIS SUBROUTINE FINDS THE EIGEN VALUES AND VECTORS OF PM               10400030
ISN 0004              REAL*4  AA,B,C                                                         10400040
ISN 0005              DIMENSION AA(60),B(6),QV(6),U(6),V(6),C(6,6),W(6),PM(6,6)              10400050
ISN 0006              DOUBLE PRECISION P(6)                                                  10400060
ISN 0007              IK=1                                                                   10400070
ISN 0008              DO 1000 K=1,6                                                          10400080
ISN 0009              DO 1000 I=K,6                                                          10400090
ISN 0010              AA(IK) = PM(I,K)                                                       10400100
ISN 0011         1000 IK = IK + 1                                                            10400110
ISN 0012              N=6                                                                    10400120
ISN 0013              M=21                                                                   10400130
ISN 0014              LEAD = 1                                                               10400140
ISN 0015              CALL SYMBIG(AA,N,LEAD,N,M,B,P,QV,U,V,MISS)                             10400150
ISN 0016              IF(MISS)1010,1020,1010                                                 10400160
ISN 0017         1010 WRITE(6,1040)                                                          10400170
ISN 0018         1040 FORMAT(17H ERROR IN BIGSYM )                                           10400180
ISN 0019              GO TO 60                                                               10400190
                C     FIND EIGENVECTORS OF PM(I,J)                                           10400200
ISN 0020         1020 LOW = 1                                                                10400210
ISN 0021              KOUNT = 6                                                              10400220
ISN 0022              MID = 6                                                                10400230
ISN 0023              CALL SECURE(C,LOW,KOUNT,MID,W)                                         10400240
                C                                                                            10400250
ISN 0024           60 RETURN                                                                 10400260
ISN 0025              END                                                                    10400270
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID
ISN 0002          SUBROUTINE AFX(JSW)
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
           C      ***************************************************************
ISN 0004          COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI
ISN 0005          COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM
ISN 0006          COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C
ISN 0007          COMMON /BLK4/ HPHI,HTHT,HPSI
ISN 0008          COMMON/BLK5/A(6,6),Q(6,6),PM(6,6),F(6),DF(6,6)
ISN 0009          COMMON/BLK 11/DB1,DB2,DG1,DG2
ISN 0010           COMMON/BLK77/X(15)
ISN 0011          COMMON/BLK78/X1E,X3E,X5E
           C
           C      EXACT MODEL STATE EQUATIONS
           C
ISN 0012          IF(JSW.GT.0)GO TO 312
           C       ( TRACKERS 3-4    (AMES 1-2)   )
           C
           C      EQUATIONS ARE IN THE FORM    X-DOT = A  X  + F(X)
           C
           C      WHERE X IS A SIX COMPONENT COLUMN VECTOR AS IS F(X)
           C
           C      AND A IS 6X6 MATRIX
           C
           C      X-VECTOR IS(PHI,VPHI,THT,VTHT,PSI,VPSI)
           C
           C      ***************************************************************
           C
ISN 0014          WRITE(6,1070)PHI,VPHI,THT,VTHT,PSI,VPSI,TM,T1,T2,
                 1XKM,XKC,A11,A13,D12,GAM1C,GAM2C,BET1C,BET2C,AITREN,HTHT,HPHI,HPSI
ISN 0015     1070 FORMAT(1H /10X,14H INITIAL STATE / 6E13.6 /10X,13H INPUT CONSTS/
                 110X, 29H TM,T1,T2,XKM,XKC,A11,A13,D12 / 8E14.7 / 10X, 30H GAM1C,GA
                 2M2C,BET1C,BET2C,(RAD) / 4E14.7 / 10X,10H INERTIA = E14.7 ,/10X,
                 317H HTHT,HPHI,HPSI = 3E14.7//)
           C      ***************************************************************
ISN 0016          PI = 3.1415926
ISN 0017          DTR  = PI/180.0
ISN 0018          RTD  = 180.0/PI
           C      ***************************************************************
           C
ISN 0019          KK=0
ISN 0020          SB2C  = SIN(BET2C)
ISN 0021          CB2C  = COS(BET2C)
ISN 0022          TB2C = SB2C/CB2C
ISN 0023          SG1C  = SIN(GAM1C)
ISN 0024          CG1C  = COS(GAM1C)
ISN 0025          SG2C  = SIN(GAM2C)
ISN 0026          CG2C  = COS(GAM2C)
ISN 0027          SB1C  = SIN(BET1C)
ISN 0028          CB1C  = COS(BET1C)
ISN 0029          TB1C  = SB1C/CB1C
ISN 0030           SGAM1C = SG1C
ISN 0031           SGAM2C = SG2C
ISN 0032           CGAM1C = CG1C
ISN 0033           CGAM2C = CG2C
           C
```

```
                C      ****************************************************************** 10500560
ISN 0034               DIF = GAMIC - GAM2C                                                10500570
ISN 0035               SDIF = SIN(DIF)                                                    10500580
                C                                                                         10500590
                C      ****************************************************************** 10500600
                C                                                                         10500610
                C      CALCULATION OF THE A-MATRIX                                        10500620
                C                                                                         10500630
ISN 0036               DO 10 I=1,6                                                        10500640
ISN 0037               DO 10 J=1,6                                                        10500650
ISN 0038            10 A(I,J)= 0.0                                                        10500660
ISN 0039               A(1,2)= -XKM*XKC/AITREN                                            10500670
ISN 0040               A(2,1)= 1./TM                                                      10500680
ISN 0041               A(2,2)=       -(1.+ 4.5*XKM*XKC/AITREN)/TM                         10500690
ISN 0042               A(2,3)= -TB1C*CG1C/TM                                              10500700
ISN 0043               A(2,4)=(4.5*XKM*XKC/(TM*AITREN))*TB1C*CG1C                         10500710
ISN 0044               A(2,5)= TB1C*SG1C/TM                                               10500720
ISN 0045               A(2,6)=  -4.5*XKM*XKC*TB1C*SG1C/(TM*AITREN)                        10500730
ISN 0046               A(3,4)=  -XKM*XKC/AITREN                                           10500740
ISN 0047               A(5,6)= -XKM*XKC/AITREN                                            10500750
ISN 0048               A(6,5)= D12*SDIF/TM                                                10500760
ISN 0049               A(6,6)=-(1.+ 4.5*XKC*XKM*D12*SDIF/AITREN)/TM                       10500770
ISN 0050               A(4,3)= A(6,5)                                                     10500780
ISN 0051               A(4,4)= A(6,6)                                                     10500790
                C                                                                         10500800
ISN 0052               WRITE(6,5000)((A(I,J),J=1,6),I=1,6)                                10500810
ISN 0053          5000 FORMAT(1H / 1X,11H   A-MATRIX/(1X,6E13.6))                         10500820
                C                                                                         10500830
                C      ****************************************************************** 10500840
                C                                                                         10500850
                C      CALCULATION  OF DB1,DB2,DG1,DG2                                    10500860
                C                                                                         10500870
                C      NEEDS ANGLES FROM STATE VECTOR AND COMMAND ANGLES AS INPUT         10500880
                C                                                                         10500890
                C                                                                         10500900
                C                                                                         10500910
ISN 0054           312 PHI= X(1) + X1E                                                    10500920
ISN 0055               THT= X(3)+X3E                                                      10500930
ISN 0056               PSI= X(5)+X5E                                                      10500940
ISN 0057               SPHI = SIN(PHI)                                                    10500950
ISN 0058               CPHI = COS(PHI)                                                    10500960
ISN 0059               STHT = SIN(THT)                                                    10500970
ISN 0060               CTHT = COS(THT)                                                    10500980
ISN 0061               SPSI = SIN(PSI)                                                    10500990
ISN 0062               CPSI = COS(PSI)                                                    10501000
ISN 0063               TTHT = STHT/CTHT                                                   10501010
ISN 0064               PUP = 1.0/(XKC*TM)                                                 10501020
                C                                                                         10501030
ISN 0065               GR1= CPSI*CTHT*SB1C                                                10501040
ISN 0066               GR2= SPSI*CTHT*CG1C*CB1C                                           10501050
ISN 0067               GR3= STHT*SG1C*CB1C                                                10501060
ISN 0068               GR4= GR1 + GR2 + GR3                                               10501070
ISN 0069               DB1= ARSIN(GR4) - BET1C                                            10501080
                C                                                                         10501090
ISN 0070               DG1 =                                 ATAN(   (-SB1C*(SPSI*        10501100
                     1 SPHI + CPSI*STHT*CPHI) +CG1C*CB1C *(CPSI*SPHI -SPSI*STHT*CPHI)     10501110
```

```
                 2 +SG1C*CB1C*CTHT*CPHI ) /  (SB1C *(-SPSI*CPHI + CPSI*STHT*SPHI)
                 3 +CG1C*CB1C*(CPSI*CPHI + SPSI*STHT*SPHI) -CB1C*SG1C*CTHT*SPHI) )
                 4                                              - GAM1C
           C
ISN 0071          GL1= CPSI*CTHT*SB2C-SPSI*CTHT*CG2C*CB2C-STHT*SG2C*CB2C
ISN 0072          DB2= ARSIN(GL1) - BET2C
           C
ISN 0073          DG2 = -GAM2C + ATAN(  (SB2C*(SPSI*SPHI +CPSI*STHT*CPHI) +CG2C*
                 1 CB2C*(CPSI*SPHI-SPSI*STHT*CPHI) +SG2C*CB2C*CTHT*CPHI )/ (SB2C*
                 2 (SPSI*CPHI-CPSI*STHT*SPHI)+CG2C*CB2C*(CPSI*CPHI+SPSI*STHT*SPHI)
                 3 -CTHT*SPHI*SG2C*CB2C ) )
ISN 0074          IF(BET1C.EQ. 0.0) GO TO 850
ISN 0076          APE1= ABS(DB1/BET1C)
ISN 0077          IF(APE1 .LT. 1.D-10)DB1= SG1C*THT + CG1C*PSI
ISN 0079      850 IF(GAM1C .EQ. 0.0) GO TO 851
ISN 0081          APE2= ABS(DG1/GAM1C)
ISN 0082          IF(APE2 .LT. 1.D-10)DG1= PHI - TB1C*CG1C*THT + TB1C*SG1C*PSI
ISN 0084      851 IF(BET2C .EQ. 0.0) GO TO 852
ISN 0086          APE3= ABS(DB2/BET2C)
ISN 0087          IF(APE3 .LT. 1.D-10)DB2= -SG2C*THT - CG2C*PSI
ISN 0089      852 IF(GAM2C .EQ. 0.0) GO TO 853
ISN 0091          APE4= ABS(DG2/GAM2C)
ISN 0092          IF(APE4 .LT. 1.D-10)DG2= PHI + TB2C*CG2C*THT - TB2C*SG2C*PSI
           C
           C      *****************************************************************
           C
           C      CALCULATION OF NONLINEAR F(X)
           C
           C
ISN 0094      853 SUM5=(COS(DG2+GAM2C))*DB1 + (COS(DG1+GAM1C))*DB2
ISN 0095          SUM6=(SIN(DG2+GAM2C))*DB1 + (SIN(DG1+GAM1C))*DB2
ISN 0096          EPHI= DG1
ISN 0097          ETHT= D12*SUM5
ISN 0098          EPSI=-D12*SUM6
ISN 0099          PJ1=(XKM*XKC/AITREN)*(-SIN(DG1+GAM1C)*X(4)-COS(DG1+GAM1C)*X(6))
ISN 0100          PJ2=(XKM*XKC/AITREN)*(SIN(DG2+GAM2C)*X(4)+COS(DG2+GAM2C)*X(6))
ISN 0101          PJ3=(XKM*XKC/AITREN)*(-X(2)-TAN(DB2+BET2C)*(COS(DG2+GAM2C)*X(4)
                 1      - SIN(DG2+GAM2C)*X(6)))
ISN 0102          PJ4=(XKM*XKC/AITREN)*(-X(2)+TAN(DB1+BET1C)*(COS(DG1+GAM1C)*X(4)
                 1      -SIN(DG1+GAM1C)*X(6)))
ISN 0103          EPHID= PJ4
ISN 0104          ETHTD= D12*(COS(DG2+GAM2C)*PJ1+COS(DG1+GAM1C)*PJ2-SIN(DG2+GAM2C)
                 1         *DB1*PJ3-SIN(DG1+GAM1C)*DB2*PJ4)
ISN 0105          EPSID=-D12*(SIN(DG2+GAM2C)*PJ1+SIN(DG1+GAM1C)*PJ2+DB1*COS(DG2+
                 1       GAM2C)*PJ3 + DB2*COS(DG1+GAM1C)*PJ4)
ISN 0106          F(1)= -XKM*XKC/AITREN*(TTHT*SPHI*X(4)+TTHT*CPHI*X(6))
           C
ISN 0107          ARGF2 =XKC*(EPHI + 4.5*EPHID)
ISN 0108          IF(ABS(ARGF2).LE.F2LIM)F2= ARGF2
ISN 0110          IF(    (ARGF2).GT.F2LIM)F2= F2LIM
ISN 0112          IF( ARGF2.LT.-F2LIM)F2=-F2LIM
ISN 0114          F(2)= 1./(XKC*TM)*(F2 -AITREN*HPHI/XKM)-1./TM*(X(1)-(TB1C*CG1C)
                 1      *X(3)+TB1C*SG1C*X(5) +AITREN*HPHI/(XKC*XKM)- 4.5*XKM*XKC/
                 1      AITREN*(X(2)-TB1C*CG1C*X(4) + TB1C*SG1C*X(6)))
           C
ISN 0115          F(3)= -XKM*XKC/AITREN*((CPHI-1.)*X(4)-SPHI*X(6))
```

```
                C
ISN 0116               ARGF4= XKC*(ETHT+4.5*ETHTD)
ISN 0117               IF(ABS(ARGF4).LE.F2LIM)F2= ARGF4
ISN 0119               IF(    (ARGF4).GT.F2LIM)F2= F2LIM
ISN 0121               IF( ARGF4.LT.-F2LIM)F2=-F2LIM
ISN 0123               F(4)= 1./(XKC*TM)*(F2-AITREN*HTHT/XKM)
                     1  -1./TM*(D12*(SDIF*X(3)+AITREN*HTHT/(XKC*XKM)- 4.5*XKM*XKC/
                     1       AITREN*(CG2C*SG1C*X(4)-CG1C*SG2C*X(4))))
                C
ISN 0124               F(5)= -XKM*XKC/AITREN*(SPHI*X(4)/CTHT+(CPHI/CTHT-1.)*X(6))
ISN 0125               ARGF6= XKC*(EPSI + 4.5*EPSID)
ISN 0126               IF(ABS(ARGF6).LE.F2LIM)F2=ARGF6
ISN 0128               IF(    (ARGF6).GT.F2LIM)F2=F2LIM
ISN 0130               IF( ARGF6.LT.-F2LIM)F2=-F2LIM
ISN 0132               F(6)=1./(XKC*TM)*(F2-AITREN*HPSI/XKM)+1./TM*D12*(-SDIF*X(5)
                     1      +AITREN*HPSI/(XKC*XKM)-4.5*XKM*XKC/AITREN*X(6)*
                     1      (SG2C*CG1C-SG1C*CG2C))
                C      ****************************************************************
ISN 0133               KK=KK+1
ISN 0134               IF(KK.EQ.2)GO TO 700
ISN 0136               GO TO 702
ISN 0137           700 WRITE(6,5002)DB1,DB2,DG1,DG2
ISN 0138          5002 FORMAT(6H DB,DG/1X,4E13.6)
ISN 0139           702 CONTINUE
                C
ISN 0140               IF(JSW.GT.0)GO TO 602
ISN 0142               WRITE(6,5001)(F(I),I=1,6)
ISN 0143          5001 FORMAT(1H /1X,12H F(X)-VECTOR/1X,6E13.6)
ISN 0144           602 CONTINUE
                C
                C
                C      ****************************************************************
ISN 0145               RETURN
ISN 0146               END
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID
ISN 0002          SUBROUTINE QGEN(THETA,PHIV,XLAM)                                    10600010
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)                                           10600020
            C                                                                         10600030
            C     GENERATION OF POSITIVE DEFINITE Q MATRIX                            10600040
ISN 0004          DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP                   10600050
ISN 0005          COMMON/BLK5/A(6,6),Q(6,6),PM(6,6),F(6),DF(6,6)                      10600060
ISN 0006          N=6                                                                 10600070
            C                                                                         10600080
ISN 0007          DIMENSION THETA(28),PHIV(8),XLAM(9),ZTHETA(28),TTHETA(28),          10600090
                1             BA(20,20),SS(20,20,20),CC(20,20),Z(20,20,20),           10600100
                2            BM(20,20),SM(20,20),AAR(20,20),G(20,20),QQ(20,20)        10600110
            C          )9,1=K,)K(MALX(,)8,1=J,)J(VIHP(,)82,1=I,)I(ATEHT()3601,5(DA   10600120
ISN 0008     1063 FORMAT(6E12.4)                                                      10600130
ISN 0009          EXTERNAL DMOD                                                       QGEN
ISN 0010          PI = 3.1415926                                                      10600140
ISN 0011          PI2 = PI/2.                                                         10600150
ISN 0012          NN=(N-1)*(N-2)/2                                                    10600160
ISN 0013          DO 6I=1,NN                                                          10600170
ISN 0014          BAD=THETA(I)                                                        10600180
ISN 0015        6 TTHETA(I)= DMOD(BAD,PI2)                                            QGEN
            C     WE HAVE NOW INDEXED THETA.                                          10600200
            C     NOW WANT CONTINUED PRODUCT OF SS(I,J,L) FOR L=K+1,N .               10600210
            C     FOR EACH K=1,N-1 OBTAIN Z(K,I,J).                                   10600220
ISN 0016           NNI = N-1                                                          10600230
ISN 0017       69 DO 20 K=1,NNI                                                       10600240
            C                                                                         10600250
ISN 0018          DO 8 I=1,N                                                          10600260
ISN 0019          DO 8 J=1,N                                                          10600270
ISN 0020        8 BA(I,J)=0.0                                                         10600280
ISN 0021          DO 99 I=1,N                                                         10600290
ISN 0022       99 BA(I,I)=1.0                                                         10600300
            C                                                                         10600310
ISN 0023          KK=K+1                                                              10600320
ISN 0024          DO 10 L=KK,N                                                        10600330
            C                                                                         10600340
ISN 0025          DO 15 I=1,N                                                         10600350
ISN 0026          DO 15 J=1,N                                                         10600360
ISN 0027       15 SS(I,J,L)=0.0                                                       10600370
ISN 0028          DO 98 I=1,N                                                         10600380
ISN 0029       98 SS(I,I,L)=1.0                                                       10600390
            C     WE DEVELOP SS(I,J,L) AS FUNCTION THETA(L,K,N) FOR L L.T. N          10600400
            C     AND SS(I,J,L) FUNCTION OF PHIV(K) FOR L=N                           10600410
ISN 0030          IF(L-N)25,23,23                                                     10600420
ISN 0031       25 M=((2*N -K-2)*(K-1)/2)+N-L                                          10600430
ISN 0032          SS(K,K,L)=COS(TTHETA(M))                                            10600440
ISN 0033          SS(L,L,L)=COS(TTHETA(M))                                            10600450
ISN 0034          SS(K,L,L)=-SIN(TTHETA(M))                                           10600460
ISN 0035          SS(L,K,L)=SIN(TTHETA(M))                                            10600470
ISN 0036          GO TO 35                                                            10600480
ISN 0037       23 SS(K,K,L)=COS(PHIV(K))                                              10600490
ISN 0038          SS(L,L,L)=COS(PHIV(K))                                              10600500
ISN 0039          SS(K,L,L)=-SIN(PHIV(K))                                             10600510
ISN 0040          SS(L,K,L)=SIN(PHIV(K))                                              10600520
            C                                                                         10600530
ISN 0041       35 DO 70 I=1,N                                                         10600540
```

```
ISN 0042          DO 70 J=1,N                                                10600550
ISN 0043       70 CC(I,J)=0.0                                                10600560
            C                                                                10600570
ISN 0044          DO 50 M=1,N                                                10600580
ISN 0045          DO 50 J=1,N                                                10600590
ISN 0046          DO 50 I=1,N                                                10600600
ISN 0047       50 CC(M,J)=BA(M,I)*SS(I,J,L) +CC(M,J)                         10600610
ISN 0048          DO110 I=1,N                                                10600620
ISN 0049          DO110 J=1,N                                                10600630
ISN 0050      110 BA(I,J)=CC(I,J)                                            10600640
ISN 0051       10 CONTINUE                                                   10600650
ISN 0052          DO 20 I=1,N                                                10600660
ISN 0053          DO 20 J=1,N                                                10600670
ISN 0054       20 Z(K,I,J)=BA(I,J)                                           10600680
            C                                                                10600690
ISN 0055          DO 7 I=1,N                                                 10600700
ISN 0056          DO 7 J=1,N                                                 10600710
ISN 0057        7 BM(I,J)=0.0                                                10600720
ISN 0058          DO 16 I=1,N                                                10600730
ISN 0059       16 BM(I,I)=1.0                                                10600740
            C                                                                10600750
ISN 0060          DO 40 K=1,NNI                                              10600760
            C                                                                10600770
ISN 0061          DO 75 I=1,N                                                10600780
ISN 0062          DO 75 J=1,N                                                10600790
ISN 0063       75 SM(I,J)=0.0                                                10600800
            C                                                                10600810
ISN 0064          DO 55 M=1,N                                                10600820
ISN 0065          DO 55 J=1,N                                                10600830
ISN 0066          DO 55 I=1,N                                                10600840
ISN 0067       55 SM(M,J)=Z(K,M,I)*BM(I,J)+SM(M,J)                           10600850
ISN 0068          DO 40 I=1,N                                                10600860
ISN 0069          DO 40 J=1,N                                                10600870
ISN 0070       40 BM(I,J)=SM(I,J)                                            10600880
            C                                                                10600890
            C     BM(I,J) IS CONTINUED PRODUCT OF Z(K,I,J) FROM K=1 TO N-1   10600900
            C                                                                10600910
ISN 0071          IF(PP)41,41,19                                             10600920
ISN 0072       19 CONTINUE                                                   10600930
ISN 0073       18 FORMAT(8H BM(I,J)/(6E15.7))                                10600940
ISN 0074       41 DO 78 I=1,N                                                10600950
ISN 0075          DO 78 J=1,N                                                10600960
ISN 0076       78 AAR(I,J)=BM(J,I)                                           10600970
            C                                                                10600980
            C     AAR(I,J) IS TRANSPOSE BM(I,J)                              10600990
            C                                                                10601000
ISN 0077          DO 82 I=1,N                                                10601010
ISN 0078          DO 82 J=1,N                                                10601020
ISN 0079       82 G(I,J)=0.0                                                 10601030
ISN 0080          DO 85 I=1,N                                                10601040
ISN 0081       85 G(I,I)=XLAM(I)                                             10601050
            C     G(I,J) IS THE LAMDA MATRIX                                 10601060
            C                                                                10601070
ISN 0082          DO 86 I=1,N                                                10601080
ISN 0083          DO 86 J=1,N                                                10601090
ISN 0084       86 QQ(I,J)=0.0                                                10601100
```

