AD-A094 811    AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC  F/G 22/3
               THREE BURN INERTIAL UPPER STAGE OPTIMAL ORBIT TRANSFER.(U)
               DEC 80   M N BROWN
UNCLASSIFIED   AFIT/GA/AA/80D-1                                           NL
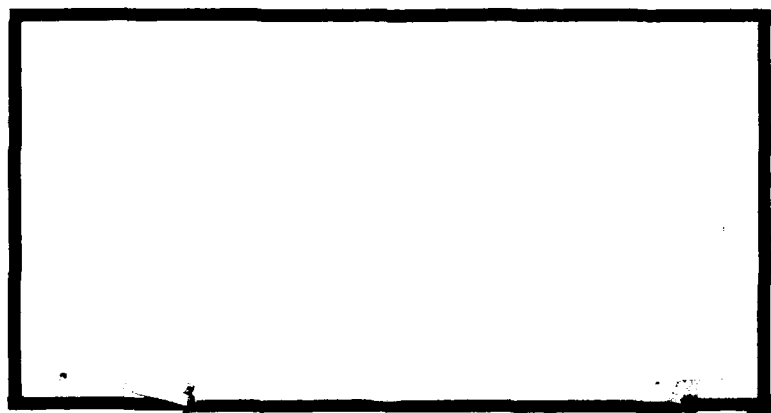
1 OF 1
AD A
094811

END
DATE
FILMED
8-81
DTIC

LEVEL II

DTIC
ELECTE
FEB 1 0 1981
S E D

**UNITED STATES AIR FORCE**
**AIR UNIVERSITY**
# AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

81 2 09 053

THREE BURN INERTIAL UPPER STAGE
OPTIMAL ORBIT TRANSFER

Thesis

AFIT/GA/AA/80D-1    Mark N. Brown
                    Captain   USAF

# THREE BURN INERTIAL UPPER STAGE
# OPTIMAL ORBIT TRANSFER

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science

X

by

Mark N. Brown, B.S.
Capt            USAF

Graduate Astronautical Engineering

December 1980

A

# Contents

## List of Figures

## List of Tables

## List of Symbols

x   -   distance (x-direction)

y   -   distance (y-direction)

z   -   distance (z-direction)

xd   -   velocity (x-direction)

yd   -   velocity (y-direction)

zd   -   velocity (z-direction)

T1   -   coast time from parking orbit fix to first stage firing

A1   -   alpha rotation of first stage alignment

B1   -   beta rotation of the first stage alignment

T2   -   coast time from first stage firing to second stage firing

A2   -   alpha rotation of second stage alignment

B2   -   beta rotation of the second stage alignment

T3   -   coast time from second stage firing to third stage firing

A3   -   alpha rotation of the third stage alignment

B3   -   beta rotation of the third stage alignment

$\Delta V1$   -   change in velocity due to first stage firing

$\Delta V2$   -   change in velocity due to second stage firing

$\Delta V3$   -   change in velocity due to third stage firing

G   -   performance index (Tf)

F   -   augmented performance index

X   -   state vector

M   -   final condition vector

$\nu$   -   Lagrange multiplier vector (final conditions)

A  -  unknown parameter vector

$\dot{a}$  -  da/dt (a arbitrary)

$a^T$ -  a transposed (a arbitrary)

$\mu$  -  geocentric gravitational parameter

AFIT/GA/AA/80D-1

## Abstract

The Inertial Upper Stage (IUS) being developed for use aboard the Space Shuttle is composed of three solid fuel stages plus a satellite payload. One mission of the IUS system is to launch from a shuttle parking orbit and place the satellite in geosynchronous orbit in minimum time. Actual Space Shuttle parking orbit data and IUS characteristics were used in this study to examine the sequential timing and orientation in inertial space of each stage as it is fired while the spacecraft moves along a transfer orbit to geosynchronous orbit. In addition, the sensitivity of the total transfer time and the final orbital state was found as a result of not meeting one or all of the time and orientation parameters.

This problem is unique in that it considers an optimal orbit transfer problem involving solid fuel stages of fixed thrust and burn time. Previous work with liquid fuel engines examined orbital transfers with the intent of minimizing the amount of propellant or required velocity change needed to accomplish the transfer.

## THREE BURN INERTIAL UPPER STAGE
## OPTIMAL ORBIT TRANSFER

### I. Introduction

Previous work involving optimal orbit transfers was centered around liquid fuel engines. The driving concern with a liquid system is to minimize the amount of propellant to be carried on the spacecraft to keep the overall weight as low as possible. This is especially important for spacecraft of one stage, where the engine is required to be restarted for additional burns. There are a number of studies, of which Reference 9 is an example, that explore the transfer from one orbit to another using two impulsive burns. One study was found which considered three burn optimal transfers (Ref 10), but minimized the total velocity change needed.

The advent of the Space Shuttle made it possible to carry a small spacecraft into near earth orbit inside the cargo bay and then launch it from that parking orbit into another orbit. The storage problems associated with liquid fuel systems made solid fuel engines preferable for this type of spacecraft. The initial configuration for the Inertial Upper Stage consisted of two solid fuel engines. This allowed for the use of previous orbital transfer studies to define the exact capabilities of the configuration. The desire to expand this capability led to the design of the

three stage IUS which necessitated the need to examine a
three burn optimal transfer to geosynchronous orbit in mini-
mum time.

As in past studies, the firing of each stage of the
spacecraft is treated as an impulsive burn. This assumption
is justified because the burn time of a rocket engine is very
small when compared to the total time spent in an orbital
transfer. In addition, transfers in the near vicinity of the
earth are normally analyzed in three phases which yield in-
creasing degrees of accuracy.

The first phase is to consider the earth and the space-
craft as being alone in inertial space. The earth is con-
sidered perfectly spherical and homogeneous, and the effects
of the other major bodies such as the sun and moon are ignored.
These assumptions allow for a two-body analysis of an object
under the influence of a single gravitational field. The
equations of motion used to describe the behavior of an
object in this environment are accurate enough to allow for
initial mission planning. The second phase is to take the
two-body equations of motion and add the oblateness and non-
homogeneity effects on the earth. The results are more
accurate, but they differ only slightly from those obtained
from the two-body equations alone. The last step is to again
modify the equations of motion to account for the effects of
the sun, moon, and other major bodies. The end product is,
indeed, accurate but extremely expensive and tedious to extract.

This study follows the tradition of previous work by

assuming impulsive burns and two-body orbital behavior for
this initial mission analysis.

## II.  Problem Statement

The starting place for this problem is the near earth
parking orbit of the Space Shuttle.  The IUS System Program
office at Patrick AFB (Ref 1) was kind enough to provide the
following data describing the shuttle parking orbit at the
moment of orbit attainment:

TABLE I

Initial Parking Orbit Data

| | | |
|---|---|---|
| | latitude | 21.3624° |
| | longitude | 59.2825° |
| | altitude | $7.11693 \times 10^5$ ft |
| | inclination | 28.78871° |
| | eccentricity | 0.01497 |
| | velocity | $2.561851 \times 10^4$ ft/sec |
| $x = 1.029312 \times 10^7$ ft | $xd = -2.248185 \times 10^4$ ft/sec | |
| $y = 1.732354 \times 10^7$ ft | $yd = 9.356206 \times 10^3$ ft/sec | |
| $z = 7.881747 \times 10^6$ ft | $zd = 7.958385 \times 10^3$ ft/sec | |

where $x$, $y$, $z$, $xd$, $yd$, and $zd$ are the position and velocity
components in the earth-centered inertial (ECI) coordinate
frame.  In order to describe subsequent positions and times,
this initial fix (i.e. initial conditions) was chosen as a
starting place for the orbital problem at time equal to zero.
For the IUS, the engine burn times are of fixed duration as
shown in Table II; therefore, the time of burn initiation and

4

## TABLE II

## IUS Characteristics

| Stage | Vehicle Weight (lbs) | Specific Impulse (sec) | Burn Time (sec) | Delta-V (ft/sec) | Propellant Weight (lbs) |
|---|---|---|---|---|---|
| 1,2, &3 | 56119 | 294.332 | 123.91 | 4242.175 | 20263 |
| 2&3 | 33813.6 | 294.189 | 140.35 | 9565.712 | 21505.6 |
| 3 | 8887.5 | 302.492 | 94.49 | 10997.98 | 6016.6 |

the orientation of each burn in inertial space are the variables of interest.

This problem is worked entirely in an ECI frame of reference. Figure 1 describes the orientation of each stage prior to engine initiation. Since the assumption has been made that each burn will be treated as impulsive, this corresponds to orienting the Delta-V vector in inertial space.
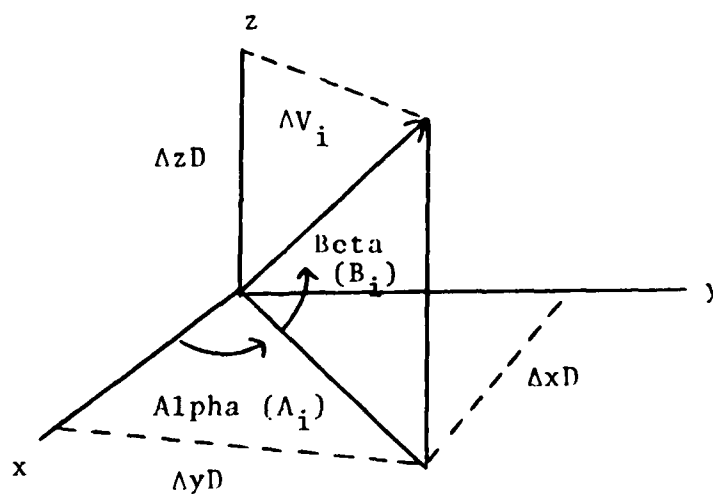


Fig 1. Spherical Coordinates Definition

The angles alpha and beta shown in Fig 1 completely describe
the orientation of the spacecraft and its Delta-V vectors in
inertial space. A third parameter, time, can then be used to
pinpoint the position of the spacecraft along each segment of
the transfer orbit. A pictorial representation of this is as
follows:
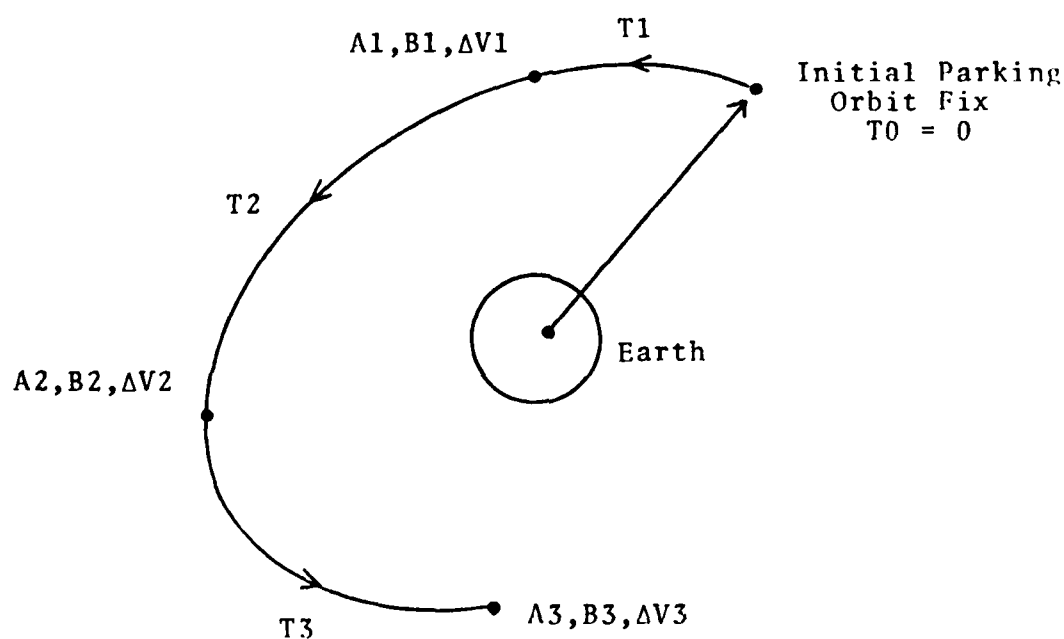


Fig 2. Transfer Orbit

where

T1 = coast time to firing of first stage

A1 & B1 = orientation of first stage at firing

T2 = coast time to firing of second stage

A2 & B2 = orientation of second stage at firing

T3 = coast time to firing of third stage

6

A3 & B3 = orientation of third stage at firing

ΔV1, ΔV2, ΔV3 = impulsive velocity changes per burn

This allows the entire orbit from initial parking orbit to final position to be described by the nine parameters T1, A1, B1, T2, A2, B2, T3, A3, and B3.

