Control Design for Landing HS9 Booster of India's Reusable Launch Vehicle

Aditya Prakash (190065)[1], Azhar Tanweer (190223)[1], Bidhan Arya (190238)[1], Rahul Godara (190663)[1], Fahad (190799)[1]

Abstract

Reusable Launch Vehicle - Technology Demonstrator (RLV - TD) is one of the most technologically challenging endeavours of ISRO towards developing essential technologies for a fully reusable launch vehicle to enable low cost access to space. The entire RLV consists of a booster and RLV-TD. Although, the current plan of ISRO is to restore the RLV-TD, but it is still dropping HS9 Booster into Bay of Bengal. The paper aims at developing control algorithm for vertically landing the HS9 Booster safely on surface. Several assumptions has been made to simplify the problem. The paper only focuses on the control aspect of the problem and does not include thermodynamic analysis.

1. Introduction

Reducing the specific cost of payload delivery into orbit and removing astronauts and essential cargo from orbit to the earth is a critical operational task of modern astronautics development. The success of such a solution will determine the future of space exploration and the relocation of some industries into space. However, the high expense of getting into space severely hampers space research and use. The consensus is that a reusable launch vehicle is the best way to obtain low-cost, dependable, and on-demand space access. Moving from disposable components and structures of the carrier and spacecraft to reusable ones reduces the cost of space launches. In this direction, ISRO has been trying to develop reusable launch vehicles under the RLV Technology Demonstration Programme. This project intends to cut the cost of delivering payloads to low Earth orbit by nearly 80%. The RLV-TD (Reusable Launch Spacecraft-Technology Demonstration Program) is a program of technology demonstration flights to demonstrate the feasibility of a completely reusable vehicle with two stages to orbit (TSTO). A winged reusable launch vehicle technology demonstrator (RLV-TD) evaluated hypersonic flight, autonomous landing, powered cruise, and hypersonic flight using air-breathing propulsion.

ISRO planned four RLV-TD test flights: HEX (Hypersonic Flight Experiment), LEX (Landing Experiment), REX (Return Flight Experiment), and SPEX (Secondary Flight Experiment) (Scramjet

Propulsion Experiment). Out of these, ISRO has tested the first HEX flight. The RLV-TD was launched from a single-stage solid-fuel booster mounted on top of it (the HS9 booster).

This paper focuses on the control aspect of the RLV to provide it with autonomy. Since the vehicle's dynamics are highly interactive and the vehicle is affected by a multitude of uncertainties and unknown external disturbances, designing a guidance and control system for the RLV, particularly in the reentry phase, is a difficult task. In addition to the complicated dynamics of the equations of motion for manoeuvring flight, path restrictions like heat rate, structural loads, and dynamic pressure limit the reentry flight corridor to a minimal area, making system design much more difficult. The primary purpose of reentry guidance is to generate guidance commands, such as the angle of attack and bank angle, that lead the vehicle from the entry beginning point into a neighbourhood around the landing site while staying within the path limitations. A robust attitude control system and an adaptive guidance system are essential for mission success. Reentry attitude control aims to create a controller that ensures the attitude angle follows the guidance commands despite uncertainties and unknown external disturbances. Even though numerous guidance and control technologies have been developed over the past few decades, the current RLV integrated guidance and control system is mainly based on the offline trajectory. The study's subject is the real-time reentry trajectory and attitude coordinate control. The focus of this study is to combine advanced trajectory optimization and control techniques into a single unified framework, as well as to develop an integrated guidance and control scheme that ensures that the RLV has adaptation capabilities in the presence of unexpected events, allowing the mission objectives to be achieved with minimal ground input.

2. Problem Formulation

The RLV (Reusable Launch Vehicle)[1] is launched from the First Launch Pad at Satish Dhawan Space Centre. The RLV consists of RLV-TD mounted on HS9 rocket booster. The current plan of ISRO is to restore the RLV-TD, but it is still dropping HS9 Booster into Bay of Bengal. [2] The paper aims at developing control algorithm for vertically landing the HS9 Booster safely on surface.

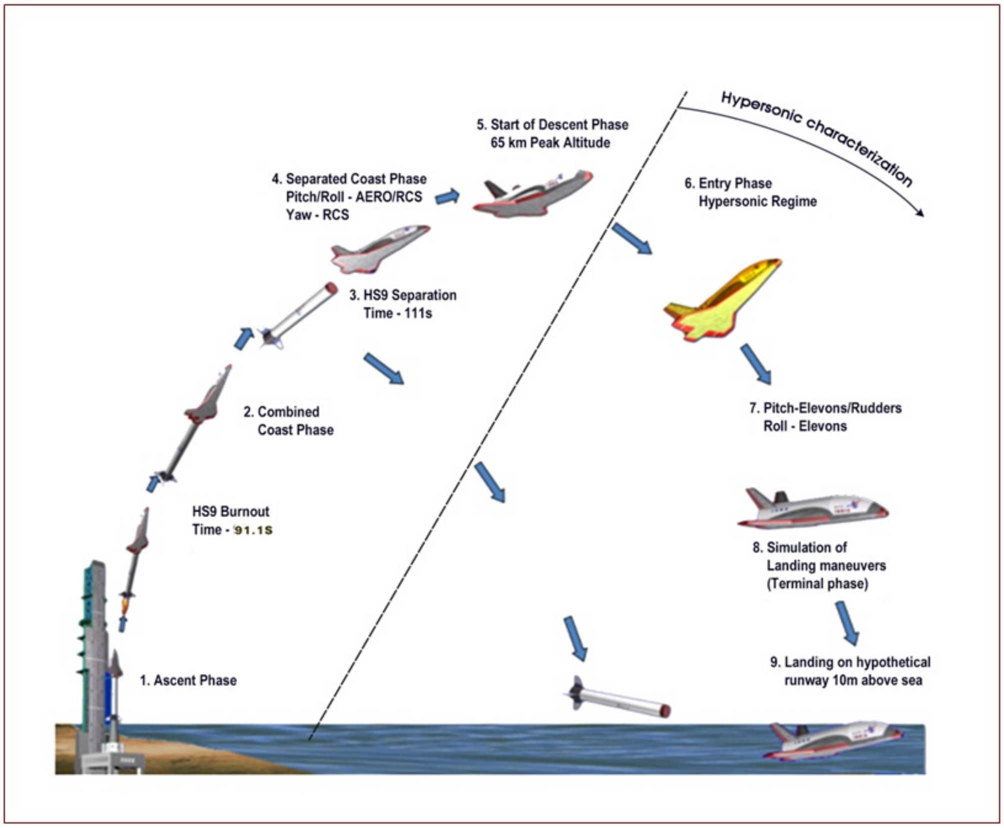


Figure 1: RLV-TD mounted on HS9 rocket flight, ©ISRO

2.1. Components of HS9 Booster

- **Structural Part:** Assume the booster to be of cylindrical shape with length (L) and diameter (d). Also, assume that the booster is symmetric about x, y, z axis (represented in fig. 2). Hence, the x,y,z axis represents the principal axis of the booster. Also the design of booster is such that the centre of mass coincides with the aerodynamic centre of the booster. This ensures that there is no moment on the booster due to aerodynamic forces. Also, for simulation purpose, it has been assumed that the CG (centre of gravity) lies at the half the middle point of the booster. This is not a strict assumption on the booster but just for numerical calculation purpose.

- **Liquid Rocket Engine:** Liquid Rocket Engine has been used in the booster to provide **thrust vector control** during landing. It is assumed that the fuel of the liquid rocket engine is optimised and the thrust provided by it is variable. The variable thrust is essential to controlling the velocity and the height of the booster.

- **Small Thrusters**: Fours small thrusters are used to control attitude of the booster. In the fig. 2, thrusters 2 and 3 are coupled to provide torque across positive z-axis and thrusters 1 and 4 are coupled to provide torque across negative z-axis. The thrusters provide constant thrust and thus constant torques. All thrusters are identical.

- **Control System:** Separate control system has been used for booster to implement control for the booster.

