60 Concord Street
Wilmington, Massachusetts

CR-119922

R-120U

LIFTING-ROBOT-SPACE SHUTTLE PRELIMINARY MISSION PROFILE OPTIMIZATION PROGRAM

PART I MATHEMATICAL FORMULATION

May 29, 1971

Prepared for

Marshall Space Flight Center
National Aeronautics and Space Administration
Huntsville, Alabama

Generated under
Contract NAS8-25713

Prepared by

Robert G. Gottlieb
Project Engineer

Approved by

Allan Dushman
Manager, Systems Analysis
and Design Department

DYNAMICS RESEARCH CORPORATION
Systems Division
60 Concord Street
Wilmington, Massachusetts 01887

FOREWORD

This report presents the results of development and implementation of steepest-ascent optimization theory as applied to trajectory computation. This program depicts the results of the developments performed by efforts contracted by the Trajectory and Optimization Theory Branch of the Astrodynamics and Guidance Theory Division of the Aero-Astrodynamics Laboratory at MSFC. Questions and requests pertaining to this program should be addressed to Mr. Ron Toelle at the Trajectory and Optimization Theory Branch, Aero-Astrodynamics Laboratory, MSFC.

ABSTRACT

LIFTING ROBOT is a minimum Hamiltonian-steepest ascent multistage lifting booster optimization program. It can simulate up to 15 thrust or coast events, provides aerodynamic lift and drag as a function of Mach number and angle of attack, and can maintain "g" limits by throttling.

The payoff and terminal constraints can be selected from a library of 26 functions. In addition, intermediate point constraints, selected from the same library, may be imposed on the trajectory following any one of the thrust events.

Through the use of input switches, a variety of vehicle parameters can be optimized in conjunction with the control variables χ-pitch and χ-yaw. Tank limits of stages being optimized can be held and performance reserves as a function of ΔV, can be calculated. The impact point of any stage can be calculated and publishable tables can be printed. The working coordinate system and the environmental simulation conform to Apollo standards.

This document contains the LIFTING ROBOT input description and an example problem.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

TABLE OF CONTENTS (CONT'D)

SYMBOL INDEX

This index gives an alphabetical list of symbols and page no(s) where definition(s) given. More than one page number implies a multiple use of a symbol.

SYMBOL INDEX (CONT'D)

SYMBOL INDEX (CONT'D)

SYMBOL INDEX (CONT'D)

Coordinates

1. INTRODUCTION

The program described in this report is designed to optimize a large variety of multistage lifting booster trajectories. This objective is achieved through the use of the Min-H* steepest-ascent trajectory optimization technique described in Reference 1. Briefly, the steepest-ascent technique requires that a reasonable, but nevertheless arbitrary choice of the controls be used to calculate a nominal trajectory. In general, neither the desired terminal state will result, nor will the performance index be optimum. Next, by solving the adjoint differential equations associated with the linearized perturbation equations about the nominal trajectory, impulse response functions may be determined for arbitrary small variations in the control variables, and influence coefficients may be determined for arbitrary small variations in the control parameters. The choice of small changes in these controls, which simultaneously moves the terminal state closer to the desired terminal state and improves the performance index, is calculated. This change in the controls is added to the nominal control history and the process is repeated until the optimum is reached.

The LIFTING ROBOT program can simulate a multistage lifting booster having up to 15 thrust events. The program can be used for both ground-launch and jump-start trajectories. Both the aerodynamics and throttling of the Space Shuttle are simulated. The working coordinate system and the environmental simulation conform to Apollo standards.

---

*Minimum Hamiltonian

The payoff and terminal constraints can be selected from a library of twenty-six functions. In addition intermediate point constraints, selected from the same library, may be imposed on the trajectory following any one of the thrust events. Also, the instantaneous impact point of the first stage may be forced to avoid specified regions.

Through the use of input switches, a variety of vehicle parameters may be optimized in conjunction with the control variables χ-pitch and χ-yaw. Tank limits of stages being optimized can be held and performance reserves as a function of ΔV, can be calculated. Staging on either fuel or time can be specified. Also, the impact point of any stage can be calculated and publishable tables can be printed.

The LIFTING ROBOT program has straightforward automatic convergence logic and a dynamic updating scheme for the control parameter weighting matrix. Convergence should be as reliable and sure as the original ROBOT program [9].

For the most part, this report is devoted to a description of the mathematical model used in formulating LIFTING ROBOT and to such a limited discussion of the logic structure as affords a complete description of program flexibility.

2. COORDINATE SYSTEMS

The basic reference coordinate system in the LIFTING ROBOT program is the inertial geocentric cartesian coordinate system $\hat{X}\,\hat{Y}\,\hat{Z}$ shown in Fig. 1. This coordinate system has the $\hat{Y}$ axis pointing north, the $\hat{X}$ and $\hat{Z}$ axes in the equatorial plane, and the $\hat{Z}$ axis in the meridian plane that contains the launch site at gyro release time. In the LIFTING ROBOT program gyro release time or guidance reference release (GRR) is a reference time occurring either prior to or at liftoff.

Next described is the inertial cartesian plumbline coordinate system $\hat{x}\,\hat{y}\,\hat{z}$, in which the equations of motion are written.

The plumbline coordinate system $\hat{x}\,\hat{y}\,\hat{z}$, shown in Fig. 2 is formed from $\hat{X}\,\hat{Y}\,\hat{Z}$ by first rotating counterclockwise about $\hat{X}$ through θ_1 and then clockwise about $\hat{y}$ through A_z - 90. A_z is the launch azimuth angle and $\theta_1 = \pi/2 - \theta_o$ where θ_o is the geodetic latitude of the launch site. Both A_z and θ_o are input quantities.

The equations for transforming a vector from the $\hat{X}\,\hat{Y}\,\hat{Z}$ system to the $\hat{x}\,\hat{y}\,\hat{z}$ system are

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = A \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix}$$

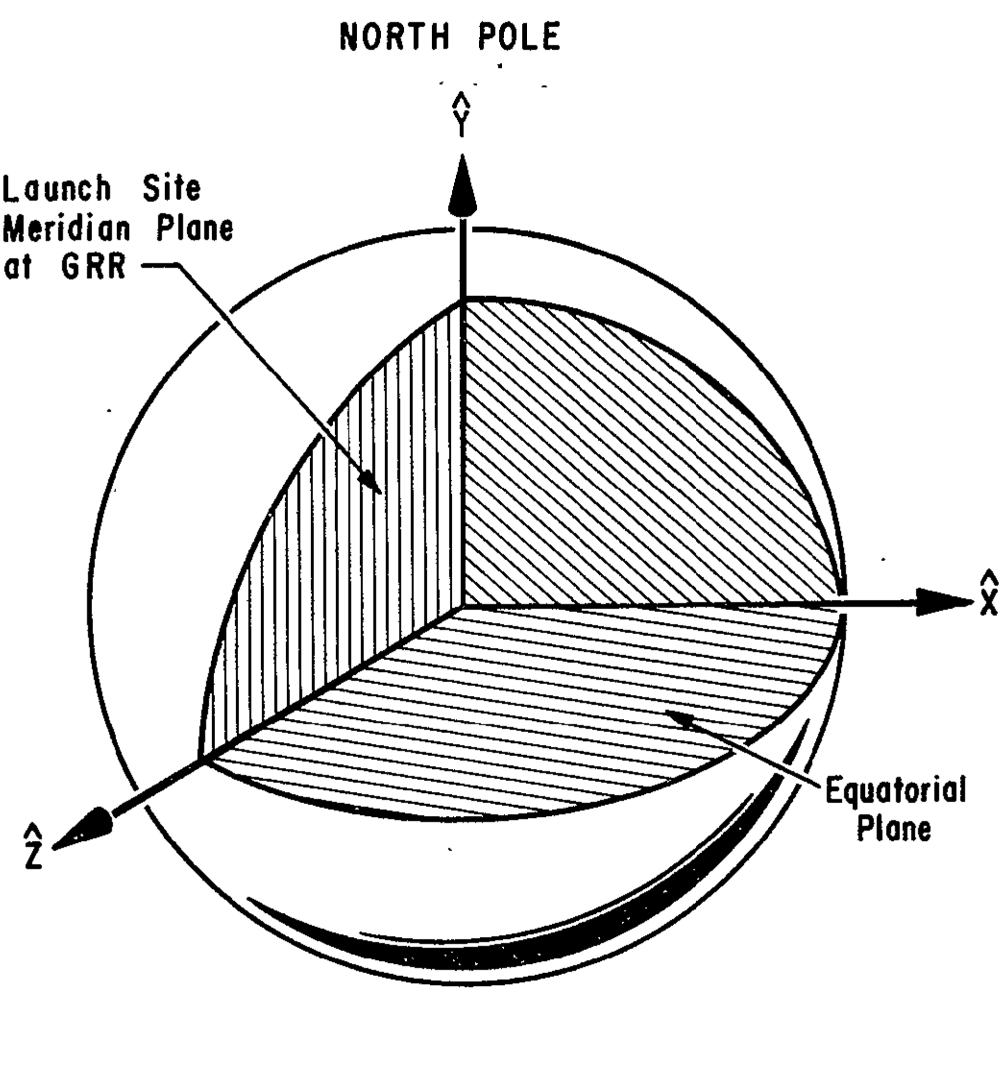

FIG. 1. REFERENCE COORDINATE SYSTEM $\hat{X}\ \hat{Y}\ \hat{Z}$

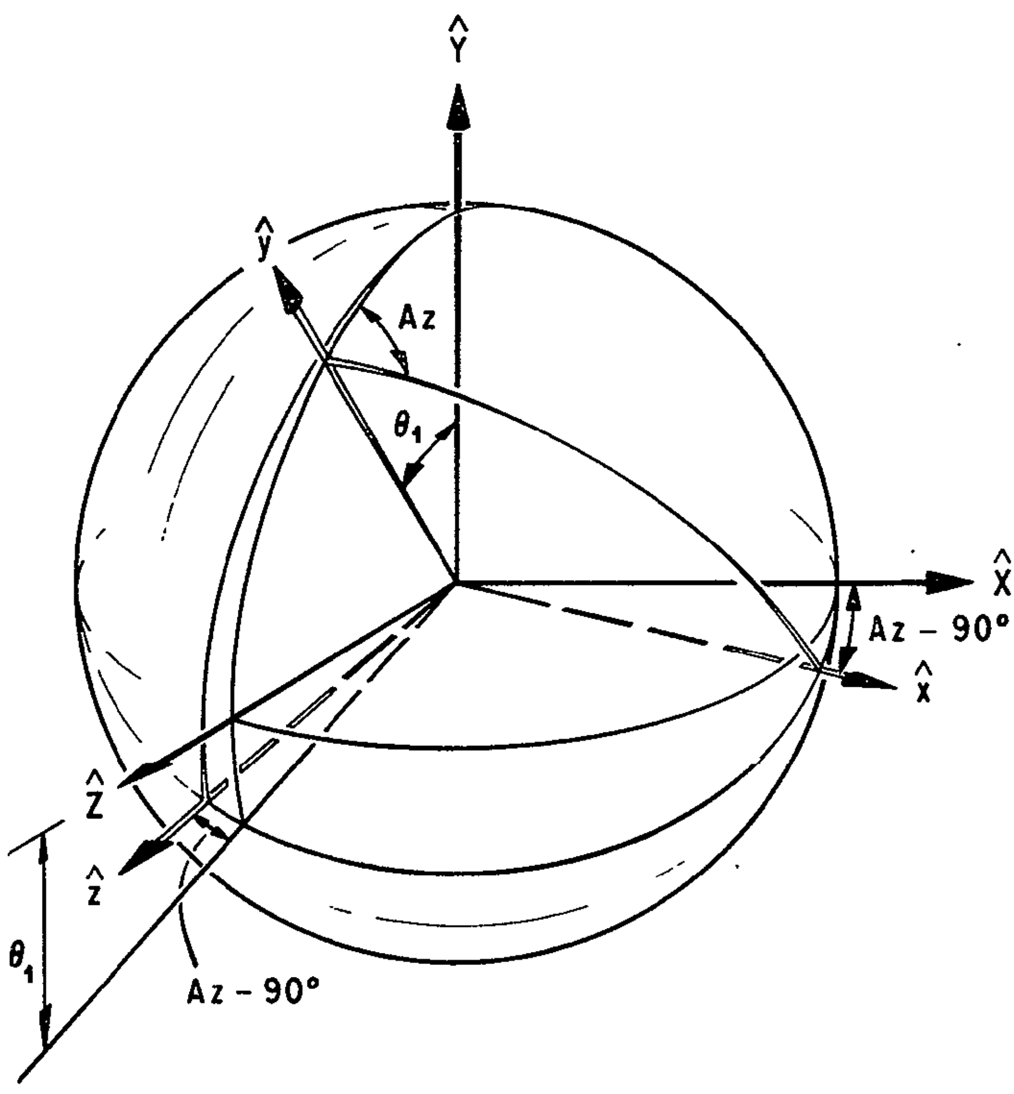

FIG. 2. PLUMBLINE COORDINATE SYSTEM $\hat{x}\ \hat{y}\ \hat{z}$

where

$$A = \begin{bmatrix} \sin A_z & \cos A_z \sin\theta_1 & -\cos A_z \cos\theta_1 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ \cos A_z & -\sin A_z \sin\theta_1 & \sin A_z \cos\theta_1 \end{bmatrix}$$

Since A_z is an input constant and θ_1 is the complement of an input constant, the matrix A is also constant.

In the plumbline system the position coordinates x, y, z and the velocity components w, u, v are measured in the $\hat{x}$, $\hat{y}$, $\hat{z}$ directions, respectively.

The plumbline system in LIFTING ROBOT differs from the Apollo 13 coordinate system [3] only in the names of the axes, i. e.,

$$\begin{bmatrix} \hat{z} \\ \hat{x} \\ \hat{y} \end{bmatrix}_{\text{Apollo 13}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}_{\text{ROBOT}}$$

The third coordinate system used in LIFTING ROBOT is the geocentric spherical polar coordinate system $\hat{\phi}\ \hat{r}\ \hat{\theta}$ with coordinates ϕ, r, and θ. The $\hat{\phi}\ \hat{r}\ \hat{\theta}$ axes, shown in Fig. 3, point in the direction of increasing ϕ, r, and θ, respectively, and may be formed by first rotating counterclockwise about $\hat{Y}$ through ϕ and then rotating counterclockwise about $\hat{\phi}$ through θ.

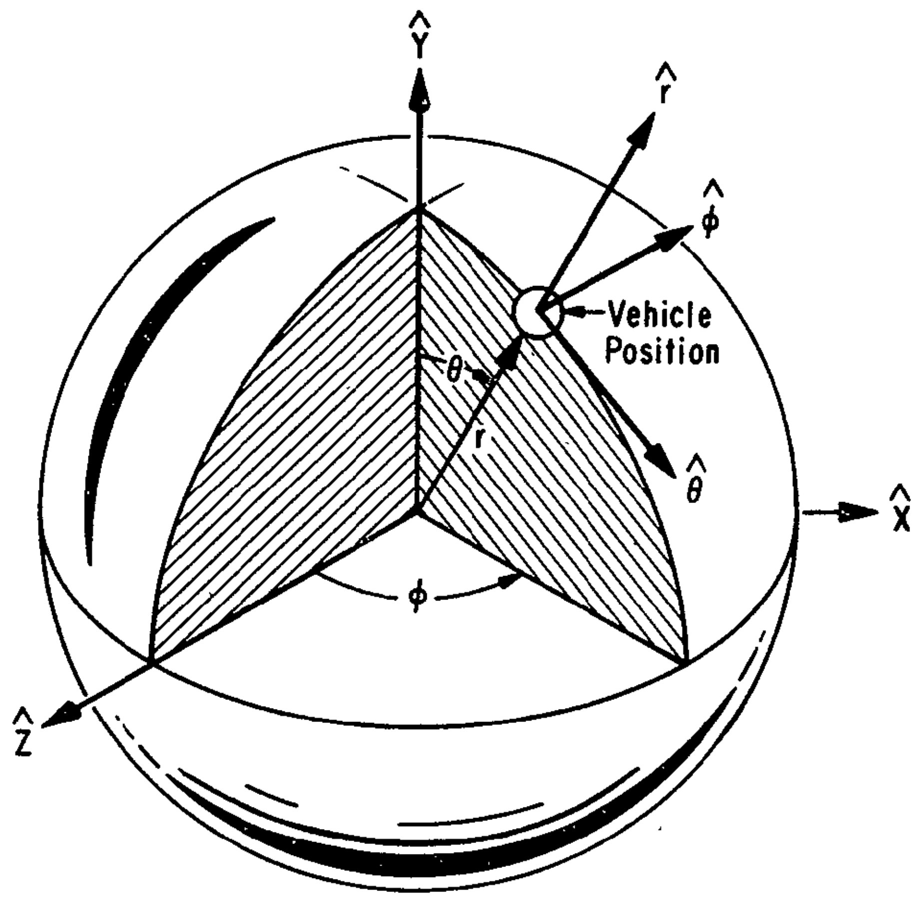

FIG. 3. GEOCENTRIC SPHERICAL COORDINATE SYSTEM $\hat{\phi}\,\hat{r}\,\hat{\theta}$

The projections of r on $\hat{X}\,\hat{Y}\,\hat{Z}$ are

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = r \begin{bmatrix} \sin\theta\sin\phi \\ \cos\theta \\ \sin\theta\cos\phi \end{bmatrix}$$

and therefore the projections of r on $\hat{x}\,\hat{y}\,\hat{z}$ are

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = r\ A \begin{bmatrix} \sin\theta\sin\phi \\ \cos\theta \\ \sin\theta\cos\phi \end{bmatrix}$$

The transformation from x, y, z to ϕ, r, θ is therefore

$$\phi = \tan^{-1}\left(\frac{a_{11}x + a_{21}y + a_{31}z}{a_{13}x + a_{23}y + a_{33}z}\right)$$

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \cos^{-1}((a_{12}x + a_{22}y + a_{32}z)/r)$$

where the a_{ij} are elements of the A matrix described previously.

The equations for transforming a vector from the $\hat{\phi}\ \hat{r}\ \hat{\theta}$ system to the $\hat{X}\,\hat{Y}\,\hat{Z}$ system are

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} = B \begin{bmatrix} \hat{\phi} \\ \hat{r} \\ \hat{\theta} \end{bmatrix}$$

where

$$B = \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & \sin\phi\cos\theta \\ 0 & \cos\theta & -\sin\theta \\ -\sin\phi & \cos\phi\sin\theta & \cos\phi\cos\theta \end{bmatrix}$$

and therefore the equations for transforming a vector from the $\hat{\phi}\,\hat{r}\,\hat{\theta}$ system to the $\hat{x}\hat{y}\hat{z}$ system may be written

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = D \begin{bmatrix} \hat{\phi} \\ \hat{r} \\ \hat{\theta} \end{bmatrix}$$

where $D = A \cdot B$

Also, the inertial velocity components in the $\hat{\phi}$, $\hat{r}$, $\hat{\theta}$ directions, w_s, u_s, v_s respectively, may be written

$$\begin{bmatrix} w_s \\ u_s \\ v_s \end{bmatrix} = D^T \begin{bmatrix} w \\ u \\ v \end{bmatrix}^*$$

If the multiplication $A \cdot B$ is performed and substitutions for ϕ and θ are made in terms of x, y and z, the elements of the D matrix become

$$d_{11} = (a_{22}z - a_{32}y)/r \sin\theta$$

$$d_{21} = (a_{32}x - a_{12}z)/r \sin\theta$$

$$d_{31} = (a_{12}y - a_{22}x)/r \sin\theta$$

$^*( )^T$ Denotes matrix transpose.

$$d_{12} = x/r$$

$$d_{22} = y/r$$

$$d_{32} = z/r$$

$$d_{13} = (d_{12} \cos\theta - a_{12})/\sin\theta$$

$$d_{23} = (d_{22} \cos\theta - a_{22})/\sin\theta$$

$$d_{33} = (d_{32} \cos\theta - a_{32})/\sin\theta$$

3. GEOPHYSICAL PROPERTIES

Described in this section are three geophysical properties of the earth which affect a rocket trajectory: gravitational acceleration, geometric form, atmospheric properties.

3.1 GRAVITATIONAL ACCELERATIONS

The gravitational potential function, $U(r, \theta)$, used in LIFTING ROBOT is [4]

$$U(r, \theta) = \frac{\mu_e}{r}\left[1 + \frac{CJ}{3}\left(\frac{R_e}{r}\right)^2 (1 - 3\cos^2\theta) + \frac{H}{5}\left(\frac{R_e}{r}\right)^3 (3 - 5\cos^2\theta)\cos\theta + \frac{DJ}{35}\left(\frac{R_e}{r}\right)^4 (3 - 30\cos^2\theta + 35\cos^4\theta)\right]$$

where CJ, H, DJ, R_e, μ_e are input parameters which are, however, preset to

$$\begin{aligned}
CJ &= 1.62345 \times 10^{-3} \\
H &= -0.575 \times 10^{-5} \\
DJ &= 0.7875 \times 10^{-5} \\
R_e &= \text{Earth equatorial radius} \\
&= 6378165.\ \text{m} \\
\mu_e &= \text{Product of universal gravity constant and earth mass} \\
&= 3.986032 \times 10^{14}\ \text{m}^3/\text{sec}^2
\end{aligned}$$

The components of the gravitational acceleration vector in the plumbline system are calculated as the first partial derivatives of $U(r, \theta)$ with respect to the plumbline position coordinates, i.e., g_x, g_y and g_z are calculated as

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{\partial U}{\partial z} \end{bmatrix} = \frac{\partial U}{\partial r} \begin{bmatrix} \frac{\partial r}{\partial x} \\ \frac{\partial r}{\partial y} \\ \frac{\partial r}{\partial z} \end{bmatrix}^{*} + \frac{\partial U}{\partial \theta} \begin{bmatrix} \frac{\partial \theta}{\partial x} \\ \frac{\partial \theta}{\partial y} \\ \frac{\partial \theta}{\partial z} \end{bmatrix}^{*}$$

These equations may be rearranged into the form

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = G_{11} \begin{bmatrix} x \\ y \\ z \end{bmatrix} - G_{TO} \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}$$

where

$$G_{11} = -\frac{\mu_e}{r^3} \left[1 + CJ\left(\frac{R_e}{r}\right)^2 (1 - 5\cos^2\theta) + H\left(\frac{R_e}{r}\right)^3 (3 - 7\cos^2\theta)\cos\theta \right.$$

$$\left. + DJ\left(\frac{R_e}{r}\right)^4 \left(\frac{3}{7} - (6 - 9\cos^2\theta)\cos^2\theta\right) \right]$$

---

*These partials are given in Appendix B.

$$G_{TO} = \frac{\mu_e}{r^2} \left[2CJ \left(\frac{R_e}{r}\right)^2 \cos\theta - H \left(\frac{R_e}{r}\right)^3 \left(\frac{3}{5} - 3\cos^2\theta\right) + DJ \left(\frac{R_e}{r}\right)^4 \left(\frac{12}{7} - 4\cos^2\theta\right) \cos\theta \right]$$

Equations of the same general form are used in the Saturn V flight computer. [5]

In the event that a spherical earth is to be simulated these equations become

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = G_{11} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where

$$G_{11} = -\mu_e / r^3$$

and, of course,

$$G_{TO} = 0$$

3.2 GEOMETRIC FORM

The earth is taken to be an ellipsoid, [6] as shown in Fig. 4., which rotates about its polar axis with an angular velocity Ω_e.

