2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Successive Convexification with Feasibility Guarantee via Augmented Lagrangian for Non-Convex Optimal Control Problems

Kenshiro Oguri

Abstract— **This paper proposes an algorithm that solves non-convex optimal control problems with a theoretical guarantee for global convergence to a feasible local solution of the original problem. The proposed algorithm extends the recently proposed successive convexification (`SCvx`) algorithm to address its key limitation: lack of feasibility guarantee to the original non-convex problem. The main idea of the proposed algorithm is to incorporate the `SCvx` iteration into an algorithmic framework based on the augmented Lagrangian method to enable the feasibility guarantee while retaining favorable properties of `SCvx`. Unlike the original `SCvx`, our approach iterates on both of the optimization variables and the Lagrange multipliers, which facilitates the feasibility guarantee as well as efficient convergence, in a spirit similar to the alternating direction method of multipliers (ADMM). Convergence analysis shows the proposed algorithm's strong global convergence to a *feasible* local optimum of the original problem and its convergence rate. These theoretical results are demonstrated via numerical examples with comparison against the original `SCvx` algorithm.**

I. Introduction

This paper proposes a new algorithm for solving non-convex optimal control problems by extending the successive convexification (`SCvx`) algorithm [1], [2], a recent algorithm based on sequential convex programming (SCP), and presents a convergence analysis of the proposed algorithm. Most of the real-world problems are non-convex, as seen in aerospace, robotics, and other engineering applications, due to their nonlinear dynamics and/or non-convex constraints. While lossless convexification is available for a certain class of problems [3]–[5], many remain non-convex. Among various options to tackle non-convex problems [6], SCP is gaining renewed interest as a powerful tool in light of the recent advance in convex programming [1], [2], [7]–[10].

Any SCP algorithms take the approach that repeats the *convexify-and-solve* process to march toward a local solution of a non-convex problem. This approach is common to many optimization algorithms, including difference of convex programming, which decomposes a problem into convex and concave parts and approximates the concave part [11]; and sequential quadratic programming [12], which is adopted in many nonlinear programming software, such as SNOPT [13].

Among recent SCP algorithms, `SCvx` [1], [2] (and its variation, `SCvx-fast` [14]) and guaranteed sequential trajectory optimization (`GuSTO`) [8], [9] are arguably the most notable ones due to their rigorous theoretical underpinnings, with successful application to various problems [15], [16]. See [7] for a comprehensive review. In particular, [1] (`SCvx`) and [8] (`GuSTO`) analyze the performance of these algorithms in depth, and provide theoretical guarantees on their powerful capabilities, through different approaches: the Karush–Kuhn–Tucker (KKT) conditions for `SCvx` while Pontryagin's minimum principle for `GuSTO`. On the other hand, like any algorithms, each algorithm has their own limitations, including: the convergence guarantee to a KKT point of the penalty problem but not of the original problem (i.e., lack of feasibility guarantee) in `SCvx` [1]; the requirement on the dynamical systems to be control-affine in `GuSTO` [8]. Note that `SCvx` achieves a solution to the original problem for a "sufficiently large" penalty weight [1], which the user needs to find through trials and errors.

The objective of this paper is to fill the gap in those theoretical aspects of the existing SCP algorithms by building on the algorithmic foundation laid by `SCvx`. This study proposes a new SCP algorithm that guarantees feasibility to the *original* problem while retaining `SCvx`'s favorable properties, including the minimal requirements on the dynamical system. The main idea is to integrate the `SCvx` iteration into an algorithmic framework of the augmented Lagrangian (AL) method [17]. The AL method is a nonlinear programming technique proposed by Hestenes [18] and Powell [19] in 1960s to iteratively improve the multiplier estimate, addressing drawbacks of the quadratic penalty method [17].

The proposed algorithm iterates on the original optimization variables *and* the multipliers of the associated Lagrangian function in the primal-dual formalism. This facilitates the feasibility guarantee as well as efficient convergence, in a spirit similar to the alternating direction method of multipliers (ADMM) [20]. The proposed algorithm is named `SCvx*`, as it inherits key properties of `SCvx` and augments it by the feasibility guarantee, represented by "$\star$". A preliminary version of `SCvx*` has been successfully applied to space trajectory optimization under uncertainty [21], although the present paper is the first to provide comprehensive, rigorous convergence analysis of `SCvx*`.

The main contributions of `SCvx*` are threefold. Under the Assumption 1 in Section II-A, the proposed algorithm

1) provides the convergence guarantee to a *feasible* local optimum of the original problem, eliminating the need of trials and errors for tuning the penalty weight;
2) provides the strong global convergence to a single local solution of the original problem with minimal requirements on the problem structure; and
3) provides the linear/superlinear convergence rate of Lagrange multipliers with a slight algorithm modification.

K. Oguri is with the School of Aeronautics and Astronautics, Purdue University, IN 47907, USA (koguri@purdue.edu).

II. Preliminary

A. Problem Statement

We consider solving discrete-time non-convex optimal control problems given by Problem 1, where $x_s \in \mathbb{R}^{n_x}$ and $u_s \in \mathbb{R}^{n_u}$ represent the state and control at s-th time instance ($n_x, n_u \in \mathbb{N}$). $g_{\text{affine}}(\cdot)$ and $g_i(\cdot)$ represent the affine and non-affine equality constraint functions while $h_{\text{cvx}}(\cdot)$ and $h_j(\cdot)$ the convex and non-convex inequality constraint functions. $N, p, q \in \mathbb{N}$ are the number of discrete time steps, equality constraints, and inequality constraints, respectively. $f_0(\cdot)$ is assumed to be strictly convex and continuously differentiable in x_s, u_s $\forall s$, without loss of generality.[1]

Problem 1 (Non-convex Optimal Control Problem)**.**

$$\begin{aligned} \min_{x,u} \quad & f_0(x,u) \\ \text{s.t.} \quad & x_{s+1} = f_s(x_s,u_s), \quad s = 1,2,...,N-1, \\ & g_i(x,u) = 0, \quad i = 1,2,...,p, \\ & h_j(x,u) \le 0, \quad j = 1,2,...,q, \\ & g_{\text{affine}}(x,u) = 0, \quad h_{\text{cvx}}(x,u) \le 0 \end{aligned}$$

where $x = [x_0^\top, x_1^\top, ..., x_N^\top]^\top$, $u = [u_0^\top, u_1^\top, ..., u_{N-1}^\top]^\top$.

Defining $z = [x^\top, u^\top]^\top \in \mathbb{R}^{n_z}$, where $n_z = n_x(N+1) + n_u N$, Problem 1 can be cast as a non-convex optimization problem given in Problem 2, where $g_i(\cdot)$ incorporates the dynamical constraints $x_{s+1} = f_s(x_s, u_s)$.

