Evaluating Numerical Solving Methods for Optimal Powered Descent

Anyi X. Lin*, Favour A. Adekola† Joseph D. Farkas‡ Parker R. Melling‡ Niara Marwah‡
Georgia Institute of Technology, Atlanta, GA, 30332

In autonomous navigation for vertical take-off and landing (VTOL) rockets, the generation of reliable and efficient paths for the vehicle to adhere to remains a difficult task. Powered descent trajectory optimization to achieve minimal fuel consumption is a nonconvex problem with nonlinear constraints such as minimum throttle and thrust pointing constraints. The lossless convexification of the soft landing problem relaxes these nonconvex constraints while ensuring that the optimal solution to the relaxed problem remains the same as the optimal solution to the original nonconvex problem. This allows for efficient numerical solving methods to be employed. In this paper, we analyze two numerical optimization methods: time-based discretization and pseudo-spectral methods using Chebyshev polynomial parameterization. Time-based discretization splits the problem into many time steps, enforcing constraints at each one. This method is more sensitive to the number of time steps, resulting in a large number of variables per problem and more iterations necessary for convergence. Parameterization by Chebyshev polynomials enforces constraints and dynamics at a fewer number of nodes with global differentiation matrices. This results in fewer iterations necessary for the same level of accuracy, but longer calculations per iteration. We assess the effectiveness of these methods by the number of variables and the runtime of each algorithm to determine the best approach for realtime onboard computation.

I. Nomenclature

t_f	=	time of flight for powered descent
$r(t)$	=	position at time t, where $r \in \mathbb{R}^3$
$v(t)$	=	velocity at time t, where $v \in \mathbb{R}^3$
$m(t)$	=	mass at time t, where $m \in \mathbb{R}$
$T_c(t)$	=	net thrust vector at time t, where $T_c \in \mathbb{R}^3$
$\Gamma(t)$	=	slack variable that bounds thrust magnitude, where $\Gamma \in \mathbb{R}$
g	=	gravitational acceleration vector of Earth, where $g \in \mathbb{R}^{\nvDash}$
α	=	mass consumption rate
I_{sp}	=	specific impulse for thrusters
$\hat{n}$	=	thrust direction
S	=	matrix defining convex constraints
ϕ	=	gimbal angle for the thrusters
c	=	clone slope vector
ρ_1	=	lower bound on thrust magnitude
ρ_2	=	upper bound on thrust magnitude
θ_{alt}	=	glide slope angle
$\tilde{\theta}_{alt}$	=	bound on the allowable glide slope angle
x	=	rocket position and velocity vector
D	=	Chebyshev differentiation matrix
z_0	=	natural log of mass

*GTPL GNC Lead, AIAA Undegraduate Student Member
†GTPL GNC Vice-Lead, AIAA Undergraduate Student Member
‡GTPL GNC Member, AIAA Undergraduate Student Member

II. Introduction

When building software for autonomous systems, engineers have to first deal with how to design appropriate perception, controls, and path planning for their applications. While perception and controls bear the task of predicting the present or immediate future, path planning and navigation focuses on foretelling the future for the best course of action. Additionally, requirements for real-time functionality, adherence to dynamic constraints, and minimized cost further increase the expected complexity of navigation. With this, it becomes clear that the generation of a reliable, efficient path for a system to take is a non-deterministic polynomial-time ("NP") hard problem[1].

There has been a recent demand for reusable rockets and, with it, greater demand for optimal navigation algorithms for these vehicles. Companies such as Blue Origin, SpaceX, and Stoke Space, have spent billions of dollars designing rockets for vertical take-off and vertical landing (VTVL). These rockets are unique and have a variety of constraints that need to be taken into account. A VTVL rocket often is required to remain upright throughout its flight and to perform a soft landing towards the end of its flight to remain reusable. To aid in this, it is common to utilize a gimbal on the thrusters to allow controllability of the thrust vector of the vehicle. This thrust vectoring is often limited in range of motion, and most systems offer around 5° to 15°, depending on their design. Finally, the mass of a rocket is constantly decreasing during flight as it burns through fuel to produce thrust. Given that propellant is a significant cost driver, a well-designed trajectory should aim to minimize the total fuel consumption for a given flight path.

