Optimal Brachistochrone Spacecraft Controls

AE 504 Final Project | Linyi Hou

May 11, 2020

1 Motivation

The goal of this project is to formulate an optimal control strategy for interplanetary spaceflight using high-thrust, high-specific impulse spacecraft, colloquially known as "torchships".

In science-fiction, torchships are spacecraft equipped with advanced propulsion technology that allow them to constantly accelerate at multiple g's for days or even weeks. This significantly reduces the time required for interplanetary travel. The trajectory the spacecraft follows minimizes travel time by using maximum thrust for the entire flight, and is thus know as a brachistochrone trajectory.

However, there is not much discussion on the detailed operations of torchships. Often times the flight profile is described simply as "flip-and-burn", where the spacecraft accelerates toward its target for half the journey, and decelerates for the other half, as shown in Figure 1.

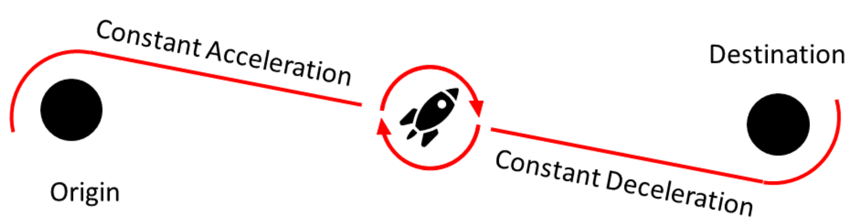


Figure 1: The "flip-and-burn" maneuver

The flip-and-burn maneuver is a simplification of the spacecraft's trajectory. It does not account for the difference in velocity vectors, the effects of gravity, or the thrust vector of the spacecraft. In this project, we will consider all three of the above and attempt to derive an optimal control strategy for brachistochrone spaceflight.

2 Previous Work

As mentioned in Section 1, little literature exists on brachistochrone trajectories for torchships. However, certain elements of the problem have been studied extensively.

The original brachistochrone problem was formulated by Johann Bernoulli in 1696 [1]. In Bernoulli's brachistochrone problem, a particle travels on a path between two points under a gravity field and no friction; the objective was to find the path that minimizes travel time.

Furthermore, continuous thrust trajectories have been analyzed and used substantially in the past. In this project we will be referencing several authors including Professor Prussing [2]. Continuous thrust trajectories are almost always low thrust and seek to minimize fuel consumption, so their applicability to the brachistochrone problem can sometimes be limited.

In the subsequent chapters, we will attempt to consolidate the brachistochrone and continuous thrust trajectories into one optimization problem.

3 Nomenclature

Scalars	H	Hamiltonian
	J	cost functional
	K	final cost
	L	running cost
	t	time
	μ	gravitational parameter of the central body
	Γ	thrust magnitude
Vectors	$\boldsymbol{f}$	equation of motion
	$\boldsymbol{g}$	gravity vector
	$\boldsymbol{r}$	position vector
	$\boldsymbol{v}$	velocity vector
	$\boldsymbol{x}$	state vector
	$\boldsymbol{u}$	thrust unit vector
	$\boldsymbol{\lambda}$	Lagrange multiplier
Matrices	$\boldsymbol{G}$	gravity gradient matrix
	$\boldsymbol{\Phi}$	transition matrix
Subscripts	0	initial state
	f	final state
	max	maximum value

4 Assumptions

Let us consider a spacecraft traveling between Earth and Mars. We assume that restricted two-body dynamics apply, so the spacecraft is only subject to the gravitational pull of the Sun, while the Sun remains stationary. Gravity is modeled by the inverse-square law:

$$\boldsymbol{g} = -\frac{\mu}{r^3}\boldsymbol{r} \tag{1}$$

and thus the equation of motion (EOM) that the spacecraft follows is expressed as

$$\dot{\boldsymbol{r}} = \boldsymbol{v}\ ,\ \ \dot{\boldsymbol{v}} = \boldsymbol{a} = \Gamma\boldsymbol{u} + \boldsymbol{g} \tag{2}$$