Problem 2 (Non-convex Optimization Problem)**.**

$$\begin{aligned} \min_{z} \quad & f_0(z) \\ \text{s.t.} \quad & g_i(z) = 0, \quad i = 1,2,...,p + Nn_x, \\ & h_j(z) \le 0, \quad j = 1,2,...,q, \\ & g_{\text{affine}}(z) = 0, \quad h_{\text{cvx}}(z) \le 0 \end{aligned}$$

Define $g_{\text{all}}(\cdot)$ and $h_{\text{all}}(\cdot)$ to represent all the equality and inequality constraints in Problem 2 as: $g_{\text{all}}(z) = [g(z)^\top, g_{\text{affine}}(z)^\top]^\top$ and $h_{\text{all}}(z) = [h(z)^\top, h_{\text{cvx}}(z)^\top]^\top$, where $g = [g_1, ..., g_{p+Nn_x}]^\top$ and $h = [h_1, ..., h_q]^\top$ are vectorized non-affine and non-convex constraint functions.

Relevant conditions for a local minimum of Problem 2 are provided in Theorem 1 and Theorem 2. These Theorems are obtained by introducing the Lagrangian function $\mathcal{L}(\cdot)$ with Lagrange multiplier vectors λ and μ of appropriate size as:

$$\mathcal{L}(z, \lambda, \mu) = f_0(z) + \lambda \cdot g_{\text{all}}(z) + \mu \cdot h_{\text{all}}(z) \tag{1}$$

and then applying Theorem 12.1 of [6] and Theorem 12.6 of [6] to Problem 2, respectively. $(\cdot)$ is the dot product operator.

Theorem 1 (First-order Necessary Conditions)**.** *Suppose that z_* solves Problem 2 and that the linear independent constraint qualification (LICQ) holds at z_*. Then there exist Lagrange multiplier vectors λ_* and μ_* such that the KKT conditions are satisfied, i.e., $\nabla_z \mathcal{L}(z_*, \lambda_*, \mu_*) = 0$, $[g_{\text{all}}(z_*)]_i = 0$, $[h_{\text{all}}(z_*)]_j \le 0$, $[\mu_*]_j \ge 0$, and $[\mu_*]_j [h_{\text{all}}(z_*)]_j = 0$, $\forall i, j$.*

[1] If the original objective does not satisfy the assumption, one can introduce a new variable (say τ) and turning the non-convex objective into a non-convex inequality constraint bounded above by τ.

Theorem 2 (Second-order Sufficient Conditions)**.** *Suppose that for some feasible point z_* of Problem 2, there exist λ_* and μ_* that satisfy the KKT conditions given in Theorem 1 and that $\nabla^2_{zz}\mathcal{L}(z_*, \lambda_*, \mu_*)$ is positive definite on the plane tangent to the constraints, i.e., $v^\top \nabla^2_{zz}\mathcal{L}(z_*, \lambda_*, \mu_*) v > 0$, $\forall v \in \{v \neq 0 \mid v^\top \nabla_z g_{\text{all}}(z_*) = 0, v^\top \nabla_z h_{\text{active}}(z_*) = 0\}$, where h_{active} denotes the active inequality constraint vector. Then z_* is a strict local solution for Problem 2.*

Assumption 1. A local solution z_* of Problem 2 together with a unique set of multiplier vectors λ_* and μ_* satisfies the standard second-order sufficient condition for constrained optimization given in Theorem 2. Around z_*, $g_{\text{all}}(\cdot)$ and $h_{\text{all}}(\cdot)$ are continuously differentiable and satisfy LICQ.

B. Augmented Lagrangian Method

The augmented Lagrangian method [17] augments the Lagrangian function Eq. (1) as:

$$\mathcal{L}_w(z, \lambda, \mu) = \mathcal{L}(\cdot) + \frac{w}{2} g_{\text{all}} \cdot g_{\text{all}} + \frac{w}{2} [h_{\text{all}}]_+ \cdot [h_{\text{all}}]_+ \tag{2}$$

where $w \in \mathbb{R}$, and $[x]_+ = \max\{0, x\}$, working element-wise if x is a vector. This work takes advantage of the property of the augmented Lagrangian method that guarantees the convergence of the variable z and multipliers λ, μ to the optimum, z_*, λ_*, μ_*, even if the minimization of $\mathcal{L}_w(\cdot)$ is inexact at each iteration, provided that a few assumptions are met. Lemma 1 gives a summary of this favorable property and the required assumptions in a form tailored to Problem 2. The superscript (k) denotes the quantities at k-th iteration.

Lemma 1 (Augmented Lagrangian Convergence with Inexact Minimization)**.** *Suppose that for Problem 2, a sequence of $\{z_*^{(k)}\}$ satisfies $\|\nabla_z \mathcal{L}_{w^{(k)}}(z_*^{(k)}, \lambda^{(k)}, \mu^{(k)})\|_2 \le \delta^{(k)}$ where $\delta^{(k)} \to 0$ and $\{\lambda^{(k)}, \mu^{(k)}, w^{(k)}\}$ are updated as:*

$$\lambda^{(k+1)} = \lambda^{(k)} + w^{(k)} g_{\text{all}}(z_*^{(k)}), \tag{3a}$$

$$\mu^{(k+1)} = [\mu^{(k)} + w^{(k)} h_{\text{all}}(z_*^{(k)})]_+, \tag{3b}$$

$$w^{(k+1)} = \beta w^{(k)} \quad (\beta > 1) \tag{3c}$$

where $\{\lambda^{(k)}, \mu^{(k)}\}$ are bounded. Then, $w^{(k)}$ eventually exceeds a threshold w_* that gives $\nabla^2_{zz}\mathcal{L}_{w_*}(z_*^{(k)}, \lambda^{(k)}, \mu^{(k)}) \succ 0$, and any sequence $\{z_*^{(k)}, \lambda^{(k)}, \mu^{(k)}\}$ globally converges to a local optimum of Problem 2, $\{z_*, \lambda_*, \mu_*\}$.

Proof. The proof is by applying Propositions 2.14, 3.1, and 3.2 of [17] to Problem 2 (see [22] for an explicit discussion about the global convergence) under Assumption 1. □

Noting that $z_*^{(k)}$ represents an approximate minimizer of $\mathcal{L}_{w^{(k)}}(\,\cdot\,, \lambda^{(k)}, \mu^{(k)})$, Lemma 1 clarifies that inexact minimization at each augmented Lagrangian iteration must be asymptotically exact, i.e., $\|\nabla_z \mathcal{L}_w(\cdot)\|_2 \to 0$ as $k \to \infty$.

