

Final Proposal : Solar System & Interstellar Mission

Wei-Hsiang Yu 游惟翔

Department of Physics, National Tsing Hua University, Hsinchu, Taiwan

April 29, 2021

Abstract

This project predicts to improve the solar system simulation based on the program we had done in lecture 7 to make it be able to simulate **a more real solar system** and use it to find **trajectories of interstellar probes** that human had launched to the outer space.

1 Introduction

From the class in lecture 7, professor show his program about a n-body system, which can scatter many points in the window, and those object points are projected or even eaten by a star located in the center of the window. I consider that maybe I can simulate one object projected by multiple stars, and that is gravity assist.

So we need a more accurate solar system to acquire a real circumstance for doing simulation, and based on this circumstance and the launched record that some satellites which had launched into the outer space in human history to test this program is exact or not.

There are two satellites will be predicted to simulate in this project: *Voyager2*[1] & *Galileo*[2]. The first one applied gravity assist to speed up over the escape velocity of the Sun, the later one applied gravity assist to speed up for saving the fuel and speed down to make it could be captured by Jupiter.(See Fig.1)

Finally, if those tests are successful, maybe I will try some other mission's trajectory (like the mission of *Ulysses*, it needed to stop the velocity acquired by launch from earth to 0 but also needed to get the velocity of doing rotation around the Sun to pass the Sun's north and south poles)(See Fig.1(c))

2 Methodology

There are something need to discuss in this part:

- **Destination** / flyby relay planet of our satellite
- Mechanism for program to **find trajectories** (how it choose or skip this time's trajectory)

2.1 Destination / Flyby Planet / Time

So, first we need to determine which destination we want to arrive.

If destination is interstellar, the velocity of satellite need to over the escape velocity of sun(See Fig.2),

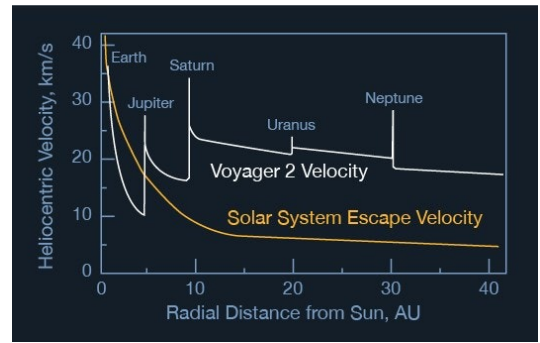


Figure 2: The velocity of voyager2 during mission and the escape velocity of the Sun.[3]

If the destination is the certain planet, we need to make our satellite moving as fast as the revolution velocity of the target planet, then our satellite can be captured by the terminal planet.

And we can also record what planet we want to pass (may be pass them to speed up or pass to speed down. . .) Choose the time as initial launch time to start our simulation. Here I think I will start the launch time before the real launch time of voyager2 maybe 1 month before to find that whether this program will get the same result as voyager2 mission report.

2.2 Instruction Scheme:Find Trajectory

This part will be the subroutine of the update part in the program done during the class. Given the initial condition of the launched time coordinate, we then do **update until the satellite is closed enough the relay planet**.

If the closed enough condition is satisfied, we can then **use the pass position to determine whether this satellite need to be speed up/speed down** or even will collide with the relay planet. In this part the determination is based on the condition where the destination is. If the destination is the interstellar, we hope that the velocity can increase, but if this planet is the terminal we want to arrive, the determination will hope this flyby being a speed down process.

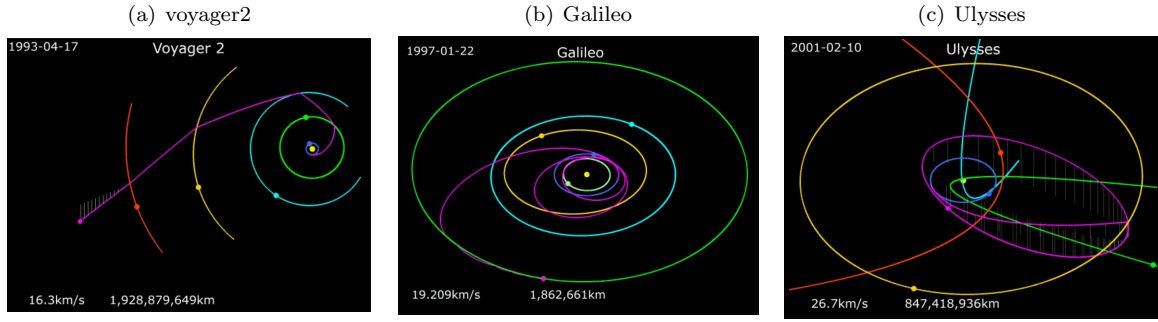


Figure 1: trajectory of three satellite (purple line)

