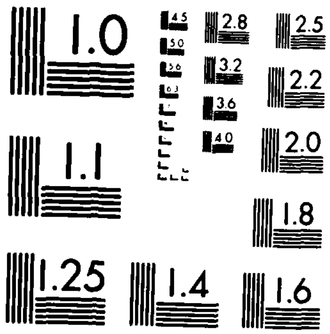
AD-A151 693   MINIMUM TIME TURNS USING VECTORED THRUST(U) AIR FORCE   1/3
INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF
ENGINEERING G L SCHNEIDER DEC 84 AFIT/GAE/AA/84D-24

UNCLASSIFIED   F/G 1/2   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MINIMUM TIME TURNS

USING VECTORED THRUST

THESIS

Garret L. Schneider
Captain, USAF

AFIT/GAE/AA/84D-24

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/AA/84D-24

MINIMUM TIME TURNS

USING VECTORED THRUST

THESIS

Garret L. Schneider
Captain, USAF

AFIT/GAE/AA/84D-24

Approved for public release; distribution unlimited

AFIT/GAE/AA/84D-24

MINIMUM TIME TURNS USING VECTORED THRUST

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 in Aeronautical Engineering

Garret L. Schneider, B.S.

Captain, USAF

December 1984

Approved for public release; distribution unlimited

Acknowledgements

I would like to thank my thesis advisor, Lt Col George Watt, for his considerable help, direction, and patience during the course of this research. I also appreciate the help and advice of Dr. Robert Calico and Capt Lanson Hudson of my thesis committee. Finally, I could not have completed this thesis without the loving support and understanding of my wife, Susanne.

Garret L. Schneider

Table of Contents

Page

List of Figures

Page

Page

List of Tables

List of Symbols

C - control variable constraint function

C_D - drag coefficient

C_{D_0} - parasite drag coefficient

C_L - lift coefficient

C_{L_α} - lift curve slope

D - drag

E - specific energy

g - gravitational acceleration

h - altitude

K_1 - induced drag parameter

L - lift

m - mass

Q - sideforce

S - state variable constraint function

S_W - wing area

t - time

T - thrust

$\underline{U}$ - control vector

V - Velocity

V_C - corner velocity

W - weight

X - distance (x-direction)

$\underline{X}$ - state vector

Y - distance (y-direction)

List of Symbols (Continued)

α	-	angle of attack
γ	-	flight path angle
ϵ	-	thrust angle of attack
μ	-	bank angle
ν	-	thrust sideslip angle
π	-	throttle control variable
ρ	-	density
ρ_0	-	density at sea level
σ	-	density ratio
ϕ	-	pay-off function
χ	-	heading angle
$\underline{\psi}$	-	terminal constraints vector
Ω	-	stopping condition
$\dot{a}$	-	da/dt (a arbitrary)
$(\ )_0$	-	initial value
$(\ )_f$	-	final value
$(\ )^*$	-	nominal value, along the nominal path
$(\underline{\ })$	-	vector
$\{\underline{\ }\}$	-	matrix
$(\ )^T$	-	transpose
$(\ )^{-1}$	-	inverse

$$0 \leq \pi \leq 1 \tag{22}$$

The angle of attack is limited by both the maximum lift limit, $C_{L_{max}}$, and the maximum load factor, $(L/W)_{max}$. The effect on angle of attack is shown as a function of airspeed in Fig. 1. The velocity where these two limits meet is the corner velocity (V_C). The corner velocity is the velocity at which the aircraft achieves its maximum turn rate and therefore plays a very important role in minimum turning time problems.

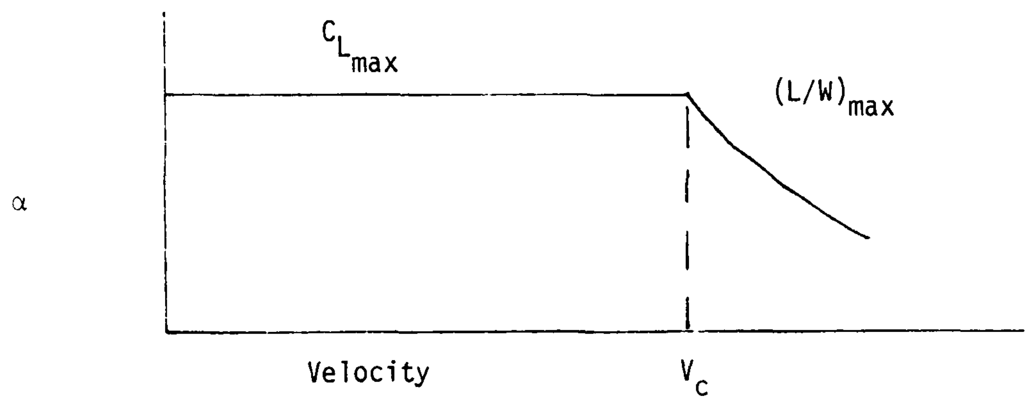


Fig 1. Maximum angle of attack vs. velocity

For velocities below the corner velocity, the angle of attack is bounded by the maximum lift limit. Solving Eq (15) for the maximum angle of attack results in

$$\alpha \leq 0.2 \text{ radians} \quad (V < V_C) \tag{23}$$

Atmospheric Model

The NASA 1962 Standard Atmosphere (8) was used for this study. Atmospheric density was taken to be $\rho = \sigma\rho_0$, where σ is the density ratio and is defined as

$$\sigma = \frac{\rho}{\rho_0} \quad \left\{ 1 - \left(\frac{n-1}{n}\right) \frac{g_0}{RT_0} h \right\}^{\left(\frac{1}{n-1}\right)} \tag{21}$$

where

$\rho_0 = 0.002377$ slugs/ft^3

$g_0 = 32.174$ ft/sec^2

$T_0 = 518.688\ ^oR$

$n = 1.235$

$R = 1715$ ft^2/sec^2 - ^oR

Control Variable Constraints

Two control variables, the throttle setting and the angle of attack, are constrained by physical considerations.

The thrust can neither be greater than the maximum thrust nor less than the minimum thrust, which is taken to be zero. Since the throttle setting is defined in Eq (19) in terms of the maximum thrust, the throttle setting is limited to

$$\frac{T}{W} = \frac{T_{max}\pi}{W} = \left(\frac{T}{W}\right)_{max} \pi \tag{20}$$

The formulation of lift, drag, and thrust to weight ratios has introduced two control variables, α and π, and several aircraft parameters. The values of these parameters complete the specification of the aircraft model. The values used in this investigation were chosen to agree with previous studies so results may be compared and are

$W = 12{,}150$ Lb	$S_W = 237$ ft^2
$K_1 = 0.05$	$\left(\frac{T}{W}\right)_{max} = 1.5$
$C_{L_\alpha} = 5.0$	$C_{D_0} = 0.02$
$\left(\frac{L}{W}\right)_{max} = 7.22$	$C_{L_{max}} = 1.0$

These values represent the nominal aircraft. As pointed out by Brinson (7:9), the drag model is unrealistically low. Also, a thrust to weight ratio of 1.5 is considerably higher than that achieved by modern high performance aircraft. After results have been obtained with these parameters, the values of $(T/W)_{max}$ and K_1 will be varied to examine the effects of thrust vectoring on a more realistic aircraft model.

$$C_L = C_{L_\alpha} \alpha \tag{15}$$

$$C_D = C_{D_0} + K_1 C_L^2 \tag{16}$$

Substituting these expressions into Eqs (13) and (14) and dividing by the weight gives

$$\frac{L}{W} = \frac{\rho V^2 S_W}{2W} C_{L_\alpha} \alpha \tag{17}$$

$$\frac{D}{W} = \frac{\rho V^2 S_W}{2W} (C_{D_0} + K_1 C_L^2) \tag{18}$$

Since the maximum available thrust remains constant during the turn, thrust can be written as

$$T = T_{max} \pi \tag{19}$$

where π is now the control variable for thrust and is referred to as the throttle or power setting. Dividing by W gives the thrust to weight ratio

$$\dot{\gamma} = \frac{g}{V} \{ \frac{T}{W} (\sin\epsilon\cos\mu - \cos\epsilon\sin\nu\sin\mu)$$

$$+ \frac{L}{W} \cos\mu - \cos\gamma \} \qquad (12)$$

These equations are written in the wind axes and describe the aircraft motion with respect to an earth-fixed coordinate frame. The state variables are X , Y , h , V , χ , and γ . The variables μ , ϵ , and ν are controls. The aircraft weight, W , and gravitational acceleration, g , are assumed constant during the maneuver. The initial altitude for all maneuvers is 13,990 feet. The gravitational acceleration at that altitude, $g = 32.131$ ft/sec^2 (8:160), is used as the constant value during the turn.

The forces L , D , and T will be discussed in the next paragraphs and will gave rise to two more control variables.

Aircraft Characteristics

The common coefficient forms of the aerodynamic lift and drag forces are

$$L = \frac{\rho V^2 S_W C_L}{2} \qquad (13)$$

$$D = \frac{\rho V^2 S_W C_D}{2} \qquad (14)$$

From incompressible aerodynamic and thin airfoil theories, the lift and drag coefficients can be expressed as

$$m\dot{V} = T\cos\varepsilon\cos\nu - D - mg\sin\gamma \tag{4}$$

$$\dot{\chi}\cos\mu\cos\gamma - \dot{\gamma}\sin\mu =$$

$$\frac{1}{mV}\{T\cos\varepsilon\sin\nu - Q + mg\sin\mu\cos\gamma\} \tag{5}$$

$$\dot{\chi}\sin\mu\cos\gamma + \dot{\gamma}\cos\mu =$$

$$\frac{1}{mV}\{T\sin\varepsilon + L - mg\cos\mu\cos\gamma\} \tag{6}$$

Since this study does not allow for sideforce, Q = 0. Rearranging, the equations of motion become

$$\dot{X} = V\cos\gamma\cos\chi \tag{7}$$

$$\dot{Y} = V\cos\gamma\sin\chi \tag{8}$$

$$\dot{h} = V\sin\gamma \tag{9}$$

$$\dot{V} = g\left\{\frac{T}{W}\cos\varepsilon\cos\nu - \frac{D}{W} - \sin\gamma\right\} \tag{10}$$

$$\dot{\chi} = \frac{g}{V\cos\gamma}\left\{\frac{T}{W}(\cos\varepsilon\sin\nu\cos\mu + \sin\varepsilon\sin\mu) + \frac{L}{W}\sin\mu\right\} \tag{11}$$

II. The Minimum Time to Turn Problem

Before results can be analyzed and compared, it is necessary to completely define all aspects of the problem. The problem will be defined in terms of the maneuver to be flown, the equations describing the motion of the aircraft, the characteristics which model the aircraft, atmospheric properties, and practical physical constraints on the control variables.

The Maneuver

The maneuver is defined by specifying initial and final conditions. The turn is initiated with the aircraft in straight (zero heading angle) and level (zero flight path angle) flight at an altitude of 13,990 feet and a specified initial velocity. The maneuver is completed when the aircraft reaches a final heading angle of 180^{0} with zero flight path angle.

Equations of Motion

The equations of motion for flight of a point mass aircraft over a flat earth are derived by Miele (11:42-49) as

$$\dot{X} = V \cos\gamma \cos\chi \tag{1}$$

$$\dot{Y} = V \cos\gamma \sin\chi \tag{2}$$

$$\dot{h} = V \sin\gamma \tag{3}$$

both to include sideforce. Optimal control schedules, trajectories, and times were reported for three sets of initial conditions, two of which are the same as those used by Johnson (5) and Finnerty (6). Since the original aircraft model from (4) was considered unrealistic due to its low induced drag and high thrust to weight ratio, Brinson (7) also varied these two parameters while excluding sideforce in an attempt to match the results of Well and Berger (10). While only moderately successful in this attempt, the results are useful for evaluating the benefits of thrust vectoring to reduce turning times.

The results of these previous studies are summarized in Appendix G.

The results of this study are then discussed and the use of thrust vectoring is compared against other methods of reducing turning time. Finally, conclusions and recommendations are given in Section VII.

Summary of Current Knowledge

Humphreys, Hennig, Bolding and Helgeson (4) used a sequential gradient-restoration algorithm to determine the optimal controls required for an aircraft to make a minimum time turn in three dimensions. For two different sets of initial conditions, a variety of final conditions and thrust to weight ratios were considered. Three controls were used: angle of attack, bank angle, and thrust.

Johnson (5) used a suboptimal numerical technique to find optimal control schedules which minimized the turning time. The same aircraft model and two cases reported in (4) were used to verify the technique and to compare the effects of in-flight thrust reversing on reducing the time to turn. Finnerty (6) used this same technique and thrust-reversing aircraft, but restricted the maneuvers to the vertical plane.

Well and Berger (10) used a different optimization technique, a multiple-shooting algorithm, to investigate minimum time 180^{o} turns. However, since the specific aircraft characteristics they used are different from those given in (4) and subsequently used by Johnson (5) and Finnerty (6), a direct comparison of results is impossible. The conclusions of (10) do serve as a good qualitative check of optimal minimum time maneuver sequences.

Most recently, Brinson (7) used the aircraft model from (4) and the suboptimal numerical technique developed by Johnson (5), but modified thin airfoil theories. Since the duration of the maneuver is small, typically on the order of 10 seconds, the fuel consumed during the maneuver is negligible and the aircraft weight remains constant. The aircraft engine is considered an ideal jet. Since the maneuver covers a small altitude range, the corresponding small variations in atmospheric density have little effect on the engine performance, and the maximum available thrust remains constant. Additionally, it is assumed that the aircraft flies a coordinated turn (i.e., zero sideslip).

Finally, the controls are allowed to vary instantaneously. This simplification eliminates the need to consider controller response characteristics which, although obviously an important consideration in the implementation of any control schedule, would only cloud the comparison of results against those of studies investigating other ways to reduce turning time.

Approach

The particular aircraft model, initial state, and final conditions used in this investigation were chosen from previous studies so meaningful comparisons could be made between the various control strategies.

Unfortunately, this results in a two-point boundary value problem which cannot be solved in closed form. Therefore, a numerical technique must be employed to find the optimal controls and resultant trajectories. The choice of the particular technique used in this study, the steepest-ascent method (9), is discussed in Section III. The method is presented in detail in Section IV, while the implementation and solution of the problem is covered in Section V.

Assumptions

Several assumptions regarding the aircraft, its dynamics, and the controls are incorporated into this study. These assumptions are not only necessary to reduce the complexity of the problem to manageable proportions, but are common in this type of study as a source of meaningful comparisons of results. The assumptions are :

1. aircraft is point mass
2. flat, non-rotating earth
3. NASA 1962 Standard Atmosphere
4. constant gravitational acceleration
5. $C_L = C_{L_\alpha} \alpha$ (up to stall)
6. $C_D = C_{D_0} + K_1 C_L^2$ (parabolic drag polar)
7. aircraft weight remains constant during maneuver
8. ideal jet engine
9. constant maximum available thrust
10. coordinated turn (no sideslip)
11. instantaneous controls

The aircraft is modelled as a point mass over a flat, non-rotating earth. Atmospheric properties are taken from the NASA 1962 Standard Atmosphere (8) and the gravitational acceleration is assumed to be constant over the small altitude range covered during the maneuver. The lift coefficient is taken to be a linear function of the angle of attack up to the stall limit, while the drag coefficient is a function of the square of the lift coefficient (parabolic drag polar). These lift and drag assumptions are substantiated by incompressible aerodynamic and

Several studies present optimal controls and trajectories to minimize turning time. Humphreys, Hennig, Bolding and Helgeson (4) investigated three-dimensional aircraft dynamics. Johnson (5) included the possibility of in-flight thrust reversal. Finnerty (6) constrained maneuvers to the vertical plane. Brinson (7) considered the effects of sideforce in reducing the time to turn. However, the use of vectored thrust has not been addressed. This study, then, examines the effects of thrust vectoring on minimizing the time to turn for a high performance aircraft.

Problem Statement

This investigation seeks the optimal control schedules and resulting trajectories which will minimize the time to maneuver a high performance aircraft with thrust vectoring capability from a given initial state to a prescribed set of final conditions.

The controls to be optimized are bank angle, angle of attack, thrust, thrust angle of attack, and thrust sideslip angle. These controls are subject to practical physical constraints: maximum angle of attack limit, maximum structural load factor, and minimum and maximum thrust.

To evaluate the effects of thrust vectoring on in-flight maneuverability, the optimal controls and resultant trajectories and minimum turning times will be obtained and compared against the results of previous studies.

MINIMUM TIME TURNS USING VECTORED THRUST

I. Introduction

Background

In air-to-air combat, minimum time turns are important for both the attacker and the evader. The normal procedure is to bank the aircraft and to use only a component of the lift vector for turning the aircraft. The maximum turn rate that an aircraft can achieve is limited by physical constraints; in particular, structural and angle of attack limits. Since aircraft designed for similar missions (i.e., fighter aircraft) tend to have similar design considerations and constraints, they consequently exhibit similar turning rates and times.

In an effort to reduce turning time and increase in-flight maneuverability, the Air Force is currently investigating the use of vectored engine thrust and expects "performance may very well be boosted" (1). By vectoring thrust, it is possible that an aircraft will get to its corner velocity in less time - the corner velocity being the velocity at which the aircraft achieves its maximum turn rate.

Noteably, a two-dimensional convergent/divergent nozzle has been successfully ground demonstrated (2). The F-15 advanced technology short takeoff and landing (STOL) demonstrator aircraft with thrust-directing two-dimensional engine nozzles will be flight tested in 1988 (3). The "technology developed in the STOL demonstrator program will 'have far-reaching implications for the Air Force, especially for the next generation of fighters.'" This flight demonstration is also considered "...critical for future aircraft programs."

Abstract

The objective of this investigation is to determine the optimal controls and trajectories which minimize the time to turn for a high performance aircraft with thrust vectoring capability. All determinations are subject to practical physical constraints. The determined controls and trajectories are then compared against other methods of turning in minimum time to conclude the effects and advantages of thrust vectoring.

The results indicate that the use of vectored thrust can substantially reduce turning times and increase in-flight maneuverability. The greater the velocity at which the turn is initiated, the more the range of thrust vectoring capability is used and the greater the reduction in turning time.

For velocities above the corner velocity, the angle of attack is limited by the maximum load factor, 7.22. Substituting this value into Eq (17) gives

$$\frac{\rho V^2 S_W C_{L_\alpha}\,\alpha}{2W} \leq 7.22 \qquad (V > V_C) \tag{24}$$

Using the relation for the density, Eq (21), and substituting for known parameters reduces Eq (24) to

$$\alpha \leq \frac{62286.8}{\sigma V^2} \qquad (V > V_C) \tag{25}$$

Eqs (23) and (25) describe the angle of attack limits as shown in Fig 1. The corner velocity is the velocity at which the lift and load factor limits are equal and may be solved for by equating Eqs (23) and (25)

$$0.2 = \frac{62286.8}{\sigma V_C^{\,2}} \qquad (V = V_C) \tag{26}$$

which leads to

$$V_C = 558.06 \ \sigma^{-1/2} \qquad (27)$$

The incorporation of these angle of attack constraints is detailed in Section V. The importance of the corner velocity is addressed in Section VI, Results.

No constraints were placed on the thrust angle of attack, ε , and sideslip, ν . While it does not appear that this is physically practical (2, 3), these angles were allowed full range in order to determine how much range of thrust vectoring would be exploited if it were available. The effects of constraining the two thrust angles were later examined and are discussed in Section VI, Results. No constraint was placed on the bank angle.

III. The Optimal Control Problem

The formulation of the minimum turning time problem involves first-order non-linear differential equations and partial specification of initial and final conditions on the state variables. The optimal control problem is to determine, out of all possible programs for the control variables, the one program that minimizes or maximizes a terminal quantity while simultaneously satisfying the required state variable initial and final conditions. This is a two-point boundary value problem which cannot be solved in closed form. A numerical solution is required.

Many techniques and methods are available to solve this type of problem and are widely reported in the literature. Three methods were considered for this study. Johnson (5) transformed the optimal control problem into a parameter optimization problem by assuming a known mathematical form with a number of unknown constants for the control variables. This reduces the problem to one of finding the coefficients which satisfy the conditions of the problem.

Finnerty (6) and Brinson (7) also used this technique, modifying a computer program developed by Johnson (5) to incorporate maneuver constraints and aircraft characteristics particular to their studies. The parameter optimization method was considered for this investigation, but was rejected for two reasons. First, although a computer program already existed, it had been modified so many times that accurate documentation was non-existent. Since the existing program could not be used and a new program would have to be written, one advantage of the parameter optimization technique was negated. Second, it was feared that the addition of more control variables, as required to add thrust vectoring capability, and the corresponding increase in coefficients to be determined would result in prohibitively long computer execution times.

The generalized reduced gradient method, as reported by Gabriele and Ragsdell (12), was the next candidate technique considered for solving this optimal control problem. However, the non-linear control variable inequality constraints required for the minimum time to turn problem could not be easily incorporated. The use of a penalty function to attach the constraints to the terminal pay-off function was deliberated, but it was felt that this might compromise the optimality of any solutions which might be obtained. Therefore, this technique was also rejected.

A steepest-ascent method, presented in detail by Bryson and Denham (9), was chosen to determine the optimal controls for minimizing turning time for a high performance aircraft with thrust vectoring capability. The procedure begins with a non-optimal control variable program. The equations of motion are integrated forward from the initial conditions using these nominal controls. In general, the resulting state variable time histories will not satisfy the specified final conditions. Small perturbations of the control variables about the nominal trajectory are considered to drive the terminal quantities to their specified values while simultaneously extremizing a pay-off function. For this problem, the pay-off function is the final time. By continuing this process along the direction of steepest ascent (or descent) in the control variable hyperspace, a control variable program that minimizes the time to turn while satisfying final conditions is obtained.

Two qualities of the steepest-ascent method led to its being chosen to solve this optimum programming problem. First, the method is straightforward. Second, state variable inequality constraints are easily incorporated.

It should be noted that neither the steepest-ascent method nor any other method for numerically solving this type of optimal control problem is guaranteed to find globally optimal solutions. Only relative minima or maxima may be found. The determination of a global extremum must be made by examining all of the local extrema. Since all of the techniques considered have this same drawback, it was not a factor in the choice of which method to use.

The steepest-ascent method is described in detail next, in Section IV. The specific application of the method to the problem at hand is presented in Section V.

IV. The Steepest-Ascent Method

The steepest-ascent method (9) is a systematic procedure for determining optimum programs for nonlinear systems with terminal constraints. The technique begins with a nominal (non-optimum) control variable program. By locally linearizing about these nominal controls, the program is improved in steps until a pay-off function (final time) is extremized while specified final conditions are satisfied.

Problem Statement

A general problem of finding the maximum of a nonlinear function of many variables subject to nonlinear constraints on these variables may be stated as follows:

Determine $\underline{U}(t)$ in the interval $t_0 \leq t \leq t_f$ so as to maximize

$$\phi = \phi(\underline{X}(t_f), t_f) \tag{28}$$

subject to constraints

$$\underline{\Psi} = \underline{\Psi}[\underline{X}(t_f), t_f] = 0 \tag{29}$$

$$\dot{\underline{X}} = \underline{f}[\underline{X}(t), \underline{U}(t), t] \tag{30}$$

$$t_0 \text{ and } \underline{X}(t_0) \text{ given} \tag{31}$$

$$t_f \text{ determined by } \Omega = \Omega[\underline{X}(t_f), t_f] = 0 \tag{32}$$

The nomenclature for this problem statement follows:

$$\underline{U}\ (t) = [U_1\ (t),\ \ldots\ U_m\ (t)]^T \tag{33}$$

is an $m \times 1$ vector of control variable functions, which are free to be chosen;

$$\underline{X}\ (t) = [X_1\ (t),\ \ldots\ X_n\ (t)]^T \tag{34}$$

is an $n \times 1$ vector of state variable functions, which result from a choice of $\underline{U}\ (t)$ and specified values of $\underline{X}\ (t_0)$;

$$\underline{\Psi} = [\Psi_1,\ \ldots\ \Psi_p]^T \tag{35}$$

is a $p \times 1$ vector of terminal constraint functions, each of which is a known function of $\underline{X}\ (t_f)$ and t_f;

$$\underline{f} = [f_1,\ \ldots\ f_n]^T \tag{36}$$

is an $n \times 1$ vector of known functions of $\underline{X}\ (t)$, $\underline{U}\ (t)$, and t ; ϕ is the pay-off function and is a known function of $\underline{X}\ (t_f)$ and t_f ; $\Omega = 0$ is the stopping condition that determines the final time t_f , and is a known function of $\underline{X}\ (t_f)$ and t_f .

Formulation of the Method

The steepest-ascent method begins with a reasonable, nominal control variable program, $\underline{U}^*(t)$. These controls and the initial conditions, Eq (31), are used in the differential equations of motion, Eq (30), to numerically calculate the state variable time history $\underline{X}^*(t)$ until $\Omega = 0$. In general, this nominal trajectory will not satisfy the terminal conditions $\Psi = 0$ or yield the maximum possible value of ϕ .

Small perturbations $\delta\underline{U}(t) = \underline{U}(t) - \underline{U}^*(t)$ about the nominal control variable program cause perturbations in the state variable programs, $\delta\underline{X}(t) = \underline{X}(t) - \underline{X}^*(t)$. Substituting these perturbation relations into the differential equations of motion, Eq (30), yields, to first order in the perturbations, the linear differential equations for $\delta\underline{X}(t)$

$$\frac{d}{dt}[\delta\underline{X}(t)] = [\underline{F}(t)]\delta\underline{X}(t) + [\underline{G}(t)]\,\delta\underline{U}(t) \tag{37}$$

where

$$\{\underline{F}(t)\} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial X_1}\right)^*, & \cdots\cdots & \left(\frac{\partial f_1}{\partial X_n}\right)^* \\ \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial X_1}\right)^*, & \cdots\cdots & \left(\frac{\partial f_n}{\partial X_n}\right)^* \end{bmatrix} \tag{38}$$

$$\{ G(t) \} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial U_1} \right)^*, & \cdots\cdots & \left(\frac{\partial f_1}{\partial U_m} \right)^* \\ \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial U_1} \right)^*, & \cdots\cdots & \left(\frac{\partial f_n}{\partial U_m} \right)^* \end{bmatrix}$$

and ( )* indicates the partial derivatives are evaluated along the nominal path.

There are three sets of differential equations adjoint to the differential equations of motion and three sets of influence functions $\underline{\lambda}$ which tell how much a terminal condition is changed by a small change in some initial state variable. The three sets of influence functions are $\underline{\lambda}_\phi$, $\underline{\lambda}_\Psi$, and $\underline{\lambda}_\Omega$.

The general form of the adjoint differential equations is

$$\frac{d}{dt} \underline{\lambda} = -[\underline{F} (t)]^T \underline{\lambda} \tag{40}$$

with boundary conditions

$$\underline{\lambda}_\phi^T (t_f) = \left(\frac{\partial \phi}{\partial \underline{X}} \right)^*_{t = t_f} \tag{41}$$

$$\underline{\lambda}_{\Psi}^{T}(t_f) = \left(\frac{\partial \underline{\Psi}}{\partial \underline{X}}\right)^{*}_{t = t_f} \tag{42}$$

$$\underline{\lambda}_{\Omega}^{T}(t_f) = \left(\frac{\partial \Omega}{\partial \underline{X}}\right)^{*}_{t = t_f} \tag{43}$$

where

$$\frac{\partial \phi}{\partial \underline{X}} = \left\{ \frac{\partial \phi}{\partial X_1}, \ldots \frac{\partial \phi}{\partial X_n} \right\} \tag{44}$$

$$\frac{\partial \underline{\Psi}}{\partial \underline{X}} = \begin{bmatrix} \left(\frac{\partial \Psi_1}{\partial X_1}\right), & \cdots & \left(\frac{\partial \Psi_1}{\partial X_n}\right) \\ \vdots & & \vdots \\ \left(\frac{\partial \Psi_p}{\partial X_1}\right), & \cdots & \left(\frac{\partial \Psi_p}{\partial X_n}\right) \end{bmatrix} \tag{45}$$

$$\frac{\partial \Omega}{\partial \underline{X}} = \{ \frac{\partial \Omega}{\partial X_1} , \ldots . \frac{\partial \Omega}{\partial X_n} \} \tag{46}$$

The adjoint equations, Eq (40), must be integrated backward since the boundary conditions are given at the terminal point, $t = t_f$.

The steepest-ascent method seeks to find the $\delta \underline{U}\ (t)$ programs that maximize the change in the pay-off function, $d\phi$, for a given value of the integral

$$(dP)^2 = \int_{t_0}^{t_f} \delta \underline{U}^T (t) \{\underline{W}\ (t)\} \delta \underline{U}\ (t)\ dt \tag{47}$$

Since dP is the "length" of the step in the $\underline{U}$ hyperspace, dP must be chosen small enough for the linearization leading to Eq (37) to be reasonable. $[\underline{W}\ (t)]$ is an arbitrary symmetric $m \times m$ matrix of weighting functions which may be used to improve tne convergence of the steepest-ascent procedure.

The proper change in the control variable program, $\delta \underline{U}(t)$, is (9:257)

$$\delta \underline{U}(t) = \pm \{\underline{W}\}^{-1} \{\underline{G}\}^{T} (\underline{\lambda}_{\phi\Omega} - \underline{\lambda}_{\psi\Omega} I_{\psi\psi}^{-1} I_{\psi\phi})$$

$$\cdot \left[\frac{(dP)^2 - d\underline{\beta}^T I_{\psi\psi}^{-1} d\underline{\beta}}{I_{\phi\phi} - I_{\psi\phi}^{T} I_{\psi\psi}^{-1} I_{\psi\phi}} \right]^{1/2}$$

$$+ \{\underline{W}\}^{-1} \{\underline{G}\}^{T} \underline{\lambda}_{\psi\Omega} I_{\psi\psi}^{-1} d\underline{\beta} \tag{48}$$

where

$$d\underline{\beta} = d\underline{\Psi} - \underline{\lambda}_{\psi\Omega}^{T} (t_0)\, \delta \underline{X}(t_0) \tag{49}$$

$$\underline{\lambda}_{\phi\Omega} = \underline{\lambda}_{\phi} - \frac{\dot{\phi}}{\dot{\Omega}} \underline{\lambda}_{\Omega} \tag{50}$$

$$\underline{\lambda}_{\psi\Omega} = \underline{\lambda}_{\psi} - \underline{\lambda}_{\Omega} \frac{\dot{\underline{\psi}}^{T}}{\dot{\Omega}} \tag{51}$$

$$\dot{\phi} = \left(\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial \underline{X}} \underline{f} \right)^{*}_{t = t_f} \tag{52}$$

$$\dot{\underline{\Psi}} = \left(\frac{\partial \underline{\Psi}}{\partial t} + \frac{\partial \underline{\Psi}}{\partial \underline{X}} \underline{f} \right)^{*}_{t = t_f} \tag{53}$$

$$\dot{\Omega} = \left(\frac{\partial \Omega}{\partial t} + \frac{\partial \Omega}{\partial \underline{X}} \underline{f} \right)^{*}_{t = t_f} \tag{54}$$

$$I_{\Psi\Psi} = \int_{t_0}^{t_f} \underline{\lambda}_{\Psi\Omega}^{T} \{\underline{G}\} \{\underline{W}\}^{-1} \{\underline{G}\}^{T} \underline{\lambda}_{\Psi\Omega} \, dt \tag{55}$$

$$I_{\Psi\phi} = \int_{t_0}^{t_f} \underline{\lambda}_{\Psi\Omega}^{T} \{\underline{G}\} \{\underline{W}\}^{-1} \{\underline{G}\}^{T} \underline{\lambda}_{\phi\Omega} \, dt \tag{56}$$

$$I_{\phi\phi} = \int_{t_0}^{t_f} \underline{\lambda}_{\phi\Omega}^{T} \{\underline{G}\} \{\underline{W}\}^{-1} \{\underline{G}\}^{T} \underline{\lambda}_{\phi\Omega} \, dt \tag{57}$$

The + sign in Eq (48) is used if ϕ is to be increased; the - sign is used if ϕ is to be decreased. The numerator under the square root in Eq (48) can become negative if $d\underline{\beta}$ is chosen too large; thus there is a limit to the size of $d\underline{\beta}$ for a given dP. Since dP is chosen to insure valid linearization, the $d\underline{\beta}$ asked for must also be limited.

For the change in the control variable program given by Eq (48), the predicted change in the pay-off function ϕ is

$$d\phi = \pm \{((dP)^2 - d\underline{\beta}^T I_{\Psi\Psi}^{-1} d\underline{\beta}) \cdot (I_{\phi\phi} - I_{\Psi\phi}^T I_{\Psi\Psi}^{-1} I_{\Psi\phi})\}^{1/2} + I_{\Psi\phi}^T I_{\Psi\Psi}^{-1} d\underline{\beta} + \underline{\lambda}_{\phi\Omega}^T (t_0)\, \delta\underline{X} (t_0) \tag{58}$$

If $d\underline{\Psi} = 0$ and $\delta\underline{X} (t_0) = 0$, from Eq (49), $d\underline{\beta} = 0$ and Eq (58) becomes

$$\frac{d\phi}{dP} = \pm (I_{\phi\phi} - I_{\Psi\phi}^T I_{\Psi\Psi}^{-1} I_{\Psi\phi})^{1/2} \tag{59}$$

which is a "gradient" in the function space. As the optimum control variable program is approached and the terminal constraints are met ($d\underline{\Psi} = 0$), this gradient must tend to zero.

Control Variable Inequality Constraint. The maximum load factor limit is the constraint in effect at velocities above the corner velocity and is given by Eq (25). Putting that equation in the form of Eq (61) gives

$$C(\underline{X}\ ,\ \underline{U}) = \alpha - \frac{62286.8}{\sigma V^2} = 0 \tag{93}$$

While the angle of attack is on this constraint boundary, the adjoint differential equations are given by Eq (65). Two additional vectors are required for the calculation of the influence functions. The first is

$$\frac{\partial C}{\partial \underline{U}} = \{0,\ 1,\ 0,\ 0,\ 0\}^T \tag{94}$$

and the only non-zero elements of the second, $\partial C/\partial \underline{X}$, are

The Gradient and Control Variable Changes

The calculation of the control variable changes, $\delta \underline{U}(t)$, and the gradient of the function space in Eqs (48) and (55) through (57) require the gradient matrix $[\underline{G}]$. The elements of this matrix are given in Appendix B. Taking $\delta \underline{X}(t_0) = 0$, the constant values associated with the gradient and control program changes are

$$d\beta = d\Psi = d\gamma \tag{89}$$

$$\underline{\lambda}_{\phi}(t) = 0 \tag{90}$$

$$\underline{\lambda}_{\psi}^{T}(t) = \{0, 0, 0, 0, 0, 1\} \tag{91}$$

$$\underline{\lambda}_{\Omega}^{T}(t) = \{0, 0, 0, 0, 1, 0\} \tag{92}$$

Inequality Constraints

Two control variables are constrained by physical considerations: the angle of attack, α , and the throttle control, π . The constraint on the throttle setting, $0 \leq \pi \leq 1$, and the maximum lift coefficient limit on the angle of attack are not constraints in the sense that a variable is determined in terms of the remaining control and/or state variables. Rather, they are bounds on the minimum or maximum values that these variables may attain. The only constraint on a function of the control and/or state variables is the maximum load factor limit on the angle of attack (Fig 1).

$$f_6 = \dot{\gamma} = \frac{g}{V} \; \{ \left(\frac{T}{W}\right)_{max} \pi(\sin\varepsilon\cos\mu - \cos\varepsilon\sin\nu\sin\mu)$$

$$- \cos\gamma + C_1 \sigma V^2 C_{L_\alpha} \alpha\cos\mu \} \qquad (85)$$

where

$$C_1 = \frac{\rho_0 S_W}{2W}$$

Adjoint Equations

Integration of the adjoint differential equations, Eq (40), requires the adjoint matrix $[\underline{F}]$. The elements of this matrix are given in Appendix A. The boundary conditions for the adjoint equations, Eqs (41) through (46), are

$$\underline{\lambda}_\phi (t_f) = 0 \qquad (86)$$

$$\underline{\lambda}_\psi^T (t_f) = \{0, 0, 0, 0, 0, 1\} \qquad (87)$$

$$\underline{\lambda}_\Omega^T (t_f) = \{0, 0, 0, 0, 1, 0\} \qquad (88)$$

with

$$\underline{f} = [f_1, f_2, \ldots f_6]^T \tag{79}$$

and

$$f_1 = \dot{X} = V \cos\gamma \quad \cos\chi \tag{80}$$

$$f_2 = \dot{Y} = V \cos\gamma \quad \sin\chi \tag{81}$$

$$f_3 = \dot{h} = V \sin\gamma \tag{82}$$

$$f_4 = \dot{V} = g\{(\frac{T}{W})_{max}\pi \cos \epsilon \cos\nu \quad - \sin\gamma$$

$$- C_1\sigma V^2 \ (C_{D_0} + K_1 \ C_{L_\alpha}^{\ 2} \ \alpha^2) \ \} \tag{83}$$

$$f_5 = \dot{\chi} = \frac{g}{V\cos\gamma} \ \{ (\frac{T}{W})_{max} \pi(\cos\varepsilon\sin\nu\cos\mu + \sin\varepsilon\sin\mu)$$

$$+ C_1\sigma V^2 C_{L_\alpha} \ \alpha\sin\mu \ \} \tag{84}$$

The forward integration of the equations of motion is stopped when the heading angle reaches 180^0.

The terminal constraint function, also part of the maneuver definition, is

$$\Psi = \gamma_f = 0 \tag{77}$$

The aircraft is required to have a flight path angle of zero at the end of the maneuver.

Equations of Motion

The nonlinear differential equations of motion, Eqs (30) and (36), were given earlier as Eqs (7) through (12). However, this earlier form did not explicitly involve the state and control variables given in Eqs (72) and (73). Substituting Eqs (17), (18), (20), and (21) into Eqs (7) through (12) and rearranging yields the following form of the equations of motion

$$\dot{\underline{X}} = \underline{f}\,[\underline{X}\,(t),\ \underline{U}\,(t)] \tag{78}$$

The state vector, $\underline{X}(t)$, is

$$\underline{X}(t) = [X(t),\ Y(t),\ h(t),\ V(t),\ \chi(t),\ \gamma(t)]^T \tag{73}$$

and the only non-zero initial conditions are

$$h(t_0) = h_i = 13{,}990 \text{ feet} \tag{74}$$

and

$$V(t_0) = V_i \tag{75}$$

Three values of initial velocity, V_i, will be used in this study. The remaining initial conditions, $X(t_0)$, $Y(t_0)$, $\chi(t_0)$, and $\gamma(t_0)$, are all zero.

The stopping condition which determines the final time is part of the definition of the maneuver

$$\Omega = |\chi| - \pi = 0 \tag{76}$$

where π is in radians.

V. Solving the Minimum Time Problem

The steepest-ascent method, as detailed in the previous section, is now specifically applied to the optimum programming problem of minimum turning time for a high performance aircraft with thrust vectoring capability.

The problem, as formulated here, was written into a FORTRAN 5 computer program for numerical solution. A copy of the program is included in Appendix H.

Variables

The quantity to be extremized is the final time, t_f . The final time can be minimized by maximizing the pay-off function

$$\phi = -t_f \tag{71}$$

The control vector, $\underline{U}$ (t), is

$$\underline{U}(t) = [\mu(t), \alpha(t), \pi(t), \epsilon(t), \nu(t)]^T \tag{72}$$

$$dt_f = -\frac{1}{\dot{\Omega}} \int_{t_0}^{t_f} \lambda_\Omega^T \, \underline{G} \, \delta \underline{U} \, dt \tag{70}$$

If $|dt_f|$ is greater than a preselected maximum allowable value, $\delta\underline{U}(t)$ is scaled down to achieve this value. Again, time intervals on the constraint boundary must be taken into account.

10. A new nominal control variable program is obtained using Eq (60). Steps (1) through (10) are repeated until the terminal constraints $\underline{\Psi} = 0$ are satisfied and $I_{\phi\phi} - I_{\Psi\phi}^T I_{\Psi\Psi}^{-1} I_{\Psi\phi}$, the square of the gradient, tends to zero.

4. Eqs (55), (56), and (57) are then integrated backward to obtain the values of $I_{\Psi\Psi}$, $I_{\Psi\phi}$, and $I_{\phi\phi}$. The limits of integration are modified, if necessary, to take into account any time intervals that the solution may be on a constraint boundary.

5. The values of ϕ , Ψ_1 , Ψ_2 , ... Ψ_p achieved by the nominal path are examined. Desired terminal condition changes $d\Psi_1$, $d\Psi_2$, $d\Psi_p$ are chosen to bring the next solution closer to the required terminal constraints $\underline{\Psi} = 0$.

6. A reasonable value of $(dP)^2/(t_f - t_0)$ is selected. This is a mean-square deviation of the control variable programs from the nominal to the next step. This simplification of Eq (47) may need to be adjusted for time intervals that the solution is on a constraint boundary.

7. The quantity $(dP)^2 - d\underline{\Psi}^T I_{\Psi\Psi}^{-1} d\underline{\Psi}$ is calculated. If negative, $d\underline{\Psi}$ is scaled down to make this quantity zero. If the quantity is positive, it is left as is.

8. Using dP and $d\underline{\Psi}$, modified in step (7) if necessary, and taking $\delta\underline{X}(t_0) = 0$, $\delta\underline{U}(t)$ is calculated from Eq (48).

9. If t_f , the final time, is not specified or is being extremized, the predicted change in final time for the next step, dt_f , is calculated as

While on the constraint boundary, say from $t = t_1$ to $t = t_2$, a control variable is specified in terms of the state variables and possibly the remaining control variables by the relations $C = 0$ or $S = 0$. No variation in the constrained control variable is calculated for $t_1 < t < t_2$ since the variable is not free over this interval. This same variable is omitted over the interval $t_1 < t < t_2$ from the above-mentioned four integrations, since the integrated values determine the step length and gradient in the $\underline{U}$ hyperspace with respect to free $\delta\underline{U}$.

During $t_1 < t < t_2$, the dimension of the control variable hyperspace is reduced by one. This reduction is accomplished by zeroing the appropriate elements of the weighting matrix $[\underline{W}]$.

Computing Procedure

The following steps detail how the steepest-ascent method is used to solve a general optimum programming problem (9:251).

1. The nominal trajectory is calculated by integrating the nonlinear differential equations, Eq (30), with a nominal control variable program and starting from specified initial conditions, Eq (31).

2. The influence functions $\underline{\lambda}_\phi$, $\underline{\lambda}_{\psi_1}$, λ_{ψ_2}, ... λ_{ψ_p} $\underline{\lambda}_\Omega$ are calculated by backward integration of the adjoint differential equations, Eq (40), (65), or (69), as appropriate. The required partial derivatives are evaluated along the nominal trajectory.

3. The quantities $\underline{\lambda}_{\phi\Omega}$, $\underline{\lambda}_{\psi_1\Omega}$, ... $\underline{\lambda}_{\psi_p\Omega}$ are calculated using Eqs (50) and (51).

$$\frac{dS}{dt} = \frac{\partial S}{\partial t} + \frac{\partial S}{\partial \underline{X}} \dot{\underline{X}} \qquad (67)$$

$$= \frac{\partial S}{\partial t} + \frac{\partial S}{\partial \underline{X}} \underline{f} \qquad (68)$$

and $\underline{f}$ ($\underline{X}$, $\underline{U}$, t) appears in Eq (68), dS/dt may be an explicit function of the control variables $\underline{U}$ (t). If not, succeeding time derivatives of S may be considered until $S^{(k)}$ does explicitly involve the controls. Let this value of k be called q ; this is called a q^{th} order state variable inequality constraint. $S^{(q)} = 0$ now plays exactly the same role as C = 0 did for control variable inequality constraints. The differential equations for the influence functions λ (t) are the same as Eq (65) with C replaced by $S^{(q)}$

$$\frac{d}{dt} \underline{\lambda} = - \left\{ \{\underline{F}\} - \{\underline{G}\} \left(\frac{\partial S^{(q)}}{\partial \underline{U}} \right)^{-1} \left(\frac{\partial S^{(q)}}{\partial \underline{X}} \right) \right\}^{T} \underline{\lambda} \qquad (69)$$

Effects Upon Intervals of Integration. When either a type C or type S constraint is in effect, i.e., C = 0 or S = 0 , the integration in Eqs (47), (55), (56), and (57) leading to the calculation of $(dP)^2$, $I_{\psi\psi}$, $I_{\psi\phi}$, and $I_{\phi\phi}$ must be altered.

Therefore, on the constraint boundary $C = 0$, the form of the adjoint differential equations, Eq (40), becomes

$$\frac{d}{dt} \underline{\lambda} = - \{ \{\underline{F}\} - \{\underline{G}\} (\frac{\partial C}{\partial \underline{U}})^{-1} (\frac{\partial C}{\partial \underline{X}})\}^{T} \underline{\lambda} \tag{65}$$

<u>State Variable Inequality Constraint</u>. In a type S constraint relation, Eq (62), the constraint does not explicitly involve the control variable program $\underline{U}(t)$. While the solution of the optimal control problem is on the constraint boundary, $S = 0$. Since the constraint function S must vanish, its time derivatives must also be zero.

$$\frac{d^k S}{dt^k} = S^{(k)} = 0 \tag{66}$$

for $k = 0, 1, 2, \ldots$ on the constraint boundary. Since

$$S\,[\underline{X}(t),\, t] \le 0 \tag{62}$$

where C is a scalar function of $\underline{X}(t)$, $\underline{U}(t)$, and t; S is a scalar function of $\underline{X}(t)$ and t; and $\underline{U}(t)$ must remain within the limits imposed by $C \le 0$ or $S \le 0$.

Control Variable Inequality Constraint. The type C constraint relation, Eq (61), explicitly involves the control variable program $\underline{U}(t)$. The constraint function may also involve the state variables $\underline{X}(t)$ and/or explicitly involve the independent variable t (time) .

While on the constraint boundary, $C = 0$ and the neighboring solutions must satisfy

$$\left(\frac{\partial C}{\partial \underline{X}}\right)^{*} \delta\underline{X} + \left(\frac{\partial C}{\partial \underline{U}}\right)^{*} \delta\underline{U} = 0 \tag{63}$$

Neighboring solutions must also satisfy the perturbation differential equations, Eq (37). Substituting Eq (63) into Eq (37) yields the set of perturbation equations which a neighboring solution must satisfy if it is to remain on the constraint boundary $C = 0$

$$\frac{d}{dt}(\delta\underline{X}) = \left\{ \{\underline{F}\} - \{\underline{G}\}\left(\frac{\partial C}{\partial \underline{U}}\right)^{-1}\left(\frac{\partial C}{\partial \underline{X}}\right)\right\}\delta\underline{X} \tag{64}$$

New control variable programs are obtained as

$$\underline{U}(t)_{new} = \underline{U}(t)_{old} + \delta\underline{U}(t) \tag{60}$$

where $\delta\underline{U}(t)$ is given by Eq (48). This new control variable program is used in the differential equations of motion, Eq (30), and the entire process is repeated until the terminal constraints are met and the gradient, given by Eq (59), is close to zero. The optimum control variable program has then been obtained.