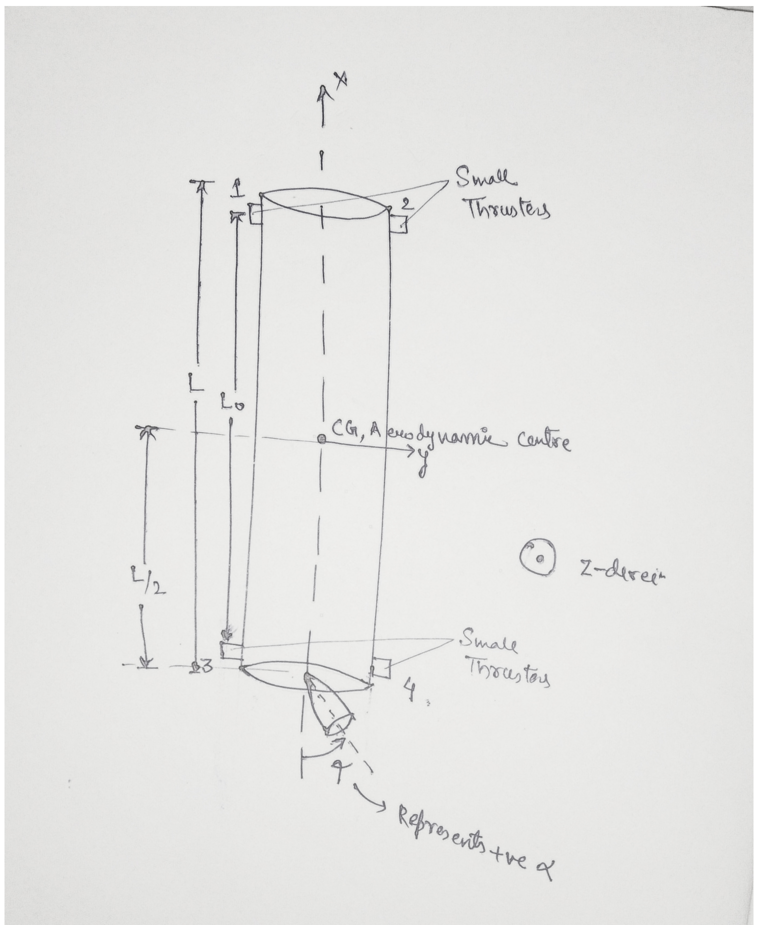


Figure 2: HS9 Booster Design

2.2. *Dynamics of Booster*

The motion of the booster is considered to be planar. Hence we will be dealing only with 2D motion. This is based on the assumption that all forces lies on the plane of the motion. Four coordinate system has been defined in fig. 3. The $\xi_1(\hat{u}_x, \hat{u}_y)$ represents the inertial frame attached with Earth. The $\xi_2(\hat{r}, \hat{\theta})$ represents the polar coordinate system centered at Earth's center. The $\xi_3(\hat{u}_t, \hat{u}_n)$ represents the coordinate system attached with velocity vector such that $\hat{u}_t$ is in the direction of velocity vector. Finally, the $\xi_4(\hat{x}, \hat{y})$ coordinate system is the body fixed frame of the booster attached to the booster as in fig. 2. The states required to represent the dynamics of the booster are shown clearly in fig. 3

The states required to represent the dynamics of the booster are:

- Flight Path Angle (γ)
- Orientation of the rocket (ϕ) such that $\phi + \gamma + \beta$ where β is the angle of attack.
- Distance of rocket from Earth's surface (h) such that $r = h + R_o$ where R_o is Earth's Radius
- Velocity of the rocket (v)
- Angular Velocity of the booster about it's COM (ω)

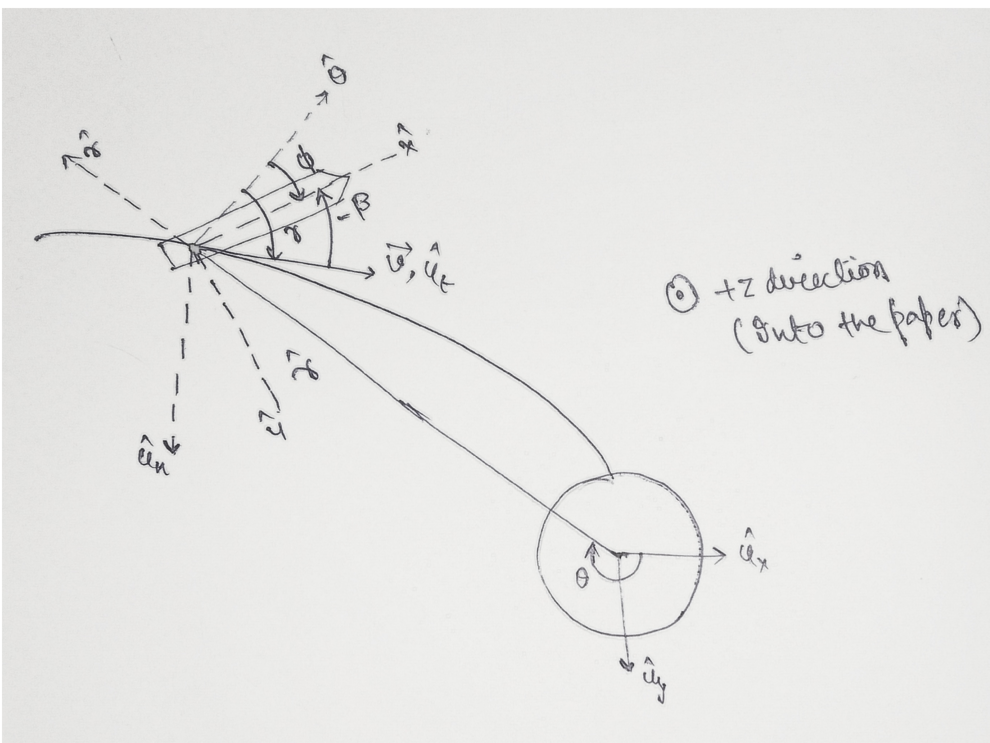


Figure 3: Coordinate System

2.2.1. Equations of Motion

In ξ_3, the equation of the booster is given by [3],

$$m(\dot{v} + \vec{w} * \vec{v}) = \vec{F}_* + m\vec{g} + \vec{D} + \vec{L} \tag{1}$$

We have,

$$\dot{\vec{v}} = \dot{v}\hat{u}_t + v\dot{r}\hat{u}_n$$

$$\vec{w} * \vec{v} = \frac{v^2}{2}\cos\gamma\hat{u}_n$$

$$\vec{F}_* = F_*\cos(\alpha - \beta)\hat{u}_t - F_*\sin(\alpha - \beta)\hat{u}_n$$

$$\vec{L} = -L\hat{u}_n$$

$$\vec{D} = -D\hat{u}_t$$

$$\vec{g} = g\cos\gamma\hat{u}_n + g\sin\gamma\hat{u}_t$$

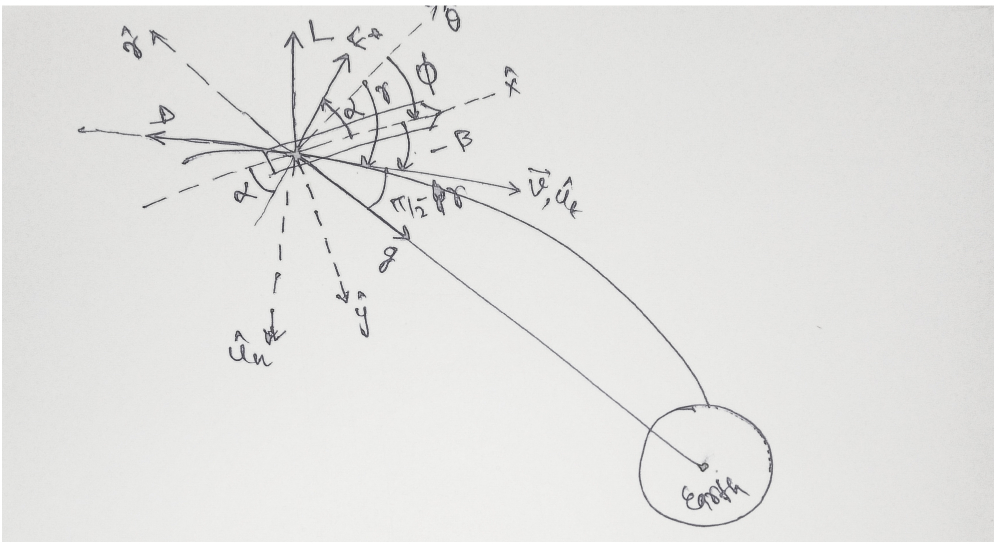


Figure 4: Booster Dynamics

Using all the above equations in Eq. (1), we get

$$\dot{v} = \frac{F_*\cos(\alpha - \phi + \gamma) - D}{m} + g\sin\gamma \tag{2}$$

$$\dot{\gamma} = \frac{-F_*\sin(\alpha - \phi + \gamma) - L}{mv} + \left(\frac{g}{v} - \frac{v}{r}\right)\cos\gamma \tag{3}$$