For an autonomous rocket, we could naively construct a linear path from our current position to our goal; however, a linear path is often unachievable given the dynamics of a VTVL rocket. Even if the rocket could meet the demands of a linear path, it would still suffer from a non-optimal fuel consumption. These issues are improved by approaches such as spline-based pathing [2] or Successive Convexification [3], but these algorithms are susceptible to getting stuck at local minima during optimization. Fortunately, Lossless Convexification (LCvx) guarantees the convergence of the soft landing problem to a global minima by relaxing the non-convex minimum thrust and thrust pointing constraints. This ensures that the optimal solution to the relaxed convex second-order cone problem (SOCP) is the same as the original problem, allowing us to utilize numerical solving methods for the SOCP to get our solution in polynomial time[4].

Propulsive Landers at Georgia Tech (GTPL) is developing 1000 N and 2500 N hybrid rocket engine vehicles capable of VTVL, and as a part of these goals the student organization needs to decide on which numerical method approach to utilize to solve their optimal landing SOCP. In this paper, we compare the differences between solving the SOCP using time-based discretization with Zero-Order Hold (ZOH) versus pseudospectral parameterization through Chebyshev-Gauss-Lobatto (CGL) nodes. At the end, we analyze the performance of the two methods to decide on which is best suited for real-time, onboard computation.

III. Problem Statement

In our approach to the powered descent guidance problem, we take a very similar approach to Açıkmeşe and Ploen[4]. We model the rocket as a point mass, and our primary states include position, velocity, and mass. Our only control is the thrust vector, and to aid in the convexification, we introduce the slack variable ($\Gamma(t)$). Since our current rocket is designed to reach an altitude of 50 meters, we model the gravity field as a constant and the planetary rotation as zero. With this in mind, we do not project our rocket to reach high velocities either, so we neglect aerodynamics in our models. Angular dynamics are also neglected in the model, as they evolve much more rapidly than translational dynamics and attitude control authority is typically much higher than translational control authority. This means that the angular dynamics will not affect the translational dynamics greatly. The three primary equations governing our 3 degrees of freedom (DoF) approach to the powered descent guidance problem include:

$$\dot{r}(t) = v(t) \tag{1}$$

$$\dot{v}(t) = g + \frac{T_c(t)}{m(t)} \tag{2}$$

$$\begin{aligned} \dot{m}(t) &= -\alpha |T_c(t)| \\ \alpha &= \frac{1}{I_{sp} g} \end{aligned} \tag{3}$$

The primary objective is to minimize fuel consumption (minimize $\int_0^{t_f} |T_c(t)| dt$), which is the same as maximizing the final mass. The time minimizing solution is identical to the fuel consumption minimizing solution for the 1-D version of the soft landing problem[5]. Since it has previously been demonstrated that these two methods are similar, our approach minimizes the overall flight time. The physical limitations of the problem also need to be translated into mathematical constraints. One such constraint is the thrust magnitude - a minimum and maximum throttle the rocket is allowed to achieve:

$$0 < \rho_1 \leq |T_c(t)| \leq \rho_2 \tag{4}$$

Since ρ_1 creates a nonconvexity set, we introduce $\Gamma(t)$ such that $|T_c(t)| \leq \Gamma(t)$. Therefore, our minimum fuel problem becomes minimizing $\int_0^{t_f} \Gamma(t)dt$. Despite the slack variable encompassing a physically impossible region $(0 - \rho_1)$, the solution to the problem doesn't change [6]. Blackmore, Açıkmeşe, and Carson proved that the optimal Hamiltonian is maximized only when the constraint is active using Pontryagin's Maximum Principle, meaning that the convexification expands the possible thrust magnitudes into impossible regions; however, these regions are never utilized by the optimizer while transforming the thrust vector region into a convex set. Also, there exist limitations with regard to the thrust pointing. For our rocket, we have a maximum gimbal angle of 15°; thus, we implement its equivalent constraint mathematically:

$$\hat{n}^T T_c(t) \geq |T_c(t)| \cos(\phi) \tag{5}$$

Where $\phi = 15°$. Additionally, we have terminal constraints for our states $r(t_f) = 0$ and $v(t_f) = 0$, which describes a soft landing beginning at the origin. Finally, we introduce a toggleable glide cone constraint $|Sr(t)| \leq c^T r(t)$, where $c = [-\tan\theta_{alt} \quad 0 \quad 0]^T$. When enabled, the rocket avoids surrounding terrain and edge case trajectories that pass through the ground.

