Low-Thrust Interplanetary Trip to Saturn Using Ion Thrusters

Team 447
Problem A

November 8, 2020

Abstract

With the latest expanses in the electrical propulsion systems, it is now possible to think about the interplanetary low-thrust travel with the ion thrusters. It is called low thrust because the acceleration that given by this engine's force is very small compared to other forces but its sustainability comes from long time usage of the engine. In this paper, we carefully investigate the feasibility of Earth to Saturn travel only using ion thrusters. The problem is the optimum control of the thrust to reach Saturn with the maximum mass and minimum time elapsed. After implementing a solar system simulator we first investigate the escaping from the earth gravitational field using different methods of thrust control and after we escape from the earth we use gravity assist from Jupiter to get faster and reach Saturn. The propulsion system applies constant thrust in the direction of the orbital velocity to increase the semimajor axis at the fastest rate. After the flyby to Jupiter, the thrusters are fired back to slow down so that minimizes the velocity relative to Saturn at arrival, thus facilitating the orbit insertion. We show that with our proposed method, it's possible to reach Saturn in approximately 8 years with a loss of 1.3 tons of fuel.The amount of fuel we propose is feasible compared to mass of the spaceshuttle being send.

1 Introduction

Interplanetary travel will be crucial for human life in the future, however, the efficiency and feasibility of such flights is very low if we want to reach distant planets. Because interplanetary distances are so big compared to escape velocity from the earth and the fuel needed to accelerate the spacecraft to such big velocities is a crucial problem. Low-thrust acceleration methods gained huge attention in recent years because of the potential of reducing the payload mass to a huge extent compared to chemical propellers. Ion thrusters play a big role here since its fuel consumption is low and its continuous thrust force can be utilized to find more effective trajectories than chemical fuels. Optimization of low and continuous thrust is a different and more complex problem than short impulse optimization since in the latter we can compute the trajectories before and after the engines firing with very good precision. However, in the continuous thrust case, we need numerical integration to find trajectories that are computationally more expensive than the direct calculation of orbits.[6]

In the process of interplanetary travel techniques such as gravity assist [7] which uses other planets to gain momentum are used. And optimization techniques such as Reinforcement Learning [8] which employs machine learning using the simulations of planets are also used in the most recent years.

The core objective of this paper is to replace a spacecraft from an orbit around the earth to an orbit around Saturn while utilizing the low thrust Ion engines. While conducting this operation, some constraints are proposed. Controlling the ion thruster is definitely a crucial factor. A moderate amount of flight time should be concerned and finally, the amount of fuel is also available to some extent. Modeling of this problem has to adequately take into account all of these constraints.

Symbol	Meaning	Numerical Values
G	Gravitational constant	6.67x10^{-11}m$^3 kg^-1 s^-1$
g	Gravitational acceleration at sea level	9.8 ms^-2
M_s	Mass of the Sun	1.98x$10^{30} kg$
M_E	Mass of the Earth	5.97x$10^{24} kg$
M_{saturn}	Mass of the Saturn	5.86x$10^{26} kg$
$M_{jupiter}$	Mass of the Jupiter	1.89x$10^{27} kg$
V_{saturn}	Velocity of the Saturn	9680 m/s
$V_{jupiter}$	Velocity of the Jupiter	13007 m/s
r_m	Distance to the moon	384.400 km
I_{sp}	Specific impulse of thruster	4000 s
M_{thi}	Initial Mass of the rocket	5000 kg
T_i	Initial period around earth	90 m
T_f	Final period around Saturn	40 h

1.1 Problem Statement

problem definition can be stated as following: Given the constant force of the thrusters find the control function on the direction of force applied so that our spacecraft can maintain a trajectory from Earth orbit to Saturn orbit. We need to make a few assumptions to be able to calculate and simulate the trajectory.
Assumptions:

- We assume the motions of the bodies are Keplerian orbit. Therefore, we can assume all the motions in the 2D plane and simplify our equations.
- We assume that our rocket is under the influence of a single force field at a time. Meaning that until we escape from Earth only gravitational force acting on the rocket is Earth and after we escape from it, the only acting force field is the Sun.
- We assume that after we escape from Jupiter using a gravitational assist we are no longer close to any significant force field so we assume our trajectory is linear between Jupiter and Saturn.

With these assumptions since there are only two forces (gravitational force and thruster force) acting on the body we can have a model and equation of motions as following;

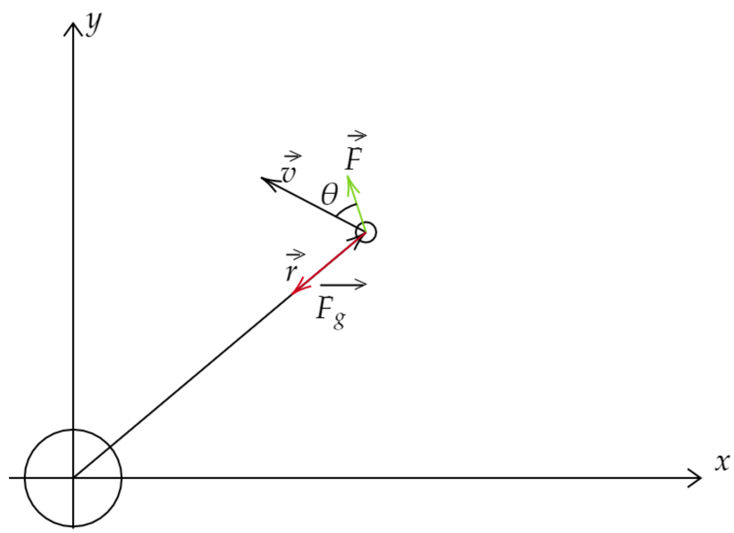

Figure 1: Free body diagram of the system. Only two forces are assumed to be interacting with the spacecraft: Gravitational force and thrust force.

