

Fuel-Optimal Spacecraft Guidance for Landing in Planetary Pits

Neal S. Bhasin
neal@cmu.edu

CMU-RI-TR-16-10

*Submitted in partial fulfillment of the
requirements for the degree of
Masters of Science in Robotics*

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Masters Committee:
William “Red” Whittaker, Chair
Christopher G. Atkeson
Christopher Cunningham

April, 2016

Abstract

Propulsive spacecraft enable scientific discovery and exploration of the worlds beyond Earth. Autonomous spacecraft have landed on Earth, the Moon, Mars, Mercury, Venus, Titan, asteroids, and a comet. Recently discovered planetary pits allow access to subsurface voids valuable for scientific discovery and sustained exploration. With recent advancements in embedded convex optimization software and trajectory optimization theory, increasingly sophisticated autonomous missions will be able to safely and efficiently reach these unexplored destinations.

This thesis develops and tests an algorithm for fuel-optimal landing into planetary pits. By representing the safe regions outside and inside a planetary pit as distinct convex spaces, techniques for optimal guidance based on convex optimization are extended to find trajectories into pits. A search routine for time of flight and time of entry into the pit finds globally fuel-optimal landing trajectories. This time search softens constraints on maximum thrust and landed vehicle mass to reliably find solutions without sensitivity to initialization. The algorithm is implemented within a modeling language and uses an embedded solver for convex optimization. The resulting implementation is therefore practical and effective for use in future missions.

The algorithm is tested in simulated landing scenarios that vary vehicle parameters, mission constraints, and pit dimensions. The feasibility and optimality of generated trajectory solutions are examined along with algorithm runtime. This research determines that fuel-optimal guidance capable of landing within planetary pits is viable for future missions.

Contents

1	Background	3
1.1	Planetary Pits and Caves	3
1.2	Canyons	4
1.3	Propulsive Spacecraft Landing	4
2	Research Goal	4
3	Related Work	5
3.1	Polynomial Guidance	5
3.2	Pinpoint Landing and Optimal Guidance	6
4	Method	7
4.1	G-FOLD	7
4.1.1	Time of Flight Search	9
4.2	Pit Landing Guidance	10
4.2.1	Two-Dimensional Time Search	11
4.3	QCML	13
4.4	ECOS	14
5	Results	15
5.1	Scenario 1: Planar Divert	15
5.2	Scenario 2: Descent Glide Slope	17
5.3	Scenario 3: 3D Braking	19
5.4	Scenario 4: Varied Parameters	21
5.5	Runtime	22
6	Conclusion	24
7	Future Work	24
References		25

1 Background

1.1 Planetary Pits and Caves

Subsurface caves on the Moon and Mars are locations of high scientific interest and may ultimately become destinations for safe human habitation. While their existence has long been hypothesized, they have never been directly observed since they cannot be seen from orbit. Pits were recently discovered on the Moon (2009) [16] and Mars (2007) [8]. Some pits enable access to subsurface voids and are likely to lead into lava tube caves.

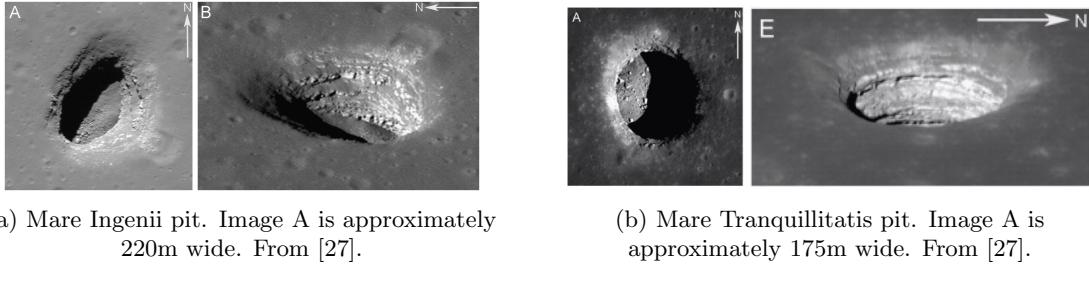


Figure 1: Lunar pits imaged by LRO under varying camera and sunlight angles.

The floors of pits and their corresponding subsurface voids offer several exploration benefits and scientific targets for future missions and habitats. A survey of lunar pits by Haruyama et. al [17] indicates protection from radiation damage from galactic cosmic rays and solar proton events, protection from meteorite impacts, and milder temperatures without the severe fluctuations experienced on the surface. The regolith layers exposed in the inner walls of pits could provide information about the history of the lunar magnetic field, volcanic activity, and solar activity. Although the lunar surface is covered with regolith [18], there is no wind to carry dust into a lava tube. The interiors of lava tubes would likely be pristine, dust-free environments. Observations and samples from within a lava tube would provide significant information about cave morphologies and the environment at the time of lava tube formation.

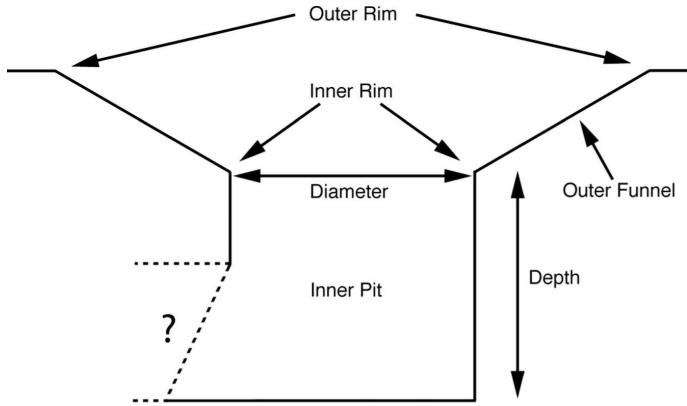
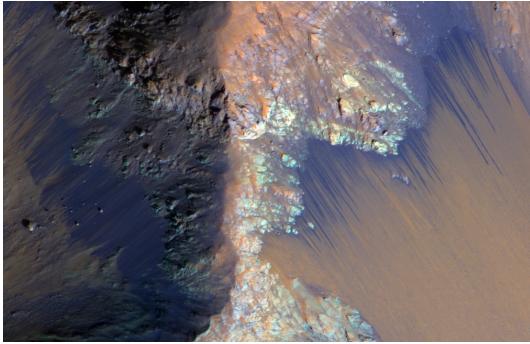


Figure 2: Diagram of planetary pit and potential subsurface cave. From [33].

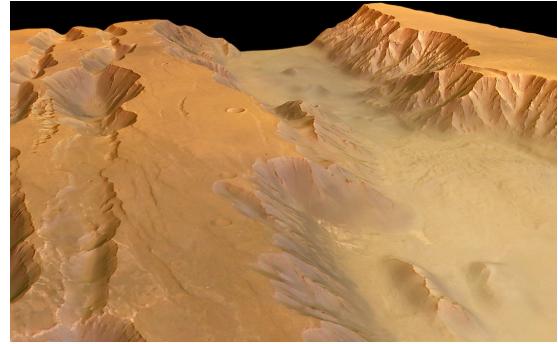
Mars has little magnetic field to protect from radiation and very low atmospheric density, admitting a high flux of meteorites. Subsurface voids on Mars therefore provide the same exploration benefits as lunar voids [8]. They are also a compelling location to search for past or current life, and may be the only place on Mars with conditions necessary for life [6].

1.2 Canyons

Recurring Slope Lineae (RSL) [22] are recently discovered active Martian surface features that are believed to be the result of seasonal flows of liquid water [23]. These are high-priority destinations for scientific investigation and could one day be critical for human settlement. These dark and narrow features occur on some steep and warm slopes. Landing at the bottom of a canyon like Coprates Chasma (Figure 3) would require similar landing capability to landing in planetary pits. From there, surface rovers could observe the canyon walls from close proximity.



(a) Enhanced-color nadir view of RSL on the slopes of Coprates Chasma. Credit: NASA/JPL/University of Arizona



(b) Perspective view of Coprates Chasma. Copyright: ESA/DLR/FU Berlin (G. Neukum)

Figure 3: Recurring Slope Lineae (RSL) in Coprates Chasma likely indicate liquid water on Mars.

1.3 Propulsive Spacecraft Landing

Propulsive spacecraft landing is a particularly challenging problem in robotic path planning. Landing spacecraft burn significant amounts of propellant in a short time in order to reduce velocity for a soft touchdown. The spacecraft mass properties and dynamics therefore change over time, as a function of control input. Throttled spacecraft thrusters typically have upper and lower thrust limits after ignition [5].

Computing power is limited on spacecraft in high-radiation environments, requiring the use of specialized lightweight and deterministic software typical in embedded systems. Communication with Earth is not always possible and has high latency when available. Robotic landing guidance must therefore occur fully autonomously and reliably produce trajectories in real-time.

Launch costs are extremely high for interplanetary missions. Every kilogram of fuel saved during landing can reduce mission costs or increase the amount of landed payload. More fuel-efficient guidance algorithms increase the maneuver capability of existing vehicles to expand the space of reachable landing destinations or improve the ability to correct dispersions from earlier mission stages.

2 Research Goal

This research seeks to develop a practical and effective algorithm for fuel-optimal spacecraft landing guidance within planetary pits.

For practical use across multiple missions and landing scenarios, the algorithm implementation must be accessible to non-expert developers. An effective guidance algorithm must quickly generate feasible trajectories that precisely target the desired landing state. A fuel-optimal algorithm must find the trajectory that satisfies all constraints while using the globally minimal amount of fuel.

3 Related Work

3.1 Polynomial Guidance

The Apollo lunar landing program developed original techniques for autonomous targeted spacecraft landing guidance [15]. Computation was extremely limited and the algorithms were tailored to allow for retargeting and manual flight by astronauts. In the targeted approach phase of powered descent, Apollo guidance (program P64) used boundary state constraints and a fixed time-to-go to find closed-form solutions for acceleration profiles [25]. The vertical acceleration is a linear function of time, while the lateral accelerations are quadratic polynomials. The commanded position over time is therefore a quartic polynomial of time [20].

The trajectories generated by Apollo guidance are not guaranteed to satisfy vehicle thrust constraints and the calculated guidance commands can therefore introduce dispersions which must be corrected by later trajectory recalculations [15] [31]. Because there are no constraints on the vehicle state (except at the trajectory boundaries), generated trajectories may go underground or collide with obstacles. These trajectories are also not optimal with respect to time or fuel consumption [25].

