Warm-Start of Interior-Point Methods Applied to Sequential Convex Programming

Andrea Carlo Morelli, Christian Hofmann, Francesco Topputo

Abstract

Sequential convex programming (SCP) is an iterative technique that solves nonconvex optimization problems by considering a sequence of convex subproblems whose solutions eventually converge, under certain conditions, to the solution of the original problem. The most common convex solvers divide in interior-point, first-order, and active-set methods. In particular, the former have shown good performance. Even though SCP is quite efficient in terms of required computational time with respect to standard nonlinear optimization solvers, its potential is still not completely exploited when interior-point convex solvers are used as no information on the solution of a previous convex subproblem is used to construct the initial guess for the strictly-related following one. This is because interior-point methods are notoriously difficult to *warm-start*. In this paper, a technique to warm-start interior-point methods that solve second-order cone programs (SOCP) is developed and integrated within the sequential convex programming. The strategy can be used independently of the specific problem as long as it is expressed as a standard SOCP. The low-thrust space trajectory optimization problem is considered to assess the efficacy of the proposed strategy through extensive numerical simulations. It is shown that the warm-start algorithm outperforms the standard algorithm in terms of computational time and overall convergence when the widely-used solver ECOS is used.

A. C. Morelli is a PhD Student at the Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20158 Milan, Italy, E-mail: (andreacarlo.morelli@polimi.it, christian.hofmann@polimi.it).

C. Hofmann is a PhD Graduate at the Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20158 Milan, Italy, E-mail: (andreacarlo.morelli@polimi.it, christian.hofmann@polimi.it).

F. Topputo is Full Professor at the Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20158 Milan, Italy, E-mail: francesco.topputo@polimi.it.

This research is part of EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864697).

I. Introduction

Technological advances have allowed humanity to design systems that perform complex tasks like never before. Increasing autonomy of such systems represents a desirable goal to have a more reliable, less error-prone, and more accurate execution of their tasks. In some cases, when fast decisions need to be made, autonomy even becomes indispensable. This is the case, for instance, for the landing phase of Space X's Falcon 9 first stage[1], where real-time commands must be put into effect to react to the external, unpredictable perturbations that influence the trajectory of the launcher. However, executing online tasks poses several challenges. When dealing with real-time trajectory generation problems, the requirements for a suitable algorithm include reliability (i.e., the capability of finding a feasible solution whenever necessary) and computational affordability, i.e., the capability of converging in little time. Convex optimization [1] represents a promising approach for real-time trajectory optimization due to its convergence properties and reduced computational time required [2]. Its applications include the design and analysis of communication systems and signal processing [3], optimal guidance problems in the aerospace sector [4], composite structures optimization [5], and various motion planning problems [6].

Unfortunately, most real-life problems are inherently nonconvex, and therefore different techniques have been developed to solve them through convex optimization. In many applications, some of the nonconvex parts of the original problem are convexified using the so-called lossless convexification [7], [8], often resulting in second-order cone constraints. This is the case, for example, for the constraints on the thrust variables of the vast majority of the aerospace optimization problems, including low-thrust trajectory optimization [9], [10], [11], [12], powered descent guidance [13], [14], and in general of different aerial and ground trajectory optimization problems [15]. When a lossless convexification is not possible, nonconvex constraints are usually convexified by linearization [16], which introduces approximations. Consequently, problems that require successive convexification are solved using sequential convex programming [16], [17], an iterative trust region-based method that considers a sequence of convex subproblems whose solutions eventually converge to the solution of the original problem under certain conditions [16].

Depending on their formulation, different classes of convex problems can be considered [18]; one

[1] https://www.spacex.com/vehicles/falcon-9/. Last access: June 2023

rather wide category is represented by second-order cone programs, which are solved through tailored interior-point [19], first-order [20], or active set [21] methods. In particular, interior-point methods are believed to be the most suitable ones for autonomous guidance applications [22]. Therefore, this work focuses on the development of a warm-start strategy for SOCPs solved by interior-point methods and in the context of sequential convex programming. As a matter of fact, the state-of-the-art SCP is partially inefficient. The reason is that at each iteration, the initial guess for the primal and dual variables is derived from scratch by the convex optimization solver [23]. Notoriously, interior-point methods are difficult to warm-start as their solution lies on the boundaries of the feasible set, and thus the solution of a problem itself cannot be used as an initial guess for a similar problem because for the latter it would not be well-centered, and this would cause slow progress or even divergence [24], [25].

Different strategies have been proposed to exploit the information coming from the solution of an unperturbed convex problem to warm-start a perturbed one. In particular, some works [26], [27], [24], [28] developed methods to warm-start linear programs. Attention was also given to techniques specifically designed for nonconvex nonlinear programming that make use of interior-point methods [29]. In [30], a strategy to warm-start second-order cone optimization problems via rounding over optimal Jordan frames was proposed, which consists of solving two additional linear optimization problems to find a suitable initial guess. Moreover, [25] developed a strategy valid for both linear and conic quadratic programs that requires no additional optimization. However, this strategy was only applied to solve artificially perturbed convex problems and only allowed the exploitation of the last interior-point solver iteration to warm-start the perturbed one. To the best of the authors' knowledge, a warm-start strategy for second-order cone programs was never developed and tested within the SCP algorithm, where a sequence of strictly-related problems is considered. Only attempts to warm-start the whole SCP process have been made: for example, [31] developed a technique based on machine learning to find an accurate initial trajectory guess for the algorithm.

Therefore, the contribution of this article is threefold: first of all, it extends the strategy developed in [25], also allowing for the choice of which interior-point solver iteration is to be used to warm-start the next SCP iteration. Secondly, it proposes a modified version of the open-source Embedded COnic Solver [23] such that it includes the possibility of warm-starting the solver itself. Finally, it develops a methodology to integrate the proposed warm-start strategy in the context of the SCP algorithm, with the purpose of minimizing the number of iterations and

CPU time to reach convergence. The remainder of the article is structured as follows. Section II describes the problem and Section III introduces sequential convex programming. Section IV presents the combined SCP and warm-start strategy. Section V contains the performed simulations to show the efficacy of the strategy and finally, Section VI concludes the work.

II. Problem Formulation

Consider a cone program of the form [23]

$$\begin{aligned}\text{minimize} \quad & \mathbf{c}^\top \mathbf{x} && \text{(1a)}\\ \text{subject to:} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} && \text{(1b)}\\ & \mathbf{G}\mathbf{x} \preceq_{\boldsymbol{K}} \mathbf{h}, && \text{(1c)}\end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^{n\times 1}$ is the decision vector, $\mathbf{c} \in \mathbb{R}^{n\times 1}$ is a vector of constants, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^{m\times 1}$ define the equality constraints, and $\mathbf{G} \in \mathbb{R}^{p\times n}$ and $\mathbf{h} \in \mathbb{R}^{p\times 1}$ define the generalized inequality constraints with respect to a convex cone $\boldsymbol{K}$, defined as the product of N proper cones $\boldsymbol{K}_i$ such that

$$\boldsymbol{K} = \boldsymbol{K}_1 \times \boldsymbol{K}_2 \times \ldots \times \boldsymbol{K}_N, \tag{2}$$

where the sum of the dimensions of the cones $\boldsymbol{K}_i$, with $i = 1, \ldots, N$, sums up to p. In this work, problems that involve two kinds of proper cones are considered: the positive orthant $\mathbb{R}_+$ such that $\boldsymbol{K}_i = \mathbb{R}^n_+$, and the second order cones $\boldsymbol{K}^S_i$ defined as

$$\boldsymbol{K}^S_i = \{(\mathbf{v}_0, \mathbf{v}_1) \in \mathbb{R} \times \mathbb{R}^{n-1} | \mathbf{v}_0 \geq \|\mathbf{v}_1\|_2\}. \tag{3}$$

