Optimal Control for Minimum Fuel Pinpoint Landing

Devin Ardeshna, Nick Delurgio, and Pol Francesch Huc

{ardeshna, delurgio, pfrances}@stanford.edu

June 8, 2023

Abstract

Autonomous landing is a modern problem in the Aerospace industry, inspired by the desire for reusable launch vehicles to reduce cost to orbit. Major innovations in this field were made in the early 2010s, where fuel and landing optimal trajectories were designed using Convex Optimization in reference to the Mars landing problem. This project starts by revisiting these innovations in the guidance problem and then expands the work to include more sophisticated dynamics, including drag and attitude dynamics. We also address the tracking problem to ensure that the vehicle follows the prescribed trajectory in the presence of disturbances. Code for this project can be found in our GitHub repository. Additionally, a video presentation can be found on Google Drive.

1 Introduction

For decades, the space sector has been burdened by the expense of sending spacecraft to orbit. The high cost is primarily driven by expendable launch vehicles, which cannot be used multiple times. In 2015, SpaceX successfully performed the first autonomous landing of a Falcon 9 booster. Since then the reusable Falcon 9 boosters have dramatically reduced cost to orbit, enabling rapid economic growth in the space sector. Blue Origin has also achieved autonomous landing, and several launch vehicles are currently being developed to iterate and improve upon the Falcon 9 design (New Glenn, Neutron, Starship, Terran R, etc).

Autonomously landing rockets is a challenging problem which required immense innovation to achieve, both theoretically and practically. The challenge and importance of optimal control in autonomous landing serves as the motivation for this report. In this project, we aim to apply and experiment with optimal control techniques to achieve autonomous landing in simulation, both considering the guidance problem and the tracking problem.

2 Literature Review

In the early 2000s, the pinpoint landing problem started gaining attention due to the renewed interest in Mars exploration, including the Curiosity Rover mission. Existing powered descent guidance (PDG) algorithms were still inherited from the Apollo era and solved a simplified problem in closed-form by assuming quartic polynomial trajectories [1]. However, these algorithms did not optimize fuel usage which significantly limited large diverts during descent.

To address this shortcoming, Açıkmeşe and Ploen [2] propose a pinpoint landing algorithm that solves a minimum fuel trajectory optimization problem. This trajectory optimization considers 3DOF translational dynamics but ignores rotational dynamics since the attitude control bandwidth is often much higher. The authors prove that applying a convex relaxation produces an optimal solution to the original nonconvex problem and they call this process "lossless convexification". They formulate this relaxation as a second order cone program (SOCP) for which there exist efficient solvers with well understood convergence properties.

Blackmore et al. [3] extend this work to the case when no feasible trajectory to the target exists. They define the minimum landing error problem and propose a procedural optimization approach that solves two SOCPs sequentially. The first program minimizes the distance between the target and the achievable landing location. The second program generates a minimum fuel trajectory that achieves this minimum landing error. This approach is again extended by Açıkmeşe et al. [4] to include thrust pointing constraints, which can be made convex. Finally, Açıkmeşe et al. [5] perform flight testing of this powered descent algorithm (dubbed "G-FOLD") in collaboration with Masten Space Systems. Using the vertical lander vehicle "Xombie", they successfully perform large diverts of 550, 650 and 750 meters.

Szmuk and Acikmese [6] consider the 6DOF minimum time pinpoint landing problem with both translational and rotational dynamics. They extend prior work on the 3DOF free-final-time and the 6DOF fixed-final-time minimum fuel problems. The authors employ successive convexification (SCvx) to repeatedly solve a fixed-final-time convex sub-problem. The solution is shown to converge to a local minimum of the original, nonconvex minimum time problem.

Sánchez-Sánchez and Izzo [7] pivot from convex programming and prove that DNNs can be trained offline and used to generate near-optimal landing trajectories for a variety of scenarios. Their intended goal is to reduce computation time in the lander, as they show that solving the optimal control problem is more computationally cumbersome than performing inference on the DNN. Further investigation in Cheng et al. [8] shows that DNNs can be used to learn dynamics and then use them to solve the optimal control problem under irregular gravity fields. Their work shows that DNNs can more accurately recover the dynamics than classical techniques, which then enables better landing trajectories.