Also, we have,

$$\dot{h} = -v\sin\gamma \tag{4}$$

where

$$D = \frac{1}{2}\rho(r)v^2 C_D A_{\perp} = \frac{1}{2}v^2\rho_o(r)C_D\left(\frac{\pi dL}{2}\sin(\gamma - \phi)\right)e^{\frac{-h}{H}}$$

$$L = \frac{1}{2}\rho(r)v^2 C_D A_\perp = \frac{1}{2}v^2\rho_o(r)C_L\left(\frac{\pi dL}{2}\sin(\gamma-\phi)\right)e^{\frac{-h}{H}}$$

and for the height range, we assume g to be constant and equal to $g_o = 9.81$ ms^{-}2.

Using Moment balance, we get

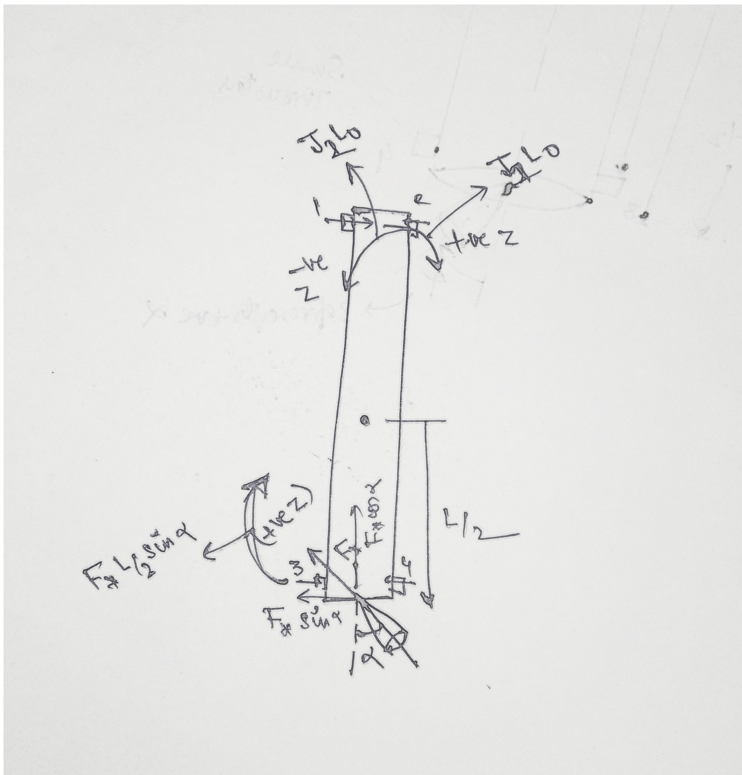


Figure 5: Booster Moment Calculation

$$\dot{\omega} = \frac{F_* L}{2I}\sin\alpha + \frac{J_1 L_o}{I} - \frac{J_2 L_o}{I} \tag{5}$$

where

$$\dot{\phi} = \omega \tag{6}$$

and J_1 is the thrust at thrusters 1 and 4 (couple) and J_2 is the thrust at thrusters 2 and 3 (couple), I is the moment of inertia of the booster about z-axis. Refer to fig. 2 for thrusters position and L_o and L and directional convention for α. As per the direction convention, we need positive α to produce positive torque about z-axis.

Equations (2), (3), (4), (5) and (6) are the state equations for the system with γ, ϕ, h, v and ω being the state and J_1, J_2, F_* and α being the control inputs.

2.3. Control Problem

For further analysis, we consider the following state variables: $\chi = [h,\, v,\, \gamma,\, \omega\ ,\, \phi]^T$. The control problem of the reentry has been divided into 4 phases:

- **Phase I**: In this phase, the booster rocket engine is off and the booster is turned from it's initial condition to a condition in which $\phi = -\pi$, $\omega = 0$ (i.e. booster is horizontal w.r.t. radial direction). This phase starts after the booster separation has occurred and then the booster reaches it's maximum height when $v = 0$. This is done so that booster is perpendicular to the velocity direction, the drag is maximum and hence velocity decreases rapidly.

- **Phase II**: In this phase, no control has been utilised. The booster falls horizontally while gravity, drag and lift works on it. However, for real life applications, stability control is needed to make the rocket stable.

- **Phase III**: In this phase, the booster is turned from present condition to a condition in which $\phi = -\frac{\pi}{2}$ and $\omega = 0$. The aim is to make the booster vertical. This is done for until the booster is vertical. The rocket engine is off in this case as well.

- **Phase IV**: In this phase, the booster rocket engine is turned on. And thus, the α control comes into picture. The aim is to make, $v = 0$, $h = 0$, $\phi = -\frac{\pi}{2}$, $\omega = 0$ and $\gamma = \frac{\pi}{2}$

2.4. Booster Parameters and Other constants

Mass of Booster:	85000 kg
Length of Booster:	45 m
Diameter of Booster:	10 m
C_D (Drag Coefficient):	1.22
C_L (Lift Coefficient):	0.45
I (Moment of Inertia):	1416700 Nm^2
Thrust of small thrusters:	219 N
u_m:	0.007 Nm
g_o:	9.81 ms^{-2}
R_o:	6371 km
ρ_o:	1.225 kgm^{-3}
H:	6.7 km

Table 1: Booster Parameters and Other constants

2.5. Initial Condition

The booster separation occurs at 65 km where the booster solid rocket engine has burned all it's fuel and the velocity of the RLV has reached zero. The booster is separated and currently is in the state of zero velocity, ready to free fall. This is the initial condition for the simulation purpose:

- Height (h): 65000 m
- Velocity (v): 0 m/s
- Flight Path Angle(γ): $\pi/2$
- Orientation of Rocket (ϕ): $\pi/6$
- Angular Rotation velocity of Booster (ω): 0

3. Phase I

The aim of the phase is to turn the rocket horizontal ($\phi = -\pi$), so that the rocket experiences maximum drag. In this phase I, the rocket engine is turned off. Therefore, $F_* = 0$. Also, in this case, it has been considered that the control input is only through small thrusters and they work in such a way that either pair 1-4 works or 2-3 works. In this way, considering that all the thrusters provide same thrust, the control input is given by

$$u = \begin{cases} \frac{J_1 L_o}{I} = u_m, & \text{when pair 1-4 is active} \\ -\frac{J_2 L_o}{I} = -u_m, & \text{when pair 2-3 is active} \end{cases} \tag{7}$$

This results in following state equations:

$$\dot{x} = f(x) = \begin{bmatrix} -v \sin\gamma \\ -\frac{1}{2} v^2 \frac{\rho_o}{m} C_D \left(\frac{\pi dL}{2} \sin(\gamma - \phi)\right) e^{\frac{-h}{H}} + g_o \sin\gamma \\ -\frac{1}{2} v \frac{\rho_o}{m} C_L \left(\frac{\pi dL}{2} \sin(\gamma - \phi)\right) e^{\frac{-h}{H}} + \left(\frac{g_o}{v} - \frac{v}{h+R_o}\right) \cos\gamma \\ u \\ \omega \end{bmatrix} \tag{8}$$

$$\dot{x} = f(x) = \begin{bmatrix} -v \sin\gamma \\ \kappa_D v^2 \sin(\gamma - \phi) e^{\frac{-h}{H}} + g_o \sin\gamma \\ \kappa_L v \sin(\gamma - \phi) e^{\frac{-h}{H}} + \left(\frac{g_o}{v} - \frac{v}{h+R_o}\right) \cos\gamma \\ u \\ \omega \end{bmatrix} \tag{9}$$

where $\kappa_D = -\frac{1}{2}\frac{\rho_o}{m} C_D \frac{\pi dL}{2}$ and $\kappa_L = -\frac{1}{2}\frac{\rho_o}{m} C_L \frac{\pi dL}{2}$ are constants.