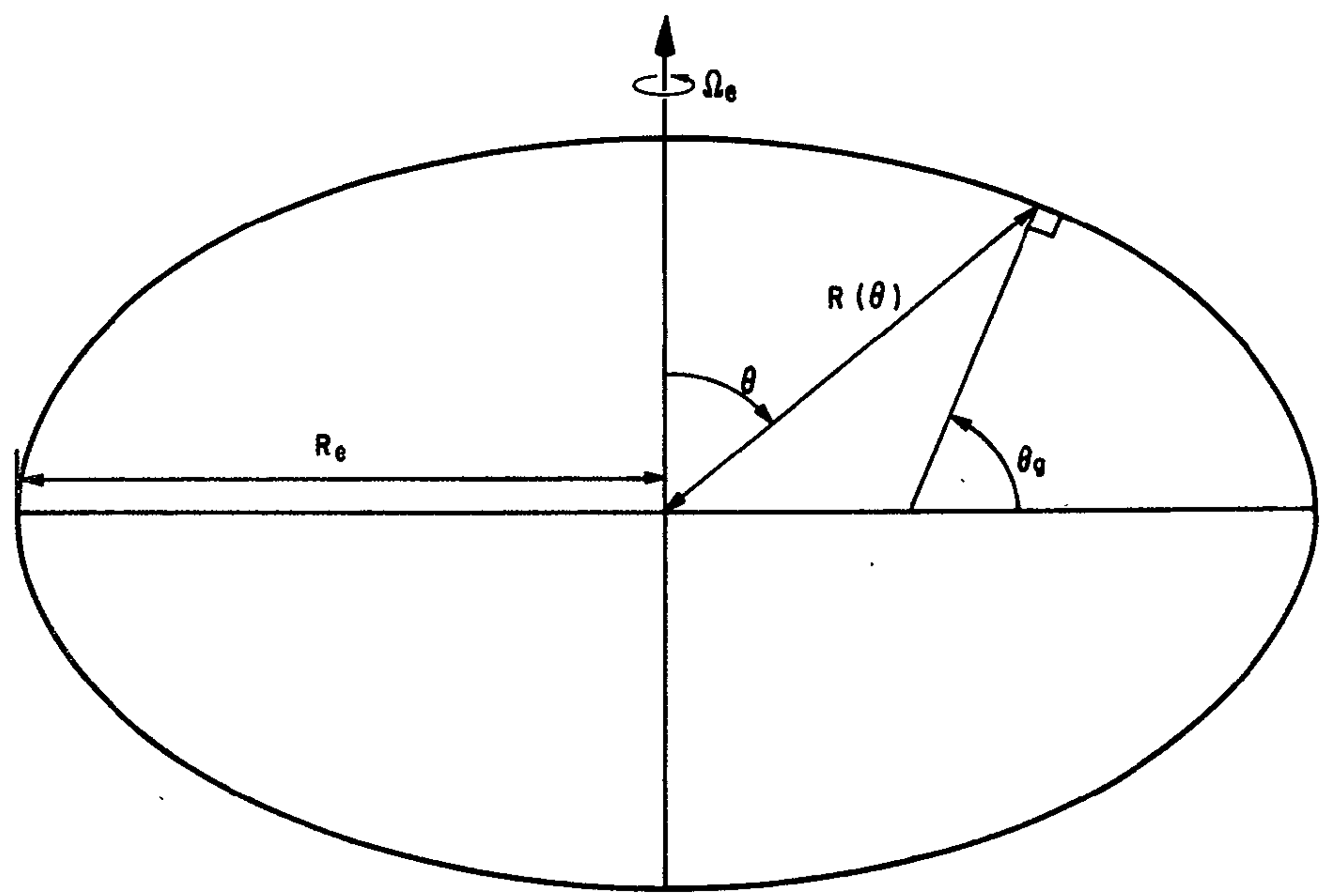

FIG. 4. THE EARTH — AN ELLIPSOID

The angular velocity, Ω_e, and the flattening, f, are input constants that are preset to

$$\Omega_e = 7.2921158 \times 10^{-5} \text{ rad/sec}$$

$$f = 1/298.3$$

The relationship between geocentric colatitude, θ, and geodetic latitude, θ_g, is expressed by

$$\text{ctn}\,\theta = (1-f)^2 \tan\theta_g$$

The radius of the earth as a function of colatitude, $R(\theta)$, is

$$R(\theta) = (1-f)\,R_e / \sqrt{(1-f)^2 \sin^2\theta + \cos^2\theta}$$

The derivative of $R(\theta)$ with respect to θ, which is needed in order to calculate the time at which maximum dynamic pressure occurs, and altitude related terms in the adjoint equations is given by

$$\frac{dR(\theta)}{d\theta} = \frac{R(\theta)^3\, f(2-f)\sin\theta\cos\theta}{(R_e)^2(1-f)^2}$$

3.3 ATMOSPHERIC PROPERTIES

The earth is assumed to have an atmosphere which rotates with it at the same angular velocity, so that there is no wind over the earth's rotating surface.

The PRA63 model atmosphere [7] routine on the MSFC system tape is presently used to calculate density, ρ, pressure, p_a and speed of sound, s, as a function of the altitude, h, where h is calculated from

$$h = r - R(\theta)$$

During the adjoint integration, analytic derivatives of ρ, p_a, and s; $\frac{1}{\rho}\frac{d\rho}{dh}$, $\frac{dp}{dh}$ and $\frac{1}{s}\frac{ds}{dh}$, respectively are calculated in PRB63 which differs from PRA63 only in that these derivatives are calculated.

4. CONTROL VARIABLES

The time history of the orientation in space of the centerline, $\hat{c}$, of the boost vehicle is determined by the control variable attitude angles χ_p (chi-pitch) and χ_y (chi-yaw) shown in Fig. 5.

In addition to defining the position of the centerline, $\hat{c}$, χ_p and χ_y may be thought of as defining the auxiliary coordinate axes $\hat{p}\,\hat{c}\,\hat{\phi}$ shown in Fig. 5. This auxiliary coordinate system is formed by rotating clockwise about $\hat{z}$ through χ_p and then counterclockwise about $\hat{p}$ through χ_y.

The equations for transforming a vector from the $\hat{p}\,\hat{c}\,\hat{\phi}$ system into the $\hat{x}\,\hat{y}\,\hat{z}$ system are

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = C \begin{bmatrix} \hat{p} \\ \hat{c} \\ \hat{\phi} \end{bmatrix}$$

where

$$C = \begin{bmatrix} \cos\chi_p & \sin\chi_p \cos\chi_y & -\sin\chi_p \sin\chi_y \\ -\sin\chi_p & \cos\chi_p \cos\chi_y & -\cos\chi_p \sin\chi_y \\ 0 & \sin\chi_y & \cos\chi_y \end{bmatrix}$$

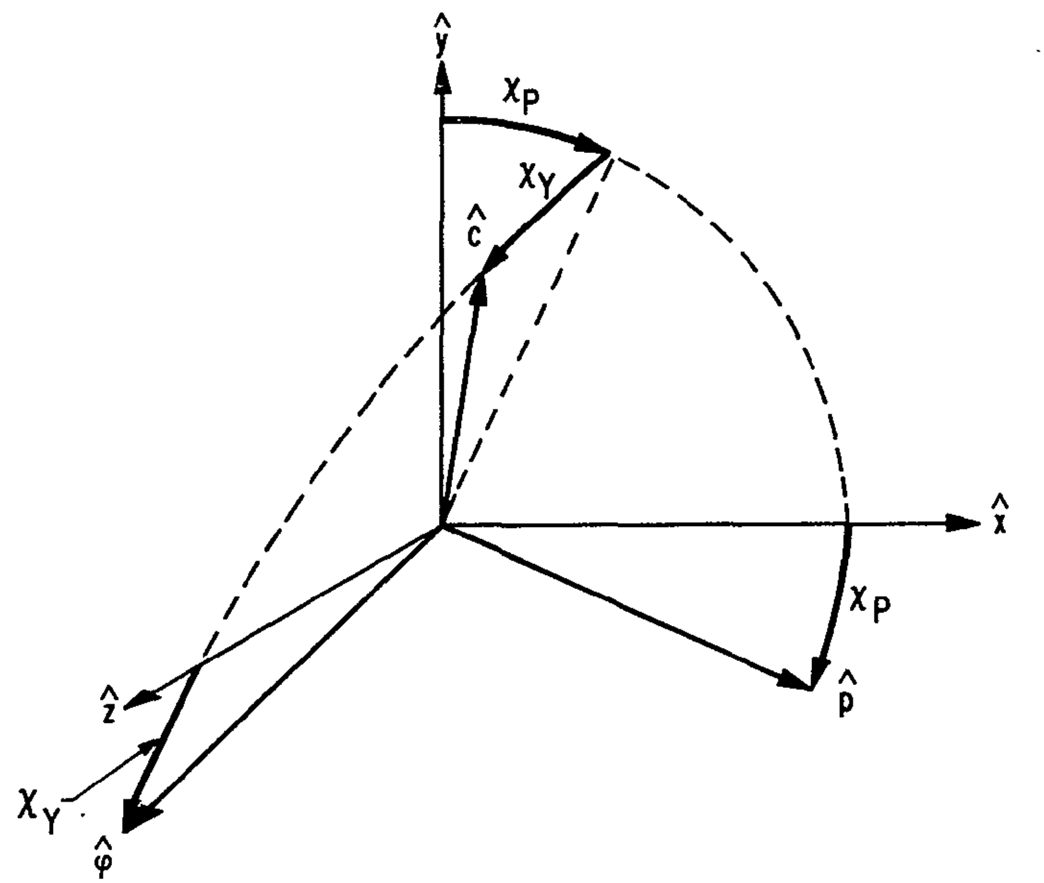

FIG. 5. χ_P, χ_Y AND $\hat{p}\hat{c}\hat{\varphi}$ COORDINATE SYSTEM

The unit vector $\hat{c}$ which defines the centerline of the vehicle in the plumbline system is

$$\hat{c} = \begin{bmatrix} \sin\chi_p \quad \cos\chi_y \\ \cos\chi_p \quad \cos\chi_y \\ \sin\chi_y \end{bmatrix}$$

5. AERODYNAMIC FORCES

The passage of the vehicle through the atmosphere gives rise to aerodynamic forces defined to act coincident with and normal to the vehicle body axis $\hat{c}$. In LIFTING ROBOT, the relative velocity, $\overline{V}_R$, is considered to be the velocity of the vehicle relative to the atmosphere.

$$\overline{V}_R = \begin{pmatrix} \underline{w} \\ \underline{u} \\ \underline{v} \end{pmatrix} = \begin{pmatrix} w \\ u \\ v \end{pmatrix} - D \begin{pmatrix} r\Omega_e \sin\theta \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} w - (a_{22} z - a_{32} y)\, \Omega_e \\ u - (a_{32} x - a_{12} z)\, \Omega_e \\ v - (a_{12} y - a_{22} x)\, \Omega_e \end{bmatrix}$$

It is convenient to define the magnitude of the relative velocity, V_R

$$V_R \equiv |\overline{V}_R| = \sqrt{\underline{w}^2 + \underline{u}^2 + \underline{v}^2}$$

and a unit vector, $\hat{V}_R$, in the direction of $\overline{V}_R$ as

$$\hat{V}_R \equiv \overline{V}_R / V_R$$

The aerodynamic normal force in LIFTING ROBOT is assumed to lie in the plane defined by $\hat{c}$ and $\hat{V}_R$. This allows the aerodynamic forces to be described in the plumbline system using the nonorthogonal coordinates $\hat{c}$ and $\hat{V}_R$ and the total angle of attack, α. This situation is depicted in Fig. 6.

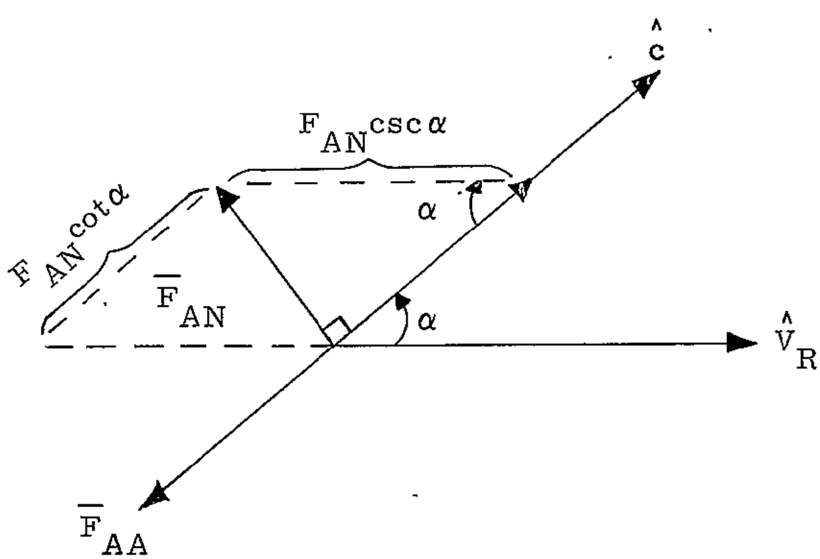

FIGURE 6 AERODYNAMIC FORCE QUANTITIES

It is obvious from the figure that the vector of force due to F_{AA} is

$$\overline{F}_{AA} = -F_{AA}\,\hat{c}$$

and that the vector of force due to F_{AN} can be found by summing components along $\hat{c}$ and $\hat{V}_R$, i.e.,

$$\overline{F}_{AN} = F_{AN}(\cot\alpha\,\hat{c} - \csc\alpha\,\hat{V}_R)$$

Also, α is simply

$$\alpha = \cos^{-1}(\hat{c}\cdot\hat{V}_R)$$

The dynamic pressure, q, is calculated as

$$q = \frac{1}{2}\rho V_R^2$$

The Mach number, M, is calculated as

$$M = V_R/s$$

These can be used to calculate F_{AA} and F_{AN} as

$$F_{AA} = q\,S\,C_A(M,\alpha)$$

$$F_{AN} = q\,S\,C_N(M,\alpha)$$

where S is the reference area and $C_A(M, \alpha)$ and $C_N(M, \alpha)$ are assumed to be

$$C_A = a(M) \cos\alpha + b(M) \sin^2\alpha + c(M)$$

$$C_N = a(M) \sin\alpha - b(M) \sin\alpha \cos\alpha$$

This form for C_A and C_N is consistent with reasonable assumptions for lift and drag as is shown in Appendix D. The coefficients a, b and c can be fit directly to aerodynamic data for C_A and C_N or constructed from lift and drag data.

Tables of a(M), b(M) and c(M) are provided in subroutine CACN in LIFTING ROBOT. These may be changed by the user via a new data statement.

As is shown below, the form of C_A and C_N is such that F_{AA} and F_{AN} do not have to be constructed in order to write the aerodynamic force vector in the plumbline system.

Summing $\overline{F}_{AA}$ and $\overline{F}_{AN}$, the aerodynamic force vector, $\overline{F}_A$, is found to be

$$\overline{F}_A = (F_{AN} \cot\alpha - F_{AA}) \hat{c} - F_{AN} \csc\alpha \hat{V}_R$$

Using the definition of C_A and C_N above gives

$$\overline{F}_A = \begin{bmatrix} F_{Ax} \\ F_{Ay} \\ F_{Az} \end{bmatrix} = -qS(b+c)\,\hat{c} + qS(b\cos\alpha - a)\hat{V}_R$$

6. BOOSTER CONFIGURATION

In LIFTING ROBOT, the simulation of the thrust profile of a multistage booster is accomplished by synthesizing the profile from a sequence of up to 15 thrust events. By allowing the grouping of these thrust events into stages to be specified by input rather than by fixed internal logic, a great deal of generality is obtained. Fig. 7 depicts five thrust events as an example of such a sequence.

The ith thrust event is characterized by five (seven in the atmosphere) items:

1)	F_i	Reference thrust per engine
2)	m_i	Flow rate per engine
3)	ν_{i1}	Number of inboard engines
	ν_{i2}	Cant angle of inboard engines
	ν_{i3}	Number of outboard engines
	ν_{i4}	Cant angle of outboard engines
4)	τ_i	Thrust event duration
5)	W_i^J	Weight jettisoned at the end of each thrust event

and in the atmosphere ....

6)	Ae_i	Engine exit area
7)	S_i	Aerodynamic reference area

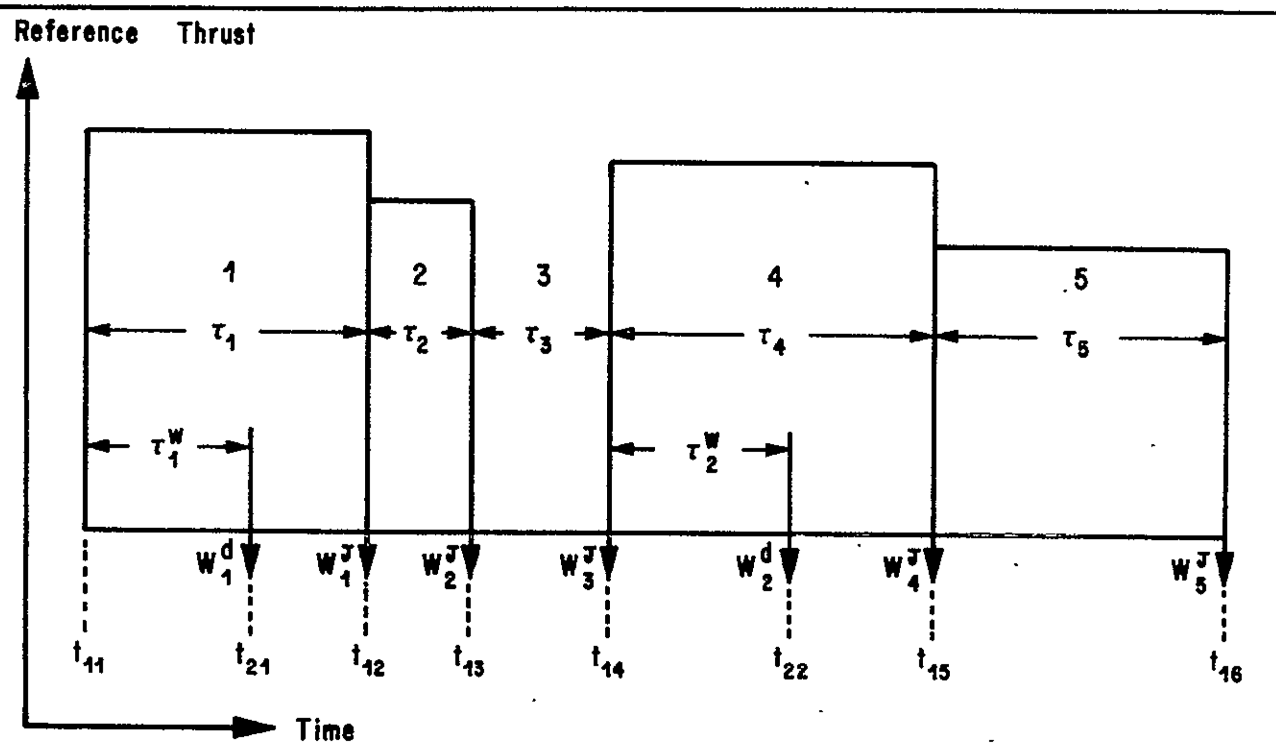

FIG. 7. THRUST PROFILE - A SEQUENCE OF THRUST EVENTS

The convention used in LIFTING ROBOT for labeling thrust event and miscellaneous weight drop event times is also depicted in Fig. 7. Thust event times are labeled t_{1i} and miscellaneous weight drop event times are labeled t_{2i}.

Note that there are six t_{1i} but only five thrust events of duration τ_i. The same is true of a picket fence, in that there is always one more picket than there are spaces. The τ_i may therefore be thought of as "spaces", and the i subscript of t_{1i} as the "picket" number, with i = 1 at the beginning of the first thrust event.

From the figure it is apparent that the t_{1i} are calculated as

$$t_{1i} + 1 = t_{1i} + \tau_i$$

with t_{1i} being defined as some input initial time.

In addition to thrust event items, Fig. 7 also depicts two <u>miscellaneous</u> weight drops. A miscellaneous drop weight, as distinguished from a jettison weight, can be dropped at <u>any</u> time. The ith miscellaneous weight drop is characterized by three items:

1) W_i^d Miscellaneous weight dropped
2) τ_i^w Time interval between beginning of n_i^w th thrust event and miscellaneous weight drop.
3) n_i^w Weight drop time is calculated from the beginning of this thrust event. Can also be

thought of as "picket" number of the thrust event time to which τ_i^w is added to get miscellaneous weight drop time.

The ith miscellaneous weight drop occurs at t_{2i}. The t_{2i} are calculated as

$$t_{2i} = t_{1j} + \tau_i^w$$

where

$$j = n_i^w .$$

Note that with this definition, none, one or many miscellaneous weight drop events may be defined relative to any given thrust event, and may occur during that or any other thrust event. The only restriction being that t_{2i+1} must be greater than t_{2i}.

In LIFTING ROBOT the thrust events are grouped into stages through the use of the input array NØVENT. The first member of the NØVENT array should contain the number of thrust events in the atmosphere, the second member contains the number in the second stage, etc. However they are grouped, all thrust events must be accounted for!

6.1 THRUST AND FLOW RATE

The use of the four numbers ν_{i1}, ν_{i2}, ν_{i3}, ν_{i4}, to describe the effective number of engines leads to a rather cumbersome notation if they are used in each equation where the number of engines is required. Consequently, an effective number of engines operator, ν_i, is defined to be:

$$\nu_i = \begin{cases} \nu_{i1} \cos \nu_{i2} + \nu_{i3} \cos \nu_{i4} & \text{if } \nu_i \text{ multiplies } F_i \text{ or } Ae_i \\ \nu_{i1} + \nu_{i3} & \text{if } \nu_i \text{ multiplies } \dot{m}_i \text{ or } c\dot{m}_i{}^* \end{cases}$$

The input thrust levels for first stage component rockets are considered to be nominal sea level thrusts. The total thrust, T, for all thrust events considered to be in the first stage is calculated from

$$T = \nu_i(F_i + Ae_i(p_s - p_a)) \equiv T_v - \nu_i Ae_i p_a$$

where p_s is the sea level atmospheric pressure and T_v is the vacuum thrust level.

The input thrust levels for all thrust events other than those in the first stage are considered to be vacuum thrust levels, and the total thrust is calculated from

---

*$c\dot{m}_i$ is defined in Section 7.4.2.

$$T = \nu_i(F_i - Ae_i p_a) \equiv T_v - \nu_i Ae_i p_a$$

while still in the atmosphere and

$$T = \nu_i F_i \equiv T_v$$

once the atmosphere is dropped.

The total flow rate, $\dot{m}$, in any thrust event is calculated from

$$\dot{m} = \nu_i \dot{m}_i$$

6.2 THRUST TABLES, FLOW-RATE TABLES AND DELTA-WEIGHT TABLES

If the input variables $JTHR_i = 0$, thrust and flowrate of the ith thrust event are calculated as above.

If $JTHR_i = +1$, $\dot{m}_i$ is calculated from a table of $\dot{m}_i$ vs time and F_i is calculated from a table of F_i vs. time. If $JTHR_i = -1$, F_i is calculated from a table of F_i vs. time and $\dot{m}(t)$ is calculated as

$$m(t) = m(t - \Delta t) + \Delta m(t - \Delta t) - \Delta m(t)$$

where $\Delta m(t)$ is a table of mass loss vs. time.

6.3 THROTTLING

If T_v/m in the ith thrust event is greater than $GLIM_i \equiv g_o \cdot GLIMG_i$ where GLIMG is an input vector, the throttle value τ is calculated as

$$\tau = m \cdot GLIM_i / T_v$$

and T_v and $\dot{m}$ become

$$T_v = \tau T_v = m\ GLIM_i$$

$$\dot{m} = \tau \dot{m} = (\dot{m}\ GLIM_i / T_v)\ m$$

Throttling should only be used when $JTHR_i = 0$.

6.4 STAGING ON FUEL

If the input variable $MSWCH_i = +1$, staging will be on time. If $MSWCH_i = -1$, the ith thrust event will terminate when a weight loss equal to $FUELG_i$ has occurred. The actual burn time τ_i will then be calculated. FUELG is an input vector.

7. THE FORWARD TRAJECTORY

This section contains the equations of motion integrated in LIFTING ROBOT, a description of the forward trajectory flight phases, the terminal functions which may be selected to define an optimization problem, and various users' options associated with the forward trajectory.

7.1 THE EQUATIONS OF MOTION

In general form, the equations of motion integrated in LIFTING ROBOT are written

$$\dot{p}_1 = \dot{w} = F_x/m + g_x$$

$$\dot{p}_2 = \dot{u} = F_y/m + g_y$$

$$\dot{p}_3 = \dot{v} = F_z/m + g_z$$

$$\dot{p}_4 = \dot{x} = w$$

$$\dot{p}_5 = \dot{y} = u$$

$$\dot{p}_6 = \dot{z} = v$$

$$\dot{p}_7 = \dot{\mu} = -m$$

$\dot{p}_8$ = Impact point penalty function. Integrated if terminal constraint No. 26 is activated. See Section 7.10.

$$\dot{p}_9 = T/m$$

$$\dot{p}_{10} = \frac{T}{m}\,[1 - (\hat{c} \cdot \hat{V}_I)]$$

$$\dot{p}_{11} = -(w \cdot g_x + u \cdot g_y + v \cdot g_z)/V_I$$

$$\dot{p}_{12} = q\ V_R/(\frac{\pi}{2} - \alpha)$$

$$\dot{p}_{13} = -\frac{qS}{m}\,[(b\cos\alpha - a)\,(\hat{V}_I \cdot \hat{V}_R) - (b + c)(\hat{c} \cdot \hat{V}_I)]$$

Equations 9 through 13 are integrated only on a converged trajectory.

The forcing functions F_x, F_y and F_z are

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = [T - qS(b + c)]\hat{c} + qS(b\cos\alpha - a)\hat{V}_R$$

with, of course, q set to zero when the atmosphere is dropped.

Equations of motion 9, 10, 11 and 13 are easily derived by forming the dot product $\dot{\bar{V}}_I \cdot \hat{V}_I$ and adding and subtracting T/m. p_9 is known as the characteristic velocity, p_{10} is known as the turning loss, p_{11} is known as the gravity loss and p_{13} is known as the drag loss. p_{12} is an aerodynamic heating indicator.

The mass, m, is calculated from

$$m = \mu + m_a$$

where μ is continuous and consists of the total propellants to be burned plus the payload, and

$$m_a = (\sum_i W_i^d + \sum_j W_j^J)/g_o$$

The constant g_o relates mass to weight and is taken to be

$$g_o = 9.80665 \text{m/sec}^2$$

Since m_a is constant from one weight drop to the next, $\dot{\mu} = -\dot{m}$ for all $t \neq t_{1i}$ or t_{2i}.

