Convex Model Predictive Control for the Soft Landing Problem

Ibrahima Sory Sow
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
isow@andrew.cmu.edu

***Abstract*—Planetary soft landing is a crucial aspect of the Entry, Descent, and Landing (EDL) process for spacecraft, particularly with the growing number of scientific missions to outer planets and the need for reusable spacecraft. The Powered Descent Guidance (PDG) algorithm is an essential component of this process and can be expressed as a finite horizon optimal control problem with nonconvex state and control constraints. In this work, we utilize recent convexification techniques to facilitate onboard real-time computation of the PDG algorithm for the soft landing problem. By introducing a slack variable and a change of variable, we relax the nonconvex thrust bounds and pointing constraints and eliminate the nonlinearities in the dynamics. As a result, the original nonconvex problem is transformed into a convex problem where the optimal solution is also optimal for the original problem. We implement these algorithms using model predictive controllers, demonstrate their performance in the presence of disturbance, and highlight recursive feasibility problems. Finally, we propose potential avenues for improvement.**

***Index Terms*—optimal control, trajectory optimization, dynamics, rocket landing**

I. Introduction

The planetary soft landing problem refers to the final stage of the planetary Entry, Descent, and Landing (EDL) process, wherein a powered spacecraft must safely land on a predetermined location on the surface of a planet. After the atmospheric re-entry where the landing accuracy (distance between the targeted and actual landing location) has been degraded [1], the lander starts from an uncertain high-velocity state and attempts to achieve a soft, or impact-free, landing as close as possible to the target location (pinpoint landing) with zero velocity while considering multiple state and control constraints. This problem is also known as the Powered Descent Guidance (PDG) problem as it involves using thruster firings to control the descent. The optimal control community has long considered this problem a benchmark and has recently regained interest in it due to the increased number of exploration missions to outer planets and the growing importance of spacecraft reusability on Earth.

PDG algorithms, typically formulated as finite horizon optimal control problems, aim to optimize spacecraft trajectories for both landing accuracy and fuel efficiency. The latter is a significant concern within the aerospace community, given the limited availability of onboard fuel. These algorithms must account for various state and control constraints, including satisfying the governing physics, respecting the thrust limitations, or adhering to attitude constraints. Thrusters have design limitations on the maximum achievable thrust and are not easily throttled off after ignition, presenting a notable challenge in the fuel optimization problem. Moreover, the maximum pointing angle of thrusters relative to the spacecraft imposes restrictions on maneuverability during the descent. For instance, typical gimbaled thrust vectoring systems have a range of $\pm 12^\circ$ [2]. Onboard sensors often require a specific viewpoint, constraining further the achievable attitude of the spacecraft. This was exemplified in the recent Mars 2020 mission, where a direct surface view was crucial for onboard sensors to facilitate terrain-relative navigation [3]. Lastly, as modern hardware has advanced to enable onboard computation of control laws, PDG algorithms must meet numerical efficiency requirements and provide strict performance guarantees.

The development of reliable PDG algorithms is a challenging task due to the difficulty of expressing the aforementioned constraints in a suitable form. In particular, nonconvexities arise from thruster minimum bounds and thrust vector pointing constraints, while the time-varying mass introduces nonlinearities in the dynamics. Although some nonlinear optimization techniques have been attempted [4], they fail to provide guarantees on global optimum or convergence rate. Nonetheless, alternative approaches have been successfully demonstrated on real spacecraft and have achieved impressive results.

