HELPING THE SPACING GUILD NAVIGATE TO ARRAKIS

SPENCER ASHTON, CURTIS EVANS, CANNON TUTTLE, TYLER SANDERS

Abstract. We find the optimal path and control of navigating from one planet to another planet by using the laws of gravity to help with the acceleration. This is done by using Pontryagin maximum principle, and we discovered sensitivity in our system when optimizing over t_f and found a solution to numerical instabilty when adding massive planets such as the sun.

1. Background

The **Dune** Guild Naviators are tasked with delivering House Atreides from the planet Caladan to Arrakis. Usually they solve the optimal control in their mind while taking spice, but since spice production has been disrupted they've instead decided to turn to ACME students on Earth to solve the optimal control for them. They have asked the ACME students to find the best path while minimizing fuel consumption.

2. Mathematical Representation

We used Newton's second law of motion $F = ma$ to arrive at our state equations where m_p is the mass of the planet, $\mathbf{x}_s$ is the x and y position of the space ship, and $\mathbf{x}_p$ is the x and y position of the planet, and $\mathbf{u}$ is the acceleration in the x and y direction for the spaceship. We will have a set of planets $P = \{p_1, p_2, \ldots, p_n\}$. TODO talk about what G is and how this equation makes sense.

$$\ddot{\mathbf{x}} = -G \sum_{p \in P} \frac{m_p}{||\mathbf{x}_s - \mathbf{x}_p||_2^3} (\mathbf{x}_s - \mathbf{x}_p) + \mathbf{u}$$

Converting $\ddot{\mathbf{x}}$ into a first order differential equation we get the following state equation.

$$\begin{pmatrix} \dot{x}_s \\ \dot{y}_s \\ \ddot{x}_s \\ \ddot{y}_s \end{pmatrix} = \begin{pmatrix} \dot{x}_s \\ \dot{y}_s \\ -G \sum_{p \in P} \frac{m_p (x_s - x_p)}{((x_s - x_p)^2 + (y_s - y_p)^2)^{3/2}} + u_x \\ -G \sum_{p \in P} \frac{m_p (y_s - y_p)}{((x_s - x_p)^2 + (y_s - y_p)^2)^{3/2}} + u_y \end{pmatrix}$$

Date: April 14, 2024.

We decided to minimize the control used to get the optimal fuel usage. That gives us the following cost functional and boundary conditions.

$$J[u] = \int_0^{t_f} ||\mathbf{u}||_2^2 dt$$

$$\mathbf{x}_s(0) = \text{Caladan's Position},\ \dot{\mathbf{x}}_s(0) = \text{Caladan's Velocity}$$

$$\mathbf{x}_s(t_f) = \text{Arrakis' Position},\ \dot{\mathbf{x}}_s(t_f) = \text{Arrakis' Velocity}$$

3. Solution

We are now ready to use Pontryagin's maximum principle with

$$H = \mathbf{p} \cdot \mathbf{f}(\mathbf{x}) - L$$

Applying this to our state equation and Lagrangian we get the following Hamiltonian.

$$H = p_1\dot{x}_s + p_2\dot{y}_s + p_3\left[-G\sum_{p\in P}\frac{m_p(x_s - x_p)}{((x_s-x_p)^2+(y_s-y_p)^2)^{3/2}} + u_x\right]$$

$$+p_4\left[-G\sum_{p\in P}\frac{m_p(y_s - y_p)}{((x_s-x_p)^2+(y_s-y_p)^2)^{3/2}} + u_y\right] - u_x^2 - u_y^2$$

This give us the following co-state evolution equations by using $\mathbf{p}' = -\frac{DH}{D\mathbf{x}}$

$$\dot{p}_1 = p_3 G\left(\sum_{p\in P}\frac{m_p}{((x_s-x_p)^2+(y_s-y_p)^2)^{3/2}} - \frac{3m_p(x_s-x_p)^2}{((x_s-x_p)^2+(y_s-y_p)^2)^{5/2}}\right)$$

$$\dot{p}_2 = p_4 G\left(\sum_{p\in P}\frac{m_p}{((x_s-x_p)^2+(y_s-y_p)^2)^{3/2}} - \frac{3m_p(y_s-y_p)^2}{((x_s-x_p)^2+(y_s-y_p)^2)^{5/2}}\right)$$

$\dot{p}_3 = -p_1$

$\dot{p}_4 = -p_2$.