By this artifice, the seventh state variable, μ, is made to be continuous at all times, including those times at which mass is discontinuous. The primary advantage of integrating this particular choice of state variable is the ease with which mass can be reconstructed during the adjoint integration. Since μ is continuous, it may be stored as a function of time on the forward trajectory, and therefore, even if $\dot{m}$ is a time varying function obtained from a thrust tape, the mass can be calculated during the adjoint integration by looking μ up, updating m_a at the t_{1i} and t_{2i} and adding the two together.

7.2 GROUND-LAUNCH TRAJECTORY FLIGHT PHASES

The flight profile of a ground-launch trajectory is separated into a number of phases. These phases are depicted graphically for a booster having n-1 thrust events. The symbols in Fig. 8 are discussed below.

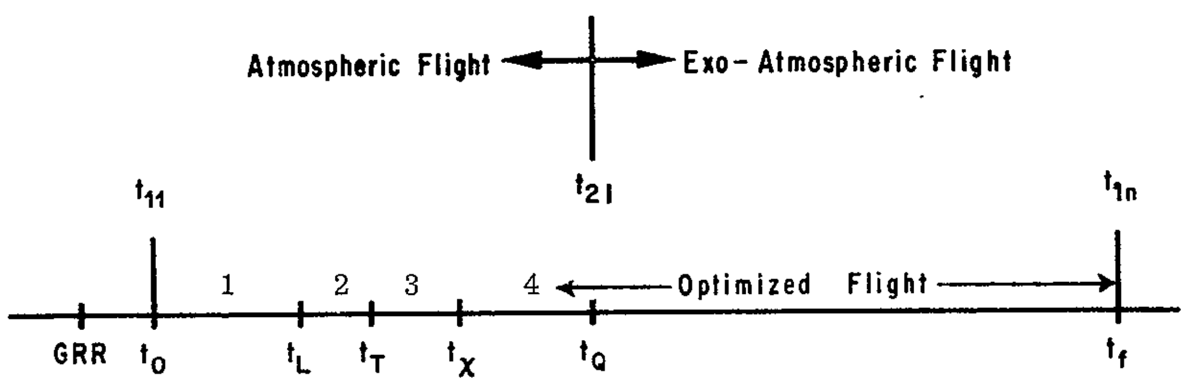

FIG. 8. GROUND-LAUNCH FLIGHT PHASES

7.2.1 GRR, Δt_o, t_o

The input quantity Δt_o is the time interval between the time the coordinate systems are defined, GRR, and the lift-off time, t_o. t_o is an input constant which is generally taken to be zero. t_{11}, the time the first thrust event begins, is set to t_o. If a non-zero value of Δt_o is used, the boost vehicle, which is fixed to the earth, will not be in the $\hat{Y}\hat{Z}$ plane at lift-off.

7.2.2 Ground-Launch Initial Conditions

The calculation of the initial, t_o, conditions for a ground-launch trajectory proceeds directly from Δt_o and the input value of the geodetic latitude of the launch site, θ_o. The geocentric colatitude of the launch site is

$$\theta = \pi/2 - \tan^{-1}((1-f)^2 \tan\theta_o)$$

The radius of the launch site is $R(\theta)$ and the initial velocity of the launch site is

$$V_o = R(\theta)\,\Omega_e \sin\theta$$

The longitude angle subtended by the launch site during the time interval Δt_o is

$$\Delta\phi_o = \Omega_e \Delta t_o$$

The initial plumbline velocity components are

$$\begin{bmatrix} w_o \\ u_o \\ v_o \end{bmatrix} = V_o A \begin{bmatrix} \cos\Delta\,\phi_o \\ 0 \\ -\sin\Delta\,\phi_o \end{bmatrix}$$

The initial plumbline position coordinates are

$$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = R(\theta)\, A \begin{bmatrix} \sin\Delta\ \phi_o \sin\theta \\ \cos\theta \\ \cos\Delta\ \phi_o \sin\theta \end{bmatrix}$$

The initial value of the seventh state variable is calculated from m_a, and the input value of initial mass, m_o, as

$$\mu_o = m_o - m_a$$

The initial value of state variables 9 - 13 is of course zero.

7.2.3 Lift-off -- Phase 1

The interval $t_o \rightarrow t_L$, Phase 1 of Fig. 8, is the lift-off portion of the trajectory. During this interval the control variables χ_p and χ_y are chosen so that the launch vehicle will clear the launch tower.

Since the launch tower is constructed normal to the reference ellipsoid, the angular separation of the launch tower and north is θ_L, where

$$\theta_L = \theta_1$$

The longitude of the launch site is ϕ_L, where

$$\phi_L = \Delta\phi_o + \Omega_e(t-t_o)$$

A unit vector in the launch tower direction can be transformed into the plumbline system as

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = A \begin{bmatrix} \sin\phi_L \sin\theta_L \\ \cos\theta_L \\ \cos\phi_L \sin\theta_L \end{bmatrix} = \begin{bmatrix} \sin\chi_p \cos\chi_y \\ \cos\chi_p \cos\chi_y \\ \sin\chi_y \end{bmatrix}$$

Therefore, in the interval $t_o \rightarrow t_L$, $\hat{c}$ is calculated to be

$$\hat{c} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix}$$

Since the A matrix and θ_L are constant and θ_L depends only on t; χ_p, χ_y and hence $\hat{c}$ during lift-off are functions of time only. t_L is an input constant.

7.2.4 Tilt-Over -- Phase 2

During the interval $t_L \rightarrow t_T$, Phase 2 of Fig. 8, the vehicle is caused to tilt over in the $\hat{x}\,\hat{y}$ plane by calculating χ_p and χ_y as

$$\chi_p = \dot{\chi}(t-t_L)$$

$$\chi_y = 0$$

where $\dot{\chi}$ is a trajectory parameter and t_T is an input constant.

During $\chi_y = 0$ flight, the equations given previously for $\hat{c}$ reduce to

$$\hat{c} = \begin{bmatrix} \sin\chi_p \\ \cos\chi_p \\ 0 \end{bmatrix}$$

7.2.5 Pitch-Plane Gravity Turn -- Phase 3

Following the tilt-over, a pitch-plane gravity turn can be flown in which

$$\chi_y = 0$$

and χ_p is chosen so that the angle of attack in the pitch $(\hat{x}\,\hat{y})$ plane is zero. This requires that during Phase 3

$$\hat{c} = \begin{bmatrix} \underline{w}/\sqrt{\underline{w}^2 + \underline{u}^2} \\ \underline{u}/\sqrt{\underline{w}^2 + \underline{u}^2} \\ 0 \end{bmatrix}$$

7.2.6 Optimal Flight -- Phase 4

The pitch-plane gravity turn terminates at t_χ, an input constant marking the beginning of Phase 4.

Prior to t_χ the thrust vector control angles χ_p and χ_y are obtained as a direct consequence of internal logic phases. After t_χ, χ_p and χ_y are considered to be tabular functions of time. Time, χ_p and χ_y can be specified at a maximum of 196 tabular points. These are broken up into four sets of control tables with a limit of 49 points each. Through input it is possible to specify the thrust event "picket" number at which control tables start and stop and the number of points in a table. Control tables should not continue across a coast or an intermediate point constraint*. Since Simpson's rule is used to integrate products of impulse response functions during the adjoint solution, there should always be an <u>odd</u> number of points in a control table.

The steepest ascent process converges on the optimal χ_p, χ_y time histories by updating the tabular control programs of χ_p and χ_y (if specified by input) at each iteration. If the input quantity KWTA is

---

*Described in Section 7.3.

set to 3, both χ_p and χ_y are varied. If KWTA is input as 2, χ_y is held at zero and χ_p is varied.

7.2.7 Exo-Atmospheric Flight

At t_Q the atmosphere is dropped and the 12th and 13th state variables are no longer integrated. The LIFTING ROBOT program shifts to a different set of derivative routines at this point in order to avoid bypassing terms that have to do with the atmosphere. The internal logic of LIFTING ROBOT is arranged so that t_Q is a miscellaneous weight drop event time, i.e.,

$$t_Q = t_{2I}$$

with I being the number of the miscellaneous weight drop event which terminates Phase 4. The end of Phase 4 (or possibly Phase 3 - see below). marks the end of atmospheric flight and hence t_Q must be defined on <u>every</u> ground launch trajectory. Note that this implies that there must always be at least one miscellaneous weight drop event. If none is actually desired, then a zero weight must be dropped.

7.2.8 Elimination of Phase 3 or Phase 4

If the input variable TCHFRZ is input equal to TTILT there will be no gravity turn and optimal flight will begin immediately after Phase 2.

If TCHFRZ is input greater than t_Q there will be no optimal flight within the atmosphere and Phase 3 will last until the edge of the atmosphere at which point optimal control begins.

7.3 INTERMEDIATE AND/OR TERMINAL FUNCTIONS

In order to define an optimization problem it is necessary to specify the trajectory constraints as well as the quantity to be maximized or minimized. Table 1 consists of a library of 26 (at present) intermediate and/or terminal functions and their formulas. Any one of these functions may be selected as the payoff and be maximized or minimized at the terminal time. Any physically realizable set of the remaining functions may be selected as trajectory constraints and imposed at the terminal time.. In addition, any physically realizable set of these functions may be imposed as constraints at an intermediate time by inputting the number of the thrust event following which the constraints are imposed as NVRST. Additional functions can be added by setting up a new GO TO number in ACSTOP and computing the magnitude of the function and its partial derivatives with respect to the first 7 plumbline states.

7.4 CONTROL PARAMETERS, PROPELLENT TANK LIMITS AND FLIGHT PERFORMANCE RESERVES

In addition to optimizing the χ_p and χ_y time histories, the LIFTING ROBOT program can simultaneously optimize control parameters selected by input from the control parameter library.

TABLE 1. FUNCTION LIBRARY

Code No.	Function Name	Symbol	Formula
1	Mass (Payload if payoff)	m	
2	Inertial Velocity	V_I	$V_I = \sqrt{w^2 + u^2 + v^2}$
3	Inertial Flight Path Angle	γ	$\gamma = \sin^{-1}\left(\frac{u_s}{V_I}\right)$
4	Radius	r	$r = \sqrt{x^2 + y^2 + z^2}$
5	Energy	C_3	$C_3 = V_I^2 - \frac{2\mu_e}{r}$
6	Angular Momentum	C_1	$C_1 = \sqrt{A^2 + B^2 + C^2}$ where $A = yv - uz$ $B = zw - vx$ $C = xu - wy$
7	Inertial Longitude	ϕ	$\phi = \tan^{-1}\left(\frac{a_{11}x + a_{21}y + a_{31}z}{a_{13}x + a_{23}y + a_{33}z}\right)$
8	Inertial Heading Angle	β	$\beta = \tan^{-1}\left(\frac{w_s}{v_s}\right)$
9	Colatitude	θ	$\theta = \cos^{-1}(a_{12}x + a_{22}y + a_{32}z)/r$
10	Inclination	i	$i = \cos^{-1}(\sin\theta \sin\beta)$

Table 1. (Cont'd)

Code No.	Function Name	Symbol	Formula
11	Line of Nodes	ω	$\omega = \phi + \tan^{-1}(\cos\theta \tan\beta)$
12	Semi-Latus Rectum	p	$p = r^2 (w_s^2 + v_s^2) / \mu_e$
13	Eccentricity	e	$e = \sqrt{1 + p \cdot C_3 / \mu_e}$
14	Radius of Perigee	Γ_p	These two constraints must be activated together. When they are specified by KCDPHI the state is propagated forward from t_f until $\gamma = 0$. The values of Γ at the points where $\gamma = 0$ are compared and the smaller is set in Γ_p and the larger in Γ_a. A set of adjoint equations is integrated backwards from both Γ_p and Γ_a to t_f. The values of the λ's at t_f are then stored into the vector partials of Γ_p and Γ_a with respect to the state at cutoff. A call to APPG initiates this process, and APPG is described in Appendix E.
15	Radius of Apogee	Γ_a	

Table 1 (Cont'd)

Code No.	Function Name	Symbol	Formula
16	True Anomaly	η	$q = (w_s^2 + v_s^2) \cdot r / \mu_e$ $\eta = \tan^{-1}\left(\frac{q u_s / \sqrt{w_s^2 + v_s^2}}{q - 1}\right)$
17	Argument of Perigee	a	$a = \eta + \tan^{-1}\left(\frac{\cos\theta}{\sin\theta\cos\beta}\right)$
18	Outgoing Asymptote	Asym	$A = \overline{r} \cdot \overline{s}$ $B = \sqrt{r^2 - A^2}$ $Asym = \frac{C_1}{\mu_e}(C_1 - B\sqrt{C_3}) - r + A$ $\overline{s}$ is a unit vector in direction of asymptote
19	Asymptote Plane	Asm Pl	$Asm\ Pl = \overline{C}_1 \cdot \overline{s}$ $\overline{C}_1$ is the angular momentum vector
20	Rendezvous W	Rend W	Rend W = $W - W_{target}$
21	Rendezvous U	Rend U	Rend U = $U - U_{target}$
22	Rendezvous V	Rend V	Rend V = $V - V_{target}$
23	Rendezvous X	Rend X	Rend X = $X - X_{target}$
24	Rendezvous Y	Rend Y	Rend Y = $Y - Y_{target}$
25	Rendezvous Z	Rend Z	Rend Z = $Z - Z_{target}$
26	Impact Penalty Function	PEN	PEN = p_8

7.4.1 <u>Control Parameters</u>

Table 2 contains the members of the control parameter library.

Library No.	Parameter Name	Symbol
1	1st thrust event duration	τ_{11}
2	2nd thrust event duration	τ_{12}
⋮	⋮	⋮
15	15th thrust event duration	$\tau_{1,15}$
16	Launch Weight	m_o
17	Tilt-over Chi-dot	$\dot{\chi}$
18	Launch Azimuth	A_z

TABLE 2 CONTROL PARAMETERS

The library number of each parameter to be optimized is specified by putting a nonzero value into the equivalently numbered elements of the input array KDB. Thus it is the position of nonzero elements in KDB which indicates an active parameter. Although all τ_{1i} are provided, a library number, only those τ_{1i} terminating after optimal control begins may be selected for optimization.

7.4.2 <u>Propellant Tank Limits</u>

In a great number of real problems the total propellant in a given stage is fixed, albeit allocated among a number of different thrust events. Also, since the available fuel and oxidizer will not, in general, be exhausted simultaneously when mixture-ratio shifts are considered, tank limits in LIFTING ROBOT are based upon "critical" propellant rather than actual propellant. In what follows it is assumed that $JTHR_i = 0$ for all thrust events involved, and that throttling does not occur.

The τ_{1i} can be connected by logic so as to maintain the relationship

$$m_x = \sum_i \nu_i \, c\dot{m}_i \, \tau_{1i}$$

where m_x, when tank limits alone are considered, is defined by the <u>input</u> values of the critical flow rate $c\dot{m}_i$ and the τ_{1i}. Since m_x cannot vary when the τ_{1i} are being varied by the steepest-ascent process

$$\Sigma_i \, \nu_i \, c\dot{m}_i \, d\tau_{1i} = 0$$

Therefore, all the connected thrust events cannot be optimized independently. One of the τ_{1i}, the jth, must be dependent and result from a choice of the others, i. e.,

$$d\tau_{1j} = -\sum_{i \neq j} \frac{\nu_i \, c\dot{m}_i}{\nu_j \, c\dot{m}_j} \, d\tau_{1i}$$

The procedure used in LIFTING ROBOT for specifying that the jth thrust event is connected to the ith with the ith being independent, i. e., KDB(i) $\neq$ 0, is to put the difference between j and i into the same element of the input array KDT, i.e., KDT(i) = j - i. One restriction on this procedure is that j must be greater than i. If no connection is desired, the appropriate element of KDT is set to zero. Note that if KDT(i) = j - i > 0, then KDB(j) must be zero since the same parameter cannot be specified as both dependent and independent. (If this requirement is not met, the program will print a warning, run a forward trajectory and go to the next case.) If, for example, KDB(4) $\neq$ 0 indicating that τ_{14} is to be optimized and if KDT(4) = 1, then τ_{15} is altered to keep m_x constant and KDB(5) must be zero. If, on the other hand, KDT(4) = 2, then τ_{14} is altered to keep m_x constant and KDB(6) must be zero. If, however, KDT(4) = 0, then τ_{14} is optimized without regard to limits.

As is implied by the equation for $d\tau_{1j}$, the same thrust event can be specified as dependent by more than one independent parameter. For example, the input arrays

KDT = 0, 0, 0, 4, 3, 2, 0, 0, 0, 0, 0, 0,0, 0, 0

KDB = 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1

indicate that m_o, $\dot{\chi}$, τ_{14}, τ_{15}, and τ_{16} are to be optimized and that

$$d\tau_{18} = -\frac{\nu_4 c\dot{m}_4}{\nu_8 c\dot{m}_8}\, d\tau_{14} - \frac{\nu_5 c\dot{m}_5}{\nu_8 c\dot{m}_8}\, d\tau_{15} - \frac{\nu_6 c\dot{m}_6}{\nu_8 c\dot{m}_8}\, d\tau_{16}$$

20

It should be noted that although the rationale for the development of the connection logic comes from the necessity of holding stage tank limits, the connection logic is independent of stage specification.

7.4.3 Flight Performance Reserves

Flight performance reserves (hereafter called FPR) is a name given to the propellants held in reserve on a design flight to provide an increment of velocity over and above the design velocity in the event it should be necessary on an actual flight. As such, FPR are jettisoned along with the jettison weight of the last thrust event and do not appear as part of the payload. Again it is assumed that $JTHR_i = 0$ for all thrust events involved, and that throttling does not occur.

The input quantity IPR is the number of the thrust event from which the FPR are withheld. If IPR = 0, FPR are not calculated. There are several accompanying requirements if IPR is not to be zero. First of all the IPR th thrust event must be in the last stage. Secondly, the maximum amount of critical propellant in the last stage, m_x, must be input as WPMX. Thirdly, although the IPR th does not have to be the last thrust event, no thrust event which follows the IPR th may be optimized.

The LIFTING ROBOT program calculates FPR on the basis of two input ΔV requirements. These are, ΔV_g to account for geometry perturbations and ΔV_p to account for performance perturbations. FPR are related to ΔV_g and ΔV_p through the equations

$$GPR = m_c(1 - e^{-\Delta V_g/Vex})$$

$$PPR = (m_c - GPR)(1 - e^{-\Delta V_p/Vex})$$

$$FPR = GPR + PPR$$

where m_c is the mass at cutoff of the IPR th thrust event, and $Vex = g_o I_{sp}$ of the IPR th thrust event. Defining

$$k_1 = 1 - e^{-\Delta V_g/Vex}$$

$$k_2 = 1 - e^{-\Delta V_p/Vex}$$

The FPR can be calculated as

$$FPR = m_c \, k_4$$

where

$$k_4 = k_1 + k_3$$

$$k_3 = (1 - k_1)\, k_2$$

Denoting IPR by j, and the mass at the beginning of the IPR th thrust event by m_j, the cutoff mass, m_c, can be written as

$$m_c = m_j - \nu_j \, \dot{m}_j \, \tau_{1j}$$

The problem of course is to find τ_{1j} such that the sum of the critical propellant contained in the FPR and that consumed during the remainder of the last stage is equal to m_x. This may be written

$$m_x = \sum_{i \neq j} \nu_i c\dot{m}_i \tau_{1i} + \nu_j c\dot{m}_j (\tau_{1j} + \tau_p)$$

where the summation by i is over the thrust events in the last stage, and τ_p is defined by

$$\tau_p = \frac{FPR}{\nu_j \dot{m}_j} = k_4 \left(\frac{m_j}{\nu_j \dot{m}_j} - \tau_{1j} \right)$$

This leads to

$$\tau_{1j} = \frac{1}{\nu_j c\dot{m}_j (1-k_4)} \left(m_x - k_4 m_j \frac{c\dot{m}_j}{\dot{m}_j} - \sum_{i \neq j} \nu_i c\dot{m}_j \tau_{1i} \right)$$

If in addition to FPR, τ_{1i} are optimized in the last stage, a different form of the equation for τ_{1j} is useful in the calculation of the steepest ascent influence coefficients. Denoting the mass at the beginning of the first thrust event in the last stage by m_L and noting that

$$m_j = m_L - \sum_{i<j} \nu_i \dot{m}_i \tau_{1i} - m^d$$

where m^d is the sum of the weights dropped (if any) in the interval between m_L and m_j, the equation for τ_{1j} may be written

$$\tau_{1j} = \frac{1}{\nu_j c\dot{m}_j (1-k_4)} \left(m_x - k_4 \frac{c\dot{m}_j}{\dot{m}_j} (m_L - m^d) + \sum_{i<j} \left(k_4 \frac{c\dot{m}_j}{\dot{m}_j} \dot{m}_i - c\dot{m}_i\right) \nu_i \tau_{1i}\right.$$

$$\left. - \sum_{i>j} \nu_i \, c\dot{m}_i \, \tau_{1i}\right)$$

The situation that exists when τ_{1i} in the last stage are optimized and FPR are calculated, and when there is KDT connection between the ith and IPR th thrust events is essentially the same, since m_x is constant in either case. The difference is that in straight KDT connection the input thrust event durations define an m_x, whereas with FPR, m_x = WPMX defines τ_{1j}. The similarity between IPR and KDT connection can readily be seen for the case where $\Delta V_g = \Delta V_p = 0$, in which case $k_4 = 0$, and for both IPR and KDT connection

$$d\tau_j = - \sum_{i \neq j} \frac{\nu_i \, c\dot{m}_i}{\nu_j \, c\dot{m}_j} \, d\tau_{1i}$$

The situations are in fact so similar logically that the LIFTING ROBOT program sets up and uses KDT connection logic whenever τ_{1i} are optimized in a stage that has FPR.

7.5 JUMP START

The input variable JUMP is the thrust event "picket" number at which a trajectory begins. If JUMP = 1, the trajectory will progress through the ground-launch logic. If JUMP $\neq$ 1, the trajectory will begin at that thrust event "picket" number. If JUMP $\leq$ NØVENT(1) the trajectory will begin in the atmosphere. If not, the trajectory will begin out of the atmosphere. The starting state is specified through the input array VIV, the starting time by TZERØ and the starting weight by WØ1. When there is a jump start, t_Q is set to TZERØ and all KDB and KDT below the jump start point are set to zero. IF VIV(7) = 0, the plumbline state w, u, v, x, y, z must read into VIV(1) - VIV(6). If VIV(7) = 2, V_I, γ, r, azimuth (A_z), latitude (θ') and ω must be read into VIV(1) → VIV(6).

Setting

$$a = 180 - A_z$$

and

$$\theta = 90 - \theta'$$

$$\underline{\omega} = \tan^{-1}(\cos\theta \tan a)$$

$$\phi = \omega - \underline{\omega}$$

Then, constructing a B matrix using θ and ϕ above and using the launch site A matrix, a D matrix can be constructed and used to calculate the initial plumbline state as

$$\begin{bmatrix} w_o \\ u_o \\ v_o \end{bmatrix} = V_I D' \begin{bmatrix} \cos\gamma \cos a \\ \sin\gamma \\ \cos\gamma \sin a \end{bmatrix}$$

$$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = r D \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

7.6 10 KM, QMAX, 14 KM

The program prints out as it crosses 10 km altitude, 14 km altitude and the point of maximum dynamic pressure. In order to find the latter, the time derivative of dynamic pressure, $\dot{q}$, is used. $\dot{q}$ is calculated by forming the dot product of the partials of q wrt the plumbline state, $\frac{\partial q}{\partial p}$, and the time derivatives of the plumbline state, $\dot{p}$. That is,

$$\dot{q} = \frac{\partial q}{\partial p} \dot{p}$$

where

$$\left[\frac{\partial q}{\partial p}\right]^T = \begin{bmatrix} \rho \underline{w} \\ \rho \underline{u} \\ \rho \underline{v} \\ \frac{q}{\rho}\frac{d\rho}{dh}\left(d_{12} - \frac{dR(\theta)}{d\theta} d_{13}\right) - \rho(a_{32}\underline{u} - a_{22}\underline{v})\,\Omega_e \\ \frac{q}{\rho}\frac{d\rho}{dh}\left(d_{22} - \frac{dR(\theta)}{d\theta} d_{23}\right) - \rho(a_{12}\underline{v} - a_{32}\underline{w})\,\Omega_e \\ \frac{q}{\rho}\frac{d\rho}{dh}\left(d_{32} - \frac{dR(\theta)}{d\theta} d_{33}\right) - \rho(a_{22}\underline{w} - a_{12}\underline{u})\,\Omega_e \end{bmatrix}$$

and $\frac{d\rho}{dh}$ is calculated numerically using the PRA63 atmosphere routine.

7.7 IMPACT POINT

The LIFTING ROBOT program integrates the trajectory of the jettison weight of the IMPth thrust event (W^J_{IMP}) to impact $(h = 0)$ if the input constant IMP is > 0. The forcing functions F_x, F_y and F_z on the impact trajectory are

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} -\rho V_R \underline{w} \\ -\rho V_R \underline{u} \\ -\rho V_R \underline{v} \end{bmatrix} .$$

where ρ = 0 for altitudes greater than 690 km and ρ calculated from PRA63 as a function of altitude for $h < 690$ km. Only the first 6 equations of motion are integrated on the impact trajectory.