On Linearizing the above equation about the reference trajectory, $\chi_r = [h_o,\ v_o,\ \pi/2,\ 0,\ \text{-}\pi]^T$, where h_o and v_o are optimal values, we get

$$\dot{x} = Ax + Bu \tag{10}$$

where $A = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ \frac{\kappa_D}{H} v_{o1}^2 e^{-\frac{h_{01}}{H}} & -2 v_{o1} \kappa_D e^{-\frac{h_{01}}{H}} & 0 & 0 & 0 \\ \frac{\kappa_L}{H} v_{o1} e^{-\frac{h_{01}}{H}} & -\kappa_L e^{-\frac{h_{01}}{H}} & -\left(\frac{g_o}{v_{o1} - \frac{v_{o1}}{h_{o1} + R_o}}\right) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ and $x = \chi - \chi_r$

As we can see from the matrix A that in linearized state the system is **not fully controllable**, only state variables ω and ϕ are controllable. Hence, we further proceed by applying Sliding Mode Control to control ω and ϕ (turn the booster) and then solve the values of h, v and γ analytically or numerically to get values at each time step. For simulation purpose, Runge-Kutta Method has been used.

3.1. SMC Controller Design

Now we have the $x = [\phi,\, \omega]^T$, then the state space equation of the booster orientation is given by $\dot{x} = Ax + Bu$ where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Assume the initial state of the booster to be $x = [x_i,\, 0]^T$ where $-\pi/2 \leq x_i \leq \pi/2$. It has been assumed that the separation does not impart any angular velocity to the booster. Finally we want the booster to have the following state or orientation $x(t_f) = [\text{-}\pi,\, 0]^T$. Sliding Mode control has been used to provide the control.

Since, $x_f \leq x_i$, initially the control will be in negative direction (i.e. pair 2-3 is active). Then at $\hat{t}$, the control becomes positive (i.e. pair 1-4 is active). The desired state is finally achieved at t_f. So, we have following control design,

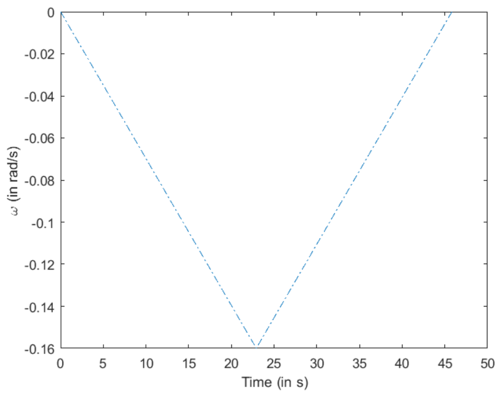
$$u(t) = \begin{cases} -u_m, & 0 \leq t \leq \hat{t} \\ u_m, & \hat{t} \leq t \leq t_f \end{cases} \tag{11}$$

where $\hat{t} = \sqrt{\frac{\pi + x_i}{u_m}}$ and and $t_f = 2\hat{t}$. Then solving for x using state space form, $x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}B(\tau)u(\tau)d\tau$ gives

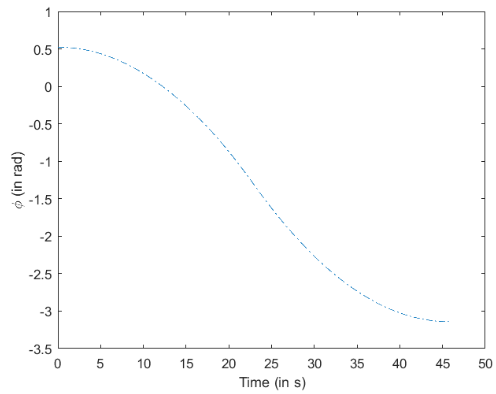
$$x(t) = \begin{cases} \begin{bmatrix} x_i - \frac{u_m}{2}t^2 \\ -u_m t \end{bmatrix}, & 0 \leq t \leq \hat{t} \\ \begin{bmatrix} x_i + (\hat{t}^2 + t^2/2 - 2t\hat{t})u_m \\ u_m(t - 2\hat{t}) \end{bmatrix}, & \hat{t} \leq t \leq t_f \end{cases} \tag{12}$$

For detailed calculation, refer to Appendix 1.

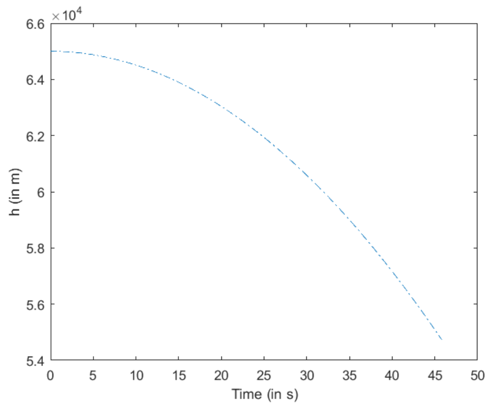
Now using the value of $\phi(t)$ obtained from above equation and using Runge-Kutta Method, the values of h, v, γ are solved till time t_{f1} (which is optimized value).



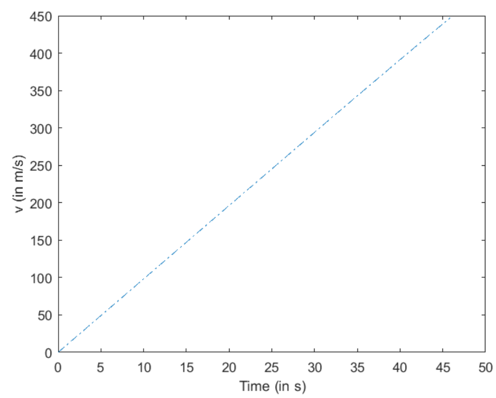
(a) ω (Rotation Velocity of the Booster)



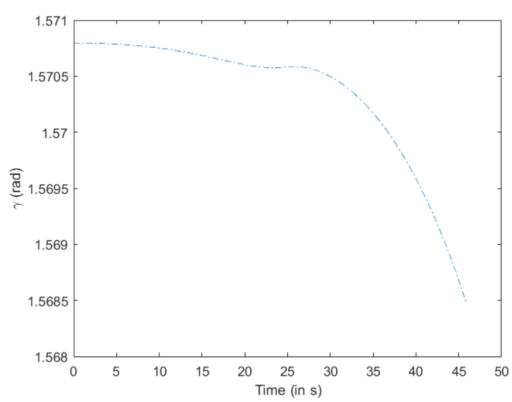
(b) ϕ (Orientation of the booster)



(c) h (Height of the booster)



(d) v (Velocity of the booster)



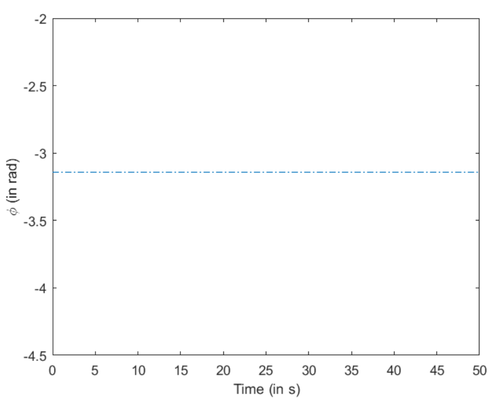
(e) γ (Flight Path Angle of the booster)

Figure 6: Simulation Result for Phase I

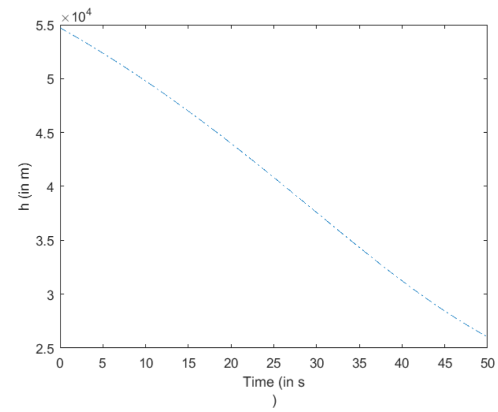
4. Phase II

In this phase, the booster falls horizontally. No control is used in this case. However, in real application, stability control needs to be incorporated for the booster to be stable.