To create a first order system, define a state vector and its equation of motion

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \end{bmatrix}\ ,\ \dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \Gamma\boldsymbol{u} + \boldsymbol{g} \end{bmatrix} \tag{3}$$

For convenience, rewrite Equation (3) as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{4}$$

Furthermore, define an upper bound to the thrust available:

$$\max\Gamma = \Gamma_{max} \tag{5}$$

5 Problem Statement

Out goal is to minimize time-of-flight between two locations in an inverse-square gravitational field using continuous thrust, with fixed initial and final positions and velocities.

The running cost is $L = 1$, and there is no final cost K. Therefore the cost functional

$$J(\boldsymbol{u}) = \int_{t_0}^{t_f} L(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\ dt + K(t_f, \boldsymbol{x_f}) \tag{6}$$

can be simplified to

$$J = \int_{t_0}^{t_f} 1\ dt = t_f - t_0 \tag{7}$$

The spacecraft is subject to initial and final state constraints as well as EOM constraints:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \boldsymbol{x}(t_0) &= \boldsymbol{x_0} \\ \boldsymbol{x}(t_f) &= \boldsymbol{x_f} \end{aligned} \tag{8}$$

and a thrust limit

$$|\Gamma| \leq \Gamma_{max} \tag{9}$$

Summarizing, the optimization/minimization problem is therefore defined as

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u}} \quad & t_f - t_0 && \text{(10a)} \\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), && \text{(10b)} \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_0, && \text{(10c)} \\
& \boldsymbol{x}(t_f) = \boldsymbol{x}_f, && \text{(10d)} \\
& |\Gamma| \leq \Gamma_{max} && \text{(10e)}
\end{aligned}$$

6 Solution Approach

6.1 Hamiltonian

We can define the Hamiltonian as follows:

$$\begin{aligned}
H(\lambda, \boldsymbol{x}, \boldsymbol{u}) &= L + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\
&= 1 + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})
\end{aligned} \tag{11}$$

We can further dissect the Lagrange multiplier λ. Let

$$\boldsymbol{\lambda}^T = \left[\boldsymbol{\lambda}_r^T,\ \boldsymbol{\lambda}_v^T\right] \tag{12}$$

Expanding Equation (11) using Equations (3) and (12), we get

$$H = 1 + \boldsymbol{\lambda}_r^T \boldsymbol{v} + \boldsymbol{\lambda}_v^T \left(-\frac{\mu}{r^3}\boldsymbol{r} + \Gamma \boldsymbol{u}\right) \tag{13}$$

with optimality conditions for the free time, fixed final state problem listed as follows:

$$\begin{aligned}
\boldsymbol{0} &= \frac{\partial H}{\partial \boldsymbol{u}} && \text{(14a)} \\
\dot{\boldsymbol{\lambda}} &= -\frac{\partial H}{\partial \boldsymbol{x}} && \text{(14b)} \\
0 &= H && \text{(14c)}
\end{aligned}$$

6.2 Solving for the Control Input

Since the Hamiltonian is linear in $\boldsymbol{u}$ as shown in Equation (14a), we cannot set $\frac{\partial H}{\partial \boldsymbol{u}} = \boldsymbol{0}$ and expect a viable solution. Instead, we must minimize H.

Observe that the only term that $\boldsymbol{u}$ influences is $\boldsymbol{\lambda}_V^T \Gamma u$, so we are able to choose $\boldsymbol{u}$ to always minimize that term. Therefore, choose $\boldsymbol{u}$ such that:

$$\boldsymbol{u} = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \tag{15}$$

By extension of Equation (15), we know that the term $\boldsymbol{\lambda}_V^T \Gamma u$ is always negative. Thus the choice of Γ that minimizes H is the maximum value, $\Gamma = \Gamma_{max}$.

