# Sensitivity Analysis for Powered Descent Guidance: Overcoming degeneracy

Hubert Ménou and Eric Bourgeois and Nicolas Petit

***Abstract*— This paper exposes a method to handle inaccurate modeling and/or initial state errors during the Powered Descent Guidance (PDG), a critical phase of atmospheric rocket landing. For this, we develop a replanification method having reliable online computational capabilities. From a reference descent scenario, an optimal correction problem is formulated. After revisiting results on Non Linear Programming sensitivity for degenerate optimization problems, we conclude that Quadratic Programming (QP) provides a local solution to the replanification problem. Using three illustrative PDG scenarios, we stress degeneracy and show how QP is used to evaluate the upper Dini derivatives at stake. Further, we discuss to what extent QP also provides a quantitatively reasonable solution outside a small neighboorhood of the reference scenarios.**

## I. Introduction

Due to its critical nature, Powered Descent Guidance (PDG), the final phase of atmospheric rocket landing, is often addressed as an Optimal Control Problem (OCP) having many parameters such as initial states, aerodynamic coefficients and engine efficiency [1], [2]. No known analytic solution is available for the general optimal atmospheric PDG problem. Generating a numerical solution in real-time is challenging, especially with nonlinear dynamics and multiple constraints [3]. To this purpose, it is of interest to explore a parametric problem description and perform first-order extrapolations in the vicinity of reference solutions.

Sensitivity of parametric OCPs have been studied for a long time [4]. Assuming that one knows a solution for some prescribed parameter value, the idea is to deduce the neighboring solutions for small parameter variations [5]. Due to the practical drawbacks of indirect methods [6, Sec. 4.3], we will focus on solving OCPs via direct methods: OCPs are described, or at least approximated, using finite-dimensional variables, and then solved using Nonlinear Programming (NLP) [6]–[8]. Therefore, sensitivity analysis boils down to studying parametric NLP.

Classical results state under mild conditions that if, for the reference parameter, the multipliers are (strictly) positive, then the optimal solution is continuously differentiable [9]. Such problems are called *non-degenerate*. Computing the first order expansion of their solution only requires to invert a linear system, resulting from the application of the Implicit Function Theorem (IFT) on the Karush-Kuhn-Tucker (KKT) conditions [9, Thm. 3.2.2], [10]. This is a powerful result, with many applications [11], [12]. However, this positive multiplier assumption is not always satisfied, and degenerate problems with non-differentiable solutions are often encountered[1].

This paper focuses on two elements. First, it revisits a method to handle sensitivity analysis at degenerate points, for which weaker kind of differentiability is available - namely, existence of the upper Dini derivative - and details how to compute it. Second, it reports the key observation that PDG scenarios can be degenerate.

The paper is organized as follows. First, in Section II, a general class of OCP is formulated and then discretized. Secondly, in Section III, NLP sensitivity results are recalled and several computational aspects are discussed. Then, this framework is applied to rocket landing in Section IV. A rocket model is introduced and the necessity to be able to handle degenerate scenarios is proved. Also, a way to assess the proper reference parameter vicinity - ie. where the corrections remain valid - is presented. Finally, comments and extensions are provided in Section V.

### *Notations*

Vector inequalities must be understood element-wise.

For any map $h : \mathbb{R}^p \to \mathbb{R}^q$, denote its jacobian $h_x := \frac{\partial h}{\partial x} \in \mathbb{R}^{q\times p}$. When it exists, the upper Dini derivative of $h$ at $x$ in the direction $d$ is denoted by

$$D_d^+ h(x) := \lim_{t\downarrow 0} \frac{h(x+td)-h(x)}{t} \in \mathbb{R}^q.$$

## II. Optimal Control Problem

First, a class of optimal control problems is introduced, and then discretized via a direct approach.

### *A. General correction problem*

Consider a controlled system defined by its dynamic function $f$. For a control $u : [0,1] \to \mathbb{R}^m$, a parameter $p \in \mathbb{R}^{N_p}$, and for $t \in [0,1]$, the flow $\Phi_f$ of $f$ is defined such that $\Phi_f(t, x^0, p; u) = x(t)$ where the state $x$ of dimension $n$ satisfies the Initial Value Problem (IVP)

$$x(0) = x^0 \quad \text{and} \quad \dot{x} = f(x,u,p). \tag{1}$$

[1]A simple example being minimizing $z^2$ such that $z \geq p$, whose solution is $z = 0$ for $p < 0$ and $z = p$ for $p \geq 0$.

H. Ménou (hubert.menou@mines-paristech.fr) and N. Petit (nicolas.petit@mines-paristech.fr) are with MINES ParisTech, Centre Automatique et Systèmes, PSL University, 60 bd. St Michel, 75272 Paris Cedex, France, and E. Bourgeois (eric.bourgeois@cnes.fr) is with CNES, Direction des lanceurs, 52, rue Jacques Hillairet, 75612 Paris Cedex, FRANCE.

