Trajectory Optimization for Landing a Reusable Rocket Booster

Ashwin Gupta

***Abstract*— In this work I model the dynamics of a 2D rocket system that is in process of deorbiting. The goal is to plan a trajectory with direct collocation that enables landing the rocket upright at a designated location with the most minimal consumption of fuel.**

I. INTRODUCTION

Reusable rocket booster technology has revolutionized the space launch industry by significantly reducing costs. By eliminating the need to build a new booster from scratch for every launch, launch providers are able to provide cheaper flights to their customers. However, reusable boosters do have the disadvantage of needing to conserve fuel for landing which in turn decreases the amount of fuel that can be used to deliver payload mass to orbit. As such, it is beneficial to use the minimal amount of fuel possible during the landing procedure.

In this work I model the dynamics of a 2D rocket system that is in process of deorbiting. This means the rocket begins with a large horizontal velocity and high altitude. The goal is to plan a trajectory with direct collocation that enables landing the rocket upright at a designated location with the most minimal consumption of fuel. Landing is defined to be a state where there is zero velocity at the time when the rocket first touches the ground.

II. SIMULATION AND DYNAMICS

A. System Overview

The system of interest is a rocket with position and orientation tracked in a 2d world frame. This is a system largely of my own creation, however some inspiration is taken from [1].

B. System State

The system has the following state vector:

$$q_t = \begin{bmatrix} x_t & y_t & \theta_t & \dot{x_t} & \dot{y_t} & \dot{\theta}_t & f \end{bmatrix}^T$$

x_t and y_t represent the 2d position of the vehicle in world coordinates and θ_t represents its orientation. f is the fuel consumption of the vehicle.

C. Initial Conditions

The initial conditions of the system are a large positive y_0, a large positive $\dot{x_0}$, and all other variables set to 0.

D. Control Inputs

The control inputs to the system are as follows:

$$u_t = \begin{bmatrix} F_t \\ \alpha_t \end{bmatrix}$$

F_t is the forced being applied from the thruster. α_t represents the gimbal of the thruster.

E. Diagrams

Diagrams of the system can be seen in the figures below. Figure 1 shows the two coordinate systems. The rocket's position and rotation arre measured in world coordinates. Figure 2 shows the thrust vector in the ship coordinate system and gimbal angle α.

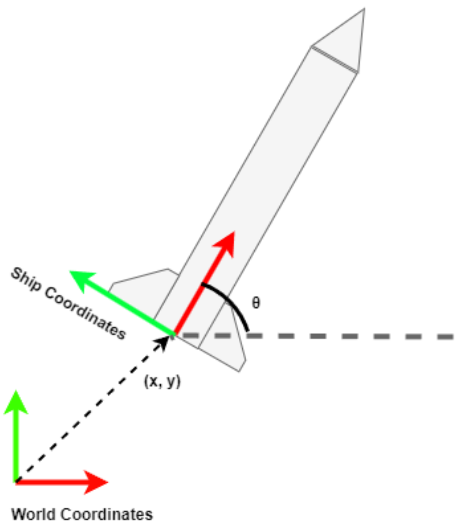

Fig. 1. Rocket model coordinate systems

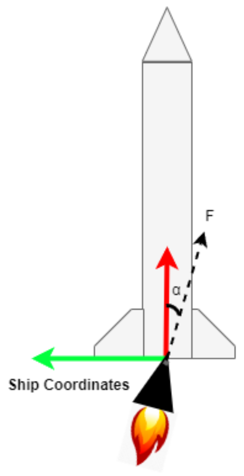

Fig. 2. Rocket model control inputs in ship frame

F. Dynamics

Let m refer to the mass of the booster. I refers to the moment of inertia, h refers to the height of the booster and k is a constant with units fuel consumption per impulse. The dynamics of the system can be written as follows, assuming the only other force acting on the booster other than its own thruster is gravity. Note that R_{ship}^{world} is a rotation matrix from the ship frame to the world frame, and it is a function of the rocket's heading θ_t.