The tendency of changing velocity is relative to **how closed the satellite and planet is** and **where does the satellite pass through the planet.**(See Fig.3)

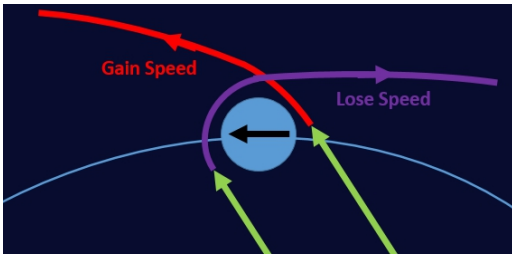


Figure 3: speed up or speed down

Above is the steps in one loop, and will recursive update subroutine until the satellite is satisfied with our destination. But there are still some problems need to discuss in this program, and will mention in the next part.

3 Task & Improvement

There are also some tasks will encounter during build this project:

- Getting the information of the **accurate planet coordinate** in special time.
- Define **how close satellite and relay planet.**
- Decrease the time of finding the flyby opportunity.
- Mechanism of **adjusting satellite direction** during speeding up or down in gravity assist.

3.1 Accurate Planet Coordinate

In section 2 we choose a certain time as our satellite initial launch time, but the relative location of each planet in the solar system is required. I found that JPL(Jet Propulsion Laboratory), a institute of NASA, establishing a data base called "*horizons*"[4] to **provide a accuracy planet coordinate data** of our solar system from 50 years ago.

This accuracy coordinate data also has **been sorted out by a Python package "*poliastro*"**[5] and is able to export to a comprehensible data form.

3.2 Define How Close

During the update part, the condition of how closed the satellite and relay planet is, determinate whether the update do continually or not and the changing tendency of the satellite velocity Respectively. So a good definition of "closed enough" is a important segment in this program.

This "closed condition" can be considered by the escape velocity generated by each planet the satellite needs to pass.

3.3 Decrease spending time

There will be a lot of failures attempts during the update subroutine when our satellite doesn't satisfied the closed condition. Then the loop need to go back to the initial launch time/last departure time and make this time value pulse dt ($time+dt$) to make another simulation ,but this restart step will consume a lot of time.

So I may try to define the determination to find how much distance is the satellite and relay planet is, and base on this information to adjust the time interval value. (i.e. making dt become bigger or smaller base on different conditions)

3.4 Adjusting Direction

Last thing is given a small adjustment ability of our satellite. This will be applied when the satellite is coincided with the specified condition of satisfied **closed enough / pass position.**

By small adjusting direction during gravity assist, we can make the next loop's update become more easier. If the update subroutine fails in the next step, program can first change the adjustment in the last gravity assist instead of restarting the simulation by $time+dt$.

This task maybe also need to consider the fuel consumption when doing adjustment, this will influence the update mass during the later time, and change the trajectory.

References

- [1] Voyager 2 - Planetary Voyage
<https://web.archive.org/web/20131127192310/http://voyager.jpl.nasa.gov/science/planetary.html>
- [2] Galileo - solar system exploration
<https://solarsystem.nasa.gov/missions/galileo/overview/>
- [3] Gravity assist: The simple physics trick that's allowed humanity to explore deep space
<https://blog.jatan.space/p/gravity-assist-the-simple-physics-trick-thats-allowed-humanity-to-explore-deep-space>
- [4] JPL - horizons
<https://ssd.jpl.nasa.gov/horizons.cgi>
- [5] Python package - poliastro
<https://docs.poliastro.space/en/latest/examples/Going%20to%20Jupiter%20with%20Python%20using%20Jupyter%20and%20poliastro.html>
- [6] Gravity_assist
https://en.wikipedia.org/wiki/Gravity_assist