The generalized inequality in (1c) is equivalent to [23]

$$\mathbf{G}\mathbf{x} \preceq_{\boldsymbol{K}} \mathbf{h} \quad \Leftrightarrow \quad \mathbf{h} - \mathbf{G}\mathbf{x} \in \boldsymbol{K} \quad \Leftrightarrow \quad \mathbf{G}\mathbf{x} + \mathbf{s} = \mathbf{h},\ \mathbf{s} \in \boldsymbol{K}. \tag{4}$$

Interior-point methods usually solve the primal cone program in (1a)–(1c) simultaneously with its dual form, described by [23]

$$\begin{aligned}\text{maximize} \quad & -\mathbf{b}^\top \mathbf{y} - \mathbf{h}^\top \mathbf{z} && \text{(5a)}\\ \text{subject to:} \quad & \mathbf{G}^\top \mathbf{z} + \mathbf{A}^\top \mathbf{y} + \mathbf{c} = \mathbf{0} && \text{(5b)}\\ & \mathbf{z} \succeq_{\boldsymbol{K}^*} \mathbf{0}, && \text{(5c)}\end{aligned}$$

where $\mathbf{y}$ and $\mathbf{z}$ are the dual variables of the problem, and $\boldsymbol{K}^*$ is the dual convex cone, which, in the considered applications, is always equal to the convex cone $\boldsymbol{K}$. The primal and dual problems can be solved together through a primal-dual path-following interior-point method. The main idea of these methods is to build a new optimization problem that includes both the primal and dual ones and find their solutions by driving the so-called *central path* to satisfy its Karush-Kuhn-Tucker (KKT) conditions. The central path of the new problem is defined as

$$\tau\mathbf{c} + \mathbf{A}^\top\mathbf{y} + \mathbf{G}^\top\mathbf{z} = \mathbf{0} \tag{6a}$$

$$\mathbf{A}\mathbf{x} - \tau\mathbf{b} = \mathbf{0} \tag{6b}$$

$$\mathbf{G}\mathbf{x} + \mathbf{s} - \tau\mathbf{h} = \mathbf{0} \tag{6c}$$

$$\mathbf{c}^\top\mathbf{x} + \mathbf{b}^\top\mathbf{y} + \mathbf{h}^\top\mathbf{z} + \kappa = 0 \tag{6d}$$

$$(\mathbf{s}, \mathbf{z}) \succeq_{\boldsymbol{K}} \mathbf{0}, \quad (\tau, \kappa) \geq 0, \tag{6e}$$

where two additional scalar variables τ and κ have been introduced. These variables allow for the detection of infeasibility of either the primal or dual problem, or both. Note that the variables $\mathbf{s}$ and $\mathbf{z}$ must always be inside the convex cone, and the variables τ and κ must always be positive. It can be shown that (6a)–(6e) imply

$$\mathbf{s}^\top\mathbf{z} + \tau\kappa = 0, \tag{7}$$

which means that $\mathbf{s}$, $\mathbf{z}$, τ, and κ must satisfy a complementary condition, i.e., $s_i z_i = 0, \forall i = 1, \ldots, p$ and $\tau\kappa = 0$. In other words, all the feasible points of the optimization problem associated with the central path defined in (6a)–(6e) lie on the boundary of the feasible set [32]. Interior-point methods, however, as their name also suggests, can only deal with iterates that lie in the interior of the feasible set. Consequently, a new optimization problem is defined such that it has strictly feasible points that respect $\mathbf{s}^\top\mathbf{z} + \tau\kappa > 0$ and the condition in (7) only holds at the optimal point. More information about interior-point methods can be found in [32].

Suppose to have a convex problem k associated with the matrices $\mathbf{A}_k$ and $\mathbf{G}_k$ and vectors $\mathbf{b}_k$, $\mathbf{c}_k$, and $\mathbf{h}_k$. Let us indicate the solution of this optimization problem as $(\mathbf{x}_k^*, \mathbf{y}_k^*, \mathbf{z}_k^*, \mathbf{s}_k^*, \tau_k^*, \kappa_k^*)$. Consider now a new convex problem $k+1$ associated with the matrices $\mathbf{A}_{k+1}$ and $\mathbf{G}_{k+1}$ and vectors $\mathbf{b}_{k+1}$, $\mathbf{c}_{k+1}$, and $\mathbf{h}_{k+1}$ such that

$$\begin{aligned} &\mathbf{A}_{k+1} = \mathbf{A}_k + \Delta\mathbf{A}_k, \quad \mathbf{G}_{k+1} = \mathbf{G}_k + \Delta\mathbf{G}_k \\ &\mathbf{b}_{k+1} = \mathbf{b}_k + \Delta\mathbf{b}_k, \quad \mathbf{c}_{k+1} = \mathbf{c}_k + \Delta\mathbf{c}_k \\ &\mathbf{h}_{k+1} = \mathbf{h}_k + \Delta\mathbf{h}_k, \end{aligned} \tag{8}$$

where the perturbation terms $\Delta(\cdot)_k$ are such that most of their elements are *small* with respect to the correspondent elements of the original matrices and vectors. In this work, if

$$\|\Delta(\cdot)_k\|_1 \ll \|(\cdot)_k\|_1 \tag{9}$$

for all the matrices and vectors $(\cdot)_k$, then the problems k and $k+1$ are defined as *closely related*. Since the proposed approach is tailored for SCP applications, where two instances of the same problem linearized at different but close reference solution are solved, it will be assumed that the condition in Eq. (9) holds. In this case, it is reasonable to believe that the solution $(\mathbf{x}^*_{k+1}, \mathbf{y}^*_{k+1}, \mathbf{z}^*_{k+1}, \mathbf{s}^*_{k+1}, \tau^*_{k+1}, \kappa^*_{k+1})$ of the new problem is similar to the solution of the original one. Intuitively, it would come natural to use the solution of problem k as the initial guess for problem $k+1$. However, the solution of the former problem lies close to the boundary of the feasible set (i.e., it is not well-centered) as imposed by the condition in (7) and thus should not be used, as this would result in slow progress of the algorithm or even run into numerical problems. This article addresses the problem of finding a suitable initial guess for problem $k+1$ exploiting the information that can be retrieved from the iterates of the convex optimization solver associated with problem k.

III. General Formulation of an SOCP Subproblem Within SCP

In this work, the convex problem k associated with the matrices $\mathbf{A}_k$ and $\mathbf{G}_k$ and the vectors $\mathbf{b}_k$, $\mathbf{h}_k$, and $\mathbf{c}_k$ represents the kth iteration of a sequential convex programming algorithm [16], [33].

