TR-792-9-547
MAY 1969

NONLINEAR OPTIMAL GUIDANCE ALGORITHMS

INTERIM REPORT

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ELECTRONICS RESEARCH CENTER
Computer Research Laboratory

UNDER CONTRACT NAS12-500

NORTHROP - HUNTSVILLE
NORTHROP CORPORATION
P.O. BOX 1484 HUNTSVILLE, ALABAMA 35807
TELEPHONE (205) 837-0580

NONLINEAR OPTIMAL GUIDANCE ALGORITHMS

Interim Report

May 1969

By

J. F. Andrus

I. F. Burns

J. Z. Woo

PREPARED FOR:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ELECTRONICS RESEARCH CENTER
COMPUTER RESEARCH LABORATORY

Under Contract NAS12-500

REVIEWED AND APPROVED BY:

J. A. Barclay, Manager
Northrop-Huntsville

NORTHROP-HUNTSVILLE
HUNTSVILLE, ALABAMA

FOREWORD

This interim report summarizes the progress to date of work performed by Northrop-Huntsville while under contract to the Computer Research Laboratory of the NASA Electronics Research Center, Cambridge, Massachusetts (Contract NAS12-500).

Mr. W. E. Miner has served as the NASA technical coordinator during this period.

SUMMARY

This report describes and compares a number of non-linear guidance schemes that require information from a precomputed reference trajectory. However, it is not necessary that the space vehicle closely follow the nominal path. The methods are indirect in nature and are based on one step iterative techniques for solution of the nonlinear boundary equations. For each guidance command, the schemes require accurate evaluation of the functions g_i defining the boundary conditions. The derivatives of the functions g_i required in the iterative techniques are obtained cheaply by correcting precomputed derivatives corresponding to the reference trajectory.

The guidance algorithms considered may be applied to a large variety of space missions, including those requiring bang-bang thrust magnitude control. In general, a broad class of guidance algorithms is described and this report selects one technique from this class which most efficiently solves the optimal guidance problem.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF TABLES

Section I

INTRODUCTION

1.1 BACKGROUND

The purpose of this document is to describe work completed to date under Contract NAS12-500 with the Computer Research Laboratory of the NASA Electronics Research Center. The major goal is to obtain approximate analytical solutions for optimal guidance functions for ascent to orbit. The calculus of variations has been used to formulate necessary conditions for the optimal guidance functions. Efforts have been directed toward deriving approximate solutions of the nonlinear, two-point boundary condition problems that result.

Since March 1968 a different technique has been employed in the development of expansions of the functions g_i (describing the terminal end constraints $g_i = 0$) and the corresponding guidance functions.

Previously, Taylor series expansions in time, about the initial point of the trajectory, were used. The resulting equations were solved by iterative means for the initial values of the Lagrange multipliers, from which the optimal control could be determined. A good approximation to the final time t_F was assumed to be available. However, no initial approximations to the Lagrange multipliers were employed. Some of the main findings of this former approach were: For certain missions, where change in altitude did not exceed 15 kilometers, third- and fourth-order Taylor series would yield accurate results even for range angles to 180 degrees. Accuracy began to fall off rapidly for greater altitude changes unless much higher order series were used. The results achieved from this approach were documented in references 1 and 2.

Since Taylor series expansions in time of the g_i's above the fourth-order are prohibitively complicated, it was concluded that the expansions about the initial point of the flight path should be abandoned and the method should be modified accordingly. In addition it was decided that the method should be augmented and simplified by making use of prior knowledge of the space flight mission to be accomplished.

The different approach still consists of the solution of the end constraints $g_i = 0$, where g_i is considered to be a function of t_F and the initial values of the Lagrange multipliers. Values of g_i, corresponding to given values of t_F and the initial multipliers, are obtained by means of numerical integration of the equations of motion and the Euler-Lagrange equations. A large class of guidance schemes are embodied in the new approach, but in the case of each scheme g_i is expanded in Taylor series about approximations to t_F and the initial Lagrange multipliers obtained from a reference optimal trajectory. These series are terminated after several terms and set equal to zero. The resulting system of polynomial equations are either inverted to obtain explicit expressions for t_F and the initial multipliers, or solved numerically for these corrections. In all of these schemes, derivatives of g_i with respect to t_F and the initial multipliers are required. These derivatives correspond to the initial state and the reference t_F and the initial multipliers. They may be computed numerically (by integration of differential equations referred to as the "equations of variation") or obtained approximately (in ways to be described) from the reference trajectory.

These new nonlinear guidance schemes, unlike the method of Silber and Hunt in reference 3 or second variation guidance (references 4 and 5) are self-correcting; i.e., errors introduced by a drift away from the reference path are removed.

It is possible to strengthen the new methods by combining them (in a manner to be discussed) with the method of Silber and Hunt or the second variation method. In regard to the use of the latter method, there is a note provided in Appendix A, which has been published in the AIAA Journal (ref. 6).

Another guidance method, which combines the new approach with the former of expanding in time about the initial point of the path is given in Appendix B. There is no intention at present of implementing the theory of Appendix A or B.

1.2 PROBLEM DESCRIPTION

The derivation of necessary conditions, by means of the calculus of variations (COV), can be found elsewhere (ref. 7) and will not be repeated here.

The Motion and Euler-Lagrange (MEL) differential equations describing the optimal paths for minimum fuel consumption are

$$\ddot{x} = \frac{F}{m|\lambda|}\lambda - \frac{\mu}{|x|^3}x$$
$$\ddot{\lambda} = \frac{\mu}{|x|^3}\left(-\lambda + \frac{x \cdot \lambda}{|x|^2}x\right) \tag{1-1}$$

where x and $\dot{x}$ are the position and velocity vectors with respect to a non-rotating earth-centered cartesian coordinate system, λ and $-\dot{\lambda}$ are the corresponding Lagrangian multipliers, μ is the gravitational constant, F is the constant thrust magnitude, $m = m_I - \beta(t-t_o)$, and β is the constant fuel burning rate magnitude. The subscripts o and f signify initial and final values, respectively. Let ξ represent an N vector of discrete unknown quantities, e.g., missing initial values, final time, and possibly other unspecified quantities. In addition, define y to be an s vector of initial state parameters, e.g., position, velocity, thrust to weight ratio, and mass flow rate to weight ratio.

The initial and final end constraints may be represented by the equations

$$f_i(\eta, y, \xi) = 0 \qquad (i = 1, \ldots, N)$$

where $\eta = \eta(y, \xi)$ includes the final states and multipliers considered as functions of the initial values. These end constraints are usually geometric end conditions, transversality equations from the COV, and scaling conditions. Let

$$g_i(y, \xi) = f_i[\eta(y, \xi), y, \xi].$$

Then it is desired to solve the equations

$$g_i(y, \xi) = 0 \qquad (i = 1, \ldots, N) \tag{1-2}$$

for ξ in terms of initial state parameters y. Implicit in equations (1-2) is the solution to the differential equations (1-1).

To illustrate the notation, consider a minimum fuel constant burn mission into a prescribed terminal orbit from a specified position and velocity. The initial state vector has the form $y = (x_o, \dot{x}_o, F/m_o, \beta/m_o)^T$ and $t = t_o$ is given. Then ξ becomes the 7 vector $(\lambda_o, \dot{\lambda}_o, t_f)^T$. The seven boundary conditions include five geometric terminal conditions, one transversality equation, and a scaling condition. As a second example, consider a bang-bang control mission with the same geometric constraints of the first example. Further impose the requirement that the trajectory be of a burn-coast-burn nature. The initial state vector y is the same as before with $t = t_o$ given. Now ξ becomes a 9 vector $(\lambda_o, \dot{\lambda}_o, t_1, t_2, t_f)^T$ where t_1 and t_2 are the switch times relating to the end of the first burn arc and the beginning of the second burn arc, respectively. The corresponding nine boundary conditions include the seven of the first example plus an evaluation of the switching function at t_1 and t_2.

In the following sections various iterative techniques for solving equations (1-2) are described. Section II gives a preliminary discussion of iterating functions for one equation in one unknown. Sections III and IV extend two particular methods to N dimensions (nothing new) and applies them to the optimal guidance equations (1-2). Section V presents some numerical results comparing the methods of Sections III and IV. Included also is a discussion of convergence of the two techniques. Based on the studies of convergence and overall performance, Section VI describes a guidance routine based on the best numerical procedure. Sections VII and VIII conclude the report with a discussion of overall performance, summary, conclusions, and extensions.

1.3 INTRODUCTORY COMMENTS

Many guidance schemes can be derived by applying various analytic and numerical techniques to the system of nonlinear equations (1-2).

The method of Silber and Hunt considers equations (1-2) as identities in y, i.e.,

$$g_i[y, \xi(y)] \equiv 0 \qquad (i = 1, \ldots, N)$$

Then the necessary assumptions from implicit function theory are made and Taylor series expansion of $\xi(y)$ about some nominal $\tilde{y}$ are determined.

$$\xi_i(y) = \xi_i(\tilde{y}) + \sum_{\alpha=1}^{S} \frac{\partial \xi_i}{\partial y_\alpha}(\tilde{y})\, \Delta y_\alpha + \frac{1}{2} \sum_{\alpha=1}^{S} \sum_{\beta=1}^{S} \frac{\partial^2 \xi_i}{\partial y_\alpha \partial y_\beta}(\tilde{y})\, \Delta y_\alpha \Delta y_\beta + \ldots$$

Thus, an explicit formula for ξ in terms of the initial state is immediately obtained. If one proceeds further and determines functions of time for the nominal state values and derivatives and substitutes these into the above series, then explicit time and state dependent expressions have been obtained for ξ. Since the control is directly dependent upon ξ, then second variation guidance (ref. 5) can be considered equivalent to the Silber-Hunt series expansion truncated after first-order terms. Similar remarks hold for higher order series and extensions of second variation guidance.

The linear guidance and second variations techniques are non-iterative in nature and consequently not self-correcting. The methods are computationally fast but require a large amount of preparation. The general assumption of these schemes is that the vehicle will fly in some linear region about the reference trajectory, and hence a linear series is sufficient or the region is at worst quadratic hence a second-order expansion is adequate, and so on.

The guidance techniques of this report are designed to give a self-correcting algorithm while still taking advantage of a precomputed nominal trajectory to reduce the required computation.

To visualize the relation between the method of Silber and Hunt and the iterative techniques, consider a simplified geometric explanation. Let y and ξ be simple variables along with the corresponding boundary function $g(y, \xi)$. In Figure 1-1 a three-dimensional surface $g(y, \xi)$ has been sketched. For simplicity assume that the nominal $\tilde{y}$ is zero at some fixed time. It is desired to obtain the trace $g(y, \xi) = 0$ which lies in the y, ξ plane. Suppose the vehicle is currently at the true state $\bar{y}$. Then it is required to calculate ξ denoted at point 1. The linear method of Silber and Hunt uses the tangent line through $\tilde{\xi}$ in the $\bar{y}$, ξ plane to estimate ξ by point 2. Of course, higher order methods would pass higher order polynomials through $\tilde{\xi}$ in the y, ξ plane. A "linear" iterative method (e.g. Newton Raphson) uses the tangent line through the point $[\bar{y}, \tilde{\xi}, g(\bar{y}, \tilde{\xi})]$ in the ξ, g plane denoted by point 3. The intersection of this line with the y, ξ plane is the estimate ξ^* for ξ. This process can be repeated by using the tangent line through $[\bar{y}, \xi^*, g(\bar{y}, \xi^*)]$ to obtain a new estimate for ξ. The procedure may be repeated until $g(\bar{y}, \xi^*)$ is sufficiently small. Again, the use of higher order approximations at each stage should improve the speed of convergence.

The computation of the parameters of the approximating curve at point 3 is somewhat time consumming and it must be done at least once for every change in y. However, it seems plausible to approximate them from the corresponding parameters at $\tilde{\xi}$. Further details may be found in subsection 2.3 and the numerical results of subsection 5.4 indicate the approximations work well.

When comparing any guidance schemes it is very difficult to say, a priori, that one is better than the other. Each procedure must be empirically tested and compared. Iteration versus closed formulas leads to subtle questions and numerical investigation would resolve a few important ones. In fact, it will be seen (Section V), that closed formulas fail more often in some problems than implicit formulas requiring iteration.

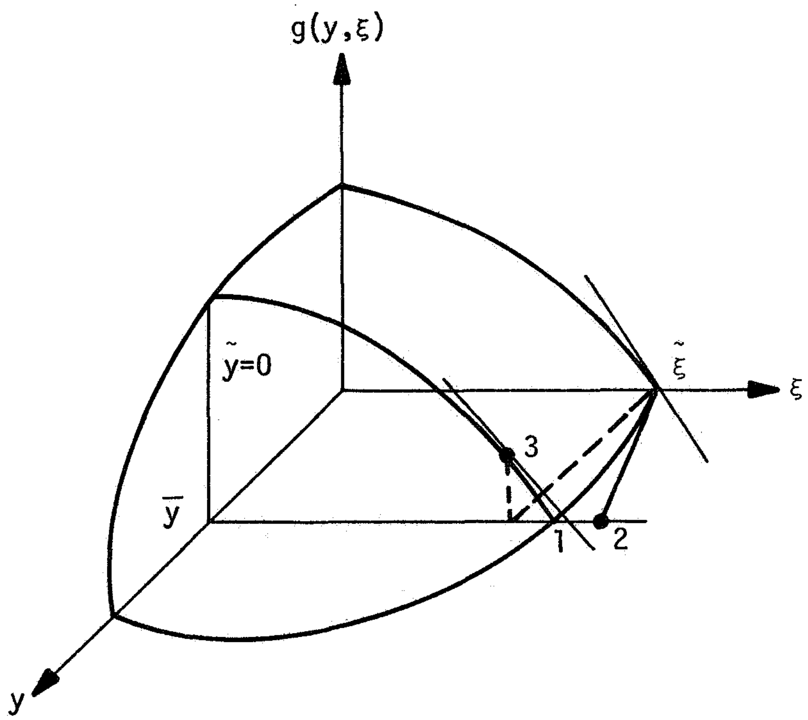

Figure 1-1. GEOMETRICAL INTERPRETATION

Section II

ITERATING FUNCTIONS AND INVERSION

2.1 INVERSION AND ONE POINT ITERATING FUNCTIONS

Here the intention is to clarify the relationship between series inversion and "iteration". For simplicity in presentation, a function of one variable is used.

Given $$f(z) = 0 \tag{2-1}$$

find the values of z, the zeros or roots, that satisfy equation (2-1). Constructing infinite processes that involve f(z) and its derivatives such that values of z can be obtained which satisfy equation (2-1) is the primary goal. In other words, one desires to construct functions that generate a convergent sequence of approximations $\{z_i\}$ to a zero $\bar{z}$, i.e.,

$$\lim_{i\to\infty} z_i = \bar{z} \quad .$$

Let these functions be denoted as

$$z_{i+1} = \phi(z_i) = \phi(f, f', f'', \ldots; z_i). \tag{2-2}$$

Traub (ref. 8) classifies these as one point iteration functions since z_{i+1} depends upon only one point z_i. If the recursion (2-2) is to have meaning then the identity

$$\bar{z} = \phi(\bar{z})$$

must hold.

An r^{th} order iteration is characterized by the conditions

$$\phi(\bar{z}) = \phi'(\bar{z}) = \phi''(\bar{z}) \ldots \phi^{(r-1)}(\bar{z}) = 0 \text{ and } \phi^{(r)}(\bar{z}) \neq 0 \tag{3-3}$$

where

$$\phi^{(r)} = \frac{\partial^r \phi}{\partial z^r}$$

An iteration of arbitrary order can be produced by construction of a ϕ function that satisfies the conditions of equation (2-3). One such construction is

$$\phi = z + \sum_{\alpha=1}^{r-1} (-1)^{\alpha} \frac{f^{\alpha}}{\alpha!} \left(\frac{1}{f'} \partial\right)^{\alpha-1} \frac{1}{f'} - f^{r}\mu_{r} \tag{2-4}$$

where μ_r is an arbitrary function. By $\left(\frac{1}{f'} \partial\right)^{\alpha-1}$ denote an operator that stands for the following operation: Differentiate the function following the symbol, multiply it by $\frac{1}{f'}$, then differentiate the new function again and multiply it by $1/f'$,... continue for $\alpha-1$ times. For example

$$\left(\frac{1}{f'} \partial\right)^{3} \frac{1}{f'} = \frac{1}{f'} \frac{d}{dz} \left[\frac{1}{f'} \frac{d}{dz} \left(\frac{1}{f'} \frac{d}{dz} \frac{1}{f'}\right)\right]$$

The first four terms of equation (2-4) are:

$$\phi = z - \frac{f}{1!} \frac{1}{f'} + \frac{f^2}{2!} \frac{f''}{(f')^3} - \frac{f^3}{3!} \frac{[3(f'')^2 - f'f''']}{(f')^5} + \ldots$$

which is recognized as the series expansion for the inverse of $f(z)$.

Note that the order of iteration is one more than the corresponding order of the series. For example, the first-order inverse series is

$$\phi = z - \frac{f}{f'}$$

which is also the Newton-Raphson iteration and which is of order two. The third-order inverse series written above defines a fourth-order iteration.

Now it can also be shown (ref. 8) that m iterations of an N^{th} order iteration function are equivalent to one iteration with an iteration function of order N^m. Thus, two iterations with Newton-Raphson should be equivalent to one iteration with, or a single evaluation of, a third-order inverse series.

It would appear that there is no essential difference between using several iterations with an inverse series of an order appropriately higher; provided, of course, that both methods converge.

The higher order inverse series requires higher order derivatives. Except for functions such as polynomials where the higher derivatives become less complicated, the higher order iterations (or inverse series) are likely to be more and more ineffecient in terms of computation. Another point to consider is that there may be significant differences in the behavior of convergence between iteration and use of an inverse series of equivalent accuracy.

2.2 INTERPOLATORY ITERATING FUNCTIONS

Given p approximations to a root $\bar{z}$ of f(z), e.g., z_{N+1}, z_{N+2}, ..., z_{N+p}, it seems reasonable to obtain a new approximation, z_{N+p+1}, by calculating a root of the interpolation polynomial determined by the p approximations. Then repeat again with the points z_{N+2}, z_{N+3}, ..., z_{N+p+1}.

The types of interpolation polynomials are many and varied. However, here, interest is directed toward hyperosculatory interpolation. In particular it is desired that the interpolation polynomial agree with f and various derivatives of f at the p approximations.

In Section III only one interpolation point is used and the function value with its first two derivatives are constrained equal to the interpolation polynomial.

For this case it is shown (ref. 8) that the order of the iteration is equal to the order of the interpolation polynomial.

2.3 APPROXIMATE INVERSE SERIES BY NUMERICAL INTEGRATION FORMULAS

The technique discussed here has been examined but not investigated numerically. For this reason no recommendations concerning its use are offered at this time. There is a method for solution of nonlinear equations called "Variation of Parameters" attributed to Davidenko (ref. 9). The basic approach is as follows:

Given $$f(z) = w$$

find a root $z = \bar{z}$ of $f(z) = 0$. The idea is to start with some approximation z_o to $\bar{z}$ with a corresponding parameter value w_o and vary w_o in some continuous fashion such that w_o moves to zero and z_o to $\bar{z}$. If

$$\frac{df}{dz} = f'(z) \neq 0 \text{ for } z \in D$$

where D is some region about the point z_o and is assumed to contain $\bar{z}$, then

$$\frac{dz}{dw} = \frac{1}{f'(z)} \quad z \in D \tag{2-5}$$

Now consider equation (2-5) as a differential initial value problem with

$$z(w_o) = z_o$$

Then it is desired to integrate equation (2-5) from w_o to 0. The path of integration is assumed to be such that z remains in D.

First, suppose that one solves the differential equation by Taylor series. The interval of expansion is $(0 - w_o)$ and thus obtain

$$\begin{aligned}\bar{z} &= z(w_o) + \frac{dz}{dw}(w_o)(-w_o) + \frac{d^2z}{dw^2}(w_o)\frac{(-w_o)^2}{2!} + \ldots \\ &= z_o + \frac{1}{f'(z_o)}[-f(z_o)] + \frac{d}{df}\left(\frac{1}{f'}\right)\left(-\frac{f(z_o)}{2!}\right)^2 + \ldots \\ &= z_o + \frac{1}{f'(z_o)}[-f(z_o)] + \frac{[-f''(z_o)]}{f'(z_o)^2}\frac{[-f(z_o)]^2}{2!} + \ldots\end{aligned}$$

It is evident that the solution is the inverse series, as expected.

Now, regard the root-finding problem as equivalent to solving a differential equation. Then there are opportunities to simplify the series inversion problem. In particular, Runge-Kutta integration formulas can be used to construct approximations to the inverse series.

Recall the Runge-Kutta integration formulas are derived so that solutions to differential equations obtained by them will agree with Taylor series solutions of some order. Thus, a third-order Runge-Kutta formula agrees with at least a third-order Taylor series solution, etc.

Now suppose that one solves the differential equation (2-5) by means of various Runge-Kutta integration formulas. These formulas can be written in a general, familiar format as

$$y_{n+1} = y_n + h\ \Phi_i$$

where the differential equation is

$$y' = \psi(y; x)$$

Here the subscript n+1 refers to the independent variable x+h and n to x. In terms of the root finding initial value problem equation (2-5), rewrite the formulas as

$$z_{n+1} = z_n - f(z_n)\ \Phi_i$$

Note that the step size, h, is $-f(z_n)$; Φ_i represents one from the family of Runge-Kutta integration formulas. The philosophy of the step size determination can be explained as follows: Integrate from w_o to 0 in one step giving a stepsize of $-w_o = -f(z_o)$. Then having determined $\tilde{z} = \bar{z}$, try to improve $\tilde{z}$ by setting $z_1 = \tilde{z}$ and integrating from w_1 to 0, i.e., $h = -w_1 = -f(z_1)$ and obtain a new $\tilde{z}$. Thus, the iterative method is sufficiently defined. Of course many variations of this procedure are possible, in particular concerning stepsize control of the numerical integration.

Consider the simplest Runge-Kutta formula. (The Euler or Point-Slope Method)

$$\Phi_1 = \frac{1}{f'(z_n)}$$

The approximate solution of the differential equation is

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}$$

which is the Newton-Raphson Method.

Other Runge integration formulas yield other iterations - e.g.,

$$\Phi_2 = \frac{1}{2}(k_1 + k_2) \qquad \text{(HEUN)}$$

$$k_1 = \frac{h}{f'(z_n)} \qquad k_2 = \frac{h}{f'\ (z_n + k_1)}$$

$$\Phi_3 = \frac{1}{6} (k_1 + 4k_2 + k_3) \qquad \text{(RUNGE)}$$

$$k_1 = \frac{h}{f'(z_n)}$$

$$k_2 = \frac{h}{f'(z_n + \frac{1}{2} k_1)}$$

$$k_3 = \frac{h}{f'(z_n + 2k_2 - k_1)}$$

$$\Phi_4 = \frac{1}{4} (k_1 + 3 k_3) \qquad \text{(HEUN)}$$

$$k_1 = \frac{h}{f'(z_n)}$$

$$k_2 = \frac{h}{f'(z_n + \frac{1}{3} k_1)}$$

$$k_3 = \frac{h}{f'(z_n + \frac{2}{3} k_2)}$$

There are available Runge-Kutta formulas for various higher orders. In the above formulas Φ_1 is the first-order integration method, Φ_2 is second-order, and Φ_3 and Φ_4 are third-order. As iteration methods for roots the orders are: Φ_1, second-order, Φ_2 third-order, and Φ_3 and Φ_4 fourth-order. Higher order integration methods will, of course, provide higher order iteration methods for root-finding, or more accurate approximations to the inverse functions.

An advantage of this approach is that higher than first derivatives are not required, as is the case when the formal inverse series is used. Partially offsetting this advantage is the need to perform additional function evaluations. Usually $p \geq n$ function evaluations are necessary for a Runge-Kutta formula of order n that approximates an inverse series of order n which, in turn, defines an iteration of order n + 1.

Section III

GUIDANCE BY SOLUTION OF POLYNOMIAL EQUATIONS

In this section the system of nonlinear equations (1-2) is treated. Here, consideration is given to interpolatory iteration functions. Only one iteration (i.e., the solution of one set of polynomials) is considered and numerical results (see Section V) indicate this is sufficient.

3.1 DEVELOPMENT OF EQUATIONS

Let ξ' be an approximation to the solution ξ. Then pass a p^{th} degree polynomial (i.e., N polynomials in the N variables $\xi_i - \xi'_i$) such that its value at ξ' agrees with $g(y, \xi')$. Similarly, constrain the first p-1 derivatives of the polynomials to agree with the first p-1 derivatives of g_i at the point (y, ξ'). This is, of course, equivalent to a truncated Taylor expansion of $g_i(y, \xi)$ about ξ'. Then let

$$\Delta\xi = \xi - \xi'$$

$$g_i^{(j)} = \frac{\partial g_i}{\partial \xi_j} ,$$

$$g_i^{(j,k)} = \frac{\partial^2 g_i}{\partial \xi_j \partial \xi_k} .$$

A Taylor series expansion of $g_i(y, \xi) = 0$ about ξ' yields

$$g_i(y, \xi') + \sum_{j=1}^{N} \Delta\xi_j \, g_i^{(j)}(y, \xi') + \frac{1}{2!} \sum_{j=1}^{N} \sum_{k=1}^{N} \Delta\xi_j \Delta\xi_k \, g_i^{(j,k)}(y, \xi')$$

$$+ \frac{1}{3!} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{\ell=1}^{N} \Delta\xi_j \Delta\xi_k \Delta\xi_\ell \, g_i^{(j,k,\ell)}(y,\xi') + \ldots = 0 \qquad (3\text{-}1)$$

$$(i = 1, 2, \ldots, N)$$

The guidance scheme requires solution of the resulting system of polynomial equations for $\Delta\xi$ by means of, e.g., the Newton-Raphson method of iteration. Once the polynomial coefficients have been calculated, the iterative solution of equation (3-1) requires no additional trajectory calculations or numerical integrations.

3.2 CALCULATION OF DERIVATIVES

The derivatives $g_i^{(j)}(y, \xi')$, $g_i^{(j,k)}(y, \xi')$, ... may be calculated by means of the equations of variations (ref. 3,5). Although the latter method may be used in numerical studies, it is out of the question (at least in the case of higher derivatives) in onboard implementation of the guidance shceme. Instead, the derivatives may be approximated as follows:

$$g_i^{(j)}(y, \xi') \doteq g_i^{(j)}(\tilde{y}, \tilde{\xi}) + \sum_{\alpha=1}^{S} \Delta y_\alpha \, g_{iy_\alpha}^{(j)}(\tilde{y}, \tilde{\xi}) + \sum_{\alpha=1}^{N} (\xi'_\alpha - \tilde{\xi}_\alpha) \, g_i^{(j,\alpha)}(\tilde{y}, \tilde{\xi})$$

$$g_i^{(j,k)}(y, \xi') \doteq g_i^{(j,k)}(\tilde{y}, \tilde{\xi}) \qquad (3\text{-}2)$$

where $\tilde{\xi}$, $\tilde{y}$ are the reference values and $\Delta y = y - \tilde{y}$. Given y, the reference values $\tilde{y}$ and $\tilde{\xi}$ may be determined by any of several procedures. Appendix A discusses one technique, but in implementing the guidance routine a time-to-go criterion is used (Section VI). Derivatives of g_i with respect to t_F may be computed precisely without the use of either the equations of variation or equations (3-2).

Numerical calculations of the derivatives, corresponding to an S-IVB injection into circular orbit, indicate that the derivatives do not vary radically as a function of y within a rather large neighborhood of $\tilde{y}$. On the other hand, the Lagrange multipliers and t_F change appreciably. Furthermore, as long as the functional values, $g_i(y, \xi')$, are computed accurately, it is not necessary to have very accurate higher derivatives in order to compute accurate guidance commands. Therefore, it is reasonable to use equations (3-2) to determine the derivatives.

3.3 UPDATING THE DERIVATIVES

In the preceding subsection the calculation of derivatives corresponding to some fixed initial time were considered (i.e., ξ was defined in terms of a fixed initial time). However, in the guidance problem the derivatives $g_i^{(j)}(\tilde{y}, \tilde{\xi})$, $g_i^{(j,k)}(\tilde{y}, \tilde{\xi})$, etc. must be updated from time t_1 to time t_2 as the space flight progresses. This may be done by either of the following two means:

- Numerical integration of the adjoint differential equations (ref. 10) forward over the short time interval between t_1 and t_2.
- Evaluation of polynomials expressing the reference derivatives as functions of time. These polynomials can be determined before flight.

It should be noted that forward integration of the adjoint equations may be numerically unstable, but the severity of this problem is not thought to be great.

3.4 USE OF THE METHOD OF SILBER AND HUNT AS A REFINEMENT

A refinement to the guidance schemes disucssed above is the use of the technique of Silber and Hunt (ref. 3) to determine a first correction ξ' to ξ for given Δy. Thus

$$\tilde{\xi}' = \tilde{\xi} + \sum_{i=1}^{S} \Delta y_i \tilde{\xi}_{y_i} + \frac{1}{2!} \sum_{i=1}^{S} \sum_{j=1}^{S} \Delta y_i \Delta y_j \tilde{\xi}_{y_i y_j}$$

$$+ \frac{1}{3!} \sum_{i=1}^{S} \sum_{j=1}^{S} \sum_{k=1}^{S} \Delta y_i \Delta y_j \Delta y_k \tilde{\xi}_{y_i y_j y_k}$$

where $\tilde{\xi}$, $\tilde{\xi}_{y_i}$, etc., are to be evaluated along the reference trajectory. Again the latter derivatives can be expressed as polynomial functions of time for updating or they can be obtained by integrating matrix Riccati equations from the time of one guidance command to the next (ref. 10).

Section IV

INVERSION FORMULAS FOR GUIDANCE

4.1 DEVELOPMENT OF EQUATIONS

Here, N-dimensional inversion formulas for the equations (1-2) are presented. The inverse series can be derived by letting the equations

$$g_i(y, \xi) = w_i \qquad (i = 1, \ldots, N)$$

define ξ implicity as a function of w_i for fixed y. Then expanding ξ in a Taylor series about $w_i' = g(y, \xi')$ and evaluating the series at $w_i = 0$ gives the resulting inversion. This straight forward inversion is carried out in reference 11 with the following result:

$$\Delta\xi = -A^{-1}\left\{g + \frac{1}{2!}\sum_{j=1}^{N}\sum_{k=1}^{N}\left[\left(\sum_{\alpha=1}^{N} c_{j\alpha}g_\alpha\right)\left(-\sum_{\beta=1}^{N}\sum_{\gamma=1}^{N}\theta_{\beta\gamma}^{(k)}\; g_\beta g_\gamma + \sum_{\beta=1}^{N} c_{k\beta}g_\beta\right)\right]g^{(j,k)} - \frac{1}{3!}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{\ell=1}^{N}\left[\left(\sum_{\alpha=1}^{N} c_{j\alpha}g_\alpha\right)\left(\sum_{\beta=1}^{N} c_{k\beta}g_\beta\right)\left(\sum_{\gamma=1}^{N} c_{\ell\gamma}g_\gamma\right)\right]g^{(j,k,\ell)}\right\} \qquad (4\text{-}1)$$

where we have truncated after third-degree terms in g_α, g_β, and g_γ and where

$$A = \left[g_i^{(j)}(y, \tilde{\xi})\right]$$

$$C = (c_{ij}) = A^{-1}$$

$$\theta_{\beta\gamma}^{(i)} = \frac{\partial^2\Delta\xi_i}{\partial g_\beta \partial g_\gamma} = i^{th} \text{ component of } \left[-A^{-1}\sum_{j=1}^{N}\sum_{k=1}^{N} c_{j\beta}c_{k\beta}\; g^{(j,k)}\right]$$

In formula (4-1), g and its derivatives are to be evaluated at (y, ξ'). The derivatives $g_i^{(j)}(y, \xi')$, $g_i^{(j,k)}(y, \xi')$, etc, may be determined in the manner discussed in subsections 3.2 and 3.3.

An explicit and self-correcting guidance formula, giving $\Delta\xi$ in terms of Δy, may be obtained by expressing the nominal derivatives $g_i^{(j)}(y, \tilde{\xi}),\ldots$ in terms of polynomial functions of time (which update the derivatives as discussed in subsection 3.3), and substitution of equations (3-2) into equation (4-1). However, it is to be recalled that $g(y, \xi')$ would be computed by means of numerical integration.

Section V

COMPARISON OF METHODS

This section deals with the comparison of the inversion technique of Section IV and the polynomial equation method of Section III. The numerical results include comparison with the method of Silber and Hunt.

5.1 CONVERGENCE IN THE LARGE

The discussion of this subsection is by necessity intuitive and will be limited to a discussion of the solution of a single equation $g(\xi) = 0$ in one unknown ξ. However, the ideas can be generalized.

The desired solution is ξ_s (Figure 5-1). A Newton iteration with proper damping (limiting) of the corrections would converge to ξ_s for any initial approximation between ξ_A and ξ_B. The region (not radius) of convergence of an undamped Newton method would be quite a bit smaller (but the rate of convergence usually faster). A second degree polynomial, passing through the point $(\tilde{\xi}, g(\tilde{\xi}))$ and having its first and second derivatives equal to those of g at $\xi = \tilde{\xi}$, would appear to have a larger region of convergence than Newton's method. Perhaps, in many cases, the region would be nearly as large as that of the damped Newton method. However, inversion about $\tilde{\xi}$ would lead to a series which does not converge outside of the interval I indicated in Figure 5-1, because the radius of convergence of the series would be less than $g(\xi)_A - g(\xi)$, there being a singular point in the inverse series at $\xi = \xi_A$. Although the inversion formula may give an explicit solution, any advantage this may have is reduced by the limited region of convergence of the inverse series.

It will be seen that some of these intuitive observations are clearly substantiated by numerical results to be given.