$$M_{th}(t)\frac{d^2\vec{r}}{dt^2} = F_{th}\frac{\vec{r}}{|r|} - GM_E M_{th}(t)\frac{\vec{r}}{|r|^3} \tag{1}$$

$$\frac{d^2\vec{r}}{dt^2} = \frac{F_{th}}{M_{th}(t)}\hat{\theta} - GM_E\frac{\vec{r}}{|r|^3} \tag{2}$$

We assume mass of the rocket is constantly decreasing with α as the consequence of constant fuel consumption.

$$M_{th}(t) = M_0 - \alpha t \tag{3}$$

In terms of state equations, we define our state s as:

$$r = \begin{bmatrix} x \\ y \end{bmatrix}, \quad s = \begin{bmatrix} r \\ \dot{r} \\ M_{th}(t) \end{bmatrix} \in \mathrm{R}^5 \tag{4}$$

And our control variable a as:

$$a = \begin{bmatrix} \theta \end{bmatrix}, \quad \theta \in [0, 2\pi] \tag{5}$$

Our objective is the find the control on a that will lead us to our objective location in space.

2 Methods

2.1 Ion Thruster

An ion thruster is a device that is widely used to take benefits from electric repulsion for spacecraft to travel interplanetary or even to change orbital positions. The charged particles are ionized and forced to lose their electrons to get positive ions then these electrons are accelerated mainly using basic Lorentz force. The spacecraft given in the question is equipped with ion thrusters that can provide a constant thrust of 400 millinewtons, with a specific impulse of 4,000 seconds. The main equation describing the ion thruster is given by :

$$F_{th} = 2\frac{\eta P}{g I_{sp}} \tag{6}$$

And:

$$I_{sp} = \frac{F_{th}}{\alpha g} \tag{7}$$

where:

- η is the efficiency of the engine
- g is the acceleration at earth's sea level
- P is the electrical power used by the thruster in W
- I_{sp} is the specific impulse in seconds.
- α is the flow rate of fuel mass[1]

We found $\alpha = 10^-5$ but we preferred to use its half value for real life application version. Because we need to take into account the geometry corrections, temperature changes etc. [2]

2.2 Gravity Assist Maneuver

Gravity Assist Maneuver is a brilliant way to change the speed, path, and direction of the spacecraft. In fact, it is the usage of relative movement of the spacecraft when they are passing near a planet or a star. The basic idea is the same as the elastic collision of two bodies. In the reference frame of the planet, spacecraft should flee the planet with the same approach velocity. For escaping from the earth in our case, we use the moon for taking slingshot to increase the speed of the spacecraft. For instance, the Voyager 2 used four times these maneuvers to escape from the solar system. Especially Jupiter played a huge role in that operation to make the spacecraft's velocity increased substantially.

$$|\vec{V_{ri}}| = |\vec{V_{thi}} - \vec{V_m}| = \sqrt{(V_{thi}^2 + V_m^2 - 2V_{thi}V_m cos(\alpha))} \tag{8}$$

$$|\vec{V_{rf}}| = |\vec{V_{thf}} - \vec{V_m}| = \sqrt{(V_{thf}^2 + V_m^2 - 2V_{thf}V_m cos(\beta))} \tag{9}$$

$$|\vec{V_{ri}}| = |\vec{V_{rf}}| \tag{10}$$

$$V_{thf} = V_m cos(\beta) + \sqrt{(V_m cos(\beta))^2 + V_{thi}^2 - 2V_{thi}V_m cos(\alpha)} \tag{11}$$

For this equation, α is the angle between the first velocity of the thruster with the planet whereas β is the final angle. As it indicates V_{thf} is increasing with the low and positive β and higher α.

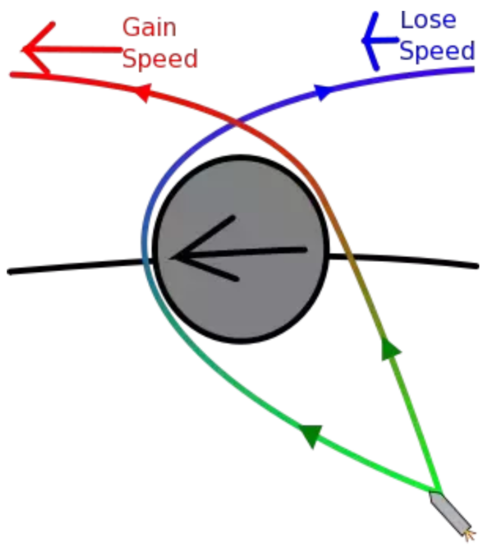

Figure 2: How to increase the speed and decrease the speed by using positive or negative leaving angles respectively. In addition to that approach angle should not be small to gain speed as indicated in the figure. [4]