IV. Approaches

One key constraint is the time of flight t_f constraint. A method of finding a minimum and maximum bound on possible time of flights is laid out in [6]. As mentioned in our problem statement, a time-optimal solution is similar to a fuel-optimal solution. Therefore, a simple line search can be performed on time of flights starting from the minimum bound. The problem can be attempted to be solved with various times of flights, and the first solution to converge can be used. However, increasing the resolution of the problem also means that each attempt to solve with various times of flights also takes longer, making the line search more inefficient. This can be remedied by using a dual-stage approach to finding the time of flight, using a coarse resolution version to find a time of flight that works and then solving at that time of flight with a fine resolution version. This enables the line search to run faster while enabling a high resolution final solution. This approach remains consistent between our two proposed solutions.

A. Time-Based Discretization

The first numerical analysis approach utilized for this powered descent guidance problem was time-based discretization. Numerical discretization is a translation of continuous functions into a discrete number of points to approximate the values of the original functions. Instead of solving the original SOCP, we can solve the finite-discretized version for our optimal landing. This approach allows one to impose constraints at specific nodes, which can be uniformly distributed, allowing for a more predictable optimization structure. Additionally, as long as the control set is convex, the interpolated values in the ZOH time-based discretization approach remain inside the convex set, preventing internode constraint violations. This method offers better scalability than pseudospectral methods due to its local coupling, where constraints depend only upon their neighbors, yielding sparse constraint Karush-Kuhn-Tucker (KKT) system matrices [7]. The discretization approach's accuracy depends on a hyperparameter that defines the number of nodes to represent the original continuous function. Being an approximation, discretization introduces errors which can accumulate over longer trajectories. This results in a significant tradeoff between accuracy and computational cost; in order to increase the accuracy and reduce the internode violations of the computed trajectory, it is required to introduce additional nodes and, thus, sacrifice more compute. Finally, the uniform mesh predominantly utilized with this method is fairly inefficient and often ineffective for accounting for sections of the trajectory with fast-changing dynamics.

In our time-based discretization approach, the infinite-dimensional problem is discretized by assuming that the control inputs remain constant over fixed time intervals. We represent this discretized time interval with nodes $t_0, ..., t_f$. t_f is given by the line search as defined previously. The constraints at each time step are defined based upon variables from the current time step and the previous one, creating a sparse constraints matrix. Most constraints are very similar to their continuous time versions, with the only notable difference being that the physics are determined by summation across discrete time steps rather than continuously integrated. For instance, instead of $\dot{r}(t) = r(t) + g + \frac{T_c(t)}{m(t)}$, the velocity constraint is expressed as $\dot{r}(t_{k+1}) = v(t_k) + g * \delta t + \frac{T_c(t_k)}{m(t_k)} * \delta t$ for some $k \in (0, t_f)$. A more detailed elaboration of the discretization can be found in [4]. The primary difference with our discretization is that our minimum thrust constraint is approximated with a linear Taylor expansion rather than a quadratic expansion. This was done because our solver does not allow for the quadratic constraint without using methods such as successive convexification to transform the constraints into conic constraints, which detracts from the focus on the lossless convexification problem solving only once.

