Astrodynamics Vol. 1, No. 1, 23–40, 2017 https://doi.org/10.1007/s42064-017-0003-8

Survey of convex optimization for aerospace applications

Xinfu Liu[1] (✉), **Ping Lu**[2], and **Binfeng Pan**[3]

1. *Beijing Institute of Technology, Beijing 100081, China*
2. *San Diego State University, San Diego, California 92182-1308, USA*
3. *Northwestern Polytechnical University, Xi'an 710072, China*

ABSTRACT

Convex optimization is a class of mathematical programming problems with polynomial complexity for which state-of-the-art, highly efficient numerical algorithms with predeterminable computational bounds exist. Computational efficiency and tractability in aerospace engineering, especially in guidance, navigation, and control (GN&C), are of paramount importance. With theoretical guarantees on solutions and computational efficiency, convex optimization lends itself as a very appealing tool. Coinciding the strong drive toward autonomous operations of aerospace vehicles, convex optimization has seen rapidly increasing utility in solving aerospace GN&C problems with the potential for onboard real-time applications. This paper attempts to provide an overview on the problems to date in aerospace guidance, path planning, and control where convex optimization has been applied. Various convexification techniques are reviewed that have been used to convexify the originally nonconvex aerospace problems. Discussions on how to ensure the validity of the convexification process are provided. Some related implementation issues will be introduced as well.

KEYWORDS

convex optimization
optimal control
convexification
convex relaxation

Review Article

Received: 19 December 2016
Accepted: 28 February 2017

1 Introduction

Many aerospace engineering problems require solving an optimal control problem with an optimization objective and various constraints on the system state and controls. The optimization objective could be, for instance, minimizing the fuel consumption in rocket launch, spacecraft rendezvous, or planetary landing. Constraints may include the equations of motion, the limits on thrust and aerodynamic forces, and necessary safety considerations. Traditionally, two kinds of methods can be used to solve the problem, the indirect and direct methods. The former applies the optimal control theory to derive the necessary conditions and then solve the resulting two-point boundary value problem (TPBVP). The latter discretizes the original continuous-time problem into a nonlinear programming (NLP) problem, which is then solved by an NLP algorithm. For nonlinear problems with many constraints, the necessary conditions are generally very complicated, and the TPBVP is known to be highly sensitive to initial guesses of the solution. Hence, the direct method is much more often used in practice.

Though the direct method has been widely used in the literature [1], general NLP problems are non-deterministic polynomial-time hard (NP-hard), which means that the amount of computation required to solve the problem will not be limited by a bound determinable *a priori*. In practical terms, the computation time may be very long if the problem is solved at all. Unknowable computational time and lack of assured algorithm convergence are the kind of obstacles that would preclude its applications in the aerospace engineering problems that demand reliable and rapid solutions. Note that for closed-loop guidance and control, the optimization problem must be solved in real time, which is generally hard to be achieved using the direct method to solve a general optimization problem. Nevertheless, convex optimization problems are computationally tractable in the sense that they can be solved by polynomial-time algorithms [2, 3]. Hence,

✉ lau.xinfu@gmail.com

in recent years many researchers have made significant efforts to solve aerospace engineering problems by convex optimization. This is done by recognizing and formulating the problem as a convex optimization problem whenever possible, or applying various convexification techniques to convert the original nonconvex formulation into a convex optimization problem.

Over the last 15 years a wide range of cases of aerospace applications of convex optimization have been investigated. Techniques have been developed along the way to enable and expand the classes of problems that can be solved by convex optimization. The area has attracted a growing number of researchers. It is against this backdrop that we believe a survey on the state of convex optimization applications in aerospace engineering would be timely. This paper provides such a summative assessment.

In the remaining of this paper, we will first briefly define several common classes of convex optimization problems. Then, we give a literature review on the aerospace engineering problems in different vehicle platforms and mission scenarios that have been solved by convex optimization to date. It should be pointed out that the review focuses on the trajectory planning/optimization and guidance problems. Since most realistic problems do not lend themselves readily as convex optimization problems, the methods developed in the literature to transform the problem into a convex one are described. Then the question on whether the convexified problem has the same solution with the original problem is discussed, as well as convergence issues on a successive solution procedure which is a popular and powerful way to solve nonconvex problems. Finally, certain implementation aspects are addressed.

2 Convex optimization

This section gives a brief introduction on convex optimization, which refers to a special class of optimization where the objective function is convex, the equality constraints are linear, and the inequality constraints define a convex admissible set. First, mathematical formulations of three types of convex optimization and their relationships are provided which are commonly used in the literature to solve aerospace engineering problems.

(1) Linear Programming (LP).

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geqslant 0 \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the decision variable vector and the problem parameters include $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$, where $m \leqslant n$. The last equation means that every component of the vector $\boldsymbol{x}$ is non-negative.

(2) Second-Order Cone Programming (SOCP):

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & ||\boldsymbol{F}_i\boldsymbol{x} + \boldsymbol{d}_i|| \leqslant \boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{x} + q_i, \quad i = 1, \ldots, l \end{aligned} \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the decision variable and the problem parameters include $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{F}_i \in \mathbb{R}^{n_i\times n}$, $\boldsymbol{d}_i \in \mathbb{R}^{n_i}$, $\boldsymbol{p}_i \in \mathbb{R}^n$, and $q_i \in \mathbb{R}$. Note that a convex quadratic programming problem can transformed into SOCP readily [3].