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = R_{ship}^{world}\begin{bmatrix}\frac{F}{m}\cos\alpha\\ \frac{F}{m}\sin\alpha\end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} + \begin{bmatrix}0\\ -9.8\end{bmatrix}$$

$$\ddot{\theta} = \frac{-F\sin\alpha}{I}\frac{h}{2}$$

$$\dot{f} = kF$$

Thus the whole dynamics of the system as a function of the state vector $q \in \mathbf{R}^7$ and control input $u \in \mathbf{R}^2$ can be written as:

$$\dot{q} = g(q,u) = \begin{bmatrix} q_4 \\ q_5 \\ q_6 \\ \frac{u_1}{m}(\cos q_3 \cos u_2 - \sin q_3 \sin u_2) \\ \frac{u_1}{m}(\sin q_3 \cos u_2 + \cos q_3 \sin u_2) - 9.8 \\ -\frac{hu_1 \sin u_2}{2I} \\ ku_2 \end{bmatrix}$$

G. Simulated Model

All simulation is done in the Matlab programming language. Additionally, an animation is generated in the Matlab programming language. The ODE presented above is implemented in Matlab and simulated with ode45. For testing the dynamics, the force and gimbal angle are assumed to vary linearly between the collocation knot points.

H. Simulation Verification

First, the model is tested with no inputs. The trajectory of the rocket exhibits the expected falling motion due to gravity as seen in figure 3 Additionally, some individual time series plots display correctly that there is no motion in x or theta, but parabolicly decreasing motion in y as expected by the kinematics of an object under constant vertical acceleration.

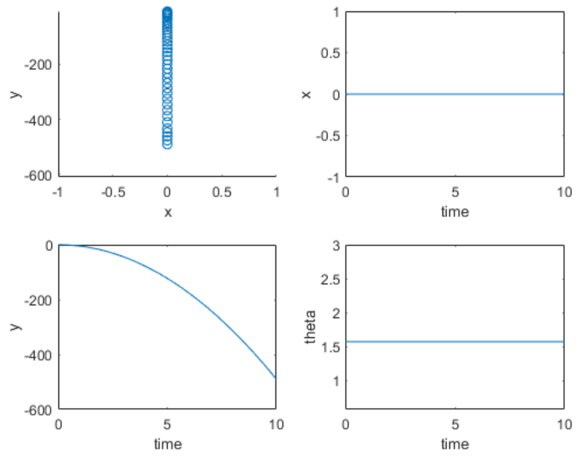

Fig. 3. Rocket model no inputs

I also test the model with a simple set of inputs. Constant thrust is applied and the engine is gimbaled first to the left for 5 seconds by 0.001 radians and then to the right for 5 seconds by 0.002 radians. The results of this test are seen in figure 4. This set of results matches our expected behavior. First, we see the rocket is progressing upwards. It initially turns slightly right, and then turns heavily left. In the time series plots we observe that theta decreases then increases as we would expect this sort of thrust vectoring to produce.

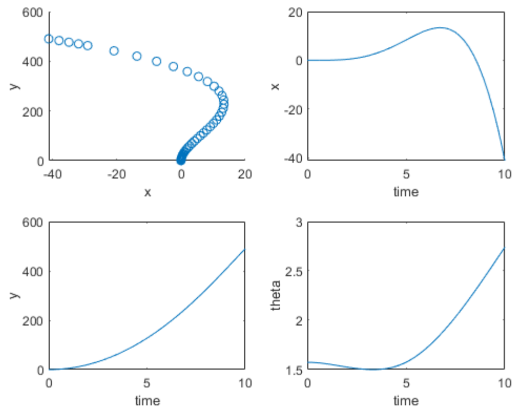

Fig. 4. Rocket model with inputs

I. Further Simulation Efforts

In addition to the static trajectory plots from simualting the dynamics, I include an animation utility that can animate the position and heading of the rocket as well as a thrust vector illustrating the command of the rocket at a particular moment. A static capture of the animation utility in action is seen in figure 5.

III. CONTROLLER

A. Control Type

I use feedforward, open-loop control to generate the trajectory over a variable duration that is part of the decision

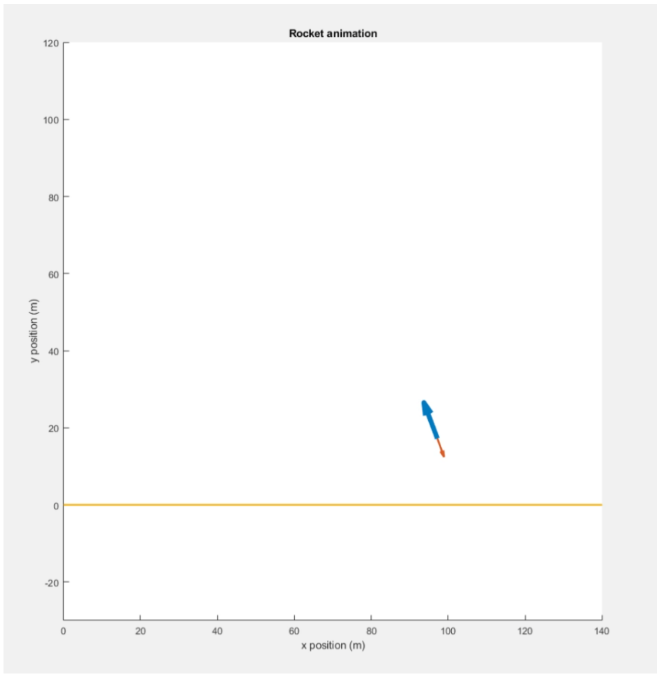

Fig. 5. Animation

vector. I use direct collocation since this is the most computationally feasible method for a system with a somewhat large state vector like this one. The trajectory can be generated offline, so speed is not a major concern.