B. Chebyshev Polynomial Paramterization

The second numerical method we assessed was pseudospectral parameterization with Chebyshev polynomials on Chebyshev-Gauss-Lobatto (CGL) nodes. Parameterization focuses on representing a continuous function using more computationally efficient functions such as constant, linear, or other polynomial functions. Here, Chebyshev polynomials are used as approximators while their defined set of CGL nodes are the basis for enforcing constraints. This method of parameterization avoids the Runge phenomenon, which is when the parameterization wildly oscillates near endpoints with equally spaced nodes [8] which tends to be the case for time-based discretization. Time of flight will also be determined through the line search method described before. The inequality constraints remain largely similar too, with the only exception being the calculation of the current time in z_0 for the minimum and maximum thrust bounds. However, because CGL nodes are globally coupled, changes to one node reflect changes in the interpolation across all nodes, so the dynamics constraints require the usage of a global differentiation matrix [7]. The position constraint becomes $D\dot{r}(\tau) = \frac{t_f}{2}\dot{r}(\tau)$, the velocity constraint becomes $D\dot{v}(\tau) = \frac{t_f}{2}g + \frac{t_f}{2}u(\tau)$, and the mass constraint becomes $D\dot{w}(\tau) = -\frac{t_f}{2}\dot{\alpha}\dot{\sigma}(\tau)$. D couples the constraints across all nodes, while the $\frac{t_f}{2}$ is necessary for the transformation from the time domain of $\tau \in [-1, 1]$ for the Chebyshev polynomial to the time domain of $t \in [0, t_f]$. The constraint matrix is much smaller but denser than the ZOH matrix, leading to longer solve times per iteration and potential numerical instability with floating point numbers. However, because the dynamics constraints are globally coupled, the Chebyshev parameterization method requires fewer iterations than ZOH to converge to a solution. Additionally, Chebyshev parameterization exhibits spectral, or exponential, convergence. This means that the accuracy scales with the number of nodes at an exponential rate, rather than linearly scaling like the ZOH method.

V. Methodology

For comparing our methods, we prepared four flight profiles. The first two are a 50 m direct descent with uncapped velocity, as well as a 50 m descent with an offset of 10 m in the x-axis and 20 m in the y-axis. The next two are created through repeating these profiles with velocity capped at 5 m/s. The four flight profiles together cover typical operating conditions for our lander vehicle, providing enough dynamic range to make an informed determination of which algorithm is better suited for real-time usage. The vehicle's operating conditions are derived based on the requirements of a 50 m hop described by the Collegiate Propulsive Landers Challenge[9].

To make accurate and meaningful comparisons between the ZOH method and Chebyshev parameterization method, there must be a common "ground truth" against which both methods are compared. It is known that an extremely high resolution discretization using the zero-order hold method will converge towards the ground truth of a piecewise solution to the control function[10]. Therefore, we have chosen to represent our ground truth using a ZOH method using a final pass with resolution delta time of 0.0005 s. Every flight profile is run once using these settings to create a baseline for comparison. Additionally, several constraints are held constant throughout our ZOH and Chebyshev parameterization runs. The minimum thrust is set at and the maximum thrust is 1000 N. The dry mass is 50 kg, while the fuel mass is set at 30 kg. The α for the $\dot{m}$ to thrust relation is $\frac{1}{9.81*180}$. The gimbal range is set at 15°. The glide slope is 5°. These constraints are set to mirror those of the 1000 N lander vehicle being built at Propulsive Landers at Georgia Tech.

To provide a fair comparison of both methods, we set up four resolutions: very coarse, coarse, fine, and very fine. This allows us to highlight how both methods perform in different use cases, ranging from real-time planning to high-fidelity simulation. Each increase in resolution corresponds to an increase in the number of nodes in the Chebyshev parameterization and a decrease in the time between discretized states in the ZOH discretization. We hypothesize that an increase in resolution corresponds to an increase in solve time and an improvement in accuracy when compared to the ground truth mentioned earlier. We adjust the resolution of both methods such that the solve times for the ZOH discretization and Chebyshev parameterization methods are similar for one pass of the solver, tolerating at most a 10% discrepancy. Note that this solve time does not include the time needed to conduct a line search for the lowest time of flight that ensures the solver converges to a valid and feasible trajectory.

We run each flight profile across each resolution for both the ZOH discretization and Chebyshev parameterization. We then perform a root mean square error (RMSE) analysis on each of these paths to their ground truths. This RMSE analysis is run on all decision variables, which includes position, velocity, and the thrust vector, and properly accounts for multi-dimensional variables by using the euclidean distance between states. We select the RMSE of position and thrust vector to determine how well the solution of a given run converged to the ground truth.