Inequality Constraints

Inequality constraints on functions of the control and/or state variables have been incorporated into optimal programming problems by many investigators by adjoining penalty functions to the pay-off function. Denham and Bryson (13) and Bryson, Denham, and Dreyfus (14) include such constraints "in a manner which is naturally consistent with the necessary conditions for an extremal solution. Calculation of the influence functions on terminal quantities takes into account that portions of the path are on the constraint boundary" (13:25). While the authors only treat the case of a scalar control variable, their work is extended here to a vector of control variables.

Two types of inequality constraints are considered:

$$C[\underline{X}(t), \underline{U}(t), t] \leq 0 \tag{61}$$

and

$$\frac{\partial C}{\partial X_3} = \frac{\partial C}{\partial h} = \frac{C_2 C_3 (62286.8)}{\sigma V^2 \ (1 - C_2 h)} \tag{95}$$

$$\frac{\partial C}{\partial X_4} = \frac{\partial C}{\partial V} = \frac{2(62286.8)}{\sigma V^3} \tag{96}$$

where

$$C_2 = \frac{(n-1)}{n} \ \frac{g_0}{RT_0}$$

$$C_3 = \frac{1}{n-1}$$

<u>State Variable Inequality Constraints</u>. At the corner velocity, the constraint relation is given by Eq (26)

$$S(\underline{X}) = \frac{62286.8}{\sigma V^2} - 0.2 = 0 \tag{97}$$

This is a type S constraint since the control variables, $\underline{U}$, are not explicitly involved. Upon rearranging and taking the first time derivative of S, which must also vanish

$$\dot{S} = -0.2 \{2\sigma V\dot{V} - \frac{C_2 C_3 \sigma V^2 \dot{h}}{(1 - C_2 h)}\} \tag{98}$$

where C_2 and C_3 are as given following Eq (96). Substitution of Eqs (82) and (83) into Eq (98) yields a constraint relation which explicitly involves control variables. This relation and its partial derivatives, as required for the adjoint differential equations, Eq (69), are detailed in Appendix C.

The question now arises as to when the type S constraint is in effect and when the type C constraint is the proper relation to use for the integration of the adjoint differential equations. The type S constraint was used when the velocity was within some tolerance of the corner velocity. This velocity tolerance was varied from 0.1 to 50 feet/second, but no definite relationship between the tolerance and change in the optimal solutions was found.

Overall, the addition of the type S constraint to the numerical procedure appeared to have very little effect. Optimal solutions obtained with both type S and C constraints incorporated were essentially the same as those obtained with only the type C constraint included in the computer program.

The Weighting Matrix

The weighting matrix $[\underline{W}(t)]$ introduced in Eq (47) was taken to be an identity matrix of dimension five for two reasons. First, this reduced the complexity of the problem, with a concurrent reduction in computer execution time. Second, not enough was initially known about the problem to make a more appropriate choice. Some computer runs were made with individual diagonal elements of the weighting matrix increased/decreased by as much as a factor of ten from the nominal value of one, but the results were inconclusive as to the influence of individual elements of $[\underline{W}]$ on the convergence to an optimal solution.

Further research discovered that "the proper weighting matrix is the inverse of the Hessian" (15:58). The existing computer program was modified so that

$$W_{ij}(t) = \frac{\partial^2 t_f}{\partial U_i \partial U_f} \tag{99}$$

Since the final time is not an explicit function of the control variables, implicit numerical differentiation was required. Also, $[\underline{W}]$ was no longer a constant matrix, and had to be evaluated at every time step over the interval of integration from $t = t_0$ to $t = t_f$. Since the time increment used for integrating the equations of motion was 0.1 second, the final time was nominally on the order of 10 seconds, and there are five control variables, this modification resulted in an unacceptable increase in computer execution time and was abandoned.

Finally, it was thought that the diagonal elements of a constant weighting matrix could be used to normalize the directions in the five-dimensional control variable hyperspace. Normalizing the range of all controls to be from zero to one or between ± one should change the relative influence of the angle of attack and throttle controls as compared to the bank and thrust angle controls since the latter three controls were not restricted but allowed to range between $\pm 180^{\circ}$. However, for most solutions of interest, the angle of attack and throttle controls stayed at or near their maximum values. Therefore, normalization of the control space was not incorporated.

All solutions were generated with a constant 5 x 5 identity matrix as $[\underline{W}]$. Over any time interval during which control variable constraints were in effect, the element of $[\underline{W}]$ corresponding to the constrained control was set to zero to eliminate that variable's contribution to the calculation of the gradient and the next iteration's control variable program.

Step Length

Successive improvements in the control variable program are obtained by moving a step length along the direction of the steepest ascent in the control variable hyperspace. This step length is given by Eq (47). The suggested procedure (9:251) recommends selecting a reasonable step length rather than calculating a value for $(dP)^2$. Initially, this suggested procedure was followed, assuming small, constant values for the elements of $\delta\underline{U}$. However, the addition of these five arbitrary values only further complicated the problem without noticeably improving convergence to an optimal solution.

The determination of the step length was revised to be directly proportional to the error in the terminal constraint, $\gamma(t_f)$, in the following way

$$dP = [t_f \{2\gamma(t_f)\}^2]^{1/2} \tag{100}$$

This greatly improved convergence to solutions. The choice of the constant 2 in Eq (100) was purely arbitrary. All solutions were obtained with this method of calculating the step length.

Convergence Criteria

Angular. The numerical technique was considered to have reached an optimal solution when the terminal constraint was satisfied within 0.001 radians ($|\gamma(t_f)| \leq 0.001$). The final time was determined by the stopping condition being satisfied to within 0.001 radians ($|\chi| - \pi \leq 0.001$).

Gradient. It was originally expected that the gradient would approach zero as an optimal solution was reached. This expectation was never completely realized. Upon modification of the computer program to remove the influence of a constrained variable upon the calculation of the gradient while the constraint was in effect, the magnitudes of the gradients were indeed reduced, as expected. However, lower turning times were not consistently accompanied by lower gradients.

Difficulty in obtaining acceptable convergence when the optimal solution possesses a singular arc is not new. H.J. Kelley encountered this same problem in his work in optimization as early as the 1960s.

Denham and Bryson reported over 20 years ago that: "Modifications to improve convergence in singular arc problems are being investigated by the authors and others" (13:29), but the problem apparently remains unresolved. It is noted, though, that the optimal solutions obtained in this study did move toward the singular arc, as expected.

It is recalled here that the numerical solution of this optimal control problem only generates relative minima or maxima. The determination of a global extremum, or optimum solution, must be made by examination of all of the local extrema. This study does not presume to have found all local extrema in the function space and therefore cannot claim an optimum solution. Because of the poor correlation between lower turning times and correspondingly lower gradients, the gradient was not given major consideration in the evaluation of which control program and trajectory were most optimal; the time to turn was used as the major criterion.

VI. Results

Optimal solutions were obtained for a wide variety of cases, covering a range of initial conditions and aircraft characteristics. The results for the nominal aircraft, which has been used as the baseline for the previous studies (4, 5, 6, 7) are discussed first. The effects of varying aircraft characteristics are examined next.

Nominal Aircraft

The nominal, or baseline, aircraft, as previously discussed, has both unrealistically low induced drag and a high thrust/weight ratio. However, five cases run with this model for comparison purposes are presented. The best, or most optimal, results are given in Table I and will be discussed in detail. These three cases show that significant reductions in turning time can be obtained through the use of vectored thrust.

Table 1

Best Results: Vectored Thrust, Nominal Aircraft

Case #	V_i (ft/sec)	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
1	420	662	11866	1951	10.21
3	621	690	13219	637	8.24
4	903	660	13714	-6186	8.60

The remaining two cases are less optimal results and will be presented later and in less detail. All cases in this study began at an initial altitude of 13,990 feet. All results obtained for the nominal a- are summarized in Table IV, Appendix D.

Corner Velocity. The corner velocity V_C, as given by Eq (27), is 692 feet/second (fps) at the initial altitude of 13,990 feet and varies very little over the altitude ranges encountered during the maneuvers (V_C = 700 fps at 14,700 feet, V_C = 670 fps at 12,000 feet).

The importance of the corner velocity is that it is the velocity at which the aircraft achieves its maximum turn rate. It was expected and found to be true that trajectories which stay closer to the corner velocity result in faster turning times. The maintenance of the corner velocity is shown in Figs. 2 and 3 for Case 1, Figs. 4 and 5 for Case 3, and Figs. 6 and 7 for Case 4.

Thrust Vectoring. The greater the aircraft's initial velocity, the more thrust vectoring capability was used to get the aircraft to, and keep it at, its corner velocity. As shown in Fig. 8, for a low initial velocity (Case 1, V_i = 420 fps), only slightly over 12^0 of thrust angle of attack and 3^0 of thrust sideslip are used. The reduction of the thrust angles to zero corresponds to the throttle being "chopped" to zero partway through the maneuver (Fig. 9). For a higher initial velocity (Case 3, V_i = 621 fps), Fig. 10 shows the ranges of thrust vectoring increase to 70^0 angle of attack and 8^0 sideslip as partial or full throttle is used throughout the turn (Fig. 11). At the highest initial velocity considered (Case 4, V_i = 903 fps), Fig. 12 shows an even greater increase in the range of thrust angles used, to 90^0 angle of attack and 180^0 sideslip.

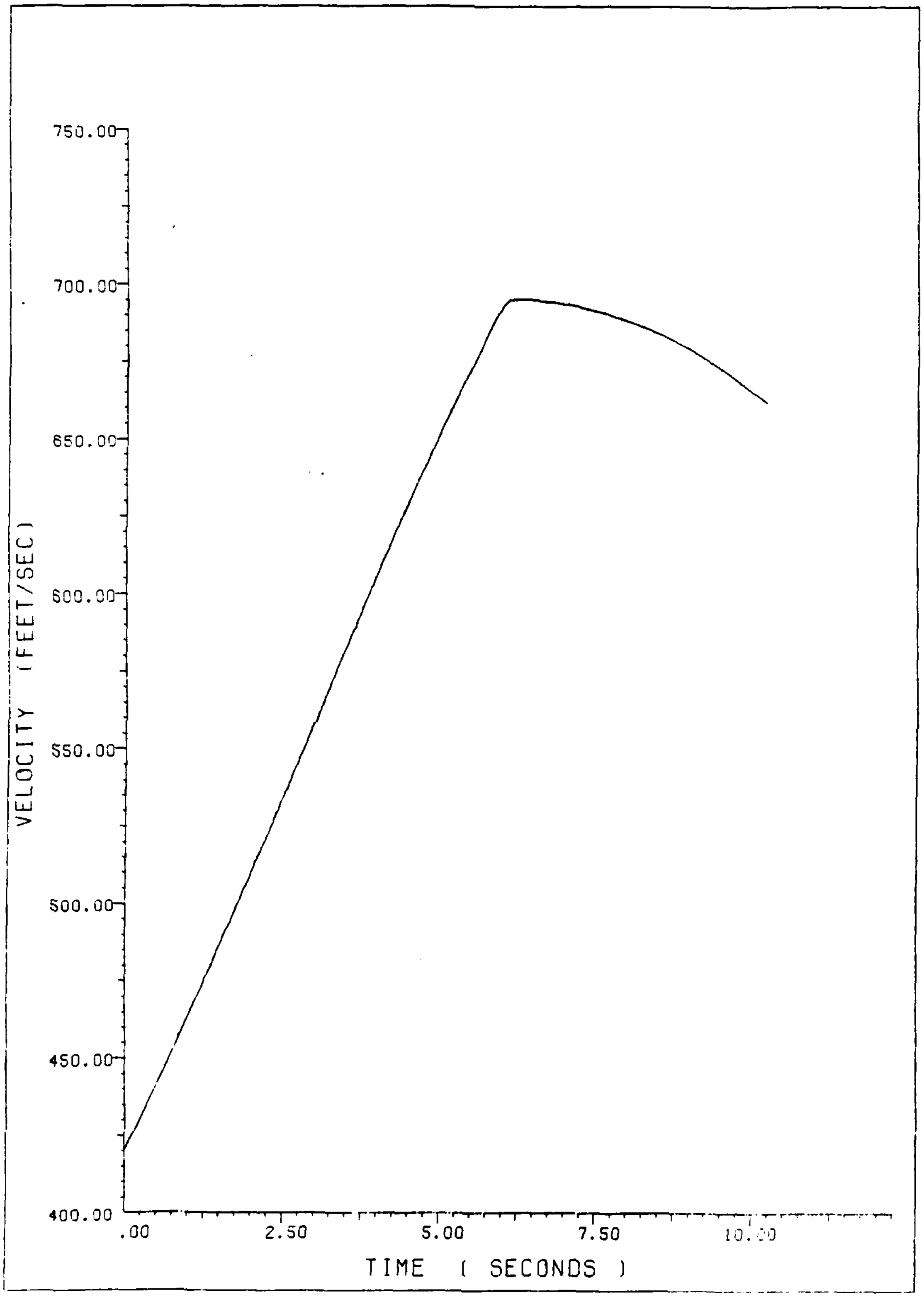


FIG 2. VELOCITY VS. TIME FOR CASE 1

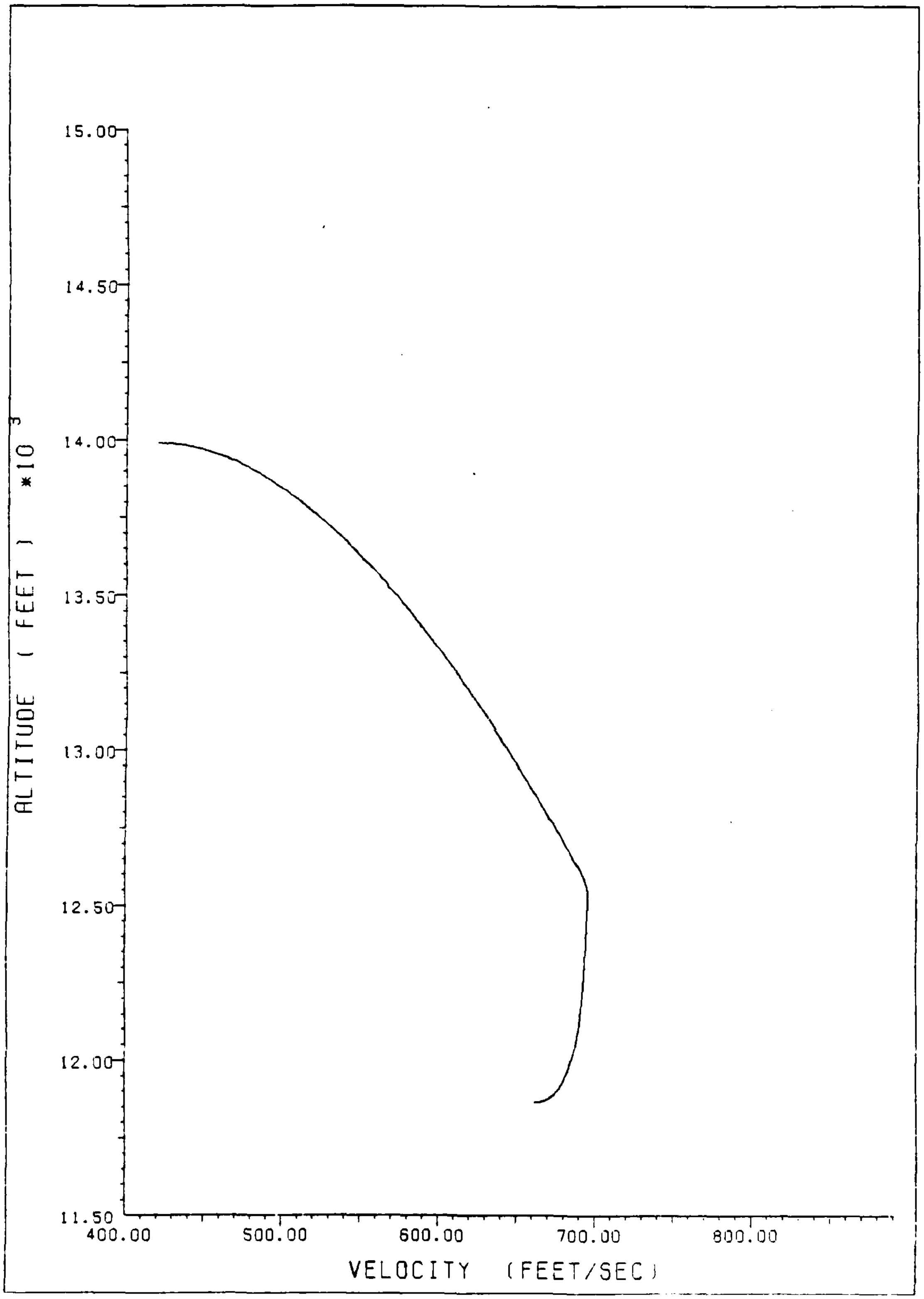


FIG 3. ALTITUDE VS. VELOCITY FOR CASE 1

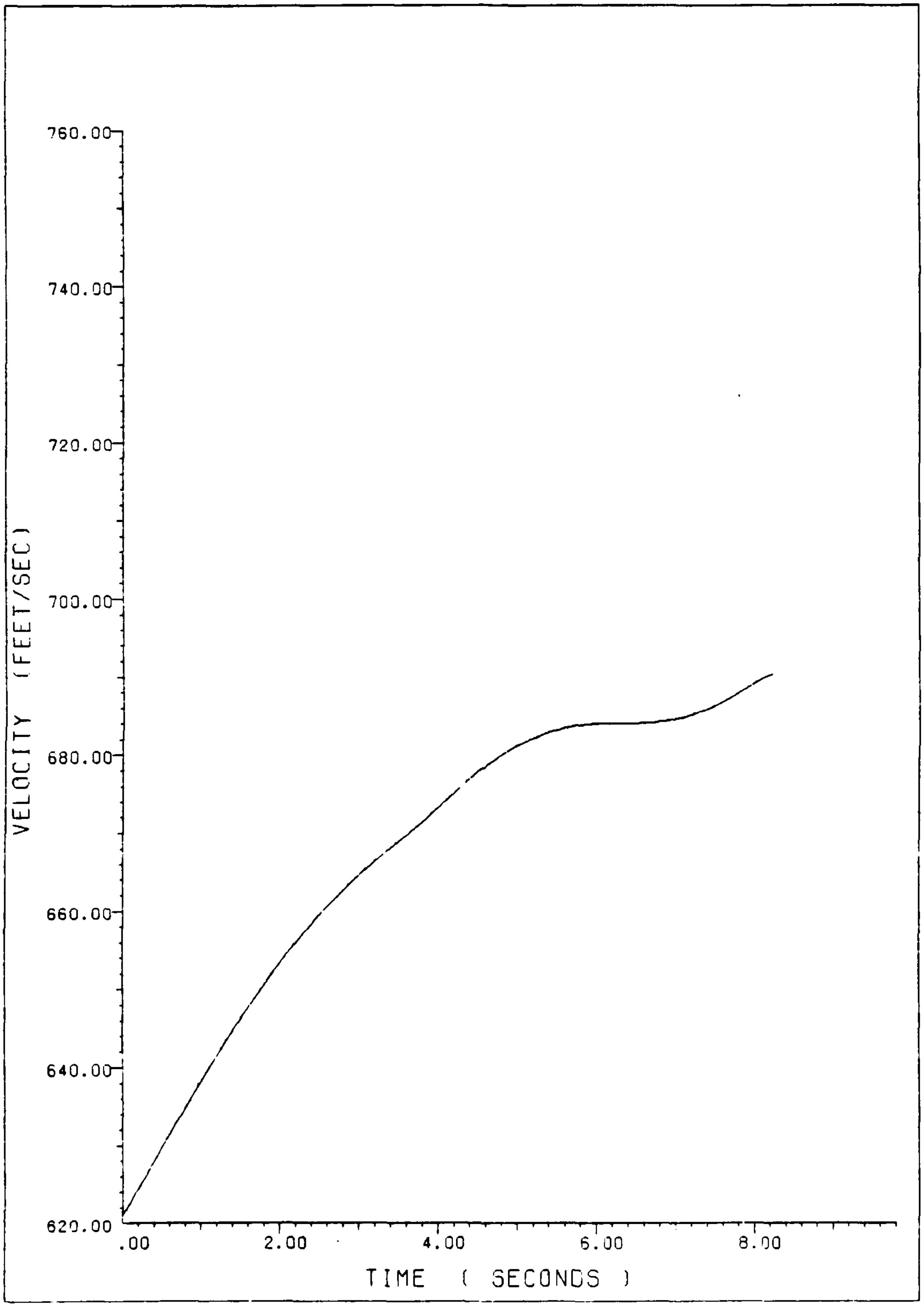


FIG 4. VELOCITY VS. TIME FOR CASE 3

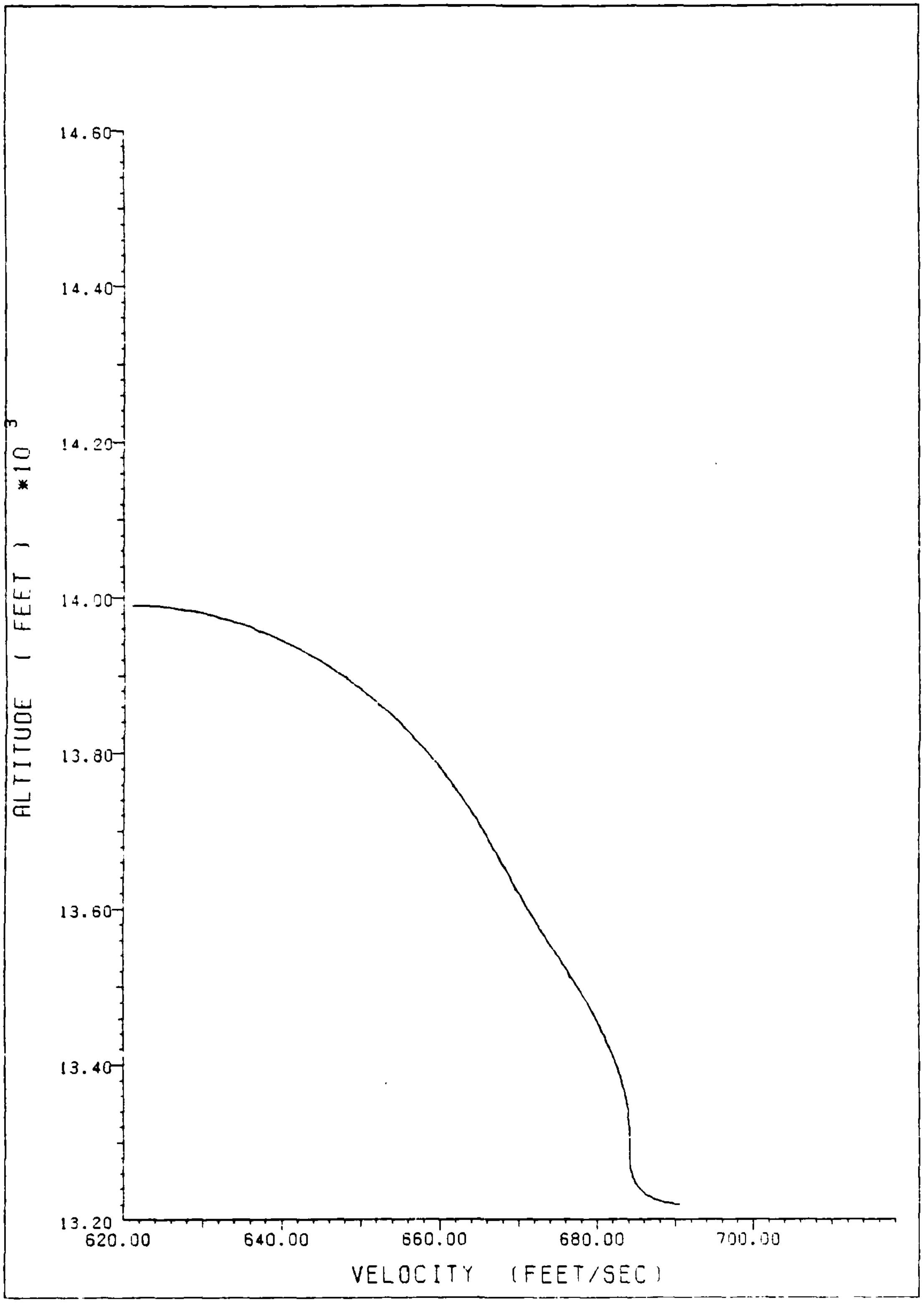


FIG 5. ALTITUDE VS. VELOCITY FOR CASE 3

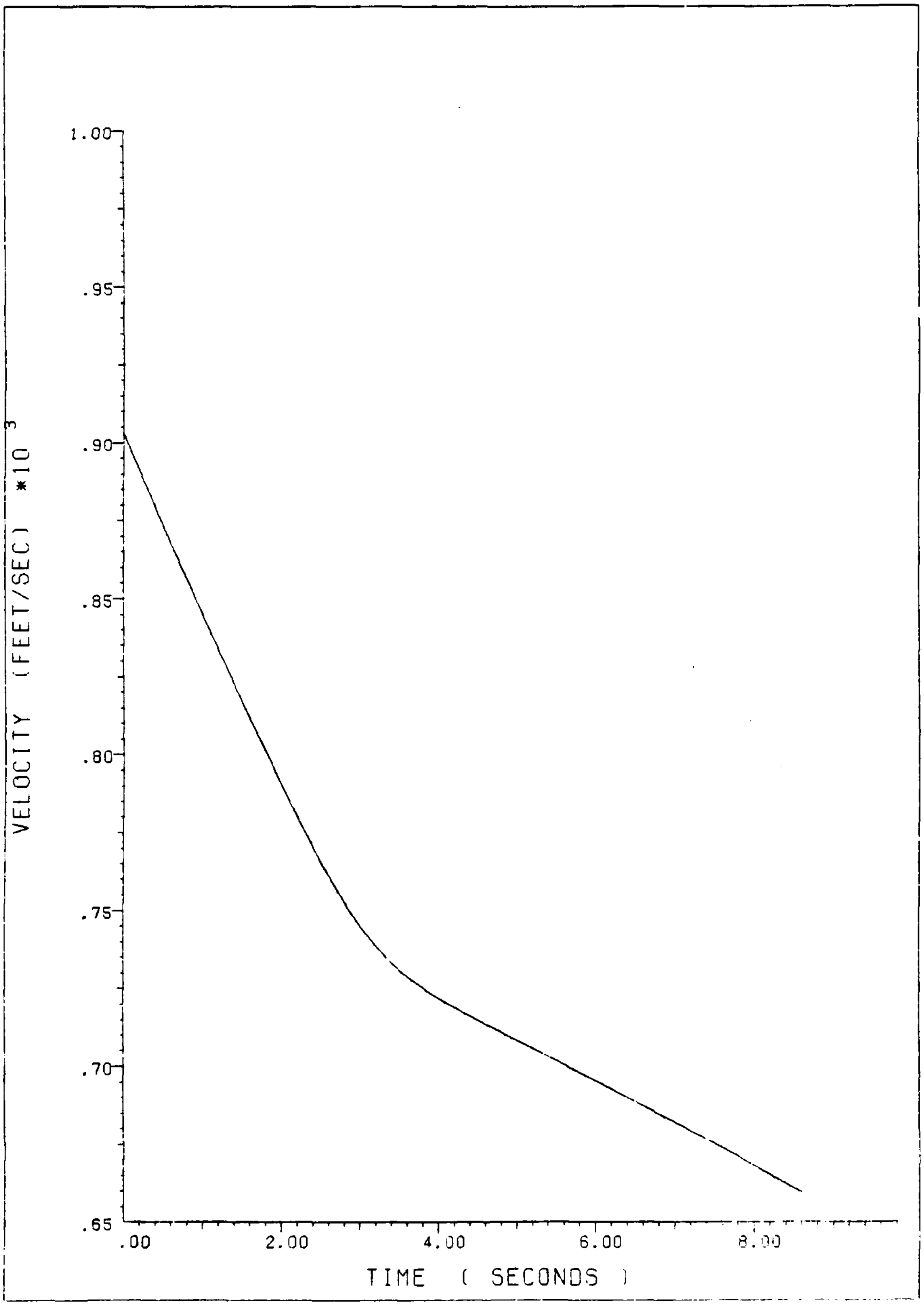


FIG 6. VELOCITY VS. TIME FOR CASE 4

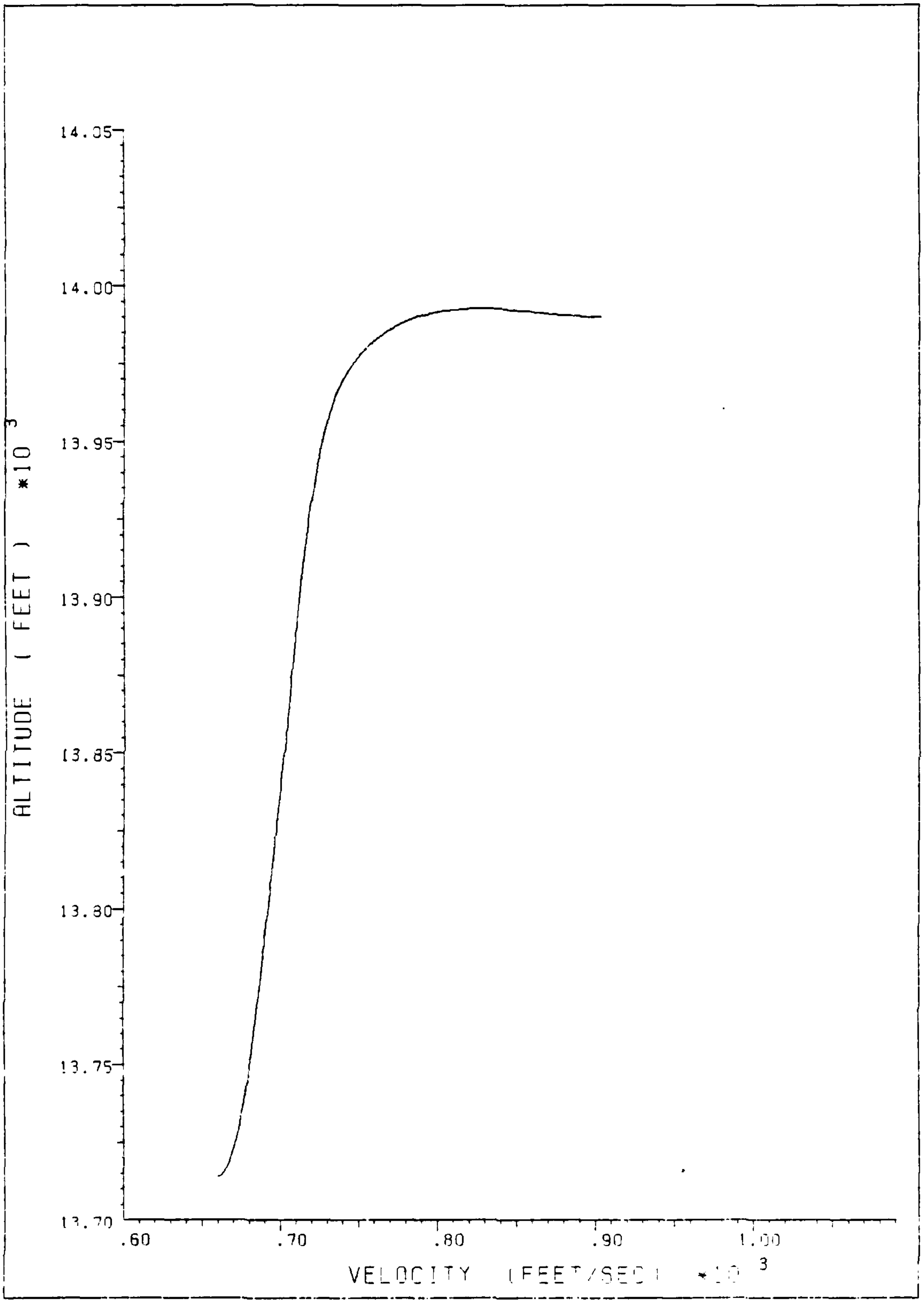


FIG 7. ALTITUDE VS. VELOCITY FOR CASE 4

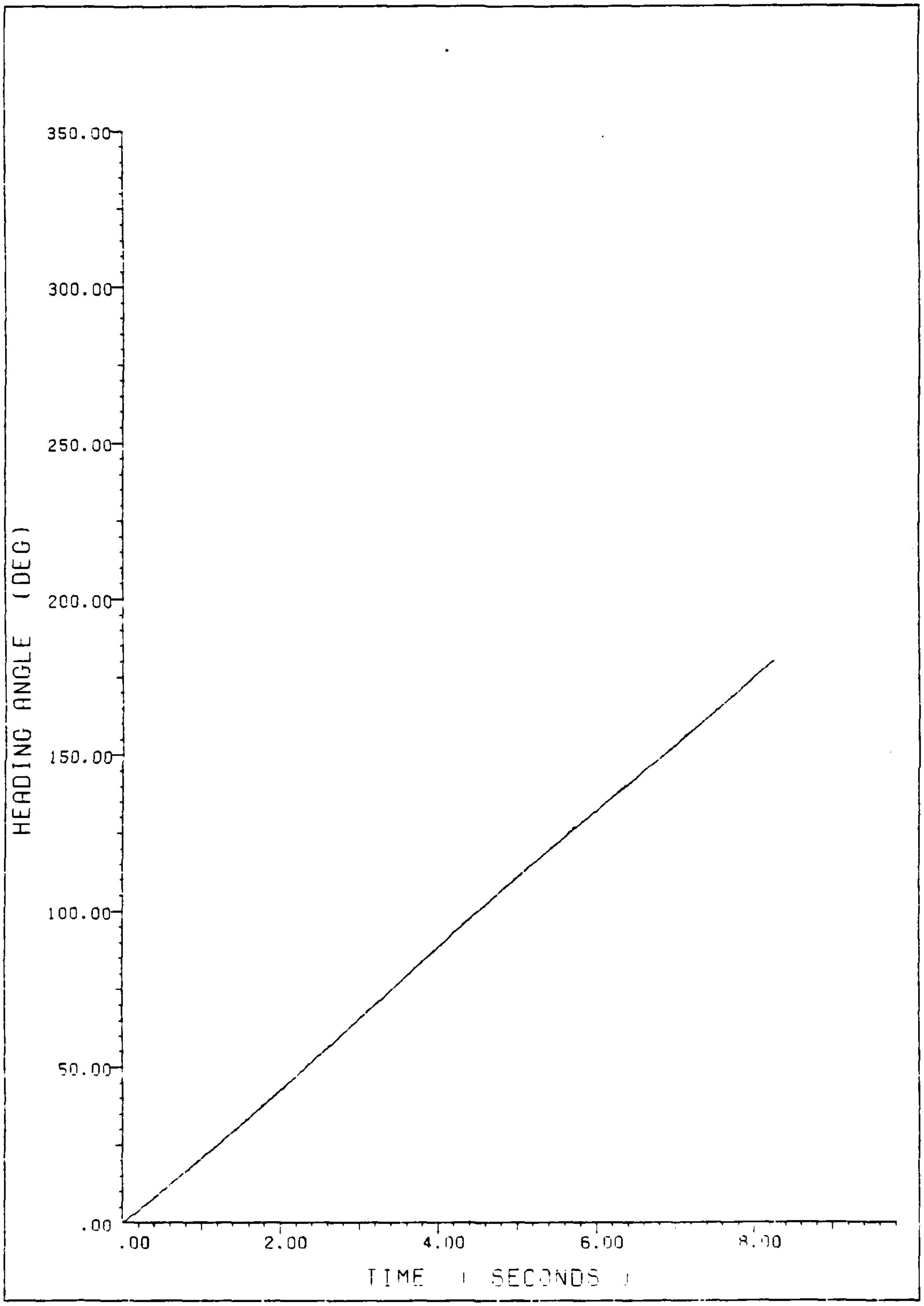


FIG 19. HEADING ANGLE VS. TIME FOR CASE 3

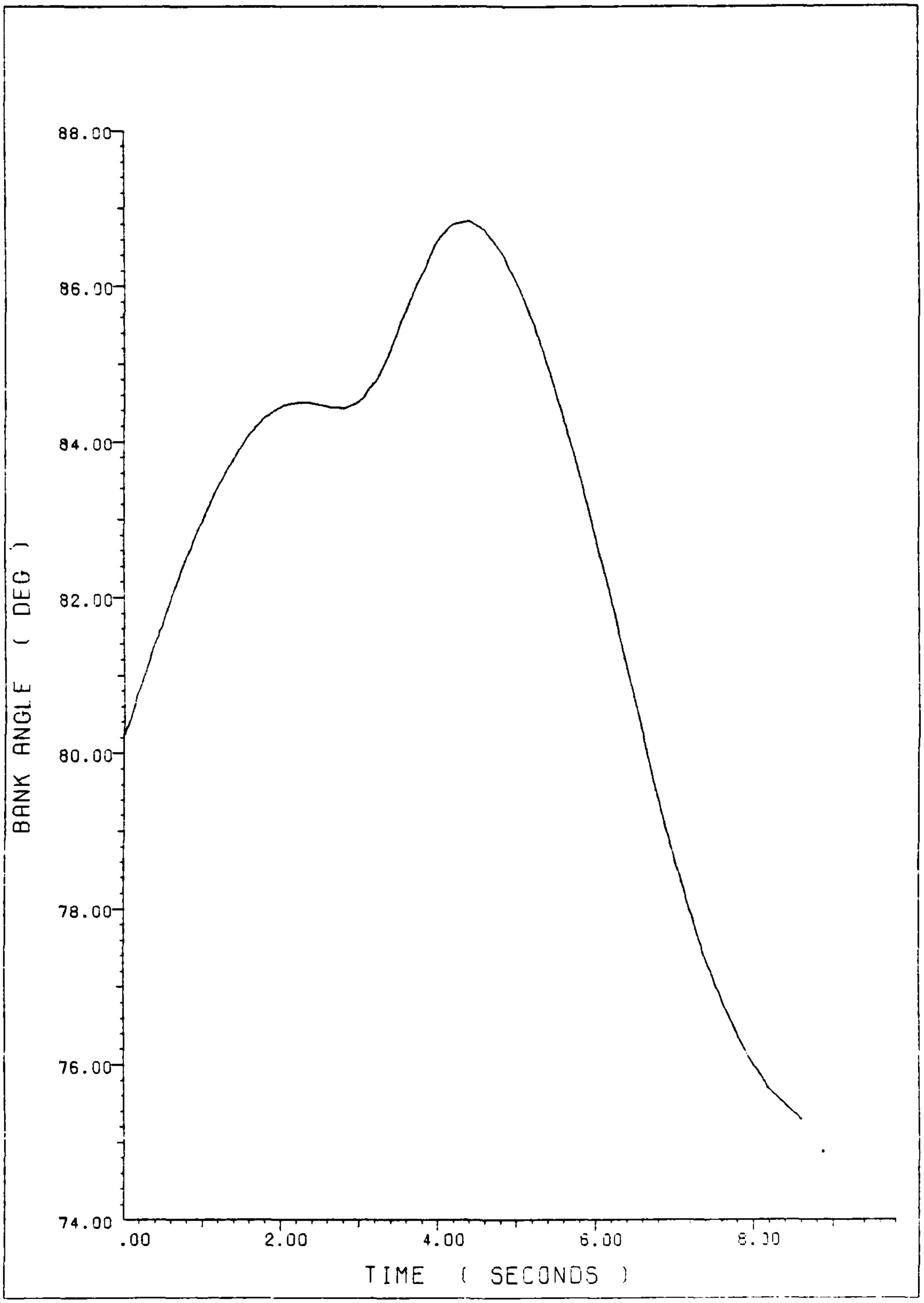


FIG 18. BANK ANGLE VS. TIME FOR CASE 4

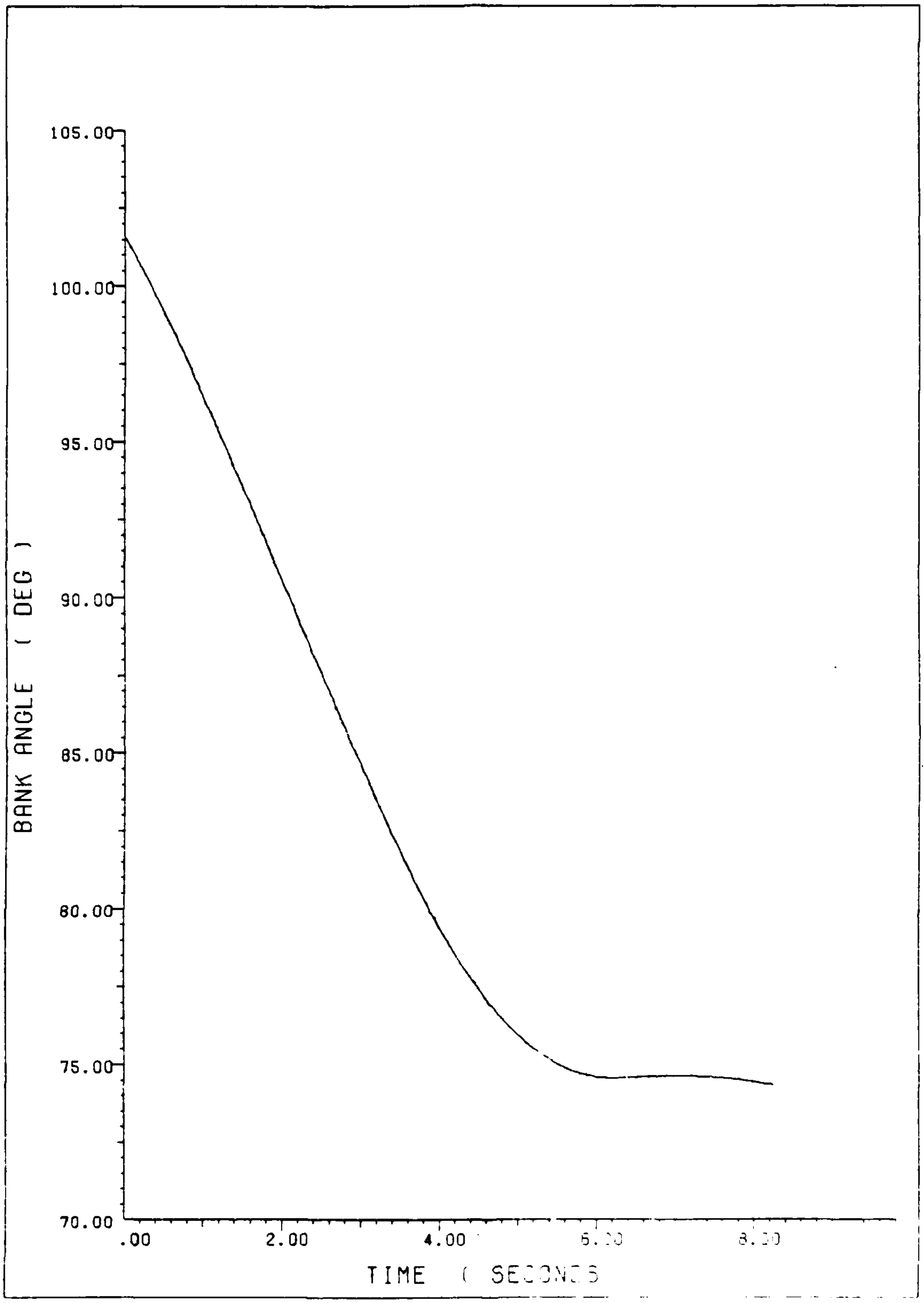


FIG 17. BANK ANGLE VS. TIME FOR CASE 3

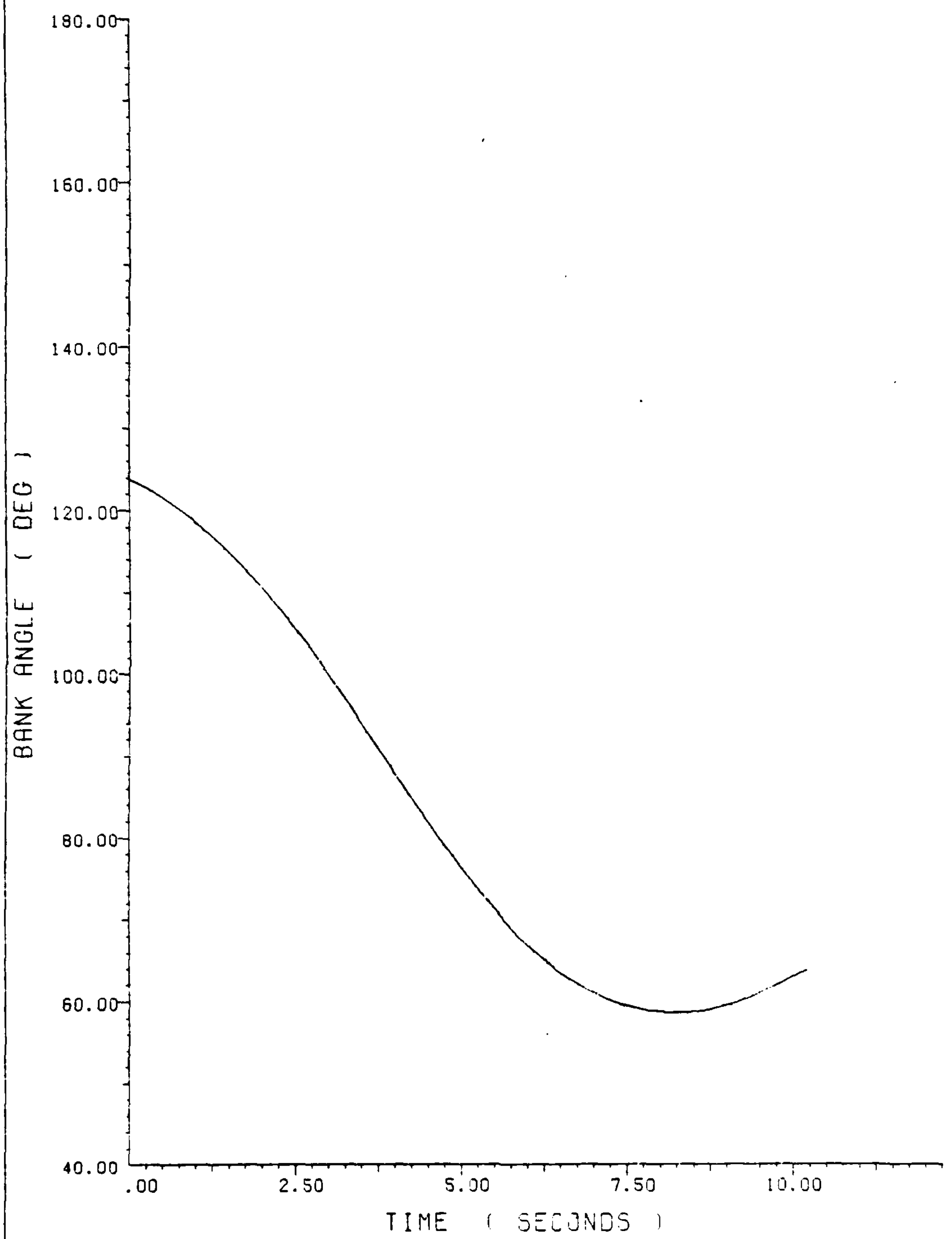


FIG 16. BANK ANGLE VS. TIME FOR CASE 1

The bank angle programs, Figs. 16 through 18, agree with the trends reported b, Well and Berger (10). For initial velocities less than the corner velocity, the aircraft banked and descended to accelerate toward V_c. For initial velocities greater than the corner velocity, the aircraft bank angle was much less as airspeed was bled off to decelerate to V_c. However, because of the thrust reversal capability, the use of the vertical plane to gain or lose airspeed was not very prominent.

