CR-171989

CSDL-T-950

AN ADAPTIVE NUMERIC PREDICTOR-CORRECTOR GUIDANCE ALGORITHM FOR ATMOSPHERIC ENTRY VEHICLES

by

Kenneth Milton Spratlin

May 1987

Master of Science Thesis
Massachusetts Institute of Technology

(NASA-CR-171989) AN ADAPTIVE NUMERIC PREDICTOR-CORRECTOR GUIDANCE ALGORITHM FOR ATMOSPHERIC ENTRY VEHICLES M.S. Thesis - MIT, Cambridge (Draper (Charles Stark) Lab.) 206 p Avail: NTIS HC A10/MF A01 CSCL 22A G3/17 N87-26934 Unclas 0090665

The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, Massachusetts 02139

AN ADAPTIVE NUMERIC PREDICTOR-CORRECTOR

GUIDANCE ALGORITHM FOR ATMOSPHERIC

ENTRY VEHICLES

by

Kenneth Milton Spratlin

B.A.E., Georgia Institute of Technology
(1985)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN
AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1987

Signature of Author Kenneth M. Spratlin
Department of Aeronautics and Astronautics
May 8, 1987

Approved by Richard H. Battin 5/8/87
Professor Richard H. Battin
Department of Aeronautics and Astronautics
Thesis Supervisor

Approved by Timothy J. Brand 5/8/87
Timothy J. Brand
Technical Supervisor, CSDL

Accepted by ____________________
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

AN ADAPTIVE NUMERIC PREDICTOR-CORRECTOR

GUIDANCE ALGORITHM FOR ATMOSPHERIC

ENTRY VEHICLES

by

Kenneth Milton Spratlin

Submitted to the Department of Aeronautics and Astronautics
on May 8, 1987, in partial fulfillment of the
requirements for the Degree of Master of Science in
Aeronautics and Astronautics

ABSTRACT

An adaptive numeric predictor-corrector guidance algorithm is developed for atmospheric entry vehicles which utilize lift to achieve maximum footprint capability. Applicability of the guidance design to vehicles with a wide range of performance capabilities is desired so as to reduce the need for algorithm redesign with each new vehicle. Adaptability is desired to minimize mission-specific analysis and planning. The guidance algorithm motivation and design are presented.

Performance is assessed for application of the algorithm to the NASA Entry Research Vehicle (ERV). The dispersions the guidance must be designed to handle are presented. The achievable operational footprint for expected worst-case dispersions is presented. The algorithm performs excellently for the expected dispersions and captures most of the achievable footprint.

Thesis Supervisor: Dr. Richard H. Battin

Title: Adjunct Professor of Aeronautics and Astronautics

Technical Supervisor: Timothy J. Brand

Title: Division Leader, The Charles Stark Draper Laboratory, Inc.

ACKNOWLEDGEMENT

This report was prepared at The Charles Stark Draper Laboratory, Inc. in support of NASA Langley Research Center (LaRC) Task Order No. 87-43 under Contract NAS9-17560 with the NASA Lyndon B. Johnson Space Center (JSC).

Publication of this report does not constitute approval by the Draper Laboratory or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

I wish to take this opportunity to thank some of the many people who have helped me during the preparation of this thesis and my stay at M.I.T.. I would like to express my sincere gratitude to The Charles Stark Draper Laboratory for making my graduate study possible through the Draper Fellowship. I wish to thank all of the members of the Guidance and Navigation Analysis Division for their technical assistance and willingness to share their expertise. In particular I would like to thank T. Brand, J. Higgins, and A. Engel for their patience, assistance, and support during the preparation of this thesis and work on the Aeroassist Flight Experiment. I wish to thank my thesis advisor, Dr. R.H. Battin, for his assistance and in particular for his abilities as an educator. A special thanks for his encouragement and support goes to B. Kriegsman who recently passed away. He is greatly missed. I found the opportunity to interact with the people at CSDL to be the most rewarding aspect of my studies at M.I.T..

I also wish to thank H. Stone and R. Powell of NASA/LaRC for the opportunity to work on the ERV entry guidance problem and their assistance in answering my questions.

Most importantly, I wish to thank my parents and sisters for their unending love and support over the years.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

PRECEDING PAGE BLANK NOT FILMED

PAGE 12 INTENTIONALLY BLANK

SYMBOLS

a	= acceleration magnitude
$\vec{a}$	= acceleration vector
$\vec{a}_I$	= inertial acceleration measured by the inertial measurement unit
$\vec{A}_I$	= total inertial acceleration vector
AOTV	= Aerobraking Orbital Transfer Vehicle
BTU	= British Thermal Unit
$\bar{c}$	= mean aerodynamic chord
$\cos(\Delta\phi)$	= cosine of incremental lift for heat rate control
C'	= proportionality factor for the linear viscosity-temperature relationship
C_D	= aerodynamic drag coefficient
C_L	= aerodynamic lift coefficient
C_s	= speed of sound
CPU	= central processing unit
CR	= crossrange
CSDL	= The Charles Stark Draper Laboratory, Inc.
det	= determinant of sensitivity matrix
DR	= downrange
DOF	= degree of freedom
ERV	= Entry Research Vehicle
f_{earth}	= flattening of oblate Earth
$\vec{F}_I$	= inertial force vector
g	= gravitational acceleration magnitude
$\vec{g}_I$	= gravitational acceleration vector

PRECEDING PAGE BLANK NOT FILMED

PAGE 14 INTENTIONALLY BLANK

GPS	=	Global Positioning System
h	=	altitude
$\dot{h}$	=	derivative of altitude with time
$\ddot{h}$	=	second-derivative of altitude with time
h_s	=	scale height for exponential atmosphere model
$\vec{i}$	=	unit vector
J_2	=	second zonal harmonic coefficient
JSC	=	Lyndon B. Johnson Space Center
k	=	term in geodetic to geocentric latitude conversion
K	=	term in heat rate control equation
$\vec{K}$	=	velocity vector term in the integration algorithm
$\vec{K}'$	=	acceleration vector term in the integration algorithm
$K_{\Delta t}$	=	gain on acceleration magnitude in variable time step equation
$K_{L/D}$	=	multiplicative scale factor on the nominal L/D
$K_{\dot{Q}}$	=	gain on heat rate error in heat rate control equation
$K_{\ddot{Q}}$	=	gain on rate of change of heat rate in heat rate control equation
K_ρ	=	multiplicative scale factor on the standard density
K_1	=	gain in first-order filter for density smoothing
K_2	=	gain in first-order filter for L/D smoothing
L/D	=	lift-to-drag ratio
$LaRC$	=	Langley Research Center
m	=	mass
M	=	Mach Number
$\hat{M}_I^{EF}$	=	inertial-to-Earth-fixed transformation matrix
M_0	=	mean molecular weight of air at sea level
$n.m.$	=	nautical mile
$NASA$	=	National Aeronautics and Space Administration

POST	=	Program to Optimize Simulated Trajectories
$\bar{q}$	=	dynamic pressure
Q	=	heat load
$\dot{Q}$	=	heat rate
$\ddot{Q}$	=	time rate of change of heat rate
R	=	position vector magnitude
$R_{equator}$	=	radius of oblate Earth at equator
$\vec{R}_I$	=	inertial position vector
R_{pole}	=	radius of oblate Earth at pole
Re	=	Reynolds Number
S	=	Sutherland's constant in viscosity equation
S	=	aerodynamic reference area
SEADS	=	Shuttle Entry Air Data System
t_{GMT}	=	Greenwich mean time
T'	=	reference temperature
T_M	=	molecular scale temperature
T_{static}	=	freestream static temperature
T_{wall}	=	wall temperature
TAEM	=	Terminal Area Energy Management
$\bar{V}$	=	Viscous Interaction Parameter
V_I	=	magnitude of inertial velocity
$\vec{V}_I$	=	inertial velocity vector
V_R	=	magnitude of Earth-relative velocity
$\vec{V}_R$	=	Earth-relative velocity vector
z	=	geocentric colatitude of the postion vector
α	=	angle of attack

β = angle of sideslip

δ = small incremental change

Δ = incremental change

γ = ratio of specific heats for air

λ = longitude

μ = coefficient of viscosity for air

μ = Earth gravitational constant

$\vec{\omega}_{earth}$ = Earth's rotation vector

ω_n = natural frequency of heat rate control response

ϕ = bank angle

$\mathscr{R}$ = universal gas constant

ρ = atmospheric density

σ = standard deviation

τ = time constant of filter

ζ = damping ratio of heat rate control response

SUBSCRIPTS

aero = aerodynamic

c = geocentric

cmd = command

d = desired

des = desired

drag = drag term

e = error

EI = entry interface

f = final

g = geodetic

imu = inertial measurement unit

$inplane$ = projection of target unit vector into plane formed by the position vector and the relative velocity vector

lat = direction perpendicular to the plane formed by the position vector and the relative velocity vector

$lift$ = lift term

lim = limiting value or boundary

L/D = lift-to-drag ratio

max = maximum

min = minimum

nom = nominal

$perpen$ = direction perpendicular to the plane formed by the position vector and the relative velocity vector

$pole$ = direction of the north pole

R = direction of the position vector

sl = sea level

std = standard value

t = target aim point

ρ = atmospheric density

SUPERSCRIPTS

EF	=	coordinatized in Earth-fixed coordinates
imu	=	value measured by inertial measurement unit on current cycle
imu past	=	value measured by inertial measurement unit on past cycle
$\wedge$	=	estimated or measured
$'$	=	value from previous guidance cycle

1.0 INTRODUCTION

Routine access to space and the maintenance of a Space Station will increasingly require greater flexibility in mission planning and the requirement for lower system maintenance costs. The launch and recovery phases of space flight have historically been the most demanding phases of space flight and therefore require the most development effort and investment. Mission flexibility requires more frequent launch and deorbit opportunities. For the case of re-entry vehicles, deorbit opportunities are defined by the ranging capability of the vehicle. A high L/D vehicle increases the available deorbit opportunities increasing mission flexibility. High L/D vehicles also are of interest for over-flight missions for the purpose of reconnaissance.

Entry guidance algorithms developed to date have been highly vehicle-specific and required great development and maintenance efforts over the life of the vehicle. These algorithms were not applicable to other vehicles without extensive modification.

This study seeks to design an adaptive entry guidance algorithm that maximizes the usable footprint by making full use of the available vehicle capability. This algorithm should also be easy to maintain throughout the vehicle definition phase and operational life. Minimizing the number of mission-dependent input parameters (I-loads) is desirable. The algorithm should also be easily transported to other vehicles to minimize development cost. Transportability is accomplished by minimizing vehicle-specific features of the algorithm. Explicit heat rate control should be provided to allow full use of the entry corridor up to the heat rate limits.

This study seeks to design such an algorithm. A candidate entry guidance algorithm is defined for the NASA Entry Research Vehicle (ERV), but is easily adapted to other vehicles with minimal modification. The proposed algorithm attains almost complete coverage of the achievable footprint, while employing a simple one-phase entry algorithm with explicit heat rate control. Vehicle-specific features and I-loads are minimized, reducing algorithm development and maintenance costs.

The ERV [1] is a proposed high-performance entry vehicle designed as a test bed for future technology development in the areas of:

1. Maneuvering entry/synergetic plane change
2. Atmospheric uncertainties
3. Advanced thermal protection systems
4. Aerodynamic/aeroheating prediction
5. Adaptive guidance and navigation
6. Load-bearing thermostructures

The ERV is designed for deployment from the Space Shuttle, after which the ERV enters the atmosphere for demonstration of the synergetic plane change, over-flight, and entry missions. Figure 1 on page 78 shows a three-view drawing of the ERV and the surface areas of the aerodynamic control surfaces. Also seen is the size of the ERV in relation to the diameter of the Shuttle payload bay in which the ERV must fit.

2.0 MOTIVATION

2.1 INTRODUCTION

The goal of any entry guidance algorithm is to successfully guide the vehicle to the desired final state for the largest range of dispersions possible without violating any vehicle constraints while also maximizing the achievable footprint. It is also desirable to minimize the mission and vehicle-specific aspects of the guidance algorithm so as to minimize pre-mission analysis and planning. Transportability of the algorithm from one vehicle to another significantly reduces guidance algorithm development effort and cost.

To maximize the footprint attainable, the guidance algorithm must follow the optimal path to any particular point in the footprint. The algorithms developed to date for such vehicles as the Apollo capsule [2] and the Space Shuttle [3] have attempted to do this by fitting the optimal trajectory with phases that follow important parameters (reference profiles) over some range of conditions. These guidance algorithms were required to be computationally efficient because of the limited on-board computer resources available. Analytic expressions for the reference profiles allowed for low execution time and tailoring of the trajectory for vehicle-specific constraints. For example, trajectories for these vehicles had to be shaped to reduce and control the maximum heat rate experienced below that allowed for the available thermal protection system materials.

The Space Shuttle entry guidance system employs three major modes with seven phases:

1. Entry
    a. Pre-entry
    b. Temperature control
    c. Equilibrium glide
    d. Constant drag
    e. Transition
2. Terminal Area Energy Management
3. Approach and Landing

Except for the pre-entry phase which is open-loop, each phase is described by an analytic expression relating the desired drag and altitude rate (the measured feedback terms used) to the desired profile. Because the algorithms are tailored for a particular vehicle and the reference profiles do not follow the optimal profile to all points in the footprint, guidance algorithms developed to date can not be easily adapted to other vehicles or provide full coverage of the theoretically achievable footprint.

The next generation of entry vehicles will not be so constrained due to advances in thermal protection system materials and computer technology. For example, flight computers are now capable of supercomputer speeds on the order of 40 million instructions per second utilizing parallel processing architecture [4]. A different approach to guidance that attempts to follow an optimal profile to maximize footprint capability is therefore possible.

The proposed approach is a predictor-corrector algorithm that numerically predicts the final state for a particular control variable history and then corrects the control variable history to satisfy the specified final state constraints. This approach, proposed previously for various guidance problems, has most often been impractical because of the long trajectories that must be predicted and the slow computer speeds.

Such an approach has been employed for the Space Shuttle Powered Explicit Guidance (PEG) [5] used for second stage ascent and orbit insertion burns where the trajectory is short enough to be predicted with the available computer resources. The Shuttle algorithm numerically predicts the gravitational effects during the powered flight phase with a 10 step integration of the 500 second trajectory.

A predictor-corrector has also been proposed for Aerobraking Orbital Transfer Vehicles (AOTV) [6] which would utilize more advanced computers. This algorithm numerically integrates the equations of motion along a skimming trajectory through the upper atmosphere that is approximately 500 seconds long and requires about 100 integration steps.

The trajectories flown by the ERV or any high L/D entry vehicle are typically from 30 to 100 minutes long from entry interface (400K feet) to landing, so the computational demand for such an algorithm is very great early in the entry when the time to landing is long. However, because the entry is long and the vehicle has excess ranging capability for all but a small region along the edge of the footprint, the accuracy of the early predictions need not be as high as for the later predictions. Hence, large time steps can be used early in the predictor algorithm. Later, when the vehicle nears the landing site, the time remaining is short, and hence, the prediction is short. This allows the predictor-corrector to be executed more often near landing just like the current analytic algorithms. Throughout the entry, vehicles using an analytic guidance algorithm with reference profiles must closely follow the reference profile if the assumed reference profile is to guide the vehicle to the correct final state. A predictor-corrector effectively recomputes a new reference profile each time it is executed, so the guidance execution rate can be much lower than that for analytic algorithms.

2.2 DISPERSIONS

Before the guidance algorithm can be designed, the possible dispersions that may affect the trajectory must be considered. The Shuttle entry guidance system is required to reach the Terminal Area Energy Management (TAEM) interface with less than a 2.5 nautical mile position error from the target aim point. The dispersions of significance to an entry vehicle trajectory include:

1. Vehicle characteristics
   a. Mass
   b. Aerodynamics
   c. Maneuver rates
2. Environment characteristics
   a. Atmospheric density
   b. Atmospheric winds
   c. Atmospheric properties influencing aerodynamic flow regimes (temperature, mean free path, etc.)
3. Initial entry state vector
   a. Velocity
   b. Flight path angle
   c. Heading
4. Propagation errors in navigation state vector

Of these potential dispersion sources, only the vehicle mass, aerodynamics, and the atmospheric density and winds will be significant. By the early 1990's, almost perfect navigation can be expected through use of the Global Positioning System (GPS). If the deorbit burn guidance and control systems are assumed to correctly guide to the navigated state and there are no navigation errors, then the dispersions in the initial entry state vector are negligible.

The vehicle mass should be known accurately, so for this study, a 3σ error of $\pm 5\%$ is assumed. Experience from the Space Shuttle program shows that the vehicle aerodynamics should be known to within $\pm 5\%$ for the force coefficients on the first flight. Only the stability derivatives and control effectiveness were missed significantly [7]. Even though the force coefficients may be known to excellent accuracy, reduced control effectiveness can reduce the possible trim angle of attack range reducing the maximum L/D achievable. Therefore, for this study, a $\pm 10\%$ dispersion in the lift and drag coefficients is considered. It should be noted that the first few flights of a new vehicle are usually targeted to the middle of the footprint to maximize margin and allow for accurate determination of the vehicle characteristics before the full ranging capability of the vehicle is used. After the first few flights, the aerodynamic characteristics should be known to within a few percent, so only about a $\pm 3\%$ dispersion must be considered.

The atmospheric dispersions were obtained from two sources. Reference [8] specifies the atmospheric dispersions to which aerospace vehicles must be designed. The average of the steady state winds at four geographic locations is shown in Figure 2 on page 79. This model was incorporated into the simulator environment with a magnitude scale factor to simulate less than worst-case winds. The wind direction was selected for each run made with winds and held constant throughout the trajectory. Reference [8] specifies Reference [9] as the source for atmospheric density dispersions. However, the recent Shuttle flights have provided estimated density data of a quality never before available. Atmospheric density profiles derived from Shuttle accelerometer measurements of the normal force acceleration and the estimated normal force coefficient and relative velocity vector are presented in Reference [10]. Figure 3 on page 80, taken from that report, shows the envelope of the derived density profiles for the first 12 Shuttle flights. Of particular interest is the range of dispersions seen: -47% to +12%. Figures 4 on page 81 and 5 on page 82 show the density profiles for the STS-1 and STS-9 Shuttle flights. High frequency density shear components and constant density biases from the standard atmosphere are seen. For this study, constant density biases of $\pm 30\%$ and the Shuttle derived density profiles from Reference [10] were used.

2.3 REFERENCE TRAJECTORIES

The size of the footprint for a particular vehicle is determined by the range in vehicle L/D and the constraints placed on the trajectory such as heat rate limits. The edges of the footprint correspond to the use of maximum or minimum L/D. Maximum downrange or crossrange, for example, requires maximum L/D, while minimum downrange requires minimum L/D.

The determination of the optimal angle of attack and bank angle control histories for maximum crossrange and downrange has been the topic of many papers [11] [12] [13] . Wagner [12] used several optimization techniques to evaluate the maximum crossrange achievable for a multiphase bank angle history flown at maximum L/D. The multiphase bank profiles considered are shown in Figure 6 on page 83. It is seen that as the number of phases increases, the multiphase profile approaches the optimal continuous profile also shown in this figure. It was determined that a three-phase bank angle profile as illustrated in Figure 6 achieved almost the same crossrange as a continuous bank profile. This is shown in Figure 7 on page 84 reproduced here from that paper. Further, as the number of phases increases, the optimum bank angle profile approaches a continuous profile that is almost linear with velocity as shown in Figure 8 on page 85. It was also shown that flying at the maximum L/D maximizes the crossrange attained.

This result is confirmed in Reference [13] which utilized a nonlinear programming technique to optimize the Space Shuttle trajectory for the maximum downrange and maximum crossrange cases. The maximum downrange trajectory requires flying at zero bank angle and at the angle of attack corresponding to maximum L/D as shown in Figure 9 on page 86. The control histories for the maximum crossrange case are shown in Figures 10 on page 87 and 11 on page 88. Again, the optimal control history is the angle of attack corresponding to maximum L/D and an almost linear bank angle profile with velocity.

Optimized trajectories for the ERV were reported in Reference [14]. These trajectories were determined using the Program to Optimize Simulated Trajectories (POST) [15] and imposed the following constraints on the trajectories:

1. Maximum heat rate of 125 BTU/sq ft/sec
2. Maximum heat load of 150K BTU/sq ft

The achievable footprint with these constraints, reported in Reference [14], is shown here in Figure 12 on page 89 . Subsequently, the heat load limit was increased to 175K BTU/sq ft resulting in the larger footprint shown in Figure 12. As will be seen, these footprints omit a large area in the minimum downrange region that is achievable within the heating constraints. Also shown is the footprint of the Space Shuttle which has a maximum hypersonic L/D of 1.2 as compared with 1.8 for the ERV.

Figure 13 on page 90 shows the altitude history for the maximum downrange, maximum crossrange, and minimum downrange cases. Figures 14 on page 91 and 15 on page 92 show the bank angle and angle of attack histories for these trajectories. Figures 16 on page 93 and 17 on page 94 show the heat rate and heat load histories for these cases.

Figure 15 shows that the constant angle of attack corresponding to maximum L/D is flown for the edge of the footprint except for the minimum downrange case. For the minimum downrange case, the angle of attack corresponding to the minimum L/D on the back side of the L/D curve (high drag coefficient) is flown early, followed by a ramp in angle of attack starting at 1500 seconds after entry interface. This ramp corresponds to the vehicle actually turning around and flying slightly back uprange, so maximum L/D is desired later to maximize the distance flown uprange.. The angle of attack for the maximum downrange case is slightly greater than that for maximum L/D because this trajectory exceeds the heat load limit if flown at maximum L/D. The maximum downrange region of the footprint is therefore limited by the heat load limit set for the ERV. If the limit were relaxed, flight at maximum L/D would allow a longer downrange trajectory.

Figure 14 shows that the bank angle profile for maximum crossrange is approximately linear with time which is almost linear with velocity, which suggests that a linear bank angle profile with velocity is sufficient. The maximum downrange case has a constant bank angle of zero which is again linear with velocity. The minimum downrange case does not have a linear bank profile. As was mentioned previously, for this case, the vehicle turns around and flys back uprange.

The results of these studies suggest that use of a constant angle of attack profile and a linear bank with velocity profile will capture a large portion of the achievable footprint. As will be seen in the results, these profiles suffice to capture most of the footprint reported in Reference [14] and additionally reach a large area in the minimum downrange region out-side the reported footprint. Only a small area of the reported footprint in the minimum downrange region is unachievable.

Also of interest are the peaks in heat rate seen in Figure 16. Because the peaks in heat rate are very short, explicit control of the heat rate should be possible in the maximum heat rate regions without significantly impacting the guidance.

2.4 GUIDANCE APPROACH

The guidance design will attempt to maximize the size of the footprint while flying a constant angle of attack profile and a linear bank angle with velocity profile. The predictor algorithm integrates the equations of motion forward in time using the assumed control profile and the necessary environment and vehicle models. The corrector then determines (using multiple predicted trajectories with various control histories) the sensitivities of the final state constraints to the control variables. The sensitivities are then used to compute the required control variable values to reach the desired final state conditions. Heat rate control is provided locally during the regions of maximum heating without significantly affecting the assumed control histories. Also, in-flight measurements are utilized to increase the accuracy of the predicted trajectories by compensating for off-nominal conditions.

Such a simple profile for the maximum downrange and crossrange cases simplifies the modeling of the control histories in the predictor. The only remaining question is how much of the footprint this profile will capture. As will be seen in Subsection "4.2 Open-Loop

Footprint" on page 57, such a profile achieves almost complete coverage of the achievable footprint.

Also of concern is the linearity and convergence properties of the final state constraints with the control variables. As will be seen, over almost all of the footprint except near the edges, the constraints are highly linear and convergent with the control variables. Operationally, only about 75% of the achievable footprint is used to ensure guidance margin. Thus, the question of nonconvergence near the edges is avoided.

3.0 GUIDANCE DESIGN

3.1 INTRODUCTION

This section describes the implementational details of the guidance scheme described in the previous section. The equations of motion and environment and vehicle characteristics modeled in the predictor algorithm are described. The corrector algorithm to control the final state constraints with the two available control variables is derived. Also derived are the heat rate control and in-flight measurement algorithms. The heat rate control algorithm provides control of the peaks in stagnation heat rate during the early portion of entry. The in-flight measurement algorithm utilizes accelerations measured by the navigation system to more accurately model the expected environment and vehicle characteristics in the predictor algorithm. Because the predictor-corrector algorithm is computationally intensive, areas where significant execution time savings have been or can be realized are indicated. Program listings of the algorithm coded in the HAL/S computer language are presented in "Appendix B. ALGORITHM PROGRAM LISTINGS" on page 135.

As will be seen, the only inputs to the guidance system are the environment and vehicle models, the assumed control profiles, and the navigated state vector. The state vector is an input to any guidance system. The other inputs are developed for the analysis of any new vehicle. Therefore, the guidance system is highly transportable between vehicles because only the vehicle characteristics and aerodynamics model must be changed for a new vehicle.

3.2 UNIT TARGET VECTOR

The target aim point to which the vehicle is to be guided is specified by the longitude and geodetic latitude of the Terminal Area Energy Management (TAEM) interface point which occurs at 80K feet for the Shuttle. This point is selected based on the guidance algorithm employed during the TAEM guidance phase. TAEM guidance provides precise control of vehicle energy during the final stages of entry to guide to a specified runway with acceptable energy. For computational ease, the longitude and geodetic latitude are converted to a target unit vector in Earth-fixed coordinates by first computing the geocentric latitude from,

$$\phi_c = \tan^{-1}\left(\frac{\tan(\phi_g)}{k}\right) \tag{1}$$

where,

$$k = \left(\frac{R_{equator}}{R_{pole}}\right)^2 = \left(\frac{1}{1 - f_{earth}}\right)^2 \tag{2}$$

The unit target vector is then computed from,

$$\vec{i}_t^{EF} = \begin{bmatrix} \cos(\phi_c) \ \cos(\lambda) \\ \cos(\phi_c) \ \sin(\lambda) \\ \sin(\phi_c) \end{bmatrix} \tag{3}$$

Alternatively,

$$\vec{i}_t^{EF} = \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix}_t^{EF} \tag{4}$$

where,

$$i_z = \sin(\phi_c) \tag{5}$$

$$i_x = \cos(\lambda)\ \sqrt{1 - i_z^2} \tag{6}$$

$$i_y = \mathrm{sign}(\lambda)\ \sqrt{1 - i_x^2 - i_z^2} \tag{7}$$

3.3 COMMANDED ATTITUDE COMPUTATION

Because the predictor can not be executed as frequently as analytic guidance algorithms early in the entry, and because it in fact does not have to be executed as frequently, it is necessary to update the commands sent to the vehicle autopilot more frequently than the predictor-corrector execution rate. Typically, this would be done at the rate of current analytic guidance algorithms, e.g., the Space Shuttle rate of .52 hz. The commanded bank angle, ϕ_{cmd}, is computed for the linear bank with velocity profile as shown in Figure 18 on page 95 from the desired bank angle, ϕ_d, and the current navigated inertial velocity magnitude, V_I ,

$$\phi_{cmd} = \phi_d \ \frac{V_I - V_f}{V_{EI} - V_f} \tag{8}$$

to yield the near-optimal linear bank with velocity profile. The desired angle of attack control history is a constant angle of attack, and therefore,

$$\alpha_{cmd} = \alpha_d \tag{9}$$

As implemented in the current design, the guidance algorithm executive is executed at 1.0 hz. The attitude commands are updated at this frequency using Eq. (8) and (9) . The predictor-corrector algorithm is executed at .02 hz. during the entire entry phase, although it is practical to run it much more frequently late in the trajectory when the length of the trajectory to be predicted is short. The possible execution rate of the predictor-corrector for a typical flight computer is addressed in Subsection "4.8 Algorithm Execution Time" on page 66.

3.4 CORRECTOR ALGORITHM

The corrector algorithm is executed to update the commanded attitude control history to be flown. The guidance algorithm controls to two final state constraints, downrange error and crossrange error, using two control variables, a constant angle of attack and the intercept of the bank profile at the entry interface velocity as shown in Figure 18 on page 95 and expressed in Eq. (8) .

Expanding the downrange and crossrange errors in a Taylor series expansion of the control variables and neglecting the second-order and higher terms yields,

$$\Delta DR_e = \frac{\partial DR_e}{\partial \alpha_d} \Delta\alpha_d + \frac{\partial DR_e}{\partial \phi_d} \Delta\phi_d + \ldots \tag{10}$$

$$\Delta CR_e = \frac{\partial CR_e}{\partial \alpha_d} \Delta\alpha_d + \frac{\partial CR_e}{\partial \phi_d} \Delta\phi_d + \ldots \tag{11}$$

To intercept the target, the change in the constraint errors must null the predicted errors, or,

$$\Delta DR_e = -DR_e \tag{12}$$

$$\Delta CR_e = -CR_e \tag{13}$$

Equations (10) through (13) provide a set of two simultaneous equations in two unknowns,

$$\begin{bmatrix} \frac{\partial DR_e}{\partial \alpha_d} & \frac{\partial DR_e}{\partial \phi_d} \\ \frac{\partial CR_e}{\partial \alpha_d} & \frac{\partial CR_e}{\partial \phi_d} \end{bmatrix} \begin{bmatrix} \Delta\alpha_d \\ \Delta\phi_d \end{bmatrix} = \begin{bmatrix} -DR_e \\ -CR_e \end{bmatrix} \tag{14}$$

which are solved for the control variable changes required,

$$\Delta\alpha_d = (\frac{\partial DR_e}{\partial \phi_d} CR_e - \frac{\partial CR_e}{\partial \phi_d} DR_e) \; / \; \det \tag{15}$$

$$\Delta\phi_d = (\frac{\partial CR_e}{\partial \alpha_d} DR_e - \frac{\partial DR_e}{\partial \alpha_d} CR_e) \; / \; \det \tag{16}$$

where det is the determinant of the matrix in Eq. (14). The partial derivatives are approximated by finite difference equations of the form,

$$\frac{\partial DR_e}{\partial \phi_d} = \frac{DR_e(\phi_d = \phi_3) - DR_e(\phi_d = \phi_1)}{\phi_3 - \phi_1} \tag{17}$$

There are four partial derivatives that must be evaluated. They can be evaluated from three predicted trajectories with control histories selected as:

1. $\alpha_1 = \alpha'_{d,} \qquad \phi_1 = \phi'_d$

2. $\alpha_2 = \alpha'_d + \delta\alpha_{d,} \quad \phi_2 = \phi'_d$

3. $\alpha_3 = \alpha'_{d,} \qquad \phi_3 = \phi'_d + \delta\phi_d$

where the primes denote the control variables from the previous guidance solution. The new guidance commands are then,

$$\alpha_d = \alpha'_d + \Delta\alpha_d \tag{18}$$

$$\phi_d = \phi'_d + \Delta\phi_d \tag{19}$$

Protection must be provided for the case where the determinant in Eq. (15) and (16) is small or identically zero which corresponds to a loss of control authority of the control variables over the control constraints. In this case, no change is made to the control variables, and the guidance command from the previous cycle is used. As the vehicle approaches the TAEM interface altitude, the control authority decreases. Large control variable changes become necessary to null the constraint errors in the short flight time remaining. This problem can be avoided in one of two ways. First, the guidance commands can be frozen at a selected point before the termination altitude. For entry guidance, this approach is not preferred because the vehicle still has not landed. Alternatively, the target aim point can be lowered below the TAEM interface altitude point at which TAEM guidance is activated. The decreasing control authority problem is therefore reduced.

For the simulated trajectories in this report, the first approach is employed because it is desired to evaluate guidance performance by considering the dispersions in the final state at the TAEM interface altitude. Because the guidance algorithm controls only the final state and not the intermediate states, it is necessary to target for the point at which the guidance is terminated.

3.5 PREDICTOR ALGORITHM

3.5.1 *Introduction*

The predictor algorithm is a simplified three-degree-of-freedom (3-DOF) trajectory simulator complete with models for those environment and vehicle characteristics necessary to model the translational equations of motion of the vehicle. Because the predictor is computationally intensive, the algorithm must be carefully designed to minimize computation, and the coding of the algorithm in a particular computer language should make use of any language-specific features to reduce computational requirements. Also, because the corrector only utilizes the final state vector errors to correct the control variables, only the accuracy of the predicted final state vector need be considered in selecting those effects to be modeled.

The environmental effects of concern for the long trajectories flown by entry vehicles over large altitude and velocity ranges are:

1. Variation of atmospheric properties with altitude
2. Earth oblateness effect on gravity vector
3. Effect of atmospheric rotation with Earth on relative velocity vector
4. Movement of runway due to Earth rotation

The vehicle characteristics of importance are:

1. Vehicle mass
2. Aerodynamic coefficient variation with flight regime
3. Aerodynamic coefficient variation with angle of attack
4. Control history during trajectory

Dispersions to be considered are:

1. Vehicle mass variation from nominal
2. Winds
3. Atmospheric density variation from nominal atmosphere
4. Aerodynamic coefficient variation from nominal

These dispersions can be measured in-flight because they affect the sensed acceleration measured by the vehicle's inertial navigation system. The estimation of these dispersions is discussed in Subsection "3.6 Estimators" on page 48.

The predictor performs the following computations upon being called by the corrector with a desired control variable history:

1. Initialize the predictor state to the navigated state vector
2. Compute any ancillary parameters from the state vector
3. Compute the total acceleration vector from the predictor state vector and the environment and vehicle models using the control variable profiles specified by the corrector
4. Integrate the equations of motion forward in time one time step
5. Check the predictor termination conditions
    a. Repeat steps 3 and 4 if the conditions are not met
    b. Continue on to step 6 if the conditions are met
6. Compute and return to the corrector the final predicted state errors from the target state vector and the predicted final state vector

3.5.2 Equations of Motion

The corrector provides a time-homogeneous navigated state vector comprised of,

1. The GMT time tag of the state vector, t_{GMT}
2. The inertial position vector, $\vec{R}_I$
3. The inertial velocity vector, $\vec{V}_I$

Also provided is the control variable history to be followed for the prediction. The equations of motion to be integrated are,

$$\frac{d\vec{R}_I}{dt} = \vec{V}_I \tag{20}$$

$$\frac{d\vec{V}_I}{dt} = \vec{A}_I \tag{21}$$

The acceleration is computed from the atmosphere and vehicle models as follows,

$$\vec{A}_I = \frac{\vec{F}_I}{m} = \vec{g}_I + \vec{a}_{aero} \tag{22}$$

The gravitational acceleration, $\vec{g}_I$, is computed including the J_2 term as,

$$\vec{g}_I = -\frac{\mu}{|\vec{R}_I|^2}\,\vec{i}_g \tag{23}$$

where,

$$\vec{i}_g = \vec{i}_R + \frac{3}{2}\,J_2\,\frac{R^2_{equator}}{|\vec{R}_I|^2}\,((1-5\,z^2)\,\vec{i}_R + 2\,z\,\vec{i}_{pole}) \tag{24}$$

and,

$$z = \vec{i}_R \cdot \vec{i}_{pole} \tag{25}$$

The aerodynamic acceleration, $\vec{a}_{aero}$, is computed from,

$$\vec{a}_{aero} = a_{lift}\,\vec{i}_{lift} + a_{drag}\,\vec{i}_{drag} \tag{26}$$

where,

$$a_{lift} = \frac{C_L\,\bar{q}\,S}{m} \tag{27}$$

$$a_{drag} = \frac{C_D\ \bar{q}\ S}{m} \tag{28}$$

$$\bar{q} = \frac{1}{2}\ \rho\ V_R^2 \tag{29}$$

$$V_R^2 = \vec{V}_R \bullet \vec{V}_R \tag{30}$$

$$\vec{V}_R = \vec{V}_I - \vec{\omega}_{earth} \times \vec{R}_I \tag{31}$$

$$\vec{i}_{drag} = -\frac{\vec{V}_R}{|\vec{V}_R|} \tag{32}$$

$$\vec{i}_{lift} = (\vec{i}_{drag} \times \vec{i}_{lat})\ \cos(\phi) + \vec{i}_{lat}\ \sin(\phi) \tag{33}$$

$$\vec{i}_{lat} = \frac{\vec{i}_R \times \vec{i}_{drag}}{|\vec{i}_R \times \vec{i}_{drag}|} \tag{34}$$

$$\vec{i}_R = \frac{\vec{R}_I}{|\vec{R}_I|} \tag{35}$$

The acceleration due to lift, a_{lift}, is more easily computed from,

$$a_{lift} = \frac{L}{D}\ a_{drag} \tag{36}$$

since the nominal lift-to-drag ratio, L/D, is corrected using in-flight accelerometer measurements of the actual vehicle sensed aerodynamic accelerations.

The atmospheric density, ρ, is computed by the atmosphere model using the position vector, $\vec{R}_I$. The 1962 U.S. Standard Atmosphere model is employed and is described in Reference [16]. If another atmosphere model is selected as being a more accurate estimate of the day-of-flight atmosphere, this model would replace the 1962 U.S. Standard Atmosphere model. An operational vehicle might employ monthly or seasonal atmospheres from such sources as the GRAM Atmosphere [9] or even day-of-flight measurements to more accurately model the expected atmosphere in the predictions. The level of accuracy required in the atmosphere model will depend on the vehicle ranging capability and the amount of that capability to be used for a particular entry. Entries to the edges of the footprint will demand a very accurate atmosphere model.

The aerodynamic coefficients are highly vehicle dependent. To minimize computational requirements, they should be updated during the prediction as infrequently as possible. Of course, the update frequency required depends on the trajectory flown and the rate of change of the aerodynamic coefficients with flight regime change. The aerodynamic coefficient model for the ERV is presented in "Appendix A. ERV AERODYNAMICS MODEL" on page 133.

The density, ρ, from the atmosphere model and the lift-to-drag ratio, L/D, from the aerodynamic model are both corrected by in-flight measurements as covered in Subsection "3.6 Estimators" on page 48. The estimated dispersions are compensated for using the following equations,

$$\rho = K_\rho \, \rho_{std} \tag{37}$$

$$\frac{L}{D} = K_{\frac{L}{D}} \left(\frac{C_L}{C_D} \right)_{nom} \tag{38}$$

where the density and lift-to-drag ratio scale factors, K_ρ and $K_{\frac{L}{D}}$, are provided by the estimator and are held constant throughout the prediction being made.

The control history to be followed is the constant angle of attack, α_d, and the linear bank angle with velocity, ϕ. The latter is computed from,

$$\phi = \phi_d \frac{V_I - V_f}{V_{EI} - V_f} \tag{39}$$

where ϕ_d is the intercept of the linear bank angle profile at the entry interface velocity, V_{EI}. Because the entry interface and final velocities are not known a priori, and because small variations in them have little effect on the predicted trajectory compared with the selected control variables' values, the velocities are selected as constant values that cover all expected dispersions in the entry and final velocities. These values are,

$$V_{EI} = 26{,}000 \text{ ft/sec}$$

$$V_f = 1{,}000 \text{ ft/sec}$$

3.5.3 Integration of the Equations of Motion

The equations of motion are integrated using the 4th order Runge-Kutta algorithm with a variable time step to minimize the number of time steps required to integrate the trajectory to the final state. The 4th order Runge-Kutta algorithm requires four evaluations of the acceleration per time step, but permits a time step more than four times as large as an algorithm requiring only one acceleration evaluation per time step. The Runge-Kutta solution [17] for the differential equations of motion of the form,

$$\frac{d\vec{R}_I}{dt} = \vec{V}_I \tag{40}$$

$$\frac{d\vec{V}_I}{dt} = f(t, \vec{R}_I, \vec{V}_I) \tag{41}$$

is,

$$\vec{R}_l(t+\Delta t) = \vec{R}_l(t) + \frac{\Delta t}{6}(\vec{K}_0 + 2\vec{K}_1 + 2\vec{K}_2 + \vec{K}_3) \tag{42}$$

$$\vec{V}_l(t+\Delta t) = \vec{V}_l(t) + \frac{\Delta t}{6}(\vec{K'}_0 + 2\vec{K'}_1 + 2\vec{K'}_2 + \vec{K'}_3) \tag{43}$$

where,

$$\vec{K}_0 = \vec{V}_l \tag{44}$$

$$\vec{K}_1 = (\vec{V}_l + \frac{\vec{K'}_0}{2}) \tag{45}$$

$$\vec{K}_2 = (\vec{V}_l + \frac{\vec{K'}_1}{2}) \tag{46}$$

$$\vec{K}_3 = (\vec{V}_l + \vec{K'}_2) \tag{47}$$

$$\vec{K'}_0 = f(t, \vec{R}_l, \vec{V}_l) \tag{48}$$

$$\vec{K'}_1 = f(t + \frac{\Delta t}{2}, \vec{R}_l + \Delta t\frac{\vec{K}_0}{2}, \vec{V}_l + \Delta t\frac{\vec{K'}_0}{2}) \tag{49}$$

$$\vec{K'}_2 = f(t + \frac{\Delta t}{2}, \vec{R}_l + \Delta t\frac{\vec{K}_1}{2}, \vec{V}_l + \Delta t\frac{\vec{K'}_1}{2}) \tag{50}$$

$$\vec{K'}_3 = f(t + \Delta t, \vec{R}_l + \Delta t\vec{K}_2, \vec{V}_l + \Delta t\vec{K'}_2) \tag{51}$$

The time step is varied inversely with the total acceleration on the vehicle. This method of time step control was selected because of its simplicity. The time step control equation is of the form,

$$\Delta t = \frac{K_{\Delta t}}{|\vec{A}_l|} \tag{52}$$

and the time step is limited between a minimum and maximum value,

$$\Delta t = midval(\Delta t_{min}, \Delta t, \Delta t_{max}) \tag{53}$$

The optimization of the integration algorithm is important in developing a flight quality algorithm, but is beyond the scope of this study. Higher-order integration algorithms with time step control methods [17] may yield significant reductions in the required computation time.

3.5.4 Termination Conditions for the Predictor

After each integration time step, the predicted state is compared with the termination condition. The termination condition is defined by the altitude of TAEM interface (80K feet). Because the predicted state at the TAEM interface altitude may have a relatively large altitude rate and range rate, the predictor must be terminated accurately to provide an altitude-homogeneous set of predicted state errors. Also, the variable time step control may allow large integration time steps if the acceleration is low near the final state, further complicating the task of terminating accurately. Reasonable altitude homogeneity is ensured by forcing use of the minimum integration time step starting some safe altitude above the termination altitude.

3.5.5 Final State Error Computation

The final state errors are computed from the unit target vector and the predicted final state vector. Because the target is fixed to the Earth and moves a significant distance during the long entry trajectory, the rotation of the Earth must be considered. This is done by transforming the final state vector from inertial to Earth-fixed coordinates with the rotation matrix $\hat{M}_I^{EF}$ which is computed from the predicted termination time, the known orientation of the Earth at some epoch time, and the known rotation rate of the Earth. This computation is performed in the Earth-Fixed-From-Reference subroutine of the predictor-corrector which may actually be a GN&C utility function also employed by the navigation principal function.

The downrange and crossrange errors are defined as shown in Figure 19 on page 96. The errors are computed by first computing the downrange (in-plane) and crossrange (perpendicular) directions as follows,

$$\vec{R}_I^{EF} = \hat{M}_I^{EF} \vec{R}_I \tag{54}$$

$$\vec{i}_R^{EF} = \frac{\vec{R}_I^{EF}}{|\vec{R}_I^{EF}|} \tag{55}$$

$$\vec{V}_R^{EF} = \hat{M}_I^{EF} \vec{V}_R^I \tag{56}$$

$$\vec{i}_{perpen}^{EF} = \frac{\vec{i}_R^{EF} \times \vec{V}_R^{EF}}{|\vec{i}_R^{EF} \times \vec{V}_R^{EF}|} \tag{57}$$

$$\vec{i}_{inplane}^{EF} = \frac{\vec{i}_t^{EF} - (\vec{i}_t^{EF} \bullet \vec{i}_{perpen}^{EF})\, \vec{i}_{perpen}^{EF}}{|\vec{i}_t^{EF} - (\vec{i}_t^{EF} \bullet \vec{i}_{perpen}^{EF})\, \vec{i}_{perpen}^{EF}|} \tag{58}$$

The downrange and crossrange errors are then,

$$DR_e = R_{equator} \cos^{-1}(\vec{i}_R^{EF} \times \vec{i}_{inplane}^{EF})\ sign((\vec{i}_R^{EF} \times \vec{i}_{inplane}^{EF}) \bullet \vec{i}_{perpen}^{EF}) \tag{59}$$

$$CR_e = R_{equator} \cos^{-1}(\vec{i}_{inplane}^{EF} \bullet \vec{i}_t^{EF})\ sign((\vec{i}_{inplane}^{EF} \times \vec{i}_t^{EF}) \bullet (\vec{i}_{perpen}^{EF} \times \vec{i}_{inplane}^{EF})) \tag{60}$$

These errors have the dimensions of $R_{equator}$ and are converted to nautical miles for ease of interpretation.

3.5.6 *Algorithm Coding*

A few comments regarding implementation of the predictor are appropriate. The computations required to update the aerodynamic coefficients are the major computational load for the predictor. It was found that it is not necessary to update the aerodynamics on each of the four acceleration evaluations of the 4th order Runge-Kutta algorithm. They are therefore only evaluated once each integration time step. The computational load could be reduced further if they are only updated when the independent variables (altitude, viscous interaction parameter, and Mach Number) change by a significant amount from the previous update. Also, although not done in this implementation, the aerodynamic coefficients should be curve-fit if possible to avoid a table lookup and interpolation implementation. It is noted in Figures 20 on page 97 and 21 on page 98 that the aerodynamic coefficients do not change very much below 300K feet until the Mach Number decreases below 2, so perhaps, two tables or curve-fits would suffice instead of the thirty tables currently used.

3.6 ESTIMATORS

The final state predicted by the predictor algorithm for a particular control history is a function of the assumed environment and vehicle characteristics. The accuracy of the predicted final state can be increased, and hence, the guidance margin increased, if in-flight measurements are utilized to make the assumed models more accurately reflect the conditions actually experienced by the vehicle.

The accelerations modeled in the predictor are due to gravity and the aerodynamic forces. The gravity acceleration can be modeled to sufficient accuracy using standard gravity models. However, the aerodynamic accelerations are subject to significant variations due to uncertainties in the atmospheric density, atmospheric winds, vehicle aerodynamics, and vehicle mass. These uncertainties can be compensated for in the predictor by applying a multiplicative scale factor to the lift and drag accelerations modeled in the predictor that is equal to the ratio of the actual accelerations experienced to the predicted accelerations at any point in the trajectory.

The measured lift and drag accelerations are derived from the inertial measurement system sensed acceleration assuming a zero sideslip angle as follows,

$$\hat{a}_{drag} = -\vec{a}_I \bullet \frac{\vec{V}_R}{|\vec{V}_R|} \tag{61}$$

$$\hat{a}_{lift} = \sqrt{\vec{a}_I \bullet \vec{a}_I - \hat{a}^2_{drag}} \tag{62}$$

where the inertial acceleration, $\vec{a}_I$, is computed by back-differencing the accumulated sensed velocity counts from the inertial measurement unit,

$$\vec{a}_I = \frac{\vec{V}_I^{imu} - \vec{V}_I^{imu\ past}}{\Delta t_{imu}} \tag{63}$$

In the predictor, the aerodynamic accelerations are,

$$a_{drag} = \frac{C_D\ S}{m}\ \frac{1}{2}\ \rho\ V_R^2 \tag{64}$$

$$a_{lift} = \frac{L}{D} a_{drag} \tag{65}$$

Data from the Shuttle program [10] shows that the primary dispersion affecting the aerodynamic acceleration is in the atmospheric density. Further, over large altitude ranges, this dispersion can be modeled to an accuracy sufficient for the prediction process as a constant multiplicative bias. Therefore, for implementational purposes, the dispersion in the aerodynamic accelerations due to the atmospheric uncertainties will be lumped into a density scale factor as follows,

$$K_\rho = \frac{\hat{\rho}}{\rho_{std}} \tag{66}$$

where,

$$\hat{\rho} = \frac{2\,\hat{a}_{drag}}{V_R^2} \left(\frac{m}{C_D\, S} \right)_{nom} \tag{67}$$

and the values for the nominal vehicle characteristics and the nominal atmospheric density are determined using the predictor models for the vehicle state at the time of the measurement. Because the nominal ballistic coefficient is assumed in deriving the measured density, and the measured acceleration is due to the actual ballistic coefficient, uncertainties in the ballistic coefficient will be reflected in the measured density. The equation for the drag acceleration in the predictor is then,

$$a_{drag} = \left(\frac{C_D\, S}{m} \right)_{nom} \frac{1}{2} V_R^2\, K_\rho\, \rho_{std} \tag{68}$$

or substituting for K_ρ from Eq. (66) yields,

$$a_{drag} = \left(\frac{C_D\, S}{m} \right)_{nom} \frac{1}{2} V_R^2\, \hat{\rho} \tag{69}$$

Substituting for $\hat{\rho}$ from Eq. (67) then yields,

$$a_{drag} = \hat{a}_{drag} \tag{70}$$

so the modeled drag is corrected for the dispersed drag coefficient, density, relative velocity, and vehicle mass.

In general, the measured drag acceleration is a noisy signal and will exhibit short term variations due to short lived local atmospheric dispersions [10]. Filtering of the density scale factor is therefore necessary and is implemented using a first-order filter,

$$K_{\rho} = (1 - K_1)\, K_{\rho} + K_1 \frac{\hat{\rho}}{\rho_{std}} \tag{71}$$

which has a time constant , τ_{ρ}, of,

$$\tau_{\rho} = -\frac{\Delta t}{\ln(1 - K_1)} \tag{72}$$

where Δt is the sample rate of the measured drag acceleration, and K_1 is the filter gain. A similar lift-to-drag ratio scale factor is derived and applied to the lift acceleration,

$$a_{lift} = K_{\frac{L}{D}} \left(\frac{L}{D}\right)_{nom} a_{drag} \tag{73}$$

where,

$$K_{\frac{L}{D}} = \frac{\widehat{\left(\frac{L}{D}\right)}}{\left(\frac{L}{D}\right)_{nom}} \tag{74}$$

and,

$$\left(\frac{\hat{L}}{D}\right) = \frac{\hat{a}_{lift}}{\hat{a}_{drag}} \tag{75}$$

Again, filtering is necessary,

$$K_{\frac{L}{D}} = (1 - K_2)\, K_{\frac{L}{D}} + K_2 \frac{\left(\frac{\hat{L}}{D}\right)}{\left(\frac{L}{D}\right)_{nom}} \tag{76}$$

yielding a time constant, $\tau_{\frac{L}{D}}$, of,

$$\tau_{\frac{L}{D}} = -\frac{\Delta t}{\ln(1 - K_2)} \tag{77}$$

A time constant of 25 seconds was selected for both the density and L/D filters. This value filtered out the high frequency density shear components seen in the Shuttle profiles while still providing adequate response to long term disturbances.

3.7 HEAT RATE CONTROL

The primary trajectory constraint on entry vehicles is the maximum heat rate the vehicle can withstand. In general, the thermal protection system material is selected to withstand the maximum local heat rate on any particular portion of the vehicle, and the material thickness is selected to withstand the total integrated heat load over the trajectory. Accurate pre-flight predictions of the expected heat rate during entry can significantly reduce the thermal protection system weight yielding significant performance increases for an entire mission.

Inspecting the reference trajectories in Figure 16 on page 93 shows that sharp peaks in the heat rate occur. If these peaks are accurately controlled, and this control can be accomplished using only short term departures from the predictor assumed control history, no significant departure will occur from the desired trajectory.