7.8 ANALYTIC COAST

Universal coast equations have been installed in LIFTING ROBOT. These equations solve the two-body problem in cartesian coordinates. A description of the method is contained in Appendix A.

An analytic coast may be specified in any two zero-thrust thrust events. The method of specification is through NCØST1 and NCØST2.

NCØST1 = No. of thrust event for 1st coast
NCØST2 = No. of thrust event for 2nd coast

If the analytic coast is used, the problem will print out START COAST and END COAST at the beginning and end of an analytic coast. No print-out will occur during an analytic coast. The analytic coast should be used with a spherical earth. It goes without saying that the analytic coast results in a great time savings if long coasts are considered.

7.9 RENDEZVOUS

If terminal functions 20 through 25 are activated, a rendezvous will take place and additional input, which gives the position and velocity of the target at t_o, is required. The position and velocity of the target in the reference coordinate system at the nodal point are calculated using the radius of the target orbit at the nodal point, its flight path angle, inclination, and velocity; RTGT, GAMTGT, INCTGT and VELTGT, respectively.

$$x_N = 0$$

$$y_N = 0$$

$$z_N = \text{RTGT}$$

$$w_N = \text{VELTGT} \cos(\text{GAMTGT}) \cos(\text{INCTGT})$$

$$u_N = \text{VELTGT} \cos(\text{GAMTGT}) \sin(\text{INCTGT})$$

$$v_N = \text{VELTGT} \sin(\text{GAMTGT})$$

The position and velocity of the target at any later time, t, are calculated by coasting τ_{tar} seconds using the initial conditions specified above.

The target is positioned in its orbit through the use of the input variable BTATGT. The input state defined above is caused to coast a length of time equal to

$$\tau_{tar} = t + \frac{\text{BTATGT}}{\text{RAD}} \cdot \frac{\text{RTGT}}{\text{VELTGT}}$$

where t is current time in LIFTING ROBOT and BTATGT is an angle turned through in a circular orbit.

The final target state is then rotated into the plumbline system using the A matrix. The difference between this state and the vehicle state at t_f must be zero for a rendezvous to occur.

7.9.1 Rendezvous Options

If IAA is input as 1, the launch azimuth will be calculated internally to coincide with the azimuth of the target as it passes the launch site, i.e.,

$$A_z = \sin^{-1}(\cos(\text{INCTGT})/\sin\theta_1)$$

7.10 THE IMPACT POINT PENALTY FUNCTION STATE VARIABLE

If IPCNST $\neq 0$, the derivative for the eighth state variable is calculated by mapping the instantaneous state down to R_e using an analytic form for the f and g series. (See Appendix E). The elapsed time, DTI, to impact is also calculated. The latitude and longitude at impact θ_p and ϕ_p respectively are then calculated. The calculations are

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = f \begin{bmatrix} x \\ y \\ z \end{bmatrix} + g \begin{bmatrix} w \\ u \\ v \end{bmatrix}$$

$$\text{DTI} = r\, S_1 + \sigma_o\, S_2 + \mu_e\, S_3$$

$$\theta_p = \sin^{-1}((x_p a_{12} + y_p a_{22} + z_p a_{32})/R_e)$$

$$\phi_p = \tan^{-1}\left[\frac{x_p a_{11} + y_p a_{21} + z_p a_{31}}{x_p a_{13} + y_p a_{23} + z_p a_{33}}\right] - Re(t + \Delta t_o + DTI) - \phi_o$$

The following vectors are then formed

$$\begin{bmatrix} \Delta\phi_1 \\ \vdots \\ \Delta\phi_{N\emptyset I} \end{bmatrix} = \begin{bmatrix} \phi_p - LL\emptyset_1 \\ \vdots \\ \phi_p - LL\emptyset_{N\emptyset I} \end{bmatrix}$$

and

$$\begin{bmatrix} \Delta\theta_1 \\ \vdots \\ \Delta\theta_{N\emptyset I} \end{bmatrix} = \begin{bmatrix} \theta_p - LLA_1 \\ \vdots \\ \theta_p - LLA_{N\emptyset I} \end{bmatrix}$$

where LLA = TH1/RAD where THL is an input vector of latitudes and LL$\emptyset$ = PH1/RAD where PH1 is an input vector of longitudes of vectors of impact area allipses. N$\emptyset$I is the number of active ellipses.

There follows

$$\epsilon_i = \sigma^2_{\phi i}\,\Delta\phi_i^2 + \rho_{12i}\,\Delta\phi_i\Delta\theta_i + \sigma^2_{\theta i}\,\Delta\theta_i^2,$$

$$i = 1,\ N\emptyset I$$

where $\sigma^2_{\phi i}$, ρ_{12i} and $\sigma^2_{\theta i}$ are the arrays S2LØ, RH12 and S2LA respectively which are calculated in AINIT.

If ϵ_i is greater than 1, ϵ_i is redefined to be

$$\epsilon_i = 1. + \epsilon_{k_i}(\epsilon_i - 1)$$

$$z_i = c_{n_i} e^{-\epsilon_i}$$

where c_{n_i} and ϵ_{k_i} are set to 1 and 2 respectively. Then

$$\dot{p}_8 = \sum_{i=1}^{N\emptyset I} z_i$$

This has the effect of setting up a state variable which increases at two different rates depending on the closeness of the impact point to one of the impact area centers. The ϵ_{k_i} are intended to be used to cause very little penalty to be generated when far from an impact area center, and the c_{n_i} are intended to weight the centers relative to each other.

The arrays S2LØ, S2LA and RH12 are calculated as

$$\rho_i^2 \equiv (\text{LATWTH}_i/\text{LONWTH}_i)^2 - 1$$

$$\alpha \equiv \text{R}\emptyset\text{TA}/\text{RAD}$$

$$\beta_i \equiv \text{RAD}/\text{LATWTH}_i$$

$$S2L\emptyset_i = \beta_i^2 (1 + \rho_i^2 \cos^2 \alpha_i)$$

$$S2LA_i = \beta_i^2 (1 + \rho_i^2 \sin^2 \alpha_i)$$

$$RH12_i = \beta_i^2 \rho_i^2 \sin 2\alpha_i$$

where LATWTH and LONWTH are input arrays of latitude width and longitude width assuming the impact area elipses were aligned with lines of constant latitude and longitude, and ROTA is an input array of ellipse rotation angles.

7.11 OUTPUT TABLES

By inputting a nonzero value of NTABLE output tables suitable for publication can be obtained. The output tables are printed only for converged trajectories.

If tables are desired, additional input described in Appendix D is required.

8. THE BACKWARD TRAJECTORY

Since the steepest ascent method converges on the optimum set of controls by adding beneficial changes to the nominal set, the effect of small changes in the controls on the terminal and intermediate functions must be calculated. This is accomplished through the use of the adjoint differential equations. One solution of the adjoint differential equations is required for each terminal or intermediate function being either optimized or constrained. The adjoint solutions proceed backward in time from the final time for the payoff and terminal constraints, and from the intermediate constraint time if there are intermediate constraints.

The adjoint variables are used to form impulse response functions which give the effect of changes in χ_p and $\dot{\chi}_y$ and influence coefficients which give the effect of changes in the parameters. These impulse response functions and influence coefficients are then used in the steepest ascent formulae to calculate beneficial changes in the controls.

Notation traditionally used to describe the adjoint solution is introduced below.

ϕ — The scalar payoff function.

ψ — An m x 1 matrix of constraints. Includes both terminal and intermediate constraints. (Constraints satisfied when $\psi = 0$.)

An m x 1 matrix of constant Langrange multipliers associated with the constraints. (This ν should not be confused with the effective number of engines ν_i defined in Section 7.)

Φ The augmented scalar payoff function $\phi + \nu^T \psi$.

λ_ϕ A 7 x 1 matrix of particular adjoint solutions associated with the payoff function.

λ_ψ A 7 x m matrix of particular adjoint solutions associated with the constraints.

λ A 7 x 1 matrix of adjoint solutions associated with the function Φ. When appearing without a subscript, λ is the equivalent of the Euler-Lagrange variables used in the calculus of variations (c.o.v. λ's) and are formed as

$$\lambda = \lambda_\phi + \lambda_\psi \nu$$

8.1 BOUNDARY CONDITIONS

The boundary condition on the Euler-Lagrange variables λ are known to be

$$\lambda^T = \frac{\partial \Phi}{\partial p}$$

Consequently, the boundary conditions on λ_ϕ and λ_ψ are chosen to be

$$\lambda_\phi^T = \left.\frac{\partial \phi}{\partial p}\right|_{t = t_f}$$

and

$$\lambda_\psi^T = \left.\frac{\partial \psi}{\partial p}\right|_{t = \begin{cases} t_f & \text{for terminal constraints} \\ t_{1j},\ j = \text{NVRST} + 1 & \text{for intermediate constraints} \end{cases}}$$

8.2 THE ADJOINT DIFFERENTIAL EQUATIONS

Defining the 8 x m + 1 matrix λ_z to be $\lambda_z = [\lambda_\phi \vdots \lambda_\psi]$, a vector variational Hamiltonian H_z can be defined as $H_z = \lambda_z^T \dot{p}$. The Euler-Lagrange or adjoint differential equations become

$$\dot{\lambda}_z = -\left(\frac{\partial H_z}{\partial p}\right)^T \quad \text{(for backwards integration)}$$

m + 1 sets of adjoint equations are integrated backwards to t_Q* (one set for ϕ and one set for each of the m ψ's). The reconstruction of the plumbline state, needed to calculate $(\partial H_z/\partial p)$ during the adjoint run, is accomplished by looking up stored values of the state as a function of time.

The differential equations for the λ_z are most conveniently written in vector form. To that end, define

$$\lambda_z^v \equiv \begin{bmatrix} \lambda_z^w \\ \lambda_z^u \\ \lambda_z^v \end{bmatrix}$$

$$\lambda_z^r \equiv \begin{bmatrix} \lambda_z^x \\ \lambda_z^y \\ \lambda_z^z \end{bmatrix}$$

---

* t_Q in Section 8 is considered to be the time where optimal control begins.

where the superscript on λ corresponds to the state derivative it multiplies in $\lambda_z^T \dot{p}$. The total λ_z vector is now

$$\lambda_z = \begin{bmatrix} \lambda_z^v \\ --- \\ \lambda_z^r \\ --- \\ \lambda_z^m \\ \lambda_z^p \end{bmatrix}$$

Out of the atmosphere $\dot{\lambda}_z$ can be written

$$\dot{\lambda}_z^v = \lambda_z^r$$

$$\dot{\lambda}_z^r = J^* \lambda_z^v$$

$$\dot{\lambda}_z^m = \begin{cases} -[F_x \; F_y \; F_z] \cdot \lambda_z^v / m^2 & \text{if not throttling} \\ -\lambda_z^m \; \dot{m} \; GLIM/T_v & \text{if throttling} \end{cases}$$

$$\dot{\lambda}_z^p = 0$$

---

* See Appendix C for a definition of J.

In the atmosphere $\dot{\lambda}_z$ can be written

$$\dot{\lambda}_z^v = \lambda_z^r - \alpha_2 \overline{V}_r + \alpha_3 \lambda_z^v + \alpha_4 \hat{c}$$

$$\dot{\lambda}_z^r = J \lambda_z^v - \alpha_1 \left(\frac{\partial h}{\partial \overline{r}}\right)^T - \alpha_2 \left(\frac{V_r \partial V_r}{\partial \overline{r}}\right)^T + \alpha_3 \frac{\partial(\lambda_z^v \cdot \overline{V}_r)^T}{\partial \overline{r}} + \alpha_4 \frac{\partial(\hat{c} \cdot \overline{V}_r)^T}{\partial \overline{r}}$$

$$\dot{\lambda}_z^m = \begin{cases} -[F_x \; \dot{F}_y \; F_z] \cdot \lambda_z^v / m^2 & \text{if not throttling} \\ \frac{qS}{m^2} \left(\left((b+c)\, \hat{c} - (b \cos \alpha - a) \hat{V}_r \right) \cdot \lambda_z^v \right) - \lambda_z^m \dot{m} \text{ GLIM/THR} & \text{if throttling} \end{cases}$$

$$\dot{\lambda}_z^p = 0$$

where

$$\alpha_1 = \left[\frac{A_e}{m} \frac{dp}{dh} + \frac{qS}{m}\left((b+c) \frac{1}{\rho} \frac{d\rho}{dp} - M \frac{\partial(b+c)}{\partial M} \frac{1}{s} \frac{ds}{dh}\right)\right] (\lambda_z^v \cdot \hat{c})$$

$$- \frac{qS}{m}\left([b \cos \alpha - a) \frac{1}{\rho} \frac{d\rho}{dh} - M\left(\frac{\partial b}{\partial M} \cos \alpha - \frac{\partial a}{\partial M}\right) \frac{1}{s} \frac{ds}{dh}\right] \overline{(\lambda_z^v \cdot \hat{V}_R)}$$

$$\alpha_2 = \frac{1}{2} \frac{\rho S}{m} \left[\left(a - M\left(\frac{\partial b}{\partial M} \cos \alpha - \frac{\partial a}{\partial M}\right)\right)(\lambda_z^v \cdot \hat{V}_R) + \left(2(b+c) + M \frac{\partial(b+c)}{\partial M}\right)(\lambda_z^v \cdot \hat{c}) \right]$$

$$\alpha_3 = \frac{1}{2} \frac{\rho S V_R}{m} (b \cos \alpha - a)$$

$$\alpha_4 = \frac{1}{2} \frac{\rho S V_R}{m} b \, (\lambda_z^v \cdot \hat{v}_R)$$

also, $\cos \alpha = (\hat{c} \cdot \hat{V}_R)$,

$\frac{\partial a}{\partial M}$, $\frac{\partial b}{\partial M}$ and $\frac{\partial c}{\partial M}$ are calculated by the cubic fit SPLINE routine,

and

$\frac{dp}{dh}$, $\frac{1}{\zeta}\frac{d\rho}{dh}$ and $\frac{1}{s}\frac{dS}{dh}$ are calculated analytically by PRB63

In addition,

$$\left(\frac{\partial h}{\partial \bar{r}}\right)^T = \begin{cases} \frac{\bar{r}}{r} & \text{if spherical earth} \\ \frac{\bar{r}}{r} - \left(\bar{r} \cos\theta - r \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix}\right) \frac{R(\theta)^3}{r^2} \cdot \frac{f(2-f)\cos\theta}{R_e^2(1-f)^2} & \text{if oblate earth} \end{cases}$$

$$\left(V_R \frac{\partial V_R}{\partial \bar{r}}\right) = \Omega_e \begin{bmatrix} \underline{v}\, a_{22} - \underline{u}\, a_{32} \\ \underline{w}\, a_{32} - \underline{v}\, a_{12} \\ \underline{u}\, a_{12} - \underline{w}\, a_{22} \end{bmatrix}$$

$$\frac{\partial(\lambda_z^v \cdot \bar{v}_R)^T}{\partial \bar{r}} = \Omega_e \begin{bmatrix} \lambda_z^v a_{22} - \lambda_z^u a_{32} \\ \lambda_z^w a_{32} - \lambda_z^v a_{12} \\ \lambda_z^u a_{12} - \lambda_z^w a_{22} \end{bmatrix}$$

$$\frac{\partial(\hat{c} \cdot V_R)^T}{\partial \bar{r}} = \Omega_e \begin{bmatrix} \sin\chi_y\, a_{22} - \cos\chi_p \cos\chi_y\, a_{32} \\ \sin\chi_p \cos\chi_y\, a_{32} - \sin\chi_y\, a_{12} \\ \cos\chi_p \cos\chi_y\, a_{12} - \sin\chi_p \cos\chi_y\, a_{22} \end{bmatrix}$$

Note that $\dot{\lambda}_z^p = 0$ always, hence λ_z^p is not integrated. If one element of z happens to represent the penalty function, then that element of λ_z^p is 1 and all others are zero. However, if one element of z happens to represent the penalty function, the derivatives of λ_p^v and λ_p^r are affected. This can be seen by noting that since $\lambda_p^p = 1$, the variational Hamiltonian will be

$$H_p = \lambda_p^v \cdot \dot{\overline{V}}_I + \lambda_p^r \cdot \overline{V}_I - \lambda_p^m \dot{m} + \dot{p}_8$$

Therefore, to the standard derivatives of H with respect to the first seven states must be added

$$\left[\left(\frac{\partial \dot{p}_8}{\partial \overline{V}_I}\right) \vdots \left(\frac{\partial \dot{p}_8}{\partial \overline{r}}\right) \right]^T$$

These partial derivatives are calculated analytically in IMPPT. (See Appendix F).

8.3 IMPULSE RESPONSE FUNCTIONS AND I INTEGRALS

The impulse response functions for χ_p and χ_y are defined by the equations

$$G_{zp}^T = \frac{\partial(\lambda_z^T \dot{p})}{\partial \chi_p} = T_m (\Gamma_z^w \cos\chi_y \cos\chi_p - \Gamma_z^u \cos\chi_y \sin\chi_p)$$

$$G_{zy}^T = \frac{\partial(\lambda_z^T \dot{p})}{\partial \chi_y} = T_m (-\Gamma_z^w \sin\chi_y \sin\chi_p - \Gamma_z^u \sin\chi_y \cos\chi_p + \Gamma_z^v \cos\chi_y)$$

where

$$T_m \equiv \begin{cases} \dfrac{T}{m} & \text{out of the atmosphere} \\ \dfrac{T - qS\,(b+c)}{m} & \text{in the atmosphere} \end{cases}$$

$$\begin{bmatrix} \Gamma_z^w \\ \Gamma_z^u \\ \Gamma_z^v \end{bmatrix} \equiv \begin{cases} \lambda_z^v & \text{out of the atmosphere} \\ \lambda_z^v + (\lambda_z^v \cdot \hat{V}_R)\,\dfrac{qSb}{T_m}\,\hat{V}_R & \text{in the atmosphere} \end{cases}$$

Impulse response functions are calculated at every tabular point in use in the $\chi_p - \chi_y$ control tables.

Denoting G_z by

$$G_z^T = \begin{bmatrix} G_{\phi p} \\ \hline G_{\psi p} \end{bmatrix} \quad \text{if KWTA} = 2$$

$$G_z^T = \begin{bmatrix} G_{\phi p} & G_{\phi y} \\ \hline G_{\psi p} & G_{\psi y} \end{bmatrix} \quad \text{if KWTA} = 3$$

The $m+1 \;\; x \;\; m+1$ matrix of control variable "I" integrals is calculated during the backward trajectory as

$$I_{zz}^a = \left[\begin{array}{c|c} I_{\phi\phi}^a & I_{\phi\psi}^a \\ \hline I_{\psi\phi}^a & I_{\psi\psi}^a \end{array}\right] = \int_{t_Q}^{t_f} G_z^T W_a^{-1} G_z \, dt, \quad I_{zz}^a(t_f) = 0$$

where W_a^{-1} is a time varying weighting matrix defined in Section 8.6.

If there are intermediate constraints, the above definitions of I_{zz}^{a} may be used provided that after the intermediate constraint time the elements of G_z corresponding to the intermediate constraints are taken to be zero.

8.4 INFLUENCE COEFFICIENTS

The influence coefficients for a parameter give the changes in trajectory functions resulting from a unit change in that parameter and hence may be considered trajectory to trajectory partial derivatives. The influence coefficients for lift-off weight and tilt-over $\dot{\chi}$ are calculated using numerical derivatives; whereas, those for the τ_{1i} are calculated using analytic partials.

8.4.1 Influence Coefficients for Lift-Off Weight, Tilt-Over $\dot{\chi}$ and A_z

If the launch weight is to be optimized, two trajectories are run from $t_o \rightarrow t_Q$ with m_o changed by $\pm \Delta m_o$. The influence coefficients for the 16th parameter are then calculated as

$$L_{zm} = \left[\frac{p^{+}(t_Q) - p^{-}(t_Q)}{2\,\Delta_{mo}} \right]^T \lambda_z (t_Q)$$

where p^{+} and p^{-} refer to the plumb line state from positive and negative variations of Δm_o respectively.

If the tilt-over $\dot{\chi}$ is to be optimized, two trajectories are run from $t_o \rightarrow t_Q$ with $\dot{\chi}$ changed by $\pm \Delta\dot{\chi}$. The influence coefficients for the 17th parameter are then calculated as

$$L_{z\chi} = \left[\frac{p^{+}(t_Q) - p^{-}(t_Q)}{2\,\Delta\dot{\chi}} \right]^T \lambda_z(t_Q)$$

If the launch azimuth A_z is to be optimized, one trajectory is run from t_o to t_f with A_z changed by ΔA_z. The influence coefficients for the 18th parameter are then calculated as

$$L_{zA_z} = \left[\left(\frac{\phi}{\psi} \right)^{+} - \left(\frac{\phi}{\psi} \right) \right] / \Delta A_z$$

8.4.2 Influence Coefficients for the τ_{1i}

The calculation of the influence coefficients for the τ_{1i} proceeds through three phases. In the first phase the influence coefficients for the effect of shifting the time at which a discontinuity occurs are calculated as

$$L_{\dot{z}i} = \Delta \dot{p}^T \lambda_z$$

for the effect of shifting each t_{1i} and as

$$Y_{zj} = \Delta \dot{p}^T \lambda_z$$

for the effect of shifting each t_{2j} where $\Delta\dot{p} = [\dot{p}^- - \dot{p}^+]$ is the discontinuity in the plumbline state derivatives resulting from a discontinuous change in either thrust or mass or both. The $\Delta\dot{p}$ are calculated and stored during the forward trajectory; and L_{zi} and Y_{zj} are calculated and stored during the backward trajectory.

When t_{1i} is the final time, $\dot{p}^+$ is set to zero. When t_{1i} is the intermediate orbit time, $\dot{p}^+$ is set to zero for the multiplication of those columns of λ_z corresponding to the intermediate constraints. In the second phase cognizance is taken of the fact that the t_{2j} are pinned to the t_{1i} via τ_j^w. Since the τ_j^w are constant,

$$dt_{2j} = dt_{1i}, \quad i = \text{NOWD}(j)$$

and therefore the following additions are performed in sequence with j running from $1 \rightarrow nw$

$$L_{zi} = L_{zi} + Y_{zj}, \quad i = \text{NOWD}(j)$$

where nw is the total number of miscellaneous weight drop events.

In the third phase cognizance is taken of the fact that for the τ_{1i} to be parameters

$$\frac{\partial t_{1j}}{\partial \tau_i} = \frac{\partial t_{1i}}{\partial \tau_i} = 1, \quad j > i$$

and therefore the following additions are performed in sequence with i running from $nv-1 \rightarrow 1$

$$L_{zi} = L_{zi} + L_{zj}\,, \quad j = i + 1$$

where nv is the total number of thrust events.

8.4.3 Influence Coefficients with Tank Limits and FPR

If flight performance reserves are withheld from the jth thrust event, the influence coefficients for launch weight and τ_{1i} become

$$L_{zm} = L_{zm} - \nu_j \dot{m}_j \frac{k_4}{(1-k_4)} L_{zj}$$

$$L_{zi} = L_{zi} + \frac{\nu_i \dot{c}m_i k_4}{\nu_j \dot{m}_j (1-k_4)} L_{zj} \quad (i < i_L)$$

where i_L is the first thrust event in the last stage, and

$$L_{zi} = L_{zi} - \left(1 - k_4 \frac{\dot{c}m_j \dot{m}_i}{\dot{c}m_i \dot{m}_j}\right) \frac{\nu_i \dot{c}m_i}{\nu_i \dot{c}m_j (1 - k_4)} L_{zj} \quad (i_L \leq i < j)$$

In the above equations, when and if the element corresponding to payload is augmented, k_4 is set to zero.

The proper augmentation of L_{zi}, when tank limits alone are considered and the jth thrust event is connected to the ith via KDT, is

$$L_{zi} = L_{zi} - \frac{\nu_i \dot{c}m_i}{\nu_j \dot{c}m_j} L_{zj} \quad (i < j)$$

Note that this result can be obtained from the last equation given above for FPR if k_4 and i_L are taken to be zero.

