ANALYTICAL SENSITIVITY BASED GUIDANCE ALGORITHM FOR REUSABLE LAUNCH VEHICLES

Ashok Joshi*, K. Sivan** and Savithri Amma+

Abstract

The present study is aimed at the development of an accurate and robust reentry guidance strategy for Reusable Launch Vehicles (RLVs), based on onboard trajectory planning and by using the concept of analytical sensitivities. The onboard planning algorithm generates a feasible trajectory from any reentry interface to specific target location, while satisfying all the path constraints, prior to the start of the reentry. For this purpose, the optimal control problem of reentry guidance is converted into an equivalent targeting problem in Nonlinear Programming and a simple solution methodology is devised to generate the three-dimensional trajectory. The profile tracking algorithm is developed based on the well known Linear Quadratic Regulator technique. In this strategy, equations of motion are described in polar coordinates and a direct analytical method for computation of sensitivity matrix elements is used, which ensures better mission planning and faster convergence. An additional feature of the proposed algorithm is the inclusion of an integral term in control law, which tracks the trajectory without a need for instantaneous bank reversals, even with a dispersed environment. The performance results establish adequacy and usefulness of the algorithm.

Nomenclature

$\boldsymbol{C}$	= constraint vector
C_D	= drag coefficient
C_L	= lift coefficient
D	= drag force
e_t; e_c	= target and constraint error
g ; g_R ; g_ϕ	= acceleration due to gravity; J2 Harmonics
h ; h_o ; h_f; h_d	= altitude; initial value; final value; desired value
H	= weighting matrix
J	= performance index, Jacobian matrix
J_2	= second gravitational harmonics
$K(e)$, $K_I(e)$	= control gains
L	= lift force
m	= vehicle mass
M	= mach number
n_z	= load factor
q; $q_{\max}$	= dynamic pressure; maximum allowable value
Q	= weighing matrix
$\dot{Q}$; $\dot{Q}_{\max}$	= stagnation point heat rate; maximum allowable value
r ; r_o ; r_f; r_d	= radial distance; initial value; final value; desired value
S	= reference area; minimum norm search direction vector
t_o ; t_f ; t_a	= initial time; final time; time limit for constant α
$\boldsymbol{u}_c$; $\boldsymbol{u}^*$; $\boldsymbol{u}_{ref}$	= control vector; optimum value; reference value
$\boldsymbol{V}$; $\boldsymbol{V}_f$; $\boldsymbol{V}_d$	= relative velocity; final relative velocity; desired relative velocity
V_{cir}	= circular velocity
$\boldsymbol{X}$; $\boldsymbol{X}_f$; $\boldsymbol{X}_d$	= instantaneous state vector; final state vector; desired state vector
$\boldsymbol{Y}(t)$	= target error vector
α ; α_n	= angle of attack; nominal value at the beginning
$\dot{\alpha}$	= rate of change of angle of attack
β	= side slip angle
γ ; γ_f; γ_d	= flight path angle; final value; desired value
η_i	= path constraint error sensitivity states

* Professor, Department of Aerospace Engineering, Indian Institute of Technology Bombay, Powai, Mumbai-400 076, India
Email : ashokj@aero.iitb.ac.in
** Group Director + Engineer
Mission Synthesis and Simulation Group, Vikram Sarabhai Space Centre, Thiruvananthapuram-695 022, India

Manuscript received on 19 Sep 2006; Paper reviewed and accepted on 15 May 2007

ξ_i	= target error sensitivity state
λ ; λ_f; λ_d	= vehicle longitude; final value; desired value
μ	= gravitational constant
ρ	= atmospheric density
σ ; σ_d ; σ_f	= bank angle; value at the initiation of the manoeuvre; final value
σ_t	= maximum allowable limit on the final bank angle
$\dot{\sigma}_1$ *to* $\dot{\sigma}_6$	= parameterized bank angle rates
ϕ ; ϕ_f; ϕ_d	= latitude; final value; desired value
ψ ; ψ_f; ψ_d	= azimuth; final value; desired value
Ω_e , ω_e	= rotational velocity of earth

Introduction

In general, reentry guidance strategy for RLVs consists of two components : (1) generation of a feasible reference trajectory to meet the specified target, satisfying all the path constraints and (2) generation of steering commands, in terms of angle of attack and bank angle, to track the reference trajectory. The flight proven reentry guidance algorithms [1,2] to date have been implemented by fitting the optimal trajectory with phases that follow a ground computed reference trajectory profile stored onboard. Studies reveal that the Space Shuttle guidance algorithms [1,2] need improvement in order to meet the requirements of new generation Reusable Launch Vehicles (RLVs), which have a wide dynamic range of operation and are subjected to minimal aerodynamic characterization at ground to reduce the overall mission cost. In recent studies, autonomous and adaptive reentry guidance algorithms, which generate a reference trajectory onboard, based on flight conditions and mission requirements, are being developed. In a three dimensional predictive reentry guidance approach [3], the Space Shuttles two dimensional reentry planning method is extended to three dimensions, in order to achieve the desired down range and cross range through drag and lateral acceleration profiles, while a nonlinear tracking control law is used for trajectory control. Ronneke [4] has presented an adaptive reentry guidance algorithm based on autonomous trajectory planning and nonlinear tracking. In another guidance scheme for Hope-X [5], iterative predictor-corrector method is adopted to meet the required energy height at Terminal Area Energy Management (TAEM) interface and reference drag profile is generated onboard from the derivative of energy height with respect to range-to-go. Fuhry [6] has developed an efficient and simple reentry guidance algorithm based on numerical predictor-corrector approach for Kistler K-1 orbital vehicle. Dukeman [7] has presented a simple reentry trajectory guidance algorithm based on optimal Linear Quadratic Regulator (LQR) for tracking longitudinal variables.

Zimmerman et.al [8] have presented an effective integrated reentry guidance algorithm, which contains onboard trajectory planning with heat constraints, that uses the numerical predictor-corrector method and a trajectory tracking law based on LQR technique. Evolved Acceleration Guidance Logic for Entry (EAGLE) developed by Saraf et.al [9] consists of two integrated components : (1) a three-dimensional trajectory planner which generates reference drag acceleration and (2) lateral acceleration profiles, along with reference state and bank angle profiles. Recently, Shen and Lu [10] have developed an efficient methodology for faster design of three-degrees-of-freedom reentry trajectories, subjected to all common equality and inequality constraints, applicable to onboard trajectory planning in entry guidance. Along with this algorithm, onboard trajectory planning for sub-orbital flights developed by Shen and Lu [11] covers the full envelope of reentry trajectory planning. In order to overcome the issues related to conventional bank reversals, Shen and Lu [12] have developed a new automated lateral guidance logic based on cross range. Joshi and Sivan [13] have presented a predictor-corrector algorithm based on Ref.[6], which includes the angle of attack and bank angle modulations for improving the reentry capability and satisfying the path constraints. All the existing strategies either carry out real-time planning onboard during the reentry phase or use the ground computed reference trajectory, which is only tracked during the reentry phase.

In this study, a new reentry guidance algorithm is proposed, which uses the numerical predictor-corrector for off-line onboard trajectory planning and the LQR technique, along with an integral term in control law, for accurate tracking of the trajectory. Guidance algorithm presented in this paper addresses the important issue of accurate estimation of the error sensitivities through an analytical approach, by defining errors sensitivities as the additional system states. The solution methodology presented in this paper has the potential to be implemented on the flight guidance computers of the future generation of the RLVs.

Reentry Guidance Problem

Reentry Dynamics

The three degrees of freedom (3 DOF) system dynamics, in polar coordinates with respect to an oblate rotating planet [14] is given below :

$$\dot{V} = -\left(\frac{D}{m}\right) + g_R \sin\gamma - g_\phi \cos\gamma \cos\psi$$
$$+ r\,\Omega_e^2 \cos\phi\,(-\cos\gamma\cos\psi\sin\phi + \sin\gamma\cos\phi) \quad (1)$$

$$\dot{\gamma} = \frac{1}{V}\left(\frac{L}{m}\right)\cos\sigma + \frac{1}{V}\left[g_R\cos\gamma - g_\phi\sin\gamma\cos\psi + \frac{V^2}{r}\cos\gamma\right]$$
$$+\ 2\,\Omega_e \sin\psi\cos\phi$$
$$+ \frac{r\Omega_e^2\cos\phi}{V}(\sin\gamma\cos\psi\sin\phi + \cos\gamma\cos\phi) \quad (2)$$

$$\dot{\psi} = \frac{1}{V\cos\gamma}\left(\frac{L}{m}\right)\sin\sigma + \frac{1}{V\cos\gamma}\left(g_\phi\sin\psi\right)$$
$$\frac{V}{r}\cos\gamma\sin\psi\tan\phi + 2\Omega_e\,(\sin\phi - \tan\gamma\cos\psi\cos\phi)$$
$$+ \frac{r\Omega_e^2}{V\cos\gamma}\cos\phi\sin\psi\sin\phi \quad (3)$$

$$\dot{r} = V\sin\gamma\,; \quad \dot{\phi} = \frac{V\cos\gamma\cos\psi}{r}\,; \quad \dot{\lambda} = \frac{V\cos\gamma\sin\psi}{r\cos\phi} \quad (4)$$

The gravity components with J2 term are given below :

$$g_R = -\frac{\mu}{r^2}\left[1 - \frac{3}{2}J_2\left(\frac{R_e}{r}\right)^2 (3\sin^2\phi - 1)\right] \quad (5)$$

$$g_\phi = \frac{3\mu}{r^2}\,J_2\left(\frac{R_e}{r}\right)^2 \sin\phi\cos\phi \quad (6)$$

The aerodynamic lift and drag forces are given below :

$$L = \frac{1}{2m}\rho\,(r)\,V^2\,SC_L\,(\alpha\,, M\,, r)\,;$$
$$D = \frac{1}{2m}\rho\,(r)\,V^2\,SC_D\,(\alpha\,, M\,, r) \quad (7)$$