The Apollo landings were incredibly successful, resulting in the first and only manned lunar landings to date. The six attempted landings of Apollo 11, 12, 14, 15, 16, and 17 between 1969 and 1972 all safely touched down on the lunar surface through the combination of astronaut control and the polynomial guidance routine. With the exception of Apollo 11, all landings were within 1km of the initial targeted landing site [24].

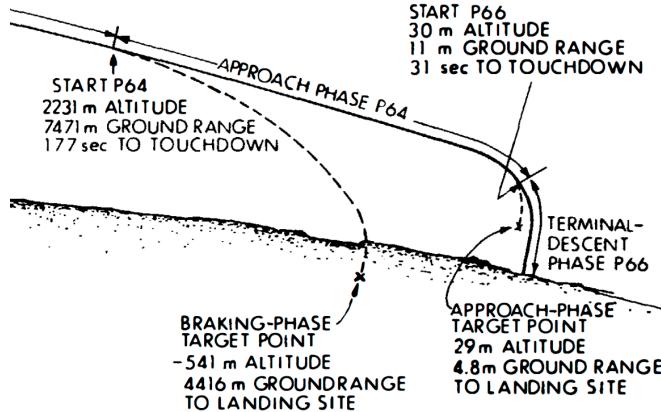


Figure 4: Diagram of final Apollo powered descent phases. From [20].

Since the end of the Apollo program, the approach guidance method has been extended for increased performance [15] [25]. The modified Apollo guidance routine uses quadratic acceleration profiles in each axis, and allows the time-to-go to be a free parameter. A line search for the time-to-go is used to find trajectories that respect vehicle constraints, minimize landing time, or minimize fuel consumption. This approach can greatly improve performance without significantly increasing implementation complexity, but does not result in globally optimal trajectories [25].

The Mars Science Laboratory mission used an Apollo-based polynomial guidance approach for powered descent during its 2012 landing [26]. As the fully autonomous system did not have precision navigation [34] or hazard detection capabilities, the guidance method did not divert to target a landing site chosen online. The algorithm targeted a fixed altitude and downrange distance relative to the backshell and parachute, which are separated at the start of powered descent. The polynomial trajectory aimed to zero the horizontal velocity and bring the vertical velocity to a predetermined level prior to a constant velocity vertical descent [26].

3.2 Pinpoint Landing and Optimal Guidance

Landing missions to date have typically targeted benign surface regions to improve the probability of safe landing. Multiple technologies for autonomous precision navigation and hazard detection have been recently developed and tested, including the NASA Autonomous precision Landing Hazard Avoidance Technology (ALHAT) [28], Astrobotic Autolanding System (AAS) by Astrobotic Technology Inc. [10], and NASA JPL Lander Vision System (LVS) [19]. The development of these systems and precision landing guidance will enable autonomous pinpoint landing [34], capable of landing within 100m of a target. This capability enables missions to regions of higher scientific interest and locations within hazardous regions.

The Guidance algorithm for Fuel Optimal Large Diverts (G-FOLD) algorithm [1] was developed, starting in 2004, to achieve optimal powered descent guidance performance [2]. The algorithm explicitly minimizes fuel consumption to fully utilize spacecraft divert capability, and achieve maximum landing precision. It generates feasible trajectories by incorporating thrust magnitude and pointing constraints, with state constraints on maximum velocity and an approach glide slope for surface avoidance [1].

G-FOLD operates by solving optimization problems to search for the time of flight, thrust profile, and translational vehicle trajectory that minimize fuel consumption while satisfying state and control constraints. By augmenting the optimization problem for a certain time of flight with an additional slack variable and using an equivalent set of control constraints, the landing optimization problem is convex and can be solved to global optimality with Interior Point Method algorithms [5] [4]. With recent techniques for automated custom code generation of interior point method solvers, an optimization routine specifically tailored to the landing guidance problem is able to generate trajectories in real-time [12].

G-FOLD was successfully tested on the terrestrial Masten Space Systems Xombie vehicle in 2012 [2]. Three lateral divert maneuvers of 550, 650, and 750m were generated offline before the flight test and each was successfully executed within 1m of the targeted position and 1.25m/s of the velocity throughout the entire flight. Further tests of G-FOLD in 2013 [29] generated divert maneuvers with onboard software in flight. Two successful diverts of 570m and 800m were flown within 3m of targeted position and 1m/s of velocity.

4 Method

4.1 G-FOLD

The G-FOLD (Guidance for Fuel-Optimal Large Diverts) [3] [4] [5] algorithm solves the fuel-optimal and minimum-landing-error problems by using an equivalent convex relaxation of the non-convex spacecraft control constraints. The fuel-optimal landing problem can be directly described as:

t_f	Total time of flight
$T(t)$	Total commanded thrust as a function of time
$r(t)$	Spacecraft position
$m(t)$	Spacecraft mass
m_{wet}	Initial spacecraft mass with fuel
m_{dry}	Spacecraft structure mass
I_{sp}	Specific impulse
g_0	Standard Earth gravity
ϕ	Cant angle of thrust
g	Landing body gravity
ρ_1, ρ_2	Spacecraft lower and upper thrust limits

$$\begin{aligned}
& \min_{t_f, T(t)} \quad \int_0^{t_f} \|T(t)\|_2 dt \\
\text{subject to} \quad & \ddot{r}(t) = g + \frac{T(t)}{m(t)} \quad && \text{Dynamic Constraints} \\
& \dot{m}(t) = -\frac{1}{I_{sp}g_0 \cos \phi} \|T(t)\|_2 \\
& 0 < \rho_1 \leq \|T(t)\|_2 \leq \rho_2 \quad && \text{Thrust Limits} \\
& m(0) = m_{wet} \quad m(t_f) \geq m_{dry} \quad && \text{Boundary Conditions} \\
& r(0) = r_0 \quad r(t_f) = r_f \\
& \dot{r}(0) = \dot{r}_0 \quad \dot{r}(t_f) = \dot{r}_f
\end{aligned}$$

The goal of this optimization problem is to minimize fuel consumption while finding thrust inputs that take the spacecraft from an initial state to the target state, subject to constraints on thrust and vehicle dynamics. The thrust $T(t) \in \mathbb{R}^3$ applied by a spacecraft at any time t has some upper and lower magnitude limits. The lower limit ρ_1 is non-zero since the engines typically cannot be throttled off during landing [4]. The thrust constraints:

$$0 < \rho_1 \leq \|T(t)\|_2 \leq \rho_2$$

represent a non-convex set due to the lower bound on the l^2 -norm of $T(t)$. The authors of G-FOLD introduce a scalar slack variable $\Gamma(t) \in \mathbb{R}$ and relax the thrust constraints (shown in Fig. 5) as:

$$\|T(t)\|_2 \leq \Gamma(t) \quad 0 < \rho_1 \leq \Gamma(t) \leq \rho_2 \quad \dot{m}(t) = -\frac{1}{I_{sp}g_0} \Gamma(t)$$

Mass consumption is a function of the upper bound slack variable Γ , but the dynamics are still constrained by the thrust vector T . It is therefore wasteful for a trajectory solution to not have the constraint $\|T(t)\|_2 \leq \Gamma(t)$ active.



Figure 5: Non-convex thrust magnitude constraints and convex relaxed thrust control. From [4].

Using the maximum principle of optimal control theory, the constraint $\|T(t)\|_2 \leq \Gamma(t)$ and one of the constraints $\rho_1 \leq \Gamma(t) \leq \rho_2$ are shown in [5] to be active at any point in time within an optimal solution to the fuel-optimal landing problem. The optimal solution to the relaxed problem is therefore always a feasible and optimal solution to the original non-convex problem.

The algorithm discretizes the problem and makes a change of variables to minimize the logarithm of fuel usage, with $\sigma = \frac{\Gamma}{m}$, $u = \frac{T}{m}$. The time of flight t_f is fixed, and found using a separate outer-loop optimization described below. This allows the problem to be expressed as a Second Order Cone Program (SOCP). Letting $\alpha = \frac{1}{I_{sp}g_0 \cos \phi}$, the resulting convex optimization problem (adapted from Problem 4 of [3]) is given by:

$$\begin{aligned}
& \min_{u_i, \sigma_i} && -z_N \\
\text{subject to} & && r_{k+1} = r_k + \frac{\Delta t}{2}(v_k + v_{k+1}) - \frac{\Delta t^2}{12}(u_{k+1} - u_k) \\
& && v_{k+1} = v_k + \frac{\Delta t}{2}(u_k + u_{k+1}) + g\Delta t \\
& && z_{k+1} = z_k - \frac{\alpha \Delta t}{2}(\sigma_k + \sigma_{k+1}) \\
& && \|u_k\| \leq \sigma_k \\
& && z_{0,k} = \ln(m_{wet} - \alpha \rho_2 k \Delta t) \\
& && \mu_{1,k} = \rho_1 e^{-z_{0,k}} \quad \mu_{2,k} = \rho_2 e^{-z_{0,k}} \\
& && \mu_{1,k} \left[1 - (z_k - z_{0,k}) + \frac{(z_k - z_{0,k})^2}{2} \right] \leq \sigma_k \leq \mu_{2,k} \left[1 - (z_k - z_{0,k}) \right] \\
& && z_{0,k} \leq z_k \leq \ln(m_{wet} - \alpha \rho_1 k \Delta t) \\
& && z_0 = \ln m_{wet} \quad z_N \geq \ln m_{dry} \quad N \Delta t = t_f \\
& && r_0 = r_0 \quad r_f = r_f \\
& && v_0 = v_0 \quad v_f = v_f
\end{aligned}$$

A glide slope constraint maintains a minimum altitude as a function of distance from the landing site to avoid simple surface obstacles. For an angle θ between the glide slope and ground plane, and altitude r_z with lateral position $r_{x,y}$, this constraint can be represented as the cone:

$$r_z \geq \tan \theta \|r_{x,y}\|$$

The algorithm further incorporates constraints (not shown here) to limit the maximum spacecraft velocity and pointing angle of the thrust vector.

The final problem formulation has many non-trivial constraints and is best implemented with the use of a modeling language for convex optimization, as manually converting it into a canonical SOCP problem form is difficult and prone to mathematical errors that may be hard to identify.