As stated before, the calculation of the satellite's position and velocity as it moves through each segment of the transfer orbit will be done using a two-body formulation. Several prediction algorithms are in existance today; all of them solve variations of the differential equation

$$\overset{..}{R} = - \frac{\mu}{R^3} \overline{R} \qquad (1)$$

where $\mu$ is the geocentric gravitational parameter, $\overline{R}$ is the vector position of the satellite, $\overset{..}{R}$ is the vector accelera- of the satellite, and R is the scalar distance from the center of the attracting body to the satellite. Two such algorithms were used in this study. The first is an actual series solu- tion of the two-body differential equations known as the f and g series solution to the Kepler problem (Ref 3). The f and g series solution affords a rapid means of determining position and velocity as a function of time and just as importantly, avoids the penalties associated with the numeri- cal integration of differential equations. A second prediction algorithm, known as Hamilton's two body equations (Ref 2), is composed of six first-order differential equations and was used only to check the accuracy of the subroutine written to do the prediction problem.

7

By using the f and g series in a subroutine, an algorithm defining satellite position and velocity as a function of all nine parameters can be formed.

### Algorithm

1. Given: initial position, velocity, and time (X and T0 = 0)

2. Coast in parking orbit for time T1 and calculate final position and velocity.

3. Align the first burn in inertial space using A1 and B1 and calculate its component velocity contribution. Add this to the velocity from (2) to obtain a new state vector.

4. Coast in new orbit for time T2 and calculate a final state vector. Repeat the procedures used in (3) for the second stage and find the new orbit parameters.

5. Coast in final orbit for time T3, add the vectoral velocity contribution of the third stage, and calculate the final position and velocity.

The requirement that the satellite's final orbit be geosynchronous yields five constraints on the final state vector as defined in the ECI frame.

1. The z component of position must equal zero.

2. The zd component of velocity must equal zero.

3. The satellite's speed must equal geosynchronous speed.

4. The satellite's distance must equal geosynchronous

8

distance.

    5.   The final orbit must be circular.

The problem then is to find values for the parameters T1, A1, B1, T2, A2, B2, T3, A3, and B3 that allow the satellite to transfer from its initial position in the parking orbit to a final orbit which satisfies the specified end conditions in minimum time. It is readily apparent that this problem is not amenable to closed form analytic treatment, which implies the need to solve it iteratively.

# III.   The Optimization Problem

A good definition of a parameter optimization problem
is how to change parameters in order to satisfy end conditions
with the least effort.   Optimization theory provides the dir-
ection needed to change each parameter to reduce end constraint
errors and also offers figures of merit that can be used to
tell how well the procedure is progressing.   This problem, as
stated, involves finding the values of nine parameters that
force the final orbit to satisfy five equality constraints
and allow the satellite to complete the orbital transfer in
minimum time.   This defines the performance index as

$$G = T1 + T2 + T3 \qquad (2)$$

which is the total time of flight.   The initial conditions
for the problem are simply the initial position and velocity
components supplied for the parking orbit for reference time
equal to zero.   The final conditions or end constraints are
labeled by the vector M, where

$$M(1) = zd$$
$$M(2) = z$$
$$M(3) = (xd^2 + yd^2 + zd^2)^{1/2} - 1.0096 \times 10^4 \qquad (3)$$
$$M(4) = (x^2 + y^2 + z^2)^{1/2} - 1.3811 \times 10^8$$
$$M(5) = x(xd) + y(yd) + z(zd)$$

where $1.3811 \times 10^8$ feet is geosynchronous orbital distance,

10

and $1.0096 \times 10^4$ feet per second is synchronous speed. Each constraint is written so that when its numerical value equals zero, the desired condition has been obtained. By calculating the norm of the M vector, a figure of merit is produced which indicates the degree to which the end constraints are satisfied. The optimization problem is then to drive the norm of the M vector to zero and the performance index to its minimum value.

One procedure for accomplishing this is referred to as suboptimal control. In an optimal control problem, the desired controls, which are functions of time, are directly calculated. For this problem that would mean the alpha and beta parameters are functions of time. The assumption that all three burns are impulsive converts the alpha and beta controls to scalar parameters. The suboptimal control technique approximates these controls using polynomials.

In this problem the controls are the three pairs of angles needed to align the stages for firing and the times of firing. If all the parameters are expressed in one vector A, then

$$A = [T1 \ A1 \ B1 \ T2 \ A2 \ B2 \ T3 \ A3 \ B3]^T \qquad (4)$$

This vector A then contains all the information needed to find the final position and velocity of the satellite.

$$X_f = X_f(A) \qquad (5)$$

In addition, the vector A also supplies all the information needed to evaluate the performance index and the end condition

11

constraints.

$$G = G(\Lambda)$$
$$M = M(\Lambda) \tag{6}$$

If an augmented performance index, F, is defined functionally as

$$F(\Lambda,\nu) = G(\Lambda) + \nu^T M(\Lambda) \tag{7}$$

where $\nu$ represents a vector of Lagrange multipliers, it must satisfy the first variational requirements

$$F_\Lambda^T(\Lambda,\nu) = 0$$

$$F_\Lambda = \frac{\partial F}{\partial \Lambda} \tag{8}$$

$$F_\nu^T(\Lambda,\nu) = M(\Lambda) = 0$$

to be a minimum solution. $F_\Lambda$ is the partial of the augmented performance index with respect to the parameter vector $\Lambda$. The $F_\Lambda$ matrix represents the change in performance resulting from a change in each parameter in $\Lambda$(gradient). Since there are nine parameters in $\Lambda$, $F_\Lambda$ is a row vector of nine elements. Hull and Edgeman (Ref 6) describe a second-order parameter optimization technique and algorithm specifically for application to suboptimal control problems. Johnson (Ref 7) and Peterson (Ref 8) used this algorithm to develop a computer program to analyze aircraft time to turn problems. Their program uses three controls, differential equations of motion, and two end condition constraints. The program was modified to accommodate nine parameter controls, a series solution to differential equations of motion, and five end condition

12

constraints.  In short, the program which adopts Hull and Edgeman's algorithm uses second-order information to determine how to change the parameters in the vector $\Lambda$ and the Lagrange multiplier vector, $\nu$, such that the vectors $F_\Lambda$ and M are driven to zero.  The vector relationships used to change $\Lambda$ and $\nu$ are derived from Eq (8).  For any guessed values of $\Lambda$ and $\nu$ that are not a solution, the equality will not hold.

$$F_\Lambda^{\ T} \neq 0 \qquad\qquad M(\Lambda) \neq 0 \qquad\qquad (9)$$

If Eq (9) is then linearized about $\Lambda$ and $\nu$, then

$$\delta F_\Lambda^{\ T} = F_{\Lambda\Lambda}\delta\Lambda + M_\Lambda^{\ T}\ \delta\nu \qquad\qquad (10)$$

$$\delta M = M_\Lambda\delta\Lambda \qquad\qquad (11)$$

and define

$$\delta F_\Lambda^{\ T} = -\ PF_\Lambda^{\ T} \qquad\qquad (12)$$

$$\delta M = -QM \qquad\qquad (13)$$

where P and Q are scaling factors, yields

$$F_{\Lambda\Lambda}\delta\Lambda + M_\Lambda^{\ T}\delta\nu = -PF_\Lambda^{\ T} \qquad\qquad (14)$$

$$M_\Lambda\delta\Lambda = -QM \qquad\qquad (15)$$

which can be solved for $\delta\nu$ and $\delta\Lambda$.

$$\delta\nu = (M_\Lambda F_{\Lambda\Lambda}^{\ -1} M_\Lambda^{\ T})^{-1}(-PM_\Lambda F_{\Lambda\Lambda}^{\ -1} F_\Lambda^{\ T} + QM) \qquad\qquad (16)$$

$$\delta\Lambda = -F_{\Lambda\Lambda}^{\ -1}(PF_\Lambda^{\ T} + M_\Lambda^{\ T}\ \delta\nu) \qquad\qquad (17)$$

where

$$M_\Lambda = \frac{\partial M}{\partial\Lambda} \qquad\text{and}\qquad F_{\Lambda\Lambda} = \frac{\partial^2 F}{\partial^2\Lambda}$$

13

and P and Q are scaling factors which control optimization and end condition satisfaction. $F_{\Lambda\Lambda}$ represents a change in slope and the direction in which it is increasing or decreasing. The algorithm developed to iteratively change $\Lambda$ and $\nu$ follows:

1. Guess $\Lambda$ and $\nu$
2. Use the f and g series solution to find $X_f$
3. Compute M, $M_\Lambda$, $M_{\Lambda\Lambda}$, $F_\Lambda$, and $F_{\Lambda\Lambda}$
4. Pick values for P and Q and compute $\delta\nu$ and $\delta\Lambda$
5. Set $\Lambda = \Lambda + \delta\Lambda$ and $\nu = \nu + \delta\nu$
6. Check convergence and if unsatisfied, go to step (2)

The computer program uses this algorithm plus a gradient approach that allows the direct calculation of an initial $\nu$ vector which eliminates guessing five parameters. The procedure for doing this was also developed by Hull and Edgeman and uses first-order information as follows:

$$\nu = (M_\Lambda M_\Lambda{}^T)^{-1}[(Q/P)M - M_\Lambda G_\Lambda{}^T] \qquad (18)$$

$$\delta_\Lambda = -PF_\Lambda{}^T \qquad (19)$$

where

$$G_\Lambda = \frac{\partial G}{\partial \Lambda}$$

For a typical problem, the gradient portion of the program would be used to locate the vicinity of the functional minimum. At this point where gradient methods lose their effectiveness, the second-order algorithm would be used to

14

rapidly converge the problem. As this procedure is followed, the scaling factors P and Q are gradually raised from very small values to final values of one. In the final convergence cycles, P and Q must be equal to one to yield an optimal solution. Although this procedure would appear straightforward, its application to specific problems can, at times, require some finesse as will be shown later.

## Numerical Methods
(Ref 6:484)

The first- and second-order optimization routines require the matrices M, $M_A$, $M_{AA}$, and $F_{AA}$. These matrices were evaluated using numerical techniques. $F_A$ and $F_{AA}$ are defined as

$$F_A = G_A + v^T M_A \tag{20}$$

$$F_{AA} = G_{AA} + \sum_{i=1}^{5} v_i M_{AA_i} \tag{21}$$

since

$$G = T1 + T2 + T3 \tag{22}$$

$$G_A = [1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0] \tag{23}$$

and

$$G_{AA} = [0]_{9 \times 9} \tag{24}$$

The only quantities that need to be calculated are M, $M_A$, and $M_{AA}$.

