Fast Desensitized Optimal Control for Rocket Powered Descent and Landing*

Tommaso Robbiani†
Politecnico di Milano, Milan 20156, Italy

Marco Sagliano‡
DLR, German Aerospace Center, Bremen 28359, Germany

Francesco Topputo§
Politecnico di Milano, Milan 20156, Italy

Hans Seywald¶
Binera Inc., Rockville 20850, United States

This research revisits Desensitized Optimal Control Theory (DOC) for its application to a computationally challenging benchmark: a rocket descent and landing scenario. The primary objective is to assess the efficacy of the proposed method in mitigating the impact of perturbations on the final state, thereby establishing a framework capable of simultaneously optimizing guidance and control for the specified case. Additionally, our focus is on formulating a rapid and computationally efficient approach to enhance speed without compromising accuracy. The investigation begins with a comprehensive analysis of the fundamental components of the method, particularly the sensitivity terms and the computation of feedback gains, with a comparison of alternative formulations to evaluate their relative computational efficiency. Subsequently, the application of this methodology to the target problem is thoroughly examined with an a-priori performance index and characterized to reach the most efficient formulation, through the introduction of the idea of *dominant sensitivities*. Case-dependent modifications are explained and implemented to improve the methodology performances, resulting in the introduction of the *Marginal DOC Coefficient*, and the results are critically compared against those obtained using conventional methods through an extensive Monte Carlo analysis campaign.

*Presented as a conference paper AIAA-2024-0096 at the AIAA SciTech Forum and Exposition, Orlando, 08-12 Jan 2024

†PhD Student, Department of Aerospace Science and Technology, Via La Masa, 34; tommaso.robbiani@polimi.it

‡Senior GNC Research Engineer, Guidance, Navigation and Control Department, Robert Hooke Str. 7; marco.sagliano@dlr.de. Senior Member AIAA.

§Full Professor, Department of Aerospace Science and Technology, Via La Masa, 34; francesco.topputo@polimi.it. Senior Member AIAA.

¶Senior Technical Advisor, 77 S. Washington Street; hans.seywald@gmail.com. Member AIAA.

Nomenclature

$\boldsymbol{A}$	[ ]	Continuous linear dynamic matrix	s	[ ]	Performance Index
$\boldsymbol{B}$	[ ]	Continuous linear control matrix	T	N	Nominal thrust Value
$\boldsymbol{C}$	[ ]	Continuous linear parametric vector	t	s	Generic time
$\boldsymbol{f}$	[ ]	Generic Vector Function	$\boldsymbol{u}$	[ ]	Control Vector
g_M	$\mathrm{m\,s^{-2}}$	Mars's Gravity Acceleration	$\boldsymbol{v}$	$\mathrm{m\,s^{-1}}$	Velocity Vector
g_0	$\mathrm{m\,s^{-2}}$	Earth's Gravity Acceleration	v_i	$\mathrm{m\,s^{-1}}$	Velocity Coordinates
h	m	Altitude	$\boldsymbol{x}$	[ ]	State Vector
I_m	[ ]	Identity Matrix of dimension	$x,\ y$	m	In-Plane Coordinates
I_{sp}	s	Thrusters Specific Impulse	α	[ ]	Sensitivity Term Weight
J	[ ]	Augmented Cost Function	α_k	[ ]	State Sensitivity Weight
$\boldsymbol{K}$	[ ]	Feedback Gain Matrix	β_l	[ ]	Parametric Sensitivity Weight
m	kg	Lander's Mass	γ	deg	Thrusters Pointing Angle
n_p	[ ]	Number of parameters	η	[ ]	Saturation Handling Coefficient
n_q	[ ]	Number of control variables	θ	[ ]	Control Rate Weight
n_T	[ ]	Number of thrusters	$\boldsymbol{\Lambda}$	[ ]	Reduced Sensitivity Matrix
n_x	[ ]	Number of states	ν	[ ]	Marginal DOC Coefficient
n_Λ	[ ]	Number of $\boldsymbol{\Lambda}$ functions	ϕ	[ ]	Mayer Cost Function
n_Ω	[ ]	Number of $\boldsymbol{\Omega}$ functions	$\boldsymbol{\Omega}$	[ ]	Parametric Sensitivity Vector
$\boldsymbol{P}$	[ ]	Riccati Matrix	ω	[ ]	Control Norm Weight
p	[ ]	Generic Parameter	$(\cdot)_f$	[ ]	Final
$\boldsymbol{Q}$	[ ]	LQR State Error weight	$(\cdot)_0$	[ ]	Initial
$\boldsymbol{R}$	[ ]	LQR Control weight	$(\cdot)^*$	[ ]	Optimal
$\boldsymbol{S}$	[ ]	State Sensitivity Matrix			

I. Introduction

The advancement of rocket powered descent and landing technology stands as a pivotal factor in reshaping the future space economy landscape. The cost reductions associated with launchers' reusability have already altered space access and are poised to have a significant impact in the coming years. A crucial aspect lies in crafting optimal trajectories and the corresponding control design architectures for the powered descent and landing phase, which was the subject of several studies, [1–6]. In this context, uncertainties and disturbances can put the mission at risk if not properly addressed [7, 8]. Tackling these challenges requires a well-defined control strategy, typically designed independently

from the guidance system [9]. Many researches focus on the stabilization of the system when facing perturbations through Robust Control, e.g., through H_∞ techniques [10], which found recent application for expendable and reusable launchers in their structured form [11, 12]. These strategies are capable of mitigating perturbations but their detrimental effect on the feed-forward solution is only estimated a posteriori through nonlinear analysis and test flights. Even though this approach is largely adopted, the question whether a more effective strategy based on a concurrent design of guidance and control at the same time is open. Recent advancements in robust model predictive control (MPC) could significantly contribute to address the complex challenges of rocket landing problems. In this context, recent works include [13], which proposes a non-fragile robust MPC framework for nonlinear systems with constrained inputs, and [14], which offers an explicit feedback synthesis approach with guaranteed approximation quality and stability. A particularly prominent strategy is Robust Tube MPC [15], where a feedback controller ensures the system's state remains within a tube centered on a nominal trajectory, effectively containing uncertainties [16, 17]. For systems with state and control dependent uncertainties, semi-definite programming has been employed to compute invariant tubes [18]. However, as with most robust MPC formulations, these approaches may lead to conservative results due to the need to account for worst-case scenarios. The first approaches analyzing the possibility of sensitivity reduction through linear regulators date back to more than 40 years ago [19–21], and were later expanded to handle constraint restrictions to control inputs (constraint tightening method) [22]. In nonlinear problems, Stochastic Optimal Control problems are often formulated through the Bellman equation or the Hamilton-Jacobi-Bellman equation [23]. However, these methods are computationally expensive due to the high-dimensional search space and generally produce only open-loop solutions. To address the issue of high computational costs, alternative strategies such as Covariance Steering and Tube Stochastic Optimal Control have been proposed. Covariance Steering [24–26] attempts to set the covariance matrix terms as end-point constraints but it relies on a fully linear system derived from a predefined and sub-optimal mass profile. Recent efforts have integrated mean mass and covariance optimization within the covariance control framework, addressing both landing applications [27], where the vehicle's mass is treated as a random variable, and low-thrust scenarios [28], where mass covariance is managed through a variable transformation under a set of convex constraints. Conversely, tube stochastic optimal control [29] employs the unscented transform to convert a stochastic optimal control problem into a deterministic one. However, this approach heavily depends on the problem's mathematical formulation and is only valid for specific uncertainty distributions. In his works, Geller [30] introduced an innovative method for trajectory control and navigation analysis, employing the LinCov to predict 3σ trajectory dispersions and navigation errors. The approach was analytically extended to event triggers in orbital rendezvous applications [31] and to provide a computationally efficient alternative to Monte Carlo methods for error budget analyses for hosted payload systems [32]. LinCov offers simplicity and computational efficiency, however, the technique shows reduced accuracy when handling long uncertainty propagation, large initial uncertainties, or non-Gaussian uncertainty propagation. Another recent strategy utilizing implicit sets of functions is proposed in [33]; however, this method suffers from an indeterminate

upper bound for the Computational Process Unit (CPU) time, and focuses exclusively on feasible trajectories. More recently, Kim et al. developed a funnel synthesis algorithm via semidefinite programming to balance the maximization of the controlled invariant set with the minimization of disturbance effects [34].

In this complex scenario, Desensitized Optimal Control (DOC) was first proposed by Seywald et al. in 1996 [35]. The fundamental concept of the theory involves augmenting the problem's cost function with sensitivity-related terms, which can take various mathematical formulations and physical interpretations. Consequently, the original cost index of the problem is minimized along with the sensitivities, reducing the impact of the addressed perturbations. This theory serves as a comprehensive approach for mitigating the effects of uncertainties and disturbances on an optimal trajectory while simultaneously ensuring the optimization of both guidance and state feedback control gains in a unified manner. In DOC, robustness is inherently embedded within the trajectory design, with the controller playing a role in reinforcing and enhancing this robustness, achieving synchronous optimization.

The approach has been further extended in other works to consider also initial conditions perturbations and parametric uncertainties [36] and to embed control constraint limitations [37]. Even though a significant amount of work has still to be done in its theoretical definition [38, 39], the methodology has been successfully applied to the landing problem both for the aerodynamic phase [40] and the powered phase [41], for the entry guidance [42], with aerodynamic uncertainties [43] and with direct collocation and nonlinear programming [44].

Despite the presented advantages, a major drawback of the DOC approach is the dramatic increase in the number of differential equations underlying the dynamics of the problem. The sensitivity matrix-based approach requires propagating the n_x original states, the n_p uncertain parameters, and the elements in the sensitivity matrix, resulting in a total of $(n_x + n_p)^2 + n_x + n_p$ number of states [35]. This increase in complexity limits the method's applicability, even for simple problems outside the aerospace domain. However, this bottleneck is particularly significant for trajectory optimization, where for design and analysis many alternatives might be required and evaluated, rendering the DOC strategy less practical, unless methods are incorporated to speed up the numerical calculations. Recent works [45–47] successfully tackled the problem through reduced-order schemes. However, all the proposed approaches lack in generality as they respectively focus on parametric uncertainties rather than uncertain states, with fixed final-time problems and open-loop applications, paving the way to computational improvements for just specific cases.

This work presents a computationally efficient approach (consequently dubbed *fast*) to make general OCP solutions more robust. The proposed method is a comprehensive and analytically efficient approach that preserves key features of DOC strategy, such as integrated control design (as in [48, 49]) and sensitivity penalization. It establishes a unified framework capable of addressing both state and parametric uncertainties across various control problems in an efficient manner by reducing the number of optimization variables, thereby improving computational efficiency compared to existing DOC formulations. This general framework is then adapted to the Rocket Landing Problem, offering three further improvements to the already existing methodologies. First, we show that finding and desensitizing a specific

sensitivity leads to the best global results, and therefore we introduce the concept of *dominant sensitivity*. By addressing the significant issue of increased state variables required by the standard DOC theory, we show that the concept of dominant sensitivity results in a substantial reduction in the required CPU time needed to compute a valid solution. Second, we generalize the original strategy presented in [41] to handle control saturation, through the introduction of what we call *marginal DOC coefficient*, which does have a relevant impact on the performance that we obtained, especially in terms of final state covariance reduction. Third, we present a synthetic a-priori performance index able to estimate the final state uncertainties for the considered scenario, based on the direct link between covariance and sensitivity [50]. This is done in order to simplify the trajectory design process, reducing the need for expensive a-posteriori simulations to assess the performances of the guidance and control subsystem.

The paper is structured as follows: Section II presents the novel aspects of our approach that substantially reduce computational time without compromising accuracy. In Section III, we present the numerical benchmark provided by the Rocket Powered Descent and Landing problem on Mars, introducing the a-priori performance index, exploring the proprieties of the ad-hoc defined marginal DOC coefficient and proposing the dominant sensitivity concept. Finally, Section IV contains the results obtained and their discussion, focusing on the method performances and comparing the computational effort of existing strategy with the proposed one. This work culminates in the conclusions outlined in Section V.

II. Fast Desensitized Optimal Control Formulation

Sensitivities are used to assess how perturbations in states or parameters affect future states. Consequently, two primary categories emerge: sensitivity matrices [35], which map the impact of state perturbations on future states, and sensitivity functions [45], which gauge the effect of parametric uncertainties. Starting from the early definition of DOC sensitivity matrix, we first focus on the definition of the reduced-order formulation, highlighting the connected computational enhancement with respect to the classical sensitivity matrices. The sensitivity function for efficiently handling parametric perturbation is then revised, and the Fast DOC cost function is presented. In the second part of the section we compare different strategies to implement the integrated control design, giving the reasons for the LQR architecture choice in the Fast DOC formulation.