Terminal Target Conditions and Path Constraints

The terminal target conditions are stated as requirements for achieving a specified Terminal Area Energy Management (TAEM) interface state. Normally a RLV is said to have reached TAEM interface when its velocity reduces to the desired TAEM velocity. At that point on the trajectory there are stringent requirements on range to heading alignment cylinder, altitude and flight path angle. All these requirements can be met if the terminal trajectory state parameters achieve the values that are dictated by the TAEM interface requirements. Also, in the proposed algorithm, it is easier to handle the trajectory state parameters rather than interface conditions in terms of derived parameters. Therefore in the generic algorithm, the terminal target conditions are defined in terms of the state variables as,

$$V_f - V_d = 0\,; \quad \gamma_f - \gamma_d = 0\,; \quad \psi_f - \psi_d = 0$$
$$r_f - r_d = 0\,; \quad \phi_f - \phi_d = 0\,; \quad \lambda_f - \lambda_d = 0 \quad (8)$$

where, $V_f, \gamma_f, \psi_f, r_f, \phi_f, \lambda_f$ are the state at the TAEM interface and $V_d\,, \gamma_d\,, \psi_d\,, r_d\,, \phi_d\,, \lambda_d$ are the corresponding desired values. In addition to the above target conditions, in order to avoid large transients in TAEM guidance, the bank angle magnitude at TAEM interface has to be less than a specified value [9] as,

$$|\,\sigma_f\,| \le \sigma_t \quad (9)$$

Here, σ_f is the bank angle at TAEM interface and σ_t is its maximum allowable limit. The applicable path constraints e.g. equilibrium glide, heat flux, load factor and dynamic pressure, are given below :

$$\frac{L}{m}\cos(\sigma) \le \left[g - \frac{V^2}{r}\right]\cos(\gamma)\,;$$

$$\dot{Q} = \frac{11030}{\sqrt{R_N}}\left[\frac{\rho}{\rho_0}\right]^{1/2}\left[\frac{V}{V_{cir}}\right]^{3.15} \le \dot{Q}_{\max} \quad (10)$$

$$n_z = \frac{D}{m}\left[1 + (L/D)^2\right]^{1/2} \le n_{\max}\,;$$

$$q = (1/2)\,\rho\,V^2 \le q_{\max} \quad (11)$$

Overall Guidance Concept

The generic reentry guidance algorithm presented in this paper contains two parts viz., a trajectory planning algorithm executed before reentry phase and a profile tracking algorithm executed during reentry. After the de-boost manoeuvre, the vehicle follows Keplerian motion and, therefore, the expected state at reentry (altitude of 120 km) can be computed analytically from the navigation data at the end of de-boost, which can be taken as the initial conditions for the actual reentry. In general, about 3000s of flight time is available between the end of de-boost and start of the reentry and this time interval is proposed to be used for executing the computationally intensive planning algorithm for generating the off-line reference trajectory.

The trajectory planning solution is achieved through a NLP approach for targeting which takes care of widely different initial conditions at reentry interface and meets the target within vehicles reentry capability, without violating any of the path constraints. Nominal vehicle data and environment characteristics are used in this numerical iteration procedure. Output of the algorithm are (1) nominal control history and (2) the corresponding reference trajectory profile from the designated reentry interface to the TAEM interface satisfying all the path constraints and target conditions. If the vehicle and flight environment parameters during reentry are exactly the same as those assumed for trajectory planning, the reference control history as designed above will meet the mission requirements. However, in reality, this is not true and hence the planned control history will not ensure the desired mission objectives. To overcome these difficulties during reentry phase, the profile tracking algorithm is designed, which modulates the control history generated by the planning algorithm, so as to follow the reference trajectory. Since all the trajectory parameters are tracked from re-entry interface to TAEM interface, all the path constraints are satisfied while meeting target conditions. The overall schematic of the proposed generic guidance algorithm is given in Fig.1.

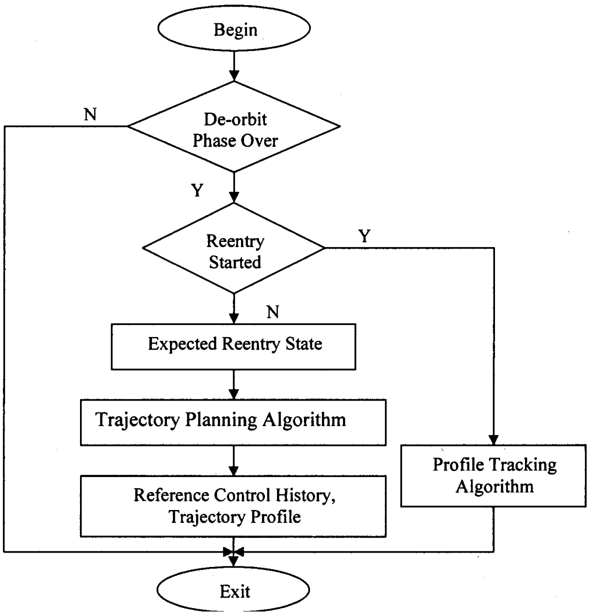
Begin
De-orbit Phase Over
N
Y
Reentry Started
Y
N
Expected Reentry State
Trajectory Planning Algorithm
Reference Control History, Trajectory Profile
Profile Tracking Algorithm
Exit

Fig.1 Overall schematic of generic reentry guidance algorithm

Generic Reentry Guidance Algorithm

Trajectory Planning Guidance Law

The target error vector at TAEM interface is defined as

$$Y=\left[V_f - V_d \; \gamma_f - \gamma_d \; \psi_f - \psi_d \; r_f - r_d \; \phi_f - \phi_d \; \lambda_f - \lambda_d \; \sigma_f - \sigma_t\right]^T \tag{12}$$

where, $(.)_f$ denotes the parameters at the predicted TAEM interface and $(.)_d$, $(.)_t$ denote the corresponding desired values. The individual component of terminal error vector is considered into a terminal scalar function with relative weightages, and the scalar error e_t is computed as,

$$e_t = \boldsymbol{Y}^T \boldsymbol{H} \boldsymbol{Y} \tag{13}$$

where, $\boldsymbol{H}$ is a weighing matrix for the terminal errors. The path constraint error is defined as an integral of square of the amount by which the constraint is violated. For the i^{th} constraint, the instantaneous constraint error is defined as

$$\delta C_i = \begin{Bmatrix} \Delta C_i & \text{if } \Delta C_i = C_i - C_{l_i} > 0 \\ 0\,, & \textit{otherwise} \end{Bmatrix} \tag{14}$$

where, C_{l_i} is the limit of i^{th} constraint parameter. The individual component of path constraint error vector is combined into another scalar function with weightages, and the corresponding scalar error, e_c is computed as

$$e_c = \int_{t_0}^{t_f} \delta \boldsymbol{C}^T \boldsymbol{W} \, \delta \boldsymbol{C} \; dt \tag{15}$$

where, $\delta\boldsymbol{C}$ is constraint error vector with components defined as given in Eqn. (14) and $\boldsymbol{W}$ is weighing matrix used for the relative weightage between different constraints. The guidance solution is achieved when $e_t = 0$ and

$e_c = 0$ simultaneously. Therefore, the guidance problem can be considered as a targeting problem, which can be stated as follows :

Given X_0 *and* t_0, *compute* $\boldsymbol{u}_c$ (t) $= \{\alpha(t) \quad \sigma(t)\}^T$ *to achieve* $e_t = 0$, $e_c = 0$ *subject to the dynamics given by* $\dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{u}_c)$. In reality, exact zeros for the errors are difficult to achieve and hence the solution is declared achieved once the errors are less than the allowable tolerances. The diagonal elements of weighing matrices are selected as inverse of the square of the allowable error in each parameter which ensures the solution when $e_t \leq 1$ and $e_c \leq 1$. The optimal control problem can now be converted into NLP problem by parameterizing the control history $\boldsymbol{u}_c(t)$ as given below :

$$u_c(t) = u_k + \frac{(u_{k+1} - u_k)(t - t_k)}{(t_{k+1} - t_k)}, \quad t_k < t \;\; t_{k+1} \tag{16}$$

Here u_k are the control parameters. If $\boldsymbol{u}$ denotes the vector of control parameters, then this problem can be viewed as targeting problem in NLP, requiring computation of the control vector $\boldsymbol{u}^*$ which ensures that the error vector $e(\boldsymbol{u}^*) = \begin{bmatrix} e_t(\boldsymbol{u}^*) \\ e_c(\boldsymbol{u}^*) \end{bmatrix}$ goes to zero, subject to the system dynamics, $\dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}(t), \boldsymbol{u})$ and the initial conditions $\boldsymbol{X}_0$. The solution methodology developed for this problem is as follows :

The solution is started with an initial guess for the control vector, $\boldsymbol{u}_0$ and the error vector for this reference control vector, $\boldsymbol{e}(\boldsymbol{u}_0)$ is computed. The next step is to find the increment on the control vector, $\delta\boldsymbol{u}$ so that the control vector, $\boldsymbol{u}^* = \boldsymbol{u}_0 + \delta\,\boldsymbol{u}$ ensures that the error vector, $\boldsymbol{e}(\boldsymbol{u}^*)$ goes to zero. Defining the gradient vectors of the target error and constraint error at the reference control vector $\boldsymbol{u}_0$ as

$$\nabla^T_{e_t}(\boldsymbol{u}_0) = \begin{bmatrix} \frac{\partial e_t}{\partial u_1} & \frac{\partial e_t}{\partial u_2} & \cdots & \frac{\partial e_t}{\partial u_k} \end{bmatrix}_{u_0} \quad ;$$

$$\nabla^T_{e_c}(\boldsymbol{u}_0) = \begin{bmatrix} \frac{\partial e_c}{\partial u_1} & \frac{\partial e_c}{\partial u_2} & \cdots & \frac{\partial e_c}{\partial u_k} \end{bmatrix}_{u_0} \tag{17}$$

The sensitivity matrix of the error vector is given as

$$\boldsymbol{S}(\boldsymbol{u}_0) = \begin{bmatrix} \nabla^T_{e_t}(\boldsymbol{u}_0) \\ \nabla^T_{e_c}(\boldsymbol{u}_0) \end{bmatrix}$$

and the solution to the above problem is obtained by finding optimum search direction and optimum step length along the search direction for driving the error vector to zero. Assuming the target error and constraint error $\boldsymbol{u}_0$ to be linear, the unique optimum search direction $\Delta\boldsymbol{u}$ is the one, which solves the linearized error equation

$$\boldsymbol{S}(\boldsymbol{u}_0)\,\Delta\,\boldsymbol{u} + \boldsymbol{e}(\boldsymbol{u}_0) = 0 \tag{18}$$

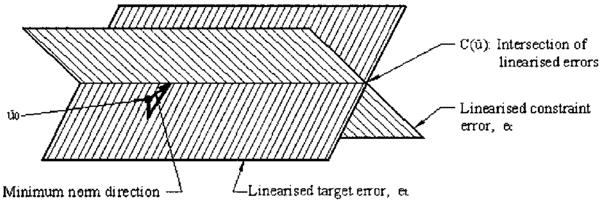
C(ū): Intersection of linearised errors
Linearised constraint error, ec
ū0
Minimum norm direction Δu
Linearised target error, et

Fig.2 Minimum norm correction direction

and minimizes the length of $\Delta\boldsymbol{u}$. The solution to the above equation defines a linear manifold $C(\boldsymbol{u}_0)$ (in this case, a straight line) as given in Fig.2. The desired minimum norm correction $\Delta\boldsymbol{u}$ is then the vector of minimum length from $\boldsymbol{u}_0$ to the linear manifold $C(\boldsymbol{u}_0)$. Analytically, this is given as

$$\Delta\,\boldsymbol{u} = -\boldsymbol{S}^T(\boldsymbol{u}_0)\left[\boldsymbol{S}(\boldsymbol{u}_0)\,\boldsymbol{S}^T(\boldsymbol{u}_0)\right]^{-1}\boldsymbol{e}(\boldsymbol{u}_0) \tag{19}$$