2.3 Flight Stages

2.3.1 Stage 1 (Escaping From the Earth)

Stage 1 consists of escaping from Earth via the ion thruster engine that leads the velocity to increase as time goes. It is necessary to optimize this time by controlling the thruster. We evaluate three options. The first way is to keep the engine always back so that force is always parallel to the velocity vector of the spacecraft. The second way is keeping the engine always in the radial direction as a similar concept to solar sailing like focusing on the Earth. The third way is to combine these two methods by estimating the direction of the sum of velocity and radial vectors. Which again seems rational because always the velocity and the distance from earth should increase.
Let us further analyze all of the three methods by finding the equations of motions and simulating them. First we need the initial and final velocities of circular motions that 90 minutes period around the Earth and 40 hours period around the Saturn.

then the inside the root is as follows.:

$$\frac{GM_{planet}M_{th}(t)}{r_i^2} = \frac{M_{th}(t)V_{thi}^2}{r_i} \tag{12}$$

$$\frac{2\pi r_i}{V_{thi}} = T_i \tag{13}$$

$$\frac{2\pi GM_{planet}}{T_i} = V_{thi}^3 \tag{14}$$

Then $V_{thi} = 7740\frac{m}{s}$ and $V_{thf} = 11820\frac{m}{s}$ are the initial and final velocities of the spacecraft respectively.

First way: Force is in the direction of velocity for all time EQM:

$$F_{th}\frac{\frac{d\vec{r}}{dt}}{|\frac{d\vec{r}}{dt}|} - \frac{GM_E M_{th}(t)\vec{r}}{|r|^3} = M_{th}(t)\frac{d^2\vec{r}}{dt^2} \tag{15}$$

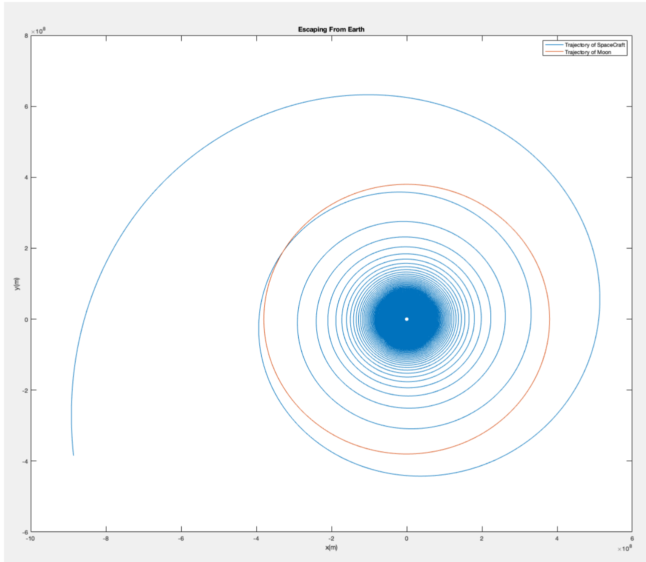

Figure 3: (Red line stands for the moon orbit Blue line describes spacecraft. It was assumed circular. Origin is the location of Earth.) This is the best way to leave the earth because of the direction of the force(albeit it is very small). At 150 R_E it is assumed to left the earth. [3]

Second way: Engine kept radially inwards for all time.EQM:

$$F_{th}\frac{\vec{r}}{|r|} - \frac{GM_E M_{th}(t)\vec{r}}{|r|^3} = M_{th}(t)\frac{d^2\vec{r}}{dt^2} \quad (16)$$

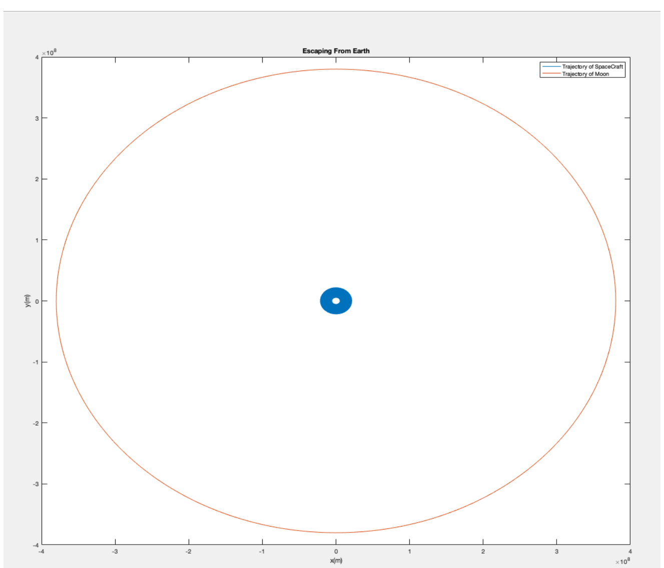

Figure 4: (Red line stands for the moon orbit Blue line describes spacecraft. It was assumed circular.Origin is the location of Earth.) These methods work to some extent and fail at that point. Spacecraft can not find enough propulsion to accelerate itself and leave the Earth.

