[redacted]

N67 13183
(ACCESSION NUMBER)
(THRU)
230
(PAGES)
1
(CODE)
CR-80574
(NASA CR OR TMX OR AD NUMBER)
30
(CATEGORY)

FACILITY FORM 602

GPO PRICE $ ________

CFSTI PRICE(S) $ ________

Hard copy (HC) 3.75

Microfiche (MF) 1.25

ff 653 July 65

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

**REPORT NO. MM-66-19**

The addendum contains minor changes to the Computer Program as implemented by JPL.

# FINAL REPORT
# ON
# SURVEYOR LUNAR TOUCHDOWN STABILITY STUDY

SUBMITTED TO

CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

JULY 8, 1966
IN FULFILLMENT OF CONTRACT NO. 951304
(PHASES I AND II)

PREPARED BY
R. G. ALDERSON, PROJECT ENGINEER AND D. A. WELLS
THE BENDIX CORPORATION
BENDIX PRODUCTS AEROSPACE DIVISION
ANALYTICAL MECHANICS DEPARTMENT
SOUTH BEND, INDIANA 46620

THE Bendix CORPORATION

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100.

TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

## LIST OF TABLES

## FOREWORD

This report was prepared for the Jet Propulsion Laboratory, Pasadena, California, by the Bendix Products Aerospace Division of The Bendix Corporation, South Bend, Indiana. The task was accomplished under Contract NASA 7-100, Subcontract JPL 951304 during the period from July 1965 to July 1966.

## SECTION I

## INTRODUCTION

The Surveyor Lunar Touchdown Stability Study, for which this report constitutes final documentation, was a theoretical investigation of the dynamics of the Surveyor lunar landing. Contractually, the study was divided into two phases which delineate, broadly speaking, between program development and parametric study efforts.

In Phase I, the objective of the study was to simulate analytically the dynamics of the Surveyor landing, including loads as well as motions. This simulation required theoretical analysis of the landing, and implemented the analysis via a digital computer program. The analysis treated the spacecraft as a rigid body, and each leg set was treated as a plane linkage with a rigid lower link hinged to the spacecraft and a compressible, energy absorbing upper link. Spacecraft motions were predicted as functions of the externally applied forces (gravity forces, crush and friction forces from footpads and blocks). Changes in leg set geometry were also predicted from the footpad crush and friction forces, but shock absorber forces were utilized as well. The lunar surface was considered rigid.

The computer program developed in Phase I utilized an integration method of the variable interval Runge-Kutta type with error checking to control truncation error. This integration technique is well suited to the landing dynamics problem where spacecraft motions and configuration are used to predict the applied forces, the forces are used to define accelerations in the various degrees of freedom, and the accelerations are integrated to obtain new motions and configuration. A program improvement effort was also conducted in Phase I with the result that program efficiency was increased substantially.

The parametric study conducted in Phase II was designed to ascertain whether or not planar landings could be considered the most critical from the standpoint of toppling. A comparison of many planar cases (in which motions and configuration possess a plane of symmetry) with non-planar variations revealed that there are non-planar conditions which will produce toppling at horizontal velocities for which the corresponding planar cases are stable. The encroachment of the non-planar stability boundary into areas which are stable in the planar cases does not appear to be widespread, however.

Phase II also included further program development to provide the means with which to interpret certain aspects of Surveyor telemetered data. The result of this development was a program modification which permits calculation of ground slope and cross slope angle as functions of both the vehicle configuration at or prior to touchdown and the time at which each foot impacts the surface.

The following sections of this report discuss in depth the various aspects of the Surveyor Lunar Touchdown Stability Study. Section II presents the derivation of the mathematical model, Section III describes the content and usage of the computer program, and Section IV reports on the results of computer studies performed using the program.

## SECTION II

## MATHEMATICAL MODEL

### 2.1 GENERAL TECHNIQUE

In order to accomplish the objectives of this study, it is essential that the computer simulation of the Surveyor touchdown be based firmly upon a sound, detailed dynamical analysis. This analysis must provide a mathematical model which describes the spacecraft motion during the lunar landing. A prerequisite to accurate position and velocity predictions is an adequate determination of forces and torques acting on the spacecraft. Every effort has been made to provide an analysis which will permit accurate calculations by means of an efficient computer program.

The analytical approach taken by Bendix is to represent the spacecraft as a system possessing 12 degrees of freedom:

- 3 translations of the CG in mutually orthogonal directions
- 3 rotations of the spacecraft about mutually orthogonal axes
- 3 rotations of the lower links of the landing gear relative to the spacecraft
- 3 rotations of the footpads relative to the lower links

The choice of the rigid body translation and rotation as degrees of freedom arises naturally from consideration of the objectives of the analysis. The choice of the remaining degrees of freedom (i.e. rotation of the lower links and footpads) deserves further comment, however.

Prediction of the rigid body motions is dependent upon establishing the external forces and torques acting on the system. These forces and torques can be considered to arise exclusively from the ground reaction and friction forces and from gravity. In this view of the system, it is not necessary to consider the shock absorber forces directly in the determination of the rigid body motions.

It is necessary, however, to determine changes in the geometry of the leg sets and footpads so that the amount of footpad crushing can be found. This permits calculation of the ground reaction and friction forces, and the moment arms through which they act. Then a logical way to account for changes in leg set geometry is to treat the lower links as additional degrees of freedom which do require consideration of shock absorber forces. Similarly, the ground reaction forces depend upon footpad orientation, so that these also are awarded the status of degrees of freedom.

TABLE 2-1. NOMENCLATURE FOR MATHEMATICAL MODEL

| Symbol | Definition |
|---|---|
| $\Psi$ | pitch |
| $\Phi$ | yaw |
| $\Xi$ | roll |
| $\mu_j$ | coefficient of jth footpad sliding friction |
| $\mu_{Bj}$ | coefficient of jth block sliding friction |
| $\theta_j$ | angular orientation of jth leg set, relative to No. 1 leg set |
| $\gamma_j$ | rotation of jth lower strut |
| $\alpha_j$ | rotation of jth footpad |
| $\overrightarrow{r_{hpu_j}}$ | position of vehicle jth upper hardpoint relative to CG in vehicle coordinates |
| $\overrightarrow{r_{hp\ell_j}}$ | position of vehicle jth lower hardpoint relative to CG in vehicle coordinates |
| $\overrightarrow{r_{fp_j}}$ | position of jth footpad pivot relative to CG in vehicle coordinates |
| $\Delta CG$ | height of CG above spacecraft base plane |
| $\ell_p$ | distance between roll axis and lower hardpoint |
| $\ell_L$ | length of lower strut |
| $\ell_A$ | distance between upper and lower hardpoints |
| $\beta$ | inclination of $\ell_A$ |
| $\vec{\omega}$ | vehicle angular velocity |
| $\ell_{D_j}$ | length of upper strut |
| $\theta_s$ | ground slope |
| $\lambda$ | cross slope angle |

TABLE 2-1. NOMENCLATURE FOR MATHEMATICAL MODEL (CONT.)

| Symbol | Definition |
|---|---|
| $\overrightarrow{r_{B_j}}$ | position of surface of jth block relative to CG in vehicle coordinates |
| $\ell_c$ | distance between roll axis and crushable block |
| $h_{B_j}$ | thickness of jth crushable block |
| $\overrightarrow{r_{c_{jk}}}$ | position of surface centroid of kth segment of jth footpad relative to CG in vehicle coordinates |
| $d_f$ | diameter of footpad |
| $\overrightarrow{R_{CG}}$ | position of vehicle CG in ground coordinates |
| $C_f$ | nominal footpad crush pressure |
| $d_B$ | diameter of crushable block |
| $C_B$ | nominal block crush pressure |
| $m_{p_j}$ | mass of jth footpad |
| $g$ | acceleration of gravity |
| $M$ | mass of spacecraft body |
| $F_p$ | shock absorber precharge |
| $K_s$ | spring rate of shock absorber extension stop |
| $R_D$ | nominal shock absorber damping coefficient ($R_D = R_c$ in compression and $R_R$ during rebound) |
| $\nu$ | mechanical friction coefficient |
| $K_\alpha$ | spring rate of footpad stop |
| $I_{xx}, I_{xy}$ etc. | moments and products of inertia about CG |
| $I_\gamma$ | moment of inertia of lower strut about lower hardpoint |
| $I_\alpha$ | moment of inertia of footpad about pivot |

TABLE 2-1. NOMENCLATURE FOR MATHEMATICAL MODEL (CONC.)

| Symbol | Definition |
|---|---|

Note: The following conventions of notation are used:

a) differentiation with respect to time is denoted by a dot; $\dot{\Xi}$, for example

b) vectors written in upper case, unprimed, refer to ground coordinates

c) vectors written in upper case, primed, refer to surface coordinates

d) vectors written in lower case, unprimed, refer to vehicle coordinates

e) vectors written in lower case, primed, refer to strut coordinates

f) vectors written in lower case, double primed, refer to footpad coordinates

g) subscript "o" refers to initial condition

## 2.2 COORDINATE SYSTEMS AND TRANSFORMATIONS

Several orthogonal, right-handed coordinate systems have been employed to provide complete description of the vehicle. The reference, or ground coordinate system is shown in Figure 2-1, in which a positive ground slope $\theta_s$ is shown. The Z-axis is directed parallel to the gravity vector, positive down, the X-axis is directed perpendicular to the principal slope, and the Y-axis completes the orthogonal triad, positive into the (positive) slope. The origin of the ground coordinates lies on the surface, directly below the initial position of the vehicle CG.

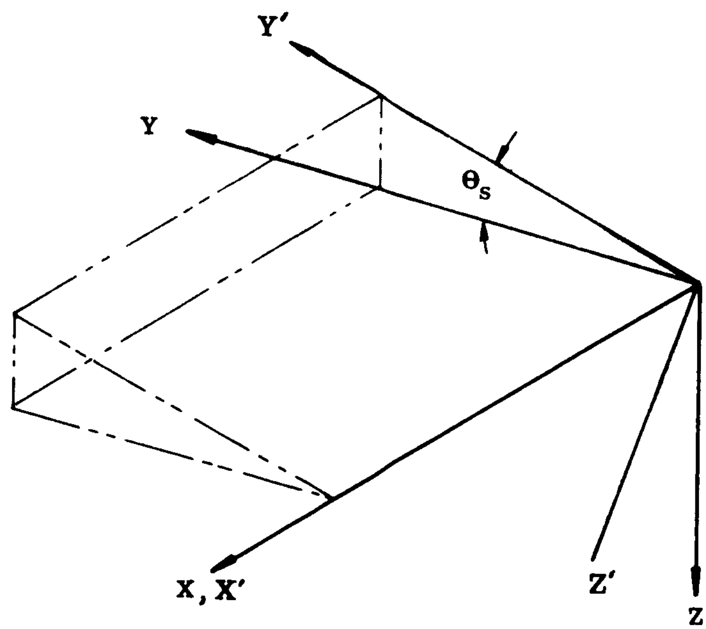



Figure 2-1. Ground and Surface Coordinate Systems

The vehicle coordinate system (Figure 2-2) has its origin at the vehicle CG, and moves with the vehicle. The z-axis is positive down along the vehicle centerline, the y-axis is positive in the direction of the No. 1 leg set, and the x-axis completes the triad. To relate the vehicle coordinates to a system parallel to ground coordinates, define the three rigid body rotations $\Psi$ (pitch), $\Phi$(yaw) and $\Xi$ (roll). Any angular orientation of the vehicle coordinate system can be obtained by imagining a coordinate system originally parallel to the ground axis. This imaginary system is brought into alignment with the vehicle axes by successively rotating $\Psi$ about the x-axis, $\Phi$ about the new position of the y-axis and $\Xi$ about the final position of the z-axis.

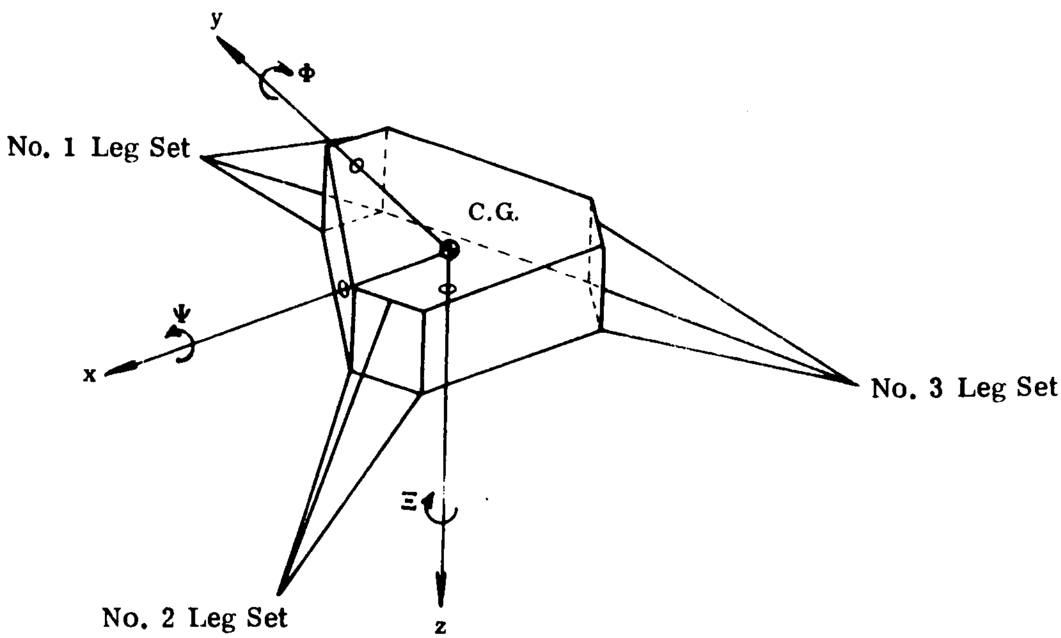



Figure 2-2. Vehicle Coordinate System

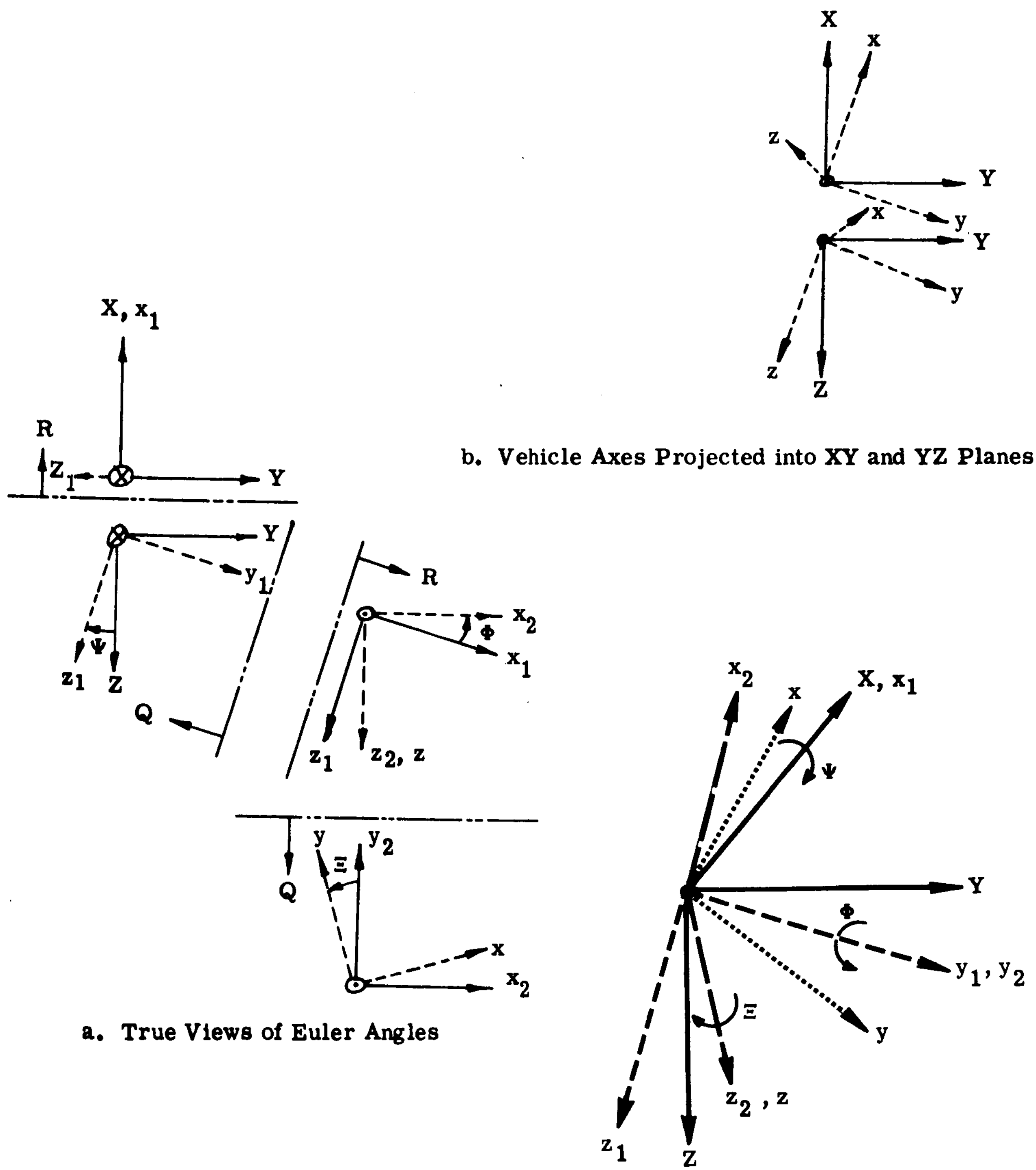
These rotations will be clarified in Figure 2-3, where two subsidiary coordinate systems are utilized in addition to the vehicle coordinates (x, y, z) and ground coordinates (X, Y, Z) already defined. Figure 2-3a depicts the successive rotations in true views. First the vehicle axes are considered to be in alignment with X, Y and Z, then rotated through $\Psi$ about X to form the new coordinate system $x_1$, $y_1$ and $z_1$, where $x_1$ coincides with X. The second rotation $\Phi$ about $y_1$ produces another set of coordinates $x_2$, $y_2$ and $z_2$ where $y_2$ and $y_1$ are coincident. The final rotation $\Xi$ about $z_2$ brings the coordinates into the vehicle coordinate system x, y and z where z coincides with $z_2$.

Figure 2-3b shows the vehicle axes projected into the XY and YZ planes of the ground coordinate system. Figure 2-3c illustrates these various coordinate systems in perspective.

A vector $\vec{a}$ expressed in vehicle coordinates can be transformed to coordinates parallel to ground coordinates by

$$\vec{A} = A_x \vec{I} + A_y \vec{J} + A_Z \vec{K} = \left[b_{rs}\right] \vec{a} \qquad (1)$$

where $\vec{a} = a_x \vec{i} + a_y \vec{j} + a_z \vec{k}$




b. Vehicle Axes Projected into XY and YZ Planes

a. True Views of Euler Angles

c. Coordinate Systems in Perspective

Figure 2-3. Development of Euler Angles

and

$$
\left[b_{rs}\right] = \begin{bmatrix} \cos\Phi\cos\Xi & -\cos\Phi\sin\Xi & \sin\Phi \\ \sin\Psi\sin\Phi\cos\Xi + \cos\Psi\sin\Xi & -\sin\Psi\sin\Phi\sin\Xi + \cos\Psi\cos\Xi & -\sin\Psi\cos\Phi \\ -\cos\Psi\sin\Phi\cos\Xi + \sin\Psi\sin\Xi & \cos\Psi\sin\Phi\sin\Xi + \sin\Psi\cos\Xi & \cos\Psi\cos\Phi \end{bmatrix}
$$

The notation of (1) will be used frequently and is to be interpreted as

$$
\begin{aligned}
A_x &= b_{11}\, a_x + b_{12}\, a_y + b_{13}\, a_z \\
A_y &= b_{21}\, a_x + b_{22}\, a_y + b_{23}\, a_z \\
A_z &= b_{31}\, a_x + b_{32}\, a_y + b_{33}\, a_z
\end{aligned}
$$

Since $\left[b_{rs}\right]$ is the matrix of an orthogonal transformation, the inverse $\left[b_{rs}\right]^{-1}$ is merely the transpose of $\left[b_{rs}\right]$.

Figures 2-4 and 2-5 showing the spacecraft geometry are also useful in visualizing two other moving coordinate systems which are defined for convenience. The strut coordinate systems are fixed in each of the lower links, with the origin at the footpad pivot. The y' axis is parallel to the lower link, positive outboard, the z' axis is perpendicular to the lower link, positive down, and the x' axis completes the triad. A vector $\vec{a}'$ in the jth strut coordinate system can be transformed to vehicle coordinates by

$$
\vec{a} = a_x\vec{i} + a_y\vec{j} + a_z\vec{k} = \left[g_{rs}\right]_j \vec{a}' \tag{2}
$$

where

$$
\vec{a}' = a'_x\vec{i}' + a'_y\vec{i}' + a'_z\vec{k}'
$$

and

$$
\left[g_{rs}\right]_j = \begin{bmatrix} \cos\theta_j & \cos\gamma_j \sin\theta_j & -\sin\gamma_j \sin\theta_j \\ -\sin\theta_j & \cos\gamma_j \cos\theta_j & -\sin\gamma_j \cos\theta_j \\ 0 & \sin\gamma_j & \cos\gamma_j \end{bmatrix}
$$

$\theta_j$ is positive in a left handed sense relative to the spacecraft z-axis.

The footpad coordinate systems are defined similarly to the strut coordinates, except for an additional rotation about the footpad hinge line. The z" axis is coincident with the footpad centerline, positive down. Then a vector $\vec{a}''$ in the jth footpad coordinate system can be transformed to vehicle coordinates by

$$
\vec{a} = a_x\vec{i} + a_y\vec{j} + a_z\vec{k} = \left[d_{rs}\right]_j \vec{a}'' \tag{3}
$$

where $$\vec{a''} = a''_x \vec{i''} + a''_y \vec{j''} + a''_z \vec{k''}$$

and

$$\left[d_{rs}\right]_j = \begin{bmatrix} \cos\theta_j & \cos(\gamma_j - \alpha_j)\sin\theta_j & -\sin(\gamma_j - \alpha_j)\sin\theta_j \\ -\sin\theta_j & \cos(\gamma_j - \alpha_j)\cos\theta_j & -\sin(\gamma_j - \alpha_j)\cos\theta_j \\ 0 & \sin(\gamma_j - \alpha_j) & \cos(\gamma_j - \alpha_j) \end{bmatrix}$$

$\alpha_j$ is positive in a left handed sense relative to the x' axis in strut coordinates.

One additional fixed coordinate system has been defined. The surface coordinate system differs from the ground coordinates only by a rotation $\theta_s$ about the X-axis. In surface coordinates the X' and Y' axis lie in the plane of the surface, and the Z' axis is normal to the surface, positive down. A vector $\vec{A'}$ in surface coordinates can be transformed to ground coordinates by

$$\vec{A} = A_x \vec{I} + A_y \vec{J} + A_z \vec{K} = \left[C_{rs}\right] \vec{A'} \tag{4}$$

where $$\vec{A'} = A'_x \vec{I'} + A'_y \vec{J'} + A'_Z \vec{K'}$$

and

$$\left[C_{rs}\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_s & \sin\theta_s \\ 0 & -\sin\theta_s & \cos\theta_s \end{bmatrix}$$

$\theta_s$ is positive in a left handed sense relative to X-axis in ground coordinates

## 2.3 SPACECRAFT GEOMETRY

The positional and kinematical configuration of the spacecraft is described by the equations to follow. Refer to Figures 2-1 through 2-5 for definition of the basic elements of the spacecraft geometry. Unless otherwise specified, the geometrical equations are written in vehicle coordinates.

The positions of the vehicle hardpoints relative to the CG are given by

(upper) $$\vec{r}_{hpu_j} = (\ell_p + \ell_A \sin\beta)\sin\theta_j \vec{i} + (\ell_p + \ell_A \sin\beta)\cos\theta_j \vec{j} + (\Delta CG - \ell_A \cos\beta)\vec{k} \tag{5}$$

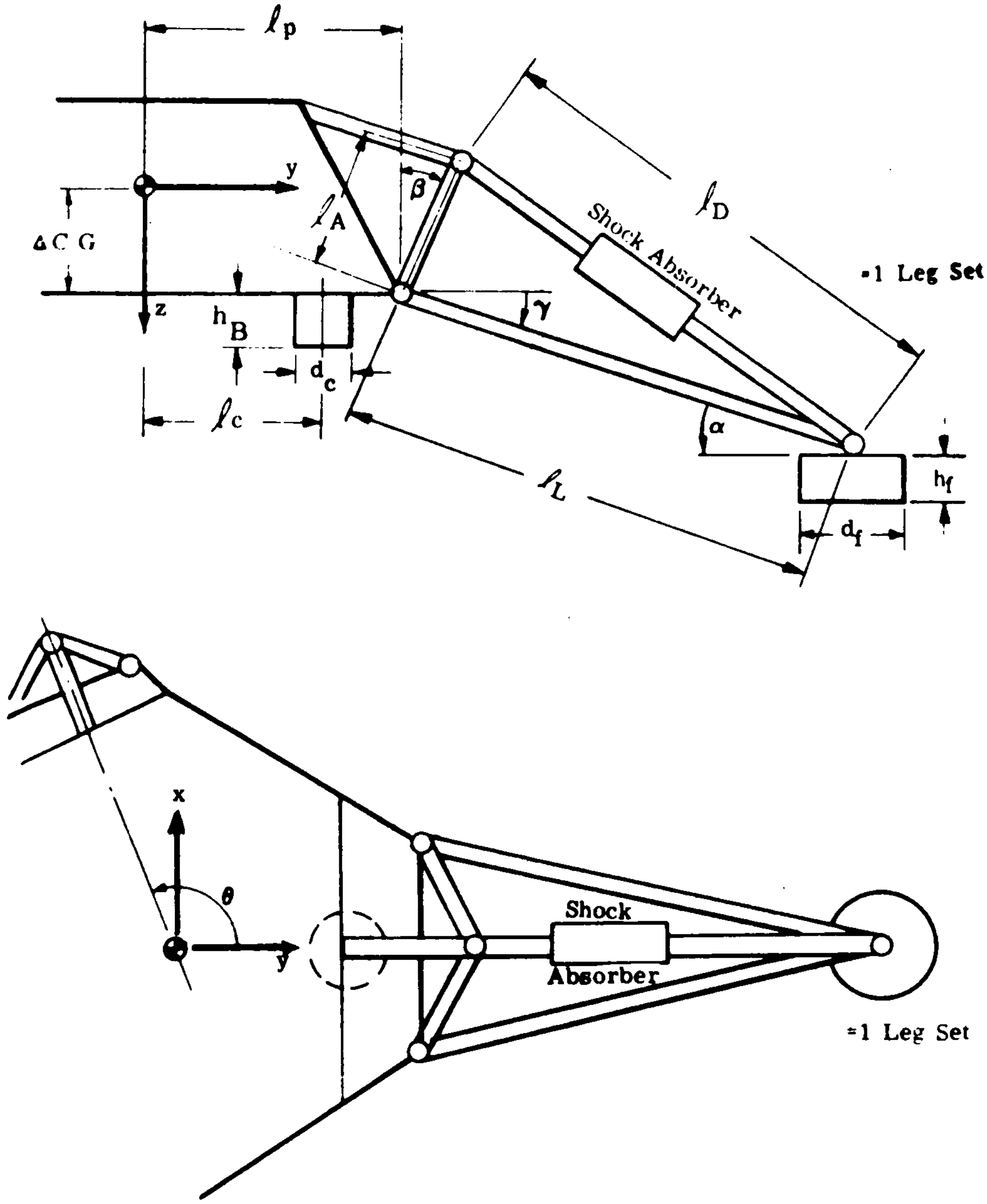



Figure 2-4. Vehicle Geometry

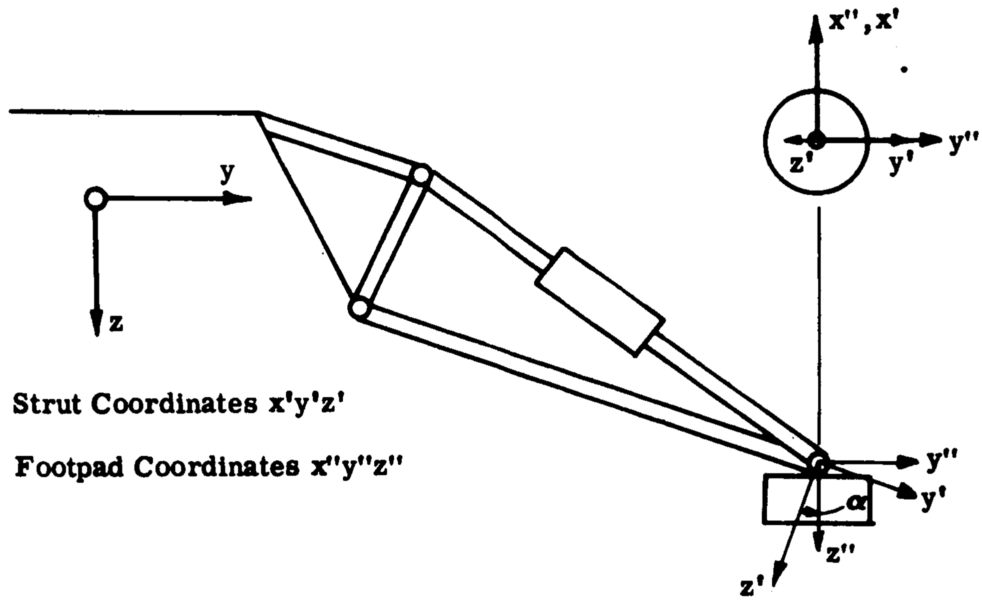



Figure 2-5. Strut and Footpad Coordinate System

(lower) $\overrightarrow{r_{hp\ell_j}} = \ell_p \sin\theta_j \vec{i} + \ell_p \cos\theta_j \vec{j} + \Delta CG \vec{k}$ (6)

The footpad pivots are described by

$$\overrightarrow{r_{fp_j}} = (\ell_p + \ell_L \cos\gamma_j)\sin\theta_j \vec{i} + (\ell_p + \ell_L \cos\gamma_j)\cos\theta_j \vec{j} + (\Delta CG + \ell_L \sin\gamma_j)\vec{k} \tag{7}$$

and

$$\overrightarrow{\dot{r}_{fp_j}} = -\ell_L(\sin\gamma_j \sin\theta_j \dot{\gamma}_j \vec{i} + \sin\gamma_j \cos\theta_j \dot{\gamma}_j \vec{j} - \cos\gamma_j \dot{\gamma}_j \vec{k}) + \vec{\omega} \times \overrightarrow{r_{fp_j}} \tag{8}$$

The velocity of the pivots in surface coordinates is

$$\overrightarrow{\dot{R}'_{fp_j}} = \dot{X}'_{fp_j} \vec{I'} + \dot{Y}'_{fp_j} \vec{J'} + \dot{Z}'_{fp_j} \vec{K'} = \left[C_{rs}\right]^{-1} \left\{ \overrightarrow{\dot{R}_{CG}} + \left[b_{rs}\right] \overrightarrow{\dot{r}_{fp_j}} \right\} \tag{9}$$

where

$$\overrightarrow{\dot{R}_{CG}} = \dot{X}\vec{I} + \dot{Y}\vec{J} + \dot{Z}\vec{K}$$

The sliding velocity comprises the components of (9) parallel to the surface

$$\dot{s}_{fp_j} = \sqrt{\dot{X}'^{2}_{fp_j} + \dot{Y}'^{2}_{fp_j}} \tag{10}$$

The length of the jth upper (shock absorber) link is described by

$$\ell_{D_j} = \left[\ell_A^2 + \ell_L^2 + 2\ell_A \ell_L \sin(\gamma_j - \beta)\right]^{1/2} \tag{11}$$

and

$$\dot{\ell}_{D_j} = \left[\ell_A \ell_L \cos(\gamma_j - \beta)\dot{\gamma}_j\right] / \ell_{D_j} \tag{12}$$

The surface of the jth crushable block (in the uncrushed condition) is located by

$$\overrightarrow{r_{Bo_j}} = \ell_c \sin\theta_j \vec{i} + \ell_c \cos\theta_j \vec{j} + (\Delta CG + h_{Bo})\vec{k} \tag{13}$$

The surface of each crushable footpad is separated into two subdivisions, one on either side of the hinge line. These subdivisions, shown in Figure 2-6, are described by the location of the centroids of the individual surface areas. In footpad coordinates, the location of the kth segment of the jth footpad (uncrushed) is

$$\overrightarrow{r_{co}''}_{jk} = x_{c_k} \vec{i}'' + y_{c_k} \vec{j}'' + h_{fo} \vec{k}'' \tag{14}$$

where

$$x_{c_k} = 0 \qquad k = 1, 2$$

$$y_{c_k} = 2\, d_f/3\pi \qquad k = 1$$

$$= -2\, d_f/3\pi \qquad k = 2$$

Then the position of the kth centroid of the jth footpad (uncrushed), measured relative to the CG in vehicle coordinates is obtained from (3), (7) and (14):

$$\overrightarrow{r_{co_{jk}}} = \overrightarrow{r_{fp_j}} + \left[d_{rs}\right]_j \overrightarrow{r_{co}''}_{jk} \tag{15}$$

The amount by which the various crushable elements are compressed is obtained by calculating the "interference" between an uncrushed element and the surface. Calculation of this interference requires determination of the vehicle position relative to the surface. Using (1) and (4), a unit vector normal (inward) to the surface can be expressed in vehicle coordinates by

$$\vec{n} = \left[b_{rs}\right]^{-1} \left[C_{rs}\right] \vec{K}' \tag{16}$$

and since a unit vector parallel to the crushable block centerline is $\vec{k}$ (vehicle coordinates), then the angle between the block centerline and the surface normal is obtained from

$$\cos \xi_B = \vec{n} \cdot \vec{k} \tag{17}$$

In surface coordinates the position of the center of the uncrushed block surface is

$$\overrightarrow{R'_{Bo_j}} = \left[C_{rs}\right]^{-1} \left\{ \overrightarrow{R_{CG}} + \left[b_{rs}\right] \overrightarrow{r_{Bo_j}} \right\} \tag{18}$$

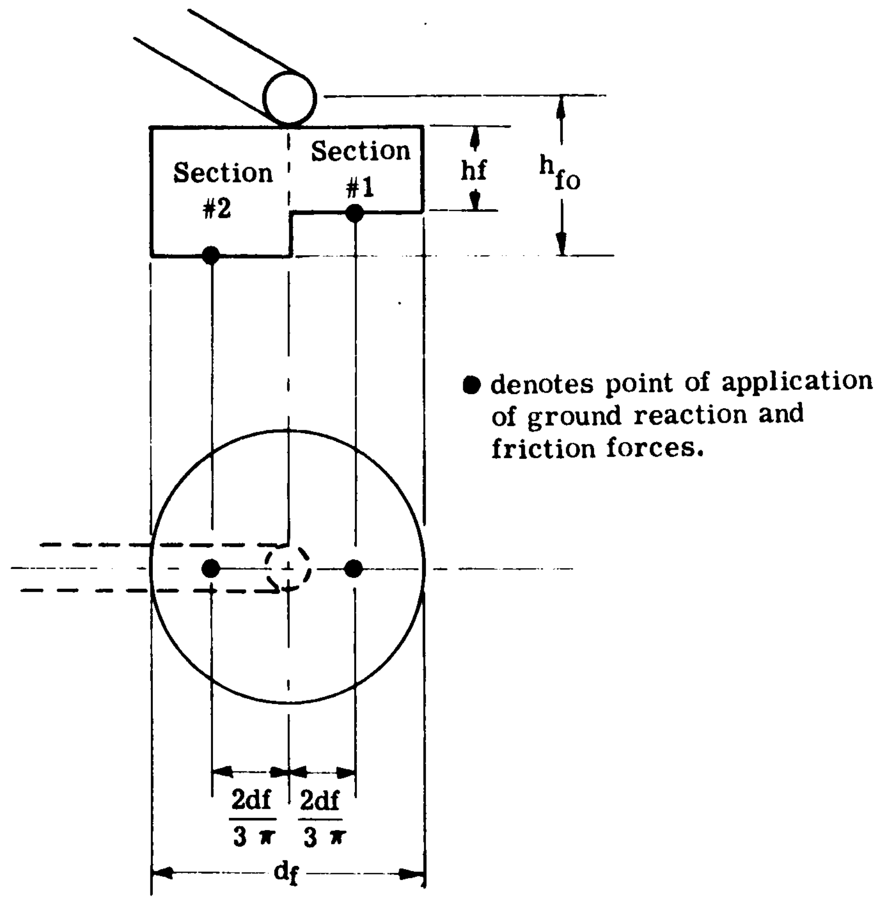



Figure 2-6. Footpad Sectioning

where

$$\overrightarrow{R_{CG}} = X\vec{I} + Y\vec{J} + Z\vec{K}$$

and we have used (13) together with $\overrightarrow{R_{CG}}$, the position vector of the CG in ground coordinates. The $Z'_{Bo_j}$ component of $\overrightarrow{R'_{Bo_j}}$ in (18) is measured normal to the ground surface and locates the surface of an uncrushed block relative to the surface. If $Z'_{Bo_j}$ is positive, "interference" between the ground and the crushable block is predicted. No actual interference is possible, naturally, but crushing will occur. The thickness of the crushed block is

$$h_{B_j} = h_{Bo} - Z'_{Bo_j} / \cos \xi_B \tag{19}$$

Then the center of the surface of the jth crushable block (crushed condition) is given in vehicle coordinates by

$$\overrightarrow{r_{B_j}} = \ell_c \sin \theta_j \vec{i} + \ell_c \cos \theta_j \vec{j} + (\Delta CG + h_{B_j}) \vec{k} \tag{20}$$

which is similar to (13). Since the vehicle coordinates constitute a moving coordinate system,

$$\dot{\overrightarrow{r_{B_j}}} = \vec{\omega} \times \overrightarrow{r_{B_j}} + \dot{h}_{B_j} \vec{k} \tag{21}$$

where

$$\vec{\omega} = \omega_x \vec{i} + \omega_y \vec{j} + \omega_z \vec{k}$$

and the velocity of the block surface in surface coordinates is

$$\dot{\overrightarrow{R'_{B_j}}} = \left[C_{rs}\right]^{-1} \left\{ \dot{\overrightarrow{R_{CG}}} + \left[b_{rs}\right] \dot{\overrightarrow{r_{B_j}}} \right\} \tag{22}$$

where $\dot{\overrightarrow{R_{CG}}}$ is the velocity of the CG in ground coordinates. But when the block is in contact with the surface, the component of $\dot{\overrightarrow{R_{CG}}}$ normal to the surface exactly cancels the corresponding component of $\dot{\overrightarrow{r_{B_j}}}$ so that (22) reduces to

$$\dot{\overrightarrow{R'_{B_j}}} = \dot{X}'_{B_j} \vec{I'} + \dot{Y}'_{B_j} \vec{J'} \quad \text{in surface coordinates} \tag{23}$$

where

$$\dot{X}'_{B_j} = \dot{X}_{CG} + b_{11}(\omega_y z_{B_j} - \omega_z y_{B_j}) + b_{12}(\omega_z x_{B_j} - \omega_x z_{B_j}) + b_{13}(\omega_x y_{B_j} - \omega_y x_{B_j} + \dot{h}_{B_j})$$

$$\dot{Y}'_{B_j} = \cos\theta_s \left[\dot{Y}_{CG} + b_{21}(\omega_y z_{B_j} - \omega_z y_{B_j}) + b_{22}(\omega_z x_{B_j} - \omega_x z_{B_j}) + b_{23}(\omega_x y_{B_j} - \omega_y x_{B_j} + \dot{h}_{B_j})\right]$$
$$- \sin\theta_s \left[\dot{Z}_{CG} + b_{31}(\omega_y z_{B_j} - \omega_z y_{B_j}) + b_{32}(\omega_z x_{B_j} - \omega_x z_{B_j}) + b_{33}(\omega_x y_{B_j} - \omega_y x_{B_j} + \dot{h}_{B_j})\right]$$

The sliding velocity of a block is

$$\dot{s}_{B_j} = \left|\overrightarrow{\dot{R}'_{B_j}}\right| \qquad (24)$$

when the block is in contact with the surface. It is convenient to neglect the small contribution of $\dot{h}_{B_j}$ in the determination of sliding velocity (24). While $\dot{h}_{B_j}$ can be calculated, several computer operations would be necessary to evaluate the terms. Since (24) is used primarily to establish the sliding friction coefficient (Figure 2-7), and since this coefficient is relatively insensitive to $\dot{s}_{B_j}$ above .05 ft/sec., the omission of $\dot{h}_{B_j}$ is justifiable.

Following an approach similar to that used for the blocks, a unit vector normal to the surface of the footpad is defined in vehicle coordinates

$$\overrightarrow{R_{fn_j}} = \sin\left[\alpha_j - \gamma_j\right] \sin\theta_j \vec{i} + \sin\left[\alpha_j - \gamma_j\right] \cos\theta_j \vec{j} + \cos\left[\alpha_j - \gamma_j\right] \vec{k} \qquad (25)$$

so that the angle between the footpad centerline and the surface normal is obtained from (16) and

$$\cos\xi_j = \overrightarrow{R_{fn_j}} \bullet \vec{n} \qquad (26)$$

The position of points on the surface of the jth footpad (uncrushed) in surface coordinates is obtained using (1), (4) and (15).

$$\overrightarrow{R'_{co_{jk}}} = \left[C_{rs}\right]^{-1} \left\{\overrightarrow{R_{CG}} + \left[b_{rs}\right] \overrightarrow{r_{co_{jk}}}\right\} \qquad (27)$$

The $Z'_{co_{jk}}$ component of (27) defines the "interference" measured normal to the surface, so that the thickness of the crushed kth section of the jth footpad is

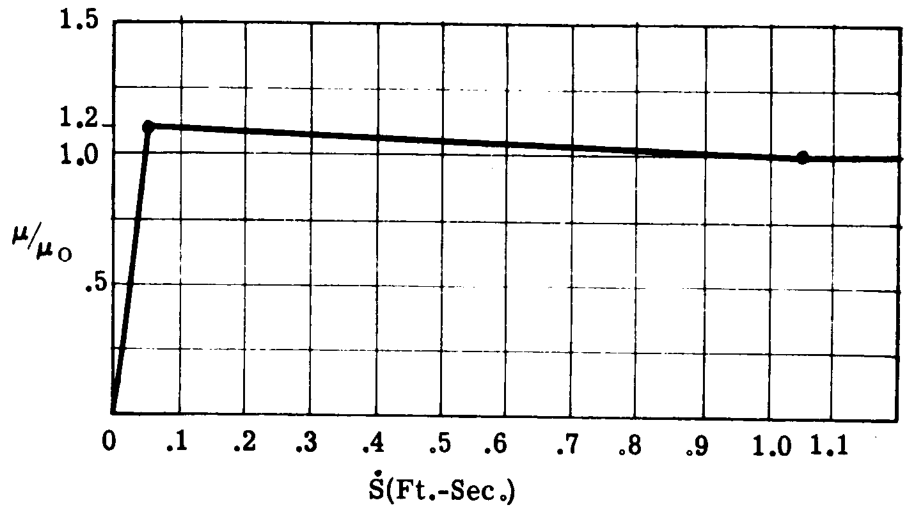



Figure 2-7. Sliding Friction Coefficient versus Velocity

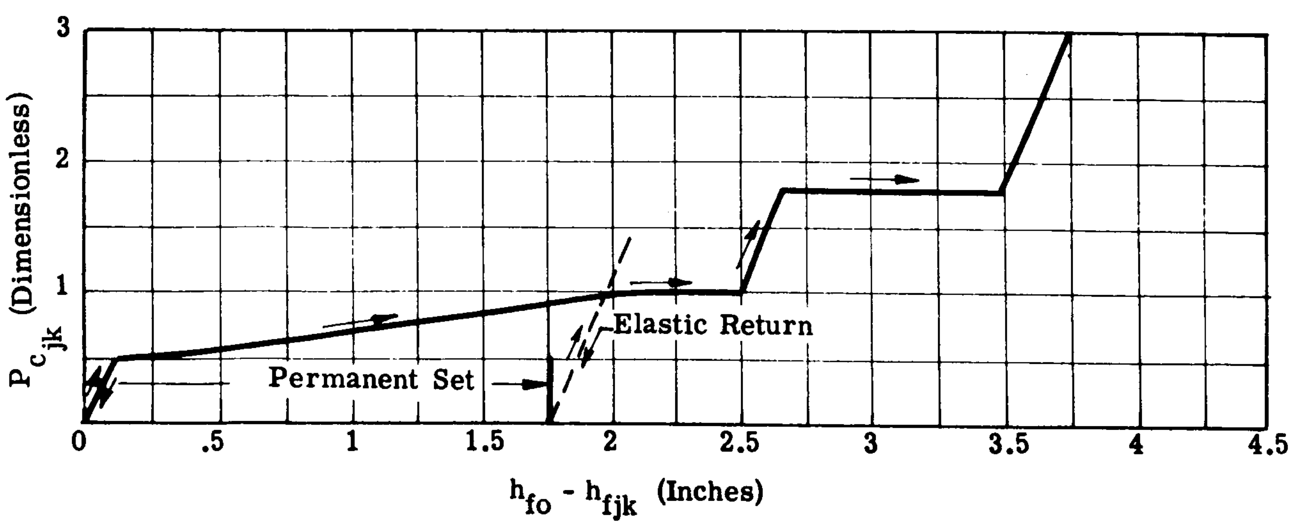



Figure 2-8. Footpad Crush Pressure Profile

$$h_{f_{jk}} = h_{fo} - Z'_{co_{jk}} / \cos \xi_j \tag{28}$$

## 2.4 DETERMINATION OF FORCES AND TORQUES

Forces generated in the footpads and crushable blocks are of two types: a crush force acting normal to the ground surface and a friction force due to footpad sliding. The friction force is in opposition to the sliding velocity (parallel to surface) and is proportional to the crush force. The factor of proportionality, or friction coefficient, is actually a function of the sliding velocity, as shown in Figure 2-7.

Figures 2-8 and 2-9 present the force-deflection characteristics of the footpads and blocks, respectively. These force-deflection characteristics are denoted $P_{c_{jk}}$ and $P_{cB_j}$, and define the crush pressure profile of the pads and blocks. The crushable elements follow an elastic relationship during unloading and reloading. Permanent set may exist in the crushable elements.

Taking into account the variation of footpad crush force and contact area with angle of loading (Reference (a))

$$F_{c\xi f_j} = \frac{\pi d_f^2}{8} \frac{C_f}{\cos \xi_j} (0.75 + 0.25 \cos 2 \xi_j) \sum_{k=1}^{2} P_{c_{jk}} \tag{29}$$

$$= C_{\xi_j} \sum_{k=1}^{2} P_{c_{jk}}$$

and using (16) the force vector acting on the footpad hinge due to crushing is

$$\overrightarrow{F_{c\xi f_j}} = -F_{c\xi f_j} \vec{n} \tag{30}$$

A unit vector in the direction of the sliding velocity in surface coordinates is obtained from (9) and (10)

$$\vec{t}_j = \frac{\dot{X}'_{fp_j}}{\dot{s}_{fp_j}} \vec{I'} + \frac{\dot{Y}'_{fp_j}}{\dot{s}_{fp_j}} \vec{J'} \tag{31}$$

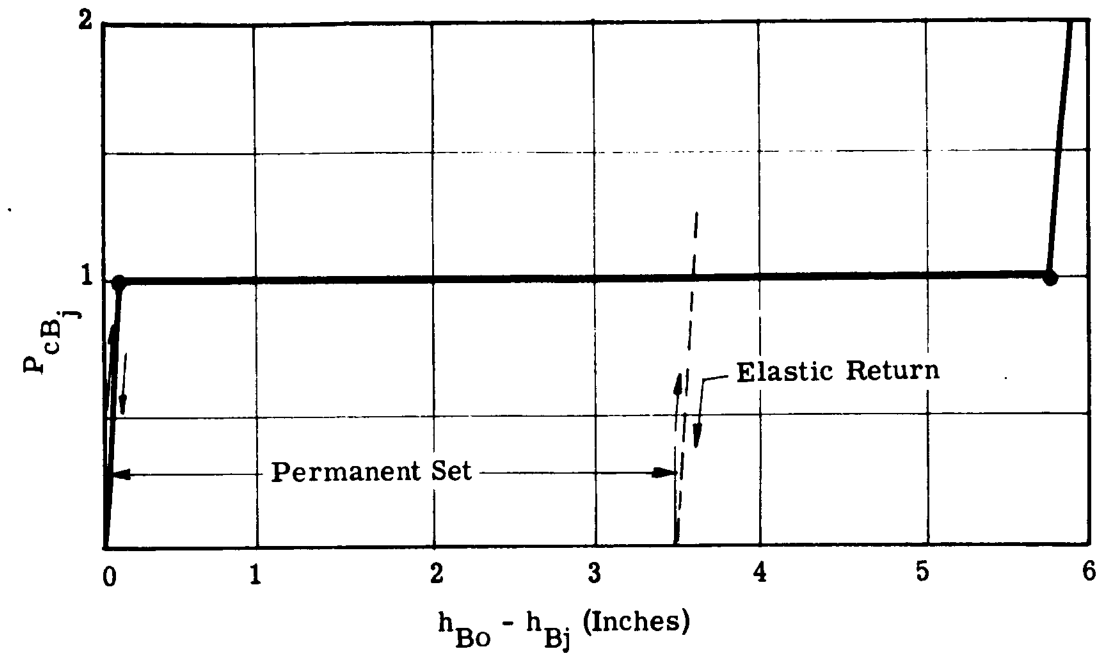



Figure 2-9. Block Crush Pressure Profile

Then the friction force, expressed in vehicle coordinates is

$$\overrightarrow{F_{\mu\xi f_j}} = -\mu_j F_{c\xi f_j} \left[b_{rs}\right]^{-1} \left[C_{rs}\right] \overrightarrow{t_j} \quad (32)$$

using (30), (31) and Figure 2-6.

Derivation of the forces due to the blocks parallels (29) through (32), with the added simplification that the block is not subdivided into sections.

The effect of loading angle on contact area is ignored since $\xi B_j$ must be a small angle (less than about 10° for a contacting block).

Then

$$F_{c\xi B_j} = \frac{\pi d_B^2}{4} C_B(0.75 + 0.25 \cos 2\xi_{B_j}) \, P_{cB_j} \quad (33)$$

and from (16)

$$\overrightarrow{F_{c\xi B_j}} = -F_{c\xi B_j} \overrightarrow{n} \quad (34)$$

A unit vector in the direction of the block sliding velocity is

$$\overrightarrow{t_{B_j}} = \frac{\overrightarrow{\dot{R}'_{B_j}}}{\dot{s}_{B_j}}$$

in surface coordinates, using (23) and (24). Then

$$\overrightarrow{F_{\mu\xi B_j}} = -\mu_{B_j} F_{c\xi B_j} \left[b_{rs}\right]^{-1} \left[C_{rs}\right] \overrightarrow{t_{B_j}} \tag{35}$$

in vehicle coordinates.

The gravity forces acting on the system (vehicle coordinates) are

$$\overrightarrow{F_{g_j}} = m_{p_j} g \left[b_{rs}\right]^{-1} \overrightarrow{k} \tag{36}$$

for the footpads (a small effect) and

$$\overrightarrow{F_G} = Mg \left[b_{rs}\right]^{-1} \overrightarrow{k} \tag{37}$$

for the main mass.

The summation of external forces in vehicle coordinates is then

$$\overrightarrow{F} = \overrightarrow{F_G} + \sum_{j=1}^{3} \left[\overrightarrow{F_{g_j}} + \overrightarrow{F_{c\xi f_j}} + \overrightarrow{F_{\mu\xi f_j}} + \overrightarrow{F_{c\xi B_j}} + \overrightarrow{F_{\mu\xi B_j}}\right] \tag{38}$$

from (30), (32), (34), (35), (36) and (37).

The torque about the spacecraft axes, due to the external forces is obtained with the help of (7) and (20)

$$\overrightarrow{T} = \sum_{j=1}^{3} \left\{\overrightarrow{r_{fp_j}} \times (\overrightarrow{F_{c\xi f_j}} + \overrightarrow{F_{\mu\xi f_j}}) + \overrightarrow{r_{B_j}} \times (\overrightarrow{F_{c\xi B_j}} + \overrightarrow{F_{\mu\xi B_j}})\right\} \tag{39}$$

The forcing functions expressed by (38) and (39) are sufficient to define the spacecraft rigid body rotations and motion of the CG. Other forces, however, must be accounted for in the determination of rotation of the lower link with respect to the spacecraft hard-points, and in prediction of footpad rotations.

Using (11), the deflection of the shock absorbers is defined:

$$\delta_j = \ell_{Do} - \ell_{D_j} \tag{40}$$

The shock absorber is preloaded so that initially, some compression exists in the metal, in the amount of

$$\delta_E = F_p/K_s \tag{41}$$

When $\delta_j \leq \delta_E$, the combined spring and preload force is

$$F_{s_j} = K_s \delta_j \tag{42}$$

When $\delta_j > \delta_E$, the combined spring and preload force is

$$F_{s_j} = K_D S_{K_j} (\delta_j - \delta_E) + F_p \tag{43}$$

where $S_{K_j}$, the nonlinear spring profile, is shown in Figure 2-10.

The shock absorber damping force is

$$F_{D_j} = -\dot{\ell}_{D_j} \left| \dot{\ell}_{D_j} \right| R_D S_{D_j} \tag{44}$$

where $S_{D_j}$, the nonlinear damping profile, is also shown in Figure 2-10. $R_D = R_c$ in compression and $R_D = R_R$ in rebound. $\dot{\ell}D_j$ is obtained from (12).

The mechanical friction force is

$$F_{f_j} = -\nu \frac{\dot{\ell}_{D_j}}{\left| \dot{\ell}_{D_j} \right|} \left| F_{s_j} \right| \tag{45}$$

Then the total shock absorber force is

$$F_{1_j} = F_{s_j} + F_{f_j} + F_{D_j} \tag{46}$$

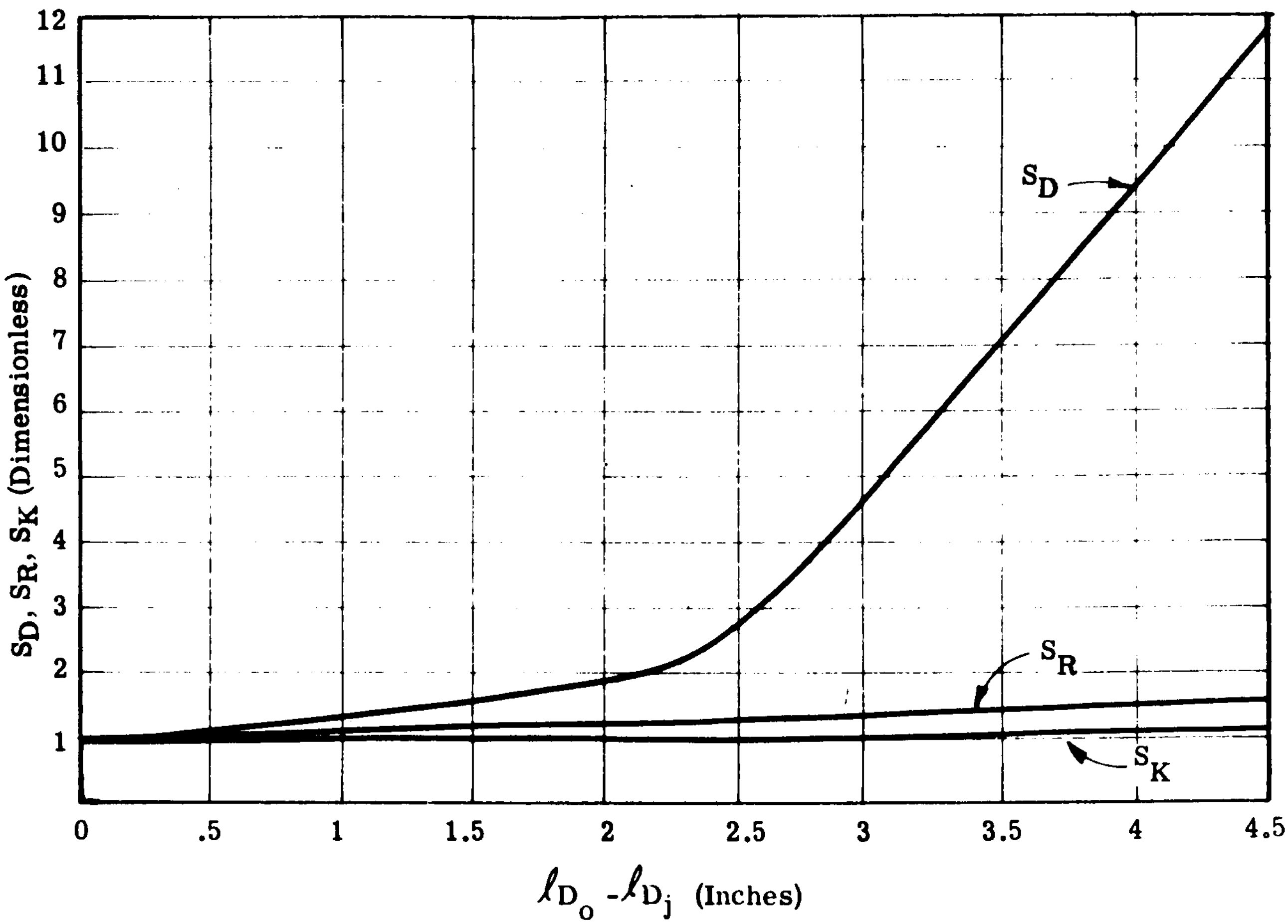



Figure 2-10. Shock Absorber Spring and Damping Profiles

Then using (5), (7), (11) and (46), the vector representing the shock absorber force on the footpad hinge (vehicle coordinate) is

$$\overrightarrow{F_{1_j}} = \frac{F_{1_j}}{\ell_{D_j}} \left(\overrightarrow{r_{fp_j}} - \overrightarrow{r_{hpu_j}}\right) \tag{47}$$

The sum of forces acting on the hinge in strut coordinates is obtained using (30), (32), (36) and (47).

$$\overrightarrow{R_j} = R_x\overrightarrow{i'} + R_y\overrightarrow{j'} + R_z\overrightarrow{k'} = \left[g_{rs}\right]_j^{-1} \left[\overrightarrow{F_{c\xi f_j}} + \overrightarrow{F_{\mu \xi f_j}} + \overrightarrow{F_{g_j}} + \overrightarrow{F_{1_j}}\right] \tag{48}$$

Only the $R_z$ component of (48) produces a torque about the lower hardpoint tending to change $\gamma_j$. $R_x$ and $R_y$ produce reactions. Then the torque tending to rotate the lower link is

$$T_{\gamma_j} = \ell_L R_{z_j} \tag{49}$$

The crush force on the kth section of the jth footpad is given in footpad coordinates by

$$\vec{F}''_{c_{jk}} = F''_{cx_{jk}} \vec{i}'' + F''_{cy_{jk}} \vec{j}'' + F''_{cz_{jk}} \vec{k}'' = -C_{\xi_j} P_{c_{jk}} \left[d_{rs}\right]_j^{-1} \vec{n} \tag{50}$$

using (16) and (29).

The friction force, expressed in footpad coordinates and using (29), (31) and (32) is

$$\vec{F}''_{\mu_{jk}} = F''_{\mu x_{jk}} \vec{i}'' + F''_{\mu y_{jk}} \vec{j}'' + F''_{\mu z_{jk}} \vec{k}'' = -\mu_j C_{\xi_j} P_{c_{jk}} \left[d_{rs}\right]_j^{-1} \left[b_{rs}\right]^{-1} \left[C_{rs}\right] \vec{t}_j \tag{51}$$

Then the torque, due to crushing and friction, tending to rotate the footpad about its hinge is

$$T^*_{\alpha_{jk}} = -y_{c_k} \left[F''_{cz_{jk}} + F''_{\mu z_{jk}}\right] + h_{f_{jk}} \left[F''_{cy_{jk}} + F''_{\mu y_{jk}}\right] \tag{52}$$

using (50) and (51). A positive torque denotes a tendency to increase $\alpha_j$. If the footpads are on the stops, additional torque will be generated by elastic forces

$$T_{s_j} = \begin{cases} -K_\alpha(\alpha_j - \alpha_1) & \alpha_j < \alpha_1 \\ 0 & \alpha_1 \le \alpha_j \le \alpha_2 \\ -K_\alpha(\alpha_j - \alpha_2) & \alpha_2 < \alpha_j \end{cases} \tag{53}$$

Then the total torque tending to rotate the footpad is

$$T_{\alpha_j} = \sum_{k=1}^{2} T^*_{\alpha_{jk}} + T_{s_j} \tag{54}$$

## 2.5 EQUATIONS OF MOTION

The forces and moments developed in the above equations constitute the forcing functions in the differential equations of motion. Using (38) the translational equations (uncoupled) of motion written in ground coordinates are

$$\left[b_{rs}\right] \vec{F} = M \ddot{\vec{R}}_{CG} \tag{55}$$

which can be integrated twice to determine CG position and velocity. The Euler equations for rotational motion utilize the torques presented in (39) and the definition of $\vec{\omega}$ in (21):

$$\vec{T} = \dot{\vec{L}} + \vec{\omega} \times \vec{L} \tag{56}$$

where the angular momentum $\vec{L}$ (vehicle coordinates) is

$$\vec{L} = [J] \vec{\omega} \tag{57}$$

also

$$\dot{\vec{L}} = [J] \dot{\vec{\omega}} \tag{58}$$

and

$$J = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{59}$$

It is convenient to rewrite (56) as

$$\dot{\vec{\omega}} = [J]^{-1} \left\{ \vec{T} - \vec{\omega} \times \vec{L} \right\} \tag{60}$$

which are three equations, uncoupled in $\dot{\omega}_x$, $\dot{\omega}_y$, $\dot{\omega}_z$.
The integration of (60) provides the angular velocity components $\omega_x$, $\omega_y$, $\omega_z$.

Before integrating a second time, the angular velocity components must be transformed to components of the time rate of change of the Euler angles.

To accomplish this, recall that the angular specification of the spacecraft requires successive rotation in $\Psi$ , $\Phi$ , and $\Xi$ (Figure 2-3). Keeping in mind the order of rotation,

$$\begin{aligned} \dot{\vec{\Psi}} &= \dot{\Psi} \vec{I} = \dot{\Psi} \vec{i}_1 = \dot{\Psi} (\cos \Phi \, \vec{i}_2 + \sin \Phi \, \vec{k}_2) \\ &= \dot{\Psi} \left[ \cos \Phi \, (\cos \Xi \, \vec{i} - \sin \Xi \, j) + \sin \Phi \, \vec{k} \right] \end{aligned} \tag{61}$$

$$\dot{\vec{\Phi}} = \dot{\Phi}\vec{j}_1 = \dot{\Phi}\vec{j}_2$$
$$= \dot{\Phi}\,(\sin \Xi\,\vec{i} + \cos \Xi\,\vec{j}) \qquad (61)\ \text{(Cont.)}$$

$$\dot{\vec{\Xi}} = \dot{\Xi}\vec{k}$$

But since

$$\vec{\omega} = \omega_x\vec{i} + \omega_y\vec{j} + \omega_z\vec{k} \qquad (62)$$
$$= \dot{\vec{\Psi}} + \dot{\vec{\Phi}} + \dot{\vec{\Xi}}$$

then

$$\begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = [A] \begin{Bmatrix} \dot{\Psi} \\ \dot{\Phi} \\ \dot{\Xi} \end{Bmatrix} \qquad (63)$$

where

$$[A] = \begin{bmatrix} \cos\Phi\,\cos\Xi & \sin\Xi & 0 \\ -\cos\Phi\,\sin\Xi & \cos\Xi & .0 \\ \sin\Phi & 0 & 1 \end{bmatrix} \qquad (64)$$

The determinant of $[A]$ is

$$\det[A] = \cos\Phi$$

Therefore $[A]^{-1}$ exists if $\cos\Phi \neq 0$. Note that in general, $\det[A] \neq 1$, so that $[A]^{-1}$ is <u>not</u> merely the transpose of $[A]$. Rather,

$$[A]^{-1} = \frac{1}{\cos\Phi} \begin{bmatrix} \cos\Xi & -\sin\Xi & 0 \\ \cos\Phi\,\sin\Xi & \cos\Phi\,\cos\Xi & 0 \\ -\sin\Phi\,\cos\Xi & \sin\Phi\,\sin\Xi & \cos\Phi \end{bmatrix} \qquad (65)$$

provided $\cos\Phi \neq 0$.

Then

$$\begin{Bmatrix} \dot{\Psi} \\ \dot{\Phi} \\ \dot{\Xi} \end{Bmatrix} = [A]^{-1} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \tag{66}$$

and (66) provides three equations, uncoupled in the Eulerian angular rates, which can be integrated to obtain $\Psi$, $\Phi$, and $\Xi$.

The restriction on the existence of $[A]^{-1}$, i.e. $\cos \Phi \neq 0$ does not constitute a serious limitation since the spacecraft will rarely experience this orientation without failing the stability test ( see para. 2.7). But since it is occasionally expedient to omit the stability test, further discussion of this point is in order.

Suppose $\cos \Phi = 0$ so that $\Phi = \pm \frac{2n-1}{2} \pi (n=1,2,......)$. Reference to Figure 2-3 will show that under these circumstances, the $\Psi$ and $\Xi$ rotations are performed about the same axis in space. Hence, the $\dot{\Psi}$ and $\dot{\Xi}$ angular rates cannot be distinguished. When $\cos \Phi = 0$, $\sin \Phi = \pm 1$, (63) and (64) give

$$\omega_x = \dot{\Phi} \sin \Xi$$

$$\omega_y = \dot{\Phi} \cos \Xi$$

from which

$$\dot{\Phi} = \sqrt{\omega_x^{\,2} + \omega_y^{\,2}}$$

also

$$\omega_z = \pm \dot{\Psi} + \dot{\Xi} \tag{67}$$

Then $\dot{\Xi}$ can be arbitrarily chosen (the previous value is convenient). From above

$$\pm \dot{\Psi} = \omega_z - \dot{\Xi}_{PREV}$$

where the sign of $\sin \Phi$ determines the proper choice of sign. The Euler angles can be obtained by integration as before.

With the help of (49) the equations of motion for the lower links are

$$T_{\gamma_j} = I_\gamma \ddot{\gamma}_j \tag{68}$$

It has been assumed that $\gamma_j$ can be considered as an instantaneous rotation about the lower hardpoint which is temporarily fixed in space. This assumption should be valid in light of the vast difference in the inertia properties of the vehicle body and the lower link.

Using (54) and the solutions of (68), the differential equations governing the footpad rotation are

$$T_{\alpha_j} = I_\alpha (\ddot{\alpha}_j - \ddot{\gamma}_j) \tag{69}$$

The integration of (55), (56), (68) and (69) provides new values for the kinematics and position of the spacecraft. These new values provide the starting point for the next pass through the equations developed in this section. A detailed discussion of the integration method will be found in Section III.

## 2.6 GROUND CLEARANCE

Spacecraft ground clearance is measured normal to the surface and represents the minimum distance between the lower hardpoints and the surface. Using (6), the positions of the lower hardpoints in surface coordinates are

$$\begin{aligned} R'_{hp\ell_j} &= X'_{hp\ell_j} \overrightarrow{I'} + Y'_{hp\ell_j} \overrightarrow{J'} + Z'_{hp\ell_j} \overrightarrow{K'} \\ &= \left[C_{rs}\right]^{-1} \left\{\overrightarrow{R_{CG}} + \left[b_{rs}\right] \overrightarrow{r_{hp\ell_j}}\right\} \end{aligned} \tag{70}$$

and the ground clearance is

$$GC = -Z'_{hp\ell} \tag{71}$$

where $-Z'_{hp\ell}$ is the minimum value of $-Z'_{hp\ell_j}$

## 2.7 STABILITY DETERMINATION

Evaluation of vehicle stability is accomplished through a subsidiary set of equations which parallel the determination of the vehicle motions. A particular landing is considered "unstable" if the spacecraft attains a toppling configuration. A landing is considered "stable" if the vehicle motions subside to such a degree that toppling is unlikely.

Figure 2-11 illustrates some of the geometry of the stability calculation. The "principal plane of motion" is defined by the vectors $\overrightarrow{R_{CG}}$ and $g$, a unit vector from the CG in the direction of the gravity force. All vectors in this discussion are expressed in ground coordinates except as otherwise indicated.