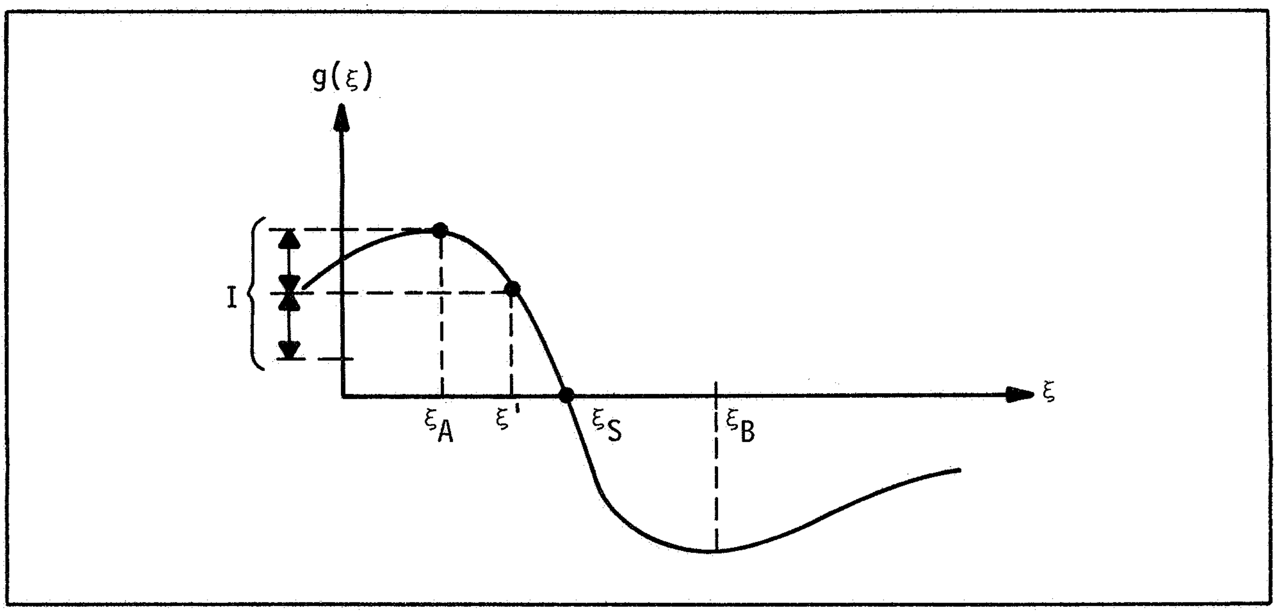

Figure 5-1. GEOMETRY OF CONVERGENCE

5.2 ASYMPTOTIC CONVERGENCE

In reference 6 it is shown that, after the convergence process is well underway (asymptotic convergence), the approximation afforded by an N^{th} degree polynomial or N^{th} order inversion formula, of the types we have discussed, is an $(N + 1)^{th}$ order iteration function. Also, M applications of the method are equivalent to one application of an $(N + 1)^{M\ th}$ order formula. Therefore, two applications of Newton-Raphson (order 2) gives the equivalent of a fourth-order iteration function, the same as that of a third degree inversion or polynomial formula.

Each Newton-Raphson iteration requires one evaluation of the function to be driven to zero and one evaluation of its first derivative. Each N^{th} order inversion or polynomial solution requires one evaluation of the function and the first through the N^{th} derivatives. If the derivatives require no more time to evaluate than the functions themselves, then four units (or less) of time are necessary for two Newton-Raphson iterations while four units (or less) are required for one third-degree inversion or polynomial solution. However, in the guidance problem it is much more time consuming to compute the derivatives than the functional values themselves, assuming the equations of variations or finite differences to be used in calculating the derivatives. Therefore, it is clear that if higher degree formulas are to be used for guidance, the derivatives must be determined in some other manner. For this reason it is recommended that the derivatives be calculated from precomputed derivatives from the reference trajectory as shown in subsection 3.2. It remains to be determined whether or not these approximate derivatives will be accurate enough to give the advantage to the higher degree methods over applications of the Newton-Raphson algorithm which requires more functional evaluations in order to give an iterative formula of comparable order.

5.3 NUMERICAL COMPARISON

We now consider a numerical comparison of the guidance formulas discussed in this report. The symbol N will represent the degree of the formula used.

The problem under numerical study is that of an S-IVB minimum time injection into a 105 nautical mile circular orbit from a point 5 miles below the orbit. The initial and final end constraints for the two dimensional problem are

$$f_1 = x_f \cdot x_f - R_{co}^2 = 0 \qquad \text{(orbital radius)}$$

$$f_2 = \dot{x}_f \cdot \dot{x}_f - V_{co}^2 = 0 \qquad \text{(orbital velocity)}$$

$$f_3 = x_f \cdot \dot{x}_f = 0 \qquad \text{(orthogonality)}$$

$$f_4 = \lambda_o \cdot \lambda_o - 1 = 0 \qquad \text{(scaling)}$$

$$f_5 = \lambda_{10}\dot{x}_{20} - \lambda_{20}\dot{x}_{10} - \dot{\lambda}_{10}x_{20} + \lambda_{20}x_{10} = 0 \text{ (transversality)}$$

and

$$y = \left(x_{10}, x_{20}, \dot{x}_{10}, \dot{x}_{20}, \frac{F}{m_o}, \frac{\beta}{m_o}\right)^T$$

with

$$\xi = (\lambda_{10}, \lambda_{20}, \dot{\lambda}_{10}, \dot{\lambda}_{20}, t_f)^T .$$

Generally speaking the transversality condition should be imposed at the terminal point, however, in this case the function f_5 is a constant of the motion and its initial value is equivalent to the terminal value. This, of course, reduces the number of derivatives required at final time. Thus it is required to solve the equations

$$g_i(y, \xi) = 0 \qquad (i = 1, \ldots, 5)$$

where, for example

$$g_1(y, \xi) = f_1[\eta(y, \xi), y, \xi]$$

$$= x_f(y, \xi) \cdot x_f(y, \xi) - R_{co}^2 \ .$$

The reference trajectory (which satisfies, in this case, all boundary conditions) has the following parameters defining it:

$x_{10} = 1761674.2$ meters

$x_{20} = 6314804.0$ meters

$\dot{x}_{10} = 6546.5205$ meters/sec

$\dot{x}_{20} = -1728.0676$ meters/sec

Final altitude $\doteq$ 105 nautical miles:

$R_{co} = 6565710.$ meters

$V_{co}^2 = \mu / R_{co}$

$m_o = 16645.5$ "Mass Units"

$\beta = 22.0179$ "Mass Units"/sec (Mass flow rate)

$c = 4120.193$ Meters/sec (exhaust velocity)

$F = c\beta$

Multipliers

$\lambda_{10} = .974$

$\lambda_{20} = .228$

$\dot{\lambda}_{10} = -.179 \times 10^{-2}$

$\dot{\lambda}_{20} = -.456 \times 10^{-2}$

$t_F = 170.3$ seconds.

The initial state can be defined by means of an altitude A, a velocity magnitude V, the angle θ between the local horizontal and the velocity vector, and the mass m_o. In order to give the guidance algorithms a severe test, 16 perturbations on the initial reference values of A, V, θ, and m were made.

The perturbations were ± 5 percent in A_o, V_o, and m_o with ± 1 degree changes in θ_o. Some of these perturbations are rather unreasonable. Some are severe enough to throw the initial radius of the trajectory into the 105 nautical mile orbit. Table 5-1 lists the perturbations and also defines the initial state of each of the 16 cases.

The true multipliers and final time for these 16 cases were obtained by using a Newton-Raphson type of iteration. The "nominal values" (i.e., the initial values taken from the reference trajectory) were used as initial guesses in the iterative process. (Here and following the words "nominal" and "reference" are used interchangeably.) The iteration differed from the classical algorithm in that the full corrections were damped so as not to exceed certain tolerances. The derivatives on each step of the iteration were computed by integrating the equations of variation. In all cases four or five iterations were sufficient.

The expansion of Silber and Hunt was obtained for the first point of the reference trajectory up to second-order terms. The series gives the ξ_i explicitly in terms of the perturbations Δy_i. The angle χ and its time derivative $\dot{\chi}$ are computed directly from the ξ_i. These computations are outlined in detail in Section VI. The angle χ and its time derivative $\dot{\chi}$ are measured in degrees with time in seconds. In Table 5-2 the results of the first ($N = 1$) and second ($N = 2$) order expansions in the 16 perturbations are given. Also the true values are tabulated with the corresponding percent errors. The corresponding multipliers λ_o and $\dot{\lambda}_o$ for Table 5-2 are tabulated in Appendix C as well as those for the other tables presented in this section.

Based on the data in Table 5-2, it was decided to use the second-order expansion of Silber and Hunt in the guidance algorithm to give starting values for the first guidance command (see Section VI for further description). In the majority of the cases the second-order expansion significantly improved the results compared to the first order.

The guidance formulas described in Sections III and IV require the computation of various derivatives. In Table 5-3 the derivatives in the guidance formulas are calculated by integrating the equations of variation. The resulting

Table 5-1. PERTURBATIONS OF NOMINAL TRAJECTORY

CASE	A	V	M	θ
1	+	+	+	+
2	+	+	-	+
3	+	+	+	-
4	+	+	-	-
5	+	-	+	+
6	+	-	-	-
7	+	-	+	-
8	+	-	-	+
9	-	+	+	+
10	-	+	-	+
11	-	+	+	-
12	-	+	-	-
13	-	-	+	+
14	-	-	-	-
15	-	-	+	-
16	-	-	-	+
Nominal	100 n mi	6780.6832 meters/sec	16645.597 "mass units"	0 degrees

Table 5-2. SILBER-HUNT (S-H) EXPANSION

CASE		X	$\dot{X}$	t_F	% ERROR X	% ERROR $\dot{X}$	% ERROR t_F
1	True Values	-31.2	-.36	128.6			
	Nominal	76.8	.23	170.3	345.9	164.8	-32.5
	S-H N=1	-66.8	-.06	110.8	-113.9	117.2	13.8
	S-H N=2	-44.8	-.20	121.7	-43.3	44.4	5.3
2	True Values	-30.2	-.39	116.8			
	Nominal	76.8	.23	170.3	354.7	159.2	-45.9
	S-H N=1	-66.6	-.09	93.7	-120.9	121.9	19.8
	S-H N=2	-40.6	-.19	114.2	-47.8	51.3	2.3
3	True Values	65.7	.302	127.1			
	Nominal	76.8	.23	170.3	-17.0	23.5	-34.0
	S-H N=1	66.8	.28	130.0	-1.7	6.3	-3.1
	S-H N=2	65.1	.30	128.9	.9	1.1	-1.4
4	True Values	66.7	.34	114.4			
	Nominal	76.8	.23	170.3	-15.2	31.6	-48.9
	S-H N=1	66.1	.31	113.9	-.5	8.7	.4
	S-H N=2	65.3	.33	118.0	2.1	1.2	-3.2
5	True Values	-77.5	-.02	223.2			
	Nominal	76.8	.23	170.3	199.1	-1107.6	23.7
	S-H N=1	-70.2	-.10	212.5	9.5	624.7	4.8
	S-H N=2	-71.6	-.04	230.9	7.6	321.7	-3.4
6	True Values	68.5	.17	212.5			
	Nominal	76.8	.23	170.3	-12.2	-38.5	19.8
	S-H N=1	71.3	.13	212.8	-4.1	24.5	-.1
	S-H N=2	66.1	.19	215.0	3.4	-14.7	-1.2
7	True Values	66.0	.15	237.6			
	Nominal	76.8	.23	170.3	-16.4	-55.8	28.3
	S-H N-1	70.9	.10	229.9	-7.5	32.7	3.2
	S-H N=2	62.7	.18	240.7	5.0	-22.6	-1.3

Table 5-2. SILBER-HUNT (S-H) EXPANSION (Continued)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
8	True Values	-74.4	-.001	201.6			
	Nominal	76.8	.23	170.3	203.2	18666.1	15.5
	S-H N=1	-70.0	-.07	195.3	6.0	-5885.6	3.1
	S-H N=2	-68.4	-.08	208.0	8.1	-6221.0	-3.2
9	True Values	79.4	.48	129.9			
	Nominal	76.8	.23	170.3	3.2	51.5	-31.1
	S-H N=1	87.1	.32	125.9	-9.8	32.2	3.0
	S-H N=2	83.7	.40	129.2	-5.4	15.2	.5
10	True Values	75.1	.55	119.7			
	Nominal	76.8	.23	170.3	-2.4	57.8	-42.3
	S-H N=1	87.5	.35	108.8	-16.5	35.7	9.1
	S-H N=2	80.7	.46	121.0	-7.5	16.3	-1.1
11	True Values	39.9	.16	167.9			
	Nominal	76.8	.23	170.3	-92.5	-48.9	-1.5
	S-H N=1	40.7	.18	146.2	-1.9	-16.6	12.9
	S-H N=2	28.5	.07	161.9	28.5	52.3	3.6
12	True Values	38.9	.17	154.3			
	Nominal	76.8	.23	170.3	-97.5	-38.1	-10.4
	S-H N=1	40.9	.19	129.0	-5.1	-16.0	16.4
	S-H N=2	27.0	.07	150.4	30.6	57.5	2.6
13	True Values	81.2	.16	228.6			
	Nominal	76.8	.23	170.3	5.4	-45.3	25.5
	S-H N=1	82.3	.13	227.6	-1.4	16.0	.5
	S-H N=2	81.7	.16	230.9	-.7	-1.5	-1.0
14	True Values	54.2	.15	229.7			
	Nominal	76.8	.23	170.3	-41.7	-52.2	25.8
	S-H N=1	43.8	.13	227.9	19.3	12.6	.8
	S-H N=2	58.2	.09	235.3	-7.3	43.1	-2.5

Table 5-2. SILBER-HUNT (S-H) EXPANSION (Concluded)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
15	True Values	53.7	.13	256.0			
	Nominal	76.8	.23	170.3	-43.1	-78.0	33.4
	S-H N=1	43.6	.12	245.1	18.9	8.6	4.3
	S-H N=2	57.5	.05	261.6	-7.1	63.0	-2.2
16	True Values	81.9	.18	206.4			
	Nominal	76.8	.23	170.3	6.2	-29.1	17.5
	S-H N=1	82.6	.16	210.4	-.9	9.3	-2.0
	S-H N=2	82.1	.18	207.4	-.3	-.8	-.5

Table 5-3. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES

CASE		X	$\dot{X}$	t_F	% ERROR X	% ERROR $\dot{X}$	% ERROR t_F
1	Inversion						
	N=1	-80.7	.210	116.1	-158.9	159.1	9.6
	N=2	-60.6	.004	112.5	-94.2	101.0	12.4
	N=3	-45.5	-.190	115.8	-46.0	46.0	9.8
	Polynomial Sol.						
	N=1	-80.7	.210	116.1	-158.9	159.1	9.6
	N=2	-54.4	-.212	131.7	-74.4	40.2	-2.4
	*N=3	77.5	.368	-42.5	348.5	203.4	133.0
	Newton-Raphson						
	2 Iterations (Damped on 2nd)	-35.2	-.376	115.8	-12.8	-5.8	9.8
2	Inversion						
	N=1	-86.5	.269	108.8	-186.4	168.7	6.8
	N=2	-68.1	.119	107.0	-125.4	130.5	8.3
	N=3	-53.7	-.075	102.9	-77.9	80.7	11.9
	Polynomial Sol.						
	N=1	-86.5	.269	108.8	-186.4	168.7	6.8
	N=2	-58.7	-.212	124.9	-94.3	45.8	-6.9
	*N=3	-53.4	-.045	123.9	-76.9	88.4	-6.0
	Newton-Raphson						
	2 Iterations (Damped on 1st)	-41.9	-.321	103.0	-38.6	17.9	11.8
3	Inversion						
	N=1	69.9	.252	128.0	-6.3	16.5	-.7
	N=2	67.0	.286	126.7	-1.9	5.2	.3
	N=3	66.0	.294	126.8	-.5	2.8	.2
	Polynomial						
	N=1	69.9	.252	128.0	-6.3	16.5	-.7
	N=2	68.4	.308	125.9	-4.1	-1.7	.9
	N=3	65.0	.304	126.8	1.0	-.7	.2

Table 5-3. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		X	$\dot{X}$	t_F	% ERROR X	% ERROR $\dot{X}$	% ERROR t_F
	Newton-Raphson						
	2 Iterations	65.6	.303	126.7	.1	-.3	.3
4	Inversion						
	N=1	70.0	.280	118.0	-4.8	17.1	-3.1
	N=2	67.7	.310	114.6	-1.4	8.2	-.1
	N=3	67.0	.326	114.3	-.4	3.5	.0
	Polynomial Sol.						
	N=1	70.0	.280	118.0	-4.8	17.1	-3.1
	N=2	75.4	.326	111.1	-12.9	3.6	2.8
	N=3	63.9	.349	113.2	4.2	-3.2	1.0
	Newton-Raphson						
	2 Iterations	66.7	.337	114.0	.0	.5	.3
5	Inversion						
	N=1	-67.8	-.148	216.4	12.5	860.8	3.0
	N=2	-72.4	.017	234.1	6.5	10.8	-4.8
	N=3	-87.8	.191	219.1	-13.2	-883.7	1.8
	Polynomial Sol.						
	N=1	-67.8	-.148	216.4	12.5	860.8	3.0
	N=2	-79.0	.018	220.6	-1.8	8.0	1.1
	N=3	-78.4	.027	221.7	-1.1	-40.2	.6
	Newton-Raphson						
	2 Iterations	-80.6	.079	218.2	-3.9	-305.7	2.2
	Inversion						
	N=1	58.8	.155	215.8	14.0	7.2	-1.5
	N=2	74.7	.133	213.2	-9.1	20.4	-.3
	N=3	66.3	.194	211.2	3.1	-16.1	.6

Table 5-3. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
6	Polynomial Sol.						
	N=1	58.8	.155	215.8	14.0	7.2	-1.5
	N=2	67.2	.166	212.5	1.8	.5	.0
	N=3	68.4	.166	212.3	.0	.4	.0
	Newton-Raphson						
	2 Iterations	68.0	.170	211.1	.6	-1.5	.6
7	Inversion						
	N=1	46.4	.108	246.4	29.6	26.9	-3.7
	N=2	-88.5	-.067	240.9	234.0	145.1	-1.3
	N=3	57.8	.260	228.9	12.3	-75.3	3.6
	Polynomial Sol.						
	N=1	46.4	.108	246.4	29.6	26.9	-3.7
	N=2	62.6	.145	238.4	5.1	2.2	-.3
	N=3	65.8	.148	237.1	.3	.3	.2
	Newton-Raphson						
	2 Iterations (Damped on 1st)	66.1	.154	232.3	-.1	-3.5	2.2
8	Inversion						
	N=1	-70.4	-.075	193.8	5.4	-7265.3	3.8
	N=2	-67.7	-.092	206.9	9.0	-9007.4	-2.6
	N=3	-78.2	.070	201.5	-5.0	6960.8	.0
	Polynomial Sol.						
	N=1	-70.4	-.075	193.8	5.4	-7265.3	3.8
	N=2	-75.8	.006	199.6	-1.8	689.5	.9
	N=3	-75.2	.007	201.2	-.9	784.7	.1
	Newton-Raphson						
	2 Iterations	-75.3	.002	201.1	-1.1	2037.3	.2

Table 5-3. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
9	Inversion						
	N=1	72.7	.380	129.9	8.4	20.4	.0
	N=2	76.4	.448	129.9	3.7	5.9	.0
	N=3	78.1	.467	129.9	1.5	1.9	.0
	Polynomial Sol.						
	N=1	72.7	.380	129.9	8.4	20.4	.0
	N=2	85.9	.435	137.4	-8.2	8.6	-5.7
	*N=3	74.6	.474	124.7	6.0	.4	4.0
	Newton-Raphson						
	2 Iterations	78.8	.471	128.5	.7	1.2	1.0
10	Inversion						
	N=1	69.1	.373	121.0	7.9	31.9	-1.0
	N=2	72.0	.485	119.8	4.0	11.5	-.0
	N=3	73.4	.519	119.9	2.1	5.3	-.1
	Polynomial Sol.						
	N=1	69.1	.373	121.0	7.9	31.9	-1.0
	*N=2	87.4	.479	131.6	-16.4	12.6	-9.9
	*N=3	45.8	.512	79.9	38.9	6.5	33.2
	Newton-Raphson						
	2 Iterations	74.0	.533	117.4	1.5	2.7	1.9
11	Inversion						
	N=1	40.8	.178	144.6	-2.1	-14.6	13.8
	N=2	27.4	.064	161.5	31.4	58.5	3.8
	N=3	32.3	.097	171.8	19.1	37.3	-2.3
	Polynomial Sol.						
	N=1	40.8	.178	144.6	-2.1	-14.6	13.8
	N=2	38.4	.159	157.2	3.8	-2.4	6.3
	N=3	39.8	.160	163.7	.4	-3.0	2.5
	Newton-Raphson						
	2 Iterations	39.1	.146	168.1	2.1	6.0	-.1

Table 5-3. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		χ	$\dot{\chi}$	t_F	% ERROR χ	% ERROR $\dot{\chi}$	% ERROR t_F
12	Inversion						
	N=1	43.8	-.492	132.3	-12.5	393.7	14.2
	N=2	29.8	.100	144.4	23.5	40.0	6.4
	N=3	29.2	.092	154.8	24.9	44.8	-.3
	Polynomial Sol.						
	N=1	43.8	-.492	132.3	-12.5	393.7	14.2
	N=2	39.2	.183	144.7	-.6	-9.1	6.2
	N=3	39.0	.173	151.4	-.4	-3.4	1.8
	Newton-Raphson						
	2 Iterations	35.7	.189	151.4	8.2	-12.5	1.8
13	Inversion						
	N=1	84.0	.086	232.5	-3.4	46.1	-1.7
	N=2	79.6	.207	229.2	2.0	-30.1	-.2
	N=3	81.9	.133	228.0	-.8	16.5	.2
	Polynomial Sol.						
	N=1	84.0	.086	232.5	-3.4	46.1	-1.7
	N=2	80.0	.156	229.2	1.4	1.6	-.2
	N=3	80.8	.161	228.5	.4	-.9	.0
	Newton-Raphson						
	2 Iterations	80.8	.162	225.8	.4	-1.8	1.2
14	Inversion						
	N=1	33.6	.081	232.0	37.9	46.8	-1.0
	N=2	77.8	-.247	243.0	-43.6	262.9	-5.8
	N=3	66.3	.289	216.5	-22.2	-90.6	5.7
	Polynomial Sol						
	N=1	33.6	.081	232.0	37.9	46.8	-1.0
	N=2	50.6	.150	227.9	6.6	1.7	.7
	N=3	54.2	.154	228.2	.0	-1.7	.6

Table 5-3. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Concluded)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
	Newton-Raphson						
	2 Iterations	53.6	.193	198.0	1.0	-27.5	13.7
15	Inversion						
	N=1	24.2	.008	265.7	54.8	94.0	-3.7
	N=2	-25.9	-.830	286.1	148.3	739.4	-11.7
	N=3	71.0	.431	207.0	-32.1	-232.0	19.1
	Polynomial Sol.						
	N=1	24.2	.008	265.7	54.8	94.0	-3.7
	N=2	46.7	.115	255.5	13.0	11.4	.1
	N=3	53.3	.131	253.9	.6	-1.2	.8
	Newton-Raphson						
	2 Iterations	57.1	.141	229.3	-6.3	-8.3	10.4
16	Inversion						
	N=1	84.2	.143	207.7	-2.7	20.1	-.6
	N=2	81.1	.195	206.7	.9	-8.7	-.1
	N=3	82.0	.175	206.2	-.1	2.2	.0
	Polynomial Sol.						
	N=1	84.2	.143	207.7	-2.7	20.1	-.6
	N=2	81.5	.178	206.5	.4	.3	-.0
	N=3	81.8	.179	206.4	.1	-.1	.0
	Newton-Raphson						
	2 Iterations	81.8	.180	205.9	.0	-.4	.2

χ and $\dot{\chi}$ for the 16 perturbations are tabulated along with the percentage errors. The inversion formulas of orders one, two, and three, along with the solutions of the first-, second-, and third-degree polynomials are given. In addition the results of a Newton-Raphson procedure at the end of two iterations are listed. Thus examples of second-order (inversion N=1 and polynomial solution N = 1 which are identical), third-order (inversion N = 2, polynomial solution N = 2), and fourth-order (inversion N = 3, polynomial solution N = 3, second iteration of Newton-Raphson) iteration functions have been compared. The few computations in which damping was used in the Newton-Raphson procedure are indicated. In these cases it should be noted that the damping obscures the true asymptotic convergence rate and a fourth-order classification of two Newton-Raphson iterations is not correct. The cases where the iteration on the polynomial equations did not converge are noted by an asterisk. The data listed are the results after the last iteration. The percentage errors are rounded to the nearest tenth percent. In some cases (e.g., cases 5 and 8) the true values of $\dot{\chi}$ are near zero (i.e., an order of magnitude less than the nominal values) and the corresponding percentage errors are very large. However, some of the values are very good estimates.

In Table 5-4 the derivatives of the guidance formulas were computed by updating corresponding derivatives from the reference trajectory (as described in subsection 3.2). The resulting χ and $\dot{\chi}$ are listed for each of the 16 cases. Here, only the N = 1 and N = 2 orders of the polynomial formulas are considered. The data in Tables 5-3 and 5-4 were obtained by using the initial point of the nominal trajectory as starting values. Comparing the values in Table 5-4 to the corresponding ones in Table 5-3 one observes that the results differ very little. One may conclude that the updated derivatives from the reference trajectory are sufficiently accurate while reducing the computations considerably.

Study of the tabulated errors in Table 5-3 clearly indicates the superiority of the polynomial solutions over the inversion formulas, computation time not considered. Quantitatively, for example, the percentage errors in the second degree polynomial solutions for χ exceeded 15 percent in only 2 cases out of 16 while the second-order inversion formulas exceeded 15 percent in 7 cases.

Table 5-4. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING NOMINAL STARTING VALUES

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
1	Polynomial Sol.						
	N=1	-80.8	.21	116.1	-158.5	159.0	9.7
	*N=2	18.4	-.05	153.7	158.9	85.7	-19.6
2	Polynomial Sol.						
	N=1	-86.6	.27	108.9	-187.0	169.1	6.8
	*N=2	-57.1	-.21	120.3	-89.4	47.5	-3.0
3	Polynomial Sol.						
	N=1	69.9	.27	128.1	-6.4	12.0	-.8
	N=2	68.9	.31	125.5	-5.0	-3.4	1.2
4	Polynomial Sol.						
	N=1	70.0	.28	118.1	-4.9	17.1	-3.2
	N=2	73.8	.32	111.7	-10.7	3.3	2.3
5	Polynomial Sol.						
	N=1	-67.7	-.15	216.3	12.6	872.0	3.1
	N=2	-79.0	-.02	220.5	-1.9	-14.1	1.2
6	Polynomial Sol.						
	N=1	59.0	.15	215.8	13.8	7.2	-1.5
	N=2	67.0	.17	212.3	2.1	1.0	0.0
7	Polynomial Sol.						
	N=1	46.6	.11	246.5	29.4	27.0	-3.7
	N=2	63.3	.15	237.5	4.1	1.0	0.0
8	Polynomial Sol.						
	N=1	-70.3	-.07	193.6	5.5	-5911.4	3.9
	N=2	-74.5	-.01	199.5	-.1	-542.4	1.0
9	Polynomial Sol.						
	N=1	72.7	.38	130.0	8.4	20.2	-.1
	N=2	87.0	.43	139.0	-9.6	9.2	-7.0

Table 5-4. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING NOMINAL STARTING VALUES (Concluded)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
10	Polynomial Sol.						
	N=1	68.9	.39	121.1	8.2	29.7	-1.2
	N=2	86.3	.47	129.6	-15.0	13.4	-8.3
11	Polynomial Sol.						
	N=1	40.8	.17	144.8	-2.1	-14.1	13.7
	N=2	39.2	.16	156.3	1.7	-7.3	6.9
12	Polynomial Sol.						
	N=1	43.8	.21	132.4	-12.5	-22.9	14.2
	N=2	39.3	.19	142.8	-1.1	-11.3	7.5
13	Polynomial Sol.						
	N=1	84.1	.09	232.4	-3.6	46.5	-1.7
	N=2	80.1	.16	229.0	1.4	.3	-.2
14	Polynomial Sol.						
	N=1	33.7	-.08	232.1	37.9	46.9	-1.1
	N=2	50.1	.15	227.8	7.7	3.6	.8
15	Polynomial Sol.						
	N=1	24.3	.008	265.8	54.8	93.9	-3.8
	N=2	48.2	.121	254.4	10.3	6.9	.6
16	Polynomial Sol.						
	N=1	84.3	.14	207.7	-2.9	21.1	-.6
	N=2	81.7	.18	206.6	.2	1.5	0.0

But even weighing in the total numerical computation one must favor the polynominal solutions. For example, a second-order inversion is equivalent (in computation time) to generating the corresponding second-degree polynomial and performing two Newton-Raphson iterations toward solving the equations. In all cases, three additional iterations were imposed. However, in many cases two iterations would have been sufficient. This fact may be determined by comparing the solution of the first degree polynomials (which is the result of the first iteration on all higher degree polynomials) with the solution of the second-degree polynomials. This coupled with the probability of improving the efficiency of the Newton-Raphson iteration (see Section VI for further comments) makes inversion and solution of the polynomial equations about equal in terms of computation time. But, most important is the question of convergence, i.e., the comparison of the tabulated percentage errors. There is no doubt that for the perturbations considered the performance of the polynomial solutions exceeds that of the inversion formulas.

One further comment on these comparisons needs to be made. The term "convergence" when applied to the polynomial equations has two different meanings. The solution of the polynomial equations represents one iteration of an interpolatory iteration function. In this sense, convergence was discussed in subsections 5.1 and 5.2. Furthermore, only one iteration is being considered. Secondly, convergence must be discussed when considering iterative techniques for solution of the polynomial equations at each step of the larger process. Suppose this iteration does not converge sufficiently within the number of iterations allowed? This question is critical in determining the usefulness of an interpolatory iteration. However, for the second degree polynomials only two cases did not converge. When this happens the logical move is to use the solution obtained from the first iteration. For this problem the results were very satisfactory. The question of convergence in the large is a very difficult one and a problem of this magnitude requires empirical verification.

Based on these results a guidance algorithm is presented in Section VI. Considering time and storage limitations, the second-degree interpolatory iteration was chosen.

As suggested in subsection 3.4 the expansion of Silber and Hunt was used as a refinement. Recall the starting values used in Table 5-4 were the nominal values, and were generally very poor, which can be seen by examining the errors in Table 5-2. Instead, the second-order expansion of Silber and Hunt was used to generate starting values. Then the second-degree polynomials were obtained with corrected derivatives from the nominal trajectory. Table 5-5 contains the results of this procedure under the heading "First Guidance Command". In comparing the percentage errors in χ and $\dot{\chi}$ of Table 5-5 to those of the second-order expansion of Silber and Hunt in Table 5-2 one sees that the error is reduced in every case. Even the solution of the linear polynomials (N = 1) reduces the error in all cases except one. Furthermore, it is verified that starting with the expansion of Silber and Hunt improves the performance of the guidance formulas over that by starting with reference values. Concerning the errors in t_f, it is observed that in most cases the error of the guidance formulas and the Silber-Hunt expansion is acceptable. However, in two cases (11 and 12) the error in the guidance formula was much greater than the corresponding Silber-Hunt error. For this reason it was decided to return the Silber-Hunt estimate of time-to-go on the first guidance cycle. In order to investigate the initial behavior of the guidance package a second guidance cycle was computed with no change in the initial state y. The results are tabulated in Table 5-5 under the heading "Second Guidance Command". This represents essentially two iterations of the interpolatory iteration function. All errors were driven to more than acceptable limits.

The results of the numerical study are compared qualitatively on the basis of accuracy and convergence in Table 5-6. The ratings are determined by computing a weighted percent error, by weighting χ and $\dot{\chi}$ two and t_f one. The reasoning here being that χ and $\dot{\chi}$ are of direct initial importance whereas the beginning values of t_f are merely indicators of future state and do not affect current action. It is clear from Table 5-6 that the use of the Silber-Hunt expansion greatly improves the convergence of the guidance formulas.

In summary the main conclusions of this numerical study are listed:

(A) The N = 1 polynomial solutions improved the Silber-Hunt approximations of χ and $\dot{\chi}$ in every case. The advantage of this combination of methods is obvious.