6.3 Solving for the Lagrange Multiplier

We can compute $\dot{\boldsymbol{\lambda}}$:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}} = -\begin{bmatrix} \boldsymbol{O}_3 & \boldsymbol{G} \\ \boldsymbol{I}_3 & \boldsymbol{O}_3 \end{bmatrix} \boldsymbol{\lambda} \tag{16}$$

where the gravity gradient matrix $\boldsymbol{G}$ can be computed from Equation (17) [2].

$$\begin{aligned} \boldsymbol{G} &= -\frac{\partial H}{\partial \boldsymbol{r}} \\ &= \frac{\mu}{r^5}\left(3\boldsymbol{r}\boldsymbol{r}^T - r^2 \boldsymbol{I}_3\right) \end{aligned} \tag{17}$$

We can express $\boldsymbol{\lambda}$ using the transition matrix as follows:

$$\boldsymbol{\lambda}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{\lambda}(t_0) \tag{18}$$

Using results from [3], the transition matrix can be expressed in truncated form as

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{I}_3 + \boldsymbol{G}(\boldsymbol{r}_0)\dfrac{\delta t^2}{2!} & \boldsymbol{I}_3 \delta t + \boldsymbol{G}(\boldsymbol{r}_0)\dfrac{\delta t^2}{3!} \\ \boldsymbol{G}(\boldsymbol{r}_0)\delta t & \boldsymbol{I}_3 + \boldsymbol{G}(\boldsymbol{r}_0)\dfrac{\delta t^2}{2!} \end{bmatrix} \tag{19}$$

where $\delta t = t - t_0$. This expression is limited by the growth of truncated terms over time. The maximum allowable time interval is also given by [3]:

$$\delta t_{max} = \frac{r}{2v} \tag{20}$$

In the context of interplanetary flight in the solar system, the value of δt_{max} evaluates to approximately 30 days for an Earth departure orbit. A rough estimate of the minimum Γ_{max} value required fulfill this time limit can be found using the "flip-and-burn" method.

The distance d flown using the "flip-and-burn" method at acceleration Γ_{max} for time t is

$$d = \frac{1}{4}\Gamma_{max} t^2 \leq \frac{1}{4}\Gamma_{max} {\delta t_{max}}^2 \tag{21}$$

note that the coefficient is $\frac{1}{4}$ instead of $\frac{1}{2}$, because deceleration is required for the latter half of the flight. Rearranging the above to solve for Γ_{max}:

$$\Gamma_{max} \geq \frac{4d}{{\delta t_{max}}^2} \tag{22}$$

where d is the separation between the origin and the destination. Substituting in the maximum separation of Earth and Mars yields $\Gamma_{max} \geq 0.23 m/s^2$.

6.4 The Optimization Problem

Now that we have established the conditions for optimality in the context of brachistochrone trajectories, it is possible to solve the optimization problem using computational tools.

Since $\boldsymbol{\lambda}(t)$ is known, the control $\Gamma \boldsymbol{u}$ at any time t is also known. It is therefore possible to integrate an initial guess, $\boldsymbol{\lambda}_0$, to an unknown final time t_f, and check whether the final state converges to the desired final state. The formulation is as follows:

$$\begin{aligned}
\text{Solve for: } & [\boldsymbol{\lambda}_0 \ , \ t_f] \\
\text{with dynamics: } & \boldsymbol{u}(t) = [\boldsymbol{I}_3 \ \boldsymbol{0}_3] \cdot \boldsymbol{\lambda}(t) \\
& \boldsymbol{\lambda}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{\lambda}_0 \\
& \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\
\text{and I.C.s: } & \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\
& \boldsymbol{\lambda}(t_0) = \boldsymbol{\lambda}_0 \\
\text{and B.C.s: } & \boldsymbol{x}(t_f) = \boldsymbol{x}_f \\
& H(t_0) = 1 + \boldsymbol{\lambda}_0^T \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0) = 0
\end{aligned} \tag{23}$$

For a 3-D case, there are 7 variables and 7 boundary conditions. Note that the last boundary condition $H(t_0) = 0$ comes from Equation (14c).