A. Sensitivity Matrix, Sensitivity Function and Fast DOC Cost Function

In the DOC framework, any optimal control problem's cost function can be augmented with terms corresponding to the categories of sensitivities. Without loss of generality, we can restrict ourselves to problems formulated in Mayer form, since Lagrange and Bolza problems can always be transformed into Mayer ones [51]. In the DOC framework, the cost term J_S, associated with the sensitivity matrix, and J_Ω, linked to the sensitivity function, are appropriately weighted by a non-negative coefficient α. These terms are penalized concurrently with the original problem's cost

function $\phi(\boldsymbol{x}(t_f), t_f)$, where $\boldsymbol{x}(t)$ represents the state vector and t_f denotes the final time

$$J = \phi(\boldsymbol{x}(t_f), t_f) + \alpha \left(J_S + J_\Omega\right) \tag{1}$$

Considering a generic dynamic system represented by $\boldsymbol{f}(\boldsymbol{x}(t), t) \in \mathbb{R}^{n_x}$, subject to the initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \mathbb{R}^{n_x}$, the sensitivity matrix $\mathbf{S} \in \mathbb{R}^{n_x \times n_x}$ is defined according to Eq. (2) with the identity matrix $\mathbf{I}_{n_x}$ as initial condition.

$$\mathbf{S}(t, t_0, \boldsymbol{x}_0) = \frac{\partial}{\partial \boldsymbol{x}_0} \boldsymbol{X}(t, t_0, \boldsymbol{x}_0), \tag{2}$$

Here, $\boldsymbol{X}(t, t_0, \boldsymbol{x}_0)$ represents the state at time t obtained by propagating the dynamical equations $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), t)$ from the given initial conditions, $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. For brevity, the matrix $\mathbf{S}(t, t_0, \boldsymbol{x}_0) \in \mathbb{R}^{n_x \times n_x}$ can henceforth be denoted as $\mathbf{S}(t)$ and explicit time dependencies on time drop in favor of a simpler notation whenever not needed. The dynamics of $\mathbf{S}(t)$ is given by:

$$\dot{\mathbf{S}}(t) = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t)}{\partial \boldsymbol{x}} \right)_{x = X(t, t_0, \boldsymbol{x}_0)} \mathbf{S}(t) = \mathbf{A}(t)\, \mathbf{S}(t) \quad \mathbf{S}(t_0) = \mathbf{I}_{n_x} \tag{3}$$

where the matrix $\mathbf{A}(t) \in \mathbb{R}^{n_x \times n_x}$ comes from the linearized problem dynamics.

Nevertheless, in many practical applications, the sensitivity of the final state with respect to state perturbations has seminal importance. For example, in a landing scenario, the mission's success depends critically on the accuracy of both final position and final velocity. With the definition provided earlier, the sensitivity of the final state can be expressed in two distinct ways. The first method is straightforward and involves evaluating the sensitivity matrix $\mathbf{S}(t)$ defined in Eq. (2) at the final time, measuring the impact of perturbations arising at the initial time, while the sensitivity matrix $\mathbf{S}_2(t)$ assesses how perturbations at any time affect the final states, exploiting the corresponding inverse of the sensitivity matrix, as reported in Eq. (4). The physical interpretation of this matrix is discussed in [35], where its mathematical properties are also established.

$$\mathbf{S}_1 = \mathbf{S}(t_f) \qquad \mathbf{S}_2(t) = \mathbf{S}(t_f, t, \boldsymbol{x}) = \mathbf{S}(t_f)\, \mathbf{S}^{-1}(t) \tag{4}$$

Both matrices can be obtained by propagating the differential equation Eq. (3) and computing the required terms.

According to [35], the definition of $\mathbf{S}_2(t)$ requires augmenting the state vector with the components of $\mathbf{S}(t)$ as well as with another set of variables, the $\mathbf{S}_f$ components, constant values matching the final condition of $\mathbf{S}(t)$, leading to a significant increase in the number of required state variables. A second potential issue of the classical DOC methodology is related to the computation of the inverse of $\mathbf{S}(t)$, which can be computationally costly and problematic for ill-conditioned matrices. An innovative strategy to address the latter issue is to propagate $\mathbf{S}^{-1}(t)$ rather than $\mathbf{S}(t)$ and then invert it only at the final time to obtain $\mathbf{S}(t_f)$, as shown in $\mathbf{S}_2(t)$ in Eq. (4). The differential equations governing

$\mathbf{S}(t)$ are linear, making the dynamics of $\mathbf{S}^{-1}(t)$ straightforward to obtain. However, while this architecture mitigates numerical issues related to matrix inversion, it still requires the same number of additional state variables as in its original formulation to match the final conditions. Therefore, the most efficient solution is to propagate the entire $\mathbf{S}_2(t)$ matrix of Eq. (4) [36], where the initial boundary value problem becomes a final boundary value problem, not requiring $\mathbf{S}_f$ states, as the final conditions correspond to the identity matrix $\mathbf{I}_{n_x}$.

$$\dot{\mathbf{S}}_2(t) = -\mathbf{S}_2(t)\,\mathbf{A}(t), \quad \mathbf{S}_2(t_f) = \mathbf{I}_{n_x} \tag{5}$$

According to the definition of sensitivity, all the elements of the matrices establish connections among state variables: if the connected states are independent, the corresponding sensitivity terms becomes zero. Thus, it is reasonable to focus on specific components of the sensitivity matrix, rather than the full $\mathbf{S}_1$ or $\mathbf{S}_2(t)$ matrices. The rows of a sensitivity matrix indicate how a single final state component depends on other states, either at the initial or a specific time. Conversely, the column mappings reveal how the entire final state depends on single time-varying states. Depending on the application and analytical objective, different rows and columns may be of interest. These selected terms can be stacked into some new vectors $\mathbf{\Lambda} \in \mathbb{R}^{n_x \times 1}$, representing a linear combination of rows (denoted by the superscript r) or columns (denoted by the superscript c) of the $\mathbf{S}$ matrices. The subscripts (1) or (2) are adopted in order to distinguish whether the rows/columns are components of $\mathbf{S}_1$ or $\mathbf{S}_2(t)$. This concept, partially introduced in [50] under the category of reduced dimensionality formulations, exploits the linearity of the differential equations governing sensitivity matrix dynamics to define the sensitivity of a user-defined function of the final state. This formulation enables reduced-order representations of the problem by focusing on the most relevant components of the sensitivity matrices, achieved through the appropriate definition of n_Λ $\mathbf{\Lambda}$ functions. This reduces the computational complexity, decreasing the number of required states from n_x^2 to $n_\Lambda\, n_x$, compared to standard DOC approaches. As a result, it supports tailored optimization strategies that adapt to varying problem complexities while significantly improving computational efficiency. However, this approach ensures no loss of generality, as the original sensitivity matrix can be fully reconstructed by defining appropriate $n_\Lambda = n_x$ functions.

The simplest approach to obtain $\mathbf{\Lambda}$ vectors is to compute $\mathbf{S}_1$ or $\mathbf{S}_2(t)$ through the propagation of the nominal $\mathbf{S}(t)$ matrix and then extract the required terms. However, both formulations waste the numerical advantages of defining just n_Λ vectors as the full $\mathbf{S}$ matrices are propagated though the n_x^2 linear differential equations. A convenient reduced-order formulation is obtained by noting that each $\mathbf{\Lambda}(t)_2^r$ vector collects a liner combination of rows of $\mathbf{S}_2(t)$. Therefore, the transpose of these vectors obey the same differential equations describing the evolution of $\mathbf{S}_2(t)$ and have a vector of $\mathbf{\Lambda}_{f,2}^r \in \mathbb{R}^{n_x \times 1}$ with non-zero values in correspondence of the building rows of $\mathbf{\Lambda}(t)_2^r$ as final conditions [36].

$$\dot{\mathbf{\Lambda}}(t)_2^r = -\mathbf{A}(t)^T\,\mathbf{\Lambda}(t)_2^r, \quad \mathbf{\Lambda}(t_f)_2^r = \mathbf{\Lambda}_{f,2}^r \tag{6}$$

It is straightforward to notice that a computational advantage is gained if $n_\Lambda < n_x$, as this formulation involves just $n_\Lambda\, n_x$ differential equations. The smaller n_Λ, the greater the improvement with respect to the n_x^2 differential equations in Eq. (5). The same idea can be extended to $\mathbf{\Lambda}_1^c$ with dual initial conditions (check Eq. (5)), namely $\mathbf{\Lambda}(t_0)_1^c = \mathbf{\Lambda}_{0,1}^c \in \mathbb{R}^{n_x \times 1}$ with non-zero positive values in correspondence of the building columns. The other two formulations ($\mathbf{\Lambda}(t)_2^c$ and $\mathbf{\Lambda}_1^r$) are not compatible with the presented reduced-order scheme, requiring the propagation of the full $\mathbf{S}_1$ or $\mathbf{S}_2(t_f)$ matrices to be defined. Among the two model order reduction strategies, in this work we select the $\mathbf{\Lambda}(t)_2^r$ formulation and neglect $\mathbf{\Lambda}_1^c$ due to a specific physical interpretation: since the primary focus is on the final state rather than the perturbations themselves, it is more meaningful to consider rows of the sensitivity matrix, measuring how a component of the final state is affected by all perturbations, rather than penalizing the sensitivity of the full final state with respect to a single initial perturbation, as would be for $\mathbf{\Lambda}_1^c$. Furthermore, this choice allows a more conservative approach when building the cost term J_S, as it enables the integral penalization of the $\mathbf{\Lambda}(t)_2^r$. In certain dynamic systems, sensitivity peaks can occur at specific time instants, making the system highly sensitive to perturbations at those moments. This effect is not captured when considering the sensitivity $\mathbf{\Lambda}_1^c$, while a penalization in integral form of the 2-norm of $\mathbf{\Lambda}(t)_2^r$ can effectively reduce the sensitivity peaks along the trajectory. In addition to generic physical state perturbations (e.g., initial errors), this formulation also allows for the mitigation of disturbances arising at any time if they are included in the dynamics and properly expressed as function of the states of the problem. Therefore, the resulting sensitivity cost term is

$$J_S = \int_{t_0}^{t_f} \sum_{k=1}^{n_\Lambda} \alpha_k \|\mathbf{\Lambda}(t)_{k,2}^r\|_2 \, \mathrm{d}t = \int_{t_0}^{t_f} \sum_{k=1}^{n_\Lambda} \alpha_k \|\mathbf{\Lambda}(t)_k\|_2 \, \mathrm{d}t \tag{7}$$

where $\mathbf{\Lambda}(t)_2^r$ is from now on simply referred to as $\mathbf{\Lambda}(t)$ for notation convenience. The choice of n_Λ functions to be penalized is problem-dependent: for the considered Rocket Landing application, this choice is addressed in detail in Sec. III.D.

More generally, in certain scenarios, the dynamic evolution of the system is influenced by n_p model parameters, whose uncertainties play a significant role, making it worthwhile to penalize the second sensitivity term in Eq. (1), J_Ω. Without loss of generality, a single uncertain parameter $p(t)$ can be considered, and the dynamics accounting for it reads $\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{f}}(\boldsymbol{x}, t, p)$. Although this topic has been addressed in [35], a more convenient formulation is presented in [45], capable of reducing the number of states required to map the sensitivity of the state vector with respect to variations in the model parameters. This formulation involves a typical sensitivity definition and can be mapped through a vector, $\mathbf{\Omega}(t) \in \mathbb{R}^{n_x \times 1}$, defined as:

$$\mathbf{\Omega}(t) = \frac{\partial \boldsymbol{x}(t, p)}{\partial p}_{x = x(t, p_0)} \tag{8}$$

where p represents the model parameter, with a nominal value of p_0. As the primary goal of desensitization in this case is to minimize the impact of the perturbation p on the final state, it is straightforward to penalize the norm of $\mathbf{\Omega}(t)$, evaluated at the final time t_f, that is obtained by propagation of the following differential equation.

$$\dot{\mathbf{\Omega}}(t) = \tilde{\mathbf{A}}(t)\mathbf{\Omega}(t) + \left(\frac{\partial \tilde{\boldsymbol{f}}(\boldsymbol{x}, t, p)}{\partial p}\right)_{x=x(t,p_0)} \triangleq \tilde{\mathbf{A}}(t)\,\mathbf{\Omega}(t) + \tilde{\boldsymbol{C}}(t), \qquad \mathbf{\Omega}(t_0) = \mathbf{0} \tag{9}$$

Although we considered further penalty terms, such as integral ones, this approach emerges as the most relevant from a physical perspective. A total of $n_\Omega \leq n_p$ functions can be penalized, resulting in the sensitivity function cost term:

$$J_\Omega = \sum_{l=1}^{n_\Omega} \beta_l \|\mathbf{\Omega}(t_f)_l\|_2 \tag{10}$$

After establishing the foundational elements, the cost terms associated with the sensitivity matrix as in Eq. (7) and the sensitivity function of Eq. (10) can be merged to build the comprehensive DOC cost function, as follows.

$$J = \phi(\boldsymbol{x}(t_f), t_f) + \alpha \left(\int_{t_0}^{t_f} \sum_{k=1}^{n_\Lambda} \alpha_k \|\mathbf{\Lambda}(t)_k\|_2 \, \mathrm{d}t + \sum_{l=1}^{n_\Omega} \beta_l \|\mathbf{\Omega}(t_f)_l\|_2 \right) \tag{11}$$

This formulation allows for the restoration of the original problem by setting α to zero. The importance of sensitivity terms with respect to the original cost of the problem increases with higher α values, prompting the minimization algorithm to prioritize their reduction. As this formulation supports selective propagation, each sensitivity term is computed only when necessary, specifically when its multiplicative coefficient in the cost function is greater than zero.