4.1.1 Time of Flight Search

Conservative bounds on the time of flight t_f , are derived in [5] from the vehicle mass and acceleration limits.

$$\begin{aligned} t_{min} &\leq t_f \leq t_{max} \\ t_{min} &= \frac{m_{dry} \|v_0 - v_f\|}{\rho_2} \\ t_{max} &= \frac{m_{wet} - m_{dry}}{\alpha \rho_1} \end{aligned}$$

The authors of G-FOLD observe in simulations [3] that the fuel cost is a unimodal function of t_f . The optimal time of flight in the fuel-optimal landing problem can therefore be found with any method for bounded one-dimensional optimization, where each function evaluation requires formulating and solving one SOCP problem.

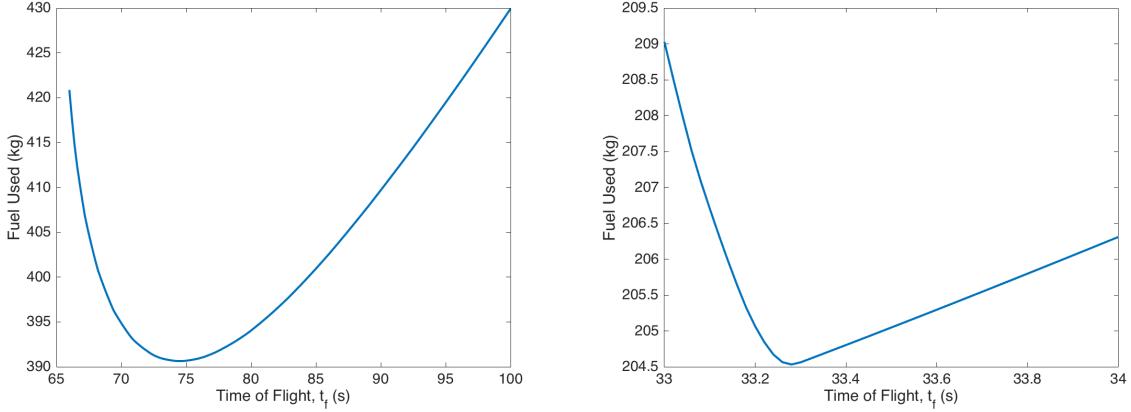


Figure 6: Optimization over time of flight for two significantly different landing scenarios.

4.2 Pit Landing Guidance

In the pit landing problem, the goal is to reach a target state inside a planetary pit with a fuel-optimal trajectory while avoiding the surface and the walls of the pit. The glide slope constraint implemented in G-FOLD is effective for avoiding ground obstacles when precisely landing on an approximately flat surface. However, it would not work for landing in most pits as the slope would have to be prohibitively large or the pit unrealistically shallow.

The safe region for the spacecraft approaching the pit when above the surface can be represented with a glide slope constraint, while the safe region inside the pit can be represented as a cylinder that fills the interior of the pit with some buffer for avoiding the walls (Fig. 7b). The complete safe position region for the problem is the union of these two regions (Fig. 7a) and is therefore non-convex. However, the glide slope cone and the pit cylinder regions are both individually convex. If the two regions are constructed such that they do not intersect, a landing spacecraft will only ever be within one region at any given time. This is accomplished by constraining the region outside of the pit to have positive altitude (above the pit inner rim), while the region inside the pit must have nonpositive altitude. The position constraints (at any given time) are therefore convex.

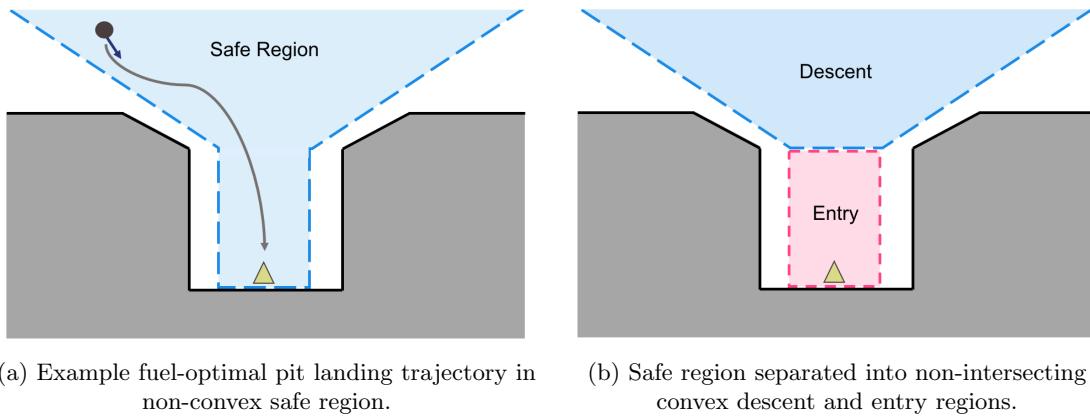


Figure 7: Safe position regions for the pit landing guidance problem.

The position constraints can effectively be enabled or disabled at different times with the use of indicator functions for the two regions. When the spacecraft is in descent mode, the indicator function for descent M_D is set to 1 and the indicator M_E for entry is set to 0. In entry mode, M_D is set to 0 and M_E is set to 1. Both sides of each position constraint equation are multiplied by the corresponding indicator. This disables a constraint by reducing it to $0 \geq 0$, which is always satisfied.

The pit landing position constraints can then be expressed as (note M_D and M_E are functions of time):

θ	Descent glide slope angle
M_D	Descent indicator
M_E	Entry indicator
p_r	Pit radius
p_b	Pit wall buffer
p_f	Pit floor altitude

$$\begin{aligned} M_D r_z &\geq M_D \tan \theta (||r_{x,y}|| - p_r + p_b) \\ M_D r_z &\geq 0 \end{aligned}$$

$$\begin{aligned} M_E ||r_{x,y}|| &\leq M_E (p_r - p_b) \\ M_E r_z &\geq M_E p_f \\ M_E r_z &\leq 0 \end{aligned}$$

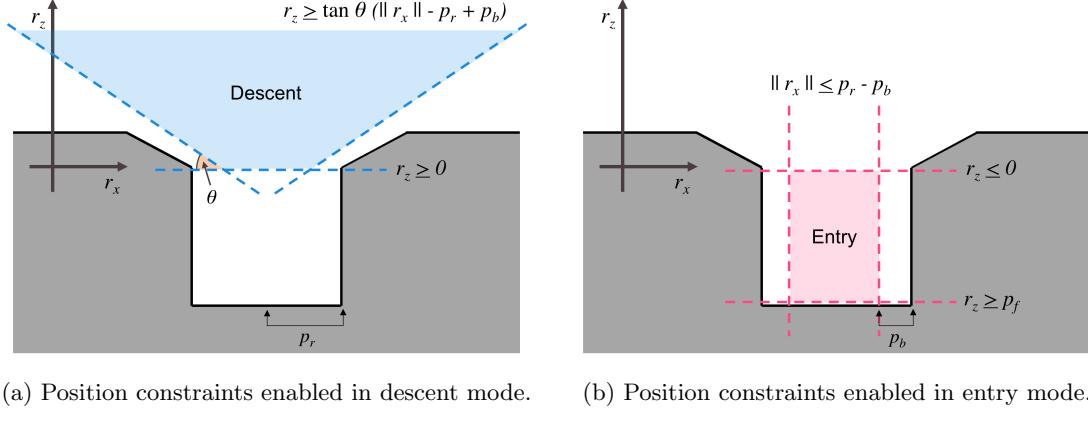


Figure 8: Position space constraints (in 2D) for the pit landing guidance problem.

The target state is within the pit, so the spacecraft will end in entry mode. With the additional constraint that the vertical velocity is negative while in entry mode, $M_E v_z \leq 0$, the spacecraft is guaranteed to stay within entry mode after a transition from descent mode. The values of the indicator functions M_D and M_E can then be computed given t_e , the unique time of entry into the pit.

$$(M_D, M_E) = \begin{cases} (1, 0) & \text{if } t < t_e \\ (0, 1) & \text{if } t \geq t_e \end{cases}$$

For fixed M_D and M_E , the position constraints for descent and entry modes are convex. The resulting problem is discretized in the same manner as in G-FOLD and solved as an SOCP.

4.2.1 Two-Dimensional Time Search

In the pit landing problem, both time of entry t_e and time of flight t_f must be computed in an outer optimization process.

Assuming the final target state has lower velocity than the initial state, the conservative upper and lower bounds on t_f from G-FOLD still hold. Simple bounds on t_e are given by:

$$0 \leq t_e \leq t_f$$

Tighter bounds on t_e and t_f can be derived in scenarios where there is also a maximum velocity constraint.

An exhaustive search for minimum fuel use over trajectories generated for all possible values of t_e and t_f is too computationally intensive for real-time guidance. The program must therefore employ a multidimensional numerical optimization method like the Nelder-Mead simplex method [21]. The two-dimensional search is complicated by the possibility of non-feasible iterates. If a value of t_e is chosen during initialization or optimization that is too close to 0 or t_f such that the trajectory is infeasible, the fuel use is undefined and it is unclear what values should be tested next.

By relaxing the upper thrust limit and dry mass constraints, each optimization problem converges to a solution trajectory for any value of t_e and t_f within simple upper and lower bounds. This is similar to the strategy employed by [32] to avoid artificial infeasibility introduced by linearization of aerodynamic drag constraints. The trajectories produced at each search iteration are not guaranteed to be dynamically feasible, but the landed mass and control thrust can be trivially tested for feasibility within the original problem constraints.