If the errors are actually linear, then the Eqn. (19) ensures zero errors. For the realistic case, an optimum length along the minimum-norm direction is required to minimize the error and this is computed by one dimensional minimization method. The study indicates that quadratic interpolation method is sufficient to find the optimum step length. The function to be minimized along the search direction, $\Delta\boldsymbol{u}$ is the sum of the squares of the errors as,

$$J_0(\delta) = |\,\boldsymbol{e}(\boldsymbol{u}_0 + \delta\,\Delta\,\boldsymbol{u})\,|^2 \tag{20}$$

This gives

$$J_0(0) = |\boldsymbol{e}(\boldsymbol{u}_0)|^2 \text{ and } J_0'(0) = 2e^T(\boldsymbol{u}_0)\, S(u_0)\, \Delta u \quad (21)$$

$\Delta \boldsymbol{u}$ is estimated assuming the constraints are linear. This approximation is valid for small interval and hence, provides reasonable estimate for δ as unity for minimizing J_0.

$$J_0(1) = |\boldsymbol{e}(\boldsymbol{u}_0 + \Delta \boldsymbol{u})|^2 \quad (22)$$

A quadratic polynomial is fitted for the function given in Eqn. (20) with $J_0(0)$, $J_0'(0)$, $J_0(1)$ and the optimum step length δ^* is computed. The optimum control vector is computed as

$$\boldsymbol{u}^* = \boldsymbol{u}_0 + \delta^* \Delta \boldsymbol{u} \quad (23)$$

The optimum solution is achieved through iteration with updated value of $\boldsymbol{u} = \boldsymbol{u}^*$ till the error vector is within the allowable tolerances.

Models for the Off-line Onboard Trajectory Planning Algorithm

The models required for the application of the generic trajectory planning algorithm are given in this section. The development involves the selection of control vector and computation of correct sensitivity matrix for three-dimensional planning. The model is developed for a simplified RLV mission, without any loss of generality, with the guidance objective of steering the vehicle from any reentry interface point to the specified YAEM location (altitude, latitude, longitude) without violating the heat flux constraint. In order to avoid large transients during TAEM phase, the bank angle at the TAEM interface has to be less than a specified level. These models are general enough so that these can be used for any RLV mission trajectory planning, with different objectives, targets and path constraints. The TAEM interface target conditions assumed for the model development are,

$$r_f - r_d = 0\,;\ \phi_f - \phi_d = 0\,;\ \lambda_f - \lambda_d = 0\,;\ \sigma_f - \sigma_t = 0 \quad (24)$$

The path constraint considered here is the requirement of stagnation point heat rate being less than the allowable limit, $\dot{Q}_{\max}$ as given below :

$$\dot{Q} - \dot{Q}_{\max} \le 0\,;\ \dot{Q} = \frac{11030}{\sqrt{R_N}} \left[\frac{\rho}{\rho_0}\right]^{1/2} \left[\frac{V}{V_{cir}}\right]^{3.15} \quad (25)$$

Control Vector

The three-dimensional trajectory planning is achieved by steering through α and σ profiles. In the present steering, it is planned to achieve the required three-dimensional trajectory only through σ modulation, keeping the predefined profile for α intact. Usage of predefined profile for α provides a major advantage of ensuring the vehicle and mission constraints such as trim limits as well as better thermal management. In the present planning algorithm development, the angle of attack steering is done through the fixed angle of attack profile as given in Fig.3. At any instant of time, the angle of attack is computed as,

$$\begin{aligned} &\text{for } t \le t_a, \quad \alpha(t) = \alpha_n\,; \\ &\text{for } t > t_a, \quad \alpha(t) = \alpha_n - \dot{\alpha}(t - t_a) \end{aligned} \quad (26)$$

where predefined values are used for α_n and $\dot{\alpha}$. The bank angle history is assumed to be zero during initial phase of reentry when the vehicle flies with high angle of attack, which ensures better thermal management in high velocity regime. A non-zero value of starting σ_d is initiated after a pre-defined time and from this time onwards till the TAEM interface point, the bank angle modulations are assumed to achieve the desired trajectory to meet the target, satisfying the path constraint. Since the three-dimensional trajectory planning solution is achieved through NLP, the σ history is parameterized as given in Fig.4. At any instant of time after t_1, the bank angle is computed as

$$\sigma(t) = \sigma_d + \left[\sum_{j=1}^{i-1} \dot{\sigma}_j (t_{J+1} - t_j)\right] + \dot{\sigma}_i (t - t_i) \quad (27)$$

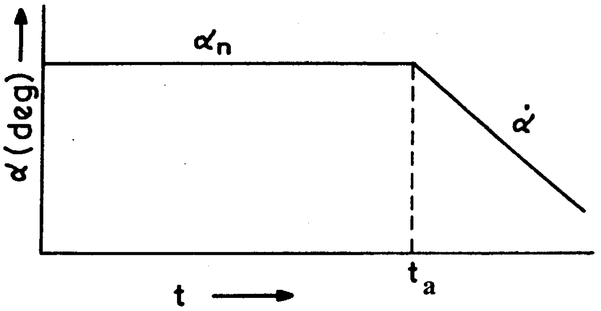

Fig.3 Angle of attack history

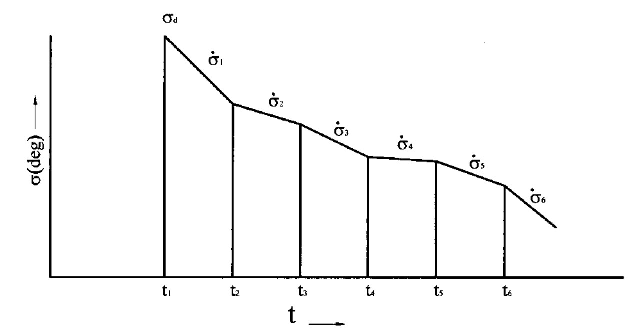

Fig.4 Bank angle history

The parameters considered are the initial bank angle at the time of bank initiation and bank angle rates during the remaining phase of the flight, whereas the times of initiation of rates are predefined values. It is to be noted here that larger number of parameters ensure faster solution. In the present study, only seven parameters are considered for the model development. Therefore, the control vector assumed for this study is given as,

$$\boldsymbol{u} = \begin{bmatrix} \sigma_d & \dot{\sigma}_1 & \dot{\sigma}_2 & \dot{\sigma}_3 & \dot{\sigma}_4 & \dot{\sigma}_5 & \dot{\sigma}_6 \end{bmatrix}^T \tag{28}$$

As per the mission requirements, the parameters can be increased and the model developed in this section can be used for the extended case also.

Error Vector

The terminal constraint is to achieve the specified TAEM location along with the specified bank angle as given in Eqn. (24). In the planning algorithm, the trajectory is propagated up to the specified TAEM interface altitude and the error vector and therefore, target error and constraint error are defined as,

$$\boldsymbol{Y} = \begin{bmatrix} \phi\,(r_f) - \phi_d & \lambda\,(r_f) - \lambda_d & \sigma\,(r_f) - \sigma_t \end{bmatrix}^T \tag{29}$$

$$e_c = \int_{t_0}^{t_f} \Delta \dot{Q}_c^{\,2}\, dt \quad ; \; \delta \dot{Q}_c = \begin{Bmatrix} \Delta \dot{Q} & \text{if } \; \Delta \dot{Q} = \dot{Q} - \dot{Q}_1 \\ 0\,, & otherwise \end{Bmatrix} \tag{30}$$

Sensitivity Matrix Computation

In gradient based methods, the efficiency and accuracy of the solution depends on the correctness of the sensitivity matrix. In literature [8, 13] the sensitivity matrix computations are generally carried out using the finite differencing. In these methods, the correctness of the sensitivity depends on the perturbation levels used. Further, the process of generating sensitivity using finite differencing method for the multiple control vectors also requires large computational effort as well as large capacity storage for all the trajectory related data. In addition, the resolution of particular computer system strongly affects the perturbation selection and hence cannot be made general. In order to avoid these problems, the following analytical approach is proposed. For the system dynamics given as $\dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{u})$ where, $\boldsymbol{u}$ is the control vector, the derivatives of the sensitivity of state can be defined as,

$$\frac{d}{dt}\left(\frac{\partial X}{\partial u_i}\right) = \frac{\partial \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{u})}{\partial u_i} \tag{31}$$

where, $\frac{\partial \boldsymbol{X}}{\partial u_i}$ is the sensitivity of system state vector with respect to the control parameter, u_i. This allows one to define additional states for the target error sensitivity, which are as given below :

$$\xi_1 = \begin{Bmatrix} \frac{\partial V}{\partial \sigma_d} & \frac{\partial \gamma}{\partial \sigma_d} & \frac{\partial \psi}{\partial \sigma_d} & \frac{\partial r}{\partial \sigma_d} & \frac{\partial \phi}{\partial \sigma_d} & \frac{\partial \lambda}{\partial \sigma_d} \end{Bmatrix}^T$$

$$\xi_2 = \begin{Bmatrix} \frac{\partial V}{\partial \dot{\sigma}_1} & \frac{\partial \gamma}{\partial \dot{\sigma}_1} & \frac{\partial \psi}{\partial \dot{\sigma}_1} & \frac{\partial r}{\partial \dot{\sigma}_1} & \frac{\partial \phi}{\partial \dot{\sigma}_1} & \frac{\partial \lambda}{\partial \dot{\sigma}_1} \end{Bmatrix}^T$$

$$\xi_3 = \begin{Bmatrix} \frac{\partial V}{\partial \dot{\sigma}_2} & \frac{\partial \gamma}{\partial \dot{\sigma}_2} & \frac{\partial \psi}{\partial \dot{\sigma}_2} & \frac{\partial r}{\partial \dot{\sigma}_2} & \frac{\partial \phi}{\partial \dot{\sigma}_2} & \frac{\partial \lambda}{\partial \dot{\sigma}_2} \end{Bmatrix}^T$$

$$\xi_4 = \begin{Bmatrix} \frac{\partial V}{\partial \dot{\sigma}_3} & \frac{\partial \gamma}{\partial \dot{\sigma}_3} & \frac{\partial \psi}{\partial \dot{\sigma}_3} & \frac{\partial r}{\partial \dot{\sigma}_3} & \frac{\partial \phi}{\partial \dot{\sigma}_3} & \frac{\partial \lambda}{\partial \dot{\sigma}_3} \end{Bmatrix}^T$$

$$\xi_5 = \begin{Bmatrix} \frac{\partial V}{\partial \dot{\sigma}_4} & \frac{\partial \gamma}{\partial \dot{\sigma}_4} & \frac{\partial \psi}{\partial \dot{\sigma}_4} & \frac{\partial r}{\partial \dot{\sigma}_4} & \frac{\partial \phi}{\partial \dot{\sigma}_4} & \frac{\partial \lambda}{\partial \dot{\sigma}_4} \end{Bmatrix}^T$$

$$\xi_6 = \begin{Bmatrix} \frac{\partial V}{\partial \dot{\sigma}_5} & \frac{\partial \gamma}{\partial \dot{\sigma}_5} & \frac{\partial \psi}{\partial \dot{\sigma}_5} & \frac{\partial r}{\partial \dot{\sigma}_5} & \frac{\partial \phi}{\partial \dot{\sigma}_5} & \frac{\partial \lambda}{\partial \dot{\sigma}_5} \end{Bmatrix}^T$$

$$\xi_7 = \begin{Bmatrix} \frac{\partial V}{\partial \dot{\sigma}_6} & \frac{\partial \gamma}{\partial \dot{\sigma}_6} & \frac{\partial \psi}{\partial \dot{\sigma}_6} & \frac{\partial r}{\partial \dot{\sigma}_6} & \frac{\partial \phi}{\partial \dot{\sigma}_6} & \frac{\partial \lambda}{\partial \dot{\sigma}_6} \end{Bmatrix}^T \tag{32}$$