Table 5-5. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
1	Polynomial Sol.						
	First Guid. Com.						
	N=1	-33.7	-.37	122.7	-7.7	-4.5	4.6
	N=2	-34.4	-.35	123.5	-10.1	.8	4.0
	Second Guid. Com.						
	N=1	-31.5	-.36	127.8	-.6	-2.0	.6
	N=2	-31.7	-.36	127.9	-1.4	-1.7	.5
2	Polynomial Sol.						
	First Guid. Com.						
	N=1	-32.4	-.40	110.7	-73.8	-2.9	5.2
	N=2	-33.4	-.38	112.1	-10.7	1.7	4.0
	Second Guid. Com.						
	N=1	-30.5	-.40	116.0	-1.1	-1.2	.6
	N=2	-30.6	-.40	116.1	-1.4	-1.2	.6
3	Polynomial Sol.						
	First Guid. Com.						
	N=1	65.6	.30	127.2	.2	.0	.1
	N=2	65.6	.30	127.2	.1	.0	.1
	Second Guid. Com.						
	N=1	65.6	.30	127.1	.0	.0	.0
	N=2	65.6	.30	127.1	.0	.0	.0
4	Polynomial Sol.						
	First Guid. Com.						
	N=1	66.5	.34	114.6	.4	.0	-.2
	N=2	66.6	.34	114.6	.2	.0	-.2
	Second Guid. Com.						
	N=1	66.7	.34	114.4	.0	.0	.0
	N=2	66.7	.34	114.4	.0	.0	.0

Table 5-5. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES (Continued)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
5	Polynomial Sol.						
	First Guid. Com.						
	N=1	-76.8	-.002	227.7	.9	106.8	-2.0
	N=2	-77.0	-.002	227.7	.6	89.5	-2.0
	Second Guid. Com.						
	N=1	-77.7	-.01	223.5	-.2	.5	-.1
	N=2	-77.6	-.02	223.2	-.2	.0	-.0
6	Polynomial Sol.						
	First Guid. Com.						
	N=1	68.7	.17	211.6	-.3	.0	.4
	N=2	68.7	.17	212.3	-.4	.0	.1
	Second Guid. Com.						
	N=1	68.5	.17	212.5	.0	.0	.0
	N=2	68.5	.17	212.5	.0	.0	.0
7	Polynomial Sol.						
	First Guid. Com.						
	N=1	66.4	.15	234.3	-.6	.0	1.4
	N=2	66.5	.15	237.0	-.7	.0	.3
	Second Guid. Com.						
	N=1	66.2	.15	237.4	-.4	.0	.1
	N=2	66.2	.15	237.5	-.3	.0	.0
8	Polynomial Sol.						
	First Guid. Com.						
	N=1	-73.8	-.02	203.1	.9	-1564.8	-.7
	N=2	-74.0	-.02	202.9	.6	-1200.9	-.6
	Second Guid. Com.						
	N=1	-74.5	-.005	201.6	-.1	-272.6	.0
	N=2	-74.5	-.004	201.5	-.0	-217.2	.0

Table 5-5. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES (Continued)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
9	Polynomial Sol.						
	First Guid. Com.						
	N=1	79.5	.47	129.9	-.2	.7	.0
	N=2	79.7	.47	130.2	-.4	1.4	-.2
	Second Guid. Com.						
	N=1	79.4	.48	129.9	.0	.0	.0
	N=2	79.4	.48	129.9	.0	.0	.0
10	Polynomial Sol.						
	First Guid. Com.						
	N=1	75.3	.54	119.3	-.3	1.4	.3
	N=2	75.5	.54	119.9	-.6	1.6	-.1
	Second Guid. Com.						
	N=1	75.1	.55	119.7	.0	.2	.0
	N=2	75.1	.55	119.7	.0	.2	.0
11	Polynomial Sol.						
	First Guid. Com.						
	N=1	47.3	.13	214.9	-18.6	14.8	-28.0
	*N=2	14.2	-.07	322.2	64.5	146.1	-92.0
	Second Guid. Com.						
	N=1	34.7	.13	157.8	13.0	16.3	6.0
	N=2	37.2	.15	164.9	6.7	7.0	1.7
12	Polynomial Sol.						
	First Guid. Com.						
	N=1	49.1	.15	204.6	-26.3	13.4	-32.6
	*N=2	34.5	.08	246.2	11.2	50.9	-59.6
	Second Guid. Com.						
	N=1	32.2	.12	143.2	17.4	26.2	7.2
	N=2	35.2	.15	151.1	9.6	12.3	2.1

Table 5-5. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES (Concluded)

CASE		x	$\dot{x}$	t_F	% ERROR x	% ERROR $\dot{x}$	% ERROR t_F
	Polynomial Sol.						
	First Guid. Com.						
	N=1	81.3	.16	228.7	-.2	.3	.0
13	N=2	81.3	.16	228.7	-.1	.3	.0
	Second Guid. Com.						
	N=1	81.2	.16	228.7	.0	.0	.0
	N=2	81.2	.16	228.7	.0	.0	.0
	Polynomial Sol.						
	First Guid. Com.						
	N=1	51.6	.14	225.7	4.8	8.7	1.7
14	N=2	52.9	.14	234.1	2.5	5.5	-1.9
	Second Guid. Com.						
	N=1	53.7	.15	231.1	1.0	1.0	-.6
	N=2	53.8	.15	231.1	.8	.8	-.6
	Polynomial Sol.						
	First Guid. Com.						
	N=1	50.6	.10	246.5	5.8	22.6	3.7
15	N=2	52.4	.10	262.9	2.4	15.7	-2.7
	Second Guid. Com.						
	N=1	51.7	.13	257.2	3.8	4.1	-.5
	N=2	52.0	.13	258.8	3.3	3.1	-1.1
	Polynomial Sol.						
	First Guid. Com.						
	N=1	81.9	.18	206.4	.0	.0	.0
16	N=2	81.9	.18	206.4	.0	.0	.0
	Second Guid. Com.						
	N=1	81.9	.18	206.4	.0	.0	.0
	N=2	81.9	.18	206.4	.0	.0	.0

Table 5-6. QUALITATIVE COMPARISON OF THE GUIDANCE FORMULAS

	RATINGS A	B	C
Nominal	0	0	16
Silber-Hunt Expansion			
N=1	3	8	5
N=2	4	8	4
Inversion (Integrated derivatives, Nom. starting values)			
N=1	0	9	7
N=2	4	6	6
N=3	5	3	7
Polynomial (Integrated derivatives, Nom. starting values)			
N=1	0	9	7
N=2	10	4	2
N=3	13	0	3
Damped Newton-Raphson (2 Iterations)	11	4	1
Polynomial (Corrected nominal derivatives, Nom. starting values)			
N=1	0	9	7
N=2	10	4	2
Polynomial (Corrected nominal derivatives, S-H starting values)			
N=1	10	3	3
N=2	12	2	2
Polynomial (Second Guidance Command)			
N=1	14	1	1
N=2	14	2	0

E = weighted error magnitude

	$E \le 5\%$	A
$5\% <$	$E \le 15\%$	B
	$E > 15\%$	C

(B) The polynomial solutions were obviously more effective than the inversion formulas.

(C) The N = 2 polynomial solution on the first guidance command gave little improvement over the N = 1 polynomials. In cases 11 and 12 the iteration on the N = 2 polynomials did not converge.

(D) The N = 2 polynomial solution gave significant improvement over the N = 1 polynomial solution on the second guidance command in cases 11 and 12 where it was most needed. However, in view of conclusion (C) the use of N = 2 is held open until further investigation.

(E) The Silber-Hunt second-order terms improved the first-order terms more than is implied by Table 5-6, especially for t_f. However, the N = 2 terms failed to improve the N = 1 terms in a few cases (notably cases 11 and 12) and even deteriorated the N = 1 estimates. Considering the additional computing time and storage requirements for N = 2, the advisability of using it for calculating approximate multipliers is open to question.

Section VI

DEFINITION OF A GUIDANCE PACKAGE

The content of this section describes a guidance routine based on the polynomial solutions discussed earlier. However, in order that this section may be complete and used as a reference all the necessary equations are repeated. The test mission will be a minimum fuel, constant burn injection into a specified circular orbit. For simplicity, the space dimensionality will be constrained to two.

6.1 A BRIEF SUMMARY

The problem may be simply stated, 'Given the state of the vehicle Y, return the thrust direction and its time derivative (i.e., χ and $\dot{\chi}$, see Figure 6-1) such that the optimality criterion (minimum payload) is satisfied'. In general, χ and $\dot{\chi}$ are calculated from the solution of a set of simultaneous polynomial equations whose coefficients are approximated from a nominal trajectory. Currently a 'time-to-go' criterion is used to select a point from the nominal trajectory.

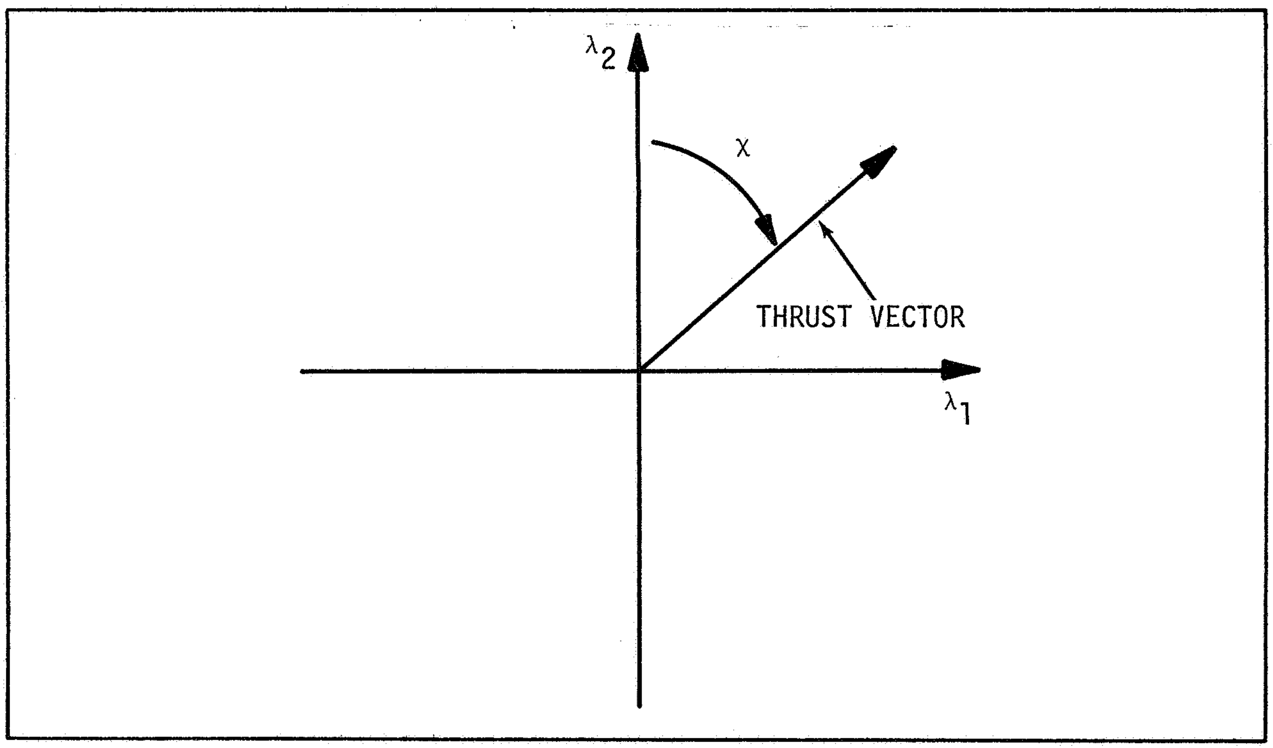

Figure 6-1. DEFINITION OF THRUST ANGLE χ

The guidance routine is required to return a χ and $\dot{\chi}$ for a set of state vectors proceeding along some path. Thus a sequence of related problems must be solved by the routine. Each problem needs an initial approximation to the solution of the polynomial equations. Here, the solution of the previous problem equations with a linear update is used as the approximation. On the very first call to the routine the second-order expansion of Silber and Hunt (ref. 3) is used to obtain the approximation.

The solution to the polynomial equations begin to degenerate when the time-to-go becomes small because many of the coefficients are approaching zero. Thus, when the time-to-go becomes less than some prespecified value (ΔT_{HOM}) the control laws are assumed linear and a simple update of χ and $\dot{\chi}$ is performed.

A set of five second-degree polynomials are used. The coefficients of the linear and second-order terms are obtained from simple expansions about the nominal trajectory as mentioned before. The constant terms are calculated by numerical integration. The accurate evaluation of these terms gives the guidance routine its self-correcting behavior as well as a measure of its success.

In the following, ξ represents the vector of multipliers and time-to-go associated with the state Y.

6.2 STATEMENT OF COMPUTATIONAL ALGORITHM

A concise description of the calculation procedure is contained here. The notation is defined followed by a general flow chart (Figure 6-2) and algorithmic description anoted by a detailed explanation.

6.2.1 Definition of Symbols and Notation

X	cartesian position vector $(x_1, x_2)^t$
$\dot{X}$	cartesian velocity vector $(\dot{x}_1, \dot{x}_2)^t$
$\frac{F}{m}$	thrust to mass ratio

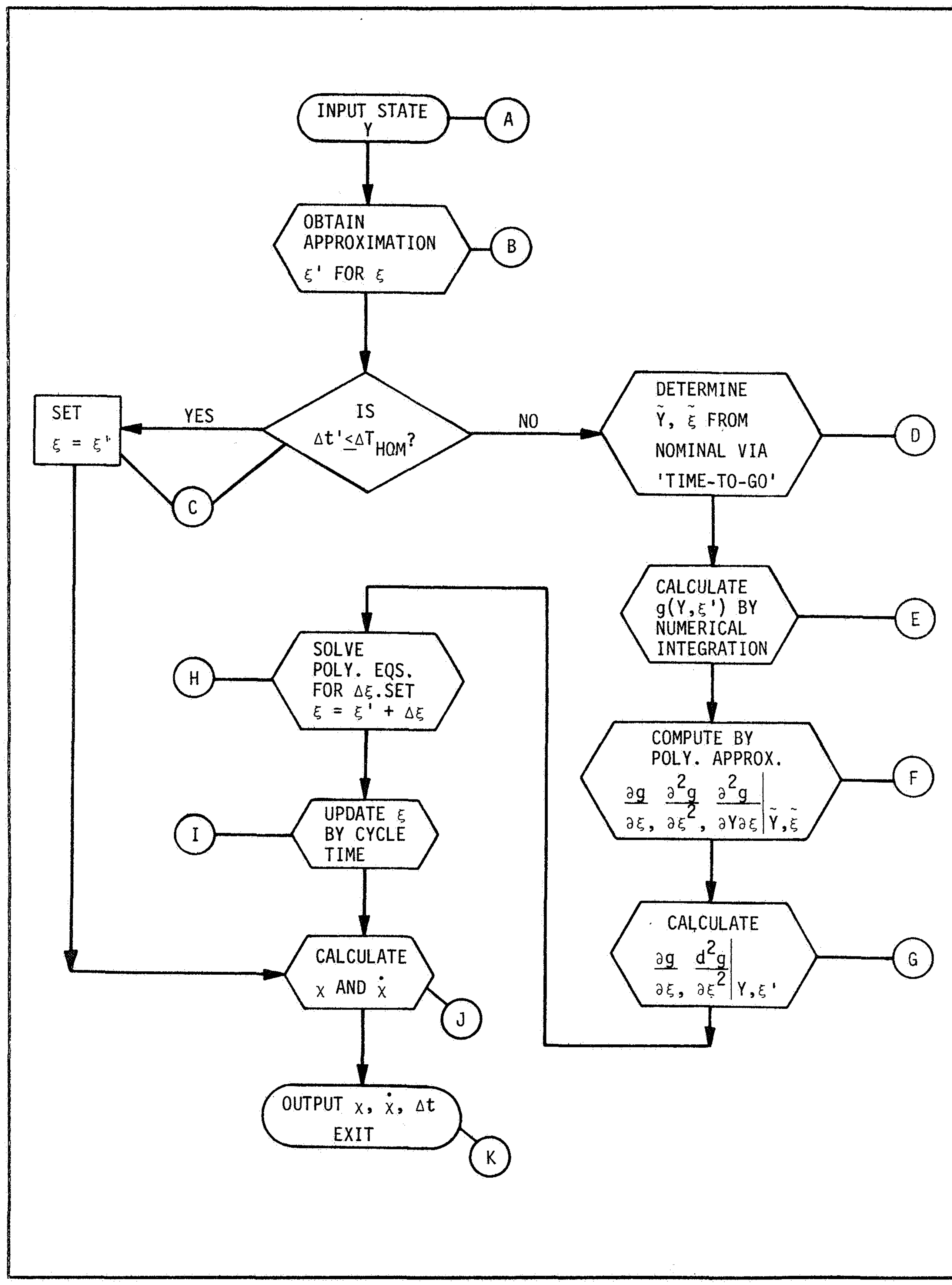

Figure 6-2. FLOWCHART FOR GUIDANCE PACKAGE

$\frac{\beta}{m}$	mass flow rate magnitude to mass ratio
Y	initial state vector = $\left(X, \dot{X}, \frac{F}{m}, \frac{\beta}{m}\right)^t$
Δt	time-to-go
λ	vector of Lagrangian multipliers $(\lambda_1, \lambda_2)^t$ associated with X
$\dot{\lambda}$	vector of negated Lagrangian multipliers $(\dot{\lambda}_1, \dot{\lambda}_2)^t$ associated with $\dot{X}$
ξ	vector of unknown quantities = $(\lambda_o, \dot{\lambda}_o, \Delta t)^t$ *
ΔT_{ex}	External cycle time of guidance package, i.e. the elapsed time between exit from the guidance routine and upon entry again
ΔT_{IN}	Internal cycle time, i.e. the elapsed time between entry into guidance routine and exit
ΔT_{SWCH}	Trajectory constant which indicates when the derivatives are calculated by linear interpolation.
ΔT_{NOM}	Initial value of nominal Δt
ΔT_{HOM}	Trajectory constant which indicates when a simple guidance law will be invoked.
T	Time on nominal trajectory from which $\tilde{Y}$, $\tilde{\xi}$ has been obtained
R_{co}, V_{co}	Radius and velocity at cutoff; input constants.

The values Y and ξ represent the true state of the vehicle and the solution obtained from the polynomials, respectively. A "prime" adjoined to ξ indicates estimates for ξ. Quantities from the nominal trajectory corresponding to Y are denoted $\tilde{Y}$ and $\tilde{\xi}$.

6.2.2 General Description of Flow Chart for Guidance Package

A. Input Y = $\left(X, \dot{X}, \frac{F}{m_o}, \frac{\beta}{m_o}\right)^t$; request for guidance command. The vector Y designates the current true state of the vehicle.

* *The subscripts "o" and "f" will designate the quantity at the initial time and the final time respectively.*

B. If this is the first guidance request initialize $\tilde{Y}$, $\tilde{\xi}$ from first point of nominal. Set T = 0 and then use Silber's expansion to obtain the approximate ξ' (see detailed descriptions). Proceed to step C.

If this is not the first guidance request then obtain $\xi'$from the previous calculated ξ as follows:

$$\lambda' = \lambda + \Delta T_{ex} \dot{\lambda}$$

$$\lambda' = \dot{\lambda} + \Delta T_{ex} \ddot{\lambda}$$

$$\Delta t' = \Delta t - \Delta T_{ex}$$

C. Test to see if the estimated time-to-go $\Delta t'$ is less than or equal to ΔT_{HOM}. In other words the question to be answered 'Is it close enough to cutoff time so that a simple linear guidance may be used?' The degree of closeness is indicated by the input constant ΔT_{HOM}. If the answer is yes than ξ' is an accurate estimate of ξ thus set $\xi = \xi'$ and proceed to step J.

D. At this step it is desired to obtain a $\tilde{Y}$, $\tilde{\xi}$ with the same time-to-go as ξ' or simply $\tilde{\Delta t} = \Delta t'$. This is accomplished by integrating the MEL equations with the initial conditions $\tilde{Y}$, $\tilde{\xi}$ (from last guidance call) from t = T to t = $\Delta T_{NOM} - \Delta t'$. Then reset $\tilde{Y}$, $\tilde{\xi}$ from the final values of the integration. Also set $\tilde{\Delta t} = \tilde{\Delta t} - (\Delta T_{NOM} - \Delta t')$ and $T = \Delta T_{NOM} - \Delta t'$. The numerical integration here will probably be over a small interval. Based on numerical studies on this nominal trajectory (ref. 12) it seems desirable to use only one integration step with a fifth-order Runge-Kutta (See detailed description).

E. Integrate the MEL equations with the initial conditions Y, ξ' from t = 0 to t = $\Delta t'$. Then compute $g(Y, \xi')$ from the initial and final values of the integration (see detailed description).

Based on numerical studies of perturbations of the nominal trajectory it seems desirable to use 3 integration steps for this case (ref. 12).

F. Obtain derivatives

$$\frac{\partial g}{\partial \xi}(\tilde{Y}, \tilde{\xi}), \quad \frac{\partial^2 g}{\partial \xi^2}(\tilde{Y}, \tilde{\xi}), \text{ and } \frac{\partial^2 g}{\partial y \partial \xi}(\tilde{Y}, \tilde{\xi})$$

from nominal trajectory via least square polynomials. Evaluate polynomials with t = T. (See detailed description.)

G. Calculate

$$\frac{\partial g}{\partial \xi}(Y, \xi') = \frac{\partial g}{\partial \xi}(\tilde{Y}, \tilde{\xi}) + \frac{\partial^2 g}{\partial Y \partial \xi}(\tilde{Y}, \tilde{\xi})(Y - \tilde{Y}) + \frac{\partial^2 g}{\partial \xi^2}(\tilde{Y}, \tilde{\xi})(\xi' - \tilde{\xi})$$

$$\frac{\partial^2 g}{\partial \xi^2}(Y, \xi') = \frac{\partial^2 g}{\partial \xi^2}(\tilde{Y}, \tilde{\xi})$$

(See detailed description)

H. Solve the following polynomials for $\Delta\xi$, by Newton-Raphson iteration:

$$0 = g(Y, \xi') + \frac{\partial g}{\partial \xi}(Y, \xi')\,\Delta\xi + \frac{\partial^2 g}{\partial \xi^2}(Y, \xi')\,\frac{\Delta\xi^2}{2}$$

Calculate $\xi = \xi' + \Delta\xi$

(See detailed descriptions.)

I. With ΔT_{IN} being the cycle time of guidance package update ξ by the same formulas of step B, i.e.

$$\lambda = \lambda + \Delta T_{IN}\,\dot{\lambda}$$

$$\lambda = \dot{\lambda} + \Delta T_{IN}\,\ddot{\lambda}$$

$$\Delta t = \Delta t - \Delta T_{IN}$$

J. Calculate

$$\chi = \text{Arctan}\,\frac{\lambda_1}{\lambda_2}$$

$$\dot{\chi} = \frac{\lambda_2\dot{\lambda}_1 - \lambda_1\dot{\lambda}_2}{\lambda_1^2 + \lambda_2^2}$$

K. Output χ, $\dot{\chi}$, Δt and exit.

6.2.3 Detailed Descriptions

The letters in parenthesis refer to those in subsection 6.2.2. Some items are sufficiently described in subsection 6.2.2 and no further description is required.

(B) Expansion of Silber and Hunt - On the first guidance request the expansion of Silber and Hunt will be used to give a 'crack', ξ', at the unknown quantities ξ. These expressions are in the form

$$\xi_i' = \tilde{\xi}_i + \sum_{j=1}^{6} \frac{\partial \xi_i}{\partial Y_i} (\tilde{Y}, \tilde{\xi}) \Delta Y_j + \frac{1}{2} \sum_{j=1}^{6} \sum_{k=1}^{6} \frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k} (\tilde{Y}, \tilde{\xi}) \Delta Y_j \Delta Y_k \qquad (6\text{-}1)$$

where $\Delta Y_j = Y_j - \tilde{Y}_j$ and

$$\frac{\partial \xi_i}{\partial Y_j}, \frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k}$$

are stored constants. Taking advantage of the fact that

$$\frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k} = \frac{\partial^2 \xi_i}{\partial Y_k \partial Y_j},$$

then

$$\sum_{j=1}^{6} \sum_{k=1}^{6} \frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k} \Delta Y_j \Delta Y_k = \sum_{j=1}^{6} \frac{\partial^2 \xi_i}{\partial Y_j^2} \Delta Y_j^2 + 2 \sum_{j=2}^{6} \sum_{k=1}^{j-1} \frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k} \Delta Y_j \Delta Y_k$$

Thus equation (6-1) may be written as

$$\xi_i' = \tilde{\xi}_i + \sum_{j=1}^{6} \left[\frac{\partial \xi_i}{\partial Y_j} + \frac{\partial^2 \xi_i}{\partial Y_j^2} \frac{\Delta Y_j}{2} \right] \Delta Y_j + \sum_{j=2}^{6} \sum_{k=1}^{j-1} \frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k} \Delta Y_j \Delta Y_k \qquad (6\text{-}2)$$

Then it is necessary to store the constants

		No.
$\frac{\partial \xi_i}{\partial Y_j} (\tilde{Y}, \tilde{\xi})$	i=1,...5; j=1,...6	30
$\frac{\partial^2 \xi_i}{\partial Y_j^2} (\tilde{Y}, \tilde{\xi})$	i=1,...5; j=1,...6	30
$\frac{\partial^2 \xi_i}{\partial Y_j \partial Y_k} (\tilde{Y}, \tilde{\xi})$	i=1,...5; j=2,...6; k=1,...j-1	75
	TOTAL	135

(D), (E) Integration Package - This should be a separate subroutine which integrates the MEL equations.

$$\ddot{X} = \frac{F}{m\Lambda} \lambda - \frac{\mu}{R^3} X$$

$$\ddot{\lambda} = \frac{\mu}{R^3} \left(- \lambda + \frac{3\lambda \cdot X}{R^2} X \right)$$

where $\Lambda = (\lambda \cdot \lambda)^{1/2}$, $R = (X \cdot X)^{1/2}$, $\dot{m} = -\beta$, and μ, F, β, are constants. More compactly written

$$\ddot{p} = f(t, p) \; .$$

Input to this routine should be the initial values of X, $\dot{X}$, λ, $\dot{\lambda}$, $\frac{F}{m}$, $\frac{\beta}{m}$, t_o (initial time), t_f (final time), and NSTEP (number of integration steps). Output should be the final values of X, $\dot{X}$, λ, $\dot{\lambda}$, $\frac{F}{m}$, and $\frac{\beta}{m}$.

The integration formula is a Runge-Kutta 5-3 formula (fifth-order approximate with 3 evaluations of f(t, p) required) where h is the integration stepsize. This formula was derived by Andrus in reference 12.

$$K_1 = hf(t, p)$$

$$K_2 = hf\left(t + \frac{h}{5}, p + \frac{h}{5}\dot{p} + \frac{h}{50} K_1\right)$$

$$K_3 = hf\left(t + \frac{2}{3} h, p + \frac{2}{3} h\dot{p} - \frac{h}{27} K_1 + \frac{7}{27} h K_2\right)$$

$$p(t+h) = p + h\dot{p} + \frac{h}{168} (7 K_1 + 50 K_2 + 27 K_3)$$

$$K_4 = hf(t + h, p(t + h))$$

$$p(t+h) = \dot{p} + \frac{1}{336} (14 K_1 + 125 K_2 + 162 K_3 + 35 K_4)$$

It should be noted that this formula requires 4 evaluations of f on the first integration step but only three thereafter since K_4 at step n is K_1 at step n+1. The routine should flow as shown in Figure 6-3.

(E) Evaluation of $g(Y,\xi')$ - When injecting into a circular orbit in two dimensions one must satisfy the following conditions

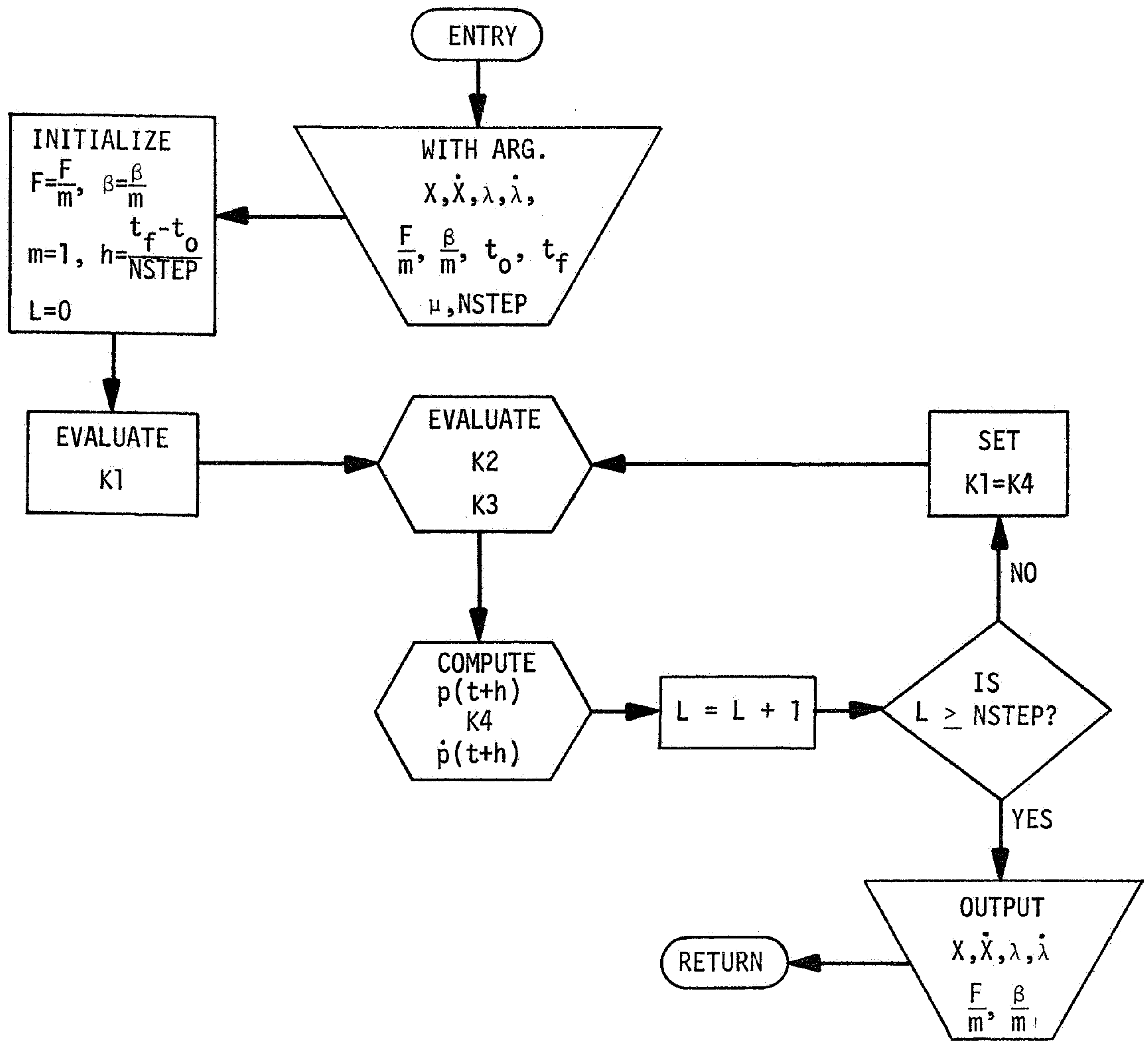

Figure 6-3. FLOWCHART FOR RK 5-3

$$g_1(Y, \xi') = X_{1f}^2 + X_{2f}^2 - R_{co}^2 = 0$$

$$g_2(Y, \xi') = \dot{X}_{1f}^2 + \dot{X}_{2f}^2 - V_{co}^2 = 0$$

$$g_3(Y, \xi') = X_{1f}\dot{X}_{1f} + X_{2f}\dot{X}_{2f} = 0$$

$$g_4(Y, \xi') = \lambda_{10}'^2 + \lambda_{20}'^2 - a = 0$$

$$g_5(Y, \xi') = \lambda_{10}'\dot{X}_{20} - \lambda_{20}'\dot{X}_{10} - \dot{\lambda}_{10}'X_{20} + \dot{\lambda}_{20}'X_{10} = 0$$

where $a = \tilde{\lambda}_1^2 + \tilde{\lambda}_2^2$

Notice that the last two g's are evaluated at the initial time.

(F), (G) The Derivatives - Since the last two g's are functions of variables at the initial time then their partial derivatives with respect to the ξ'_j are easily calculated.

$$\frac{\partial g_4}{\partial \xi_1} = 2\lambda_{10}', \quad \frac{\partial g_4}{\partial \xi_2} = 2\lambda_{20}', \quad \frac{\partial g_4}{\partial \xi_j} = 0, j = 3, 4, 5.$$

$$\frac{\partial g_5}{\partial \xi_1} = \dot{X}_{20}, \quad \frac{\partial g_5}{\partial \xi_2} = -\dot{X}_{10}, \quad \frac{\partial g_5}{\partial \xi_3} = -X_{20}, \quad \frac{\partial g_5}{\partial \xi_4} = X_{10}, \quad \frac{\partial g_5}{\partial \xi_5} = 0.$$

$$\frac{\partial^2 g_4}{\partial \xi_1^2} = 2, \quad \frac{\partial^2 g_4}{\partial \xi_2^2} = 2, \quad \frac{\partial^2 g_4}{\partial \xi_i \partial \xi_j} = 0 \qquad i=1,\ldots 5,\ j=1,\ldots 5 \text{ except as noted.}$$

Also

$$\frac{\partial^2 g_5}{\partial \xi_i \partial \xi_j} = 0 \qquad i=1,\ldots 5;\ j=1,\ldots 5.$$

It can be shown that

$$\frac{\partial g_i}{\partial \xi_5} = \dot{g}_i \qquad i=1, 2, 3$$

$$\frac{\partial^2 g_i}{\partial \xi_5^2} = \ddot{g}_i \qquad i=1, 2, 3.$$

which may be obtained directly from the results of the numerical integration in step (E). Thus, the derivatives that need to be approximated from the reference (nominal) trajectory are

	TOTAL
$\frac{\partial^2 g_i}{\partial \xi_j}(Y, \xi')$; $i = 1, 2, 3$; $j = 1, 2, 3, 4$	12
$\frac{\partial^2 g_i}{\partial \xi_j \partial \xi_k}(Y, \xi')$; $i = 1, 2, 3$; $j = 1, 2, 3, 4, 5$; $k = 1, \ldots, j$; j and k both not simultaneously 5	42*

These derivatives are approximated by expanding about the nominal trajectory and truncating

$$\frac{\partial g_i}{\partial \xi_j} \doteq \frac{\partial g_i}{\partial \xi_j}(\tilde{Y}, \tilde{\xi}) + \sum_{k=1}^{6} \frac{\partial^2 g_i}{\partial y_k \partial \xi_j}(\tilde{Y}, \tilde{\xi})\Delta Y_k + \sum_{k=1}^{5} \frac{\partial^2 g_i}{\partial \xi_k \partial \xi_j}(\tilde{Y}, \tilde{\xi})(\xi'_k - \tilde{\xi}_k)$$

$$\frac{\partial^2 g_i}{\partial \xi_j \partial \xi_k}(Y, \xi') = \frac{\partial^2 g_i}{\partial \xi_j \partial \xi_k}(\tilde{Y}, \tilde{\xi})$$

where $\Delta Y_k = Y_k - \tilde{Y}_k$.

---

* *Since* $\frac{\partial^2 g_i}{\partial \xi_j \partial \xi_k} = \frac{\partial^2 g_i}{\partial \xi_k \partial \xi_j}$

Thus it is necessary to have stored

		TOTAL
$\frac{\partial g_i}{\partial \xi_j}(\tilde{Y}, \tilde{\xi})$	$i = 1, 2, 3;\ j = 1, 2, 3, 4$	12
$\frac{\partial^2 g_i}{\partial y_k \partial \xi_j}(\tilde{Y}, \tilde{\xi})$	$i = 1, 2, 3;\ j = 1, 2, 3, 4;\ k = 1, 2, 3, 4, 5, 6$	72
$\frac{\partial^2 g_i}{\partial \xi_j \partial \xi_k}(\tilde{Y}, \tilde{\xi})$	$i = 1, 2, 3;\ j = 1, 2, \ldots, 5;\ k = 1, \ldots, j$	42
	TOTAL	126

These derivatives will be stored in the form of third-degree polynomials in the variable t (time on the nominal, $t = \Delta t_{NOM} - \Delta t$).

(F) Interpolation for Derivatives - Due to the fact that all the derivatives

$$\frac{\partial g_i}{\partial \xi_j}(\tilde{Y}, \tilde{\xi}) \qquad i = 1, 2, 3; \quad j = 1, 2, 3, 4$$

are zero when $\Delta t = 0$ then their values are fairly well approximated by linear interpolation for small Δt. When $\Delta t' \leq \Delta T_{SWCH}$ then the time-to-go is "small" and linear interpolation is used, otherwise the third-degree polynomials are used.