8.4.4 Influence Coefficients with Staging on Fuel

If any of the thrust events terminate on fuel, i.e., $\dot{MSWCH}_i < 0$, then the launch weight influence coefficient and the L_{zi} must be altered. This is becuase a change in mass causes a change in burn times (if there is throttling). Since fuel is constant, the difference between initial mass m_i and final mass m_f of the thrust event is constant, i.e.,

$$\text{fuel} = m_i - m_f = \text{const}$$

Therefore

$$dm_i = dm_f$$

Also, the mass when throttling begins is always constant since throttling begins when T/m = GLIM. Thus

$$T/GLIM = m_i - \dot{m}\,(t_\tau - t_i)$$

and hence

$$dt_\tau = \frac{dm_i}{\dot{m}}$$

Since dm/dt when there is throttling is

$$\frac{dm}{dt} = - \dot{m} \frac{GLIM}{T_v} m$$

$$m_f \doteq \frac{T_v}{GLIM} \exp\left(- \frac{\dot{m}(GLIM)}{T_v} (t_f - t_T)\right)$$

and

$$dm_f = dm_i = - \frac{\dot{m}\, GLIM}{T_v} m_f \left(dt_f - \frac{dm_i}{\dot{m}}\right)$$

Therefore

$$\frac{dt_f}{dm_i} = \left(\frac{1}{\dot{m}} - \frac{1}{\frac{\dot{m}\ GLIM\ m_f}{T_v}}\right) = \left(\frac{1}{\dot{m}_i} - \frac{1}{\dot{m}_f}\right)$$

Notice that if there is no throttling

$$\frac{dt_f}{dm_i} = 0 \text{ since } \dot{m}_i = \dot{m}_f$$

If the launch weight parameter is active and there is staging on fuel in say the jth thrust events

$$L_{zm} = L_{zm} + \sum_j L_{zj} \left(\frac{dt_f}{dm_i}\right)_j$$

In addition, the following is done for each active τ_i incluence coefficient

$$L_{z_i}^{\cdot} = L_{z_i} + \sum_{j>i} L_{z_j}^{\cdot} (\dot{m}_f)_i \left(\frac{dt_f}{dm_i}\right)_j$$

8.4.5 Parameter I Matrices

Grouping the active parameters into a matrix $\dot{L}_z = [L_\phi \vdots L_\psi]$ the m+1 x m+1 parameter "I" matrix I_{zz}^b can be formed as

$$I_{zz}^{b} = \left[\begin{array}{c|c} I_{\phi\phi}^b & I_{\phi\psi}^b \\ \hline I_{\psi\phi}^b & I_{\psi\psi}^b \end{array}\right] = L_z^T W_b^{-1} L_z$$

where W_b^{-1} is a weighting matrix defined in Section 8.6

8.5 STEEPEST ASCENT FORMULAE

Denoting the vector of active control parameters by b, and the vector of active control variables by a, (if KWTA = 2, a is a scalar equal to χ_p) the steepest ascent formulae for the changes in the controls are

$$\delta a = \pm W_a^{-1} (G_\phi - G_\psi I_{\psi\psi}^{-1} I_{\psi\phi}) E - W_a^{-1} G_\psi I_{\psi\psi}^{-1} k\psi$$

$$db = \pm W_b^{-1} (L_\phi - L_\psi I_{\psi\psi}^{-1} I_{\psi\phi}) E - W_b^{-1} L_\psi I_{\psi\psi}^{-1} k\psi$$

where

$$I_{\psi\psi} = I^a_{\psi\psi} + I^b_{\psi\psi}$$

$$I_{\psi\phi} = I^a_{\psi\phi} + I^b_{\psi\phi}$$

In the control equations above the plus sign is used when ϕ is to be maximized, the minus sign is used when ϕ is to be minimized, $0 \leq E \leq 1$ is a constant chosen to aid convergence, ψ is the vector of terminal constraints violations, and k is the decimal fraction of the constraint violation to remove.

If there are connected thrust events involving tank limits only, the $d\tau_{1i}$ for optimized thrust events will appear as elements of the db vector and the corresponding $d\tau_{1j}$ must be calculated as indicated in Section 7.4.2.

The changes in the controls calculated using the above equations are then added to the nominal set to get the controls for the next iteration.

The change in the payoff function ϕ resulting from the control changes is

$$d\phi = \pm (I_{\phi\phi} - I_{\phi\psi} I^{-1}_{\psi\psi} I_{\psi\phi}) E - I_{\phi\psi} I^{-1}_{\psi\psi} k\psi$$

where the sign is chosen as before and

$$I_{\phi\phi} = I^a_{\phi\phi} + I^b_{\phi\phi}$$

8.6 THE AUTOMATIC CONVERGENCE SCHEME

It is the function of the automatic scheme to pick k, E, W_a^{-1} and W_b^{-1} in order to speed convergence, and to terminate a run when it does converge. The logic for picking k and E is straightforward and is directly related to iteration number. On the first iteration, E is set to zero and k is chosen such that $.25 \leq k \leq 1$. A starting value of k can be input as DP2, however the program will ignore $k < .25$ or $k > 1$. If k is input ≥ 1 the iteration number is advanced to 2.* On the second iteration $k = 1$ and $E = 0$. On the third iteration $k = 1$ and $E = QY/2$, where QY is an input constant which should be chosen $0 < QY \leq 1$. On the fourth and subsequent iterations $k = 1$ and $E = QY$.

The choice of the weighting matrices W_a^{-1} and W_b^{-1} is also dependent on iteration number. On the first iteration W_a^{-1} is chosen to be

$$W_a^{-1} = \begin{cases} \dfrac{m}{T}^{**} & \text{if KWTA} = 2 \\ \begin{bmatrix} \dfrac{m}{T} & 0 \\ 0 & \dfrac{m}{T} \end{bmatrix} & \text{if KWTA} = 3 \end{cases}$$

---

*If k starts at .25 the k sequence is .25, .5, 1. ... with E = 0., 0., 0., QY/2, QY ...

**m/T is defined to be $1/T_m$

and W_b^{-1} is chosen to be

$$W_b^{-1} = \begin{bmatrix} W_1 P_1 & 0 & \cdots & 0 \\ 0 & W_2 P_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & W_n P_n \end{bmatrix}$$

where the W_i are an input set of weighting numbers for the np active parameters. The W_i are input as WIBT and should generally be left at their preset value of 1 unless experience dictates otherwise. On the first and subsequent iterations the P_i are chosen automatically so that the largest contribution of the ith parameters to the diagonal of $I^a_{\psi\psi}$ is equal to one. Denoting the influence coefficients of the ith parameter on the constraints by L^i_ψ, the P_ith scale factor is

$$P_i = \frac{1}{\underset{j=1,m}{\text{Max}} \left(\dfrac{(L^i_{\psi j})^2}{I^a_{\psi_j \psi_j}} \right)}$$

For the second and third iterations the constant Lagrange multipliers on the constraints, ν, are

$$\nu = -I^{-1}_{\psi\psi} I_{\psi\phi}$$

thereafter, ν is formed as

$$\nu = -I_{\psi\psi}^{-1} I_{\psi\phi} \mp I_{\psi\psi}^{-1} \psi$$

where the minus sign is used of maximizing and the plus sign if minimizing.

Once ν has been calculated, min-H on the control variables can begin since the Euler-Lagrange multipliers λ can be formed as

$$\lambda = \lambda_{\phi} + \lambda_{\psi} \nu$$

the variational Hamiltonian H has

$$H = \lambda^T \dot{p}$$

for the first partial of H with respect to χ_p and χ_y as

$$H_a = [\frac{\partial H}{\partial \chi_p} \ \frac{\partial H}{\partial \chi_y}]$$

and the second partial of H with respect to χ_p and χ_y as

$$H_{aa} = \begin{bmatrix} \frac{\partial^2 H}{\partial \chi_p^2} & \frac{\partial^2 H}{\partial \chi_p \partial \chi_y} \\ \frac{\partial^2 H}{\partial \chi_p \partial \chi_y} & \frac{\partial^2 H}{\partial \chi_y^2} \end{bmatrix}$$

Therefore on the second and subsequent iterations W_a^{-1} is taken to be

$$W_a^{-1} = \mp H_{aa}^{-1}$$

where the minus sign is used of maximizing and the plus sign is used if minimizing..

The elements of H_{aa} are

$$\frac{\partial^2 H}{\partial \chi_p^2} = -T_m(\Gamma^w \cos\chi_y \sin\chi_p + \Gamma^u \cos\chi_y \cos\chi_p)$$

$$\frac{\partial H^2}{\partial \chi_p \partial \chi_y} = -T_m(\Gamma^w \sin\chi_y \cos\chi_p - \Gamma^u \sin\chi_y \sin\chi_p)$$

$$\frac{\partial^2 H}{\partial \chi_y^2} = -T_m(\Gamma_1^w \cos\chi_y \sin\chi_p + \Gamma^u \cos\chi_y \cos\chi_p + \Gamma^v \sin\chi_y)$$

where

$$\begin{bmatrix} \Gamma^w \\ \Gamma^u \\ \Gamma^v \end{bmatrix} = \begin{bmatrix} \Gamma^w_\phi \\ \Gamma^u_\phi \\ \Gamma^v_\phi \end{bmatrix} + \begin{bmatrix} \Gamma^w_\psi \\ \Gamma^u_\psi \\ \Gamma^v_\psi \end{bmatrix} \nu$$

If H_{aa} is ill conditioned, χ_p and χ_y satisfying $H_a = 0$ are used to calculate a backup H_{aa} having elements

$$\frac{\partial^2 H}{\partial \chi_p^2} = \mp T_m [(\Gamma^w)^2 + (\Gamma^u)^2] / \sqrt{(\Gamma^w)^2 + (\Gamma^u)^2 + (\Gamma^v)^2}$$

$$\frac{\partial^2 H}{\partial \chi_p \partial \chi_y} = 0$$

$$\frac{\partial^2 H}{\partial \chi_y^2} = \mp T_m \sqrt{(\Gamma^w)^2 + (\Gamma^u)^2 + (\Gamma^v)^2}$$

where the minus sign is used if maximizing and the plus sign is used if minimizing.

If KWTA = 2, H_{aa} is

$$H_{aa} = -T_m (\Gamma^w \sin\chi_p + \Gamma^u \cos\chi_p)$$

with backup

$$H_{aa} = \mp T_m \sqrt{(\Gamma^w)^2 + (\Gamma^u)^2}$$

If a backup H_{aa} matrix is used, the output quantity KAT will be 1; otherwise KAT = 0. KAT should never be 1 on a converged run.

On each iteration a normalized total influence coefficient for each parameter, $\underline{L}^i$ is formed as

$$\underline{L}^i = (L_\phi^i + L_\psi^i \nu) / (\max_{j=1,m} (|L_\phi^i|, |\nu_j L_{\psi_j}^i|))$$

Prior to the fourth iteration the input values of W_i are used in the construction of W_b^{-1}. For the fourth and subsequent iterations each W_i is altered according to the following logic:

$$W_i \text{ unchanged if } |\underline{L}^i| < .005$$

$$W_i \text{ unchanged if } |\underline{L}^i|_{\text{present}} < (1 - \frac{E}{2}) |\underline{L}^i|_{\text{last}}$$

otherwise

$$W_i = 2W_i \text{ if } |\underline{L}^i_{\text{present}} - \underline{L}^i_{\text{Last}}| < |\underline{L}^i|_{\text{present}}$$

$$W_i = W_i/2 \text{ if } |\underline{L}^i_{\text{present}} - \underline{L}^i_{\text{last}}| \geq |\underline{L}^i|_{\text{present}}$$

The W_i are printed out as WIBT between iterations.

This dynamic updating of W_i will generally insure smooth convergence of the parameters. The relative magnitude of the W_i on a converged run can be used as a guide in picking input WIBT.

If there have been at least 3 iterations, if $|dm_o| < 100$ kg, if $|d\dot{\chi}| < .00002$ radians, if all $|d\tau_{1i}| < .5$ seconds and if in addition all $|\underline{L}^i| < .005$, the parameters are considered to be converged and the output quantity BETCON will be T; otherwise BETCON will be F.

The convergence test for the control variables χ_p and χ_y is

$$\begin{bmatrix} |d\chi_p|\ \max \\ |d\chi_y|\ \max \end{bmatrix} = \begin{array}{l}\text{Max} \\ \text{overall} \\ \text{points in} \\ \text{chi-tables}\end{array} \quad |H_{aa}^{-1} H_a| < .005$$

This implies that the max deviation of either χ_p or χ_y from the optimum anywhere along the trajectory is less than .005 radians. The max deviation in χ_p from the optimum is labeled DEL CHIP MAX in the output and the max deviation in χ_y is labeled DEL CHIY MAX.

As soon as $|d\chi_p|_{max} < .005$, $|d\chi_v|_{max} < .005$ and BETCON is T, a run is considered converged. A final forward trajectory is then run at the input print interval, integrated impact (if any) and output tables (if any) are run from this trajectory and then LIFTING ROBOT looks for input for the next case.

REFERENCES

1. Gottlieb, R. G., "Rapid Convergence to Optimum Solutions Using A Min-H Strategy," AIAA Journal, Vol. 5, No. 2, pp. 322-329 (February 1967).

2. Battle, C. T., and Gottlieb, R. G., "Optimization of Multistage Three Dimensional Boost Trajectories, Pt. 1 - Program Formulation," Raytheon Co., Space and Information Systems Div., Report BR-2758-1 (1 May 1964).

3. "Project Apollo Coordinate System Standards" NASA SE 008-001-1 (June 1965).

4. Clarke, V. C., "Constants and Related Data for Use in Trajectory Calculations as Adopted by the ad hoc NASA Standard Constants Committee" JPL TR 32-064 (March 6, 1964).

5. Martin, D. T. Sievers, R. F., O'Brian, R. M., Rice, A. F., "Saturn V Guidance, Navigation, and Targeting," AIAA J. of Spacecraft and Rockets, Vol. 4, No. 7, pp. 891-898, (July 1967).

6. Fischer, I., "An Astrogeodetic World Datum from Geoidal Heights Based on the Flattening F = 1/298, 3," J. of Geophysical Research, Vol. 65, No. 7, (July 1960).

7. Smith, O. E., and Weidner, D. K., "A Reference Atmosphere for Patrick AFB, Florida, Annual (1963 Revision" NASA TM X-53139, (September 23, 1964).

8. Gottlieb, R. G., "Universal Coast Equations and Their Application to Trajectory Optimization" DRC Report AAR-1(U), July 27, 1969.

9. Gottlieb, R. G., Robot-Apollo and AAP Preliminary Mission Profile Optimization Program, NASA CR-61223, May 1968.

10. Goodyear, W. H., "Completely General Closed-Form Solution for Coordinates and Partial Derivatives of the Two-Body Problem," The Astronomical Journal, Vol. 70, No. 3, April 1965.

11. Pitkin, Edward T., "Second Transition Partial Derivatives Via Universal Variables, "The Journal of the Astronautical Sciences, Vol. XIII, No. 5, Sept.-Oct. 1966.

APPENDIX A

UNIVERSAL COAST EQUATIONS AND THEIR APPLICATION TO TRAJECTORY OPTIMIZATION

First, we begin the analysis of the universal coast equations with a statement of plausibility: Since any conic is completely described in terms of six elements, generally c_1, c_3, etc., it is not unreasonable to search for a solution of the equations of motion which has as the six elements the state vector at the initiation of coast, i.e., x_o, $\dot{x}_o$.

A.1 EQUATIONS OF MOTION (SCALAR FORM)

The equations of motion in terms of the radius vector x are:

$$\ddot{x} = -\frac{\mu x}{r^3} \qquad (1)$$

where

$$r = (x \cdot x)^{1/2} \qquad (2)$$

Calculating the second derivative of r from Eq. (2) gives

$$r\dot{r} = \dot{x} \cdot x \qquad (3)$$

and

$$\ddot{r} = \frac{1}{r}\left[\dot{x}^2 - \frac{2\mu}{r} - \dot{r}^2 + \frac{\mu}{r}\right] \qquad (4)$$

Since the energy $\alpha = \dot{x}^2 - \frac{2\mu}{r}$, Eq. (4) becomes

$$\ddot{r} = \frac{1}{r}(\alpha - \dot{r}^2 + \frac{\mu}{r}). \qquad (5)$$

Now make a change in the independent variable defined by the equation

$$dt = r\, d\psi . \tag{6}$$

Consequently,

$$\frac{dr}{d\psi} = \frac{dr}{dt}\frac{dt}{d\psi} = \dot{r}\, r \tag{7}$$

$$\frac{d^2 r}{d\psi^2} = \frac{d(\dot{r}\, r)}{dt}\frac{dt}{d\psi} = \ddot{r}\, r^2 + \dot{r}^2 r \tag{8}$$

If we denote differentiation by ψ with a prime, then, from Eq. (7)

$$\dot{r} = \frac{1}{r}\, r' \tag{9}$$

and hence

$$r'' = \ddot{r}\, r^2 + \frac{(r')^2}{r} \tag{10}$$

By substituting Eqs. (5) and (9) into Eq. (10),

$$r'' = r\alpha - \frac{r'^2}{r} + \mu + \frac{r'^2}{r} \tag{11}$$

or

$$r'' = r\alpha + \mu . \tag{12}$$

Now we seek an expansion of r in terms of ψ about the point r_o as

$$r = r_o + r_o'\psi + \frac{r_o''\psi^2}{2!} + \frac{r_o'''\psi^3}{3!} + \frac{r_o''''\psi^4}{4!} + \ldots \tag{13}$$

We already have r_o'' and the higher derivatives of r_o are given by

$$\left.\begin{aligned} r''' &= \alpha r', \\ r'''' &= \alpha r'' = \alpha^2 r + \alpha\mu, \\ r''''' &= \alpha^2 r', \\ r'''''' &= \alpha^2 r'' = \alpha^3 r + \alpha^2 \mu, \end{aligned}\right\} \tag{14}$$

etc.

Therefore r may be written

$$r = r_0 + r_0'\psi + r_0 \frac{\alpha\psi^2}{2!} + \frac{\mu\psi^2}{2!} + \alpha r_0' \frac{\psi^3}{3!} + r_0 \frac{\alpha^2\psi^4}{4!} + \mu \frac{\alpha\psi^4}{4!} + r_0' \frac{\alpha^2\psi^5}{5!} + \dots \tag{15}$$

The terms in Eq. (15) may be grouped in the following way:

$$\begin{aligned} r &= r_0 \left[1 + \frac{\alpha\psi^2}{2!} + \frac{\alpha^2\psi^4}{4!} + \frac{\alpha^3\psi^6}{6!} + \dots\right] \\ &+ r_0'\left[\psi + \frac{\alpha\psi^3}{3!} + \frac{\alpha^2\psi^5}{5!} + \frac{\alpha^3\psi^7}{7!} + \dots\right] \\ &+ \mu \left[\frac{\psi^2}{2!} + \frac{\alpha\psi^4}{4!} + \frac{\alpha^2\psi^6}{6!} + \dots\right] \end{aligned} \tag{16}$$

Define:

$$\left.\begin{aligned} S_0 &= 1 + \frac{\alpha\psi^2}{2!} + \frac{\alpha^2\psi^4}{4!} + \frac{\alpha^3\psi^6}{6!} + \dots \\ S_1 &= \psi + \frac{\alpha\psi^3}{3!} + \frac{\alpha^2\psi^5}{5!} + \frac{\alpha^3\psi^7}{7!} + \dots \\ S_2 &= \frac{\psi^2}{2!} + \frac{\alpha\psi^4}{4!} + \frac{\alpha^2\psi^6}{6!} + \dots \\ S_3 &= \frac{\psi^3}{3!} + \frac{\alpha\psi^5}{5!} + \frac{\alpha^2\psi^7}{7!} + \dots \end{aligned}\right\} \tag{17}$$

In fact, Eq. (17) can be written in general form as

$$S_n = \sum_{p=0}^{q} \frac{\alpha^p \psi^{2p+n}}{(2p+n)!} , \qquad (18)$$

where q, the number of terms desired, depends on the number of significant figures desired.

Using the definitions in Eq. (17) and defining $\sigma_0 = r_0' = r_0 \dot{r}_0$, Eq. (16) may be written:

$$r = r_0 S_0 + \sigma_0 S_1 + \mu S_2 \qquad (19)$$

Also, since $t = \int r \, d\psi$,

$$t = t_0 + r_0 S_1 + \sigma_0 S_2 + \mu S_3 . \qquad (20)$$

Eq. (20) is solved iteratively for that value of ψ which yields the desired coast time $(t - t_0)$. The fact that $\frac{dt}{d\psi} = r$ is useful here. Only the highest two values of S_n need be calculated as a series since

$$S_n = \alpha S_{n+2} + \frac{\psi^n}{n!} . \qquad (21)$$

A. 2 EQUATIONS OF MOTION (VECTOR FORM)

Now go back to Eq. (2) and proceed to derive an expansion for the state vector $x, \dot{x}$ similar to that derived for r. Again we note that

$$\frac{dx}{d\psi} = \frac{dx}{dt} \frac{dt}{d\psi} = \dot{x} r = x' \qquad (22)$$

and

$$\frac{d^2x}{d\psi^2} = \frac{d}{dt} (\dot{x} r) \frac{dt}{d\psi} = \ddot{x} r^2 + \dot{x} \dot{r} r = x'' . \qquad (23)$$

Therefore,

$$x'' = -\frac{\mu x}{r} + \frac{x' r'}{r}. \tag{24}$$

$$x''' = \frac{r\left(-\mu x' + x'' r' + r'' x'\right) - \left(x' r' - \mu x\right) r'}{r^2}$$

$$x''' = \frac{-\mu r x' + r r' \dfrac{(x' r' - \mu x)}{r} + r x' (r\alpha + \mu) - (x' r' - \mu x) r'}{r^2}$$

$$x''' = \frac{-\mu r x' + x' r' r' - \mu x r' + x' r^2 \alpha + x' \mu r - x' r' r' + \mu x r'}{r^2}$$

$$x''' = x' \alpha , \tag{25}$$

$$\left.\begin{aligned} x'''' &= x'' \alpha = \alpha \frac{x' r'}{r} - \alpha \frac{\mu x}{r} , \\ x''''' &= x' \alpha^2 , \end{aligned}\right\} \tag{26}$$

etc.

Expand x and x' in a Taylor series about the initial point:

$$\left.\begin{aligned} x &= x_0 + x_0' \psi + \frac{x_0'' \psi^2}{2!} + \frac{x_0''' \psi^3}{3!} + \ldots \\ x' &= x_0' + x_0'' \psi + \frac{x_0''' \psi^2}{2!} + \frac{x_0'''' \psi^3}{3!} + \ldots \end{aligned}\right\} \tag{27}$$

Substituting Eqs. (25) and (26) into x we obtain

$$\begin{aligned} x = {} & x_0 + x_0' \psi + \frac{x_0' r_0'}{r_0} \frac{\psi^2}{2!} - \frac{\mu x_0}{r_0} \frac{\psi^2}{2!} \\ & + x_0' \frac{\alpha \psi^3}{3!} + \frac{x_0' r_0'}{r_0} \frac{\alpha \psi^4}{4!} - \frac{\mu x_0}{r_0} \frac{\alpha \psi^4}{4!} + \ldots \end{aligned} \tag{28}$$

Collecting terms, Eq. (28) becomes

$$\begin{aligned} x = {} & x_0 \left[1 - \frac{\mu}{r_0} \left(\frac{\psi^2}{2!} + \frac{\alpha \psi^4}{4!} + \ldots \right) \right] \\ & + x_0' \left[\left(\psi + \alpha \frac{\psi^3}{3!} + \ldots \right) + \frac{r_0'}{r_0} \left(\frac{\psi^2}{2!} + \alpha \frac{\psi^4}{4!} + \ldots \right) \right] \end{aligned} \tag{29}$$

or

$$x = x_0 \left(1 - \frac{\mu S_2}{r_0}\right) + \frac{x_0'}{r_0} (S_1 r_0 + r_0' S_2). \tag{30}$$

Since

$$r_0' = \sigma_0 ,$$

$$\frac{x_0'}{r_0} = \dot{x}_0 ,$$

$$S_1 r_0 + \sigma_0 S_2 = t - t_0 - \mu S_3 ,$$

the equations for the state vector x may be written

$$x = f x_0 + g \dot{x}_0 \tag{31}$$

where

$$\left.\begin{aligned} f &= 1 - \frac{\mu S_2}{r_0} \\ g &= t - t_0 - \mu S_3 . \end{aligned}\right\} \tag{32}$$

and

Substitution of the expressions for the higher derivatives of x into the expansion for x' results in

$$x' = r\dot{x} = \left(- \frac{\mu S_1}{r_0}\right) x_0 + (r - \mu S_2) \dot{x}_0 . \tag{33}$$

Hence

$$\dot{x} = \dot{f} x_0 + \dot{g} \dot{x}_0 \tag{34}$$

where

$$\left.\begin{aligned} \dot{f} &= \frac{df}{d\psi} \frac{d\psi}{dt} = - \frac{\mu S_1}{r r_0} \\ \dot{g} &= \frac{dg}{d\psi} \frac{d\psi}{dt} = 1 - \frac{\mu S_2}{r} . \end{aligned}\right\} \tag{35}$$

We have succeeded in the first of our objectives, namely, to express the state at one time as an explicit function of the state at another time.