This work was supported by CNES and Mines ParisTech.

If $f$ is twice differentiable, so is $\Phi_f$ [13, Prop. 2.46].

Define a parametric control $u(z_1, t)$ where $z_1 \in \mathbb{R}^{N_1}$ is the parameter, such that $u(0, .) \equiv 0$ over $[0, 1]$. Typically, $u$ can denote piecewise polynomial functions in $t$ with linear dependency on $z_1$ [7, Sec. 3]. Also, denote the vector $z_2 \in \mathbb{R}^{N_2}$ s.t. $z = (z_1, z_2) \in \mathbb{R}^N$. Provided a reference control $\bar{u}$, we are interested in the following optimal control problem

$$\min_{z\in\mathbb{R}^N} \quad J(z,p) \tag{2a}$$
$$\text{s.t:} \quad \dot{x}(t) = f(x(t), \bar{u}(t) + u(z_1, t), p),\ \forall t \in [0,1] \tag{2b}$$
$$x(0) = h(p, z_2) \tag{2c}$$
$$g(x(t), \bar{u}(t) + u(z_1, t)) \le 0, \ \ \forall t \in [0,1] \tag{2d}$$
$$\Psi(x(1)) = 0 \tag{2e}$$

where $J$, $h$, $g$, $\Psi$ are twice-differentiable.

*Remark 1:* To handle problems with a final time $t_f$ other than 1, it is sufficient to consider the scaled time $\tilde{t} = t/t_f$ and multiply the dynamic equation by $t_f$. For free-final time problems, also add $t_f$ as an extra state of trivial dynamic in $f$, as shown later in Section IV-B.

We assume that $\bar{u}$, a reference control, is such that:

*Assumption 1:* If $p = 0$, then $z = 0$ satisfies the mixed state-control constraints (2d) and the terminal constraints (2e).

It is important to note that (2) is a *correction* problem. Indeed, $\bar{u}$ does not need to be the solution of any optimization problem. It can be whatever control that steers the system from $x^0$ to the condition $\Psi(x(1)) = 0$, while enforcing $g(x, \bar{u}) \le 0$. Naturally, the choice of $\bar{u}$ has a direct impact on the sensitivity analysis that follows.

Also, note that $J$ is general enough to convey a Mayer cost [7] by considering a map $\phi : \mathbb{R}^n \to \mathbb{R}$ and

$$J(z,p) = \phi\left(\Phi_f(1, h(p, z_2), p; u(z_1, .))\right). \tag{3}$$

From a practical perspective, computing the double derivative of (3) requires to evaluate the second derivative of $f$ (see eg. [13, Sec. 3.2]) which may require a significant analytical and numerical efforts in non-trivial contexts. To simplify this matter, $J$ may be taken directly quadratic in $z$ and mixed linear in $p$ and $z$ s.t.

$$J(z,p) = \frac{1}{2} z^\top W z + p^\top Q z \tag{4}$$

for some weighting matrices $W$ (symmetric) and $Q$.

### B. Discretization and NLP

To obtain a finite dimensional optimization problem with a finite constraint number, (2d) can be discretized following the direct collocation approach [6].

Denote by $0 = t_0 < t_1 < \ldots < t_{N_t} = 1$ an arbitrary subdivision of $[0, 1]$ in $N_t$ intervals. We choose to enforce (2d) only at those prescribed time instances. Consequently, Problem 2 writes as a nonlinear program

$$\text{NLP}(p) := \quad \min_{z\in\mathbb{R}^N} \quad J(z,p) \tag{5a}$$
$$\text{s.t:} \quad G(z,p) \le 0 \tag{5b}$$
$$H(z,p) = 0 \tag{5c}$$

where $x^k(z,p) := \Phi_f(t_k, h(p, z_2), p; u(z_1, .))$ fully conveys the dynamics equation and

$$G(z,p) := \begin{pmatrix} g(x^0(z,p), \bar{u}(t_0) + u(z_1, t_0)) \\ \vdots \\ g(x^{N_t}(z,p), \bar{u}(t_{N_t}) + u(z_1, t_{N_t})) \end{pmatrix}, \tag{6a}$$
$$H(z,p) := \Psi(x^{N_t}(z,p)). \tag{6b}$$

We will now focus on $\text{NLP}(p)$.

## III. Sensitivity Analysis Results

The shape of $\text{NLP}(p)$ is classic [14]. Provided a solution $z_0$ corresponding to a parameter $p_0$, the goal is to approximate the optimal value $z^*(p)$ near $p = p_0$.

In this section, results defining the first order expansion of $z^*$ at $p_0$ are recalled, and a method to effectively compute it is discussed. Without loss of generality, assume $p_0 = 0$ and $z_0 = 0$.

