State Constraint Analysis for Powered Descent Guidance

Samuel Hornus, Jean-Baptiste Caillau, Lamberto Dell'Elce, Jean-Baptiste Pomet

State Constraint Analysis for Powered Descent Guidance

Samuel Hornus,[1] Jean-Baptiste Caillau,[2] Lamberto Dell'Elce,[3] Jean-Baptiste Pomet [3]
[1] Université de Lorraine, Inria, Nancy, France
[2] Université Côte d'Azur, CNRS, Inria, LJAD
[3] Université Côte d'Azur, Inria, CNRS, LJAD

May 19, 2026

Introduction

The powered-descent guidance (PDG) problem for soft planetary landing has received considerable attention in recent years, driven by applications to reusable launchers and robotic missions to celestial bodies. The problem seeks to determine the optimal trajectory that brings a vehicle to a prescribed landing site while minimizing fuel consumption and satisfying constraints on thrust magnitude, thrust direction, and obstacle avoidance. [1] pioneered the use of convex optimization techniques for this problem, demonstrating real-time computation capabilities. [2] develops a complete solution using and indirect method for solving the PDG, but without constraint on the state or the thrust direction. More recently, [3] proposed a direct-indirect hybrid strategy that combines the computational efficiency of direct methods with the optimality guarantees of indirect approaches, but still without state or control constraint. [4] revisited the problem, providing new insights into the structure of optimal solutions.

While finite-dimensional optimization techniques have proven highly successful for real-time implementation, the mathematical structure of the PDG problem with combined control and state constraints remains incompletely understood. The recent work of [5] highlighted open questions regarding the characterization of optimal trajectories, particularly in the presence of glide-slope constraints for obstacle avoidance. The present work addresses these gaps by providing an explicit expression for the control along state-constrained arcs and implementing a complete indirect method that accounts for both pointing constraints (on thrust direction) and glide-slope constraints (on vehicle position). We systematically explore the space of trajectory structures through extensive numerical experiments, revealing the rich variety of optimal solutions that emerge from different initial conditions.

The paper is organized as follows. Section 1 formulates the PDG problem with three-dimensional motion, minimal fuel objective, and combined control and state constraints, then derives the necessary optimality conditions from Pontryagin's maximum principle. Section 2 analyzes the interaction between pointing and glide-slope constraints, providing the key expression for the Lagrange multiplier along state-constrained arcs. Section 3 describes our numerical approach, combining direct optimization for initialization with indirect shooting methods for high-precision solutions. In Section 4 we present extensive numerical experiments revealing 56 distinct trajectory structures. Section 5 analyzes the conditions under which both constraints can be simultaneously active.

1 The powered-descent guidance problem

We further the study of the powered-descent guidance (PDG) problem in the following context:

- no drag; we assume the velocity in this late stage is small enough that drag can be ignored
- constant gravity; again, gravity is assumed to not vary significantly along an optimal trajectory
- the control is constrained with a thrust pointing (directional) constraint, and magnitude lower and upper bounds
- glide-slope *state* constraint

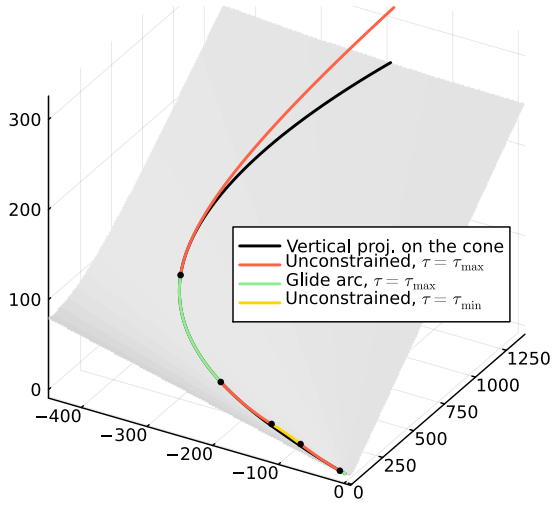

Figure 1: A sample trajectory with six arcs including two glide arcs (green). The last arc is very short; please zoom in to view it.

We use subscripts x, y and z for the cartesian coordinates. The z-axis is aligned with gravity and points upwards. For any 3D vector $p = (p_x, p_y, p_z)$, we define $\overline{p} = (p_x, p_y, 0) \in \mathbb{R}^3$, which we sometimes interpret as $(p_x, p_y) \in \mathbb{R}^2$ when convenient. We also define $\widehat{p} = \frac{p}{\|p\|}$.

The system state is $\boldsymbol{x} = (r, v, m) \in \mathbb{R}^7$. $m(t)$ is the vehicle mass at time t. The maximal thrust that the vehicle can provide is denoted T.

We consider a *glide-slope* constraint on the state: $h(r) \equiv r_z - \tan\gamma\|\overline{r}\| \geq 0$, where $0 \leq \gamma < \pi/2$. The glide-slope constraint defines an inverted cone with apex at the origin and forces the vehicle to stay above the cone. This models the avoidance of the ground and small hills and boulders around the landing site. We define $n(r) \equiv \left(-\tan\gamma\widehat{\overline{r}}; 1\right)$, the (non-unit) normal to the glide-slope cone at any point in the intersection of the cone and the vertical plane through r and the origin (when that plane is unique). The normal vector is not defined when $\overline{r} = 0$ (see remark 2.4 in [5]). Then $h(r) = n(r) \cdot r$ and $n(r) = \nabla h$. When the system moves along a trajectory, we use the shorthand $n(t) \equiv n(r(t))$.

The system's control is $u \in U \subset \mathbb{R}^3$. It is such that Tu is the instantaneous thrust vector. The set U is static, non-convex and defined by two constraints:

- a constraint on the norm of u: $0 < \tau_{\min} \leq \|u\| \leq \tau_{\max} \leq 1$, and
- a constraint on the direction of u: $u_z \geq \|u\| \cos\theta^\star$, which we call the *pointing constraint*.

The parameter $\theta^\star$ can be arbitrary, but some nice properties hold when $\theta^\star < \frac{\pi}{2}$ (Section 2), or when $\theta^\star \in (\gamma, \pi - \gamma)$ (Section 6). We write the control as $u(t) = \tau(t)d(t)$ where the unit vector d is the thrust direction, and $\tau(t) \in [\tau_{\min}, \tau_{\max}]$ is the throttle. The pointing constraint also writes as $d_z \geq \cos\theta^\star$. The powered descent guidance (PDG) problem asks for the control $u(t)$ that minimizes fuel consumption, *i.e.* maximizes the final mass $m(t_f)$. It is defined formally as follows:

$$
\begin{aligned}
\min_{u(t)\in U} \; & -m(t_f) \quad \text{where } t_f \text{ is the free final time, subject to} \\
& \dot{r}(t) = v(t) \\
& \dot{v}(t) = \frac{T}{m(t)}u(t) + g = K(t)d(t) + g, \quad \text{with } K(t) = \frac{T\tau(t)}{m(t)} \\
& \dot{m}(t) = -\alpha T\|u(t)\| = -\alpha T\tau(t) \\
& h(r(t)) \geq 0, \quad u_z(t) \geq \|u(t)\|\cos\theta^\star, \quad \tau_{\min} \leq \|u(t)\| \leq \tau_{\max}, \\
& r(0) = r_0, \quad v(0) = v_0 \quad \text{and} \quad r(t_f) = v(t_f) = 0.
\end{aligned} \tag{1}
$$

The constant parameter $\alpha > 0$ adjusts the fuel flow, see Section 5.

Remark 1. *(i) The vector n is normal to the glide cone, so $n(t)$ lives on a horizontal circle located at height $z = 1$ and radius $\tan\gamma$. $\dot{n}(t)$ is tangent to this circle at $n(t)$, and $\ddot{n}(t) \cdot r(t) \geq 0$.*

(ii) The z component of $\dot{n}$ is zero, therefore, for any vector w, we have $\dot{n} \cdot w = \dot{n} \cdot \overline{w}$. In particular, $\dot{n} \cdot \dot{v} = \frac{T}{m}\dot{n} \cdot u = K\dot{n} \cdot d$.

