Simplified 6-DOF Trajectory Simulations for Stardust SRC and Space Shuttle

Presented By:

Sanjana Srivastava and Nagachandra Nagaraj

for AE 598 Planetary Entry, Fall 2021

Department of Aerospace Engineering, UIUC, Illinois

ACKNOWLEDGEMENT

This work is submitted to Professor Zachary Putnam for project evaluation for the course AE 598 Planetary Entry in the Fall semester of 2021 at the University of Illinois at Urbana-Champaign. We would like to express our gratitude to Professor Putnam and the Department of Aerospace Engineering at UIUC for providing us with the opportunity to conduct independent research on relevant topics as being taught in our courses.

We are incredibly thankful to Professor Putnam for his guidance regarding technical aspects of this project as well as personally for providing an appropriate environment to work especially in unforeseen circumstances.

Nagarachandra and I conducted our preliminary research together, after finalizing our equations and trajectories chosen, Nagachandra did the coding work of the equations in Python, while I carried out the literature research for comparisons and documented the report.

ABSTRACT

The aim of this work is to gain a better understanding of the multitude of aspects and factors involved in optimization of a mission trajectory, from deriving relevant equations, to procuring as accurate aerodynamic and structural databases as possible and to understand the process of solving the relevant equations of motions to achieve the best simulations for mission trajectory design. Optimal trajectory design involves research into the robustness of solvers used, maximum accuracy in estimating the guiding equations for the trajectory and best estimating structural and other limitations to develop the most accurate simulation possible. Spacecraft trajectory design is an immensely complex engineering problem and has made great strides in precision and accuracy from earlier years.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Trajectory optimization and reentry dynamics have been a complex engineering problem in the spaceflight industry. It is critical to develop as precise and accurate calculations as possible to ensure the success of a mission, and trajectory planning is arguably one of the most crucial factors. Trajectory optimization involves incorporation of aerodynamic forces, vehicle restrictions, maneuvering capabilities, precise orbital calculations, safety considerations, and a multitude of several other factors. A wide range of research has been conducted in this field, with formulation of different mathematical models, as well as finding the efficacy of different types of algorithmic solutions for the trajectories, from numerical solutions to using Artificial Intelligence and Neural Networks. This paper aims to develop a 6-DOF model of the equations of motions for two different trajectory simulations, code and solve the model using numerical integration in Python software and compare the results to similar research as found in literature. This work deals with the Stardust reentry capsule and the Space Shuttle trajectory simulations.

The Stardust mission was successfully launched on 7th February 1999 from Cape Canaveral with the aim to bring back comet samples from the Wild-4 comet for its analysis. The capsule successfully returned to Earth on January 15th, 2006, at the UTTR in Utah. NASA's Space Shuttle was a partially reusable low-earth orbit (LEO) spacecraft operational from 1981 to 2011, launching multiple satellites, interplanetary probes, as well as the Hubble Space Telescope. The shuttle also carried as many as seven astronauts into space at a time, and was later used for servicing the HST and other military applications later in its lifetime.

CHAPTER 2

LITERATURE REVIEW

2.1 Optimal Control Based Six-Degree-of-Freedom Mission Planning for Reentry Trajectories [1]

Peter Davis conducted extensive thesis research for the Air Force Institute of Technology in 3DOF and 6DOF formulations for reentry dynamics, presenting a fully developed 6DOF dynamic system for an unpowered reentry vehicle. The paper aims to build a high-fidelity simulation by defining state variables and control inputs, developing an entire 6DOF aerodynamic database, and using GPOPS-II optimal control software to solve the problem. First, a proper planetary model with atmospheric and gravitational database is defined, along with an extensive system of six reference frames used to derive the equations of motion. The final 12 6-DOF equations of motion with 12 state variables are defined as follows:

