A dissertation report on

"Mechanics of Orbit Using Python(MOPy)"

Submitted to

In partial fulfilment of the requirements for the award of degree

Bachelor of Technology
In
Aerospace Engineering

Submitted by:

Ramkiran L.
17030141AE007
lramkiranBTECH17@ced.alliance.edu.in

Manjunath
17030141AE009
manjunathBTECH17@ced.alliance.edu.in

Monisha Patel A.
17030141AE013
pamonishaBTECH17@ced.alliance.edu.in

Thoshitha R. Kumar
17030141AE027
kuthoshithaBTECH17@ced.alliance.edu.in

Under the guidance of
Dr. Gisa G.S.
Assistant Professor
Department of Aerospace Engineering,
Alliance College of Engineering and Design,
Alliance University
Bengaluru.

Department of Aerospace Engineering
Alliance College of Engineering and Design
Alliance University, Bengaluru - 562106
Batch - 2017-21
Year - 2021

CERTIFICATE

This is to certify that **Mr. Ramkiran L. (17030141AE007), Mr. Manjunath (17030141AE009), Ms. Monisha Patel A. (17030141AE012)** and **Ms. Thoshitha R. Kumar (17030141AE027)** students of **Aerospace Engineering, Bachelor of Technology 2017-21** batch at **Alliance College of Engineering and Design (ACED), Alliance University, Bengaluru** has completed the project report titled *"Mechanics of Orbit using Python"* under my guidance in partial fulfillment for the award of Bachelor of Technology degree in Aerospace Engineering, Alliance University, Bangalore during the year 2020-2021.

Prof. Gisa G.S	**Prof. Velmurugarajan K.**
Internal Guide	Head of the Department
Department of Aerospace Engineering	Department of Aerospace Engineering
ACED, Alliance University	ACED, Alliance University
Bengaluru	Bengaluru

Dr. Reeba Korah

Interim Dean
Department of Aerospace Engineering
ACED, Alliance University
Bengaluru

External Viva

	Name of Examiners	Signature with date
1.		
2.		

DECLARATION

We, Ramkiran L, Manjunath, Monisha Patel A, Thoshitha R. Kumar students of 8th Semester Bachelor of Technology in Aerospace Engineering, Alliance College of Engineering and Design (ACED), Alliance University, Bengaluru, hereby declare that the entire project work entitled "**Mechanics of Orbit using Python**" is an authentic record of the work that has been carried out independently by us during final year of our B.Tech at ACED, under the esteemed guidance **Dr. Gisa G.S**, Assistant Professor, Department of Aerospace Engineering, Alliance college of Engineering and Design, Alliance University.

This project report is submitted in partial fulfillment of requirements for the award of the degree of Bachelor of Technology in Aerospace Engineering. The results embodied in this dissertation are original and it has not been submitted in part or full for any degree in any University.

Place: Bengaluru
Date:17/062021

Ramkiran.L
17030141AE007

Manjunath
17030141AE009

Monisha Patel A.
17030141AE012

Thoshitha R. Kumar
17030141AE027

ACKNOWLEDGEMENT

Abstract

Orbital mechanics is the study of the motions of artificial satellites and space vehicles moving under the influence of forces. It plays a vital role in planning space missions in various aspects such as designing orbital trajectory for various missions, calculating lagrangian points etc., and can find many engineering applications including ascent trajectories, reentry and landing, rendezvous computations, lunar and interplanetary trajectories.

In recent years, as there is a lot of progress unfolding in space industry, many aspiring students are keen on gaining knowledge and pursue careers in space industry. As such, strong foundations of the fundamentals are required for the students for them to get ahead in the field. There are lots of tools like STK, FreeFlyer etc., for learning but these don't start from the bare minimum of the concepts. This led to the idea of developing of MOPy.