Remark 2. *The function $K : \mathbb{R}^+ \mapsto \mathbb{R}^+$ above is used to simplify expressions. On an interval with constant $\tau \equiv \|u\|$, it has the following derivatives: $K = T\tau/m, \quad \dot{K} = \alpha K^2, \quad \ddot{K} = 2\alpha^2 K^3$.*

Let $p^0 \in \mathbb{R}, \boldsymbol{p} = (p_r, p_v, p_m) \in \mathbb{R} \mapsto \mathbb{R}^3$ be the costate vector. We adjoin the glide slope constraint with $\mu(t) \geq 0$, to the hamiltonian:

$$\begin{aligned} H(\boldsymbol{x}, u, \boldsymbol{p}, \mu) &= p_r \cdot v + p_v \cdot \left(\frac{T}{m}u + g\right) - p_m \alpha T\|u\| + \mu h(r) \\ &= p_r \cdot v + p_v \cdot g + \mu h(r) + \left(\frac{T}{m}p_v \cdot u - p_m \alpha T\|u\|\right) \\ &= H_0 + TH_1, \text{ with } H_0 = p_r \cdot v + p_v \cdot g + \mu h(r) \text{ and } H_1 = \tau\left(\frac{1}{m}p_v \cdot d - p_m\alpha\right). \end{aligned}$$

H does not depend on p^0 since the "running cost" is zero. H_0 is the part of H that does not depend on the control u. From Pontryagin's maximum principle, we know that if u is an optimal control, then there exists $p^0 \leq 0$, an adjoint vector $\boldsymbol{p}$ that is absolutely continuous and $\mu \geq 0$, such that

$$\begin{aligned} \dot{p}_r(t) &= -\frac{\partial H}{\partial r}(t) = -\mu(t)\frac{\partial h}{\partial r}(t) = -\mu(t)n(t), \\ \dot{p}_v(t) &= -\frac{\partial H}{\partial v}(t) = -p_r(t), \\ \dot{p}_m(t) &= -\frac{\partial H}{\partial m}(t) = \frac{T}{m^2(t)}p_v(t) \cdot u(t), \\ &\operatorname{supp}\mu(t) \subset \{t \mid h(r(t)) = 0\}. \end{aligned} \tag{2}$$

Note that p_r is constant on intervals along which the glide slope constraint is not active. On the same intervals, p_v is linear. The final time t_f is free, as is the final mass. Therefore:

$$\begin{aligned} H(t_f) &= -p^0 \left.\frac{\partial(-m)}{\partial t}\right|_{t=t_f} = 0, \\ p_m(t_f) &= p^0 \left.\frac{\partial(-m)}{\partial m}\right|_{t=t_f} = -p^0 \geq 0. \end{aligned}$$

In our implementation, we assume that trajectories are not abnormal and use $p_0 = -1$.

1.1 Maximum principle

The statements in this paragraph come from [5]. The optimal control maximizes H_1 over $\{(\tau, d) \mid u = \tau d \in U\}$. τ is positive because $\tau \geq \tau_{\min} > 0$. Thus, the optimal control's direction, d, maximizes

$$\Psi(d) = \frac{1}{m}p_v \cdot d - p_m\alpha, \tag{3}$$

which is linear. Maximizing a linear functional over a bounded set is equivalent to maximizing it over the convex hull of the set. Here, we thus maximize over the intersection of the unit ball and a half-space. In non-degenerate cases, we get

$$d = \begin{cases} \left(\sin\theta^\star \widehat{\overline{p_v}}, \cos\theta^\star\right) & \text{if } p_{v,z} \leq \|p_v\|\cos\theta^\star, \text{ and we say that the pointing constraint is active,} \\ \widehat{p_v} & \text{"otherwise."} \end{cases} \tag{4}$$

(We omit the two cases ($\overline{p_v} = 0$ and $p_{v,z} < 0$) and $p_v = 0$ from consideration). Finally, $\tau = \begin{cases} \tau_{\min} & \text{if } \Psi(d) < 0, \\ \tau_{\max} & \text{if } \Psi(d) > 0 \end{cases}$ Again, in our implementation, we ignore the possibility of singular arc along which Ψ is zero. We defer their treatment to future work.

1.2 Structure of an optimal trajectory

Leparoux *et al.* show that all optimal trajectories have a Max-(Min or Singular)-Max structure for τ, wherein each arc can degenerate to nonexistence. Excluding singular arcs, we have three binary cases that govern the optimal control along an arc: 1) what is the sign of Ψ?, 2) is the pointing constraint active?, 3) is the glide-slope constraint active?

2 The pointing constraint

In this section, we first assume the glide-slope state constraint is not active. The pointing constraint is neither a state nor a mixed constraint, but it is important to consider it separately because it makes the control set U non-smooth. When the state (glide-slope) constraint is not active, the position adjoint, p_r, is constant and the velocity adjoint, p_v, follows a straight line motion: $p_v(t) = p_v^0 - tp_r$. The path of $\widehat{p_v}$ is the projection of that line segment onto the unit-sphere S^2. We see that $\widehat{p_v}$ moves *monotonously* along a half great-circle of S^2 and while the pointing constraint is not active, the thrust direction d equals $\widehat{p_v}$.

When the pointing constraint becomes active, d changes course non-continuously and moves along the horizontal circle $C = \{\|\overline{d}\| = \sin\theta^\star, d_z = \cos\theta^\star\}$.

2.0.1 An interesting interaction between state and control constraints.

Now, let us assume that $\theta^\star < \pi/2$. The space of d is then a spherical cap centered at the north pole with half-angle $\theta^\star$. The boundary of that cap is the circle C defined above. C is a circle smaller than a great-circle. Therefore $\dot{d}$ always shows a discontinuity when the pointing constraint becomes active (*i.e.* when $\widehat{p_v}(t)$ crosses C southward). More importantly, once $\widehat{p_v}$ exits the inside of the cap, it can not re-enter it, because $\widehat{p_v}$ moves along a *half* great-circle. The only way for $\widehat{p_v}$ to get back inside (*i.e.*, to deactivate the pointing constraint) is for $p_v(t)$ to change its direction $p_r(t)$, which can happen only by activating the glide slope state constraint, because $\dot{p}_r = -\mu n$ where μ is the adjoint (SH: multiplier?) of the state constraint. The other possible situation is when $\widehat{p_v}$ starts its trajectory outside the spherical cap. It can then enter once and exit once (as long as the state constraint stays inactive, of course). In our experiment, we observe that $\approx 76\,\%$ of optimal trajectories have $\widehat{p_v}(0)$ outside of the cap (the pointing constraint is active) and $\widehat{p_v}$ enters the cap later on and stays inside until the end, with or without activation of the state constraint.

3 The glide-slope state constraint

We now assume that the control pointing constraint is inactive and compute the control during an arc of positive length along which the glide slope state constraint is active, *i.e.* $h(t) = 0$. In this section, we derive an expression for $\mu(t)$, the multiplier of the glide-slope constraint. The symbol μ appears in the fourth derivative of h. Inside the constrained arc, h and its derivatives are zero.

We have $h(t) \equiv h(r(t)) = 0 = n(t) \cdot r(t)$ where $n(t) \equiv \left(-\tan\gamma\,\widehat{\overline{r}(t)}; 1\right)$. In the expression of n, the vector $\overline{r}$ is normalized, so its time derivative is orthogonal to $\overline{r}$, therefore: $\dot{n} \cdot r = 0$ and, dropping the dependence on t, $\dot{h} = n \cdot v$. So along the constrained arc, we also have $n(t) \cdot v(t) = 0$ (*i.e.* the velocity is in the tangent plane to the cone at $r(t)$). To help in the computation of the next derivatives, we use the auxiliary function f(see Section A), so that $\dot{n} = \frac{\tan\gamma}{\|\overline{r}\|^3} f(\overline{r}, \overline{v})$. From Equation (29) (Cauchy-Schwartz), we see that $\dot{n} \cdot v \leq 0$. Recall that the vector g is the local gravity: $g = (0, 0, -g_0)$ with $g_0 > 0$.

On the constrained arc: $\ddot{h} = 0 = \dot{n} \cdot v + n \cdot \dot{v} = \dot{n} \cdot v + Kn \cdot d - g_0$. We find

$$\ddot{h} = Kn \cdot d - (g_0 - \dot{n} \cdot v) \equiv Kn \cdot d - G(\boldsymbol{x}) = 0. \tag{5}$$

Note that $G(\boldsymbol{x}) = g_0 - \dot{n} \cdot v$ is positive, so the thrust pushes us, thankfully, away from and above the cone.

