IN-05-CR
153100
P.61

DEVELOPMENT

OF A

NON-LINEAR SIMULATION

FOR

GENERIC HYPERSONIC VEHICLES

-ASUHS1

Juan Salas, T. Alan Lovell, and David K. Schmidt

Aerospace Research Center
College of Engineering and Applied Sciences
Arizona State University
Tempe, AZ 85287-8006

(NASA-CR-192710) DEVELOPMENT OF A NON-LINEAR SIMULATION FOR GENERIC HYPERSONIC VEHICLES - ASUHS1 Interim Task Report (Arizona State Univ.) 61 p

N93-22003

Unclas

G3/05 0153100

Interim Task Report

Grant No. NAG 1-1341

NASA Langley Research Center
Hampton, VA

Ms. Irene Gregory
Technical Monitor

Report ARC93-1

February, 1993

ABSTRACT

A nonlinear simulation is developed to model the longitudinal motion of a vehicle in hypersonic flight. The equations of motion pertinent to this study are presented. Analytic expressions for the aerodynamic forces acting on a hypersonic vehicle which were obtained from Newtonian Impact Theory are further developed. The control surface forces are further examined to incorporate vehicle elastic motion. The purpose is to establish feasible equations of motion which combine rigid body, elastic and aeropropulsive dynamics for use in nonlinear simulations. The software package SIMULINK® is used to implement the simulation. Also discussed are issues needing additional attention and potential problems associated with the implementation (with proposed solutions).

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The object of the Hypersonic Vehicle Simulation ASUHS1 is to integrate the equations of motion for a flexible hypersonic vehicle in atmospheric flight. The derivation of the equations of motion is fully documented in Reference [1]. An aeroelastic model is presented in Reference [2]. This class of vehicle has been shown to exhibit a high degree of coupling among the aerodynamic, propulsive, and elastic effects. It also exhibits highly unstable pitch behavior [4].

What follows is a discussion of ASUHS1, which simulates the dynamics of the hypersonic vehicle model using the simulation software package SIMULINK® [3]. The nonlinear simulation developed in this report will serve as a tool to better understand hypersonic vehicle dynamics. For this simulation, only motion in a vertical plane (symmetric flight) is considered.

Reference [2] includes analytical expressions for the linearized aerodynamic and propulsive forces on the vehicle. The total aerodynamic forces on the lower forebody, however, are expressed in terms of (nonlinear) integral equations. In this report, these integral equations are solved in closed form and used in the simulation. In addition, the equations for the aerodynamic control surface forces and moments are modified to incorporate the effect of vehicle bending not included in [2].

The organization of this report is as follows: Chapter 2 first outlines the structure of ASUHS1. Chapter 3 presents the equations of motion for this class of vehicle and simplifies them for motion restricted to the vertical plane. In Chapter 4 the aeropropulsive force equations are presented. The computational procedure is discussed in Chapter 5, with numerical data given in Chapter 6. Potential problems of which the user should be aware complete this report.

2. STRUCTURE OF SIMULATION

The simulation structure is best explained by referring to Figure 1. The two basic tasks of ASUHS1 are to compute the aeropropulsive forces acting on the vehicle and to integrate the equations of motion for this class of vehicle. The first of the above tasks is carried out in the block **FORCES**, while the second is done in the block **EOM**. Each of these blocks consists of several sub-blocks, all of which utilize common elements found in the SIMULINK® library and also coded functions (M-files) found in the local directory. These MATLAB® functions are fully documented in Appendix A.

The initialization procedure for ASUHS1 consists of running two script files, *inpar* and *initstate*. The first file initializes the parameter values associated with the spherical rotating earth and atmospheric models. These values will generally remain the same regardless of the study being conducted. The vehicle's length, elastic and mass properties, and engine parameters, which may vary from study to study, are also defined in this script file. The second file initializes the initial conditions and control settings. These values may be altered as well, according to the desired flight condition.

The aeropropulsive forces (and moments) are categorized as: 1)Aerodynamic forces on the vehicle's lower forebody, 2)Aerodynamic forces on the vehicle's control surface, 3)Engine thrust forces, 4)External nozzle exhaust forces. These forces are computed from separate functions of vehicle states and controls. Calculation of the these forces includes the effect of the vehicle's elastic motion, which changes the angle of the lower forebody and lower aft-body, as well as control surface position relative to the vehicle's center of gravity. The forces are summed just before leaving the **FORCES** block.

The block **EOM** makes use of the vehicle equations of motion over a spherical rotating earth [1]. This block utilizes the above forces and moments, plus the gravitational "force." The state derivatives are calculated and numerically integrated. The method of numerical integration is chosen by the user from SIMULINK®'s simulation parameters.

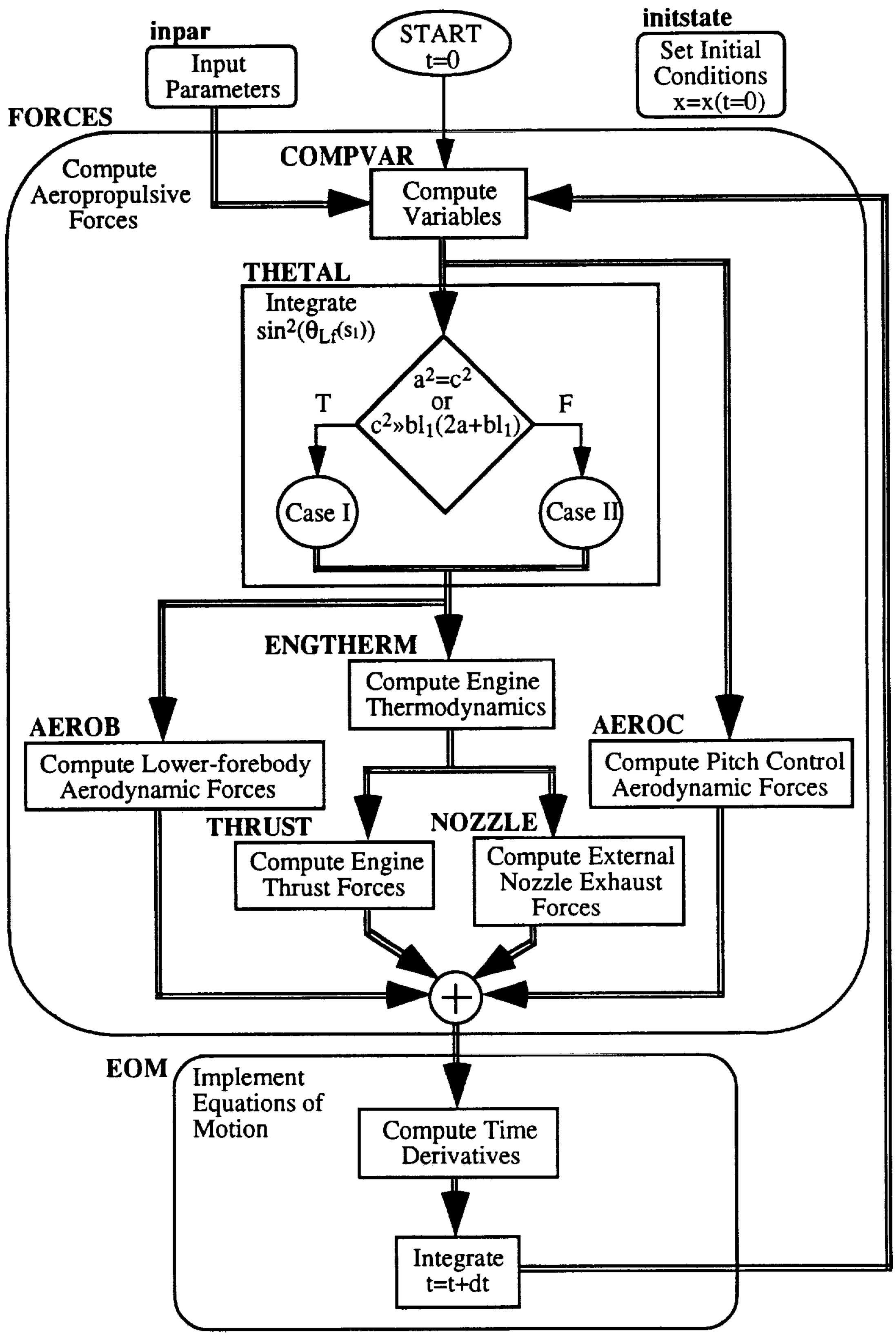

Figure 1. Hypersonic Vehicle Simulation block structure.

3. EQUATIONS OF MOTION

An integrated model for elastic hypersonic vehicles was rigorously developed in Reference [1]. This approach accounted for the interaction between rigid body motion, elastic deformation, fluid flow, rotating machinery, wind and a spherical rotating earth. What resulted were 12+n equations of motion: the 3 force equations + 3 moment equations + n elastic deformation equations + 3 kinematic equations + 3 Euler angle equations. In this simulation only symmetric motion (i.e., motion in the vertical plane) is currently considered.

For this initial development of the simulation, many of the interactions accounted for in [1] are neglected. For example, the effects of rotating machinery are neglected. Forces associated with mass flow effects not directly contributing to engine thrust are, at this time, considered negligible in comparison to the aeropropulsive forces; the corresponding moments are also neglected. A no-wind assumption eliminates all the wind velocity components. Finally, the vehicle is assumed to possess a plane of symmetry that passes through the body x-axis and z-axis with constant moments of inertia. These assumptions simplify the force equations (4.1-17) to (4.1-19), and moment equations (4.2-8) to (4.2-10) in [1].

To further reduce the 12+n equations of motion, aeropropulsive roll and yaw moments and body y-axis force (lateral force) are assumed absent [2]. Secondly, in ASUHS1 a wings-level flight condition is currently assumed. These two facts eliminate roll motion (ϕ=p=0) and yaw rate (r=0) from the present simulation. In addition, only one elastic degree of freedom is considered. This leaves three force equations, one moment equation and one elastic deformation equation as the field equations to be implemented. They are the body x-axis force equation (4.1-17) [1], y-axis force equation (4.1-18) [1], z-axis force equation (4.1-19) [1], y-axis moment equation (4.2-9) [1] and the first generalized elastic deformation equation (4.3-2) [1]. These equations are listed below:

$$\begin{aligned}\dot{u} = &-wq - \omega_e[w(T_{21}\cos\lambda - T_{23}\sin\lambda) - v(T_{31}\cos\lambda - T_{33}\sin\lambda)] \\ &- R\omega_e^2[T_{11}\frac{\sin 2\lambda}{2} + T_{13}\cos^2\lambda] + g_o\left(\frac{R_e}{R}\right)^2 T_{13} + \frac{X}{m}\end{aligned} \tag{3.1}$$

$$\begin{aligned}\dot{v} = &-\omega_e[u(T_{31}\cos\lambda - T_{33}\sin\lambda) - w(T_{11}\cos\lambda - T_{13}\sin\lambda)] \\ &- R\omega_e^2[T_{21}\frac{\sin 2\lambda}{2} + T_{23}\cos^2\lambda] + g_o\left(\frac{R_e}{R}\right)^2 T_{23} + \frac{Y}{m}\end{aligned} \tag{3.2}$$

$$\dot{w} = uq - \omega_e[v(T_{11}\cos\lambda - T_{13}\sin\lambda) - u(T_{21}\cos\lambda - T_{23}\sin\lambda)] - R\omega_e^2[T_{31}\frac{\sin 2\lambda}{2} + T_{33}\cos^2\lambda] + g_o\left(\frac{R_e}{R}\right)^2 T_{33} + \frac{Z}{m} \tag{3.3}$$

$$\dot{q} = \frac{M}{I_{yy}} \tag{3.4}$$

$$\ddot{\eta} = -\omega_1^2\eta - 2\zeta_1\omega_1\dot{\eta} + \frac{Q_{E1}}{m_{E1}} \tag{3.5}$$

where,
- ω_e = Earth rotation rate about its axis
- R_e = Earth radius
- R = Distance from vehicle c.g. to Earth's center ($R=R_e+h$)
- g_o = Gravitational acceleration at the Earth's surface
- λ = Latitude of vehicle position
- m = Vehicle mass (per unit width of the vehicle)
- I_{yy} = Vehicle moment of inertia about body y-axis (per unit width of the vehicle)
- ω_1 = Undamped natural frequency of 1st generalized elastic degree of freedom
- ζ_1 = Damping ratio of 1st generalized elastic degree of freedom
- m_{E1} = Generalized mass of 1st generalized elastic degree of freedom (per unit width of the vehicle)
- T = Co-ordinate transformation matrix from vehicle-carrying frame (X_V, Y_V, Z_V) to body frame (X_B, Y_B, Z_B), see Figure 2. The entries of the co-ordinate transformation matrix T may be written in terms of the quaternion components described in [1], or in terms of the roll, pitch and yaw angles (ϕ, θ, ψ). Both representations for T are shown below:

$$T = \begin{bmatrix} \beta_1^2-\beta_2^2-\beta_3^2+\beta_4^2 & 2(\beta_1\beta_2+\beta_3\beta_4) & 2(\beta_1\beta_3-\beta_2\beta_4) \\ 2(\beta_1\beta_2-\beta_3\beta_4) & -\beta_1^2+\beta_2^2-\beta_3^2+\beta_4^2 & 2(\beta_2\beta_3+\beta_1\beta_4) \\ 2(\beta_1\beta_3+\beta_2\beta_4) & 2(\beta_2\beta_3-\beta_1\beta_4) & -\beta_1^2-\beta_2^2+\beta_3^2+\beta_4^2 \end{bmatrix}$$

$$T = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}$$

The following equations may be used to determine the quaternions from the Euler angles:

$$\beta_1 = -\frac{\sin(\theta)\sin(\psi)}{4\cos(\theta/2)\cos(\theta/2)}$$

$$\beta_2 = \frac{\sin(\theta)\cos(\psi/2)}{2\cos(\theta/2)}$$

$$\beta_3 = \frac{\sin(\psi)\cos(\theta/2)}{2\cos(\psi/2)}$$

$$\beta_4 = \cos(\theta/2)\cos(\psi/2)$$

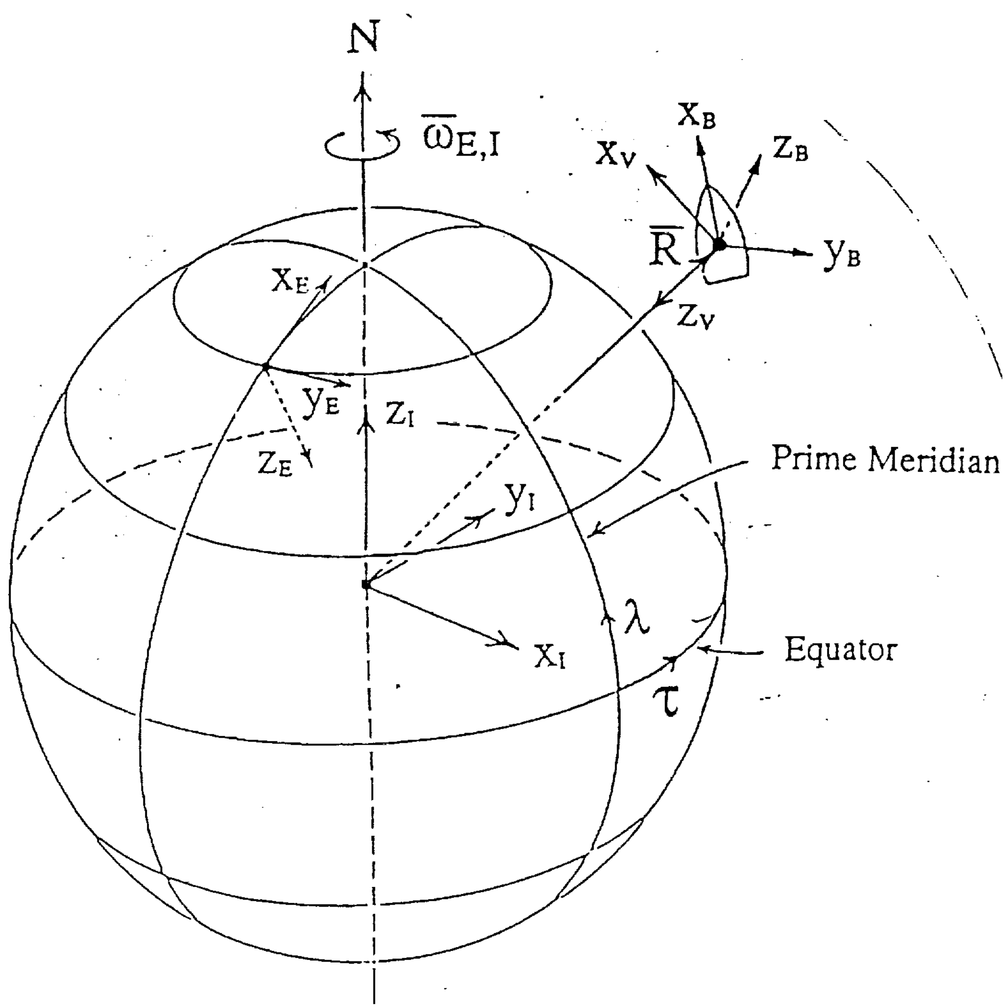

Figure 2. Vehicle trajectory variables and coordinate frames [1].

The implication of symmetric flight constrains motion to a vertical plane fixed in an inertial reference (in this case, Earth fixed center). But we will require for now that the vehicle's lateral velocity in the "E" frame (see Figure 2) equal zero (v=0) for all time. Obviously, the initial condition and constraint equation which meet the requirements of symmetric flight are:

$$v(t=0) = v_o = 0$$

$$\dot{v}(t) = 0, \qquad \forall\ t>0 \tag{3.6}$$

To implement equation (3.6) requires that the applied lateral force, Y, cancel the Coriolis and centripetal acceleration terms in equation (3.2). (Note that the gravity term in equation (3.2) is zero because $T_{23}=0$ for wings level.) In conventional aircraft dynamics these terms are usually neglected since the vehicle speed is relatively small. This is not necessarily the case in hypersonic flight; consequently the Coriolis and centripetal acceleration terms are taken into account. In light of this unorthodox circumstance, Y is utilized in this initial model as a constraining force to ensure symmetric flight, and the lateral force equation is eliminated. The simplified force equations are:

$$\begin{aligned}\dot{u} = -w(q + \omega_e T_{21}\cos\lambda) - R\omega_e^2[T_{11}\frac{\sin 2\lambda}{2} + T_{13}\cos^2\lambda] \\ + g_o\left(\frac{R_e}{R}\right)^2 T_{13} + \frac{X}{m}\end{aligned} \tag{3.7}$$

$$\begin{aligned}\dot{w} = u(q + \omega_e T_{21}\cos\lambda) - R\omega_e^2[T_{31}\frac{\sin 2\lambda}{2} + T_{33}\cos^2\lambda] \\ + g_o\left(\frac{R_e}{R}\right)^2 T_{33} + \frac{Z}{m}\end{aligned} \tag{3.8}$$

The aeropropulsive forces and moments (X, Z, M and Q_{E1}) which appear in equations (3.7), (3.8), (3.4) and (3.5), respectively, are expressed in terms of the dynamic variables and controls. These expressions are examined in Chapter 4. The twelve dynamic variables to be integrated with three control inputs are:

h = Geometric altitude - [ft]
u = Vehicle forward speed - [ft/s] (relative to "E" frame)
w = Vehicle sink rate - [ft/s] (relative to "E" frame)
q = Vehicle rigid body pitch rate - [rad/s]
η = First generalized elastic co-ordinate - [-]
$\dot{\eta}$ = First generalized elastic co-ordinate rate - [1/s]
λ = Latitude of vehicle position - [rad]

$\tau =$ Longitude of vehicle position - [rad]
$\beta_1 =$ First quaternion component - [-]
$\beta_2 =$ Second quaternion component - [-]
$\beta_3 =$ Third quaternion component - [-]
$\beta_4 =$ Fourth quaternion component - [-]

$\delta =$ Pitch control surface deflection - [rad]
$\overline{A}_D =$ Diffuser area ratio control - [-]
$T_o =$ Combustor total temperature change control - [R°]

In addition, three kinematic equations, one differential identity and four quaternion equations are necessary to describe the rates of change of the altitude, latitude, longitude, first generalized elastic co-ordinate and quaternions:

$$\dot{h} = -[T_{13}u + T_{33}w] \tag{3.9}$$

$$\dot{\lambda} = \frac{1}{R}[T_{11}u + T_{31}w] \tag{3.10}$$

$$\dot{\tau} = \frac{\sec\lambda}{R}[T_{12}u + T_{32}w] \tag{3.11}$$

$$\dot{\eta} = \dot{\eta} \tag{3.12}$$

$$\dot{\beta}_1 = \frac{1}{2}(p_v\beta_4 - q_v\beta_3 + r_v\beta_2) \tag{3.13}$$

$$\dot{\beta}_2 = \frac{1}{2}(p_v\beta_3 + q_v\beta_4 - r_v\beta_1) \tag{3.14}$$

$$\dot{\beta}_3 = \frac{1}{2}(-p_v\beta_2 + q_v\beta_1 + r_v\beta_4) \tag{3.15}$$

$$\dot{\beta}_4 = \frac{1}{2}(-p_v\beta_1 - q_v\beta_2 - r_v\beta_3) \tag{3.16}$$

where,

$$p_v = \frac{1}{R}[uT_{12}T_{13}\tan\lambda + w(T_{13}T_{32}\tan\lambda + T_{12}T_{31} - T_{11}T_{32})] + \omega_e(T_{13}\sin\lambda - T_{11}\cos\lambda)$$

$$q_v = q + \frac{1}{R}[u(T_{11}T_{22} - T_{12}T_{21}) + w(T_{22}T_{31} - T_{21}T_{32})] - \omega_e T_{21}\cos\lambda$$

$$r_v = \frac{1}{R}[u(T_{12}T_{33}\tan\lambda + T_{11}T_{32} - T_{12}T_{31}) + wT_{32}T_{33}\tan\lambda] + \omega_e(T_{33}\sin\lambda - T_{31}\cos\lambda)$$

The block **EOM** determines the right-hand sides of equations (3.4)-(3.16) for time integration.

4. AEROPROPULSIVE FORCES AND MOMENTS

4.1. Introduction

Analytical expressions for the aerodynamic and propulsive forces acting on a hypersonic vehicle have been developed in [2]. Also, a linearized model has been obtained for modal analysis and control law design [2, 4]. For implementation purposes, the integral expressions in the model may be solved in closed form.

For completeness, the expressions for the aeropropulsive forces are rewritten here from [2]. Recall they are composed of four separate constituents: 1)Aerodynamic forces on the forebody, 2)Aerodynamic forces on the control surface, 3)Engine thrust, 4)External nozzle exhaust.

1) $$X_A=-\{P_\infty l_1+q_\infty C_{pn}\int_0^{l_1} \sin^2\theta_{Lf}(s_1)ds_1\}\cdot\sin(\tau_1+\Delta\tau_1\eta) \tag{4.1-1}$$

$$Z_A=P_\infty L-\{P_\infty l_1+q_\infty C_{pn}\int_0^{l_1} \sin^2\theta_{Lf}(s_1)ds_1\}\cdot\cos(\tau_1+\Delta\tau_1\eta) \tag{4.1-2}$$

$$M_A=\{\frac{P_\infty l_1^2}{2}+q_\infty C_{pn}\int_0^{l_1} s_1\sin^2\theta_{Lf}(s_1)ds_1\}-[(h-z_{cg})\sin(\tau_1+\Delta\tau_1\eta)+(L_1-x_{cg})\cos(\tau_1+\Delta\tau_1\eta)]\cdot$$
$$\{P_\infty l_1+q_\infty C_{pn}\int_0^{l_1} \sin^2\theta_{Lf}(s_1)ds_1\} - P_\infty L(x_{cg}-\frac{L}{2}) \tag{4.1-3}$$

$$Q_A=\Delta\tau_1\cdot\{\frac{P_\infty l_1^2}{2}+q_\infty C_{pn}\int_0^{l_1} s_1\sin^2\theta_{Lf}(s_1)ds_1\} \tag{4.1-4}$$

2) $$X_{cs}=-q_\infty C_{pn}\sin^2(\alpha+\delta)\sin(\delta)\frac{S_{cs}}{b} \tag{4.1-5}$$

$$Z_{cs}=-q_\infty C_{pn}\sin^2(\alpha+\delta)\cos(\delta)\frac{S_{cs}}{b} \tag{4.1-6}$$

$$M_{cs}=X_{cs}z_{cs}-Z_{cs}x_{cs} \tag{4.1-7}$$

$$Q_{cs}=0 \tag{4.1-8}$$

3) $$X_T = \{[\gamma P_e M_e^2 + (P_e - P_\infty)] - \frac{[\gamma P_1 M_1^2 + (P_1 - P_\infty)]}{\bar{A}_D \bar{A}_N}\} \cdot \frac{A_e}{b} \qquad (4.1\text{-}9)$$

$$Z_T = 0 \qquad (4.1\text{-}10)$$

$$M_T = X_T (h - z_{cg}) \qquad (4.1\text{-}11)$$

$$Q_T = 0 \qquad (4.1\text{-}12)$$

4) $$X_E = P_\infty l_2 \left[\frac{\bar{P} \ln(\bar{P})}{(\bar{P}-1)}\right] \cdot \sin(\tau_2 + \Delta\tau_2 \eta) \qquad (4.1\text{-}13)$$

$$Z_E = -P_\infty l_2 \left[\frac{\bar{P} \ln(\bar{P})}{(\bar{P}-1)}\right] \cdot \cos(\tau_2 + \Delta\tau_2 \eta) \qquad (4.1\text{-}14)$$

$$M_E = P_\infty \cdot \left\{ l_2 r_2 \left[\frac{\bar{P} \ln(\bar{P})}{(\bar{P}-1)}\right] - l_2^2 \frac{\bar{P}}{(\bar{P}-1)} \left[1 - \frac{\ln(\bar{P})}{(\bar{P}-1)}\right] \right\} \qquad (4.1\text{-}15)$$

$$Q_E = \Delta\tau_2 P_\infty l_2^2 \frac{\bar{P}}{(\bar{P}-1)} \left[1 - \frac{\ln(\bar{P})}{(\bar{P}-1)}\right] \qquad (4.1\text{-}16)$$

where, $\bar{P} = \frac{P_e}{P_\infty}$

$$r_2 = (h - z_{cg}) \sin(\tau_2 + \Delta\tau_2 \eta) - (L_1 - x_{cg}) \cos(\tau_2 + \Delta\tau_2 \eta)$$

4.2. Algebraic Solutions

4.2.1 Explicit Expressions for Local Flow Deflection Angles

The pressure distribution on a hypersonic vehicle is strongly dependent upon the air flow deflection angle [2]. (See Figure 3.) The lower forebody of the vehicle deflects the air flow an angle θ. Due to vehicle pitch rate and elastic effects, this angle is dependent upon the position s_1 along the vehicle's lower forebody. Expressing the local velocity as a function of s_1, an explicit expression for the local flow deflection angle $\theta_{Lf}(s_1)$ is obtained. Similarly, the control surface deflects the air flow by an angle θ which is dependent upon control surface location. (See Figure 3.) Here the control surface deflection rate $\dot{\delta}$ and its chord length are assumed negligible. This allows the local flow deflection angle at the control surface θ_{Lcs} to be considered constant for the entire surface.