Third way: Engine kept both radially inward and in the direction of velocity for all time.let c be the vector:

$$c = \frac{\vec{r}}{|r|} + \frac{\frac{d\vec{r}}{dt}}{|\frac{d\vec{r}}{dt}|} \tag{17}$$

$$F_{th}\frac{\vec{c}}{|c|} - \frac{GM_E M_{th}(t)\vec{r}}{|r|^3} = M_{th}(t)\frac{d^2\vec{r}}{dt^2} \tag{18}$$

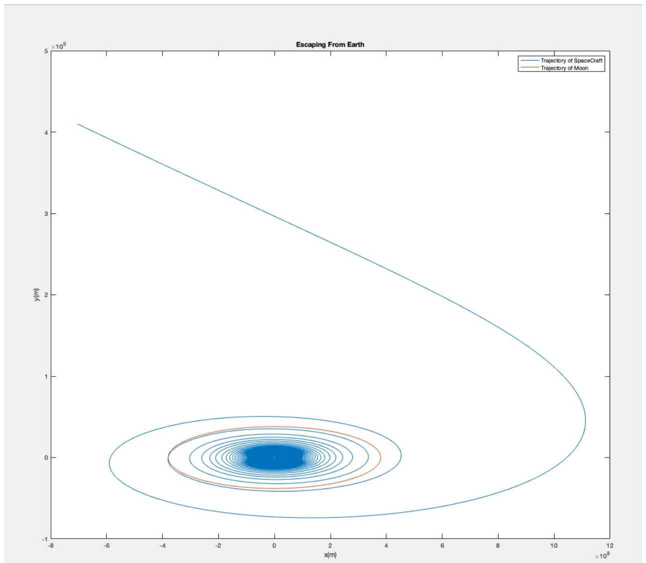

Figure 5: (Red line stands for the moon orbit Blue line describes spacecraft. It was assumed circular.Origin is the location of Earth.) In this process again approximately 1.7 years the direction of the force is in the same direction with the velocity then it is changed to a radially inward direction.

2.3.2 Passing from Stage 1 to Stage 2 (Exploring new ways)

Let us write some of the necessary and important results coming from stage 1.Angle between moon and the spacecraft(it can be deduced from the graph that it is very small.) $\alpha_{mbs} = 4°$. As radius increased by time goes the spiral shape of the orbit of the spacecraft has not differed much from the circular shape. At the end of the stage 1 $t_{stage1} = 5.909.10^7 s$ and the velocity of the spacecraft is $V_{thi} = 694\frac{m}{s}$. The main questioning in this process is exploring whether it is possible to take gravitational slingshot from the moon to reach escape velocity in a very short time interval. Escape velocity near the moon can be calculated by equalizing total energy to zero.

Then it can be found by very easy formula given follows.:

$$V_{escm} = \sqrt{\frac{2GM_E}{r_m}} = 1.44\frac{km}{s} \tag{19}$$

Then let us consider whether this lower limit can be obtained with the gravity assist formula as we know the angle of approaching.

$$V_{thf} = V_m cos(\beta) + \sqrt{(V_m cos(\beta))^2 + V_{thi}^2 - 2V_{thi}V_m cos(\alpha)} \tag{20}$$

Note that moon velocity is given $V_m = 1022\frac{m}{s}$ and at that point velocity of the spacecraft is about $V_m = 1044\frac{m}{s}$ $\frac{V_{thi}}{V_m} = 1.02$ then the inside the root is as follows:

then the inside the root is as follows.:

$$\sqrt{cos(\beta))^2 - 0.994)} \tag{21}$$

In that equation β forced to equal to 0.Which leads no change in the velocity. Then this Gravity assist method is not applicable for us at this stage.

2.3.3 Stage 2

Stage 2 is the process that spacecraft experiences after it escapes from the Earth gravitational field until it reaches Jupiter. The aim here is to reach Jupiter as soon as possible and make the Gravity Assist Maneuver in there. As the gravitational effect of other planets is quite weak, we will only take the gravitational effect of the sun. We will consider the origin of our coordinate system as the center of the sun. Equation of the motion of the spacecraft under these assumptions:

$$-\frac{GM_sM_{th}(t)}{r(t)^3}\vec{r}+\vec{F}=M_{th}(t)\vec{r''} \tag{22}$$

Here $\vec{r}$ is the position vector of our spacecraft. Equation of motion of the Jupiter under these assumptions:

$$-\frac{GM_sM_{jup}}{r_{jup}^3}\vec{r_{jup}}=M_{jup}\vec{r''_{jup}} \tag{23}$$

Here r_{jup} is the position vector of Jupiter.Let's define the angle between the spacecraft's velocity vector and the direction of the thrust force as θ. If we write vector $\vec{F}$ in terms of velocity vector $\vec{r'}$ and θ:

$$\vec{F}=|\vec{F}|\frac{\vec{r'}}{|\vec{r'}|}R_\theta \tag{24}$$

Here R_θ is rotation matrix in 2-dimension:

$$R_\theta=\begin{bmatrix}cos(\theta) & -sin(\theta)\\ sin(\theta) & cos(\theta)\end{bmatrix} \tag{25}$$

After writing the set of equation of motions, we simulated it and obtained the trajectories of spacecraft and Jupiter with varying θ angles.We obtained the travel time and the velocity of the spacecraft just before making gravity assist maneuver with Jupiter. Trajectories when $\theta=0$:

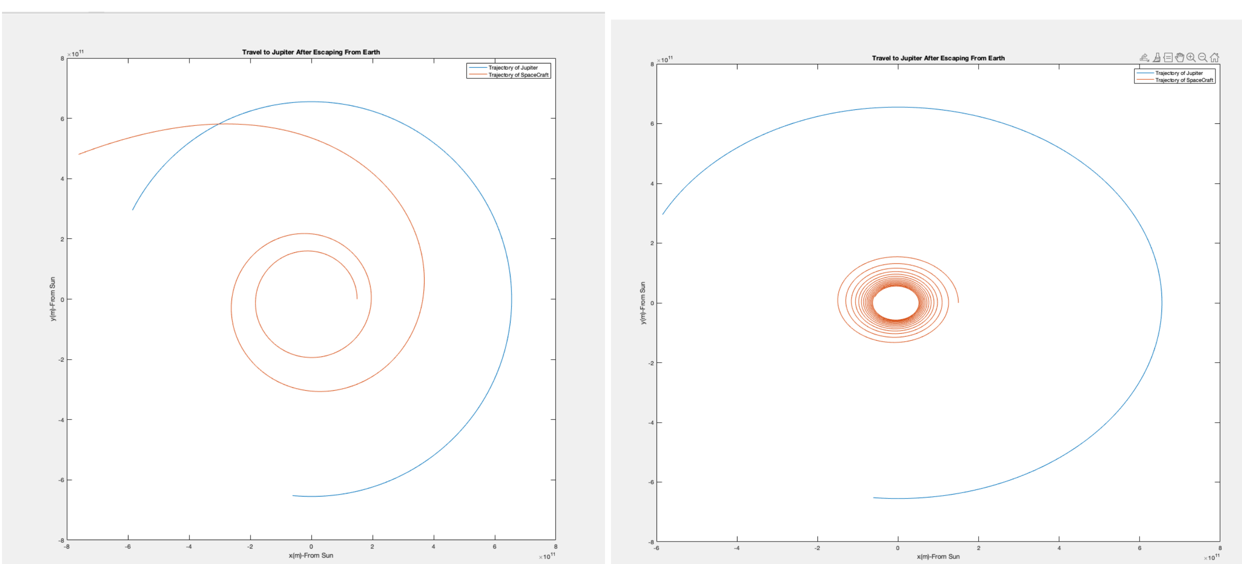

Figure 6: Trajectories when $\theta=0$ on the left and $\theta=\pi$ on the right.Blue line represents the trajectory of Jupiter and red line represents the trajectory of spacecraft. When $\theta=\pi$, spacecraft moving to the direction of sun

When we increase the angle θ until some value we can reach Jupiter, however travel time increases in that cases. It is impossible to reach Jupiter for θ beyond a certain values. Trajectories when θ is different from zero:

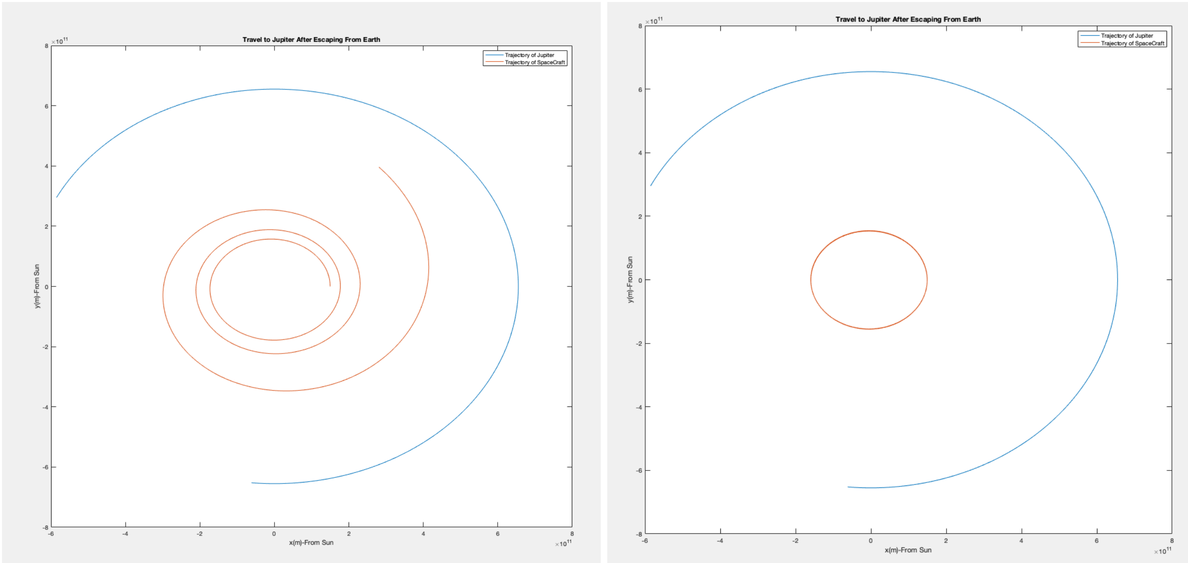

Figure 7: Trajectories when $\theta = \pi/4$ on the left and $\theta = \pi/2$ on the right.Blue line represents the trajectory of Jupiter and red line represents the trajectory of spacecraft. In the case $\theta = 0$, the spacecraft has already reached Jupiter, but has not reached when $\theta = \pi/4$. When $\theta = \pi/2$ spacecraft stuck in a circular motion