(3) Semidefinite Programming (SDP):

$$\begin{aligned} \min \quad & \mathrm{trace}(\boldsymbol{C}\boldsymbol{X}) \\ \text{s.t.} \quad & \mathrm{trace}(\boldsymbol{A}_i\boldsymbol{X}) = b_i, \quad i = 1, \ldots, m \\ & \boldsymbol{X} \succeq 0 \end{aligned} \tag{3}$$

where $\boldsymbol{X} \in \boldsymbol{S}^n$ ($\boldsymbol{S}^n$ denotes the set of symmetric $n \times n$ matrices) is comprised of the decision variables and the problem parameters include $\boldsymbol{C}, \boldsymbol{A}_i \in \boldsymbol{S}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$. The constraint $\boldsymbol{X} \succeq 0$ means that $\boldsymbol{X}$ is positive semidefinite.

The preceding problems are also called conic programming since the the admissible sets formed by the inequality constraints in LP, SOCP, and SDP correspond to linear cone (or nonnegative orthant), second-order cone, and positive semidefinite cone, respectively. Their relationships are as follows:

$$\mathrm{LP} \subseteq \mathrm{SOCP} \subseteq \mathrm{SDP}$$

which implies that SDP has the most modeling power among them. It should be pointed out that LP may be insufficient or inaccurate to model some complex problems, and for SDP existing algorithms it still does not scale well to the problem size, meaning that the solution speed deteriorates rapidly as the problem size increases. In contrast, SOCP makes a good balance between modeling power and computational efficiency. It includes second-order cone constraints to represent or approximate complex constraints and existing interior point methods can solve it very efficiently even with relatively a large problem size.

To apply convex optimization to solve a complex aerospace engineering problem, the original optimal control problem should be converted into a form that can be discretized as a convex optimization problem, and then a single or a sequence of convex optimization problems are solved to approach the solution of the original problem. This generally poses challenges in a problem with non-convexity. For example, the equality constraints in LP, SOCP, and SDP are all linear, and thus this requires that the dynamics must be linear so that they become linear algebraic equations after discretization.

3 A literature review

As opposed to an exhaustive account on the literature, the main purpose of this paper is to offer a window into the exciting representative research works that demonstrate the power of convex optimization in efficiently solving various complex aerospace problems. At the same time, it is our hope that this paper will stimulate more interests in developing more methods and applications to further expand the research in the area.

3.1 Applications in low-speed UAVs

Low-speed UAVs such as quadcopters, where the aerodynamic forces may be ignored, have been an active research area in recent years due to its vast potentials in both commercial and military applications. The ability of fast computation on planned trajectories or controls is very important for such agile vehicles to perform various missions. Convex optimization has played a significant role in achieving this goal. In Ref. [4], the problem of designing state interception trajectories for a quadcopter is formulated as a convex optimization problem in each decoupled axis. This can make the flight trajectories be generated in real time. When multiple UAVs need to travel from an initial state to a final state without collision, the corresponding trajectory planning problem is nonconvex. Nevertheless, this problem can be efficiently solved by sequential convex programming (SCP) after the nonconvex constraints are approximated by convex ones [5]. To reduce the probability of infeasibility resulting from convex approximations of the collision-avoidance constraints, an incremental SCP method was proposed to tighten the constraints incrementally [6]. For multiple UAVs to travel in formation in environments with static or dynamic obstacles, a centralized algorithm based on SCP was proposed in Ref. [7] to compute an optimal formation, and a distributed version was also based on SCP [8]. In Ref. [9], both centralized and decentralized algorithms based on convex optimization were investigated for collision avoidance of multiple aerial vehicles.

3.2 Applications in spacecraft

In the literature, convex optimization has been widely used in spacecraft. Typical examples include the promising application of convex optimization in the Mars precise landing problems with nonconvex constraints on the thrust magnitude [10–15]. Convex optimization has also found applications in rapidly designing descent trajectories for asteroid landing [16, 17]. Some general results of lossless convexification on problems with certain type of nonconvex control constraints were given in Refs. [18, 19]. Spacecraft rendezvous and proximity operations with the target spacecraft in any Keplerian orbit, which corresponds to a highly nonlinear optimal control problem, were solved by successive second-order cone programming (SOCP) [20–23]. The application of convex optimization in the spacecraft rendezvous guidance problem can also be found in Refs. [24, 25]. In spacecraft trajectory planning, collision-avoidance constraints, station-keeping constraints, and navigation uncertainty can all be transformed into convex constraints [26, 27]. Formation reconfiguration with collision-avoidance constraints [28] and the formation control problem of multiple spacecraft [29] were all solved by convex optimization for onboard implementation. For impulsive maneuver-based proximity operations with linear continuous constraints on the spacecraft relative trajectory, semidefinite programming is employed to get tractable solutions [30]. Semidefinite programming was also introduced to solve the path planning problem of spatial rigid motion with attitude constraints [31] and the formation initialization control problem in formation flying [32]. In Ref. [33], de Bruijn *et al.* proposed a novel affine formulation of the dynamics of geostationary satellites and formulated the geostationary satellite station-keeping problem as a linear programming problem. Later, de Bruijn *et al.* [34] extended the approach and solved the station-keeping problem for a fleet of geostationary satellites as a convex optimization problem. For swarms of spacecraft with hundreds to thousands of agents, the optimal guidance and reconfiguration problem was solved using sequential convex programming [35].