In computing the forces on the vehicle, the trigonometric sine of each flow deflection angle is sought rather than the angle itself. These quantities are defined as:

$$\sin(\theta_{Lf}(s_1))=\frac{\vec{\mathbf{V}}_{Lf}(s_1)\cdot\hat{\mathbf{n}}_f}{\left|\vec{\mathbf{V}}_{Lf}(s_1)\right|}\qquad\qquad \sin(\theta_{Lcs})=\frac{\vec{\mathbf{V}}_{Lcs}\cdot\hat{\mathbf{n}}_{cs}}{\left|\vec{\mathbf{V}}_{Lcs}\right|}\qquad\qquad(4.2\text{-}1)$$

For the lower forebody:

$$\vec{\mathbf{V}}_{Lf}(s_1)=\vec{\mathbf{V}}_{\infty}+\vec{\mathbf{V}}_{q}(s_1)+\vec{\mathbf{V}}_{\dot{\eta}}(s_1),\quad \hat{\mathbf{n}}_f=\sin(\tau_1+\Delta\tau_1\eta)\hat{\mathbf{i}}+\cos(\tau_1+\Delta\tau_1\eta)\hat{\mathbf{k}}$$

where,
$$\vec{\mathbf{V}}_{\infty}=V_{\infty}\cos(\alpha)\hat{\mathbf{i}}+V_{\infty}\sin(\alpha)\hat{\mathbf{k}}$$
$$\vec{\mathbf{V}}_{q}(s_1)=\vec{\omega}_{rb}\times\vec{\mathbf{r}}(s_1);\quad \vec{\omega}_{rb}=q\hat{\mathbf{j}},\quad \vec{\mathbf{r}}(s_1)=\{s_1\cos(\tau_1+\tau_1\eta)-x_1\}\hat{\mathbf{i}}+\{z_1-s_1\sin(\tau_1+\tau_1\eta)\}\hat{\mathbf{k}}$$
$$\vec{\mathbf{V}}_{\dot{\eta}}(s_1)=-s_1\Delta\tau_1\dot{\eta}\hat{\mathbf{n}}$$

Thus the local velocity at the lower forebody $\vec{\mathbf{V}}_{Lf}(s_1)$ is:

$$\vec{\mathbf{V}}_{Lf}(s_1)=\{V_{\infty}\cos(\alpha)+z_1q-s_1(q+\Delta\tau_1\dot{\eta})\sin(\tau_1+\Delta\tau_1\eta)\}\hat{\mathbf{i}}+$$
$$\{V_{\infty}\sin(\alpha)+x_1q-s_1(q+\Delta\tau_1\dot{\eta})\cos(\tau_1+\Delta\tau_1\eta)\}\hat{\mathbf{k}}$$

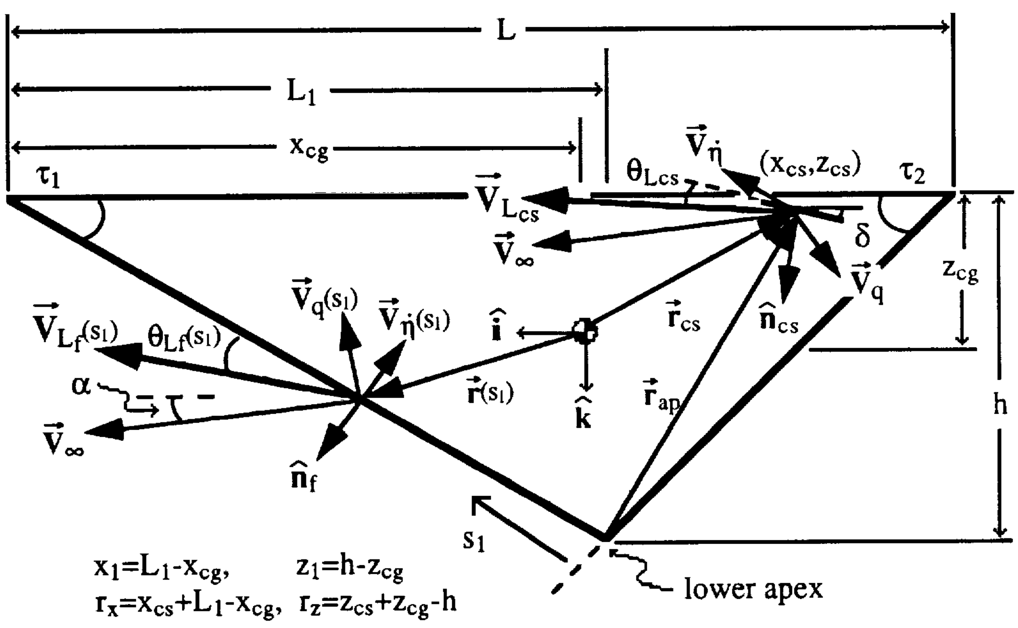

Figure 3. Geometric description of the hypersonic vehicle velocities.

and $\sin^2(\theta_{Lf}(s_1))$ is determined from equation (4.2-1):

$$\sin^2(\theta_{Lf}(s_1)) = \frac{a^2+2abs_1+b^2s_1^2}{c^2+2abs_1+b^2s_1^2} = 1 + \frac{a^2-c^2}{c^2+2abs_1+b^2s_1^2} \tag{4.2-2}$$

where, $a = V_\infty\sin(\alpha+\tau_1+\Delta\tau_1\eta)+q\{z_1\sin(\tau_1+\Delta\tau_1\eta)+x_1\cos(\tau_1+\Delta\tau_1\eta)\}$
$b = -(q+\Delta\tau_1\dot{\eta})$
$c^2 = \{V_\infty\cos(\alpha)+qz_1\}^2+\{V_\infty\sin(\alpha)+qx_1\}^2$

For the control surface:

$$\vec{V}_{Lcs} = \vec{V}_\infty + \vec{V}_q + \vec{V}_{\dot{\eta}}, \quad \hat{\mathbf{n}}_{cs}=\sin(\delta-\Delta\tau_2\eta)\hat{\mathbf{i}} + \cos(\delta-\Delta\tau_2\eta)\hat{\mathbf{k}}$$

where, $\vec{V}_q=\vec{\omega}_{rb}\times\vec{r}_{cs}$; $\vec{r}_{cs}=(x_{cs}-\Delta\tau_2\eta r_z)\hat{\mathbf{i}}+(z_{cs}+\Delta\tau_2\eta r_x)\hat{\mathbf{k}}$
$\vec{V}_{\dot{\eta}} = -\Delta\tau_2\dot{\eta}\hat{\mathbf{j}}\times\vec{r}_{ap}$; $\vec{r}_{ap}=(r_x-\Delta\tau_2\eta r_z)\hat{\mathbf{i}}+(r_z+\Delta\tau_2\eta r_x)\hat{\mathbf{k}}$

With these expressions the local velocity at the control surface $\vec{V}_{Lcs}$ is computed:

$$\vec{V}_{Lcs}=\{V_\infty\cos(\alpha)+q(z_{cs}+\Delta\tau_2\eta r_x)-\Delta\tau_2\dot{\eta}(r_z+\Delta\tau_2\eta r_x)\}\hat{\mathbf{i}} + \{V_\infty\sin(\alpha)-q(x_{cs}-\Delta\tau_2\eta r_z)+\Delta\tau_2\dot{\eta}(r_x-\Delta\tau_2\eta r_z)\}\hat{\mathbf{k}}$$

and $\sin^2(\theta_{Lcs})$ is determined from equation (4.2-1):

$$\sin^2(\theta_{Lcs}) = \frac{\{\sin(\alpha+\delta-\Delta\tau_2\eta) + a_{cs} + b_{cs}\}^2}{\{c_{cs}\}^2+\{d_{cs}\}^2} \tag{4.2-3}$$

where, $a_{cs}= \frac{q-\Delta\tau_2\dot{\eta}}{V_\infty}[(\Delta\tau_2\eta r_x+z_{cs})\sin(\delta-\Delta\tau_2\eta)+(\Delta\tau_2\eta r_z-x_{cs})\cos(\delta-\Delta\tau_2\eta)]$

$b_{cs}= \frac{\Delta\tau_2\dot{\eta}}{V_\infty}[z_1\sin(\delta-\Delta\tau_2\eta)+x_1\cos(\delta-\Delta\tau_2\eta)]$

$c_{cs}= \cos(\alpha)+\frac{q}{V_\infty}(z_{cs}+\Delta\tau_2\eta r_x) - \frac{\Delta\tau_2\dot{\eta}}{V_\infty}(r_z+\Delta\tau_2\eta r_x)$

$d_{cs}= \sin(\alpha) - \frac{q}{V_\infty}(x_{cs}-\Delta\tau_2\eta r_z) + \frac{\Delta\tau_2\dot{\eta}}{V_\infty}(r_x-\Delta\tau_2\eta r_z)$

4.2.2 Integration

To compute the aerodynamic forces and moments acting on the lower forebody, the pressure distribution along the lower forebody is integrated over s_1. In equations (4.1-1)-(4.1-4) two integrals appear in the aerodynamic forebody forces which reduce to:

$$\int_0^{l_1} \sin^2(\theta_{Lf}(s_1))ds_1 \qquad\qquad \int_0^{l_1} s_1\sin^2(\theta_{Lf}(s_1))ds_1$$

These two integrals make apparent the need to express $\sin^2(\theta_{Lf}(s_1))$ as an explicit function of s_1. Equation (4.2-2) exhibits this function as a ratio of two quadratic polynomials in s_1, where each of the above integrals may be expressed as a sum of two separate integrals:

$$\int_0^{l_1} \sin^2(\theta_{Lf}(s_1))ds_1 = \int_0^{l_1} ds_1 + (a^2-c^2)\int_0^{l_1} \frac{1}{c^2+2abs_1+b^2s_1^2}\,ds_1$$

$$\int_0^{l_1} s_1\sin^2(\theta_{Lf}(s_1))ds_1 = \int_0^{l_1} s_1ds_1 + (a^2-c^2)\int_0^{l_1} \frac{s_1}{c^2+2abs_1+b^2s_1^2}\,ds_1$$

Essentially, to compute the aerodynamic forces and moments on the vehicle's lower forebody requires analytic solutions of four integrals, two of which have trivial solutions while the other two are determined from an integral table.

$$\mathbf{I}_1 = \int_0^{l_1} \frac{1}{c^2+2abs_1+b^2s_1^2}\,ds_1 \qquad\qquad \mathbf{I}_2 = \int_0^{l_1} \frac{s_1}{c^2+2abs_1+b^2s_1^2}\,ds_1$$

From [5] the closed form solutions of $\mathbf{I}_1$ and $\mathbf{I}_2$ depend on the values of a, b and c. For computational purposes there are two cases that may arise. One case occurs when either $a^2=c^2$ *or* the constant (c^2) in the integrand's denominator is much greater than the terms containing s_1. A conservative measure for the second condition uses the largest value for s_1, i.e., $c^2 \gg 2abl_1+(bl_1)^2$. The second case occurs when *both* of these conditions are violated:

Case I. $a^2=c^2$ *or* $c^2 \gg 2abl_1+(bl_1)^2$

Case II. $a^2 \neq c^2$ *and* c^2 is not $\gg 2abl_1+(bl_1)^2$

Case I.

$$\int_0^{l_1} \sin^2(\theta_{Lf}(s_1))ds_1 = \left(\frac{a}{c}\right)^2 l_1 , \qquad \int_0^{l_1} s_1\sin^2(\theta_{Lf}(s_1))ds_1 = \frac{1}{2}\left(\frac{al_1}{c}\right)^2$$

Case II. let $\sigma=c^2-a^2$.

<u>$\sigma>0$</u>

$$\mathbf{I}_1 = \frac{1}{|b|\sqrt{\sigma}}\left[\tan^{-1}\left(\frac{b^2l_1+ab}{|b|\sqrt{\sigma}}\right) - \tan^{-1}\left(\frac{ab}{|b|\sqrt{\sigma}}\right)\right]$$

<u>$\sigma<0$</u>

$$\mathbf{I}_1 = \frac{1}{2|b|\sqrt{-\sigma}} \ln\left(\frac{\dfrac{b^2l_1}{ab-|b|\sqrt{-\sigma}}+1}{\dfrac{b^2l_1}{ab+|b|\sqrt{-\sigma}}+1}\right)$$

$$\mathbf{I}_2 = \frac{1}{2b^2}\ln\left\{\left(\frac{bl_1}{c}\right)^2 + \frac{2ab}{c^2}l_1 + 1\right\} - \frac{a}{b}\mathbf{I}_1$$

$$\int_0^{l_1} \sin^2(\theta_{Lf}(s_1))ds_1 = l_1 - \sigma\mathbf{I}_1 , \qquad \int_0^{l_1} s_1\sin^2(\theta_{Lf}(s_1))ds_1 = \frac{l_1^2}{2} - \sigma\mathbf{I}_2$$

4.3. Modification of Control Surface Forces and Moments

Expressions for the forces and moments produced by the control surface were presented in [2]. But these expressions do not include: 1)Effect of elastic motion at the control surface location, 2)Contribution of the control surface force to the first elastic generalized force. Modifications to include these effects are derived here.

Considering the elastic motion at the control surface location will modify the local flow deflection angle θ_{Lcs}. In Section 4.2 elastic motion was considered and the resulting expression for $\sin^2(\theta_{Lcs})$ includes this effect. The effect of deflection rate $\dot{\delta}$ and chord length of the control surface on the pressure distribution are considered negligible, and the local flow deflection angle is considered constant over the control surface chord. This in turn yields a constant pressure distribution for the entire control surface. Equations (4.1-5)-(4.1-8) (taken from [2]) are modified by replacing $\sin^2(\alpha+\delta)$ with $\sin^2(\theta_{Lcs})$ of equation (4.2-3). Also δ, the control surface deflection, becomes $(\delta-\Delta\tau_2\eta)$ so that:

$$X_{cs}= -q_\infty C_{PN}\sin^2(\theta_{Lcs})\sin(\delta-\Delta\tau_2\eta)\frac{S_{cs}}{b} \tag{4.3-1}$$

$$Z_{cs}= -q_\infty C_{PN}\sin^2(\theta_{Lcs})\cos(\delta-\Delta\tau_2\eta)\frac{S_{cs}}{b} \tag{4.3-2}$$

The pitching moment produced by the control surface is:

$$M_{cs}= X_{cs}\,(z_{cs}+\Delta\tau_2\eta r_x) - Z_{cs}\,(x_{cs}-\Delta\tau_2\eta r_z) \tag{4.3-3}$$

The control surface contribution to the elastic generalized force was not considered in [2]. The virtual work produced by the control surface force can be differentiated with respect to the generalized displacement to yield the generalized force Q_{cs} contributed by the control surface:

$$Q_{cs}=\frac{\partial\delta W}{\partial\eta}=\frac{\partial(\vec{\mathbf{F}}_{cs}\cdot\Delta\vec{\mathbf{r}})}{\partial\eta}$$

where, $\vec{\mathbf{F}}_{cs}= X_{cs}\hat{\mathbf{i}} + Z_{cs}\hat{\mathbf{k}}$

$\Delta\vec{\mathbf{r}} = -\Delta\tau_2\eta r_z\hat{\mathbf{i}} + \Delta\tau_2\eta r_x\hat{\mathbf{k}}$

so that:

$$Q_{cs}= -\frac{\partial X_{cs}}{\partial\eta}\Delta\tau_2\eta r_z - \Delta\tau_2 r_z X_{cs} + \frac{\partial Z_{cs}}{\partial\eta}\Delta\tau_2\eta r_x + \Delta\tau_2 r_z Z_{cs}$$