The Apollo Guidance algorithm [5] relied on simple closed-form polynomial solutions to ensure a successful landing, but this approach restricted the set of initial starting conditions and did not include any of the aforementioned constraints. Recent works on convexification [6] and [7] have paved the way for onboard numerical algorithms based on convex optimization. These works transform the original nonconvex problem into a convex problem by tightly relaxing the nonconvex thrust constraints and proving that the optimal solution of the relaxed problem is also the optimal solution of the original one, using a method referred to as "lossless convexification". The algorithm, coined "G-FOLD" (Guidance algorithm for Fuel Optimal Large Diverts), was successfully demonstrated in an Earth-based testbed. The aerospace company SpaceX further experimented with the same approach to land their reliable Falcon 9 on Earth. To date, SpaceX has successfully landed Falcon 9 rockets 179 times. However, it has been revealed that SpaceX uses CVXGEN [8], an efficient QP solver, for their

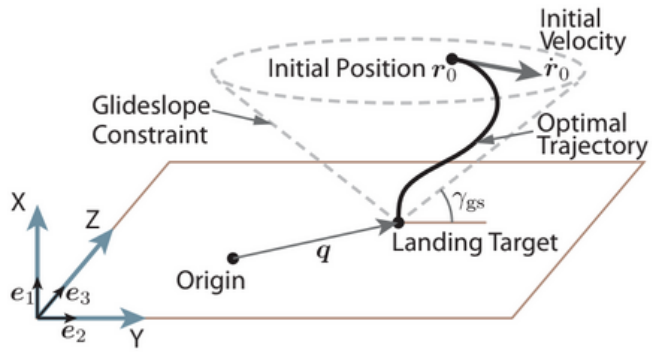

Fig. 1. Reference frame used throughout this paper. The cone of the glideslope constraints ensures a safe approach to the landing target.

onboard optimization, which suggests a simpler approach to achieve Falcon 9 landings.

This report compares different receding horizon controller approaches leveraging recent convexification works [6] for the descent phase of spacecraft. The powered descent phase is subject to uncertainties from unknown parameters in the dynamics and ignored disturbance forces, so we introduce disturbances in the control actions rather than state disturbances. We propose three model-predictive control (MPC) approaches and assess their performance in the presence of disturbances:

- MPC on the minimum landing error problem
- MPC on the fuel optimal problem
- MPC on a prioritized framework solving the two first problems

Section II describes the dynamics model and problem formulations, while Section III presents the experiments and comparisons performed. In Section IV, we discuss and analyze the obtained results, and the conclusion summarizes the paper and explores other areas of improvement and research directions. The code implementation of the project is available at https://github.com/Ibrassow/soft_landing_mpc.

II. Methods

The planetary soft landing problem can be expressed as finding a state and control trajectory, $(r(t), \dot{r}(t))$ and T_c, that minimizes fuel consumption while bringing the lander from an initial state $(r_0, \dot{r}_0)$ to a target location with zero velocities.

A. Original problem formulation

The planetary lander is modeled using a simplified point mass model and a thrust vector as control. In this work, we decouple the translational dynamics from the rotational dynamics and assume that a low-level attitude controller has a fast enough bandwidth to keep the assumption reasonable. Vehicle orientation constraints are indirectly applied by restricting the directions of the thrust vector. The considered scenario assumes a uniform gravity field, a constant rotation rate, and negligible aerodynamic forces during the powered descent landing. The dynamics of the system are modeled by the following equations:

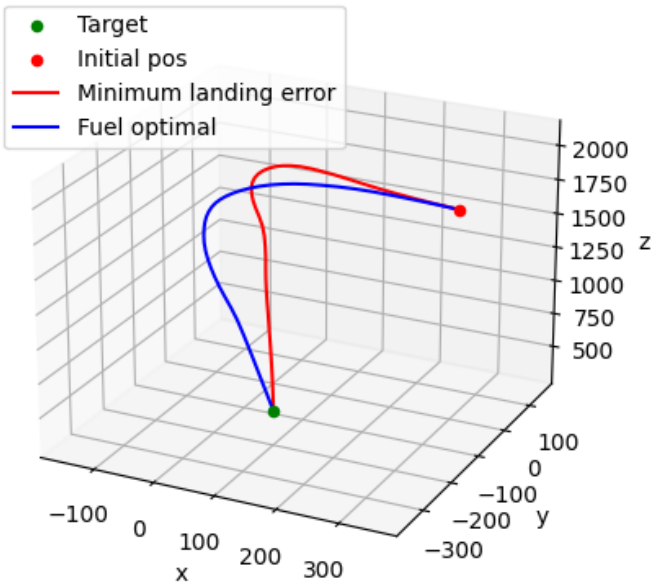

Fig. 2. Three-dimensional trajectories of the minimum landing error and minimum fuel usage problem.