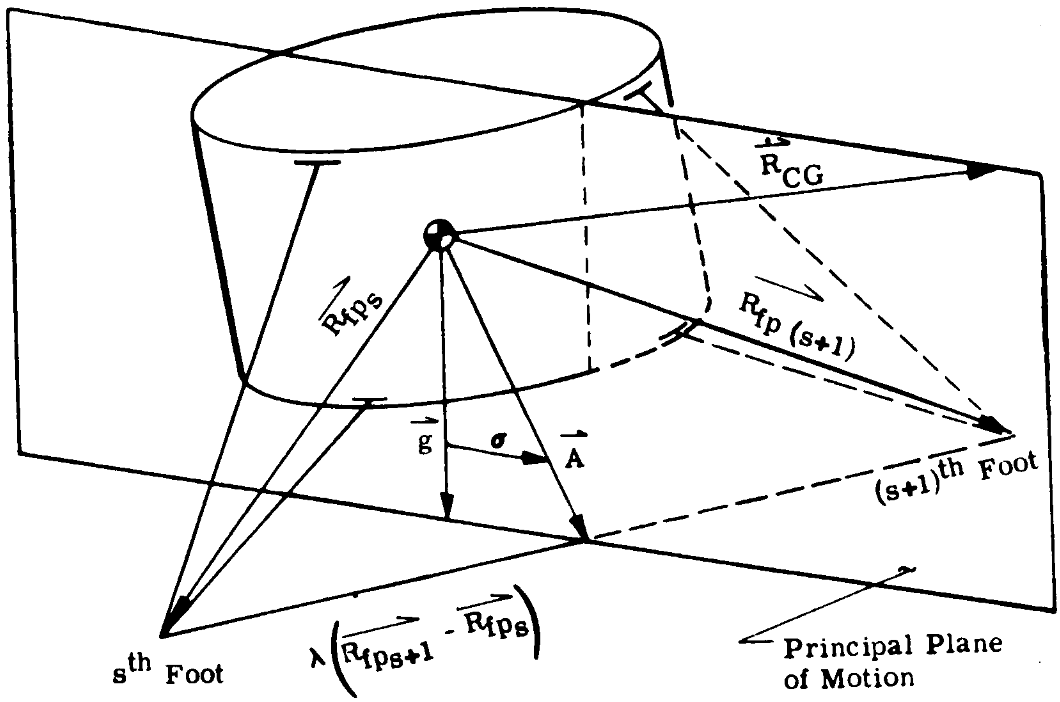



Figure 2-11. Stability Geometry

Toppling is considered imminent when $\vec{g}$ passes outboard of the triangle joining the three footpad pivot points, a situation which can be encountered in two ways. The most frequent situation will be "pitch instability" where the spacecraft topples in the direction of $\dot{\vec{R}}_{CG}$, with the two "critical feet" remaining astride of the principal plane. (At any given time the "critical feet" are that pair, astride of the principal plane, which are furthest advanced in the direction of the motion.) The second type of unstable configuration is "yaw instability." This unusual situation would occur if the vehicle were to rotate to a position with all three feet on the same side of the principal plane.

As long as $\vec{g}$ passes inboard of the feet, it is possible to define a meaningful, quantitative measure of stability. Define first the vector $\vec{A}$ in the principal plane, extending from the CG to the point at which the line joining the critical feet intersects the principal plane. The stability angle $\sigma$ is then formed by the vectors $\vec{g}$ and $\vec{A}$, remaining positive for stable configurations. For pitch instability the stability angle becomes zero at neutral stability, but for yaw instability the angle ceases to be defined at the neutral stability point.

The stability angle is evaluated as follows: Using (7), a vector from the CG to the footpad pivot (ground coordinates) is

$$\overrightarrow{R_{fp_j}} = \left[b_{rs}\right] \overrightarrow{r_{fp_j}} \tag{72}$$

The location of the feet with respect to the principal plane is determined by successive tests of the quantity

$$(\overrightarrow{\dot{R}_{CG}} \times \vec{g}) \cdot \overrightarrow{R_{fp_j}}$$

For a specific foot, this quantity will be positive if the foot is to the "left" of the principal plane (looking from the CG along $\overrightarrow{R_{CG}}$) and negative if the foot is to the "right." Then for any configuration in which the feet are astride of the principal plane, $(\overrightarrow{R_{CG}} \times \vec{g}) \cdot \overrightarrow{R_{fp_j}}$ will be either positive for exactly one specific foot, or negative for exactly one specific foot. In either case, denote the subscript for the foot $j=s$. This is one of the critical feet, and is located by $\overrightarrow{R_{fp_s}}$.

Considering the case where $(\overrightarrow{\dot{R}_{CG}} \times \vec{g}) \cdot \overrightarrow{\dot{R}_{fp_s}}$ is negative, the sth foot can be thought of as being to the right of the principal plane. Since the feet are numbered CCW (looking down), the other critical foot will be $j=s+1$, in this instance.

Denoting the vector from the sth to the (s+1)th foot by $\overrightarrow{R_{fp_{s+1}}} - \overrightarrow{R_{fp_s}}$ we have

$$\vec{A} = \overrightarrow{R_{fp_s}} + \lambda\,(\overrightarrow{R_{fp_{s+1}}} - \overrightarrow{R_{fp_s}}) \tag{73}$$

where $\lambda$ is a positive number between zero and one. To determine $\lambda$, form the scalar product of (73) with a unit vector $\vec{n}$ normal to the principal plane:

$$\vec{n} = \frac{\overrightarrow{\dot{R}_{CG}} \times \vec{K}}{|\overrightarrow{\dot{R}_{CG}} \times \vec{K}|} = \frac{\dot{Y}_{CG}\,\vec{I} - \dot{X}_{CG}\,\vec{J}}{\sqrt{\dot{X}_{CG}^{\,2} + \dot{Y}_{CG}^{\,2}}} \tag{74}$$

also

$$\vec{A} \cdot \vec{n} = \left[\overrightarrow{R_{fp_s}} + \lambda\,(\overrightarrow{R_{fp_{s+1}}} - \overrightarrow{R_{fp_s}})\right] \cdot \vec{n} \tag{75}$$

Since $\vec{A}$ and $\vec{n}$ are orthogonal, (74) and (75) yield the scalar equation

$$\lambda = \frac{\dot{X}_{CG} Y_{fp_s} - \dot{Y}_{CG} X_{fp_s}}{\dot{Y}_{CG} (X_{fp_{s+1}} - X_{fp_s}) - \dot{X}_{CG}(Y_{fp_{s+1}} - Y_{fp_s})} \tag{76}$$

so that from (73) and (76) $A_x, A_y, A_z$ (components of A) can be evaluated:

$$\begin{aligned} A_x &= X_{fp_s} + \lambda (X_{fp_{s+1}} - X_{fp_s}) \\ A_y &= Y_{fp_s} + \lambda (Y_{fp_{s+1}} - Y_{fp_s}) \\ A_z &= Z_{fp_s} + \lambda (Z_{fp_{s+1}} - Z_{fp_s}) \end{aligned} \tag{77}$$

from which

$$|\vec{A}| = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{78}$$

and the stability angle is

$$\sigma = \cos^{-1} \frac{A_z}{|\vec{A}|} \tag{79}$$

For positive $(\dot{\vec{R}}_{CG} \times \vec{g}) \cdot \vec{R}_{fp}$, the calculation of $\sigma$ is similar, except that the sth and (s+1)th feet are significant.

## 2.8 TOUCHDOWN GEOMETRY WHEN FOOTPAD IMPACT TIMES ARE SPECIFIED

The mathematical model developed in this section has been modified to permit calculation of spacecraft dynamics based on specified impact times for the footpads. The coordinates of the impact points of each foot are then used to calculate ground slope, cross slope angle and to transform the spacecraft angular orientation into the most convenient coordinate system.

Let the initial angular orientation be denoted $\Psi_o$, $\Phi_o$, and $\Xi_o$. The coordinates of the impact points are known in preliminary ground coordinates and are represented by

$$\vec{R}_j = X_j \vec{I}_o + Y_j \vec{J}_o + Z_j \vec{K}_o \tag{80}$$

An impact point is defined by the coordinates of the surface centroids of the uncrushed footpad sections No. 1 (Figure 2-6) at the respective impact times. These coordinates are in the preliminary ground coordinate system which is defined at zero time in the manner described on page 2-5.

The three impact points ($P_1$, $P_2$, $P_3$ corresponding to the respective footpads) form a plane. Define

$$\begin{aligned}\vec{A} &= A_x \vec{I}_o + A_y \vec{J}_o + A_z \vec{K}_o \\ &= (X_2 - X_1)\vec{I}_o + (Y_2 - Y_1)\vec{J}_o + (Z_2 - Z_1)\vec{K}_o \\ \vec{B} &= B_x \vec{I}_o + B_y \vec{J}_o + B_z \vec{K}_o \\ &= (X_3 - X_1)\vec{I}_o + (Y_3 - Y_1)\vec{J}_o + (Z_3 - Z_1)\vec{K}_o\end{aligned} \tag{81}$$

where $\vec{A}$ and $\vec{B}$ lie in the plane of the impact points (see Figure 2-12).

Obtain a unit inward normal vector to the surface from

$$\begin{aligned}\vec{n} &= n_x \vec{I}_o + n_y \vec{J}_o + n_z \vec{K}_o \\ &= \frac{\vec{A} \times \vec{B}}{|\vec{A} \times \vec{B}|}\end{aligned} \tag{82}$$

Check the direction of $\vec{n}$ by testing the component in the Z direction. If $n_z$ is positive, the desired vector has been defined. If $n_z$ is negative, reverse the signs of the components from (82).

Since $\vec{K}_o$ is a unit vector directed vertically down, the ground slope is given by

$$\cos \theta_s = \vec{n} \cdot \vec{K}_o = n_z \tag{83}$$

Next define the horizontal unit vector into the slope

$$\begin{aligned}\vec{a} &= a_x \vec{I}_o + a_y \vec{J}_o \\ &= -(\vec{n} \times \vec{K}_o) \times \vec{K}_o / \left|(\vec{n} \times \vec{K}_o) \times \vec{K}_o\right| \\ &= (n_x \vec{I}_o + n_y \vec{J}_o) / \sqrt{n_x^2 + n_y^2}\end{aligned} \tag{84}$$

If the horizontal velocity of the CG is defined as

$$\vec{V}_H = \dot{X} \vec{I}_o + \dot{Y} \vec{J}_o \tag{85}$$

then the cross slope angle is given by

$$\cos \Lambda = (\vec{V}_H \cdot \vec{a}) / |\vec{V}_H| \tag{86}$$

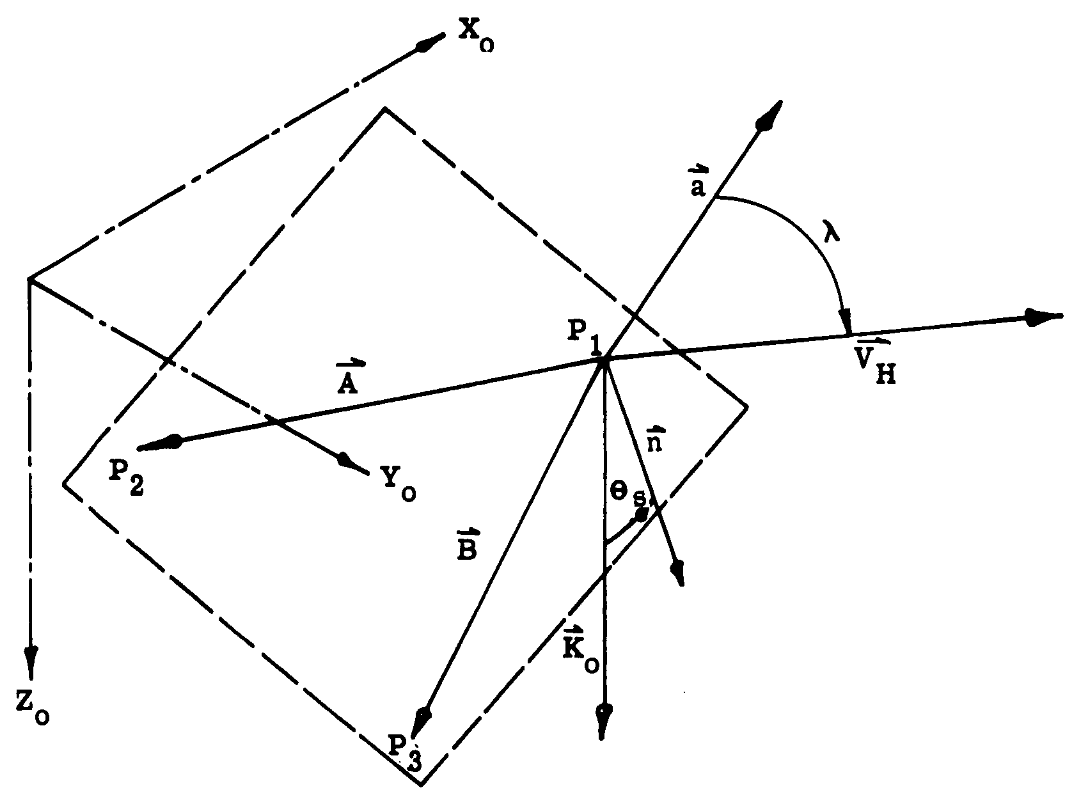



Figure 2-12. Determination of Cross Slope Angle and Ground Slope From Known Footpad Impact Points

It is convenient to specify the spacecraft angular orientation in a new ground coordinate system which is defined in the manner of Figure 2-1. The new ground coordinate system is obtained by rotating the preliminary system through an angle $\xi$ about the $Z_0$-axis (see Figure 2-13). The angle of rotation is given by

$$\cos \xi = \vec{a} \cdot \vec{J}_0 = a_y \tag{87}$$

The angle lies between ± 180° and is positive if the rotation of coordinates is positive in a right hand sense relative to the preliminary system. The sense of the rotation can be determined from the sign of $a_x$ from (84). A negative value of $a_x$ implies a positive rotation $\xi$. The Euler angles of the spacecraft rotation relative to the new ground coordinates can then be obtained from the rotation of coordinates about Z:

$$\begin{aligned}
\sin \Phi &= \sin \Phi_0 \cos \xi - \sin \Psi_0 \cos \Phi_0 \sin \xi \\
\sin \Psi &= \left[\sin \Phi_0 \sin \xi + \sin \Psi_0 \cos \Phi_0 \cos \xi\right] / \cos \Phi \\
\sin \Theta &= \left[\cos \Phi_0 \sin \Theta_0 \cos \xi - (\cos \Psi_0 \cos \Theta_0 \right. \\
&\quad \left. - \sin \Psi_0 \sin \Phi_0 \sin \Theta_0) \sin \xi\right] / \cos \Phi
\end{aligned} \tag{88}$$

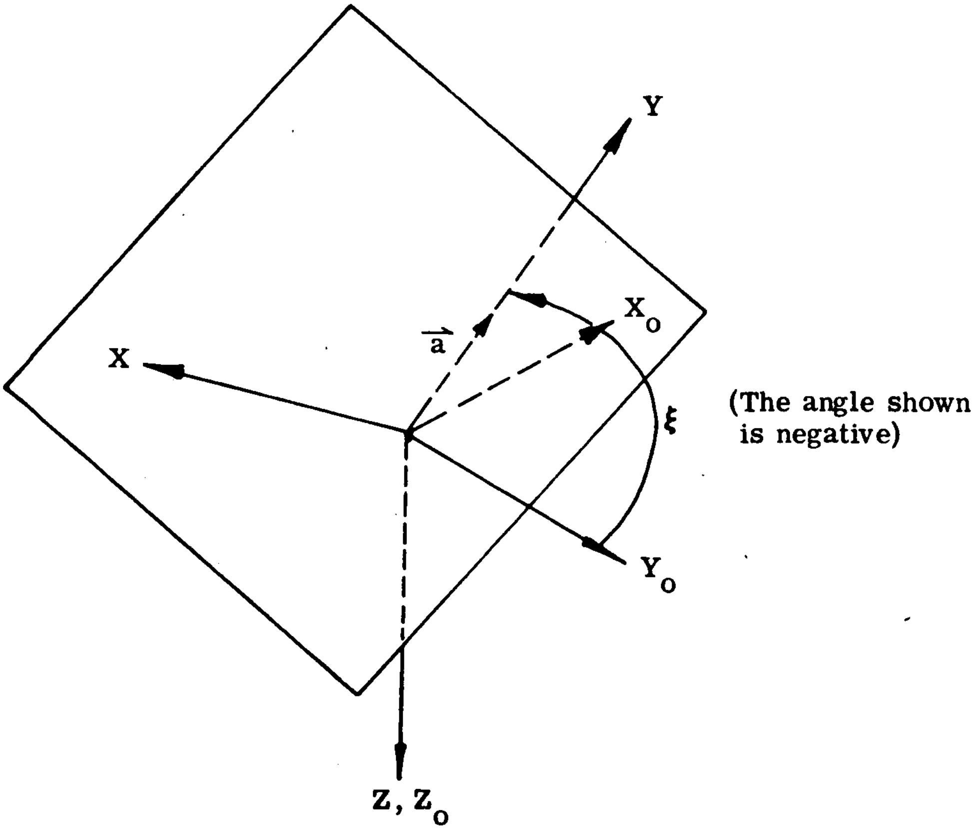



Figure 2-13. Rotation of Preliminary Ground Coordinates

## SECTION III

## COMPUTER SIMULATION OF HARD SURFACE LANDING

### 3.1 FLOW THROUGH COMPUTER PROGRAM

This particular computer program differs somewhat from similar digital computer simulations in that the integration subroutine controls the program after a relatively simple initialization phase (see Figure 3-1 for a basic block diagram of the program). That is, the integration subroutine (RKP) receives the initial conditions, performs the required integration, prepares values for printing, error checks the integration and, in general, controls the operation of the program. Two subprograms are required by the integration subroutine. One is a subprogram to calculate an array of derivatives from an array of integrated variables. This is handled by subroutine FEP and various subroutines which it calls. The second is a subprogram which will receive the integrated variables in an array and translate them into the required output format. This is done in subroutine PRT.

At the beginning of the program, a series of four tables are read in as input data by the subroutine TABLIN. These tables define characteristics of the footpads, crushable blocks and the shock absorbers. After the table-read, the data describing the initial configuration and conditions are read by subroutine INPUT. This information is then printed on the summary sheet. The final subroutine which is called in this initialization phase is subroutine INIT. This subroutine adjusts the input variables for units (degrees to radians, for instance), calculates and/or sets various constants which are used later in the program and calculates additional initial information based on the input configuration and conditions. The three subroutines, TABLIN, INPUT, and INIT, are called from the main program. After these subroutines have been called, the YI array of variables is set in the main program. This array in part consists of the initial values of the twenty-four variables to be integrated during the course of the program. These initial values, established in subroutines INPUT and INIT, comprise the positions and velocities of the CG, rigid body rotation about the CG, rotations of the three lower struts and rotations of the three footpads. The remaining elements of the YI array are zeroed in preparation for calculations to determine the work done on the footpads, blocks, and shock absorbers. Either five or fifteen elements of the YI array are used for work calculations, depending upon which output option is activated (see Subsection 3.4.4.3). Next, an error (ERR) array is established. This array contains total allowable truncation errors permissible per second of simulation time. Upon completion of this task, the main program transfers control unconditionally to the integration subroutine, RKP.

Subroutine RKP performs a multitude of tasks including a fair amount of housekeeping as well as integration, its primary function. It must set or respond to various flags which signal events, such as the termination of program flag. It must generate the independent

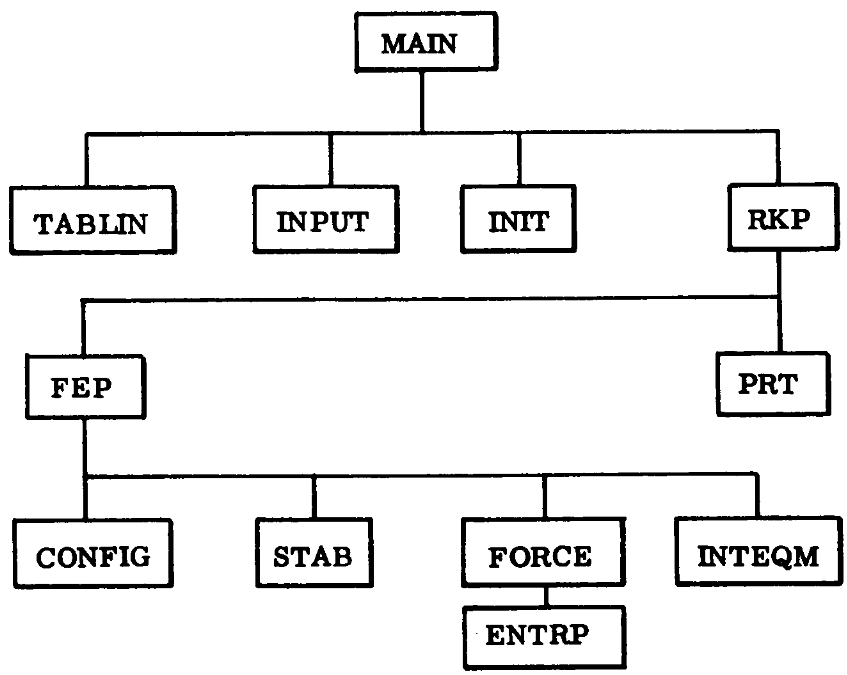



**Figure 3-1. Block Diagram of Overall Computer Program**

variable (time). It must calculate the truncation error, compare the error to the allowable error and adjust the integration interval in accordance with the size of this error. It must ascertain whether output printing is required during an integrated interval and if so, interpolate from the integrated values a set of values consistent with the instant in which output information was required. It must determine if the input information received from the MAIN program is meaningful within rather broad boundaries and last, but not least, subroutine RKP must be able to integrate efficiently and accurately.

The integration subroutine RKP calls only two other subroutines: FEP and PRT. FEP (Function Evaluation Program) is required to have the capability of computing an array (D) of derivatives of the Y array at the time, T, when supplied with the values of the Y array at time, T. A further requirement on FEP is that the derivative array must be independent of past history and calculations and only dependent on the Y array and initial constants. The subroutine PRT is required to accept output information for printing at a rate specified by the PRT subroutine.

Subroutine FEP receives an array (Y) of integrated values from subroutine RKP. In FEP, these values are set equal to others whose symbolism is somewhat more meaningful (e.g. ALPH(1) = Y(4)). This information is then transferred to a subroutine which calculates the configuration of the vehicle in detail (subroutine CONFIG). Based on the configuration, a second subroutine makes a determination of the stability of the vehicle (STAB). Next, calculations of the footpad, block and shock absorber forces as functions of the configuration are performed. These forces are used to calculate the torques on the lower struts, the torques about the CG and the forces affecting the CG. These rather lengthy calculations are performed in subroutine FORCE.

The forces and torques in conjunction with the input mass and inertia terms are combined in the differential equations of motion to generate the derivatives. These calculations are done in subroutine INTEQM. Finally these derivatives are appropriately transferred to the derivative (D) array at the end of subroutine FEP.

In summary, the MAIN program calls the subroutines TABLIN, INPUT and INIT for initialization. The MAIN program transfers control to subroutine RKP. RKP calls subroutine FEP which in sequence calls CONFIG, STAB, FORCE and INTEQM. Subroutine FEP basically acts as an interface between the integration routine, RKP, and the CONFIG, STAB, FORCE and INTEQM subroutines. This procedure is repeated and the integration proceeds. As output information is generated, it is transferred to subroutine PRT.

## 3.2 INTEGRATION METHOD

The integration method used in this program may be described as a variable interval, error checking Runge-Kutta integration method. The basic integration method is a fourth order Runge-Kutta integration procedure.

With the constant interval Runge-Kutta process, four evaluations of the derivatives are required in order to integrate over a given interval. However, the procedure in this program requires eleven evaluations of the derivatives to integrate over a given interval. This would appear to entail a large running time penalty over other methods. However, this has not been our experience.

In a given interval, a fourth order Runge-Kutta will require evaluations of derivatives at the beginning of the interval, two in the middle of the interval and one at the end of the interval. The method which is used in this program does this and additionally cuts the given interval in half and repeats this four-step process twice more to again integrate over the original full interval. This effectively results in two answers to the same problem. Both of these answers, by their very nature, are in error from the exact answer. The difference between the two answers is proportional to the major portion of the truncation error in the smaller interval integration. The difference is used to improve the answer derived from integrating over the two half-intervals. This improved answer is theoretically as accurate as a fifth-order Runge-Kutta procedure. The remaining truncation error is then calculated and compared to the allowable error. If the error is less than the allowable error, the next integration interval will be expanded proportionately. If the error is greater than the allowable error, the interval size is cut in half. This permits the use of the derivatives calculated in the first half interval and seven, instead of eleven, more sets of derivatives are required. The error checking procedure is repeated and the evaluation process is repeated.

Since the Runge-Kutta procedure is performed completely three times during a given interval, twelve evaluations of the derivatives would appear to be required. However, the derivative array calculated at the beginning of the full interval will also be identical to the derivatives required for the beginning of the first half-interval; therefore, recalculation is unnecessary. This also is the explanation of why seven instead of eight evaluations of the derivatives are required in the event that an interval must be cut in half.

This method is particularly well adapted to the Surveyor program. A typical landing on a hard surface will normally involve one or more abrupt contacts with the surface resulting in generally discontinuous derivatives and invariably a free flight phase where the derivatives are essentially constant and in some cases zero. During an abrupt contact, the integration interval will be reduced,and during a free flight phase, the interval will be broadened. Regardless of the situation, the accuracy of the calculation remains within the specified limits.

Other integration methods have been investigated, but for this particular problem, none have surpassed the variable interval Runge-Kutta integration method. Another variable interval method, Milne integration with Runge-Kutta starting, works quite well in problems where the derivatives experience only occasional abrupt changes. In the case of the Surveyor landing, however, abrupt changes are not unusual so that the Milne integration routine is not often able to choose a larger integration interval. The reluctance of the Milne routine to take a larger time interval stems from the conservative interval adjustment policy which must be followed in order to avoid a premature increase in interval size, and the resulting running time penalty for restarting.

Constant interval methods ordinarily will be suffering inaccuracies during periods of abrupt change due to an insufficiently fine time interval. During smooth periods, such as free flight, these methods will be "loafing" in their failure to take full advantage of the situation. Even so, good results have been obtained with a constant interval Runge-Kutta ($\Delta t = .002$ sec.), but no advantage in running time has resulted. Trapezoidal integration improves running time only in the event that time increments are dangerously large from the standpoint of accuracy.

## 3.3 ERROR CHECKING PROCEDURE

The basis of the error evaluation procedure is the foundation of the Runge-Kutta method, the Taylor Series. In general, the order of a Runge-Kutta integration procedure is a function of the last term of the Taylor Series which is included in its derivation. A Taylor Series may be written:

$$y_1 = y_o + hy_o^{I} + \frac{1}{2!} h^2 y_o^{II} + \frac{1}{3!} h^3 y_o^{III} + \frac{1}{4!} h^4 y_o^{IV} + \frac{1}{5!} h^5 y_o^{V} + \frac{1}{6!} h^6 y_o^{VI} + ----$$

where: $h$ is the interval of the independent variable,

$y_o$ is the value of the dependent variable at the beginning of the interval, h,

$y_1$ is the value of the dependent variable at the end of the interval, h,

and $y_o^n$ is the n-th derivative of the dependent variable at the beginning of the interval.

A fourth order Runge-Kutta procedure is used so the truncation error is:

$$e_1 = \frac{1}{5!} h^5 y_o^{V} + \frac{1}{6!} h^6 y_o^{VI} + ----$$

The error for an interval half as large would be:

$$e_{\frac{1}{2}} = \frac{1}{5!} \left(\frac{h}{2}\right)^5 y_o^{V} + \frac{1}{6!} \left(\frac{h}{2}\right)^6 y_o^{VI} + -----$$

$$= \frac{1}{5!} \left(\frac{1}{32}\right) h^5 y_o^{V} + \frac{1}{6!} \left(\frac{1}{64}\right) h^6 y_o^{VI} + -----$$

The total error for two half-intervals would be equal to $2 \cdot e_{\frac{1}{2}}$ or:

$$e_2 = \frac{1}{5!} \left(\frac{1}{16}\right) h^5 y_o^{V} + \frac{1}{6!} \left(\frac{1}{32}\right) h^6 y_o^{VI} + -----$$

Since $y_o^{V}$, and $y_o^{VI}$, etc.,are not known, the exact value of these errors cannot be evaluated. Let the exact value of the integrated variable at the end of the interval h be $y_1$. Then the value calculated by Runge-Kutta using one step over the interval will be $y_1 + e_1$ and using two half-intervals will be $y_1 + e_2$. If the difference between these values is calculated, the result is:

$$
\begin{aligned}
& (y_1 + e_1) - (y_1 + e_2) \\
= \quad & e_1 - e_2 \\
= \quad & \frac{1}{5!} h^5 y_o^{\ V} + \frac{1}{6!} h^6 y_o^{\ VI} + \text{----} \\
& -\left[\frac{1}{5!}\left(\frac{1}{16}\right) h^5 y_o^{\ V} + \frac{1}{6!}\left(\frac{1}{32}\right) h^6 y_o^{\ VI} + \text{------}\right] \\
= \quad & \frac{1}{5!}\left(\frac{15}{16}\right) h^5 y_o^{\ V} + \frac{1}{6!}\left(\frac{31}{32}\right) h^6 y_o^{\ VI} + \text{------}
\end{aligned}
$$

It now may be noted that the first term is exactly fifteen times the fifth order error in $e_2$. If one-fifteenth of the difference, $e_1 - e_2$, is now subtracted from $e_2$, the remaining error, $e_3$, is now of the order of:

$$
\begin{aligned}
& e_3 = e_2 - \frac{1}{15}(e_1 - e_2) \\
= \quad & \frac{1}{6!}\left[\left(\frac{1}{32}\right) - \left(\frac{31}{15 \cdot 32}\right)\right] h^6 y^{\ VI} + \text{----} \\
= \quad & \frac{1}{6!}\left(\frac{-1}{30}\right) h^6 y^{VI} + \text{------}
\end{aligned}
$$

Since $e_3$ is the remaining error in our calculation, we require that the absolute value of this number be less than or equal to the allowable error; that is:

$$
|e_3| \leq \text{(allowable error)}
$$

The procedure which is now used starts with the calculation of the difference between the improved integrated value, $y_1 + e_3$, and the integrated value resulting from integrating over the entire interval, $y_1 + e_1$. That is:

$$
\begin{aligned}
& (y_1 + e_1) - (y_1 + e_3) \\
= \quad & e_1 - e_3 \\
= \quad & \left[\frac{1}{5!} h^5 y_o^{\ V} + \frac{1}{6!} h^6 y_o^{\ VI} + \text{-----}\right] - \left[\frac{1}{6!}\left(-\frac{1}{30}\right) h^6 y_o^{\ VI} + \text{-------}\right] \\
= \quad & \frac{1}{5!} h^5 y_o^{\ V} + \frac{1}{6!}\left(\frac{31}{30}\right) h^6 y_o^{\ VI} + \text{------} \\
\simeq \quad & \frac{1}{5!} h^5 y_o^{\ V}
\end{aligned}
$$

In a reasonably well-behaved Taylor Series, the quantity $h^6 y_0^{VI}$ would be very much less than $h^5 y_0^{V}$. For the following, the assumption is made that $h^6 y_0^{VI}$ is, at worst, equal to $h^5 y_0^{V}$. This is believed to be a conservative assumption. With the aid of this assumption, an evaluation of $e_3$ will be made in terms of $(y+e_1) - (y+e_3)$.

$$
\begin{aligned}
e_3 &\simeq \frac{1}{6!}\left(-\frac{1}{30}\right) h^6 y^{VI} \\
&\simeq \frac{1}{6!}\left(-\frac{1}{30}\right) h^5 y^{V} \\
&\simeq \frac{-1}{180}\left(\frac{1}{5!} \quad h^5 y^{V}\right) \\
&\simeq -\frac{1}{180}\left((y_1 + e_1) - (y_1 + e_3)\right)
\end{aligned}
$$

Now, returning to our error criterion:

$$\left| e_3 \right| \leq \text{(allowable error)}$$

Substituting for $e_3$:

$$1/180 \cdot \left|(y_1 + e_1) - (y_1 + e_3)\right| \leq \text{(allowable error)}$$

This method has been used and found to be satisfactory.

As the program presently stands, elimination of the fifth-order truncation error is performed on all integrated variables. Error checking is performed on twenty-one. No checks are performed on the three footpad angular rates and on the work done on the system. The accuracy of the work integration in no way affects the accuracy of the program. It is used as a check to see if the work done plus the final kinetic and potential energy equals the initial kinetic and potential energy. The required accuracy of this calculation is minimal. The footpad calculations have been modified to such an extent that a check on the footpad angular rate error is meaningless. Error checks are performed on footpad angular position, however.

The integration interval is adjusted on the basis of the error checks. If the error check fails, the interval is cut in half. The calculations pertaining to the first half-interval are now used as the values for the full interval and two new half-interval calculations (based on the smaller interval) are then computed. If the error check fails a second time, the interval is again cut in half and this process is repeated until one of two things occur: the error is passed or the interval is reduced to a preset minimum. If this situation occurs, further interval reduction is futile since the number of significant figures available on the Univac 1107 are insufficient for a meaningful comparison of the errors. Under these circumstances, the error check is by-passed after a diagnostic note on the summary sheet is printed.

If the error check is passed, a new interval is started. The size of this new interval is a function of the maximum absolute value of the ratio of the truncation error to the allowable error in the present interval. For a situation in which the error check has been passed, this absolute ratio must be between 0 and 1. The trial size of the next interval will be:

$$h_{1,2} = (2 - U_{2\ max.}) \cdot h_{0,1}$$

where $U_{2\ max.}$ is the aforementioned ratio, $h_{0,1}$ is the present interval size and $h_{1,2}$ is the new trial interval size. For an error equal to the truncation error, the interval size remains the same. If no error occurs, the interval will be doubled. The remainder of the possibilities lie between these two extremes.

## 3.4 PROGRAM DETAIL

This sub-section describes the usage and composition of the computer program in sufficient detail to enable the user to understand thoroughly the content of the program, and to apply the program to studies of Surveyor hard surface landings. To supplement the very general description of the program flow appearing in 3.1, additional descriptions at two increased levels of detail are presented on the following pages.

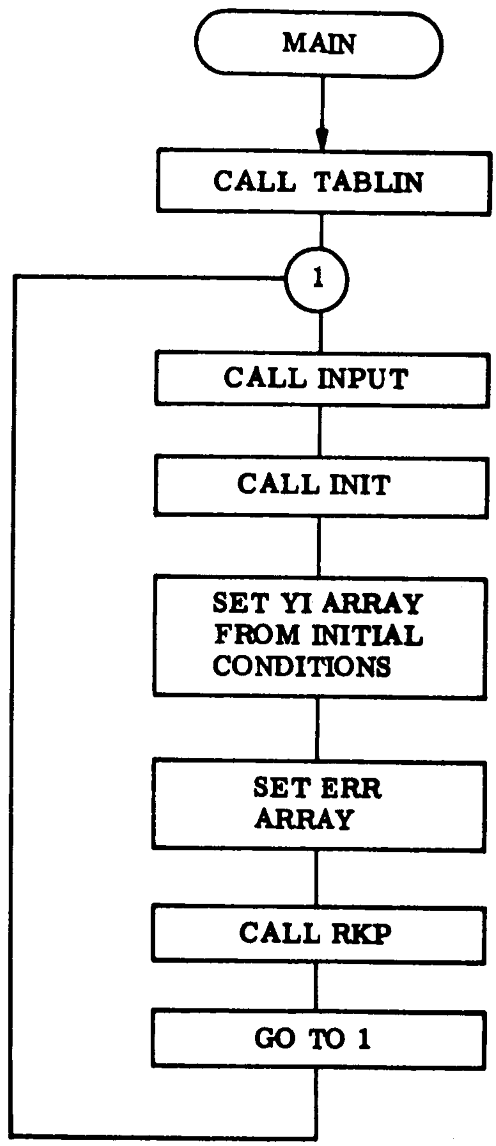
The functional block diagram describes in moderate detail the type of calculations which are being made at various locations in the program, while the program listing presents the complete detail of the calculations as they are actually performed. A complete definition of variables, cross references to Section II and program usage instructions are also provided.

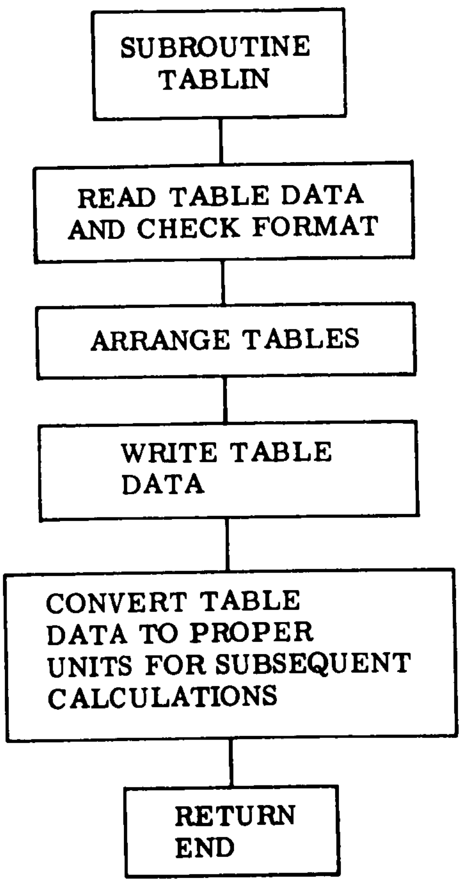
### 3.4.1 FUNCTIONAL BLOCK DIAGRAMS

Block diagrams describing the functions of the computer program are contained in the following pages. The subroutines are presented in the following order:

1. MAIN
2. TABLIN
3. INPUT
4. INIT
5. FEP
6. CONFIG
7. STAB
8. FORCE
9. ENTRP
10. INTEQM
11. RKP
12. PRT

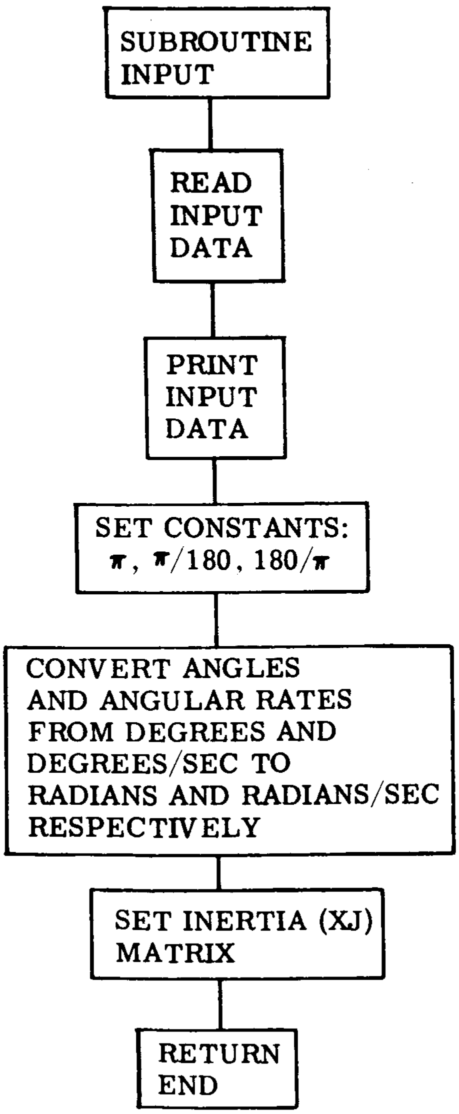
This order will also be maintained in subsequent sections.


MAIN
CALL TABLIN
1
CALL INPUT
CALL INIT
SET YI ARRAY FROM INITIAL CONDITIONS
SET ERR ARRAY
CALL RKP
GO TO 1




READS
TABLES-

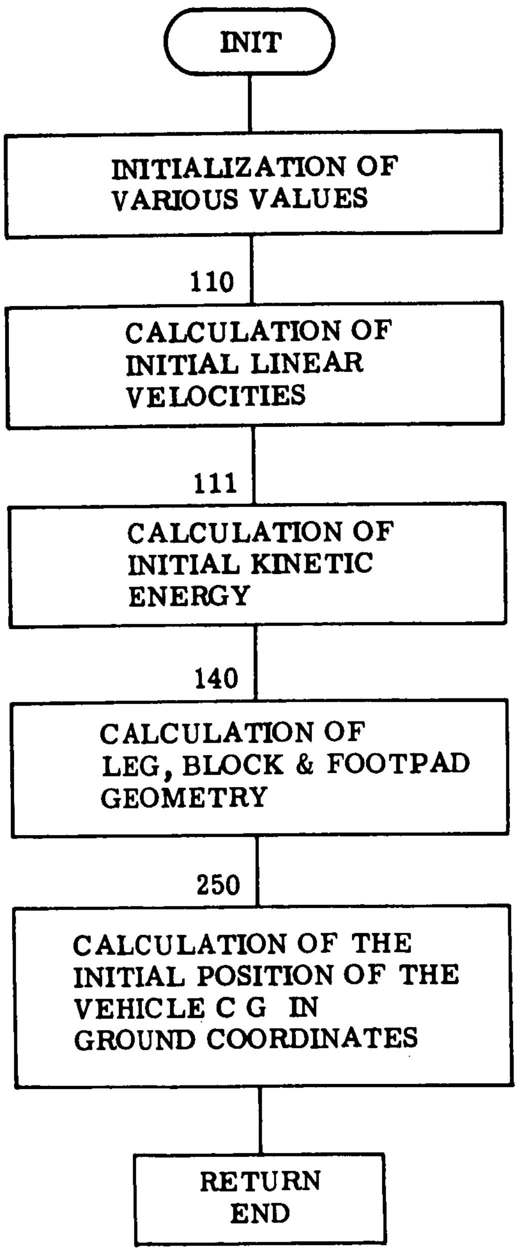
1. Spring Characteristics vs. Shock Absorber Travel.
2. Damping Characteristics vs. Shock Absorber Travel.
3. Crush Characteristics vs. Footpad Compression.
4. Crush Characteristics vs. Crushable Block Compression




Reads and
Modifies
Input Data

Consists of Reading:

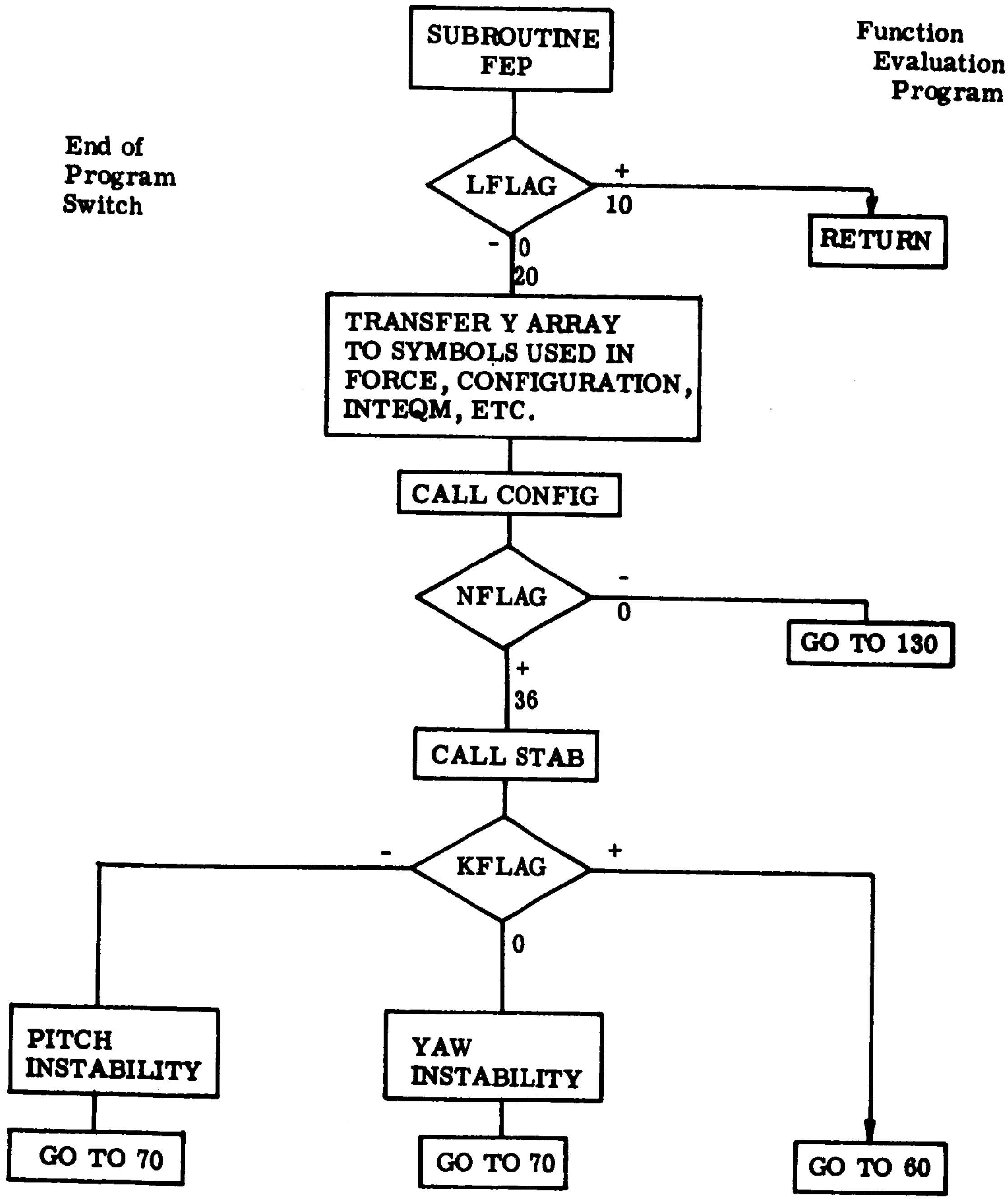
| | |
|---|---|
| XM | XDOTOC |
| XIX | YDOTOC |
| YIY | ZDOTOC |
| ZIZ | XKD (1), XKD (2), XKD (3) |
| XIY | XKS |
| XIZ | FP (1), FP (2), FP (3) |
| YIZ | XNU |
| XMP | XLDMIN |
| XLDO | CB (1), CB (2), CB (3) |
| DELCG | CF (1), CF (2), CF (3) |
| XLP | THETD (1) |
| XLL | THETD (2) |
| XLA | THETD (3) |
| G | THETSD |
| ALPHAD (1) | PSIOD |
| ALPHAD (2) | PHIOD |
| ALPHK (1) | XIOD |
| ALPHK (2) | XLAMDD |
| BETAD | OMEGA (1) |
| XLC | OMEGA (2) |
| HBO | OMEGA (3) |
| XIALPH | RR (1), RR (2), RR (3) |
| XIGAM | RUNNO |
| HFO | SERNO |
| DB | NPHASE |
| DF | VH |
| RC (1), RC (2), | VV |
| RC (3) | IWILEY |
| SRL (1), SRL (2) | ICODE |
| SRL (3) | XDTP |
| XMUF | TFO |
| XMUB | HM |
| TAU (1), TAU (2), | |
| TAU (3) | ERR (4) through |
| | ERR (24) |

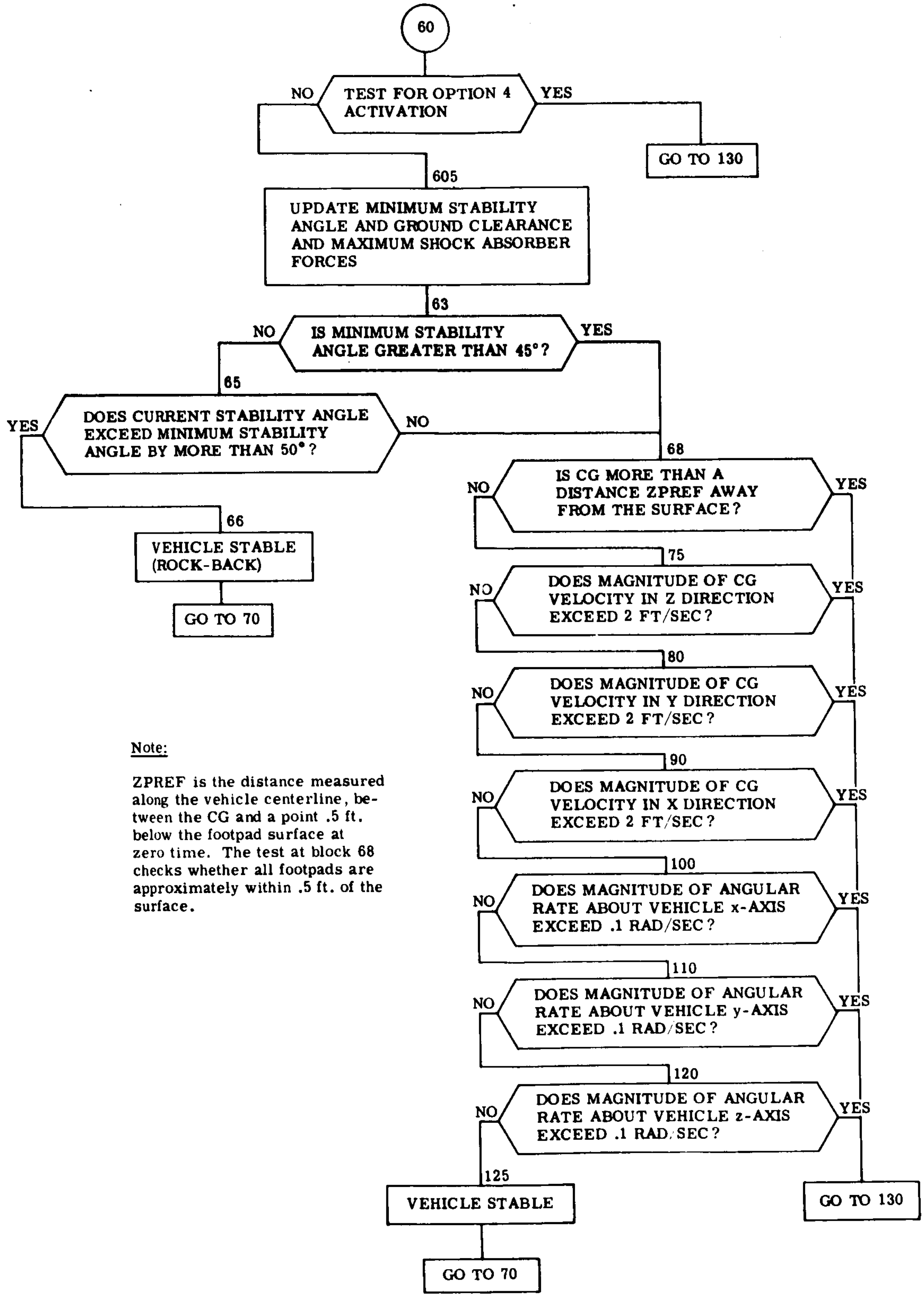



The following program variables or arrays are assigned a value in subroutine INIT:

SINTHS, COSTHS
ISET
A - XA, YA, ZA
A - DELEL
A - IBEEN
DELMAX
A - SLOPE1, SLOPE2
XC, YX, ZC
PSI, PHI, XI
A - B
XDOTC, YDOTC, ZDOTC
PFLAG
TZERO
A - XCDP, YCDP
A - GAMMA
A - ALPHA
A - SINTH, COSTH
A - PSB, PSBQ
A - XB, YB, ZBO
A - XHPU, YHPU, ZHPU
A - GAMDOT
A - ALPDOT
A - XFP, YFP, ZFP
A - HPL
A - DRS
A - ZCPMAX
A - PSP, PSPQ
ZPREF
VZERO

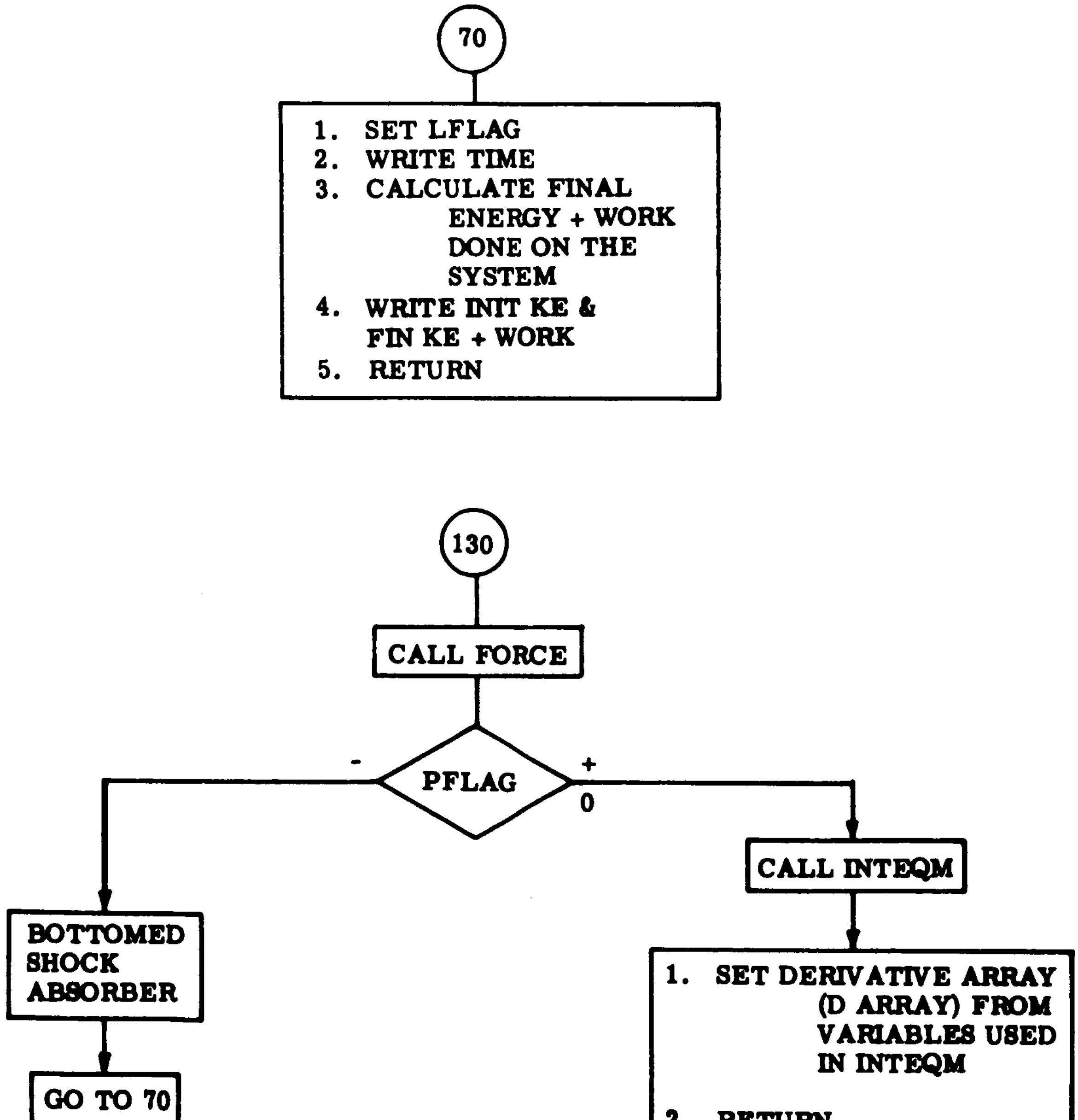
"A" denotes array.

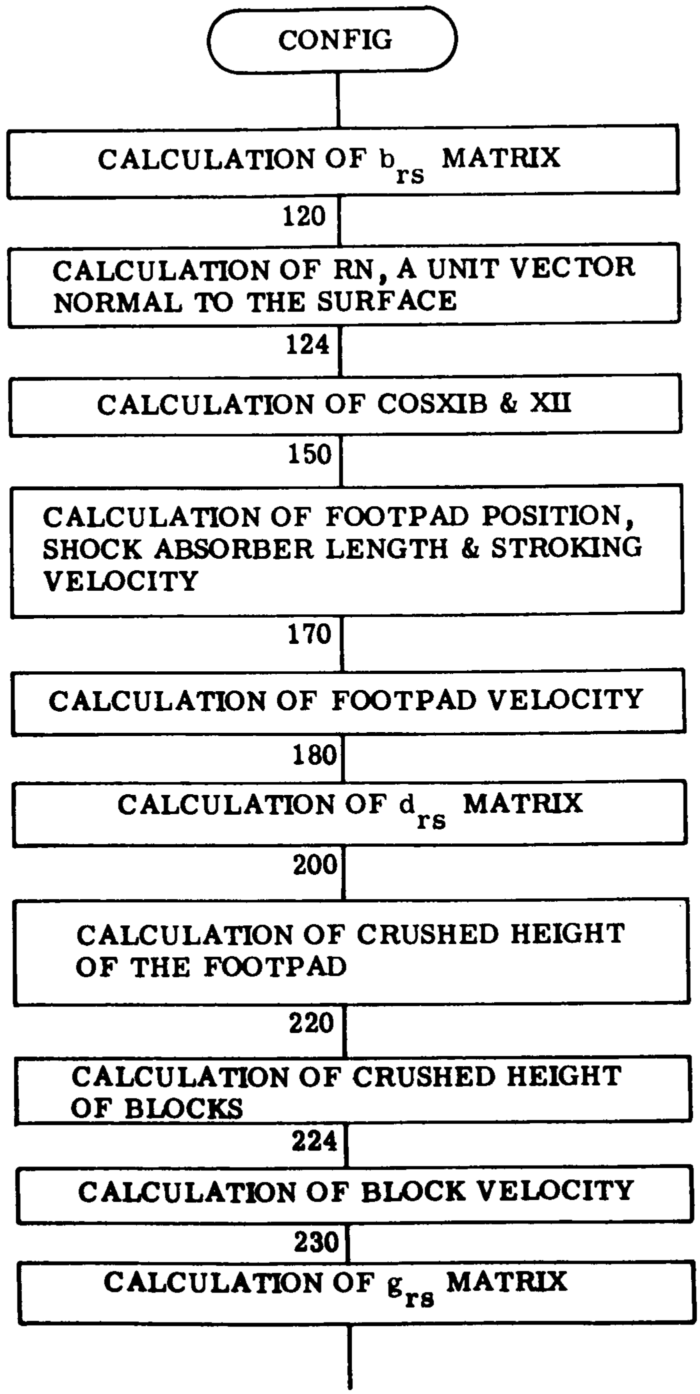

SUBROUTINE FEP
Function Evaluation Program
End of Program Switch
LFLAG
+
10
RETURN
-
0
20
TRANSFER Y ARRAY TO SYMBOLS USED IN FORCE, CONFIGURATION, INTEQM, ETC.
CALL CONFIG
NFLAG
-
0
GO TO 130
+
36
CALL STAB
KFLAG
-
+
0
PITCH INSTABILITY
GO TO 70
YAW INSTABILITY
GO TO 70
GO TO 60

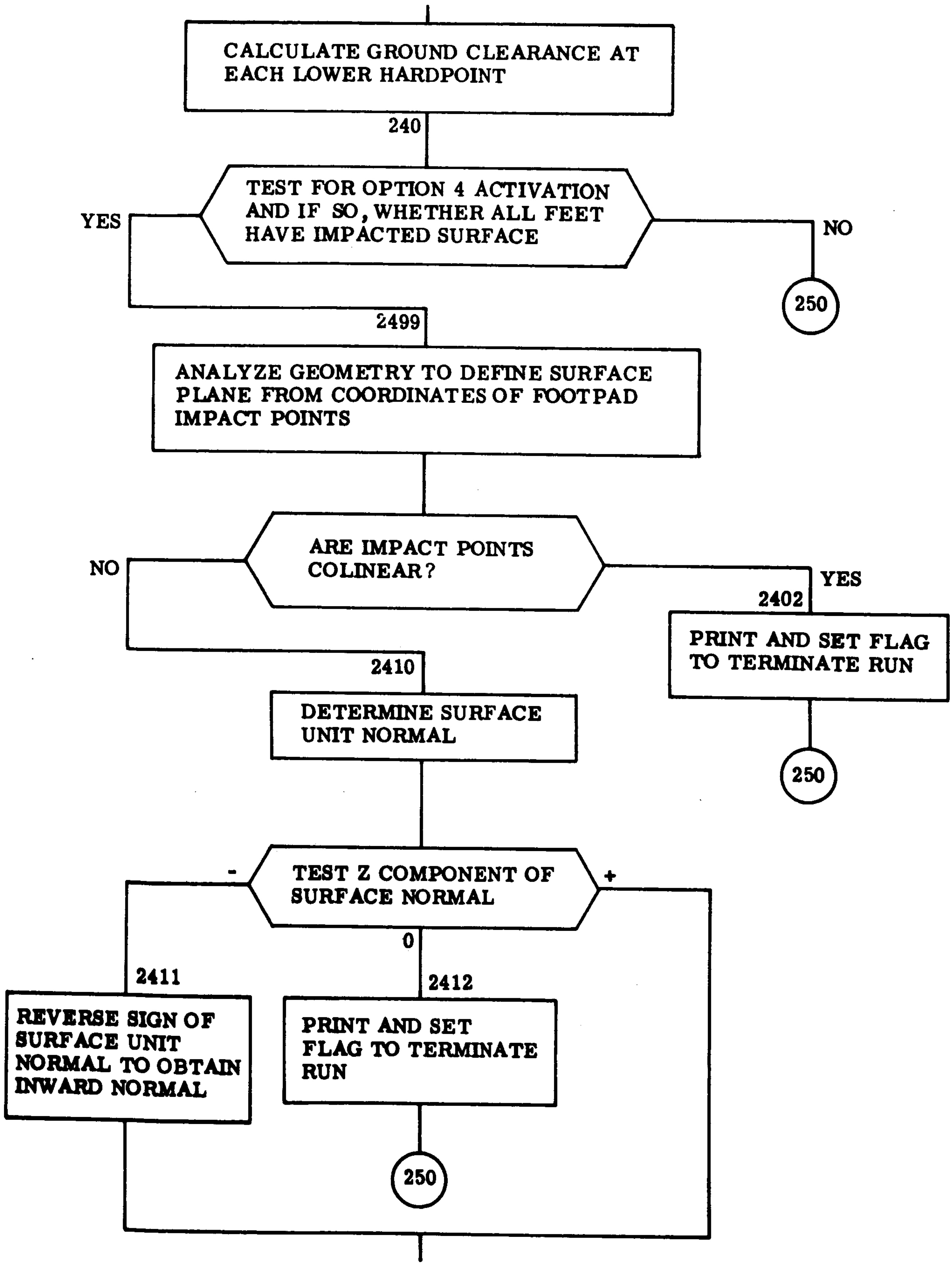


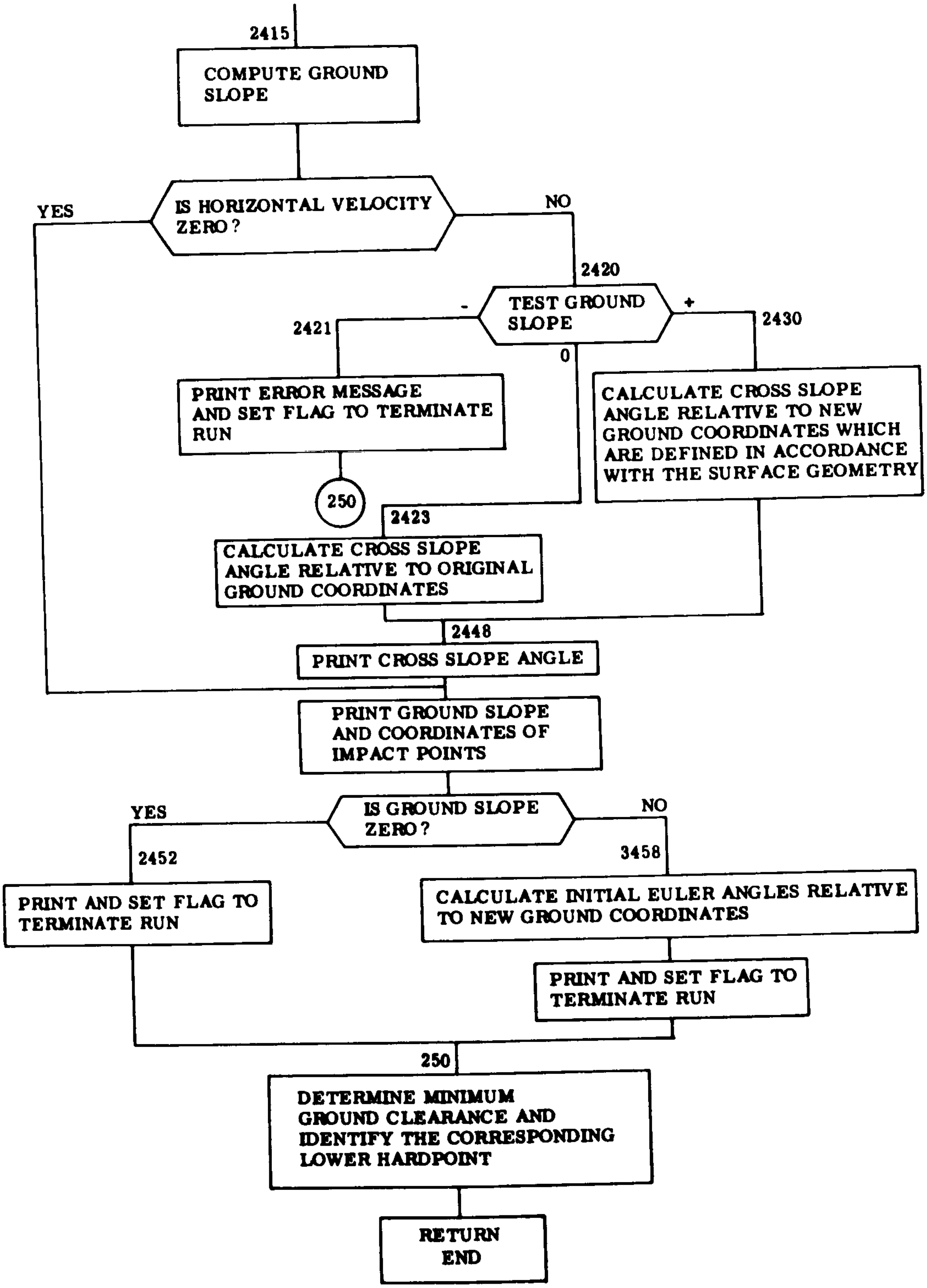

Note:

ZPREF is the distance measured along the vehicle centerline, between the CG and a point .5 ft. below the footpad surface at zero time. The test at block 68 checks whether all footpads are approximately within .5 ft. of the surface.