```
ISN 0085          DO 88 I=1,N
ISN 0086          DO 88 J=1,N
ISN 0087          DO 88 M=1,N
ISN 0088       88 QQ(I,J)=G(I,M)*BM(M,J)+QQ(I,J)
            C
            C     QQ(I,J)=LAMDA MATRIX *BM(I,J)
            C
ISN 0089          DO 90 I=1,N
ISN 0090          DO 90 J=1,N
ISN 0091       90 Q(I,J)=0.0
ISN 0092          DO 95 I=1,N
ISN 0093          DO 95 J=1,N
ISN 0094          DO 95 M=1,N
ISN 0095       95 Q(I,J)=AAR(I,M)*QQ(M,J)+Q(I,J)
ISN 0096          RETURN
ISN 0097          END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002              SUBROUTINE PEAIQ(KEEP)
ISN 0003              IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004              COMMON/BLK5/A(6,6),Q(6,6),PM(6,6),F(6),DF(6,6)
                C
ISN 0005              N=6
                C
                C
ISN 0006              DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP
ISN 0007              DIMENSION AAMOD(37,37),P(36),QV(36),E(6,6),AM(6,6)
                     1 ,ISTEP(37),ATP(6,6),PA(6,6),QP(6,6)
                C
                C     N =  DIMENSION OF A - MATRIX (INPUT ON CARD NO.2 )
                C     A =  INPUT MATRIX
                C
                C
                C
                C
                C     INITIALIZE  PM , QP, ATP , PA , AM , E
                C
                C
                C
ISN 0008              NN = N*N
ISN 0009              DO 37 I=1,N
ISN 0010              DO 37 J=1,N
ISN 0011              PM(I,J) = 0.0
ISN 0012              QP(I,J) = 0.0
ISN 0013              ATP(I,J)= 0.0
ISN 0014              PA(I,J) = 0.0
ISN 0015              AM(I,J) = 0.0
ISN 0016           37 E(I,J)  = 0.0
                C
ISN 0017         2963 FORMAT(1H /(1X,9E14.7) )
                C
                C
                C     MAKING Q PERFECTLY SYMMETRIC
ISN 0018               NK = 2
ISN 0019           42 DO 76 JJ=1,5
ISN 0020               Q(NK,JJ) = Q(JJ,NK)
ISN 0021              NOOK = NK-1
ISN 0022               IF(NOOK .EQ.JJ) GO TO 77
ISN 0024           76  CONTINUE
ISN 0025           77 IF(NK.EQ.6)GO TO 87
ISN 0027               NK = NK + 1
ISN 0028               GO TO 42
ISN 0029           87  CONTINUE
                C
                C     SETTING Q-MATRIX TO Q-VECTOR
                C
ISN 0030              IA=1
ISN 0031              DO 62 I=1,N
ISN 0032              DO 62 J=1,N
ISN 0033              QV(IA) = Q(I,J)
ISN 0034           62 IA=IA+1
                C
                C
```

```
          C
          C     THIS SECTION CALCULATES THE AAMOD MATRIX
          C     WHICH IS GIVEN BY
          C           XXXX                                  XXXX
          C           X                                        X
          C           X AT+ A11.I     A21.I       A31.I        X
          C           X                                        X
          C           X  A12.I       AT+ A22.I    A32.I        X
          C           X                                        X
          C           X  A13.I        A23.I      AT+ A33.I     X
          C           X                                        X
          C           XXXX                                  XXXX
          C
          C
          C     INITIALIZATION OF AAMOD
          C
          C
ISN 0035        IF(KEEP .GT. 0)GO TO 66
ISN 0037        DO 10  L = 1,NN
ISN 0038        DO 10  LA= 1,NN
ISN 0039     10 AAMOD(L,LA) = 0.0
          C
          C     DEFINE UNIT MATRIX OF ORDER  N ,CALL IT E
          C
ISN 0040        DO 30 MAA = 1,N
ISN 0041     30 E(MAA,MAA) = 1.0
          C
          C
          C
          C
          C     THIS SECTION CALCULATES THE SECTION OF THE  AMOD
          C     MATRIX WHICH IS COMPRISED OF  A(J,I) * E
ISN 0042        IP =  -N
ISN 0043        DO 50  MR=1,N
ISN 0044        IP = IP + N
ISN 0045        IPP = -N
ISN 0046        DO 50  JR =1,N
ISN 0047        IPP = IPP +N
ISN 0048        DO 40  K= 1,N
ISN 0049        DO 40 KA= 1,N
ISN 0050        AM(K,KA) = A(JR,MR) * E(K,KA)
ISN 0051        KPIP = K+IP
ISN 0052        KAP  = KA+IPP
ISN 0053     40 AAMOD(KPIP,KAP)=AM(K,KA)
ISN 0054     50 CONTINUE
          C
          C
          C
          C
          C
          C     THIS SECTION ADDS THE A MATRIX TO THE DIAGONAL NXN
          C     ELEMENTS OF AAMOD
          C
          C
ISN 0055        DO 60   K = 1,N
ISN 0056        IP  = (K-1)* N
```

```
ISN 0057          DO 55  LT = 1,N
ISN 0058          DO 55  LM = 1,N
ISN 0059          ILT = IP+LT
ISN 0060          IPM = IP+LM
ISN 0061       55 AAMOD(ILT,IPM)= AAMOD(ILT,IPM)+ A(LM,LT)
ISN 0062       60 CONTINUE
            C
            C     THAT FINISHES THE CALCULATION OF AMOD ,NOW WE MUST
            C     PRINT IT OUT BECAUSE SREVNI WIPES  OUT AAMOD
            C
            C                                  )NN,1=I,)NN,1=J,)J,I(DOMAA(()3692,6(ETI
            C     NOW FOR A-INVERSE
            C
ISN 0063          IDEM=NN+1
            C
ISN 0064          CALL MINVD(AAMOD,IDEM,NN,ISTEP,IERR)
            C
            C
            C     NOW AAMOD INVERSE HAS REPLACED AAMOD
            C
            C                                                          )192,6(ETI
ISN 0065      291 FORMAT(1H / 1X,22HAAMOD-INVERSE  MATRIX   , //)
            C                                                     NN,1=I 371
            C                                         )NN,1=J,)J,I(DOMAA()3692,6(ETI
ISN 0066       66 DO 70 IB =1,NN
ISN 0067          P(IB) =0.0
ISN 0068          DO 70  IC = 1,NN
ISN 0069       70 P(IB) =  P(IB) -AAMOD(IB,IC) * QV(IC)
            C     SET P-VECTOR TC P-MATRIX TO GET Q-PRIME FROM-ATP-PA =QP
ISN 0070          K=1
ISN 0071          DO25 I=1,N
ISN 0072          DO25 J=1,N
ISN 0073          PM(I,J) = P(K)
ISN 0074       25 K=K+1
            C     CALCULATION OF ATP (A TRANSPOSE P )
            C     CALCULATION OF PA  (P-MATRIX X A )
ISN 0075          DO 26 I=1,N
ISN 0076          DO 26 J=1,N
ISN 0077          DO 26 K=1,N
ISN 0078          ATP(I,J) = ATP(I,J) + A(K,I) *PM(K,J)
ISN 0079       26 PA(I,J)  = PA(I,J)  +PM(I,K) * A(K,J)
ISN 0080          DO 27 I=1,N
ISN 0081          DO 27 J=1,N
ISN 0082       27 QP(I,J) = -ATP(I,J) - PA(I,J)
ISN 0083          RETURN
ISN 0084          END
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

ISN 0002          FUNCTION SCAPR(X,Y,SUM,L,IX,IY)                                     10900010
ISN 0003          REAL*8 X(IX,1),Y(IY,1)                                              10900020
ISN 0004          REAL*8 SUM,SCAPR                                                    10900030
ISN 0005          IF (L .EQ. 0) GO TO 110                                             10900040
ISN 0007          DO 100  J = 1,L                                                     10900050
ISN 0008      100 SUM = SUM + X(1,J)*Y(1,J)                                           10900060
ISN 0009      110 SCAPR = SUM                                                         10900070
ISN 0010          RETURN                                                              10900080
ISN 0011          END                                                                 10900090
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          FUNCTION DOTPR(X,Y)                                        11000010
ISN 0003      IMPLICIT REAL*8 (A-H,O-Z)                                      11000020
ISN 0004      REAL*8 X(1),Y(1)                                               11000030
ISN 0005      COMMON /INFO/ SUM,NUMBER,INCR,INCC                             11000040
ISN 0006      DOTPR  = SCAPR(X,Y,SUM,NUMBER,INCR,INCC)                       11000050
ISN 0007      RETURN                                                         11000060
ISN 0008       END                                                           11000070
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE MINVD(A,IDIM,N,ISTEP,IERR)                         11100010
           C                MATRIX INVERSION          DOUBLE PRECISION          11100020
ISN 0003          DOUBLE PRECISION A,ABSAI,ABSAL,TEMP,FAC,ABSA,DABS            11100030
ISN 0004          DIMENSION A(IDIM,1),ISTEP(1)                                  11100040
ISN 0005          K=1                                                           11100050
ISN 0006          IERR=0                                                        11100060
ISN 0007          NP1=N+1                                                       11100070
ISN 0008       30 LL=1                                                          11100080
ISN 0009          DO 35 J=1,N                                                   11100090
ISN 0010       35 A(J,NP1)=A(J,1)                                               11100100
ISN 0011       40 I=1                                                           11100110
ISN 0012          L=2                                                           11100120
ISN 0013       45 ABSAI=DABS(A(I,1))                                            11100130
ISN 0014          ABSAL=DABS(A(L,1))                                            11100140
ISN 0015          IF(ABSAI-ABSAL)50,55,55                                       11100150
ISN 0016       50 I=L                                                           11100160
ISN 0017       55 IF(L-N)60,56,56                                               11100170
ISN 0018       56 IF(A(I,1))65,85,65                                            11100180
ISN 0019       60 L=L+1                                                         11100190
ISN 0020          GO TO 45                                                      11100200
ISN 0021       65 IF(K-1)70,90,70                                               11100210
ISN 0022       70 M=1                                                           11100220
ISN 0023       75 IF(I-ISTEP(M))80,84,80                                        11100230
ISN 0024       80 IF(M-K+1)81,82,82                                             11100240
ISN 0025       81 M=M+1                                                         11100250
ISN 0026          GO TO 75                                                      11100260
ISN 0027       82 DO 83 J=1,N                                                   11100270
ISN 0028       83 A(J,1)=A(J,NP1)                                               11100280
ISN 0029          GO TO 90                                                      11100290
ISN 0030       84 IF(LL-N)86,85,85                                              11100300
ISN 0031       85 IERR=1                                                        11100310
ISN 0032          GO TO 610                                                     11100320
ISN 0033       86 LL=LL+1                                                       11100330
ISN 0034          A(I,1)=0.D0                                                   11100340
ISN 0035          GO TO 40                                                      11100350
ISN 0036       90 ISTEP(K)=I                                                    11100360
ISN 0037          J=1                                                           11100370
ISN 0038      100 IF(J-I)110,120,110                                            11100380
ISN 0039      110 A(J,NP1)=0.D0                                                 11100390
ISN 0040          GO TO 130                                                     11100400
ISN 0041      120 A(J,NP1)=1.D0                                                 11100410
ISN 0042      130 IF(J-N)140,150,150                                            11100420
ISN 0043      140 J=J+1                                                         11100430
ISN 0044          GO TO 100                                                     11100440
ISN 0045      150 J=1                                                           11100450
ISN 0046          TEMP=A(I,1)                                                   11100460
ISN 0047      160 A(I,J)=A(I,J)/TEMP                                            11100470
ISN 0048          IF(J-NP1)170,180,180                                          11100480
ISN 0049      170 J=J+1                                                         11100490
ISN 0050          GO TO 160                                                     11100500
ISN 0051      180 J=1                                                           11100510
ISN 0052      190 IF(J-I)200,290,200                                            11100520
ISN 0053      200 IF(A(J,1)-1.D0)230,210,230                                    11100530
ISN 0054      210 DO 220 M=1,NP1                                                11100540
ISN 0055          A(J,M)=A(J,M)-A(I,M)                                          11100550
```

```
220 CONTINUE
    GO TO 290
230 IF(A(J,1)+1.D0)260,240,260
240 DO 250 M=1,NP1
    A(J,M)=A(J,M)+A(I,M)
250 CONTINUE
    GO TO 290
260 IF(A(J,1))270,290,270
270 FAC=A(J,1)
    DO 280 M=1,NP1
280 A(J,M)=A(J,M)-A(I,M)*FAC
290 IF(J-N)300,340,340
300 J=J+1
    GO TO 190
340 DO 350 J=1,N
    DO 350 M=1,N
    MP1=M+1
350 A(J,M)=A(J,MP1)
    IF(K-N)360,390,390
360 K=K+1
    GO TO 30
390 DO 400 J=1,N
400 A(NP1,J)=ISTEP(J)
    M=1
410 I=ISTEP(M)
    IF(I-M)420,470,420
420 DO 430 J=1,N
    TEMP=A(M,J)
    A(M,J)=A(I,J)
430 A(I,J)=TEMP
    J=M
440 IF(M-ISTEP(J))450,460,450
450 J=J+1
    GO TO 440
460 ISTEP(J)=I
470 IF(M-N)480,490,490
480 M=M+1
    GO TO 410
490 DO 500 J=1,N
500 ISTEP(J)=A(NP1,J)
530 M=1
540 I=ISTEP(M)
    IF(I-M)550,570,550
550 DO 560 J=1,N
    TEMP=A(J,I)
    A(J,I)=A(J,M)
560 A(J,M)=TEMP
    J=ISTEP(M)
    ISTEP(M)=ISTEP(J)
    ISTEP(J)=J
    GO TO 540
570 IF(M-N)580,610,610
580 M=M+1
    GO TO 540
610 RETURN
    END
```

```
LEVEL 2 FEB 67                                  OS/360   FORTRAN H                              DATE   69.163/11.56.14

        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

  ISN 0002          SUBROUTINE BOXNO (R1,R2)                                          11200010
  ISN 0003          T1 =  SQRT(-2.0 *ALOG(RDM (DUM)))                                 11200020
  ISN 0004          T2 =  6.2831853 * RDM (DUM)                                       11200030
  ISN 0005          R1 =  T1 * COS(T2)                                                11200040
  ISN 0006          R2 =  T1 * SIN(T2)                                                11200050
  ISN 0007          RETURN                                                            11200060
  ISN 0008          END                                                               11200070
```

```
LEVEL 2 FEB 67                    OS/360   FORTRAN H                    DATE  69.163/11.56.17

        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID
 ISN 0002          FUNCTION RDM(IX)                                          11300010
 ISN 0003          DATA IY/5757403/                                          11300020
 ISN 0004          IY=IY*65539                                               11300030
 ISN 0005          IF(IY.GE.0) GO TO 6                                       11300040
 ISN 0007          IY=IY+2147483647+1                                        11300050
                                                                             11300060
 ISN 0008        6 YFL=IY                                                    11300070
 ISN 0009          RDM=YFL*.4656613E-9                                       11300080
 ISN 0010          RETURN                                                    11300090
 ISN 0011          ENTRY RDMIN( IX )                                         11300100
 ISN 0012          IY=IX                                                     11300110
 ISN 0013          RDMIN=IX                                                  11300120
 ISN 0014          RETURN                                                    11300130
 ISN 0015          ENTRY RDMOUT ( IX )                                       11300140
 ISN 0016          IX=IY                                                     11300150
 ISN 0017          RDMOUT=IX                                                 11300160
 ISN 0018          RETURN                                                    11300170
 ISN 0019          END
```

```
LEVEL 2 FEB 67                     OS/360  FORTRAN H                     DATE  69.163/11.56.20

        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

  ISN 0002          SUBROUTINE SETCLK
  ISN 0003          COMMON/GEORGE/INIT
  ISN 0004          CALL CLOCKS(INIT)
  ISN 0005          RETURN
  ISN 0006          END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          FUNCTION DARSIN(X)
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004          DARSIN=ATAN(X/SQRT(1.0-X**2))
ISN 0005         RETURN
ISN 0006          END
```

```
         COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID
  ISN 0002            SUBROUTINE SYMBIG(A,NO,INDEX1,INDEX2,ASIZE,B,P,Q,U,V,MISS)
                C HOUSEHOLDER TRI-DIAGONALIZATION ROUTINE FOR REAL SYMMETRIC MATRICES.
                C FOLLOWED BY STURM-SEQUENCE COMPUTATION OF THE EIGENVALUES. PROGRAM
                C SHOULD WORK FOR ORDERS UP TO 1000 AT LEAST, INCLUDING 1 AND 2, BUT IT
                C IS AIMED PRIMARILY AT LARGE ORDERS. AN ATTEMPT IS MADE TO USE SCRATCH
                C TAPES ABOUT AS ECONOMICALLY AS POSSIBLE, AND INNER PRODUCTS ARE ACCUM-
                C   ULATED IN D.P. TO ACHIEVE HIGH ACCURACY.
                C
                C
                C
                C
  ISN 0003            DIMENSION A(1),Q(1),U(1),V(1),B(1),P(1)
  ISN 0004            DOUBLE PRECISION P,SUM,PI
  ISN 0005            INTEGER ASIZE
  ISN 0006            EQUIVALENCE (SUM,SUN)
                C
                C INITIALIZE.
                C
  ISN 0007            NUMBER = INDEX2
  ISN 0008            CALL OVERFL(I)
  ISN 0009            N = IABS(NO)
  ISN 0010            N1 = N - 1
  ISN 0011            N2 = N - 2
                C
                C TEST TO SEE IF MATRIX IS ALREADY IN CORE.
                C
  ISN 0012            IF (NUMBER) 70,2000,80
                C
                C NO. REWIND TAPES, SET THE INDICATORS INCORE AND KEY, AND READ IN THE
                C   FIRST ROW OF MATRIX. THE ARRAY A IS USED TO HOLD AS MANY ROWS AS
                C   POSSIBLE. AT EACH STEP THE FIRST ROW IN CORE IS ELIMINATED AND ONE
                C   OR MORE NEW ROWS BROUGHT IN UNTIL ALL REMAINING ROWS ARE IN CORE.
                C
  ISN 0013         70 REWIND 2
  ISN 0014            REWIND 3
  ISN 0015            REWIND 4
  ISN 0016            INCORE = 1
  ISN 0017            KEY = 0
  ISN 0018            READ (2)  (A(I), I = 1,N)
  ISN 0019            GO TO 90
                C
                C YES. SET THE INDICATOR INCORE AND DEFINE THE INDICES WHICH DESIGNATE
                C   FIRST AND LAST ELEMENTS OF FIRST ROW IN CORE. THE ARRAY A ALWAYS
                C   HOLDS THE ENTIRE MATRIX AND AS EACH ROW IS ELIMINATED, IT IS RE-
                C   PLACED BY THE CORRESPONDING V.
  ISN 0020         80 INCORE = 0
  ISN 0021            K2K2 = 1
  ISN 0022            K2N = N
  ISN 0023         90 NUMMER = IABS(NUMBER)
                C
                C DISPOSE OF TRIVIAL CASES.
                C
  ISN 0024            IF (ASIZE - N - N1) 100,110,110
  ISN 0025        100 MISS = 1
  ISN 0026            RETURN
```

```
ISN 0027        110 IF (N2) 120,125,130
ISN 0028        120 B(1) = A(1)
ISN 0029            GO TO 750
ISN 0030        125 U(2) = 0.
              C
              C INCORE = 0 MEANS THE (REMAINING) MATRIX FITS IN CORE. INCORE = 1 MEANS
              C   THAT SOME ROWS ARE STILL ON TAPE.
              C IF KEY = 0 (K EVEN) READ T2 AND WRITE T3. REVERSE IF KEY = 1 (K ODD).
              C M IS NUMBER OF ROWS (STARTING WITH K+1) WHICH FIT IN CORE. LIM IS THE
              C   INDEX OF LAST SUCH ROW. EVENTUALLY LIM REACHES N, AND THEREAFTER
              C   READING AND WRITING ARE SUPPRESSED.
              C
ISN 0031        130 K2 = 1
ISN 0032            K3 = 2
ISN 0033            K1K1 = 1
ISN 0034            NK1 = N
ISN 0035            NEXTM = LIMIT(N,ASIZE)
ISN 0036            LIM = NEXTM
              C
              C MAJOR LOOP. EXCEPT FOR K = 0 AND K = N-2, THE KTH STEP COMPLETES STAGE
              C   K OF THE HOUSEHOLDER ALGORITHM AND BEGINS STAGE K+1.  K = 0 PREPARES
              C   FOR STAGE 1, AND K = N-2 (VIRTUALLY) COMPLETES THE ALGORITHM.
              C
              C IF K IS G.T. 0, THE LOOP STARTS WITH P UPPER K AND V UPPER K STORED IN
              C   Q(K+1)...Q(N) AND U(K+1)...U(N). THE 1ST K DIAGONAL ELEMENTS ARE IN
              C   Q(1)...Q(K), WITH OFFDIAG ELEMENTS IN U(1)...U(K) AND THEIR SQUARES
              C   NECESSARY, ON TAPE. IF L IS THE 1ST K-VALUE FOR WHICH ROWS L&1...N
              C   IN V(1)...V(K). ROWS K+1...N OF A UPPER K ARE STORED IN CORE AND. IF
              C   FIT IN CORE, THEN A UPPER K STARTS IN A(1) FOR K L.T.E. L. AND IN
              C   A((K-L)(N-L) - (K-L)(K-L-1)/2 + 1) FOR K G.T. L.
              C
              C NIX = 0, BUT A POINTLESS PROVISION OF FORTRAN IV PROHIBITS EXPLICIT 0.
              C
ISN 0037            NIX = 0
ISN 0038            DO 690  K = NIX,N2
ISN 0039            K1 = K2
ISN 0040            K2 = K3
ISN 0041            K3 = K + 3
ISN 0042            NK = NK1
ISN 0043            NK1 = NK - 1
ISN 0044            M = NEXTM
              C
              C TEST TO SEE IF ENTIRE (REMAINING) MATRIX FITS IN CORE
              C
ISN 0045            IF (INCORE) 2000,140,150
              C
              C YES. IF K G.T. 0, FORM TAU, Q, AND NEW A. IF K = 0, SKIP.
              C
ISN 0046        140 K1K1 = K2K2
ISN 0047            K1N = K2N
ISN 0048            IF (K) 2000,250,200
              C
              C NO. TRY AGAIN NEXT TIME
              C
ISN 0049        150 INCORE = N - LIM
ISN 0050            K1N = NK
```

```
                C
                C IF K = 0, READ IN (AT LEAST) EARLY ROWS INTO UPPER TRIANGULAR ARRAY.
                C
ISN 0051              IF (K) 2000,160,180
ISN 0052          160 II = N + 1
ISN 0053              IN = N + N1
ISN 0054              DO 170 I = 2,LIM
ISN 0055              II = IN + 1
ISN 0056          170 IN = IN + N - I
ISN 0057              GO TO 250
                C
                C NOW SKIP TO FORMATION OF NEW SUM, V, ALF, AND P.
                C
                C
                C
                C IF K IS G.T.0, FORM TAU AND GET Q (FROM OLD P AND V, NOW IN Q AND U.)
                C
                C IF MATRIX FITS, DON'T FRET ABOUT TAPES 2 AND 3.
                C
                C OTHERWISE GET SET FOR MORE READING AND WRITING.
                C
ISN 0058          180 IF (INCORE) 2000,200,190
ISN 0059          190 REWIND 2
ISN 0060              REWIND 3
ISN 0061          200 SUM = 0.D0
ISN 0062              DO 210  I = K1,N
ISN 0063          210 SUM = SUM + Q(I)*U(I)
ISN 0064              TAU = .5 * ALF * SUM
ISN 0065              DO 220  I = K1,N
ISN 0066          220 Q(I) = Q(I) - TAU*U(I)
                C
                C CONVERT ROWS K+1...LIM TO ROWS OF A UPPER K+1
                C
ISN 0067              II = K1K1
ISN 0068              IN = K1N
ISN 0069              DO 240  I = K1,LIM
ISN 0070              J = I
ISN 0071              DO 230  IJ = II,IN
ISN 0072              A(IJ) = A(IJ) - Q(I)*U(J) - Q(J)*U(I)
ISN 0073          230 J = J + 1
ISN 0074              II = IN + 1
ISN 0075          240 IN = IN + N - I
                C
                C IF K = N-2, NO NEW SUM, V, ALF, ETC. ARE NEEDED. KICK OUT AND MOP UP.
                C
ISN 0076          250 SUM = 0.D0
ISN 0077              K1K2 = K1K1 + 1
ISN 0078              IF (N - K2) 2000,690,260
ISN 0079          260 DO 270  IJ = K1K2,K1N
ISN 0080          270 SUM = SUM + A(IJ)*A(IJ)
ISN 0081              Q(K+1) = A(K1K1)
ISN 0082              V(K+1) = SUN
ISN 0083              U(K+1) = SQRT(SUN)
                C
                C IF SUM = 0, ROW K+1 ALREADY CONFORMS TO TRI-DIAG FORM. MAKE SPECIAL
                C   DEFINITION OF V AND ALF.
```