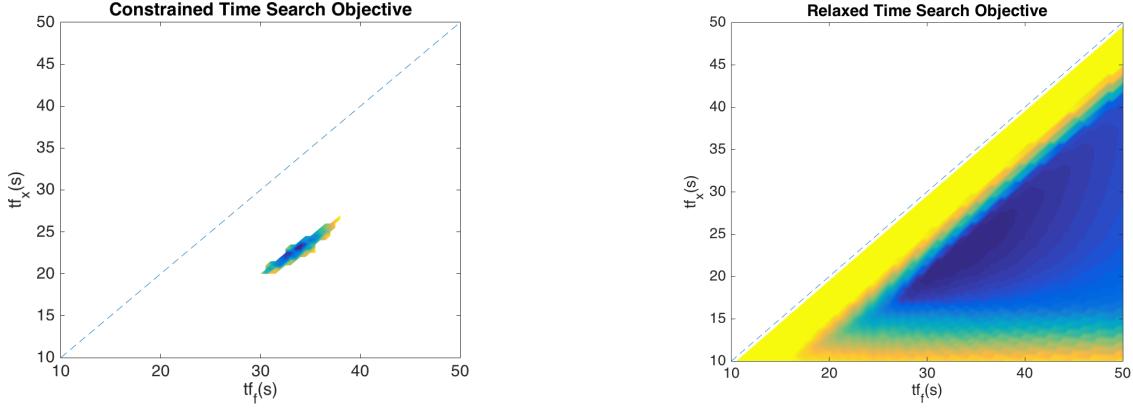
Weighted loss can be applied to the violation of thrust and dry mass constraints to transform them into soft constraints. Each trajectory optimization will then converge to a solution that balances the weighted constraint violation with the logarithm of fuel use. If there is a feasible solution to the original problem, $p_r = 0$ is feasible within the relaxed problem. For sufficiently large loss weight, the relaxed optimization problem will then effectively converge to the optimal solution of the original constrained problem.

$$\begin{aligned} w_r & \quad \text{Relaxation loss weight} \\ p_r & \quad \text{Acceleration relaxation term} \end{aligned}$$

$$\text{Modified objective function : } \min_{u_i, \sigma_i, p_r} -z_N + w_r p_r$$

$$\begin{aligned} \text{Relaxed acceleration constraints : } & \quad p_r \geq 0 \\ & \quad \|u_k\| \leq p_r + \sigma_k \end{aligned}$$

In the implementation of pit landing guidance, the discretized SOCP problem directly represents acceleration instead of thrust. The acceleration constraint is therefore relaxed as shown above, with weight chosen such that $w_r p_r \gg z_N$ for values of p_r above the minimum thrust control resolution. The acceleration from the relaxation term does not consume fuel. The dry mass constraint is removed and the final mass of the optimal trajectory is verified in order to transition to an abort mode if there is insufficient fuel.



(a) Objective function minimizes fuel use. Dry mass and thrust magnitude are constrained, creating a small region of feasibility.

(b) Objective function minimizes fuel consumption and weighted acceleration relaxation. No dry mass constraint and relaxed maximum thrust constraint.

Figure 9: Optimization over time of entry and time of flight.

4.3 QCML

QCML (Quadratic Cone Modeling Language) [7] is a library with a domain specific language for representing convex optimization problems, similar to CVX [14]. CVX is a popular modeling system implemented within Matlab for representing various classes of problems in the DCP (Disciplined Convex Programming) ruleset [13] and solving them using different optimizers. In contrast, QCML is a tool for developers aiming to deploy optimization programs. It parses SOCP problems that follow the DCP ruleset and generates front-end code to interface with the SOCP solver ECOS [11].

QCML automatically generates MATLAB, Python, or C code that takes an instance of an optimization problem described in the modeling language and transforms it into a canonical SOCP form. The transformed problem is then solved using a SOCP solver, and the solution is transformed back into the variables of the original representation. The MATLAB implementation is particularly useful for testing and verifying the problem formulation and the C implementation is most appropriate for a lightweight and rapid embedded flight software implementation. The generated front-end code is called a “canonicalizer” and initializes the canonical problem form variables. All code generated by QCML is deterministic and only performs memory allocation on initialization in order to create fixed size matrix variables. The software can therefore guarantee deterministic runtime and is appropriate for mission use.

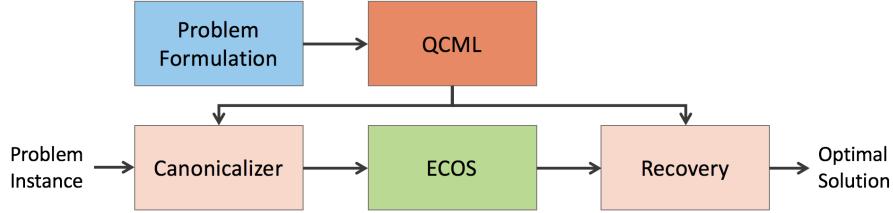


Figure 10: Block diagram of system architecture using QCML and ECOS.

As the trajectory optimization problem is parameterized and represented in QCML, it is possible to change algorithm parameters for use with different missions or vehicles without modifications to the source code. Developers can remove unnecessary constraints or add new constraints and objective function terms (provided the problem can still be represented within SOCP) without modifications to the solver. This is especially useful for scientists and engineers with problem-specific knowledge but minimal knowledge of convex analysis and numerical optimization. By avoiding modifications to the SOCP solver, QCML increases the amount of software that can be reused between different landing missions or on one mission requiring multiple optimization problems.

$$\begin{aligned}
&t \in \mathbb{R}^n \quad M \in \mathbb{R}^{3 \times n} \\
&f^* \in \mathbb{R}^3 \quad T_{max} \in \mathbb{R}^+ \\
&\min_t \sum_n t_i \\
&\text{subject to } Mt = f^* \\
&\quad 0 \leq t \leq T_{max}
\end{aligned}
\quad
\begin{aligned}
&\text{dimensions} && n \\
&\text{variable} && t(n) \\
&\text{parameter} && M(3, n) \\
&\text{parameter} && f_star(3) \\
&\text{parameter} && T_max \text{ positive} \\
&\text{minimize} && \text{sum}(t) \\
&\text{subject to} && \\
&&& M * t == f_star \\
&&& 0 \leq t \\
&&& t \leq T_max
\end{aligned}$$

Figure 11: An hour allocation optimization problem and a QCML implementation.

4.4 ECOS

ECOS (Embedded Conic Solver) [11] is an open-source SOCP solver for embedded computing platforms. It is single-threaded and written entirely in ANSI-C code. The only external library required is a modified version of the SparseLDL [9] package for sparse matrix computation. The optimizer is a primal-dual interior point method that uses the Mehrotra predictor-corrector method. ECOS uses dynamic memory allocation only during the initialization phase, after which different instances of the same optimization problem can be solved without reinitialization.

Problems must be represented in the canonical form:

$$\begin{aligned} & \text{minimize } c^T x \\ & \text{subject to } Ax = b \\ & \quad Gx \preceq_K h \end{aligned}$$

where K represents the product of the positive orthant, second-order, and exponential cones. Formulating an optimization problem in canonical form often requires introducing slack variables and manipulating the equations for constraints and the objective function. As code generated by QCML uses this canonical form, the optimization problem can be represented in its original form but solved with ECOS or any other SOCP solver using the same canonical form.

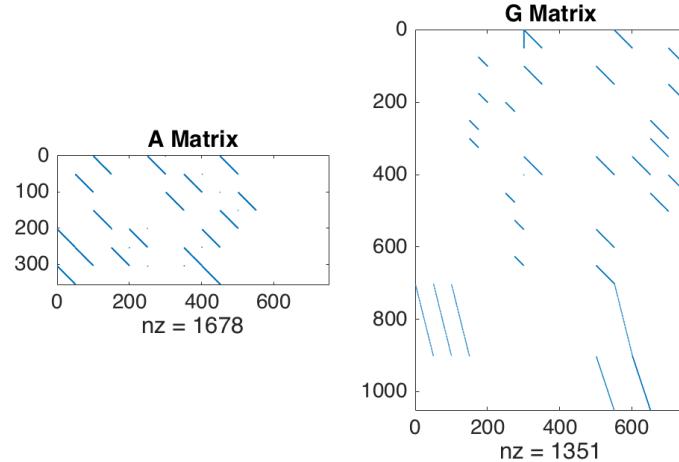


Figure 12: Sparsity pattern for A and G matrix variables in the skylight-landing problem.

5 Results

5.1 Scenario 1: Planar Divert

Variable	Units	Value
r_0	m	$[-800 \quad 0 \quad 250]$
v_0	m/s	$[30 \quad 0 \quad -20]$
r_f	m	$[0 \quad 0 \quad -200]$
v_f	m/s	$[0 \quad 0 \quad 0]$
I_{sp}	s	225
m_{wet}	kg	1680
m_{dry}	kg	1560
ρ_1	N	13258
ρ_2	N	4972
ϕ	deg	27
g	m/s^2	$[0 \quad 0 \quad -3.7114]$
θ	deg	0
p_r	m	250
p_b	m	100
p_f	m	-200
t_e^*	s	15.4
t_f^*	s	25.9
m_f^*	kg	1568.7

In this 2D landing scenario, a 1560kg lander has 120kg of fuel and targets a landing site in the center of a Martian pit. The pit is 200m deep and has a 250m radius. There is a 100m wall buffer and no glide slope constraint. A feasible and fuel-optimal trajectory is found after about 15 iterations of the simplex method used in the two-dimensional time search. t_e^* , t_f^* , and m_f^* are the optimal time of entry, time of flight, and landed mass.

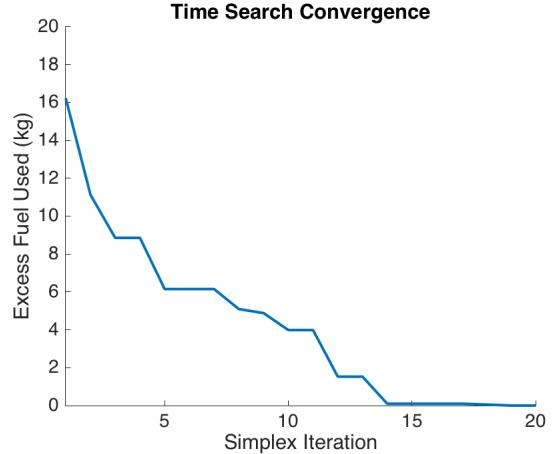


Figure 13: Excess fuel use compared to solution from exhaustive grid search for iterations of time search in Scenario 1.

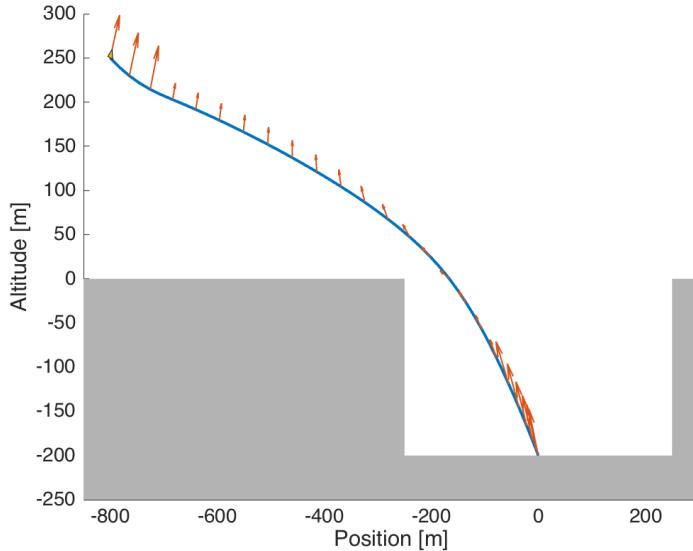


Figure 14: Fuel-optimal trajectory for Scenario 1. Exaggerated thrust command vectors are shown at every other discretized position.