Heat rate control can be accomplished using either angle of attack, bank angle, or a combination of both. Of these, bank angle alone is preferred because a constant angle of attack trajectory is assumed and because angle of attack changes the vehicle drag coefficient resulting in a rapid change in energy rate and a rapid departure from the desired trajectory. Also, most entry vehicles restrict the angle of attack range during maximum heat rate regions to reduce the area on the vehicle that must be protected from the high heat rate. Although the ERV does not need to restrict the angle of attack range, and hence, the guidance does not provide for such a capabilty, the restriction can be handled by replacing the constant angle of attack control history by a reference angle of attack control history about which a constant angle of attack bias is applied for control.

Heat rate control is accomplished by computing the incremental bank angle required to fly along the specified heat rate boundary (assumed to be a constant heat rate for any flight regime) and then modulating bank angle according to the guidance value or the guidance value plus the incremental lift for heat rate control, whichever requires more lift up. Hence, no effort is made to pull the vehicle down into the atmosphere to follow the heat rate boundary; instead, lift up is applied if the vehicle is flying "too low". The incremental lift for heat rate control is computed to provide a second-order control response as follows,

$$\cos(\Delta\phi) \;=\; \frac{K_{\ddot{Q}}}{\bar{q}}\,(\ddot{Q} \;-\; \ddot{Q}_{des}) \;+\; \frac{K_{\dot{Q}}}{\bar{q}}\,(\dot{Q} \;-\; \dot{Q}_{\lim}) \tag{78}$$

To fly along a constant heat rate boundary,

$$\dot{Q}_{\lim} \;=\; \text{constant} \tag{79}$$

and the desired rate of change of heat rate, $\ddot{Q}_{des}$, is,

$$\ddot{Q}_{des} \;=\; 0 \tag{80}$$

so,

$$\cos(\Delta\phi) \;=\; \frac{K_{\ddot{Q}}}{\bar{q}}\,\ddot{Q} \;+\; \frac{K_{\dot{Q}}}{\bar{q}}\,(\dot{Q} \;-\; \dot{Q}_{\lim}) \tag{81}$$

The stagnation heat rate is determined using the Engineering Correlation Formula [18] for a one foot radius reference sphere as,

$$\dot{Q} \;=\; 17700\,\sqrt{\rho}\,\left(\frac{V_R}{10000}\right)^{3.05} \tag{82}$$

The time rate of change of heat rate, $\ddot{Q}$, is determined by back-differencing the heat rate between guidance cycles,

$$\ddot{Q} \;=\; \frac{\dot{Q} \;-\; \dot{Q}_{past}}{\Delta t} \tag{83}$$

The equations of motion assuming small flight path angle yield,

$$\ddot{h} \;=\; \frac{C_L\,\bar{q}\,S}{m}\cos(\phi) \;-\; g \tag{84}$$

Considering only the perturbations due to the incremental lift, $\cos(\Delta\phi)$, from Eq. (81) yields,

$$\ddot{h} - \frac{C_L\ S}{m} K_{\ddot{Q}}\ \ddot{Q} + K_{\dot{Q}}\ (\dot{Q} - \dot{Q}_{\text{lim}}) = 0 \tag{85}$$

Proper selection of the gains $K_{\dot{Q}}$ and $K_{\ddot{Q}}$ is accomplished by linearizing Eq. (85) in altitude and assuming that the time rate of change of V_R is small compared to the change in $\sqrt{\rho}$. With these assumptions,

$$\dot{Q} = 17700 \left(\frac{V_R}{10000}\right)^{3.05} \frac{d\sqrt{\rho}}{d\,h}\ h \tag{86}$$

and,

$$\ddot{Q} = 17700 \left(\frac{V_R}{10000}\right)^{3.05} \frac{d\sqrt{\rho}}{d\,h}\ \frac{d\,h}{d\,t} \tag{87}$$

Therefore, the homogeneous second-order differential equation in altitude is,

$$\ddot{h} + K\ K_{\ddot{Q}}\ \dot{h} + K\ K_{\dot{Q}}\ h = 0 \tag{88}$$

where,

$$K = -\frac{C_L\ S}{m}\ 17700 \left(\frac{V_R}{10000}\right)^{3.05} \frac{d\sqrt{\rho}}{d\,h} \tag{89}$$

The natural frequency and damping ratio of the second-order differential equation are,

$$\omega_n = \sqrt{K\ K_{\dot{Q}}} \tag{90}$$

$$\zeta = \frac{K\ K_{\ddot{Q}}}{2\ \omega_n} \tag{91}$$

or alternatively, for a desired natural frequency and damping ratio, $K_{\dot{Q}}$ and $K_{\ddot{Q}}$ are selected as,

$$K_{\dot{Q}} = \frac{\omega_n^2}{K} \tag{92}$$

$$K_{\ddot{Q}} = \frac{2\ \zeta\ \omega_n}{K} \tag{93}$$

The derivative in Eq. (89) can be evaluated assuming an exponential atmosphere of the form,

$$\rho = \rho_{sl}\ e^{-\left(h/h_s\right)} \tag{94}$$

yielding,

$$\frac{d\sqrt{\rho}}{d\,h} = -\frac{\sqrt{\rho_{sl}}}{2\,h_s}\ e^{-\left(h/2\,h_s\right)} \tag{95}$$

This logic is contained in the guidance algorithm in the Heat Rate Control subroutine. The incremental lift required for heat rate control is provided to the Attitude Command subroutine which adds it into the guidance command if it requires more lift up than the guidance command. This occurs when the incremental lift given by Eq. (81) is greater than zero,

$$\cos(\Delta\phi) > 0 \tag{96}$$

Appropriate values of the natural frequency and damping ratio were determined parametrically as,

$$\omega_n = 0.10\ \frac{\text{rad}}{\text{sec}} \tag{97}$$

$$\zeta = 1.00 \tag{98}$$

4.0 PERFORMANCE

4.1 SIMULATOR

Open-loop and closed-loop entry trajectories were simulated for the Entry Research Vehicle (ERV) using a derivative of the 6-DOF Aeroassist Flight Experiment Simulator (AFESIM) [19] developed at The Charles Stark Draper Laboratory which is coded in the HAL computer language. For this study, the aerodynamic model described in "Appendix A. ERV AERODYNAMICS MODEL" on page 133 and the wind model shown in Figure 2 on page 79 were incorporated into the AFESIM. The characteristics of the ERV [14] are listed in Table 1 on page 73. The entry conditions with which all trajectories were initialized are also listed in Table 1 on page 73. Because only the performance characteristics of the guidance were being evaluated, the simulator was operated in the 3-DOF mode.

4.2 OPEN-LOOP FOOTPRINT

Open-loop trajectories were run using the constant angle of attack and linear bank with velocity profiles to determine the portion of the footprint achievable. All trajectories were terminated at the TAEM interface altitude of 80K feet, so the footprint can be increased about 100 nautical miles in all directions due to the range flown below 80K feet.

Figure 22 on page 99 shows the lift-to-drag ratio, L/D, for the ERV at Mach 10 versus angle of attack, α. It is seen that maximum L/D is obtained at an angle of attack of 15 degrees. It is desirable to fly on the back side of the L/D curve (angle of attack greater than 15 degrees) so as to maximize the drag coefficient for a given L/D. This reduces heating by causing a quicker loss of velocity early in entry than flying at the same L/D on the front side of the L/D curve. The L/D versus angle of attack curve shows the same shape with the maximum L/D at 15 degrees for all flight regimes with only a variation in the magnitude of L/D across the angle of attack range. Therefore, angle of attack is modulated between 15 and 50 degrees for the footprint with 15 degrees corresponding to maximum L/D and 50 degrees corresponding to minimum L/D.

The open-loop footprint is shown in Figure 23 on page 100. Also shown for comparison is the reported footprint for a heat load limit of 175K BTU/sq ft (shown earlier in Figure 12 on page 89). That footprint included the range flown below 80K feet, hence the slight differences. It is seen that almost the entire reportedly achievable footprint is captured with the assumed control profile. Most importantly, all of the maximum crossrange region is reached when the range flown below 80K feet is included. Also, most of the minimum downrange region of the footprint was captured even though the control profiles used do not correspond to the optimal profiles determined using POST and shown in Figures 14 on page 91 and 15 on page 92. Additionally, the footprint reported in Reference [14] does not include the large area in the minimum downrange region that the open-loop trajectories reached. The small area not reached in the minimum downrange region by the control profiles is relatively unimportant because the downrange ranging capability of the vehicle can be adjusted by changing the deorbit time. A vehicle in low earth orbit travels at about four nautical miles per second, so downrange is easily adjusted while on-orbit.

Because the predictor-corrector guidance algorithm will follow the same control histories as used to generate the open-loop footprint for a nominal trajectory, the guidance algorithm can reach all of the open-loop footprint for nominal conditions. It is seen that the achievable footprint is bounded by the heat rate and heat load limits imposed on the ERV. At least an additional 2000 nautical miles of ranging capability in the downrange direction exists if the heat limits are relaxed.

4.3 EFFECT OF DISPERSIONS ON FOOTPRINT

The effect of dispersions on the achievable footprint was determined by repeating the open-loop trajectories with the dispersions discussed in Subsection "2.2 Dispersions" on page 26. The worst-case (3σ) dispersions are summarized in Table 2 on page 73. Table 3 on page 74 shows the dispersions in downrange and crossrange for three of the control histories in the maximum downrange region of the footprint. It is seen that only variations in the lift and drag coefficients cause significant dispersions in the final state. Also, it is seen that the effect of a +10% C_L dispersion is the same as that of a -10% C_D dispersion. This is expected because both dispersions cause the same increase in the vehicle L/D. The same occurs for a -10% C_L dispersion and a +10% C_D dispersion, both of which decrease the vehicle L/D.

The effects of the dispersions on trajectories to the maximum crossrange region of the footprint are seen in Table 4 on page 75. Again, it is seen that aerodynamic dispersions have the greatest effect. A dispersion that increases L/D increases the range, while a dispersion that decreases L/D decreases the range.

Table 5 on page 76 shows the effects of the dispersions on the minimum downrange region of the footprint. The worst-case range dispersions again occur for the aerodynamic dispersions.

4.4 ESTIMATOR PERFORMANCE

Figure 24 on page 101 shows the time response of the density filter with a 25 second time constant for the STS-9 atmosphere. This trajectory also has dispersions of +1.9% in C_D, -3.2% in mass, and a 63.8% crosswind. Therefore, the filter output does not follow the actual density dispersion also shown in the figure. When the acceleration level is below 0.07 g's, the measurements are not incorporated, so the filter is inactive before 300 seconds and from 600 to 850 seconds. As the velocity drops, the wind becomes a greater contributor to the measured density error, hence the divergence in the measured density ratio starting at 1000 seconds. Figure 25 on page 102 shows the response of the L/D filter with a 25 second time constant for a -1.9% C_L and a +1.9% C_D dispersion. Again, the winds affect the measurement by creating errors in the navigated angle of attack, so the estimated L/D ratio is slightly in error.

The use of an air data system like the Shuttle Entry Air Data System (SEADS) could significantly improve the estimation process by providing accurate estimates of the angle of attack, atmospheric density, and wind magnitude and direction. More accurate estimates will increase the guidance margin, thereby increasing the achievable footprint for dispersed trajectories.

4.5 CLOSED-LOOP PERFORMANCE

Based on the results of the open-loop trajectories with dispersions, worst-case dispersions were selected for each of three regions of the footprint: maximum downrange, maximum crossrange, and minimum downrange. Closed-loop trajectories with the predictor-corrector guidance algorithm were then run to the three regions of the footprint. The three target points selected for the closed-loop performance evaluation are shown in Figure 23 on page 100. The 3σ errors defined in Table 2 on page 73 were scaled such that the total error due to multiple error sources would still represent a 3σ dispersion so as to test the guidance system for reasonably probable dispersion cases [20]. To run all dispersions at their 3σ levels would be unrealistic.

The nominal and dispersed results for trajectories to each of the three regions are listed in Tables 6 on page 77 through 8 on page 77 . Plots of selected parameters from these cases are included. Figures 26 on page 103 through 31 on page 108 present the altitude, velocity, heat rate, heat load, downrange, and crossrange time histories for the nominal maximum downrange trajectory. Figures 32 on page 109 through 37 on page 114 present the altitude, velocity, heat rate, heat load, downrange, and crossrange time histories for the nominal maximum crossrange trajectory. Figures 38 on page 115 through 43 on page 120 present the altitude, velocity, heat rate, heat load, downrange, and crossrange time histories for the nominal minimum downrange trajectory. In each of these cases, it is seen that the heat rate does not approach the heat rate limit, so no incremental bank angle is needed for heat rate control.

The control histories for the nominal maximum downrange trajectory and the dispersed case listed second in Table 6 on page 77 are presented in Figure 44 on page 121 and

Figure 45 on page 122 . Figure 44 shows the angle of attack histories for the nominal and dispersed maximum downrange cases. It is seen that a two degree change in angle of attack is required early in the trajectory increasing to four degrees by the end of the trajectory. Figure 45 shows the bank angle histories for the nominal and dispersed maximum downrange cases. The bank angle required shows no change from zero degrees for this case.

The control histories for the nominal maximum crossrange trajectory and the dispersed case listed second in Table 7 on page 77 are presented in Figure 46 on page 123 and Figure 47 on page 124 . Again, it is seen that a four degree change in angle of attack is required for the dispersed case. The bank angle history shows no change for the dispersed case from that of the nominal case.

The control histories for the nominal minimum downrange trajectory and the dispersed case listed second in Table 8 on page 77 are presented in Figure 48 on page 125 and Figure 49 on page 126 . For this case, approximately a one degree change in angle of attack is required. No change is required in the bank angle profile.

The required change in angle of attack for each of the dispersed cases shown was primarily due to the change in the vehicle L/D as this was shown to be the primary dispersion source in Subsection "4.3 Effect of Dispersions on Footprint" on page 59. The breaking point of the guidance occurs when the vehicle does not have enough L/D range to overcome the loss in L/D due to aerodynamic dispersions. As mentioned previously, the Shuttle entry guidance algorithm was required to guide to the TAEM interface aim point to within 2.5 nautical miles of position. The results presented for the nominal and dispersed cases show that this requirement is met with the predictor-corrector guidance algorithm. Also, the trajectory plots show that the algorithm achieves this performance with very infrequent guidance updates (.02 hz.) and with very small control variable changes from the nominal constant angle of attack and linear bank with velocity profiles. Most i.nportantly, almost all of the achievable footprint is captured using the predictor-corrector algorithm.

4.6 HEAT RATE CONTROL PERFORMANCE

The closed-loop trajectories shown previously did not require heat rate control because the maximum heat rate experienced was significantly lower than the limit imposed on the ERV. The time responses for bank angle, angle of attack, and heat rate for the beginning of a typical trajectory with and without heat rate control are shown in Figures 50 on page 127 through 52 on page 129. The resulting bank angle versus velocity profile is shown in Figure 53 on page 130. These trajectories are for the middle of the footprint where the peak heat rate does not exceed the limit for the ERV. Therefore, for illustrative purposes, the heat rate limit was reduced to 100 BTU/sq ft/sec. Comparing the trajectories with and without heat rate control, it is seen that the heat rate control takes place over a fairly long time range, but requires a significant departure from the linear bank profile over only a very short velocity range. The impact on the trajectory is therefore small, and the predictor-corrector stays converged on almost the same control history even though the vehicle does not follow the assumed control profile during the heat rate control area.

4.7 OVERCONTROL

For those trajectories not at the edge of the footprint, excess vehicle capability exists that can be utilized to increase guidance margin for dispersions that may occur later in the trajectory. For example, a 13,800 nautical mile downrange trajectory for the ERV only requires flying at 20 degrees angle of attack instead of 15 degrees for the nominal trajectory. The ERV can modulate angle of attack between 15 degrees (maximum L/D) and 50 degrees (minimum L/D) on the back side of the L/D curve, so the modulation capability is not equally centered about the commanded angle of attack if flying at 20 degrees. By flying at 15 degrees (maximum L/D) early in the trajectory, guidance can center the remaining guidance capability equally about the aim point to cover dispersions in all directions, not just those that require less L/D to reach the target point. This approach is referred to here as overcontrol or command biasing.

Overcontrol can be implemented in several ways. First, the command can be biased from the desired command when that command is not in the center of the modulation range. As the vehicle flies a biased angle of attack, for example, the predicted final state will differ from that for the unbiased command in such a direction that the next guidance command will be moved in the direction opposite to the bias. By biasing in the proper direction, the command can be driven toward the center of the modulation range. If the guidance requires an L/D higher than that in the middle of the L/D range, flying at an even higher L/D will drive the required L/D toward the middle. Secondly, the target aim point can be moved from the nominal aim point early in the entry. For example, for a trajectory to the maximum downrange region of the footprint, the target aim point can be moved even farther downrange. Of course, at some point in the trajectory, the aim point must be moved back to the desired point.

The first approach was implemented in the predictor-corrector algorithm by biasing the angle of attack by five degrees when it was more than two degrees away from 30 degrees. The biasing was terminated at an inertial velocity of 13,500 feet per second so as to allow the guidance to fly the proper control history near the end of the trajectory to reach the target aim point.

Figure 54 on page 131 compares the angle of attack control history for a 13,760 nautical mile downrange trajectory with the dispersions used for the closed-loop trajectories shown earlier. Without overcontrol, the vehicle misses the target aim point by 19.20 nautical miles. This occurs because the wind contribution to the dispersion increases as the vehicle velocity drops, so the multiplicative scale factor on density does not properly model this dispersion. As the wind contribution increases, a higher L/D is required, and the angle of attack is driven to 15 degrees or maximum L/D. Because maximum L/D was not utilized earlier in the trajectory, the vehicle did not reach the target. Late in the trajectory, the predictor-corrector goes unconverged as control authority is exhausted, causing the angle of attack to jump between 15 and 30 degrees. By this point, the target aim point was unreachable anyway due to the dispersions.

With command biasing, the commanded angle of attack early in the trajectory is that corresponding to maximum L/D or 15 degrees. It is seen that biasing drives the commanded angle of attack to 25 degrees once the biasing is terminated at a velocity of 13,500 feet per second or a time of 3,700 seconds. Later, when the wind dispersion drives the angle of attack toward 15 degrees, there is significant margin remaining, and the angle of attack is only driven to 24.5 degrees by the dispersion. With command biasing, the miss distance at TAEM interface is only 0.27 nautical miles. Therefore, guidance margin is increased by using overcontrol. More of the theoretically achievable footprint is attainable for dispersed cases. An even larger magnitude dispersion could have been handled late in the trajectory since the angle of attack was not driven to that for maximum L/D.

Further work is needed in this area to determine the proper way to utilize overcontrol to maximize guidance margin for the expected dispersions. The probability of the various dispersions occurring and the histories of those dispersions along a trajectory must be considered. For example, if a "thick" atmosphere is encountered early in the trajectory equal to the worst-case expected dispersion, it is highly unlikely that the atmosphere will get "thicker" later in the trajectory. Therefore, it is unnecessary to preserve guidance margin in the direction needed to cover a "thicker" atmosphere beyond that already required for the expected worst-case atmosphere. Such considerations should be taken into account in the design of the overcontrol algorithm.

4.8 ALGORITHM EXECUTION TIME

An estimate of the execution time required for the predictor-corrector algorithm was made using the execution time estimate feature of the HAL compiler. The estimate is for the AP101 Shuttle flight computer. Figure 55 on page 132 shows the execution time required in seconds as a function of the time to the TAEM interface point for a maximum downrange trajectory. It is seen that early in the entry when the trajectory to be predicted is long, the predictor requires 43.7 seconds of CPU time. When only 500 seconds to the TAEM interface point remains, the required time drops to 4.5 seconds. This figure can also be interpreted as the minimum update interval for the predictor-corrector. Also, the guidance command will be computed and sent to the vehicle autopilot a period of time after the start of the guidance cycle equal to the required execution time. It is seen that early in the entry, a significant delay occurs between the start of the guidance cycle and the computation of the guidance command. This delay was not simulated in the closed-loop trajectories, but will have a minimal effect on the guidance margin because the guidance is not trying to fly a reference trajectory like the analytic guidance algorithms. The predictor-corrector is numerically computing a trajectory that will fly directly to the target aim point. Any error that builds up between the start of the guidance cycle and the issuing of the guidance command can be nulled easily since the entry is long, and the error will shrink as the delay decreases with decreasing time to the TAEM interface point.

The Shuttle AP101 CPU is the product of early 1970's technology and is significantly slower than flight computers that might be employed in future entry vehicles. The 80C86 CPU for example is two to five times faster than the AP101 CPU, so the execution time required shown in Figure 55 on page 132 can be scaled down by a factor of two to five. Computers utilizing parallel processing architecture could predict the three required trajectories simultaneously in three CPUs, cutting the required execution time by a factor of three. If scaled by a factor of four due to the faster CPU and a factor of three due to parallel processing architecture, the maximum time required drops to 3.6 seconds, and the time with 500 seconds remaining to the TAEM interface point drops to 0.4 seconds. The predictor-corrector is therefore a viable guidance scheme for future entry vehicles.

5.0 *FUTURE RESEARCH TOPICS AND CONCLUSIONS*

5.1 FUTURE RESEARCH TOPICS

Several topics for further algorithm development and optimization are discussed. These are:

1. Further reductions in CPU execution time
2. Use of an air data system for in-flight measurements
3. Use of overcontrol to increase guidance margin
4. Control of more than two state constraints

Optimization of the predictor algorithm and the integration scheme can yield significant reductions in execution time beyond that already attained. Simplifying the aerodynamic model can yield a great reduction in execution time and an equally important reduction in the computer core required. The current model has 30 tables, each with 51 breakpoints over the angle of attack range. A curve fit of the aerodynamic coefficients over the angle of attack range and the flow regimes would reduce the core required to store the model data and the computations required for each lookup.

The estimator algorithm was shown to be effective in determining the dispersions from in-flight measurements. However, the estimator is unable to differentiate between density dispersions and atmospheric winds. Figure 24 on page 101 showed that the multiplicative density scale factor did not accurately model the wind contribution to the drag acceleration because the relative contribution of the wind to the dispersion increases as the vehicle

PRECEDING PAGE BLANK NOT FILMED

PAGE 68 INTENTIONALLY BLANK

velocity drops late in the trajectory. An air data system could provide an independent measurement of the atmospheric winds, improving the estimation process and increasing guidance margin by increasing the accuracy of the predicted trajectories.

The concept of overcontrol was introduced and shown to be effective for at least one dispersed case. Further investigations should be made to determine how much overcontrol is optimal for the expected dispersions. It may be possible to use the sensitivities of the constraints to the control variables to determine a proper amount of command biasing for any particular dispersion at any point in the trajectory.

Only the downrange error and crossrange error at TAEM interface are controlled in the current design. The vehicle energy is not controlled which can allow significant dispersions in the ranging capability during the TAEM phase of entry. Approaches include redefining the TAEM aim point in terms of a desired energy level or utilizing a third control variable to provide control over an energy level constraint. The Space Shuttle makes use of a split rudder as a speedbrake to provide a large energy control capability. Such an approach could be utilized with the predictor-corrector by computing the sensitivity of the three constraints to the three control variables. This would require four predictions instead of the three currently needed, but the fourth prediction could be made only during the latter part of entry to clean up any dispersions in energy level that occur during the entry due to dispersions. The CPU execution time would then increase by one-third over that currently projected when the third constraint is controlled.

5.2 CONCLUSIONS

A predictor-corrector entry guidance algorithm has been demonstrated that exhibits excellent performance and almost complete coverage of the achievable footprint. This algorithm employs a simple control variable history to achieve near-optimal guidance for the maximum downrange and maximum crossrange trajectories. Explicit heat rate control is employed without significantly impacting the achievable footprint. This is achieved because unlike previous guidance algorithms that included a long heat rate control phase with no active targeting, the proposed algorithm always actively targets to the aim point and only controls heat rate in the short high heat rate regions as required.

The algorithm has been demonstrated to handle atmospheric and aerodynamic dispersions within the capability of the vehicle. The required computer execution time is shown to be within the capability of new flight computers.

Algorithm adaptability is provided through the utilization of in-flight measurements to improve the accuracy of the predicted trajectory. Algorithm maintenance is simplified because there are no reference trajectories used, and there are a minimum of vehicle/mission-specific input parameters (I-loads). Transportability of the algorithm between different entry vehicles is provided by eliminating vehicle-specific entry phases other than the heat rate control phase which only requires the input of a heat rate limit. The guidance algorithm does require a vehicle aerodynamic model, but this is developed in the normal vehicle definition phase anyway.

In summary, an entry guidance algorithm has been developed that achieves near-optimal performance while maximizing flexibility, adaptability, and transportability. Although more computationally intensive than analytic algorithms, execution of the predictor-corrector is within the capability of current flight computers. It is hoped that this guidance approach will significantly reduce the development and maintenance costs for new entry guidance systems.

Table 1. Characteristics of the ERV and Trajectory Entry Conditions

Mass	186.0	slugs
Reference area	177.6	sq ft
Mean Aerodynamic Chord	25.0	ft
Altitude	400,000.0	ft
Inertial Velocity	25,778.843	ft/sec
Flight Path Angle	−0.996	deg
Inclination	28.50	deg
Latitude	−28.071	deg
Longitude	−69.313	deg
Vacuum Apogee	150.0	n.m.
Vacuum Perigee	20.0	n.m.

Table 2. Dispersions Used in Performance Study

Dispersion	Symbol	Magnitude (%)	Direction (deg)
Aerodynamics			
Lift Coefficient	CL^+	+10%	
	CL^-	−10%	
Drag Coefficient	CD^+	+10%	
	CD^-	−10%	
Vehicle Properties			
Mass	M^+	+5%	
	M^-	−5%	
Atmospheric Properties			
Density	ρ^+	+30%	
	ρ^-	−30%	
Tailwind	TW	99%	61.5 deg
Positive Crosswind	CW^+	99%	151.5 deg
Headwind	HW	99%	241.5 deg
Negative Crosswind	CW^-	99%	331.5 deg

Table 3. Dispersed Cases for Maximum Downrange Region

α_d (deg)	ϕ_d (deg)	Dispersion	Downrange (n.m.)	Crossrange (n.m.)
15	0	*NOMINAL*	14817	497
15	0	CL^+	16445	456
15	0	CL^-	13229	420
15	0	CD^+	13242	422
15	0	CD^-	16810	427
15	0	M^+	14890	498
15	0	M^-	14739	496
15	0	ρ^+	14615	494
15	0	ρ^-	15089	498
15	0	*TW*	14882	498
15	0	CW^+	14829	497
15	0	*HW*	14751	496
15	0	CW^-	14802	497
20	0	*NOMINAL*	13849	463
20	0	CL^+	15388	499
20	0	CL^-	12355	344
20	0	CD^+	12384	347
20	0	CD^-	15711	491
20	0	M^+	13909	466
20	0	M^-	13785	460
20	0	ρ^+	13659	453
20	0	ρ^-	14105	475
20	0	*TW*	13901	466
20	0	CW^+	13857	464
20	0	*HW*	13796	461
20	0	CW^-	13838	463
25	0	*NOMINAL*	12116	321
25	0	CL^+	13415	439
25	0	CL^-	10822	178
25	0	CD^+	10858	183
25	0	CD^-	13715	456
25	0	M^+	13161	325
25	0	M^-	12068	316
25	0	ρ^+	11950	305
25	0	ρ^-	12339	342
25	0	*TW*	12155	324
25	0	CW^+	12121	321
25	0	*HW*	12076	317
25	0	CW^-	12108	320

Table 4. Dispersed Cases for Maximum Crossrange Region

α_d (deg)	ϕ_d (deg)	Dispersion	Downrange (n.m.)	Crossrange (n.m.)
15	60	*NOMINAL*	8308	1822
15	60	CL^+	9035	2149
15	60	CL^-	7602	1506
15	60	CD^+	7605	1531
15	60	CD^-	9200	2193
15	60	M^+	8342	1825
15	60	M^-	8271	1819
15	60	ρ^+	8197	1823
15	60	ρ^-	8465	1820
15	60	*TW*	8260	1737
15	60	CW^+	8473	1890
15	60	*HW*	8313	1866
15	60	CW^-	7984	1692
20	60	*NOMINAL*	7791	1661
20	60	CL^+	8476	1966
20	60	CL^-	7125	1370
20	60	CD^+	7132	1392
20	60	CD^-	8622	2007
20	60	M^+	7820	1664
20	60	M^-	7761	1659
20	60	ρ^+	7689	1662
20	60	ρ^-	7934	1660
20	60	*TW*	7766	1607
20	60	CW^+	7846	1686
20	60	*HW*	7775	1673
20	60	CW^-	7580	1572
25	60	*NOMINAL*	6940	1344
25	60	CL^+	7539	1602
25	60	CL^-	6355	1100
25	60	CD^+	6363	1119
25	60	CD^-	7661	1635
25	60	M^+	6963	1345
25	60	M^-	6915	1342
25	60	ρ^+	6848	1344
25	60	ρ^-	7070	1342
25	60	*TW*	6933	1317
25	60	CW^+	6958	1355
25	60	*HW*	6921	1347
25	60	CW^-	6819	1290

Table 5. Dispersed Cases for Minimum Downrange Region

α_d (deg)	ϕ_d (deg)	Dispersion	Downrange (n.m.)	Crossrange (n.m.)
30	90	*NOMINAL*	2982	938
30	90	CL^+	3014	1102
30	90	CL^-	2934	778
30	90	CD^+	2914	793
30	90	CD^-	3045	1119
30	90	M^+	2996	937
30	90	M^-	2969	938
30	90	ρ^+	2914	943
30	90	ρ^-	3078	930
30	90	TW	2993	914
30	90	CW^+	2992	941
30	90	HW	2996	939
30	90	CW^-	2982	907
30	80	*NOMINAL*	3851	1041
30	80	CL^+	4011	1233
30	80	CL^-	3680	858
30	80	CD^+	3668	875
30	80	CD^-	4060	1254
30	80	M^+	3865	1041
30	80	M^-	3865	1041
30	80	ρ^+	3778	1046
30	80	ρ^-	3954	1034
30	80	TW	3856	1020
30	80	CW^+	3841	1044
30	80	HW	3834	1042
30	80	CW^-	3828	1007
30	70	*NOMINAL*	4932	1075
30	70	CL^+	5262	1279
30	70	CL^-	4602	881
30	70	CD^+	4600	898
30	70	CD^-	5337	1304
30	70	M^+	4949	1075
30	70	M^-	4915	1075
30	70	ρ^+	4854	1078
30	70	ρ^-	5043	1069
30	70	TW	4934	1057
30	70	CW^+	4925	1077
30	70	HW	4914	1074
30	70	CW^-	4884	1040

Table 6. Maximum Downrange Region Closed-Loop Results

Dispersions					Final State		
CL (%)	CD (%)	Mass (%)	ρ (%)	Wind (%)	ϕ_g (deg)	λ (deg)	Error (n.m.)
TARGET POINT					+ 9.800	+144.700	-
NOMINAL					+ 9.799	+144.702	0.13
- 1.9	+ 1.9	- 3.2	+19.1	63.8 HW	+ 9.799	+144.701	0.08
- 1.9	+ 1.9	- 3.2	STS 1	63.8 HW	+ 9.803	+144.694	0.30
- 1.9	+ 1.9	- 3.2	STS 9	63.8 HW	+ 9.802	+144.696	0.27
- 1.9	+ 1.9	- 3.2	STS11	63.8 HW	+ 9.801	+144.698	0.13

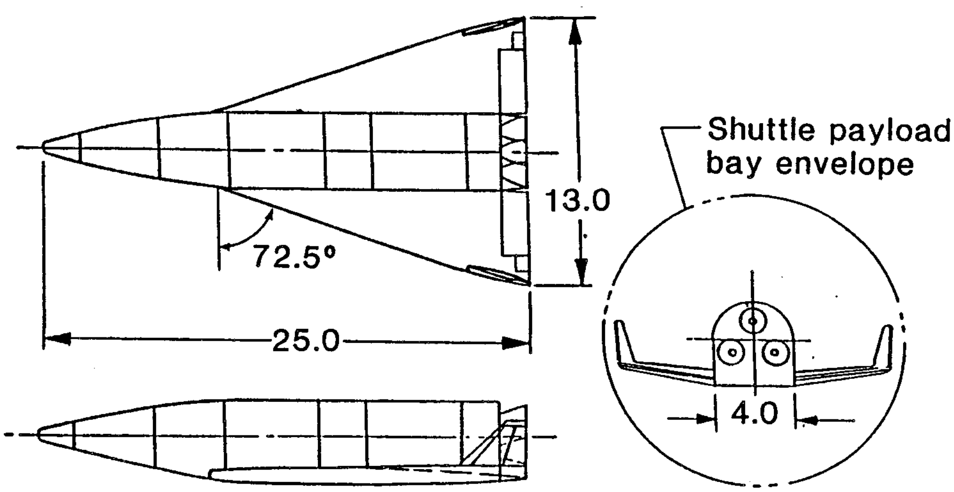
Table 7. Maximum Crossrange Region Closed-Loop Results

Dispersions					Final State		
CL (%)	CD (%)	Mass (%)	ρ (%)	Wind (%)	ϕ_g (deg)	λ (deg)	Error (n.m.)
TARGET POINT					- 2.300	+ 55.000	-
NOMINAL					- 2.301	+ 55.000	0.06
- 1.9	+ 1.9	- 3.2	+19.1	63.8 CW^-	- 2.295	+ 54.998	0.32
- 1.9	+ 1.9	- 3.2	STS 1	63.8 CW^-	- 2.304	+ 55.001	0.25
- 1.9	+ 1.9	- 3.2	STS 9	63.8 CW^-	- 2.301	+ 54.990	0.60
- 1.9	+ 1.9	- 3.2	STS11	63.8 CW^-	- 2.299	+ 55.000	0.06

Table 8. Minimum Downrange Region Closed-Loop Results

Dispersions					Final State		
CL (%)	CD (%)	Mass (%)	ρ (%)	Wind (%)	ϕ_g (deg)	λ (deg)	Error (n.m.)
TARGET POINT					-14.900	+ 9.800	-
NOMINAL					-14.897	+ 9.798	0.22
+ 1.9	- 1.9	+ 3.2	-19.1	63.8 TW	-14.902	+ 9.801	0.13
+ 1.9	- 1.9	+ 3.2	STS 1	63.8 TW	-14.901	+ 9.800	0.06
+ 1.9	- 1.9	+ 3.2	STS 9	63.8 TW	-14.900	+ 9.800	0.00
+ 1.9	- 1.9	+ 3.2	STS11	63.8 TW	-14.899	+ 9.799	0.08

Theoretical wing area = 177.40 ft^2
Body flap area = 7.50 ft^2
Elevon area per side = 6.07 ft^2
Aileron area per side = 0.88 ft^2
Tip fin area per side = 4.16 ft^2

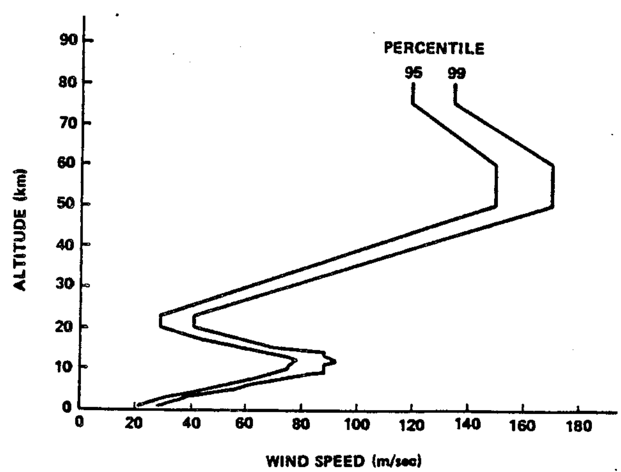


Reproduced from Reference (14)

Figure 1. Three-View Drawing of the ERV

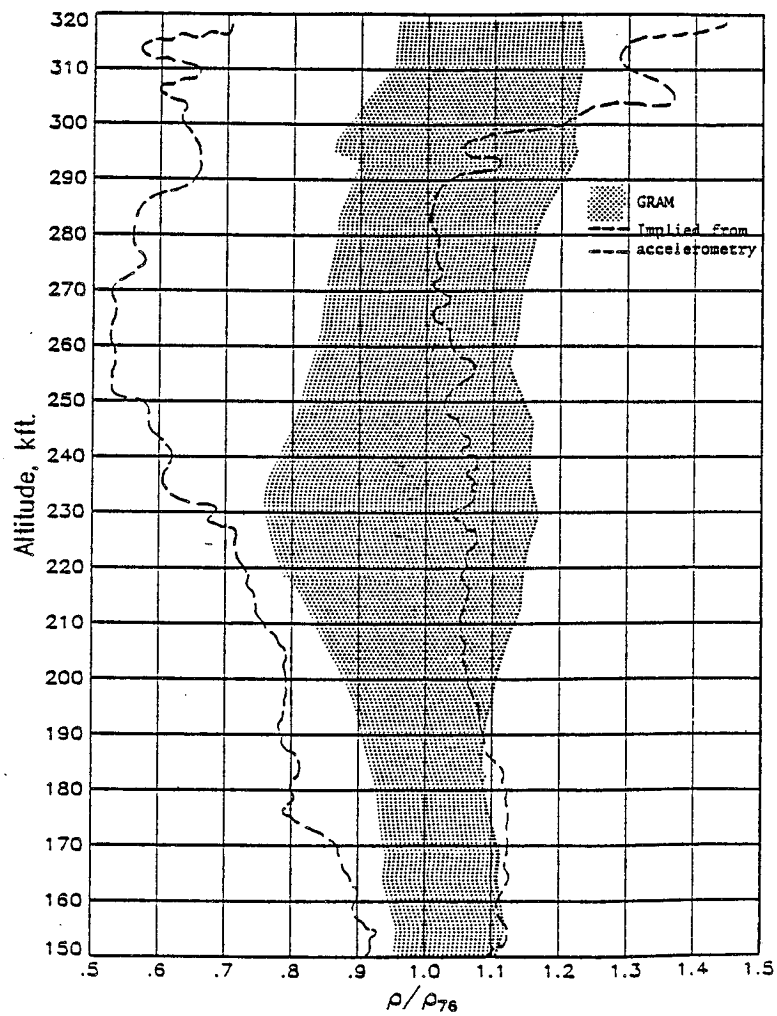
SCALAR WIND SPEED V (m/sec) STEADY-STATE ENVELOPES AS FUNCTIONS OF ALTITUDE H (km) FOR TWO PROBABILITIES P (%) ENCOMPASSING ALL FOUR LOCATIONS

P = 95				P = 99			
H	V	H	V	H	V	H	V
1	22	17	44	1	28	15	70
3	31	20	29	3	38	20	41
		23	29	5	56	23	41
6	54	50	150	6	60	50	170
		60	150	7	68	60	170
10	75	75	120	9	88	75	135
11	76	80	120	11	88	80	135
12	78			12	92		
13	74			13	88		
				14	88		



Reproduced from Reference (8)

Figure 2. Atmospheric Wind Profile



Reproduced from Reference (10)