3.3 Applications in high-speed atmospheric vehicles

High-speed atmospheric vehicles such as hypersonic glide vehicles are subject to significant aerodynamic forces to control their motion in space. In general, the dynamics for these vehicles are highly nonlinear and there exist various strict (nonconvex) path constraints for safety considerations or physical limits. Hence, it is very challenging to apply convex optimization to solve the model-based trajectory planning/optimization problems related to these vehicles. For reentry vehicles the entry dynamics with bank angle as the control input are highly nonlinear. Liu *et al.* [36] proposed redefining the controls and finally obtained the corresponding linear dynamics plus additional nonconvex control constraints which were then relaxed into convex constraints. The original complex entry trajectory optimization problem was successfully converted into an SOCP problem and successive SOCP was introduced to approach the solution of the original problem. Later, convex optimization was also applied to solve the (nonconvex) maximum-crossrange problem in entry flight [37]. In addition, for hypersonic glide vehicles with high lift-to-drag ratios, convex optimization was used to efficiently generate very smooth glide trajectories [38]. In these applications, only the bank angle is considered as the control input.

For systems with both the angle of attack and bank angle as control inputs, the dynamics are more nonlinear, which makes application of convex optimization to such systems even more difficult. In Ref. [39], the optimal flight of aerodynamically controlled vehicles in the terminal phase was considered, where both the angle of attack and bank angle were control inputs and a nonlinear dynamic pressure constraint was also enforced. This highly nonlinear and nonconvex optimization problem was solved by convex optimization after a constructive method was used to handle the nonlinear dynamics. To further broaden the application of convex optimization, Liu [40] extended convex optimization to the fuel-optimal rocket landing problem in which the engine thrust adds another freedom of controls in addition to the aerodynamic forces. Though the simultaneous presence of thrust and aerodynamics forces is still a challenge to be dealt with by convex optimization, it will find broad and important applications in atmospheric vehicles with propulsion. Szmuk *et al.* [41] also considered the rocket landing problem where the aerodynamic drag is modeled as a quadratic function of the velocity, and this nonconvex problem was shown to be efficiently solved by convex optimization.

From the preceding three categories, it is seen that convex optimization has been widely applied to solve various problems in different research areas. Besides, mixed-integer convex optimization has also found many applications. Though it is nonconvex due to the existence of integer variables, it can model problems with logical constraints and can be efficiently solved by available software such as CPLEX as long as the number of integer variables is small. In Ref. [42], mixed-integer convex optimization was used to solve the spacecraft reorientation problem with pointing constraints. Richards *et al.* [43] proposed using mixed-integer linear programming (MILP) to solve the spacecraft trajectory planning problem with collision-avoidance constraints. MILP was also used for the formation flying control of multiple spacecraft [44] and collision-free trajectory planning of UAVs [45–48].

With the success of convex optimization in many aerospace applications, in the next section we will discuss in detail on how to convexify a nonconvex problem to fit the framework of convex optimization.

4 Non-convexity and convexification techniques

In this section, we first introduce common non-convexity seen in aerospace engineering problems, and then present various convexification techniques to handle the non-convexity. In addition, we will especially introduce how the convexification techniques can be used to convexify nonlinear dynamics and concave state inequality constraints.

4.1 Non-convexity

Suppose that the optimal control problem we need to solve has the following general form.

Problem $\mathcal{O}$:

$$\min \quad \varphi(\boldsymbol{x}(t_{\mathrm{f}}), t_{\mathrm{f}}) + \int_{t_0}^{t_{\mathrm{f}}} \ell(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\, \mathrm{d}t \tag{4}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t),\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{5}$$

$$\boldsymbol{s}_1(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leqslant 0 \tag{6}$$

$$\boldsymbol{s}_2(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = 0 \tag{7}$$

$$\boldsymbol{\psi}_1(\boldsymbol{x}(t_{\mathrm{f}}), t_{\mathrm{f}}) \leqslant 0,\ \boldsymbol{\psi}_2(\boldsymbol{x}(t_{\mathrm{f}}), t_{\mathrm{f}}) = 0 \tag{8}$$

where t represents the independent variable (not necessarily the time) with its initial value t_0 and terminal value t_{f} given, $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, and $\boldsymbol{u} \in \mathbb{R}^m$ is the control vector. The functions $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ in Eqs. (6) and (7) can be functions of either $\boldsymbol{x}$ or $\boldsymbol{u}$, or of both variables. Equation (8) contains terminal inequality and equality constraints.

Problem $\mathcal{O}$ is nonconvex optimal control problem if its discretized version is not a convex optimization problem. Non-convexity of the problem can be from the objective function and/or the constraints. The objective function in Eq. (4) will not be linear after discretization as long as φ or ℓ is nonlinear. The dynamics in Eq. (5) are nonlinear in many aerospace applications, such as in the hypersonic gliding and rocket landing problems [36, 40]. The nonlinear dynamics will become nonlinear algebraic equality constraints after discretization and they may be the major sources of non-convexity. The inequality path constraints in Eq. (6) are nonconvex if the feasible set defined by them is not convex. For LP, SOC, and SDP, the feasible set must be representable by the intersection of linear cones, second-order cones, and semidefinite cones. Examples of nonconvex path constraints include collision/obstacle avoidance, heating rate constraint in entry flight, nonzero lower bound on the thrust magnitude of a rocket, etc. The equality path constraints in Eq. (7) are nonconvex except $\boldsymbol{s}_2$ are linear functions. Finally, terminal constraints in Eq. (8) are nonconvex if the feasible set of $\boldsymbol{\psi}_1 \leqslant 0$ can not be represented by cones or $\boldsymbol{\psi}_2$ are nonlinear functions.

Remark 1: In Problem $\mathcal{O}$, selecting an appropriate independent variable is important. When the total time t_{f} is given, it is natural to choose time as the independent variable. Nevertheless, when t_{f} is free, we may still choose time as the independent variable, but need to use an outer loop to search for the optimal time. Alternatively, other monotonically changing variables with given intervals can act as the independent variable. This will remove the outer loop for finding the optimal time, and at the same time may be very helpful for easily convexifying certain nonconvex constraints (see the convexification of the nonconvex dynamic constraint in Ref. [39]).