### A. Directional differentiability

For multipliers $\eta$ and $\lambda$, introduce the Lagrangian

$$L(z, \eta, \lambda, p) := J(z,p) + \eta^\top G(z,p) + \lambda^\top H(z,p). \tag{7}$$

Recall that a tuple $(z, \eta, \lambda, p)$ is said to satisfy the KKT conditions if $\eta \ge 0$ and

$$L_z(z, \eta, \lambda, p) = 0, \quad H(z,p) = 0, \quad \eta^\top G(z,p) = 0. \tag{8}$$

At $p_0$, the multipliers satisfying the KKT conditions are denoted $\eta_0$ and $\lambda_0$. The goal is to highlight general conditions under which the following (directional) expansion is defined

$$z^*(p) = z_0 + D_p^+ z^*(0) + o(\|p\|). \tag{9}$$

*Strict Complementarity Slackness* (SCS), ie. $\eta_0 > 0$, is often assumed to prove local uniqueness and differentiability of $z^*$, by applying the IFT on the KKT conditions [9]. However, this assumption does not necessarily hold, even on simple examples, as shown in the introduction. To circumvent this limitation, one can use *Strong* Second Order Sufficiency Conditions (SOSC)[2], as expressed in [15].

*Definition 1 (Strong SOSC):* There exist $a > 0$ s.t.

$$\nu^\top L_{zz}(z_0, \eta_0, \lambda_0, p_0)\nu \ge a\|\nu\|^2 \tag{10}$$

for all $\nu \in \mathbb{R}^{N_z}$ s.t. $H_z(z_0, p_0)\nu = 0$ and $G_z^j(z_0, p_0)\nu = 0$ for every component $j$ s.t. $G^j(z_0, p_0) = 0$ and $(\eta_0)_j > 0$.

As shown by Jittorntrum [15], relaxing the SCS using Strong SOSC instead eventually leads to directional differentiability properties, in the sense of the upper

[2]For a useful interpretation of this condition, see [10, Sec. 2.3].

Dini derivative. For any direction $p$, consider problem NLP($tp$), where $t \geq 0$ is a scalar.

*Theorem 1 (From [15, Thm. 3-4]):* Assume that at $z_0$

(i) the linear independence condition holds (ie. the columns of $H_z(z_0, p_0)$ and $G_z^j(z_0, p_0)$ for the active constraints $j$ are linearly independent),

(ii) Strong SOSC holds,

(iii) $J$, $G$, and $H$ are twice continuously differentiable in a neighborhood of $z_0$.

Then, there exists a unique continuous function $t \to (z^*(tp), \eta^*(tp), \lambda^*(tp))$ as the (local) solution of NLP($tp$), for $t \geq 0$, such that $z^*(0) = z_0$, $\eta^*(0) = \eta_0$ and $\lambda^*(0) = \lambda_0$. Furthermore, its right-hand derivative exists.

Thus, for any $p$ and since $z_0 = 0$, the following holds

$$z^*(tp) = D_p^+ z^*(0)t + o(t). \tag{11}$$

*Remark 2:* Directional differentiability results also exist via the indirect approach for parametric OCP [16].

### B. Computation method

When SCS is assumed, computing the expansion of $z^*$ only requires to invert a linear system [9]. However, this system becomes singular when SCS is not satisfied and the correction cannot be computed, hence the need for the above-mentioned tools, and the computational method detailed below. Jittorntrum's proof for Theorem 1 relies on a side Quadratic Program (QP) to serve intermediate theoretical purposes, which actually has powerful practical use.

*Proposition 1 (Adapted from [15, Eq. 24]):* Solving[3]

$$\mathrm{QP}(p) := \min_{\Delta z \in \mathbb{R}^N} \frac{1}{2} \Delta z^\top L_{zz}[0] \Delta z + p^\top L_{pz}[0] \Delta z \tag{12a}$$

$$\text{s.t: } G[0] + G_z[0]\Delta z + G_p[0]p \leq 0 \tag{12b}$$

$$H_z[0]\Delta z + H_p[0]p = 0 \tag{12c}$$

provides the value of $\Delta z := D_p^+ z^*(0)$.

Similar methods can be found in more recent work as well, see e.g. Bonnans & Shapiro [17, Sec. 5.2].

The primary goal of QP($p$) is to compute an approximation of $z^*(p)$ in the direction $p$, which works locally according to Theorem 1. However, inequality constraints that are inactive at $z_0$ can become active when $p$ increases (in norm). Provided that the approximation (12b) represent well the constraints, it will help QP($p$) to provide reasonable approximations of NLP($p$). In other words, QP($p$) manages non-local constraints using first-order constraint approximations [10].

The main advantage of solving QP($p$) instead of NLP($p$) is that one can rely on fast, mature and dependable[4] QP solvers [18], [19], and thus can use it for real-time applications. However, note that QP($p$) cannot be used as a brick for Successive QP, as the intermediate matrices would take too much time to compute.