B. Integrated Feedback Control

Feedback control is a cornerstone of DOC theory. As elucidated in [35], the controller gains can be integrated into the optimization process via the affine feedback control law valid in the neighboring of the optimal solution:

$$\boldsymbol{u}(t) = \boldsymbol{u}^*(t) + \boldsymbol{K}(t)\,(\boldsymbol{x}(t) - \boldsymbol{x}^*(t))$$

Here, $\boldsymbol{u}(t) \in \mathbb{R}^{n_q \times 1}$ is the control vector, the $\boldsymbol{K}(t) \in \mathbb{R}^{n_q \times n_x}$ matrix encapsulates the feedback gains, while the superscript * denotes the optimal solution of the problem. Consequently, the closed-loop state dynamics can be defined as:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}, t) = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}^*(t) + \boldsymbol{K}^*(t)\,(\boldsymbol{x} - \boldsymbol{x}^*(t)), t\right) \tag{12}$$

This significantly transforms the problem formulation by introducing differential equations that govern $\mathbf{\Lambda}(t)$ and $\mathbf{\Omega}(t)$ behaviors. Assuming $\hat{\boldsymbol{A}}(t)$ and $\hat{\boldsymbol{B}}(t)$ as the matrices governing the dynamics of the linearized version of Eq. (12), Eq. (6) and Eq. (9) respectively become:

$$\dot{\mathbf{\Lambda}}(t) = -\left(\hat{\boldsymbol{A}}(t) + \hat{\boldsymbol{B}}(t)\boldsymbol{K}(t)\right)^T \mathbf{\Lambda}(t), \quad \mathbf{\Lambda}(t_f) = \mathbf{\Lambda}_f \tag{13}$$

$$\dot{\boldsymbol{\Omega}}(t) = \left(\hat{A}(t) + \hat{B}(t)K(t)\right)\boldsymbol{\Omega}(t) + \hat{\boldsymbol{C}}(t), \quad \boldsymbol{\Omega}(t_0) = \mathbf{0} \tag{14}$$

If the linearized system is stable, the sensitivity matrix dynamics will also be stable. In the absence of feedback gains, if the system dynamics equation is unstable, the terms in the sensitivity matrix grow, starting from the initial identity matrix condition. This underscores the essence of feedback control laws, which find a clear explanation through sensitivity definitions. Various strategies exist for computing these gains. Users may prescribe these gains [37], treat them as constants determined by the solver, or define them as time-varying variables [35]. Another approach involves obtaining these gains through structured methodologies [41]. The *User Defined* solution is the most straightforward and offers the notable advantage of significantly reducing the computational effort required by the solver. However, a deep understanding of the problem is necessary to tune the feedback gains, and this may not always be achievable. Furthermore, fixed user-defined values can lead to sub-optimal solutions, as they restrict the exploration of the research state-space of the guidance and control contemporary optimization. Conversely, *Free gains* (that is, treating the gains as additional virtual controls added to the problem) provide the highest level of flexibility to the optimizer, but it is then difficult to include stability requirements for the resulting closed-loop system in the formulation. Although different solutions have been tested and implemented, the most effective approach involves penalizing the feedback gains matrix $\boldsymbol{K}(t)$ and the rates of change of its elements, $\boldsymbol{U}_K(t)$, with appropriate non-negative weights in the cost function defined in Eq. (11).

$$J_K = J + \omega \int_{t_0}^{t_f} \|\boldsymbol{K}(t)\|_2 \, \mathrm{d}t + \theta \int_{t_0}^{t_f} \|\boldsymbol{U}_K(t)\|_2 \, \mathrm{d}t$$

The first term imposes a quadratic constraint on feedback effort, as shown in [35], while the second term, newly introduced in this work, aims to generate smoother solutions by incorporating the gain rates as additional control variables. Despite its versatility, this method also presents challenges. Traditional regulator tuning methods such as LQR or H-inf provide system stability, which is not inherently guaranteed with this formulation. Furthermore, fine-tuning the controller performance is challenging because the effects of different ω and θ values on the solution's optimality are not clearly predictable. Consequently, we decided to opt for a classical LQR control structure for mainly two reasons: first, to ensure stability of the resulting closed-loop architecture. Second, we adopted this strategy due to its simplicity of implementation and straightforward tuning process, since the gains are computed by propagating Riccati Equations to obtain the evolution of $\boldsymbol{P}(t) \in \mathbb{R}^{n_x \times n_x}$ matrix alongside trajectory generation. In doing so, we can also leverage the symmetry properties of the matrix $\boldsymbol{P}(t)$ to implement the synthesis of the gains in a more efficient way. Even though a weighted-frequency approach could lead to more robust solutions, (e.g., *Structured Approaches* [12, 52]), capable of efficiently handling different perturbations (e.g., wind at different frequencies), we opted for LQR due to the demonstrative nature of this work and the simpler implementation in DOC framework, although the approach proposed

here can be extended to incorporate such more advanced approaches. Our choice aligns with existing literature [41], and our focus was on understanding how state and control weight matrices shape the final solution's optimality, both in terms of guidance and control. One critical aspect is that, by selecting a specific control synthesis technique, as we are doing with LQR, the gains cannot be longer freely adjusted and utilized as optimization variables. Thus, at each iteration of the trajectory optimization process the optimal gains are computed through Riccati Matrix propagation. This choice can be seen as a trade-off between traditional decoupled methods, where guidance and control are independently designed, and the fully-integrated free gains approach aforementioned, where the coupling between guidance and control is complete, but stability issues arise.

III. Powered Landing Scenario

A. Mission and Scenario

The primary objective of the powered landing phase is to reach the prescribed target position, with a predetermined final velocity while minimizing a specified cost index, e.g., time, fuel consumption, energy [3]. This study focuses solely on mass optimization, and the cost function is tailored accordingly. To simplify the problem, a basic dynamical model is employed. The landing body is treated as a point mass, therefore neglecting the rotational degrees of freedom to avoid the complexity of a full rigid-body motion. Aerodynamic forces, rotational effects of Mars, and variations in the planet gravity field are also disregarded, considering a constant gravity acceleration value g_m. The rocket thrusters consist of a set of n_T identical engines, canted at an angle γ with respect to the net thrust direction, all providing the same nominal thrust value, T. Specific impulse, I_{sp}, and Earth's standard gravity, g_0, are two additional parameters that quantify the thrusters' performance. Further details regarding the vehicle configuration and assumptions can be found in [53]. The equations of motion are formulated with respect to a target-centered, surface-fixed, reference frame. The axes, denoted as $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$, span the horizontal plane tangent to the planet surface at the landing point. The $\hat{\boldsymbol{h}}$ axis points opposite to the direction of the local gravity field. Despite being inherently non-inertial, this system is assumed to be inertial due to the relatively short time of flight associated with this scenario. The state vector $\boldsymbol{x} \in \mathbb{R}^{7\times 1}$ comprises displacements (x, y, h) and velocities (v_x, v_y, v_h), along with the mass (m) of the lander. The lander motion is controlled through thrust modulation using the vector $\boldsymbol{u}(t) = \left[\, u_x(t),\ u_y(t),\ u_h(t) \,\right]$ of size 3×1, with each component corresponding to control commands along the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{h}}$ axes, respectively.

The dynamics of the system are described by the set of differential relationships of Eq. (15), subject to control constraints and no-subsurface flight (Eq. (16)).

$$
\begin{aligned}
\dot{x} &= v_x, \qquad & \dot{v}_x &= \frac{u_x\, T\, n_T\, \cos\gamma}{m}, \\
\dot{y} &= v_y, \qquad & \dot{v}_y &= \frac{u_y\, T\, n_T\, \cos\gamma}{m}, \\
\dot{h} &= v_h, \qquad & \dot{v}_h &= \frac{u_h\, T\, n_T\, \cos\gamma}{m} - g_m, \\
& & \dot{m} &= -\frac{u\, T\, n_T}{I_{sp}\, g_0}
\end{aligned}
\tag{15}
$$

$$
0 < u_{min} \leq u = \sqrt{u_x^2 + u_y^2 + u_h^2} \leq u_{max} \qquad h(t) > 0, \qquad \forall t < t_f \tag{16}
$$

The norm of the thrust vector $\boldsymbol{u}$, denoted as u, is constrained to satisfy specific lower, u_{min}, and upper limits, u_{max}, in terms of thrust magnitude. Additional constraints, such as maximum glide slope angle, discussed in [53], are disregarded due to the preliminary nature of this work. The numerical values of the parameters are provided in Tab. 1, following the corresponding benchmarks ([41, 53]).

Table 1 Parameters Values

Parameter	Value	Unit
n_T	3	-
γ	27	deg
g_m	3.71	$\mathrm{m\,s^{-2}}$
g_0	9.81	$\mathrm{m\,s^{-2}}$
T	3100	N
I_{sp}	225	s
u_{min}	0.3	-
u_{max}	0.8	-

The boundary conditions for the problem, reported in Tab. 2, are consistent with the ones described in [41]. The final time is left free to vary to provide more flexibility to the optimization algorithm.

Table 2 Boundary Conditions

State	Initial Condition	Final Condition	Unit
x	1900	0	m
y	800	0	m
h	3100	0	m
v_x	40	0	$\mathrm{m\,s^{-1}}$
v_y	20	0	$\mathrm{m\,s^{-1}}$
v_h	-50	0	$\mathrm{m\,s^{-1}}$
m	1905	`free`	kg

The nominal solution of the problem, referred to as noDOC, can be obtained, revealing a control profile characterized

by a classical *max-min-max* profile, (see [3] for an analysis of the possible control structures of this class of problems).

B. Assumptions for the DOC Architecture

Among the potential sources of disturbances, such as aerodynamic perturbations or non-nominal component behaviors, only a subset of error sources are explicitly addressed in this work. Although the methodology proposed in Sec. II.A can attenuate the influence of state perturbations occurring at any time, it explicitly addresses only linear perturbations that depend on state variables or parameters. External disturbances on systems that are linear do not affect stability, since the characteristic polynomial is unaffected by disturbances. For the nonlinear case however, the notion of input-to-state stability can be used [54–57]. For instance, in the aerodynamics model, the effect of wind gusts in the $\hat{\boldsymbol{x}}$ direction can be mitigated by penalizing the final state sensitivity with respect to v_x; however, this does not account for wind gust amplitudes, as they are not states. More advanced modeling approaches are beyond the scope of this work. A similar rationale applies to the uncertain drag coefficient, often modeled as linearly dependent on a parameter: without explicit aerodynamic considerations, penalizing its related sensitivity function is impractical. Accordingly, this paper focuses on large initial state estimation errors and fluctuations in nominal thrust values, both of which have the potential to result in complete vehicle loss [7, 8]. The influence of the formers is alleviated by penalizing specific $\boldsymbol{\Lambda}(t)$ terms, while the impact of the latter is mitigated through a penalty applied to a suitable $\boldsymbol{\Omega}(t)$ function, depending just on the maximum thrust parameter T, as proposed in Sec. II.A. For the specific application we consider in this work, the position uncertainties are assumed one order of magnitude larger that the velocity ones.

C. Proposed DOC Architecture for the Powered Landing Scenario

Given the assumptions formulated in the previous discussion, the most comprehensive implementation strategy involves expanding the state vector with $\boldsymbol{\Lambda}(t)$ vectors, the $\boldsymbol{\Omega}(t)$ components, and the elements of the Riccati Matrix, which are penalized in the cost function. Since thrust is the unique uncertain parameter in the problem, a single $\boldsymbol{\Omega}(t_f)$ function is penalized, defined as the state sensitivity to thrust perturbations. Furthermore, thrust uncertainties predominantly affect the accuracy of the final state in the vertical direction due to the intensive control exerted in that direction to counteract the gravitational acceleration. Additionally, since the altitude (h) is the integral of the vertical velocity (v_h), penalizing its sensitivity to thrust positively affects the dispersion of $v_h(t_f)$ as well. This rationale justifies penalizing the third component of $\boldsymbol{\Omega}(t_f)$, as defined in Eq. (8), in the cost function. The use of the norm for this sensitivity component is not required since its value is always positive. According to the proposed $\boldsymbol{\Lambda}(t)$ formulation (see Sec. II.A), the final condition $\boldsymbol{\Lambda}_f$ is a vector whose elements in correspondence of the building rows are all equal to 1. However, for easiness of formulation, the $\boldsymbol{\Lambda}(t)$ definition is limited to some individual rows extracted from the $\boldsymbol{S}_2(t)$ matrix. Consequently, to derive the dynamics associated with the designated row for penalization, the terminal conditions outlined in Eq. (13) correspond to a vector comprising zeros, with a unique value equal to 1 positioned at the

selected row. For example, if the sensitivity of the final altitude (third state) is penalized, $\mathbf{\Lambda}_f$ of Eq. (13) is a seven element zero vector with a 1 in its third position. In case $n_\Lambda > 1$ sensitivities are penalized, n_Λ $\mathbf{\Lambda}(t)$ vectors are defined each with final conditions specified as above.