The M matrix, or error matrix, is easily evaluated after determining the final position and velocity vectors using the f and g series solution to the Kepler problem. The $M_A$ and $M_{AA}$ matrices are determined using a central

15

differences numerical derivative technique. The technique uses the initial values of the $\Lambda$ vector parameters $(\Lambda_n)$ to calculate an initial M. Each $\Lambda_n$ is then positively perturbed by

$$A_{n+} = \Lambda_n + \delta_n \tag{25}$$

and then negatively by

$$A_{n-} = A_n - \delta_n \tag{26}$$

Using $\Lambda_{n+}$ and $A_{n-}$, $M_+$ and $M_-$ are calculated. The central differences representation for $M_{\Lambda_n}$ is then given by

$$M_{\Lambda_n} = \frac{M_+ - M_-}{2\delta_n} + \sigma(\delta_n^2) \tag{27}$$

where $\sigma(\delta_n^2)$ represents an error term of order $\delta_n^2$ where $\delta_n$ is a very small positive number. The $M_\Lambda$ matrix contains five rows. The first row is determined using $M_1$ in Eq (27), the second row using $M_2$, and so on. The $M_{\Lambda\Lambda}$ matrices are determined in a similar manner; however, two elements $\Lambda_n$ and $\Lambda_m$, must be perturbed both positively and negatively to obtain $M_{++}$, $M_{+-}$, $M_{-+}$, and $M_{--}$. The central differences representation for $M_{\Lambda_n\Lambda_m}$ is

$$M_{\Lambda_n\Lambda_m} = \frac{M_+ - 2M + M_-}{\delta_n^2} + \sigma(\delta_n^2) \tag{28}$$

for $n = m$ and

$$M_{\Lambda_n\Lambda_m} = \frac{M_{++} - M_{+-} - M_{-+} + M_{--}}{4\delta_n\delta_m} + \sigma(\delta_n\delta_m)$$

for $n \neq m$. The $M_{\Lambda\Lambda}$ matrix is really five matrices. One is determined using $M_1$ values in the above equations, and then

the others by $M_2$ through $M_5$ in turn. The error terms in these equations can be ignored if $\delta_n$ and $\delta_m$ are quite small. The $\delta$ used for the central differences technique is

$$\delta_n = \text{DELTA} \cdot A_n \qquad (29)$$

and if the absolute value of $\delta_n$ is larger than DELTA, then

$$\delta_n = \text{DELTA} \qquad (30)$$

where DELTA is another small positive number. In this problem, DELTA was initially chosen to equal $1.0 \times 10^{-6}$.

## Selecting Initial Parameter Values

Picking nine parameters to serve as an initial guess for the orbital transfer problem is far from a simple matter. Not only must the parameters be compatible, but they should also provide an initial "miss" relative to the end condition constraints small enough to allow for easy convergence. At this point, some engineering insight was applied with only marginal success. Since the actual parking orbit has an eccentricity of 0.01497, one would think parameters good for a perfectly circular orbit would be good initial guesses for the actual. With this in mind, a test orbit was constructed in an ECI frame with ascending node on the X-axis at $2.16737 \times 10^7$ feet, inclined 22.8 degrees at a velocity of 25,618 feet per second. The first parameter, T1, was determined by calculating the time of flight to the descending node. Since the final orbit would lie in the X-Y plane, the first burn

17

was aligned to eliminate as much inclination as possible using

$$\Delta V_i = 2V \sin \theta/2 \qquad (31)$$

where $\Delta V_i$ is the change in velocity due to stage burning, V is the original velocity of the spacecraft at nodal crossing, and $\theta$ represents the angular change in the inclination. This allowed A1 and B1 to be determined. T2 was chosen to be the time of flight to the next nodal crossing. The second burn was aligned in an attempt to eliminate the remaining inclination yielding A2 and B2. T3 was again chosen to be the time to the next nodal crossing, and A3 and B3 were varied to find the smallest error in the end conditions. This orbit and the actual orbit differ only by a rotation about the Z-axis to align the nodes--ignoring eccentricity. This angle was then added to the three alpha parameters, and T1 was adjusted to reflect the coast time to first nodal crossing from the initial fix. The initial miss vector that this procedure yielded was of magnitude $10^{12}$, largely due to the circularity constraint. Random adjustment of individual parameters proved to be a hopeless means of reducing the error. The circularity constraint proved to be far too sensitive to allow for rapid convergence. To decrease the sensitivity of this constraint, the circularity was multiplied by a scaling factor of $10^{-5}$. This allowed the program to drive down the errors in the other four constraints. Each time this occurred, the scaling factor was increased by a factor of ten and the process repeated. This allowed the scaling factor to eventually

18

be removed once the vicinity of the functional minimum was reached.

# IV.  Results

## Choosing P, Q, and DELTA

The selection of the P and Q scaling factors is unique
to every optimization problem.  By their selection, one can
select several options.  The first and most commonly used is
to choose P very small and Q large as compared with P.  By
doing this, the program will attempt to satisfy the end con-
straints before doing any optimization.  A second method is
to choose P larger than Q in an attempt to satisfy the end
conditions after finding an optimal region.  The third common
method is to choose P and Q equal and, thereby, optimize
while trying to satisfy the end constraints.  All three
methods were tried on this problem, with only the first being
successful.  It was found necessary to first satisfy the end
conditions and then search for an optimal solution.

Second-order optimization methods are very unstable
far from the optimal solution.  This requires very small
changes in the control parameters through each iteration to
prevent divergence.  The magnitudes of these "delta" quanti-
ties can be controlled using P and Q.  Trial and error
showed that for P equal to $1.0 \times 10^{-15}$ and Q equal to 1.0
$\times 10^{-6}$, sufficiently small parameter changes were generated
to allow the program to run for an average of 25 iterations
before divergence occurred.  This was caused by the numerical

routine generating an excessively large change in one of the
parameters.  The program was modified to allow continuous
operation by examining the norm of these parameter changes.
Initially, whenever the norm exceeded 0.38, the divergence
was seen to occur.  The program was then modified to reiniti-
alize all the Lagrange multipliers and restart itself whenever
this was observed.  As the program began to approach the
vicinity of the functional minimum, the numerical routine
became more stable.  The limit on the norm of the parameter
changes was relaxed to 0.5 and finally to 1.0.  This simple
procedure proved extremely valuable after unsuccessfully
attempting to use a straight gradient method.

A third parameter, DELTA, also showed a large impact
on the success of the routine.  For a similar problem using
differential equations of motion, DELTA would be chosen in
the $10^{-4}$ to $10^{-6}$ range.  This insures the accurate calcula-
tion of the $M_\Lambda$ and $M_{\Lambda\Lambda}$ matrices.  In addition, the size of
the parameter changes (da's) is directly proportional to
DELTA, since the "delta" perturbing values equal DELTA times
each element of $\Lambda$.  In this problem, DELTA equal to $10^{-6}$ was
used based on this previous experience.  It was found that
the numerical routine stagnated when the solution was being
approached.  It appears the function contours were steep to
the point where the parameter changes equal to $10^{-6}$ times
their present value would not allow the program to move deeper
into the contour valleys.  At this point, several iterations
were run with the same initial parameter values, but with

21

different DELTAs. By comparing the norms of the constraint errors, it was found that performance improved as DELTA was progressively decreased. Best performance resulted from a DELTA of $10^{-10}$, which was then used to find the solution presented here.

## Gradient Method

The gradient method, which uses first-order information in an attempt to move toward an optimal solution, proved to be a dismal failure. Because the performance index was defined as the sum of three times, the gradient method tried to drive all three to zero. Gradient information was used, however, to calculate initial values of Lagrange multipliers by Eq (18). These were then used in the second-order method.

## Numerical Results

The second-order method used in the computer program, when modified to run continuously, steadily drove toward a solution that satisfied the end constraints and then toward an optimal solution. The parameters that satisfied the end constraints changed very little during the optimization portion of the program, which suggests that this may be a locally unique solution. The optimization iteration process was stopped with P and Q equal to one when the norm of the parameter changes was less than $10^{-9}$ and the norm of the Lagrange multiplier changes was also very small. The $\Lambda$ matrix solution and the sensitivity matrix, $M_A$, are shown in the following tables.

22

## TABLE III

## Parameter Solution Set

| | |
|---|---|
| T1 = 2030.2449995 sec | T2 = 3847.461750268 sec |
| A1 = -0.435361007 rad | A2 = 0.732341983313 rad |
| B1 = 0.7111815869 rad | B1 = -2.731217478098 rad |

$$T3 = 7470.83934000 \text{ sec}$$

$$A3 = 1.30013074628 \text{ rad}$$

$$B3 = 0.02236409011 \text{ rad}$$

## TABLE IV

## End Constraint Errors

$M(1) = zd = -2.5465 \times 10^{-10}$ ft/sec

$M(2) = z = 9.5367 \times 10^{-7}$ ft

$M(3) =$ synchronous speed $= -1.0477 \times 10^{-9}$ ft/sec

$M(4) =$ synchronous distance $= 8.583 \times 10^{-6}$ ft

$M(5) =$ circularity (R V) $= 1.318 \times 10^{-2}$

Total Transfer Time = 13348.54608 sec =
3.70792 hrs

Figures 3 through 8 depict the transfer orbit from
several points of view. Figure 5 shows a result that is some-
what surprising. The transfer orbit remains in the same plane
as the parking orbit. In other words, the transfer opts to
increase the altitude and velocity of the spacecraft early
in the transfer and complete the inclination change in the
final burn. As Eq (31) points out, the cost to change

23

TABLE V

Position and Velocity After Firing of Each Stage

Position and Velocity After Firing First Stage:

$x = -2.0373124 \times 10^7$ ft    $xd = 1.11011967 \times 10^4$ ft/sec

$y = -7.6147903 \times 10^6$ ft    $yd = -2.21389677 \times 10^4$ ft/sec

$z = -1.3724214 \times 10^6$ ft    $zd = -9.3978524 \times 10^3$ ft/sec

Position and Velocity After Firing Second Stage:

$x = -2.885451299 \times 10^6$ ft    $xd = -2.05530541 \times 10^4$ ft/sec

$y = 3.20133328 \times 10^7$ ft    $yd = -1.176502728 \times 10^4$ ft/sec

$z = 1.257707996 \times 10^7$ ft    $zd = -5.014671267 \times 10^3$ ft/sec

Final Position and Velocity after Third Burn:

$x = 1.379638 \times 10^8$ ft    $xd = 464.30038$ ft/sec

$y = -6.351478 \times 10^6$ ft    $yd = 10085.31809$ ft/sec

$z = -8.118152 \times 10^{-5}$ ft    $zd = 2.8067 \times 10^{-9}$ ft/sec

inclination is less as the spacecraft velocity decreases.
The point at which geosynchronous orbit is obtained is where
the spacecraft velocity is lowest. This points out that the
assumption used to obtain initial values for the parameters
by eliminating the inclination of the transfer orbit was
wrong. Figure 6 depicts the entire transfer orbit in three
dimensions. Figure 7 is the same as Fig 6, except that the
scaling on the Z-axis has been changed to show separation of
the near earth portions. Figure 8 is the confirmation that
the final orbit is, indeed, circular and geosynchronous.
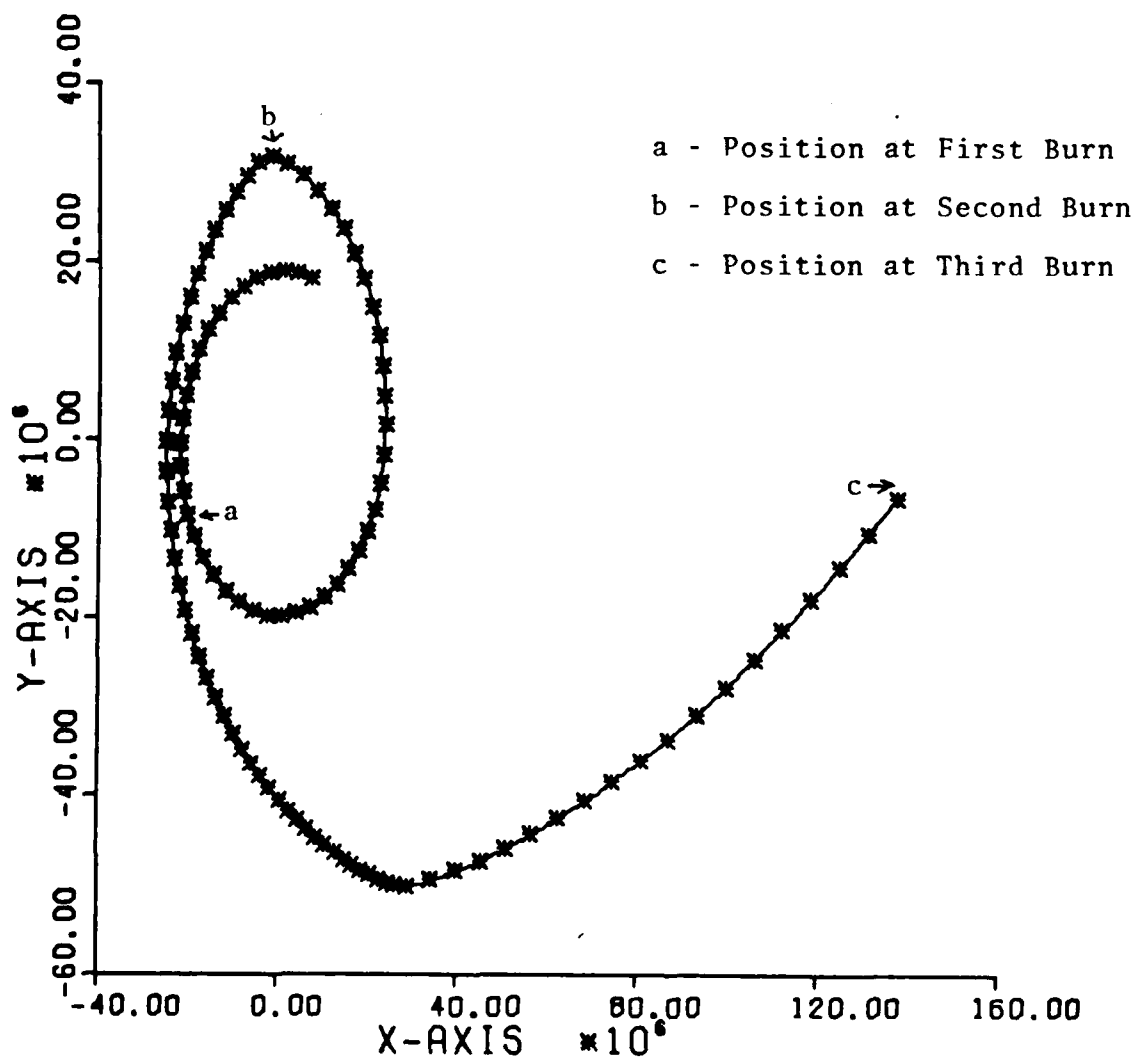
Fig 3.   Depiction of Transfer Orbit in ECI X-Y Plane