$$
\begin{aligned}
\dot{r} &= v\sin(\gamma) \\
\dot{\mu} &= \frac{v\cos(\gamma)\cos(\psi)}{r\cos(\lambda)} \\
\dot{\lambda} &= \frac{v\cos(\gamma)\sin(\psi)}{r} \\
\dot{v} &= \frac{\rho v^2}{2}\frac{S}{m}\{C_y\sin(\beta) + C_x\cos(\alpha)\cos(\beta) + C_z\cos(\beta)\sin(\alpha)\} - \frac{\mu_\oplus}{r^2}\sin(\gamma) \\
&\quad - r\omega_\oplus^2\cos(\lambda)\left(-\cos(\lambda)\sin(\gamma) + \sin(\lambda)\cos(\gamma)\sin(\psi)\right) \\
\dot{\psi} &= \Big(\frac{\rho v^2}{2}\frac{S}{m}\{(C_y(\cos(\sigma)\cos(\beta)\cos(\eta) + \cos(\beta)\sin(\sigma)\sin(\eta)) \\
&\quad + C_x(\cos(\sigma)(\sin(\alpha)\sin(\eta) - \cos(\alpha)\cos(\eta)\sin(\beta)) \\
&\quad - \sin(\sigma)(\cos(\eta)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\eta))) \\
&\quad - C_z(\cos(\sigma)(\cos(\alpha)\sin(\eta) + \cos(\eta)\sin(\alpha)\sin(\beta)) \\
&\quad - \sin(\sigma)(\cos(\alpha)\cos(\eta) - \sin(\alpha)\sin(\beta)\sin(\eta)))\} \\
&\quad - 2v\omega_\oplus\{\sin(\lambda)\cos(\gamma) - \cos(\lambda)\sin(\gamma)\sin(\psi)\} \\
&\quad - \frac{v^2\cos^2(\gamma)}{r}\cos(\psi)\tan(\lambda) - r\omega_\oplus^2\cos(\lambda)\sin(\lambda)\cos(\psi)\Big)\frac{1}{v\cos(\gamma)} \\
\dot{\gamma} &= \Big(\frac{\rho v^2}{2}\frac{S}{m}\{(C_y(\cos(\sigma)\cos(\beta)\sin(\eta) - \cos(\beta)\sin(\sigma)\cos(\eta)) \\
&\quad - C_x(\cos(\sigma)(\sin(\alpha)\cos(\eta) + \cos(\alpha)\sin(\eta)\sin(\beta)) \\
&\quad + \sin(\sigma)(\sin(\eta)\sin(\alpha) - \cos(\alpha)\sin(\beta)\sin(\eta))) \\
&\quad + C_z(\cos(\sigma)(\cos(\alpha)\cos(\eta) - \sin(\eta)\sin(\alpha)\sin(\beta)) \\
&\quad - \sin(\sigma)(\cos(\alpha)\sin(\eta) + \sin(\alpha)\sin(\beta)\cos(\eta)))\} \\
&\quad - \frac{\mu_\oplus}{r^2}\cos(\gamma) + 2v\omega_\oplus\{\cos(\lambda)\cos(\psi)\} \\
&\quad + \frac{v^2\cos^2(\gamma)}{r} + r\omega_\oplus^2\{\cos(\lambda)\cos(\gamma) + \sin(\lambda)\sin(\gamma)\sin(\psi)\}\Big)\frac{1}{v} \\
\dot{p} &= \frac{1}{I_{xx}}\left\{L + I_{zx}(\dot{r} + pq) + (I_{yy} - I_{zz})qr + I_{yz}(q^2 - r^2) + I_{xy}(\dot{q} - rp)\right\} \\
\dot{q} &= \frac{1}{I_{yy}}\left\{M + I_{zx}(r^2 - p^2) + (I_{zz} - I_{xx})rp + I_{yz}(\dot{r} - pq) + I_{xy}(\dot{p} + qr)\right\} \\
\dot{r} &= \frac{1}{I_{zz}}\left\{N + I_{zx}(\dot{p} - qr) + (I_{xx} - I_{yy})pq + I_{yz}(\dot{q} + rp) + I_{xy}(p^2 - q^2)\right\} \\
\begin{matrix}\dot{\sigma}\\ \dot{\alpha}\\ \dot{\beta}\end{matrix} &= \mathbf{G}^{-1}\begin{bmatrix} p - p_F \\ q - q_F \\ r - r_F \end{bmatrix}_B
\end{aligned}
$$

Davis then conducted a Neural Network analysis for the aerodynamic database and defined the test scenario of a pull-up maneuver. Finally, optimal control analysis is conducted for the minimum time trajectory. The 6-DOF simulations were found to be realistic and within reasonable limits.

1.2. Optimal Guidance Command Generation and Tracking for Resuable Launch Vehicle Reentry [2]

In 2005, Bollino developed a robust GnC architecture for resuable launch vehicles incorporating optimal trajectory simulations by coherently integrating three previous developed architectures. His work involves developing an inner-loop control design and solving the optimal control problem using the Legendre Pseudospectral Method followed by a nonlinear problem solver to solve for reentry applications. He describes the outer and inner-loop command generation architecture using a simplified and decoupled version of the full 6-DOF equations of motion. The equations used by Bollino are:

$$\dot{x} = V\cos\gamma\cos\psi$$
$$\dot{y} = V\cos\gamma\sin\psi$$
$$\dot{z} = V\sin\gamma$$
$$\dot{V} = -\frac{D}{m} - g\sin\gamma$$
$$\dot{\gamma} = \frac{L}{mV} - \frac{g\cos\gamma}{V}$$
$$\dot{\psi} = \frac{L\sin\phi}{mV\cos\gamma}$$

Aerodynamic data was computed using table lookup based on previous atmospheric models. The final optimal control problem developed was summarized as:

$$\min_u J\left(\underline{x}(\tau),\underline{u}(\tau),\tau_0,\tau_f\right)=E\left(\underline{x}(\tau_0),\underline{x}\left(t_f\right),\tau_0,\tau_f\right)+\int_{\tau_0}^{\tau_f}F\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right)d\tau$$

subject to $\quad \dot{\underline{x}}=\underline{f}\left(\underline{x},\underline{u},\tau\right)$

$$\underline{h}_l\le\underline{h}\left(\underline{x},\underline{u},\tau\right)\le\underline{h}_u$$

$$\underline{e}_l\le\underline{e}\left(\underline{x}(\tau_0),\underline{x}\left(\tau_f\right),\tau_0,\tau_f\right)\le\underline{e}_u$$

$$\underline{x}_l\le\underline{x}(\tau)\le\underline{x}_u$$

$$\underline{u}_l\le\underline{u}(\tau)\le\underline{u}_u$$

The inner-loop of the architecture was developed to be reconfigurable using dynamic inversion with control allocation. It was found that the cost functions were well within acceptable tolerance limits.