Differentiating X_{cs} and Z_{cs} with respect to η yields:

$$\frac{\partial X_{cs}}{\partial \eta} = q_\infty C_{pn}\frac{S_{cs}}{b}\left[\Delta\tau_2\cos(\delta-\Delta\tau_2\eta)\sin^2(\theta_{Lcs}) - \sin(\delta-\Delta\tau_2\eta)\frac{\partial \sin^2(\theta_{Lcs})}{\partial \eta}\right]$$

$$\frac{\partial Z_{cs}}{\partial \eta} = -\, q_\infty C_{pn}\frac{S_{cs}}{b}\left[\Delta\tau_2\sin(\delta-\Delta\tau_2\eta)\sin^2(\theta_{Lcs}) - \cos(\delta-\Delta\tau_2\eta)\frac{\partial \sin^2(\theta_{Lcs})}{\partial \eta}\right]$$

Differentiating equation (4.2-3) with respect to η and substituting back into the above equations results in a lengthy expression for Q_{cs}:

$$Q_{cs} = \Delta\tau_2\left(\begin{array}{c} (\Delta\tau_2\eta r_x - r_z)X_{cs} + (\Delta\tau_2\eta r_z + r_x)Z_{cs} \\ + 2\Delta\tau_2\eta(r_x Z_{cs} - r_z X_{cs})\cdot\left[\dfrac{\{q(x_1+\Delta\tau_2\eta r_z)-\Delta\tau_2\eta\Delta\tau_2\dot{\eta} r_z\}\sin(\delta-\Delta\tau_2\eta)}{V_\infty(\sin(\alpha+\delta-\Delta\tau_2\eta)+a_{cs}+b_{cs})}\right. \\ + \dfrac{\{\Delta\tau_2\eta\Delta\tau_2\dot{\eta} r_x - q(z_1+\Delta\tau_2\eta r_x)\}\cos(\delta-\Delta\tau_2\eta)}{V_\infty(\sin(\alpha+\delta-\Delta\tau_2\eta)+a_{cs}+b_{cs})} \\ - \dfrac{V_\infty\cos(\alpha+\delta-\Delta\tau_2\eta)}{V_\infty(\sin(\alpha+\delta-\Delta\tau_2\eta)+a_{cs}+b_{cs})} \\ \left. - \dfrac{(q-\Delta\tau_2\dot{\eta})(r_x c_{cs}+r_z d_{cs})}{V_\infty(c_{cs}^2+d_{cs}^2)}\right] \end{array}\right) \qquad (4.3\text{-}4)$$

5. COMPUTING THE AEROPROPULSIVE FORCES AND MOMENTS

5.1. Computation of Useful Variables

The aeropropulsive force equations (4.1-1) to (4.1-16) require various quantities which depend on the vehicle states, controls and physical parameters. Many of these quantities are computed in the block COMPVAR. For example, the freestream temperature and pressure are functions of altitude and are determined according to the ARDC Standard Atmosphere. Also determined are freestream velocity, Mach number and dynamic pressure, as well as angle of attack. In addition, COMPVAR calculates the instantaneous elastic deflection of the vehicle and its elastic deflection rate from the first generalized elastic co-ordinate and its mode shape.

5.2. Aerodynamic Body Forces and Moments

As noted in Chapter 4, the aerodynamic forces acting on a hypersonic vehicle are expressed in terms of the local flow deflection angle θ along the forebody and at the control surface. The block THETAL determines $\sin^2(\theta_{Lf})$ at the forebody by computing the coefficients a, b and c in equation (4.2-2). It then computes the closed-form solutions for the integral terms, as discussed in Subsection 4.2.2. The block AEROB utilizes equations (4.1-1) to (4.1-4) to compute the aerodynamic forces and moments produced by the pressure distribution on the vehicle's lower forebody and upper surface.

5.3 Aerodynamic Control Surface Forces and Moments

The block AEROC calculates $\sin^2(\theta_{Lcs})$, using the parameters a_{cs}, b_{cs}, c_{cs}, and d_{cs} in equation (4.2-3), and computes the control surface forces and moments in accordance with equations (4.3-1) to (4.3-4).

5.4 Engine Thermodynamics and Thrust Forces and Moments

The block ENGTHERM is made up of four sub-blocks: STAGE1, STAGE2, STAGE3, and STAGEE. Given freestream conditions, engine control settings (A_D and T_o), and the local flow deflection angle at the engine inlet ($\theta_{Lf}(s_1{=}0)$), the functions in ENGTHERM are used to determine inlet conditions at the engine diffuser, combustor, nozzle, and at the engine exit. The equations (5.4-1) to (5.1-12) below are used [2]. A numerical search using Newton's method is used to solve equations (5.4-4) and (5.4-10) iteratively for the Mach number. The block THRUST then computes the engine-module thrust force and moment using equations (4.1-9) to (4.1-12).

External Diffuser:

$$M_1 = \frac{M_\infty \cos(\theta_{Lf})}{\sqrt{1 + \frac{1}{2}(\gamma-1)M_\infty^2 \sin^2(\theta_{Lf})}} \tag{5.4-1}$$

$$P_1 = P_\infty\left[1 + \frac{1}{2}\gamma C_p M_\infty^2 \sin^2(\theta_{Lf})\right] \tag{5.4-2}$$

$$T_1 = P_\infty\left[1 + \frac{1}{2}(\gamma-1)M_\infty^2 \sin^2(\theta_{Lf})\right] \tag{5.4-3}$$

Internal Diffuser:

$$\frac{\left(1+\frac{1}{2}(\gamma-1)M_2^2\right)^{\frac{\gamma+1}{\gamma-1}}}{M_2^2} = \bar{A}_D^2 \frac{\left(1+\frac{1}{2}(\gamma-1)M_1^2\right)^{\frac{\gamma+1}{\gamma-1}}}{M_1^2} \tag{5.4-4}$$

$$P_2 = P_1\left[\frac{1+\frac{1}{2}(\gamma-1)M_1^2}{1+\frac{1}{2}(\gamma-1)M_2^2}\right]^{\frac{\gamma}{\gamma-1}} \tag{5.4-5}$$

$$T_2 = T_1\left[\frac{1+\frac{1}{2}(\gamma-1)M_1^2}{1+\frac{1}{2}(\gamma-1)M_2^2}\right] \tag{5.4-6}$$

Combustor:

$$\frac{M_3^2\left(1+\frac{1}{2}(\gamma-1)M_3^2\right)}{\left(1+\gamma M_3^2\right)^2} = \frac{M_2^2\left(1+\frac{1}{2}(\gamma-1)M_2^2\right)}{\left(1+\gamma M_2^2\right)^2} + \frac{M_2^2}{\left(1+\gamma M_2^2\right)^2}\frac{T_o}{T_2} \tag{5.4-7}$$

$$P_3 = P_2\left[\frac{1+\gamma M_2^2}{1+\gamma M_3^2}\right] \tag{5.4-8}$$

$$T_3 = T_2\left[\frac{\left(1+\gamma M_2^2\right)}{\left(1+\gamma M_3^2\right)}\frac{M_3}{M_2}\right]^2 \tag{5.4-9}$$

Internal Nozzle:

$$\frac{\left(1+\frac{1}{2}(\gamma-1)M_e^2\right)^{\frac{\gamma+1}{\gamma-1}}}{M_e^2} = \bar{A}_N^2 \frac{\left(1+\frac{1}{2}(\gamma-1)M_3^2\right)^{\frac{\gamma+1}{\gamma-1}}}{M_3^2} \tag{5.4-10}$$

$$P_e = P_3\left[\frac{1+\frac{1}{2}(\gamma-1)M_3^2}{1+\frac{1}{2}(\gamma-1)M_2^2}\right]^{\frac{\gamma}{\gamma-1}} \tag{5.4-11}$$

$$T_e = T_3\left[\frac{1+\frac{1}{2}(\gamma-1)M_3^2}{1+\frac{1}{2}(\gamma-1)M_e^2}\right] \tag{5.4-12}$$

5.5. External Nozzle Forces and Moments

The engine exhaust plume is bounded above by the vehicle lower aftbody, and below by a free shear layer. This variable structure is termed the external nozzle. The pressure distribution along the lower aftbody produces forces and moments on the vehicle aftbody. The block NOZZLE computes these quantities using equations (4.1-13) to (4.1-16), taken from [2].

6. VEHICLE CONFIGURATION AND FLIGHT CONDITION

6.1. Model Parameters

This section lists several pertinent constants associated with the spherical rotating earth and ARDC standard atmosphere models. Many atmospheric parameters are dependent on altitude and are contained in the function *strdatm*. Also given are vehicle dimensions, elastic and mass properties, and engine parameters for a particular vehicle configuration. All of the values below are defined in the script file *inpar*.

Spherical Rotating Earth:

Symbol Text	Symbol Code	Value	Description	Units
R_e	re	2.09256E+6	Earth radius	[ft]
μ_e	mu	1.40764E+16	Earth gravitational parameter	[ft^3/s^2]
ω_e	ome	7.297205E-5	Earth rotation rate about its axis	[rad/s]

ARDC Standard Atmosphere (1959):

Symbol Text	Symbol Code	Value	Description	Units
γ	gam	1.43	Ratio of specific heats	[-]
R_A	gasc	1.716545E+3	Atmospheric gas constant	[$ft^2/°R \cdot s^2$]
C_{pn}	cpn	2.0	Pressure coefficient	[-]

(A complete model for a spherical rotating earth and ARDC Standard Atmosphere is given in [1].)

Vehicle Dimensions and Mass Properties (Example):

Symbol Text	Symbol Code	Value	Description	Units
τ_1	tau1	14.0	Vehicle nose angle	[deg]
τ_2	tau2	20.0	Vehicle tail angle	[deg]
$\Delta\tau_1$	deltau1	1.0	Forebody elastic mode shape	[deg]
$\Delta\tau_2$	deltau2	1.0	Aftbody elastic mode shape	[deg]
L	l	150	Total vehicle length	[ft]
h	h	22.2	Total vehicle height	[ft]
L_1	l1	89.03	Vehicle forebody length	[ft]
L_2	l2	60.97	Vehicle aftbody length	[ft]
x_{cg}	xcg	90.0	c.g. position aft of vehicle nose	[ft]
z_{cg}	zcg	11.25	c.g. position below vehicle nose	[ft]
x_{cs}	xcs	-52.5	Control surface position w.r.t. center of gravity in Body x-axis	[ft]
z_{cs}	zcs	-11.25	Control surface position w.r.t. center of gravity in Body z-axis	[ft]
S_{cs}	scs	22.5	Control surface area (per vehicle width)	[ft^2/ft]
m	m	500.0	Vehicle mass (per vehicle width)	[slug/ft]
I_{yy}	iyy	1.0E+6	Vehicle Body y-axis moment of inertia (per vehicle width)	[$ft^2 \cdot$slug/ft]
m_{E1}	me1	40.0	Generalized mass associated with the 1st generalized elastic DoF (per vehicle width)	[slug/ft]
ω_1	om1	18.0	Undamped natural frequency of 1st generalized elastic DoF	[rad/s^2]
ζ_1	zet1	0.01	Damping ratio of 1st generalized elastic DoF	[-]

Engine Parameters (Example):

Symbol Text	Symbol Code	Value	Description	Units
$\bar{A}_N$	an	6.35	Nozzle area ratio	[-]
A_e	ae	8.88	Nozzle exit area (per vehicle width)	[ft^2/ft]

(A complete description of the geometry for this particular vehicle is given in [2].)

6.2. Initial Conditions--Example

The formulation of ASUHS1 is based upon a wings level (ϕ=0°) flight condition. For the example presented here, the initial vehicle flight-path angle is zero (γ=0°, $\theta=\alpha$), with the velocity-vector directed eastward (ψ=90°). This information specifies the vehicle's initial orientation.

Furthermore, suppose the vehicle position is specified with an altitude of 85,000 ft., over the intersection of the equator (latitude λ=0°) and prime meridian (longitude τ=0°). (See Figure 2.) This information specifies the three vehicle states h, λ and τ. The vehicle's initial Mach number for this flight condition is 8.0. The ambient conditions according to the ARDC Standard Atmosphere at are:

$P_\infty = 45.82$ lb/ft^2
$T_\infty = 394.3$ °R
$M_\infty = 8.0$
$V_\infty = 7870.5$ ft/s

At this point a trimmed flight condition (operating point) must be determined. Appendix C presents a trimming method which may be used to determine this trimmed flight condition. The method assumes a particular flight condition *structure*. Here the flight condition structure is defined as the statement of those variables that are specified ("fixed") and those that are allowed to vary ("free"). The flight condition structure used for this method specifies: vehicle position (h, τ, λ), Mach number (M_∞), roll and yaw attitude (ϕ, ψ), flight path angle (γ), and engine temperature control (T_0) and nozzle area ratio ($\overline{A}_N$). The vehicle rotation and elastic rates (q, $\dot{\eta}$), and accelerations ($\dot{u}$, $\dot{w}$, $\dot{q}$, $\ddot{\eta}$) are specified to be zero. The chosen free variables are: angle of attack (α), pitch control surface deflection (δ), diffuser area control ($\overline{A}_D$), and elastic deflection (η). Under this scenario the resulting trimmed vehicle states and controls are:

$h_0 = 85{,}000$ [ft]
$u_0 = 7{,}806$ [ft/s]
$w_0 = -1002$ [ft/s] (α_0=-7.317°)
$q_0 = 0.0$ [rad/s]
$\eta_0 = 1.243$ [-]
$\dot{\eta}_0 = 0.0$ [1/s]
$\lambda_0 = 0.0$ [rad]
$\tau_0 = 0.0$ [rad]

$\beta 1_0 = 0.04512$ [-]
$\beta 2_0 = -0.04512$ [-]
$\beta 3_0 = 0.7057$ [-]
$\beta 4_0 = 0.7057$ [-]

$\delta_0 = 25.21$ [deg]
$\overline{A}_{D_0} = 0.5004$ [-]
$T_0 = 2000$ [R°]

Due to round-off of the free variables, the accelerations do not exactly result in zero values. The following accelerations result from the given initial conditions and controls:

$\dot{u}_0 = 4.475E-4$ [ft/s^2]
$\dot{w}_0 = 7.514E-4$ [ft/s^2]
$\dot{q}_0 = 1.336E-5$ [rad/s^2]
$\ddot{\eta}_0 = -1.402E-2$ [1/s^2]

The script file *initstate.* is currently set up to perform an example simulation using the above initial conditions and a doublet input for δ (shown below in Figure 4). The resulting trajectory is presented in Figure 5.