Figure 3. Envelope of Density Profiles Derived from Shuttle Flights

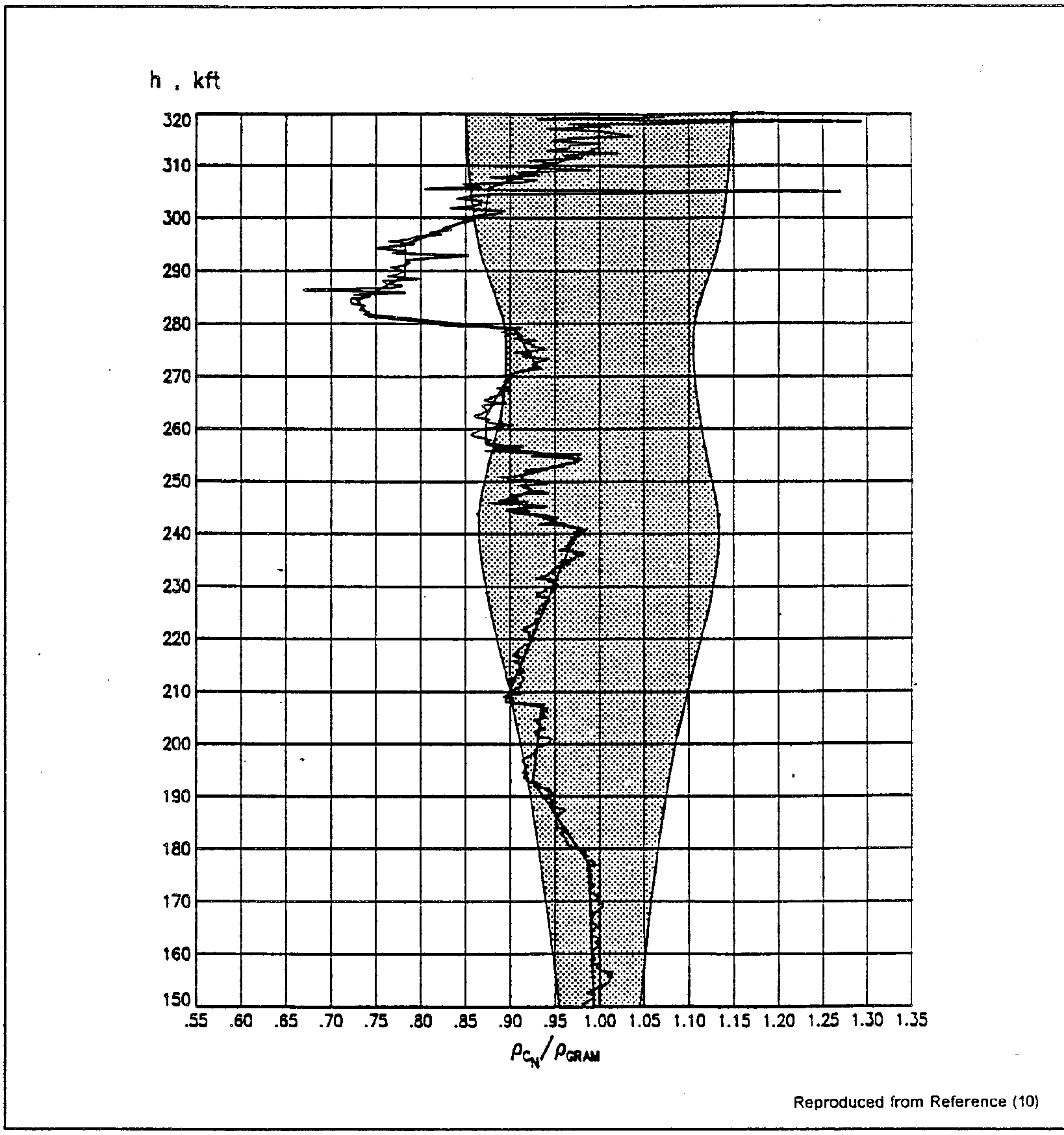


Figure 4. STS-1 Density Profile Comparison

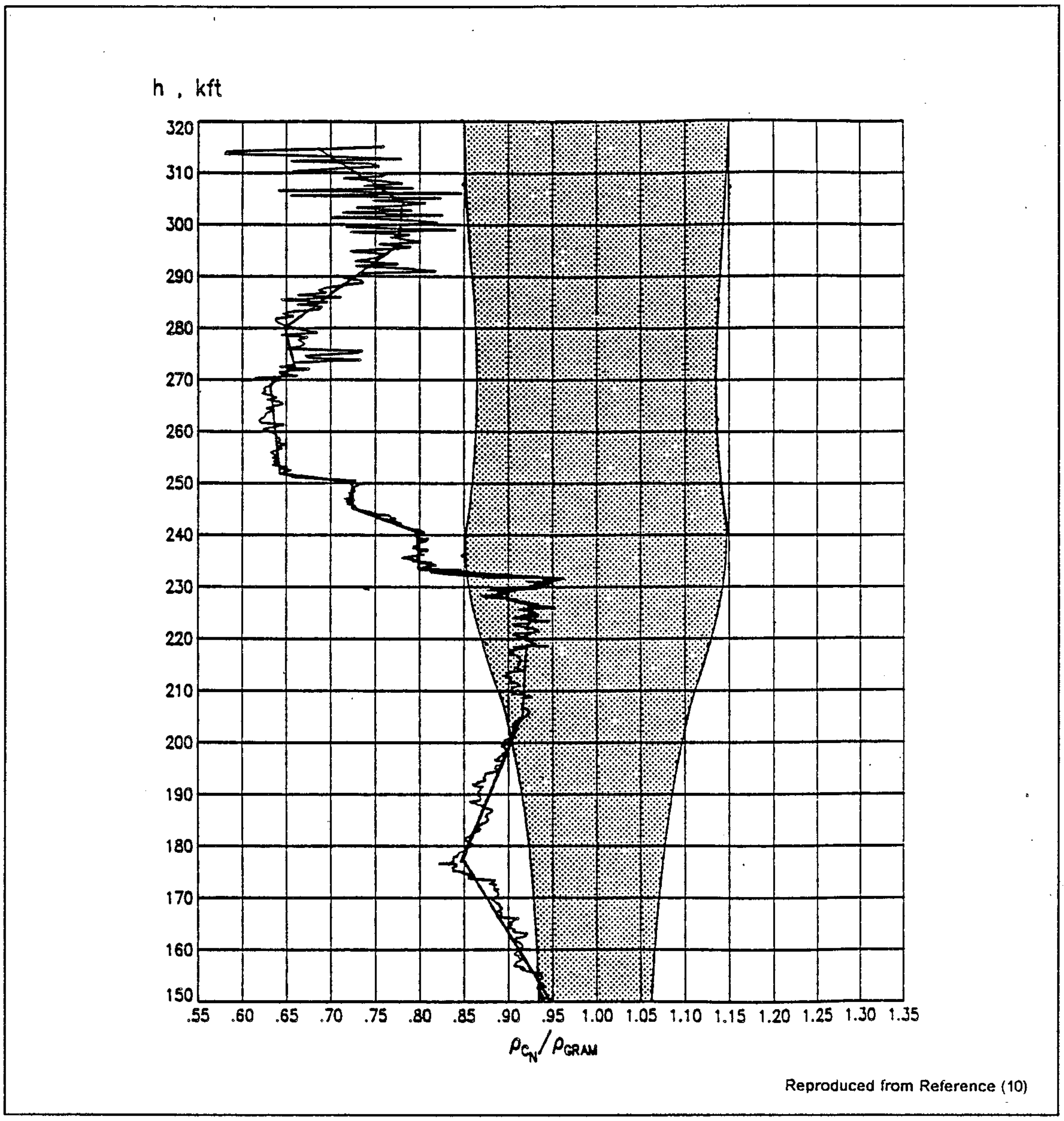


Figure 5. STS-9 Density Profile Comparison

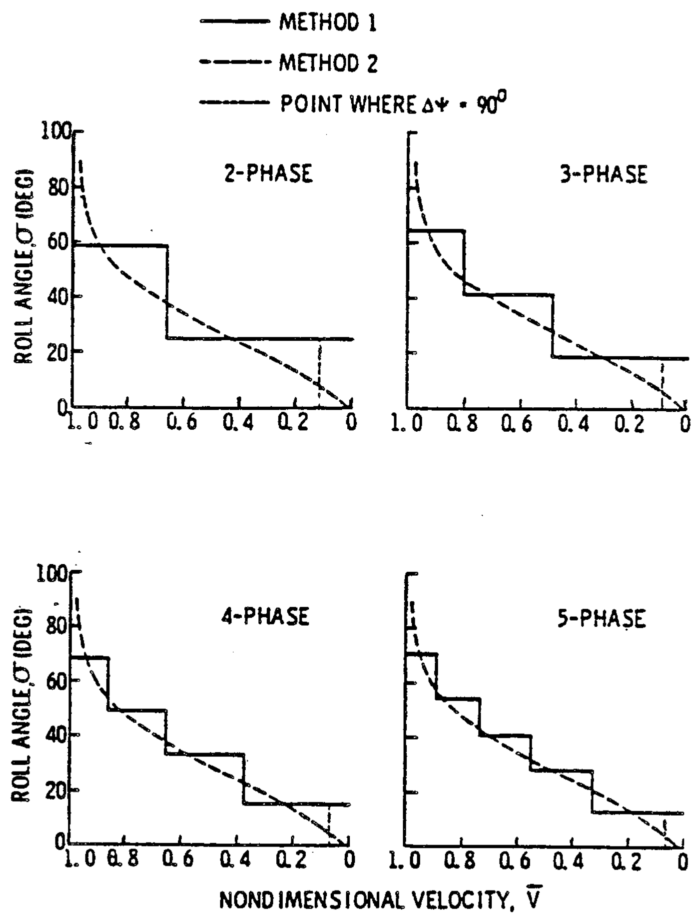


Figure 6. Multiphase Bank Angle Program for L/D = 1.5

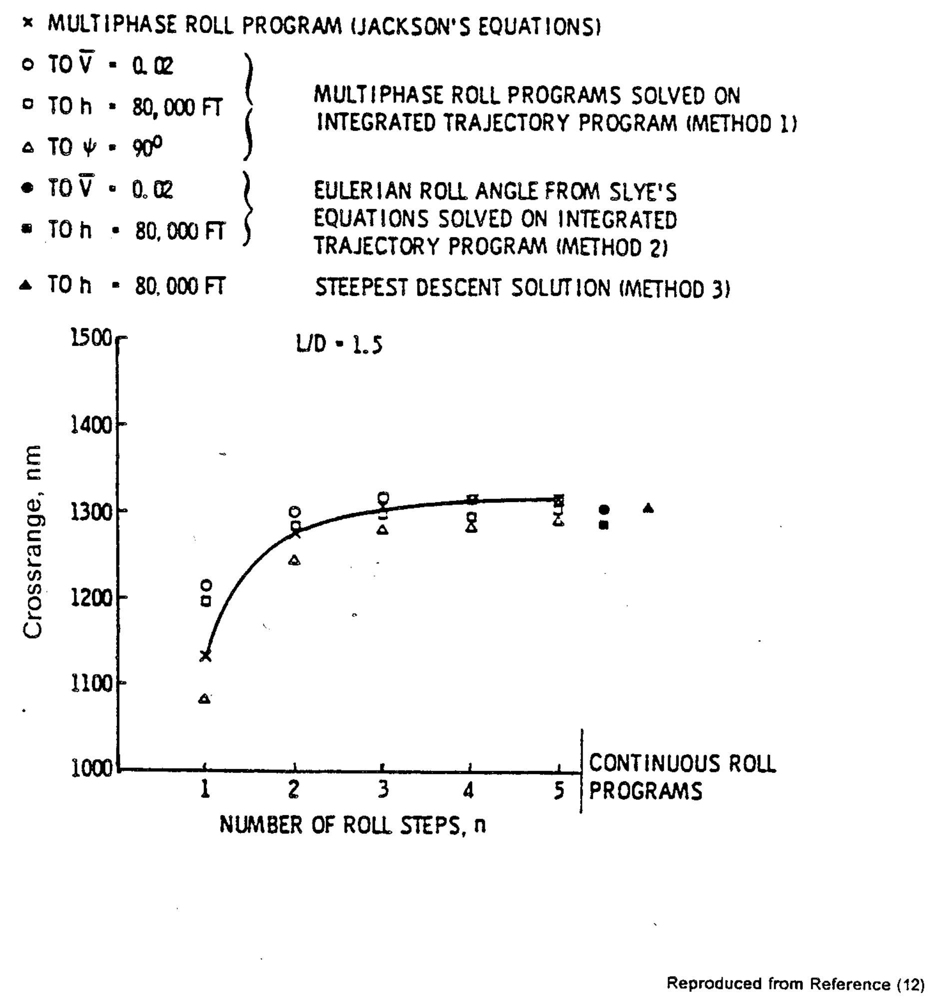


Figure 7. Crossrange Versus Number of Bank Steps

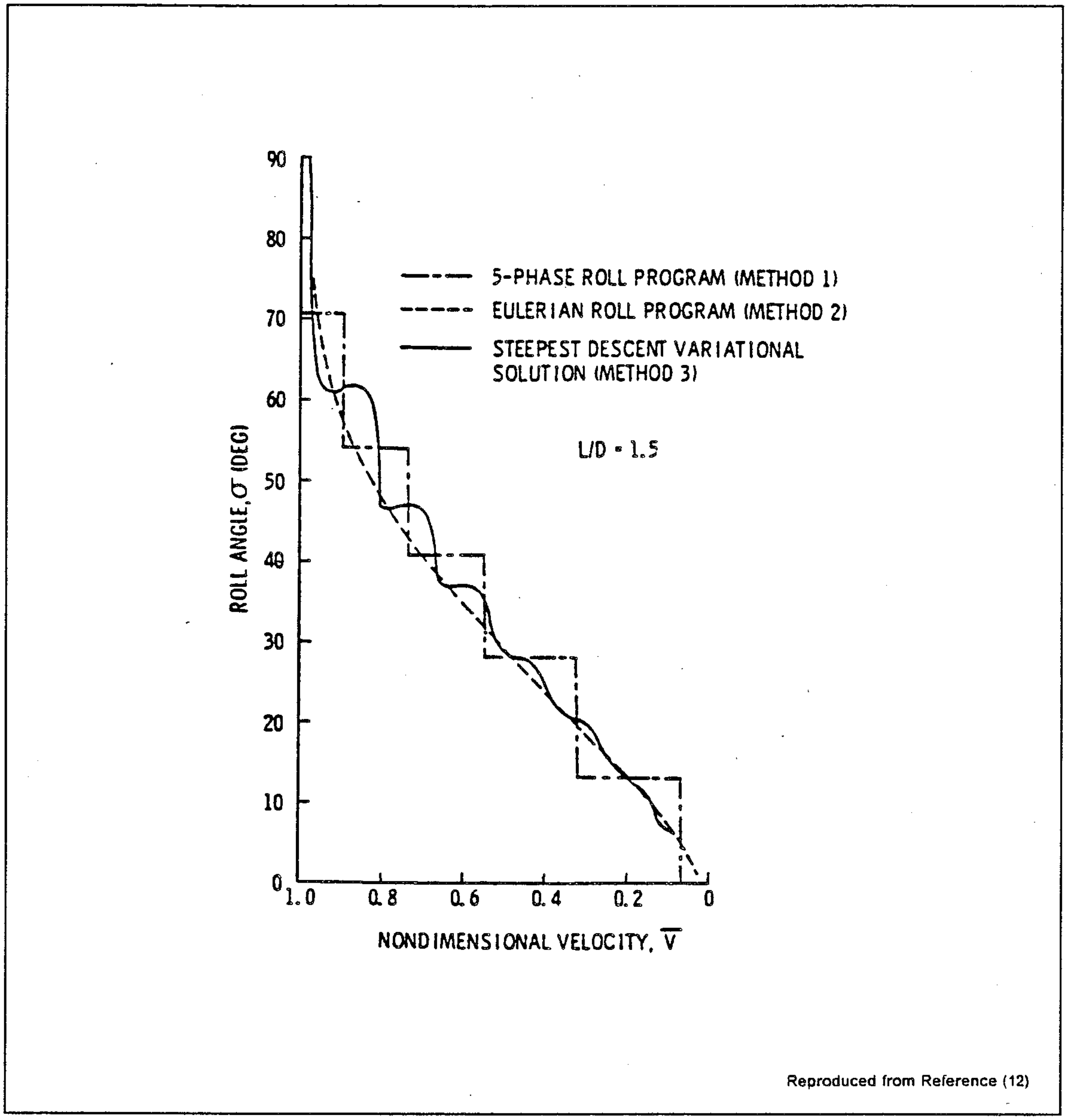


Figure 8. Comparison of Optimum Bank Angle Programs

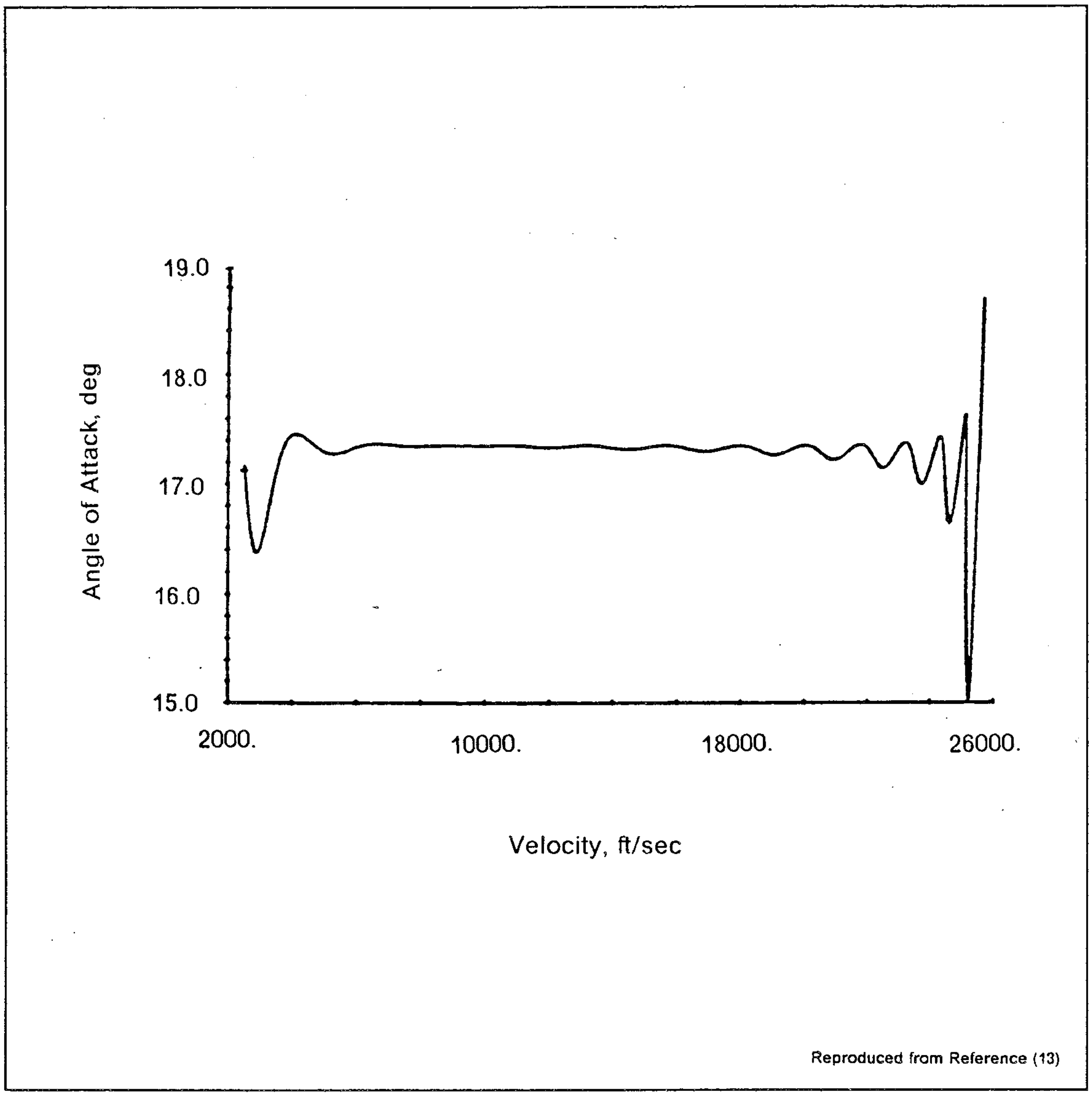
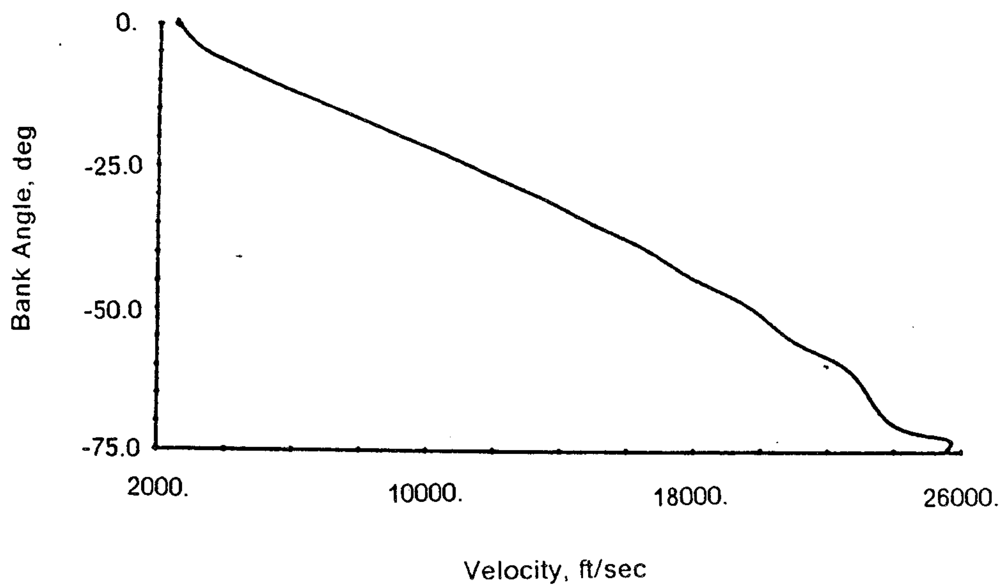


Figure 9. Optimum Shuttle Angle of Attack Profile for Maximum Downrange



Reproduced from Reference (13)

Figure 10. Optimum Shuttle Bank Angle Profile for Maximum Crossrange

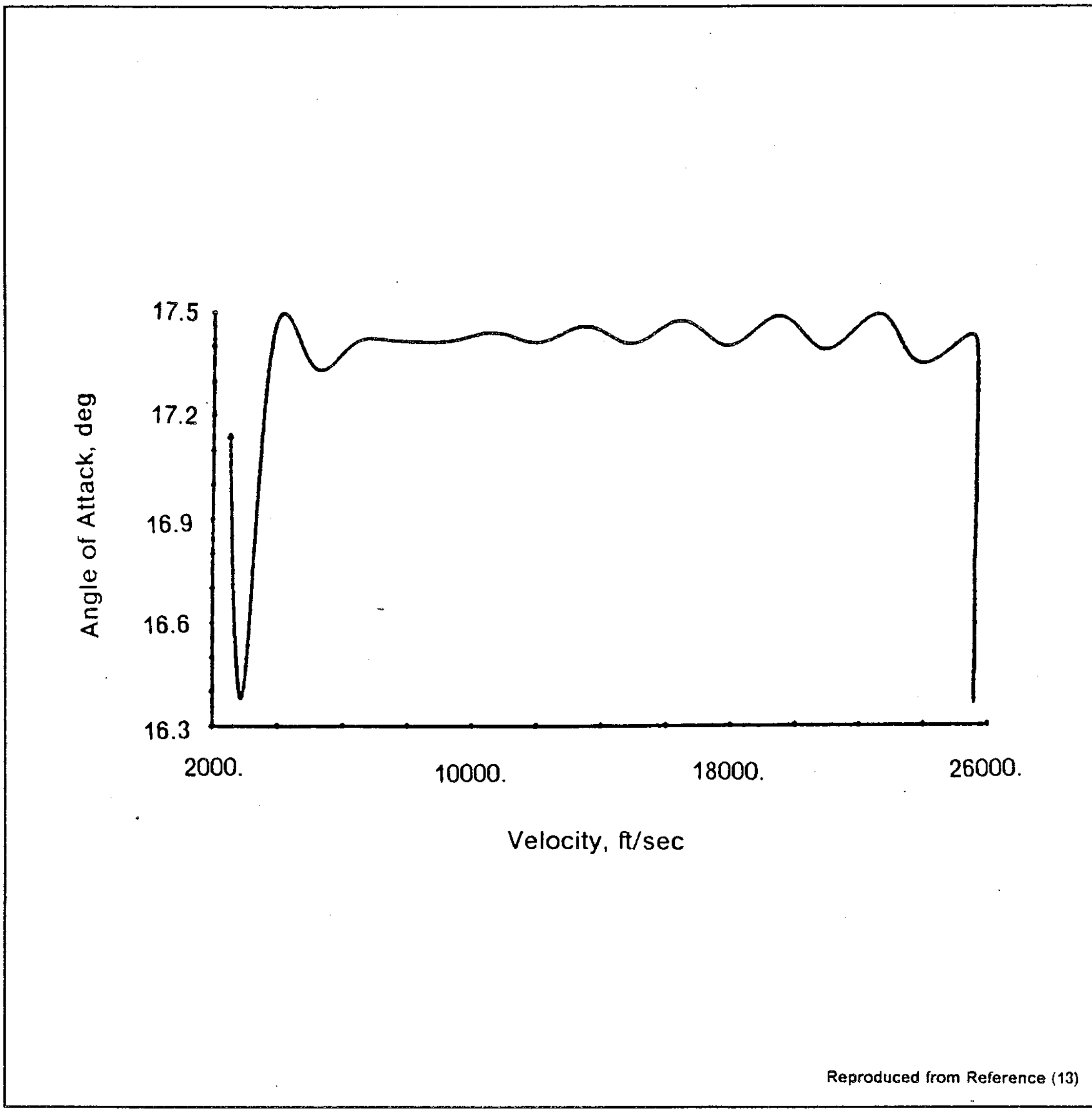
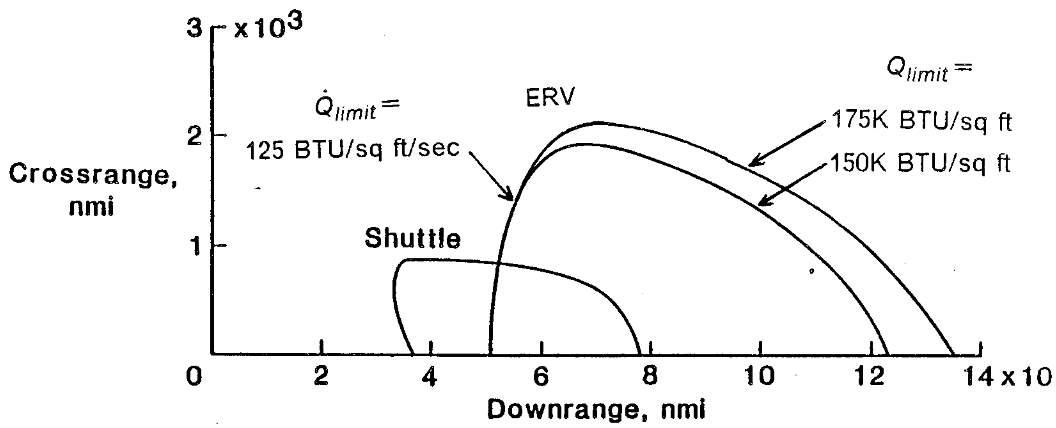


Figure 11. Optimum Shuttle Angle of Attack Profile for Maximum Crossrange



Reproduced from Reference (14)

Figure 12. Landing Footprint for the ERV

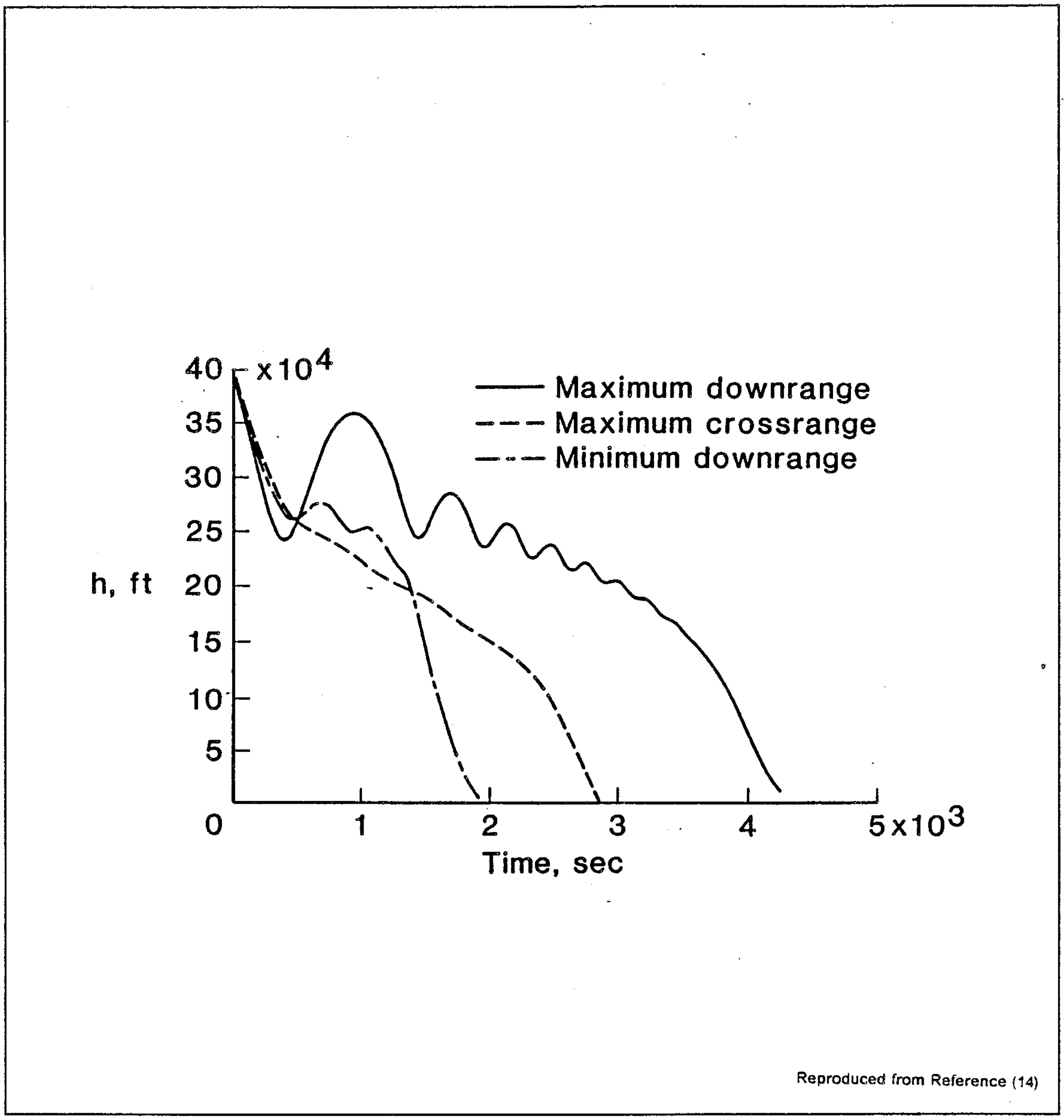


Figure 13. Altitude Histories for the Entry Missions of the ERV

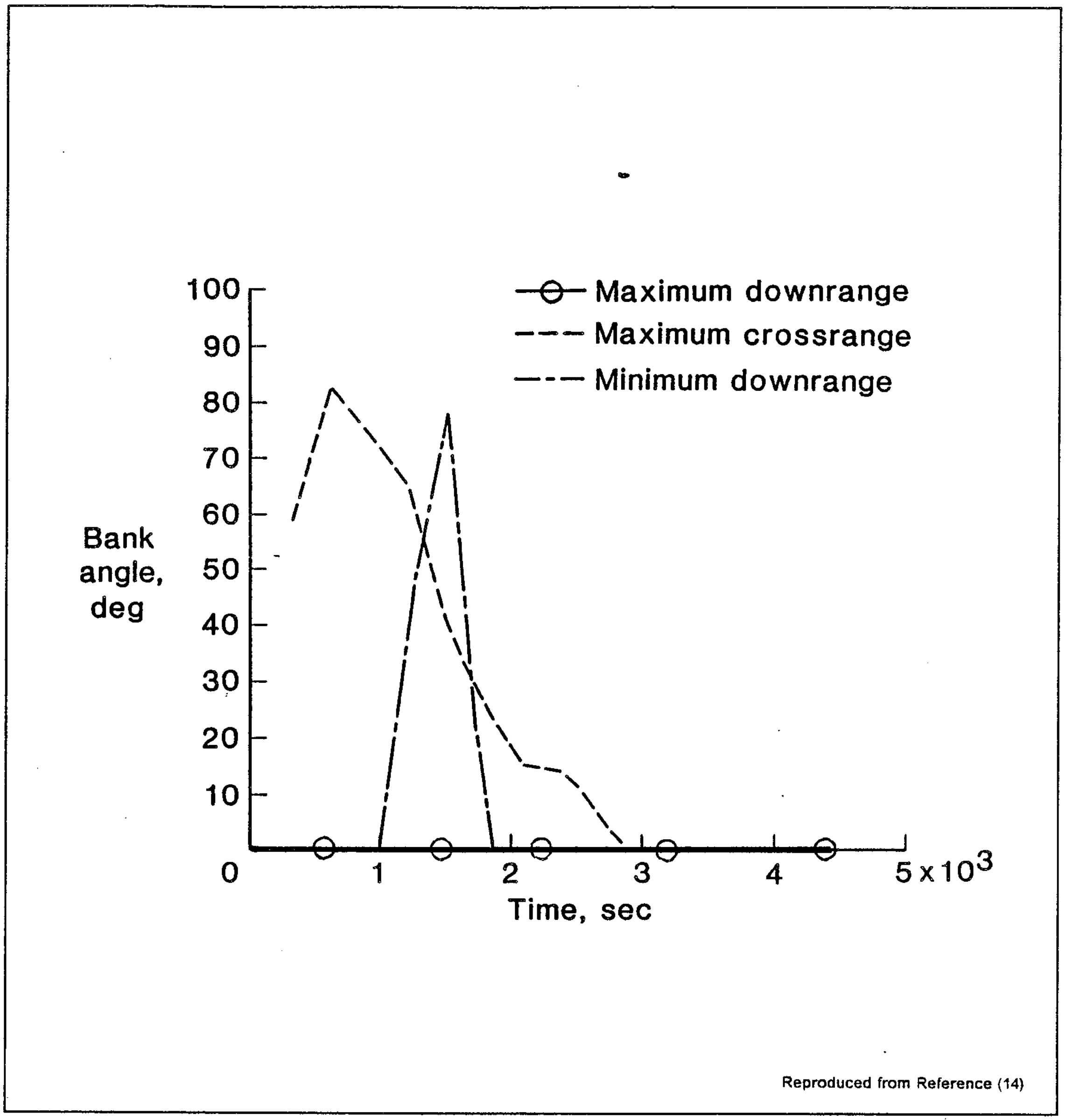
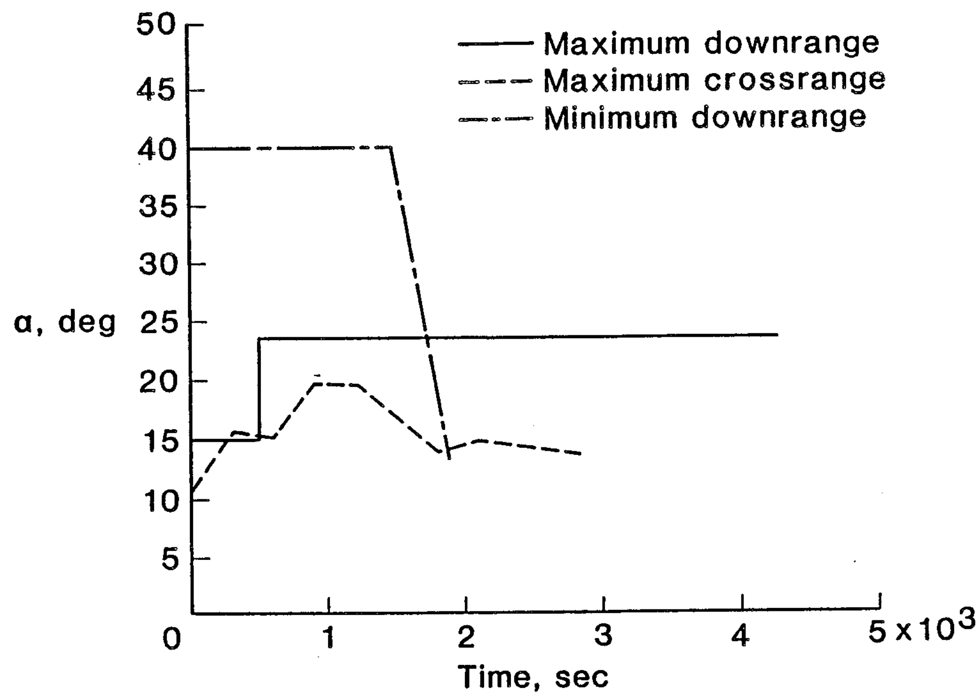


Figure 14. Bank Angle Histories for the Entry Missions of the ERV



Reproduced from Reference (14)

Figure 15. Angle of Attack Histories for the Entry Missions of the ERV

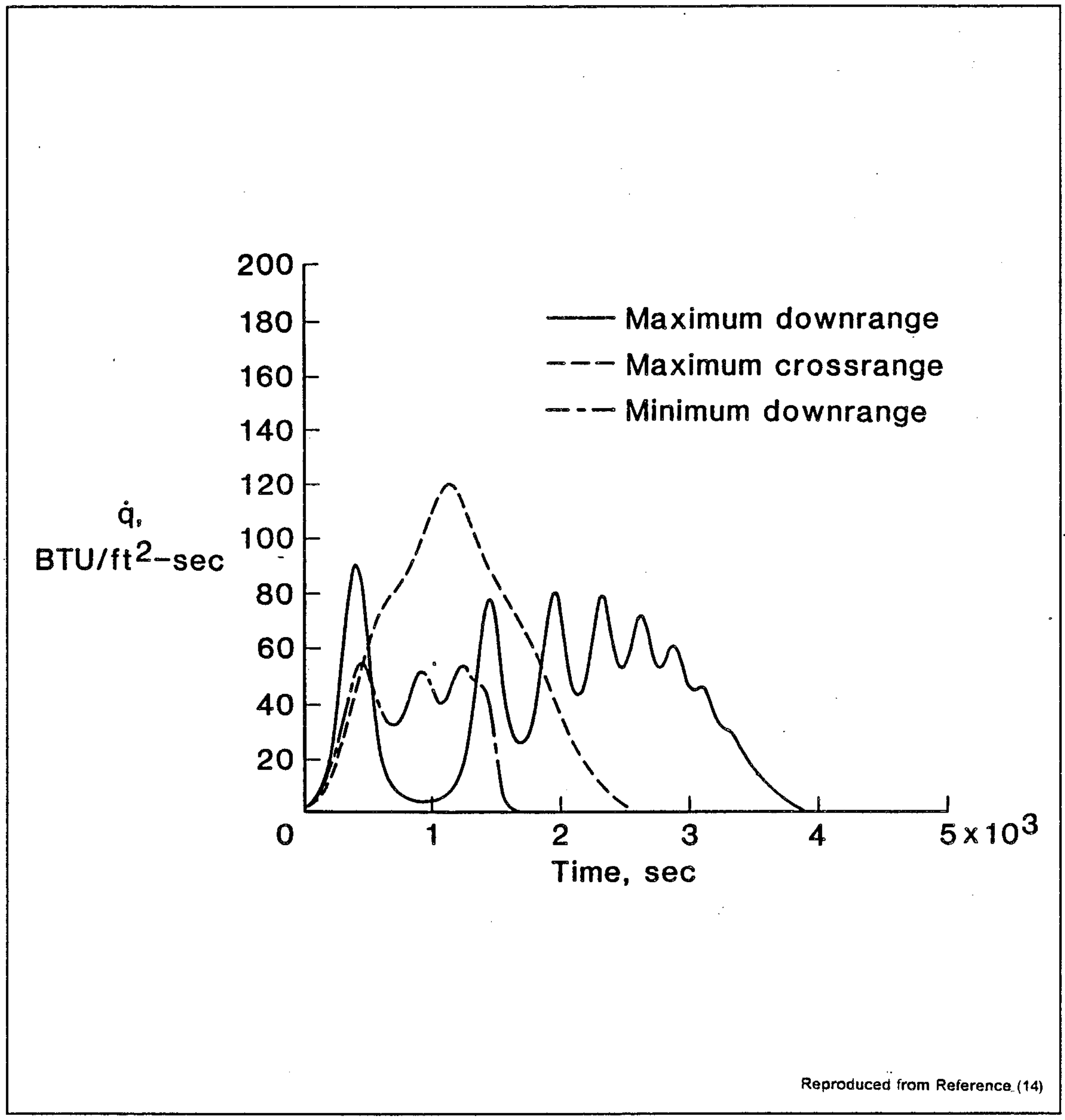


Figure 16. Heat Rate Histories for the Entry Missions of the ERV

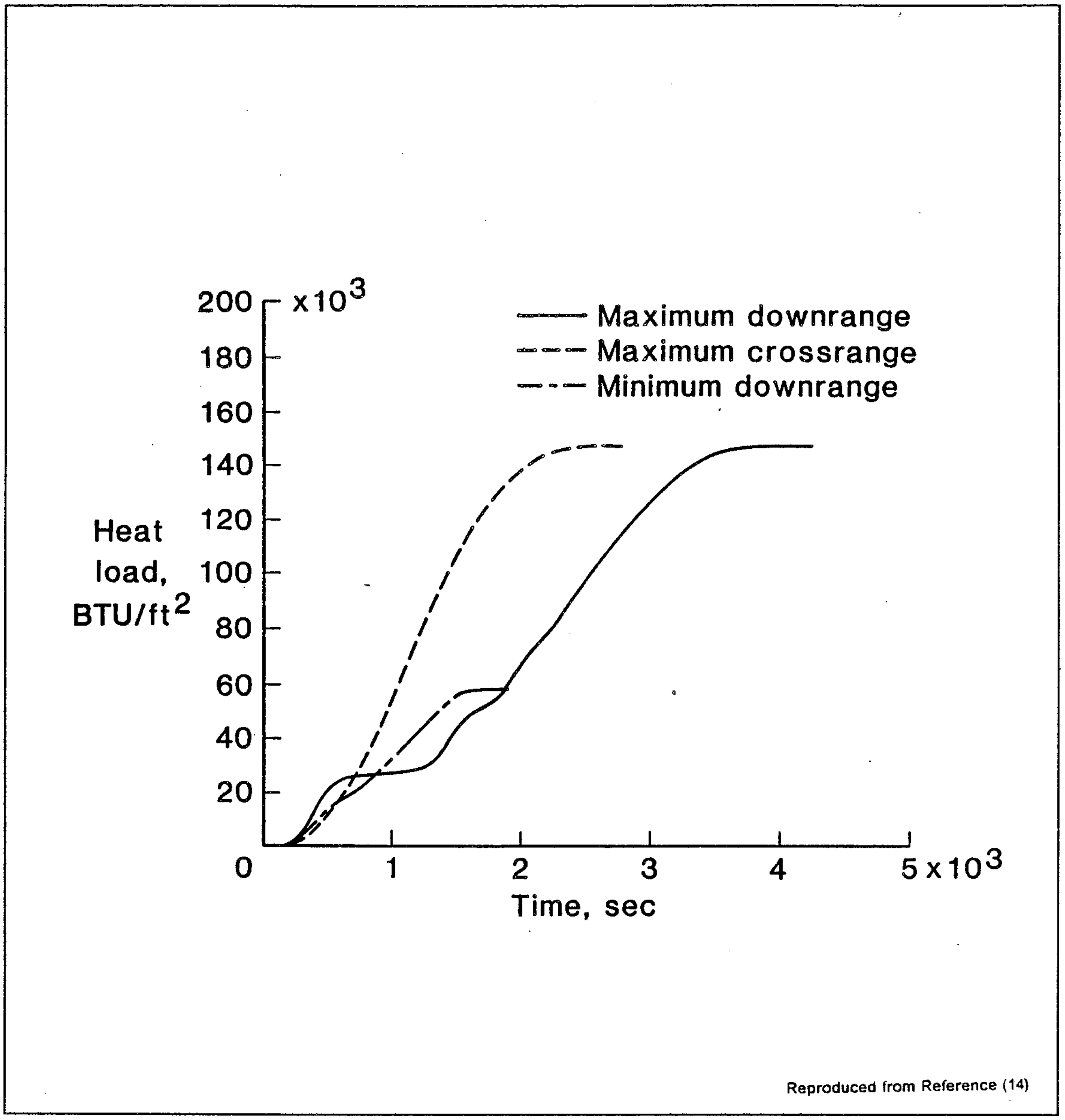


Figure 17. Heat Load Histories for the Entry Missions of the ERV

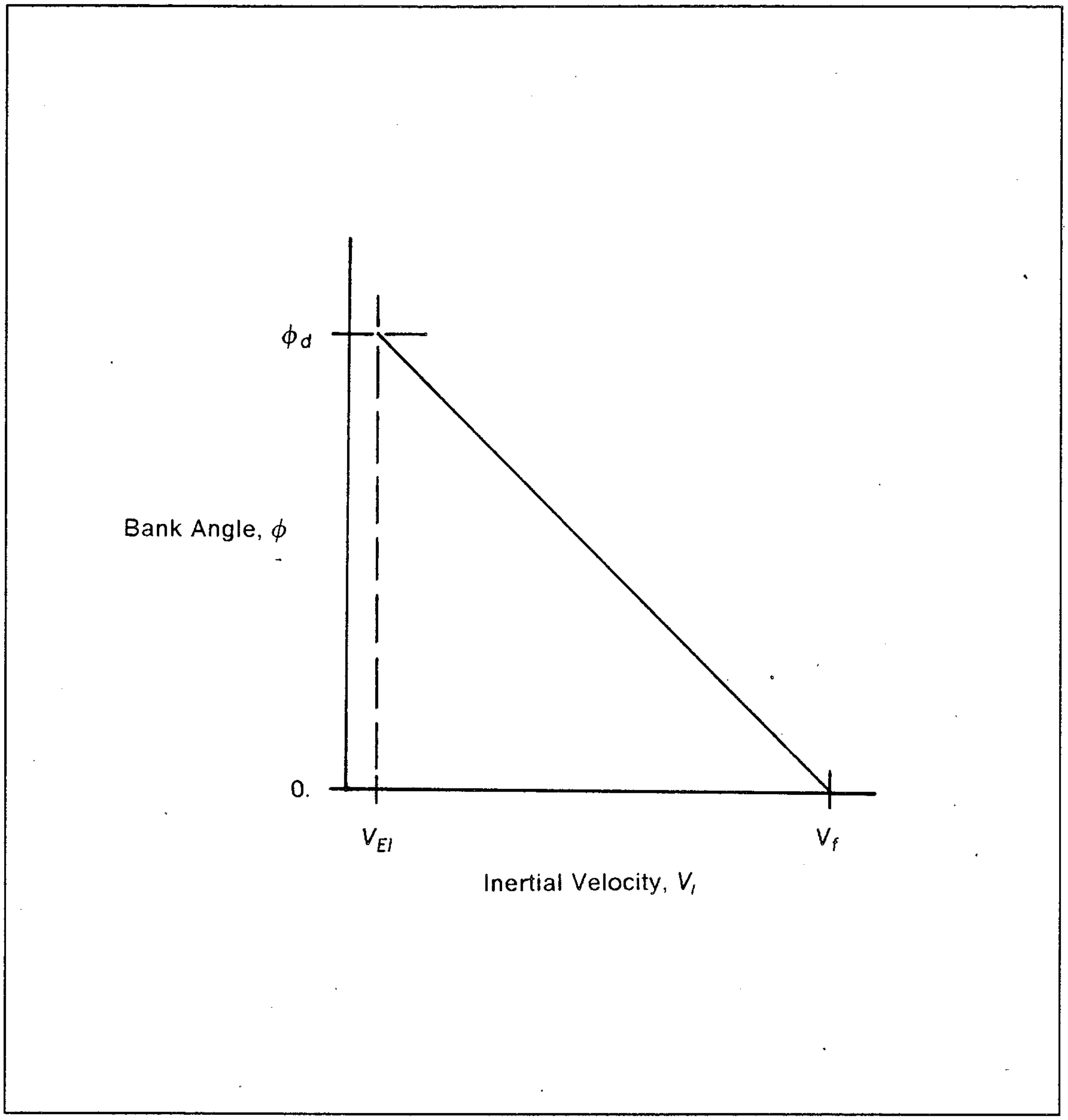


Figure 18. Bank Angle Versus Velocity Profile

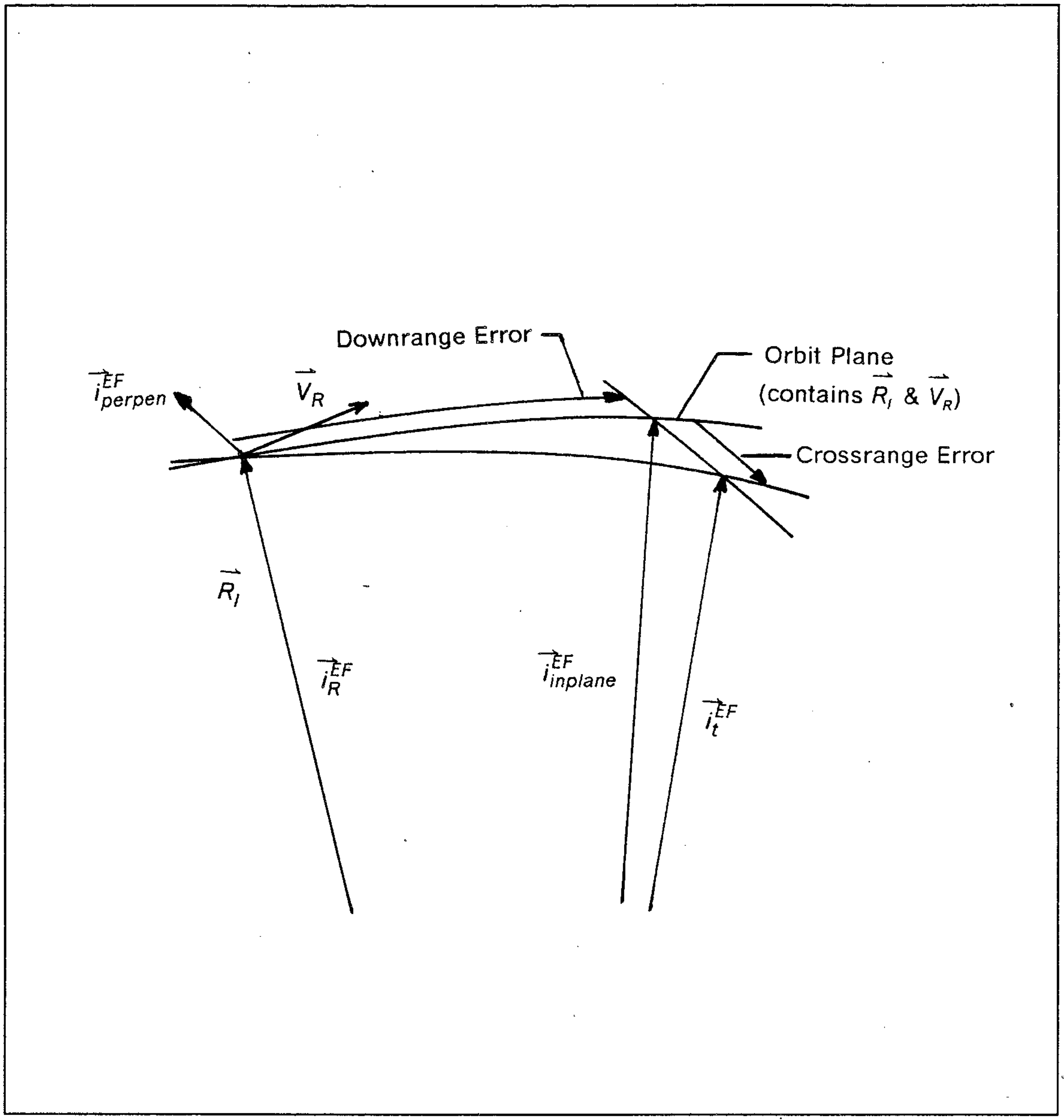


Figure 19. Definitions of Downrange and Crossrange Errors

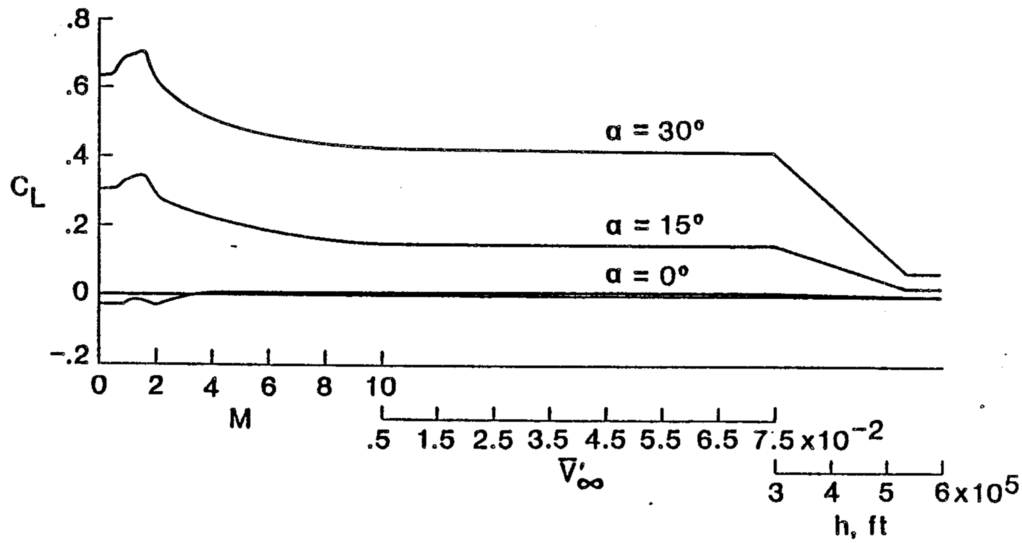
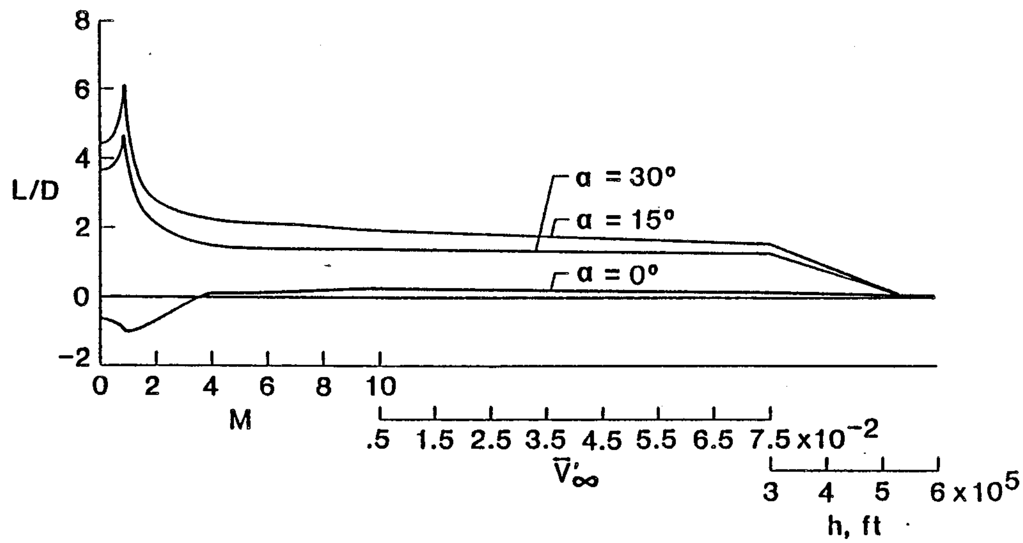


Figure 20. Predicted Lift Coefficient Profile for the ERV



Reproduced from Reference (14)

