AD/A-005 280

MINIMUM FUEL CONTINUOUS LOW THRUST
ORBITAL TRANSFER

Eugene A. Smith

Air Force Institute of Technology
Wright-Patterson Air Force Base, Ohio

December 1974

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GA/MC/74-5	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER **AD/A-005280**
4. TITLE (and Subtitle) MINIMUM FUEL CONTINUOUS LOW THRUST ORBITAL TRANSFER		5. TYPE OF REPORT & PERIOD COVERED AFIT Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Eugene A. Smith Captain USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AU) Wright-Patterson AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Institute of Technology (AU) Wright-Patterson AFB OH 45433		12. REPORT DATE December 1974
		13. NUMBER OF PAGES 99
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)

Approved for public release; distribution unlimited.

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

JERRY C. HIX, Captain, USAF
Director of Information

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

Minimum fuel	1201 RT
Low thrust	2111 RT
Transfer orbits	2203 UF

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151

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

Minimum fuel continuous low thrust orbit transfer is an optimal control problem which requires the application of iterative numerical methods to solve the resulting two point boundary value problem. A combination gradient-neighboring extremal algorithm gives accurate results when both the initial and terminal orbits are coplanar. These results are then used to extend the problem to include out-of-plane transfers between orbits of widely varying inclination. Control histories, in terms of thrust direction, are plotted

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

as a function of time for both coplanar and non-coplanar transfers. Steering
programs for coplanar transfers are characterized by a rapid thrust reversal
which is dependent on eccentricity for the time of its occurrence. In the
latter case, out-of-plane thrusting is primarily a function of the inclination
change between orbits. On the basis of these results, the particular numeri-
cal approach used provides an effective method for generating optimal low
thrust orbital transfer guidance.

ia

MINIMUM FUEL CONTINUOUS LOW THRUST

ORBITAL TRANSFER

THESIS

GA/MC/74-5   Eugene A. Smith
                                   Captain   USAF

Approved for public release; distribution unlimited.

$i b$

MINIMUM FUEL CONTINUOUS LOW THRUST ORBITAL 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

by

Eugene A. Smith

Captain   USAF

Graduate Astronautics

December 1974

$i \mathcal{C}$

Preface

This thesis on low thrust orbital transfer was suggested by an article which appeared in a 1971 issue of The Journal of the Astronautical Sciences by J.L. Starr and R.D. Sugar. While their analysis was limited to transfer between coplanar orbits, it was my desire to extend this problem to the more general case of transfer between orbits of differing inclination.

The application of modern control theory to such a non-linear system ultimately necessitated computer utilization. As is often the case, this computational phase of engineering was not only time-consuming, but frustrating as well.

In my case, the latter difficulty was eventually surmounted through the encouragement and understanding of my wife, Glenda. The knowledge and advice of my advisor, Professor Gerald M. Anderson of the Air Force Institute of Technology, was also invaluable towards the successful completion of this work.

Eugene A. Smith

Contents

List of Figures

v

List of Symbols

English Symbols

a	Non-dimensional parameter in gradient algorithm
b_i	Linear weighting factors in augmented cost
e	Eccentricity
$\bar{e}_i$	Unit vector in the i direction
H	Hamiltonian
H_u	Gradient of the Hamiltonian
h	Specific angular momentum
i	Inclination angle between non-coplanar orbits
J	Cost function
J_a	Augmented cost function
ℓ_i	ith direction cosine control variable
m	Mass
$\dot{m}$	Constant mass flow rate
p	Semi-latus rectum
r	Radial distance from geocenter to spacecraft
S_{ii}	Quadratic weighting factors in augmented cost
T	Thrust magnitude
TPBVP	Two point boundary value problem
TU	Geocentric time unit
t	Time
t_f	Final time
U1	Steering angle measured clockwise from local horizontal
U2	Steering angle measured clockwise out of instantaneous orbital plane

u	Scalar control variable
V_r	Radial velocity component
V_θ	Tangential velocity component
V_ϕ	Normal velocity component
$\bar{x}$	Vector of state variables
x_i	ith component of state vector
z	Direction normal to initial orbital plane

Greek Symbols

δ	Variation
ε	Non-dimensional parameter in neighboring extremal algorithm
η	Angle in right spherical triangle
θ	True anomaly in initial orbital plane
θ'	True anomaly in inclined orbital plane
$\bar{\lambda}$	Vector of costate variables
λ_i	ith component of costate vector
μ	Geocentric gravitational constant
$\bar{\nu}$	Vector of constant Lagrange multipliers
ν_i	ith component of Lagrange multiplier vector
ϕ	Angle measured clockwise from z
ϕ'	Angle measured clockwise from initial orbital plane
$\bar{\psi}$	Vector of terminal constraints
ψ_i	ith component of terminal constraint vector
Ω	Condition on Hamiltonian at final time
$\bar{\omega}$	Angular velocity vector of coordinate system

Subscripts

o   Condition at initial time

f   Condition at final time

Superscripts

$\cdot$ First time derivative

$\cdot\cdot$ Second time derivative

$\ast$ Optimal control variable

—   Vector quantity

Abstract

Minimum fuel continuous low thrust orbit transfer is an optimal control problem which requires the application of iterative numerical methods to solve the resulting two point boundary value problem. A combination gradient-neighboring extremal algorithm gives accurate results when both the initial and terminal orbits are coplanar. These results are then used to extend the problem to include out-of-plane transfers between orbits of widely varying inclination. Control histories, in terms of thrust direction, are plotted as a function of time for both coplanar and non-coplanar transfers. Steering programs for coplanar transfers are characterized by a rapid thrust reversal which is dependent on eccentricity for the time of its occurrence. In the latter case, out-of-plane thrusting is primarily a function of the inclination change between orbits. On the basis of these results, the particular numerical approach used provides an effective method for generating optimal low thrust orbital transfer guidance.

MINIMUM FUEL CONTINUOUS LOW THRUST ORBITAL TRANSFER

I. Introduction

While our present chemical propulsion systems have led to tremendous accomplishments in the field of space exploration, it seems apparent that the day of large, bulky, impulsive-thrust rockets will soon be ending. The ever increasing costs of these systems coupled with the depletion of our energy resources will soon make the development of a low thrust propulsion system, such as an ion or nuclear engine, imperative.

The purpose of this thesis is to examine the minimum fuel orbit transfer problem in the context of a continuous low thrust propulsion system. Since the propulsion system is assumed to give constant magnitude low thrust, a minimum fuel transfer becomes synonymous with a minimum time transfer. While this particular problem has been analyzed extensively in terms of coplanar orbital transfer (Ref 5:169-204), the overall purpose of obtaining numerical solutions here is two-fold: investigate the possibility of using a combined gradient ar ' neighboring extremal algorithm for coplanar transfer, and apply th :oplanar results as a reference input in extending the problem to non-coplanar transfer.

The problem is to determine the optimal thrust direction to transfer a vehicle from some point on an initial orbit, to some desired terminal orbit in minimum time, using a constant magnitude low thrust propulsion system.

In Chapter II both the initial and terminal orbits are assumed to be coplanar and coapsidal, while the problem is extended to a non-

coplanar terminal orbit in Chapter III.   In each case, the equations
of motion are developed from the restricted two-body problem with addi-
tional terms to account for vehicle thrust.   For this analysis a 5000
pound vehicle with a thrust-to-weight ratio of 0.0125 is assumed.

In this paper the problem is approached with optimal control
theory.   The thrust direction becomes the control variable and, through
the equations of motion, a Two Point Boundary Value Problem (TPBVP) is
formulated.   Numerical solutions are then obtained to the resulting
TPBVP using a combination gradient-neighboring extremal algorithm for
the case of coplanar orbits.   Solutions for the non-coplanar orbits
are obtained by a neighboring extremal algorithm using the coplanar
results as an initial reference.

The results are presented graphically in Chapter IV in terms of
the optimal transfer trajectory and optimal control history.   The con-
clusions are discussed in Chapter V.

II. Coplanar Transfer

Formulation of the Coplanar TPBVP

Using the notation as shown in Fig. 1, the state equation as developed in Appendix A are:

$$\dot{r} = Vr \tag{1}$$

$$\dot{\theta} = \frac{V\theta}{r} \tag{2}$$

$$\dot{V}r = \frac{V\theta^2}{r} - \frac{\mu}{r^2} + \frac{T \sin u}{m_o - \dot{m}t} \tag{3}$$

$$\dot{V}\theta = \frac{-VrV\theta}{r} + \frac{T \cos u}{m_o - \dot{m}t} \tag{4}$$

where $\dot{m}$ is a constant mass flow rate and u is the thrust direction as measured clockwise from the local horizontal.
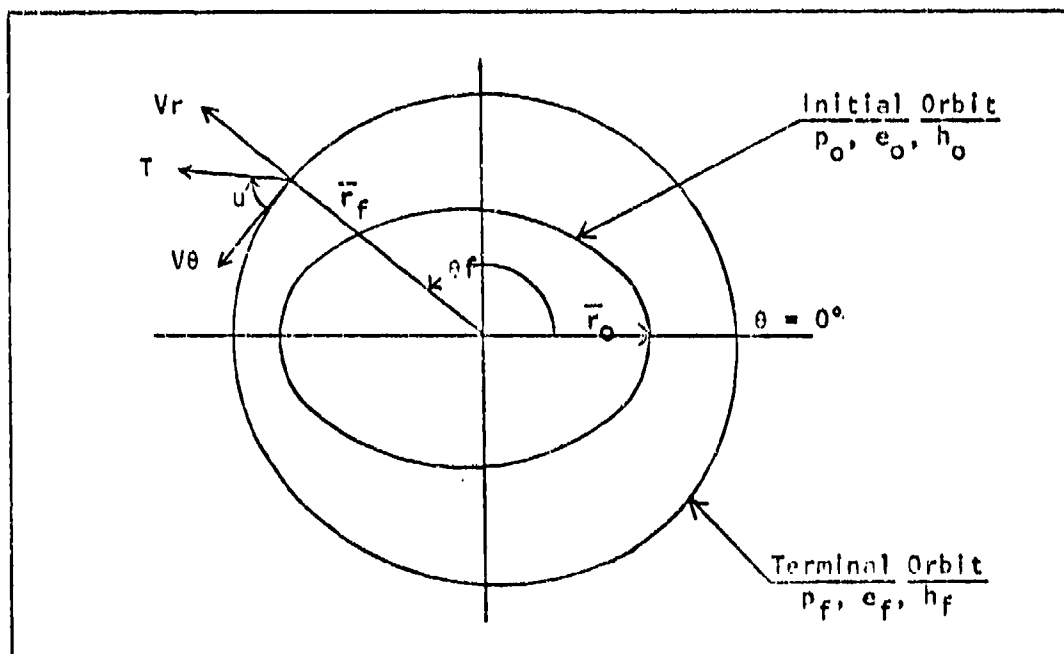
Fig. 1. Coplanar Coordinate System

Specifying the semi-latus rectum and eccentricity of the initial and final orbits as p_o, e_o, p_f, and e_f, respectively, and the true anomaly of departure from the initial orbit as θ_o, the initial conditions are given by

$$r_o = \frac{p_o}{1 + e_o \cos \theta_o} \tag{5}$$

$$Vr_o = \frac{e_o h_o \sin \theta_o}{p_o} \tag{6}$$

$$V\theta_o = \frac{h_o}{r_o} \tag{7}$$

where h_o is the specific angular momentum of the initial orbit.

The terminal constraints are then determined as

$$\psi_1 = r_f - \frac{p_f}{1 + e_f \cos \theta_f} = 0 \tag{8}$$

$$\psi_2 = Vr_f - \frac{e_f h_f \sin \theta_f}{p_f} = 0 \tag{9}$$

$$\psi_3 = V\theta_f - \frac{h_f}{r_f} = 0 \tag{10}$$

where h_f is the specific angular momentum of the terminal orbit and θ_f, the true anomaly at arrival to the terminal orbit, is unconstrained.

Using the Mayer formulation of the cost function in terms of a minimum time trajectory

$$J = t_f \tag{11}$$

which yields the following for the Hamiltonian

4

$$H = \lambda_1 Vr + \lambda_2 \frac{V\theta}{r} + \lambda_3 \left(\frac{V\theta^2}{r} - \frac{\mu}{r^2} + \frac{T \sin u}{m_o - \dot{m}t} \right)$$

$$+ \lambda_4 \left(\frac{-VrV\theta}{r} + \frac{T \cos u}{m_o - \dot{m}t} \right) \tag{12}$$

The costate equations thus become

$$\dot{\lambda}_1 = \frac{V\theta}{r^2} \lambda_2 + \left(\frac{V\theta^2}{r^2} - \frac{2\mu}{r^3} \right) \lambda_3 - \frac{VrV\theta}{r^2} \lambda_4 \tag{13}$$

$$\dot{\lambda}_2 = 0 \tag{14}$$

$$\dot{\lambda}_3 = - \lambda_1 + \frac{V\theta}{r} \lambda_4 \tag{15}$$

$$\dot{\lambda}_4 = - \frac{1}{r} \lambda_2 - \frac{2V\theta}{r} \lambda_3 + \frac{Vr}{r} \lambda_4 \tag{16}$$

with the following transversality conditions

$$\lambda_1(t_f) = \nu_1 + \nu_3 \frac{h_f}{r_f^2} \tag{17}$$

$$\lambda_2(t_f) = - \frac{p_f e_f \sin \theta_f}{(1 + e_f \cos \theta_f)^2} \nu_1 - \frac{e_f h_f \cos \theta_f}{p_f} \nu_2 \tag{18}$$

$$\lambda_3(t_f) = \nu_2 \tag{19}$$

$$\lambda_4(t_f) = \nu_3 \tag{20}$$

$$H(t_f) = - 1 \tag{21}$$

where ν_1, ν_2, and ν_3 are constant Lagrange multipliers.

Minimizing the Hamiltonian with respect to u and enforcing the strengthened Legendre-Clebsch condition gives

5

$$\sin u^* = \frac{-\lambda_3}{\sqrt{\lambda_3{}^2 + \lambda_4{}^2}} \qquad \text{and} \qquad \cos u^* = \frac{-\lambda_4}{\sqrt{\lambda_3{}^2 + \lambda_4{}^2}} \qquad (22)$$

as the optimal control.

Summary of the Coplanar TPBVP

Defining the state vector components r, θ, Vr, and $V\theta$ as x_1, x_2, x_3, and x_4, respectively, the coplanar TPBVP becomes

$$\dot{x}_1 = x_3 \qquad\qquad\qquad x_1(0) = r_o \qquad (23)$$

$$\dot{x}_2 = \frac{x_4}{x_1} \qquad\qquad\qquad x_2(0) = \theta_o \qquad (24)$$

$$\dot{x}_3 = \frac{x_4{}^2}{x_1} - \frac{\mu}{x_1{}^2} \qquad\qquad x_3(0) = Vr_o \qquad (25)$$

$$\qquad\qquad - \frac{T\lambda_3}{(m_o - \dot{m}t)\sqrt{\lambda_3{}^2 + \lambda_4{}^2}}$$

$$\dot{x}_4 = \frac{-x_3 x_4}{x_1} \qquad\qquad\quad x_4(0) = V\theta_o \qquad (26)$$

$$\qquad\qquad - \frac{T\lambda_4}{(m_o - \dot{m}t)\sqrt{\lambda_3{}^2 + \lambda_4{}^2}}$$

$$\dot{\lambda}_1 = \frac{x_4}{x_1{}^2}\lambda_2 + \qquad\qquad \lambda_1(t_f) = \nu_1 + \nu_3 \frac{h_f}{x_{1f}{}^2} \qquad (27)$$

$$\left(\frac{x_4{}^2}{x_1{}^2} - \frac{2\mu}{x_1{}^3}\right)\lambda_3$$

$$- \frac{x_3 x_4}{x_1{}^2}\lambda_4$$

$$\dot{\lambda}_2 = 0$$

$$\lambda_2(t_f) = \frac{-p_f e_f \sin x_{2f}}{(1 + e_f \cos x_{2f})^2} \nu_1$$

$$\frac{-e_f h_f \cos x_{2f}}{p_f} \nu_2 \qquad (28)$$

$$\dot{\lambda}_3 = -\lambda_1 + \frac{x_4}{x_1} \lambda_4 \qquad\qquad \lambda_3(t_f) = \nu_2 \qquad (29)$$

$$\dot{\lambda}_4 = -\frac{1}{x_1} \lambda_2 - \frac{2x_4}{x_1} \lambda_3 \qquad\qquad \lambda_4(t_f) = \nu_3 \qquad (30)$$

$$+ \frac{x_3}{x_1} \lambda_\nu$$

$$\psi_1 = x_{1f} - \frac{p_f}{1 + e_f \cos x_{2f}} = 0 \qquad (31)$$

$$\psi_2 = x_{3f} - \frac{e_f h_f \sin x_{2f}}{p_f} = 0 \qquad (32)$$

$$\psi_3 = x_{4f} - \frac{h_f}{x_{1f}} = 0 \qquad (33)$$

$$\Omega = H(t_f) + 1 = 0 \qquad (34)$$

```
 8 constants of integration      4 initial conditions
+3 unknown multipliers ν_i      +3 terminal constraints
+  t_f                          +5 transversality conditions
= 12 unknowns                   = 12 equations
```

Thus in theory it is possible to solve this TPBVP. However, due to the non-linearity of the equations and their explicit dependence on time an analytic solution could not be found.

Gradient Algorithm

The gradient method proved to be an excellent means of obtaining

7

approximate solutions to this TPBVP. The particular approach used here is the method of multipliers suggested by Hestenes (Ref 3:10). It consists of reformulating the optimal control problem by forming an augmented penalty function of the form

$$J_a = t_f + \tfrac{1}{2} (S_{11} \psi_1^2 + S_{22} \psi_2^2 + S_{33} \psi_3^2) + b_1 \psi_1 + b_2 \psi_2 + b_3 \psi_3$$

(35)

where both quadratic and linear terms in ψ have been added to Eq (11) to enforce the terminal constraints (Ref 4:408). The S terms in Eq (35) are weighting terms which may be varied as necessary according to the terminal error. The multipliers b_i are updated from the relationship

$$b_i = S_{ii} \psi_i$$

(36)

in such a manner as to increase in magnitude as the terminal conditions become closer to being satisfied.

This change in formulation eliminates the terminal constraints, Eqs (8), (9), and (10), from the problem. The only additional change is then through the transversality conditions, Eqs (17), (18), (19), and (20), which now have the form

$$\lambda_1(t_f) = S_{11} \left(x_{1f} - \frac{p_f}{1 + \sigma_f \cos x_{2f}} \right)$$

$$+ S_{33} \left(x_{4f} - \frac{h_f}{x_{1f}} \right) \frac{h_f}{x_{1f}^2} + b_1 + b_3 \frac{h_f}{x_{1f}^2}$$

(37)

8

$$\lambda_2(t_f) = S_{11}\left(x_{1f} - \frac{p_f}{1 + e_f \cos x_{2f}}\right)$$

$$\left(\frac{-p_f e_f \sin x_{2f}}{(1 + e_f \cos x_{2f})^2}\right) + S_{22}\left(x_{3f} - \frac{e_f h_f \sin x_{2f}}{p_f}\right)$$

$$\left(\frac{-e_f h_f \cos x_{2f}}{p_f}\right) - b_1\left(\frac{p_f e_f \sin x_{2f}}{(1 + e_f \cos x_{2f})^2}\right)$$

$$- b_2\left(\frac{e_f h_f \cos x_{2f}}{p_f}\right) \tag{33}$$

$$\lambda_3(t_f) = S_{22}\left(x_{3f} - \frac{e_f h_f \sin x_{2f}}{p_f}\right) + b_2 \tag{39}$$

$$\lambda_4(t_f) = S_{33}\left(x_{4f} - \frac{h_f}{x_{1f}}\right) + b_3 \tag{40}$$

The gradient algorithm is described through the following steps:

(1) With some initial estimate for u as a function of
time, e.g., $u_{ref}(t) = 0$, the original state equations,
(1) thru (4), are integrated forward in time until
$\frac{dJ_a}{dt_f} = 0$ and $\frac{d^2 J_a}{dt_f^2} > 0$ which determines an initial
final time, t_f.

(2) Eqs (37) thru (40) are used as initial conditions
with $b_i = 0$ on the first iteration to integrate the
adjoint equations, (13) thru (16), backward in time
using the non-optimal reference trajectory from
step 1.

(3) Simultaneously with step 2, the gradient, H_u,
is evaluated at each point along this reference

9

trajectory.

(4) Since to first order, $\delta J_a = H_u \, \delta u$, a one dimensional
search is made on the parameter a where

$$\delta u = -a H_u, \quad a > 0 \tag{41}$$

to decrease the augmented cost function, J_a.

(5) The control is updated through

$$u_{new} = u_{ref} + \delta u \tag{42}$$

where u_{new} becomes the new estimate in step 1.

(6) The b_i are determined from Eq (36) and added to those
used in step 2. This is an important part of this
procedure because as the terminal constraints become
closer to being satisfied, the linear terms in Eq
(35) must become dominant to ensure continued con-
vergence.

(7) Steps 1 thru 6 are repeated as necessary until the
terminal constraints are satisfied to yield the
optimal control, $u^*(t)$, and the minimum time trans-
fer trajectory.