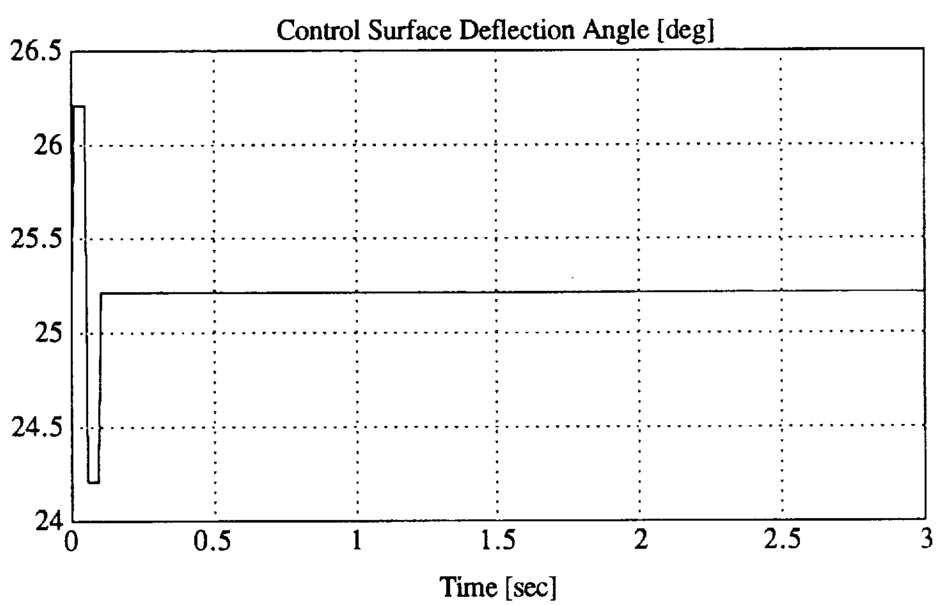

Figure 4. Doublet input for Control Surface Deflection Angle δ (±1 degree from trim).

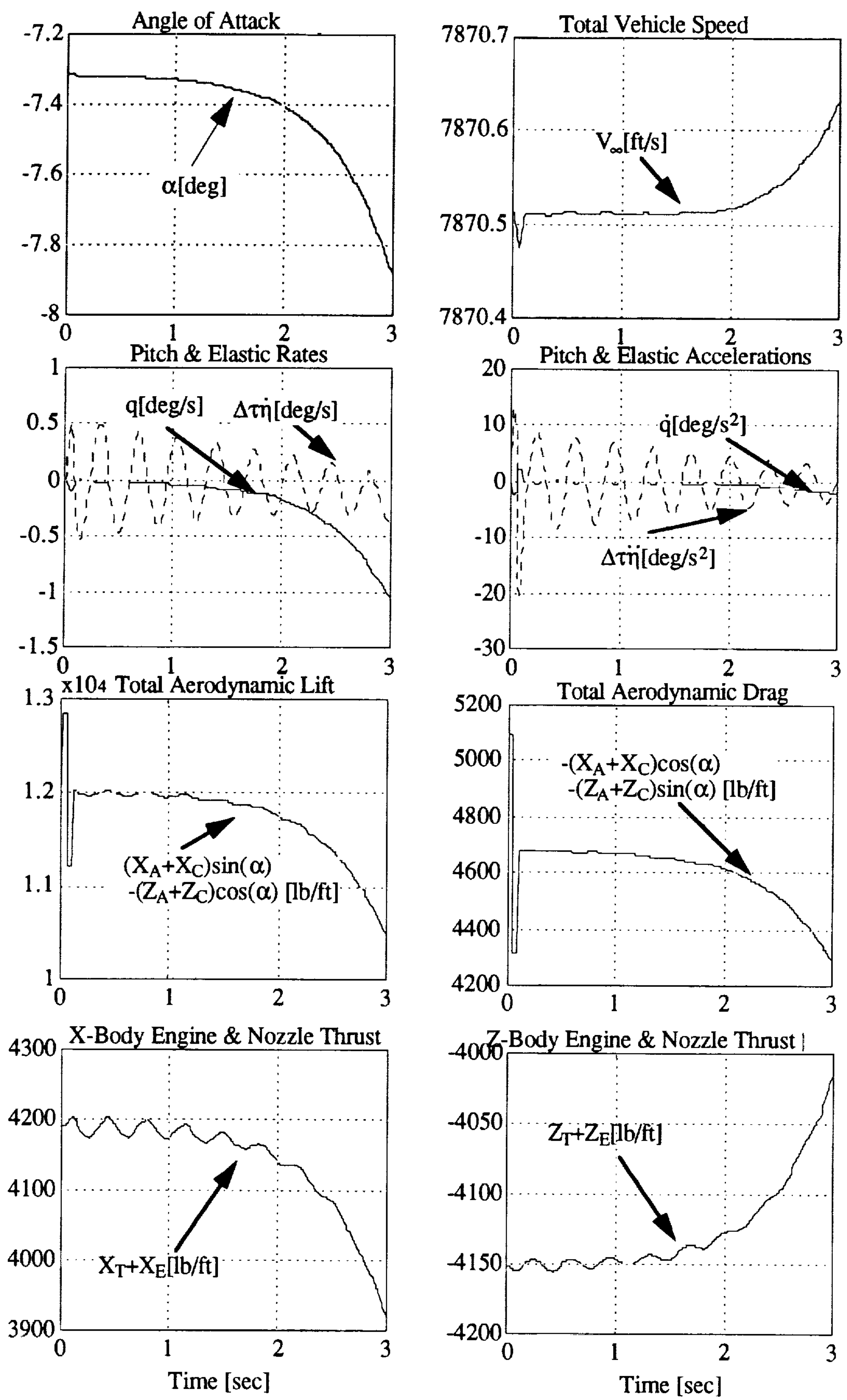

Figure 5. Simulation time responses to a doublet input in δ.

7. DISCUSSION AND POTENTIAL PROBLEMS

The expressions derived in this work rely entirely upon the aeroelastic model presented in Reference [2]. This elastic model was intended to provide the analyst with first-order effects of the expected interactions between the engine and aircraft and also between the rigid body and elastic degrees of freedom. Thus, if the elastic model is changed greatly, the present model may or may not capture all first-order effects. Certainly, if the elastic model is altered, the expressions derived herein may no longer be valid.

A major contribution of this work has been to integrate the pressure distribution along the vehicle's lower forebody. It was shown that the integrals of the pressure distribution produced by this hypersonic vehicle model have closed-form solutions. The structure of these solutions depends upon the vehicle's dynamic state (Case I or Case II in Subsection 4.2.2). In computing these solutions, the magnitudes of some parameters may be small but non-zero. A measure for zero (to be adjusted at the user's discretion) must be incorporated. From experience, a term smaller than 1E-6 (the current zero measure in the function *intheta.*) will be considered as zero. The current zero measure provides a smooth numerical transition from Case I to Case II.

The derivation of the engine thermodynamic conditions [2] relies on the *a priori* assumption of non-choked flow (i.e., Mach numbers always greater than unity throughout the engine-module). This certainly is the case for the vehicle configuration presented here, but in general non-choked flow is not guaranteed for arbitrary engine control settings. This issue may become critical when attempting to trim about a new flight condition. In addition, if the control settings are allowed to vary (e.g., in a closed-loop control simulation), the analyst must ensure that engine controls do not choke the fluid flow and thus violate the engine modeling assumptions. If the engine controls do choke the fluid flow, a "choked flow" warning is printed to MATLAB®'s command window. If this occurs it is suggested that the simulation be cancelled by typing "control-C," if the simulation does not cease by itself.

One peculiarity encountered while running ASUHS1 is the initial warning that is flagged to MATLAB®'s command window. This warning is a result of using the previous integration time step values of M_2 and M_e as the initial guesses in the Newton iteration search. This method reduces computation time but at the same time introduces this annoying warning. This warning, due to this implementation of the Newton iteration search, may be ignored. Due to this implementation in *stage2* and *stagee*, the simulation start time must be set at zero.

REFERENCES

1. Bilimoria, K.D., and Schmidt, D.K., "An Integrated Development of the Equations of Motion for Elastic Hypersonic Flight Vehicles", Report ARC92-3, Aerospace Research Center, Arizona State University, July 1992.

2. Chavez, F. and Schmidt, D.K., "An Integrated Analytical Aeropropulsive/Aeroelastic Model for the Dynamic Analysis of Hypersonic Vehicles", Report ARC92-2, Aerospace Research Center, Arizona State University, June 1992.

3. The MathWorks, Inc., *SIMULINK User's Guide*, Natick, Massachusetts, March 1992.

4. Schmidt, D.K., "Dynamics and Control of Hypersonic Aeropropulsive/Aeroelastic Vehicles", AIAA Paper No.92-4326, Proceedings of the AIAA Conference on Guidance, Navigation and Control, August 1992, pp161-171.

5. Beyer, W.H., *CRC Standard Mathematical Tables*, Boca Raton, Florida, 1981, pp297-298.

6. The MathWorks, Inc., *MATLAB User's Guide*, Natick, Massachusetts, March 1988.

APPENDIX

A. DESCRIPTION OF ASUHS1 MATLAB® FUNCTIONS

Function Name: *strdatm*

Description: This function implements the 1959 ARDC Standard Atmosphere to compute the freestream temperature and pressure given the vehicle's altitude. This model was obtained from Reference [1].

Input Data:

Symbol Text	Symbol Code	Description	Units
h	h	Geometric altitude as measured from the earth's surface	[ft]
R_e	re	Earth's radius	[ft]
g_o	go	Gravitational acceleration at the earth's surface	[ft/s^2]
R_A	gasc	Atmospheric gas constant	[ft^2/°R-s^2]

Output Data:

Symbol Text	Symbol Code	Description	Units
T_∞	tinf	Atmospheric freestream temperature	[°R]
P_∞	pinf	Atmospheric freestream pressure	[lb/ft^2]

Procedure: Geopotential altitude h_g is first computed from geometric altitude:

$$h_g = h\left(\frac{R_e}{R_e+h}\right)$$

With a geopotential altitude the atmospheric layer is determined and the corresponding values for T_{ref}, C, h_{ref} and P_{ref} are determined according the ARDC Standard Atmosphere. These four parameters are used to compute the freestream temperature and pressure:

$$T_\infty = T_{ref} + C(h_g - h_{ref})$$

$$\text{for } C=0, \quad P_\infty = P_{ref}\left\{\exp\left[-\frac{g_o}{R_A \cdot T_{ref}}(h_g - h_{ref})\right]\right\}$$

$$\text{for } C\neq 0, \quad P_\infty = P_{ref}\left[1+\frac{C}{T_{ref}}(h_g - h_{ref})\right]^{\left(\frac{-g_o}{CR_A}\right)}$$

Function Name: *speedalpha*

Description: This function computes the total vehicle velocity and angle of attack given the vehicle velocity components.

Input Data:

Symbol Text	Symbol Code	Description	Units
u	u	Vehicle forward speed	[ft/s]
w	w	Vehicle sink rate	[ft/s]

Output Data:

Symbol Text	Symbol Code	Description	Units
V_∞	vinf	Total vehicle speed (in pitch plane)	[ft/s]
α	al	Vehicle angle of attack	[rad]

Procedure: The total vehicle speed is computed as:

$$V_\infty=\sqrt{u^2+w^2}$$

The vehicle angle of attack is determined by:

$$\alpha=\tan^{-1}\left(\frac{w}{u}\right)$$

Function Name: *machq*

Description: This function computes the freestream Mach number and freestream dynamic pressure.

Input Data:

Symbol Text	Symbol Code	Description	Units
T_∞	tinf	Atmospheric freestream temperature	[°R]
P_∞	pinf	Atmospheric freestream pressure	[lb/ft^2]
V_∞	vinf	Total vehicle speed (in pitch plane)	[ft/s]
R_A	gasc	Atmospheric gas constant	[ft^2/°R-s^2]
γ	gam	Ratio of specific heats	[-]

Output Data:

Symbol Text	Symbol Code	Description	Units
M_∞	minf	Freestream Mach number	[-]
q_∞	qinf	Freestream dynamic pressure	[lb/ft^2]

Procedure: The speed of sound a_{sonic} is computed using the isentropic flow equation:

$$a_{sonic}=\sqrt{\gamma R_A T_\infty}$$

With the sonic speed the freestream Mach number is determined from:

$$M_\infty=\frac{V_\infty}{a_{sonic}}$$

Freestream dynamic pressure is then computed from:

$$q_\infty=\frac{\gamma}{2}P_\infty M_\infty^2$$

Function Name: *elasdef*

Description: This function computes the elastic deflections and deflection rates for the forebody and aftbody sections.

Input Data:

Symbol Text	Symbol Code	Description	Units
η	eta	First generalized elastic co-ordinate	[-]
$\dot{\eta}$	deta	First generalized elastic co-ordinate rate	[1/s]
$\Delta\tau_1$	deltau1	Forebody elastic mode shape	[rad]
$\Delta\tau_2$	deltau2	Aftbody elastic mode shape	[rad]
τ_1	tau1	Vehicle nose angle	[rad]
τ_2	tau2	Vehicle tail angle	[rad]

Output Data:

Symbol Text	Symbol Code	Description	Units
$\Delta\tau_1\eta$	deltaun1	Elastic deflection of forebody	[rad]
$\Delta\tau_2\eta$	deltaun2	Elastic deflection of aftbody	[rad]
τ_n	ttau1	Total (effective) vehicle nose angle	[rad]
τ_t	ttau2	Total (effective) vehicle tail angle	[rad]
$\Delta\tau_1\dot{\eta}$	dtau1	Elastic deflection rate of forebody	[rad/s]
$\Delta\tau_2\dot{\eta}$	dtau2	Elastic deflection rate of aftbody	[rad/s]

Procedure: The elastic deflections are computed by multiplying the mode shape components with the elastic co-ordinate, where the first component corresponds to the forebody section and second component corresponds to the aftbody section. The elastic deflections are then added to the nose and tail angles to yield the "effective" nose and tail angles of the vehicle:

$$\tau_n = \tau_1 + \Delta\tau_1\eta$$

$$\tau_t = \tau_2 + \Delta\tau_2\eta$$

The elastic deflection rates are computed by multiplying the corresponding mode shape components with the elastic co-ordinate rate.

Function Name: *coef*

Description: This function computes the "coefficients" involved in the expression for $\sin^2(\theta_L(s_1))$.