III. The Proposed Algorithm: SCvx*

This section presents the proposed algorithm, SCvx*. The convergence analysis of SCvx* is given in Section IV.

A. Non-convex Penalty Problem with Augmented Lagrangian

While the augmented Lagrangian function Eq. (2) is introduced from the viewpoint of primal-dual formalism, it can be also viewed from a penalty method standpoint. Adopting this viewpoint, Eq. (2) can be equivalently expressed as $\mathcal{L}_w(z, \lambda, \mu) = f_0(z) + P(g_{\text{all}}(z), h_{\text{all}}(z), w, \lambda, \mu)$, where $P(g, h, w, \lambda, \mu)$ denotes the penalty function defined as:

$$P(\cdot) = \lambda \cdot g + \frac{w}{2} g \cdot g + \mu \cdot h + \frac{w}{2} [h]_+ \cdot [h]_+ \tag{4}$$

With the penalty function of the form given by Eq. (4), we now formulate our non-convex penalty problem based on Problem 2. As our algorithm is based on SCP, our penalty problem penalizes the violations of non-convex constraints only ($\because$ convex constraints are imposed in each convex programming); hence, redefine the Lagrange multipliers as

$$\lambda = [\lambda_1, \lambda_2, ..., \lambda_{p+Nn_x}]^\top, \ \mu = [\mu_1, \mu_2, ..., \mu_q]^\top \geq 0. \tag{5}$$

This leads to our non-convex penalty problem, Problem 3.

Problem 3 (Non-convex penalty problem with AL)**.**

$$\begin{aligned} \min_{z} \quad & J(z) = f_0(z) + P(g(z), h(z), w, \lambda, \mu) \\ \text{s.t.} \quad & g_{\text{affine}}(z) = 0, \ h_{\text{cvx}}(z) \leq 0 \end{aligned}$$

where $g = [g_1, g_2, ..., g_{p+Nn_x}]^\top$ *and* $h = [h_1, h_2, ..., h_q]^\top$.

B. Convex Penalty Problem with Augmented Lagrangian

Problem 3 is clearly non-convex due to the nonlinearity and non-convexity of $g(\cdot)$ and $h(\cdot)$. To solve the problem via SCP, we linearize them about a reference variable $\bar{z}$ at each iteration, which yields $\widetilde{g} = 0$ and $\widetilde{h} \leq 0$, where

$$\begin{aligned} \widetilde{g}(z) &= g(\bar{z}) + \nabla_z g(\bar{z}) \cdot (z - \bar{z}), \\ \widetilde{h}(z) &= h(\bar{z}) + \nabla_z h(\bar{z}) \cdot (z - \bar{z}) \end{aligned} \tag{6}$$

However, imposing $\widetilde{g} = 0$ and $\widetilde{h} \leq 0$ in the convex subproblem can lead to the issue of *artificial infeasibility* [2], and hence these linearized constraints are relaxed as follows:

$$\widetilde{g}(z) = \xi, \quad \widetilde{h}(z) \leq \zeta \tag{7}$$

where $\xi \in \mathbb{R}^{p+Nn_x}$, $\zeta \in \mathbb{R}^q$, and $\zeta \geq 0$. Although the original SCvx literature [1], [2] introduces *virtual control* and *virtual buffer* terms separately, the former is naturally incorporated in ξ. It is easy to verify this; noting that linearized dynamical constraints are given by

$$x_{s+1} = A_s x_s + B_s u_s + c_s + E_s \xi, \quad s = 1, 2, ..., N \tag{8}$$

where $E_s \in \mathbb{R}^{n_x \times p+Nn_x}$ extracts the virtual control term at s-th time instance from ξ, and

$$\begin{aligned} A_s &= \nabla_x f_s(\bar{x}_s, \bar{u}_s), \ B_s = \nabla_u f_s(\bar{x}_s, \bar{u}_s), \\ c_s &= f_s(\bar{x}_s, \bar{u}_s) - A_s \bar{x}_s - B_s \bar{u}_s \end{aligned} \tag{9}$$

it is clear that Eq. (8) can be incorporated into $\widetilde{g}(z) = \xi$.

On the other hand, the linearization and constraint relaxation lead to another issue called *artificial unboundedness* [1]. To avoid this, we impose a constraint on the variable update magnitude with a trust region bound $r > 0$, given by

$$\|\bar{z} - z\|_\infty \leq r \tag{10}$$

The trust region method is common in many algorithms for nonlinear programming [6]. This prevents the optimizer from exploring the solution space "too far" from $\bar{z}$.

Problem 4 gives the convex subproblem at each iteration. Problem 4 is convex in z, ξ, ζ since $[x]_+^2 = (\max\{0, x\})^2$, which appears in Eq. (4), is convex in $x \in \mathbb{R}$.

Problem 4 (Convex penalty subproblem with AL)**.**

$$\begin{aligned} \min_{z,\xi,\zeta} \quad & L(z, \xi, \zeta) = f_0(z) + P(\xi, \zeta, w, \lambda, \mu) \\ \text{s.t.} \quad & \widetilde{g}(z) = \xi, \ \widetilde{h}(z) \leq \zeta, \ \zeta \geq 0, \\ & \|\bar{z} - z\|_\infty \leq r, \ g_{\text{affine}}(z) = 0, \ h_{\text{cvx}}(z) \leq 0 \end{aligned}$$

Algorithm 1 SCvx⋆

Input: $\bar{z}^{(1)}, r^{(1)}, w^{(1)}, \epsilon_{\text{opt}}, \epsilon_{\text{feas}}, \rho_0, \rho_1, \rho_2, \alpha_1, \alpha_2, \beta, \gamma$