There will be 126 polynomials as noted earlier designated say

$$p_i(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3 \qquad i = 1, \ldots, 126$$

These should be evaluated using Horner's method, i.e.

$$p_i(t) = a_{i0} + t[a_{i1} + t(a_{i2} + a_{i3}t)] \tag{6-3}$$

The first time that $\Delta t' \leq \Delta T_{SWCH}$ then the derivatives are calculated by equation (6-3) and at the same time set $a_{i0} = \frac{Pi}{\gamma}$ $i = 1, \ldots 126$. where $\gamma = \Delta t'$. Then thereafter, calculate

$$p_i(t) = \Delta t' \, a_{i0} \, . \tag{6-4}$$

The following flow diagram (Figure 6-4) describes the computations of step (F):

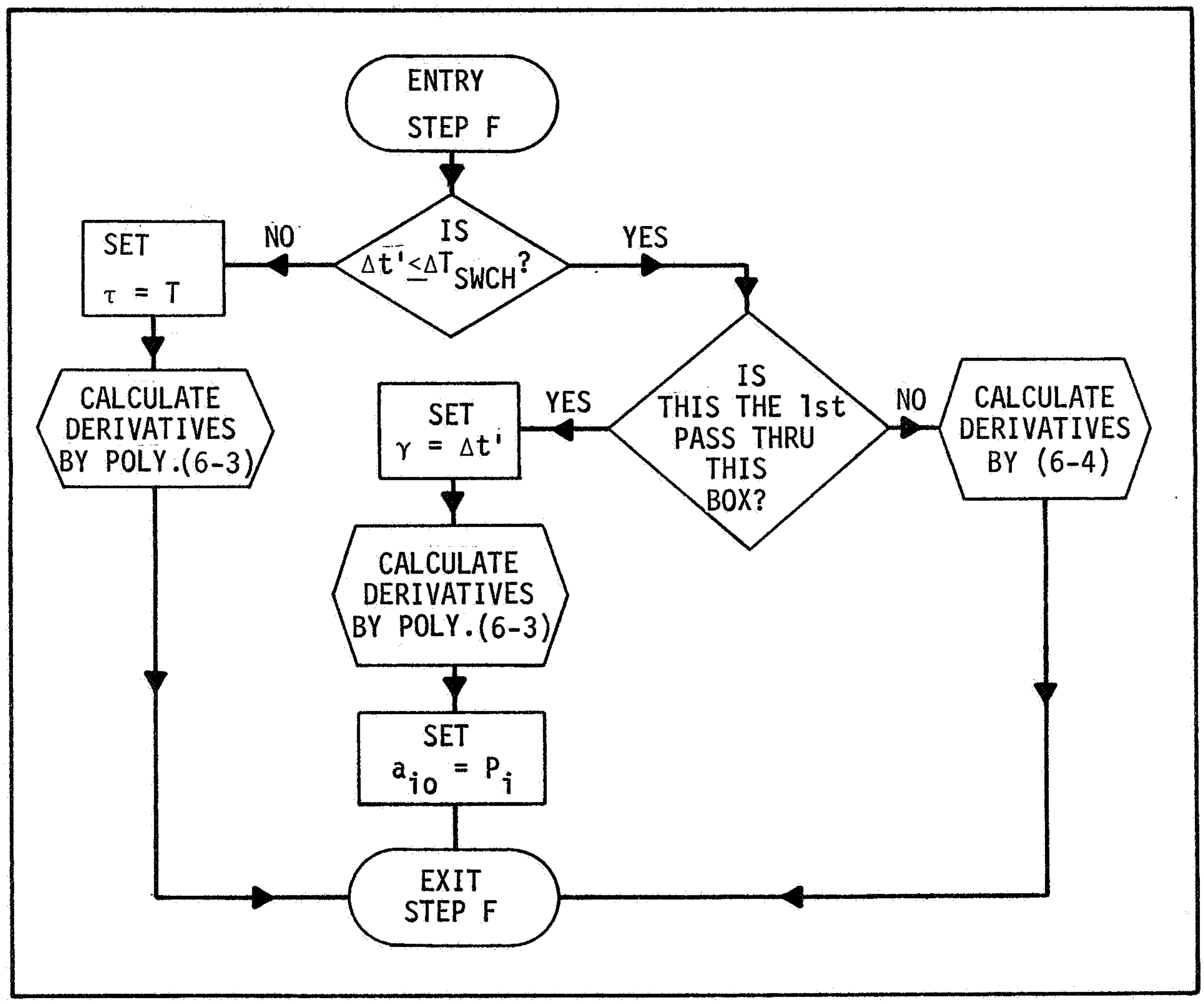

Figure 6-4. FLOWCHART FOR STEP F

(H) Solution of Polynomials - Given the true state Y it is desired to find the unknowns ξ such that the boundary conditions are zero, i.e.

$$g_i(Y, \xi) = 0 \qquad i = 1, \ldots, 5$$

Expanding in a Taylor series about ξ' and truncating after second-order terms gives

$$g_i(Y, \xi) \doteq g_i(Y, \xi') + \sum_{k=1}^{5} \frac{\partial g_i}{\partial \xi_k}(Y, \xi')\, \Delta\xi_k$$

$$+ \frac{1}{2} \sum_{k=1}^{5} \sum_{L=1}^{\xi} \frac{\partial^2 g_i}{\partial \xi_k \partial \xi_L}(Y, \xi')\, \Delta\xi_k \Delta\xi_L = 0 \qquad i = 1, 2, \ldots, 5$$

where

$$\Delta\xi_k = \xi_k - \xi'_k \, .$$

The coefficients of the polynomials are computed in steps (E) and (G). There are 5 polynomial equations in 5 unknowns $\Delta\xi_k$, $k = 1, \ldots, 5$. The object here is to solve the polynomials by Newton's iteration. To rewrite the polynomials in a nicer form

$$p_i(S) = a^{(i)} + \sum_{k=1}^{5} b_k^{(i)} S_k + \frac{1}{2} \sum_{k=1}^{5} \sum_{L=1}^{5} c_{kL}^{(i)} S_k S_L \qquad i = 1, \ldots, 5 \qquad (6\text{-}5)$$

where

$$S_k = \Delta\xi_k \, .$$

Since

$$\frac{\partial^2 g_i}{\partial \xi_k \partial \xi_L} = \frac{\partial^2 g_i}{\partial \xi_L \partial \xi_k}$$

then

$$c_{kL}^{(i)} = c_{Lk}^{(i)} .$$

Then one might expect to save a few operations by taking advantage of the symmetry property. Consider

$$\sum_{k=1}^{5} \sum_{L=1}^{5} c_{kL}^{(i)} S_k S_L = \sum_{k=1}^{5} c_{kk}^{(i)} S_k^2 + \sum_{\substack{k=1 \\ k \neq L}}^{5} \sum_{L=1}^{5} c_{kL}^{(i)} S_k S_L$$

$$= \sum_{k=1}^{5} c_{kk}^{(i)} S_k^2 + \sum_{k=2}^{5} \sum_{L=1}^{k-1} c_{kL}^{(i)} S_k S_L + \sum_{L=2}^{5} \sum_{k=1}^{L-1} c_{kL}^{(i)} S_k S_L$$

$$= \sum_{k=1}^{5} c_{kk}^{(i)} S_k^2 + 2 \sum_{k=2}^{5} \sum_{L=1}^{k-1} c_{kL}^{(i)} S_k S_L$$

Thus, instead of having 25 summations there are only 15. Then the polynomials may be written

$$p_i(S) = a^{(i)} + \sum_{k=1}^{5} \left(b_k^{(i)} + \frac{1}{2} c_{kk}^{(i)} S_k \right) S_k + \sum_{k=2}^{5} \sum_{L=1}^{k-1} c_{kL}^{(i)} S_k S_L \qquad (6-6)$$

In the Newton iteration an initial guess is made at S_k, say $S_k^{(o)}$; then $p_i[S^{(o)}]$ is evaluated. Next the partial derivatives are required

$$\frac{\partial p_i}{\partial s_j} [S^{(o)}] \; .$$

Differentiating equation (6-5) wrt S_j obtain

$$\frac{\partial p_i}{\partial s_j} (S) = b_j^{(i)} + \frac{1}{2} \sum_{k=1}^{5} c_{kj}^{(i)} S_k + \frac{1}{2} \sum_{L=1}^{5} c_{jL}^{(i)} S_L = b_j^{(i)} + \sum_{k=1}^{5} c_{kj}^{(i)} S_k \qquad (6-7)$$

Then the 5x5 matrix $\frac{\partial p}{\partial s} [S^{(o)}]$ is computed and the linear equations

$$\frac{\partial p}{\partial s} [S^{(o)}] \quad \Delta S^{(o)} = - p[S^{(o)}] \qquad (6-8)$$

are solved for $\Delta S^{(o)}$ and a new $S^{(1)} = S^{(o)} + \Delta S^{(o)}$ is computed. The process is repeated until $\Delta S^{(n)}$ is sufficiently small. A computational flow chart follows in Figure 6-5.

If for some reason the linear equations become singular or the iteration fails to converge, then it is assumed that $\Delta t'$ is so near zero that the equations have become singular. Then ΔT_{HOM} is set to $\Delta t'$ and execution proceeds to step (C).

6.3 ESTIMATION OF EXECUTION TIME AND STORAGE

The arithmetic execution time of the SIV-B computer, LVDC, is assumed of the following form: fixed point add time α, multiply time 5α, divide time 8α, where $\alpha = 82\mu$ sec (ref. 13). Further, it is assumed that all arithmetic operations are performed in fixed-point mode. Then giving a maximum of five iterations on the polynomial equations, an operation count reveals that a cycle time of less than 1.6 seconds is determined by the Algorithm as defined. It should be noted that the execution time is linear in the number of iterations T on the polynomial equatons with a very large positive derivative. To be specific

$$\text{Extime} \doteq .630 + .180\ T \text{ seconds}$$

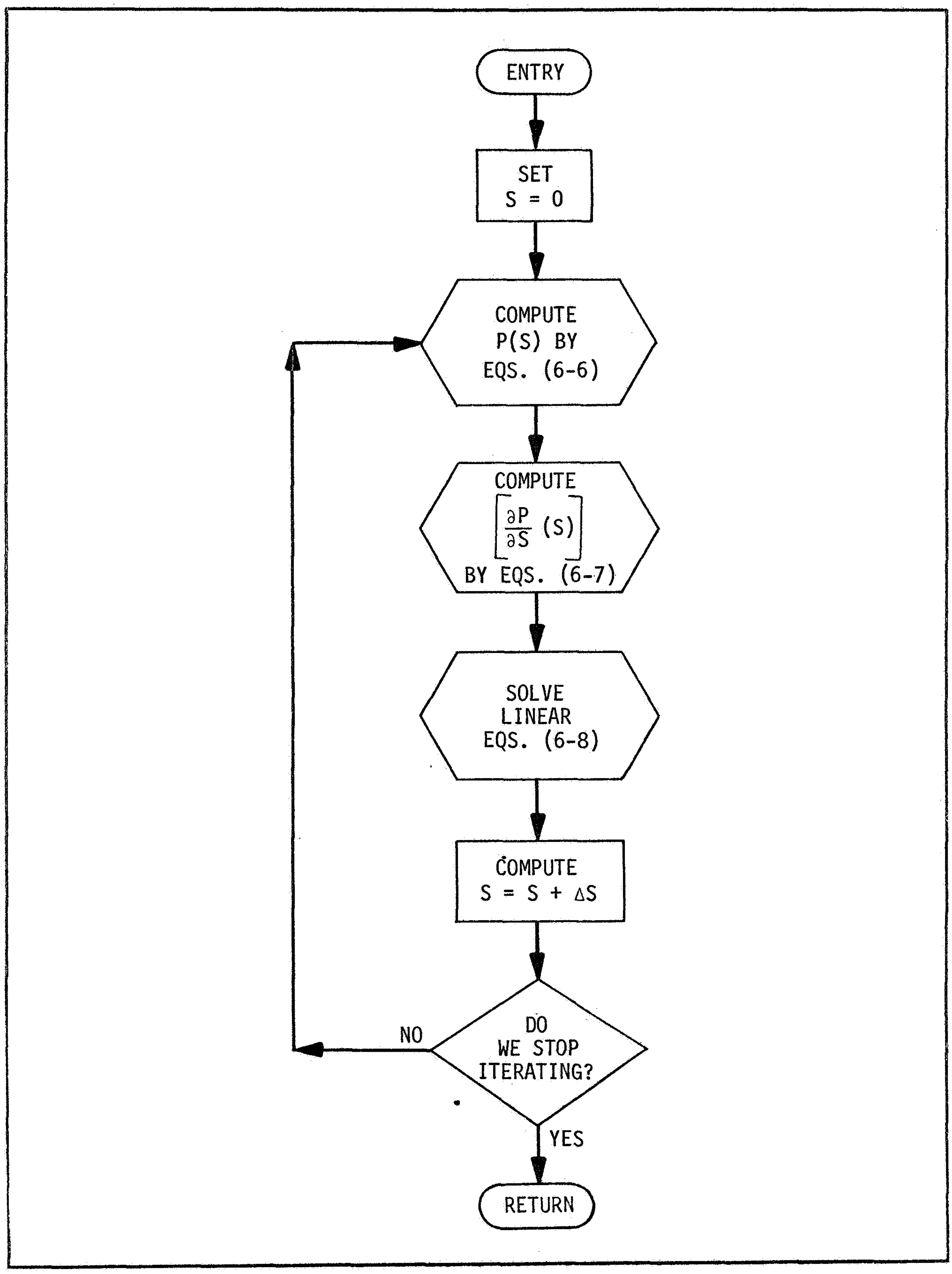

Figure 6-5. FLOWCHART FOR NEWTON-RAPHSON

It is expected that only one or two iterations will be required making the execution time around .8 to 1.1 seconds.

The storage requirements are determined from a UNIVAC 1108 load map of a model deck. It is assumed that one digital word contains one entire data representation. The storage of the model deck is less then 4000 words.

The total execution time is estimated by counting arithmetic operations only. The auxilliary bookkeeping instructions are assumed negligible. In addition some advantage can be taken of the overlap feature of the multiply instruction which allows simultaneous execution of one or two additional minimal cycle instructions (e.g., load and store instructions).

Table 6-1 gives a detailed summary of the arithmetic operation counts necessary in each block of the general flowchart presented earlier. The symbols are defined as follows:

- q dimension of space (2 or 3)
- N dimension of isolation space (i.e., ξ)
- p_1 number of integration steps on nominal trajectory (Block D)
- p_2 number of integration steps in computing $g(Y, \xi')$ (Block E)
- r degree of approximating polynomials for derivatives
- s dimension of state vector Y
- T number of Newton-Raphson iterations on polynomials (maximum of 5)
- a number of intermediate and terminal functions in g vector.

In the numerical integration of blocks D and E it is required to extract a square root. Here, this operation is approximated by 4 adds, 4 multiplies, and 3 divides, i.e., 3 Newton iterations preceded by a normalization. In block J it is required to calculate an inverse tangent which is approximated by a continued fraction of 6 adds, 2 multiplies, and 3 divides. The times of blocks A, C, and K are assumed negligible.

The execution of the guidance package may be one of four different modes depending on the estimated time-to-go $\Delta t'$. The first mode consists of the maximum time and requires the use of Silber's expansion on the first call to

Table 6-1. OPERATION COUNTS FOR GUIDANCE ROUTINE

STEP NUMBER \ OPERATION	ADD	MULTIPLY	DIVIDE
B	$2q + 1$ $N[1+2S+\frac{S(S-1)}{2}]$*	$2q$ $N[2S+S(S-1)]$*	0
D, E	$[P_1+P_2]\ [41q+27]+5q+9$	$[P_1+P_2]\ [53q+36]+9q+19$	$[P_1+P_2]\ 27+0$
F	$a(N-1)\ [1+S+\frac{N+2}{2}]\ r$ (0)*	$a(N-1)\ [1+S+\frac{N+2}{2}]\ r$ (126)*	0
G	$12[1+2(S+N)]$	$12[S+N]$	0
H	$\frac{1}{2}\left[\frac{11}{3}N^3 + 9N^2\right]T$	$\left[\frac{7}{3}N^3 + 3N^2\right]T$	N^2T
I	$2q + 1$	q	0
J	$7(q-1)$	$4(q-1)$	$4(q-1)$

*In block B, indicates Silber's expansion of Mode 1.

In block F, indicates Linear interpolation of Mode 3.

the guidance routine. The second mode is the standard procedure and is used for the time-to-go between the first call and up to $\Delta t' = \Delta T_{SWCH}$. The third mode uses a linear interpolation when calculating the derivatives. The fourth mode is used when $\Delta t' < \Delta T_{HOM}$ and the vehicle is assumed so close to cut-off that linear guidance laws are sufficiently accurate.

It should be emphasized that the execution times previously stated are for a two dimensional injection into a circular orbit. For a general three dimensional problem where it is desired to inject into some prescribed orbit the corresponding execution time is of course greater. A rough calculation using Table 6-1 as a guide (i.e., q = 3, N = 7, S = 8, T = 5) indicates that the execution time is approximately two times that of the two-dimensional problem or 3.0 seconds. Again the execution time is linear in the number of iterations T. For the three dimensional problem the expression

$$\text{Extime} = 1.180 + .356\ T \text{ seconds}$$

characterizes this relation for mode 1 time.

The total effort of the guidance routine is almost entirely composed of three distinct parts. The numerical integration comprises approximately one-fifth of the execution time, the calculation of the derivatives about one-fifth, and the solution of the polynomial equations entails three-fifths. Thus, if the computing time is to be reduced then a good place to begin the reduction is in the iterative solution of the polynomial equations. Here it has been proposed that a Newton-Raphson iteration be used. It is felt that some modification in the iterative procedure could save at least three-tenths of the total time spent in solving the polynomial equations. This resolves into a 1.3 second cycle time for the two-dimensional problem and a 2.4 second cycle time for the 3-D problem.

The storage required for a three-dimensional deck is estimated to be less than two times that required for the storage of the model deck or less than 8000 words.

Section VII

DISCUSSION OF PERFORMANCE

The purpose here is to criticize as objectively as possible the guidance Algorithm of Section VI. Various measures of performance have been outlined in reference 14 and these points are discussed explicitly. Each performance criteria is defined and then followed by supporting critical statements.

7.1 OPTIMALITY

Given that there is a performance index to be minimized, e.g., propellant expenditure; how does the obtained value of the performance index compare to the theoretical minimum?

The minimum value is defined here to be that value obtained by satisfying various necessary minimum conditions of the calculus of variations. This minimum is of course a local minimum and is considered acceptable for this problem. The various errors in the Algorithm are discussed in subsection 7.2. These errors generally subtract from the performance index and it is difficult to state "a priori" just what the total effect is. Empirical study usually gives a good idea of the performance. However, numerical results are not available on this point yet.

7.2 ACCURACY

Given that approximations are introduced into the derivation and mechanization of the guidance equations; what are the resulting errors in the desired terminal conditions? These errors can be classified according to:

- APPROXIMATION ERRORS - Due to analytic approximations introduced into the derivation of the guidance equations.

The physical assumptions of a spherical homogeneous earth and two-body approximations introduce some error. However, this is expected not to be severe. In addition the approximation of the boundary conditions by the second-degree polynominals serves as a perturbation. The results of Section V indicate these errors are certainly acceptable.

- COMPUTER ERRORS - Due to the inaccuracies of the numerical Algorithms used to implement the guidance equations.

A computer error analysis has not been carried out for this Algorithm. However, favorable to this point is the iterative property (with a self-correcting nature) of the guidance scheme.

- MECHANIZATION ERRORS - Due to the inability of the vehicle to physically respond to the guidance commands.

Currently no provisions are made for problems of bounded control or state variables. However, it is possible to extend the guidance Algorithm to cover such cases.

7.3 REGION OF APPLICABILITY

What is the range of perturbations which can be adequately treated by the guidance mode?

In terms of the state space, a second-order region is covered by the Silber-Hunt expansion whereas the iterative refinement of the guidance formulas certainly extends this region. The five percent perturbations in state of the problem of Section V was adequately handled by the guidance formulas.

7.4 COMPUTER FACTORS

What are the real time onboard and/or earth-based computer requirements, in particular, how much storage space is required, what is the length of the computing cycle for iteration of the guidance equations, and how complex must the computer be?

This information is covered in detail in subsection 6.3. In summary, the computer requirements appear to be sufficiently fulfilled by the state-of-the-art hardware, quite probably the LVDC currently used on the S-IVB.

7.5 PREFLIGHT PREPARATION

What is the cost in time and money of preflight preparation of the guidance equations, in particular how long does it take to prepare the guidance system to accomplish a given mission?

The guidance routine as defined in Section VI requires the generation of a nominal trajectory and the computation of the least square polynomials approximating the required derivatives. This job is being accomplished by a program which takes less than 15 minutes of execution on an IBM 7094 for the problem of Section VI.

7.6 FLEXIBILITY

What are the types of missions which the guidance mode can perform, and how well can it adapt to changes in the mission?

Currently, the guidance mode is designed to handle a two-dimensional, constant-burn, minimum-fuel injection into a circular orbit. The reference trajectory is representative of S-IVB type vehicles. However, the basic guidance scheme is very general and requires only that the boundary conditions be expressed as equality constraints. Adapting to changes in mission is accomplished by generating a corresponding nominal trajectory and computing the required derivative least square polynomials. This dependence upon a precomputed reference trajectory is the primary drawback of this guidance scheme.

7.7 GROWTH POTENTIAL

What is the potential applicability of the guidance mode to future missions?

The flexibility of the guidance routine discussed earlier certainly projects the possible capability of this package for use in future missions.

The Algorithm can be extended to cover:

- Low thrust
- Bang-Bang control

- Orbital transfer, rendezvous, intercept
- Bounded control and state
- N-body problems
- Oblateness effects

I.e., almost any conceivable mission in which disturbance from an earth determined reference program is "small".

Section VIII

CONCLUSIONS AND RECOMMENDATIONS

The numerical algorithms discussed in this report add self-correcting features to second variation guidance. However, the techniques are still dependent on proximity to a reference path and require storage of coefficients of polynomial functions of time.

The comparison of the polynomial solutions and the inversion formulas on an S-IVB type trajectory clearly indicates the superiority of the polynomial solutions. Based on these numerical results a guidance algorithm was designed with the polynomial solutions used as the driving element.

One of the more important and unique features of the guidance algorithm was the rapid technique used for computing approximate first and second derivatives of functions of the final state with respect to current Lagrange multipliers: The first derivatives were approximated by correcting nominal derivatives to account for deviations from the current nominal state and nominal Lagrange multipliers. The second derivatives were simply set equal to the nominal derivatives. This technique for finding approximate derivatives led to guidance commands which did not vary much from the guidance algorithm utilizing true derivatives.

Included in the numerical study was an independent use of the expansion of Silber and Hunt. The results showed clearly that a combination of the Silber-Hunt expansion with the polynomial solutions proved much more useful than either taken individually.

The guidance formulas described in Sections III and IV were of arbitrary order. However, the numerical results of Section V indicated that the second or higher order formulas are of questionable utility.

More studies of the guidance algorithm in simulated flight are recommended in order that:

- The best combination of auxilliary algorithms may be selected.
- Features which contribute little to the performance may be eliminated.

A more detailed study should be made of the approximations to the true derivatives in order to find improvements to the method of approximation.

Finally, an effort should be made to make the method less dependent upon a single stored reference trajectory.

Section IX

REFERENCES

1. Thompson, M. L., et al., "An Analytical Approach to Solution of Two-Point Boundary Condition Problems in Optimal Guidance", Summary Report for NASA Contract NASW-1165, Northrop-Nortronics-Huntsville Memo TM-292-6-038, June 1966.

2. Andrus, J. F., et al., "Analytical Research in Guidance Theory", Final Report for NASA Contract NAS12-500, Northrop-Nortronics-Huntsville Report TR-792-9-283, November 1967.

3. Silber, R. and Hunt, R. W., "Space Vehicle Guidance - A Boundary Value Formulation", NASA Tech. Memo X-53059, June 1966.

4. Kelley, H. J., "An Optimal Guidance Approximation Theory", IEEE Trans. on Automatic Control., Vol. AC-9, 1964, pp. 375-380.

5. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., "Optimization and Control of Nonlinear Systems using the Second Variation", J. Society of Industrial and Applied Math. Control, Series A1, 1963, pp. 193-223.

6. Powers, W. F., "A Method For Comparing Trajectories in Optimum Linear Perturbation Guidance Schemes", AIAA Journal, Vol 6, No. 12, December 1968.

7. Leitman, G. (editor), Optimization Techniques, Academic Press, 1962.

8. Traub, J. F., Iterative Methods for the Solution of Equations, Prentice-Hall, 1964.

9. Davidenko, D. F., "On a New Method of Numerical Solution of Systems of Nonlinear Equations", Doklady Akad. Nauk U_ZSSR (N.S.) Vol. 88, 1953 (Russian), pp. 601-602.

10. Athan, Michael and Falb, Peter L., Optimal Control, McGraw-Hill, 1966.

11. Andrus, J. F., "Explicit Solutions to Problems of Optimal Guidance", Summary Report for NASA Contract NAS8-20082, Northrop-Nortronics-Huntsville Report TR-792-8-303, January 1968.

12. Andrus J. F., "Runge-Kutta Formulas For Second-Order Differential Equations", Northrop-Huntsville, Report TR-792-8-322, September 1968.

13. IBM, Systems Training Notes on Launch Vehicle Digital Computer/Launch Vehicle Data Adapter, IBM Federal Systems Division, Huntsville, Ala.

14. Pfeiffer, C. G., "An Analysis of Guidance Modes" in Second Compilation of Papers on Trajectory Analysis and Guidance Theory, PM-67-21, NASA-ERC, January 1967.

Appendix A

A METHOD FOR COMPARING TRAJECTORIES IN OPTIMUM LINEAR PERTURBATION GUIDANCE SCHEMES*

In the application of neighboring optimum feedback guidance schemes the choice of the optimum reference state to compare with the perturbed state is not straightforward. Recent studies have shown that time-to-go is preferrable to clock time and performance index-to-go as a lookup parameter. The guidance Algorithm of Section VI uses the time-to-go criterion to select a point from the reference trajectory. An alternate way of determining the lookup parameter is presented in this analysis. The parameter is determined by solving iteratively a nonlinear algebraic equation in one unknown which is derived from the basic assumption of neighboring optimum feedback guidance (i.e., that the perturbed state is close to the optimum state). This method does not involve an estimation of the perturbed final time whereas time-to-go requires such an estimate.

In recent years the idea of using a linear (and possibly higher order) perturbation of a predetermined optimum trajectory for the feedback guidance of space vehicles has been advanced by a number of investigators (refs. A1 - A4). That is, if $\{x^*(t), u^*(t)\}$ represents a trajectory and control which minimizes

$$J = g(t_f, x_f) + \int_{t_o}^{t_f} L\ (t, x, u)dt \qquad \text{(A-1)}$$

and satisfies the constraints

$$\dot{x}_i = f_i(t, x_1, \ldots, x_n, u_1, \ldots, u_m) \quad (i=1,\ldots,n) \qquad \text{(A-2)}$$

This appendix is contributed by William F. Powers, Assistant Professor, Department of Aerospace Engineering, The University of Michigan, Ann Arbor, who has served as a consultant to Northrop-Huntsville during this contract period.

$$\psi_i(t_o, x_o, t_f, x_f) = 0, \qquad (i=1,\ldots,p\leq 2n+2) \qquad \text{(A-3)}$$

then references A1 - A3 present methods which determine

$$\delta u_i(t) = \sum_{j=1}^{n} G_{ij}\ (t,\tau)\ \delta x_j(\tau), \qquad (i=1,\ldots,m) \qquad \text{(A-4)}$$

where the δx_j functions are perturbed values of the state at $\tau \varepsilon\ [t_o, t_f]$, the $G_{ij}(t,\tau)$ functions are the feedback gains associated with the time τ, and the functions $\delta u_i(t) = u_i^*\ (t) + \delta u_i(t)$ define the optimal controls for $t\varepsilon\ [\tau, t_f]$ if no further disturbances occur. In reference A4, the Lagrange multipliers (which result from the Euler-Lagrange equations associated with the variational problem) for the perturbed trajectory are obtained as power series in the state perturbations, δx_i, and the maximum principle is then used to determine the corresponding δu_i. In the usual case, the δu_i's are determined so that equations (A-3) are satisfied and the perturbed trajectory is optimal in some sense.

Assume that the values of $x_i^*(\tau)$, $u_i^*(\tau)$, and $G_{ij}^*(t,\tau)$ are stored onboard for each t, $\tau\varepsilon[t_o, t_f]$. Then the time, τ, is actually a parameter which associates the feedback gain, $G_{ij}\ (t,\tau)$, with the function space point $(x_1^*(\tau),\ldots, x_n^*(\tau), u_1^*(\tau),\ldots, u_m^*(\tau))$ of the optimal trajectory. An inherent ambiguity in these schemes is the way that the "lookup" parameter $\tau\varepsilon[t_o, t_f]$ is determined for a state $(x_1,\ldots, x_n)$ which is "close" to the optimal trajectory, but not on it. At first glance it appears that the time, say τ_1, at which the vehicle arrives at $(x_1,\ldots x_n)$ is also the value of the lookup parameter, $\tau\varepsilon[t_o, t_f]$. However, τ_1 may be greater than t_f, and/or $x(\tau_1)$ may not be "close" to $x^*(\tau_1)$, whereas $x(\tau_1)$ may be close to some other point on the optimum trajectory, say $x^*(\tau_2)$ (Figure A-1).

In references A5 and A6 an unpublished suggestion by J. C. Dunn is used to alleviate this ambiguity. In reference A5 it is shown that by using time-to-go as the lookup parameter (e.g., in Figure A-1, τ_3 is the time-to-go lookup parameter when the perturbed trajectory is at τ_1 with time-to-go equal to T) instead of clock time, the linear and quadratic guidance for Zermelo's problem is much improved. In reference A6 time-to-go is used to give excellent results

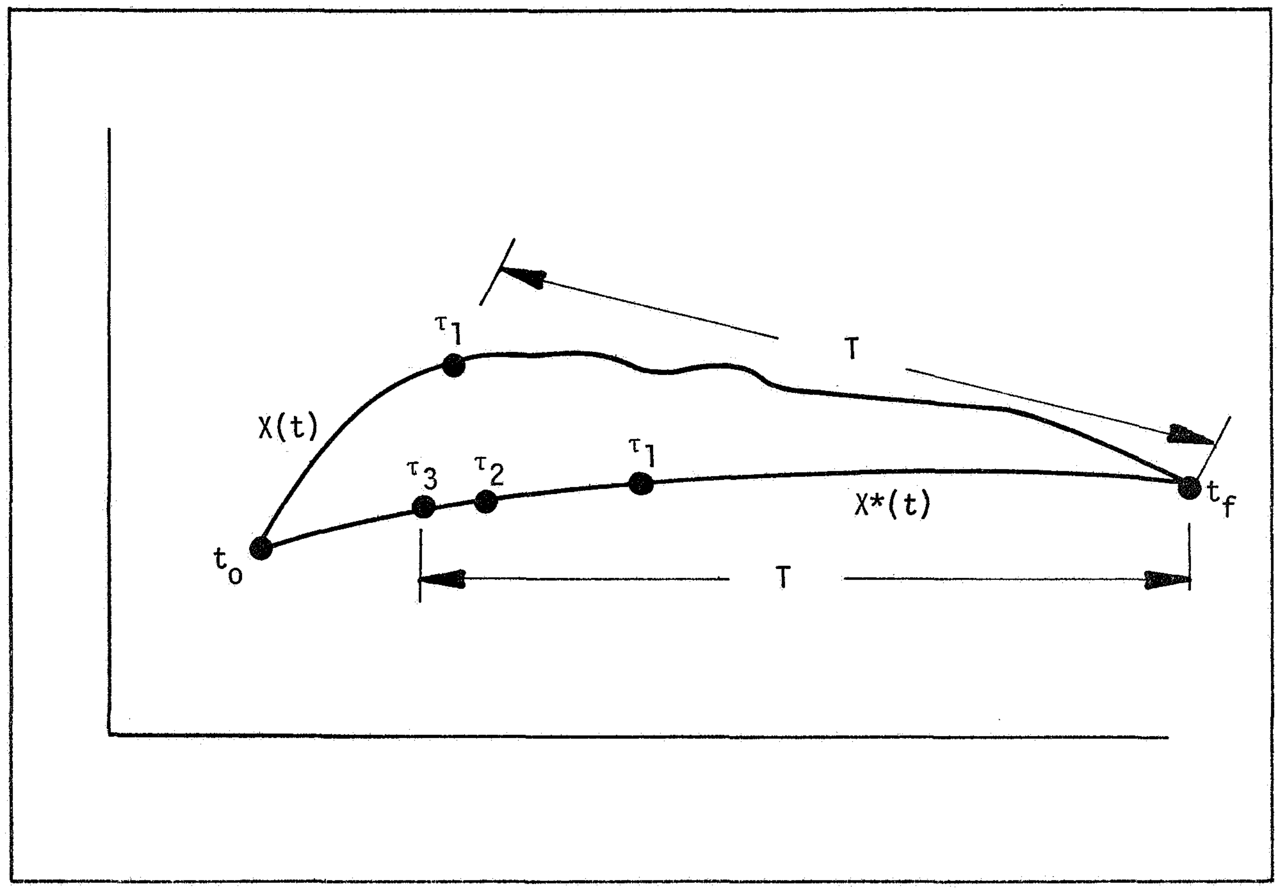

Figure A-1. TIME PARAMETERS ASSOCIATED WITH OPTIMUM AND NEAR-OPTIMUM TRAJECTORIES

for the linear guidance of a reentry vehicle. In both of these analyses estimates of the change in terminal time, t_f, are used to determine the time-to-go on the perturbed trajectories.

Since the basic assumption of a linear perturbation feedback guidance scheme is that the perturbed state and control are "close" to the optimum state and control, respectively, then this should be the main guideline in the selection of the lookup parameter. Thus, one should choose the lookup parameter in such a way that the perturbed state, x, and control, u, are as close as possible to the functions $x^*(t)$ and $u^*(t)$. Since the perturbed control is given as a function of the perturbed state by the guidance scheme, then only the satisfaction of the requirement x "close" to $x^*(t)$ can be used to choose the lookup parameter. Therefore, an alternate method for selecting the lookup parameter $\tau\varepsilon[t_o, t_f]$ is the following: determine the value of $\tau\varepsilon[t_o, t_f]$ at which the "distance" between the point x and the curve x*(t) is minimized (Figure A-2). This method does not involve an estimated change in the terminal time, t_f, on the perturbed trajectory.