This method is quite straightforward in approach, but as with
most gradient schemes, it requires many iterations and is slow to
converge near the optimum. However, a technique used in the actual
computer program was very effective in overcoming this problem and
will be discussed in Chapter IV. Moreover, the results obtained
served as an excellent reference input to the neighboring extremal
algorithm.

Neighboring Extremal Algorithm

A major obstacle in using a neighboring extremal algorithm is the necessity of a good initial estimate. This difficulty is overcome by using the results of the gradient method.

The particular neighboring extremal algorithm used here consists of estimating the unspecified terminal conditions of the TPBVP and generating a transition matrix, which is used to adjust the terminal conditions so as to satisfy the specified initial conditions (Ref 1: 219). This iterative procedure is as follows:

(1) Estimate the following eight parameters: x_{1f}, x_{2f}, x_{3f}, x_{4f}, v_1, v_2, v_3, and t_f. Initially this is done using the results from the gradient method.

(2) The eight parameters in step 1 determine λ_{1f}, λ_{2f}, λ_{3f}, λ_{4f}, ψ_1, ψ_2, ψ_3, and Ω from Eqs (27) thru (34).

(3) The state and costate equations are integrated backward from t_f to give x_{10}, x_{20}, x_{30}, and x_{40} which are, in general, different from the specified initial conditions.

(4) Through small perturbations in the eight parameters of step 1, a transition matrix of the form

$$\left[\frac{\partial \ (x_{10}, \ x_{20}, \ x_{30}, \ x_{40}, \ \psi_1, \ \psi_2, \ \psi_3, \ \Omega)}{\partial \ (x_{1f}, \ x_{2f}, \ x_{3f}, \ x_{4f}, \ v_1, \ v_2, \ v_3, \ t_f)} \right]$$

is numerically generated such that

$$
\begin{bmatrix} \delta \, \overline{x} \, (0) \\ d\overline{\psi} \\ d\Omega \end{bmatrix} = \left[\frac{\partial \, (x_{10}, \, x_{20}, \, x_{30}, \, x_{40}, \, \psi_1, \, \psi_2, \, \psi_3, \, \Omega)}{\partial \, (x_{1f}, \, x_{2f}, \, x_{3f}, \, x_{4f}, \, \nu_1, \, \nu_2, \, \nu_3, \, t_f)} \right] \begin{bmatrix} \delta \, \overline{x} \, (t_f) \\ d\overline{\nu} \\ dt_f \end{bmatrix}
$$

(43)

(5) The desired changes in $\delta \, \overline{x} \, (0)$, $d\overline{\psi}$, $d\Omega$ are determined from

$$
\begin{bmatrix} \delta \, \overline{x} \, (0) \\ d\overline{\psi} \\ d\Omega \end{bmatrix} = - \, \epsilon \begin{bmatrix} \overline{x}_0 - \overline{x} \, (0) \\ \overline{\psi} \\ \Omega \end{bmatrix} \qquad 0 < \epsilon \leq 1
$$

(44)

since the TPBVP is completely solved when the error vector on the right side of Eq (44) is driven to 0, or equivalently, all boundary conditions are satisfied.

(6) Inverting the transition matrix of step 4 and using the values of $\delta \, \overline{x} \, (0)$, $d\overline{\psi}$, and $d\Omega$ determined in step 5, Eq (43) is employed to solve for $\delta \, \overline{x} \, (t_f)$, $d\overline{\nu}$, and dt_f.

(7) The estimates in step 1 are now updated according to the relationship

$$
\begin{bmatrix} \overline{x}_f \\ \overline{\nu} \\ t_f \end{bmatrix}_{new} = \begin{bmatrix} \overline{x}_f \\ \overline{\nu} \\ t_f \end{bmatrix}_{step \, 1} + \begin{bmatrix} \delta \, \overline{x}(t_f) + \dot{\overline{x}}(t_f) \, dt_f \\ d\overline{\nu} \\ dt_f \end{bmatrix}
$$

(45)

and the procedure is repeated until the right side of Eq (44) is arbitrarily small.

Upon convergence of the neighboring extremal algorithm, $u^*(t)$ may be determined through Eq (22). However, such convergence is not always guaranteed. It is actually based on two unrelated factors which are covered in Chapter IV: accuracy of the initial estimate and sensitivity effects.

While both of these factors were encountered to some degree, extremely accurate results were obtained through a combination gradient-neighboring extremal algorithm. More importantly, these results provided the necessary basis for the investigation of the non-coplanar orbital transfer problem.

III. Non-Coplanar Transfer

Coordinate System

In a natural extension of the coplanar transfer problem derived in the preceding chapter, a spherical coordinate system is introduced as shown in Fig. 2. The coordinate z is normal to the initial orbital plane with the angle ϕ described in the usual manner. The orbits are coapsidal in that a clockwise rotation through the inclination angle i about the semi-latus rectum of the initial orbit results in a common apsidal line for both orbits.

While this restriction does not allow for the most general orbital transfer case, it is imposed here as an initial attempt to extend this problem to out-of-plane transfers. Moreover, the numerical technique to be presented could be used to eventually handle the most general non-coplanar transfer problem. .
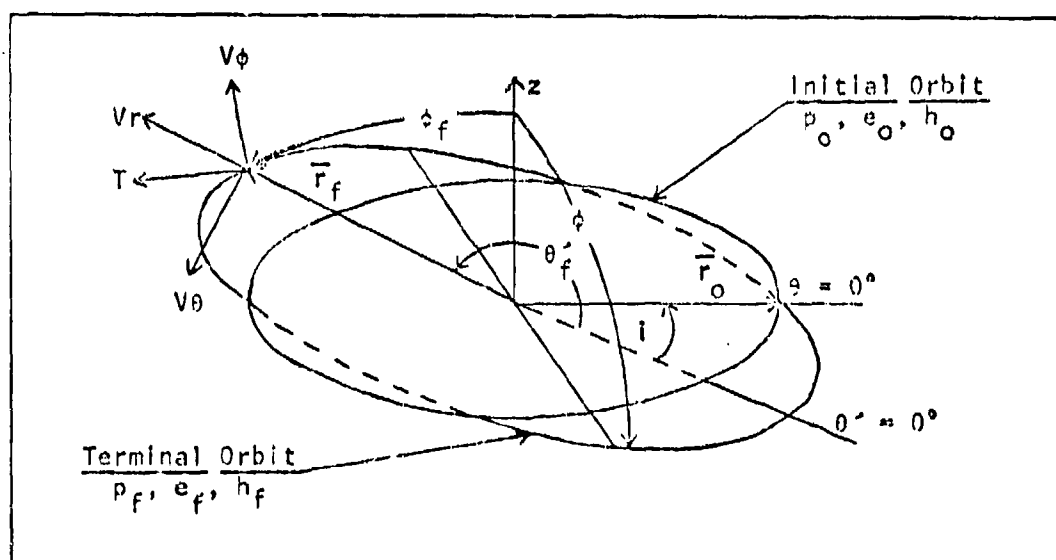

Fig. 2. Non-Coplanar Coordinate System

Formulation of the Non-Coplanar TPBVP

While this development proceeds in a similar manner to Chapter II, the spherical trigonometric relationships necessary to formulate this particular TPBVP are much more complex. For this reason, many of the details involved in the actual derivation of this problem are contained in Appendix B.

The state equations are as follows

$$\dot{r} = Vr \tag{46}$$

$$\dot{\theta} = \frac{V\theta}{r} \tag{47}$$

$$\dot{\phi} = \frac{V\phi}{r} \tag{48}$$

$$\dot{Vr} = \frac{V\phi^2}{r} + \frac{V\theta^2}{r} \sin^2 \phi - \frac{\mu}{r^2} + \frac{T\ell_1}{m_o - \dot{m}t} \tag{49}$$

$$\dot{V\theta} = \frac{-VrV\theta}{r} - \frac{2V\phi V\theta \cot \phi}{r} + \frac{T\ell_2}{(m_o - \dot{m}t) \sin \phi} \tag{50}$$

$$\dot{V\phi} = \frac{-VrV\phi}{r} + \frac{V\theta^2 \sin \phi \cos \phi}{r} + \frac{T\ell_3}{m_o - \dot{m}t} \tag{51}$$

where ℓ_1, ℓ_2, and ℓ_3 are the direction cosines between the thrust vector and the r-θ-ϕ coordinates, respectively. They are introduced in lieu of the angle control of Chapter II for ease in establishing optimality criteria later.

In addition to specifying p_o, e_o, p_f, e_f, and θ_o as in the coplanar problem, the desired inclination change i must also be selected. Initial conditions are determined by

$$r_o = \frac{p_o}{1 + e_o \cos \theta_o} \tag{52}$$

$$\psi_o = 90° \tag{53}$$

$$Vr_o = \frac{e_o h_o \sin \theta_o}{p_o} \tag{54}$$

$$V\theta_o = \frac{h_o}{r_o} \tag{55}$$

$$V\phi_o = 0 \tag{56}$$

with terminal constraints

$$\psi_1 = r_f - \frac{p_f}{1 + e_f \cos \theta'_f} = 0 \tag{57}$$

$$\psi_2 = \phi_f - [\phi_o + \arctan (\tan i \cos \theta_f)] = 0 \tag{58}$$

$$\psi_3 = Vr_f - \frac{e_f h_f \sin \theta'_f}{p_f} = 0 \tag{59}$$

$$\psi_4 = V\theta_f - \frac{h_f}{r_f} (\cos i - \sin i \cos \theta_f \cot \phi_f) = 0 \tag{60}$$

$$\psi_5 = V\phi_f + \frac{h_f \sin i \sin \theta_f}{r_f} = 0 \tag{61}$$

where

$$\theta'_f = \arccos \left\{ \frac{\sin [\arctan (\tan i \cos \theta_f)]}{\sin i} \right\} \tag{62}$$

Again the Mayer formulation of the cost function gives

$$J = t_f \tag{63}$$

and the Hamiltonian becomes

$$H = \lambda_1 \, Vr + \lambda_2 \, \frac{V\theta}{r} + \lambda_3 \, \frac{V\phi}{r}$$

$$+ \lambda_4 \left(\frac{V\phi^2}{r} + \frac{V\theta^2 \, \sin^2 \phi}{r} - \frac{\mu}{r^2} + \frac{T\ell_1}{m_0 - \dot{m}t} \right)$$

$$+ \lambda_5 \left[\frac{-VrV\theta}{r} - \frac{2V\phi V\theta \, \cot \phi}{r} + \frac{T\ell_2}{(m_0 - \dot{m}t) \, \sin \phi} \right]$$

$$+ \lambda_6 \left(\frac{-VrV\phi}{r} + \frac{V\theta^2 \, \sin \phi \, \cos \phi}{r} + \frac{T\ell_3}{m_0 - \dot{m}t} \right) \tag{64}$$

This leads to the costate equations

$$\dot{\lambda}_1 = \frac{V\theta}{r^2} \, \lambda_2 + \frac{V\phi}{r^2} \, \lambda_3 + \left(\frac{V\phi^2}{r^2} + \frac{V\theta^2 \, \sin^2 \phi}{r^2} - \frac{2\mu}{r^3} \right) \lambda_4$$

$$+ \left(\frac{-VrV\theta}{r^2} - \frac{2V\theta V\phi \, \cot \phi}{r^2} \right) \lambda_5$$

$$+ \left(\frac{-VrV\phi}{r^2} + \frac{V\theta^2 \, \sin \phi \, \cos \phi}{r^2} \right) \lambda_6 \tag{65}$$

$$\dot{\lambda}_2 = 0 \tag{66}$$

$$\dot{\lambda}_3 = \frac{-2 \sin \phi \, \cos \phi \, V\theta^2}{r} \, \lambda_4$$

$$+ \left[\frac{-2V\theta V\phi}{r \, \sin^2 \phi} + \frac{T\ell_2 \, \cos \phi}{(m_0 - \dot{m}t) \, \sin^2 \phi} \right] \lambda_5$$

$$+ \frac{V\theta^2 \, (\sin^2 \phi - \cos^2 \phi)}{r} \, \lambda_6 \tag{67}$$

$$\dot{\lambda}_4 = - \lambda_1 + \frac{V\theta}{r} \, \lambda_5 + \frac{V\phi}{r} \, \lambda_6 \tag{68}$$

$$\dot{\lambda}_5 = - \frac{1}{r} \, \lambda_2 - \frac{2V\theta \, \sin^2 \phi}{r} \, \lambda_4 + \left(\frac{Vr}{r} + \frac{2V\phi \, \cot \phi}{r} \right) \lambda_5$$

$$\frac{-2V\theta \, \sin \phi \, \cos \phi}{r} \, \lambda_6 \tag{69}$$

$$\dot{\lambda}_6 = -\frac{1}{r}\lambda_3 - \frac{2V\phi}{r}\lambda_4 + \frac{2V\theta \cot \phi}{r}\lambda_5 + \frac{Vr}{r}\lambda_6 \tag{70}$$

and transversality conditions

$$\lambda_1(t_f) = \nu_1 + \nu_4 \frac{h_f}{r_f^2}(\cos i - \sin i \cos \theta_f \cot \phi_f)$$

$$- \nu_5 \frac{h_f}{r_f^2}\sin i \sin \theta_f \tag{71}$$

$$\lambda_2(t_f) = -\nu_1 \frac{p_f e_f \sin \theta'_f}{(1 + e_f \cos \theta'_f)^2}\frac{\partial \theta'_f}{\partial \theta_f}$$

$$+ \nu_2 \frac{\sin \theta_f \tan i}{1 + \tan^2 i \cos^2 \theta_f} - \nu_3 \frac{e_f h_f \cos \theta'_f}{p_f}\frac{\partial \theta'_f}{\partial \theta_f}$$

$$- \nu_4 \frac{h_f}{r_f}\sin i \sin \theta_f \cot \phi_f$$

$$+ \nu_5 \frac{h_f}{r_f}\sin i \cos \theta_f \tag{72}$$

$$\lambda_3(t_f) = \nu_2 - \nu_4 \frac{h_f \sin i \cos \theta_f}{r_f \sin^2 \phi_f} \tag{73}$$

$$\lambda_4(t_f) = \nu_3 \tag{74}$$

$$\lambda_5(t_f) = \nu_4 \tag{75}$$

$$\lambda_6(t_f) = \nu_5 \tag{76}$$

$$H(t_f) = -1 \tag{77}$$

Minimizing the Hamiltonian with respect to the controls ℓ_1, ℓ_2, and ℓ_3 yields

$$\ell_1^* = \frac{-\lambda_4}{\sqrt{\lambda_4^2 + (\lambda_5/\sin \phi)^2 + \lambda_6^2}} \tag{78}$$

13

$$\ell_2^* = \frac{-\lambda_5/\sin\phi}{\sqrt{\lambda_4^2 + (\lambda_5/\sin\phi)^2 + \lambda_6^2}} \qquad (79)$$

$$\ell_3^* = \frac{-\lambda_6}{\sqrt{\lambda_4^2 + (\lambda_5/\sin\phi)^2 + \lambda_6^2}} \qquad (80)$$

as the optimal control variables. Note that these conditions automatically satisfy the control constraint, $\ell_1^2 + \ell_2^2 + \ell_3^2 = 1$, as well as the strengthened Legendre-Clebsch condition.

Summary of the Non-Coplanar TPBVP

Defining the state vector components r, θ, ϕ, Vr, $V\theta$, and $V\phi$ as x_1, x_2, x_3, x_4, x_5, and x_6, respectively, and letting

$$\Lambda = \sqrt{\lambda_4^2 + (\lambda_5/\sin x_3)^2 + \lambda_6^2} \qquad (81)$$

the non-coplanar TPBVP becomes

$$\dot{x}_1 = x_4 \qquad\qquad x_1(0) = r_0 \qquad (82)$$

$$\dot{x}_2 = \frac{x_5}{x_1} \qquad\qquad x_2(0) = \theta_0 \qquad (83)$$

$$\dot{x}_3 = \frac{x_6}{x_1} \qquad\qquad x_3(0) = 90° \qquad (84)$$

$$\dot{x}_4 = \frac{x_6^2}{x_1} + \frac{x_5^2}{x_1}\sin^2 x_3 \qquad\qquad x_4(0) = Vr_0 \qquad (85)$$

$$\quad - \frac{\mu}{x_1^2} - \frac{T\lambda_4}{(m_0 - \dot{m}t)\,\Lambda}$$

$$\dot{x}_5 = \frac{-x_4 x_5}{x_1} - \frac{2x_5 x_6 \cot x_3}{x_1} \qquad\qquad x_5(0) = V\theta_0 \qquad (86)$$

$$\quad - \frac{T\lambda_5}{(m_0 - \dot{m}t)\,\Lambda\sin x_3}$$

$$\dot{x}_6 = \frac{-x_4 x_6}{x_1}$$

$$+ \frac{x_5{}^2 \sin x_3 \cos x_3}{x_1}$$

$$- \frac{T\lambda_6}{(m_o - \dot{m}t)\,\Lambda}$$

$$x_6(0) = 0 \qquad\qquad (87)$$

$$\dot{\lambda}_1 = \frac{x_5}{x_1{}^2}\,\lambda_2 + \frac{x_6}{x_1{}^2}\,\lambda_3$$

$$+ \left(\frac{x_6{}^2 + x_5{}^2 \sin^2 x_3}{x_1{}^2} - \frac{2\mu}{x_1{}^3}\right)\lambda_4$$

$$+ \left(\frac{-x_4 x_5 - 2 x_5 x_6 \cot x_3}{x_1{}^2}\right)\lambda_5$$

$$+ \left(\frac{-x_4 x_6 + x_5{}^2 \sin x_3 \cos x_3}{x_1{}^2}\right)\lambda_6$$

$$\lambda_1(t_f) = \nu_1 + \nu_4 \frac{h_f}{x_{1f}{}^2}\,(\cos i$$

$$- \sin i \cos x_{2f} \cot x_{3f})$$

$$- \nu_5 \frac{h_f}{x_{1f}{}^2} \sin i \sin x_{2f}$$

$$(83)$$

$$\dot{\lambda}_2 = 0$$

$$\lambda_2(t_f) = - \nu_1 \frac{p_f e_f \sin \theta^{*}_f}{(1 + e_f \cos \theta^{*}_f)^2}\,\frac{\partial \theta^{*}_f}{\partial \theta_f}$$

$$+ \nu_2 \frac{\sin x_{2f} \tan i}{1 + \tan^2 i \cos^2 x_{2f}}$$

$$- \nu_4 \frac{h_f}{x_{1f}} \sin i \sin x_{2f} \cot x_{3f}$$

$$+ \nu_5 \frac{h_f}{x_{1f}} \sin i \cos x_{2f} \qquad\qquad (89)$$

$$\dot{\lambda}_3 = \frac{-2 \sin x_3 \cos x_3 x_5{}^2}{x_1}\,\lambda_4$$

$$+ \left[\frac{-2 x_5 x_6}{x_1 \sin^2 x_3}\right.$$

$$\left. - \frac{T\lambda_5 \cos x_3}{(m_o - \dot{m}t)\,\Lambda \sin^2 x_3}\right]\lambda_5$$

$$\lambda_3(t_f) = \nu_2$$

$$- \nu_4 \frac{h_f \sin i \cos x_{2f}}{x_{1f} \sin^2 x_{3f}}$$

20

$$+ \frac{x_5{}^2 (\sin^2 x_3 - \cos^2 x_3)}{x_1} \lambda_6 \tag{90}$$

$$\dot{\lambda}_4 = -\lambda_1 + \frac{x_5}{x_1} \lambda_5 + \frac{x_6}{x_1} \lambda_6 \qquad \lambda_4(t_f) = \nu_3 \tag{91}$$

$$\dot{\lambda}_5 = -\frac{1}{x_2} \lambda_2 - \frac{2x_5 \sin^2 x_3}{x_1} \lambda_4 \qquad \lambda_5(t_f) = \nu_4$$

$$+ \left(\frac{x_4}{x_1} + \frac{2x_6 \cot x_3}{x_1} \right) \lambda_5$$

$$- \frac{2x_5 \sin x_3 \cos x_3}{x_1} \lambda_6 \tag{92}$$

$$\dot{\lambda}_6 = -\frac{1}{x_1} \lambda_3 - \frac{2x_6}{x_1} \lambda_4 \qquad \lambda_6(t_f) = \nu_5$$

$$+ \frac{2x_5 \cot x_3}{x_1} \lambda_5 + \frac{x_4}{x_1} \lambda_6 \tag{93}$$

$$\psi_1 = x_{1f} - \frac{p_f}{1 + e_f \cos \theta'_f} = 0 \tag{94}$$

$$\psi_2 = x_{3f} - [\psi_o + \arctan (\tan i \cos x_{2f})] = 0 \tag{95}$$

$$\psi_3 = x_{4f} - \frac{e_f h_f \sin \theta'_f}{p_f} = 0 \tag{96}$$

$$\psi_4 = x_{5f} - \frac{h_f}{x_{1f}} (\cos i - \sin i \cos x_{2f} \cot x_{3f}) = 0 \tag{97}$$

$$\psi_5 = x_{6f} - \frac{h_f}{x_{1f}} \sin i \sin x_{2f} = 0 \tag{98}$$

$$\Omega = H(t_f) + 1 = 0 \tag{99}$$

12 constants of integration	6 initial conditions
+5 unknown multipliers ν_i	+5 terminal constraints
+t_f	+7 transversality conditions
= 18 unknowns	= 18 equations

As in the coplanar transfer TPBVP, sufficient relationships exist
to obtain numerical solutions for the non-coplanar problem. Further-
more, it is apparent that a specified inclination change of 0 reduces
this TPBVP to the coplanar case.