B. Cost Function

This cost function to minimize is as follows, where z_t is the decision variable, f is the fuel consumption element of the state, and T is the final time. T is part of the decision variable. The number of states in the decision vector is $N = 20$ nodes, evenly spaced over the duration.

$$z_t = \begin{bmatrix} F_1 & \alpha_1 & \dots & \alpha_{20} & x_1 & y_1 & \dots & \dot{\theta}_{20} & f_{20} & T \end{bmatrix}^T$$

$$z^* = \arg\min_{z_t} f_T$$

C. Constraints

Defect constraints are enforced via a trapezoidal integration scheme. That is

$$q_{k+1} = \frac{g(q_k, u_k) + g(q_{k+1}, u_{k+1})}{2} \frac{T}{N-1} + q_k$$

F_t is the forced being applied from the thruster, and is bounded between 0 and F_{max}. Likewise, α_t represents the gimbal of the thruster, and is bounded as well by α_{max} and $-\alpha_{max}$. Lastly, there is a constraint such that the final velocity is 0 and the final position matches the target.

IV. CONTROL PERFORMANCE

A. Parameters

The physical constants were chosen as seen in table I. The initial state is constrained to be as follows.

constant	value
h	3
m	10
k	3
I	$\frac{1}{12}mh^2$

TABLE I
CONSTANTS

$$q_0 = \begin{bmatrix} 0 & 100 & \frac{\pi}{8} & 30 & 0 & 0 & 0 \end{bmatrix}^T$$

The final state is constrained to be as follows where λ is a free variable.

$$q_T = \begin{bmatrix} 100 & 0 & \frac{\pi}{2} & 0 & 0 & 0 & \lambda \end{bmatrix}^T$$

B. Results and Discussion

Plots of the resulting trajectory in each component of the state vector can be seen in figures 6 through figure 11. The optimization completes in roughly 30 seconds and all constraints are satisifed. The solver was succesful with initial conditions of all ones. We observe a roughly parabolic trajectory in x and y as we expect from a rocket that is deorbiting. The x and y trajectories individually are also monotonic as expected and are relatively smooth. The fuel consumption and fuel efficient force trajectories are somewhat unexpected. The force was delivered as a series of impulses with no obvious pattern other than a large impulse at the end. I would've expected to see a more smooth function here. This suggests that the fuel optimal trajectory is not a simple function, but rather just bursts of thrust. This suggests that this system is a good canididate for optimal control as opposed to hand designing some control system. A video animation of the rocket landing can be seen here. A video overview of this project can be seen here.