The warm-start strategy presented in this article is applied to second-order cone programs that represent optimal control problems. In other words, the problem of finding the control action $\mathbf{u}(t) \in \mathbb{R}^{n_u \times 1}$ that minimizes an objective function while ensuring that some constraints on the state $\mathbf{w}(t) \in \mathbb{R}^{n_w \times 1}$ and on the control itself are respected is considered. A generic nonlinear, nonconvex optimal control problem can be expressed in the form

$$\underset{\mathbf{u}(t)}{\text{minimize}} \quad \int_{t_0}^{t_f} L(\mathbf{u}(t))\,\mathrm{d}t \tag{10a}$$

$$\text{subject to:} \quad \dot{\mathbf{w}}(t) = \mathbf{f}(\mathbf{w}(t), \mathbf{u}(t)) \tag{10b}$$

$$\mathbf{q}(\mathbf{w}(t), \mathbf{u}(t)) = 0 \tag{10c}$$

$$\mathbf{g}(\mathbf{w}(t), \mathbf{u}(t)) \leq 0 \tag{10d}$$

$$u_0^2(t) + u_1^2(t) + \cdots + u_{n_u-1}^2(t) = u_{n_u}^2(t) \tag{10e}$$

$$w_i(t_0) = w_{i,0}, \quad i = 1, \ldots, n_w \tag{10f}$$

$$w_i(t_f) = w_{i,f}, \quad i = 1, \ldots, n_f \leq n_w, \tag{10g}$$

where t_0 and t_f are the initial and final times of the problem, respectively. $L(\mathbf{u}(t))$ is a function of the control variables, $f(\mathbf{w}(t), \mathbf{u}(t)) \in \mathbb{R}^{n_w \times 1}$ is the dynamics of the system, and $\mathbf{q}(\mathbf{w}(t), \mathbf{u}(t)) \in \mathbb{R}^{n_q \times 1}$ and $\mathbf{g}(\mathbf{w}(t), \mathbf{u}(t)) \in \mathbb{R}^{n_g \times 1}$ are the equality and inequality constraint functions, respectively. In general, the problem in (10a)–(10g) is not a SOCP due to the nonlinear constraints. If the convex function $L(\mathbf{u}(t)) = u_{n_u}(t)$ is assumed, after having applied lossless convexification and having linearized the nonlinear dynamics, the equality, and inequality nonlinear constraints around a reference solution $(\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t))$, the problem becomes

$$\begin{aligned}
\underset{u(t)}{\text{minimize}} \quad & \int_{t_0}^{t_f} u_{n_u}(t)\,\mathrm{d}t + \int_{t_0}^{t_f} \lambda \left\|\boldsymbol{\nu}(t)\right\|_1 \,\mathrm{d}t + \\
& \int_{t_0}^{t_f} \lambda \left\|\boldsymbol{\phi}(t)\right\|_1 \,\mathrm{d}t + \qquad\qquad \text{(11a)} \\
& \int_{t_0}^{t_f} \lambda \max(0, \boldsymbol{\eta}(t))\,\mathrm{d}t \\
\text{subject to:} \quad & \dot{\mathbf{w}}(t) = \mathbf{f}(\bar{\mathbf{w}}(t), \bar{u}(t)) + \\
& \mathbf{J}\left(\mathbf{w}(t) - \bar{\mathbf{w}}(t)\right) + \qquad\qquad \text{(11b)} \\
& \mathbf{B}(\mathbf{u}(t) - \bar{\mathbf{u}}(t)) + \boldsymbol{\nu}(t) \\
& \mathbf{q}(\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t)) + C(\mathbf{w}(t) - \bar{\mathbf{w}}(t)) + \\
& \mathbf{D}(\mathbf{u}(t) - \bar{\mathbf{u}}(t)) = \boldsymbol{\phi}(t) \qquad\qquad \text{(11c)} \\
& \mathbf{g}(\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t)) + \mathbf{E}(\mathbf{w}(t) - \bar{\mathbf{w}}(t)) + \\
& \mathbf{F}(\mathbf{u}(t) - \bar{\mathbf{u}}(t)) \leq \boldsymbol{\eta}(t) \qquad\qquad \text{(11d)} \\
& u_0^2(t) + u_1^2(t) + \cdots + u_{n_u-1}^2(t) \leq u_{n_u}^2(t) \qquad\qquad \text{(11e)} \\
& \left\|\mathbf{w}(t) - \bar{\mathbf{w}}(t)\right\|_1 \leq R \qquad\qquad \text{(11f)} \\
& \left\|\mathbf{u}(t) - \bar{\mathbf{u}}(t)\right\|_1 \leq R_u \qquad\qquad \text{(11g)} \\
& w_i(t_0) = w_{i,0}, \quad i = 1, \ldots, n_w \qquad\qquad \text{(11h)} \\
& w_i(t_f) = w_{i,f}, \quad i = 1, \ldots, n_f, \qquad\qquad \text{(11i)}
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{J} &:= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t)}, \ \mathbf{B} := \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t)}, \\
\mathbf{C} &:= \left.\frac{\partial \mathbf{q}}{\partial \mathbf{w}}\right|_{\bar{w}(t), \bar{\mathbf{u}}(t)}, \ \mathbf{D} := \left.\frac{\partial \mathbf{q}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{w}}(t), \bar{u}(t)}, \qquad\qquad (12) \\
\mathbf{E} &:= \left.\frac{\partial \mathbf{g}}{\partial \mathbf{w}}\right|_{\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t)}, \ \mathbf{F} := \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{w}}(t), \bar{\mathbf{u}}(t)}.
\end{aligned}$$

The slack variables $\boldsymbol{\nu}(t)$, $\boldsymbol{\phi}(t)$, and $\boldsymbol{\eta}(t)$ have been introduced to avoid the so-called artificial infeasibility [16]. On top of that, the trust-region constraints in (11f) and (11g) are added to ensure that the linearization of the constraints is valid, and R and R_u are the trust-region radii of states and controls, respectively. In (11h) and (11i), $w_{i,0}$ and $w_{i,f}$ indicate the ith prescribed initial and final boundary condition, respectively. After discretizing the problem in (11a)–(11i) and collecting the resulting decision variables $w(t_i)$, $u(t_i)$, $i = 1, \dots, M$ in one single vector $\mathbf{x} = [w(t_1), \dots, w(t_M), u(t_1), \dots, u(t_M)]^\top$, it represents a SOCP that can be written in the form of (1a)–(1c) and consequently solved by means of some convex solver. The initial guess for such a problem is commonly built from scratch by the convex solver by considering additional optimization problems without any information required by the user, and this is what is referred to as the cold-start approach. In this article, the proposed warm-start strategy is compared with the cold-start implemented in [32].

IV. Warm-Start Strategy

A suitable warm-start strategy embedded in the SCP algorithm should have at least the following three characteristics: first, it should reduce the computational effort of the standard algorithm. This is clearly the primary objective of a warm-start strategy. Secondly, it should not deteriorate the convergence properties of its cold-start version. Finally, it should not increase the complexity of the algorithm, as this may result in an increased chance of failure or unpredictable numerical problems or errors. In the light of these considerations, the warm-start strategy presented in [25] is extended and adapted to the SCP algorithm, resulting in an efficient and robust method to warm-start a sequence of closely-related second-order cone programs.