4.2 Convexification techniques

To apply convex optimization to solve the original nonconvex Problem $\mathcal{O}$, appropriate convexification techniques are required to convexify the problem so that it can finally be discretized as a convex optimization problem. This subsection aims to introduce the convexification techniques commonly used in the literature.

4.2.1 Equivalent transformation

For certain non-convexity in Problem $\mathcal{O}$, equivalent transformation acts as an effective convexification technique since it does not involve any approximation. This technique is widely used in convexifying objective functions that cannot be discretized as linear functions. For instance, it is straightforward to equivalently transform:

$$\min \int_{t_0}^{t_{\mathrm{f}}} \|\boldsymbol{u}\|\, \mathrm{d}t \tag{9}$$

into

$$\min \int_{t_0}^{t_{\mathrm{f}}} \eta\, \mathrm{d}t \tag{10}$$

$$\text{s.t.} \quad \|\boldsymbol{u}\| \leqslant \eta \tag{11}$$

where the objective function can be discretized as a linear function. This transformation generates a new constraint in Eq. (11). For effective transformation, we require the constraint to fit the framework of convex optimization. Obviously, Eq. (11) is a second-order cone constraint, which is convex.

When the objective function is complicated and highly nonlinear, it may not be straightforward to make an equivalent transformation as in the previous example. Then, we can consider constructing a new problem with a linear objective function, while the only requirement is that it has an equivalent optimization objective with the original problem. A good example is from the maximum-crossrange problem in entry flight in which the optimization objective is to maximize the crossrange when the downrange is given for a hypersonic glide vehicle [37]. This problem has the following optimization objective:

$$\min \quad \sin\phi_{\mathrm{f}}\sin\phi_{\mathrm{P}} + \cos\phi_{\mathrm{f}}\cos\phi_{\mathrm{P}}\cos(\theta_{\mathrm{f}} - \theta_{\mathrm{P}}) \tag{12}$$

where θ_{P} and ϕ_{P} are given parameters, and θ_{f} and ϕ_{f} are decision variables of the final longitude and latitude, respectively. The objective function in Eq. (12) is highly nonlinear. Fortunately, the optimization objective has been proved to be equivalent with the following one [37]:

$$\min \quad \theta_{\mathrm{f}} \text{ or } \phi_{\mathrm{f}} \text{ or } -\phi_{\mathrm{f}} \tag{13}$$

where the objective function is linear and which objective function to choose depends on a specific mission (see Proposition 2 in Ref. [37]).

Equivalent transformation can also be used to convexify highly nonlinear path constraints. In entry flight, the heating rate constraint for a hypersonic glide vehicle is as follows:

$$k_Q\sqrt{\rho}V^{3.15} \leqslant \dot{Q}_{\max} \tag{14}$$

where k_Q is a given constant, ρ is the air density (which is a function of the radial distance r of the vehicle), and V is the speed of the vehicle. Equation (14) is a highly nonlinear constraint on the altitude and velocity of the vehicle. Nevertheless, when energy is selected as the independent variable, it can be equivalently transformed into the following linear constraint on r [38]:

$$r(e) \geqslant l(e) \tag{15}$$

where the value of l is numerically obtained from Eq. (14) at the discretized points.

4.2.2 Change of variables

Change of variables is generally used to replace nonlinear terms by linear terms. This is a standard and popular technique for convexification. Nonlinear terms frequently appear in dynamics of flight vehicles. For example, the dynamics of a spacecraft contain a nonlinear term $\boldsymbol{T}/m$, where $\boldsymbol{T}$ is the thrust vector and m is the mass. This term can be defined as a new variable [10], i.e.

$$\boldsymbol{u} := \boldsymbol{T}/m \tag{16}$$

With the preceding change of variables, the dynamics will be linear with respect to $\boldsymbol{u}$. Another representative example is the nonlinear terms $\cos\sigma$ and $\sin\sigma$, where σ is the bank angle, present in the dynamics of a hypersonic glide vehicle. Performing the following change of variables [36]:

$$u_1 := \cos\sigma,\ u_2 := \sin\sigma \tag{17}$$

will make the dynamics linear with respect to the new variables u_1 and u_2.

4.2.3 Successive linearization

Successive linearization refers to the process of repeatedly linearizing a nonlinear term at a known solution obtained at the previous iteration. It is a popular and simple technique used to convert nonlinear terms into linear terms. Specifically, a nonlinear term $\boldsymbol{f}(\boldsymbol{z})$ can be approximated by

$$\boldsymbol{f}(\boldsymbol{z}) \approx \boldsymbol{f}(\boldsymbol{z}^{(k)}) + \boldsymbol{f}_{\boldsymbol{z}}(\boldsymbol{z}^{(k)})(\boldsymbol{z} - \boldsymbol{z}^{(k)}) \tag{18}$$

where $\boldsymbol{z}^{(k)}$ is a known solution from the kth iteration, and $\boldsymbol{f}_{\boldsymbol{z}}(\boldsymbol{z}^{(k)})$ is the derivative of $\boldsymbol{f}(\boldsymbol{z})$ with respect to $\boldsymbol{z}$ at $\boldsymbol{z}^{(k)}$.