3 Technical Contribution

We first implement a 3DOF convex optimization landing guidance algorithm from the literature [2]. We follow this trajectory with an LQR controller to ensure that tracking error remains bounded, even with the inclusion of disturbances such as drag. We also experiment with a Model Predictive Control (MPC) algorithm which solves the minimum-fuel problem online. Finally, we use successive convexification (SCvx) to approach the 6DOF minimum-fuel guidance problem.

4 3DOF Fuel-Optimal Pinpoint Landing

4.1 Dynamics

To start, we consider the 3DOF dynamics of the landing problem, using a flat earth reference frame. These dynamics are given as follows:

$$\ddot{\mathbf{r}} = \mathbf{g} + \frac{\mathbf{T}}{m}, \quad \dot{m} = -\alpha ||\mathbf{T}||_2 \tag{1}$$

where $\mathbf{r}$ is the vehicle position, $\mathbf{g} \in \mathbb{R}^3$ is the constant gravity vector in the fixed frame, $\mathbf{T} \in \mathbb{R}^3$ is the thrust vector in the fixed frame, $m \in \mathbb{R}$ is the vehicle mass, and $\alpha \in \mathbb{R}$ is the efficiency metric of the thruster to relate thrust to propellant consumption. These dynamics are nonlinear because mass is not constant.

We can also append the dynamics of $\ddot{\mathbf{r}}$ with $-\frac{B}{m}||\dot{\mathbf{r}}||\dot{\mathbf{r}}$ to include drag, where B is the ballistic coefficient of the lander. These dynamics will be used to introduce disturbances that the closed loop control schemes will compensate for.

4.2 Minimum-Fuel Convex Programming

We solve the same optimal control problem that was presented in the midterm report, but with an additional thrust pointing constraint to ensure a reasonable attitude of the lander: $\hat{\mathbf{n}}^T\mathbf{T} \geq ||\mathbf{T}|| \cos(\theta)$, where θ is the maximum allowed angle of the thrust vector with respect to $\hat{\mathbf{n}}$. The convex problem with the additional thrust pointing constraint is given as [4]:

$$\begin{gathered}
\min_{\mathbf{u}_0,\dots,\mathbf{u}_N,\sigma_0,\dots,\sigma_N} -z_N \quad \text{subject to, for } k=0,\dots,N, \\
\mathbf{r}_{k+1} = \mathbf{r}_k + \frac{\Delta t}{2}\left(\dot{\mathbf{r}}_k + \dot{\mathbf{r}}_{k+1}\right) + \frac{\Delta t^2}{12}\left(\mathbf{u}_{k+1} - \mathbf{u}_k\right) \\
\dot{\mathbf{r}}_{k+1} = \dot{\mathbf{r}}_k + \frac{\Delta t}{2}\left(\mathbf{u}_k + \mathbf{u}_{k+1}\right) + g\Delta t \\
z_{k+1} = z_k - \frac{\alpha \Delta t}{2}\left(\sigma_k + \sigma_{k+1}\right), \\
\|\mathbf{u}_k\| \leq \sigma_k, \quad \hat{\mathbf{n}}^T\mathbf{u}_k \geq \sigma_k \cos(\theta) \\
\mu_{1,k}\left[1 - (z_k - z_{0,k}) + \frac{(z_k - z_{0,k})^2}{2}\right] \leq \sigma(t) \leq \mu_{2,k}\left[1 - (z_k - z_{0,k})\right], \\
z_{0,k} = \ln\left(m_{wet} - \alpha\rho_2 k\Delta t\right), \quad \mu_{1,k} = \rho_1 e^{-z_{0,k}}, \quad \mu_{2,k} = \rho_2 e^{-z_{0,k}} \\
z_{0,k} \leq z_k \leq \ln\left(m_{wet} - \alpha\rho_1 t\right), \\
z_0 = \ln m_{wet}, \quad \mathbf{r}(0) = \mathbf{r}_0, \quad \mathbf{r}_N = \dot{\mathbf{r}}_N = \mathbf{0}, \quad N\Delta t = t_f.
\end{gathered} \tag{2}$$

In this problem, a change of variables is performed, where $z = \log(m)$, $\mathbf{u} = \mathbf{T}/m$, and $\sigma = \Gamma/m$. A slack variable Γ is introduced to make the lower thrust bound constraint convex. Although it is not obvious, [2] shows that the constraint $\|\mathbf{T}(t)\| \leq \Gamma(t)$ (and hence $\|\mathbf{u}(t)\| \leq \sigma(t)$) is tight; therefore, the optimal solution of this problem is equivalent to the original problem. This problem is also discretized using a trapezoidal integration scheme. The solution of this problem is approximately the same as the original problem, and will minimize propellant consumption during the execution of a landing maneuver for a fixed final time.