Mechanics of Orbit using Python(MOPy) is learning tool designed and developed with the purpose of introducing the core concepts of Orbital Mechanics. MOPy uses a interactive UI with tool tips and a 3D Environment with an interactive virtual universe. The 3D Environment helps the learner to visualize the concepts in a much accessible and easier way.

1   Introduction

1.1   About Software

1.2   List of Features

Sl. No	Section
1	Calculation of Orbital Elements
2	2D and 3D orbit
3	Various Parameters at any given point
4	2D and 3D Orbits
5	Calculation of Julian Day
5	Euler Angles
6	Sphere Of Influence
7	Sensitivity Analysis
8	Position of one Spacecraft w.r.t Another
9	Calculation of State and Velocity Vector
10	Orbital Transfer

Table 1: List of Features present in MOPy

1.3   Python Libraries Used

1. **NumPy**: This brings MATLAB like functionality of using Matrices and their operations to python. This enables us to do a lot of stuff without much hassle.

2. **SciPy** - This enables us to add many features involving more complex computing scenarios as it has features for scientific and technical computing. For example, it has different kinds of solvers for integration which we can use for solving acceleration vector equation to obtain the position vector for an orbit.

3. **Matplotlib** - This is a plotting tool that is a extension on NumPy that gives the functionality of plotting many different kinds of graph. This library is somewhat similar to the plotting feature of MATLAB.

4. Panda3D - This is a Game Engine based on C++ that takes in syntax from both C++ and Python. This provides real-time 3D visualizations and simulations based on the code.

5. **SQLite3** - The entire details of the planetary bodies like the orbital elements, planetary ephemeris and others are stored in a local database. SQLite is used as it enables the offline functionality.

6. **Qt Deisgner** - This enables MATLAB's App Designer like feature of dragging and dropping the UI elements and creating the GUI. This is based on Qt, a cross platform GUI toolkit developed by the Qt Company

7. **PyQt5** & **PySide2** - These both are the python binding libraries of Qt.

8. **PyInstaller** - This library lets us convert our python code(.py) into executable file(.exe)

1.4 Market Research

There are mainly two kinds of softwares.

1. Simulation Based programs like STK, FreeFlyer etc.,

2. Sandbox Based programs like Universe SandBox, Kerbal Space Program etc.,

The simulation based programs are mainly used to simulate missions and solve problems based on the instance. Both the applications given in the example are used in the industry for all kinds of missions, ranging from very small scale missions that are performed by the students to complicated missions that are performed by NASA and ISRO.

The Sandbox based programs are the stuff that are run by the physics engine that are baked into it. They use a 3D visualization toolkit or engine which lets the user easily interact with the UI, and change the parameters directly from the environment.

In the analysis done by Morgan Stanley named "Investing in Space Exploration", it is stated as - The revenue generated by the global space industry may increase to more than $1 trillion by 2040. [2]
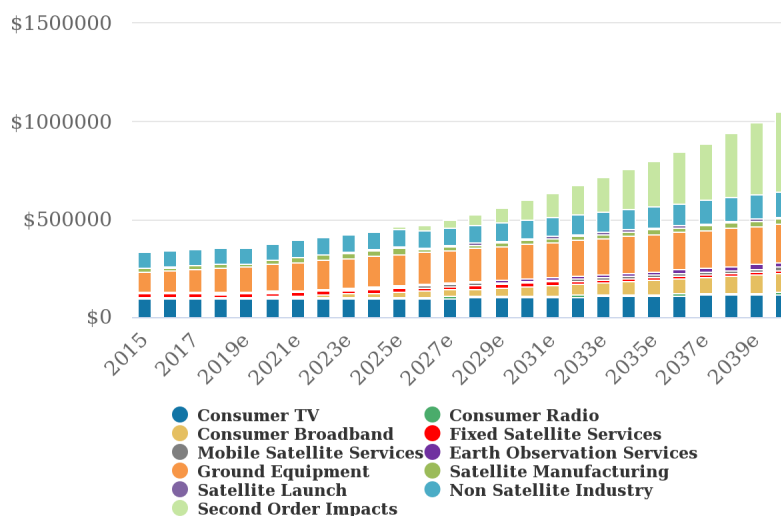

Figure 1: Global Market Trend according to Morgan Stanley.