<u>Heading and Flight Path Angles</u>. The heading angle progressed linearly, or very nearly linearly, with time for all cases. Fig. 19 (Case 3) is representative of all heading angle time histories.

The flight path angle time histories for Cases 1, 3, and 4 all exhibited the same parabolic shape shown in Fig. 20. The lower initial velocity case, Case 1, has a much larger peak value of negative flight path angle, approximately -33°, than Case 3 (peak value -13.5°). The larger peak value corresponds to the steeper dive made from lower initial velocity to accelerate to V_c.

Case 4, starting from a high initial velocity (V_i = 903 fps), initially shows a small period of positive flight path angle in Fig. 21. This is only maintained very briefly as the aircraft mainly used reverse thrust to bleed off airspeed.

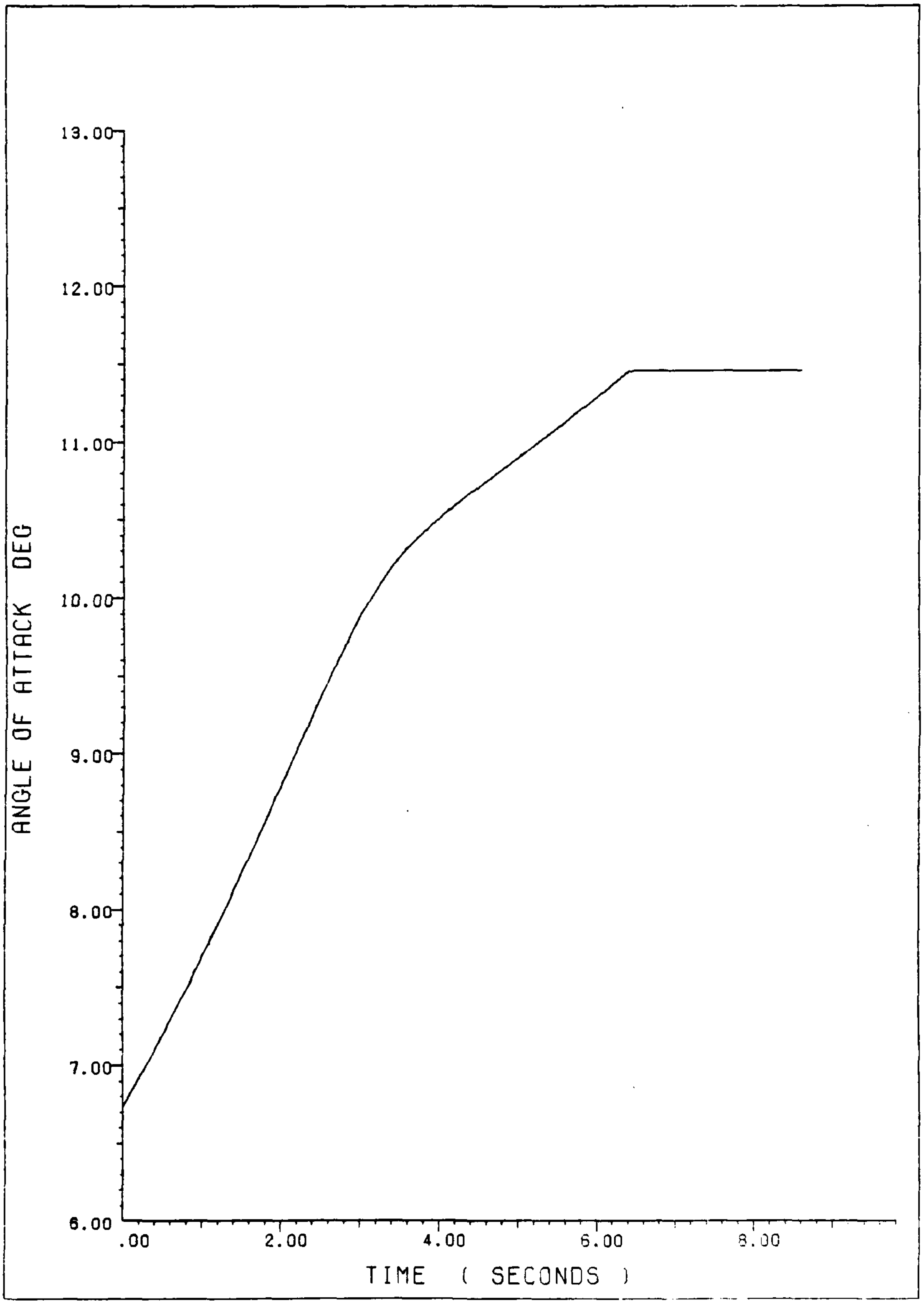


FIG 15. ANGLE OF ATTACK VS. TIME FOR CASE 4

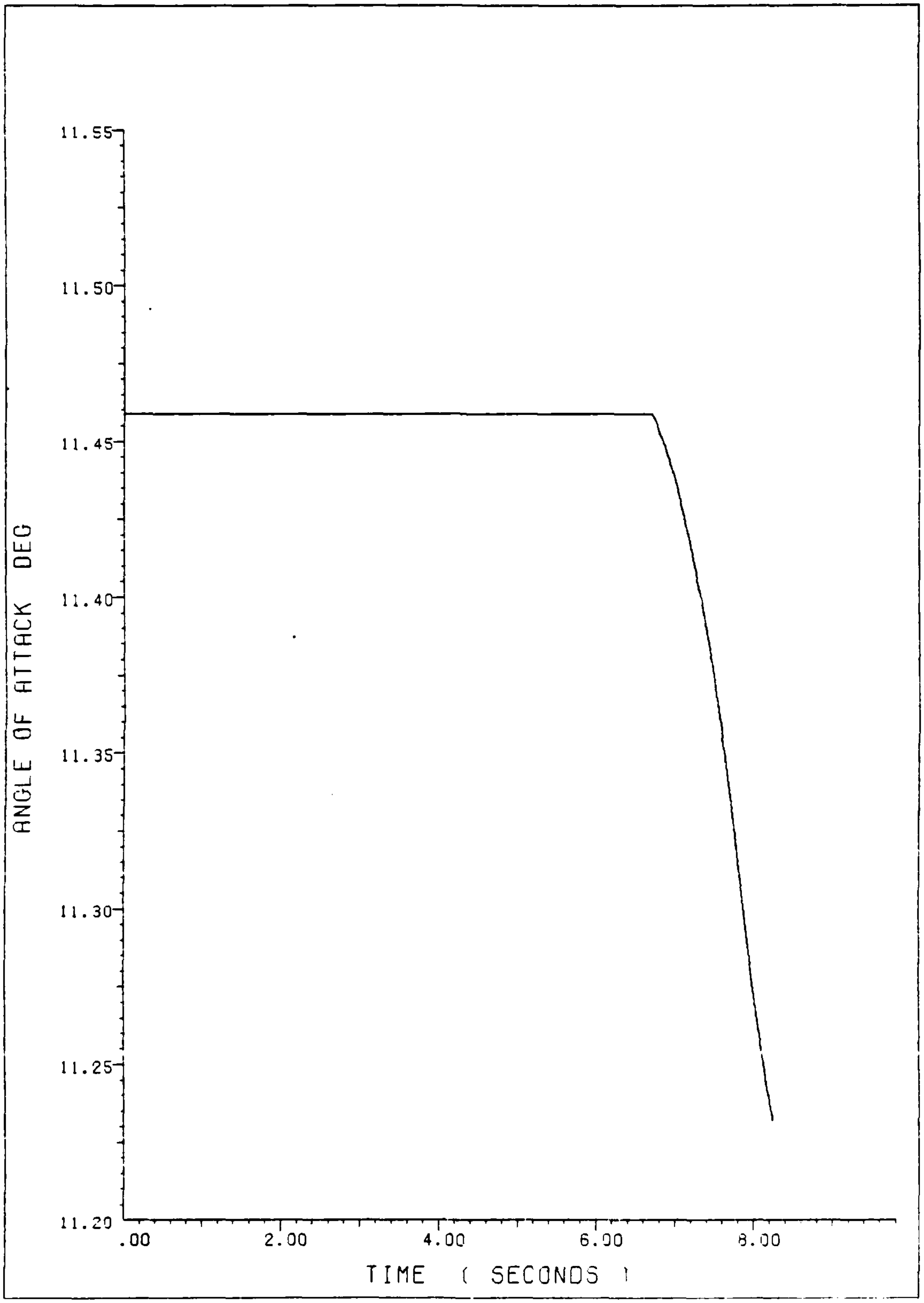


FIG 14. ANGLE OF ATTACK VS. TIME FOR CASE 3

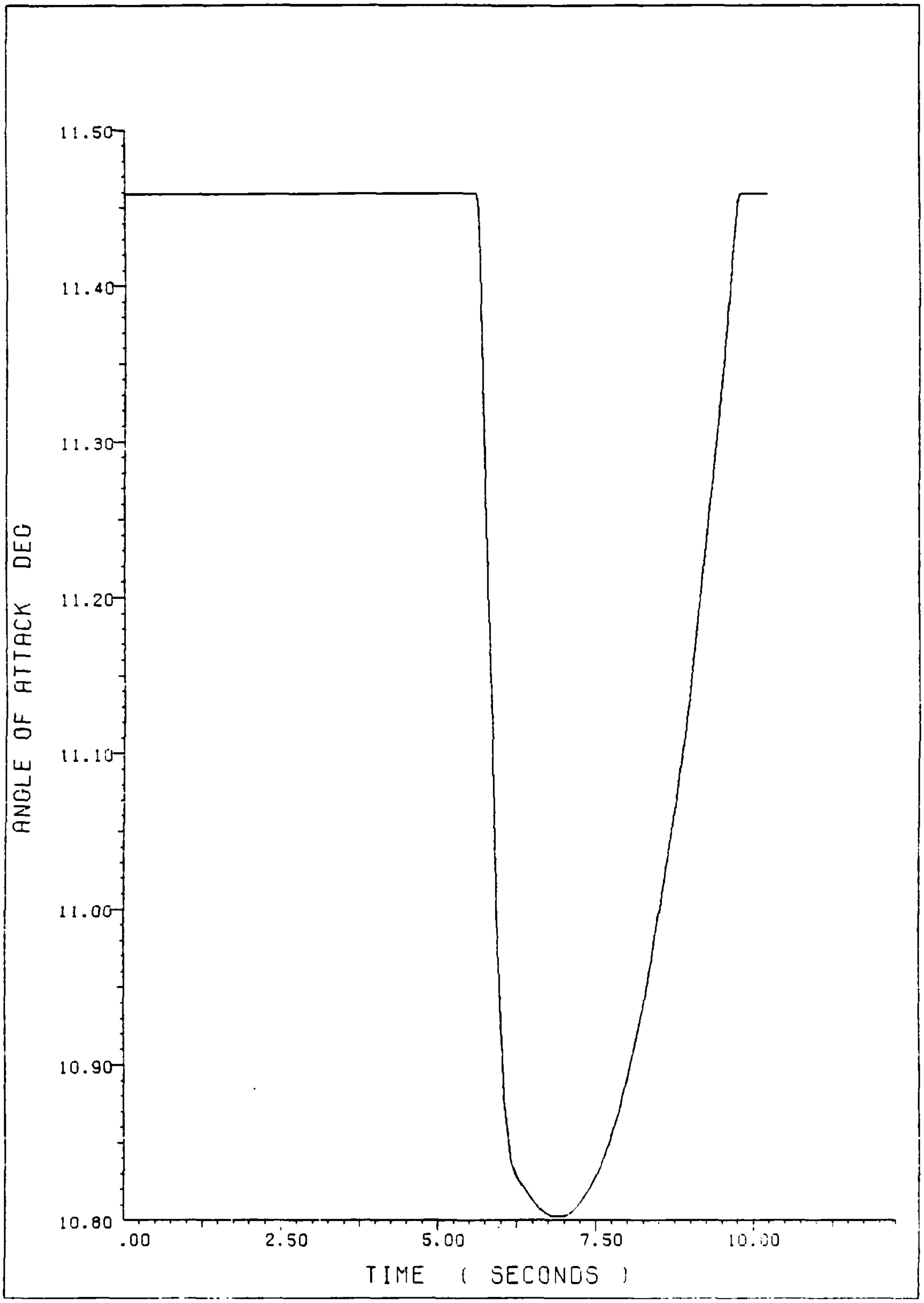


FIG 13. ANGLE OF ATTACK VS. TIME FOR CASE 1

Thrust vectoring cannot increase the aircraft's available thrust, nor can it increase the aircraft's velocity. Therefore, this capability had very little effect on the low speed cases where additional velocity was needed to significantly reduce the turning time. Consequently, only a minor improvement in turning time was found in Case 1.

The benefit of vectored thrust is apparent at the higher initial velocities. Thrust is vectored through large angles and turning times are reduced significantly over those of previous studies (4:99, 5:60, 6:59, 7:52). The most effective use of thrust vectoring is seen in Case 4 (Fig. 12, V_i = 903 fps). The maneuver was flown at full throttle with the thrust initially vectored to 180^0, effectively reversing thrust and quickly decelerating the aircraft to its corner velocity. The thrust angles were then modulated to keep the aircraft at V_c (approximately 690 fps) for the rest of the turn.

In all three cases, the thrust was directed into the turn, supplementing the aircraft's lift.

Other Controls. The remaining controls, angle of attack and bank angle, are presented in Figs. 13 through 18. The optimal program for the angle of attack, Figs. 13 through 15, is to maintain maximum angle of attack, α_{max} being dictated by the velocity as displayed in Fig. 1.

The angle of attack time history for Case 1, Fig. 13, is a good example of the optimal solution maintaining the highest possible angle of attack. The aircraft first overshot the corner velocity (Figs. 2 and 3) and the angle of attack dropped only 0.65^0 before the overshoot was halted and the aircraft decelerated back toward V_c. increasing the angle of attack until the maximum lift limit was again encountered.

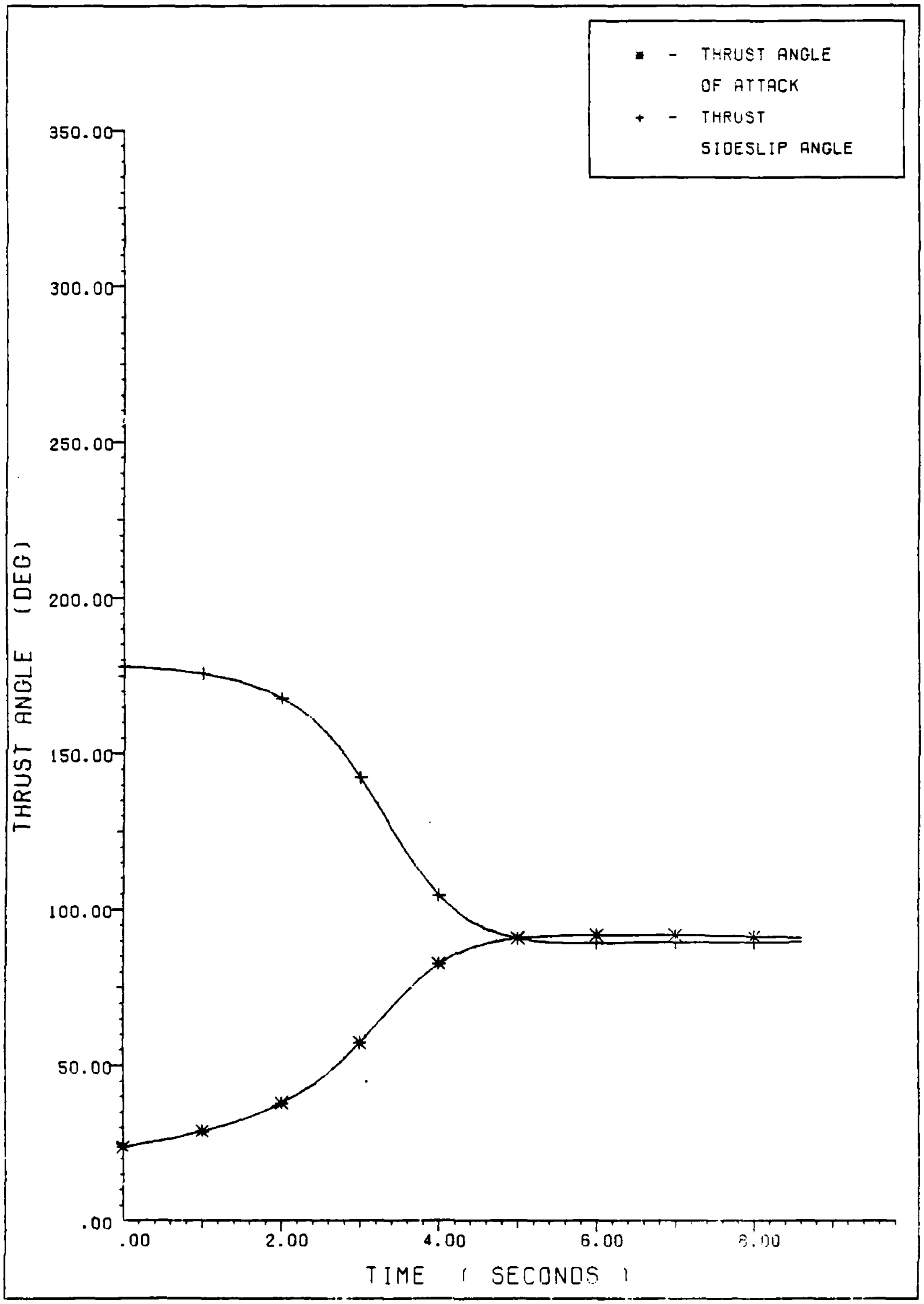


FIG 12. THRUST ANGLES VS. TIME FOR CASE 4

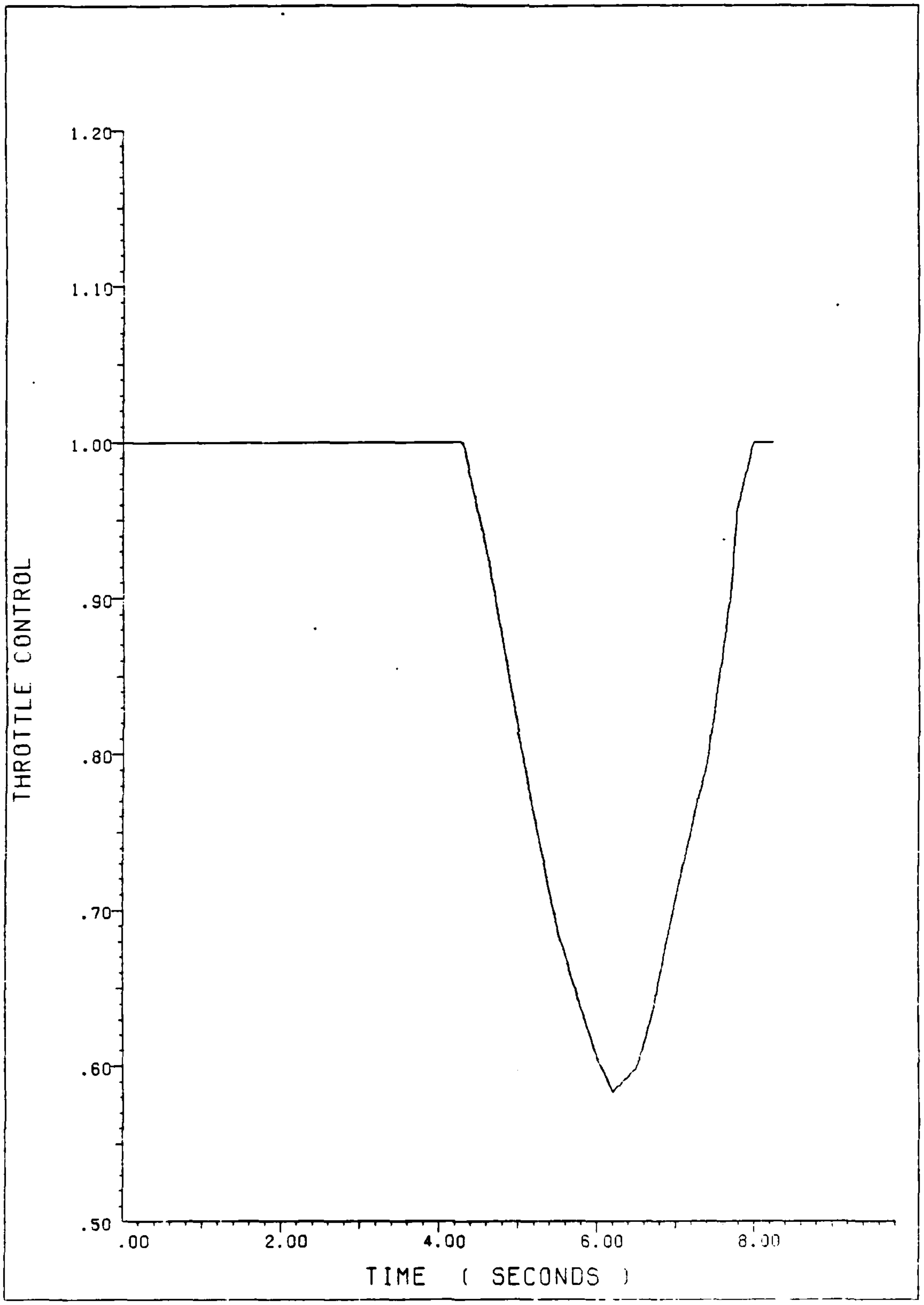


FIG 11. THROTTLE CONTROL VS. TIME FOR CASE 3

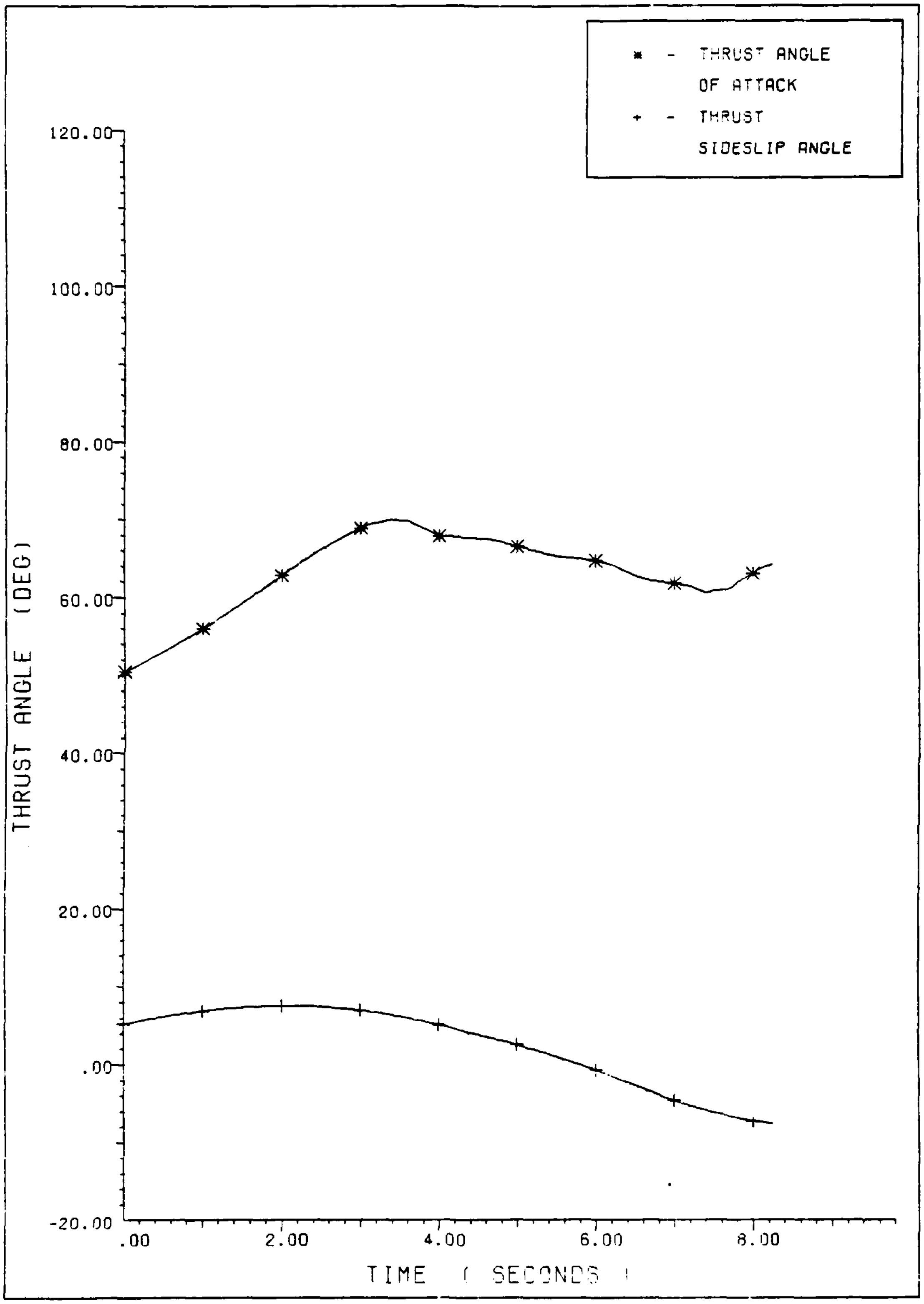


FIG 10. THRUST ANGLES VS. TIME FOR CASE 3

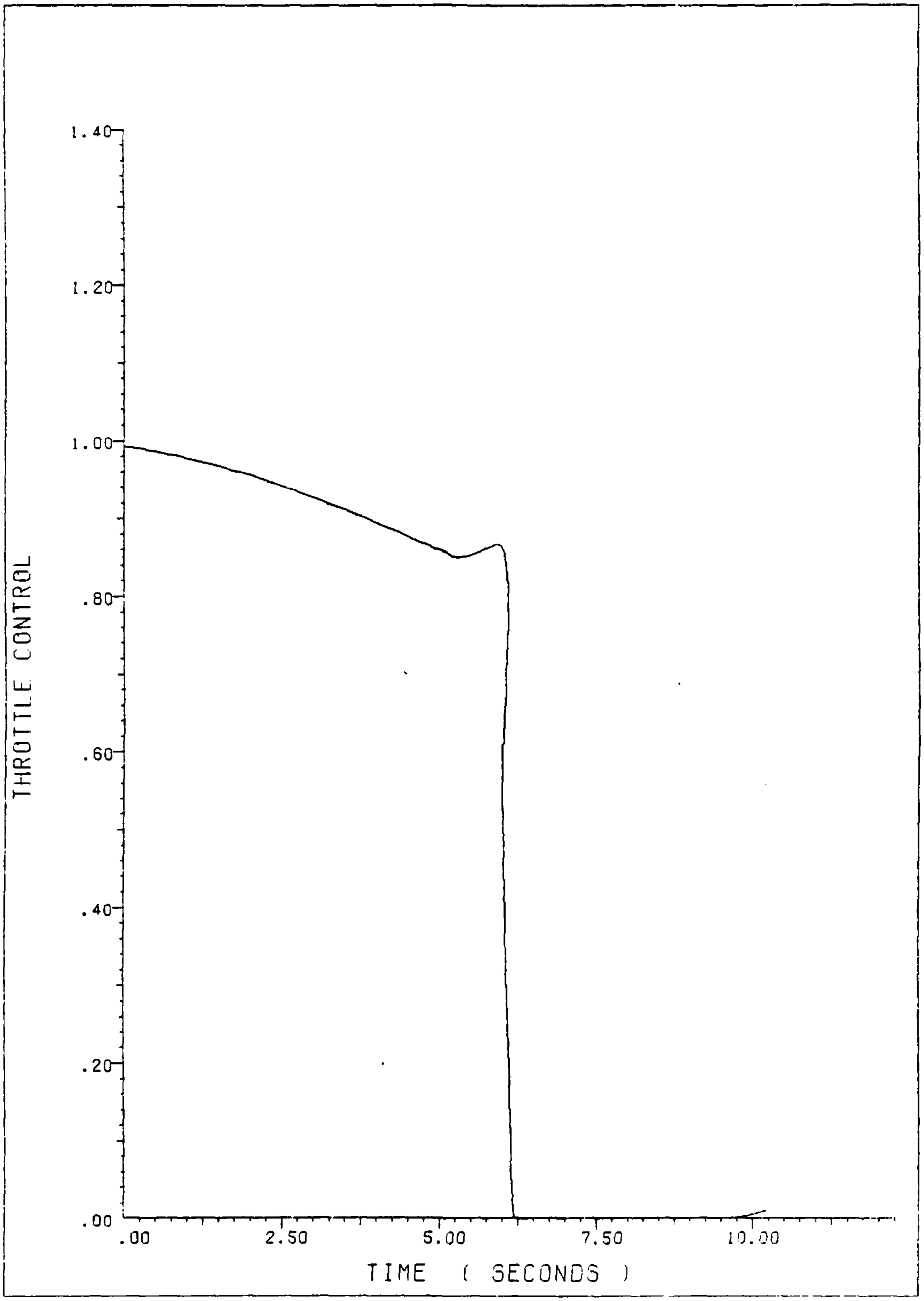


FIG 9. THROTTLE CONTROL VS. TIME FOR CASE I

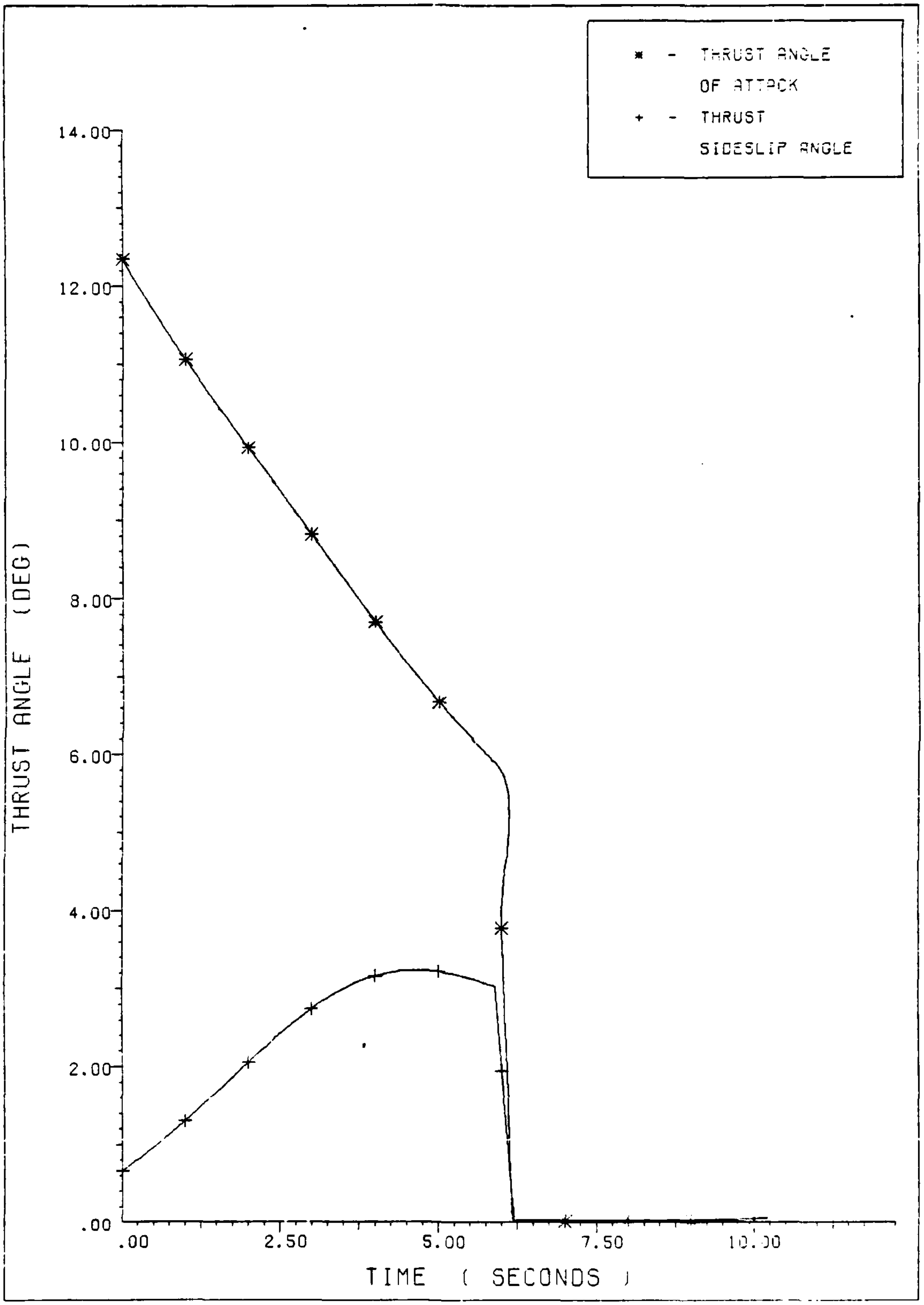


FIG 8. THRUST ANGLES VS. TIME FOR CASE 1

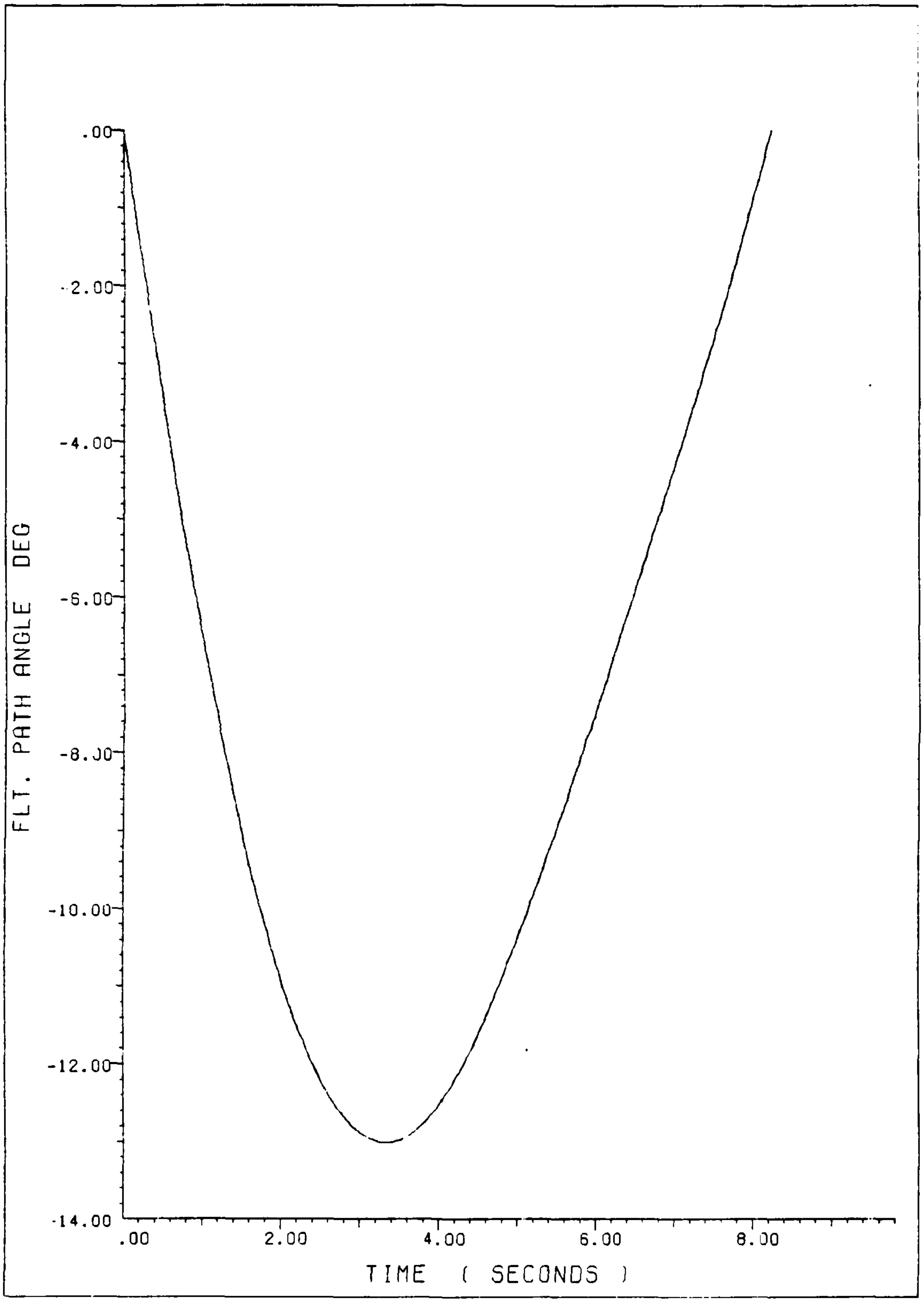


FIG 20. FLIGHT PATH ANGLE VS. TIME FOR CASE 3

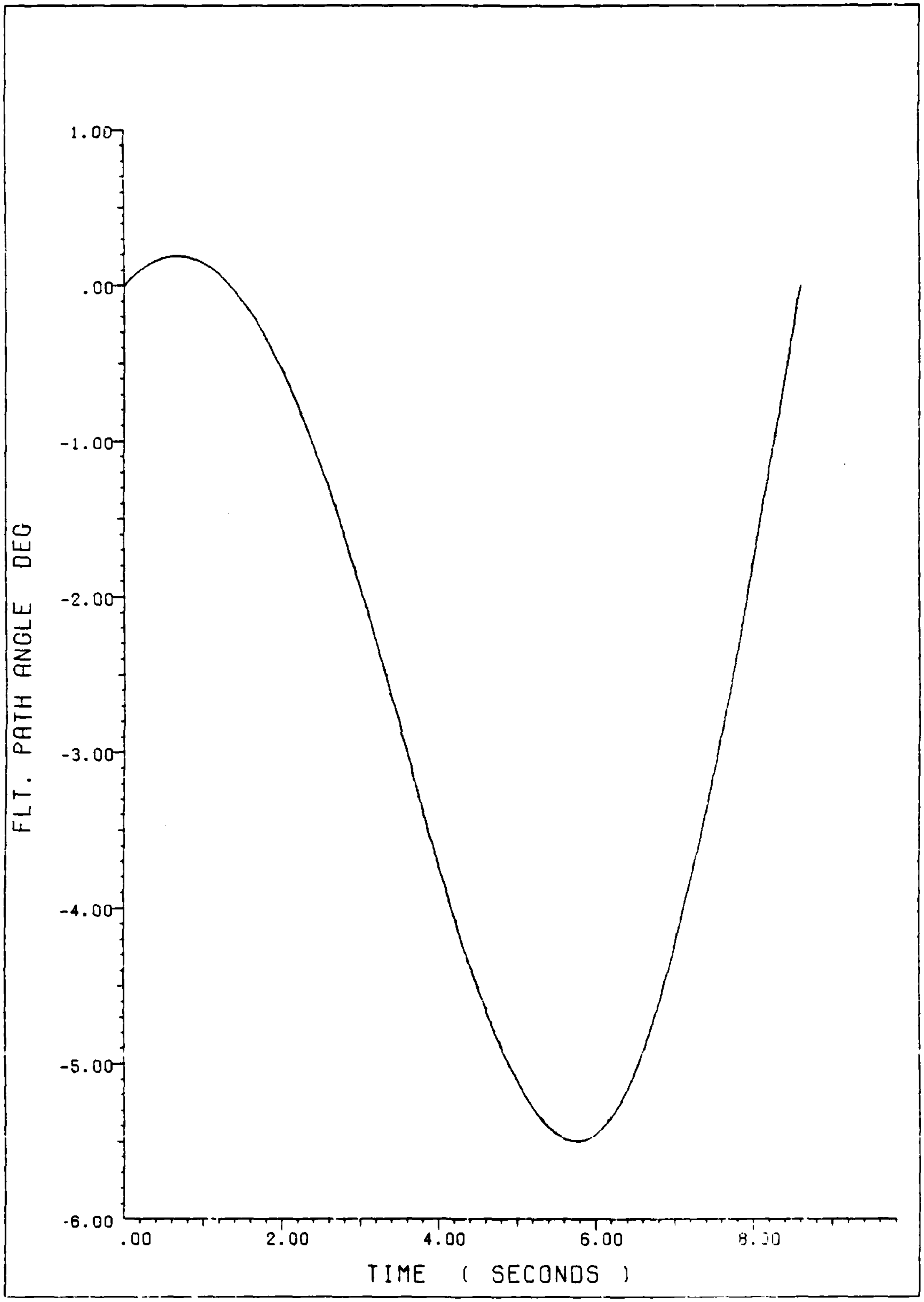


FIG 21. FLIGHT PATH ANGLE VS. TIME FOR CASE 4

Comparison Against Previous Results. The use of vectored thrust improves turning time more than the other methods of reducing turning time previously investigated (4, 5, 6, 7). The best turning times achieved with vectored thrust for the nominal aircraft are summarized in Table I and the results of previous studies are summarized in Tables VII through X, Appendix G.

A comparison of minimum turning times obtained with vectored thrust against those using direct sideforce (7:52) shows: vectored thrust 0.15 seconds faster for V_i = 420 fps (Case 1), vectored thrust 1.23 seconds faster for V_i = 621 fps (Case 3), and vectored thrust 2.08 seconds faster for V_i = 903 fps (Case 4). The benefits of thrust vectoring increase with increased initial airspeed.

Specific Energy. An important practical consideration in air-to-air combat maneuvering is an aircraft's specific energy, given by

$$E = h + \frac{V^2}{2g} \qquad (101)$$

Previous studies have used different values of g in the calculation of E. For this investigation, g was taken to be 32.131 feet/second^2, the value of the gravitational acceleration at the initial altitude.

As high an energy level as possible is desired so that kinetic and potential energies may be traded to one's advantage. Specific energy gains/losses through the maneuvers are included in the tables of results and show that faster turning times are achieved at the expense of specific energy. However, no attempt was made in this study to optimize turning performance with respect to specific energy.

Table II

Less Optimal Results: (Cases 2 and 5)

Vectored Thrust, Nominal Aircraft

(corresponding better results, Cases 1 and 4, also shown for ease of comparison)

Case #	V_i (ft/sec)	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
1	420	662	11866	1951	10.21
2	420	863	11635	6490	10.58
4	903	660	13714	-6186	8.60
5	903	738	14676	-3528	9.27

Less Optimal Results. Two other cases, Cases 2 and 5, were also run with the nominal aircraft but achieved less optimal solutions. These cases are summarized in Table II and the factors which led to slower turning times are discussed. The less optimal solutions resulted from slight changes in the initial control variable programs from those used to obtain the more optimal results. This points out the steepest-ascent method's sensitivity to initial control variable programs and its tendency to converge to the nearest, rather than most optimal, solution. Control and state variable time histories for Cases 2 and 5 are included as Figs. 22 through 34 in Appendix E.

Cases 2 and 5 are compared against Case 1 and Case 4, respectively. It is first noted that the slower turns of Cases 2 and 5 resulted in higher final specific energies, as expected.

The slower turning times are a result of the aircraft not maintaining the corner velocity. Figs. 22 and 23, for Case 2, show that the aircraft did not attempt to stay at V_c at all. Only small amounts of thrust vectoring (Fig. 24) and much higher throttle settings (Fig. 25) were used, resulting in higher velocities and much lower angles of attack (Fig. 27).

The importance of maintaining V_c is also seen in a comparison of Cases 4 and 5. In Case 5, the slower turn, the aircraft never slowed enough to reach the corner velocity (Figs. 28 and 29). While full or very nearly full throttle was used (Fig. 30), in Case 5 thrust vectoring was not used to reverse the thrust for maximum deceleration (Fig. 31). Instead, the aircraft maintained positive flight path angle (Fig. 32) and initially banked less (Fig. 33), climbing rather than reversing thrust to decelerate. This use of the vertical plane instead of thrust reversal to decelerate was clearly less optimal, resulting in higher velocities, lower angles of attack (Fig. 34), and a slower turning time.

Variation of Aircraft Characteristics

Brinson (7) varied aircraft thrust to weight ratio (T/W) and induced drag (K_1) and determined optimal turning solutions for a more realistic aircraft without any maneuver enhancement capabilities. These same parameter variations were used in this study to evaluate the benefits of thrust vectoring on a more realistic aircraft. Brinson's results (7:53) are summarized in Table X, Appendix G. The results of this study are summarized in Table III. State and control variable time histories are included as Figs. 35 through 59 in Appendix F.

Table III

Results: Vectored Thrust,

Variation of Aircraft Characteristics

Case #	V_i (ft/sec)	$(\frac{T}{W})_{max}$	K_1	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
6	420	0.75	0.05	687	11651	2260	10.92
7	420	1.50	0.22	643	11499	1198	10.47
8	621	0.75	0.05	634	13175	- 561	9.22
9	621	1.50	0.22	518	12896	-2920	9.06
10	621	1.50	0.22	660	12423	- 790	9.45
11	903	0.75	0.05	710	13589	-5245	9.79
12	903	1.50	0.22	461	13452	-9920	9.11

Influence of Initial Control Programs. The variations in aircraft characteristics were expected to result in solutions and trajectories distinctly different from those obtained for the nominal aircraft. However, when the same or similar initial (nominal) controls were used for a given initial velocity but varying aircraft characteristics, the solutions obtained had very similar control variable programs. It again seems apparent that the steepest-ascent method, as implemented here, is heavily influenced by the initial control variable program. This influence was observed for all initial velocities and casts doubts on the global optimality of the solutions obtained for variations of aircraft characteristics.

Low Initial Velocity (V_i = 420 fps). Cases 6 and 7 are solutions obtained with T/W reduced from 1.5 to 0.75 and K_1 increased from 0.05 to 0.22, respectively. No improvements in time to turn were found when the aircraft initiated the maneuver from this low velocity. Any use of thrust vectoring decreased the amount of thrust available for acceleration, thereby making it more difficult for the aircraft to get to V_c, if that is even possible, and resulting in slightly longer turning times.

Medium Initial Velocity (V_i - 621 fps). Case 8 is the solution for lower T/W (0.75) with nominal K_1 (0.05). As seen from Brinson's thrust required/available curves (7:43), this aircraft model is not thrust limited, but simply has less available thrust than the nominal aircraft. This lower available thrust resulted in the aircraft maintaining full throttle for the entire turn. Even with full throttle, the aircraft did not reach the corner velocity (Figs. 35 and 36), so the angle of attack remained at the limit imposed by $C_{L_{max}}$ ($\alpha = 11.459^0$) for the entire turn. The remaining state and control variable programs (Figs. 37 through 39) remained essentially the same as for Case 3.

A variety of comparisons may be made among the results obtained in this investigation and by Brinson (7:53). Two are presented here.

First, a comparison between Cases 3 and 8 of this study: given a low drag aircraft with thrust vectoring capability, reducing the available thrust by 50% (from T/W = 1.50, Case 3, to T/W = 0.75, Case 8) increased the time to turn by 0.98 seconds.

Second, a comparison between Case 8 of this study and Brinson's result for an aircraft with no maneuver enhancement capability: given an aircraft characterized by T/W = 0.75 and K_1 = 0.05, the addition and use of thrust vectoring capability (Case 8) enabled the aircraft to turn 0.39 seconds faster.