CHAPTER 3
METHODOLOGY

This paper uses the simplified, decoupled 6-DOF equations of motions as developed by Bollini [2] coded into Python and solved using numerical integration. The aerodynamic database and trajectory simulations selected are for the Stardust reentry capsule and NASA's resuable LEO Space Shuttle. The basic equations of motions used are:

$$\dot{x} = V\cos\gamma\cos\psi$$
$$\dot{y} = V\cos\gamma\sin\psi$$
$$\dot{z} = V\sin\gamma$$
$$\dot{V} = -\frac{D}{m} - g\sin\gamma$$
$$\dot{\gamma} = \frac{L}{mV} - \frac{g\cos\gamma}{V}$$
$$\dot{\psi} = \frac{L\sin\phi}{mV\cos\gamma}$$

The lift and drag forces are calculated using:

$$L = \frac{1}{2}\rho(z)V^2C_L(\alpha, M)S_{ref}$$
$$D = \frac{1}{2}\rho(z)V^2C_D(\alpha, M)S_{ref}$$

3.1. Stardust Sample Return Capsule

The aerodynamic database for Stardust Sample Return, namely lift and drag coefficients Cl and Cd, were taken from Shoemaker et al.'s Trajectory Estimation of the Hayabusa Sample Return Capsule Using Optical Sensors [3]. The Stardust sample return carrying samples from comet Wild-2 entered Earth's atmosphere at a velocity of 12.9km/s, making it the fastest reentry for a man-made spacecraft till date. The flight path angle at reentry was found to be -8.2 degrees.

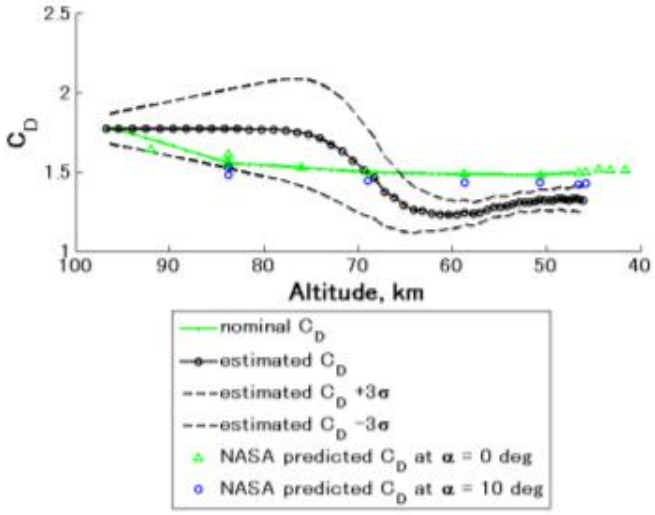

Figure 1: Stardust Cd estimates [3]

3.2. Space Shuttle

Aerodynamic data for the Space Shuttle is taken from Dilao et al's Dynamic Guidance of Gliders in Planetary Atmospheres [4]. Typical Space Shuttle reentry involved four phases – reentry burn, atmospheric reentry, gliding phase, and final approach and landing. The paper details the variables and structural limitations involved in Space Shuttle's reentry dynamics and conducts Runge-Kutta simulations for the dynamic guidance trajectories.

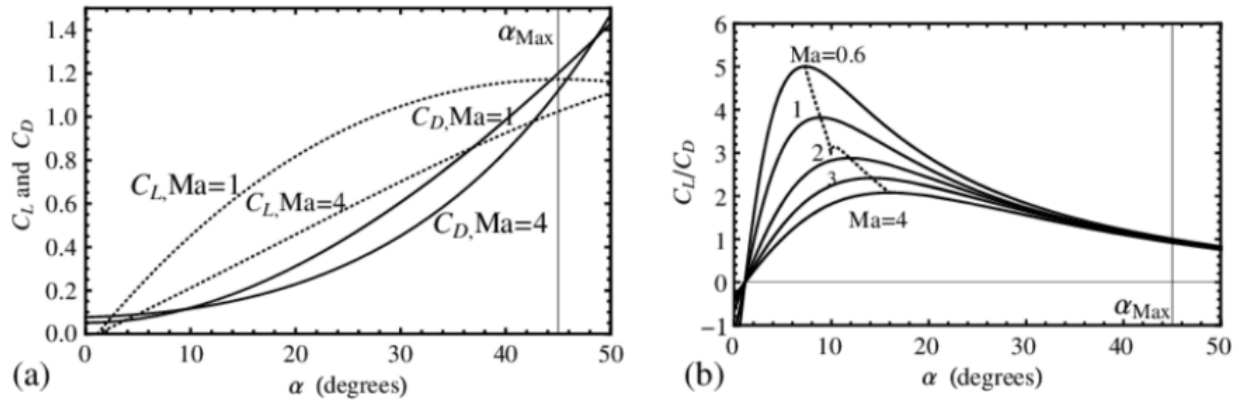

Figure 2: (a) Cl and Cd for Space Shuttle as a function of AOA, and (b) L/D ratio as a function of AOA for different Mach numbers [4].