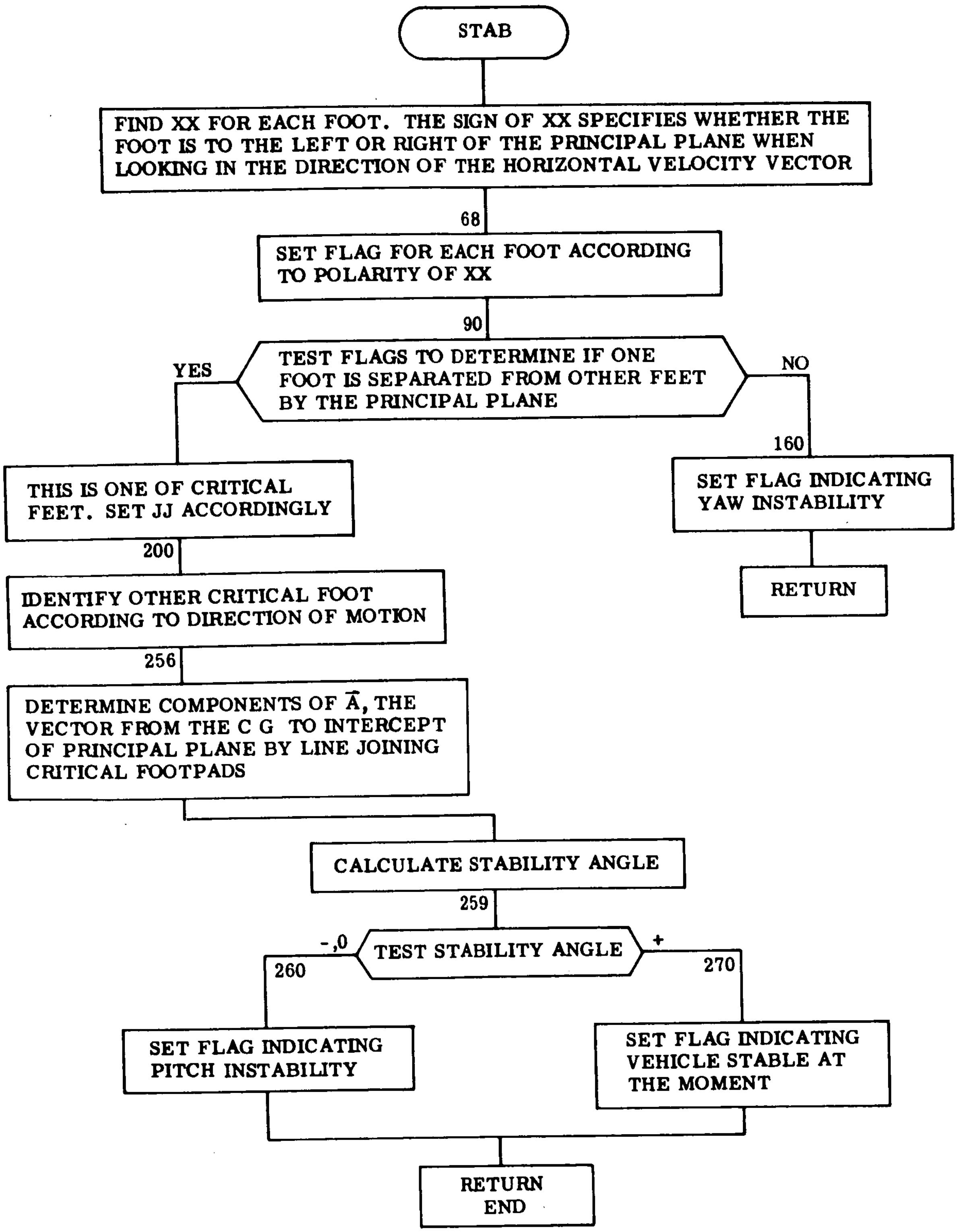

70
1. SET LFLAG
2. WRITE TIME
3. CALCULATE FINAL ENERGY + WORK DONE ON THE SYSTEM
4. WRITE INIT KE & FIN KE + WORK
5. RETURN
130
CALL FORCE
PFLAG
-
+
0
CALL INTEQM
BOTTOMED SHOCK ABSORBER
GO TO 70
1. SET DERIVATIVE ARRAY (D ARRAY) FROM VARIABLES USED IN INTEQM
2. RETURN

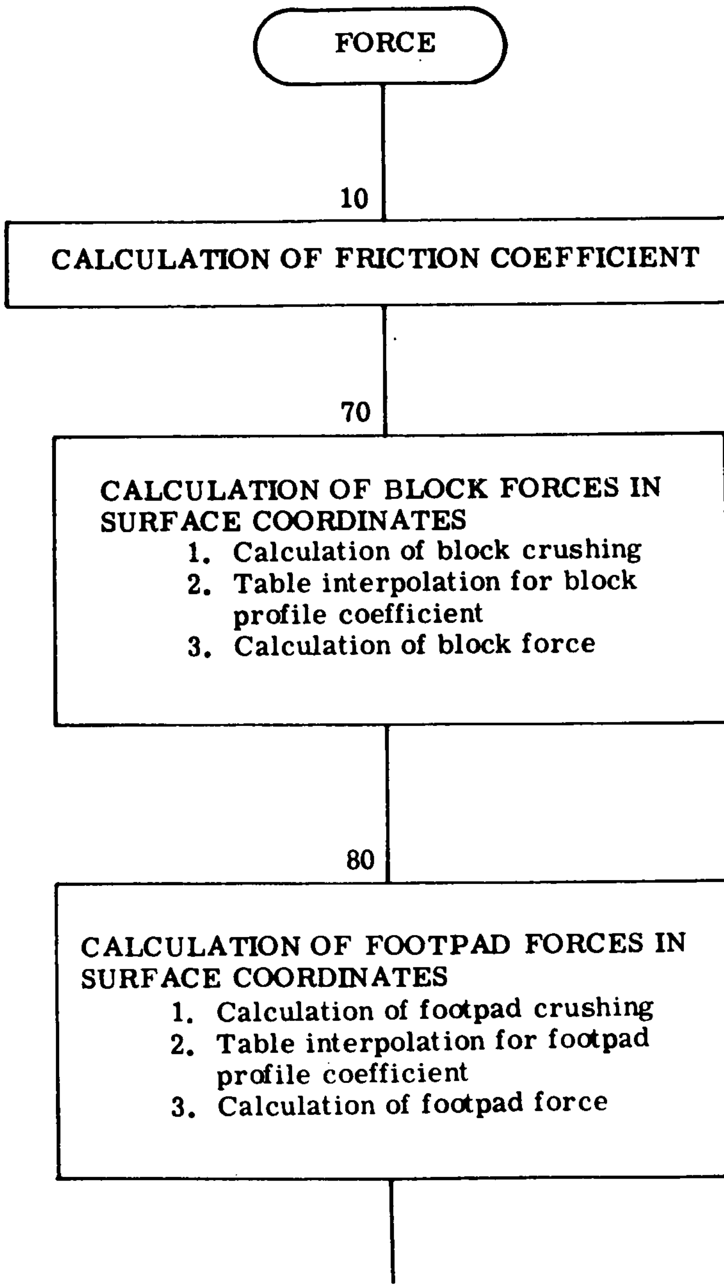

CONFIG
CALCULATION OF $b_{rs}$ MATRIX
120
CALCULATION OF RN, A UNIT VECTOR NORMAL TO THE SURFACE
124
CALCULATION OF COSXIB & XII
150
CALCULATION OF FOOTPAD POSITION, SHOCK ABSORBER LENGTH & STROKING VELOCITY
170
CALCULATION OF FOOTPAD VELOCITY
180
CALCULATION OF $d_{rs}$ MATRIX
200
CALCULATION OF CRUSHED HEIGHT OF THE FOOTPAD
220
CALCULATION OF CRUSHED HEIGHT OF BLOCKS
224
CALCULATION OF BLOCK VELOCITY
230
CALCULATION OF $g_{rs}$ MATRIX


CALCULATE GROUND CLEARANCE AT EACH LOWER HARDPOINT
240
TEST FOR OPTION 4 ACTIVATION AND IF SO, WHETHER ALL FEET HAVE IMPACTED SURFACE
YES
NO
250
2499
ANALYZE GEOMETRY TO DEFINE SURFACE PLANE FROM COORDINATES OF FOOTPAD IMPACT POINTS
ARE IMPACT POINTS COLINEAR?
NO
YES
2402
PRINT AND SET FLAG TO TERMINATE RUN
250
2410
DETERMINE SURFACE UNIT NORMAL
TEST Z COMPONENT OF SURFACE NORMAL
-
+
0
2411
REVERSE SIGN OF SURFACE UNIT NORMAL TO OBTAIN INWARD NORMAL
2412
PRINT AND SET FLAG TO TERMINATE RUN
250



2415
COMPUTE GROUND SLOPE
IS HORIZONTAL VELOCITY ZERO?
YES
NO
2420
TEST GROUND SLOPE
-
+
0
2421
2430
PRINT ERROR MESSAGE AND SET FLAG TO TERMINATE RUN
CALCULATE CROSS SLOPE ANGLE RELATIVE TO NEW GROUND COORDINATES WHICH ARE DEFINED IN ACCORDANCE WITH THE SURFACE GEOMETRY
250
2423
CALCULATE CROSS SLOPE ANGLE RELATIVE TO ORIGINAL GROUND COORDINATES
2448
PRINT CROSS SLOPE ANGLE
PRINT GROUND SLOPE AND COORDINATES OF IMPACT POINTS
IS GROUND SLOPE ZERO?
YES
NO
2452
3458
PRINT AND SET FLAG TO TERMINATE RUN
CALCULATE INITIAL EULER ANGLES RELATIVE TO NEW GROUND COORDINATES
PRINT AND SET FLAG TO TERMINATE RUN
250
DETERMINE MINIMUM GROUND CLEARANCE AND IDENTIFY THE CORRESPONDING LOWER HARDPOINT
RETURN
END


STAB
FIND XX FOR EACH FOOT. THE SIGN OF XX SPECIFIES WHETHER THE FOOT IS TO THE LEFT OR RIGHT OF THE PRINCIPAL PLANE WHEN LOOKING IN THE DIRECTION OF THE HORIZONTAL VELOCITY VECTOR
68
SET FLAG FOR EACH FOOT ACCORDING TO POLARITY OF XX
90
TEST FLAGS TO DETERMINE IF ONE FOOT IS SEPARATED FROM OTHER FEET BY THE PRINCIPAL PLANE
YES
NO
160
SET FLAG INDICATING YAW INSTABILITY
RETURN
THIS IS ONE OF CRITICAL FEET. SET JJ ACCORDINGLY
200
IDENTIFY OTHER CRITICAL FOOT ACCORDING TO DIRECTION OF MOTION
256
DETERMINE COMPONENTS OF $\vec{A}$, THE VECTOR FROM THE C G TO INTERCEPT OF PRINCIPAL PLANE BY LINE JOINING CRITICAL FOOTPADS
CALCULATE STABILITY ANGLE
259
TEST STABILITY ANGLE
-,0
+
260
270
SET FLAG INDICATING PITCH INSTABILITY
SET FLAG INDICATING VEHICLE STABLE AT THE MOMENT
RETURN
END


FORCE
10
CALCULATION OF FRICTION COEFFICIENT
70
CALCULATION OF BLOCK FORCES IN SURFACE COORDINATES
1. Calculation of block crushing
2. Table interpolation for block profile coefficient
3. Calculation of block force
80
CALCULATION OF FOOTPAD FORCES IN SURFACE COORDINATES
1. Calculation of footpad crushing
2. Table interpolation for footpad profile coefficient
3. Calculation of footpad force

120

TRANSFER OF CRUSH AND FRICTION FORCES TO VEHICLE COORDINATES FROM SURFACE COORDINATES

180

CALCULATION OF SHOCK ABSORBER FORCES

1. Calculate Stroke
2. Table interpolation for damping profile and spring rate profile coefficients
3. Calculate spring force and preload
4. Test for stroke or restroke and calculate friction and damping forces
5. Sum all shock absorber forces
6. Resolve shock absorber force into vehicle coordinates

192

**CALCULATION OF TORQUES ABOUT LOWER HARDPOINT**

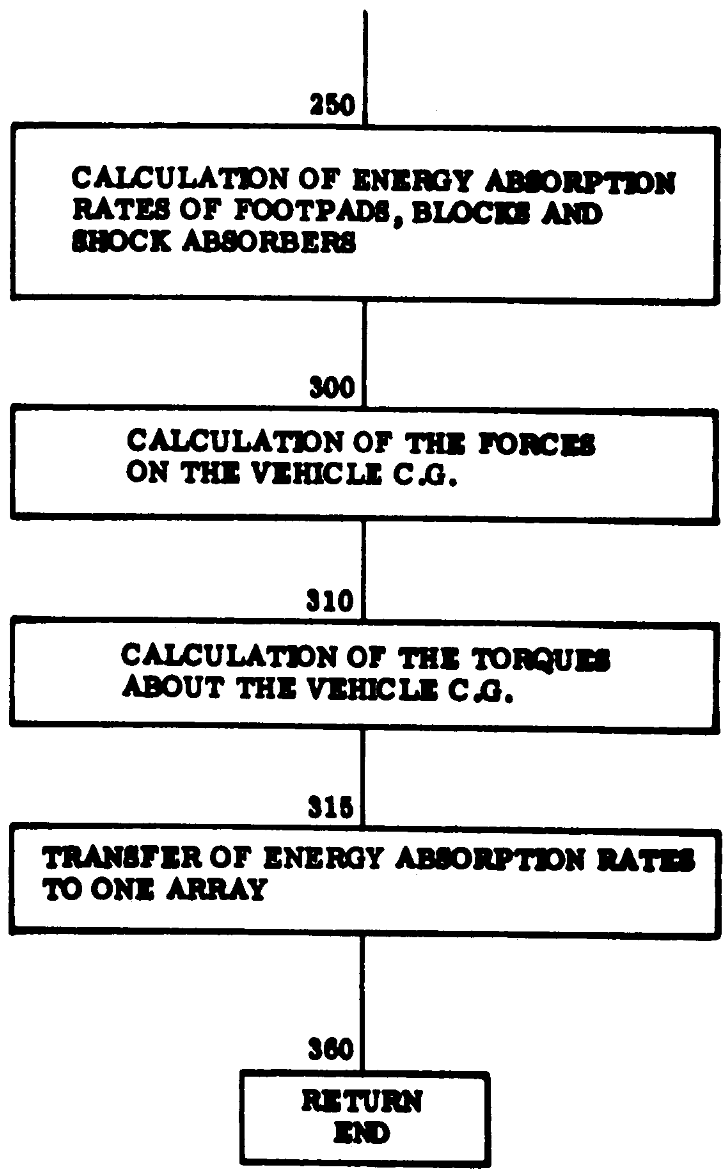
1. Sum footpad and shock absorber forces (vehicle coordinates)
2. Transfer this sum to strut coordinates
3. Calculation of forces and moments on lower hardpoint in vehicle coordinates
4. Calculation of torque about lower hardpoint in a direction which will cause rotation of the lower leg.

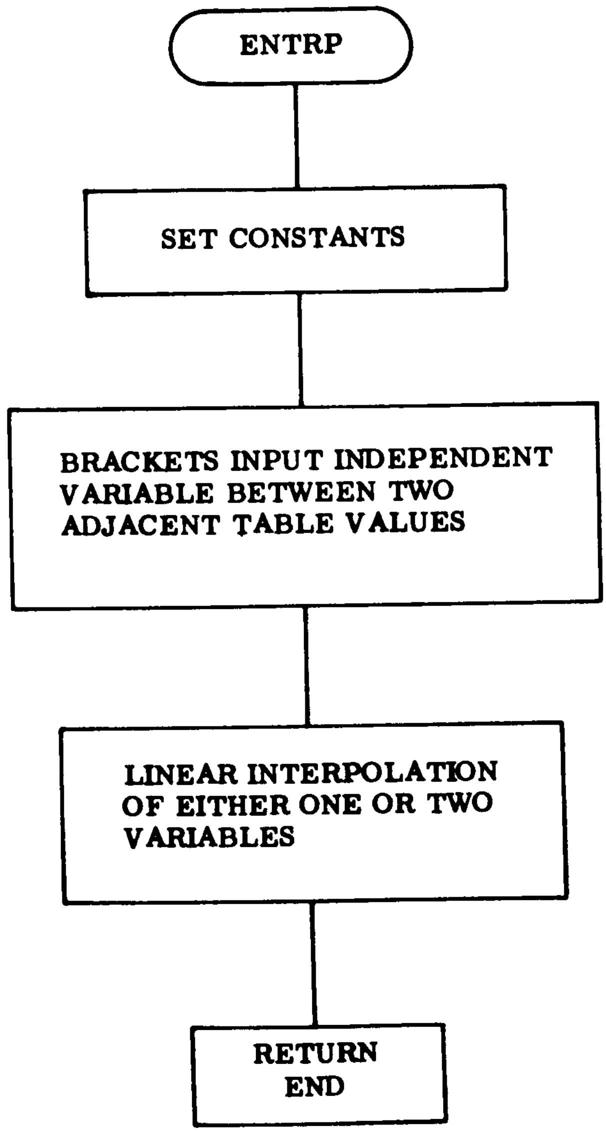
205

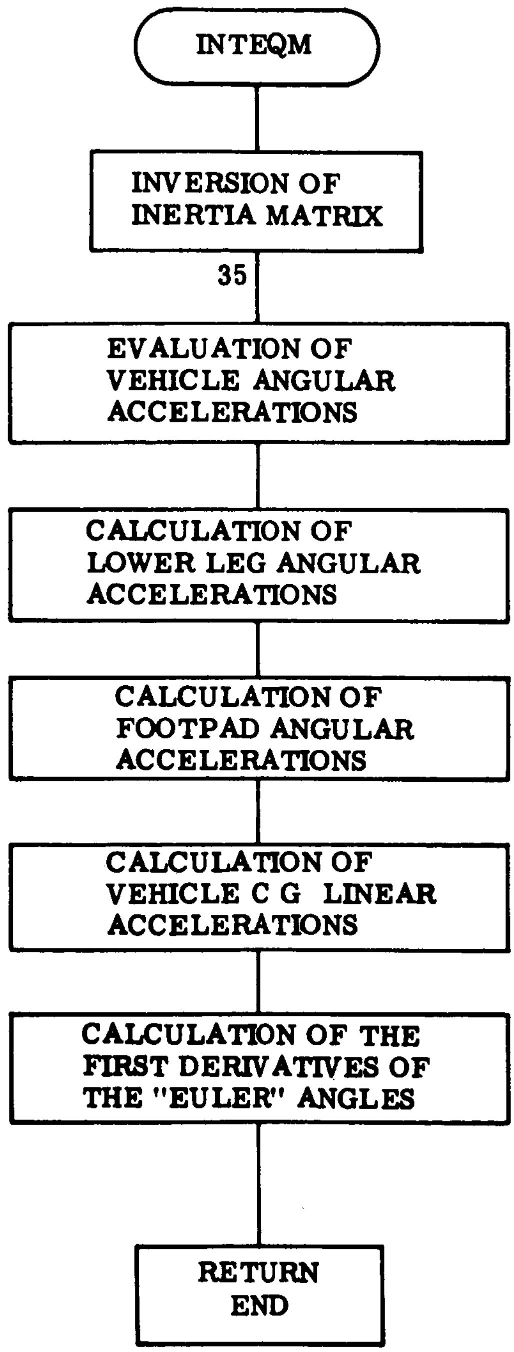
**CALCULATION OF TORQUE ON FOOTPAD**

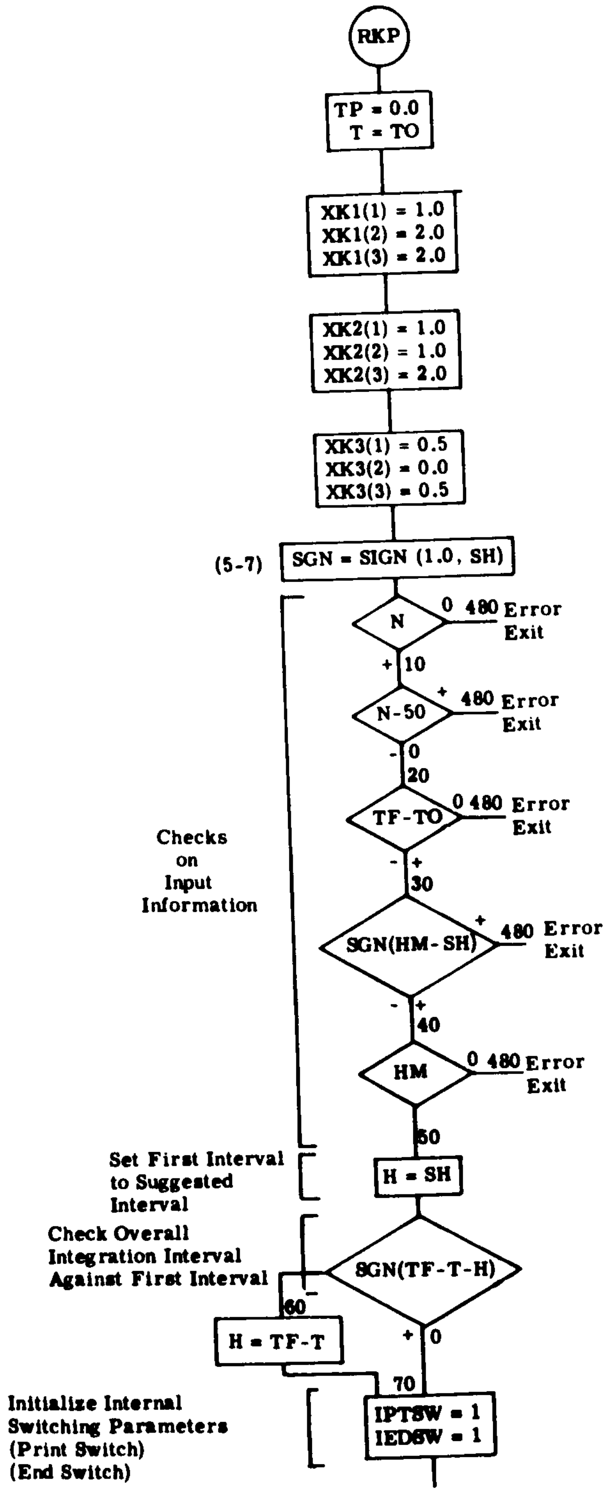
224

**FOOTPAD STOP CALCULATIONS**

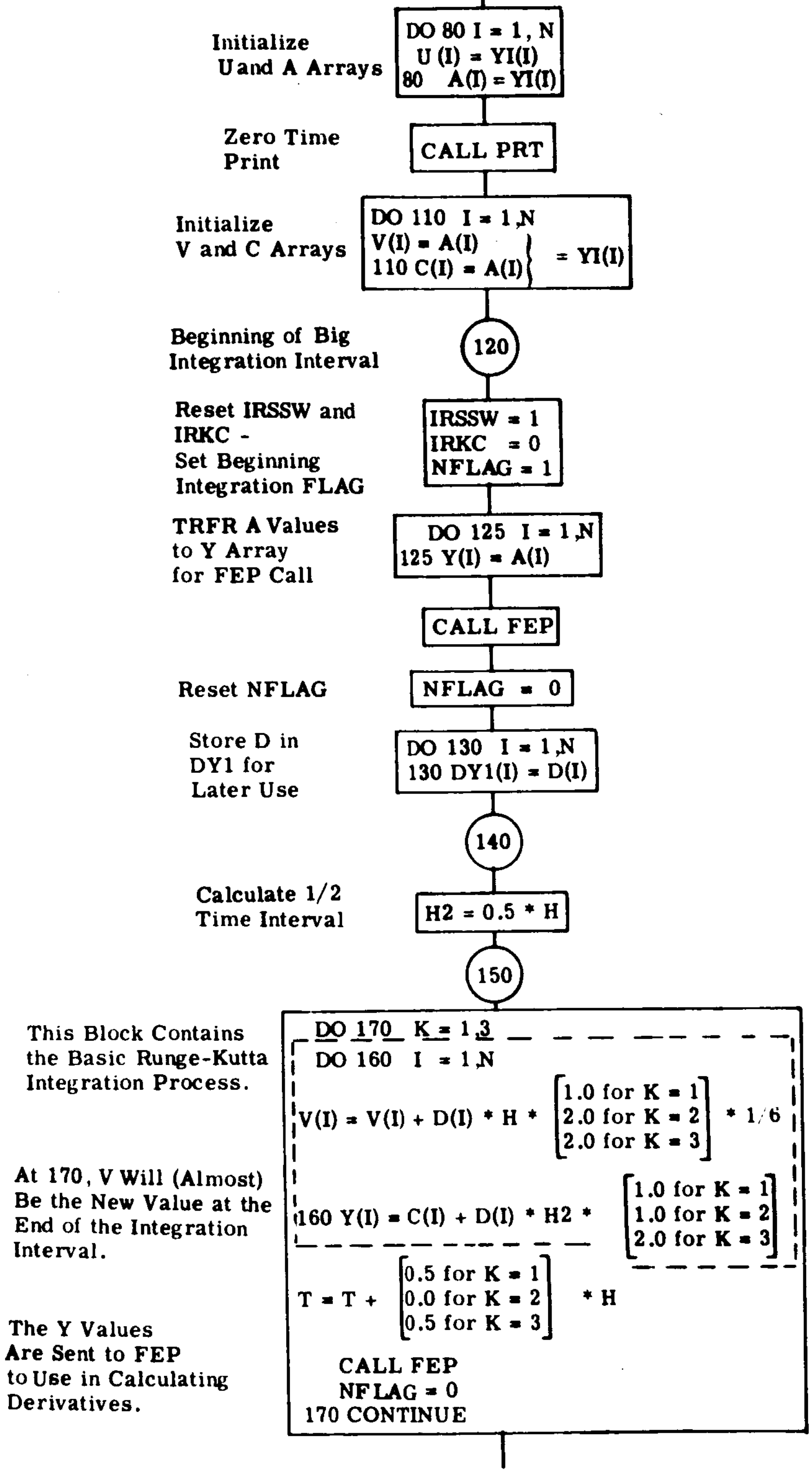

250
CALCULATION OF ENERGY ABSORPTION RATES OF FOOTPADS, BLOCKS AND SHOCK ABSORBERS
300
CALCULATION OF THE FORCES ON THE VEHICLE C.G.
310
CALCULATION OF THE TORQUES ABOUT THE VEHICLE C.G.
315
TRANSFER OF ENERGY ABSORPTION RATES TO ONE ARRAY
360
RETURN
END


ENTRP
SET CONSTANTS
BRACKETS INPUT INDEPENDENT VARIABLE BETWEEN TWO ADJACENT TABLE VALUES
LINEAR INTERPOLATION OF EITHER ONE OR TWO VARIABLES
RETURN
END


INTEQM
INVERSION OF INERTIA MATRIX
35
EVALUATION OF VEHICLE ANGULAR ACCELERATIONS
CALCULATION OF LOWER LEG ANGULAR ACCELERATIONS
CALCULATION OF FOOTPAD ANGULAR ACCELERATIONS
CALCULATION OF VEHICLE C G LINEAR ACCELERATIONS
CALCULATION OF THE FIRST DERIVATIVES OF THE "EULER" ANGLES
RETURN
END


RKP
TP = 0.0
T = TO
XK1(1) = 1.0
XK1(2) = 2.0
XK1(3) = 2.0
XK2(1) = 1.0
XK2(2) = 1.0
XK2(3) = 2.0
XK3(1) = 0.5
XK3(2) = 0.0
XK3(3) = 0.5
(5-7)
SGN = SIGN (1.0, SH)
N
0 480 Error Exit
+ 10
N-50
+ 480 Error Exit
- 0
20
TF-TO
0 480 Error Exit
- + 30
Checks on Input Information
SGN(HM-SH)
+ 480 Error Exit
- + 40
HM
0 480 Error Exit
50
Set First Interval to Suggested Interval
H = SH
Check Overall Integration Interval Against First Interval
SGN(TF-T-H)
60 -
H = TF-T
+ 0
70
Initialize Internal Switching Parameters (Print Switch) (End Switch)
IPTSW = 1
IEDSW = 1

SUBROUTINE RKP



Complete Runge-
Kutta Process

```
      DO 180  I = 1,N
  180 V(I) = V(I) + D(I) * H * 1/6
```

IRKC = 0
IRKC = 1 IRKC = 2

```
      IRKC = IRKC + 1
      GO TO (190, 210, 240), IRKC
```

Picks up Derivatives
Stored Earlier
Saves V Values Just
Calculated in U Array
Moves A Array to V
Array
Resets Time

Resets H

Signals Start
of Second Pass

```
      DO  200  I = 1,N
      D(I)  =  DY 1(I)
      U(I)  =  V(I)
  200 V(I) = A(I)
      T = T - H
      H = H2
      NFLAG = -1
      GO TO 140
```

210

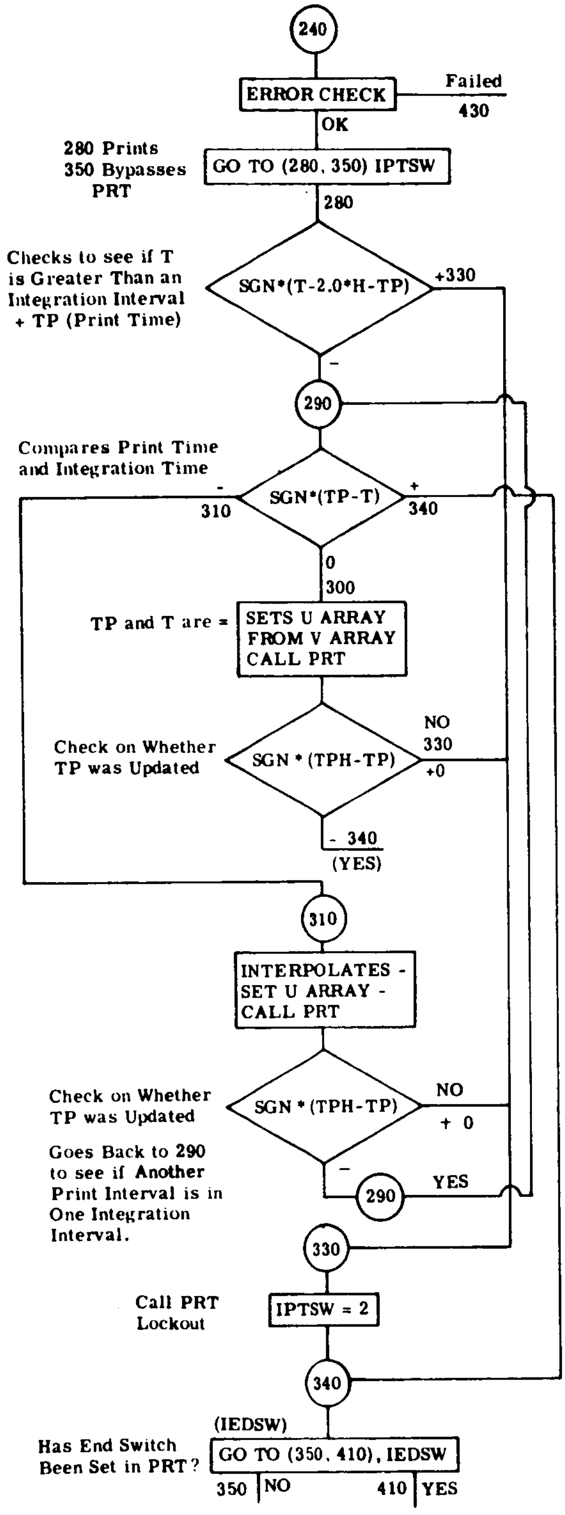
Moves Previous Calculated
Values (V) to Y Array.
Same as above to C Array.

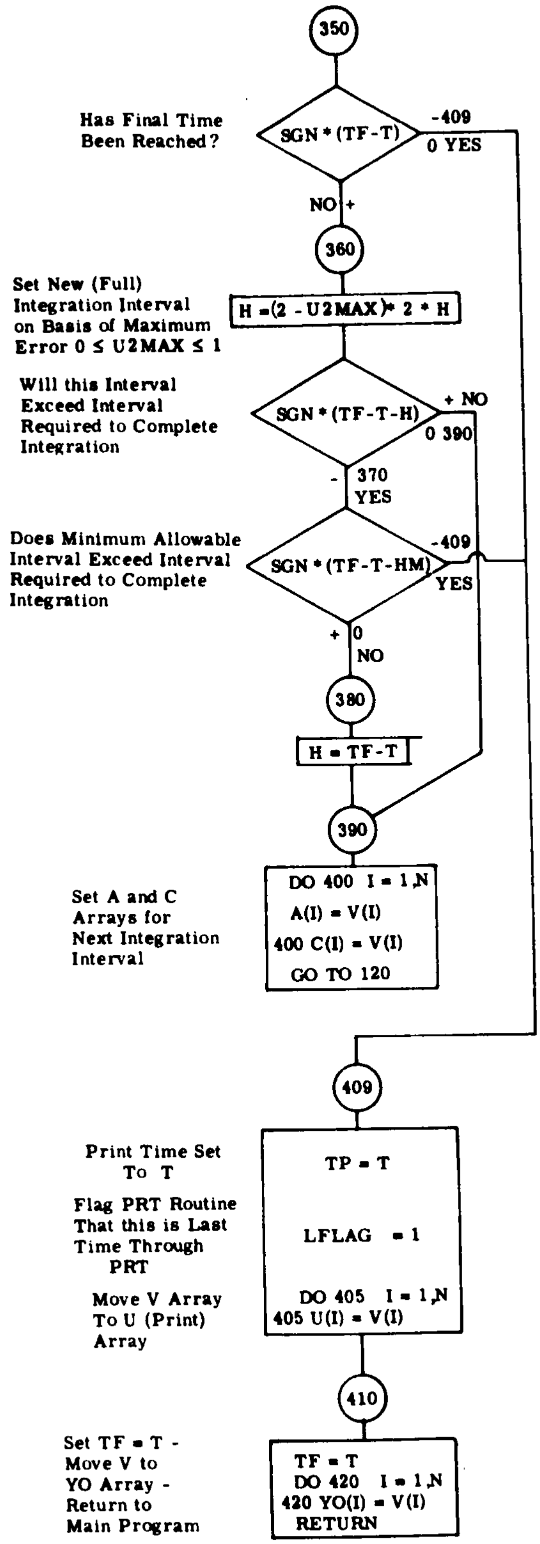
```
      DO  220   I = 1,N
      Y(I)  =  V(I)
  220
      C(I)  =  V(I)
```

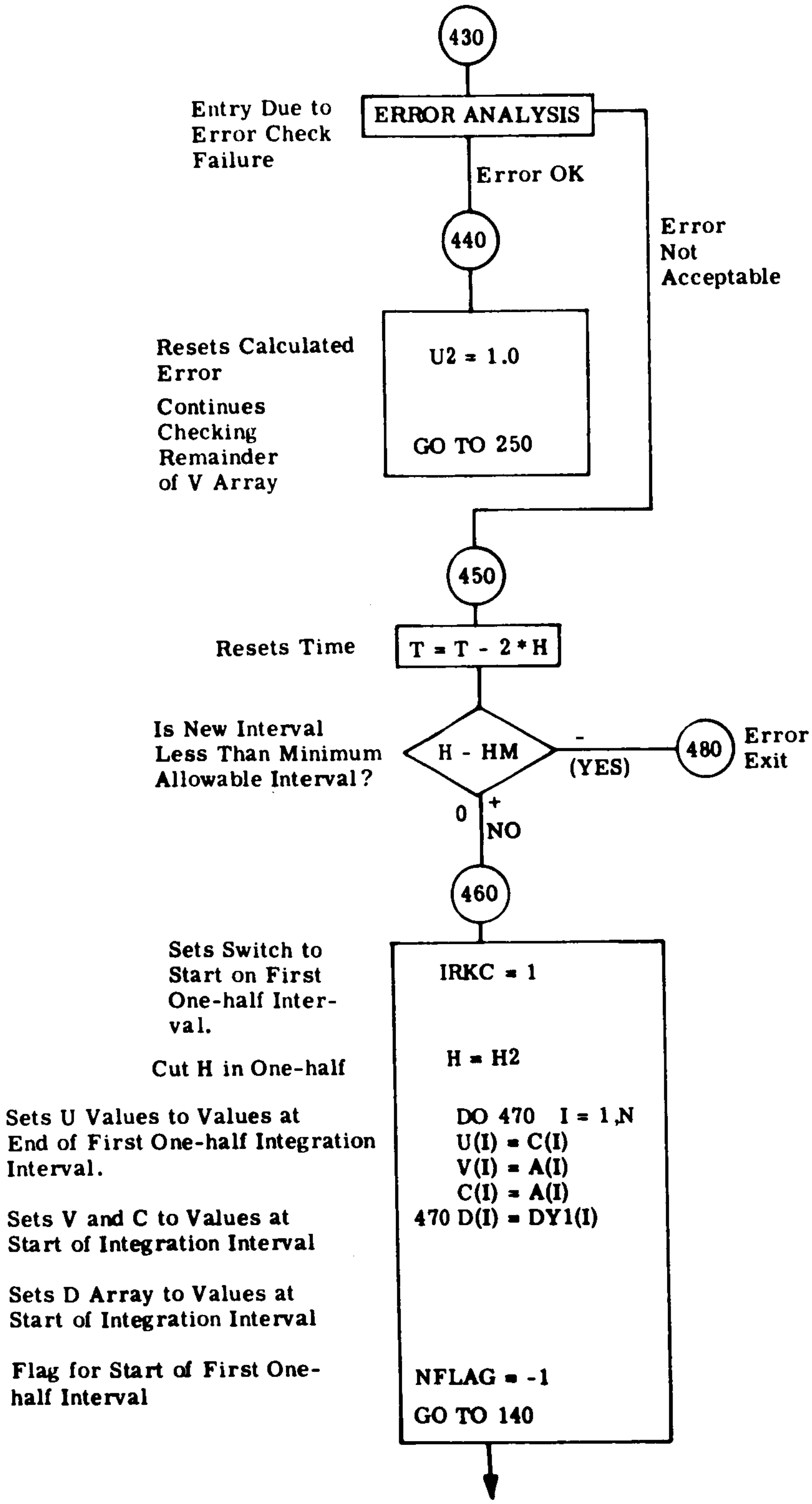
Start of Second
One-half Interval

Save Derivatives
For Interpolation.

```
      CALL FEP
      DO 230  I = 1,N
  230 DY2(I) = D(I)
      GO TO 150
```


240
ERROR CHECK
Failed
430
OK
280 Prints
350 Bypasses
PRT
GO TO (280, 350) IPTSW
280
Checks to see if T is Greater Than an Integration Interval + TP (Print Time)
SGN*(T-2.0*H-TP)
+330
-
290
Compares Print Time and Integration Time
SGN*(TP-T)
-
310
+
340
0
300
TP and T are =
SETS U ARRAY FROM V ARRAY CALL PRT
Check on Whether TP was Updated
SGN * (TPH-TP)
NO
330
+0
- 340
(YES)
310
INTERPOLATES - SET U ARRAY - CALL PRT
Check on Whether TP was Updated
SGN * (TPH-TP)
NO
+ 0
Goes Back to 290 to see if Another Print Interval is in One Integration Interval.
-
290
YES
330
Call PRT Lockout
IPTSW = 2
340
(IEDSW)
Has End Switch Been Set in PRT?
GO TO (350, 410), IEDSW
350
NO
410
YES


350
Has Final Time Been Reached?
SGN * (TF-T)
-409
0 YES
NO +
360
Set New (Full) Integration Interval on Basis of Maximum Error 0 ≤ U2MAX ≤ 1
H =(2 - U2MAX)* 2 * H
Will this Interval Exceed Interval Required to Complete Integration
SGN * (TF-T-H)
+ NO
0 390
- 370 YES
Does Minimum Allowable Interval Exceed Interval Required to Complete Integration
SGN * (TF-T-HM)
-409
YES
+ 0 NO
380
H = TF-T
390
Set A and C Arrays for Next Integration Interval
DO 400 I = 1,N
A(I) = V(I)
400 C(I) = V(I)
GO TO 120
409
Print Time Set To T
TP = T
Flag PRT Routine That this is Last Time Through PRT
LFLAG = 1
Move V Array To U (Print) Array
DO 405 I = 1,N
405 U(I) = V(I)
410
Set TF = T - Move V to YO Array - Return to Main Program
TF = T
DO 420 I = 1,N
420 YO(I) = V(I)
RETURN


430
Entry Due to Error Check Failure
ERROR ANALYSIS
Error OK
Error Not Acceptable
440
Resets Calculated Error
U2 = 1.0
Continues Checking Remainder of V Array
GO TO 250
450
Resets Time
T = T - 2*H
Is New Interval Less Than Minimum Allowable Interval?
H - HM
-
(YES)
480
Error Exit
0
+
NO
460
Sets Switch to Start on First One-half Interval.
IRKC = 1
Cut H in One-half
H = H2
Sets U Values to Values at End of First One-half Integration Interval.
DO 470 I = 1,N
U(I) = C(I)
V(I) = A(I)
C(I) = A(I)
470 D(I) = DY1(I)
Sets V and C to Values at Start of Integration Interval
Sets D Array to Values at Start of Integration Interval
Flag for Start of First One-half Interval
NFLAG = -1
GO TO 140

480 ERROR EXIT

```
     PRINT 490
490 FORMAT //28 H RUNGE-KUTTA SUBROUTINE ERROR//
     STOP
     END
```

TERMINATION ERRORS

1. Number of Equations $\leq 0$
2. Number of Equations $> 50$
3. Final Time = Initial Time
4. (Suggested Interval) < (Minimum Interval)
5. Minimum Interval = 0

FOR EQUATIONS OF THE FORM:

$$\left.\frac{dy}{dx}\right|_i = f(x_i, y_i)$$

RUNGE-KUTTA EQUATIONS

$$k_1 = h \cdot f(x_o, y_o)$$

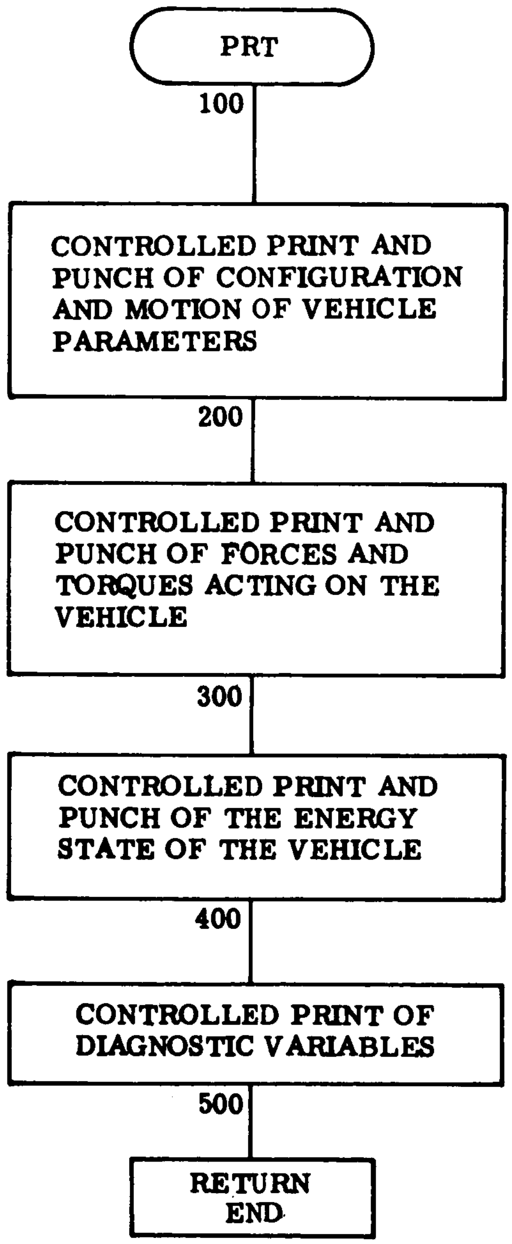
$$k_2 = h \cdot f(x_o + \frac{1}{2} h, y_o + \frac{1}{2} k_1)$$

$$k_3 = h \cdot f(x_o + \frac{1}{2} h, y_o + \frac{1}{2} k_2)$$

$$k_4 = h \cdot f(x_o + h, y_o + k_3)$$

$$y_1 = y_o + \frac{1}{6} k_1 + \frac{1}{3} k_2 + \frac{1}{3} k_3 + \frac{1}{6} k_4$$

$$x_1 = x_o + h$$


PRT
100
CONTROLLED PRINT AND PUNCH OF CONFIGURATION AND MOTION OF VEHICLE PARAMETERS
200
CONTROLLED PRINT AND PUNCH OF FORCES AND TORQUES ACTING ON THE VEHICLE
300
CONTROLLED PRINT AND PUNCH OF THE ENERGY STATE OF THE VEHICLE
400
CONTROLLED PRINT OF DIAGNOSTIC VARIABLES
500
RETURN END

### 3.4.2 DEFINITION OF PROGRAM VARIABLES

On the following pages, the program variables and arrays are defined. The definitions, although somewhat terse, will assist in bridging the gap between the equations in Section II, the block diagrams and the program listing. Only variables actually used and/or defined in the program have been listed. The variables are segregated by the subroutine in which they are used and/or defined and are listed alphabetically. Since some variables appear in more than one subroutine, definitions may be duplicated in order to avoid the need for cross-reference.

The format of this section is as follows: first, the parameter or array name; second, enclosed in parenthesis, the common list (if any) in which the variable or array exists; third, the definitions; finally, enclosed in parenthesis, the dimensions (if any) appropriate to the subject variable or array.

MAIN

| | |
|---|---|
| DEGRAD (N7) | Conversion factor (rad/deg). |
| ERR (N1) | Allowable integration truncation error per second (ft/sec, ft/sec/sec, degrees/sec , degrees/sec/sec ). |
| GAMDOT (N7) | Lower strut angular rate (rad/sec). |
| GAMMA (N7) | Lower strut angular position (rad). |
| HM (N1) | Minimum integration interval size (sec). |
| I | DO variable. |
| IOPT (N24) | Program switch controlling various output options. |
| J | DO index. |
| N (N1) | Denotes the number of differential equations to be integrated. |
| OMEG (N7) | Angular rates about vehicles axes (rad/sec). |
| PHIO (N7) | Initial yaw angle of the vehicle (rad). |
| PSIO (N7) | Initial pitch angle of the vehicle (rad). |
| SH (N1) | Suggested size of first integration interval (sec). |
| TAMIN | Minimum of the three footpad contact times (sec). |
| TAU (N12) | Time at which a given footpad contacts the surface (sec). |
| TF (N1) | The time to which the integration will proceed unless other stops are encountered (sec). |

| | |
|---|---|
| TO (N1) | Starting integration time (sec). |
| XC (N7) | Vehicle position - X-direction - Ground coordinates (feet). |
| XDOTC (N7) | Vehicle velocity - X-direction - Ground coordinates (feet/sec). |
| XIO (N7) | Initial roll angle of the vehicle (rad). |
| YC (N7) | Vehicle position - Y- direction - Ground coordinates (feet). |
| YDOTC (N7) | Vehicle velocity - Y-direction - Ground coordinates (feet/sec). |
| YI (N1) | List of initial values for the integration routine (various dimensions). |
| ZC (N7) | Vehicle position - Z-direction - Ground coordinates (feet). |
| ZDOTC (N7) | Vehicle velocity - Z-direction - Ground coordinates (feet/sec). |

<u>TABLIN</u>

| | |
|---|---|
| BLOCK1 (N10) | Table of Characteristics for Block No. 1. |
| BLOCK2 (N10) | Table of Characteristics for Block No. 2. |
| BLOCK3 (N10) | Table of Characteristics for Block No. 3. |
| CLOD | Temporary storage for table values. |
| FOOT1 (N10) | Table of Characteristics for Footpad No. 1. |
| FOOT2 (N10) | Table of Characteristics for Footpad No. 2. |
| FOOT3 (N10) | Table of Characteristics for Footpad No. 3. |
| I | Counter - value denotes number of separate tables which have been read. |
| IFORK | An array of the code numbers associated with each table. |
| J | DO index. |
| J11 | Number of independent variables in the Footpad No. 1 table. |
| J12 | Number of independent variables in the Footpad No. 2 table. |
| J13 | Number of independent variables in the Footpad No. 3 table. |
| J21 | Number of independent variables in the Block No. 1 table. |
| J22 | Number of independent variables in the Block No. 2 table. |

| | |
|---|---|
| J23 | Number of independent variables in the Block No. 3 table. |
| J31 | Number of independent variables in the Shock Absorber No. 1 table. |
| J32 | Number of independent variables in the Shock Absorber No. 2 table. |
| J33 | Number of independent variables in the Shock Absorber No. 3 table. |
| K | First digit of the table code number. |
| KSUM | Sum of the first digits of the code numbers. |
| L | Second digit of the table code numbers. |
| LF | Second table value in integer form. |
| LSUM | Sum of the second digits of the code numbers. |
| L3 | Total number of values in a given table. |
| M | Leg number corresponding to a given table. |
| SHOCK1 (N10) | Table of Characteristics for Shock Absorber No. 1. |
| SHOCK2 (N10) | Table of Characteristics for Shock Absorber No. 2. |
| SHOCK3 (N10) | Table of Characteristics for Shock Absorber No. 3. |
| STUPE | Temporary storage for the values on a given card. |

INPUT

| | |
|---|---|
| BETA (N7) | The angle between the vehicle centerline and a line joining the upper and lower hardpoints on a leg set (rad). |
| BETAD | Same as BETA except for units (degrees). |
| CODE | A numeric which controls program output. |
| DA (N9) | Utility array used for temporary storage. |
| DEGRAD (N7) | Conversion factor (rad/deg). |
| DELCG (N11) | Distance between CG and lower hardpoint measured along the vehicle centerline (feet). |
| DTP (N24) | Time interval for output printing and punching (sec). |

| | |
|---|---|
| ERO | Allowable integration truncation error per second as read in from cards. |
| ERR (N1) | Allowable integration truncation error per second (ft/sec, ft/sec/sec, degrees/sec, degrees/sec/sec). |
| FP (N11) | Shock Absorber preload (lb). |
| G (N11) | Gravitational acceleration (ft/sec/sec). |
| HBO (N11) | Initial height of the crushable element in the blocks (feet). |
| HFO (N11) | Initial height of the crushable element in the footpads (feet). |
| HM (N1) | Minimum integration interval size (sec). |
| ICODE | Equivalent to CODE2 on Input Data in integer form, controls punch output. |
| IOFF | Program switch which turns printing and punching completely off if certain output options are requested. |
| IOPT (N24) | Program switches which turn on or off various output print options and enable corresponding punch options. |
| IWILEY (N25) | Program switches which turn on or off selected punch options which have also been enabled by IOPT |
| J | DO Index. |
| JN (N24) | Program switch which turns on all punch options which have corresponding print options turned on. |
| JNO | Similar to JN. This value is set from input data and is then held for setting JN at the start of each run. |
| K | DO Index. |
| MISS (N23) | Program switch which controls matrix inversion of the inertia matrix in subroutine INTEQM. Inversion takes place only if new inertia data is read. |
| MM | Value corresponding to input card number. |
| NPHASE (N11) | Program switch which controls the type of linear velocity which will be accepted as input data. Printed on input data sheet as PHASE NO. |
| OMEG (N7) | Vehicle angular velocity in vehicle coordinates (rad/sec). |
| OMEGA | Initial vehicle angular velocity in vehicle coordinates (deg/sec). |

| | |
|---|---|
| PHIO (N7) | Initial yaw angle (rad). |
| PHIOD | Initial yaw angle (deg). |
| PI | Numerical constant, pi. |
| PSIO (N7) | Initial vehicle pitch angle (rad). |
| PSIOD | Initial vehicle pitch angle (deg). |
| RADDEG (N7) | Conversion factor (deg/rad). |
| RC (N11) | Damping coefficient for shock absorber during stroking ($lb\text{-}sec^2/ft^2$). |
| RR (N11) | Damping coefficient for shock absorber during restroking ($lb\text{-}sec^2/ft^2$). |
| SERNO (N11) | Run identification number. |
| SRL (N11) | Coefficient used to determine damping profile for shock absorber restroking ($ft^{-1}$). |
| TAU (N12) | Footpad contact times for option 4 (sec). |
| TF (N1) | Final integration time (sec). |
| TFO | Final integration time as read from input data (sec). |
| THETA (N7) | Angular orientation of leg sets relative to y-axis (rad). |
| THETAS (N7) | Ground slope (rad). |
| THETD | Angular orientation of leg sets relative to y-axis (deg). |
| THETSD | Ground slope (deg). |
| VH (N11) | Vehicle horizontal velocity in ground coordinates (ft/sec). |
| VV (N11) | Vehicle vertical velocity in ground coordinates (ft/sec). |
| WILEY | Program switch equivalent to CODE2 in input data printout, which controls output punching. |
| XDOTOC (N12) | Initial vehicle velocity in the X-direction (ft/sec). |
| XDTP | Time interval for output printing and punching as read from data cards (sec). |
| XIGAM (N11) | Moment of Inertia of lower leg ($ft\text{-}lb\text{-}sec^2$). |

| | |
|---|---|
| XIO (N7) | Initial roll angle of the vehicle (rad). |
| XIOD | Initial roll angle of the vehicle (deg). |
| XIX, XIY, XIZ (N11) | Moment of inertia of vehicle (ft-lb-sec$^2$). |
| XJ (N7) | Inertia matrix of the vehicle (ft-lb-sec$^2$). |
| XKD (N11) | Nominal shock absorber spring rate (lb/ft). |
| XKS (N11) | Shock absorber spring rate for small deflections (lb/ft). |
| XLA (N11) | Distance between upper and lower hardpoints (ft). |
| XLAMDA (N11) | Angular orientation of the horizontal velocity vector with respect to the Y-axis (rad). |
| XLAMDD | Angular orientation of the horizontal velocity vector with respect to the Y-axis (deg). |
| XLC (N11) | Vehicle centerline to block centerline distance (ft). |
| XLDMIN (N11) | Minimum shock absorber length (ft). |
| XLDO (N11) | Initial length of shock absorber (ft). |
| XLL (N11) | Length of lower link of a leg set (ft). |
| XLP (N11) | Vehicle centerline to lower hardpoint distance (ft). |
| XM (N11) | Vehicle mass (lb-sec$^2$/ft). |
| XMUB (N11) | Block friction coefficient. |
| XMUF (N11) | Footpad friction coefficient. |
| XNU (N11) | Shock absorber friction damping coefficient. |
| YDOTOC (N12) | Initial vehicle velocity in Y-direction - Ground Coordinates (ft/sec). |
| YIY, YIZ (N11) | Moment of inertia of vehicle (lb-ft-sec$^2$). |
| ZDOTOC (N12) | Initial vehicle velocity - Z-direction - Ground Coordinates (ft/sec). |
| ZIZ | Moment of inertia of vehicle (ft-lb-sec$^2$). |

INIT

| | |
|---|---|
| A (N13) | Utility array. |
| AA (N13) | Utility array. |
| ALPH (N7) | Footpad angle (rad). |
| A1 (N9) | Utility array. |
| A7 (N9) | Utility array. |
| B (N15) | Transformation matrix - vehicle to ground coordinates - corresponds to $b_{rs}$ matrix. |
| BETA (N7) | The angle between the vehicle centerline and a line joining the upper and lower hardpoints on a leg set (rad). |
| COSTH (N14) | Cosine of THETA, the angle between a leg set and the y-axis. |
| COSTHS (N14) | Cosine of ground slope angle, THETAS. |
| DA | Utility array. |
| DELCG (N11) | Distance between CG and lower hardpoint measured along the vehicle centerline (feet). |
| DELEL (N14) | Shock absorber preload divided by spring rate (feet). |
| DELMAX (N11) | Maximum shock absorber stroke (feet). |
| Dnn (N9) | Utility variables. |
| DRS (N15) | Transformation matrix-footpad to vehicle coordinates - corresponds to $d_{rs}$ matrix. |
| FP (N11) | Shock absorber preload (lb.). |
| G (N11) | Gravitational acceleration (ft/sec$^2$). |
| GAMDOT (N7) | Lower leg angular rates (rad/sec). |
| GAMMA (N7) | Lower leg angular position (rad). |
| GAMMAO | Lower leg initial angular position (deg). |
| HBO (N11) | Uncrushed height of blocks (feet). |
| HFO (N11) | Uncrushed height of footpads (feet). |
| HPL (N9) | Position of the lower hardpoints in vehicle coordinates (ft). |

| | |
|---|---|
| I | DO Index. |
| IBEEN (N19) | Program switch used in conjunction with footpad and block contact time printout. |
| ISET (N12) | Program switch used in conjunction with option 4. |
| J | DO Index. |
| K | DO Index. |
| L | DO Index. |
| L1 | DO Index. |
| L2 | DO Index |
| NPHASE (N11) | Program switch which controls the type of linear velocity which will be accepted as input data; printed on input data sheet as PHASE NO. |
| OMEG (N7) | Vehicle angular velocity in vehicle coordinates (rad/sec). |
| PFLAG (N7) | Program switch which is set in the event one or more shock absorbers are bottomed. |
| PHI (N7) | Vehicle angular position in ground coordinates (rad). |
| PHIO (N7) | Initial vehicle angular position in ground coordinates (rad). |
| PSB (N14) | Amount of block crushing (ft). |
| PSBQ (N19) | Amount of block crushing at the start of an integration interval (ft). |
| PSI (N7) | Vehicle pitch angle (rad). |
| PSIO (N7) | Initial pitch angle (rad). |
| PSP (N14) | Amount of footpad crushing (ft). |
| PSPQ (N19) | Amount of footpad crushing at the start of an integration interval (ft). |
| SINTH (N14) | Sine of the angular orientation of leg sets relative to y-axis. |
| SINTHS (N14) | Sine of the ground slope angle. |
| THETA (N7) | Angular orientation of leg sets relative to y-axis (rad). |
| THETAS (N7) | Ground slope (rad). |
| TZERO (N14) | Initial kinetic energy (ft-lb). |

| | |
|---|---|
| VH (N11) | CG horizontal velocity (ft/sec). |
| VV (N11) | CG vertical velocity (ft/sec). |
| VZERO (N14) | Initial potential energy (ft-lb). |
| XA (N12) | Footpad position at contact - X-direction - Ground Coordinates (ft). |
| XB (N15) | Block position - Vehicle Coordinates - x-direction (ft). |
| XC (N7) | CG position - X-direction - Ground Coordinates (ft). |
| XDOTC (N7) | CG velocity - X-direction - Ground Coordinates (ft/sec). |
| XDOTOC (N12) | Initial CG velocity - X-direction - Ground Coordinates (ft/sec). |
| XFP (N15) | Distance from CG to footpad pivot - x-direction - Vehicle Coordinates (ft). |
| XHPU (N14) | Distance from CG to upper hardpoint - x-direction - Vehicle Coordinates (ft). |
| XI (N7) | Roll angle of the vehicle (rad). |
| XIO (N7) | Initial roll angle of the vehicle (rad). |
| XJ (N7) | Vehicle inertia matrix (ft.-lb-sec$^2$). |
| XKS (N11) | Shock absorber spring rate (lb/ft). |
| XLA (N11) | Distance between upper and lower hardpoints (ft). |
| XLAMDA (N11) | Angular orientation of the horizontal velocity vector with respect to the Y-axis (rad). |
| XLC (N11) | Vehicle centerline to block centerline distance (ft). |
| XLDMIN (N11) | Minimum shock absorber length (ft). |
| XLDO (N11) | Initial length of shock absorber (ft). |
| XLL (N11) | Length of lower link of a leg set (ft). |
| XLP (N11) | Vehicle centerline to lower hardpoint distance (ft). |
| XM (N11) | Vehicle mass (lb-sec$^2$/ft). |
| YA (N12) | Footpad position at contact - Y-direction - Ground Coordinates (ft). |
| YB (N15) | Block position - Vehicle Coordinates - y-direction (ft). |

| | |
|---|---|
| YC (N7) | CG position - Y-direction - Ground Coordinates (ft). |
| YDOTC (N7) | CG velocity - Y-direction - Ground Coordinates (ft/sec). |
| YDOTOC (N12) | Initial CG velocity - Y-direction - Ground Coordinates (ft/sec). |
| YFP (N15) | Distance from CG to footpad pivot - y-direction - Vehicle Coordinates (ft). |
| YHPU (N14) | Distance from CG to upper hardpoint - y-direction - Vehicle Coordinates (ft). |
| ZA (N12) | Footpad position at contact - Z-direction - Ground Coordinates (ft). |
| ZBO (N14) | Distance from CG to the uncrushed, contacting surface of the blocks in the z-direction, vehicle coordinates (ft). |
| ZC (N7) | CG position - Z-direction - Ground Coordinates (ft). |
| ZCPMAX | Maximum distance between CG and footpad surface - Z′-direction - Surface Coordinates (ft). |
| ZDOTC (N7) | CG velocity - Z-direction - Ground Coordinates (ft/sec). |
| ZDOTOC (N12) | Initial CG velocity - Z-direction - Ground Coordinates (ft/sec). |
| ZFP (N15) | Distance from CG to footpad pivot - z-direction - Vehicle Coordinates (ft). |
| ZHPU (N14) | Distance from CG to upper hardpoint - z-direction - Vehicle Coordinates (ft). |
| ZPREF (N14) | Distance from CG to footpad uncrushed surface plus .50ft - z-direction - Vehicle Coordinates (ft). |

FEP

| | |
|---|---|
| A (N13) | Utility array used in the calculation of the final kinetic energy. |
| COSTHS (N14) | Cosine of the ground slope angle. |
| D (N2) | Derivative array which is returned to Runge-Kutta. |
| DA (N9) | Utility array. |
| DWORK (N20) | An array of energy and work rates (ft-lb/sec). |
| D59, D69 | Intermediate variable used in the calculation of the final kinetic energy. |
| EF | Final kinetic energy (ft-lb). |

| | |
|---|---|
| EPD | Potential energy difference (ft-lb). |
| GAMDD (N6) | Lower leg angular acceleration (rad/sec$^2$). |
| GAMDOT (N7) | Lower leg angular velocity (rad/sec). |
| GAMMA (N7) | Lower leg angular position (rad). |
| GCMIN | Minimum ground clearance (ft). |
| I | DO Index. |
| IBEEN (N19) | Program switch used in conjunction with footpad and block contact time printout. |
| IOPT (N24) | Program switch used to control output options. |
| ISTABL | Program switch which is set as a function of termination situation. |
| J | DO Index. |
| K | DO Index. |
| KFLAG (N7) | Program switch set in subroutine STAB to indicate a stable or unstable vehicle. |
| LFLAG (N8) | Program switch which indicates to the integration routine that the run is to be terminated. |
| NFLAG (N18) | Program switch which indicates various stages of the integration process. |
| NNFLAG (N19) | Program switch which indicates whether the blocks are in contact with the surface. |
| OMEG (N7) | Vehicle angular velocity in vehicle coordinates (rad/sec). |
| OMEGDT (N6) | Vehicle angular acceleration in vehicle coordinates (rad/sec$^2$). |
| PFLAG (N7) | Program switch which is set in the event one or more shock absorbers are bottomed. |
| PHI (N7) | Vehicle yaw angle (rad). |
| PHID (N6) | Vehicle yaw rate in ground coordinates (rad/sec). |
| PSI (N7) | Vehicle pitch angle (rad). |
| PSID (N6) | Vehicle pitch rate in ground coordinates (rad/sec). |
| Q (N6) | Vehicle acceleration in ground coordinates (ft/sec$^2$). |

| | |
|---|---|
| RADDEG (N7) | Conversion factor (deg/rad). |
| SHKMAX | Maximum shock absorber force (lb). |
| SIGMIN | Minimum stability angle (rad). |
| SINTHS (N14) | Sine of the ground slope angle. |
| T, TA | Instantaneous value of time (sec). |
| TZERO (N14) | Initial kinetic energy (ft-lb). |
| UZERO (N14) | Initial potential energy (ft-lb). |
| W (N9) | Utility array used for the storage of values for printing. |
| WORK | Dissipated energy (ft-lb). |
| XC (N7) | CG position - X-direction - Ground Coordinates (ft). |
| XDOTC (N7) | CG velocity - X-direction - Ground Coordinates (ft/sec). |
| XI (N7) | Roll angle of the vehicle (rad). |
| XID (N6) | Roll rate of the vehicle in ground coordinates (rad/sec). |
| XIGAM (N11) | Moment of inertia of lower leg (ft-lb-$sec^2$) |
| XJ (N7) | Inertia matrix of the vehicle (ft-lb-$sec^2$). |
| XM (N11) | Vehicle mass (lb-$sec^2$/ft). |
| Y (N2) | The array of integrated variables. |
| YC (N7) | CG position - Y-direction - Ground Coordinates (ft). |
| YDOTC (N7) | CG velocity - Y-direction - Ground Coordinates (ft/sec). |
| ZC (N7) | CG position - Z-direction - Ground Coordinates (ft) |
| ZDOTC (N7) | CG velocity - Z-direction - Ground Coordinates (ft/sec). |
| ZPREF (N14) | Distance from CG to footpad uncrushed surface plus .50 ft. - z-direction - Vehicle Coordinates (ft). |

<u>CONFIG</u>

| | |
|---|---|
| A (N13) | Utility array. |
| AA (N13) | Utility array. |

| | |
|---|---|
| AMAG | Magnitude of horizontal component of surface unit normal (ft). |
| AX | X component in ground coordinates of vector $\vec{A}$ from impact point of footpad No. 1 to No. 2 (ft). |
| AXBMAG | Magnitude of $\vec{A} \times \vec{B}$, a vector normal to the surface (ft). |
| AXBX | X component in ground coordinates of $\vec{A} \times \vec{B}$ (ft). |
| AXBY | Y component in goound coordinates of $\vec{A} \times \vec{B}$ (ft). |
| AXBZ | Z component in ground coordinates of $\vec{A} \times \vec{B}$ (ft). |
| AY | Y component in ground coordinates of vector $\vec{A}$ from impact point of footpad No. 1 to No. 2 (ft). |
| AZ | Z component in ground coordinates of vector $\vec{A}$ from impact point of footpad No. 1 to No. 2 (ft). |
| A1 (N9) | Utility array. |
| A6 (N9) | Utility array. |
| A7 (N9) | Utility variable. |
| A8 (N9) | Utility array. |
| B (N15) | Transformation matrix - vehicle to ground coordinates - corresponds to $b_{rs}$. |
| BBB | Program flag used to determine sign of cross slope angle. |
| BX | X component of vector $\vec{B}$ from impact point of footpad No. 1 to No. 3 (ft). |
| BY | Y component of vector $\vec{B}$ from impact point of footpad No. 1 to No. 3 (ft). |
| BZ | Z component of vector $\vec{B}$ from impact point of footpad No. 1 to No. 3 (ft). |
| COSLAM | Cosine of cross slope angle $\lambda$ . |
| COSPHI | Cosine of PHIO when used as Euler angle relative to new ground coordinates. |
| COSPSI | Cosine of PSIO when used as Euler angle relative to new ground coordinates. |
| COSTH (N14) | Cosine of THETA, the angle between a leg set and the y-axis. |
| COSTHS (N14) | Cosine of ground slope angle, THETAS. |

| | |
|---|---|
| COSXI | Cosine of XIO when used as Euler angle relative to new ground coordinates. |
| COSXI (N16) | Cosine of angle between surface normal and footpad centerline. |
| COSXIB | Cosine of angle between surface normal and block centerline. |
| COSZET | Cosine of ZETA. |
| CRANK1 | Utility variable used in definition of initial Euler angles relative to new ground coordinates. |
| CRANK2 | Utility variable used in definition of initial Euler angles relative to new ground coordinates. |
| CRANK3 | Utility variable used in definition of initial Euler angles relative to new ground coordinates. |
| DELCG (N11) | Distance between CG and lower hardpoint measured along the vehicle centerline (ft). |
| D1, D2, ---, D20 | Utility variables. |
| DRS (N15) | Transformation matrix-footpad to vehicle coordinates - corresponds to $d_{rs}$ matrix. |
| FOOT (N22) | Position of footpad pivot in surface coordinates (ft). |
| GAMDOT (N7) | Lower leg angular rates (rad/sec). |
| GAMMA (N7) | Lower leg angular position (rad). |
| GRS (N16) | Transformation matrix - strut to vehicle coordinates - corresponds to $g_{rs}$ matrix. |
| HBO (N11) | Uncrushed height of blocks (ft). |
| HFO (N11) | Uncrushed height of footpads (ft). |
| HPL (N9) | Position of the lower hardpoints in vehicle coordinates (ft). |
| I | DO Index. |
| IOPT (N24) | Program switches which turn on or off various output print options and enable corresponding punch options. |
| ISET (N12) | Program switch used in conjunction with option 4. |
| ISUM | Program switch which senses when all the footpads have contacted the surface during option 4. |

| | |
|---|---|
| J | DO Index. |
| L | DO Index. |
| LFLAG (N8) | Program switch which indicates to the integration routine that the run is to be terminated. |
| L2 | DO Index. |
| NFLAG (N18) | Program switch which indicates various stages of the integration process. |
| NNFLAG (N19) | Program switch which indicates whether the blocks are in contact with the surface. |
| NPHASE (N11) | Program switch which controls the type of linear velocity which will be accepted as input data. Printed on input data sheet as PHASE NO. |
| OMEG (N7) | Vehicle angular velocity in vehicle coordinates (rad/sec). |
| PHI (N7) | Vehicle yaw angle (rad). |
| PHIO (N7) | Initial vehicle yaw angle relative to original or new ground coordinates (rad). |
| PI (N7) | Numerical constant, $\pi$. |
| PSB (N14) | Amount of block crushing (ft). |
| PSBQ (N19) | Amount of block crushing at the start of an integration interval (ft). |
| PSI (N7) | Vehicle pitch angle (rad). |
| PSIO (N7) | Initial pitch angle relative to original or new ground coordinates (rad). |
| PSP (N14) | Amount of footpad crushing (ft). |
| PSPQ (N19) | Amount of footpad crushing at the start of an integration interval (ft). |
| RADDEG (N7) | Conversion factor (deg/rad). |
| RN (N16) | A unit vector, normal to the surface in vehicle coordinates. |
| S (N16) | Sliding speed of footpad sections (ft/sec). |
| SINPHI | Sine of PHIO when used as Euler angle relative to new ground coordinates. |
| SINPSI | Sine of PSIO when used as Euler angle relative to new ground coordinates. |
| SINTH (N14) | Sine of the angular orientation of leg sets relative to y-axis. |
| SINTHS (N14) | Sine of the ground slope angle. |
| SINXI | Sine of XIO when used as Euler angle relative to new ground coordinates. |

| | |
|---|---|
| SINZET | Cosine of ZETA. |
| STUFF | Utility variable used in definition of initial Euler angles relative to new ground coordinates. |
| SUMY | The penetration of a footpad in the Y-direction, ground coordinates during option 4 (ft). |
| SUMZ | The penetration of a footpad in the Z-direction, ground coordinates during option 4 (ft). |
| TA (N16) | Instantaneous value of time (sec). |
| TAU (N12) | Footpad contact times for option 4 (sec). |
| THEDEG | Ground slope (deg). |
| THETAS (N7) | Ground slope (rad). |
| UAX | X component in original ground coordinates of horizontal unit vector in direction of maximum uphill slope (ft). |
| UAY | Y component in original ground coordinates of horizontal unit vector in direction of maximum uphill slope (ft). |
| UNX | X component in original ground coordinates of unit surface normal (ft). |
| UNY | Y component in original ground coordinates of unit surface normal (ft). |
| UNZ | Z component in original ground coordinates of unit surface normal (ft). |
| VH (N11) | CG horizontal velocity (ft/sec). |
| VHORIZ | CG horizontal velocity for option 4 (ft/sec). |
| VV (N11) | CG vertical velocity (ft/sec). |
| VVERT | CG vertical velocity for option 4 (ft/sec). |
| W (N9) | Utility array used for the storage of values for printing. |
| XA (N12) | Footpad position at contact - x-direction - Ground Coordinates (ft). |
| XB (N15) | Block position - Vehicle coordinates - x-direction (ft). |
| XC (N7) | CG position - x-direction - Ground Coordinates (ft). |
| XDOTC (N7) | CG velocity - x-direction - Ground Coordinates (ft/sec). |
| XDOTOC (N12) | Initial CG velocity - x-direction - Ground Coordinates (ft/sec). |

| | |
|---|---|
| XDOTOQ | Initial CG velocity, X-direction, ground coordinates used in option 4 (ft/sec). |
| XDOTP (N16) | Footpad velocity - X′-direction - Surface Coordinates (ft/sec). |
| XFP (N15) | Distance from CG to footpad pivot - x-direction - Vehicle Coordinates (ft). |
| XFPC (N17) | Distance from CG to footpad pivot - X-direction - Ground Coordinates (ft). |
| XI (N7) | Roll angle of the vehicle (rad). |
| XII (N16) | Angle between surface normal and block or footpad centerline. |
| XIO (N7) | Initial roll angle of vehicle relative to original or new ground coordinates (rad). |
| XLA (N11) | Distance between upper and lower hardpoints (ft). |
| XLAMDA (N11) | Angular orientation of the horizontal velocity vector with respect to the Y-axis (rad). |
| XLD (N16) | Length of the shock absorber (ft). |
| XLDDOT (N16) | Shock absorber stroking velocity (ft/sec). |
| XLL (N11) | Length of lower link of a leg set (ft). |
| XLP (N11) | Vehicle centerline to lower hardpoint distance (ft). |
| XSLDEG | Calculated cross slope angle (deg). |
| XSLOPE | Calculated cross slope angle (rad). |
| YA (N12) | Footpad position at contact - Y-direction - Ground Coordinates (ft). |
| YB (N15) | Block position - Vehicle Coordinates - y-direction (ft). |
| YC (N7) | CG position - Y-direction - Ground Coordinates (ft). |
| YDOTC (N7) | CG velocity - Y-direction - Ground Coordinates (ft/sec). |
| YDOTOC (N12) | Initial CG velocity - Y-direction - Ground Coordinates (ft/sec). |
| YDOTOQ | Initial CG velocity - Y-direction - Ground Coordinates - used in option 4 (ft/sec). |
| YDOTP (N16) | Footpad velocity - Y′-direction - Surface Coordinates (ft/sec). |
| YFP (N15) | Distance from CG to footpad pivot - y-direction - Vehicle Coordinates (ft). |
| YFPC (N17) | Distance from CG to footpad pivot - Y-direction - Ground Coordinates (ft). |

ZA (N12) Footpad position at contact - Z-direction - Ground Coordinates (ft).