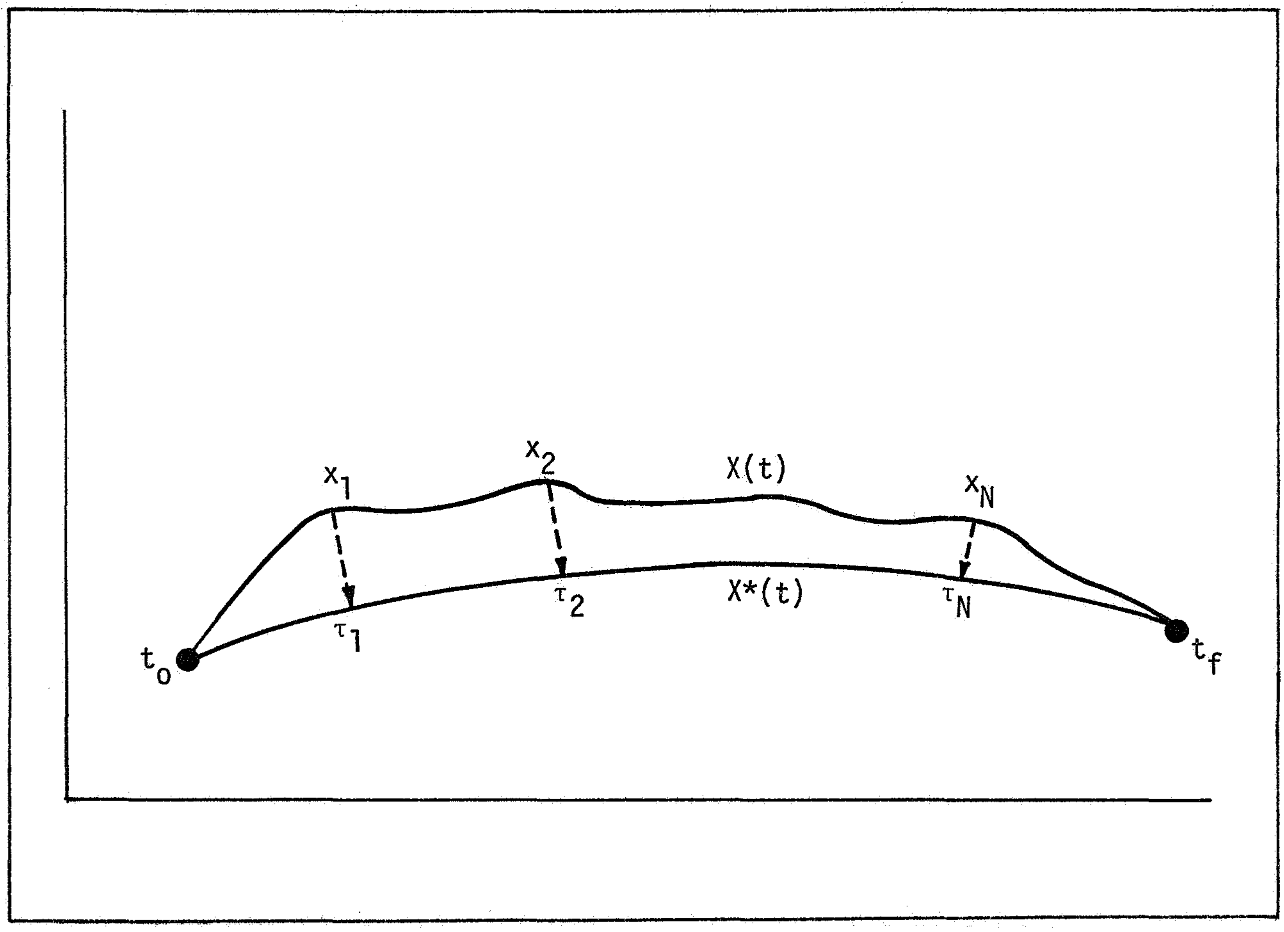

Figure A-2. MINIMUM STATE-SPACE DISTANCE COMPARISONS FOR DETERMINING THE LOOKUP-PARAMETER

To illustrate the application of this method, assume the following situation: the optimum state variables can be represented by polynomials in time, i.e.,

$$x_i^*(t) = \sum_{j=0}^{K+1} a_{ij} t^j, \quad t\varepsilon[t_o, t_f] \quad (i=1,\ldots,n) \tag{A-5}$$

which result from a curve-fitting procedure. In a general analysis, the functional form of the distance function will depend upon the state variables which are employed. For this example, assume the following distance function

$$\rho\ [x, x^*(t)] = [k_1(x_1 - x_1^*(t))^2 + \ldots + k_n(x_n - x_n^*(t))^2]^{1/2} \quad \text{(A-6)}$$

where $x \varepsilon R^n$ is a point "near" the curve $x^*(t) \varepsilon R^n$ and the k_i's are scale factors. The method requires that the following problem be solved: "Determine the $t \varepsilon [t_o, t_f]$ which minimizes the quantity $\rho^2\ [x, x^*(t)]$, where x is a given point in R^n and $x^*(t)$ is given by equation (A-5)." Thus, the result is just an ordinary minimization problem. A necessary condition for this minimization is that the lookup parameter, t, satisfy:

$$\frac{d\rho^2}{dt} = 0 = 2k_1\ [x_1 - x_1^*(t)]\frac{dx_1^*}{dt} + \ldots + 2k_n\ [x_n - x_n^*(t)]\ \frac{dx_n^*}{dt}\ . \quad \text{(A-7)}$$

Equation (A-7) can be rewritten as

$$C_o(a;x) + C_1(a;x)\ t + \ldots + C_{k(k+1)}\ (a;x)\ t^{k(k+1)} = 0, \quad \text{(A-8)}$$

which can be determined as a function of x and t before the flight. Instead of attempting to solve for the roots of equation (A-8) analytically, Newton's method could be used to iterate for the lookup parameter onboard since the clock time is a good estimate of the lookup parameter. Suppose that τ is the solution of equation (A-8) for a given state x. Then,

$$u_i(t) = u_i^*(t) + \sum_{j=1}^{n} G_{ij}(t,\tau)\ [x_i - x_i^*(\tau)] \quad (i=1,\ldots,m) \quad \text{(A-9)}$$

represents the control program for all $t \varepsilon\ [\tau, t_f]$. Note that the actual clock time at $t=\tau$ might be $\tau \pm \Delta t$ ($\Delta t > 0$) so that clock time is not involved in equation (A-9). This allows for an automatic adjustment of the terminal time on the perturbed trajectory.

Appendix A References

A1. Bryson, A. E., and Denham, W. F., "Multivariable Terminal Control for Minimum Mean Square Deviation from a Nominal Path", Proceedings of Vehicle Systems Optimization Symposium (IAS, New York, 1961), pp. 91-97.

A2. Kelley, H. J., "Guidance Theory and Extremal Fields", IRE Trans. on Auto. Cont., Vol. AC-7, pp. 75-82 (1962).

A3. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., "Optimization and Control of Nonlinear Systems using the Second Variation", J. Soc. Ind. Appl. Math. Control, Ser. A1, pp. 193-223 (1963).

A4. Silber, R., "Space Vehicle Guidance - A Boundary Value Formulation", 15th Guidance and Space Flight Theory Meeting, NASA-MSFC, October 1963. (Also, with R. W. Hunt, published in NASA TM X-53059, June 1964.)

A5. Kelley, H. J., "An Optimal Guidance Approximation Theory," IEEE Trans. on Auto. Cont., Vol. AC-9, pp. 375-380 (1964).

A6. Speyer, J. L. and Bryson, A. E., "A Neighboring Optimum Feedback Control Scheme Based on Estimated Time-to-go with Application to Re-Entry Flight Paths", AIAA J. 6, pp. 769-776 (1968).

Appendix B

EXPLICIT FORMULAS FOR FIRST ORDER METHOD

B.1 INTRODUCTION AND PRELIMINARIES

The object of this study is to solve the differential boundary-condition problem related to optimal guidance as derived from the application of the calculus of variations (cov). In particular, the primary goal is to obtain approximate closed form solutions.

The differential equations may be expressed in the form

$$\ddot{X} = \frac{F}{m|\lambda|}\lambda - \frac{\mu}{|X|^3} X, \tag{B-1}$$

$$\ddot{\lambda} = \frac{\mu}{|X|^3}\left(-\lambda + 3\,\frac{X\cdot\lambda}{|X|^2}\,X\right). \tag{B-2}$$

Here, the "super dot" notation indicates differentiation with respect to time. The vectors $X = (X_1, \ldots, X_n)^T$ and $\dot{X} = (\dot{X}_1, \ldots, \dot{X}_n)^T$ represent Cartesian position and velocity of the space vehicle relative to the stationary earth center. The scalars μ, F, and m represent the Gaussian gravitational constant, the vehicle's thrust and mass, respectively. Mass as a function of time is described by

$$\dot{m} = -\beta \tag{B-3}$$

where β is the mass flow rate magnitude. The LaGrange multipliers are denoted by the vectors $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ and $-\dot{\lambda} = (-\dot{\lambda}_1, \ldots, -\dot{\lambda}_n)^T$. The symbol "$|\ |$" denotes the Euclidean norm when applied to vectors, otherwise the absolute value function. The symbols X, $\dot{X}$, λ, and $\dot{\lambda}$ will designate n component vectors where n is the dimension of the space being studied (n = 2 or 3); the vector components are represented by the usual subscript convention. The notation $X\cdot\lambda$ represents the scalar product of X with λ.

Equations (B-1) and (B-2) are often called the equations of motion and the Euler-Lagrange equations, respectively. The form of equation (B-1) reflects the two-body, spherical earth approximation of the physical problem and in this report is assumed to be sufficiently accurate.

At the initial time, t_o, it is assumed that the quantities X, $\dot{X}$, and m are specified. The boundary conditions are given by the equations

$$f_i(\eta,y,\xi) = 0, \qquad i = 1, 2, \ldots, p. \tag{B-4}$$

The f_i are functional constraints upon X, $\dot{X}$, λ, $\dot{\lambda}$, and m at the initial and/or final time, t_f. The vector y represents the variables X, $\dot{X}$, and m at $t = t_o$, while η represents X, $\dot{X}$, λ, $\dot{\lambda}$, and m at $t = t_f$. The p component vector ξ designates those quantities not explicitly known which are required to obtain the solution of the boundary condition problem, e.g., λ and $\dot{\lambda}$ at $t = t_o$ and possibly final time, t_f. Generally the f's include geometric end constraints, transversality conditions from cov, and possibly some scaling conditions.

Implicit in equation (B-4) is the relation

$$g_i(y,\xi) = f_i(\eta(y,\xi),y,\xi) = 0, \quad i = 1, \ldots, p. \tag{B-5}$$

Under the proper assumptions Andrus (ref. B-1) obtains inversion formulas for ξ as a function of an approximate $\tilde{\xi}$, y, and $\eta(y,\tilde{\xi})$. Explicit formulas are given for the coefficients of an expansion of the form

$$\Delta\xi_k = \sum_{i_1} C^{(1)}_{K_{i_1}} f_{i_1} + \sum_{i_1,i_2} C^{(2)}_{K_{i_1 i_2}} f_{i_1} f_{i_2} + \sum_{i_1,i_2,i_3} C^{(3)}_{K_{i_1 i_2 i_3}} f_{i_1} f_{i_2} f_{i_3} + \ldots \tag{B-6}$$

where $\sum\limits_{i_1,i_2}$ represents $\sum\limits_{i_1=1}^{P} \sum\limits_{i_2=1}^{P}$, etc., $f_j = f_j(y,\tilde{\xi})$, and $\Delta\xi_k = \xi_k - \tilde{\xi}_k$.

B-1. Andrus, J. F., "Explicit Solutions to Problems of Optimal Guidance", Northrop-Huntsville Tech. Report TR-792-8-303, *January 1968.*

If for some N and k we let

$$S = C^{(N)}_{ki_1 i_2 \ldots i_N}$$

then it is evident that

$$S = S(\eta(y,\tilde{\xi}),\ y,\ \tilde{\xi}).$$

Thus, due to the functional dependence upon η the coefficients in the inverse series as well as the f_i's will require the value of quantities at the final point of the trajectory. These values, generally, are not known explicitly, hence one must replace η with an approximation $\tilde{\eta}$. The approximations may be obtained by various techniques, e.g., Taylor series or Runge-Kutta expansions as functions of $t_f - t_o$.

The following subsections apply these techniques to a particular problem and derive a set of closed form expressions for the $\Delta\xi_i$'s. It is emphasized that the method used is more general than the following may indicate. Due to the closed form nature of the solution, assumptions valid in one application which are made to simplify the expressions, would not necessarily be valid in another. However, the general procedure is the same.

B.2 USE OF LINEAR TERMS IN INVERSE SERIES

In equation (B-6), if only the first "order" term is retained then we obtain the approximation

$$\Delta\xi_k = \sum_{i_1} C^{(1)}_{k_{i_1}} f_{i_1}$$

or in matrix notation

$$\Delta\xi = Cf(\eta,y,\xi)\Big|_{(y,\tilde{\xi})} \qquad \text{(B-7)}$$

where (ref. B-1)

$$-C^{-1} = A = \frac{\partial f_i}{\partial \xi_i}(\eta, y, \xi)\Bigg|_{(y,\tilde{\xi})} .$$

The assumption of the nonsingularity of A is implicit in this analysis. Equation (B-7) has the equivalent form

$$A\Delta\xi = -f \; . \qquad \text{(B-8)}$$

The solution of the linear equations (B-8) may be effected by any of several methods.

B.2.1 TYPES OF ITERATION

Given an approximation $\tilde{\xi}$ to ξ in equation (B-5), then under certain conditions the solution to equation (B-8) gives an improved approximation, $\tilde{\xi} + \Delta\xi$, to ξ. This procedure is known commonly as the method of Newton-Raphson. If we define $\xi^{(0)} = \tilde{\xi}$ and $\xi^{(1)} = \tilde{\xi} + \Delta\xi = \xi^{(0)} + \Delta\xi^{(0)}$ then a recursive definition follows and can be formulated by

$$\xi^{(n+1)} = \xi^{(n)} + \Delta\xi^{(n)} \qquad \text{(B-9)}$$

where

$$A\Delta\xi^{(n)} = -f \qquad \text{(B-10)}$$

with

$$A = A\left[\eta(y,\xi^{(n)}), y, \xi^{(n)}\right],$$

$$f = f\left[\eta(y,\xi^{(n)}), y, \xi^{(n)}\right].$$

In carrying out iteration (B-9) it may sometimes be useful to introduce a scaling factor at each step of the iteration. This is commonly done in the form

$$\xi^{(n+1)} = \xi^{(n)} + a_n \Delta\xi^{(n)} \qquad \text{(B-11)}$$

where $0 < a_n \leq 1$, $n = 0, 1, 2, \ldots$. There are various ways for selecting a_n and a judicious choice can reap a considerable reduction in labor.

Another variation in (B-9) is related to the computation of A at each step of the iteration. Often, labor can be reduced if at steps n+1, n+2, ..., n+p the A of step n is used. The index p may be selected by monitoring some norm of f and using A until this norm stops decreasing.

B.2.2 SOME NOTATION

Define for $t_o \leq t \leq t_f$ the scalar products

$$\begin{aligned}
R^2 &= X \cdot X, \\
V^2 &= \dot{X} \cdot \dot{X}, \\
\Lambda^2 &= \lambda \cdot \lambda, \\
\theta &= X \cdot \lambda, \\
\theta_1 &= \dot{X} \cdot \lambda, \\
\theta_2 &= X \cdot \dot{\lambda}, \\
\theta_3 &= \dot{X} \cdot \dot{\lambda}, \\
\theta_4 &= X \cdot \dot{X}, \\
\theta_5 &= \lambda \cdot \dot{\lambda}, \\
\theta_6 &= \dot{\lambda} \cdot \dot{\lambda}.
\end{aligned}$$

To simplify the notation of equations (B-1) and (B-2) let

$$b_1 = \frac{1}{m\Lambda},$$

$$b_2 = \frac{1}{R^3},$$

$$b_3 = \frac{b_2 \theta}{R^2}.$$

Then equations (B-1) and (B-2) have the equivalent form

$$\ddot{X} = Fb_1\lambda - \mu b_2 X , \qquad \text{(B-12)}$$

$$\ddot{\lambda} = \mu(-b_2\lambda + 3b_3 X). \qquad \text{(B-13)}$$

To handle the time derivatives of the above equations introduce

$$\dot{b}_1 = b_1 b_4$$

$$b_4 = \frac{\beta}{m} - \frac{\theta_5}{\Lambda^2}$$

$$\dot{b}_4 = b_5 = \frac{\beta^2}{m^2} - \frac{1}{\Lambda^2}\left(3\mu b_3\theta + \theta_6 - \mu b_2\Lambda^2 - \frac{2\theta_5^2}{\Lambda^2}\right)$$

$$\dot{b}_5 = b_6 = \frac{2\beta^3}{m^3} - \frac{1}{\Lambda^2}\left\{3\mu b_3\left(2\left[\theta_1+2\theta_2\right] - \theta\left[\frac{5\theta_4}{R^2} + \frac{6\theta_5}{\Lambda^2}\right]\right)\right.$$

$$\left. + \mu b_2\left(\frac{3\theta_4}{R^2}\Lambda^2 + 2\theta_5\right) - \frac{2\theta_5}{\Lambda^2}\left(3\theta_6 - \frac{4\theta_5^2}{\Lambda^2}\right)\right\}.$$

When the subscripts "o" and "f" are appended to symbols, the quantities are assumed initial and final values, respectively. The definitions of the above symbols shall be referred to as the equations of set I.

B.2.3 APPROXIMATION OF $\eta(y, \xi)$

Recall from subsection B.1 that

$$\eta(y,\xi) = \begin{bmatrix} X \\ \dot{X} \\ m \\ \lambda \\ \dot{\lambda} \end{bmatrix}_{t=t_f}$$

Thus for $t_o \leq t \leq t_f$, we can define $\bar{\eta}(y,\xi,t)$ by the differential equations (B-1), (B-2), and (B-3) with $\eta(y,\xi) = \bar{\eta}(y,\xi,t_f)$. The problem becomes that of approximating $\bar{\eta}(y,\xi,t_f)$ in terms of the known quantities $\bar{\eta}(y,\xi,t_o)$.

B.2.3.1 Taylor Series Approximations

With y and ξ fixed and assuming m and f continuous, it is evident from equations (B-1), (B-2), and (B-3) that there exists ℓ continuous time derivatives of $\bar{\eta}(y,\xi,t)$ for $t_o \leq t \leq t_f$. Hence by Taylor's theorem

$$\bar{\eta}(y,\xi,t_f) = \bar{\eta}(y,\xi,t_o) + \ldots + \bar{\eta}^{(\ell-1)}(y,\xi,t_o)\frac{\Delta t^{\ell-1}}{(\ell-1)!} + R_\ell \qquad \text{(B-14)}$$

where $\Delta t = t_f - t_o$

and $R_\ell = \bar{\eta}^{(\ell)}(y,\xi,\varepsilon)\dfrac{\Delta t^\ell}{\ell!}$,

with $t_o \leq \varepsilon \leq t_f$.

In the following only the truncated portion of equation (B-14) shall be used, i.e., R_ℓ will be assumed negligible and dropped.

From the definition of $\bar{\eta}$ this will require expansions of X_f, $\dot{X}_f$, λ_f, $\dot{\lambda}_f$, and m_f. A glance at equation (B-3) with the assumption β is constant with respect to time reveals the exact representation

$$m_f = m_o - \beta\Delta t. \qquad \text{(B-15)}$$

Expansions for X_f and λ_f are obtained below and those for $\dot{X}_f$ and $\dot{\lambda}_f$ may be obtained by differentiating the respective series. Let S represent one of X or λ. We shall use the convention

$$\left.\frac{d^n S}{dt^n}\right|_{t=t_o} = S_o^{(n)} = b_{n1}\lambda_o + b_{nz}\dot{\lambda}_o + b_{n3}X_o + b_{nx}\dot{X}_o \qquad \text{(B-16)}$$

Differentiating equation (B-16) with respect to time and using equations (B-1) and (B-2) gives

$$S_o^{(n+1)} = (\dot{b}_{n1} - \mu b_2 b_{n2} + F b_1 b_{n4})\lambda_o + (\dot{b}_{n2} + b_{n1})\dot{\lambda}_o$$

$$+ (\dot{b}_{n3} + 3\mu b_3 b_{n2} - \mu b_2 b_{n4})X_o + (\dot{b}_{n4} + b_{n3})\dot{X}_o$$

$$= b_{n+1,1}\lambda_o + b_{n+1,2}\dot{\lambda}_o + b_{n+1,3}X_o + b_{n+1,4}\dot{X}_o$$

From the equation above a recursive definition for the coefficients in equation (B-16) may be induced. Let

$$B_n = (b_{n1}, b_{n2}, b_{n3}, b_{n4})^T$$

and

$$A = \begin{bmatrix} 0 & -\mu b_2 & 0 & Fb_1 \\ 1 & 0 & 0 & 0 \\ 0 & 3\mu b_3 & 0 & -\mu b_2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Then it follows

$$B_{n+1} = \dot{B}_n + AB_n. \quad \text{(B-17)}$$

To compute B_n for $X_o^{(n)}$ use the initial condition

$$B_o = (1, 0, 0, 0)^T.$$

For $\lambda_o^{(n)}$ use

$$B_o = (0, 0, 1, 0)^T.$$

Equation (B-17) may be used directly to compute the derivatives of X_o and λ_o. These derivatives are derived in subsection B.3.5.1 up to $X_o^{(5)}$, $\lambda_o^{(5)}$.

B.2.3.2 Other Approximations

It might prove advantageous to consider approximations implied by various numerical integration techniques. For example a single-step method like the Runge-Kutta formulas or possibly multistep methods locked with the Runge-Kutta starters. Even Romberg-type formulas might submit to simplification.

The point to be made here is there are many different types of approximations that can be tried and further investigation is suggested.

B.3 APPLICATIONS: INJECTION INTO CIRCULAR ORBIT

Consider a two-dimension guidance problem in which the object is to burn with a constant thrust magnitude, F, into a prespecified circular orbit. It is assumed that the rocket engines will be initially thrusting at $t = t_o$, the vehicle's position and velocity are known in the sense of equation (B-1), as we well as the mass. The mass flow rate, β, in equation (B-3), is constant and assumed specified. The problem is to find the values of the Lagrange multipliers, λ and $-\dot{\lambda}$, at $t = t_o$ which define the optimal direction in which the rocket engines should thrust in order to minimize the fuel consumption.

In order to inject into a circular orbit the radius (measured from the earth's center) and velocity vectors must be perpendicular, as well as having prescribed values. These geometric conditions can be expressed in the form

$$X_f \cdot X_f - R_{co}^2 = 0 \qquad \text{(radius condition)}$$

$$\dot{X}_f \cdot \dot{X}_f - V_{co}^2 = 0 \qquad \text{(velocity condition)} \qquad \text{(B-18)}$$

$$X_f \cdot \dot{X}_f = 0 \qquad \text{(orthogonality condition)}$$

where R_{co} and V_{co} are specified radius and velocity at cut-off. The subscript "f" indicates that evaluation is performed at $t = t_f$. Equations (B-18) completely describe a circular orbit in two dimensions. The cov ties down the injection point into the orbit by the addition of the transversality condition

$$\lambda_{10}\dot{X}_{20} - \lambda_{20}\dot{X}_{10} - \dot{\lambda}_{10}X_{20} + \dot{\lambda}_{20}X_{10} = 0, \qquad \text{(B-19)}$$

where the subscript "o" indicates initial values. Inspection of equations (B-1) and (B-2) reveals them to be homogeneous in λ, i.e., for a > 0, a λ also satisfies the equations, hence a scaling equation is needed to insure uniqueness of the solution, e.g.,

$$\lambda_o \cdot \lambda_o - 1 = 0 \qquad \text{(B-20)}$$

Equations (B-18), (B-19), and (B-20) may be adjoined to give five boundary conditions corresponding to the set of (B-4), i.e.,

$$\begin{aligned}
f_1 &= X_f \cdot X_f - R_{co}^2 = 0 \\
f_2 &= \dot{X}_f \cdot \dot{X}_f - V_{co}^2 = 0 \\
f_3 &= X_f \cdot \dot{X}_f = 0 \\
f_4 &= \lambda_o \cdot \lambda_o - 1 = 0 \\
f_5 &= \lambda_{10}\dot{X}_{20} - \lambda_{20}\dot{X}_{10} - \dot{\lambda}_{10}X_{20} + \dot{\lambda}_{20}X_{10} = 0
\end{aligned} \qquad \text{(B-21)}$$

The vectors y and η become

$$y = \begin{bmatrix} X_o \\ \dot{X}_o \\ m_o \end{bmatrix}$$

and

$$\eta = \begin{bmatrix} X_f \\ \dot{X}_f \\ \lambda_f \\ \dot{\lambda}_f \\ m_f \end{bmatrix}$$

Since the final time is unknown, t_f is inserted into the ξ vector and we obtain

$$\xi = \begin{bmatrix} \lambda_o \\ \dot{\lambda}_o \\ t_f \end{bmatrix} . \qquad \text{(B-22)}$$

Differentiating the f_i, i = 1, ..., 5, of equation (B-21) with respect to ξ_j, j = 1, ..., 5, gives

$$A = \begin{bmatrix} 2X_f \cdot X_f^{(1)} & 2X_f \cdot X_f^{(2)} & 2X_f \cdot X_f^{(3)} & 2X_f \cdot X_f^{(4)} & 2X_f \cdot \dot{X}_f \\ 2\dot{X}_f \cdot \dot{X}_f^{(1)} & 2\dot{X}_f \cdot \dot{X}_f^{(2)} & 2\dot{X}_f \cdot \dot{X}_f^{(3)} & 2\dot{X}_f \cdot \dot{X}_f^{(4)} & 2\dot{X}_f \cdot \ddot{X}_f \\ X_f \cdot \dot{X}_f^{(1)} + \dot{X}_f \cdot X_f^{(1)} & X_f \cdot \dot{X}_f^{(2)} + \dot{X}_f \cdot X_f^{(2)} & X_f \cdot \dot{X}_f^{(3)} + \dot{X}_f \cdot X_f^{(3)} & X_f \cdot \dot{X}^{(4)} + \dot{X}_f \cdot X_f^{(4)} & X_f \cdot \ddot{X}_f + \dot{X}_f \cdot \dot{X}_f \\ 2\lambda_{10} & 2\lambda_{20} & 0 & 0 & 0 \\ \dot{X}_{20} & -\dot{X}_{10} & -X_{20} & X_{10} & 0 \end{bmatrix} . \qquad \text{(B-23)}$$

Here the notation $X_f^{(j)} = \frac{\partial X_f}{\partial \xi_j}$ is used. Since $\xi_5 = t_f$ then it is straightforward to show that $X_f^{(5)} \doteq \dot{X}_f$. The following subsections will discuss a method of solving equation (B-8) with the relevant approximation of $\eta(y,\tilde{\xi})$.

To facilitate some notational problems, we define for i, j, and k ranging over the integers 1, 2, 3, and 4

$$\alpha_{ij} = \det \begin{bmatrix} X \cdot X^{(i)} & X \cdot X^{(j)} & X \cdot \dot{X} \\ \dot{X} \cdot \dot{X}^{(i)} & \dot{X} \cdot \dot{X}^{(j)} & \dot{X} \cdot \ddot{X} \\ X^{(i)} \cdot \dot{X} + \dot{X}^{(i)} \cdot X & X^{(j)} \cdot \dot{X} + \dot{X}^{(j)} \cdot X & \dot{X} \cdot \dot{X} + X \cdot \ddot{X} \end{bmatrix},$$

$$\beta_{j} = \det \begin{bmatrix} \frac{1}{2} f_1 & X \cdot X^{(j)} & X \cdot \dot{X} \\ \frac{1}{2} f_2 & \dot{X} \cdot \dot{X}^{(j)} & \dot{X} \cdot \ddot{X} \\ f_3 & X^{(j)} \cdot \dot{X} + \dot{X}^{(j)} \cdot X & \dot{X} \cdot \dot{X} + X \cdot \ddot{X} \end{bmatrix},$$

$$\pi_{ijk} = \det \begin{bmatrix} X \cdot X^{(i)} & X \cdot X^{(j)} & X \cdot X^{(k)} \\ \dot{X} \cdot \dot{X}^{(i)} & \dot{X} \cdot \dot{X}^{(j)} & \dot{X} \cdot \dot{X}^{(k)} \\ X^{(i)} \cdot \dot{X} + \dot{X}^{(i)} \cdot X & \text{--} & \text{--} \end{bmatrix},$$

$$\zeta_{ij} = \det \begin{bmatrix} \frac{1}{2} f_1 & X \cdot X^{(i)} & X \cdot X^{(j)} \\ \frac{1}{2} f_2 & \dot{X} \cdot \dot{X}^{(i)} & \dot{X} \cdot \dot{X}^{(j)} \\ f_3 & X^{(i)} \cdot \dot{X} + \dot{X} \cdot X^{(i)} & X^{(j)} \cdot \dot{X} + \dot{X} \cdot X^{(j)} \end{bmatrix}.$$

It is to be understood that all evaluations are terminal i.e., at $t = t_f$. If S represents any of the above quantities for fixed i, j, or k, then it is evident that

$$S = S\left[\eta(y,\tilde{\xi}),\ y,\ \tilde{\xi}\right].$$

B.3.1 CRAMER'S RULE

The linear equations (B-8) may be solved by the application of Cramer's rule, i.e.,

$$\Delta\xi_j^{(n)} = -\frac{\det(A_j)}{\det(A)}\ , \quad j = 1, 2, 3, 4, 5,$$

where A_j is the matrix A with the j^{th} column replaced by f.

Using equation (B-23) and expanding about the fourth row of A gives

$$\det(A) = 8\left\{-\lambda_{10}\det\begin{bmatrix} X\cdot X^{(2)} & X\cdot X^{(3)} & X\cdot X^{(4)} & X\cdot\dot{X} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ -\dot{X}_{10} & -X_{20} & X_{10} & 0 \end{bmatrix} + \lambda_{20}\det\begin{bmatrix} X\cdot X^{(1)} & X\cdot X^{(3)} & X\cdot X^{(4)} & X\cdot\dot{X} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \dot{X}_{20} & -X_{20} & X_{10} & 0 \end{bmatrix}\right\}$$

Expanding the resulting determinants about the fourth row and using the predefined α_{ij} yields

$$\det(A) = 8\left\{-\lambda_{10}\left[\dot{X}_{10}\alpha_{34} - X_{20}\alpha_{24} - X_{10}\alpha_{23}\right] + \lambda_{20}\left[-\dot{X}_{20}\alpha_{34} - X_{20}\alpha_{14} - X_{10}\alpha_{13}\right]\right\}$$

or more conveniently

$$\det(A) = 8\lambda_o\cdot\left(-\alpha_{34}\dot{X}_o + \begin{bmatrix} \alpha_{23} & \alpha_{24} \\ -\alpha_{13} & -\alpha_{14} \end{bmatrix} X_o\right). \tag{B-24}$$

Thus det(A) resolves into the inner product of λ_o and A vector V which is a combination of $\dot{X}_o$ and X_o.

Now, turning our attention to $\det(A_j)$ notice that λ_{10} and λ_{20} may be scaled such that $f_4 = 0$. Then for j = 1

$$\det(A_1) = -8\det\begin{bmatrix} \frac{1}{2}f_1 & X\cdot X^{(2)} & X\cdot X^{(3)} & X\cdot X^{(4)} & X\cdot\dot{X} \\ \frac{1}{2}f_2 & \dot{X}\cdot\dot{X}^{(2)} & --- & --- & --- \\ f_3 & X^{(2)}\cdot\dot{X} + \dot{X}^{(2)}\cdot X & --- & --- & --- \\ 0 & \lambda_{20} & 0 & 0 & 0 \\ f_5 & -\dot{X}_{10} & -X_{20} & X_{10} & 0 \end{bmatrix}.$$

Expanding about the fourth row gives

$$\det(A_1) = -8\lambda_{20}\det\begin{bmatrix} \frac{1}{2}f_1 & X\cdot X^{(3)} & X\cdot X^{(4)} & X\cdot\dot{X} \\ \frac{1}{2}f_2 & \dot{X}\cdot\dot{X}^{(3)} & --- & --- \\ f_3 & X^{(3)}\cdot\dot{X} + \dot{X}^{(3)}\cdot X & --- & --- \\ f_5 & -X_{20} & X_{10} & 0 \end{bmatrix}$$

Examination of f_5 reveals that one may choose λ_{10} or λ_{20} such that $f_5 = 0$. Then expanding about the last row gives

$$\det(A_1) = -8\lambda_{20}\left\{-X_{20}\beta_4 - X_{10}\beta_3\right\}$$
$$= 8\lambda_{20}X_o\cdot\begin{pmatrix}\beta_3\\ \beta_4\end{pmatrix}. \quad \text{(B-25)}$$

Similarly for j = 2, 3, 4, 5,

$$\det(A_2) = -8\lambda_{10}X_o\cdot\begin{pmatrix}\beta_3\\ \beta_4\end{pmatrix}, \quad \text{(B-26)}$$

$$\det(A_3) = 8\lambda_o\cdot\left(\beta_4\dot{X}_o + X_{10}\begin{pmatrix}\beta_2\\ -\beta_1\end{pmatrix}\right), \quad \text{(B-27)}$$

$$\det(A_4) = -8\lambda_o\cdot\left(\beta_3\dot{X}_o - X_{20}\begin{pmatrix}\beta_2\\ -\beta_1\end{pmatrix}\right), \quad \text{(B-28)}$$

$$\det(A_5) = 8\lambda_o\cdot\left(\zeta_{34}\dot{X}_o - \begin{pmatrix}\zeta_{23} & \zeta_{24}\\ -\zeta_{13} & \zeta_{14}\end{pmatrix}X_o\right). \quad \text{(B-29)}$$

In equations (B-24) through (B-29) the subscripted quantities α, β, and ζ are all to be evaluated at the current approximation $\tilde{\xi}_5 = t_f$. Thus, in order to get closed form expressions for $\Delta\xi_j$, the α's, β's, and ζ's must be approximated by some artifice.