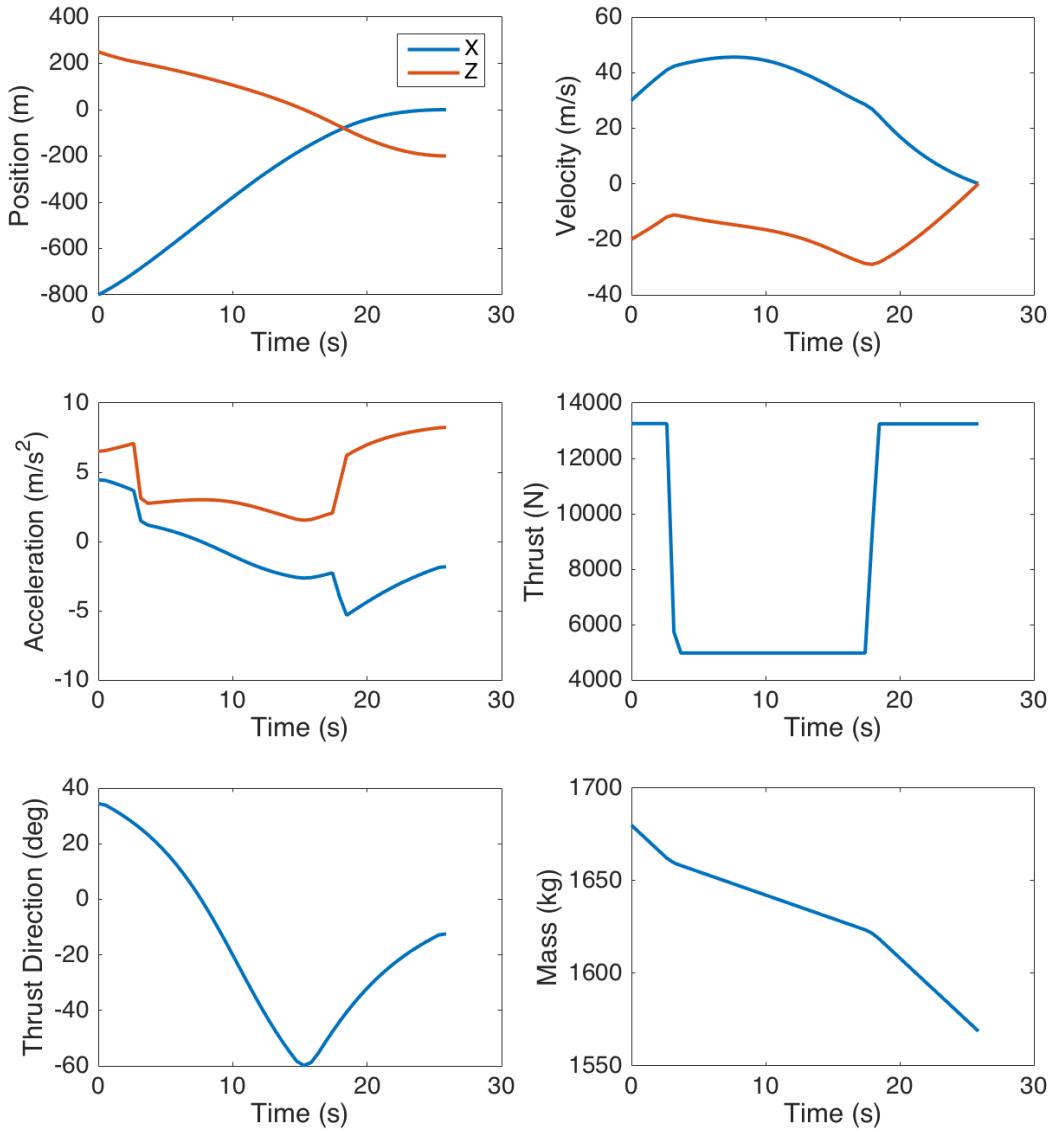


Figure 15: States from optimal trajectory solution of Scenario 1. Note the thrust is always maximized or minimized within an optimal trajectory.

5.2 Scenario 2: Descent Glide Slope

Variable	Units	Value
r_0	m	[−750 0 300]
v_0	m/s	[−15 0 −20]
r_f	m	[0 0 −200]
v_f	m/s	[0 0 0]
I_{sp}	s	225
m_{wet}	kg	1680
m_{dry}	kg	1530
ρ_1	N	13258
ρ_2	N	4972
ϕ	deg	27
g	m/s ²	[0 0 −3.7114]
θ	deg	20
p_r	m	250
p_b	m	100
p_f	m	−200
t_e^*	s	22.7
t_f^*	s	33.2
m_f^*	kg	1534.8

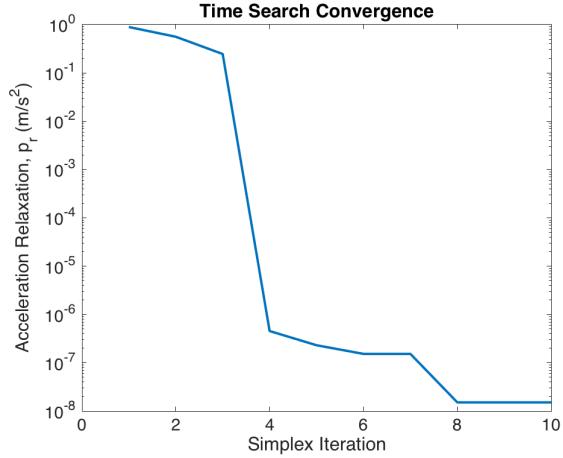


Figure 16: Magnitude of acceleration relaxation term for iterations of time search in Scenario 2. Initial iterations require additional acceleration capability but p_r is effectively 0 at convergence.

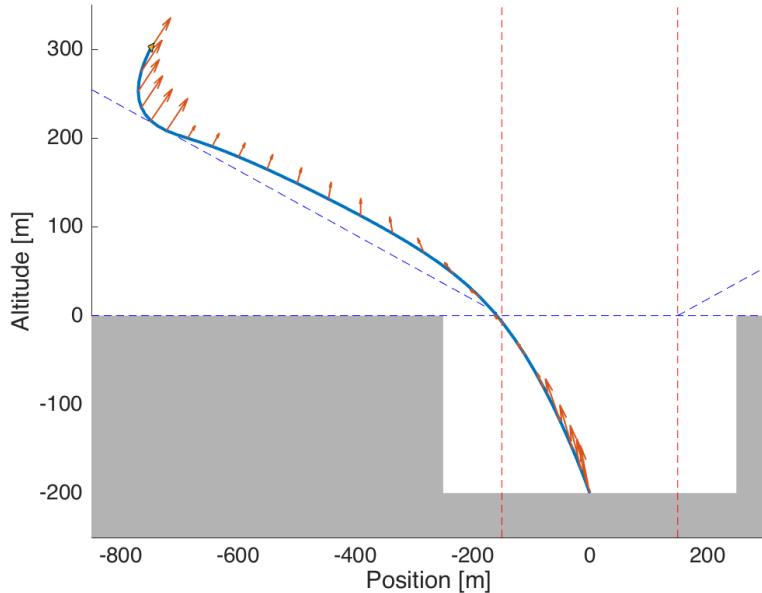


Figure 17: Fuel-optimal trajectory for Scenario 2. The descent and entry state constraints are shown with dashed lines.

The 20° descent glide slope constraint is active and forces the lander to make a more aggressive and costly maneuver. In general, more restrictive state constraints will increase the amount of fuel necessary.

The state constraints are applied to the discretized state variables and are therefore not guaranteed to be satisfied at times between variables. In this scenario, the glide slope and the pit wall buffer constraints are slightly violated between state variables. This can be resolved with the use of additional state constraints as in [3] or by simply using more conservative values for the existing constraints.

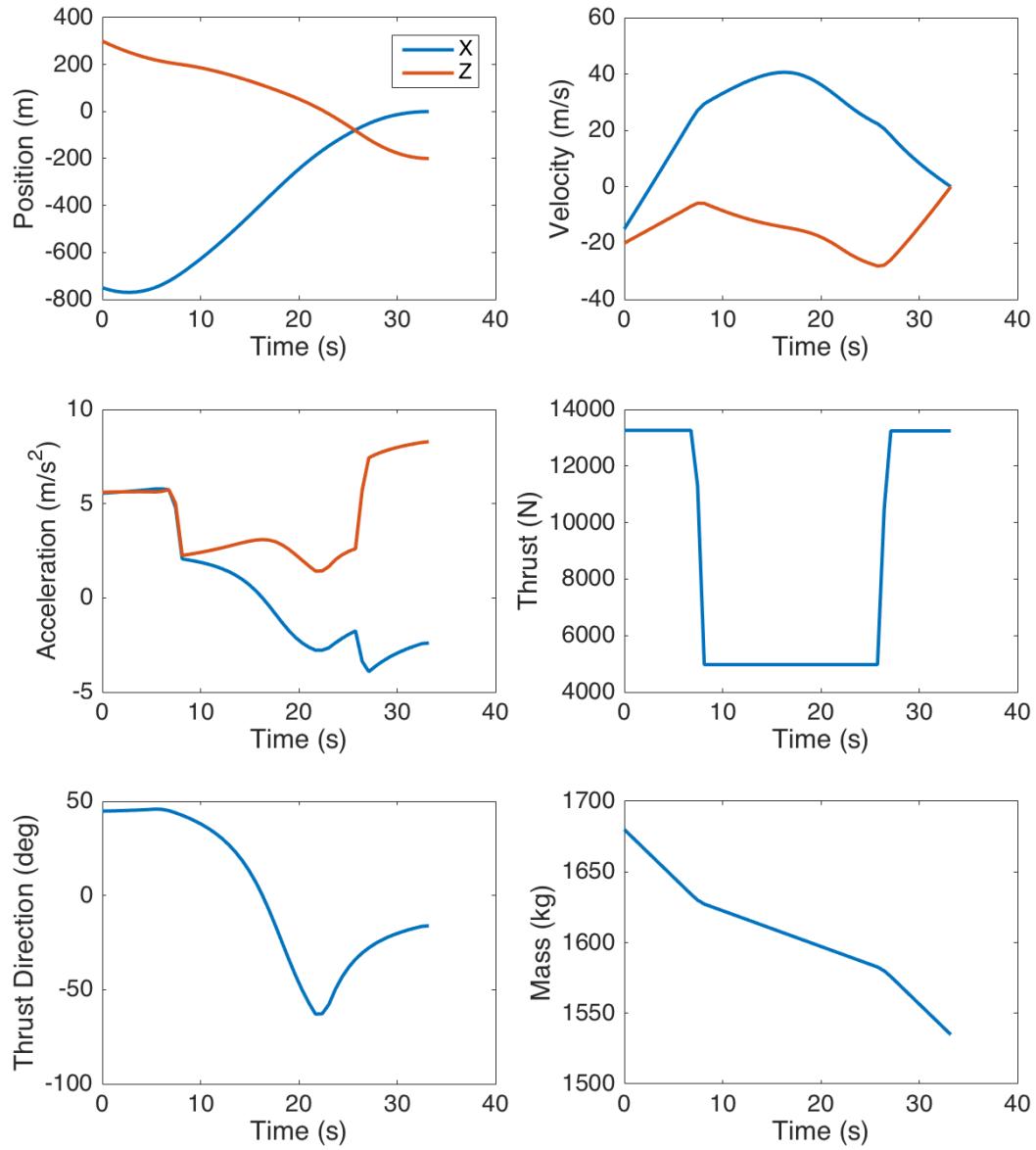


Figure 18: States from optimal trajectory solution of Scenario 2.