$p_1(t_f) = p_2(t_f) = 0 = p_3(t_f) = p_4(t_f)$

Now we need to find $\tilde{\mathbf{u}}$ by maximizing the Hamiltonian i.e. $\frac{DH}{D\mathbf{u}} = 0$.

$$\frac{\partial H}{\partial u_x} = p_3 - 2u_x \text{ and } \frac{\partial H}{\partial u_y} = p_4 - 2u_y$$

now solving for u_x and u_y we get

$$\tilde{u}_x = \frac{p_3}{2} \text{ and } \tilde{u}_y = \frac{p_4}{2}.$$

Plugging $\tilde{u}_x$ and $\tilde{u}_y$ back into the state and costate evolution equations we get the following boundary value problem.

$$\dot{x}_s = \dot{x}_s$$
$$\dot{y}_s = \dot{y}_s$$
$$\ddot{x}_s = -G\sum_{p\in P}\frac{m_p(x_s - x_p)}{((x_s - x_p)^2 + (y_s - y_p)^2)^{3/2}} + \frac{p_3}{2}$$
$$\ddot{y}_s = -G\sum_{p\in P}\frac{m_p(y_s - y_p)}{((x_s - x_p)^2 + (y_s - y_p)^2)^{3/2}} + \frac{p_4}{2}$$
$$\dot{p}_1 = p_3 G\left(\sum_{p\in P}\frac{m_p}{((x_s - x_p)^2 + (y_s - y_p)^2)^{3/2}} - \frac{3m_p(x_s - x_p)^2}{((x_s - x_p)^2 + (y_s - y_p)^2)^{5/2}}\right)$$
$$\dot{p}_2 = p_4 G\left(\sum_{p\in P}\frac{m_p}{((x_s - x_p)^2 + (y_s - y_p)^2)^{3/2}} - \frac{3m_p(y_s - y_p)^2}{((x_s - x_p)^2 + (y_s - y_p)^2)^{5/2}}\right)$$
$$\dot{p}_3 = -p_1$$
$$\dot{p}_4 = -p_2.$$

With boundary conditions

$$x_s(t_0) = x_{p_1}(t_0),\ y_s = y_{p_1}(t_0),\ \dot{x}_s(t_0) = \dot{x}_{p_1}(t_0),\ \dot{y}_s(t_0) = \dot{y}_{p_1}(t_0)$$

and

$$x_s(t_f) = x_{p_2}(t_f),\ y_s = y_{p_2}(t_f),\ \dot{x}_s(t_f) = \dot{x}_{p_2}(t_f),\ \dot{y}_s(t_f) = \dot{y}_{p_2}(t_f)$$

if we are optimizing over t_f we get the added boundary condition that $H(t_f) = 0$, and if we are optimizing over t_0 we get $H(t_0) = 0$. This problem is now set up to use `solve_bvp`.

4. Solution

Our first task was figuring out how to insert planets into our system. We needed to have have a time dependent function that would give us the planet's position and velocity.

4.1. **Goals.** We hope to acheive an optimal path from Caladan to Arrakis with a fixed final time, a free final time where we need $H(t_f) = 0$ as an added boundary condition, and an optimal control with multiple solar systems and planets. We also want to figure out the best path with a given arbitrary layout for a given solar system. Essentially the Spacing Guild would like to use our algorithms to help navigate from any planet. TODO should we talk about optimizing over t_0 if so we should say. We realize that in planetary travel we would like to get to a given planet on a fixed day and then leave our leaving day free. That would mean we are optimixing over t_0 so we would have a free parameter with the added boundary condition that $H(t_0) = 0$.