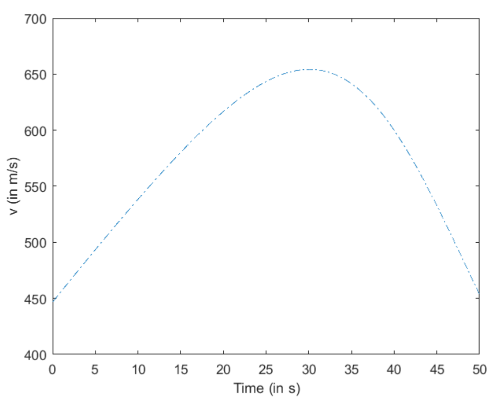
4.1. Simulation Results



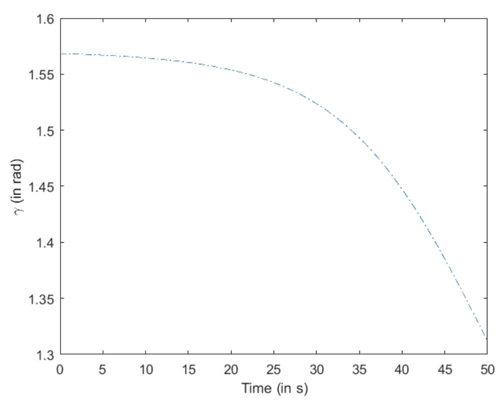
(a) ϕ (Orientation of the booster)



(b) h (Height of the booster)



(c) v (Velocity of the booster)



(d) γ (Flight Path Angle of the booster)

Figure 7: Simulation Result for Phase II

5. Phase III

In this phase, the state equations are same as in Phase I and so the result of the linearization will be same. So, also in this case, only ω and ϕ are controllable. The objective of the control is to make the booster vertical. The only difference from Phase I is the initial and final condition of ϕ. Take x = $[\phi,\omega]^T$. Then, x(t_i) = $[-\pi,0]^T$ and x(t_f) = $[-\pi/2\ ,\ 0]^T$. However, in this case, $x(t_f) \geq x(t_i)$, hence initially the control input will be positive. Therefore, we have the following control designs:

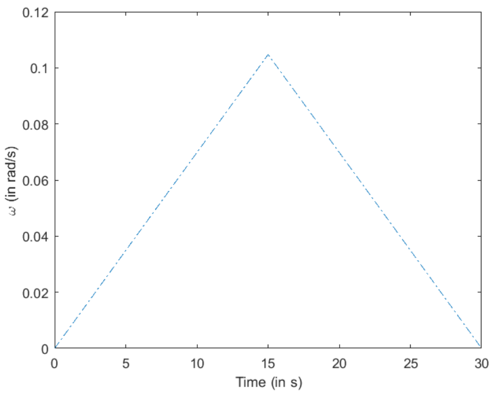
$$u(t) = \begin{cases} u_m, & 0 \leq t \leq \hat{t} \\ -u_m, & \hat{t} \leq t \leq t_f \end{cases} \tag{13}$$

where $\hat{t} = \sqrt{\frac{\pi}{2u_m}}$ and and $t_f = 2\hat{t}$. Then solving for x using state space form, $x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}B(\tau)u(\tau)d\tau$ gives

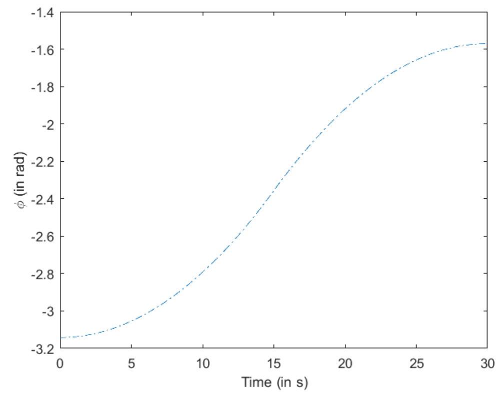
$$x(t) = \begin{cases} \begin{bmatrix} -\pi + \frac{u_m t^2}{2} \\ u_m t \end{bmatrix}, & 0 \leq t \leq \hat{t} \\ \begin{bmatrix} -\pi + 2u_m t\hat{t} - u_m \hat{t}^2 - u_m t^2/2 \\ 2u_m \hat{t} - u_m t \end{bmatrix}, & \hat{t} \leq t \leq t_f \end{cases} \tag{14}$$

For detailed calculation, refer to Appendix 1.

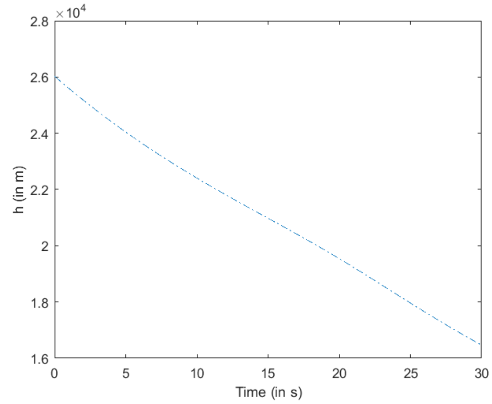
Now using the value of $\phi(t)$ obtained from above equation and using Runge-Kutta Method, the values of h, v, γ are solved till time t_f in this case.



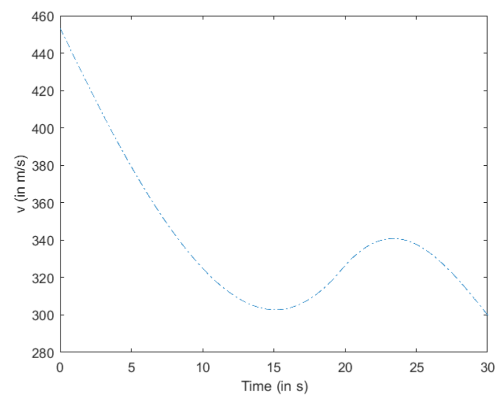
(a) ω (Rotation Velocity of the Booster)



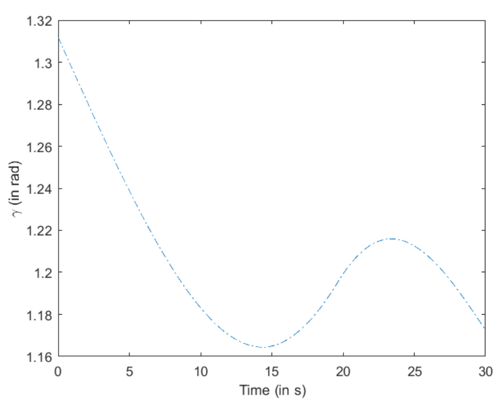
(b) ϕ (Orientation of the booster)



(c) h (Height of the booster)



(d) v (Velocity of the booster)



(e) γ (Flight Path Angle of the booster)