$k = 1$, $\Delta J^{(0)} = \chi^{(0)} = \delta^{(1)} = \infty$, $\lambda^{(1)} = \mu^{(1)} = 0$
while $\Delta J^{(k-1)} > \epsilon_{\text{opt}}$ or $\chi^{(k-1)} > \epsilon_{\text{feas}}$ **do**
  $\{\widetilde{g}^{(k)}, \widetilde{h}^{(k)}\} \leftarrow$ derived via Eq. (6) at $\bar{z}^{(k)}$
  $\{z_*^{(k)}, \xi_*^{(k)}, \zeta_*^{(k)}\} \leftarrow$ solve Problem 4
  $\{\Delta J^{(k)}, \Delta L^{(k)}, \chi^{(k)}\} \leftarrow$ Eq. (13)
  if $\Delta L^{(k)} = 0$ **then**
    $\rho^{(k)} \leftarrow 1$
  else
    $\rho^{(k)} \leftarrow \Delta J^{(k)} / \Delta L^{(k)}$
  $\{\bar{z}, w, \lambda, \mu, \delta\}^{(k+1)} \leftarrow \{\bar{z}, w, \lambda, \mu, \delta\}^{(k)}$ ▷ *default*
  if $\rho^{(k)} \geq \rho_0$ **then** ▷ *accept the step*
    $\bar{z}^{(k+1)} \leftarrow z_*^{(k)}$ ▷ *solution update*
    if Eq. (15) is satisfied **then**
      $\{\lambda, \mu, w\}^{(k+1)} \leftarrow$ Eq. (3) ▷ *multiplier update*
      $\delta^{(k+1)} \leftarrow$ Eq. (16) ▷ *stationarity tol. update*
  $r^{(k+1)} \leftarrow$ Eq. (17) ▷ *trust region update*
  $k \leftarrow k + 1$
return $(z_*^{(k)}, \lambda^{(k)}, \mu^{(k)})$

C. SCvx⋆ Algorithm

We are now ready to present the proposed SCvx⋆ algorithm. Algorithm 1 summarizes SCvx⋆. The key steps of Algorithm 1 are discussed in the rest of this section.

1) Successive linearization: Let us compactly express the penalty function Eq. (4) at k-th iteration as:

$$P^{(k)}(g, h) \triangleq P(g, h, w^{(k)}, \lambda^{(k)}, \mu^{(k)}) \tag{11}$$

Likewise, the penalized objectives of Problems 3 and 4 at k-th iteration are expressed as:

$$\begin{aligned} J^{(k)}(z) &\triangleq f_0(z) + P^{(k)}(g(z), h(z)) \\ L^{(k)}(z, \xi, \zeta) &\triangleq f_0(z) + P^{(k)}(\xi, \zeta) \end{aligned} \tag{12}$$

Given a user-provided initial reference point $\bar{z}^{(1)}$, the linearization process follows Section III-B, which instantiates Problem 4 at each iteration. Problem 4 is solved to convergence, yielding the solution at k-th iteration, $z_*^{(k)}, \xi_*^{(k)}, \zeta_*^{(k)}$.

Every time after Problem 4 is solved, SCvx⋆ calculates:

$$\Delta J^{(k)} = J^{(k)}(\bar{z}^{(k)}) - J^{(k)}(z_*^{(k)}) \tag{13a}$$

$$\Delta L^{(k)} = J^{(k)}(\bar{z}^{(k)}) - L^{(k)}(z_*^{(k)}, \xi_*^{(k)}, \zeta_*^{(k)}) \tag{13b}$$

$$\chi^{(k)} = \|g(z_*^{(k)}), [h(z_*^{(k)})]_+\|_2 \tag{13c}$$

where $\Delta J^{(k)}, \Delta L^{(k)}$, and $\chi^{(k)}$ represent the actual cost reduction, predicted cost reduction, and the infeasibility.

2) Step acceptance: After solving Problem 4, SCvx⋆ accepts the solution and updates $\bar{z}^{(k)}$ if a certain criterion is met. With $\rho_0 \in (0, 1)$, the acceptance criterion is given by

$$\rho_0 \leq \rho^{(k)}, \quad \rho^{(k)} = \Delta J^{(k)} / \Delta L^{(k)} \tag{14}$$

where $\rho^{(k)}$ measures the relative decrease of the objective; an iteration is accepted only if $\rho^{(k)}$ is greater than ρ_0, which helps avoid accepting bad steps (e.g., those which do not improve the non-convex objective). This criterion is based on the original SCvx [2], but not exactly the same; this point is made precise in the following remark.

Remark 1. *The definition of $\Delta L^{(k)}$ in Eq.* (13b) *is different from SCvx [1], [2]. As SCvx considers a fixed penalty weight, their definition of ΔL with our notation corresponds to $J^{(k-1)}(\bar{z}^{(k)}) - L^{(k)}(z_*^{(k)}, \xi_*^{(k)}, \zeta_*^{(k)})$, which is not always non-negative because $J^{(k)}(\bar{z}^{(k)}) \neq J^{(k-1)}(\bar{z}^{(k)})$. With the careful definition of $\Delta L^{(k)}$ as in Eq.* (13b)*, a key result $\Delta L^{(k)} \geq 0$ is guaranteed in SCvx⋆, as proved in Lemma 3.*

3) Lagrange multiplier update: Although the multipliers λ and μ are fixed in each convex subproblem, they must be updated to march toward the convergence of Problem 2. SCvx⋆ updates λ and μ when the current iteration is accepted *and* the following condition is met:

$$|\Delta J^{(k)}| < \delta^{(k)}, \tag{15}$$

where $\delta^{(k)} \in \mathbb{R}$ is updated such that $\delta^{(k)} \to 0$ as $k \to \infty$. The motivation behind this criterion is to satisfy the asymptotically exact minimization requirement clarified in Lemma 1. A simple design for updating $\delta^{(k)}$ is:

$$\delta^{(k+1)} = \begin{cases} |\Delta J^{(k)}| & \text{if } \delta^{(k)} = \infty \\ \gamma \delta^{(k)} & \text{otherwise} \qquad (\gamma \in (0,1)) \end{cases} \tag{16}$$

when Eq. (15) is met. Any other scheme than Eq. (16) may be used as long as it satisfies $\delta^{(k)} \to 0$ as $k \to \infty$.

Every time when Eq. (15) is met, SCvx⋆ updates $w^{(k)}$, $\lambda^{(k)}$, $\mu^{(k)}$ using Eq. (3), where $g_{\text{all}}(\cdot)$ and $h_{\text{all}}(\cdot)$ must be replaced by $g(\cdot)$ and $h(\cdot)$. Section IV-A shows that this scheme ensures satisfying the convergence conditions in Lemma 1. A stricter condition than Eq. (15) is also possible to guarantee the convergence rate, as shown in Section IV-B.

4) Trust region update: The trust region radius r plays an important role in preventing artificial unboundedness. $\rho^{(k)}$ in Eq. (14) is used to quantify the quality of the current radius $r^{(k)}$. Like original SCvx [2], given the user-defined initial radius $r^{(1)} > 0$ and thresholds ρ_1, $\rho_2 \in \mathbb{R}$ that satisfy $\rho_0 < \rho_1 < \rho_2$, SCvx⋆ updates $r^{(k)}$ as follows:

$$r^{(k+1)} = \begin{cases} \max\{r^{(k)}/\alpha_1, r_{\min}\} & \text{if } \rho^{(k)} < \rho_1 \\ r^{(k)} & \text{elseif } \rho^{(k)} < \rho_2 \\ \min\{\alpha_2 r^{(k)}, r_{\max}\} & \text{else} \end{cases} \tag{17}$$

where $\alpha_1 > 1$ and $\alpha_2 > 1$ determine the contracting and enlarging ratios of $r^{(k)}$, respectively, and $0 < r_{\min} < r_{\max}$. Although the convergence proof in Section IV-A does not require $r^{(k)}$ be bounded from above in theory, SCvx⋆ implements the upper bound $r_{\max}$ for numerical stability.