A report by Antrix and PwC, it is stated that the indian space sector can become a $50 Billion industry, or about one per cent of India's projected $5 Trillion economy, by 2024 from the current $7 billion, according to a report by the Antrix and PwC.[3]
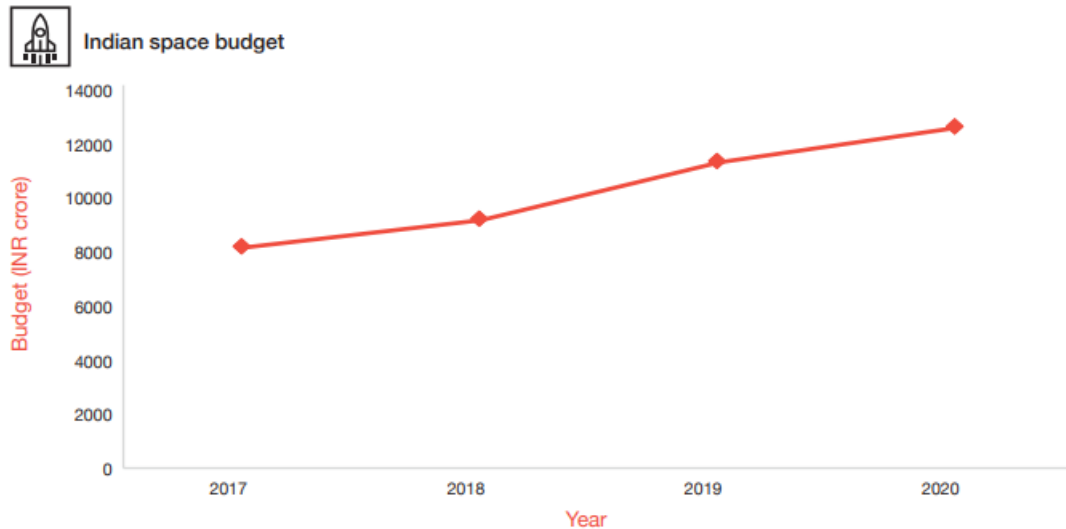

Figure 2: Indian Space Budget over the years.

1.5 Objective

The development in the field of space technology is constantly increasing as shown in the afore-mentioned studies. Such being the case, many students are showing interest to learn more and more about space technology and its related concepts. Considering all such possibilities, we have come up with an idea to develop a learning tool beneficial to learn more about Orbital Mechanics The main objectives of this project are as follows:

1. Design and develop a software to learn concepts and solve Problems related orbital mechanics to understand the basics.

2. Provide a user friendly learning tool such that the user can operate even with the minimum knowledge about the concepts of orbital mechanics.

3. Help user to visualize the virtual view of the space mission.

1.6   Front End Development

1.6.1   Introduction

Front-end development deals with the Graphical User-Interface aspect of the software. It is the key developmental process that defines how the user experiences the features we have developed. The interface between the user and the back end code is GUI. The inputs from the user is taken from the GUI. So the design must be intuitive and clear. There are many libraries that can be used to develop a GUI like PyQt, Pyside, Kivy, Tkinter, etc. In our case, we have opted for PyQt5, Pyside2 and designed GUI in Qt-Designer. Then linked the Back-End scripts through the Integrated Development Environment(IDE) by Microsoft i.e, Visual Studio Code.

1.6.2   Home Page

When the application opens the Home Page will load. In it, there is a Dropdown box containing all the features available, from which the user can choose which feature they want to use. When the user selects any of the features from the drop-down and clicks on the go button at the bottom, they are navigated to that screen where they can use the feature they selected. And then if they want to navigate back to the Home-Page they can click on the Home button provided at the top left corner of the screen. All the features are listed in the upper-mentioned table.