Input Data:

Symbol Text	Symbol Code	Description	Units
V_∞	vinf	Freestream velocity	[ft/s]
α	al	Vehicle angle of attack	[rad]
τ_n	ttau1	Total (effective) vehicle nose angle	[rad]
$\Delta\tau_1\dot{\eta}$	dtau1	Elastic deflection rate of forebody	[rad/s]
q	q	Vehicle pitch rate	[rad/s]
x_1	x1	Location of c.g. with respect to lower apex in body x-axis	[ft]
z_1	z1	Location of c.g. with respect to lower apex in body z-axis	[ft]

Output Data:

Symbol Text	Symbol Code	Description	Units
a	a	Represents the local flow normal to lower forebody at the engine inlet	[-]
b	b	Represents the effective pitch rate along the lower forebody surface	[1/ft]
c^2	csqrd	Represents the total velocity squared at the engine inlet	[1/ft^2]

Procedure: The three "coefficients" are computed according to equation (4.2-2) in Section 4.2.1.

Function Name: *intheta*

Description: This function computes the algebraic solutions of the integrals involving $\sin^2(\theta_L(s_1))$.

Input Data:

Symbol Text	Symbol Code	Description	Units
a	a	Represents the local flow normal to lower forebody at the engine inlet	[-]
b	b	Represents the effective pitch rate along the lower forebody surface	[1/ft]
c^2	csqrd	Represents the total velocity squared at the engine inlet	[1/ft²]
l_1	ell	Length along vehicle forebody	[ft]

Output Data:

Symbol Text	Symbol Code	Description	Units
-	sqth	$\sin^2(\theta_L(s_1{=}0))$ i.e., computed at the engine inlet	[-]
-	isqth	Integral of $\sin^2(\theta_L(s_1))ds_1$ from $s_1=0$ to $s_1=l_1$	[ft]
-	issqth	Integral of $s_1\sin^2(\theta_L(s_1))ds_1$ from $s_1=0$ to $s_1=l_1$	[ft²]

Procedure: This function begins by first setting an acceptable measure for zero, 0_{meas} (currently 0_{meas}=1E-6). The parameters σ and s are then computed as:

$$\sigma = c^2 - a^2$$

$$s = \sqrt{|\sigma|}$$

The logic to determine which algebraic solution applies is as follows:

If $s > 0_{meas}$ and $c^2 < \dfrac{|2abl_1+(bl_1)^2|}{0_{meas}}$

If $\sigma > 0$

Case II (σ>0)

Else

Case II (σ<0)

Else

Case I

Comparing s (rather than σ) to 0_{meas} permits use of the same 0_{meas} value to be used in the two inequalities in the first line of this logic sequence. The implementation of the analytic solutions in Section 4.2.2 avoids, whenever possible, division by $|b|$, σ or $\sqrt{\sigma}$ since these quantities are typically very small.

After computing the appropriate algebraic solutions the term $\sin^2(\theta_{Lf}(0))$ is computed as:

$$\sin^2(\theta_{Lf}(s_1=0)) = \frac{a^2}{c^2}$$

Function Name: *stage1*

Description: This function computes the thermodynamic conditions (M_1, P_1, T_1) at the engine inlet (diffuser inlet) referred to as Stage 1. Newtonian Impact Theory was applied in the derivation of these conditions [2].

Input Data:

Symbol Text	Symbol Code	Description	Units
q_∞	qinf	Freestream dynamic pressure	[lb/ft^2]
M_∞	minf	Freestream Mach number	[-]
P_∞	pinf	Atmospheric freestream pressure	[lb/ft^2]
T_∞	tinf	Atmospheric freestream temperature	[°R]
-	sqth	Evaluation of $\sin^2(\theta_L(s_1))$ at s_1=0 (engine inlet)	[-]
γ	gam	Ratio of specific heats	[-]
C_{pn}	cpn	Pressure coefficient	[-]

Output Data:

Symbol Text	Symbol Code	Description	Units
M_1	m1	Mach number at engine inlet (diffuser inlet)	[-]
P_1	p1	Pressure at engine inlet (diffuser inlet)	[lb/ft^2]
T_1	t1	Temperature at engine inlet (diffuser inlet)	[°R]

Procedure: The function first determines if the external diffuser has choked the fluid flow at the engine inlet. If choked flow does occur then a warning is flagged to the command window.

To facilitate computation a useful parameter Mg_∞ is computed:

$$Mg_\infty = 1 + \frac{\gamma-1}{2} M_\infty^2$$

Equations (5.4-1)-(5.4-3) in Section 5.4 are then implemented to compute M_1, P_1 and T_1, respectively.

Function Name: *stage2*

Description: This function computes the thermodynamic conditions (M_2, P_2, T_2) at the exit to the internal diffuser (combustor inlet) referred to as Stage 2. A supersonic isentropic flow assumption was made in the derivation of these conditions [2]. To solve for M_2 a Newton iteration search was implemented.

Input Data:

Symbol Text	Symbol Code	Description	Units
$\bar{A}_D$	ad	Diffuser area ratio control	[-]
M_1	m1	Mach number at engine inlet (diffuser inlet)	[-]
P_1	p1	Pressure at engine inlet (diffuser inlet)	[lb/ft^2]
T_1	t1	Temperature at engine inlet (diffuser inlet)	[°R]
γ	gam	Ratio of specific heats	[-]
γ_m	gamm	A useful constant for computation, $\gamma_m=\gamma/(\gamma-1)$	[-]
γ_s	sph	A useful constant for computation, $\gamma_s=(\gamma+1)/(\gamma-1)$	[-]
t	time	Simulation time	[sec]
M_{2i}	m2i	First initial guess for the Newton iteration search	[-]

Output Data:

Symbol Text	Symbol Code	Description	Units
$\bar{A}_D$	ad	Diffuser area ratio control	[-]
M_2	m2	Mach number at engine stage 2 (diffuser exit)	[-]
P_2	p2	Pressure at engine stage 2 (diffuser exit)	[lb/ft^2]
T_2	t2	Temperature at engine stage 2 (diffuser exit)	[°R]

Procedure: The function first computes the useful parameter Mg_1:

$$Mg_1 = 1+\frac{\gamma-1}{2}M_1^2$$

It then determines if the diffuser area ratio control will choke the fluid flow at the diffuser exit. If choked flow occurs a warning is flagged to MATLAB®'s command window. If the value for $\bar{A}_D$ maintains supersonic flow, a Newton iteration search is utilized to solve for M_2.

To reduce the number of cycles executed by the Newton iterative search the previous value for M_2 is used as the initial guess to start the search. To overcome the problem during the initial time calculations (there is no previous M_2 available), the following logic sequence is used:

$$
\begin{aligned}
&\text{If } \ t > 0.0 \\
&\quad M_2 = M_{2i} \\
&\text{Else} \\
&\quad M_2 = \frac{M_1}{2.5}
\end{aligned}
$$

The function then goes on to iteratively solve for M_2 using equation (5.4-4) in Section 5.4, via Newton's method.

With a value for M_2 the useful parameter Mg_2 is computed:

$$Mg_2 = 1 + \frac{\gamma - 1}{2} M_2^2$$

Equations (5.4-5) and (5.4-6) in Section 5.4 are then implemented which compute P_2 and T_2, respectively.

Function Name: *stage3*

Description: This function computes the thermodynamic conditions (M_3, P_3, T_3) at the exit to the combustor (nozzle inlet) referred to as Stage 3. A constant area heat addition assumption was made in the derivation of these conditions [2], where T_o is the total temperature change across the combustor. M_3 is determined via the quadratic formula.

Input Data:

Symbol Text	Symbol Code	Description	Units
T_o	to	Total temperature change control (across combustor)	[°R]
M_2	m2	Mach number at engine stage 2 (combustor inlet)	[-]
P_2	p2	Pressure at engine stage 2 (combustor inlet)	[lb/ft^2]
T_2	t2	Temperature at engine stage 2 (combustor inlet)	[°R]
γ	gam	Ratio of specific heats	[-]

Output Data:

Symbol Text	Symbol Code	Description	Units
T_o	to	Total temperature change control (across combustor)	[°R]
M_3	m3	Mach number at engine stage 3 (combustor exit)	[-]
P_3	p3	Pressure at engine stage 3 (combustor exit)	[lb/ft^2]
T_3	t3	Temperature at engine stage 3 (combustor exit)	[°R]

Procedure: The function begins by computing useful parameters, g_2 and Mg_2, which are used continuously throughout this routine:

$$g_2 = 1 + \gamma M_2^2, \qquad Mg_2 = 1 + \frac{\gamma - 1}{2} M_2^2$$

The total temperature change control is checked to verify that the flow within the combustor is non-choked. If choked flow occurs, a warning is flagged to MATLAB®'s command window. If the value for T_o maintains supersonic flow, then equation (5.4-7) in Section 5.4 is represented as a quadratic function in M_3^2 and the quadratic formula applied to solve for the appropriate root, i.e., the real positive root which is greater than unity.

With a value for M_3 the useful parameter g_3 is computed:

$$g_3 = 1 + \gamma M_3^2$$

The equations (5.4-8) & (5.4-9) in Section 5.4 are then implemented to compute P_3 and T_3, respectively.

Function Name: *stagee*

Description: This function computes the thermodynamic conditions (M_e, P_e, T_e) at the exit to the internal nozzle (engine exit) referred to as Stage e. A supersonic isentropic flow assumption was made in the derivation of these conditions [2]. To solve for M_e a Newton iteration search was implemented.

Input Data:

Symbol Text	Symbol Code	Description	Units
$\overline{A}_N$	an	Nozzle area ratio	[-]
M_3	m3	Mach number at engine stage 3 (nozzle inlet)	[-]
P_3	p3	Pressure at engine stage 3 (nozzle inlet)	[lb/ft^2]
T_3	t3	Temperature at engine stage 3 (nozzle inlet)	[°R]
γ	gam	Ratio of specific heats	[-]
γ_m	gamm	A useful constant for computation, $\gamma_m=\gamma/(\gamma-1)$	[-]
γ_s	sph	A useful constant for computation, $\gamma_s=(\gamma+1)/(\gamma-1)$	[-]
t	time	Simulation time	[sec]
M_{ei}	mei	First initial guess for the Newton iteration search	[-]

Output Data:

Symbol Text	Symbol Code	Description	Units
$\overline{A}_N$	an	Nozzle area ratio	[-]
M_e	mex	Mach number at engine exit (nozzle exit)	[-]
P_e	pex	Pressure at engine exit (nozzle exit)	[lb/ft^2]
T_e	tex	Temperature at engine exit (nozzle exit)	[°R]

Procedure: The function first computes the useful parameter Mg_3:

$$Mg_3 = 1+\frac{\gamma-1}{2}M_3^2$$

It then determines if the nozzle area ratio will choke the fluid flow at the nozzle exit. If choked flow occurs a warning is flagged to MATLAB®'s command window. If the value for $\overline{A}_N$ maintains supersonic flow a Newton iteration search is utilized to solve for M_e.

To reduce the number of cycles executed by the Newton iterative search the previous value for M_e is used as the initial guess to start the search. To overcome the problem during the initial time calculations (there is no previous M_e available) the integration time variable t is also used as input to this function. The logic sequence is as follows:

If $t > 0.0$
$M_e = M_{ei}$
Else
$M_e = 3M_3$

The function then goes on to iteratively solve for M_e using equation (5.4-10) in Section 5.4, via Newton's method.

With a value for M_e the useful parameter Mg_e is computed:

$$Mg_e = 1 + \frac{\gamma - 1}{2} M_e^2$$

The equations (5.4-11) & (5.4-12) in Section 5.4 are then implemented to compute P_e and T_e, respectively.

Function Name: *aerobody*

Description: This function computes the aerodynamic forces and moments acting on the vehicle forebody. Note these forces are determined per unit width of the vehicle.

Input Data:

Symbol Text	Symbol Code	Description	Units
q_∞	qinf	Freestream dynamic pressure	[lb/ft^2]
P_∞	pinf	Atmospheric freestream pressure	[lb/ft^2]
τ_n	ttau1	Total (effective) vehicle nose angle	[rad]
-	isqth	Integral of $\sin^2(\theta_L(s_1))ds_1$ from $s_1=0$ to $s_1=l_1$	[ft]
-	issqth	Integral of $s_1\sin^2(\theta_L(s_1))ds_1$ from $s_1=0$ to $s_1=l_1$	[ft^2]
l_1	ell	Length along vehicle forebody	[ft]
C_{pn}	cpn	Pressure coefficient	[-]
x_1	x1	Location of c.g. with respect to lower apex in body x-axis	[ft]
z_1	z1	Location of c.g. with respect to lower apex in body z-axis	[ft]
L	l	Vehicle length	[ft]
x_{cg}	xcg	Distance from vehicle nose to c.g.	[ft]
$\Delta\tau_1$	deltau1	Forebody elastic mode shape	[rad]

Output Data:

Symbol Text	Symbol Code	Description	Units
X_A	xa	Body x-axis aerodynamic force acting on vehicle forebody (per unit width of the vehicle)	[lb/ft]
Z_A	za	Body z-axis aerodynamic force acting on vehicle forebody (per unit width of the vehicle)	[lb/ft]
M_A	ma	Body y-axis aerodynamic moment acting on vehicle forebody (per unit width of the vehicle)	[ft·lb/ft]
Q_A	qa	First generalized force due to aerodynamic force acting on vehicle forebody (per unit width of the vehicle)	[ft·lb/ft]

Procedure: The function computes the magnitude of the total aerodynamic force acting on the vehicle forebody:

$$F_A = P_\infty l_1 + q_\infty C_{pn}\int_0^{l_1} \sin^2(\theta_{Lf}(s_1))ds_1$$

The function then goes on to compute X_A, Z_A, M_A and Q_A using equations (4.1-1) to (4.1-4) in Section 4.1.

Function Name: *aerocont*

Description: This function computes the aerodynamic forces and moments acting on the control surface. Note these forces are determined per unit width of the vehicle.