5) Convergence check: SCvx⋆ detects the convergence to Problem 2 and terminates the iteration if:

$$\Delta J^{(k)} \leq \epsilon_{\text{opt}} \quad \wedge \quad \chi^{(k)} \leq \epsilon_{\text{feas}} \tag{18}$$

where $\epsilon_{\text{opt}}, \epsilon_{\text{feas}} \in \mathbb{R}$ are small positive user-defined scalars representing the optimality and feasibility tolerances.

IV. Convergence Analysis

This section presents the convergence analysis of SCvx⋆. Section IV-A shows the global strong convergence to Problem 2 while Section IV-B discusses its convergence rate.

A. Convergence

Let us first introduce Lemma 2.

Lemma 2 (Local Optimality Necessary Condition)**.** *If z_* is a local minimizer of $J^{(k)}$ in Problem 3, then z_* is a stationary point of $J^{(k)}$ with the current $w^{(k)}, \lambda^{(k)}, \mu^{(k)}$.*

Proof. Apply Theorem 2.2 of [6] to Problem 3. □

We then present Lemma 3, which states the non-negativity of $\Delta L^{(k)}$ as well as the stationarity of $J^{(k)}$ when $\Delta L^{(k)} = 0$. This extends Theorem 3 of [2] (also Theorem 3.10 of [1]) to account for the effect of varying $w^{(k)}, \lambda^{(k)}, \mu^{(k)}$.

Lemma 3. *The predicted cost reductions $\Delta L^{(k)}$ in Eq.* (13b) *satisfy $\Delta L^{(k)} \geq 0$ for all k. Also, $\Delta L^{(k)} = 0$ implies that the reference point $z = \bar{z}^{(k)}$ is a stationary point of $J^{(k)}$.*

Proof. Since $(z_*^{(k)}, \xi_*^{(k)}, \zeta_*^{(k)})$ solves Problem 4, we have

$$\begin{aligned} L^{(k)}(z_*^{(k)}, \xi_*^{(k)}, \zeta_*^{(k)}) &\leq L^{(k)}(\bar{z}^{(k)}, g(\bar{z}^{(k)}), h(\bar{z}^{(k)})) \\ &= f_0(\bar{z}^{(k)}) + P^{(k)}(g(\bar{z}^{(k)}), h(\bar{z}^{(k)})) = J^{(k)}(\bar{z}^{(k)}) \end{aligned} \tag{19}$$

Thus, it implies that $\Delta L^{(k)} \geq 0$ for all k and that $\Delta L^{(k)} = 0$ holds if and only if $z_*^{(k)} = \bar{z}^{(k)}$. From this, $\Delta L^{(k)} = 0$ implies that $z = \bar{z}^{(k)}$ is a local minimizer of $J^{(k)}$, and hence, from Lemma 2, a stationary point of $J^{(k)}$. □

Lemma 3 is a key for SCvx⋆ to inherit two favorable aspects of the original SCvx algorithm, namely, (1) the assured acceptance of iteration and (2) the assured stationarity of limit points. Lemmas 4 and 5 clarify these two aspects in the context of SCvx⋆ by extending those of SCvx.

Lemma 4. *The SCvx⋆ iterations are guaranteed to be accepted (i.e., Line 11 is satisfied) within a finite number of iterations after an iteration is rejected.*

Proof. The proof is straightforward by combining Lemmas 2 and 3 and the proof for Lemma 3 of [2] (or Lemma 3.11 of [1]), where the generalized differential and the generalized directional derivative can be replaced with the gradient and directional derivative (∵ unlike `SCvx`, the penalty function of `SCvx⋆` is differentiable due to the formulation based on the augmented Lagrangian method). □

Lemma 5. *A sequence $\{z_*^{(k)}\}$ generated by `SCvx⋆` when the Lagrange multipliers and penalty weight are fixed is guaranteed to have limit points, and any limit point $\hat{z}$ is a stationary point of Problem 3.*

Proof. The proof is straightforward by combining Lemmas 3 and 4 and the proof for Theorem 4 of [2] (or Theorem 3.13 of [1]), where note from Eq. (17) that $r^{(k)} \geq r_{\min} > 0$. □

Remarkably, Lemma 5 implies that, when the values of $\lambda^{(k)}, \mu^{(k)}, w^{(k)}$ remain fixed, we have $\Delta J^{(k_i)} \to 0$ as $i \to \infty$, where $\{z_*^{(k_i)}\}$ is a subsequence of $\{z_*^{(k)}\}$. This assures the satisfaction of Line 13 within a finite (typically a few) number of iterations after $\lambda^{(k)}, \mu^{(k)}, w^{(k)}$ are last updated. This key property is formally stated in Lemma 6.

Lemma 6. *The `SCvx⋆` multipliers and penalty weights are guaranteed to be updated (i.e., Line 13 is satisfied) within a finite number of iterations after their last update.*

Proof. The proof is by contradiction. Suppose Line 13 is not satisfied for indefinite number of iterations, i.e., $|\Delta J^{(k)}| \geq \delta^{(k)}$ for $k \to \infty$. It implies $\lambda^{(k)}, \mu^{(k)}, w^{(k)}$ remain the same for $k \to \infty$. However, when $\lambda^{(k)}, \mu^{(k)}, w^{(k)}$ remain the same values, there is at least one subsequence with $\Delta J^{(k_i)} \to 0$ due to Lemma 5, which eventually satisfies $|\Delta J^{(k)}| < \delta^{(k)}$ for any $\delta^{(k)} > 0$ without requiring infinite k. This contradicts $|\Delta J^{(k)}| \geq \delta^{(k)}$ for $k \to \infty$, and thus implies Lemma 6. □

We are now ready to present the main result of this paper on the convergence property of the `SCvx⋆` algorithm.