Figure 21. Predicted L/D Profile for the ERV

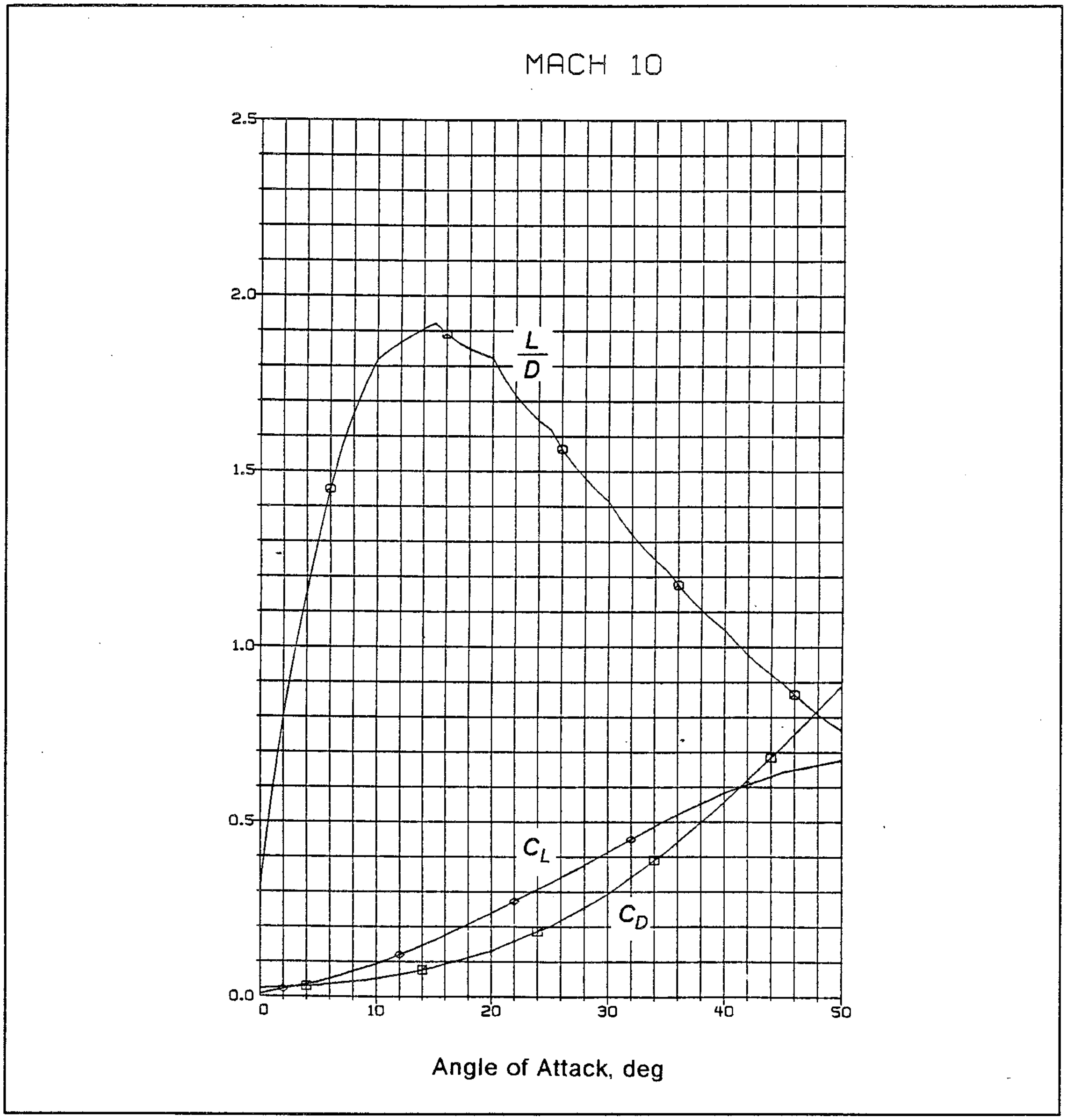


Figure 22. Predicted L/D versus Angle of Attack Profile for the ERV

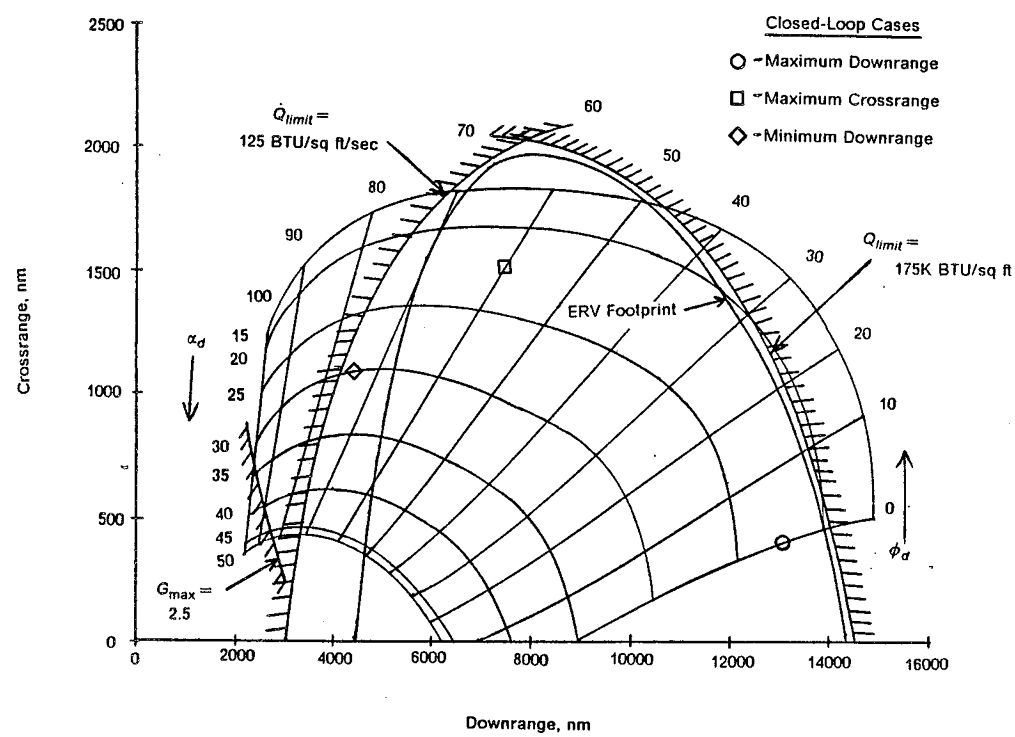


Figure 23. ERV Open-Loop Footprint with the Control Profile

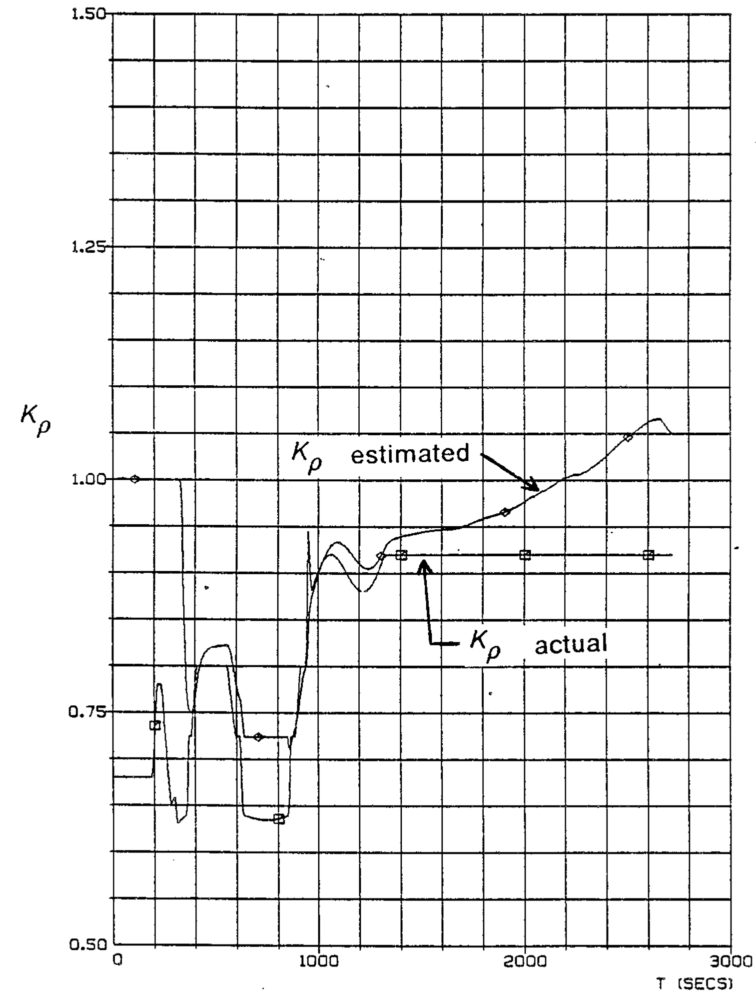


Figure 24. Time Response of the Density Filter

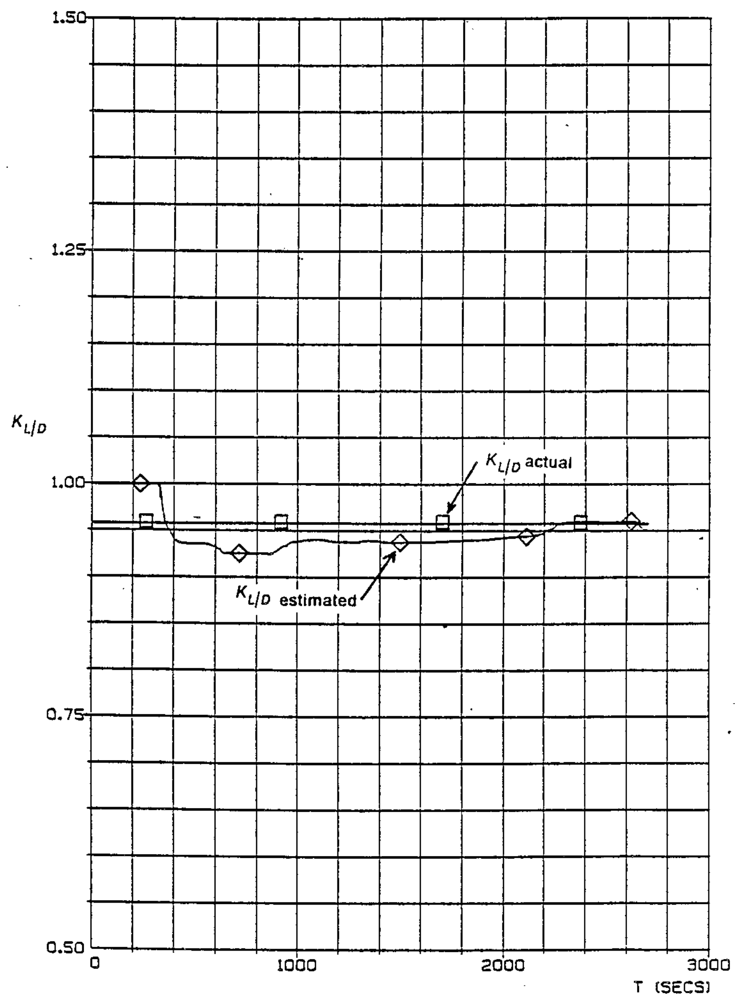


Figure 25. Time Response of the L/D Filter

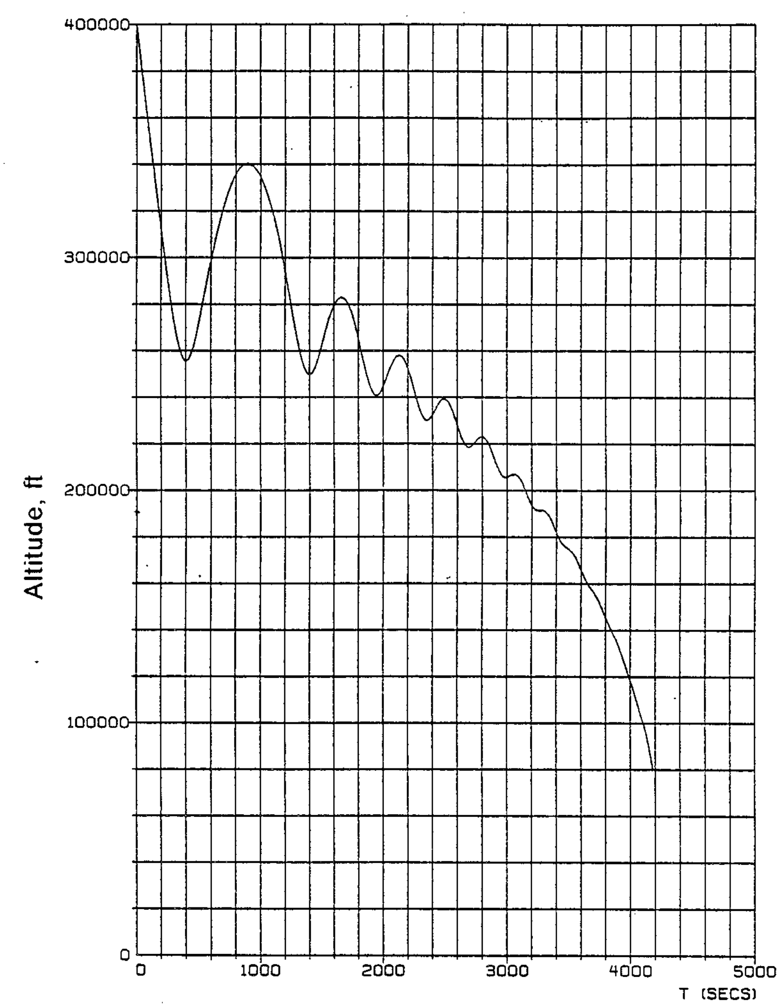


Figure 26. Closed-Loop Altitude History for the Maximum Downrange Case

C-2

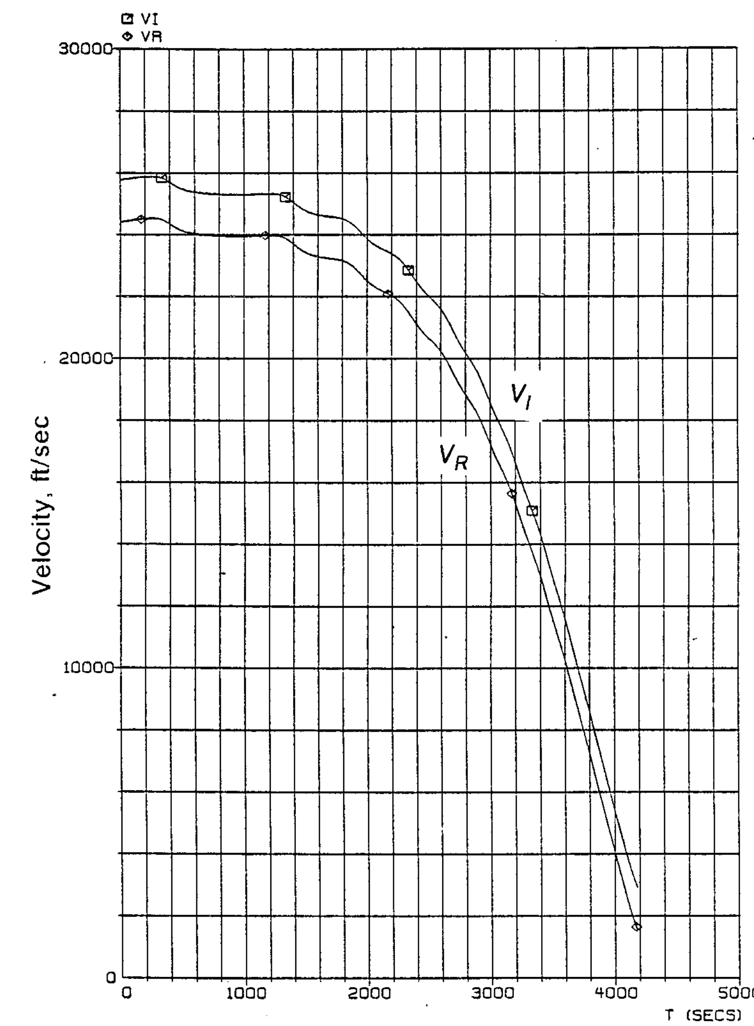


Figure 27. Closed-Loop Velocity History for the Maximum Downrange Case

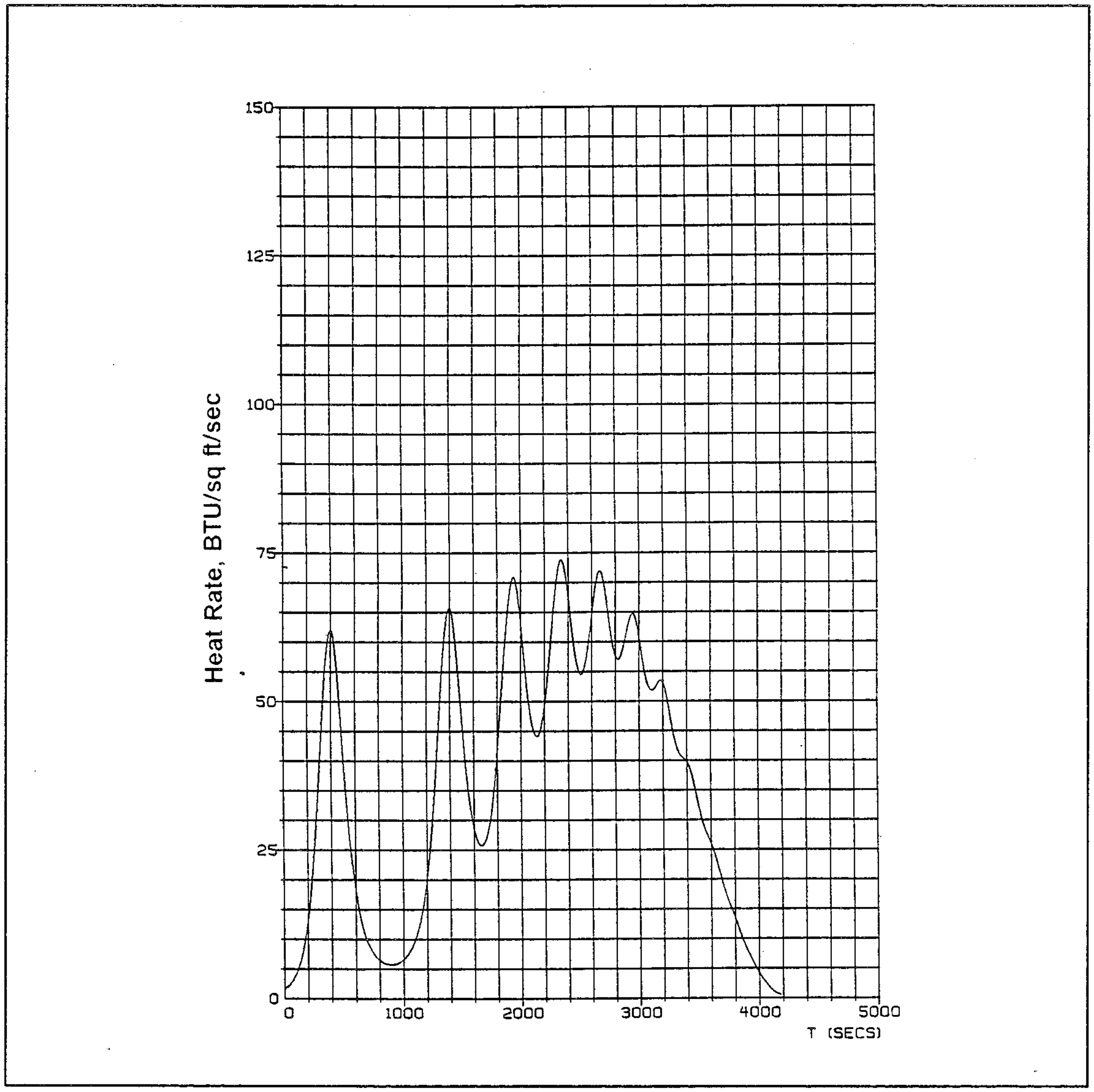


Figure 28. Closed-Loop Heat Rate History for the Maximum Downrange Case

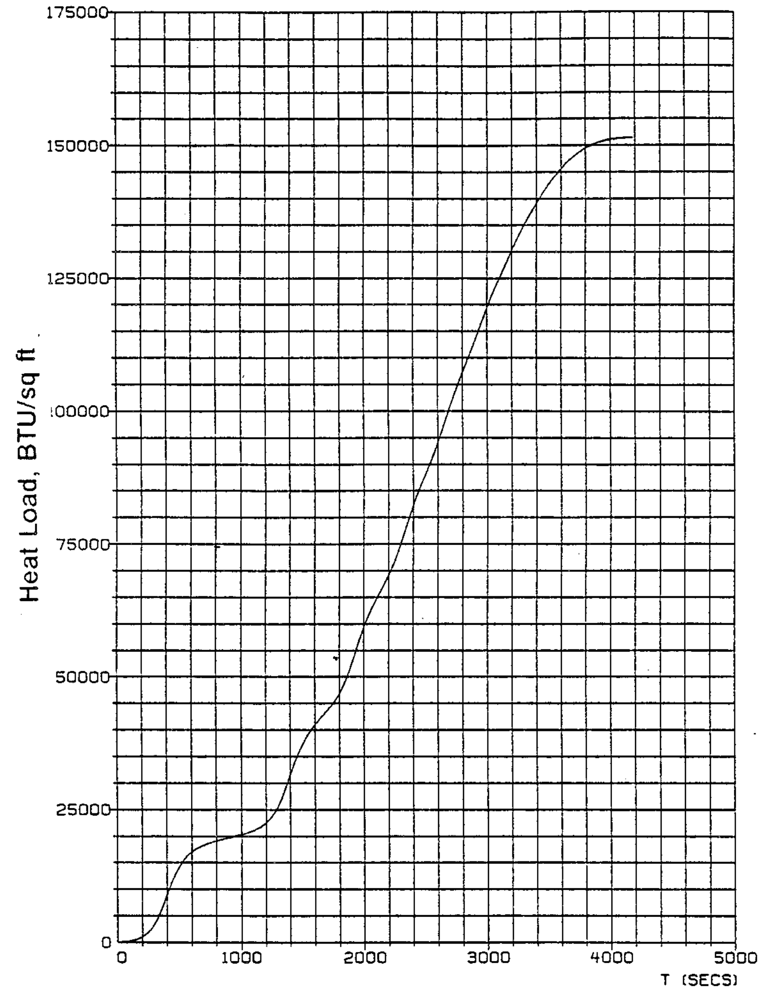


Figure 29. Closed-Loop Heat Load History for the Maximum Downrange Case

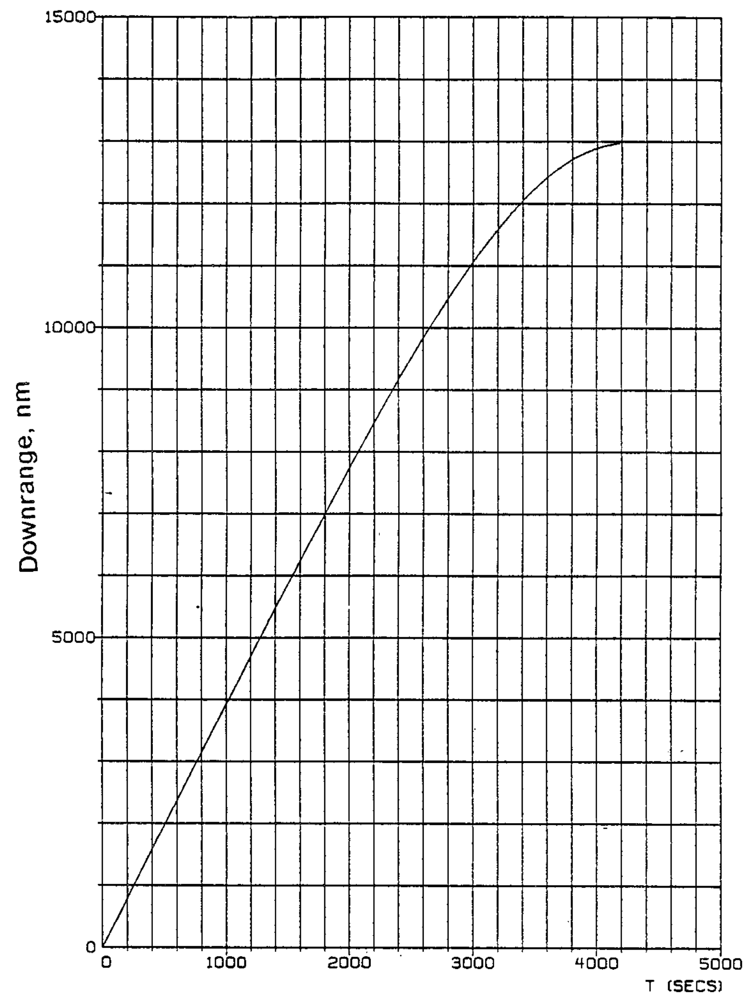


Figure 30. Closed-Loop Downrange History for the Maximum Downrange Case

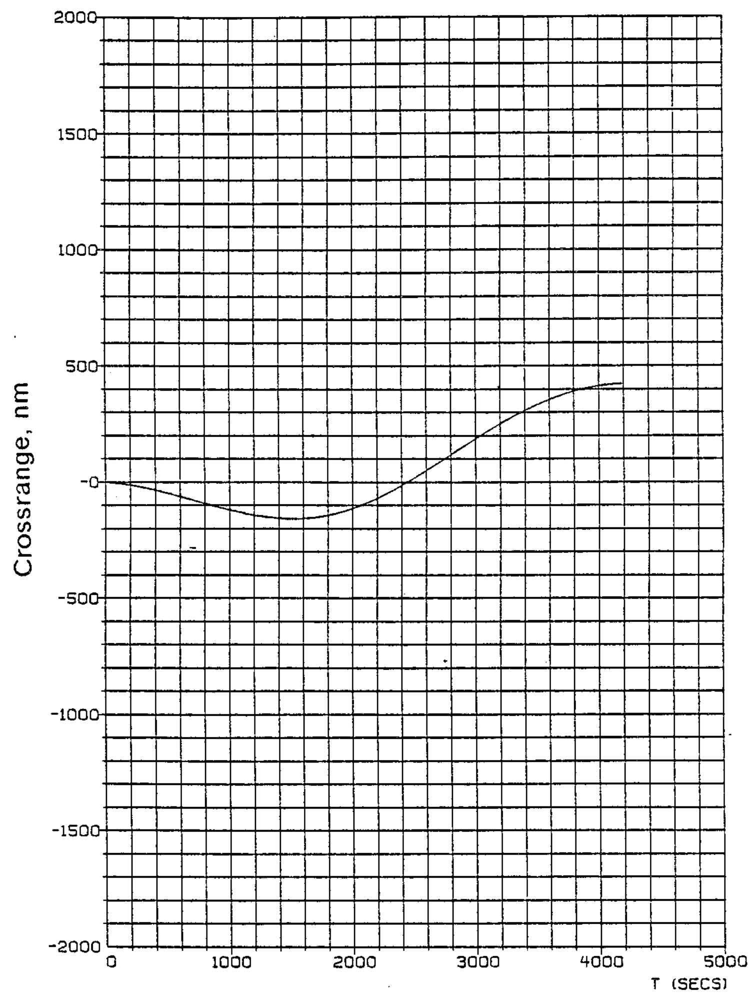


Figure 31. Closed-Loop Crossrange History for the Maximum Downrange Case

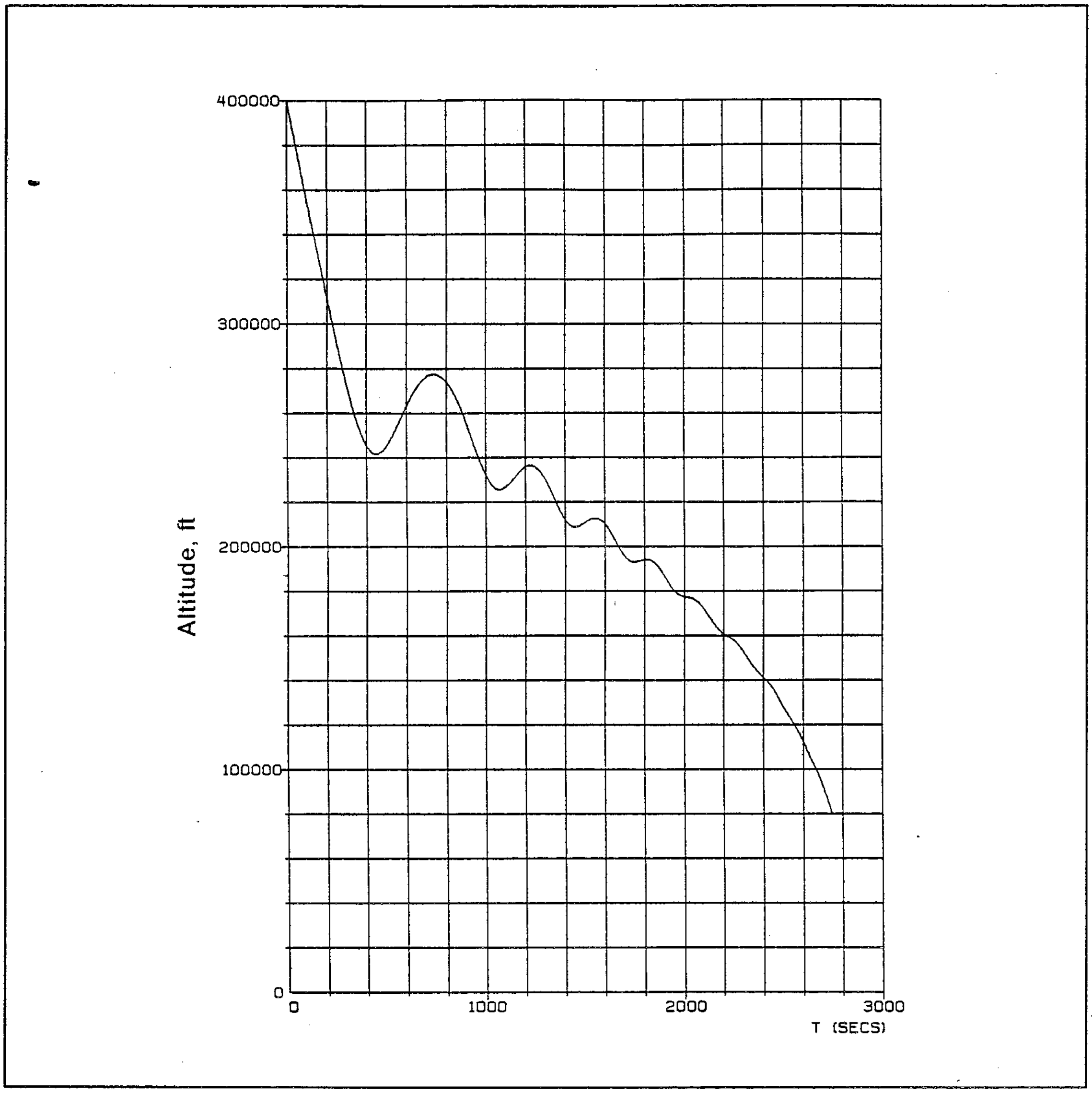


Figure 32. Closed-Loop Altitude History for the Maximum Crossrange Case

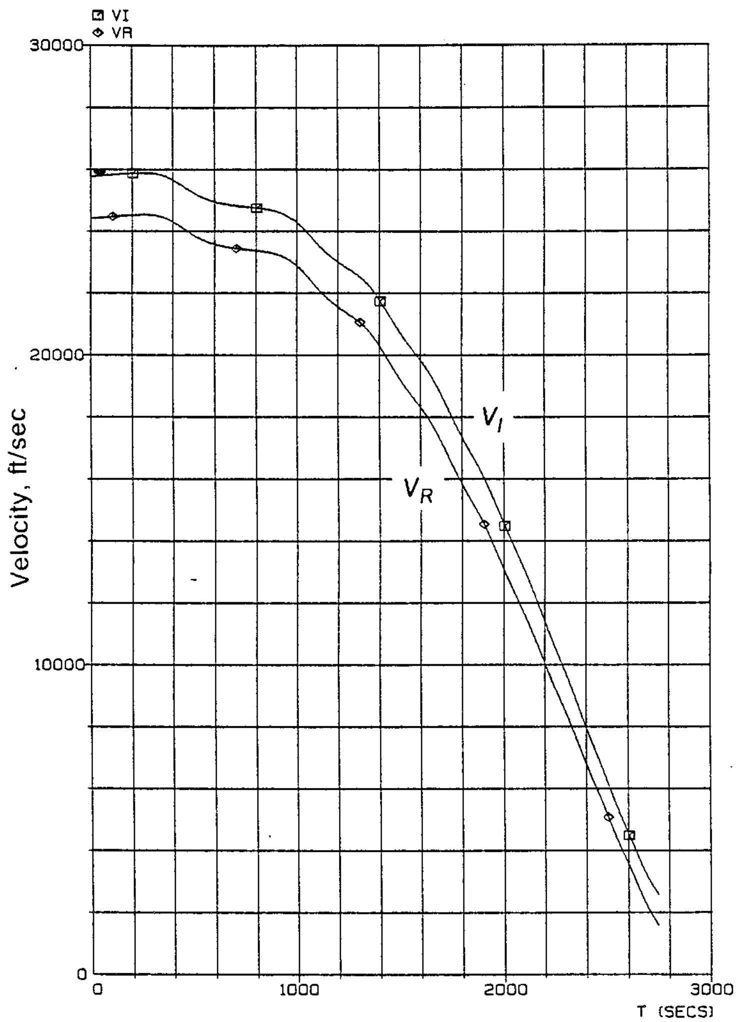


Figure 33. Closed-Loop Velocity History for the Maximum Crossrange Case

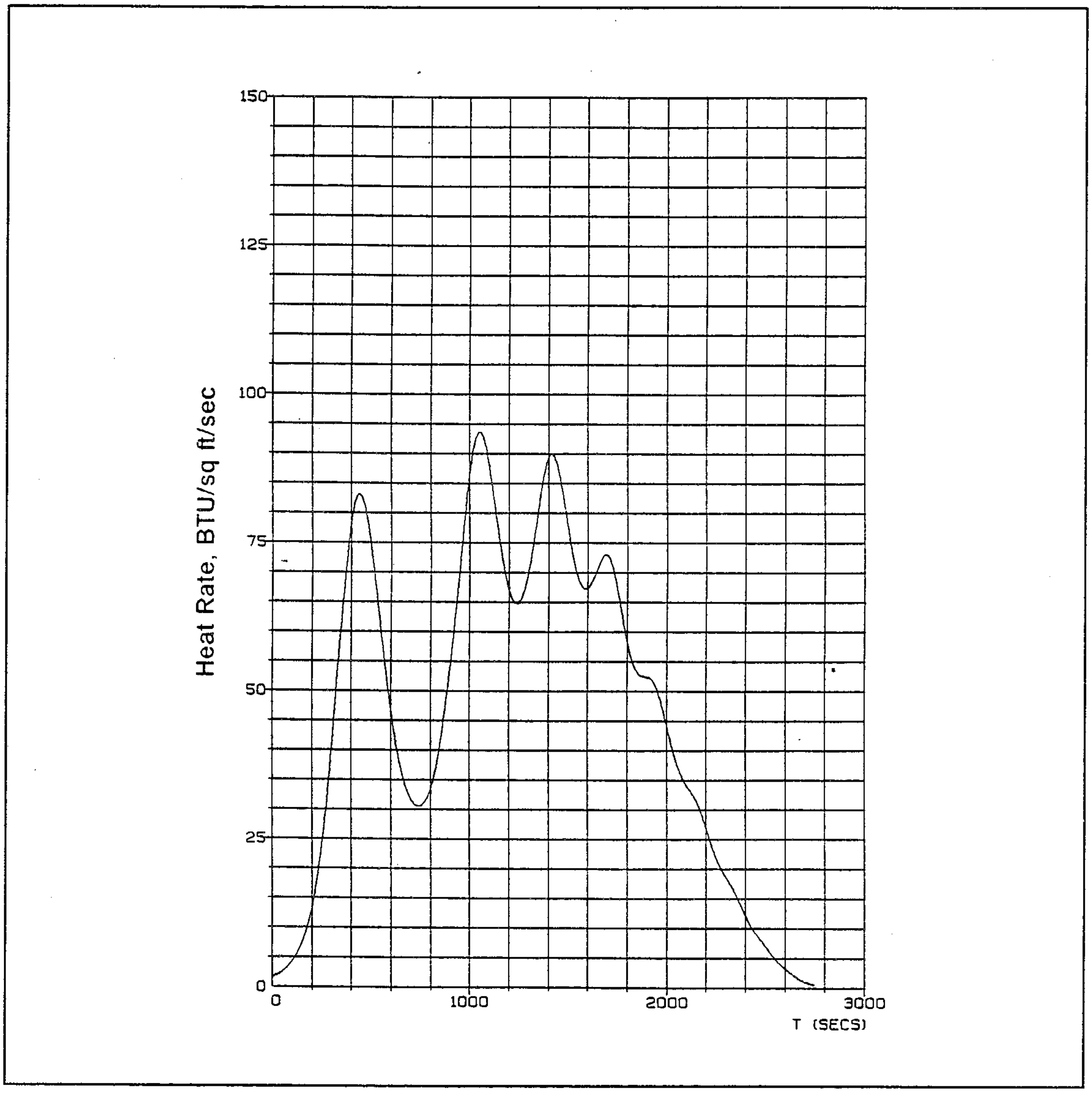


Figure 34. Closed-Loop Heat Rate History for the Maximum Crossrange Case

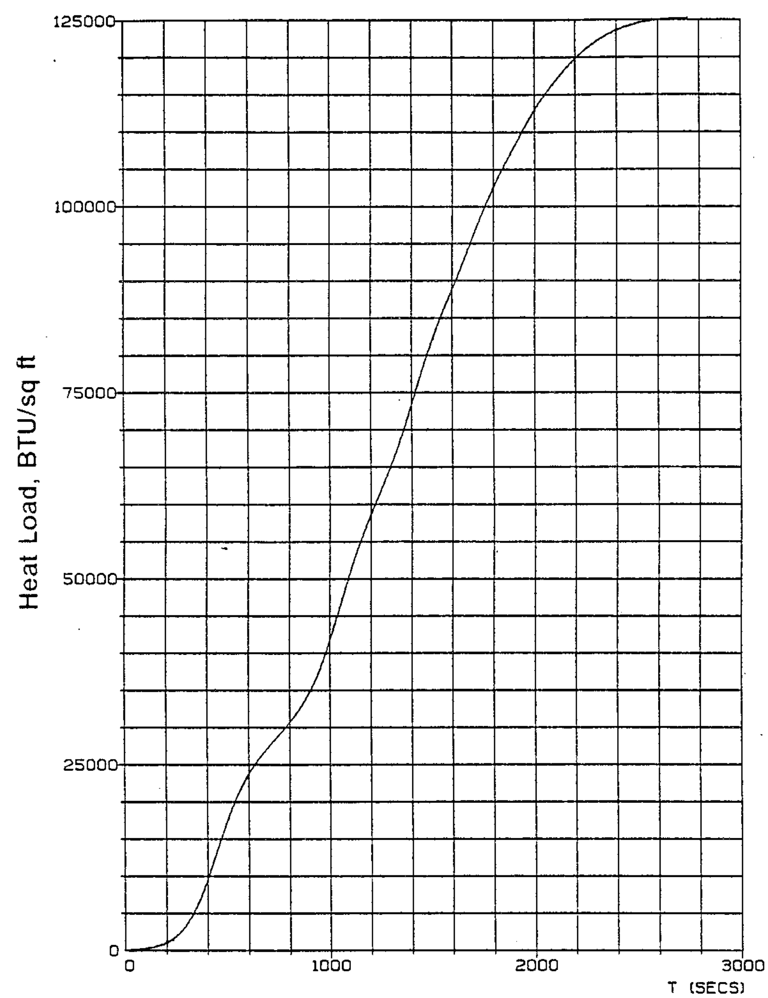


Figure 35. Closed-Loop Heat Load History for the Maximum Crossrange Case

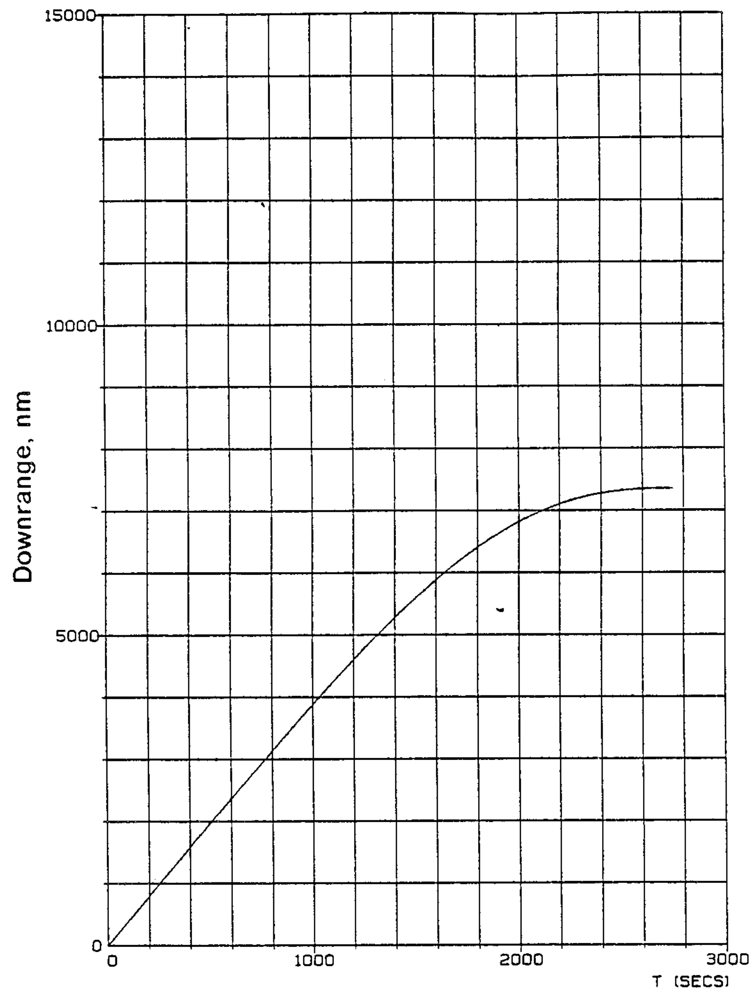


Figure 36. Closed-Loop Downrange History for the Maximum Crossrange Case

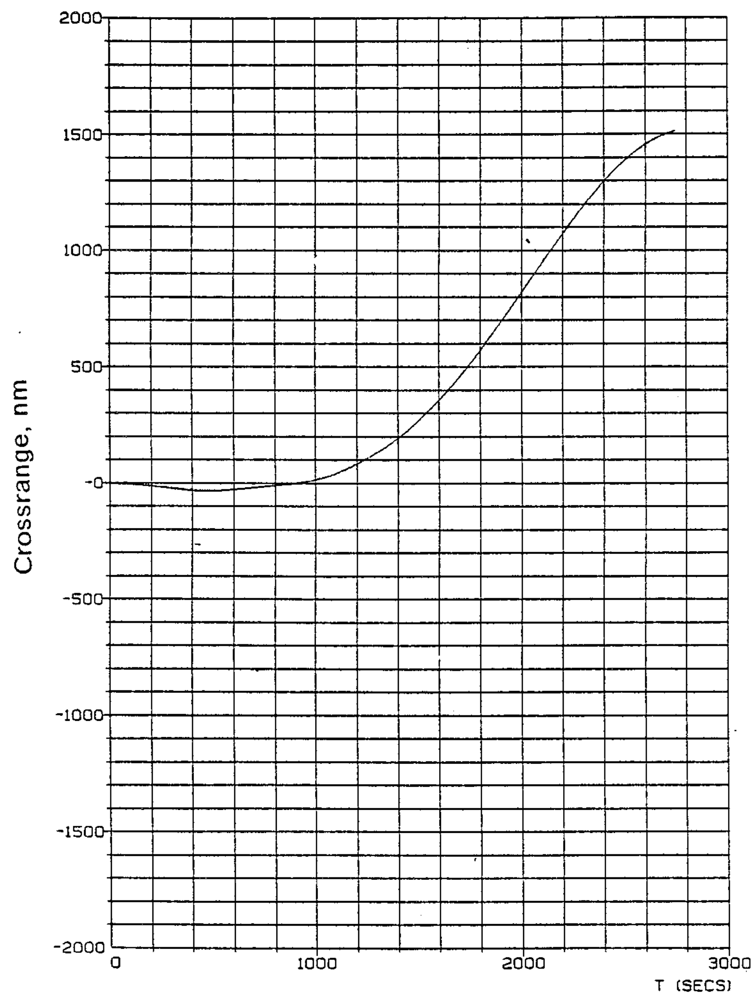


Figure 37. Closed-Loop Crossrange History for the Maximum Crossrange Case

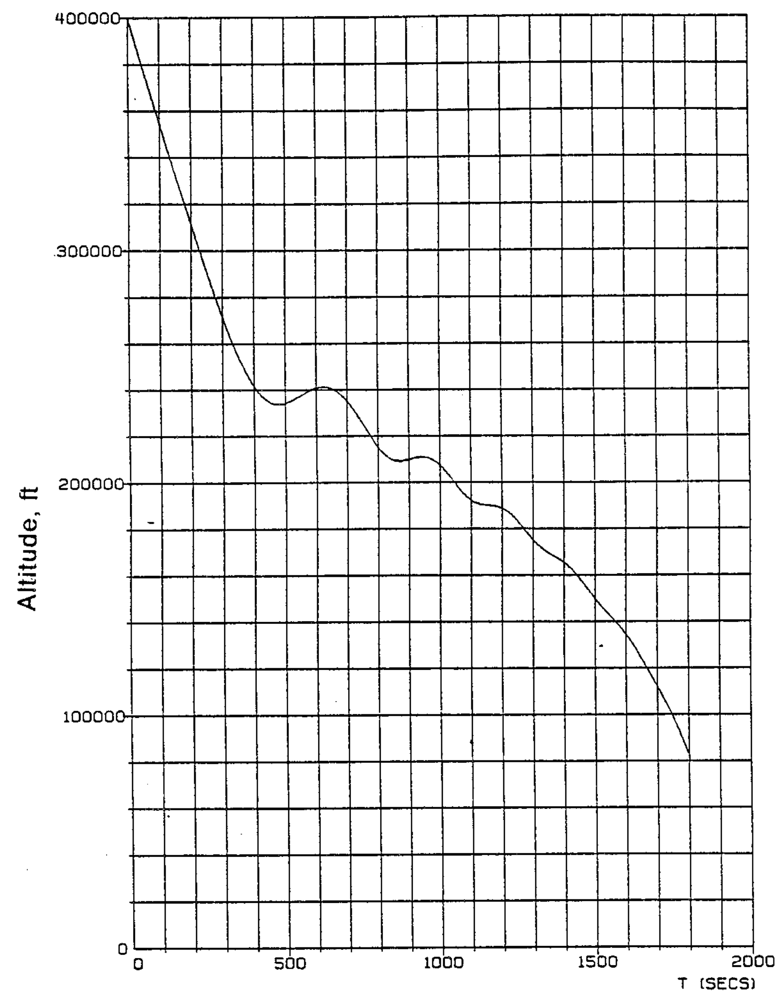


Figure 38. Closed-Loop Altitude History for the Minimum Downrange Case

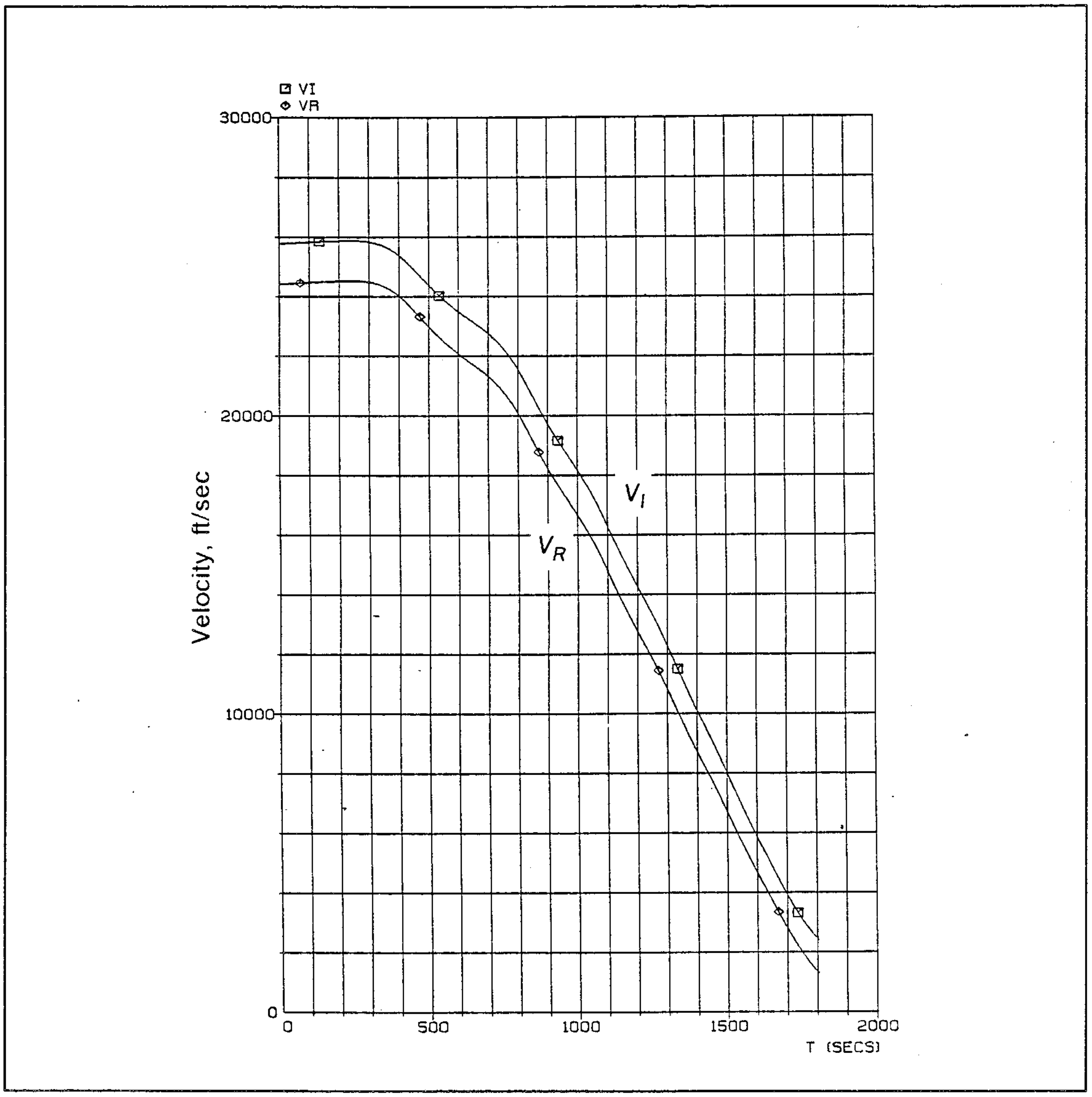


Figure 39. Closed-Loop Velocity History for the Minimum Downrange Case

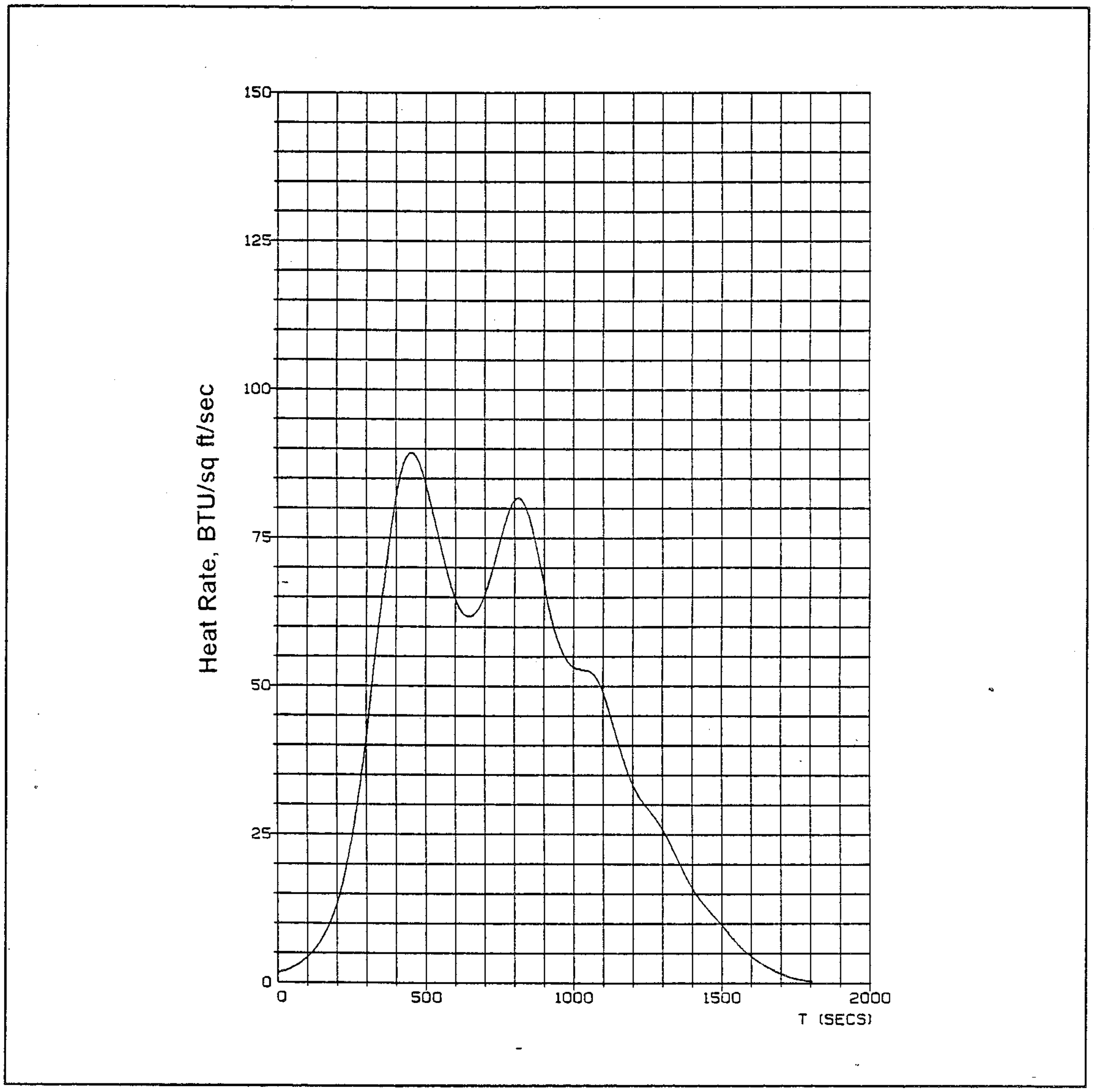


Figure 40. Closed-Loop Heat Rate History for the Minimum Downrange Case

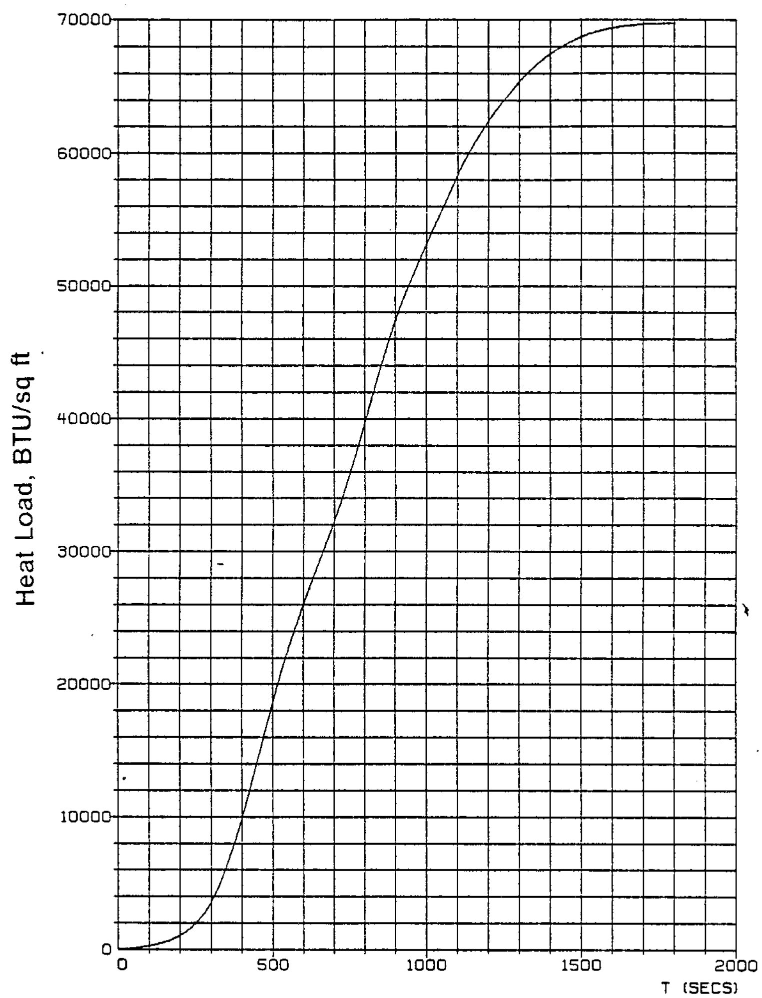


Figure 41. Closed-Loop Heat Load History for the Minimum Downrange Case

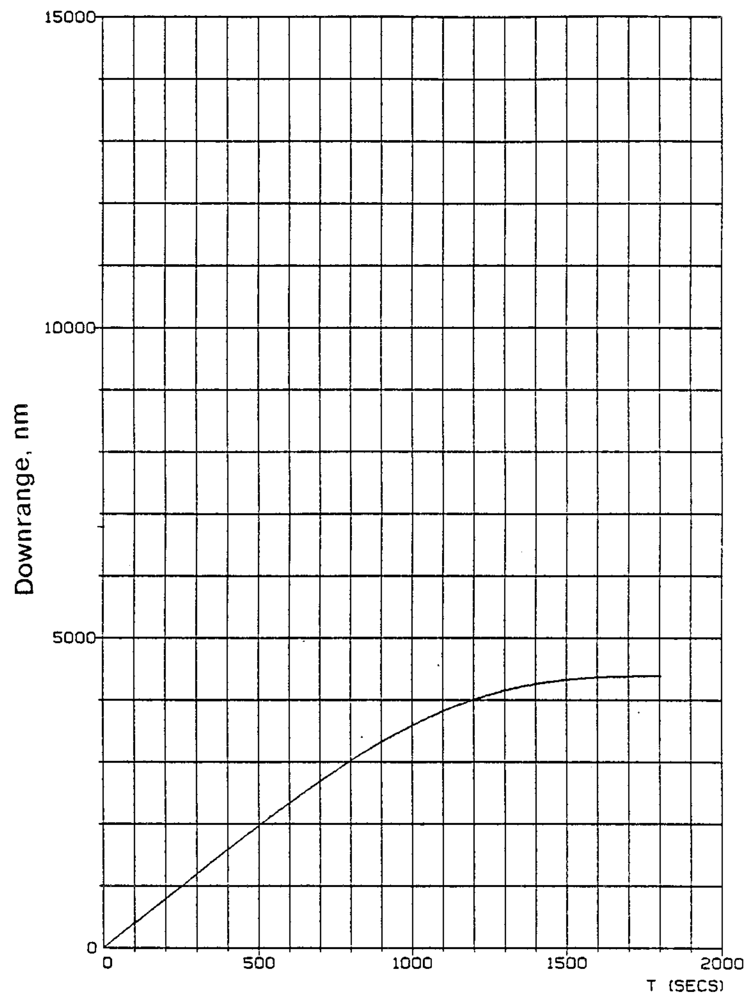


Figure 42. Closed-Loop Downrange History for the Minimum Downrange Case

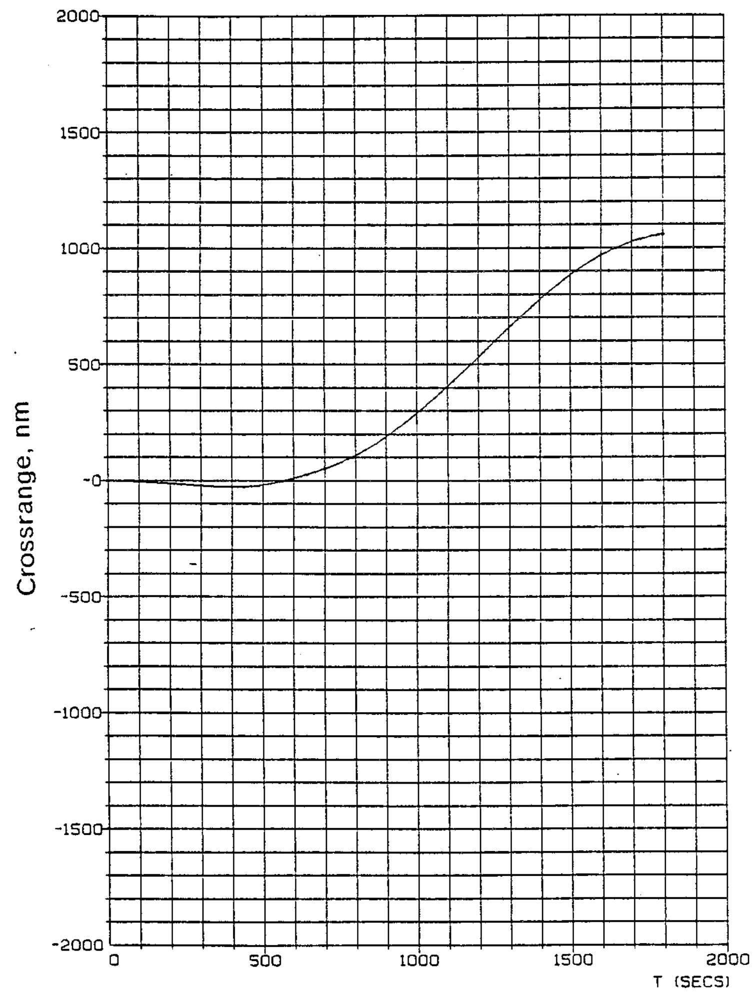


Figure 43. Closed-Loop Crossrange History for the Minimum Downrange Case

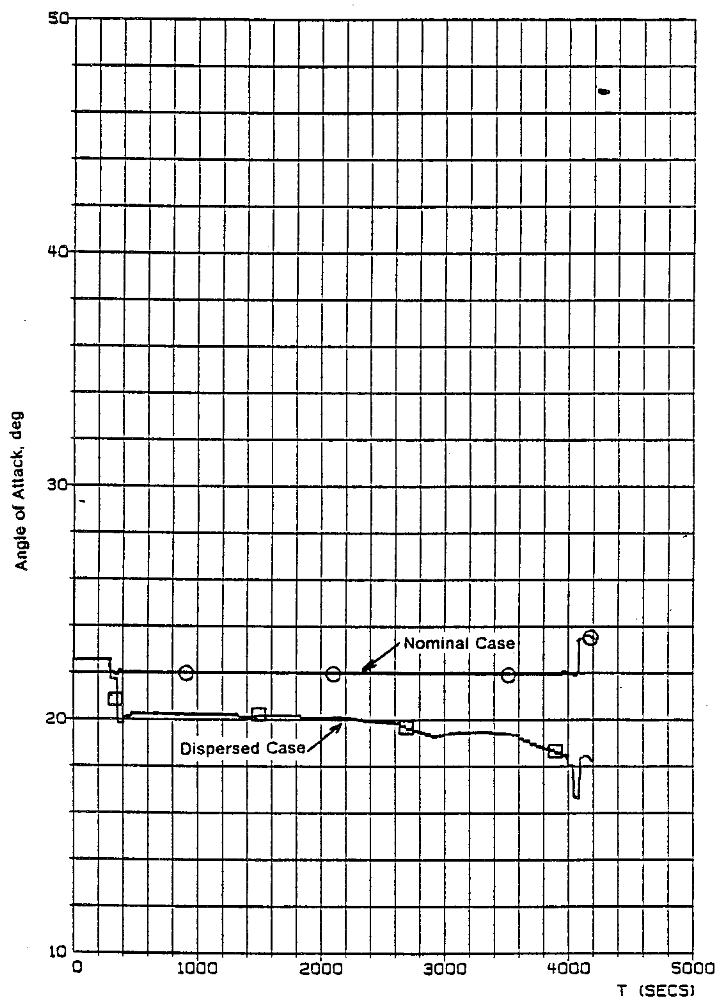


Figure 44. Angle of Attack Comparison for the Maximum Downrange Case

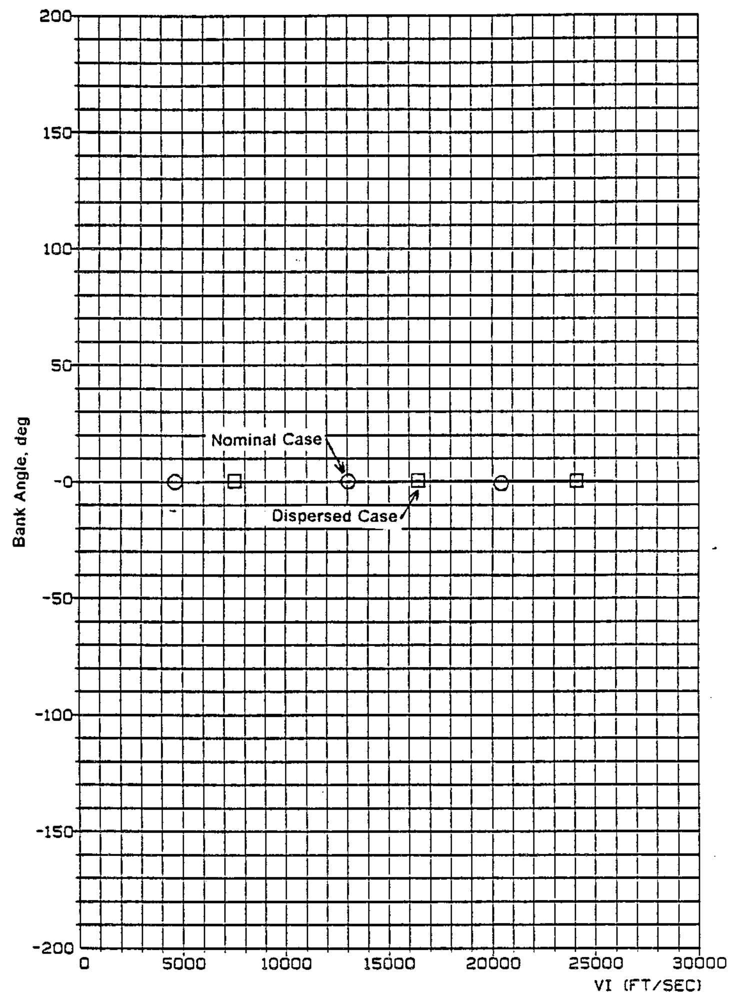


Figure 45. Bank Angle Comparison for the Maximum Downrange Case

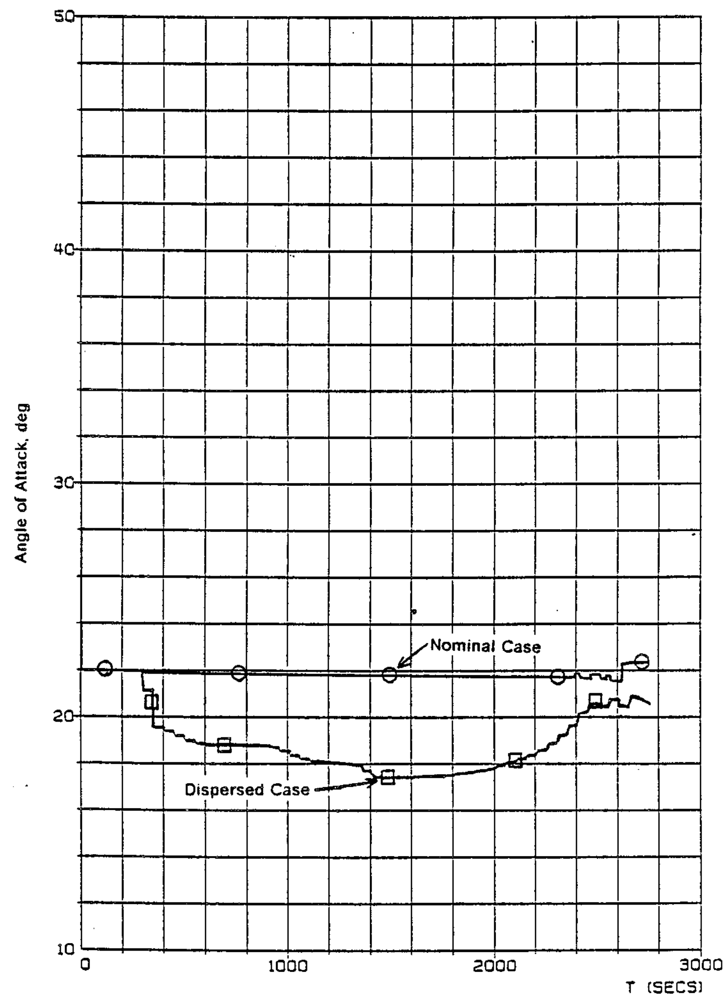


Figure 46. Angle of Attack Comparison for the Maximum Crossrange Case

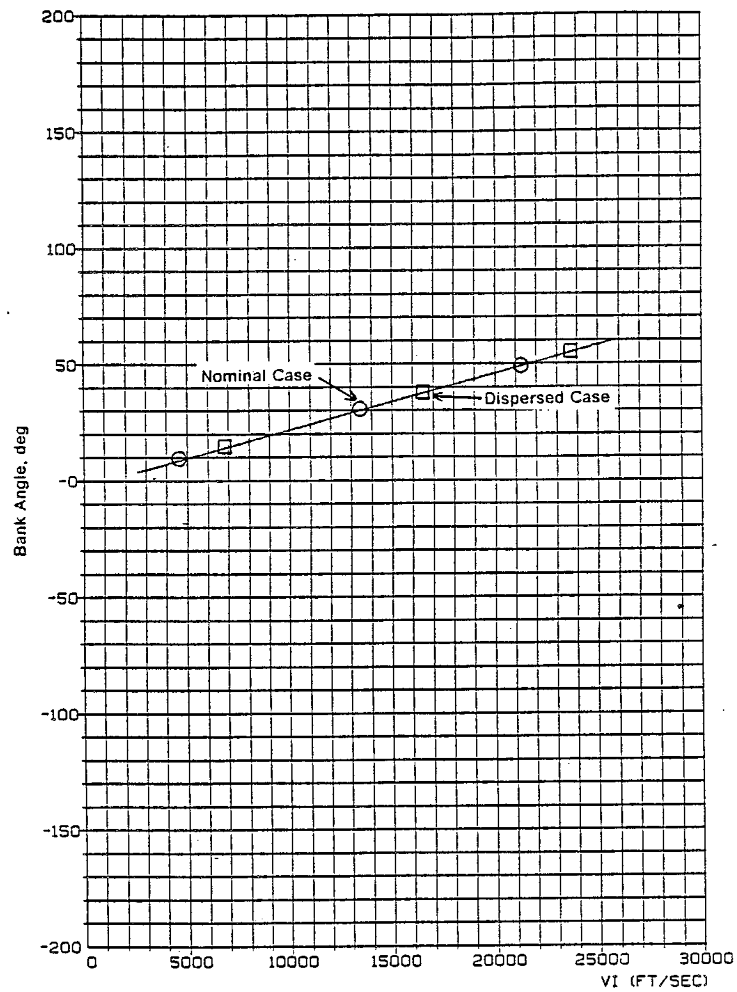


Figure 47. Bank Angle Comparison for the Maximum Crossrange Case

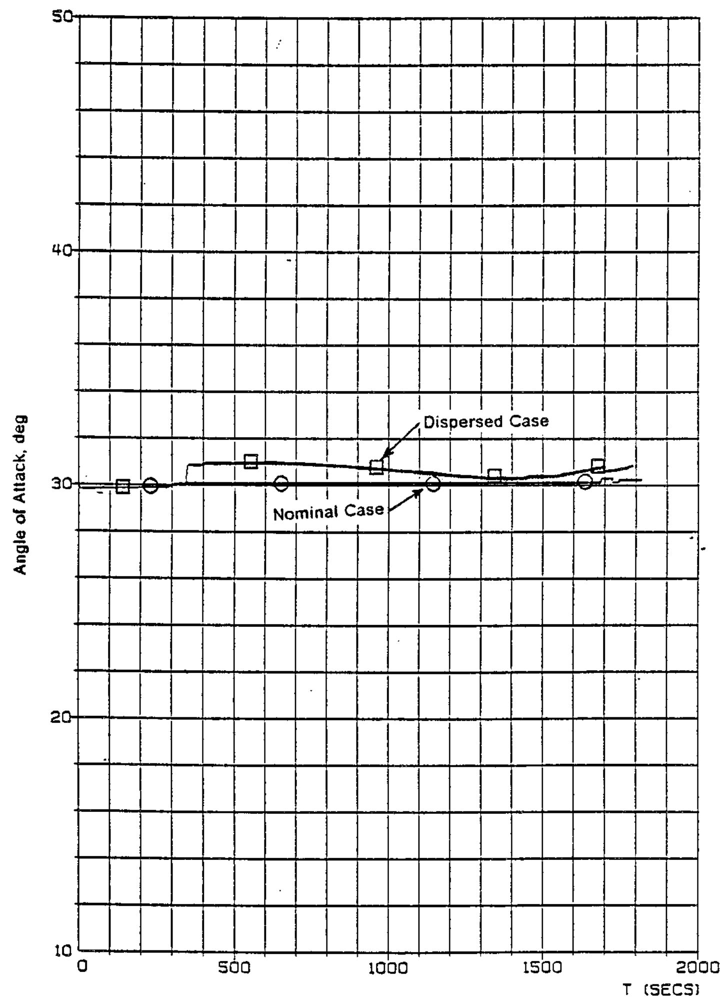


Figure 48. Angle of Attack Comparison for the Minimum Downrange Case

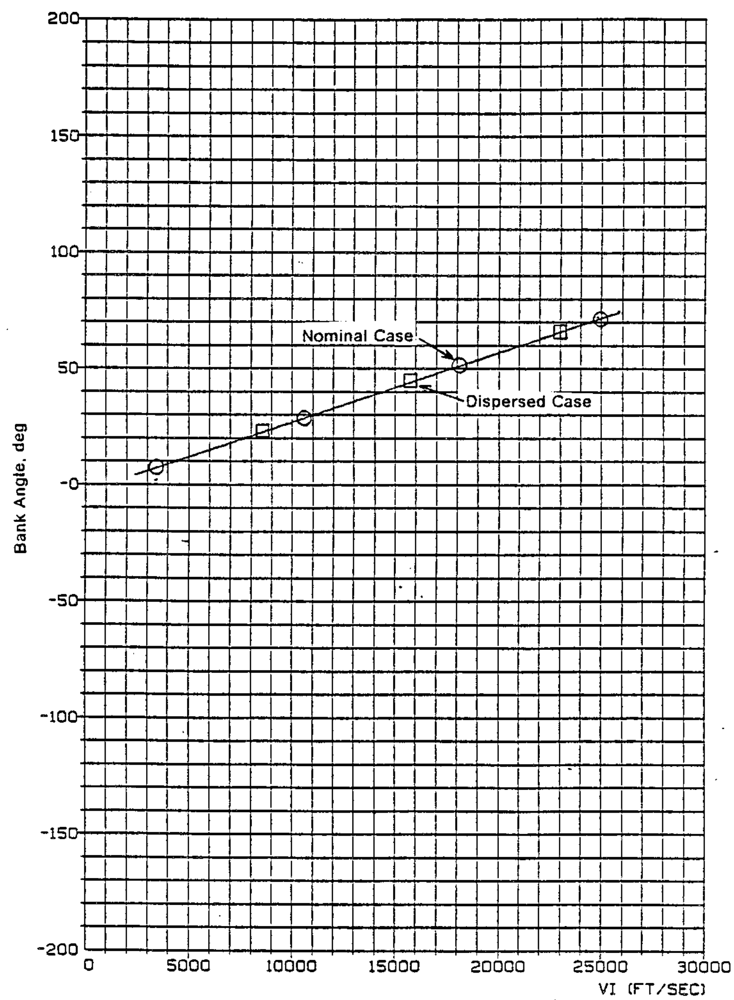


Figure 49. Bank Angle Comparison for the Minimum Downrange Case

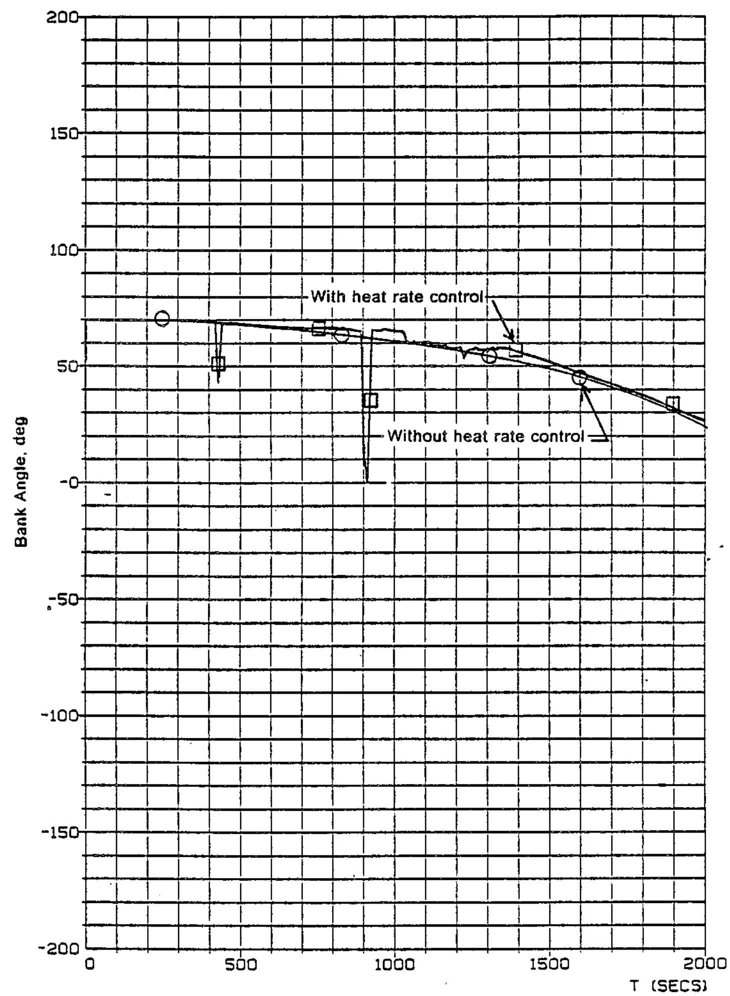


Figure 50. Bank Angle Versus Time Comparison for Heat Rate Control

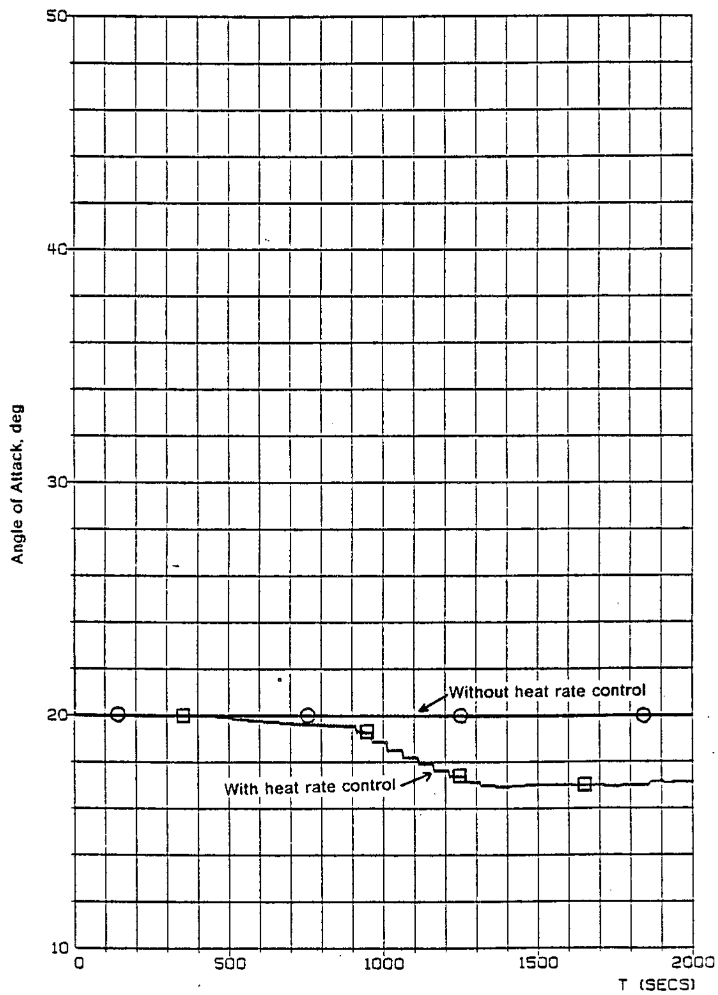


Figure 51. Angle of Attack Versus Time Comparison for Heat Rate Control

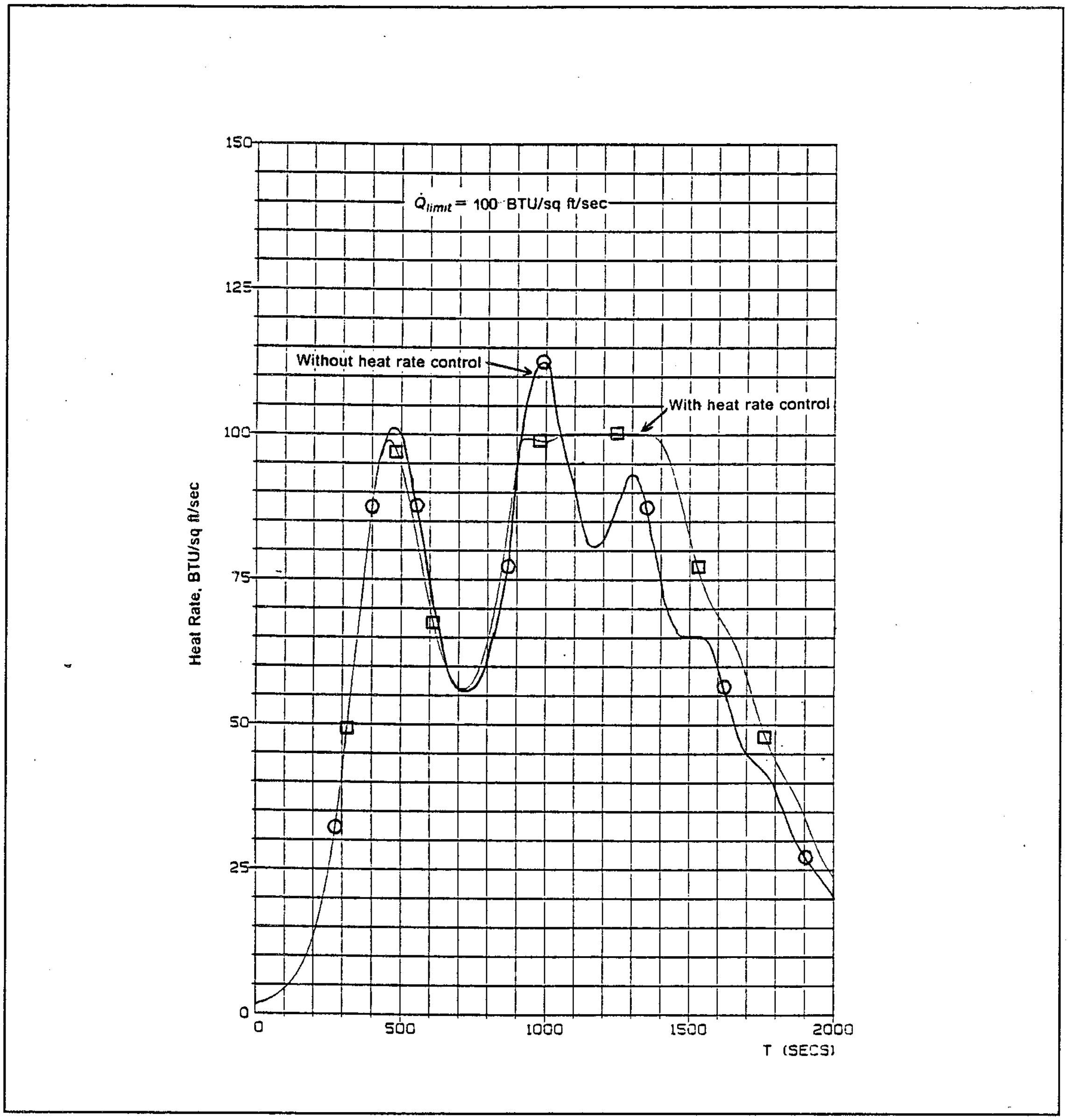


Figure 52. Heat Rate Versus Time Comparison for Heat Rate Control

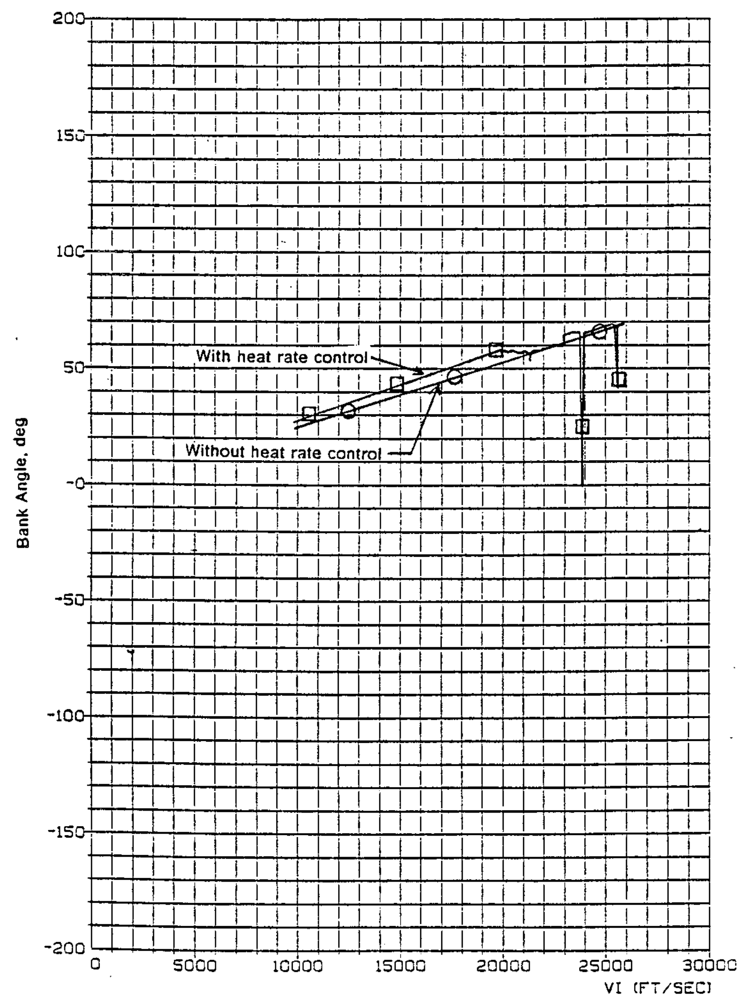


Figure 53. Bank Angle Versus Velocity Comparison for Heat Rate Control

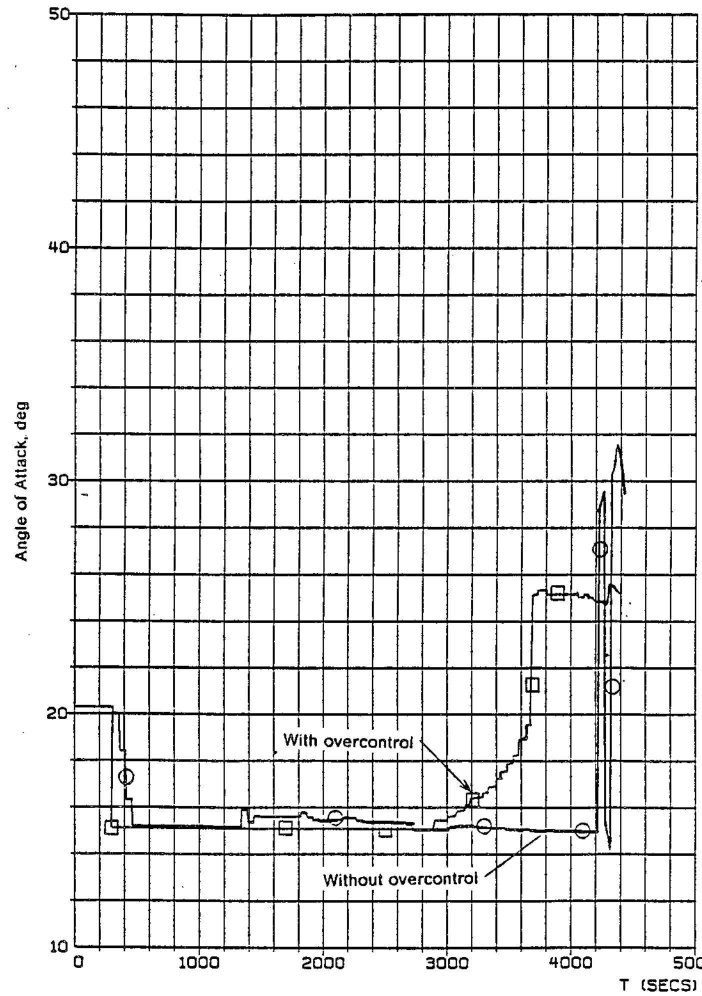


Figure 54. Angle of Attack Versus Time Comparison with Overcontrol

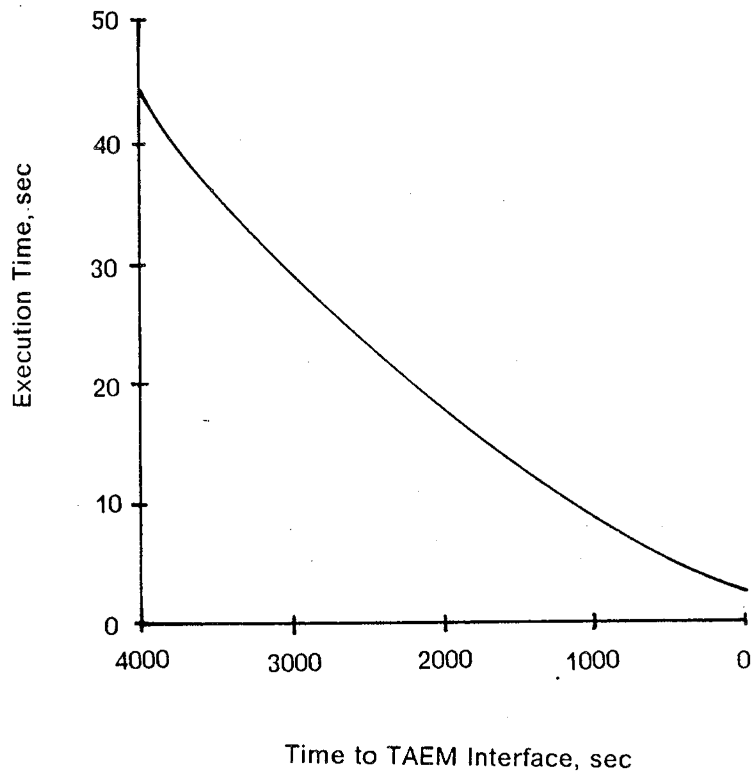


Figure 55. Required Execution Time for the Predictor-Corrector

APPENDIX A. ERV AERODYNAMICS MODEL

The aerodynamics of the ERV were reported in Reference [14], and the longitudinal performance coefficients, C_L and L/D, are shown in Figures 20 on page 97 and 21 on page 98. Figure 22 on page 99 shows a typical L/D versus angle of attack profile. This profile is for a Mach Number of 10, but across the flow regimes, the maximum L/D always occurs at an angle of attack of approximately 15 degrees. This data was incorporated into the aerodynamic model of the simulator and into the aerodynamic model of the predictor. It is seen that the aerodynamic flow regimes are a function of:

1. Mach Number, M
2. Viscous Interaction Parameter, $\overline{V}$
3. Altitude, h

The Mach Number, M, is computed from,

$$M = \frac{V_R}{C_s} \tag{99}$$

where the speed of sound, C_s, is computed from,

$$C_s = \sqrt{\gamma \frac{\mathcal{R}}{M_0} T_M} \tag{100}$$

The viscous interaction parameter, $\overline{V}$, is computed from,

$$\overline{V} = M \sqrt{\frac{C'}{Re}} \tag{101}$$

where,

$$C' = \left(\frac{T'}{T_{static}}\right)^{0.5} \left[\frac{T_{static} + 122.1 \times 10^{-(5/T_{static})}}{T' + 122.1 \times 10^{-(5/T')}} \right]^{1.0} \tag{102}$$

and,

$$\frac{T'}{T_{static}} = 0.468 + 0.532 \frac{T_{wall}}{T_{static}} + 0.195 \frac{\gamma - 1}{2} M^2 \tag{103}$$

The Reynolds Number, Re, is calculated from,

$$Re = \frac{\rho \; V_R \; \bar{c}}{\mu} \tag{104}$$

where the coefficient of viscosity for air, μ, is given by,

$$\mu = \frac{\beta \; T_{static}^{3/2}}{S + T_{static}} \tag{105}$$

APPENDIX B. ALGORITHM PROGRAM LISTINGS

Compiled listings of the flight software principal functions for the predictor-corrector guidance algorithm as coded for use in the 6-DOF Aeroassist Flight Experiment Simulator (AFESIM) follow. The algorithms are coded in the HAL/S computer language. The principal functions are:

1. IL_LOAD - Values for all constants and I-loads
2. FSW_SEQ - Flight Software Sequencer
3. ORB_NAV - Orbit Navigation Algorithm
4. AERO_GUID - Predictor-Corrector Guidance Algorithm

At the beginning of each principal function is a description of the function and the input/output parameters. At the end of each principal function is a cross reference table listing the program line at which each variable is referenced or computed.

```
STMT                               SOURCE                                                CURRENT SCOPE

 211 M| IL_LOAD:                                                                         | IL_LOAD

 211 M| PROCEDURE;                                                                       | IL_LOAD

     C| ----------------------------------------------------------------------          | IL_LOAD
     C| FUNCTION: VALUES OF I-LOADS AND CONSTANTS                                        | IL_LOAD
     C| INPUTS:   NONE                                                                   | IL_LOAD
     C| OUTPUTS:  ALL I-LOADS AND CONSTANTS LISTED                                       | IL_LOAD
     C| COMMENTS: NONE                                                                   | IL_LOAD
     C| ----------------------------------------------------------------------          | IL_LOAD


     C| -------------------------------------                                            | IL_LOAD
     C| MATH CONSTANTS AND CONVERSION FACTORS                                            | IL_LOAD
     C| -------------------------------------                                            | IL_LOAD
 212 M|    DECLARE PI SCALAR DOUBLE CONSTANT(3.1415926535897932385);                     | IL_LOAD
 213 M|    DECLARE DEG_TO_SEC SCALAR DOUBLE CONSTANT(3600);                              | IL_LOAD
 214 M|    DECLARE SEC_TO_DEG SCALAR DOUBLE CONSTANT(1 / DEG_TO_SEC);                    | IL_LOAD
 215 M|    DECLARE DEG_TO_RAD SCALAR DOUBLE CONSTANT(PI / 180);                          | IL_LOAD
 216 M|    DECLARE RAD_TO_DEG SCALAR DOUBLE CONSTANT(1 / DEG_TO_RAD);                    | IL_LOAD
 217 M|    DECLARE SEC_TO_RAD SCALAR DOUBLE CONSTANT(DEG_TO_RAD / 3600);                 | IL_LOAD
 218 M|    DECLARE RAD_TO_SEC SCALAR DOUBLE CONSTANT(1 / SEC_TO_RAD);                    | IL_LOAD
 219 M|    DECLARE FT_TO_M SCALAR DOUBLE CONSTANT(0.3048);                               | IL_LOAD
 220 M|    DECLARE M_TO_FT SCALAR DOUBLE CONSTANT(1 / FT_TO_M);                          | IL_LOAD
 221 M|    DECLARE FT_TO_NM SCALAR DOUBLE CONSTANT(FT_TO_M / 1852);                      | IL_LOAD
 222 M|    DECLARE NM_TO_FT SCALAR DOUBLE CONSTANT(1 / FT_TO_NM);                        | IL_LOAD
 223 M|    DECLARE G_TO_FPS2 SCALAR DOUBLE CONSTANT(9.80665 M_TO_FT);                    | IL_LOAD
 224 M|    DECLARE FPS2_TO_G SCALAR DOUBLE CONSTANT(1 / G_TO_FPS2);                      | IL_LOAD
 225 M|    DECLARE LBM_TO_KG SCALAR DOUBLE CONSTANT(.45359237);                          | IL_LOAD
 226 M|    DECLARE KG_TO_LBM SCALAR DOUBLE CONSTANT(1 / LBM_TO_KG);                      | IL_LOAD
 227 M|    DECLARE SLUG_TO_KG SCALAR DOUBLE CONSTANT(LBM_TO_KG G_TO_FPS2);               | IL_LOAD
 228 M|    DECLARE KG_TO_SLUG SCALAR DOUBLE CONSTANT(KG_TO_LBM FPS2_TO_G);               | IL_LOAD
 229 M|    DECLARE LBF_TO_N SCALAR DOUBLE CONSTANT(4.4482216152605);                     | IL_LOAD
 230 M|    DECLARE N_TO_LBF SCALAR DOUBLE CONSTANT(1 / LBF_TO_N);                        | IL_LOAD
```

```
STMT                                          SOURCE                                                            CURRENT SCOPE

      C|  -----------------                                                                               | IL_LOAD
      C|  FSW SEQ VARIABLES                                                                               | IL_LOAD
      C|  -----------------                                                                               | IL_LOAD

 231 M|     AERO_DAP_CNT = 1;                                                                             | IL_LOAD

 232 M|     AERO_DAP_PHS = 0;                                                                             | IL_LOAD

 233 M|     AERO_GUID_CNT = 5;                                                                            | IL_LOAD

 234 M|     AERO_GUID_PHS = 0;                                                                            | IL_LOAD

 235 M|     ORB_NAV_CNT = 5;                                                                              | IL_LOAD

 236 M|     ORB_NAV_PHS = 0;                                                                              | IL_LOAD

      C|  ----------                                                                                      | IL_LOAD
      C|  EPOCH DATA                                                                                      | IL_LOAD
      C|  ----------                                                                                      | IL_LOAD

      E|              *                                                                                   |
 237 M|     EF_TO_REF_AT_EPOCH = MATRIX          (+1.0, +0.0, +0.0, +0.0, +1.0, +0.0, +0.0, +0.0, +1.0);  | IL_LOAD
      S|                                 @DOUBLE,3,3                                                      |

 238 M|     T_EPOCH = 0;                                                                                  | IL_LOAD

      C|  -------------------------                                                                       | IL_LOAD
      C|  EARTH PHYSICAL PARAMETERS                                                                       | IL_LOAD
      C|  -------------------------                                                                       | IL_LOAD

 239 M|     EARTH_FLAT = 1 / 298.3;                                                                       | IL_LOAD

 240 M|     EARTH_J2 = 1082.7E-6;                                                                         | IL_LOAD