4.2.4 Successive approximation

Successive approximation is an alternative to convert nonlinear terms into linear ones. This is a technique that is different from successive linearization. If the nonlinear term $\boldsymbol{f}(\boldsymbol{z})$ can be rewritten in the following form:

$$\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{g}(\boldsymbol{z})\boldsymbol{z} \tag{19}$$

then successive approximation takes the following approximation:

$$\boldsymbol{f}(\boldsymbol{z}) \approx \boldsymbol{g}(\boldsymbol{z}^{(k)})\boldsymbol{z} \tag{20}$$

where $\boldsymbol{g}(\boldsymbol{z}^{(k)})$, the value of $\boldsymbol{g}$ at the kth iteration, approximates $\boldsymbol{g}(\boldsymbol{z})$ in Eq. (19). Note that the original term $\boldsymbol{f}(\boldsymbol{z})$ (or equivalently $\boldsymbol{g}(\boldsymbol{z})\boldsymbol{z}$) is nonlinear with respect to $\boldsymbol{z}$, whereas its approximation $\boldsymbol{g}(\boldsymbol{z}^{(k)})\boldsymbol{z}$ is linear with respect to $\boldsymbol{z}$.

4.2.5 Relaxation

For convenience of discussion, we first introduce several concepts used in this paper. *Relaxation* refers to one of the many convexification techniques, that is to relax a nonconvex constraint from an original problem into a convex constraint. When this technique is used and the resulting relaxed problem becomes convex, we can call the new problem a *convex relaxation* of the original nonconvex problem. The convex relaxation can specifically mean *LP relaxation*, *SOCP relaxation*, or *SDP relaxation*, depending on what type of convex optimization the relaxed problem belongs to. In general, the convex relaxation has a larger feasible set than the original problem, which implies that the convex relaxation generally has a smaller cost for a minimization problem. When they have the same cost with the same solution, we say that the convex relaxation is *exact*, or call the situation as *exact convex relaxation* (which will be introduced later in Section 5.1).

Now let us get back to the relaxation technique. There are roughly three kinds of methods to relax a nonconvex constraint, which all involve expanding the original nonconvex feasible set to a convex one.

The first method considers getting the convex hull of the feasible set defined by the nonconvex constraint. Note that the convex hull of any nonconvex set is convex. A simple example is to relax the following nonconvex binary constraint:

$$x = \{0, 1\} \tag{21}$$

into a continuous constraint:

$$0 \leqslant x \leqslant 1 \tag{22}$$

which forms a convex hull of the nonconvex set defined by Eq. (21).

There are also other examples in which the relaxation technique has played a significant role in convexifying nonconvex constraints. In entry flight of a hypersonic glide vehicle with bank angle σ as the control input, applying the change of variables in Eq. (17) will generate:

$$u_1^2 + u_2^2 = 1 \tag{23}$$

which has its nonconvex feasible set shown in Fig. 1(a). The convex hull of this set is represented by the following convex constraint [36]:

$$u_1^2 + u_2^2 \leqslant 1 \tag{24}$$

Relaxing Eq. (23) into Eq. (24) can be visually seen in Fig. 1.

In three-dimensional terminal guidance of aerodynamically controlled missiles, the control inputs include both the angle of attack α and bank angle σ. Performing the following change of variables:

$$\bar{u}_1 = \bar{\eta}\cos\sigma,\ \bar{u}_2 = \bar{\eta}\sin\sigma,\ \bar{u}_3 = \bar{\eta}^2 \tag{25}$$

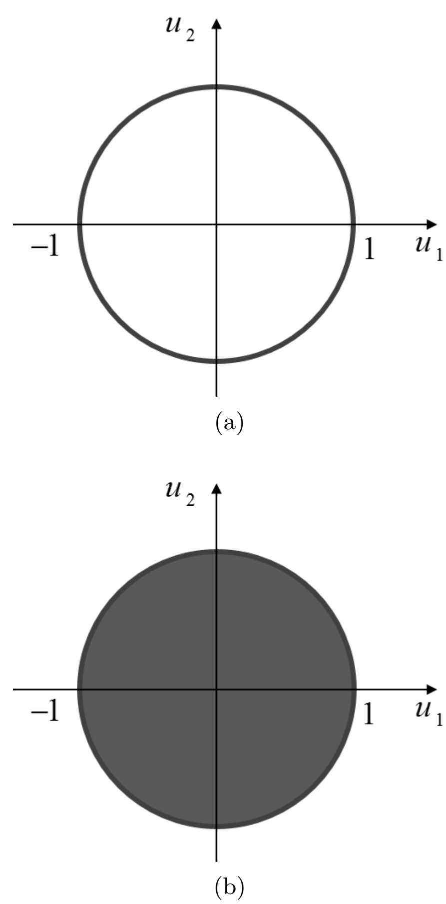


Fig. 1 Relaxing the nonconvex control constraint $u_1^2+u_2^2=1$ (its feasible set is the circle in (a)) into the convex control constraint $u_1^2+u_2^2 \leqslant 1$ (its feasible set is the circular plate in (b)) [36].

where $\bar{\eta}$ is a variable dependent on α, will yield [39]:

$$\bar{u}_1^2 + \bar{u}_2^2 = \bar{u}_3 \tag{26}$$

where $0 \leqslant \bar{u}_3 \leqslant \bar{\eta}_{\max}^2$.

The preceding nonconvex constraint can be relaxed as [39]:

$$\bar{u}_1^2 + \bar{u}_2^2 \leqslant \bar{u}_3 \tag{27}$$

which forms a convex set. The feasible sets of Eqs. (26) and (27) are visually shown in Fig. 2.

