AAS 16-346

PRELIMINARY DESIGN OF TRAJECTORIES FROM EARTH TO JUPITER

Phillip A. Johnson[*], Carlos A. Trivino[†]

Trajectory design is an integral part of mission planning, and there are a theoretically infinite number of possible trajectories between two planets. In this paper, we will leverage computational methods and high-powered computing to determine possible trajectories from Earth to Jupiter which utilize gravity-assists to decrease mission cost. We will find that, with advancements in rocket size or through use of resonant orbits, there are multiple possible trajectories from Earth to Jupiter utilizing intermediate planets.

INCLUDED FILES

In this section is a list of included files and their functionality.

`main.py` - The main script. A sample run can be run with command line arguments "python main.py s", and a large run can be run with command line arguments "python main.py"

`classes.py` - This contains the PlanetDict,Planet,Node,EndNode, and Path classes which will be detailed in later sections

`orbitpropagation.py` - This script is used in main.py to determine flight time and launch opportunities

`bigsim.pkl,bigsim_path.pkl` - These are saved pkl files from a large run. They can be loaded by changing the saved_object boolean in line 399 to True.

INTRODUCTION

The approach to this project is separated into multiple parts. Firstly, Tisserand curves are utilized to understand which planets are reachable from a start position and V_∞. These curves tell us important information, but they most importantly greatly reduce the necessary computation power. Next, root-solving is utilized to find the curve intersections. Knowledge of these intersections allows for use of a search algorithm to find possible paths from a start position. Once the paths are found, the flight time and launch opportunities are calculated.

In the later sections, this paper will detail the methodology behind each step and how the use of MIT's high-powered computation tool, the Supercloud[1], allowed for propagation of a large set of V_∞ values.

[*]Undergraduate, AeroAStro, MIT, 526 Beacon St. Boston, MA 02215
[†]Undergraduate, AeroAstro, MIT, 526 Beacon St. Boston, MA 02215

SUPPLEMENTARY CLASSES

A few classes were created in order to aid with consistency.

Planet

The `Planet` class uses planet data[2] to create objects for each planet that holds information such as the planet's name, mass, distance from the sun, radius, and more.

PlanetDict

The `PlanetDict` class maps a string (the planet name) to the planet object. This ensures that the same planet object is used across the entire script so that any new information stored in it during runs is maintained.

Node

The `node` class is used in path finding and maintains information that can be used to classify which curve it is on and which curve it came from. This object also holds information on if a resonant orbit is needed to reach it or not.

Path

The `path` class contains nodes and allows for easy bookkeeping on each path found.

TISSERAND PLOTTING AND INTERSECTION CALCULATIONS

In order to plot the Tisserand curves, the resulting non-dimensional values of the semi-major axis and eccentricity based on the hyperbolic excess velocity, V_∞ are calculated using the following formulas[3]:

$$a = \frac{1}{||1 - V_\infty^2 - 2 * V_\infty * cos(\alpha)||} \tag{1}$$

$$e = \sqrt{1 - \frac{1}{a} * \left(\frac{3 - \frac{1}{a} - V_\infty^2}{2}\right)^2} \tag{2}$$

Where α is the angle between the spacecraft and the planet before the fly-by. From these non-dimensional values, the perihelion and orbital period can be calculated:

$$r_p = a * r_1 * (1 - e) \tag{3}$$

$$T = \frac{2\pi}{\sqrt{\mu}} * (a * r_1)^{\frac{3}{2}} \tag{4}$$

Where a is dimensionalized by multiplying by the radius between the fly-by planet and the Sun, r_1. Plotting the perihelion as the x-axis and the period as the y-axis yields a Tisserand Plot showing in Figure 1.