[3]The compact notation $L[0] = L(z_0, \eta_0, \lambda_0, p_0)$ is used throughout the paper. Similar notation is used for $H$ and $G$.

[4]Not suffering from initial guess sensitivity to converge.

As a summary of the preceding discussion, provided a reference control $\bar{u}$ for Problem 2, computing in advance $J$, $G$, $H$, their first and second derivatives at $(z_0, p_0)$, and checking the assumptions of Theorem 1, one is able to reliably compute the expansion (11) with low computational effort using QP($p$).

A useful simplifying aspect should be noted. If $J$ is of the form (4), Assumption 1 implies that $\eta_0 = 0$ and $\lambda_0 = 0$ necessarily hold. Indeed, $z = 0$ globally minimizes $z \to J(z, 0)$ and $z = 0$ satisfies the constraints. Therefore, considering that the second order derivatives of $G$ and $H$ only appear in the cost of QP($p$), computing QP($p$) requires *only* the first order derivatives of $G$ and $H$. Thus, (12a) boils down to (4). This remark actually remains valid for general $J$ provided that $J_z(z_0, p_0) = 0$.

## IV. Application to Rocket Landing

The preceding results are now applied to the PDG problem. First, a rocket model and the choices corresponding to Problem (2) are described. Then, three examples showing the importance of the above-mentioned results are discussed. To alleviate the writing, variable dependencies are only written when truly necessary to avoid ambiguity.

### A. Rocket model

The approach is applied to PDG with a planar rocket model [20], [21], pictured in Figure 1. For $h \geq 0$ the altitude, $v_h < 0$ the vertical speed, $z$ the horizontal position, $v_z$ the horizontal speed and $m \geq m_{\mathrm{dry}}$ the total mass of the rocket, the dynamic equations are $\dot{h} = v_h$, $\dot{z} = v_z$ and

$$\dot{v}_h = -g + \frac{1}{m}\left(F_L \sin\theta + (T - F_D)\cos\theta\right) \tag{13a}$$

$$\dot{v}_z = \frac{1}{m}\left(F_L \cos\theta - (T - F_D)\sin\theta\right) \tag{13b}$$

$$\dot{m} = -q \tag{13c}$$

where $q$ is the engine flow, $\theta$ the attitude, and

$$\text{(Thrust)} \quad T := gI_{\mathrm{SP}}q - P_{atm}(h)S_E, \tag{14a}$$

$$\text{(Speed)} \quad V_r := \sqrt{v_h^2 + (v_z - w(h))^2}, \tag{14b}$$

$$\text{(Drag)} \quad F_D := \frac{1}{2}\rho(h)V_r^2 S_R C_D(M_a, \alpha), \tag{14c}$$

$$\text{(Lift)} \quad F_L := \frac{1}{2}\rho(h)V_r^2 S_R C_L(M_a, \alpha), \tag{14d}$$

$$\text{(Mach)} \quad M_a := V_r / S_{SP}(h), \tag{14e}$$

$$\text{(Incidence)} \quad \alpha := \arctan\left(\frac{v_z - w(h)}{|v_h|}\right) - \theta. \tag{14f}$$

Here, $g$ denotes the gravity, $I_{\mathrm{SP}}$ the engine specific impulse, $P_{atm}$ and $\rho$ the atmospheric pressure and density, $C_D$ and $C_L$ the drag and lift coefficients, $S_{SP}$ the sound speed, $w$ the horizontal wind and $S_R$ and $S_E$ characteristic rocket surfaces. The pressure bias in the thrust expression conveys the air flow envelop effect on the rocket when landing. When the rocket goes down vertically with no wind, $\alpha = \theta = 0$. The wind profile is assumed to be piece-wise affine, null at $h = 0$ and defined by its values $w_1$ at 5 km and $w_2$ at 10 km.

### B. OCP and NLP

The elements needed to build Problem 2 and NLP($p$) are presented here. The state $x$ is $(h, v_h, z, v_z, m)^\top$ and the control $u$ is $(q, \alpha)^\top$. Depending on the examples presented below, $p$ may convey any set of parameters describing the rocket model or its initial condition. It yields a dynamic equation $\dot{x} = f(x, u, p)$.