25
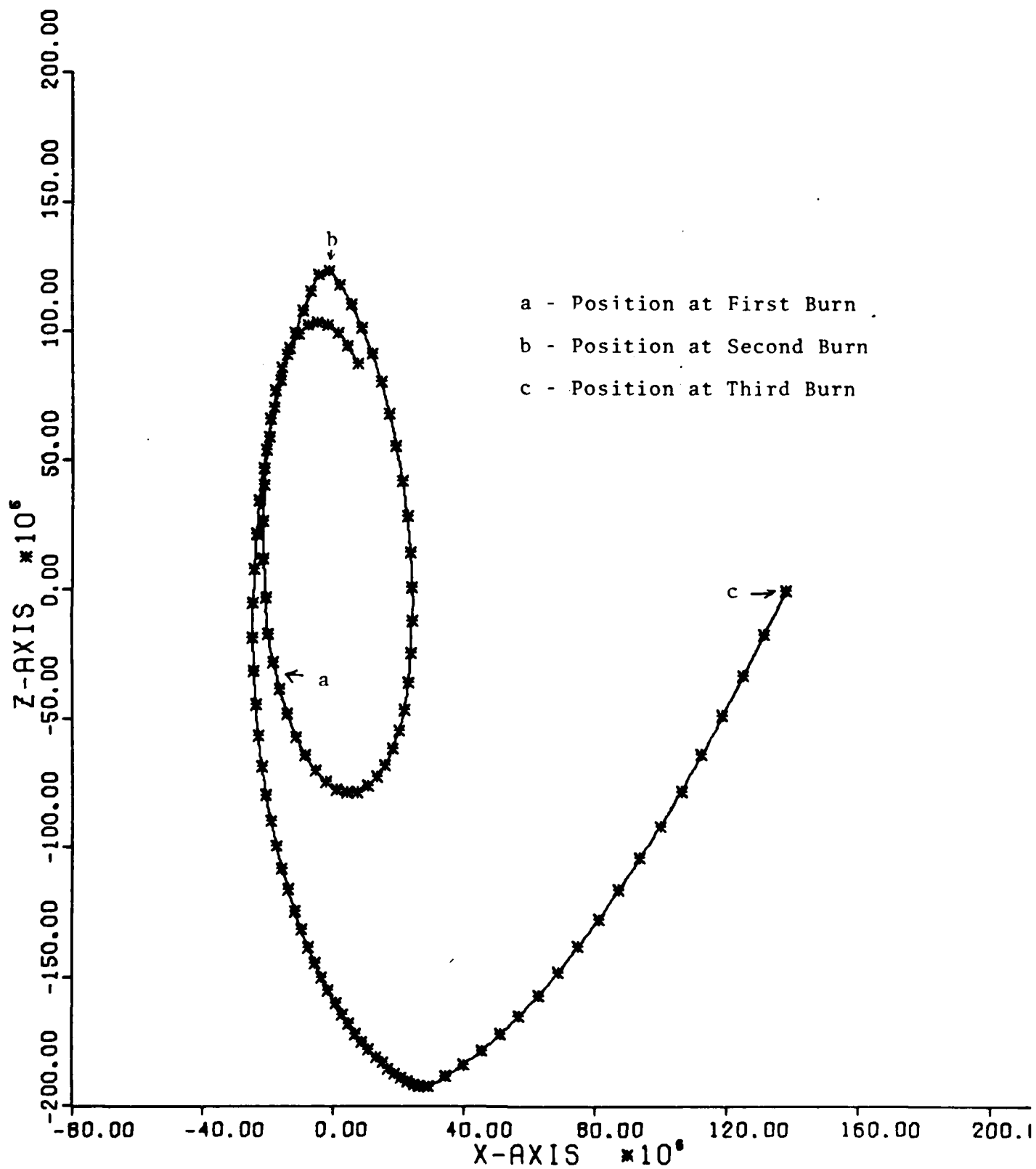
Fig 4. Depiction of Transfer Orbit in ECI X-Z Plane