A. 3 TRANSFORMATION OF ADJOINTS

We proceed now with the second of our objectives: to develop the partials of the state at one time with respect to the state at another time. This is necessary to transform the adjoints across the coast phase of the trajectory. It is well known that

$$\lambda' \delta x)_{t_a} = \lambda' \delta x)_{t_b} \tag{36}$$

Let

$$\Phi = \frac{\partial x(t_b)}{\partial x(t_a)} \tag{37}$$

Then

$$\delta x(t_b) = \Phi \delta x(t_a) \tag{38}$$

and

$$\lambda'(t_a)\, \delta x(t_a) = \lambda'(t_b)\, \Phi \delta x(t_a) \tag{39}$$

Therefore, since $\delta x(t_a)$ is arbitrary,

$$\lambda'(t_a) = \lambda'(t_b)\, \Phi , \tag{40}$$

and the λ's can be transformed backwards across a coast as soon as Φ is determined. We adopt a convention of Pitkin [2], letting x represent the state at t_o and y the corresponding state at t. Therefore,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} f + \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} g ; \quad \begin{bmatrix} y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \dot{f} + \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \dot{g} \tag{41}$$

$$\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_6} \\ \vdots & & \vdots \\ \frac{\partial y_3}{\partial x_1} & \cdots & \frac{\partial y_3}{\partial x_6} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_6} \end{bmatrix} + \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial x_1} & \cdots & \frac{\partial g}{\partial x_6} \end{bmatrix}$$

$$+ \begin{bmatrix} f & & 0 & \cdot & g & & 0 \\ & f & & \cdot & & g & \\ 0 & & f & \cdot & 0 & & g \end{bmatrix} \tag{42}$$

Similarly,

$$\begin{bmatrix} \frac{\partial y_4}{\partial x_1} & \cdots & \frac{\partial y_4}{\partial x_6} \\ \vdots & & \vdots \\ \frac{\partial y_6}{\partial y_1} & \cdots & \frac{\partial y_6}{\partial x_6} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} \frac{\partial \dot{f}}{\partial x_1} & \cdots & \frac{\partial \dot{f}}{\partial x_6} \end{bmatrix} + \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \begin{bmatrix} \frac{\partial \dot{g}}{\partial x_1} & \cdots & \frac{\partial \dot{g}}{\partial x_6} \end{bmatrix}$$

$$+ \begin{bmatrix} \dot{f} & & 0 & \cdot & \dot{g} & & 0 \\ & \dot{f} & & \cdot & & \dot{g} & \\ 0 & & \dot{f} & \cdot & 0 & & \dot{g} \end{bmatrix} \tag{43}$$

The only difficult part of this is the calculation of the partials of $f, g, \dot{f}, \dot{g}$ with respect to the state X. Taking them one at a time, we begin with f.

$$f = 1 - \frac{\mu S_2}{r_0} = \frac{r_0 - \mu S_2}{r_0} \tag{44}$$

and

$$\frac{\partial f}{\partial X} = \frac{r_0 \left[\frac{\partial r_0}{\partial X} - \mu \frac{\partial S_2}{\partial X}\right] - \left[r_0 - \mu S_2\right] \frac{\partial r_0}{\partial X}}{r_0^2} \tag{45}$$

or

$$\frac{\partial f}{\partial X} = \frac{1}{r_0} \left[(1 - f) \frac{\partial r_0}{\partial X} - \mu \frac{\partial S_2}{\partial X}\right] \tag{46}$$

Also,

$$\frac{\partial S_2}{\partial x} = \frac{\partial S_2}{\partial \psi}\frac{\partial \psi}{\partial x} + \frac{\partial S_2}{\partial \alpha}\frac{\partial \alpha}{\partial x} \tag{47}$$

One can show the following:

$$\left.\begin{aligned}
&\frac{\partial S_n}{\partial \psi} = S_{n-1} \; ; \quad \frac{\partial S_0}{\partial \psi} = \alpha S_1 \\
&\frac{\partial S_n}{\partial \alpha} = \frac{1}{2}\left[\psi S_{n+1} - n S_{n+2}\right] \\
&\frac{\partial S_0}{\partial \alpha} = \left(S_2 + \alpha \frac{\partial S_2}{\partial \alpha}\right)
\end{aligned}\right\} \tag{48}$$

The partials of ψ with respect to x proceed from the fact that the coast time is fixed, i. e.

$$\frac{\partial (t-t_0)}{\partial x}\delta x = 0 = \left[r_0 \frac{\partial S_1}{\partial x} + \sigma_0 \frac{\partial S_2}{\partial x} + \mu \frac{\partial S_3}{\partial x} + S_1 \frac{\partial r_0}{\partial x} + S_2 \frac{\partial \sigma_0}{\partial x}\right]\delta x \, . \tag{49}$$

Therefore,

$$(r_0 S_0 + \sigma_0 S_1 + \mu S_2)\frac{\partial \psi}{\partial x} = -\left[\left(r_0 \frac{\partial S_1}{\partial \alpha} + \sigma_0 \frac{\partial S_2}{\partial \alpha} + \mu \frac{\partial S_3}{\partial \alpha}\right)\frac{\partial \alpha}{\partial x} + S_1 \frac{\partial r_0}{\partial x} + S_2 \frac{\partial \sigma_0}{\partial x}\right]. \tag{50}$$

and $\frac{\partial \psi}{\partial x}$ is given by

$$\frac{\partial \psi}{\partial x} = -\frac{1}{r}\left[\left(r_0 \frac{\partial S_1}{\partial \alpha} + \sigma_0 \frac{\partial S_2}{\partial \alpha} + \mu \frac{\partial S_3}{\partial \alpha}\right)\frac{\partial \alpha}{\partial x} + S_1 \frac{\partial r_0}{\partial x} + S_2 \frac{\partial \sigma_0}{\partial x}\right]. \tag{51}$$

Recalling from Eq. (48) that

$$\left.\begin{aligned}
&\frac{\partial S_1}{\partial \alpha} = \frac{1}{2}\left[\psi S_2 - S_3\right] \\
&\frac{\partial S_2}{\partial \alpha} = \frac{1}{2}\left[\psi S_3 - 2 S_4\right] \\
&\frac{\partial S_3}{\partial \alpha} = \frac{1}{2}\left[\psi S_4 - 3 S_5\right]
\end{aligned}\right\} \tag{52}$$

and defining

$$\frac{\partial T}{\partial \alpha} \equiv r_0 \frac{\partial S_1}{\partial \alpha} + \sigma_0 \frac{\partial S_2}{\partial \alpha} + \mu \frac{\partial S_3}{\partial \alpha}, \tag{53}$$

we substitute in Eq. (51) to obtain

$$\frac{\partial \psi}{\partial X} = -\frac{1}{r} \left[\frac{\partial T}{\partial \alpha} \frac{\partial \alpha}{\partial X} + S_1 \frac{\partial r_0}{\partial X} + S_2 \frac{\partial \sigma_0}{\partial X} \right]. \tag{54}$$

From Eq. (47),

$$\frac{\partial S_2}{\partial X} = -\frac{S_1}{r} \frac{\partial T}{\partial \alpha} \frac{\partial \alpha}{\partial X} - \frac{S_1^2}{r} \frac{\partial r_0}{\partial X} - \frac{S_1 S_2}{r} \frac{\partial \sigma_0}{\partial X} + \frac{\partial S_2}{\partial \alpha} \frac{\partial \alpha}{\partial X}. \tag{55}$$

Hence, from Eq. (46) $\frac{\partial f}{\partial X}$ is given by

$$\frac{\partial f}{\partial X} = \frac{(1-f)}{r_0} \frac{\partial r_0}{\partial X} + \frac{\mu S_1}{r r_0} \frac{\partial T}{\partial \alpha} \frac{\partial \alpha}{\partial X} + \frac{\mu S_1^2}{r r_0} \frac{\partial r_0}{\partial X} + \frac{\mu S_1 S_2}{r r_0} \frac{\partial \sigma_0}{\partial X} - \frac{\mu}{r_0} \frac{\partial S_2}{\partial \alpha} \frac{\partial \alpha}{\partial X} \tag{56}$$

or

$$\frac{\partial f}{\partial X} = \left[\frac{(1-f)}{r_0} - \dot{f} S_1 \right] \frac{\partial r_0}{\partial X} - \dot{f} S_2 \frac{\partial \sigma_0}{\partial X} - \left[\dot{f} \frac{\partial T}{\partial \alpha} + \frac{\mu}{r_0} \frac{\partial S_2}{\partial \alpha} \right] \frac{\partial \alpha}{\partial X} \tag{57}$$

Eq. (57) can, of course, be written:

$$\frac{\partial f}{\partial X} = \frac{\partial f}{\partial r_0} \frac{\partial r_0}{\partial X} + \frac{\partial f}{\partial \sigma_0} \frac{\partial \sigma_0}{\partial X} + \frac{\partial f}{\partial \alpha} \frac{\partial \alpha}{\partial X} \tag{58}$$

where

$$\left.\begin{aligned} \frac{\partial f}{\partial r_0} &= \frac{(1-f)}{r_0} - \dot{f} S_1 \\ \frac{\partial f}{\partial \sigma_0} &= -\dot{f} S_2 \\ \frac{\partial f}{\partial \alpha} &= -\left(\dot{f} \frac{\partial T}{\partial \alpha} + \frac{\mu}{r_0} \frac{\partial S_2}{\partial \alpha} \right) . \end{aligned}\right\} \tag{59}$$

Now we derive the partials of g with respect to X :

$$\frac{\partial g}{\partial X} = -\mu \frac{\partial S_3}{\partial X} = -\mu S_2 \frac{\partial \psi}{\partial X} - \mu \frac{\partial S_3}{\partial \alpha} \frac{\partial \alpha}{\partial X} \qquad (60)$$

$$\frac{\partial g}{\partial X} = \frac{\mu S_2}{r} \frac{\partial \tau}{\partial \alpha} \frac{\partial \alpha}{\partial X} + \frac{\mu S_2}{r} S_1 \frac{\partial r_0}{\partial X} + \frac{\mu S_2^2}{r} \frac{\partial \sigma_0}{\partial X} - \mu \frac{\partial S_3}{\partial \alpha} \frac{\partial \alpha}{\partial X} \qquad (61)$$

Hence

$$\frac{\partial g}{\partial X} = \frac{\partial g}{\partial r_0} \frac{\partial r_0}{\partial X} + \frac{\partial g}{\partial \sigma_0} \frac{\partial \sigma_0}{\partial X} + \frac{\partial g}{\partial \alpha} \frac{\partial \alpha}{\partial X} \qquad (62)$$

where

$$\left.\begin{aligned} \frac{\partial g}{\partial r_0} &= (1-\dot{g}) S_1 \\ \frac{\partial g}{\partial \sigma_0} &= (1-\dot{g}) S_2 \\ \frac{\partial g}{\partial \alpha} &= \left[(1-\dot{g}) \frac{\partial \tau}{\partial \alpha} - \mu \frac{\partial S_3}{\partial \alpha} \right] \end{aligned}\right\} \qquad (63)$$

We derive now the partials of $\dot{f}$ with respect to X .

$$\frac{\partial \dot{f}}{\partial X} = \frac{r r_0 \left(-\mu \frac{\partial S_1}{\partial X}\right) + \mu S_1 \left(r_0 \frac{\partial r}{\partial X} + r \frac{\partial r_0}{\partial X}\right)}{(r r_0)^2} \qquad (64)$$

$$\frac{\partial \dot{f}}{\partial X} = -\frac{1}{r r_0} \left[\mu \frac{\partial S_1}{\partial X} + \dot{f} \left(r \frac{\partial r_0}{\partial X} + r_0 \frac{\partial r}{\partial X} \right) \right] \qquad (65)$$

$$\frac{\partial r}{\partial X} = S_0 \frac{\partial r_0}{\partial X} + S_1 \frac{\partial \sigma_0}{\partial X} + \left(r_0 \frac{\partial S_0}{\partial \alpha} + \sigma_0 \frac{\partial S_1}{\partial \alpha} + \mu \frac{\partial S_2}{\partial \alpha} \right) \frac{\partial \alpha}{\partial X} + \left(\sigma_0 S_0 + (\mu + \alpha r_0) S_1 \right) \frac{\partial \psi}{\partial X} \qquad (66)$$

The only new term here is

$$\frac{\partial S_0}{\partial \alpha} = S_2 + \alpha \frac{\partial S_2}{\partial \alpha} \qquad \text{[See Eq. (48).]}$$

Therefore,

$$\left.\begin{aligned}
\frac{\partial r}{\partial X} &= S_0 \frac{\partial r_0}{\partial X} + S_1 \frac{\partial \sigma_0}{\partial X} + \frac{\partial r}{\partial \alpha} \frac{\partial \alpha}{\partial X} + \frac{\partial r}{\partial \psi} \frac{\partial \psi}{\partial X} \\
&\text{where} \\
\frac{\partial r}{\partial \alpha} &= r_0 S_2 + \sigma_0 \frac{\partial S_1}{\partial \alpha} + (\mu + r_0 \alpha) \frac{\partial S_2}{\partial \alpha} \\
\frac{\partial r}{\partial \psi} &= \sigma_0 S_0 + (\mu + r_0 \alpha) S_1
\end{aligned}\right\} \quad (67)$$

$$\frac{\partial S_1}{\partial X} = S_0 \frac{\partial \psi}{\partial X} + \frac{\partial S_1}{\partial \alpha} \frac{\partial \alpha}{\partial X} \qquad (68)$$

Substituting Eqs. (67) and (68) into Eq. (65) yields

$$\frac{\partial \dot{f}}{\partial X} = -\frac{\mu S_0}{r r_0} \frac{\partial \psi}{\partial X} - \frac{\mu}{r r_0} \frac{\partial S_1}{\partial \alpha} \frac{\partial \alpha}{\partial X} - \frac{\dot{f}}{r_0} \frac{\partial r_0}{\partial X}$$

$$- \frac{\dot{f}}{r} S_0 \frac{\partial r_0}{\partial X} - \frac{\dot{f}}{r} S_1 \frac{\partial \sigma_0}{\partial X} - \frac{\dot{f}}{r} \frac{\partial r}{\partial \alpha} \frac{\partial \alpha}{\partial X} - \frac{\dot{f}}{r} \frac{\partial r}{\partial \psi} \frac{\partial \psi}{\partial X}. \qquad (69)$$

This can be rewritten

$$\frac{\partial \dot{f}}{\partial X} = \frac{1}{r^2} \left(\frac{\mu S_0}{r_0} + \dot{f} \frac{\partial r}{\partial \psi} \right) \frac{\partial T}{\partial \alpha} \frac{\partial \alpha}{\partial X} + \frac{1}{r^2} \left(\frac{\mu S_0}{r_0} + \dot{f} \frac{\partial r}{\partial \psi} \right) S_1 \frac{\partial r_0}{\partial X}$$

$$+ \frac{1}{r^2} \left(\frac{\mu S_0}{r_0} + \dot{f} \frac{\partial r}{\partial \psi} \right) S_2 \frac{\partial \sigma_0}{\partial X} - \left(\frac{\dot{f}}{r_0} + \frac{\dot{f} S_0}{r} \right) \frac{\partial r_0}{\partial X}$$

$$- \frac{\dot{f}}{r} S_1 \frac{\partial \sigma_0}{\partial X} - \frac{1}{r} \left(\frac{\mu}{r_0} \frac{\partial S_1}{\partial \alpha} + \dot{f} \frac{\partial r}{\partial \alpha} \right) \frac{\partial \alpha}{\partial X}. \qquad (70)$$

Hence,

$$\left.\begin{aligned}
\frac{\partial \dot{f}}{\partial X} &= \frac{\partial \dot{f}}{\partial r_0}\,\frac{\partial r_0}{\partial X} + \frac{\partial \dot{f}}{\partial \sigma_0}\,\frac{\partial \sigma_0}{\partial X} + \frac{\partial \dot{f}}{\partial \alpha}\,\frac{\partial \alpha}{\partial X} \\
\text{where} \\
\frac{\partial \dot{f}}{\partial r_0} &= \dot{f}\left[\frac{\partial r}{\partial \psi}\,\frac{S_1}{r^2} - \frac{1}{r_0} - \frac{2S_0}{r}\right] \\
\frac{\partial \dot{f}}{\partial \sigma_0} &= \frac{1}{r}\left[\frac{(1-f)S_0}{r} + \frac{\dot{f}S_2}{r}\,\frac{\partial r}{\partial \psi} - \dot{f}S_1\right] \\
\frac{\partial \dot{f}}{\partial \alpha} &= \frac{1}{r}\left[\frac{1}{r}\left(\frac{\mu S_0}{r_0} + \dot{f}\,\frac{\partial r}{\partial \psi}\right)\frac{\partial r}{\partial \alpha} - \frac{\mu}{r_0}\,\frac{\partial S_1}{\partial \alpha} - \dot{f}\,\frac{\partial r}{\partial \alpha}\right]
\end{aligned}\right\} \quad (71)$$

Now the partials of $\dot{g}$ with respect to X are derived in a similar fashion:

$$\dot{g} = \frac{r - \mu S_2}{r} \qquad \text{(Eq. 35)}$$

$$\frac{\partial \dot{g}}{\partial X} = \frac{r\left(\frac{\partial r}{\partial X} - \mu\,\frac{\partial S_2}{\partial X}\right) - (r - \mu S_2)\,\frac{\partial r}{\partial X}}{r^2} \qquad (72)$$

$$\frac{\partial \dot{g}}{\partial X} = \frac{\mu}{r^2}\left[S_2\,\frac{\partial r}{\partial X} - r\,\frac{\partial S_2}{\partial X}\right] \qquad (73)$$

$$\frac{\partial \dot{g}}{\partial X} = \frac{1}{r}\left[(1-\dot{g})\,\frac{\partial r}{\partial X} - \mu\,\frac{\partial S_2}{\partial X}\right] \qquad (74)$$

$$\begin{aligned}
\frac{\partial \dot{g}}{\partial X} = \frac{1}{r}\Bigg[&(1-\dot{g})\,S_0\,\frac{\partial r_0}{\partial X} + (1-\dot{g})\,S_1\,\frac{\partial \sigma_0}{\partial X} + (1-\dot{g})\,\frac{\partial r}{\partial \alpha}\,\frac{\partial \alpha}{\partial X} \\
&+ (1-\dot{g})\,\frac{\partial r}{\partial \psi}\,\frac{\partial \psi}{\partial X} - \mu S_1\,\frac{\partial \psi}{\partial X} - \mu\,\frac{\partial S_2}{\partial \alpha}\,\frac{\partial \alpha}{\partial X}\Bigg]
\end{aligned} \qquad (75)$$

$$\frac{\partial \dot{g}}{\partial X} = \frac{1}{r}\Bigg[(1-\dot{g})\,S_0\,\frac{\partial r_0}{\partial X} + (1-\dot{g})\,S_1\,\frac{\partial \sigma_0}{\partial X} + \left((1-\dot{g})\,\frac{\partial r}{\partial \alpha} - \mu\,\frac{\partial S_2}{\partial \alpha}\right)\frac{\partial \alpha}{\partial X}$$

$$-\frac{1}{r}\left((1-\dot{g})\,\frac{\partial r}{\partial \psi} - \mu S_1\right)\frac{\partial \tau}{\partial \alpha}\,\frac{\partial \alpha}{\partial X} - \frac{1}{r}\left((1-\dot{g})\,\frac{\partial r}{\partial \psi} - \mu S_1\right)S_1\,\frac{\partial r_0}{\partial X}$$

$$-\frac{1}{r}\left((1-\dot{g})\,\frac{\partial r}{\partial \psi} - \mu S_1\right)S_2\,\frac{\partial \sigma_0}{\partial X}\Bigg] \qquad (76)$$

$$\frac{\partial \dot{g}}{\partial X} = \frac{\partial \dot{g}}{\partial r_0}\,\frac{\partial r_0}{\partial X} + \frac{\partial \dot{g}}{\partial \sigma_0}\,\frac{\partial \sigma_0}{\partial X} + \frac{\partial \dot{g}}{\partial \alpha}\,\frac{\partial \alpha}{\partial X} \qquad (77)$$

where

$$\frac{\partial \dot{g}}{\partial r_0} = \frac{1}{r}\left[(1-\dot{g})\,S_0 - \frac{S_1}{r}\left((1-\dot{g})\,\frac{\partial r}{\partial \psi} - \mu S_1\right)\right]$$

$$\frac{\partial \dot{g}}{\partial \sigma_0} = \frac{1}{r}\left[(1-\dot{g})\,S_1 - \frac{S_2}{r}\left((1-\dot{g})\,\frac{\partial r}{\partial \psi} - \mu S_1\right)\right]$$

$$\frac{\partial \dot{g}}{\partial \alpha} = \frac{1}{r}\left[(1-\dot{g})\,\frac{\partial r}{\partial \alpha} - \mu\,\frac{\partial S_2}{\partial \alpha} - \frac{1}{r}\left((1-\dot{g})\,\frac{\partial r}{\partial \psi} - \mu S_1\right)\frac{\partial \tau}{\partial \alpha}\right]$$

Using Eq. (42) we can define a matrix A as

$$A \equiv \begin{bmatrix} \frac{\partial Y_1}{\partial X_1} & \cdots & \frac{\partial Y_1}{\partial X_6} \\ \vdots & & \vdots \\ \frac{\partial Y_3}{\partial X_1} & & \frac{\partial Y_3}{\partial X_6} \end{bmatrix} = \begin{bmatrix} X_1 & X_4 \\ X_2 & X_5 \\ X_3 & X_6 \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial X_1} & \cdots & \frac{\partial f}{\partial X_6} \\ \frac{\partial g}{\partial X_1} & \cdots & \frac{\partial g}{\partial X_6} \end{bmatrix}$$

$$+ \begin{bmatrix} f & 0 & \cdot & g & 0 \\ & f & \cdot & & g \\ 0 & f & \cdot & 0 & g \end{bmatrix} \qquad (78)$$

$$A = \begin{bmatrix} x_1 & x_4 \\ x_2 & x_5 \\ x_3 & x_6 \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial r_0} & \frac{\partial f}{\partial \sigma_0} & \frac{\partial f}{\partial \alpha} \\ \frac{\partial g}{\partial r_0} & \frac{\partial g}{\partial \sigma_0} & \frac{\partial g}{\partial \alpha} \end{bmatrix} \begin{bmatrix} \frac{\partial r_0}{\partial X} \\ \frac{\partial \sigma_0}{\partial X} \\ \frac{\partial \alpha}{\partial X} \end{bmatrix} + \begin{bmatrix} f & 0 & \vdots & g & 0 \\ & f & \vdots & & g \\ 0 & f & \vdots & 0 & g \end{bmatrix} \tag{79}$$

Note that

$$\left.\begin{aligned} \frac{\partial r_0}{\partial X} &= \begin{bmatrix} \frac{x_1}{r_0} & \frac{x_2}{r_0} & \frac{x_3}{r_0} & 0 & 0 & 0 \end{bmatrix} \\ \frac{\partial \sigma_0}{\partial X} &= \begin{bmatrix} x_4 & x_5 & x_6 & x_1 & x_2 & x_3 \end{bmatrix} \\ \frac{\partial \alpha}{\partial X} &= \begin{bmatrix} \frac{2\mu x_1}{r_0^3} & \frac{2\mu x_2}{r_0^3} & \frac{2\mu x_3}{r_0^3} & 2x_4 & 2x_5 & 2x_6 \end{bmatrix} \end{aligned}\right\} \tag{80}$$

Defining B to be

$$B \equiv \begin{bmatrix} \frac{\partial y_4}{\partial x_1} & \cdots & \frac{\partial y_4}{\partial x_6} \\ \vdots & & \vdots \\ \frac{\partial y_6}{\partial x_1} & \cdots & \frac{\partial y_6}{\partial x_6} \end{bmatrix} \tag{81}$$

we have

$$B = \begin{bmatrix} x_1 & x_4 \\ x_2 & x_5 \\ x_3 & x_6 \end{bmatrix} \begin{bmatrix} \frac{\partial \dot f}{\partial r_0} & \frac{\partial \dot f}{\partial \sigma_0} & \frac{\partial \dot f}{\partial \alpha} \\ \frac{\partial \dot g}{\partial r_0} & \frac{\partial \dot g}{\partial \sigma_0} & \frac{\partial \dot g}{\partial \alpha} \end{bmatrix} \begin{bmatrix} \frac{\partial r_0}{\partial X} \\ \frac{\partial \sigma_0}{\partial X} \\ \frac{\partial \alpha}{\partial X} \end{bmatrix} + \begin{bmatrix} \dot f & 0 & \vdots & \dot g & 0 \\ & \dot f & \vdots & & \dot g \\ 0 & \dot f & \vdots & 0 & \dot g \end{bmatrix} \tag{82}$$

and therefore, the transition matrix Φ defined in equation (37) is written:

$$\Phi = \begin{bmatrix} A \\ \hdashline B \end{bmatrix} \tag{83}$$

If the transition matrix is rewritten in partitioned form as

$$\Phi = \left[\begin{array}{c|c} \phi_{11} & \phi_{12} \\ \hline \phi_{21} & \phi_{22} \end{array}\right] \tag{84}$$

it can be shown that Φ^{-1} is easily found in terms of transposition of its own elements, i. e.

$$\Phi^{-1} = \left[\begin{array}{c|c} \phi_{22}^T & -\phi_{12}^T \\ \hline -\phi_{21}^T & \phi_{11}^T \end{array}\right] . \tag{85}$$

A. 4 APPLICATION TO SPACE TRAJECTORIES - VARIATION OF PARAMETERS

The equations developed thus far can be used to develop a very straight-forward Variation of Parameters integration method. The parameters to be varied of course are the six initial conditions x_0, $\dot{x}_0$. Again letting x represent the state $(x_0, \dot{x}_0)$ at t_0 and y represent the state at time t we can write

$$\frac{dx}{dt} = \frac{\partial x}{\partial t} + \frac{\partial x}{\partial y}\frac{dy}{dt} \tag{86}$$

Treating the time derivatives as the sum of a two body part and a part due to perturbative accelerations, Eq. (86) becomes

$$\frac{dx}{dt} = \left(\frac{\partial x}{\partial t}\right)_{2B} + \frac{\partial x}{\partial y}\left[\left(\frac{dy}{dt}\right)_{2B} + \left(\frac{dy}{dt}\right)_P\right] \tag{87}$$

Since x is constant on a two body orbit the sum of the first two terms is zero, i. e.

$$\left(\frac{\partial x}{\partial t}\right)_{2B} + \frac{\partial x}{\partial y}\left(\frac{dy}{dt}\right)_{2B} = 0 \tag{88}$$

Now, an osculating two body orbit will match position and velocity with the actual orbit, and since the first three elements of $\frac{dy}{dt}$ are $\dot{y}$, equation (87) becomes

$$\frac{dx}{dt} = \frac{\partial x}{\partial y}\begin{bmatrix} 0 \\ \cdots \\ a_p \end{bmatrix} \tag{89}$$

where a_p are the perturbative accelerations.