The terminal condition (2e) here corresponds to $h(t_f) = z(t_f) = v_z(t_f) = 0$ and $v_h(t_f) = -\varepsilon_f$ where $\varepsilon_f > 0$ denotes a non-zero small landing speed. For any time $t$, the constraints (2d) correspond to

$$q_{\min} \leq q(t) \leq q_{\max}, \qquad \alpha_{\min} \leq \alpha(t) \leq \alpha_{\max}. \tag{15}$$

Since the terminal condition implicitly determines the free final-time $t_f$, we add it to the dynamics as an extra state $\tilde{x} := (x^\top, t_f)^\top$. Thus, for all $s$ in $[0, 1]$, it yields

$$\dot{\tilde{x}}(s) = \tilde{f}(\tilde{x}(s), u(s), p) := \begin{pmatrix} t_f(s) f(x(s), u(s), p) \\ 0 \end{pmatrix}. \tag{16}$$

For the next examples, consider $N_t = 4$ and a parametric correction conveyed by a Cubic Spline [7], which requires the knowledge of $u$ at each time $t_0, \ldots, t_{N_t}$ (taken uniformly distributed) and its slopes at $t_0$ and $t_{N_t}$. Thus, $z$ equals

$$z := (u(t_0)^\top, \ldots, u(t_{N_t})^\top, \dot{u}(t_0)^\top, \dot{u}(t_{N_t})^\top, \tau)^\top \tag{17}$$

where $u(t_k) := (q(t_f), \alpha(t_k))^\top$, $\tau$ denotes the free-final time change and $N = 2(N_t+3)+1 = 15$. The cost (2a) is then taken quadratic in $z$, where $W$ from (4) is diagonal, with only positive scaling weights $w_q$ (resp. $w_\alpha$ and $w_{t_f}$) on components associated to the flow (resp. the incidence and the final time), ie. $J(z, p) = \dfrac{1}{2} z^\top W z$ and

$$W = \mathrm{Diag}\left(w_q, w_\alpha, w_q, w_\alpha, \ldots, w_{t_f}\right). \tag{18}$$

After having specified the framework for Problem 2 and NLP($p$), let us focus on the reference control $\bar{u}$ that will be used below. Consider a reference scenario of trajectory $\bar{x}$, shown in black in Figure 3-(e), starting at $x^0$, flying for $t_f$, with a constant engine flow $\bar{q} \equiv q_{\mathrm{cst}}$. The reference incidence control $\bar{\alpha}$ is such that it switches once between negative and positive values, as shown in Figure 3-(a) in black. $\bar{\alpha}$ is build such that it reaches its maximum value $\alpha_{\max}$ at some time $t_2$, and remains far from its minimum value $\alpha_{\min} = -\alpha_{\max}$ on the other side.

The assumptions of Theorem 1 are satisfied. First, the linear independence assumption is checked numerically, on the terminal condition and the only active inequality constraint, namely $\alpha(t_2) = \alpha_{\max}$. Second, as noted at the end of Section III, such a quadratic cost $J$ leads to null multipliers for $p = 0$ (even though one control constraint is active). Thus, $L_{zz}[0] = W$ is positive definite, showing

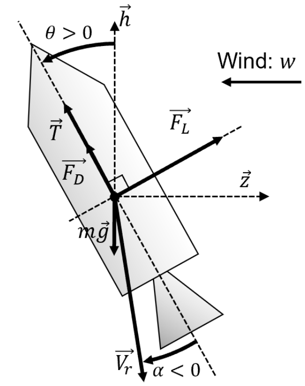


Fig. 1. Forces and angles at stake.

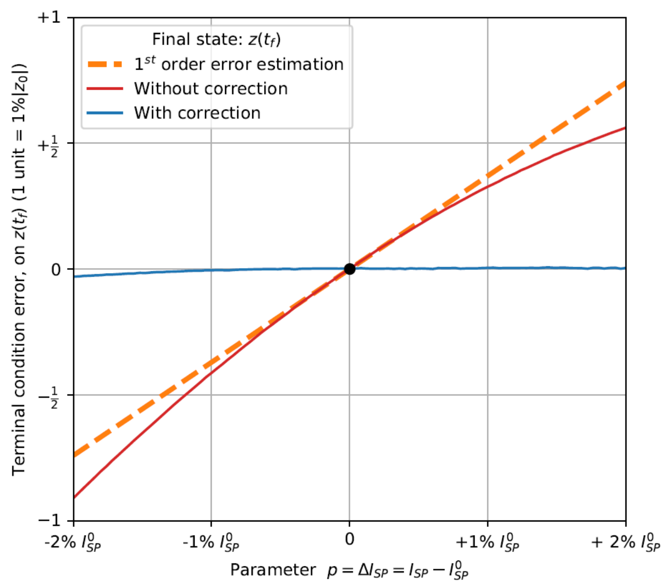


Fig. 2. Sensitivity of the terminal horizontal position with respect to variations of the $I_{\mathrm{SP}}$. In red is $z(t_f)$ when no correction is applied to $I_{\mathrm{SP}}$ variations. In dashed orange is the first order estimation of the red curve, extracted from the problem linearization. In blue is $z(t_f)$ when applying the corrections given by QP($\Delta I_{\mathrm{SP}}$). Terminal condition (2e) is thus locally well satisfied after correction.

that Strong SOSC holds. It also shows that SCS is not satisfied since $\eta_0 = 0$.