Since state $\boldsymbol{x}$ cannot be expressed analytically due to the changing $\boldsymbol{g}$ vector, the above system must be integrated numerically. In Section 7, we will describe our solution to the above problem in detail.

7 Solution

The formulation shown in Equation (23) was solved using the MATLAB function fsolve() in combination with numerical integration using the 4-stage Runge-Kutta method. We will consider the baseline case of an Earth-Mars brachistochrone transfer at constant 1g thrust. For purposes of demonstrating functionality in three dimensions, the inclination of Mars is significantly increased. B.C.s are shown in Table 1.

	x	y	z	vx	vy	vz
Origin	151×10^9	0	0	0	30×10^3	0
Target	120×10^9	203×10^9	55.5×10^9	-17.6×10^3	13.5×10^3	8.11×10^3

Table 1: Baseline Case Boundary Conditions

7.1 Numerical Method

The 4-stage Runge-Kutta (RK4) method is used to integrate the equations of motion. The RK4 method is outlined as follows:

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{x}(t_k) + \frac{1}{6}\Delta t\left(\boldsymbol{y}_1 + 2\boldsymbol{y}_2 + 2\boldsymbol{y}_3 + \boldsymbol{y}_4\right) \tag{24}$$

where

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{f}(\boldsymbol{x}(t_k)) \\ \boldsymbol{y}_2 &= \boldsymbol{f}(\boldsymbol{x}(t_k) + \frac{1}{2}\Delta t\boldsymbol{y}_1) \\ \boldsymbol{y}_3 &= \boldsymbol{f}(\boldsymbol{x}(t_k) + \frac{1}{2}\Delta t\boldsymbol{y}_2) \\ \boldsymbol{y}_4 &= \boldsymbol{f}(\boldsymbol{x}(t_k) + \Delta t\boldsymbol{y}_3) \end{aligned} \tag{25}$$

The RK4 method was chosen for its 4th order truncation error convergence rate with respect to the time step Δt. This means we can obtain an accurate solution without spending too much computation time. However, it should be noted that using RK4 integration will produce a small error in the final solution, which means the resulting control strategy will also be affected. We do not expect this to have any noticeable effect on the final solution when using a time step of 200 seconds in the context of an interplanetary transfer. Furthermore, the solution can always be refined in practice after the problem is solved.

7.2 Trust-Region Solver

MATLAB uses the trust-region method in fsolve() to iteratively solve the optimization problem. A trust-region method uses a model function to approximate the original non-linear objective function within a trust-region centered about the current best solution. The method then steps forward to a point in the direction of steepest descent (in the model function), after which the objective function is reevaluated at the new point. If the objective function decreases noticeably, then the model function is considered to be a good representation of the original objective function, and the trust-region is expanded. Conversely, the trust region shrinks. A 2-D visualization of the above process is shown in Figure 2.

To initialize the trust-region algorithm, an initial guess must be provided. It was found that

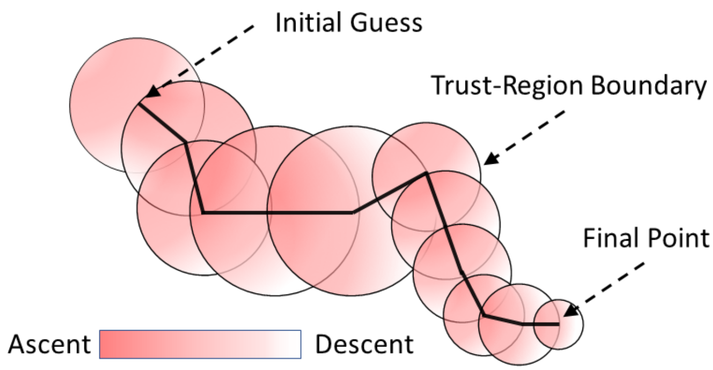


Figure 2: Trust-region optimization

most initial guesses would cause the algorithm to arrive at a local minimum where the trust-region would continuously shrink while the constraints are not fulfilled. A working initial guess was experimentally determined.