An important aspect to consider for this class of problems is related to the *bang-bang* nature of the thrust profile, which does not allow for a complete feedback action, since on *min* arcs it is not possible to reduce the thrust. Conversely, on *max* arcs no increases in the thrust are allowed. The DOC methodology handles saturation by introducing a factor $\eta(t)$ to scale down feedback gains near control bounds [41]. The parameter $\eta(t) \in [0, 1]$ is defined as,

$$\eta(t) = \frac{4\,(u^*(t) - u_{min})\,(u_{max} - u^*(t))}{(u_{max} - u_{min})^2} \tag{17}$$

where $u^*(t)$ refers to the the optimal control profile. The parameter $\eta(t)$ modifies Eq. (13) and Eq. (14), that respectively assume the following expressions.

$$\dot{\mathbf{\Lambda}}(t) = -\left(\boldsymbol{A}_R(t) + \eta(t)\boldsymbol{B}_R(t)\boldsymbol{K}(t)\right)^T \mathbf{\Lambda}(t), \quad \mathbf{\Lambda}(t_f) = \mathbf{\Lambda}_f \tag{18}$$

$$\dot{\mathbf{\Omega}}(t) = \left(\boldsymbol{A}_R(t) + \eta(t)\boldsymbol{B}_R(t)\boldsymbol{K}(t)\right) \mathbf{\Omega}(t) + \boldsymbol{C}_R(t), \quad \mathbf{\Omega}(t_0) = \mathbf{0} \tag{19}$$

with $\boldsymbol{A}_R(t)$, $\boldsymbol{B}_R(t)$ and $\boldsymbol{C}_R(t)$ matrices reported in Appendix. The feedback gains appearing in Eq. (18) and Eq. (19) are computed through Riccati matrix $\boldsymbol{P}(t) \in \mathbb{R}^{6\times6}$ from the differential Riccati Equation, with an appropriate choice of the state penalty matrix $\boldsymbol{Q}(t) \in \mathbb{R}^{6\times6}$ and the control penalty matrix $\boldsymbol{R}(t) \in \mathbb{R}^{3\times3}$:

$$-\dot{\boldsymbol{P}}(t) = \boldsymbol{P}(t)\bar{\boldsymbol{A}}_R(t) + \bar{\boldsymbol{A}}_R^T(t)\boldsymbol{P}(t) - \boldsymbol{P}(t)\bar{\boldsymbol{B}}_R(t)\boldsymbol{R}(t)^{-1}\bar{\boldsymbol{B}}_R(t)^T\boldsymbol{P}(t) + \boldsymbol{Q}(t), \quad \boldsymbol{P}(t_f) = \boldsymbol{Q}(t_f) \tag{20}$$

$$\bar{\boldsymbol{K}}(t) = -\boldsymbol{R}(t)^{-1}\bar{\boldsymbol{B}}_R(t)^T\boldsymbol{P}(t) \rightarrow \boldsymbol{K}(t) = \begin{bmatrix} \bar{\boldsymbol{K}}(t) & \mathbf{0}_{6\times1} \\ \mathbf{0}_{1\times6} & 0 \end{bmatrix} \tag{21}$$

In this specific application, no corrective feedback for mass error is applied (as it is assumed that no direct measurements of fuel consumption are available), resulting in a 3×6 feedback gain matrix. The $\bar{\boldsymbol{A}}_R(t) \in \mathbb{R}^{6\times6}$ and $\bar{\boldsymbol{B}}_R(t) \in \mathbb{R}^{6\times3}$ matrices in Eq. (20) equivalent the ones reported in Appendix with the removal of the last column and the last row from $\boldsymbol{A}_R$, and the last row from $\boldsymbol{B}_R$ and $\boldsymbol{C}_R$ to reflect the feedback gain matrix size reduction. Due to the symmetric nature of $\boldsymbol{P}(t)$ matrix, just its upper triangular elements integrated into the state and collected within the $\boldsymbol{P}_u \in \mathbb{R}^{21\times1}$ vector. The complete optimization problem cost function is the following.

$$\min_{\boldsymbol{u}} \; J_r = -m(t_f) + \alpha \left(\int_{t_0}^{t_f} \sum_{k=1}^{n_\Lambda} \alpha_k \|\mathbf{\Lambda}(t)_k\|_2 \, \mathrm{d}t + \beta\, \mathbf{\Omega}_3(t_f) \right) \tag{22}$$

to be optimized on the state vector:

$$\boldsymbol{x}_{DOC}(t) = \left[\, \boldsymbol{x}(t),\ \boldsymbol{\Lambda}_1(t),\ \ldots,\ \boldsymbol{\Lambda}_{n_\Lambda}(t),\ \boldsymbol{\Omega}(t),\ \boldsymbol{P}_u(t) \,\right] \tag{23}$$

with a maximum dimension $7 + 7\,n_\Lambda + 7 + 21$, depending on the number of penalized $\boldsymbol{\Lambda}(t)$ functions (see Eq.(11)) and on whether the sensitivity function $\boldsymbol{\Omega}(t)$ is included in the formulation, as discussed in the following sections.

Fast DOC Problem Formulation

Cost function		
	$\min_{\boldsymbol{u}}\ J_r = -m(t_f) + \alpha \left(\int_{t_0}^{t_f} \sum_{k=1}^{n_\Lambda} \alpha_k \lVert \boldsymbol{\Lambda}(t)_k \rVert_2 \,\mathrm{d}t + \beta\, \boldsymbol{\Omega}_3(t_f) \right)$	See Eq. (22)
State Vector		
	$\boldsymbol{x}_{DOC}(t) = \left[\, \boldsymbol{x}(t),\ \boldsymbol{\Lambda}_1(t),\ \ldots,\ \boldsymbol{\Lambda}_{n_\Lambda}(t),\ \boldsymbol{\Omega}(t),\ \boldsymbol{P}_u(t) \,\right]$	See Eq. (23)
Control Vector		
	$\boldsymbol{u}(t) = \left[\, u_x(t),\ u_y(t),\ u_h(t) \,\right]$	
Physical States Dynamics		
	$\dot{x} = v_x, \qquad \dot{v}_x = \dfrac{u_x\, T\, n_T\, \cos\gamma}{m} \qquad \dot{m} = -\dfrac{u\, T\, n_T}{I_{sp}\, g_0}$	
	$\dot{y} = v_y, \qquad \dot{v}_y = \dfrac{u_y\, T\, n_T\, \cos\gamma}{m},$	See Eq. (15)
	$\dot{h} = v_h, \qquad \dot{v}_h = \dfrac{u_h\, T\, n_T\, \cos\gamma}{m} - g_m,$	
DOC States Dynamics		
	$\dot{\boldsymbol{\Lambda}}(t) = -\left(\boldsymbol{A}_R(t) + \eta(t)\boldsymbol{B}_R(t)\boldsymbol{K}(t)\right)^T \boldsymbol{\Lambda}(t)$	See Eq. (18)
	$\dot{\boldsymbol{\Omega}}(t) = \left(\boldsymbol{A}_R(t) + \eta(t)\boldsymbol{B}_R(t)\boldsymbol{K}(t)\right) \boldsymbol{\Omega}(t) + \boldsymbol{C}_R(t)$	See Eq. (19)
Feedback Gains Dynamics		
	$-\dot{\boldsymbol{P}}(t) = \boldsymbol{P}(t)\bar{\boldsymbol{A}}_R(t) + \bar{\boldsymbol{A}}_R^T(t)\boldsymbol{P}(t) - \\ + \boldsymbol{P}(t)\bar{\boldsymbol{B}}_R(t)\boldsymbol{R}(t)^{-1}\bar{\boldsymbol{B}}_R(t)^T\boldsymbol{P}(t) + \boldsymbol{Q}(t)$	See Eq. (20)
	$\bar{\boldsymbol{K}}(t) = -\boldsymbol{R}(t)^{-1}\bar{\boldsymbol{B}}_R(t)^T\boldsymbol{P}(t) \rightarrow \boldsymbol{K}(t) = \begin{bmatrix} \bar{\boldsymbol{K}}(t) & \boldsymbol{0}_{6\times 1} \\ \boldsymbol{0}_{1\times 6} & 0 \end{bmatrix}$	See Eq. (21)
State constraints		
	$h(t) > 0, \qquad \forall t < t_f$	See Eq. (16)
Control constraints		
	$0 < u_{min} \le u = \sqrt{u_x^2 + u_y^2 + u_h^2} \le u_{max}$	See Eq. (16)
Feedbak Capability Factor		
	$\eta(t) = \dfrac{4\,(u^*(t) - u_{min})\,(u_{max} - u^*(t))}{(u_{max} - u_{min})^2}$	See Eq. (17)

The effectiveness of the introduction of the feedback capability factor $\eta(t)$ relies in making the two objectives in Eq. (22) competing. The fuel-optimal only trajectory leads to a almost-zero $\eta(t)$ shifting the closed-loop problem to

an open-loop one, with a huge growth of sensitivity. On the other hand, the minimization of the sensitivities requires $\eta(t)$ to be as large as possible, i.e., close to one, leading $u(t)$ to be at the half-way between u_{max} and u_{min} with a severe increase of fuel-mass consumption. The balance between these two extreme situations is managed through the coefficient α in Eq. (22). For small α values, the effectiveness of the DOC terms diminishes, as the solution is closer and closer to the *max-min-max* profile. However, an increase in the value of α improves performance in terms of sensitivities at the expense of a larger fuel consumption, as the original cost index is less penalized than in the case for which $\alpha = 0$. For the noDOC case, we set $\eta(t) = 1$ as the definition of Eq. (17) would lead to zero-feedback, i.e., open-loop solutions.

D. Implementation Strategies and Key Novelties

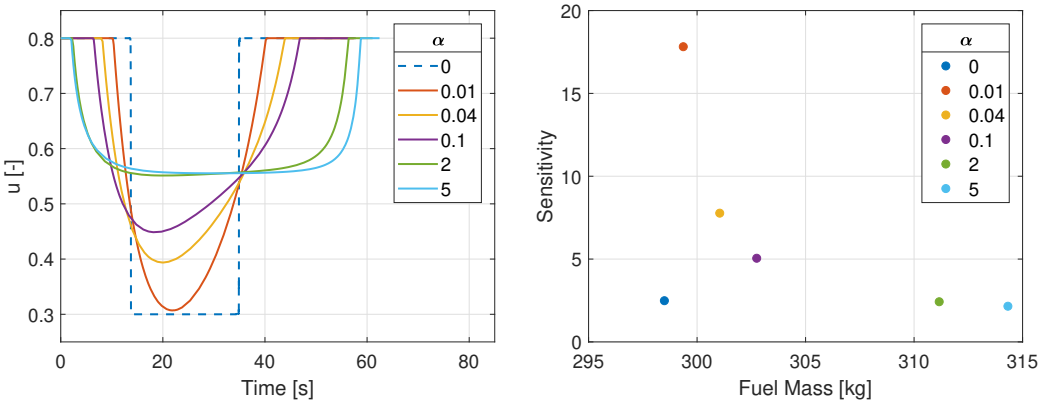
Even though the architecture is complete, some more points require to be addressed. This section focuses on the specific novelties we are proposing. First, we define the *a-priori performance index*. Second, we introduce two new ideas, namely, the *marginal DOC-coefficient*, and the concept of *dominant sensitivity*. Finally, we propose some relevant criteria for the definition of the Riccati Equation penalty matrices of Eq. (20). The effective design choices are reported in IV.

To account for uncertainties and disturbances, probabilistic approaches, such as Monte Carlo analysis, are typically utilized to assess the performance of the designed system. This approach can easily lead to large, time-consuming validation campaigns. We observed that an a-priori, synthetic performance index could be derived from the relationship between sensitivity and covariance matrices [50], where a reduced sensitivity produces a more compact final covariance hyper-ellipsoid. For the considered case, the sensitivity cost term is made of two contributes depending on the sensitivity matrix and the sensitivity function (Eq. (22)). However, the link between sensitivities and covariance matrices is just valid for the sensitivity matrix [50]. Therefore, this index is defined just for the case without penalization of the thrust parametric sensitivity, namely $\beta = 0$ in Eq. (22). Even though this might seem a strong restriction, in Sec. IV we will show that the thrust uncertainties have minor relevance with respect to the initial state errors, therefore the performance index computed in the $\beta = 0$ case provides good predictions even when thrust magnitude perturbations are considered.

After solving the optimal control problem, the $\boldsymbol{S}(t_f)$ matrix can be obtained offline through Eq. (3), properly modified to account for the feedback gains and the feedback capability factor. With $\boldsymbol{S}(t_f)$ comprising n^2 terms, a synthetic performance indicator for evaluating the efficiency of the solution is necessary. The element $\boldsymbol{S}_{i,j}(t_f)$ within this sensitivity matrix characterizes the sensitivity of the i^{th} state component with respect a perturbation in the j^{th}. Since anticipated perturbations in position are roughly one order of magnitude larger than those in velocity, an appropriate metric for assessing the standard deviation of the i^{th} state at final time is the a-priori performance index s_i, defined as follows.

$$s_i = \sqrt{\sum_{j=1}^{6} w_j^2 \, \boldsymbol{S}_{i,j}(t_f)^2} \qquad w_j = \begin{cases} 10, & \text{for } j = 1,2,3 \\ 1, & \text{for } j = 4,5,6 \end{cases} \tag{24}$$

A demonstrative example of the use of s_i is visible in Fig. 1. Here, a simplified cost function is defined by setting $\alpha_k = 1, \beta = 0$ in Eq. (22), with the penalization of a single sensitivity, $n_\Lambda = 1$, namely the $y(t_f)$ one. This results in $\mathbf{\Lambda}_f = [0, 1, 0, 0, 0, 0, 0]$ in Eq. (18). By varying the values of α in Eq. (22), different trajectories are generated. When $\alpha = 0$ the solution to the noDOC problem is obtained, i.e. only the fuel mass consumption is minimized with $\eta(t) = 1$, resembling the typical *bang-bang* control profile (Fig. 1a). For positive α values, the sensitivity term is penalized within the cost function and the control profiles is modified to allow for a non-zero feedback capability factor to reduce the sensitivity terms. This effect reaches its limit as $u(t)$ gets closer to the half-way between u_{min} and u_{max}. To provide a physical interpretation of the a-priori performance index, Fig. 1b displays the values of s_3 computed from the control profiles in Fig. 1a. This index predicts the final altitude dispersion, and its behavior is representative of other indices (e.g., s_2 or s_6), which exhibit similar trends. Variations of s_3 are compared with the fuel mass consumptions, as sensitivity and fuel mass reductions are competitive objectives (Sec. III.C). As α increases, fuel consumption increases relative to the fuel-optimal *bang-bang* profile, and a reduction in sensitivity would be expected due to the penalization of the sensitivity term. However, this is not observed for solutions with small α values. For instance, with $\alpha = 0.01$, s_3 is large due to the closeness of the corresponding $u(t)$ to the bounds in Fig. 1a, resulting in minimal feedback along the trajectory for the feedback capability factor effect. As α increases, the sensitivity is reduced with respect the noDOC solution due to the restoration of the feedback control, reaching its minimum values for $\alpha = 5$. It is important to note that performance predictions using s_i do not account for saturation effects, which can degrade actual performance relative to the predictions. However, control saturation presents an issue for the noDOC case, as it does not occur in DOC cases due to the inherent nature of the feedback capability factor, which reduces the gains as the control profile approaches the boundaries. The reported values are accurate predictions the DOC method performances in terms of fuel mass consumption mass and sensitivities, as validated by MC results in Sec. IV. In contrast, the predictions for the noDOC case are optimistic, indicating a greater potential for improvement with the proposed DOC strategy.