Figure 3: Home Page of MOPy

2 Detailed Explanation of Each Feature
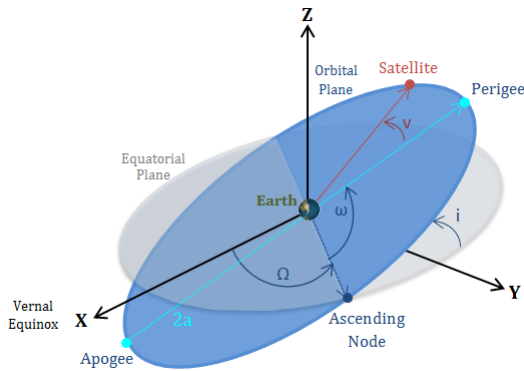
2.1 Calculation of Orbital Elements
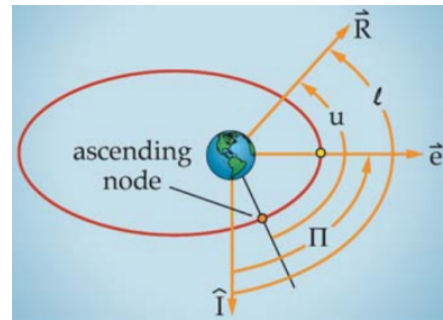
2.1.1 Theory

Figure 4: Classical Orbital Elements

Figure 5: Alternate Orbital Elements

Classical Orbital Elements

1. **Semi-major Axis(a):** This is a constant that defines the size of the orbit. In Circular it is the radius of the circle. It is the longest diameter of an ellipse.

2. **Eccentricity(e):** This the parameter of the conic section which determines its shape. It is defined as the ratio of distance b/w two foci to the Major-axis.

3. **Inclination(e):** It is the tilt of the orbit w.r.t the equatorial plane, measured at ascending node. It can also be defined as the angle from $\hat{K}$ unit vector to the specific angular momentum vector $\vec{h}$.

4. **Right Ascension of Ascending Node(Ω):** RAAN horizontally orients the ascending node of the orbit w.r.t the Equatorial plane's vernal equinox, measured in an equatorial plane This is not defined when the inclination is 0 or 180 degree. It lies between 0 to 360 degrees.

5. **Argument of Perigee(ω):** Argument of Perigee defines the orientation of the ellipse in the orbital plane, as an angle measured from the ascending node to the periapsis measure in the direction of the spacecraft's motion. This is not defined when inclination is 0 or 180 degree or eccentricity is 0. It lies between 0 to 360 degree.

6. **True Anomaly(ν):** True Anomaly defines the position of the spacecraft w.r.t perigee. It's the angle between the spacecraft and the perigee of the orbit. It lies between 0 to 360 degree.

Alternate Orbital Elements

1. **Longitude of Perigee**(Pi) is the angle from the principle direction to perigee. This is used whenever inclination is either 0° or 180° as there is no ascending node. It's lies between 0^0 to 360^0. In the figure(5) Longitude of perigee is represented as "Π".

2. **True Longitude** is the angle from the principle direction to the spacecraft's position. This is used whenever there is no perigee and the the inclination is either 0^0 or 180^0. It's lies between 0^0 to 360^0. It lies between 0 to 360 degree. In the figure(5)True Longitude is represented as "l" In the figure(5)True Longitude is represented as "l" In the figure(5) Longitude of perigee is represented as "Π".

2.1.2 Algorithm

3 Conclusion

4 Future Scope

References

[1] Bate, Roger R., Donald D. Muller and Jerry E. White, *"Fundamentals Of Astrodynamics"*,New York, NY, Dover Publications,1971.

[2] *"Space: Investing in the Final Frontier"*,Jul, 24th, 2020, Retrieved on 25th May 2021 from Morgan Stanley,MorganStanley.com

[3] *"Preparing to Scale New Heights: Privatisation of India's Space Sector"*