8 Results and Discussion

The resulting trajectory and controls are shown in Figures 3 and 4, respectively.

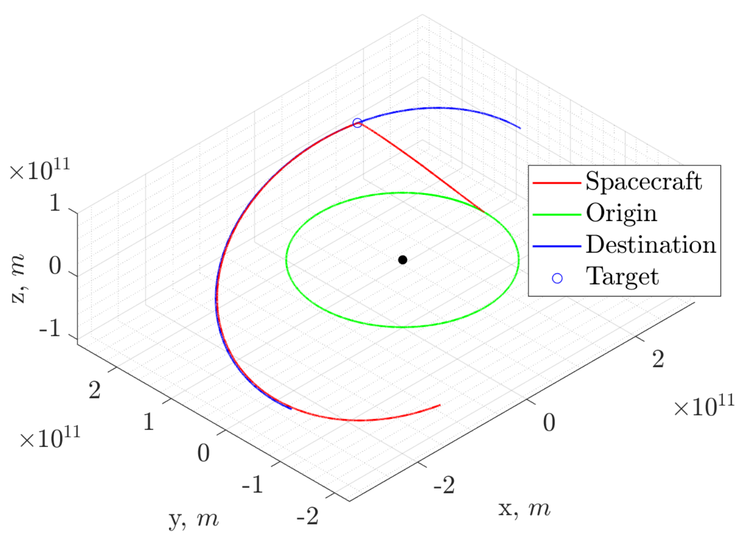
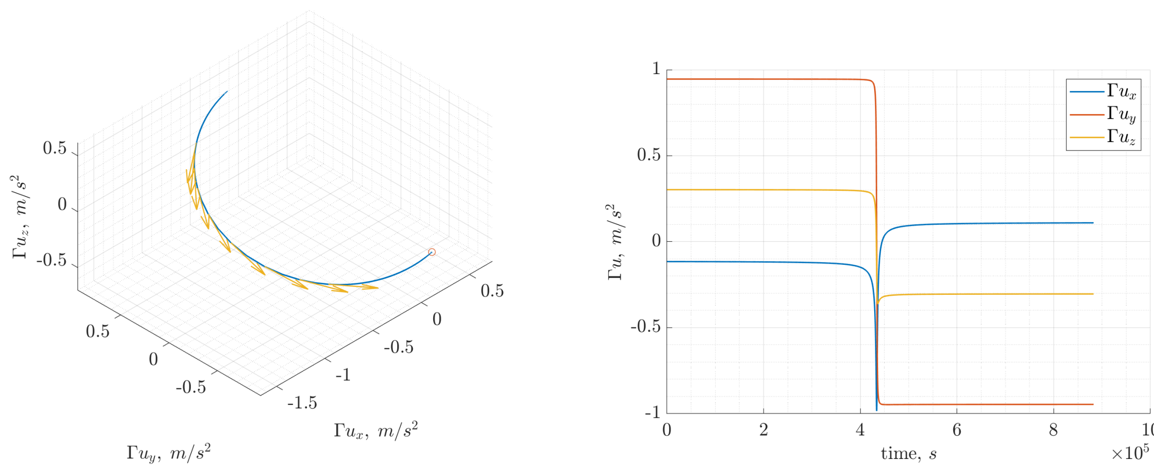


Figure 3: Optimal flight trajectory (ToF $\approx$ 10 days)

While the trajectory shown in Figure 3 does not resemble an orbit, we must remember that torchships in science fiction have propulsion capabilities unparalleled by modern technology. The capability of delivering 1g of thrust for days or weeks gives the torchship an incredible



(a) Thrust vector visualized in 3D

(b) Thrust vector elements over time

Figure 4: Optimal control strategy

amount of ΔV and causes the resulting trajectory to look like a line. By examining Figure 4, we see that the resulting thrust solution resembles the "flip-and-burn" control scheme described in Section 1, which gives us some confidence in the solution.