A. Overview

Let us indicate with $(\cdot)_k^{(i)}$ the variables $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{s}, \tau, \kappa)$ at the ith iterate of the convex solver at the kth SCP iteration. The initial guess for the $(k+1)$th SCP iteration is defined as

$$\begin{cases} \mathbf{x}_{k+1}^{(0)} = \mathbf{x}_k^{(\alpha_k)} \\ \mathbf{y}_{k+1}^{(0)} = \mathbf{y}_k^{(\alpha_k)} \\ \mathbf{z}_{k+1}^{(0)} = (1-\lambda_k)\mathbf{z}_k^{(\alpha_k)} + \lambda_k \mathbf{e} \\ \mathbf{s}_{k+1}^{(0)} = (1-\lambda_k)\mathbf{s}_k^{(\alpha_k)} + \lambda_k \mathbf{e} \\ \tau_{k+1}^{(0)} = \tau_k^{(\alpha_k)} \\ \kappa_{k+1}^{(0)} = (\mathbf{s}_{k+1}^{(0)})^\top \mathbf{z}_{k+1}^{(0)}/p, \end{cases} \tag{13}$$

where $\alpha_k \in \mathbb{N}^+$ such that $\alpha_k \in [1, \dots, I_k]$ with I_k being the last iteration of the convex optimization solver of the SCP problem k, $\lambda_k \in [0, 1]$ is a weighting parameter, and $\mathbf{e}$ is the so-called *identity element* for the convex cone $\boldsymbol{K}$ [32]. The idea behind this strategy is the following. As already mentioned, the problems associated with the SCP iterations k and $k+1$ are similar, and thus it is reasonable to believe that they also have similar solutions. Therefore, it is intuitive to consider an iterate α_k of the kth convex problem and use it as the initial guess for problem $(k+1)$. If an advanced iterate is considered (i.e., if α_k is close to I_k), however, the variables z and s may be too close to the boundaries of the feasible set for the iterate α_k to be used as an initial guess for problem $k+1$. Consequently, the variables $\mathbf{z}$ and s are initialized with a linear combination of the selected iterate of problem k and the identity element e, which brings the variables inside the convex cone $\boldsymbol{K}$. Note that, different from [26], we do not consider only the last iterate I_k of the convex solver for this procedure. This is because we observed that the last iterate I_k is not always the most effective to reduce the computational effort of the algorithm regardless of the value of the weighting parameter λ_k. Therefore, we allow for more flexibility by introducing an additional degree of freedom to the strategy, represented by the *warm-start index* α_k.

B. Integration With Sequential Convex Programming

The integration of the warm-start method presented in Section IV-A with the sequential convex programming algorithm consists of developing a strategy to define how to select the parameters α_k and λ_k at each SCP iteration. In general, our numerical simulations, as well as results from previous work [26], show that the more similar the matrices and vectors of two problems k and $k+1$ are, the more the warm-start for problem $k+1$ benefits from an advanced solver iterate of problem k. However, the higher α_k, the closer the variables $\mathbf{z}$ and s are to the boundaries of the feasible set, and therefore a higher value of the parameter λ_k is required.

A reasonable measure of how much two problems differ from each other is the sum Σ_k of the l-norms of the difference of the associated matrices and vectors. In other words,

$$\Sigma_k = \left\|\Delta \mathbf{A}_k\right\|_l + \left\|\Delta \mathbf{G}\right\|_l + \left\|\Delta \mathbf{b}_k\right\|_l + \left\|\Delta \mathbf{h}_k\right\|_l + \left\|\Delta \mathbf{c}_k\right\|_l , \tag{14}$$

where l can be any norm. The quantities in (14) are defined in accordance with (8). In this work, it is assumed that the optimal cobtrol problem is written in dimensionless form, and therefore the quantity Σ_k is dimensionless. Although a mathematical proof is out of the scope of the work, our simulations show that the parameter Σ_k is a good indicator of how much two subsequent convex problems in the context of SCP differ. In Fig. 1, the left y-$axis$, the values of Σ for a typical problem are shown as a function of the SCP iteration k. The right y-$axis$, the metric

$$\Delta S_k = \|S_{k+1} - S_k\|_l,\ l = \infty$$

is represented as a function of the SCP iteration k, where

$$S_k = \|\Delta\mathbf{x}_k\|_l + \|\Delta\mathbf{y}_k\|_l + \|\Delta\mathbf{s}_k\|_l + \|\Delta\mathbf{z}_k\|_l \tag{15}$$

and

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta\mathbf{x}_k, \quad \mathbf{y}_{k+1} = \mathbf{y}_k + \Delta\mathbf{y}_k \\ \mathbf{s}_{k+1} &= \mathbf{s}_k + \Delta\mathbf{s}_k, \quad \mathbf{z}_{k+1} = \mathbf{z}_k + \Delta\mathbf{z}_k \end{aligned} \tag{16}$$

The figure shows that there is indeed a strong dependency of the parameter ΔS_k on Σ_k.

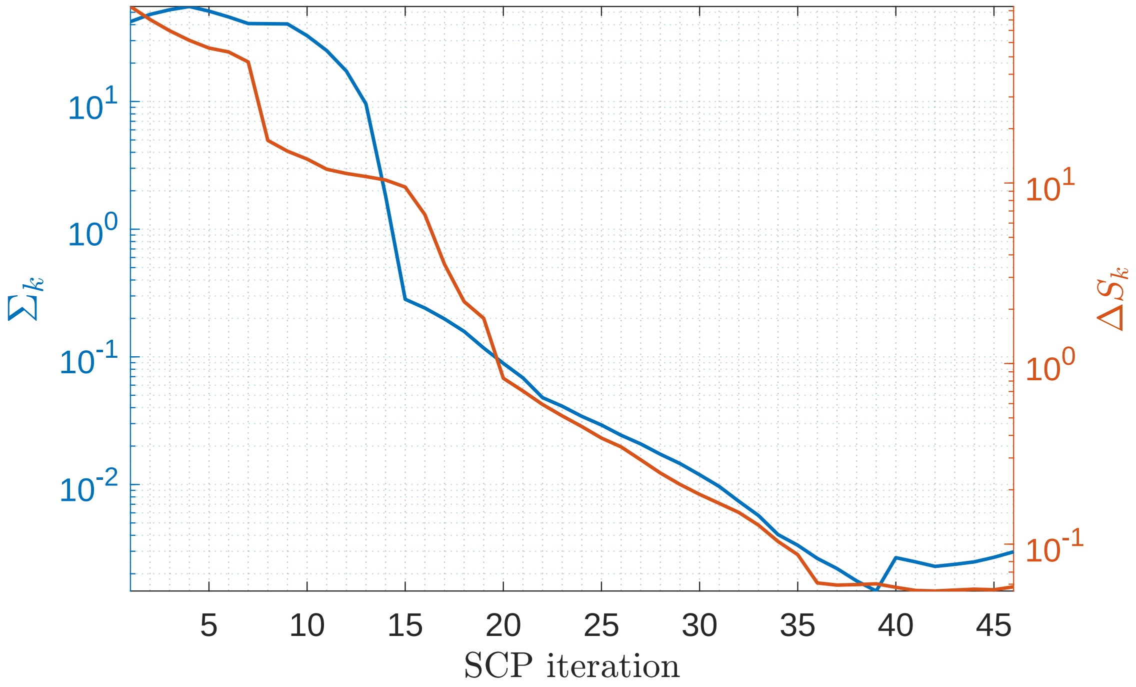


Fig. 1: Σ_k and norm of solution difference as a function of the SCP iteration.