In practice, QP($p$) has been implemented using `cvxopt` [18], and the flow derivatives required in (6) using material from [13]. We can now dive into the examples.

### C. Direct application to $I_{SP}$ error correction

The first example aims at showing that in some neighborhood of the reference parameters, the expected expansion (11) provides accurate corrections enforcing the terminal constraints.

In this example, let us consider a single parameter : the error on the engine specific impulse $p = \Delta I_{\mathrm{SP}} := I_{\mathrm{SP}} - I_{\mathrm{SP}}^0$, where $I_{\mathrm{SP}}^0$ is the reference value.

Due to the terminal constraints, the four first states, namely $h$, $v_h$, $z$, $v_z$ are expected to be null at the final time (except $v_h(t_f) = -\varepsilon_f$). If the landing occurs with $p \neq 0$, but without correcting it, non-zero values are

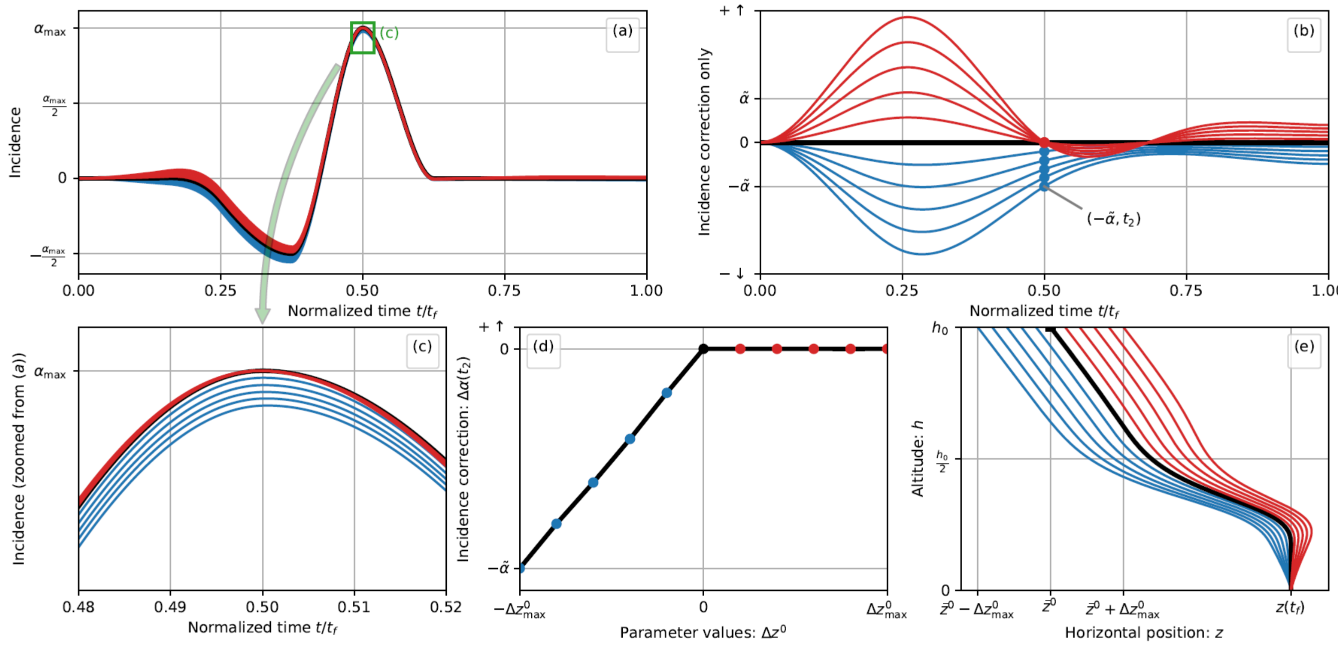


Fig. 3. Illustration of the non-differentiability of the optimal correction on a standard scenario. The landing trajectory is shown in $(e)$. The landing trajectories have been re-played on the nonlinear system, after using the correction from QP$(p)$. Blue features correspond to negative parameter values $\Delta z^0 < 0$ and red features to $\Delta z^0 > 0$. The focus is mainly on the incidence correction coefficient $\Delta\alpha(t_2)$. Its greatest magnitude value, defined in $(b)$, is denoted $\tilde{\alpha}$. The non-smooth behavior of $\Delta\alpha(t_2)$ w.r.t. parameter changes is clearly seen in $(d)$.

expected to appear. However, if the change in parameter is corrected using Proposition 1, the terminal constraints are assumed to be locally satisfied.