ZB (N15) Block position - Vehicle Coordinates - z-direction (ft).

ZBO (N14) Distance from CG to the uncrushed, contacting surface of the blocks in the z-direction-Vehicle Coordinates (ft).

ZC (N7) CG position - Z-direction - Ground Coordinates (ft).

ZDOTC (N7) CG velocity - Z-direction - Ground Coordinates (ft/sec).

ZDOTP (N16) Footpad velocity - $Z'$-direction - Surface Coordinates (ft/sec).

ZETA Angle of rotation between original and new ground coordinates (rad).

ZETDEG Angle of rotation between original and new ground coordinates (deg).

ZFP (N15) Distance from CG to footpad pivot - z-direction - vehicle coordinates (ft).

ZFPC (N17) Distance from CG to footpad pivot - Z-direction - Ground Coordinates (ft).

STAB

AW Magnitude of $\vec{A}$, a vector from CG to intercept of principal plane by line joining critical footpads (ft).

AX X-component of $\vec{A}$ in ground coordinates (ft).

AY Y-component of $\vec{A}$ in ground coordinates (ft).

AZ Z-component of $\vec{A}$ in ground coordinates (ft).

DA (N9) Utility array.

D1 (N9) Utility variable used here in definition of $\vec{A}$.

IFLAG Program switch.

II Program switch.

J DO Index.

JJ Program switch.

KFLAG (N7) Program switch set to indicate a stable or unstable vehicle.

N Program switch.

THETAS (N7) Ground slope (rad).

| | |
|---|---|
| W (N9) | Utility array used for the storage of values for printing; used here for stability angle. |
| XDOTC (N7) | CG velocity - X-direction - Ground Coordinates (ft/sec). |
| XFPC (N17) | Distance from CG to footpad pivot - X-direction - Ground Coordinates (ft). |
| XX (N9) | Utility variable used to determine critical footpads. |
| XXX | Utility variable used to define $\vec{A}$. |
| YDOTC (N7) | CG velocity - Y-direction - Ground Coordinates (ft/sec). |
| YFPC (N17) | Distance from CG to footpad pivot - Y-direction - Ground Coordinates (ft). |
| ZFPC (N17) | Distance from CG to footpad pivot - Z-direction - Ground Coordinates (ft). |

FORCE

| | |
|---|---|
| A (N13) | Utility array. |
| A1 (N9) | Utility array. |
| A2 (N9) | Utility array. |
| A3 (N9) | Utility array. |
| A4 (N9) | Utility array. |
| A5 (N9) | Utility array. |
| A6 (N9) | Utility array. |
| B (N15) | Transformation matrix - vehicle to ground coordinates - corresponds to $b_{rs}$ matrix. |
| BLOCKn(N10) | Table of block crush characteristics. |
| COSTHS (N14) | Cosine of ground slope angle, THETAS. |
| COSXI (N16) | Cosine of angle between surface normal and footpad centerline. |
| DA (N9) | Utility array. |
| DEL | Instantaneous shock absorber stroke (ft). |
| DELEL (N14) | Shock absorber preload divided by spring rate (ft). |
| DELMAX (N11) | Maximum shock absorber stroke (ft). |

| | |
|---|---|
| DELWB | The rate energy is being absorbed due to block crushing (ft-lb/sec). |
| DELWD | The rate energy is being absorbed or stored by the shock absorber (ft-lb/sec). |
| DELWF | The rate energy is being absorbed due to footpad crushing (ft-lb/sec). |
| DELWU | The rate energy is being dissipated due to footpad and block friction (ft-lb/sec). |
| DWORK (N20) | An array of energy rate terms (ft-lb/sec). |
| Dnn (N9) | Utility variables. |
| DRS (N15) | Transformation matrix-footpad to vehicle coordinates - corresponds to $d_{rs}$ matrix. |
| F(N9) | Force on the vehicle CG in ground coordinates (lb). |
| FACTOR | Utility variable. |
| FOOTn (N10) | Table of footpad crush characteristics. |
| FP (N11) | Shock absorber preload (lb). |
| GADOT | Absolute value of GAMDOT (rad/sec). |
| GAMDOT (N7) | Lower leg angular rates (rad/sec). |
| GRS (N16) | Transformation matrix - strut to vehicle coordinates - corresponds to $g_{rs}$ matrix. |
| HPL (N9) | Position of the lower hardpoints in vehicle coordinates (ft). |
| I | DO Index. |
| IOPT (N24) | Program switches which turn on or off various output print options and enable corresponding punch options. |
| J | DO Index. |
| JFLAG | Program switch which is set as a function of the sign of shock absorber stroking velocity. |
| K | DO Index. |
| KDUM | Program switch used in subroutine ENTRP. |
| L | Program switch which is set as a function of shock absorber stroke. |
| MAGIC | Program switch which is set if the shock absorber is restroking, the footpad is on the surface and the shock absorber is on or beyond the extension stop. |

| | |
|---|---|
| MFLAG | Program switch which is set if a footpad is off the surface. |
| MUMP | Program switch which is set if the shock absorber is restroking, the footpad is off the surface and the shock absorber is on or beyond the extension stop. |
| N | DO Index. |
| NFLAG (N18) | Program switch which indicates various stages of the integration process. |
| NNFLAG (N19) | Program switch which indicates whether the blocks are in contact with the surface. |
| PFLAG (N7) | Program switch which is set in the event one or more shock absorbers are bottomed. |
| PSB (N14) | Amount of block crushing (ft). |
| PSBQ (N19) | Amount of block crushing at the start of an integration interval (ft). |
| PSP (N14) | Amount of footpad crushing (ft). |
| PSPQ (N19) | Amount of footpad crushing at the start of an integration interval (ft). |
| R | Forces at the footpad pivot in strut coordinates (lb). |
| RC (N11) | Damping coefficient for shock absorber during stroking ($lb\text{-}sec^2/ft^2$). |
| RD | Damping coefficient for shock absorber; equal to RC or RR depending on shock absorber motion ($lb\text{-}sec^2/ft^2$). |
| RN (N16) | A unit vector normal to the surface in vehicle coordinates. |
| RR (N11) | Damping coefficient for shock absorber during restroking ($lb\text{-}sec^2/ft^2$). |
| S (N16) | Sliding speed of footpad section (ft/sec). |
| SD | Shock absorber damping profile value. |
| SHOCKn (N10) | Table of shock absorber rate and damping characteristics. |
| SINTHS (N14) | Sine of the ground slope angle. |
| SK | Shock absorber spring rate profile value. |
| SRL (N11) | Coefficient used to determine damping profile for shock absorber restroking ($ft^{-1}$). |
| TGAMMA (N5) | Torque about the lower hardpoint (ft-lb). |
| TQ (N5) | Torque about the CG (ft-lb). |

| | |
|---|---|
| W (N9) | Utility array used for the storage of values for printing. |
| XB (N15) | Block position - Vehicle Coordinates - x-direction (ft). |
| XDOTP (N16) | Footpad velocity - X′-direction - Surface coordinates (ft/sec). |
| XFP (N15) | Distance from CG to footpad pivot - x-direction - Vehicle Coordinates (ft). |
| XHPU (N14) | Distance from CG to upper hardpoint - x-direction - Vehicle Coordinates (ft). |
| XIGAM (N11) | Moment of inertia of lower leg (ft-lb-sec$^2$). |
| XKD (N11) | Nominal shock absorber spring rate (lb/ft). |
| XKS (N11) | Shock absorber spring rate for small deflections (lb/ft). |
| XII (N16) | Angle between surface normal and block or footpad centerline. |
| XLD (N16) | Length of the shock absorber (ft). |
| XLDDOT (N16) | Shock absorber stroking velocity (ft/sec). |
| XLDO (N11) | Initial length of shock absorber (ft). |
| XLL (N11) | Length of lower link of a leg set (ft). |
| XMUB (N11) | Block friction coefficient. |
| XMUF (N11) | Footpad friction coefficient. |
| XNU (N11) | Shock absorber friction damping coefficient. |
| YB (N15) | Block position - Vehicle Coordinates - y-direction (ft). |
| YDOTP (N16) | Footpad velocity - Y′-direction - Surface Coordinates (ft/sec). |
| YFP (N15) | Distance from CG to footpad pivot - y-direction - Vehicle Coordinates (ft). |
| YHPU (N14) | Distance from CG to upper hardpoint - y-direction - Vehicle Coordinates (ft). |
| ZB (N15) | Block position - Vehicle Coordinates - z-direction (ft). |
| ZDOTP (N16) | Footpad velocity - Z′-direction - Surface Coordinates (ft/sec). |
| ZFP (N15) | Distance from CG to footpad pivot - z-direction - Vehicle Coordinates (ft). |
| ZHPU (N14) | Distance from CG to upper hardpoint - z-direction - Vehicle Coordinates (ft). |

ENTRP

| | |
|---|---|
| FX | Array of interpolated values. |
| FX1 | First interpolated value. |
| FX2 | Second interpolated value. |
| I | DO Index. |
| J | Integer value of the second table value, the number of corresponding values in the table. |
| K | Program switch which specifies the number of values to be interpolated. |
| L | Subscript of FX array. |
| M | Subscript of the dependent variable which is the first value greater than the value which will finally be the output. |
| M1 | Subscript of the last dependent variable. |
| N | Program switch. |
| T | The specified table. |
| X | The value of the independent variable which will control the value of the dependent variable(s). |

INTEQM

| | |
|---|---|
| A | Array used in the inversion of the inertia matrix. |
| AA | Array used in the inversion of the inertia matrix. |
| B (N15) | Transformation matrix - vehicle to ground coordinates - corresponds to $b_{rs}$ matrix. |
| BAZ | Utility array. |
| CPHI | Cosine of PHI. |
| CXI | Cosine of XI. |
| Dnn | Utility variables. |
| F (N9) | Force on the vehicle CG in ground coordinates (lb). |
| GAMDD (N6) | Lower leg angular acceleration (rad/sec$^2$). |

| | |
|---|---|
| I | DO Index. |
| J | DO Index. |
| K | Utility variable. |
| L | Utility variable. |
| LL | Utility variable. |
| M | DO Index. |
| MISS (N23) | Program switch which controls matrix inversion of the inertia matrix. |
| N | DO Index. |
| OMEG (N7) | Vehicle angular velocity in vehicle coordinates (rad/sec). |
| OMEGDT (N6) | Vehicle angular acceleration in vehicle coordinates (rad/sec$^2$). |
| PAR | Utility variable. |
| PHI (N7) | Vehicle yaw (rad). |
| PHID (N6) | Vehicle yaw rate in ground coordinates (rad/sec). |
| PSID (N6) | Vehicle pitch rate in ground coordinates (rad/sec). |
| Q (N6) | Linear accelerations of the vehicle CG in ground coordinates (ft/sec$^2$). |
| SPHI | Sine of PHI. |
| SUM | Utility variable. |
| SXI | Sine of XI. |
| TEMP | Utility variable. |
| TGAMMA (N5) | Torque about the lower hardpoint (ft-lb). |
| TPHI | Tangent of PHI. |
| TQ (N5) | Torque about the CG (ft-lb). |
| XID (N6) | Vehicle roll rate in ground coordinates (rad/sec). |
| XIGAM (N11) | Moment of inertia of lower leg (ft-lb-sec$^2$). |
| XIX, XIY, XIZ (N11) | Moment of inertia of vehicle (ft-lb-sec$^2$). |

| | |
|---|---|
| XJ (N7) | Inertia matrix of the vehicle (ft-lb-sec$^2$). |
| XJINV | Inverse of the inertia matrix of the vehicle (ft$^{-1}$-lb$^{-1}$-sec$^{-2}$). |
| XM (N11) | Vehicle mass (lb-sec$^2$/ft). |
| YIZ (N11) | Moment of inertia of vehicle (ft-lb-sec$^2$). |
| ZIZ (N11) | Moment of inertia of vehicle (ft-lb-sec$^2$). |

<u>RKP</u>

| | |
|---|---|
| A | The values of the integrated variables at the start of an integration interval. |
| AA | Utility variable. |
| ABSER | Absolute value of the integration truncation error. |
| ABSUM | An array of the absolute sums of truncation errors. |
| ARITH | An array of the arithmetic sums of truncation errors. |
| AVG | Average integration interval size (sec). |
| BB | Utility variable. |
| C | The values of the integrated variables at the center of an integration interval. |
| C1 | Utility variable. |
| D (N2) | The derivative array. |
| DD | Utility variable. |
| DD2 | Utility variable. |
| DY1 | An array of the derivatives evaluated at the beginning of an integration interval. |
| DY2 | An array of the derivatives evaluated at the center of an integration interval. |
| EE | Utility variable. |
| ERR (N1) | Allowable integration truncation error per second (ft/sec, ft/sec/sec, rad/sec, rad/sec/sec). |

| | |
|---|---|
| ERROR | Integration truncation error. |
| H (N3) | Integration interval size (sec). |
| HM (N1) | Minimum allowable integration interval (sec). |
| HP | Time between the start of the integration interval and the time for which interpolation of the integrated variables is to be done (sec). |
| HSQ | H squared ($sec^2$). |
| HTD | H cubed ($sec^3$). |
| HTH | H to the fourth power ($sec^4$). |
| H2 | One-half of the integration interval (sec). |
| I | DO Index. |
| ICNT | Integration interval counter. |
| IEDSW (N3) | Program switch which is used for early termination of the program. |
| IPTSW | Program switch which controls calls to subroutine PRT. |
| IRKC | Program switch which controls the Runge-Kutta process internally. |
| IRSSW | Program reset switch. |
| K | DO Index. |
| K1 | Cut distribution index. |
| K4 | Interval distribution index. |
| K9 | Critical variable index. |
| LFLAG (N8) | Program termination switch. |
| N(N1) | DO Index - corresponds to the number of variables to be integrated. |
| NFLAG (N18) | Program switch which indicates various stages of the integration process. |
| NK1 | Cut distribution counter. |
| NK4 | Interval distribution counter. |
| NK9 | Critical variable distribution counter. |
| SGN | Program switch. |

| | |
|---|---|
| SH (N1) | Suggested integration interval (sec). |
| T (N2) | Time (sec). |
| TF (N1) | Final Time (sec). |
| TO (N1) | Initial time (sec). |
| TP (N3) | Value of time sent to subroutine PRT (sec). |
| TPH | Value of TP prior to calling subroutine PRT (sec). |
| U (N3) | Array of integrated variables sent to subroutine PRT for output. |
| U2 | Ratio of truncation error to allowable error. |
| U2MAX | Maximum ratio of truncation error to allowable error. |
| V | Utility array. |
| XK1, XK2, XK3 | Arrays of constants used in the Runge-Kutta equations. |
| XLOG10 | Logarithm of 10 to the base e. |
| XLOG2 | Logarithm of 2 to the base e. |
| Y (N2) | Array of the integrated values sent to function evaluation for calculation of derivatives. |
| YI (N1) | Array of initial values of the variables to be integrated. |
| YO (N1) | Array of final values of the variables which were integrated. |
| ZAP | Utility variable. |
| ZIP | Utility variable. |

PRT

| | |
|---|---|
| A1 (N9) | Utility array. |
| CPHI | Cosine of PHI. |
| CXI | Cosine of XI. |
| DA (N9) | Utility array used for temporary storage. |
| DEGRAD (N7) | Conversion factor (rad/deg). |

| | |
|---|---|
| DTP (N24) | Time interval for output printing and punching (sec.) |
| D59 | Utility variable. |
| D69 | Utility variable. |
| EK | Instantaneous kinetic energy (ft-lb). |
| EP | Instantaneous difference in potential energy (ft-lb). |
| FOOT (N22) | Position of footpad pivot in surface coordinates (ft). |
| H (N1) | Integration interval size (sec). |
| I | DO Index. |
| IOPT (N24) | Program switches which turn on or off various output print options and enable corresponding punch options. |
| IWILEY (N25) | Program switches which turn on or off selected punch options which have also been enabled by IOPT. |
| J | DO Index. |
| JN (N24) | Program switch which turns on all punch options which have corresponding print options enabled. |
| K | DO Index. |
| L | Computed subscript. |
| LSET | Program switch. |
| M | Counter. |
| ODEAR | Utility array. |
| PHID | Vehicle yaw rate in ground coordinates (deg/sec). |
| PHI | Vehicle roll (rad). |
| PSID | Vehicle pitch rate in ground coordinates (rad/sec). |
| PSP (N14) | Amount of footpad crushing (ft). |
| RADDEG (N7) | Conversion factor (deg/rad). |
| SPHI | Sine of PHI. |
| SXI | Sine of XI. |

| | |
|---|---|
| TP (N3) | Value of time corresponding to the U array received (sec). |
| TPHI | Tangent of PHI. |
| U (N3) | An array of integrated, interpolated values for printing from subroutine RKP. |
| VZERO (N14) | Initial potential energy (ft-lb). |
| W (N9) | Utility array used for the storage of values for printing. |
| W342 | The value of W (3, 4, 2), stability angle, in degrees (deg). |
| XID | Roll rate of the vehicle in ground coordinates (deg/sec). |
| XIGAM (N11) | Moment of inertia of lower leg (ft-lb-sec$^2$). |
| XJ (N7) | Inertia matrix of the vehicle (ft-lb-sec$^2$). |
| XM (N11) | Vehicle mass (lb-sec$^2$/ft). |
| ZERO | Initial value of the Z-component of the CG position in ground coordinates (ft). |