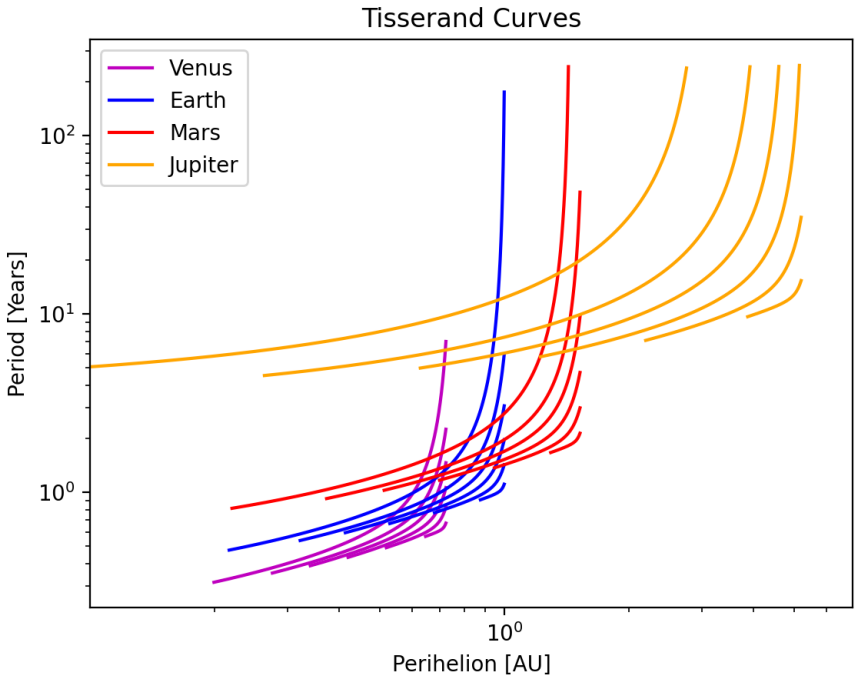

Figure 1. Tisserand Curves for $V_\infty = 1, 3, 5, 7, 9, 12$ km/s

Intersections of this plot are found by calculation the roots of the function:

$$f(\alpha) = T_1(\alpha) - T_2(\alpha) \tag{5}$$

Finding the intersections allows for path-finding.

PATH FINDING

From the α values found from the Tisserand Curves, a Breadth First Search algorithm is utilized to calculate the potential paths from the start planet (Earth).

This algorithm functions simply:

1. Populate the queue with all Earth V_∞ curves

2. For each curve, add to the queue each curve it intersects with, checking that the intersection curve is not the same curve as the one it came from

3. Check if the intersection curve is the goal planet (Jupiter), and append the curve to the queue if not

4. Once the queue is empty, return all found paths

Supercloud Computing

Because search algorithms are computationally-intensive, the MIT Supercloud was leveraged to reduce run-time and allow for longer runs. This allows for extremely large state-spaces of V_∞

values, and we were able to effectively search through curves of V_∞ ranging from 1 to 18 km/s for Earth, Venus, Mars, and Jupiter. Future advancements to the code would include parallel processing of expensive tasks (i.e. root finding, orbit propagation) to improve run-time and allow for more refined ranges of V_∞.

FLIGHT TIME AND LAUNCH OPPORTUNITIES

Flight Time

Knowing the departure V_∞ and α values allows for forward propagation using Kepler's equation of motion to determine the flight time between each segment of the path to Jupiter. The `propogate` method in the `orbitpropogation.py` script uses `odeint` from SciPy to forward propagate until the desired radius is reached. From this, the position of the spacecraft at intersection and time of flight is also extracted, which will be used for finding launch opportunities. For each leg of the trip, (i.e. each pair of (departure planet, arrival planet)), the angle travelled $\theta_{travelled}$ is also recorded as it will be utilized in the following section.

Launch Opportunities

In order to find possible launch dates for a given path, we first need to find the relative angles between each pair of planets in the path. To do this we calculate

$$\Theta_{req} = \theta_{arr} - \theta_{dep}$$

Where θ_{arr} is the required position of the arrival planet in order for our spacecraft to intercept and flyby it and θ_{dep} is the initial angle of the departure planet before flying the current leg of the trip. While this value of Θ_{req} is good to know, we want to know what the required angle between each planet will be at the time of initial earth escape. This requires slight modifications of the following form

$$\theta_{arr,i} = \theta_{arr} - \omega_{arr} * T_{total}$$

$$\theta_{dep,i} = \theta_{dep} - \omega_{dep} * T_{beginning \longrightarrow curr}$$

where T_{total} is the total time of flight **including** the current leg of the path and $T_{beginning \longrightarrow curr}$ is the total time of flight **excluding** the time required to complete the current leg of the trip. This finally yields

$$\Theta_{req,i} = \theta_{arr,i} - \theta_{dep,i}$$

$$= \theta_{arr} - \omega_{arr} * T_{total} - (\theta_{dep} - \omega_{dep} * T_{beginning \longrightarrow curr})$$

$$= \theta_{travelled} - \omega_{arr} * T_{total} + \omega_{dep} * T_{beginning \longrightarrow curr}$$

Since we recorded our $\theta_{travelled}$ in the previous section and we know the other parameters, we have enough information to know the required angles at launch. For each (departure planet, arrival planet) pair in our path, θ_{req} is computed and recorded. Finally, in order to actually find dates that satisfy these launch conditions we propagate the angular positions of each planet forward in time. If, for a given time value, the first leg of the trip satisfies its angle requirement, the program checks if the next leg of the trip also satisfies its own angle requirement, and if so moves on to the next leg over and over. If all legs are satisfied, it returns the time value (in years) which will be our earliest departure date for a given flyby trajectory. If even one of the paths does not satisfy its requirements, however, it will keep iterating through time values until either finding a possible launch date, or hitting the max launch window which is discussed more in the next section.

CONSTRAINTS

Before detailing the results, there are some constraints that were made based on current possibilities. The maximum current C3 possible (according to the NASA High Energy Performance Calculator[4]) is 10 km/s. Thus, constraints were placed such that paths with a velocity higher than 10 km/s leaving Earth were not considered. Additionally, for a safe fly-by the maximum interplanetary change in α is detailed as:

$$\Delta\alpha = 2sin^{-1}\frac{1}{(1+(r_p+200km)(\frac{v_\infty^2}{\mu}))} \tag{6}$$

This equation is how we determined whether or not a resonant orbit will be needed in order to reduce the instantaneous $\Delta\alpha$ due to a gravity-assist.