CHAPTER 4

RESULTS AND ANALYSIS

4.1. Stardust

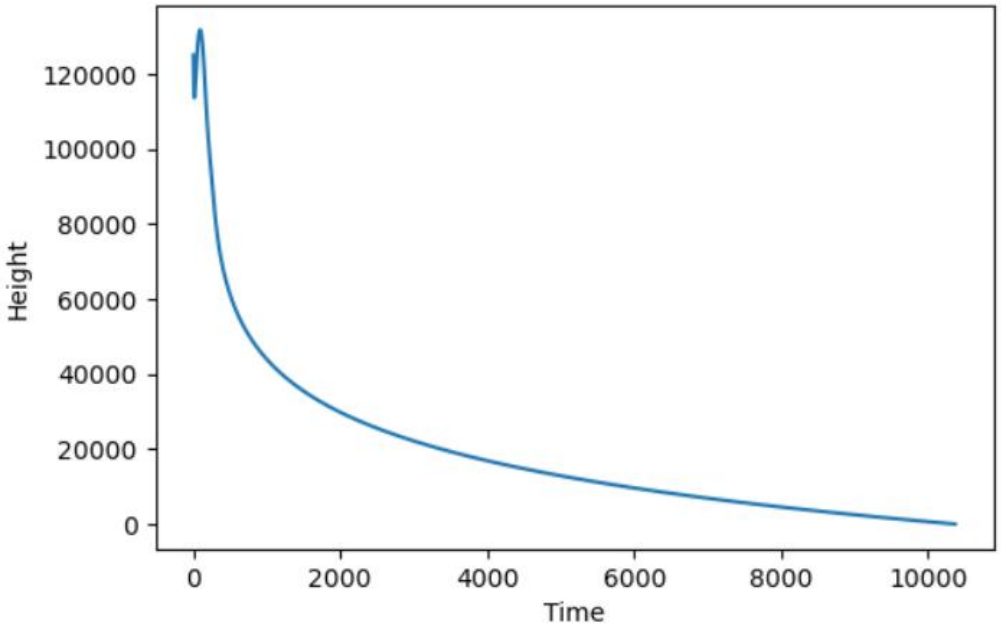

Figure 3: Height (m) v/s Time (s) for Stardust

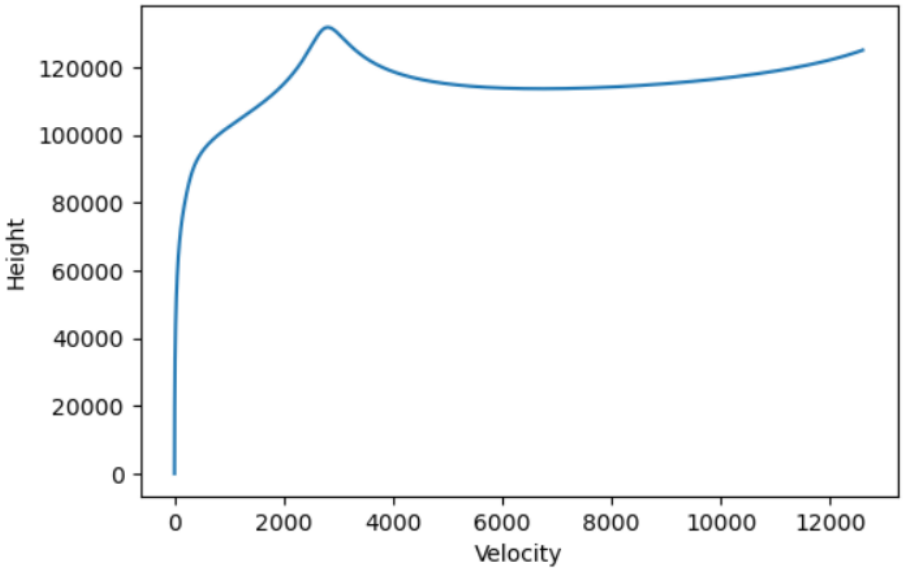

Figure 4: Height (m) v/s Velocity (m/s) for Stardust

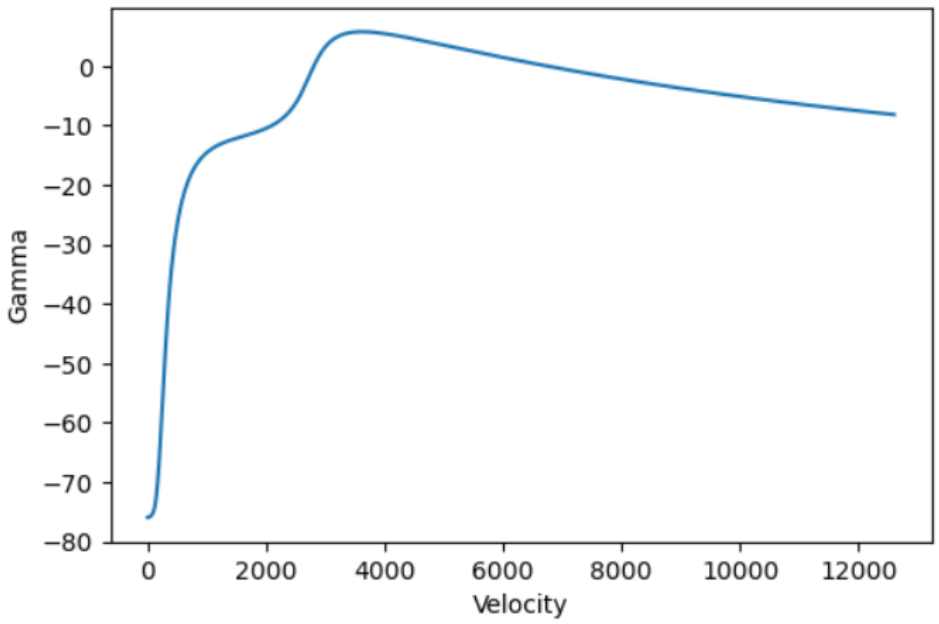

Figure 5: Flight path angle (deg) v/s Velocity (m/s) for Stardust