      E|                                         3                                                        |
 241 M|     EARTH_MU = (3.986012E14) / (.3048) ;                                                          | IL_LOAD

      E|          -                                                                                       |
 242 M|     EARTH_POLE = VECTOR         (2.8991969340471790E-3, -5.1580036678323420E-5,                   | IL_LOAD
      S|                         @DOUBLE,3                                                                |

 242 M|     9.9999579598948170E-1);                                                                       | IL_LOAD

 243 M|     EARTH_R = (6378166.0) / .3048;                                                                | IL_LOAD

 244 M|     EARTH_RATE = 7.292114883223324E-5;                                                            | IL_LOAD

      E|       -                     -                                                                    |
 245 M|     WE_NAV = EARTH_RATE EARTH_POLE;                                                               | IL_LOAD
```

```
STMT                                   SOURCE                                   CURRENT SCOPE

    C| ----------------------------------------------                          | IL_LOAD
    C| PREDICTOR-CORRECTOR I-LOADS (MISSION-SPECIFIC)                          | IL_LOAD
    C| ----------------------------------------------                          | IL_LOAD

    C| -------------------------------------------                             | IL_LOAD
    C| TARGET AIM POINT FOR MAXIMUM DOWNRANGE CASE                             | IL_LOAD
    C| DOWNRANGE  = 13849 N.M.                                                 | IL_LOAD
    C| CROSSRANGE =   497 N.M.                                                 | IL_LOAD
    C| -------------------------------------------                             | IL_LOAD

    C| -------------------------------------                                   | IL_LOAD
    C| INITIAL CONDITIONS:                                                     | IL_LOAD
    C| -------------------------------------                                   | IL_LOAD
    C| LATITUDE                -28.071 DEG                                     | IL_LOAD
    C| LONGITUDE               -69.313 DEG                                     | IL_LOAD
    C| INCLINATION              28.50  DEG                                     | IL_LOAD
    C| FLIGHT PATH ANGLE        -0.996 DEG                                     | IL_LOAD
    C| INERTIAL VELOCITY    25778.843 FT/SEC                                   | IL_LOAD
    C| ALTITUDE            400000.0   FT                                       | IL_LOAD
    C| -------------------------------------                                   | IL_LOAD

    C| ---------------------------------------------                           | IL_LOAD
    C| GEODETIC LATITUDE OF TAEM INTERFACE AIM POINT                           | IL_LOAD
    C| ---------------------------------------------                           | IL_LOAD

246 M|   LAT_TARGET = 3.084;                                                   | IL_LOAD

    C| -------------------------------------                                   | IL_LOAD
    C| LONGITUDE OF TAEM INTERFACE AIM POINT                                   | IL_LOAD
    C| -------------------------------------                                   | IL_LOAD

247 M|   LONG_TARGET = 157.659;                                                | IL_LOAD

    C| -------------------------------------------------                       | IL_LOAD
    C| INITIAL CONTROL VALUES FOR MAXIMUM DOWNRANGE CASE                       | IL_LOAD
    C| -------------------------------------------------                       | IL_LOAD

248 M|   PHI_EI = 0.0;                                                         | IL_LOAD

249 M|   ALPHA_EI = 20.0;                                                      | IL_LOAD

    C| -------------------------------------------                             | IL_LOAD
    C| ESTIMATOR FILTER GAINS (TAU = 25.0 SECONDS)                             | IL_LOAD
    C| -------------------------------------------                             | IL_LOAD

250 M|   K_RHO_FILTER_GAIN = .03921;                                           | IL_LOAD

251 M|   L_OVER_D_FILTER_GAIN = .03921;                                        | IL_LOAD

    C| --------------------                                                    | IL_LOAD
    C| VEHICLE MASS (SLUGS)                                                    | IL_LOAD
    C| --------------------                                                    | IL_LOAD

252 M|   MASS_NAV = 186.0;                                                     | IL_LOAD
```

```
STMT                                   SOURCE                                                 CURRENT SCOPE

    C|  ---------------------------------------------------------------------           | IL_LOAD
    C|  PREDICTOR-CORRECTOR I-LOADS (DESIGN PARAMETERS NOT NORMALLY CHANGED)            | IL_LOAD
    C|  ---------------------------------------------------------------------           | IL_LOAD

    C|  -----------------------------------------------------------------               | IL_LOAD
    C|  ALTITUDES AT WHICH GUIDANCE QUITS, FREEZES COMMAND, TARGETS, AND                | IL_LOAD
    C|  USES MINIMUM TIME STEP FOR INTEGRATION                                          | IL_LOAD
    C|  -----------------------------------------------------------------               | IL_LOAD

253 M|    ALT_EXIT = 400000.0;                                                          | IL_LOAD

254 M|    ALT_FREEZE_GUID = 100000.0;                                                   | IL_LOAD

255 M|    ALT_TAEM = 80000.0;                                                           | IL_LOAD

256 M|    ALT_TAEM_BIAS = ALT_TAEM + 10000.0;                                           | IL_LOAD

    C|  -------------------------------------                                           | IL_LOAD
    C|  G-LEVEL AT WHICH TO ACTIVATE GUIDANCE                                           | IL_LOAD
    C|  -------------------------------------                                           | IL_LOAD

257 M|    G_RUN_GUIDANCE = 0.07;                                                        | IL_LOAD

    C|  --------------------------                                                      | IL_LOAD
    C|  GRAVITY MODEL WITH J2 TERM                                                      | IL_LOAD
    C|  --------------------------                                                      | IL_LOAD

258 M|    GRAVITY_MODEL = 1;                                                            | IL_LOAD

    C|  ---------------------                                                           | IL_LOAD
    C|  REFERENCE AREA OF ERV                                                           | IL_LOAD
    C|  ---------------------                                                           | IL_LOAD

259 M|    S_REF = 177.4;                                                                | IL_LOAD

    C|  ------------------------------------------------------------------              | IL_LOAD
    C|  TIME INCREMENT BETWEEN EXECUTIONS OF PREDICTOR-CORRECTOR EXECUTIVE              | IL_LOAD
    C|  ------------------------------------------------------------------              | IL_LOAD

260 M|    DT_AEROGUID = 1.0;                                                            | IL_LOAD

    C|  -------------------------------------------------------------                   | IL_LOAD
    C|  NUMBER OF EXECUTIONS OF PREDICTOR-CORRECTOR EXECUTIVE BETWEEN                   | IL_LOAD
    C|  UPDATES OF COMMANDED ATTITUDE USING PREDICTOR-CORRECTOR                         | IL_LOAD
    C|  -------------------------------------------------------------                   | IL_LOAD

261 M|    GUID_PASS_LIM = 50;                                                           | IL_LOAD
```

```
STMT                                     SOURCE                                        CURRENT SCOPE

     C| -------------------------------------------                                   | IL_LOAD
     C| LINEAR BANK WITH VELOCITY PROFILE CONSTANTS                                   | IL_LOAD
     C| -------------------------------------------                                   | IL_LOAD

 262 M|   PHI_DES_MAX = 180.0;                                                        | IL_LOAD

 263 M|   PHI_MAX = 90.0;                                                             | IL_LOAD

 264 M|   V_FINAL_MAG = 1000.0;                                                       | IL_LOAD

 265 M|   V_INITIAL_MAG = 26000.0;                                                    | IL_LOAD

 266 M|   V_MAG_CHANGE = V_INITIAL_MAG - V_FINAL_MAG;                                 | IL_LOAD

     C| ------------------------------------------                                    | IL_LOAD
     C| CONSTANT ANGLE OF ATTACK PROFILE CONSTANTS                                    | IL_LOAD
     C| ------------------------------------------                                    | IL_LOAD

 267 M|   ALPHA_MAX = 45.0;                                                           | IL_LOAD

 268 M|   ALPHA_MIN = 15.0;                                                           | IL_LOAD

     C| ------------------------------------                                          | IL_LOAD
     C| VARIABLE TIME STEP CONTROL CONSTANTS                                          | IL_LOAD
     C| ------------------------------------                                          | IL_LOAD

 269 M|   DELTA_T_PRED_GAIN = 200.0;                                                  | IL_LOAD

 270 M|   DELTA_T_PRED_MAX = 20.0;                                                    | IL_LOAD

 271 M|   DELTA_T_PRED_MIN = 2.0;                                                     | IL_LOAD

     C| ---------------------------                                                   | IL_LOAD
     C| HEAT RATE CONTROL CONSTANTS                                                   | IL_LOAD
     C| ---------------------------                                                   | IL_LOAD

 272 M|   HS = 23500.0;                                                               | IL_LOAD

 273 M|   OMEGA_QDOT = 0.10;                                                          | IL_LOAD

 274 M|   QDOT_LIMIT = 125.0;                                                         | IL_LOAD

 275 M|   RHO_SL = 0.002378;                                                          | IL_LOAD

 276 M|   ZETA_QDOT = 1.00;                                                           | IL_LOAD

 277 M| CLOSE IL_LOAD;                                                                | IL_LOAD

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
```

```
AERO_DAP_CNT*, AERO_DAP_PHS*, AERO_GUID_CNT*, AERO_GUID_PHS*, ORB_NAV_CNT*, ORB_NAV_PHS*, EF_TO_REF_AT_EPOCH*, T_EPOCH*
EARTH_FLAT*, EARTH_J2*, EARTH_MU*, EARTH_POLE*, EARTH_R*, EARTH_RATE*, WE_NAV*, EARTH_RATE, EARTH_POLE, LAT_TARGET*
LONG_TARGET*, PHI_EI*, ALPHA_EI*, K_RHO_FILTER_GAIN*, L_OVER_D_FILTER_GAIN*, MASS_NAV*, ALT_EXIT*, ALT_FREEZE_GUID*, ALT_TAEM*
ALT_TAEM_BIAS*, ALT_TAEM, G_RUN_GUIDANCE*, GRAVITY_MODEL*, S_REF*, DT_AEROGUID*, GUID_PASS_LIM*, PHI_DES_MAX*, PHI_MAX*
V_FINAL_MAG*, V_INITIAL_MAG*, V_MAG_CHANGE*, V_INITIAL_MAG, V_FINAL_MAG, ALPHA_MAX*, ALPHA_MIN*, DELTA_T_PRED_GAIN*
DELTA_T_PRED_MAX*, DELTA_T_PRED_MIN*, HS*, OMEGA_QDOT*, QDOT_LIMIT*, RHO_SL*, ZETA_QDOT*
```

**** C O M P I L A T I O N L A Y O U T ****

```
IL_POOL: EXTERNAL COMPOOL;

IL_LOAD: PROCEDURE;
```

S Y M B O L & C R O S S R E F E R E N C E T A B L E L I S T I N G :

(CROSS REFERENCE FLAG KEY: 4 = ASSIGNMENT, 2 = REFERENCE, 1 = SUBSCRIPT USE, 0 = DEFINITION)