Input Data:

Symbol Text	Symbol Code	Description	Units
q_∞	qinf	Freestream dynamic pressure	[lb/ft^2]
V_∞	vinf	Freestream velocity	[ft/s]
q	q	Vehicle pitch rate	[rad/s]
$\Delta\tau_2\dot{\eta}$	dtau2	Elastic deflection rate of aftbody	[rad/s]
$\Delta\tau_2\eta$	deltaun2	Elastic deflection of aftbody	[rad]
α	al	Vehicle angle of attack	[rad]
δ	delta	Pitch control surface deflection	[rad]
l_1	el1	Length along vehicle forebody	[ft]
C_{pn}	cpn	Pressure coefficient	[-]
r_x	rx	Location of control surface w.r.t. lower apex in body x-axis	[ft]
r_z	rz	Location of control surface w.r.t. lower apex in body z-axis	[ft]
h	h	Vehicle height	[ft]
x_{cg}	xcg	Distance from vehicle nose to c.g. in body x-axis	[ft]
z_{cg}	zcg	Distance from vehicle nose to c.g. in body z-axis	[ft]
x_{cs}	xcs	Location of control surface with respect to c.g. in body x-axis	[ft]
z_{cs}	zcs	Location of control surface with respect to c.g. in body x-axis	[ft]
s_{cs}	scs	Control surface area (per unit width of the vehicle)	[ft^2/ft]
$\Delta\tau_2$	deltau2	Aftbody elastic mode shape	[rad]

Output Data:

Symbol Text	Symbol Code	Description	Units
X_C	xc	Body x-axis aerodynamic force acting on control surface (per unit width of the vehicle)	[lb/ft]
Z_C	zc	Body z-axis aerodynamic force acting on control surface (per unit width of the vehicle)	[lb/ft]
M_C	mc	Body y-axis aerodynamic moment acting on control surface (per unit width of the vehicle)	[ft·lb/ft]
Q_C	qc	First generalized force due to aerodynamic force acting on control surface (per unit width of the vehicle)	[ft·lb/ft]

Procedure: The function calculates the individual terms of equation (4.2-2) in Section 4.2 in order to compute $\sin^2(\theta_{Lcs})$. Equations (4.3-1) to (4.3-4) are then implemented to determine X_C, Z_C, M_C and Q_C, respectively.

Function Name: *engthrust*

Description: This function computes the forces and moments generated by the engine module thrust. It is assumed that the resultant thrust force is in the body x-axis direction. Note these forces are determined per unit width of the vehicle.

Input Data:

Symbol Text	Symbol Code	Description	Units
$\overline{A}_D$	ad	Diffuser area ratio control	[-]
q_∞	qinf	Freestream dynamic pressure	[lb/ft^2]
M_1	m1	Mach number at engine inlet	[-]
P_1	p1	Pressure at engine inlet	[lb/ft^2]
M_e	mex	Mach number at engine exit	[-]
P_e	pex	Pressure at engine exit	[lb/ft^2]
A_e	ae	Nozzle exit area	[ft^2/ft]
γ	gam	Ratio of specific heats	[-]
$\overline{A}_N$	an	Nozzle area ratio	[-]
z_1	z1	Location of c.g. with respect to lower apex in body z-axis	[ft]

Output Data:

Symbol Text	Symbol Code	Description	Units
X_T	xt	Body x-axis engine thrust force (per unit width of the vehicle)	[lb/ft]
Z_T	zt	Body z-axis engine thrust force (per unit width of the vehicle)	[lb/ft]
M_T	mt	Body y-axis engine thrust pitching moment (per unit width of the vehicle)	[ft·lb/ft]
Q_T	qt	First generalized force due to engine thrust (per unit width of the vehicle)	[ft·lb/ft]

Procedure: The function computes the engine thrust forces and moments by direct implementation of equations (4.1-9) to (4.1-12) in Section 4.1.

Function Name: *extnozzle*

Description: This function computes the aeropropulsive forces and moments on the lower aftbody of the vehicle considered to be the upper surface of the external nozzle. A pressure distribution which varies as $1/(1+s_2)$ has been assumed [2], where s_2 is the position along the lower aftbody from the lower apex. Note these forces are determined per unit width of the vehicle.

Input Data:

Symbol Text	Symbol Code	Description	Units
τ_t	ttau2	Total (effective) vehicle tail angle	[rad]
P_∞	pinf	Atmospheric freestream pressure	[lb/ft^2]
x_1	x1	Location of c.g. with respect to lower apex in body x-axis	[ft]
z_1	z1	Location of c.g. with respect to lower apex in body z-axis	[ft]
P_e	pex	Pressure at engine exit (nozzle exit)	[lb/ft^2]
l_2	el2	Length along vehicle aftbody	[ft]
τ_2	tau2	Vehicle tail angle	[rad]
$\Delta\tau_2$	deltau2	Aftbody elastic mode shape	[rad]
$\Delta\tau_2\eta$	deltaun2	Elastic deflection of aftbody	[rad]

Output Data:

Symbol Text	Symbol Code	Description	Units
X_E	xe	Body x-axis aeropropulsive force acting on the external nozzle (per unit width of the vehicle)	[lb/ft]
Z_E	ze	Body z-axis aeropropulsive force acting on the external nozzle (per unit width of the vehicle)	[lb/ft]
M_E	me	Body y-axis aeropropulsive pitching moment acting on the external nozzle (per unit width of the vehicle)	[ft·lb/ft]
Q_E	qe	First generalized aeropropulsive force acting on the external nozzle (per unit width of the vehicle)	[ft·lb/ft]

Procedure: The function first computes useful computational parameters:

$$\bar{P} = \frac{P_e}{P_\infty}$$

$$r_2 = (h - z_{cg})\sin(\tau_2 + \Delta\tau_2\eta) - (L_1 - x_{cg})\cos(\tau_2 + \Delta\tau_2\eta)$$

$$\bar{l} = \frac{\overline{P}\ \ln(\overline{P})}{\overline{P}-1}$$

$$\overline{m} = \frac{1-\ln(\overline{P})}{\overline{P}-1}$$

With these parameters equations (4.1-13) to (4.1-16), from [2], are implemented to compute X_E, Z_E, M_E, and Q_E, respectively.

Function Name: *eomsphr*

Description: This function computes the time derivatives of the twelve dynamic variables for time integration.

Input Data:

Symbol Text	Symbol Code	Description	Units
X	xtot	Body x-axis aeropropulsive force (per unit width of the vehicle)	[lb/ft]
Z	ztot	Body z-axis aeropropulsive force (per unit width of the vehicle)	[lb/ft]
M	mtot	Body y-axis aeropropulsive moment (per unit width of the vehicle)	[ft·lb/ft]
Q_{E1}	qtot	Generalized aeropropulsive force associated with the first generalized elastic degree of freedom (per unit width of the vehicle)	[ft·lb/ft]
h	h	Geometric altitude measured from the earth's surface	[ft]
u	u	Vehicle forward speed	[ft/s]
v	v	Vehicle lateral speed	[ft/s]
w	w	Vehicle sink rate	[ft/s]
q	q	Vehicle rigid body pitch rate	[rad/s]
η	eta	First generalized elastic co-ordinate	[-]
$\dot{\eta}$	etadot	First generalized elastic co-ordinate rate	[-]
λ	lam	Latitude of vehicle position	[rad]
τ	tau	Longitude of vehicle position	[rad]
β_1	b1	First quaternion component	[-]
β_2	b2	Second quaternion component	[-]
β_3	b3	Third quaternion component	[-]
β_4	b4	Fourth quaternion component	[-]
ω_e	ome	Earth rotation rate	[rad/s]
R_e	re	Earth radius	[ft]
g_o	go	Gravitational acceleration at Earth's surface	[ft/s^2]
m	m	Vehicle mass (per unit width of the vehicle)	[slug/ft]
I_{yy}	iyy	Vehicle moment of inertia about Body y axis (per unit width of the vehicle)	[ft^2·slug/ft]
ω_1	om1	Undamped natural frequency of 1st generalized elastic degree of freedom	[rad/s]
ζ_1	zet1	Damping ratio of 1st generalized elastic degree of freedom	[-]
m_{E1}	me1	Generalized mass associated with the first generalized elastic degree of freedom (per unit width of the vehicle)	[ft^2·slug/ft]

Output Data:

Symbol Text	Symbol Code	Description	Units
$\dot{h}$	dh	Time rate of change of geometric altitude	[ft/s]
$\dot{u}$	du	Body x-axis acceleration	[ft/s^2]
$\dot{w}$	dw	Body z-axis acceleration	[ft/s^2]
$\dot{q}$	dq	Body y-axis angular acceleration	[rad/s^2]
$\dot{\eta}$	deta	First generalized elastic co-ordinate rate	[1/s]
$\ddot{\eta}$	ddeta	First generalized elastic co-ordinate acceleration	[1/s^2]
$\dot{\lambda}$	dlam	Time rate of change of vehicle latitude	[rad/s]
$\dot{\tau}$	dtau	Time rate of change of vehicle longitude	[rad/s]
$\dot{\beta}_1$	db1	Time rate of change of first quaternion component	[1/s]
$\dot{\beta}_2$	db2	Time rate of change of second quaternion component	[1/s]
$\dot{\beta}_3$	db3	Time rate of change of third quaternion component	[1/s]
$\dot{\beta}_4$	db4	Time rate of change of fourth quaternion component	[1/s]

Procedure: The function begins by first computing the Vehicle-carrying frame to Body frame co-ordinate transformation matrix. Vehicle distance from the Earth's center is then computed. With these quantities equations (3.4) to (3.16) in Chapter 3 are implemented. The last four differential equations are implemented after the quantities p_v, q_v and r_v are computed. Note: The twelve output quantities above are integrated and then fed into the "states" block (Figure A1) in the order that they appear above. Thus, for example, if the time history of w after a simulation is desired, it is found in the third column of the workspace variable "states." In addition, the quantities $\dot{u}$, $\dot{w}$, $\dot{q}$ and $\ddot{\eta}$ are fed into the block "acc" in that order.

B. PROCEDURAL OUTLINE OF ASUHS1

The software development of ASUHS1 is centered around the "user-friendly" environment of SIMULINK®'s block diagram structure. It was designed to execute on any Apple Macintosh computer with SIMULINK® (or SIMULAB®) installed. . The suggested RAM size (1 Mbyte) allocated for MATLAB/S execution should be expanded to 2 Mbytes. This will allow the simulation to perform over 3,500 integration time steps under the current ASUHS1 set-up. The complete ASUHS1 package resides on a $3^1/_2$" floppy disk which may be acquired from Professor David K. Schmidt at the Aerospace Research Center.

The simulation package consists of seventeen MATLAB® functions and two script files. One of the functions is ASUHS1 itself, fifteen are functions called by ASUHS1 during execution and the other function is an auxiliary routine used to trim the vehicle (to be discussed later in this appendix). The two script files perform the task of loading vehicle parameters and initializing the controls and vehicle dynamic variables into the MATLAB® workspace.

To initialize ASUHS1 execute the following steps:

1)Load the script file *inpar*.
2)Load the script file *initstate* .
3)Activate the SIMULINK® block structure window and open ASUHS1.

Each of these steps is detailed below.

1) Within the script file *inpar* the Earth and atmosphere model parameters are specified, as well as vehicle and engine parameters. To load the parameter values presented in Section 6.1, simply type *inpar* from within MATLAB®'s command window:

```
»inpar
```

To change any of these parameter values the user may edit *inpar*, or create a new script file. To load the new parameter values into the workspace, follow the same procedure discussed above. (Note that if any parameters are altered, it may be necessary to determine a new trimmed flight condition.)

2) ASUHS1 initial conditions are specified within the script file *initstate*. This includes the vehicle dynamic variables and control inputs used to establish a trimmed flight condition. To load the initial conditions presented in Section 6.2, simply type *initstate* from within MATLAB®'s command window:

```
»initstate
```

These initial conditions coincide with the parameter values in Section 6.1 to provide a trimmed flight condition. As discussed in Section 6.2, this flight condition is obtained by specifying a vehicle Mach number, flight path angle, position and orientation while holding all angular rates and accelerations at zero. If a different flight condition is desired, the user must follow a trim procedure, such as the one outlined in Appendix C, to obtain a new trimmed flight condition. After obtaining new initial conditions, the file *initstate* may be edited or a new script file created to store the appropriate initial conditions. To load the new flight conditions into the workspace, follow the same procedure discussed above.

Note that because some of the variables computed in *initstate* depend on values given by *inpar*, step (1) above should always precede step (2).

3) To load ASUHS1 in its block diagram structure, SIMULINK®'s block diagram window must first be activated. To activate, type *simulink* within MATLAB®'s command window:

```
»simulink
```

SIMULINK®'s block library will appear. With the simulink window active (title bar in the simulink window is highlighted), open the file titled ASUHS1. After a few seconds the ASUHS1 window will appear as in Figure A1.

The ASUHS1 window displays the three control inputs, multiplexed together, as the input to the hypersonic vehicle model, block **HV MODEL**. The outputs of this block are the freestream and engine thermodynamic conditions, vehicle dynamic variables, trim accelerations, aeropropulsive forces, engine thrust and angle of attack. The last two outputs are provided for feedback control while the others are written to MATLAB®'s workspace for further analysis. Note that a time signal is also written to the workspace as a time reference for the other output data.

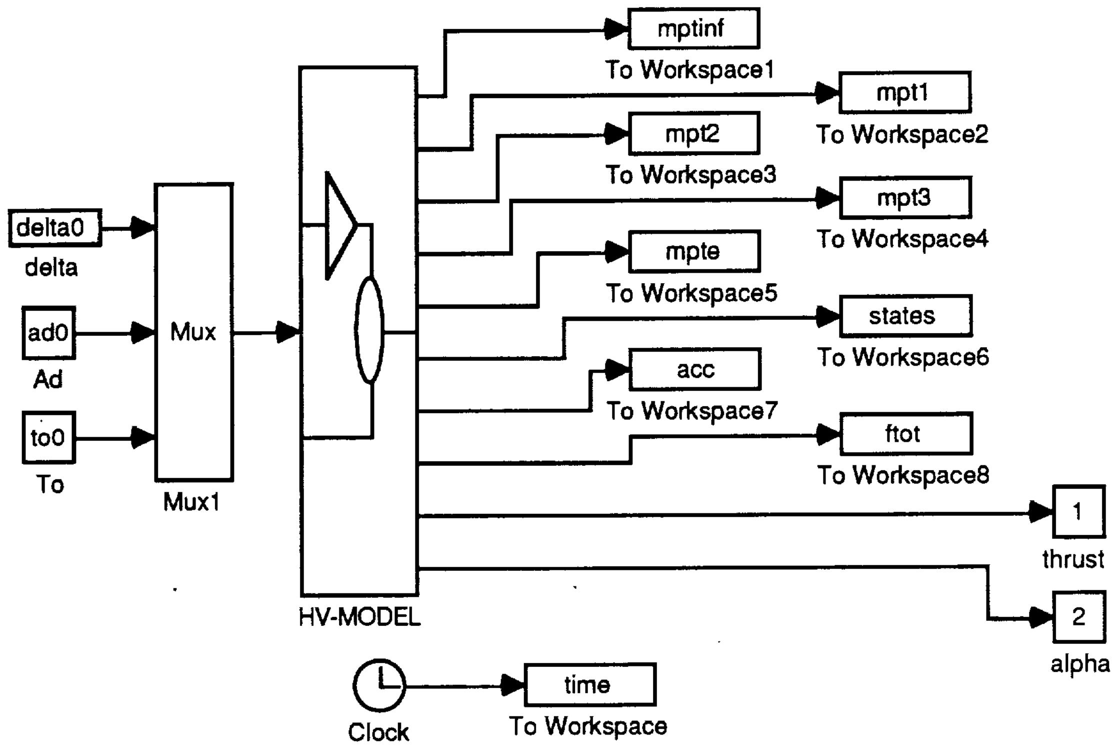

Figure A1. ASUHS1 block diagram in SIMULINK®.