```
C
      IF (SUM) 2000,280,290
  280 V(K+2) = 1.
      ALF = 2.
      GO TO 320
C
C ORDINARY DEFINITION OF V AND ALF.
C
  290 IF (A(K1K2)) 310,310,300
  300 U(K+1) = -U(K+1)
  310 V(K+2) = A(K1K2) - U(K+1)
      ALF = 1. / (V(K+1) + ABS(A(K1K2)*U(K+1)))
  320 J = K1K1 + 2
      DO 330  I = K3,N
      V(I) = A(J)
  330 J = J + 1
C
C IF MATRIX WAS INITIALLY IN CORE, STORE V UPPER K+1 IN K+1 ST ROW OF A.
C
      IF (NUMBER) 336,2000,333
  333 A(K1K1) = ALF
      A(K1K1+1) = V(K+2)
      GO TO 440
C
C OTHERWISE, WRITE V UPPER K+1 ON TAPE 4.
C
  336 WRITE (4) ALF, (V(I), I = K2,N)
C
C IF MATRIX FAIL TO FIT, ELIMINATE ROW K+1 (NOW SUPERFLUOUS) AND MOVE
C   OTHER ROWS FORWARD TO MAKE ROOM FOR 1 OR MORE NEW ROWS.
C
      IF (INCORE) 2000,440,340
  340 LK1 = NK + 1
      MOVE = II - LK1
      J = LK1
      DO 350  I = 1,MOVE
      A(I) = A(J)
  350 J = J + 1
      II = MOVE + 1
      IN = NK + MOVE - M
C
C UPDATE LIM. BRING IN NEW ROWS AND CONVERT TO ROWS OF A UPPER K+1.
C
      NEXTM = LIMIT(NK1,ASIZE)
      LIMINC = LIM + 1
      LIM = K1 + NEXTM
      DO 430  I = LIMINC,LIM
  360 IF (KEY) 2000,370,380
  370 READ (2) (A(IJ), IJ = II,IN)
      GO TO 390
  380 READ (3) (A(IJ), IJ = II,IN)
  390 IF (K) 2000,420,400
  400 J = I
      DO 410  IJ = II,IN
      A(IJ) = A(IJ) - Q(I)*U(J) - Q(J)*U(I)
  410 J = J + 1
```

```
ISN 0123        420 II = IN + 1
ISN 0124        430 IN = IN + N - I
ISN 0125            K2K2 = 1
ISN 0126            K2N = NK1
ISN 0127            GO TO 450
ISN 0128        440 K2K2 = K1N + 1
ISN 0129            K2N = K1N + NK1
              C
              C COMPUTE POSITIONS K+2...LIM OF AV AND SUM UP TO LIM FOR THE LATE POSI-
              C   TIONS, IF ANY. AT STEP I, ITH ELEMENT OF AV IS COMPLETED, AND EFFECT
              C   OF COLUMN I ON ELEMENTS I+1...N IS TAKEN INTO ACCOUNT.
              C
ISN 0130        450 II = K2K2
ISN 0131            IN = K2N
ISN 0132            DO 460  I = K2,N
ISN 0133        460 P(I) = 0.D0
ISN 0134            DO 500  I = K2,LIM
ISN 0135            J = I
ISN 0136            DO 470  IJ = II,IN
ISN 0137            P(I) = P(I) + A(IJ)*V(J)
ISN 0138        470 J = J + 1
ISN 0139            IF (N - I) 2000,500,480
ISN 0140        480 II1 = II + 1
ISN 0141            J = I + 1
ISN 0142            DO 490  IJ = II1,IN
ISN 0143            P(J) = P(J) + A(IJ)*V(I)
ISN 0144        490 J = J + 1
ISN 0145            II = IN + 1
ISN 0146        500 IN = IN + N - I
              C
              C IF SOME ROWS ARE STILL ON TAPE, READ THEM IN, 1 AT A TIME. EACH ROW IS
              C   CONVERTED (IF K G.T. 0), THEN USED IN CALCULATION OF AV, THEN PUT ON
              C   THE OTHER TAPE TO MAKE ROOM FOR THE NEXT ROW.
              C
ISN 0147            IF (N - LIM) 2000,660,510
ISN 0148        510 II = 1
ISN 0149            IN = N - LIM
ISN 0150            LIMINC = LIM + 1
ISN 0151            DO 640  I = LIMINC,N
ISN 0152            IF (KEY) 2000,520,530
ISN 0153        520 READ (2) (B(IJ), IJ = II,IN)
ISN 0154            GO TO 540
ISN 0155        530 READ (3) (B(IJ), IJ = II,IN)
ISN 0156        540 IF (K) 2000,570,550
ISN 0157        550 J = I
ISN 0158            DO 560  IJ = II,IN
ISN 0159            B(IJ) = B(IJ) - Q(I)*U(J) - Q(J)*U(I)
ISN 0160        560 J = J + 1
ISN 0161        570 J = I
ISN 0162            DO 580  IJ = II,IN
ISN 0163            P(I) = P(I) + B(IJ)*V(J)
ISN 0164        580 J = J + 1
ISN 0165            IF (N - I) 2000,610,590
ISN 0166        590 J = I + 1
ISN 0167            II1 = II + 1
ISN 0168            DO 600  IJ = II1,IN
```

```
ISN 0169          P(J) = P(J) + B(IJ)*V(I)
ISN 0170      600 J = J + 1
ISN 0171      610 IF (KEY) 2000,620,630
ISN 0172      620 WRITE (3) (B(IJ),IJ = II,IN)
ISN 0173          GO TO 640
ISN 0174      630 WRITE (2) (B(IJ),IJ = II,IN)
ISN 0175      640 IN = IN - 1
ISN 0176      650 KEY = 1 - KEY
ISN 0177      660 CONTINUE
                C
                C PLACE NEW V AND P IN U AND Q.
                C
ISN 0178      670 DO 680  I = K2,N
ISN 0179          U(I) = V(I)
ISN 0180      680 Q(I) = ALF * P(I)
ISN 0181      690 CONTINUE
                C
                C MOP UP BY COMPLETING ARRAYS OF DIAGONAL, OFF-DIAG AND SQUARE ELEMENTS.
                C
ISN 0182          Q(N-1) = A(K1K1)
ISN 0183          U(N-1) = A(K1K1+1)
ISN 0184          V(N-1) = U(N-1)*U(N-1)
ISN 0185          Q(N) = A(K1K1+2)
ISN 0186      700 IF (NO) 710,2000,720
ISN 0187      710 WRITE (6,10) (Q(I), I = 1,N)
ISN 0188       10 FORMAT (/////18H TRI-DIAGONAL FORM////5X,18H DIAGONAL ELEMENTS//(6
                 1X,1P8E15.7))
ISN 0189          WRITE (6,20) (U(I), I = 1,N1)
ISN 0190       20 FORMAT (///5X,22H SUB-DIAGONAL ELEMENTS//(6X,1P8E15.7))
ISN 0191      720 CALL OVERFL(I)
ISN 0192          GO TO (730,740),I
ISN 0193      730 MISS = 2
ISN 0194          GO TO 1000
ISN 0195      740 CALL GROPER(N,INDEX1,NUMMER,Q,U,P,V,B)
ISN 0196      750 MISS = 0
ISN 0197          GO TO 1000
ISN 0198     2000 MISS = 3
ISN 0199     1000 RETURN
ISN 0200          ENTRY SECURE(X,LOW,KOUNT,MID,W)
ISN 0201          DIMENSION X(MID,1),W(1)
ISN 0202          CALL TRIVEC(A,Q,U,V,W,P,P(N/2 + 1),N)
ISN 0203          K = LOW
ISN 0204          DO 800  I = 1,KOUNT
ISN 0205          CALL VLANDT(B(K),X(1,I))
ISN 0206      800 K = K + 1
ISN 0207          RETURN
ISN 0208          END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

```
          SUBROUTINE GROPER(N,LIM1,NUMB,D,OFFD,PFFD,SEC,SIGMA)
          DIMENSION PFFD(1)
          DIMENSION D(1),OFFD(1),SEC(1),SIGMA(1)
          LIM2 = LIM1 + NUMB - 1
          CALL PREP(N,D,SEC,ROOT,LORD)
          N1 = N - 1
          BOUND = AMAX1(ABS (D(1))+ABS (OFFD(1)),ABS (OFFD(N1))+ABS (D(N)))
          IF (N - 2) 16,200,100
      100 DO  1    I = 2,N1
        1 BOUND = AMAX1(BOUND,ABS (OFFD(I-1)) + ABS (D(I)) + ABS (OFFD(I)))
      200 DO 2  I = LIM1,LIM2
          SIGMA(I) = -BOUND
        2 PFFD(I) = BOUND
          LORD = 0
          RUTE = 1.0
          L = LIM1 - 1
        3 K = L + 1
          IF (K - LIM2) 4,4,13
        4 ROOT = .5 * (SIGMA(K) + PFFD(K))
        5 DO 6  I = K,LIM2
          IF (PFFD(K) - PFFD(I)) 7,6,7
        6 L = I
        7 IF (ROOT - RUTE) 8,3,8
        8 CALL DET(LORD)
          DO 11  I = K,L
          IF(I -LORD) 9,9,10
        9 SIGMA(I) = ROOT
          GO TO 11
       10 PFFD(I) = ROOT
       11 CONTINUE
          RUTE = ROOT
          IF (ROOT) 4,12,4
       12 KING = LORD
          GO TO 4
       13 IF (KING - LIM1) 16,14,14
       14 DO 15  I = LIM1,KING
       15 SIGMA(I) = PFFD(I)
       16 RETURN
            END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE TRIVEC(A,D,OFFD,P,Q,R,S,N)
            C  SYMMETRIC MATRIX EIGENVECTOR CALCULATION.
            C  GIVEN THE ENTRIES (D AND OFFD) OF THE HOUSEHOLDER TRI-DIAGONAL FORM B
            C    OF A REAL SYMMETRIC MATRIX A, AND GIVEN A GOOD APPROXIMATE ROOT OF
            C    B (AND A) THIS FORTRAN 4 SUBROUTINE COMPUTES A UNIT EIGENVECTOR X
            C    OF B, THEN TRANSFORMS IT TO A UNIT VECTOR OF A, USING THE VECTORS W
            C    STORED IN THE A ARRAY.
ISN 0003          DIMENSION A(1),D(1),OFFD(1),P(1),Q(1),R(1),S(1),X(1)
ISN 0004          DOUBLE PRECISION SUM
ISN 0005          COMMON /INFO/ SUM,M,IX,IA
            C
            C  PART 1.  PRELIMINARIES.
            C
ISN 0006          IX = 1
ISN 0007          IA = 1
ISN 0008          N1 = N - 1
ISN 0009          N2 = N - 2
ISN 0010          RETURN
ISN 0011          ENTRY VLANDT(ROOT,X)
ISN 0012          ASSIGN 170 TO KOUNT
ISN 0013          TOL = 0.
ISN 0014          DO 100  I = 1,N
ISN 0015          P(I) = D(I) - ROOT
ISN 0016          Q(I) = OFFD(I)
ISN 0017          R(I) = 0.
ISN 0018          TOL = AMAX1(TOL,ABS(D(I)))
ISN 0019      100 X(I) = RDM(X) + .1
ISN 0020          TOL = (TOL + 1.E-15) * 1.E-15
            C
            C  PART 2.  MATRIX DECOMPOSITION.
            C
ISN 0021          DO 150  I = 1,N1
ISN 0022          T = ABS (P(I))
ISN 0023          U = ABS (OFFD(I))
ISN 0024          IF (T + U - TOL) 110,120,120
ISN 0025      110 P(I) = TOL
ISN 0026          T = P(I)
ISN 0027      120 IF (T - U) 130,140,140
ISN 0028      130 S(I) = P(I)/OFFD(I)
ISN 0029          S(I) = OR(S(I), 1)
ISN 0030          TEMP = Q(I)
ISN 0031          P(I) = OFFD(I)
ISN 0032          Q(I) = P(I+1)
ISN 0033          R(I) = Q(I+1)
ISN 0034          P(I+1) = TEMP - S(I)*Q(I)
ISN 0035          Q(I+1) = -S(I)*R(I)
ISN 0036          GO TO 150
ISN 0037      140 S(I) = OFFD(I)/P(I)
ISN 0038          S(I) = AND(S(I),-2)
ISN 0039          P(I+1) = P(I+1) - S(I)*Q(I)
ISN 0040      150 CONTINUE
ISN 0041          IF (ABS(P(N)) .LT. TOL)  P(N) = TOL
ISN 0043          GO TO 210
            C
            C  PART 3.  RIGHT SIDE MODIFICATION.
```

```
C
  170 ASSIGN 330 TO KOUNT
      DO 200  I = 1,N1
      TEMP = AND(S(I), 1)
      IF (TEMP) 180,190,180
  180 T = X(I)
      X(I) = X(I+1)
      X(I+1) = T - S(I)*X(I)
      GO TO 200
  190 X(I+1) = X(I+1) - S(I)*X(I)
  200 CONTINUE
C
C  PART 4.  TRIANGULAR SYSTEM SOLUTION.
C
  210 X(N) = X(N)/P(N)
      X(N1) = (X(N1) - Q(N1)*X(N)) / P(N1)
      DO 220  I = 2,N1
      K = N - I
  220 X(K) = (X(K) - Q(K)*X(K+1) - R(K)*X(K+2)) / P(K)
C
C  PART 5. SCALING TO UNIT VECTOR.
C
  230 SUM = 0.D0
      M = N
      SCALAR = SQRT(DOTPRO(X,X))
      DO 250  I = 1,N
  250 X(I) = X(I)/SCALAR
      GO TO KOUNT, (170,330,370)
C
C  PART 6.  TRANSFORMATION BY ORTHOGONAL MATRICES.
C
  330 L = (N*(N+1))/2 - 4
      DO 360  I = 1,N2
      NI = N - I
      SUM = 0.D0
      M = I + 1
      SCALAR = -A(L-1) * DOTPRO(X(NI),A(L))
      IJ = L
      DO 350  J = NI,N
      X(J) = X(J) + SCALAR*A(IJ)
  350 IJ = IJ + 1
  360 L = L - I - 3
      ASSIGN 370 TO KOUNT
      GO TO 230
  370 RETURN
      END
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,LIST,DECK,LOAD,MAP,NOEDIT,ID
ISN 0002            FUNCTION LIMIT(N,M)
ISN 0003            K = N
ISN 0004
ISN 0004            L = 0
ISN 0005            DO 100  I = 1,N
ISN 0006            L = L + K
ISN 0007            K = K - 1
ISN 0008            IF (M - L) 120,110,100
ISN 0009        100 CONTINUE
ISN 0010            LIMIT = N
ISN 0011            RETURN
ISN 0012        110 LIMIT = I
ISN 0013            RETURN
ISN 0014        120 LIMIT = I - 1
ISN 0015            RETURN
ISN 0016             END
```

E-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,MAP
IEW0132 CLOCKS
IEW0132 DET
IEW0132 PREP
****FORTHCLG DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
**MODULE HAS BECOME NOT EXECUTABLE

DIAGNOSTIC MESSAGE DIRECTORY

IEW0132 ERROR - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE.

MODULE MAP

CONTROL SECTION			ENTRY							
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
IHCFMODR	00	68								
			AMOD	00	DMOD	26				
POW16	68	38								
			DPOW16	68						
DOTPRO	A0	4C								
INFO	F0	14								
$MAIN=	108	3D116								
			$MAIN	3BA80						
BLK1	3D220	48								
BLK2	3D268	50								
BLK3	3D2B8	20								
BLK4	3D2D8	18								
BLK5	3D2F0	4B0								
BLK11	3D7A0	20								
BLK77	3D7C0	78								
BLK78	3D838	18								
GEORGE	3D850	4								
ABORT	3D858	18								
BLK68	3D870	8								
BLKDS	3D878	6C								
CLOCK=	3D8E8	18A								
			CLOCK	3D928						
GEROGE	3DA78	4								
DSRCH=	3DA80	9EC								
			DSRCH	3DD20						
SCAPR=	3E470	22C								
			SCAPR	3E498						
AFX=	3E6A0	12AC								
			AFX	3E988						
DEIGN=	3F950	4C2								
			DEIGN	3FB90						
IMEQD=	3FE18	C10								
			IMEQD	3FE88						
PEAIQ=	40A28	3C1A								
			PEAIQ	43E30						

NAME	ORIGIN	LENGTH
QGEN=	44648	25C34
DOTPR=	6A280	156
MINVD=	6A3D8	B78
BOXNO=	6AF30	1D[illegible]
RDM=	6B130	280
SETCLK=	6B3B0	DA
DARSIN=	6B490	15A
SYMBIG=	6B5F0	156C
GROPER=	6CB60	4B6
TRIVEC=	6D018	7DE
LIMIT=	6D7F8	132
IHCFEXIT*	6D930	1C
IHCFIOSH*	6D950	D30
IHCUATBL*	6E680	148
IHCFCOMH*	6E7C8	1CDC
=PROCES *	6F8A8	352
SYSERR= *	6FC00	7E4
SYSFDV *	703E8	8
SYSCNT *	703F0	3C
SYSOPT *	70430	C
SYSRXT= *	70440	2
IHCLEXP *	70448	1CC
IHCLSCN *	70618	190
INTSW *	707A8	8
IHCLLOG *	707B0	178
IHCFRXPI*	70928	94
IHCLSQRT*	709C0	96
FINAGL *	70A58	6C
IHCLATN2*	70AC8	22C

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
QGEN	69340						
DOTPR	6A288						
MINVD	6A450						
BOXNO	6AF78						
RDM	6B168	RDMIN	6B260	RDMOUT	6B30C		
SETCLK	6B3B0						
DARSIN	6B4B0						
SYMBIG	6B740	SECURE	6C9C8				
GROPER	6CBB0						
TRIVEC	6D088	VLANDT	6D2E0				
LIMIT	6D818						
EXIT	6D930						
FIOCS=	6D950						
IBCOM=	6E7C8	FDIOCS=	6E884	=IBXOUT	6F5AC	=BUFLOC	6F79C
SYSERR	6FDC8						
SYSRXT	70440						
DEXP	70448						
DCOS	70618	DSIN	70636				
DLOG10	707B0	DLOG	707CC				
FRXPI=	70928						
DSQRT	709C0						
TAMPER	70A58	ACTION	70A8A	STNDRD	70AA2		
DATAN2	70AC8	DATN[illegible]	70AE4	DATAN	70AFE		

NAME	ORIGIN	LENGTH
IHCLTNCT*	70CF8	188
IHCSSCN *	70E80	11C
IHCSSQRT*	70FA0	AC
IHCSLOG *	71050	10C
IHCFOVER*	71160	50
IHCFCVTH*	711B0	107C

ENTRY ADDRESS 3BA80
TOTAL LENGTH 7222C

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
DCOTAN	70CF8	DTAN	70D14	QDTAN	70E2C		
COS	70E80	SIN	70E9C				
SQRT	70FA0						
ALOG10	71050	ALOG	7106C				
OVERFL	71160						
ADCON=	711B0	FCVZO	712FC	FCVAO	713A2	FCVLO	71432
FCVIO	71768	FCVEO	71C5A	FCVCO	71E5C	INT6SW	72211

APPENDIX II

SIMULATION PROGRAM

This appendix contains a flow chart and listing of the simulations of the various models of the Ames system. The flow chart (Fig. II-1) is of the MAIN program, which is suitable for all simulations. The only change to accommodate the simulations occurs in the equations of motion in subroutine AFX.

The listings given are suitable for all four models, and the AFX subroutine for: 1) AN, 2) MV, 3)error limited, and 4) 6D are also given. ANL simply has limits e_ϕ, e_θ, e_ψ using three pairs of statements of the form:

$$\text{IF} \quad (e_\phi \quad .\text{GT}. \quad \text{V2}) \quad e_\phi = \text{V2}$$

$$\text{IF} \quad (e_\phi \quad .\text{LT}. \quad -\text{V2}) \quad e_\phi = -\text{V2} \quad .$$