Neighboring Extremal Algorithm

This similarity suggests the likelihood of obtaining solutions to
the non-coplanar TPBVP by initially making a small inclination change
to an optimal coplanar transfer trajectory. The use of a gradient al-
gorithm to obtain a reference input for the neighboring extremal al-
gorithm can thus be obviated.

The particular neighboring extremal algorithm used is identical
in nature to that outlined in Chapter II and for that reason will not
be detailed again here. In fact, the only changes arise through an
inclusion of the four added initial and terminal conditions to this
enlarged 12 state system.

The original technique of specifying a small inclination change
with an optimal coplanar trajectory as a reference input, proved quite
successful in obtaining accurate numerical solutions through this
neighboring extremal algorithm. Moreover, once a solution is deter-
mined for a small inclination change, it is then possible to use this
solution as a reference input and increase the specified inclination
change. This procedure is repeated to obtain numerical results for
relatively large values of inclination change.

The major drawback of this approach, however, is the excessive number of iterations required for convergence, especially in view of sensitivity effects which may severely limit the size of the variation in inclination. This problem will be discussed in the next chapter, where a curve-fitting technique is presented and shown to be very effective in overcoming this obstacle.

IV. Discussion and Results

Computer Implementation

In applying the numerical algorithm of Chapters II and III, two computer programs were written: CPTRAN for the coplanar transfer and NCPTRAN for the non-coplanar problem. A separate listing for these two programs is contained in Appendix C and D, respectively.

For practical reasons alone, the semi-latus rectum of both the initial and final orbits remains fixed in this analysis. With one exception, transfer from the initial orbit was always specified to start at perigee. Thus the only parameters that varied were the eccentricities of the initial and terminal orbits as well as the inclination angle in the non-coplanar problem.

Geocentric canonical units were used for all computational work with the results displayed accordingly. Thus, the unit of length was the radius of the earth and the gravitational constant μ was normalized to one. This resulted in a time unit (TU) of 13.447 minutes. Vehicle mass was also normalized to one to produce a constant mass flow rate of .005. In all cases, accuracy was required to the fifth decimal place before convergence of a solution was presumed.

Analysis of the Coplanar Problem

As previously mentioned, the effectiveness of the neighboring extremal algorithm is based, to a large extent, on the reference input provided through the gradient method. It was thus desirable to achieve approximate solutions to the coplanar TPBVP from the gradient algorithm in as few iterations as possible. This was accomplished in the following manner.

The first few iterations show the augmented cost function of Eq (35) to decrease rapidly, after which a plateau is reached where further reductions in the cost become quite small. It is at this point that the b multipliers of Eq (36) are exploited to overcome this difficulty and to speed up the convergence process. In CPTRAN a check was thus made on the net decrease in J_a. If this decrease was less than 10^{-1} the time consuming process of updating the control, $u_{ref}(t)$, and starting a new iteration was bypassed. Instead, the b_i were updated according to step 6 of the algorithm until a reasonable decrease in J_a was achieved. This small adjustment was instrumental in obtaining results accurate to the second decimal place in 50 iterations.

In transitioning to the neighboring extremal algorithm, the final values of u and t_F obtained above are used to compute a reference trajectory for the coplanar TPBVP. The eight parameters needed in step 1 of the procedure are thus determined. In every case this initial estimate was sufficiently close to the optimal for convergence of the algorithm.

The linear assumption inherent in Eq (43) places great emphasis on the selection of ε in Eq (44). As the boundary conditions become closer to being satisfied this linearity assumption is more valid and the value of ε can be increased accordingly, up to a maximum value of 1. On the other hand, a poor initial estimate completely destroys such a supposition, and no matter how small ε is picked, use of this method would be questionable. The value of ε used in CPTRAN was thus continuously updated by dividing .005 by the root-mean-square of the error vector in Eq (44) until ε reached its maximum value of 1.

While a complete sensitivity analysis is beyond the scope of this thesis, a short note on sensitivity effects is in order. Due to the nature of the costate or Euler-Lagrange equations, as $\bar{x}$ increases along an extremal path $\bar{\lambda}$ decreases in magnitude. This property in turn causes large inaccuracies in the transition matrix of the neighboring extremal algorithm through round-off and integration errors (Ref 1: 215). It is then possible for an overcorrection in the terminal conditions of step 6 in the neighboring extremal algorithm and divergence from the solution.

It was found that a perturbation on the order of 10^{-2} in step 4 of the neighboring extremal algorithm produced a transition matrix equivalent to one produced using any smaller perturbation. For this reason a perturbation of .01 was used in CPTRAN to generate the transition matrix. The slight delay encountered in final convergence of the solution is thus explained by the effect of the above noted sensitivity in forming a transition matrix of sufficient accuracy to improve the previous estimate.

The effectiveness of the combination gradient-neighboring extremal algorithm is that no knowledge of the solution is required beforehand. For any initial estimate of the control, no matter how poor, accurate solutions to the TPBVP are ultimately achieved. However, this flexibility resulted in a penalty of increased computer time. Thus, once an optimal solution was obtained for a particular set of orbital parameters, it was more efficient to use this solution as an estimate in the neighboring extremal algorithm when small variations in these orbital parameters were desired. In some instances, such as a small change in the eccentricity of the terminal orbit, it was then

possible to bypass the gradient portion of this method. This caused
a 50% reduction in the computational time required for convergence.

Results for the Coplanar Problem

Transfer between two circular orbits was chosen as an obvious
starting place. The well known spiral transfer trajectory was obtained
as shown in Fig. 3, with the control history plotted as a function of
transfer time in Fig. 4. This steering program compared favorably with
that obtained by Faulders in a similar problem (Ref 2:45). On all
figures PO, EO, PF, and EF refer to the semi-latus rectum and eccen-
tricity of the initial and terminal orbit, respectively.

Results for various combinations of eccentricity were also ob-
tained. As one might suspect, there is a remarkable similarity in con-
trol histories when eccentricities are the same or even approximately
so. This effect is demonstrated by comparing Fig. 4 with the control
history plots of Figs. 5 and 6. Note that all three cases are charac-
terized by a rapid reversal in thrust direction through the halfway
point in transfer time.

This is in sharp contrast to orbits of differing eccentricity.
Corresponding trajectories and steering programs for two such cases
are displayed in Figs. 7 thru 10. Fig. 8 shows a delay in the thrust
reversal noted above whereas the opposite effect is observed in Fig.
10. That is to say, in transfering to an orbit of greater eccentri-
city the net effect is a shift of the control plot to the right.
Similarly, a shift to the left results when the initial orbit is of
larger eccentricity.

Fig. 9 shows the only case considered where departure occurred
from other than perigee, $\theta_o = 45°$ to be exact. In this particular

27
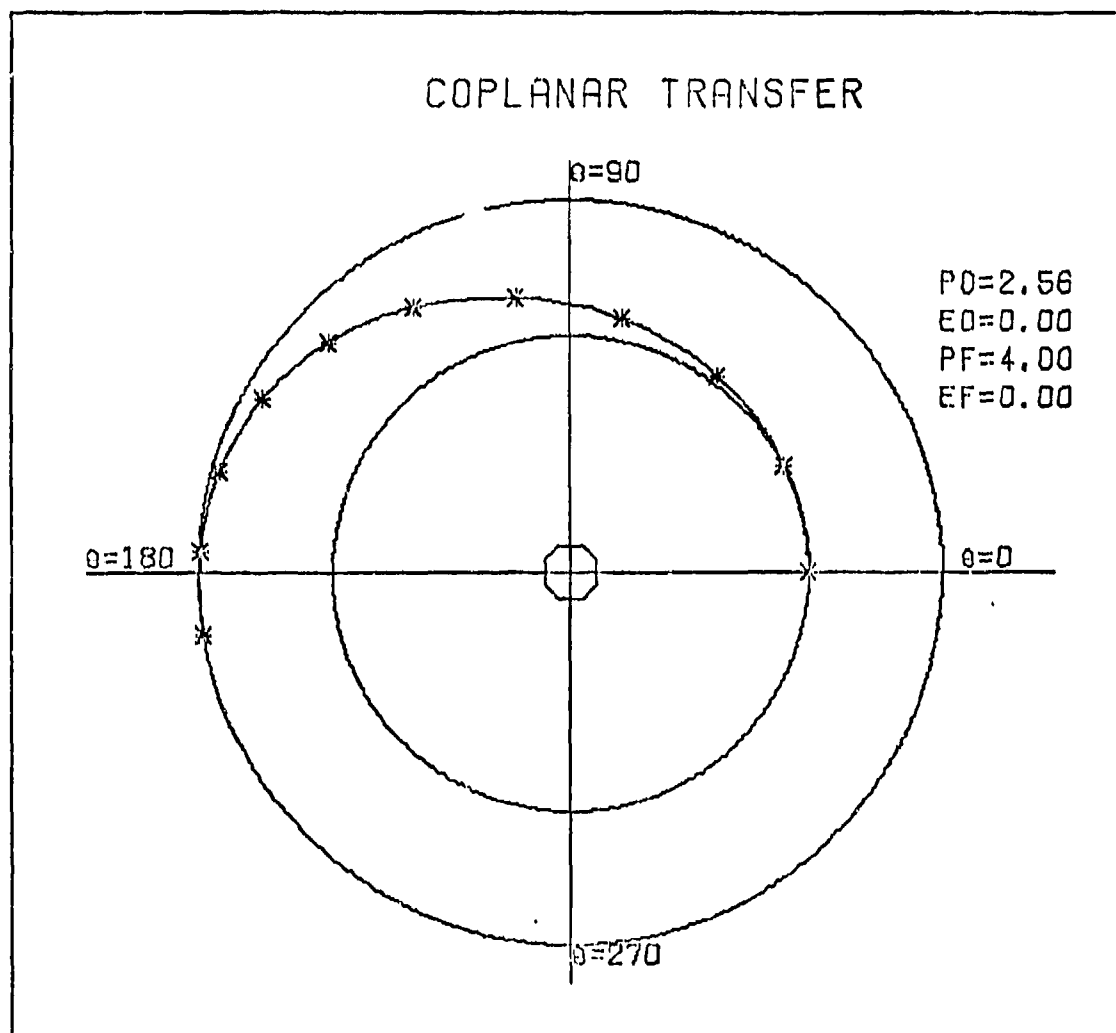
Fig. 3. Coplanar Transfer Between Circular Orbits

Fig. 4. Control History for Coplanar Transfer Between Circular Orbits

COPLANAR TRANSFER

PO=2.56
EC=0.00
PF=4.00
EF=0.00
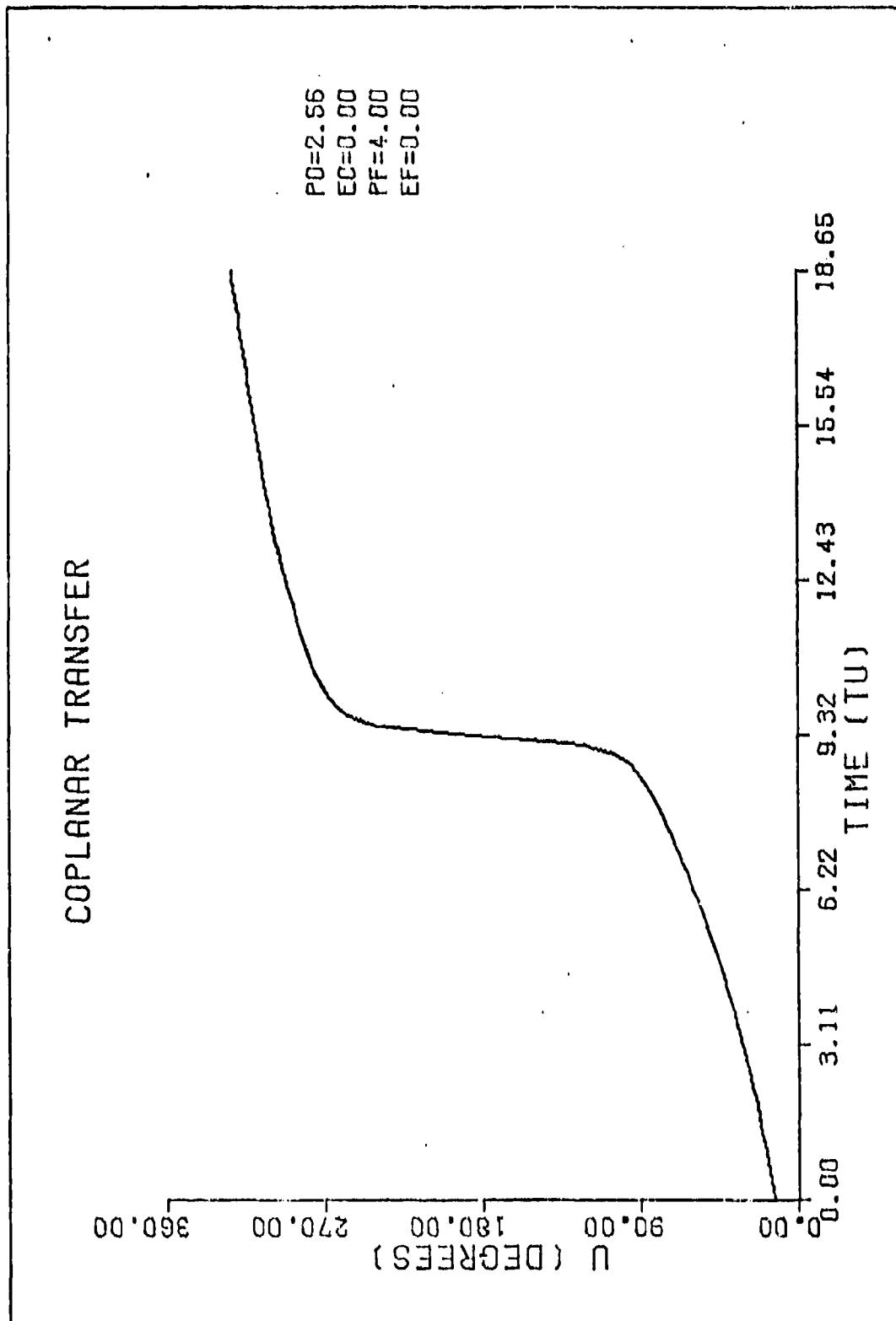
Fig. 5. Control History for Coplanar Transfer Between Elliptical Orbits(e_o = .1, e_f = .1)

Fig. 6. Control History for Coplanar Transfer Between Elliptical Orbits(e_o = .3, e_f = .25)

Fig. 7. Coplanar Transfer Between Elliptical Orbits (e_o = .1, e_f = .25)