4.3 LQR Tracking

We generate a trajectory by solving the convex problem with a fixed final time and using the non-drag dynamics from Equation 1. We then save this trajectory and implement a tracking LQR in order to follow this trajectory under numerical integration errors and a drag perturbation.

LQR tracking is implemented by calculating a gain at each time step and updating the closed-loop control as:

$$u^t_{cl} = u^t_{ol} - K^t(x^t_{actual} - x^t_{planned}),$$

where K^t is computed from the linearization of the dynamics A^t and B^t, and Q and R is matrices to evaluate the cost of state errors and control inputs, respectively. We then compare the performance of LQR closed-loop tracking against the open-loop tracking.

4.4 MPC

The MPC implementation works by solving the previously defined convex problem (using non-drag dynamics from Equation 1) and applying the first control input. It then re-plans at the new state until the terminal state is reached. Note that in our implementation the time horizon is defined as the time left to land. The step size is adjusted for each run such that the number of steps stays constant.

As the lander approaches the end of the trajectory, the convex program for minimum fuel landing becomes infeasible. G-FOLD solves this problem by switching to a minimum-landing error problem at this point [5]. For our case, once the program becomes infeasible, we simply complete the last feasible trajectory in the open-loop or with LQR tracking and present both approaches.

5 6DOF Fuel-Optimal Pinpoint Landing

5.1 Dynamics

The 6DOF dynamics are as given in Szmuk and Acikmese [6]:

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{g}_{\mathcal{I}} + \frac{R_{\mathcal{I}/\mathcal{B}}\mathbf{T}_{\mathcal{B}}(t)}{m(t)}, \quad & \dot{q}_{\mathcal{B}/\mathcal{I}} &= \frac{1}{2}\Omega(\omega_{\mathcal{B}}(t))q_{\mathcal{B}/\mathcal{I}}(t), \\ \dot{m}(t) &= -\alpha||\mathbf{T}_{\mathcal{B}}(t)||_2, \quad & \dot{\omega}_{\mathcal{B}}(t) &= J_{\mathcal{B}}^{-1}([\mathbf{r}_{\mathcal{T},\mathcal{B}}\times]\mathbf{T}_{\mathcal{B}}(t) - [\omega_{\mathcal{B}}(t)\times]J_{\mathcal{B}}\omega_{\mathcal{B}}(t)), \end{aligned} \tag{3}$$

where $\mathcal{I}$ represents the inertial frame and $\mathcal{B}$ represents the body-fixed frame. $q_{\mathcal{B}/\mathcal{I}} \in \mathbb{R}^4$ is the quaternion representing the rotation between these two frames which is the attitude of the vehicle. $J_{\mathcal{B}}$ is the inertia tensor and $\omega_{\mathcal{B}}$ is the angular velocity of the vehicle, with respect to the inertial frame and represented in body-fixed coordinates. $\mathbf{T}_{\mathcal{B}}$ is the thrust vector in body-fixed coordinates.

5.2 Successive Convexification for Minimum-Fuel Trajectories

Currently, there is no formulation of the minimum-fuel landing problem for 6DOF dynamics that is convex. Therefore, successive convexification (SCvx), a form of sequential convex programming (SCP), is used in order to iterate and find a local minima for the problem. As discussed in detail in [6], through clever implementation of trust regions Δ and dynamic relaxation $\bar{\boldsymbol{\nu}}$, an iteration of the SCP can be formulated. While [6] provides the SCP for minimum-time trajectories, we make small modifications such that it is programmed for minimum fuel:

$$\underset{m_f^i, \mathbf{u}_k^i}{\text{minimize}} \quad -m_f^i + w_\nu \left\|\overline{\boldsymbol{\nu}}^i\right\|_1 + w_\Delta^i \left\|\overline{\boldsymbol{\Delta}}^i\right\|_2 + w_{\Delta_\sigma} \left\|\Delta_\sigma\right\|_1$$