The corresponding dynamic equations for the sensitivity coefficients are given below :

$$\dot{\xi}_i = \boldsymbol{L}\,\xi_i + \boldsymbol{M}_{\boldsymbol{i}} \tag{33}$$

The elements of matrix $\boldsymbol{L}$ and vectors $\boldsymbol{M}_i$ are derived from the system dynamics. The states for the constraint error sensitivity are as given below :

$$\eta = \left\{ \frac{\partial e_c}{\partial \sigma_d} \; \frac{\partial e_c}{\partial \dot{\sigma}_1} \; \frac{\partial e_c}{\partial \dot{\sigma}_2} \; \frac{\partial e_c}{\partial \dot{\sigma}_3} \; \frac{\partial e_c}{\partial \dot{\sigma}_4} \; \frac{\partial e_c}{\partial \dot{\sigma}_5} \; \frac{\partial e_c}{\partial \dot{\sigma}_6} \right\}^T \tag{34}$$

and the corresponding dynamic equations for the sensitivity coefficients are given below :

$$\dot{\eta}_1 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{11} - \frac{\beta}{2}\,\xi_{14}\right];$$

$$\dot{\eta}_2 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{21} - \frac{\beta}{2}\,\xi_{24}\right]$$

$$\dot{\eta}_3 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{31} - \frac{\beta}{2}\,\xi_{34}\right];$$

$$\dot{\eta}_4 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{41} - \frac{\beta}{2}\,\xi_{44}\right]$$

$$\dot{\eta}_5 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{51} - \frac{\beta}{2}\,\xi_{54}\right];$$

$$\dot{\eta}_6 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{61} - \frac{\beta}{2}\,\xi_{64}\right]$$

$$\dot{\eta}_7 = 2\,\dot{Q}^*\;\Delta\dot{Q}_c \left[\frac{3.15}{V}\,\xi_{71} - \frac{\beta}{2}\,\xi_{74}\right] \tag{35}$$

The dynamic Eqns. given in (33, 35) are integrated along with the system dynamics given in Eqns. (1-4) and these additional states at final time are used for the sensitivity matrix computations defined in Eqn. (17).

Trajectory Control Guidance Law

In the present study, an efficient and simple trajectory control algorithm for regulating the actual trajectory, to be close to the reference trajectory, based on LQR technique [7,12] is used. In ref. [7,12], the LQR theory is applied effectively for tracking the longitudinal parameters whereas lateral control is achieved by bank reversals. In the present study, it is seen that with the proper design of gains and tracking of all the six state variables simultaneously along with an integrator, the tracking algorithm works very well and also eliminates the need for instantaneous bank reversals, employed in all tracking algorithms reported in literature.

Application of LQR Theory for Trajectory Control

The general linear quadratic control problem for the terminal controller is to find the control history $\boldsymbol{u}_C(t)$ to minimize the performance index,

$$J = x^T(t_f)\,S_f\,x(t_f) + \int_{t_0}^{t_f} \left[x^T(t)\,\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}_{\boldsymbol{c}}^{\boldsymbol{T}}(t)\,\boldsymbol{R}\,\boldsymbol{u}_{\boldsymbol{c}}(t) \right] dt \tag{36}$$

subject to the linear system dynamics, given by

$$\dot{x} = A(t)\,x + B(t)\,\boldsymbol{u} \tag{37}$$

The objective of the profile tracking algorithm is to estimate the changes in the reference control history to minimize the tracking error with respect to reference trajectory and minimize the control effort [7].

Even though the reentry system is highly nonlinear, the state deviation from the reference trajectory profile, $\delta\boldsymbol{x}$ can be assumed as linear. Hence, for trajectory control purpose, the problem can be redefined as to "estimate the vector of control deviation $\delta\boldsymbol{u}$ (with respect to reference control) to minimize the performances index", J as,

$$J = \delta x^T(t_f)\,S_f\,\delta x(t_f) + \int_{t_0}^{t_f} \left[\delta x^T(t)\,\boldsymbol{Q}\,\delta\boldsymbol{x}(t) + \delta\boldsymbol{u}_{\boldsymbol{c}}^{\boldsymbol{T}}(t)\,\boldsymbol{R}\,\boldsymbol{u}_{\boldsymbol{c}}(t) \right] dt \tag{38}$$

where

$\delta\boldsymbol{x}$ is the vector of state deviations (from the reference trajectory profile)

$\delta\boldsymbol{u}_{\boldsymbol{c}}$ is the vector of control deviations (from the reference control history)

The state deviations, $\delta\boldsymbol{x}$ are given by the linear system dynamics as

$$\delta\dot{x} = A(t)\,\delta x + B(t)\,\boldsymbol{u}_{\boldsymbol{c}} \tag{39}$$

where $A(t)$ and $B(t)$ matrices are obtained by linearizing the reentry nonlinear equations of motion about the reference trajectory and control. The study brings out the fact that tracking all six parameters of state vector, with a modified control law, eliminates the need for instantaneous bank reversal. In view of the fact that the angle of attack and bank angle profiles, generated by trajectory planning algorithm, are also the reference control profiles, the changes on 'α' and 'σ' are used as the control variables. To ensure better guidance accuracy, the detailed three-dimensional model used in trajectory planning algorithm should be used in profile tracking phase also. Therefore the general three-dimensional linear model is derived by linearizing the nonlinear model given in Eqns. (1-4) with respect to the reference profile. Theoretically, a linear regulator control law for the reference trajectory must be based on $A(t)$ and $B(t)$ as functions of time. However, since these functions vary slowly with time, $A(t)$ and $B(t)$ are approximated at discrete times t_k by constant matrices $A(t_k)$ and $B(t_k)$. $A(t_k)$ and $B(t_k)$ can be treated as locally time invariant and hence, at each time, t_k, the linear control law takes the form

$$\delta \boldsymbol{u_c} = -\boldsymbol{K}(t_k)\,\delta x \tag{40}$$

where, the feedback gains $\boldsymbol{K}(t_k)$ are constant. These gains are obtained by minimizing an infinite-horizon quadratic performance criterion

$$J = \int_{t_0}^{\infty} \left[\delta x^T(t)\, \boldsymbol{Q}\, \delta \boldsymbol{x}(t) + \delta \boldsymbol{u_c}^T(t)\, \boldsymbol{R}\, \delta\, \boldsymbol{u_c}(t) \right] dt \tag{41}$$

and weighing matrices $\boldsymbol{Q}$ (positive semi definite) and $\boldsymbol{R}$ (positive definite) can be selected to balance the conflicting demands of good tracking performance against minimization of the control effort, in addition to emphasizing the variable which is to be tracked. The optimum gain is given by

$$\boldsymbol{K}(t_k) = \boldsymbol{R}^{-1} \boldsymbol{B}^T(t_k)\, \boldsymbol{P}(t_k) \tag{42}$$

where the constant matrix $\boldsymbol{P}(t_k)$ is the solution of the algebraic Riccati equation,

$$\boldsymbol{PA}^T + \boldsymbol{A}^T\boldsymbol{P} - \boldsymbol{PBR}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = 0 \tag{43}$$

Control Law Design

The main parameters for LQR design are the weighing matrix elements for $\boldsymbol{Q}$ and $\boldsymbol{R}$ and the gain matrix $\boldsymbol{K}$. In the design, trade-offs must be made between control effort and tracking performance. In the present study, the initial guess of weights is made using the following rule of thumb, which states that the mean-square values of the individual terms in the performance index should approximate the same order of magnitude. Thus, a simple approach is to define Q and R as

$$\boldsymbol{Q} = diag\ (q_1, q_2, \ldots q_i) > 0\,; \quad \boldsymbol{R} = diag\ (r_1, r_2, \ldots r_r) > 0 \tag{44}$$

and to use an initial guess of q_i and r_r ("Bryson Rule" [7,15]) as

$$q_i = \left[(\delta\, x_i)_{\max} \right]^{-2}\ ; \quad r_r = \left[(\delta\, u_r)_{\max} \right]^{-2} \tag{45}$$

$(\delta\, x_i)_{\max}$ is the maximum allowable value of tracking error in the ith state $\boldsymbol{X}$ and

$(\delta\, u_r)_{\max}$ is the maximum allowable deviation in rth control parameter.

In general, the gain matrix computation can be carried out online for the reference trajectory profile, generated onboard, by solving algebraic Riccati equation, which in turn computes the gains [15]. Alternatively, it is seen that, consistent with observations of Ref.[7] and Ref. [12], a set of gain tables designed for a trajectory work satisfactorily for widely varying reentry trajectories. For a reference trajectory designed on ground, the matrices $A(t_k)$ and $B(t_k)$ are generated. The gain matrix $\boldsymbol{K}(t_k)$ is obtained offline via LQR function of the software MATLAB©, which computes the optimal gain matrix $\boldsymbol{K}(t_k)$ for a set of given $\boldsymbol{A}(t_k)$ and $\boldsymbol{B}(t_k)$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices. The gain matrices $\boldsymbol{K}(t_k)$ are stored onboard and the same gain matrices, which are scheduled as a function of the energy, are used for all the planning and dispersion cases. Even though the control law given by Eqn. (40) gives consistent performance for widely varying reentry trajectories as given above, it is seen that for the case of parameter dispersions in terms of drag, lift, atmospheric parameters and mass, the deviation on the TAEM location with respect to the planned parameters in terms of latitude and longitude is about 1° on latitude, which also increases monotonically. In order to reduce this steady state error, an integral term in the control law is proposed as a more appropriate choice. After detailed studies, it is seen that introduction of an integral term in the angle of attack command with latitude error alone (equivalent to cross range errors as per simulated cases) brings the error close to zero. It is also seen that while the lower gain values bring the error to zero, higher gain values produce oscillations on command histories. The study brings out the fact that the integrator gain needs to be scheduled, also as a function of energy, to bring the error to zero. The general form of the proposed proportional-integral control law is :

$$\delta \boldsymbol{u_c} = -\boldsymbol{K}(e)\,\delta x - \boldsymbol{K_I}(e) \int \delta\phi \; d\,e \tag{46}$$

where, $\delta x = [\delta V \;\; \delta\gamma \;\; \delta\,\psi \;\; \delta r \;\; \delta\phi \;\; \delta\lambda]^T$ is the dispersion of the state vector with respect to the reference state vector and $\boldsymbol{K} \in R^{2x6}$ and $\boldsymbol{K_I} \in R^{2x1}$. The command steering angle are

$$\boldsymbol{u_c} = \boldsymbol{u_{ref}} + \delta \boldsymbol{u_c} \tag{47}$$

where, $\boldsymbol{u_{ref}}$ is the reference control vector.