### 3.4.3 Program Listing

A listing of the finalized hard surface version of the Surveyor landing dynamics computer program is reproduced on the following pages.

```
      COMMON/N1/  N,TC,TF,SH,HM,YI(50),YO(50),ERR(50)
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, FFLAG, FFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N12/XDOTOC,YDOTOC,ZDOTOC,TAU(3),ISET(3),XA(3),YA(3),ZA(3)
      COMMON/N15/ D(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DFS(3,3,3)
      COMMON/N24/IOPT(10), JN, DTF
      CALL CLOCK
      CALL   TABLIN
      IPRINT = 0
      TF     = 0.0
      HM     = 0.0
    1 TC     = 0.0
      WRITE (6,20)
   20 FORMAT (1H1)
      CALL   INPUT
      IF     (IOPT(3)) 21,21,22
   21 N      = 29
      GO TO  23
   22 N      = 39
   23 CALL   INIT
      IF     (IOPT(4)) 26,26,24
   24 TAMIN  = AMIN1(TAU(1), TAU(2), TAU(3))
      IF     (TAMIN) 26,26,25
   25 SH     = TAMIN + HM
      GO TO  27
   26 SH     = .0005
   27 DO     10 I = 1,3
      YI(I)  = ALPDOT(I)
      YI(I+3)= ALPH(I)
      YI(I+6)= GAMDOT(I)
      YI(I+9)= GAMMA(I)
   10 YI(I+18) = OMEG(I)
      YI(13) = XDOTC
      YI(14) = YDOTC
      YI(15) = ZDOTC
      YI(16) = XC
      YI(17) = YC
      YI(18) = ZC
      YI(22) = PSIO
      YI(23) = PHIO
      YI(24) = XIO
      DO     18 J= 25,50
   18 YI(J)  = 0.0
      DO     185 J= 4,12
  185 ERR(J) = ERR(J)*DEGRAD
      DO     186 J= 19,24
  186 ERR(J) = ERR(J) * DEGRAD
      CALL   RKP
      CALL CLOCK
      GO TO 1
      END
```

```
      SUEROUTINE TABLIN
      DIMENSION STUPE(8), CLCD(100), IFORK(18)
      COMMON/N10/FOOT1(100), FOOT2(100), FOOT3(100),
 101               BLOCK1(100), FLOCK2(100), BLOCK3(100),
 102               SHOCK1(100), SHOCK2(100), SHOCK3(100)
900   FORMAT (8F10.3)
      I        = 0
      READ     (5,900)  (STUPE(J), J=1,8)
 10   I        = I+1
      K        = IFIX(STUPE(1)/9.9)
      L        = IFIX(STUPE(1)-FLOAT(9.99*K))
      IFCRK(I) = K
      IFCRK(I+9) = L
      IF       ((K-2)/2) 30,20,30
 20   IF       ((L-2)/2) 30,100,30
 30   WRITE(6,40) K,L
 40   FORMAT (10X52H** TABLE ERROR NO. 1' FIRST NO. OUT OF RANGE.  NO. =
 401  I10,I10 //)
      STOP
100   DO       110 J= 1,8
110   CLOD(J) = STUPE(J)
      LF       = IFIX(CLOD(2))
      L3       = LF*2+2
140   IF       (LF) 150,150,170
150   WRITE (6,160) K,L
160   FORMAT (10X27H** TABLE ERROR NO. 3' LIST I1,I1,1X11HLENGTH = 0. //
1601    )
      STOP
170   IF       (K-3) 180,171,171
171   L3       = LF*3+2
180   IF       (L3-8) 190,190,182
182   IF       (L3-100) 189,189,184
184   WRITE    (6,186) K,L,L3
186   FORMAT (10X48H** TABLE ERROR NO. 2' LIST EXCEEDS ARRAY LENGTH. /
18610X12HLIST IS NO. I1,I1,1H./ 10X15HLIST LENGTH IS I10,1H. //)
      STOP
189   READ (5,900) (CLOD(J), J= 9,L3)
190   GO TO (200,300,400), K
200   GO TO  (210,240,270), L
210   DO 220 J= 1,L3
220   FOOT1(J) = CLOD(J)
      J11      = LF
      GO TO  500
240   DO 250 J= 1,L3
250   FOOT2(J) = CLOD(J)
      J12      = LF
      GO TO 500
270   DO 280 J= 1,L3
280   FOOT3(J) = CLOD(J)
      J13      = LF
      GO TO 500
300   GO TO (310,340,370), L
310   DO 320 J= 1,L3
320   BLOCK1(J) = CLOD(J)
      J21      = LF
      GO TO  500
```

```
340 DO 350 J= 1,L3
350 BLOCK2(J) = CLOD(J)
    J22     = LF
    GO TO 500
370 DO 380 J= 1,L3
380 BLOCK3(J) = CLOD(J)
    J23     = LF
    GO TO 500
400 GO TO (410,430,470),L
410 DO 420 J= 1,L3
420 SHOCK1(J) = CLOD(J)
    J31     = LF
    GO TO 500
430 DO 440 J= 1,L3
440 SHOCK2(J) = CLOD(J)
    J32     = LF
    GO TO 500
470 DO 480 J= 1,L3
480 SHOCK3(J) = CLOD(J)
    J33     = LF
500 READ    (5,900) (STUPE(J), J= 1,8)
    IF      (STUPE(1)) 10,510,10
510 KSUM    = 0
    LSUM    = 0
    DO      520 J= 1,I
    KSUM    = KSUM + IFORK(J)
520 LSUM    = LSUM + IFORK (J+9 )
    IF      (KSUM-18) 530,650,530
530 IF      (KSUM-6) 550,540,550
540 IF      (LSUM-3) 550,570,550
550 WRITE   (6,560) KSUM, LSUM
560 FORMAT  (10X47H** TABLE ERROR NO. 4' INSUFFICIENT TABLE DATA ( I10,
5601 3H),( I10, 2H), //)
    STOP
570 K       = 2+2*J11
    J12     = J11
    J13     = J11
    DO      580 J= 1,K
    FOOT2(J) = FOOT1(J)
580 FOOT3(J)= FOOT1(J)
    K       = 2+2*J21
    J22     =J21
    J23     =J21
    DO      590 J= 1,K
    BLOCK2(J) = BLOCK1(J)
590 BLOCK3(J) = BLOCK1(J)
    K       = 2 + 3*J31
    J32     =J31
    J33     =J31
    DO      600 J= 1,K
    SHOCK2(J) = SHOCK1(J)
600 SHOCK3(J) = SHOCK1(J)
650 M       = 1
    K       = J11+2
    WRITE   (6,660) M, (FOOT1(J), J=3,K)
660 FORMAT  (10X12HFOOT PAD NO. I1 / (7F10.3))
    L       = 3 + J11
    K       = 2 + 2*J11
```

```
      WRITE  (6,670)(FOOT1(J), J= L,K)
  670 FORMAT (7F10.3)
      M      = 2
      K      = J12+2
      WRITE  (6,660) M, (FOOT2(J), J=3,K)
      L      = 3 + J12
      K      = 2+2*J12
      WRITE  (6,670) ( FOOT2(J),J=L,K)
      M      = 3
      K      = J13+2
      WRITE  (6,660) M, (FOOT3(J), J=3,K)
      L      = 3+J13
      K      = 2+2*J13
      WRITE(6,670) ( FOOT3(J),J=L,K)
      M      = 1
      K      = J21+2
      WRITE  (6,700) M, (BLOCK1(J), J=3,K)
  700 FORMAT (10X9HBLOCK NO. I1/ (7F10.3))
      L      = 3+J21
      K      = 2+2*J21
      WRITE  (6,670) (BLOCK1(J), J= L,K)
      M      = 2
      K      = J22+2
      WRITE  (6,700) M, (BLOCK2(J), J= 3,K)
      L      = 3+J22
      K      = 2+2*J22
      WRITE  (6,670) (BLOCK2(J), J=L,K)
      M      = 3
      K      = J23+2
      WRITE  (6,700) M, (BLOCK3(J), J= 3,K)
      L      = 3+J23
      K      = 2+2*J23
      WRITE  (6,670) (BLOCK3(J), J= L,K)
      M      = 1
      K      = J31+2
      WRITE  (6,750) M, (SHOCK1(J), J=3,K)
  750 FORMAT (10X16HSHOCK ABSORBER NO. I1 / (7F10.3))
      L      = 3+J31
      K      = 2+2*J31
      WRITE  (6,670)(SHOCK1(J), J= L,K)
      L      = K+1
      K      = K+J31
      WRITE  (6,670) (SHOCK1(J), J= L,K)
      M      = 2
      K      = J32+2
      WRITE  (6,750) M, (SHOCK2(J), J= 3,K)
      L      = 3+J32
      K      = 2+2*J32
      WRITE  (6,670)(SHOCK2(J), J= L,K)
      L      = K+1
      K      = K+J32
      WRITE  (6,670)(SHOCK2(J), J= L,K)
      M      = 3
      K      = J33+2
      WRITE  (6,750) M, (SHOCK3(J), J= 3,K)
      L      = 3+J33
      K      = 2+2*J33
      WRITE  (6,670)(SHOCK3(J), J= L,K)
```

```
      L        = K+1
      K        = K+J33
      WRITE  (6,670)(SHOCK3(J), J= L,K)
      J11      = J11+2
      DO       810 J= 3,J11
  810 FOOT1(J) = FOOT1(J)/12.0
      J12      = J12+2
      DO       820 J= 3,J12
  820 FOOT2(J) = FOOT2(J)/12.0
      J13      = J13+2
      DO       830 J= 3,J13
  830 FOOT3(J) = FOOT3(J)/12.0
      J21      = J21+2
      DO       840 J= 3,J21
  840 BLOCK1(J) = BLOCK1(J)/12.0
      J22      = J22+2
      DO       850 J= 3,J22
  850         BLOCK2(J) = BLOCK2(J)/12.0
      J23      = J23+2
      DO       860 J= 3,J23
  860 BLOCK3(J) = BLOCK3(J)/12.0
      J31      = J31+2
      DO       870 J= 3,J31
  870 SHOCK1(J) = SHOCK1(J)/12.0
      J32      =J32+2
      DO       880 J= 3,J32
  880 SHOCK2(J) = SHOCK2(J) / 12.0
      J33      =J33+2
      DO       890 J= 3,J33
  890 SHOCK3(J) = SHOCK3(J) /12.0
  999 RETURN
      END
```

```
      SUBROUTINE INPUT
      DIMENSION XMFP(3), ALPHK(2)
      DIMENSION          OMEGA(3), THETD(3), ALPHAD(3) ,ERC(50)
      COMMON/N1/  N,TO,TF,SH,HM,YI(50),YO(50),ERR(50)
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, FFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N11/CL(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMFP,XIP,DELCG,XLP,
     1         XLL,XLA,G,ALPHK,XLC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2         XLDOP,XLDC,XKL(3),XKS,    XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3         XMUB,RUNNC,SEKNC,VV,VH,DB,XTALPH,XIGAM,DELMAX,SRL(3),FP(3)
      COMMON/N12/XDOTOC,YDOTOC,ZDOTOC,TAU(3),ISET(3),XA(3),YA(3),ZA(3)
      COMMON/N23/MISS
      COMMON/N24/IOPT(10), JN, DTF
      COMMON/N25/IWILEY(10)
      IOFF    = 0
       DTF    = 0.0
  100 DO 101 K=1,5
  101 DA(K) = 0.0
      READ (5,900) MM,( DA(J), J=1,5)
      IF      (MM)109,200,109
  109 GO TO   (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 ),MM
    1 XM  = DA(1)
      XIX = DA(2)
      YIY = DA(3)
      ZIZ = DA(4)
      XIY = DA(5)
      MISS   = 0
      WRITE (6,921) MM
      GO TO 100
    2 XIZ   = DA(1)
      YIZ   = DA(2)
      XMF   = DA(3)
      XLLO  = DA(4)
      DELCG= DA(5)
      MISS   = 0
      WRITE (6,921) MM
      GO TO 100
    3 XLP = DA(1)
      XLL = DA(2)
      XLA = DA(3)
      G            = DA(4)
      ALPHAD(1)= DA(5)
      WRITE (6,921) MM
      GO TO 100
    4 ALPHAD(2)= DA(1)
      ALPHK(1) = DA(2)
      ALPHK(2) = DA(3)
      BETAD      = DA(4)
      XLC        = DA(5)
      WRITE (6,921) MM
      GO TO 100
```

```
    5 HBC     = DA(1)
      XIALPH= DA(2)
      XICAM  = DA(3)
      HFC    = DA(4)
      DB     = DA(5)
      WRITE (6,921) MM
      GO TO 100
    6 LF     = DA(1)
      XDCTUC= DA(2)
      YDCTUC= DA(3)
      ZDCTUC= DA(4)
      WRITE (6,921) MM
      GO TO 100
    7 XML    = DA(1)
      XLLMIN= DA(2)
      XKS     = DA(3)
      WRITE (6,921) MM
      GO TO 100
    8 THETL(1) = DA(1)
      THETL(2) = DA(2)
      THETL(3) = DA(3)
      WRITE (6,921) MM
      GO TO 100
    9 THETSD = DA(1)
      PSIOL  = DA(2)
      PHIOL  = DA(3)
      XICD   = DA(4)
      XLAMUD = DA(5)
      WRITE (6,921) MM
      GO TO 100
   10 OMEGA(1)= DA(1)
      OMEGA(2)= DA(2)
      OMEGA(3)= DA(3)
      WILEY  = DA(4)
      ICODE  = IFIX (ABS (WILEY))
      DO 1001 J= 1,10
 1001 IWILEY(J) = 0
      GO TO  (1011,1012,1013,1020,1015,1016,1017,1018,1020,1020),ICODE
 1011 IWILEY(1) = 1
      GO TO  1020
 1012 IWILEY(2) = 1
      GO TO  1020
 1013 IWILEY(3) = 1
      GO TO  1020
 1015 IWILEY(1) = 1
      IWILEY(2) = 1
      GO TO  1020
 1016 IWILEY(1) = 1
      IWILEY(3) = 1
      GO TO  1020
 1017 IWILEY(2) = 1
      IWILEY(3) = 1
      GO TO  1020
 1018 DO 1014 J= 1,3
 1014 IWILEY(J) = 1
 1020 CONTINUE
      WRITE (6,921) MM
      GO TO 100
```

```
   11 CODE   = DA(1)
      JNO    = IFIX(SIGN(1.0,CODE))
      ICODE  = IFIX(ABS (CODE))
      DO     110 J= 1,10
  110 IOFT(J) = 0
      GO TO (111,112,113,114,115,116,117,118,119,1181), ICODE
  111 IOFT(1) = 1
      GO TO 120
  112 IOFT(2) = 1
      GO TO 120
  113 IOFT(3) = 1
      GO TO 120
  114 IOFT(4) = 1
      GO TO  119
  115 IOFT(1) = 1
      IOFT(2) = 1
      GO TO 120
  116 IOFT(1) = 1
      IOFT(3) = 1
      GO TO 120
  117 IOFT(2) = 1
      IOFT(3) = 1
      GO TO 120
  118 IOFT(1) = 1
      IOFT(2) = 1
      IOFT(3) = 1
      GO TO 120
 1181 IOFT(5) = 1
      GO TO 120
  119 IOFF    = 1
  120 SERNO  = DA(2)
      NPHASE = DA(3)
      VH     = DA(4)
      VV     = DA(5)
      WRITE (6,921) MM
      GO TO 100
   12 XDTP    = DA(1)
      TFC     = DA(2)
      HM      = DA(3)
      ERC(4) = DA(4)
      ERC(5) = DA(5)
      WRITE  (6,921) MM
      GO TO  100
   13 DO 131 J = 1,5
  131 ERC(J+5) = DA(J)
      WRITE  (6,921) MM
      GO TO  100
   14 DO      141 J=1,5
  141 ERC(J+10) = DA(J)
      WRITE  (6,921) MM
      GO TO  100
   15 DO 151 J = 1,5
  151 ERC(J+15) = DA(5)
      WRITE  (6,921) MM
      GO TO  100
   16 DO 161 J = 1,4
  161 ERC(J+20) = DA(J)
      RC(1) = DA(5)
```

```
      WRITE(6,921) MM
      GO TO  100
   17 DO 171 J=1,2
  171 RC(J+1) = DA(J)
      DO 172 J = 1,3
  172 XKL(J) = DA(J+2)
      WRITE(6,921) MM
      GO TO  100
   18 DO 181 J = 1,3
  181 FP(J)  = DA(J)
      DO 182 J = 1,2
  182 RR(J) = DA(J+3)
      WRITE  (6,921) MM
      GO TO  100
   19 RR(3)  = DA(1)
      DO 191 J = 1,3
  191 SRL(J) = DA(J+1)
      CB(1) = DA(5)
      WRITE  (6,921) MM
      GO TO  100
   20 DO 250 J= 1,2
  250 CB(J+1) = DA(J)
      DO 260 J = 1,3
  260 CF(J) = DA(J+2)
      WRITE  (6,921) MM
      GO TO  100
   21 XMLF    = DA(1)
      XMLB    = DA(2)
      DO 290 J = 1,3
  290 TAL(J) = DA(J+2)
      WRITE  (6,921) MM
      GO TO  100
  200 IF      (IOFF) 310,310,300
  300 DTF     = 0.0
      GO TO   320
  310 DTF     = XDTP
  320 WRITE  (6,922) SERNO, NPHASE
      MM=1
      WRITE (6,901) MM, XM, XIX, YIY, ZIZ, XIY
      MM=2
      WRITE (6,902) MM, XIZ, YIZ, XMP, XLDO, DELCG
      MM=3
      WRITE (6,903) MM, XLF, XLL, XLA, G   , ALPHAD(1)
      MM=4
      WRITE (6,904) MM, ALPHAD(2) , ALPHK(1), ALPHK(2), BETAD, XLC
      MM=5
      WRITE (6,905)  MM, HEO, XIALPH, XIGAM, HFO, DB
      MM=6
      WRITE  (6,906) MM, DF,     XDOTOC, YDOTOC, ZDOTOC
      MM=7
      WRITE  (6,907) MM, XNU, XLDMIN,XKS
      MM=8
      WRITE  (6,908) MM,           THETD(1), THETD(2), THETD(3)
      MM=9
      WRITE  (6,909) MM, THETSD, PSIOD, PHIOD, XIOD, XLAMDD
      MM=10
      WRITE  (6,910) MM, (J,OMEGA(J) , J=1,3) ,WILEY
      MM=11
```

```
      WRITE  (6,911) MM, CODE, SEENO, NPHASE, VH, VV
      MM     = 12
      WRITE  (6,912) MM, LTP, TFO,HM, (ERO(J), J= 4,5)
      MM     = 13
      WRITE  (6,913) MM,(J,ERO(J) , J= 6,10)
      MM     = 14
      WRITE  (6,914) MM, (J,ERO(J), J = 11,15)
      MM     = 15
      WRITE  (6,915) MM,(J,ERO(J), J= 16, 20)
      MM     = 16
      WRITE  (6,916) MM,(J,ERO(J), J= 21,24), RC(1)
      MM     = 17
      WRITE  (6,917) MM, (J,RC(J), J=2,3), (J,XKD(J), J= 1,3)
      MM     = 18
      WRITE  (6,918) MM,(J,FP(J), J= 1,3), (J,RR(J), J= 1,2)
      MM     = 19
      WRITE  (6,919) MM, RR(3) , (J,SRL(J), J= 1,3) , CB(1)
      MM     = 20
      WRITE  (6,920) MM,(J,CB(J), J=2,3), (J,CF(J), J= 1,3)
      MM     = 21
      WRITE  (6,1921) MM, XMUF,XMUB, (J,TAU(J), J= 1,3)
      PI = 3.14159265
      DEGRAD= PI / 180.0
      RADDEG= 180.0 / PI
C     CHANGE INPUT ANGLES FROM DEGREES TO RADIANS
      THETA(1) = DEGRAD * THETD(1)
      THETA(2) = DEGRAD * THETD(2)
      THETA(3) = DEGRAD * THETD(3)
      THETAS   = DEGRAD * THETSD
      PSIO     = DEGRAD * PSIOD
      PHIO     = DEGRAD * PHIOD
      XID      = DEGRAD * XICD
      ALPHA(1) = DEGRAD * ALPHAD(1)
      ALPHA(2) = DEGRAD * ALPHAD(2)
      BETA     = DEGRAD * BETAD
      XLAMDA   = DEGRAD * XLAMDD
      OMEG(1) = OMEGA(1)*DEGRAD
      OMEG(2) = OMEGA(2)*DEGRAD
      OMEG(3) = OMEGA(3)*DEGRAD
      XMFP(1) = XMP
      XMFP(2) = XMP
      XMFP(3) = XMP
C
C     SET INERTIA MATRIX   XJ(3,3)
C
      XJ(1,1) =  XIX
      XJ(1,2) = -XIY
      XJ(1,3) = -XIZ
      XJ(2,1) = -XIY
      XJ(2,2) =  YIY
      XJ(2,3) = -YIZ
      XJ(3,1) = -XIZ
      XJ(3,2) = -YIZ
      XJ(3,3) =  ZIZ
      DO     850 J = 4,24
  850 ERF(J) = ERO(J)
      DO     851 J = 1,10
  851 IOPT(J) = IABS (IOPT(J))
```

```
      TF      = TFO
      JF      = JNC
      IF      (IWILEY(1) + IWILFY(2) + IWILEY(3)) 1102,1102,1101
 1101 JN      = 0
 1102 CONTINUE
  900 FORMAT(I7,5X,5E12.5)
  901 FORMAT(5H CARD,I3,10H      XM  =,E12.5,10H     IXX  =,E12.5,10H     I
     1YY  =,E12.5,10H     IZZ  =,F12.5,10H     IXY  =,E12.5/)
  902 FORMAT(5H CARD,I3,10H      IXZ  =,E12.5,10H     IYZ  =,E12.5,10H     X
     1MP  =,E12.5,10H    XLDO  =,F12.5,10H  DFLCG  =,F12.5/)
  903 FORMAT(5H CARD,I3,10H      XLF  =,E12.5,10H     XLL  =,F12.5,10H     X
     1LA  =,E12.5,10H        G =,F12.5,10H ALPHA(1)=,F12.5/)
  904 FORMAT(5H CARD,I3,10H ALPHA(2)=,E12.5,10H ALPHK(1)=,E12.5,10H ALPH
     1K(2)=,E12.5,10H    BETA  =,F12.5,10H     XLC  =,E12.5/)
  905 FORMAT(5H CARD,I3,10H      HFO  =,E12.5,10H  XIALPH =,E12.5,10H    XI
     1GAM =,E12.5,10H      HFO =,F12.5,10H      DB =,E12.5/)
  906 FORMAT (1X4HCARDI3,6X4HDF = E12.5,2X8HXDOTOC = E12.5, 2X8HYDOTOC =
 9061  E12.5, 2X8HZDOTOC = F12.5 /)
  907 FORMAT (1X4HCARD I3,5X5HXNU = E12.5,2X8HXLDMIN = E12.5,5X5HXKS =
 9071 E12.5 /)
  908 FORMAT(5H CARD,I3,                                          10H THET
     1D(1)=,E12.5,10H THETD(2)=,E12.5,10H THETD(3)=,E12.5/)
  909 FORMAT(5H CARD,I3,10H  THETAS =,E12.5,10H    PSIO  =,E12.5,10H     P
     1HIC =,E12.5,10H      XIO =,F12.5,10H  XLAMDD =,E12.5/)
  910 FORMAT (1X4HCARD I3, 3(1X6HOMEGA(I1,2H)=E12.5),3X7HCODE2 = E12.5/)
  911 FORMAT (5H CARD,I3,10H     CODE =,F12.5,10H  SER NO =,F12.5,10H PHA
 9111SE NO=,I12   ,10H       VH =,E12.5,10H       VV =,E12.5/)
  912 FORMAT (1X4HCARDI3,5X5HDTF = E12.5,6X4HTF = E12.5,6X4HHM = E12.5,
 9121 2X8HERR(4) = E12.5, 2X8HERR(5) = E12.5 / )
  913 FORMAT (1X4HCARDI3, 4(2X4HERR(I1,3H) = E12.5), 1X4HERR(I2,3H) = E1
 9131 2.5/)
  914 FORMAT (1X4HCARDI3, 5(1X4HERR(I2,3H) = E12.5)/)
  915 FORMAT (1X4HCARDI3, 5(1X4HERR(I2,3H) = E12.5)/)
  916 FORMAT (1X4HCARDI3,4(1X4HERR(I2,3H) = E12.5), 3X7HRC(1) = E12.5/)
  917 FORMAT (1X4HCARDI3, 2(3X3HRC(I1,3H) = F12.5), 3(2X4HXKP(I1,3H) = E
 9171 12.5)/)
  918 FORMAT (1X4HCARDI3, 3(3X3HFP(I1,3H) = F12.5),2(3X3HRR(I1,3H) = E12
 9181 .5)/)
  919 FORMAT (1X4HCARDI3,3X7HRP(3) = E12.5,3(2X4HSRL(I1,3H) = E12.5), 3X
 9191 7HCB(1) = E12.5 / )
  920 FORMAT (1X4HCARDI3, 2(3X3HCB(I1,3H) = F12.5),3(3X3HCF(I1,3H) = E12
 9201 .5)/)
 1921 FORMAT (1X4HCARDI3,4X6HXMUF = E12.5,4X6HXMUB = E12.5, 3(2X4HTAU(I1
19211 , 3H) = E12.5)/)
  921 FORMAT(39H   NEW DATA CARD FOR THIS CASE IS NO. =,I5)
  922 FORMAT (//27H  THIS CASE IS SERIES NO. = F12.3,
 9221         20H      AND PHASE NO. = , I4// 21H  INPUT DATA FOLLOWS. //)
      RETURN
      END
```

```
      SUBROUTINE INIT
      DIMENSION XMPP(3), ALPHK(2)
      DIMENSION A(3), AA(3)
      COMMON/N7/ ALPDOT(3), ALFH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N10/FOOT1(100), FOOT2(100), FOOT3(100),
  101               BLOCK1(100), BLOCK2(100), BLOCK3(100),
  102               SHOCK1(100), SHOCK2(100), SHOCK3(100)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMPP,XIP,DELCG,XLP,
     1          XLL,XLA,G,ALPHK,XLC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2          XLDOP,XLDO,XKD(3),XKS,    XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3          XMUB,RUNNO,SERNO,VV,VH,DB,XIALPH,XIGAM,DELMAX,SRL(3),FP(3)
      COMMON/N12/XDOTOC,YDOTOC,ZDOTOC,TAU(3),ISET(3),XA(3),YA(3),ZA(3)
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),PSB(3),XHPU(3),
     1          YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELEL(3),FIMAX(3),
     2          ZFREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ B(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DFS(3,3,3)
      COMMON/N19/NNFLAG(3),PSBO(3),PSPQ(3,12),IBEEN(6)
  100 SINTHS = SIN ( THETAS)
      COSTHS = COS ( THETAS)
      DO 105 J = 1,3
      ISET(J) = 0
      XA(J)  = 0.0
      YA(J) = 0.0
      ZA(J)  = 0.0
      DELEL(J) = FP(J)/XKS
      FIMAX(J) = 0.0
      IBEEN(J) = 0
      IBEEN(J+3) = 0
  105 CONTINUE
      DELMAX =  XLDO - XLDMIN
      K       = IFIX(FOOT1(2))+3
      SLOPE1(1) = (FOOT1(K+1)-FOOT1(K))/(FOOT1(4)-FOOT1(3))
      K       = IFIX(FOOT2(2))+3
      SLOPE1(2) = (FOOT2(K+1)-FOOT2(K))/(FOOT2(4)-FOOT2(3))
      K       = IFIX(FOOT3(2))+3
      SLOPE1(3) = (FOOT3(K+1)-FOOT3(K))/(FOOT3(4)-FOOT3(3))
      K       = IFIX(BLOCK1(2))+3
      SLOPE2(1) = (BLOCK1(K+1)-BLOCK1(K))/(BLOCK1(4)-BLOCK1(3))
      K       = IFIX(BLOCK2(2))+3
      SLOPE2(2) = (BLOCK2(K+1)-BLOCK2(K))/(BLOCK2(4)-BLOCK2(3))
      K       = IFIX(BLOCK3(2))+3
      SLOPE2(3) = (BLOCK3(K+1)-BLOCK3(K))/(BLOCK3(4)-BLOCK3(3))
  110 XC= 0.0
      YC= 0.0
      ZC= 0.0
      PSI = PSIO
      PHI = PHIO
      XI  = XIO
      D1      = COS(PSI)
```

```
      D2 = COS ( PHI)
      D3 = COS ( XI )
      D4 = SIN ( PSI)
      D5 = SIN ( PHI)
      D6 = SIN ( XI )
      B(1,1) = D2 * D3
      B(1,2) = -D2 *D6
      B(1,3) = D5
      B(2,1) = D4 *D5 *D3 + D1 *D6
      B(2,2) = -D4 *D5 *D6 +D1 *D3
      B(2,3) = -D4 *D2
      B(3,1) = -D1 *D5 *D3 + D4 * D6
      B(3,2) = D1 *D5 *D6 +D4 *D3
      B(3,3) = D1 *D2
      GO TO  (1102,1101,1103), NPHASE
 1101 XDOTC =-VH * SIN (XLAMDA)
      YDOTC = VH * COS (XLAMDA)
      ZDOTC = VV
      GO TO  111
 1102 XDOTC= XDOTOC
      YDOTC= YDOTOC
      ZDOTC= ZDOTOC
      GO TO  111
 1103 A(1)    = XDOTOC
      A(2)    = YDOTOC
      A(3)    = ZDOTOC
      DO      1104 I = 1,3
      AA(I)   = 0.0
      DO      1104 L = 1,3
 1104 AA(I)   = AA(I) + B(I,L) * A(L)
      XDOTC   = AA(1)
      YDOTC   = AA(2)
      ZDOTC   = AA(3)
  111 PFLAG   = +1.0
C     ABOVE P FLAG IS USED IN BOX 1815 OF FORCE SUBROUTINE.
C
C     NOW CALCULATE  T SUB ZERO
C
      DO 112  I =1,3
      AA(I) = 0.0
      DO 112  L =1,3
      AA(I) = AA(I) +  XJ(I,L) * OMEG(L)
  112 CONTINUE
      D1 = 0.0
      DO 114  L =1,3
      D1 = D1 + OMEG(L)*AA(L)
  114 CONTINUE
      TZERO= .5*( XM * ( XDOTC * XDOTC + YDOTC * YDOTC + ZDOTC * ZDOTC)
     1 + D1 )
C
C     THE FOLLOWING CHANGES ( TO COMMENTS CARD MARKED END OF CHANGES )
C     ARE CHANGES MADE IN DECK NO. 2 ONLY.
C
      XCDP (1) = 0.0
      XCDP (2) = 0.0
      YCDP(1) = ( 2.0 / 3.0 ) * DF / PI
      YCDP(2)  = - YCDP(1)
C
```

```
C      END OF DECK NO. 2 CHANGES
C
       D19 = XLA * XLA + XLL * XLL
       D20 = 2.0 * XLA *XLL
       GAMMAO = BETA + ASIN ((XLDO*XLDO - D19) / D20)
       ZCFMAX = 0.0
       D17 = SIN ( BETA )
       D18 = COS ( BETA )
       DO 240 J =1,3
   140 GAMMA(J) = GAMMAO
       ALPH(J)  = GAMMAO
       SINTH(J) = SIN ( THETA(J) )
       COSTH(J) = COS ( THETA(J) )
       PSI(J) = 0.0
       PSEQ(J)   = 0.0
       XB(J) = XLC * SINTH(J)
       YB(J) = XLC * COSTH(J)
       ZBC(J) = DELCG + HPC
       XHFU(J) = SINTH(J) *( XLF + XLA *  D17 )
       YHFU(J) = COSTH(J) *( XLF + XLA *  D17 )
       ZHFU(J) = DELCG - XLA *  D18
       GAMDOT(J) = 0.0
       ALFDOT(J) = 0.0
       XFP(J) = SINTH(J) *( XLP + XLL * COS ( GAMMA(J) ))
       YFP(J) = COSTH(J) *( XLP + XLL * COS ( GAMMA(J) ))
       ZFP(J) = DELCG + XLL * SIN ( GAMMA(J) )
       HPL  (J,1) = XLP * SINTH (J)
       HPL  (J,2) = XLP * COSTH (J)
       HPL  (J,3) = DELCG
   150 DRS(J,1,1) = COSTH(J)
       DRS(J,1,2) = SINTH(J)
       DRS(J,1,3) = 0.0
       DRS(J,2,1) = - SINTH(J)
       DRS(J,2,2) = COSTH(J)
       DRS(J,2,3) = 0.0
       DRS(J,3,1) =  0.0
       DRS(J,3,2) =  0.0
       DRS(J,3,3) =  1.0
       A6(1) = XFP(J)
       A6(2) = YFP(J)
       A6(3) = ZFP(J)
C
C     K  DO LOOPS HERE AND AT  200 ARE CHANGED IN DECK NO. 2 .
C
       DO 190 K = 1,2
   160 AA(1) = XCDP(K)
       AA(2) = YCDP(K)
       AA(3) = HFO
       DO 162 L1 =1,3
       A(L1) = A6(L1)
       DO 162 L2 =1,3
       A(L1) = A(L1) + DRS(J,L1,L2) * AA(L2)
   162 CONTINUE
       DO 164 L1 =1,3
       A1(L1) = 0.0
       DO 164 L2 =1,3
       A1(L1) = A1(L1) + B(L1,L2) * A(L2)
   164 CONTINUE
```

```
      A7 = SINTHS * A1(2) + COSTHS * A1(3)
  170 IF( A7 - ZCPMAX)190,190,160
  180 ZCPMAX = A7
  190 CONTINUE
  200 DO 220 K = 1,2
  210 FSF(J,K) = 0.0
      FSFQ(J,K) = 0.0
  220 CONTINUE
  240 CONTINUE
  250 ZC=  -ZCPMAX / COSTHS
      ZPREF  = - ZFP(3) - HFO - 0.50
      VZERO = - XM * G * ZC
      DA(7) = XM * G
      DA(8) = PI * DF / 8.0 * DF
      DA(9) =  65.0 * DEGRAD
      DA(10) = PI * DB * DB / 4.0
      RETURN
      END
```

```
      SUBROUTINE FLP
      DIMENSION XMFP(3), ALPHK(2),SHKMAX(3)
      DIMENSION A(3), AA(3)
      COMMON/N2/T,Y(50),D(50)
      COMMON/N5/ TALPHA(3), TGAMMA(3), F(3), TG(3)
      COMMON/N6/OMEGDT(3), ALFHID(3), GAMDL(3), G(3), PSID, PHID, XID,
     1 DELWRK
      COMMON/N7/ ALFDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N8/ LFLAG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, FA(10), P(3,4,6), HPI(3,3)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMFP,XIP,DELCG,XLP,
     1        XLL,XLA,G,ALPHK,XIC,RHO,NPHASE,HFO,HFMIN,DF,HC(3),RR(3),
     2        XLDOP,XLDO,XKD(3),XKS,   XMU,XLFMIN,CC,CF(3),XLAMDA,XMUF,
     3        XMUB,RUNNO,SERNO,VV,VH,DB,XTALPH,XIGAM,DELMAX,SPL(3),FP(3)
      COMMON/N13/ A, AA
      COMMON/N14/SLOPL1(3),SLOPE2(3),XCDP(12),YCDP(12),FSB(3),XHPU(3),
     1        YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELFL(3),F1MAX(3),
     2        ZFREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ E(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DFS(3,3,3)
      COMMON/N16/ RN(3), XII(3,2), XLD(3), XLDDOT(3), XPCTP(3,2),
     1 YDOTF(3,2), S(3,2), COSXI  (3), HF(3,12), ZDOTF(3,2), HB(3),
     2 GFS(3,3,3), TA
      COMMON/N17/ XFPC(3), YFPC(3) , ZFPC(3)
      COMMON/N18/ NFLAG
      COMMON/N19/NNFLAG(3),PSEC(3),FSPC(3,12),IBEEN(6)
      COMMON/N20/DWORK(15)
      COMMON/N24/ICPT(10), JN, LTF
      IF      (LFLAG) 19,19,10
   10 IF      (LFLAG-1) 70,70,15
   15 ISTALL = 0
      GO TO  70
   19 IF      (NFLAG) 20,20,24
   20 DO      23 I= 4,24
      IF      (ABS(Y(I)) - 10.E4)23,21,21
   21 DO      22 J= 1,24
   22 D(J)    = 0.0
      D(22)   = 10.E6
      GO TO  150
   23 CONTINUE
   24 DO      30 I= 1,3
      ALFDOT(I) = Y(I)
      ALPH(I)   = Y(I+3)
      GAMDOT(I)= Y(I+6)
      GAMMA(I) = Y(I+9)
   30 OMEG(I) = Y(I+18)
      XDOTC   = Y(13)
      YDOTC   = Y(14)
      ZDOTC   = Y(15)
      XC      = Y(16)
      YC      = Y(17)
```

```
      ZC       = Y(16)
      PSI      = Y(22)
      PHI      = Y(23)
      XI       = Y(24)
      TA       = T
      CALL     CONFIG
      IF (NFLAG  ) 130,130,36
   36 CALL     STAB
      DO 36 J=1,3
      IF       (IBEEN(J)) 37,37,38
   37 IF       (FSP(J,1) + FSP(J,2) ) 384,38,384
   38 CONTINUE
  381 DO       383 J=1,3
      IF       (IBEEN(J+3)) 382,382,383
  382 IF       (PSB(J)) 386,383,386
  383 CONTINUE
      GO TO  388
  384 WRITE (6,385) J,T
  385 FORMAT (10X12HFOOTPAD NO. I2,40H HAS INITIALLY CONTACTED SURFACE A
 3851T T = F10.4, 9H SECONDS./)
      IBEEN(J) = 1
      GO TO  36
  386 WRITE (6,387) J,T
  387 FORMAT (10X15HBODY BLOCK NO. I2, 40H HAS INITIALLY CONTACTED SURFA
 3871CE AT T = F10.4,9H SECONDS. / )
      IBEEN(J+3) = 1
      GO TO  383
  388 IF       (KFLAG) 40,50,60
   40 WRITE (6,45)
   45 FORMAT (45H*****  VEHICLE IS UNSTABLE IN PITCH  *****      )
      ISTALL = -1
      GO TO 70
   50 WRITE (6,55)
   55 FORMAT (45H***** VEHICLE IS UNSTABLE IN YAW  *****         )
      ISTALL = -1
   70 LFLAG  = 1
      SIGMIN = SIGMIN*RADDEG
      WRITE  (6,71) T, SIGMIN
   71 FORMAT (10X7HTIME =  F10.4, 10X 9HSIGMIN = F10.4,8H DEGREES/)
      D59 = 0.0
      D69 = 0.0
      WORK     = 0.0
      K        = 5 + IABS (ICPT(3)) * 10
      DO       72 J= 1,K
   72 WORK     = WORK + Y(J+24)
      DO       76 K=1,3
      A(K)     = 0.0
      DO       74 J=1,3
   74 A(K)     = A(K)+ OMEG   (J)*XJ(K,J)
      D59      = A(K)*OMEG   (K)+D59
   76 D69      = GAMDOT(K)*GAMDOT(K)+D69
      EF       = D59 /2.+D69*XIGAM/2.          +XM/2.*(XDOTC*XDOTC
     1         +YDOTC * YDOTC + ZDOTC * ZDOTC)
      EPD      = VZERO+DA(7)*ZC
      WRITE (6,78) TZERO,EPD,WORK,EF,(SHKMAX(J),J=1,3),GCMIN
   78 FORMAT (
 78110X40HINITIAL KINETIC ENERGY (EKO)           = F20.5,7H FT LBS/
 78210X40HDIFFERENCE IN POTENTIAL ENERGY (EPD)   = F20.5,7H FT LBS/
```

```
76310)4CHDISSIPATED ENERGY (EDIS)                    = F20.5,7H FT LBS/
76410)40HFINAL KINETIC ENERGY (EKF)                  = F20.5,7H FT LBS//
765100)30HMAXIMUM SHOCK ABSORBER FORCES- /
78620)17HLEG NO.1 FA(1) =   F20.10/
78720)17HLEG NO.2 FA(2) =   F20.10/
76820)17HLEG NO.3 FA(3) =   F20.10//
76910)51HMINIMUM CLEARANCE BETWEEN GROUND PLANE AND FRAME = F10.4, 4H
761 FT.  /)
    RETURN
 60 IF       (IOPT(4)) 605,605,130
605 IF       (T) 61,61,62
 61 SIGMIN = 10.E8
    GCMIN  = 10.E8
    DO       611 J=1,3
611 SHKMAX(J) = 0.0
 62 SIGMIN =AMIN1 (SIGMIN, W(3,4,2))
    GCMIN  = AMIN1(GCMIN,W(1,4,2))
    DO       621 J=1,3
621 SHKMAX(J) = AMAX1(SHKMAX(J), W(J,4,1))
 63 IF       (SIGMIN - 0.7654 ) 65,68,68
 65 IF       (W(3,4,2) - SIGMIN - 0.874) 66,68,66
 66 WRITE    (6,67)
 67 FORMAT (42H$$$$$ VEHICLE IS STABLE (ROCK-BACK) $$$$$ /)
    ISTALL = 1
    GO TO  70
 68 IF       (ZFREF-(YC*SINTHS+ZC*COSTHS))  75,75,130
 75 IF       (ZDOTC*ZDOTC-4.0) 80,80,130
 80 IF       (YDOTC*YDOTC- 4.0) 90,90,130
 90 IF       (XDOTC*XDOTC-4.0) 100,100,130
100 IF       (OMEG(1)*OMEG(1) - .01) 110,110,130
110 IF       (OMEG(2)* OMEG(2) - .01)120,120,130
120 IF       (OMEG(3)*OMEG(3) - .01 ) 125,125,130
125 WRITE (6,126)
126 FORMAT (45H***** VEHICLE IS STABLE *****                     )
    ISTALL = 1
    GO TO  70
130 CALL FORCE
    IF       (PFLAG) 131,132,132
131 ISTALL = -1
    GO TO  70
132 CALL    INTEQM
    DO      140 I=1,3
    IF    (MFLAG) 136,136,134
134 D(I)    = ALPHDD(I)
136 D(I+3) = ALPDOT(I)
    D(I+3) = AMAX1 (-100.0,( AMIN1 (ALPDOT(I), 100.0)))
    D(I+6) = GAM DD(I)
    D(I+9) = GAMDOT(I)
    D(I+12) = Q(I)
    D(I+15)= Y(I+12)
140 D(I+18)= OMEGDT(I)
    D(22)   = PSID
    D(23)   = PHID
    D(24)   = XID
    DO      145 J=1,15
145 D(J+24) = DWORK(J)
150 RETURN
    END
```

```
      SUBROUTINE CONFIG
      DIMENSION XMFP(3), ALPHK(2)
      DIMENSION A(3) , AA(3)
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, FFLAG, VFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N8/LFLAG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), V(3,4,6), HPL(3,3)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMFP,XIP,DELCG,XLP,
     1        XLL,XLA,G,ALPHK,XIC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2        XLDOP,XLDO,XKL(3),XKS,    XNU,XLFMIN,CC,CF(3),XLAMDA,XMUF,
     3        XMUB,RUNNO,SERNO,VV,VH,DB,XTALPH,XIGAM,DELMAX,SPL(3),FP(3)
      COMMON/N12/XDOTOC,YDOTOC,ZDOTOC,TAU(3),ISET(3),XA(3),YA(3),ZA(3)
      COMMON/N13/ A, AA
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),PSB(3),XHPU(3),
     1        YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELFL(3),FIMAX(3),
     2        ZPREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ E(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DFS(3,3,3)
      COMMON/N16/ RN(3), XII(3,2), XLD(3), XLDDOT(3), XDOTP(3,2),
     1 YDOTP(3,2), S(3,2), COSXI (3), HF(3,12), ZDOTP(3,2), HB(3),
     2 GFS(3,3,3), TA
      COMMON/N17/ XFPC(3), YFPC(3) , ZFPC(3)
      COMMON/N18/NFLAG
      COMMON/N19/NNFLAC(3),PSBC(3),PSPQ(3,12),IBEEN(6)
      COMMON/N22/FOCT(3,3)
      COMMON/N24/ICPT(10), JN, DTF
      IF      (PSI*PSI+PHI*PHI+XI*XI-10000) 110,109,109
  109 PSI     = PI/2.0
      PHI     = 0.0
      XI      = 0.0
  110 D11=  COS (PSI)
      D12=  COS (PHI)
      D13=  COS (XI)
      D14=  SIN (PSI)
      D15=  SIN (PHI)
      D16=  SIN (XI)
      B(1,1) = D12*D13
      B(1,2) =-D12*D16
      B(1,3) = D15
      B(2,1) = D14*D15*D13+D11*D16
      B(2,2) =-D14*D15*D16+D11*D13
      B(2,3) =-D14*D12
      B(3,1) =-D11*D15*D13+D14*D16
      B(3,2) = D11*D15*D16+D14*D13
      B(3,3) = D11*D12
C
C     D17,D18,D19,D20 ARE SET IN INIT TO SIN(BETA),COS(BETA),
C     XLA*XLA P XLL*XLL ,  2.0*XLA*XLL RESPECTIVELY
C
  120 A(1) =   0.0
      A(2) =   SINTHS
      A(3) =   COSTHS
```

```
      DO 124  L =1,3
      RN(L) = 0.0
      DO 124 L2 =1,3
      RN(L) = RN(L) + E(L2,L)   * A(L2)
  124 CONTINUE
      COSXIB = B(2,3) * SINTHS + B(3,3) * COSTHS
      XII(1,2) =  ACOS ( COSXIB )
      XII(2,2) = XII(1,2)
      XII(3,2) = XII(1,2)
  130 IF( COSXIB * COSXIB   - .000001 )140,150,150
  140 COSXIB =  .001
  150 DO 240 J = 1,3
  160 D7 = SIN ( GAMMA(J))
      D8 = COS ( GAMMA(J))
      XFP(J) =  SINTH(J) *( XLF + XLL * D8)
      YFP(J) =  COSTH(J) *( XLF + XLL * D8)
      ZFP(J) =  DELCG + XLL * D7
      XLD(J) = SQRT ( D19 + D20 * ( D7 * D18 - D8 * D17 ) )
      XLDDOT(J) =    XLA * XLL * GAMDOT(J) * (D8*D18 + D7 *D17) /XLD(J)
 1604 AA(1) = XFP(J)
      AA(2) = YFP(J)
      AA(3) = ZFP(J)
      DO 162 I=1,3
      A6(I) = 0.0
      DO 162 L =1,3
      A6(I) = A6(I) + E(I,L) * AA(L)
  162 CONTINUE
        XFPC(J) = A6(1)
        YFPC(J) = A6(2)
        ZFPC(J) = A6(3)
  170 A1(1) = -XLL * GAMDOT(J) * D7 * SINTH(J)
      A1(2) = -XLL * GAMDOT(J) * D7 * COSTH(J)
      A1(3) =  XLL * GAMDOT(J) * D8
      AA(1) = A1(1) + OMEG(2)* ZFP(J) - OMEG(3)* YFP(J)
      AA(2) = A1(2) + OMEG(3)* XFP(J) - OMEG(1)* ZFP(J)
      AA(3) = A1(3) + OMEG(1)* YFP(J) - OMEG(2)* XFP(J)
      DO 172 I =1,3
      A(I) = 0.0
      DO 172 L =1,3
      A(I) = A(I) + B(I,L) * AA(L)
  172 CONTINUE
      XDOTP(J,1) = XDOTC + A(1)
      YDOTP(J,1) = COSTHS *( YDOTC + A(2)) -SINTHS *( ZDOTC + A(3))
      ZDOTP(J,1) =SINTHS*(YDOTC+A(2))+COSTHS*(ZDOTC+A(3))
      S(J,1) = SQRT ( XDOTP(J,1) * XDOTP(J,1) + YDOTP(J,1) *YDOTP(J,1))
  180 D1 = SIN ( ALPH(J) - GAMMA(J))
      D2 = COS ( ALPH(J) - GAMMA(J) )
      A6(1) = D1 * SINTH(J)
      A6(2) = D1 * COSTH(J)
      A6(3) =  D2
      COSXI(J) = RN(1) * A6(1) + RN(2) *A6(2) + RN(3) * A6(3)
      XII(J,1) = ACOS ( COSXI(J))
      DRS(J,1,1) = COSTH(J)
      DRS(J,1,2) = SINTH(J) *D2
      DRS(J,1,3) = SINTH(J) *D1
      DRS(J,2,1) =-SINTH(J)
      DRS(J,2,2) = COSTH(J) *D2
      DRS(J,2,3) = COSTH(J) *D1
```

```
      DRS(J,3,1) = 0.0
      DRS(J,3,2) =-D1
      DRS(J,3,3) = D2
      IF( XII(J,1) - DA(9) )200,200,220
  200 A(3) = HFO
      DO 216 K = 1,2
  201 A(1) = XCDP(K)
      A(2) = YCDP(K)
      DO 202 L =1,3
      AA(L) = 0.0
      DO 202 I =1,3
      AA(L) = AA(L) + DRS(J,L,I) * A(I)
  202 CONTINUE
      A1(1) = XFP(J) + AA(1)
      A1(2) = YFP(J) + AA(2)
      A1(3) = ZFP(J) + AA(3)
      D1      = 0.0
      D2      = 0.0
      D3      = 0.0
      DO 204 L=1,3
      D1      = D1 + B(1,L) * A1(L)
      D2      = D2 + B(2,L) * A1(L)
      D3      = D3 + B(3,L) * A1(L)
  204 CONTINUE
      IF      (IOPT(4)) 2048,2048,2041
 2041 IF      (TA-TAU(J)) 206,2042,2042
 2042 IF      (ISET(J)) 2043,2043,2045
 2043 IF      (NFLAG) 206,206,2044
 2044 ISUM    = 0
      DO      3001 L= 1,3
 3001 ISUM    = ISUM + ISET(L)
      IF      (ISUM) 3002,3002,3005
 3002 PSIO    = PSI
      PHIO    = PHI
      XIO     = XI
      IF      (NPHASE - 3 ) 3004,3003,3004
 3003 VHORIZ = SQRT ( XDOTC*XDOTC + YDOTC*YDOTC)
      XDOTCG = XDOTC
      YDOTCG = YDOTC
 3004 VVERT   = ZDOTC
 3005 ISET(J) = 1
      SUMY    = 0.0
      SUMZ    = 0.0
      XA(J)   = D1 + XC
      YA(J)   = D2 + YC
      ZA(J)   = D3 + ZC
      GO TO   206
 2045 SUMY    = D2 + YC - YA(J)
      SUMZ    = D3 + ZC -ZA(J)
      A7      = SINTHS * SUMY + COSTHS * SUMZ
      GO TO   2049
 2048 FOCT(J,1) = XFPC(J) + XC
      FOCT(J,2) = (YFPC(J) + YC)*COSTHS-(ZFPC(J) + ZC)*SINTHS
      FOCT(J,3) = (YFPC(J) + YC) * SINTHS + (ZFPC(J) + ZC) * COSTHS
      A7        = (D2+YC)*SINTHS+(D3+ZC)*COSTHS
 2049 HF(J,K) = HFO - A7 / COSXI(J)
  205 IF( PSP(J,K) + HF(J,K) - HFO)218,218,206
  206 HF(J,K) = HFO - PSP(J,K)
```

```
  216 CONTINUE
      IF      (IOPT(4)) 220,220,219
  219 NNFLAG(J) = 0
      GO TO  230
  220 A6(1) = XB(J)
      A6(2) = YB(J)
      A6(3) = ZBO(J)
      D3 = 0.0
      D4 = 0.0
      DO 221 L=1,3
      D3 = D3 +  B(2,L) * A6(L)
      D4 = D4 +  B(3,L) * A6(L)
  221 CONTINUE
      A7        =SINTHS *( D3 + YC) + COSTHS *( D4 + ZC)
      HB(J) = HEO-A7/COSXIE
C
      NNFLAG(J) = +1
      IF   (NFLAG) 2220,222,222
 2220 FSE(J) = FSBG(J)
  222 IF( FSE(J) + HB(J) - HEO)224,224,223
  223 NNFLAG(J) = 0
      GO TO 230
  224 ZB(J) = DELCG + FB(J)
      AA(1) = OMEG(2)* ZB(J)- OMEG(3)* YB(J)
      AA(2) = OMEG(3)* XB(J)- OMEG(1)* ZB(J)
      AA(3) = OMEG(1)* YB(J)- OMEG(2)* XB(J)
      DO 229 L=1,3
      A(L) = 0.0
      DO 229 I =1,3
      A(L) = A(L) + B(L,I) * AA(I)
  229 CONTINUE
      XDOTP(J,2) = XDOTC + A(1)
      YDOTP(J,2) = COSTHS *( YDOTC +A(2)) - SINTHS *( ZDOTC + A(3))
      ZDOTP(J,2) =SINTHS*(YDOTC+A(2))+COSTHS*(ZDOTC+A(3))
      S(J,2) =  SQRT ( XDOTP(J,2) * XDOTP(J,2) + YDOTP(J,2) * YDOTP(J,
     1 2))
  230 GRS(J,1,1) = COSTH(J)
      GRS(J,1,2) = SINTH(J) * D8
      GRS(J,1,3) =-SINTH(J) * D7
      GRS(J,2,1) =-SINTH(J)
      GRS(J,2,2) = COSTH(J) * D8
      GRS(J,2,3) =-COSTH(J) * D7
      GRS(J,3,1) = 0.0
      GRS(J,3,2) = D7
      GRS(J,3,3) = D8
      DO 232 L= 2,3
      A(L) = 0.0
      DO 232 I= 1,3
      A(L) = A(L) + B(L,I)*HFL(J,1)
  232 CONTINUE
      AB(J,1)= SINTHS *(YC+A(2))+COSTHS*(ZC+A(3))
  240 CONTINUE
      IF      (IOPT(4)) 250,250,2401
 2401 IF      (ISET(1)*ISET(2)*ISET(3)) 250,250,2499
C
C     THIS SECTION COMPUTES GROUND SLOPE AND CROSS SLOPE ANGLE IF
C     FOOTPAD IMPACT TIMES ARE SPECIFIED  AND OPTION 4 IS ENABLED
C
```

```
2499 AX     = XA(2) - XA(1)
     AY     = YA(2) - YA(1)
     AZ     = ZA(2) - ZA(1)
     BX     = XA(3) - XA(1)
     BY     = YA(3) - YA(1)
     BZ     = ZA(3) - ZA(1)
     AXBX   = AY*BZ - AZ*BY
     AXBY   = AZ*BX - AX*BZ
     AXBZ   = AX*BY - AY*BX
     AXBMAG = SQRT (AXBX*AXBX + AXBY*AXBY + AXBZ*AXBZ)
     WRITE  (6,2498) TA
2498 FORMAT (/8X5HTA = F7.4,6H SECONDS )
     IF     (AXBMAG) 2402,2402,2410
2402 WRITE  (6,2403)
2403 FORMAT(/32H*****IMPACT POINTS COLINEAR*****  //)
     WRITE  (6,2404) XA(1),YA(1),ZA(1)
2404 FORMAT (5X8HXA(1) = F7.2,5X8HYA(1) = F7.2,5X8HZA(1) = F7.2  /)
     WRITE  (6,2405) XA(2), YA(2), ZA(2)
2405 FORMAT (5X8HXA(2) = F7.2,5X8HYA(2) = F7.2,5X8HZA(2) = F7.2  /)
     WRITE  (6,2406) XA(3),YA(3),ZA(3)
2406 FORMAT (5X8HXA(3) = F7.2,5X8HYA(3) = F7.2,5X8HZA(3) = F7.2  /)
     LFLAG  = 1
     GO TO  250
2410 UNX    = AXBX/AXBMAG
     UNY    = AXBY/AXBMAG
     UNZ    = AXBZ/AXBMAG
     IF     (UNZ) 2411,2412,2415
2411 UNX    = -UNX
     UNY    = -UNY
     UNZ    = -UNZ
     GO TO  2415
2412 WRITE  (6,2413)
2413 FORMAT(/24H*****LANDED ON WALL*****    //)
2414 WRITE  (6,2404) XA(1),YA(1), ZA(1)
     WRITE  (6,2405) XA(2),YA(2), ZA(2)
     WRITE  (6,2406) XA(3),YA(3),ZA(3)
     LFLAG  = 1
     GO TO  250
2415 THETAS = ACOS(UNZ)
     GO TO  (2416,2417,3011),NPHASE
2416 VHORIZ = SQRT  (XDOTOC*XDOTOC+YDOTOC*YDOTOC)
     XDOTCO = XDOTOC
     YDOTCO = YDOTOC
     GO TO  3011
2417 XDOTCO = -VH*SIN(XLAMDA)
     YDOTCO = VH*COS(XLAMDA)
     VHORIZ = VH
3011 IF     (VHORIZ) 2418,2418,2420
2418 WRITE  (6,2419) VVERT
2419 FORMAT (/37H***** VH=0   LAMDA=0(ARBITRARY)   VV= F20.5//)
     GO TO  2450
2420 IF     (THETAS) 2421,2423,2430
2421 WRITE  (6,2422)
2422 FORMAT(/22H*****ERROR THETAS*****  //)
     GO TO  2414
2423 IF     (XDOTCO) 2424,2425,2425
2424 EBE    = +1.0
     GO TO  2426
```

```
2425 BBE     = -1.0
2426 COSLAM  = YDOTCQ/VHORIZ
     XSLOPE  = PBB*ACOS(COSLAM)
     GO TO   2448
2430 AMAG    = SQRT (UNX*UNX+UNY*UNY)
     UAX     = UNX/AMAG
     UAY     = UNY/AMAG
     IF      (UAX*YDOTCQ-UAY*XDOTCQ) 2431,2432,2432
2431 BBE     = -1.0
     GO TO   2433
2432 BBE     = +1.0
2433 COSLAM  = (UAX*XDOTOG+UAY*YDOTOG)/VHORIZ
     XSLOPE  = PBB*ACOS(COSLAM)
2448 XSLDEG  = XSLOPE * RADDEG
     WRITE   (6,2447) VHORIZ,VVERT
2447 FORMAT  (/8X5HVH = F5.2, 14H FT/SEC   VV = F5.2,7H FT/SEC   )
     WRITE   (6,2449) XSLDEG
2449 FORMAT(/5X8HLAMDA = F5.2,6H DEGREES /)
2450 THEDEG  = THETAS * RADDEG
     WRITE   (6,2451) THEDEG
2451 FORMAT  (4X9HTHETAS = F5.2,8H DEGREES  //)
     WRITE  (6,2404) XA(1),YA(1),ZA(1)
     WRITE   (6,2405) XA(2),YA(2),ZA(2)
     WRITE   (6,2406) XA(3),YA(3),ZA(3)
     IF      (THETAS) 2452,2452,2458
2452 PSIO = PSIO*RADDEG
     PHIO = PHIO*RADDEG
     XIO = XIO*RADDEG
     WRITE   (6,2454) PSIO
2454 FORMAT  (7X6HPSI = F7.2,6H DEGREES/)
     WRITE   (6,2455) PHIO
2455 FORMAT  (7X6HPHI = F7.2,6H DEGREES/)
     WRITE   (6,2456) XIO
2456 FORMAT  (8X5HXI = F7.2,8H DEGREES/)
     LFLAG   = 1
     GO TO   250
2458 IF      (VHORIZ) 3458,3458,3459
3458 AMAG    = SQRT ( UNX*UNX + UNY*UNY )
     UAX     = UNX/AMAG
     UAY     = UNY/AMAG
3459 IF      (UAX) 2459,2460,2461
2459 BBE     = +1.0
     GO TO   2462
2460 BBE     = 0.0
     GO TO   2462
2461 BBE     = -1.0
2462 COSZET  = UAY
     ZETA    = BBR*ACOS(COSZET)
     ZETDEG  = ZETA*RADDEG
     WRITE  (6,2463)ZETDEG
2463 FORMAT(6X7HZETA = F5.2,8H DEGREES  /)
     SINZET = SIN(ZETA)
     SINPHI = SIN(PHIO)
     COSPHI = COS(PHIO)
     SINPSI = SIN(PSIO)
     COSPSI = COS(PSIO)
     SINXI  = SIN(XIO)
     COSXI  = COS(XIO)
```

```
      CRANK1 = SINPHI*COSZET-SINPSI*COSPHI*SINZET
      STUFF  = SQRT (1.0 - (SINPHI*COSZET-SINPSI*COSPHI*SINZET)**2)
      CRANK2 = (SINPHI*SINZET + SINPSI*COSPHI*COSZET)/STUFF
      CRANK3 = (COSPHI*SINXI*COSZET-(COSPSI*COSXI-SINPSI*SINPHI*SINXI)*
     1         INZET)/STUFF
      PHIO   = ASIN (CRANK1)
      PSIO   = ASIN (CRANK2)
      XIO    = ASIN (CRANK3)
      WRITE  (6,2470)
 2470 FORMAT (32H***** CHECK QUADRANT OF XI *****  //)
      GO TO  2452
C
C     END OF SPECIAL GROUND SLOPE AND CROSS SLOPE ANGLE CALCULATION
C
  250 IF (AB(1,1)-AB(2,1))258,258,252
  252 IF (AB(1,1)-AB(3,1))256,256,254
  254 J=1
      GO TO 262
  256 J=3
      GO TO 262
  258 IF (AB(2,1)-AB(3,1))256,256,260
  260 J=2
  262 W(1,4,2) = -AB(J,1)
      W(2,4,2)  = FLOAT(J)
      RETURN
      END
```

```
      SUBROUTINE STAB
      DIMENSION XMFF(3), ALPHK(2)
      DIMENSION IFLAG(3)
      COMMON/N7/ ALPDOT(3), ALFF(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, FFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(2,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), V(3,4,6), HPL(3,3)
      COMMON/N11/CD(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMFP,XIF,DELCG,XLP,
     1          XLL,XLA,G,ALFHK,XIC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RP(3),
     2          XLDOP,XLDC,XKD(3),XKS,    XNU,XLFMIN,CC,CF(3),XLAMDA,XMUF,
     3          XMUB,RUNNO,SERNO,VV,VH,D6,XIALPH,XIGAM,DELMAX,SFL(3),FP(3)
      COMMON/N15/ D(3,3), XB(3), YB(3), ZB(3), XFF(3), YFP(3), ZFP(3),
     1 DFS(3,3,3)
      COMMON/N17/ XFPC(3), YFPC(3) , ZFPC(3)
      IF (ABS (XDOTC) -.001) 20,20,10
   10 II = 1
      GO TO 60
   20 IF ( ABS ( YDOTC ) -.001) 30,10,10
   30 IF ( ABS ( THETAS ) - .001) 50,40,40
   40 II = 2
      GO TO 60
   50 II = 3
   60 DO 90 J= 1,3
      GO TO ( 62,64,66) , II
   62 XX = YDOTC * XFPC(J) - XDOTC * YFPC(J)
      GO TO 68
   64 XX = - THETAS / ABS ( THETAS ) * XFPC(J)
      GO TO 68
   66 XX = XFPC(J)
   68 IF (XX) 80,70,70
   70 IFLAG(J)= 1
      GO TO 89
   80 IFLAG(J) = 0
   89 DA(5) = 1.0
   90 CONTINUE
      IF( IFLAG(1) - IFLAG(2) )170,130,100
  100 IF( IFLAG(3) )120,120,110
  110 JJ = 2
      GO TO 200
  120 JJ = 1
      GO TO 200
  130 IF( IFLAG(1) - IFLAG(3) )150,160,140
  140 JJ = 3
      GO TO 200
  150 JJ = 3
      GO TO 200
  160 KFLAG = 0
      RETURN
  170 IF( IFLAG(3) )190,190,180
  180 JJ = 1
      GO TO 200
  190 JJ = 2
      GO TO 200
```

```
200 N = JJ +1
    IF( N-3 )220,220,210
210 N = 1
220 IF( IFLAG(JJ) )245,245,230
230 N = JJ -1
    IF( N)240,240,245
240 N = 3
245 GO TO (250,251,251 ),II
250 D1= XDOTC * ( YFPC(N) - YFPC(JJ) ) - YDOTC *( XFPC(N) - XFPC(JJ))
    IF   (ABS(D1)-.0001) 251,251,255
251 XXX = XFPC(JJ) / ( XFPC(JJ) - XFPC(N))
      GO TO 256
255 XXX=   ( YDOTC * XFPC(JJ) - XDOTC * YFPC(JJ)) / D1
256 AX =   XXX * ( XFPC(N) - XFPC(JJ) ) + XFPC(JJ)
    AY =   XXX * ( YFPC(N) - YFPC(JJ) ) + YFPC(JJ)
    AZ =   XXX * ( ZFPC(N) - ZFPC(JJ) ) + ZFPC(JJ)
    AW =   SQRT ( AX * AX +AY*AY + AZ *AZ )
    GO TO   ( 257,258,258),II
257 W(3,4,2) = SIGN (ACOS (AZ/AW),(AX*XDOTC+AY*YDOTC))
    GO TO  259
258 W(3,4,2) = ACOS    (AZ/AW)
259 IF ( W(3,4,2)) 260,260,270
260 KFLAG = -1
    RETURN
270 KFLAG = +1
    RETURN
    END
```

```
      SUBROUTINE FORCE
      DIMENSION XMFF(3), ALPHK(2)
      DIMENSION A(3), AA(3), R(3,3)
      DIMENSION             DELWF(3), FCB(3), DELWD(3), DELWF(3), PC(12),
     1           DELWU(3,2)
      COMMON/N5/ TALPHA(3), TGAMMA(3), F(3), TG(3)
      COMMON/N6/OMEGDT(3), ALFHDD(3), GAMDD(3), G(3), PSID, PHID, XID,
     1 DELWFK
      COMMON/N7/ ALPDOT(3), ALFH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, FFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPI(3,3)
      COMMON/N10/FOOT1(100), FOOT2(100), FOOT3(100),
  101              BLOCK1(100), BLOCK2(100), BLOCK3(100),
  102              SHOCK1(100), SHOCK2(100), SHOCK3(100)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMFP,XIP,DELCG,XLF,
     1         XLL,XLA,G,ALFHK,XIC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2         XLDOP,XLLO,XKD(3),XKS,    XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3         XMUB,RUNNO,SERNO,VV,VH,DB,XTALPH,XIGAM,DELMAX,SPL(3),FP(3)
      COMMON/N13/ A, AA
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),FSB(3),XHPU(3),
     1         YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELFL(3),F1MAX(3),
     2         ZPFEF,SINTH(3),COSTH(3),TZFPO,VZERO,ZBO(3)
      COMMON/N15/ B(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DIS(3,3,3)
      COMMON/N16/ RN(3), X11(3,2), XLD(3), XLDDOT(3), XLDTF(3,2),
     1 YLDTF(3,2), S(3,2), COSXI (3), HF(3,12), ZLDTF(3,2), HB(3),
     2 GIS(3,3,3), TA
      COMMON/N18/ NFLAG
      COMMON/N19/NNFLAG(3),PSBG(3),PSFG(3,12),IBEEN(6)
      COMMON/N20/DWORK(15)
      COMMON/N24/ICPT(10), JN, DTF
      NUMP    = 0
      NACIC   = 0
      DO 290 J =1,3
      DO 170 N=1,2
      IF      (N-1) 9,9,2
    2 IF (NNFLAG(J))4,4,10
    4 DO 6 L=1,3
      A4(J,2,L)=0.0
      A5(J,2,L)=0.0
    6 CONTINUE
      W(J,4,3) = 0.0
      A3(2)   = 0.0
      IF (NFLAG) 801,170,8
    8 PSBO(J) = PSB(J)
      GO TO 170
  801 PSB(J)  = PSBG(J)
      GO TO   170
    9 DO 901 L=1,2
      IF      (HF(J,L) + PSF(J,L) - HFO + .000001 ) 10,901,901
  901 FC(L)   = 0.0
      NFLAG   = 1
```

```
      DO 902 L= 1,3
      A4(J,1,L) = 0.0
  902 A5(J,1,L) = 0.0
      A3(1)   = C.0
      W(J,4,4) = 0.0
      IF      (NFLAG) 905,170,903
  903 DO 904 L= 1,2
  904 FSPG(J,L) = FSP(J,L)
      GO TO  170
  905 DO 906 L= 1,2
  906 FSP(J,L) = PSPQ(J,L)
      GO TO  170
   10 NFLAG   = 0
      IF      (N-1) 12,12,13
   12 XMU     = XMUF
      GO TO   14
   13 XMU     = XMUL
   14 IF      (S(J,N)-.05) 20,20,30
   20 A2(  N) = 24.0 *      S(J,N)   * XMU
      GO TO 60
   30 IF(  S(J,N)   - 1.05 ) 50,50,40
   40 A2(  N) = XMU
      GO TO 60
   50 A2(  N) =  XMU  *( 1.21-.20 *        S(J,N))
   60 IF( N -1)80,80,70
   70 DELB = HBO - HB(J)
      IF      (NFLAG) 7001,7002,7002
 7001 FSB(J) = PSBG(J)
 7002 IF      ( DELB - PSB(J) - .000001)  7004,7004,7010
 7004 FCB(J)  = 0.0
      GO TO 7092
 7010 IF( ZDOTF(J,2)        )7090,7020,7020
 7020 KDUM    = 1
      GO TO   (7021,7022,7023),J
 7021 CALL    ENTRP(DELB,PCB(J), DUM, BLOCK1, KDUM)
      GO TO   7080
 7022 CALL    ENTRP (DELB, FCB(J), DUM, BLOCK2, KDUM)
      GO TO   7080
 7023 CALL    ENTRP (DELB, FCB(J), DUM, BLOCK3, KDUM)
 7080 A7      = DELB - PCB(J) / SLOPE2(J)
 7084 IF ( A7 - PSB(J)) 7090,7085,7085
 7085 FSB(J) = A7
      GO TO 7092
 7090 PCB(J) = SLOPE2(J)*(DELB-FSB(J))
 7092 IF ( NFLAG ) 7093,75,7094
 7093 PSB(J) = PSBG(J)
      GO TO 75
 7094 FSBQ(J) = PSB(J)
   75 A3(N) =  DA(10) * PCB(J) * ( .75 + .25 * COS ( 2.0 * XII (J,N) ) )
     1        *CB(J)
      W(J,4,3)=A3(N)*COSXI(J)
      GO TO 120
   80 IF(  XII(J,N) -     DA(9)        )100,100,90
   90 A3(  N) = 187.0
      GO TO 115
  100 SUMPC = 0.0
 1005 DO 1095 K =1,2
 1010 DELP = HFO - HF(J,K)
```

```
      IF      (NFLAG) 1011,1012,1012
 1011 FSP(J,K) = PSPQ(J,K)
 1012 IF      (DELP - PSP(J,K) - .000001) 1013,1013,1014
 1013 FC(K) = 0.0
      GO TO 1030
 1014 IF( XDOTP(J,1) )        1020,1040,1040
 1020 FC(K) = SLOPE1(J)*(DELP-FSP(J,K))
 1030 IF      (NFLAG) 1092,1092,1034
 1034 FSPG(J,K) = FSP(J,K)
      GO TO 1092
 1040 KDUM    = 1
      GO TO   (1041,1042,1043), J
 1041 CALL    ENTRP (DELP, FC(K), DUM, FOOT1, KDUM)
      GO TO   1082
 1042 CALL    ENTRP (DELP, FC(K), DUM, FOOT2, KDUM)
      GO TO   1082
 1043 CALL    ENTRP (DELP, FC(K) , DUM, FOOT3, KDUM)
 1082 XX          = DELP - FC(K) / SLOPE1(J)
 1086 IF (XX-PSP(J,K)) 1020,1087,1087
 1087 FSP(J,K) = XX
      GO TO 1030
 1092 SUMPC = SUMPC + FC(K)
 1095 CONTINUE
  110 DA(5) =   DA(6) / COSXI(J)   * ( .75 + .25 * COS ( 2.0 * XII(J,N) ))
     1         *CF(J)
      A3(  N) =  DA(5) * SUMFC
  115 W(J,4,4)=A3(N)*COSXI(J)
  120 DO 125  L =1,3
      A4(J,N,L) =   - A3(  N) * RN(L)
  125 CONTINUE
  130 IF (S(J,N))140,140,150
  140 DO 145  L=1,3
C
C
C     WHEN  A5  IS ZEROED HERE, THEN  AA(L)  SHOULD ALSO
C     EQUAL  0.0 IN BOX 210 AND SETTING DA(L) EQUAL TO ZERO WILL DO
C     JUST THIS. THIS IS DONE ONLY FOR N = 1.
C
      A5(J,N,L) = 0.0
      IF ( N-1 ) 142,142,145
  142 DA(L) = 0.0
  145 CONTINUE
      GO TO 170
  150 A(1) = XDOTP(J,N) /         S(J,N)
      A(2) = YDOTP(J,N) /         S(J,N)   * COSTHS
      A(3) = YDOTP(J,N) /         S(J,N)   * ( - SINTHS)
      DO 156 L=1,3
      A1(L) = 0.0
      DO 155 I =1,3
      A1(L) = A1(L) + B(I,L) * A(I)
  155 CONTINUE
      IF ( N-1 ) 153,153,156
  153 DA(L) = A1(L)
  156 CONTINUE
      D5 =   -A2(N) * A3(N)
      IF      (N-1) 157,157,160
  157 W(J,1,6) = A3(N)
      W(J,2,6) = D5
```

```
      W(J,J+6) = A2(N)
  160 DO 165 L =1,3
      A5(J,N,L) = D5 * A1(L)
  165 CONTINUE
  170 CONTINUE
  180 DEL =  XLDO - XLD(J)
      L       = 1
      IF ( ABS(DEL) - .000001  ) 1802,1803,1803
 1802 DEL = 0.0
 1803 IF      (DEL-DELEL(J)) 1805,1805,1810
 1805 D1 = XKS * DEL
      L       = 2
      GO TO  1820
 1810 IF      (NFLAG) 1820,1820,1811
 1811 IF( DEL - DELMAX ) 1820,1820,1815
 1815 FFLAG = -1.0
      WRITE (6,940) J
  940 FORMAT(//24H    SHOCK ABSORBER NO.   ,I4,6X,45H HAS BOTTOMED OUT.
     1THIS IS END OF THIS CASE.//)
      RETURN
 1820 KDUM    = 2
      GO TO (1821,1822,1823), J
 1821 CALL   ENTRP (DEL, SK, SF, SHOCK1, KDUM)
      GO TO  1865
 1822 CALL   ENTRP (DEL, SK, SF, SHOCK2, KDUM)
      GO TO 1865
 1823 CALL   ENTRP (DEL, SK, SF, SHOCK3, KDUM)
 1865 GO TO  (1866,1870), L
 1866 D1       = XKD(J) * SK * (DEL-DELEL(J)) + FP(J)
 1870 IF( XLDDOT(J) )1880,1880,1875
 1875 JFLAG = -1
      RD       = RR(J)
      SD       = 1.0 + SRL(J) * DEL
      GO TO 1890
 1880 JFLAG = 1
      RD       = RC(J)
 1890 IF      (GAMDOT(J)) 1896,1889,1889
 1889 IF      (MFLAG) 1940,1940,1891
 1891 IF      (DEL) 1892,1892,1896
 1892 MUMP     = 1
 1893 D2       = 0.0
      D3       = 0.0
      D4       = 0.0
      DO 1894 L=1,3
 1894 W(L,J+1) = 0.0
      GO TO  191
 1896 D2       = XNU*FLOAT(JFLAG)*ABS(D1)
 1897 D3       = FLOAT(JFLAG)*XLDDOT(J)*XLDDOT(J)*RD*SD
C     LITTLE OL SMOOTHY
      GADOT    = ABS(GAMDOT(J))
      IF       (GADOT- .20) 1898,1898,1899
 1898 FACTOR = GADOT/ .20
      D23      = (((FACTOR-3.0)*FACTOR) + 3.0)*FACTOR
      D2       = D2*D23
      D3       = D3*D23
 1899 D4       = D1 + D2 + D3
      GO TO    1903
 1940 IF       (DEL) 1904,1904,1896
```

```
1904 MAGIC   = 1
     D2      = 0.0
     D3      = 0.0
     D4      = 0.0
1903 W(1,J,1) = D4 / XLD(J) * ( XHPU(J) - XFP(J))
     W(2,J,1) = D4 / XLD(J) * ( YHPU(J) - YFP(J))
     W(3,J,1) = D4 / XLD(J) * ( ZHPU(J) - ZFP(J))
 191 DO 192 L=1,3
     W(L,J,5) = A4(J,1,L) + A5(J,1,L)
     W(L,J,3) = A4(J,2,L) + A5(J,2,L)
     A(L) = A4(J,1,L) + A5(J,1,L) - W(L,J,1)
 192 CONTINUE
     DO 194  L=1,3
     R(J,L) = 0.0
     DO 194  I =1,3
     R(J,L) =  R(J,L) + GRS(J,I,L) * A(I)
 194 CONTINUE
     IF      (MAGIC) 1944,1944,1941
1941 IF      (R(J,3)) 1944,1944,1942
1942 DO 1943 L=1,3
     W(L,J,1) = A4(J,1,L) +A5(J,1,L)
1943 R(J,L) = 0.0
1944 MAGIC   = 0
     A6(1) = R(J,1)
     A6(2) = R(J,2)
     DO 196 L=1,3
     W(L,J,2)=0.0
     DO 196 I=1,2
     W(L,J,2)=W(L,J,2)+GRS(J,L,I)*A6(I)
 196 CONTINUE
     W(1,J,4)=W(3,J,2)*(YFP(J)-HPL(J,2))-W(2,J,2)*(ZFP(J)-HPL(J,3))
     W(2,J,4)=W(1,J,2)*(ZFP(J)-HPL(J,3))-W(3,J,2)*(XFP(J)-HPL(J,1))
     W(3,J,4)=W(2,J,2)*(XFP(J)-HPL(J,1))-W(1,J,2)*(YFP(J)-HPL(J,2))
 200 W(J,4,1) = SQRT  ( W(1,J,1)*W(1,J,1) + W(2,J,1)*W(2,J,1)+W(3,J,1)*
    1W(3,J,1))
     IF      ( MUMF  ) 204,204,203
 203 TGAMMA(J) = -75.*XIGAM *AMAX1((GAMDOT(J) - .001), 0.0)
     GAMDOT(J) = 0.0
     MUMP    = 0
     GO TO  205
 204 TGAMMA(J) =XLL * R(J,3)
 205 TALPHA(J) = 0.0
     IF      (MFLAG) 210,210,224
 210 DO 215 L = 1,3
     A(L) = 0.0
     AA(L) = 0.0
     DO 215 I =1,3
     A(L)   = A(L) + DRS(J,I,L) * RN(I)
     AA(L) = AA(L) + DRS(J,I,L) * DA(I)
 215 CONTINUE
 220 D5 = A(3) + A2(1) * AA(3)
     D6 = A(2) + A2(1) * AA(2)
     IF(X1I(J,1) - DA(9) )221,221,223
 221 DO 222 K =1,2
 222 TALPHA(J) = TALPHA(J) - DA(5) * PC(K) * ( - YCDP(K) *  D5
    1  + HF(J,K)   * D6   )
     GO TO 224
 223 TALPHA(J) = -187.0 * ( -DF / 2.0 * D5 + HF(J,K) / 2.0 * D6 )
```

```
224 IF (ALPH(J) - ALPHA(1)) 225,226,226
225 N=1
    D5 = +1.0
    GO TO 229
226 IF (ALPH(J) - ALPHA(2)) 235,235,228
228 N=2
    D5= -1.0
229 D6      = ALPDOT(J)*(1.0-1000.0*(D5*(ALPHA(N)-ALPH(J))))
    IF      (ALPDOT(J)*D5) 230,235,235
230 IF      (D5*D6) 233,231,231
231 ALFDOT(J) = 0.0
    GO TO  235
233 ALFDOT(J) = D6
235 CONTINUE
250 DELWE(J) = A3(2) *  ZDOTP(J,2)
    DELWL(J) = - D4*XIDDOT(J)
    DO 255 N =1,2
    DELWU(J,N)=A2(N) * A3(N) *       S(J,N)
255 CONTINUE
260 IF( XII(J,1) -   DA(9)        )270,270,280
270 DELWF(J) = 0.0
    D6 = DA(5) * ZDOTP(J,1)
    DO 275 K =1,2
    DELWF(J) = DELWF(J) + D6 * PC(K)
275 CONTINUE
    GO TO 290
280 DELWF(J) = 0.0
290 CONTINUE
300 DO 304 L = 1,3
    A(L) = 0.0
    DO 302 J = 1,3
    IF( NNFLAG(J) )301,301,303
301 A(L) = A(L) + A4(J,1,L) + A5(J,1,L)
    GO TO 302
303 A(L) = A(L) + A4(J,1,L) + A5(J,1,L) +
   1 A4(J,2,L) + A5(J,2,L)
302 CONTINUE
    F(L) = DA(7)   * E(3,L) + A(L)
304 CONTINUE
310 TQ(1)   = 0.0
    TQ(2)= 0.0
    TQ(3)= 0.0
    DO 3101 J= 1,5
3101 DWCRK(J) = 0.0
    DO 315 J=1,3
    IF( NNFLAG(J)  )311,311,314
311 TQ(1) = TQ(1) + YFP(J) * ( A4(J,1,3) + A5(J,1,3)          )
   1                -ZFP(J) * ( A4(J,1,2) + A5(J,1,2)          )
    TQ(2) = TQ(2) + ZFP(J) * ( A4(J,1,1) + A5(J,1,1)          )
   1                -XFP(J) * ( A4(J,1,3) + A5(J,1,3)          )
    TQ(3) = TQ(3) + XFP(J) * ( A4(J,1,2) + A5(J,1,2)          )
   1                -YFP(J) * ( A4(J,1,1) + A5(J,1,1)          )
    GO TO 315
314 TQ(1) = TQ(1) + YB(J) *( A4(J,2,3) + A5(J,2,3) ) - ZB(J) *( A4(J,2
   1 ,2) + A5(J,2,2) ) + YFP(J) *( A4(J,1,3) + A5(J,1,3)          )
   2 - ZFP(J) * ( A4(J,1,2) + A5(J,1,2)          )
    TQ(2) = TQ(2) + ZB(J) *( A4(J,2,1) + A5(J,2,1) ) - XB(J) *( A4(J,2
   1 ,3) + A5(J,2,3) ) + ZFP(J) *( A4(J,1,1) + A5(J,1,1)          )
```

```
     2 - XFF(J) * ( A4(J,1,3) + A5(J,1,3)                  )
      TQ(3) = TQ(3) + XB(J) *( A4(J,2,2) + A5(J,2,2) ) - YB(J) *( A4(J,2
     1 ,1) + A5(J,2,1) ) + XFP(J) *( A4(J,1,2) + A5(J,1,2)          )
     2 - YFF(J) * ( A4(J,1,1) + A5(J,1,1)                  )
  315 CONTINUE
      IF      (IOPT(3)) 340,320,340
  320 DO      325 J=1,5
  325 DWORK(J) = 0.0
      DO      330 J=1,3
      DWORK(1) = DWORK(1) + DELWD(J)
      DWORK(2) = DWORK(2) + DELWF(J)
      DWORK(3) = DWORK(3) + DELWF(J)
      DWORK(4) = DWORK(4) + DELWU(J,1)
  330 DWORK(5) = DWORK(5) + DELWU(J,2)
      GO TO  360
  340 DO      345 J=1,15
  345 DWORK(J) = 0.0
      DO      350 J=1,3
      K       = (J-1)*5
      DWORK(K+1) = DELWD(J)
      DWORK(K+2) = DELWF(J)
      DWORK(K+3) = DELWB(J)
      DWORK(K+4) = DELWU(J,1)
  350 DWORK(K+5) = DELWU(J,2)
  360 RETURN
      END
```

```
      SUBROUTINE ENTRP (X, FX1, FX2, T, K)
      DIMENSION FX(2), T(2)
      L        = 1
      N        = K
      J        = IFIX(T(2))
      M1       = J+2
      DO       10 I= 4,M1
      IF       (X-T(I)) 4,2,10
    2 M        = J*L
      FX(L)    = T(I+M)
      GO TO    (99,3), N
    3 L        = 2
      N        = 1
      GO TO    2
    4 M        = J*L+I
      FX(L)    = (T(M)-T(M-1))/(T(I)-T(I-1))*(X-T(I-1))+T(M-1)
      GO TO    (99,5), N
    5 L        = 2
      N        = 1
      GO TO    4
   10 CONTINUE
   20 M        = J*L+M1
      FX(L)    = (T(M)-T(M-1))/(T(M1)-T(M1-1))*(X-T(M1-1))+T(M-1)
      GO TO    (99,25), N
   25 L        = 2
      N        = 1
      GO TO    20
   99 FX1      = FX(1)
      FX2      = FX(2)
      RETURN
      END
```

```
      SUBROUTINE INTEGM
      DIMENSION XMPP(3), ALPHK(2)
      DIMENSION A(3,3), AA(3,3), XJINV(3,3), BAZ(3)
      COMMON/N5/ TALPHA(3), TGAMMA(3), F(3), TQ(3)
      COMMON/N6/OMEGDT(3), ALPHDP(3), GAMDD(3), G(3), PSID, PHID, XID,
     1 DELWRK
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), Y(3,4,6), HPI(3,3)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMPP,XIP,DFLCG,XLF,
     1          XLL,XLA,G,ALPHK,XIC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2          XLDOP,XLLO,XKL(3),XKS,    XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3          XMUB,RUNNO,SERNO,VV,VH,DB,XIALPH,XIGAM,DELMAX,SRL(3),FP(3)
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),PSB(3),XHPU(3),
     1          YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DFLFL(3),F1MAX(3),
     2          ZPREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ B(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DIS(3,3,3)
      COMMON/N23/MISS
      IF       (MISS) 5,5,35
    5 DO       6 I=1,3
      DO       6 J=1,3
    6 A(I,J) = XJ(I,J)
      DO        9 J=1,3
    9 AA(J,J) = 1
      DO       20 I=1,3
      TEMP     = A(I,I)
      DO       10 J=1,3
      A(I,J) = A(I,J)/TEMP
   10 AA(I,J) = AA(I,J)/TEMP
      K        = I+1
      L        = I+3-1
      DO       20 M=K,L
      IF       (M-3) 21,21,22
   21 LL       = M
      GO TO    23
   22 LL       = M-3
   23 TEMP     = A(LL,I)
      DO       20 J=1,3
      A(LL,J) = A(LL,J)-A(I,J)*TEMP
   20 AA(LL,J) = AA(LL,J)-AA(I,J)*TEMP
      MISS     = 1
      DO       31 I=1,3
      DO       31 J=1,3
   31 XJINV(I,J) = AA(I,J)
      DO       33 K=1,3
      DO       33 M= 1,3
      SUM      = 0.0
      DO       32 N=1,3
   32 SUM      = XJ(K,N)*XJINV(N,M) + SUM
   33 AA(K,M) = SUM
      WRITE (6,34) (((XJ(M,N),N=1,3),(XJINV(M,N),N=1,3),(AA(M,N),N=1,3))
```

```
     1          ,M=1,3)
  34 FORMAT (/3(2XF10.5),6X3(2XF10.5),6X3(2XF10.5)/
     1          3(2XF10.5),6X3(2XF10.5),6X3(2XF10.5)/
     2          3(2XF10.5),6X3(2XF10.5),6X3(2XF10.5)/ )
  35 BAZ(1) = TQ(1) +(OMEG(2)*OMEG(2)-OMEG(3)*OMEG(3))*YIZ+OMEG(1)*
     1          OMEG(2)*XIZ-OMEG(1)*OMEG(3)*XIY-(ZIZ-YIY)*OMEG(2)*OMEG(3)
     BAZ(2) = TQ(2) +(OMEG(3)*OMEG(3)-OMEG(1)*OMEG(1))*XIZ+OMEG(2)*
     1          OMEG(3)*XIY-OMEG(2)*OMEG(1)*YIZ-(XIX-ZIZ)*OMEG(3)*OMEG(1)
     BAZ(3) = TQ(3) +(OMEG(1)*OMEG(1)-OMEG(2)*OMEG(2))*XIY+OMEG(3)*
     1          OMEG(1)*YIZ-OMEG(3)*OMEG(2)*XIZ-(YIY-XIX)*OMEG(1)*OMEG(2)
     DO     50 J= 1,3
     PAR    = 0.
     OMEGDT(J) = 0.0
     DO     40 K=1,3
     OMEGDT(J)=XJINV(J,K)*BAZ(K) + OMEGDT(J)
  40 PAR    = B(J,K)*F(K)+PAR
     G(J)   = PAR/XM
  42 GAMDD(J) = TGAMMA(J)/XIGAM
     ALPHDD(J)= TALPHA(J)/XIALPH + GAMDD(J)
  50 CONTINUE
C
C    D12, D15, D16, D13 ARE SET  '  =   TO COS(PHI), SIN(PHI),
C     SIN (XI),  AND COS(XI) RESPECTIVELY IN CONFIG AND HELD FOR HERE.
C
     CPHI   = D12
     IF     (CPHI  ) 70,60,70
  60 XID    = XID
     PSID   = SIGN ((OMEG(3)-XID),PHI)
     PHID   = SQRT (OMEG(2)*OMEG(2) + OMEG(1)*OMEG(1))
     GO TO  90
  70 SPHI   = D15
     TPHI   = SPHI/CPHI
     SXI    = D16
     CXI    = D13
     A(1,1) = CXI/CPHI
     A(1,2) = - SXI/CPHI
     A(1,3) = 0.0
     A(2,1) = SXI
     A(2,2) = CXI
     A(2,3) = 0.0
     A(3,1) = -CXI*TPHI
     A(3,2) = SXI*TPHI
     A(3,3) = 1.0
     PSID   = 0.
     PHID   = 0.
     XID    = 0.
     DO     80 J= 1,3
     PSID   = A(1,J)*OMEG(J)+PSID
     PHID   = A(2,J)*OMEG(J)+PHID
  80 XID    = A(3,J)*OMEG(J)+XID
  90 RETURN
     END
```

```
      SUBROUTINE RKP
      DIMENSION A(50),C(50),V(50),DY1(50),DY2(50)
      DIMENSION XK1(3),XK2(3),XK3(3)
      DIMENSION ERROR(50), ABSER(50), ABSUM(50) , ARITH(50)
      DIMENSION  NK1(8), NK4(8), NK9(50)
      COMMON/N1/  N,TO,TF,SH,HM ,YI(50),YO(50),ERR(50)
      COMMON/N2/T,Y(50),D(50)
      COMMON/N3/TP,U(50),H,IELSW
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N8/ LFLAG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),PSB(3),XHPU(3),
     1        YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELFL(3),FIMAX(3),
     2        ZPREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ B(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DFS(3,3,3)
      COMMON/N18/ NFLAG
      LFLAG   = -1
      TP      = 0.0
      T       = TO
      XK1(1)  = 1.0
      XK1(2)  = 2.0
      XK1(3)  = 2.0
      XK2(1)  = 1.0
      XK2(2)  = 1.0
      XK2(3)  = 2.0
      XK3(1)  = 0.5
      XK3(2)  = 0.0
      XK3(3)  = 0.5
      XLOG10  = ALOG(10.0)
      XLOG2   = ALOG(2.0)
    4 IF      (SH) 5,6,6
    5 SGN     = -1.0
      GO TO   7
    6 SGN     = 1.0
    7 CONTINUE
      IF      (N) 480,480,10
   10 IF      (N-50) 20,20,480
   20 IF      (TF-TO) 30,480,30
   30 IF      (SGN*(HM-SH)) 40,40,480
   40 IF      (HM) 50,480,50
   50 H       = SH
      IF      (SGN*(TF-T-H)) 60,70,70
   60 H       = TF-T
   70 IPTSW   = 1
      IELSW   = 1
      KK1     = 0
      ICNT    = 0
      DO      71 I= 1,8
      NK4(I)  = 0
   71 NK1(I)  = 0
```

```
      DO      72 I= 1,50
      ARITH(I) = 0.0
      ABSUM(I) = 0.0
   72 NKS(I) = 0
      DO      80  I= 1,N
      U(I)    = YI(I)
   80 A(I)    = YI(I)
      CALL    PRT
      DO      110 I= 1,N
      V(I)    = A(I)
  110 C(I)    = A(I)
  120 IRSSW   = 1
      IRKC    = 0
      NFLAG   = 1
      DO      125 I=1,N
  125 Y(I)    = A(I)
      CALL    FEP
      IF      (LFLAG) 126,126,409
  126 NFLAG   = 0
      DO      130 I = 1,N
  130 DY1(I)  = D(I)
  140 H2      = .5*H
  150 DO      170 K=1,3
      ZIF     = H*XK1(K)/6.0
      ZAF     = H2*XK2(K)
      DO      160 I=1,N
      V(I)    = V(I) + D(I) * ZIF
  160 Y(I)    = C(I) + D(I) * ZAF
      T       = T+XK3(K)*H
      CALL    FEP
      NFLAG   = 0
  170 CONTINUE
      ZIF     = H/6.0
      DO      180 I = 1,N
  180 V(I)    = V(I) + D(I) * ZIF
      IRKC    = IRKC + 1
      GO TO   (190,210,240),IRKC
  190 DO      200 I=1,N
      D(I)    = DY1(I)
      U(I)    = V(I)
  200 V(I)    = A(I)
      T       = T-H
      H       = H2
      NFLAG   = -1
      GO TO   140
  210 DO      220 I=1,N
      Y(I)    = V(I)
  220 C(I)    = V(I)
      CALL    FEP
      DO      230 I=1,N
  230 DY2(I)  = D(I)
      GO TO   150
  240 U2MAX   = 0.
      DO      270 I=1,N
      V(I)    = V(I)-(U(I)-V(I))/15.0
      ERROR(I) = (U(I) - V(I)) / 180.0
      ABSER(I) = ABS (ERROR(I))
      IF      ((I-14)/11) 270,245,270
```

```
245 U2      = ABSER(I)/(2.0*H*ERR(I))
    IF      (U2-1.0) 250,250,430
250 IF      (U2-U2MAX) 270,270,260
260 U2MAX   = U2
    K9      = I
270 CONTINUE
    GO TO   (280,350), IPTSW
280 IF      (SGN*(T-2.0*H-TF)) 290,330,330
290 IF      (SGN*(TP-T)) 310,300,340
300 TPH     = TP
    DO      305 I= 1,N
305 U(I)    = V(I)
    CALL    PRT
    IF      (SGN*(TPH-TP)) 340,330,330
310 TPH     = TP
    HP      = TP-T+H+H
    HSQ     = H*H
    HTD     = HSQ*H
    HTH     = HSQ*HSQ
    DO      320 I=1,N
    EE      = A(I)
    DD      = DY1(I)
    C1      = C(I)
    DD2     = DY2(I)
    AA      = ((V(I)-5.*EE)*.25+C1-H*(DD*0.5+DD2))/HTH
    BB      = (2.0*(EE-C1-AA*HTH)+H*(DD+DD2))/HTD
    C1      = (C1-EE-AA*HTH-H*DD-BB*HTD)/HSQ
320 U(I)    = EE+HP*(DD+HP*(C1+HP*(BB+HP*AA)))
    CALL    PRT
    IF      (SGN*TPH-TP)  290,330,330
330 IPTSW   = 2
340 GO TO   (350,410) , IEDSW
350 IF      (SGN*(TF-T)) 403,403,360
360 IF      (2.0*H-1.0) 363,363,361
361 WRITE (6,362) H
362 FORMAT (//60X3HH=  F10.5/)
    K4      = 1
    GO TO   364
363 K4      = MINO(8,(INT(ALOG(0.5/H)/XLOG10) + 1))
364 NK4(K4) = NK4(K4) + 1
    K1      = MINO( 6,(INT(ALOG(SH/H*.5)/XLOG2)+1))
    NK1(K1) = NK1(K1) + 1
    ICNT    = ICNT + 1
    NK9(K9) = NK9(K9) + 1
    H       = (2.0-U2MAX)*2.0*H
    SH      = H
    DO      369 I = 1,N
    ARITH(I) = ARITH(I) + ERROR(I)
369 ABSUM(I) = ABSUM(I) + ABSER(I)
    IF      (SGN*(TF-T-H)) 370,390,390
370 IF      (SGN*(TF-T-HM))403,360,380
380 H       = TF-T
390 DO      400 I= 1,N
    A(I)    = V(I)
400 C(I)    = V(I)
    GO TO   120
403 LFLAG   = 2
    CALL    FEP
```

```
409 TP     = T
    LFLAG  = 1
    DO     405 I= 1,N
405 U(I)   = V(I)
421 WRITE  (6,422) ( ARITH(I), I= 1,29)
422 FORMAT (/10X34HARITHMETIC SUM OF ERRORS FOLLOWS-  /
4221        8(1XE14.7)/8(1XE14.7)/8(1XE14.7)/5(1XE14.7)/)
    WRITE  (6,423) ( ABSUM(I), I=1,29)
423 FORMAT (/10X35HSUM OF THE ABSOLUTE ERRORS FOLLOWS-  /
4231        8(1XE14.7)/8(1XE14.7)/8(1XE14.7)/5(1XE14.7)/)
    AVG    = (T-TO)/AMAX1(FLOAT(ICNT),1.0)
    WRITE (6,427) ICNT, AVG, (NK4(I), I=1,8), (NK1(K1),K1=1,8),
   1        (NK9(K9),K9=1,N)
427 FORMAT (/10X19HNO. OF INTERVALS =  I6/
4271        10X16HAVG. INTERVAL =  F11.8/
4272        10X21HINTERVAL DISTRIBUTION  / 10X 8I8//
4273        10X 16HCUT DISTRIBUTION  / 10X8I8//
4274        10X30HCRITICAL VARIABLE DISTRIBUTION / 5(10X10I8/))
    CALL   PRT
410 TP     = T
    DO     420  I=1,N
420 YO(I)  = V(I)
    RETURN
430 IF     (H - HM) 431,431,433
431 IRSSW  = 2
    WRITE (6,432) T,I,H,V(I), U2
432 FORMAT (11H*****TIME= F10.4,5X3HI= I2,5X10HINTERVAL= F10.6,5X7HVAL
   1UE= F10.4, 5X7HERROR= F10.5)
433 GO TO  (450,440),IRSSW
440 U2     = 1.0
    GO TO  250
450 T      = T-2.0*H
460 IRKC   = 1
    H      = H2
    DO     470 I=1,N
    U(I)   = C(I)
    V(I)   = A(I)
    C(I)   = A(I)
470 D(I)   = DY1(I)
    NFLAG  = -1
    GO TO  140
480 H      = 10.E6
    WRITE  (6,490)
490 FORMAT (//10X28HRUNGE-KUTTA SUBROUTINE ERROR //)
    STOP
    END
```

```
      SUBROUTINE PRT
      DIMENSION XMPP(3), ALPHK(2)
      DIMENSION ODLAP (55)
      COMMON/N3/TP,U(50),H,IELSW
      COMMON/N7/ ALPDOT(3), ALFF(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, FFLAG, PFLAG, PSIO,
     2 PHIC, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADLEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N11/CL(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMFP,XIF,DELCG,XLP,
     1        XLL,XLA,G,ALPHK,XLC,HBO,NPHASE,HFO,HFMIN,FF,RC(3),RR(3),
     2        XLDOP,XLDC,XKD(3),XKS,    XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3        XMUB,RUNNC,SERNC,VV,VH,DB,XTALPH,XIGAM,DELMAX,SRL(3),FP(3)
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),FSB(3),XHPU(3),
     1        YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DFLEL(3),F1MAX(3),
     2        ZFREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N16/ FN(3), XII(3,2), XLD(3), XLDDOT(3), XDOTF(3,2),
     1 YDOTF(3,2), S(3,2), COSXI (3), HF(3,12), ZDOTF(3,2), HB(3),
     2 GFS(3,3,3),TA
      COMMON/N22/FOOT(3,3)
      COMMON/N24/IOPT(10),JN,DTF
      COMMON/N25/IWILEY(10)
      IF      (IOPT(1) + IOPT(2) + IOPT(3)) 1,3,1
    1 WRITE   (6,2)
    2 FORMAT  (/)
    3 LSET    = 2
      IF       (JN) 4,9,50
    4 JN       = 2
      GO TO 10
    9 JN       = 1
   10 WRITE   (6,20)
   20 FORMAT  (///10X32H$$$$$ PUNCH OPTION ENABLED $$$$$  //)
      PHI=SQRT(50.0)
   50 DO      60 J = 1,3
      IF      (IOPT(J)) 80,60,60
   60 WRITE   (6,70)
   70 FORMAT  (10X46HTHE FOLLOWING ARRAYS ARE ORGANIZED AS FOLLOWS-  //)
      DO      75 I = 1,3
      DO      75 K = 1,3
   75 FOOT(I,K) = 0.0
      W(3,4,2) = 0.0
      W(1,4,2) = 0.0
      ZERO    = U(18)
      GO TO  100
   80 CONTINUE
  100 IF      (IOPT(1)) 110,200,104
  104 IOPT(1) = -1
      WRITE   (6,105)
  105 FORMAT  (19X1HT,
 1051         14X6HOMEGAX,14X6HOMEGAY,14X6HOMEGAZ,15X5HSIGMA,16X2HMC/
 1051  19X1HX,19X1HY,19X1HZ,16X4HXDOT,16X4HYDOT,16X4HZDOT/
 1052  17X3HPSI,17X3HPHI,18X2HXI,14X6HPSIDOT,14X6HPHIDOT,15X5HXIDOT/
 105314X6HGAMMA1,14X6HGAMMA2, 14X6HGAMMA3,13X7HGAMDOT1,13X7HGAMDOT2,13X
 1054  7HGAMDOT3/17X3HXP1,17X3HYP1,17X3HZP1,17X3HXP2,17X3HYP2,17X3HZP2/
```

```
1055 17X3HXF3,17X3HYF3,17X3HZF3//)
     LSET    = 1
     GO TO   200
 110 CPHI    = COS(U(23))
     IF      (CPHI) 130,120,130
 120 XID     = XID
     FSID    = SIGN((U(21)-XID), U(23))
     PHID    = SQRT(U(20)*U(20) + U(19)*U(19))
     GO TO   145
 130 SPHI    = SIN(U(23))
     TPHI    = SPHI/CPHI
     SXI     = SIN(U(24))
     CXI     = COS(U(24))
     AB(1,1) = CXI/CPHI
     AB(1,2) = -SXI/CPHI
     AB(1,3) = 0.0
     AB(2,1) = SXI
     AB(2,2) = CXI
     AB(2,3) = 0.0
     AB(3,1) = -CXI*TPHI
     AB(3,2) = SXI*TPHI
     AB(3,3) = 1.0
     FSID    = 0.0
     PHID    = 0.0
     XID     = 0.0
     DO      140 J=1,3
     FSID    = AB(1,J)*U(J+18) + FSID
     PHID    = AB(2,J) * U(J+18) + PHID
 140 XID     = AB(3,J) * U(J+18) + XID
 145 DO 150 J= 1,3
     U(J+18) = U(J+18) * RADDEG
     U(J+21) = U(J+21) * RADDEG
     U(J+6)  = U(J+6) * RADDEG
 150 U(J+9)  = U(J+9) * RADDEG
     FSID    = FSID * RADDEG
     PHID    = PHID * RADDEG
     XID     = XID  * RADDEG
     W342    = W(3,4,2) * RADDEG
     U(18)   = U(18) - ZERO
     WRITE   (6,160) TP, (U(J+18), J=1,3), W342, W(1,4,2), (U(J+15), J
    1 =1,3),(U(J+12),J=1,3), (U(J+21),J=1,3), FSID,PHID, XID, (U(J+9),
    2 J=1,3),(U(J+6),J=1,3), ((FCOT(J,K),K=1,3),J=1,3)
 160 FORMAT (/5(6F20.5/), 3F20.5)
     IF      (IWILEY(1)) 180,180,190
 180 GO TO   (1955,190),JN
 190 PUNCH 195, TP, (U(J+18), J=1,3), W342,W(1,4,2), (U(J+15),J
    1 =1,3),(U(J+12),J=1,3), (U(J+21),J=1,3), PSID,PHID, XID, (U(J+9),
    2 J=1,3),(U(J+6),J=1,3), ((FCOT(J,K), K=1,3), J=1,3)
 195 FORMAT (5(6E12.5,7X1H1/), 3E12.5,43X1H1)
1955 DO      196 J = 1,3
     U(J+18) = U(J+18) * DEGRAD
 196 U(J+6)  = U(J+6) * DEGRAD
     U(18)   = U(18) + ZERO
 200 IF      (IOPT(2))208,300,205
 205 IOPT(2) = -1
     WRITE   (6,206)
 206 FORMAT(19X1HT,17X3HFA1,17X3HFA2,17X3HFA3,16X4HFAX1,16X4HFAY1/
2061 16X4HFAZ1,16X4HFAX2,16X4HFAY2,16X4HFAZ2,16X4HFAX3,16X4HFAY3/
```

```
2062 16X4HFAZ3,16X4HFBX1,16X4HFBY1,16X4HFBZ1,16X4HFBX2,16X4HFBY2/
2063 16X4HFBZ2,16X4HFBX3,16X4HFBY3,16X4HFBZ3,16X4HMBX1,16X4HMBY1/
2064 16X4HMBZ1,16X4HMBX2,16X4HMBY2,16X4HMBZ2,16X4HMBX3,16X4HMBY3/
2065 16X4HMBZ3,16X4HFCX1,16X4HFCY1,16X4HFCZ1,16X4HFCX2,16X4HFCY2/
2066 16X4HFCZ2,16X4HFCX3,16X4HFCY3,16X4HFCZ3,16X4HFFX1,16X4HFFY1/
2067 16X4HFFZ1,16X4HFFX2,16X4HFFY2,16X4HFFZ2,16X4HFFX3,16X4HFFY3 /
2068 16X4HFFZ3,15X5HPSP11,15X5HPSP12,15X5HPSP21,15X5HPSP22,15X5HPSP31/
2069 15X5HPSP32//)
     DO      207 J= 1,55
 207 ODEAR(J) = 0.0
     LSET    = 1
     GO TO   300
 208 IF      (TP) 209,209,210
 209 WRITE   (6,220) (ODEAR(J), J=1,55)
     IF      (IWILEY(2) + JN - 1) 300,300,2095
2095 PUNCH   290, (ODEAR(J), J= 1,55)
     GO TO   300
 210 WRITE (6,220) TP, (W(J,4,1), J=1,3), ((W(J,L,1),J=1,3),L=1,3),
2101 ((W(J,L,2),J=1,3),L=1,3), ((W(J,L,4),J=1,3),L=1,3),
2102 ((W(J,L,3), J=1,3), L=1,3),((W(J,L,5),J=1,3),L=1,3),
2103 ((PSF(J,K), K=1,2), J=1,3)
 220 FORMAT (/9(6F20.5/), F20.5)
     IF      (IWILEY(2)) 270,270,280
 270 GO TO   (300,280) , JN
 280 PUNCH 290,     TP, (W(J,4,1), J=1,3), ((W(J,L,1),J=1,3),L=1,3),
2101 ((W(J,L,2),J=1,3),L=1,3), ((W(J,L,4),J=1,3),L=1,3),
2102 ((W(J,L,3), J=1,3), L=1,3),((W(J,L,5),J=1,3),L=1,3),
2103 ((PSF(J,K), K=1,2), J=1,3)
 290 FORMAT (9(6F12.5,7X1H2/), F12.5,67X1H2)
 300 IF      (IOPT(3)) 310,400,304
 304 IOPT(3) = -1
     WRITE   (6,305)
 305 FORMAT (19X1HT, 18X2HEK, 18X2HEP,17X3HES1,17X3HES2,17X3HES3/
3051 17X3HEF1,17X3HEF2,17X3HEF3, 17X3HEB1,17X3HEB2,17X3HEB3/
3052 16X4HEFF1,16X4HEFF2,16X4HEFF3,16X4HEBF1,16X4HEBF2,16X4HEBF3///)
     LSET    = 1
     GO TO   400
 310 D59     = 0.0
     D69     = 0.0
 316 DO      330 K=1,3
     A1(K)   = 0.0
     DO      320 J=1,3
 320 A1(K) = A1(K) + U(J+18)*XJ(K,J)
     D59     = D59 + A1(K) * U(K+18)
 330 D69     = D69 + U(K+6)*U(K+6)
     EK      = D59/2.+D69*XIGAM/2.     +XM/2.*(U(13)*U(13)+U(14)*U
    1(14) + U(15) * U(15))
     EP      = -(VZERO + DA(7) * U(18))
     M       = 0
     DO      338 K= 20,24
     DO      338 J = 1,3
     L       = 5*J + K
     M       = M + 1
 338 ODEAR(M) = U(L)
     WRITE   (6,340) TP, EK, EP, (ODEAR(J), J = 1,15)
 340 FORMAT (/3(6F20.5/))
     IF      (IWILEY(3)) 370,370,390
 370 GO TO   (400,390),JN
```

```
390 PUNCH 395,      TP, EK, EP, (OLEAR(J), J = 1,15)
395 FORMAT (3(6E12.5,7X1H3/))
400 IF      (IOPT(5)) 410,500,404
404 IOPT(5) = -1
    WRITE  (6,405)
405 FORMAT (/10X53HTHE FOLLOWING        ARRAYS ARE ORGANIZED AS FOLLOWS
 751- // 16X4HTIME, 12X8HINTERVAL/3X12HALPHA DOT(1), 3X12HALPHA DOT(2)
 752 , 3X12HALPHA DOT(3), 7X8HALPHA(1), 7X8HALPHA(2), 7X8HALPHA(3),
 753 3X12HGAMMA DOT(1), 3X12HGAMMA DOT(2) / 3X12HGAMMA DOT(3) , 7X8HGA
 754MMA(1), 7X8HGAMMA(2), 7X8HGAMMA(3), 10X5HX DOT, 10X5HY DOT, 10X5HZ
 755 DOT, 14X1HX / 14X1HY, 14X1HZ, 8X7HOMEGA X, 8X7HOMEGA Y, 8X7HOMEGA
 756 Z, 12X3HPSI, 12X3HPHI, 13X2HXI /
 75710X5HWORK1,10X5HWORK2,10X5HWORK3,10X5HWORK4,10X5HWORK5  //)
410 DO      420 J= 1,12
420 U(J)    = U(J) * RADDEG
    DO      430 J= 19,24
430 U(J)    = U(J) * RADDEG
    W342    = W(3,4,2) * RADDEG
    WRITE   (6,440) TP, H, W342, (U(J), J= 1,29)
440 FORMAT (3(10XF10.5)/8(5XF10.5)/8(5XF10.5)/8(5XF10.5)/5(5XF10.5)//)
500 GO TO   (3,510), LSET
510 TP      = TP + DTP
    RETURN
    END
```

### 3.4.4 Usage of Computer Program

In this section, details of various options, the method of table and data input and program output are delineated. Following this presentation, a sample of the program output is presented.

#### 3.4.4.1 Input Functions (Tables)

The input tables, consisting of the profiles for the footpads, blocks, and shock absorbers are read in with an F format using 8 fields, 10 columns wide (8F10.X). All the tables use the same input sequence:

1. An identification number which specifies whether the table refers to the footpad, block or shock absorber and secondly, specifies the particular leg set to which the data applies.

2. The number of corresponding points in the table.

3. The independent variable list (units of inches).

4. The dependent variable list or lists (dimensionless).

The order of the input cards is critical within a table and no blank fields are allowed within a table. However, the order in which the tables are introduced does not affect the operation.

When all of the tables have been read, a blank card must follow to indicate that all of the tables have been read.

The identification number mentioned above will always start in the first field. Blank fields after the last table value are permitted (and ignored). The identification number consists of a two digit number. The first digit refers to table type and the second digit to leg set. In regard to the first digit, Number 1 refers to footpad, Number 2 refers to blocks and Number 3 refers to shock absorbers. The second digit is numerically equal to the leg set number 1, 2, or 3. As an example, the block table for leg set Number 3 would have an identification number of 23.0.

In some cases, the table data will be the same for all leg sets. In this situtation two methods may be used to read in the data. The first method is to read in the data for each leg set as described above. The second method is to read in the data for the number one (1) leg set only. The program will then construct tables for the second and third leg set automatically.

The actual table data which is read in is the crush profile for the footpads, the crush profile for the blocks and the spring rate profile and the damping profile during compression for the shock absorber.

Example:

Consider the following situation:

1. Footpad No. 1 - 2 values (2 independent and 2 dependent variables)
2. Footpad No. 2 - 6 values.
3. Footpad No. 3 - 9 values
4. Block No. 1, 2 & 3 - 3 values
5. Shock Absorber No. 1, 2, 3 - 4 values

   (Since there are two sets of dependent variables, this table will have 4 independent variables and 8 dependent variables.)

Using $X_N$ to denote the Nth independent variable, $F_N$ & $G_N$ to denote the Nth dependent variables and B to denote Blank, the cards would be punched in the following manner:

| Card Number | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| (1) | 11.0 | 2.0 | $X_1$ | $X_2$ | $F_1$ | $F_2$ | B | B |
| (2) | 12.0 | 6.0 | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ |
| (3) | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | B | B |
| (4) | 13.0 | 9.0 | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ |
| (5) | $X_7$ | $X_8$ | $X_9$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ |
| (6) | $F_6$ | $F_7$ | $F_8$ | $F_9$ | B | B | B | B |
| (7) | 21.0 | 3.0 | $X_1$ | $X_2$ | $X_3$ | $F_1$ | $F_2$ | $F_3$ |
| (8) | 23.0 | 3.0 | $X_1$ | $X_2$ | $X_3$ | $F_1$ | $F_2$ | $F_3$ |
| (9) | 22.0 | 3.0 | $X_1$ | $X_2$ | $X_3$ | $F_1$ | $F_2$ | $F_3$ |
| (10) | 31.0 | 4.0 | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $F_1$ | $F_2$ |
| (11) | $F_3$ | $F_4$ | $G_1$ | $G_2$ | $G_3$ | $G_4$ | B | B |
| (12) | 32.0 | 4.0 | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $F_1$ | $F_2$ |
| (13) | $F_3$ | $F_4$ | $G_1$ | $G_2$ | $G_3$ | $G_4$ | B | B |
| (14) | 33.0 | 4.0 | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $F_1$ | $F_2$ |
| (15) | $F_3$ | $F_4$ | $G_1$ | $G_2$ | $G_3$ | $G_4$ | B | B |
| (16) | B | B | B | B | B | B | B | B |

Note that blocks No. 2 and No. 3 (cards 9 & 8) have been purposely put out of sequence. This is permissible.

The table data is summarized in the output printing according to a format which is identical regardless of the input option being utilized. The tables are printed using 7 fields, each field 10 columns wide (7F10.3). Each table is identified separately by a note such as "BLOCK NO. 2". The list of independent variables follows this note. The list or lists of dependent variables then follows.

In addition to such machine diagnostics as improper card format which should stop the program, there are several additional programmed diagnostics which will stop the program. Prior to stopping the program, a message will be printed which indicates why the program was terminated. In the following discussion a sample diagnostic will be followed by an explanation of how it might be generated.

1. ** TABLE ERROR NO. 1, FIRST NO. OUT OF RANGE. No. = X X
   This diagnostic will appear if either of the digits in the identification number is not 1, 2 or 3.

2. ** TABLE ERROR NO. 2, LIST EXCEEDS ARRAY LENGTH. LIST IS NO. XX . LIST LENGTH IS X .
   This diagnostic will appear if the total list length exceeds 100. Identification of the table and the calculated list length is provided.

3. ** TABLE ERROR NO. 3, LIST XX LENGTH = 0.
   This diagnostic will appear if a number equal to or less than zero is specified for the list length.

4. ** TABLE ERROR NO. 4, INSUFFICIENT TABLE DATA ( X ), ( X ).