Fig. 8. Control History for Coplanar Transfer Between Elliptical Orbits(e_o = .1, e_f = .25)
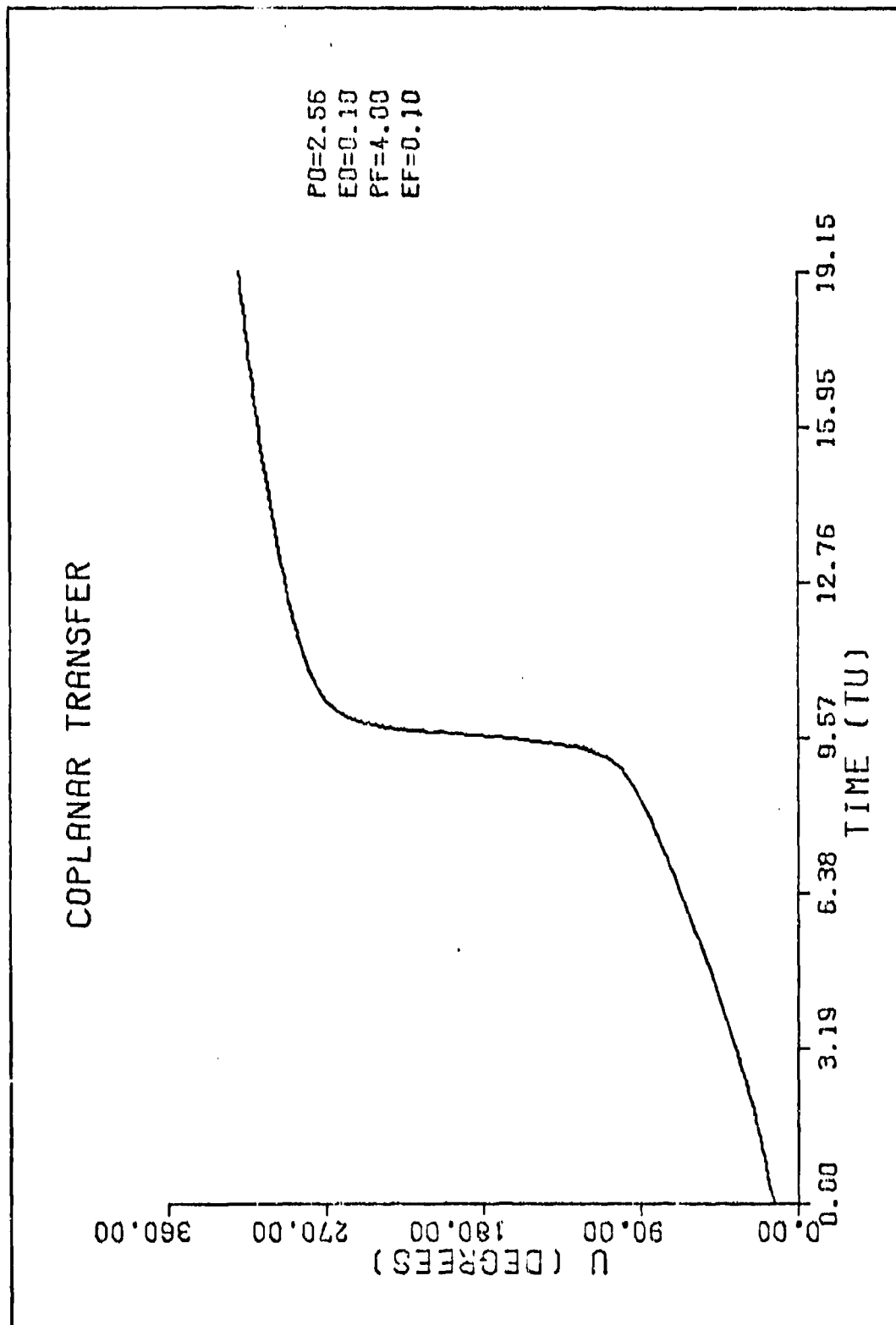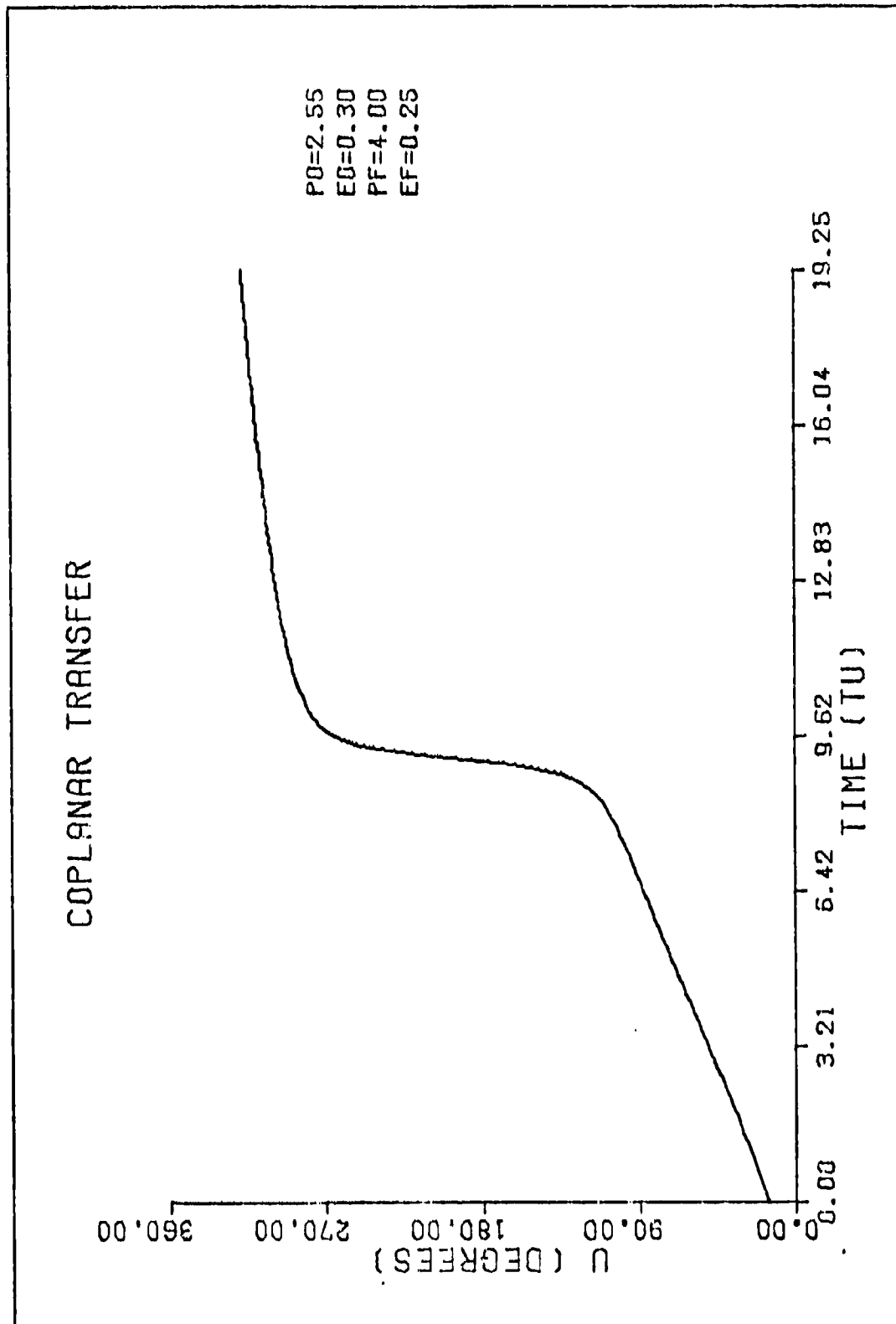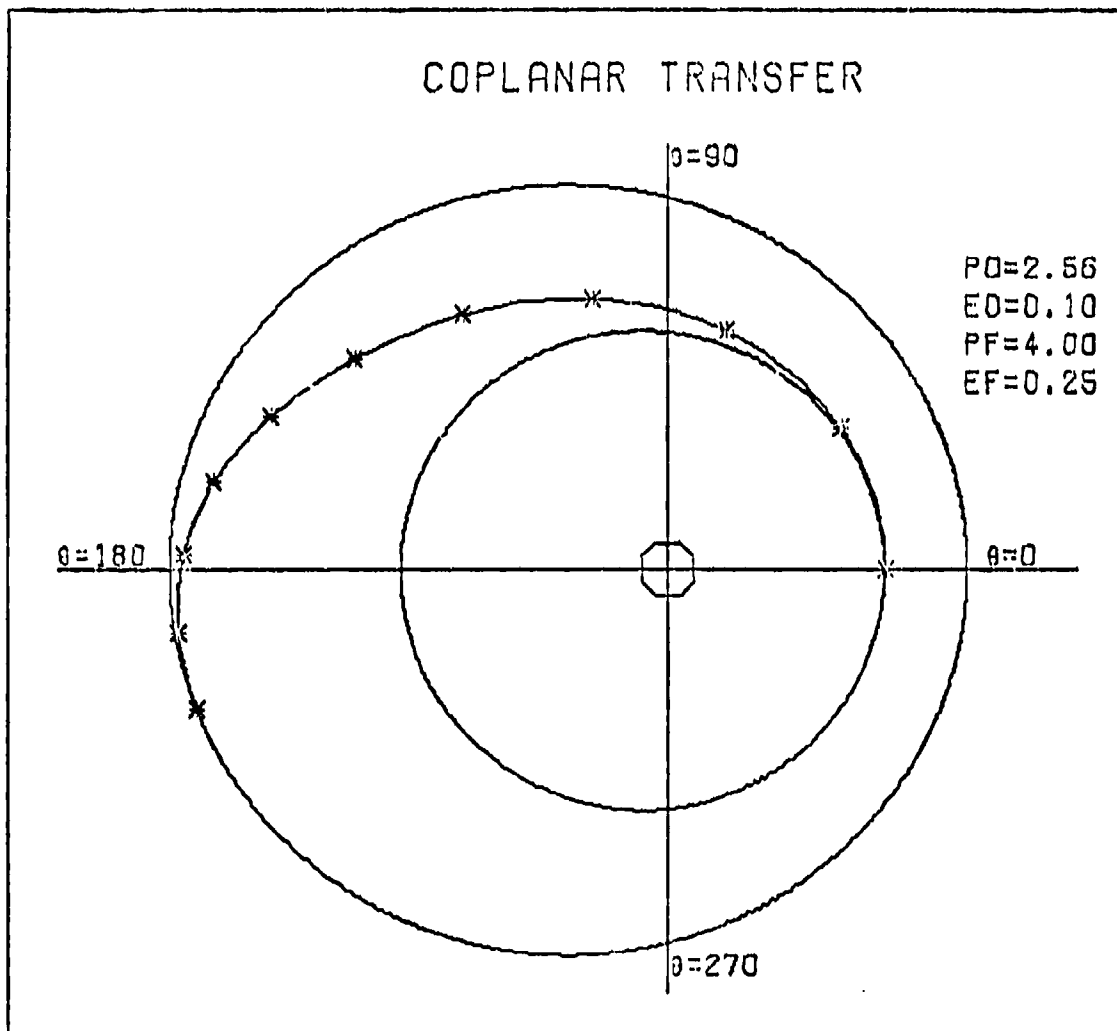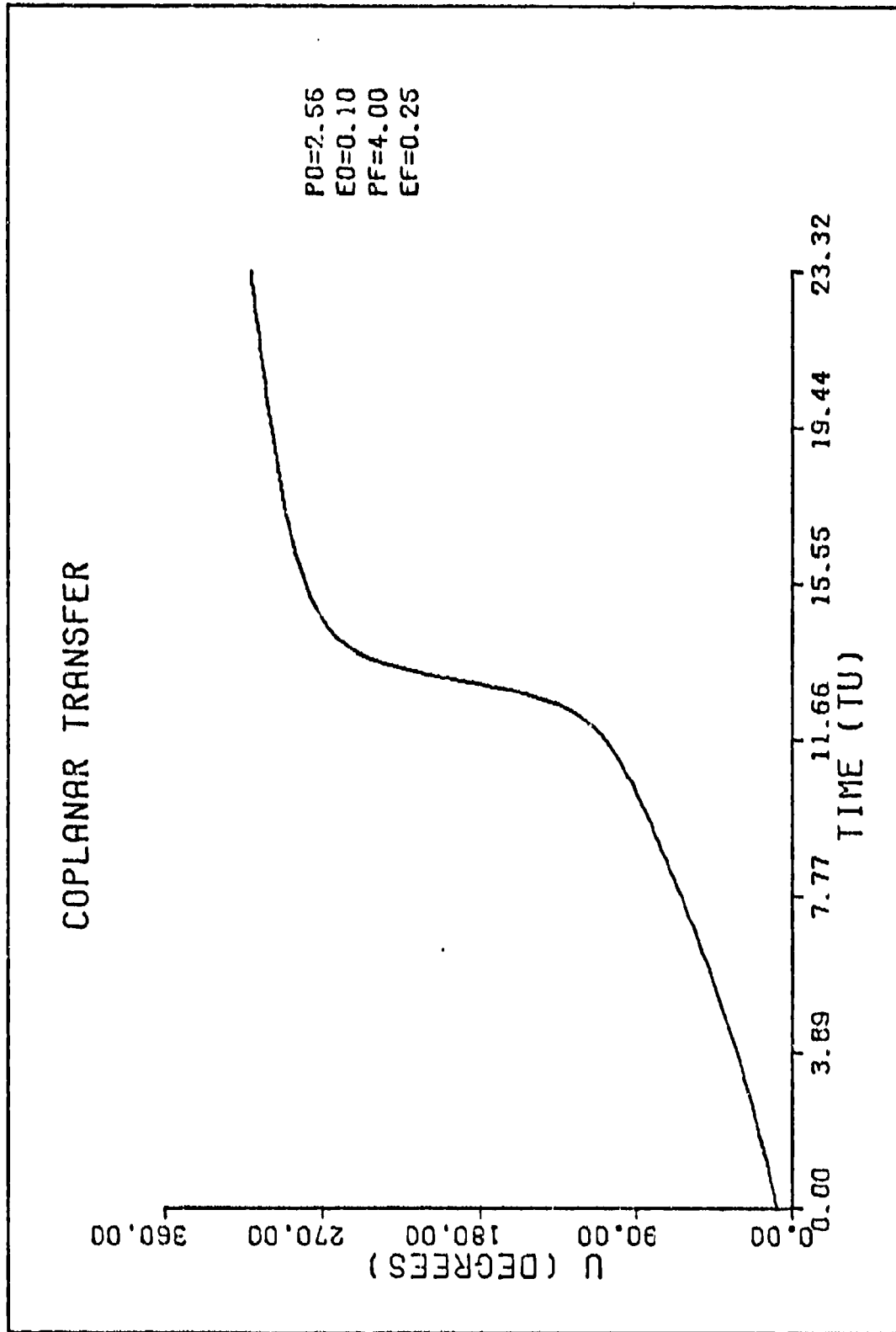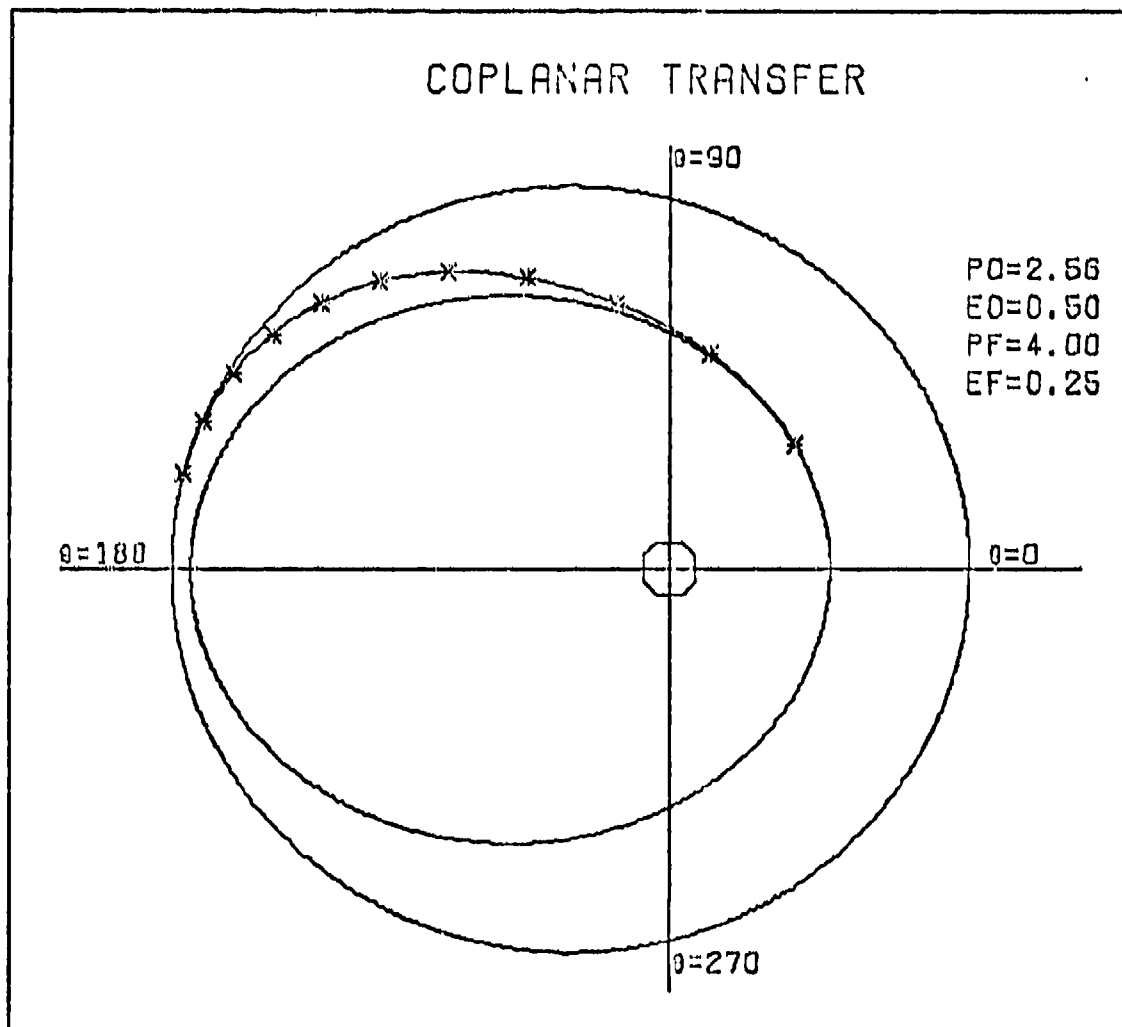
Fig. 9. Coplanar Transfer Between Elliptical Orbits (e_0 = .5, e_f = .25)
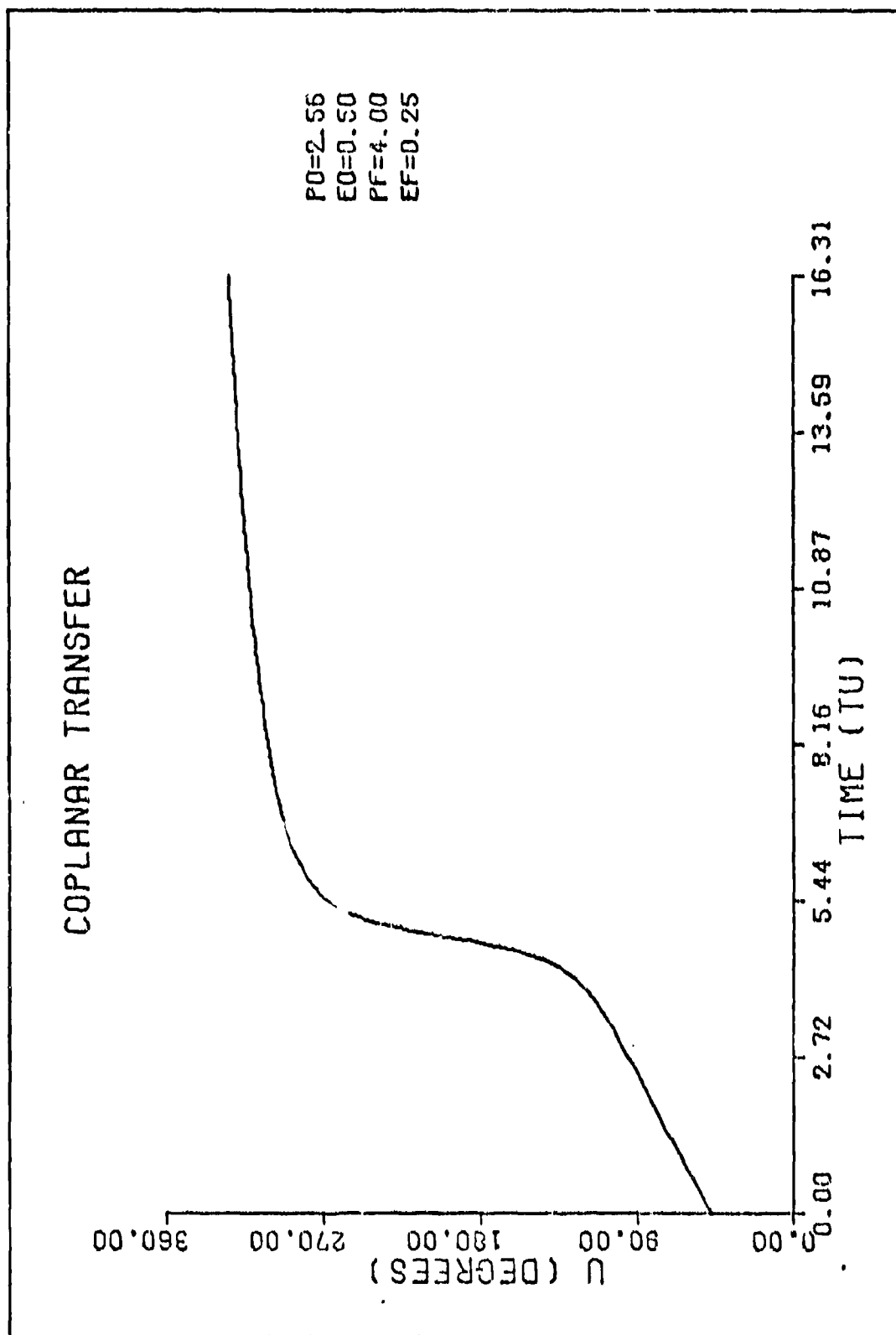
Fig. 10. Control History for Coplanar Transfer Between Elliptical Orbits(e_0 = .5, e_f = .25)

instance a minimal change in the control law resulted. However, total time of transfer was some .71 TU's less than the same problem where departure was specified at perigee. This fact is mentioned here only to show that no general statement can be made concerning transfer time. It is strictly a function of the eccentricities of the orbits and the starting point from the initial orbit.

Analysis of the Non-Coplanar Problem

In Chapter III the recursive method of obtaining solutions for increasing values of inclination between specified orbits was explained. This technique was initially employed here inasmuch as reference solutions were already available from the coplanar analysis. Very slow convergence of a gradient algorithm in this twelve state system with three control variables was also anticipated.

On the other hand, solutions for an inclination change approaching 90° were an ultimate objective. An alternative approach was thus necessary to avoid the arbitrariness of the recursive technique. For instance, after successfully achieving an inclination change (INC) of 36°, this procedure would only work for an increase in inclination of less than one-fourth of a degree using the 36° transfer as the initial estimate. This was clearly unacceptable.

The approach that was adopted was straightforward and very effective in surmounting such uncertainty. It consisted of using the twelve parameters $\bar{x}_f$, $\bar{v}$, and t_f obtained from the previous three solutions for respective values of inclination. A least square curve-fit was then made to estimate these parameters for the new value of inclination. To illustrate, consider the above mentioned problem where

the recursive technique was stalled at 36°. The solutions obtained

for inclination change of 30°, 33°, and 36° provides reference points

for each of the twelve parameters needed in step 1 of the neighboring

extremal algorithm. Curve-fitting thus provided a second order poly-

nomial which in turn produced an estimate of $\bar{x}_f$, $\bar{v}$, and t_f for an in-

clination change of 39°.

Figs. 11 and 12 show a plot of these parameters as a function of

inclination change for a transfer between the same circular orbits

used in the coplanar analysis. Besides displaying the obvious non-

linear characteristics of the problem, these graphs may be interpreted

as sensitivity plots, since they show the effect of a small variation

in inclination. Note that while small changes in inclination resulted

in relatively small changes in $\bar{x}_f$ and t_f, such is not the case for $\bar{v}$.

Again using the 36° value of inclination, it can be observed from Fig.

12 that v_2, v_3, v_4, and v_5 are rapidly changing for even a small in-

crease in inclination. This explains the difficulty encountered in the

recursive method since results for an inclination change of 36° are

thus shown to provide a poor estimate for any larger value of inclina-

tion.