Cases 9 and 10 are solutions with nominal T/W (1.50) and increased K_1 (0.22). As shown in Brinson's thrust required/available curves (7:45), this aircraft model is thrust limited: it cannot attain V_c. The difference between these two cases is that Case 9 (Figs. 40 through 44) closely followed the control variable program of Case 3, while in Case 10 (Figs. 45 through 49), far less vectoring of the thrust occurred (Fig. 49). Since in both cases the aircraft was thrust limited, in neither case was V_c reached and the angle of attack remained at α_{max} = 11.459^{o} (0.2 radians). The greater use of thrust vectoring in Case 9 resulted in the aircraft slowing during the turn (Figs. 40 and 41), while in Case 10, using less thrust vectoring, the aircraft remained near its maximum attainable velocity (Figs. 45 and 46). The turn was completed 0.39 seconds faster in Case 9 than in Case 10, but at a sacrifice of specific energy, E. The final specific energy in Case 10 was 2,127 feet more than that of Case 9.

Again, several comparisons may be made among the results obtained in this investigation (Cases 3, 9, and 10) and by Brinson (7:53). Two are presented here.

First, a comparison among Cases 3, 9, and 10 of this study: given a high thrust/weight aircraft with thrust vectoring capability, the increase in induced drag (from K_1 = 0.05, Case 3, to K_1 = 0.22, Cases 9 and 10) increased the time to turn by 0.82 seconds for the faster time solution (Cases 9 and 10) increased the time to turn by 0.82 seconds for the faster time solution (Case 9) and 1.21 seconds for the higher energy solution (Case 10).

Second, a comparison among Cases 9 and 10 of this study and Brinson's results for an aircraft without maneuver enhancement capability: given an aircraft characterized by T/W = 1.50 and K_1 = 0.22, the addition and use of thrust vectoring capability resulted in a turning time 0.26 seconds faster at a sacrifice of approximately 1,900 feet of specific energy (Case 9) or 0.13 seconds slower with a slight gain (approximately 225 feet) of specific energy (Case 10).

High Initial Velocity (V_i = 903 fps). The control variable programs for Case 11 (reduced T/W, Figs. 50 through 54) and Case 12 (increased K_1, Figs. 55 through 59) are very similar to those for the nominal aircraft. The maneuver was flown with full throttle and the thrust initially reversed to decelerate toward V_c. The thrust angles then went to 90^0, directing the thrust into the turn to supplement lift. The higher T/W and higher K_1 in Case 12 allowed the aircraft to decelerate much more rapidly than the low thrust, low drag model of Case 11. In Case 12, the corner velocity was reached and the turn completed 0.68 seconds faster than in Case 11. In Case 11, V_c was never reached.

The benefits of thrust vectoring are clear when these optimal controls and trajectories are compared against those of an aircraft without maneuver enhancement capabilities (Table X, Appendix G). For the low thrust, low drag aircraft, the use of vectored thrust reduced the time to turn by 1.03 seconds. The turning time for the high thrust, high drag aircraft model was reduced by 1.40 seconds.

Limited Thrust Angles. Since it is unlikely that an aircraft would have an unlimited range through which to vector thrust (2, 3), Cases 13 through 16 were run to examine the effects of limiting the thrust angles. Thrust angle of attack was limited to 20° with a thrust sideslip limit of either 0° or 10°. Nominal aircraft characteristics were used. The results are summarized in Table VI, Appendix D.

Overall, the results obtained with limited thrust angles are inconclusive. This is perhaps due to the small number of runs made and solutions obtained, but more likely a result of the influence of the initial control variable programs, as previously discussed. A more thorough investigation needs to be made to examine the effects of realistically limited thrust angles.

When the maneuver was started from the medium initial velocity, 621 fps, thrust vectoring was used but the thrust angle of attack limit of 20° was always encountered and in effect. When the thrust sideslip was limited to 0°, this limit also remained in effect throughout the maneuver. However, when the thrust sideslip limit was 10°, the limit was never reached. In both cases, limiting the thrust angles increased the time to turn.

For the high speed cases, (V_i = 903 fps), the turning times also increased when the thrust angles were limited. When thrust was vectored, the thrust angle of attack limit of 20° was encountered. However, the fastest turns were obtained when no thrust vectoring was used at all.

VII. Conclusions and Recommendations

Three major conclusions are drawn based upon the results discussed in the previous section.

1. Major reductions in turning time can be realized through the use of vectored thrust. The higher the initial velocity, the greater the reduction in turning time. For an initial velocity of 903 feet/second at an initial altitude of 13,990 feet, the use of vectored thrust reduces the time to turn by 2 - 2.5 seconds.

2. Thrust vectoring was used to supplement the aircraft's lift by directing the thrust into the turn.

3. The steepest ascent method, as implemented in this study, is heavily influenced by the choice of initial control variable programs. More optimal solutions may be obtained with different starting control variable programs.

Although the results obtained in this study may not reflect the most optimal uses of thrust vectoring, even these less-than-optimum solutions show the dramatic improvements in turning time realizable with vectored thrust. Any more optimal solutions would only further advance this point.

Five recommendations are offered for future research.

1. The previous studies of Johnson (5), Finnerty (6), and Brinson (7) should be revised to include the singular control, or type S constraint, when the aircraft velocity is equal to the corner velocity. Their methods of reducing turning time may be more effective than previously thought. Such an effort would be simplified by the fact that all three previous studies used the same technique, with Finnerty (6) and Brinson (7) modifying the computer program developed by Johnson (5) to suit their particular investigations.

2. This study's results may be improved if the implementation of the type S constraint is revised to include corner constraint requirements on a control as it enters and exists a singular arc (13, 14).

3. The question of why the gradient does not go to zero when the optimal solution possesses a singular arc remains to be answered (9, 13:29).

4. In future investigations of thrust vectoring to reduce turning time, the problem may be simplified by keeping constant full throttle and maximum angle of attack (as dictated by the velocity, Fig. 1). This would reduce the number of controls to three and allow solutions to be displayed in the three-dimensional control space.

5. The effects of limited thrust angles should be investigated, particularly in light of the upcoming flight test program (3) and the current status of two-dimensional engine nozzles (2).

Appendix A : Adjoint Matrix

The adjoint matrix $\{\underline{F}\}$ is defined as

$$\{\underline{F}(t)\} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial X_1}\right)^* & \cdots\cdots & \left(\frac{\partial f_1}{\partial X_n}\right)^* \\ \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial X_1}\right) & \cdots\cdots & \left(\frac{\partial f_n}{\partial X_n}\right)^* \end{bmatrix} \tag{38}$$

Letting $F_{i_j} = \partial f_i / \partial X_j$, the non-zero elements of $\{\underline{F}\}$ are

$$F_{14} = \cos\gamma\cos\chi \tag{102}$$

$$F_{15} = -V\cos\gamma\sin\chi \tag{103}$$

$$F_{16} = -V\sin\gamma\cos\chi \tag{104}$$

$$F_{24} = \cos\gamma\sin\chi \tag{105}$$

$$F_{25} = V\cos\gamma\cos\chi \tag{106}$$

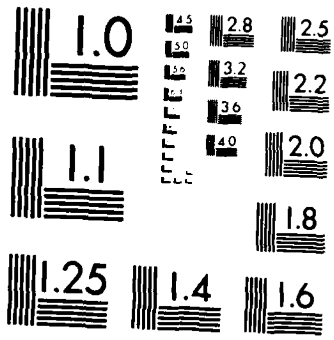
$$F_{26} = -V\sin\gamma\sin\chi \tag{107}$$

$$F_{34} = \sin\gamma \tag{108}$$

$$F_{36} = V\cos\gamma \tag{109}$$

AD-A151 693 MINIMUM TIME TURNS USING VECTORED THRUST(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING G L SCHNEIDER DEC 84 AFIT/GAE/AA/84D-24

UNCLASSIFIED F/G 1/2 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

$$F_{43} = gC_1C_2C_3V^2(C_{D_0} + K_1C_{L_\alpha}^2\alpha^2)\{(1-C_2h)\exp(C_3 - 1)\} \tag{110}$$

$$F_{44} = -2gC_1\sigma V(C_{D_0} + K_1C_{L_\alpha}^2\alpha^2) \tag{111}$$

$$F_{46} = -g\cos\gamma \tag{112}$$

$$F_{53} = \frac{-g}{\cos\gamma} C_1C_2C_3VC_{L_\alpha}\alpha\sin\mu\{(1 - C_2h)\exp(C_3 - 1)\} \tag{113}$$

$$F_{54} = \frac{g}{\cos\gamma} \{\sigma C_1C_{L_\alpha}\alpha\sin\mu - \frac{\pi}{V^2}\left(\frac{T}{W}\right)_{max}(\cos\varepsilon\sin\nu\cos\mu + \sin\varepsilon\sin\mu)\} \tag{114}$$

$$F_{56} = \frac{g\sin\gamma}{V\cos^2\gamma}\{\left(\frac{T}{W}\right)_{max}\pi(\cos\varepsilon\sin\nu\cos\mu + \sin\varepsilon\sin\mu) + C_1\sigma V^2C_{L_\alpha}\alpha\sin\mu\} \tag{115}$$

$$F_{63} = -Vg\ C_1C_2C_3C_{L_\alpha}\alpha\cos\mu(1 - C_2h)\exp(C_3 - 1) \tag{116}$$

$$F_{64} = C_1\sigma C_{L_\alpha}\alpha\cos\mu + \frac{1}{V^2}\{\cos\gamma - g\left(\frac{T}{W}\right)_{max}\pi(\sin\varepsilon\cos\mu - \cos\varepsilon\sin\nu\sin\mu)\} \tag{117}$$

$$F_{66} = \frac{g\sin\gamma}{V} \tag{118}$$

where

$$\sigma = (1 - C_2 h)^{C_3}$$

$$C_1 = \frac{\rho_0 S_W}{2W}$$

$$C_2 = \frac{(n-1)}{n} \frac{g_0}{RT_0}$$

$$C_3 = \frac{1}{n-1}$$

Appendix B : Gradient Matrix

The gradient matrix $\{\underline{G}\}$ is defined as

$$\{\underline{G}\ (t)\} = \begin{bmatrix} \left(\frac{\partial f_1}{\partial U_1}\right)^* & \cdots\cdots & \left(\frac{\partial f_1}{\partial U_m}\right)^* \\ \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial U_1}\right)^* & \cdots\cdots & \left(\frac{\partial f_n}{\partial U_m}\right)^* \end{bmatrix} \tag{39}$$

Letting $G_{i_j} = \partial f_i/\partial U_j$, the non-zero elements of $\{\underline{G}\}$ are

$$G_{42} = -2gC_1\sigma V^2 K_1 C_{L_\alpha}^{\ 2}\ \alpha \tag{119}$$

$$G_{43} = g\left(\frac{T}{W}\right)_{max} \cos\varepsilon\cos\nu \tag{120}$$

$$G_{44} = -g\left(\frac{T}{W}\right)_{max} \pi\sin\varepsilon\cos\nu \tag{121}$$

$$G_{45} = -g\left(\frac{T}{W}\right)_{max} \pi\cos\varepsilon\sin\nu \tag{122}$$

$$G_{51} = \frac{g}{V\cos\gamma}\{\left(\frac{T}{W}\right)_{max} \pi(\sin\varepsilon\cos\mu - \cos\varepsilon\sin\nu\sin\mu) \tag{123}$$

$$+\ C_1\sigma V^2 C_{L_\alpha} \alpha\cos\mu\}$$

$$G_{52} = \frac{g}{V\cos\gamma} C_1\sigma V^2 C_{L_\alpha} \sin\mu \tag{124}$$

$$G_{53} = \frac{g}{V\cos\gamma}\{(\frac{T}{W})_{max} (\cos\varepsilon\sin\nu\cos\mu + \sin\varepsilon\sin\mu)\} \tag{125}$$

$$G_{54} = \frac{g}{V\cos\gamma} \{(\frac{T}{W})_{max} \pi(\sin\mu\cos\varepsilon - \sin\varepsilon\sin\nu\cos\mu)\} \tag{126}$$

$$G_{55} = \frac{g}{V\cos\gamma} (\frac{T}{W})_{max} \pi\cos\varepsilon\cos\nu\cos\mu \tag{127}$$

$$G_{61} = \frac{-g}{V} \{(\frac{T}{W})_{max} \pi(\sin\varepsilon\sin\mu + \cos\varepsilon\sin\nu\cos\mu)$$

$$+ C_1\sigma V^2 C_{L_\alpha} \alpha\sin\mu\} \tag{128}$$

$$G_{62} = \frac{g}{V} C_1\sigma V^2 C_{L_\alpha} \cos\mu \tag{129}$$

$$G_{63} = \frac{g}{V} (\frac{T}{W})_{max} (\sin\varepsilon\cos\mu - \cos\varepsilon\sin\nu\sin\mu) \tag{130}$$

$$G_{64} = \frac{g}{V} (\frac{T}{W})_{max} \pi(\cos\varepsilon\cos\mu + \sin\varepsilon\sin\nu\sin\mu) \tag{131}$$

$$G_{65} = \frac{-g}{V} (\frac{T}{W})_{max} \pi\cos\varepsilon\cos\nu\sin\mu \tag{132}$$

where

$$\sigma = (1 - C_2 h)^{C_3}$$

$$C_1 = \frac{\rho_0 S_W}{2W}$$

$$C_2 = \frac{(n-1)}{n} \quad \frac{g_0}{RT_0}$$

$$C_3 = \frac{1}{n-1}$$

Appendix C : State Variable Inequality Constraint

Substitution of Eqs (82) and (83) into Eq (98) gives the following form of state variable inequality constraints

$$\dot{S} = \frac{(0.2)C_2C_3\sigma V^3 \sin\gamma}{(1-C_2h)}$$

$$- (0.4)g\sigma V \{(\frac{T}{W})_{max}\ \pi\cos\varepsilon\cos\nu - \sin\gamma$$

$$- C_1\sigma V^2 (C_{D_0} + K_1 C_{L_\alpha}^{\ 2}\ \alpha^2)\} \qquad (133)$$

where

$$C_1 = \frac{\rho_0 S_W}{2W}$$

$$C_2 = \frac{(n-1)}{n} \quad \frac{g_0}{RT_0}$$

$$C_3 = \frac{1}{n-1}$$

After rearranging, the elements of the required partial derivatives, $\partial\dot{S}/\partial\underline{U}$ and $\partial\dot{S}/\partial\underline{X}$, are

$$\frac{\partial \dot{S}}{\partial U_1} = \frac{\partial \dot{S}}{\partial \mu} = 0 \tag{134}$$

$$\frac{\partial \dot{S}}{\partial U_2} = \frac{\partial \dot{S}}{\partial \alpha} = \frac{2(62286.8)}{VW} g\rho_0 S_W K_1 C_{L_\alpha}^2 \alpha \tag{135}$$

$$\frac{\partial \dot{S}}{\partial U_3} = \frac{\partial \dot{S}}{\partial \pi} = \frac{-2(62286.8)}{V^3\sigma} g\left(\frac{T}{W}\right)_{max} \cos\varepsilon\cos\nu \tag{136}$$

$$\frac{\partial \dot{S}}{\partial U_4} = \frac{\partial \dot{S}}{\partial \varepsilon} = \frac{2(62286.8)}{V^3\sigma} g\left(\frac{T}{W}\right)_{max} \pi\sin\varepsilon\cos\nu \tag{137}$$

$$\frac{\partial \dot{S}}{\partial U_5} = \frac{\partial \dot{S}}{\partial \nu} = \frac{2(62286.8)}{V^3\sigma} g\left(\frac{T}{W}\right)_{max} \pi\cos\varepsilon\sin\nu \tag{138}$$

$$\frac{\partial \dot{S}}{\partial X_1} = \frac{\partial \dot{S}}{\partial X} = 0 \tag{139}$$

$$\frac{\partial \dot{S}}{\partial X_2} = \frac{\partial \dot{S}}{\partial Y} = 0 \tag{140}$$

$$\frac{\partial \dot{S}}{\partial X_3} = \frac{\partial \dot{S}}{\partial h} = \frac{(62286.8)C_2^2C_3(C_3 + 1)\sin\gamma}{V\{(1 - C_2h)\exp(C_3 + 2)\}}$$

$$- \frac{2(62286.8)gC_2C_3\{\left(\frac{T}{W}\right)_{max} \pi\cos\varepsilon\cos\nu - \sin\gamma\}}{V^3\{(1 - C_2h)\exp(C_3 + 1)\}} \tag{141}$$

$$\frac{\partial \dot{S}}{\partial X_4} = \frac{\partial \dot{S}}{\partial V} = \frac{62286.8}{V^2} \{g(6\{(\frac{T}{W})_{max} \frac{\pi \cos\varepsilon \cos\nu - \sin\gamma\}}{V^2 \sigma}$$

$$+ C_1 \quad \{C_{D_0} + K_1 C_{L_\alpha}^{\;2} \alpha^2\}) \quad - \quad \frac{C_2 C_3 \sin\gamma}{\{(1 - C_2 h)\exp(C_3 + 1)\}} \} \qquad (142)$$

$$\frac{\partial \dot{S}}{\partial X_5} = \frac{\partial \dot{S}}{\partial \chi} = 0 \qquad (143)$$

$$\frac{\partial \dot{S}}{\partial X_6} = \frac{\partial \dot{S}}{\partial \gamma} = \frac{(62286.8)\cos\gamma}{V\sigma} \quad \{ \frac{C_2 C_3}{(1 - C_2 h)} \quad + \quad \frac{2g}{V^2} \} \qquad (144)$$

Appendix D : Summary of Results

Table IV

Summary of Results:

Vectored Thrust, Nominal Aircraft

(all initial altitudes 13,990 feet)

Case #	V_i (ft/sec)	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
1	420	662	11866	1951	10.21
2	420	863	11635	6490	10.58
3	621	690	13219	637	8.24
4	903	660	13714	-6186	8.60
5	903	738	14676	-3528	9.27