Since the transition matrix $\mathbf{\Phi}$ was created using truncated terms and the RK4 numerical method has integration error, the control profile is not exactly optimal. However, we know that the resulting time of flight is 10 days, which is shorter than the δt limit established in Equation (20), which is 30 days. We have also addressed the integration error in Section 7.1. Therefore in theory the errors in the control profile should not be significant.

8.1 Baseline Comparison

The effect of the Sun's gravitational field is very small at Earth's distance. Intuitively, it is possible to generate a solution comparable to the one outlined in Section 6 by simply ignoring gravity in the calculations, then adding the force of gravity back into the controls. We will now compare our optimal control method against this "gravity correction" method using the baseline case shown in Table 1.

The gravity correction method is not shown in this paper, but is trivial to determine since gravity is neglected in all calculations. Applying the gravity correction method to the baseline case of an yields a comparable trajectory shown in Figure 5.

The two trajectories are almost identical. We find that the optimal control approach yields a time of flight of 10.217 days, while the gravity correction approach results in a flight time of 10.251 days. The optimal control approach performs slightly better in this scenario.

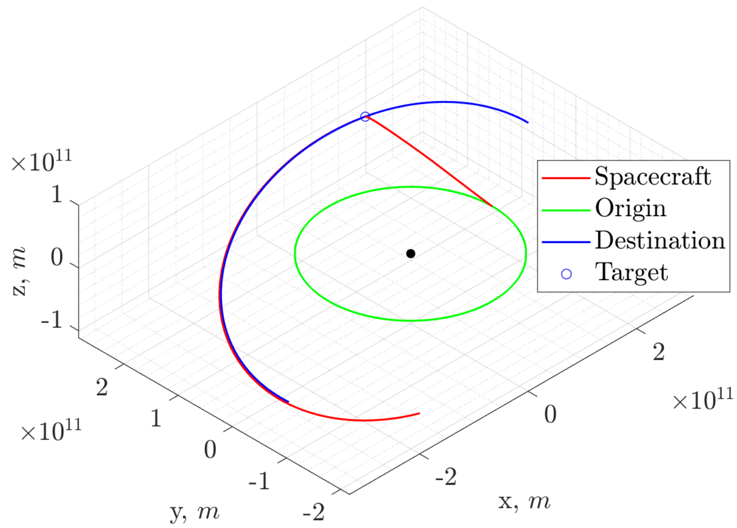


Figure 5: Gravity correction method trajectory

8.2 Flying Close to the Sun

We observe that the trajectory is essentially a straight line between the origin and the destination. What if this path nearly intersects the Sun? We will compare the optimal control method against the gravity correction method to determine their effectiveness at addressing this scenario.