B.3.2 APPROXIMATION OF BOUNDARY FUNCTIONS

It is required to have some closed form approximating expressions for f_i, i = 1, 2, and 3, defined by equations (B-21). Notice it is not necessary to expand the boundary functions f_4 and f_5 since they are initial conditions. A further savings of labor is obtained by the fact that

$$f_1^{(n)} = \frac{2}{n} F_3^{(n-1)} , \; n = 1, 2, \ldots . \tag{B-30}$$

Hence truncated Taylor series shall be obtained for f_1 and f_2 only. Using the notation

$$f_1 = R_1 + R_2\Delta t + R_3\Delta t^2 + \ldots + R_6\Delta t^5 + e_1 \tag{B-31}$$

and

$$f_2 = V_1 + V_2\Delta t + \ldots + V_5\Delta t^4 + e_2, \tag{B-32}$$

then

$$f_3 = \frac{1}{2} R_2 + R_3\Delta t + \frac{3}{2} R_4\Delta t^2 + \ldots + \frac{5}{2} R_6\Delta t^4 + e_3.$$

The e_i, i = 1, 2, and 3, are the remainder terms and $R_{n+1} = \frac{1}{n} \dot{R}_n$, n = 1, 2, ..., with R_o being f_1 evaluated at t_o. This notation is similarly defined for V_n.

The expressions for R_i and V_i are derived in section B.3.5.2 and shall hereafter be designated those of set II.

B.3.3 APPROXIMATION OF TERMINAL ELEMENTS IN CRAMER'S RULE

The purpose of this subsection is to derive experssions for the subscripted quantities ζ,α, and β of subsection B.3.1. Before proceeding to these expansions some notational artifices are introduced. From the definition of ζ, α, and β the quantities $X \cdot X^{(j)}$, $\dot{X} \cdot \dot{X}^{(j)}$, and $X^{(j)} \cdot \dot{X} + \dot{X}^{(j)} \cdot X$, j = 1, 2, 3, 4, are explicitly involved. Define for j = 1, 2, 3, 4

$$P_j(X) = X \cdot X^{(j)} \tag{B-33}$$

Then

$$P_j(\dot{X}) = \dot{X} \cdot \dot{X}^{(j)}$$

and

$$\dot{P}_j(X) = X^{(j)} \cdot \dot{X} + \dot{X}^{(j)} \cdot X \ .$$

The following expansions will have coefficients which are functions of the $P_j(X_o)$ and its time derivatives. The necessary formulas are derived and listed in subsection B.3.6 as set III.

Consider first the ζ_{ij}'s, then introduce the notation

$$\zeta_{ijf} = \zeta_{ij}\Big|_{t=t_f} \quad \text{and} \quad \zeta_{ijo} = \zeta_{ij}\Big|_{t=t_o} \ .$$

Expanding ζ_{ijf} in a Taylor series out to fourth order terms gives

$$\zeta_{ijf} = \zeta_{ij_o}^{(1)} + \zeta_{ijo}^{(2)}\Delta t + \zeta_{ijo}^{(3)}\Delta t^2 + \zeta_{ijo}^{(4)}\Delta t^3 + \zeta_{ijo}^{(5)}\Delta t^4 + \tilde{\zeta}_{ijo}^{(6)}\Delta t^5$$

where

$$\zeta_{ijo}^{(n+1)} = \frac{1}{n!} \frac{d^n\zeta_{ij}}{dt^n}\Bigg|_{t=t_o} = \frac{1}{n} \dot{\zeta}_{ijo}^{(n)}, \quad n = 1, 2, \ldots,$$

$$\Delta t = t_f - t_o$$

and

$$\tilde{\zeta}_{ijo}^{(6)} = \frac{d^6\zeta_{ij}}{dt^6}\Bigg|_{t=\varepsilon} \frac{1}{6!}, \quad t_o \leq \varepsilon \leq t_f \ .$$

Similar notation will apply to α and β. By definition

$$\zeta_{ij} = \det \begin{bmatrix} \frac{1}{2}f_1 & P_i(X) & P_j(X) \\ \frac{1}{2}f_2 & P_i(\dot{X}) & P_j(\dot{X}) \\ f_3 & \dot{P}_i(X) & \dot{P}_j(X) \end{bmatrix}$$

and it follows that

$$\zeta_{ijo}^{(1)} \equiv D = \det \begin{bmatrix} \frac{1}{2}R_1 & P_i(X_o) & P_j(X_o) \\ \frac{1}{2}V_1 & P_i(\dot{X}_o) & P_j(\dot{X}_o) \\ R_2 & \dot{P}_1(X_o) & \dot{P}_j(X_o) \end{bmatrix} = 0$$

since

$$P_i(X_o) = \dot{P}_i(X_o) = P_i(\dot{X}_o) = 0 \qquad i = 1, \ldots, 4.$$

By differentiating the determinant D and using the row wise derivative rule then

$$\zeta_{ijo}^{(2)} \equiv \dot{D} = D_1 + D_2 + D_3 = 0$$

where the subscripts refer to the rows being differentiated. Similarly

$$\zeta_{ijo}^{(3)} \equiv \frac{1}{2}\ddot{D} = \frac{1}{2}(D_{11} + 2D_{12} + D_{22} + 2D_{23} + D_{33} + 2D_{13})$$

$$\equiv \frac{1}{2}(2D_{13} + D_{22} + 2D_{23} + D_{33}) = 0$$

These results follow due to two additional facts. First

$$\ddot{P}_i(X_o) = \dot{P}_i(\dot{X}_o) = 0 \qquad i = 3, 4$$

and secondly because the third row of the matrix is the time derivative of the first row. Thus, whenever the first row is differentiated one more time than the third row, the resulting determinant is identically zero for all t and furthermore all additional derivatives of that term may be dropped.

Continuing

$$\zeta_{ijo}^{(4)} \equiv \frac{1}{6}\dddot{D} = \frac{1}{6}(2D_{113} + 4D_{123} + 3D_{133} + D_{122} + D_{222} + 3D_{223} + 3D_{233} + D_{333})$$

$$= \frac{1}{6}(4D_{123} + 3D_{133} + D_{222} + 3D_{223} + 3D_{233} + D_{333})$$

$$= \frac{1}{2}(D_{223} + D_{233})$$

$$= \frac{1}{4} R_1 \left[\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - \ddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right] ,$$

$$\zeta_{ijo}^{(5)} \equiv \frac{1}{24} D^{(iv)} = \frac{1}{24}(7D_{1223} + 10D_{1233} + 3D_{1133} + 4D_{1333} + D_{2222} + 4D_{2223} + 6D_{2233} + 4D_{2333} + D_{3333})$$

$$= \frac{1}{24}(4D_{2223} + 7D_{1223} + 6D_{2233} + 10D_{1233} + 4D_{2333} + 3D_{1133})$$

$$= \frac{1}{24} R_1 \left[2\dot{P}_i(\dot{X}_o)P_j^{(iv)}(X_o) - 2\ddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) + 3\ddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 3\dddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right] + \frac{1}{24} R_2 \left[10\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 7\ddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right] - \frac{1}{16} V_1 \ddot{P}_i(X_o)\dddot{P}_j(X_o) .$$

$$\zeta_{ijo}^{(6)} \equiv \frac{1}{60} D^{(v)} = \frac{1}{60}(11D_{12223} + 23D_{12233} + 13D_{11233} + 18D_{12333} + 7D_{11333} + 5D_{13333} + D_{22222} + 5D_{22223} + 10D_{22233} + 10D_{22333} + 5D_{23333} + D_{33333})$$

$$= \frac{1}{60}(11D_{12223} + 23D_{12233} + 13D_{11233} + 18D_{12333} + 7D_{11333} + 5D_{22223} + 10D_{22233} + 10D_{22333} + 5D_{23333})$$

$$= \frac{1}{60} R_1\left[-\frac{5}{2}\ddot{P}_i(X_o)P_j^{(iv)}(\dot{X}_o) + 5\dddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 5\dddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) + 5\ddot{P}_i(\dot{X}_o)P_j^{(iv)}(X_o) - 5P_i^{(iv)}(X_o)\ddot{P}_j(\dot{X}_o) + \frac{5}{2}\dot{P}_i(\dot{X}_o)P_j^{(v)}(X_o)\right]$$
$$+ \frac{1}{60} R_2\left[11\ddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) + 23\ddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 23\dddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) + 18\dot{P}_i(\dot{X}_o)P_j^{(iv)}(X_o) \quad - \frac{1}{60} V_1 \frac{7}{2}\ddot{P}_i(X_o)P_j^{(v)}(X_o)\right]$$
$$- \frac{1}{60} V_2\left[13\ddot{P}_i(X_o)\dddot{P}_j(X_o)\right] + \frac{1}{60}\dot{P}_i(\dot{X}_o)\left[2R_3\dddot{P}_j(X_o)\right]$$

For α_{ij} we have by definition

$$\alpha_{ij} = \det\begin{bmatrix} P_i(X) & P_j(X) & \frac{1}{2}\dot{f}_1 \\ P_i(\dot{X}) & P_j(\dot{X}) & \frac{1}{2}\dot{f}_2 \\ \dot{P}_i(X) & \dot{P}_j(X) & \dot{f}_3 \end{bmatrix}.$$

By interchanging columns 2 and 3, then 1 and 2, it is seen that α_{ij} is the determinant of the same matrix as ζ_{ij} with the exception of the first column which is just the time derivative of the first column of ζ_{ij}. Therefore, the formulas for $\alpha_{ijo}^{(n)}$ may be obtained from those of $\zeta_{ijo}^{(n)}$ by replacing the elements of the first column by those in $\alpha_{ijo}^{(n)}$, i.e., R_n by nR_{n+1}, V_n by nV_{n+1}. This gives

$$\alpha_{ijo}^{(1)} = \alpha_{ijo}^{(2)} = \alpha_{ijo}^{(3)} = 0,$$

$$\alpha_{ijo}^{(4)} = \frac{1}{4} R_2\left[\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - \ddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o)\right] ,$$

$$\begin{aligned}\alpha_{ijo}^{(5)} = \frac{1}{24} R_2\Big[&2\dot{P}_i(\dot{X}_o)P_j^{(iv)}(X_o) - 2\ddot{P}_i(X_o)\ddddot{P}_j(\dot{X}_o) + 3\ddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) \\ &- 3\dddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o)\Big] + \frac{1}{12} R_3\Big[10\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) \\ &- 7\dot{P}_i(X_o)\ddot{P}_j(\dot{X}_o)\Big] - \frac{1}{16} V_2\ddot{P}_i(X_o)\dddot{P}_j(X_o).\end{aligned}$$

$$\begin{aligned}\alpha_{ijo}^{(6)} = \frac{1}{60} R_2\Big[&- \frac{5}{2} \ddot{P}_i(X_o)P_j^{(iv)}(\dot{X}_o) + 5\dddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) \\ &- 5\dddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) + 5\ddot{P}_i(\dot{X}_o)P_j^{(iv)}(X_o) - 5P_i^{(iv)}(X_o)\ddot{P}_j(\dot{X}_o) \\ &+ \frac{5}{2} \dot{P}_i(\dot{X}_o)P_j^{(v)}(X_o)\Big] + \frac{1}{30} R_3 \Big[11\ddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) \\ &+ 23\ddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 23\dddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) + 18\dot{P}_i(\dot{X}_o)P_j^{(iv)}(X_o)\Big] \\ &- \frac{1}{60} V_2\left[\frac{7}{2} \ddot{P}_i(X_o)P_j^{(v)}(X_o)\right] - \frac{1}{30} V_3\left[13\ddot{P}_i(X_o)\dddot{P}_j(X_o)\right] \\ &+ \frac{1}{10} R_4\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o)\end{aligned}$$

Proceeding similarly with β_i

$$\beta_i = \det \begin{bmatrix} \frac{1}{2} f_1 & P_i(X) & \frac{1}{2} \dot{f}_1 \\ \frac{1}{2} f_2 & P_i(\dot{X}) & \frac{1}{2} \dot{f}_2 \\ f_3 & \dot{P}_1(X) & \dot{f}_3 \end{bmatrix} .$$

Then

$$\beta_{io}^{(1)} \equiv D = \det \begin{bmatrix} \frac{1}{2} R_1 & P_i(X_o) & R_2 \\ \frac{1}{2} V_1 & P_i(\dot{X}_o) & V_2 \\ R_2 & \dot{P}_i(X_o) & 2R_3 \end{bmatrix} = 0$$

Differentiating the determinant in column fashion

$$\beta_{io}^{(2)} \equiv \dot{D} = D_1 + D_2 + D_3 = D_2 + D_3 = D_2$$

$$= \frac{1}{2} R_1 \left[2R_3 \dot{P}_i(\dot{X}_o) - V_2 \ddot{P}_i(\dot{X}_o) \right]$$

$$\beta_{io}^{(3)} \equiv \frac{1}{2} \ddot{D} = \frac{1}{2}(D_{12} + D_{22} + 2D_{23} + D_{13} + D_{33})$$

$$\equiv \frac{1}{2}(D_{22} + 2D_{23} + D_{13} + D_{33}) = \frac{1}{2}(D_{22} + 2D_{23})$$

$$= \frac{1}{4} R_1 \left[2R_3 \ddot{P}_i(\dot{X}_o) - V_2 \dddot{P}_i(X_o) \right] - \frac{1}{4} V_1 \left[2R_3 \ddot{P}_i(X_o) \right.$$

$$\left. - R_2 \dddot{P}_i(X_o) \right] + \frac{1}{2} R_2 \left[R_2 \ddot{P}_i(\dot{X}_o) - V_2 \ddot{P}_i(X_o) \right]$$

$$+ \frac{1}{2} R_1 \left[6R_4 \dot{P}_i(\dot{X}_o) - 2V_3 \ddot{P}_i(X_0) \right] + R_2 \left[\frac{1}{2} V_1 \ddot{P}_i(X_o) \right.$$

$$\left. - R_2 \dot{P}_i(\dot{X}_o) \right]$$

$$\beta_{io}^{(4)} \equiv \frac{1}{6}\dddot{D} = \frac{1}{6}(D_{122} + D_{222} + 3D_{223} + 3D_{123} + 3D_{233} + D_{113} + 2D_{133} + D_{333})$$

$$= \frac{1}{6}(D_{222} + 3D_{223} + 3D_{123} + 3D_{233} + 2D_{133} + D_{333})$$

$$= \frac{1}{6}(D_{222} + 3D_{223} + 3D_{123} + 3D_{233})$$

$$= \frac{1}{12} R_1\left[2R_3\dddot{P}_i(\dot{X}_o) - V_2P_i^{(iv)}(X_o)\right] - \frac{1}{12} V_1\left[2R_3\dddot{P}_i(X_o) - R_2P_i^{(iv)}(X_o)\right]$$

$$+ \frac{1}{6} R_2\left[V_2\dddot{P}_i(X_o) - R_2\dddot{P}_i(\dot{X}_o)\right] + \frac{1}{4} R_1\left[6R_4\ddot{P}_i(\dot{X}_o) - 2V_3\dddot{P}_i(X_o)\right]$$

$$- \frac{1}{4} V_1\left[6R_4\ddot{P}_i(X_o) - 2R_3\dddot{P}_i(X_o)\right] + \frac{1}{2} R_2\left[2V_3\ddot{P}_i(X_o) - 2R_3\ddot{P}_i(\dot{X}_o)\right]$$

$$+ \frac{1}{4} R_2\left[6R_4\dot{P}_i(\dot{X}_o) - 2V_3\ddot{P}_i(X_o)\right] + R_3\left[\frac{1}{2} V_2\ddot{P}_i(X_o) - 2R_3\dot{P}_i(\dot{X}_o)\right]$$

$$+ \frac{1}{4} R_1\left[24R_5\dot{P}_i(\dot{X}_o) - 6V_4\ddot{P}_i(X_o)\right] + 3R_4\left[\frac{1}{2} V_1\ddot{P}_i(X_o) - R_2\dot{P}_i(\dot{X}_o)\right]$$

$$\beta_{io}^{(5)} = \frac{1}{24} D^{(iv)} = \frac{1}{24}(D_{1222} + D_{2222} + 4D_{2223} + 6D_{1223} + 6D_{2233} + 3D_{1123}$$

$$+ 8D_{1233} + 4D_{2333} + 2D_{1133} + 3D_{1333} + D_{3333})$$

$$= \frac{1}{24}(D_{2222} + 4D_{2223} + 6D_{1223} + 6D_{2233} + 8D_{1233} + 4D_{2333}$$

$$+ 2D_{1133} + 3D_{1333} + D_{3333})$$

$$= \frac{1}{24}(D_{2222} + 4D_{2223} + 6D_{1223} + 6D_{2233} + 8D_{1233} + 4D_{2333})$$

$$= \frac{1}{48} R_1\left[2R_3P_i^{(iv)}(\dot{X}_o) - V_2P_i^{(v)}(X_o)\right] - \frac{1}{48} V_1\left[2R_3P_i^{(iv)}(X_o) - R_2P_i^{(v)}(X_o)\right]$$

$$+ \frac{1}{24} R_2\left[V_2P_i^{(iv)}(X_o) - R_2P_i^{(iv)}(\dot{X}_o)\right] + \frac{1}{12} R_1\left[6R_4\dddot{P}_i(\dot{X}_o) - 2V_3P_i^{(iv)}(X_o)\right]$$

$$- \frac{1}{12} V_1\left[6R_4\dddot{P}_i(X_o) - 2R_3P_i^{(iv)}(X_o)\right] + \frac{1}{6} R_2\left[2R_3\dddot{P}_i(\dot{X}_o) - 2V_3\dddot{P}_i(X_o)\right]$$

$$+ \frac{1}{8} R_2 \left[6R_4 \ddot{P}_i(\dot{X}_o) - 2V_3 \dddot{P}_i(X_o) \right] - \frac{1}{8} V_2 \left[6R_4 \ddot{P}_i(X_o) - 2R_3 \dddot{P}_i(X_o) \right]$$

$$+ \frac{1}{2} R_3 \left[2V_3 \ddot{P}_i(X_o) - 2R_3 \ddot{P}_i(\dot{X}_o) \right] + \frac{1}{8} R_1 \left[24R_5 \ddot{P}_i(\dot{X}_o) - 6V_4 \dddot{P}_i(X_o) \right]$$

$$- \frac{1}{8} V_1 \left[24R_5 \ddot{P}_i(X_o) - 6R_4 \dddot{P}_i(X_o) \right] + \frac{1}{4} R_2 \left[6V_4 \ddot{P}_i(X_o) - 6R_4 \ddot{P}_i(\dot{X}_o) \right]$$

$$+ \frac{1}{6} R_2 \left[24R_5 \dot{P}_i(\dot{X}_o) - 6V_4 \ddot{P}_i(X_o) \right] + 2R_4 \left[\frac{1}{2} V_1 \ddot{P}_i(X_o) - R_2 \dot{P}_i(\dot{X}_o) \right]$$

$$+ \frac{1}{12} R_1 \left[120R_6 \dot{P}_i(\dot{X}_o) - 24V_5 \ddot{P}_i(X_o) \right] + 4R_6 \left[\frac{1}{2} V_1 \ddot{P}_i(X_o) - R_2 \dot{P}_i(\dot{X}_o) \right]$$

B.3.4 NUMERICAL PROCEDURE

Implementation of these results could proceed in these steps.

1. Input initial guesses at ξ and formula constants.
2. Initialize λ_o, and $\dot{\lambda}_o$ such that $f_4 = f_5 = 0$.
3. Compute new ξ by Newton Raphson via formulas of Set I, II, and III.
4. Proceed to step 2 and repeat another iteration or exit if convergence or divergence is detected.

B.3.5 SUPPLEMENTARY FORMULAS

Here various expressions are derived to supplement those of earlier subsections.

B.3.5.1 Time Derivatives of X and λ

It is desired to obtain the expansions

$$X_f = X_o + \dot{X}_o \Delta t + \ddot{X}_o \frac{\Delta t^2}{2!} + \dddot{X}_o \frac{\Delta t^3}{3!} + X_o^{(4)} \frac{\Delta t^4}{4!} + X_o^{(5)} \frac{\Delta t^5}{5!}$$

$$\dot{X}_f = \dot{X}_o + \ddot{X}_o \Delta t + \dddot{X}_o \frac{\Delta t^2}{2!} + X_o^{(4)} \frac{\Delta t^3}{3!} + X_o^{(5)} \frac{\Delta t^4}{4!} .$$

In subsection B.3.1 it is shown that

$$X_o^{(n)} = b_{n1}\lambda_o + b_{n2}\dot{\lambda}_o + b_{n3}X_o + b_{n4}\dot{X}_o \quad , \quad (n = 0, 1, 2, \ldots) \qquad \text{(B-34)}$$

where the b_{ni}'s satisfy the differential recurrence relation defined by equation (B-17) with the initial condition $B_o = (0, 0, 1, 0)^T$. Direct application of the recurrence formula gives

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -\mu b_2 & 0 & Fb_1 \\ 1 & 0 & 0 & 0 \\ 0 & 3\mu b_3 & 0 & -\mu b_2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} .$$

In the same fashion

$$B_2 = \begin{bmatrix} Fb_1 \\ 0 \\ -\mu b_2 \\ 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} Fb_1 b_4 \\ Fb_1 \\ \dfrac{3\mu b_2 \theta_4}{R^2} \\ -\mu b_2 \end{bmatrix}, \quad B_4 = \begin{bmatrix} Fb_1 (b_5 - 2\mu b_2 + b_4^2) \\ 2Fb_1 b_4 \\ 6\mu b_3 Fb_1 - \dfrac{3\mu b_2}{R^2}\left(\dfrac{5\theta_4^2}{R^2} + \dfrac{2}{3}\dfrac{\mu}{R} - V^2\right) \\ \dfrac{6\mu b_2}{R^2}\theta_4 \end{bmatrix},$$

$$B_5 = \begin{bmatrix} Fb_1\left(b_6 + \dfrac{12\mu b_2 \theta_4}{R^2} + 3b_4 b_5 - 4\mu b_2 b_4 + b_4^3\right) \\ Fb_1\left(3b_4^2 - 2\mu b_2 + 3b_5\right) \\ Fb_1\left(\dfrac{b_2}{R^2}\left[2\theta_1 + \theta_2 - \dfrac{10\theta_4\theta}{R^2}\right] + 2b_3 b_4\right)6\mu - \dfrac{3\mu b_2}{R^2}\dfrac{\theta_4}{R^2}\left(15V^2 - \dfrac{10\mu}{R} - \dfrac{35\theta_4^2}{R^2}\right) \\ 12\mu b_3 Fb_1 - \dfrac{3\mu b_2}{R^2}\left(\dfrac{15\theta_4^2}{R^2} + \dfrac{8}{3}\dfrac{\mu}{R} - 3V^2\right) \end{bmatrix} .$$

Similarly one obtains for λ using $B_o = (1, 0, 0, 0)^T$

$$B_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -\mu b_2 \\ 0 \\ 3\mu b_3 \\ 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} \frac{3\mu b_2}{R^2} \\ -\mu b_2 \\ \frac{3\mu b_2}{R^2}(\theta_1 + \theta_2 - \frac{5\theta_4\theta}{R^2}) \\ 3\mu b_3 \end{bmatrix},$$

$$B_4 = \begin{bmatrix} 6\mu b_3 F b_1 - \frac{3\mu b_2}{R^2}\left(\frac{5\theta_4^2}{R^2} + \frac{2}{3}\frac{\mu}{R} - V^2\right) \\ \frac{6\mu b_2}{R^2}\theta_4 \\ \frac{3\mu b_2}{R^2}\left(Fb_1\Lambda^2 + 2\theta_3 + 4\mu b_2\theta + \frac{5\theta}{R^2}\left[2\frac{\theta_4^2}{R^2} - Fb_1\theta - V^2\right] - \frac{5\theta_4}{R^2}\left[2\theta_1 + 2\theta_2 - \frac{5\theta_4\theta}{R^2}\right]\right) \\ \frac{6\mu b_2}{R^2}\left(\theta_1 + \theta_2 - \frac{5\theta_4\theta}{R^2}\right) \end{bmatrix}.$$

B.3.5.2 Expansion of Terminal Conditions in Circular Problem

Equations (B-21), (B-31), and (B-32) give

$$R_1 = X_o \cdot X_o - R_{co}^2 = R^2 - R_{co}^2$$

$$R_2 = (2X \cdot \dot{X})_o = 2\theta_4$$

$$R_3 = (X \cdot \ddot{X} + \dot{X} \cdot \dot{X})_o = Fb_1 X_o \cdot \lambda_o - \mu b_2 X_o \cdot X_o + V^2 = Fb_1\theta - \mu b_2 R^2 + V^2$$

$$R_4 = \frac{1}{3}(X\cdot\dddot{X} + 3\dot{X}\cdot\ddot{X})_o = \frac{1}{3}(F\dot{b}_1\theta + Fb_1\dot{\theta} + \mu b_2\theta_4 + 2V\dot{V})$$

$$= \frac{1}{3}\left[Fb_1(b_4\theta + \theta_1 + \theta_2) + \mu b_2\theta_4 + 2Fb_1\theta_1 - 2\mu b_2\theta_4\right]$$

$$= \frac{1}{3}\left[Fb_1(b_4\theta + 3\theta_1 + \theta_2) - \mu b_2\theta_4\right]$$

$$R_5 = \frac{1}{4}\dot{R}_4 = \frac{1}{12}\left[F\dot{b}_1(b_4\theta + 3\theta_1 + \theta_2) + Fb_1(b_5\theta + b_4\dot{\theta} + 3\dot{\theta}_1 + \dot{\theta}_2) + 3\mu b_2\frac{\theta_4^2}{R^2} - \mu b_2\dot{\theta}_4\right]$$

$$= \frac{1}{12}\left\{Fb_1\left(b_4^2\theta + 3b_4\theta_1 + b_4\theta_2 + b_5\theta + b_4\theta_1 + b_4\theta_2 + 3\left[Fb_1\Lambda^2 - \mu b_2\theta + \theta_3\right] + \theta_3 + 2\mu b_2\theta\right) + 3\mu b_2\frac{\theta_4^2}{R^2} - \mu b_2\left[Fb_1\theta - \frac{\mu}{R} + V^2\right]\right\}$$

$$= \frac{1}{12}\left\{Fb_1\left(\left[b_4^2 + b_5 - 2\mu b_2\right]\theta + 4b_4\theta_1 + 2b_4\theta_2 + 4\theta_3 + 3Fb_1\Lambda^2\right) - \mu b_2\left(V^2 - \frac{\mu}{R} - \frac{3\theta_4^2}{R^2}\right)\right\}$$

$$R_6 = \frac{1}{60}\left\{Fb_1\left(\left[b_4\theta + \dot{\theta}\right]\left[b_4^2 + b_5 - 2\mu b_2\right] + \left[b_4\theta_1 + \dot{\theta}_1\right]4b_4 + \left[b_4\theta_2 + \dot{\theta}_2\right]2b_4 + \left[b_4\theta_3 + \dot{\theta}_3\right]4 + \left[b_4\Lambda^2 + 2\Lambda\dot{\Lambda}\right]3Fb_1 + \left[2b_4b_5 + b_6 + \frac{6\mu b_2\theta_4}{R^2}\right]\theta + 4b_5\theta_1 + 2b_5\theta_2 + 3Fb_1b_4\Lambda^2\right) + \frac{3\mu b_2\theta_4}{R^2}\left(V^2 - \frac{\mu}{R} - \frac{3\theta_4^2}{R^2}\right) - \mu b_2\left(2V\dot{V} + \mu b_2\theta_4 + \frac{6\theta_4^3}{R^4} - \frac{6\theta_4}{R^2}\left[Fb_1\theta - \mu b_2R^2 + V^2\right]\right)\right\}$$

$$= \frac{1}{60}\left\{Fb_1\left(\left[b_4^3 + 3b_4b_5 - 2\mu b_2b_4 + b_6 + \frac{12\mu b_2\theta_4}{R^2}\right]\theta + \left[5b_4^2 + 5b_5 - 8\mu b_2\right]\theta_1\right.\right.$$

$$\left. + \left[3b_4^2 + 3b_5 - 6\mu b_2\right]\theta_2 + 10b_4\theta_3 + 12\mu b_3\theta_4 + 10Fb_1\theta_5 + 10b_4Fb_1\Lambda^2\right)$$

$$\left. + \frac{3\mu b_2\theta_4}{R^2}\left(V^2 - \frac{\mu}{R} - \frac{3\theta_4^2}{R^2}\right) - \mu b_2\left(-\mu b_2\theta_4 + \frac{6\theta_4^3}{R^4} - \frac{6\theta_4}{R^2}\left[V^2 - \mu b_2R^2\right]\right)\right\}$$

$$= \frac{1}{60}\left\{Fb_1\left(\left[b_4^3 + 3b_4b_5 - 2\mu b_2b_4 + b_6 + \frac{24\mu b_2\theta_4}{R^2}\right]\theta + \left[5b_4^2 + 5b_5 - 8\mu b_2\right]\theta_1\right.\right.$$

$$\left. + \left[3b_4^2 + 3b_5 - 6\mu b_2\right]\theta_2 + 10b_4\theta_3 + 10Fb_1\left[\theta_5 + b_4\Lambda^2\right]\right)$$

$$\left. + \frac{3\mu b_2\theta_4}{R^2}\left(3V^2 - 3\frac{\mu}{R} - \frac{5\theta_4^2}{R^2}\right) + 2\mu b_2^2\theta_4\right\}$$

Similarly calculate for the velocity equation

$$V_1 = \dot{X}_o \cdot \dot{X}_o - V_{co}^2 = V^2 - V_{co}^2$$

$$V_2 = (2\dot{X}\cdot\ddot{X})_o = 2Fb_1\theta_1 - 2\mu b_2\theta_4$$

$$V_3 = \frac{1}{2}\dot{V}_2 = Fb_1b_4\theta_1 + Fb_1\dot{\theta}_1 + 3\mu b_2\frac{\theta_4^2}{R^2} - \mu b_2\dot{\theta}_4$$

$$= Fb_1\left(b_4\theta_1 + Fb_1\Lambda^2 - \mu b_2\theta + \theta_3\right) - \mu b_2\left(Fb_1\theta - \frac{\mu}{R} + V^2 - \frac{3\theta_4^2}{R^2}\right)$$

$$= Fb_1\left(-2\mu b_2\theta + b_4\theta_1 + \theta_3 + Fb_1\Lambda^2\right) - \mu b_2\left(V^2 - \frac{\mu}{R} - \frac{3\theta_4^2}{R^2}\right)$$

$$V_4 = \frac{1}{3}\dot{V}_3 = \frac{1}{3}\left\{Fb_1\left(\left[b_4\theta + \dot{\theta}\right]\left[-2\mu b_2\right] + \left[b_4\theta_1 + \dot{\theta}_1\right]b_4 + \left[b_4\theta_3 + \dot{\theta}_3\right]\right.\right.$$

$$\left. + \left[b_4\Lambda^2 + 2\Lambda\dot{\Lambda}\right]Fb_1 + \frac{6\mu b_2\theta_4}{R^2}\theta + b_5\theta_1 + Fb_1b_4\Lambda^2\right)$$

$$\left. + \frac{3\mu b_2\theta_4}{R^2}\left(V^2 - \frac{\mu}{R} - \frac{3\theta_4^2}{R^2}\right) - \mu b_2\left(2V\dot{V} + \mu b_2\theta_4 + \frac{6\theta_4^3}{R^4} - \frac{6\theta_4\dot{\theta}_4}{R^2}\right)\right\}$$

$$= \frac{1}{3}\left\{Fb_1\left(\left[-3\mu b_2 b_4 + \frac{15\mu b_2\theta_4}{R^2}\right]\theta + \left[-5\mu b_2 + b_4^2 + b_5\right]\theta_1 - 3\mu b_2\theta_2 + 2b_4\theta_3 + 3Fb_1\left[\theta_5 + b_4\Lambda^2\right]\right) + \frac{3\mu b_2\theta_4}{R^2}\left(3V^2 - 3\frac{\mu}{R} - \frac{5\theta_4^2}{R^2}\right) + \mu^2 b_2^2\theta_4\right\}$$

$$V_5 = \frac{1}{12}\left\{Fb_1\left(\left[b_4\theta + \dot{\theta}\right]\left[\frac{15\mu b_2\theta_4}{R^2} - 3\mu b_2 b_4\right] + \left[b_4\theta_1 + \dot{\theta}_1\right]\left[b_4^2 + b_5 - 5\mu b_2\right] + \left[b_4\theta_2 + \dot{\theta}_2\right]\left[-3\mu b_2\right] + \left[b_4\theta_3 + \dot{\theta}_3\right]2b_4 + 6Fb_1 b_4\left[\theta_5 + b_4\Lambda^2\right] + \left[-\frac{75\mu b_2\theta_4^2}{R^4} + \frac{15\mu b_2}{R^2}\dot{\theta}_4 + \frac{9\mu b_2\theta_4}{R^2}b_4 - 3\mu b_2 b_5\right]\theta + \left[\frac{15\mu b_2\theta_4}{R^2} + 2b_4 b_5 + b_6\right]\theta_1 + \frac{9\mu b_2\theta_4}{R^2}\theta_2 + 2b_5\theta_3 + \left[\dot{\theta}_5 + b_5\Lambda^2 + 2b_4\Lambda\dot{\Lambda}\right]3Fb_1\right) + \left(-\frac{15\mu b_2\theta_4^2}{R^4} + \frac{3\mu b_2\dot{\theta}_4}{R^2}\right)\left(3V^2 - 3\frac{\mu}{R} - \frac{5\theta_4^2}{R^2}\right) + \frac{3\mu b_2\theta_4}{R^2}\left(6V\dot{V} + 3\mu b_2\theta_4 + \frac{10\theta_4^3}{R^4} - \frac{10\theta_4\dot{\theta}_4}{R^2}\right) - \frac{6\mu^2 b_2^2\theta_4^2}{R^2} + \mu^2 b_2^2\dot{\theta}_4\right\}$$

$$= \frac{1}{12}\left\{Fb_1\left(\left[\frac{15\mu b_2 b_4\theta_4}{R^2} - 3\mu b_2 b_4^2 - \mu b_2 b_4^2 - \mu b_2 b_5 + 5\mu^2 b_2^2 - 6\mu^2 b_2^2 - \frac{75\mu b_2\theta_4^2}{R^4} + \frac{15\mu b_2}{R^2}\left(Fb_1\theta - \mu b_2 R^2 + V^2\right) + \frac{9\mu b_2\theta_4 b_4}{R^2} - 3\mu b_2 b_5 + 9Fb_1\mu b_3 + \frac{3\mu b_2}{R^2}\left(3V^2 - 3\frac{\mu}{R} - \frac{5\theta_4^2}{R^2}\right) - \frac{30\mu b_2\theta_4^2}{R^4} + \mu^2 b_2^2\right]\theta + \left[\frac{15\mu b_2\theta_4}{R^2} - 3\mu b_2 b_4\right.\right.\right.$$

$$+ b_4^3 + b_4 b_5 - 5\mu b_2 b_4 - 2\mu b_2 b_4 + \frac{15\mu b_2 \theta_4}{R^2} + 2b_4 b_5 + b_6 + \frac{18\mu b_2 \theta_4}{R^2}\Bigg] \theta_1$$

$$+\left[\frac{15\mu b_2 \theta_4}{R^2} - 3\mu b_2 b_4 - 3\mu b_2 b_4 - 2\mu b_2 b_4 + \frac{9\mu b_2 \theta_4}{R^2}\right]\theta_2 + \Bigg[b_4^2 + b_5 - 5\mu b_2$$

$$- 3\mu b_2 + 2b_4^2 + 2b_5\Bigg] \theta_3 + \left[6\mu b_3 b_4\right] \theta_4 + \left[14 b_4 F b_1\right] \theta_5 + 3Fb_1 \theta_6$$

$$+\left[\frac{\Lambda^2}{3}\left(b_4^2 + b_5 - 5\mu b_2\right) + 2b_4^2\Lambda^2 - \mu b_2 \Lambda^2 + b_5 \Lambda^2\right] 3Fb_1\Bigg) + \frac{3\mu b_2}{R^2}\Bigg(\Bigg[V^2$$

$$- \mu b_2 R^2 - \frac{5\theta_4^2}{R^2}\Bigg]\left[3V^2 - 3\mu b_2 R^2 - \frac{5\theta_4^2}{R^2}\right] + \theta_4 \Bigg[-6\mu b_2 \theta_4 + \frac{10\theta_4^3}{R^4} + 3\mu b_2 \theta_4$$

$$- \frac{10\theta_4}{R^2}\left(V^2 - \mu b_2 R^2\right) - 2\mu b_2 \theta_4\Bigg]\Bigg) + \mu b_2 \left(V^2 - \mu b_2 R^2\right) \Bigg\}$$

$$V_5 = \frac{1}{12}\Bigg\{Fb_1\Bigg(\Bigg[\frac{30\mu b_2 b_4 \theta_4}{R^2} - 4\mu b_2 b_4^2 - 4\mu b_2 b_5 - 24\mu^2 b_2^2 - \frac{120\mu b_2 \theta_4^2}{R^4}$$

$$+ 24Fb_1 \mu b_3 + \frac{24\mu b_2 V^2}{R^2}\Bigg] \theta + \left[\frac{48\mu b_2 \theta_4}{R^2} - 10\mu b_2 b_4 + b_4^3 + 3b_4 b_5 + b_6\right] \theta_1$$

$$+\left[\frac{24\mu b_2 \theta_4}{R^2} - 8\mu b_2 b_4\right] \theta_2 + \left[3b_4^3 + 3b_5 - 8\mu b_2\right] \theta_3 + 14Fb_1 b_4 \theta_5$$

$$+ 3Fb_1 \theta_6 + \left[7b_4^2 + 4b_5 - 8\mu b_2\right] Fb_1 \Lambda^2\Bigg) + \frac{3\mu b_2}{R^2}\Bigg(\Bigg[V^2 - \mu b_2 R^2$$

$$- \frac{5\theta_4^2}{R^2}\Bigg] \left[3V^2 - 3\mu b_2 R^2 - \frac{5\theta_4^2}{R^2}\right] + \theta_4 \left[5\mu b_2 \theta_4 + \frac{10\theta_4^3}{R^4} - \frac{10\theta_4 V^2}{R^2}\right]\Bigg)$$

$$+ \mu b_2 \left(V^2 - \mu b_2 R^2\right) \Bigg\}$$

B.3.5.3 Time Derivatives of $X_o^{(j)}$

Using the recurrence formula of (B-17), the time derivatives of $X^{(j)}$ may be obtained by differentiating equation (B-34) with respect to ξ_j, j = 1, 2, 3, 4. Then

$$X_o^{(n)(j)} = b_{n1}^{(j)}\lambda_o + b_{n1}\lambda_o^{(j)} + b_{n2}^{(j)}\dot{\lambda}_o + b_{n2}\dot{\lambda}_o^{(j)} + b_{n3}^{(j)}X_o + b_{b4}^{(j)}\dot{X}_o$$

since $X_o^{(j)} = \dot{X}_o^{(j)} = 0$. The recurrence formulas of subsection B.2.3.1 have been used to calculate b_{n1} and b_{n2} for n = 1, 2, 3, 4, and 5 in subsection B.3.5. Let

$$B_n^{(j)} = \left(b_{n1}^{(j)}, b_{n2}^{(j)}, b_{n3}^{(j)}, b_{n4}^{(j)} \right)^T$$