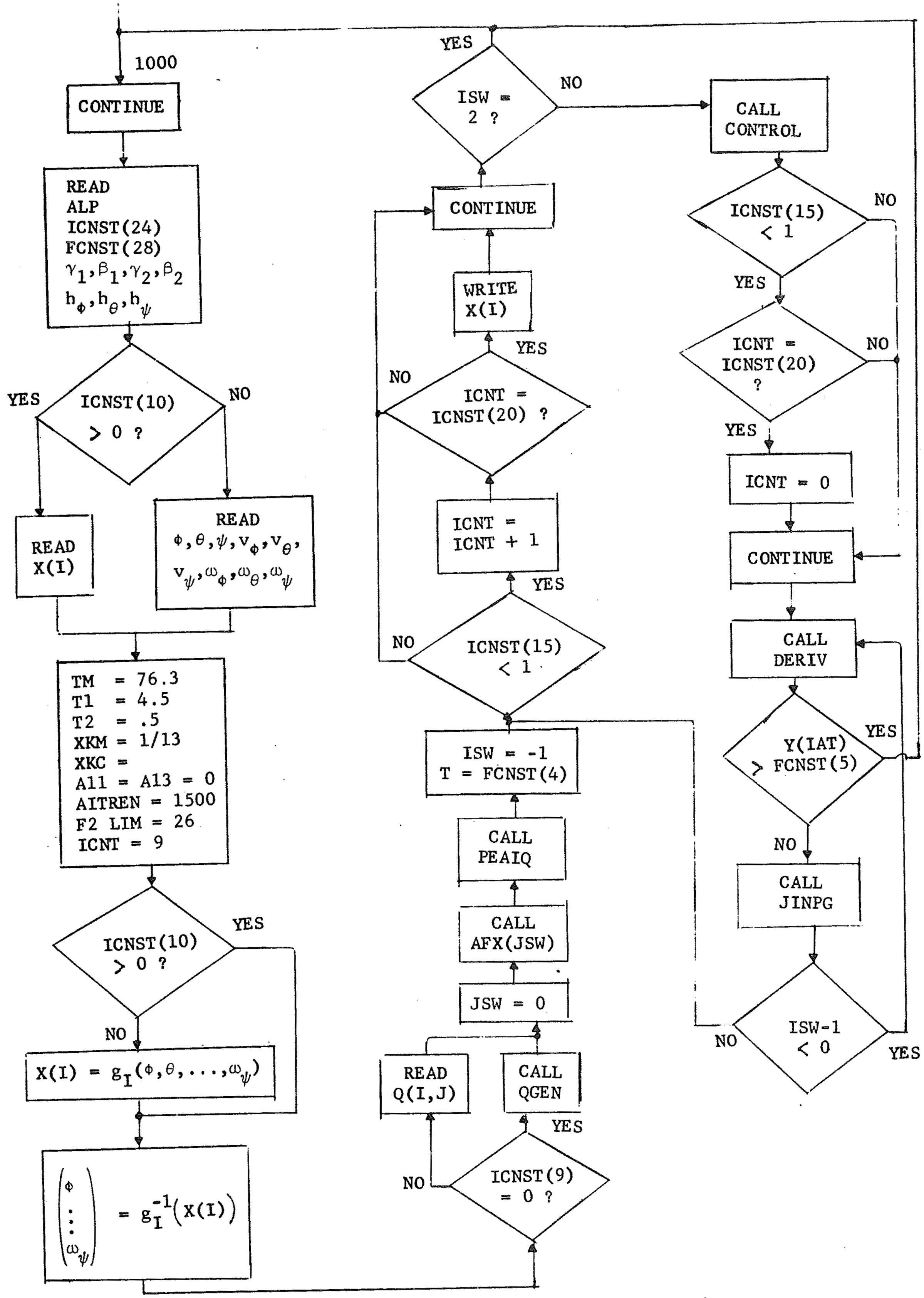


Fig. II-1 Generic Simulation Program Flow Chart

AN & MV LISTING


```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002            IMPLICIT REAL*8 (A-H,O-Z)
              C     NONLINEAR CONTROL SYSTEM SIMULATION
              C     MODIFIED FOR NASA AMES OAO SYSTEM EQUATIONS
              C     NOMENCLATURE
              C        XZERO=    INITIAL STATE
              C        Y     =    PROCESS OUTPUT
              C        X     =    NOISY   OUTPUT
              C        V     =    CONTROL INPUT
              C        U     =    DISTURBED INPUT
              C        DY    =    STATE  DERIVATIVE
              C        K     =    DIMENSION OF STATE SPACE
              C        L     =    DIMENSION OF CONTROL SPACE
              C        T     =    TIME
ISN 0003            COMMON/BLKX/YZRO(9)
ISN 0004            COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI
ISN 0005            COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM
ISN 0006            COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C
ISN 0007            COMMON /BLK4/ HPHI,HTHT,HPSI
ISN 0008            COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0009            COMMON/BLK6/ IFLAG
ISN 0010             COMMON/BLK77/X(15)
ISN 0011            COMMON/BLK78/X1E,X4E,X7E
ISN 0012            DIMENSION  ICNST(24),FCNST(28),XZERO(15),      Y(15),U(15),V(15),
                   1DY(15),ALP(18)
ISN 0013            COMMON  ICNST,FCNST,ALP
ISN 0014            EQUIVALENCE (ICNST(1),K),(ICNST(2),L)
ISN 0015            EQUIVALENCE (FCNST(1),H),(FCNST(2),TLIM),(FCNST(3),EMAX)
ISN 0016       1000 CONTINUE
ISN 0017        102 READ (5,1101) ALP
ISN 0018       1101 FORMAT(18A4)
              C     HEADINGS
ISN 0019            WRITE (6,7190)
ISN 0020       7190 FORMAT (1H1 )
ISN 0021            WRITE (6,190)
ISN 0022        190 FORMAT(55H                           NONLINEAR CONTROL SYSTEM SIMULATION)
ISN 0023            WRITE(6,191)(ALP(J),J=1,18)
ISN 0024        191 FORMAT(1H / 10X,18A4)
              C     ICNST IS A VECTOR OF 24 INTEGERS  ON  ONE  CARD
ISN 0025        100 READ (5,101) (ICNST(J),J=1,24)
ISN 0026        101 FORMAT (24I3)
ISN 0027            IFLAG=ICNST(24)
ISN 0028            IAT= ICNST(8)
              C     FCNST IS A  VECTOR OF 28 FLOAT  CONSTANTS ON  4  CARDS
ISN 0029            READ (5,111) (FCNST(J),J=1,28)
ISN 0030        111 FORMAT(7E10.4)
              C     XZERO IS  THE INITIAL  STATE   ON   2  CARDS
              C     ICNST(10)GT 0 PERMITS DIRECT STATE INPUT
ISN 0031            IF(ICNST(10).GT.0)GO TO 750
ISN 0033            READ(5,1001)PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI
ISN 0034            GO TO 751
ISN 0035       1001 FORMAT(5E14.7)
ISN 0036        750 READ(5,1001) (X(I),I=1,9)
ISN 0037        751 TM    = 76.8
ISN 0038            T1    =  4.5
ISN 0039            T2    =  0.5
```

```
ISN 0040          XKM    =  1.0/13.0
ISN 0041          XKC    =  2.685 E005
ISN 0042          A13    =  0.0
ISN 0043          A11     = 0.0
ISN 0044          AITREN= 1500.0
ISN 0045          F2LIM = 26.0
ISN 0046          ICNT=9
ISN 0047          READ(5,1001)GAM1C,GAM2C,BET1C,BET2C
ISN 0048          READ(5,1001)HPHI,HTHT,HPSI
ISN 0049          PI= 3.1415926
ISN 0050          DTR = PI/180.
ISN 0051          GAM1C = GAM1C * DTR
ISN 0052          GAM2C = GAM2C * DTR
ISN 0053          BET1C = BET1C * DTR
ISN 0054          BET2C = BET2C * DTR
ISN 0055          DIF = GAM1C - GAM2C
ISN 0056          D12 = 2.0 * DIF / ABS(DIF)
ISN 0057          SDIF=SIN(DIF)
ISN 0058          SG1C  = SIN(GAM1C)
ISN 0059          CG1C  = COS(GAM1C)
ISN 0060          SG2C  = SIN(GAM2C)
ISN 0061          CG2C  = COS(GAM2C)
ISN 0062          SB1C  = SIN(BET1C)
ISN 0063          CB1C  = COS(BET1C)
ISN 0064          TB1C  = SB1C/CB1C
            C     DEFINE STATE VECTOR TO BE USED AS INITIAL STATE
ISN 0065          X1E = ( AITREN)/(XKM*XKC)*(HPHI-HTHT*(A11*SG1C-A13*SG2C-CG1C*TB1C
                1  )/(D12*SDIF) &HPSI*(A11*CG1C-A13*CG2C&SG1C*TB1C)/(D12*SDIF))
ISN 0066          X4E = ( AITREN)/(XKM*XKC)*(HTHT/(D12*SDIF))
ISN 0067          X7E = (-AITREN)/(XKM*XKC)*(HPSI/(D12*SDIF))
ISN 0068          IF(ICNST(10).GT.0) GO TO 1999
ISN 0070          X(1) = PHI*DTR-X1E
ISN 0071          X(2)=(VPHI-AITREN*HPHI)/10.0
ISN 0072          X(3)= WPHI+T1*AITREN/(T2*XKM)*HPHI
ISN 0073          X(4) = THT*DTR-X4E
ISN 0074          X(5)=(VTHT-AITREN*HTHT)/10.0
ISN 0075          X(6)= WTHT+T1*AITREN/(T2*XKM)*HTHT
ISN 0076          X(7) = PSI*DTR-X7E
ISN 0077          X(8)=(VPSI-AITREN*HPSI)/10.0
ISN 0078          X(9)= WPSI+T1*AITREN/(T2*XKM)*HPSI
ISN 0079     1999 WRITE(6,2000)(X(I),I=1,9)
ISN 0080     2000 FORMAT(1H /10X,15H INITIAL STATE ,/(1X,5E20.7))
ISN 0081          IF(ICNST(10).EQ.0) GO TO 4377
ISN 0083          PHI = (X(1)+X1E)/DTR
ISN 0084          VPHI=10.0*X(2)+AITREN*HPHI
ISN 0085          WPHI=X(3)-T1*AITREN/(T2*XKM)*HPHI
ISN 0086          THT = (X(4)+X4E)/DTR
ISN 0087          VTHT=10.0*X(5)+AITREN*HTHT
ISN 0088          WTHT=X(6)-T1*AITREN/(T2*XKM)*HTHT
ISN 0089          PSI = (X(7)+X7E)/DTR
ISN 0090          VPSI=10.0*X(8)+AITREN*HPSI
ISN 0091          WPSI=X(9)-T1*AITREN/(T2*XKM)*HPSI
ISN 0092     4377 YZRO(1) = PHI
ISN 0093          YZRO(2) = VPHI
ISN 0094          YZRO(3) = WPHI
ISN 0095          YZRO(4) = THT
```

```
ISN 0096          YZRO(5) = VTHT
ISN 0097          YZRO(6) = WTHT
ISN 0098          YZRO(7) = PSI
ISN 0099          YZRO(8) = VPSI
ISN 0100          YZRO(9) =WPSI
ISN 0101          WRITE(6,2500)(YZRO(I),I=1,9)
ISN 0102     2500 FORMAT(1H /10X, 7H YZRO =,/(1X,5E20.7))
ISN 0103          DO 36 I=1,15
ISN 0104       36 XZERO(I) = X(I)
            C     IF ICNST(9) IS GT ZERO Q-MATRIX MUST BE SPECIFIED,OTHERWISE IT'S R ANDOM
ISN 0105          IF(ICNST(9).EQ.0) GO TO 843
ISN 0107          READ(5,88) (( Q(I,J),J=1,9),I=1,9)
ISN 0108       88 FORMAT(5E16.7)
ISN 0109          GO TO 844
ISN 0110      843 CALL QGEN
ISN 0111      844 JSW =0
ISN 0112          CALL AFX(JSW)
ISN 0113          CALL PEAIQ
            C     INITIALIZATION
ISN 0114          ISW = -1
ISN 0115          T= FCNST(4)
ISN 0116          DO  50   J=1,K
ISN 0117       50 Y(J)  =   XZERO(J)
            C     WRITING  OF  OUTPUT. THIS FORMAT ONLY GOOD  FOR  10TH ORDER
            C     SYSTEM  OR LOWER
ISN 0118      200 IF(ICNST(15).GE.1)GO TO 900
ISN 0120          ICNT=ICNT+1
ISN 0121          IF(ICNT.NE.ICNST(20)) GO TO 900
ISN 0123          WRITE (6,201) T
ISN 0124      201 FORMAT(1H /,30X,5H TIME F10.5)
ISN 0125          WRITE (6,202) (Y(J),J=1,K)
ISN 0126      202 FORMAT(6H STATE / 10H    TRUE  , (5E20.8))
ISN 0127      900 CONTINUE
ISN 0128          DO 10  I=1,K                                                NOISE
ISN 0129       10 X(I) = Y(I)
ISN 0130      230 IF (ISW.EQ.2) GO TO 1000
ISN 0132          CALL CNTRL(V,T)
ISN 0133          IF(ICNST(15).GE.1)GO TO 901
ISN 0135          IF(ICNT.NE.ICNST(20)) GO TO 901
ISN 0137      240 WRITE(6,241) (V(J),J=1,L)
ISN 0138      241 FORMAT(8H CONTROL, / 10H    TRUE   ,(5E20.8))
ISN 0139          ICNT=0
ISN 0140      901 CONTINUE
ISN 0141          DO 11 I = 1, L
ISN 0142       11 U(I) = V(I)
ISN 0143      300 CALL  DERIV (DY,Y, U, T)
ISN 0144          IF( Y(IAT) .GT. FCNST(5) ) GO TO 1000
ISN 0146      350 CALL  JINPG (Y,DY,T,TLIM,H,ISW,K,EMAX)
            C                                                      WSI)0003,6(ETI RW
ISN 0147     3300 FORMAT( 6H ISW= ,I5,19H JUST OUT OF JINPG ///)
ISN 0148          IF(ISW-1)300, 200,200
ISN 0149          END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          SUBROUTINE  JINPG(QN,QNM, TIME, TLIM, STEP,  ISW, K, EMAX )
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
           C      FORTH ORDER RUNGE-KUTTA  SUBROGRAM
           C      DIMENSION OF STATE SPACE CANNOT EXCEED 15
           C      NOMENCLATURE
           C              QN    =  CURRENT VALUE  OF  STATE
           C              QNM   =  CURRENT VALUE  OF  STATE  DERIVATIVE
           C              K     =  DIMENSION STATE SPACE
ISN 0004          DIMENSION SUM(15), QN1(15), QN(15), QNM(15)                    NCSS
ISN 0005          IF(ISW.GE.0) GO TO 100
           C      FIRST TIME
ISN 0007        1 ISW= 0
ISN 0008          TZERO =TIME
ISN 0009          EMAX= EMAX
ISN 0010          ICNTR=1
ISN 0011          H2   = 0.5*STEP
ISN 0012          DO  25 I=1,K
ISN 0013          SUM(I) =QNM(I)
ISN 0014          QN1(I) =QN(I)
ISN 0015          QN(I)  =QN(I) & H2* QNM(I)
ISN 0016       25 CONTINUE
ISN 0017          TIME   = TIME & H2
ISN 0018          GO TO 999
           C      TEST FOR ERROR
ISN 0019      100 IF ( ISW.EQ.1 ) GO TO 1
ISN 0021          IF ( ISW.EQ.2) GO TO 1999
           C      TEST FOR FINAL ENTRY
ISN 0023          IF(ICNTR.EQ.3) GO TO 200
ISN 0025          DO 125  I=1,K
ISN 0026          SUM(I) = SUM(I) &  2.0*QNM(I)
ISN 0027      125 QN(I)  =QN1(I)  & H2* QNM(I)
ISN 0028          TIME   = TZERO&H2
ISN 0029          ICNTR  = ICNTR&1
ISN 0030          H2 = H2&H2
ISN 0031      999 RETURN
           C      ERROR  IN SETTING INPUT TO PROGRAM
ISN 0032     1999 WRITE(6,800)
ISN 0033      800 FORMAT(40H INTEGRATION  ERROR, ISW AT ENTRY =1,2    )
ISN 0034          CALL EXIT
           C      LAST  TIME  THRU
ISN 0035      200 H6 = STEP/6.0
ISN 0036          DO    210   I=1,K
ISN 0037      210 QN(I) = QN1(I) & H6*(SUM(I) & QNM(I))
ISN 0038          ISW= 1
ISN 0039          IF(TIME.GT.0.0) GO TO 68
ISN 0041          ATIME =DABS(TIME)
ISN 0042          ATLIM =DABS(TLIM)
ISN 0043          IF(ATIME.GE.ATLIM) GO TO 777
ISN 0045          GO TO 999
ISN 0046       68 IF(TIME.GE.TLIM)GO TO 777
ISN 0048          GO TO  999
ISN 0049      777 ISW = 2                                                        NCSS
ISN 0050          GO TO 999
ISN 0051          END
```

```
            COMPILER OPTIONS - NAME=  $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
   ISN 0002              SUBROUTINE CNTRL(V,T)
   ISN 0003              IMPLICIT REAL*8 (A-H,O-Z)
   ISN 0004              REAL*4 N1,N2,NA,NO,KA,KO,NXA,NRSS
   ISN 0005              DIMENSION N1(1000),N2(1000)
   ISN 0006              DIMENSION ICNST(24),FCNST(28),PX(9),V(15),ALP(12)
   ISN 0007              COMMON/A/ XCOM(1000,13)
   ISN 0008              COMMON/BLKX/YZRO(9)
   ISN 0009              DIMENSION YCOM(12),BUFFER(512)
   ISN 0010              DIMENSION TITLE(12),VA(13),DVA(13)
   ISN 0011              DIMENSION PF(9),QX(9)
   ISN 0012              COMMON ICNST,FCNST,ALP
   ISN 0013              COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
   ISN 0014               COMMON/BLK77/X(15)
   ISN 0015              DATA TITLE/'X1','X2','X3','X4','X5','X6','X7','X8','X9','V',
                        1'VDOT','NORM'/
   ISN 0016              DATA ITEST/1/
   ISN 0017              DATA  ICOUNT/0/
   ISN 0018              KAT = 1000.
   ISN 0019              JSW=1
   ISN 0020          600 FORMAT(2F10.4)
   ISN 0021              ICOUNT = ICOUNT + 1
   ISN 0022              IF(T.EQ.FCNST(4)) JSW=0
   ISN 0024              IF(T.EQ.FCNST(4))  K=1
   ISN 0026              N=K
   ISN 0027              V(1) = 0.0
   ISN 0028              T=T
   ISN 0029              CALL AFX(JSW)                                                        CNTRL
                   C                                                     )9,1=I,)I(F( )1005,6(ETI RW
   ISN 0030         5001 FORMAT(1H / 1X,12H F(X)-VECTOR / 1X,9E13.6)
   ISN 0031              CALL VLAP(PX,VL)
   ISN 0032              CALL VDOTA(PF,QX,VDOT)
                   C     PLOTTING ROUTINES
                   C     THE FOLLOWING ARE PLOTTED --- THE INDIVIDUAL STATE COMPONENTS---
                   C                               --- LIAPUNOV FCT ---
                   C                               --- DERIVATIVE OF LIAPUNOV FCT ---
                   C                               --- NORM OF THE STATE ---
                   C
                   C     THE MATRIX XCOM CONTAINS UP TO 1000 SETS OF THE 12 ITEMS ABOVE
                   C
                   C     NORM OF THE STATE CALC
                   C
   ISN 0033              XNORM = 0.0
   ISN 0034              DO 375 I=1,9
   ISN 0035          375 XNORM = XNORM + X(I)**2
   ISN 0036              IF(T.GT.FCNST(4)) GO TO 380
   ISN 0038              DO 379 I=1,9
   ISN 0039              DO 379 J=1,9
   ISN 0040          379 XCOM(I,J) = 0.0
   ISN 0041          380 IF(ICOUNT.NE.ITEST) GO TO 2000
   ISN 0043              ITEST = ITEST + 10
   ISN 0044              IF(K.GT.100.)GO TO 620
   ISN 0046              WRITE(6,630)(X(I),I=1,6)
   ISN 0047          630 FORMAT(6E15.7)
   ISN 0048          620 CONTINUE
   ISN 0049              DO 385 I=1,9
```

```
ISN 0050    385 XCOM(K,I)= X(I)
ISN 0051        XCOM(K,10) = VL
ISN 0052        XCOM(K,11) = VDOT
ISN 0053        XCOM(K,12) = SQRT(XNORM)
ISN 0054        WRITE(6,403)XCOM(K,12)
ISN 0055    403 FORMAT(1H /1X, 7H NORM =,E14.7/)
ISN 0056        XCOM(K,13) = T
ISN 0057        IF(K.NE.KAT)GO TO 1000
ISN 0059        DO 5 I=1,12
ISN 0060      5 YCOM(I) =   XCOM(1,I)
ISN 0061        DO 10 J=1,12
ISN 0062        DO 10 I=1,KAT
ISN 0063     10 XCOM(I,J)=XCOM(I,J)/YCOM(J)
ISN 0064        WRITE(6,99)K,N
ISN 0065     99 FORMAT( 4H0 K= I5 , 4H N= I5)
ISN 0066        CALL PLOTS(BUFFER,512)
ISN 0067        DO 71 I=1,KAT
ISN 0068     71 N2(I)=XCOM(I,13)
ISN 0069        CALL SCALE(N2,N,10.,NA,NO,1,0)
ISN 0070        DO 777 J=1,12
ISN 0071        DO 72 I=1,KAT
ISN 0072     72 N1(I)=XCOM(I,J)
ISN 0073        WRITE(6,630)(N1(II),II=1,100)
ISN 0074        CALL SCALE(N1,N,6.,KA,KO,1,0)
ISN 0075        CALL AXIS(0.0,0.0,4HTIME,4,10.,0.,NA,NO,0)
ISN 0076        CALL AXIS(0.0,0.0,TITLE(J),4,6.,90.,KA,KO,0)
ISN 0077        CALL LINE(N2,N1,N,1,1,-3,1,.05)
ISN 0078        CALL EFORM(YCOM(J),NXA,EXP)
ISN 0079        WRITE(6,326) NXA
ISN 0080    326 FORMAT( '  A = ',E14.7)
ISN 0081        CALL NUMBER(3.,6.5,.2,NXA,0.0,5)
ISN 0082        CALL SYMBL4(5.,6.5,.2,'E',0.0,1)
ISN 0083        CALL NUMBER(5.25,6.5,.2,EXP,0.0,-1)
ISN 0084        IF(J.GT.9) GO TO 9
ISN 0086        CALL SYMBL4(5.75,6.5,.2,'=',0.0,1)
ISN 0087        WRITE(6,327) YZRO(J)
ISN 0088    327 FORMAT( ' YZRO(J) = ' ,E14.7)
ISN 0089        NRSS = YZRO(J)
ISN 0090        CALL NUMBER(6.25,6.5,.2,NRSS,0.0,3)
ISN 0091        WRITE(6,222) NRSS
ISN 0092    222 FORMAT( '  NRSS = ',E14.7)
ISN 0093      9 CALL PLOT(12.,0.,-3)
ISN 0094    777 CONTINUE
ISN 0095        CALL EPLOT
ISN 0096        WRITE(6,98)
ISN 0097     98 FORMAT(14H0EPLOT CALLED      )
ISN 0098        DO 778 I=1,50
ISN 0099    778 WRITE(6,779)N1(I),N2(I)
ISN 0100    779 FORMAT( 2H0 2E20.7)
ISN 0101   1000 K=K+1
ISN 0102   2000 CONTINUE
ISN 0103        RETURN
ISN 0104        END
```

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002           SUBROUTINE DERIV(DY,Y,V,T)
ISN 0003           IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004           DIMENSION  ICNST(24),FCNST(28),Y(15),V(15),DY(15),ALP(18)
ISN 0005           COMMON  ICNST,FCNST,ALP
ISN 0006           COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0007           JSW = 1
             C     THIS PROGRAM INTEGRATES THE AMES EQUATIONS OF MOTION IN TIME
ISN 0008           IF(T.EQ.FCNST(4)) GO TO 58
ISN 0010           CALL AFX(JSW)
ISN 0011        58 DO 25 I=1,9
ISN 0012        25 DY(I) = 0.0
             C                                                    )9,1=I,)I(YD( )11,6(ETI RW
ISN 0013           DO 45 I=1,9
ISN 0014        45 DY(I) = DY(I) + F(I)
             C                                                    )9,1=I,)I(YD( )01,6(ETI RW
ISN 0015        10 FORMAT( ' DY 1 THRU 9   ' /  (1X,9E13.6) //)
ISN 0016        11 FORMAT( ' DY 1 THRU 9   ' /  (1X,9E13.6) //)
ISN 0017           T=T
             C                                  )9,1=I,)I(F(,)9,1=I,)I(Y( )001,6(ETI RW
ISN 0018       100 FORMAT( '   Y = '/ 9E13.6 / '   F(X) = ' / 9E13.6/)
ISN 0019           RETURN
ISN 0020           END
```

```
       COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002             SUBROUTINE AFX(JSW)
ISN 0003             IMPLICIT REAL*8 (A-H,O-Z)
               C     IFLAG(0,1)15(EXACT,ONLY M.V. NON-LINEARITY)
               C     *****************************************************************
ISN 0004             COMMON /BLK1/ PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI
ISN 0005             COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM
ISN 0006             COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C
ISN 0007             COMMON /BLK4/ HPHI,HTHT,HPSI
ISN 0008             COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0009             COMMON/BLK6/ IFLAG
ISN 0010             COMMON/BLK 11/DB1,DB2,DG1,DG2
ISN 0011              COMMON/BLK77/X(15)
ISN 0012             COMMON/BLK78/X1E,X4E,X7E

               C     *****************************************************************
               C
ISN 0015             WRITE(6,1070)PHI,VPHI,WPHI,THT,VTHT,WTHT,PSI,VPSI,WPSI,TM,T1,T2,
                    1XKM,XKC,A11,A13,D12,GAM1C,GAM2C,BET1C,BET2C,AITREN,HTHT,HPHI,HPSI
ISN 0016        1070 FORMAT(1H /10X,14H INITIAL STATE / 9E13.6 / 10X,13H INPUT CONSTS /
                    110X, 29H TM,T1,T2,XKM,XKC,A11,A13,D12 / 8E14.7 / 10X, 30H GAM1C,GA
                    2M2C,BET1C,BET2C,(RAD) / 4E14.7 / 10X,10H INERTIA = E14.7 ,/10X,
                    317H HTHT,HPHI,HPSI = 3E14.7//)
               C     *****************************************************************
ISN 0017             PI = 3.1415926
ISN 0018             DTR  = PI/180.0
ISN 0019             RTD  = 180.0/PI
               C     *****************************************************************
               C
ISN 0020             SB2C  = SIN(BET2C)
ISN 0021             CB2C  = COS(BET2C)
ISN 0022             TB2C = SB2C/CB2C
ISN 0023             SG1C  = SIN(GAM1C)
ISN 0024             CG1C  = COS(GAM1C)
ISN 0025             SG2C  = SIN(GAM2C)
ISN 0026             CG2C  = COS(GAM2C)
ISN 0027             SB1C  = SIN(BET1C)
ISN 0028             CB1C  = COS(BET1C)
ISN 0029             TB1C  = SB1C/CB1C
ISN 0030              SGAM1C = SG1C
ISN 0031              SGAM2C = SG2C
```

218