As expected, upon approaching a specified inclination change of

85° convergence became extremely slow. This was so for two reasons:

the extremely long transfer times caused round-off and integration

errors to have a much greater effect than in the coplanar problem

where this factor was discussed, and, at an inclination change of 90°,

the state equations go undefined as ϕ approaches 0. This in itself

causes extreme sensitivity to any variation in a near optimal trajec-

tory. The overall result was a complete breakdown in the neighboring
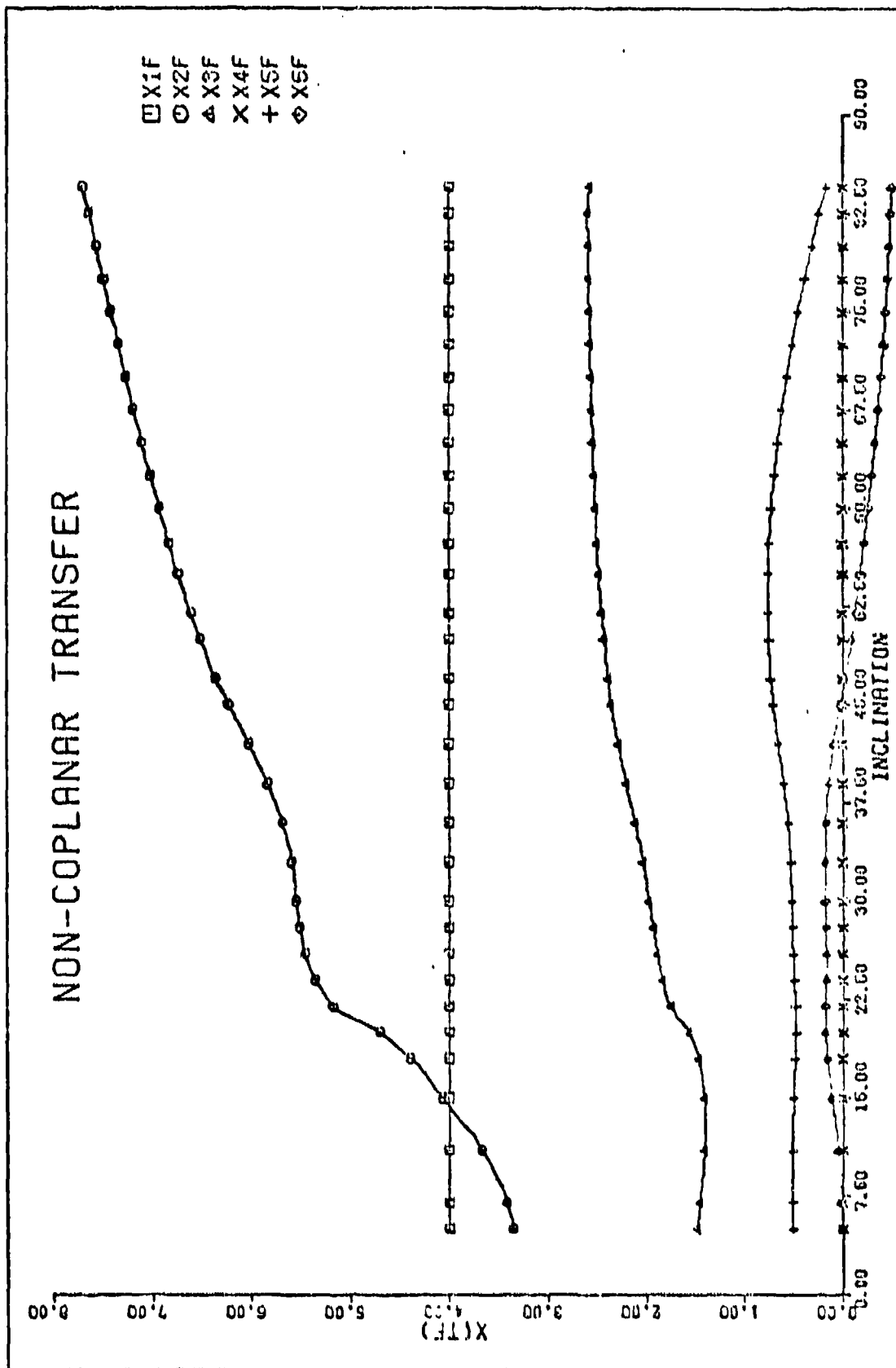
Fig. 11. Variation of $\bar{x}_f$ to Inclination Change for Transfer Between Circular Orbits
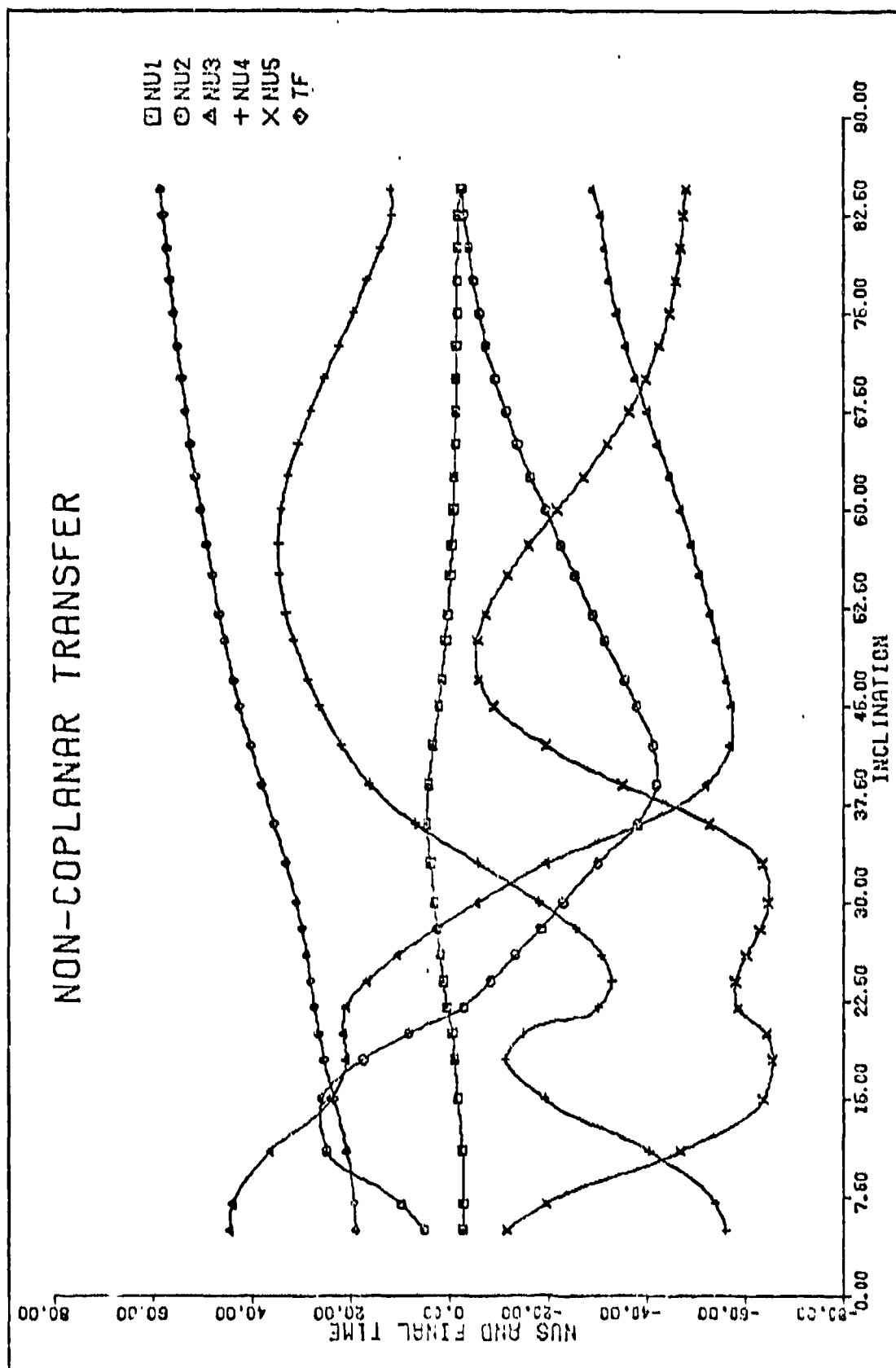
Fig. 12. Variation of $\bar{\nu}$ and t_f to Inclination Change for Transfer Between Circular Orbits

GA/MC/74-5

extremal algorithm and for this reason no solutions were obtained be-
yond an inclination change of 85°. Actually, this apparent limitation
could have been overcome by a simple rotation of the coordinate system
to avoid this singularity.

Results for the Non-Coplanar Problem

For ease in understanding, the control histories are displayed
graphically as two thrust directions. U1, in keeping with the co-
planar problem for comparative purposes, represents the thrust direc-
tion as measured clockwise from the local horizontal in the r-θ plane.
U2 is then defined as the thrust direction measured clockwise out of
the r-θ plane. Previous notation is used on all figures with the
addition of INC to denote inclination change in degrees.

Fig. 13 shows the control histories for a small out-of-plane
transfer between the same circular orbits used in the coplanar analy-
sis. Note that U1 is almost identical to the control shown in Fig. 4,
as should be the case. U2 is also reasonable since very little out-of-
plane thrusting would be required.

As demonstrated in Figs. 14 and 15, a significant change in con-
trol histories occurred as inclination was increased. In each case,
U1 bears little resemblance to any steering program obtained in the
coplanar problem, since most of the thrust is now directed out-of-
plane. On the other hand, U2 displayed a remarkable similarity for
an inclination change of 30°, even though the transfer was between
circular orbits in Fig. 14 and widely varying elliptical orbits in
Fig. 15.

These results suggest the following relationships: out-of-plane
control history (U2) is primarily a function of specified inclination

40
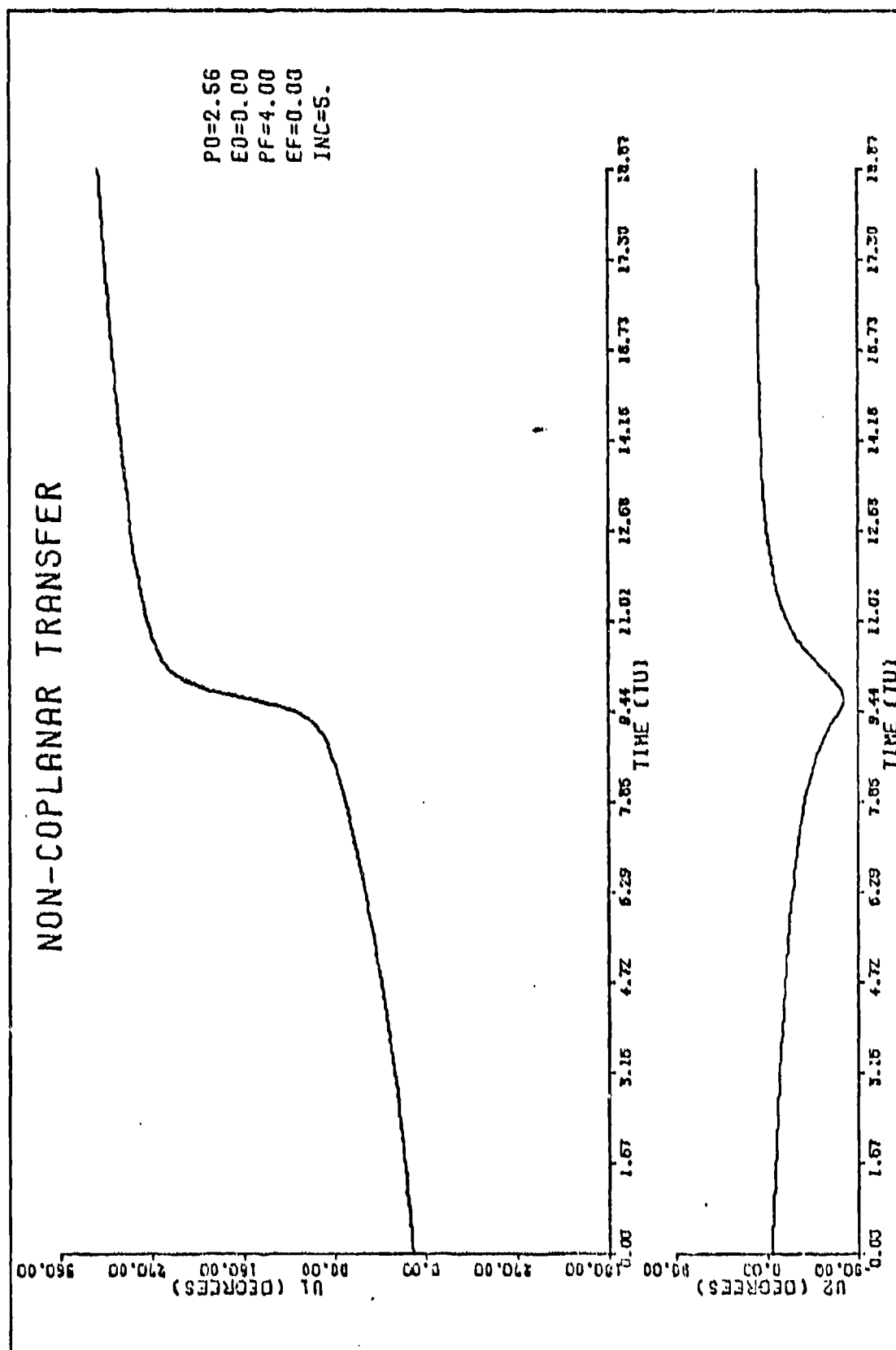
Fig. 13. Control History for Non-Coplanar Transfer Between Inclined Circular Orbits (I = 5°)
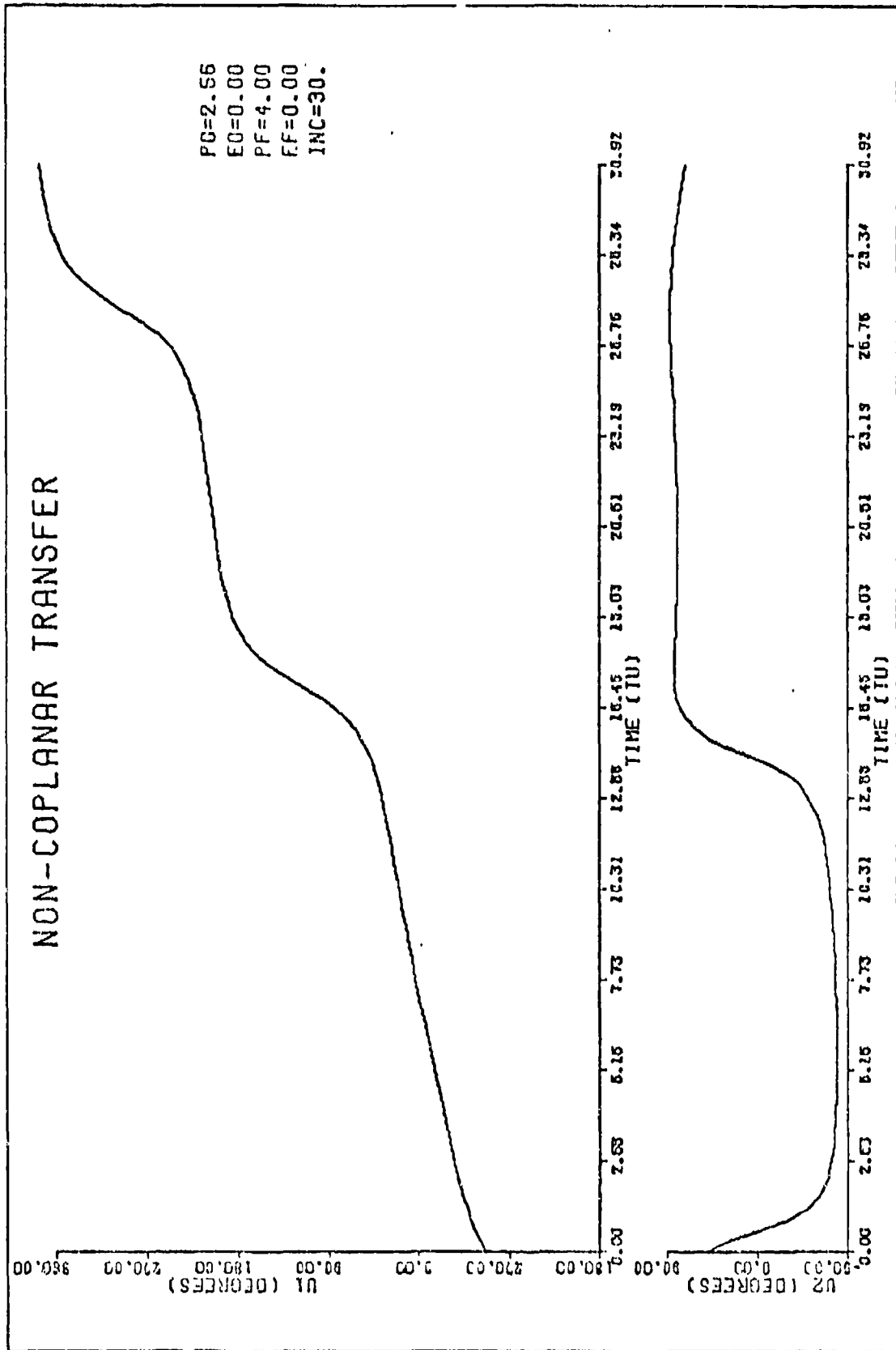
Fig. 14. Control History for Non-Coplanar Transfer Between Inclined Circular Orbits(I = 30°)
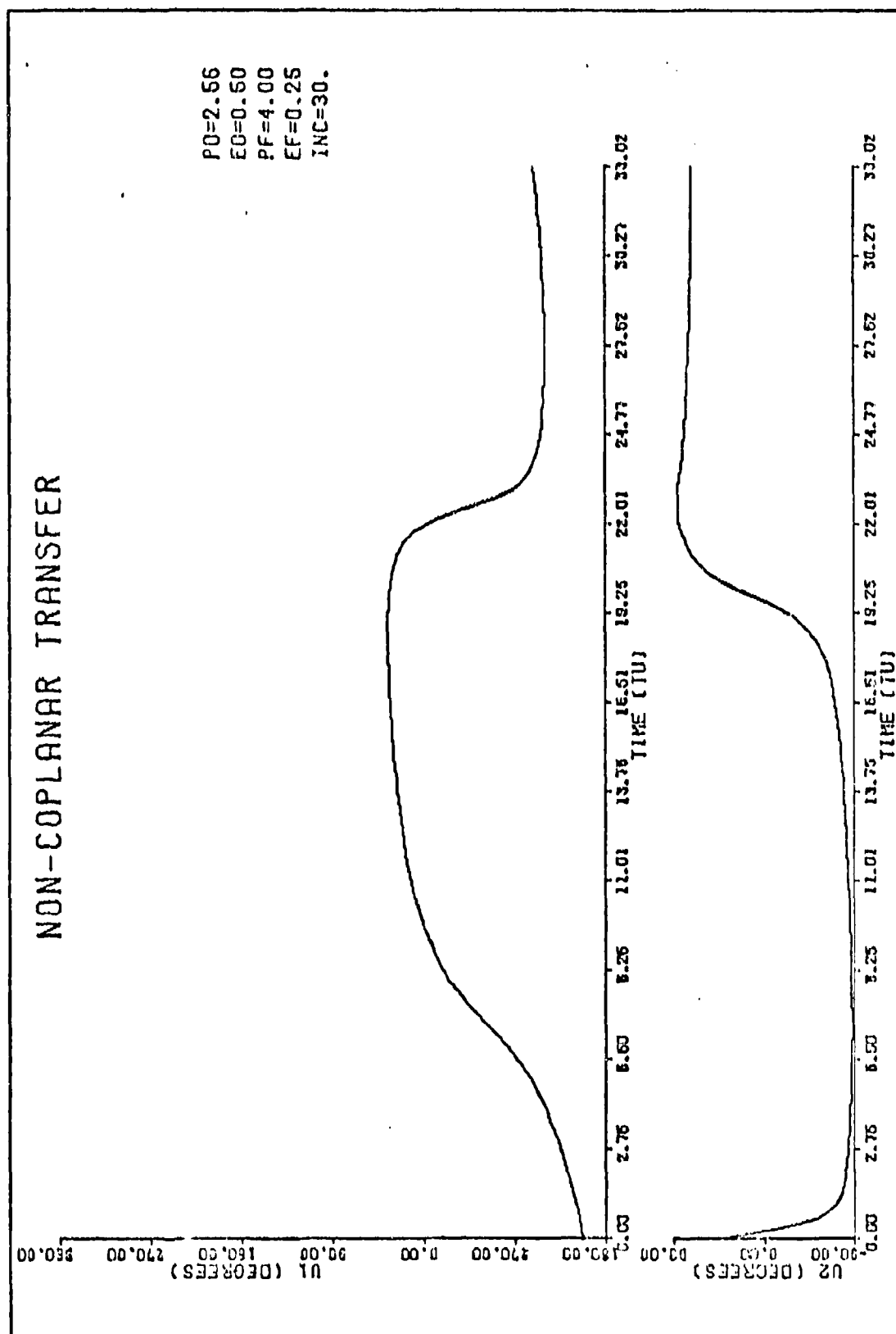
Fig. 15. Control History for Non-Coplanar Transfer Between Inclined Elliptical Orbits (i=30°, e_o=.5, e_f=.25)

change while the in-plane steering program (U1) varies with a change
in eccentricity in much the same manner as a coplanar transfer. Fig.
16 is further confirmation of these relationships.

Further inspection of Figs. 14-16 reveals that U2 is not unlike
a bang-bang controller in that the out-of-plane thrust is, for the
most part, nearly normal ($\pm 90°$) to the instantaneous orbital plane of
the transfer trajectory. This particular result was obtained in all
transfers of 30°-45° inclination change and is not surprising in view
of the coordinate system used. It is mentioned here because transfers
in this inclination range are often required to achieve a polar or
equatorial orbit for vehicles launched from the United States.

Graphical presentation of transfer trajectories was prohibitive
in view of the added dimensionality of the non-coplanar problem. How-
ever, for values of inclination greater than 45°, transfer trajectories
in excess of one earth revolution were obtained. Fig. 17 portrays a
typical control history for a transfer of this type. Due to the long
transfer time required, the previous bang-bang property of U2 is no
longer present.