Control Law

Assuming α_{ref} and σ_{ref} as the reference angle of attack and bank angle, $V_{ref}, \gamma_{ref}, \psi_{ref}, r_{ref}, \phi_{ref}, \lambda_{ref}$ as the reference trajectory parameters and $V, \gamma, \psi, r, \phi, \lambda$ as actual trajectory parameters, the guidance commands required to track the reference trajectory parameters are given as,

$$\begin{bmatrix} a_c \\ \sigma_c \end{bmatrix} = \begin{bmatrix} \alpha_{ref} \\ \sigma_{ref} \end{bmatrix} + \begin{bmatrix} \delta\,\alpha_c \\ \delta\,\sigma_c \end{bmatrix} \tag{48}$$

The changes in the guidance commands with respect to the reference values are given by

$$\begin{bmatrix} \delta a_c \\ \delta\,\sigma_c \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \end{bmatrix} \begin{bmatrix} V - V_{ref} \\ \gamma - \gamma_{ref} \\ \psi - \psi_{ref} \\ r - r_{ref} \\ \phi - \phi_{ref} \\ \lambda - \lambda_{ref} \end{bmatrix}$$

$$+ \begin{bmatrix} K_i \\ 0 \end{bmatrix} \int (\phi - \phi_{ref})\, dt \tag{49}$$

Generic Guidance Algorithm

The trajectory planning algorithm is a numerical iterative predictor-corrector method. Assuming nominal vehicle data and environmental parameters and starting with an initial guess for the control vector, the predictor numerically propagates the trajectory from reentry interface to the terminal altitude. The error and gradient vectors, along with the sensitivity matrix, are computed for the predicted trajectory and using this information, the corrector computes the optimum search direction, step length and updates the control vector. This procedure is iterated till the optimum solution is achieved. With the optimum control vector history available, the reference trajectory profiles are generated next, which ensure that the target conditions meet all the path constraints. These computations are carried out prior to reentry and the reference profiles are stored as functions of energy.

The gains required for LQR algorithm are also stored as functions of energy. During the reentry phase, the profile tracking algorithm computes the states and energy from the navigation data and the stored data is interpolated for further use. Profile tracking algorithm computes the changes in the control histories to track the reference trajectory. The flow diagram of the generic algorithm is given in Fig.5.

Performance of the Generic Reentry Guidance Law

In order to evaluate the performance of the proposed generic integrated guidance algorithm, detailed simulation studies are carried out for a reentry mission of achieving the TAEM interface location without violating heat rate constraint (problem defined for the model development). The performance measures for the trajectory planning algorithm are :

- faster convergence
- improved accuracy of achieving the target and path constraints and
- robustness and mission flexible capability.

The performance measures for the profile tracking algorithm are :

- accuracy of achieving the target
- robustness under dispersed flight environment.

For these simulation studies, the generic guidance law is coded as a separate module and integrated with the simulator developed for evaluating RLV control and guidance

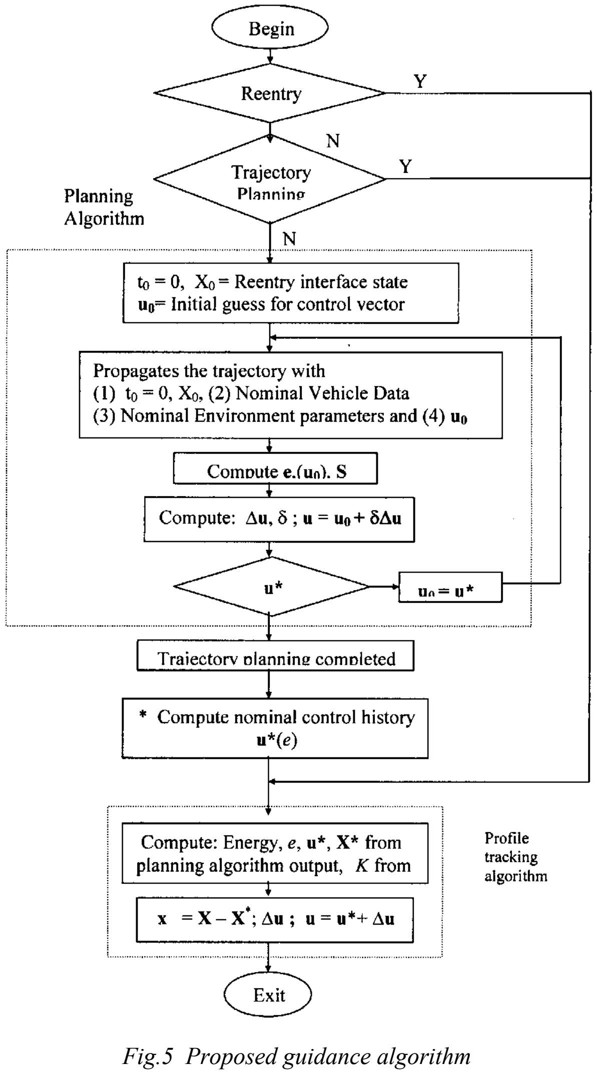
Begin
Reentry
Y
N
Trajectory Planning
Y
Planning Algorithm
N
t0 = 0, X0 = Reentry interface state
u0= Initial guess for control vector
Propagates the trajectory with
(1) t0 = 0, X0, (2) Nominal Vehicle Data
(3) Nominal Environment parameters and (4) u0
Compute e.(u0). S
Compute: Δu, δ ; u = u0 + δΔu
u*
u0 = u*
Trajectory planning completed
* Compute nominal control history u*(e)
Compute: Energy, e, u*, X* from planning algorithm output, K from
x = X – X*; Δu ; u = u*+ Δu
Profile tracking algorithm
Exit

Fig.5 Proposed guidance algorithm

performance [16]. Typical data available in literature [17] is used for the vehicle characteristics, whereas standard models are used for simulating environment. The initial conditions, target parameters, constraint limit and design parameters, used to evaluate the algorithm performance, are purely arbitrary and do not belong to any specific RLV mission.

Reentry Interface, Target and Heat Flux Limit

The simulations are carried out from the reentry interface to the target TAEM interface point. Typically reentry interface point assumed for the study is given by

$$t_0 = 0$$

$$V_0 = 7635.7\, m/s, \quad \gamma_0 = 2.1^o, \quad \psi_0 = 100^o,$$

$$h_0 = 121.809\,\text{km}, \quad \phi_0 = 0^o, \quad \lambda_0 = 0^o$$

Correspondingly, the target TAEM interface location is defined as

$$h_d = 25\, km, \quad \phi_d = -10^o; \quad \lambda_d = 64.7^o$$

and the desired value of bank angle at TAEM interface is assumed as zero. The limit on the heat rate is assumed as 70 W/cm^2, which is consistent with the current state-of-the-art in heat resistant materials and the thermal protection technology.

Design Parameters

The main design parameters required for the trajectory planning algorithm are: (1) Initial guess for the control vector (2) Weighting matrix for the target error and (3) Algorithm termination criteria. Similarly, the important design parameters for profile tracking algorithm are the elements of the weighing matrix, which are required for the control gains generation for profile tracking. The design parameters used are given below :

The fixed α profile defined by $\alpha_n = 45°$, $\dot{\alpha} = -0.0767^o/s$ and $t_a = 1030\ s$ is used in the trajectory planning algorithm. The initial guess for σ profile is given by $\sigma_d = 82°$ and $t_1 = 240s$, $t_2 = 500s$, $t_3 = 650s$, $t_4 = 800s$, $t_5 = 950s$, $t_6 = 1200s$

$$\dot{\sigma}_1 = -0.01^o/s, \ \dot{\sigma}_2 = -0.03^o/s, \ \dot{\sigma}_3 = -0.01^o/s,$$
$$\dot{\sigma}_4 = -0.01^o/s, \ \dot{\sigma}_5 = -0.01^o/s, \ \dot{\sigma}_6 = -0.01^o/s$$

The initial guess as given above is used for all trajectory planning cases. The time instants are purely arbitrary. For shorter reentry trajectory with duration less than 1200s, the time instants are modified. However, for on-board implementation, these time instants are to be decided in real time. In order to de-link the break points of the steering profile from the duration of reentry flight phase, it is necessary that these instants be defined in terms of energy. The weighing matrix elements are selected such that the maximum allowable dispersion on target latitude and longitude is 0.01^o and on the final bank angle is 0.1^o. Accordingly the weighing matrix elements are given as

$$H = \begin{bmatrix} 36000000 & 0 & 0 \\ 0 & 36000000 & 0 \\ 0 & 0 & 360000 \end{bmatrix}$$

The planning algorithm execution is terminated under one of the following conditions :

i) Number of iterations ≥ 10

ii) $e_t < 1$ (corresponding to 0.03^o tolerance) and $e_c \leq 100$.

iii) Optimum step length computed by quadratic interpolation algorithm $\delta \leq 1.0 \times 10^{-04}$. This criterion indicates that the solution achieved is very close to the previous iteration value.

The weighing matrix elements for LQR design are selected based on the trade-off between profile tracking performance and control effort, while attempting close tracking of all the six trajectory parameters. The Q and R matrix elements used for this study are given as :

$$H = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3600 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3600 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 360000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 360000 \end{bmatrix};$$

$$R = \begin{bmatrix} 600 & 0 \\ 0 & 100 \end{bmatrix}$$

In the present study, the gains are computed off-line through LQR function of MATLAB© and stored onboard as functions of energy. For a typical optimum reference trajectory, the linearized dynamics is generated at different flight instants and, along with the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices as defined above, is fed into the MATLAB© LQR function to generate optimum gain matrix, $\boldsymbol{K}$. Similarly, the integrator gain matrix $\boldsymbol{K_I}$ elements are computed through offline simulations. Gains $\boldsymbol{K}$ and $\boldsymbol{K_I}$ are scheduled as functions of energy and the same gain tables are used for all the performance simulations including different trajectory planning cases.