These two behaviors are well seen in Figure 2, representing the terminal horizontal position $z(t_f)$. The other terminal constraints on $h$, $v_h$ and $v_z$ have voluntarily been omitted, as their correction curve is much flatter than for $z$. In other words, it means that even though this terminal constraint component has the "worst" correction curve, it still demonstrates that the first order corrections brought by QP$(p)$ work well in a non-trivial neighborhood of $p = 0$ in practice.

### D. Non-smooth sensitivity to initial horizontal position

The second example aims at showing that the optimal solutions of NLP$(p)$ are indeed only Dini-differentiable and not smooth, even in a standard landing scenario.

Let us consider a single parameter, the error in initial horizontal position $p = \Delta z^0 := z^0 - \bar{z}^0$.

The directional-derivatives of $\alpha$ at the middle point $t_2$ in the directions $\Delta z^0 = 1$ and $\Delta z^0 = -1$ differ, as shown in Figure 3-(d). This behavior has a real-world interpretation. When $\Delta z^0 > 0$, using more incidence on $\alpha(t_2)$ is not an option as the constraint is already active and becomes strictly active (the associated multiplier becomes positive when $\Delta z^0 > 0$). However, when $\Delta z^0 < 0$, lowering $\alpha(t_2)$ is possible, allowing the corrections presented.

### E. Non-local $I_{SP}$ and initial horizontal position error correction

This last example focuses on assessing whether the corrections from QP$(p)$ can be used non-locally.

Let us first define a metric that assesses a correction correctness. When a parameter $p \neq 0$ is estimated, a correction is computed using QP$(p)$, and then used on the non-linear system in an open-loop fashion. The first-order changes in the terminal constraint is supposed to be null locally near $p \neq 0$. However, for significant values of $p$, nonlinear behavior will eventually make the corrections fail. Thus, we seek a performance function $\Gamma$ s.t. $\Gamma(p) \leq 1$ if and only if the terminal constraints are *acceptable*. Given some scaling factors $w_h$, $w_{v_h}$, $w_z$, $w_{v_z}$, consider the map

$$\mu(x) := \max\left\{\frac{|h|}{w_h}, \frac{|v_h + \varepsilon_f|}{w_{v_h}}, \frac{|z|}{w_z}, \frac{|v_z|}{w_{v_z}}\right\}. \tag{19}$$

When $\mu$ is applied on the final state $x(t_f)$, the scaling factors define what terminal constraint errors are *acceptable*. Thus, using the notations from (6), a suitable performance function is given by evaluating the terminal point

$$\Gamma(p) := \mu\left(x^{N_t}\left(D_p^+ z^*(0), p\right)\right). \tag{20}$$

Then, building upon the two previous example, consider the two-dimension parameter $p = (\Delta z^0, \Delta I_{\mathrm{SP}})^\top$. Evaluating $\Gamma$ on a square grid of $51 \times 51$ elements, it yields the level-set map shown in Figure 4. It shows a non-trivial area where first-order corrections from QP$(p)$ implies acceptable terminal condition satisfaction on the the true nonlinear system.

Note that this kind of green valleys that extend on the figure corners, or "cross pattern", conveys the fact that some errors compensate each other in a constructive way. Also, the non-smooth behavior of the contours at the

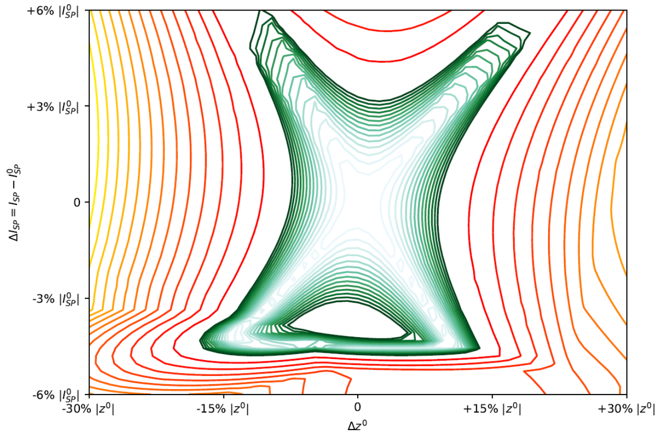


Fig. 4. Level sets for the performance function defined in (20). The contours in green denotes values lower than 1, where the parameter changes are corrected by QP($p$) within an acceptable tolerance. In orange shades are the values greater than 1.

bottom of Figure 4 correspond to the activation of one engine flow control constraint, hence this sharp rupture.

## V. Extensions and discussions

First, note that the choice of parameters in Section IV is arbitrary, and the number of parameters was chosen small for visualization purposes. It does not change the approach to add a great number of parameters (e.g. detailed atmospheric profiles, non-linear gravity model).

Second, linear parametric corrections $u$ are advantageous. Indeed, if $u(z,t)$ is taken linear in $z$ (e.g. Cubic Splines or Hermite polynomials [7]) and if $\bar{u}$ is described the same way, then the linearization (12b) of bounded control constraints, such as (15), is actually exact.