As we can see from the simulation results, in order to reach Jupiter as soon as possible after escaping from the gravitational field or Earth, we must apply thrust force along the way parallel to the speed of the spacecraft and in the direction that will increase the speed($\theta = 0$ case). Under the conditions that we explained above, we obtained the following results from the simulation on Matlab:

$$t_2 = 5.42 years \tag{26}$$

Here t_2 is the duration of the trip to Jupiter after escaping from Earth. And the mass of fuel that spacecraft use in this journey is $855kg$. The speed of spacecraft with respect to Sun's reference frame just before the gravitational assist maneuver is:

$$V_2 = 16.60 km/s \tag{27}$$

Gravitational Assist Maneuver with Jupiter: To be able to compute the speed of spacecraft after the maneuver we have to know the initial and final angles between the velocity vectors of Jupiter and spacecraft as we mentioned above. From the simulation results, we found the initial angle between the velocity vectors of spacecraft and Jupiter as $\alpha = 28.68$ in degrees. We can assume that the final angle between the velocity vectors of spacecraft and Jupiter as $\beta = 0$. If we use (10)

$$V_{2f} = V_{jup} cos(\beta) + \sqrt{(V_{jup} cos(\beta))^2 + V_{2i}^2 - 2V_{2i} V_{jup} cos(\alpha)} \tag{28}$$

Here $V_{2i} = 16.60$ km/s is the velocity of spacecraft with respect to Sun reference frame just before the maneuver, V_{2f} is the speed of spacecraft with respect to Sun reference frame just after the maneuver and the $V_j up = 14.04 km/s$ speed of Jupiter.α and β angles are mentioned above. All of these results are taken from Matlab simulation. If we compute the final speed of spacecraft:

$$V_{2f} = 22.02 km/s \tag{29}$$

It means that we can increase velocity of spacecraft $V_{2i} = 16.60$ km/s to $V_{2f} = 22.02$ km/s with a gravity assist maneuver. We named the remaining part of the journey to Saturn as Stage 3

2.3.4 Stage 3

In this stage of our flight, we are exiting from Jupiter at a very high speed.$V_{2f} = 22.02 \frac{km}{s}$ We know that $V_{Jupiter} = 13.070 \frac{km}{s}$ and escape velocity at a distance nearest the Jupiter can be found directly from multiplying the velocity of Jupiter with the square root of 2 so:

$$V_{escJ} = \sqrt{2} V_{Jupiter} = 18.48 \frac{km}{s} \tag{30}$$

which is lower than the velocity of the spacecraft. This means that after taking additional velocity from Jupiter remaining trajectory to Saturn can be assumed as linear. So the remaining equation of motion and initial values are independent of the sun.

$$-F_{th}\frac{\vec{r}}{|r|} = M_{th}(t)\frac{d^2\vec{r}}{dt^2} \tag{31}$$

$$M_{thi} = 3900kg \tag{32}$$

$$r_i = R_{saturn} - R_{jupiter} = 6.46x10^1 1\frac{m}{s} \tag{33}$$

$$M_{th}(t) = M_{thi} - \alpha t \tag{34}$$

$$\frac{d\vec{r}}{dt} = 22.02\frac{km}{s} \tag{35}$$

When we search solution via Matlab the final values of the velocity and mass are obtained as follows:

$$t_{stage3} = 3.16x10^7 s \tag{36}$$

$$M_{thf} = 3692kg \tag{37}$$

$$V_{2f} = 18.66\frac{km}{s} \tag{38}$$

All left is adjusting the velocities such that they correspond to the desired orbit velocity because this V_{2f} is relative to the sun. The angle between this vector and the Saturn velocity vector should be $\theta = 33.22$

3 Results

We carefully divided the problem into three stages: Escaping from earth, using gravitational assist from the Jupiter and reaching to Saturn. In order to find elapsed time and lost mass during the flight we implemented the simulation in the MATLAB and run it until convergence with the Saturn. Here are the results for every stage:

Quantity	Stage 1	Stage 2	Stage 3	Total
Δt (years)	1.87	5.42	1.00	8.29
Δm (kg)	295	855	158	1308

The journey takes a total of 8.29 years and a total of 1308kg of fuel is consumed. The remaining fuel can be used to return to the Earth. The control mechanism for stage 1 is eagerly discussed in three different ways. The most efficient way is keeping force in the direction of the velocity. For stage 2 control mechanism was carefully evaluated with four different angle values. To enable us to do that, rotation matrices were defined. Again the most efficient way is the $\theta = 0$ case can be seen from the simulations. In stage 3, due to the high velocity gained from Jupiter's assist force was directed back to make spacecraft slower and reduce its velocity to feasible values for orbiting around Saturn.

4 Conclusions

We divide the trip to Saturn into three stages. In the first stage, we aimed to escape from the gravitational field of Earth. To achieve this within a minimal time, we tried to apply the thrust force at different angles. Firstly we aimed to reach the Moon first to make a gravity assist maneuver with Moon, however, we realized that it was impossible to perform such a maneuver. After escaping from Earth, we begin the second stage. In this stage, we aimed to reach Jupiter to make a gravity assist maneuver with it. In this journey, we tried to vary thrust forced angles and simulated these cases. After gaining a velocity from this maneuver we begin stage 3. Because we gained a huge speed from the maneuver in stage 2, we assumed that the remaining route of spacecraft from Jupiter to Saturn would be linear. In this stage, we applied thrust force in the backward direction to reduce the speed of the spacecraft. This is because the speed required for the spacecraft to fit in an orbit in Saturn should be less than the speed at the beginning of stage 3.