Table 1 Run Configuration Summary

Run	Time of Flight (s)	ZOH dt (s)	CGL Nodes
Direct Descent | Very Coarse	7.500	0.4000	5
Direct Descent | Coarse	7.100	0.2000	10
Direct Descent | Fine	7.025	0.0250	30
Direct Descent | Very Fine	7.013	0.0125	46
Direct Limited Descent | Very Coarse	11.700	0.7800	6
Direct Limited Descent | Coarse	11.400	0.3000	10
Direct Limited Descent | Fine	11.100	0.0500	34
Direct Limited Descent | Very Fine	11.100	0.0125	60
Offset Descent | Very Coarse	7.500	0.5000	5
Offset Descent | Coarse	7.300	0.2000	10
Offset Descent | Fine	7.230	0.0300	31
Offset Descent | Very Fine	7.229	0.0062	68
Offset Limited Descent | Very Coarse	12.600	1.0500	6
Offset Limited Descent | Coarse	12.000	0.4000	10
Offset Limited Descent | Fine	12.000	0.0500	34
Offset Limited Descent | Very Fine	11.975	0.0125	63

Table 1 details runs, which consists of a flight plan and a resolution. For each run, we provide the (minimum) time of flight for which the run converged, the delta time used by the ZOH solver to discretize, and the number of nodes used for interpolation by the Chebyshev solver.

VI. Results

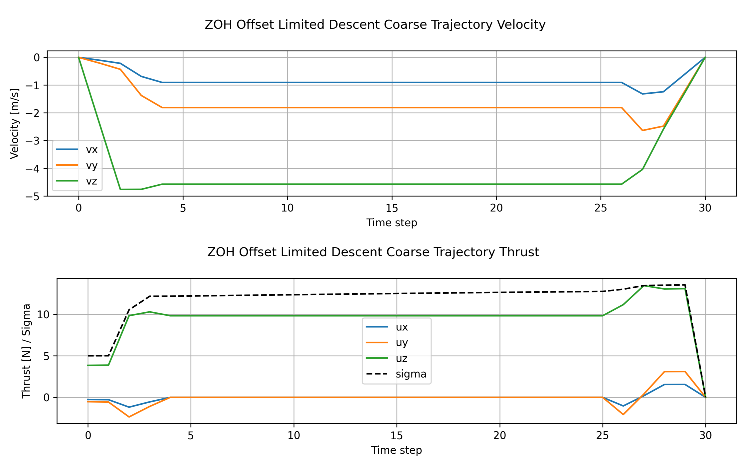

Fig. 1 Generated trajectory velocity and thrust for the offset limited descent.

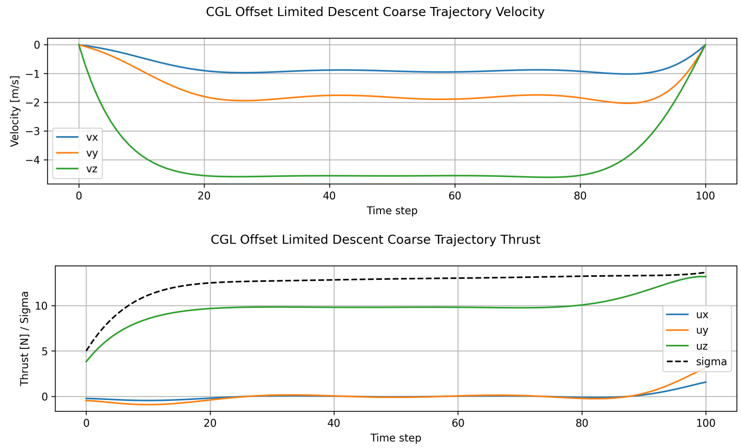

Fig. 2 Generated trajectory velocity and thrust for the offset limited descent.

Figure 1 and 2 show velocity and thrust / sigma over time for the velocity limited offset descent flight plan at coarse resolution, as calculated by both the Zero Order Hold (ZOH) and Chebyshev (CGL) implementations. We understand the ZOH solution to converge to a piecewise, discrete solution for thrust, while the Chebyshev implementation, by nature of polynomials, produces a smoother solution, at least at relatively low node counts[7]. Our graphs further support this.