Theorem 3 (Global Strong Convergence with Feasibility)**.** *`SCvx⋆` achieves global convergence to a feasible local optimum of the original problem, Problem 2.*

Proof. Let $\{z_*^{(k_i)}\}$ be a subsequence of $\{z_*^{(k)}\}$ that consists of the iterations where the multipliers are updated; such subsequences are guaranteed to exist due to Lemma 6. Then, Eq. (16) ensures $\delta^{(k_i)} > \delta^{(k_i+1)}$, and due to Lemma 5, $\Delta J^{(k_i)} \to 0$ and $\delta^{(k_i)} \to 0$ in the limit. Again due to Lemma 5, the limit point is a stationary point of Problem 3, satisfying $\nabla_z J^{(k)} = 0$. Then, $\nabla_z \mathcal{L}_{w^{(k)}} \to 0$ also holds in the limit since every $z_*^{(k)}$ satisfies $g_{\text{affine}} = 0$ and $h_{\text{cvx}} \leq 0$ within convex programming. Thus, the `SCvx⋆` iteration guarantees $\|\nabla_z \mathcal{L}_{w^{(k)}}\|_2 \to 0$ in the limit, with the multiplier update Eq. (3). Therefore, as $w^{(k)}$ exceeds the threshold w_* given in Lemma 1 after finite iterations, `SCvx⋆` achieves the global convergence to a feasible optimum of Problem 2. □

Remarks below discuss two key improvements that the `SCvx⋆` algorithm provides over the original `SCvx` algorithm.

Remark 2 (Feasibility)**.** *The converged solution generated by `SCvx⋆` is feasible to Problem 2, while the original `SCvx` algorithm does not provide such a feasibility guarantee.*

Remark 3 (Accelerated convergence)**.** *`SCvx⋆` iterates not only on the variable $z^{(k)}$ but also on the Lagrange multipliers $\lambda^{(k)}$ and $\mu^{(k)}$, which, besides providing the feasibility guarantee, facilitates the convergence by iteratively improving the multiplier estimate rather than using a fixed value.*

B. Convergence rate

Having the augmented Lagrangian method as the basis of the algorithm facilitates the analysis of the convergence rate of `SCvx⋆`. Based on [17], [22], linear or superlinear convergence rate of the Lagrangian multipliers can be achieved when $\delta^{(k)}$ decreases to zero as fast as $\|\lambda^{(k)} - \lambda_*\|_2 / w^{(k)}$. To achieve this, we may replace Eq. (15) by

$$|\Delta J^{(k)}| \leq \min\{\delta^{(k)}, \eta \chi^{(k)}\}, \tag{20}$$

where η is a positive scalar. With this criterion, Proposition 2 of [22] states that the augmented Lagrange multiplier iteration Eq. (3) converges to z_*, λ_*, μ_* superlinearly if $w^{(k)} \to \infty$, and linearly if $w^{(k)} \to w_{\max} < \infty$, where $w_{\max} \in (0, \infty)$ is the upper bound of the penalty weight. It must be noted that these convergence rates are about the Lagrange multiplier iteration but not with respect to k.

The choice of η can be arbitrary to achieve the above convergence rate in theory. A simple yet effective approach is to initialize η by ∞ at first, and then update it by $\eta \leftarrow |\Delta J^{(k)}| / \chi^{(k)}$ when Eq. (20) is met for the first time.

Here, we must ensure that `SCvx⋆` retains the favorable property of guaranteed multiplier update (Lemma 6) under the stricter condition Eq. (20). Lemma 7 addresses this. Once the convergence in λ, μ is achieved, then λ, μ will not be updated anymore while $z^{(k)}$ converges to a feasible local minimum of Problem 2 due to Lemmas 1 and 5.

Lemma 7. *Suppose that Eq.* (20) *instead of Eq.* (15) *is used for the multiplier update criterion. Then, until the convergence in λ and μ is achieved, the `SCvx⋆` iteration guarantees that the values of λ and μ are updated within a finite number of iterations after their last update.*

Proof. For conciseness, the proof is focused on problems with equality constraints only, as any inequality constraints can be converted to equality constraints by introducing dummy variables without changing the results in the augmented Lagrangian framework (see Section 3.1 of [17]). Thus, μ and h are not explicitly considered in this proof.

It is clear from Lemma 5 that the claim is true if $\chi^{(k)} > 0$ holds until the convergence in λ is achieved. It is also clear that $\lambda^{(k)} \neq \lambda_*$ until the convergence in λ is achieved. Thus, let us show $\chi^{(k)} > 0$ when $\lambda^{(k)} \neq \lambda_*$ by contradiction.

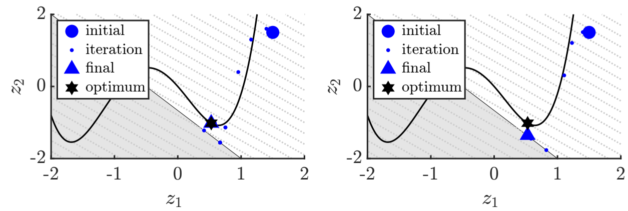
Suppose that there exists certain $\lambda^{(k)} (\neq \lambda_*)$ such that lead to $\chi^{(k)} = 0$. For $z_*^{(k)}$ that solves Problem 4, it is clear from Eq. (13c) that $\chi^{(k)} = 0$ if and only if $g(z_*^{(k)}) = 0$. Since $\chi^{(k)} = 0$, Eq. (20) is not satisfied, and hence the values of $\lambda^{(k)}$ remain fixed until $\Delta J^{(k)} = 0$ is achieved in the limit.

TABLE I
SCvx⋆ PARAMETERS ($\epsilon = \epsilon_{\text{opt}} = \epsilon_{\text{feas}}$)

ϵ	$\{\rho_0, \rho_1, \rho_2\}$	$\{\alpha_1, \alpha_2, \beta, \gamma\}$	$\{r^{(1)}, r_{\min}, r_{\max}\}$
10^{-5}	$\{0, 0.25, 0.7\}$	$\{2, 3, 2, 0.9\}$	$\{0.1, 10^{-10}, 10\}$

TABLE II
EXAMPLE 1: CONVERGENCE RESULTS (N/A: NON-CONVERGENCE)

$w^{(1)}$ value	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}	10^{5}
SCvx⋆ # ite.	39	33	31	42	40	51	56
SCvx # ite.	N/A	N/A	35	31	N/A	N/A	N/A



(a) SCvx⋆ result, 33 iterations (b) SCvx result, *not* converged

Fig. 1. Example 1 convergence behavior for $w^{(1)} = 1$ in the z_1-z_2 space. The black curve represents the non-convex equality constraint Eq. (21b), and the gray shaded area is the infeasible area from Eq. (21c).