Then the results may be obtained by directly differentiating the formulas of equation (B-34). To aid in this derivation it is required to have the partial derivatives of the elements defined by the formuals of set I. A direct calculation gives at $t = t_o$

$$R^{(j)} = V^{(j)} = 0 \quad , \quad j = 1, 2, 3, \text{ and } 4$$

$$\Lambda^{(j)} = \frac{\lambda_j}{\Lambda} \quad , \quad j = 1, 2$$

$$= 0 \quad , \quad j = 3, 4$$

$$\theta^{(j)} = X_j \quad , \quad j = 1, 2$$

$$= 0 \quad , \quad j = 3, 4$$

$$\theta_1^{(j)} = \dot{X}_j \quad , \quad j = 1, 2$$

$$= 0 \quad , \quad j = 3, 4$$

$$\theta_2^{(j)} = 0 \quad , \quad j = 1, 2$$

$$= X_{j-2} \quad , \quad j = 3, 4$$

$$\theta_3^{(j)} = 0 \quad , \quad j = 1, 2$$

$$= \dot{X}_{j-2} \quad , \quad j = 3, 4$$

$$\theta_4^{(j)} = 0 \quad , \quad j = 1, 2, 3, 4$$

$$\theta_5^{(j)} = \dot{\lambda}_j \quad , \quad j = 1, 2$$

$$= \lambda_{j-2} \quad , \quad j = 3, 4$$

$$\theta_6^{(j)} = 0 \quad , \quad j = 1, 2$$

$$= 2\dot{\lambda}_{j-2} \quad , \quad j = 3, 4$$

$$b_1^{(j)} = \frac{-b_1 \lambda_j}{\Lambda^2} \quad , \quad j = 1, 2$$

$$= 0 \quad , \quad j = 3, 4$$

$$b_2^{(j)} = 0 \quad , \quad j = 1, 2, 3, 4$$

$$b_3^{(j)} = \frac{b_2 X_j}{R^2} \quad , \quad j = 1, 2$$

$$= 0 \quad , \quad j = 3, 4$$

$$b_4^{(j)} = \frac{1}{\Lambda^2}\left(\frac{2\theta_5}{\Lambda^2}\lambda_j - \dot{\lambda}_j\right) \quad , \quad j = 1, 2$$

$$= -\frac{1}{\Lambda^2}\lambda_{j-2} \quad , \quad j = 3, 4$$

$$b_5^{(j)} = \frac{2}{\Lambda^4}\left(3\mu b_3 \theta - \frac{4\theta_5^2}{\Lambda^2} + \theta_6\right)\lambda_j + \frac{4\theta_5}{\Lambda^4}\dot{\lambda}_j - \frac{6\mu b_3}{\Lambda^2} X_j \quad , \quad j = 1, 2$$

$$= \frac{2}{\Lambda^2}\left(\frac{2\theta_5}{\Lambda^2}\lambda_{j-2} - \dot{\lambda}_{j-2}\right) \quad , \quad j = 3, 4$$

$$b_6^{(j)} = -\frac{2}{\Lambda^2}\left\{\left[\frac{3\mu b_3}{\Lambda^2}\left(\theta\left\{\frac{12\theta_5}{\Lambda^2} + 5\frac{\theta_4}{R^2}\right\} - 2\left\{\theta_1 + \theta_2\right\}\right)\right.\right.$$

$$\left. + \frac{2\mu b_2}{\Lambda^2}\left(\theta_2 - \frac{2\theta_5}{\Lambda^2}\right) - \frac{\theta_5}{\Lambda^2}\left(3Fb_1 - \frac{12\theta_6}{\Lambda^2} + \frac{20\theta_5^2}{\Lambda^3}\right)\right]\lambda_j$$

$$+ \left[\mu b_2\left(2 - \frac{9\theta^2}{R^2\Lambda^2}\right) + Fb_1 - \frac{3\theta_6}{\Lambda^2} + \frac{12\theta_5^2}{\Lambda^3}\right]\dot{\lambda}_j + \frac{3\mu b_2}{R^2}\left[\theta_1 + \theta_2\right.$$

$$\left.\left. - \frac{5\theta_4\theta}{R^2} - \frac{6\theta\theta_5}{\Lambda^2}\right]X_j + 3\mu b_3\dot{X}_j\right\} \qquad , \quad j = 1,\ 2$$

$$= -\frac{2}{\Lambda^2}\left\{\left[\mu b_2\left(2 - \frac{9\theta^2}{R^2\Lambda^2}\right) + Fb_1 - \frac{3\theta_6}{\Lambda^2} + \frac{12\theta_5^2}{\Lambda^3}\right]\lambda_{j-2} - \frac{6\theta_5}{\Lambda^2}\dot{\lambda}_{j-2}\right.$$

$$\left. + \mu b_2\left[\frac{3\theta}{R^2} - 1\right]X_{j-2}\right\} \qquad , \quad j = 3,\ 4$$

In terms of the above formulas j = 1, 2, 3, 4

$$B_o^{(j)} = B_1^{(j)} = 0$$

$$B_2^{(j)} = \begin{bmatrix} Fb_1^{(j)} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad B_3^{(j)} = \begin{bmatrix} Fb_1^{(j)}b_4 + Fb_1b_4^{(j)} \\ Fb_1^{(j)} \\ 0 \\ 0 \end{bmatrix}, \quad B_4^{(j)} = \begin{bmatrix} Fb_1^{(j)}(b_5 - 2\mu b_2 + b_4^2) + Fb_1(b_5^{(j)} + 2b_4b_4^{(j)}) \\ 2(Fb_1^{(j)}b_4 + Fb_1b_4^{(j)}) \\ 6\mu b_3^{(j)}Fb_1 + 6\mu b_3Fb_1^{(j)} \\ 0 \end{bmatrix}$$

$$B_5^{(j)} = \begin{bmatrix} Fb_1^{(j)}\left(b_6 + \frac{12\mu b_2\theta_4}{R^2} + 3b_4b_5 - 4\mu b_2b_4 + b_4^3\right) \\ \quad + Fb_1\left(b_6^{(j)} + 3b_4^{(j)}b_5 + 3b_4b_5^{(j)} - 4\mu b_2b_4^{(j)} + 3b_4^2b_4^{(j)}\right) \\ Fb_1^{(j)}\left(3b_4^2 - 2\mu b_2 + 3b_5\right) + Fb_1\left(6b_4b_4^{(j)} + 3b_5^{(j)}\right) \\ Fb_1^{(j)}\left(\frac{b_2}{R^2}\left[2\theta_1 + \theta_2 - \frac{10\theta_4\theta_4}{R^2}\right] + 2b_3b_4\right)6\mu \\ \quad + Fb_1\left(\frac{b_2}{R^2}\left[2\theta_1^{(j)} + \theta_2^{(j)} - \frac{10\theta_4\theta_4^{(j)}}{R^2}\right] + 2b_3^{(j)}b_4 + 2b_3b_4^{(j)}\right)6\mu \\ 12\mu b_3^{(j)}Fb_1 + 12\mu b_3Fb_1^{(j)} \end{bmatrix}$$

B.3.5.4 Some Auxiliary Formulas

It is required to have expressions for various time derivatives of $P_j(X_o)$ and $P_j(\dot{X}_o)$ defined by equation (B-33) in order to complete the formulas for ζ, α, and β. It is required to have

$$P_j^{(n)}(X_o), \quad j = 1, \ldots, 4; \quad n = 1, \ldots, 5.$$

$$P_j^{(n)}(\dot{X}_o), \quad j = 1, \ldots, 4; \quad n = 1, \ldots, 4.$$

Using the formulas for B_n and $B_n^{(j)}$ calculate

$$\begin{aligned} X_o^{(m)}\cdot X_o^{(n)(j)} = {} & b_{m1}b_{n1}^{(j)}\Lambda^2 + b_{m1}b_{n1}\Lambda^{(j)}\Lambda + \left[b_{m1}b_{n2}^{(j)} + b_{m2}b_{n1}^{(j)}\right]\theta_5 \\ & + b_{m1}b_{n2}\lambda_o\cdot\dot{\lambda}_o^{(j)} + \left[b_{m1}b_{n3}^{(j)} + b_{m3}b_{n1}^{(j)}\right]\theta + \left[b_{m1}b_{n4}^{(j)} + b_{m4}b_{n1}^{(j)}\right]\theta_1 \\ & + b_{m2}b_{n1}\dot{\lambda}_o\cdot\lambda_o^{(j)} + b_{m2}b_{n2}^{(j)}\theta_6 + \frac{1}{2}b_{m2}b_{n2}\theta_6^{(j)} + \left[b_{m2}b_{n3}^{(j)} \right. \\ & \left. + b_{m3}b_{n2}^{(j)}\right]\theta_2 + \left[b_{m2}b_{n4}^{(j)} + b_{m4}b_{n2}^{(j)}\right]\theta_3 + b_{m3}b_{n1}\theta^{(j)} + b_{m3}b_{n2}\theta_2^{(j)} \\ & + b_{m3}b_{n3}^{(j)}R^2 + \left[b_{m3}b_{n4}^{(j)} + b_{m4}b_{n3}^{(j)}\right]\theta_4 + b_{m4}b_{n1}\theta_1^{(j)} + b_{m4}b_{n2}\theta_3^{(j)}\ . \end{aligned}$$

Proceeding with $X_o^{(j)} = \dot{X}_o^{(j)} = 0,\ j = 1, \ldots, 4$

$$P_j(X_o) = 0$$

$$\dot{P}_j(X_o) = 0$$

$$\ddot{P}_j(X_o) = X_o \cdot \ddot{X}_o^{(j)} = b_{03}b_{21}^{(j)} + b_{03}b_{21}{}^{(j)}$$

$$\dddot{P}_j(X_o) = 3\dot{X}_o \cdot \ddot{X}_o^{(j)} + X_o \cdot \dddot{X}_o^{(j)}$$

$$= 3\left(b_{14}b_{21}^{(j)}\theta_1 + b_{14}b_{21}\theta_1^{(j)}\right) + b_{03}b_{31}^{(j)}\theta_2 + b_{03}b_{31}\theta^{(j)} + b_{03}b_{32}\theta_2^{(j)}$$

$$P_j^{(iv)}(X_o) = 6\ddot{X}_o \cdot \ddot{X}_o^{(j)} + 4\dot{X}_o \cdot \dddot{X}_o^{(j)} + X_o \cdot X_o^{(iv)(j)}$$

$$= 6\left(b_{21}b_{21}^{(j)}\Lambda^2 + b_{21}^2\Lambda^{(j)}\Lambda + b_{23}b_{21}^{(j)}\theta + b_{23}b_{21}\theta^{(j)}\right) + 4\left(b_{14}b_{31}^{(j)}\theta_1\right.$$

$$\left. + b_{14}b_{32}^{(j)}\theta_3 + b_{14}b_{31}\theta_1^{(j)} + b_{14}b_{32}\theta_3^{(j)}\right) + b_{03}b_{41}^{(j)}\theta + b_{03}b_{42}^{(j)}\theta_2$$

$$+ b_{03}b_{41}^{(j)}\theta^{(j)} + b_{03}b_{42}\theta_2^{(j)} + b_{03}b_{43}^{(j)}R^2$$

$$P_j^{(iv)}(X_o) = 6b_{21}b_{21}^{(j)}\Lambda^2 + 6b_{21}^2\Lambda^{(j)}\Lambda + \left(6b_{23}b_{21}^{(j)} + b_{03}b_{41}^{(j)}\right)\theta$$

$$+ \left(6b_{23}b_{21} + b_{03}b_{41}^{(j)}\right)\theta^{(j)} + 4\left(b_{14}b_{31}^{(j)}\theta_1 + b_{14}b_{32}^{(j)}\theta_3 + b_{14}b_{31}\theta_1^{(j)}\right.$$

$$\left. + b_{14}b_{32}\theta_3^{(j)}\right) + b_{03}b_{42}^{(j)}\theta_2 + b_{03}b_{42}\theta_2^{(j)} + b_{03}b_{43}^{(j)}R^2$$

$$P_j^{(v)}(X_o) = 10\dddot{X}_o \cdot \ddot{X}_o^{(j)} + 10\ddot{X}_o \cdot \dddot{X}_o^{(j)} + 5\dot{X}_o \cdot X_o^{(iv)(j)} + X_o \cdot X^{(v)(j)}$$

$$= 10\left(b_{31}b_{21}^{(j)}\Lambda^2 + b_{31}b_{21}\Lambda^{(j)}\Lambda + b_{32}b_{21}^{(j)}\theta_5 + b_{33}b_{21}^{(j)}\theta + b_{34}b_{21}^{(j)}\theta_1\right.$$

$$\left. + b_{32}b_{21}\dot{\lambda}_o \cdot \lambda_o^{(j)} + b_{33}b_{21}\theta^{(j)} + b_{34}b_{21}\theta_1^{(j)}\right) + 10\left(b_{21}b_{31}^{(j)}\Lambda^2\right.$$

$$+ b_{21}b_{31}\Lambda^{(j)}\Lambda + b_{21}b_{32}^{(j)}\theta_5 + b_{21}b_{32}\lambda_o \cdot \dot{\lambda}_o^{(j)} + b_{23}b_{31}^{(j)}\theta + b_{23}b_{32}^{(j)}\theta_2$$

$$+ b_{23}b_{31}\theta^{(j)} + b_{23}b_{32}\theta_2^{(j)}\Big) + 5\Big(b_{14}b_{42}^{(j)}\theta_3 + b_{14}b_{43}^{(j)}\theta_4 + b_{14}b_{41}\theta_1^{(j)}$$

$$+ b_{14}b_{42}\theta_3^{(j)}\Big) + b_{03}b_{51}^{(j)}\theta + b_{03}b_{52}^{(j)}\theta_2 + b_{03}b_{51}\theta^{(j)} + b_{03}b_{52}\theta_2^{(j)}$$

$$+ b_{03}b_{53}^{(j)}R^2 + b_{03}b_{54}^{(j)}\theta_4$$

$$= 10\left[b_{31}b_{21}^{(j)} + b_{21}b_{31}^{(j)}\right]\Lambda^2 + 20b_{31}b_{21}\Lambda^{(j)}\Lambda + 10\left[b_{32}b_{21}^{(j)}\right.$$

$$\left. + b_{21}b_{32}^{(j)}\right]\theta_5 + \left[10\left(b_{33}b_{21}^{(j)} + b_{23}b_{31}^{(j)}\right) + b_{03}b_{51}^{(j)}\right]\theta + 10b_{34}b_{21}^{(j)}\theta_1$$

$$+ 20b_{32}b_{21}\dot{\lambda}_o\cdot\lambda_o^{(j)} + \left[10\left(b_{33}b_{21} + b_{23}b_{31}\right) + b_{03}b_{51}\right]\theta^{(j)}$$

$$+ 5\left[2b_{34}b_{21} + b_{14}b_{41}\right]\theta_1^{(j)} + \left[10b_{23}b_{32}^{(j)} + b_{03}b_{52}^{(j)}\right]\theta_2$$

$$+ \left[10b_{23}b_{32} + b_{03}b_{52}\right]\theta_2^{(j)} + 5b_{14}b_{42}^{(j)}\theta_3$$

$$+ \left[5b_{14}b_{43}^{(j)} + b_{03}b_{54}^{(j)}\right]\theta_4 + 5b_{14}b_{42}\theta_3^{(j)} + b_{03}b_{53}^{(j)}R^2$$

$$P_j(\dot{X}_o) = 0$$

$$\dot{P}_j(\dot{X}_o) = \dot{X}_o\cdot\ddot{X}_o^{(j)} = b_{14}b_{21}^{(j)}\ {}_1 + b_{14}b_{21}\theta_1^{(j)}$$

$$\ddot{P}_j(\dot{X}_o) = 2\ddot{X}_o\cdot\ddot{X}_o^{(j)} + \dot{X}_o\cdot\dddot{X}_o^{(j)}$$

$$= 2\left(b_{21}b_{21}^{(j)}\Lambda^2 + b_{21}^2\Lambda^{(j)}\Lambda + b_{23}b_{21}^{(j)}\theta + b_{23}b_{21}\theta^{(j)}\right) + b_{14}b_{31}^{(j)}\theta_1$$

$$+ b_{14}b_{32}^{(j)}\theta_3 + b_{14}b_{31}\theta_1^{(j)} + b_{14}b_{32}\theta_3^{(j)}$$

$$\dddot{P}_j(\dot{X}_o) = 6\dddot{X}_o\cdot\ddot{X}_o^{(j)} + 3\ddot{X}_o\cdot\dddot{X}_o^{(j)} + \dot{X}_o\cdot X_o^{(iv)(j)}$$

$$= 6\left(b_{31}b_{21}^{(j)}\Lambda^2 + b_{31}b_{21}\Lambda^{(j)}\Lambda + b_{32}b_{21}^{(j)}\theta_5 + b_{33}b_{21}^{(j)}\theta + b_{34}b_{21}^{(j)}\theta_1\right.$$

$$\left. + b_{32}b_{21}\dot{\lambda}_o\cdot\lambda_o^{(j)} + b_{33}b_{21}\theta^{(j)} + b_{34}b_{21}\theta_1^{(j)}\right) + 3\left(b_{21}b_{31}^{(j)}\Lambda^2\right.$$

$$+ b_{21}b_{31}\Lambda^{(j)}\Lambda + b_{21}b_{32}^{(j)}\theta_5 + b_{21}b_{32}\lambda_o\cdot\dot{\lambda}_o^{(j)} + b_{23}b_{31}^{(j)}\theta$$

$$+ b_{23}b_{32}^{(j)}\theta_2 + b_{23}b_{31}\theta^{(j)} + b_{23}b_{32}\theta_2^{(j)}\Big) + b_{14}b_{43}\theta_4$$

$$+ b_{14}b_{41}\theta_1^{(j)} + b_{14}b_{42}\theta_3^{(j)}$$

$$= 3\left(2b_{31}b_{21}^{(j)} + b_{21}b_{31}^{(j)}\right)\Lambda^2 + 9b_{31}b_{21}\Lambda^{(j)}\Lambda + 3\Big(2b_{32}b_{21}^{(j)}$$

$$+ b_{21}b_{32}^{(j)}\Big)\theta_5 + 3\left(2b_{33}b_{21}^{(j)} + b_{23}b_{31}^{(j)}\right)\theta + 6b_{34}b_{21}^{(j)}\theta_1$$

$$+ 6b_{32}b_{21}\dot{\lambda}_o\cdot\lambda_o^{(j)} + 3\left(2b_{33}b_{21} + b_{23}b_{31}\right)\theta^{(j)} + \Big(6b_{34}b_{21}$$

$$+ b_{14}b_{41}\Big)\theta_1^{(j)} + 3b_{21}b_{32}\lambda_o\cdot\dot{\lambda}_o^{(j)} + 3b_{23}b_{32}^{(j)}\theta_2 + 3b_{23}b_{32}\theta_2^{(j)}$$

$$+ b_{14}b_{43}^{(j)}\theta_4 + b_{14}b_{42}\theta_3^{(j)}$$

$$P_j^{(iv)}(\dot{X}_o) = 10X_o^{(iv)}\cdot\ddot{X}_o^{(j)} + 10\dddot{X}_o\cdot\dddot{X}_o^{(j)} + 4\ddot{X}_o\cdot X_o^{(iv)} + \dot{X}_o\cdot X_o^{(v)(j)}$$

$$= 10\Big(b_{41}b_{21}^{(j)}\Lambda^2 + b_{41}b_{21}\Lambda^{(j)}\Lambda + b_{42}b_{21}^{(j)}\theta_5 + b_{43}b_{21}^{(j)}\theta + b_{44}b_{21}^{(j)}\theta_1$$

$$+ b_{42}b_{21}\dot{\lambda}_o\cdot\lambda_o^{(j)} + b_{43}b_{21}\theta^{(j)} + b_{44}b_{21}\theta_1^{(j)}\Big) + 10\Big(b_{31}b_{31}^{(j)}\Lambda^2$$

$$+ b_{31}b_{31}\Lambda^{(j)}\Lambda + \left[b_{31}b_{32}^{(j)} + b_{32}b_{31}^{(j)}\right]\theta_5 + b_{31}b_{32}\lambda_o\cdot\dot{\lambda}_o^{(j)} + b_{33}b_{31}^{(j)}\theta$$

$$+ b_{34}b_{31}^{(j)}\theta_1 + b_{32}b_{31}\dot{\lambda}_o\cdot\lambda_o^{(j)} + b_{32}b_{32}^{(j)}\theta_6 + \frac{1}{2}b_{32}b_{32}\theta_6^{(j)}$$

$$+ b_{33}b_{32}^{(j)}\theta_2 + b_{34}b_{32}^{(j)}\theta_3 + b_{33}b_{31}\theta^{(j)} + b_{33}b_{32}\theta_2^{(j)} + b_{34}b_{31}\theta_1^{(j)}$$

$$+ b_{34}b_{32}\theta_3^{(j)}\Big) + 4\Big(b_{21}b_{41}^{(j)\,2} + b_{21}b_{41}\Lambda^{(j)}\Lambda + b_{21}b_{42}^{(j)}\theta_5$$

$$+ b_{21}b_{42}\lambda_o\cdot\dot{\lambda}_o^{(j)} + \left[b_{21}b_{43}^{(j)} + b_{23}b_{41}^{(j)}\right]\theta + b_{23}b_{42}^{(j)}\theta_2$$

$$+ b_{23}b_{41}\theta^{(j)} + b_{23}b_{42}\theta_2^{(j)} + b_{23}b_{43}^{(j)}R^2\Big) + b_{14}b_{51}^{(j)}\theta_1$$

$$+ b_{14}b_{52}^{(j)}\theta_3 + b_{14}b_{53}^{(j)}\theta_4 + b_{14}b_{51}\theta_1^{(j)} + b_{14}b_{52}\theta_3^{(j)}$$

$$= 2\left(5b_{41}b_{21}^{(j)} + 5b_{31}b_{31}^{(j)} + 2b_{21}b_{41}^{(j)}\right)\Lambda^2 + 2\left(5b_{41}b_{21}\right.$$

$$\left. + 5b_{31}^2 + 2b_{21}b_{41}\right)\Lambda^{(j)}\Lambda + 2\left(5b_{42}b_{21}^{(j)} + 5\left[b_{31}b_{32}^{(j)}\right.\right.$$

$$\left.\left. + b_{32}b_{31}^{(j)}\right] + 2b_{21}b_{42}^{(j)}\right)\theta_5 + 2\left(5b_{43}b_{21}^{(j)} + 5b_{33}b_{31}^{(j)}\right.$$

$$\left. + 2\left[b_{21}b_{43}^{(j)} + b_{23}b_{41}^{(j)}\right]\right)\theta + \left(10b_{44}b_{21}^{(j)} + 10b_{34}b_{31}^{(j)}\right.$$

$$\left. + b_{14}b_{51}^{(j)}\right)\theta_1 + 10\left(b_{42}b_{21} + b_{32}b_{31}\right)\dot{\lambda}_o \cdot \lambda_o^{(j)} + 2\left(5b_{43}b_{21}\right.$$

$$\left. + 5b_{33}b_{31} + 2b_{23}b_{41}\right)\theta^{(j)} + \left(10b_{44}b_{21} + 10b_{34}b_{31} + b_{14}b_{51}\right)\theta_1^{(j)}$$

$$+ \left(10b_{31}b_{32} + 4b_{21}b_{42}\right)\lambda_o \cdot \dot{\lambda}_o^{(j)} + 10b_{32}b_{32}^{(j)}\theta_6 + 5b_{32}^2\theta_6^{(j)}$$

$$+ 2\left(5b_{33}b_{32}^{(j)} + 2b_{23}b_{42}^{(j)}\right)\theta_2 + \left(10b_{34}b_{32}^{(j)} + b_{14}b_{52}^{(j)}\right)\theta_3$$

$$+ 2\left(5b_{33}b_{32} + 2b_{23}b_{42}\right)\theta_2^{(j)} + \left(10b_{34}b_{32} + b_{14}b_{52}\right)\theta_3^{(j)}$$

$$+ 4b_{23}b_{43}^{(j)}R^2 + b_{14}b_{53}^{(j)}\theta_4$$

Making some substitutions

$$\ddot{P}_j(X_o) = Fb_1^{(j)}\theta + Fb_1\theta^{(j)}$$

$$\dddot{P}_j(X_o) = 3\left(Fb_1^{(j)}\theta_1 + Fb_1\theta_1^{(j)}\right) + \left(Fb_1^{(j)}b_4 + Fb_1b_4^{(j)}\right)\theta_2 + Fb_1b_4\theta^{(j)}$$

$$+ Fb_1\theta_2^{(j)}$$

$$P_j^{(iv)}(X_o) = 6F^2b_1b_1^{(j)}\Lambda^2 + F^2b_1^2\Lambda^{(j)}\Lambda + \left(b_{41}^{(j)} - 6\mu Fb_2b_1^{(j)}\right)\theta + \left(b_{41}^{(j)}\right.$$

$$\left. - 6\mu Fb_1b_2\right)\theta^{(j)} + 4\left(b_{31}^{(j)}\theta_1 + Fb_1^{(j)}\theta_3 + Fb_1b_4\theta_1^{(j)} + Fb_1\theta_3^{(j)}\right)$$

$$+ b_{42}^{(j)}\theta_2 + 2Fb_1b_4\theta_2^{(j)} + b_{43}^{(j)}R^2$$

$$P_j^{(v)}(X_o) = 10\left(F^2b_1b_4b_1^{(j)} + Fb_1b_{31}^{(j)}\right)\Lambda^2 + 20F^2b_1^2b_4\Lambda^{(j)}\Lambda + 20F^2b_1b_1^{(j)}\theta_5$$

$$+ \left(10\left[Fb_1^{(j)}b_{33} - \mu b_2b_{31}^{(j)}\right] + b_{51}^{(j)}\right)\theta - 10\ b_2Fb_1^{(j)}\theta_1$$

$$+ 20F^2b_1^2\dot{\lambda}_o\cdot\lambda_o^{(j)} + \left(10\left[Fb_1b_{33} - \mu b_2Fb_1b_4\right] + b_{51}\right)\theta^{(j)}$$

$$+ 5\left(b_{41} - 2\mu b_2Fb_1\right)\theta_1^{(j)} + \left(b_{52}^{(j)} - 10\mu b_2Fb_1^{(j)}\right)\theta_2$$

$$+ \left(b_{52} - 10\mu b_2Fb_1\right)\theta_2^{(j)} + 5b_{42}^{(j)}\theta_3 + \left(5b_{43}^{(j)} + b_{54}^{(j)}\right)\theta_4$$

$$+ 10Fb_1b_4\theta_3^{(j)} + b_{53}^{(j)}R^2$$

$$\dot{P}_j(\dot{X}_o) = Fb_1^{(j)}\theta_1 + Fb_1\theta_1$$

$$\ddot{P}_j(\dot{X}_o) = 2\left(F^2b_1b_1^{(j)}\Lambda^2 + F^2b_1^2\Lambda^{(j)}\Lambda - \mu b_2Fb_1^{(j)}\theta - \mu b_2Fb_1\theta^{(j)}\right)$$

$$+ b_{31}^{(j)}\theta_1 + Fb_1^{(j)}\theta_3 + Fb_1b_4\theta_1^{(j)} + Fb_1\theta_3$$

$$\dddot{P}_j(\dot{X}_o) = 3\left(2F^2b_1b_4b_1^{(j)} + Fb_1b_{31}^{(j)}\right)\Lambda^2 + 9F^2b_1^2b_4\Lambda^{(j)}\Lambda + 9F^2b_1b_1^{(j)}\theta_5$$

$$+ 3\left(2b_{33}Fb_1^{(j)} - \mu b_2Fb_1^{(j)}\right)\theta - 6\mu b_2Fb_1^{(j)}\theta_1 + 6F^2b_1^2\dot{\lambda}_o\cdot\lambda_o^{(j)}$$

$$+ 3\left(2b_{33}Fb_1 - \mu b_2Fb_1\right)\theta^{(j)} + \left(b_{41} - 6\mu b_2Fb_1\right)\theta_1^{(j)}$$

$$+ 3F^2b_1^2\lambda_o\cdot\dot{\lambda}_o^{(j)} - 3\mu b_2Fb_1^{(j)}\theta_2 - 3\mu b_2Fb_1\theta_2^{(j)} + b_{43}^{(j)}\theta_4 + 2Fb_1b_4\theta_3^{(j)}$$

$$P_j^{(iv)}(\dot{X}_o) = 2\left(5Fb_1^{(j)}b_{41} + 5Fb_1b_4b_{31}^{(j)} + 2Fb_1b_{41}^{(j)}\right)\Lambda^2 + 2\left(7Fb_1b_{41}\right.$$

$$\left. + 5F^2b_1^2b_4^2\right)\Lambda^{(j)}\Lambda + 2\left(15F^2b_1b_4b_1^{(j)} + 9Fb_1b_{31}^{(j)}\right)\theta_5$$

$$+ 2\left(5Fb_1^{(j)}b_{43} + 5Fb_1^{(j)}b_{33} + 2\left[Fb_1b_{43}^{(j)} - \mu b_2b_{41}^{(j)}\right]\right)\theta$$

$$+ \left(10Fb_1^{(j)}b_{44} + 10b_{34}b_{31}^{(j)} + b_{51}^{(j)}\right)\theta_1 + 30F^2b_1^2b_4\lambda_o\cdot\lambda_o^{(j)}$$

$$+ 2\left(5Fb_1b_{43} + 5Fb_1b_4b_{33} - 2\mu b_2b_{41}\right)\theta^{(j)} + \left(10Fb_1b_{44}\right.$$

$$\left. + 10Fb_1b_4b_{34} + b_{51}\right)\theta_1^{(j)} + 18F^2b_1^2b_4\lambda_o\cdot\dot{\lambda}_o^{(j)} + 10F^2b_1b_1^{(j)}\theta_6$$

$$+ 5F^2b_1^2\theta_6^{(j)} + 2\left(5Fb_1^{(j)}b_{33} - 2\mu b_2b_{42}^{(j)}\right)\theta_2 + \left(-10\mu b_2Fb_1^{(j)}\right.$$

$$\left. + b_{52}^{(j)}\right)\theta_3 + 2\left(5Fb_1b_{33} - 4\mu b_2Fb_1b_4\right)\theta_2^{(j)} + \left(b_{52} - 10\mu b_2Fb_1\right)\theta_3^{(j)}$$

$$- 4\mu b_2b_{43}^{(j)}R^2 + b_{53}^{(j)}\theta_4$$

B.3.6 SUMMARY OF EQUATIONS AND FORMULAS

The necessary formulas that provide an improved approximation, under proper conditions, to ξ by Newton's method are listed here for convenience.