Remark 3. *Normalizing n we obtain $K\hat{n} \cdot d = \cos(\gamma)g_0 - \dot{n} \cdot v$. Since $\hat{n}$ and d are unit vectors, there is a solution (for d) if and only if $K \geq \cos(\gamma)g_0 - \dot{n} \cdot v$. Observe that $\cos(\gamma)g_0 - \dot{n} \cdot v \geq \cos(\gamma)g_0$. Thus, if $K < \cos(\gamma)g_0 \Leftrightarrow T\tau < m\cos(\gamma)g_0$ then we can not have a constrained arc of positive length. We may only have a "touch point".*

3.1 The third derivative of h

We assume that $p_v \neq 0$ a.e. on the constrained interval, so that $d = \frac{p_v}{\|p_v\|}$: an expression that we can differentiate: $\dddot{h} = \ddot{n} \cdot v + 2K\dot{n} \cdot d + Kn \cdot \dot{d} + \dot{K}n \cdot d$. Omitting the detailed derivation, the components are:

$$\dot{d} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{p_v}{\|p_v\|}\right) = \frac{1}{\|p_v\|}\left((p_r \cdot d)d - p_r\right) = \frac{1}{\|p_v\|} f(d, p_r), \tag{6}$$

$$\dot{n} = \frac{\tan\gamma}{\|\overline{r}\|^3} f(\overline{r}, \overline{v}),$$

$$\ddot{n} = -3R\dot{n} + \frac{K\tan\gamma}{\|\overline{r}\|^3} f(\overline{r}, \overline{d}) - \frac{\tan\gamma}{\|\overline{r}\|^3} f(\overline{v}, \overline{r}), \quad \text{(using (31))}$$

$$\ddot{n} \cdot v = -3R\dot{n} \cdot v + \frac{K\tan\gamma}{\|\overline{r}\|^3} f(\overline{r}, \overline{d}) \cdot v - 0 \quad \text{(using (28) and (27))}$$

$$= -3R\dot{n} \cdot v + \frac{K\tan\gamma}{\|\overline{r}\|^3} f(\overline{r}, \overline{v}) \cdot d \quad \text{(using (28))}$$

$$\ddot{n} \cdot v = -3R\dot{n} \cdot v + K\dot{n} \cdot d. \tag{7}$$

Substituting, we get:

$$\dddot{h} = 3(Kd - Rv) \cdot \dot{n} + (K\dot{d} + \dot{K}d) \cdot n. \tag{8}$$

Remark 4. *The function $R : \mathbb{R}^+ \mapsto \mathbb{R}$ above is introduced to simplify expressions. It has the following derivatives:*

$$R = \frac{\overline{r} \cdot \overline{v}}{\|\overline{r}\|^2}, \quad \dot{R} = \frac{\|\overline{v}\|^2 + K\overline{r} \cdot \overline{d}}{\|\overline{r}\|^2} - 2R^2.$$

3.2 Finding μ with the fourth derivative of h

Let's continue to differentiate so we can find μ. We differentiate $\dddot{h}$ and substitute (7) again:

$$\begin{aligned} h^{(4)} = {} & 3K\ddot{n} \cdot d \\ & + \dot{n} \cdot \left(3\left(3R^2 - \dot{R}\right)v + \left(4\dot{K} - 6RK\right)d + 4K\dot{d}\right) \\ & + n \cdot \left(\ddot{K}d + 2\dot{K}\dot{d}\right) + Kn \cdot \ddot{d}. \end{aligned} \tag{9}$$

In the equation above, one term is interesting: $\ddot{d}$. Expanding it, the function μ appears and we'll be able to compute μ from the other terms. The following derivations assume that the pointing control constraint is not active, *i.e.* $d = \widehat{p_v}$. We find

$$\begin{aligned} \ddot{d} &= \frac{p_r \cdot p_v}{\|p_v\|^3} f(d, p_r) + \frac{1}{\|p_v\|}\frac{\mathrm{d}}{\mathrm{d}t}((p_r \cdot d)d - p_r) \\ &= \frac{p_r \cdot d}{\|p_v\|}\dot{d} + \frac{1}{\|p_v\|}\left((-\mu n \cdot d)d + (p_r \cdot \dot{d})d + (p_r \cdot d)\dot{d} + \mu n\right) \\ \ddot{d} &= \frac{1}{\|p_v\|}\left((p_r \cdot \dot{d})d + 2(p_r \cdot d)\dot{d} - \mu\left((d \cdot n)d - n\right)\right) \\ &= \frac{1}{\|p_v\|}\left((p_r \cdot \dot{d})d + 2(p_r \cdot d)\dot{d} - \mu f(d, n)\right). \end{aligned} \tag{10}$$

Introducing a new symbol Ω we get $h^{(4)} = \Omega - \frac{K\mu}{\|p_v\|} f(d, n) \cdot n = 0$, where Ω does not depend on μ; So $\mu = \frac{\Omega\|p_v\|}{Kf(d, n) \cdot n}$. When the pointing constraint is *not* active, substituting $\ddot{d}$ with (10), we get:

$$\begin{aligned} \Omega = {} & 3K\ddot{n} \cdot d \\ & + \dot{n} \cdot \left(3\left(3R^2 - \dot{R}\right)v + \left(4\dot{K} - 6RK\right)d + 4K\dot{d}\right) \\ & + n \cdot \left(\ddot{K}d + 2\dot{K}\dot{d} + \frac{K}{\|p_v\|}\left((p_r \cdot \dot{d})d + 2(p_r \cdot d)\dot{d}\right)\right). \end{aligned}$$

On a glide-slope constrained arc, we use the expression above for μ in order to integrate the dynamics of our system (2).

4 Direct and indirect methods

To explore the different structures that can arise, we want to obtain a great many solutions from random initial conditions. Given a random initial position and velocity, we first obtain a direct solution, or generate a new random initial condition if the problem is not feasible. We then use the direct solution to initialize the structure and parameters necessary for obtaining an indirect solution. If we find a zero of the shooting function, we then check (numerically) that the indirect solution satisfies all constraints in order to validate the structure extracted from the indirect solution.

4.1 The direct solution

We use JuMP and Ipopt within a Julia program. The problem is directly discretized into a non-linear program. We use the Crank-Nicolson implicit finite difference scheme for the dynamics constraints. When the time grid is uniform, we implement a basic multigrid resolution method, employing a low resolution solution as a warm startup for twice finer resolution. In our batch experiments, we start with a direction solution over 32 uniform time steps. The solution is analyzed (Section 4.2) to extract the various junction times. If necessary we refine the solution to 64, then 128 steps, then using up to two rounds of local (non uniform) refinement of the time grid by subdividing (4x) the time intervals where junctions have been found and their immediate neighbors. Whenever the time grid is refined, we re-analyze it and try to solve the indirect problem.

4.1.1 Scaling of variables.

We use input units of meters, meters-per-second and kilograms. To improve convergence in the Ipopt solver, we normalize separately the position (with $\mathcal{R}$), velocity (with $\mathcal{V}$) and mass (with $\mathcal{M}$). The dynamics become:

$$\begin{aligned}
\dot{r}(t) &= \frac{\mathcal{V}}{\mathcal{R}} v(t), \\
\dot{v}(t) &= \frac{1}{\mathcal{V}} \left(\frac{T}{\mathcal{M} m(t)} u(t) + g \right), \\
\dot{m}(t) &= \frac{-\alpha}{\mathcal{M}} T \tau(t).
\end{aligned}$$

Before analyzing the direct solution, we un-normalize the state and costate. It is trivial for the state variables. The costate variables must be divided by $\mathcal{R}$, $\mathcal{V}$ and $\mathcal{M}$ respectively. Since we want to ensure that $p_m(t_f) = 1$, we further scale all costate variables by the inverse of $p_m(t_f)$ which is close to but not exactly one in the direct solution. Currently, our implementation uses $\mathcal{R} = 2000\,m$, $\mathcal{V} = 100\,m/s$ and $\mathcal{M} = m(0) = 1905\,kg$. Clearly it would be further beneficial to tailor these scaling factors to each problem instance.

4.2 Analysis of the direct solution

The adjoint vector (p_r, p_v, p_m) is obtained as (the opposite of) the Lagrange multipliers of the dynamics constraints. It's value at the final time is obtained by linear extrapolation. Looking at the direct solution, we detect and gather junctions in a list, that we then sort and post-process before initializing the indirect solution parameters.

4.2.1 Detecting the activation of the pointing constraint.

We simply find the roots of the function $t \mapsto p_{v,z} - \cos\theta^\star \|p_v\|$ (linearly interpolated between samples). We add a `Pointing`$(t, \texttt{start})$ junction if the function is decreasing at the root t, and a `Pointing`$(t, \texttt{stop})$ junction otherwise. The value of the function at $t = 0$ determines whether the pointing constraint is active along the first arc of the trajectory. We have never seen more that two `Pointing` junctions.