Figure 8: Simulation Result for Phase III

6. Phase IV

In this case, there will not be any use of small thrusters and only the main rocket engine will be used for control. The control input in this case is only α. We take $\chi = [h,\ v,\ \gamma,\ \omega,\ \phi]^T$ as the state variables and $\eta = [\alpha,\ _*]^T$ as the control inputs. The state equations are:

$$\dot{\chi} = f(\chi) = \begin{bmatrix} -v\sin\gamma \\ \frac{F_*}{m}\cos(\alpha-\phi+\gamma) - \frac{1}{2}v^2\frac{\rho_o}{m}C_D\left(\frac{\pi dL}{2}\sin(\gamma-\phi)\right)e^{\frac{-h}{H}} + g_o\sin\gamma \\ \frac{-F_*}{mv}\sin(\alpha-\phi+\gamma) - \frac{1}{2}v\frac{\rho_o}{m}C_L\left(\frac{\pi dL}{2}\sin(\gamma-\phi)\right)e^{\frac{-h}{H}} + \left(\frac{g_o}{v} - \frac{v}{h+R_o}\right)\cos\gamma \\ \frac{F_*L}{2I}\sin\alpha \\ \omega \end{bmatrix} \tag{15}$$

Assuming that the rocket is aligned in the radial direction and velocity is also in radial direction we have, $\gamma = \phi$.

$$\dot{\chi} = f(\chi) = \begin{bmatrix} -v\sin\gamma \\ \frac{F_*}{m}\cos(\alpha) + \kappa_D v^2 e^{\frac{-h}{H}} + g_o\sin\gamma \\ \frac{-F_*}{mv}\sin(\alpha) + \kappa_L v e^{\frac{-h}{H}} + \left(\frac{g_o}{v} - \frac{v}{h+R_o}\right)\cos\gamma \\ \frac{F_*L}{2I}\sin\alpha \\ \omega \end{bmatrix} \tag{16}$$

where $\kappa_D = -\frac{1}{2}\frac{\rho_o}{m}C_D\frac{\pi d^2}{4}$ and $\kappa_L = -\frac{1}{2}\frac{\rho_o}{m}C_L\frac{\pi d^2}{4}$ are constants. In this case, $A_\perp = \frac{\pi d^2}{4}$. For linearisation, the reference trajectory is taken to be $\chi_r = [0,\ 1,\ \pi/2,\ 0,\ -\pi/2]^T$ and $\eta_r = [0,\ 0]^T$. Here final velocity is taken to be 1 ms^{-1} to avoid singularity.

On Linearisation we get [Refer to Appendix 1 for derivation],

$$\dot{x} = Ax + Bu \tag{17}$$

where $A = \begin{bmatrix} 0.00e+0 & -1.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 \\ 1.03e-7 & -1.38e-3 & 0.00e+0 & 0.00e+0 & 0.00e+0 \\ -3.80e-8 & -2.55e-4 & 9.81e+0 & 0.00e+0 & 0.00e+0 \\ 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 \\ 0.00e+0 & 0.00e+0 & 0.00e+0 & 1.00e+0 & 0.00e+0 \end{bmatrix}$ and $B = \begin{bmatrix} 0.00e+0 & 0.00e+0 \\ 0.00e+0 & 1.18e-5 \\ -2.77e+2 & 0.00e+0 \\ 3.74e+2 & 0.00e+0 \\ 0.00e+0 & 0.00e+0 \end{bmatrix}$

and $x = \chi - \chi_r$ and $u = \eta - \eta_r$.

6.1. Controllability

Initially only α was to be used as control input but the system was not fully controllable. Hence, F_* is also taken as control input. A variable thrust can be provided by using a number of engines in the booster. The controllability matrix comes out to be:

$$P = \begin{bmatrix} 0.00e+0 & 0.00e+0 & 0.00e+0 & -1.18e-5 & 0.00e+0 & 1.62e-8 & 0.00e+0 & -2.12e-11 & 0.00e+0 & 2.76e-14 \\ 0.00e+0 & 1.18e-5 & 0.00e+0 & -1.62e-8 & 0.00e+0 & 2.12e-11 & 0.00e+0 & -2.76e-14 & 0.00e+0 & 3.60e-17 \\ -2.77e+2 & 0.00e+0 & -2.72e+3 & -3.00e-9 & -2.67e+4 & -2.94e-8 & -2.62e+5 & -2.88e-7 & -2.57e+6 & -2.83e-6 \\ 3.74e+2 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 \\ 0.00e+0 & 0.00e+0 & 3.74e+2 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 & 0.00e+0 \end{bmatrix}$$

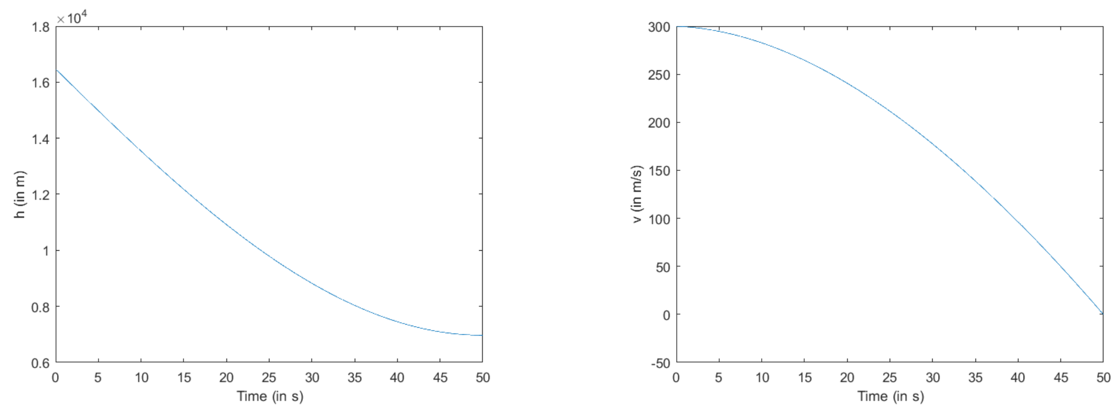
Rank of P = 5. Hence the system is controllable.

6.2. Control Design

Design the Proportional controller of the form of $u = -Kx$ and use Ackermann formula to find K. Using Relative Authority of g = $[30, 10]^T$ and desired poles are = -0.001, -0.0003, -0.004, -0.007, -0.0015. Using Ackerman Formula, K matrix comes out to be:

$$K = \begin{bmatrix} 0.0346 & -26.5865 & -0.0354 & 0.0000 & -0.0000 \\ 0.0115 & -8.8622 & -0.0118 & 0.0000 & -0.0000 \end{bmatrix}$$

6.3. Simulation Results



(a) h (Height of the booster)

(b) v (Velocity of the booster)

Figure 9: Simulation Result for Phase IV

7. Conclusion

In this paper, the control system for rocket reentry and landing was designed using Sliding Mode control and Proportional Control. The rocket was effectively able to land vertically on the surface.

8. Appendix 1: Calculations

8.1. Phase I and Phase II Calculations

We have

$$x = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$\therefore A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$; $e^{A(t-t_i)} = \begin{bmatrix} 1 & t-t_i \\ 0 & 1 \end{bmatrix}$

Initial condition, $x(t_i) = \begin{bmatrix} x_i \\ 0 \end{bmatrix}$ and $x(t_f) = \begin{bmatrix} x_f \\ 0 \end{bmatrix}$

Now we have,

$$H = 1 + \lambda_1 x + \lambda_2 u \quad \text{(Assuming } L = 1\text{)}$$

We have, $\dot{\lambda} = -\dfrac{\partial H}{\partial x} \lambda$

$$\dot{\lambda}_1 = 0 \Rightarrow \lambda_1 = c_1$$

$$\dot{\lambda}_2 = -\lambda_1 \Rightarrow \dot{\lambda}_2 = -c_1 \Rightarrow \lambda_2 = -c_1 t + c_2$$

Now, $\lambda_2 = 0 \Rightarrow \hat{t} = \dfrac{c_2}{c_1}$ gives the switching condition for the control design.