Performance Results for Planning Algorithm

In order to evaluate the performance of the integrated guidance algorithm for the widely varied reentry trajectories and to demonstrate its applicability in practical situations, simulations are carried out from different reentry interface locations to achieve the specified target without violating the allowable heat rate value. These conditions are achieved by varying the reentry interface latitude values from -15^o to $+20^o$, while other parameters are assumed same as those given in Table-1. To establish the robustness of the algorithm, for each reentry trajectory defined in Table-1, dispersions on aerodynamics, vehicle and environmental characteristics as defined in Table-2

Table-1 : Reentry interface conditions and targets

Parameters	Case-1	Case-2	Case-3	Case-4	Case-5
Reentry Interface					
h (km)	121.8	121.8	121.8	121.8	121.8
V (m/s)	7635.7	7635.7	7635.7	7635.7	7635.7
γ (deg)	-2.1	-2.1	-2.1	-2.1	-2.1
Az (deg)	100	100	100	100	100
ϕ (deg)	20	15	0	-10	-15
λ (deg)	0	0	0	0	0
Target Aimed					
ϕ_d (deg)	-16	-16	-16	-16	-16
λ_d (deg)	64.7	64.7	64.7	64.7	64.7
Constraint Imposed					
$\dot{Q}$ (W/cm^2)	≤ 70	≤ 70	≤ 70	≤ 70	≤ 70

are simulated. The performance of trajectory planning algorithm in terms of solution convergence for a atypical case is given in Fig.6. It is seen from the figure that the trajectory planning algorithm drives both target and constraint errors to zero within 6 iterations. Similar observations are seen for other cases also. For all the reentry trajectories, the predefined angle of attack profile is used, whereas bank angle profiles generated by the trajectory planning algorithm for different cases are given in Fig.7. It is seen that, in order to meet the specified target from widely different reentry interface conditions, the bank angles are altered significantly. Also, it is seen that the algorithm drives the final bank angle to zero for all the cases and that while the heat rate constraint is active, there is reduction in bank profile as expected. It is further observed that for cases-3 to 5, the bank angle varies slowly from positive side to negative side and back to zero. From the trajectory profile generated by the planning algorithm for different cases, as given in Fig.8, it is seen that the algorithm steers the vehicle from different reentry interface points to the specified TAEM interface point. The target conditions, aimed and achieved, show that the accurate solution is achieved for all the reentry interface conditions. The heat rate profiles given in Fig.9 bring out successful constraint satisfaction. These results also indicate that the algorithm steers the vehicle to the specified TAEM location at -16^0 latitude from the widely varying reentry latitude varying from +20^0 to -15^0. The algorithm produces the same results for the left cross range as well and in these cases the steering profiles are negative of the right cross range cases. Also, in the present study, the

Table-2 : Simulated dispersion environments

Sl. No.	Parameters	Dispersion Levels
1	C_D	+10%
2	C_D	-10%
3	C_L	+10%
4	C_L	-10%
5	m	+5%
6	m	-5%
7	ρ	+25%
8	ρ	-25%

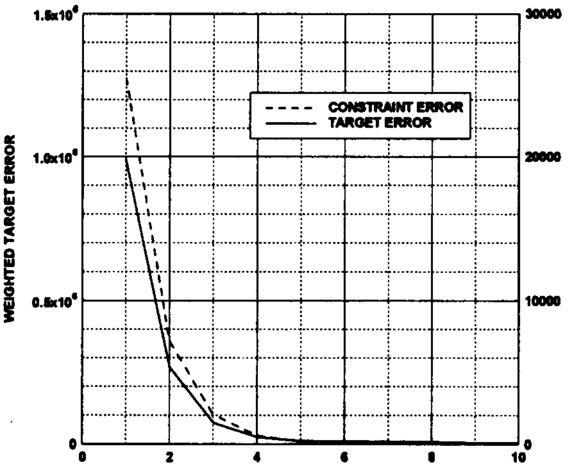

Fig.6 Error convergence for Case-2

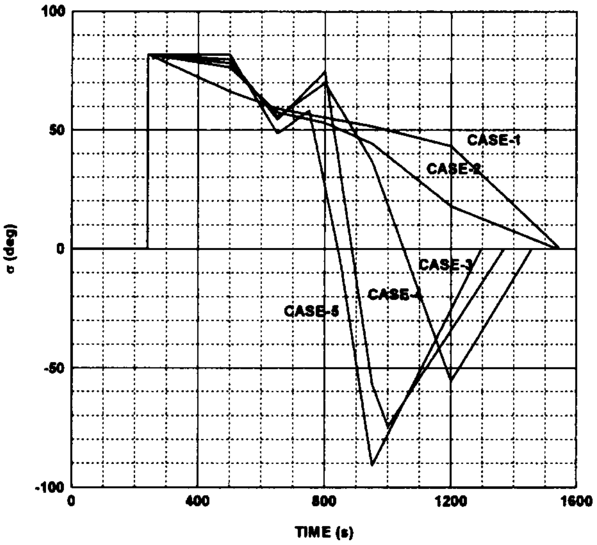

Fig.7 Bank angle profile generated by trajectory planning algorithm for different reentry trajectories

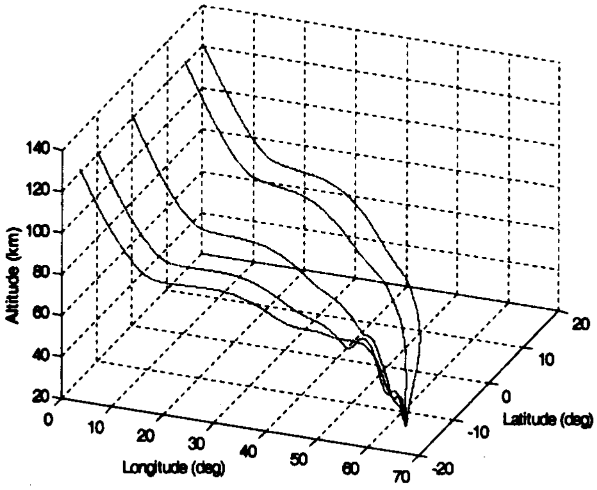

Fig.8 Trajectory profiles for different reentry trajectories generated by trajectory planning algorithm

steering rate initiation times are manually changed to produce faster convergence for the shorter range cases.

In order to decide these time instants online and unambiguously, the break points can be implemented as functions of energy. Also, results indicate that with more numbers of break points, the bank angle profiles are smoother and the solution converges very fast with only a marginal increase in the computational load during each iteration.

Performance Results of Profile Tracking Algorithm

In order to evaluate the trajectory control algorithm performance, two widely varying reentry trajectory cases are analysed for nominal performance and results for the planned and actual control profiles, altitude profiles, ground trace profiles and heat rate profiles are given in Figs. 10-17. The achieved targets for these cases are given in Table-3. The corresponding results for case-3 reentry trajectory with different dispersions, simulated as given in Table-2, are given in Figs.18-23. It is seen that the reference control profiles, from the planning algorithm, are modulated under dispersed environments such that the reference trajectory profiles are tracked. In view of this, the major changes in the control profiles are seen in the initial phase itself, which brings the reentry trajectory close to the nominal one. This fact is manifest in the profile tracking in that, towards the end of the trajectory, the changes in the control histories are small. The maximum

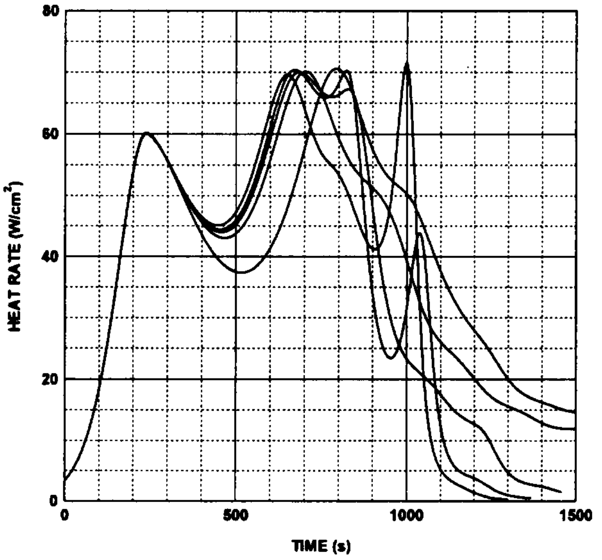

Fig.9 Heat rate profiles for different reentry trajectories

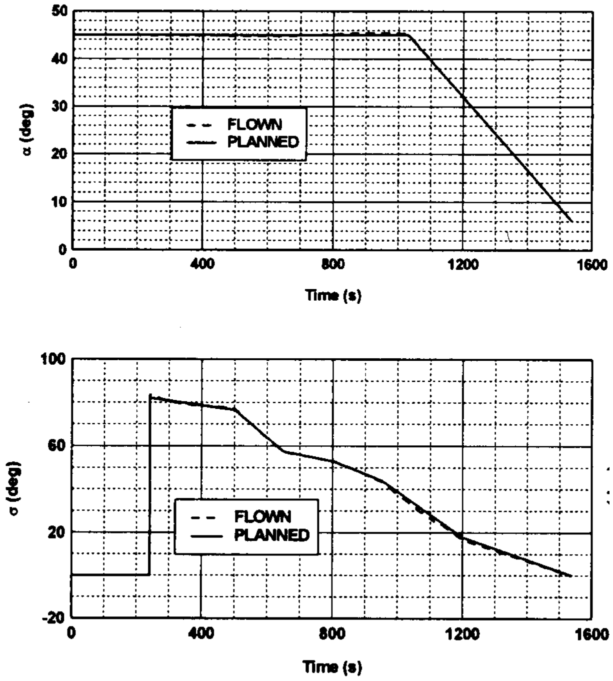

Fig.10 Control history profiles for case-2

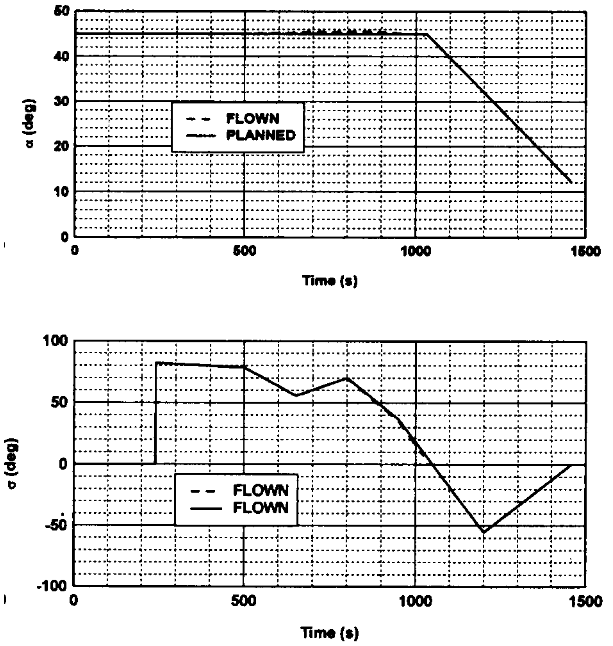

Fig.11 Control history profiles for case-3

observed trajectory parameter dispersions at TAEM interface and the maximum heat rate for these cases are also given in Table-4.