4.2.2 Detecting the bang switches.

The throttle τ is either $\tau_{\min}$ or $\tau_{\max}$, depending on the sign of $\Psi(t) = \frac{1}{m(t)} p_v(t) \cdot d(t) - p_m(t)\alpha$. We can determine the switch times by either finding the roots of $\Psi(t)$ (linearly interpolated) or looking at the function $\tau(t)$. Both works reasonably well. We find that looking at $\tau(t)$ works a bit better even if the times that are obtained are less precise. At the initial time, $\tau(0)$ is not always "bang" in the direct solution, and we find that a decision threshold of $0.8\tau_{\min} + 0.2\tau_{\max}$ works well (*i.e.* the threshold is lower than the mean). For each detected switch time t, we add a `Switch`$(t, \texttt{toMin})$ or `Switch`$(t, \texttt{toMax})$ to the list of junctions.

4.2.3 Detecting the activation of the glide slope state constraint.

This is by far the most sensitive part of the analysis. The state constraint can be active along an arc, or only at an instantaneous `touch` junction. The latter happens most often.

Our initial method, briefly described here, proved unreliable. First we would threshold the function h (close to zero) to obtain `touch` junctions and `start`/`stop` junctions (that define a *constrained interval*). Then, at these junctions, the value of μ (recall that μ is the state constraint multiplier) would be estimated as the jump along the cone normal n, that is $-(p_r(t_{i+1}) - p_r(t_i)) \cdot n(t_i)$. The threshold value on h had to be tweaked on some trajectories. The estimated constrained intervals and jump values obtained in this way would often differ significantly from the actual structure of the optimal trajectory. This often led to large deviation of the value of p_r, which, by integration, since $\dot{p}_v = -p_r$ along many arcs, led to dramatic deviation of p_v which indicates the thrust direction...

Instead, we work on the dual p_c of the state constraint as computed in the direct solution. Note that by construction, $\forall i, p_c(t_i) \geq 0$. We only estimate the timing of the `touch`, `start` and `stop` junctions and use a trick in the indirect solution method that lets us avoid estimating the jump value (μ or j; we detail this later). We define a *glide structure* as a sequence of `touch` points and `start`/`stop` intervals for the glide-slope constraint activation. The analysis of p_c will result in a list of several possible glide structures. We test each in turn and keep the first glide structure for which a zero of the shooting function is found, whose state trajectory also numerically respects the state constraint.

Step A. If $\max_i p_c(t_i) \leq 10^{-7}$ we conclude that the state constraint is never active and return an empty list of glide structures. Otherwise, we normalize p_c so that $\max_i p_c(t_i) = 1$. The value 10^{-7} is of course dependent on the direct solver and its internal configuration (IpOpt in our case).

We expect to see higher values of p_c where the glide-slope constraint is active. We expect to see local maxima of p_c at `Glide touch`, `Glide start`, `Glide stop` junctions, and some high values between two consecutive `start` and `stop` junctions. Inside a constrained interval, p_c might have small oscillations that are not relevant.

Step B. We keep only the ordered list of local extrema of p_c, since the other values are never used. It is advantageous to consider that $p_c(t) = +\infty$ outside $[0, t_f]$ so that the initial and final times can be local minima but never local maxima. This is justified since:

- we assume that the trajectory starts strictly above the glide-slope cone and
- we do not need to account for the activation of the state constraint at the final time.

Step C. Small fluctuations of p_c are eliminated by filtering out small local minimum-maximum pairs. We use persistence-sensitive simplification.[1] Define the function Lil'bigger$(i, j) = p_c[i] < 1.3 \times p_c[j]$ that tests if one value is not much bigger that another. If (i,j) are the indices (in p_c) of a max-min persistence feature, then we delete the feature if

- Lil'bigger(i, j) (this removes small fluctuations), or
- if they are neighboring local extrema and Lil'bigger$(i, i-1) \wedge$ Lil'bigger$(i, i+1)$ ($p_c[i]$ is a very smooth local maximum).

The latter case appears when h has a local minimum high above the glide-slope cone. All remaining local maxima are considered important and indicative of the activation of the glide-slope constraint.

[1] `https://www.csc.kth.se/~weinkauf/notes/persistence1d.html`

Step D. We now have a sequence of alternating local maxima and minima. For clarity, we re-index then from 1 to $2N-1$ and add *virtual* maxima at indices 0 and $2N$. Even indices are local maxima; odd indices are local minima. We write $[i, i']$ for the group of maxima with indices in $[i, i']$. We start from the group of local maxima $[2, 2N]$ and use the minima to break the group into subgroups. This is akin to thresholding. The minima, indexed by j, are processed by increasing value of $p_c[j]$ (with a slight abuse of indexing). When processing the local minima at index j, we split the unique group G of local maxima with indices both below and above j into two groups: $\{i \in G \mid i < j\}$ and $\{i \in G \mid j < i\}$. For example, if $p_c[5] < p_c[7] < p_c[3]$, we obtain the sequence: $\{[2,8]\} \to \{[2,4],[6,8]\} \to \{[2,4],[6,6],[8,8]\} \to \{[2,2],[4,4],[6,6],[8,8]\}$. The four collections of groups correspond to four glide structures.

Step E. Whenever a group of maxima is split, the whole collection of groups is interpreted as a possible glide structure, and added to the list of glide structures to be tested.

- if a group is a singleton $[2i, 2i]$ with $1 \leq i < N$, we add a `Glide`$(t_{2i}, \texttt{touch})$ to the list of junctions,
- $G = [2N-2, 2N] \Rightarrow$ add a `Glide`$(t_{2N-2}, \texttt{touch})$,
- $G = [2l, 2N] \Rightarrow$ add a `Glide`$(t_{2l}, \texttt{start})$ and set that glide arc to last until the final time t_f,
- $G = [2l, 2r < 2N] \Rightarrow$ add a `Glide`$(t_{2l}, \texttt{start})$ and a `Glide`$(t_{2r}, \texttt{stop})$.

4.2.4 Gathering the junctions of all types.

When all the junctions are found and sorted by time, we verify the presence of a junction between a `Glide`$(t_1, j_1, \texttt{start})$ and a `Glide`$(t_2, j_2, \texttt{stop})$. If there is any, then both `Glide` junctions become `Glide`$(t_i, j_i, \texttt{touch})$ junctions, see Propositions 2 and 3.

4.3 Junctions data for the indirect solution.

To each junction, we associate

- its parameters,
- a procedure that is executed when the junction "happens", during the trajectory integration,
- a procedure that partially evaluates the shooting function, usually with respect to the junction's parameters.

The number of parameters equals the dimension of the partial evaluation. We regroup the junctions' parameters to the left of the arrow and the partial evaluations to the right in order to form the complete shooting function $\mathfrak{F} : \mathbb{R}^n \to \mathbb{R}^n$, of which we seek a root.

Recall that $\dot{p}_r(t) = -\mu(t)n(t)$. The value of μ is non-negative along glide-slope constrained arcs and zero outside. Following [6], we assume that p_r can undergo a discontinuous change only

- at "touch points" for the glide-slope constraint, and
- at the endpoints of arcs of positive length along which the glide-slope constraint is active.

With most initial conditions, we have no `Glide` arc, but just zero or more `Glide touch` junctions. In that case, p_v is piecewise linear, with slope $-p_r$. The obvious shooting function would use, as parameters, p_r and its discontinuities at the `Glide touch` junctions. But small deviations in the value of p_r get integrated into large deviation in p_v which indicate the thrust's direction. We have experienced that this is one main reason for the indirect method to fail to find a physically feasible solution (some solution arcs go back in time). Instead, we use as parameters the corners of the piecewise linear p_v, *i.e.* the values of p_v at times $0 = t_g^0, t_g^1, \ldots, t_g^k, t_g^{k+1} = t_f$ where we have $k \geq 0$ `Glide touch` junctions. Along a `Glide` arc, we observe that p_v deviates slightly from linearity; see below.

The start junction is unique and happens at the beginning of the trajectory. Its 7 real parameters comprise $p_v(0)$, $p_v(t_g^1)$ and $p_m(0)$. Upon "happening" at time 0, the costates are copied from the junction parameters into the integrator state. The costate $p_r(0)$ is computed as $(p_v(0) - p_v(t_g^1))/t_g^1$ where t_g^1 is the time parameter of the next [`Glide` or end] junction in the sequence of junctions. The partial evaluation of the shooting function is $\mathfrak{F}_0() = (r(t_f), v(t_f), p_m(t_f) - 1) \in \mathbb{R}^7$.

The end junction is unique as well. Its only parameter is t_f. The partial evaluation of the end junction is $\mathfrak{F}_f() = H(t_f) \in \mathbb{R}$, *i.e.* the hamiltonian should be zero.