Since λ_2 is a linear function of time. Hence only once there will be switching.

Assume initially $u = u_0$, then,

For $0 \le t \le \hat{t}$, we have,

$$\hat{x}(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)} B(\tau)\, u(\tau)\, d\tau$$

$$= \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ 0 \end{bmatrix} + \int_0^t \begin{bmatrix} 1 & t-\tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_0\, d\tau$$

$$= \begin{bmatrix} x_i \\ 0 \end{bmatrix} + u_0 \int_0^t \begin{bmatrix} (t-\tau)\, d\tau \\ 1\, d\tau \end{bmatrix}$$

$$= \begin{bmatrix} x_i \\ 0 \end{bmatrix} + u_0 \begin{bmatrix} \left(t\tau - \frac{\tau^2}{2}\right)_0^t \\ (\tau)_0^t \end{bmatrix}$$

$$= \begin{bmatrix} x_i + u_0 \left[t^2 - \frac{t^2}{2}\right] \\ 0 + u_0 t \end{bmatrix} = \begin{bmatrix} x_i + \frac{u_0 t^2}{2} \\ u_0 t \end{bmatrix}$$

$$\therefore \hat{x}(\hat{t}) = \begin{bmatrix} x_i + \frac{u_0 \hat{t}^2}{2} \\ u_0 \hat{t} \end{bmatrix}$$

Now at time $t = \hat{t}$, switching occurs and thus, $u = -u_0$

$\therefore$ For $\hat{t} \le t \le t_f$

$$\hat{x}(t) = e^{A(t-\hat{t})} x(\hat{t}) + \int_{\hat{t}}^{t} e^{A(t-\tau)} B(\tau)\, u(\tau)\, d\tau$$

$$= \begin{bmatrix} 1 & t-\hat{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i + \frac{u_0 \hat{t}^2}{2} \\ u_0 \hat{t} \end{bmatrix} \circledast$$

$$+\int_{\hat t}^{t} \begin{bmatrix}1 & t-\tau\\ 0 & 1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}(-u_0)\,d\tau$$

$$\therefore \hat x(t) = \begin{bmatrix} x_i + \frac{u_0\hat t^2}{2} + u_0(\hat t t - \hat t^2) \\ u_0 \hat t\end{bmatrix} - u_0 \begin{bmatrix}\int_{\hat t}^{t} (t-\tau)\,d\tau \\ \int_{\hat t}^{t} d\tau\end{bmatrix}$$

$$= \begin{bmatrix} x_i + u_0 t\hat t - \frac{u_0\hat t^2}{2} \\ u_0\hat t\end{bmatrix} - u_0\begin{bmatrix}\left(t^2-\frac{t^2}{2}\right) - \left(t\hat t - \frac{\hat t^2}{2}\right) \\ t-\hat t\end{bmatrix}$$

$$= \begin{bmatrix} x_i + u_0 t\hat t - \frac{u_0\hat t^2}{2} - \frac{u_0 t^2}{2} + u_0 t\hat t - u_0\frac{\hat t^2}{2} \\ u_0\hat t - u_0 t + u_0\hat t\end{bmatrix}$$

$$= \begin{bmatrix} x_i + 2u_0 t\hat t - u_0\hat t^2 - \frac{u_0 t^2}{2} \\ 2u_0\hat t - u_0 t\end{bmatrix}$$

$$\therefore \hat x(t) = \begin{bmatrix} x_i + 2u_0 t\hat t - u_0\hat t^2 - \frac{u_0 t^2}{2} \\ 2u_0\hat t - u_0 t\end{bmatrix}$$

$\therefore$ For $\hat x(t_f) = \begin{bmatrix}x_f\\ 0\end{bmatrix}$

we have,

$$2u_0\hat t - u_0 t_f = 0 \Rightarrow t_f = 2\hat t$$

And,

$$x_i + 2u_0 t_f \hat{t} - u_0 \hat{t}^2 - \frac{u_0 t_f^2}{2} = x_f$$

$$\Rightarrow \quad 2u_0 \times (2\hat{t})\hat{t} - u_0 \hat{t}^2 - \frac{u_0 (2\hat{t})^2}{2} = x_f - x_i$$

$$\Rightarrow \quad 4\hat{t}^2 u_0 - \hat{t}^2 u_0 - 2\hat{t}^2 u_0 = x_f - x_i$$

$$\hat{t}^2 u_0 = x_f - x_i$$

$$\hat{t} = \sqrt{\frac{x_f - x_i}{u_0}}$$

For PHASE I.

$$u = \begin{cases} -u_m \ ; & 0 \le t \le \hat{t} \\ u_m \ ; & \hat{t} \le t \le t_f \end{cases}$$

$-\pi/2 \le x_i \le \pi/2$ and $x_f = -\pi$ and $u_0 = -u_m$

$\therefore$ we get,

$$\hat{t} = \sqrt{\frac{-\pi - x_i}{-u_m}} = \sqrt{\frac{\pi + x_i}{u_m}}$$

and $t_f = 2\hat{t}$

And,

$$x = \begin{cases} \begin{bmatrix} x_i - \frac{u_m t^2}{2} \\ -u_m t \end{bmatrix} ; & 0 \le t \le \hat{t} \\ \begin{bmatrix} x_i + (-2t\hat{t} + \hat{t}^2 + \frac{t^2}{2}) u_m \\ u_m t - 2u_m \hat{t} \end{bmatrix} & \hat{t} \le t \le t_f \end{cases}$$

For ~~PHASH~~ PHASE II

$$u = \begin{cases} u_m \; ; & 0 \le t \le \hat{t} \\ -u_m \; ; & \hat{t} \le t \le t_f \end{cases}$$

$x_i = -\pi$ and $x_f = -\pi/2$ and $u_0 = u_m$

$\therefore$ We get,
$$\hat{t} = \sqrt{\frac{-\pi/2 + \pi}{u_m}} = \sqrt{\frac{\pi}{2u_m}}$$

$$t_f = 2\hat{t}$$

And,

$$x = \begin{cases} \begin{bmatrix} -\pi + \dfrac{u_m t^2}{2} \\ u_m t \end{bmatrix} & ; 0 \le t \le \hat{t} \\ \begin{bmatrix} -\pi + 2u_m t\hat{t} - u_m\hat{t}^2 - \dfrac{u_m t^2}{2} \\ 2u_m\hat{t} - u_m t \end{bmatrix} & \hat{t} \le t \le t_f \end{cases}$$

We have,

$$X = [h, v, \gamma, \omega, \phi]^T \; ; \; \eta = [\alpha, F_*]^T$$

$$\dot{X} = f(\,) = \begin{bmatrix} -v\sin\gamma \\ \frac{F_*}{m}\cos\alpha + k_D v^2 e^{-h/H} + g_0 \sin\gamma \\ -\frac{F_*}{mv}\sin\alpha + k_L v e^{-h/H} + \left(\frac{g_0}{v} - \frac{v}{h+R_0}\right)\cos\gamma \\ \frac{F_* L}{2I}\sin\alpha \\ \omega \end{bmatrix}$$