From any window within the ASUHS1 block diagram structure the simulation parameters (start and stop time, integration algorithm, etc...) may be altered, via the "Parameters..." selection within the Simulation pull-down menu. The default algorithm used is RK-45 with integration time step $\in [10^{-5}, 10^{-2}]$ and relative error of 10^{-3}.

Given that the model parameters, initial conditions, ASUHS1 and simulation parameters are loaded correctly, the simulation is now ready for execution. To run a simulation, select "Start" from the Simulation pull-down menu. The simulation will run up to the stop time. The responses in Figure 5, obtained for the numerical data presented in Chapter 6, are reproduced below in Figure A2. These plots may be used to verify proper execution of ASUHS1. Due to the highly unstable behavior of this vehicle's configuration (lifting forebody), only short open-loop simulations are practical.

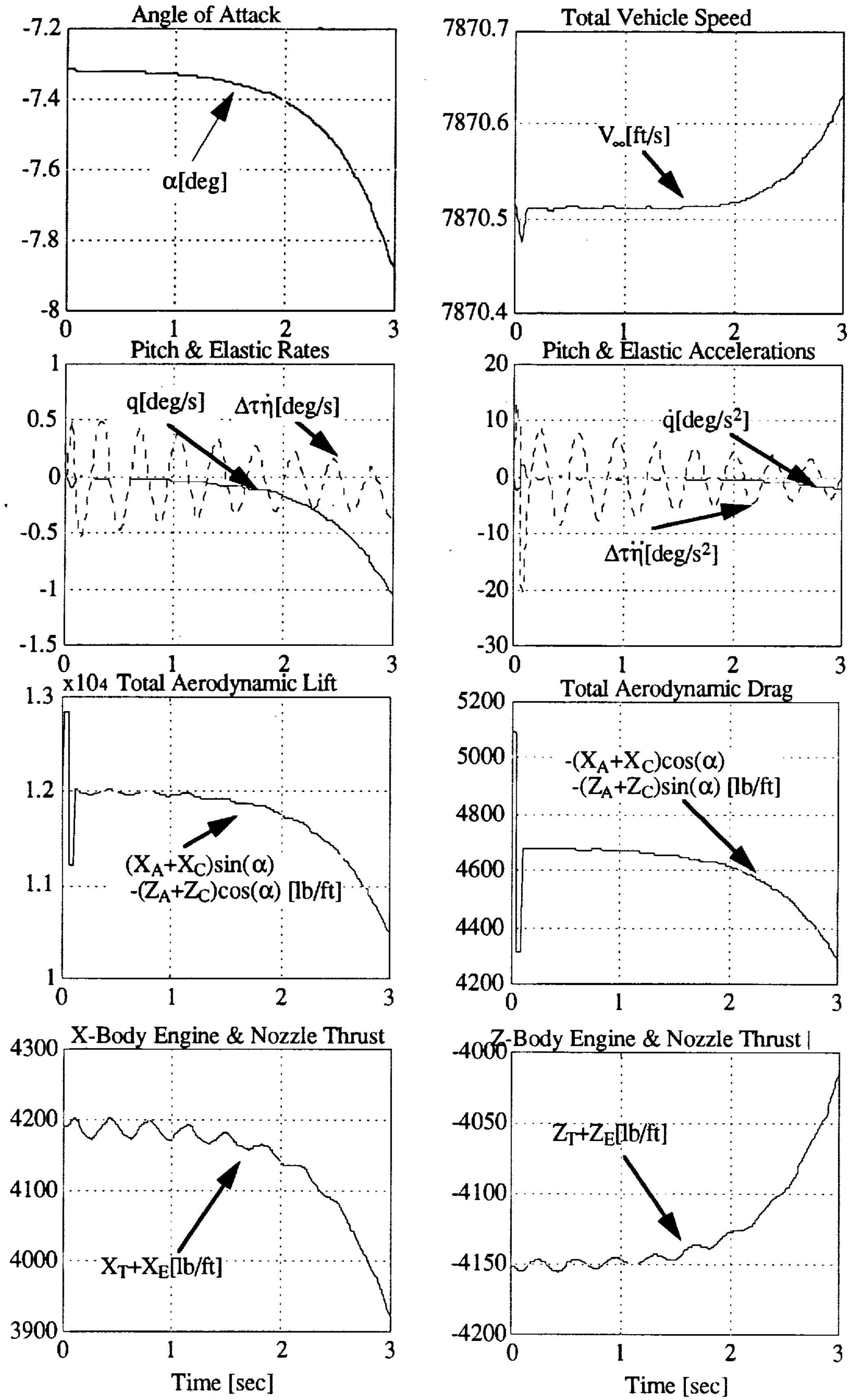

Figure A2. Simulation time responses to a doublet input in δ.

C. TRIM PROCEDURES

In this appendix a procedure is outlined which may be used to determine a trimmed flight condition for ASUHS1. The procedure makes use of the nonlinear equation solver, *fsolve* (included within MATLAB®), and relies on the flight condition structure given in Section 6.2. If a different flight condition structure is desired, this procedure should be modified to account for the different fixed variable and free variable conditions.

An equilibrium or trim condition for this flight condition structure requires that the accelerations $\dot{u}$, $\dot{w}$, $\dot{q}$ and $\ddot{\eta}$ be zero. The free parameters (α, δ, $\overline{A}_D$ and η) are determined using the *fsolve* function, which is fully documented in [6], in conjunction with the coded function *acczero*. The latter function makes use of the aeropropulsive force and moment equations in the vehicle model, as well as the equations of motion.

The computational trim procedure is as follows: For a given set of controls and dynamic variables, *fsolve* first employs *acczero* to compute the aeropropulsive forces and moments (via the equations in Chapter 4) and then the four accelerations of interest (via equations (3.4), (3.5), (3.7) and (3.8)). *fsolve* then compares each of the accelerations to zero and updates α, δ, $\overline{A}_D$ and η accordingly. For each adjustment in the free parameters made by *fsolve*, *acczero* calculates the new accelerations. This iterative process continues until the four accelerations are zero. The inputs to *fsolve* are the function *acczero* itself and the first guesses for α, δ, $\overline{A}_D$ and η. The outputs of *fsolve* are the values for α, δ, $\overline{A}_D$ and η which force the accelerations to zero. (These values are the trimmed vehicle and control settings and should be entered by the user into *initstate* for α_o, δ_o, $\overline{A}_{Do}$, and η_o.) From MATLAB®'s command window, type:

```
»controls=fsolve('acczero',[-7.2  25.0  0.45 1.0])
```

where, -7.2°, 25.0°, 0.45 and 1.0 are the first guesses for α, δ, $\overline{A}_D$ and η, respectively. (Note that all vehicle parameters and fixed variables are required to be specified within the *acczero* function. Thus any changes to the vehicle configuration or flight condition will require changes in the *acczero* code.)

The critical question of how to choose the first guesses will now be addressed This is an important yet subtle step since using arbitrary values may produce unrealistic controls or vehicle configuration. A method used in this study plots the aeropropulsive forces and moments X, Z and M, as functions of angle of attack α for different pitch control surface settings δ and different values of diffuser area ratio $\overline{A}_D$. For each choice of α, δ and $\overline{A}_D$, the forces and moments are to be computed with $\ddot{\eta}$ trimmed (i.e., with $\ddot{\eta}=0$).

The method of trimming $\ddot{\eta}$ makes use of the spring restoring force in the vehicle elastic model. The approach begins by computing the aeropropulsive forces and moments for the given choice of α, δ and $\overline{A}_D$, and $\eta=1$. The first generalized elastic co-ordinate acceleration $\ddot{\eta}$ is then computed via equation (3.5) with $\dot{\eta}$ set equal to zero. If $\ddot{\eta}$ is not zero, the elastic co-ordinate η is perturbed by an amount $\Delta\eta$ so that the perturbed spring force, $m_{E1}\omega_1^2\Delta\eta$ (the model's torsional spring constant times the perturbed elastic co-ordinate), cancels this acceleration. The amount of perturbation is determined by examining equation (3.5) in Chapter 3, so that:

$$\Delta\eta = \frac{\ddot{\eta}}{\omega_1^2}$$

Because the generalized force Q_{E1} is dependent on the elastic deflection itself, several iterations (aeropropulsive force and acceleration computations followed by further perturbation of η) are required to reduce this acceleration to zero. (Note that the final value of η in this iterative procedure will likely be different for each choice of α, δ and $\overline{A}_D$. This final η is not important; only the values of X, Z and M are of interest.)

Along with the aeropropulsive forces and moments, the gravitational force components (vehicle weight) are also plotted. The control settings where the aeropropulsive force and gravitational force components intersect provide a good first guess for α, δ and $\overline{A}_D$. In Figure A3 we see that with $\overline{A}_D \sim 0.45$, $\delta \sim 25°$ and $\alpha \sim -7.2°$, the trim conditions for X, Z and M are all nearly met, so these are good first guesses for the first three inputs to the *fsolve* command. The first guess for η, the fourth input, is not critical. A realistic value of 1.0 has been chosen here, which turns out to be close to the actual trim value given in Section 6.2.

This appendix has briefly described a procedure to determine trim conditions. Alternatively, the *trim* command in SIMULINK® could be used to obtain trimmed vehicle and control settings. However, with the present ASUHS1 set-up, experience has exposed fundamental difficulties associated with the implementation and convergence of the *trim* command. Nevertheless, there may exist some vehicle configurations for which the SIMULINK® *trim* command will execute appropriately. Thus the user who is familiar with the *trim* command may use it to determine a trim condition for his particular configuration if he so desires.

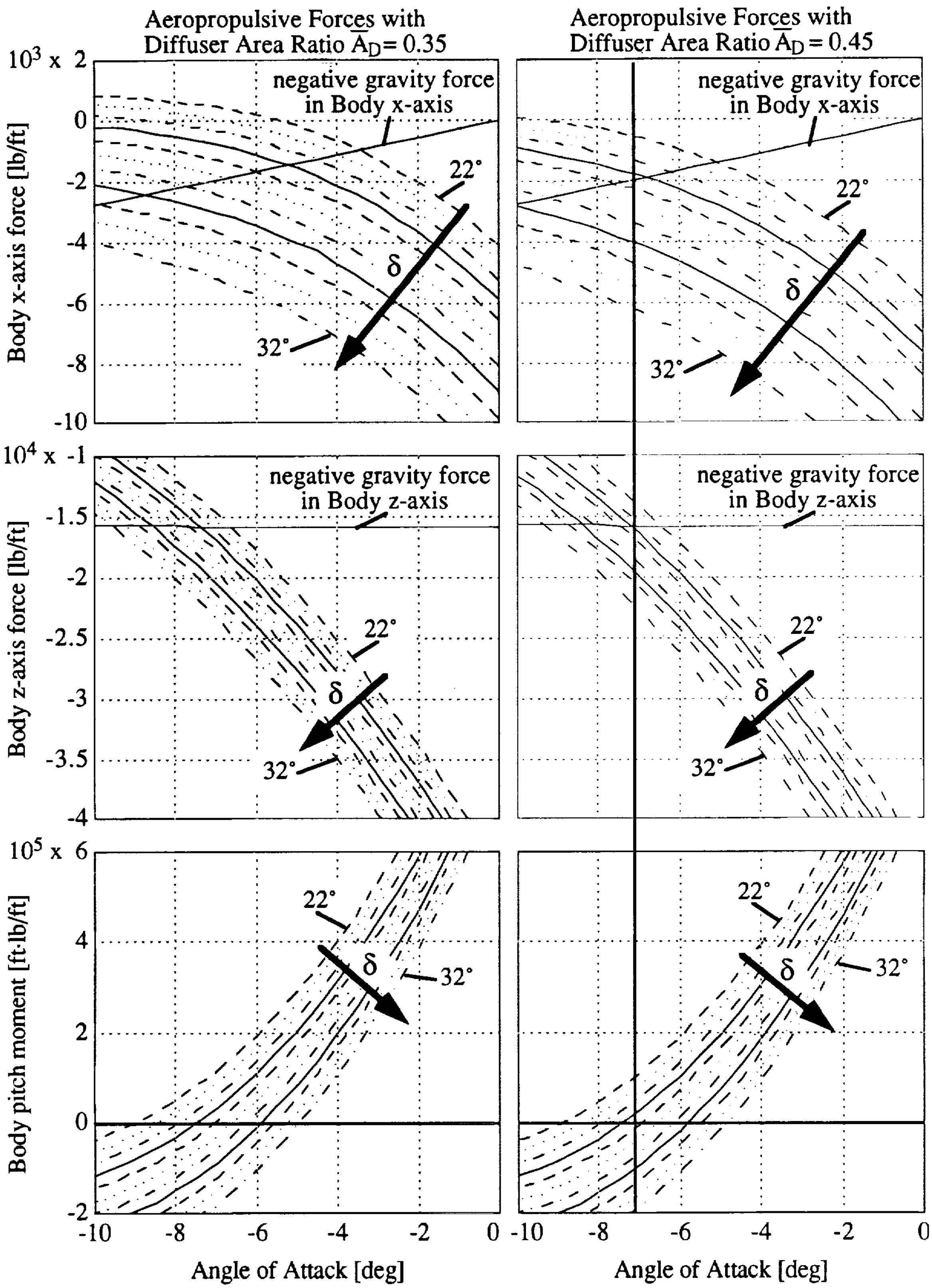

Figure A3. Aeropropulsive forces plotted as functions of angle of attack. Different pitch control surface and diffuser area control settings were used.