(a) DOC Control Profiles **(b) Sensitivity s_3 and fuel consumption**

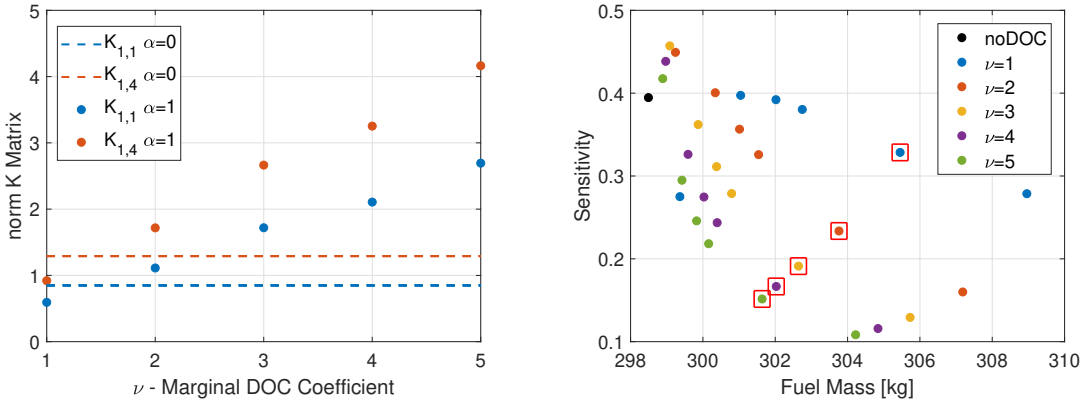
Fig. 1 Saturation Handling Strategy Effect on the control profiles and the sensitivity values.

The major issue addressed in this work is the limitations coming from $\eta(t) \in [0, 1]$, resulting in a reduction of feedback action compared to noDOC reference case. However, the key novelty we propose is the introduction of a positive scalar multiplicative factor ν, that we refer to as the *marginal DOC coefficient*, capable of restoring the original feedback control level without compromising other factors. The feedback capability factor is redefined as:

$$\tilde{\eta}(t) = \nu\, \eta(t) \tag{25}$$

and leads to the modification of Eq. (17), Eq. (18) and Eq. (19). In order to assess the impact of Eq. (25), an example with the same parameter setting of the previous paragraph is analyzed. Figure 2a reports the 2-norm of two representative elements of the feedback gain matrix in time, compared for the noDOC case and for a DOC case with $\alpha = 1$. While the effect of the feedback gains cannot be directly inferred from their norms, this figure provides insight into the working principle of the marginal DOC coefficient. For $\nu = 1$, i.e., no influence from the marginal coefficient, the DOC values remain below those of noDOC due to the effect of the feedback capability factor. However, for $\nu = 2$, the noDOC threshold is exceeded, and the feedback gains are restored in terms of their norm. As expected, larger values of ν result in higher norms of the feedback gains. As for Fig. 1b, the performance improvements achieved through the proposed solutions are analyzed in terms of sensitivity reduction relative to fuel mass consumption, with increasing α weights moving to the right of Fig. 2b. According to Eq. (24), s_4 is defined as a synthetic index representing the dispersion of $v_x(t_f)$, the fourth element of the state vector. Similar to the previous example, this index is reported without loss of generality, as other indices exhibit analogous behavior, as increments of α reduce sensitivity at the cost of higher fuel mass consumption. To illustrate the impact of the marginal DOC coefficient, the specific case of $\alpha = 0.3$ is taken as a reference. As ν increases, final sensitivities decrease, leading to an overall enhancement in the performance of the DOC procedure for the same α. This improvement is attributed to the increased feedback action, which allows for reduced fuel mass consumption while increasing the performance level, i.e., the same final sensitivity value. The underlying mechanism can be explained by noting that setting $\nu > 1$, has a significant impact on the control profiles. Specifically, maximum values of $\eta(t)$, i.e. close to one, are no longer necessary to reduce sensitivities, as the marginal DOC coefficient ν already enhances the influence of the feedback gains on sensitivities. Consequently, for a fixed α, the control profile moves closer to the boundaries as ν increases, resulting in both reduced mass consumption and lower sensitivities. As for the previous scenario, the performances worsening due to saturation limits applies just to the noDOC case and similar conclusions can be drawn.

Even though the increment of ν leads to advantages both in terms of sensitivities and fuel mass consumptions, an upper limit for its values has to be defined. The first limiting factor on $\tilde{\eta}(t)$ value is the Gain Margin [58], which denotes the maximum multiplicative factor for feedback gains that ensures system stability for linear systems. Given the Multiple Input Multiple Output (MIMO) nature of the system, which comprises 7 physical states and 3 inputs, ν is chosen to



(a) Feedback gains norms for different ν values **(b) Sensitivity s_4 and fuel consumption**

Fig. 2 Marginal DOC Coefficient impact on feedback matrix terms and on sensitivity and fuel mass trade-off.

remain within the minimum Gain Margin value associated with the linearized system. Even if the considered system is nonlinear, this criterion is considered applicable to its linearized approximation at any time, due to its quasi-linearity, and due to the assumed small state variations about the nominal one. Please, note that, despite not formally holding, due to the presence of the nonlinearity, the suggested criterion can give a reasonable guideline about what Gain Margin to expect from the linearized version of the same system, which is a useful indication for its subsequent verification and validation, although we highlight that for the general nonlinear case eigenvalues with negative real part do not automatically guarantee stability [59]. Conversely, the second criterion focuses on the risk associated with excessively large ν values, which can lead to sharp variations in $\tilde{\eta}(t)$ near the control bounds. This scenario poses a risk of breaching the control bounds due to the heightened feedback gains, contradicting the intended purpose of mitigating saturation through $\tilde{\eta}(t)$. In practice, the first criterion tends to be more restrictive due to the stability properties of the system.

When considering LQR tuning within the DOC scenario, the process shifts from being partially guided by trade-offs in terms of control effort and state errors (as in common applications), towards a fully sensitivity-driven approach, given the efficient saturation handling strategy. Tuning is also challenging due to the integrated control framework, that influences performances and CPU time of the solver when extremely small or large entries for the $\boldsymbol{Q}(t)$ or $\boldsymbol{R}(t)$ matrices are chosen. Both the $\boldsymbol{Q}(t)$ and $\boldsymbol{R}(t)$ matrices are set as diagonal to streamline the tuning and $\boldsymbol{Q}(t)$ is constrained to be positive-definite to penalize the state error at any instant, differently from previous works [41]. Due to the symmetry of the problem, all diagonal elements of $\boldsymbol{R}(t)$ are configured to have identical values. In the context of the Fast DOC we are proposing, where saturation is not an issue, the $\boldsymbol{R}(t)$ matrix values could theoretically be reduced indefinitely to achieve optimal disturbance management. However, practical applications introduce various sources of noise, stemming from factors such as sensors and model parameters, thereby introducing uncertainties. Large gains might lead to compromised noise rejection capabilities. Although these factors are not explicitly addressed in this study, reasonable bounds for $\boldsymbol{R}(t)$ values are considered. In this specific scenario, the lower bound for the values of $\boldsymbol{R}(t)$ is linked to the intrinsic behavior

of the LQR feedback gains, which tend to increase during the final stages. This growth becomes more pronounced when small values of $\boldsymbol{R}(t)$ are used. Setting $\boldsymbol{R}(t)$ too low introduces sharp discontinuities in the DOC problem formulation, resulting in a significant increase in CPU time and, in some cases, the failure of NLP convergence. A possible mitigation is the increment of $\boldsymbol{R}(t)$ along the trajectory, not viable with the adopted off-the-shelf software [60] due to the significant rise in CPU time. As a result, $\boldsymbol{R}$ is set as a constant with sufficiently large values. Similar issues arise with time-varying $\boldsymbol{Q}(t)$, which are addressed by adopting constant state error weights $\boldsymbol{Q}$, dropping the time-dependency from now. Given the significant level of symmetry of the problem, using the same weights for different components results in similar sensitivity patterns for positions and velocities, with minor variations coming from distinct initial conditions. However, when individual weights change, different behaviors emerge, leading to improvements in one directions while worsening others, as shown in [61].

We saw in Sec. III.B that a standard desensitization technique, i.e., penalizing the sensitivity of each state final state, results in 49 additional state variables, as they are given by $7, n_\Lambda$ with $n_\Lambda = 7$. Therefore, the objective is to identify the minimum number of final states to be desensitized that can yield satisfactory results in terms of sensitivity of the entire final state, also acknowledging that not all design choices lead to well-behaved or converging solutions. Clearly, the complete opposite approach would be to adopt a *single-state* desensitization, representing the most computationally efficient solution as the state of Eq. (23) stores a single $\boldsymbol{\Lambda}(t)$ function, although this approach can seem too reductive. However, for the specific problem at hand this is not entirely true. Due to the inherent nature of the feedback capability factor, the penalization of a single sensitivity term in Eq. (22), i.e., $n_\Lambda = 1$, influences $\eta(t)$. As the weight α increases, $\eta(t)$ is maximized, which in turn positively impacts the reduction of all sensitivities, as the feedback effect becomes more relevant across all directions. Consequently, penalizing one sensitivity effectively reduces the others, due to feedback control restoration.

Another noteworthy outcome emerges for the considered problem. Consider, for instance, the three functions $\boldsymbol{\Lambda}(t)$ associated with the three final positions, i.e., $\boldsymbol{\Lambda}_1(t)$, $\boldsymbol{\Lambda}_2(t)$, and $\boldsymbol{\Lambda}_3(t)$. Their dynamics follow Eq. (18) but differ only in the final condition vectors $\boldsymbol{\Lambda}_f$, which are zero vectors except for a one in the first, second, or third position, respectively. Furthermore, the matrices of the linearized system $\boldsymbol{A}_R(t)$ and $\boldsymbol{B}_R(t)$ exhibit a structure that is essentially invariant across the three directions. Along with the assumption of a diagonal $\boldsymbol{K}(t)$ matrix, the dynamics of the considered sensitivity vectors are similar, differing only by a permutation of the rows of $\dot{\boldsymbol{\Lambda}}_k(t)$ for $k = 1, 2, 3$, with minor differences arising from variations in the control components and feedback gains. Therefore, if $\boldsymbol{\Lambda}_1(t)$ (or any of the other position sensitivities) is penalized in the cost function, its dynamics will change in a way that affects the dynamics of $\boldsymbol{\Lambda}_2(t)$ and $\boldsymbol{\Lambda}_3(t)$, leading to similar decreasing trends as α in Eq. (22) increases. This behavior is highlighted in the results (see Sec. IV), and the same pattern is observed for the velocity-related sensitivities, i.e., $\boldsymbol{\Lambda}_4(t)$, $\boldsymbol{\Lambda}_5(t)$, and $\boldsymbol{\Lambda}_6(t)$. Due to this similar trends, two major options emerge: penalizing a final position sensitivity or a final velocity one. By inspecting the performance index defined in Eq. (24) we observed that penalizing a final position sensitivity leads to smoother

convergence properties and aligns well with the design of the controller, differently from what we observed when final velocity sensitivities were penalized. Since positions are the integrals of their corresponding velocities, minimizing the sensitivity of a position naturally leads to a reduction in the sensitivity of the corresponding velocity. In other words it turned out that the most efficient strategy is to penalize the highest signals in the kinematic chain of integrals representing the equations of motion. Once that the decision to penalize the positions is taken, three options arise: desensitizing the final x, the final y, or the final h. For the considered application and the initial conditions specified in Tab. 2, experimental observations indicate that desensitizing $y(t_f)$ generally yields slightly better overall performance compared to $x(t)$ and $h(t)$ desensitization. However, penalizing the sensitivities of $x(t_f)$ or $h(t_f)$ does not result in a significant performance decrement compared to desensitizing $y(t_f)$, with different outcomes observed under varying initial conditions. Consequently, the concept of *dominant sensitivity* emerges from single-state desensitization, denoting the single component of the final state that requires desensitization to achieve optimal performance. The flexibility of Fast DOC method allows for the exploration of a *multiple-states* desensitization strategy through evaluations of the performance index for each case. One alternative strategy, that can be seen as a trade-off between the standard DOC approach and the single-state desensitization entails penalizing another position sensitivity alongside the dominant one. Yet, when multiple final positions are desensitized, the overall sensitivity falls between the sensitivities of the individual positions when desensitized separately. Therefore, once the dominant sensitivity is chosen, the desensitization of further position components does not necessarily enhance performance due to the coupling among the states. Another strategy involves penalizing multiple velocity sensitivities concurrently with the dominant position state. Since the dynamics of velocities and positions differ, this approach could potentially offer further performance improvements. Various combinations of relative weights and different components have been tested. However, desensitizing multiple velocities, together with a position component, proved to be ineffective. Consequently, we opted for penalizing just a single velocity sensitivity alongside the dominant position one. Thus, desensitizing v_h along with y yields the most effective results for this configuration. Note, however that this last choice brought a marginal improvement, since the most effective enhancement was the discovery of the dominant sensitivity behavior for the problem under analysis.

Finally, in addition to terminal state desensitizations, the architecture proposed here supports *thrust desensitization* as well, achieved by expanding the state with the seven components of $\mathbf{\Omega}(t)$, as in Eq. (22). Due to the distinct nature of perturbations (thrust and initial conditions), exact predictions regarding the final standard deviation are not possible through the defined performance indexes of Eq. (24). Consequently, an accurate performance analysis resulting from this design choice is evaluated through a Monte Carlo analysis.