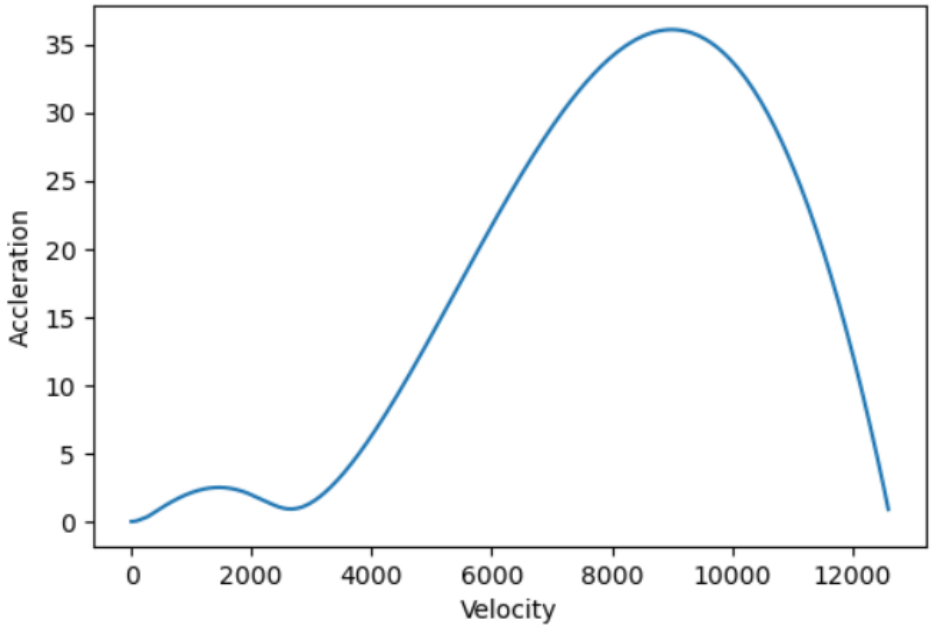

Figure 6: Acceleration (m/s^2) v/s Velocity (m/s) for Stardust

These results are compared with Desai et al.'s work in Entry Trajectory Issues for the Stardust Sample Return Capsule [5], as shown in the figure:

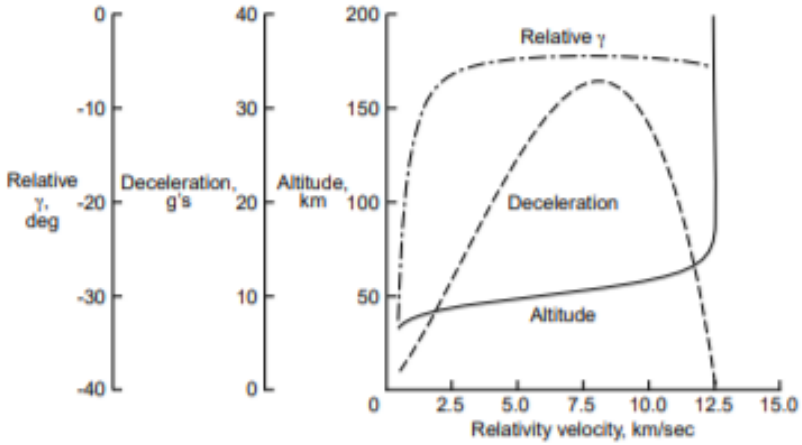

Figure 7: Stardust mission entry [5]