Conclusions

In this paper, a reentry guidance law is presented, which is capable of steering the vehicle from any reentry interface point to the specified target without an initial feasible reference trajectory. The proposed guidance scheme consists of (1) an off-line onboard trajectory planning algorithm and (2) a trajectory control algorithm based on LQR. The planning algorithm solves the three-dimensional problem as a targeting problem without a need for instantaneous bank reversal algorithm. The sensitivity

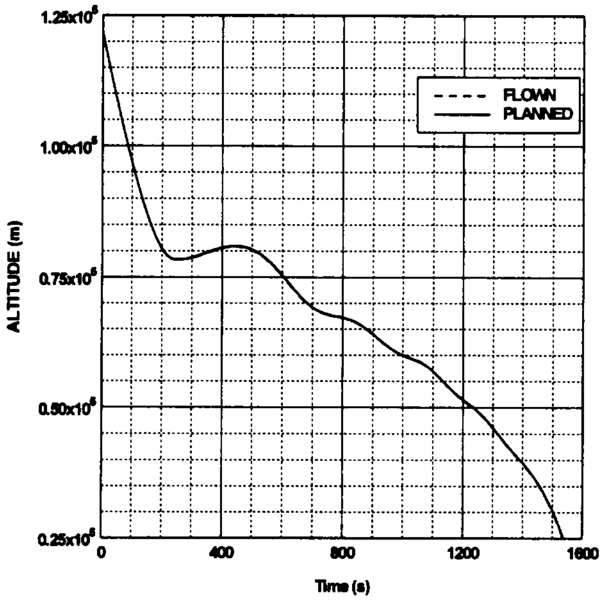

Fig.12 Altitude profiles for case-2

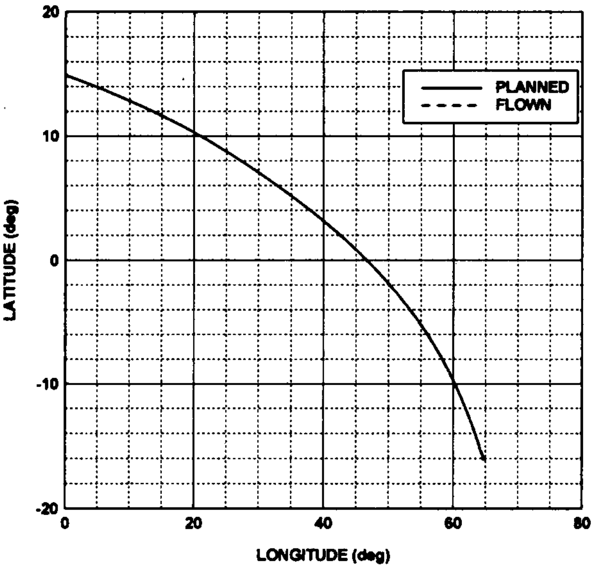

Fig.14 Ground trace for case-2

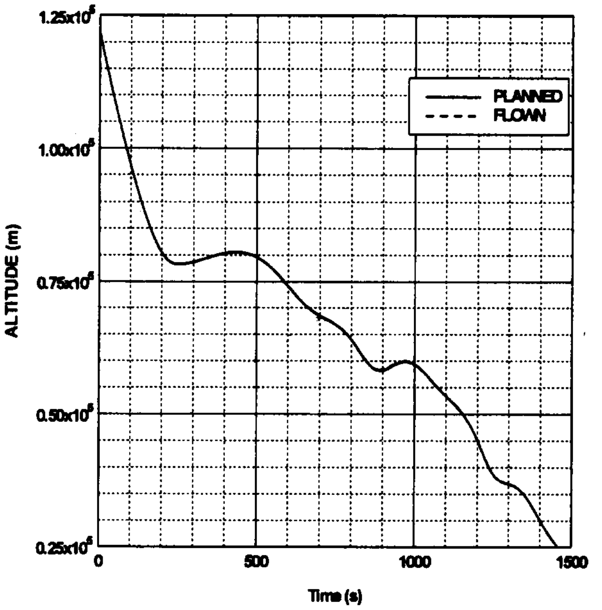

Fig.13 Altitude profiles for case-3

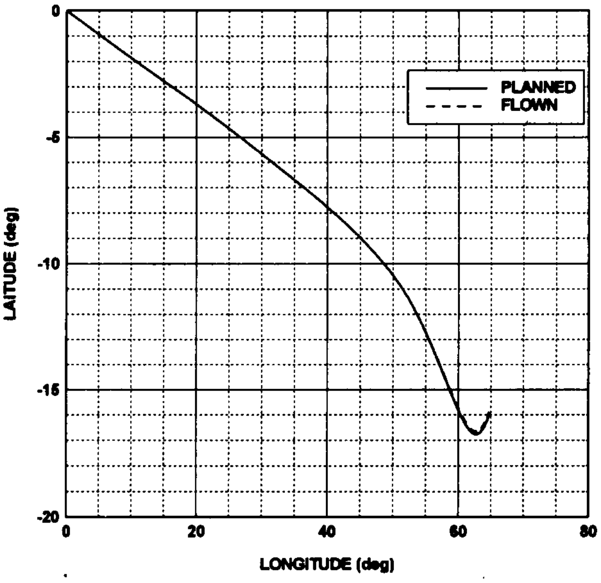

Fig.15 Ground trace for case-3

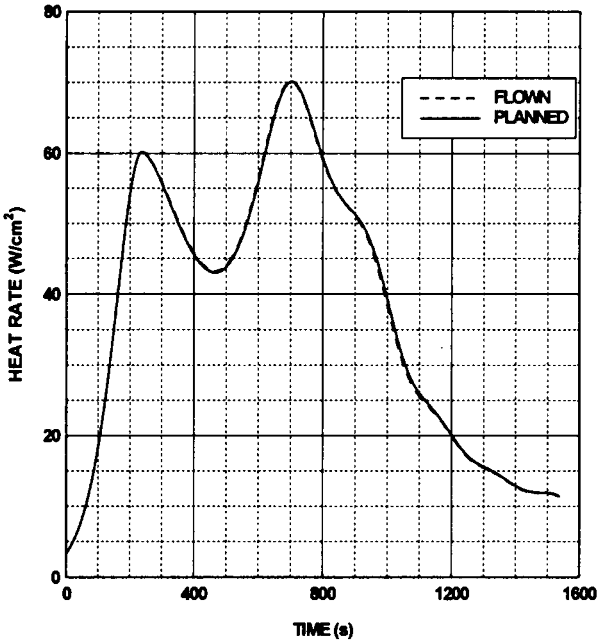

Fig.16 Heat rate profiles for case-2

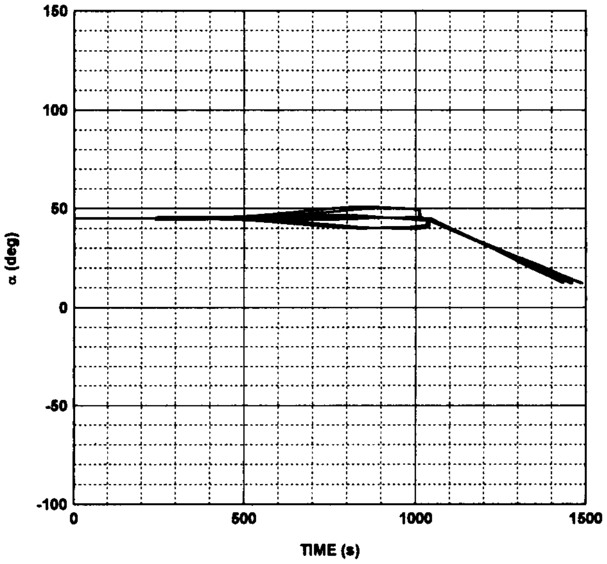

Fig.18 Angle of attack profiles for dispersed flight conditions (case-3)

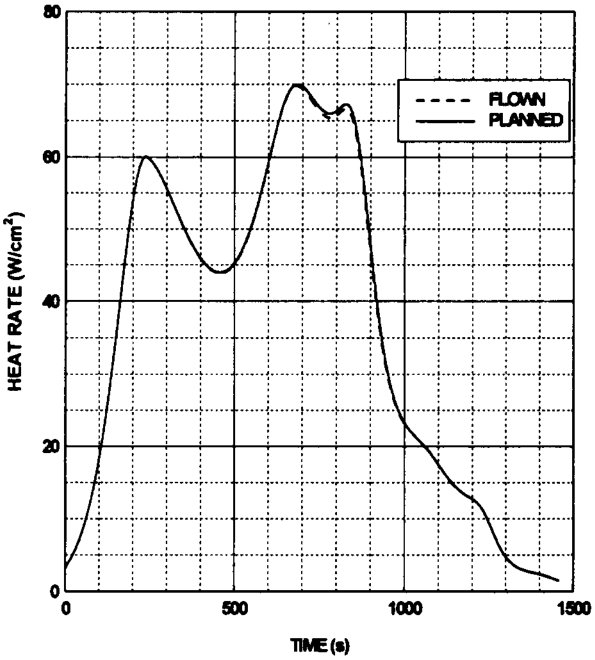

Fig.17 Heat rate profiles for case-3

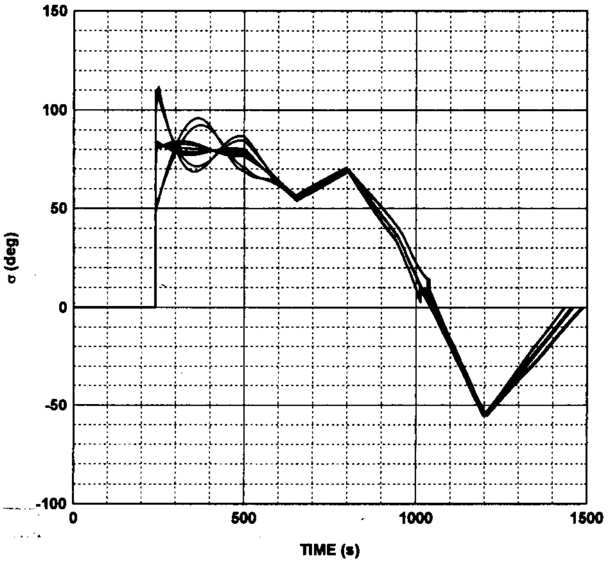

Fig.19 Bank angle profiles for dispersed flight conditions (case-3)