$$\begin{aligned} \dot{x} &= A(\omega)x(t) + B\left(g + \frac{T_c(t)}{m(t)} + w(t)\right) \\ \dot{m} &= -\alpha\|T_c(t)\| \end{aligned} \tag{1}$$

where $x(t) = (r(t), \dot{r}(t))$, $m(t)$ is the mass of the spacecraft, $w(t)$ represents force disturbances

$$\begin{aligned} A(\omega) &= \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ -S(\omega) & -2S(\omega) \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} \\ S(\omega) &= \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \end{aligned} \tag{2}$$

$\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the angular velocity of the planet, $\boldsymbol{g}$ is the constant gravity vector, $\boldsymbol{g}$ is the gravitational vector, $\alpha > 0$ is the mass depletion rate. Note that the rocket equation was used to model the mass depletion due to fuel consumption.

The powered descent guidance algorithm must satisfy a set of state and control constraints as outlined in Section I. The state constraints consist of a glideslope constraint and a maximum speed constraint. The glideslope constraint, illustrated in Fig. 1, ensures that the lander approaches the target from a safe distance from the ground. The upper bound on speed, V_{max}, restricts the velocity to the subsonic regime for planets with an atmosphere to avoid control thrusters becoming unreliable. Both of these convex state constraints define the feasible set of positions and velocities for the lander, denoted by $\boldsymbol{X}$, described by

$$\boldsymbol{X} = (\boldsymbol{r}, \dot{\boldsymbol{r}}) \in \mathbb{R}^6 : \|\dot{\boldsymbol{r}}\| \leq V_{\max}, \\ \|\boldsymbol{E}(\boldsymbol{r} - \boldsymbol{r}(t_f)) - \boldsymbol{c}^T \boldsymbol{r} - \boldsymbol{r}(t_f)\| \leq 0 \quad (3)$$

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{c} = \frac{\boldsymbol{e_1}}{\tan \gamma_{gs}}, \quad \gamma_{gs} \in (0, \frac{\pi}{2})$$

where γ_{gs} is the minimum glideslope angle as displayed in Fig. 1 and c defines the corresponding cone having its vertex at $\boldsymbol{r}(t_f)$). Furthermore, the problem requires obeying the following initial and final state constraints

$$m(0) = m_0, \quad m(t_f) \geq m_0 - m_f > 0 \quad (4)$$

$$\mathbf{r}(0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0 \quad (5)$$

$$\mathbf{e}_1^T \mathbf{r}(t_f) = 0, \quad \dot{\mathbf{r}}(t_f) = \mathbf{0} \quad (6)$$

The control constraints comprise three distinct constraints: an upper bound on the thrust, a lower bound on the thrust, and a limitation on the deviation angle from the direction vector $\hat{\boldsymbol{n}}$, with $\|\hat{\mathbf{n}}\| = 1$. The second constraint transforms the set of allowable thrust values into a nonconvex set, while the third constraint also becomes nonconvex when $\theta > \pi/2$. The constraints can be summarized as follows:

$$0 < T_{\min} \leq \|\mathbf{T_c}\| \leq T_{\max}, \quad \hat{\mathbf{n}}^T \frac{\mathbf{T_c}(t)}{\|\mathbf{T_c}(t)\|} \geq \cos(\theta) \quad (7)$$