A `Pointing` *junction* exists when the pointing constraint changes status. Beside its type (`start` or `stop`), the junction has one parameter for the shooting function: its time t_p^i. Upon "happening", the integrator ODE is updated. The partial evaluation of the shooting function is $\mathfrak{F}_p^i() = p_{v,z}(t_p^i) - \cos\theta^\star \|p_v(t_p^i)\| \in \mathbb{R}$.

A `Switch` *junction* exists when the thrust norm changes. Beside its type (`toMin` or `toMax`), the junction has one parameter for the shooting function: its time t_s^i ($i \in \{1, 2\}$). Upon "happening", the integrator ODE is updated. The partial evaluation of the shooting function is $\mathfrak{F}_s^i() = \Psi(t_s^i) \in \mathbb{R}$ (see Equation (3)).

A `Glide touch` *junction* exists when the trajectory bounces on the glide-slope cone. Beside its type, `touch`, it has 4 real parameters: its time t_g^i and the value $p_v(t_g^{i+1}) \in \mathbb{R}^3$ of p_v at the time of the next [`Glide` or end] junction. (Recall that $1 \leq i \leq k$ and $t_g^{k+1} = t_f$.) Upon "happening", the integrator ODE is updated and the discontinuity to p_r is applied as $p_r(t_g^{i+}) = \frac{p_v(t_g^i) - p_v(t_g^{i+1})}{t_g^{i+1} - t_g^i}$ where

- the value of $p_v(t_g^i)$ is read from the integrator's state,
- the values of t_g^i and $p_v(t_g^{i+1})$ are the junction's parameters and
- the value of t_g^{i+1} is taken from the next [`Glide` or end] junction's parameters.

To the $4 = 1 + 3$ numerical parameters of the junction correspond a partial evaluation of the shooting function $\mathfrak{F}_g^i() \in \mathbb{R}^4$. At the junction, we must make sure that the zeroth and first derivative of h are zero, *i.e.* that the position lies on the cone and the velocity is tangent to it. We must also verify that the discontinuity in p_r is along the cone's normal: Writing $t = t_g^i$ for short, we get $\mathfrak{F}_g^i() = (r(t) \cdot n(t), v(t) \cdot n(t), \overline{\Delta p_r \times n(t)}) \in \mathbb{R}^4$, where $\Delta p_r = (p_t(t^+) - p_r(t^-))$. Note that, since $n_z(t) = 1 \neq 0$, $\overline{\Delta p_r \times n(t)} = 0 \Rightarrow \Delta p_r \times n(t) = 0$.

A `Glide start` *junction* exists when the trajectory starts to glide on the glide-slope cone, instead of merely bouncing on it. From a programming point of view, it is treated in exactly the same way as a `Glide touch` junction. But we interpret its second parameter differently. Beside its type, `start`, it has 4 real parameters: its time t_g^i and the "virtual" value p'_v which the $p_v(\cdot)$ curve should "aim" at time t_g^i, that is, (t_g^i, p'_v) will not be a corner of the integrated $p_v(\cdot)$ curve, but serves to define the right derivative of $p_v(\cdot)$ at t_g^{i+}, just as for a `Glide touch` junction: $p_r(t_g^{i+}) = \frac{p_v(t_g^i) - p'_v}{t_g^{i+1} - t_g^i}$. The partial evaluation of the shooting function is the same as that of a `Glide touch` junction.

A `Glide stop` *junction* exists when the trajectory stops gliding on the glide-slope cone. Beside its type, `stop`, it has 4 real parameters: its time t_g^i and the value $p_v(t_g^{i+1}) \in \mathbb{R}^3$ of p_v at the time of the next [`Glide` or end] junction. Upon "happening", the integrator ODE is updated and the discontinuity to p_r is applied just as above. The previous `Glide start` junction's partial evaluation of the shooting function made sure that h and $\dot{h}$ are zero upon entering the glide, at time t_g^{i-1}. For the `Glide stop` junction, the partial evaluation of the shooting function makes sure that $\ddot{h}(t_g^{i-1})$ and $\dddot{h}(t_g^{i-1})$ are zero as well, and verifies that the discontinuity in p_r, at time t_g^i, is along the cone's normal, making it 4-dimensional, as required. Note that the multiplier μ, whose expression we know (and use!), guarantees that all the derivatives $h^{(j)}, j \geq 4$ are zero at all times along a glide arc. We conclude that h stays zero along a glide arc, as required.

The special case of a glide-til-the-end. Sometimes, the vehicle fancies a glide on the cone until the very end of the trajectory. Consider the corresponding `Glide stop` and end junctions. All the parameters of the `Glide stop` junctions are useless: its time parameter t_g^k coincides with t_f. And the parameter $p_v(t_g^{k+1})$ is not needed because the integration stops. But, we still need to ensure that a part of the partial shooting function of the `Glide stop` junction evaluates to zero. Since the jump in p_r is not effected, we don't need to test that the discontinuity is along the cone's normal. It remains to test that the second and third derivatives of h at t_g^{k-1} are zero upon entering the "glide-til-the-end" final arc. We write these two numbers in the place of the values $(r_x(t_f), r_y(t_f))$ from the partial shooting function $\mathfrak{F}_0$ associated with the start junction. Indeed, during the last arc, the vehicle is gliding on the cone, therefore $r_z(t_f) = 0 \Rightarrow r_x(t_f) = r_y(t_f) = 0$, and the latter two values are redundant. We end up, as required, with a shooting function $\mathfrak{F} : \mathbb{R}^n \to \mathbb{R}^n$ for some $n \geq 12$.

4.4 Testing a candidate structure

A candidate structure gives rise to a fixed shooting function $\mathfrak{F}$, which is fed to a standard root finder: We call CMINPACK::fsolve(), which then provides an "optimized" trajectory, in the form of values for the junctions' parameters, that is (should be) a root of $\mathfrak{F}$. We gather the following Boolean information:

- `root_found`: Did fsolve() find a root? (this boolean comes from fsolve()'s returned data).
- `root_almost_found`: If not, has it almost found a root? which we define as the value of the shooting function having decreased by at least 10^4. We also define

$$\texttt{is\_root} \equiv \texttt{root\_found} \quad \text{OR} \quad \texttt{root\_almost\_found}.$$

- `orderly_times`: Are the junctions still ordered correctly in time?
- `similar_trajectories`: Are the direct and indirect trajectories similar? The answer is yes when the distance between both trajectories is less than 81 meters when sampled at 20 random times. The value 81 is arbitrary, but found adequate to accommodate for the low number of nodes in the most direct solutions (32 steps, 33 nodes).
- `glide_slope_ok`: Is the glide-slope constraint respected? Given a threshold $h' = -10^{-4}\,\mathrm{m}$, we consider that the constraint is not respected if at least two consecutive samples are below the threshold.
- `on_target`: How well does the indirect trajectory reach its target? It is equal to $\|r(t_f)\|^2 + \|v(t_f)\|^2 < 10^{-4}$, which implies that $r(t_f)$ is less that 1 cm from the target and $v(t_f)$ is less than 1 cm/s. Although not satisfying numerically, we consider that is is satisfying "in the real life."

Define `Acceptable` $\equiv$ `orderly_times` AND `similar_trajectories` AND `glide_slope_ok` AND `on_target` AND `is_root`.

If `Acceptable` is true, then the trajectory is accepted as `All_good` if `root_found` and as `Almost_good` otherwise. If `Acceptable` is false, repairs are attempted. If these fail, the root search is deemed a failure and another candidate structure is tested, or the failure is reported if there is no other candidate structure.

4.4.1 Repairs.

We decribe two ways to try to "repair" the structure, that we found quite effective.

The failure of a candidate structure is often due to a missing (non detected) glide-slope constraint activation at the end of the trajectory. Assume that (i) `Acceptable` fails only due to `glide_slope_ok` and (ii) the last junction is not a `Glide` junction. Let t be the time of the last `Glide` junction if it exists, or $t = 0$. We then check if the glide-slope state constraint is violated along a time interval $[t_v, t_f)$ with $t < t_v$. If that is the case, we add `Glide touch` junction at time $t_t = 0.7t_v + 0.3t_f$ and call fsolve(). If the resulting trajectory still violates the state constraint and (t_t, t_f) is free of junction, we replace the new junction with a `Glide start` (so that the vehicle "glides-til-the-end") and try again.