Boundary Conditions:

$$\begin{aligned}
m_0^i &= m_{\text{wet}} \\
\mathbf{r}_{\mathcal{I},0}^i &= \mathbf{r}_{\mathcal{I},i} & \mathbf{r}_{\mathcal{I},K}^i &= \mathbf{0} \\
\mathbf{v}_{\mathcal{I},0}^i &= \mathbf{v}_{\mathcal{I},i} & \mathbf{v}_{\mathcal{I},K}^i &= \mathbf{v}_{\mathcal{I},f} \\
& & q_{\mathcal{B}/\mathcal{I},K}^i &= q_{\mathcal{B}/\mathcal{I},f} \\
\boldsymbol{\omega}_{\mathcal{B},0}^i &= \boldsymbol{\omega}_{\mathcal{B},i} & \boldsymbol{\omega}_{\mathcal{B},K}^i &= \mathbf{0} \\
& & \mathbf{e}_2 \cdot \mathbf{u}_K^i &= \mathbf{e}_3 \cdot \mathbf{u}_K^i = 0
\end{aligned}$$

Dynamics:

$$\mathbf{x}_{k+1}^i = \bar{A}_k^i \mathbf{x}_k^i + \bar{B}_k^i \mathbf{u}_k^i + \bar{C}_k^i \mathbf{u}_{k+1}^i + \bar{\Sigma}_k^i \sigma^i + \bar{\mathbf{z}}_k^i + \boldsymbol{\nu}_k^i$$

State Constraints:

$$\begin{aligned}
m_{\text{dry}} &\le m_k^i, \qquad \tan \gamma_{gs} \left\| H_{23} \mathbf{r}_{\mathcal{I},k}^i \right\|_2 \le \mathbf{e}_1 \cdot \mathbf{r}_{\mathcal{I},k}^i \\
\cos \theta_{\max} &\le 1 - 2 \left\| H_q q_{\mathcal{B}/\mathcal{I},k}^i \right\|_2^2, \qquad \left\| \omega_{\mathcal{B},k}^i \right\|_2 \le \omega_{\max}
\end{aligned}$$

Control Constraints:

$$T_{\min} \le B_g(\tau_k)\, \mathbf{u}_k^i, \quad \left\| \mathbf{u}_k^i \right\|_2 \le T_{\max}, \quad \cos \delta_{\max} \left\| \mathbf{u}_k^i \right\|_2 \le \mathbf{e}_1 \cdot \mathbf{u}_k$$

Trust Regions:

$$\delta \mathbf{x}_k^i \cdot \delta \mathbf{x}_k^i + \delta \mathbf{u}_k^i \cdot \delta \mathbf{u}_k^i \le \Delta_k^i, \quad \left\| \delta \sigma^i \right\|_1 \le \Delta_\sigma^i$$

In this formulation, i represents the iteration of SCP. The slack variable and trust region constraints are part of the objective such that the dynamic relaxation is valid. SCvx keeps iterating until the cost function reaches a local minimum, indicating convergence.

5.3 LQR Tracking

In order to follow the trajectory generated by SCvx, we employ an LQR controller. We run the controller at 10Hz and linearly interpolate between trajectory setpoints. At each timestep, we linearize the dynamics around the current state to solve for the optimal gain K^t. The control law is the same as the 3DOF LQR controller.

6 Results

6.1 Scenario

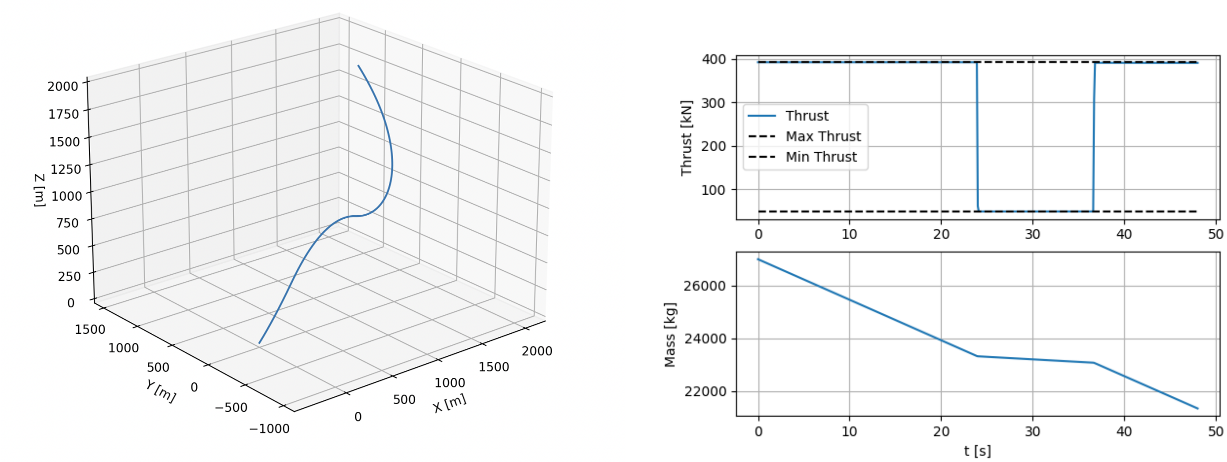
For our scenario we took estimated values from the current design of Blue Origin's New Shepard vehicle and tabulated them in Table 1. The choice of maximum throttle is used in order to provide the LQR tracking controller with additional control authority such that it does not exceed the maximum vehicle thrust. This is important for since such a closed-loop control law does not have any thrust magnitude constraints.