IV. Simulations and Results

For the generation of the numerical results presented in this section, we utilized an off-the-shelf pseudospectral transcription software leveraging Radau collocation [60], exploiting the automatic differentiation via the Adigator

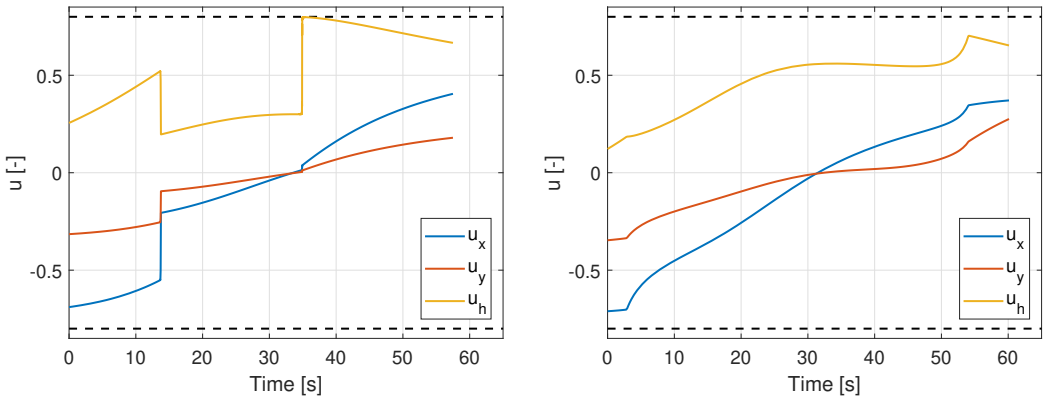
tool [62] to enhance convergence speed. For addressing the Nonlinear Programming (NLP) problem, IPOPT [63] is selected as NLP solver. We evaluated the performance of the proposed method through several sets of Monte Carlo (MC) simulations including states and thrust perturbations. For the states we considered Gaussian initial position perturbations of 100 m and initial velocity perturbations of 10 m s^{-1} (3σ). The thrust was perturbed by including a bias and a time-varying noise, both equivalent to 2% (3σ) of the maximum nominal thrust value. All simulations entail sampling the nominal control profiles while keeping the final time t_f at its nominal value. Each Monte Carlo campaign was made of 1500 individual runs. To ensure a fair comparison of the techniques, for the noDOC we set slightly smaller and larger bounds for maximum and minimum values adopted to compute the reference trajectory, thus allowing space for feedback as in both directions. Specifically, the bounds for the feedforward solution in the noDOC case are restricted to 0.4-0.7 to provide a 0.1 throttle margin for feedback, offering a competing benchmark for the DOC strategy, where the bounds remained equal to 0.3 and 0.8. With this choice both the noDOC and DOC strategies can effectively exploit the same thrust range when the feedforward and the feedback terms are combined.

As a guiding principle, the tuning process for closed-loop solutions has been designed to ensure that the final horizontal position error remains within 10 meters (with a 3σ deviation), limit the vertical position error to 1 meter, restrict the final horizontal velocity error to 2 meters per second, and maintain the vertical velocity error at 1 meter per second. Identical LQR weights have been considered for both the noDOC and desensitized solutions (DOC solutions) to offer a valid performance comparison. First, we show the results obtained with the single-state desensitization, followed by the multiple-states desensitization analysis (specifically, with only two states desensitized, as described at the end of Sec. III.D). Then, we show the results obtained when thrust uncertainties are considered. Finally, we analyze the CPU time obtained with our approach compared with the state-of-the-art DOC techniques.

A. Single State Desensitization

In the proposed design, the marginal DOC coefficient ν, defined in Eq. (25), is set equal to 2 to be in line with the classical 6 dB value of the Gain Margin. Additionally, the matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$ are defined as $\boldsymbol{R} = \text{diag}[2.5,\ 2.5,\ 2.5]$ and $\boldsymbol{Q} = \text{diag}[0.63,\ 0.63,\ 9.49,\ 100,\ 100,\ 50]$. While the orders of magnitude of the components in the $\boldsymbol{Q}$ matrix resembles the design choices in [41], specific adjustments have been implemented. Notably, we increased the altitude-related weight, and decreased the element corresponding to the vertical velocity, since a larger gain on altitude already impacts vertical velocity errors. Furthermore, as previously emphasized, we only penalized the dominant sensitivity $\boldsymbol{\Lambda}_2(t)$ associated to $y(t_f)$, obtained by setting $\boldsymbol{\Lambda}_f$ of Eq.(18) as a zero vector with a 1 in the second component. This design choice implies setting $\alpha_2 = 1$, and $\beta = 0$ in Eq. (22). To better illustrate the potential of the method, we present the control commands for the noDOC solution (Fig. 3a) and for the DOC solution with $\alpha = 2$ (Fig. 3b). The proposed methodology results in smoother control commands than the classical bang-bang solution, making it more compatible with the thrust variation limitations of rocket motors and, theoretically, better suited for integration with attitude control

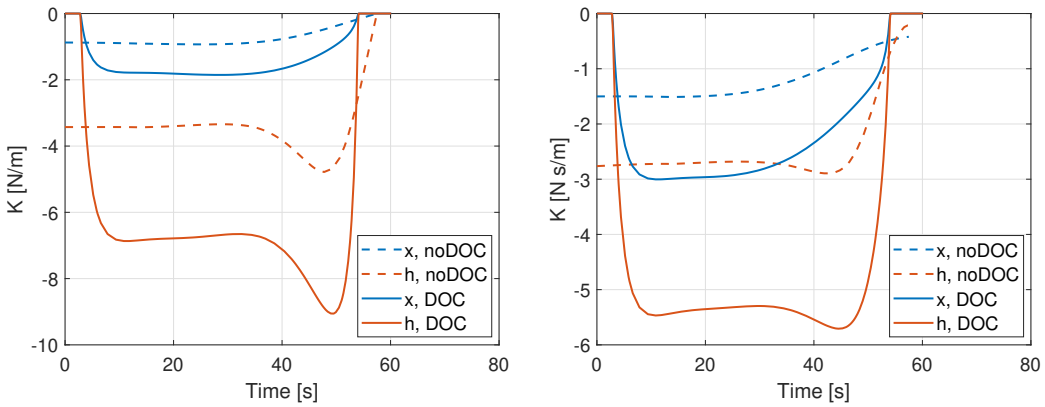
systems.



(a) Control Commands - noDOC Solution **(b) Control Commands - DOC Solution, $\alpha = 2$**

Fig. 3 Vertical and lateral control commands for noDOC and DOC ($\alpha = 2$) solutions.

The integrated control design, employing diagonal LQR weighting matrices (see Sec. III.D for details and rationale), results in a time-varying diagonal feedback gain matrix with six non-zero diagonal entries, as no feedback is applied to the mass error component, in accordance with Eq. (21). In the context of DOC solutions, the computed feedback gains are scaled by the feedback capability factor and the marginal DOC coefficient, yielding the results shown in Fig. 4. For ease of representation, only the gains in the $\hat{\boldsymbol{x}}$ direction are reported, as those in the $\hat{\boldsymbol{y}}$ direction are almost identical.



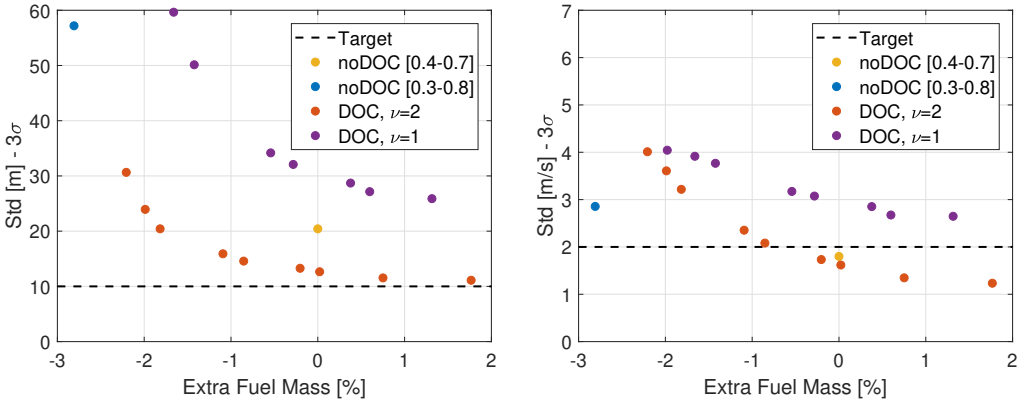
(a) Feedback gains for position errors **(b) Feedback gains for velocity errors**

Fig. 4 Optimal Feedback Gains for noDOC and DOC ($\alpha = 2$) solutions.

The adaptive nature of the proposed architecture can be further emphasized. Specifically, the feedback gains are higher in the intermediate phase of the trajectory, while they are reduced at the beginning and the end, where the available margin for feedback is inherently limited. Fundamentally, the method maximizes the range over which strong feedback can be applied to counteract errors and disturbances, while still respecting the actuator saturation that inevitably occurs at the trajectory boundaries. This behavior emerges as a consequence of the optimal structure of the solution. Moreover, an appropriate balance between proportional (position-related) and derivative (velocity-related) feedback gains has

been achieved across all configurations, ensuring satisfactory stability margins. As discussed earlier, the gain margin is sufficiently large to allow the marginal DOC coefficient ν to exceed a value of 1 without compromising robustness.

The validity of the marginal DOC coefficient concept is assessed by comparing the final state dispersions shown in Fig. 5 and Fig. 6, corresponding respectively to $\nu = 2$ and $\nu = 1$ (which means, no marginal DOC coefficient in use). Given the high similarity between the results in terms of x and y only the plots for x are shown for brevity. In Figures 5a and 5b the 3-σ errors for x and v_x variables are depicted on the y axes, and the corresponding fuel consumption increment is shown on the x axis. We can see a trend of reduced error moving towards the right, as solutions are generated with higher α weights in Eq (22), showing that the presented DOC strategy proves to be effective in the horizontal direction.



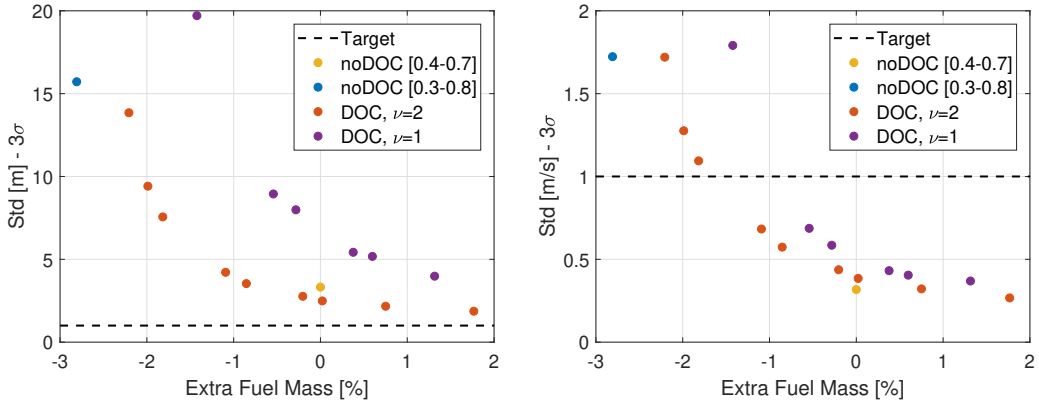
(a) 3σ **standard deviation of** $x(t_f)$ **(b)** 3σ **standard deviation of** $v_x(t_f)$

Fig. 5 MC 3σ **dispersion for** $x(t_f)$ **and** $v_x(t_f)$ **with single-state desensitization.**

The marginal DOC coefficient plays a significant role, demonstrating a positive impact on the components of the final state, diminishing standard deviations as ν values increase. Generally, we observed that the use of DOC effectively reduces the dispersion of final states compared to the noDOC scenario while maintaining similar fuel consumption, or even enhancing performance with slightly higher fuel consumption. Potential performance enhancements could be achieved with further increments in the weight factor α. Nonetheless, it is worth noting that the results are approaching a performance limit, and achieving significant additional enhancements may become challenging. Finally, note that the MC results for this case, as well as for all the other ones presented in this section, confirmed the validity of the linear control technique designed, since that the overall closed-loop nonlinear system was stable for all the cases analyzed.

Figures 6a and 6b show the performance in the vertical direction, highlighting dispersion reduction and fuel mass increase. The improvements observed in horizontal positions (Fig. 5a) extend to the $\hat{\boldsymbol{h}}$ direction, with final values deviating by less than one meter from the 1-meter (3σ) requirement. Despite differences in the assumed values, the errors in both $\hat{\boldsymbol{x}}$ (or $\hat{\boldsymbol{y}}$) and $\hat{\boldsymbol{h}}$ directions follow similar trends, confirming the experimental validation of the sensitivity analyses underlying the dominant sensitivity concept (see Sec. III.D). Figure 6b presents another key finding: while

DOC generally improves the final state, it does not reduce the vertical velocity dispersion, despite maintaining the same fuel mass. However, a 0.8% increase in fuel consumption allows for a trajectory that restores initial performance. Further increments of α in the cost function lead to even better dispersion results.