4.2. Space Shuttle

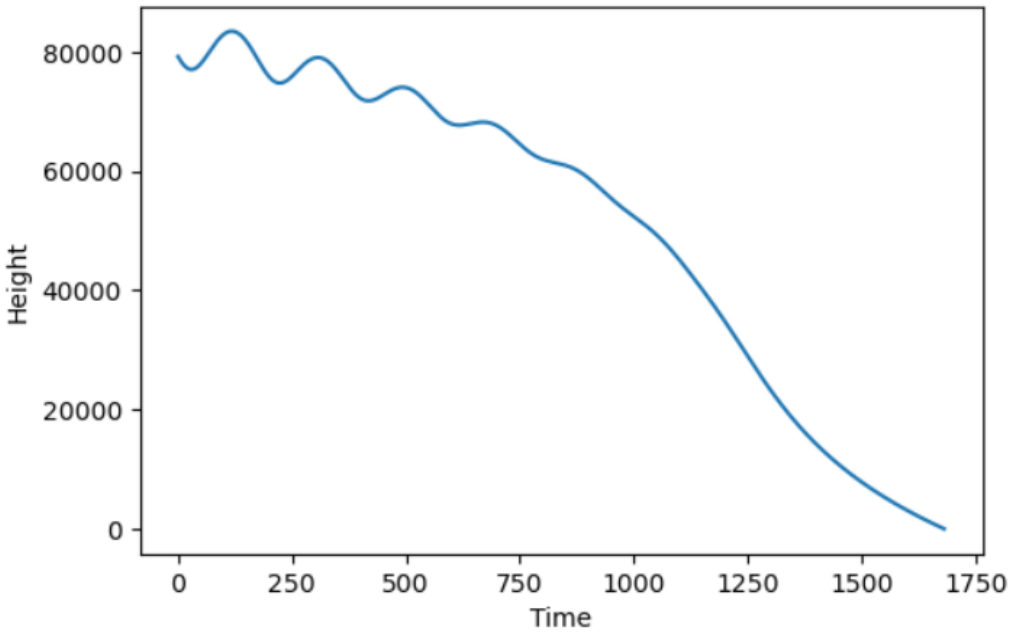

Figure 8: Height (m) v/s Time (s) for Space Shuttle

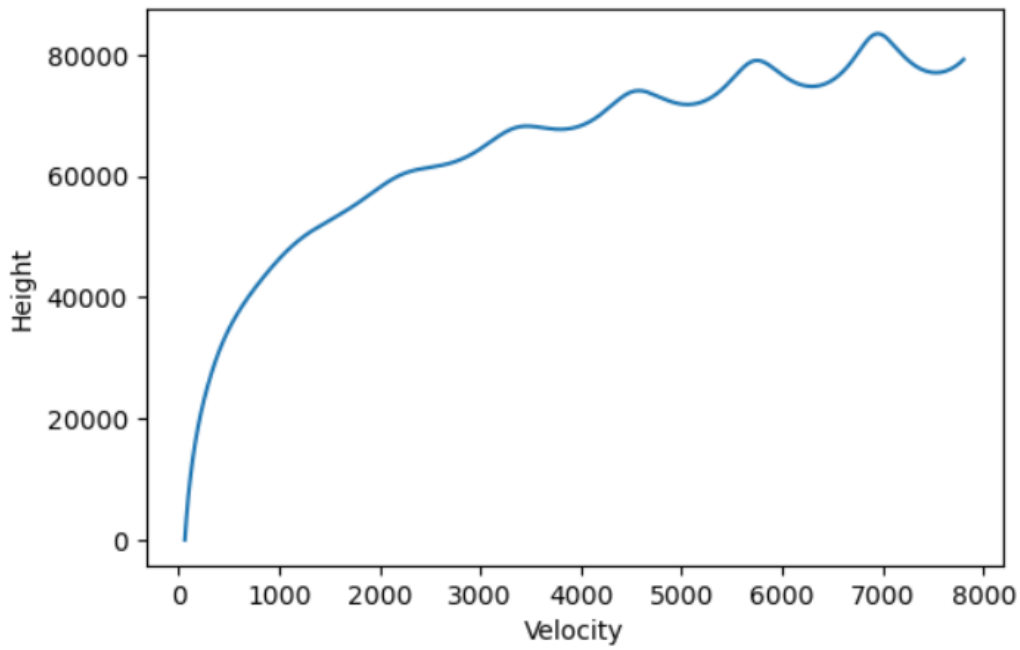

Figure 9: Height (m) v/s Velocity (m/s) for Space Shuttle

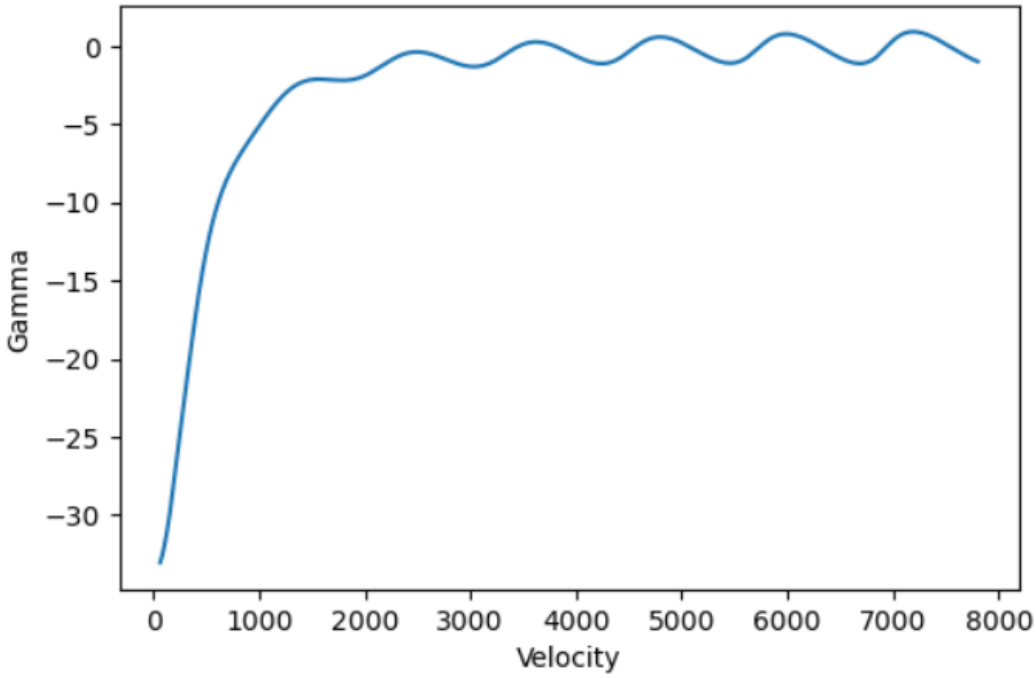

Figure 10: Flight path angle (deg) v/s Velocity (m/s) for Space Shuttle

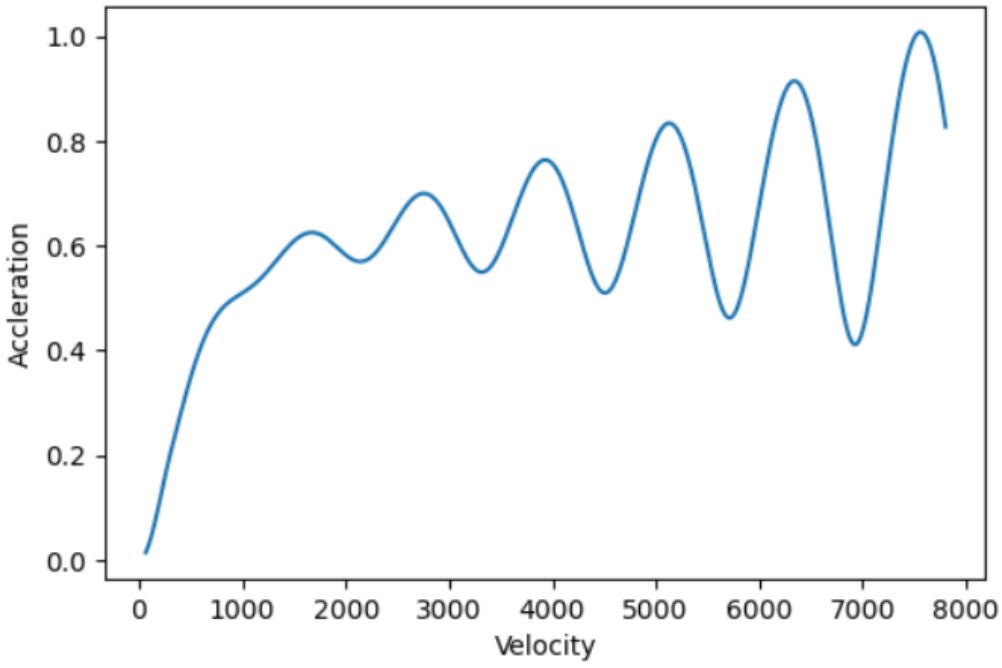