Assume that `orderly_times` and `similar_trajectories` are both true and that there is no `Glide` junction. Assume that the trajectory does not reach its target, *i.e.* both `on_target` and `is_root` are false. In this case, the only freedom of the trajectory comes from the initial adjoint at $t = 0$. We believe/conjecture that, in this case, there might not be enough floating-point precision to ensure that the integration of the whole trajectory can reach the target position and velocity. We attempt to remedy the situation by adding a dummy junction at $t = (t_f - t_0)/2$ that forces some state/adjoint coordinates at t^+ while the shooting function seeks to minimize the difference of these coordinates between t^+ and t^-. This does not always works, but does help sometimes. A deeper study of this phenomenon is clearly necessary. In the experiment below, this dummy junction operates only on the 3 position state coordinates.

5 Numerical experiments

We use the martian lander described in [1]. Since we are only interested in the structure of the optimal trajectory, we do not set a maximal fuel consumption. Still, our initial conditions are chosen reasonably so that the lander never uses, say, more than one metric ton of fuel.

The initial wet mass is $m_0 = 1905\,kg$. The maximal thrust is $T = 16572.72\,N$. Throttle range is given by $\tau_{\min} = 0.3$ and $\tau_{\max} = 0.8$. The parameter $\alpha \approx 5.0863 \times 10^{-4}\,s/m$ is computed as $\alpha = \frac{1}{g_e I_{\text{sp}} \cos\Psi}$ with $g_e = 9.807\,m/s^2$, $I_{\text{sp}} = 225\,s$ and $\Psi = 27\,\text{deg}$. The pointing constraint parameter is $\theta^\star = 60\,\text{deg}$. The glide-slope angle is $\gamma = 10\,\text{deg}$ and martian gravity is $g_0 = 3.7114\,m/s^2$.

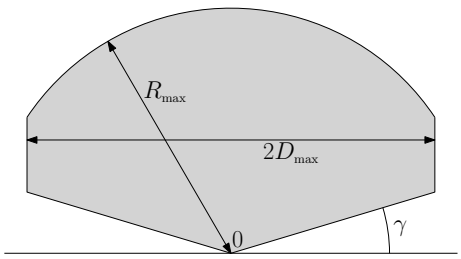

Figure 2: The initial position r_0 is sampled in the gray volume.

The landing target is set at the origin. The initial position r_0 is generated within a maximal range and a maximal (3D) distance to the target. The sampling of r_0 is rotationally symmetric around the vertical axis above the target. We uniformly sample $w \in [0, D_{\max}]$: the horizontal distance (range) of r_0 to the origin ($w = \|\overline{r_0}\|$). Then the initial height $r_{0,z}$ is uniformly sampled in $[w\tan\gamma, \sqrt{R_{\max}^2 - w^2}]$. A choice of $R_{\max} \geq \frac{D_{\max}}{\cos\gamma}$ guarantees that this interval is nonempty. In our implementation we choose $D_{\max} = 2000\,\text{m}$ and $R_{\max} = 1.5 \times \frac{D_{\max}}{\cos\gamma}$. For the initial velocity, we uniformly choose $v_{0,z}$ in $[-100, 25]\,\text{m/s}$ and $\overline{v_0}$ in $[-100, 100]^2\,\text{m/s}$.

We perform a simple check that detects (and rejects) some unfeasible initial state: We consider a 1D motion along the line with direction $n(r_0)$ and test that the maximal thrust along the normal to the glide-slope cone is enough to avoid crashing into the cone. We write $v_n = v_0 \cdot \widehat{n}(r_0)$, $r_n = r_0 \cdot \widehat{n}(r_0)$ and $a = \frac{T\tau_{\max}}{m_0} - g_0\cos\gamma$. The test fails if $v_n^2 > 2ar_n$. A new initial state is sampled as long as the current one fails the sanity check.

We then try to obtain a direct solution (Section 4.1). If this fails, we consider the problem as unfeasible and sample new initial state. Otherwise, we proceed to analyze the direct solution (Section 4.2) and try to find an indirect solution (Section 4.3).

5.1 Statistics

We analyze the trajectories obtained from 1100000 random initial conditions. For 32 of these, our software failed to find a satisfying structure. Statistics for the 1099968 successes are show in Tables 1 and 2. Failures are informally discussed in§5.1.2.

5.1.1 Statistics for the successes.

We obtained 56 different trajectory structures. Table 1 shows the 13 most frequent structures. Together, they account for 98.431 % of the Glide arcs of positive length appear in 0.750 % of all trajectories.

5.1.2 A look at the failures

We describe the 39 failures obtained in the batch of 1100000 random initial conditions. We believe that 31 of them actually have no feasible solution. And we are able to find a solution for the remaining 8 ones, at the cost of a slightly longer computation time:

- 28 of them probably have no solution : the direct and indirect solution vary substantially, The lander is initially moving away from the target. The direct solution shows a `Glide touch` junction before turning back towards the landing target. The indirect solution shows a similar trajectory but "crashes" into the glide-slope cone, *i.e.* the state constraint can not be satisfied at the contact with the cone. The best indirect solutions found all have structure [M Mθ V Mθ mθ m M]. We conjecture that these 28 initial conditions have no feasible solutions but are very close to conditions with a feasible solution, and the direct solution have enough numerical freedom to find a false solution. (At higher time resolution, IpOpt fails to find a solution.)
- 3 of them behave just as the 28 above, but with a different structure, namely [M V M Mθ V Mθ M m M]. It seems that the crash in the cone at the first `Glide touch` is inevitable, although at most

Count	Structure	Freq. (‰)
483581	Mθ mθ m M	439.632
246409	Mθ M m M	224.015
86573	mθ m M	78.705
58440	M m M	53.129
55009	M $\star$ M m M $\star$ M	50.010
45907	M $\star$ M m M	41.735
31537	M m M $\star$ M	28.671
25648	m M	23.317
19841	M Mθ $\star$ Mθ M m M	18.038
9772	M	8.884
8855	Mθ M	8.050
6472	Mθ M m M $\star$ M	5.884
4663	M $\star$ M m M M$\boldsymbol{\gamma}$	4.239

'M' stands for a bang-max arc ($\tau = \tau_{\max}$), 'm' means bang-min. We adjoin $\boldsymbol{\gamma}$ when the arc is state-constrained and θ when it is pointing-constrained. The star $\star$ stands for a `Glide touch` junction.

Table 1: The 13 most frequent "successfully" recovered structures.

Count	Structure	Freq. (‰)
920997		837.294
89810	$\star$	81.648
56886	$\star$ $\star$	51.716
23606	$\star_\theta$	21.461
4749	$\star$ $\overline{\boldsymbol{\gamma}}$	4.317
3251	$\overline{\boldsymbol{\gamma}}$	2.956
410	$\star$ $\star_\theta$	0.373
108	$\boldsymbol{\gamma}$ $\star$	0.098
44	$\boldsymbol{\gamma}$	0.040
40	$\boldsymbol{\gamma}$ $\overline{\boldsymbol{\gamma}}$	0.036
32	$\boldsymbol{\gamma}$ $\star_\theta$	0.029
10	$\star$ $\boldsymbol{\gamma}$ $\star$	0.009
8	$\star$ $\star$ $\star_\theta$	0.007
6	$\star$ $\boldsymbol{\gamma}$ $\star_\theta$	0.005
3	$\star$ $\boldsymbol{\gamma}$	0.003
3	$\star$ $\star$ $\overline{\boldsymbol{\gamma}}$	0.003
3	$\star$ $\star$ $\star$	0.003
1	$\star$ $\star$ $\boldsymbol{\gamma}$	0.001
1	$\star_\theta$ $\star$	0.001

A. Statistics on the glide-slope state constraint activation. The symbol $\overline{\boldsymbol{\gamma}}$ represents a state-constrained arc that lasts until the final time t_f.

Count	Structure	Freq. (‰)
837042	θ ...	760.969
238838	...	217.132
24045	... $\theta_\star$ ...	21.860
23	... θ	0.021
18	$\theta_\star$ ...	0.016
2	θ θ	0.002

B. Statistics on the pointing control constraint activation. The ellipsis represents some arc(s) of positive length that are not pointing-constrained.

Count	Structure	Freq. (‰)
961400	M m M	874.025
113666	m M	103.336
24902	M	22.639

C. Statistics on the bang structure.

Table 2: Statistics on successes.

0.8 mm deep, hence the absence of an indirect solution. Another less likely reason might be that the first `Glide touch` (V) should in fact be a positive-duration glide arc, which is not detected due to a too coarse time resolution.

– For 7 of them, the best structure found is very simple: [mθ m M]. The trajectory comes close to hit the target position and velocity, but not enough for our stringent threshold. For all 7, an `All_good` solution is found, with the same structure, if we increase the number of integration steps per arc from 50 to 150.

– For last initial condition, the best structure found is [M m M V M]. It fails only due to a dip of 0.3 mm below the glide-slope cone. Like the 7 above, we are able to find an `All_good` solution by increasing the number of integration steps per arc from 50 to 150.