The diagnostic will appear in the event that more or fewer tables than are required have been read in. This diagnostic is only a function of the identification numbers. Two numbers are printed out in the diagnostic statement. The first is the sum of the first digits of the identification number which should be 18 or 6 depending on the input option which is used and the second is the sum of the second digits which should be 18 or 3 depending on the option which is used. Hopefully, printing out these sums will give some insight into the error in the table data. Table error No. 4 will appear if the wrong number of tables is read in. For example, if individual tables are read in for each of the three footpads, while reading in only one table for the blocks and one table for the shock absorbers, then the diagnostic message will appear.

### 3.4.4.2 Input Constants

The input constants used in the program describe the spacecraft geometry and touchdown configuration, nominal values of parameters for elements which generate forces, allowable truncation errors and various codes for selection of computation mode and output options. These constants are read in on a set of 21 data cards. Six fields are used on the data cards. The first field is in columns 1 - 7 and contains the card number in integer (I) format. The next five columns are not used (8 - 12). The remaining 5 fields are 12 columns wide and use floating point (E) format. The final columns on the card (73-80) are not used. A blank card follows the input data.

When runs are to be stacked, it is necessary only to read in data cards containing new data. This card or cards must also be followed by a blank card in order to execute the program. The input summary printout will specify the cards which have been replaced, and will also list the data from all 21 cards.

The input constants found on the 21 cards are defined below. The units are combinations of feet, pounds, seconds, and degrees.

| Card No. | Parameter | Description |
|---|---|---|
| 1 | XM | Mass of spacecraft. |
| | IXX | |
| | IYY | moments and products of inertia about spacecraft axes |
| | IZZ | The inertia matrix is |
| | IXY | $J = \begin{bmatrix} IXX & -IXY & -IXZ \\ -IXY & IYY & -IYZ \\ -IXZ & -IYZ & IZZ \end{bmatrix}$ |
| 2 | IXZ | |
| | IYZ | |
| | XMP | mass of footpad (no longer used in calculations) |
| | XLDO | original length of shock absorber link |
| | DELCG | distance from spacecraft CG to lower hardpoint plane |
| 3 | XLP | distance from spacecraft CL to lower hardpoints |
| | XLL | length of lower link |
| | XLA | length of link joining upper and lower hardpoints |
| | G | gravitational acceleration |
| | ALPHA (1) | lower limit on footpad angle relative to lower link |
| 4 | ALPHA (2) | upper limit on footpad angle relative to lower link |
| | ALPHK (1) | spring rate of footpad lower limit stop |
| | ALPHK (2) | spring rate of footpad upper limit stop |
| | BETA | angle between XLA and spacecraft CL |
| | XLC | distance from spacecraft CL to block CL |

| Card No. | Parameter | Description |
|---|---|---|
| 5 | HBO | thickness of uncrushed block |
| | XIALPH | moment of inertia of footpad about pivot |
| | XIGAM | moment of inertia of lower link about lower hardpoint |
| | HFO | thickness of uncrushed footpad |
| | DB | diameter of block |
| 6 | DF | diameter of footpad |
| | XDOTOC | velocity of spacecraft CG in ground coordinates |
| | YDOTOC | (PHASE NO. = 1) or in vehicle coordinates (PHASE |
| | ZDOTOC | NO. = 3). See Card 11 below. |
| | ——— | |
| 7 | XNU | mechanical friction coefficient for shock absorber |
| | XLDMIN | minimum length permitted for shock absorber link |
| | XKS | spring rate of shock absorber stop |
| | ——— | |
| | ——— | |
| 8 | THETD (1) | angular orientation of leg sets No. 1, 2, 3 with re- |
| | THETD (2) | spect yz plane (vehicle coordinates). Angles increase |
| | THETD (3) | in left-handed sense. |
| | ——— | |
| | ——— | |
| 9 | THETAS | ground slope |
| | PSIO | pitch angle } relative to ground coordinates |
| | PHIO | yaw angle } relative to ground coordinates |
| | XIO | roll angle } relative to ground coordinates |

| Card No. | Parameter | Description |
|---|---|---|
| 9 (cont'd) | XLAMDD | cross slope angle, measured between CG velocity vector and YZ plane (ground coordinates). Applicable only for PHASE NO.= 2 (see Card 11, below) |
| 10 | OMEGA (1) | angular rates about spacecraft x, y, z-axes |
| | OMEGA (2) | |
| | OMEGA (3) | |
| | CODE2 | for selection of punch options (see next section) |
| 11 | CODE | for selection of print options (see next section) |
| | SER NO | serial number (arbitrary) |
| | PHASE NO | for selection of method of specifying spacecraft CG velocities. PHASE NO=1 or 3 denotes that XDOTOC, YDOTOC, ZDOTOC (Card 6) are applicable; PHASE NO=2 denotes that XLAMDD (Card 9), VH and VV (this card) are to be used. |
| | VH | horizontal velocity of spacecraft CG — Applicable only if Phase No. = 2 |
| | VV | vertical velocity of spacecraft CG — Applicable only if Phase No. = 2 |
| 12 | DTP | print/punch interval |
| | TF | final time |
| | HM | minimum integration interval |
| | ERR (4) | allowable truncation error (per second)-#1 footpad angle |
| | ERR (5) | allowable truncation error (per second)-#2 footpad angle |
| 13 | ERR (6) | allowable truncation error (per second)-#3 footpad angle |
| | ERR (7) | allowable truncation error (per second)-#1 lower link angular rate |

| Card No. | Parameter | Description |
|---|---|---|
| 13 (cont'd) | ERR (8) | allowable truncation error (per second)-#2 lower link angular rate |
| | ERR (9) | allowable truncation error (per second)-#3 lower link angular rate |
| | ERR (10) | allowable truncation error (per second)-#1 lower link angle |
| 14 | ERR (11) | allowable truncation error (per second)-#2 lower link angle |
| | ERR (12) | allowable truncation error (per second)-#3 lower link angle |
| | ERR (13) | allowable truncation error (per second)-CG velocity (X-direction) |
| | ERR (14) | allowable truncation error (per second)-CG velocity (Y-direction) |
| | ERR (15) | allowable truncation error (per second)-CG velocity (Z-direction) |
| 15 | ERR (16) | allowable truncation error (per second)-CG position (X-direction) |
| | ERR (17) | allowable truncation error (per second)-CG position (Y-direction) |
| | ERR (18) | allowable truncation error (per second)-CG position (Z-direction) |
| | ERR (19) | allowable truncation error (per second)-spacecraft angular rate (x-axis) |

| Card No. | Parameter | Description |
|---|---|---|
| 15 (cont'd) | ERR (20) | allowable truncation error (per second)-spacecraft angular rate (y-axis) |
| 16 | ERR (21) | allowable truncation error (per second)-spacecraft angular rate (z-axis) |
| | ERR (22) | allowable truncation error (per second)-pitch angle |
| | ERR (23) | allowable truncation error (per second)-yaw angle |
| | ERR (24) | allowable truncation error (per second)-roll angle |
| 17 | RC (1), RC (2), RC (3) | nominal damping coefficient for shock absorbers #1, 2 & 3 during stroking |
| | XKD (1), XKD (2), XKD (3) | nominal spring rate for shock absorbers #1, 2 & 3 |
| 18 | FP (1), FP (2), FP (3) | preload for shock absorbers #1, 2 & 3 |
| 19 | RR (1), RR (2), RR (3) | nominal damping coefficient for shock absorbers #1, 2 & 3 during rebound |
| | SRL (1), SRL (2), SRL (3) | slope of profile for rebound damping in shock absorbers #1, 2 & 3 |
| 20 | CB (1), CB (2), CB (3) | nominal crush pressure for blocks #1, 2 & 3 |

| Card No. | Parameter | Description |
|---|---|---|
| 20 (cont'd) | CF (1) | nominal crush |
| | CF (2) | pressure for |
| | CF (3) | footpads #1, 2 & 3 |
| 21 | XMUF | nominal sliding friction coefficient for footpads |
| | XMUB | nominal sliding friction coefficient for blocks |
| | TAU (1) | impact times for footpads #1, 2 & 3. These |
| | TAU (2) | are specified if CODE=4 is specified |
| | TAU (3) | (see following section) |

### 3.4.4.3 Output

At the completion of a run, the program has the capability of providing several output options. In the time history mode (i.e.,CODE≠4), the program will predict the spacecraft dynamics which result from a specific set of touchdown conditions. Computation will continue until one of the following decisions is made:

1. Vehicle stable (motion has subsided, or vehicle is "rocking back" from a critical attitude).

2. Vehicle unstable (toppling).

3. Time expired (problem time exceeded TF).

The standard output for the time history mode contains a statement of this decision, together with the time at which the decision was made and the minimum value of stability angle which was encountered in the run. After this printout, a statement regarding the energy of the system is printed. The initial kinetic energy of the system, the difference between the initial and final potential energy, the dissipated energy of the system (including any stored energy in the shock absorber) and the final kinetic energy are printed. From an energy balance point of view, the original kinetic energy plus the difference in potential energy minus the "dissipated" energy should equal the final kinetic energy.

Following the energy printout, the maximum shock absorber force for three leg sets are printed. The minimum ground clearance is printed after this shock absorber force printing.

The printing above pertains to the dynamics of the vehicle. The output which will be described now pertains to details concerning the overall integration process. The arithmetic sum of the truncation errors accepted during the integration process is first printed. This array is 29 long and reads from left to right; 1 - 8 in the first row, 9 - 16 in the second row, 17 - 24 in the third row and 25 - 29 in the fourth row. The numbers refer to the number assigned to each of the integrated variables. Footpad angular velocity for the three footpads are 1 - 3, footpad angular position are 4 - 6, lower leg angular velocities are 7 - 9, lower leg angular positions are 10 - 12, CG velocities are 13 - 15, CG positions are 16 - 18, CG angular velocities are 19 - 21, CG angular positions are 22 - 24 and 25 - 29 are energies. The units are radians, feet and seconds. The arithmetic sum would be of most interest to one who is attempting to determine the total truncation error of the integrated variables. Since the sign of the truncation error may be positive or negative, some cancellation of errors is possible. The next array is similar to the arithmetic sum array but this array contains the sum of the absolute values of the truncation error. These values give an indication of how closely the allowable and actual error agree. If one variable controlled the size of the integration interval, then the absolute sum should approach but always be less than the allowable value. In general, various integrated values control the integration interval and these numbers will be significantly less than the allowable values.

The next group of printout relates to a summary of the integration process. The first number is simply the total number of integration intervals. The next number is the average interval size in seconds. The next array describes how the intervals were distributed. The first number indicates the number of intervals between .1 and 1 second, the second indicates the number of intervals between .01 and .1, and so on. The next array describes the distribution of the number of times the integration routine had to halve an interval in order to pass on error check. The first number is for zero cuts, the second is 1 cut, the third is for 2 cuts, and so on. This is one of the best indicators of the "smoothness" of the program in general and secondly, the efficiency of the integration method. Ideally, the number for zero cuts would be equal to the number of integration intervals. Practically, this zero cut number has been greater than half of the total integration intervals. The final array shows which of the integrated variables was instrumental in determining the next interval. The array is numbered in the same sequence as above and 1 - 10 are in the first row, 11 - 20 in the second, and so on.

Finally, output printing at the final time reached in the integration is performed and the program is finished.

In addition to the standard output for the program in the time history mode, three other basic options for printing/punching are available. These options provide detailed time histories of the vehicle motion (option 1), loads (option 2) and energy (option 3).

Output option 1, time history of vehicle motion, is in the following format:

| | | | | | |
|---|---|---|---|---|---|
| T | OMEGAX | OMEGAY | OMEGAZ | SIGMA | MC |
| X | Y | Z | XDOT | YDOT | ZDOT |
| PSI | PHI | XI | PSIDOT | PHIDOT | XIDOT |
| GAMMA1 | GAMMA2 | GAMMA3 | GAMDOT1 | GAMDOT2 | GAMDOT3 |
| XP1 | YP1 | ZP1 | XP2 | YP2 | ZP2 |
| XP3 | YP3 | ZP3 | | | |

The symbolism is:

| | |
|---|---|
| T | time (sec) |
| OMEGA | vehicle angular velocity in vehicle coordinate system (deg/sec) |
| SIGMA | stability angle (deg) |
| MC | ground clearance (ft) |
| X, Y, Z | location of vehicle CG in ground coordinates; origin at vehicle CG at T = 0 sec. (ft) |
| XDOT, YDOT, ZDOT | velocity of CG in ground coordinates (ft/sec) |
| PSI, PHI, XI | vehicle orientation in ground coordinates (deg) |
| PSIDOT, PHIDOT, XIDOT | vehicle angular rates in ground coordinates (deg/sec) |
| GAMMA | lower strut angular orientation in vehicle coordinates (deg) |
| GAMDOT | lower strut angular rates in vehicle coordinates (deg/sec) |
| XPi, YPi, ZPi | location of the ith footpad pivot in surface coordinates (ft) |

Output option 2, time history of loads, torques and deformations, is in the following format:

| | | | | | |
|---|---|---|---|---|---|
| T | FA1 | FA2 | FA3 | FAX1 | FAY1 |
| FAZ1 | FAX2 | FAY2 | FAZ2 | FAX3 | FAY3 |
| FAZ3 | FBX1 | FBY1 | FBZ1 | FBX2 | FBY2 |
| FBZ2 | FBX3 | FBY3 | FBZ3 | MBX1 | MBY1 |
| MBZ1 | MBX2 | MBY2 | MBZ2 | MBX3 | MBY3 |
| MBZ3 | FCX1 | FCY1 | FCZ1 | FCX2 | FCY2 |
| FCZ2 | FCX3 | FCY3 | FCZ3 | FFX1 | FFY1 |
| FFZ1 | FFX2 | FFY2 | FFZ2 | FFX3 | FFY3 |
| FFZ3 | PSP11 | PSP12 | PSP21 | PSP22 | PSP31 |
| PSP32 | | | | | |

The symbolism is:

| | |
|---|---|
| T | time (sec) |
| FAi | force in ith shock absorber (lb) |
| FAXi, FAYi, FAZi | force components of ith upper hardpoint in the vehicle coordinate system (lb) |

| | |
|---|---|
| FBXi, FBYi, FBZi | force components of ith lower hardpoint in the vehicle coordinate system (lb) |
| MBXi, MBYi, MBZi | moment components of ith lower hardpoint in the vehicle coordinate system (ft-lb) |
| FCXi, FCYi, FCZi | force components of ith body block in vehicle coordinates (lb) |
| FFXi, FFYi, FFZi | force components of ith footpad pivot in the vehicle coordinate system (lb) |
| PSPij | deformation of the jth segment of the ith footpad along the centerline of the footpad (ft) |

Output option 3, time history of energies and energy states of various elements, is in the following format:

| | | | | | |
|---|---|---|---|---|---|
| T | EK | EP | ES1 | ES2 | ES3 |
| EF1 | EF2 | EF3 | EB1 | EB2 | EB3 |
| EFF1 | EFF2 | EFF3 | EBF1 | EBF2 | EBF3 |

The symbolism is:

| | |
|---|---|
| T | time (sec) |
| EK | total kinetic energy (ft-lb) |
| EP | potential energy (ft-lb) |
| ESi | total energy stored and dissipated in ith shock absorber (ft-lb) |
| EFi | total energy, excluding friction, stored and dissipated in the ithfootpad (ft-lb) |
| EBi | total energy, excluding friction, stored and dissipated in the ith body block (ft-lb) |
| EFFi | total energy dissipated by friction by the ith footpad (ft-lb) |
| EBFi | total energy dissipated by friction by the ith body block (ft-lb) |

The output option is selected by assigning the appropriate numbers to CODE and CODE 2 on input cards 11 and 10. The code numbers are:

| Code No. | Output Options Specified | Code No. | Output Options Specified |
|---|---|---|---|
| 1 | 1 - (motion time history) | 5 | 1 & 2 |
| 2 | 2 - (load time history) | 6 | 1 & 3 |
| 3 | 3 - (energy time history) | 7 | 2 & 3 |
| 4 | 4 - (surface determination, to be discussed later) | 8 | 1, 2 & 3 |
| | | 9 | No output |

To select a printing option, set the appropriate code number in CODE. If it is desired to punch all the options which are printed, insert a minus sign in the value of CODE. Punch options may be specified independently of CODE by inserting one of the above code numbers in CODE2 (always positive).

To illustrate the use of the print/punch options, consider the following example. Suppose option 2 and 3 have been selected to be printed and option 2 is to be punched. Code number 7 would appear in CODE and code number 2 would appear in CODE2. It should be noted that data cannot be punched which is not printed. That is, option 2 and 3 could not be punched if only option 1 were to be printed. No error will result from placing any combination of numbers between 1 and 10 in these coded locations.

A code number of 10 in CODE enables diagnostic type printing. This option bears a resemblance to option 1, and has been retained in the program primarily for historical interest. In CODE2 a code number of 10 is ignored.

An error ("out of range computed go to") will be generated if a number or numbers beyond the range of 1 to 10 are put in CODE and/or CODE2, with the exception of negative numbers being permitted in CODE.

If CODE = 4 a very special output printing results. Under these circumstances internal modifications are made to the program which permit the specification of footpad impact times and calculate the ground slope and cross slope angle from the resulting motions. The input data required for the CODE = 4 version is essentially the same as for the time history version, with the addition of TAU (1), TAU (2), TAU (3), the times at which the respective footpads strike the surface. Any times greater than or equal to zero may be specified, but it is necessary to specify the spacecraft attitude and motion at zero time. The spacecraft attitude can be specified in any convenient coordinate system, provided the z-axis is positive vertically downward.

The output of the CODE = 4 version consists of statements of the value of the cross slope angle LAMDA, ground slope THETAS, and coordinates of the impact points in the convenient coordinate system. The values of 4 angles are then printed. ZETA is the required rotation between the convenient coordinate system and the customary ground coordinate system, while PSI, PHI, XI are the touchdown pitch, yaw, roll angle specifications relative to the customary ground coordinate system. Due to an ambiguity in the method of determination, it is possible that the roll angle XI will be in error. This can be checked by means of the approximation

$$XI \simeq XIO - ZETA$$

which is exact only for a level vehicle, but should be only a few degrees off for a moderately tipped vehicle. If the printed data satisfies the above approximation, the correct value of XI has been printed. Otherwise the supplement of the printed value of XI must be chosen.

The ambiguity which exists in the determination of the roll angle XI results from the nature of the inverse sine function evaluation. The output of this evaluation is always an angle in the first or fourth quadrant. Consider the angle in the second quadrant, $\Xi = 90° + \Delta$ . If the sine of this angle is computed, a number N between 0 and 1.0 will result. Now if we take the inverse sine of N, the angle we have in mind is $\Xi = 90° + \Delta$ . But the computer will calculate that the inverse sine of N is $\Xi_{CALC} = 90° - \Delta$. Note that the sines of $\Xi$ and $\Xi_{CALC}$ are numerically equal.

If $|\Xi| \leq 90°$, there is no problem and the above approximation will be satisfied. Otherwise the approximation will reveal a discrepancy, and an adjustment must be applied. Suppose

that $\Xi_{TRUE} = 90° + \Delta$

then $\Xi_{CALC} = 90° - \Delta$

so $\Xi_{TRUE} + \Xi_{CALC} = 180°$

and $\Xi_{TRUE}$ is seen to be the supplement of $\Xi_{CALC}$. A similar result is obtained for

$\Xi_{TRUE} = -(90° + \Delta)$ which is in the third quadrant.

in which case

$$\Xi_{TRUE} + \Xi_{CALC} = -180°$$

where $\Xi_{TRUE}$ and $\Xi_{CALC}$ will be negative.

To summarize:

If $|\Xi_{TRUE}| \leq 90°$, $\Xi_{TRUE} = \Xi_{CALC}$

If $|\Xi_{TRUE}| > 90°$, $\Xi_{TRUE} =$ supplement of $\Xi_{CALC}$

### 3.4.5 Sample Program Output

A sample of the program output follows. This sample presents options 1, 2 & 3 with no punching. The program was terminated by the decision that the vehicle was stable.

REAL TIME CLOCK INTERROGATED AT 11:25:30

| | | | | | | |
|---|---|---|---|---|---|---|
| FOOT PAD NO.1 | | | | | | |
| .000 | .100 | .250 | .500 | .750 | 1.000 | 1.250 |
| 1.500 | 1.750 | 2.000 | 2.490 | 2.666 | 3.500 | 4.500 |
| .000 | .466 | .502 | .563 | .629 | .697 | .767 |
| .841 | .920 | 1.000 | 1.000 | 1.623 | 1.823 | 6.503 |
| FOOT PAD NO.2 | | | | | | |
| .000 | .100 | .250 | .500 | .750 | 1.000 | 1.250 |
| 1.500 | 1.750 | 2.000 | 2.490 | 2.666 | 3.500 | 4.500 |
| .000 | .468 | .502 | .563 | .629 | .697 | .767 |
| .841 | .920 | 1.000 | 1.000 | 1.623 | 1.823 | 6.503 |
| FOOT PAD NO.3 | | | | | | |
| .000 | .100 | .250 | .500 | .750 | 1.000 | 1.250 |
| 1.500 | 1.750 | 2.000 | 2.490 | 2.666 | 3.500 | 4.500 |
| .000 | .466 | .502 | .563 | .629 | .697 | .767 |
| .841 | .920 | 1.000 | 1.000 | 1.623 | 1.823 | 6.503 |
| BLOCK NO.1 | | | | | | |
| .000 | .100 | 5.750 | 7.750 | | | |
| .000 | 1.000 | 1.000 | 21.000 | | | |
| BLOCK NO.2 | | | | | | |
| .000 | .100 | 5.750 | 7.750 | | | |
| .000 | 1.000 | 1.000 | 21.000 | | | |
| BLOCK NO.3 | | | | | | |
| .000 | .100 | 5.750 | 7.750 | | | |
| .000 | 1.000 | 1.000 | 21.000 | | | |
| SHOCK ABSORBER NO.1 | | | | | | |
| .000 | .250 | .500 | .750 | 1.000 | 1.250 | 1.500 |
| 1.750 | 2.000 | 2.250 | 2.500 | 2.750 | 3.000 | 3.250 |
| 3.500 | 3.750 | 4.000 | 4.250 | 4.500 | 6.500 | |
| 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 1.005 | 1.012 | 1.020 | 1.028 | 1.036 | 1.044 | 1.052 |
| 1.062 | 1.076 | 1.096 | 1.114 | 1.151 | 1.447 | |
| 1.000 | 1.060 | 1.160 | 1.240 | 1.360 | 1.480 | 1.600 |
| 1.740 | 1.890 | 2.130 | 2.770 | 3.610 | 4.960 | 6.090 |
| 7.240 | 8.370 | 9.520 | 10.670 | 11.600 | 20.840 | |
| SHOCK ABSORBER NO.2 | | | | | | |
| .000 | .250 | .500 | .750 | 1.000 | 1.250 | 1.500 |
| 1.750 | 2.000 | 2.250 | 2.500 | 2.750 | 3.000 | 3.250 |
| 3.500 | 3.750 | 4.000 | 4.250 | 4.500 | 6.500 | |
| 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 1.005 | 1.012 | 1.020 | 1.028 | 1.036 | 1.044 | 1.052 |
| 1.062 | 1.076 | 1.096 | 1.114 | 1.151 | 1.447 | |
| 1.000 | 1.080 | 1.160 | 1.240 | 1.360 | 1.480 | 1.600 |
| 1.740 | 1.890 | 2.130 | 2.770 | 3.610 | 4.960 | 6.090 |
| 7.240 | 8.370 | 9.520 | 10.670 | 11.600 | 20.840 | |
| SHOCK ABSORBER NO.3 | | | | | | |
| .000 | .250 | .500 | .750 | 1.000 | 1.250 | 1.500 |
| 1.750 | 2.000 | 2.250 | 2.500 | 2.750 | 3.000 | 3.250 |
| 3.500 | 3.750 | 4.000 | 4.250 | 4.500 | 6.500 | |
| 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 1.005 | 1.012 | 1.020 | 1.028 | 1.036 | 1.044 | 1.052 |
| 1.062 | 1.076 | 1.096 | 1.114 | 1.151 | 1.447 | |
| 1.000 | 1.080 | 1.160 | 1.240 | 1.360 | 1.480 | 1.600 |
| 1.740 | 1.890 | 2.130 | 2.770 | 3.610 | 4.960 | 6.090 |
| 7.240 | 8.370 | 9.520 | 10.670 | 11.600 | 20.840 | |

```
NEW DATA CARD FOR THIS CASE IS NO. =    1
NEW DATA CARD FOR THIS CASE IS NO. =    2
NEW DATA CARD FOR THIS CASE IS NO. =    3
NEW DATA CARD FOR THIS CASE IS NO. =    4
NEW DATA CARD FOR THIS CASE IS NO. =    5
NEW DATA CARD FOR THIS CASE IS NO. =    6
NEW DATA CARD FOR THIS CASE IS NO. =    7
NEW DATA CARD FOR THIS CASE IS NO. =    8
NEW DATA CARD FOR THIS CASE IS NO. =    9
NEW DATA CARD FOR THIS CASE IS NO. =   10
NEW DATA CARD FOR THIS CASE IS NO. =   11
NEW DATA CARD FOR THIS CASE IS NO. =   12
NEW DATA CARD FOR THIS CASE IS NO. =   13
NEW DATA CARD FOR THIS CASE IS NO. =   14
NEW DATA CARD FOR THIS CASE IS NO. =   15
NEW DATA CARD FOR THIS CASE IS NO. =   16
NEW DATA CARD FOR THIS CASE IS NO. =   17
NEW DATA CARD FOR THIS CASE IS NO. =   18
NEW DATA CARD FOR THIS CASE IS NO. =   19
NEW DATA CARD FOR THIS CASE IS NO. =   20
NEW DATA CARD FOR THIS CASE IS NO. =   21

THIS CASE IS SERIES NO. =        1.000     AND PHASE NO. =   1

INPUT DATA FOLLOWS.

CARD  1       XM   =  .20000+02     IXX  =  .14222+03     IYY  =  .14222+03     IZZ  =  .14222+03     IXY  =  .00000
CARD  2      IXZ   =  .00000        IYZ  =  .00000        XMP  =  .32400-01    XLLO  =  .31275+01   DELCG  =  .15417+01
CARD  3      XLP   =  .31875+01     XLL  =  .33750+01     XLA  =  .95833-00       G  =  .53100+01 ALFHA(1)=  .75000-00
CARD  4 ALPHA(2)=  .45000+02 ALPHK(1)=  .14300+05 ALPHK(2)=  .14300+05    BETA  =  .41000+02     XLC  =  .24167+01
CARD  5      HEO   =  .66667-00  XIALPH =  .37300-02   XIGAM  =  .13470+01     HFO  =  .54167-00      DE  =  .58333-00
CARD  6       DF   =  .10000+01  XDOTOC =  .00000    YDOTOC  = -.92769+01  ZDOTOC  =  .14246+02
CARD  7      XNU   =  .50000-01  XLDMIN =  .27525+01     XKS  =  .10000+06
CARD  8 THETD(1)=  .00000     THETD(2)=  .12000+03 THETD(3)=  .24000+03
CARD  9   THETAS   =  .15000+02    PSIO  =  .50000+01    PHIC  =  .00000        XIO  =  .00000   XLAMDE  =  .00000
CARD 10 OMEGA(1)=  .00000     OMEGA(2)=  .00000   OMEGA(3)=  .00000      CODE2  =  .40000+01
CARD 11     CODE   =    6.00000  SER NO  =    1.00000 PHASE NO=          1       VH  = -.00000       VV  = -.00000
CARD 12      LTF   =  .25000+01      TF  =  .60000+01      HM  =  .10000-05  ERR(4)  =  .60000+01  ERR(5)  =  .60000+01
CARD 13   ERR(6)   =  .60000+01  ERR(7)  =  .20000+01  ERR(8)  =  .20000+01  ERR(9)  =  .20000+01 ERR(10)  =  .40000+01
CARD 14  ERR(11)   =  .40000+01 ERR(12)  =  .40000+01 ERR(13)  =  .16667-00 ERR(14)  =  .16667-00 ERR(15)  =  .16667-00
CARD 15  ERR(16)   =  .16667-00 ERR(17)  =  .16667-00 ERR(18)  =  .16667-00 ERR(19)  =  .16667-00 ERR(20)  =  .16667-00
```

```
CARD 16 ERR(21) =   .16667-00 ERR(22) =   .66667-00 ERR(23) =   .66667-00 ERR(24) =   .66667-00   FC(1) =   .5400C+02
CARD 17   RC(2) =   .54000+02   RC(3) =   .54000+02  XKD(1) =   .36000+04  XKD(2) =   .36000+04  XKD(3) =   .36000+04
CARD 18   FP(1) =   .16500+03   FP(2) =   .16500+03   FP(3) =   .16500+03   RR(1) =   .86400+03   RR(2) =   .86400+03
CARD 19   RR(3) =   .86400+03  SRL(1) =   .17760+01  SRL(2) =   .17760+01  SPL(3) =   .17760+01   CE(1) =   .54256+04
CARD 20   CB(2) =   .54256+04   CB(3) =   .54256+04   CF(1) =   .15788+04   CF(2) =   .15788+04   CF(3) =   .15788+04
CARD 21   XNUF =    .80000-00    XNUB =   .80000-00  TAU(1) =   .10000+00  TAU(2) =   .10000+00  TAU(3) =   .10000+00


THE FOLLOWING ARRAYS ARE ORGANIZED AS FOLLOWS-

        T        OMEGAX        OMEGAY        OMEGAZ         SIGMA            MC
        X             Y             Z          XDOT          YDOT          ZDOT
      PSI           PHI            XI        PSIDOT        PHIDOT         XIDOT
   GAMMA1        GAMMA2        GAMMA3       GAMDOT1       GAMDOT2       GAMDOT3
      XP1           YP1           ZP1           XP2           YP2           ZP2
      XP3           YP3           ZP3

        T           FA1           FA2           FA3          FAX1          FAY1
     FAZ1          FAX2          FAY2          FAZ2          FAX3          FAY3
     FAZ3          FBX1          FBY1          FBZ1          FBX2          FBY2
     FBZ2          FBX3          FBY3          FBZ3          MBX1          MBY1
     MBZ1          MBX2          MBY2          MBZ2          MBX3          MBY3
     MBZ3          FCX1          FCY1          FCZ1          FCX2          FCY2
     FCZ2          FCX3          FCY3          FCZ3          FFX1          FFY1
     FFZ1          FFX2          FFY2          FFZ2          FFX3          FFY3
     FFZ3         FSP11         FSP12         FSP21         FSP22         FSP31
    FSP32

        T            EK            EF           ES1           ES2           ES3
      EF1           EF2           EF3           EB1           EB2           EB3
     EFF1          EFF2          EFF3          EBF1          EBF2          EBF3


   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000       -.00000        .00000      -9.27689      14.24567
  5.00000        .00000        .00000        .00000        .00000        .00000
 17.99779      17.99779      17.99779        .00000        .00000        .00000
   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000        .00000

   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000        .00000        .00000        .00000        .00000
   .00000        .00000        .00000        .00000        .00000        .00000
```

| | | | | | |
|---|---|---|---|---|---|
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| .00000 | 2669.99800 | -.00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 142.22000 | -.00000 | -.00000 | .00703 | -.00000 | -.00000 | 1.00000 | .00000 | .00000 |
| -.00000 | 142.22000 | -.00000 | -.00000 | .00703 | -.00000 | .00000 | 1.00000 | .00000 |
| -.00000 | -.00000 | 142.22000 | .00000 | .00000 | .00703 | .00000 | .00000 | 1.00000 |

FOOTPAD NO. 1 HAS INITIALLY CONTACTED SURFACE AT T = .0008 SECONDS.

FOOTPAD NO. 2 HAS INITIALLY CONTACTED SURFACE AT T = .2098 SECONDS.

FOOTPAD NO. 3 HAS INITIALLY CONTACTED SURFACE AT T = .2098 SECONDS.

BODY BLOCK NO. 2 HAS INITIALLY CONTACTED SURFACE AT T = .2861 SECONDS.

BODY BLOCK NO. 3 HAS INITIALLY CONTACTED SURFACE AT T = .2861 SECONDS.

| | | | | | |
|---|---|---|---|---|---|
| 2.50000 | 2.51278 | -.01172 | -.05077 | 16.11791 | 1.95151 |
| -.00002 | -5.79831 | 2.64008 | -.00135 | -.10675 | -.20005 |
| -35.64609 | -.00005 | -.01154 | 7.51278 | -.01178 | -.05076 |
| 18.00984 | 16.99278 | 16.99394 | .05730 | 6.02396 | 6.03005 |
| .00015 | 1.95474 | -3.79192 | 5.55538 | -7.05337 | -.45895 |

| | | |
|---|---|---|
| -5.55560 | -7.05303 | -.45899 |

| | | | | | |
|---|---|---|---|---|---|
| 2.50000 | .00000 | 420.05303 | 222.25428 | .00000 | .00000 |
| .00000 | -159.18938 | 91.90800 | -120.97522 | 160.77855 | 92.62557 |
| -122.19098 | .00000 | .00000 | .00000 | 217.62328 | -37.65703 |
| 63.42710 | -212.12048 | -30.69845 | 60.91192 | -.00000 | -.00000 |
| -.00000 | -64.96735 | 37.52048 | 245.37101 | -67.79320 | -39.14043 |
| -255.93843 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | 57.31434 | 54.69735 | -53.32101 | -47.63418 | 63.95230 |
| -48.03634 | .01204 | .02424 | .10692 | .06645 | .10692 |
| .08645 | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 2.50000 | .06591 | -301.01688 | 225.40839 | 664.92071 | 664.90738 |
| 6.13150 | 54.64150 | 54.63699 | .00000 | 168.54622 | 168.54624 |
| 168.02017 | 417.79952 | 417.77364 | .00000 | 89.75712 | 89.75728 |

$$$$ VEHICLE IS STABLE (ROCK-BACK) $$$$$

TIME = 2.6207 SIGMIN = 14.5642 DEGREES

| | | |
|---|---|---|
| INITIAL KINETIC ENERGY (EKO) | = | 2889.99600 FT LBS |
| DIFFERENCE IN POTENTIAL ENERGY (EPD) | = | 304.54803 FT LBS |
| DISSIPATED ENERGY (EDIS) | = | 3188.21390 FT LBS |
| FINAL KINETIC ENERGY (EKF) | = | 6.25950 FT LBS |

MAXIMUM SHOCK ABSORBER FORCES-

| | |
|---|---|
| LEG NO.1 FA(1) = | 1318.7941000000 |
| LEG NO.2 FA(2) = | 3045.9441000000 |
| LEG NO.3 FA(3) = | 3045.9464000000 |

MINIMUM CLEARANCE BETWEEN GROUND PLANE AND FRAME = .4957 FT.

ARITHMETIC SUM OF ERRORS FOLLOWS-

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -.6246487-04 | -.1542510-03 | -.1436707-03 | -.2524913-04 | -.6217573-05 | -.8217679-05 | -.9804237-03 | -.2172276-01 |
| -.2012520-01 | -.1379264-05 | -.1300924-04 | -.1241369-04 | -.5605817-05 | .9783995-04 | .3195027-04 | .4604700-07 |
| -.1761401-06 | -.5409117-06 | -.9752216-04 | .1641970-04 | .1387546-04 | .3093191-07 | .2047133-08 | -.1660563-06 |
| .1009591-02 | -.1520434-02 | .0000000 | .2971095-02 | .0000000 | | | |

SUM OF THE ABSOLUTE ERRORS FOLLOWS-

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .1111262-03 | .5123239-03 | .5096816-03 | .2525162-04 | .6217573-05 | .8217679-05 | .3104477-02 | .3645563-01 |
| .3513201-01 | .6611555-05 | .5396876-04 | .5114786-04 | .1158025-03 | .8401766-03 | .1334466-02 | .1173362-06 |
| .2370674-05 | .4080926-05 | .3347769-03 | .6485635-04 | .1442387-03 | .7129081-06 | .1210270-06 | .2672984-06 |
| .6339898-02 | .2556933-02 | .0000000 | .3858704-02 | .0000000 | | | |

NO. OF INTERVALS = 537
AVG. INTERVAL = .00525269
INTERVAL DISTRIBUTION

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 73 | 350 | 110 | 4 | 0 | 0 | 0 |

CUT DISTRIBUTION

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 289 | 159 | 46 | 36 | 4 | 1 | 0 | 0 |

CRITICAL VARIABLE DISTRIBUTION

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 79 | 204 | 172 | 0 |
| 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 39 | 2 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| | | | | | |
|---|---|---|---|---|---|
| 2.62069 | 6.01231 | .30220 | .57477 | 85.49601 | 1.96080 |
| -.00003 | -5.61408 | 2.66708 | -.00207 | .00465 | .73890 |
| -33.75453 | -.00319 | -.00122 | 6.01232 | .30207 | .57510 |
| 18.00984 | 17.91008 | 17.90507 | .05730 | -5.76435 | -1.58231 |
| -.00004 | 1.92276 | -3.54207 | 5.54141 | -7.16612 | -.46158 |
| -5.54197 | -7.16623 | -.46250 | | | |
| | | | | | |
| 2.62069 | .00000 | 152.67363 | 157.22436 | .00000 | .00000 |
| .00000 | -109.23920 | 63.06927 | -66.01339 | 112.50230 | 64.95325 |
| -88.56529 | .00000 | .00000 | .00000 | 85.72577 | 76.69100 |
| 11.60074 | -115.64415 | 31.96154 | 27.19091 | -.00000 | -.00000 |
| -.00000 | -98.22428 | 56.70981 | 350.94483 | -76.64657 | -44.36740 |
| -274.64756 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | -20.96732 | 138.30182 | -63.43817 | -1.97333 | 97.60944 |
| -57.19812 | .01204 | .02424 | .10692 | .08645 | .10692 |
| .06645 | | | | | |
| | | | | | |
| 2.62069 | 6.25950 | -304.54803 | 225.40639 | 662.21365 | 662.21692 |
| 6.13150 | 54.69632 | 54.58573 | -.00000 | 168.54622 | 168.54624 |
| 168.02017 | 419.22076 | 419.11573 | -.00000 | 89.75712 | 89.75728 |

REAL TIME CLOCK INTERROGATED AT 11:34:28

The equations developed in Section II are programmed in subroutines INIT, CONFIG, STAB, FORCE, and INTEQM. In order to further associate physical concepts with the program listing, the following cross reference between the listing and Section II is presented. Since there is not a one-for-one correspondence between Section II equations and statements in the listing, and since not every statement in the listing is numbered, then the statement numbers tabulated below should be regarded as approximate locations for the FORTRAN version of the Section II equations.

<u>INIT</u>

| Statement No. from Listing | Section II Equation No. |
|---|---|
| 110 | 1 |
| 114 | 14, 13, 5, 7, 6 |
| 150 | 3 |
| 160 | 15 |

<u>CONFIG</u>

| Statement No. from Listing | Section II Equation No. |
|---|---|
| 110 | 1 |
| 120 | 16 |
| 124 | 17 |
| 160 | 7, 11, 12 |
| 1604<br>162<br>170 | 8 |
| 172 | 9, 10 |
| 180 | 25, 26, 3 |
| 201 | 14 |
| 202 | 15 |
| 202<br>3005 | 27 |
| 2045<br>2049 | 28 |

| Statement No. from Listing | Section II Equation No. |
|---|---|
| 220<br>221 | 18 |
| 221 | 19 |
| 224<br>229 | 21, 22, 24 |
| 230 | 2 |
| 232 | 70 |
| 2499<br>2410 | 81, 82 |
| 2415 | 83 |
| 2430 | 84 |
| 2433 | 86 |
| 3458 | 84 |
| 2462 | 87 |
| 2463 | 88 |
| 262 | 71 |

**STAB**

| Statement No. from Listing | Section II Equation No. |
|---|---|
| 62 | 72 |
| 250<br>255 | 76 |
| 256 | 77, 78 |
| 257 | 79 |

**FORCE**

| Statement No. from Listing | Section II Equation No. | Statement No. from Listing | Section II Equation No. |
|---|---|---|---|
| 75 | 33 | 1899 | 46 |
| 110 | 29 | 1903 | 47 |
| 120 | 30 | 191<br>192 | 48 |
| 150 | 31, 32 | 204 | 49 |
| 156<br>160 | 32 | 222 | 52 |
| 180 | 40 | 301<br>303<br>302 | 38, 55 |
| 1805 | 42 | 311<br>314 | 39 |
| 1866 | 43 | | |
| 1896 | 45 | | |
| 1897 | 44 | | |

**INTEQM**

| Statement No. from Listing | Section II Equation No. |
|---|---|
| 31<br>35 | 60 |
| 40 | 55 |
| 42 | 68, 69 |
| 60 | 67 |
| 70<br>80 | 65, 66 |

## SECTION IV

## COMPUTER STUDIES

### 4.1 SPECIAL STUDIES

In the course of developing the computer program, investigations were planned to answer such questions as:

1. Could the computer simulation of the Surveyor landing be made more efficient?
2. Were there any integration methods immediately available which would surpass variable interval Runge-Kutta in overall performance?
3. Could the strategy of interval adjustment be made more efficient?
4. Was it feasible to relax the error specifications in order to improve running time?
5. What would be the effect of modifications to the representation of the sliding friction coefficient?

Based on the results of studies pertaining to these questions, the parametric studies were begun with confidence that the efficiency was optimum. A discussion of these studies appears in the following paragraphs.

#### 4.1.1 Simulation Efficiency

It was apparent from examination of the Phase I test cases (see Reference (b)) that program running time could be improved substantially by "smoothing" some calculations and by eliminating others when they were unnecessary to the determination of spacecraft motion. The "smoothing" refers to the elimination of unrealistic abrupt changes in force levels, and to the removal of numerical "noise" which can occur when a balance between two opposing forces is sought. These phenomena can have a ruinous effect on the error checking integration method without contributing to the vehicle dynamics.

This smoothing process was employed to advantage in easing the transition from the shock absorber stroking to the restroking condition. As this change takes place, the shock absorber damping and friction forces change sign. In order to prevent step changes in these forces which are neither physically realistic nor conducive to rapid computation, a function was introduced in the simulation which allows these forces to pass through zero with no discontinuity. This higher order smoothing function has the characteristics of zeroing the dissipative forces at zero stroking velocity, but smoothly blending them into their normal computation mode when the stroke velocity has achieved a significant value.

During the restroking with the footpads off the surface, an approximate equilibrium between the shock absorber spring and damping forces exists. Due to the difficulty of accurately determining the difference between two substantial numbers, it is virtually impossible to get an accurate determination of the shock absorber force during this situation. As a result, the motion of the lower leg is erratic and small integration intervals result. Furthermore, these motions are inconsequential to the spacecraft dynamics since the footpad in question is not then in contact with the surface.

Because of the ineffectiveness of the footpad at this time, it should be satisfactory to employ an approximate technique to describe restroking with the footpad off the surface. If the inertia force is regarded as a small constant, the following differential equation is approximately true for this condition:

$$K + c\dot{\gamma}^2 + k\gamma = 0$$

where $\gamma$ is the angular depression of the lower leg. This equation implies the assumption that $\gamma$ changes approximately in proportion with the shock absorber stroke. Rearranging,

$$\dot{\gamma} = \sqrt{a\gamma + b}$$

which expresses $\dot{\gamma}$ as a function of $\gamma$ and two constants which were empirically determined from calculations performed using the more rigorous expression of the differential equation. Figure 4-1 shows the adequate representation of the $\gamma$ and $\dot{\gamma}$ time histories which is provided by the approximation. Integration of the semi-empirical $\dot{\gamma}$ was used to return the leg to the stop if all the following conditions were met:

1. Footpads off the surface.

2. Shock absorber restroking velocity from standard calculation equal or in excess of the programmed approximate equation.

3. Lower leg not on extension stop.

The above change improved the running time somewhat and in general, had a smoothing effect on the calculations during off-the-ground restroking. This approximation was used to advantage throughout the parametric study, but had to be removed from the final version of the program. Since the approximation is applicable only to a specific set of shock absorber parameters, its presence in the program would unnecessarily decrease the generality of the program.

Originally, the lower leg extension stop was represented by a very stiff spring. After the leg encountered the stop, an oscillation of the lower leg resulted. The damping force was quite low, and oscillations persisted for some time. A different stop was programmed which set the lower leg rates to zero when the stop was encountered. This froze the motion of the lower leg at the stop and was effective in eliminating the oscillations.

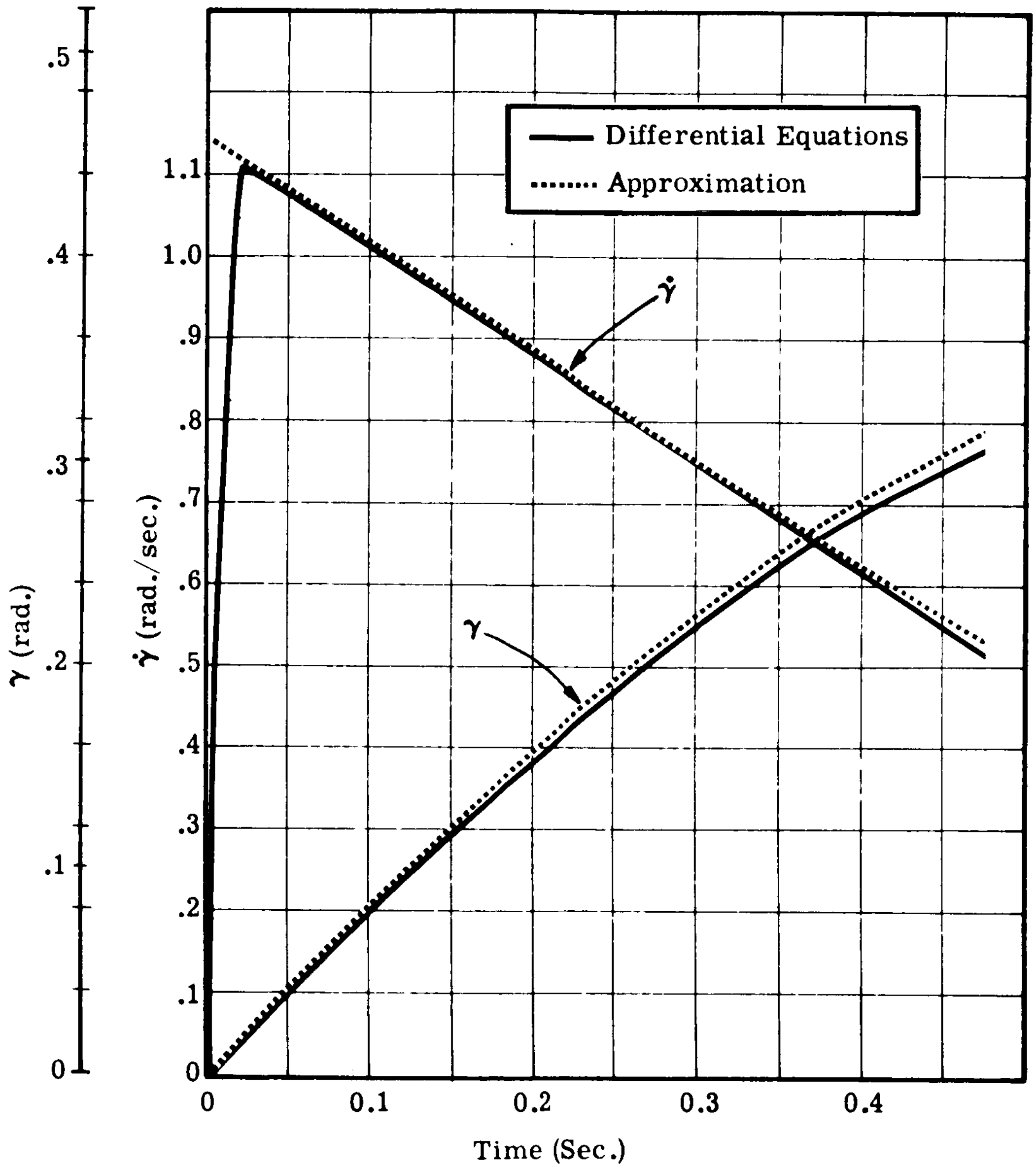



Figure 4-1. $\gamma$ and $\dot{\gamma}$ versus Time for Free Shock Absorber Return

Another aspect of the mechanical stop problem concerned the lower leg angular rate ($\dot{\gamma}$) that results from integration of the lower leg angular acceleration ($\ddot{\gamma}$). Under normal situations, $\dot{\gamma}$ is used to calculate the lower leg angular position ($\gamma$). However, when the shock absorber is on the extension stop, this is no longer true. In this instance, the forces are arbitrarily set to zero resulting in a $\ddot{\gamma}$ of zero and a $\gamma$ equal to the last value calculated. No problem exists until a force tending to stroke the leg appears. At this time, $\dot{\gamma}$ has the wrong value, and some time must pass before $\dot{\gamma}$ is integrated to the correct number so that the leg moves off the stop. Even though this time is small, the vehicle acts as if it has stiff legs for this time period. To alleviate this condition, a fictitious value of $\ddot{\gamma}$ is generated to run $\dot{\gamma}$ to a small value. This procedure is only used when the lower leg is on the stop and $\dot{\gamma}$ has a value greater than the above small value. Since $\dot{\gamma}$ is sensed to determine whether the shock absorber is stroking or restroking, $\dot{\gamma}$ must be held to a small positive value to keep from unnecessarily and incorrectly leaving the stop.

In the original version of the program, the calculation of the crushable block forces was suspended if the blocks were not in contact with the surface. Since one or more footpads are often off the surface and generating no forces, a procedure similar to that for the blocks was instituted for the footpads in the event they were not in contact with the surface. This increased the program efficiency by eliminating some unnecessary calculations.

These program modifications were all incorporated for the purpose of streamlining the simulation without affecting accuracy. For the original version of the program, the ratio of computer time to problem time was 215:1; this figure was reduced to 175:1 by means of the modifications.

Another method of reducing running time is simply an earlier termination of the run. This is feasible in some cases, since a stable landing can often be identified before the vehicle motions subside. This identification makes use of the results of the stability angle calculations described in Section II. When the spacecraft rocks up on one or two feet but does not topple, the calculated stability angle attains a minimum value at the end of the rocking motion and then experiences a substantial jump which is due to changing the reference used to calculate the stability angle at that instant. The program was modified to utilize this jump with the result that running time was saved in a large number of runs performed during the parametric studies.

### 4.1.2 Comparison of Integration Methods

In order to evaluate our choice of integration method, a limited study of various integration methods was conducted. The methods investigated were:

1. Variable interval, error checking Runge-Kutta.

2. Variable interval, error checking Milne with Runge-Kutta start and restart.

3. Constant interval (.002 second) Runge-Kutta.

4. Trapezoidal (.002 and .0005 second).

Method 1 was used as a basis for comparison purposes. Methods 3 and 4 are inherently simpler, but suffer from the disadvantage that no error control or estimation is possible.

The first comparison of integration methods was performed for a problem which was simple compared to the Surveyor landing dynamics problem. The problem was to determine the motion-time history of a bouncing ball which was idealized as an undamped, single degree of freedom system. The motion was predictable in closed form, and since the system was undamped, the motion was periodic. Further, the motion was similar to Surveyor in that abrupt contacts with the surface were made from time to time. Both error checking integration methods kept the problem under control. Milne integration (2) was slower to expand the integration interval and failed to expand the interval to as large a value as method 1. The lack of speed was due to a requirement for several past values to be in existence prior to interval expansion.

Reluctance of Method 2 to increase the interval was due to the method of interval expansion used in the available Milne integration routine. Interval expansion is restricted to interval doubling; that is, if the error is sufficiently small, the interval will be doubled. For a case in which a given interval results in small errors but twice that interval produces excessive errors, interval expansion will not occur. Therefore, this method will in general be running at an integration interval which is less than optimum and consequently smaller than the interval selected by method 1.

Method 4 with an integration interval of .0005 second had accuracy problems when the ball was in contact with the surface and was "loafing" when the ball was off the surface. The method seemed to have an "energy generator"; that is, the ball bounced to higher positions with each cycle. Method 3 performed with satisfactory accuracy, but like the other methods, was unable to improve upon the running time for Method 1.

Comparisons using Case 2 from Phase I (see Reference (b)) also support the selection of the variable interval Runge-Kutta integration. Method 2 failed to recover from the initial impact as the integration interval was reduced to a minimum and remained there. No meaningful data could be obtained. Method 3 resulted in satisfactory results, but no time improvement was noted. Method 4 with an interval of .002 second experienced permature termination of the run due to the absurd result that shock absorber bottoming occurred. Reducing the interval to .0005 second resulted in satisfactory answers but lost any running time advantage over either Methods 1 or 3. In general, no usable running time advantage was found in any of Methods 2, 3 or 4 over Method 1. Further, accuracy was as good or better in Method 1 than in the other methods.

### 4.1.3 Interval Adjustment Strategy

The strategy used in adjusting the integration interval size was examined in a further effort to improve program efficiency. It is tempting to reason that since there will be physically meaningful discontinuities in the applied forces (such as at recontact after free flight), and since it is a virtual certainty that the time interval will have to be closed down to a very small value in order to get through this discontinuity, then it would seem that computation could be saved when the discontinuity is encountered by immediately reducing the time interval to a small value. The time-consuming process of error checking, halving the interval and again error checking, etc., might be eliminated under these circumstances.

This is specious reasoning, however, as it fails to recognize that during the free flight, intervals as large as .3 seconds have been observed. When the discontinuity is encountered with this interval, there is a probability of .5 that by cutting the interval in half, the error check will be passed because the vehicle will not yet have contacted the surface.

Carrying this one step further, the probability that the error check will be passed after n cuts is:

$$P = \sum_{j=1}^{n} (1/2)^j$$

This means, for example, that if the time interval were .3 second when the recontact discontinuity was encountered, then the probability would be .875 that the error check would be passed with a time interval of .0375 or greater. This is clearly superior to arbitrarily dropping the time interval to the minimum allowable (say .00001) as soon as the discontinuity occurs.

It is possible that further gains in speed could have been achieved by increasing the time interval more quickly when a smooth portion of the landing occurs, such as the free flight phase. Although time interval selection might not have been quite optimum, the strategy used to select intervals appeared to be rather good. This contention was supported by a special calculation which was incorporated in the computer program. Printed out were the number of integration intervals in a run, the average size, the distribution of sizes, and a distribution of the number of times that successive cuts in interval size had to be made to pass an error check. This information showed that in all runs made using the existing strategy, no interval cutting was required in over half of the integration steps. This is indicative of a rather efficient process.

One attempt was made to use a slightly modified interval adjusting technique. This method was more conservative for errors close to the maximum and more liberal for errors close to zero error than the present method. The modified technique was found to be significantly less efficient than the original method.

### 4.1.4 Error Specification

Having obtained confidence that the program is functioning efficiently, the possibility of relaxing the truncation error specification was investigated. Since these specifications are used in the error check, they control the interval size and therefore the problem running time. Additional special printouts were instituted as an aid in assessing the effect of changing the error specifications. Two of these printouts provided summations of truncation errors actually incurred in the integration of the problem variables. Summations are performed both on an absolute and an algebraic basis. A third special printout provided a distribution of the critical variables in the form of a tabulation of how many times each of the dependent variables restricted the interval size.

The allowable error specifications on a per second basis used in the Phase I test cases (see Reference (b)) are tabulated on the next page.

6 degrees for footpad angles

2 degrees/second for lower strut angular rates - $E(\dot{\gamma})$

4 degrees for lower strut angles

1/6 foot/second for CG velocities

1/6 foot for CG position

1/6 degree/second for rotational rates about the CG - $E(\omega)$

2/3 degree for rotation about the CG

With these specifications the special printout affirmed that $\dot{\gamma}$ (lower leg angular rate) was the most critical variable. The body angular rates ($\omega$) provided a significant amount of limiting, but much less so than $\dot{\gamma}$.

Several combinations of specified $\dot{\gamma}$ and $\omega$ errors were applied to Case 2, and the effects on running are tabulated below:

| $E(\dot{\gamma})$ Deg/Sec/Sec | $E(\omega)$ Deg/Sec/Sec | $\frac{\text{Computer Running Time}}{\text{Problem Time}}$ |
|---|---|---|
| 2 | 1/6 | 175 |
| 4 | 1/6 | 155 |
| 8 | 1/6 | 120 |
| 16 | 1/6 | 120 |
| 8 | 1/2 | 130 |
| 16 | 1/2 | 120 |
| 8 | 1/4 | 125 |

It is evident from this tabulation that relaxation of the error criteria does decrease running time. Note, however, that running time does not decrease uniformly as the allowable errors are increased. Evidently $E(\dot{\gamma})$ and $E(\omega)$ are both restricting the interval size in the latter 5 cases, so that when one set of errors is relaxed, the other set provides additional limiting.

Figure 4-2 illustrates the effect of increased error criteria on the pitch angle versus time plot. This time history, chosen because of the importance of pitch angle in stability determination, shows that two of the specified error criteria result in departures of 6° and 8° from the original choice of $E(\dot{\gamma}) = 2$ and $E(\omega) = 1/6$. The remaining 4 combinations were not distinguishable from the original choice.

The choice of $E(\dot{\gamma}) = 8$ and $E(\omega) = 1/6$ is attractive because of its relatively low running time ratio (120:1) as well as its good agreement with respect to the pitch angle time history. The variation in the pitch angle time history with specified error unfortunately

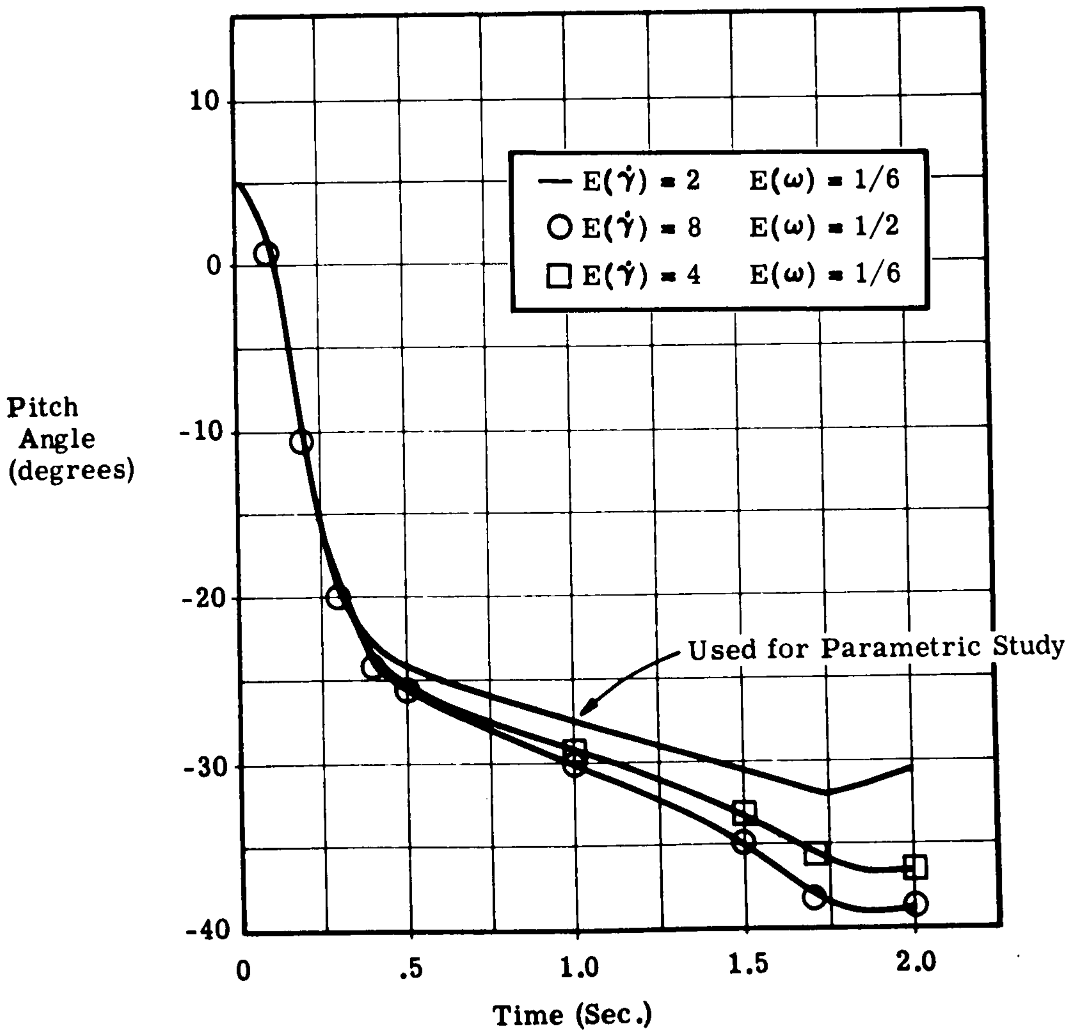



Figure 4-2. Pitch Angle versus Time for Various Error Criteria

is not clearly understood, so that a conservative choice of criteria was selected for the parametric studies. The values selected were the original values which included $E(\dot{\gamma})$ = 2 and $E(\omega)$ = 1/6.

### 4.1.5 Sliding Friction

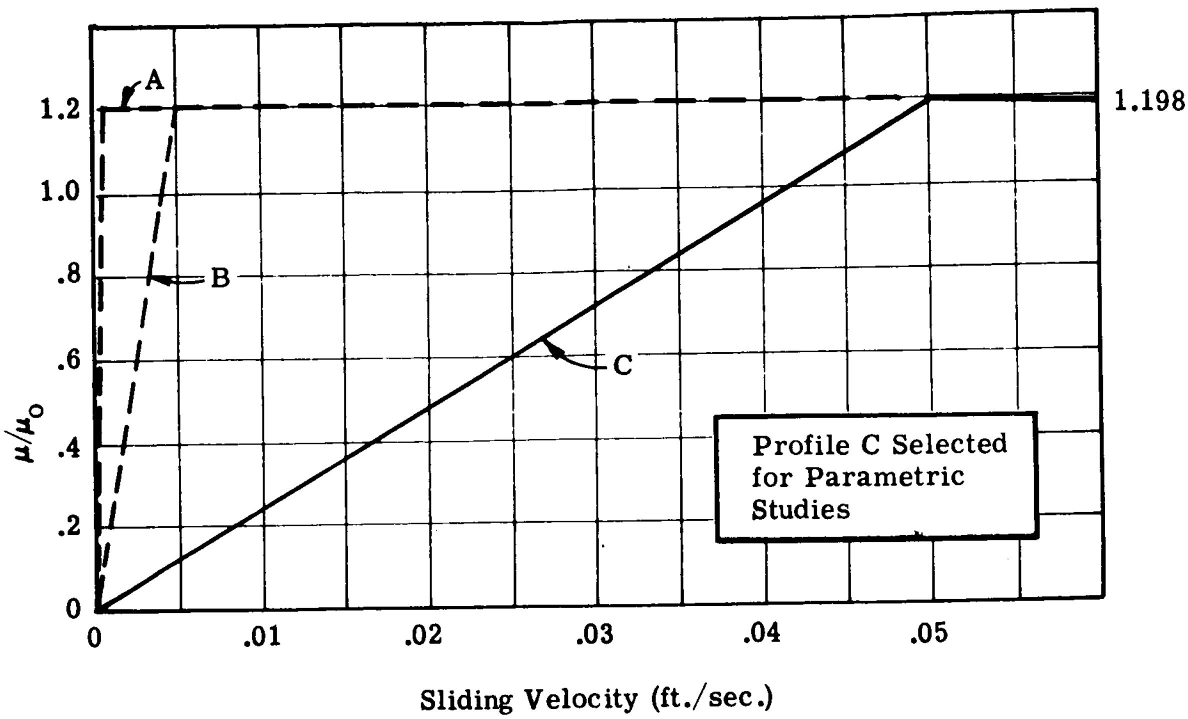
A study was undertaken to establish whether or not the phenomenon of footpad chatter would be predicted by the computer program. Accordingly, the nominal coefficient of friction was raisedfrom .8 to 1.4, and the variation of sliding friction coefficient as a function of velocity was changed. Figure 4-3 shows the various forms of the profile which were considered. Profile C is shown in two different scales; all profiles coincide for sliding velocities of .05 foot/second and above.

Computer runs which were conducted using $\mu$ = 1.4 and the various friction coefficient profiles revealed two distinct difficulties, both of which had a disastrous effect on running time. First, the expected difficulty associated with a sharp change in sliding friction force at low velocities was experienced. This problem was most severe for the "step" representation of the friction profile (profile A), while profile C was sufficiently smooth that little difficulty was encountered. For profile A, and to a lesser extent profile B, the abrupt change in friction coefficient at low velocities did not permit the error checks to be passed, so that the integration interval was reduced to the minimum value.

This problem was accentuated by a second difficulty, which was an accuracy problem. It appears that the use of a minimum time interval of .000001 second can result in error checking meaningless numbers due to the number of significant figures carried in the single precision mode on the UNIVAC 1107. In effect, round-off errors generated in the computations were being checked instead of truncation errors. This problem develops when the integration interval approaches the minimum allowable time interval.

Once the time interval has been reduced to a minimum, it becomes nearly impossible to pass an error check, open up the time interval and thereby return the calculated values to the realm of significance. This difficulty is not restricted to the occurrence of rapidly changing friction forces, however. In fact it was first identified in a situation where the sliding velocity was in excess of 1 foot/second, but the torques tending to rotate the spacecraft were approximately in balance. As was previously noted, equilibrium conditions of this type are difficult to describe accurately. Therefore, the time interval closed down and having done so, could not recover. It has been found that increasing the minimum time interval to .00001 seconds has alleviated the problem of accuracy, but the problem of sharp changes in friction force remain with profiles A and B.

Figure 4-4 compares the pitch angle time histories throughout the first 1/2 second of the landing for the cases in which $\mu$ = 1.4 (profiles A, B and C) with the time history for $\mu$ = .8 (profile C). First, note that there is very little difference among the three plots for $\mu$ = 1.4 data. Although the amount of data is admittedly sparse, there is no indication that profile C is inadequate. Since this profile results in smoother calculations at no loss in accuracy, it was used for the parametric studies.