26

a - Position at First Burn

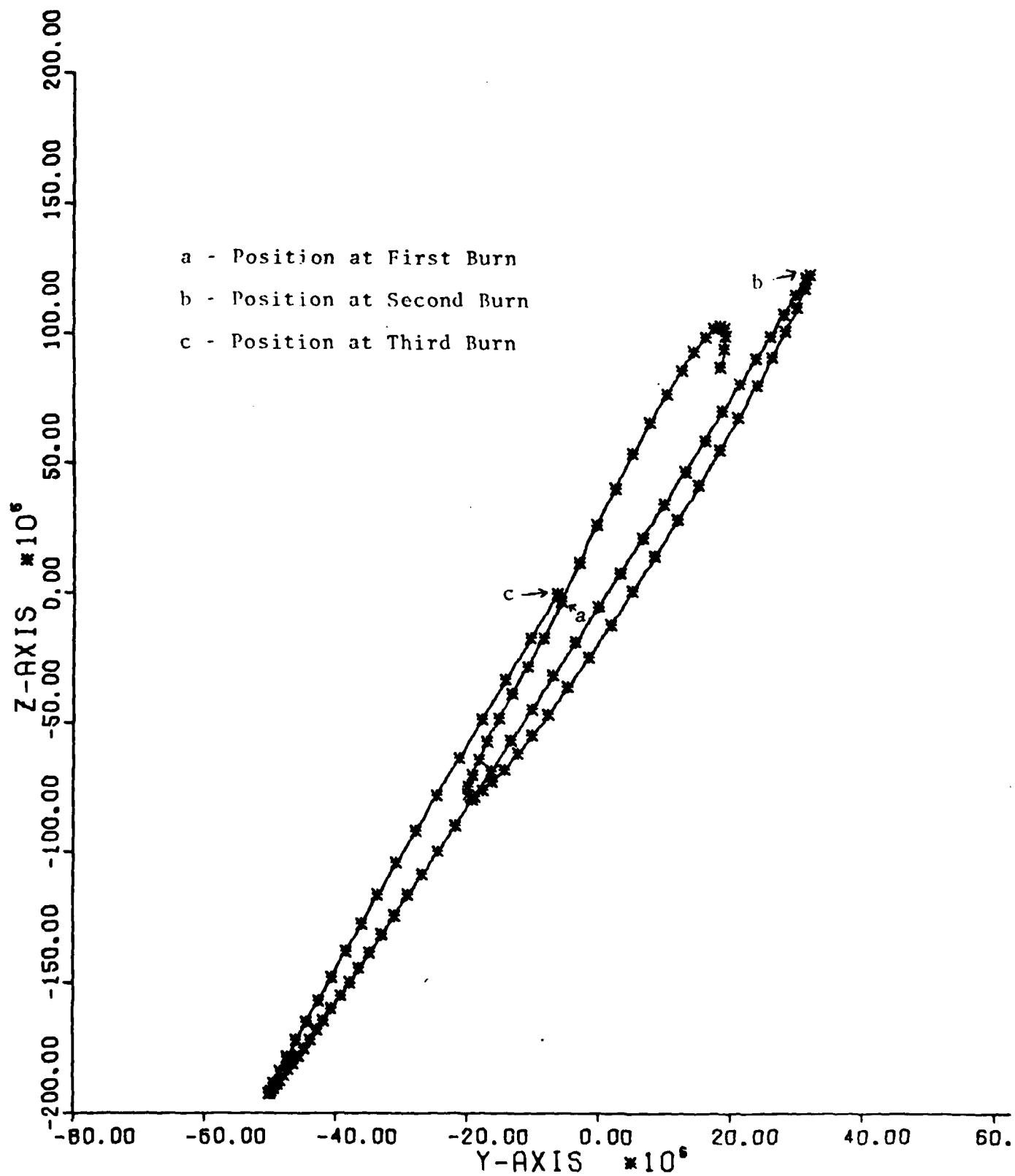b - Position at Second Burn

c - Position at Third Burn

Fig 5. Depiction of Transfer Orbit in ECI Y-Z Plane

27

# 3-D VIEW OF TRANSFER ORBIT
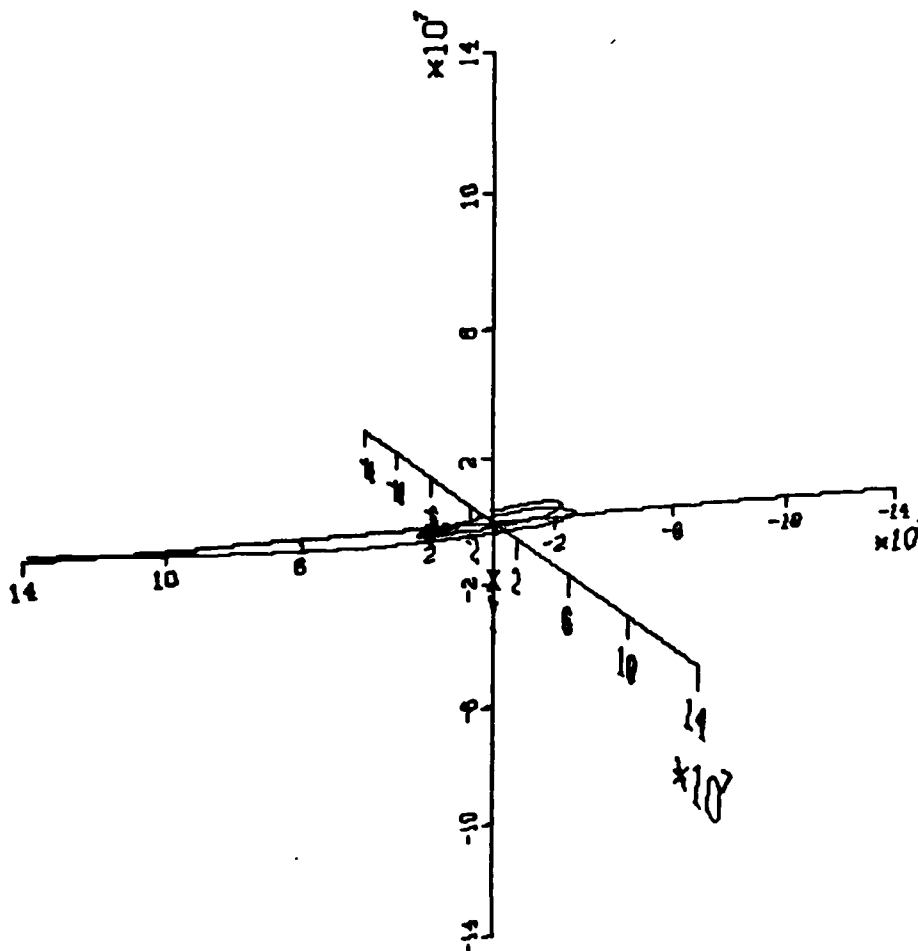
Fig 6.   Depiction of Transfer Orbit in 3-D

# 3-D VIEW OF TRANSFER ORBIT
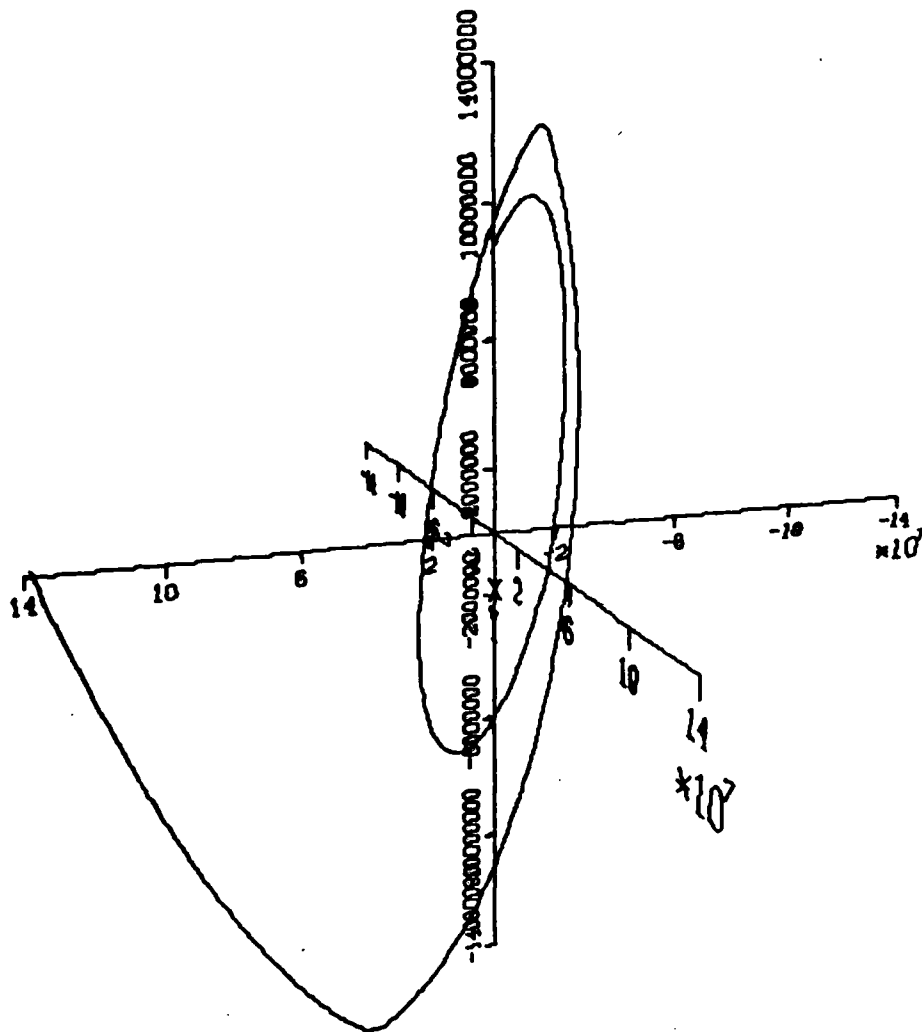


Fig 7.   Depiction of Transfer Orbit in 3-D
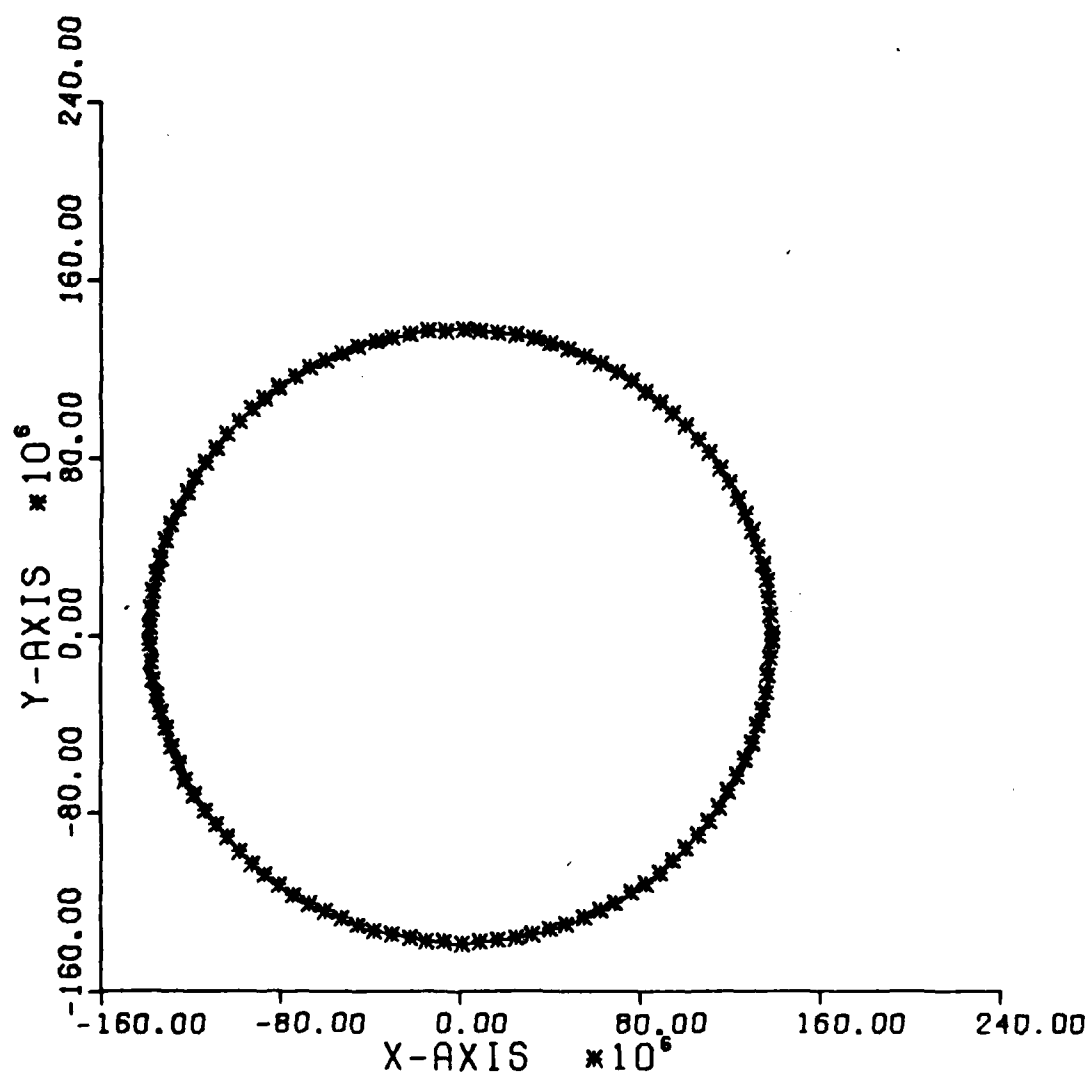
Fig 8.   Depiction of Final Orbit in ECI X-Y Plane

30

The $M_\Lambda$ matrix which is the partial of M with respect to the $\Lambda$ matrix indicates the sensitivity of each end condition to a change in any one of the $\Lambda$ matrix parameters.

TABLE VI

$M_\Lambda$ Matrix

|    | T1 | A1 | B1 |
|----|------|------|------|
| M1 | 9.9414 | -11045.0 | -17707.0 |
| M2 | -78377.0 | 83614230.0 | 147191047.0 |
| M3 | 26.306 | -28542.0 | -45384.0 |
| M4 | -497674.0 | 473346710.0 | 756855918.0 |
| M5 | 9.9097 | $-7.73 \times 10^{12}$ | $-1.215 \times 10^{13}$ |
|    | T2 | A2 | B2 |
| M1 | -10.353 | 3133.023 | 1377.528 |
| M2 | -248.21 | 4481256.0 | 60289542.0 |
| M3 | -26.293 | 7204.65 | 6391.623 |
| M4 | 304516.131 | -103134155.0 | -82096056.0 |
| M5 | $1.596 \times 10^9$ | $1.3178 \times 10^{12}$ | $3.077 \times 10^{11}$ |
|    | T3 | A3 | B3 |
| M1 | -0.802865 | 0 | 10995.048 |
| M2 | 14068.347 | 0 | 0 |
| M3 | -2.0329 | 2449.40 | -239.582 |
| M4 | 50201.67 | 0 | 0 |
| M5 | -646330649.0 | $-1.48 \times 10^{12}$ | $-7.567 \times 10^9$ |

The $M_\Lambda$ matrix is extremely useful when one is interested in how changes in one of the parameters in the $\Lambda$ matrix will effect the end conditions. As an example, consider the last column, which represents the change in end conditions due to

TABLE VII

$M_A$ Values for B3

|     | B3 |
| --- | --- |
| M1 | 10995.048 |
| M2 | 0 |
| M3 | -239.582 |
| M4 | 0 |
| M5 | $-7.567 \times 10^9$ |

a change in B3.  These numbers indicate that if the third
stage was misaligned by plus one radian, it would increase
the zd component of velocity by 10995.048 feet per second.
Since B3 has no effect on position, the numbers that corres-
pond to z and geosynchronous distance are both zero.  The
satellite's speed would decrease by 240 feet per second,
and the circularity of the orbit by $7 \times 10^9$.  This clearly
illustrates the use of the $M_A$ matrix as well as the wide
differences in magnitudes of the individual errors.

## V.  Application

The application of the information found in this study is both simple and straightforward.  The parking orbit and the transfer orbit are fixed in inertial space due to the choice of an ECI reference frame.  The earth, on the other hand, spins relative to this frame.  To place a satellite in geosynchronous orbit over a specific spot on the equator, one has basically two choices.  The first is to time the launch of the satellite from the earth into its parking orbit and then along the transfer orbit such that the final position of the transfer orbit coincides with the desired equatorial position.  A second and more practical approach would be to launch into the parking orbit whenever convenient. The period of the parking orbit is roughly 90 minutes, which allows for a transfer every hour and a half.

Because this problem was worked in an ECI frame of reference, the position of the satellite at the end of the transfer relative to its position at the beginning is always constant.  Therefore, the angular difference in their positions, as seen in the X-Y plane and measured about the Z-axis, is constant.  By allowing for the rotation of the earth, a transfer to a given position over the equator can be calculated in terms of longitude.  Optimal launch conditions occur when the longitude of the parking orbit initial fix is

114.90 degrees greater than the longitude of the desired

geosynchronous orbital position. For example, if the initial

orbit was obtained at longitude 23.90°E and the transfer begun,

the final position would be at longitude 91.0°W, which corres-

ponds to the Galapagos Island chain, after a flight time of

3.70792 hours.

# VI. Recommendations

One of the major assumptions made in this study was that the three engine burns were impulsive. As was shown, this reduced the alpha and beta controls to scalar parameters. In reality, the engines operate for specific lengths of time as shown in Table II. Since the computer program has the capability to determine the alpha and beta controls as polynomial functions of time, it would be interesting to remove the impulsive burn assumption and rework this problem using the solution found here as a good initial guess.

This study found only one solution to this transfer problem and suggests that it may be unique. If the original parking orbit were equatorial, there would exist two mirror image solutions. It would be interesting to see if the methods used in this study would yield both solutions.