```
ISN 0032          CGAM1C = CG1C
ISN 0033          CGAM2C = CG2C
          C
          C       ******************************************************************
ISN 0034          DIF = GAM1C - GAM2C
ISN 0035          SDIF = SIN(DIF)
          C
          C       ******************************************************************
          C
          C       CALCULATION OF THE  A-MATRIX
          C
ISN 0036          DO 10 I=1,9
ISN 0037          DO 10 J=1,9
ISN 0038       10 A(I,J) = 0.0
          C
ISN 0039          A(1,2)=-10.0/AITREN
ISN 0040          A(7,8)=A(1,2)
ISN 0041          A(2,1)=XKM*XKC*(1.0+T1/T2)/(10.0*TM)
ISN 0042          A(2,2)=-1.0/TM
ISN 0043          A(5,5)=A(2,2)
ISN 0044          A(8,8)=A(5,5)
ISN 0045          A(3,3)=-1.0/T2
ISN 0046          A(6,6)=A(3,3)
ISN 0047          A(9,9)=-1/T2
ISN 0048          A(3,1)=-XKC*T1/T2**2
ISN 0049          A(2,3)=XKM/(10.0*TM)
ISN 0050          A(5,6)=A(2,3)
ISN 0051          A(8,9)=A(2,3)
ISN 0052          A(2,4) = A(2,1) *(-TB1C*CG1C)
ISN 0053          A(2,7)=A(2,1)*TB1C*SG1C
ISN 0054          A(3,4)=A(3,1)*A(2,4)/A(2,1)
ISN 0055          A(3,7)=A(3,1)*A(2,7)/A(2,1)
ISN 0056          A(4,5)=A(1,2)
ISN 0057          A(5,4) = A(2,1) * D12 * SDIF                        AFX
ISN 0058          A(6,4)=A(3,1)*A(5,4)/A(2,1)
ISN 0059          A(8,7)=A(5,4)
ISN 0060          A(9,7)=A(6,4)
          C
ISN 0061          WRITE(6,5000) ( (A(I,J),J=1,9),I=1,9)
ISN 0062     5000 FORMAT(1H / 1X,11H  A-MATRIX  /(1X, 9E13.6 ))
ISN 0063          VIC=XKC*T1/T2
          C
          C       ******************************************************************
          C
          C       CALCULATION  OF DB1,DB2,DG1,DG2
          C
          C       NEEDS ANGLES FROM STATE VECTOR AND COMMAND ANGLES AS INPUT
          C
          C
          C
ISN 0064      312 PHI= X(1) + X1E
ISN 0065          THT= X(4) + X4E
ISN 0066          PSI= X(7) + X7E
ISN 0067          SPHI = SIN(PHI)
ISN 0068          CPHI = COS(PHI)
ISN 0069          STHT = SIN(THT)
```

```
ISN 0070          CTHT = COS(THT)
ISN 0071          SPSI = SIN(PSI)
ISN 0072          CPSI = COS(PSI)
ISN 0073          TTHT = STHT/CTHT                                           AFX
ISN 0074          XNORW = XKC*T1/T2                                          AFX
ISN 0075          PUP = 1.0/(XKC*TM)                                         AFX
            C
ISN 0076          R1=TB1C/CG1C
ISN 0077          R2=TB2C/CG2C
ISN 0078          TG1=SG1C/CG1C
ISN 0079          TPHI=SPHI/CPHI
ISN 0080          IF (ABS(X(4))-0.15)9991,9991,9992
ISN 0081     9991 B4=-0.5*X(4)**2*(1.0-X(4)**2/12.0)
ISN 0082          GO TO 9993
ISN 0083     9992 B4=CTHT-1.0
ISN 0084     9993 IF (ABS(X(7))-0.15) 9994,9994,9995
ISN 0085     9994 B7=-0.5*X(7)**2*(1.0-X(7)**2/12.0)
ISN 0086          GO TO 9996
ISN 0087     9995 B7=CPSI-1.0
ISN 0088     9996 EALF=B4*TG1-(SPSI*TPHI+CPSI*STHT)*R1+CPSI*TPHI-SPSI*STHT
ISN 0089          EBET=B7+SPSI*STHT*TPHI-(SPSI-TPHI*STHT*CPSI)*R1-TPHI*CTHT*TG1
ISN 0090          DG1=ATAN((EALF-EBET*TG1)/(1.0+EBET+(TG1+EALF)*TG1))
ISN 0091          G1=DG1+GAM1C
ISN 0092          EALF=B4*TG2+(SPSI*TPHI+CPSI*STHT)*R2+CPSI*TPHI-SPSI*STHT
ISN 0093          EBET=B7+SPSI*STHT*TPHI+(SPSI-TPHI*STHT*CPSI)*R2-TPHI*CTHT*TG2
ISN 0094          TG2=SG2C/CG2C
ISN 0095          DG2=ATAN((EALF-EBET*TG2)/(1.0+EBET+(TG2+EALF)*TG2))
ISN 0096          G2=DG2+GAM2C
ISN 0097          R=CPSI*CTHT*SB1C+SPSI*CTHT*CG1C*CB1C+STHT*SG1C*CB1C
ISN 0098          DB1=ASIN(R)
ISN 0099          DB1=DB1-BET1C
ISN 0100          IF (ABS(DB1)-0.1) 9997,9998,9998
ISN 0101     9997 DB1=DBET(B4,B7,TB1C,STHT,SPSI,CG1C,SG1C,+1.0)
ISN 0102     9998 R=CPSI*CTHT*SB2C-SPSI*CTHT*CG2C*CB2C-STHT*SG2C*CB2C
ISN 0103          DB2=ASIN(R)
ISN 0104          DB2=DB2-BET2C
ISN 0105          IF (ABS(DB2)-0.1) 9999,10009,10009
ISN 0106     9999 DB2=DBET(B4,B7,TB2C,STHT,SPSI,CG2C,SG2C,-1.0)
ISN 0107    10009 CONTINUE
            C
            C     ****************************************************************
            C
            C     CALCULATION OF NONLINEAR F(X)
            C
            C
ISN 0108          SUM2=SUM4/T2
ISN 0109          SUM3=AITREN/(10.0*TM)
ISN 0110          SUM4=AITREN*T1/(T2*XKM)
ISN 0111          SUM5 =(COS(DG2&GAM2C))* DB1 & (COS(DG1 & GAM1C)) * DB2
ISN 0112          SUM6=-(SIN(DG2&GAM2C))* DB1 - (SIN(DG1 & GAM1C)) * DB2
ISN 0113          GAIN = XKC * (T1&T2)/T2                                    AFX
ISN 0114          IF(IFLAG) 10019,10019,10029
ISN 0115    10019 CONTINUE
ISN 0116          F(1)=A(1,2)*(X(2)+TTHT*(SPHI*X(5)+CPHI*X(8)))
            C
ISN 0117          ARGF2=GAIN*DG1+X(3)-SUM4*HPHI
```

```
ISN 0118          IF (ABS(ARGF2).LE.F2LIM) F2 = ARGF2
ISN 0120          IF (ARGF2.GT.F2LIM) F2 = F2LIM
ISN 0122          IF (ARGF2.LT.-F2LIM) F2= -F2LIM
ISN 0124          F(2)=A(2,2)*X(2)+A(2,3)*F2-SUM3*HPHI
            C
ISN 0125          F(3)=A(3,3)*X(3)+A(3,1)*DG1+SUM2*HPHI
ISN 0126          F(4)=A(1,2)*(CPHI*X(5)-SPHI*X(8))
            C
ISN 0127          ARGF5=GAIN*D12*SUM5+X(6)-SUM4*HTHT
ISN 0128          IF(ABS(ARGF5).LE.F2LIM) F2 =ARGF5
ISN 0130          IF(ARGF5.GT.F2LIM) F2=F2LIM
ISN 0132          IF(ARGF5 .LT. -F2LIM)F2 = -F2LIM
ISN 0134          F(5)=A(2,2)*X(5)+A(2,3)*F2-SUM3*HTHT
            C
ISN 0135          F(6)=A(3,3)*X(6)+A(3,1)*D12*SUM5+SUM2*HTHT
            C
ISN 0136          F(7)=A(1,2)*(SPHI*X(5)+CPHI*X(8))/CTHT
            C
ISN 0137          ARGF8=+GAIN*D12*SUM6+X(9)-SUM4*HPSI
ISN 0138          IF(ABS(ARGF8).LE.F2LIM) F2 = ARGF8
ISN 0140          IF(ARGF8.GT.F2LIM) F2 = F2LIM
ISN 0142          IF(ARGF8.LT.-F2LIM) F2=-F2LIM
ISN 0144          F(8)=A(2,2)*X(8)+A(2,3)*F2-SUM3*HPSI
            C
ISN 0145          F(9)=A(3,3)*X(9)+A(3,1)*D12*SUM6+SUM2*HPSI
            C
ISN 0146          IF( JSW.GT.0) GO TO 10030
ISN 0148          WRITE(6,5003)IFLAG
ISN 0149     5003 FORMAT(' IFLAG=',I5,10X,'ALL NONLINEARLITIES'/)
ISN 0150    10030 GO TO 10039
            C
ISN 0151    10029 CONTINUE
ISN 0152          ARGF2=GAIN*EPHI+X(3)-SUM4*HPHI
ISN 0153          IF (ABS(ARGF2).LE.F2LIM) F2 = ARGF2
ISN 0155          IF (ARGF2.GT.F2LIM) F2 = F2LIM
ISN 0157          IF (ARGF2.LT.-F2LIM) F2= -F2LIM
ISN 0159          F(2)=A(2,2)*X(2)+A(2,3)*F2-SUM3*HPHI
ISN 0160          F(3)=A(3,3)*X(3)+A(3,1)*EPHI
ISN 0161          ARGF5=GAIN*ETHT+X(6)-SUM4*HTHT
ISN 0162          IF(ABS(ARGF5).LE.F2LIM) F2 =ARGF5
ISN 0164          IF(ARGF5.GT.F2LIM) F2=F2LIM
ISN 0166          IF(ARGF5 .LT. -F2LIM)F2 = -F2LIM
ISN 0168          F(5)=A(2,2)*X(5)+A(2,3)*F2-SUM3*HTHT
ISN 0169          F(6)=A(3,3)*X(6)+A(3,1)*ETHT
ISN 0170          ARGF8=GAIN*EPSI+X(9)-SUM4*HPSI
ISN 0171          IF(ABS(ARGF8).LE.F2LIM) F2 = ARGF8
ISN 0173          IF(ARGF8.GT.F2LIM) F2 = F2LIM
ISN 0175          IF(ARGF8.LT.-F2LIM) F2=-F2LIM
ISN 0177          F(8)=A(2,2)*X(8)+A(2,3)*F2-SUM3*HPSI
ISN 0178          F(9)=A(3,3)*X(9)+A(3,1)*EPSI
            C
ISN 0179          IF( JSW.GT.0) GO TO 10039
ISN 0181          WRITE(6,5002)IFLAG
ISN 0182     5002 FORMAT('IFLAG=',I5,10X,'MCTOR VOLTAGE ONLY NONLINEARITY'/)
ISN 0183    10039 CONTINUE
            C     ******************************************************************
```