Unlike the coplanar problem, transfer time between non-coplanar
orbits is very much dependent on required inclination change. In fact,
the TF plot of Fig. 12 shows transfer time as a monotonically increas-
ing function of specified inclination change for the given circular
orbits.
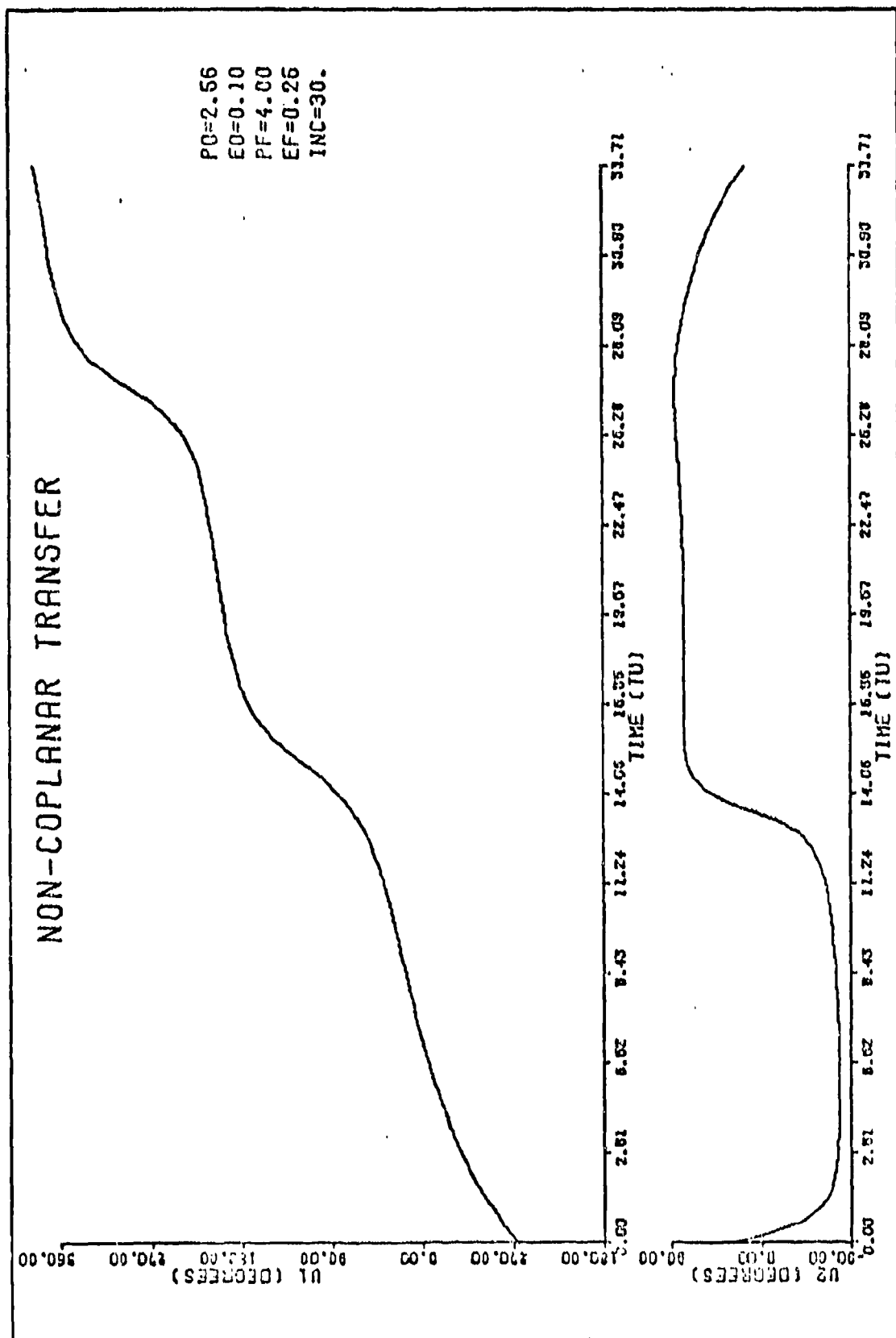
Fig. 16. Control History for Non-Coplanar Transfer Between Inclined Elliptical Orbits($i=30°$, $e_o=.1$, $e_f=.25$)
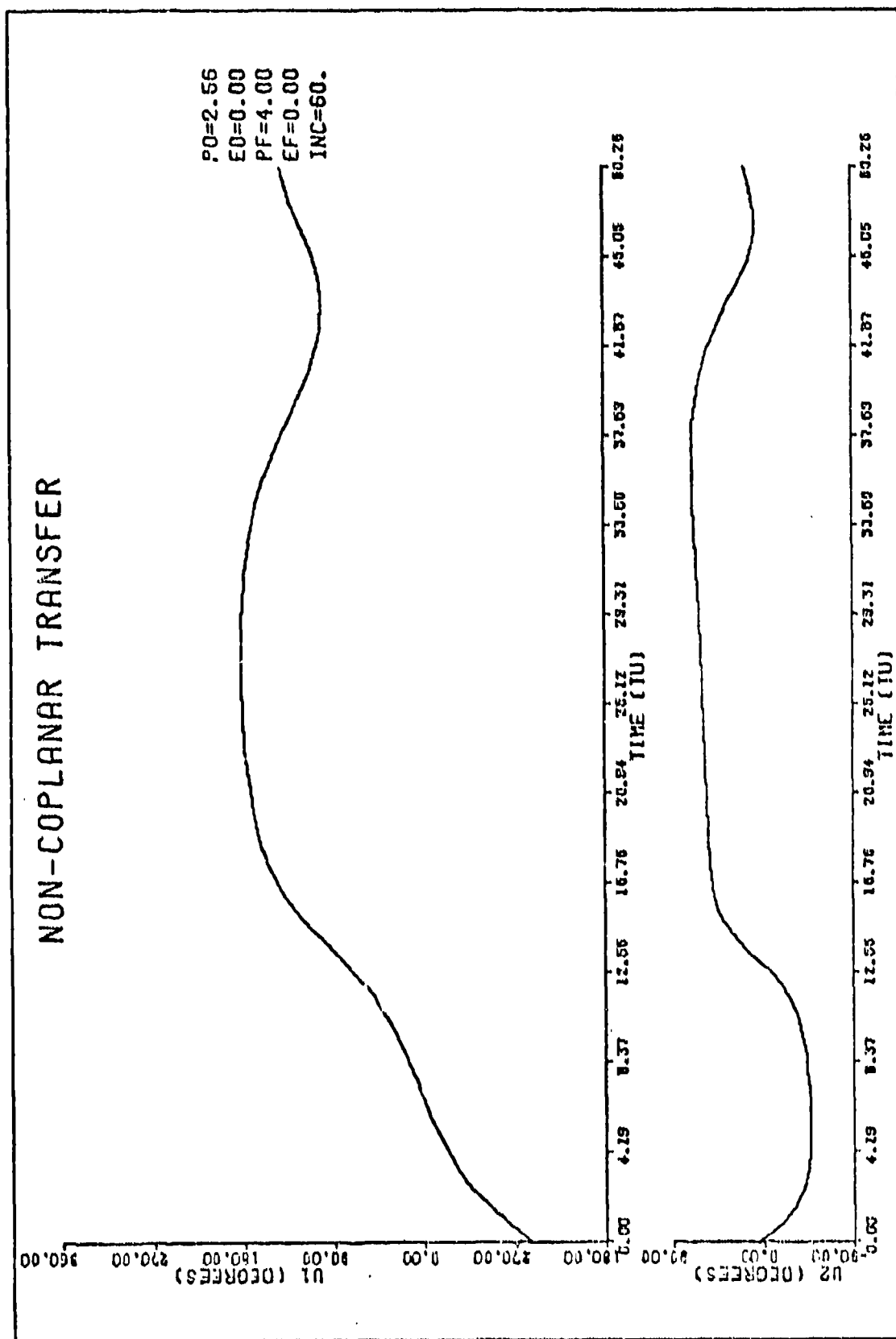
Fig. 17. Control History for Non-Coplanar Transfer Between Inclined Circular Orbits(I = 60°)

V. Conclusions

The problem of minimum time transfer to some desired terminal orbit using a constant magnitude, low thrust propulsion system has been investigated.

For the case of a coplanar transfer, the application of a combination gradient-neighboring extremal algorithm was shown to be a most effective means of obtaining numerical solutions to the resulting non-linear TPBVP. The particular appeal of this method was that no knowledge of the solution was required a priori to obtain extremely accurate results. Thus, a combination gradient-neighboring extremal algorithm would seem to be applicable to a wide class of orbital transfer problems.

On the other hand, for greater efficiency in computer time, it is possible to bypass the gradient portion of this method once an optimal solution is obtained for transfer between specified orbits. This is accomplished by using the above solution as a direct input to the neighboring extremal algorithm when making some small variation, such as increasing eccentricity, in the given orbital parameters.

With this approach, the coplanar analysis was instrumental in extending the problem to non-coplanar terminal orbits. Upon achieving solutions to the non-coplanar problem for small values of inclination, a method of using these initial results was developed. It consisted of generating new estimates to the neighboring extremal algorithm by a polynomial approximation of parameters as a function of inclination. In this manner, solutions to the non-coplanar problem were achieved between orbits of widely varying inclination.

In view of the accuracy of these results, this approach forms the basis of an effective numerical technique for completely general optimal low thrust orbital transfer.

Bibliography

1. Bryson, A.E. and Y.-C. Ho. Applied Optimal Control. Waltham MA: Ginn and Company, 1969.

2. Faulders, C.R. "Minimum-Time Steering Programs for Orbital Transfer with Low Thrust Rockets." Astronaut. Acta, 7:35-49 (1961).

3. Miele, A., et al. "On the Method of Multipliers for Mathematical Programming Problems." Aero-Astronautics Report No. 99. Rice University: 1972.

4. Sage, A.P. Optimal Systems Control. Englewood Cliffs NJ: Prentice-Hall, Inc., 1968.

5. Starr, J.L. and R.D. Sugar. "Some Computational Aspects of Minimum Fuel Continuous Low Thrust Orbit Transfer Problems." The Journal of the Astronautical Sciences, 19:169-204 (November-December 1971).

Appendix A

Derivation of Equations for Coplanar Transfer

State Equations

Using the standard vector notation of $\bar{e}_r$ and $\bar{e}_\theta$ to denote unit vectors in the r-θ plane, the position vector of the vehicle at any time may be written as

$$\bar{r} = r\bar{e}_r \tag{A-1}$$

with the angular velocity of the coordinate system given by

$$\bar{\omega} = \dot{\theta}\bar{e}_n \tag{A-2}$$

where $\bar{e}_n$ is a unit vector normal to the r-θ plane. Differentiating Eq (A-1) twice yields

$$\dot{\bar{r}} = \dot{r}\bar{e}_r + r\dot{\theta}\bar{e}_\theta \tag{A-3}$$

$$\ddot{\bar{r}} = (\ddot{r} - r\dot{\theta}^2)\bar{e}_r + (2\dot{r}\dot{\theta} + r\ddot{\theta})\bar{e}_\theta \tag{A-4}$$

as vehicle velocity and acceleration, respectively.

The restricted two-body equation with terms to account for vehicle thrust has the form

$$\ddot{\bar{r}} = \left(\frac{-\mu}{r^2} + \frac{T}{m_0 - \dot{m}t}\sin u\right)\bar{e}_r + \frac{T}{m_0 - \dot{m}t}\cos u\,\bar{e}_\theta \tag{A-5}$$

where u is measured clockwise from the $\bar{e}_\theta$ direction.

Now let

$$\dot{r} = Vr \tag{A-6}$$

$$\dot{\theta} = \frac{V\theta}{r} \qquad (A-7)$$

and equating vector components on the right hand side of Eqs (A-4) and (A-5)

$$\dot{V}r = \frac{V\theta^2}{r} - \frac{\mu}{r^2} + \frac{T}{m_0 - \dot{m}t} \sin u \qquad (A-8)$$

$$\dot{V}\theta = \frac{-VrV\theta}{r} + \frac{T}{m_0 - \dot{m}t} \cos u \qquad (A-9)$$

which are the four state equations.

Boundary Conditions

The general conic solution to the restricted two-body equation has the form

$$r = \frac{p}{1 + e \cos \theta} \qquad (A-10)$$

Differentiating Eq (A-10) yields

$$Vr = \frac{eh \sin \theta}{p} \qquad (A-11)$$

where h is conserved specific angular momentum such that

$$h = r^2 \dot{\theta} \qquad (A-12)$$

and applying Eq (A-7) in Eq (A-12) results in

$$V\theta = \frac{h}{r} \qquad (A-13)$$

Thus, Eqs (A-10), (A-11), and (A-13) are the three conditions which must be satisfied in both the initial and terminal orbits.

<center>Appendix B</center>

<center>Derivation of Equations for Non-Coplanar Transfer</center>

State Equations

The standard vector notation of $\bar{e}_r$, $\bar{e}_\theta$, and $\bar{e}_\phi$ is again used to denote unit vectors in the r-θ-ϕ spherical coordinate system, which has angular velocity

$$\bar{\omega} = \dot{\theta} \cos\phi\, \bar{e}_r + \dot{\phi}\, \bar{e}_\theta - \dot{\theta} \sin\phi\, \bar{e}_\phi \tag{B-1}$$

Differentiating the position vector, $r\bar{e}_r$, gives

$$\dot{\bar{r}} = \dot{r}\bar{e}_r + r\dot{\theta} \sin\phi\, \bar{e}_\theta + r\dot{\phi}\, \bar{e}_\phi \tag{B-2}$$

as vehicle velocity and

$$\ddot{\bar{r}} = (\ddot{r} - r\dot{\phi}^2 - r\dot{\theta}^2 \sin^2\phi)\, \bar{e}_r$$

$$+ (r\ddot{\theta} \sin\phi + 2\dot{r}\dot{\theta} \sin\phi + 2r\dot{\phi}\dot{\theta} \cos\phi)\, \bar{e}_\theta$$

$$+ (r\ddot{\phi} + 2\dot{r}\dot{\phi} - r\dot{\theta}^2 \sin\phi \cos\phi)\, \bar{e}_\phi \tag{B-3}$$

as vehicle acceleration.

The restricted two-body equation with additional terms for vehicle thrust becomes

$$\ddot{\bar{r}} = \left(\frac{-\mu}{r^2} + \frac{T\ell_1}{m_0 - \dot{m}t} \right) \bar{e}_r$$

$$+ \frac{T\ell_2}{m_0 - \dot{m}t}\, \bar{e}_\theta + \frac{T\ell_3}{m_0 - \dot{m}t}\, \bar{e}_\phi \tag{B-4}$$

where ℓ_1, ℓ_2, and ℓ_3 are the direction cosines between the thrust vector and $\bar{e}_r$, $\bar{e}_\theta$, and $\bar{e}_\phi$, respectively.

In addition to

$$\dot{r} = Vr \tag{B-5}$$

$$\dot{\theta} = \frac{V\theta}{r} \tag{B-6}$$

let

$$\dot{\phi} = \frac{V\phi}{r} \tag{B-7}$$

and again equating acceleration terms in Eqs (B-3) and (D-4)

$$\dot{V}r = \frac{V\phi^2}{r} + \frac{V\theta^2}{r}\sin^2\phi - \frac{\mu}{r^2} + \frac{T\ell_1}{m_o - \dot{m}t} \tag{B-8}$$

$$\dot{V}\theta = \frac{-VrV\theta}{r} - \frac{2V\theta V\phi \cot\phi}{r} + \frac{T\ell_2}{(m_o - \dot{m}t)\sin\phi} \tag{B-9}$$

$$\dot{V}\phi = \frac{-VrV\theta}{r} + \frac{V\theta^2 \sin\phi \cos\phi}{r} + \frac{T\ell_3}{m_o - \dot{m}t} \tag{B-10}$$

which are the six state equations.

Initial Conditions

Three of the five unspecified initial conditions are the same as those derived in Appendix A, Eqs (A-10), (A-11), and (A-13). ϕ of the initial orbit is fixed at 90° in view of the definition of the coordinate system in Chapter III. This in turn sets the value of $\dot{V}\phi$ at 0 everywhere in the initial orbit.

Terminal Conditions

The establishment of terminal conditions for an inclined elliptical orbit requires the use of spherical trigonometric relationships to solve for the angles ϕ^- and θ^- as shown in Fig. 18.

From Napier's rule for right spherical triangles
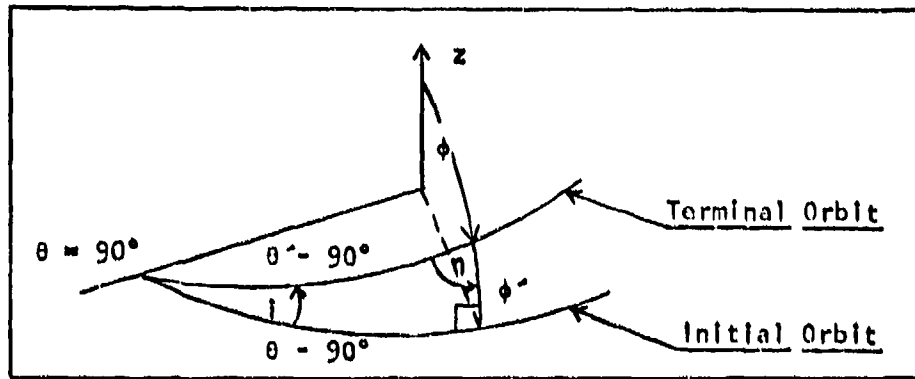
Fig. 18. Lateral View of Inclined Orbit

$$\sin \eta = \frac{\sin i \cos \theta}{\sin \phi^{\prime}} = \frac{\cos i}{\cos \phi^{\prime}} \tag{B-11}$$

or

$$\phi^{\prime} = \arctan (\tan i \cos \theta) \tag{B-12}$$

Similarly, since

$$\cos \theta^{\prime} = \frac{\sin \phi^{\prime}}{\sin i} \tag{B-13}$$

$$\theta^{\prime} = \arctan \left\{ \frac{\sin [\arctan (\tan i \cos \theta)]}{\sin i} \right\} \tag{B-14}$$

Determination of $\theta^{\prime}$ thus allows use of the general conic solution in the terminal orbit where

$$r = \frac{p}{1 + e \cos \theta^{\prime}} \tag{B-15}$$

$$Vr = \frac{eh \sin \theta^{\prime}}{p} \tag{B-16}$$

Furthermore, ϕ may now be determined as

$$\phi = \phi_{0} + \phi^{\prime} = 90^{\circ} + \arctan (\tan i \cos \theta) \tag{B-17}$$

54

The final two conditions are established through the definition of conserved specific angular momentum

$$\bar{h} = \bar{r} \times \dot{\bar{r}} = rV\phi \ \bar{e}_\theta - rV\theta \sin \phi \bar{e}_\phi \tag{B-18}$$

In the terminal orbit, $\bar{h}$ may be written as

$$\bar{h} = h \sin i \ \bar{e}_x + h \cos i \ \bar{e}_z \tag{B-19}$$

where

$$\bar{e}_x = \cos \theta \sin \phi \ \bar{e}_r + \cos \theta \cos \phi \ \bar{e}_\phi - \sin \theta \ \bar{e}_\theta \tag{B-20}$$

$$\bar{e}_z = \cos \phi \ \bar{e}_r - \sin \phi \ \bar{e}_\phi \tag{B-21}$$

After substitution of Eqs (B-20) and (B-21), Eq (B-19) is equated to Eq (B-18) to yield