Given a target location on the surface of the planet (0,q), where $q \in \mathbb{R}^2$ is the plane coordinates of the landing location, and the described dynamics and constraints, the planetary soft landing problem can be formulated either as the nonconvex minimum landing error problem

$$\min_{\boldsymbol{T_c}} \|\boldsymbol{E}\boldsymbol{r}(t_f) - \boldsymbol{q}\| \quad (8)$$
$$\text{subject to } (2), (4), (5), (6), \text{and } (7) \quad \forall t \in [0, t_f]$$

or the nonconvex minimum fuel problem

$$\min_{\boldsymbol{T_c}} \quad \int_0^{t_f} \alpha \|\boldsymbol{T_c}(t)\| dt \quad (9)$$
$$\text{subject to} \quad (2), (4), (5), (6), (7)$$
$$\text{and } \|\boldsymbol{E}\boldsymbol{r}(t_f) - \boldsymbol{q}\| \geq \|\boldsymbol{d}^* - \boldsymbol{q}\| \quad \forall t \in [0, t_f] \quad (10)$$

where $\mathbf{d}^*$ is the final position of the minimum landing error problem.

The prioritized framework consists of solving first the minimum landing error problem to get the predicted achievable landing locations given the current state. The resulting target location is then passed as a constraint (10) to the second optimization problem, which optimizes the trajectory for minimum fuel consumption.

B. Convexification

The above formulation suffers from nonconvexities in both the dynamics and the thrust constraints. When discretized, the differential equation that represents the spacecraft's mass depletion leads to nonconvex nonlinear equality constraints. The lower bound on thrust, $\rho_1 > 0$, causes the set of allowable thrust to become nonconvex. Even in the optimistic case of $\rho_1 = 0$, the pointing constraint is nonconvex for $\theta > \pi/2$. Using the key results presented in [6] [7], we can safely relax the original nonconvex problem by introducing a new slack variable Γ which replaces $\|\boldsymbol{T_c}(t)\|$ and leads to the new following set of control constraints

$$\|\mathbf{T_c}(t)\| \leq \Gamma(t), \qquad 0 < T_{\min} \leq \|\Gamma(t)\| \leq T_{\max} \quad (11)$$
$$, \quad \hat{\mathbf{n}}^T \mathbf{T_c}(t) \geq \Gamma(t) \cos(\theta)$$

This reformulation changes the minimization objective of the fuel optimal problem (9) into

$$\min_{\boldsymbol{T_c}} \int_0^{t_f} \Gamma(t) dt \quad (12)$$

Furthermore, an additional change of variables is performed in order to remove the nonlinearities caused by the term $\boldsymbol{T_c}/m$ in the dynamics

$$\sigma = \frac{\Gamma}{m}, \quad u = \frac{\boldsymbol{T_c}}{m}, \quad z = \ln(m) \quad (13)$$

which reformulates the mas depletion dynamics as follows

$$\dot{z} = \frac{\dot{m}(t)}{m(t)} = -\alpha\sigma \quad (14)$$

The spacecraft dynamics are now a set of linear differential equations and a further approximation is needed to solve the problem with a convex solver since the control thrusters became nonconvex. The reader is referred to [9] for more details on the approximation approach for the implementation.

III. Results

This section illustrates numerical examples of three model predictive controllers based on the problem formulation of Section II. All examples are subject to additional control disturbances w aiming to replicate the effects of fluid slosh, structural oscillations, and wind as external disturbances. w is a 1 by 3 vector whose components are drawn at each time step k from a normal (Gaussian) distribution of mean 0 and standard deviation of 0.5 m/s^2. The continuous dynamics have been discretized using the exponential map [10] with a timestep of 1 sec. In the current problem formulation, the time is not an optimization variable, which further constrains the landing to finish in a predefined final time amounting to the number of knot points. Future work will aim at reformulating the soft landing problem as a free-time problem. We solved the convex optimization problems using CVXPY [11] and the solver ECOS [12]. In all experiments, simulation parameters were fixed to