Due to Lemma 5, the limit point is a stationary point of Problem 3, satisfying $0 = \nabla_z J^{(k)} = \nabla_z f_0 + (\lambda + wg) \cdot \nabla_z g$. Using $g = 0$ due to $\chi^{(k)} = 0$, this leads to $0 = \nabla_z f_0 + \lambda \cdot \nabla_z g$, which implies that the limit point of $\{z_*^{(k)}\}$ is a feasible stationary point of Problem 2, i.e., $\nabla_z \mathcal{L} = 0$, and hence satisfies the KKT conditions of Problem 2, since every $z_*^{(k)}$ also satisfies $g_{\text{affine}} = 0$ and associated multiplier conditions within convex programming. This contradicts $\lambda^{(k)} \neq \lambda_*$, and thus $\chi^{(k)} \neq 0$ by contradiction, implying $\chi^{(k)} > 0$ because $\chi^{(k)}$ must be non-negative. □

V. NUMERICAL EXAMPLES

This section presents numerical examples to demonstrate SCvx⋆ and compare the performance to SCvx. Note that Algorithm 1 boils down to SCvx by ignoring Lines 13 to 15 and replacing Eq. (4) by an l_1 penalty function $P(g, h, w) = w\|g\|_1 + w\|[h]_+\|_1$. CVX [23] is used with Mosek [24].

SCvx⋆ parameters commonly used for the two examples are listed in Table I. In each example, $w^{(1)}$ is varied to investigate the performance of SCvx⋆ and SCvx for different penalty weights. $w_{\max} = 10^8$ is set for SCvx⋆ to avoid numerical instability. If the algorithm does not converge in 100 iterations, it is terminated and deemed unconverged.

A. Example 1: Simple Problem with Crawling Phenomenon

The first example is a simple non-convex optimization problem from [25] to demonstrate that SCvx⋆ can also overcome the so-called *crawling phenomenon*, which is known to occur for a class of SCP algorithms. The non-convex problem from [25] is defined in the form of Problem 2 as follows:

$$\min_{-2 \leq z \leq 2} \quad z_1 + z_2 \tag{21a}$$
$$\text{s.t.} \quad z_2 - z_1^4 - 2z_1^3 + 1.2z_1^2 + 2z_1 = 0, \tag{21b}$$
$$-z_2 - (4/3)z_1 - 2/3 \leq 0 \tag{21c}$$

which is solved by SCvx⋆ and SCvx with various $w^{(1)}$. The same initial reference point as [25], $\bar{z}^{(1)} = [1.5, 1.5]$, is used.

Table II summarizes the comparison of SCvx⋆ and SCvx by listing the number of iterations required for convergence with respect to different values of $w^{(1)}$. "N/A" indicates non-convergence achieved within the maximum iteration limit (= 100). Table II illustrates that SCvx⋆ constantly achieves the convergence regardless of the initial values of $w^{(k)}$; this is in sharp contrast to the SCvx results, which successfully converge to a feasible local minimum only for the two cases: $w^{(1)} = 10$ and 100, emphasizing the sensitivity to the value of $w^{(1)}$ (which is held constant over iterations in SCvx).

Fig. 1 depicts the convergence behavior for $w^{(1)}$=1. It clarifies that the SCvx⋆ iteration successfully converges to the optimum while SCvx is not able to satisfy the non-convex equality constraint (on the black curve). This example illustrates two non-converging modes of SCvx: 1) non-improving feasibility for $w^{(1)}$=$10^{-1}, 10^0$; and 2) crawling phenomenon for $w^{(1)}$=$10^3, 10^4, 10^5$. In contrast, SCvx⋆ effectively addresses these non-converging modes, consistent with the two key improvements stated in Remarsks 2 and 3.

B. Example 2: Quad Rotor Path Planning

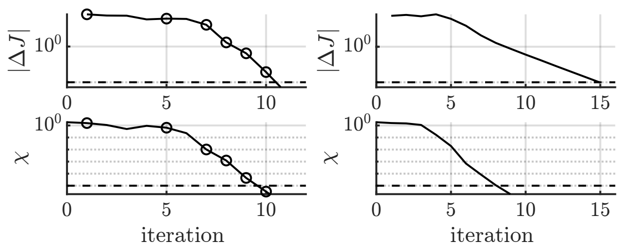
The second example is a quad-rotor non-convex optimal control problem from the original SCvx literature [1]. The problem is defined in the form of Problem 1 as follows:

$$\begin{aligned}
\min_{x,u} \quad & \sum_{s=1}^{N} \Gamma_s \Delta t \\
\text{s.t.} \quad & x_{s+1} = x_s + \int_{t_s}^{t_{s+1}} \begin{bmatrix} v \\ T/m - k_D\|v\|_2 v + g \end{bmatrix} \mathrm{d}t, \ \forall s \\
& \|p_s - p_{\text{obj},j}\|_2 \geq R_{\text{obs},j}, \quad j = 1, 2, \\
& x_1 = x_{\text{ini}}, \ x_N = x_{\text{fin}}, \ T_1 = T_N = -mg, \\
& [1 \ 0 \ 0] \cdot p_s = 0, \ \|T_s\|_2 \leq \Gamma_s, \ T_{\min} \leq \Gamma_s \leq T_{\max}, \\
& \cos\theta_{\max}\Gamma_s \leq [1 \ 0 \ 0] \cdot T_s, \quad \forall s
\end{aligned}$$

where $p, v \in \mathbb{R}^3$, and $m \in \mathbb{R}$ denote the position, velocity, and mass of the vehicle; $T \in \mathbb{R}^3$ is the thrust vector; $\Gamma \in \mathbb{R}$ represents the thrust magnitude (at convergence); $g = [-9.81, 0, 0]^\top \text{m/s}^2$ is the gravity acceleration; $k_D = 0.5$ is the drag coefficient; $p_{\text{obj},j}$ and $R_{\text{obj},j}$ are the position and radius of j-th obstacle (defined the same as [1]); $x_{\text{ini}} = [0\,\text{m}, 0\,\text{m}, 0\,\text{m}, 0\,\text{m/s}, 0.5\,\text{m/s}, 0\,\text{m/s}]^\top$ and $x_{\text{fin}} = [0\,\text{m}, 10\,\text{m}, 0\,\text{m}, 0\,\text{m/s}, 0.5\,\text{m/s}, 0\,\text{m/s}]^\top$ are the initial and final states; $\{T_{\min}, T_{\max}\} = \{1.0, 4.0\}\,\text{N}$; $\theta_{\max} = \pi/4$. The state and control variables are $x_s = [r_s^\top, v_s^\top]^\top \in \mathbb{R}^6$ and $u_s = [T_s^\top, \Gamma_s]^\top \in \mathbb{R}^4$, where the zeroth-order-hold control is used for the discretization, i.e., $u_s = u(t) \ \forall t \in [t_s, t_{s+1})$. $t_N = 5.0$ seconds with $N = 31$. For $\bar{z}^{(1)}$, the straight line that connects x_{init} and x_{fin} is used for x_s while $-mg$ and $\|mg\|_2$ are used for T_s and Γ_s, respectively.