In terms of the notation of Eq. (85), Eq. (89) may be written

$$\frac{dx}{dt} = \begin{bmatrix} -\phi_{12}^T \\ \cdots \\ \phi_{11}^T \end{bmatrix} a_p \tag{90}$$

Since ϕ_{12} and ϕ_{11} are themselves functions of x and t and since a_p is also a function of x and t through the transformation given by Eq. (41), Eq. (90) is a well defined differential equation for x.

APPENDIX B

FIRST PARTIAL DERIVATIVES OF SPHERICAL-PLUMBLINE TRANSFORMATIONS

The matrix N is defined to be the matrix of the first partial derivatives

$$N = \frac{\partial S}{\partial P}$$

where S here is the 6x1 vector of spherical components

$$S = \begin{bmatrix} w_s \\ u_s \\ v_s \\ \phi \\ r \\ \theta \end{bmatrix}$$

and P is the 6x1 vector of plumbline components

$$P = \begin{bmatrix} w \\ u \\ v \\ x \\ y \\ z \end{bmatrix}$$

The matrix N may be partitioned into four 3 x 3 submatrices, i. e.,

$$N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$$

These submatrices are defined by the equations:

$$N_{11} = \begin{bmatrix} \frac{\partial w_s}{\partial w} & \frac{\partial w_s}{\partial u} & \frac{\partial w_s}{\partial v} \\ \frac{\partial u_s}{\partial w} & \frac{\partial u_s}{\partial u} & \frac{\partial u_s}{\partial v} \\ \frac{\partial v_s}{\partial w} & \frac{\partial v_s}{\partial u} & \frac{\partial v_s}{\partial v} \end{bmatrix} = D^T$$

$$N_{12} = \begin{bmatrix} \frac{\partial w_s}{\partial x} & \frac{\partial w_s}{\partial y} & \frac{\partial w_s}{\partial z} \\ \frac{\partial u_s}{\partial x} & \frac{\partial u_s}{\partial y} & \frac{\partial u_s}{\partial z} \\ \frac{\partial v_s}{\partial x} & \frac{\partial v_s}{\partial y} & \frac{\partial v_s}{\partial z} \end{bmatrix}$$

where

$$\frac{\partial w_s}{\partial x} = (a_{32}\, u - a_{22}\, v - w_s(d_{13} \cos\theta + d_{12}\sin\theta))/(r \sin\theta)$$

$$\frac{\partial w_s}{\partial y} = (a_{12}\, v - a_{32}\, w - w_s(d_{23} \cos\theta + d_{22}\sin\theta))/(r \sin\theta)$$

$$\frac{\partial w_s}{\partial z} = (a_{22}\, w - a_{12}\, u - w_s(d_{33} \cos\theta + d_{32}\sin\theta))/(r \sin\theta)$$

$$\frac{\partial u_s}{\partial z} = (w - d_{12}\, u_s)/r$$

$$\frac{\partial u_s}{\partial x} = (u - d_{22}\, u_s)/r$$

$$\frac{\partial u_s}{\partial z} = (v - d_{32}\, u_s)/r$$

$$\frac{\partial v_s}{\partial x} = (d_{11}\, w_s\, \mathrm{ctn}\theta - d_{13}\, u_s)/r$$

$$\frac{\partial v_s}{\partial y} = (d_{21}\, w_s\, \mathrm{ctn}\theta - d_{23}\, u_s)/r$$

$$\frac{\partial v_s}{\partial z} = (d_{31}\, w_s\, \mathrm{ctn}\theta - d_{33}\, u_s)/r$$

$$N_{21} = \begin{bmatrix} \frac{\partial \phi}{\partial w} & \frac{\partial \phi}{\partial u} & \frac{\partial \phi}{\partial v} \\ \frac{\partial r}{\partial w} & \frac{\partial r}{\partial u} & \frac{\partial r}{\partial v} \\ \frac{\partial \theta}{\partial w} & \frac{\partial \theta}{\partial u} & \frac{\partial \theta}{\partial v} \end{bmatrix} = 0$$

$$N_{22} = \begin{bmatrix} \frac{\partial \phi}{\partial x} & \frac{\partial \phi}{\partial y} & \frac{\partial \phi}{\partial z} \\ \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{d_{11}}{r \sin\theta} & \frac{d_{21}}{r \sin\theta} & \frac{d_{31}}{r \sin\theta} \\ d_{12} & d_{22} & d_{32} \\ \frac{d_{13}}{r} & \frac{d_{23}}{r} & \frac{d_{33}}{r} \end{bmatrix}$$

APPENDIX C

FIRST PARTIAL DERIVATIVES OF GRAVITATIONAL ACCELERATION WITH RESPECT TO PLUMBLINE POSITION COORDINATES

The matrix J is defined to be the matrix of first partial derivatives of the gravitational acceleration vector in the plumbline system with respect to the plumbline position coordinates. This matrix is used in the gravity related terms of the adjoint (Euler-Lagrange) equations.

$$J = \begin{bmatrix} \frac{\partial g_x}{\partial x} & \frac{\partial g_x}{\partial y} & \frac{\partial g_x}{\partial z} \\ \frac{\partial g_y}{\partial x} & \frac{\partial g_y}{\partial y} & \frac{\partial g_y}{\partial x} \\ \frac{\partial g_z}{\partial x} & \frac{\partial g_z}{\partial y} & \frac{\partial g_z}{\partial z} \end{bmatrix} = \begin{bmatrix} g_{xx} & g_{xy} & g_{xz} \\ g_{yx} & g_{yy} & g_{yz} \\ g_{zx} & g_{zy} & g_{zz} \end{bmatrix}$$

$$J = G_{11} I + \begin{bmatrix} x & a_{12} \\ y & a_{22} \\ z & a_{32} \end{bmatrix} \begin{bmatrix} G_{22} & G_{23} \\ G_{32} & G_{33} \end{bmatrix} \cdot \begin{bmatrix} x & y & z \\ a_{12} & a_{22} & a_{32} \end{bmatrix}$$

G_{11} is defined in Section 3.1, I is a 3 x 3 identity matrix, the a_{ij} are elements of the A matrix, and

3

$$G_{22} = \frac{1}{r}\left(\frac{\partial G_{11}}{\partial r} + \frac{\text{ctn}\theta}{r}\frac{\partial G_{11}}{\partial \theta}\right)$$

$$= -\frac{3}{r^2}\left[G_{11} - \frac{\mu_e}{r^3}\left(CJ\left(\frac{R_e}{r}\right)^2\left(\frac{2}{3} - \frac{20}{3}\cos^2\theta\right) + H\left(\frac{R_e}{r}\right)^3(4 - 14\cos^2\theta)\cos\theta\right.\right.$$

$$\left.\left. + DJ\left(\frac{R_e}{r}\right)^4\left(\frac{4}{7} - (12 - 24\cos^2\theta)\cos^2\theta\right)\right)\right]$$

$$G_{23} = G_{32} = -\frac{1}{r\sin\theta}\frac{\partial G_{11}}{\partial\theta} = -\frac{1}{r}\left(\frac{\partial G_{TO}}{\partial r} + \frac{\text{ctn}\theta}{r}\frac{\partial G_{TO}}{\partial\theta}\right)$$

$$= \frac{\mu_e}{r^4}\left[10CJ\left(\frac{R_e}{r}\right)^2\cos\theta - H\left(\frac{R_e}{r}\right)^3(3 - 21\cos^2\theta)\right.$$

$$\left. + DJ\left(\frac{R_e}{r}\right)^4(12 - 36\cos^2\theta)\cos\theta\right]$$

$$G_{33} = \frac{1}{r\sin\theta}\frac{\partial G_{TO}}{\partial\theta} .$$

$$= -\frac{\mu_e}{r^3}\left[2CJ\left(\frac{R_e}{r}\right)^2 + 6H\left(\frac{R_e}{r}\right)^3\cos\theta + \frac{12}{7}DJ\left(\frac{R_e}{r}\right)^4(1 - 7\cos^2\theta)\right]$$

The fact that J is symmetric can be anticipated, since J is also the matrix of second partial derivatives of the gravitational potential function $U(r, \theta)$ with respect to the plumbline position coordinates.

In the event that a spherical earth is being simulated J reduces to

$$J = G_{11} I + \begin{bmatrix} x \\ y \\ z \end{bmatrix} G_{22} [x\ y\ z]$$

with

$$G_{22} = -\frac{3}{r^2} G_{11}$$

since $G_{23} = G_{32} = G_{33} = 0.$

APPENDIX D

RELATION OF C_A AND C_N TO C_L AND C_D

The aerodynamic formulation in Section 5 is predicated upon the assumption that the lift and drag coefficients may be approximated by

$$C_L = C_{L_\alpha} \alpha$$

$$C_D = C_{D_o} + C_{D_a} \alpha^2$$

and that

$$\alpha \approx \sin\alpha$$

$$\alpha^2 \approx 2(1 - \cos\alpha)$$

Using these assumptions C_L and C_D become

$$C_L = C_{L_\alpha} \sin\alpha$$

$$C_D = C_{D_o} + 2C_{D_a}(1 - \cos\alpha)$$

Since

$$C_A = C_D \cos\alpha - C_L \sin\alpha$$

$$C_N = C_L \cos\alpha + C_D \sin\alpha$$

there results

$$C_A = (C_{D_o} + 2C_{D_a})\cos\alpha + (2C_{D_a} - C_{L_\alpha})\sin^2\alpha - 2C_{D_a}$$

$$C_N = (C_{D_o} + 2C_{D_a})\sin\alpha - (2C_{D_a} - C_{L_\alpha})\sin\alpha\cos\alpha$$

Therefore the aerodynamic coefficients a, b, c of Section 5 are

$$a = C_{D_o} + 2C_{D_a}$$

$$b = 2C_{D_a} - C_{L_\alpha}$$

$$c = -2C_{D_a}$$

D.1 REDUCTION OF DATA

Given $C_A(\alpha, M)$ and $C_N(\alpha, M)$ it is easy to construct

$$C_L(\alpha, M) = C_N\cos\alpha - C_A\sin\alpha$$

$$C_D(\alpha, M) = C_A\cos\alpha + C_N\sin\alpha$$

therefore, C_L and C_D may be thought of as perfectly general.

It is not unreasonable to expect to be able to approximate a given set of $C_L(\alpha, M)$, $C_D(\alpha, M)$ data as

$$C_L = C_{L_o} + C_{L_\alpha} \alpha*$$

$$C_D = C_{D_o} + C_{D_L} C_L^2$$

where
$$C_{D_L} \equiv \frac{dC_D}{dC_L^{\,2}}$$

and where $\alpha*$ denotes the reference or data-base angle of attack.

Since the angle of attack is the angle between some reference direction and the velocity vector, define a <u>new</u> reference direction by the angle of attack, α, related to the one on which the data is based by

$$\alpha = \alpha* + \delta$$

where δ is a bias angle.

In terms of α, C_L is

$$C_L = C_{L_o} + C_{L_\alpha}(\alpha - \delta)$$

In order that $C_L = 0$ when $\alpha = 0$ it is necessary that

$$\delta = C_{L_o}/C_{L_\alpha}$$

In general δ will not be exactly constant with respect to Mach number, however, a good least squares constant bias may be calculated by choosing a set of Mach numbers, M_i, and then forming

$$\delta = \sum_i C_{L_\alpha}(M_i) \cdot C_{L_o}(M_i) / \sum_i C_{L_\alpha}(M_i)$$

In terms of α, C_D becomes

$$C_D = C_{D_o} + C_{D_L}(C_{L_o} + C_{L_\alpha}(\alpha - \delta))^2$$

$$= C_{D_o} + C_{D_L}((C_{L_o} - C_{L_\alpha}\delta) + C_{L_\alpha}\alpha)^2$$

Recall that δ was chosen such that

$$C_{L_o} - C_{L_\alpha}\delta \approx 0$$

Therefore to good approximation

$$C_D = C_{D_o} + C_{D_L} C_{L_\alpha}^2 \alpha^2$$

or

$$C_D = C_{D_o} + C_{D_a} \alpha^2$$

where

$$C_{D_a} \equiv C_{D_L} C_{L_\alpha}^2$$

which is consistent with the original assumptions about the form of C_D.

APPENDIX E

RADIUS OF APOGEE AND RADIUS OF PERIGEE CONSTRAINTS

The radius of apogee and radius of perigee constraints are treated in the APPG package which includes subroutines APPG, ADDR, AGEØ, FØRKM and SYSDER. The approach used is to compute the radii of apogee and perigee r_a and r_p, respectively, and the associated adjoint vectors $\underline{\lambda}_{\psi_{r_a}}$ and $\underline{\lambda}_{\psi_{r_p}}$ through a numerical integration of the orbital equations of motion and the associated adjoint equations. The orbital Euler-Lagrange equations are

$$\dot{\underline{x}} = \underline{f}(\underline{x})$$

$$\dot{\underline{\lambda}} = -\left(\frac{\partial H}{\partial \underline{x}}\right)^T$$

where H is the variational Hamiltonian. These equations take the form

$$\begin{aligned}
\dot{w} &= G_x & \dot{\lambda}_1 &= -\lambda_4 \\
\dot{u} &= G_y & \dot{\lambda}_2 &= -\lambda_5 \\
\dot{v} &= G_z & \dot{\lambda}_3 &= -\lambda_6 \\
\dot{x} &= w & \dot{\lambda}_4 &= -\lambda_1 G_{xx} - \lambda_2 G_{yx} - \lambda_3 G_{zx} \\
\dot{y} &= u & \dot{\lambda}_5 &= -\lambda_1 G_{xy} - \lambda_2 G_{yy} - \lambda_3 G_{zy} \\
\dot{z} &= v & \dot{\lambda}_6 &= -\lambda_1 G_{xz} - \lambda_2 G_{yz} - \lambda_3 G_{zz}
\end{aligned}$$

where (G_x, G_y, G_z) is the gradient and G_{xx}, G_{yy}, etc. are the second partials of the gravity potential. Boundary conditions are $\underline{x}(t_o)$ the state at injection given and

$$\underline{\lambda}^T_{\psi_{r_a}}(t_a) = \left.\frac{\partial r}{\partial \underline{x}}\right|_{t=t_a}$$

$$\underline{\lambda}^T_{\psi_{r_p}}(t_p) = \left.\frac{\partial r}{\partial \underline{x}}\right|_{t=t_p}$$

where t_a and t_p represent the times at which the orbit passes through its apogee and perigee, respectively.

The procedure used in the numerical integration of the orbital equations is to first integrate the system equations forward in time from $\underline{x}(t_o)$ to some time t_1 at which time r passes through its first extremum. The stopping condition used is

$$\Omega(t_1) = \underline{r} \cdot \dot{\underline{r}}\Big|_{t=t_1} = [xw + yu + zv]_{t=t_1} = 0$$

The adjoint equations are then integrated backward in time from t_1 to t_o from the terminal conditions

$$\underline{\lambda}^T(t_1) = \left.\frac{\partial r}{\partial \underline{x}}\right|_{t=t_1}$$

Define $\underline{\lambda}_{\psi_{r_1}}$ as the solution at time t_o of this backward integration.

The second extrêmum is found by integrating the system equations backward in time from $\underline{x}(t_o)$ to some time t_2 at which time r passes through its first extremum. The adjoint equations are then integrated forward in time from the initial conditions

$$\underline{\lambda}^T(t_2) = \left.\frac{\partial r}{\partial \underline{x}}\right|_{t=t_2}$$

Let $\underline{\lambda}_{\psi_{r_2}}$ be the solution of this forward integration of the adjoint equations at time t_o. Then a simple comparison of $r(t_1)$ and $r(t_2)$ can be used to set r_a, r_p, $\underline{\lambda}_{\psi_{r_a}}$ and $\underline{\lambda}_{\psi_{r_p}}$. If $r(t_2)$ is greater than $r(t_1)$

$$r_p = r(t_1) \qquad \underline{\lambda}_{\psi_{r_p}} = \underline{\lambda}_{\psi_{r_1}}$$

$$r_a = r(t_2) \qquad \underline{\lambda}_{\psi_{r_a}} = \underline{\lambda}_{\psi_{r_2}}$$

Otherwise

$$r_p = r(t_2) \qquad \underline{\lambda}_{\psi_{r_p}} = \underline{\lambda}_{\psi_{r_2}}$$

$$r_a = r(t_1) \qquad \underline{\lambda}_{\psi_{r_a}} = \underline{\lambda}_{\psi_{r_1}}$$

The advantage of using this approach to determine r_a, r_p, $\lambda_{\psi_{r_a}}$ and $\lambda_{\psi_{r_p}}$ is that it does not require analytic expressions for r_a, r_p, etc., in terms of the state at injection. In general derivation of analytic expressions for r_a, r_p, etc., for an oblate gravity field is quite difficult and some sort of approximation is generally used. By computing these quantities through a numerical integration of the orbit equations, this approximation is avoided.

The only limit on the accuracy for this approach is the numerical integration error. Numerical integration of the orbit equations in APPG employs a fourth order Runge Kutta differential equation solver with a fixed step size of 20 sec for the system equations and 40 sec for the adjoint equations. Numerical accuracy appears to be adequate but can be improved if desired by decreasing the step size DT.

APPENDIX F
THE IMPACT POINT PENALTY

The impact trace of a launch vehicle will be forced to avoid certain regions by utilizing an impact trace penalty function. A new state variable is therefore defined to be

$$\dot{x}_{n+1} = \sum_{i=1}^{k} C_i \, e^{-Z_i^T B_i Z_i}$$

where C_i are weighting factors, B_i is a 2x2 symmetric matrix associated with each point (θ_i, θ_i) that shapes the exponential function, and

$$Z_i = \begin{bmatrix} \theta_p - \theta_i \\ \phi_p - \phi_i \end{bmatrix}$$

By properly choosing C_i, B_i, θ_i and ϕ_i the impact trace penalty function can be designed to cover forbidden zones in the θ, ϕ, (latitude, longitude) space. By constrainting the termal value of x_{n+1} the impact trace can be made to avoid the "peaks" in $\dot{x}_{n+1}$. An efficient algorithm for calculating the latitude and longitude at impact which is consistent with the partial derivatives needed for the adjoints is developed below.

In order to avoid iterating on the change in eccentric anomaly necessary to give the proper coast time, it will be computed analytically and then used in the Goodyear coast equations.

Figure 9 shows a schematic of the coast to impact.

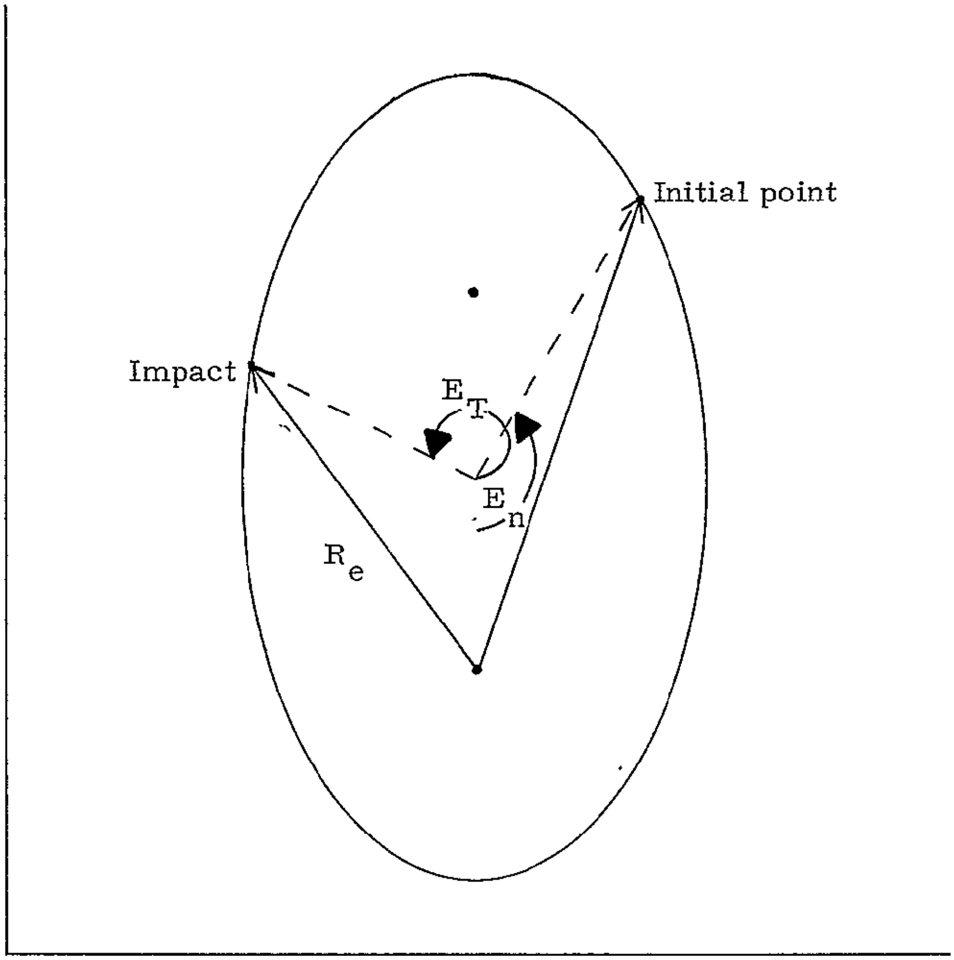

FIGURE 9 SCHEMATIC OF COAST TO IMPACT

Obviously,

$$E_n = \cos^{-1}\left(\frac{a - r}{ae}\right)$$

and

$$E_T = 2\pi - \cos^{-1}\left(\frac{a - R_e}{ae}\right)$$

It is assumed here that only coasts initiating with positive flight path angles are of interest.

The charge in eccentric anomaly is

$$\Delta E = 2\pi - (\theta_1 + \theta_2)$$

where

$$\theta_1 \triangleq \cos^{-1}\left(\frac{a - R_e}{ae}\right)$$

$$\theta_2 \triangleq \cos^{-1}\left(\frac{a - r}{ae}\right)$$

Thus

$$\cos\Delta E = \cos(\theta_1 + \theta_2)$$

$$\sin\Delta E = -\sin(\theta_1 + \theta_2)$$

Since both θ_1 and θ_2 will be calculated to be less than pi, the sine will be positive for both angles; therefore we can calculate

$$\cos\Delta E = \cos\theta_1 \cos\theta_2 - [(1-\cos^2\theta_1)(1-\cos^2\theta_2)]^{1/2}$$

$$\sin\Delta E = -[\cos\theta_2 \sqrt{1-\cos^2\theta_1} + \cos\theta_1\sqrt{1-\cos^2\theta_2}]$$

with

$$\cos\theta_1 = -\frac{a - R_e}{ae}$$

$$\cos\theta_2 = \frac{a-r}{ae}$$

The correspondence between ΔE and Goodyear's universal variable, ψ, is known to be

$$\psi = (a/\mu)^{1/2}\,\Delta E$$

Also,

$$S_o = C_o = \cos\Delta E$$

$$S_1 = \psi C_1 = (a/\mu)^{1/2} \sin\Delta E$$

$$S_2 = \psi^2 C_2 = (a/\mu)(1 - S_o)$$

$$S_3 = \psi^3 C_3 = (a/\mu)(\psi - S_1)$$

$$S_4 = \psi^4 C_4 = (a/\mu)(\frac{\psi^2}{2!} - S_2)$$

$$S_5 = \psi^5 C_5 = (a/\mu)(\frac{\psi^3}{3!} - S_3)$$

Then defining

$$\sigma_o = (\bar{r} \cdot \bar{v})_{initial}$$

along with

$$F = 1 - \mu S_2/r$$

$$g = rS_1 + \sigma_o S_2$$

the position coordinates at impact are

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = f \begin{pmatrix} x \\ y \\ z \end{pmatrix} + g \begin{pmatrix} w \\ u \\ v \end{pmatrix}$$

The time to impact is

$$\Delta t = rS_1 + \sigma_o S_2 + \mu S_3$$

Using the A matrix of Section 2,

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = A^T \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

the latitude at impact is

$$\theta_{\ell_p} = \sin^{-1}(y/R_e)$$

and the earth relative longitude is

$$\phi_{\ell_p} = \tan^{-1}(x/z) - \Omega_e(\Delta t + DTZ + T) - ALØNGØ$$

Using these values, $\dot{x}_{n+1}$ can be calculated.

During the adjoint run, the partial derivatives of $\dot{x}_{n+1}$ with respect to the state along the trajectory must be calculated.