Moreover, more intricate mixed state-control constraints are under investigation (e.g. acceleration bounds), and will be the matter of future publications.

Finally, consider an offline/online procedure consisting in first computing a library of reference trajectories and its sensitivities before the flight, and then using QP($p$) on the closest reference trajectory during the flight. Such an approach has already been described in previous works [10] and used in robotics [12] for example. The need to be able to guarantee that the pre-computed trajectory library is rich enough to cover all needs is crucial for critical applications [22]. Thus, checking that the union of acceptable level sets for every reference trajectory - as in Figure 4 - represents a finite-covering of the possible inputs would demonstrate the completeness of the approach.

## VI. Conclusion

This paper has revisited useful results from the literature, regarding sensitivity analysis for optimal control problem. Its application to Powered Descent Guidance proved the need and how to circumvent the strict complementarity slackness assumption. Practical methods have been illustrated to assess the quality of the approximated corrections on the true nonlinear system.

## References

[1] B. A. Steinfeldt, M. J. Grant, D. A. Matz, R. D. Braun, and G. H. Barton, "Guidance, Navigation, and Control System Performance Trades for Mars Pinpoint Landing," *Journal of Spacecraft and Rockets*, vol. 47, pp. 188–198, Jan. 2010.

[2] L. Blackmore, "Autonomous Precision Landing of Space Rockets," *The Bridge*, 2016.

[3] M. Szmuk and B. Acikmese, "Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time," in *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018.

[4] A. Bensoussan, *Perturbation methods in optimal control*, vol. 5. Wiley, 1988.

[5] C. Büskens and H. Maurer, "Sensitivity Analysis and Real-Time Control of Parametric Optimal Control Problems Using Nonlinear Programming Methods," in *Online Optimization of Large Scale Systems*, pp. 57–68, 2001.

[6] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming.* Advances in Design and Control, SIAM, 2001.

[7] D. Kraft, "On Converting Optimal Control Problems into Nonlinear Programming Problems," in *Schittkowski K. (eds) Computational Mathematical Programming*, vol. 15 of *NATO ASI Series (Series F: Computer and Systems Sciences)*, 1985.

[8] J. Vlassenbroeck, "A chebyshev polynomial method for optimal control with state constraints," *Automatica*, vol. 24, pp. 499–506, July 1988.

[9] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, vol. 165 of *Mathematics in Science and Engineering.* Elsevier, 1983.

[10] C. Büskens and H. Maurer, "Sensitivity Analysis and Real-Time Control of Parametric Optimal Control Problems Using Boundary Value Methods," in *Online Optimization of Large Scale Systems*, pp. 17–55, 2001.

[11] R. Braun, I. Kroo, and P. Gage, "Post-optimality analysis in aerospace vehicle design," in *Aircraft Design, Systems, and Operations Meeting*, (Monterey,CA,U.S.A.), American Institute of Aeronautics and Astronautics, Aug. 1993.

[12] A. Reiter, H. Gattringer, and A. Müller, "Real-Time Computation of Inexact Minimum-Energy Trajectories Using Parametric Sensitivities," in *Advances in Service and Industrial Robotics*, vol. 49, pp. 174–182, Springer International Publishing, 2018.

[13] J. F. Bonnans, *Course on Optimal Control, Part I: the Pontryagin approach.* SOD311 Ensta Paris Tech, Aug. 2019.

[14] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming. Sequential Unconstrained Minimization Techniques.* Research Series, research analysis corporation ed., 1968.

[15] K. Jittorntrum, "Solution point differentiability without strict complementarity in nonlinear programming," in *Sensitivity, Stability and Parametric Analysis*, vol. 21, pp. 127–138, Springer Berlin Heidelberg, 1984.

[16] S. A. Deshpande, D. Bonvin, and B. Chachuat, "Directional Input Adaptation in Parametric Optimal Control Problems," *SIAM Journal on Control and Optimization*, vol. 50, pp. 1995–2024, Jan. 2012.

[17] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems.* New York, NY: Springer New York, 2000.

[18] M. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: Convex Optimization," Aug. 2020.

[19] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," Mar. 2014.

[20] N. X. Vinh, A. Busemann, and R. D. Culp, "Hypersonic and planetary entry flight mechanics," *NASA STI/Recon Technical Report A*, vol. 81, 1980.

[21] S. N. D'Souza and N. Sarigul-Klijn, "Survey of planetary entry guidance algorithms," *Progress in Aerospace Sciences*, pp. 64–74, 2014.

[22] M. Clark, X. Koutsoukos, J. Porter, R. Kumar, G. Pappas, O. Sokolsky, I. Lee, and L. Pike, "A Study on Run Time Assurance for Complex Cyber Physical Systems:," tech. rep., Defense Technical Information Center, Fort Belvoir, VA, Apr. 2013.