Reference Trajectory,

$$X_r = [0, 1, \pi/2, 0, -\pi/2]^T$$

$$\eta_r = [0, 0]^T$$

And,

$$x = \xi - \xi_r \; ; \; u = \eta - \eta_r$$

Then $\dot{x} = Ax + Bu$ where

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial h} & \frac{\partial f_1}{\partial v} & \cdots & \frac{\partial f_1}{\partial \phi} \\ \vdots & & & \\ \frac{\partial f_5}{\partial h} & & \cdots & \frac{\partial f_5}{\partial \phi} \end{bmatrix}_{(x_r, \eta_r)} \text{ and } B = \begin{bmatrix} \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial F_*} \\ \vdots & \vdots \\ \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial F_*} \end{bmatrix}_{(x_r, \eta_r)}$$

$$
\therefore A = \begin{bmatrix}
0 & -\sin\gamma & -v\cos\gamma & 0 & 0 \\
\dfrac{-K_D v^2 e^{-h/H}}{H} & 2vK_D e^{-h/H} & -g_0\cos\gamma & 0 & 0 \\
\dfrac{K_L v e^{-h/H}}{H} + \dfrac{v\cos\gamma}{(h+R_0)^2} & K_L e^{-h/H} + \left(\dfrac{-g_0}{v^2} - \dfrac{1}{h+R_0}\right)\cos\gamma & \left(\dfrac{g_0}{v} - \dfrac{v}{h+R_0}\right) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

$$
= \begin{bmatrix}
0 & -1 & 0 & 0 & 0 \\
\dfrac{-K_D}{H} & 2K_D & 0 & 0 & 0 \\
\dfrac{K_L}{H} & K_L & g_0 - \dfrac{1}{R_0} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

and $$B = \left.\begin{bmatrix} 0 & 0 \\ -\frac{F_*}{m}\sin\alpha & \frac{1}{m}\cos\alpha \\ -\frac{F_*}{mv}\cos\alpha & -\frac{1}{mv}\sin\alpha \\ \frac{F_* L}{2I}\cos\alpha & -\frac{L\sin\alpha}{2I} \\ 0 & 0 \end{bmatrix}\right|_{\alpha_r, r}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m} \\ -\frac{F_*}{m} & 0 \\ \frac{F_* L}{2I} & 0 \\ 0 & 0 \end{bmatrix}$$

Also, Initially

$$x_i = \varepsilon_{v_i} - \varepsilon_{r_i}$$

And,

$$\varepsilon_f = x_f + \varepsilon_{r_f}$$

9. Appendix II: MATLAB Code

9.1. simulation.m

```
% Phase I
x0 = [pi/6;0];
tf1 = 15;
J0 = pi/(0.0002*tf1^2);
% t less than t_hat
um = pi/(2*tf1^2);
t_cap = sqrt((pi+x0(1))/((um)));
t_f = 2 * t_cap;
timestep = 0.001;
t1 = 0:timestep:t_cap;
num_step1 = size(t1);
x1 = [x0(1) - um .* (t1.^2)/2; -um.*t1];
% t greater than t_hat
t2 = t_cap:timestep:t_f;
num_step2 = size(t2);
x2 = [x0(1) + (t2.^2/2 + ...
t_cap^2 - 2*t2 * t_cap)*um;-2*um*t_cap + um*t2];

time = [t1 t2];
x = [x1 x2];
u = [-um*ones(num_step1) um*ones(num_step2)];
% Solving for h, v, gamma
xi = [65000; 0; pi/2]; % Initial Conditions
result = runge_kutta(xi, x(1,:), timestep, num_step1(1,2)+num_step2(1,2));
figure
plot(time, x(1,:),'-.')
figure
plot(time, x(2,:), '-.')
figure
plot(time, u, '-.')
```

```
figure
plot(time, result(1,:), '-.')
figure
plot(time, result(2,:), '-.')
figure
plot(time, result(3,:), '-.')

% Phase II
time = 0:timestep:50;
num_step = size(time);
x = [x2(1, end)*ones(num_step); zeros(num_step)];
u = zeros(num_step);
xi = result(:,end); % Initial Conditions
result = runge_kutta(xi, x(1,:), timestep, num_step(1,2));
figure
plot(time, x(1,:),'-.')
figure
plot(time, x(2,:), '-.')
figure
plot(time, u, '-.')
figure
plot(time, result(1,:), '-.')
figure
plot(time, result(2,:), '-.')
figure
plot(time, result(3,:), '-.')

% Phase III

x0 = [x2(1, end);0];
t_cap = sqrt(pi/(2*um));
t_f = 2 * t_cap;
```

```
timestep = 0.001;
t1 = 0:timestep:t_cap;
num_step1 = size(t1);
x1 = [-pi + um .* (t1.^2)/2; um.*t1];
% t greater than t_hat
t2 = t_cap:timestep:t_f;
num_step2 = size(t2);
x2 = [-pi + (-t2.^2/2 - ...
t_cap^2 + 2*t2 * t_cap)*um;2*um*t_cap - um*t2];
time = [t1 t2];
x = [x1 x2];
u = [um*ones(num_step1) -um*ones(num_step2)];
% Solving for h, v, gamma
xi = result(:,end); % Initial Conditions
result = runge_kutta(xi, x(1,:), timestep, num_step1(1,2)+num_step2(1,2));
figure
plot(time, x(1,:),'-.')
figure
plot(time, x(2,:), '-.')
figure
plot(time, u, '-.')
figure
plot(time, result(1,:), '-.')
figure
plot(time, result(2,:), '-.')
figure
plot(time, result(3,:), '-.')

% Phase IV
g_0 = 9.81;
R_o = 6371 * 1000;
%Rocket Parameters
```

```
m = 85000;
L = 45;
d = 10;
rho0 = 1.225;
H = 6700;
CD = 1.22;
CL = 0.45;
kD = -(rho0*CD*pi*d^2)/(8*m);
kL = -(rho0*CL*pi*d^2)/(8*m);
F = 23575574.48;
I =2/3*m*(d/2)^2;
A = [0 -1 0 0 0; -kD/H 2*kD 0 0 0; kL/H kL g_0 - 1/R_o 0 0; 0 0 0 0 0; 0 0 0 1 0];
B = [0 0;0 1/m;-F/m 0;F*L/(2*I) 0;0 0];
P= ctrb(A,B);
r = rank(P);
g = [30;10];
b = B * g;
dP= [-0.001, -0.0003, -0.004, -0.007, -0.0015];
k = acker(A, b, dP);
K = g * k
xr = [0;1;pi/2;0;-pi/2];
xi = [result(:,end);x2(1, end);0] - xr;

time = 0:0.001:50;
size_t = size(time);
len = size_t(2);
x = zeros(5, len);
u = zeros(2, len);
for idx = 1:len
    x_t = expm((A - B * K)*time(idx))*xi;
    x(:,idx) = x_t + xr;
    u(:, idx) = K * x_t;
```

```
end


figure
plot(time, x(1,:))
figure
plot(time, x(2,:))
```

9.2. f.m

```
function result = f(x, phi)
    v = x(2);
    h = x(1);
    gamma = x(3);
    g_0 = 9.81;
    R_o = 6371 * 1000;
    %Rocket Parameters
    m = 85000;
    L = 45;
    d = 10;
    rho0 = 1.225;
    H = 6700;
    CD = 1.22;
    CL = 0.45;
    kD = -(rho0*CD*pi*d*L)/(4*m);
    kL = -(rho0*CL*pi*d*L)/(4*m);
    result = [-v*sin(gamma); kD*(v^2)*abs(sin(gamma - phi))*exp(-(h/H)) ...
        + g_0*sin(gamma); kL*v*abs(sin(gamma - phi))*exp(-(h/H))+(g_0/(v + 10^(-4))...
        - v/(h + R_o))*cos(gamma)];
end
```

9.3. runge_kutta.m

```
%  runge-kutta fourth order


function y = runge_kutta(xi,phi,h, n)
```

```
    y = zeros(3,n);
    for i=1:n
        k1 = h*f(xi, phi(1,i));
        k2 = h*f(xi + 0.5 * k1, phi(1,i));
        k3 = h*f(xi + 0.5 * k2, phi(1,i));
        k4 = h*f(xi + k3, phi(1,i));
%         update
        xi = xi + (1.0 / 6.0)*(k1 + 2 * k2 + 2 * k3 + k4);
        y(:,i) = xi;
    end
end
```

References

[1] ISRO, Rlv-td.
URL https://www.isro.gov.in/launcher/rlv-td

[2] ISRO, India's reusable launch vehicle-technology demonstrator (rlv-td), successfully flight tested.
URL https://www.isro.gov.in/update/23-may-2016/india%E2%80%99s-reusable-launch-vehicle-technology-demonstrator-rlv-td-successfully

[3] U. Walter, U. Walter, Astronautics, Springer, 2008.