```
              C
ISN 0184            IF(JSW.GT.0)GO TO 602
ISN 0186            WRITE(6.5001) (F(I),I=1,9)
ISN 0187       5001 FORMAT(1H / 1X,12H F(X)-VECTOR / 1X,9E13.6)
ISN 0188        602 CONTINUE
              C
              C
              C     ******************************************************************
ISN 0189            RETURN
ISN 0190            END
```

/ AFX /

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	C	R*8	000000	F	C	R*8	000798	I		I*4	000088	J		I*4	00008C
Q	C	R*8	N.R.	R		R*8	000098	X	C	R*8	0000C0	B4		R*8	0000A0
B7		R*8	0000A8	DF	C	R*8	N.R.	F2		R*8	000080	G1		R*8	0000B8
G2		R*8	0000C0	PI		R*8	0000C8	PM	C	R*8	N.R.	R1		R*8	0000D0
R2		R*8	0000D8	TM	C	R*8	000000	T1	C	R*8	000008	T2	C	R*8	000010
AFX		R*8	N.R.	A11	C	R*8	000028	A13	C	R*8	000030	DB1	C	R*8	000000
DB2	C	R*8	000008	DG1	C	R*8	000010	DG2	C	R*8	000018	DIF		R*8	0000E0
DTR		R*8	0000E8	D12	C	R*8	000038	JSW		I*4	000090	PHI	C	R*8	000000
PSI	C	R*8	000030	PUP		R*8	0000F0	RTD		R*8	0000F8	TG1		R*8	000100
TG2		R*8	000108	THT	C	R*8	000018	VIC		R*8	000110	XKC	C	R*8	000020
XKM	C	R*8	000018	X1E	C	R*8	000000	X4E	C	R*8	000008	X7E	C	R*8	000010
ASIN	XF	R*8	000000	CB1C		R*8	000118	CB2C		R*8	000120	CG1C		R*8	000128
CG2C		R*8	000130	CPHI		R*8	000138	CPSI		R*8	000140	CTHT		R*8	000148
DBET	XF	R*8	000000	EALF		R*8	000150	EBET		R*8	000158	EPHI		R*8	000160
EPSI		R*8	000168	ETHT		R*8	000170	GAIN		R*8	000178	HPHI	C	R*8	000000
HPSI	C	R*8	000010	HTHT	C	R*8	000008	SB1C		R*8	000180	SB2C		R*8	000188
SDIF		R*8	000190	SG1C		R*8	000198	SG2C		R*8	0001A0	SPHI		R*8	0001A8
SPSI		R*8	0001B0	STHT		R*8	0001B8	SUM2		R*8	0001C0	SUM3		R*8	0001C8
SUM4		R*8	0001D0	SUM5		R*8	0001D8	SUM6		R*8	0001E0	TB1C		R*8	0001E8
TB2C		R*8	0001F0	TPHI		R*8	0001F8	TTHT		R*8	000200	VPHI	C	R*8	000008
VPSI	C	R*8	000038	VTHT	C	R*8	000020	WPHI	C	R*8	000010	WPSI	C	R*8	000040
WTHT	C	R*8	000028	XNORW		R*8	000208	IFLAG	C	I*4	000000	GAM1C	C	R*8	000000
GAM2C	C	R*8	000008	F2LIM	C	R*8	000048	BET1C	C	R*8	000010	BET2C	C	R*8	000018
ARGF2		R*8	000210	ARGF5		R*8	000218	ARGF8		R*8	000220	IBCOM=	XF	I*4	000000
SGAM1C		R*8	000228	SGAM2C		R*8	000230	CGAM1C		R*8	000238	CGAM2C		R*8	000240
AITREN	C	R*8	000040	DSIN	IF		000000	DCOS	IF		000000	DATAN	IF		000000

COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```
ISN 0002          FUNCTION DBET(B4,B7,TB,STHT,SPSI,CG,SG,SIGN)
ISN 0003          XKAP=(B7+B4+B7*B4)*TB+((1.0+B4)*SPSI*CG+STHT*SG)*SIGN
ISN 0004          XMU=XKAP
ISN 0005        2 SQ=XMU*XMU
ISN 0006          XF=XMU-XKAP-TB*0.5*SQ*(1.0+0.25*SQ*(1.0+0.5*SQ))
ISN 0007          IF (XF*XMU.EQ.0.0) GO TO 1
ISN 0009          XMU=XMU-XF/(1.0-TB*XMU*(1.0+0.5*SQ*(1.0+0.75*SQ)))
ISN 0010          IF (ABS(XF/XMU).GE.1.0E-06) GO TO 2
ISN 0012        1 CONTINUE
ISN 0013          DBET=ASIN(XMU)
ISN 0014          RETURN
ISN 0015          END
```

```
LEVEL  2 FEB 67                                    OS/360   FORTRAN H                                     DATE  69.179/08.14.47

          COMPILER OPTIONS - NAME=  $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NCDECK,LOAD,MAP,NOEDIT,ID
   ISN 0002              SUBROUTINE PEAIG
   ISN 0003              IMPLICIT REAL*8 (A-H,O-Z)
                   C
   ISN 0004              COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
   ISN 0005              N = 9
                   C
                   C
   ISN 0006              DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP
   ISN 0007              DIMENSION AAMOD(82,82),P(81),QV(81),E(9,9),AM(9,9)
                        1 ,ISTEP(82),ATP(9,9),FA(9,9),QP(9,9)
                   C
                   C     N =  DIMENSION OF A - MATRIX (INPUT ON CARD NO.2 )
                   C     A =  INPUT MATRIX
                   C
                   C
                   C
                   C
                   C     INITIALIZE  PM , QP, ATP , PA , AM , E
                   C
                   C
                   C
   ISN 0008              NN = N*N
   ISN 0009              DO 37 I=1,N
   ISN 0010              DO 37 J=1,N
   ISN 0011              PM(I,J) = 0.0
   ISN 0012              QP(I,J) = 0.0
   ISN 0013              ATP(I,J)= 0.0
   ISN 0014              PA(I,J) = 0.0
   ISN 0015              AM(I,J) = 0.0
   ISN 0016           37 E(I,J)  = 0.0
                   C                                            )N,1=I,)N,1=J,)J,I(A(()3692,6(ETI RW 14
   ISN 0017         2963 FORMAT(1H /(1X,9E14.7) )
                   C
                   C
                   C     MAKING Q PERFECTLY SYMMETRIC
   ISN 0018               NK = 2
   ISN 0019           42 DO 76  JJ = 1,8
   ISN 0020               Q(NK,JJ) = Q(JJ,NK)
   ISN 0021              NOOK = NK-1
   ISN 0022               IF(NOOK  .EQ. JJ) GO TO 77
   ISN 0024           76  CONTINUE
   ISN 0025           77  IF  (NK.EQ.9) GO TO 87
   ISN 0027               NK = NK & 1
   ISN 0028               GO TO 42
   ISN 0029           87  CONTINUE
                   C
                   C     SETTING Q-MATRIX TO Q-VECTOR
                   C
   ISN 0030              IA=1
   ISN 0031              DO 62 I=1,N
   ISN 0032              DO 62 J=1,N
   ISN 0033              QV(IA) = Q(I,J)
   ISN 0034           62 IA=IA&1
   ISN 0035              WRITE(6,206)
   ISN 0036          206 FORMAT(1H /  1X, 12H  Q-VECTOR    // )
```

```
ISN 0037          WRITE(6,2963)(QV(IP),IP=1,NN)
            C
            C
            C
            C     THIS SECTION CALCULATES THE AAMOD MATRIX
            C     WHICH IS GIVEN BY
            C           XXXX                                  XXXX
            C           X                                        X
            C           X AT& A11.I     A21.I       A31.I       X
            C           X                                        X
            C           X  A12.I       AT& A22.I    A32.I       X
            C           X                                        X
            C           X  A13.I        A23.I     AT& A33.I     X
            C           X                                        X
            C           XXXX                                  XXXX
            C
            C
            C     INITIALIZATION OF AAMOD
            C
            C
ISN 0038          DO 10  L = 1,NN
ISN 0039          DO 10  LA= 1,NN
ISN 0040       10 AAMOD(L,LA) = 0.0
            C
            C     DEFINE UNIT MATRIX OF ORDER  N ,CALL IT E
            C
ISN 0041          DO 30 MAA = 1,N
ISN 0042       30 E(MAA,MAA) = 1.0
            C
            C
            C
            C
            C     THIS SECTION CALCULATES THE SECTION OF THE  AMOD
            C     MATRIX WHICH IS COMPRISED OF  A(J,I) * E
ISN 0043          IP =  -N
ISN 0044          DO 50  MR=1,N
ISN 0045          IP = IP & N
ISN 0046          IPP = -N
ISN 0047          DO 50  JR =1,N
ISN 0048          IPP = IPP &N
ISN 0049          DO 40  K= 1,N
ISN 0050          DO 40 KA= 1,N
ISN 0051          AM(K,KA) = A(JR,MR) * E(K,KA)
ISN 0052          KPIP = K&IP
ISN 0053          KAP  = KA&IPP
ISN 0054       40 AAMOD(KPIP,KAP)=AM(K,KA)
ISN 0055       50 CONTINUE
            C
            C
            C
            C
            C
            C     THIS SECTION ADDS THE A MATRIX TO THE DIAGONAL NXN
            C     ELEMENTS OF AAMOD
            C
            C
```

```
ISN 0056          DO 60   K = 1,N
ISN 0057          IP  = (K-1)* N
ISN 0058          DO 55  LT = 1,N
ISN 0059          DO 55  LM = 1,N
ISN 0060          ILT = IP&LT
ISN 0061          IPM = IP&LM
ISN 0062       55 AAMOD(ILT,IPM)= AAMOD(ILT,IPM)& A(LM,LT)
ISN 0063       60 CONTINUE
            C
            C     THAT FINISHES THE CALCULATION OF AMOD ,NOW WE MUST
            C     PRINT IT OUT BECAUSE SREVNI WIPES  OUT AAMOD
            C
            C                                                        )002,6(ETI RW
ISN 0064      200 FORMAT(1H /   1X,16H AAMOD - MATRIX  // )
            C                                  )NN,1=I,)NN,1=J,)J,I(DOMAA()3692,6(ETI RW
            C     NOW FOR A-INVERSE
            C
ISN 0065          IDEM=NN&1
            C
ISN 0066          CALL MINVD(AAMOD,IDEM,NN,ISTEP,IERR)
            C
            C
            C     NOW AAMOD INVERSE HAS REPLACED AAMOD
            C
            C                                                        )192,6(ETI RW
ISN 0067      291 FORMAT(1H / 1X,22HAAMOD-INVERSE  MATRIX  , //)
            C                                                   NN,1=I 371  OD
            C                                   )NN,1=J,)J,I(DOMAA()3692,6(ETI RW 371
ISN 0068          DO 70  IB = 1,NN
ISN 0069          P(IB) =0.0
ISN 0070          DO 70  IC = 1,NN
ISN 0071       70 P(IB) =  P(IB) -AAMOD(IB,IC) * QV(IC)
ISN 0072          WRITE(6,295)
ISN 0073      295 FORMAT(1H / 1X,14H  P-MATRIX    ,//)
ISN 0074          WRITE(6,2963)  (P(IB),IB=1,NN)
            C     SET P-VECTOR TO P-MATRIX TO GET Q-PRIME FROM-ATP-PA =QP
ISN 0075          K=1
ISN 0076          DO25 I=1,N
ISN 0077          DO25 J=1,N
ISN 0078          PM(I,J) = P(K)
ISN 0079       25 K=K&1
            C     CALCULATION OF ATP (A TRANSPOSE P )
            C     CALCULATION OF PA  (P-MATRIX X A )
ISN 0080          DO 26 I=1,N
ISN 0081          DO 26 J=1,N
ISN 0082          DO 26 K=1,N
ISN 0083          ATP(I,J) = ATP(I,J) & A(K,I) *PM(K,J)
ISN 0084       26 PA(I,J)  = PA(I,J)  &PM(I,K) * A(K,J)
ISN 0085          DO 27 I=1,N
ISN 0086          DO 27 J=1,N
ISN 0087       27 QP(I,J) = -ATP(I,J) - PA(I,J)
ISN 0088          WRITE(6,9765)
ISN 0089     9765 FORMAT(1H / 1X,36H  Q FROM PUTTING P INTO -ATP-PA = Q , //)
ISN 0090          DO941 I=1,N
ISN 0091      941 WRITE(6,2963)  (QP(I,J),J=1,N)
ISN 0092          RETURN
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
      ISN 0002                 SUBROUTINE MINVD(A,IDIM,N,ISTEP,IERR)
      ISN 0003                 IMPLICIT REAL*8 (A-H,O-Z)
                    C                   MATRIX INVERSION            DOUBLE PRECISION
      ISN 0004                 DOUBLE PRECISION A,ABSAI,ABSAL,TEMP,FAC,ABSA,DABS
      ISN 0005                 DIMENSION A(IDIM,1),ISTEP(1)
      ISN 0006                 K=1
      ISN 0007                 IERR=0
      ISN 0008                 NP1=N&1
      ISN 0009              30 LL=1
      ISN 0010                 DO 35 J=1,N
      ISN 0011              35 A(J,NP1)=A(J,1)
      ISN 0012              40 I=1
      ISN 0013                 L=2
      ISN 0014              45 ABSAI=DABS(A(I,1))
      ISN 0015                 ABSAL=DABS(A(L,1))
      ISN 0016                 IF(ABSAI-ABSAL)50,55,55
      ISN 0017              50 I=L
      ISN 0018              55 IF(L-N)60,56,56
      ISN 0019              56 IF(A(I,1))65,85,65
      ISN 0020              60 L=L&1
      ISN 0021                 GO TO 45
      ISN 0022              65 IF(K-1)70,90,70
      ISN 0023              70 M=1
      ISN 0024              75 IF(I-ISTEP(M))80,84,80
      ISN 0025              80 IF(M-K&1)81,82,82
      ISN 0026              81 M=M&1
      ISN 0027                 GO TO 75
      ISN 0028              82 DO 83 J=1,N
      ISN 0029              83 A(J,1)=A(J,NP1)
      ISN 0030                 GO TO 90
      ISN 0031              84 IF(LL-N)86,85,85
      ISN 0032              85 IERR=1
      ISN 0033                 GO TO 610
      ISN 0034              86 LL=LL&1
      ISN 0035                 A(I,1)=0.D0
      ISN 0036                 GO TO 40
      ISN 0037              90 ISTEP(K)=I
      ISN 0038                 J=1
      ISN 0039             100 IF(J-I)110,120,110
      ISN 0040             110 A(J,NP1)=0.D0
      ISN 0041                 GO TO 130
      ISN 0042             120 A(J,NP1)=1.D0
      ISN 0043             130 IF(J-N)140,150,150
      ISN 0044             140 J=J&1
      ISN 0045                 GO TO 100
      ISN 0046             150 J=1
      ISN 0047                 TEMP=A(I,1)
      ISN 0048             160 A(I,J)=A(I,J)/TEMP
      ISN 0049                 IF(J-NP1)170,180,180
      ISN 0050             170 J=J&1
      ISN 0051                 GO TO 160
      ISN 0052             180 J=1
      ISN 0053             190 IF(J-I)200,290,200
      ISN 0054             200 IF(A(J,1)-1.D0)230,210,230
      ISN 0055             210 DO 220 M=1,NP1
```

```
ISN 0056          A(J,M)=A(J,M)-A(I,M)
ISN 0057      220 CONTINUE
ISN 0058          GO TO 290
ISN 0059      230 IF(A(J,1)&1.D0)260,240,260
ISN 0060      240 DO 250  M=1,NP1
ISN 0061          A(J,M)=A(J,M)&A(I,M)
ISN 0062      250 CONTINUE
ISN 0063          GO TO 290
ISN 0064      260 IF(A(J,1))270,290,270
ISN 0065      270 FAC=A(J,1)
ISN 0066          DO 280  M=1,NP1
ISN 0067      280 A(J,M)=A(J,M)-A(I,M)*FAC
ISN 0068      290 IF(J-N)300,340,340
ISN 0069      300 J=J&1
ISN 0070          GO TO 190
ISN 0071      340 DO 350  J=1,N
ISN 0072          DO 350  M=1,N
ISN 0073          MP1=M&1
ISN 0074      350 A(J,M)=A(J,MP1)
ISN 0075          IF(K-N)360,390,390
ISN 0076      360 K=K&1
ISN 0077          GO TO 30
ISN 0078      390 DO 400  J=1,N
ISN 0079      400 A(NP1,J)=ISTEP(J)
ISN 0080          M=1
ISN 0081      410 I=ISTEP(M)
ISN 0082          IF(I-M)420,470,420
ISN 0083      420 DO 430  J=1,N
ISN 0084          TEMP=A(M,J)
ISN 0085          A(M,J)=A(I,J)
ISN 0086      430 A(I,J)=TEMP
ISN 0087          J=M
ISN 0088      440 IF(M-ISTEP(J))450,460,450
ISN 0089      450 J=J&1
ISN 0090          GO TO 440
ISN 0091      460 ISTEP(J)=I
ISN 0092      470 IF(M-N)480,490,490
ISN 0093      480 M=M&1
ISN 0094          GO TO 410
ISN 0095      490 DO 500  J=1,N
ISN 0096      500 ISTEP(J)=A(NP1,J)
ISN 0097      530 M=1
ISN 0098      540 I=ISTEP(M)
ISN 0099          IF(I-M)550,570,550
ISN 0100      550 DO 560  J=1,N
ISN 0101          TEMP=A(J,I)
ISN 0102          A(J,I)=A(J,M)
ISN 0103      560 A(J,M)=TEMP
ISN 0104          J=ISTEP(M)
ISN 0105          ISTEP(M)=ISTEP(J)
ISN 0106          ISTEP(J)=J
ISN 0107          GO TO 540
ISN 0108      570 IF(M-N)580,610,610
ISN 0109      580 M=M&1
ISN 0110          GO TO 540
ISN 0111      610 RETURN
```

```
        COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002          SUBROUTINE QGEN
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
            C
            C     GENERATION OF POSITIVE DEFINITE Q MATRIX
ISN 0004          DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP
ISN 0005          COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0006          N=9
            C
ISN 0007          COMMON/A/   THETA(28),PHIV(8),XLAM(9),ZTHETA(28),
                1             BA(20,20),SS(20,20,20),CC(20,20),Z(20,20,20),
                2             BM(20,20),SM(20,20),AAR(20,20),G(20,20),QQ(20,20),
                3             TTHETA(28)
            C          )9,1=K,)K(MALX(,)8,1=J,)J(VIHP(,)82,1=I,)I(ATEHT()3601,5(DA ER
ISN 0008    1063 FORMAT(6E12.4)
ISN 0009          PI = 3.1415926
ISN 0010          PI2 = PI/2.
ISN 0011          DO 206 I=1,28
ISN 0012          THETA(I) = RDM(DUM)
ISN 0013      206 THETA(I) = -PI2 + THETA(I)*PI
ISN 0014          DO 207 I=1,8
ISN 0015          PHIV(I) = RDM(DUM)
ISN 0016      207 PHIV(I) = -PI + PHIV(I)* 2.* PI
ISN 0017          DO 208 I=1,9
ISN 0018          XLAM(I) = RDM(DUM)
ISN 0019      208 XLAM(I) = 0. + XLAM(I) * 100.
ISN 0020          WRITE(6,3) THETA,PHIV,XLAM
ISN 0021        3 FORMAT(23H DATA-THETA,PHIV,XLAM    /(1X,9E14.7))
ISN 0022          NN=(N-1)*(N-2)/2
ISN 0023          DO 6I=1,NN
ISN 0024          BAD=THETA(I)
ISN 0025        6 TTHETA(I)= AMOD(BAD,PI2)
            C     WE HAVE NOW INDEXED THETA.
            C     NOW WANT CONTINUED PRODUCT OF SS(I,J,L) FOR L=K&1,N .
            C     FOR EACH K=1,N-1 OBTAIN Z(K,I,J).
ISN 0026           NNI = N-1
ISN 0027       69 DO 20 K=1,NNI
            C
ISN 0028          DO 8 I=1,N
ISN 0029          DO 8 J=1,N
ISN 0030        8 BA(I,J)=0.0
ISN 0031          DO 99 I=1,N
ISN 0032       99 BA(I,I)=1.0
            C
ISN 0033          KK=K&1
ISN 0034          DO 10 L=KK,N
            C
ISN 0035          DO 15 I=1,N
ISN 0036          DO 15 J=1,N
ISN 0037       15 SS(I,J,L)=0.0
ISN 0038          DO 98 I=1,N
ISN 0039       98 SS(I,I,L)=1.0
            C     WE DEVELOP SS(I,J,L) AS FUNCTION THETA(L,K,N) FOR L L.T. N
            C     AND SS(I,J,L) FUNCTION OF PHIV(K) FOR L=N
ISN 0040          IF(L-N)25,23,23
ISN 0041       25 M=((2*N -K-2)*(K-1)/2)&N-L
```

```
ISN 0042          SS(K,K,L)=COS(TTHETA(M))
ISN 0043          SS(L,L,L)=COS(TTHETA(M))
ISN 0044          SS(K,L,L)=-SIN(TTHETA(M))
ISN 0045          SS(L,K,L)=SIN(TTHETA(M))
ISN 0046          GO TO 35
ISN 0047       23 SS(K,K,L)=COS(PHIV(K))
ISN 0048          SS(L,L,L)=COS(PHIV(K))
ISN 0049          SS(K,L,L)=-SIN(PHIV(K))
ISN 0050          SS(L,K,L)=SIN(PHIV(K))
            C
ISN 0051       35 DO 70 I=1,N
ISN 0052          DO 70 J=1,N
ISN 0053       70 CC(I,J)=0.0
            C
ISN 0054          DO 50 M=1,N
ISN 0055          DO 50 J=1,N
ISN 0056          DO 50 I=1,N
ISN 0057       50 CC(M,J)=BA(M,I)*SS(I,J,L) &CC(M,J)
ISN 0058          DO110 I=1,N
ISN 0059          DO110 J=1,N
ISN 0060      110 BA(I,J)=CC(I,J)
ISN 0061       10 CONTINUE
ISN 0062          DO 20 I=1,N
ISN 0063          DO 20 J=1,N
ISN 0064       20 Z(K,I,J)=BA(I,J)
            C
ISN 0065          DO 7 I=1,N
ISN 0066          DO 7 J=1,N
ISN 0067        7 BM(I,J)=0.0
ISN 0068          DO 16 I=1,N
ISN 0069       16 BM(I,I)=1.0
            C
ISN 0070          DO 40 K=1,NNI
            C
ISN 0071          DO 75 I=1,N
ISN 0072          DO 75 J=1,N
ISN 0073       75 SM(I,J)=0.0
            C
ISN 0074          DO 55 M=1,N
ISN 0075          DO 55 J=1,N
ISN 0076          DO 55 I=1,N
ISN 0077       55 SM(M,J)=Z(K,M,I)*BM(I,J)&SM(M,J)
ISN 0078          DO 40 I=1,N
ISN 0079          DO 40 J=1,N
ISN 0080       40 BM(I,J)=SM(I,J)
            C
            C     BM(I,J) IS CONTINUED PRODUCT OF Z(K,I,J) FROM K=1 TO N-1
            C
ISN 0081          IF(PP)41,41,19
ISN 0082       19 CONTINUE
ISN 0083       18 FORMAT(8H BM(I,J)/(6E15.7))
ISN 0084       41 DO 78 I=1,N
ISN 0085          DO 78 J=1,N
ISN 0086       78 AAR(I,J)=BM(J,I)
            C
            C     AAR(I,J) IS TRANSPOSE BM(I,J)
```

```
C
      DO 82 I=1,N
      DO 82 J=1,N
   82 G(I,J)=0.0
      DO 85 I=1,N
   85 G(I,I)=XLAM(I)
C     G(I,J) IS THE LAMDA MATRIX
C
      DO 86 I=1,N
      DO 86 J=1,N
   86 QQ(I,J)=0.0
      DO 88 I=1,N
      DO 88 J=1,N
      DO 88 M=1,N
   88 QQ(I,J)=G(I,M)*BM(M,J)&QQ(I,J)
C
C     QQ(I,J)=LAMDA MATRIX *BM(I,J)
C
      DO 90 I=1,N
      DO 90 J=1,N
   90 Q(I,J)=0.0
      DO 95 I=1,N
      DO 95 J=1,N
      DO 95 M=1,N
   95 Q(I,J)=AAR(I,M)*QQ(M,J)&Q(I,J)
      WRITE(6,93) (( Q(I,J),J=1,N),I=1,N)
   93 FORMAT(1H /1X,7H Q(I,J)/(1X,9E14.7) )
      RETURN
      END
```

```
LEVEL 2 FEB 67                              OS/360  FORTRAN H                                  DATE  69.179/08.15.32

          COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
   ISN 0002            SUBROUTINE VLAP(PX,VL)
   ISN 0003            IMPLICIT REAL*8 (A-H,O-Z)
   ISN 0004            DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP
   ISN 0005            COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
   ISN 0006             COMMON/BLK77/X(15)
   ISN 0007            COMMON/BLKV/ SL
   ISN 0008            DIMENSION PX(9)
   ISN 0009            VL=0.0                                                            VL 40
   ISN 0010            DO 250 I=1,9                                                      VL 50
   ISN 0011        250 PX(I)=0.0                                                         VL 60
   ISN 0012            DO 251 I=1,9                                                      VL 70
   ISN 0013            DO 251 J=1,9                                                      VL 80
   ISN 0014        251 PX(I) = PX(I) & PM(I,J)*X(J)                                      VL 90
   ISN 0015            DO 252 I=1,9                                                      VL100
   ISN 0016        252 VL = VL & X(I)*PX(I)                                              VL110
                C                                                           LV )352,6(ETI RW
   ISN 0017        253 FORMAT(1H  10X, 5H VL = ,E14.7 )
   ISN 0018            SL = VL
   ISN 0019            RETURN                                                            VL120
   ISN 0020            END                                                               VL130
```

```
      COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

ISN 0002        SUBROUTINE VDOTA(PF,QX,VDCT)
ISN 0003        IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004        DOUBLE PRECISION AAMOD,P,QV,A,E,AM,Q,PM,ATP,PA,QP
ISN 0005        COMMON /BLK5/ A(9,9),Q(9,9),PM(9,9),F(9),DF(9,9)
ISN 0006        COMMON /BLK17/ EPS,XK12
ISN 0007         COMMON/BLK77/X(15)
ISN 0008         DIMENSION PF(9),QX(9)
ISN 0009        SPE = 0.
ISN 0010        VDOT =0.0                                                     VDOT  40
ISN 0011        DO 250 I=1,9                                                  VDOT  50
ISN 0012        PF(I)=0.0                                                     VDOT  60
ISN 0013    250 QX(I)=0.0                                                     VDOT  70
ISN 0014        DO 251 I=1,9                                                  VDOT  80
ISN 0015        DO 251 J=1,9                                                  VDOT  90
ISN 0016        QX(I)= QX(I) & Q(I,J)*X(J)                                    VDOT100
ISN 0017    251 PF(I)=PF(I) & PM(I,J)*F(J)                                    VDOT110
ISN 0018        DO 252 I=1,9                                                  VDOT120
ISN 0019    252 VDOT = VDOT - X(I)*(QX(I) - 2.0 * PF(I) )
ISN 0020        DO 253 I=1,9
ISN 0021    253 SPE = X(I)*QX(I) + SPE
ISN 0022        EPS = .05*ABS(SPE)
            C041 TODV                                          TODV )0531,6(ETI RW
            C                                                   EPS )5531,6(ETI RW
ISN 0023    355 FORMAT( 6H XQX= 1E14.7)
ISN 0024    350 FORMAT(1H  1X,26H LIAPUNCV FCT DERIV =      ,E14.7   )
ISN 0025        RETURN                                                        VDOT160
ISN 0026        END                                                           VDOT170
```

```
COMPILER OPTIONS - NAME= $MAIN,OPT=02,LINECNT=56,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

ISN 0002          SUBROUTINE AFX(JSW)
ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
            C     ****************************************************************
ISN 0004          COMMON /BLK1/ PHI,VPHI,THT,VTHT,PSI,VPSI
ISN 0005          COMMON /BLK2/ TM,T1,T2,XKM,XKC,A11,A13,D12,AITREN,F2LIM
ISN 0006          COMMON /BLK3/ GAM1C,GAM2C,BET1C,BET2C
ISN 0007          COMMON /BLK4/ HPHI,HTHT,HPSI
ISN 0008          COMMON /BLK5/ A(6,6),Q(6,6),PM(6,6),F(6),DF(6,6)
ISN 0009          COMMON/BLK 11/DB1,DB2,DG1,DG2
ISN 0010           COMMON/BLK77/X(15)
ISN 0011          COMMON/BLK78/X1E,X3E,X5E
            C
            C     EXACT MODEL STATE EQUATIONS
            C
ISN 0012          IF(JSW.GT.0)GO TO 312
            C      ( TRACKERS 3-4   (AMES 1-2)   )
            C
            C
            C     WHERE X IS A  NINE COMPONENT COLUMN VECTOR   AS  IS  F(X)
            C
            C     AND  A  IS  9X9  MATRIX
            C
            C     X-VECTOR  IS  ( PHI , VPHI , WPHI , THT , VTHT , WTHT , PSI ,
            C
            C          VPSI ,    WPSI )
            C
            C     ****************************************************************
            C
ISN 0014          WRITE(6,1070)PHI,VPHI,THT,VTHT,PSI,VPSI,TM,T1,T2,
                 1XKM,XKC,A11,A13,D12,GAM1C,GAM2C,BET1C,BET2C,AITREN,HTHT,HPHI,HPSI
ISN 0015     1070 FORMAT(1H /10X,14H INITIAL STATE / 6E13.6 / 10X,13H INPUT CONSTS /
                 110X, 29H TM,T1,T2,XKM,XKC,A11,A13,D12 / 8E14.7 / 10X, 30H GAM1C,GA
                 2M2C,BET1C,BET2C,(RAD) / 4E14.7 / 10X,10H INERTIA = E14.7 ,/10X,
                 317H HTHT,HPHI,HPSI = 3E14.7//)
            C     ****************************************************************
ISN 0016          PI = 3.1415926
ISN 0017          DTR  = PI/180.0
ISN 0018          RTD  = 180.0/PI
            C     ****************************************************************
            C
ISN 0019          SB2C  = SIN(BET2C)
ISN 0020          CB2C  = COS(BET2C)
ISN 0021          TB2C = SB2C/CB2C
ISN 0022          SG1C  = SIN(GAM1C)
ISN 0023          CG1C  = COS(GAM1C)
ISN 0024          SG2C  = SIN(GAM2C)
ISN 0025          CG2C  = COS(GAM2C)
ISN 0026          SB1C  = SIN(BET1C)
ISN 0027          CB1C  = COS(BET1C)
ISN 0028          TB1C  = SB1C/CB1C
ISN 0029           SGAM1C = SG1C
ISN 0030           SGAM2C = SG2C
ISN 0031           CGAM1C = CG1C
ISN 0032           CGAM2C = CG2C
            C
```

```
                 C       ****************************************************************
ISN 0033                 DIF = GAM1C - GAM2C
ISN 0034                 SDIF = SIN(DIF)
                 C
                 C       ****************************************************************
                 C
                 C       CALCULATION OF THE  A-MATRIX
                 C
ISN 0035                 DO 10 I=1,6
ISN 0036                 DO 10 J=1,6
ISN 0037              10 A(I,J) = 0.0
                 C
ISN 0038                 A(1,2)=-10.0/AITREN
ISN 0039                 A(2,1)=XKM*XKC/(10.0*TM)
ISN 0040                 A(2,2)=-(1.0/TM-A(2,1)*45.0/AITREN)
ISN 0041                 A(2,3)=-A(2,1)*TB1C*CG1C
ISN 0042                 A(2,4)= 4.5*A(1,2)*A(2,3)
ISN 0043                 A(2,5)=A(2,1)*SG1C*TB1C
ISN 0044                 A(2,6)=4.5*A(1,2)*A(2,5)
ISN 0045                 A(3,4)=A(1,2)
ISN 0046                 A(4,3)=A(2,1)*D12*SDIF
ISN 0047                 A(4,4)=A(4,3)*A(1,2)*4.5-1./TM
ISN 0048                 A(5,6)=A(1,2)
ISN 0049                 A(6,5)=A(2,1)*D12*SDIF
ISN 0050                 A(6,6)=  +4.5*A(6,5)*A(1,2)-1./TM
                 C
ISN 0051                 WRITE(6,5000) ((A(I,J),J=1,6),I=1,6)
ISN 0052            5000 FORMAT(1H / 1X,10H A-MATRIX /(1X, 6E13.6 ))
                 C
                 C       ****************************************************************
                 C
                 C       CALCULATION  OF DB1,DB2,DG1,DG2
                 C
                 C       NEEDS ANGLFS FROM STATE VECTOR AND COMMAND ANGLES AS INPUT
                 C
                 C
                 C
ISN 0053             312 PHI= X(1) + X1E
ISN 0054                 THT= X(3) + X3E
ISN 0055                 PSI= X(5) + X5E
ISN 0056                 SPHI = SIN(PHI)
ISN 0057                 CPHI = COS(PHI)
ISN 0058                 STHT = SIN(THT)
ISN 0059                 CTHT = COS(THT)
ISN 0060                 SPSI = SIN(PSI)
ISN 0061                 CPSI = COS(PSI)
ISN 0062                 TTHT = STHT/CTHT                                          AFX
ISN 0063                 XNORW = XKC*T1/T2                                         AFX
ISN 0064                 PUP = 1.0/(XKC*TM)                                        AFX
                 C
ISN 0065                 R1=TB1C/CG1C
ISN 0066                 R2=TB2C/CG2C
ISN 0067                 TG1=SG1C/CG1C
ISN 0068                 TG2=SG2C/CG2C
ISN 0069                 TPHI=SPHI/CPHI
ISN 0070                 IF (ABS(X(4))-0.15)9991,9991,9992
```

```
ISN 0071      9991 B4=-0.5*X(4)**2*(1.0-X(4)**2/12.0)
ISN 0072           GO TO 9993
ISN 0073      9992 B4=CTHT-1.0
ISN 0074      9993 IF (ABS(X(7))-0.15) 9994,9994,9995
ISN 0075      9994 B7=-0.5*X(7)**2*(1.0-X(7)**2/12.0)
ISN 0076           GO TO 9996
ISN 0077      9995 B7=CPSI-1.0
ISN 0078      9996 EALF=B4*TG1-(SPSI*TPHI+CPSI*STHT)*R1+CPSI*TPHI-SPSI*STHT
ISN 0079           EBET=B7+SPSI*STHT*TPHI-(SPSI-TPHI*STHT*CPSI)*R1-TPHI*CTHT*TG1
ISN 0080           DG1=ATAN((EALF-EBET*TG1)/(1.0+EBET+(TG1+EALF)*TG1))
ISN 0081           G1=DG1+GAM1C
ISN 0082           EALF=B4*TG2+(SPSI*TPHI+CPSI*STHT)*R2+CPSI*TPHI-SPSI*STHT
ISN 0083           EBET=B7+SPSI*STHT*TPHI+(SPSI-TPHI*STHT*CPSI)*R2-TPHI*CTHT*TG2
ISN 0084           DG2=ATAN((EALF-EBET*TG2)/(1.0+EBET+(TG2+EALF)*TG2))
ISN 0085           G2=DG2+GAM2C
ISN 0086           R=CPSI*CTHT*SB1C+SPSI*CTHT*CG1C*CB1C+STHT*SG1C*CB1C
ISN 0087           DB1=ASIN(R)
ISN 0088           DB1=DB1-BET1C
ISN 0089           IF (ABS(DB1)-0.1) 9997,9998,9998
ISN 0090      9997 DB1=DBET(B4,B7,TB1C,STHT,SPSI,CG1C,SG1C,+1.0)
ISN 0091      9998 R=CPSI*CTHT*SB2C-SPSI*CTHT*CG2C*CB2C-STHT*SG2C*CB2C
ISN 0092           DB2=ASIN(R)
ISN 0093           DB2=DB2-BET2C
ISN 0094           IF (ABS(DB2)-0.1) 9999,10009,10009
ISN 0095      9999 DB2=DBET(B4,B7,TB2C,STHT,SPSI,CG2C,SG2C,-1.0)
ISN 0096     10009 CONTINUE
             C
             C     ********************************************************************
             C
             C     CALCULATION OF SIX DIM F VECTOR
             C
             C
ISN 0097           B1=DB1+BET1C
ISN 0098           B2=DB2+BET2C
ISN 0099           SG1=SIN(G1)
ISN 0100           CG1=COS(G1)
ISN 0101           TB1=SIN(B1)/COS(B1)
ISN 0102           TB2=SIN(B2)/COS(B2)
ISN 0103           SG2=SIN(G2)
ISN 0104           CG2=COS(G2)
ISN 0105           SUM2=SUM4/T2
ISN 0106           SUM3=AITREN/(10.0*TM)
ISN 0107           SUM4=AITREN*T1/(T2*XKM)
ISN 0108           SUM7=CG2*DB1+CG1*DB2
ISN 0109           SUM8=SG1*DB2+SG2*DB1
ISN 0110           SUM8=-SUM8
ISN 0111           SUM9=-SG2*CG2*TB2*DB1+SG1*CG1*TB1*DB2
ISN 0112           GAIN = XKC * (T1&T2)/T2                                     AFX
ISN 0113           F(1)=A(1,2)*(X(2)+TTHT*(SPHI*X(4)+CPHI*X(6)))
ISN 0114           A1=-1/TM
ISN 0115           ARGF2=    (DG1+A(1,2)*(X(2)-TB1*(CG1*X(4)-SG1*X(6)))*4.5)*XKC
ISN 0116           IF (ABS(ARGF2).LE.F2LIM) F2 = ARGF2
ISN 0118           IF (ARGF2.GT.F2LIM) F2 = F2LIM
ISN 0120           IF (ARGF2.LT.-F2LIM) F2= -F2LIM
ISN 0122           F(2)=A1*X(2)-SUM3*HPHI+A(2,1)*F2/XKC
ISN 0123           F(3)=A(1,2)*(CPHI*X(4)-SPHI*X(6))
```

```
ISN 0124          ARGF4=XKC*D12*(SUM7+A(1,2)*(SUM8*X(2)+(SDIF+SUM9)*X(4)+(SG2**2*TB2
                 1*DB1-SG1**2*TB1*DB2)*X(6))*4.5)
ISN 0125          IF(ABS(ARGF4).LE.F2LIM) F4 =ARGF4
ISN 0127          IF(ARGF4.GT.F2LIM) F4=F2LIM
ISN 0129          IF(ARGF4 .LT. -F2LIM)F4 = -F2LIM
ISN 0131          F(4)=A1*X(4)-SUM3*HTHT+A(2,1)*F4/XKC
ISN 0132          F(5)=A(1,2)*(SPHI*X(4)+CPHI*X(6))/CTHT
ISN 0133          ARGF6=XKC*D12*(+SUM8-A(1,2)*(SUM7*X(2)+(CG2**2*TB2*DB1-CG1**2*TB1*
                 1DB2)*X(4)+(-SDIF+SUM9)*X(6))*4.5)
ISN 0134          IF(ABS(ARGF6).LE.F2LIM) F6 = ARGF6
ISN 0136          IF(ARGF6.GT.F2LIM) F6 = F2LIM
ISN 0138          IF(ARGF6.LT.-F2LIM) F6=-F2LIM
ISN 0140          F(6)=A1*X(6)-SUM3*HPSI+A(2,1)*F6/XKC
            C
            C
            C     ******************************************************************
            C
ISN 0141          IF(JSW.GT.0)GO TO 602
ISN 0143          WRITE(6,5001) (F(I),I=1,6)
ISN 0144     5001 FORMAT(1H / 1X,12H F(X)-VECTOR / 1X,6E13.6)
ISN 0145      602 CONTINUE
            C
            C
            C     ******************************************************************
ISN 0146          RETURN
ISN 0147          END
```