Our strategy proposes a linear dependence between Σ_k and the value of the weighting parameter λ_k to be used to warm-start the SCP iteration $k + 1$, such that

$$\lambda_k = f_\lambda \Sigma_k, \tag{17}$$

where $f_\lambda > 0$ is a user-selected constant parameter. Since the weighting parameter λ_k must not exceed 1, at each SCP iteration k, its value is updated as

$$\lambda_k = \min(f_\lambda \Sigma_k, 1). \tag{18}$$

The rationale behind this choice is the following: if the value of Σ_k is small, i.e., if two subsequent SCP problems are similar, it is convenient to modify the iterate k as little as possible.

Consequently, it is advantageous to have a small value of λ_k as well. Still, it is also important to add some centrality to the selected initial guess to allow for quick progress of the algorithm. However, at the limit, i.e., when $\Sigma_k \to 0$, we consider $\lambda_k = 0$ because we would basically be solving the same problem twice, and therefore, the initial guess would also be the actual solution. On the other hand, if the value of Σ_k is large, i.e., if two subsequent SCP problems differ significantly, it is preferable to have a well-centered initial guess because there would be no relevant benefit from considering an unmodified iterate of the previous optimization subproblem. Note that $\lambda_k \geq 0$ according to (17); in particular, we have $\lambda_k = 0 \Leftrightarrow \Sigma_k = 0$. Since this would correspond to solving the exact same convex problem twice (and this is never the case in the context of SCP), we can conclude that $\lambda_k > 0$ for all k.

Our policy to select the iterate of the SCP problem k to warm-start the SCP problem $k+1$ (i.e., to select the value of α_k) is defined as follows. We introduce a new parameter $\delta_{\alpha,k}$ such that

$$\delta_{\alpha,k} = \frac{2}{1+\exp[f_\alpha \log_{10}(\lambda_k)]} - 1, \tag{19}$$

where $f_\alpha \geq 0$ is another user-selected constant parameter. Next, we define the value of α_k as

$$\alpha_k = \text{round}(\delta_{\alpha,k} I_k), \tag{20}$$

where round$(\cdot)$ is a function that rounds a value to the nearest integer. In addition, we impose that $\alpha_k = 1$ if round$(\delta_{\alpha,k} I_k) < 1$. Note from (19) that the parameter $\delta_{\alpha,k}$ is defined as a modified sigmoid function depending on λ_k because this assures that the inequalities $0 \leq \delta_{\alpha,k} \leq 1$ always hold. In fact, since $\delta_{\alpha,k}$ indicates a fraction of I_k, it cannot exceed those bounds. Figure 2 shows $\delta_{\alpha,k}$ as a function of λ_k for different values of the parameter f_α.

Equations (19) and (20) essentially mean that the smaller λ_k is, i.e., the more similar two subsequent SCP subproblems are, the more advanced the iterate of the convex solver is to be considered to warm-start the following optimization process. On the contrary, if Σ_k is large, it is reasonable to believe that the two subsequent subproblems have considerably different solutions. Consequently, an advanced iterate should not be used. Instead, taking a well-centered (i.e., larger λ_k), less advanced iterate may be a more appropriate choice.

It is important to underline that the update of the parameters α_k and λ_k only happens when an SCP iteration is accepted. In fact, it would not be helpful to consider information coming from a rejected iteration to warm-start a new convex optimization subproblem.

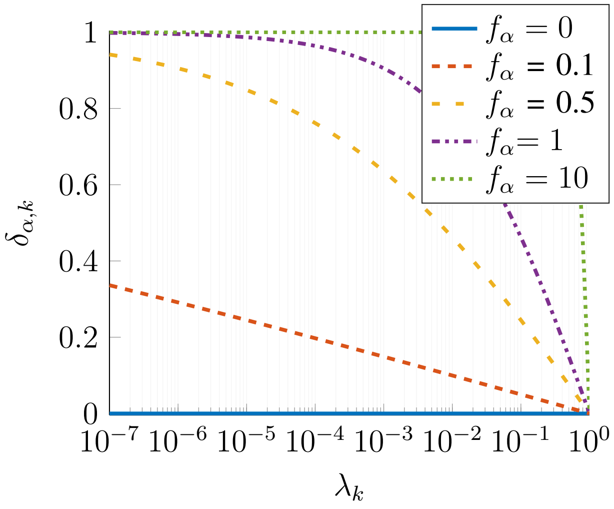


Fig. 2: Value of $\delta_{\alpha,k}$ as a function of λ_k and f_α.

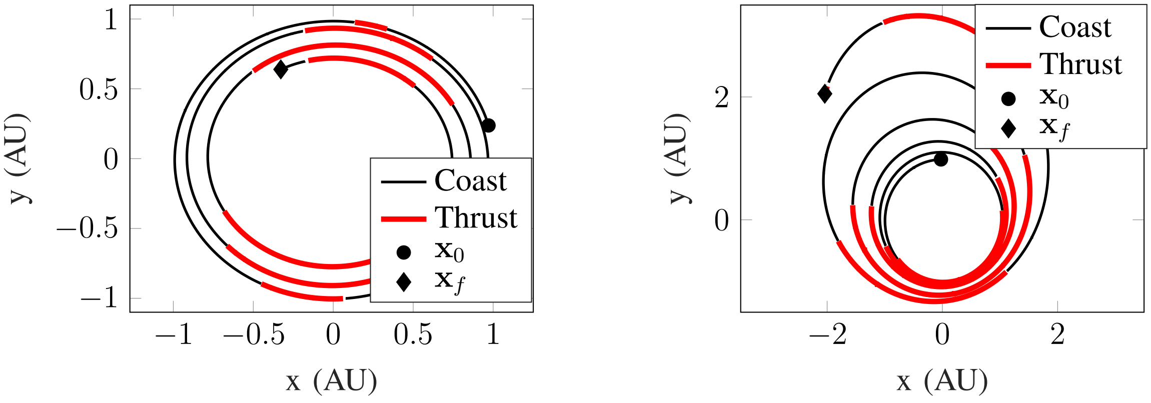
V. Numerical Simulations

In order to show the effectiveness of our warm-start strategy, we specify (11a)–(11i) for the low-thrust trajectory optimization problem, whose convex subproblems are extensively described in [2] for Cartesian coordinates. We consider the transfer trajectories from the Earth to Venus and to asteroid Dionysus. Data about the transfers are reported in Table I, whereas Fig. 3 shows typical solutions of the considered problems in terms of transfer trajectories. In Table I and Fig. 3, $\mathbf{x}_0 = [\mathbf{r}_0; \mathbf{v}_0]$ and $\mathbf{x}_f = [\mathbf{r}_f; \mathbf{v}_f]$ are the initial and final spacecraft states, respectively, with $\mathbf{r}_i, \mathbf{v}_i$ $(i = 0, f)$ indicating the position and velocity vectors; m_0 and m_f are the initial and final spacecraft mass, respectively; T_{max} is the maximum thrust of the spacecraft engine, and I_{sp} is its specific impulse. Finally, t_f indicates the time of flight of the transfer. More information on the SCP algorithm and the values of the SCP parameters used in this article can be found in [2]. The arbitrary-order Hermite–Legendre–Gauss–Lobatto (HLGL) discretization method [12] is exploited to discretize the convex subproblems of the SCP procedure, as it represents a widely used and fairly efficient collocation scheme. The resulting discretized SOCPs have thousands of variables and large matrices $\mathbf{A}$ and $\mathbf{G}$ in (1). Table III details the size of each vector and matrix associated with the considered examples. It can be noted that they represent large and complex optimization problems. With respect to the warm-start parameters, we use $l = \infty$ for (14).