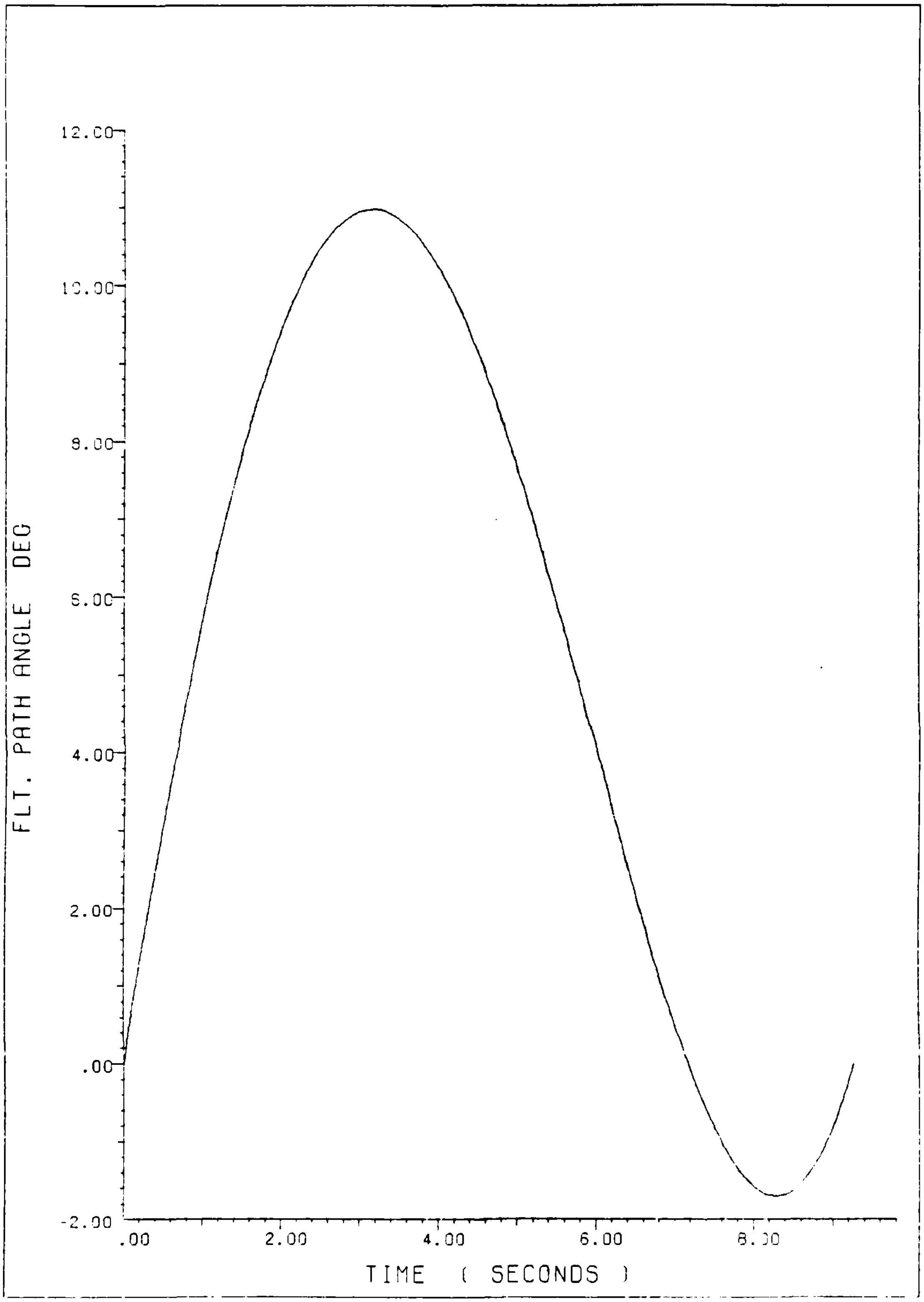


FIG 32. FLIGHT PATH ANGLE VS. TIME FOR CASE 5

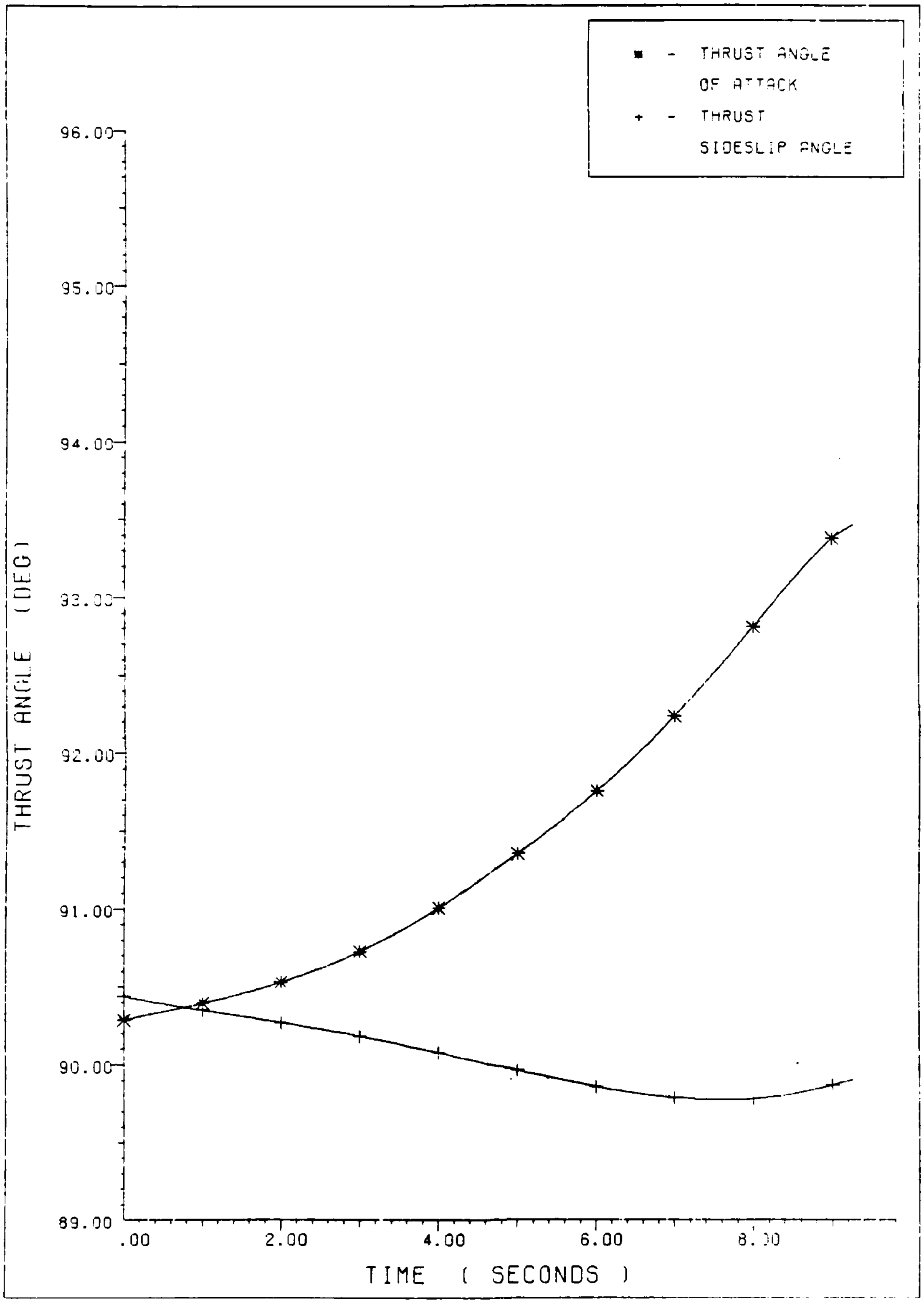


FIG 31. THRUST ANGLES VS. TIME FOR CASE 5

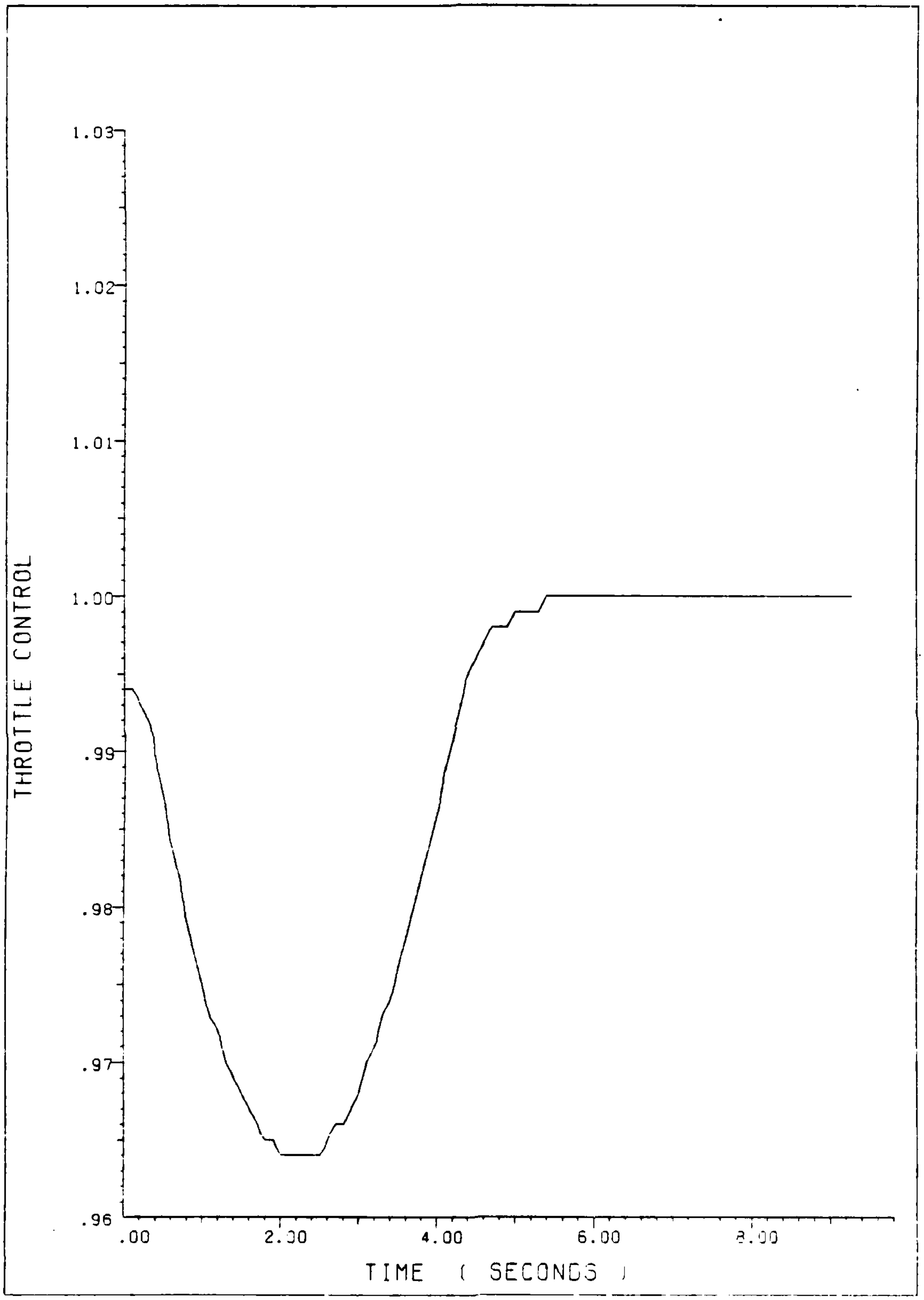


FIG 30. THROTTLE CONTROL VS. TIME FOR CASE 5

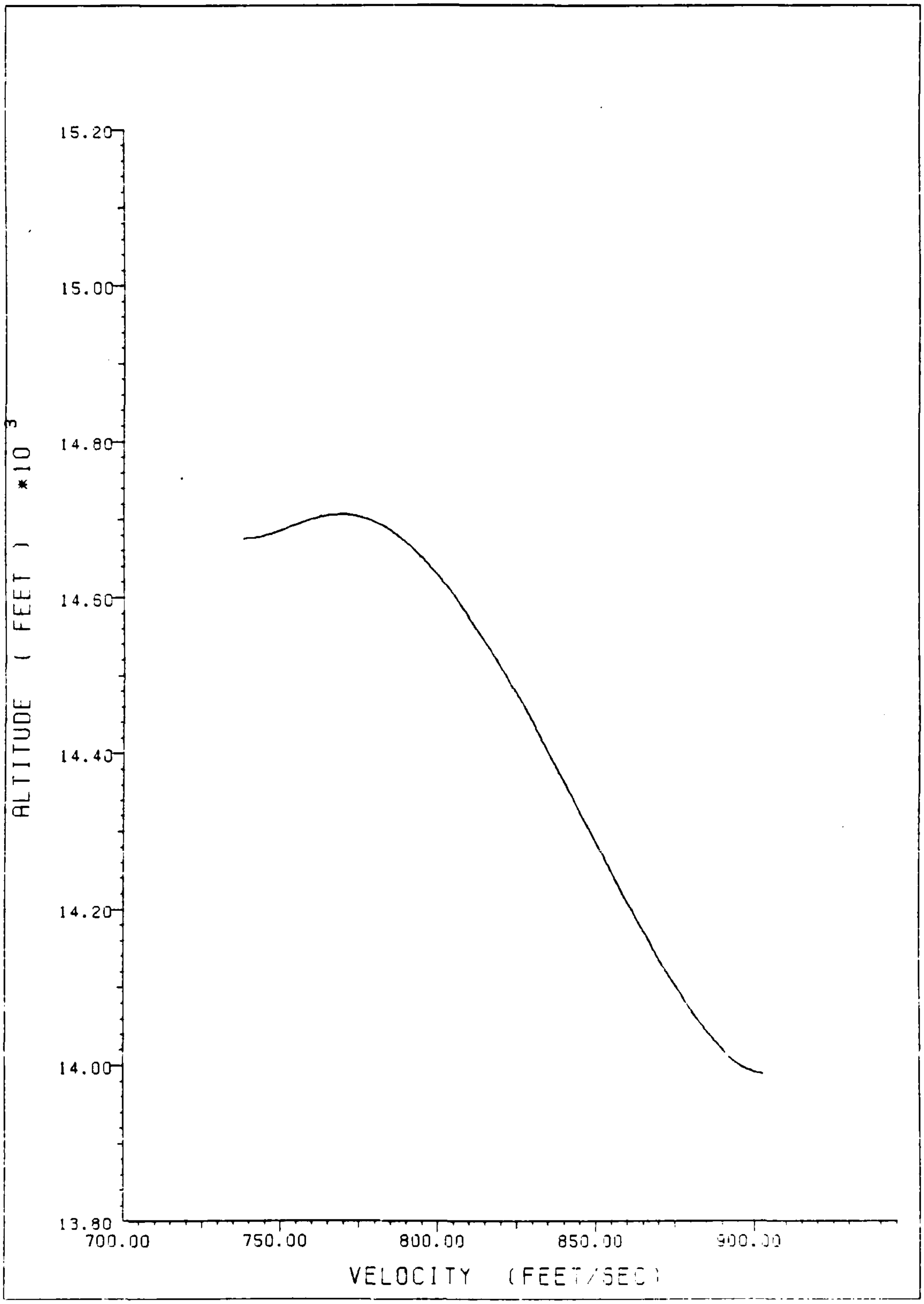


FIG 29. ALTITUDE VS. VELOCITY FOR CASE 5

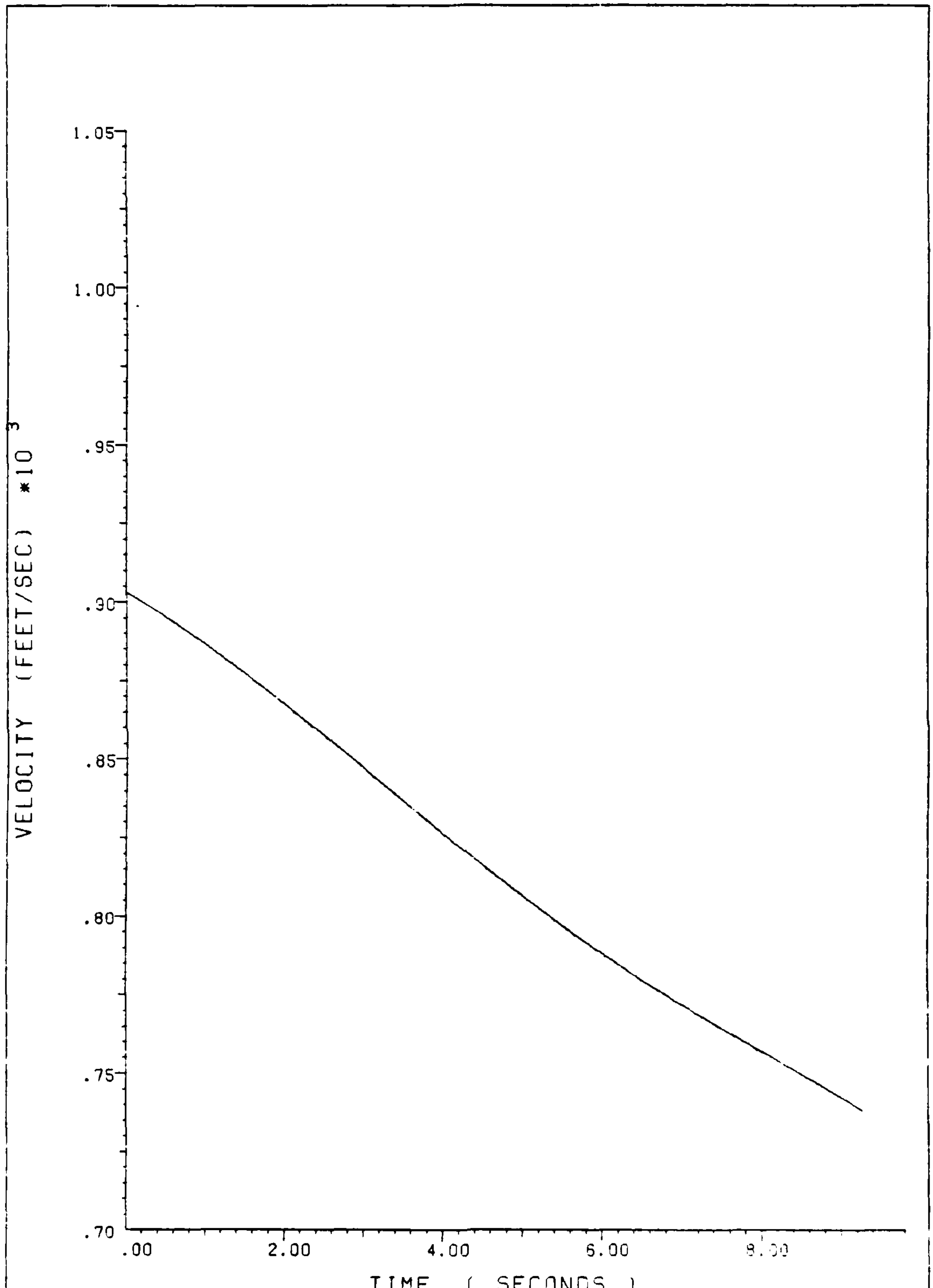


FIG 28. VELOCITY VS. TIME FOR CASE 5

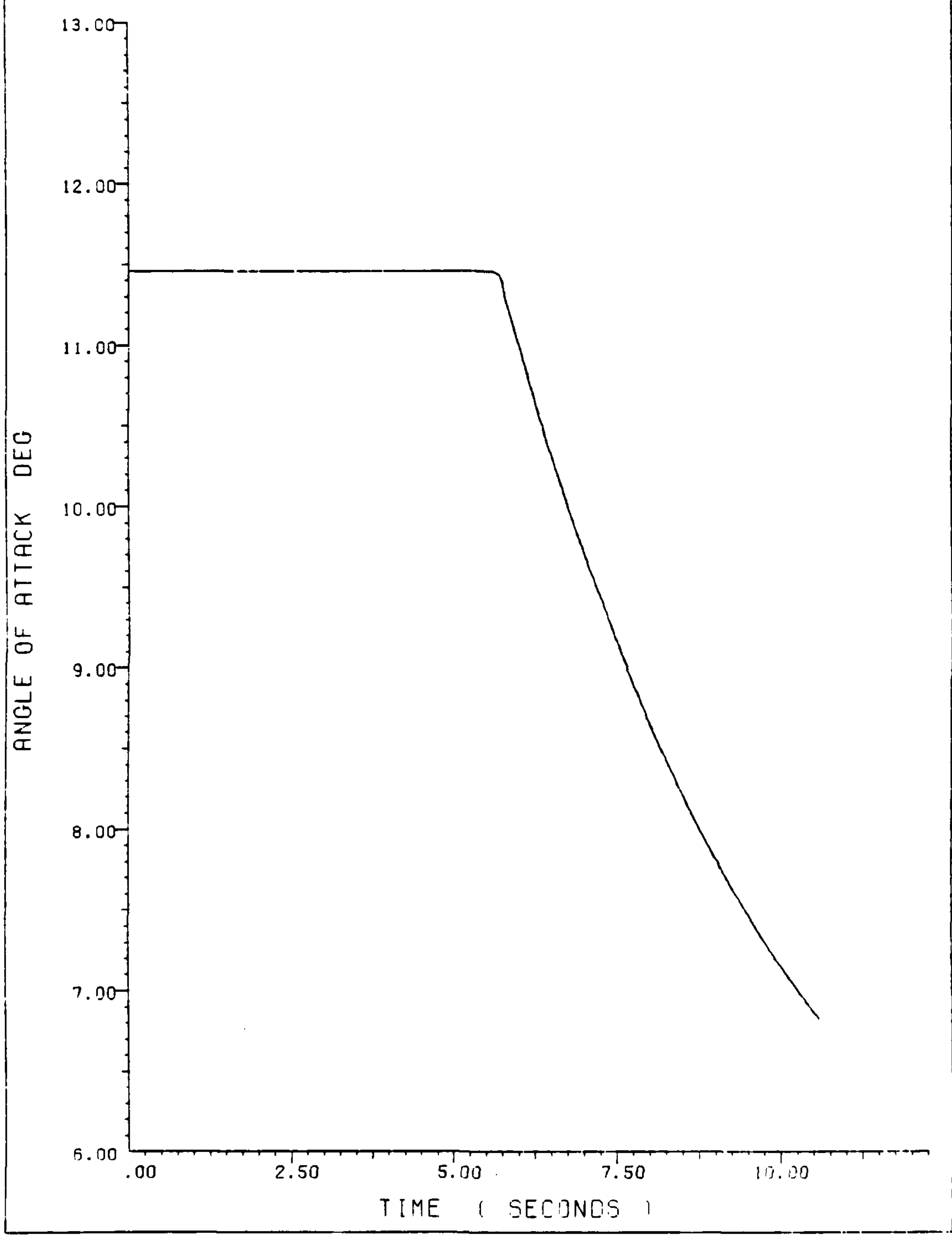


FIG 27. ANGLE OF ATTACK VS. TIME FOR CASE 2

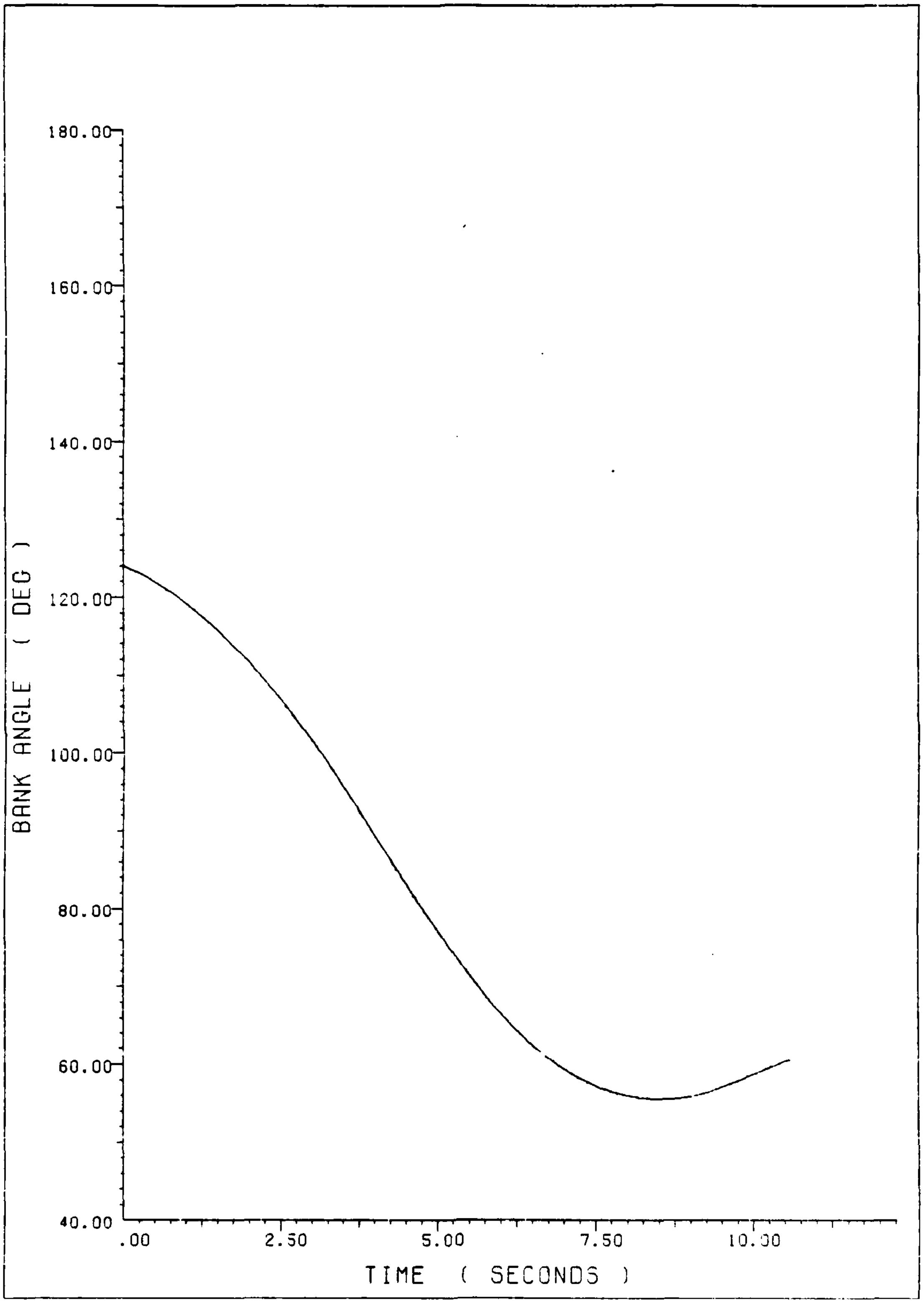


FIG 26. BANK ANGLE VS. TIME FOR CASE 2

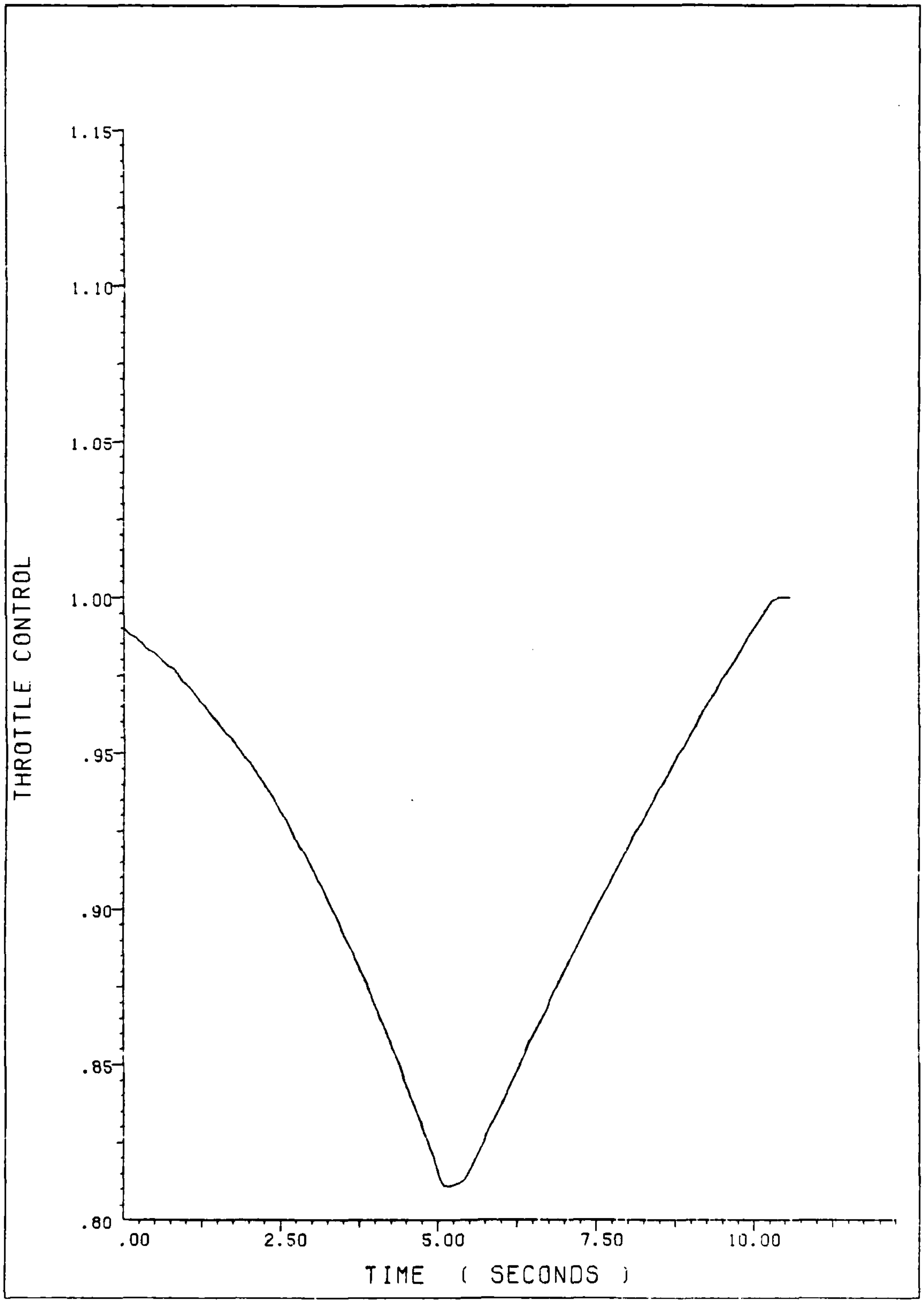


FIG 25. THROTTLE CONTROL VS. TIME FOR CASE 2

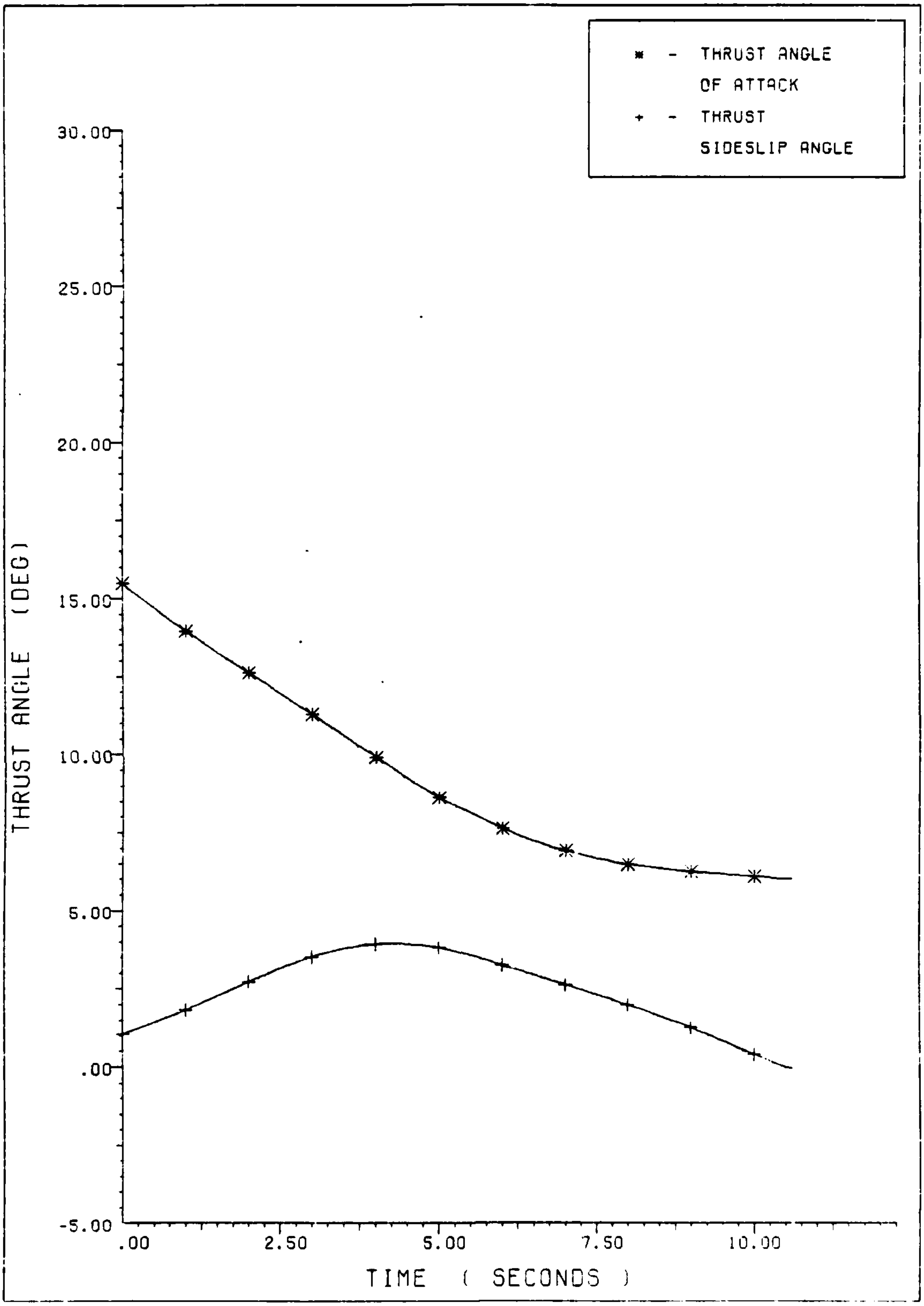


FIG 24. THRUST ANGLES VS. TIME FOR CASE 2

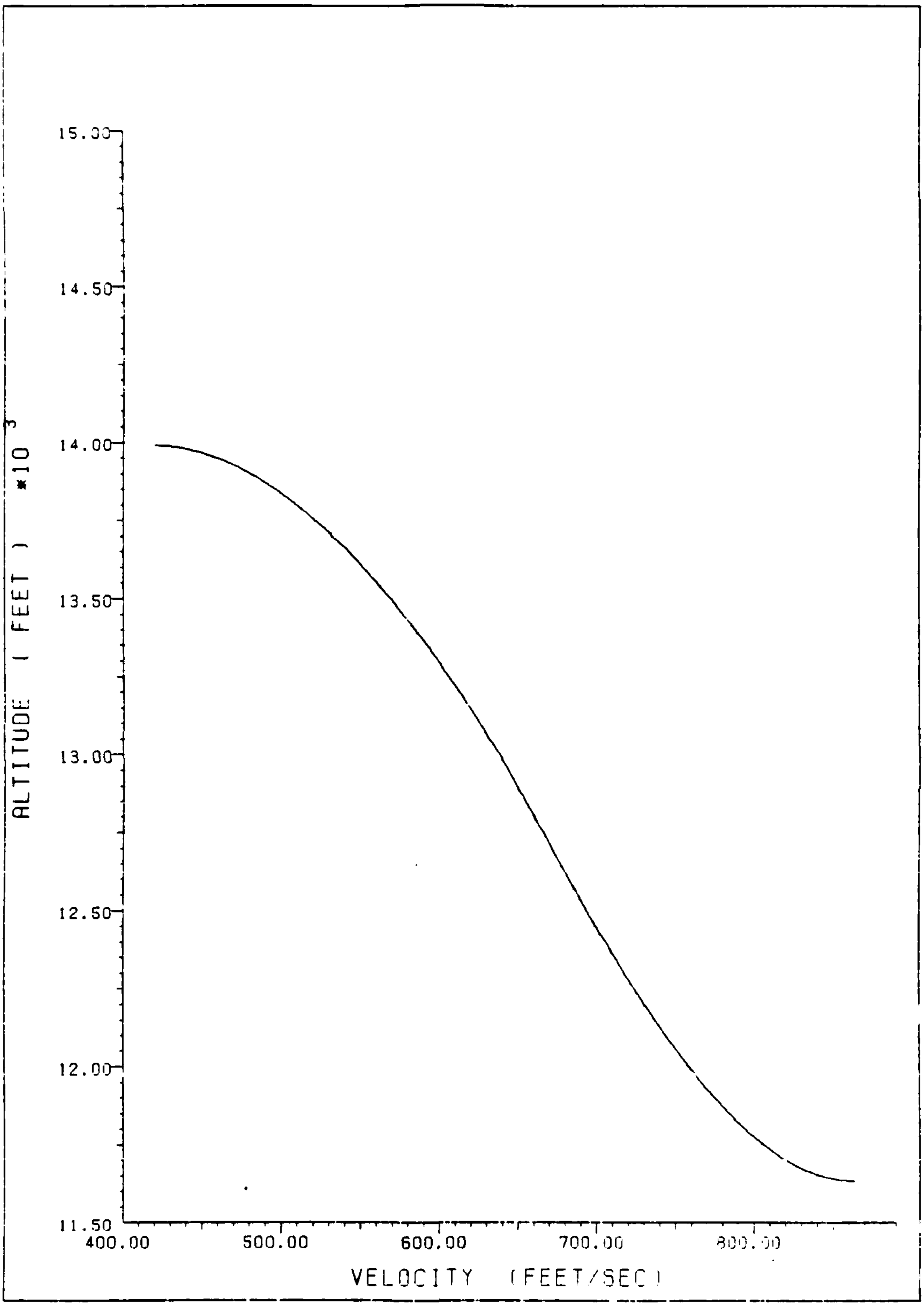


FIG 23. ALTITUDE VS. VELOCITY FOR CASE 2

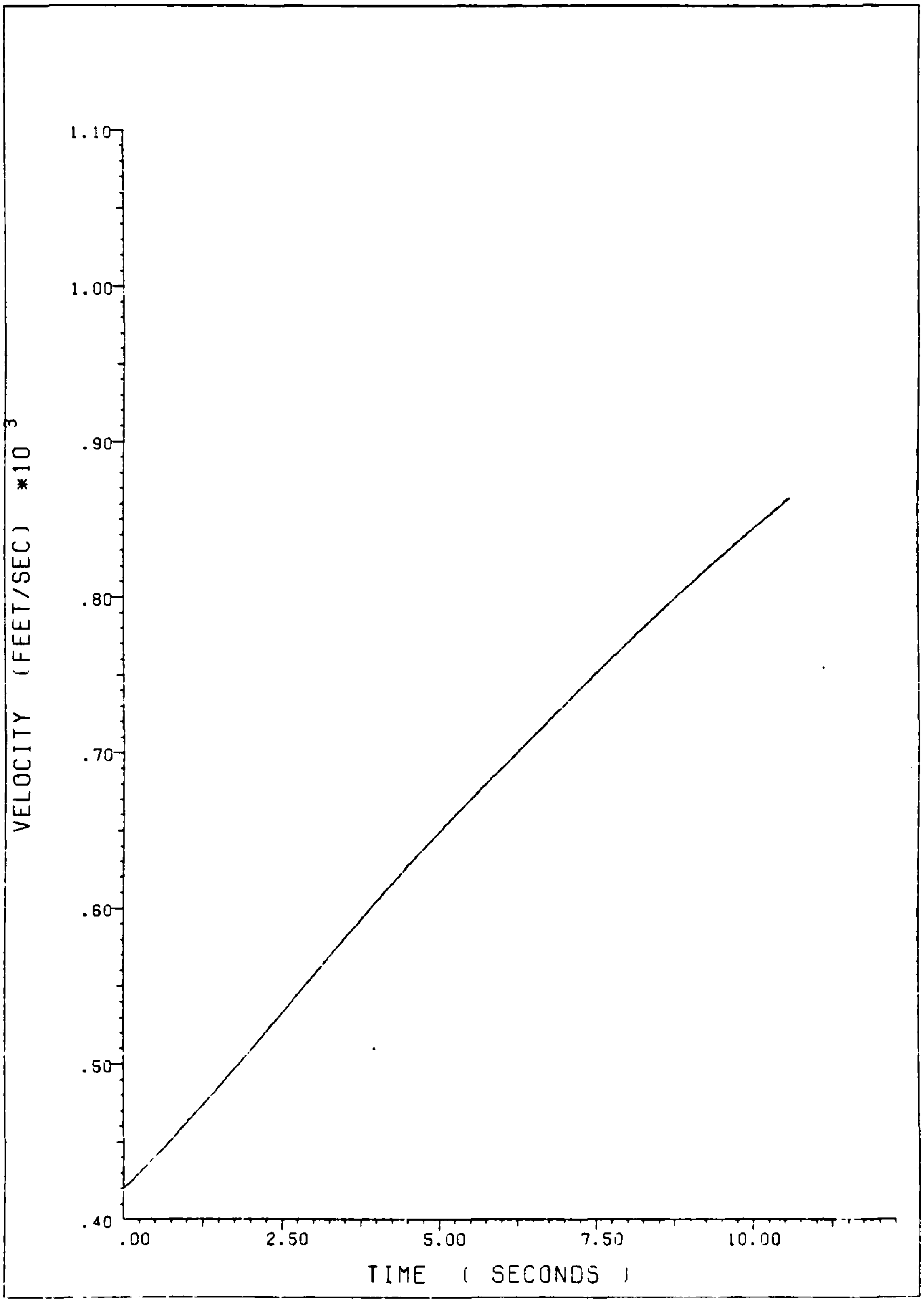


FIG 22. VELOCITY VS. TIME FOR CASE 2

Appendix E : Time Histories for Nominal Aircraft, Less Optimal Solutions

Table VI

Summary of Results:

Vectored Thrust, Limited Thrust Angles

(all initial altitudes 13,990 feet)

(nominal aircraft)

Case #	V_i (ft/sec)	Limits (Deg) ϵ	Limits (Deg) ν	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
13	621	20	0	768	13670	2857	9.91
14	621	20	10	776	13702	3082	10.36
15	903	20	0	740	13912	-4245	11.21
16	903	20	10	740	13912	-4245	11.21

Table V

Summary of Results:

Vectored Thrust, Variation of Aircraft Characteristics

(all initial altitudes 13,990 feet)

Case #	V_i (ft/sec)	$(\frac{T}{W})_{max}$	K_1	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
6	420	0.75	0.05	687	11651	2260	10.92
7	420	1.50	0.22	643	11499	1198	10.47
8	621	0.75	0.05	634	13175	-561	9.22
9	621	1.50	0.22	518	12896	-2920	9.06
10	621	1.50	0.22	660	12423	-790	9.45
11	903	0.75	0.05	710	13589	-5245	9.79
12	903	1.50	0.22	461	13452	-9920	9.11