6 Analysis of glide-slope arcs

In this section, we look at the possible interactions between the glide-slope (state) constraint and the pointing (control) constraint. We argue that not much can happen while the lander is sliding along the glide-slope cone.

Proposition 1. *If $\theta^\star < \pi/2$. If the pointing constraint is activated at a time $t > 0$, then a* `Glide touch` *is necessary in order to deactivate the pointing constraint. See Section 2.*

Proposition 2. *Let $I = (a, b), a < b$, be a time interval along which the glide-slope constraint is active. Then there is no* `Switch` *junction in I.*

Proof. At any time in I we have $\ddot{h} = Kn \cdot d - G(\boldsymbol{x}) = 0$ with $K > 0$ and $G(\boldsymbol{x}) > 0$. Hence, $n \cdot d > 0$ and any `Switch` junction triggers a discontinuity of K and a non-zero discontinuity of $\ddot{h}$. This contradicts the fact that $\ddot{h}$ must stay zero in I. □

Proposition 3. *Assume that $\gamma \in [0, \pi/2)$ and $\theta^\star \in (\gamma, \pi - \gamma)$. Assume that $\|p_v\| > 0$. Let $I = (a, b), a < b$, be a time interval along which the glide-slope constraint is active. If there is a* `Pointing start` *junction at time $t^\star \in I$, then we must have $\dot{\theta}(t^\star) = \ddot{\theta}(t^\star) = 0$.*

Proof of Proposition 3. We consider the third derivative of h and prove that if the pointing constraint becomes active at time $t^\star \in I$, then, unless $\dot{\theta}(t^\star) = \ddot{\theta}(t^\star) = 0$, $\dddot{h}$ undergoes a non-zero discontinuity that contradicts the existence of the interval I, along which $\dddot{h}$ must stay zero. It is convenient to work in cylindrical and spherical coordinates. The position is parameterized as

$$r = \begin{pmatrix} \rho\, e^{i\varphi} \\ z \end{pmatrix} = \begin{pmatrix} \rho\cos\varphi \\ \rho\sin\varphi \\ z \end{pmatrix} \text{ or } x + iy = \rho e^{i\varphi}. \tag{11}$$

Differentiating (11) yields

$$v = \begin{pmatrix} (\dot{\rho} + i\,\rho\,\dot{\varphi})\, e^{i\varphi} \\ \dot{z} \end{pmatrix} \text{ and } \dot{v} = \begin{pmatrix} \left(\ddot{\rho} - \rho\dot{\varphi}^2 + i\,(\rho\,\ddot{\varphi} + 2\,\dot{\rho}\,\dot{\varphi})\right)\, e^{i\varphi} \\ \ddot{z} \end{pmatrix}. \tag{12}$$

We also define spherical coordinate angles for p_v in the classical orthonormal basis $(e_\sigma, e_\theta, e_\psi)$:

$$e_\sigma = \begin{pmatrix} \sin\theta\, e^{i\psi} \\ \cos\theta \end{pmatrix}, \qquad e_\theta = \begin{pmatrix} \cos\theta\, e^{i\psi} \\ -\sin\theta \end{pmatrix}, \qquad e_\psi = \begin{pmatrix} i\, e^{i\psi} \\ 0 \end{pmatrix} \tag{13}$$

$$\dot{e}_\sigma = \dot{\theta} e_\theta + \dot{\psi}\sin\theta e_\psi, \qquad \dot{e}_\theta = -\dot{\theta} e_\sigma + \dot{\psi}\cos\theta e_\psi, \qquad \dot{e}_\psi = -\dot{\psi}\left(\sin\theta\, e_\sigma + \cos\theta\, e_\theta\right). \tag{14}$$

Using $p_r = -\dot{p}_v$, we find

$$p_v = \sigma e_\sigma \quad \text{and} \quad p_r = -\dot{\sigma} e_\sigma - \sigma\dot{\theta} e_\theta - \sigma\dot{\psi}\sin\theta e_\psi. \tag{15}$$

The direction of the control is given by $d = \begin{pmatrix} \sin\breve{\theta}\, e^{i\psi} \\ \cos\breve{\theta} \end{pmatrix}$ with

$$\breve{\theta} = \begin{cases} \theta & \text{if } 0 \le \theta \le \theta^* \text{ (inactive pointing constraint)} \\ \theta^* & \text{if } \theta^* \le \theta < \pi \text{ (active pointing constraint)} \end{cases} \tag{16}$$

The dynamics give us $\dot{v} = K\, d + \begin{pmatrix} 0 \\ 0 \\ -g_0 \end{pmatrix}$. Using (12), this yields

$$\begin{pmatrix} \left(\ddot{\rho} - \rho\dot{\varphi}^2 + i\,(\rho\,\ddot{\varphi} + 2\,\dot{\rho}\,\dot{\varphi})\right)\, e^{i\varphi} \\ \ddot{z} \end{pmatrix} = K \begin{pmatrix} \sin\breve{\theta}\, e^{i\psi} \\ \cos\breve{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -g_0 \end{pmatrix} :$$

$$\begin{aligned} \ddot{\rho} &= \rho\,\dot{\varphi}^2 + K\, \sin\breve{\theta}\cos(\psi - \varphi) \\ \rho\,\ddot{\varphi} &= -2\,\dot{\rho}\,\dot{\varphi} + K\, \sin\breve{\theta}\sin(\psi - \varphi) \\ \ddot{z} &= K\cos\breve{\theta} - g_0 \end{aligned} \tag{17}$$

Now let us compute derivatives of h :

$$\begin{aligned} h &= z - \rho\tan\gamma = 0\,, \\ \ddot{h} &= \ddot{z} - \ddot{\rho}\tan\gamma = 0\,. \end{aligned} \tag{18}$$

Substituting the dynamics (17) in $\ddot{z} - \ddot{\rho}\tan\gamma$ yields

$$\ddot{h} = K\cos\breve{\theta} - g_0 - \Big(\rho\,\dot{\varphi}^2 + K\, \sin\breve{\theta}\cos(\psi - \varphi)\Big)\tan\gamma = 0. \tag{19}$$

Differentiating once more, one gets (the first line uses $\dot{K} = \alpha\, K^2$ and the second line comes from differentiating $\rho\dot{\varphi}^2$, and substituting the dynamics again):

$$\begin{aligned} \dddot{h} &= \alpha\, K\,\left(g_0 + \rho\,\dot{\varphi}^2\tan\gamma\right) \\ &\quad + \Big(3\,\dot{\rho}\,\dot{\varphi}^2 + K\, \sin\breve{\theta}(\dot{\psi} - 3\,\dot{\varphi})\, \sin(\psi - \varphi)\Big)\tan\gamma \\ &\quad - K\left(\sin\breve{\theta} + \cos\breve{\theta}\cos(\psi - \varphi)\tan\gamma\right)\frac{\mathrm{d}}{\mathrm{d}t}\breve{\theta} = 0 \end{aligned} \tag{20}$$

where (see (16))

$$\frac{\mathrm{d}}{\mathrm{d}t}\breve{\theta} = \begin{cases} \dot{\theta} & \text{if } 0 \le \theta < \theta^* \text{ (inactive pointing constraint)} \\ 0 & \text{if } \theta^* < \theta < \pi \text{ (active pointing constraint)} \end{cases} \tag{21}$$

Define $A(\theta, \phi, \psi) \equiv \sin\theta + \cos\theta\tan\gamma\cos(\psi - \varphi)$. We see that everything in $\dddot{h}$ is continuous with respect to time, except the last term, that has, at $\theta = \theta^*$, a jump with size $KA(\theta^\star, \phi, \psi)\dot{\theta}$; hence $\ddot{h}\,(t)$ can be identically zero in a neighborhood of a time such that θ is smaller that θ^* before $t^\star$ and larger than θ^* after $t^\star$ only if

$$KA(\theta^\star, \phi, \psi)\dot{\theta} = 0. \tag{22}$$

On the one hand, $K > 0$. On the other hand, our assumption that $[\gamma \in [0, \pi/2)$ and $\theta^\star \in (\gamma, \pi - \gamma)]$ implies $A(\theta^\star, \phi, \psi) > 0$. So, if the pointing constraint is activated, we must have $\dot{\theta}(t^\star) = 0$. Assume now that $\ddot{\theta}(t^\star) \neq 0$. Since $\theta(t) < \theta^\star$ for $t < t^\star$, $\ddot{\theta}(t^\star)$ must be negative and the pointing constraint can not be activated at time $t^\star$. So, if the pointing constraint is activated, we must have $\ddot{\theta}(t^\star) = 0$. □