Additionally, we limited the launch opportunity window to within 200 years.

RESULTS

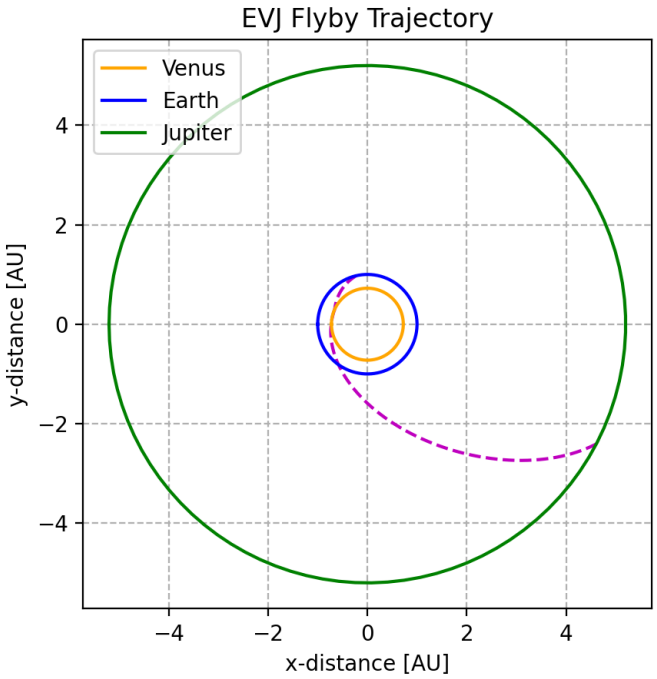

Figure 2. Non-resonant EVJ trajectory for $V_{\infty,E} = 9$ km/s, $V_{\infty,V} = 12$ km/s

Non-Resonant Paths

The only non-resonant path leaving Earth at a $V_\infty \leq 9$ km/s is EARTH9 to VENUS12 to JUPITER. The minimum flight time for this mission would be 677.1 days, and the earliest launch opportunity would be 2053.358. This mission is simulated in Figure 2.

There are more possible non-resonant paths, but they would require significant advancements in rocket size. One such path would be EARTH9 to MARS15 to JUPITER, which would only take 502.2 days and have a launch opportunity of 2090.59.

More non-resonant paths are detailed in the attached file `nonresonant_paths.xlsx`

Resonant Paths

It is worth looking at resonant paths as well as this will allow us to determine low-ΔV paths. With resonant orbits, the V_∞ values will be unchanged, but there is added complexity of determining the correct altitude of the fly-by as well as the correct phasing. That being said, certain resonant paths are desirable due to the low fuel cost.

Shown here are examples of three minimum-energy trajectories which require a resonant orbit (E = Earth, V = Venus, M = Mars, J = Jupiter):

Planet Path	$V_{\infty,launch}$ [km/s]
EVEMJ	$2.8 \rightarrow 4 \rightarrow 4 \rightarrow 3$
EVEVJ	$2.8 \rightarrow 4 \rightarrow 7 \rightarrow 11$
EMJ	$4 \rightarrow 3$

As seen in the table, resonant orbits can be incredible useful in gravity-assists. As an example, the Earth $\rightarrow$ Mars $\rightarrow$ Jupiter path is impossible to do with a safe altitude. However, adjusting the altitude so that a safer turning angle can occur, and entering a resonant orbit will greatly reduce the required ΔV cost at the cost of greater complexity and likely very rare launch opportunities. It is also worth noting that, in a path like this, we would likely choose to fly by Earth twice instead of Mars as Earth is more massive and will have a larger effect on the turning angle.

The file `allpaths.xlsx` contains information of every single path calculated. Unfortunately, this file does not contain information about launch opportunities and flight time, but it does show just a sample of how many paths could be feasible with the inclusion of resonant orbits.

FUTURE IMPROVEMENTS

There are a couple shortcomings with this project. The largest of them is the fact that the resonant orbits were not modeled. Future improvements would include modeling resonant orbits to find the optimal flyby altitude on first and second pass of the planet which will also give a better idea of mission requirements and cost.

Another potential shortcoming is the speed. Although the code runs in polynomial time, there are many improvements which could be made in terms of multiprocessing and faster root-solving (especially with time of flight calculations).

REFERENCES

[1] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas, "Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis," *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 2018, pp. 1–6, 10.1109/HPEC.2018.8547629.

[2] "Properties of the Solar System," `https://hbcp.chemnetbase.com/faces/documents/14_02/14_02_0001.xhtml`.

[3] D. Palma, "Preliminary Trajectory Design of a Mission to Enceladus," *Tecnico Lisboa*, Dec 2016.
[4] "Launch Vehicle Performance Website," https://elvperf.ksc.nasa.gov/Pages/Query.aspx.

1234