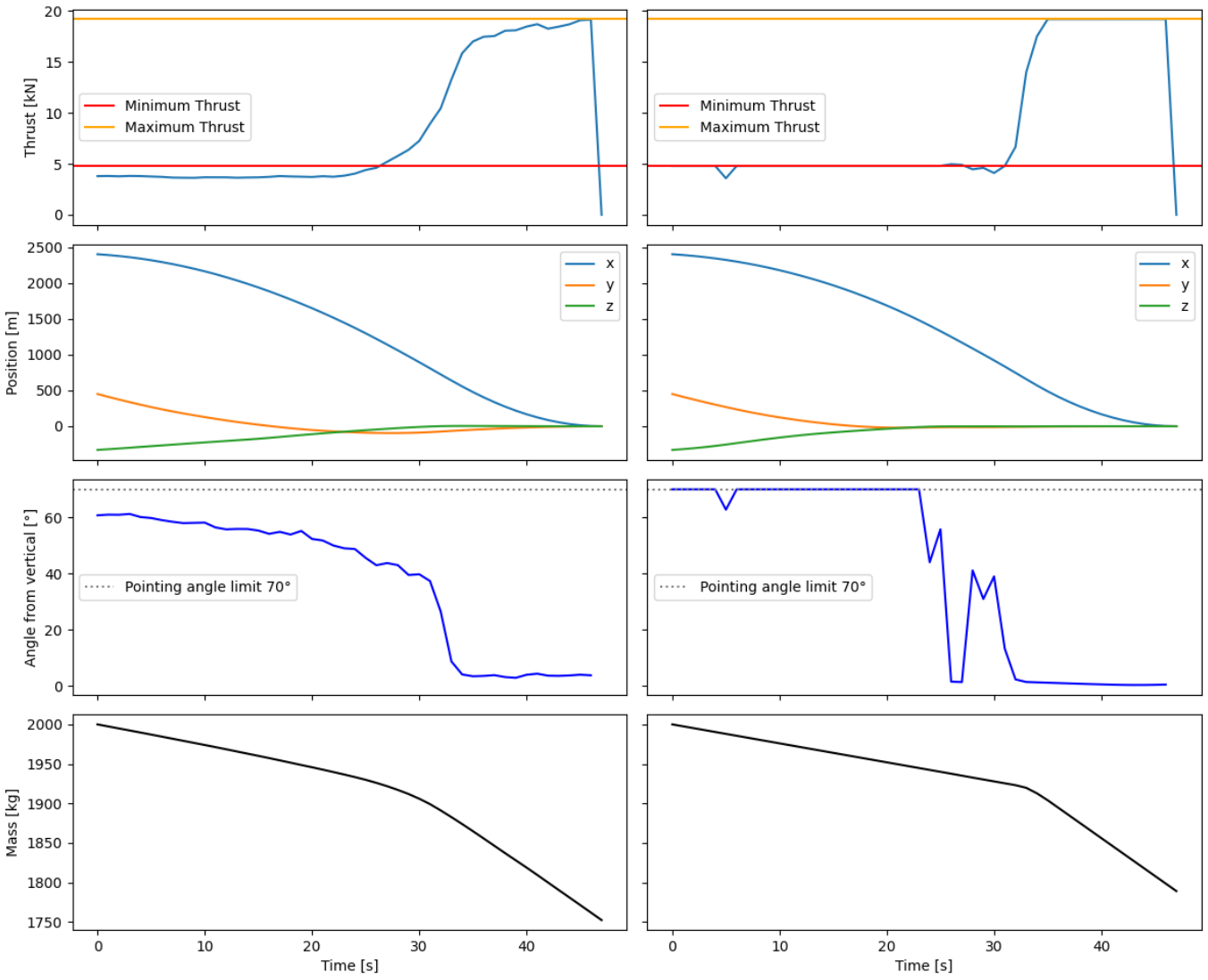

Fig. 3. Direct comparison between the minimum landing error and minimum fuel usage problem of the thrust, position, pointing angle of the thrust vector, and the mass at each time step.