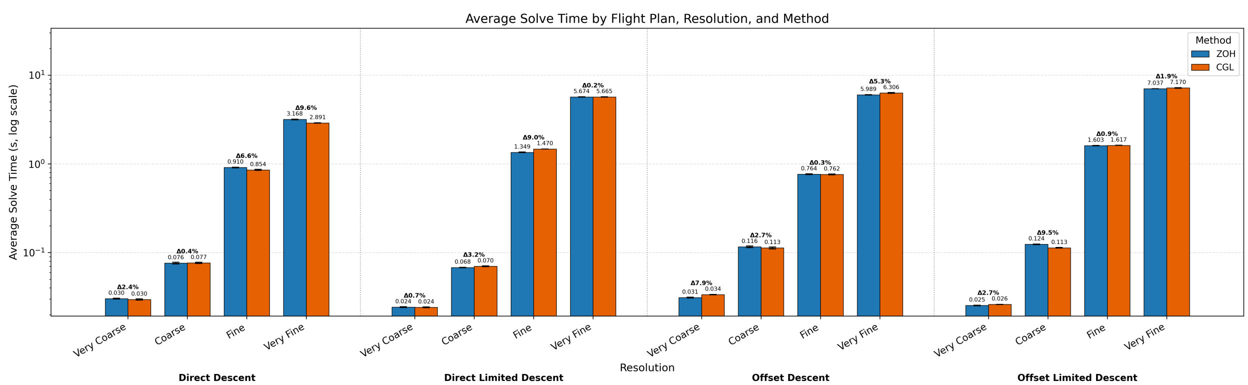

Fig. 3 The average solve time for our runs.

Figure 3 shows the mean solve time for each run across both solvers. Our methodology sets the resolutions such that the average solve time does not differ by more than 10% in any given pair of runs.

The RMSE of the rocket's position and thrust in Fig.4 and Fig.5 are calculated by comparing 100 equally spaced samples from the trajectories produced by each method and the ground truth.

Here are the general trends we noticed while reviewing the results. The trajectories computed by the ZOH solver seem to converge to the ground truth with significantly lower position RMSE compared to the CGL trajectories. In contrast, the position RMSE of the CGL-produced trajectories seem to flatline, even with a greatly increased number of nodes. The CGL implementation theoretically models the exact same problem as the ZOH solver, and should converge to the same solution, but it appears this will take a number of nodes infeasible for our use cases. To explain it's flatline behavior, we believe the CGL converges to a CGL ground truth instead of a ZOH, but this ground truth would essentially be the same problem as the ZOH produced ground truth.

The CGL implementation does appear to be more consistent overall, performing better than ZOH in terms of position RMSE at very coarse resolutions when compared to the ground truth. This suggests that the polynomial approximation captures the fundamental dynamics of the "spectral ground truth" more effectively with fewer parameters, whereas the ZOH method requires higher resolution to overcome its piecewise limitations. Although it fails to converge as close to the ground truth at higher resolutions, when compared to the ZOH implementation, it provides overall smoother controls, which would be advantageous and more realistic for rocket control. This is backed by Fig.6 where ZOH

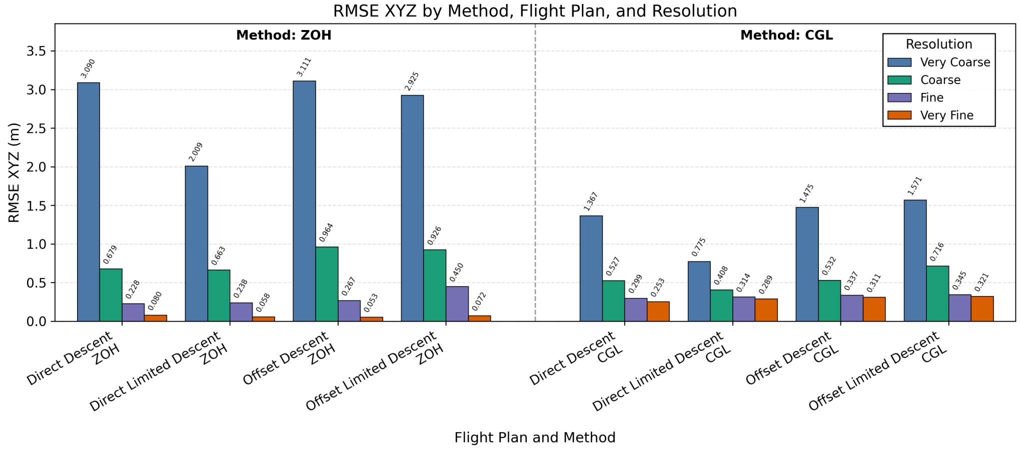

Fig. 4 RMSE for position.