Figure 11: Acceleration (m/s^2) v/s Velocity (m/s) for Space Shuttle

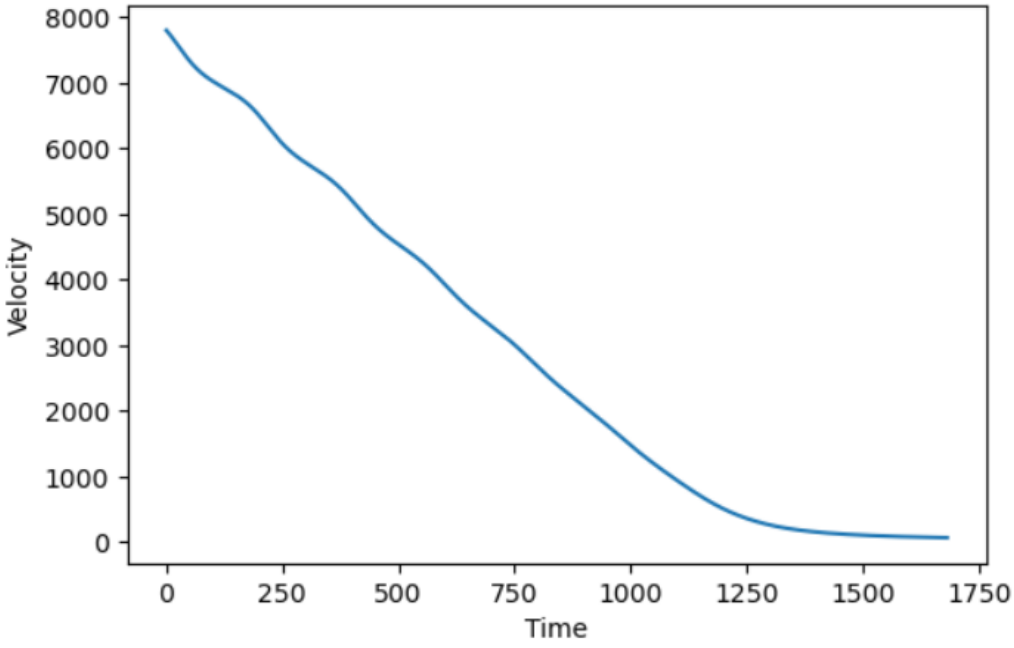

Figure 12: Velocity (m/s) v/s Time (s) for Space Shuttle

These results are compared with the Space Shuttle Reentry Calculations as done by Henrique Ferrolho [6] as shown below:

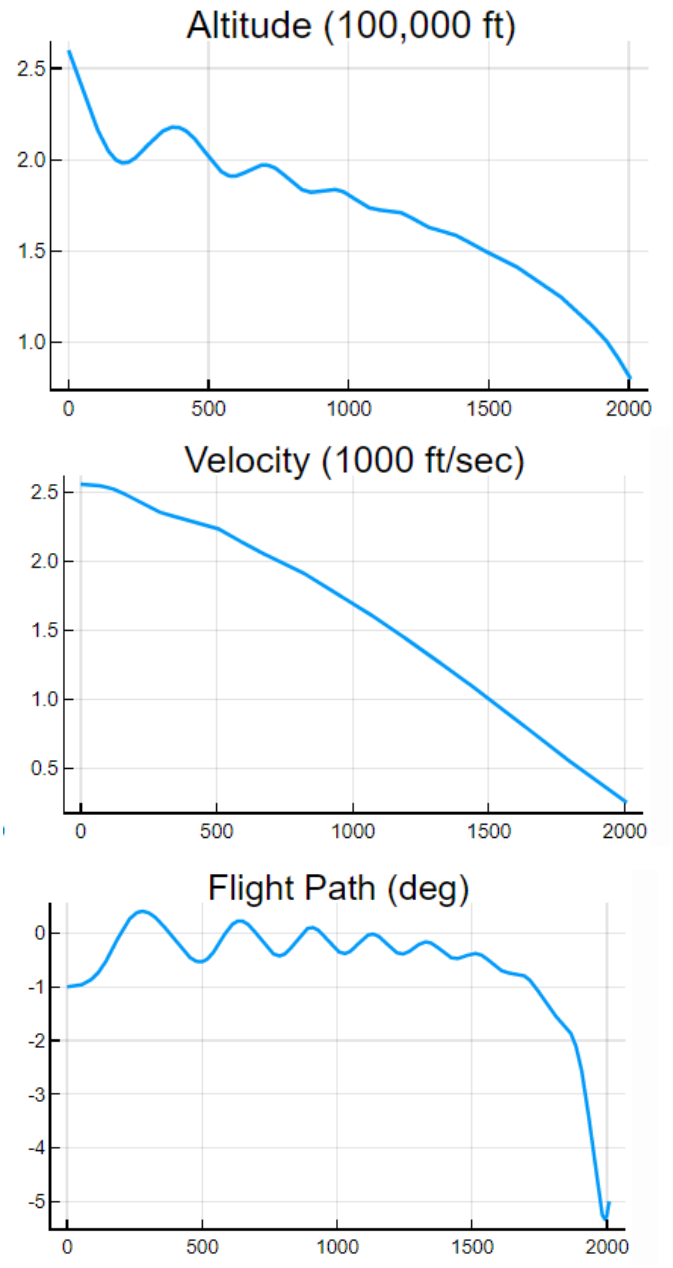

Figure 13: Trajectory plots for Space Shuttle [6].