- $m_0 = 2000$ kg
- $m_f = 300$ kg
- $T_{\max} = 24000$ N
- $\rho_1 = 0.2\ T_{\max}$
- $\rho_2 = 0.8\ T_{\max}$
- $\gamma_{\text{gs}} = 30°$
- $V_{\max} = 90$ m/s
- $\theta = 70°$

The initial position and velocity of the spacecraft $(r_0, \dot{r}_0)$ for this comparison are

- $r_0 = (2400, 450, -330)$ m
- $\dot{r}_0 = (-10, -40, 10)$ m/s

We choose Martian conditions for the gravity vector and angular rotation, according to the frame of reference in Fig. 1

- $w = (2.53 \times 10^{-5}, 0, 6.62 \times 10^{-5}))$ rad/s
- $g = (-3.71, 0, 0)$ m/s^2

Fig. 2 depicts the spatial trajectory of each model predictive controller for the same conditions as defined above while Fig. 3 compares the thrust, position, pointing angle of the thrust vector, and the mass evolution over the whole trajectory for the two controllers. The number of knot points was set to 48.

For the prioritized framework, simulation parameters pertaining to the attitude and the initial conditions of the spacecraft $(r_0, \dot{r}_0)$ have been chosen far from the landing target location as an illustration of the capabilities of the algorithms

- $r_0 = (1840, 1020, -650)$ m
- $\dot{r}_0 = (-48, -11, 7)$ m/s
- $\gamma_{\text{gs}} = 70°$
- $\theta = 50°$

Fig. 4 illustrates the three-dimensional trajectory of the spacecraft running this scheme while Fig. 5 shows the time evolution of the thrust, state, pointing angle of the thrust vector

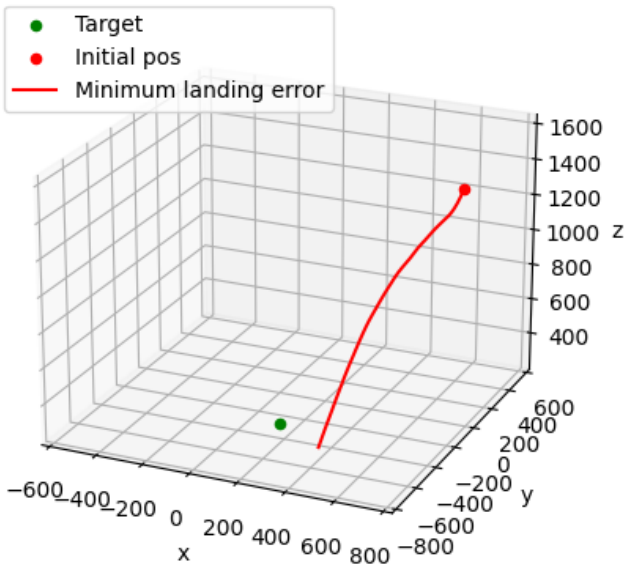

Fig. 4. Plotted three-dimensional trajectory of the prioritized approach. We can see that the spacecraft could not reach the target location but followed a fuel-optimal path to the closest attainable location.

and the mass over the whole trajectory.

IV. Discussion

First, it should be noted that the initial conditions and disturbances were carefully chosen so that the state remains in the feasible regions of both controllers. Indeed, we noticed that the introduction of disturbances breaks the recursive feasibility of the optimization problem. As such, results from Fig. 2 and Fig. 3 came from receding horizon controllers running only for the first 3/4 of the trajectory while the rest of the trajectory had to rely on the previously obtained optimal solution at the previous steps. The prioritized shown in Fig. 4 framework performed better but still suffer, to a lesser extent, from the recursive feasibility problem. A proposed solution would be to use the prioritized framework to generate a reference trajectory and track it with a lighter and simpler controller, i.e. quadratic objective function and linearized dynamics.

Fig. 3 gives more detailed insights into the trade-off between accuracy and fuel efficiency. A notable observation is the bang-bang nature of the thrust usage by the fuel optimal problem. Coupled with its tendency to lie on the limit of the allowable pointing angle, it seems that the controller use as much as possible the natural dynamics to orient itself and relies on one last corrective maneuver to reach the target, even if it means an aggressive one. In contrast, the minimum landing error problem tries to produce a less efficient, but direct, path to the goal. The fuel optimal problem saved 36.67 kg compared to the minimum landing problem. While these results might seem small, it should be reminded that the current initial conditions might not be the best suited to produce appreciable differences.