The suboptimal control technique, as applied to this problem, did yield a valid optimal solution, but the method did prove expensive in terms of computer time. It would be worthwhile to apply one or two other second order or quasi-second order optimization methods to this problem to see if a more efficient method exists.

# Bibliography

1. McAlister, Captain C., USAF. "Space Shuttle Parking Orbit Data and Inertial Upper Stage Performance Data." 6555 ASTG/LVU, Patrick AFB, Florida, 10 March 1980.

2. Bate, Mueller, and White. Fundamentals of Astrodynamics. New York: Dover Publications, Inc., 1971.

3. Weisel, Captain W., USAF. Lecture materials distributed in MC 7.20, Analytical Dynamics, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1980.

4. Rader, Captain J., USAF. Lecture materials distributed in MC 5.24, Optimization Techniques I, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1979.

5. _____. Lecture materials distributed in MC 6.24, Optimization Techniques II, Wright-Patterson AFB, Ohio, 1980.

6. Hull and Edgeman. "Suboptimal Control Using a Second Order Parameter Optimization Method," Journal of Optimization Theory and Application, 17: 482-491 (December 1975).

7. Johnson, Captain T., USAF. "Minimum Time Turns With Thrust Reversal." Unpublished MS Thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1979.

8. Peterson, Captain S., USAF. "Time Constrained Maximum Energy Turns." Unpublished MS Thesis, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1979.

9. Marchal, Christian. "Optimal Transfers Between Elliptical Orbits." Communication presented to the 16th International Congress of Astronautics, Athens, 12-18 September 1965.

10. Engineering Science Operations. "Optimal Three Burn Orbit Transfer." The Aerospace Corporation, El Segundo, California.