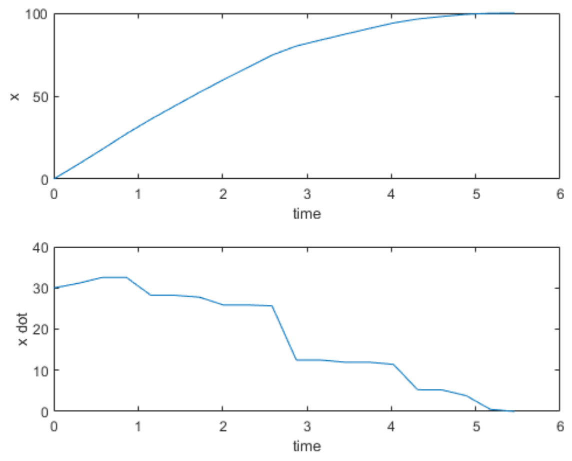

Fig. 6. Rocket x and x dot vs time

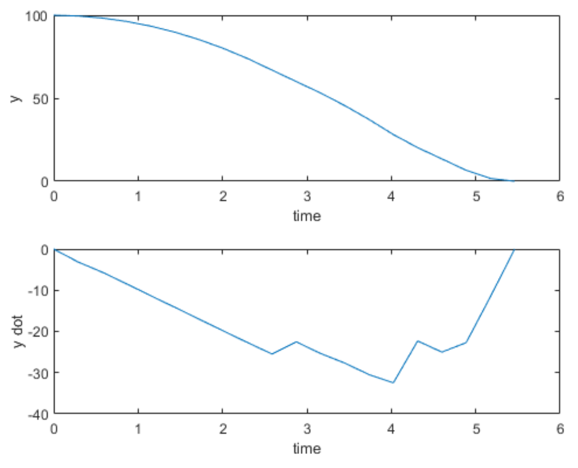

Fig. 7. Rocket y and y dot vs time

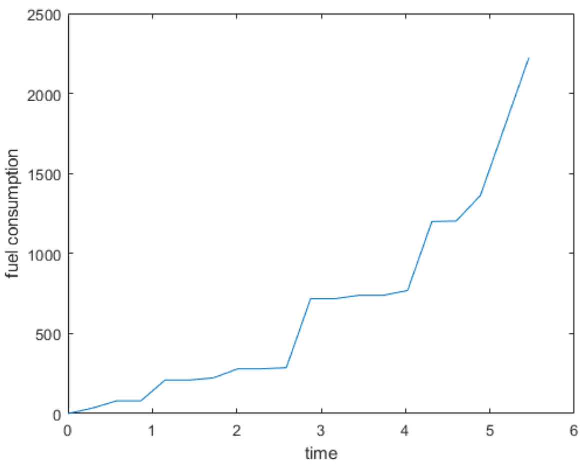

Fig. 10. Rocket fuel vs time

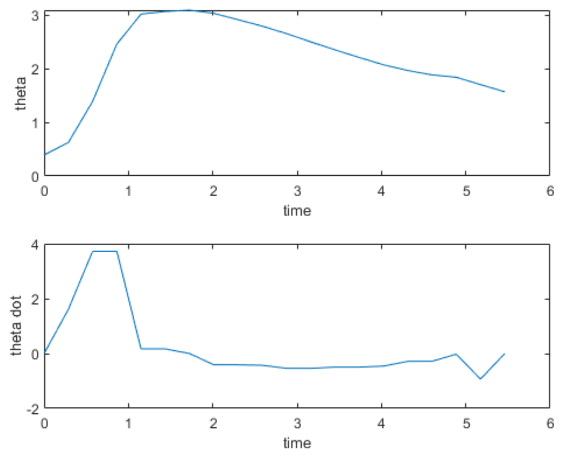

Fig. 8. Rocket theta and theta dot vs time

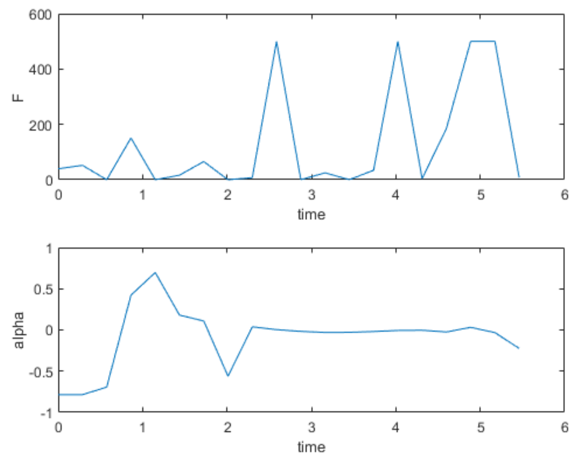

Fig. 11. Inputs vs time

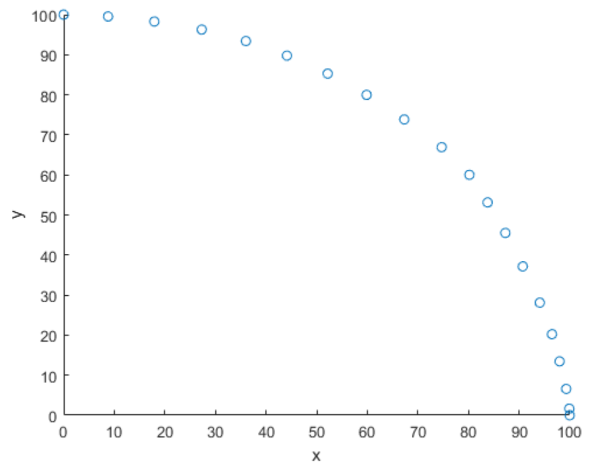

Fig. 9. Rocket x vs y

References

[1] Rocket Sim, Pitch Analysis of Rigid Body Rocket, https://rocketsim.wordpress.com/2017/03/24/pitch-analysis-of-rigid-body-rocket-part-1-equations-of-motion/