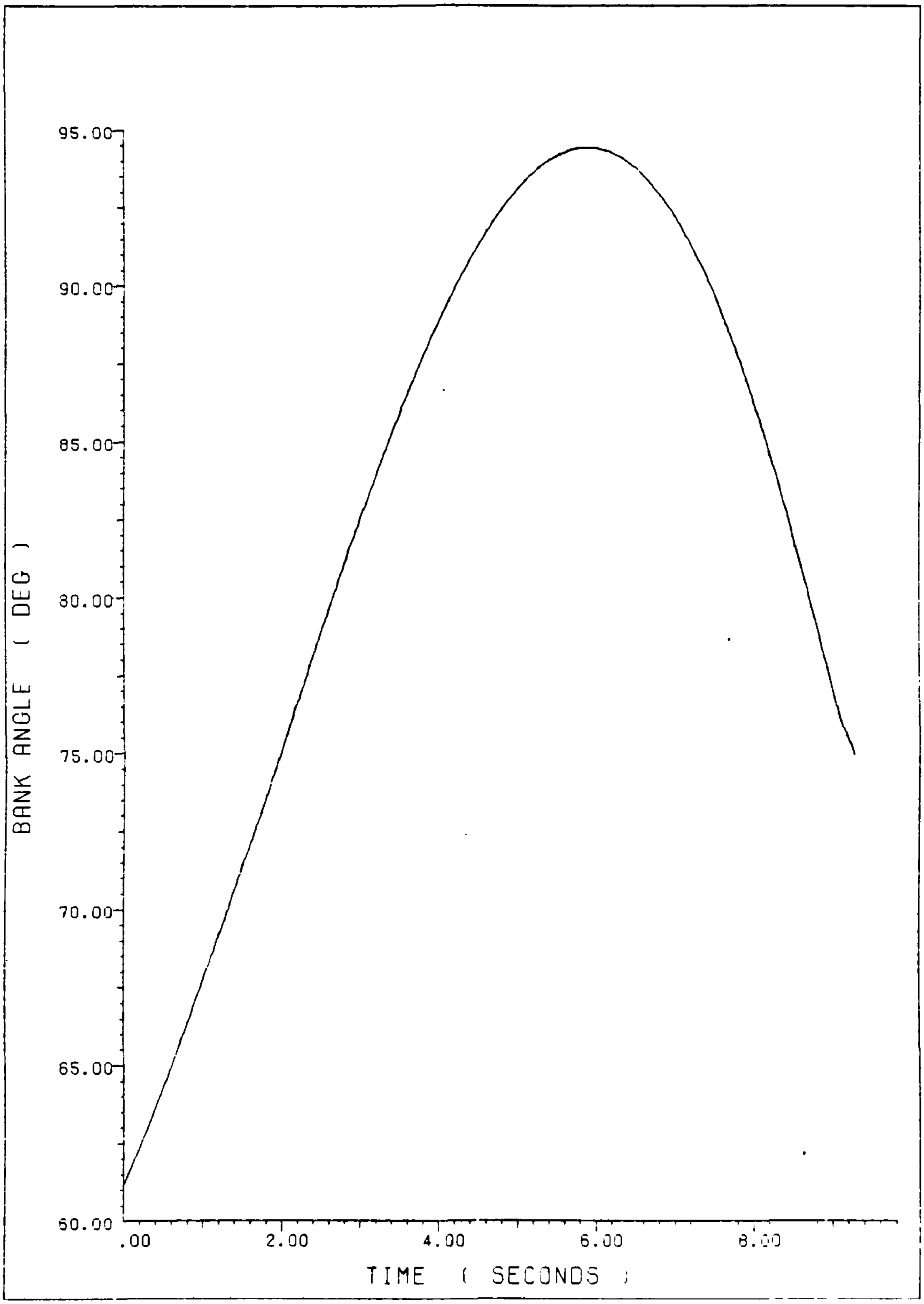


FIG 33. BANK ANGLE VS. TIME FOR CASE 5

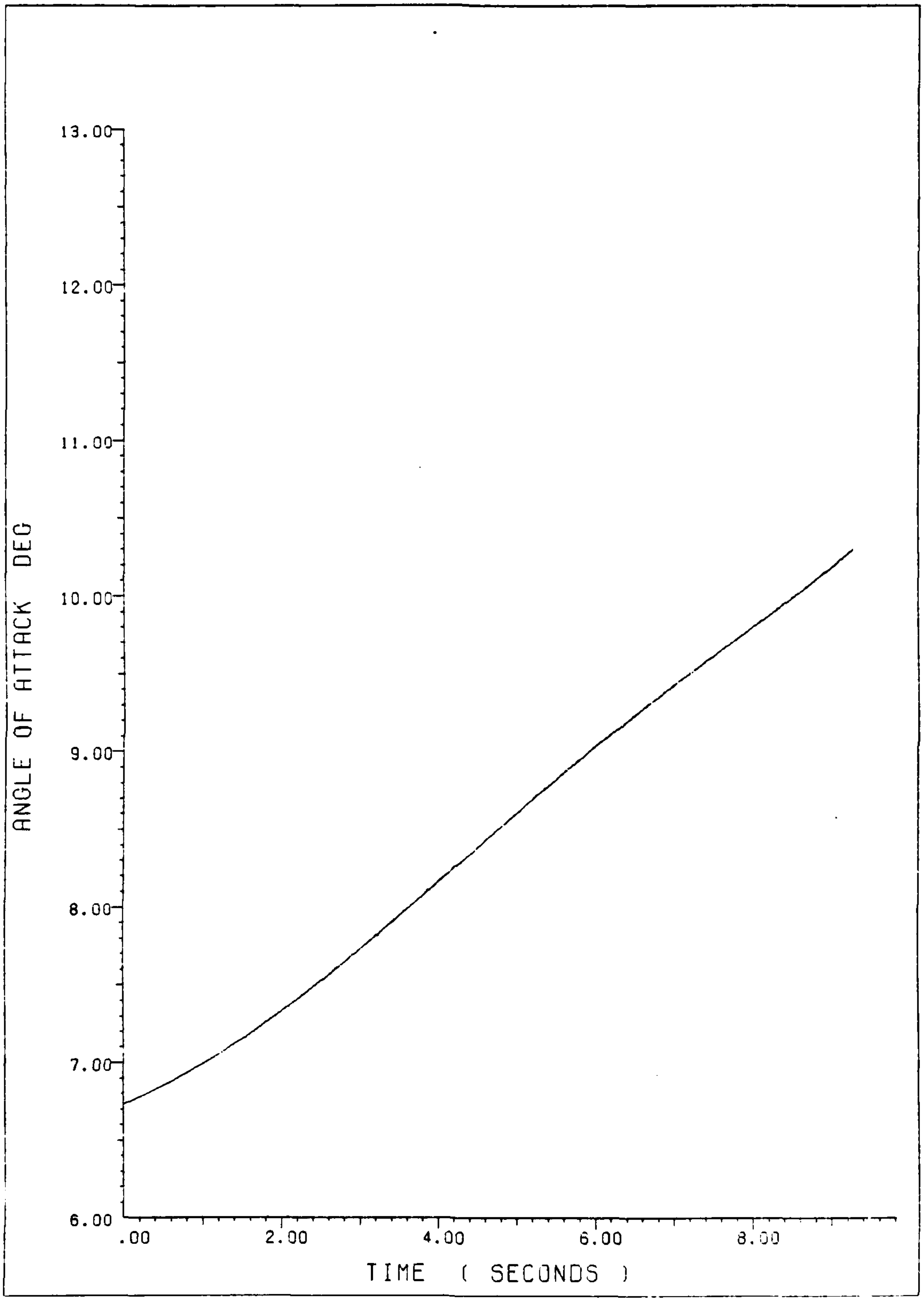


FIG 34. ANGLE OF ATTACK VS. TIME FOR CASE 5

Appendix F : Time Histories for Variations of Aircraft Characteristics

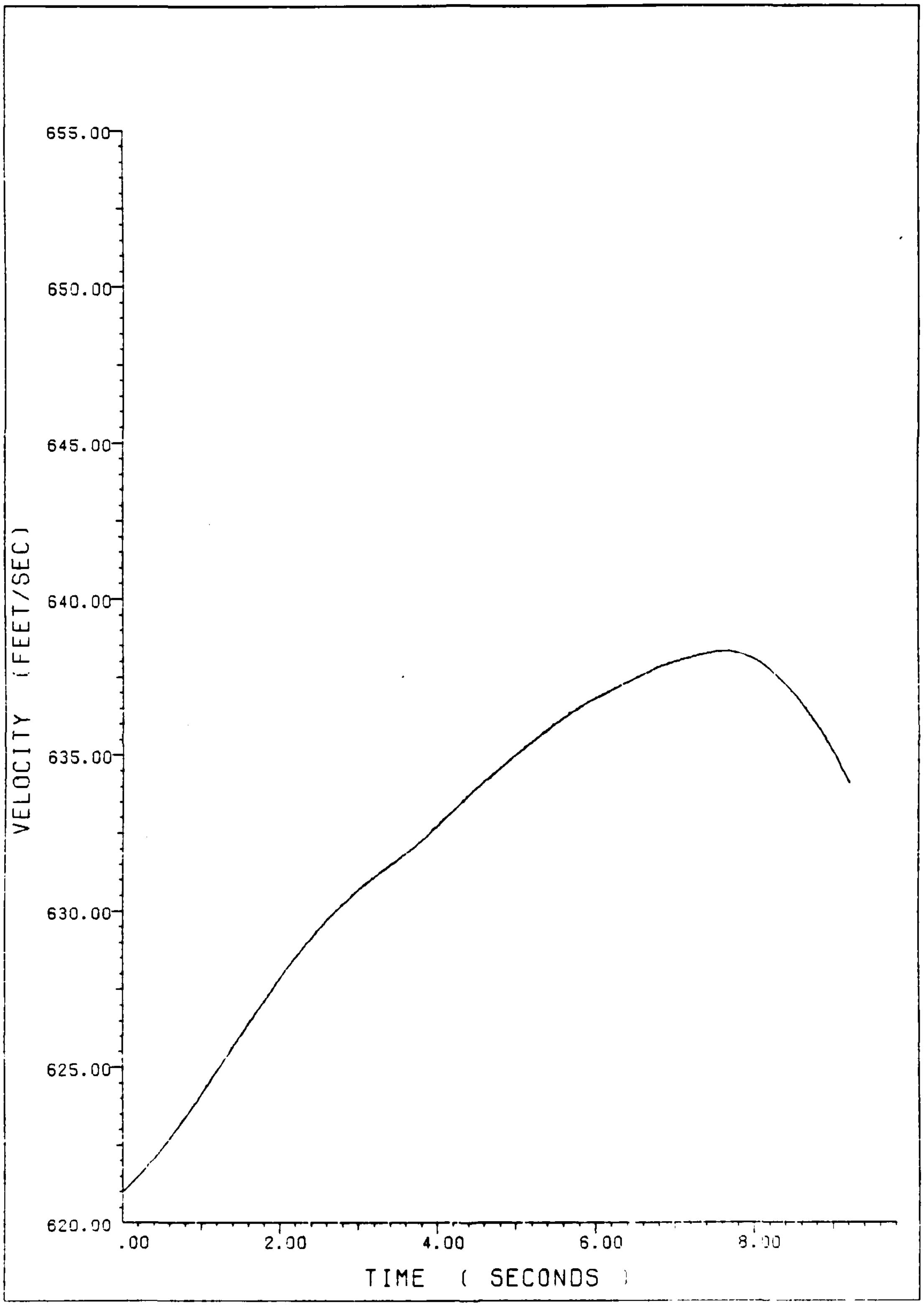


FIG 35. VELOCITY VS. TIME FOR CASE 8

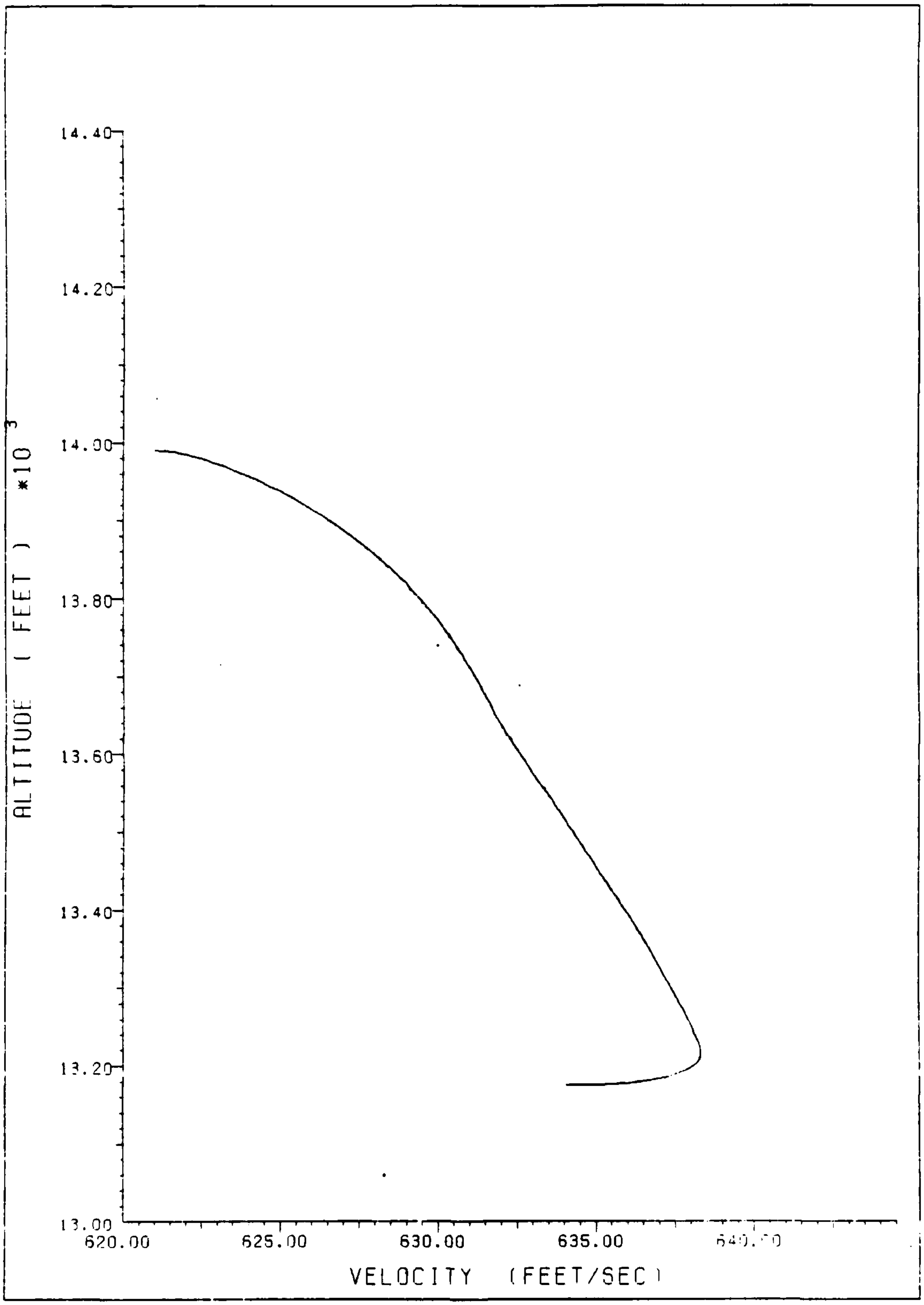


FIG 36. ALTITUDE VS. VELOCITY FOR CASE 3

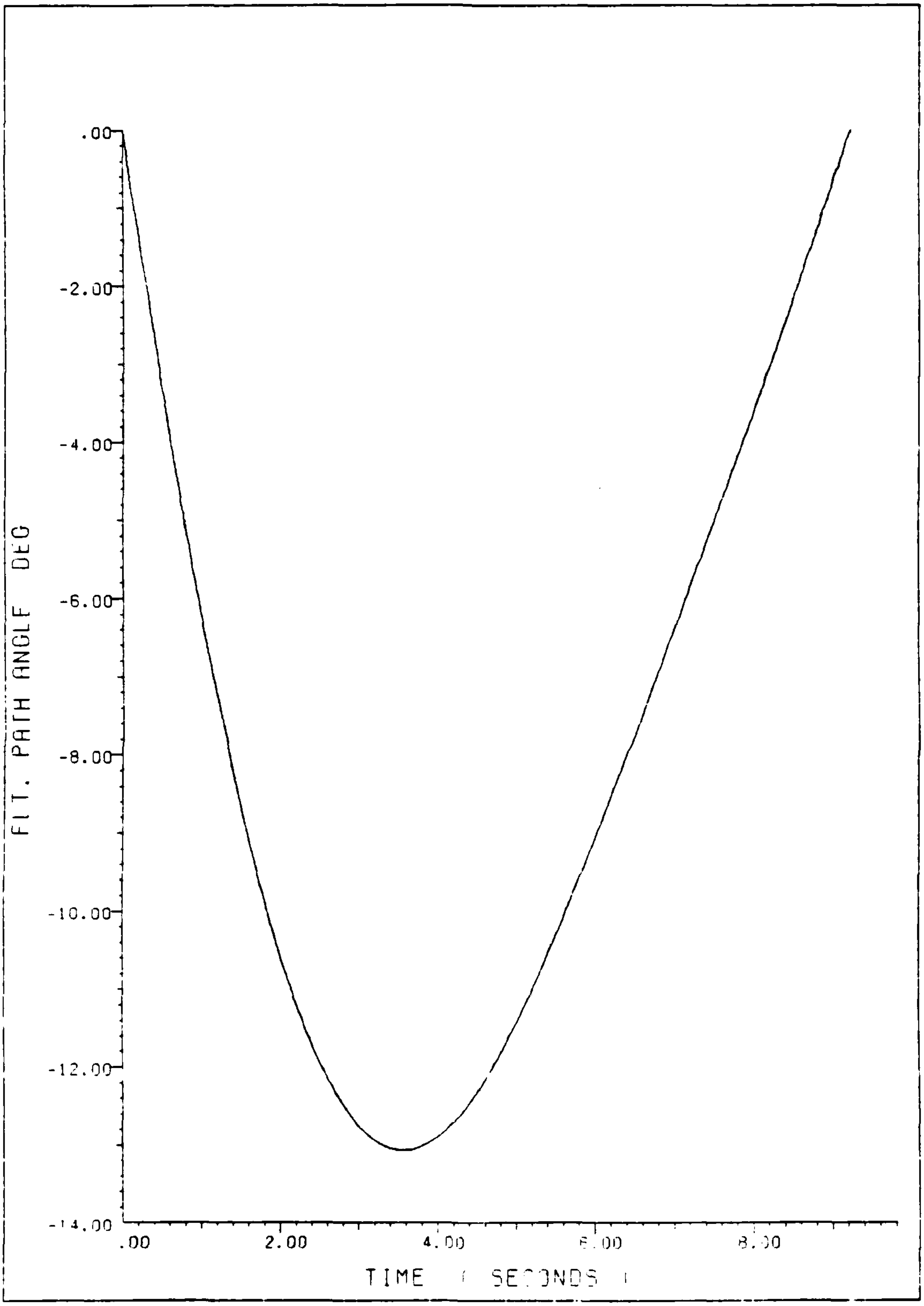


FIG 37. FLIGHT PATH ANGLE VS. TIME FOR CASE 8

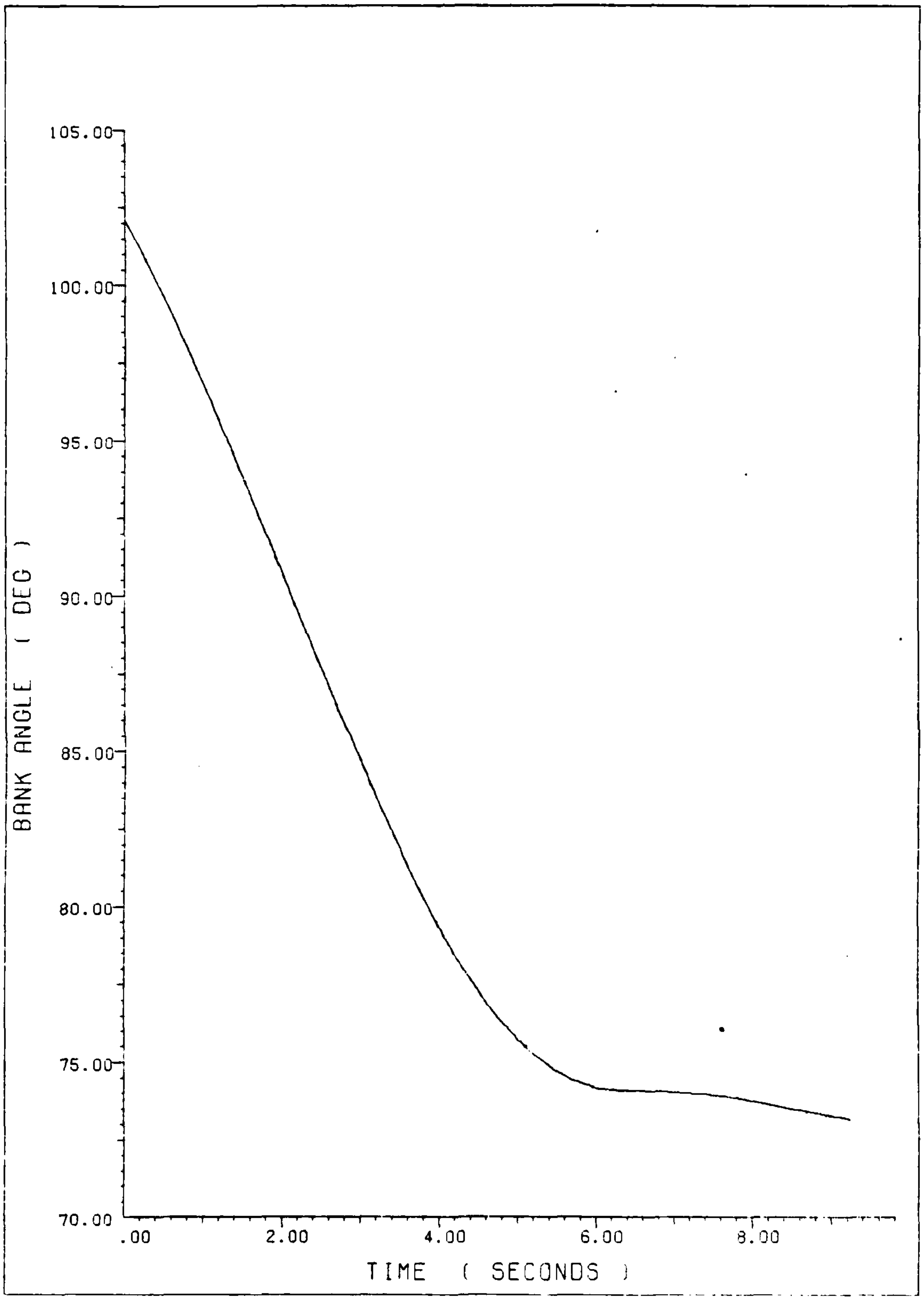


FIG 38. BANK ANGLE VS. TIME FOR CASE 3

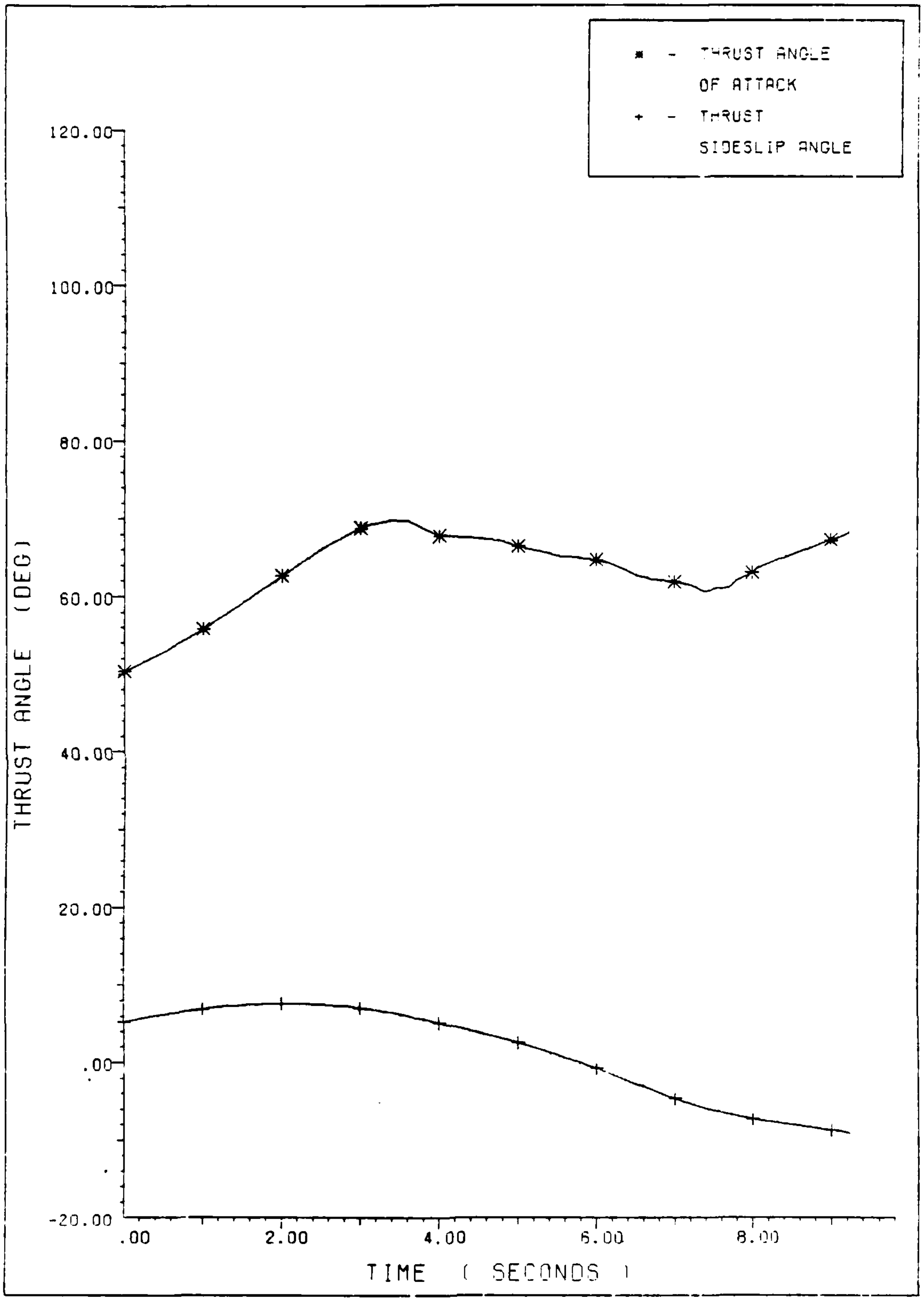


FIG 39. THRUST ANGLES VS. TIME FOR CASE 8

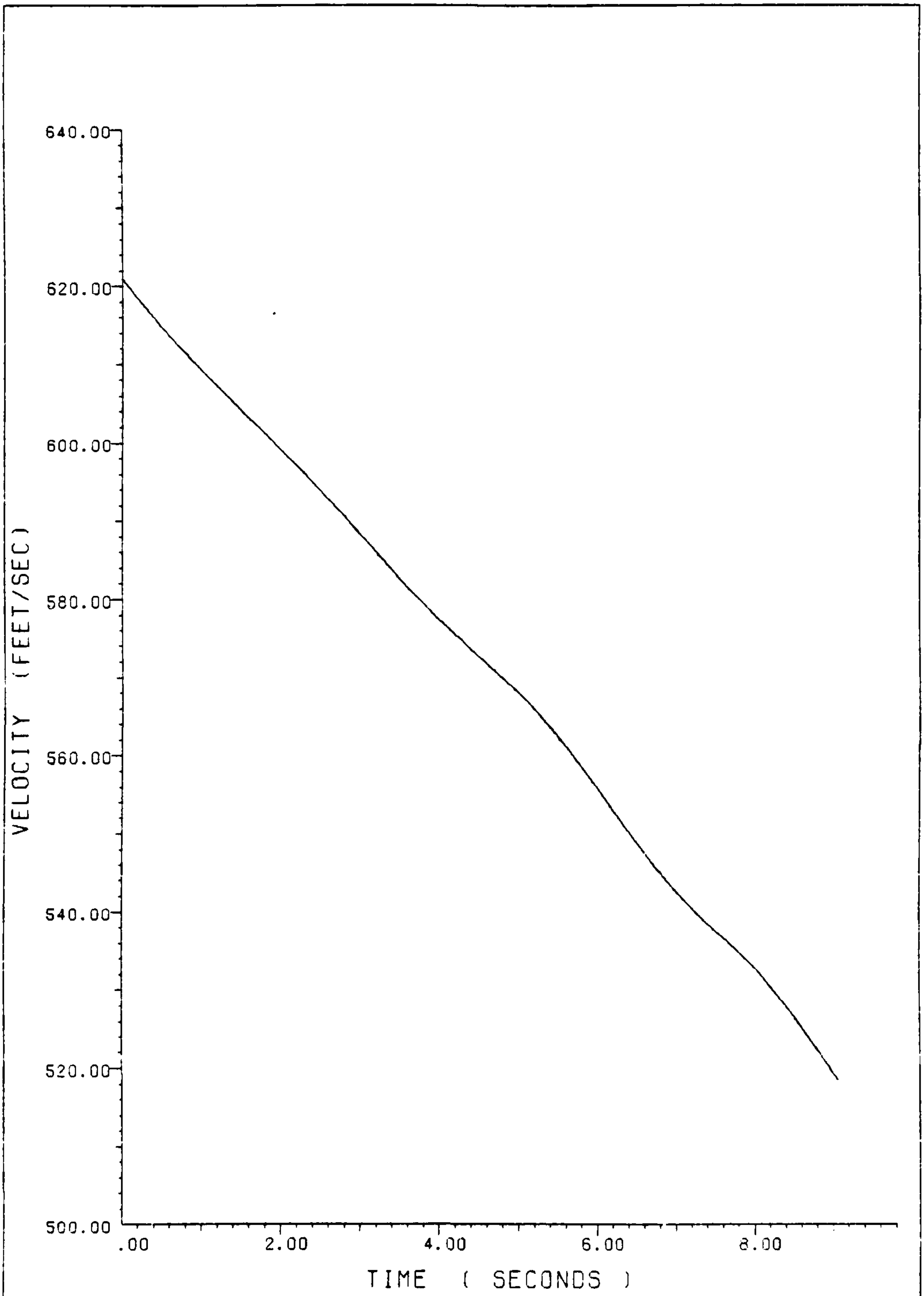


FIG 40. VELOCITY VS. TIME FOR CASE 9

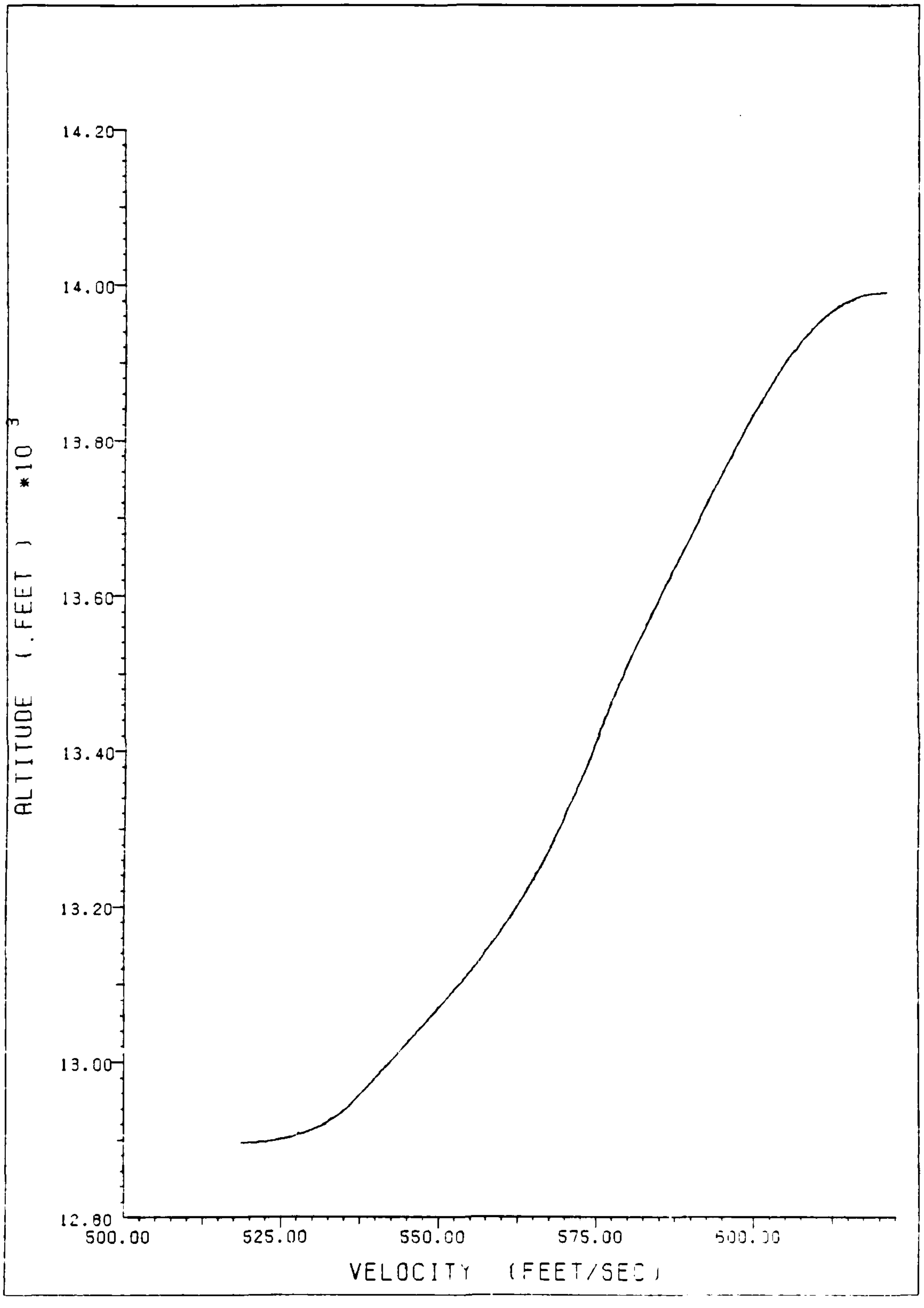


FIG 41. ALTITUDE VS. VELOCITY FOR CASE 9

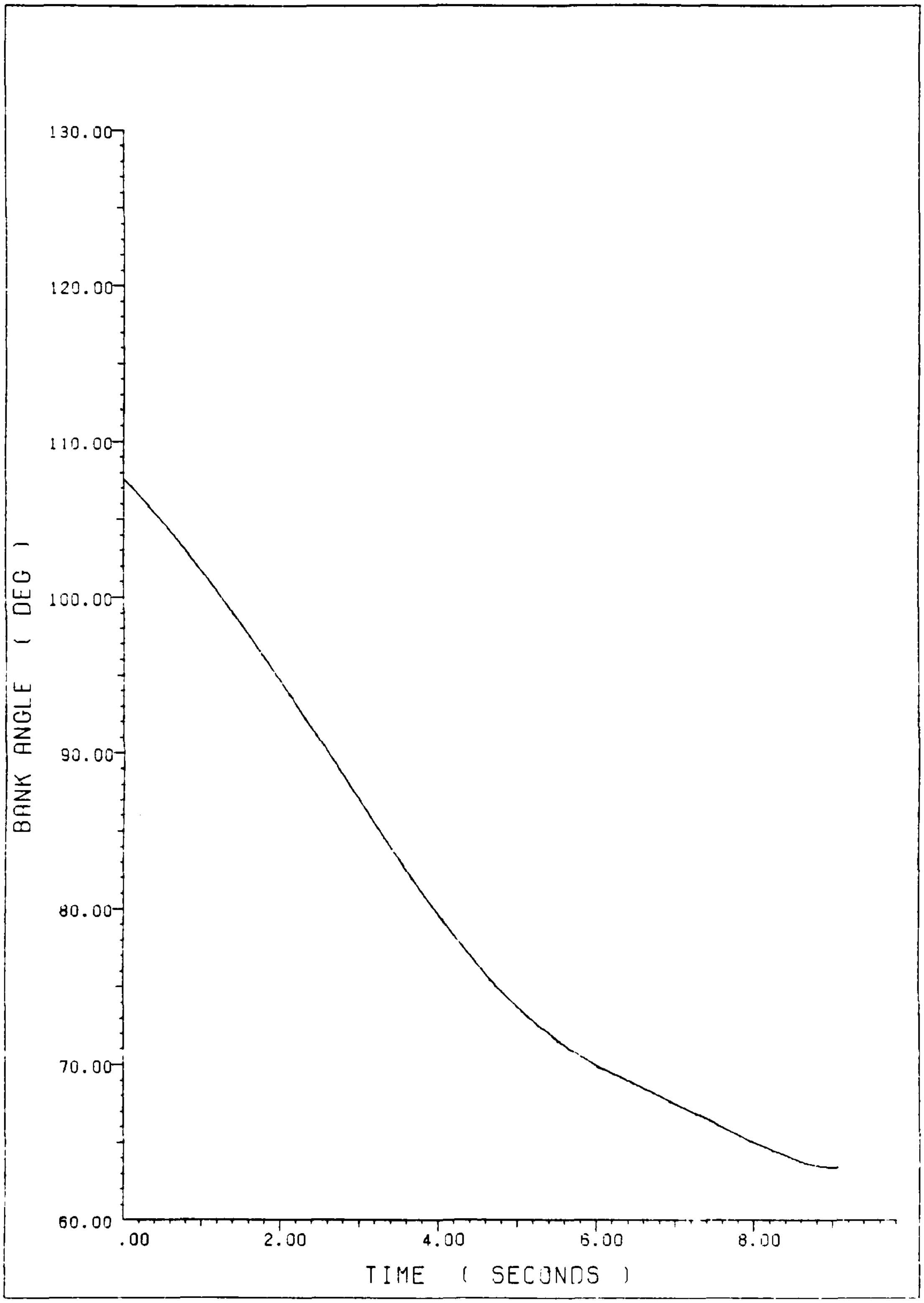


FIG 42. BANK ANGLE VS. TIME FOR CASE 9

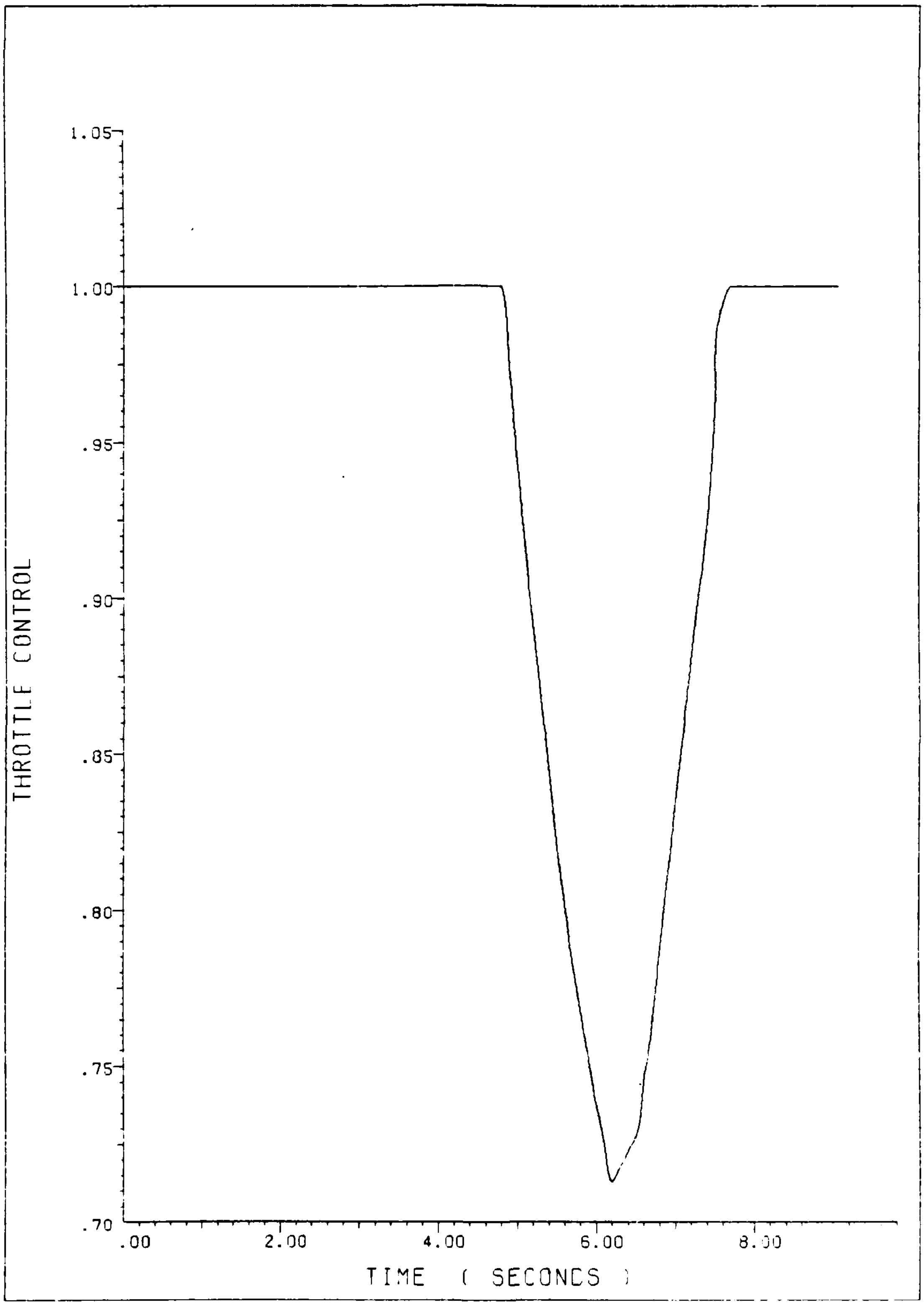


FIG 43. THROTTLE CONTROL VS. TIME FOR CASE 9

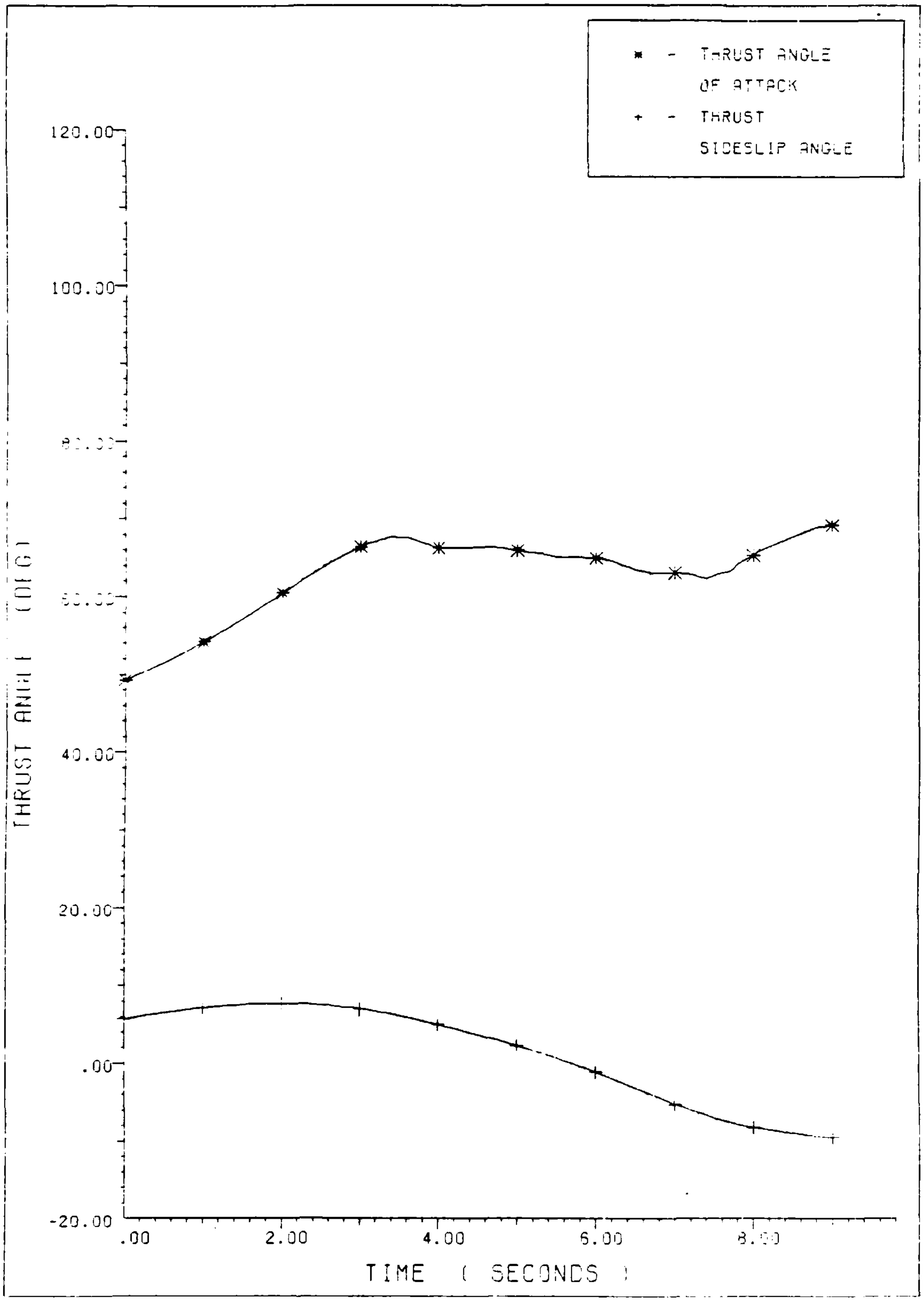


FIG 44. THRUST ANGLES VS. TIME FOR CASE 9

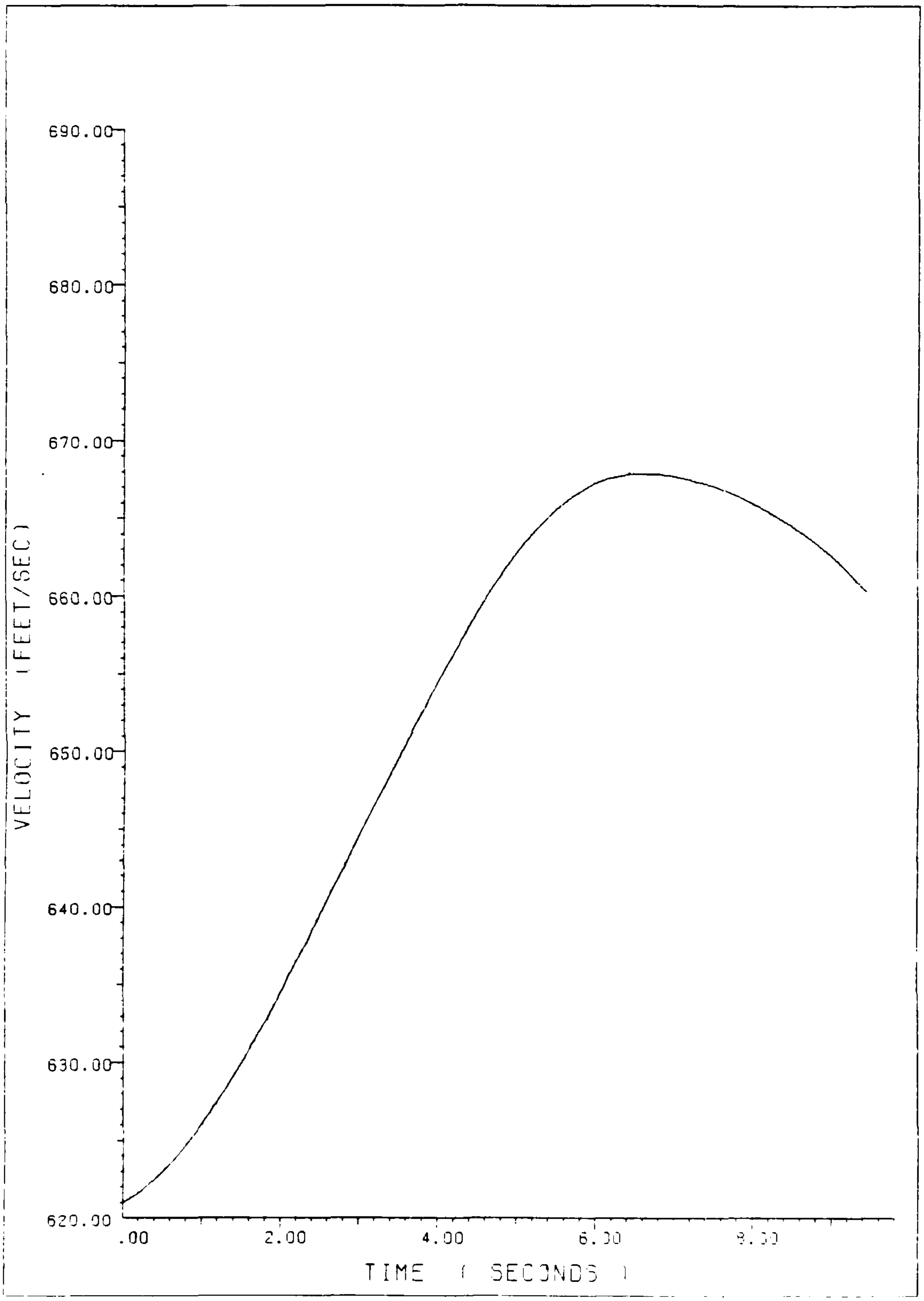


FIG 45. VELOCITY VS. TIME FOR CASE 1[illegible]