The calculation proceeds as follows:

$$\frac{\partial \dot{x}_{n+1}}{\partial \theta_p} = -\Sigma C_i e^{-z_i^T B_i z_i}(2b_{i11}(\theta_p - \theta_i) + b_{i12}(\phi_p - \phi_i)$$

$$\frac{\partial \dot{x}_{n+1}}{\partial \phi_p} = -\Sigma C_i e^{-z_i^T B_i z_i}(b_{i12}(\theta_p - \theta_i) + 2b_{i22}(\dot{\phi}_p - \phi_i))$$

$$dr = 0 = \frac{\partial r}{\partial x_p} \delta x_p + \dot{r}\, dt_f$$

$$\therefore \quad dt_f = \frac{-1}{\dot{r}} \frac{\partial r}{\partial x_p} \delta x_p$$

where

$$\dot{r} = r_p \cdot \dot{r}_p / R_e$$

$$\delta x_p = \left[\frac{\partial x_p}{\partial x} \;\vdots\; \frac{\partial x_p}{\partial v} \right] \begin{pmatrix} \delta x \\ \delta v \end{pmatrix}$$

$$d\theta_p = \frac{\partial \theta_p}{\partial x_p} \delta x_p + \frac{\partial \theta_p}{\partial x_p} \dot{x}_p \, dt_f$$

$$d\phi_p = \frac{\partial \phi_p}{\partial x_p} \delta x_p + \left(\frac{\partial \theta_p}{\partial x_p} \dot{x}_p - \Omega_e \right) dt_f$$

which can be written

$$d\theta_p = \frac{\partial \theta_p}{\partial x_p} \left[I - \frac{1}{\dot{r}} \dot{x}_p \frac{\partial r}{\partial x_p} \right] \delta x_p$$

$$d\phi_p = \left\{ \frac{\partial \phi_p}{\partial x_p} \left[I - \frac{1}{\dot{r}} \dot{x}_p \frac{\partial r}{\partial x_p} \right] + \frac{\Omega_e}{\dot{r}} \frac{\partial r}{\partial x_p} \right\} \delta x_p$$

Note that $\dot{x}_p \dfrac{\partial r}{\partial x_p}$ is a 3x3 outer product

$$\begin{pmatrix} \dfrac{\partial \theta_p}{\partial x_p} \\ \\ \dfrac{\partial \phi_p}{\partial x_p} \end{pmatrix} = \begin{bmatrix} 0 & \dfrac{1}{R_e \cos \theta_p} & 0 \\ \\ \dfrac{z}{x^2 + z^2} & 0 & \dfrac{-x}{x^2 + z^2} \end{bmatrix} A^T$$

$$d\dot{x}_{n+1} = \frac{\partial \dot{x}_{n+1}}{\partial \theta_p} d\theta_p + \frac{\partial \dot{x}_{n+1}}{\partial \phi_p} d\phi_p$$

or

$$d\dot{x}_{n+1} = \left\{ \left(\frac{\partial \dot{x}_{n+1}}{\partial \phi_p} \frac{z}{x^2+z^2} \quad \frac{\partial \dot{x}_{n+1}}{\partial \theta_p} \frac{1}{R_e \cos\theta_p} \quad \frac{-\partial \dot{x}_{n+1}}{\partial \phi_p} \frac{x}{x^2+z^2} \right) A^T \Gamma \right.$$

$$\left. + \frac{\partial \dot{x}_{n+1}}{\partial \phi_p} \frac{\Omega_e}{\dot{r}} \frac{\partial r}{\partial x_p} \right\} \delta x_p$$

where

$$\Gamma \triangleq I_3 - \frac{1}{\dot{r}} \dot{x}_p \frac{\partial r}{\partial x_p}$$

Some simplifications in the numerical work required can be made. Defining

$$a \triangleq \frac{\partial \dot{x}_{n+1}}{\partial \phi_p} \cdot \frac{1}{x^2+z^2}$$

$$b \triangleq \frac{\partial \dot{x}_{n+1}}{\partial x_p} \cdot \frac{1}{R_e \cos\theta_p}$$

This gives

$$\left[\frac{\partial \dot{x}_{n+1}}{\partial x_p}\right] = (PR_1 \; PR_2 \; PR_3)\Gamma + \frac{\partial \dot{x}_{n+1}}{\partial \phi_p} \frac{\Omega_e}{r_p \cdot v_p} \Gamma_p^T$$

where

$$PR_1 = (Z\,a_{11} - X\,a_{13})\,a + b\,a_{12}$$

$$PR_2 = (Z\,a_{21} - X\,a_{23})\,a + b\,a_{22}$$

$$PR_3 = (Z\,a_{31} - X\,a_{33})\,a + b\,a_{32}$$

Then defining

$$c \triangleq (PR_1\,\dot{x}_p + PR_2\,\dot{y}_p + PR_3\,\dot{z}_p)/(r_p \cdot v_p)$$

$$d \triangleq \frac{\partial \dot{x}_{n+1}}{\partial \phi_p} \quad \frac{\Omega_e}{r_p \cdot v_p}$$

gives

$$\frac{\partial \dot{x}_{n+1}}{\partial x_p} = (PR_1 + (d-c)x_p \;\; PR_2 + (d-c)y_p \;\; PR_3 + \overline{(d-c)z_p)}$$

Finally

$$\left[\frac{\partial \dot{x}_{n+1}}{\partial x} \;\Big|\; \frac{\partial \dot{x}_{n+1}}{\partial v}\right] = \frac{\partial \dot{x}_{n+1}}{\partial x_p} \left[\frac{\partial x_p}{\partial x} \;\Big|\; \frac{\partial x_p}{\partial v}\right]$$

where the partials of state at impact with respect to state along trajectory will be calculated using Goodyear's partials.

The velocity components at impact are calculated from

$$\dot{f} \triangleq - \frac{\mu S_1}{R_e r}$$

$$\dot{g} \triangleq 1 - \frac{\mu S_2}{R_e}$$

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{z}_p \end{pmatrix} = \dot{f} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \dot{g} \begin{pmatrix} w \\ u \\ v \end{pmatrix}$$

APPENDIX G

INPUT DESCRIPTION AND EXAMPLE PROBLEM

The user of the LIFTING ROBOT program will find it helpful to sketch a thrust profile before setting up input for the problem he wishes simulated. Sketched in Fig. 9 is an 8 thrust event representation of a three stage Saturn V thrust history. Vertical lines and horizontal lines will be referred to as "pickets" and "spaces", respectively. The "picket" numbers in Fig.10 are circled. Note, there is always one more picket than spaces. A thrust event must be defined every time there is a discontinuity in the total thrust. Dashed vertical lines represent miscellaneous weight drops. Spaces are thrust duration times and are denoted TAUT. The elapsed time between the Jth miscellaneous weight drop event (dashed vertical lines) and some thrust event picket is denoted TAUW(J). The particular thrust event picket to use is denoted NØWD(J). LIFTING ROBOT drops the atmosphere at the IWDCHI th miscellaneous weight drop event. Therefore, a miscellaneous weight must be dropped where the atmosphere is to end even if it is a zero (0) weight drop. Min-H optimization begins at either t_{χ} or t_Q whichever occurs first.

The LIFTING ROBOT program controls vehicle flight by looking up χ_p and χ_y as a function of time out of control tables. The Min-H steepest ascent process adjust these tabular points until they take on optimal values. A "control table" consists therefore of three tabular arrays: time, χ_p, χ_y. LIFTING ROBOT contains four control tables, each containing a maximum of 49 points. In order to provide generality

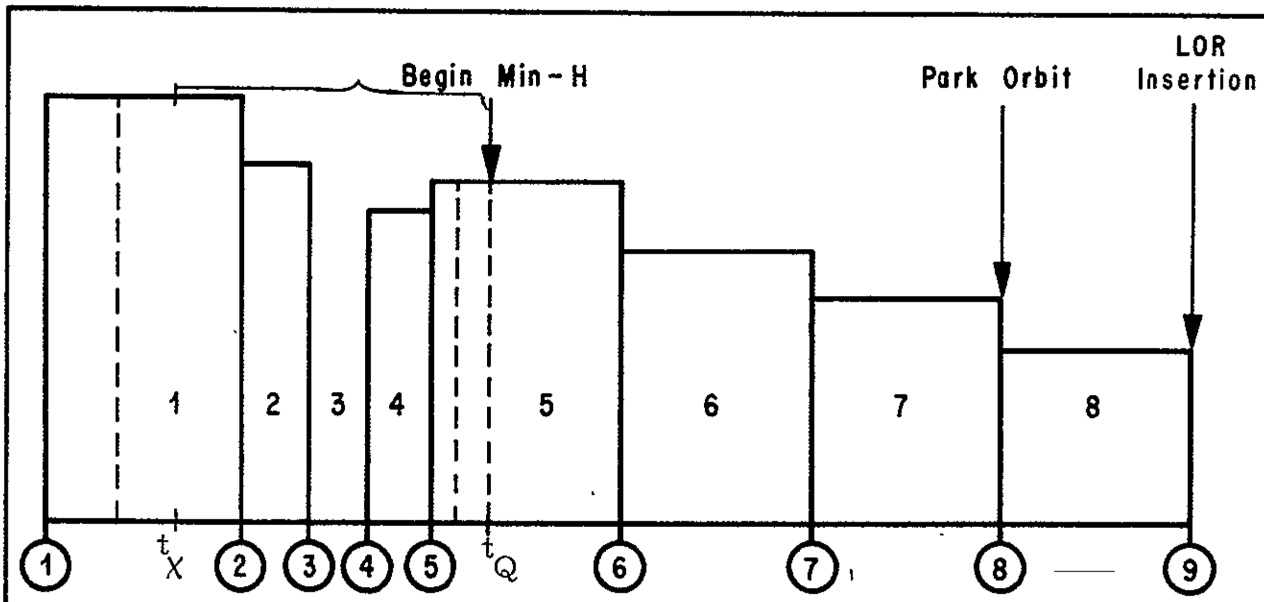

FIG. 10. SATURN V THRUST PROFILE

for the user, the Jth control table begins at the NBGCT(J) th picket, ends at the NENDCT(J) th picket and has a maximum of $NP(J) \leq 49$ points. NP should be odd for all tables in use and zero for all others. Control tables should not extend over coasts or over an intermediate constraint point. If Min-H is to begin in the middle of a thrust event, NBGCT(1) should be set to the picket at the beginning of the thrust event.

NAMELIST INPUT DESCRIPTION

INPUT SYMBOL*	SIZE	EXPLANATION	PRESET VALUE	UNITS
HEAD	(15)	Identification for print out (60 characters)		
TZERØ		Initial time		sec
TLIFT		End of lift-off; beginning of tilt		sec
TTILT		End of tilt		sec
TCHFRZ		Begin Min-H		sec
DTZ		Time from GRR to Lift-off		sec
FLBS	(15)	Thrust per engine/thrust event	0.	lbs
TNE	(4, 15)	Number of engines/thrust event Four numbers for each thrust event: the number of inboard engines, their cant angle (deg), the number of outboard engines, their cant angle. (deg).	0.	
WDØT	(15)	Flow rate per engine/thrust event	0.	lbs/sec
CWØT	(15)	Critical flow rate per engine/thrust event		lbs/sec

*$ INPUT in Col. 2. All Data begins in Col. 2

INPUT SYMBOL	SIZE		PRESET VALUE	UNITS
WDLBS	(15)	Weight dropped during a weight drop event	0.	lbs
WTJET	(15)	Jettison weight/thrust event	0.	lbs
AE	(15)	Engine exit area/thrust event	0.	m^2
S	(15)	Aerodynamic reference area/thrust event	0.	m^2
TAUT	(15)	Thrust event duration time/thrust event	0.	sec
TAUW	(15)	Elapsed time between a thrust event and a weight drop event	0.	sec
NØWD	(15)	Denotes a picket number from which TAUW is defined	0	
NØEVNT	(5)	The total number of thrust events which comprise a stage	0	
PRINT	(15)	Print time increment/thrust event	10.	sec
STEP	(15)	Integration step-size increment for forward run/thrust event	8.	sec

INPUT SYMBOL	SIZE		PRESET VALUE	UNITS
BSTEP	(15)	Integration step-size increment for backward run/thrust event	16.	sec
AZ		Launch azimuth	90.	deg
LAT		Initial geodetic latitude	28.531855	deg
XJEXT		= 1. if maximizing payoff =-1. if minimizing payoff	1.	
CASE		Case number	1.	
DP2		Decimal fraction of constraint error to remove in first iteration	.5	
QY		Decimal fraction of H_a to remover per iteration	.8	
CHIDT		$\dot{\chi}$ for tilt-over during first stage pitch	.1	deg/sec
WØ1		Lift-off weight at TZERØ	0.	lbs
DELVG		ΔV for geometry reserves	0.	m/sec.
DELVP		ΔV for performance reserves	0.	m/sec.

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
WPM		Maximum critical propellant in stage from which performance reserves are taken	0.	lbs.
TCHIR		Time of chi roll initiation (for report tables)		sec
CHRDØT		Roll rate (for report tables)		deg/sec
FAZ		Azimuth at which Fin 1 points (for report tables)		deg
ALØNGØ		Longitude of the launch site (measured positive west)	80.5649528	deg
EU		Upper error bound in forward integration	1.E-5	
BEU		Upper error bound in backward integration	2.E-5	
AYL		Used for error check in forward integration	2.E-3	
BYL		Used for error check in backward integration	4.E-5	
HMN		Minimum step-size for forward integration	.25	sec

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
BHMN		Minimum step-size for backward integration	.50	sec
CMUE		Gravitational constant	3.986032E14	m^3/sec^2
ØMEGA		Angular rotational velocity of earth	7.2921158E-5	rad/sec
CJ		First coefficient in gravitational expansion	1.62345E-3	
H		Second coefficient in gravitational expansion	-5.75E-06	
DJ		Third coefficient in gravitational expansion	7.875E-06	
FLAT		Flattening of Fischer ellipsoid	1/298.3	
RE		Equatorial radius	6378165.0	m
GZERØ		Relates mass to weight	9.80665	m/sec^2
JØRB		= 1 if spherical earth = 0 if oblate earth	0	
JUMP		Jump start at this picket number if JUMP > 1	1	

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
IWDCHI		The number of the weight drop event where atmosphere is dropped	1	
KIND		Type of integration used: =1 for variable step size Adams-Moulton =2 for Runge Kutta =3 for fixed step Adams	3	
KWTA		=2 if χ_p only optimized =3 if χ_p and χ_y optimized	2	
NMAX		Total number of iterations	0	
NTABLE		=1 if output tables are wanted for publication	0	
NVRST		Intermediate constraints imposed at termination of this thrust event. Must be zero if no intermediate constraints wanted.	0	
IPR		Thrust event from which performance reserves are taken. (IPR must be zero if no performance reserves are wanted). If IPR≠0, WPM and CWDØT must be input.	0	

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
LAST		=0 if only one case is run; =1 if more cases are run	0	
KRDER		Order of differences in integration package for forward run.	3	
KINDB		Type of integration used in backward run (See JRBETC+3)	3	
KDERB		Order of differences in integration package used for backward run.	3	
IMP		Jettison weight of this thrust event will be integrated to impact. (Cannot be the last thrust event).	0	
JTHR	(15)	=0 if input thrust and flowrate used =1 if thrust and flowrate found in ATTRAC =-1 if thrust and delta weight found in ATTRAC	0	
KCDPHI	(10)	Terminal function codes. (Code in KCDPHI(1) is payoff)		

INPUT SYMBOL	SIZE		PRESET VALUE	UNITS
PSIREQ	(10)	Constraint values desired at terminal point. (Value in PSIREQ(1) is constraint for code in KCDPHI(2), etc.)		
KCDRES	(6)	Intermediate constraint function codes		
PSIRST	(6)	Constraint values desired at restart point.		
KDB		Control parameter switches		
KDB(1)		Insert 1 to optimize TAUT(1)	0	
KDB(2)		Insert 1 to optimize TAUT(2)	0	
⋮		⋮		
KDB(16)		Insert 1 to optimize WØ1		
KDB(17)		Insert 1 to optimize CHIDT		
KDB(18)		Insert 1 to optimize AZ		
KDT	(17)	Companion vector to KDB. Contains in corresponding locations the number of the thrust event from the present one which is to be altered in order to hold tank limit.	0	

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
WIBT	(17)	Used to speed up or slow down convergence of one parameter relative to another. 1st element of WIBT goes with 1st active parameter, 2nd with 2nd active parameter, etc.	1.	
NBGCT	(4)	Jth control table begins at NBGCT(J)th picket		
NENDCT	(4)	Jth control table ends at NENDCT(J)th picket		
NP	(4)	The number of points in a control table (Must be an odd number of points.)	0 pts	
		CONTROL TABLES		
TTBL(1)	49 pts.	1st time table (real time from TZERØ)		sec
TTBL(51)	"	2nd time table (real time from TZERØ)		
TTBL(101)	"	3rd time table (real time from TZERØ)		
TTBL(151)	"	4th time table (real time from TZERØ)		

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
CPTBL(1)	(49 pts.)	1st χ_p table		rad
CPTBL(5)	"	2nd χ_p table		
CPTBL(101)	"	3rd χ_p table		
CPTBL(151)	"	4th χ_p table		
CYTBL(1)	(49 pts.)	1st χ_y table		rad
CYTBL(51)	"	2nd χ_y table		
CYTBL(101)	"	3rd χ_y table		
CYTBL(151)	"	4th χ_y table		
VIV	(8)	Vector of initial conditions for a jump start, If VIV(7)=0, input: $\dot{x}, \dot{y}, \dot{z}, x, y, z$ ($\dot{z}, \dot{x}, \dot{y}, z, x, y$ Apollo 13) If VIV(7)=2, input: V_I, γ, r, A_z, Lat., Node		
RTASC		Right ascension of outgoing asymptote	0	deg
DECL		Declination of outgoing asymptote	0	deg
VELTGT		Rendezvous Target velocity at node	0	m/sec
GAMTGT		Rendezvous Target Flight Path angle at Node	0	deg

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
RTGT		Rendezvous Target Radius at node	0	m
INCTGT		Rendezvous Target Inclination	0	deg
LATTGT		Rendezvous Target Lauhch site Latitude Displacement (Ignored)	0	deg
BTATGT		Rendezvous Target Position Phase Angle	0	deg
NCOST1		Thrust Event for 1st Analytic Coast	0	
NCOST2		Thrust Event for 2nd Analytic Coast	0	
IAA		=1 calculates launch azimuth for rendizvous ≠1 uses input A_z	0	
IPHIT		=1 calculates LATTGT ≠1 uses input LATTGT	0	
LPRINT		=0 Ignored =1 Suppress χ table print-out =2 Suppress trajectory print out	0	

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
ICØNT		No. of times complete trajectory integration is carried out after convergence	1	
IAEOK		Used in ATTRAC for decaying exit areas	0	
LØNGPR		Number of complete trajectory printouts after convergence Cannot be greater than ICØNT	1	
TH1	(10)	Impact point ellipse centers-latitude	0	deg
PH1	(10)	Impact point ellipse centers-longitude	0	deg
RØTA	(10)	Impact point ellipse rotation angle (pre-counterclockwise)	0	deg
LATWTH	(10)	Impact point ellipse latitude width	0	deg
LØNWTH	(10)	Impact point ellipse longitude width	0	deg

INPUT SYMBOL	SIZE	EXPLANATION	PRESET VALUE	UNITS
NØI		No. of impact point ellipses	1	
IPCNST		= 0 if no impact option = 1 if impact penalty integration is desired	0	
MSWCH	(15)	=1 if staging on time =-1 if staging on fuel	0	
GLIMG	(15)	Axial "g" limit/thrust event	0	g's
FUELG	(15)	Fuel consumed in thrust event where MSWCH =-1	0	lbs

$ INPUT 2

TITLE	-	48 Columns of BCD information
ØFFICE	-	12 columns of BCD information
DATE	-	12 columns of BCD information
NCASE	-	Fixed point case number; should be < 1000
SRID	-	360 columns of BCD information
$END	-	

The alignment of the codes and constraints is

KCDPHI	=	payoff code	1st constraint code	2nd constraint code	etc.
PSIREQ	=		1st constraint value	2nd constraint value	

KCDRES	=	1st constraint code	2nd constraint code	etc.
PSIRST	=	1st constraint value	2nd constraint value	

EXAMPLE PROBLEM

Maximize payload into perigee of a 50-100 nm orbit having an inclination of 55 degrees. Launch at a 38 degree azimuth from Cape Kennedy over an oblate earth using a two-stage space shuttle vehicle. Both stages must be throttled to maintain a 3 "g" axial acceleration limit and are staged on fuel depletion. Controls to be optimized are: lift-off weight and tilt-over $\dot{\chi}$ as well as χ_p and χ_y after 140 sec.

Data for this problem is given below:

Thrust Event	1	2	3
Thrust/engine (lb)	520000.	0.	597000.
Flow rate/engine (lb/sec)	1298.4	0.	1300.6536
Jettison weight (lb)	700000.	0.	0.
Number of engines	12	0	2
Engine exit area (m^2)	2.1869	0.	0.
Aerodynamic ref. area (m^2)	929.	929.	0.
Burn times*(sec)	205.	10.	297.
Integration step (sec)	1.	2.	4.
Print Interval (sec)	20.	20.	20.

Lift-off time = 0

Begin $\dot{\chi}$ tilt at 8 secs

End $\dot{\chi}$ tilt at 30 secs

Begin optimization at 140 secs.

Group thrust events as follows:

1st stage - 2 thrust events; 2nd stage - 1 thrust event

Start 1st control table at picket 1 and end at picket 2 (25 points)

Start 2nd control table at picket 2 and end at picket 3 (5 points)

Start 3rd control table at picket 3 and end at picket 4 (35 points)

---

*Just guesses since staging is on fuel

Estimate χ_p and χ_y in the three tables to be:

	1st table	2nd table	3rd table
χ_p	from 1.14 to 1.22	from 1.34 to 1.37	from 1.23 to 1.74
χ_y	from 0 to -.0015	from -.002 to -.0025	from -.0017 to -.0018

Estimate starting tilt over $\dot{\chi}$ to be .155 deg/sec

Estimate launch weight to be 4788230 lbs

Injection conditions are:

Vel. = 7876.4195 m/sec

Gam = 0

R = 6470762. m

Incl. = 55°

There is a 10 sec coast between the first and second thrust events.

Report tables are to be output.

This problem converges in 6 iterations. All constraints are met within a small tolerance and the max payload is 352840.8 lbs.

A listing of the input cards and aero data for this problem are on the following page.

LIFTING-ROBOT NAMELIST INPUT DATA

```
$INPUT
TZERO=0., TLIFT=8., TTILT=30., TCHFRZ=140.,  DTZ=0.,
FLBS=520000.,0.,597000.,
TNE=12.,0.,0.,0.,0.,0.,0.,0.,2.,
WDOT=1298.4,0.,1300.6536,
WTJET=700000.,
AE=2.1869, S=929.,929.,
TAUT=205.,10.,297.,
TAUW=12., NOWD=2,
NOEVNT=2,1,
PRINT=20.,20.,20.,
STEP=1.,2.,4., BSTEP=2.,4.,8.,
AZ=38., LAT=28.531,
XJEXT=1., CASE=1., DP2=.5, QY=.75,
CHIDT=.155,
W01=.478823E7,
ALONGO=80.6354,
KIND=1, KWTA=3, NMAX=10,
NTABLE=1,
KCDPHI=1,2,3,4,10,
PSIREQ=7876.4195,0.,6470762.,55.,
KDB(16)=1,1,
WIBT=2.,.1,
NBGCT=1,2,3, NENDCT=2,3,4,
NP=25,5,35,
TTBL=140.,200., CPTBL=1.14,1.22, CYTBL=0.,-.0015,
TTBL(51)=200.,210., CPTBL(51)=1.34,1.37, CYTBL(51)=-.002,-.0025,
TTBL(101)=210.,475., CPTBL(101)=1.23,1.74, CYTBL(101)=-.0017,-.0018,
ICONT=1, LONGPR=1,
MSWCH=-1,1,-1,
GLIMG=3.,3.,3.,
FUELG=3053849.6,0.,681542.5,
$END
$INPUT2
NCASE=1,
$END
```

```
      SUBROUTINE CACN(M,L)
C     AERO DATA --  A(M),B(M) AND C(M)  --  CONF 19 BOOSTER HCP ORBITER
      REAL M,MT
      COMMON/AROOTA/ACF,BCF,CCF,DADM,DBDM,DCDM
      COMMON/TABLK/K,MMM(11)
      DIMENSION  MT(20),ATB(20),BTB(20),CTB(20)
      DATA N/12/
      DATA( MT(I),I=1,12)/0.,.5,.85,1.,1.5,2.,3.,4.,5.,6.,7.,20./
      DATA(ATB(I),I=1,12)/2.890,4.652,8.422,7.917,4.781,4.853,
     1 4.386,3.330,2.809,2.651,2.643,2.643/
      DATA(BTB(I),I=1,12)/.2754,1.692,4.598,4.043,1.946,2.563,
     1 2.761,2.069,1.734,1.644,1.644,1.644/
      DATA(CTB(I),I=1,12)/-2.796,-4.786,-8.322,-7.767,-4.641,-4.740,
     1 -4.308,-3.2720,-2.765,-2.619,-2.619,-2.619/
      CALL SPLINE(L,K,N, M , MT,ACF,ATB,BCF,BTB,CCF,CTB,DADM,DBDM,DCDM)
      RETURN
      END
```