While optimizing for precision is not always the best approach when fuel-constrained, absolute fuel efficiency sacrifices too much of the accuracy of the pinpoint landing aimed solely at the target. The prioritized approach strikes a good balance by ensuring a fuel-optimal trajectory that is the closest to the target landing location when run in a receding horizon way. It handles better uncertainties but does so for a more restricted set of initial conditions, which questions its usability in real-case scenarios where initial conditions are hard to predict.

V. Conclusion

We conducted a head-to-head comparison of the minimum landing error and fuel optimal problems and found that a balance between the two respective optimization objectives was necessary when both had equal weights. As an alternative, we introduced a prioritized framework as a receding horizon controller, where we solved the first problem for the closest reachable target location that is fed back into the constraint of the second problem. However, we demonstrated limitations for real-time onboard implementation of these three approaches in the face of uncertain dynamics, as they lost their recursive feasibility. Consequently, we propose using the prioritized framework as a one-time planner and handling the tracking of the generated trajectory through a simpler and lighter quadratic receding horizon tracker for future implementation. This method would handle more important disturbances and should not suffer from the feasibility problem.

More generally, future work directions include the implementation of a fully constrained case and the establishment of formal proofs of robustness guarantees, i.e. being able to state performance guarantees with formal bounds on the uncertainty. Additionally, we aim to improve the fidelity of the current dynamics model by replacing the lumped mass model with a full 6 DOF dynamics model of the spacecraft, which would allow us to consider the pointing constraints directly in the optimization scheme.

References

[1] K. S. Tracy, G. Falcone, and Z. Manchester, “Robust entry guidance with atmospheric adaptation,” AIAA SCITECH 2023 Forum, 2023. doi:10.2514/6.2023-0301

[2] G. P. Sutton and O. Biblarz, “Thrust Vector Control,” in Rocket Propulsion Elements, Hoboken, NJ: John Wiley amp; Sons, Inc, 2017, pp. 671–689

[3] A. E. Johnson et al., “Real-time terrain relative navigation test results from a relevant environment for Mars Landing,” AIAA Guidance, Navigation, and Control Conference, 2015. doi:10.2514/6.2015-0851

[4] U. Topcu, J. Casoliva, and K. D. Mease, “Minimum-fuel powered descent for Mars pinpoint landing,” Journal of Spacecraft and Rockets, vol. 44, no. 2, pp. 324–331, 2007. doi:10.2514/1.25023

[5] A. R. Klumpp, “Apollo Lunar Descent Guidance,” Automatica, vol. 10, no. 2, pp. 133–146, 1974. doi:10.1016/0005-1098(74)90019-3

[6] B. Acikmese, J. M. Carson, and L. Blackmore, “Lossless convexification of nonconvex control bound and pointing constraints of the Soft Landing Optimal Control Problem,” IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pp. 2104–2113, 2013. doi:10.1109/tcst.2012.2237346

[7] J. M. Carson, B. Acikmese, and L. Blackmore, “Lossless convexification of powered-descent guidance with non-convex thrust bound and pointing constraints,” Proceedings of the 2011 American Control Conference, 2011. doi:10.1109/acc.2011.5990959