Strengths of Our Solution

- We just used approximately one-quarter of our mass all the way from Earth to Saturn

- Our travel time turned out to be close to similar space missions that have happened in the past.[5]
- We were able to observe how the results would change by controlling the thrust force with our simulations.

Weakness of Our Solution

- We did not optimize the trajectory of Jupiter to Saturn. We relied on assumptions for the location of the Saturn and linearity of that trajectory.
- The effects of other planets were neglected during all processes we took into account one celestial body in our equations.
- Due to the complexity of the calculation we did not try to calculate what should be the initials values are.

References

[1] John E. Foster, Tom Haag, Michael Patterson, and Jr. Williams, George J. The high power electric propulsion (hipep) ion thruster. *40th Joint Propulsion Conference and Exhibit*, pages 12–13, 09 2004.

[2] Dan M. Goebel and Ira Katz. *Fundamentals of Electric Propulsion: Ion and Hall Thrusters*. JPL SPACE SCIENCE AND TECHNOLOGY SERIES, 2008.

[3] Paul W. Keaton. Low-thrust rocket trajectories. *Los Alamos National Laboratory*, pages 9–10, 03 1987.

[4] Kerbal Space Program Wiki. Tutorial: Gravity Assist. `https://wiki.kerbalspaceprogram.com/wiki/Tutorial:_Gravity_Assist`, 2013.

[5] Nasa. Tutorial: Voyager1 Timeline. `https://voyager.jpl.nasa.gov/mission/timeline/#event-a-once-in-a-lifetime-alignment`, 2020.

[6] Sean Tang and Bruce A. Conway. Optimization of low-thrust interplanetary trajectories using collocation and nonlinear programming. *Journal of Guidance, Control, and Dynamics*, 18(3):599–604, 1995.

[7] Chit Hong Yam, Thomas McConaghy, Kuan-Hua Chen, and James Longuski. Design of low-thrust gravity-assist trajectories to the outer planets. *International Astronautical Federation - 55th International Astronautical Congress 2004*, 1:476–491, 01 2004.

[8] Alessandro Zavoli and Lorenzo Federici. Reinforcement learning for low-thrust trajectory design of interplanetary missions, 2020.

5 Appendix

5.1 Codes

```
%%--------------------Code for Stage1---------------------------

%defining constants:
d_0 = 6650000;
global t_f1
t_f1 = 50000000;
global t_f2
t_f2 = 20000000;
t_interval = 1;
global F_mag
F_mag = 400e-3;
global m_0
m_0 = 5000;
global G
```

```
G = 6.67e-11;
global M
M= 5.97e24;
global alpha
alpha = 5e-6;
R_moon = 3.8e8;
global beta_1
beta_1 = pi/2;
global beta_2
beta_2 = pi/4;

%defining initial conditions
r_0 = [0 d_0];
r_0dot = [sqrt(G*M/d_0) 0];
y_0 = [r_0 r_0dot];
F_0 = [0 F_mag];

[t,y] = ode23t(@eq2, [0:1000:t_f1], y_0);
[t z] = ode23t(@eq2prime,[t_f1:1000:t_f2+t_f1],y(end,:));

dtot = [y;z];

th = 0:pi/100:2*pi;
x_moon = R_moon * cos(th) ;
y_moon = R_moon * sin(th) ;
plot(-y(:,1),y(:,2))
hold on
plot(x_moon, y_moon)
legend('Trajectory of SpaceCraft','Trajectory of Moon')
xlabel('x(m)')
ylabel('y(m)')
title('Escaping From Earth')

% finding the intersection time with moon
t_stage1=0;
for i = 1: 1 :length(y(:,1))
    if(  (332997000-0.5e6)<y(i,1) &&  y(i,1)<(332997000+0.5e6) &&
       (183066000-0.5e6)<y(i,2) && y(i,2)<(183066000+0.5e6) )
        t_stage1 = i*1000;
        break;
    end
end
index = i;
```

```
%---------Code for Stage2---------------------
R_E = 1.496e11; % in meters
R_sat = 6.55e11; % in meters
phi = -pi*0.47;
T_sat = 11.86*365*24*60*60; %in sec
t_f2 = 1.715e8;

global M_sun
M_sun = 1.989E30;  % in kg
global G
global m_0
global alpha
```

```
global theta
theta = 0;

%time passed at the first stage:
t_1 = 59.09e6; %in sec.

%mass at the beginning of stage 2:
global m_1
m_1 = m_0 - alpha*t_1;

%initial velocities:
V_esc = 30.480e3;
V_sat = sqrt(G*M_sun/R_sat);

%initial conditions:
r_i =[-R_E,0];
r_sat_i = [-R_sat*cos(phi),R_sat*sin(phi)];
r_0dot_i = [0 V_esc];
r_satdot_i = [-V_sat*sin(phi),-V_sat*cos(phi)];

%initial vectors:
q_1i = r_i;
q_2i = r_0dot_i;
q_sat1i = r_sat_i;
q_sat2i = r_satdot_i;

q_i = [q_1i,q_2i];
q_sat_i = [q_sat1i,q_sat2i];

%Solving ODE'S:
[t,q_sat] = ode23t(@Stage2_ODESat, [0:1e5:t_f2], q_sat_i);
[t,q] = ode23t(@Stage2_ODE, [0:1e5:t_f2], q_i);


plot(q_sat(:,1),q_sat(:,2))
hold on
plot(-q(:,1),q(:,2))
legend('Trajectory of Jupiter','Trajectory of SpaceCraft')
xlabel('x(m)-From Sun')
ylabel('y(m)-From Sun')
title('Travel to Jupiter After Escaping From Earth')
```

```
%----------Code for Stage 3---------------------

t_stage1 = 59.09e6;
t_stage2 = 1.715e8;

global alpha
global m_0
global m_3
m_3 = m_0 - alpha*(t_stage1+t_stage2);

t_up = 1;
t_down =3.165e7;

global m_3down_i
```

```
m_3down_i = m_3 - alpha*t_up;

V_assisted = 22.02e3;

%Accelaration Forward:
a_i = [0,V_assisted];
[t,a] = ode45(@Stage_3Up, [0 t_up], a_i);

%Accelaration BAckward:
b_i = a(end,:);
[t,b] = ode45(@Stage_3Down, [0 t_down], b_i);
```

```
function [dadt] = Stage_3Up(t,a)
    global F_mag
    global m_3
    global alpha

    dadt = zeros(1,2);
    dadt(2) = F_mag/(m_3-alpha*t);
    dadt(1) = a(2);
    dadt = transpose(dadt);
end
```

```
function [dbdt] = Stage_3Down(t,b)
    global F_mag
    global m_3down_i
    global alpha

    dbdt = zeros(1,2);
    dbdt(2) = -(F_mag/(m_3down_i-alpha*t));
    dbdt(1) = b(2);
    dbdt = transpose(dbdt);
end
```

```
function [dqdt] = Stage2_ODE(t,q)
    global F_mag
    global G
    global M_sun
    global m_1
    global alpha
    global theta

    R_theta = [cos(theta),-sin(theta);sin(theta),cos(theta)];
    dqdt = zeros(1,4);
    dqdt(3:4) = -(G*M_sun/mag(q(1:2))^3)*q(1:2)+( F_mag/( mag(q(3:4))*(m_1-
        alpha*t) ) )* transpose(transpose(q(3:4))* R_theta);
    dqdt(1:2) = q(3:4);
    dqdt = transpose(dqdt);


end
```

```
function [dq_satdt] = Stage2_ODESat(t,q_sat)
    global G
    global M_sun