The objective of the following numerical simulations is to compare the standard cold-start strategy implemented in common convex solvers with the simplest possible warm-start approach, i.e.,

TABLE I: Simulation values for Earth-Venus and Earth-Dionysus transfers [34], [35].

Parameter	Earth-Venus	Earth-Dionysus
$\mathbf{r}_0$, LU	$[0.97083220, 0.23758440, -1.67106 \times 10^{-6}]^\top$	$[-0.02431767, 0.98330142, -1.51168 \times 10^{-5}]^\top$
$\mathbf{v}_0$, VU	$[-0.25453902, 0.96865497, 1.50402 \times 10^{-5}]^\top$	$[-1.01612926, -0.02849401, 1.69550 \times 10^{-6}]^\top$
m_0, kg	1500	4000
$\mathbf{r}_f$, LU	$[-0.32771780, 0.63891720, 0.02765929]^\top$	$[-2.04061782, 2.05179130, 0.55428895]^\top$
$\mathbf{v}_f$, VU	$[-1.05087702, -0.54356747, 0.05320953]^\top$	$[-0.14231932, -0.45108800, 0.01894690]^\top$
m_f, kg	free	free
T_{max}, N	0.33	0.32
I_{sp}, s	3800	3000
t_f, days	1000	3534



(a) Earth-Venus

(b) Earth-Dionysus

Fig. 3: Typical Earth-Venus and Earth-Dionysus transfer trajectories.

to consider an iterate of the previous convex subproblem and use it as the initial guess for the following subproblem without adding any centrality (i.e., $\delta_{\alpha,k} = const. > 0$ and $\lambda_k \equiv 0$). Throughout the rest of the article, we will refer to this as the basic warm-start strategy. The basic warm-start strategy analysis also serves to better understand the improvements that our advanced warm-start approach (i.e., $\alpha_k > 0$ and $\lambda_k > 0$) is able to provide. In order to understand the influence of δ_α^{B}, f_α^{A}, and f_λ^{A} (where the superscripts $(\cdot)^{\text{B}}$ and $(\cdot)^{\text{A}}$ indicate the basic and advanced strategies, respectively) on the results, we first perform a parametric analysis. For the basic warm-start strategy, the values $\boldsymbol{\delta}_{\alpha}^{\mathrm{B}} = [0.1, 0.3, 0.5]$ are investigated, whereas the vectors

$$\boldsymbol{f}_{\alpha}^{\mathrm{A}} = 0 : \Delta : 1, \quad \Delta = 0.1, \tag{21}$$

$$\boldsymbol{f}_{\lambda}^{\mathrm{A}} = 10^{-k}, \quad k = 0, \dots 8 \tag{22}$$

are considered for the advanced strategy. A total of 20 SCP problems are considered for each combination of the aforementioned parameters, using perturbed cubic-interpolated initial guesses and an Hermite–Simpson discretization scheme [12] with 150 and 250 nodes for the Earth to Venus and Earth to Dionysus transfers, respectively. It can be noted that:

1. the case with $f_{\alpha} = 0$ corresponds to $\alpha_k \equiv 1$ but, in general, $\lambda_k \neq 0$;
2. the case with $f_{\lambda} = 1$, according to (17) and (18), corresponds to having a high value of λ_k.

They represent the trivial combinations of parameters of the advanced warm-start strategy: in the first case, the convex solver always considers the very first iteration of problem k to warm-start problem $k + 1$; in the second case, the information retrieved from problem k is (almost) not exploited at all.

A. *Parametric Analysis for Earth-Venus Transfer*

The cold-start strategy converged in 5 out of 20 simulations, and it required an average of 1154.6 convex solver iterations for each SCP problem (with an average of 36.8 SCP iterations). The results associated with the basic warm-start strategy are presented in Table II. Interestingly, small values of δ_k^{B} seem to increase the convergence of the SCP algorithm. When larger values of the parameter are considered, however, the strategy fails. The results associated with the advanced warm-start strategy are shown in Figs. 4a and 4b. In Fig. 4a, larger values correspond to more converged cases and therefore a better behavior; in Fig. 4b, lower values correspond to lower warm-to-cold iterations ratio and therefore a desirable behavior. To ensure a fair comparison, only the cases for which both the algorithms converged are considered. As expected, the convergence of the strategy is poor when an advanced iterate (i.e., large f_{α}) of problem k is used to warm-start problem $k + 1$ and little centrality (i.e., small f_{λ}) is added; the same happens when one of the first iterates of problem k (i.e., $f_{\alpha} \approx$ 0-0.3) is used together with high centrality (i.e., large f_{λ}). Selecting either low f_{λ} associated with small f_{α} or large f_{λ} associated with high f_{α} results in the best behavior, although having too much centrality or a very large value of f_{α} (in general, greater than 1) is not beneficial either. For what concerns the number of solver iterations, a large number of combinations of parameters is effective in improving the performance of the cold-start algorithm. In particular, the case with $f_\alpha = 0.1$ and $f_\lambda = 10^{-5}$ is extremely efficient in decreasing the computational effort of the SCP procedure, with an iterations ratio of $\approx 70\%$. Due to the properties of convex optimization problems, each SCP convex subproblem has a global solution only. Both the cold-start and the warm-start strategies should therefore find, in case they converge, the same final spacecraft mass. Consequently, the overall SCP final spacecraft mass using the two approaches should also be the same up to some numerical error. This is indeed supported by numerical simulations, which show that the normalized change of final spacecraft mass is in the order of 10^{-6}, and therefore negligible.

B. Parametric Analysis for Earth-Dionysus Transfer

The cold-start strategy converged in 17 out of 20 simulations, and it required an average of 2032.7 convex solver iterations for each SCP problem (with an average of 53.6 SCP iterations). Since this second transfer has a long time of flight and the orbital parameters of the Earth and Dionysus are significantly different, it is reasonable that the number of solver and SCP iterations is larger with respect to the previous test case. Moreover, the convergence of the HLGL discretization has proven to be more effective with the Earth–Dionysus transfer than with the Earth–Venus one [2]. Table II shows the basic warm-start strategy results. They are similar to the ones obtained with the previous example: as the value of δ_k^{B} increases, convergence follows the opposite trend. Figures 5a and 5b show the results in terms of converged cases and iterations ratio for the advanced warm-start strategy, respectively. There is only a small area where the strategy encounters issues to converge; as expected, this happens for large values of f_α and little centrality added. Interestingly, the iterations ratio plot is extremely similar to the Earth-Venus one. This indicates that the best warm-start parameters may only depend on the structure of the problem and not on the specific data. In that case, an additional benefit of the proposed strategy is represented by its generality. Although the combination of parameters $f_\alpha = 0.1$ and $f_\lambda = 10^{-5}$ provides very good results (with an iterations ratio of $\approx 73\%$), the cases that performed best in terms of solver iterations are the ones with $\{f_\alpha = 0.1, f_\lambda = 10^{-2}\}$, $\{f_\alpha = 0.2, f_\lambda = 10^{-2}\}$, and $\{f_\alpha = 0.1, f_\lambda = 10^{-8}\}$, which obtained an iterations ratio of $\approx 70\%$. However, while the former combination of parameters resulted in the same number of converged cases as the cold-start, the latter converged in one case less (as Fig. 5a shows).