APPENDIX A

Computer Program SubOpt

```
100=        PROGRAM ORBIT(INPUT,OUTPUT)
110=        DIMENSION DA(9),MAA3(9,9),MAA4(9,9),MAA5(9,9),WZ(9)
120=        DIMENSION A(9),AA(9),GA(9)
130=        DIMENSION M(5),MA(5,9),MAA1(9,9),MAA2(9,9)
140=        DIMENSION B(6),DB(6),BB(6),BP(6),BP1(6),BP2(6),BP3(6),BP4(6)
150=        DIMENSION DEL(9),MAMAT(5,5),MAMATI(5,5),MAGAT(5)
160=        DIMENSION FAT(9),FAA(9,9),FAAI(9,9),MAFAAI(5,9)
170=        DIMENSION W(5,5),WI(5,5),C(5),D(9),PNU(5),DPNU(5),PNUMA(9)
180=        DIMENSION E(9),FA(9),FFAA(9,9),FFAAI(9,9)
190=        DIMENSION GAA(9,9)
200=        DIMENSION MTXFXT(12,12),FFAAB(22,22)
210=        DIMENSION MTX(12,12),MTXT(12,12),CONA(12),CONB(12),MTXF(12,12)
220=        COMMON/MISC/NPH,NPI,NA,TF,DT,N,S,LL,PP,TOT(8),FIXT
230=        COMMON X,Y,Z,XD,YD,ZD
240=        COMMON/QUES/ALPHA
250=        REAL M,M1,M2,M3,M4,MM1,MM2,MM3,MM4,MM5,MM6,MM7,MM8,MA
260=        REAL M5,M6,M7,M8,M9,M10
270=        REAL MM9,MM10,MM11,MM12,MM13,MM14,MM15
280=        REAL MM16,MM17,MM18,MM19,MM20,MAA1,MAA2,MAA3,MAA4,MAA5
290=        REAL MAMAT,MAMATI,MAGAT,MAFAAI,NORMA,NORMM,NORMP
300=C
310=C  PROGRAM SUBOPT IS A SUBOPTIMAL CONTROL TECHNIQUE USED TO FIND AN
320=C  APPROXIMATE SOLUTION TO AN OPTIMAL CONTROL PROBLEM.   THE SOLUTION TO
330=C  MOST OPTIMAL CONTROL PROBLEMS IF THE CONTROLS CANNOT BE SOLVED
340=C  ANALYTICALLY IS TO GUESS THE CONTROLS AND THE LAGRANGE MULTIPLIERS
350=C  AND SEE IF END CONDITIONS ARE MET AND THE PERFORMANCE INDEX MINIMIZED
360=C  THE SUBOPTIMAL CONTROL TECHNIQUE ASSUMES THE CONTROLS ARE A LINEAR
370=C  COMBINATION OF POLYNOMIALS WITH UNKNOWN COEFFICIENTS.   IN SO DOING
380=C  THE PROBLEM IS CHANGED FROM A FUNCTIONAL MINIMIZATION PROBLEM TO A
390=C  PARAMETER OPTIMIZATION PROBLEM.
400=C     IN THIS EXAMPLE THE CONTROLS ARE NOT FUNCTIONS OF TIME.
410=C     THIS SIMPLIFIES THE PROGRAM TO ONE OF PARAMETER OPTIMIZATION.
420=C
430=C     THIS VERSION ADOPTS THE ORBITAL TRANSFER PROBLEM
440=C
450=C     A   =MATRIX OF PARAMETERS(T1,A1,B1,T2,A2,B2,T3,A3,B3)
460=C     DA  =MATRIX OF PARAMETER CHANGES (NP X 1)
470=C     NE  =NUMBER OF STATE VARIABLES
480=C     NP  =TOTAL NUMBER OF PARAMETERS
490=C     NC  =NUMBER OF END CONDITIONS TO BE SATISFIED
500=C     P  =SCALING FACTOR PERFORMANCE INDEX
510=C     Q  =SCALING FACTOR FOR END CONDITION CONSTRAINTS (M)
520=C  G   = SCALAR PERFORMANCE INDEX
530=C  GA  = PARTIAL DERIVATIVES OF G WRT A'S (1 X NP)
540=C  GAA = SECOND PARTIAL DERIVATIVES OF G WRT A'S (2 - NP X NP)
550=C  M   = MATRIX OF PRESCRIBED FINAL CONDITIONS (NC X 1)
560=C  MA  = PARTIAL DERIVATIVES OF M WRT A'S (NC X NP)
570=C  MAA = SECOND PARTIAL DERIVATIVES OF M WRT A'S (2 - NP X NP)
580=C  B   =STATE VARIABLES (X,Y,Z,XD,YD,ZD)
590=C  F   = AUGMENTED PERFORMANCE INDEX (G + PNUT*M)
600=C  FA  = PARTIAL DERIVATIVE OF F WRT A'S (1 X NP)
```

```
610=C   FAT = PARTIAL DERIVATIVES OF F WRT A'S TRANSPOSED (NP X 1)
620=C   FAA = SECOND PARTIAL DERIVATIVE OF F WRT A'S (NP X NP)
630=C   PNU = LAGRANGE MULTIPLIERS (NC X 1)
640=C   DPNU= CHANGE IN LAGRANGE MULTIPLIERS (NC X 1)
650=C
660=C      THE CIRCULAR PARKING ORBIT IS INCLINED 22.8 DEGREES
670=C      AT AN ALTITUDE OF 2.1637E07 FT
680=C
690=       T=0.0
700=       X=1.029312E07
710=       Y=1.732354E07
720=       Z=7.881747E06
 `0=       XD=-2.248185E04
,40=       YD=9.356206E03
750=       ZD=7.958385E03
760=C
770=C      THE THREE BURNS ARE ORIENTED IN TIME AND SPACE
780=C      BY NINE PARAMETERS
790=C
800=       T1=2030.244999504
810=       A1=-.43536100070055
820=       B1=.71118158869911
830=       T2=3847.461750268
840=       A2=.7323419833131
850=       B2=-2.731217478098
860=       T3=7470.839340005
870=       A3=1.300130746286
880=       B3=.0223640901197
890=C
900=C      THE FINAL ORBIT IS CALCULATED USING THE F & G SERIES
910=C      SOLUTION TO THE TWO BODY PROBLEM.
920=C
930=       R=(X*X+Y*Y+Z*Z)**0.5
940=       V=(XD*XD+YD*YD+ZD*ZD)**0.5
950=       PRINT*,"X(0)=   ",X
960=       PRINT*,"Y(0)=   ",Y
970=       PRINT*,"Z(0)=   ",Z
980=       PRINT*,"XD(0)=  ",XD
990=       PRINT*,"YD(0)=  ",YD
1000=      PRINT*,"ZD(0)=  ",ZD
1010=      PRINT*,"RANGE=  ",R
1020=      PRINT*,"VELOCITY=  ",V
1030=      PRINT*," "
1040=      NP=9
1050=      NC=5
1060=      NE=6
1070=      ITER=0
  90=      MAX=100
.J90=      ZED=1.
1100=      DELTA=1.E-08
1110=      IMET=2
1120=      Q=1.
1130=      P=1.
1140=      FNORMA=10.0
1150=      FGNORM=10.0
1160=      CC1=1.0E-04
1170=      CC2=1.0E-02
1180=      CC3=1.0E-08
1190=      CC4=1.0E-04
1200=      CC5=1.0E-04
```

```
1210=          B(1)=X
1220=          B(2)=Y
1230=          B(3)=Z
1240=          B(4)=XD
1250=          B(5)=YD
1260=          B(6)=ZD
1270=          A(1)=T1
1280=          A(2)=A1
1290=          A(3)=B1
1300=          A(4)=T2
1310=          A(5)=A2
 20=          A(6)=B2
 30=          A(7)=T3
1340=          A(8)=A3
1350=          A(9)=B3
1360=555   ITER=0
1370=          PNU(1)=-.00001
1380=          PNU(2)=8.0E-08
1390=          PNU(3)=-.0000537
1400=          PNU(4)=-1.757E-07
1410=          PNU(5)=-1.7E-11
1420=          DPNU(1)=DPNU(2)=0.0
1430=          DPNU(3)=DPNU(4)=DPNU(5)=0.0
1440=52    FORMAT(1X,5E15.7)
1450=50    FORMAT(1X,8E15.7)
1460=C
1470=C    DETERMINING M MATRIX BY INTEGRATING DIFFERENTIAL CONSTRAINTS
1480=C
1490=          DO 1 I=1,NP
1500=1     DA(I)=0.0
1510=1000  DO 2 I=1,NP
1520=          A(I)=A(I)+DA(I)
1530=2     AA(I)=A(I)
1540=          PRINT*," "
1550=          PRINT*,"ITERATION NUMBER   ",ITER," P= ",P,"    Q= ",Q
1560=          PRINT*," "
1570=          DO 3 I=1,NE
1580=3     BB(I)=B(I)
1590=          S=1.0
1600=          CALL TRANS(AA,BB)
1610=          M(1)=ZD
1620=          M(2)=Z
1630=          M(3)=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
1640=          M(4)=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
1650=          M(5)=X*XD+Y*YD+Z*ZD
1660=          M(5)=M(5)/ZED
 70=          G=AA(1)+AA(4)+AA(7)
 80=          ITER=ITER+1
1690=          PRINT*," "
1700=          PRINT*,"A MATRIX"
1710=          DO 666 I=1,NP
1720=666   PRINT*,"A(",I,")= ",A(I)
1730=          PRINT*," "
1740=          DO 4 I=1,NC
1750=          PNU(I)=PNU(I)+DPNU(I)
1760=4     PRINT*,"M(",I,")= ",M(I)
1770=          NORMM=0.0
1780=          DO 5 I=1,NC
1790=5     NORMM=NORMM+M(I)**2
1800=          NORMM=SQRT(NORMM)          40
```

```
1810=        PRINT*," "
1820=        PRINT*,"NORM OF M = ",NORMM,"  P.I.  = ",G
1830=C
1840=C   DETERMINING MA AND MAA BY CENTRAL DIFFERENCES
1850=C
1860=        S=0.0
1870=        DO 100 I=1,NP
1880=        DO 6 J=1,NP
1890=6       AA(J)=A(J)
1900=        DEL(I)=DELTA*A(I)
1910=        IF(ABS(DEL(I)).LE.DELTA) DEL(I)=DELTA
  20=        AA(I)=A(I)+DEL(I)
1930=        DO 7 J=1,NE
1940=7       BP(J)=B(J)
1950=        CALL TRANS(AA,BP)
1960=        M1=ZD
1970=        M2=Z
1980=        M3=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
1990=        M4=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
2000=        M5=X*XD+Y*YD+Z*ZD
2010=        M5=M5/ZED
2020=        G1=AA(1)+AA(4)+AA(7)
2030=        AA(I)=A(I)-DEL(I)
2040=        DO 8 J=1,NE
2050=8       BP(J)=B(J)
2060=        CALL TRANS(AA,BP)
2070=        M6=ZD
2080=        M7=Z
2090=        M8=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
2100=        M9=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
2110=        M10=X*XD+Y*YD+Z*ZD
2120=        M10=M10/ZED
2130=        G2=AA(1)+AA(4)+AA(7)
2140=        MA(1,I)=(M1-M6)/(2.0*DEL(I))
2150=        MA(2,I)=(M2-M7)/(2.0*DEL(I))
2160=        MA(3,I)=(M3-M8)/(2.0*DEL(I))
2170=        MA(4,I)=(M4-M9)/(2.0*DEL(I))
2180=        MA(5,I)=(M5-M10)/(2.0*DEL(I))
2190=        GA(I)=(G1-G2)/(2.0*DEL(I))
2200=        DO 100 K=1,I
2210=        IF(K.EQ.I)GO TO 707
2220=        IF(IMET.EQ.1)GO TO 100
2230=        DO 9 J=1,NP
2240=9       AA(J)=A(J)
2250=        AA(I)=A(I)+DEL(I)
2260=        DEL(K)=DELTA*A(K)
  70=        IF(ABS(DEL(K)).LE.DELTA) DEL(K)=DELTA
2280=        AA(K)=A(K)+DEL(K)
2290=        DO 10 J=1,NE
2300=10      BP1(J)=B(J)
2310=        CALL TRANS(AA,BP1)
2320=        MM1=ZD
2330=        MM2=Z
2340=        MM3=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
2350=        MM4=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
2360=        MM5=X*XD+Y*YD+Z*ZD
2370=        MM5=MM5/ZED
2380=        GG1=AA(1)+AA(4)+AA(7)
2390=        DO 11 J=1,NP              41
2400=11      AA(J)=A(J)
```

```
2410=        AA(I)=A(I)+DEL(I)
2420=        AA(K)=A(K)-DEL(K)
2430=        DO 12 J=1,NE
2440=12      BP2(J)=B(J)
2450=        CALL TRANS(AA,BP2)
2460=        MM6=ZD
2470=        MM7=Z
2480=        MM8=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
2490=        MM9=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
2500=        MM10=X*XD+Y*YD+Z*ZD
2510=        MM10=MM10/ZED
2520=        GG2=AA(1)+AA(4)+AA(7)
: '0=        DO 13 J=1,NP
2_40=13      AA(J)=A(J)
2550=        AA(I)=A(I)-DEL(I)
2560=        AA(K)=A(K)+DEL(K)
2570=        DO 14 J=1,NE
2580=14      BP3(J)=B(J)
2590=        CALL TRANS(AA,BP3)
2600=        MM11=ZD
2610=        MM12=Z
2620=        MM13=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
2630=        MM14=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
2640=        MM15=X*XD+Y*YD+Z*ZD
2650=        MM15=MM15/ZED
2660=        GG3=AA(1)+AA(4)+AA(7)
2670=        DO 15 J=1,NP
2680=15      AA(J)=A(J)
2690=        AA(I)=A(I)-DEL(I)
2700=        AA(K)=A(K)-DEL(K)
2710=        DO 16 J=1,NE
2720=16      BP4(J)=B(J)
2730=        CALL TRANS(AA,BP4)
2740=        MM16=ZD
2750=        MM17=Z
2760=        MM18=((XD*XD+YD*YD+ZD*ZD)**0.5)-1.0096E04
2770=        MM19=((X*X+Y*Y+Z*Z)**0.5)-1.3811E08
2780=        MM20=X*XD+Y*YD+Z*ZD
2790=        MM20=MM20/ZED
2800=        GG4=AA(1)+AA(4)+AA(7)
2810=        MAA1(I,K)=MAA1(K,I)=(MM1-MM6-MM11+MM16)/(4.0*DEL(I)*DEL(K))
2820=        MAA2(I,K)=MAA2(K,I)=(MM2-MM7-MM12+MM17)/(4.0*DEL(I)*DEL(K))
2830=        MAA3(I,K)=MAA3(K,I)=(MM3-MM8-MM13+MM18)/(4.0*DEL(I)*DEL(K))
2840=        MAA4(I,K)=MAA4(K,I)=(MM4-MM9-MM14+MM19)/(4.0*DEL(I)*DEL(K))
2850=        MAA5(I,K)=MAA5(K,I)=(MM5-MM10-MM15+MM20)/(4.0*DEL(I)*DEL(K))
2860=        GAA(I,K)=GAA(K,I)=(GG1-GG2-GG3+GG4)/(4.0*DEL(I)*DEL(K))
2870=        GO TO 100
 80=707      MAA1(I,I)=(M1-2.0*M(1)+M6)/(DEL(I)**2)
_690=        MAA2(I,I)=(M2-2.0*M(2)+M7)/(DEL(I)**2)
2900=        MAA3(I,I)=(M3-2.0*M(3)+M8)/(DEL(I)**2)
2910=        MAA4(I,I)=(M4-2.0*M(4)+M9)/(DEL(I)**2)
2920=        MAA5(I,I)=(M5-2.0*M(5)+M10)/(DEL(I)**2)
2930=        GAA(I,I)=(G1-2.0*G+G2)/(DEL(I)**2)
2940=100     CONTINUE
2950=        IF(IMET-1)759,759,747
2960=C
2970=C   FINDING INITIAL LAGRANGE MULTIPLIERS AND DA'S (GRADIENT TECH)
2980=C
2990=759     DO 18 I=1,NC
3000=        DO 18 J=1,NC
```

```
3010=          MAMAT(I,J)=0.0
3020=          DO 18 K=1,NP
3030=18        MAMAT(I,J)=MAMAT(I,J)+MA(I,K)*MA(J,K)
3040=          CALL GAUSD(NC,1.0E-30,MAMAT,MAMATI,IER,NC)
3050=          DO 19 I=1,NC
3060=          MAGAT(I)=0.0
3070=          DO 19 J=1,NP
3080=19        MAGAT(I)=MAGAT(I)+MA(I,J)*GA(J)
3090=          DO 20 I=1,NC
3100=          PNU(I)=0.0
3110=          DO 20 J=1,NC
3120=20        PNU(I)=PNU(I)+MAMATI(I,J)*(M(J)-MAGAT(J))
3130=          PRINT*," "
3140=          DO 21 I=1,NC
3150=21        PRINT*,"PNU(",I,")=",PNU(I)
3160=          PRINT*," "
3170=          DO 34 I=1,NP
3180=          E(I)=0.0
3190=          DO 34 J=1,NC
3200=34        E(I)=E(I)+MA(J,I)*PNU(J)
3210=          DO 35 I=1,NP
3220=          FA(I)=GA(I)+E(I)
3230=35        DA(I)=-P*FA(I)
3240=          F=0.0
3250=          DO 36 I=1,NP
3260=36        F=F+DA(I)**2
3270=          GRNORM=SQRT(F)
3280=          DIFF=(FGNORM-GRNORM)/FGNORM
3290=          FGNORM=GRNORM
3300=          PRINT*," "
3310=          PRINT*,"FA MATRIX"
3320=          PRINT 50,(FA(I),I=1,NP)
3330=          PRINT*," "
3340=          PRINT*,"GRADIENT METHOD DA'S"
3350=          PRINT 50,(DA(I),I=1,NP)
3360=          PRINT*," "
3370=          IF(DIFF.LT.CC2.AND.F.EQ.1.0) V=1.0
3380=          IF(V.EQ.1.0)PRINT*,"GRADIENT METHOD CONVERGENCE"
3390=          PRINT*," "
3400=          IF(V.EQ.1.0) P=.1
3410=          ITER=0
3420=          GO TO 1000
3430=C
3440=C
3450=C
3460=C
3470=C
  80=C
3490=C
3500=C
3510=C
3520=C   CALCULATING DPNU AND DA (SECOND ORDER TECH)
3530=C
3540=747       DO 22 I=1,NP
3550=          PNUMA(I)=0.0
3560=          DO 23 J=1,NC
3570=23        PNUMA(I)=PNUMA(I)+PNU(J)*MA(J,I)
3580=22        FAT(I)=GA(I)+PNUMA(I)
3590=          PRINT*," "
3600=          PRINT*,"PNU'S"              43
```

```
3610=          PRINT 50,(PNU(I),I=1,NC)
3620=          PRINT*," "
3630=          PRINT*,"FA MATRIX"
3640=          PRINT 50,(FAT(I),I=1,NP)
3650=          DO 24 I=1,NP
3660=          DO 24 J=1,NP
3670=          FAA(I,J)=PNU(3)*MAA3(I,J)+PNU(4)*MAA4(I,J)+PNU(5)*MAA5(I,J)
3680=24        FAA(I,J)=FAA(I,J)+PNU(1)*MAA1(I,J)+PNU(2)*MAA2(I,J)+GAA(I,J)
3690=          PRINT*,"FAA= ",FAA(I,J)
3700=          NF=NP
3710=7007      CALL GAUSD(NF,1.0E-30,FAA,FAAI,JER,9)
 720=          DO 25 I=1,NC
3730=          DO 25 J=1,NP
3740=          MAFAAI(I,J)=0.0
3750=          DO 25 K=1,NP
3760=25        MAFAAI(I,J)=MAFAAI(I,J)+MA(I,K)*FAAI(K,J)
3770=          DO 26 I=1,NC
3780=          DO 26 J=1,NC
3790=          W(I,J)=0.0
3800=          DO 26 K=1,NP
3810=26        W(I,J)=W(I,J)+MAFAAI(I,K)*MA(J,K)
3820=          CALL GAUSD(NC,1.0E-30,W,WI,KER,NC)
3830=          DO 27 I=1,NC
3840=          C(I)=0.0
3850=          DO 27 J=1,NP
3860=27        C(I)=C(I)+MAFAAI(I,J)*FAT(J)
3870=          DO 28 I=1,NC
3880=          DPNU(I)=0.0
3890=          DO 28 J=1,NC
3900=28        DPNU(I)=DPNU(I)+WI(I,J)*(-P*C(J)+Q*M(J))
3910=          PRINT*," "
3920=          PRINT*,"DPNU'S"
3930=          PRINT 50,(DPNU(I),I=1,NC)
3940=          DO 29 I=1,NP
3950=          D(I)=0.0
3960=          DO 29 J=1,NC
3970=29        D(I)=D(I)+MA(J,I)*DPNU(J)
3980=          DO 30 I=1,NP
3990=          DA(I)=0.0
4000=          DO 30 J=1,NP
4010=30        DA(I)=DA(I)+FAAI(I,J)*(-P*FAT(J)-D(J))
4020=          NORMP=0.0
4030=          DO 31 I=1,NC
4040=31        NORMP=NORMP+DPNU(I)**2
4050=          NORMP=SQRT(NORMP)
4060=          PRINT*," "
 )70=          PRINT*,"NORM OF DPNU'S= ",NORMP
4080=          PRINT*," "
4090=          PRINT*,"DA MATRIX"
4100=          PRINT 50,(DA(I),I=1,NP)
4110=          NORMA=0.0
4120=          DO 32 I=1,NP
4130=32        NORMA=NORMA+DA(I)**2
4140=          NORMA=SQRT(NORMA)
4150=          DIF=(FNORMA-NORMA)/FNORMA
4160=          FNORMA=NORMA
4170=          PRINT*," "
4180=          PRINT*,"NORM OF DA'S = ",NORMA
4190=          IF(ITER.GE.10.0)GO TO 555
4200=          IF(NORMA.GE.1.0)GO TO 4371          44
```

```
4210=        Q=1.
4220=C
4230=C    CONVERGENCE CRITERIA
4240=C
4250=200     IF(NORMM.LE.CC3.AND.NORMA.LE.CC4.AND.P.EQ.1.0) GO TO 201
4260=        IF(P.GT.1.0) P=1.0
4270=        GO TO 1000
4280=4371  Q=Q/100.
4290=        GO TO 555
4300=201   PRINT*," "
4310=        PRINT*,"*CONVERGENCE*        PERFORMANCE INDEX= ",G,"    P= ",P
`20=757    STOP
.030=        END
4340=C
4350=C
4360=C    MATRIX INVERSION SUBROUTINE
4370=C
4380=C
4390=        SUBROUTINE GAUSD(M,EPS,B,C,KER,LAY)
4400=        DIMENSION B(LAY,LAY), C(LAY,LAY),A(20,20), X(20,20)
4410=        DOUBLE PRECISION Z,A,X,S,RATIO,EP
4420=        EP = EPS
4430=        N = M
4440=        DO 100  J = 1,N
4450=        DO 100  K = 1,N
4460=  100 A(J,K) = B(J,K)
4470=        DO 1 I=1,N
4480=        DO 1 J=1,N
4490=    1 X(I,J) = 0.0D0
4500=        DO 2 K=1,N
4510=    2 X(K,K) = 1.0D0
4520=   10  DO 34 L=1,N
4530=        KP=0
4540=        Z = 0.0D0
4550=        DO 12 K=L,N
4560=        IF(Z-DABS(A(K,L))) 11,12,12
4570=   11 Z = DABS(A(K,L))
4580=        KP=K
4590=   12  CONTINUE
4600=        IF(L-KP)13,20,20
4610=   13  DO 14 J=L,N
4620=        Z=A(L,J)
4630=        A(L,J)=A(KP,J)
4640=   14  A(KP,J)=Z
4650=        DO 15 J=1,N
4660=        Z=X(L,J)
`70=        X(L,J)=X(KP,J)
.80=   15  X(KP,J)=Z
4690=   20 IF(DABS(A(L,L))-EP)50,50,30
4700=   30  IF(L-N)31,34,34
4710=   31  LP1=L+1
4720=        DO 36 K=LP1,N
4730=        IF(A(K,L))32,36,32
4740=   32 RATIO=A(K,L)/A(L,L)
4750=        DO 33 J=LP1,N
4760=   33 A(K,J)=A(K,J)-RATIO*A(L,J)
4770=        DO 35 J=1,N
4780=   35  X(K,J)=X(K,J)-RATIO*X(L,J)
4790=   36  CONTINUE
4800=   34  CONTINUE
```