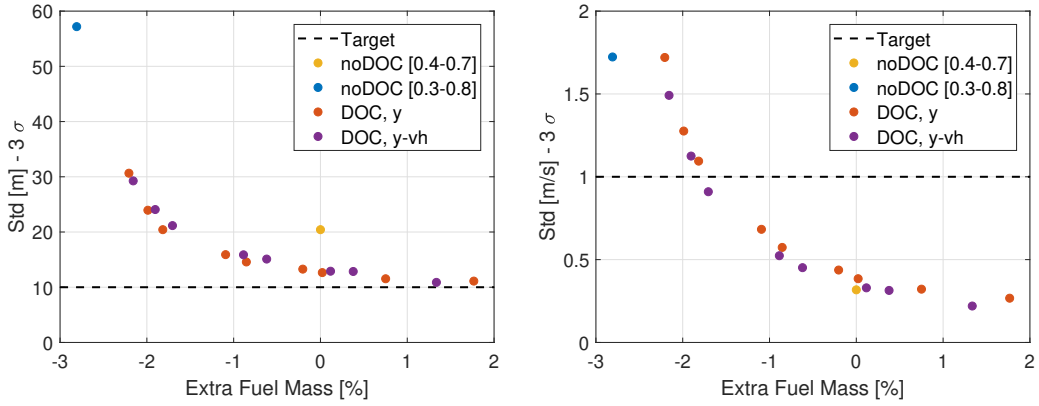


(a) 3σ **standard deviation of** $h(t_f)$ **(b)** 3σ **standard deviation of** $v_h(t_f)$

Fig. 6 MC 3σ **dispersion for** $h(t_f)$ **and** $v_h(t_f)$ **with single-state desensitization.**

B. Multiple-States Desensitization

The most effective design choice for desensitizing multiple-states is to penalize the sensitivities of y and v_h (see Sec. III.D) with weights of $\alpha_1 = 100$ and $\alpha_2 = 10$, respectively, as per Eq. (22). However, as depicted in Fig. 7, comparing the 3σ accuracy of $x(t_f)$ and $v_h(t_f)$ for single (y) and multiple-states ($y - v_h$) desensitization, this design approach proves inefficient performance improvements, despite considering a 7 extra state variables compared to the single desensitization case. Similar trends appear for $h(t_f)$ and $v_x(t_f)$ errors. Thus, the concept of dominant sensitivity is shown to be highly relevant, offering, at least for this scenario, the best performance with minimal computational effort.

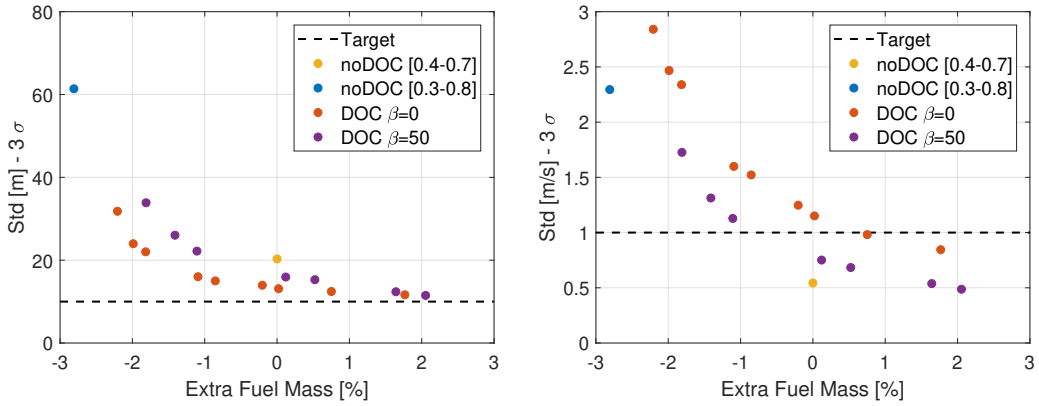


(a) 3σ **standard deviation of** $x(t_f)$ **(b)** 3σ **standard deviation of** $v_h(t_f)$

Fig. 7 MC 3σ **dispersion for** $x(t_f)$ **and** $v_h(t_f)$ **with single- and multiple-states desensitization.**

C. Thrust Uncertainties

The selected design approach for thrust sensitivity involves assigning a weight of $\beta = 50$ in Eq. (22) to penalize the third component of $\mathbf{\Omega}(t_f)$, while simultaneously penalizing the $y(t_f)$ sensitivity. The influence of thrust desensitization on the final vertical velocity is clearly depicted in Fig. 8b. In scenarios where thrust perturbations occur, solutions where thrust sensitivity is penalized, i.e. $\beta = 50$, demonstrate superior performance in the vertical direction compared to the standard single-state desensitization case with $\beta = 0$. Although the performance improvement may not be significant, this observation validates the design approach. Conversely, the degradation in performance in the $\hat{\boldsymbol{x}}$ direction (Fig 8a) is explained by (22): at a given α, the increment of β reduces the importance of the $\mathbf{\Lambda}(t)$ integral penalization, increasing the sensitivities of states to perturbations along the $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ directions.



(a) 3σ **standard deviation of** $x(t_f)$ **(b)** 3σ **standard deviation of** $v_h(t_f)$

Fig. 8 MC 3σ **dispersion for** $x(t_f)$ **and** $v_h(t_f)$ **with thrust desensitization.**

D. CPU Time Analysis

To assess the validity of the Fast DOC architecture, the CPU times required by the four strategies presented in Sec. II.A are compared when applied to the single-state desensitization presented in Sec. IV.A. In addition to the Thrust sensitivity states (Eq. (19)) and the 21 components of the upper part of the Riccati matrix,(Eq. (20)), both the propagation of the standard $\boldsymbol{S}(t)$ matrix (dubbed Standard DOC, Eq. (2)), and the alternative propagation of $\boldsymbol{S}^{-1}(t)$ (Alternative DOC) require for the problem under analysis 49 states to be defined and 49 extra constant states for final / initial conditions definition. The method using $\boldsymbol{S}_2(t)$ matrix requires less state variables due to the trivial final conditions definition (Efficient DOC, Eq. (5)), while $\mathbf{\Lambda}(t)$ propagation according to Fast DOC formulation requires just 7 extra states in the dominant-sensitivity formulation.

The CPU times reported in Tab. 3 represent the mean values obtained from 15 individual runs, conducted on a machine equipped with an Intel 12th Gen Core™ i5-1250P processor, and setting a tolerance for the NLP solver equal to 10^{-6}. To maintain generality, each test is based on the same premise: the computation of the noDOC trajectory

is initiated by setting $\alpha = 0$ and subsequently utilized as an initial estimate for DOC solutions with $\alpha = 0.01$ due to convergence issues of both the Standard and Alternative DOC techniques when larger values of α are defined. Since the analysis focuses just on state sensitivity, the thrust sensitivity components are not propagated and $\beta = 0$ in Eq. (22). Prior to a comprehensive analysis of the implementation strategies, it is notable that the computation of the noDOC solution only requires 0.33 seconds when the configuration of the nonlinear transcription method is optimized.

Table 3 Comparison of the different DOC techniques in terms of states and CPU time.

Technique	Sensitivity in use	Number of States Required	CPU Time, s
Standard DOC	$\boldsymbol{S}(t)$	49 + 49	811.44
Alternative DOC	$\boldsymbol{S}^{-1}(t)$	49 + 49	997.90
Efficient DOC	$\boldsymbol{S}_2(t)$	49	160.31
Fast DOC	$\boldsymbol{\Lambda}(t)$	7	25.24

The implementation based on Fast DOC formulation requires the fewest additional variables. The first two methods necessitate longer CPU times due to the presence of 49 extra variables required to set the final conditions of $\boldsymbol{S}(t)$ or $\boldsymbol{S}^{-1}(t)$ propagation. It is noteworthy that the Alternative DOC was initially expected to outperform the Standard DOC, as the $\boldsymbol{S}(t)$ matrix is inverted only once at the final time. However, contrary to expectations based on analytical results, the actual outcome in the context of the numerical implementation contradicts this expectation. While the precise reasons for this discrepancy are not entirely clear, it appears to be related to the conditioning number of the problem, and would therefore require further investigation and modification. The Fast DOC architecture reduces the CPU time by a factor of 30 compared to the commonly used $\boldsymbol{S}(t)$-based formulation and outperforms the Efficient DOC formulation due to the effective exploitation of the dominant sensitivity, specifically, the propagation and penalization of the second row of $\boldsymbol{S}_2(t)$. This reduced-order formulation operates six times faster than the Efficient DOC approach, supporting both the architecture design and the concept of dominant sensitivity. While the proposed scheme outperforms the implemented DOC strategies, there remains a notable gap in computational speed when compared to the most advanced state-of-the-art methods, which enable efficient computation of optimal guidance in reduced time [6, 64].

V. Conclusions

The formulation of the revisited DOC theory with a general-purpose transcription method for the examined rocket landing problem has proven to be successful, demonstrating both computational efficiency and performance improvements with respect to the corresponding state-of-the-art DOC architectures. Although the process inherently involves complexity, we could successfully implement the methodology by employing a convenient representation of sensitivity terms, identified as $\boldsymbol{\Lambda}(t)$ and $\boldsymbol{\Omega}(t)$, and by determining an optimal approach for computing the feedback gains using the LQR method. Some limitations remain, such as the constant setting of the LQR matrix $\boldsymbol{R}$ throughout the trajectory, yet the outcomes proved to be satisfactory. This formulation encompasses all the pertinent aspects related to

Desensitized Optimal Control and can be expanded to address various other problems.

Some insight into the tuning process for the analyzed problem has been provided, even if it appears to be case-dependent, raising questions about its applicability to entirely different scenarios within the same problem domain (e.g., different initial conditions). We thoroughly examined the role of the feedback capability factor $\eta(t)$, and introduced the simple, yet effective concept of marginal DOC coefficient. Furthermore, we analyzed and characterized the idea of dominant sensitivity, along with clarification of the impact of LQR weighing matrices on the optimal solution. Although various desensitization strategies have been considered, they were found to be less efficient compared to the dominant sensitivity-based approach. The resulting design surpasses the noDOC approach, offering some flexibility in trajectory design choices by varying α, a synthetic factor trading-off desensitization and original performance index of the non-desensitized problem. We observed a significant reduction in the CPU time required to compute solutions of the standard DOC formulation without the need to take advantage of computational shortcuts offered by the special structure of DOC problems, such as the analytical integration of constant states or the integration of sensitivity states outside the collocation scheme. Although this aspect can seem marginal, since it deals with off-nominal trajectory computation, it can be relevant when a large number of design options have to be rapidly evaluated. While DOC requires more CPU time than noDOC cases, in part due to the increased problem nonlinearity, but even more so because of the increase in the number of states, the advantages of the Fast DOC architecture could be more pronounced in complex rocket landing models (e.g., 6-DOFs), as the number of variables scales with n_x, compared to the $2n_x^2$ extra states required by standard DOC approaches. Finally, it is important to highlight that, despite the seemingly straightforward nature of the concepts presented, the tuning and the successful execution of the algorithm present challenges, and the level of difficulty could significantly increase for problems with stronger nonlinearities or more complex dynamical systems.

Appendix

$\boldsymbol{A}_R(t) \in \mathbb{R}^{7\times7}$, $\boldsymbol{B}_R(t) \in \mathbb{R}^{7\times3}$ and $\boldsymbol{C}_R(t) \in \mathbb{R}^{7\times1}$ non-zero elements are reported. Time dependence drops in favor of a simpler notation and λ and τ are defined as:

$$\lambda = n_T\, T\, \cos\gamma \qquad \tau = \frac{n_T\, T}{I_{sp}\, g_0}$$

$$\boldsymbol{A}_R(1,4) = 1, \quad \boldsymbol{A}_R(2,5) = 1, \quad \boldsymbol{A}_R(3,6) = 1, \quad \boldsymbol{A}_R(4,7) = -\lambda\frac{u_x}{m^2}, \quad \boldsymbol{A}_R(5,7) = -\lambda\frac{u_y}{m^2}, \quad \boldsymbol{A}_R(6,7) = -\lambda\frac{u_h}{m^2}$$

$$\boldsymbol{B}_R(4,1) = \frac{\lambda}{m}, \quad \boldsymbol{B}_R(5,2) = \frac{\lambda}{m}, \quad \boldsymbol{B}_R(6,3) = \frac{\lambda}{m}, \quad \boldsymbol{B}_R(7,1) = -\tau\frac{u_x}{m}, \quad \boldsymbol{B}_R(7,2) = -\tau\frac{u_y}{m}, \quad \boldsymbol{B}_R(7,3) = -\tau\frac{u_h}{m}$$

$$\boldsymbol{C}_R(4,1) = \frac{u_x\, n_T\, \cos\gamma}{m}, \quad \boldsymbol{C}_R(5,1) = \frac{u_y\, n_T\, \cos\gamma}{m}, \quad \boldsymbol{C}_R(6,1) = \frac{u_h\, n_T\, \cos\gamma}{m}, \quad \boldsymbol{C}_R(7,1) = \frac{u\, n_T}{I_{sp}\, g_0}$$

References

[1] Blackmore, L., Açikmeşe, B., and Scharf, D. P., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171. https://doi.org/10.2514/1.47202.

[2] Szmuk, M., Acikmese, B., and Berning, A. W., "Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints," *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2016, pp. 1–16. https://doi.org/10.2514/6.2016-0378.

[3] Lu, P., "Propellant-Optimal Powered Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 4, 2018, pp. 813–826. https://doi.org/10.2514/1.G003243.

[4] Reynolds, T. P., Szmuk, M., Malyuta, D., Mesbahi, M., Açıkmeşe, B., and Carson, J. M., "Dual Quaternion-Based Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 9, 2020, pp. 1584–1599. https://doi.org/10.2514/1.G004536.

[5] Sagliano, M., Heidecker, A., Hernández, J. M., Farì, S., Schlotterer, M., Woicke, S., Seelbinder, D., and Dumont, E., "Onboard Guidance for Reusable Rockets: Aerodynamic Descent and Powered Landing," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics (AIAA), 2021. https://doi.org/10.2514/6.2021-0862.

[6] Sagliano, M., Seelbinder, D., Theil, S., and Lu, P., "Six-Degree-of-Freedom Rocket Landing Optimization via Augmented Convex–Concave Decomposition," *Journal of Guidance, Control, and Dynamics*, Vol. 47, No. 1, 2024, pp. 20–35. https://doi.org/10.2514/1.G007570.

[7] Tolker-Nielsen, T., "ExoMars 2016 Schiaparelli Anomaly Inquiry," Tech. rep., European Space Agency (ESA), 2017. URL https://sci.esa.int/documents/33431/35950/1567260317467-ESA_ExoMars_2016_Schiaparelli_Anomaly_Inquiry.pdf.