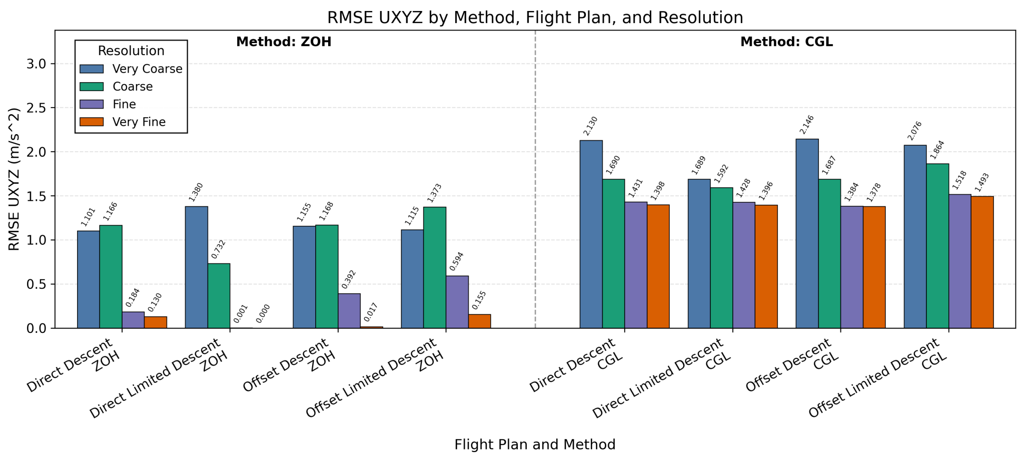

Fig. 5 RMSE for thrust vector.

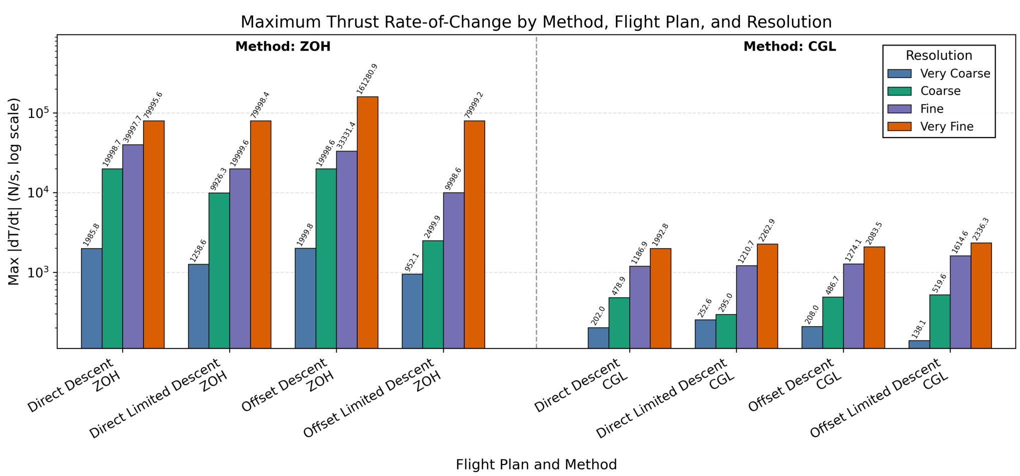

Fig. 6 Maximum thrust rate of change.

changes thrust abruptly, which is a consequence of the discretization approach since it maintains a constant-piecewise control. The Chebyshev interpolation method, on the other hand, ensures that the maximum thrust rate-of-change remains orders of magnitude lower, depending on the resolution, providing solutions that are physically plausible and representative of real thruster systems. We do nevertheless notice an increasing thrust rate of change with the CGL method corresponding with an increase in resolution. This can be attributed to the larger number of nodes given to CGL to represent sharper features for our trajectory.