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
21	AERO_DAP_CNT	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0021 4 0231 NOT REFERENCED
20	AERO_DAP_PHS	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0020 4 0232 NOT REFERENCED
23	AERO_GUID_CNT	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0023 4 0233 NOT REFERENCED
22	AERO_GUID_PHS	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0022 4 0234 NOT REFERENCED
97	ALPHA_EI	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0097 4 0249 NOT REFERENCED
209	ALPHA_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0267 NOT REFERENCED
209	ALPHA_MIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0268 NOT REFERENCED
79	ALT_EXIT	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0079 4 0253 NOT REFERENCED
209	ALT_FREEZE_GUID	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0254 NOT REFERENCED
209	ALT_TAEM	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0255 2 0256
209	ALT_TAEM_BIAS	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0256 NOT REFERENCED
215	DEG_TO_RAD	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0215 2 0216 2 0217
213	DEG_TO_SEC	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0213 2 0214
209	DELTA_T_PRED_GAIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0269 NOT REFERENCED
209	DELTA_T_PRED_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0270 NOT REFERENCED
209	DELTA_T_PRED_MIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0271 NOT REFERENCED
209	DT_AEROGUID	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0260 NOT REFERENCED
159	EARTH_FLAT	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0159 4 0239 NOT REFERENCED
160	EARTH_J2	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0160 4 0240 NOT REFERENCED
161	EARTH_MU	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0161 4 0241 NOT REFERENCED
162	EARTH_POLE	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0162 4 0242 2 0245
163	EARTH_R	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0163 4 0243 NOT REFERENCED
164	EARTH_RATE	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0164 4 0244 2 0245
4	EF_TO_REF_AT_EPOCH	3 X 3 MATRIX	DOUBLE, ALIGNED, INITIAL XREF: 0 0004 4 0237 NOT REFERENCED
224	FPS2_TO_G	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0224 2 0228
219	FT_TO_M	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0219 2 0220 2 0221
221	FT_TO_NM	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0221 2 0222
209	G_RUN_GUIDANCE	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0257 NOT REFERENCED
223	G_TO_FPS2	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0223 2 0224 2 0227
165	GRAVITY_MODEL	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0165 4 0258 NOT REFERENCED
209	GUID_PASS_LIM	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0261 NOT REFERENCED
209	HS	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0272 NOT REFERENCED
211	IL_LOAD	PROCEDURE	XREF: 0 0211 NOT REFERENCED
168	K_RHO_FILTER_GAIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0168 4 0250 NOT REFERENCED
226	KG_TO_LBM	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0226 2 0228
228	KG_TO_SLUG	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0228 NOT REFERENCED
171	L_OVER_D_FILTER_GAIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0171 4 0251 NOT REFERENCED
209	LAT_TARGET	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0246 NOT REFERENCED
229	LBF_TO_N	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0229 2 0230
225	LBM_TO_KG	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0225 2 0226 2 0227
209	LONG_TARGET	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0247 NOT REFERENCED
220	M_TO_FT	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0220 2 0223
10	MASS_NAV	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0010 4 0252 NOT REFERENCED
230	N_TO_LBF	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0230 NOT REFERENCED

ORIGINAL PAGE IS OF POOR QUALITY

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
222	NM_TO_FT	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0222 NOT REFERENCED
209	OMEGA_QDOT	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0273 NOT REFERENCED
35	ORB_NAV_CNT	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0035 4 0235 NOT REFERENCED
34	ORB_NAV_PHS	INTEGER	SINGLE, ALIGNED, INITIAL XREF: 0 0034 4 0236 NOT REFERENCED
209	PHI_DES_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0262 NOT REFERENCED
98	PHI_EI	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0098 4 0248 NOT REFERENCED
209	PHI_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0263 NOT REFERENCED
212	PI	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0212 2 0215
209	QDOT_LIMIT	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0274 NOT REFERENCED
216	RAD_TO_DEG	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0216 NOT REFERENCED
218	RAD_TO_SEC	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0218 NOT REFERENCED
209	RHO_SL	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0275 NOT REFERENCED
174	S_REF	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0174 4 0259 NOT REFERENCED
214	SEC_TO_DEG	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0214 NOT REFERENCED
217	SEC_TO_RAD	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0217 2 0218
227	SLUG_TO_KG	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0227 NOT REFERENCED
6	T_EPOCH	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0006 4 0238 NOT REFERENCED
209	V_FINAL_MAG	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0264 2 0266
209	V_INITIAL_MAG	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0265 2 0266
209	V_MAG_CHANGE	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0266 NOT REFERENCED
176	WE_NAV	3 - VECTOR	SINGLE, ALIGNED, INITIAL XREF: 0 0176 4 0245 NOT REFERENCED
209	ZETA_QDOT	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0209 4 0276 NOT REFERENCED

```
STMT                                 SOURCE                                          CURRENT SCOPE

 518 M| FSW_SEQ:                                                                    | FSW_SEQ

 518 M| PROCEDURE;                                                                  | FSW_SEQ

     C|  ----------------------------------------------------------------------     | FSW_SEQ
     C|  FUNCTION: EXECUTE FLIGHT SOFTWARE PRINCIPAL FUNCTIONS AT                   | FSW_SEQ
     C|            PROPER RATE AND IN PROPER ORDER WHEN FUNCTIONS ACTIVE            | FSW_SEQ
     C|  INPUTS:   ORB_NAV_ACT   - ORBIT NAVIGATION ACTIVE FLAG                     | FSW_SEQ
     C|            AERO_GUID_ACT - PREDICTOR-CORRECTOR ACTIVE FLAG                  | FSW_SEQ
     C|            AERO_DAP_ACT  - DIGITAL AUTOPILOT ACTIVE FLAG                    | FSW_SEQ
     C|  OUTPUTS:  NONE                                                             | FSW_SEQ
     C|  COMMENTS: NONE                                                             | FSW_SEQ
     C|  ----------------------------------------------------------------------     | FSW_SEQ


     C|  ---------------                                                            | FSW_SEQ
     C|  LOCAL VARIABLES                                                            | FSW_SEQ
     C|  ---------------                                                            | FSW_SEQ

 519 M|    DECLARE FSW_PASS INTEGER DOUBLE INITIAL(0);                              | FSW_SEQ
```

```
STMT                              SOURCE                                          CURRENT SCOPE

     E|                                                                           |
520 M|    IF ORB_NAV_ACT = ON AND MOD(FSM_PASS, ORB_NAV_CNT) = ORB_NAV_PHS THEN   | FSM_SEQ
521 M|       CALL ORB_NAV;                                                        | FSM_SEQ

     E|                                                                           |
522 M|    IF AERO_GUID_ACT = ON AND MOD(FSM_PASS, AERO_GUID_CNT) = AERO_GUID_PHS THEN | FSM_SEQ
523 M|       CALL AERO_GUID;                                                      | FSM_SEQ

     E|                                                                           |
524 M|    IF AERO_DAP_ACT = ON AND MOD(FSM_PASS, AERO_DAP_CNT) = AERO_DAP_PHS THEN | FSM_SEQ
525 M|       CALL AERO_DAP;                                                       | FSM_SEQ

526 M|    FSM_PASS = FSM_PASS + 1;                                                | FSM_SEQ
527 M| CLOSE FSM_SEQ;                                                             | FSM_SEQ

**** B L O C K   S U M M A R Y ****

EXTERNAL PROCEDURES CALLED
    ORB_NAV, AERO_GUID, AERO_DAP

COMPOOL VARIABLES USED
    ORB_NAV_ACT, ORB_NAV_CNT, ORB_NAV_PHS, AERO_GUID_ACT, AERO_GUID_CNT, AERO_GUID_PHS, AERO_DAP_ACT, AERO_DAP_CNT, AERO_DAP_PHS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
**** C O M P I L A T I O N   L A Y O U T ****
 IL_POOL: EXTERNAL COMPOOL;
 FSM_POOL: EXTERNAL COMPOOL;
 ORB_NAV: EXTERNAL PROCEDURE;
 AERO_GUID: EXTERNAL PROCEDURE;
 AERO_DAP: EXTERNAL PROCEDURE;
 FSM_SEQ: PROCEDURE;
```

```
S Y M B O L  &  C R O S S   R E F E R E N C E   T A B L E   L I S T I N G :

       (CROSS REFERENCE FLAG KEY:  4 = ASSIGNMENT, 2 = REFERENCE, 1 = SUBSCRIPT USE, 0 = DEFINITION)

 DCL  NAME                TYPE                          ATTRIBUTES & CROSS REFERENCE

 517  AERO_DAP            PROCEDURE                     EXTERNAL, VERSION=1       XREF: 0 0517  2 0525
 255  AERO_DAP_ACT        BIT(1)                        ALIGNED, INITIAL  XREF: 0 0255  2 0524
  21  AERO_DAP_CNT        INTEGER                       SINGLE, ALIGNED, INITIAL  XREF: 0 0021  2 0524
  20  AERO_DAP_PHS        INTEGER                       SINGLE, ALIGNED, INITIAL  XREF: 0 0020  2 0524
 516  AERO_GUID           PROCEDURE                     EXTERNAL, VERSION=1       XREF: 0 0516  2 0523
 256  AERO_GUID_ACT       BIT(1)                        ALIGNED, INITIAL  XREF: 0 0256  2 0522
  23  AERO_GUID_CNT       INTEGER                       SINGLE, ALIGNED, INITIAL  XREF: 0 0023  2 0522
  22  AERO_GUID_PHS       INTEGER                       SINGLE, ALIGNED, INITIAL  XREF: 0 0022  2 0522
 519  FSW_PASS            INTEGER                       DOUBLE, ALIGNED, STATIC, INITIAL  XREF: 0 0519  2 0520  2 0522  2 0524
                                                        6 0526
 518  FSW_SEQ             PROCEDURE                       XREF: 0 0518  NOT REFERENCED
 515  ORB_NAV             PROCEDURE                     EXTERNAL, VERSION=1       XREF: 0 0515  2 0521
 262  ORB_NAV_ACT         BIT(1)                        ALIGNED, INITIAL  XREF: 0 0262  2 0520
  35  ORB_NAV_CNT         INTEGER                       SINGLE, ALIGNED, INITIAL  XREF: 0 0035  2 0520
  34  ORB_NAV_PHS         INTEGER                       SINGLE, ALIGNED, INITIAL  XREF: 0 0034  2 0520
```

```
STMT                                         SOURCE                                                 CURRENT SCOPE

 523 M| ORB_NAV:                                                                                   | ORB_NAV

 523 M| PROCEDURE;                                                                                 | ORB_NAV

     C|  ----------------------------------------------------------------------                   | ORB_NAV
     C|  FUNCTION: MAINTAIN ESTIMATE OF VEHICLE STATE VECTOR AND COMPUTE                          | ORB_NAV
     C|            STATE VECTOR DERIVED PARAMETERS                                                | ORB_NAV
     C|  INPUTS:   T_ATTITUDE     - TIME TAG OF STATE VECTOR                                      | ORB_NAV
     C|            RI             - POSITION VECTOR                                               | ORB_NAV
     C|            VI             - VELOCITY VECTOR                                               | ORB_NAV
     C|            AI             - ACCELERATION VECTOR                                           | ORB_NAV
     C|            QIB            - ATTITUDE QUATERNION                                           | ORB_NAV
     C|            PHI            - BANK ANGLE                                                    | ORB_NAV
     C|  OUTPUTS:  T_NAV          - TIME TAG OF STATE VECTOR                                      | ORB_NAV
     C|            R_NAV          - POSITION VECTOR                                               | ORB_NAV
     C|            V_NAV          - VELOCITY VECTOR                                               | ORB_NAV
     C|            A_NAV          - ACCELERATION VECTOR                                           | ORB_NAV
     C|            Q_B_TO_I       - ATTITUDE QUATERNION                                           | ORB_NAV
     C|            R_NAV_MAG      - MAGNITUDE OF POSITION VECTOR                                  | ORB_NAV
     C|            UNIT_R         - UNIT VECTOR IN DIRECTION OF POSITION VECTOR                   | ORB_NAV
     C|            ALT_NAV        - ALTITUDE ABOVE FISHER ELLIPSOID                               | ORB_NAV
     C|            V_NAV_MAG      - MAGNITUDE OF VELOCITY VECTOR                                  | ORB_NAV
     C|            V_REL_NAV      - RELATIVE VELOCITY VECTOR                                      | ORB_NAV
     C|            V_REL_MAG      - MAGNITUDE OF RELATIVE VELOCITY VECTOR                         | ORB_NAV
     C|            RDOT_NAV       - RADIAL VELOCITY MAGNITUDE                                     | ORB_NAV
     C|            G_LOAD         - SENSED ACCELERATION MAGNITUDE IN G'S                          | ORB_NAV
     C|            ALPHA_NAV      - ANGLE OF ATTACK                                               | ORB_NAV
     C|            BETA_NAV       - SIDESLIP ANGLE                                                | ORB_NAV
     C|            PHI_NAV        - BANK ANGLE                                                    | ORB_NAV
     C|            ENTRY_COMPLETE - TAEM INTERFACE FLAG                                           | ORB_NAV
     C|  COMMENTS: PERFECT NAVIGATION IS ASSUMED, SO THE STATE VECTOR FROM                        | ORB_NAV
     C|            FROM THE ENVIRONMENT MODEL IS COPIED.                                          | ORB_NAV
     C|  ----------------------------------------------------------------------                   | ORB_NAV


     C|  ---------------                                                                          | ORB_NAV
     C|  LOCAL VARIABLES                                                                          | ORB_NAV
     C|  ---------------                                                                          | ORB_NAV

 524 M|    DECLARE VREL_BODY VECTOR(3) SINGLE INITIAL(0);                                         | ORB_NAV
```

```
STMT                                   SOURCE                                                CURRENT SCOPE

     C|  -----------------------------                                                      | ORB_NAV
     C|  COPY ENVIRONMENT STATE VECTOR                                                      | ORB_NAV
     C|  -----------------------------                                                      | ORB_NAV

     C|  -------------                                                                      | ORB_NAV
     C|  READ TIME TAG                                                                      | ORB_NAV
     C|  -------------                                                                      | ORB_NAV

525 M|    T_IMU_NAV = T_ATTITUDE;                                                           | ORB_NAV

     C|  -----------------                                                                  | ORB_NAV
     C|  READ STATE VECTOR                                                                  | ORB_NAV
     C|  -----------------                                                                  | ORB_NAV

     E|       -      -                                                                      |
526 M|    R_NAV = RI;                                                                       | ORB_NAV

     E|       -      -                                                                      |
527 M|    V_NAV = VI;                                                                       | ORB_NAV

     E|       -      -                                                                      |
528 M|    A_NAV = AI;                                                                       | ORB_NAV

     C|  ------------------------                                                           | ORB_NAV
     C|  READ ATTITUDE QUATERNION                                                           | ORB_NAV
     C|  ------------------------                                                           | ORB_NAV

     E|       -         -                                                                   |
529 M|    Q_B_TO_I = QIB;                                                                   | ORB_NAV

     C|  ---------------                                                                    | ORB_NAV
     C|  READ BANK ANGLE                                                                    | ORB_NAV
     C|  ---------------                                                                    | ORB_NAV

530 M|    PHI_NAV = PHI;                                                                    | ORB_NAV
```

```
STMT                                        SOURCE                                                            CURRENT SCOPE

     C| ---------------------------------------                                                              | ORB_NAV
     C| COMPUTE STATE VECTOR DERIVED PARAMETERS                                                              | ORB_NAV
     C| ---------------------------------------                                                              | ORB_NAV

531  M|   T_NAV = T_IMU_NAV;                                                                                 | ORB_NAV

     E|                          -                                                                           |
532  M|   R_NAV_MAG = ABVAL(R_NAV);                                                                          | ORB_NAV

     E|     -         -                                                                                      |
533  M|   UNIT_R = R_NAV / R_NAV_MAG;                                                                        | ORB_NAV

     E|                                                                            2                   -     |
534  M|   ALT_NAV = R_NAV_MAG - (1 - EARTH_FLAT) EARTH_R / SQRT(1 + ((1 - EARTH_FLAT)  - 1) (1 - (UNIT_R .   | ORB_NAV

     E|      -      2                                                                                        |
534  M|   EARTH_POLE) )));                                                                                   | ORB_NAV

     E|                          -                                                                           |
535  M|   V_NAV_MAG = ABVAL(V_NAV);                                                                          | ORB_NAV

     E|      -         -        -         -                                                                  |
536  M|   V_REL_NAV = V_NAV - (WE_NAV * R_NAV);                                                              | ORB_NAV

     E|                            -                                                                         |
537  M|   V_REL_MAG = ABVAL(V_REL_NAV);                                                                      | ORB_NAV

     E|                 -       -                                                                            |
538  M|   RDOT_NAV = V_NAV . UNIT_R;                                                                         | ORB_NAV

     E|                       -                                                                              |
539  M|   G_LOAD = ABVAL(A_NAV) FPS2_TO_G;                                                                   | ORB_NAV

     C| ----------------------------------                                                                   | ORB_NAV
     C| ANGLE OF ATTACK AND SIDESLIP ANGLE                                                                   | ORB_NAV
     C| ----------------------------------                                                                   | ORB_NAV

     E|      -                           -            -                                                      |
540  M|   VREL_BODY = SQFORM(SQPOSE(Q_B_TO_I), V_REL_NAV);                                                   | ORB_NAV

541  M|   ALPHA_NAV = SARCTAN2(VREL_BODY , VREL_BODY ) RAD_TO_DEG;                                           | ORB_NAV
     S|                                3          1                                                          |

542  M|   BETA_NAV = ARCSIN(VREL_BODY  / V_REL_MAG) RAD_TO_DEG;                                              | ORB_NAV
     S|                             2                                                                        |

     C| -----------------------------------------                                                            | ORB_NAV
     C| FLAG SIGNALING TAEM INTERFACE OR SKIP OUT                                                            | ORB_NAV
     C| -----------------------------------------                                                            | ORB_NAV

543  M|   IF ((ALT_NAV > ALT_EXIT) AND (RDOT_NAV > 0)) OR (ALT_NAV < ALT_TAEM) THEN                         | ORB_NAV

     E|                                                                                                      |
544  M|      AERO_BRAKE_COMPLETE = TRUE;                                                                     | ORB_NAV
```

```
STMT                                   SOURCE                                   CURRENT SCOPE

 545 M| CLOSE ORB_NAV;                                                          | ORB_NAV

**** B L O C K   S U M M A R Y ****

EXTERNAL FUNCTIONS INVOKED
    SQFORM, SQPOSE, SARCTAN2

COMPOOL VARIABLES USED
    T_IMU_NAV*, T_ATTITUDE, R_NAV*, RI, V_NAV*, VI, A_NAV*, AI, Q_B_TO_I*, QIB, PHI_NAV*, PHI, T_NAV*, T_IMU_NAV, R_NAV_MAG*, R_NAV
    UNIT_R*, R_NAV_MAG, ALT_NAV*, EARTH_FLAT, EARTH_R, UNIT_R, EARTH_POLE, V_NAV_MAG*, V_NAV, V_REL_NAV*, WE_NAV, V_REL_MAG*
    V_REL_NAV, RDOT_NAV*, G_LOAD*, A_NAV, FPS2_TO_G, Q_B_TO_I, ALPHA_NAV*, RAD_TO_DEG, BETA_NAV*, V_REL_MAG, ALT_NAV, ALT_EXIT
    RDOT_NAV, ALT_TAEM, AERO_BRAKE_COMPLETE*
```

```
**** C O M P I L A T I O N   L A Y O U T ****
ENV_POOL: EXTERNAL COMPOOL;
FSW_POOL: EXTERNAL COMPOOL;
IL_POOL: EXTERNAL COMPOOL;
DQFORM: EXTERNAL FUNCTION;
Q_ERR_ANG: EXTERNAL FUNCTION;
SARCTAN2: EXTERNAL FUNCTION;
SQFORM: EXTERNAL FUNCTION;
SQMULT: EXTERNAL FUNCTION;
SQPOSE: EXTERNAL FUNCTION;
SRV_TO_QIL: EXTERNAL FUNCTION;
ORB_NAV: PROCEDURE;
```

S Y M B O L & C R O S S R E F E R E N C E T A B L E L I S T I N G :

(CROSS REFERENCE FLAG KEY: 4 = ASSIGNMENT, 2 = REFERENCE, 1 = SUBSCRIPT USE, 0 = DEFINITION)

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
88	A_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0088 4 0528 2 0539
78	AERO_BRAKE_COMPLETE	BIT(1)	ALIGNED, INITIAL XREF: 0 0078 4 0544 NOT REFERENCED
1	AI	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0001 2 0528
75	ALPHA_NAV	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0075 4 0541 NOT REFERENCED
384	ALT_EXIT	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0384 2 0543
80	ALT_NAV	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0080 4 0534 2 0543
514	ALT_TAEM	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0514 2 0543
76	BETA_NAV	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0076 4 0542 NOT REFERENCED
518	CARG	SCALAR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0518
516	DQFORM	3 - VECTOR FUNCTION	DOUBLE, INITIAL, EXTERNAL, VERSION=2 XREF: 0 0516 NOT REFERENCED
464	EARTH_FLAT	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0464 2 0534
467	EARTH_POLE	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0467 2 0534
468	EARTH_R	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0468 2 0534
15	FPS2_TO_G	SCALAR	DOUBLE, ALIGNED, CONSTANT XREF: 0 0015 2 0019 2 0539
83	G_LOAD	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0083 4 0539 NOT REFERENCED
523	ORB_NAV	PROCEDURE	XREF: 0 0523 NOT REFERENCED
520	P	4 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0520
1	PHI	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0001 2 0530
77	PHI_NAV	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0077 4 0530 NOT REFERENCED
521	Q	4 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0521
519	Q	4 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0519
517	Q	4 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0517
520	Q	4 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0520
516	Q	4 - VECTOR	DOUBLE, ALIGNED, INPUT-PARM XREF: 0 0516
64	Q_B_TO_I	4 - VECTOR	SINGLE, ALIGNED, INITIAL XREF: 0 0064 4 0529 2 0540
517	Q_ERR_ANG	3 - VECTOR FUNCTION	SINGLE, INITIAL, EXTERNAL, VERSION=2 XREF: 0 0517 NOT REFERENCED
1	QIB	4 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0001 2 0529
522	R	3 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0522
87	R_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0087 4 0526 2 0532 2 0533 2 0536
89	R_NAV_MAG	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0089 4 0532 2 0533 2 0534
7	RAD_TO_DEG	SCALAR	DOUBLE, ALIGNED, CONSTANT XREF: 0 0007 2 0541 2 0542
90	RDOT_NAV	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0090 4 0538 2 0543
1	RI	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0001 2 0526
518	SARCTAN2	SCALAR FUNCTION	SINGLE, INITIAL, EXTERNAL, VERSION=2 XREF: 0 0518 2 0541
518	SARG	SCALAR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0518
519	SQFORM	3 - VECTOR FUNCTION	SINGLE, INITIAL, EXTERNAL, VERSION=3 XREF: 0 0519 2 0540
520	SQMULT	4 - VECTOR FUNCTION	SINGLE, INITIAL, EXTERNAL, VERSION=3 XREF: 0 0520 NOT REFERENCED
521	SQPOSE	4 - VECTOR FUNCTION	SINGLE, INITIAL, EXTERNAL, VERSION=3 XREF: 0 0521 2 0540
522	SRV_TO_QIL	4 - VECTOR FUNCTION	SINGLE, INITIAL, EXTERNAL, VERSION=2 XREF: 0 0522 NOT REFERENCED
1	T_ATTITUDE	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0001 2 0525
67	T_IMU_NAV	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0067 4 0525 2 0531
92	T_NAV	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0092 4 0531 NOT REFERENCED
516	U	3 - VECTOR	DOUBLE, ALIGNED, INPUT-PARM XREF: 0 0516
519	U	3 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0519
93	UNIT_R	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0093 4 0533 2 0534 2 0538

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
522	V	3 - VECTOR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0522
94	V_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0094 4 0527 2 0535 2 0536 2 0538
95	V_NAV_MAG	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0095 4 0535 NOT REFERENCED
96	V_REL_MAG	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0096 4 0537 2 0542
97	V_REL_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0097 4 0536 2 0537 2 0540
1	VI	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0001 2 0527
524	VREL_BODY	3 - VECTOR	SINGLE, ALIGNED, STATIC, INITIAL XREF: 0 0524 4 0540 2 0541 2 0542
481	WE_NAV	3 - VECTOR	SINGLE, ALIGNED, INITIAL XREF: 0 0481 2 0536

```
STMT                                SOURCE                                            CURRENT SCOPE

515 M| AERO_GUID:                                                                     | AERO_GUID

515 M| PROCEDURE;                                                                     | AERO_GUID

   C| ----------------------------------------------------------------                | AERO_GUID
   C| DATE:         4/1/87                                                            | AERO_GUID
   C| FUNCTION:     NUMERIC PREDICTOR/CORRECTOR ENTRY GUIDANCE ALGORITHM              | AERO_GUID
   C|               FOR THE ENTRY RESEARCH VEHICLE (ERV).                             | AERO_GUID
   C| INPUTS:       A_NAV       - SENSED INERTIAL ACCELERATION VECTOR                 | AERO_GUID
   C|               ALPHA_NAV   - ANGLE OF ATTACK                                     | AERO_GUID
   C|               ALT_NAV     - ALTITUDE ABOVE FISHER ELLIPSOID                     | AERO_GUID
   C|               G_LOAD      - SENSED ACCELERATION MAGNITUDE IN G'S                | AERO_GUID
   C|               R_NAV       - INERTIAL POSITION VECTOR                            | AERO_GUID
   C|               T_GMT       - GREENWICH MEAN TIME                                 | AERO_GUID
   C|               V_NAV       - INERTIAL VELOCITY VECTOR                            | AERO_GUID
   C|               V_NAV_MAG   - INERTIAL VELOCITY MAGNITUDE                         | AERO_GUID
   C|               V_REL_MAG   - RELATIVE VELOCITY MAGNITUDE                         | AERO_GUID
   C|               V_REL_NAV   - RELATIVE VELOCITY VECTOR                            | AERO_GUID
   C| OUTPUTS:      ALPHA_CMD   - COMMANDED ANGLE OF ATTACK                           | AERO_GUID
   C|               PHI_CMD     - COMMANDED BANK ANGLE                                | AERO_GUID
   C| DESIGNED BY:  K. SPRATLIN                                                       | AERO_GUID
   C|               C.S. DRAPER LABORATORY, INC.                                      | AERO_GUID
   C|               MAIL STOP 2B                                                      | AERO_GUID
   C|               555 TECHNOLOGY SQUARE                                             | AERO_GUID
   C|               CAMBRIDGE, MA 02139                                               | AERO_GUID
   C|               (617) 258-2441                                                    | AERO_GUID
   C| ----------------------------------------------------------------                | AERO_GUID
```

```
STMT                                SOURCE                                      CURRENT SCOPE

      C|  ---------------                                                       | AERO_GUID
      C|  LOCAL VARIABLES                                                       | AERO_GUID
      C|  ---------------                                                       | AERO_GUID

 516 M|     DECLARE ALPHA_DES SCALAR SINGLE;                                    | AERO_GUID

 517 M|     DECLARE CL_EST SCALAR SINGLE;                                       | AERO_GUID

 518 M|     DECLARE COSPHI_QDOT SCALAR SINGLE INITIAL(-1.0);                    | AERO_GUID

 519 M|     DECLARE DELTA_T_PRED SCALAR SINGLE;                                 | AERO_GUID

 520 M|     DECLARE EF_FROM_REF_AT_EPOCH MATRIX(3, 3) DOUBLE;                   | AERO_GUID

 521 M|     DECLARE GUID_PASS INTEGER SINGLE;                                   | AERO_GUID

 522 M|     DECLARE I_TARGET_EF VECTOR(3) DOUBLE;                               | AERO_GUID

 523 M|     DECLARE INITIALIZE_GUIDANCE BOOLEAN INITIAL(TRUE);                  | AERO_GUID

 524 M|     DECLARE PHI_DES SCALAR SINGLE;                                      | AERO_GUID

 525 M|     DECLARE RHO_NAV SCALAR SINGLE;                                      | AERO_GUID

      C|  --------------------------------                                      | AERO_GUID
      C|  ATMOSPHERIC PROPERTIES STRUCTURE                                      | AERO_GUID
      C|  --------------------------------                                      | AERO_GUID

 526 M|     STRUCTURE ATMOSPROP:                                                | AERO_GUID

 526 M|        1 H SCALAR SINGLE,                                               | AERO_GUID

 526 M|        1 RHO SCALAR SINGLE,                                             | AERO_GUID

 526 M|        1 TS SCALAR SINGLE,                                              | AERO_GUID

 526 M|        1 TM SCALAR SINGLE;                                              | AERO_GUID
```

```
STMT                                     SOURCE                                        CURRENT SCOPE

     C|  ----------------------------                                                  | AERO_GUID
     C|  PREDICTOR-CORRECTOR EXECUTIVE                                                 | AERO_GUID
     C|  ----------------------------                                                  | AERO_GUID

     E|         .                                                                      |
 527 M|    IF (INITIALIZE_GUIDANCE = TRUE) THEN                                        | AERO_GUID

     C|  -------------------------                                                     | AERO_GUID
     C|  FIRST-PASS INITIALIZATION                                                     | AERO_GUID
     C|  -------------------------                                                     | AERO_GUID

 528 M|       DO;                                                                      | AERO_GUID

 529 M| 1        CALL INITIAL_GUID;                                                    | AERO_GUID

     E|            .                                                                   |
 530 M| 1        INITIALIZE_GUIDANCE = FALSE;                                          | AERO_GUID

 531 M|       END;                                                                     | AERO_GUID

 532 M|    IF (G_LOAD > G_RUN_GUIDANCE) THEN                                           | AERO_GUID

     C|  ------------                                                                  | AERO_GUID
     C|  RUN GUIDANCE                                                                  | AERO_GUID
     C|  ------------                                                                  | AERO_GUID

 533 M|       DO;                                                                      | AERO_GUID

     C|  ------------------------------                                                | AERO_GUID
     C|  RUN L/D AND DENSITY ESTIMATORS                                                | AERO_GUID
     C|  ------------------------------                                                | AERO_GUID

 534 M| 1        CALL FILTERS;                                                         | AERO_GUID

     C|  ---------------------------------------                                       | AERO_GUID
     C|  RUN PREDICTOR/CORRECTOR AT CORRECT RATE                                       | AERO_GUID
     C|  ---------------------------------------                                       | AERO_GUID

 535 M| 1        IF (GUID_PASS = 0) AND (ALT_NAV > ALT_FREEZE_GUID) THEN               | AERO_GUID

 536 M| 1           CALL CORRECTOR;                                                    | AERO_GUID

     C|  ---------------------                                                         | AERO_GUID
     C|  COUNT GUIDANCE PASSES                                                         | AERO_GUID
     C|  ---------------------                                                         | AERO_GUID

 537 M| 1        GUID_PASS = GUID_PASS + 1;                                            | AERO_GUID

 538 M| 1        IF (GUID_PASS >= GUID_PASS_LIM) THEN                                  | AERO_GUID

 539 M| 1           GUID_PASS = 0;                                                     | AERO_GUID
```

```
STMT                                        SOURCE                                              CURRENT SCOPE

     C| -------------------------------------------                                             | AERO_GUID
     C| COMPUTE BANK MAGNITUDE FOR HEAT RATE CONTROL                                            | AERO_GUID
     C| -------------------------------------------                                             | AERO_GUID

 540 M| 1        CALL HEAT_RATE_CONTROL;                                                        | AERO_GUID

     C| -------------------------                                                               | AERO_GUID
     C| UPDATE COMMANDED ATTITUDE                                                               | AERO_GUID
     C| -------------------------                                                               | AERO_GUID

 541 M| 1        CALL ATTITUDE_COMMAND;                                                         | AERO_GUID

 542 M|       END;                                                                              | AERO_GUID
```

```
STMT                              SOURCE                                    CURRENT SCOPE

543 M| INITIAL_GUID:                                                       | INITIAL_GUID
543 M| PROCEDURE;                                                          | INITIAL_GUID
    C| -------------------------------------------------                   | INITIAL_GUID
    C| FUNCTION: GUIDANCE FIRST-PASS INITIALIZATION                        | INITIAL_GUID
    C| -------------------------------------------------                   | INITIAL_GUID
    C| ---------------                                                     | INITIAL_GUID
    C| LOCAL VARIABLES                                                     | INITIAL_GUID
    C| ---------------                                                     | INITIAL_GUID
544 M|   DECLARE LAT SCALAR SINGLE;                                        | INITIAL_GUID
545 M|   DECLARE LONG SCALAR SINGLE;                                       | INITIAL_GUID
546 M|   DECLARE X SCALAR SINGLE;                                          | INITIAL_GUID
547 M|   DECLARE Y SCALAR SINGLE;                                          | INITIAL_GUID
548 M|   DECLARE Z SCALAR SINGLE;                                          | INITIAL_GUID
    C| ------------------------------                                      | INITIAL_GUID
    C| ORIENTATION OF EARTH AT EPOCH                                       | INITIAL_GUID
    C| ------------------------------                                      | INITIAL_GUID
    E|           *                               *                         |
549 M|   EF_FROM_REF_AT_EPOCH = TRANSPOSE(EF_TO_REF_AT_EPOCH);             | INITIAL_GUID
    C| ----------------------------                                        | INITIAL_GUID
    C| GUIDANCE COUNT FOR PRED-CORR                                        | INITIAL_GUID
    C| ----------------------------                                        | INITIAL_GUID
550 M|   GUID_PASS = 0;                                                    | INITIAL_GUID
    C| ----------------------------                                        | INITIAL_GUID
    C| INITIAL ALPHA AND BANK GUESS                                        | INITIAL_GUID
    C| ----------------------------                                        | INITIAL_GUID
551 M|   ALPHA_DES = ALPHA_EI;                                             | INITIAL_GUID
552 M|   PHI_DES = PHI_EI;                                                 | INITIAL_GUID
    C| ------------------------                                            | INITIAL_GUID
    C| INITIAL ATTITUDE COMMAND                                            | INITIAL_GUID
    C| ------------------------                                            | INITIAL_GUID
553 M|   ALPHA_CMD = MIDVAL(ALPHA_MIN, ALPHA_EI, ALPHA_MAX);               | INITIAL_GUID
554 M|   PHI_CMD = MIDVAL((-PHI_MAX), PHI_EI, PHI_MAX);                    | INITIAL_GUID
    C| ------------------------------------------------                    | INITIAL_GUID
    C| CONVERT TARGET LONGITUDE AND LATITUDE TO RADIANS                    | INITIAL_GUID
    C| ------------------------------------------------                    | INITIAL_GUID
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
STMT                                SOURCE                                        CURRENT SCOPE

 555 M|    LONG = LONG_TARGET DEG_TO_RAD;                                          | INITIAL_GUID

 556 M|    LAT = LAT_TARGET DEG_TO_RAD;                                            | INITIAL_GUID

     C|  -----------------------------------------------                          | INITIAL_GUID
     C|  CONVERT GEODETIC LATITUDE TO GEOCENTRIC LATITUDE                          | INITIAL_GUID
     C|  -----------------------------------------------                          | INITIAL_GUID

     E|                                        2                                   |
 557 M|    LAT = ARCTAN(TAN(LAT) (1 - EARTH_FLAT) ));                              | INITIAL_GUID

     C|  ----------------------------------------------------                     | INITIAL_GUID
     C|  COMPUTE UNIT TARGET VECTOR IN EARTH-FIXED COORDINATES                     | INITIAL_GUID
     C|  ----------------------------------------------------                     | INITIAL_GUID

 558 M|    Z = SIN(LAT);                                                           | INITIAL_GUID

     E|              2                                                             |
 559 M|    X = 1 - Z ;                                                             | INITIAL_GUID

 560 M|    IF (X <= 0) THEN                                                        | INITIAL_GUID

 561 M|       X = 0;                                                               | INITIAL_GUID

 562 M|    ELSE                                                                    | INITIAL_GUID

 562 M|       X = SQRT(X) COS(LONG);                                               | INITIAL_GUID

     E|              2    2                                                        |
 563 M|    Y = 1 - X  - Z ;                                                        | INITIAL_GUID

 564 M|    IF (Y <= 0) THEN                                                        | INITIAL_GUID

 565 M|       Y = 0;                                                               | INITIAL_GUID

 566 M|    ELSE                                                                    | INITIAL_GUID

 566 M|       Y = SQRT(Y) SIGN(LONG);                                              | INITIAL_GUID

     E|        -                                                                   |
 567 M|    I_TARGET_EF = UNIT(VECTOR         (X, Y, Z));                           | INITIAL_GUID
     S|                             @DOUBLE,3                                      |

     C|  ------------------                                                        | INITIAL_GUID
     C|  INITIALIZE FILTERS                                                        | INITIAL_GUID
     C|  ------------------                                                        | INITIAL_GUID

 568 M|    K_RHO_NAV = 1.0;                                                        | INITIAL_GUID

 569 M|    K_LOD_NAV = 1.0;                                                        | INITIAL_GUID

 570 M| CLOSE INITIAL_GUID;                                                        | INITIAL_GUID

**** B L O C K   S U M M A R Y ****
```

STMT SOURCE CURRENT SCOPE

COMPOOL VARIABLES USED
EF_TO_REF_AT_EPOCH, ALPHA_EI, PHI_EI, ALPHA_CMD*, ALPHA_MIN, ALPHA_MAX, PHI_CMD*, PHI_MAX, LONG_TARGET, DEG_TO_RAD, LAT_TARGET
EARTH_FLAT, K_RHO_NAV*, K_LOD_NAV*

OUTER VARIABLES USED
EF_FROM_REF_AT_EPOCH*, GUID_PASS*, ALPHA_DES*, PHI_DES*, I_TARGET_EF*

```
STMT                                    SOURCE                                                  CURRENT SCOPE
 571 M| FILTERS:                                                                                | FILTERS
 571 M| PROCEDURE;                                                                              | FILTERS
     C|  -----------------------------------------------------                                  | FILTERS
     C|  FUNCTION: IN-FLIGHT ACCELERATION MEASUREMENT FILTERING                                 | FILTERS
     C|  -----------------------------------------------------                                  | FILTERS
     C|  ---------------                                                                        | FILTERS
     C|  LOCAL VARIABLES                                                                        | FILTERS
     C|  ---------------                                                                        | FILTERS
 572 M|    DECLARE A_DRAG_MAG SCALAR SINGLE;                                                    | FILTERS
 573 M|    DECLARE A_LIFT_MAG SCALAR SINGLE;                                                    | FILTERS
 574 M|    DECLARE ATMOS ATMOSPROP-STRUCTURE;                                                   | FILTERS
 575 M|    DECLARE CD_NOM SCALAR SINGLE;                                                        | FILTERS
 576 M|    DECLARE CL_NOM SCALAR SINGLE;                                                        | FILTERS
 577 M|    DECLARE LOD_MEAS SCALAR SINGLE;                                                      | FILTERS
 578 M|    DECLARE LOD_NOM SCALAR SINGLE;                                                       | FILTERS
 579 M|    DECLARE MACH SCALAR SINGLE;                                                          | FILTERS
 580 M|    DECLARE RHO_MEAS SCALAR SINGLE;                                                      | FILTERS
 581 M|    DECLARE V_BAR SCALAR SINGLE;                                                         | FILTERS
     C|  ---------------------------------                                                      | FILTERS
     C|  LOOK UP OF NOMINAL DENSITY AND L/D                                                     | FILTERS
     C|  ---------------------------------                                                      | FILTERS
     E|                     -          -                  +                                     |
 582 M|    CALL USATMOS62(R_NAV, EARTH_POLE) ASSIGN(ATMOS);                                     | FILTERS
     E|                                   +                                                     |
 583 M|    CALL AERO_PARAMETERS(V_REL_MAG, ATMOS) ASSIGN(V_BAR, MACH);                          | FILTERS
 584 M|    CALL LOOKUP(ALPHA_NAV, ATMOS.H, V_BAR, MACH) ASSIGN(CL_NOM, CD_NOM);                 | FILTERS
 585 M|    LOD_NOM = CL_NOM / CD_NOM;                                                           | FILTERS
     C|  ---------------------------------                                                      | FILTERS
     C|  COMPUTE DRAG AND LIFT ACCELERATION                                                     | FILTERS
     C|  ---------------------------------                                                      | FILTERS
     E|                       -         -                                                       |
 586 M|    A_DRAG_MAG = -(A_NAV . V_REL_NAV) / V_REL_MAG;                                       | FILTERS
```

```
STMT                                        SOURCE                                                    CURRENT SCOPE

    E|                        -         -                                                           |
587 M|    A_LIFT_MAG = SQRT(A_NAV . A_NAV - A_DRAG_MAG A_DRAG_MAG);                                 | FILTERS

    C| --------------------------------                                                             | FILTERS
    C| COMPUTE MEASURED L/D AND DENSITY                                                             | FILTERS
    C| --------------------------------                                                             | FILTERS

588 M|    LOD_MEAS = A_LIFT_MAG / A_DRAG_MAG;                                                       | FILTERS

    E|                                                                    2                         |
589 M|    RHO_MEAS = 2 A_DRAG_MAG MASS_NAV / (CD_NOM S_REF V_REL_MAG );                             | FILTERS

    C| --------------------------------------                                                       | FILTERS
    C| FILTER MEASURED L/D AND DENSITY RATIOS                                                       | FILTERS
    C| --------------------------------------                                                       | FILTERS

590 M|    K_LOD_NAV = (1 - L_OVER_D_FILTER_GAIN) K_LOD_NAV + L_OVER_D_FILTER_GAIN (LOD_MEAS / LOD_NOM);  | FILTERS

591 M|    K_RHO_NAV = (1 - K_RHO_FILTER_GAIN) K_RHO_NAV + K_RHO_FILTER_GAIN (RHO_MEAS / ATMOS.RHO);  | FILTERS

    C| ---------------------------------                                                            | FILTERS
    C| COMPUTE FILTERED MEASURED DENSITY                                                            | FILTERS
    C| ---------------------------------                                                            | FILTERS

592 M|    RHO_NAV = K_RHO_NAV ATMOS.RHO;                                                            | FILTERS

    C| -----------------------------                                                                | FILTERS
    C| COMPUTE FILTERED ESTIMATED CL                                                                | FILTERS
    C| -----------------------------                                                                | FILTERS

593 M|    CL_EST = K_LOD_NAV CL_NOM;                                                                | FILTERS

594 M| CLOSE FILTERS;                                                                               | FILTERS

**** B L O C K   S U M M A R Y ****

OUTER PROCEDURES CALLED
    LOOKUP, AERO_PARAMETERS, USATMOS62

COMPOOL VARIABLES USED
    R_NAV, EARTH_POLE, V_REL_MAG, ALPHA_NAV, A_NAV, V_REL_NAV, MASS_NAV, S_REF, K_LOD_NAV*, L_OVER_D_FILTER_GAIN, K_LOD_NAV
    K_RHO_NAV*, K_RHO_FILTER_GAIN, K_RHO_NAV

OUTER VARIABLES USED
    RHO_NAV*, CL_EST*

OUTER STRUCTURE TEMPLATES USED
     ATMOSPROP
```

```
STMT                                 SOURCE                                                     CURRENT SCOPE
 595 M| HEAT_RATE_CONTROL:                                                                    | HEAT_RATE_CONTROL
 595 M| PROCEDURE;                                                                            | HEAT_RATE_CONTROL
     C|  --------------------------------                                                     | HEAT_RATE_CONTROL
     C|  FUNCTION: CONTROL PEAK HEAT RATE                                                     | HEAT_RATE_CONTROL
     C|  --------------------------------                                                     | HEAT_RATE_CONTROL
     C|  ---------------                                                                      | HEAT_RATE_CONTROL
     C|  LOCAL VARIABLES                                                                      | HEAT_RATE_CONTROL
     C|  ---------------                                                                      | HEAT_RATE_CONTROL
 596 M|    DECLARE COSPHI_1 SCALAR SINGLE;                                                    | HEAT_RATE_CONTROL
 597 M|    DECLARE COSPHI_2 SCALAR SINGLE;                                                    | HEAT_RATE_CONTROL
 598 M|    DECLARE FIRST_PASS BOOLEAN INITIAL(TRUE);                                          | HEAT_RATE_CONTROL
 599 M|    DECLARE HS_2 SCALAR SINGLE;                                                        | HEAT_RATE_CONTROL
 600 M|    DECLARE K_QDOT SCALAR SINGLE;                                                      | HEAT_RATE_CONTROL
 601 M|    DECLARE K_QDOT_RATE SCALAR SINGLE;                                                 | HEAT_RATE_CONTROL
 602 M|    DECLARE K1_GAIN SCALAR SINGLE;                                                     | HEAT_RATE_CONTROL
 603 M|    DECLARE K1_QDOT SCALAR SINGLE;                                                     | HEAT_RATE_CONTROL
 604 M|    DECLARE OMEGA_QDOT_SQUARED SCALAR SINGLE;                                          | HEAT_RATE_CONTROL
 605 M|    DECLARE QBAR SCALAR SINGLE;                                                        | HEAT_RATE_CONTROL
 606 M|    DECLARE QDOT SCALAR SINGLE;                                                        | HEAT_RATE_CONTROL
 607 M|    DECLARE QDOT_PAST SCALAR SINGLE;                                                   | HEAT_RATE_CONTROL
 608 M|    DECLARE QDOT_RATE SCALAR SINGLE;                                                   | HEAT_RATE_CONTROL
 609 M|    DECLARE TWO_ZETA_OMEGA SCALAR SINGLE;                                              | HEAT_RATE_CONTROL

     E|                                                    3.05                               |
 610 M|    QDOT = 17700.0 SQRT(RHO_NAV) (V_REL_MAG / 10000)     ;                             | HEAT_RATE_CONTROL
     E|                                                                                       |
 611 M|    IF (FIRST_PASS = TRUE) THEN                                                        | HEAT_RATE_CONTROL
 612 M|       DO;                                                                             | HEAT_RATE_CONTROL
     E|                                                                                       |
 613 M| 1         FIRST_PASS = FALSE;                                                         | HEAT_RATE_CONTROL
     C|  ----------------------------                                                         | HEAT_RATE_CONTROL
     C|  HEAT RATE CONTROL CONSTANTS                                                          | HEAT_RATE_CONTROL
```

```
STMT                                SOURCE                                              CURRENT SCOPE
      C| ---------------------------                                                    | HEAT_RATE_CONTROL
 614 M| 1        HS_2 = 2 HS;                                                            | HEAT_RATE_CONTROL
 615 M| 1        K1_GAIN = S_REF SQRT(RHO_SL) 17700.0 / (HS_2 MASS_NAV);                  | HEAT_RATE_CONTROL
 616 M| 1        OMEGA_QDOT_SQUARED = OMEGA_QDOT OMEGA_QDOT;                              | HEAT_RATE_CONTROL
 617 M| 1        TWO_ZETA_OMEGA = 2 ZETA_QDOT OMEGA_QDOT;                                 | HEAT_RATE_CONTROL
      C| -------------------------                                                      | HEAT_RATE_CONTROL
      C| FIRST PASS INITIALIZATION                                                      | HEAT_RATE_CONTROL
      C| -------------------------                                                      | HEAT_RATE_CONTROL
 618 M| 1        QDOT_RATE = 0;                                                          | HEAT_RATE_CONTROL
 619 M|       END;                                                                      | HEAT_RATE_CONTROL
 620 M|    ELSE                                                                         | HEAT_RATE_CONTROL
 620 M|       QDOT_RATE = (QDOT - QDOT_PAST) / DT_AEROGUID;                              | HEAT_RATE_CONTROL
 621 M|    QDOT_PAST = QDOT;                                                            | HEAT_RATE_CONTROL
 622 M|    IF (QDOT > .5 QDOT_LIMIT) THEN                                               | HEAT_RATE_CONTROL
 623 M|       DO;                                                                       | HEAT_RATE_CONTROL
      C| --------------------------                                                     | HEAT_RATE_CONTROL
      C| ESTIMATED DYNAMIC PRESSURE                                                     | HEAT_RATE_CONTROL
      C| --------------------------                                                     | HEAT_RATE_CONTROL
 624 M| 1        QBAR = .5 RHO_NAV V_REL_MAG V_REL_MAG;                                  | HEAT_RATE_CONTROL
      C| ------------------------                                                       | HEAT_RATE_CONTROL
      C| HEAT RATE CONTROL GAINS                                                        | HEAT_RATE_CONTROL
      C| ------------------------                                                       | HEAT_RATE_CONTROL
      E|                                                   3.05                         |
 625 M| 1        K1_QDOT = K1_GAIN CL_EST (V_REL_MAG / 10000.0)     EXP(-(ALT_NAV / HS_2));| HEAT_RATE_CONTROL
 626 M| 1        K_QDOT = OMEGA_QDOT_SQUARED / K1_QDOT;                                  | HEAT_RATE_CONTROL
 627 M| 1        K_QDOT_RATE = TWO_ZETA_OMEGA / K1_QDOT;                                 | HEAT_RATE_CONTROL
      C| --------------------------                                                     | HEAT_RATE_CONTROL
      C| HEAT RATE CONTROL EQUATION                                                     | HEAT_RATE_CONTROL
      C| --------------------------                                                     | HEAT_RATE_CONTROL
 628 M| 1        COSPHI_1 = K_QDOT_RATE QDOT_RATE / QBAR;                                | HEAT_RATE_CONTROL
 629 M| 1        COSPHI_2 = K_QDOT (QDOT - QDOT_LIMIT) / QBAR;                           | HEAT_RATE_CONTROL
 630 M| 1        COSPHI_QDOT = COSPHI_1 + COSPHI_2;                                      | HEAT_RATE_CONTROL
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
STMT                                  SOURCE                                    CURRENT SCOPE

 631 M|       END;                                                              | HEAT_RATE_CONTROL
 632 M|    ELSE                                                                 | HEAT_RATE_CONTROL
 632 M|       COSPHI_QDOT = -1.0;                                               | HEAT_RATE_CONTROL
 633 M| CLOSE HEAT_RATE_CONTROL;                                                | HEAT_RATE_CONTROL

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    V_REL_MAG, HS, S_REF, RHO_SL, MASS_NAV, OMEGA_QDOT, ZETA_QDOT, DT_AEROGUID, QDOT_LIMIT, ALT_NAV

OUTER VARIABLES USED
    RHO_NAV, CL_EST, COSPHI_QDOT*
```

```
STMT                                    SOURCE                                              CURRENT SCOPE

 634 M| ATTITUDE_COMMAND:                                                                  | ATTITUDE_COMMAND
 634 M| PROCEDURE;                                                                         | ATTITUDE_COMMAND
     C|  ------------------                                                                | ATTITUDE_COMMAND
     C|  LOAD ALPHA COMMAND                                                                | ATTITUDE_COMMAND
     C|  ------------------                                                                | ATTITUDE_COMMAND
 635 M|    ALPHA_CMD = ALPHA_DES;                                                          | ATTITUDE_COMMAND
     C|  -----------------------------                                                     | ATTITUDE_COMMAND
     C|  COMPUTE AND LIMIT PHI COMMAND                                                     | ATTITUDE_COMMAND
     C|  -----------------------------                                                     | ATTITUDE_COMMAND
 636 M|    PHI_CMD = PHI_DES ABS(V_NAV_MAG - V_FINAL_MAG) / V_MAG_CHANGE;                  | ATTITUDE_COMMAND
 637 M|    PHI_CMD = MIDVAL((-PHI_MAX), PHI_CMD, PHI_MAX);                                 | ATTITUDE_COMMAND
     C|  ------------------------------------------------                                  | ATTITUDE_COMMAND
     C|  ADJUST COMMANDED BANK ANGLE FOR HEAT RATE CONTROL                                 | ATTITUDE_COMMAND
     C|  ------------------------------------------------                                  | ATTITUDE_COMMAND
 638 M|    IF (COSPHI_QDOT > 0) THEN                                                       | ATTITUDE_COMMAND
 639 M|       DO;                                                                          | ATTITUDE_COMMAND
 640 M| 1        TEMPORARY COSPHI_CMD SCALAR SINGLE;                                       | ATTITUDE_COMMAND
 641 M| 1        COSPHI_CMD = COS(PHI_CMD DEG_TO_RAD) + COSPHI_QDOT;                       | ATTITUDE_COMMAND
 642 M| 1        COSPHI_CMD = MIDVAL(-1.0, COSPHI_CMD, +1.0);                              | ATTITUDE_COMMAND
 643 M| 1        PHI_CMD = SIGN(PHI_CMD) ARCCOS(COSPHI_CMD) RAD_TO_DEG;                    | ATTITUDE_COMMAND
 644 M|       END;                                                                         | ATTITUDE_COMMAND
     C|  ----------------------------                                                      | ATTITUDE_COMMAND
     C|  LIMIT ALPHA AND PHI COMMANDS                                                      | ATTITUDE_COMMAND
     C|  ----------------------------                                                      | ATTITUDE_COMMAND
 645 M|    ALPHA_CMD = MIDVAL(ALPHA_MIN, ALPHA_CMD, ALPHA_MAX);                            | ATTITUDE_COMMAND
 646 M|    PHI_CMD = MIDVAL((-PHI_MAX), PHI_CMD, PHI_MAX);                                 | ATTITUDE_COMMAND
 647 M| CLOSE ATTITUDE_COMMAND;                                                            | ATTITUDE_COMMAND

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    ALPHA_CMD*, PHI_CMD*, V_NAV_MAG, V_FINAL_MAG, V_MAG_CHANGE, PHI_MAX, PHI_CMD, DEG_TO_RAD, RAD_TO_DEG, ALPHA_MIN, ALPHA_CMD
    ALPHA_MAX

OUTER VARIABLES USED
    ALPHA_DES, PHI_DES, COSPHI_QDOT
```