CHAPTER 5
CONCLUSION

In this paper, a simplified 6-DOF approximation of the equations of motions of a spacecraft were developed and solved for atmospheric reentry trajectories for two spacecrafts – the Stardust Sample Return Capsule and NASA's Space Shuttle. Aerodynamic and other relevant data was extracted from existing literature to ensure most accurate results as possible. The 6-DOF equations were then coded into Python and solved using numerical integration to give relevant trajectory graphs. The results obtained were then compared to existing trajectory simulations of the same spacecrafts as found in previous literature. It was found that despite simplification and decoupling of the 6-DOF equations, the results agreed well with preexisting simulations.

Research in trajectory optimization has come a long way with different models, simulations, and solvers, each aiming to be more robust than previously developed architectures. As mission design is incredibly sensitive to accuracy and precision in trajectory planning, there is always scope for better improving the efficiency and robustness of simulations for the same.

REFERENCES

[1]. Davis, Peter, "Six Degree-of-Freedom Mission Planning for Reentry Trajectories" (2021). Theses and Dissertations. 5066. *https://scholar.afit.edu/etd/5066*.

[2]. Bollino, K., Oppenheimer, M., & Doman, D. (2006). *Optimal Guidance Command Generation and Tracking for Reusable Launch Vehicle Reentry. AIAA Guidance, Navigation, and Control Conference and Exhibit.* doi:10.2514/6.2006-6691.

[3]. Shoemaker, M., Van der Ha, J.C. (2009). *Trajectory estimation of the Hayabusa sample return capsule using optical sensors, 19th Workshop on Astrodynamics and Flight Mechanics.*

[4]. Dilão, R., & Fonseca, J. (2016). *Dynamic Guidance of Gliders in Planetary Atmospheres. Journal of Aerospace Engineering, 29(1), 04015012.* doi:10.1061/(asce)as.1943-5525.0000499.

[5]. Desai, Prasun & Mitcheltree, Robert & Cheatwood, F.. (1999). *Entry Trajectory Issues for the Stardust Sample Return Capsule.*

[6]. https://ferrolho.github.io/blog/2020-05-25/space-shuttle-reentry-trajectory.

APPENDIX

Code

#importing modules

```
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
```

#defining function for numeric integration

```
def ode45(t,sol):
    x = sol[0]
    y = sol[1]
    z = sol[2]
    v = sol[3]
    G = sol[4]
    psi = sol[5]

    print(z)
    g = g_o*(R/(R+z))**2
    rho = rho_o*np.exp(g*(h_ref-z)/Rg/T_o)
    #print(rho)
    L = 0.5*rho*v**2*Cl*Aref
    D = 0.5*rho*v**2*Cd*Aref

    #equations of motions
    dx = v*np.cos(G)*np.cos(psi)
    dy = v*np.cos(G)*np.sin(psi)
    dz = v*np.sin(G)

    dv = -D/m - g*np.sin(G)
    dG = L/m/v - g*np.cos(G)/v
    dpsi = L*np.sin(phi)/m/v/np.cos(G)

    return [dx,dy,dz,dv,dG,dpsi]
```

#terminal condition when Height reaches zero

```
def eve(t,sol):
    return sol[2]
eve.terminal = True
```

#intialising variables

```
x_o = 0.0
y_o = 0.0
z_o = 79248
v_o = 7802.88
G_o = -1*np.pi/180
psi_o = 90*np.pi/180
phi = 0.0
m = 92079.2511
Cd = 0.8
Cl = 1.2
Aref = 249.9092
rho_o = 1.225
T_o = 288
h_ref = 0.0
Rg = 287
R = 6738000
g_o = 9.81
sol_o = [x_o, y_o, z_o, v_o, G_o, psi_o]
t_start = 0
t_end = 100000
```

#solving using RK45 numeric integration method

```
sol = solve_ivp(ode45, [t_start, t_end], sol_o, method = 'RK45', max_step = 0.2, events=[eve])
```

#extracting the values

```
print(sol)
Y = sol.y
#print(Y)
time = sol.t
Height = Y[2]
Velocity = Y[3]
Gamma = Y[4]
```

#calculating acceleration for every data point

```
G_A = g_o*(R/(R+Height))**2
d_a = rho_o*np.exp(G_A*(h_ref-Height)/Rg/T_o)
D_A = 0.5*d_a*Velocity**2*Cd*Aref
A_A =  -D_A/m - G_A*np.sin(Gamma)
```

#plotting figures

```
plt.figure(1,figsize=(6,4), dpi=100)
plt.plot(time,Height)
plt.xlabel('Time')
plt.ylabel('Height')
plt.figure(2,figsize=(6,4), dpi=100)
plt.plot(Velocity, Height)
plt.xlabel('Velocity')
plt.ylabel('Height')
plt.figure(3,figsize=(6,4), dpi=100)
plt.plot(Velocity, Gamma*180/np.pi)
plt.xlabel('Velocity')
plt.ylabel('Gamma')
plt.figure(4,figsize=(6,4), dpi=100)
plt.plot(Velocity, -A_A/9.81)
plt.xlabel('Velocity')
plt.ylabel('Accleration')
plt.figure(5,figsize=(6,4), dpi=100)
plt.plot(time, Velocity)
plt.xlabel('Time')
plt.ylabel('Velocity')
```