δ^{B}_{k}	Venus	Dionysus
0.1	12	16
0.3	4	3
0.5	0	0

TABLE II: Converged cases for the basic warm-start strategy.

C. Comparison of Convergence and CPU Times

We select $\{\delta^{\mathrm{B}}_{\alpha,k} = 0.1,\ f^{\mathrm{A}}_{\alpha} = 0.1,\ f^{\mathrm{A}}_{\lambda} = 10^{-5}\}$ for both transfers and run a total of 100 simulations to further address the convergence of the warm-start strategies and to compare the reduction of CPU time with respect to the cold-start approach. The choice of the parameters is justified by a compromise in terms of convergence and iterations ratio according to the results presented in the previous Sections. Table IV presents the obtained results. The extensive simulations confirm the findings in Table II: the convergence of the basic warm-start strategy outperforms the cold-start one in the case of the Earth–Venus transfer, but it slightly worsens it in the Earth–Dionysus case. In addition, the former strategy can even be effective in decreasing the computational effort of the standard algorithm. The advanced warm-start method confirms its efficacy both in terms of converged cases ($+148.15\%$ with respect to the cold-start algorithm for the Earth–Venus transfer) and CPU time (with a -20% for both of the case studies).

D. General Assessment

We briefly summarize and discuss the main findings of our simulations in the following points:

- Our simulations confirm that the solution of a SOCP cannot be directly used as initial guess for a strictly-related one: as a matter of fact, the basic warm-start strategy already completely fails when $\delta_{\alpha,k} = 0.5$.
- In the context of SCP, simply considering a less advanced iterate of the interior-point process to solve a SOCP (i.e., our basic warm-start strategy) can instead be effective means to enhance convergence (in the Earth–Venus case) or decrease the computational effort (in the Earth–Dionysus case) of the cold-start approach. On the other hand, the performance of this strategy strongly depends on the considered case, as Table IV points out.
- The proposed advanced warm-start strategy outperforms the cold-start approach when the solver ECOS is used. However, we expect a similar behavior with any interior-point convex

	Venus					Dionysus				
Parameter	A	G	c	b	h	A	G	c	b	h
Rows	1066	8730	5116	5116	5116	1766	14530	8516	8516	8516
Columns	5116	5116	1	1	1	8516	8516	1	1	1
Nonzero elements	14349	13558	1208	901	5263	23949	22558	2008	1501	8763

TABLE III: Dimensions of the matrices and vectors for the two transfers.

	Venus		Dionysus	
Start	Converged cases	CPU time (s)	Converged cases	CPU time (s)
Cold	27.00	3.50	82.00	9.39
B. Warm	52.00	3.34	81.00	7.76
A. Warm	67.00	2.82	83.00	7.43
B. W/C (%)	192.59	95.30	98.78	82.64
A. W/C (%)	248.15	80.65	102.47	79.13

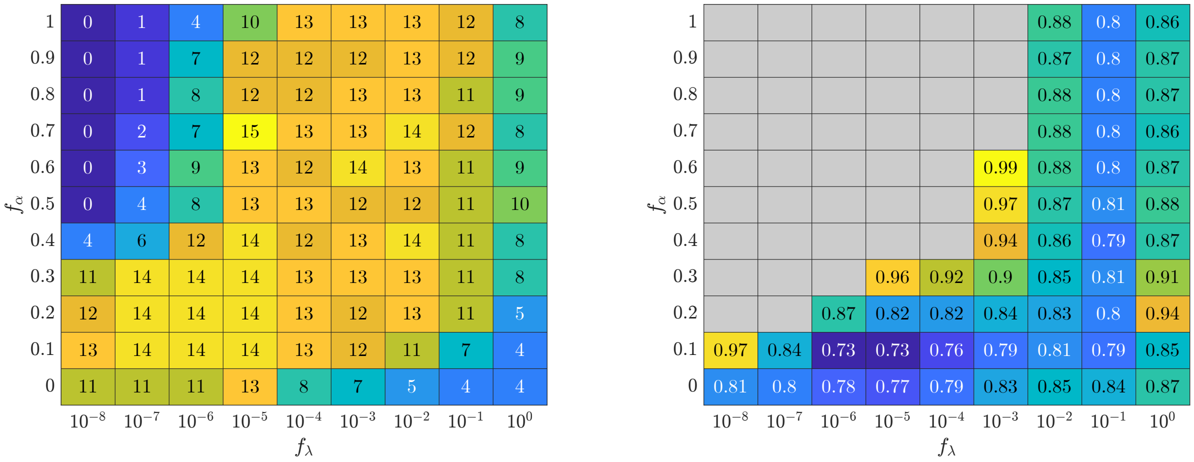
TABLE IV: Convergence and CPU time analysis results with Cartesian coordinates.

solver. Selecting the iterate to be used for the warm-start depending on the SCP iteration was also shown to be an effective strategy.

- Even though very complex examples with many variables and large matrices were considered and therefore the validity of the presented strategy is confirmed, it is likely that the best values of the advanced warm-start parameters f_α and f_λ depend on the considered specific problem.

VI. CONCLUSIONS

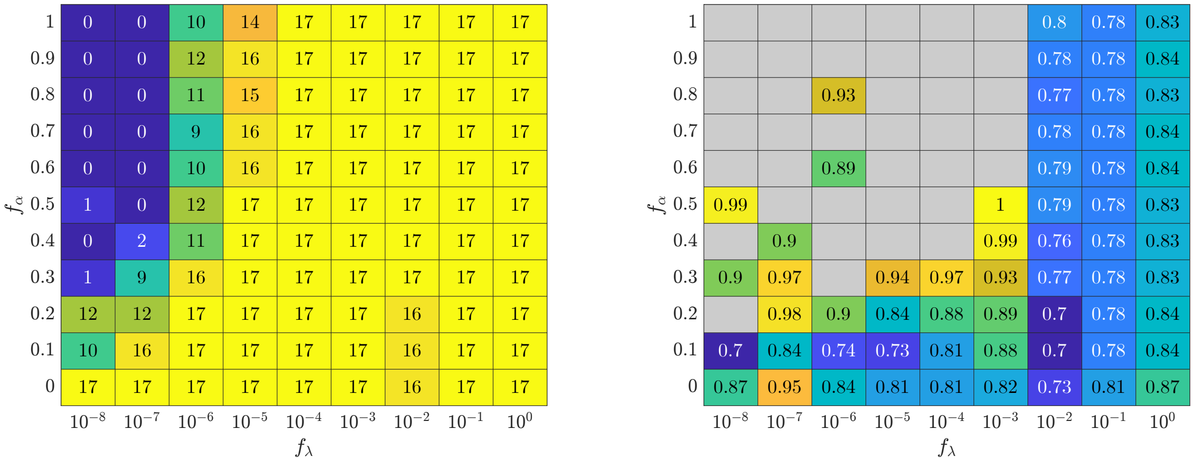
In this article, a strategy to warm-start second-order cone programs in the context of sequential convex programming is developed. It shows superiority both in terms of convergence and required computational effort, resulting in up to -30% CPU time and $+150\%$ converged cases for the considered test cases. Moreover, the results demonstrate the effectiveness of the proposed solution when it is integrated within the SCP technique to solve optimal control problems. It is still to be investigated how to quickly and efficiently select the warm-start parameters for a generic problem without performing preliminary extensive simulations.