[8] Biswal M, M. K., and Annavarapu, R. N., "Report on the Loss of Vikram Lander of Chandrayaan 2 Mission," 2021.

[9] Sagliano, M., Ishimoto, S., Macés-Hernández, J. A., Seelbinder, D., and Dumont, E., "Guidance and Control Strategy for the CALLISTO Flight Experiment," *8th European Conference for Aeronautics and Aerospace Sciences*, 2019. https://doi.org/10.13009/EUCASS2019-284.

[10] Winsor, C., and Roy, R., "The application of specific optimal control to the design of desensitized model following control systems," *IEEE Transactions on Automatic Control*, Vol. 15, No. 3, 1970, pp. 326–333. https://doi.org/10.1109/TAC.1970.1099453.

[11] Navarro-Tapia, D., Marcos, A., and Bennani, S., "Envelope Extension via Adaptive Augmented Thrust Vector Control System," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 5, 2021, pp. 1044–1052. https://doi.org/10.2514/1.G005436.

[12] Sagliano, M., Hernández, J. A. M., Farì, S., Heidecker, A., Schlotterer, M., Woicke, S., Seelbinder, D., Krummen, S., and Dumont, E., "Unified-Loop Structured H-Infinity Control for Aerodynamic Steering of Reusable Rockets," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 5, 2023, pp. 815–837. https://doi.org/10.2514/1.G007077.

[13] Shahbazzadeh, M., Salehifar, H., and Richards, C. M., "New approach to non-fragile robust model predictive control for a class of nonlinear systems with constrained inputs," *International Journal of Dynamics and Control*, Vol. 13, No. 2, 2025. https://doi.org/10.1007/s40435-025-01587-9.

[14] Ganguly, S., and Chatterjee, D., "Explicit feedback synthesis driven by quasi-interpolation for nonlinear model predictive control," *IEEE Transactions on Automatic Control*, 2025, pp. 1–8. https://doi.org/10.1109/tac.2025.3538767.

[15] Raković, S. V., *Robust Model Predictive Control*, Springer International Publishing, 2021, pp. 1965–1975. https://doi.org/10.1007/978-1-4471-5102-9_2-2.

[16] Langson, W., Chryssochoos, I., Raković, S., and Mayne, D., "Robust model predictive control using tubes," *Automatica*, Vol. 40, No. 1, 2004, pp. 125–133. https://doi.org/https://doi.org/10.1016/j.automatica.2003.08.009.

[17] Mayne, D. Q., and Kerrigan, E. C., "TUBE-BASED ROBUST NONLINEAR MODEL PREDICTIVE CONTROL1," *IFAC Proceedings Volumes*, Vol. 40, No. 12, 2007, pp. 36–41. https://doi.org/https://doi.org/10.3182/20070822-3-ZA-2920.00006, 7th IFAC Symposium on Nonlinear Control Systems.

[18] Açıkmeşe, B., Carson III, J. M., and Bayard, D. S., "A robust model predictive control algorithm for incrementally conic uncertain/nonlinear systems," *International Journal of Robust and Nonlinear Control*, Vol. 21, No. 5, 2011, pp. 563–590. https://doi.org/https://doi.org/10.1002/rnc.1613.

[19] Subbayyan, R., Sarma, V. V. S., and Vaithilingam, M. C., "An approach for sensitivity-reduced design of linear regulators," *International Journal of Systems Science*, Vol. 9, No. 1, 1978, pp. 65–74. https://doi.org/10.1080/00207727808941679.

[20] Verde, C., and Frank, P., "Sensitivity reduction of the linear quadratic regulator by matrix modification," *International Journal of Control*, Vol. 48, 1988, pp. 211–223. https://doi.org/10.1080/00207178808906170.

[21] Zames, G., "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses," *IEEE Transactions on Automatic Control*, Vol. 26, No. 2, 1981, pp. 301–320. https://doi.org/10.1109/TAC.1981.1102603.

[22] Chisci, L., Rossiter, J., and Zappa, G., "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, Vol. 37, No. 7, 2001, pp. 1019–1028. https://doi.org/https://doi.org/10.1016/S0005-1098(01)00051-6.

[23] Kappen, H. J., "Linear Theory for Control of Nonlinear Stochastic Systems," *Physical review letters*, Vol. 95, 2005, pp. 200–201. https://doi.org/10.1103/PhysRevLett.95.200201.

[24] Ridderhof, J., and Tsiotras, P., "Uncertainty Quantication and Control During Mars Powered Descent and Landing using Covariance Steering," *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018. https://doi.org/10.2514/6.2018-0611.

[25] Ridderhof, J., and Tsiotras, P., "Minimum-fuel Powered Descent in the Presence of Random Disturbances," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. https://doi.org/10.2514/6.2019-0646.

[26] Benedikter, B., Zavoli, A., Wang, Z., Pizzurro, S., and Cavallini, E., "Convex Approach to Covariance Control with Application to Stochastic Low-Thrust Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 11, 2022, pp. 2061–2075. https://doi.org/10.2514/1.g006806.

[27] Benedikter, B., Zavoli, A., Wang, Z., Pizzurro, S., and Cavallini, E., "Convex approach to stochastic control for autonomous rocket pinpoint landing," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Charlotte, NC, USA, 2022, pp. 8–11.

[28] Benedikter, B., Zavoli, A., Wang, Z., Pizzurro, S., and Cavallini, E., "Convex Approach to Covariance Control for Low-Thrust Trajectory Optimization with Mass Uncertainty," *AIAA SCITECH 2023 Forum*, 2023. https://doi.org/10.2514/6.2023-2321.

[29] Ozaki, N., Campagnola, S., and Funase, R., "Tube Stochastic Optimal Control for Nonlinear Constrained Trajectory Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 645–655. https://doi.org/10.2514/1.G004363.

[30] Geller, D. K., "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1404–1414. https://doi.org/10.2514/1.19447.

[31] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 102–111. https://doi.org/10.2514/1.36834.

[32] Christensen, R. S., and Geller, D. K., "Closed-Loop Linear Covariance Analysis for Hosted Payloads," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 10, 2018, pp. 2133–2143. https://doi.org/10.2514/1.G003333.

[33] Reynolds, T. P., Malyuta, D., Mesbahi, M., Açikmese, B., and Carson, J. M., "Funnel Synthesis for the 6-DOF Powered Descent Guidance Problem," *AIAA Scitech 2021 Forum*, 2020. https://doi.org/https://doi.org/10.2514/6.2021-0504.

[34] Kim, T., Elango, P., Reynolds, T. P., Açıkmeşe, B., and Mesbahi, M., "Optimization-Based Constrained Funnel Synthesis for Systems With Lipschitz Nonlinearities via Numerical Optimal Control," *IEEE Control Systems Letters*, Vol. 7, 2023, pp. 2875–2880. https://doi.org/10.1109/LCSYS.2023.3290229.

[35] Seywald, H., and Kumar, R., "Desensitized Optimal Trajectories," *Proceedings of the 6th Annual AAS/AIAA Space Flight Mechanics Conference*, American Astronomical Society (AAS), 1996, pp. 96–107.

[36] Seywald, K., and Seywald, H., "Desensitized Optimal Control," *AIAA SciTech 2019 Forum*, 2019. https://doi.org/10.2514/6.2019-0651.

[37] Seywald, H., "Desensitized Optimal Trajectories With Control Constraints," *Proceedings of the 13th AAS/AIAA Space Flight Mechanics Meeting*, American Astronomical Society (AAS), 2003.

[38] Seywald, H., and Seywald, K. L., "Intrinsic Desensitized Optimal Control Progress Report 1," *2020 IEEE Aerospace Conference*, 2020, pp. 1–33. https://doi.org/10.1109/AERO47225.2020.9172693.

[39] Seywald, K., and Seywald, H., "Tail Reoptimization in Desensitized Optimal Control," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-0379.

[40] Shen, H., Seywald, H., and Powell, R., "Desensitizing the Pin-Point Landing Trajectory on Mars," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008. https://doi.org/10.2514/6.2008-6943.

[41] Shen, H., Seywald, H., and Powell, R. W., "Desensitizing the Minimum-Fuel Powered Descent For Mars Pinpoint Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010, pp. 108–115. https://doi.org/10.2514/1.44649.

[42] Makkapati, V. R., Ridderhof, J., Tsiotras, P., Hart, J., and van Bloemen Waanders, B., "Desensitized Trajectory Optimization for Hypersonic Vehicles," *2021 IEEE Aerospace Conference (50100)*, IEEE, 2021. https://doi.org/10.1109/aero50100.2021.9438511.

[43] Xu, H., and Cui, H., "Robust Trajectory Design Scheme under Uncertainties and Perturbations for Mars Entry Vehicle," *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 2015, pp. 762–766. https://doi.org/10.1109/CICT.2015.132.

[44] Li, S., and Peng, Y., "Mars entry trajectory optimization using DOC and DCNLP," *Advances in Space Research*, Vol. 47, No. 3, 2011, pp. 440–452. https://doi.org/https://doi.org/10.1016/j.asr.2010.09.005.

[45] Makkapati, V. R., Dor, M., and Tsiotras, P., "Trajectory Desensitization in Optimal Control Problems," *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 2478–2483. https://doi.org/10.1109/CDC.2018.8619577.

[46] Makkapati, V. R., Maity, D., Dor, M., and Tsiotras, P., "C-DOC: Co-State Desensitized Optimal Control," *2020 American Control Conference (ACC)*, 2019, pp. 1761–1766. URL https://api.semanticscholar.org/CorpusID:203610690.

[47] Ayyanathan, P. J., and Taheri, E., "Reduced Desensitization Formulation for Optimal Control Problems," *The Journal of the Astronautical Sciences*, Vol. 71, No. 2, 2024. https://doi.org/10.1007/s40295-024-00435-w.

[48] Kim, T., and Açıkmeşe, B., "Funnel Synthesis via LMI Copositivity Conditions for Nonlinear Systems," , 2024. https://doi.org/10.48550/ARXIV.2402.15629.

[49] Sheridan, O., and Açıkmeşe, B., "Robust Fuel Optimal Trajectory Planning and Feedback Control for Constrained Linear Systems Under State- and Control-Dependent Perturbations," *IEEE Control Systems Letters*, Vol. 9, 2025, pp. 68–73. https://doi.org/10.1109/lcsys.2025.3558533.

[50] Seywald, H., and Seywald, K. L., "Desensitized Optimal Control," *Journal of Guidance, Control, and Dynamics*, in press. https://doi.org/10.2514/1.G008284.

[51] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation and Control*, Taylor & Francis, 1975. https://doi.org/10.1201/9781315137667.

[52] Simplicio, P., Marcos, A., and Bennani, S., "New Control Functionalities for Launcher Load Relief in Ascent and Descent Flight," *Proceedings of the 8th European Conference for Aeronautics and Space Sciences. Madrid, Spain, 1-4 july 2019*, 2019. https://doi.org/10.13009/EUCASS2019-275.

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

[54] Li, M., She, J., Zhang, C.-K., Liu, Z.-T., Wu, M., and Ohyama, Y., "Active disturbance rejection for time-varying state-delay systems based on equivalent-input-disturbance approach," *ISA Transactions*, Vol. 108, 2021, pp. 69–77. https://doi.org/https://doi.org/10.1016/j.isatra.2020.09.001.

[55] Xing, Y., He, X., and Li, X., "Lyapunov conditions for finite-time stability of disturbed nonlinear impulsive systems," *Applied Mathematics and Computation*, Vol. 440, 2023, p. 127668. https://doi.org/https://doi.org/10.1016/j.amc.2022.127668.

[56] Chen, W., Xu, S., Li, Y., and Zhang, Z., "Stability analysis of neutral systems with mixed interval time-varying delays and nonlinear disturbances," *Journal of the Franklin Institute*, Vol. 357, No. 6, 2020, pp. 3721–3740. https://doi.org/https://doi.org/10.1016/j.jfranklin.2020.02.038.

[57] Hu, J., Zhang, H., and Wang, Z., "Hybrid Adaptive Control of Spacecraft Attitude with Input Saturation and External Disturbance," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 3, 2019, pp. 642–649. https://doi.org/10.2514/1.G003090.

[58] Skogestad, S., *Multivariable feedback control*, Wiley, 1996.

[59] Mozelli, L. A., and Palhares, R. M., "Stability analysis of linear time-varying systems: Improving conditions by adding more information about parameter variation," *Systems & Control Letters*, Vol. 60, No. 5, 2011, pp. 338–343. https://doi.org/https://doi.org/10.1016/j.sysconle.2011.02.010.

[60] Rao, A. V., Benson, D. A., Darby, C., Patterson, M. A., Francolin, C., Sanders, I., and Huntington, G. T., "Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Trans. Math. Softw.*, Vol. 37, No. 2, 2010. https://doi.org/10.1145/1731022.1731032.

[61] Robbiani, T., Sagliano, M., Topputo, F., and Seywald, H., "Fast Desensitized Optimal Control for Rocket Powered Descent and Landing," *AIAA SCITECH 2024 Forum*, 2024. https://doi.org/10.2514/6.2024-0096.

[62] Weinstein, M. J., and Rao, A. V., "Algorithm 984: ADiGator, a Toolbox for the Algorithmic Differentiation of Mathematical Functions in MATLAB Using Source Transformation via Operator Overloading," *ACM Trans. Math. Softw.*, Vol. 44, No. 2, 2017. https://doi.org/10.1145/3104990.

[63] Wächter, A., and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *ACM Trans. Math. Softw.*, Vol. 106, 2006. https://doi.org/10.1007/s10107-004-0559-y.

[64] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E., "Autonomous Upper Stage Guidance Using Convex Optimization and Model Predictive Control," *ASCEND 2020*, American Institute of Aeronautics and Astronautics, 2020. https://doi.org/10.2514/6.2020-4268.