APPENDIX III

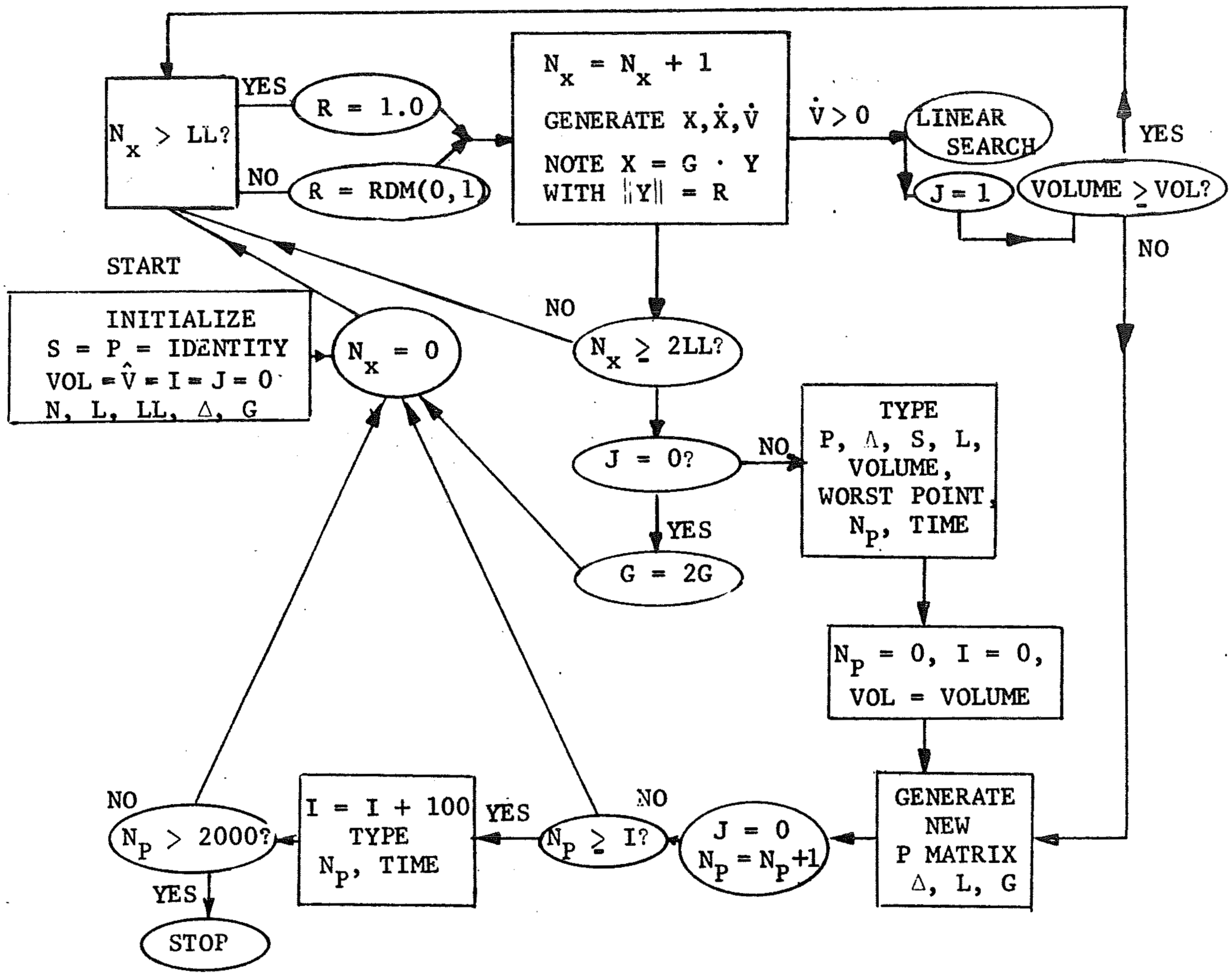
TIME SHARE COMPUTER LISTINGS

This appendix contains computer listings of three separate routines as they existed on June 12, 1969. The first is a variable step size simulation package for the OAO with zero initial momentum. It is written in FORTRAN IV for the GE Mark II time sharing service.

The second routine is a P-generation search routine for a quadratic Liapunov function for the nine-dimensional OAO. It is in FORTRAN IV for the PDP-10 time sharing service. These time share routines have been used largely to experiment, test ideas, and check; this routine, as presently written, will not perform a search but will always set $x_i = 1, i = 1,9$. Modification of line 860 to read "GØ TØ 5" and deletion of lines 1621 and 1622 will change it back into a search routine.

The third routine illustrates a typical session at the teletype. It presents a typical run for the Faulkner equation, and is explained by the flow chart (Fig. III-1). The program is written for the Davis AL/COM system in FORTRAN IV. The entire session at the teletype is presented. It begins with a list of the routines to be executed, followed by a listing of the main routine, the logic of which is shown in Fig. III-1.

The listing is next edited to demonstrate the ease with which the case $n = 2$ can be changed to $n = 9$, and the program is run. It finds the quadratic estimate of the domain $3.5\ x^2 + .25\ xy + 2.0\ y^2 \leq .05$ after examining 95 matrices. The corresponding volume is $(.139)^2 = .019$, which is not as good as reported in Table 2 after 3102 trials, but this routine



Notation

N_x Is the Number of Points x Searched to Date, LL the Total Number to be Searched, V Is the Liapunov Function, $V = x^T P x$, Λ Is the Matrix of Eigenvalues of $P = S \Lambda S^T$, $\Delta = \det P$, N_P Is the Number of P Matrices Since the Last Best, VOLUME = $L^{n/2}/\sqrt{\Delta}$, Where $L = x^T P x$ for the Present x and VOL Is the Best Volume So Far, VOL = $L^{n/2}/\sqrt{\Delta}$ with $L = w^T P w$, w the "Worst Point."

Fig. III-1 Flow Chart for the Logic of the Main Routine stops after 2100 trials. The times printed out are dummy values, because subroutine CHARGE, although shown here, had not yet been implemented. Following the run, the various subroutines are listed.

OAO SIMULATION

(1 of 3 pages)

```
100 L=0
110 DIMENSIØN Z(9)
120 DIMENSIØN RELCH(9)
130 DIMENSIØN TEMP(100)
140 CØMMØN X(9),B1CR,B2CR,G1CR,G2CR,F(9),T,D12,A,B,XK
150 EXTERNAL DER
155 EXTERNAL SAT
160 X(1)=0.5235988E-01
170 X(4)=X(1);X(7)=X(1)
180 X(2)=-0.4841714E-06
181 X(5)=-X(2)
182 X(8)=X(5)
190 X(3)=-0.1034554E-04;X(6)=-0.1241465E-04;X(9)=-0.4138216E-04
191 X(1)=.2617994E+00
192 X(2)=-.4841714E-06
193 X(3)=-.1034554E-04
194 X(4)=X(1);X(5)=-X(2)
195 X(6)=-.1241465E-04
196 X(7)=X(4);X(8)=X(5)
197 X(9)=-.4138216E-04
200 T=0.0
210 TF=2.0
220 IND=0
230 DT=1.0/2.↑7.
240 NITER=0;MTST=0
250 N=9
260C M IS THE NUMBER ØF INTEGRATIØN STEPS BETWEEN PRINTØUTS
270 M=99
280 K=M
290 B1CD=30.
300 B2CD=-30.
310 G1CD=30.
320 G2CD=-30.
325 A=2685./195.
326 XK=268500.
330 B=1./76.8
340 D12=SIGN(2.,G1CD-G2CD)
350 CØNV=57.29578
360 B1CR=B1CD/CØNV;B2CR=B2CD/CØNV;G1CR=G1CD/CØNV;G2CR=G2CD/CØNV
370 DØ 83 IØTA=1,9
380 83 Z(IØTA)=X(IØTA)
390 DBLE=0.05
400 HALF=5.0
410 GØ TØ 149
420 30 PRINT,"NEXT VALUES ØF TF AND M"
430 INPUT,TF,M
440 149 CØNTINUE
460 B=1./76.8
480 10 CALL AMPB1(IND,DER,TEMP,T,DT,X,F,N,ICØUNT,NITER,MTST)
```

```
490 IF(K-M)3,2,2
500 2 K=-1
510 PRINT,"TIME",T
530 PRINT 40,(X(I),I=1,4)
540 PRINT 41,(X(I),I=5,9)
550 3 K=K+1
560 IF(T-TF)20,30,30
570 20 CALL AMPB2(IND,DER,TEMP,T,DT,X,F,N,ICØUNT,NITER,MTST)
580 IF(L-5)91,92,92
590 92 L=0;L1=9;L2=0
600 DØ 95 IØTA=1,9
610 RELCH(IØTA)=ABS((X(IØTA)-Z(IØTA))/Z(IØTA))
620 IF(RELCH(IØTA)-DBLE)94,95,95
630 94 L1=L1-1
640 95 Z(IØTA)=X(IØTA)
650 GØ TØ 97
660 91 L=L+1
670 GØ TØ 10
680 97 IF(L1)98,98,99
690 98 IND=1
700 GØ TØ 10
710 99 DØ 100 IØTA=1,9
720 IF(RELCH(IØTA)-HALF)100,102,102
730 102 L2=L2+1
740 100 CØNTINUE
750 IF(L2)10,10,103
760 103 IND=1
770 GØ TØ 20
800 40 FØRMAT(14H X-VECTØR      ,4E14.6)
810 41 FØRMAT(5E14.6)
820 END
830C EVALUATIØN ØF F=Y DØT
840  SUBRØUTINE DER
850 CØMMØN X(9),B1CR,B2CR,G1CR,G2CR,F(9),T,D12,A,B,XK
860 C1=CØS(X(1));C4=CØS(X(4));C7=CØS(X(7))
870 S1=SIN(X(1));S4=SIN(X(4));S7=SIN(X(7))
880 CG1=CØS(G1CR);SG1=SIN(G1CR)
890 CG2=CØS(G2CR);SG2=SIN(G2CR)
900 CB1=CØS(B1CR);SB1=SIN(B1CR)
910 CB2=CØS(B2CR);SB2=SIN(B2CR)
920 Q1=(S7*S1+C7*S4*C1)*SB1
930 Q2=(C7*S1-S7*S4*C1)*CG1*CB1
940 Q3=C4*C1*SG1*CB1
950 Q4=(S7*C1-C7*S4*S1)*SB1
960 Q5=(C7*C1+S7*S4*S1)*CG1*CB1
970 Q6=C4*S1*SG1*CB1
980 ALF=-Q1+Q2+Q3
990 BET=-Q4+Q5-Q6
1000 DG1=ATAN((ALF*CG1-BET*SG1)/(BET*CG1+ALF*SG1))
1010 G1=ATAN(ALF/BET)
1020 R=C7*C4*SB1+S7*C4*CG1*CB1+S4*SG1*CB1
1030 DB1=ATAN(R/(1.-R*R)↑0.5)
1040 DB1=DB1-B1CR
1050 Q1=(S7*S1+C7*S4*C1)*SB2
```

```
1060 Q2=(C7*S1-S7*S4*C1)*CG2*CB2
1070 Q3=C4*C1*SG2*CB2
1080 Q4=(S7*C1-C7*S4*S1)*SB2
1090 Q5=(C7*C1+S7*S4*S1)*CG2*CB2
1100 Q6=C4*S1*SG2*CB2
1110 ALF=Q1+Q2+Q3
1120 BET=Q4+Q5-Q6
1130 G2=ATAN(ALF/BET)
1140 R=C7*C4*SB2-S7*C4*CG2*CB2-S4*SG2*CB2
1150 DB2=ATAN(R/(1.-R*R)↑0.5)
1160 DB2=DB2-B2CR
1180 Z2=10.*XK*DG1+9.*XK*X(3)
1190 Z5=20.*XK*(DB1*CØS(G2)+DB2*CØS(G1)+.45*X(6))
1200 Z5=.5*D12*Z5
1210 Z8=20.*XK*(-DB1*SIN(G2)-DB2*SIN(G1)+.45*X(9))
1220 Z8=.5*D12*Z8
1230 F(1)=-A*(X(2)+(X(5)*S1+X(8)*C1)*S4/C4)
1240 F(2)=-B*(X(2)-SAT(Z2)/XK)
1250 F(3)=-2.0*(X(3)+DG1)
1260 F(4)=-A*(X(5)*C1-X(8)*S1)
1270 F(5)=-B*(X(5)-SAT(Z5)/XK)
1280 F(6)=-2.0*X(6)-2.0*D12*(CØS(G2)*DB1+CØS(G1)*DB2)
1290 F(7)=-A*(X(5)*S1+X(8)*C1)/C4
1300 F(8)=-B*(X(8)-SAT(Z8)/XK)
1310 F(9)=-2.0*X(9)+2.0*D12*(SIN(G2)*DB1+SIN(G1)*DB2
1390 RETURN
1400 END
1410 FUNCTIØN SAT(Z)
1420 IF(Z-26.)5,2,2
1430 2 SAT=26.
1440 GØTØ 1
1450 5 IF(Z+26.)3,3,4
1460 3 SAT=-26.
1470 GØ TØ 1
1480 4 SAT=Z
1490 1 RETURN
1500 END

460
```

OAO QUADRATIC LIAPUNOV FUNCTION SEARCH


```
DDD  12-JUN-69  14:21

       CØMMØN X(9),F(9)
      CØMMØN D12,A,B,XK,G1CR,G2CR,B1CR,B2CR,TB1,TB2,CG1,CG2,SG1,
 SG2
      CØMMØN TG1,TG2,R1,R2,SB1,SB2,CB1,CB2
      DIMENSIØN P(9,9),PZ(9,9),PX(9),D(9),EIG(9,9),TEMP1(9),
     + TEMP2(9)
      DIMENSIØN P1(9,9),Y(9),Z(9),EL(9,9)
      A=2685./195.
      B=1./76.8
      XK=268500.
        G1CR=0.05017822
        G2CR=-G1CR
      B1CR=0.0
      B2CR=-0.5235988
      CG1=CØS(G1CR)
      CG2=CØS(G2CR)
      SG1=SIN(G1CR)
      SG2=SIN(G2CR)
      TG1=SG1/CG1
      TG2=SG2/CG2
      D12=SIGN(2.,G1CD-G2CD)
      D12=10.0*D12
      CB1=CØS(B1CR)
      CB2=CØS(B2CR)
      SB1=SIN(B1CR)
      SB2=SIN(B2CR)
      TB1=SB1/CB1
      TB2=SB2/CB2
      R1=TB1/CG1
      R2=TB2/CG2
      DØ 1 I=1,9
      DØ 2 J=1,9
      PZ(I,J)=0.0
    2 P(I,J)=0.0
      PZ(I,I)=1.0
    1 P(I,I)=1.0
        P(1,1)=.4352277E+02
        P(1,2)=.8029531E+01
      P(2,1)=P(1,2)
        P(2,2)=.3393061E+02
        P(1,3)=.1009517E+02
      P(3,1)=P(1,3)
        P(3,3)=.2354561E+01
        P(2,3)=.1674426E+01
      P(3,2)=P(2,3)
        P(4,4)=.1133817E+02
        P(4,5)=.7809173E+01
      P(5,4)=P(4,5)
        P(5,5)=.6688454E+01
        P(4,6)=.4734676E+01
      P(6,4)=P(4,6)
```

```
              P(6,6)=.2547270E+01
              P(5,6)=.2397112E+01
            P(6,5)=P(5,6)
              P(7,7)=.7626979E+01
              P(7,8)=.8563903E+00
            P(8,7)=P(7,8)
              P(8,8)=.1421839E+02
              P(7,9)=.6084420E+01
            P(9,7)=P(7,9)
              P(9,9)=.1100106E+02
              P(8,9)=-.8632137E+01
            P(9,8)=P(8,9)
              VH=.4506492E+00
              VL=10.
            LL=5000
            N=0
            SC=.02
            VH=0.0
              G=XNØRM1(-1.,0.0,1.0)
              NN=0
            GØ TØ 753
      8     K=0
      80    DØ 9 I=1,9
      9     D(I)=0.5*X(I)
      110   DØ 10 I=1,9
      10    X(I)=X(I)-D(I)
      20    K=K+1
            IF (K-15)  35, 36, 36
      35    CALL DER
            VD=0.0
            DØ 12 I=1,9
            PX(I)=0.0
            DØ 13 J=1,9
      13    PX(I)=PX(I)+P(I,J)*X(J)
      12    VD=VD+F(I)*PX(I)
            IF (VD)  17, 18, 18
      17    DØ 19 I=1,9
            D(I)=0.5*D(I)
      19    X(I)=X(I)+D(I)
            GØ TØ 20
      18    DØ 201 I=1,9
      201   D(I)=0.5*D(I)
            GØ TØ 110
      36    VL=0.0
            DØ 51 I=1,9
            PX(I)=0.0
            DØ 52 J=1,9
      52    PX(I)=PX(I)+P(I,J)*X(J)
      51    VL=VL+X(I)*PX(I)
              TYPE 631,VL
              CALL RTIME(IT)
       9900 FØRMAT(5G)
              TYPE 9900,N,VH,VL,IT
      9800    FØRMAT(' X-VECTØR ')
```

```
           TYPE 9800
           TYPE 9895,X
        XPZX=0.0
        DØ 38 I=1,9
        PX(I)=0.0
        DØ 37 J=1,9
 37     PX(I)=PX(I)+PZ(I,J)*X(J)
 38     XPZX=XPZX+X(I)*PX(I)
        IF (XPZX-0.99*VH)  50, 7, 7
 50     SCALE=SCALE+SC
        N=0
           NN=0
        VL=10.0
 99     DØ 973 I=1,9
 973    P(I,I)=0.0
        DØ 100 I=1,8
        II=I+1
        DØ 100 J=II,9
          G=XNØRM1(0.0,0.0,1.0)
        P(I,J)=PZ(I,J)+G*SCALE
 100    P(J,I)=P(I,J)
        DØ 102 I=1,9
          G=XNØRM1(0.0,0.0,1.0)
 102    P(I,I)=PZ(I,I)+G*SCALE
 5      DØ 72 I=1,9
        DØ 72 J=1,9
 72     P1(I,J)=P(I,J)
 73     CALL EIG1(P1,EIG,9,1.0E-08,TEMP1,TEMP2,9,9)
          DETP=1.0
          DØ 632 I=1,9
 632      DETP=DETP*P1(I,I)
          TYPE 633,DETP
 9898     FØRMAT(9H P MATRIX)
 633      FØRMAT(' DET P =',G)
 9897     FØRMAT(9(5E14.6/4E14.6//))
 9895     FØRMAT(5E14.6/4E14.6)
 77       DØ 71 I=1,9
          IF(P1(I,I).LE.0.0)  GØ TØ 99
 71       D(I)=(VL/P1(I,I))**0.5
 5001     DØ 74 I=1,9
          Y(I)=XNØRM1(0.,0.,.3)
 74       Y(I)=Y(I)*D(I)
        DØ 753 I=1,9
          R=R+Z(I)*Z(I)
        X(I)=0.0
        DØ 753 J=1,9
 753    X(I)=X(I)+EIG(I,J)*Y(J)
          DØ 7014 I=1,9
 7014     X(I)=1.0
          IF(N-NN)  800,  801,  800
 801      NN=NN+50
          CALL RTIME(IT)
          TYPE 802,N,IT
```

```
 802      FØRMAT(' N=',I5,' CLØCK TIME IS',G,' MILLISECØNDS')
 800      N=N+1
          CALL DER
          PRINT 631,F
 631      FØRMAT(9G)
         VD=0.0
         DØ 186 I=1,9
         PX(I)=0.0
         DØ 6 J=1,9
 6       PX(I)=PX(I)+P(I,J)*X(J)
 186     VD=VD+F(I)*PX(I)
          TYPE 8000,VD
 8000     FØRMAT(' VD AT VICS X IS',G)
          GØ TØ36
         IF (VD)  7, 8, 8
 7       IF (N-LL)  5001, 61, 61
 61      VH=VL
 9894     FØRMAT(20H SCALE, SC, NØ. ØF P)
          TYPE 9894
          RATIØ=SCALE/SC
          TYPE 9900,SCALE,SC,RATIØ
         DØ 62 J=1,9
         DØ 62 I=1,9
 62      PZ(I,J)=P(I,J)
          TYPE 9899
          TYPE 9897,PZ
         CALL EIG1(P,EIG,9,1.0E-08,TEMP1,TEMP2,9,9)
 9890     FØRMAT(' EIGENVALUES')
          PRINT 9890
          TYPE 9895,(P(I,I),I=1,9)
          TYPE 9891
 9891     FØRMAT(' EIGENVECTØR MATRIX')
          TYPE 9897,EIG
         DET=1.0
         SC=0.0
         DØ 932 I=1,9
         SC=SC+P(I,I)
 932     DET=DET*P(I,I)
         SC=1.0/SC
         SCALE=0.0
  9899   FØRMAT(1H ,'NEW P ZERØ MATRIX')
 9893     FØRMAT(1H ,'VH,DETP,SQRT(VØL)')
           TYPE 9893
          SQRTV=VH**2.25/DET**0.25
          VØL=1./(SQRTV)**0.5
          TYPE 9900,VH,DET,SQRTV,VØL
         GØ TØ 50
         END
         SUBRØUTINE DER
          CØMMØN X(9),F(9)
         CØMMØN D12,A,B,XK,G1CR,G2CR,B1CR,B2CR,TB1,TB2,CG1,CG2,SG1,
 SG2
           CØMMØNTG1,TG2,R1,R2,SB1,SB2,CB1,CB2
         C1=CØS(X(1))
```