6.2 3DOF Results

First, we show the results of solving the convex program in Equation 2 in Figure 1. The parameters in Table 1 are normalized in order to help the solver converge. Note the min-max thrust property

New Shepard		Convex Problem Constraints	
Dry mass (kg)	20,569	Glide slope γ ($^\circ$)	20
Wet mass (kg)	27,000	Thrust Angle θ ($^\circ$)	27
I_{sp} (s)	260	**Initial & Final Conditions**	
Max throttle	0.8	r_0 (m)	[1500, 500, 2000]
Min throttle	0.1	v_0 (m/s)	[50, -30, -100]
Max thrust (N)	490,000	r_f (m)	[0, 0, 0]
Inertia (10^3 kg/m^2)	[696.390, 696.390, 42.78]	v_f (m/s)	[0, 0, 0]

Table 1: Parameters used in the results



(a) Trajectory of the Lander

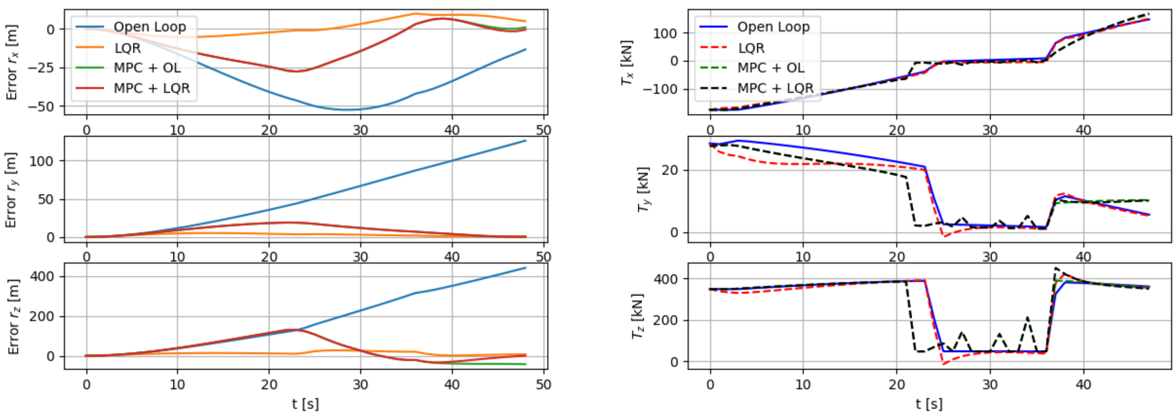
(b) Control Magnitude and Mass of the Lander.

Figure 1: Solution of the 3DOF minimum-fuel convex program for our scenario.

that the convex program finds for this trajectory. This is expected, and is a property of minimum-fuel trajectories, as discussed in Açıkmeşe and Ploen [2].

We then take this trajectory and track it using LQR. Drag is treated as a disturbance by the LQR controller. The simulation is run using ode45; since the convex program uses trapezoidal integration, this introduces integration error. Both these disturbances are tracked against using LQR. These same disturbances are applied to MPC, where the algorithm recomputes the minimum fuel trajectory at each time step to account for these disturbances. For this simulation, we set the ballistic coefficient $B = 2.5$. Note that in Figure 2a, the error is taken to be the difference in the position that this method has at time t, and the position prescribed by the trajectory calculated in Figure 1a.