$$V\theta = \frac{h}{r} (\cos i - \sin i \cos \theta \cot \phi) \tag{B-22}$$

$$V\phi = \frac{-h}{r} \sin i \sin \theta \tag{B-23}$$

Appendix C

The following contains a listing of the computer program, CPTRAN, which was used to obtain results to the coplanar TPBVP of Chapter II. Besides·those subroutines listed, SET, STEP, and MINV were also employed. SET and STEP were called for integration purposes, while MINV was needed to invert the transition matrix of the neighboring extremal algorithm. All three of these subprograms are found on the general package of AFITSUBROUTINES which is on permanent file at the Air Force Institute of Technology.

```
      PROGRAM COTRAN(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=P
     1UNCH)
      REAL L(4),NUO,NU1,NU2,NU3,MF
      DIMENSION X(8),P(8),UO(250),HU(110),TM(8,8),Y(8),V(8,1),Q(8,1)
      EXTERNAL F,G,H
      COMMON/BLKA/TH,MF,I,U(250),X1(110),Y2(110),X3(110),X4(110)
      DO 1 J=1,250
      U(J)=1.
1     UO(J)=1.
      ITERS=0 $ N=100 $ M=100 $ B1=0. $ B2=0. $ B3=0. $ PI=3.1415926536
      READ(5,2) TH,MF
2     FORMAT(2F10.8)
      READ(5,5) S11,S22,S33,EP,C
5     FORMAT(5F10.8)
      WRITE(6,5) S11,S22,S33,EP
6     FORMAT(*0FOR THIS RUN S11=*,F10.5,* S22=*,F10.5,* S33=*,F10.5,* EP
     1SILON=*,F10.5)
      READ(5,9) PF,EF,HF,DW,FAC,TOLS
9     FORMAT(6F10.6)
      READ(5,10) T,DT,P0,E0,H0,NUO,K
10    FORMAT(6F10.8,I2)
11    FORMAT(*-*,6X,*T*,11X,*R*,14X,*NU*,14X,*VR*,12X,*VNU*,12X,*U*)
C
C     INITIAL VALUES OF STATE STORED IN FIRST LOCATION OF ARRAY
C
      X(1)=P0/(1.+E0*COS(NUO)) $ X(2)=NUO
      X(3)=E0+H0*SIN(NUO)/P0 $ X(4)=H0/X(1) $ REFJ=10.**10
      X1(1)=X(1) $ X2(1)=X(2) $ X3(1)=X(3) $ X4(1)=X(4)
12    FORMAT(*0THIS IS THE*,I5,*TH ITERATION*,5X,*B1=*,F15.5,5X,*B2=*,F1
     15.5,5X,*B3=*,F15.5)
```

```
13      ITERS=ITERS+1
        AUGJ=3. $ DJ=0. $ DTMIN=DT/4.
        WRITE(6,12) ITERS,R1,R2,B3
        DO 14 I=1,M
14      U(I)=U0(I)
20      FORMAT (F10.3,5F15.8)

C
C       THIS FINDS TF SUCH THAT DJ/DTF=0.
C
21      I=T/DT+1.1 $ DJ0=DJ
22      CALL SET(4,T,X,DT,F,.0001,.FALSE.,DT,DTMIN)
        CALL STEP(4,T,X,DT,F,.0001,.FALSE.,DT,DTMIN)
        PSI1=X(1)-PF/(1.+EF*COS(X(2)-DW))
        PSI2=X(3)--EF*HF*SIN(X(2)-DW)/PF $ PSI3=X(4)-HF/X(1)
71      =S11*PSI1+S33*PSI3*HF/X(1)**2+R1+B3*HF/X(1)**2
        Z2  =S11*PSI1*(-PF)*EF*SIN(X(2)-DW)/(1.+EF*COS(X(2)-DW))**2 +S22*
     1PSI2*(-EF)*HF*COS(X(2)-DW)/PF-S1*PF*EF*SIN(X(2)-DW)/(1.+EF*COS(X:2
     2)-DW))**2-S2*EF*HF*COS(X(2)-DW)/PF
        Z3  =S22*PSI2+32 $ Z4   =S33*PSI3+B3
        IF(DT.LT.0.) GO TO 24
        I=I+1
        CALL F(4,T,X,P)
        DJ=1.+P(1)*Z1+P(2)*Z2+P(3)*Z3+P(4)*Z4
        IF(DJ0.LT.0..AND.DJ.GE.0.) GO TO 23
        GO TO 21
23      DT=-DT/2.     $   DTMIN=DT/4.
        GO TO 22
24      UX=U(I) $ U(I)=U(I-1)+(UX-U(I-1))/2.
        CALL F(4,T,X,P)
        DJ=1+P(1)*Z1+P(2)*Z2+P(3)*Z3+P(4)*Z4
```

```
      G1=T+DT $ G2=T $ G3=T-DT $ U(I)=UX
      CALL CURVFIT(G1,G2,G3,GJ0,GJ1,GJ,T)

C
C     THIS INTEGRATES THE STATE EQTNS FORWARD
C
      X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1)
      DT=T/100. $ T=0. $ DTL=-DT $ DTMIN=DT/4. $ DILMIN=-DTMIN
      DO 25 I=1,N
      CALL SET(4,T,X,DT,F,.0001,.FALSE.,DT,DTMIN)
      CALL STEP(4,T,X,DT,F,.0001,.FALSE.,DT,DTMIN)
      X1(I+1)=X(1) $ X2(I+1)=X(2) $ X3(I+1)=X(3) $ X4(I+1)=X(4)
25    CONTINUE
      PSI1=X(1)-PF/(1.+EF*COS(X(2)-DW))
      PSI2=X(3)-F*HF*SIN(X(2)-DW)/PF $ PSI3=X(4)-HF/X(1)
      Z1  =S11+PSI1+S31*PSI3*HF/X(1)**2+B1+B3*HF/X(1)**2
      Z2  =S11+PSI1*(-PF)*EF*SIN(X(2)-DW)/(1.+EF*COS(X(2)-DW))**2 +S22*
     1PSI2*(-EF)*HF*COS(X(2)-DW)/PF-B1*PF*EF*SIN(X(2)-DW)/(1.+EF*COS(X(2
     2)-DW))**2-B2*EF*HF*COS(X(2)-DW)/PF
      Z3  =S22*PSI2+B2 $ Z4  =S33*PSI3+B3 $ I=M
      CALL F(4,T,X,P)
      AUGJ=1.+.5*(S11*PSI1**2+S22*PSI2**2+S33*PSI3**2)+B1*PSI1+B2*PSI2+B3
     1*PSI3
      GJ=1.+P(1)*Z1+P(2)*Z2+P(3)*Z3+P(4)*Z4
      DIFF=GJFJ-AUGJ
      IF(DIFF.LT..1) GO TO 95
46    AUGJ0=AUGJ $ AUGJ1=AUGJ
      WRITE(6,45) T,AUGJ,PSI1,PSI2,PSI3,GJ
45    FORMAT(* FINAL T=*,F10.5,* J=*,F15.10,* PSI1=*,F10.6, * PSI2=*,
     1F10.6, * PSI3=*,F10.6,* GJ=*,F10.6)
C
```

```
C     THIS INTEGRATES THE COSTATE EQTNS BACKWARD FROM FINAL TIME
C
      L(1)=S11+PSI1+S33*PSI3*HF/X(1)**2+S31+S33*HF/X(1)**2
      L(2)=S11+PSI1+PSI1*(-PF)+F*SIN(X(2)-DW)/(1.+EF*COS(X(2)-DW))**2 +S22*
     1PSI2*(-EF)*HF*COS(X(2)-DW)/PF-S1*PF*EF*SIN(X(2)-DW)/(1.+EF*COS(X(2
     2)-DW))**2-S2*EF*HF*COS(X(2)-DW)/PF
      HU(M)=EFFTH*(L(3)*COS(U(M))-L(4)*SIN(U(M)))
      L(3)=S22*PSI2+R2 $ L(4)=S33*PSI3+B3 $ EFFTH=TH/(1.-MF*T)
      FH=L(1)*X3(M)+L(2)*X2(M)/X1(M)+L(3)*(X4(M)**2/X1(M)-1./X1(M)**2+EF
     1FTH*SIN(U(M)))+L(4)*(-X3(M)*X4(M)/X1(M)+EFFTH*COS(U(M)))
      WRITE(6,49) T,FH,L(1),L(2),L(3),L(4)
   49 FORMAT(* AT T=*,F10.3,5X,*H=*,F15.10,4F20.10)
      IF(ITERS.GT.50) GO TO 200
      DO 50 J=1,N
      I=N-J
      CALL SFT(4,T,L,DTL,G,.0001,.FALSE.,DTL,DTLMIN)
      CALL STEP(4,T,L,DTL,G,.0001,.FALSE.,DTL,DTLMIN)
      EFFTH=TH/(1.-MF*T)
   50 HU(I)=EFFTH*(L(3)*COS(U(I))-L(4)*SIN(U(I)))
   52 ITER=0 $ KOUNT=0 $ SUM=0.
   55 DO 56 J=1,M
   56 SUM=SUM+ABS(HU(J))
      SUM=SUM/M
      WRITE(6,135) SUM
      IF(SUM.LT..01) GO TO 90
   61 X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1) $ T=0.
      U(1)=UO(1)-C*HU(1)
C
C     THIS FINDS BEST VALUE OF C TO MINIMIZE J
C
```

60

```
      DO 65 I=1,4
      CALL SET(4,T,X,DT,F,.9061,.FALSE.,DT,DTMIN)
      CALL STEP(4,T,X,DT,F,.3001,.FALSE.,DT,DTMIN)
      U(I+1)=UO(I+1)-C*HU(I+1)
65    AUGJ2=T+.5*(S1I*(X(1)-PF/(1.+EF*COS(X(2)-DW)))**2+S22*(X(3)-EF*HF*
     1SIN(X(2)-DW)/PF)**2+S33*(X(4)-HF/X(1))**2)+B1*(X(1)-PF/(1.+EF*COS(
     1X(2)-DW))+B2*(X(3)-EF*HF*SIN(X(2)-DW)/PF)+B3*(X(4)-HF/X(1))
      IF(ITER.GT.0) GO TO 75
      IF(AUGJ2.GT.AUGJ1.AND.KOUNT.EQ.0) GO TO 71
      IF(AUGJ2.GT.AUGJ1.AND.KOUNT.EQ.1) GO TO 70
      IF(AUGJ2.GT.AUGJ1.AND.KOUNT.GT.0) GO TO 80
      C=2.*C $ AUGJO=AUGJ1 $ KOUNT=KOUNT+1 $ AUGJ1=AUGJ2
      GO TO 61
70    C=.5*C
71    C=.5*C $ ITER=1
      GO TO 61
75    IF(AUGJ2.GT.AUGJO.AND.ITER.EQ.10.AND.KOUNT.EQ.0) GO TO 90
      IF(AUGJ2.GT.AUGJO.AND.KOUNT.EQ.1.AND.ITER.EQ.1) GO TO 79
      IF(AUGJ2.GT.AUGJO) GO TO 76
      ITER=11 $ C=.5*C $ AUGJO=AUGJ2
      GO TO 61
76    IF(ITER.GT.10) GO TO 79
      ITER=ITER+1 $ C=.5*C
      GO TO 61
79    C=2.*C $ REFJ=AUGJ $ AUGJ=0. $ DJ=0.
      WRITE(6,110) C,ITERS
      X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1) $ T=0.
      GO TO 83
80    REFJ=AUGJ $ AUGJ=0. $ DJ=0.
      C=(3.*C/3.)*(-4.*AUGJO+5.*AUGJ1-AUGJ2)/(-2.*AUGJO+3.*AUGJ1-AUGJ2)
```

61

```
      WRITE(6,110) C,ITERS
      X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1) $ T=0.
85    DO 85 J=1,M
85    UO(J)=UO(J)-C*HU(J)
      GO TO 13
90    WRITE(6,91)
91    FORMAT(*MIN J HAS BEEN REACHED FOR THE ABOVE VALUES OF B*)
      B1=EP*SI1+B1 $ B2=EP*S22*PSI2+B2 $ B3=EP*S33*PSI3+B3
98    X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1) $ T=0.
      REFJ=1.**10
      GO TO 13
115   FORMAT(* THE AVERAGE VALUE OF THE GRADIENT=*,F15.10)
116   FORMAT(* C=*,F16.6,* GIVES THE MIN J FOR THE*,I5,*TH ITERATION*)
C
C     THIS BEGINS THE NEIGHBORING EXTREMAL ALGORITHM
C
250   HU2=L(3)/ARS(FH) $ HU3=L(4)/ARS(FH) $ TF=T
      NU1=L(2)/ARS(FH)-NU3*HF/X(1)**2
      X1I=X1(1) $ X2I=X2(1) . X3I=X3(1) $ X4I=X4(1)
      WRITE(6,205) X(1),X(2),X(3),X(4),NU1,NU2,NU3,TF
205   FORMAT(*0THE INITIAL ESTIMATES FOR THE NEIGHBORING EXTREMAL ALGORI
     1TH*,AGE*,/,8F15.3)
216   X1F=X(1) $ X2F=X(2) $ X3F=X(3) $ X4F=X(4)
      PSI1=X(1)-PF/(1.+(F*COS(X(2)-DH))
      PSI2=X(3)-:F*HF*SIN(X(2)-DH)/PF $ PSI3=X(4)-HF/X(1)
      X(5)=NU1+NU3*HF/X(1)**2
      X(6)=-PF*EF*SIN(X(2)-DH)*NU1/(1.+EF*COS(X(2)-DH))**2-EF*HF*COS(X(2
     1)-DH)*NU3
      X(7)=NU2 $ X(3)=NU3
      OMEG=X(5)*X(3)+X(6)*X(4)/X(1)+X(7)*(X(4)**2/X(1)-1./X(1)**2-(TH*
```

62

```
      1X(7))/((1.-MF*TF)*SQRT(X(7)**2+X(8)**2)))+X(8)+(-X(3)*X(4)/X(1)-
      2(TH*X(8))/((1.-MF*TF)*SQRT(X(7)**2+X(8)**2)))+1.
      DT=-TF/100. $ T=TF $ DTMIN=DT/2.
      DO 220 J=1,100
C
C     THIS INTEGRATES BACKWARD IN TIME TO THE INITIAL CONDITIONS
C
      CALL SET(K,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
220   CALL STEP(K,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
      X10=X(1) $ X20=X(2) $ X30=X(3) $ X40=X(4)
      WRITE(6,225) X10,X20,X30,X40
225   FORMAT(*3X10=*,F12.8,5X,*X20=*,F12.8,5X,*X30=*,F12.8,5X,*X40=*,F12
     1.8)
      ERR1=X10-X1I $ ERR2=X20-X2I $ ERR3=X30-X3I $ ERR4=X40-X4I
      WRITE(6,225) ERR1,ERR2,ERR3,ERR4
226   FORMAT(* ERR1=*,F12.8,5X,*ERR2=*,F12.8,5X,*ERR3=*,F12.8,5X,*ERR4=*
     1,F12.8)
      WRITE(6,227) PSI1,PSI2,PSI3,OMEG
227   FORMAT(* PSI1=*,F12.8,5X,*PSI2=*,F12.8,5X,*PSI3=*,F12.8,5X,*OMEGA=
     1*,F12.8)
      IF(ABS(ERR1).LT.TOLS.AND.ABS(ERR2).LT.TOLS.AND.ABS(ERR3).LT.TOLS.A
     1ND.ABS(ERR4).LT.TOLS.AND.ABS(PSI1).LT.TOLS.AND.ABS(PSI2).LT.TOLS.A
     2ND.ABS(PSI3).LT.TOLS.AND.ABS(OMEG).LT.TOLS) GO TO 300
C
C     THIS DETERMINES THE TRANSITION MATRIX
C
      DO 250 I=1,6
      DO 235 J=1,8
      Y(J)=0.
235   Y(I)=FAC
```

63

```
      WRITE(6,110) C,ITERS
      X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1) $ T=0.
      DO 85 J=1,M
      UO(J)=UO(J)-C*HU(J)
      GO TO 13
      WRITE(5,91)
   91 FORMAT(*0MIN J HAS BEEN REACHED FOR THE ABOVE VALUES OF B*)
      B1=EP*S11+PSI1 $ B2=EP*S22+PSI2+B2 $ B3=EP*S33*PSI3+B3
   98 X(1)=X1(1) $ X(2)=X2(1) $ X(3)=X3(1) $ X(4)=X4(1) $ T=0.
      REFJ=10.**10
      GO TO 13
  115 FORMAT(* THE AVERAGE VALUE OF THE GRADIENT=*,F15.10)
  110 FORMAT(* C=*,F16.6,* GIVES THE MIN J FOR THE*,I5,*TH ITERATION*)
C
C     THIS BEGINS THE NEIGHBORING EXTREMAL ALGORITHM
C
  200 NU2=L(3)/A2S(FH) $ NU3=L(4)/A3S(FH) $ TF=T
      NU1=L(1)/A1S(FH)-NU3*HF/X(1)**2
      X1I=X1(1) $ X2I=X2(1) . X3I=X3(1) $ X4I=X4(1)
      WRITE(6,205) X(1),X(2),X(3),X(4),NU1,NU2,NU3,TF
  205 FORMAT(*0THE INITIAL ESTIMATES FOR THE NEIGHBORING EXTREMAL ALGORI
     1THM ARE*,/,8F15.8)
  216 X1F=X(1) $ X2F=X(2) $ X3F=X(3) $ X4F=X(4)
      PSI1=X(1)-PF/(1.+EF*COS(X(2)-DW))
      PSI2=X(3)-.F*HF*SIN(X(2)-DW)/PF $ PSI3=X(4)-HF/X(1)
      X(5)=NU1+NU3*HF/X(1)**2
      X(6)=-PF+EF*SIN(X(2)-DW)*NU1/(1.+EF*COS(X(2)-DW))**2-EF*HF*COS(X(2
     1)-DW)*NU2/PF
      X(7)=NU2 $ X(8)=NU3
      OMEG=X(5)*X(3)+X(6)*X(4)/X(1)+X(7)*(X(4)**2/X(1)-1./X(1)**2-(TH*
```

```
      X(1)=X1F+Y(1)  $  X(2)=X2F+Y(2)  $ X(3)=X3F+Y(3)  $ X(4)=X4F+Y(4)
      NU1=NU1+Y(5)  $ NU2=NU2+Y(6)  $ NU3=NU3+Y(7)  $ TF=TF+Y(8)
      SI1=X(1)-PF/(1.+EF*COS(X(2)-DW))
      SI2=Y(3)-EF*HF*SIN(X(2)-DW)/PF  $  SI3=X(4)-HF/X(1)
      X(5)=NU1+NU3*HF/X(1)**2
      X(6)=-PF+EF*SIN(X(2)-DW)*NU1/(1.+EF*COS(X(2)-DW))**2-EF*HF*COS(X(2
     1)-DW)*NU2/PF
      X(7)=NU2  $ X(8)=NU3
      OMEGA=X(5)*X(3)+X(6)*X(4)/X(1)+X(7)*(X(4)**2/X(1)-1./X(1)**2-(TH*
     1X(7))/((1.-HF*TF)*SQRT(X(7)**2+X(8)**2))+X(8)*(-X(3)*X(4)/X(1)-
     2(TH*X(5))/((1.-HF*TF)*SQRT(X(7)**2+X(8)**2)))+1.
      DT=-TF/120.  $ T=TF  $ DTMIN=DT/2.
      DO 240 M=1,10)
      CALL SET(K,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
240   CALL STEP(K,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
      TM(1,I)=(X(1)-X1J)/FAC
      TM(2,I)=(X(2)-X2J)/FAC
      TM(3,I)=(X(3)-X3J)/FAC
      TM(4,I)=(X(4)-X4J)/FAC
      TM(5,I)=(SI1-PSI1)/FAC
      TM(6,I)=(SI2-PSI2)/FAC
      TM(7,I)=(SI3-PSI3)/FAC
      TM(8,I)=(OMEGA-OMEG)/FAC
      X(2)=X1F  $ X(2)=X2F  $ X(3)=X3F  $ X(4)=X4F  $ NU1=NU1-Y(5)
      NU2=NU2-Y(6)    NU3=NU3-Y(7)
250   TF=TF-Y(8)
      CALL MINV(TM,TM,8)
      CUMERR=SQRT(ERR1**2+ERR2**2+ERR3**2+ERR4**2+PSI1**2+PSI2**2+PSI3**
     12+OMEG**2)/3.
      EPS=.005/CUMERR
```

```
        IF(EPS.LT.1.) GO TO 251
        EPS=1.
251     V(1,1)=-EPS*ERR1
        V(2,1)=-EPS*ERR2
        V(3,1)=-EPS*ERR3
        V(4,1)=-EPS*ERR4
        V(5,1)=-EPS*PSI1
        V(6,1)=-EPS*PSI2
        V(7,1)=-EPS*PSI3
        V(8,1)=-EPS*OMEG
253     FORMAT(3F15.8)
        WRITE(5,254)
254     FORMAT(*80*LX(TG), DFSI, AND DOMEGA ARE*)
        WRITE(5,253) (V(I,1),I=1,8)
        CALL MATMUL(TM,V,Q,8,8,1)
        WRITE(6,255)
255     FORMAT(*90*LX(IF), DNU, AND DTF ARE*)
        WRITE(6,253) (Q(I,1),I=1,8)
C
C       THIS COMPUTES THE DESIRED CHANGES IN X,NU, AND TF IN ORDER TO COME
C       CLOSER TO MEETING THE INITIAL CONDITIONS AND CONSTRAINTS
C
        CALL H(K,TF,X,P)
        DO 260 I=1,4
250     X(I)=X(I)+Q(I,1)+Q(8,1)*P(I)
        NU1=NU1+Q(5,1)
        NU2=NU2+Q(6,1)
        NU3=NU3+Q(7,1)
        TF=TF+Q(8,1)
        WRITE(6,270) X(1),X(2),X(3),X(4),NU1,NU2,NU3,TF
```

66

```
270   FORMAT(*1THE NEW ESTIMATES ARE *,/,8F15.8)
      GO TO 215
350   T=0. : DT=TF/100. $ DTMIN=DT/2.
      WRITE(6,310)
310   FORMAT(6X,*T*,11X,*R*,14X,*RU*,14X,*VR*,12X,*VNU*,13X,*U*,8X,*U IN
     1 DEGREES*)
315   FORMAT(F10.3,6F15.6)
      DO 490 I=1,101
      OPTU=ACOS(-X(8)/SQRT(X(7)**2+X(8)**2))
      IF(X(7).GT.0..AND.X(8).GT.0..OR.X(7).GT.0..AND.X(8).LT.0.) GO TO
     1350
      GO TO 331
350   OPTU=2.*PI-OPTU
320   OPTUDEG=180.*OPTU/PI
      WRITE(6,315) T,X(1),X(2),X(3),X(4),OPTU,OPTUDEG
      WRITE(7,390) OPTU
390   FORMAT(F20.10)
      CALL SET(K,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
490   CALL STEP(K,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
      END

      SUBROUTINE F(K,T,X,P)

C     THIS SUBROUTINE COMPUTES THE FIRST TIME DERIVATIVE OF THE STATE
C     VECTOR X
C
      REAL MF
```

```
      DIMENSION X(4),P(4)
      COMMON/BLKA/TH,MF,I,U(250)
      EFFTH=TH/(1.-MF*T)
      P(1)=X(3)
      P(2)=X(4)/X(1)
      P(3)=(X(4)**2/X(1))-(1./X(1)**2)+EFFTH*SIN(U(I))
      P(4)=-X(3)*X(4)/X(1)+EFFTH*COS(U(I))
      RETURN
      END


      SUBROUTINE G(K,T,L,DL)

C     THIS SUBROUTINE COMPUTES THE FIRST TIME DERIVATIVE OF THE COSTATE
C     VECTOR LAMBDA
C
      REAL L(4),DL(4),MF
      COMMON/BLKA/TH,MF,I,U(250),X1(110),X2(110),X3(110),X4(110)
      DL(1)=(X4(I)*L(2)+(X4(I)**2-2./X1(I))*L(3)-X3(I)*X4(I)*L(4))/X1(I)
     1**2
      DL(2)=0.
      DL(3)=-L(1)+X4(I)*L(4)/X1(I)
      DL(4)=(-L(2)-2.*X4(I)*L(3)+X3(I)*L(4))/X1(I)
      RETURN
      END
```

```
      SUBROUTINE H(K,T,X,P)

C     THIS SUBROUTINE COMPUTES THE FIRST TIME DERIVATIVE OF THE COMBINED
C     STATE-COSTATE VECTOR
C
      DIMENSION X(8),P(8)
      REAL MF
      COMMON/BLK6/TH,MF
      P(1)=X(3)
      P(2)=X(4)/X(1)
      P(3)=X(4)**2/X(1)-1./X(1)**2- (TH*X(7))/((1.-MF*T)*SQRT(X(7)**2+
     1X(8)**2))
      P(4)=-X(3)*X(4)/X(1)-(TH*X(8))/((1.-MF*T)*SQRT(X(7)**2+X(8)**2))
      P(5)= X(4)*X(6)/X(1)+X(7)*((X(4)/X(1))**2-2./X(1)**3)-X(3)*X(4)
     2*X(8)/X(1)**2
      P(6)=0.
      P(7)=-X(5)+X(4)*X(8)/X(1)
      P(8)=(-X(6)-2.*X(4)*X(7)+X(3)*X(8))/X(1)
      RETURN
      END


      SUBROUTINE CURVFIT(X1,X2,X3,Y1,Y2,Y3,XMIN)

C     THIS SUBROUTINE IS USED TO FIND THE TIME AT WHICH DJ/DTF=C
C
      A= Y1*(X2-X3)-X1*(Y2-Y3)+Y2*X3-X2*Y3
      B=X1**2*(Y2-Y3)-Y1*(X2**2-X3**2)+(X2**2)*Y3-(X3**2)*Y2
```

```
      C=X1**2*(X2*Y3-X3*Y2)-X1*(X2**2*Y3-X3**2*Y2)+Y1*(X2**2*X3-X3**2*X2
     1)
      S1=(-B+SQRT(B**2-4.*A*C))/(2.*A)
      S2=(-B-SQRT(B**2-4.*A*C))/(2.*A)
      IF(S1.LT.X3.AND.S1.GT.X1) GO TO 1
      XMIN=S2
      GO TO 2
    1 XMIN=S1
    2 CONTINUE
      RETURN
      END

C
      SUBROUTINE MATMUL (A,E,C,N,M,K)
C
C     THIS SUBROUTINE IS USED FOR MATRIX MULTIPLICATION
C
      DIMENSION A(N,M),E(M,K),C(N,K)
      DO 25 I=1,N
      DO 30 L=1,K
      SUM=0.
      DO 20 J=1,M
   20 SUM=SUM+A(I,J)*E(J,L)
   30 C(I,L)=SUM
   25 CONTINUE
      RETURN
      END
```