4.2. **Simulations.** We simulated our model using the inner planets of our own solar system, but changed the names to be planets from the **Dune** universe so the Guild Naviators wouldn't be confused. The first model we ran was without the sun and a fixed final time of 200 Earth days (See Figure 1).

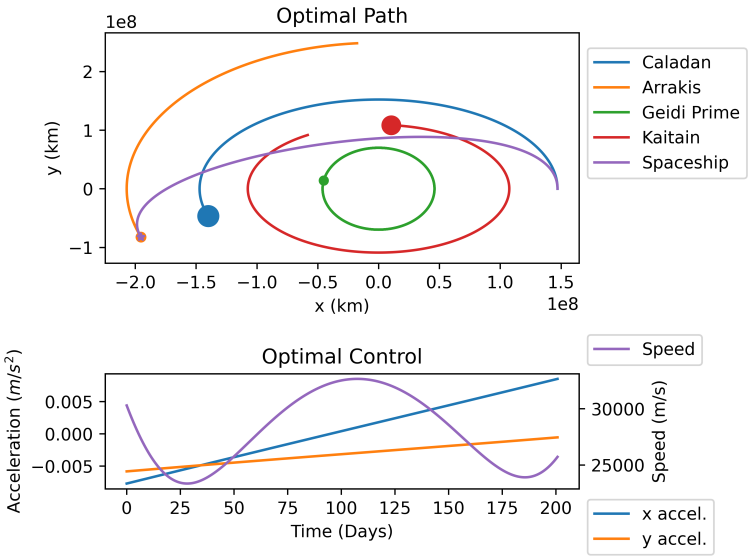

FIGURE 1. Going from Caladan to Arrakis in 200 Earth Days with a fixed final time

Our bare bones model got us from Caladan to Arrakis successfully, but we would like to save as much fuel as possible so our main goal is to minimize over t_f as well. This was done by adding a free parameter in `solve_bvp` and adding the boundary condition of $H(t_f) = 0$. We discovered that this model was sensitive to initial guess of t_f and also the initial state space guess. We ran model with varying t_f guesses and got varying results with no convergence on an optimal t_f for all the different guesses (See Figure **??**).

We needed to add the sun, but the problem became unstable due the massive size of the sun (See Figure 3). One suggestion we had was to work in logspace and we also changed how we were computing the norm in the denominator. We were using a simple function that was computing the norm from the formula, but instead we started using `scipy.linalg.norm`. These two changes solved the instability problem. This difference can be seen in Figure 4. One thing to notice in Figure 1 and Figure 4 is that the optimal control completely changes. We believe that the control with the sun is more realistic and a control that the Guild Naviators would accept.

Another problem we ran into was our units of time. Our class for planets was giving us the position of the planet in Earth years at the beginning so we thought were telling our model to get from Caladan to Arrakis in $\frac{1}{2}$ a