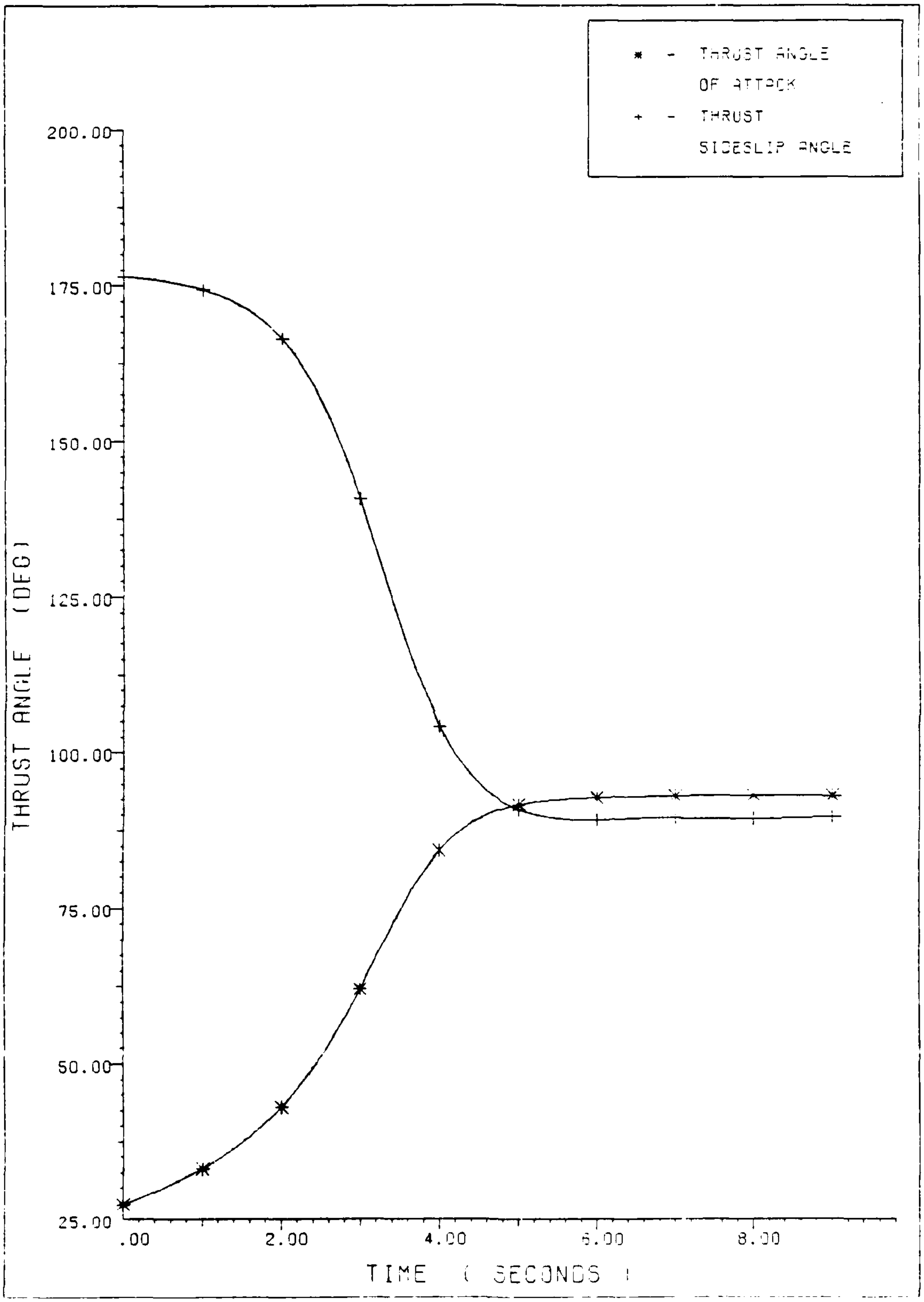


FIG 59. THRUST ANGLES VS. TIME FOR CASE 12

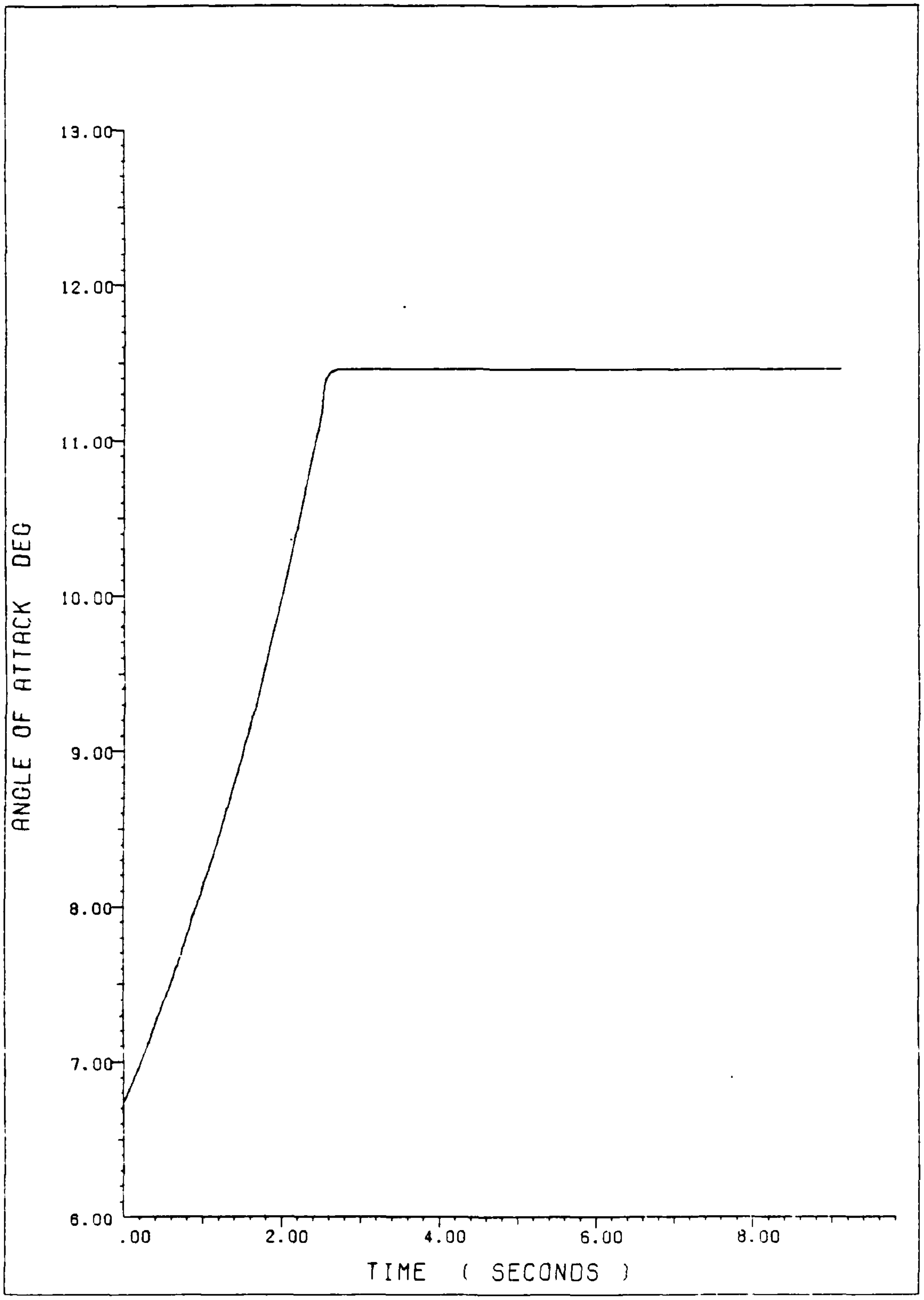


FIG 58. ANGLE OF ATTACK VS. TIME FOR CASE 12

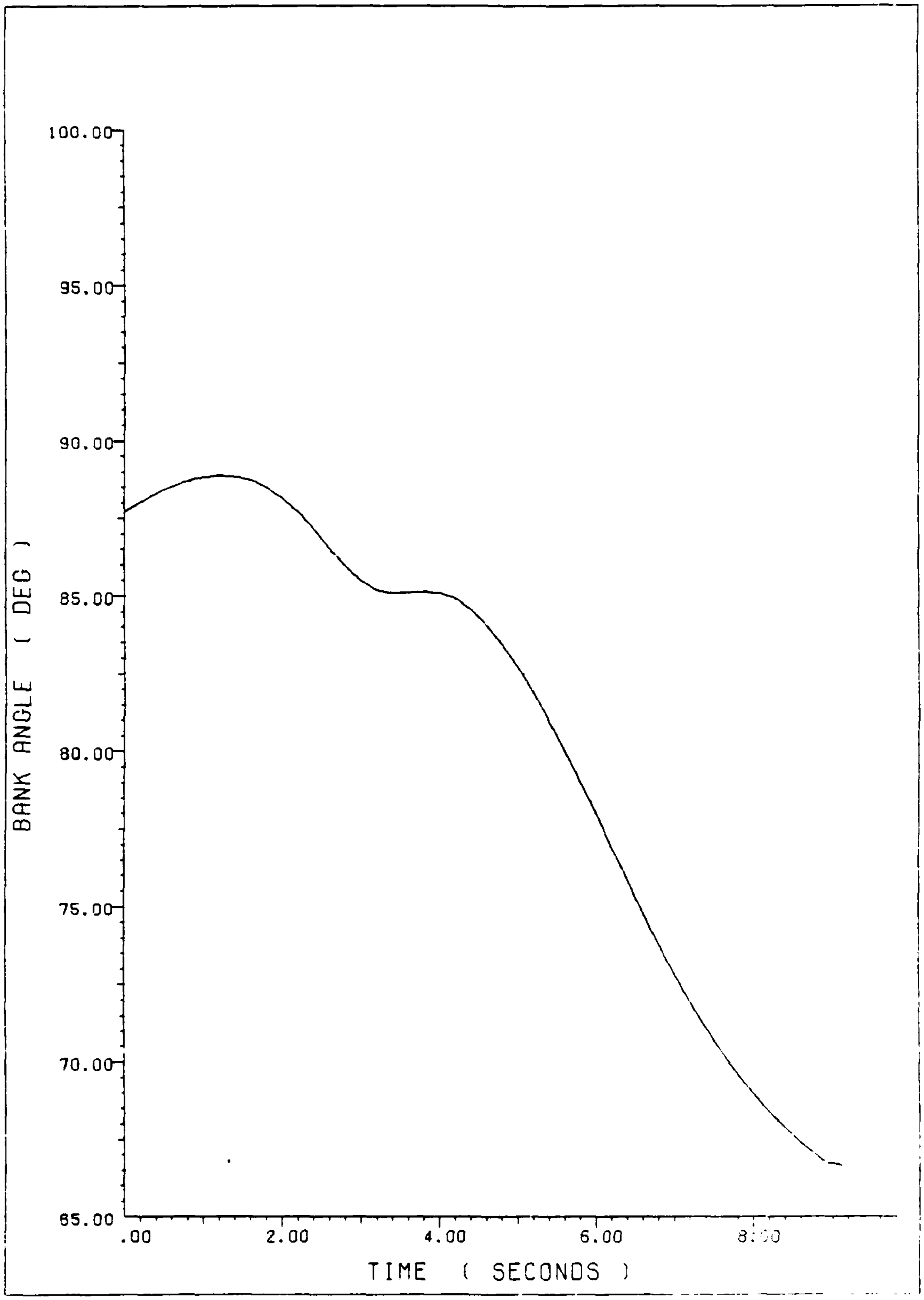


FIG 57. BANK ANGLE VS. TIME FOR CASE 12

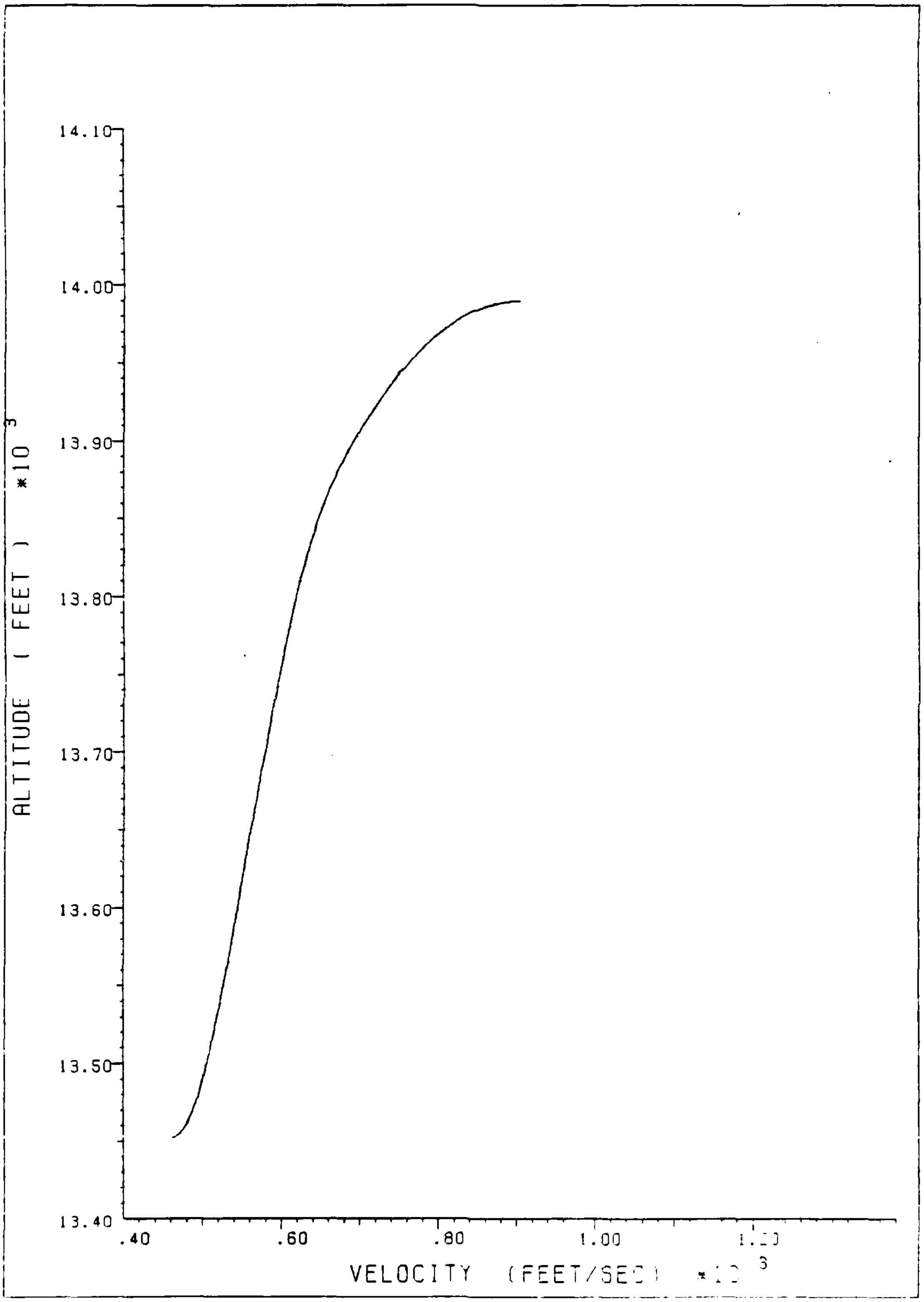


FIG 56. ALTITUDE VS. VELOCITY FOR CASE 12

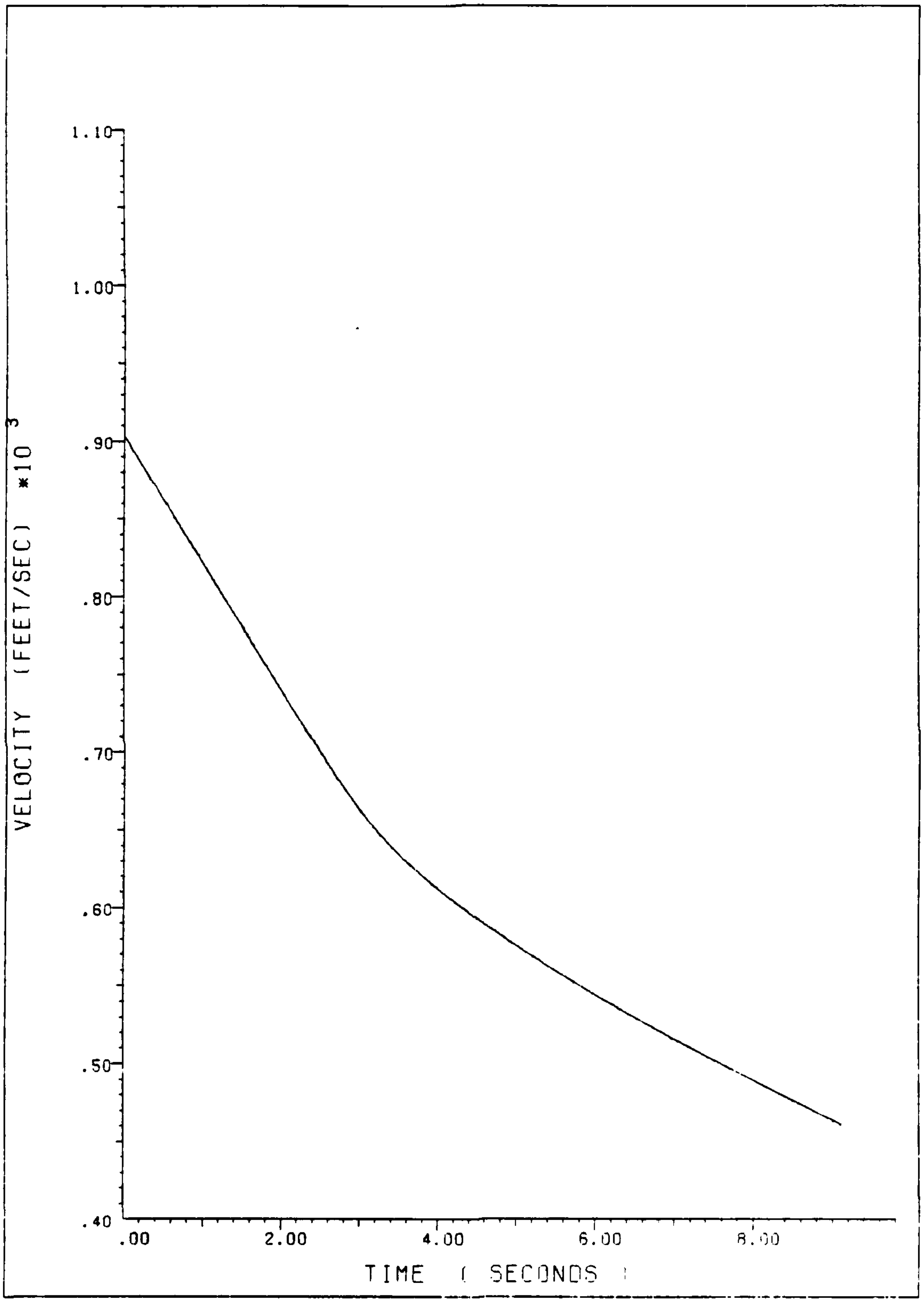


FIG 55. VELOCITY VS. TIME FOR CASE 12

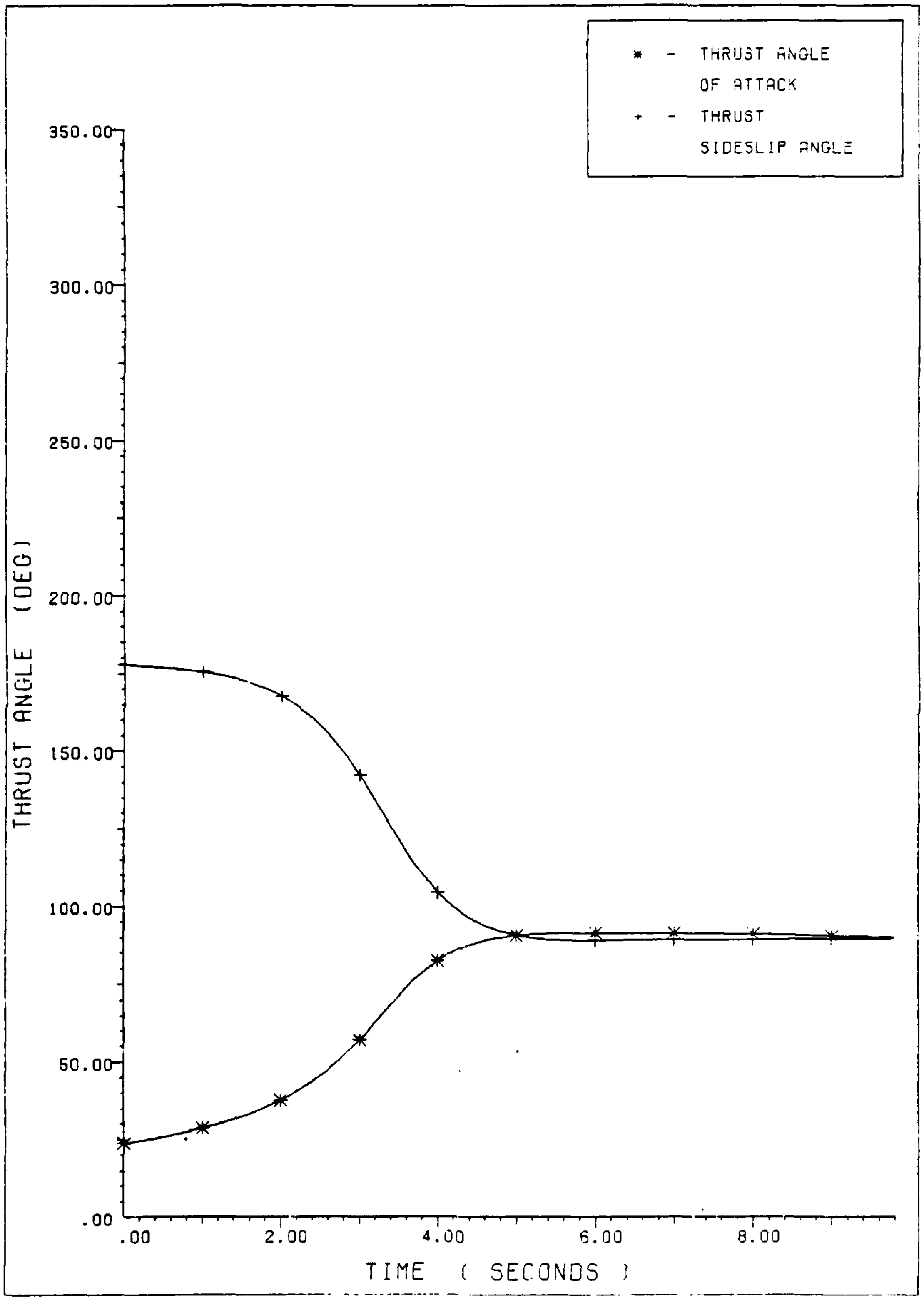


FIG 54. THRUST ANGLES VS. TIME FOR CASE II

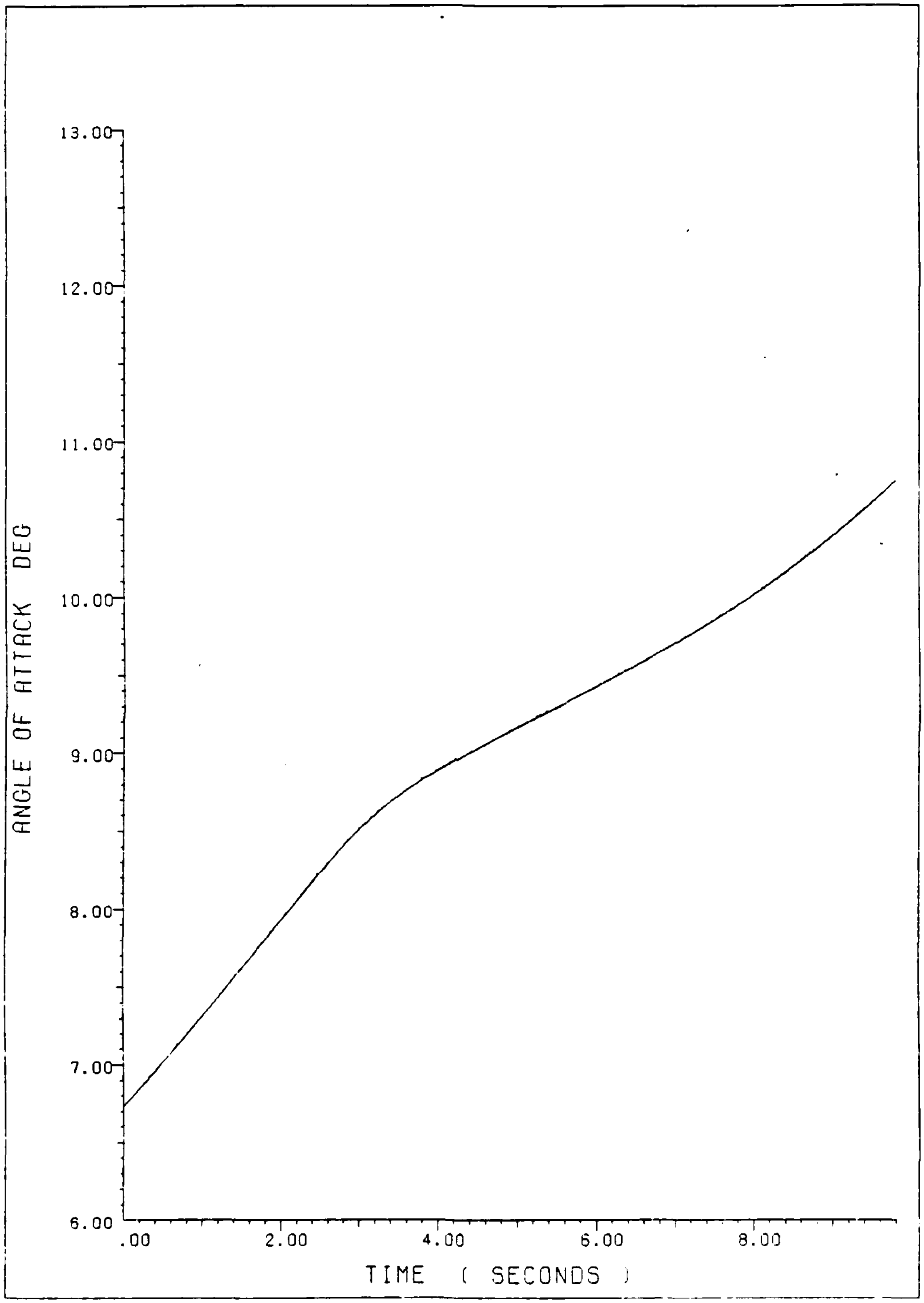


FIG 53. ANGLE OF ATTACK VS. TIME FOR CASE 11

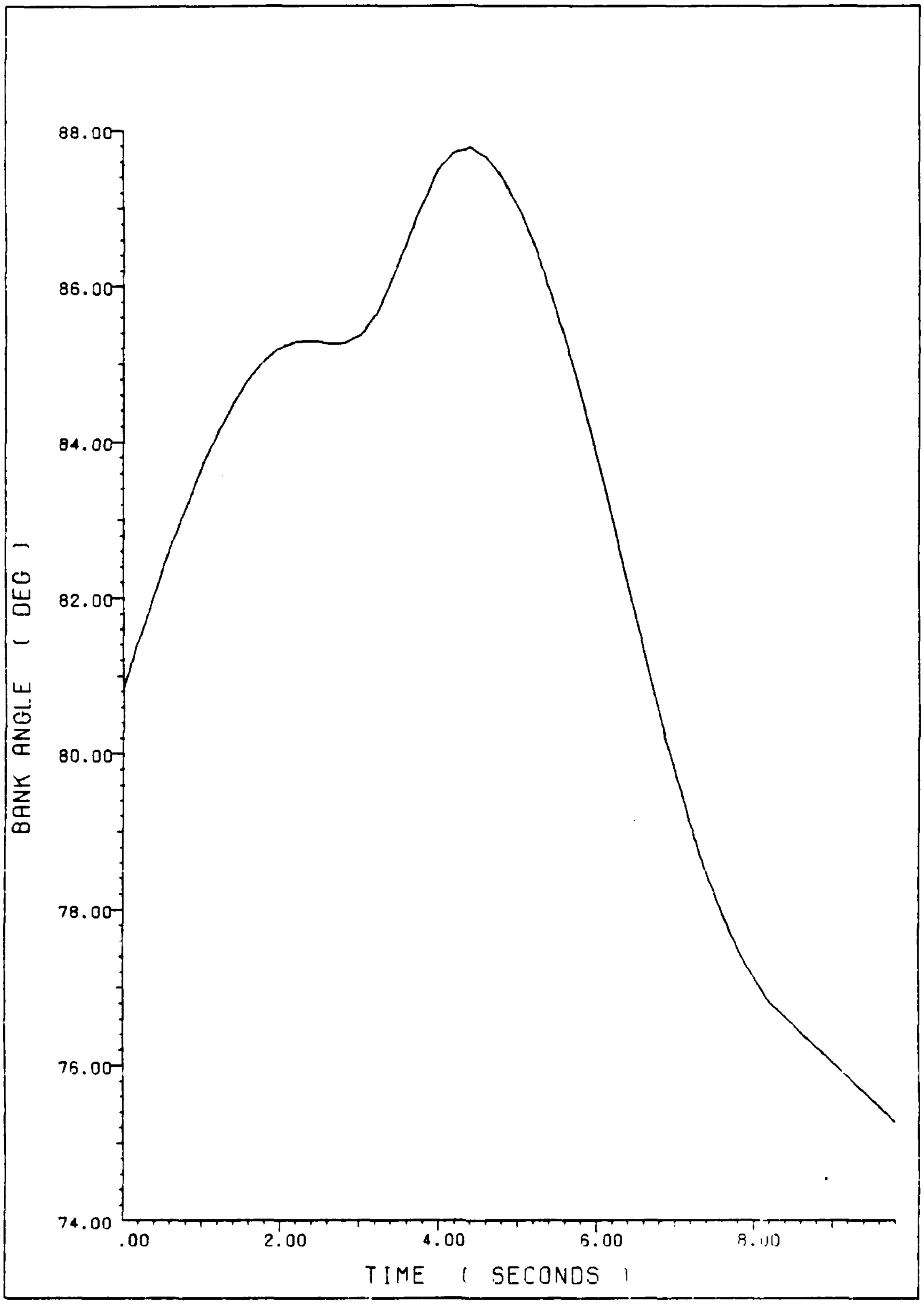


FIG 52. BANK ANGLE VS. TIME FOR CASE 11

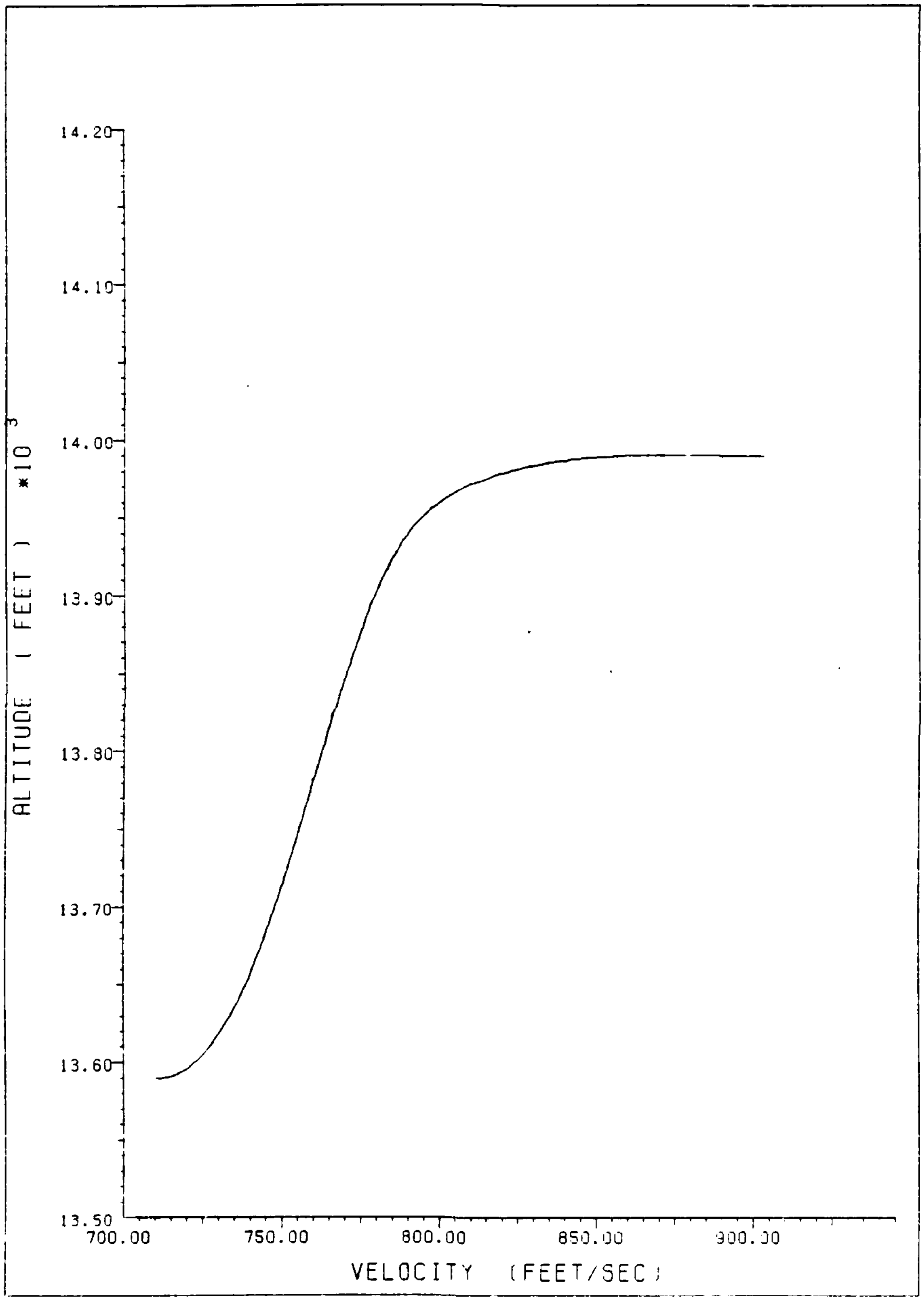


FIG 51. ALTITUDE VS. VELOCITY FOR CASE 11

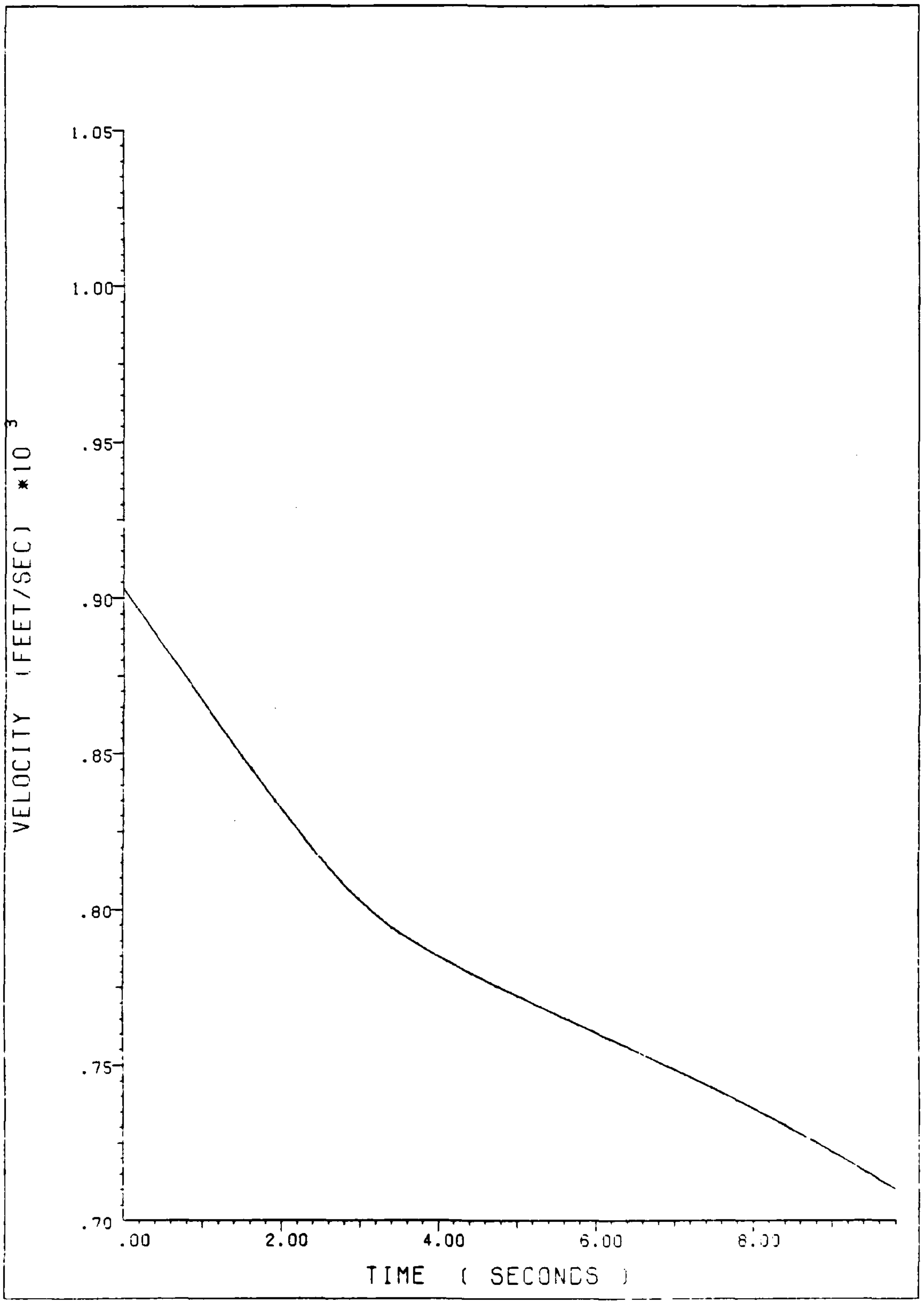


FIG 50. VELOCITY VS. TIME FOR CASE 11

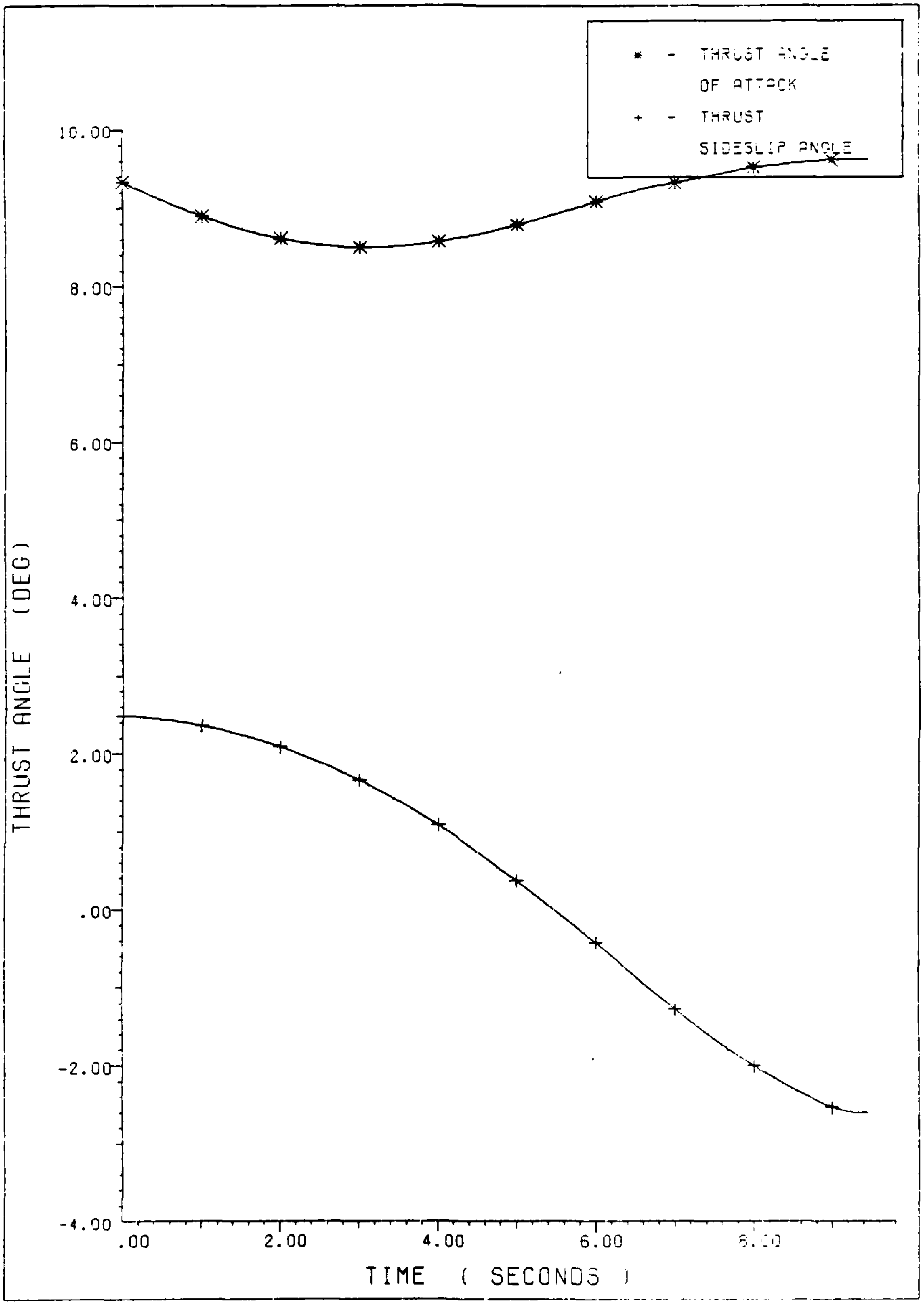


FIG 49. THRUST ANGLES VS. TIME FOR CASE 10

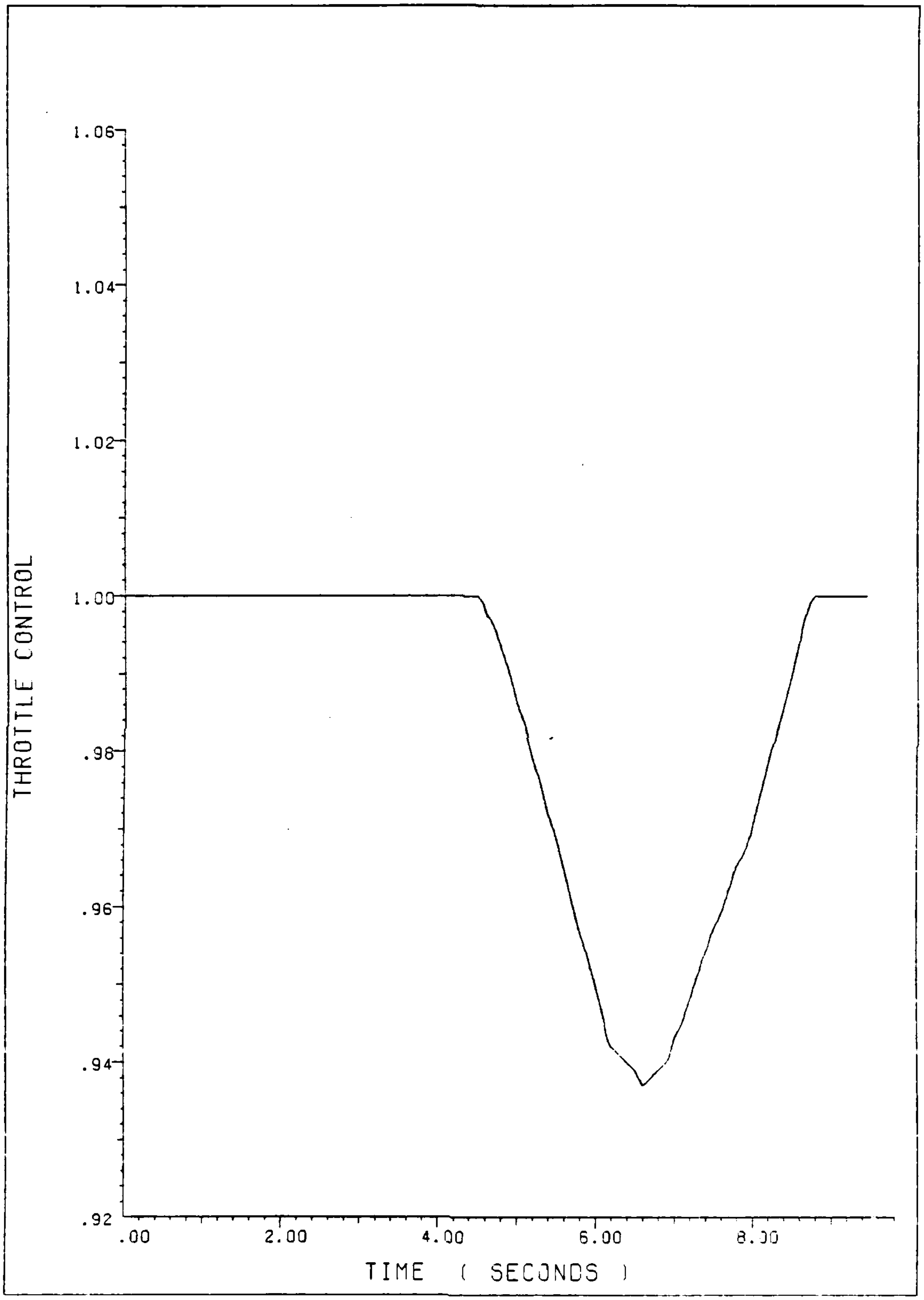


FIG 48. THROTTLE CONTROL VS. TIME FOR CASE 10

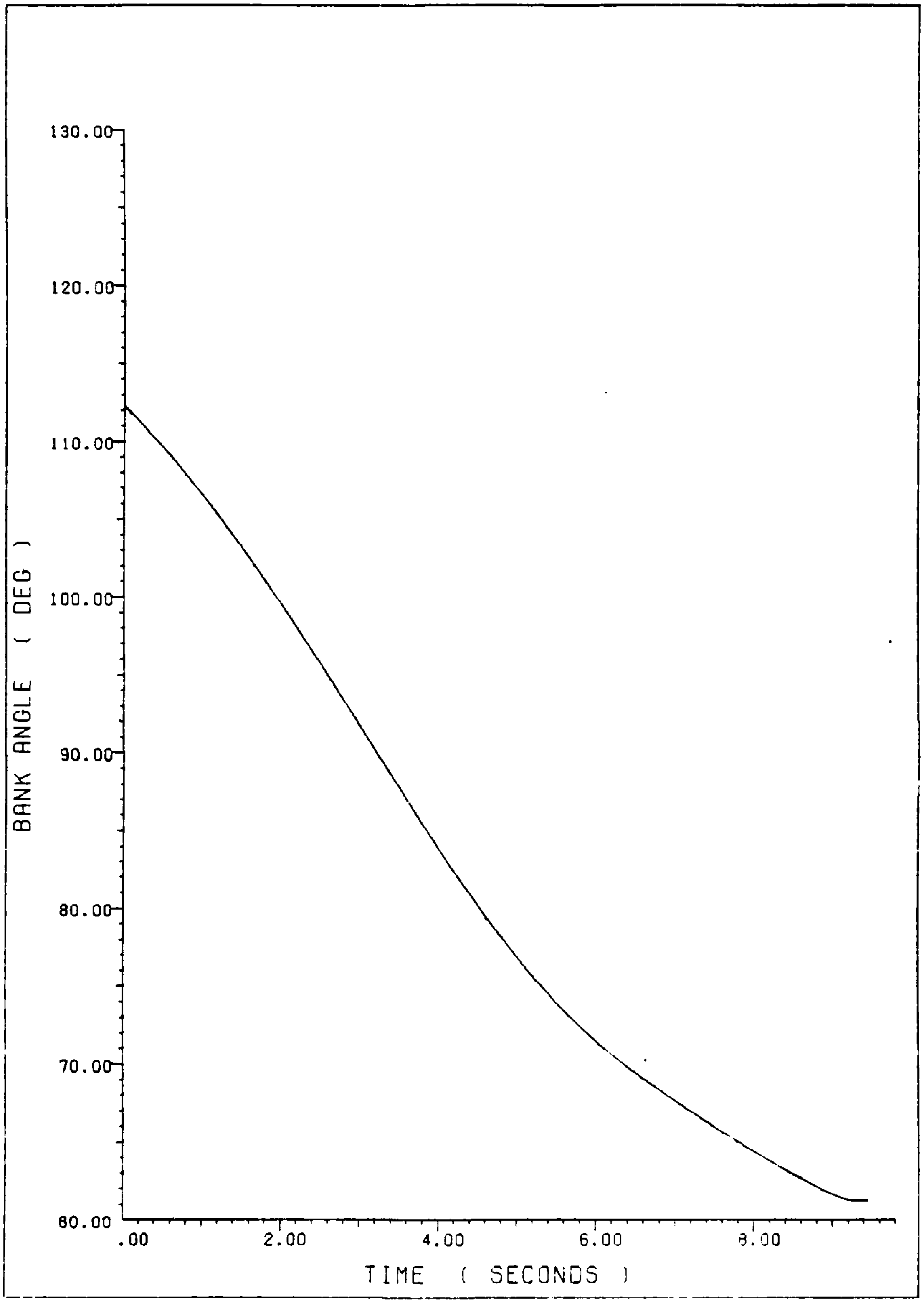


FIG 47. BANK ANGLE VS. TIME FOR CASE 10

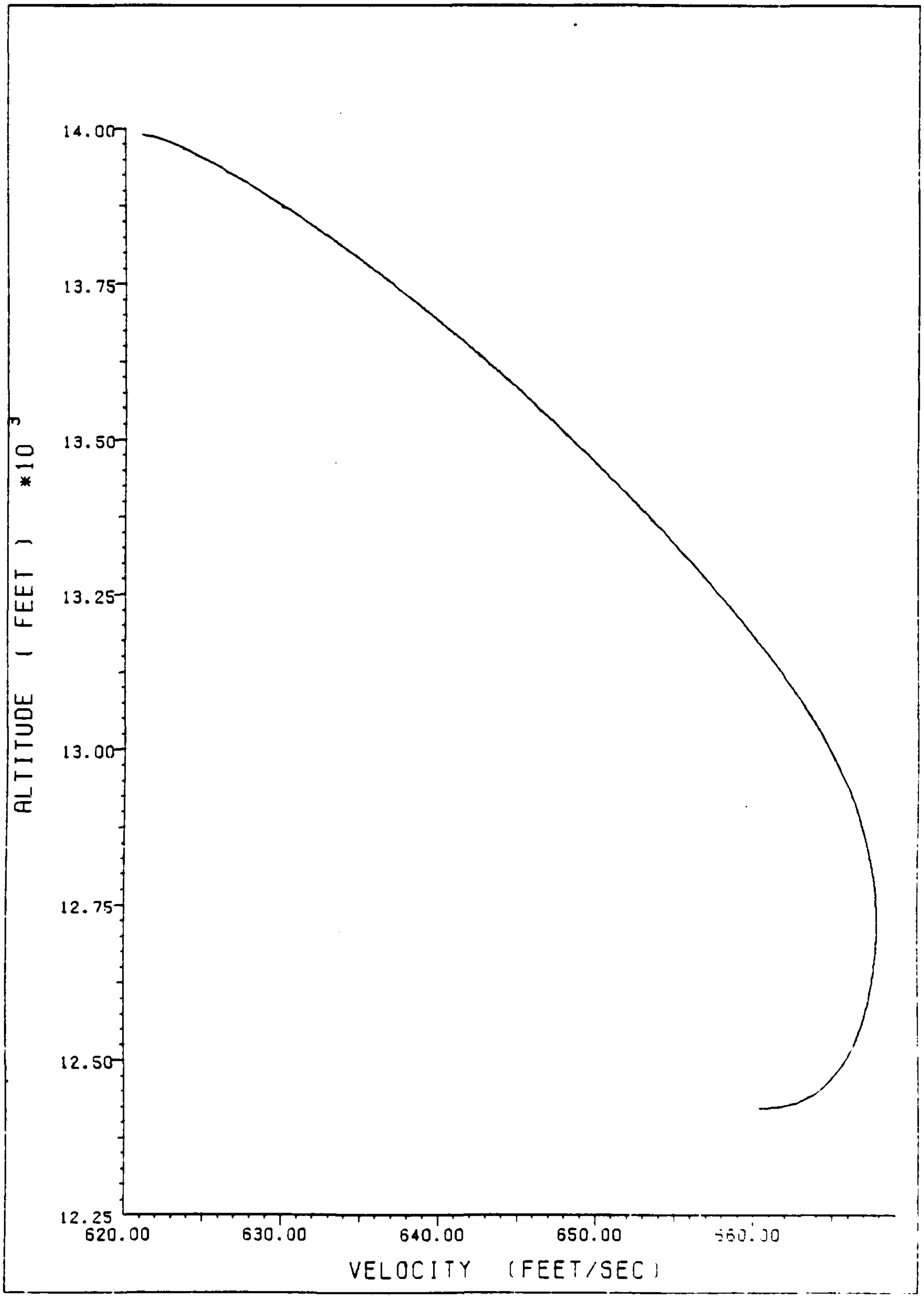


FIG 46. ALTITUDE VS. VELOCITY FOR CASE 10

Appendix G : Summary of Previous Results

Table VII

Humphreys, Hennig, Bolding, Helgeson (4)

(Three-Dimensional Turns)

(All initial altitudes 13,390 feet)

V_i (ft/sec)	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Time (sec)
621	794	12300	2719	10.5
903	886	17635	3771	11.2

Table VIII

Johnson (5) (Thrust Reversal)

(all initial altitudes 13,990 feet)

V_i (ft/sec)	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Reverse Thrust	Time (Sec)
621	781	17338	6839	No	9.575
903	674	15603	-4007	No	10.831
903	593	10429	-10778	Yes	10.523
with higher order thrust controls:					
621	728	17297	5553	No	9.554
903	783	17421	282	No	10.605
903	729	12004	-6405	Yes	10.251

Table IX

Finnerty (6) (Vertical Plane)

(All initial altitudes 13,990 ft)

V_i (ft/sec)	V_f (ft/sec)	h_f (ft)	ΔE (ft)	Throttle Control	Time (sec)
Split - S maneuver:					
621	649	10069	-3368	Constant	9.631
621	625	10001	-3911	Linear	9.554
621	677	10041	-2818	Quadratic	9.278
621	644	10053	-3484	Cubic	9.271
903	651	8636	-11448	Constant	10.782
Pull-up maneuver:					
621	747	18257	6949	Constant	9.776
903	689	19523	231	Constant	11.152
903	771	18585	1156	Linear	10.171

Table X

Brinson (7) (Sideforce)

(all initial altitudes 13,990 ft)

nominal aircraft: $(T/W)_{max} = 1.5$

$K_1 = 0.05$

V_i (ft/sec)	T/W	K_1	ΔE (ft)	Sideforce	Time (sec)
420	Nominal	Nominal	7155	No	10.5694
420	Nominal	Nominal	6711	Yes	10.3565
420	0.75	Nominal	--	No	10.5748
420	Nominal	0.22	--	No	10.1153
621	Nominal	Nominal	6606	No	9.5637
621	Nominal	Nominal	5530	Yes	9.4684
621	0.75	Nominal	--	No	9.6101
621	Nominal	0.22	-1007	No	9.3231
903	Nominal	Nominal	-4009	No	10.8261
903	Nominal	Nominal	-4473	Yes	10.6825
903	0.75	Nominal	--	No	10.8261
903	Nominal	0.22	-10910	No	10.5100

Appendix H : Program Listing

Program Length and Characteristics

Results were obtained by running this program on the Aeronautical System Division's CDC CYBER 845 computer in both interactive and batch modes. Execution times were on the order of one cp second per iteration.

The maximum amount of labelled common required for any part of the program was 7,844 words for the main program. Total compilation time for the main program and all subroutines was 4.3 cp seconds.

The following characteristics are given, broken down by main program and subroutines. Program length is the word length of the program/subroutine including code, storage for local variables, arrays, constants, temporaries, etc., but excluding buffers and common blocks. CM storage is the maximum memory used during compilation, in words.

Table XI
Computer Program Characteristics

Routine	Program Length	CM Storage
Main Program		
STEEPP	5525	32384
Subroutines		
DERIVU	129	26240
DERVUU	49	26240
DERVUX	95	26240
FIGRND	526	27264
FLIMIT	137	26240
FNCTNG	199	26240
FNCTNX	147	26240
FNCTNY	527	27264
POINTE	62	26240