5.3 Scenario 3: 3D Braking

Variable	Units	Value
r_0	m	$[-500 \quad -200 \quad 300]$
v_0	m/s	$[45 \quad 35 \quad -20]$
r_f	m	$[0 \quad 0 \quad -200]$
v_f	m/s	$[0 \quad 0 \quad 0]$
I_{sp}	s	225
m_{wet}	kg	1650
m_{dry}	kg	1550
ρ_1	N	13258
ρ_2	N	4972
ϕ	deg	27
g	m/s^2	$[0 \quad 0 \quad -3.7114]$
θ	deg	0
p_r	m	200
p_b	m	100
p_f	m	-200
t_e^*	s	10.4
t_f^*	s	19.8
m_f^*	kg	1552.8

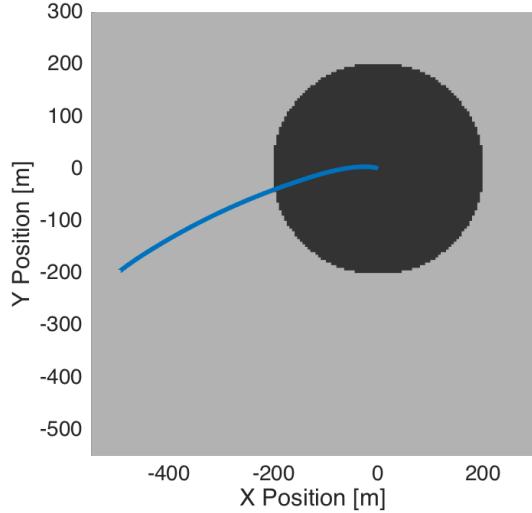


Figure 19: Top-down view of fuel-optimal trajectory in Scenario 3. The target landing site is out of plane with respect to the initial direction of motion.

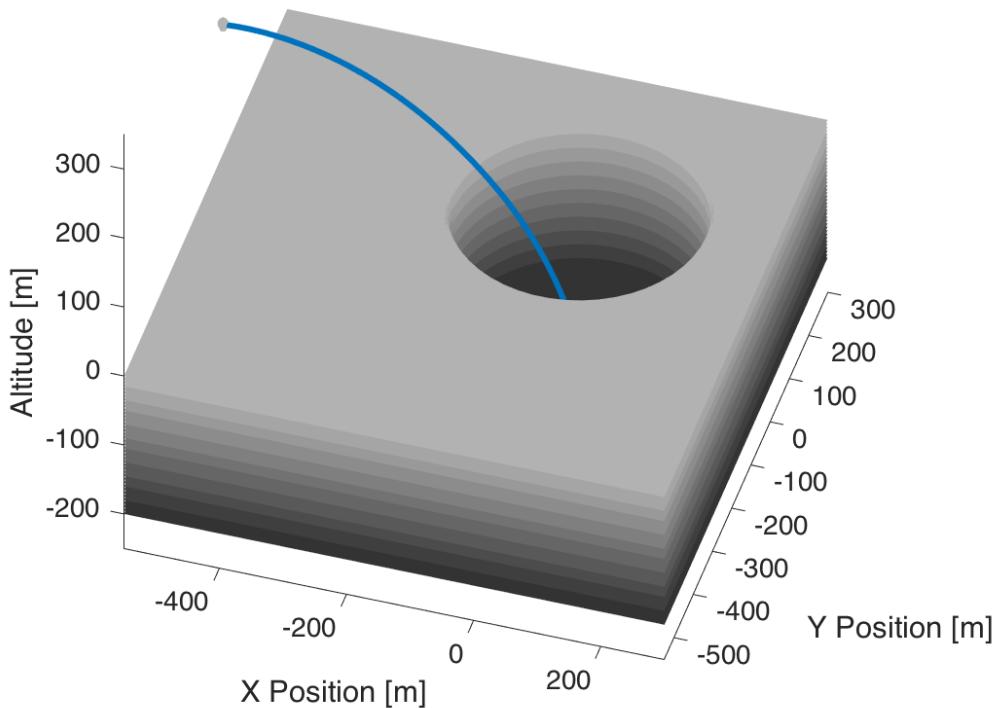


Figure 20: 3D view of fuel-optimal trajectory for Scenario 3. Depth within the pit is indicated by darker shades.

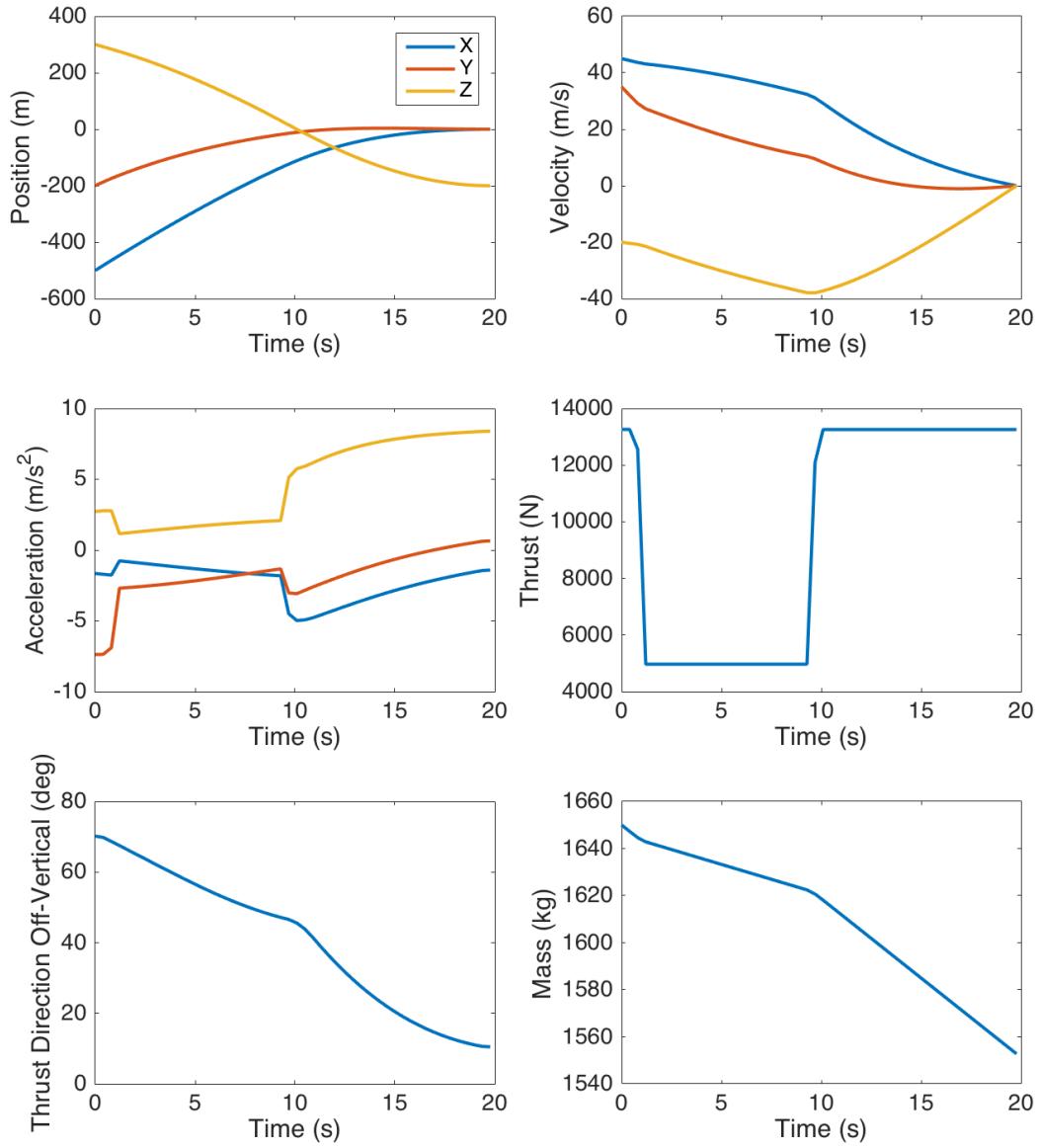
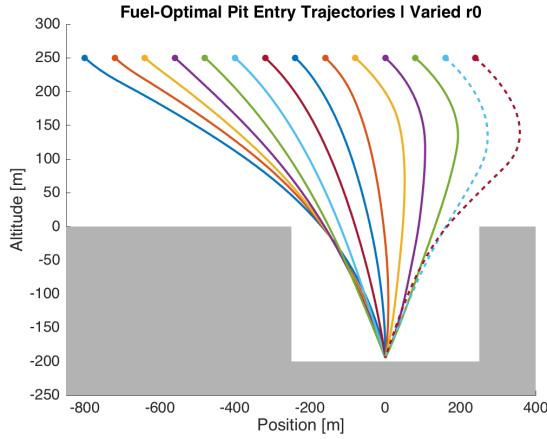


Figure 21: States from optimal trajectory solution of Scenario 3.