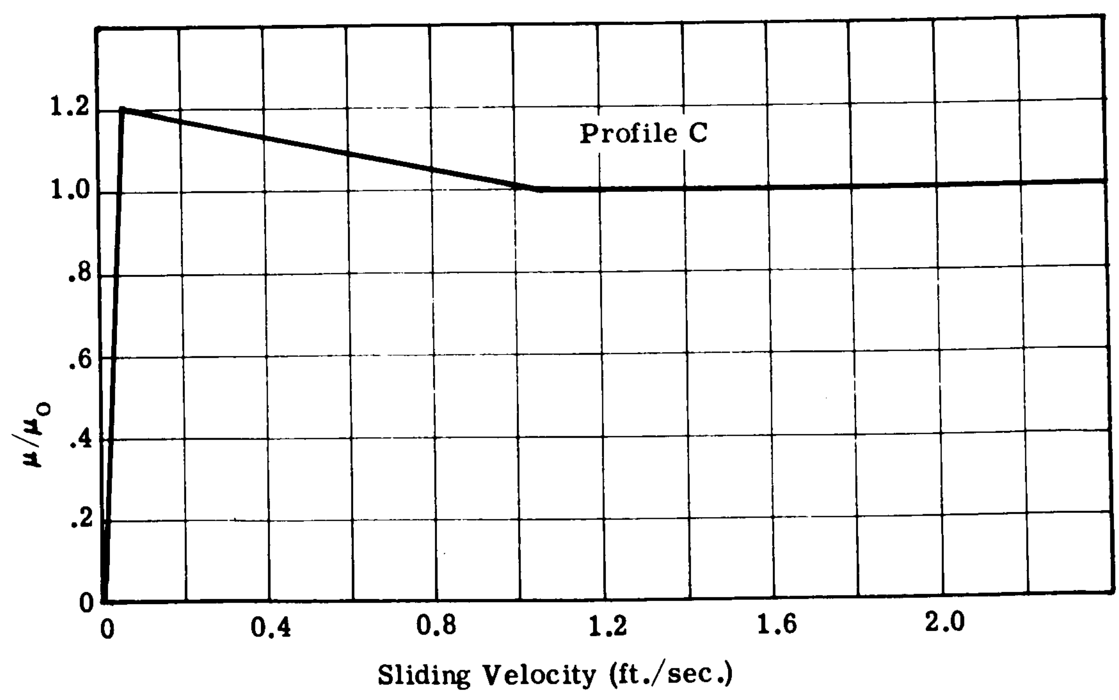




Figure 4-3. Sliding Friction Coefficient versus Sliding Velocity

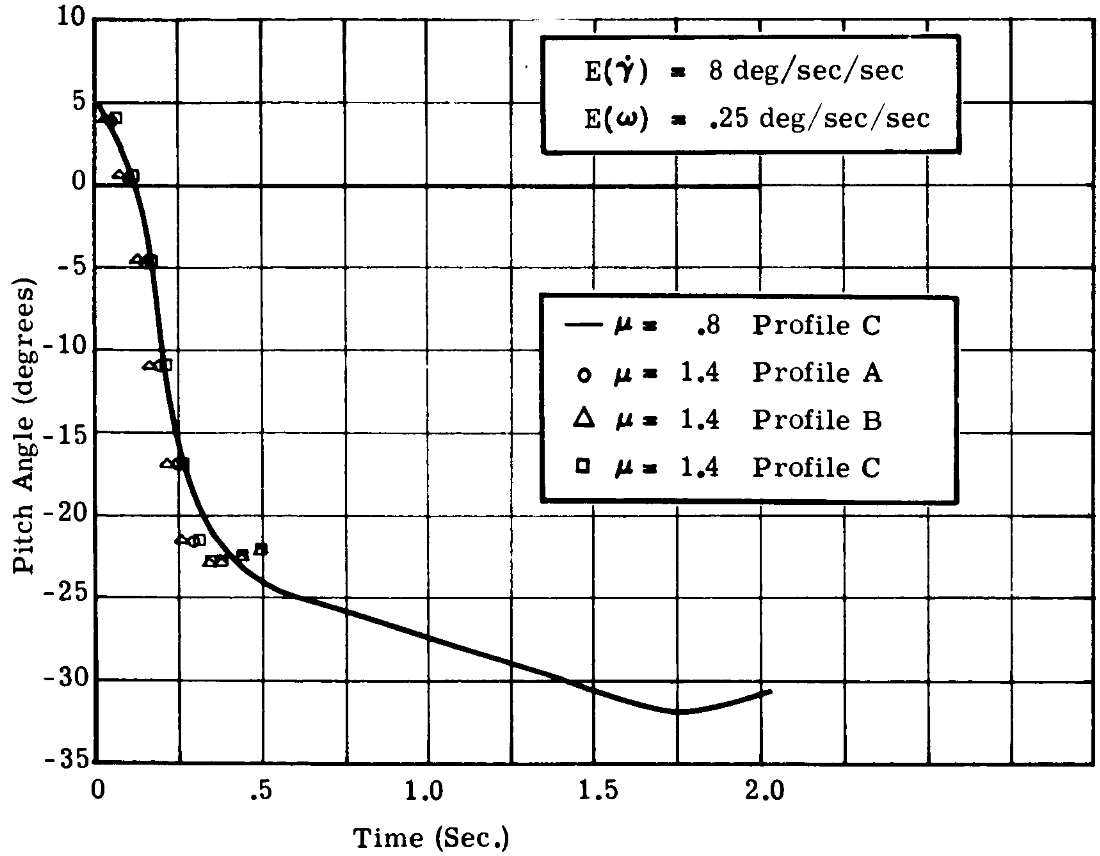



Figure 4-4. Pitch Angle versus Time for Various Friction Representations

## 4.2 PARAMETRIC STUDIES

The program development effort which has been discussed throughout this report was motivated primarily by the requirement to perform extensive parametric studies. These studies, which constituted the majority of the effort expended in Phase II, were formulated to establish Surveyor landing stability boundaries for a wide range of touchdown conditions.

The dynamics of the spacecraft landing are functions of two major types of variables: those which describe the vehicle configuration (geometry, shock absorber characteristics, etc.) and those which specify the touchdown conditions (CG velocities, angular rates, angular orientation, nominal sliding friction coefficient, etc.). The vehicle configuration was held constant throughout the parametric studies although provision has been included in the simulation to facilitate changes in these parameters.

The spacecraft configuration is described by the parameters in Table 4-1. Table 4-2 lists the variables describing the touchdown conditions. The variables which were varied in the parametric study were chosen from Table 4-2.

In order to keep the extent of the parametric study within reasonable bounds, the range of touchdown conditions to be considered was necessarily subject to some limitations. As indicated in Table 4-2, the ground slope and friction coefficient were assigned constant values, while no angular rates were considered during the parametric studies. The approach followed in varying the angular orientation and initial CG velocity vector was first to establish a baseline configuration and then to study the effect of variations about this set of parameters.

### 4.2.1 Baseline Configuration

In selecting a set of baseline conditions, it is logical to require that the configuration will result in a "planar" landing. That is, the landing can be thought of as a 2 dimensional problem in which there is no motion of the CG across the slope, nor is there any spacecraft rotation except pitch. This choice appeals to the intuition since it would seem reasonable to expect that the most critical touchdown conditions from the standpoint of overturning would involve the CG velocity in the direction of maximum slope, tending to topple the vehicle over two downhill feet. It turns out that intuition is a reliable, but not infalliable, guide; only a few conditions have been found which are more critical than the planar cases.

A second baseline condition can be arbitrarily assigned. This is the condition that the spacecraft is level at touchdown (vertical centerline). This choice has only simplicity to recommend it; the pitch and yaw angles will be zero.

It was planned that the stability profiles would be generated by varying horizontal velocity until the transition value was established. Vertical velocity was to be held at a fixed value for various angular orientations and cross-slope angles. A vertical velocity of 20 feet/second was chosen with the thought that variations in stability boundaries would be accentuated by the high vertical velocity. Figures 4-5 and 4-6 indicate that this was not the case. Rather, another phenomenon was being introduced which rendered 20 feet/second vertical velocity useless for stability considerations.

TABLE 4-1. NUMERICAL VALUES OF PARAMETERS DESCRIBING SPACECRAFT CONFIGURATION

| Parameter | Description | Value |
|---|---|---|
| $M$ | mass | 20 slugs |
| $I_{xx}$ | moment of inertia relative to x-axis | 138.12 slug-ft.$^2$ |
| $I_{yy}$ | moment of inertia relative to y-axis | 134.29 slug-ft.$^2$ |
| $I_{zz}$ | moment of inertia relative to z-axis | 145.78 slug-ft.$^2$ |
| $I_{xy}$ | moment of inertia relative to x and y-axes | 6.81 slug-ft.$^2$ |
| $I_{xz}$ | moment of inertia relative to x and z-axes | -5.87 slug-ft.$^2$ |
| $I_{yz}$ | moment of inertia relative to y and z-axes | 2.57 slug-ft.$^2$ |
| $\ell_{DO}$ | extended length of shock absorber link | 3.1275 ft. |
| $\ell_{D\ min}$ | minimum length of shock absorber link | 2.7525 ft. |
| $\Delta CG$ | height of CG above base plane | 1.445 ft. |
| $\ell_p$ | distance from vehicle centerline to lower hard-point | 3.1875 ft. |
| $\ell_L$ | length of lower link | 3.375 ft. |
| $\ell_A$ | length of A-frame link | .95833 ft. |
| $g$ | Gravitational acceleration | 5.31 ft./sec.$^2$ |
| $\alpha_1$ | footpad rotational stop (lower) | .75 deg. |
| $\alpha_2$ | footpad rotational stop (upper) | 45 deg. |
| $\beta$ | Inclination of A-frame link | 41 deg. |
| $\ell_c$ | distance from vehicle centerline to block center-line | 2.4167 ft. |
| $h_{BO}$ | uncrushed block thickness | .66667 ft. |
| $I_\alpha$ | moment of inertia of footpad about pivot | .00373 slug-ft.$^2$ |
| $I_\gamma$ | moment of inertia of lower link about hinge | 1.347 slug-ft.$^2$ |
| $h_{fO}$ | uncrushed footpad thickness | .54167 ft. |
| $d_B$ | block diameter | .58333 ft. |
| $d_f$ | footpad diameter | 1 ft. |
| $R_c$ | nominal shock absorber compressive damping coefficient | 54 lbs.-sec.$^2$/ft.$^2$ |

TABLE 4-1. NUMERICAL VALUES OF PARAMETERS DESCRIBING SPACECRAFT CONFIGURATION (CONC.)

| Parameter | Description | Value |
|---|---|---|
| $R_R$ | nominal shock absorber rebound damping coefficient | 864 lbs.-sec.$^2$/ft$^2$ |
| $K_D$ | shock absorber hydraulic spring rate | 3600 lb/ft. |
| $K_S$ | shock absorber extension stop spring rate | 100,000 lb/ft. |
| $F_P$ | shock absorber preload | 185 lb. |
| $\nu$ | shock absorber mechanical friction coefficient | .05 (dimensionless) |
| $C_B$ | nominal block crush pressure | 5425.6 lb/ft$^2$ |
| $C_f$ | nominal footpad crush pressure | 1578.8 lb/ft$^2$ |
| $\theta_1$ | angle between spacecraft y-axis and No. 1 leg set | 0 deg. |
| $\theta_2$ | angle between spacecraft y-axis and No. 2 leg set | 120 deg. |
| $\theta_3$ | angle between spacecraft y-axis and No. 3 leg set | 240 deg. |

TABLE 4-2. NUMERICAL VALUES OF PARAMETERS DESCRIBING TOUCHDOWN CONDITIONS

| Parameter | Description | Value |
|---|---|---|
| $\theta_s$ | ground slope | 15 deg. |
| $\mu$ | sliding friction coefficient | 1.0 (dimensionless) |
| $\omega_x$ | initial angular rate about vehicle x-axis | 0 deg/sec. |
| $\omega_y$ | initial angular rate about vehicle y-axis | 0 deg/sec. |
| $\omega_z$ | initial angular rate about vehicle z-axis | 0 deg/sec. |
| $\Psi_o$ | initial pitch angle | variable (deg.) |
| $\Phi_o$ | initial yaw angle | variable (deg.) |
| $\Xi_o$ | initial roll angle | variable (deg.) |
| $V_v$ | touchdown vertical velocity | variable (ft/sec.) |
| $V_H$ | touchdown horizontal velocity | variable (ft/sec.) |
| $\lambda$ | cross-slope angle | variable (deg.) |

Figure 4-5 presents the horizontal velocity versus roll angle stability profile for a level vehicle. The vertical velocity was 20 feet/second and the cross-slope angle was 0° (uphill, planar landing). The cross-slope angle is measured between the horizontal velocity vector and the Y-axis (ground coordinates). See Figure 4-22. The roll angle was varied between 0° (No. 1 foot uphill) and 60° (No. 3 foot downhill). No instances of toppling were observed, but the range of tolerable horizontal velocity was limited by shock absorber bottoming. The same result is shown in Figure 4-6 in which the horizontal velocity versus cross-slope angle stability profile was sought. The roll angle was 0°.

The data shown in these two figures strongly suggest that a systematic search for the baseline configuration be conducted. A cross-slope angle of 180° (downhill landing) was chosen, together with a roll angle of 0°, as these conditions would seem less likely to produce excessive stroking than the uphill landings. Figure 4-7 presents a vertical velocity versus horizontal velocity stability profile for these conditions. It was possible to define the stability boundary since toppling, not excessive stroke was the limiting phenomenon. Notice the lack of transition at low vertical velocities. Figure 4-8 presents data for a similar investigation, but with a 60° roll angle. Again no toppling was observed, and excessive stroke was the limiting factor at rather high velocities.

The data presented in Figures 4-5 through 4-8 are quite informative regarding the general stability characteristics of Surveyor. Figure 4-7 especially suggests that the spacecraft is an inherently stable configuration. Further, downhill landings seem to be much more critical than uphill landings, especially those downhill landings with roll angles around 0°. Figure 4-7 also suggests that a vertical velocity of 14 feet/second would be attractive for a baseline condition as it results in one of the lowest transition horizontal velocities, but it is high enough to avoid the area of high stability at vertical velocities of 10 feet/second and below. Then the baseline conditions can be summarized as follows: 14 feet/second vertical velocity, level vehicle and planar motion.

### 4.2.2 Parametric Variations for Level Vehicle

The parametric studies have made use of certain symmetries which exist in the spacecraft configuration. The three-legged configuration possesses a symmetry which repeats at roll angle intervals of 120°. The definition of cross-slope angle (see Figure 4-22) permits negative cross-slope angles to be neglected since there is no fundamental difference between positive and negative cross-slope angles. Strictly speaking, these symmetries do not exist in a dynamical sense, since the spacecraft products of inertia are non-zero. The products are inconsequential compared with the moments of inertia, however, so that the symmetries can be utilized.

The interaction of horizontal velocity, roll angle and cross-slope angle was investigated for a level vehicle with a 14 foot/second vertical velocity. Figure 4-9 is a horizontal velocity versus roll angle stability profile for 180° cross-slope angle (downhill landing). The transition which occurs at 0° roll angle (1 foot uphill) is considered to represent the baseline condition. The baseline transition horizontal velocity is between 11 and 11-1/2 feet/second. Figure 4-9 (180° cross-slope) shows that as the roll angle becomes larger, the transition horizontal velocity increases.

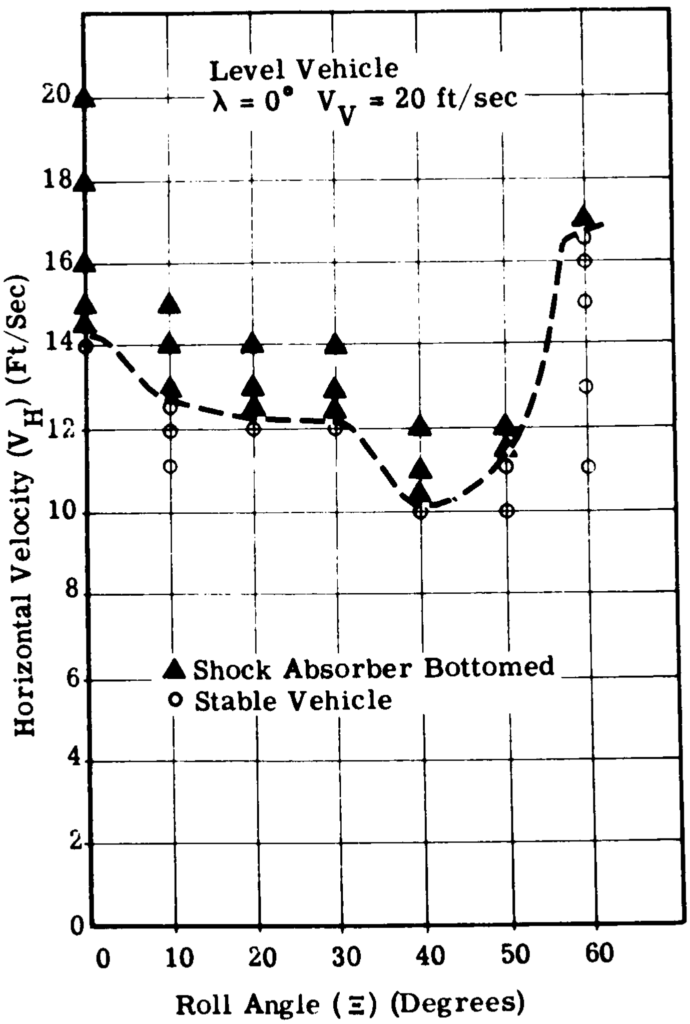



Figure 4-5. $V_H$ versus Ξ Stability Profile

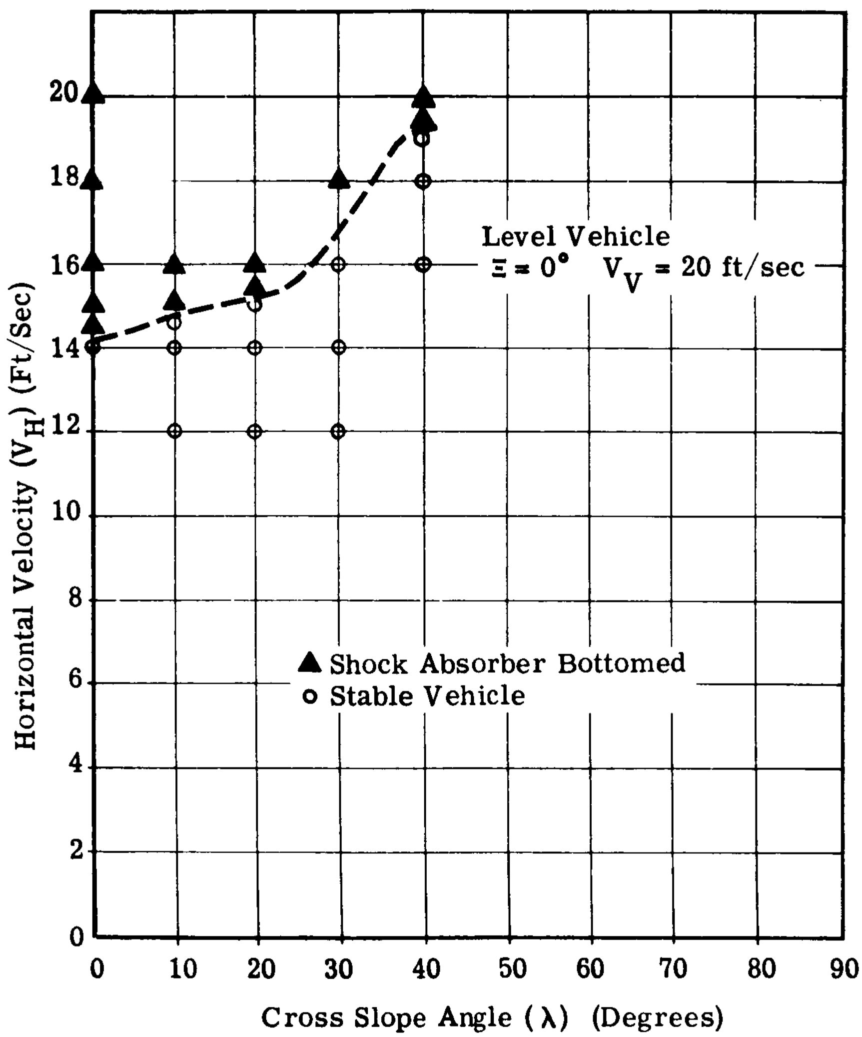



Figure 4-6. $V_H$ versus λ Stability Profile

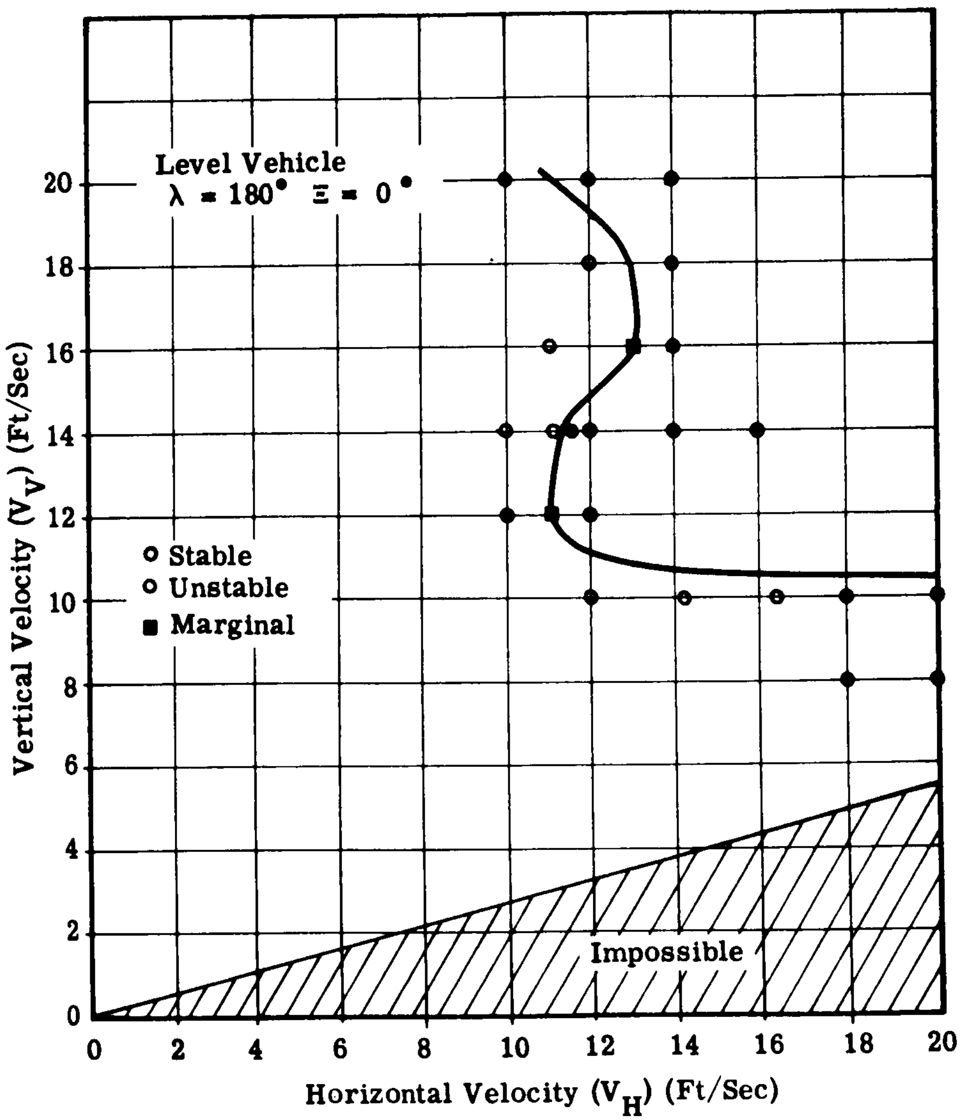



Figure 4-7. Vertical Velocity versus Horizontal Velocity

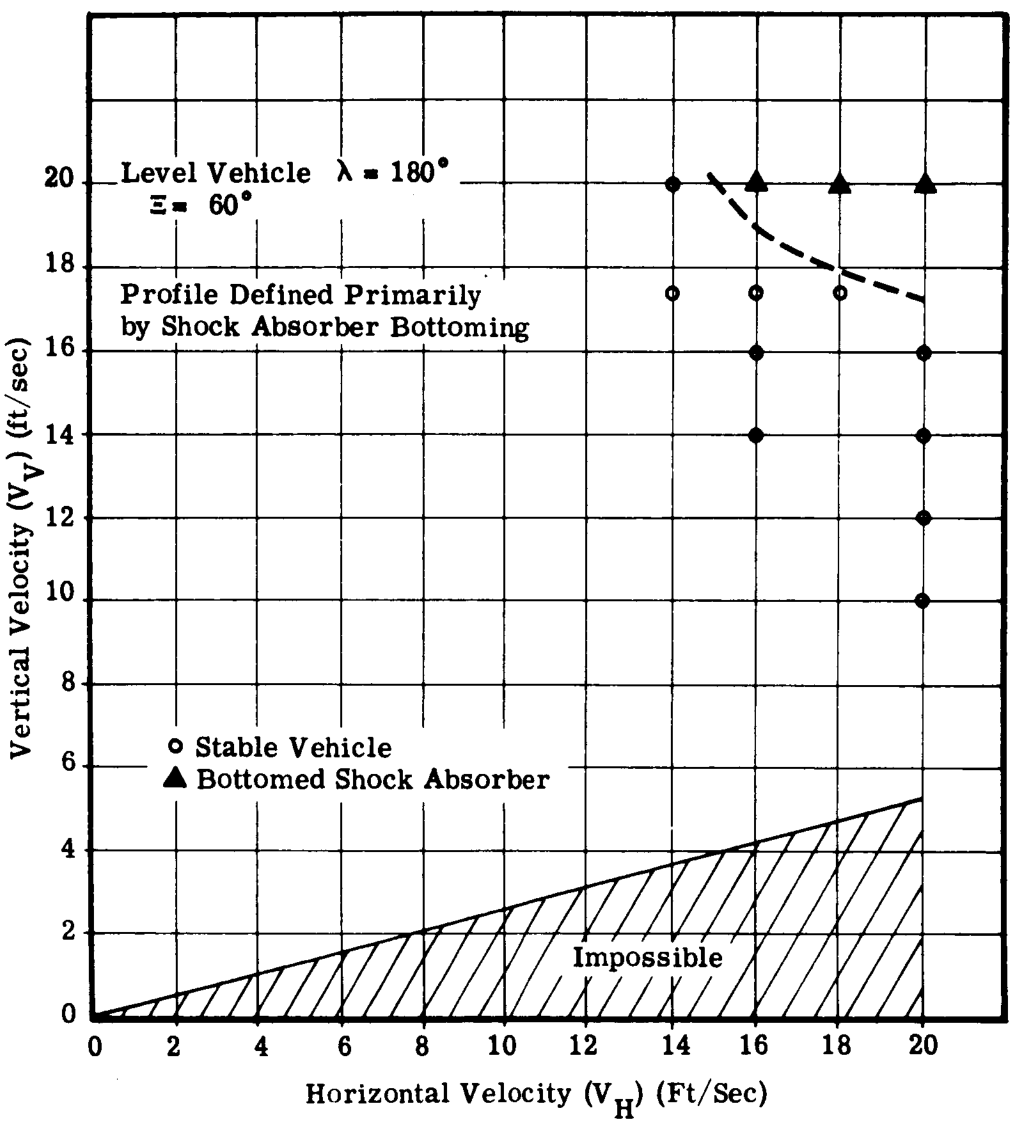



Figure 4-8. Vertical Velocity versus Horizontal Velocity

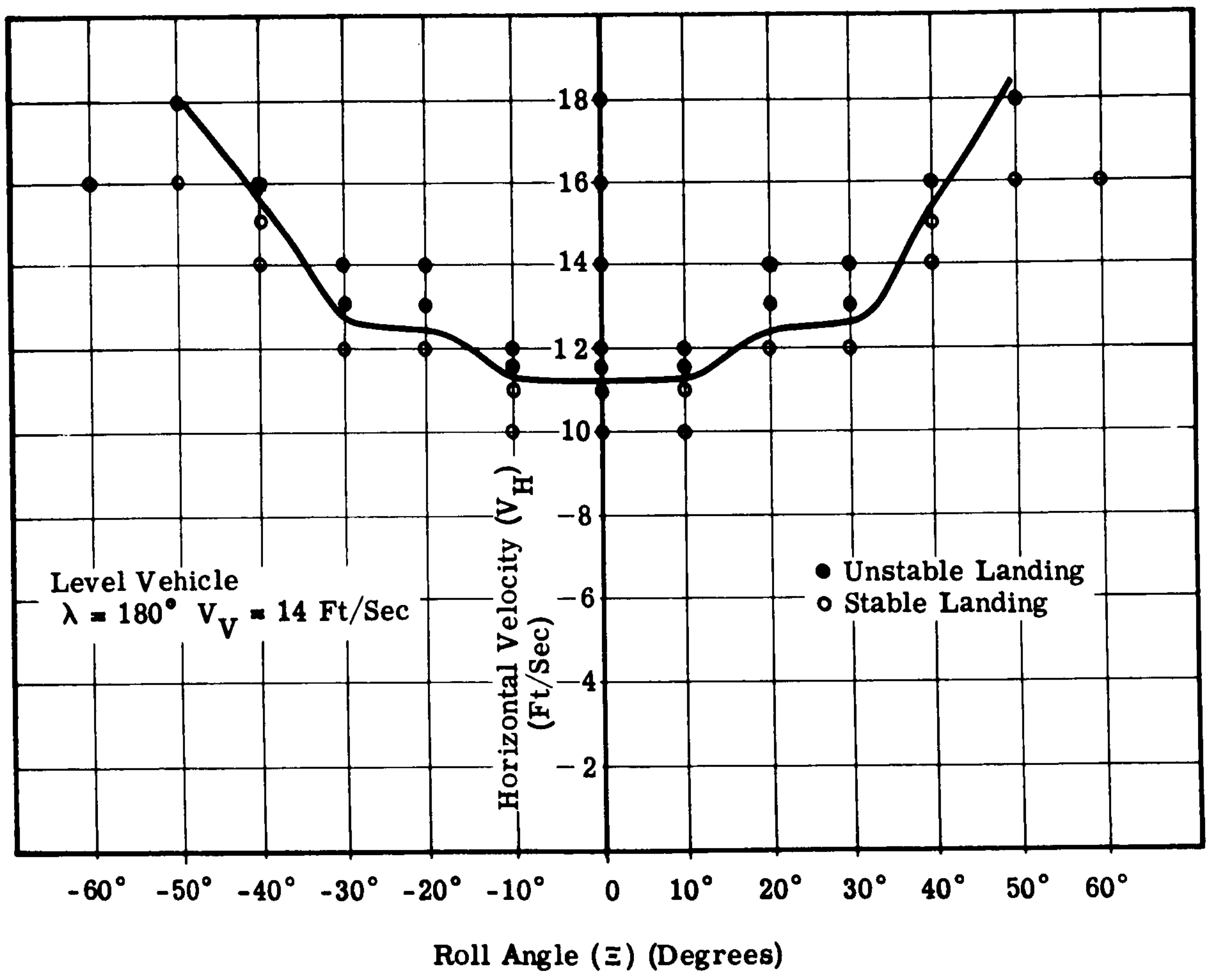



Figure 4-9. $V_H$ versus Ξ Stability Profile

Figures 4-10 through 4-19 present horizontal velocity versus roll angle stability profiles for cross-slope angles from 170° through 80° in increments of 10°. With the exception of Figure 4-19 (80° roll angle) where no transitions were found, the figures present stability profiles which can all be characterized as concave upward. That is, for each choice of cross-slope angle there is a range of roll angles for which the transition horizontal velocity is minimum. As cross-slope angle proceeds from 180° toward 80°, two trends can be observed: first, the minimum transition horizontal velocity increases; also, the range of roll angles which produce the minimum transition horizontal velocities shifts toward the negatively increasing roll angles.

These trends are perhaps more clearly illustrated by depicting the stability boundary as a surface rather than a family of curves. Figure 4-20 shows the stability surface in perspective. The surface may be interpreted as the transition horizontal velocity as a function of roll and cross-slope angles. Combinations of roll and cross-slope angles which lie above the surface will produce toppling. Figure 4-20 actually shows only half of the surface; the plot could be extended to show the symmetry about the vertical axis defined by a 0° roll angle and a 180° cross-slope angle. That is, if the plot were extended beyond the cross-slope angle of 180° (e.g.to cross-slope angles of -160°, -140°, etc.) then the additional half of the surface would look like Figure 4-20 with the signs of the scales reversed.

Figure 2-21 is the top view of the stability surface, and can be regarded as a contour map showing transition horizontal velocity as a function of cross-slope angle and roll angle. The tendency for transition horizontal velocities to increase as the horizontal velocity vector swings from the 180° cross-slope angle (downhill) around to 90° (across the slope) can be readily observed. The shift in critical transition horizontal velocities toward the negatively increasing roll angles can also be seen.

It is interesting to examine the reasons underlying these trends. The increase in stability which accompanies the departure of the horizontal velocity vector from the downhill direction is not surprising. It is to be expected that toppling is more likely to occur for a downhill landing than for a landing across the slope. The tendency for the "valley" in the contour map of Figure 4-21 to run up and to the right deserves further comment, however.

Superimposed on the contour map of Figure 4-21 is the straight line AB, which extends from the point representing a 0° roll angle and a 180° cross-slope angle (downhill base-line case) to the point representing a 30° roll angle and a 90° cross-slope angle. This line is the locus of roll angle cross-slope angle combinations for which the horizontal velocity vector at touchdown bisects a line joining the two leading feet. Intuition suggests that the critical combination of roll angle and cross-slope angle should lie on or near the line, since the vehicle touchdown configuration would seem to offer the minimum resistance to toppling under these circumstances. That is, for a roll angle of -30°, for example, a cross-slope angle of 150° would be expected to be critical. This rationale is probably adequate to explain the general character of the stability surface, but Figure 4-21 indicates the locus of critical horizontal velocities is actually rotated slightly counter-clockwise from AB.

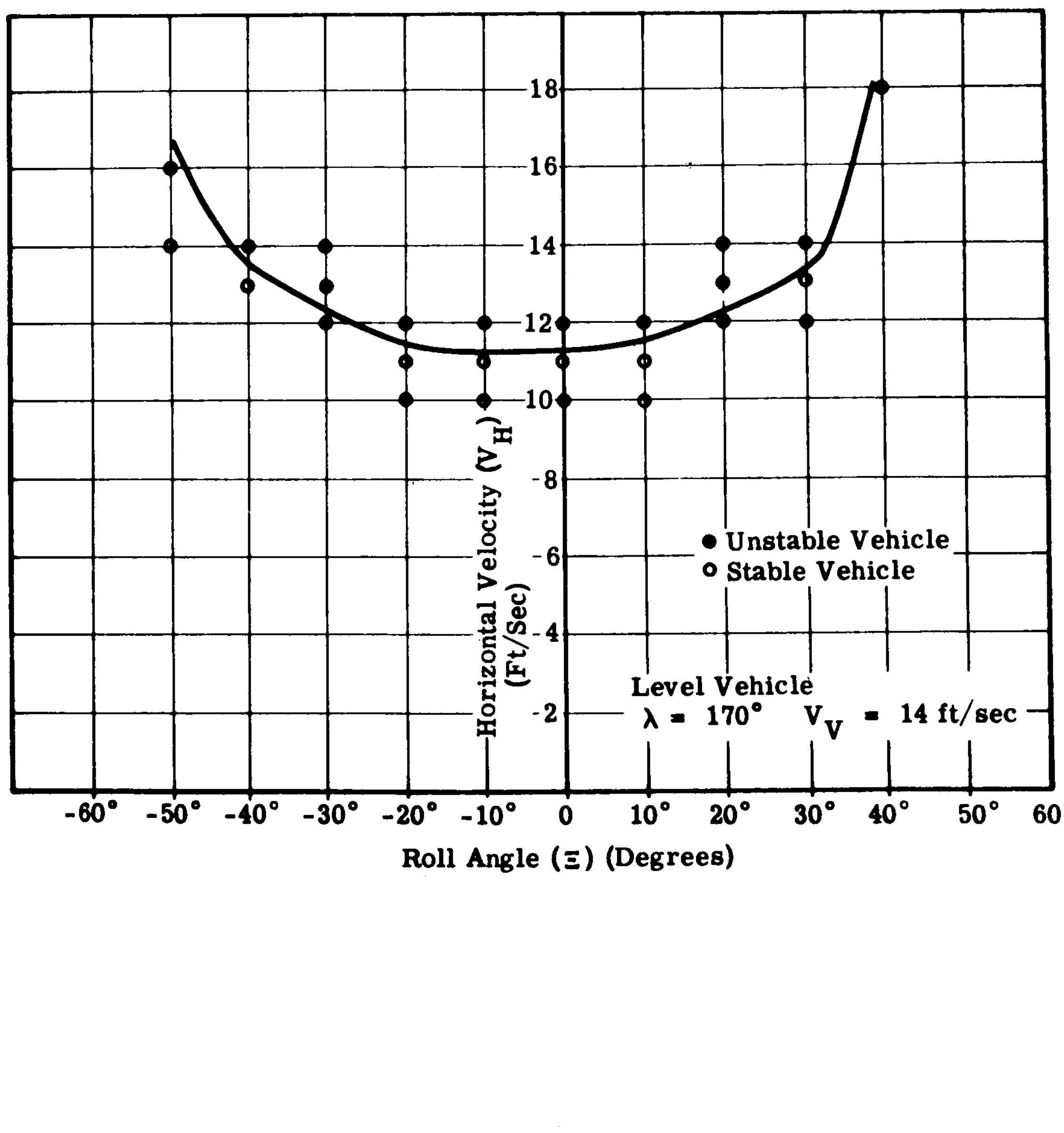



**Figure 4-10. $V_H$ versus Ξ Stability Profile**

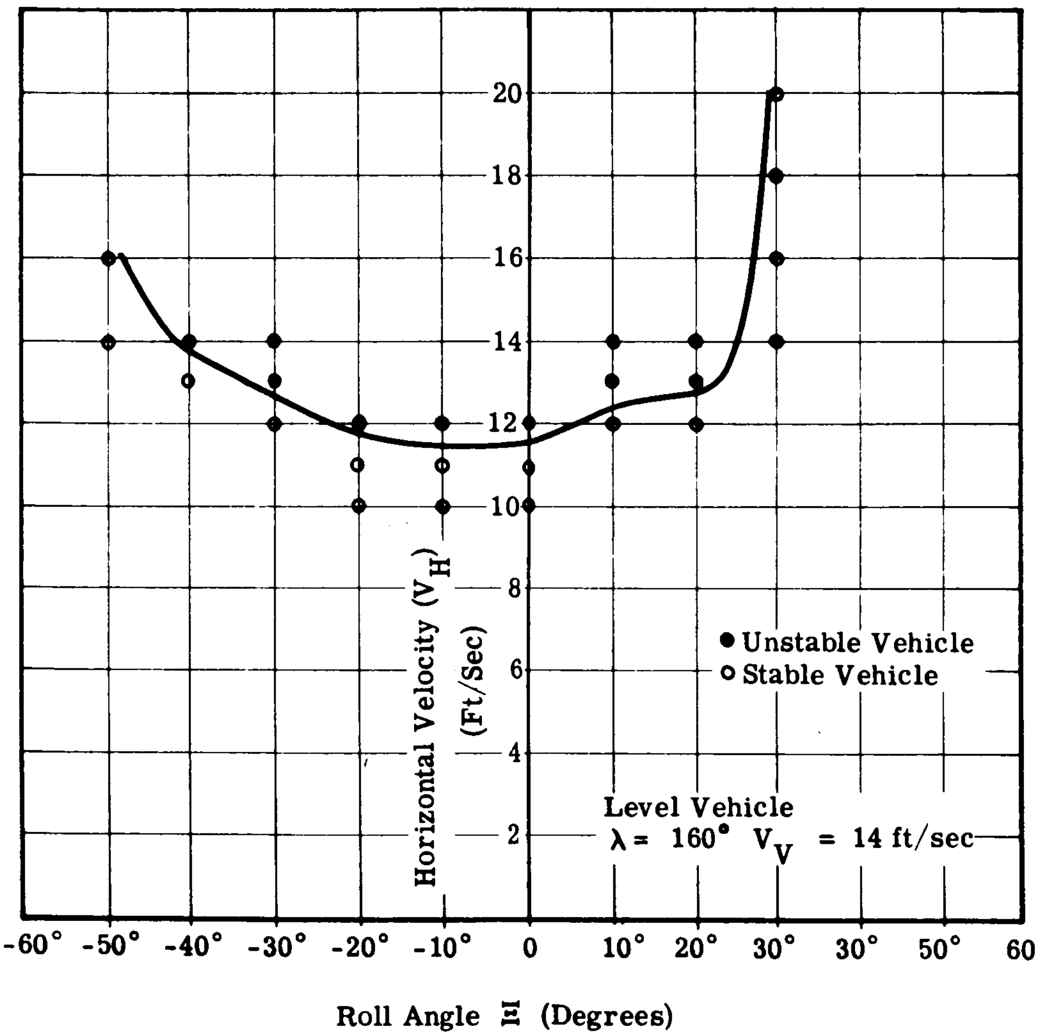



Figure 4-11. $V_H$ versus Ξ Stability Profile

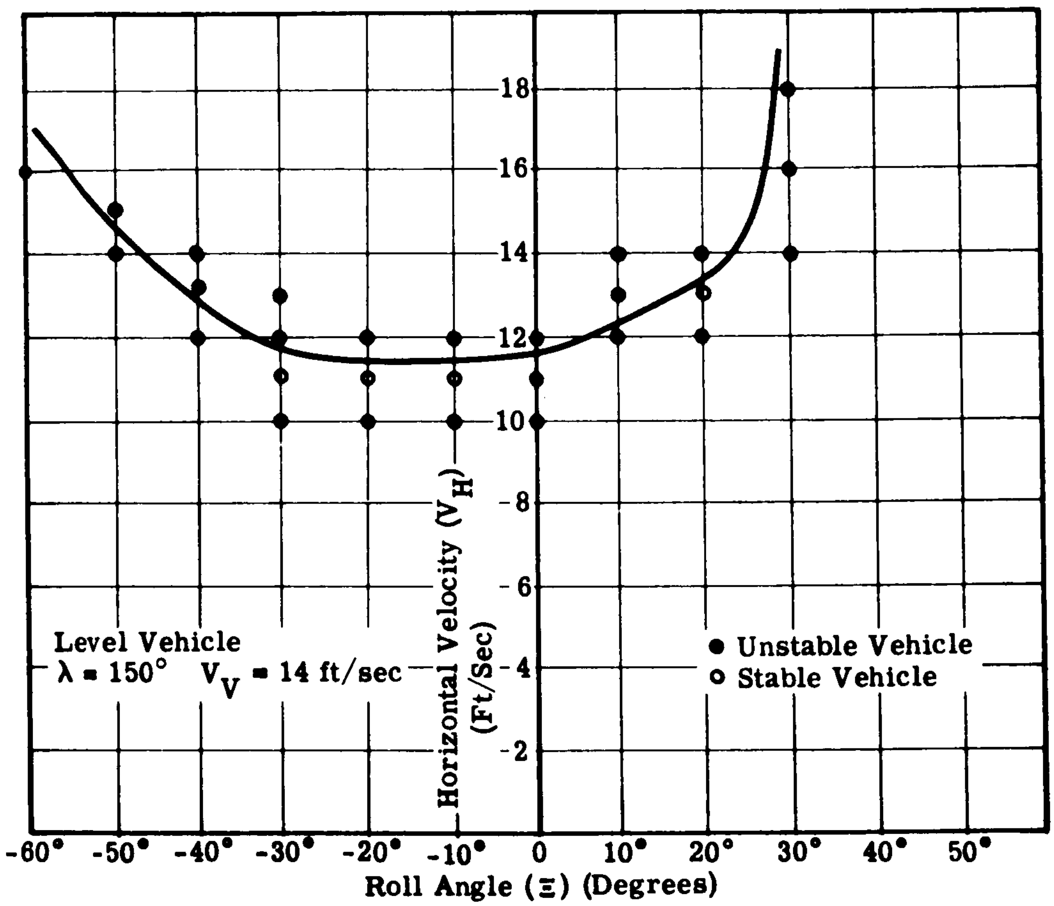



Figure 4-12. $V_H$ versus Ξ Stability Profile

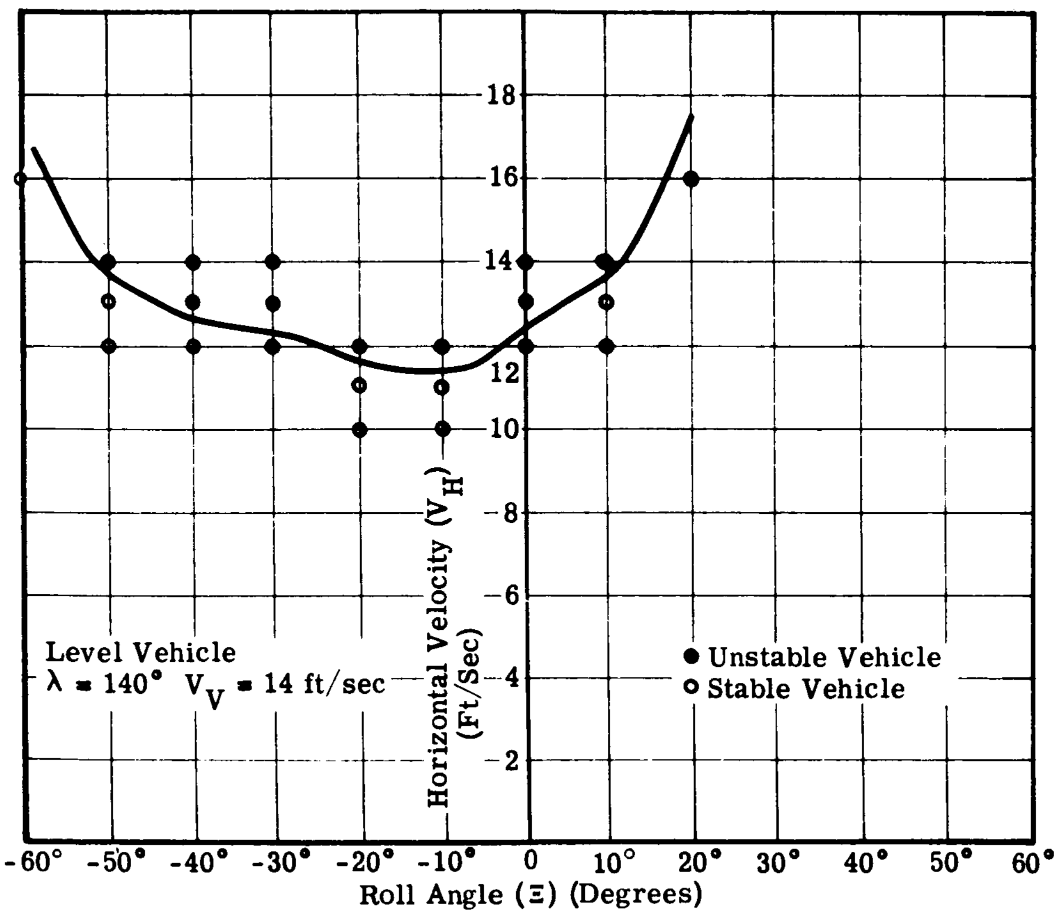



Figure 4-13. $V_H$ versus Ξ Stability Profile

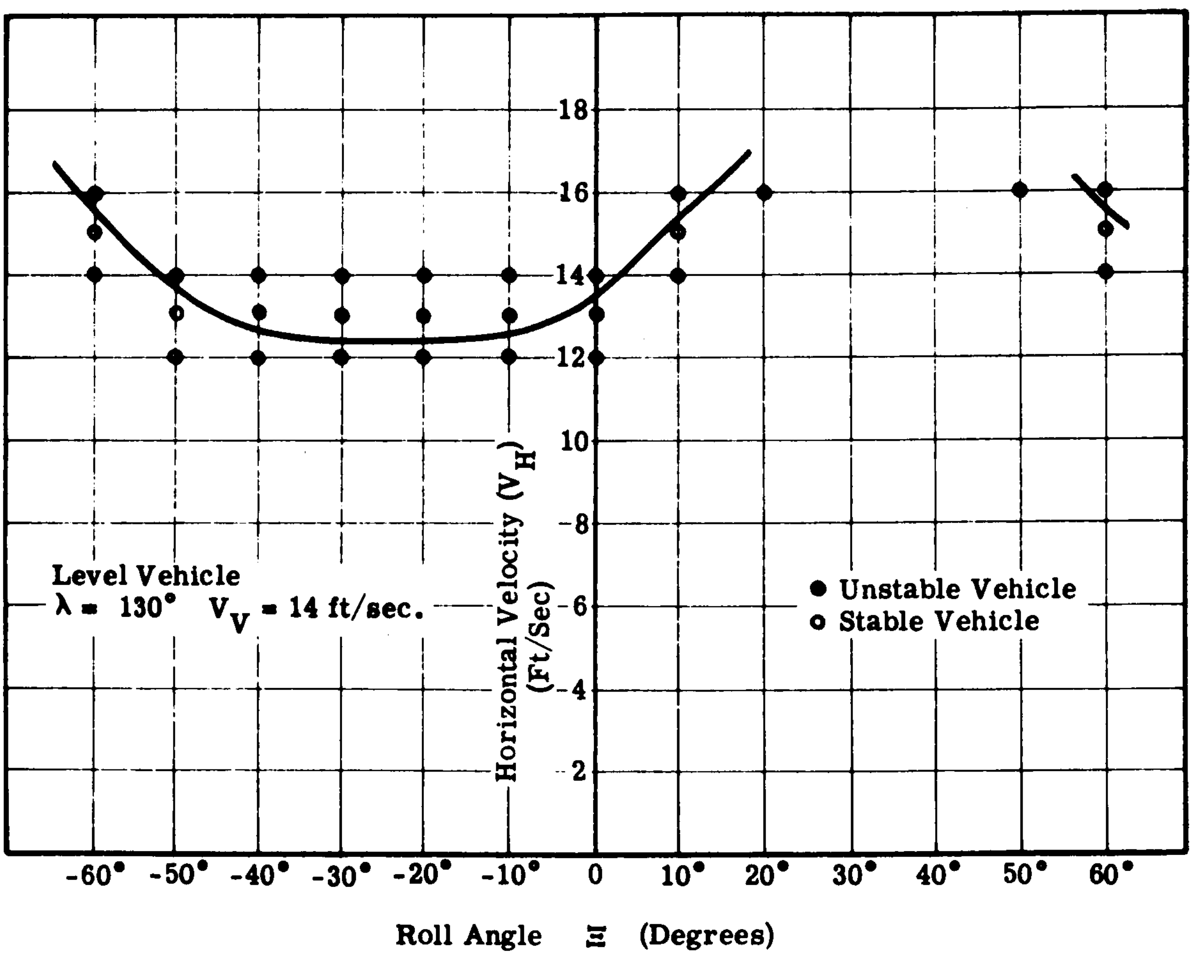



Figure 4-14. $V_H$ versus $\Xi$ Stability Profile

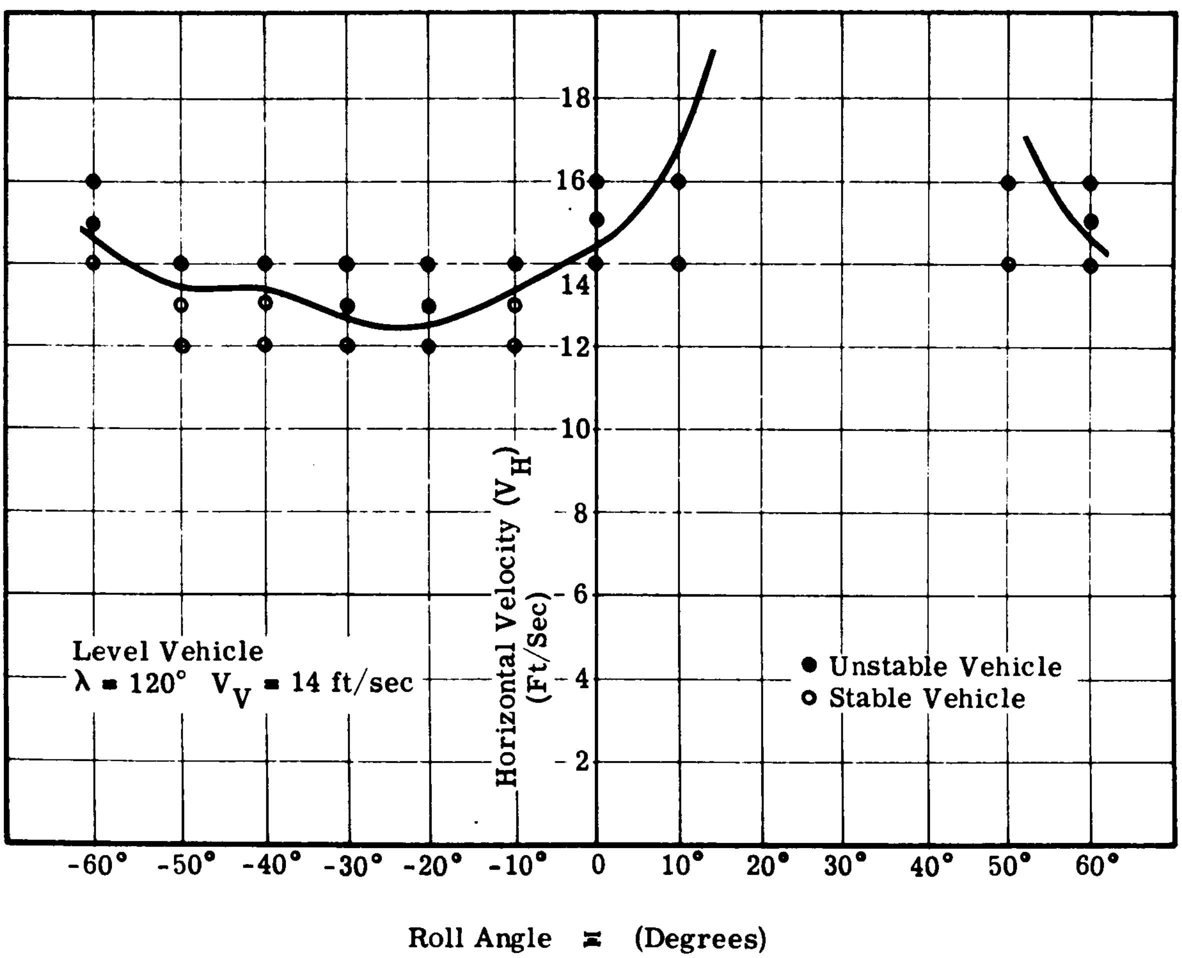



Figure 4-15. $V_H$ versus $\Xi$ Stability Profile

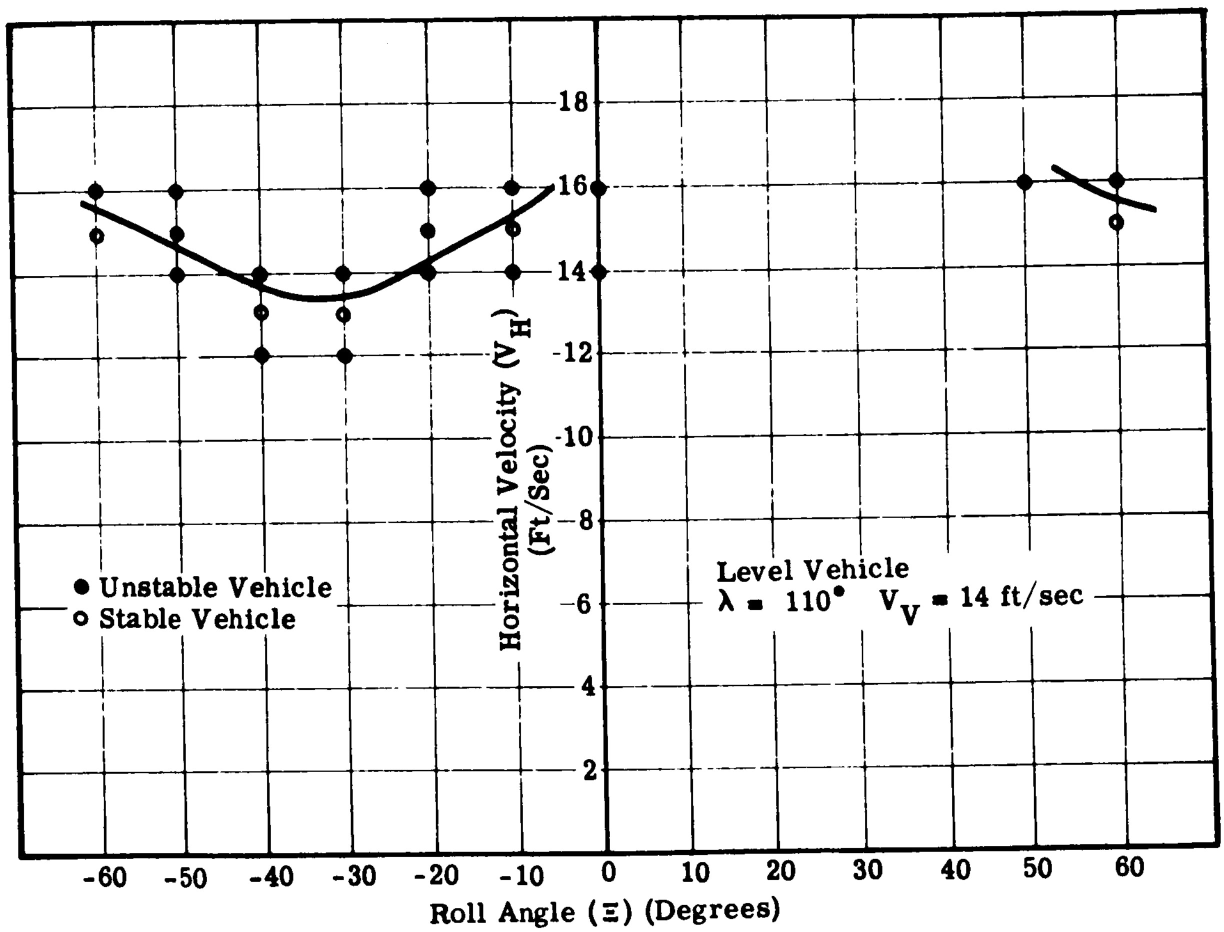



Figure 4-16. $V_H$ versus Ξ Stability Profile

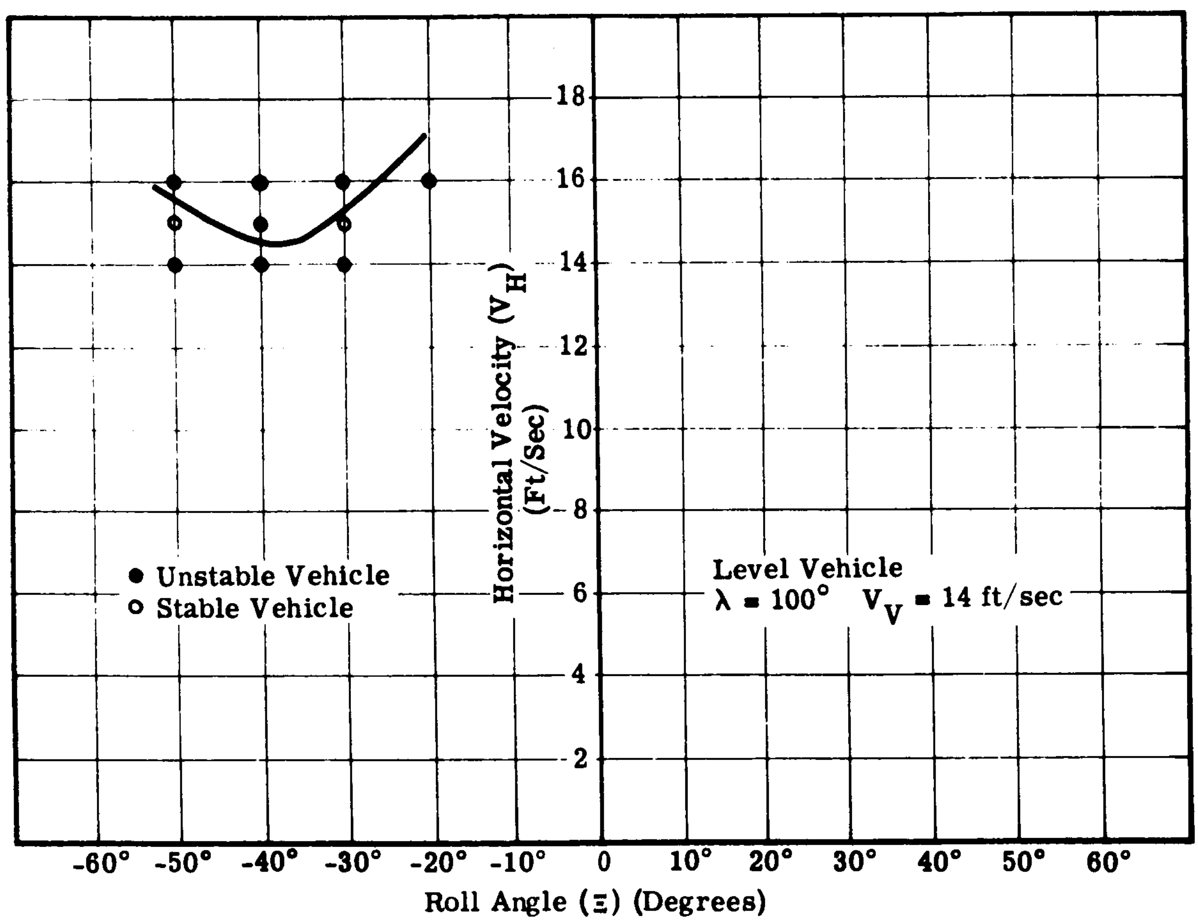



Figure 4-17. $V_H$ versus $\Xi$ Stability Profile

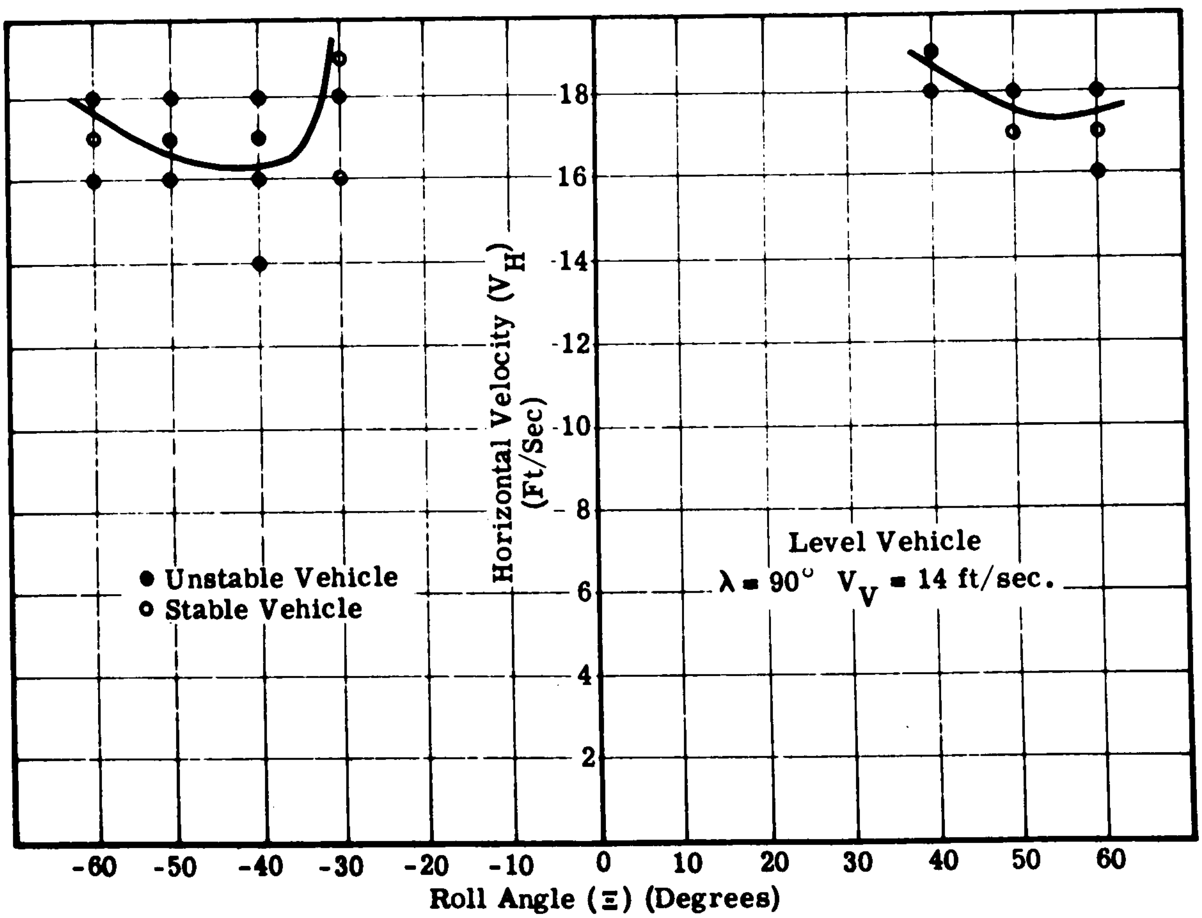



Figure 4-18. $V_H$ versus $\Xi$ Stability Profile

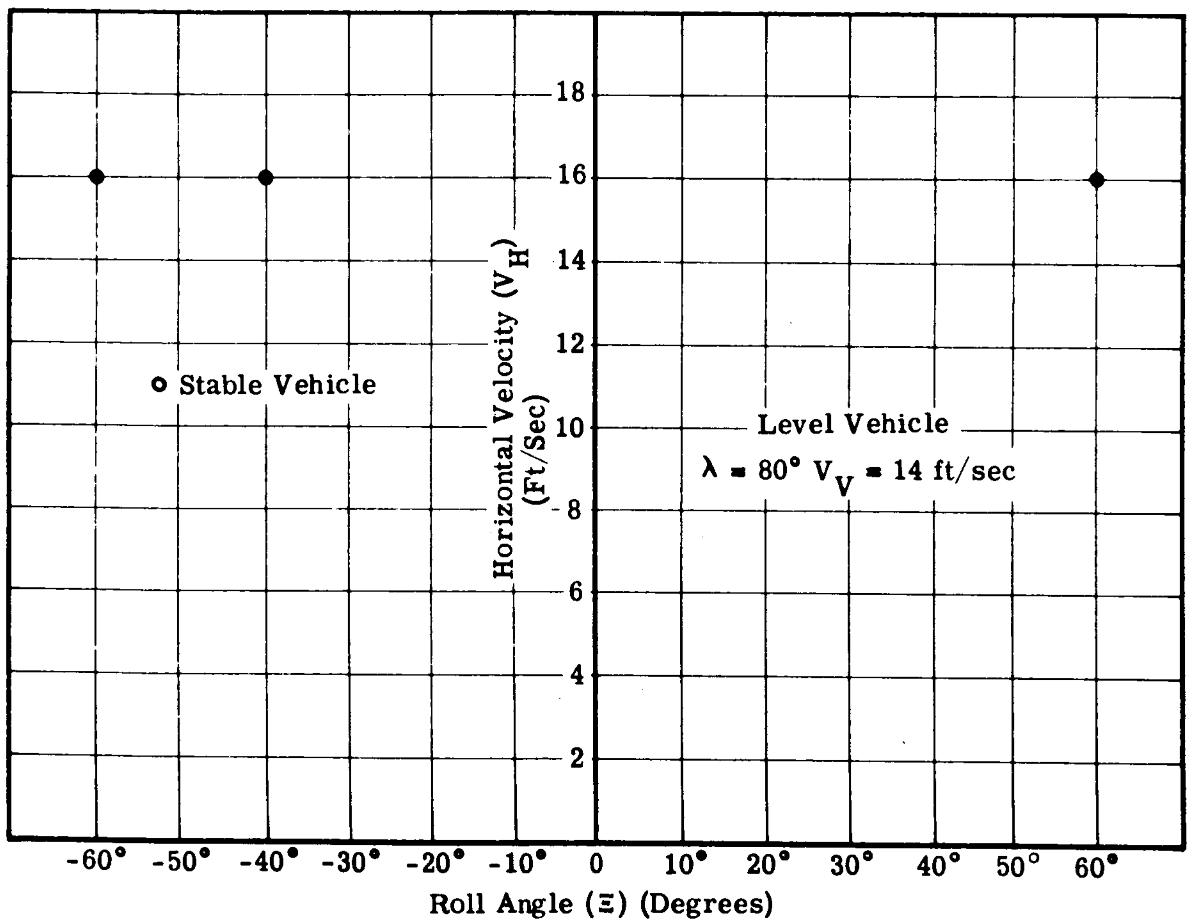



Figure 4-19. $V_H$ versus Ξ Stability Profile

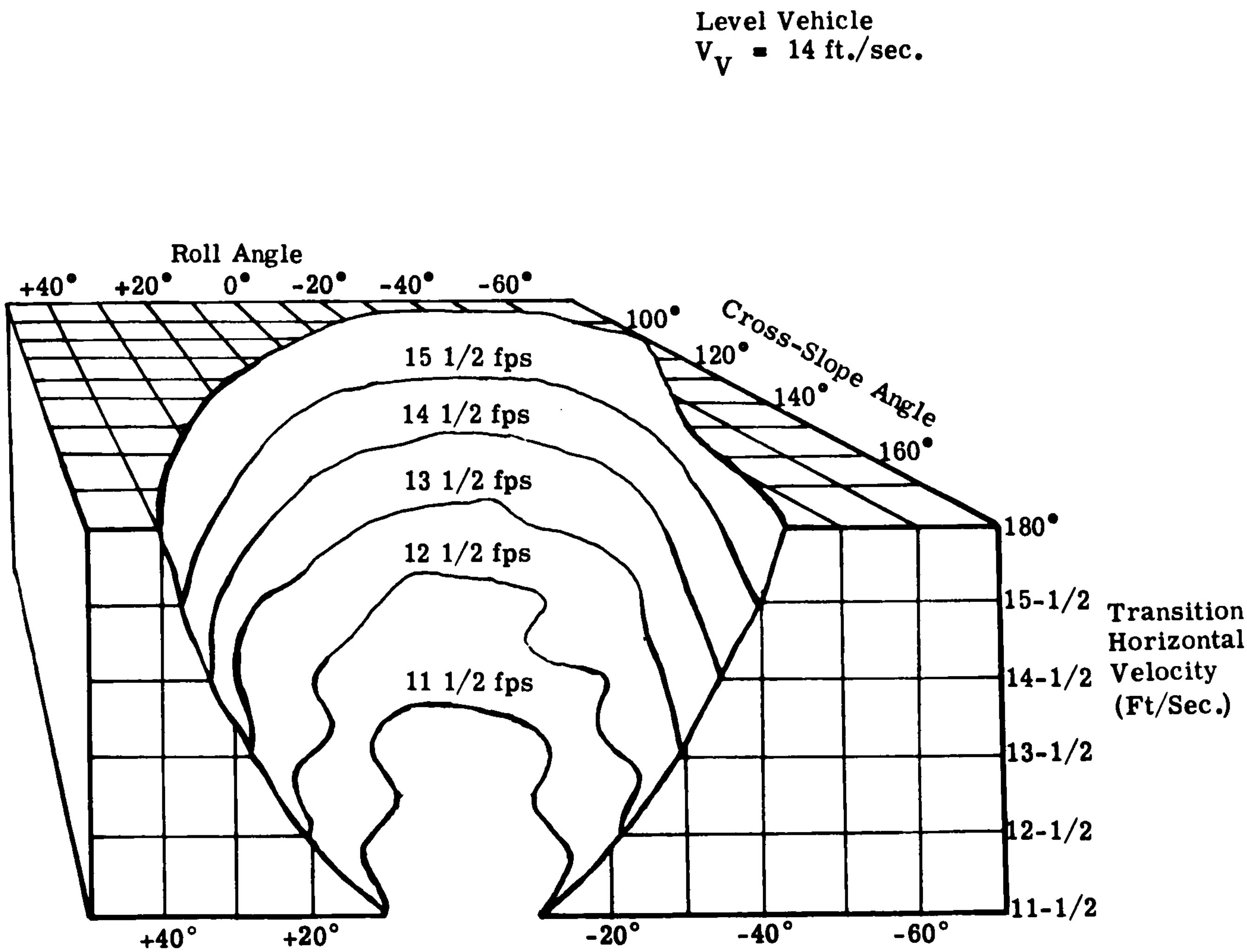



**Figure 4-20. Perspective Sketch of Surveyor Stability Surface**

BENDIX REPORT NO. MM-66-19

ADD. 1

# FINAL REPORT ON SURVEYOR LUNAR TOUCHDOWN STABILITY STUDY

Bendix MM-66-19, ADD. 1    7 Dec 1966

Approved: John A. Garba

John A. Garba

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

PASADENA, CALIFORNIA

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

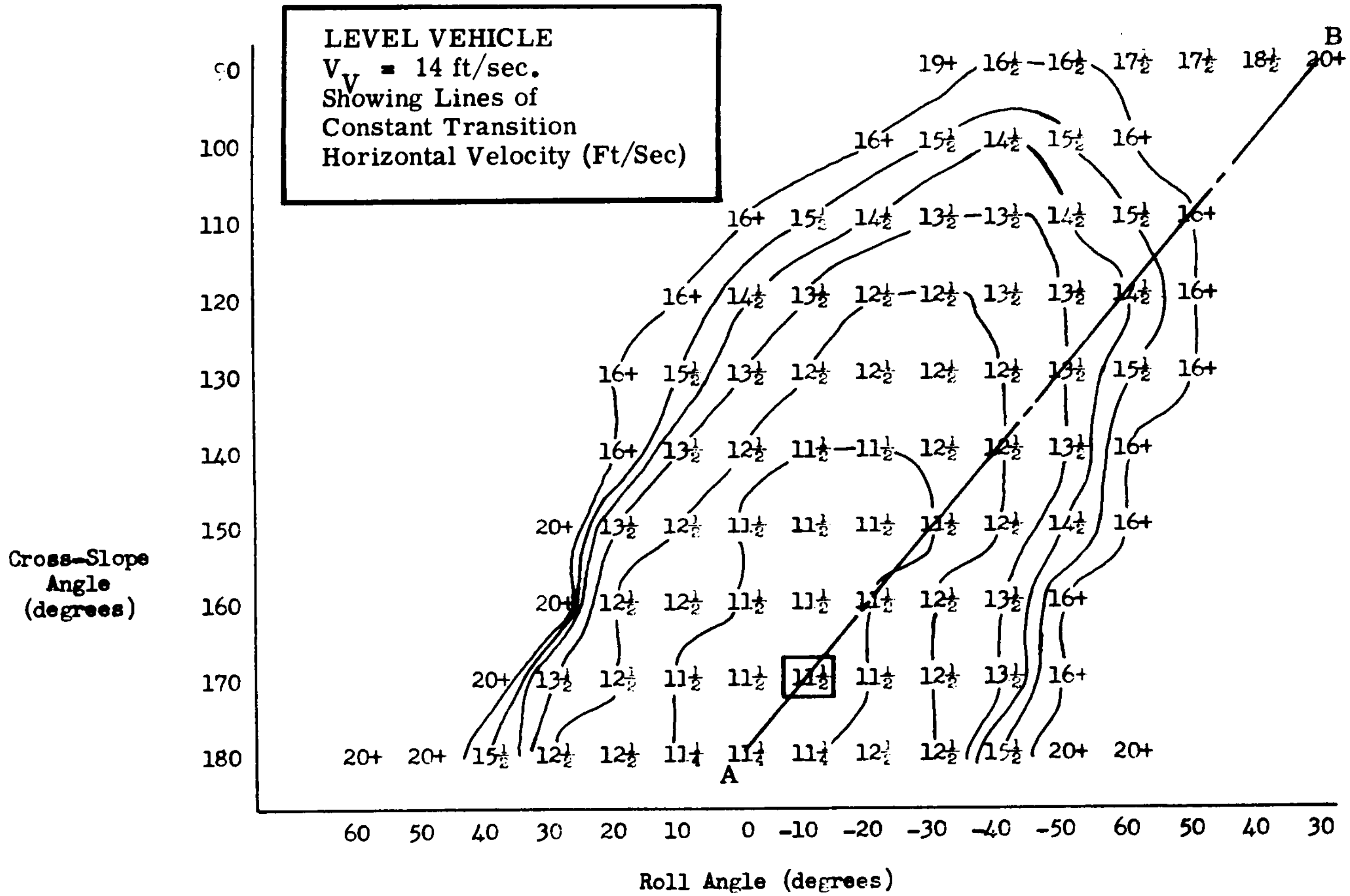



Figure 4-21. Top View of Surveyor Stability Surface

Examination of detailed time history data for touchdown conditions in the vicinity of line AB offers some explanation of the phenomenon. It appears that for these landings the crushable blocks influence the motion markedly. As one or more blocks contact the surface during a cross-slope landing, the velocity in the X-direction (across the slope) is virtually arrested, and the cross-slope angle increases toward 180°. When the blocks lose contact, this velocity again becomes significant, but less so than at touchdown. The cross-slope angle does not return to its initial value, but to a value somewhat closer to 180° than existed initially. This results in landings which appear as if the horizontal velocity vector were pointed more "downhill" than the actual touchdown conditions would indicate.