NOTATION

X = position vector

$\dot{X}$ = velocity vector

λ =
$-\dot{\lambda}$ = } Lagrange multipliers

m = mass

β = mass flow rate

R_{co} = radius at cutoff

V_{co} = velocity at cutoff

t_o = initial time

t_f = final time

μ = Gaussian gravitational constant

Set I

Initial Inner Products:

$$R^2 = X \cdot X$$
$$V^2 = \dot{X} \cdot \dot{X}$$
$$\Lambda^2 = \lambda \cdot \lambda$$
$$\theta = X \cdot \lambda$$
$$\theta_1 = \dot{X} \cdot \lambda$$
$$\theta_2 = X \cdot \dot{\lambda}$$
$$\theta_3 = \dot{X} \cdot \dot{\lambda}$$
$$\theta_4 = X \cdot \dot{X}$$
$$\theta_5 = \lambda \cdot \dot{\lambda}$$
$$\theta_6 = \dot{\lambda} \cdot \dot{\lambda}$$

Coefficients of Motion and Euler-Lagrange Equations

$$b_1 = \frac{1}{m\Lambda}$$

$$b_2 = \frac{1}{R^3}$$

$$b_3 = \frac{b_2\theta}{R^2}$$

$$\dot{b}_1 = b_1 b_4 = b_1\left(\frac{\beta}{m} - \frac{\theta_5}{\Lambda^2}\right)$$

$$\dot{b}_4 = b_5 = \frac{\beta^2}{m^2} - \frac{1}{\Lambda^2}\left[3\mu b_3\theta + \theta_6 - \mu b_2\Lambda^2 - \frac{2\theta_5^2}{\Lambda^2}\right]$$

$$\dot{b}_5 = b_6 = \frac{2\beta^3}{m^3} - \frac{1}{\Lambda^2}\left[3\mu b_3\left(2\left[\theta_1 + 2\theta_2\right] - \theta\left[\frac{5\theta_4}{R^2} + \frac{6\theta_5}{\Lambda^2}\right]\right) + \mu b_2\left(\frac{3\theta_4\Lambda^2}{R^2} + 2\theta_5\right) - \frac{2\theta_5}{\Lambda^2}\left(3\theta_6 - \frac{4\theta_5^2}{\Lambda^2}\right)\right]$$

Set II

Radius Equation:

$$F_1 = R_1 + R_2\Delta t + R_3\Delta t^2 + R_4\Delta t^3 + R_5\Delta t^4 + R_6\Delta t^5 = 0$$

$$\Delta t = t_f - t_o$$

$$R_1 = R^2 - R_{co}^2$$

$$R_2 = 2\theta_4$$

$$R_3 = Fb_1\theta - \mu b_2 R^2 + V^2$$

$$R_4 = \frac{1}{3}\left\{Fb_1\left(b_4\theta + \theta_2 + 3\theta_1\right) - \mu b_2\theta_4\right\}$$

$$R_5 = \frac{1}{12}\left\{Fb_1\left(\left[b_4^2 + b_5 - 2\mu b_2\right]\theta + 4b_4\theta_1 + 2b_4\theta_2 + 4\theta_3 + 3Fb_1\Lambda^2\right) - \mu b_2\left(V^2 - 3\frac{\theta_4^2}{R^2} - \mu b_2 R^2\right)\right\}$$

$$R_6 = \frac{1}{60}\left\{Fb_1\left(\left[b_4^3 + 3b_4b_5 - 2\mu b_2 b_4 + b_6 + \frac{24\mu b_2\theta_4}{R^2}\right]\theta + \left[5b_4^2 + 5b_5 - 8\mu b_2\right]\theta_1 + \left[3b_4^2 + 3b_5 - 6\mu b_2\right]\theta_2 + 10b_4\theta_3 + 10Fb_1\left[\theta_5 + b_4\Lambda^2\right]\right) + \frac{3\mu b_2\theta_4}{R^2}\left(3V^2 - \frac{5\theta_4^2}{R^2} - 3\mu b_2 R^2\right) + \mu^2 b_2^2\theta_4\right\}$$

Velocity Equation:

$$f_2 = V_1 + V_2\Delta t + V_3\Delta t^2 + V_4\Delta t^3 + V_5\Delta t^4 = 0$$

$$V_1 = V^2 - V_{co}^2$$

$$V_2 = 2\left(Fb_1\theta_1 - \mu b_2\theta_4\right)$$

$$V_3 = Fb_1\left(-2\mu b_2\theta + b_4\theta_1 + \theta_3 + Fb_1\Lambda^2\right) - \mu b_2\left(V^2 - \mu b_2R^2 - \frac{3\theta_4^2}{R^2}\right)$$

$$V_4 = \frac{1}{3}\Bigg\{Fb_1\left(3\left[\frac{5\mu b_2\theta_4}{R^2} - \mu b_2 b_4\right]\theta + \left[b_4^2 + b_5 - 5\mu b_2\right]\theta_1 - 3\mu b_2\theta_2 + 2b_4\theta_3 + 3Fb_1\left[\theta_5 + b_4\Lambda^2\right]\right) + \frac{3\mu b_2\theta_4}{R^2}\left(3V^2 - \frac{5\theta_4^2}{R^2} - 3\mu b_2R^2\right) + \mu^2 b_2^2\theta_4\Bigg\}$$

$$V_5 = \frac{1}{12}\Bigg\{Fb_1\Bigg(2\mu b_2\left[\frac{15b_4\theta_4}{R^2} - 2\left(b_4^2 + b_5\right) - 12\mu b_2 - \frac{60\theta_4^2}{R^4} + 12Fb_1b_3 + 12\frac{V^2}{R^2}\right]\theta + \left[\frac{48\mu b_2\theta_4}{R^2} - 10\mu b_2 b_4 + b_4^3 + 3b_4b_5 + b_6\right]\theta_1 + 8\mu b_2\left[\frac{3\theta_4}{R^2} - b_4\right]\theta_2 + \left[3b_4^3 + 3b_5 - 8\mu b_2\right]\theta_3 + 14Fb_1b_4\theta_5 + 3Fb_1\theta_6 + \left[7b_4^2 + 4b_5 - 8\mu b_2\right]Fb_1\Lambda^2\Bigg) + \frac{3\mu b_2}{R^2}\left(\left[V^2 - \mu b_2R^2 - \frac{5\theta_4^2}{R^2}\right]\cdot\left[3V^2 - 3\mu b_2R^2 - \frac{5\theta_4^2}{R^2}\right] - \frac{5\theta_4^2}{R^2}\left[2V^2 - \mu b_2R^2 - \frac{2\theta_4^2}{R^2}\right]\right) + \mu b_2\left[V^2 - \mu b_2R^2\right]\Bigg\}$$

Orthogonality Equation:

$$f_3 = \frac{1}{2}R_2 + R_3\Delta t + \frac{3}{2}R_4\Delta t^2 + 2R_5\Delta t^3 + \frac{5}{2}R_6\Delta t^4 = 0$$

Newton-Raphson Iteration Formulas - (Cramer's Rule)

$$\xi = \begin{bmatrix} \lambda_{10} \\ \lambda_{20} \\ \dot{\lambda}_{10} \\ \dot{\lambda}_{20} \\ t_f \end{bmatrix} , \qquad \xi^* = \xi + \Delta\xi$$

$$\Delta\xi_i = -\frac{\det(A_i)}{\det(A)} , \qquad i = 1, 2, \ldots, 5.$$

$$\det(A) = 8\lambda_o \cdot \left(-\alpha_{34}\dot{X}_o + \begin{bmatrix} \alpha_{23} & \alpha_{24} \\ -\alpha_{13} & -\alpha_{14} \end{bmatrix} X_o \right)$$

$$\det(A_1) = 8\lambda_{20} X_o \cdot \begin{pmatrix} \beta_3 \\ \beta_4 \end{pmatrix}$$

$$\det(A_2) = -8\lambda_{10} X_o \cdot \begin{pmatrix} \beta_3 \\ \beta_4 \end{pmatrix}$$

$$\det(A_3) = 8\lambda_o \cdot \left(\beta_4 \dot{X}_o + X_{10} \begin{pmatrix} \beta_2 \\ -\beta_1 \end{pmatrix} \right)$$

$$\det(A_4) = -8\lambda_o \cdot \left(\beta_3 \dot{X}_o - X_{20} \begin{pmatrix} \beta_2 \\ -\beta_1 \end{pmatrix} \right)$$

$$\det(A_5) = -8\lambda_o \cdot \left(-\zeta_{34}\dot{X}_o + \begin{bmatrix} \zeta_{23} & \zeta_{24} \\ -\zeta_{13} & \zeta_{14} \end{bmatrix} X_o \right)$$

Set III

Formulas for ζ, α, β.

$$\zeta_{ij} = \zeta^{(1)}_{ijo} + \zeta^{(2)}_{ijo}\,\Delta t + \zeta^{(3)}_{ijo}\Delta t^2 + \ldots + \zeta^{(6)}_{ijo}\Delta t^5$$

$$\zeta^{(1)}_{ijo} = \zeta^{(2)}_{ijo} = \zeta^{(3)}_{ijo} = 0$$

$$\zeta^{(4)}_{ijo} = \frac{1}{4} R_1 \left[\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - \ddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right]$$

$$\zeta^{(5)}_{ijo} = \frac{1}{24} R_1 \left[2\dot{P}_i(\dot{X}_o)P^{(iv)}_j(X_o) - 2\ddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) + 3\ddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) \right.$$
$$\left. - 3\dddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right] + \frac{1}{24} R_2 \left[10\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 7\ddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right]$$
$$- \frac{1}{16} V_1\ddot{P}_i(X_o)\dddot{P}_j(X_o)$$

$$\zeta^{(6)}_{ijo} = \frac{1}{60} R_1 \left[-\frac{5}{2}\ddot{P}_i(X_o)P^{(iv)}_j(\dot{X}_o) + 5\dddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 5\dddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) \right.$$
$$\left. + 5\ddot{P}_i(\dot{X}_o)P^{(iv)}_j(X_o) - 5P^{(iv)}_i(X_o)\ddot{P}_j(\dot{X}_o) + \frac{5}{2}\dot{P}_i(\dot{X}_o)P^{(v)}_j(X_o) \right]$$
$$+ \frac{1}{60} R_2 \left[11\ddot{P}_i(X_o)\dddot{P}_j(\dot{X}_o) + 23\ddot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) - 23\dddot{P}_i(X_o)\ddot{P}_j(\dot{X}_o) \right.$$
$$\left. + 18\dot{P}_i(\dot{X}_o)P^{(iv)}_j(X_o) \right] - \frac{1}{60} V_1 \left[\frac{7}{2}\ddot{P}_i(X_o)P^{(v)}_j(X_o) \right]$$
$$- \frac{1}{60} V_2 \left[13\ddot{P}_i(X_o)\dddot{P}_j(X_o) \right] + \frac{1}{30} R_3 \left[\dot{P}_i(\dot{X}_o)\dddot{P}_j(X_o) \right]$$

$$\alpha_{ij} = \alpha^{(1)}_{ij} + \alpha^{(2)}_{ijo}\Delta t + \ldots + \alpha^{(6)}_{ijo}\Delta t^5$$

the $\alpha_{ijo}^{(n)}$ are obtained by replacing in $\zeta_{ijo}^{(n)}$ R_K by KR_{K+1} and V_K by KV_{K+1}.

$$\beta_i = \beta_{io}^{(1)} + \beta_{io}^{(2)}\Delta t + \beta_{io}^{(3)}\Delta t^2 + \beta_{io}^{(4)}\Delta t^3 + \beta_{io}^{(5)}\Delta t^4$$

$$\beta_{io}^{(1)} = 0$$

$$\beta_{io}^{(2)} = \frac{1}{2} R_1 \left[2R_3\dot{P}_i(\dot{X}_o) - V_2\ddot{P}_i(\dot{X}_o) \right]$$

$$\begin{aligned}
\beta_{io}^{(3)} = {} & \frac{1}{4} R_1 \left[2R_3\ddot{P}_i(\dot{X}_o) - V_2\dddot{P}_i(X_o) \right] - \frac{1}{4} V_1 \left[2R_3\ddot{P}_i(X_o) - R_2\dddot{P}_i(X_o) \right] \\
& + \frac{1}{2} R_2 \left[R_2\ddot{P}_i(\dot{X}_o) - V_2\ddot{P}_i(X_o) \right] + \frac{1}{2} R_1 \left[6R_4\dot{P}_i(\dot{X}_o) - 2V_3\ddot{P}_i(X_o) \right] \\
& + R_2 \left[\frac{1}{2} V_1\ddot{P}_i(X_o) - R_2\dot{P}_i(\dot{X}_o) \right]
\end{aligned}$$

$$\begin{aligned}
\beta_{io}^{(4)} = {} & \frac{1}{12} R_1 \left[2R_3\dddot{P}_i(\dot{X}_o) - V_2P_i^{(iv)}(X_o) \right] - \frac{1}{12} V_1 \left[2R_3\dddot{P}_i(X_o) - R_2P_i^{(iv)}(X_o) \right] \\
& + \frac{1}{6} R_2 \left[V_2\dddot{P}_i(X_o) - R_2\dddot{P}_i(\dot{X}_o) \right] + \frac{1}{4} R_1 \left[6R_4\ddot{P}_i(\dot{X}_o) - 2V_3\dddot{P}_i(X_o) \right] \\
& - \frac{1}{4} V_1 \left[6R_4\ddot{P}_i(X_o) - 2R_3\dddot{P}_i(X_o) \right] + \frac{1}{2} R_2 \left[2V_3\ddot{P}_i(X_o) - 2R_3\ddot{P}_i(\dot{X}_o) \right] \\
& + \frac{1}{4} R_2 \left[6R_4\dot{P}_i(\dot{X}_o) - 2V_3\ddot{P}_i(X_o) \right] + R_3 \left[\frac{1}{2} V_2\ddot{P}_i(X_o) - 2R_3\dot{P}_i(\dot{X}_o) \right] \\
& + \frac{1}{4} R_1 \left[24R_5\dot{P}_i(\dot{X}_o) - 6V_4\ddot{P}_i(X_o) \right] + 3R_4 \left[\frac{1}{2} V_1\ddot{P}_i(X_o) - R_2\dot{P}_i(\dot{X}_o) \right]
\end{aligned}$$

$$\begin{aligned}
\beta_{io}^{(5)} = {} & \frac{1}{48} R_1 \left[2R_3P_i^{(iv)}(\dot{X}_o) - V_2P_i^{(v)}(X_o) \right] - \frac{1}{48} V_1 \left[2R_3P_i^{(iv)}(X_o) - R_2P_i^{(v)}(X_o) \right] \\
& + \frac{1}{24} R_2 \left[V_2P_i^{(iv)}(X_o) - R_2P_i^{(iv)}(X_o) \right] + \frac{1}{12} R_1 \left[6R_4\dddot{P}_i(\dot{X}_o) - 2V_3P_i^{(iv)}(X_o) \right] \\
& - \frac{1}{12} V_1 \left[6R_4\dddot{P}_i(X_o) - 2R_3P_i^{(iv)}(X_o) \right] + \frac{1}{6} R_2 \left[2R_3\dddot{P}_i(\dot{X}_o) - 2V_3\dddot{P}_i(X_o) \right]
\end{aligned}$$

$$+ \frac{1}{8} R_2 \left[6R_4 \ddot{P}_i(\dot{X}_o)) - 2V_3 \dddot{P}_i(X_o) \right] - \frac{1}{8} V_2 \left[6R_4 \ddot{P}_i(X_o) - 2R_3 \dddot{P}_i(X_o) \right]$$

$$+ \frac{1}{2} R_3 \left[2V_3 \ddot{P}_i(X_o) - 2R_3 \ddot{P}_i(\dot{X}_o) \right] + \frac{1}{8} R_1 \left[24R_5 \ddot{P}_i(\dot{X}_o) - 6V_4 \dddot{P}_i(X_o) \right]$$

$$- \frac{1}{8} V_1 \left[24R_5 \ddot{P}_i(X_o) - 6R_4 \dddot{P}_i(X_o) \right] + \frac{1}{4} R_2 \left[6V_4 \ddot{P}_i(X_o) - 6R_4 \ddot{P}_i(\dot{X}_o) \right]$$

$$+ \frac{1}{6} R_2 \left[24R_5 \dot{P}_i(\dot{X}_o) - 6V_4 \ddot{P}_i(X_o) \right] + 2R_4 \left[\frac{1}{2} V_1 \ddot{P}_i(X_o) - R_2 \dot{P}_i(\dot{X}_o) \right]$$

$$+ \frac{1}{12} R_1 \left[120R_6 \dot{P}_i(\dot{X}_o) - 24V_5 \ddot{P}_i(X_o) \right] + 4R_6 \left[\frac{1}{2} V_1 \ddot{P}_i(X_o) - R_2 \dot{P}_i(\dot{X}_o) \right]$$

Appendix C

TABLES OF LAGRANGE MULTIPLIERS

In Section V various numerical tables were presented which listed the thrust direction angle χ and its time derivative $\dot{\chi}$ for several of the guidance formulas discussed in Sections III and IV. However, the guidance formulas were derived to calculate directly the initial values of the Lagrange multipliers λ and $\dot{\lambda}$ from which χ and $\dot{\chi}$ were computed. It is felt that a deeper understanding of the merits of the guidance routines may be obtained by studying the corresponding multipliers.

Here the multipliers corresponding to Tables 5-2, 5-3, 5-4, and 5-5 are presented in Tables C-1, C-2, C-3, and C-4, respectively.

Table C-1. SILBER-HUNT (S-H) EXPANSION

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
1	True Values	.519	-.855	.284	.733
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.919	-.393	-.016	-.110
	S-H N = 2	.704	-.710	.151	.340
2	True Values	.503	-.865	.310	.823
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.918	-.397	-.272	-.151
	S-H N = 2	.702	-.712	.147	.324
3	True Values	.911	.412	-.268	-.700
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.919	.394	-.251	-.646
	S-H N = 2	.907	.421	-.269	-.700
4	True Values	.919	.395	-.285	-.765
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.921	.390	-.265	-.697
	S-H N = 2	.909	.418	-.290	-.775
5	True Values	.976	-.216	-.012	-.032
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.941	-.339	.059	.165
	S-H N = 2	.949	-.316	.027	.069
6	True Values	.930	.367	-.168	-.380
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.947	.321	-.136	-.279
	S-H N = 2	.915	.404	-.191	-.450
7	True Values	.914	.407	-.165	-.357
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.945	.327	-.122	-.227
	S-H N = 2	.889	.459	-.199	-.460

Table C-1. SILBER-HUNT (S-H) EXPANSION (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
8	True Values	.963	-.268	.003	.001
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.939	-.343	.047	.121
	S-H N = 2	.930	-.368	.049	.128
9	True Values	.983	.184	-.299	-.903
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.999	.050	-.194	-.575
	S-H N = 2	.994	.110	-.245	-.737
10	True Values	.966	.258	-.358	-1.086
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.999	.446	-.208	-.625
	S-H N = 2	.987	.162	-.283	-.858
11	True Values	.641	.767	-.332	-.820
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.652	.758	-.349	-.891
	S-H N = 2	.477	.879	-.384	-.978
12	True Values	.628	.778	-.358	-.909
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.654	.756	-.361	-.935
	S-H N = 2	.454	.891	-.415	-1.089
13	True Values	.988	.152	-.124	-.300
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.990	.134	-.107	-.250
	S-H N = 2	.990	.144	-.123	-.303
14	True Values	.811	.585	-.215	-.481
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.692	.722	-.264	-.610
	S-H N = 2	.850	.527	-.148	-.270

Table C-1. SILBER-HUNT (S-H) EXPANSION (Concluded)

	CASE	λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
15	True Values	.806	.592	-.200	-.428
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.689	.725	-.251	-.565
	S-H N = 2	.844	.537	-.123	-.178
16	True Values	.990	.141	-.132	-.335
	Nominal	.974	.228	-.178	-.456
	S-H N = 1	.991	.129	-.121	-.301
	S-H N = 2	.991	.137	-.132	-.336

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
1	Inversion				
	N = 1	.986	-.160	-.108	-.353
	N = 2	.870	-.491	.024	-.020
	N = 3	.713	-.700	.147	.324
	Polynomial Sol.				
	N = 1	.986	-.160	-.108	-.353
	N = 2	.813	-.582	.141	.355
	*N = 3	.976	.216	-.248	-.712
	Newton-Raphson				
	2 Iterations (Damped on 2nd)	.576	-.817	.280	.741
2	Inversion				
	N = 1	.998	-.060	-.149	-.460
	N = 2	.927	-.372	-.042	-.207
	N = 3	.806	-.591	.073	.109
	Polynomial Sol.				
	N = 1	.998	-.060	-.149	-.460
	N = 2	.854	-.519	.132	.351
	*N = 3	.803	-.595	.058	.054
	Newton-Raphson				
	2 Iterations (Damped on 1st)	.667	-.744	.226	.586
	Inversion				
	N = 1	.938	.344	-.296	-.577
	N = 2	.920	.391	-.252	-.650
	N = 3	.913	.406	-.272	-.681

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
3	Polynomial Sol.				
	N = 1	.938	.344	-.296	-.577
	N = 2	.929	.367	-.258	-.679
	N = 3	.906	.422	-.272	-.713
	Newton-Raphson				
	2 Iterations	.910	.413	-.269	-.703
	Inversion				
	N = 1	.939	.342	-.235	-.606
	N = 2	.925	.379	-.263	-.693
	N = 3	.920	.390	-.276	-.735
4	Polynomial Sol.				
	N = 1	.939	.342	-.235	-.606
	N = 2	.967	.252	-.238	-.650
	N = 3	.897	.440	-.307	-.829
	Newton-Raphson				
	2 Iterations	.918	.396	-.284	-.762
	Inversion				
	N = 1	.926	-.377	.082	.244
	N = 2	.953	-.302	.016	-.036
	N = 3	.999	-.038	-.112	-.328
5	Polynomial Sol.				
	N = 1	.926	-.377	.082	.244
	N = 2	.981	-.190	-.013	-.028
	N = 3	.979	-.200	-.016	-.044
	Newton-Raphson				
	2 Iterations	.986	-.163	-.044	-.131

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

	CASE	λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
6	Inversion				
	N = 1	.855	.517	-.195	-.434
	N = 2	.964	.263	-.129	-.276
	N = 3	.915	.401	-.191	-.454
	Polynomial Sol.				
	N = 1	.855	.517	-.195	-.434
	N = 2	.922	.387	-.172	-.387
	N = 3	.929	.367	-.167	-.378
	Newton-Raphson 2 Iterations	.927	.374	-.170	-.388
7	Inversion				
	N = 1	.724	.689	-.219	-.469
	N = 2	.999	-.026	.007	.116
	N = 3	.846	.532	-.273	-.708
	Polynomial Sol.				
	N = 1	.724	.689	-.219	-.469
	N = 2	.887	.456	-.173	-.374
	N = 3	.912	.410	-.165	-.357
	Newton-Raphson 2 Iterations (Damped on 1st)	.914	.405	-.167	-.367
8	Inversion				
	N = 1	.942	-.335	.043	.122
	N = 2	.925	-.379	.056	.150
	N = 3	.978	-.204	-.036	-.116
	Polynomial Sol.				
	N = 1	.942	-.335	.043	.122
	N = 2	.969	-.244	-.002	-.009
	N = 3	.966	-.255	-.001	-.011

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
	Newton-Raphson				
	2 Iterations	.967	-.253	-.007	-.033
9	Inversion				
	N = 1	.954	.297	-.276	-.778
	N = 2	.972	.234	-.290	-.874
	N = 3	.978	.205	-.299	-.895
	Polynomial Sol.				
	N = 1	.954	.297	-.276	-.778
	N = 2	.997	.070	-.252	-.779
	*N = 3	.963	.266	-.313	-.945
	Newton-Raphson				
	2 Iterations	.980	.194	-.298	-.896
10	Inversion				
	N = 1	.933	.357	-.356	-.833
	N = 2	.951	.308	-.339	-.999
	N = 3	.958	.285	-.350	-1.049
	Polynomial Sol.				
	N = 1	.933	.357	-.356	-.833
	*N = 2	.998	.044	-.268	-.848
	*N = 3	.716	.697	-.609	-1.840
	Newton-Raphson				
	2 Iterations	.960	.277	-.354	-1.070
	Inversion				
	N = 1	.653	.756	-.346	-.877
	N = 2	.459	.888	-.387	-.994
	N = 3	.534	.845	-.352	-.875

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
11	Polynomial Sol.				
	N = 1	.653	.756	-.346	-.877
	N = 2	.620	.783	-.352	-.892
	N = 3	.639	.768	-.338	-.843
	Newton-Raphson				
	2 Iterations	.630	.776	-.329	-.809
	Inversion				
	N = 1	.692	.721	-.345	.880
	N = 2	.496	.868	-.407	-1.065
	N = 3	.487	.872	-.404	-1.053
12	Polynomial Sol.				
	N = 1	.692	.721	-.345	.880
	N = 2	.631	.775	-.372	-.963
	N = 3	.630	.776	-.363	-.927
	Newton-Raphson				
	2 Iterations	.583	.811	-.301	-.982
	Inversion				
	N = 1	.994	.105	-.079	-.158
	N = 2	.983	.181	-.153	-.395
	N = 3	.989	.141	-.108	-.249
13	Polynomial Sol.				
	N = 1	.994	.105	-.079	-.158
	N = 2	.984	.172	-.124	-.298
	N = 3	.987	.159	-.125	-.304
	Newton-Raphson				
	2 Iterations	.987	.159	-.125	-.306

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
14	Inversion				
	N = 1	.554	.832	-.290	-.689
	N = 2	.977	.210	.078	.457
	N = 3	.915	.402	-.249	-.660
	Polynomial Sol.				
	N = 1	.554	.832	-.290	-.689
	N = 2	.772	.634	-.230	-.525
	N = 3	.811	.585	-.217	-.488
	Newton-Raphson				
	2 Iterations	.805	.593	-.249	-.602
15	Inversion				
	N = 1	.410	.911	-.289	-.676
	N = 2	.437	-.899	.635	2.004
	N = 3	.945	.325	-.310	-.902
	Polynomial Sol.				
	N = 1	.410	.911	-.289	-.676
	N = 2	.727	.685	-.222	-.485
	N = 3	.802	.597	-.203	-.437
	Newton-Raphson				
	2 Iterations (Damped on 1st)	.839	.542	-.193	-.417
	Inversion				
	N = 1	.994	.101	-.107	-.261
	N = 2	.987	.154	-.141	-.365
	N = 3	.990	.138	-.129	-.326

Table C-2. GUIDANCE FORMULAS WITH DERIVATIVES NUMERICALLY INTEGRATED AND USING NOMINAL STARTING VALUES (Concluded)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
16	Polynomial Sol.				
	N = 1	.994	.101	-.107	-.261
	N = 2	.989	.147	-.132	-.334
	N = 3	.989	.142	-.131	-.334
	Newton-Raphson 2 Iterations	.989	.141	-.131	-.335

Table C-3. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING NOMINAL STARTING VALUES

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
1	Polynomial Sol.				
	N = 1	.987	-.160	-.109	-.355
	*N = 2	.316	.949	-.208	-.344
2	Polynomial Sol.				
	N = 1	.998	-.595	-.150	-.463
	*N = 2	.840	-.543	.132	.341
3	Polynomial Sol.				
	N = 1	.939	.344	-.228	-.577
	N = 2	.933	.360	-.259	-.684
4	Polynomial Sol.				
	N = 1	.939	.343	-.236	-.607
	N = 2	.960	.278	-.246	-.665
5	Polynomial Sol.				
	N = 1	.925	-.379	.083	.245
	N = 2	.982	-.191	-.016	-.036
6	Polynomial Sol.				
	N = 1	.857	.515	-.195	-.433
	N = 2	.921	.390	-.172	-.386
7	Polynomial Sol.				
	N = 1	.726	.687	-.218	-.467
	N = 2	.893	.449	-.173	-.374
8	Polynomial Sol.				
	N = 1	.942	-.337	.044	.123
	N = 2	.964	-.267	.006	.013
9	Polynomial Sol.				
	N = 1	.955	.297	-.277	-.782
	N = 2	.999	.052	-.248	-.770

Table C-3. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING NOMINAL STARTING VALUES (Concluded)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
10	Polynomial Sol.				
	N = 1	.933	.359	-.298	-.836
	N = 2	.998	.645	-.271	-.848
11	Polynomial Sol.				
	N = 1	.653	.757	-.347	-.876
	N = 2	.632	.775	-.353	-.892
12	Polynomial Sol.				
	N = 1	.692	.722	-.346	-.880
	N = 2	.634	.774	-.375	-.971
13	Polynomial Sol.				
	N = 1	.995	.103	-.079	-.158
	N = 2	.985	.172	-.126	-.303
14	Polynomial Sol.				
	N = 1	.554	.832	-.289	-.688
	N = 2	.767	.642	-.232	-.528
15	Polynomial Sol.				
	N = 1	.411	.911	-.289	-.675
	N = 2	.745	.667	-.221	-.481
16	Polynomial Sol.				
	N = 1	.995	.099	-.107	-.259
	N = 2	.990	.144	-.131	-.330

Table C-4. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES

	CASE	λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
	Polynomial Sol.				
	First Guid. Com.				
1	N = 1	.554	-.832	.284	.748
	N = 2	.565	-.825	.270	.670
	Second Guid. Com.				
	N = 1	.522	-.853	.288	.748
	N = 2	.525	-.851	.286	.743
	Polynomial Sol.				
	First Guid. Com.				
2	N = 1	.536	-.844	.308	.825
	N = 2	.550	-.835	.292	.775
	Second Guid. Com.				
	N = 1	.508	-.862	.312	.831
	N = 2	.509	-.861	.312	.830
	Polynomial Sol.				
	First Guid. Com.				
3	N = 1	.911	.413	-.269	-.700
	N = 2	.911	.413	-.269	-.700
	Second Guid. Com.				
	N = 1	.911	.412	-.269	-.700
	N = 2	.911	.412	-.269	-.700
	Polynomial Sol.				
	First Guid. Com.				
4	N = 1	.917	.400	-.286	-.769
	N = 2	.918	.398	-.286	-.766
	Second Guid. Com.				
	N = 1	.918	.396	-.285	-.766
	N = 2	.918	.396	-.285	-.766

Table C-4. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
	Polynomial Sol.				
	First Guid. Com.				
5	N = 1	.974	-.228	-.001	.003
	N = 2	.974	-.225	-.003	-.003
	Second Guidance Com.				
	N = 1	.977	-.213	-.010	-.024
	N = 2	.977	-.214	-.010	-.026
	Polynomial Sol.				
	First Guid. Com.				
6	N = 1	.932	.363	-.168	-.382
	N = 2	.932	.363	-.167	-.378
	Second Guid. Com.				
	N = 1	.930	.366	-.168	-.379
	N = 2	.930	.366	-.168	-.379
	Polynomial Sol.				
	First Guid. Com.				
7	N = 1	.916	.400	-.168	-.368
	N = 2	.917	.399	-.165	-.358
	Second Guid. Com.				
	N = 1	.915	.403	-.164	-.355
	N = 2	.915	.403	-.164	-.355
	Polynomial Sol.				
	First Guid. Com.				
8	N = 1	.960	-.279	.013	.033
	N = 2	.961	-.276	.010	.026
	Second Guid. Com.				
	N = 1	.964	-.267	.004	.007
	N = 2	.963	-.268	.004	.006

Table C-4. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES (Continued)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
	Polynomial Sol.				
	First Guid. Com.				
9	N = 1	.983	.182	-.297	-.895
	N = 2	.984	.179	-.294	-.888
	Second Guid. Com.				
	N = 1	.983	.185	-.299	-.902
	N = 2	.983	.185	-.299	-.902
	Polynomial Sol.				
	First Guid. Com.				
10	N = 1	.967	.254	-.353	-1.069
	N = 2	.968	.251	-.351	-1.064
	Second Guid. Com.				
	N = 1	.966	.257	-.357	-1.083
	N = 2	.966	.257	-.357	-1.083
	Polynomial Sol.				
	First Guid. Com.				
11	N = 1	.735	.678	-.248	-.543
	*N = 2	.245	.970	-.280	-.601
	Second Guid. Com.				
	N = 1	.569	.822	-.365	-.926
	N = 2	.605	.796	-.350	-.877
	Polynomial Sol.				
	First Guid. Com.				
12	N = 1	.756	.655	-.248	-.550
	*N = 2	.567	.824	-.299	-.688
	Second Guid. Com.				
	N = 1	.532	.847	-.398	-1.039
	N = 2	.576	.818	-.382	-.988

Table C-4. GUIDANCE FORMULAS WITH DERIVATIVES OBTAINED BY CORRECTING REFERENCE DERIVATIVES AND USING SILBER-HUNT STARTING VALUES (Concluded)

CASE		λ_1	λ_2	$\dot{\lambda}_1 \times 10^2$	$\dot{\lambda}_2 \times 10^2$
13	Polynomial Sol.				
	First Guid. Com.				
	N = 1	.989	.150	-.123	-.299
	N = 2	.988	.151	-.123	-.299
	Second Guid. Com.				
	N = 1	.988	.153	-.124	-.300
	N = 2	.988	.153	-.124	-.300
14	Polynomial Sol.				
	First Guid. Com.				
	N = 1	.784	.621	-.217	-.481
	N = 2	.797	.604	-.215	-.477
	Second Guid. Com.				
	N = 1	.806	.592	-.216	-.484
	N = 2	.807	.590	-.216	-.484
15	Polynomial Sol.				
	First Guid. Com.				
	N = 1	.773	.635	-.191	-.384
	N = 2	.793	.610	-.190	-.388
	Second Guid. Com.				
	N = 1	.785	.620	-.206	-.440
	N = 2	.787	.617	-.206	-.440
16	Polynomial Sol.				
	First Guid. Com.				
	N = 1	.990	.141	-.132	-.334
	N = 2	.990	.141	-.132	-.335
	Second Guid. Com.				
	N = 1	.990	.141	-.132	-.335
	N = 2	.990	.141	-.132	-.335