In vertical landing of a reusable rocket, the magnitude of the rocket thrust is T and its direction in a plane is denoted by an angle β. These two variables are the control inputs. By change of variables the following nonconvex control constraint is obtained [40]:

$$\tilde{u}_1^2 + \tilde{u}_2^2 = \tilde{u}_3^2 \tag{28}$$

where $\tilde{u}_1^2 = T\cos\beta/m$, $\tilde{u}_2^2 = T\sin\beta/m$, $\tilde{u}_1^3 = T/m$ with m being the rocket mass. In addition, $\tilde{u}_3$ satisfies:

$$0 \leqslant \tilde{u}_3 \leqslant T_{\max}/m \tag{29}$$

where $T_{\max}$ is the maximum available thrust magnitude. Note that a lower bound may also be imposed on the thrust magnitude [40].

The preceding constraints correspond to a nonconvex set (see Fig. 3(a)) and its convex hull is defined by

$$\tilde{u}_1^2 + \tilde{u}_2^2 \leqslant \tilde{u}_3^2, \quad 0 \leqslant \tilde{u}_3 \leqslant T_{\max}/m \tag{30}$$

which feasible set is visually shown in Fig. 3(b).

The second method considers relaxing a nonconvex

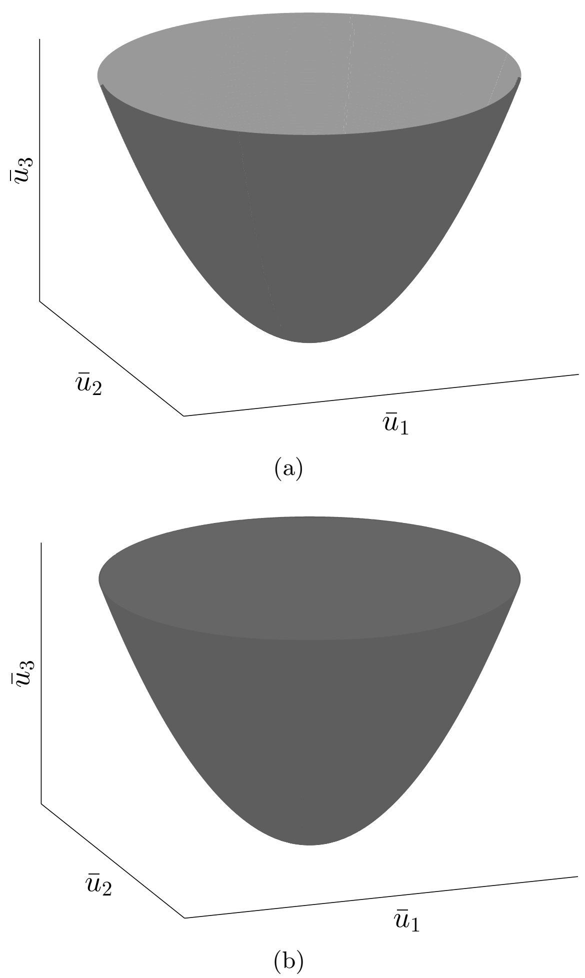


Fig. 2 Relaxing the nonconvex control constraint $\bar{u}_1^2 + \bar{u}_2^2 = \bar{u}_3$ (its feasible set is the curved surfaces in (a)) into the convex control constraint $\bar{u}_1^2 + \bar{u}_2^2 \leqslant \bar{u}_3$ (its feasible set is the whole solid in (b))[39]. Note that $0 \leqslant \bar{u}_3 \leqslant \bar{\eta}_{\max}^2$.

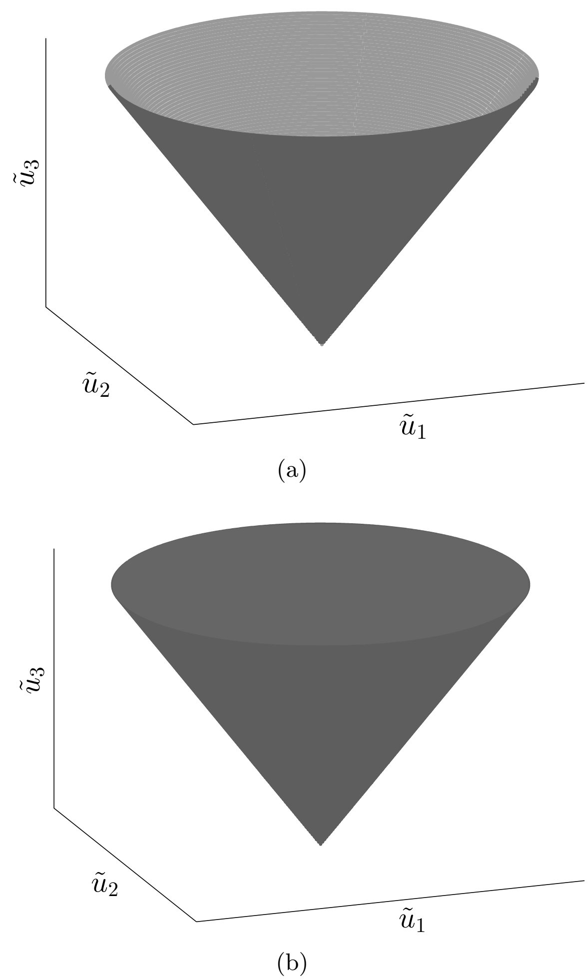


Fig. 3 Relaxing the nonconvex control constraint $\tilde{u}_1^2 + \tilde{u}_2^2 = \tilde{u}_3^2$, $0 \leqslant \tilde{u}_3 \leqslant T_{\max}/m$ (its feasible set is the curved surfaces in (a)) into the convex control constraint $\tilde{u}_1^2 + \tilde{u}_2^2 \leqslant \tilde{u}_3^2$, $0 \leqslant \tilde{u}_3 \leqslant T_{\max}/m$ (its feasible set is the whole solid in (b)) [40].