Remark 5. *Proposition 3 suggests that it is very unlikely that the pointing constraint gets activated inside a state-constrained time interval. We have, however, never seen, experimentally, both constraints active at the same time along an interval of positive length; we conjecture that this is not possible. Another way to have both constraints active would be to activate the glide-slope constraint while the pointing control constraint is already active. For completeness, we give in Section 6.1 the expression for $\mu(t)$, the glide-slope constraint's multiplier, in that case. Note that an active pointing constraint is relevant only along positive time intervals, so that we need not consider a "*`Pointing touch`*" junction inside a glide-slope constrained arc.*

Remark 6. *If the glide-slope constraint is active, $n \cdot d > 0$ holds. It implies that $\theta < \pi/2 + \gamma$ (make a drawing to see it). Also, $\gamma \in [0, \pi/4] \Rightarrow \pi/2 + \gamma \leq \pi - \gamma$. So, in Proposition 3, we can assume either*

$$\gamma \in [0, \pi/2) \text{ and } \theta^\star \in (\gamma, \pi - \gamma) \tag{23}$$

$$\text{or} \quad \gamma \in [0, \pi/4) \text{ and } \theta^\star \in (\gamma, \pi]. \tag{24}$$

6.1 The state constraint multiplier when the pointing constraint is active

For completeness, we derive here an expression for μ when both constraints are active. The pointing constraint requires that $d_z \geq \cos\theta^\star$. The maximum principle implies that if $\overline{p_v}$ is not zero and $p_{v,z} < \|\overline{p_v}\| \cos\theta^\star$ then $d = \left(\sin(\theta^\star)\widehat{\overline{p_v}}, \cos\theta^\star\right)$ with $\widehat{\overline{p_v}} \equiv \frac{\overline{p_v}}{\|\overline{p_v}\|}$; d is basically clamped to the north-pole spherical cap of half-angle $\theta^\star$. Using $\dot{d} = \sin\theta^\star \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\overline{p_v}}$, we write

$$\dot{d} = \frac{\sin\theta^\star}{\|\overline{p_v}\|}\left(\dot{\overline{p_v}} - \frac{\overline{p_v}\cdot\dot{\overline{p_v}}}{\|\overline{p_v}\|^2}\overline{p_v}\right) = \frac{\sin\theta^\star}{\|\overline{p_v}\|}\left(\frac{\overline{p_v}\cdot\overline{p_r}}{\|\overline{p_v}\|^2}\overline{p_v} - \overline{p_r}\right) = \frac{\sin\theta^\star}{\|\overline{p_v}\|} f(\widehat{\overline{p_v}}, \overline{p_r}), \tag{25}$$

$$\begin{aligned}
\ddot{d} &= \frac{\overline{p_v}\cdot\overline{p_r}\sin\theta^\star}{\|\overline{p_v}\|^3} f(\widehat{\overline{p_v}}, \overline{p_r}) + \frac{\sin\theta^\star}{\|\overline{p_v}\|}\frac{\mathrm{d}}{\mathrm{d}t}\left(\left(\widehat{\overline{p_v}}\cdot\overline{p_r}\right)\widehat{\overline{p_v}} - \overline{p_r}\right) \\
&= \frac{\overline{p_v}\cdot\overline{p_r}}{\|\overline{p_v}\|^2}\dot{d} + \frac{\sin\theta^\star}{\|\overline{p_v}\|}\left(\frac{\dot{d}\cdot\overline{p_r}}{\sin\theta^\star}\widehat{\overline{p_v}} - \mu\left(\widehat{\overline{p_v}}\cdot n\right)\widehat{\overline{p_v}} + \left(\widehat{\overline{p_v}}\cdot\overline{p_r}\right)\frac{\dot{d}}{\sin\theta^\star} + \mu\overline{n}\right) \\
&= \frac{\widehat{\overline{p_v}}\cdot\overline{p_r}}{\|\overline{p_v}\|}\dot{d} + \frac{1}{\|\overline{p_v}\|}\left(\left(\dot{d}\cdot\overline{p_r}\right)\widehat{\overline{p_v}} + \left(\widehat{\overline{p_v}}\cdot\overline{p_r}\right)\dot{d}\right) - \frac{\mu\sin\theta^\star}{\|\overline{p_v}\|} f(\widehat{\overline{p_v}}, n) \\
&= \frac{1}{\|\overline{p_v}\|}\left(\left(\dot{d}\cdot\overline{p_r}\right)\widehat{\overline{p_v}} + 2\left(\widehat{\overline{p_v}}\cdot\overline{p_r}\right)\dot{d}\right) - \frac{\mu\sin\theta^\star}{\|\overline{p_v}\|} f(\widehat{\overline{p_v}}, n).
\end{aligned} \tag{26}$$

For computing μ along a pointing-constrained interval (that is, when the *glide-slope* constraint is also active), we simply substitute Equations (25) and (26) for $\dot{d}$ and $\ddot{d}$—instead of Equations (6) and (10)—into the relevant formulas.

References

[1] B. Açıkmeşe 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.

[2] P. Lu, "Propellant-optimal powered descent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 4, pp. 813–826, 2017.

[3] F. Spada, M. Sagliano, and F. Topputo, "Direct–indirect hybrid strategy for optimal powered descent and landing," *Journal of Spacecraft and Rockets*, vol. 60, no. 6, pp. 1787–1804, 2023. DOI: 10.2514/1.A35650. [Online]. Available: https://doi.org/10.2514/1.A35650.

[4] P. Lu and C. Davami, "Rethinking propellant-optimal powered descent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 10, pp. 2016–2028, 2024.

[5] C. Leparoux, B. Hérissé, and F. Jean, "Structure of optimal control for planetary landing with control and state constraints," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 28, 2022. DOI: 10.1051/cocv/2022065. [Online]. Available: https://ensta-paris.hal.science/hal-03642339.

[6] A. E. Bryson, Jr. and Y.-C. Ho, *Applied Optimal Control, Revised Printing*. John Wiley & Sons, 1975.

A The auxiliary function f

For any vectors a and b, define $\boxed{f(a,b) \equiv (a\cdot b)a - \|a\|^2 b}$. Using $f(a,b) = -\|a\|^2\left(b - (\hat{a}\cdot b)\hat{a}\right)$, we see that f projects b onto the plane orthogonal to a, then scale it by $-\|a\|^2$. The function f simplifies some

calculus, as follows:

$$f(a,a) = 0, \tag{27}$$
$$f(a,b)\cdot c = f(a,c)\cdot b \quad \Rightarrow \quad f(a,b)\cdot a = 0, \tag{28}$$
$$f(a,b)\cdot b = f(b,a)\cdot a \leq 0. \tag{29}$$

$$\text{For } \lambda \in \mathbb{R}, f(a,b+\lambda c) = f(a,b) + \lambda f(a,c),$$
$$f(\lambda a, b) = \lambda^2 f(a,b).$$

We define $\boxed{j(a,b) = \dfrac{\mathrm{d}}{\mathrm{d}t} f(a,b)}$. Then,

$$\begin{aligned}
j(a,b) &= f(a,\dot{b}) + (\dot{a}\cdot b)a + (a\cdot b)\dot{a} - 2(a\cdot\dot{a})b, \\
j(a,b)\cdot a &= -f(a,\dot{a})\cdot b && = -f(a,b)\cdot\dot{a}, \\
j(a,b)\cdot b &= f(a,b)\cdot\dot{b} + 2f(b,a)\cdot\dot{a}, \\
j(a,b)\cdot\dot{a} &= f(a,\dot{b})\cdot\dot{a} - f(\dot{a},a)\cdot b && = f(a,\dot{a})\cdot\dot{b} - f(\dot{a},a)\cdot b, \qquad (30) \\
j(a,\dot{a}) &= f(a,\ddot{a}) - f(\dot{a},a), \qquad (31) \\
j(a,\dot{a})\cdot a &= -f(\dot{a},a)\cdot a && = -f(a,\dot{a})\cdot\dot{a} \quad \geq 0, \\
j(a,\dot{a})\cdot\dot{a} &= f(a,\ddot{a})\cdot\dot{a} && = f(a,\dot{a})\cdot\ddot{a}, \\
j(\dot{a},a) &= (a\cdot\dot{a})\ddot{a} - 2(\dot{a}\cdot\ddot{a})a + (\ddot{a}\cdot a)\dot{a}, \\
j(\dot{a},a)\cdot\dot{a} &= -f(\dot{a},a)\cdot\ddot{a}. \qquad (32)
\end{aligned}$$