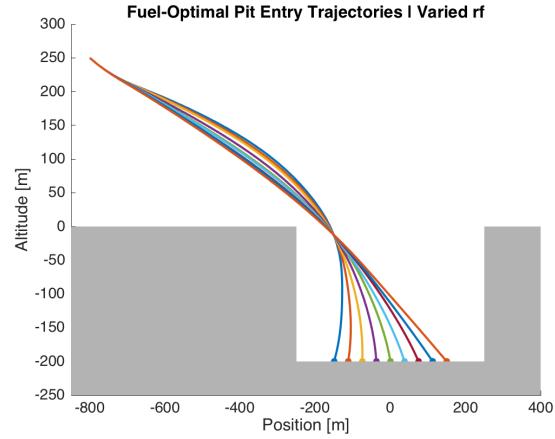
5.4 Scenario 4: Varied Parameters

Variable	Units	Value
r_0	m	$[-800 \quad 0 \quad 250]$
v_0	m/s	$[40 \quad 0 \quad -20]$
r_f	m	$[0 \quad 0 \quad -200]$
v_f	m/s	$[0 \quad 0 \quad 0]$
I_{sp}	s	225
m_{wet}	kg	1680
m_{dry}	kg	1560
ρ_1	N	13258
ρ_2	N	4972
ϕ	deg	27

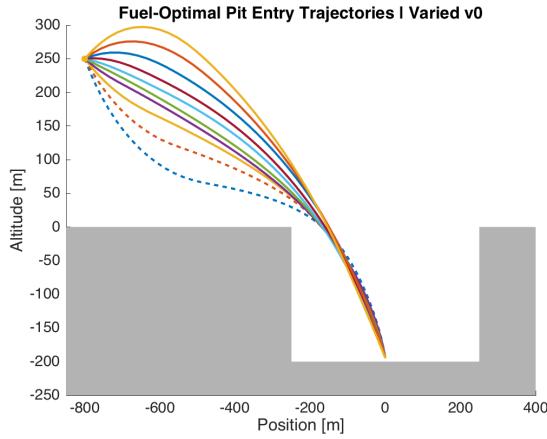
Variable	Units	Value
g	m/s^2	$[0 \quad 0 \quad -3.7114]$
θ	deg	0
p_r	m	250
p_b	m	100
p_f	m	-200



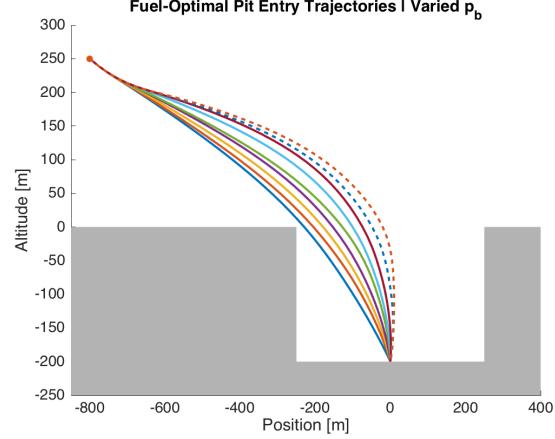
(a) r_{0x} is varied from -800 to 240 in increments of 80m.



(b) r_{fx} is varied from -150 to 150 in increments of 30m.



(c) $\|v_0\|$ is 44.7m/s with direction varied in increments of 10 degrees.



(d) p_b is varied from 40 to 240 in increments of 25m.

Figure 22: One variable from Scenario 4 is varied for each plot. Dashed lines indicate trajectories that do not satisfy the dry mass constraint. As the pit landing guidance algorithm is fuel-optimal, varied tests indicate the vehicle's full landing capabilities for a given amount of fuel.

5.5 Runtime

The runtime of the pit landing guidance problem is determined by the quantity and runtime of function evaluations performed during the time search. Each function evaluation consists of one trajectory optimization with ECOS for fixed t_e and t_f .

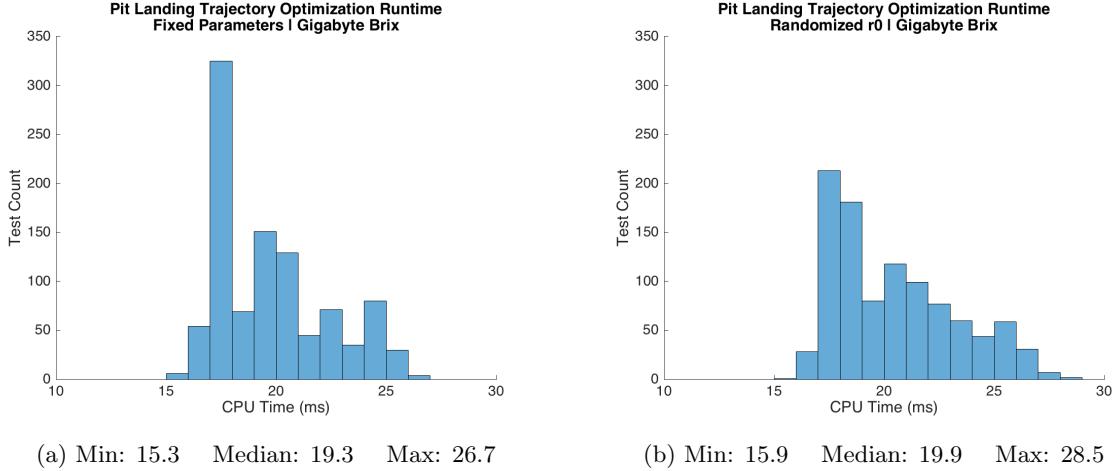


Figure 23: Variation in CPU time (ms) on Gigabyte Brix (Intel i7 2.5 - 3.1 GHz) over 1000 separate tests of pit landing algorithm with (a) fixed parameters and (b) randomized start position.

The C implementation of the pit landing algorithm is benchmarked on the Gigabyte Brix single board computer. The algorithm uses 50 discrete states to represent the trajectory, resulting in an SOCP problem with 701 variables, 355 linear constraints, and 1001 second-order cone constraints after canonicalization.

For repeated runs of a trajectory optimization problem with fixed parameters, variation in runtime comes from the dynamic memory allocation performed during the problem setup phase and context switches caused by other processes running on the same computer. Trajectory optimization problems with different parameters can take different numbers of iterations within the interior point method employed by ECOS to converge to an optimal solution (or identify infeasibility). On the Gigabyte Brix computer, the runtime of an individual trajectory optimization problem is effectively guaranteed to complete within 30ms.

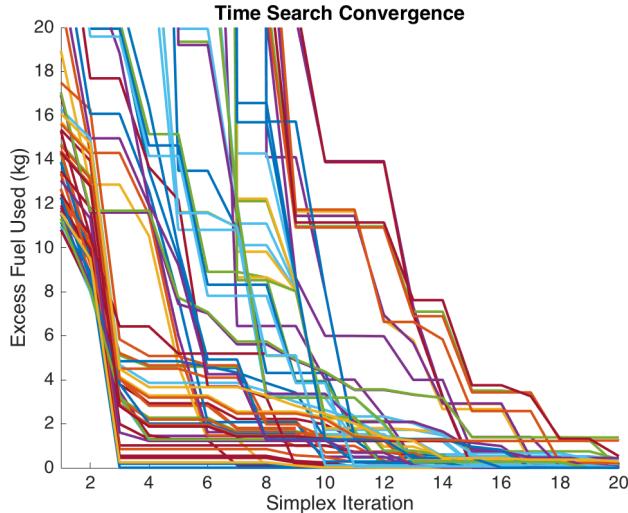


Figure 24: Convergence under randomized initial time of entry and time of flight for Scenario 1.

The time search optimization uses the Nelder-Mead simplex method [21] and will perform a maximum of two function evaluations per simplex iteration. For 80 tests of the time search in Scenario 1 with randomized initialization, 96% of tests reach under 2kg of the optimal landed mass within 20 simplex iterations. The total algorithm runtime is effectively upper bounded to 20 simplex iterations * 2 function evaluations * 30ms = 1.2 seconds.

The solution trajectory may need to be recalculated if there are large unpredicted perception or control errors. This can be performed with one trajectory optimization without repeating the time search, at the price of solution suboptimality. Trajectory replanning runtime is therefore upper bounded at 30ms and can easily be performed at low frequencies with little error introduced by calculation latency.

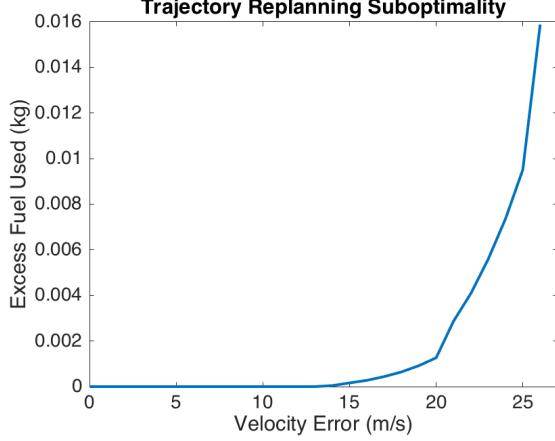


Figure 25: Excess fuel usage from replanning Scenario 1 trajectory to account for velocity error with fixed t_e, t_f . When the tracking error is small, there is effectively no impact of replanning without restarting the time search.

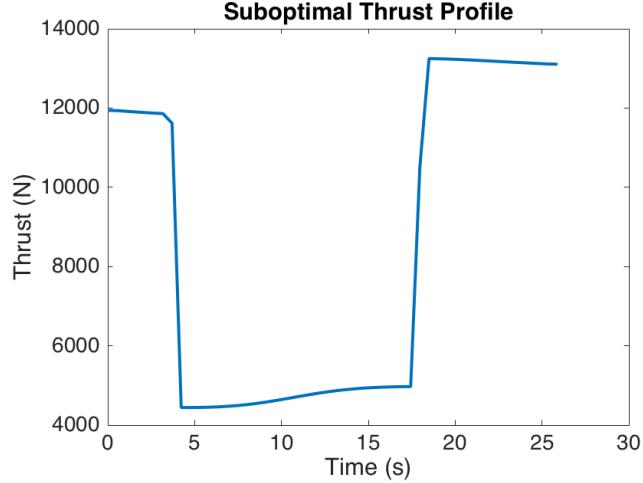


Figure 26: Example suboptimal thrust profile for Scenario 1 with 20m/s velocity error. Suboptimal thrust profiles will not always activate the upper or lower throttle constraints.