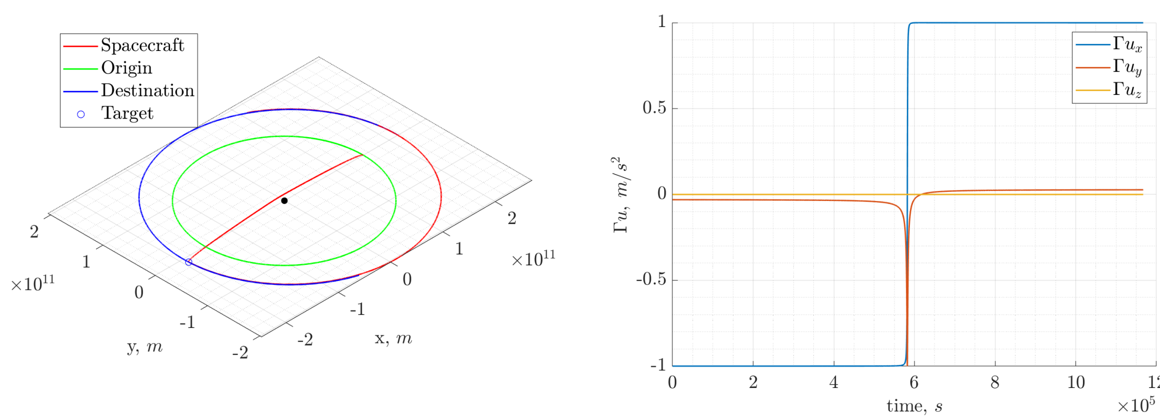


Figure 6: Optimal Control Case

While the gravity correction case performed well in the baseline case demonstrated in Section 8.1, we find that it exceeds the thrust limit due to the strong gravity correction factor when near the Sun. However, the optimal control method still provides a control scheme

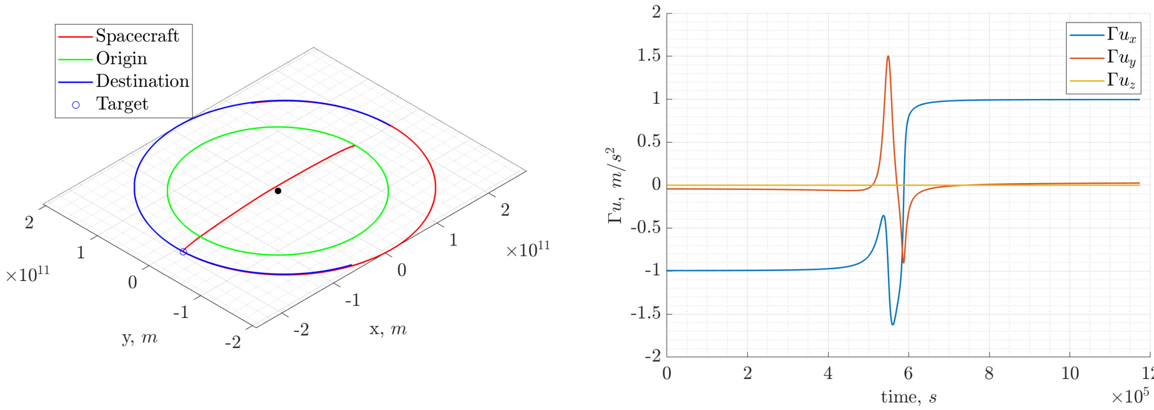


Figure 7: Gravity Correction Case

within the thrust tolerances. Furthermore, the optimal control scheme achieves a shorter time of flight: 13.512 days as compared to 13.590 days for the gravity correction method.

9 Future Work

In this project, we developed the formulation for an optimal control scheme for continuous high-thrust spacecraft. The formulation was implemented and solved using computational tools and compared with a naive solution typically described in science-fiction. The optimal control formulation was shown to perform better and more reliably in two example cases.

The current algorithm requires manual work to find a good initial guess of $\boldsymbol{\lambda}$. It may be possible to reduce manual labor by using a more robust algorithm than the trust-region method. Furthermore, it becomes increasingly difficult to find a control profile with lower thrust limits. When the thrust limit Γ_{max} is under $0.25 m/s^2$, a viable initial guess could not be found manually. It would be valuable to consolidate the results of this project with formulations for low thrust trajectories.

References

[1] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction.* USA: Princeton University Press, 2011.

[2] J. E. Prussing, *Optimal Spacecraft Trajectories.* Oxford University Press, 2018.

[3] J. S. White, “Simplified calculation of transition matrices for optimal navigation,” tech. rep., National Aeronautics and Space Administration, 1966.