Total Program Length: 7396 words

```
      PROGRAM STEEP  ( INPUT,OUTPUT )
      DIMENSION X (6,200), H (5,5), T (200), XTEMP (6), F (6),
     1   U1 (40), U2 (40), U3 (40), U4 (40), U5 (40),
     2   TU1 (40), TU2 (40), TU3 (40), TU4 (40), TU5 (40),
     3   U (5), XWORK (4,6), YPS (3,200), YO (3,200), XY (6),
     4   YPTEMP (3), YOTEMP (3), FYP (3), FYO (3),
     5   YPWORK (4,3), YOWORK (4,3), YPHIO (3,200),
     6   YPSIO (3,200), GM (3,5,200), UU (5,200), WINV (5,5),
     7   WKAREA (40), GSS (200), GSF (200), GFF (200),
     8   WG (5,3), YU1 (3,1), YU2 (3,1), GT (3,5),
     9   DU (5,200), TEMPY (3,1), UTU1 (5,1), UTU2 (5,1)
      DIMENSION  DT1 (3,1), DT2 (1,5), DTG (1,1), DTSUM (200),
     1           DUT (5,1), TUU (200), XSCR (5), UACT (5)
      COMMON / ONE /  C1,C2,C3,G,CDO,XK1,CLA,TOWMAX,XNMAX,ALFMAX
      COMMON / TWO / U1, U2, U3, U4, U5, TU1, TU2, TU3, TU4, TU5
      COMMON / THREE / TAOALO, TSSLO, TUU
      COMMON / FOUR / X, T, KOUNT, UU, GM
      COMMON / FIVE / YPHIO, YPSIO, GSS, GSF, GFF
      COMMON / SIX / LIMITR, LMTVC, LMTPI
      COMMON / SEVEN / WINV
      COMMON / EIGHT / DELTA, DT, CRITER
      INTEGER  TAOALO,TSSLO,KOUNT,KOUNTY,LIMITR,LMTVC,LMTPI
      REAL     ISISI, ISIFI, IFIFI
*
C   INPUT DATA
*
C   STATE VARIABLE INITIAL CONDITIONS
*
      READ *, ( X (I,1), I = 1, 6 )
*
```

```
C   DELTA TIME FOR NUMERICAL DIFFERENTIATION
*
      READ *, DELTA
*
C   COUNTER FOR SHORT PRINTOUT
*
      READ *, KSHORT
*
C   COUNTER FOR LONG PRINTOUT
*
      READ *, KLONG
*
C   THRUST ANGLE OF ATTACK LOCKOUT
*
      READ *, TAOALO
*
C   THRUST SIDESLIP LOCKOUT
*
      READ *, TSSLO
*
C   TIME INCREMENT FOR INTEGRATION
*
      READ *, DT
*
C   MAXIMUM NUMBER OF ITERATIONS
*
      READ *, ITRMAX
*
C   MAXIMUM CHANGE IN FINAL TIME PER ITERATION
*
      READ *, DTMAX
*
C   HEADING AND FLIGHT PATH ANGLES CONVERGENCE CRITERION
C      ( IN RADIANS )
```

```
*
      READ *, CRITER
*
C    GRADIENT CONVERGENCE CRITERION
*
      READ *, CONVRG
*
*
C    NOMINAL CONTROL VARIABLE TIME HISTORIES
C     - BREAKPOINT TABLES AND DATA TABLES -
*
*
C    BANK ANGLE
*
      READ *, ( TU1 (I), I =  1,10 )
      READ *, (  U1 (I), I =  1,10 )
      READ *, ( TU1 (I), I = 11,20 )
      READ *, (  U1 (I), I = 11,20 )
      READ *, ( TU1 (I), I = 21,30 )
      READ *, (  U1 (I), I = 21,30 )
      READ *, ( TU1 (I), I = 31,40 )
      READ *, (  U1 (I), I = 31,40 )
*
C    ANGLE OF ATTACK
*
      READ *, ( TU2 (I), I =  1,10 )
      READ *, (  U2 (I), I =  1,10 )
      READ *, ( TU2 (I), I = 11,20 )
      READ *, (  U2 (I), I = 11,20 )
      READ *, ( TU2 (I), I = 21,30 )
      READ *, (  U2 (I), I = 21,30 )
      READ *, ( TU2 (I), I = 31,40 )
      READ *, (  U2 (I), I = 31,40 )
*
```

```
C     THROTTLE
*
      READ *, ( TU3 (I), I =  1,10 )
      READ *, (  U3 (I), I =  1,10 )
      READ *, ( TU3 (I), I = 11,20 )
      READ *, (  U3 (I), I = 11,20 )
      READ *, ( TU3 (I), I = 21,30 )
      READ *, (  U3 (I), I = 21,30 )
      READ *, ( TU3 (I), I = 31,40 )
      READ *, (  U3 (I), I = 31,40 )
*
C     THRUST ANGLE OF ATTACK
*
      READ *, ( TU4 (I), I =  1,10 )
      READ *, (  U4 (I), I =  1,10 )
      READ *, ( TU4 (I), I = 11,20 )
      READ *, (  U4 (I), I = 11,20 )
      READ *, ( TU4 (I), I = 21,30 )
      READ *, (  U4 (I), I = 21,30 )
      READ *, ( TU4 (I), I = 31,40 )
      READ *, (  U4 (I), I = 31,40 )
*
C     THRUST SIDESLIP
*
      READ *, ( TU5 (I), I =  1,10 )
      READ *, (  U5 (I), I =  1,10 )
      READ *, ( TU5 (I), I = 11,20 )
      READ *, (  U5 (I), I = 11,20 )
      READ *, ( TU5 (I), I = 21,30 )
      READ *, (  U5 (I), I = 21,30 )


      READ *, ( TU5 (I), I = 31,40 )
      READ *, (  U5 (I), I = 31,40 )
```

```
*
C     VARIOUS PHYSICAL CONSTANTS
*
      READ *, RHO0 , S, WEIGHT, TOWMAX
      READ *, CD0, CLA, XK1, XNMAX, ALFMAX
*
C     MAXIMUM CONTROL VARIABLE DEVIATIONS PER ITERATION
*
      READ *, DPHI , DALPHA, DPI, DEPS, DNU
*
*
C     ECHO DATA BACK
*
*
      PRINT 101, X(1,1), X(2,1), X(3,1), X(4,1), X(5,1), X(6,1)
  101 FORMAT (//1X,"STATE VARIABLE INITIAL CONDITIONS"//
     1        1X,"X (1,1) = ",F9.2,10X,"  X - DISTANCE "//
     2        1X,"X (2,1) = ",F9.2,10X,"  Y - DISTANCE "//
     3        1X,"X (3,1) = ",F9.2,10X,"  ALTITUDE "//
     4        1X,"X (4,1) = ",F9.2,10X,"  VELOCITY "//
     5        1X,"X (5,1) = ",F9.2,10X,"  HEADING ANGLE "//
     6        1X,"X (6,1) = ",F9.2,10X,"  FLIGHT PATH ANGLE "//)
*
      PRINT 120, DELTA, KSHORT, KLONG
  120  FORMAT (//1X,"DELTA  = ",F10.5,8X,"INCREMENT FOR ",
     1        "NUMERICAL DERIVATIVES"//1X,"KSHORT = ",I4,14X,
     2        "COUNTER FOR SHORT PRINTOUTS"//1X,"KLONG  = ",
     3        I4,14X,"COUNTER FOR LONG  PRINTOUTS")
```

```
*
      PRINT 130, TAOALO, TSSLO, DT, ITRMAX, DTMAX, CRITER, CONVRG
  130 FORMAT (//1X,"TAOALO = ",I4,14X,"THRUST ANGLE OF ATTACK LOCKOUT"/
     1       /1X,"TSSLO  = ",I4,14X,"THRUST    SIDESLIP      LOCKOUT"/
     2       /1X,"DT       = ",F9.4,9X,"TIME INCREMENT FOR INTEGRATION"/
     3       /1X,"ITRMAX = ",I4,14X,"MAXIMUM NUMBER OF ",
     4        "ITERATIONS"//1X,"DTMAX  = ",F8.3,10X,
     5        "MAXIMUM CHANGE IN FINAL TIME PER ITERATION"//
     6        1X,"CRITER = ",F10.5,8X,"ANGLE CONVERGENCE ",
     7        "CRITERION"//1X,"CONVRG = ",F10.5,8X,
     8        "GRADIENT CONVERGENCE CRITERION"/)
*
*
      PRINT 170, RHOO,S,WEIGHT,TOWMAX,CD0,CLA,XK1,XNMAX,ALFMAX
  170 FORMAT (///1X,"RHOO      = ",F15.7,10X,"SEA LEVEL DENSITY"/
     1          /1X,"S         = ",F10.2,15X,"REFERENCE WING AREA"/
     2          /1X,"WEIGHT    = ",F10.2,15X,"AIRCRAFT WEIGHT"/
     3          /1X,"(T/W)MAX = ",F10.2,15X,"MAX THRUST-TO-WEIGHT"/
     4          /1X,"CD0       = ",F12.4,13X,"PARASITE DRAG COEFFICIENT"/
     5          /1X,"CLA       = ",F12.4,13X,"LIFT CURVE SLOPE"/
     6          /1X,"XK1       = ",F12.4,13X,"INDUCED DRAG PARAMETER"/
     7          /1X,"XNMAX     = ",F12.4,13X,"MAXIMUM LOAD FACTOR"/
     8          /1X,"ALFMAX    = ",F12.4,13X,"ANGLE OF ATTACK LIMIT"///)
*
*
      PRINT 180, DPHI, DALPHA, DPI, DEPS, DNU
  180 FORMAT (///1X,"MAXIMUM CONTROL VARIABLE DEVIATIONS ",
     1        "PER STEP"//1X,"(ALL IN DEGREES EXCEPT THROTTLE)"
     2        //1X,"DPHI   = ",F9.4,10X,"BANK ANGLE"//1X,
     3        "DALPHA = ",F9.4,10X,"ANGLE OF ATTACK"//1X,
     4        "DPI    = ",F9.4,10X,"THROTTLE"//1X,
     5        "DEPS   = ",F9.4,10X,"THRUST ANGLE OF ATTACK"//
     6        1X,"DNU    = ",F9.4,10X,"THRUST SIDESLIP"///)
*
```

```
*
C     NOMINAL CONTROL VARIABLE TIME HISTORIES
C       - BREAKPOINT TABLES AND DATA POINTS -
*
*
        PRINT 1003
 1003   FORMAT (///1X,"   NOMINAL CONTROL VARIABLE TIME HISTORIES"
     1          //1X,"     - BREAKPOINT TABLES AND DATA POINTS -"////
     2         1X,1X,"TIME",9X,"BANK",10X,"TIME",11X,"ANGLE OF"/
     3         1X,1X,"(SEC)",8X,"ANGLE",9X,"(SEC)",11X,"ATTACK"//)
*
        DO 140 I = 1,40
 140       PRINT 141, TU1(I), U1(I), TU2(I), U2(I)
 141       FORMAT (1X,F6.3,6X,F8.3,6X,F7.3,9X,F8.3)
*
        PRINT 1007
 1007   FORMAT (//1X,1X,"TIME",7X,"THROTTLE",9X,"TIME",10X,
     1         "THRUST",10X,"TIME",9X,"THRUST"/1X,1X,
     2         "(SEC)",23X,"(SEC)",9X,"ALPHA",11X,"(SEC)",
     3         7X,"SIDESLIP"/)
*
        DO 160 I = 1,40
 160       PRINT 161, TU3(I), U3(I), TU4(I), U4(I), TU5(I), U5(I)
 161       FORMAT (1X,F6.3,5X,F8.3,7X,F8.3,8X,F8.3,8X,F6.3,7X,F8.3)
*
*
*
C     NOW ONTO THE NUMBER CRUNCHING
*
*
```

```
      ENDIF
*
*
C     CALCULATE CHANGE IN CONTROL VARIABLE PROGRAM
*
*
      DBETA = DPSI
*
C     CALCULATE W - INVERSE * G - TRANSPOSE
*
C     ZERO MATRICES FIRST
*
      DO 710 I = 1, 3
          DO 720 J = 1, 5
                 GT (I,J) = 0.
                 WG (J,I) = 0.
 720      CONTINUE
 710  CONTINUE
*
      DO 721 I = 1, 5
          DO 722 J = 1, 200
 722             DU (I,J) = 0.
 721  CONTINUE
*
*
      DO 730 I = 1, KOUNT
          DO 740 J = 1, 3
                 YU1 (J,1) = YPSIO (J,I)
                 YU2 (J,1) = YPHIO (J,I)
                 DO 750 K = 1, 5
 750                 GT (J,K) = GM (J,K,I)
 740      CONTINUE
*
*
```

```
      ISISI = 0.
      ISIFI = 0.
      IFIFI = 0.
*
      DO 700 I = 2, KOUNT
         DTIME = ( T (I) - T (I-1) ) * 0.5
*
         ISISI = ISISI + ( GSS (I) + GSS (I-1) ) * DTIME
         ISIFI = ISIFI + ( GSF (I) + GSF (I-1) ) * DTIME
 700     IFIFI = IFIFI + ( GFF (I) + GFF (I-1) ) * DTIME
*
*
C    PICK DELTA - PSI
*
      DPSI = - 1. * X (6,KOUNT)
*
      TEMPSI = DPSI * DEGRAD
*
*
C    PICK DP
*
      DPSQ    = T (KOUNT) * ( 2. * X (6,KOUNT) )**2
*
      DP      = SQRT ( DPSQ )
*
*
      TEMP    = DPSQ - ( DPSI * DPSI / ISISI )
      IF  ( TEMP .LT. 0. )  THEN
         DPSI = DP * SQRT ( ISISI )
         TEMP = 0.
```

```
*
      DO 640 I = 1,3
          DO 650 J = 1, 5
                 DO 660 K = 1, 200
660                   GN (I,J,K) = 0.
650       CONTINUE
640   CONTINUE
*
*
C    NOW CALCULATE G MATRIX
*
*
      CALL FNCTNG
*
*
*
*
*
C    COMPUTE INTEGRANDS FOR ISISI, ISIFI, IFIFI
C         ( GSS, GSF, GFF )
*
*
      CALL FIGRND  ( IERROR )
*
      IF  ( IERROR .NE. 0 )  THEN
          PRINT 690, IERROR
690       FORMAT (///1X,"PPOBLEM IN SUBROUTINE FIGRND"//1X,
     1               "IERROR = ",I4//1X,"PROGRAM STOPPED")
          STOP
      ENDIF
*
*
C    COMPUTE ISISI, ISIFI, IFIFI
*
```

```
*
      CALL FNCTNX  ( XTEMP, U, F )
      OMGDOT = F (5)
      PSIDOT = F (6)
*
*
C   ZERO MATRICES
*
*
      DO 610 I = 1, 3
         DO 620 J = 1, 200
                YPHIO (I,J) = 0.
  620           YPSIO (I,J) = 0.
  610 CONTINUE
*
*
      DO 630 I = 1, KOUNT
         YPHIO (1,I) = YO (2,I) / OMGDOT
         YPHIO (2,I) =        1. / OMGDOT
         YPHIO (3,I) = YO (3,I) / OMGDOT
*
         YPSIO (1,I) = YPS (2,I) - YPHIO (1,I) * PSIDOT
         YPSIO (2,I) =           - YPHIO (2,I) * PSIDOT
  630    YPSIO (3,I) = YPS (3,I) - YPHIO (3,I) * PSIDOT
*
*
C   CALCULATE G MATRIX
*
*
C   ZERO MATRIX FIRST
```

```
*
      TX = T (KOUNTY) - DTY
      CALL DERVUX  ( TX, XY, U )
      CALL FNCTNY  ( XY, U, YPTEMP, YOTEMP, FYP, FYO )
      DO 570 I = 1, 3
          YPWORK (4,I) = DTY * FYP (I)
570       YOWORK (4,I) = DTY * FYO (I)
*
*
      DO 580 I = 1, 3
          YPS (I,KOUNTY-1) = YPS (I,KOUNTY) - ( YPWORK (1,I) +
     1                       2. * YPWORK (2,I) + 2. * YPWORK (3,I) +
     2                       YPWORK (4,I) ) / 6.
*
          YO  (I,KOUNTY-1) = YO  (I,KOUNTY) - ( YOWORK (1,I) +
     1                       2. * YOWORK (2,I) + 2. * YOWORK (3,I) +
     2                       YOWORK (4,I) ) / 6.
580   CONTINUE
*
*
      IF  ( KOUNTY .NE. 2 )  THEN
          KOUNTY = KOUNTY - 1
          GO TO 520
      ENDIF
*
*
C   COMPUTE LAMBDA PHI-OMEGA, LAMBDA PSI-OMEGA
*
*
      DO 590 I = 1, 5
590   U (I) = UU (I,KOUNT)
*
      DO 600 I = 1, 6
600       XTEMP (I) = X (I,KOUNT)
```

```
  520 DO 530 I = 1, 3
          YPTEMP (I) = YPS (I,KOUNTY)
  530     YOTEMP (I) = YO  (I,KOUNTY)
*
      DTY = T (KOUNTY) - T (KOUNTY-1)
*
      TX = T (KOUNTY)
      CALL DERVUX  ( TX, XY, U )
      CALL FNCTNY  ( XY, U, YPTEMP, YOTEMP, FYP, FYO )
      DO 540 I = 1, 3
          YPWORK (1,I) = DTY * FYP (I)
          YOWORK (1,I) = DTY * FYO (I)
          YPTEMP (I)   = YPS (I,KOUNTY) - 0.5 * YPWORK (1,I)
  540     YOTEMP (I)   = YO  (I,KOUNTY) - 0.5 * YOWORK (1,I)
*
      TX = T (KOUNTY) - 0.5 * DTY
      CALL DERVUX  ( TX, XY, U )
      CALL FNCTNY  ( XY, U, YPTEMP, YOTEMP, FYP, FYO )
      DO 550 I = 1, 3
          YPWORK (2,I) = DTY * FYP (I)
          YOWORK (2,I) = DTY * FYO (I)
          YPTEMP (I)   = YPS (I,KOUNTY) - 0.5 * YPWORK (2,I)
  550     YOTEMP (I)   = YO  (I,KOUNTY) - 0.5 * YOWORK (2,I)
*
      CALL FNCTNY  ( XY, U, YPTEMP, YOTEMP, FYP, FYO )
      DO 560 I = 1, 3
          YPWORK (3,I) = DTY * FYP (I)
          YOWORK (3,I) = DTY * FYO (I)
          YPTEMP (I)   = YPS (I,KOUNTY) - YPWORK (3,I)
  560     YOTEMP (I)   = YO  (I,KOUNTY) - YOWORK (3,I)
```

```
            X5DEG = X (5,I) * DEGRAD
            X6DEG = X (6,I) * DEGRAD
            PRINT 451, T (I), X (1,I), X (2,I), X (3,I),
     1                 X (4,I), X5DEG, X6DEG, T (I)
  451       FORMAT (1X,F8.3,5X,F11.3,3X,F11.3,4X,F11.3,7X,
     1              F9.3,5X,F9.4,5X,F9.4,7X,F8.3)
  450   CONTINUE
*
*
*
  460   CONTINUE
*
*
*
C    COMPUTE ADJOINT EQUATIONS BY INTEGRATING BACKWARDS
C      - LAMBDA'S PSI AND OMEGA       ( YPS AND YO )
C        - ( LAMBDA PHI = ZERO FOR ALL TIME )
*
C      - YPS "X" = YPS "Y" = YPS "CHI" = 0
C      - YO  "X" = YO  "Y" = 0                YO "CHI" = 1
*
*
C    ZERO MATRICES
*
        DO 500 I = 1, 3
            DO 510 J = 1, 200
                   YPS (I,J) = 0.
  510              YO  (I,J) = 0.
  500   CONTINUE
*
        KOUNTY = KOUNT
        YPS (3,KOUNTY) = 1.
*
*
```

```
*
      DO 420 I = 1, KOUNT
         U1DEG = UU (1,I) * DEGRAD
         U2DEG = UU (2,I) * DEGRAD
         U4DEG = UU (4,I) * DEGRAD
         U5DEG = UU (5,I) * DEGRAD
         CALL DERVUX  ( T (I), XSCR, UACT )
         U1ACT = UACT (1) * DEGRAD
         U2ACT = UACT (2) * DEGRAD
         U4ACT = UACT (4) * DEGRAD
         U5ACT = UACT (5) * DEGRAD
         PRINT 421, T (I), U1DEG, U1ACT, U2DEG, U2ACT,
     1              UU (3,I), UACT (3), U4DEG, U4ACT,
     2              U5DEG, U5ACT, T (I)
  421    FORMAT (1X,F7.3,3X,F8.3," / ",F8.3,3X,F8.3,
     1          " / ",F8.3,3X,F8.3," / ",F8.3,3X,
     2          F8.3," / ",F8.3,3X,F8.3," / ",F8.3,3X,F7.3)
  420 CONTINUE
*
*
      PRINT 1012
 1012 FORMAT (////1X,3X,"TIME",11X,"X(1)",10X,"X(2)",10X,"X(3)",
     1        13X,"X(4)",9X,"X(5)",11X,"X(6)",12X,"TIME"/
     2        1X,3X,"(SEC)",7X,"X-DISTANCE",4X,"Y-DISTANCE",5X,
     3        "ALTITUDE",9X,"VELOCITY",6X,"HEADING",6X,
     4        "FLIGHT PATH",8X,"(SEC)"/18X,"(FEET)",8X,
     5        "(FEET)",6X,"(FEET)",10X,"(FT/SEC)",7X,
     6        "(DEG)",10X,"(DEG)"//)
*
      DO 450 I = 1, KOUNT
```

```
         IPRINT = 1
         KPRNTS = 1
      ELSE
         IPRINT = 0
         KPRNTS = KPRNTS + 1
      ENDIF
*
      IF  ( ( ITER .EQ. 1 ) .OR. ( ITER .EQ. ITRMAX ) )  THEN
         IPRINT = 1
         KPRNTL = KPRNTL + 1
         GO TO 1000
      ENDIF
*
      IF  ( KPRNTL .EQ. KLONG )  THEN
         KPRNTL = 1
         IPRINT = 1
      ELSE
         KPRNTL = KPRNTL + 1
         GO TO 460
      ENDIF
*
*
 1000 PRINT 1010, ITER
 1010 FORMAT (/////1X,"****************************************"///
     1        1X,"ITERATION NUMBER  ",I4///7X,"CONTROL VARIABLE",
     2        " TIME HISTORIES",15X,"( ALL IN DEGREES EXCEPT ",
     3        "THROTTLE )"//3X,"AND  STATE  VARIABLE TIME ",
     4        "HISTORIES",15X,"( DESIRED  /  ACTUAL )"///)
      PRINT 1011
 1011 FORMAT (1X,2X,"TIME",11X,"BANK",16X,"ANGLE OF",
     1        14X,"THROTTLE",16X,"THRUST",15X,
     2        "THRUST",12X,"TIME"/1X,2X,"(SEC)",10X,
     3        "ANGLE",16X,"ATTACK",39X,
     4        "ALPHA",15X,"SIDESLIP",11X,"(SEC)"//)
```

```
*
C    PRINT CONTROL VARIABLES AND TRAJECTORY
*
*
 400  CONTINUE
*
      IF  ( ITER .EQ. 1 )  THEN
          DO 401 I = 1, 200
              DO 402 J = 1, 5
 402              UU (J,I) = 0.
 401      TUU (I) = 0.
      ENDIF
*
      DO 403 I = 1, KOUNT
          IF  ( ITER .EQ. 1 )  THEN
              CALL DERIVU  ( T (I), U )
              DO 404 J = 1, 5
 404                 UU (J,I) = U (J)
          ELSE
                 IF  ( I .GT. KNTLST )  THEN
                     CALL DERVUU  ( T (I), KNTLST, U )
                     DO 430 J = 1, 5
 430                     UU (J,I) = U (J)
                 ENDIF
          ENDIF
 403  TUU(I) = T (I)
*
*
C    CHECK TO SEE IF WANT TO PRINT TIME HISTORIES
*
*
      IF  ( KPRNTS .EQ. KSHORT )  THEN
```

```
*
*
      DO 280 I = 1,6
           X (I,KOUNT+1) = X (I,KOUNT) + ( XWORK (1,I) + 2. * XWORK (2,I)
     1                   + 2. * XWORK (3,I) + XWORK (4,I) ) / 6.
  280 CONTINUE
*
*
C  TEST HEADING ANGLE 180 DEGREES
*
*
      IF  ( ( ABS( X(5,KOUNT+1)) - PI ) .GT. CRITER )  THEN
           TEMP1 = X (5,KOUNT+1) - X (5,KOUNT)
           TEMP2 = ABS (TEMP1)
           DTX   = ( PI - ABS (X(5,KOUNT)) ) * DTX / TEMP2
           GO TO 310
      ELSE
           KOUNT    = KOUNT + 1
           TIME     = TIME  + DTX
           T(KOUNT) = TIME
      ENDIF
*
      IF  ( ( ABS  ( ABS (X(5,KOUNT)) - PI ) )  .LT. CRITER )  GO TO 400
      IF  ( KOUNT .LT. 200 )  GO TO 310
*
      IF  ( KOUNT .EQ. 200 )  THEN
           PRINT 470, KOUNT
  470      FORMAT (////1X,"KOUNT = ",I4///1X,
     1                "TRY A BETTER NOMINAL TRAJECTORY")
           STOP
      ENDIF
*
```

```
      CALL FNCTNX   ( XTEMP, U, F )
      DO 240 I = 1,6
         XWORK (1,I) = DTX * F (I)
240      XTEMP (I) = X (I,KOUNT) + 0.5 * XWORK (1,I)
*
      TX = TIME + 0.5 * DTX
      IF  ( ITER .EQ. 1 ) THEN
         CALL DERIVU  ( TX, U )
      ELSE
         CALL DERVUU  ( TX, KNTLST, U )
      ENDIF
      CALL FNCTNX   ( XTEMP, U, F )
      DO 250 I = 1,6
         XWORK (2,I) = DTX * F (I)
250      XTEMP (I) = X (I,KOUNT) + 0.5 * XWORK (2,I)
*
      CALL FNCTNX   ( XTEMP, U, F )
      DO 260 I = 1,6
         XWORK (3,I) = DTX * F (I)
260      XTEMP (I) = X (I,KOUNT) + XWORK (3,I)
*
      TX = TIME + DTX
      IF  ( ITER .EQ. 1 ) THEN
         CALL DERIVU  ( TX, U )
      ELSE
         CALL DERVUU  ( TX, KNTLST, U )
      ENDIF
      CALL FNCTNX   ( XTEMP, U, F )
      DO 270 I = 1,6
270      XWORK (4,I) = DTX * F (I)
```

```
 227      U5 (I) = U5 (I) * RADDEG
*
*
C   RUNGE - KUTTA FOURTH ORDER INTEGRATION
C      TO COMPUTE NOMINAL TRAJECTORY
*
       ITER   = 1
       KPRNTS = 1
       KPRNTL = 1
*
 999   KOUNT = 1
       TIME  = 0.
       DTX   = DT
*
C   ZERO MATRICES
*
       DO 200 I = 1,6
          DO 210 J = 2,200
 210            X (I,J) = 0.
 200   CONTINUE
*
       DO 220 I = 1,200
 220      T (I) = 0.
*
 310   DO 230 I = 1,6
 230      XTEMP (I) = X (I,KOUNT)
*
*
       TX = TIME
       IF  ( ITER .EQ. 1 ) THEN
          CALL DERIVU  ( TX, U )
       ELSE
          CALL DERVUU  ( TY, KNTLST, U )
       ENDIF
```

```
*
C     CONSTANTS
*
          C1 = RHOO * S * 0.5 / WEIGHT
          C2 = 0.0000006882
          C3 = 1. / 0.235
          PI = ACOS (-1.)
          PIMNS = -1. * PI
          G  = 32.131
*
          DEGRAD = 180. / PI
          RADDEG = 1. / DEGRAD
*
*
C     SET UP IDENTITY WEIGHTING MATRIX  ( INVERSE )
*
          DO 10 I = 1, 5
              DO 20 J = 1, 5
 20                  WINV (I,J) = 0.
 10       CONTINUE
*
          DO 30 I = 1, 5
 30           WINV (I,I) = 1.
*
*
C     CONVERT DATA FOR BANK ANGLE, ANGLE OF ATTACK,
C     THRUST ANGLE OF ATTACK, AND THRUST SIDESLIP
C     FROM DEGREES TO RADIANS
C        ( DATA INPUT AS DEGREES )
*
          DO 227, I = 1, 40
              U1 (I) = U1 (I) * RADDEG
              U2 (I) = U2 (I) * RADDEG
              U4 (I) = U4 (I) * RADDEG
```

```
      CALL DERVUX  ( T (I), XY, U )
      IF  ( LIMITR .EQ. 1 )  THEN
                WINV (2,2) = 0.
      ELSE
                WINV (2,2) = 1.
      ENDIF
*
      IF  ( LMTPI .EQ. 1 )  THEN
                WINV (3,3) = 0.
      ELSE
                WINV (3,3) = 1.
      ENDIF
*
*
      CALL VMULFP ( WINV, GT, 5, 5, 3, 5, 3, WG, 5, IERWG )
      IF  ( IERWG .NE. 0 )  THEN
                PRINT 751, IERWG
751             FORMAT (///1X,"ERROR WHEN MULTIPLIED W-INV * G-T"
     1                 ///1X,"IERWG = ",I4)
                PRINT 752, I
752             FORMAT (///1X,"INDEX FOR DO LOOP 730 = ",I4)
                STOP
      ENDIF
*
      DO 760 J = 1, 3
760             TEMPY (J,1) = YU2 (J,1) - YU1 (J,1) * ISIFI / ISISI
*
      CALL VMULFF ( WG, TEMPY, 5, 3, 1, 5, 3, UTU1, 5, IERUT1 )
      CALL VMULFF ( WG,  YU1 , 5, 3, 1, 5, 3, UTU2, 5, IERUT2 )
      IF  ( IERUT1 .NE. 0 )  THEN
```

```
                    PRINT 761, IERUT1
                    PRINT 752, I
 761                FORMAT (///1X,"ERROR WHEN MULTIPLIED W-INV * G-T *",
     1                     "TEMPY"///1X,"IERUT1 = ",I4)
                    STOP
               ELSEIF  ( IERUT2 .NE. 0 )  THEN
                    PRINT 762, IERUT2
                    PRINT 752, I
 762                FORMAT (///1X,"ERROR WHEN MULTIPLIED W-INV * G-T *",
     1                     "YU1"///1X,"IERUT2 = ",I4)
                    STOP
               ENDIF
*
*
               DO 770 J = 1, 5
                    DU (J,I) = UTU1 (J,1) * SQRT ( TEMP / ( IFIFI -
     1                     ISIFI * ISIFI / ISISI ) ) +
     2                     UTU2 (J,1) * DBETA / ISISI
 770           CONTINUE
*
 730       CONTINUE
*
*
C     LIMIT CONTROL INCREMENTS IF ON BOUNDARY
*
*
           DO 771 I = 1, KOUNT
               SIGMA = ( 1. - C2 * X (3,I) )**C3
               VTEMP = XNMAX / ( C1 * CLA )
               VC    = SQRT ( VTEMP / ( SIGMA * ALFMAX ) )
*
               IF  ( X (4,I) .LE. VC )  THEN
                    ALFBND = ALFMAX
               ELSE
```

```
                ALFBND = VTEMP / ( SIGMA * X (4,I) * X (4,I) )
            ENDIF
*
            IF  ( ( UU (2,I) + DU (2,I) ) .GT. ALFBND )
     1          DU (2,I) = ALFBND - UU (2,I)
*
            IF  ( ( UU (3,I) + DU (3,I) ) .GT. 1. )
     1          DU (3,I) = 1. - UU (3,I)
            IF  ( ( UU (3,I) + DU (3,I) ) .LT. 0. )
     1          DU (3,I) = -1. * UU (3,I)
*
            IF  ( ( UU (4,I) + DU (4,I) ) .GT. PI )
     1          DU (4,I) = PI - UU (4,I)
            IF  ( ( UU (4,I) + DU (4,I) ) .LT. PIMNS )
     1          DU (4,I) = PIMNS - UU (4,I)
*
            IF  ( ( UU (5,I) + DU (5,I) ) .GT. PI )
     1          DU (5,I) = PI - UU (5,I)
            IF  ( ( UU (5,I) + DU (5,I) ) .LT. PIMNS )
     1          DU (5,I) = PIMNS - UU (5,I)
*
  771   CONTINUE
*
*
C     CALCULATE DELTAT
*
C     ZERO MATRICES FIRST
*
        DO 780 I = 1, 200
            TUU (I) = 0.
```

```
 780       DTSUM (I) = 0.
*
      DO 781 I = 1, 3
         DO 782 J = 1, 5
 782            GT (I,J) = 0.
 781  CONTINUE
*
*
      DT1 (2,1) = 1.
*
 790  DO 800 I = 1, KOUNT
         DT1 (1,1) = YO (2,I)
         DT1 (3,1) = YO (3,I)
*
         DO 810 J = 1,3
            DO 820 K = 1, 5
               DUT (K,1) = DU (K,I)
                GT (J,K) = GM (J,K,I)
 820
 810     CONTINUE
*
         CALL VMULFM ( DT1, GT, 3, 1, 5, 3, 3, DT2, 1, IERDT2 )
         IF  ( IERDT2 .NE. 0 )  THEN
            PRINT 821, IERDT2
            PRINT 822, I
 821        FORMAT (///1X,"ERROR WHEN MULTIPLIED YO-T * G"
     1              ///1X,"IERDT2 = ",I4)
 822        FORMAT (///1X,"INDEX FOR DO LOOP 800 = ",I4)
            STOP
         ENDIF
*
         CALL VMULFF ( DT2, DUT, 1, 5, 1, 1, 5, DTG, 1, IERDTG )
         IF  ( IERDTG .NE. 0 )  THEN
            PRINT 823, IERDTG
            PRINT 822, I
```

```
 823              FORMAT (///1X,"ERROR WHEN MULTIPLIED YO-T * G * DU"
     1                   ///1X,"IERDTG = ",I4)
                  STOP
             ENDIF
*
             DTSUM (I) = UTG (1,1)
*
 800  CONTINUE
*
*
C    NOW TO CALCULATE DELTAT
*
*
      DELTAT = 0.
*
      DO 830 I = 2, KOUNT
         DTIME = ( T (I) - T (I-1) ) * 0.5
 830     DELTAT = DELTAT + ( DTSUM (I) + DTSUM (I-1) ) * DTIME
*
      DELTAT = -1. + DELTAT / OMGDOT
*
*
C    IF DELTAT OUT OF TOLERANCE,
C    SCALE DOWN CHANGE IN CONTROL VARIABLES BY 2 %
*
*
      IF  ( ABS (DELTAT) .GT. DTMAX )  THEN
         DO 840 I = 1, KOUNT
              DO 850 J = 1, 5
 850             DU (J,I) = DU (J,I) * 0.98
```

```
 840      CONTINUE
          GO TO 790
       ELSE
          IF  ( IPRINT .EQ. 1 )  THEN
              PRINT 855, ITER, T (KOUNT)
 855          FORMAT (/1X,"ITERATION NUMBER  ",I4,37X,
     1                "FINAL TIME",20X,"=",F8.3)
              PRINT 860, DELTAT, TEMPSI
 860          FORMAT (/1X,"PREDICTED CHANGE IN FINAL TIME = ",
     1                F6.3,20X,"DESIRED DELTA - PSI  ( DEG )  =",
     2                F8.3)
          ENDIF
       ENDIF
*
*
C   GET NEW CONTROL VARIABLE PROGRAM
*
*
       DO 870 I = 1, 200
          TUU (I) = T (I)
          DO 880 J = 1, 5
 880         UU (J,I) = UU (J,I) + DU (J,I)
 870   CONTINUE
*
*
C   CHECK CONVERGENCE
*
*
       CONV = IFIFI - ISIFI * ISIFI / ISISI
       IF  ( IPRINT .EQ. 1 )  THEN
          PRINT 885, DTMAX, CONV
 885      FORMAT (/1X,"MAXIMUM CHANGE ALLOWED ",8X,"= ",F6.3,
     1            20X,"SQUARE OF GRADIENT ",11X,"= ",E15.7/)
       ENDIF
```

```
*
      IF  ( ( ( ABS(X(6,KOUNT)) ) .LT. CRITER ) .AND.
     1      ( ( ABS(CONV) ) .LT. CONVRG ) )  THEN
          FPINT 888
  888     FORMAT (///1X,"******      CONVERGENCE      ******")
          STOP
      ENDIF
*
*
C   INCREMENT COUNTER, CHECK MAXIMUM, LOOP BACK UP
C   AND DO IT ALL AGAIN
*
*
      IF  (ITER .LT. ITRMAX )  THEN
          ITER = ITER + 1
          KNTLST = KOUNT
          GO TO 999
      ELSE
          FRINT 890
  890     FORMAT (///1X,"MAXIMUM NUMBER OF ITERATIONS REACHED"
     1            ///1X,"PROGRAM HALTED"///1X,"FOLLOWING IS",
     2            " CONTROL VARIABLE PROGRAM CALCULATED ",
     3            "FOR NEXT STEP"///)
          FRINT 1010, ITER
          FRINT 1011
          DO 891 I = 1, KOUNT
                U1DEG = UU (1,I) * DEGRAD
                U2DEG = UU (2,I) * DEGRAD
                U4DEG = UU (4,I) * DEGRAD
                U5DEG = UU (5,I) * DEGRAD
```

```
                PRINT 892, T (I), U1DEG, U2DEG, UU (3,I),
     1                U4DEG, U5DEG, T (I)
 892            FORMAT (1X,F8.3,6X,F8.3,7X,F8.3,10X,F8.3,
     1                10X,F8.3,8X,F8.3,7X,F8.3)
 891        CONTINUE
          ENDIF
*
*
          STOP
          END
```

```
      SUBROUTINE DERIVU  ( TIMEU, U )
      DIMENSION U (5)
      DIMENSION U1 (40), U2 (40), U3 (40), U4 (40), U5 (40),
     1  TU1 (40), TU2 (40), TU3 (40), TU4 (40), TU5 (40),
     2  TUU (200)
      COMMON / TWO / U1, U2, U3, U4, U5, TU1, TU2, TU3, TU4, TU5
      COMMON / THREE / TAOALO, TSSLO, TUU
      INTEGER  TAOALO, TSSLO
      INTEGER  POINT1, POINT2, POINT3, POINT4, POINT5
*
*
      CALL POINTE  ( TIMEU, 40, POINT1, FRACT1, TU1 )
      CALL POINTE  ( TIMEU, 40, POINT2, FRACT2, TU2 )
      CALL POINTE  ( TIMEU, 40, POINT3, FRACT3, TU3 )
      CALL POINTE  ( TIMEU, 40, POINT4, FRACT4, TU4 )
      CALL POINTE  ( TIMEU, 40, POINT5 , FRACT5, TU5 )
*
      U(1) = ( U1(POINT1+1) - U1(POINT1) ) * FRACT1 + U1(POINT1)
      U(2) = ( U2(POINT2+1) - U2(POINT2) ) * FRACT2 + U2(POINT2)
      U(3) = ( U3(POINT3+1) - U3(POINT3) ) * FRACT3 + U3(POINT3)
*
      IF  ( TAOALO .EQ. 1 )  THEN
          U (4) = 0.
      ELSE
          U(4) = ( U4(POINT4+1) - U4(POINT4) ) * FRACT4 + U4(POINT4)
      ENDIF
*
      IF  ( TSSLO  .EQ. 1 )  THEN
          U (5) = 0.
      ELSE
          U(5) = ( U5(POINT5+1) - U5(POINT5) ) * FRACT5 + U5(POINT5)
      ENDIF
*
      RETURN
      END
```

```
      SUBROUTINE DERVUU  ( TIMEUU, LENGTH, U )
      DIMENSION  U (5), TUU (200)
      DIMENSION  X (6,200), T (200), UU (5,200), GM (3,5,200)
      COMMON / THREE / TAOALO, TSSLO, TUU
      COMMON / FOUR /  X, T, KOUNT, UU, GM
      INTEGER  TAOALO, TSSLO, KOUNT
      INTEGER  LENGTH, POINT
*
*
      CALL POINTE  ( TIMEUU, LENGTH, POINT, FRACT, TUU )
*
      DO 10 I = 1, 5
   10   U (I) = ( UU(I,POINT+1) - UU(I,POINT) ) * FRACT + UU(I,POINT)
*
      RETURN
      END
```

```
      SUBROUTINE DERVUX  ( TIMEX, XY, U )
      DIMENSION XY (6), U (5), UF (5), TUU (200)
      DIMENSION X (6,200), T (200), UU (5,200), GM (3,5,200)

      COMMON / ONE / C1,C2,C3,G,CD0,XK1,CLA,TOWMAX,XNMAX,ALFMAX
      COMMON / THREE / TAOALO, TSSLO, TUU
      COMMON / FOUR / X, T, KOUNT, UU, GM
      COMMON / SIX / LIMITR, LMTVC, LMTPI
      INTEGER  KOUNT
      INTEGER  TAOALO, TSSLO, LIMITR, LMTVC, LMTPI
      INTEGER  POINT
*
*
      CALL POINTE  ( TIMEX, KOUNT, POINT, FRACT, T )
*
      DO 10 I = 1, 6
   10    XY (I) = ( X(I,POINT+1) - X(I,POINT) ) * FRACT + X(I,POINT)
*
      DO 20 I = 1, 5
   20    U (I) = ( UU(I,POINT+1) - UU(I,POINT) ) * FRACT + UU(I,POINT)
*
      CALL FLIMIT  ( XY, U, UF )
      DO 30 I = 1, 5
   30    U (I) = UF (I)
*
      RETURN
      END
```

```
      SUBROUTINE FLIMIT  ( XF, U, UF )
      DIMENSION XF (6), U (5), UF (5), TUU (200)
      COMMON / ONE / C1,C2,C3,G,CD0,XK1,CLA,TOWMAX,XNMAX,ALFMAX
      COMMON / THREE / TAOALO, TSSLO, TUU
      COMMON / SIX / LIMITR, LMTVC, LMTPI
      INTEGER  TAOALO, TSSLO, LIMITR, LMTVC, LMTPI
      REAL     MNSPI
*
*
      DO 10 I = 1, 5
 10      UF (I) = U (I)
*
      SIGMA  = ( 1. - C2 * XF(3) )**C3
      VTEMP  = XNMAX / ( C1 * CLA )
      VC     = SQRT ( VTEMP / ( SIGMA * ALFMAX ) )
      PI     = ACOS (-1.)
      MNSPI  = -1. * PI
*
*
      IF  ( XF (4) .LE. VC )  THEN
          ALFLIM = ALFMAX
          LIMITP = 0
      ELSE
          ALFLIM = VTEMP / ( SIGMA * XF (4) * XF (4) )
          LIMITR = 1
      ENDIF
*
      IF  ( ( ABS (XF (4) - VC) ) .LE. .01 )  THEN
          LMTVC = 1
      ELSE
          LMTVC = 0
      ENDIF
*
      IF  ( UF (2) .GT. ALFLIM )  UF (2) = ALFLIM
*
```

```
      IF ( UF (3) .GT. 1. ) THEN
         UF (3) = 1.
         LMTPI = 1
      ELSEIF ( UF (3) .LT. 0. ) THEN
         UF (3) = 0.
         LMTPI = 1


      ELSE
         LMTPI = 0
      ENDIF
*
      IF ( TAOALO .EQ. 1 ) THEN
         UF (4) = 0.
      ELSE
         IF ( UF (4) .GT. PI )     UF (4) = PI
         IF ( UF (4) .LT. MNSPI )  UF (4) = MNSPI
      ENDIF
*
      IF ( TSSLO  .EQ. 1 ) THEN
         UF (5) = 0.
      ELSE
         IF ( UF (5) .GT. PI )     UF (5) = PI
         IF ( UF (5) .LT. MNSPI )  UF (5) = MNSPI
      ENDIF
*
*
      RETURN
      END
```

```
      SUBROUTINE FNCTNG
      DIMENSION X (6,200), T (200), UU (5,200), GM (3,5,200)
      DIMENSION XG (6), UG (5), TUU (200)
      COMMON / ONE / C1,C2,C3,G,C00,XK1,CLA,TOWMAX,XNMAX,ALFMAX
      COMMON / THREE / TAOALO, TSSLO, TUU
      COMMON / FOUR / X, T, KOUNT, UU, GM
      COMMON / SIX / LIMITR, LMTVC, LMTPI
      INTEGER  TAOALO, TSSLO, LIMITR, LMTVC, LMTPI
      INTEGER  KOUNT
*
*
      DO 10 I = 1, KOUNT
          CALL DERVUX  ( T(I), XG, UG )
*
          GSU1    = SIN ( UG(1) )
          GCU1    = COS ( UG(1) )
          GSU4    = SIN ( UG(4) )
          GCU4    = COS ( UG(4) )
          GSU5    = SIN ( UG(5) )
          GCU5    = COS ( UG(5) )
*
          GX4     = G / XG(4)
          GX46    = GX4 / COS ( XG(6) )
          GTW     = G * TOWMAX
*
          SIGMA   = ( 1. - C2 * XG(3) )**C3
*
          GCSA    = C1 * SIGMA * CLA
          GCSA4   = GCSA * XG(4) * XG(4)
          GA2     = CLA * UG(2)
*
```

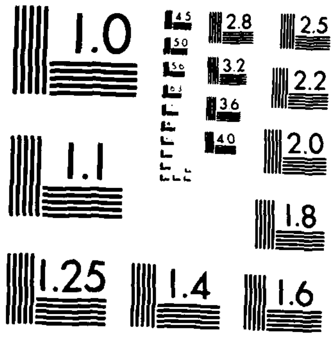
AD-A151 693 MINIMUM TIME TURNS USING VECTORED THRUST(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING G L SCHNEIDER DEC 84 AFIT/GAE/AA/84D-24 3/3

UNCLASSIFIED F/G 1/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
*
      GM (1,2,I) = -2. * G * GA2 * GCSA4 * XK1
      GM (1,3,I) = GTW * GCU4 * GCU5
      GM (1,4,I) = -GTW * UG(3) * GSU4 * GCU5
      GM (1,5,I) = -GTW * UG(3) * GCU4 * GSU5
*
      GM (2,1,I) = GX46 * ( TOWMAX * UG(3) * ( GSU4 * GCU1 -
     1                   GCU4 * GSU5 * GSU1 ) +
     2                   GCSA4 * UG(2) * GCU1 )
      GM (2,2,I) = GX46 * GCSA4 * GSU1
      GM (2,3,I) = GX46 * TOWMAX * ( GCU4 * GSU5 * GCU1 +
     1                               GSU4 * GSU1 )


      GM (2,4,I) = GX46 * TOWMAX * UG(3) * ( GSU1 * GCU4 -
     1                                      GSU4 * GSU5 * GCU1 )
      GM (2,5,I) = GX46 * TOWMAX * UG(3) * GCU4 * GCU5 * GCU1
*
      GM (3,1,I) = -GX4 * ( TOWMAX * UG(3) * ( GSU4 * GSU1 +
     1                   GCU4 * GSU5 * GCU1 ) +
     2                   GCSA4 * UG(2) * GSU1 )
      GM (3,2,I) = GX4 * GCSA4 * GCU1
      GM (3,3,I) = GX4 * TOWMAX * ( GSU4 * GCU1 -
     1                             GCU4 * GSU5 * GSU1 )
      GM (3,4,I) = GX4 * TOWMAX * UG(3) * ( GCU4 * GCU1 +
     1                                     GSU4 * GSU5 * GSU1 )
      GM (3,5,I) = -GX4 * TOWMAX * UG(3) * GCU4 * GCU5 * GSU1
*
 10   CONTINUE
*
*
      RETURN
      END
```

```
      SUBROUTINE FNCTNX  ( XX, U, DX )
      DIMENSION  XX (6), DX (6), U (5), UF (5), TUU (200)
      COMMON / ONE/  C1,C2,C3,G,CD0,XK1,CLA,TOWMAX,XNMAX,ALFMAX
      COMMON / THREE / TAOALO, TSSLO, TUU
      COMMON / SIX / LIMITR, LMTVC, LMTPI
      INTEGER  TAOALO, TSSLO, LIMITR, LMTVC, LMTPI
*
      SIGMA = ( 1. - C2 * XX (3) )**C3
*
      CALL FLIMIT  ( XX, U, UF )
      DO 10 I = 1, 5
10       U (I) = UF (I)
*
      SU1    = SIN ( U(1) )
      CU1    = COS ( U(1) )
      SU4    = SIN ( U(4) )
      CU4    = COS ( U(4) )
      SU5    = SIN ( U(5) )
      CU5    = COS ( U(5) )
      SX5    = SIN ( XX(5) )
      CX5    = COS ( XX(5) )
      SX6    = SIN ( XX(6) )
      CX6    = COS ( XX(6) )
*
      TWU3   = TOWMAX * U(3)
      GX4    = G / XX(4)
*
```

```
      CAU2   = CLA * U(2)
      CSX4   = C1 * SIGMA * XX(4) * XX(4)
      CSX42  = CSX4 * CAU2
*
*
      DX (1) = XX(4) * CX6 * CX5
      DX (2) = XX(4) * CX6 * SX5
      DX (3) = XX(4) * SX6
      DX (4) = G * ( TWU3 * CU4 * CU5 - SX6
     1             - CSX4 * ( CD0 + XK1 * CAU2 * CAU2 ) )
      DX (5) = GX4 * ( TWU3 * ( CU4 * SU5 * CU1
     1              + SU4 * SU1 ) + CSX42 * SU1 ) / CX6
      DX (6) = GX4 * ( TWU3 * ( SU4 * CU1 - CU4
     1              * SU5 * SU1 ) - CX6 + CSX42 * CU1 )
*
*
      RETURN
      END
```

```
      SUBROUTINE FNCTNY  ( XY, U, YPT, YOT, FP, FO )
      DIMENSION XY (6), U (5), YPT (3), YOT (3), FP (3), FO (3)
      COMMON / ONE /  C1,C2,C3,G,CD0,XK1,CLA,TOWMAX,XNMAX,ALFMAX
      COMMON / SIX / LIMITR, LMTVC, LMTPI
      INTEGER  LIMITR, LMTVC, LMTPI
*
*
      TEMPS  = 1. - C2 * XY (3)
      SIGMA  = TEMPS**C3
      SIGMAM = TEMPS**(C3 - 1.)
*
      VSU1   = SIN ( U(1) )
      VCU1   = COS ( U(1) )
      VSU4   = SIN ( U(4) )
      VCU4   = COS ( U(4) )
      VSU5   = SIN ( U(5) )
      VCU5   = COS ( U(5) )
      VSX6   = SIN ( XY(6) )
      VCX6   = COS ( XY(6) )
      TEMP   = TOWMAX + U(3) + ( VCU4 * VSU5 * VCU1 + VSU4 * VSU1 )
      VXY4SQ = XY(4) * XY(4)
*
      VCU    = CLA * U(2)
      VC1    = C1 * SIGMA * VCU
      VCC    = C1 * C2 * C3
*
      VCD    = CD0 + ( XK1 * VCU * VCU )
      VCG    = G / VCX6
*
*
      F34    = VSX6
      F36    = XY(4) * VCX6
```

```
*
      F43    = G * VCC * SIGMAN * VCD * VXY4SQ
      F44    = -2. * G * C1 * SIGMA * XY(4) * VCD
      F46    = -G * VCX6
*
      F53    = -VCG * VCC * SIGMAM * XY(4) * VCU * VSU1
      F54    = VCG * ( VC1 * VSU1 - ( TEMP / VXY4SQ  ) )
      F56    = VCG * VSX6 * ( TEMP + ( VC1 * VSU1 * VXY4SQ ) ) /
     1                      ( XY(4) * VCX6 )
*
      F63    = -G * VCC * XY(4) * SIGMAM * VCU * VCU1
      F64    = VC1 * VCU1 + ( ( VCX6 - G * TOWMAX * U(3) *
     1          ( VSU4 * VCU1 - VCU4 * VSU5 * VSU1 ) ) / VXY4SQ )
      F66    = G * VSX6 / XY(4)
*
*
      IF  ( LMTVC .EQ. 1 )  THEN
*
         SIGMAP = TEMPS**(C3+1.)
         TEMPN  = XNMAX / ( C1 * CLA )
         GOX4   = G / XY (4)
         GOX4C6 = GOX4 / VCX6
*
         G42 = -2. * G * VC1 * XK1 * CLA * VXY4SQ
         G43 = G * TOWMAX * VCU4 * VCU5
         G44 = -G * TOWMAX * U(3) * VSU4 * VCU5
         G45 = -G * TOWMAX * U(3) * VCU4 * VSU5
*
         G52 = VCG * XY(4) * C1 * SIGMA * CLA * VSU1
         G53 = GOX4C6 * TOWMAX * ( VCU4 * VSU5 * VCU1
     1          + VSU4 * VSU1 )
         G54 = GOX4C6 * TOWMAX * U(3) * ( VSU1 * VCU4
```

```
     1          - VSU4 * VSU5 * VCU1 )
          G55 = GOX4C6 * TOWMAX * U(3) * VCU4 * VCU5 * VCU1
*
          G62 = G * C1 * SIGMA * XY(4) * CLA * VCU1
          G63 = GOX4 * TOWMAX * ( VSU4 * VCU1
     1          - VCU4 * VSU5 * VSU1 )
          G64 = GOX4 * TOWMAX * U(3) * ( VCU4 * VCU1
     1          + VSU4 * VSU5 * VSU1 )
          G65 = -GOX4 * TOWMAX * U(3) * VCU4 * VCU5 * VSU1
*
          TEMP1 = 1. / ( SIGMA * VXY4SQ )
          TEMP2 = 2. * TEMPN * GOX4
          TEMP3 = TEMP2 * TOWMAX * TEMP1
*
          SU2D = 2. * TEMP2 * C1 * XK1 * VCU * CLA
          SU3D = -TEMP3 * VCU4 * VCU5
          SU4D = TEMP3 * U(3) * VSU4 * VCU5
          SU5D = TEMP3 * U(3) * VCU4 * VSU5
*
          SX3D = TEMP2 * TEMP1 * C2 * C3 * ( VSX6 -
     1           TOWMAX * U(3) * VCU4 * VCU5 ) / TEMPS
          SX4D = ( TEMPN / VXY4SQ ) * ( G * ( 6. * ( ( TOWMAX
     1           * U(3) * VCU4 * VCU5 - VSX6 ) * TEMP1 )
     2           + 2. * C1 * VCD ) - C2 * C3 * VSX6 /
     3           SIGMAP )
          SX6D = TEMPN * VCX6 * ( ( C2 * C3 / ( XY(4) *
     1           SIGMAP ) ) + ( 2. * GOX4 * TEMP1 ) )
*
*
          S23 = SU2D * SX3D
          S33 = SU3D * SX3D
          S43 = SU4D * SX3D
          S53 = SU5D * SX3D
*
```

```
      S24 = SU20 * SX4D
      S34 = SU30 * SX4D
      S44 = SU4D * SX40
      S54 = SU50 * SX4D
*
      S26 = SU20 * SX6D
      S36 = SU30 * SX6D
      S4E = SU4D * SX6D
      S56 = SU5D * SX6D
*
*
      F43 = F43 - ( G42*S23 + G43*S33 + G44*S43 + G45*S53 )
      F44 = F44 - ( G42*S24 + G43*S34 + G44*S44 + G45*S54 )
      F46 = F46 - ( G42*S26 + G43*S36 + G44*S46 + G45*S56 )
*
      F53 = F53 - ( G52*S23 + G53*S33 + G54*S43 + G55*S53 )
      F54 = F54 - ( G52*S24 + G53*S34 + G54*S44 + G55*S54 )
      F56 = F56 - ( G52*S26 + G53*S36 + G54*S46 + G55*S56 )
*
      F63 = F63 - ( G62*S23 + G63*S33 + G64*S43 + G65*S53 )
      F64 = F64 - ( G62*S24 + G63*S34 + G64*S44 + G65*S54 )
      F66 = F66 - ( G62*S26 + G63*S36 + G64*S46 + G65*S56 )
*
    ELSEIF  ( LIMITR .EQ. 1 )  THEN
*
      TEMPN = XNMAX / ( C1 * CLA )
*
      CX3 = -C2 * C3 * TEMPN / ( VXY4SQ * SIGMA * TEMPS )
      CX4 = 2. * TEMPN / ( VXY4SQ * XY (4) * SIGMA )
*
```

```
      G42 = -2. * G * VC1 * CLA * VXY4SQ * XK1
      TEMPG = C1 * SIGMA * XY (4) * CLA
      G52 = TEMPG * VSU1 * VCG
      G62 = TEMPG * VCU1 * G
*
      F43 = F43 - CX3 * G42
      F44 = F44 - CX4 * G42
      F53 = F53 - CX3 * G52
      F54 = F54 - CX4 * G52
      F63 = F63 - CX3 * G62
      F64 = F64 - CX4 * G62
*
      ENDIF
*
*
      FP (1) = - (               F43 * YPT(2) + F63 * YPT(3) )
      FP (2) = - ( F34 * YPT(1) + F44 * YPT(2) + F64 * YPT(3) )
      FP (3) = - ( F36 * YPT(1) + F46 * YPT(2) + F66 * YPT(3) )
*
*
      FO (1) = - (               F43 * YOT(2) + F63 * YOT(3) + F53 )
      FO (2) = - ( F34 * YOT(1) + F44 * YOT(2) + F64 * YOT(3) + F54 )
      FO (3) = - ( F36 * YOT(1) + F46 * YOT(2) + F66 * YOT(3) + F56 )
*
*
      RETURN
      END
```

```
      SUBROUTINE FIGRND  ( IERROR )
      DIMENSION  GSS (200), GSF (200), GFF (200), YPHIO (3,200),
     1   YPSIO (3,200), GM (3,5,200), X (6,200), T (200),
     2   UU (5,200), YT1 (3,1), YT3 (3,1), GT (3,5),
     3   WINV (5,5), YG1 (1,5), YG2 (5,1), YG3 (1,5),
     4   YG4 (5,1), YG1W (1,5), YG4W (5,1), GTEMP (1,1),
     5   TEMPX (6), TEMPU (5)
      COMMON / FOUR / X, T, KOUNT, UU, GM
      COMMON / FIVE / YPHIO, YPSIO, GSS, GSF, GFF
      COMMON / SIX / LIMITR, LNTVC, LMTPI
      COMMON / SEVEN / WINV
      INTEGER  KOUNT, LIMITR, LMTVC, LMTPI
*
*
C   COMPUTE INTEGRANDS FOR ISISI, ISIFI, IFIFI
C       ( GSS, GSF, GFF )
*
C   ZERO MATRICES FIRST
*
      GTEMP (1,1) = 0.
      DO 100 I = 1, 200
          GSS (I) = 0.
          GSF (I) = 0.
 100      GFF (I) = 0.
*
      DO 200 I = 1, 3
          YT1 (I,1) = 0.
          YT3 (I,1) = 0.
          DO 210 J = 1, 5
                  GT (I,J) = 0.
                 YG1 (1,J) = 0.
                 YG2 (J,1) = 0.
                 YG3 (1,J) = 0.
 210             YG4 (J,1) = 0.
 200  CONTINUE
```

```
      IERROR = 0
*
*
      DO 300 I = 1, KOUNT
         DO 310 J = 1, 3
               YT1 (J,1) = YPSIO (J,I)
               YT3 (J,1) = YPHIO (J,I)
               DO 320 K = 1, 5
320               GT (J,K) = GM (J,K,I)
310      CONTINUE
*
*
         CALL DERVUX  ( T (I), TEMPX, TEMPU )
         IF  ( LIMITR .EQ. 1 )  THEN
               WINV (2,2) = 0.
         ELSE
               WINV (2,2) = 1.
         ENDIF
*
         IF  ( LMTPI .EQ. 1 )  THEN
               WINV (3,3) = 0.
         ELSE
               WINV (3,3) = 1.
         ENDIF
*
*
         CALL VMULFM  ( YT1, GT, 3, 1, 5, 3, 3, YG1, 1, IERYG1 )
         IF  ( IERYG1 .NE. 0 )  THEN
               PRINT 331, IERYG1
               PRINT 330, I
331            FORMAT (///1X,"IERYG1 = ",I4)
330            FORMAT (///1X,"INDEX FOR DO LOOP 300 = ",I4)
               IERROR = IERYG1
               RETURN
         ENDIF
```

```
*
      CALL VMULFM  ( GT, YT1, 3, 5, 1, 3, 3, YG2, 5, IERYG2 )
      IF  ( IERYG2 .NE. 0 )  THEN
            PRINT 332, IERYG2
            PRINT 330, I
 332        FORMAT (///1X,"IERYG2 = ",I4)
            IERROR = IERYG2
            RETURN
      ENDIF
*
      CALL VMULFM  ( YT3, GT, 3, 1, 5, 3, 3, YG3, 1, IERYG3 )
      IF  ( IERYG3 .NE. 0 )  THEN
            PRINT 333, IERYG3
            PRINT 330, I
 333        FORMAT (///1X,"IERYG3 = ",I4)
            IERROR = IERYG3
            RETURN
      ENDIF
*
      CALL VMULFM  ( GT, YT3, 3, 5, 1, 3, 3, YG4, 5, IERYG4 )
      IF  ( IERYG4 .NE. 0 )  THEN
            PRINT 334, IERYG4
            PRINT 330, I
 334        FORMAT (///1X,"IERYG4 = ",I4)
            IERROR = IERYG4
            RETURN
      ENDIF
*
      CALL VMULFF  ( YG1, WINV, 1, 5, 5, 1  5, YG1W, 1, IERG1W )
```

```
      IF  ( IERG1W .NE. 0 )  THEN
            PRINT 335, IERG1W
            PRINT 330, I
335         FORMAT (///1X,"IERG1W = ",I4)
            IERROR = IERG1W
            RETURN
      ENDIF
*
      CALL VMULFF  ( WINV, YG4, 5, 5, 1, 5, 5, YG4W, 5, IERG4W )
      IF  ( IERG4W .NE. 0 )  THEN
            PRINT 336, IERG4W
            PRINT 330, I
336         FORMAT (///1X,"IERG4W = ",I4)
            IERROR = IERG4W
            RETURN
      ENDIF
*
*
C   NOW THE INTEGRANDS
*
*
      CALL VMULFF ( YG1W, YG2, 1, 5, 1, 1, 5, GTEMP, 1, IERGSS )
      IF  ( IERGSS .NE. 0 )  THEN
            PRINT 337, IERGSS
            PRINT 330, I
337         FORMAT (///1X,"IERGSS = ",I4)
            IERROR = IERGSS
            RETURN
      ENDIF
      GSS (I) = GTEMP (1,1)
```

```
*
      CALL VMULFF ( YG1W, YG4, 1, 5, 1, 1, 5, GTEMP, 1, IERGSF )
      IF  ( IERGSF .NE. 0 )  THEN
              PRINT 338, IERGSF
              PRINT 330, I
338           FORMAT (///1X,"IERGSF = ",I4)
              IERROR = IERGSF
              RETURN
      ENDIF
      GSF (I) = GTEMP (1,1)
*
      CALL VMULFF ( YG3, YG4W, 1, 5, 1, 1, 5, GTEMP, 1, IERGFF )
      IF  ( IERGFF .NE. 0 )  THEN
              PRINT 339, IERGFF
              PRINT 330, I
339           FORMAT (///1X,"IERGFF = ",I4)
              IERROR = IERGFF
              RETURN
      ENDIF
      GFF (I) = GTEMP (1,1)
*
*
300   CONTINUE
*
*
      RETURN
      END
```

```
      SUBROUTINE POINTE  ( VALUE, LENGTH, POINT, FRACT, TABLE )
*
*
C   SUBROUTINE POINTE COMPARES THE NUMBER IN "VALUE" WITH THE LIST OF
C   NUMBERS IN ASCENDING ORDER IN "TABLE", RETURNING "POINT" AND
C   "FRACT". "POINT" IS THE POSITION OF THE LARGEST NUMBER LESS THAN
C   "VALUE", AND "FRACT" IS A FRACTIONAL NUMBER REPRESENTING THE


C   DISTANCE "VALUE" IS BETWEEN THE TWO POINTS: POINT AND POINT+1.
*
C   IF THE ARGUMENT SPECIFIED IS BEYOND EITHER OF THE ARGUMENT
C   TABLE LIMITS THE FRACTIONAL VALUE RETURNED WILL EXTEND BEYOND
C   THE RANGE OF +1, -1.
C   THIS ALLOWS THE FUNCTION VALUE TO BE EXTRAPOLATED BEYOND ITS
C   TABULATED VALUES, USING THE LAST TWO BREAKPOINTS.
*
      INTEGER  LENGTH, POINT
      REAL     VALUE,  FRACT, TABLE (LENGTH)
*
      IF  ( POINT .LT. 1 )  POINT = 1
      IF  ( POINT .GE. LENGTH )  POINT = LENGTH - 1
      IF  ( VALUE - TABLE (POINT) )  10,20,40
*
C   CHECK TO SEE IF VALUE IS OUTSIDE OF TABLE LIMITS
```

```
*
 10   IF  ( POINT .EQ. 1 )  GO TO 50
      POINT = POINT - 1
      IF  ( VALUE - TABLE (POINT) )  10,20,50
*
 20   FRACT = 0.0
      RETURN
*
C   CHECK TO SEE IF VALUE IS OUTSIDE OF TABLE LIMITS
*
 30   POINT = POINT + 1
 40   IF  ( POINT + 1 .EQ. LENGTH )  GO TO 50
      IF  ( VALUE - TABLE (POINT + 1) )  50,30,30
*
 50   FRACT = ( VALUE - TABLE (POINT) ) /
     1        ( TABLE (POINT + 1) - TABLE (POINT) )
*
      RETURN
      END
```

Bibliography

1. DeMeis, Richard. "Fighter Maneuverability," Aerospace America, 22 (5): 80-83 (May 1984).

2. -----. "New Nozzle Design Aimed at F-15, F-16 Aircraft," Aviation Week & Space Technology, 117: 67-72 (September 13, 1982).

3. -----. "McDonnell Douglas to Develop STOL F-15," Aviation Week & Space Technology, 121 (15): 21 (October 8, 1984).

4. Humphreys, Robert P., George R. Hennig, William A. Bolding, and Larry A. Helgeson. "Optimal 3-Dimensional Minimum Time Turns for an Aircraft," The Journal of the Astronautical Sciences, XX (2): 88-112 (September-October 1972).

5. Johnson, Capt Thomas L. Minimum Time Turns with Thrust Reversal. MS thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1979 (AD-A079 851).

6. Finnerty, Capt Christopher S. Minimum Time Turns Constrained to the Vertical Plane. MS thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1980 (AD-A111 096).

7. Brinson, Michael R. Minimum Time Turns with Direct Sideforce. MS thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1983 (AD-A136 958).

8. NASA. U.S. Standard Atmosphere. Washington, D.C., December 1962.

9. Bryson, A. E. and W. F. Denham. "A Steepest-Ascent Method for Solving Optimum Programming Problems," Transactions of the ASME Journal of Applied Mechanics, 247-257 (June 1962).

10. Well, K. H. and E. Berger. "Minimum-Time 180° Turns of Aircraft," Journal of Optimization Theory and Application, 38 (1): 83-96 (September 1982).

11. Miele, Angelo. "Theory of Flight Paths," Flight Mechanics, 1. Massachusetts: Addison-Wesley Publishing Company, Inc. 1962.

12. Gabriele, G. A. and K. M. Ragsdell. "The Generalized Reduced Gradient Method: A Reliable Tool for Optimal Design," Transactions of the ASME Journal of Engineering for Industry, 394-400 (May 1977).

13. Denham, Walter F. and Arthur E. Bryson, Jr. "Optimal Programming Problems with Inequality Constraints II: Solution by Steepest-Ascent," AIAA Journal 2 (1): 25-34 (January 1964).

14. Bryson, A. E. Jr., W. F. Denham, and S. E. Dreyfus. "Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for Extremal Solutions," AIAA Journal, 1 (11): 2544-2550 (November 1963).

15. Denn, Morton M. Optimization by Variational Methods. New York: McGraw-Hill Book Company, 1969.

VITA

Captain Garret L. Schneider was born on 24 June 1953 in Green Bay, Wisconsin. He graduated from high school in Okinawa, Japan, in 1970. After enlisting in the U.S. Air Force, he attended the University of Washington on the Airman Education and Commissioning Program. Upon graduating and receiving the degree of Bachelor of Science in Aeronautics and Astronautics in June 1980, he was commissioned in the USAF. He was then assigned to the Air Force Flight Test Center, Edwards AFB, California, where he served as a simulation and flight test engineer until entering the School of Engineering, Air Force Institute of Technology, in June 1983.

Permanent address: 490 Polaris Road
Green Bay, Wisconsin 54302

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GAE/AA/84D-24	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) See Box 19	

12. PERSONAL AUTHOR(S)
Garret L. Schneider, B.S., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ______ TO ______	1984 December	211

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
12	01	
01	03	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Optimization; Thrust Vectoring; Vectored Thrust; Minimum Time Turns; Steepest-Ascent Method; Optimal Controls.

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

Title: MINIMUM TIME TURNS USING VECTORED THRUST

Thesis Advisor: G. W. Watt, Lt Col, USAF

Approved for public release: IAW AFR 190-17.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
G. W. Watt, PhD, Lt Col, USAF	513-255-3517	AFIT/ENY

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

The objective of this investigation is to determine the optimal controls and trajectories which minimize the time to turn for a high performance aircraft with thrust vectoring capability. All determinations are subject to practical physical constraints. The determined controls and trajectories are then compared against other methods of turning in minimum time to conclude the effects and advantages of thrust vectoring.

The results indicate that the use of vectored thrust can substantially reduce turning times and increase in-flight maneuverability. The greater the velocity at which the turn is initiated, the more the range of thrust vectoring capability is used and the greater the reduction in turning time.

END

FILMED

5-85

DTIC