constraint with the help of introducing new slack variables. This implies that the dimension of the feasible set will be increased in the relaxation process. A typical example is from the nonconvex control constraint on thrust in the Mars pinpoint landing problem [10], i.e.

$$T_{\min} \leqslant \|\boldsymbol{T}\| \leqslant T_{\max} \tag{31}$$

where $T_{\min} > 0$. The constraint (31) is obviously nonconvex. Nevertheless, it can be relaxed into the following convex control constraints [10]:

$$\|\boldsymbol{T}\| \leqslant \varGamma \tag{32}$$

$$T_{\min} \leqslant \varGamma \leqslant T_{\max} \tag{33}$$

where $\varGamma$ is a slack variable. When $\boldsymbol{T}$ is in two-dimensional, the feasible sets corresponding to Eqs. (31), (32), and (33) are shown in Fig. 4. It is seen that the dimension of the feasible set is increased by one in the relaxation process.

In the preceding two methods, the relaxed constraint has its feasible set larger than that of the original nonconvex constraint and it is always convex. The third method will be slightly different. In this method, though the relaxed constraint has a larger feasible set, it may still not be convex. Additional techniques, such as an iterative process, are required to convert the nonconvex relaxed constraint into a convex constraint. For example, consider a constraint in the form:

$$\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \tag{34}$$

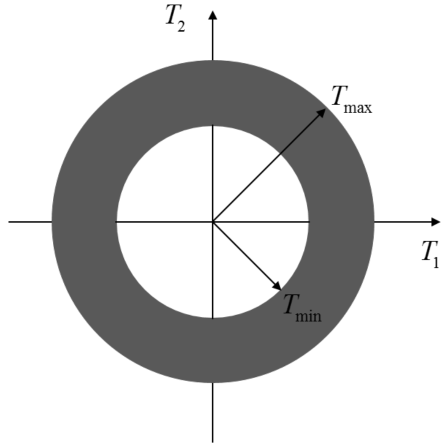
where $\boldsymbol{x} \in \mathbb{R}^n$. It is a rank-one constraint and is equivalent to

$$\boldsymbol{X} \succeq 0, \ \ \operatorname{rank}(\boldsymbol{X}) = 1 \tag{35}$$

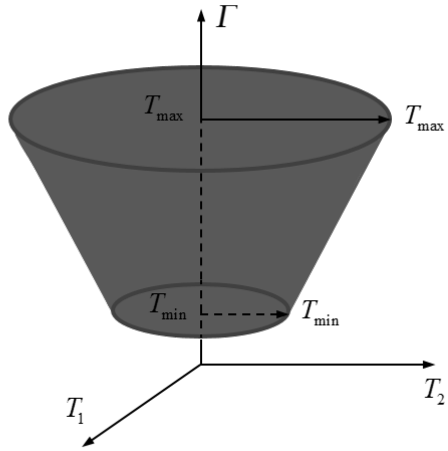
Equation (35) is a noncovex constraint. It can be relaxed as [31]:

$$\boldsymbol{X} \succeq 0, \ \ r\boldsymbol{I}_{n-1} - \boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{V} \succeq 0 \tag{36}$$

where $r \in \mathbb{R}$, $\boldsymbol{I}_{n-1} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the identity matrix, and $\boldsymbol{V} \in \mathbb{R}^{n\times(n-1)}$ is composed of the eigenvectors corresponding to the $(n-1)$ smallest eigenvalues of $\boldsymbol{X}$. The feasible set defined by Eq. (36) is larger than that of Eq. (35), and when $r = 0$ they have the same feasible sets.



(a)



(b)

Fig. 4 Relaxing the nonconvex control constraint $T_{\min} \leqslant \|\boldsymbol{T}\| \leqslant T_{\max}$ (its feasible set is the circular strip in (a)) into the convex control constraints $\|\boldsymbol{T}\| \leqslant \varGamma$ and $T_{\min} \leqslant \varGamma \leqslant T_{\max}$ (its feasible set is the whole solid in (b)) [10].

It is seen that Eq. (36) is still a nonconvex constraint since $\boldsymbol{V}$ is dependent on the unknown variable $\boldsymbol{X}$. Nevertheless, this can be resolved by using an iterative method in which $\boldsymbol{V}$ is obtained from $\boldsymbol{X}$ in the previous iteration [31].

Remark 2: All the methods introduced in this subsection are about relaxing nonconvex constraints. It should be pointed out that in some cases we may also need to relax convex constraints. Consider the situation that a convex constraint is an approximation to its corresponding original constraint. It is possible that an optimization problem including the approximated convex constraint becomes infeasible even when the optimization problem with the original constraint is feasible. To avoid the infeasibility, we can further relax the convex constraint to ensure that the problem is always feasible. For example, when a nonlinear constraint:

$$\boldsymbol{f}(\boldsymbol{x})=0 \tag{37}$$

is linearized to obtain the following convex constraint:

$$\boldsymbol{f}(\boldsymbol{x}^{(k)})-\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)})(\boldsymbol{x}-\boldsymbol{x}^{(k)})=0 \tag{38}$$