The resulting end states are given in Table 2. We see that both of the open-loop methods perform worse than the LQR methods as is expected. Note that LQR and MPC actually expend less fuel than the prescribed trajectory, which occurs due to drag taking energy out of the lander. Finally we see that between MPC + LQR and LQR there is no clear winner. They both use approximately the same amount of fuel, MPC + LQR is closer to the target, but LQR has more effectively zeroed out its velocity. It is worth noting that MPC guarentees that constraints are met,



(a) Error history of each method

(b) Control inputs of each method

Figure 2: Solution of the minimum-fuel convex program for our scenario.

whereas LQR does not.

Method	Final Wet Mass (kg)	Final Position Error (m)	Final Velocity Error (m/s)
Trajectory	21,336.02	1.30×10^{-10}	6.60×10^{-9}
Open Loop	21,332.00	458.14	12.30
LQR	21,457.77	8.81	0.66
MPC + OL	21,480.33	42.01	1.62
MPC + LQR	21,448.22	0.76	2.73

Table 2: Final values for each method

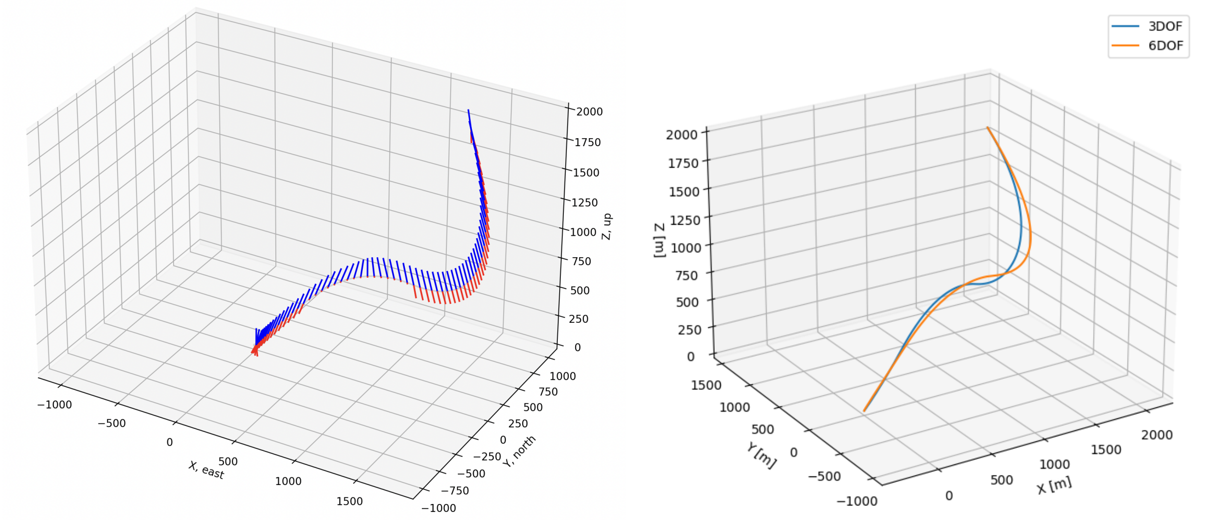
6.3 6DOF Results

We also provide results for the 6DOF minimum fuel problem. The same parameters in Table 1 are used and the resulting trajectory is shown in Figure 3a. As can be observed in Figure 3b, the 3DOF and 6DOF trajectories are quite similar. Open loop tracking of this trajectory is not realizable due to attitude integration errors. As the attitude error accumulates, the thrust vector begins to drift which pushes the vehicle further from the prescribed trajectory. However, by using a closed-loop LQR controller, we can achieve near-perfect tracking in the absence of external disturbances. Position and orientation error relative to the trajectory are shown in Figure 4a. Note that this trajectory also exhibits the bang-bang control characteristic of minimum fuel solutions.