TABLE III
EXAMPLE 2: CONVERGENCE RESULTS

$w^{(1)}$ value	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}	10^{5}
SCvx⋆ # ite.	24	17	14	11	11	11	14
SCvx # ite.	N/A	9	11	13	14	15	16



(a) SCvx⋆ result, 11 iterations (b) SCvx result, 15 iterations

Fig. 2. Example 2 convergence behavior for $w^{(1)} = 10^4$; y-axes represent the penalized objective improvement $|\Delta J^{(k)}|$ (top) and the infeasibility $\chi^{(k)}$ (bottom). The circles in (a) indicate iterations with multiplier updates.

Table III summarizes the convergence results for Example 2. This suggests that the performance of SCvx⋆ and SCvx are similar overall for this example, whereas a key difference is observed that SCvx struggles to converge to a feasible solution when $w^{(1)} = 10^{-1}$. SCvx⋆ constantly converges to a feasible local minimum irrelevant to the value of $w^{(1)}$. This property is favorable especially for large-scale problems, where the user may not afford to tune $w^{(1)}$ [21]. On the other hand, this also provides a reassuring result that, despite the lack of the theoretical feasibility guarantee, SCvx can also perform well and may be good enough for relatively simple, small-scale optimal control problems.

Fig. 2 presents the convergence behavior for $w^{(1)} = 10^4$ in terms of $|\Delta J^{(k)}|$ and $\chi^{(k)}$. The circles in Fig. 2(a) indicate the iterations when Line 13 of Algorithm 1 is satisfied and the multipliers are updated. The dashed lines represent the tolerance $\epsilon{=}\epsilon_{\rm opt}{=}\epsilon_{\rm feas}(= 10^{-5})$. While SCvx satisfies the constraints earlier, SCvx⋆ achieves the overall convergence faster, likely due to the iterative estimate of multipliers that balances the progress in optimality and feasibility.

VI. CONCLUSIONS

In this paper, a new SCP algorithm SCvx⋆ is proposed to address the lack of feasibility guarantee in SCvx by leveraging the augmented Lagrangian framework. Unlike SCvx, which uses a fixed penalty weight over iterations, SCvx⋆ iteratively improves both the optimization variables and the Lagrange multipliers, facilitating the convergence. Inheriting the favorable properties of SCvx and fusing those with the augmented Lagrangian method, SCvx⋆ provides strong global convergence to a *feasible* local optimum of the original non-convex optimal control problems with minimal requirements on the problem form. The convergence rate of SCvx⋆ is also analyzed, clarifying that linear/superlinear convergence rate with respect to the Lagrange multipliers can be achieved by slightly modifying the algorithm. These theoretical results are demonstrated via numerical examples.

REFERENCES

[1] Y. Mao, M. Szmuk, X. Xu, and B. Acikmese, "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems," *arXiv preprint*, 2019.

[2] Y. Mao, M. Szmuk, and B. Acikmese, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Conf. on Decis. and Control*, pp. 3636–3641, IEEE, 2016.

[3] B. Açıkmeşe and S. R. Ploen, "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *J. of Guid., Control and Dyn.*, vol. 30, no. 5, p. 1353–1366, 2007.

[4] B. Açıkmeşe and L. Blackmore, "Lossless convexification of a class of optimal control problems with non-convex control constraints," *Automatica*, vol. 47, no. 2, pp. 341–347, 2011.

[5] M. W. Harris, "Optimal Control on Disconnected Sets Using Extreme Point Relaxations and Normality Approximations," *IEEE Trans. on Autom. Control*, vol. 66, pp. 6063–6070, 2021.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer New York, 2006.

[7] 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 Syst. Mag.*, vol. 42, pp. 40–113, 2022.

[8] R. Bonalli, T. Lew, and M. Pavone, "Analysis of Theoretical and Numerical Properties of Sequential Convex Programming for Continuous-Time Optimal Control," *IEEE Trans. on Autom. Control*, vol. 68, pp. 4570–4585, 2022.

[9] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "GuSTO: Guaranteed Sequential Trajectory optimization via Sequential Convex Programming," in *Int. Conf. on Robot. and Automat.*, pp. 6741–6747, IEEE, 2019.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England: Cambridge University Press, 2004.

[11] A. L. Yuille and A. Rangarajan, "The Concave-Convex Procedure (CCCP)," in *Adv. in Neural Inf. Process. Syst.*, 2001.

[12] P. T. Boggs and J. W. Tolle, "A Strategy for Global Convergence in a Sequential Quadratic Programming Algorithm," *SIAM J. Numer. Anal.*, vol. 26, pp. 600–623, 1989.

[13] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM J. on Optim.*, vol. 12, pp. 979–1006, 2002.

[14] Y. Mao and B. Acikmese, "SCvx-fast: A Superlinearly Convergent Algorithm for A Class of Non-Convex Optimal Control Problems," *arXiv preprint*, 2021.

[15] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe, "Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints," *J. of Guid., Control, and Dyn.*, vol. 43, pp. 1399–1413, 2020.

[16] R. Bonalli, A. Bylard, A. Cauligi, T. Lew, and M. Pavone, "Trajectory Optimization on Manifolds: A Theoretically-Guaranteed Embedded Sequential Convex Programming Approach," in *Robot.: Sci. and Syst.*, 2019.

[17] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Elsevier, 1982.

[18] M. R. Hestenes, "Multiplier and gradient methods," *J. of Optim. Theory and Appl.*, vol. 4, pp. 303–320, 1969.

[19] M. J. D. Powell, "Algorithms for nonlinear constraints that use lagrangian functions," *Math. Program.*, vol. 14, pp. 224–248, 1978.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. and Trends in Mach. Learn.*, vol. 3, pp. 1–122, 2011.

[21] K. Oguri and G. Lantoine, "Stochastic Sequential Convex Programming for Robust Low-thrust Trajectory Design under Uncertainty," in *AAS/AIAA Astro. Special. Conf.*, 2022.

[22] D. P. Bertsekas, "On Penalty and Multiplier Methods for Constrained Minimization," *SIAM J. Control Optim.*, vol. 14, pp. 216–235,, 1976.

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

[24] Mosek ApS, "The MOSEK Optimization Toolbox for Matlab Manual, version 8.1.." http://docs.mosek.com/9.0/toolbox/index.html, 2017.

[25] T. P. Reynolds and M. Mesbahi, "The Crawling Phenomenon in Sequential Convex Programming," in *Amer. Control Conf.*, pp. 3613–3618, IEEE, 2020.