Table 4-3, which lists a portion of the time history for one set of landing conditions, illustrates this phenomenon. The touchdown conditions considered were a level vehicle, 14 feet/second vertical velocity, 12 feet/second horizontal velocity, 170° cross-slope angle and -10° roll angle. This was an unstable run which was used to determine the transition point marked by a square in Figure 4-21. Much larger variations in cross-slope angle have been observed in other cases. This case is presented for illustrative purposes only.

TABLE 4-3. ABBREVIATED TIME HISTORY

| Time (Sec) | CG Velocities $\dot{X}$ (Ft/Sec) | $\dot{Y}$ (Ft/Sec) | Cross-Slope Angle (Degrees) |
|---|---|---|---|
| 0 | -2.08 | -11.82 | 170.0 |
| .14 | -1.85 | -10.90 | 170.3 |
| .22 | -1.15 | - 9.24 | 172.9 |
| .24 | - .913 | - 8.31 | 173.7 |
| .26 | - .656 | - 7.21 | 174.8 |
| .28 | - .127 | - 4.35 | 178.3 |
| .30 | - .083 | - 2.91 | 178.3 |
| .32 | - .135 | - 2.41 | 176.8 |
| .34 | - .175 | - 2.37 | 175.8 |
| .36 | - .230 | - 2.36 | 174.4 |
| .38 | - .290 | - 2.34 | 172.9 |
| .42 | - .335 | - 2.37 | 172.0 |
| .50 | - .381 | - 2.40 | 171.0 |

Recalling that $\tan\lambda = -\frac{\dot{X}}{\dot{Y}}$, the variation in cross-slope angle is largely dependent upon the variation in $\dot{X}$. The cross-slope angle first increases and then decreases as $\dot{X}$ first decreases and then increases. The apparent cause of the increase in $\dot{X}$ is worth mentioning, since none of the crush forces have components across the slope. Evidently both leading feet are sliding due to shock absorber restroking at this time, and there is an unbalance of friction force due to an unbalance in crush force.

Another significant feature of the contour map which was investigated in detail was the discontinuity in transition horizontal velocity in the area bounded by cross-slope angles of 160° and 170° and roll angles of +20° to +40°. The transition horizontal velocity changes abruptly in this area from about 13 feet/second to over 20 feet/second for a roll angle change of only 10°. The change in contour, as might be expected, is apparently due to a gross change in the vehicle motion. For the extremely stable cases to the left, little or no change of cross-slope angle occurs during the landing. For the cases to the right of the discontinuity, the cross-slope angle changes in such a way as to put the horizontal velocity vector between the downhill legs, the least stable orientation. The change in cross-slope angle occurs shortly after one or more of the blocks contact the surface.

The main distinguishing features of the contour map seem to be due to the effect of the blocks on vehicle dynamics. It would seem reasonable that these features would be less pronounced for lower vertical velocity landings and more pronounced for landings at higher vertical velocities.

Although the footpads and shock absorbers are also absorbing energy during the landings, they do not seem to affect the direction of the horizontal velocity vector as do the crushable blocks. Footpad impacts can be associated with small changes in the roll angle (less than 5°), but these changes are small compared with some of the observed changes in velocity vector orientation (up to 30°).

To recapitulate the results of the parametric study for the level vehicle, no cases were observed to have lower transition horizontal velocities than the baseline case (11-1/4 feet/second). The transition horizontal velocity increases as the touchdown horizontal velocity vector is directed further away from the downhill orientation. As the touchdown horizontal velocity vector turns away from downhill, the range of critical roll angles (for a given cross-slope angle) shifts toward the negatively increasing direction.

### 4.2.3 Parametric Variations for Tipped Vehicle

The final aspect of the Surveyor lunar landing for which a parametric study was conducted was the effect of tipping the vehicle. Two combinations of roll angle and cross-slope angle were chosen from Figure 4-21. These combinations were chosen for their relatively low transition horizontal velocities. The combinations were a 180° cross-slope angle with 0° roll angle, and a 150° cross-slope angle with a -10° roll angle.

Before describing these investigations, it is necessary to define the method used to specify the tipped vehicle. Referring to Figure 4-22, the cross-slope angle ($\lambda$) is measured between the Y-axis in ground coordinates and the horizontal velocity vector. The cross-slope angle is measured positive in a right hand sense about the Z-axis in

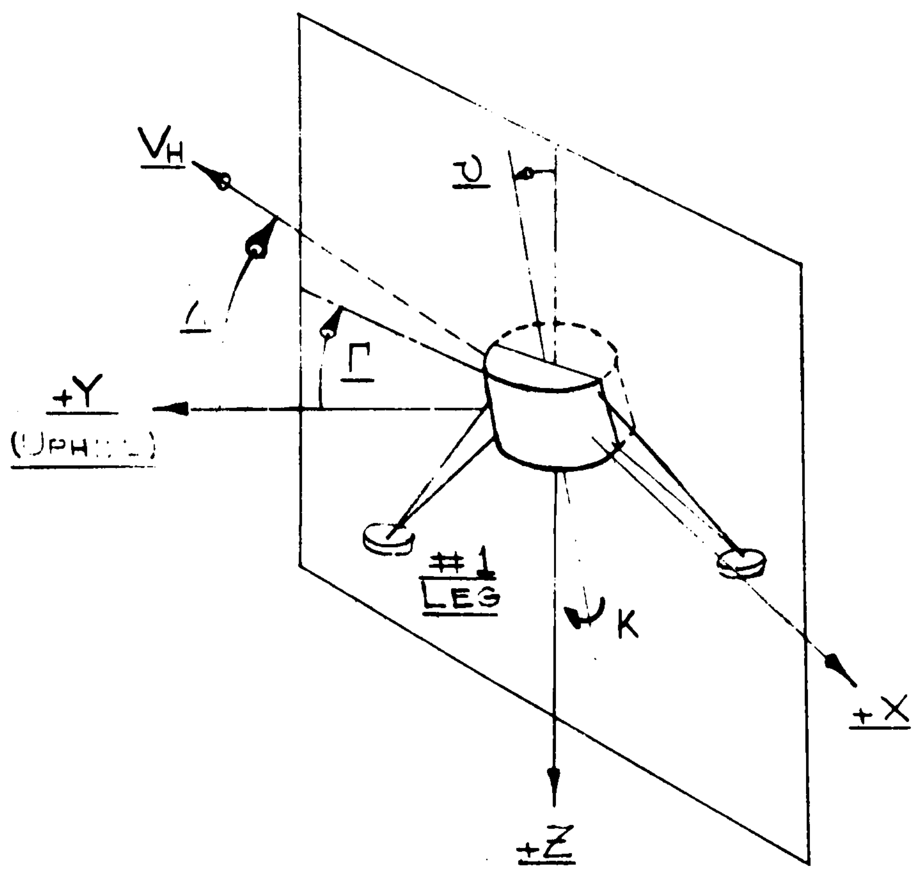



Figure 4-22. Geometry for Tipped Vehicle

ground coordinates. To specify a tipped orientation, consider a vehicle with the vehicle axes originally aligned parallel to the ground coordinate system (see Section II). The vehicle centerline is to be tipped in a certain direction by a positive angle $\nu$, measured from the vertical. The plane formed by the original and new positions of the vehicle centerline is located relative to the Y-axis in ground coordinates by the angle $\Gamma$. The angle $\Gamma$ is also measured positive in a right hand sense relative to the Z-axis in ground coordinates. Any required roll orientation is accomplished by rotation through an angle K about the vehicle centerline. The angle K is defined similarly to the Euler roll angle $\Xi$, but the two differ in magnitude.

For a given tipped vehicle specification the Euler angles can be obtained from:

$$\sin \Psi = \frac{\cos \Gamma \sin \nu}{\sqrt{1 - \sin^2 \Gamma \sin^2 \nu}}$$

$$\sin \Phi = \sin \Gamma \sin \nu$$

$$\sin (\Xi - K) = \frac{\sin \Gamma \cos \Gamma (\cos \nu - 1)}{\sqrt{1 - \sin^2 \Gamma \sin^2 \nu}}$$

Note that for small $\nu$, the difference between $\Xi$ and K is small.

Figure 4-23 is a stability profile of touchdown horizontal velocity versus the angle $\Gamma$. The vehicle centerline was tipped 9°, and the cross-slope angle was 180° (downhill landing). The nominal roll angle (K) was specified as 0°. This profile was the first instance in which transition horizontal velocities smaller than the baseline case were observed. Figure 4-23 shows transition velocities of about 10-1/2 feet/second for $\Gamma$ between 112-1/2° and 157-1/2°. The same transition velocity exists for $\Gamma$ between -112-1/2° and -157-1/2° since there is symmetry of the profile with respect to $\Gamma$.

In an effort to further understand this critical area, the detailed time history data was once again consulted, but with less impressive results than previously. A comparison was made for the cases of $\Gamma$ = 90° and $\Gamma$ = 112-1/2° with a horizontal velocity of 11 feet/second in both cases. It was observed that for $\Gamma$ = 112-1/2° (unstable), the pitch angle at the end of the free flight phase ($\Psi$ = -36.8°) was slightly more pronounced than for the stable case of $\Gamma$ = 90° ($\Psi$ = -34.2°). This small difference in pitch angle, which results mainly from differences in pitch rate during the free flight, is evidently enough to result in toppling. The cause of the pitch rate difference is not apparent, however.

Figure 4-24 presents a similar stability profile, but with the vehicle tipped only 5°. Comparison of Figures 4-23 and 4-24 reveals that the profiles differ only for $\Gamma = \pm 90°$. At these values of $\Gamma$, a vehicle initially tipped 5° topples for a horizontal velocity of 11 feet/second, while a vehicle initially tipped 9° is stable.

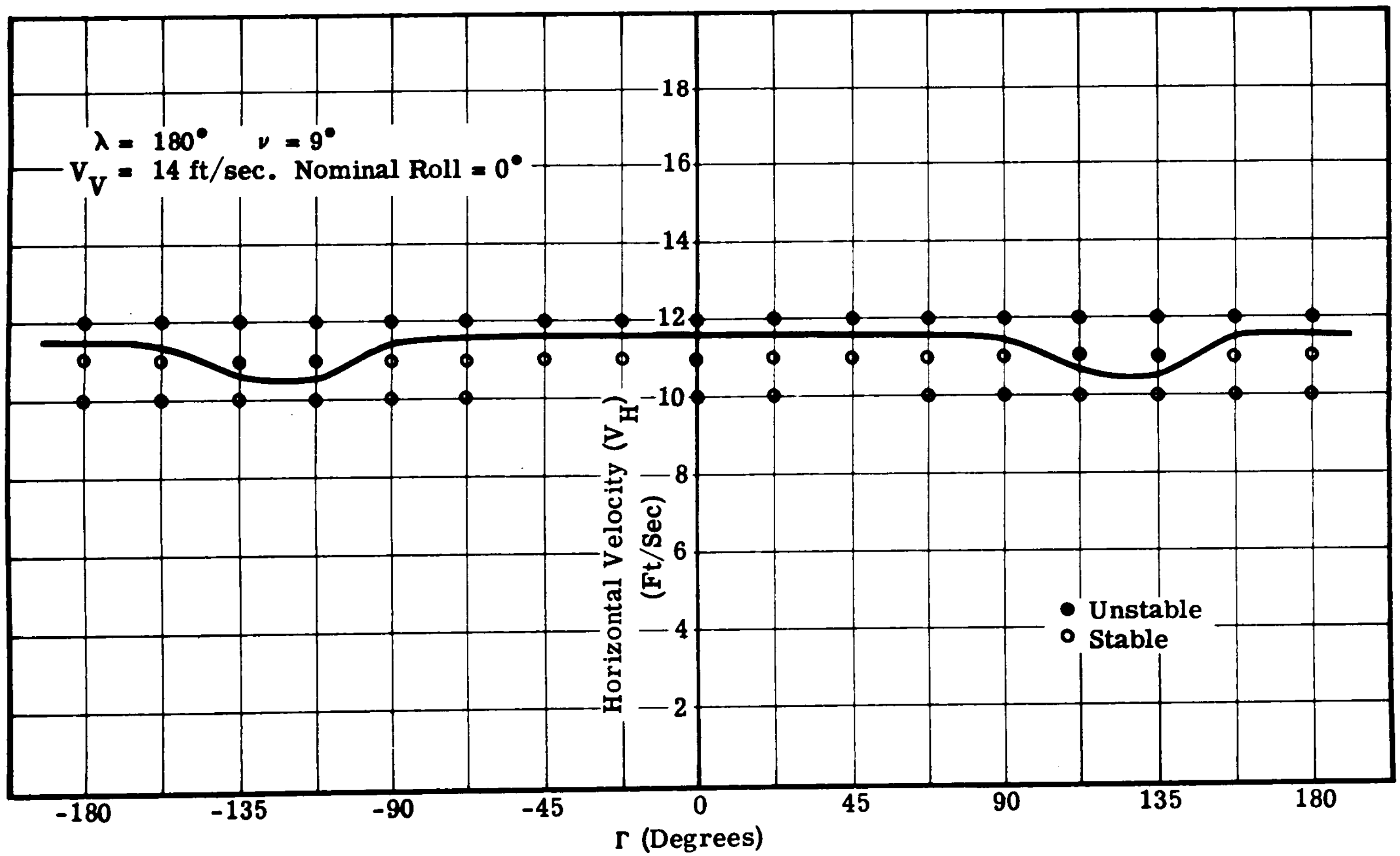



Figure 4-23. $V_H$ versus Γ Stability Profile

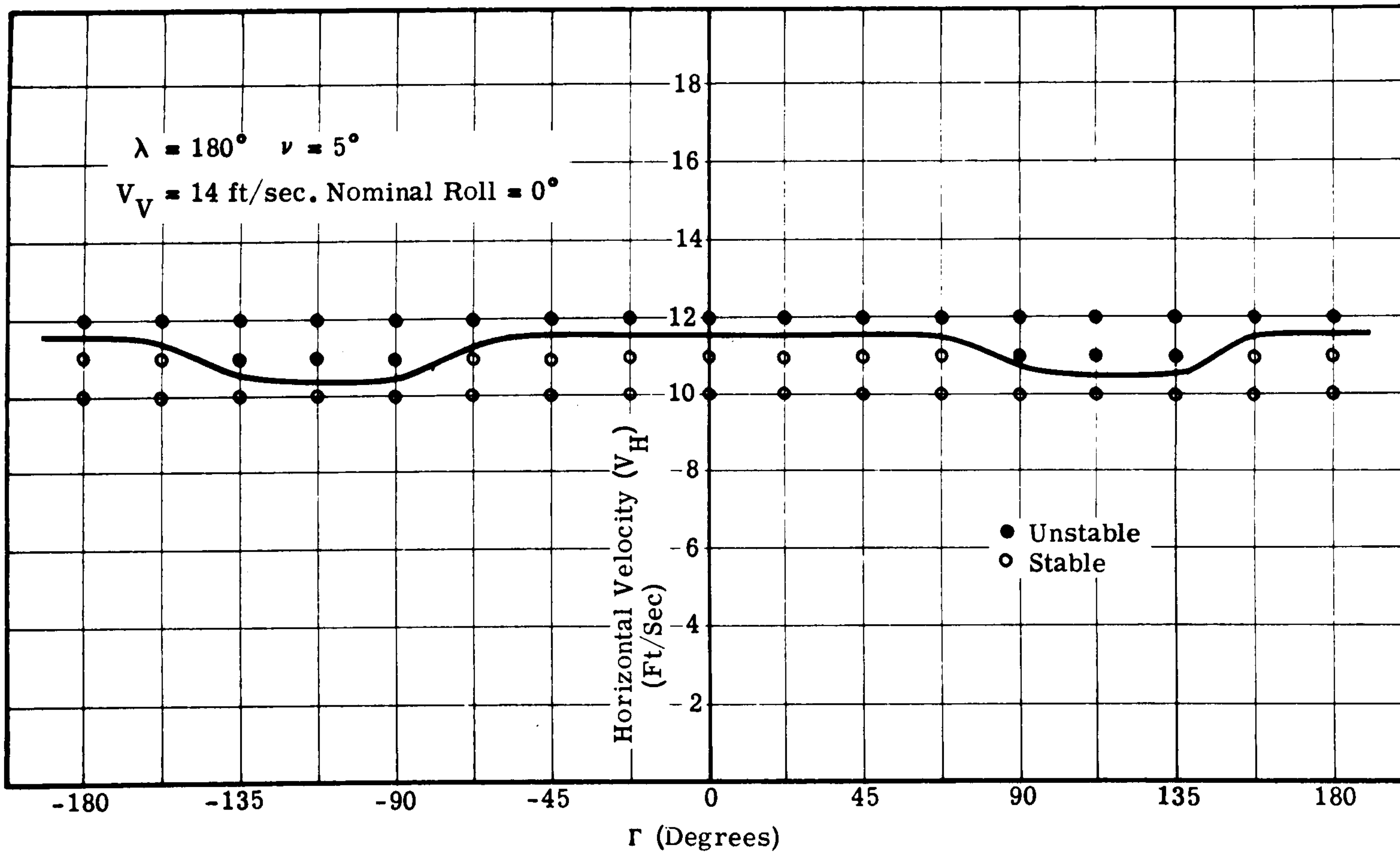



Figure 4-24. $V_H$ versus Γ Stability Profile

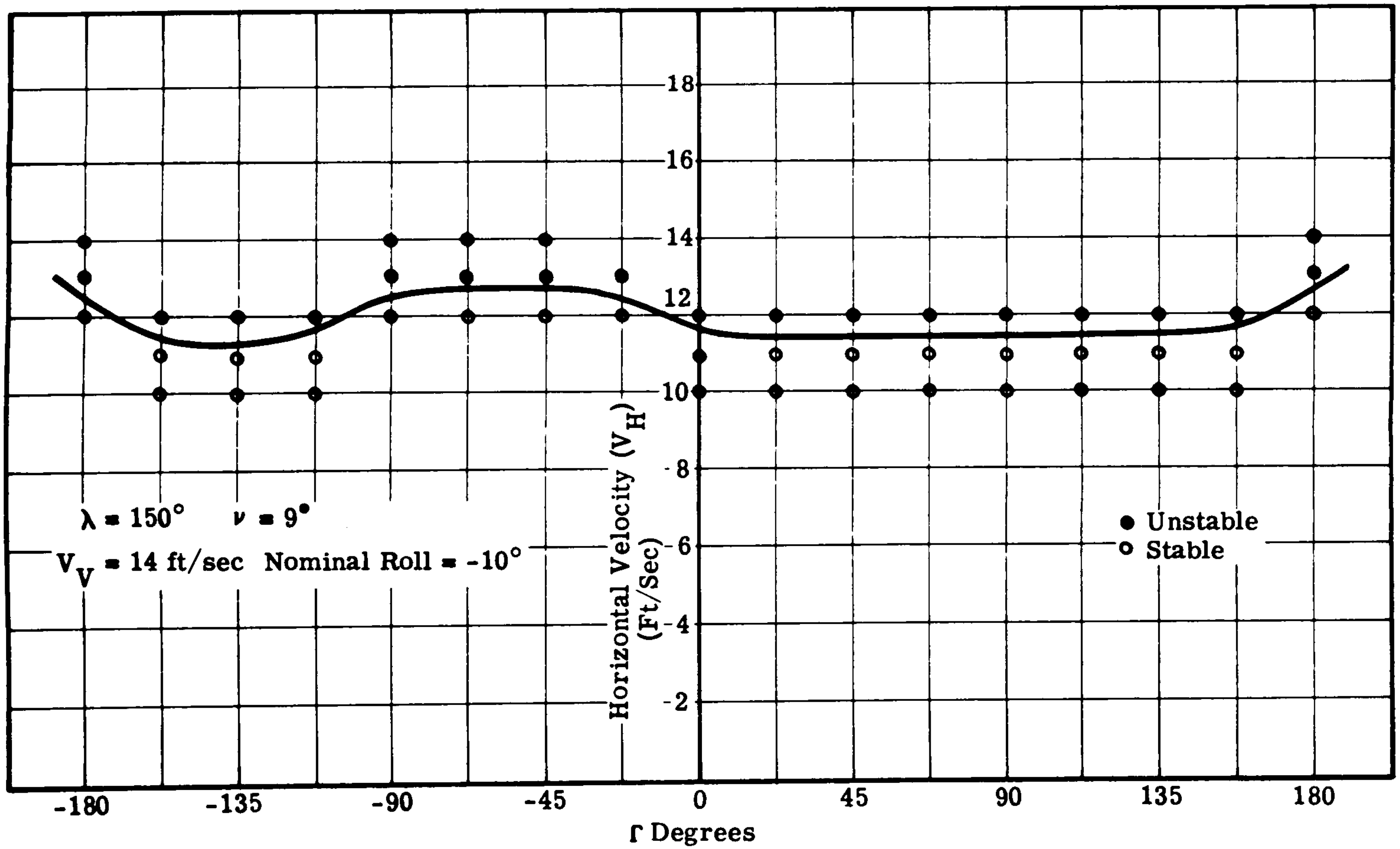



Figure 4-25. $V_H$ versus Γ Stability Profile

It may be noted that the $\Gamma = 0°$ and 180° transition points from Figures 4-23 and 4-24 correspond to pure pitch angles. These points, together with the baseline case, could be used to define a very limited stability profile of transition horizontal velocity versus pitch angle, over a range of pitch angles between ±9°. These transition velocities all lie between 11 and 12 feet/second, so that no particular trend can be identified.

The final situation which was studied concerned a vehicle tipped 9° with a cross-slope angle of 150° and a nominal roll angle of -10°. The horizontal velocity versus $\Gamma$ stability profile (Figure 4-25) shows transition horizontal velocities of 11-1/2 feet/second and above. These transitions are higher than the most critical velocities shown in Figures 4-23 and 4-24.

The results of the parametric studies for the tipped vehicle are naturally less extensive than for the level vehicle, since a much greater proportion of the effort was devoted to the level vehicle. For a cross-slope angle of 180° and a nominal roll angle of 0°, there are values of $\Gamma$ for which the transition horizontal velocities are less than for the baseline case. Limited data concerning the effect of pitch angle on stability show no particular trend.

### 4.2.4 Conclusions from Parametric Studies

The most important conclusion to be drawn from the parametric studies is that the stability picture for non-planar landings is more critical than for the baseline conditions in a few cases only. For the combinations of touchdown conditions studied, the few cases which were more critical than the baseline case exhibited transitional horizontal velocities which were about 1 foot/second less than the baseline transition.

The stability of landings which are primarily downhill is quite high for low vertical velocities. Even at higher vertical velocities, substantial horizontal velocities are required to produce toppling. For landings which are primarily uphill, the limiting consideration appears to be excessive shock absorber stroking, since the vehicle is highly stable in the uphill direction.

In general, stability increases as the touchdown configuration becomes remote from situations where the horizontal velocity vector is directed between the two downhill feet. No particular stability trend was evident for small changes in pitch angle.

## SECTION V

## REFERENCES

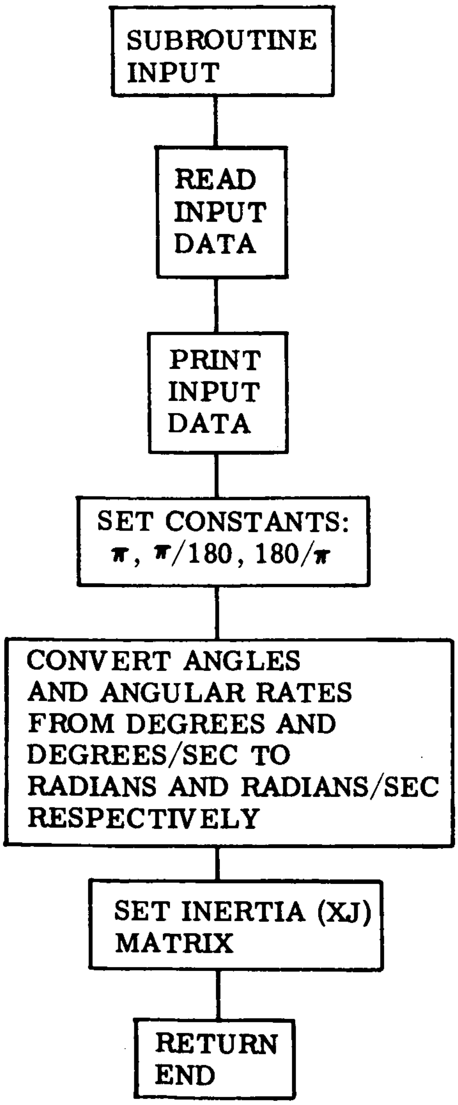

a. Appendix A to JPL Contract 951304 entitled, "Surveyor Lunar Touchdown Stability Study".

b. Bendix Report MM-65-16, "Interim Report on Surveyor Lunar Touchdown Stability Study", by R. G. Alderson and D. A. Wells, dated 30 November 1965.

# FOREWORD

The current Surveyor Lunar Touchdown Stability Computer Program version used at the Jet Propulsion Laboratory contains a provision to circumvent the stability subroutine STAB. This provision is not documented in the main body of the report. It was included to facilitate the use of this program in the flight data interpretation for a stable landing. The real time clock interrogation has been eliminated from the JPL version.

This addendum contains only those pages of the main report wherein changes are reflected.




Reads and Modifies Input Data

Consists of Reading:

| | |
|---|---|
| XM | XDOTOC |
| XIX | YDOTOC |
| YIY | ZDOTOC |
| ZIZ | XKD (1), XKD (2), XKD (3) |
| XIY | XKS |
| XIZ | FP (1), FP (2), FP (3) |
| YIZ | XNU |
| XMP | XLDMIN |
| XLDO | CB (1), CB (2), CB (3) |
| DELCG | CF (1), CF (2), CF (3) |
| XLP | THETD (1) |
| XLL | THETD (2) |
| XLA | THETD (3) |
| G | THETSD |
| ALPHAD (1) | PSIOD |
| ALPHAD (2) | PHIOD |
| ALPHK (1) | XIOD |
| ALPHK (2) | XLAMDD |
| BETAD | OMEGA (1) |
| XLC | OMEGA (2) |
| HBO | OMEGA (3) |
| XIALPH | RR (1), RR (2), RR (3) |
| XIGAM | RUNNO |
| HFO | SERNO |
| DB | NPHASE |
| DF | VH |
| RC (1), RC (2), | VV |
| RC (3) | IWILEY |
| SRL (1), SRL (2) | ICODE |
| SRL (3) | XDTP |
| XMUF | TFO |
| XMUB | HM |
| TAU (1), TAU (2), | |
| TAU (3) | ERR (4) through ERR (24) |
| | CRUX |

| | |
|---|---|
| J23 | Number of independent variables in the Block No. 3 table. |
| J31 | Number of independent variables in the Shock Absorber No. 1 table. |
| J32 | Number of independent variables in the Shock Absorber No. 2 table. |
| J33 | Number of independent variables in the Shock Absorber No. 3 table. |
| K | First digit of the table code number. |
| KSUM | Sum of the first digits of the code numbers. |
| L | Second digit of the table code numbers. |
| LF | Second table value in integer form. |
| LSUM | Sum of the second digits of the code numbers. |
| L3 | Total number of values in a given table. |
| M | Leg number corresponding to a given table. |
| SHOCK1 (N10) | Table of Characteristics for Shock Absorber No. 1. |
| SHOCK2 (N10) | Table of Characteristics for Shock Absorber No. 2. |
| SHOCK3 (N10) | Table of Characteristics for Shock Absorber No. 3. |
| STUPE | Temporary storage for the values on a given card. |

INPUT

| | |
|---|---|
| BETA (N7) | The angle between the vehicle centerline and a line joining the upper and lower hardpoints on a leg set (rad). |
| BETAD | Same as BETA except for units (degrees). |
| CODE | A numeric which controls program output. |
| CRUX | Stability key. |
| DA (N9) | Utility array used for temporary storage. |
| DEGRAD (N7) | Conversion factor (rad/deg). |
| DELCG (N11) | Distance between CG and lower hardpoint measured along the vehicle centerline (feet). |
| DTP (N24) | Time interval for output printing and punching (sec). |

```
$IBFTC MAIN    M94,XR7
      COMMON/N1/  N,TO,TF,SH,HM,YI(50),YO(50),ERR(50)
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N12/XDOTOC,YDOTOC,ZDOTOC,TAU(3),ISET(3),XA(3),YA(3),ZA(3)
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),PSB(3),XHPU(3),
     1         YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELEL(3),F1MAX(3),
     2         ZPREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ B(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DRS(3,3,3)
      COMMON/N24/IOPT(10), JN, DTP
      CALL    TABLIN
      IPRINT = 0
      TF      = 0.0
      HM      = 0.0
    1 TO      = 0.0
      WRITE (6,20)
   20 FORMAT (1H1)
      CALL    INPUT
      IF      (IOPT(3)) 21,21,22
   21 N       = 29
      GO TO   23
   22 N       = 39
   23 CALL    INIT
      IF      (IOPT(4)) 26,26,24
   24 TAMIN   = AMIN1(TAU(1), TAU(2), TAU(3))
      IF      (TAMIN) 26,26,25
   25 SH      = TAMIN + HM
      GO TO   27
   26 SH      = .0005
   27 DO      10 I = 1,3
      YI(I)   = ALPDOT(I)
      YI(I+3)= ALPH(I)
      YI(I+6)= GAMDOT(I)
      YI(I+9)= GAMMA(I)
   10 YI(I+18) = OMEG(I)
      YI(13) = XDOTC
      YI(14) = YDOTC
      YI(15) = ZDOTC
      YI(16) = XC
      YI(17) = YC
      YI(18) = ZC
      YI(22) = PSIO
      YI(23) = PHIO
      YI(24) = XIO
      DO      18 J= 25,50
   18 YI(J)   = 0.0
      DO      185 J= 4,12
  185 ERR(J) = ERR(J)*DEGRAD
      DO      186 J= 19,24
  186 ERR(J) = ERR(J) * DEGRAD
      CALL    RKP
      GO TO 1
      END
```

```
      SUBROUTINE INPUT
      DIMENSION XMPP(3), ALPHK(2)
      DIMENSION          OMEGA(3), THETD(3), ALPHAD(3) ,ERO(50)
      COMMON/N1/  N,TO,TF,SH,HM,YI(50),YO(50),ERR(50)
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMPP,XIP,DELCG,XLP,
     1       XLL,XLA,G,ALPHK,XLC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2       XLDOP,XLDO,XKD(3),XKS,   XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3       XMUB,RUNNO,SERNO,VV,VH,DB,XIALPH,XIGAM,DELMAX,SRL(3),FP(3)
      COMMON/N12/XDOTOC,YDOTOC,ZDOTOC,TAU(3),ISET(3),XA(3),YA(3),ZA(3)
      COMMON/N23/MISS
      COMMON/N24/IOPT(10), JN, DTP
      COMMON/N25/IWILEY(10)
      COMMON/N30/CRUX
      IOFF   = 0
       DTP   = 0.0
  100 DO 101 K=1,5
  101 DA(K) = 0.0
      READ (5,900) MM,( DA(J), J=1,5)
      IF     (MM)109,200,109
  109 GO TO  (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 ),MM
    1 XM   = DA(1)
      XIX = DA(2)
      YIY = DA(3)
      ZIZ = DA(4)
      XIY = DA(5)
      MISS    = 0
      WRITE (6,921) MM
      GO TO 100
    2 XIZ   = DA(1)
      YIZ   = DA(2)
      XMP   = DA(3)
      XLDO = DA(4)
      DELCG= DA(5)
      MISS    = 0
      WRITE (6,921) MM
      GO TO 100
    3 XLP = DA(1)
      XLL = DA(2)
      XLA = DA(3)
      G          = DA(4)
      ALPHAD(1)= DA(5)
      WRITE (6,921) MM
      GO TO 100
    4 ALPHAD(2)= DA(1)
      ALPHK(1) = DA(2)
      ALPHK(2) = DA(3)
      BETAD     = DA(4)
      XLC       = DA(5)
      WRITE (6,921) MM
      GO TO 100
```

```
    5 HBO    = DA(1)
      XIALPH= DA(2)
      XIGAM = DA(3)
      HFO   = DA(4)
      DB    = DA(5)
      WRITE (6,921) MM
      GO TO 100
    6 DF    = DA(1)
      XDOTOC= DA(2)
      YDOTOC= DA(3)
      ZDOTOC= DA(4)
      CRUX= DA(5)
      WRITE (6,921) MM
      GO TO 100
    7 XNU   = DA(1)
      XLDMIN= DA(2)
      XKS    = DA(3)
      WRITE (6,921) MM
      GO TO 100
    8 THETD(1) = DA(1)
      THETD(2) = DA(2)
      THETD(3) = DA(3)
      WRITE (6,921) MM
      GO TO 100
    9 THETSD = DA(1)
      PSIOD  = DA(2)
      PHIOD  = DA(3)
      XIOD   = DA(4)
      XLAMDD = DA(5)
      WRITE (6,921) MM
      GO TO 100
   10 OMEGA(1)= DA(1)
      OMEGA(2)= DA(2)
      OMEGA(3)= DA(3)
      WILEY  = DA(4)
      ICODE  = IFIX (ABS (WILEY))
      DO 1001 J= 1,10
 1001 IWILEY(J) = 0
      GO TO  (1011,1012,1013,1020,1015,1016,1017,1018,1020,1020),ICODE
 1011 IWILEY(1) = 1
      GO TO  1020
 1012 IWILEY(2) = 1
      GO TO  1020
 1013 IWILEY(3) = 1
      GO TO  1020
 1015 IWILEY(1) = 1
      IWILEY(2) = 1
      GO TO  1020
 1016 IWILEY(1) = 1
      IWILEY(3) = 1
      GO TO  1020
 1017 IWILEY(2) = 1
      IWILEY(3) = 1
      GO TO  1020
 1018 DO 1014 J= 1,3
 1014 IWILEY(J) = 1
 1020 CONTINUE
      WRITE (6,921) MM
      GO TO 100
```

```
      WRITE(6,921) MM
      GO TO  100
   17 DO 171 J=1,2
  171 RC(J+1)  = DA(J)
      DO 172 J = 1,3
  172 XKD(J)  = DA(J+2)
      WRITE(6,921) MM
      GO TO  100
   18 DO 181 J = 1,3
  181 FP(J)   = DA(J)
      DO 182 J = 1,2
  182 RR(J)  = DA(J+3)
      WRITE   (6,921) MM
      GO TO   100
   19 RR(3)   = DA(1)
      DO 191  J = 1,3
  191 SRL(J)  = DA(J+1)
      CB(1)  = DA(5)
      WRITE   (6,921) MM
      GO TO   100
   20 DO 250  J= 1,2
  250 CB(J+1)  = DA(J)
      DO 260  J = 1,3
  260 CF(J)  = DA(J+2)
      WRITE   (6,921) MM
      GO TO   100
   21 XMUF    = DA(1)
      XMUB    = DA(2)
      DO 290  J = 1,3
  290 TAU(J)  = DA(J+2)
      WRITE   (6,921) MM
      GO TO   100
  200 IF      (IOFF) 310,310,300
  300 DTP     = 0.0
      GO TO   320
  310 DTP     = XDTP
  320 WRITE   (6,922) SERNO, NPHASE
      MM=1
      WRITE (6,901) MM, XM, XIX, YIY, ZIZ, XIY
      MM=2
      WRITE (6,902) MM, XIZ, YIZ, XMP, XLDO, DELCG
      MM=3
      WRITE (6,903) MM, XLP, XLL, XLA, G   , ALPHAD(1)
      MM=4
      WRITE (6,904) MM, ALPHAD(2) , ALPHK(1), ALPHK(2), BETAD, XLC
      MM=5
      WRITE (6,905)  MM, HBO, XIALPH, XIGAM, HFO, DB
      MM=6
      WRITE   (6,906) MM, DF,     XDOTOC, YDOTOC, ZDOTOC,CRUX
      MM=7
      WRITE   (6,907) MM, XNU, XLDMIN,XKS
      MM=8
      WRITE   (6,908) MM,           THETD(1), THETD(2), THETD(3)
      MM=9
      WRITE   (6,909) MM, THETSD, PSIOD, PHIOD, XIOD, XLAMDD
      MM=10
      WRITE   (6,910) MM, (J,OMEGA(J) , J=1,3) ,WILEY
      MM=11
```

```
      TF     = TFO
      JN     = JNO
      IF     (IWILEY(1) + IWILEY(2) + IWILEY(3)) 1102,1102,1101
 1101 JN     = 0
 1102 CONTINUE
  900 FORMAT(I7,5X,5E12.5)
  901 FORMAT(5H CARD,I3,10H     XM  =,E12.5,10H    IXX  =,E12.5,10H    I
     1YY  =,E12.5,10H    IZZ  =,E12.5,10H    IXY  =,E12.5/)
  902 FORMAT(5H CARD,I3,10H    IXZ  =,E12.5,10H    IYZ  =,E12.5,10H    X
     1MP  =,E12.5,10H   XLDO  =,E12.5,10H  DELCG  =,E12.5/)
  903 FORMAT(5H CARD,I3,10H    XLP  =,E12.5,10H    XLL  =,E12.5,10H    X
     1LA  =,E12.5,10H       G =,E12.5,10H ALPHA(1)=,E12.5/)
  904 FORMAT(5H CARD,I3,10H ALPHA(2)=,E12.5,10H ALPHK(1)=,E12.5,10H ALPH
     1K(2)=,E12.5,10H   BETA  =,E12.5,10H    XLC  =,E12.5/)
  905 FORMAT(5H CARD,I3,10H    HBO  =,E12.5,10H  XIALPH =,E12.5,10H   XI
     1GAM =,E12.5,10H    HFO =,E12.5,10H      DB =,E12.5/)
  906 FORMAT (1X4HCARDI3,6X4HDF = E12.5,2X8HXDOTOC = E12.5, 2X8HYDOTOC =
 9061  E12.5, 2X8HZDOTOC = E12.5,10H STAB KEY= E12.5/)
  907 FORMAT (1X4HCARD I3,5X5HXNU = E12.5,2X8HXLDMIN = E12.5,5X5HXKS =
 9071 E12.5 /)
  908 FORMAT(5H CARD,I3,                                       10H THET
     1D(1)=,E12.5,10H THETD(2)=,E12.5,10H THETD(3)=,E12.5/)
  909 FORMAT(5H CARD,I3,10H  THETAS =,E12.5,10H   PSIO  =,E12.5,10H    P
     1HIO =,E12.5,10H    XIO =,E12.5,10H  XLAMDD =,E12.5/)
  910 FORMAT (1X4HCARD I3, 3(1X6HOMEGA(I1,2H)=E12.5),3X7HCODE2 = E12.5/)
  911 FORMAT (5H CARD,I3,10H    CODE =,F12.5,10H  SER NO =,F12.5,10H PHA
 9111SE NO=,I12  ,10H      VH =,E12.5,10H      VV =,E12.5/)
  912 FORMAT (1X4HCARDI3,5X5HDTP = E12.5,6X4HTF = E12.5,6X4HHM = E12.5,
 9121 2X8HERR(4) = E12.5, 2X8HERR(5) = E12.5 / )
  913 FORMAT (1X4HCARDI3, 4(2X4HERR(I1,3H) = E12.5), 1X4HERR(I2,3H) = E1
 9131 2.5/)
  914 FORMAT (1X4HCARDI3, 5(1X4HERR(I2,3H) = E12.5)/)
  915 FORMAT (1X4HCARDI3, 5(1X4HERR(I2,3H) = E12.5)/)
  916 FORMAT (1X4HCARDI3,4(1X4HERR(I2,3H) = E12.5), 3X7HRC(1) = E12.5/)
  917 FORMAT (1X4HCARDI3, 2(3X3HRC(I1,3H) = E12.5), 3(2X4HXKD(I1,3H) = E
 9171 12.5)/)
  918 FORMAT (1X4HCARDI3, 3(3X3HFP(I1,3H) = E12.5),2(3X3HRR(I1,3H) = E12
 9181 .5)/)
  919 FORMAT (1X4HCARDI3,3X7HRR(3) = E12.5,3(2X4HSRL(I1,3H) = E12.5), 3X
 9191 7HCB(1) = E12.5 / )
  920 FORMAT (1X4HCARDI3, 2(3X3HCB(I1,3H) = E12.5),3(3X3HCF(I1,3H) = E12
 9201 .5)/)
 1921 FORMAT (1X4HCARDI3,4X6HXMUF = E12.5,4X6HXMUB = E12.5, 3(2X4HTAU(I1
19211 , 3H) = E12.5)/)
  921 FORMAT(39H   NEW DATA CARD FOR THIS CASE IS NO. =,I5)
  922 FORMAT (//27H  THIS CASE IS SERIES NO. = F12.3,
 9221        2CH     AND PHASE NO. = , I4// 21H  INPUT DATA FOLLOWS. //)
      RETURN
      END
```

```
      SUBROUTINE FEP
      DIMENSION XMPP(3), ALPHK(2),SHKMAX(3)
      DIMENSION A(3), AA(3)
      COMMON/N2/T,Y(50),D(50)
      COMMON/N5/ TALPHA(3), TGAMMA(3), F(3), TQ(3)
      COMMON/N6/OMEGDT(3), ALPHDD(3), GAMDD(3), Q(3), PSID, PHID, XID,
     1 DELWRK
      COMMON/N7/ ALPDOT(3), ALPH(3), GAMDOT(3), GAMMA(3), XDOTC, YDOTC,
     1 ZDOTC, XC, YC, ZC, OMEG(3), PSI, PHI, XI, PFLAG, KFLAG, PSIO,
     2 PHIO, XIO, THETA(3), THETAS, ALPHA(2), BETA, XJ(3,3), PI,
     3 DEGRAD, RADDEG
      COMMON/N8/ LFLAG
      COMMON/N9/ A1(3), A2(2), A3(2), A4(3,2,3), A5(3,2,3), A6(3), A7,
     1 A8(3,3), D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12,
     2 D13,D14,D15,D16,D17,D18,D19,D20, XX, DA(10), W(3,4,6), HPL(3,3)
      COMMON/N11/CB(3),XM,XIX,YIY,ZIZ,XIY,XIZ,YIZ,XMPP,XIP,DELCG,XLP,
     1        XLL,XLA,G,ALPHK,XLC,HBO,NPHASE,HFO,HFMIN,DF,RC(3),RR(3),
     2        XLDOP,XLDO,XKD(3),XKS,   XNU,XLDMIN,CC,CF(3),XLAMDA,XMUF,
     3        XMUB,RUNNO,SERNO,VV,VH,DB,XIALPH,XIGAM,DELMAX,SRL(3),FP(3)
      COMMON/N13/ A, AA
      COMMON/N14/SLOPE1(3),SLOPE2(3),XCDP(12),YCDP(12),PSB(3),XHPU(3),
     1        YHPU(3),ZHPU(3),SINTHS,COSTHS,PSP(3,12),DELEL(3),F1MAX(3),
     2        ZPREF,SINTH(3),COSTH(3),TZERO,VZERO,ZBO(3)
      COMMON/N15/ B(3,3), XB(3), YB(3), ZB(3), XFP(3), YFP(3), ZFP(3),
     1 DRS(3,3,3)
      COMMON/N16/ RN(3), XII(3,2), XLD(3), XLDDOT(3), XDOTP(3,2),
     1 YDOTP(3,2), S(3,2), COSXI (3), HF(3,12), ZDOTP(3,2), HB(3),
     2 GRS(3,3,3), TA
      COMMON/N17/ XFPC(3), YFPC(3) , ZFPC(3)
      COMMON/N18/ NFLAG
      COMMON/N19/NNFLAG(3),PSBQ(3),PSPQ(3,12),IBEEN(6)
      COMMON/N20/DWORK(15)
      COMMON/N24/IOPT(10), JN, DTP
      COMMON/N30/CRUX
      IF      (LFLAG) 19,19,10
   10 IF      (LFLAG-1) 70,70,15
   15 ISTABL = 0
      GO TO   70
   19 IF      (NFLAG) 20,20,24
   20 DO      23 I= 4,24
      IF      (ABS(Y(I)) - 10.E4)23,21,21
   21 DO      22 J= 1,24
   22 D(J)    = 0.0
      D(22)   = 10.E6
      GO TO   150
   23 CONTINUE
   24 DO       30 I= 1,3
      ALPDOT(I) = Y(I)
      ALPH(I)   = Y(I+3)
      GAMDOT(I)= Y(I+6)
      GAMMA(I) = Y(I+9)
   30 OMEG(I) = Y(I+18)
      XDOTC   = Y(13)
      YDOTC   = Y(14)
      ZDOTC   = Y(15)
      XC      = Y(16)
      YC      = Y(17)
```

```
      78310X40HDISSIPATED ENERGY (EDIS)                    = F20.5,7H FT LBS/
      78410X40HFINAL KINETIC ENERGY (EKF)                  = F20.5,7H FT LBS//
      78510X30HMAXIMUM SHOCK ABSORBER FORCES- /
      78620X17HLEG NO.1 FA(1) =  F20.10/
      78720X17HLEG NO.2 FA(2) =  F20.10/
      78820X17HLEG NO.3 FA(3) =  F20.10//
      78910X51HMINIMUM CLEARANCE BETWEEN GROUND PLANE AND FRAME = F10.4, 4H
      781 FT.  /)
         RETURN
   60 IF      (IOPT(4)) 605,605,130
  605 IF      (T) 61,61,62
   61 SIGMIN = 10.E8
      GCMIN  = 10.E8
      DO      611 J=1,3
  611 SHKMAX(J) = 0.0
   62 SIGMIN =AMIN1 (SIGMIN, W(3,4,2))
      GCMIN  = AMIN1(GCMIN,W(1,4,2))
      DO      621 J=1,3
  621 SHKMAX(J) = AMAX1(SHKMAX(J), W(J,4,1))
   63 IF      (SIGMIN - 0.7854 ) 65,68,68
   65 IF      (W(3,4,2) - SIGMIN - 0.874) 68,68,66
   66 IF(CRUX) 666,68,666
  666 WRITE  (6,67)
   67 FORMAT (42H$$$$$ VEHICLE IS STABLE (ROCK-BACK) $$$$$ /)
      ISTABL = 1
      GO TO  70
   68 IF      (ZPREF-(YC*SINTHS+ZC*COSTHS))  75,75,130
   75 IF      (ZDOTC*ZDOTC-4.0) 80,80,130
   80 IF      (YDOTC*YDOTC- 4.0) 90,90,130
   90 IF      (XDOTC*XDOTC-4.0) 100,100,130
  100 IF      (OMEG(1)*OMEG(1) - .01) 110,110,130
  110 IF      (OMEG(2)* OMEG(2) - .01)120,120,130
  120 IF      (OMEG(3)*OMEG(3) - .01 ) 125,125,130
  125 IF(CRUX) 1025,130,1025
 1025 WRITE (6,126)
  126 FORMAT (45H***** VEHICLE IS STABLE *****                            )
      ISTABL = 1
      GO TO  70
  130 CALL FORCE
      IF      (PFLAG) 131,132,132
  131 ISTABL = -1
      GO TO  70
  132 CALL    INTEQM
      DO      140 I=1,3
      IF    (NFLAG) 136,136,134
  134 D(I)    = ALPHDD(I)
  136 D(I+3)  = ALPDOT(I)
      D(I+3)  = AMAX1 (-100.0,( AMIN1 (ALPDOT(I), 100.0)))
      D(I+6)  = GAM DD(I)
      D(I+9)  = GAMDOT(I)
      D(I+12) = Q(I)
      D(I+15)= Y(I+12)
  140 D(I+18)= OMEGDT(I)
      D(22)   = PSID
      D(23)   = PHID
      D(24)   = XID
      DO      145 J=1,15
  145 D(J+24) = DWORK(J)
  150 RETURN
      END
```

```
2062 16X4HFAZ3,16X4HFBX1,16X4HFBY1,16X4HFBZ1,16X4HFBX2,16X4HFBY2/
2063 16X4HFBZ2,16X4HFBX3,16X4HFBY3,16X4HFBZ3,16X4HMBX1,16X4HMBY1/
2064 16X4HMBZ1,16X4HMBX2,16X4HMBY2,16X4HMBZ2,16X4HMBX3,16X4HMBY3/
2065 16X4HMBZ3,16X4HFCX1,16X4HFCY1,16X4HFCZ1,16X4HFCX2,16X4HFCY2/
206616X4HFCZ2,16X4HFCX3,16X4HFCY3,16X4HFCZ3,16X4HFFX1,16X4HFFY1/
2067 16X4HFFZ1,16X4HFFX2,16X4HFFY2,16X4HFFZ2,16X4HFFX3,16X4HFFY3 /
206816X4HFFZ3,15X5HPSP11,15X5HPSP12,15X5HPSP21,15X5HPSP22,15X5HPSP31/
206915X5HPSP32//)
     DO       207 J= 1,55
 207 ODEAR(J) = 0.0
     LSET     = 1
     GO TO    300
 208 IF       (TP) 209,209,210
 209 WRITE    (6,220) (ODEAR(J), J=1,55)
     IF       (IWILEY(2) + JN - 1) 300,300,2095
2095 PUNCH    290, (ODEAR(J), J= 1,55)
     GO TO    300
 210 WRITE (6,220) TP, (W(J,4,1), J=1,3), ((W(J,L,1),J=1,3),L=1,3),
2101 ((W(J,L,2),J=1,3),L=1,3), ((W(J,L,4),J=1,3),L=1,3),
2102 ((W(J,L,3), J=1,3), L=1,3),((W(J,L,5),J=1,3),L=1,3),
2103 ((PSP(J,K), K=1,2), J=1,3)
 220 FORMAT (/9(6F20.5/), F20.5)
     IF       (IWILEY(2)) 270,270,280
 270 GO TO    (300,280) , JN
 280 PUNCH 290,     TP, (W(J,4,1), J=1,3)
 290 FORMAT (9(6E12.5,7X1H2/), E12.5,67X1H2)
 300 IF       (IOPT(3)) 310,400,304
 304 IOPT(3) = -1
     WRITE    (6,305)
 305 FORMAT (19X1HT, 18X2HEK, 18X2HEP,17X3HES1,17X3HES2,17X3HES3/
3051 17X3HEF1,17X3HEF2,17X3HEF3, 17X3HEB1,17X3HEB2,17X3HEB3/
3052 16X4HEFF1,16X4HEFF2,16X4HEFF3,16X4HEBF1,16X4HEBF2,16X4HEBF3///)
     LSET     = 1
     GO TO    400
 310 D59      = 0.0
     D69      = 0.0
 318 DO       330 K=1,3
     A1(K)    = 0.0
     DO       320 J=1,3
 320 A1(K)  = A1(K) + U(J+18)*XJ(K,J)
     D59      = D59 + A1(K) * U(K+18)
 330 D69      = D69 + U(K+6)*U(K+6)
     EK       = D59/2.+D69*XIGAM/2.      +XM/2.*(U(13)*U(13)+U(14)*U
    1(14) + U(15) * U(15))
     EP       = -(VZERO + DA(7) * U(18))
     M        = 0
     DO       338 K= 20,24
     DO       338 J = 1,3
     L        = 5*J + K
     M        = M + 1
 338 ODEAR(M) = U(L)
     WRITE    (6,340) TP, EK, EP, (ODEAR(J), J = 1,15)
 340 FORMAT (/3(6F20.5/))
     IF       (IWILEY(3)) 370,370,390
 370 GO TO    (400,390),JN
```

| Card No. | Parameter | Description |
|---|---|---|
| 5 | HBO | thickness of uncrushed block |
| | XIALPH | moment of inertia of footpad about pivot |
| | XIGAM | moment of inertia of lower link about lower hardpoint |
| | HFO | thickness of uncrushed footpad |
| | DB | diameter of block |
| 6 | DF | diameter of footpad |
| | XDOTOC<br>YDOTOC<br>ZDOTOC | velocity of spacecraft CG in ground coordinates (PHASE NO. = 1) or in vehicle coordinates (PHASE NO. = 3). See Card 11 below. |
| | STAB KEY | = 0 stability subroutine has essentially no control of program<br>= 1 stability subroutine has partial control of program |
| 7 | XNU | mechanical friction coefficient for shock absorber |
| | XLDMIN | minimum length permitted for shock absorber link |
| | XKS | spring rate of shock absorber stop |
| 8 | THETD (1)<br>THETD (2)<br>THETD (3) | angular orientation of leg sets No. 1, 2, 3 with respect yz plane (vehicle coordinates). Angles increase in left-handed sense. |
| 9 | THETAS | ground slope |
| | PSIO | pitch angle — relative to ground coordinates |
| | PHIO | yaw angle — relative to ground coordinates |
| | XIO | roll angle — relative to ground coordinates |

| | | | | | | |
|---|---|---|---|---|---|---|
| FOOT PAD NO.1 | | | | | | |
| .000 | .100 | .250 | .500 | .750 | 1.000 | 1.250 |
| 1.500 | 1.750 | 2.000 | 2.490 | 2.666 | 3.500 | 4.500 |
| .000 | .466 | .502 | .563 | .629 | .697 | .767 |
| .841 | .920 | 1.000 | 1.000 | 1.123 | 1.823 | 6.503 |
| FOOT PAD NO.2 | | | | | | |
| .000 | .100 | .250 | .500 | .750 | 1.000 | 1.250 |
| 1.500 | 1.750 | 2.000 | 2.490 | 2.666 | 3.500 | 4.500 |
| .000 | .468 | .502 | .563 | .629 | .697 | .767 |
| .841 | .920 | 1.000 | 1.000 | 1.623 | 1.823 | 6.503 |
| FOOT PAD NO.3 | | | | | | |
| .000 | .100 | .250 | .500 | .750 | 1.000 | 1.250 |
| 1.500 | 1.750 | 2.000 | 2.490 | 2.666 | 3.500 | 4.500 |
| .000 | .466 | .502 | .563 | .629 | .697 | .767 |
| .841 | .920 | 1.000 | 1.000 | 1.623 | 1.823 | 6.503 |
| BLOCK NO.1 | | | | | | |
| .000 | .100 | 5.750 | 7.750 | | | |
| .000 | 1.000 | 1.000 | 21.000 | | | |
| BLOCK NO.2 | | | | | | |
| .000 | .100 | 5.750 | 7.750 | | | |
| .000 | 1.000 | 1.000 | 21.000 | | | |
| BLOCK NO.3 | | | | | | |
| .000 | .100 | 5.750 | 7.750 | | | |
| .000 | 1.000 | 1.000 | 21.000 | | | |
| SHOCK ABSORBER NO.1 | | | | | | |
| .000 | .250 | .500 | .750 | 1.000 | 1.250 | 1.500 |
| 1.750 | 2.000 | 2.250 | 2.500 | 2.750 | 3.000 | 3.250 |
| 3.500 | 3.750 | 4.000 | 4.250 | 4.500 | 6.500 | |
| 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 1.005 | 1.012 | 1.020 | 1.026 | 1.036 | 1.044 | 1.052 |
| 1.062 | 1.076 | 1.096 | 1.114 | 1.151 | 1.447 | |
| 1.000 | 1.060 | 1.160 | 1.240 | 1.360 | 1.480 | 1.600 |
| 1.740 | 1.890 | 2.130 | 2.770 | 3.610 | 4.960 | 6.090 |
| 7.240 | 8.370 | 9.520 | 10.670 | 11.600 | 20.840 | |
| SHOCK ABSORBER NO.2 | | | | | | |
| .000 | .250 | .500 | .750 | 1.000 | 1.250 | 1.500 |
| 1.750 | 2.000 | 2.250 | 2.500 | 2.750 | 3.000 | 3.250 |
| 3.500 | 3.750 | 4.000 | 4.250 | 4.500 | 6.500 | |
| 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 1.005 | 1.012 | 1.020 | 1.026 | 1.036 | 1.044 | 1.052 |
| 1.062 | 1.076 | 1.096 | 1.114 | 1.151 | 1.447 | |
| 1.000 | 1.080 | 1.160 | 1.240 | 1.360 | 1.480 | 1.600 |
| 1.740 | 1.890 | 2.130 | 2.770 | 3.610 | 4.960 | 6.090 |
| 7.240 | 8.370 | 9.520 | 10.670 | 11.600 | 20.840 | |
| SHOCK ABSORBER NO.3 | | | | | | |
| .000 | .250 | .500 | .750 | 1.000 | 1.250 | 1.500 |
| 1.750 | 2.000 | 2.250 | 2.500 | 2.750 | 3.000 | 3.250 |
| 3.500 | 3.750 | 4.000 | 4.250 | 4.500 | 6.500 | |
| 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 1.005 | 1.012 | 1.020 | 1.026 | 1.036 | 1.044 | 1.052 |
| 1.062 | 1.076 | 1.096 | 1.114 | 1.151 | 1.447 | |
| 1.000 | 1.080 | 1.160 | 1.240 | 1.360 | 1.480 | 1.600 |
| 1.740 | 1.890 | 2.130 | 2.770 | 3.610 | 4.960 | 6.090 |
| 7.240 | 8.370 | 9.520 | 10.670 | 11.600 | 20.840 | |

```
NEW DATA CARD FOR THIS CASE IS NO. =  1
NEW DATA CARD FOR THIS CASE IS NO. =  2
NEW DATA CARD FOR THIS CASE IS NO. =  3
NEW DATA CARD FOR THIS CASE IS NO. =  4
NEW DATA CARD FOR THIS CASE IS NO. =  5
NEW DATA CARD FOR THIS CASE IS NO. =  6
NEW DATA CARD FOR THIS CASE IS NO. =  7
NEW DATA CARD FOR THIS CASE IS NO. =  8
NEW DATA CARD FOR THIS CASE IS NO. =  9
NEW DATA CARD FOR THIS CASE IS NO. = 10
NEW DATA CARD FOR THIS CASE IS NO. = 11
NEW DATA CARD FOR THIS CASE IS NO. = 12
NEW DATA CARD FOR THIS CASE IS NO. = 13
NEW DATA CARD FOR THIS CASE IS NO. = 14
NEW DATA CARD FOR THIS CASE IS NO. = 15
NEW DATA CARD FOR THIS CASE IS NO. = 16
NEW DATA CARD FOR THIS CASE IS NO. = 17
NEW DATA CARD FOR THIS CASE IS NO. = 18
NEW DATA CARD FOR THIS CASE IS NO. = 19
NEW DATA CARD FOR THIS CASE IS NO. = 20
NEW DATA CARD FOR THIS CASE IS NO. = 21

THIS CASE IS SERIES NO. =      1.000    AND PHASE NO. =   1

INPUT DATA FOLLOWS.

CARD  1       XM  =  .20000+02     IXX  =  .14222+03     IYY  =  .14222+03     IZZ  =  .14222+03     IXY  =  .00000
CARD  2      IXZ  =  .00000        IYZ  =  .00000        XMP  =  .32400-01    XLLO  =  .31275+01   DELCG  =  .15417+01
CARD  3      XLP  =  .31875+01     XLL  =  .33750+01     XLA  =  .95833-00       G  =  .53100+01 ALFHA(1)=  .75000-00
CARD  4 ALPHA(2)=  .45000+02 ALPHK(1)=  .14300+05 ALPHK(2)=  .14300+05    BETA  =  .41000+02     XLC  =  .24167+01
CARD  5      HEO  =  .66667-00  XIALFH  =  .37300-02   XIGAM  =  .13470+01     HFO  =  .54167-00      UE  =  .58333-00
CARD  6       DF  =  .10000+01  XDOTOC  =  .00000     YDOTOC  = -.92769+01  ZDOTOC  =  .14246+02    STAB KEY = .10000 + 01
CARD  7      XNU  =  .50000-01  XLDMIN  =  .27525+01     XKS  =  .10000+06
CARD  8 THETD(1)=  .00000   THETD(2)=  .12000+03 THETD(3)=  .24000+03
CARD  9   THETAS  =  .15000+02    PSIO  =  .50000+01    PHIO  =  .00000        XIO  =  .00000     XLAMDD  =  .00000
CARD 10 OMEGA(1)=  .00000   OMEGA(2)=  .00000   OMEGA(3)=  .00000      CODE2  =  .40000+01
CARD 11     CODE  =  6.00000   SER NO  =  1.00000 PHASE NO=       1         VH  = -.00000        VV  = -.00000
CARD 12      LTF  =  .25000+01      TF  =  .60000+01      HM  =  .10000-05  ERR(4)  =  .60000+01  ERR(5)  =  .60000+01
CARD 13   ERR(6)  =  .60000+01  ERR(7)  =  .20000+01  ERR(8)  =  .20000+01  ERR(9)  =  .20000+01 ERR(10)  =  .40000+01
CARD 14  ERR(11)  =  .40000+01 ERR(12)  =  .40000+01 ERR(13)  =  .16667-00 ERR(14)  =  .16667-00 ERR(15)  =  .16667-00
CARD 15  ERR(16)  =  .16667-00 ERR(17)  =  .16667-00 ERR(18)  =  .16667-00 ERR(19)  =  .16667-00 ERR(20)  =  .16667-00
```

| | | | | | |
|---|---|---|---|---|---|
| 2.82069 | 6.01231 | .30220 | .57477 | 85.49601 | 1.96080 |
| -.00003 | -5.61468 | 2.66708 | -.00207 | .00465 | .73890 |
| -33.75453 | -.00319 | -.00122 | 6.01232 | .30207 | .57510 |
| 18.00984 | 17.91008 | 17.90507 | .05730 | -5.76435 | -1.58231 |
| -.00004 | 1.92276 | -3.54207 | 5.54141 | -7.16612 | -.46158 |
| -5.54197 | -7.16623 | -.46250 | | | |

| | | | | | |
|---|---|---|---|---|---|
| 2.82069 | .00000 | 152.67363 | 157.22436 | .00000 | .00000 |
| .00000 | -109.23920 | 63.06927 | -86.01339 | 112.50230 | 64.95325 |
| -86.56529 | .00000 | .00000 | .00000 | 85.72577 | 76.69100 |
| 11.60074 | -115.64415 | 31.96154 | 27.19091 | -.00000 | -.00000 |
| -.00000 | -98.22428 | 56.70981 | 350.94483 | -76.64657 | -44.36740 |
| -274.64756 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | .00000 | .00000 | .00000 | .00000 | .00000 |
| .00000 | -20.98732 | 138.30182 | -83.43817 | -1.97333 | 97.60944 |
| -57.19812 | .01204 | .02424 | .10692 | .08645 | .10692 |
| .08645 | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 2.82069 | 6.25950 | -304.54803 | 225.40639 | 662.21365 | 662.21692 |
| 6.13150 | 54.69632 | 54.58573 | -.00000 | 168.54622 | 168.54624 |
| 168.02017 | 419.22076 | 419.11573 | -.00000 | 89.75712 | 89.75728 |