```

```
    dq_satdt = zeros(1,4);
    dq_satdt(3:4) = -( G*M_sun/mag(q_sat(1:2))^3 )*q_sat(1:2);
    dq_satdt(1:2) = q_sat(3:4);
    dq_satdt = transpose(dq_satdt);
 end
```

```
function [dqdt] = Stage2_ODE(t,q)
    global F_mag
    global G
    global M_sun
    global m_1
    global alpha
    global theta

    R_theta = [cos(theta),-sin(theta);sin(theta),cos(theta)];
    dqdt = zeros(1,4);
    dqdt(3:4) = -(G*M_sun/mag(q(1:2))^3)*q(1:2)+( F_mag/( mag(q(3:4))*(m_1-
        alpha*t) ) )* transpose(transpose(q(3:4))* R_theta);
    dqdt(1:2) = q(3:4);
    dqdt = transpose(dqdt);


end
```

```
%--------function for finding a magnitude of a vector-------
function [magnitude] = mag(v)
    n = length(v);
    magnitudesq = 0;
    for i = 1 : 1 : n
        magnitudesq = magnitudesq + v(i)^2;
    end
    magnitude = sqrt(magnitudesq);
end
```

```
function [dydt] = eq2prime(t,y)
    global F_mag
    global G
    global M
    global m_0
    global alpha


    c = y(1:2) / mag(y(1:2)) + y(3:4)/mag(y(3:4));


    dydt = zeros(1,4);
    dydt(3:4) = F_mag*c /((m_0-alpha*t)*mag(c)) - G*M*y(1:2)/mag(y(1:2))^3;
    dydt(1:2) = y(3:4);
    dydt = transpose(dydt);

end
```

```
function [dydt] = eq1(t,y,mt,m)
    global F_mag
    global G
    global M
```

```
    global m_0
    global alpha
    global t_f



    F=[0 F_mag]
    dydt = zeros(1,4);
    dydt(3:4) = y(1:2)/mag(y(1:2))*(F/(m_0-alpha*t) - G*M/mag(y(1:2))^2);
    dydt(1:2) = y(3:4);
    dydt = transpose(dydt);

end
```

```
function [dydt] = eq2(t,y)
    global F_mag
    global G
    global M
    global m_0
    global alpha
    global beta_1

    dydt = zeros(1,4);
    dydt(3:4) = F_mag*cos(beta_1)*y(3:4)/((m_0-alpha*t)*mag(y(3:4))) + F_mag*
        sin(beta_1)*y(1:2)/((m_0-alpha*t)*mag(y(1:2))) - G*M*y(1:2)/mag(y(1:2)
        )^3;
    dydt(1:2) = y(3:4);
    dydt = transpose(dydt);

end
```