OAO QUADRATIC LIAPUNOV FUNCTION SEARCH


```
      C4=CØS(X(4))
      C7=CØS(X(7))
      S1=SIN(X(1))
      S4=SIN(X(4))
      S7=SIN(X(7))
      T1=S1/C1
      IF (ABS(X(4))-0.15)  1, 1, 2
 1    B4=-0.5*X(4)*X(4)*(1.0-X(4)*X(4)/12.0)
      GØ TØ 3
 2    B4=C4-1.0
 3    IF (ABS(X(7))-0.15)  4, 4, 5
 4    B7=-0.5*X(7)*X(7)*(1.0-X(7)*X(7)/12.0)
      GØ TØ 6
 5    B7=C7-1.0
 6    EALF=B4*TG1-(S7*T1+C7*S4)*R1+C7*T1-S7*S4
      EBET=B7+S7*S4*T1-(S7-T1*S4*C7)*R1-T1*C4*TG1
      DG1=ATAN((EALF-EBET*TG1)/(1.0+EBET+(TG1+EALF)*TG1))
      G1=DG1+G1CR
      EALF=B4*TG2+(S7*T1+C7*S4)*R2+C7*T1-S7*S4
      EBET=B7+S7*S4*T1+(S7-T1*S4*C7)*R2-T1*C4*TG2
      DG2=ATAN((EALF-EBET*TG2)/(1.0+EBET+(TG2+EALF)*TG2))
      G2=G2CR+DG2
      R=C7*C4*SB1+S7*C4*CG1*CB1+S4*SG1*CB1
        DB1=ASIN(R)
      DB1=DB1-B1CR
      IF (ABS(DB1)-0.1)  7, 8, 8
 7    DB1=DBET(B4,B7,TB1,S4,S7,CG1,SG1,+1.0)
 8    R=C7*C4*SB2-S7*C4*CG2*CB2-S4*SG2*CB2
        DB2=ASIN(R)
      DB2=DB2-B2CR
      IF (ABS(DB2)-0.1)  9, 10, 10
 9    DB2=DBET(B4,B7,TB2,S4,S7,CG2,SG2,-1.0)
 10   CØNTINUE
      Z2=10.0*XK*DG1+9.0*XK*X(3)
      Z5=D12*XK*(DB1*CØS(G2)+DB2*CØS(G1)+.45*X(6))
      Z8=D12*XK*(-DB1*SIN(G2)-DB2*SIN(G1)+.45*X(9))
      F(1)=-A*(X(2)+(X(5)*S1+X(8)*C1)*S4/C4)
      F(2)=-B*(X(2)-SAT(Z2)/XK)
      F(3)=-2.0*(X(3)+DG1)
      F(4)=-A*(X(5)*C1-X(8)*S1)
      F(5)=-B*(X(5)-SAT(Z5)/XK)
      F(6)=-2.0*X(6)-0.2*D12*(CØS(G2)*DB1+CØS(G1)*DB2)
      F(7)=-A*(X(5)*S1+X(8)*C1)/C4
      F(8)=-B*(X(8)-SAT(Z8)/XK)
      F(9)=-2.0*X(9)+0.2*D12*(SIN(G2)*DB1+SIN(G1)*DB2)
      RETURN
      END
      FUNCTIØN SAT(Z)
      IF (Z-26.)  5, 2, 2
 2    SAT=26.
      GØ TØ 1
 5    IF (Z+26.)  3, 3, 4
 3    SAT=-26.
      GØ TØ 1
```

OAO QUADRATIC LIAPUNOV FUNCTION SEARCH


```
    4       SAT=Z
    1       RETURN
            END
            FUNCTIØN DBET(B4,B7,TB,S4,S7,CG,SG,SIGN)
            XKAP=(B7+B4+B7*B4)*TB+((1.0+B4)*S7*CG+S4*SG)*SIGN
            XMU=XKAP
    2       SQ=XMU*XMU
            XF=XMU-XKAP-TB*0.5*SQ*(1.0+0.25*SQ*(1.0+0.5*SQ))
            IF (XF*XMU.EQ.0.0)  GØ TØ 1
              XMU=XMU-XF/(1.0-TB*XMU*(1.0+0.5*SQ*(1.0+0.75*SQ)))
            IF (ABS(XF/XMU).GE.1.0E-06)  GØ TØ 2
    1       CØNTINUE
              DBET=ASIN(XMU)
            RETURN
            END
*

SYSTEM?..
.HELLØ
 C/P UNITS      14.4

 CØNNECT TIME  00:18
```

```
↑C

.LØGI
AL/CØM  JØB 9   LINE 24  12-JUN-69  12:51
TYPE CØMPANY PRØJECT NAME
GRUMMAN DEMØ ALL
6-11: FØR CHANGED SYSTEM UNAVAILABILITY SCHEDULE "TYPE SYS:SCHED.ØPR"

EXIT
↑C

.TYPE LIST.CØM

LIST.CØM  6/12/69  1252  542-1-1

00100   VAN.FØR
00110   PGGØ.FØR
00120   PG2.FØR
00130   LIN.FØR
00140   DER.FØR
00150   INITIAL.FØR
00160   RAND.FØR
00170   RANDU.FØR
00180   CHARGE.MAC

EXIT
↑C

.TYPE VAN.FØR

VAN.FØR  6/12/69  1253  542-1-1

00050             CØMMØN XX(9,200),FF(9,200),P(9,9),SM(9,9),THETA(28),
00055          1  PHIV(8),XLAM(9),D(9),X(9),F(9),G(9),Y(9),PX(9),
00060          2  WØRST(9),VD,RZSRFG,DEI,VØL,VØL1,XNU4,RXNU4,PI,VH,
00065          3  VL,LL,N,NØP,NPH,NTH,NØPTS
00090             III=12
00100             IGØR=0
00110             CALL INITAL
00115             LLL=2*LL
00120             ZSRFG=N
00130             XNU4=ZSRFG/4.
00140             RXNU4=1.0/XNU4
00150             RZSRFG=1./ZSRFG
00160   14        NØPTS=0
00170   1         IF(NØPTS-LL)20,20,21
00180   21        R=1.0
00190             GØ TØ 22
00200   20        R=RAND(0.0)**RZSRFG
00210   22        SIZE=0.0
00220             DØ 31 I=1,N
00230             Y(I)=(2.*RAND(0.0)-1.)
00240   31        SIZE=SIZE+Y(I)*Y(I)
```

```
      SIZE=SQRT(SIZE)
      DØ 32 I=1,N
32    Y(I)=(Y(I)/SIZE)*R*G(I)
      DØ 3 I=1,N
2     X(I)=0.0
      DØ 3 J=1,N
3     X(I)=X(I)+SM(I,J)*Y(J)
      CALL DER
      VD=0.0
      DØ 5 I=1,N
      PX(I)=0.0
      DØ 4 J=1,N
4     PX(I)=PX(I)+P(I,J)*X(J)
5     VD=VD+F(I)*PX(I)
      NØPTS=NØPTS+1
548   IF(VD)6,6,7
6     IF(NØPTS-LLL)1,11,11
7     CALL LIN
      JSW=1
      IF(VØL1-VØL)10,1,1
10    CALL PG2
      JSW=0
      NØP=NØP+1
      IF(IGØR-NØP)16,16,14
16    IGØR=IGØR+100
      TYPE 15,III,NØP
15    FØRMAT(' TIME',I10,'MS.,',I10,' P MATRICES')
      IF(NØP-2000)14,14,17
11    IF(JSW.EQ.0)GØ TØ 40
      TYPE 100,((P(I,J),J=1,N),I=1,N)
100   FØRMAT(//' NEW P ZERØ MATRIX'/2(2G/))
      TYPE 101,(XLAM(I),I=1,N)
101   FØRMAT(' EIGENVALUES',2G)
      TYPE 110,((SM(I,J),J=1,N),I=1,N)
110   FØRMAT(' EIGENVECTØR MATRIX'/2(2G/))
      TYPE 111,VL
111    FØRMAT(' DØMAIN XTPX.LT.',G)
      TYPE 1000,VØL 1
1000  FØRMAT(' SQRT VØLUME',G)
      TYPE 1001,(WØRST(K),K=1,N)
1001  FØRMAT(' WØRST PØINT',2G)
      TYPE 1010,NØP
1010  FØRMAT(' NØ. ØF P MATRICES',G///)
      NØP=0
      VØL=VØL1
      IGØR=0
      GØ TØ 10
40    DØ 41 I=1,N
41    G(I)=2.*G(I)
      GØ TØ 14
17    STØP
      END
```

```
EXIT
↑C

.SEDIT VAN.FØR
ØLD
*MR560,615$2$9$SEARCH ENDS*T560,615
00560              TYPE 100,((P(I,J),J=1,N),I=1,N)
00565    100       FØRMAT(// ' NEW P ZERØ MATRIX '/9(9G/))
00570              TYPE 101,(XLAM(I),I=1,N)
00575    101       FØRMAT(' EIGENVALUES ',9G)
00580              TYPE 110,((SM(I,J),J=1,N),I=1,N)
00585    110       FØRMAT(' EIGENVECTØR MATRIX '/9(9G/))
00590              TYPE 111,VL
00595    111        FØRMAT(' DØMAIN XTPX.LT. ',G)
00600              TYPE 1000,VØL I
00605    1000      FØRMAT(' SQRT VØLUME ',G)
00610              TYPE 1001,(WØRST(K),K=1,N)
00615    1001      FØRMAT(' WØRST PØINT ',9G)
*MR560,615$9$2$SEARCH ENDS*E

EXIT
↑C
```

```
.EXECUTE @LIST.CØM
CØMPILING: VAN.FØR

LØADING.

CØRE 8.
START 000543



NEW P ZERØ MATRIX
     1.0000000        0.0000000
     0.0000000        1.0000000

EIGENVALUES       1.0000000         1.0000000
EIGENVECTØR MATRIX
     1.0000000        0.0000000
     0.0000000        1.0000000

DØMAIN XTPX.LT.  0.2440884E-25
SQRT VØLUME  0.1562332E-12
WØRST PØINT -0.1365368E-12  0.7593781E-13
NØ. ØF P MATRICES               0
```

```
TIME          12MS.,          1 P MATRICES


NEW P ZERØ MATRIX
     2.8136173   0.3750980E-01
 0.3750980E-01       1.6194153

EIGENVALUES       2.8147944       1.6182383
EIGENVECTØR MATRIX
     0.9995080 -0.3136356E-01
 0.3136356E-01       0.9995080

DØMAIN XTPX.LT.   0.3754065E-01
SQRT VØLUME       0.1326260
WØRST PØINT       0.1051866   0.6052599E-01
NØ. ØF P MATRICES                39



TIME          12MS.,          1 P MATRICES


NEW P ZERØ MATRIX
     3.5038651       0.1267110
     0.1267110       2.0193114

EIGENVALUES       3.5141063       2.0085704
EIGENVECTØR MATRIX
    -0.9964264   0.8446520E-01
-0.8446520E-01      -0.9964264

DØMAIN XTPX.LT.   0.5133801E-01
SQRT VØLUME       0.1390094
WØRST PØINT  0.1128978E-01        0.1580458
NØ. ØF P MATRICES                94


TIME          12MS.,          1 P MATRICES
TIME          12MS.,        100 P MATRICES
TIME          12MS.,        200 P MATRICES
TIME          12MS.,        300 P MATRICES
TIME          12MS.,        400 P MATRICES
TIME          12MS.,        500 P MATRICES
TIME          12MS.,        600 P MATRICES
TIME          12MS.,        700 P MATRICES
TIME          12MS.,        800 P MATRICES
TIME          12MS.,        900 P MATRICES
TIME          12MS.,       1000 P MATRICES
TIME          12MS.,       1100 P MATRICES
TIME          12MS.,       1200 P MATRICES
TIME          12MS.,       1300 P MATRICES
```

```
TIME        12MS.,      1400 P MATRICES
TIME        12MS.,      1500 P MATRICES
TIME        12MS.,      1600 P MATRICES
TIME        12MS.,      1700 P MATRICES
TIME        12MS.,      1800 P MATRICES
TIME        12MS.,      1900 P MATRICES
TIME        12MS.,      2000 P MATRICES
TIME        12MS.,      2100 P MATRICES
EXIT
↑C

.TYPE PGGØ.FØR

PGGØ.FØR  6/12/69  1308  542-1-1

      SUBRØUTINE PGGØ
      CØMMØN XX(9,200),FF(9,200),P(9,9),SM(9,9),THETA(28),
     1 PHIV(8),XLAM(9),D(9),X(9),F(9),G(9),Y(9),PX(9),
     2 WØRST(9),VD,RZSRFG,DET,VØL,VØL1,XNU4,RXNU4,PI,VH,
     3 VL,LL,N,NØP,NPH,NTH,NØPTS
      DØ 4 I=1,N
      DØ 3 J=1,N
      P(I,J)=0.0
3     SM(I,J)=0.0
4     SM(I,1)=1.0
      DØ 2 K =1,NPH
      KP1=K+1
      C=CØS(PHIV(K))
      S=SIN(PHIV(K))
      DØ 1 M=1,N
      STEM=SM(K,M)
      SM(K,M)=SM(K,M)*C-SM(N,M)*S
1     SM(N,M)=STEM*S+SM(N,M)*C
      IF(K.GE.NPH) GØ TØ 5
      DØ 2 L =NPH, KP1,-1
      MU=((2*(N-1)-K)*(K-1))/2+N-L
      C=CØS(THETA (MU))
      S=SIN(THETA (MU))
      DØ 2 M=1,N
      STEM=SM(K,M)
      SM(K,M)=SM(K,M)*C-SM(L,M)*S
2     SM(L,M)=STEM*S+SM(L,M)*C
5     DET=1.0
      DØ 7 I=1,N
7     DET=DET*XLAM(I)
      VL=(VØL*(DET**0.25)*30.)**(RXNU4)
      DØ 6 I=1,N
      G(I)=(VL/XLAM(I))**0.5
      DØ 6 J=1,N
      DØ 6 K =1,N
6     P(I,J)=P(I,J)+SM(I,K)*XLAM(K)*SM(J,K)
      RETURN
      END
```

```
EXIT
↑C

.TYPE PG2.FØR

PG2.FØR  6/12/69  1310  542-1-1

            SUBRØUTINE PG2
            CØMMØN XX(9,200),FF(9,200),P(9,9),SM(9,9),THETA(28),
         1  PHIV(8),XLAM(9),D(9),X(9),F(9),G(9),Y(9),PX(9),
         2  WØRST(9),VD,RZSRFG,DET,VØL,VØL1,XNU4,RXNU4,PI,VH,
         3  VL,LL,N,NØP,NPH,NTH,NØPTS
            IF(NØP.LE.100)GØ TØ 20
            IF(NØP.LE.500)GØ TØ 30
            IF(NØP.LE.1000)GØ TØ 20
            IF(NØP.LE.1500)GØ TØ 30
            IF(NØP.LE.2000)GØ TØ 20
    20      DØ 22 I=1,N
    22      XLAM(I)=1./RAND(0.0)
            IF(NØP.GT.500)GØ TØ 100
    30      DØ 24 I=1,NPH
    24      PHIV(I)=PI*(-1.+2.*RAND(0.0))
            DØ 26 I=1,NTH
    26      THETA(I)=0.5*PI*(-1.+2.*RAND(0.0))
    100     CALL PGGØ
            RETURN
            END

EXIT
↑C

.TYPE LIN.FØR
```

```
LIN.FØR   6/12/69   1312   542-1-1

              SUBRØUTINE LIN
              CØMMØN XX(9,200),FF(9,200),P(9,9),SM(9,9),THETA(28),
         1    PHIV(8),XLAM(9),D(9),X(9),F(9),G(9),Y(9),PX(9),
         2    WØRST(9),VD,RZSRFG,DET,VØL,VØL1,XNU4,RXNU4,PI,VH,
         3    VL,LL,N,NØP,NPH,NTH,NØPTS
              K=0
              DØ 1 I=1,N
    1         D(I)=0.5*X(I)
    2         DØ 3 I=1,N
    3         X(I)=X(I)-D(I)
    4         K=K+1
              IF(K.GE.15)GØ TØ 11
              CALL DER
              VD=0.0
              DØ 6 I=1,N
              PX(I)=0.0
              DØ 5 J=1,N
    5         PX(I)=PX(I)+P(I,J)*X(J)
    6         VD=VD+F(I)*PX(I)
              IF(VD)7,9,9
    7         DØ 8 I=1,N
              D(I)=0.5*D(I)
    8         X(I)=X(I)+D(I)
              GØ TØ 4
    9         DØ 10 I=1,N
   10         D(I)=0.5*D(I)
              GØ TØ 2
   11         DØ 12 I=1,N
   12         WØRST(I)=X(I)
              VL=0.0
              DØ 14 I=1,N
              PX(I)=0.0
              DØ 13 J=1,N
   13         PX(I)=PX(I)+P(I,J)*X(J)
   14         VL=VL+X(I)*PX(I)
              DØ 15 I=1,N
   15         G(I)=(VL/XLAM(I))**0.5
              VØL1=VL**XNU4/DET**0.25
              RETURN
              END

EXIT
↑C

.TYPE DER.FØR
```

```
DER.FØR   6/12/69   1315   542-1-1

            SUBRØUTINE DER
            CØMMØN XX(9,200),FF(9,200),P(9,9),SM(9,9),THETA(28),
     1      PHIV(8),XLAM(9),D(9),X(9),F(9),G(9),Y(9),PX(9),
     2      WØRST(9),VD,RZSRFG,DET,VØL,VØL1,XNU4,RXNU4,PI,VH,
     3      VL,LL,N,NØP,NPH,NTH,NØPTS
            F(1)=2.*X(2)*(3.-X(2))
            F(2)=X(1)*(-10.+4.*X(1)+2.*X(2) )+X(2)*(-1.+4.*X(2))
            RETURN
                  END

EXIT
↑C

.TYPE INITAL /.FØR
SWITCH ERRØR

EXIT
↑C

.TYPE INITAL.FØR

INITAL.FØR  6/12/69   1317   542-1-1

            SUBRØUTINE INITAL
            CØMMØN XX(9,200),FF(9,200),P(9,9),SM(9,9),THETA(28),
     1      PHIV(8),XLAM(9),D(9),X(9),F(9),G(9),Y(9),PX(9),
     2      WØRST(9),VD,RZSRFG,DET,VØL,VØL1,XNU4,RXNU4,PI,VH,
     3      VL,LL,N,NØP,NPH,NTH,NØPTS
            PI=3.1415926536
            N=2
            LL=200
            NPH=N-1
            NTH=NPH*(N-2)/2
            DØ 1 I=1,N
 1          XLAM(I)=1.0
            DØ 2 I=1,NPH
 2          PHIV(I)=0.0
            DØ 3 I=1,NTH
 3          THETA (I)=0.0
            CALL PGGØ
            NØP=0
            ZARK=RAND(-5.0)
            VH=0.0
            VL = 1000.0
            VØL=0.0
            DØ 4 I =1,N
 4          G(I)=(VL/XLAM(I))**0.5
            RETURN

EXIT
↑C
```

```
.TYPE RAND.FØR

RAND.FØR  6/12/69  1318  542-1-1

            SUBRØUTINE RAND(X)
            CALL RANDU(IX,IY,YFL)
            RAND=YFL
            IX=IY
            RETURN

EXIT
↑C

.TYPE RANDU.FØR

RANDU.FØR  6/12/69  1319  542-1-1

C
C    .................................................................
C
C       SUBRØUTINE RANDU
C
C       PURPØSE
C          CØMPUTES UNIFØRMLY DISTRIBUTED RANDØM REAL NUMBERS BETWEEN
C          0 AND 1.0 AND RANDØM INTEGERS BETWEEN ZERØ AND
C          2**35. EACH ENTRY USES AS INPUT AN INTEGER RANDØM NUMBER
C          AND PRØDUCES A NEW INTEGER AND REAL RANDØM NUMBER.
C
C       USAGE
C          CALL RANDU(IX,IY,YFL)
C
C       DESCRIPTIØN ØF PARAMETERS
C          IX - FØR THE FIRST ENTRY THIS MUST CØNTAIN ANY ØDD INTEGER
C               NUMBER WITH NINE ØR LESS DIGITS. AFTER THE FIRST ENTRY,

C               IX SHØULD BE THE PREVIØUS VALUE ØF IY CØMPUTED BY THIS
 C               SUBRØUTINE.
C          IY - A RESULTANT INTEGER RANDØM NUMBER REQUIRED FØR THE NEXT

C               ENTRY TØ THIS SUBRØUTINE. THE RANGE ØF THIS NUMBER IS
C               BETWEEN ZERØ AND 2**35
C          YFL- THE RESULTANT UNIFØRMLY DISTRIBUTED, FLØATING PØINT,
C               RANDØM NUMBER IN THE RANGE 0 TØ 1.0
C
C       REMARKS
C          THIS SUBRØUTINE IS SPECIFIC TØ SYSTEM/360
C          THIS SUBRØUTINE WILL PRØDUCE 2**35-1 TERMS
C          BEFØRE REPEATING
C
C       SUBRØUTINES AND FUNCTIØN SUBPRØGRAMS REQUIRED
C          NØNE
C
```

```
C        METHØD
C           RANDØM NUMBER GENERATØR ADAPTED FRØM 7090 FAP
C           RØUTINES. (TAH 7/29/68 ALC)
C
C       ..................................................................
C
        SUBRØUTINE RANDU(IX,IY,YFL)
        EQUIVALENCE (IZER,ZER),(IYFL,ZFL)
        ZER="200000000000
        IY="377777777777.AND.(IX*"200+IX+"3117i5164025)
        IYFL=IZER.ØR.(IY/"400)
        YFL=ZFL+ZER
        RETURN
        END

EXIT
↑C

.TYPE CHARGE.MAC

CHARGE.MAC   6/12/69   1322   542-1-1

00100              ENTRY ICHAR
00110              ICHAR:    Z
00120              CALL    0,[SIXBIT/CHARGE/]
00130              JRA 16,0(16)
00140              END

EXIT
↑C

.LØGØ
TIME ØN TIME    CU'S
12:51   0:31    109.5
↑C
```

GRUMMAN AIRCRAFT ENGINEERING CORPORATION
BETHPAGE NEW YORK