Appendix D

The following contains a listing of the computer program, NCPTRAN, which was used to obtain results to the non-coplanar TPBVP of Chapter III. The AFITSUBROUTINES, SET, STEP, and MINV were again employed in the same manner as explained in Appendix C.

```
      PROGRAM MCPTPAN(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE5=OUTPUT,TAPE7=
     1PUNCH)
      REAL NU1,NU2,NU3,NU4,NU5,NUP,NUO,MF,INC
      DIMENSION X(12),P(12),TM(12,12),Y(12),V(12,12),Q(12,1)
      EXTERNAL H
      COMMON/BLKA/TM,MF
      READ(5,5) (X(I),I=1,6)
    5 FORMAT(6F10.7)
      READ(5,5) NU1,NU2,NU3,NU4,NU5,TF
      READ(5,5) PO,EO,HO,NUO,PHIC,TOLS
      READ(5,5) PF,EF,HF,INC,FAC,EPS
      READ(5,6) TH,MF
    6 FORMAT(2F10.7)
      WRITE(6,7)  (X(I),I=1,6),NU1,NU2,NU3,NU4,NU5,TF
    7 FORMAT(*0THE 1ST ESTIMATES ARE*,/,1X,*R=*,F12.3,5X,*NU=*,F12.8,5X,
     1*PHI=*,F12.8,5X,*VR=*,F12.8,5X,*VNU=*,F12.8,5X,*VPHI=*,F12.8,/,1X,
     2*NU1=*,F12.8,5X,*NU2=*,F12.8,5X,*NU3=*,F12.8,5X,*NU4=*,F12.8,5X,*N
     3U5=*,F12.3,5X,*TF=*,F12.6)
      PI=3.14159265358 $ PI2=2.*PI $ PI3=3.*PI
      X1I=PO/(1.+EO*COS(NUO))  ,  X2I=NUO $ X3I=PHIO
      X4I=EO*HO*SIN(NUO)/PO $ X5I=HO/X1I $ X6I=0.
      X1F=X(1) $ X2F=X(2)    X3F=X(3) $ X4F=X(4) $ X5F=X(5) $ X6F=X(6)
      NUP=Y(2) $ PHIP=ATAN(TAN(INC)*COS(X2F))
      IF(INC.LT..00001) GO TO 15
      NUP=ACOS(SIN(PHIP)/SIN(INC))
      IF(X2F.LT.PI) GO TO 15
      IF(X2F.GT.PI2.AND.X2F.LT.PI3) GO TO 15
      NUP=2.*PI-NUP
      PSI1=X1F-FF/(1.+EF*COS(NUP))
      PSI2=X3F-(PHIO+ATAN(TAN(INC)*COS(X2F)))
```

```
      PSI3=X1F-EF*HF*SIN(NUP)/PF
      PSI4=X5F-(HF/X1F)*(COS(INC)-SIN(INC)*COS(X3F)*COS(X2F)/SIN(X3F))
      PSI5=X6F+HF*SIN(INC)*SIN(X2F)/X1F
      DYDX2=1.
      IF(INC.LT. 0.0001) GO TO 23
      DYDX2=COS(PHIP)*SIN(X2F)*TAN(INC)/(SQRT(SIN(INC)**2-SIN(PHIP)**2)*
     1(1.+(TAN(INC)*COS(X2F))**2))
      Y(7)=NU1+NU4*HF*(COS(INC)-SIN(INC)*COS(X2F)*COS(X3F)*SIN(X3F))/X1F
     1*+2-NU5*HF*SIN(INC)*SIN(X2F)/X1F**2
      X(8)=-NU1*PF*EF*SIN(NUP)+DYDX2/(1.+EF*COS(NUP))**2+NU2*SIN(X2F)*TA
     1N(INC)/(1.+(TAN(INC)*COS(X2F))**2)-NU3*EF*HF*COS(NUP)*DYDX2/PF-NU4
     2*HF*SIN(INC)*SIN(X3F)*COS(X3F)/(SIN(X3F)*X1F)+NU5*HF*SIN(INC)*COS(
     3X2F)/X1F
      X(9)=NU2-NU4*HF*SIN(INC)*COS(X2F)/(X1F*SIN(X3F)**2)
      X(11)=NU3   X(11)=NU4 = X(12)=NU5
      EFTH=IF/(1.-NF*TF)
      RAD=SQRT(X(10)**2+(X(11)/SIN(X3F))**2+X(12)**2)
      OMEG=X(7)*X(4)+X(8)*X(5)/X(1)+X(9)*X(6)/X(1)+X(10)*((X(6)**2*X(1)+
     1X(5)**2*TAN(X(3))**2*X(1)-1.)/X(1)**2-EFTH*X(10)/RAD)+X(11)*((-X(4
     2)*X(5)-2.*X(6)*X(5)*COS(X(3))/SIN(X(3)))/X(1)-EFTH*X(11)/SIN(X(3))
     3)*RAD))+X(12)*((-X(4)*X(6)+X(5)**2*SIN(X(3))*COS(X(3)))/X(1)-EFTH*
     4X(12)/RAD)+1.
      DT=-TF/100.  $ T=TF  $ DTMIN=DT/20.
      DO 24 J=1,100
C
C     THIS TAKES THE INITIAL EXTIMATES FOR XF, NU, AND TF AND INTEGRATES
C     BACKWARD IN TIME TO THE INITIAL CONDITIONS
C
      CALL SET(12,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
   24 CALL STEP(12,T,X,DT,H,.00001,.FALSE.,DT,DTMIN)
```

```
      X1C=X(1) $ X2C=X(2) $ X3C=X(3) $ X4C=X(4) $ X5C=X(5) $ X6C=X(6)
      WRITE(6,25) X1C,X2C,X3C,X4C,X5C,X6C
25    FORMAT(*0X10=*,F12.8,5X,*X20=*,F12.8,5X,*X30=*,F12.8,5X,*X4C=*,F12
     1.8,5X,*X5C=*,F12.8,5X,*X6C=*,F12.8)
      ERR1=X1C-X1I $ ERR2=X2C-X2I $ ERR3=X3C-X3I $ ERR4=X4C-X4I
      ERR5=X5C-X5I $ ERR6=X6C-X6I
      WRITE(6,26) ERR1,ERR2,ERR3,ERR4,ERR5,ERR6
26    FORMAT(* ERR1=*,F12.8,5X,*ERR2=*,F12.8,5X,*ERR3=*,F12.8,5X,*ERR4=*
     1,F12.8,5X,*ERR5=*,F12.8,5X,*ERR6=*,F12.8)
      WRITE(6,27) PSI1,PSI2,PSI3,PSI4,PSI5,OMEG
27    FORMAT(* PSI1=*,F12.8,5X,*PSI2=*,F12.8,5X,*PSI3=*,F12.8,5X,*PSI4=*
     1,F12.8,5X,*PSI5=*,F12.8,5X,*OMEGA=*,F12.8)
      IF(ABS(ERR1).LT.TOLS.AND.ABS(ERR2).LT.TOLS.AND.ABS(ERR3).LT.TOLS.A
     1ND.ABS(ERR4).LT.TOLS.AND.ABS(PSI1).LT.TOLS.AND.ABS(PSI2).LT.TOLS.A
     2ND.ABS(PSI3).LT.TOLS.AND.ABS(PSI4).LT.TOLS.AND.ABS(PSI5).LT.TOLS.A
     3ND.ABS(ERR5).LT.TOLS.AND.ABS(ERR6).LT.TOLS.AND.ABS(OMEG).LT.TOLS)
      460 TO 100
C
C     THIS DETERMINES THE TRANSITION MATRIX
C
      DO 50 I=1,12
      DO 35 J=1,12
      Y(J)=0.
35    Y(I)=FAC
      X(1)=X1F+Y(1)    X(2)=X2F+Y(2)  $ X(3)=X3F+Y(3)  $ X(4)=X4F+Y(4)
      X(5)=X5F+Y(5)  $ X(6)=X6F+Y(6)  $ NU1=NU1+Y(7)  $ NU2=NU2+Y(8)
      NU3=NU3+Y(9)  $ NU4=NU4+Y(10)  $ NU5=NU5+Y(11)  $ TF=TF+Y(12)
      X1N=X(1) $ X2N=X(2) $ X3N=X(3) $ X4N=X(4) $ X5N=X(5) $ X6N=X(6)
      NUP=X2N $ PHIP=ATAN(TAN(INC)*COS(X2N))
      IF(INC.LT..00031) GO TO 36
```

25

26

27

35

```
      NUP=ACOS(SIN(PHIP)/SIN(INC))
      IF(X2N.LT.PI) GO TO 36
      IF(X2N.GT.PI2.AND.X2N.LT.PI3) GO TO 36
      NUP=2.*PI-NUP
36    SI1= X1N-PF/(1.+EF*COS(NUP))
      SI2= X2N-(PHIQ+ATAN(TAN(INC)*COS(X2N)))
      SI3= X4N-EF*HF*SIN(NUP)/PF
      SI4= X5N-(HF/X1N)*(COS(INC)-SIN(INC)*COS(X2N)*COS(X2N)/SIN(X3N))
      SI5= X6N+HF*SIN(INC)*SIN(X2N)/X1N
      DNDX2=1.
      IF(INC.LT..00001) GO TO 37
      DNDX2=COS(PHIP)*SIN(X2N)*TAN(INC)/(SQRT(SIN(INC)**2-SIN(PHIP)**2)*
     1(1.+(TAN(INC)*COS(X2N))**2))
      X(7)=NU1+NU4*HF*(COS(INC)-SIN(INC)*COS(X2N)*COS(X3N)/SIN(X3N))/X1N
     1**2-NU5*HF*SIN(INC)*SIN(X2N)/X1N**2
      X(8)=-NU1*PF*EF*SIN(NUP)*DNDX2/(1.+EF*COS(NUP))**2+NU2*SIN(X2N)*TA
     1N(INC)/(1.+(TAN(INC)*COS(X2N))**2)-NU3*EF*HF*COS(NUP)*DNDX2/PF-NU4
     2*HF*SIN(INC)*SIN(X2N)*COS(X3N)/(SIN(X3N)*X1N)+NU5*HF*SIN(INC)*COS(
     3X2N)/X1N
      X(9)=NU2-NU3*HF*SIN(INC)*COS(X2N)/(X1N*SIN(X3N))
      X(10)=NU3   X(11)=NU4   X(12)=NU5
      EFTH=TH/(1.-MF*TF)
37    RAD=SQRT(X(10)**2+(X(11)/SIN(X3N))**2+X(12)**2)
      OMEN=X(7)*X(4)+Y(8)*X(1)+X(9)*X(1)*X(6)/X(1)+X(10)*((X(6)**2*X(1)+
     1X(5)**2*SIN(X(3))**2*X(1)-1.)/X(1)**2-EFTH*X(10)/RAD)+X(11)*((-X(4
     2)*X(5)-2.*X(6)*X(5)*COS(X(3))/SIN(X(3)))/X(1)-EFTH*X(11)/(SIN(X(3)
     3)*RAD))+X(12)*((-X(4)*X(6)+X(5)**2*SIN(X(3))*COS(X(3)))/X(1)-EFTH*
     4X(12)/RAD)+1.
      DT=-TF/130.   T=TF   DTMIN=DT/20.
      DO 40 M=1,100
```

```
      CALL SET(12,T,X,DT,4,.00001,.FALSE.,DT,DTMIN)
      CALL STEP(12,T,X,DT,4,.00001,.FALSE.,DT,DTMIN)
      TM(1,I)=(X(1)-X13)/FAC
      TM(2,I)=(X(2)-X20)/FAC
      TM(3,I)=(X(3)-X30)/FAC
      TM(4,I)=(X(4)-X40)/FAC
      TM(5,I)=(X(5)-Y50)/FAC
      TM(6,I)=(X(6)-X60)/FAC
      TM(7,I)=(SI1-PSI1)/FAC
      TM(8,I)=(SI2-PSI2)/FAC
      TM(9,I)=(SI3-PSI3)/FAC
      TM(10,I)=(SI4-PSI4)/FAC
      TM(11,I)=(SI5-PSI5)/FAC
      TM(12,I)=(OMEN-OMEG)/FAC
      X(1)=X1F $ X(2)=X2F $ X(3)=X3F $ X(4)=X4F $ X(5)=X5F $ X(6)=X6F
      NU1=NU1-Y(7) $ NU2=NU2-Y(8) $ NU3=NU3-Y(9) $ NU4=NU4-Y(10)
      NU5=NU5-Y(11)
      TF=TF-Y(12)
      WRITE(6,51)
   51 FORMAT(*0THE TRANSITION MATRIX IS*)
      DO 52 I=1,12
      WRITE(6,53) (TM(I,J),J=1,12)
   52 
   53 FORMAT(12F10.3)
      CALL MINV(TM,TM,12)
      WRITE(6,75)
   75 FORMAT(*0THE INVERTED TRANSITION MATRIX IS*)
      DO 80 I=1,12
   80 WRITE(6,53) (TM(I,J),J=1,12)
C
C     THIS COMPUTES THE DESIRED CHANGES IN X, NU, AND TF IN ORDER TO
```

```
C       COME CLOSER TO MEETING THE INITIAL CONDITIONS AND CONSTRAINTS
C
        V(1,1)=-EPS*ERR1
        V(2,1)=-EPS*ERR2
        V(3,1)=-EPS*ERR3
        V(4,1)=-EPS*ERR4
        V(5,1)=-EPS*ERR5
        V(6,1)=-EPS*ERR6
        V(7,1)=-EPS*PSI1
        V(8,1)=-EPS*PSI2
        V(9,1)=-EPS*PSI3
        V(10,1)=-EPS*PSI4
        V(11,1)=-EPS*PSI5
        V(12,1)=-EPS*OMEG
        WRITE(6,54)
54      FORMAT(*1B*,LX(TG), DPSI, AND DOMEGA ARE*)
        WRITE(6,56) (V(I,1),I=1,12)
        CALL MATMUL(TH,V,0,12,12,1)
        WRITE(6,55)
55      FORMAT(*2D-LX(TF), DHU, AND DTF ARE*)
        WRITE(6,56) (Q(I,1),I=1,12)
56      FORMAT(6F20.10,/,6F20.10)
        CALL H(12,TF,X,P)
        DO 50 I=1,6
60      X(I)=X(I)+Q(I,1)+Q(12,1)*P(I)
        NU1=NU1+Q(7,1) $ NU2=NU2+Q(8,1) $ NU3=NU3+Q(9,1)
        NU4=NU4+Q(10,1) $ NU5=NU5+Q(11,1) $ TF=TF+Q(12,1)
        WRITE(6,70) (X(I),I=1,6),NU1,NU2,NU3,NU4,NU5,TF
70      FORMAT(*3THE NEW ESTIMATES ARE*,/,1X,*R=*,F12.8,5X,*NU=*,F12.8,5X,
        1*PHI=*,F12.8,5X,*VR=*,F12.8,5X,*VNU=*,F12.8,5X,*VPHI=*,F12.8,/,1X,
```

77

```
      2*NU1=*,F12.8,5X,*NU2=*,F12.8,5X,*NU3=*,F12.8,5X,*NU4=*,F12.8,5X,*N
      3U5=*,F12.8,5X,*TF=*,F12.8)
      GO TO 10
  100 WRITE(7,101) X1F,X2F,X3F,X4F,X5F,X5F
  101 FORMAT(6F10.7)
      WRITE(7,102) NU1,NU2,NU3,NU4,NU5,TF
  102 FORMAT(6F10.6)
      END


      SUBROUTINE H(K,T,X,P)

C
C     THIS SUBROUTINE COMPUTES THE FIRST TIME DERIVATIVE OF THE COMBINED
C     STATE-COSTATE VECTOR

      DIMENSION X(12),P(12)
      REAL MF
      COMMON/BLK4/TH,MF
      P(1)=X(4) ; RAD=SQRT(X(10)**2+(X(11)/SIN(X(3)))**2+X(12)**2)
      P(2)=X(5)/X(1)  & EFTH=TH/(1.-MF*T)
      P(3)=X(5)/X(1)
      P(4)=(X(1)*X(6)**2+X(1)*(X(5)*SIN(X(3)))**2-1.)/X(1)**2-EFTH*X(10)
      1/RAD
      P(5)=(-X(4)*X(7))-2.*X(5)*X(6)*COS(X(3))/SIN(X(3)))/X(1)-EFTH*X(11)
      1/(SIN(X(3))*RAD)
      P(6)=(-X(4)*X(5)+X(5)**2*SIN(X(3))*COS(X(3)))/X(1)-EFTH*X(12)/RAD
      P(7)=(X(5)*X(8)+X(6)*X(9)+X(10)*(X(6)**2+(X(5)*SIN(X(3)))**2-2./X(
      11))+X(11)*(-X(4)*X(5)-2.*X(5)*X(6)*COS(X(3))/SIN(X(3)))+X(12)*(-X(
```

```fortran
24)*X(6)+X(5)**2*SIN(Y(3))*COS(X(3))))/X(1)**2
      P(8)=0.
      P(9)=-2.*SIN(X(3))*COS(X(3))*X(5)**2*X(10)/X(1)+X(1)*(-2.*X(5)*X(
     16)/(X(1)*SIN(X(3))**2)-(4TH*X(11)*COS(X(3))/(SIN(X(3))**3*RAD))+
     2X(5)**2*(SIN(X(3))**2-COS(X(3))**2)*X(12)/X(1)
      P(10)=-X(7)+X(5)*X(11)/X(1)+X(6)*X(12)/X(1)
      P(11)=(-X(3)-2.*X(5)*SIN(X(3))**2*X(13)+X(11)*(X(4)+2.*X(6)*COS(X(
     13))/SIN(X(3)))-2.*X(5)*SIN(X(3))*COS(X(3))*X(12))/X(1)
      P(12)=(-X(9)-2.*X(6)*X(10)+2.*X(5)*COS(X(3))*X(11)/SIN(X(3))+X(4)*
     1X(12))/X(1)
      RETURN
      END


      SUBROUTINE MATMUL (A,E,C,N,M,K)

C     THIS SUBROUTINE IS USED FOR MATRIX MULTIPLICATION
C
C
      DIMENSION A(N,M),E(M,K),C(N,K)
      DO 25 I=1,N
      DO 30 L=1,K
      SUM=0.
      DO 20 J=1,M
   20 SUM=SUM+A(I,J)*E(J,L)
   30 C(I,L)=SUM
   25 CONTINUE
      RETURN
      END
```

79

Vita

Eugene A. Smith was born 27 July 1945, in Middleville NY. He attended West Canada Valley Central School where he was valedictorian of his 1963 graduating class. An Air Force Academy appointment followed, and upon graduation in 1967, he was commissioned a second lieutenant and sent to Reese AFB for Undergraduate Pilot Training. After UPT, he served in South East Asia flying A-1's and O-2's. From this overseas tour he received the Silver Star, the Distinguished Flying Cross, and eight Air Medals.

Upon returning to the United States, Lt Smith was assigned to Williams AFB as a T-38 instructor pilot. He instructed for two years and was in Check Section for one year.

In August of 1972 Captain Smith was notified of acceptance to the Air Force Institute of Technology in the Astronautical Engineering program.

Captain Smith currently lives in Forest Ridge, Dayton OH with his wife, Glenda and daughter, Andrea.

This thesis was typed by Katherine Randall