45

```
4810=    40   DO 43 I=1,N
4820=         II=N+1-I
4830=         DO 43 J=1,N
4840=       S = 0.0D0
4850=         IF(II-N)41,43,43
4860=    41   IIP1=II+1
4870=         DO 42 K=IIP1,N
4880=    42   S=S+A(II,K)*X(K,J)
4890=    43   X(II,J)=(X(II,J)-S)/A(II,II)
4900=         KER=1
4910=         DO 200   J = 1,N
4920=         DO 200   K = 1,N
4930=   200 C(J,K) = X(J,K)
/40=          GO TO 75
4950=    50   KER=2
4960=    70 PRINT 71
4970=    71 FORMAT(1X,*MATRIX SINGULAR IN GAUSD*)
4980=    75   CONTINUE
4990=         RETURN
5000=         END
5010=C
5020=C
5030=C
5040=         SUBROUTINE CHEBY(T)
5050=         DIMENSION Z(7)
5060=         COMMON/MISC/NPH,NFI,NA,TF,N,S,LL,FF,TOT(8),FIXT
5070=         COMMON/QUES/ALPHA
5080=         Z(1)=T
5090=         DO 1 I=2,7
5100=         K=I-1
5110=1        Z(I)=Z(K)*T
5120=         TOT(1)=1.0
5130=         TOT(2)=2.0*Z(1)-1.0
5140=         TOT(3)=8.0*Z(2)-8.0*Z(1)+1.0
5150=         TOT(4)=32.0*Z(3)-48.0*Z(2)+18.0*Z(1)-1.0
5160=         TOT(5)=128.0*Z(4)-256.0*Z(3)+160.0*Z(2)-32.0*Z(1)+1.0
5170=         TOT(6)=512.0*Z(5)-1280.0*Z(4)+1120.0*Z(3)-400.0*Z(2)+50.0*Z(1)-1.0
5180=         TOT(7)=2048.0*Z(6)-6144.0*Z(5)+6912.0*Z(4)-3584.0*Z(3)+840.0*Z(2)
5190=       1 -72.0*Z(1)+1.0
5200=         TOT(8)=8192.0*Z(7)-28672.0*Z(6)+39424.0*Z(5)-26880.0*Z(4)
5210=       1 +9408.0*Z(3)-1568.0*Z(2)+98.0*Z(1)-1.0
5220=         RETURN $ END
5230=         SUBROUTINE TRANS(A,B)
5240=         DIMENSION A(9),B(6)
5250=         COMMON X,Y,Z,XD,YD,ZD
5260=         X=B(1)
5270=         Y=B(2)
5280=         Z=B(3)
290=          XD=B(4)
5300=         YD=B(5)
5310=         ZD=B(6)
5320=         T1=A(1)
5330=         A1=A(2)
5340=         B1=A(3)
5350=         T2=A(4)
5360=         A2=A(5)
5370=         B2=A(6)
5380=         T3=A(7)
5390=         A3=A(8)
5400=         B3=A(9)
```

46

```
5410=          TOF=T1
5420=          CALL FNG(TOF,X,Y,Z,XD,YD,ZD)
5430=C
5440=C         VELOCITY CHANGE DUE TO FIRING OF FIRST STAGE
5450=C
5460=          DXD=4242.175*COS(B1)*COS(A1)
5470=          DYD=4242.175*COS(B1)*SIN(A1)
5480=          DZD=4242.175*SIN(B1)
5490=          XD=XD+DXD
5500=          YD=YD+DYD
5510=          ZD=ZD+DZD
5520=          TOF=T2
  30=          CALL FNG(TOF,X,Y,Z,XD,YD,ZD)
U540=C
5550=C         VELOCITY CHANGE DUE TO FIRING OF SECOND STAGE
5560=C
5570=          DXD=9565.712*COS(B2)*COS(A2)
5580=          DYD=9565.712*COS(B2)*SIN(A2)
5590=          DZD=9565.712*SIN(B2)
5600=          XD=XD+DXD
5610=          YD=YD+DYD
5620=          ZD=ZD+DZD
5630=          TOF=T3
5640=          CALL FNG(TOF,X,Y,Z,XD,YD,ZD)
5650=C
5660=C         VELOCITY CHANGE DUE TO FIRING OF THIRD STAGE
5670=C
5680=          DXD=10997.798*COS(B3)*COS(A3)
5690=          DYD=10997.798*COS(B3)*SIN(A3)
5700=          DZD=10997.798*SIN(B3)
5710=          XD=XD+DXD
5720=          YD=YD+DYD
5730=          ZD=ZD+DZD
5740=          R=(X*X+Y*Y+Z*Z)**0.5
5750=          V=(XD*XD+YD*YD+ZD*ZD)**0.5
5760=          RETURN
5770=          END
5780=          SUBROUTINE FNG(TOF,X,Y,Z,XD,YD,ZD)
5790=          AMU=1.4076468E16
5800=          E=1.0E-06
5810=          RO=(X*X+Y*Y+Z*Z)**0.5
5820=          EPS=(XD*XD+YD*YD+ZD*ZD)/2.0-AMU/RO
5830=          A=-AMU/(2.0*EPS)
5840=          XN=1.0
5850=1         ZZ=XN*XN/A
5860=          C=(1.0-COS(ZZ**0.5))/ZZ
5870=          S=((ZZ**0.5)-SIN(ZZ**0.5))/(ZZ**1.5)
 380=          TN=(X*XD+Y*YD+Z*ZD)*C*(XN*XN)/(AMU**0.5)
U890=          TN=TN+(1.0-RO/A)*S*(XN**3.0)
5900=          TN=TN+RO*XN
5910=          TN=TN/(AMU**0.5)
5920=          RN=(XN*XN*C)+(X*XD+Y*YD+Z*ZD)*(1.0-ZZ*S)/(AMU**0.5)
5930=          RN=RN+RO*(1.0-ZZ*C)
5940=          DT=TOF-TN
5950=          IF(DT.LE.E)GO TO 99
5960=          XN=XN+DT*(AMU**0.5)/RN
5970=          GO TO 1
5980=99        F=1.0-(XN*XN*C)/(RO)
5990=          G=TOF-(XN**3.0)*S/(AMU**0.5)
6000=          P=X
```

```
6010=        Q=Y
6020=        V=Z
6030=        X=F*X+G*XD
6040=        Y=F*Y+G*YD
6050=        Z=F*Z+G*ZD
6060=        GD=1.0-(XN*XN)*C/((X*X+Y*Y+Z*Z)**0.5)
6070=        FD=(AMU**0.5)*(ZZ*S-1.0)/(RO*((X*X+Y*Y+Z*Z)**0.5))
6080=        FD=FD*XN
 090=        XD=FD*P+GD*XD
6100=        YD=FD*Q+GD*YD
6110=        ZD=FD*V+GD*ZD
6120=        RETURN
6130=        END
6140=*EOR
6150=*EOF
```

48

## Vita

Mark N. Brown was born on 18 November 1951 in
Valparaiso, Indiana. He graduated from high school in
Valparaiso in 1969 and attended Purdue University, from
which he received the degree of Bachelor of Aeronautical and
Astronautical Engineering in May 1973. Upon graduation, he
received a commission in the USAF through the ROTC program.
He attended Undergraduate Pilot Training at Laughlin AFB
and was assigned to the 87th Fighter Interceptor Squadron at
K.I. Sawyer AFB, Michigan. He served initially as a T-33
pilot and then as an F-106 pilot until entering the School
of Engineering, Air Force Institute of Technology, in June
1979.

Permanent Address:   813 Wood St.
Valparaiso, IN
46383

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|

| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
|---|---|---|
| AFIT/GA/AA/80D-1 | AD-A094 811 | |

| 4. TITLE (and Subtitle) | 5. TYPE OF REPORT & PERIOD COVERED |
|---|---|
| THREE BURN INERTIAL UPPER STAGE OPTIMAL ORBIT TRANSFER | MS Thesis |
| | 6. PERFORMING ORG. REPORT NUMBER |

| 7. AUTHOR(s) | 8. CONTRACT OR GRANT NUMBER(s) |
|---|---|
| Mark N. Brown, Captain, USAF | |

| 9. PERFORMING ORGANIZATION NAME AND ADDRESS | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
|---|---|
| Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, OH 45433 | |

| 11. CONTROLLING OFFICE NAME AND ADDRESS | 12. REPORT DATE |
|---|---|
| | 12 December 1980 |
| | 13. NUMBER OF PAGES |
| | 60 |

| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | 15. SECURITY CLASS. (of this report) |
|---|---|
| | UNCLASSIFIED |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release IAW AFR 190-17

30 DEC 1980    FREDERIC C. LYNCH, Major, USAF
Director of Public Affairs

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Three Burn Orbital Transfer        Space Shuttle/Inertial Upper
Orbit transfer                                              Stage
Minimum Time Orbit Transfer        Parameter Optimization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Inertial Upper Stage (IUS) being developed for use aboard the Space Shuttle is composed of three solid fuel stages plus a satellite payload. One mission of the IUS system is to launch from a shuttle parking orbit and place the satellite in geosynchronous orbit in minimum time. Actual Space Shuttle

Block 20:

parking orbit data and IUS characteristics were used in this
study to examine the sequential timing and orientation in
inertial space of each stage as it is fired while the space-
craft moves along a transfer orbit to geosynchronous orbit.
In addition, the sensitivity of the total transfer time and
the final orbital state was found as a result of not meeting
one or all of the time and orientation parameters.
        This problem is unique in that it considers an optimal
orbit transfer problem involving solid fuel stages of fixed
thrust and burn time.  Previous work with liquid fuel engines
examined orbital transfers with the intent of minimizing the
amount of propellant or required velocity change needed to
accomplish the transfer.