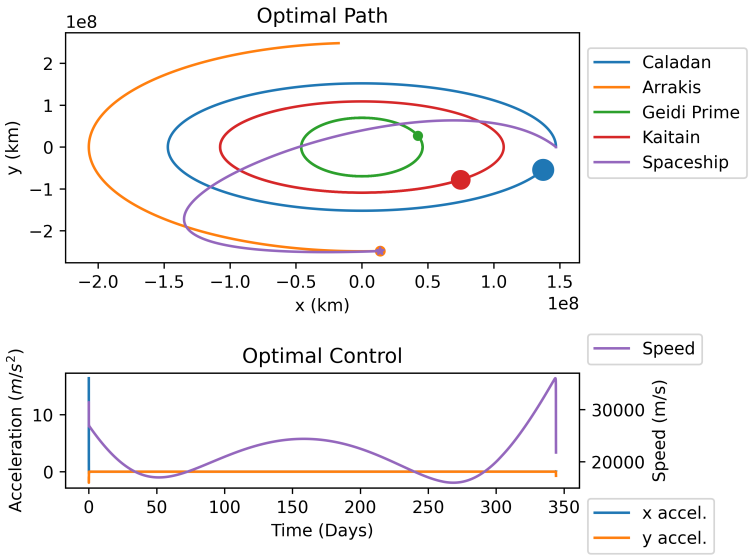

FIGURE 2. Going from Caladan to Arrakis with t_f guess as 365 days. The optimal t_f was 344 days.

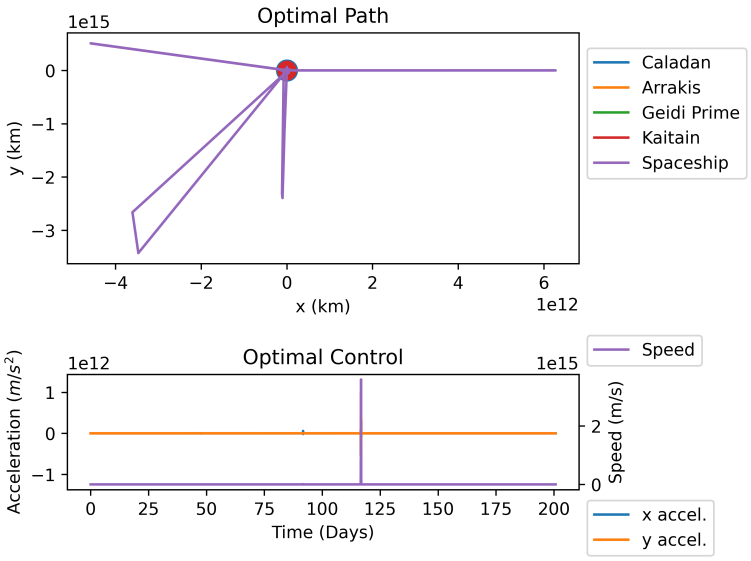

FIGURE 3. Instability with the sun

year by giving its final time as $\frac{1}{2}$. This was giving us an optimal control and speed that was inconceivable

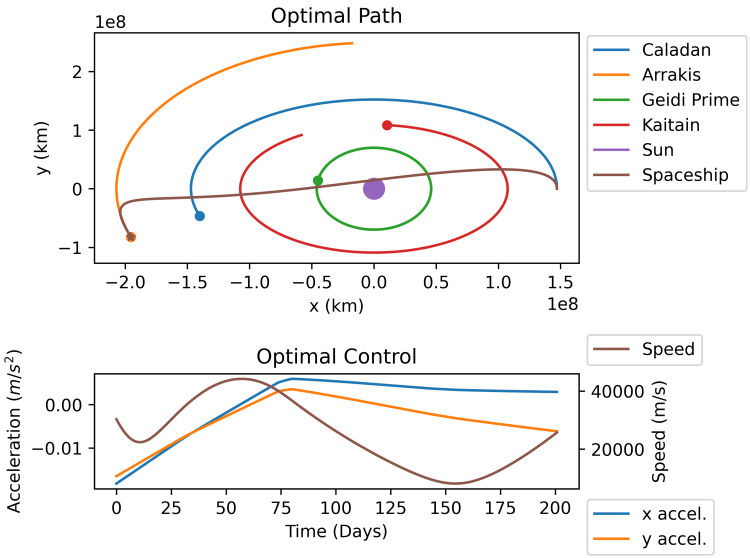

FIGURE 4. Stability with the sun

REFERENCES

[1] William M., Samuel L., Jeff S. University Physics Volume 1. 2021 OpenStax. https://openstax.org/details/books/university-physics-volume-1