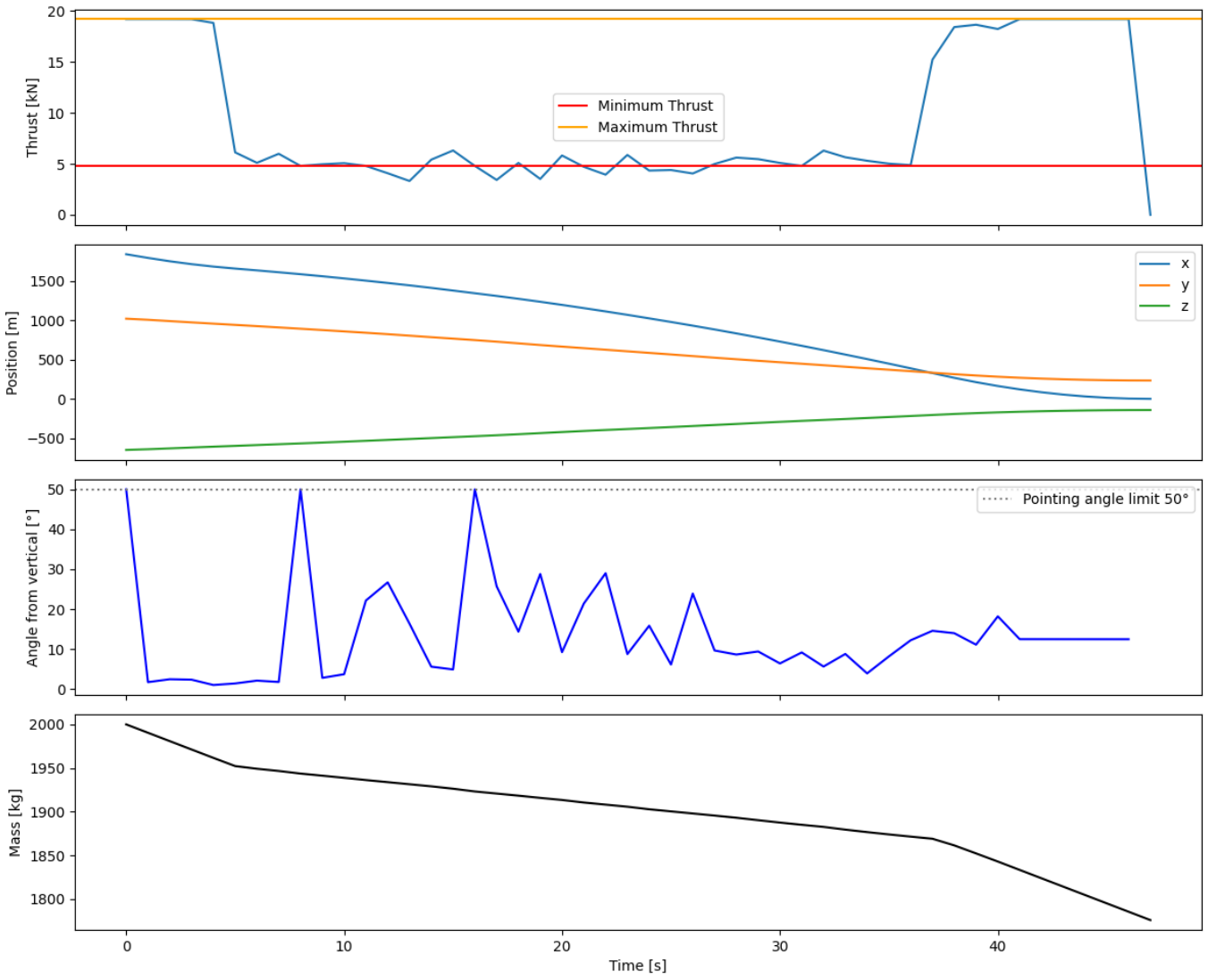

Fig. 5. Time evolution of the thrust, position, pointing angle of the thrust vector, and mass of the prioritized approach at each time step.

[8] L. Blackmore, "Autonomous Precision Landing of Space Rockets," National Academy of Engineering, Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium, 2017. Washington, DC: The National Academies Press. doi: 10.17226/23659.

[9] B. Acikmese and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars Landing," Journal of Guidance, Control, and Dynamics, vol. 30, no. 5, pp. 1353–1366, 2007. doi:10.2514/1.27553

[10] Tracy, K. and Manchester, Z., 2020, August. Model-predictive attitude control for flexible spacecraft during thruster firings. In AAS/AIAA Amer. Inst. Aeronaut., Astrodynamics Specialist Conf..

[11] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," J. Mach. Learn. Res., vol. 17, 2016.

[12] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An SOCP solver for Embedded Systems," 2013 European Control Conference (ECC), 2013. doi:10.23919/ecc.2013.6669541