ORIGINAL PAGE IS OF POOR QUALITY

```
STMT                                  SOURCE                                        CURRENT SCOPE
 648 M| CORRECTOR:                                                                  | CORRECTOR
 648 M| PROCEDURE;                                                                  | CORRECTOR
     C|  ---------------------------------------                                    | CORRECTOR
     C|  FUNCTION: PREDICTOR/CORRECTOR SEQUENCER                                    | CORRECTOR
     C|  ---------------------------------------                                    | CORRECTOR
     C|  ---------------                                                            | CORRECTOR
     C|  LOCAL VARIABLES                                                            | CORRECTOR
     C|  ---------------                                                            | CORRECTOR
 649 M|    DECLARE ALPHA_TRY SCALAR SINGLE;                                         | CORRECTOR
 650 M|    DECLARE DELTA_ALPHA SCALAR SINGLE CONSTANT(3);                           | CORRECTOR
 651 M|    DECLARE DELTA_PHI SCALAR SINGLE CONSTANT(3);                             | CORRECTOR
 652 M|    DECLARE DETERM SCALAR SINGLE INITIAL(0);                                 | CORRECTOR
 653 M|    DECLARE DDRE_DA SCALAR SINGLE;                                           | CORRECTOR
 654 M|    DECLARE DDRE_DP SCALAR SINGLE;                                           | CORRECTOR
 655 M|    DECLARE DCRE_DA SCALAR SINGLE;                                           | CORRECTOR
 656 M|    DECLARE DCRE_DP SCALAR SINGLE;                                           | CORRECTOR
 657 M|    DECLARE CRE SCALAR SINGLE;                                               | CORRECTOR
 658 M|    DECLARE CR_ERROR ARRAY(3) SCALAR SINGLE;                                 | CORRECTOR
 659 M|    DECLARE CR_ERR SCALAR SINGLE;                                            | CORRECTOR
 660 M|    DECLARE PHI_TRY SCALAR SINGLE;                                           | CORRECTOR
 661 M|    DECLARE PRED_EXIT BOOLEAN;                                               | CORRECTOR
 662 M|    DECLARE DR_ERROR ARRAY(3) SCALAR SINGLE;                                 | CORRECTOR
 663 M|    DECLARE DR_ERR SCALAR SINGLE;                                            | CORRECTOR
 664 M|    DECLARE DRE SCALAR SINGLE;                                               | CORRECTOR

 665 M|    DO FOR TEMPORARY I = 1 TO 3;                                             | CORRECTOR
 666 M| 1     DO CASE I;                                                            | CORRECTOR
 667 M| 2        DO;                                                                | CORRECTOR CASE 1
 668 M| 3           ALPHA_TRY = ALPHA_DES;                                          | CORRECTOR
 669 M| 3           PHI_TRY = PHI_DES;                                              | CORRECTOR
```

```
STMT                                SOURCE                                      CURRENT SCOPE
 670 M| 2        END;                                                           | CORRECTOR
 671 M| 2        DO;                                                            | CORRECTOR CASE 2
 672 M| 3           ALPHA_TRY = ALPHA_DES + DELTA_ALPHA;                        | CORRECTOR
 673 M| 3           PHI_TRY = PHI_DES;                                          | CORRECTOR
 674 M| 2        END;                                                           | CORRECTOR
 675 M| 2        DO;                                                            | CORRECTOR CASE 3
 676 M| 3           ALPHA_TRY = ALPHA_DES;                                      | CORRECTOR
 677 M| 3           PHI_TRY = PHI_DES + DELTA_PHI;                              | CORRECTOR
 678 M| 2        END;                                                           | CORRECTOR
 679 M| 1     END;                                                              | CORRECTOR DO CASE END
     C|  -------------------------------------------                            | CORRECTOR
     C|  CALL PREDICTOR WITH DESIRED CONTROL HISTORY                            | CORRECTOR
     C|  -------------------------------------------                            | CORRECTOR
 680 M| 1     CALL PREDICTOR;                                                   | CORRECTOR
     C|  ------------------------                                               | CORRECTOR
     C|  STORE FINAL STATE ERRORS                                               | CORRECTOR
     C|  ------------------------                                               | CORRECTOR
 681 M| 1     DR_ERROR  = DR_ERR;                                               | CORRECTOR
     S|               I                                                         |
 682 M| 1     CR_ERROR  = CR_ERR;                                               | CORRECTOR
     S|               I                                                         |
 683 M|    END;                                                                 | CORRECTOR
     C|  ----------------                                                       | CORRECTOR
     C|  COMPUTE PARTIALS                                                       | CORRECTOR
     C|  ----------------                                                       | CORRECTOR
 684 M|    DDRE_DA = (DR_ERROR  - DR_ERROR ) / DELTA_ALPHA;                     | CORRECTOR
     S|                       2           1                                     |
 685 M|    DDRE_DP = (DR_ERROR  - DR_ERROR ) / DELTA_PHI;                       | CORRECTOR
     S|                       3           1                                     |
 686 M|    DCRE_DA = (CR_ERROR  - CR_ERROR ) / DELTA_ALPHA;                     | CORRECTOR
     S|                       2           1                                     |
 687 M|    DCRE_DP = (CR_ERROR  - CR_ERROR ) / DELTA_PHI;                       | CORRECTOR
     S|                       3           1                                     |
```

```
STMT                                  SOURCE                                              CURRENT SCOPE
 688 M|    DRE = DR_ERROR ;                                                               | CORRECTOR
     S|                1                                                                  |
 689 M|    CRE = CR_ERROR ;                                                               | CORRECTOR
     S|                1                                                                  |
     C|  -------------------------------------                                            | CORRECTOR
     C|  SOLVE SET OF 2 SIMULTANEOUS EQUATIONS                                            | CORRECTOR
     C|  -------------------------------------                                            | CORRECTOR
 690 M|    DETERM = DCRE_DP DDRE_DA - DDRE_DP DCRE_DA;                                    | CORRECTOR
 691 M|    IF (DETERM ¬= 0) THEN                                                          | CORRECTOR
 692 M|       DO;                                                                         | CORRECTOR
 693 M| 1        ALPHA_DES = ALPHA_DES + (DDRE_DP CRE - DCRE_DP DRE) / DETERM;            | CORRECTOR
 694 M| 1        ALPHA_DES = MIDVAL(ALPHA_MIN, ALPHA_DES, ALPHA_MAX);                     | CORRECTOR
 695 M| 1        PHI_DES = PHI_DES - (DDRE_DA CRE - DCRE_DA DRE) / DETERM;                | CORRECTOR
 696 M| 1        PHI_DES = MIDVAL((-PHI_DES_MAX), PHI_DES, PHI_DES_MAX);                  | CORRECTOR
 697 M|       END;                                                                        | CORRECTOR
```

```
STMT                                  SOURCE                                              CURRENT SCOPE
 698 M| PREDICTOR:                                                                         | PREDICTOR
 698 M| PROCEDURE;                                                                         | PREDICTOR
     C|  -------------------------------------                                             | PREDICTOR
     C|  FUNCTION: NUMERIC PREDICTOR ALGORITHM                                             | PREDICTOR
     C|  -------------------------------------                                             | PREDICTOR
     C|  --------------                                                                    | PREDICTOR
     C|  LOCAL FUNCTION                                                                    | PREDICTOR
     C|  --------------                                                                    | PREDICTOR
 699 M|    DECLARE EARTH_FIXED_FROM_REFERENCE FUNCTION MATRIX(3, 3) DOUBLE;                | PREDICTOR
     C|  ---------------                                                                   | PREDICTOR
     C|  LOCAL VARIABLES                                                                   | PREDICTOR
     C|  ---------------                                                                   | PREDICTOR
 700 M|    DECLARE TOTAL_TIME_STEPS INTEGER SINGLE;                                        | PREDICTOR
 701 M|    DECLARE A_PRED VECTOR(3) DOUBLE;                                                | PREDICTOR
 702 M|    DECLARE A_PRED_MAG SCALAR SINGLE;                                               | PREDICTOR
 703 M|    DECLARE ALPHA_PRED SCALAR SINGLE;                                               | PREDICTOR
 704 M|    DECLARE ATMOS ATMOSPROP-STRUCTURE;                                              | PREDICTOR
 705 M|    DECLARE CD_PRED SCALAR SINGLE;                                                  | PREDICTOR
 706 M|    DECLARE CL_PRED SCALAR SINGLE;                                                  | PREDICTOR
 707 M|    DECLARE DOT SCALAR DOUBLE;                                                      | PREDICTOR
 708 M|    DECLARE EF_FROM_REF MATRIX(3, 3) DOUBLE;                                        | PREDICTOR
 709 M|    DECLARE I_INPLANE VECTOR(3) DOUBLE;                                             | PREDICTOR
 710 M|    DECLARE I_NORMAL VECTOR(3) DOUBLE;                                              | PREDICTOR
 711 M|    DECLARE INTEG_LOOP SCALAR SINGLE;                                               | PREDICTOR
 712 M|    DECLARE IR_E VECTOR(3) DOUBLE;                                                  | PREDICTOR
 713 M|    DECLARE LOD_PRED SCALAR SINGLE;                                                 | PREDICTOR
 714 M|    DECLARE MACH_PRED SCALAR SINGLE;                                                | PREDICTOR
 715 M|    DECLARE PHI_PRED SCALAR SINGLE;                                                 | PREDICTOR
 716 M|    DECLARE R_PRED VECTOR(3) DOUBLE;                                                | PREDICTOR
 717 M|    DECLARE R_MAG_PRED SCALAR DOUBLE;                                               | PREDICTOR
 718 M|    DECLARE RDOT_PRED SCALAR SINGLE;                                                | PREDICTOR
```

```
STMT                                   SOURCE                                            CURRENT SCOPE
 719 M|    DECLARE T_PRED SCALAR DOUBLE;                                                 | PREDICTOR
 720 M|    DECLARE V_BAR_PRED SCALAR SINGLE;                                             | PREDICTOR
 721 M|    DECLARE V_MAG_PRED SCALAR DOUBLE;                                             | PREDICTOR
 722 M|    DECLARE V_PRED VECTOR(3) DOUBLE;                                              | PREDICTOR
 723 M|    DECLARE V_REL_MAG_PRED SCALAR SINGLE;                                         | PREDICTOR
 724 M|    DECLARE V_REL_PRED VECTOR(3) SINGLE;                                          | PREDICTOR
 725 M|    DECLARE VR_E VECTOR(3) DOUBLE;                                                | PREDICTOR

     C|  ----------------------------------                                              | PREDICTOR
     C|  INITIALIZE PREDICTOR STATE VECTOR                                               | PREDICTOR
     C|  ----------------------------------                                              | PREDICTOR

 726 M|    T_PRED = T_GMT;                                                               | PREDICTOR

     E|      -         -                                                                 |
 727 M|    R_PRED = R_NAV;                                                               | PREDICTOR

     E|                          -                                                       |
 728 M|    R_MAG_PRED = ABVAL(R_PRED);                                                   | PREDICTOR

     E|      -         -                                                                 |
 729 M|    V_PRED = V_NAV;                                                               | PREDICTOR

     E|                          -                                                       |
 730 M|    V_MAG_PRED = ABVAL(V_PRED);                                                   | PREDICTOR

     E|          -          -          -         -                                       |
 731 M|    V_REL_PRED = V_PRED - (WE_NAV * R_PRED);                                      | PREDICTOR

     E|                                -                                                 |
 732 M|    V_REL_MAG_PRED = ABVAL(V_REL_PRED);                                           | PREDICTOR

     C|  ------------------------------                                                  | PREDICTOR
     C|  ANGLE OF ATTACK FOR PREDICTION                                                  | PREDICTOR
     C|  ------------------------------                                                  | PREDICTOR

 733 M|    ALPHA_PRED = ALPHA_TRY;                                                       | PREDICTOR

     C|  ----------------------------------------                                        | PREDICTOR
     C|  INITIALIZE 1962 U.S. STANDARD ATMOSPHERE                                        | PREDICTOR
     C|  ----------------------------------------                                        | PREDICTOR

     E|                    -          -               +                                  |
 734 M|    CALL USATMOS62(R_PRED, EARTH_POLE) ASSIGN(ATMOS);                             | PREDICTOR

     C|  -------------------------------------------                                     | PREDICTOR
     C|  FORCE 1ST TIME STEP TO BE MINIMUM TIME STEP                                     | PREDICTOR
     C|  -------------------------------------------                                     | PREDICTOR
```

```
STMT                                SOURCE                                          CURRENT SCOPE

     E|      -                                                                     |
 735 M|    A_PRED = VECTOR        (999999., 999999., 999999.);                     | PREDICTOR
     S|                   @SINGLE,3                                                |

     C|  --------------                                                            | PREDICTOR
     C|  PREDICTOR LOOP                                                            | PREDICTOR
     C|  --------------                                                            | PREDICTOR

 736 M|    DO FOR TEMPORARY TIME_INCREMENT = 1 TO 5000;                            | PREDICTOR

     C|  -------------------------------------------------------                   | PREDICTOR
     C|  COMPUTE TIME STEP FOR 4TH ORDER RUNGA-KUTTA INTEGRATION                   | PREDICTOR
     C|  -------------------------------------------------------                   | PREDICTOR

 737 M| 1     TOTAL_TIME_STEPS = TIME_INCREMENT;                                   | PREDICTOR

 738 M| 1     IF (ATMOS.H <= ALT_TAEM_BIAS) THEN                                   | PREDICTOR

 739 M| 1        DELTA_T_PRED = DELTA_T_PRED_MIN;                                  | PREDICTOR

 740 M| 1     ELSE                                                                 | PREDICTOR

 740 M| 1        DO;                                                               | PREDICTOR

     E|                                -                                           |
 741 M| 2           A_PRED_MAG = ABVAL(A_PRED);                                    | PREDICTOR

 742 M| 2           IF (A_PRED_MAG ¬= 0) THEN                                      | PREDICTOR

 743 M| 2              DELTA_T_PRED = DELTA_T_PRED_GAIN / A_PRED_MAG;              | PREDICTOR

 744 M| 2           ELSE                                                           | PREDICTOR

 744 M| 2              DELTA_T_PRED = DELTA_T_PRED_MAX;                            | PREDICTOR

 745 M| 2           DELTA_T_PRED = MIDVAL(DELTA_T_PRED_MIN, DELTA_T_PRED, DELTA_T_PRED_MAX);   | PREDICTOR

 746 M| 1        END;                                                              | PREDICTOR

     C|  ------------------------------                                            | PREDICTOR
     C|  AERODYNAMIC PROPERTIES LOOK-UP                                            | PREDICTOR
     C|  ------------------------------                                            | PREDICTOR

     E|                                       +                                    |
 747 M| 1     CALL AERO_PARAMETERS(V_REL_MAG_PRED, ATMOS) ASSIGN(V_BAR_PRED, MACH_PRED);   | PREDICTOR

 748 M| 1     CALL LOOKUP(ALPHA_PRED, ATMOS.H, V_BAR_PRED, MACH_PRED) ASSIGN(CL_PRED, CD_PRED);   | PREDICTOR

     C|  -------------                                                             | PREDICTOR
     C|  ESTIMATED L/D                                                             | PREDICTOR
     C|  -------------                                                             | PREDICTOR

 749 M| 1     LOD_PRED = K_LOD_NAV CL_PRED / CD_PRED;                              | PREDICTOR
```

ORIGINAL PAGE
COLOR PHOTOGRAPH

```
STMT                                        SOURCE                                           CURRENT SCOPE

      C|  ---------------------------------                                                  | PREDICTOR
      C|  EQUATIONS OF MOTION FOR ERV ENTRY                                                  | PREDICTOR
      C|  ---------------------------------                                                  | PREDICTOR

 750 M| 1      DO FOR INTEG_LOOP = 1 TO 4;                                                  | PREDICTOR

 751 M| 2         TEMPORARY AERO_ACCEL VECTOR(3) SINGLE;                                     | PREDICTOR

 752 M| 2         TEMPORARY CPHI SCALAR SINGLE;                                              | PREDICTOR

 753 M| 2         TEMPORARY DRAG_ACCEL SCALAR SINGLE;                                        | PREDICTOR

 754 M| 2         TEMPORARY GRAV_ACCEL VECTOR(3) DOUBLE;                                     | PREDICTOR

 755 M| 2         TEMPORARY HS_NORM_PRED SCALAR SINGLE;                                      | PREDICTOR

 756 M| 2         TEMPORARY I_LAT VECTOR(3) SINGLE;                                          | PREDICTOR

 757 M| 2         TEMPORARY I_LIFT VECTOR(3) SINGLE;                                         | PREDICTOR

 758 M| 2         TEMPORARY I_VEL VECTOR(3) SINGLE;                                          | PREDICTOR

 759 M| 2         TEMPORARY LIFT_ACCEL SCALAR SINGLE;                                        | PREDICTOR

 760 M| 2         TEMPORARY RHO_PRED SCALAR SINGLE;                                          | PREDICTOR

 761 M| 2         TEMPORARY SPHI SCALAR SINGLE;                                              | PREDICTOR

 762 M| 2         TEMPORARY U_PRED VECTOR(3) DOUBLE;                                         | PREDICTOR

 763 M| 2         TEMPORARY Z_PRED SCALAR DOUBLE;                                            | PREDICTOR


      C|  -----------------                                                                  | PREDICTOR
      C|  ESTIMATED DENSITY                                                                  | PREDICTOR
      C|  -----------------                                                                  | PREDICTOR

 764 M| 2         RHO_PRED = K_RHO_NAV ATMOS.RHO;                                            | PREDICTOR

      C|  ----------------                                                                   | PREDICTOR
      C|  BANK ANGLE MODEL                                                                   | PREDICTOR
      C|  ----------------                                                                   | PREDICTOR

 765 M| 2         PHI_PRED = PHI_TRY ABS(V_MAG_PRED - V_FINAL_MAG) / V_MAG_CHANGE;           | PREDICTOR

 766 M| 2         PHI_PRED = MIDVAL((-PHI_MAX), PHI_PRED, PHI_MAX) DEG_TO_RAD;               | PREDICTOR

 767 M| 2         CPHI = COS(PHI_PRED);                                                      | PREDICTOR

 768 M| 2         SPHI = SIN(PHI_PRED);                                                      | PREDICTOR

      C|  ------------------------                                                           | PREDICTOR
      C|  AERODYNAMIC ACCELERATION                                                           | PREDICTOR
      C|  ------------------------                                                           | PREDICTOR
```

```
STMT                                      SOURCE                                                    CURRENT SCOPE
     E|                                             2                                                |
 769 M| 2        DRAG_ACCEL = .5 RHO_PRED V_REL_MAG_PRED  CD_PRED S_REF / MASS_NAV;                  | PREDICTOR
 770 M| 2        LIFT_ACCEL = LOD_PRED DRAG_ACCEL;                                                   | PREDICTOR
     E|           -           -                                                                      |
 771 M| 2        I_VEL = V_REL_PRED / V_REL_MAG_PRED;                                                | PREDICTOR
     E|           -               -        -                                                         |
 772 M| 2        I_LAT = UNIT(I_VEL * R_PRED);                                                       | PREDICTOR
     E|           -                -        -        -                                               |
 773 M| 2        I_LIFT = UNIT(I_LAT * I_VEL) CPHI + I_LAT SPHI;                                     | PREDICTOR
     E|             -                     -                 -                                        |
 774 M| 2        AERO_ACCEL = LIFT_ACCEL I_LIFT - DRAG_ACCEL I_VEL;                                  | PREDICTOR
     C|  ---------------------------------                                                           | PREDICTOR
     C|  GRAVITY ACCELERATION WITH J2 TERM                                                           | PREDICTOR
     C|  ---------------------------------                                                           | PREDICTOR
     E|           -         -                                                                        |
 775 M| 2        U_PRED = R_PRED / R_MAG_PRED;                                                       | PREDICTOR
     E|                     -           -                                                            |
 776 M| 2        Z_PRED = U_PRED . EARTH_POLE;                                                       | PREDICTOR
     E|           -         -                                             2                2    -    |
 777 M| 2        U_PRED = U_PRED + (3 EARTH_J2 / 2) (EARTH_R / R_MAG_PRED) ((1 - 5 Z_PRED ) U_PRED + 2  | PREDICTOR
     E|                     -                                                                        |
 777 M| 2        Z_PRED EARTH_POLE);                                                                 | PREDICTOR
     E|             -                             2    -                                             |
 778 M| 2        GRAV_ACCEL = -(EARTH_MU / R_MAG_PRED ) U_PRED;                                      | PREDICTOR
     C|  ------------------                                                                          | PREDICTOR
     C|  TOTAL ACCELERATION                                                                          | PREDICTOR
     C|  ------------------                                                                          | PREDICTOR
     E|           -         -              -                                                         |
 779 M| 2        A_PRED = AERO_ACCEL + GRAV_ACCEL;                                                   | PREDICTOR
     C|  ----------------------------                                                                | PREDICTOR
     C|  CALL RUNGA-KUTTA INTEGRATOR                                                                 | PREDICTOR
     C|  ----------------------------                                                                | PREDICTOR
 780 M| 2        CALL INTEGRATOR;                                                                    | PREDICTOR
     C|  ----------------                                                                            | PREDICTOR
     C|  STATE PARAMETERS                                                                            | PREDICTOR
     C|  ----------------                                                                            | PREDICTOR
     E|                                 -                                                            |
 781 M| 2        R_MAG_PRED = ABVAL(R_PRED);                                                         | PREDICTOR
```

```
STMT                                        SOURCE                                             CURRENT SCOPE

    E|                      -                                                                 |
782 M| 2       V_MAG_PRED = ABVAL(V_PRED);                                                    | PREDICTOR

    C|  -----------------                                                                     | PREDICTOR
    C|  RELATIVE VELOCITY                                                                     | PREDICTOR
    C|  -----------------                                                                     | PREDICTOR

    E|              -       -          -       -                                              |
783 M| 2       V_REL_PRED = V_PRED - (WE_NAV * R_PRED);                                       | PREDICTOR

    E|                                       -                                                |
784 M| 2       V_REL_MAG_PRED = ABVAL(V_REL_PRED);                                            | PREDICTOR

    C|  -----------------------------                                                         | PREDICTOR
    C|  1962 U.S. STANDARD ATMOSPHERE                                                         | PREDICTOR
    C|  -----------------------------                                                         | PREDICTOR

    E|                          -         -                +                                  |
785 M| 2       CALL USATMOS62(R_PRED, EARTH_POLE) ASSIGN(ATMOS);                              | PREDICTOR

786 M| 1    END;                                                                              | PREDICTOR

    C|  ----------------                                                                      | PREDICTOR
    C|  STATE PARAMETERS                                                                      | PREDICTOR
    C|  ----------------                                                                      | PREDICTOR

787 M| 1    T_PRED = T_PRED + DELTA_T_PRED;                                                   | PREDICTOR

    E|                    -        -                                                          |
788 M| 1    RDOT_PRED = V_PRED . R_PRED / R_MAG_PRED;                                         | PREDICTOR

    C|  --------------------------                                                            | PREDICTOR
    C|  CHECK FOR ATMOSPHERIC EXIT                                                            | PREDICTOR
    C|  --------------------------                                                            | PREDICTOR

789 M| 1    IF ((ATMOS.H > 400000) AND (RDOT_PRED > 0)) THEN                                  | PREDICTOR

    E|            .                                                                           |
790 M| 1       PRED_EXIT = TRUE;                                                              | PREDICTOR

    E|             .                                                                          |
791 M| 1    IF (PRED_EXIT = TRUE) THEN                                                        | PREDICTOR

792 M| 1       EXIT;                                                                          | PREDICTOR

    C|  ------------------------                                                              | PREDICTOR
    C|  CHECK FOR TAEM INTERFACE                                                              | PREDICTOR
    C|  ------------------------                                                              | PREDICTOR

793 M| 1    IF (ATMOS.H <= ALT_TAEM) THEN                                                     | PREDICTOR

794 M| 1       EXIT;                                                                          | PREDICTOR

795 M|   END;                                                                                 | PREDICTOR
```

```
STMT                                        SOURCE                                                             CURRENT SOPE

      C| --------------------------------------                                                          | PREDICTOR
      C| COMPUTE DOWNRANGE AND CROSSRANGE ERRORS                                                         | PREDICTOR
      C| --------------------------------------                                                          | PREDICTOR

      E|        *                                                                                        |
 796 M|   EF_FROM_REF = EARTH_FIXED_FROM_REFERENCE(T_PRED);                                              | PREDICTOR

      E|   -               *        -                                                                    |
 797 M|   IR_E = UNIT(EF_FROM_REF R_PRED);                                                               | PREDICTOR

      E|   -          *         -         -        -                                                     |
 798 M|   VR_E = EF_FROM_REF (V_PRED - WE_NAV * R_PRED);                                                 | PREDICTOR

      E|      -                  -    -                                                                  |
 799 M|   I_NORMAL = UNIT(IR_E * VR_E);                                                                  | PREDICTOR

      E|       -                   -                -            -         -                             |
 800 M|   I_INPLANE = UNIT(I_TARGET_EF - (I_TARGET_EF . I_NORMAL) I_NORMAL);                             | PREDICTOR

      E|                -               -                                                                |
 801 M|   DOT = I_INPLANE . I_TARGET_EF;                                                                 | PREDICTOR
 802 M|   IF (ABS(DOT) > 1) THEN                                                                         | PREDICTOR
 803 M|       DOT = SIGN(DOT);                                                                           | PREDICTOR

      E|                                                  -              -               -          -    |
 804 M|   CR_ERR = EARTH_R FT_TO_NM ARCCOS(DOT) SIGN((I_INPLANE * I_TARGET_EF) . (I_NORMAL * I_INPLANE)); | PREDICTOR

      E|           -           -                                                                         |
 805 M|   DOT = IR_E . I_INPLANE;                                                                        | PREDICTOR
 806 M|   IF (ABS(DOT) > 1) THEN                                                                         | PREDICTOR
 807 M|       DOT = SIGN(DOT);                                                                           | PREDICTOR

      E|                                                  -           -             -                    |
 808 M|   DR_ERR = EARTH_R FT_TO_NM ARCCOS(DOT) SIGN((IR_E * I_INPLANE) . I_NORMAL);                     | PREDICTOR
```

```
STMT                                    SOURCE                                             CURRENT SCOPE

809 M| INTEGRATOR:                                                                        | INTEGRATOR

809 M| PROCEDURE;                                                                         | INTEGRATOR

    C|  --------------------------------------------------                                | INTEGRATOR
    C|  FUNCTION: 4TH ORDER RUNGA-KUTTA INTEGRATOR ALGORITHM                              | INTEGRATOR
    C|  --------------------------------------------------                                | INTEGRATOR

    C|  ---------------                                                                   | INTEGRATOR
    C|  LOCAL VARIABLES                                                                   | INTEGRATOR
    C|  ---------------                                                                   | INTEGRATOR

810 M|    DECLARE ACCUM_ACCEL VECTOR(3) DOUBLE;                                           | INTEGRATOR

811 M|    DECLARE ACCUM_VEL VECTOR(3) DOUBLE;                                             | INTEGRATOR

812 M|    DECLARE ORIG_POS VECTOR(3) DOUBLE;                                              | INTEGRATOR

813 M|    DECLARE ORIG_VEL VECTOR(3) DOUBLE;                                              | INTEGRATOR


814 M|    DO CASE INTEG_LOOP;                                                             | INTEGRATOR

815 M| 1     DO;                                                                          | INTEGRATOR CASE 1

    E|            -          -                                                            |
816 M| 2        ORIG_POS = R_PRED;                                                        | INTEGRATOR

    E|            -          -                                                            |
817 M| 2        ORIG_VEL = V_PRED;                                                        | INTEGRATOR

    E|             -          -                                                           |
818 M| 2        ACCUM_VEL = V_PRED;                                                       | INTEGRATOR

    E|               -          -                                                         |
819 M| 2        ACCUM_ACCEL = A_PRED;                                                     | INTEGRATOR

    E|           -        -                          -                                    |
820 M| 2        R_PRED = ORIG_POS + .5 DELTA_T_PRED V_PRED;                               | INTEGRATOR

    E|           -        -                          -                                    |
821 M| 2        V_PRED = ORIG_VEL + .5 DELTA_T_PRED A_PRED;                               | INTEGRATOR

822 M| 1     END;                                                                         | INTEGRATOR

823 M| 1     DO;                                                                          | INTEGRATOR CASE 2

    E|             -            -            -                                            |
824 M| 2        ACCUM_VEL = ACCUM_VEL + 2 V_PRED;                                         | INTEGRATOR

    E|               -              -            -                                        |
825 M| 2        ACCUM_ACCEL = ACCUM_ACCEL + 2 A_PRED;                                     | INTEGRATOR
```

```
STMT                                        SOURCE                                          CURRENT SCOPE

     E|         -           -                      -                                        |
 826 M| 2        R_PRED = ORIG_POS + .5 DELTA_T_PRED V_PRED;                                | INTEGRATOR
     E|         -           -                      -                                        |
 827 M| 2        V_PRED = ORIG_VEL + .5 DELTA_T_PRED A_PRED;                                | INTEGRATOR
 828 M| 1     END;                                                                          | INTEGRATOR
 829 M| 1     DO;                                                                           | INTEGRATOR CASE 3
     E|           -            -            -                                               |
 830 M| 2        ACCUM_VEL = ACCUM_VEL + 2 V_PRED;                                          | INTEGRATOR
     E|            -              -            -                                            |
 831 M| 2        ACCUM_ACCEL = ACCUM_ACCEL + 2 A_PRED;                                      | INTEGRATOR
     E|         -           -                  -                                            |
 832 M| 2        R_PRED = ORIG_POS + DELTA_T_PRED V_PRED;                                   | INTEGRATOR
     E|         -           -                  -                                            |
 833 M| 2        V_PRED = ORIG_VEL + DELTA_T_PRED A_PRED;                                   | INTEGRATOR
 834 M| 1     END;                                                                          | INTEGRATOR
 835 M| 1     DO;                                                                           | INTEGRATOR CASE 4
     E|         -           -             -        -                                        |
 836 M| 2        R_PRED = ORIG_POS + (ACCUM_VEL + V_PRED) DELTA_T_PRED / 6;                 | INTEGRATOR
     E|         -           -              -          -                                     |
 837 M| 2        V_PRED = ORIG_VEL + (ACCUM_ACCEL + A_PRED) DELTA_T_PRED / 6;               | INTEGRATOR
 838 M| 1     END;                                                                          | INTEGRATOR
 839 M|    END;                                                                             | INTEGRATOR DO CASE END
 840 M| CLOSE INTEGRATOR;                                                                     | INTEGRATOR

**** B L O C K   S U M M A R Y ****

OUTER VARIABLES USED
    INTEG_LOOP, R_PRED, V_PRED, A_PRED, R_PRED*, DELTA_T_PRED, V_PRED*
```

```
STMT                                 SOURCE                                          CURRENT SCOPE

 841 M| EARTH_FIXED_FROM_REFERENCE:                                                 | EARTH_FIXED_FROM_REFER

 841 M| FUNCTION(T) MATRIX(3, 3) DOUBLE;                                            | EARTH_FIXED_FROM_REFER

     C| -------------------------------------------------------------------        | EARTH_FIXED_FROM_REFER
     C| DATE:       01/01/80                                                       | EARTH_FIXED_FROM_REFER
     C| FUNCTION:   COMPUTE (LEFT) REFERENCE INERTIAL TO EARTH-FIXED               | EARTH_FIXED_FROM_REFER
     C|             TRANSFORMATION MATRIX OR VARIOUS APPROXIMATIONS THEREOF.       | EARTH_FIXED_FROM_REFER
     C| INPUTS:     T - THE NUMBER OF SECONDS ELAPSED SINCE TIME ORIGIN            | EARTH_FIXED_FROM_REFER
     C| OUTPUTS:    EARTH-FIXED FROM REFERENCE ROTATION MATRIX                     | EARTH_FIXED_FROM_REFER
     C| COMMENTS:   NONE                                                           | EARTH_FIXED_FROM_REFER
     C| REFERENCE:  UNDOCUMENTED                                                   | EARTH_FIXED_FROM_REFER
     C| -------------------------------------------------------------------        | EARTH_FIXED_FROM_REFER

 842 M|    DECLARE T SCALAR DOUBLE;                                                 | EARTH_FIXED_FROM_REFER

 843 M|    DECLARE CLAMBDA SCALAR DOUBLE;                                           | EARTH_FIXED_FROM_REFER

 844 M|    DECLARE SLAMBDA SCALAR DOUBLE;                                           | EARTH_FIXED_FROM_REFER

 845 M|    CLAMBDA = COS((T - T_EPOCH) EARTH_RATE);                                 | EARTH_FIXED_FROM_REFER

 846 M|    SLAMBDA = SIN((T - T_EPOCH) EARTH_RATE);                                 | EARTH_FIXED_FROM_REFER

 847 M|    RETURN (MATRIX          (CLAMBDA, SLAMBDA, 0, -SLAMBDA, CLAMBDA, 0, 0, 0, 1) | EARTH_FIXED_FROM_REFER
     S|                 @DOUBLE,3,3                                                 |

     E|           *                                                                 |
 847 M|    EF_FROM_REF_AT_EPOCH);                                                   | EARTH_FIXED_FROM_REFER

 848 M| CLOSE EARTH_FIXED_FROM_REFERENCE;                                           | EARTH_FIXED_FROM_REFER

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    T_EPOCH, EARTH_RATE

OUTER VARIABLES USED
    EF_FROM_REF_AT_EPOCH
```

```
STMT                              SOURCE                                    CURRENT SCOPE

 849 M| CLOSE PREDICTOR;                                                    | PREDICTOR

**** B L O C K   S U M M A R Y ****

OUTER PROCEDURES CALLED
    USATMOS62, AERO_PARAMETERS, LOOKUP

COMPOOL VARIABLES USED
    T_GMT, R_NAV, V_NAV, WE_NAV, EARTH_POLE, ALT_TAEM_BIAS, DELTA_T_PRED_MIN, DELTA_T_PRED_GAIN, DELTA_T_PRED_MAX, K_LOD_NAV
    K_RHO_NAV, V_FINAL_MAG, V_MAG_CHANGE, PHI_MAX, DEG_TO_RAD, S_REF, MASS_NAV, EARTH_J2, EARTH_R, EARTH_MU, ALT_TAEM, FT_TO_NM

OUTER VARIABLES USED
    ALPHA_TRY, DELTA_T_PRED*, DELTA_T_PRED, PHI_TRY, PRED_EXIT*, PRED_EXIT, I_TARGET_EF, CR_ERR*, DR_ERR*

OUTER STRUCTURE TEMPLATES USED
     ATMOSPROP
```

```
STMT                                   SOURCE                                 CURRENT SCOPE

 850 M| CLOSE CORRECTOR;                                                      | CORRECTOR

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    ALPHA_MIN, ALPHA_MAX, PHI_DES_MAX

OUTER VARIABLES USED
    ALPHA_DES, PHI_DES, ALPHA_DES*, PHI_DES*
```

```
STMT                                SOURCE                                               CURRENT SCOPE

851 M| AERO_PARAMETERS:                                                                   | AERO_PARAMETERS

851 M| PROCEDURE(V_REL_MAG, ATMOS) ASSIGN(V_BAR, MACH);                                  | AERO_PARAMETERS

    C|  ---------------------------------------------------                               | AERO_PARAMETERS
    C|  FUNCTION: COMPUTE AERODYNAMIC FLOW REGIME PARAMETERS                              | AERO_PARAMETERS
    C|  ---------------------------------------------------                               | AERO_PARAMETERS

    C|  ---------------                                                                   | AERO_PARAMETERS
    C|  LOCAL VARIABLES                                                                   | AERO_PARAMETERS
    C|  ---------------                                                                   | AERO_PARAMETERS

852 M|    DECLARE V_REL_MAG SCALAR SINGLE;                                                | AERO_PARAMETERS

853 M|    DECLARE ATMOS ATMOSPROP-STRUCTURE;                                              | AERO_PARAMETERS

854 M|    DECLARE MACH SCALAR SINGLE;                                                     | AERO_PARAMETERS

855 M|    DECLARE V_BAR SCALAR SINGLE;                                                    | AERO_PARAMETERS

856 M|    DECLARE C_PRIME SCALAR SINGLE;                                                  | AERO_PARAMETERS

857 M|    DECLARE GAMMA_VBAR SCALAR SINGLE;                                               | AERO_PARAMETERS

858 M|    DECLARE REYNOLDS_NUMBER SINGLE;                                                 | AERO_PARAMETERS

859 M|    DECLARE SPEED_OF_SOUND SCALAR SINGLE;                                           | AERO_PARAMETERS

860 M|    DECLARE T_PRIME SCALAR SINGLE;                                                  | AERO_PARAMETERS

861 M|    DECLARE T_WALL SCALAR SINGLE;                                                   | AERO_PARAMETERS

862 M|    DECLARE VISCOSITY SCALAR SINGLE;                                                | AERO_PARAMETERS

    C|  ---------------                                                                   | AERO_PARAMETERS
    C|  LOCAL CONSTANTS                                                                   | AERO_PARAMETERS
    C|  ---------------                                                                   | AERO_PARAMETERS

863 M|    DECLARE C_BAR SCALAR SINGLE CONSTANT(25.0);                                     | AERO_PARAMETERS

864 M|    DECLARE DEG_R_TO_DEG_K SCALAR SINGLE CONSTANT(9 / 5);                           | AERO_PARAMETERS

865 M|    DECLARE GAMMA SCALAR SINGLE CONSTANT(1.4);                                      | AERO_PARAMETERS

    E|                                                      3                             |
866 M|    DECLARE UNIV_GAS_CONST SCALAR SINGLE CONSTANT(8.31432 10  );                    | AERO_PARAMETERS

867 M|    DECLARE MOLE_WT_ZERO SCALAR SINGLE CONSTANT(28.9644);                           | AERO_PARAMETERS

868 M|    DECLARE SPEED_OF_SOUND_CONST SCALAR SINGLE CONSTANT(SQRT(GAMMA UNIV_GAS_CONST / MOLE_WT_ZERO)); | AERO_PARAMETERS
```

```
STMT                                   SOURCE                                                   CURRENT SCOPE

     C|  -----------                                                                             | AERO_PARAMETERS
     C|  MACH NUMBER                                                                             | AERO_PARAMETERS
     C|  -----------                                                                             | AERO_PARAMETERS

 869 M|    SPEED_OF_SOUND = SPEED_OF_SOUND_CONST M_TO_FT SQRT(ATMOS.TM);                         | AERO_PARAMETERS

 870 M|    IF SPEED_OF_SOUND = 0 THEN                                                            | AERO_PARAMETERS

 871 M|       MACH = 0;                                                                          | AERO_PARAMETERS

 872 M|    ELSE                                                                                  | AERO_PARAMETERS

 872 M|       MACH = V_REL_MAG / SPEED_OF_SOUND;                                                 | AERO_PARAMETERS

     C|  ---------------                                                                         | AERO_PARAMETERS
     C|  REYNOLDS NUMBER                                                                         | AERO_PARAMETERS
     C|  ---------------                                                                         | AERO_PARAMETERS

     E|                        -6                    1.5                                          |
 873 M|    VISCOSITY = 1.458 10   KG_TO_SLUG ATMOS.TS    / ((110.4 + ATMOS.TS) M_TO_FT);         | AERO_PARAMETERS

 874 M|    IF (VISCOSITY = 0 OR ATMOS.H > 300000) THEN                                           | AERO_PARAMETERS

 875 M|       REYNOLDS_NUMBER = 0;                                                               | AERO_PARAMETERS

 876 M|    ELSE                                                                                  | AERO_PARAMETERS

 876 M|       REYNOLDS_NUMBER = ATMOS.RHO V_REL_MAG C_BAR / VISCOSITY;                           | AERO_PARAMETERS

     C|  -----------------                                                                       | AERO_PARAMETERS
     C|  VISCOUS PARAMETER                                                                       | AERO_PARAMETERS
     C|  -----------------                                                                       | AERO_PARAMETERS

 877 M|    IF REYNOLDS_NUMBER = 0 THEN                                                           | AERO_PARAMETERS

 878 M|       V_BAR = 0;                                                                         | AERO_PARAMETERS

 879 M|    ELSE                                                                                  | AERO_PARAMETERS
```

```
STMT                                        SOURCE                                                          CURRENT SCOPE

 879 M|       DO;                                                                                        | AERO_PARAMETERS

     C|  ----------------------------                                                                    | AERO_PARAMETERS
     C|  T_WALL FOR VISCOUS PARAMETER                                                                    | AERO_PARAMETERS
     C|  ----------------------------                                                                    | AERO_PARAMETERS

 880 M| 1        IF (ATMOS.H < 249000) THEN                                                              | AERO_PARAMETERS

 881 M| 1           T_WALL = 2178.0 DEG_R_TO_DEG_K;                                                      | AERO_PARAMETERS

 882 M| 1        ELSE IF (ATMOS.H >= 249000) AND (ATMOS.H < 360000) THEN                                 | AERO_PARAMETERS

 883 M| 1           T_WALL = (5913.0 - 0.015 ATMOS.H) DEG_R_TO_DEG_K;                                    | AERO_PARAMETERS

 884 M| 1        ELSE IF (ATMOS.H >= 360000) THEN                                                        | AERO_PARAMETERS

 885 M| 1           T_WALL = 504.0 DEG_R_TO_DEG_K;                                                       | AERO_PARAMETERS

     C|  ---------------------------                                                                     | AERO_PARAMETERS
     C|  GAMMA FOR VISCOUS PARAMETER                                                                     | AERO_PARAMETERS
     C|  ---------------------------                                                                     | AERO_PARAMETERS

 886 M| 1        IF (ATMOS.H < 100000) THEN                                                              | AERO_PARAMETERS

 887 M| 1           GAMMA_VBAR = 1.4;                                                                    | AERO_PARAMETERS

 888 M| 1        ELSE IF (ATMOS.H >= 100000) AND (ATMOS.H < 170000) THEN                                 | AERO_PARAMETERS

 889 M| 1           GAMMA_VBAR = 1.7 - 3.00E-6 ATMOS.H;                                                  | AERO_PARAMETERS

 890 M| 1        ELSE IF (ATMOS.H >= 170000) AND (ATMOS.H < 225000) THEN                                 | AERO_PARAMETERS

 891 M| 1           GAMMA_VBAR = 1.375 - 1.09E-6 ATMOS.H;                                                | AERO_PARAMETERS

 892 M| 1        ELSE IF (ATMOS.H >= 225000) AND (ATMOS.H < 300000) THEN                                 | AERO_PARAMETERS

 893 M| 1           GAMMA_VBAR = 1.220 - 4.00E-7 ATMOS.H;                                                | AERO_PARAMETERS

 894 M| 1        ELSE IF (ATMOS.H >= 300000) THEN                                                        | AERO_PARAMETERS

 895 M| 1           GAMMA_VBAR = 1.1;                                                                    | AERO_PARAMETERS

     C|  -----------------------------------------                                                       | AERO_PARAMETERS
     C|  T_PRIME AND C_PRIME FOR VISCOUS PARAMETER                                                       | AERO_PARAMETERS
     C|  -----------------------------------------                                                       | AERO_PARAMETERS

     E|                                                                               2                  |
 896 M| 1        T_PRIME = (.468 + .532 T_WALL / ATMOS.TS + .195 (GAMMA_VBAR - 1) MACH  / 2) ATMOS.TS;  | AERO_PARAMETERS

     E|                                  .5                    -(5/ATMOS.TS)                        -    |
 897 M| 1        C_PRIME = (T_PRIME / ATMOS.TS)  ((ATMOS.TS + 122.1 10             ) / (T_PRIME + 122.1 10  | AERO_PARAMETERS

     E|          (5/T_PRIME)  1                                                                          |
 897 M| 1                      )) ;                                                                      | AERO_PARAMETERS
```

```
STMT                                    SOURCE                                          CURRENT SCOPE

     E|                                      .5                                        |
 898 M| 1      V_BAR = MACH (C_PRIME / REYNOLDS_NUMBER)  ;                             | AERO_PARAMETERS

 899 M|        END;                                                                    | AERO_PARAMETERS

 900 M| CLOSE AERO_PARAMETERS;                                                         | AERO_PARAMETERS
```

```
**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    M_TO_FT, KG_TO_SLUG

OUTER STRUCTURE TEMPLATES USED
     ATMOSPROP
```

```
STMT                                  SOURCE                                        CURRENT SCOPE
 901 M| LOOKUP:                                                                    | LOOKUP

 901 M| PROCEDURE(ALPHA, ALT, V_BAR, MACH) ASSIGN(CL, CD);                         | LOOKUP

     C|  ----------------------------------------------------------------          | LOOKUP
     C|  FUNCTION:  LOOK-UP OF CL AND CD VERSUS ALPHA, ALTITUDE, VISCOUS           | LOOKUP
     C|             PARAMETER, AND MACH NUMBER.                                    | LOOKUP
     C|  INPUTS:           ALPHA    - ANGLE OF ATTACK (DEG)                        | LOOKUP
     C|                    ALTITUDE - ALTITUDE ABOVE FISHER ELLIPSOID (FT)         | LOOKUP
     C|                    V_BAR    - VISCOUS PARAMETER                            | LOOKUP
     C|                    MACH     - MACH NUMBER                                  | LOOKUP
     C|  OUTPUT:           CD       - DRAG COEFFICIENT                             | LOOKUP
     C|                    CL       - LIFT COEFFICIENT                             | LOOKUP
     C|  FLOW_REGIME:      1 = USE ALTITUDE DATA                                   | LOOKUP
     C|                    2 = USE V_BAR DATA                                      | LOOKUP
     C|                    3 = USE MACH DATA                                       | LOOKUP
     C|  REFERENCE: NOT DOCUMENTED                                                 | LOOKUP
     C|  ----------------------------------------------------------------          | LOOKUP

     C|  --------------------------                                                | LOOKUP
     C|  INPUT AND OUTPUT VARIABLES                                                | LOOKUP
     C|  --------------------------                                                | LOOKUP

 902 M|    DECLARE ALPHA SCALAR SINGLE;                                            | LOOKUP

 903 M|    DECLARE ALT SCALAR SINGLE;                                              | LOOKUP

 904 M|    DECLARE CD SCALAR SINGLE;                                               | LOOKUP

 905 M|    DECLARE CL SCALAR SINGLE;                                               | LOOKUP

 906 M|    DECLARE MACH SCALAR SINGLE;                                             | LOOKUP

 907 M|    DECLARE V_BAR SCALAR SINGLE;                                            | LOOKUP

     C|  ---------------                                                           | LOOKUP
     C|  LOCAL VARIABLES                                                           | LOOKUP
     C|  ---------------                                                           | LOOKUP

 908 M|    DECLARE ALPHA_FRACT SCALAR SINGLE;                                      | LOOKUP

 909 M|    DECLARE ALPHA_MAX SCALAR SINGLE CONSTANT(50);                           | LOOKUP

 910 M|    DECLARE ALPHA_MIN SCALAR SINGLE CONSTANT(0);                            | LOOKUP

 911 M|    DECLARE ALT_L SCALAR SINGLE;                                            | LOOKUP

 912 M|    DECLARE ALT_MAX SCALAR SINGLE CONSTANT(537000);                         | LOOKUP

 913 M|    DECLARE ALT_MIN SCALAR SINGLE CONSTANT(300000);                         | LOOKUP

 914 M|    DECLARE ALT_RUN SCALAR SINGLE CONSTANT(ALT_MAX - ALT_MIN);              | LOOKUP

 915 M|    DECLARE CD_1 SCALAR SINGLE;                                             | LOOKUP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
HAL/S  STD 360-24.20          I N T E R M E T R I C S ,   I N C .          APRIL 27, 1987    14:13:28.85
STMT                                     SOURCE                                                CURRENT SCOPE
 916 M|    DECLARE CD_2 SCALAR SINGLE;                                                        | LOOKUP
 917 M|    DECLARE CL_1 SCALAR SINGLE;                                                        | LOOKUP
 918 M|    DECLARE CL_2 SCALAR SINGLE;                                                        | LOOKUP
 919 M|    DECLARE COS_ALPHA SCALAR SINGLE;                                                   | LOOKUP
 920 M|    DECLARE FRACT SCALAR SINGLE;                                                       | LOOKUP
 921 M|    DECLARE FLOW_REGIME INTEGER SINGLE;                                                | LOOKUP
 922 M|    DECLARE I INTEGER SINGLE;                                                          | LOOKUP
 923 M|    DECLARE J INTEGER SINGLE;                                                          | LOOKUP
 924 M|    DECLARE MACH_L SCALAR SINGLE;                                                      | LOOKUP
 925 M|    DECLARE MACH_MAX SCALAR SINGLE CONSTANT(10);                                       | LOOKUP
 926 M|    DECLARE MACH_MIN SCALAR SINGLE CONSTANT(2);                                        | LOOKUP
 927 M|    DECLARE SIN_ALPHA SCALAR SINGLE;                                                   | LOOKUP
 928 M|    DECLARE V_BAR_L SCALAR SINGLE;                                                     | LOOKUP
 929 M|    DECLARE V_BAR_MAX SCALAR SINGLE CONSTANT(0.0744);                                  | LOOKUP
 930 M|    DECLARE V_BAR_MIN SCALAR SINGLE CONSTANT(0.0050);                                  | LOOKUP
 931 M|    DECLARE V_BAR_TABLE ARRAY(8) SCALAR SINGLE CONSTANT(.0050, .0103, .0148, .0208, .0245, .0373,  | LOOKUP
 931 M|    .0570, .0744);                                                                     | LOOKUP
```

```
STMT                              SOURCE                                        CURRENT SCOPE

    C| --------------------------------------                                  | LOOKUP
    C| ERV AERODYNAMIC DATA TABLES NOT LISTED                                  | LOOKUP
    C| --------------------------------------                                  | LOOKUP
```

```
STMT                                    SOURCE                                         CURRENT SCOPE
    C| ----------------------------------                                              | LOOKUP
    C| DETERMINE INDEX OF ANGLE OF ATTACK                                              | LOOKUP
    C| ----------------------------------                                              | LOOKUP

938 M|   J = MIDVAL(1, (TRUNCATE(ALPHA) + 1), 50);                                     | LOOKUP

939 M|   ALPHA_FRACT = (ALPHA - TRUNCATE(ALPHA));                                      | LOOKUP

    C| ----------------------------------------                                        | LOOKUP
    C| DETERMINE FLOW REGIME TO USE FOR LOOK UP                                        | LOOKUP
    C| ----------------------------------------                                        | LOOKUP

940 M|   FLOW_REGIME = 0;                                                              | LOOKUP

941 M|   IF (ALT >= ALT_MIN) THEN                                                      | LOOKUP

942 M|      FLOW_REGIME = 1;                                                           | LOOKUP

943 M|   ELSE IF (MACH <= MACH_MAX) THEN                                               | LOOKUP

944 M|      FLOW_REGIME = 3;                                                           | LOOKUP

945 M|   ELSE                                                                          | LOOKUP

945 M|      FLOW_REGIME = 2;                                                           | LOOKUP

946 M|   DO CASE FLOW_REGIME;                                                          | LOOKUP

    C| ------------------------------------------                                      | LOOKUP
    C| INTERPOLATE FOR ALPHA USING CORRECT TABLES                                      | LOOKUP
    C| ------------------------------------------                                      | LOOKUP

    C| -------------                                                                   | LOOKUP
    C| ALTITUDE DATA                                                                   | LOOKUP
    C| -------------                                                                   | LOOKUP

947 M| 1    DO;                                                                        | LOOKUP CASE 1

948 M| 2       ALT_L = MIDVAL(ALT_MIN, ALT, ALT_MAX);                                  | LOOKUP

949 M| 2       CL_1 = CL_ALT     + (CL_ALT       - CL_ALT    ) ALPHA_FRACT;            | LOOKUP
    S|                     1,J            1,J+1          1,J                           |

950 M| 2       CL_2 = CL_ALT     + (CL_ALT       - CL_ALT    ) ALPHA_FRACT;            | LOOKUP
    S|                     2,J            2,J+1          2,J                           |

951 M| 2       CD_1 = CD_ALT     + (CD_ALT       - CD_ALT   ') ALPHA_FRACT;            | LOOKUP
    S|                     1,J            1,J+1          1,J                           |

952 M| 2       CD_2 = CD_ALT     + (CD_ALT       - CD_ALT    ) ALPHA_FRACT;            | LOOKUP
    S|                     2,J            2,J+1          2,J                           |

953 M| 2       FRACT = (ALT_L - ALT_MIN) / ALT_RUN;                                    | LOOKUP

954 M| 1    END;                                                                       | LOOKUP
```

```
STMT                                   SOURCE                                              CURRENT SCOPE
      C|  ------------                                                                     | LOOKUP
      C|  VISCOUS DATA                                                                     | LOOKUP
      C|  ------------                                                                     | LOOKUP

 955 M| 1     DO;                                                                          | LOOKUP CASE 2

 956 M| 2        V_BAR_L = MIDVAL(V_BAR_MIN, V_BAR, V_BAR_MAX);                            | LOOKUP

 957 M| 2        DO FOR TEMPORARY K = 7 TO 1 BY -1;                                        | LOOKUP

 958 M| 3           I = K;                                                                 | LOOKUP

 959 M| 3           IF (V_BAR_L >= V_BAR_TABLE ) THEN                                      | LOOKUP
     S|                                       K                                            |

 960 M| 3              EXIT;                                                               | LOOKUP

 961 M| 2        END;                                                                      | LOOKUP

 962 M| 2        CL_1 = CL_VISC     + (CL_VISC       - CL_VISC    ) ALPHA_FRACT;           | LOOKUP
     S|                    I,J               I,J+1           I,J                           |

 963 M| 2        CL_2 = CL_VISC       + (CL_VISC         - CL_VISC       ) ALPHA_FRACT;    | LOOKUP
     S|                    I+1,J               I+1,J+1           I+1,J                     |

 964 M| 2        CD_1 = CD_VISC     + (CD_VISC       - CD_VISC    ) ALPHA_FRACT;           | LOOKUP
     S|                    I,J               I,J+1           I,J                           |

 965 M| 2        CD_2 = CD_VISC       + (CD_VISC         - CD_VISC       ) ALPHA_FRACT;    | LOOKUP
     S|                    I+1,J               I+1,J+1           I+1,J                     |

 966 M| 2        FRACT = (V_BAR_L - V_BAR_TABLE ) / (V_BAR_TABLE     - V_BAR_TABLE );      | LOOKUP
     S|                                     I                 I+1               I          |

 967 M| 1     END;                                                                         | LOOKUP

      C|  ---------                                                                        | LOOKUP
      C|  MACH DATA                                                                        | LOOKUP
      C|  ---------                                                                        | LOOKUP

 968 M| 1     DO;                                                                          | LOOKUP CASE 3

 969 M| 2        MACH_L = MIDVAL(MACH_MIN, MACH, MACH_MAX);                                | LOOKUP

 970 M| 2        I = MIDVAL(1, (TRUNCATE(MACH_L / 2)), 4);                                 | LOOKUP

 971 M| 2        CL_1 = CL_MACH     + (CL_MACH       - CL_MACH    ) ALPHA_FRACT;           | LOOKUP
     S|                    I,J               I,J+1           I,J                           |

 972 M| 2        CL_2 = CL_MACH       + (CL_MACH         - CL_MACH       ) ALPHA_FRACT;    | LOOKUP
     S|                    I+1,J               I+1,J+1           I+1,J                     |

 973 M| 2        CD_1 = CD_MACH     + (CD_MACH       - CD_MACH    ) ALPHA_FRACT;           | LOOKUP
     S|                    I,J               I,J+1           I,J                           |
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 HAL/S  STD 360-24.20           I N T E R M E T R I C S ,   I N C .              APRIL 27, 1987    14:13:28.85
STMT                                                SOURCE                                                  CURRENT SCOPE

 974 M| 2        CD_2 = CD_MACH      + (CD_MACH        - CD_MACH      ) ALPHA_FRACT;                      | LOOKUP
     S|                       I+1,J            I+1,J+1          I+1,J                                        |

 975 M| 2        FRACT = (MACH_L / 2 - I);                                                                    | LOOKUP

 976 M| 1     END;                                                                                            | LOOKUP

 977 M|    END;                                                                                               | LOOKUP DO CASE END

     C|  --------------------------                                                                           | LOOKUP
     C|  INTERPOLATE BETWEEN TABLES                                                                           | LOOKUP
     C|  --------------------------                                                                           | LOOKUP

 978 M|    CL = CL_1 + (CL_2 - CL_1) FRACT;                                                                   | LOOKUP

 979 M|    CD = CD_1 + (CD_2 - CD_1) FRACT;                                                                   | LOOKUP

 980 M| CLOSE LOOKUP;                                                                                         | LOOKUP
```

```
STMT                                    SOURCE                                          CURRENT SCOPE
 981 M| USATMOS62:                                                                      | USATMOS62
 981 M| PROCEDURE(R, POLE) ASSIGN(ATMOS);                                               | USATMOS62
     C| --------------------------------------------------------------                  | USATMOS62
     C| DATE:     12/28/81                                                              | USATMOS62
     C| FUNCTION: COMPUTES THE DENSITY OF THE ATMOSPHERE (0-150KM)                      | USATMOS62
     C|           AT A SPECIFIED ALTITUDE USING THE 1962 U.S.STANDARD                   | USATMOS62
     C|           ATMOSPHERE MODEL.                                                     | USATMOS62
     C| INPUTS:       R    - INERTIAL POSITION VECTOR (FT)                              | USATMOS62
     C|               POLE - INERTIAL NORTH POLE (ND)                                   | USATMOS62
     C| OUTPUTS:      ATMOS.H     - ALTITUDE (FT)                                       | USATMOS62
     C|               ATMOS.RHO   - DENSITY (LBM/FT**3)                                 | USATMOS62
     C|               ATMOS.TS    - STATIC TEMPERATURE OF AIR (DEG K)                   | USATMOS62
     C|               ATMOS.TM    - MOLECULAR TEMPERATURE OF AIR (DEG K)                | USATMOS62
     C| NOMENCLATURE: A    - EARTH EQUATORIAL RADIUS (M)                                | USATMOS62
     C|               F    - EARTH FLATTENING (ND)                                      | USATMOS62
     C|               RO   - EARTH RADIUS (PHI = 45 DEGS) (M)                           | USATMOS62
     C|               PHI  - GEOGRAPHIC LATITUDE (DEGS)                                 | USATMOS62
     C|               PSI  - GEOCENTRIC LATITUDE (DEGS)                                 | USATMOS62
     C|               GO   - SEA-LEVEL GRAVITY (M/S/S)                                  | USATMOS62
     C|               MO   - MEAN MOLECULAR WEIGHT OF AIR (ND)                          | USATMOS62
     C|               RR   - UNIVERSAL GAS CONSTANT                                     | USATMOS62
     C|                             (JOULES / (DEG K) / (KG-MOL))                       | USATMOS62
     C| --------------------------------------------------------------                  | USATMOS62

 982 M|    DECLARE ATMOS ATMOSPROP-STRUCTURE;                                           | USATMOS62

 983 M|    DECLARE R VECTOR(3) DOUBLE;                                                  | USATMOS62
 984 M|    DECLARE POLE VECTOR(3) DOUBLE;                                               | USATMOS62

 985 M|    DECLARE GO SCALAR SINGLE CONSTANT(9.80665);                                  | USATMOS62
 986 M|    DECLARE MO SCALAR SINGLE CONSTANT(28.9644);                                  | USATMOS62
 987 M|    DECLARE RR SCALAR SINGLE CONSTANT(8314.32);                                  | USATMOS62
 988 M|    DECLARE K2 SCALAR SINGLE CONSTANT(GO MO / RR);                               | USATMOS62

 989 M|    DECLARE M_TO_FT SCALAR DOUBLE CONSTANT(1 / .3048);                           | USATMOS62
 990 M|    DECLARE KG_TO_LBM SCALAR SINGLE CONSTANT(1 / .45359237);                     | USATMOS62

 991 M|    DECLARE A SCALAR DOUBLE CONSTANT(6378178);                                   | USATMOS62
 992 M|    DECLARE F SCALAR DOUBLE CONSTANT(1 / 298.32);                                | USATMOS62
```

```
STMT                                        SOURCE                                                      CURRENT SCOPE

     E|                                               4                  2                                |
 993 M|   DECLARE RO SCALAR SINGLE CONSTANT(A SQRT((1 +.(1 - F) ) / (1 + (1 - F) )));                     | USATMOS62
 994 M|   DECLARE I INTEGER SINGLE;                                                                      | USATMOS62
 995 M|   DECLARE NSEGS INTEGER SINGLE CONSTANT(12);                                                     | USATMOS62
 996 M|   DECLARE H_BASE ARRAY(NSEGS + 1) SCALAR SINGLE CONSTANT(0, 11000, 20000, 32000, 47000, 52000,   | USATMOS62
 996 M|   61000, 79000, 90000, 100000, 110000, 120000, 150000);                                         | USATMOS62
 997 M|   DECLARE TM_BASE ARRAY(NSEGS + 1) SCALAR SINGLE CONSTANT(288.15, 216.65, 216.65, 228.65, 270.65, | USATMOS62
 997 M|   270.65, 252.65, 180.65, 180.65, 210.65, 260.65, 360.65, 960.65);                              | USATMOS62
 998 M|   DECLARE T_BASE ARRAY(NSEGS + 1) SCALAR SINGLE CONSTANT(288.15, 216.65, 216.65, 228.65, 270.65,  | USATMOS62
 998 M|   270.65, 252.65, 180.65, 180.65, 210.02, 257.00, 349.49, 892.79);                              | USATMOS62
 999 M|   DECLARE RHO_BASE ARRAY(NSEGS + 1) SCALAR SINGLE CONSTANT(1.2250, 0.36392, 0.088035, 0.013225,  | USATMOS62
 999 M|   0.0014275, 0.00075943, 0.00025109, 0.00002001, 0.000003170, 0.0000004974, 0.00000009829,       | USATMOS62
 999 M|   0.00000002436, 0.000000001836);                                                                | USATMOS62
1000 M|   DECLARE DTMDH ARRAY(NSEGS) SCALAR SINGLE INITIAL(-.0065, .0000, .0010, .0028, .0000, -.0020, -  | USATMOS62
1000 M|   .0040, .0000, .0030, .0050, .0100, .0200);                                                     | USATMOS62
1001 M|   DECLARE DTDH ARRAY(NSEGS) SCALAR SINGLE INITIAL(-.006500, .000000, .001000, .0028000, .000000000 | USATMOS62
1001 M|   , -.002000, -.004000, .000000, .002937, .0046980, .009249000, .018110);                        | USATMOS62

1002 M|   DECLARE SCALAR DOUBLE,                                                                         | USATMOS62
1002 M|             R_MAG, H, SPSI;                                                                      | USATMOS62
1003 M|   DECLARE SCALAR SINGLE,                                                                         | USATMOS62
1003 M|             DH, EXPO, ALTADJ, ALTRATIO, GRAVRATIO, TEMPRATIO;                                    | USATMOS62

     C|  -----------------------------------------                                                       | USATMOS62
     C|  DETERMINE ALTITUDE ABOVE FISHER ELLIPSOID                                                       | USATMOS62
     C|  -----------------------------------------                                                       | USATMOS62

     E|                  -                                                                               |
1004 M|   R_MAG = ABVAL(R);                                                                              | USATMOS62

     E|            -            -                                                                        |
1005 M|   SPSI = (R / R_MAG) . POLE;                                                                     | USATMOS62
```

```
STMT                                   SOURCE                                        CURRENT SCOPE

     E|                                                            2              |
1006 M|   H = (R_MAG / M_TO_FT) - A (1 - F) / SQRT(1 - F (2 - F) (1 - SPSI  ));     | USATMOS62

1007 M|   ATMOS.H = H M_TO_FT;                                                      | USATMOS62

     C| ---------------------------------------------------------------          | USATMOS62
     C| DETERMINE DENSITY, STATIC TEMPERATURE, AND MOLECULAR TEMPERATURE          | USATMOS62
     C| ---------------------------------------------------------------          | USATMOS62

1008 M|   IF (H > 150000) THEN                                                      | USATMOS62

1009 M|      ATMOS.RHO = 0;                                                         | USATMOS62

1010 M|   ELSE                                                                      | USATMOS62

1010 M|      DO;                                                                    | USATMOS62

1011 M| 1       GRAVRATIO = RO / (RO + H);                                          | USATMOS62

1012 M| 1       IF (H < 90000) THEN                                                 | USATMOS62

1013 M| 1          H = GRAVRATIO H;                                                 | USATMOS62

     E|                             2                                             |
1014 M| 1       GRAVRATIO = GRAVRATIO ;                                             | USATMOS62

1015 M| 1       DO FOR TEMPORARY N = 1 TO NSEGS;                                    | USATMOS62

1016 M| 2          I = N;                                                           | USATMOS62

1017 M| 2          IF (H < H_BASE    ) THEN                                         | USATMOS62
     S|                         N+1                                               |

1018 M| 2             EXIT;                                                         | USATMOS62

1019 M| 1       END;                                                                | USATMOS62

1020 M| 1       DH = H - H_BASE ;                                                   | USATMOS62
     S|                      I                                                    |

1021 M| 1       ATMOS.TM = TM_BASE  + DH DTMDH ;                                    | USATMOS62
     S|                        I            I                                     |

1022 M| 1       ATMOS.TS = T_BASE  + DH DTDH ;                                      | USATMOS62
     S|                       I           I                                       |

1023 M| 1       IF (H < 90000) THEN                                                 | USATMOS62

1024 M| 1          DO;                                                              | USATMOS62
```

```
STMT                                   SOURCE                                                   CURRENT SCOPE

1025 M| 2        IF (DTMDH  = 0) THEN                                                          | USATMOS62
     S|                   I                                                                    |

1026 M| 2          ATMOS.RHO = RHO_BASE  EXP(-K2 DH / TM_BASE  );                              | USATMOS62
     S|                              I                          I                              |

1027 M| 2        ELSE                                                                          | USATMOS62

1027 M| 2          DO;                                                                         | USATMOS62

1028 M| 3            EXPO = 1 + K2 / DTMDH  ;                                                  | USATMOS62
     S|                                I                                                       |

     E|                                                       EXPO                             |
1029 M| 3            ATMOS.RHO = RHO_BASE (TM_BASE  / ATMOS.TM)     ;                          | USATMOS62
     S|                                  I        I                                            |

1030 M| 2          END;                                                                        | USATMOS62

1031 M| 1      END;                                                                            | USATMOS62

1032 M| 1    ELSE                                                                              | USATMOS62

1032 M| 1      DO;                                                                             | USATMOS62

1033 M| 2        ALTADJ = R0 + H_BASE  - TM_BASE  / DTMDH  ;                                   | USATMOS62
     S|                           I          I         I                                       |

1034 M| 2        ALTRATIO = (R0 + H) / (R0 + H_BASE  );                                        | USATMOS62
     S|                                       I                                                |

1035 M| 2        TEMPRATIO = TM_BASE  / ATMOS.TM;                                              | USATMOS62
     S|                          I                                                             |

     E|                                           2                                            |
1036 M| 2        EXPO = (K2 / DTMDH  ) (R0 / ALTADJ)  ;                                        | USATMOS62
     S|                          I                                                             |

1037 M| 2        ATMOS.RHO = RHO_BASE  TEMPRATIO EXP((K2 / DTMDH  ) GRAVRATIO ALTRATIO (DH / ALTADJ)) | USATMOS62
     S|                           I                        I                                   |

     E|                           EXPO                                                         |
1037 M| 2        (TEMPRATIO ALTRATIO)     ;                                                    | USATMOS62

1038 M| 1      END;                                                                            | USATMOS62

     E|                                                 3                                      |
1039 M| 1    ATMOS.RHO = ATMOS.RHO (KG_TO_LBM / (M_TO_FT  G_TO_FPS2));                         | USATMOS62

1040 M|    END;                                                                                | USATMOS62

1041 M| CLOSE USATMOS62;                                                                       | USATMOS62
```

```
STMT                              SOURCE                              CURRENT SCOPE

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    G_TO_FPS2

OUTER STRUCTURE TEMPLATES USED
     ATMOSPROP
```

```
STMT                                SOURCE                                  CURRENT SCOPE
1042 M| CLOSE AERO_GUID;                                                    | AERO_GUID

**** B L O C K   S U M M A R Y ****

COMPOOL VARIABLES USED
    G_LOAD, G_RUN_GUIDANCE, ALT_NAV, ALT_FREEZE_GUID, GUID_PASS_LIM
```

```
**** C O M P I L A T I O N   L A Y O U T ****
 FSM_POOL: EXTERNAL COMPOOL;
 IL_POOL: EXTERNAL COMPOOL;
 AERO_GUID: PROCEDURE;
    INITIAL_GUID: PROCEDURE;
    FILTERS: PROCEDURE;
    HEAT_RATE_CONTROL: PROCEDURE;
    ATTITUDE_COMMAND: PROCEDURE;
    CORRECTOR: PROCEDURE;
       PREDICTOR: PROCEDURE;
          INTEGRATOR: PROCEDURE;
          EARTH_FIXED_FROM_REFERENCE: FUNCTION;
    AERO_PARAMETERS: PROCEDURE;
    LOOKUP: PROCEDURE;
    USATMOS62: PROCEDURE;
```