6 Conclusion

This research developed an algorithm to generate fuel-optimal landing trajectories into planetary pits. Pits and the subsurface voids they reach are compelling destinations for future scientific investigation and habitation. Previous approaches to autonomous landing guidance for propulsive spacecraft cannot safely and efficiently land within pits due to their challenging geometry. This research introduces convex time-varying state constraints and a two-dimensional time search to find globally optimal landing trajectories. Algorithm implementation is done with a modeling language and embedded solver for practical development and effective use. Testing in a variety of simulated landing scenarios verified the performance, runtime, and generality of the algorithm. Future autonomous spacecraft will safely and efficiently travel to new destinations powered by this work.

Downsides of this approach include the overall complexity of using an iterative optimization-based approach and issues with software implementation. Implementation of the pit landing problem with ECOS is sensitive to numerical scaling issues and can be unreliable if representing kilometer-scale trajectories in meters. QCML does not currently support for loops or matrix-matrix parameter multiplication in C-code generation, so some constraints must be rewritten in a less natural form.

7 Future Work

Future work may further develop theoretical convergence guarantees. Although the trajectory optimization problem is convex for fixed t_e and t_f , the general landing problem is not known to be convex. The relaxed objective function in the time search is guaranteed to have a convex domain, and simulations indicate the relaxed objective function in the two dimensional time search is unimodal. Proof of unimodality or convexity of the objective function would be beneficial for providing stronger convergence guarantees. These would guarantee a globally optimal solution trajectory for any valid parameters that describe the lander and its mission scenario.

Future work will extend the constraint capabilities of fuel-optimal landing guidance. Existing techniques can limit the range of thrust pointing angle [4] and the rate of change of thrust magnitude, but not the rate of change of thrust pointing angle. Vision-based navigation systems are sensitive to motion blur caused by rapid rotation and may impose conservative limits on the allowable vehicle attitude rate when active.

Future work will explore further applications of convex optimization to spacecraft guidance and control. The point mass assumption has been shown to be appropriate for landing spacecraft with sufficient controllability [2] [29]. One possible method for vehicle control is to map point mass commands from a simple PID controller to actual vehicle thruster commands using convex optimization [30]. The problem formulation consists of one optimization to minimize the error of desired versus feasible thrust commands and another to minimize fuel consumption. Future work may formulate and solve the thruster allocation problem using QCML and ECOS.

Acknowledgments

I would like to thank my advisor Red Whittaker for his consistent and invaluable support of my research work. I have learned so much from his example and mentorship. Behcet Acikmese is a pioneer of convex optimization techniques for spacecraft landing guidance and I am grateful for his patient explanation of that work. I am also thankful to Eric Chu and Alexander Domahidi for their work on QCML and ECOS and help in applying these tools to the pit landing problem and debugging issues.

Chris Atkeson, Chris Cunningham, Miki Szmuk, and Matt Wytock provided helpful comments for fixing implementation issues and researching future extensions. Engineers at Astrobotic Inc. including Joan Yule, Kevin Peterson, and Aaron Acton provided inspiration and insightful discussion. Lastly I am thankful for all the ways my parents and sister support me.

References

- [1] B. Acikmese, J. Casoliva, J. M. Carson, and L. Blackmore. G-FOLD: A Real-Time Implementable Fuel Optimal Large Divert Guidance Algorithm for Planetary Pinpoint Landing. In *Concepts and Approaches for Mars Exploration*, volume 1679 of *LPI Contributions*, page 4193, June 2012.
- [2] Behçet Açıkmeşe, M Aung, Jordi Casoliva, Swati Mohan, Andrew Johnson, Daniel Scharf, David Masten, Joel Scotkin, Aron Wolf, and Martin W Regehr. Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing. In *AAS/AIAA spaceflight mechanics meeting*, 2013.
- [3] Behçet Açıkmeşe, Lars Blackmore, Daniel P Scharf, and Aron Wolf. Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008.
- [4] Behcet Acikmese, John M Carson, and Lars Blackmore. Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem. *Control Systems Technology, IEEE Transactions on*, 21(6):2104–2113, 2013.
- [5] Behcet Acikmese 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.
- [6] PJ Boston, MN Spilde, DE Northup, LA Melim, DS Soroka, LG Kleina, KH Lavoie, LD Hose, LM Mallory, CN Dahm, et al. Cave Biosignature Suites: Microbes, Minerals, and Mars. *Astrobiology*, 1(1):25–55, 2001.
- [7] Eric Chu, Neal Parikh, Alexander Domahidi, and Stephen Boyd. Code Generation for Embedded Second-Order Cone Programming. In *Control Conference (ECC), 2013 European*, pages 1547–1552. IEEE, 2013.
- [8] GE Cushing, TN Titus, JJ Wynne, and PR Christensen. THEMIS Observes Possible Cave Skylights on Mars. *Geophysical Research Letters*, 34(17), 2007.
- [9] Timothy A Davis. Algorithm 849: A Concise Sparse Cholesky Factorization Package. *ACM Transactions on Mathematical Software (TOMS)*, 31(4):587–591, 2005.
- [10] NASA Space Technology Mission Directorate. Flight Opportunities 2014 Annual Report. <https://flightopportunities.nasa.gov/media/uploads/69/FY14-AR-PAGES-HIGHRES.pdf>, December 2015.
- [11] Alexander Domahidi, Eric Chu, and Stephen Boyd. ECOS: An SOCP Solver for Embedded Systems. In *Control Conference (ECC), 2013 European*, pages 3071–3076. IEEE, 2013.
- [12] Daniel Dueri, Jing Zhang, and Behçet Açıkmeşe. Automated Custom Code Generation for Embedded, Real-Time Second Order Cone Programming. In *IFAC world congress*, 2014.
- [13] Michael Grant, Stephen Boyd, and Yinyu Ye. *Disciplined Convex Programming*. Springer, 2006.
- [14] Michael Grant, Stephen Boyd, and Yinyu Ye. CVX: Matlab Software for Disciplined Convex Programming, 2008.
- [15] W. Harris. Comparison of Guidance Methods for Autonomous Planetary Lander. Master’s thesis, Rice University, July 2011.
- [16] Junichi Haruyama, Kazuyuki Hioki, Motomaro Shirao, Tomokatsu Morota, Harald Hiesinger, Carolyn H van der Bogert, Hideaki Miyamoto, Akira Iwasaki, Yasuhiro Yokota, Makiko Ohtake, et al. Possible Lunar Lava Tube Skylight Observed by SELENE Cameras. *Geophysical Research Letters*, 36(21), 2009.
- [17] Junichi Haruyama, Tomokatsu Morota, Shingo Kobayashi, Shujiro Sawai, Paul G Lucey, Motomaro Shirao, and Masaki N Nishino. Lunar Holes and Lava Tubes as Resources for Lunar Science and Exploration. In *Moon*, pages 139–163. Springer, 2012.

- [18] Grant Heiken, David Vaniman, and Bevan M French. *Lunar Sourcebook: A User's Guide to the Moon*. CUP Archive, 1991.
- [19] Andrew E Johnson, Yang Cheng, James Montgomery, Nikolas Trawny, Brent E Tweddle, and Jason Zheng. Design and Analysis of Map Relative Localization for Access to Hazardous Landing Sites on Mars. In *AIAA Guidance, Navigation, and Control Conference*, page 0379, 2016.
- [20] Allan R Klumpp. Apollo Lunar Descent Guidance. *Automatica*, 10(2):133–146, 1974.
- [21] John A Nelder and Roger Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [22] Lujendra Ojha, Alfred McEwen, Colin Dundas, Shane Byrne, Sarah Mattson, James Wray, Marion Massé, and Ethan Schaefer. HiRISE Observations of Recurring Slope Lineae (RSL) during Southern Summer on Mars. *Icarus*, 231:365–376, 2014.
- [23] Lujendra Ojha, Mary Beth Wilhelm, Scott L Murchie, Alfred S McEwen, James J Wray, Jennifer Hanley, Marion Massé, and Matt Chojnacki. Spectral Evidence for Hydrated Salts in Recurring Slope Lineae on Mars. *Nature Geoscience*, 8(11):829–832, 2015.
- [24] Richard W Orloff. *Apollo by the Numbers*. Springer-Verlag, New York, 1996.
- [25] Scott Ploen, Bechet Acikmese, and Aron Wolf. A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing. In *AIAA Guidance, Navigation, and Control Conference, Keystone, CO*, 2006.
- [26] Ravi Prakash, P Dan Burkhardt, Allen Chen, Keith A Comeaux, Carl S Guernsey, Devin M Kipp, Leila V Lorenzoni, Gavin F Mendeck, Richard W Powell, Tommaso P Rivellini, et al. Mars Science Laboratory Entry, Descent, and Landing System Overview. In *Aerospace Conference, 2008 IEEE*, pages 1–18. IEEE, 2008.
- [27] MS Robinson, JW Ashley, AK Boyd, RV Wagner, EJ Speyerer, B Ray Hawke, H Hiesinger, and CH van der Bogert. Confirmation of Sublunarean Voids and Thin Layering in Mare Deposits. *Planetary and Space Science*, 69(1):18–27, 2012.
- [28] David Rutishauser, Chirol D Epp, and Edward A Robertson. Free-Flight Terrestrial Rocket Lander Demonstration for NASA's Autonomous Landing and Hazard Avoidance Technology (ALHAT) System. *Proc. of AIAA SPACE*, 2012, 2012.
- [29] Daniel P Scharf, Martin W Regehr, Geoffrey M Vaughan, Javier Benito, Homa Ansari, Min Hane Aung, Andrew Johnson, Jordi Casoliva, Swati Mohan, Daniel Dueri, et al. ADAPT Demonstrations of Onboard Large-divert Guidance with a VTOL Rocket. In *Aerospace Conference, 2014 IEEE*, pages 1–18. IEEE, 2014.
- [30] Pablo A Servidio and R Pena. Spacecraft Thruster Control Allocation Problems. *IEEE Transactions on Automatic Control*, 50(2):245–249, 2005.
- [31] Bradley A Steinfeldt, Michael J Grant, Daniel A Matz, Robert D Braun, and Gregg H Barton. Guidance, Navigation, and Control System Performance Trades for Mars Pinpoint Landing. *Journal of Spacecraft and Rockets*, 47(1):188–198, 2010.
- [32] Michael Szmuk and Behcet A Acikmese. Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints. In *AIAA Guidance, Navigation, and Control Conference*, page 0378, 2016.
- [33] Robert V Wagner and Mark S Robinson. Distribution, Formation Mechanisms, and Significance of Lunar Pits. *Icarus*, 237:52–60, 2014.
- [34] Aron A Wolf, Claude Graves, Richard Powell, and Wyatt Johnson. *Systems for Pinpoint Landing at Mars*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2004.