7 Conclusions and Future Work

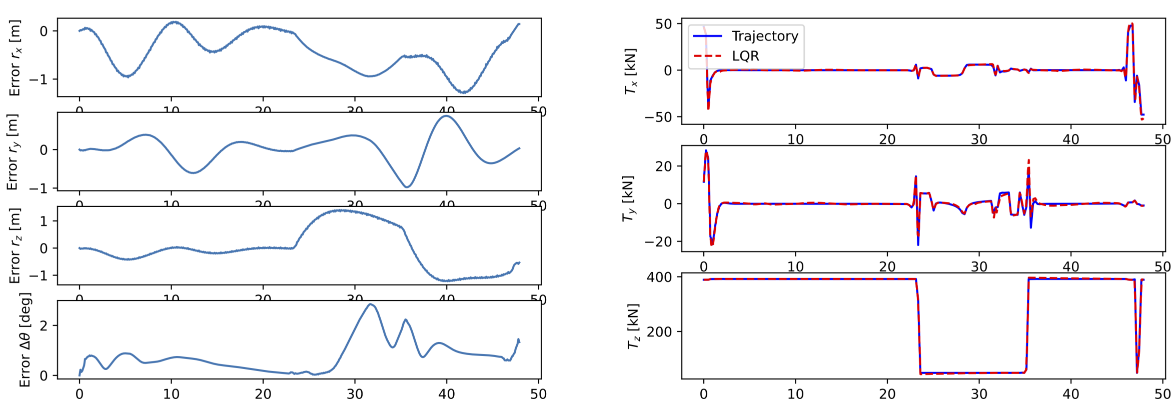
In this work we have implemented various models and control systems for the large divert rocket landing problem. It is clear that planning and then performing open loop control is not sufficient to get close to the goal, and closed-loop approaches are necessary. We compare the performance of LQR and MPC for closed loop control with 3DOF dynamics, and we show that while LQR has an advantage in computation cost, MPC has an advantage in constraint handling. Finally, we solve the 6DOF minimum fuel problem with successive convexification. We show the resulting trajectory is similar to the 3DOF solution and can also be followed with LQR.

In the future, work that expands MPC so that it is feasible for the whole trajectory must be investigated. As is mentioned, other authors have implemented a minimum-error landing problem when the minimum-fuel problem becomes infeasible [5], and this may be a good starting point for this work. Finally, trajectory following in the presence of other disturbances and model uncertainty can be explored.



(a) 6DOF trajectory with attitude and thrust vector (b) Comparison of 3DOF and 6DOF trajectories

Figure 3: Solution of the 6DOF minimum-fuel problem for our scenario.



(a) 6DOF LQR tracking error (b) 6DOF LQR thrust

Figure 4: Tracking of the 6dof minimum-fuel trajectory.

References

[1] Allan R. Klumpp. Apollo lunar descent guidance. *Automatica*, 10(2):133–146, 1974. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(74)90019-3. URL https://www.sciencedirect.com/science/article/pii/0005109874900193.

[2] Behçet Açıkmeşe and Scott R. Ploen. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007. doi: 10.2514/1.27553. URL https://doi.org/10.2514/1.27553.

[3] Lars Blackmore, Behçet Açıkmeşe, and Daniel P. Scharf. Minimum-landing-error powered-descent guidance for mars landing using convex optimization. *Journal of Guidance, Control, and Dynamics*, 33(4):1161–1171, 2010. doi: 10.2514/1.47202. URL https://doi.org/10.2514/1.47202.

[4] Behçet Açıkmeşe, John M. Carson, and Lars Blackmore. Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Transactions on Control Systems Technology*, 21(6):2104–2113, 2013. doi: 10.1109/TCST.2012.2237346.

[5] Behçet Açıkmeşe, M Aung, Jordi Casoliva, S Mohan, Andrew Johnson, Daniel Scharf, David Masten, Joel Scotkin, Aron Wolf, and M.W. Regehr. Flight testing of trajectories computed by g-fold: Fuel optimal large divert guidance algorithm for planetary landing. volume 148, 01 2013.

[6] Michael Szmuk and Behcet Acikmese. Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time. In *2018 AIAA Guidance, Navigation, and Control Conference*, January 2018. doi: 10.2514/6.2018-0617. URL http://arxiv.org/abs/1802.03827.

[7] Carlos Sánchez-Sánchez and Dario Izzo. Real-time optimal control via deep neural networks: Study on landing problems. *Journal of Guidance, Control, and Dynamics*, 41(5):1122–1135, 2018. doi: 10.2514/1.G002357. URL https://doi.org/10.2514/1.G002357.

[8] Lin Cheng, Zhenbo Wang, Yu Song, and Fanghua Jiang. Real-time optimal control for irregular asteroid landings using deep neural networks. *Acta Astronautica*, 170:66–79, 2020. ISSN 0094-5765. doi: https://doi.org/10.1016/j.actaastro.2019.11.039. URL https://www.sciencedirect.com/science/article/pii/S0094576520300151.