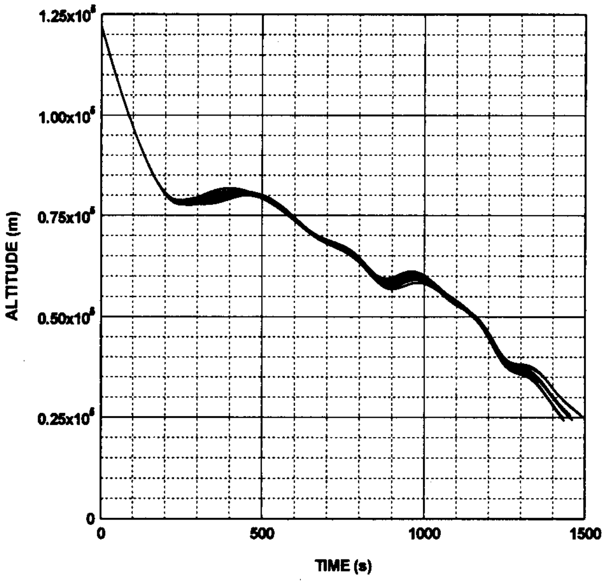

Fig.20 Altitude profiles for dispersed flight conditions (case-3)

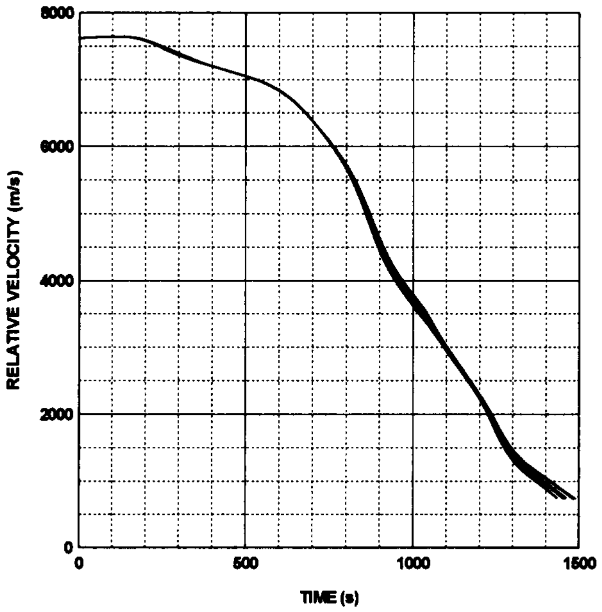

Fig.22 Relative velocity profiles for dispersed flight conditions (case-3)

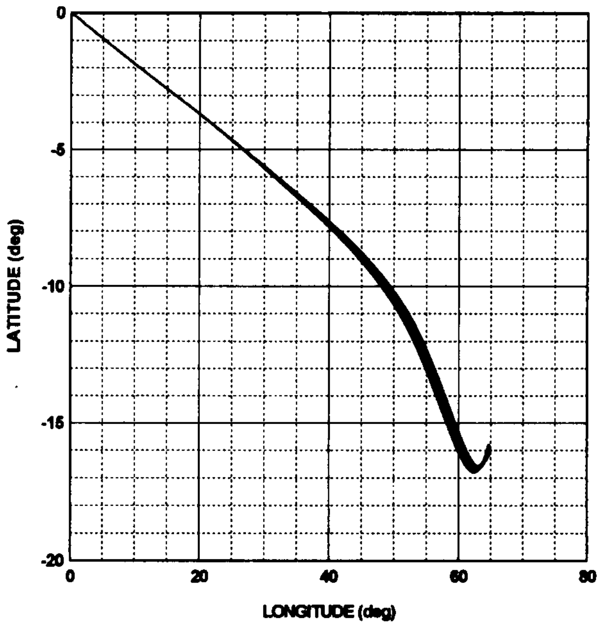

Fig.21 Ground trace profiles for dispersed flight conditions (case-3)

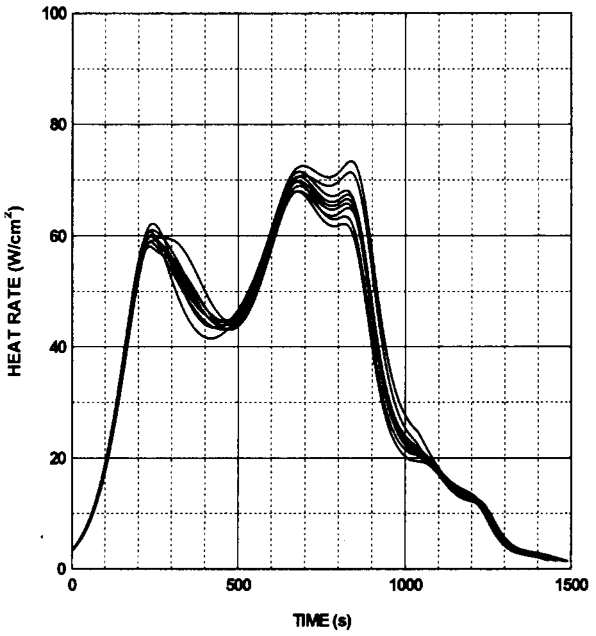

Fig.23 Heat rate profiles for dispersed flight conditions (case-3)

Table-3 : Tracking performance for different reentry trajectories

Parameters	Aimed	Achieved	
		Case-2	Case-3
Target : h_d (km)	25.0	25.05	25.0
ϕ_d (deg)	-16	-15.9	-15.94
λ_d (deg)	64.7	64.7	64.7
Constraint : $\dot{Q}$ (W/cm^2)	≤ 70	70.38	69.8

Table-4 : Maximum dispersions observed at TAEM interface

Parameters	Dispersions
Altitude	± 0.8 km
Latitude/Longitude	± 0.1 deg
Velocity	± 4 m/s
Flight path angle	± 1 deg
Flight Azimuth	± 8 deg

matrix coefficients, required for the planning algorithm, are derived as additional states and are computed through numerical Integration, which avoids the problems related to numerical computation of the gradients through finite differencing. The improved profile tracking control law contains an integrator for improving tracking accuracy. The guidance law is then applied to a typical RLV mission with the objective of steering the vehicle from any reentry interface point to the target of a specified TAEM interface location. The performance of proposed algorithm is demonstrated for widely varying reentry interface points and dispersions on vehicle and environmental parameters. Robustness of the algorithm is established by steering the vehicle only through bank angle modulation from widely separated reentry interface points. Further, a single gain profile, scheduled as function of vehicle energy, produces satisfactory performance for a wide variety of planning profiles generated onboard.

Acknowledgement

The studies reported in this paper have been carried out with the infrastructural support from the Centre for Aerospace Systems Design and Engineering (CASDE), Department of Aerospace Engineering, Indian Institute of Technology Bombay, Mumbai and the Indian Space Research Organisation (ISRO), Vikram Sarabhai Space Centre, Thiruvananthapuram and the same is gratefully acknowledged.

References

1. Harpold, J.C. and Graves, C.A., "Shuttle Entry Guidance, Journal of Astronautical Sciences", Vol.27, No.3, Jul-Sep, 1979, pp.239-268.

2. Ishimoto, S., Takizawa, M., Suzuki, H. and Morito, T., "Flight Control System of Hypersonic Flight Experiment Vehicle", AIAA Paper No. AIAA-96-3403, AIAA, Atmospheric Flight Mechanics Conference, July 1996.

3. Mease, K.D., Chen, D.T., Tandon, S., Young, D.H. and Kim, S., "A Three-dimensional Predictive Entry Guidance Approach", AIAA Paper No. AIAA 2000-3959, AIAA Guidance, Navigation and Control Conference and Exhibit, August 2000.

4. Roenneke, A., "Adaptive Onboard Guidance for Entry Vehicles", AIAA Paper No. AIAA 2001-4048, AIAA Guidance, Navigation and Control Conference, August 2001.

5. Ishizuka, K., Shimura, K. and Ishimoto, S., "A Reentry Guidance Law Employing Simple Real Time Integration", AIAA Paper No. AIAA 98-4329, AIAA, Reston, Va, August 1998.

6. Fuhry, D.P., "Adaptive Atmospheric Reentry Guidance for the Kistler K-1 Orbital Vehicle", AIAA Paper No. AIAA-92-4211, Proceedings of the AIAA Guidance, Navigation and Control Conference, Vol.2, AIAA, Reston, Va, 1999, pp.1275-1288.

7. Dukeman, G.A., "Profile Following Entry Guidance Using Linear Quadratic Regulator Theory", AIAA Paper No. AIAA 2002-4457, AIAA Guidance, Navigation and Control Conference and Exhibit, August 2002.

8. Zimmerman, C., Dukeman, G. and Hanson, J., "An Automated Method to Compute Orbital Reentry Trajectories with Heating Constraints", AIAA Paper No. AIAA 2002-4454, AIAA Guidance, Navigation and Control Conference and Exhibit, August 2002.

9. Saraf, A., Leavitt, J.A., Chen, D.T. and Mease, K.D., "Design and Evaluation of an Acceleration Guidance Algorithm for Entry", AIAA Paper No. AIAA 2003-5737, AIAA Guidance, Navigation and Control Conference and Exhibit, August 2003.

10. Shen, Z. and Lu, P., "Onboard Generation of Three-dimensional Constrained Entry Trajectories", Journal of Guidance, Control and Dynamics, Vol.26, No.1, Jan-Feb 2003, pp.111-121.

11. Shen, Z. and Lu, P., "Onboard Entry Trajectory Planning Extended to Sub-Orbital Flight", AIAA Paper No. AIAA-2003-5736, AIAA Guidance, Navigation and Control Conference and Exhibit, August 2003.

12. Shen, Z. and Lu, P., "Dynamic Lateral Entry Guidance Logic", Journal of Guidance, Control and Dynamics, Vol.27, No. 6, Nov-Dec, 2004, pp.949-959.

13. Joshi, A. and Sivan, K., "Reentry Guidance for Generic RLV Using Optimal Perturbations and Error Weights", AIAA Paper No. AIAA-2005-6438, AIAA Guidance, Navigation, and Control Conference, August 2005.

14. Regan, F.J. and Anandakrishnan, M., "Dynamics of Atmospheric Re-entry", AIAA Education Series, AIAA, Washington D.C., ISBN 1-56347-048-9, 1993.

15. Bryson, A.E. and Ho, Y.E., "Applied Optimal Control", Hemisphere Publishing Corporation, Washington D.C, 1975.

16. Joshi, A. and Sivan, K., "Modeling and Open Loop Simulation of Reentry Trajectory for RLV Applications", Proceedings of the 4th International Symposium on Atmospheric Reentry Vehicle and Systems, Arcachon, France, March 21-23, 2005.

17. Eussel, W.R (Ed)., "Aerodynamic Design Data Book, Vol.1, Orbiter Vehicle SIS-1", Report No. SDT2-SH-00600, Space Systems Group, Rockwell, Space Systems Group, Rockwell International, California, 1980.