SYMBOL & CROSS REFERENCE TABLE LISTING:

(CROSS REFERENCE FLAG KEY: 4 = ASSIGNMENT, 2 = REFERENCE, 1 = SUBSCRIPT USE, 0 = DEFINITION)

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
526	ATMOSPROP	STRUCTURE TEMPLATE	ALIGNED XREF: 0 0526 2 0574 2 0582 2 0583 2 0584 2 0591 2 0592 2 0704 2 0734 2 0738 2 0747 2 0748 2 0764 2 0785 2 0789 2 0793 2 0853 2 0869 2 0873 2 0874 2 0876 2 0880 2 0882 2 0883 2 0884 2 0886 2 0888 2 0889 2 0890 2 0891 2 0892 2 0893 2 0894 2 0896 2 0897 2 0982 2 1007 2 1009 2 1021 2 1022 2 1026 2 1029 2 1035 2 1037 2 1039
526	H	1 SCALAR	SINGLE, ALIGNED XREF: 0 0526 2 0584 2 0738 2 0748 2 0789 2 0793 2 0874 2 0880 2 0882 2 0883 2 0884 2 0886 2 0888 2 0889 2 0890 2 0891 2 0892 2 0893 2 0894 4 1007
526	RHO	1 SCALAR	SINGLE, ALIGNED XREF: 0 0526 2 0591 2 0592 2 0764 2 0876 4 1009 4 1026 4 1029 4 1037 6 1039
526	TS	1 SCALAR	SINGLE, ALIGNED XREF: 0 0526 2 0873 2 0896 2 0897 4 1022
526	TM	1 SCALAR	SINGLE, ALIGNED XREF: 0 0526 2 0869 4 1021 2 1029 2 1035
991	A	SCALAR	DOUBLE, ALIGNED, STATIC, CONSTANT XREF: 0 0991 2 0993 2 1006
572	A_DRAG_MAG	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0572 4 0586 2 0587 2 0588 2 0589
573	A_LIFT_MAG	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0573 4 0587 2 0588
87	A_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0087 2 0586 2 0587
701	A_PRED	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0701 4 0735 2 0741 4 0779 2 0819 2 0821 2 0825 2 0827 2 0831 2 0833 2 0837
702	A_PRED_MAG	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0702 4 0741 2 0742 2 0743
810	ACCUM_ACCEL	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0810 4 0819 6 0825 6 0831 2 0837
811	ACCUM_VEL	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0811 4 0818 6 0824 6 0830 2 0836
751	AERO_ACCEL	3 - VECTOR	SINGLE, TEMPORARY XREF: 0 0751 4 0774 2 0779
515	AERO_GUID	PROCEDURE	XREF: 0 0515 NOT REFERENCED
851	AERO_PARAMETERS	PROCEDURE	XREF: 2 0583 2 0747 0 0851
901	ALPHA	SCALAR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0901 0 0902 2 0938 2 0939
352	ALPHA_CMD	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0352 4 0553 4 0635 6 0645
516	ALPHA_DES	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0516 4 0551 2 0635 2 0668 2 0672 2 0676 6 0693 6 0694
401	ALPHA_EI	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0401 2 0551 2 0553
908	ALPHA_FRACT	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0908 4 0939 2 0949 2 0950 2 0951 2 0952 2 0962 2 0963 2 0964 2 0965 2 0971 2 0972 2 0973 2 0974
513	ALPHA_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0553 2 0645 2 0694
909	ALPHA_MAX	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0909 NOT REFERENCED
513	ALPHA_MIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0553 2 0645 2 0694
910	ALPHA_MIN	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0910 NOT REFERENCED
74	ALPHA_NAV	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0074 2 0584
703	ALPHA_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0703 4 0733 2 0748
649	ALPHA_TRY	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0649 4 0668 4 0672 4 0676 2 0733
901	ALT	SCALAR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0901 0 0903 2 0941 2 0948

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
513	ALT_FREEZE_GUID	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0535
911	ALT_L	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0911 4 0948 2 0953
912	ALT_MAX	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0912 2 0914 2 0948
913	ALT_MIN	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0913 2 0914 2 0941 2 0948 2 0953
79	ALT_NAV	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0079 2 0535 2 0625
914	ALT_RUN	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0914 2 0953
513	ALT_TAEM	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0793
513	ALT_TAEM_BIAS	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0738
1003	ALTADJ	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 1003 4 1033 2 1036 2 1037
1003	ALTRATIO	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 1003 4 1034 2 1037
981	ATMOS	STRUCTURE	ATMOSPROP-STRUCTURE, ALIGNED, ASSIGN-PARM XREF: 0 0981 0 0982 4 1007 4 1009 4 1021 4 1022 4 1026 6 1029 2 1035 4 1037 6 1039
851	ATMOS	STRUCTURE	ATMOSPROP-STRUCTURE, ALIGNED, INPUT-PARM XREF: 0 0851 0 0853 2 0869 2 0873 2 0874 2 0876 2 0880 2 0882 2 0883 2 0884 2 0886 2 0888 2 0889 2 0890 2 0891 2 0892 2 0893 2 0894 2 0896 2 0897
574	ATMOS	STRUCTURE	ATMOSPROP-STRUCTURE, ALIGNED, STATIC XREF: 0 0574 4 0582 2 0583 2 0584 2 0591 2 0592
704	ATMOS	STRUCTURE	ATMOSPROP-STRUCTURE, ALIGNED, STATIC XREF: 0 0704 4 0734 2 0738 2 0747 2 0748 2 0764 4 0785 2 0789 2 0793
634	ATTITUDE_COMMAND	PROCEDURE	XREF: 2 0541 0 0634
863	C_BAR	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0863 2 0876
856	C_PRIME	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0856 4 0897 2 0898
901	CD	SCALAR	SINGLE, ALIGNED, ASSIGN-PARM XREF: 0 0901 0 0904 4 0979 NOT REFERENCED
933	CD_ALT	SCALAR ARRAY	ARRAY(2,51), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0933 2 0951 2 0952
937	CD_MACH	SCALAR ARRAY	ARRAY(5,51), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0937 2 0973 2 0974
575	CD_NOM	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0575 4 0584 2 0585 2 0589
705	CD_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0705 4 0748 2 0749 2 0769
935	CD_VISC	SCALAR ARRAY	ARRAY(8,51), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0935 2 0964 2 0965
915	CD_1	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0915 4 0951 4 0964 4 0973 2 0979
916	CD_2	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0916 4 0952 4 0965 4 0974 2 0979
901	CL	SCALAR	SINGLE, ALIGNED, ASSIGN-PARM XREF: 0 0901 0 0905 4 0978 NOT REFERENCED
932	CL_ALT	SCALAR ARRAY	ARRAY(2,51), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0932 2 0949 2 0950
517	CL_EST	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0517 4 0593 2 0625
936	CL_MACH	SCALAR ARRAY	ARRAY(5,51), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0936 2 0971 2 0972
576	CL_NOM	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0576 4 0584 2 0585 2 0593
706	CL_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0706 4 0748 2 0749
934	CL_VISC	SCALAR ARRAY	ARRAY(8,51), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0934 2 0962 2 0963
917	CL_1	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0917 4 0949 4 0962 4 0971 2 0978
918	CL_2	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0918 4 0950 4 0963 4 0972 2 0978

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
843	CLAMBDA	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 0843 4 0845 2 0847
648	CORRECTOR	PROCEDURE	XREF: 2 0536 0 0648
919	COS_ALPHA	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0919 NOT USED
640	COSPHI_CMD	SCALAR	SINGLE, TEMPORARY XREF: 0 0640 4 0641 6 0642 2 0643
518	COSPHI_QDOT	SCALAR	SINGLE, ALIGNED, STATIC, INITIAL XREF: 0 0518 4 0630 4 0632 2 0638 2 0641
596	COSPHI_1	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0596 4 0628 2 0630
597	COSPHI_2	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0597 4 0629 2 0630
752	CPHI	SCALAR	SINGLE, TEMPORARY XREF: 0 0752 4 0767 2 0773
659	CR_ERR	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0659 2 0682 4 0804
658	CR_ERROR	SCALAR ARRAY	ARRAY(3), SINGLE, ALIGNED, STATIC XREF: 0 0658 4 0682 2 0686 2 0687 2 0689
657	CRE	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0657 4 0689 2 0693 2 0695
655	DCRE_DA	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0655 4 0686 2 0690 2 0695
656	DCRE_DP	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0656 4 0687 2 0690 2 0693
653	DDRE_DA	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0653 4 0684 2 0690 2 0695
654	DDRE_DP	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0654 4 0685 2 0690 2 0693
864	DEG_R_TO_DEG_K	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0864 2 0881 2 0883 2 0885
5	DEG_TO_RAD	SCALAR	DOUBLE, ALIGNED, CONSTANT XREF: 0 0005 2 0006 2 0007 2 0555 2 0556 2 0641 2 0766
650	DELTA_ALPHA	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0650 2 0672 2 0684 2 0686
651	DELTA_PHI	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0651 2 0677 2 0685 2 0687
519	DELTA_T_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0519 4 0739 4 0743 4 0744 6 0745 2 0787 2 0820 2 0821 2 0826 2 0827 2 0832 2 0833 2 0836 2 0837
513	DELTA_T_PRED_GAIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0743
513	DELTA_T_PRED_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0744 2 0745
513	DELTA_T_PRED_MIN	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0739 2 0745
652	DETERM	SCALAR	SINGLE, ALIGNED, STATIC, INITIAL XREF: 0 0652 4 0690 2 0691 2 0693 2 0695
1003	DH	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 1003 4 1020 2 1021 2 1022 2 1026 2 1037
707	DOT	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 0707 4 0801 2 0802 6 0803 2 0804 4 0805 2 0806 6 0807 2 0808
663	DR_ERR	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0663 2 0681 4 0808
662	DR_ERROR	SCALAR ARRAY	ARRAY(3), SINGLE, ALIGNED, STATIC XREF: 0 0662 4 0681 2 0684 2 0685 2 0688
753	DRAG_ACCEL	SCALAR	SINGLE, TEMPORARY XREF: 0 0753 4 0769 2 0770 2 0774
664	DRE	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0664 4 0688 2 0693 2 0695
513	DT_AEROGUID	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0620
1001	DTDH	SCALAR ARRAY	ARRAY(12), SINGLE, ALIGNED, STATIC, INITIAL XREF: 0 1001 2 1022
1000	DTMDH	SCALAR ARRAY	ARRAY(12), SINGLE, ALIGNED, STATIC, INITIAL XREF: 0 1000 2 1021 2 1025 2 1028 2 1033 2 1036 2 1037
699	EARTH_FIXED_FROM_REFERENCE	3 X 3 MATRIX FUNCTION	DOUBLE XREF: 0 0699 2 0796
463	EARTH_FLAT	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0463 2 0557
464	EARTH_J2	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0464 2 0777
465	EARTH_MU	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0465 2 0778
466	EARTH_POLE	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0466 2 0582 2 0734 2 0776 2 0777 2 0785
467	EARTH_R	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0467 2 0777 2 0804 2 0808
468	EARTH_RATE	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0468 2 0845 2 0846

```
 DCL  NAME                        TYPE                     ATTRIBUTES & CROSS REFERENCE

  708  EF_FROM_REF                3 X 3 MATRIX             DOUBLE, ALIGNED, STATIC   XREF: 0 0708  4 0796  2 0797  2 0798
  520  EF_FROM_REF_AT_EPOCH       3 X 3 MATRIX             DOUBLE, ALIGNED, STATIC   XREF: 0 0520  4 0549  2 0847
  308  EF_TO_REF_AT_EPOCH         3 X 3 MATRIX             DOUBLE, ALIGNED, INITIAL  XREF: 0 0308  2 0549
 1003  EXPO                       SCALAR                   SINGLE, ALIGNED, STATIC   XREF: 0 1003  4 1028  2 1029  4 1036
                                                           2 1037
  992  F                          SCALAR                   DOUBLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0992  2 0993
                                                           2 1006
  571  FILTERS                    PROCEDURE                  XREF: 2 0534  0 0571
  598  FIRST_PASS                 BIT(1)                   ALIGNED, STATIC, INITIAL  XREF: 0 0598  2 0611  4 0613
  921  FLOW_REGIME                INTEGER                  SINGLE, ALIGNED, STATIC   XREF: 0 0921  4 0940  4 0942  4 0944
                                                           4 0945  2 0946
  920  FRACT                      SCALAR                   SINGLE, ALIGNED, STATIC   XREF: 0 0920  4 0953  4 0966  4 0975
                                                           2 0978  2 0979
   11  FT_TO_NM                   SCALAR                   DOUBLE, ALIGNED, CONSTANT        XREF: 0 0011  2 0012  2 0804
                                                           2 0808
   82  G_LOAD                     SCALAR                   DOUBLE, ALIGNED, INITIAL  XREF: 0 0082  2 0532
  513  G_RUN_GUIDANCE             SCALAR                   SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0532
   13  G_TO_FPS2                  SCALAR                   DOUBLE, ALIGNED, CONSTANT        XREF: 0 0013  2 0014  2 0017
                                                           2 1039
  865  GAMMA                      SCALAR                   SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0865  2 0868
  857  GAMMA_VBAR                 SCALAR                   SINGLE, ALIGNED, STATIC   XREF: 0 0857  4 0887  4 0889  4 0891
                                                           4 0893  4 0895  2 0896
  754  GRAV_ACCEL                 3 - VECTOR               DOUBLE, TEMPORARY         XREF: 0 0754  4 0778  2 0779
 1003  GRAVRATIO                  SCALAR                   SINGLE, ALIGNED, STATIC   XREF: 0 1003  4 1011  2 1013  6 1014
                                                           2 1037
  521  GUID_PASS                  INTEGER                  SINGLE, ALIGNED, STATIC   XREF: 0 0521  2 0535  6 0537  2 0538
                                                           4 0539  4 0550
  513  GUID_PASS_LIM              SCALAR                   SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0538
  985  GO                         SCALAR                   SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0985  2 0988
 1002  H                          SCALAR                   DOUBLE, ALIGNED, STATIC   XREF: 0 1002  4 1006  2 1007  2 1008
                                                           2 1011  2 1012  6 1013  2 1017  2 1020  2 1023  2 1034
  526  H                          SCALAR                   **** SEE STRUCTURE TEMPLATE ATMOSPROP
  996  H_BASE                     SCALAR ARRAY             ARRAY(13), SINGLE, ALIGNED, STATIC, CONSTANT      XREF: 0 0996
                                                           2 1017  2 1020  2 1033  2 1034
  595  HEAT_RATE_CONTROL          PROCEDURE                  XREF: 2 0540  0 0595
  513  HS                         SCALAR                   SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0614
  755  HS_NORM_PRED               SCALAR                   SINGLE, TEMPORARY         XREF: 0 0755  NOT USED
  599  HS_2                       SCALAR                   SINGLE, ALIGNED, STATIC   XREF: 0 0599  4 0614  2 0615  2 0625
  665  I                          INTEGER                  SINGLE, TEMPORARY         XREF: 4 0665  2 0666  1 0681  1 0682
  922  I                          INTEGER                  SINGLE, ALIGNED, STATIC   XREF: 0 0922  4 0958  1 0962  1 0963
                                                           1 0964  1 0965  1 0966  4 0970  1 0971  1 0972  1 0973  1 0974
                                                           2 0975
  994  I                          INTEGER                  SINGLE, ALIGNED, STATIC   XREF: 0 0994  4 1016  1 1020  1 1021
                                                           1 1022  1 1025  1 1026  1 1028  1 1029  1 1033  1 1034  1 1035
                                                           1 1036  1 1037
  709  I_INPLANE                  3 - VECTOR               DOUBLE, ALIGNED, STATIC   XREF: 0 0709  4 0800  2 0801  2 0804
                                                           2 0805  2 0808
  756  I_LAT                      3 - VECTOR               SINGLE, TEMPORARY         XREF: 0 0756  4 0772  2 0773
  757  I_LIFT                     3 - VECTOR               SINGLE, TEMPORARY         XREF: 0 0757  4 0773  2 0774
  710  I_NORMAL                   3 - VECTOR               DOUBLE, ALIGNED, STATIC   XREF: 0 0710  4 0799  2 0800  2 0804
                                                           2 0808
  522  I_TARGET_EF                3 - VECTOR               DOUBLE, ALIGNED, STATIC   XREF: 0 0522  4 0567  2 0800  2 0801
                                                           2 0804
  758  I_VEL                      3 - VECTOR               SINGLE, TEMPORARY         XREF: 0 0758  4 0771  2 0772  2 0773
                                                           2 0774
```

```
DCL  NAME                           TYPE                      ATTRIBUTES & CROSS REFERENCE

542  INITIAL_GUID                   PROCEDURE                  XREF: 2 0529  0 0542
523  INITIALIZE_GUIDANCE            BIT(1)                    ALIGNED, STATIC, INITIAL  XREF: 0 0523  2 0527  4 0530
711  INTEG_LOOP                     SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0711  4 0750  2 0814
809  INTEGRATOR                     PROCEDURE                  XREF: 2 0780  0 0809
712  IR_E                           3 - VECTOR                DOUBLE, ALIGNED, STATIC   XREF: 0 0712  4 0797  2 0799  2 0805
                                                              2 0808
923  J                              INTEGER                   SINGLE, ALIGNED, STATIC   XREF: 0 0923  4 0938  1 0949  1 0950
                                                              1 0951  1 0952  1 0962  1 0963  1 0964  1 0965  1 0971  1 0972
                                                              1 0973  1 0974
957  K                              INTEGER                   SINGLE, TEMPORARY         XREF: 4 0957  2 0958  1 0959
 84  K_LOD_NAV                      SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0084  4 0569  6 0590  2 0593
                                                              2 0749
600  K_QDOT                         SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0600  4 0626  2 0629
601  K_QDOT_RATE                    SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0601  4 0627  2 0628
472  K_RHO_FILTER_GAIN              SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0472  2 0591
 83  K_RHO_NAV                      SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0083  4 0568  6 0591  2 0592
                                                              2 0764
990  KG_TO_LBM                      SCALAR                    SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0990  2 1039
 18  KG_TO_SLUG                     SCALAR                    DOUBLE, ALIGNED, CONSTANT         XREF: 0 0018  2 0873
602  K1_GAIN                        SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0602  4 0615  2 0625
603  K1_QDOT                        SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0603  4 0625  2 0626  2 0627
988  K2                             SCALAR                    SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0988  2 1026
                                                              2 1028  2 1036  2 1037
475  L_OVER_D_FILTER_GAIN           SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0475  2 0590
544  LAT                            SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0544  4 0556  6 0557  2 0558
513  LAT_TARGET                     SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0556
759  LIFT_ACCEL                     SCALAR                    SINGLE, TEMPORARY         XREF: 0 0759  4 0770  2 0774
577  LOD_MEAS                       SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0577  4 0588  2 0590
578  LOD_NOM                        SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0578  4 0585  2 0590
713  LOD_PRED                       SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0713  4 0749  2 0770
545  LONG                           SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0545  4 0555  2 0562  2 0566
513  LONG_TARGET                    SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0555
901  LOOKUP                         PROCEDURE                  XREF: 2 0584  2 0748  0 0901
 10  M_TO_FT                        SCALAR                    DOUBLE, ALIGNED, CONSTANT         XREF: 0 0010  2 0013  2 0869
                                                              2 0873
989  M_TO_FT                        SCALAR                    DOUBLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0989  2 1006
                                                              2 1007  2 1039
901  MACH                           SCALAR                    SINGLE, ALIGNED, INPUT-PARM       XREF: 0 0901  0 0906  2 0943
                                                              2 0969
579  MACH                           SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0579  4 0583  2 0584
851  MACH                           SCALAR                    SINGLE, ALIGNED, ASSIGN-PARM      XREF: 0 0851  0 0854  4 0871
                                                              4 0872  2 0896  2 0898
924  MACH_L                         SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0924  4 0969  2 0970  2 0975
925  MACH_MAX                       SCALAR                    SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0925  2 0943
                                                              2 0969
926  MACH_MIN                       SCALAR                    SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0926  2 0969
714  MACH_PRED                      SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0714  4 0747  2 0748
314  MASS_NAV                       SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0314  2 0589  2 0615  2 0769
867  MOLE_WT_ZERO                   SCALAR                    SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0867  2 0868
986  MO                             SCALAR                    SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0986  2 0988
1015 N                              INTEGER                   SINGLE, TEMPORARY         XREF: 4 1015  2 1016  1 1017
995  NSEGS                          INTEGER                   SINGLE, ALIGNED, STATIC, CONSTANT        XREF: 0 0995  2 0996
                                                              2 0997  2 0998  2 0999  2 1000  2 1001  2 1015
513  OMEGA_QDOT                     SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0616  2 0617
604  OMEGA_QDOT_SQUARED             SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0604  4 0616  2 0626
```

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
812	ORIG_POS	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0812 4 0816 2 0820 2 0826 2 0832 2 0836
813	ORIG_VEL	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0813 4 0817 2 0821 2 0827 2 0833 2 0837
25	PHI_CMD	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0025 4 0554 4 0636 6 0637 2 0641 6 0643 6 0646
524	PHI_DES	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0524 4 0552 2 0636 2 0669 2 0673 2 0677 6 0695 6 0696
513	PHI_DES_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0696
402	PHI_EI	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0402 2 0552 2 0554
513	PHI_MAX	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0554 2 0637 2 0646 2 0766
715	PHI_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0715 4 0765 6 0766 2 0767 2 0768
660	PHI_TRY	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0660 4 0669 4 0673 4 0677 2 0765
981	POLE	3 - VECTOR	DOUBLE, ALIGNED, INPUT-PARM XREF: 0 0981 0 0984 2 1005
661	PRED_EXIT	BIT(1)	ALIGNED, STATIC XREF: 0 0661 4 0790 2 0791
697	PREDICTOR	PROCEDURE	XREF: 2 0680 0 0697
605	QBAR	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0605 4 0624 2 0628 2 0629
606	QDOT	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0606 4 0610 2 0620 2 0621 2 0622 2 0629
513	QDOT_LIMIT	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0622 2 0629
607	QDOT_PAST	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0607 2 0620 4 0621
608	QDOT_RATE	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0608 4 0618 4 0620 2 0628
981	R	3 - VECTOR	DOUBLE, ALIGNED, INPUT-PARM XREF: 0 0981 0 0983 2 1004 2 1005
1002	R_MAG	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 1002 4 1004 2 1005 2 1006
717	R_MAG_PRED	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 0717 4 0728 2 0775 2 0777 2 0778 4 0781 2 0788
86	R_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0086 2 0582 2 0727
716	R_PRED	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0716 4 0727 2 0728 2 0731 2 0734 2 0772 2 0775 2 0781 2 0783 2 0785 2 0788 2 0797 2 0798 2 0816 4 0820 4 0826 4 0832 4 0836
6	RAD_TO_DEG	SCALAR	DOUBLE, ALIGNED, CONSTANT XREF: 0 0006 2 0643
718	RDOT_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0718 4 0788 2 0789
858	REYNOLDS_NUMBER	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0858 4 0875 4 0876 2 0877 2 0898
526	RHO	SCALAR	**** SEE STRUCTURE TEMPLATE ATMOSPROP
999	RHO_BASE	SCALAR ARRAY	ARRAY(13), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0999 2 1026 2 1029 2 1037
580	RHO_MEAS	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0580 4 0589 2 0591
525	RHO_NAV	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0525 4 0592 2 0610 2 0624
760	RHO_PRED	SCALAR	SINGLE, TEMPORARY XREF: 0 0760 4 0764 2 0769
513	RHO_SL	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0615
987	RR	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0987 2 0988
993	R0	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0993 2 1011 2 1033 2 1034 2 1036
478	S_REF	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0478 2 0589 2 0615 2 0769
927	SIN_ALPHA	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0927 NOT USED
844	SLAMBDA	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 0844 4 0846 2 0847
859	SPEED_OF_SOUND	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0859 4 0869 2 0870 2 0872
868	SPEED_OF_SOUND_CONST	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0868 2 0869
761	SPHI	SCALAR	SINGLE, TEMPORARY XREF: 0 0761 4 0768 2 0773
1002	SPSI	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 1002 4 1005 2 1006

DCL	NAME	TYPE	ATTRIBUTES & CROSS REFERENCE
841	T	SCALAR	DOUBLE, ALIGNED, INPUT-PARM XREF: 0 0841 0 0842 2 0845
			2 0846
998	T_BASE	SCALAR ARRAY	ARRAY(13), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0998
			2 1022
310	T_EPOCH	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0310 2 0845 2 0846
22	T_GMT	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0022 2 0726
719	T_PRED	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 0719 4 0726 6 0787 2 0796
860	T_PRIME	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0860 4 0896 2 0897
861	T_WALL	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0861 4 0881 4 0883 4 0885
			2 0896
1003	TEMPRATIO	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 1003 4 1035 2 1037
736	TIME_INCREMENT	INTEGER	SINGLE, TEMPORARY XREF: 4 0736 2 0737
526	TM	SCALAR	**** SEE STRUCTURE TEMPLATE ATMOSPROP
997	TM_BASE	SCALAR ARRAY	ARRAY(13), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0997
			2 1021 2 1026 2 1029 2 1033 2 1035
700	TOTAL_TIME_STEPS	INTEGER	SINGLE, ALIGNED, STATIC XREF: 0 0700 4 0737 NOT REFERENCED
526	TS	SCALAR	**** SEE STRUCTURE TEMPLATE ATMOSPROP
609	TWO_ZETA_OMEGA	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0609 4 0617 2 0627
762	U_PRED	3 - VECTOR	DOUBLE, TEMPORARY XREF: 0 0762 4 0775 2 0776 6 0777
			2 0778
866	UNIV_GAS_CONST	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0866 2 0868
981	USATMOS62	PROCEDURE	XREF: 2 0582 2 0734 2 0785 0 0981
901	V_BAR	SCALAR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0901 0 0907 2 0956
851	V_BAR	SCALAR	SINGLE, ALIGNED, ASSIGN-PARM XREF: 0 0851 0 0855 4 0878
			4 0898 NOT REFERENCED
581	V_BAR	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0581 4 0583 2 0584
928	V_BAR_L	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0928 4 0956 2 0959 2 0966
929	V_BAR_MAX	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0929 2 0956
930	V_BAR_MIN	SCALAR	SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0930 2 0956
720	V_BAR_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0720 4 0747 2 0748
931	V_BAR_TABLE	SCALAR ARRAY	ARRAY(8), SINGLE, ALIGNED, STATIC, CONSTANT XREF: 0 0931
			2 0959 2 0966
513	V_FINAL_MAG	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0636 2 0765
513	V_MAG_CHANGE	SCALAR	SINGLE, ALIGNED, INITIAL XREF: 0 0513 2 0636 2 0765
721	V_MAG_PRED	SCALAR	DOUBLE, ALIGNED, STATIC XREF: 0 0721 4 0730 2 0765 4 0782
93	V_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0093 2 0729
94	V_NAV_MAG	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0094 2 0636
722	V_PRED	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0722 4 0729 2 0730 2 0731
			2 0782 2 0783 2 0788 2 0798 2 0817 2 0818 2 0820 4 0821
			2 0824 2 0826 4 0827 2 0830 2 0832 4 0833 2 0836 4 0837
851	V_REL_MAG	SCALAR	SINGLE, ALIGNED, INPUT-PARM XREF: 0 0851 0 0852 2 0872
			2 0876
95	V_REL_MAG	SCALAR	DOUBLE, ALIGNED, INITIAL XREF: 0 0095 2 0583 2 0586 2 0589
			2 0610 2 0624 2 0625
723	V_REL_MAG_PRED	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0723 4 0732 2 0747 2 0769
			2 0771 4 0784
96	V_REL_NAV	3 - VECTOR	DOUBLE, ALIGNED, INITIAL XREF: 0 0096 2 0586
724	V_REL_PRED	3 - VECTOR	SINGLE, ALIGNED, STATIC XREF: 0 0724 4 0731 2 0732 2 0771
			4 0783 2 0784
862	VISCOSITY	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0862 4 0873 2 0874 2 0876
725	VR_E	3 - VECTOR	DOUBLE, ALIGNED, STATIC XREF: 0 0725 4 0798 2 0799
480	WE_NAV	3 - VECTOR	SINGLE, ALIGNED, INITIAL XREF: 0 0480 2 0731 2 0783 2 0798
546	X	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0546 4 0559 2 0560 4 0561
			6 0562 2 0563 2 0567
547	Y	SCALAR	SINGLE, ALIGNED, STATIC XREF: 0 0547 4 0563 2 0564 4 0565

```
HAL/S  STD 360-24.20          I N T E R M E T R I C S ,   I N C .           APRIL 27, 1987    14:13:28.85

DCL  NAME                             TYPE                     ATTRIBUTES & CROSS REFERENCE

                                                               6 0566  2 0567
548  Z                               SCALAR                    SINGLE, ALIGNED, STATIC   XREF: 0 0548  4 0558  2 0559  2 0563
                                                               2 0567
763  Z_PRED                          SCALAR                    DOUBLE, TEMPORARY         XREF: 0 0763  4 0776  2 0777
513  ZETA_QDOT                       SCALAR                    SINGLE, ALIGNED, INITIAL  XREF: 0 0513  2 0617
```

LIST OF REFERENCES

1. Freeman, D.C., Powell, R.W., Naftel, J.C., and Wurster, K.E., "Definition of an Entry Research Vehicle", AIAA Paper 85-0969, AIAA 20th Thermophysics Conference, Williamsburg, Virginia, June 1985.

2. Morth, R., *Reentry Guidance for Apollo*, MIT Document R-532, January 1966.

3. *Shuttle Level C FSSR, Part A, Entry Through Landing Guidance*, STS 83-0001A, December 15, 1982.

4. Lerner, E.J., "Supercomputer in a Six-Inch Cube", *Aerospace America*, Vol. 4, No. 11, November 1986, pp. 13-14.

5. Brand, T.J., Brown, D.W., and Higgins, J.P., *Space Shuttle G&N Equation Document No. 24 (Revision 2), Unified Powered Flight Guidance*, CSDL C-4108, June 1974.

6. Higgins, J.P., "An Aerobraking Guidance Concept for a Low L/D AOTV", CSDL Memo No. 10E-84-04, May 24, 1984.

7. Young, J.D., Underwood, J.M., et al., "The Aerodynamic Challenges of the Design and Development of the Space Shuttle Orbiter", *Space Shuttle Technical Conference*, NASA CP-2342 Part 1, held at the Lyndon B. Johnson Space Center, Houston, Texas, June 28-30, 1983.

8. *Terrestrial Environment (Climatic) Criteria Guidelines for Use in Aerospace Vehicle Development, 1982 Revision*, NASA TM 82473, June 1982.

9. Justus, C.G., Fletcher, G.R., Gramling, F.E., and Pace, W.B., *The NASA/MSFC Global Reference Atmosphere Model - MOD 3, (with Spherical Harmonic Wind Model)*, NASA Contractor Report 3256, March 1980.

10. Findley, J.T., Kelly, G.M., and Troutman, P.A., *Final Report - Shuttle Derived Atmospheric Density Model*, NASA Contractor Report 171824, November 1986.

11. Bryson, A.E., Mikami, K., and Battle, C.T., "Optimum Lateral Turns for a Re-Entry Glider", *Aerospace Engineering*, March, 1962, pp. 18-23.

12. Wagner, W.E., "Roll Modulation for Maximum Re-Entry Lateral Range", *Journal of Spacecraft*, Vol. 2, No. 5, September-October 1965.

13. Zondervan, K., Bauer, T., Betts, J., and Huffman, W., "Solving the Optimal Control Problem Using a Nonlinear Programming Technique Part 3: Optimal Shuttle Reentry Trajectories", AIAA Paper 84-2039, AIAA/AAS Astrodynamics Conference, Seattle, Washington, August 1984.

14. Powell, R.W., Naftel, J.C., and Cunningham, M.J., "Performance Evaluation of an Entry Research Vehicle", AIAA Paper 86-0270, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 1986.

15. *Program to Optimize Simulated Trajectories (POST) Volume I - Formulation Manual*, NASA CR-132689, April 1975.

16. *U.S. Standard Atmosphere, 1962*, Prepared under the sponsorship of NASA, U.S. Air Force, and U.S. Weather Bureau, Published by the US Government Printing Office, 1962.

17. Battin, R.H., *An Introduction to the Mathematics and Methods of Astrodynamics*, forthcoming book, 1987.

18. Scott, C.D., et al., "The Aerothermodynamics and Thermal Protection System Challenges for an Aerobraking Orbital Transfer Vehicle", Presented at the NASA Symposium on Recent Advances in TPS and Structures for Future Space Transportation Systems, held on December 13-15, 1983.

19. *Aeroassist Flight Experiment, AFESIM All-Digital Functional Simulator, Simulator Definition Documentation, Volume 1*, CSDL-R-1951, March 27, 1987.

20. Phillips, R.E., "Three Aspects of Covariance/Monte Carlo Analysis: Bias, Reducing Dimensionality and Systematically Choosing Error Vectors", CSDL Memo No. 10E-86-18, June 23, 1986.