VII. Conclusion

This paper discusses the implementation of the Lossless Convexification (LCvx) of optimal powered descent problem using time-based discretization and Chebyshev parameterization. First, the strengths and differences between the two approaches were described. Then, analyzing the trajectories and RMSE of the position and thrust, we assessed the effectiveness of each method. Finally, the experiments concluded that Chebyshev parameterization was the better approach for real-time computation of the powered descent guidance algorithm due to its better accuracy at coarser resolutions and more achievable controls. Moving forward, the choice of which numerical method to use to solve the LCvx of the soft landing problem should depend on flight computer constraints, but, for GTPL's use case, the Chebyshev parameterization using CGL nodes will be their primary approach for path planning on our hybrid rocket engine lander.

In the future, for an updated paper, it would be important to further investigate the ground truth claims for the ZOH and CGL. This adapted study would generate a ground truth path independent of ZOH and CGL methods and test the two approaches against the ground truth for a better representation of their performances. Alternatively, we could still generate a CGL ground truth path through similar means to the ZOH ground truth path and compare the approaches against their respective ground truths. However, this would make comparing the two approaches more difficult, and raise questions as to which path is "optimal."

Additionally, we could run our algorithms onboard a flight computer to better test their ability to function in real-time embedded systems. An analysis of the time complexity of each version of the algorithm may also serve as a foundation for an effective comparison of the performance of the two algorithms. We plan on analyzing the possibility of representing the 3-DoF soft landing problem as a 5-DoF problem, accounting for angular dynamics. By proving that the guarantees of optimality persist using LCvx on this greater problem, we can further improve upon the findings of this paper.

Acknowledgments

The authors would like to acknowledge the Georgia Institute of Technology for providing the institutional support, facilities, and academic environment that make student-led aerospace research possible. This work represents the collective effort of Propulsive Landers at Georgia Tech (GTPL). The authors sincerely thank the members of GTPL across operations, propulsion, avionics, structures, and guidance, navigation, and control for their contributions, long test days, and continued commitment to advancing reusable rocketry at the student scale.

References

[1] Karur, K., Sharma, N., Dharmatti, C., and Siegel, J. E., "A Survey of Path Planning Algorithms for Mobile Robots," *Vehicles*, Vol. 3, No. 3, 2021, pp. 448–468. https://doi.org/10.3390/vehicles3030027, URL https://www.mdpi.com/2624-8921/3/3/27.

[2] Kim, N., Erdenebat, S., and Law, K., *Spline-Based Flight Path Planning and Following for Aerial Navigation*, American Institute of Aeronautics and Astronautics, 2025. https://doi.org/10.2514/6.2025-99477, URL https://arc.aiaa.org/doi/abs/10.2514/6.2025-99477.

[3] Mao, Y., Szmuk, M., Xu, X., and Acikmese, B., "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems," , 2019. URL https://arxiv.org/abs/1804.06539.

[4] Acikmese, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. https://doi.org/10.2514/1.27553, URL https://doi.org/10.2514/1.27553.

[5] Meditch, J., "On the problem of optimal thrust programming for a lunar soft landing," *IEEE Transactions on Automatic Control*, Vol. 9, No. 4, 1964, pp. 477–484. https://doi.org/10.1109/tac.1964.1105758, URL http://dx.doi.org/10.1109/tac.1964.1105758.

[6] Açıkmeşe, B., Carson, J. M., and Blackmore, L., "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, 2013, pp. 2104–2113. https://doi.org/10.1109/TCST.2012.2237346.

[7] Szmuk, M., Reynolds, T. P., and Açıkmeşe, B., "Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, p. 1399–1413. https://doi.org/10.2514/1.g004549, URL http://dx.doi.org/10.2514/1.G004549.

[8] Trefethen, L., *Spectral Methods in MATLAB*, Software, environments, tools, Society for Industrial and Applied Mathematics, 2000. URL https://books.google.com/books?id=l1YfkwEfQtQC.

[9] The Definity Project, "Lander Challenge: Challenging ambitious young people to build self-landing rockets," Competition Website, 2023. URL https://landerchallenge.space/.

[10] Hager, W. W., "Runge-Kutta methods in optimal control and the transformed adjoint system," *Numerische Mathematik*, Vol. 87, No. 2, 2000, pp. 247–282. https://doi.org/10.1007/s002110000178, URL https://doi.org/10.1007/s002110000178.