(a) Number of converged cases as function of the two advanced warm-start strategy parameters.

(b) Advanced warm-to-cold iterations ratio. The shaded grey areas indicate that the iterations ratio is higher than 1.

Fig. 4: Parametric analysis for the Earth–Venus transfer.



(a) Number of converged cases as function of the two advanced warm-start strategy parameters.

(b) Advanced warm-to-cold iterations ratio. The shaded grey areas indicate that the iterations ratio is higher than 1.

Fig. 5: Parametric analysis for the Earth–Dionysus transfer.

REFERENCES

[1] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[2] C. Hofmann, A. C. Morelli, and F. Topputo, "Performance assessment of convex low-thrust trajectory optimization methods," *Journal of Spacecraft and Rockets*, vol. 60, no. 1, pp. 299–314, 2023.

[3] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on selected areas in communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[4] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, vol. 1, no. 1, pp. 23–40, Sep. 2017.

[5] M. Bruyneel and C. Fleury, "Composite structures optimization using sequential convex programming," *Advances in Engineering Software*, vol. 33, no. 7-10, pp. 697–711, 2002.

[6] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[7] M. W. Harris and B. Açıkmeşe, "Lossless convexification of non-convex optimal control problems for state constrained linear systems," *Automatica*, vol. 50, no. 9, pp. 2304–2311, 2014.

[8] B. Açıkmeşe, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, Nov. 2013.

[9] Z. Wang and M. J. Grant, "Optimization of minimum-time low-thrust transfers using convex programming," *Journal of Spacecraft and Rockets*, vol. 55, no. 3, pp. 586–598, 2018.

[10] ——, "Minimum-fuel low-thrust transfers for spacecraft: A convex approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2274–2290, 2018.

[11] C. Hofmann and F. Topputo, "Rapid low-thrust trajectory optimization in deep space based on convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 7, pp. 1379–1388, 2021.

[12] A. C. Morelli, C. Hofmann, and F. Topputo, "Robust low-thrust trajectory optimization using convex programming and a homotopic approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 2103–2116, 2021.

[13] M. Sagliano, "Generalized hp Pseudospectral-Convex Programming for Powered Descent and Landing," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, 2019.

[14] C. Yu and Y. Zhao, D. Yang, "Efficient convex optimization of reentry trajectory via the chebyshev pseudospectral method," *International Journal of Aerospace Engineering*, vol. 2019, 2019, article 1414279.

[15] Z. Wang and S. T. McDonald, "Convex relaxation for optimal rendezvous of unmanned aerial and ground vehicles," *Aerospace Science and Technology*, vol. 99, p. 105756, 2020.

[16] Y. Mao, M. Szmuk, X. Xu, and B. Açıkmeşe, "Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems," Preprint, submitted February 2019, https://arxiv.org/abs/1804.06539.

[17] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe, "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Control Systems Magazine*, vol. 42, no. 5, pp. 40–113, 2022.

[18] D. Malyuta, Y. Yu, P. Elango, and B. Açıkmeşe, "Advances in trajectory optimization for space vehicle control," *Annual Reviews in Control*, vol. 52, pp. 282–315, 2021.

[19] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of computational and applied mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.

[20] Y. Yu, P. Elango, U. Topcu, and B. Açıkmeşe, "Proportional–integral projected gradient method for conic optimization," *Automatica*, vol. 142, p. 110359, 2022.

[21] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.

[22] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, pp. 1–27, 2012.

[23] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP Solver for Embedded Systems," in *European Control Conference*, Zurich, Switzerland, 2013, pp. 3071–3076.

[24] E. John and E. A. Yıldırım, "Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension," *Computational Optimization and Applications*, vol. 41, no. 2, pp. 151–183, 2008.

[25] A. Skajaa, E. D. Andersen, and Y. Ye, "Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems," *Mathematical Programming Computation*, vol. 5, no. 1, pp. 1–25, 2013.

[26] E. A. Yildirim and S. J. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 782–810, 2002.

[27] H. Y. Benson and D. F. Shanno, "An exact primal–dual penalty method approach to warmstarting interior-point methods for linear programming," *Computational Optimization and Applications*, vol. 38, no. 3, pp. 371–399, 2007.

[28] A. Engau, M. F. Anjos, and A. Vannelli, "A primal-dual slack approach to warmstarting interior-point methods for linear programming," in *Operations Research and Cyber-Infrastructure*. Springer, 2009, pp. 195–217.

[29] H. Y. Benson and D. F. Shanno, "Interior-point methods for nonconvex nonlinear programming: regularization and warmstarts," *Computational Optimization and Applications*, vol. 40, no. 2, pp. 143–189, 2008.

[30] S. B. Cay, I. Pólik, and T. Terlaky, "Warm-start of interior point methods for second order cone optimization via rounding over optimal jordan frames," *ISE TechnicalReport 17T-006, Lehigh University*, 2017.

[31] S. Banerjee, T. Lew, R. Bonalli, A. Alfaadhel, I. A. Alomar, H. M. Shageer, and M. Pavone, "Learning-based warm-starting for fast sequential convex programming and trajectory optimization," in *2020 IEEE Aerospace Conference*. IEEE, 2020, pp. 1–8.

[32] A. Domahidi, "Methods and tools for embedded optimization and control," Ph.D. dissertation, Eidgenössische Technische Hochschule Zürich, 2013.

[33] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6741–6747.

[34] F. Jiang, H. Baoyin, and J. Li, "Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, 2012.

[35] E. Taheri, I. Kolmanovsky, and E. Atkins, "Enhanced Smoothing Technique for Indirect Optimization of Minimum-Fuel Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 11, 2016.

Andrea Carlo Morelli is a PhD student at the Deep-space Astrodynamics Research and Technology (DART) group, Politecnico di Milano, working on autonomous guidance of interplanetary CubeSats in the context of the ERC-funded project EXTREMA. He obtained his Bachelor of Science in Aerospace Engineering from Politecnico di Milano in 2018. He has been a student of the merit program Alta Scuola Politecnica and earned his Master of Science in Space Engineering at Politecnico di Milano in 2020. He spent seven months at ISAE-Supaero in Toulouse in the context of a European exchange program.

Christian Hofmann has recently completed his PhD in Aerospace Engineering in the DART group at Politecnico di Milano, Italy, where he was working on trajectory optimization and the autonomous guidance of interplanetary CubeSats. He holds a Master of Science in Aerospace Engineering from the Technical University of Braunschweig and a Bachelor of Science in Aerospace Computer Science from the University of Würzburg in Germany. He worked as a research assistant at the German Aerospace Center (DLR) before he joined the DART group in November 2018. His research interests include astrodynamics, nonlinear optimal control, interplanetary CubeSats, space trajectory optimization and computational guidance and control.

Francesco Topputo is a Full Professor of Space Systems at Politecnico di Milano, Italy, and holds a position as Visiting Professor at TU Delft, The Netherlands. His core research activities involve spacecraft flight dynamics, interplanetary CubeSat mission and system design, autonomous guidance, navigation, and control. Dr. Topputo is an ERC laureate (CoG 2019) and has been PI in 14 research projects. He leads the DART group at Politecnico di Milano. He has published 55 peer-reviewed articles in international journals and over 180 works in total.