An optimization problem with the constraint in Eq. (38) may be infeasible at the intermediate iterations. To avoid this to happen, we can relax Eq. (38) into

$$\boldsymbol{f}(\boldsymbol{x}^{(k)})-\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)})(\boldsymbol{x}-\boldsymbol{x}^{(k)})=\boldsymbol{h}_{\mathrm{l}}-\boldsymbol{h}_{\mathrm{r}} \tag{39}$$

$$\boldsymbol{h}_{\mathrm{l}}\geqslant 0,\quad \boldsymbol{h}_{\mathrm{r}}\geqslant 0 \tag{40}$$

where $\boldsymbol{h}_{\mathrm{l}}$, $\boldsymbol{h}_{\mathrm{r}}\in\mathbb{R}^m$ have the same dimensions with $\boldsymbol{f}$. The term $\boldsymbol{h}_{\mathrm{l}}-\boldsymbol{h}_{\mathrm{r}}$ in Eq. (39) is called a *disturbance term* and can be either positive or negative. This strategy can effectively remove the infeasibility resulted from the approximation from Eq. (37) to Eq. (38). To make a connection between Eq. (39) and Eq. (38), the following optimization objective can be used:

$$\min \quad J_0+c_1\sum_{i=1}^{m}(\boldsymbol{h}_{\mathrm{l}})_i+c_2\sum_{i=1}^{m}(\boldsymbol{h}_{\mathrm{r}})_i \tag{41}$$

where c_1, $c_2\in\mathbb{R}$, J_0 is the original objective function, and the last two terms are related to the relaxation process. In general, we can choose large c_1 and c_2 to put high penalty on nonzero $\boldsymbol{h}_{\mathrm{l}}$ and $\boldsymbol{h}_{\mathrm{r}}$. Note that when $\boldsymbol{h}_{\mathrm{l}}=\boldsymbol{h}_{\mathrm{r}}=0$, Eq. (39) is the same with Eq. (38). This relaxation method has been successfully used in the maximum-crossrange problem in hypersonic entry flight [37].

4.3 Convexification of nonlinear dynamics

One major work in convexification of complex aerospace engineering problems lies on how to convexify the nonlinear dynamics. The convexification techniques introduced in Section 4.2 may be combined to handle the nonlinear dynamics with the following general form:

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t) \tag{42}$$

There exist a few methods to convexify Eq. (42). A popular method is to simply use successive linearization to get:

$$\begin{aligned}\dot{\boldsymbol{x}}=&\boldsymbol{f}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)+\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)(\boldsymbol{x}-\boldsymbol{x}^{(k)})\\&+\boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)(\boldsymbol{u}-\boldsymbol{u}^{(k)})\\=&A(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\boldsymbol{x}+B(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\boldsymbol{u}\\&+\boldsymbol{c}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\end{aligned} \tag{43}$$

where $\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)$ and $\boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)$ are the derivatives of $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)$ with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively, at $(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)})$, and

$$\begin{cases}A(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)=\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\\ B(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)=\boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\end{cases} \tag{44}$$

$$\begin{aligned}\boldsymbol{c}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)=&\boldsymbol{f}(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)-A(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\boldsymbol{x}^{(k)}\\&-B(\boldsymbol{x}^{(k)},\boldsymbol{u}^{(k)},t)\boldsymbol{u}^{(k)}\end{aligned} \tag{45}$$

Successive linearization was used in Refs. [49, 50] to convexify nonlinear dynamics.

In addition, we need the following trust-region constraints for validity of the linearization process:

$$|\boldsymbol{x}-\boldsymbol{x}^{(k)}|\leqslant\boldsymbol{\delta}_{\boldsymbol{x}},\quad|\boldsymbol{u}-\boldsymbol{u}^{(k)}|\leqslant\boldsymbol{\delta}_{\boldsymbol{u}} \tag{46}$$

where the inequality sign applies componentwise, and $\boldsymbol{\delta}_{\boldsymbol{x}}$ and $\boldsymbol{\delta}_{\boldsymbol{u}}$ are constant vectors with appropriate dimensions. The constraints in Eq. (46) are all linear. Note that one may alternatively express the trust-region constraints with the norm symbol "$\|\bullet\|$", which will be second-order cone constraints.

If the nonlinear dynamics in Eq. (42) can be rewritten as

$$\dot{\boldsymbol{x}}=A(\boldsymbol{x},t)\boldsymbol{x}+B(\boldsymbol{x},t)\boldsymbol{u} \tag{47}$$

the second method is to use successive approximation to convexify Eq. (47) into the following linear dynamics:

$$\dot{\boldsymbol{x}}=A(\boldsymbol{x}^{(k)},t)\boldsymbol{x}+B(\boldsymbol{x}^{(k)},t)\boldsymbol{u} \tag{48}$$

Such a method was also previously introduced in Ref. [51]. While any function $\boldsymbol{f}(\boldsymbol{x},t)$ may be expressed as $A(\boldsymbol{x},t)\boldsymbol{x}$ with a matching $A(\boldsymbol{x},t)$, a technical challenge is that $A(\boldsymbol{x},t)$ (as well as $B(\boldsymbol{x},t)$) will need to satisfy certain conditions for the iteration sequence $\{\boldsymbol{x}^{(k)}\}$ to converge. See Ref. [51] for such a discussion in a problem with a quadratic performance index but no terminal or other constraints. For more general problems with equality and inequality constraints, rigorous theoretical results are not yet available. However, at least for space flight problems, extensive numerical evidence exists that shows this successive approach is hilghly convergent [16,17,20,22]. Note that this successive solution approach does not rely on the standard linearization. When the solution converges, it is exactly the solution to the original problem, not an approximation as a linearized solution is.

In the preceding two methods, one uses successive linearization to convert Eq. (42) into Eq. (43), and the other uses successive approximation to convert Eq. (49) into Eq. (50). The third method that combines successive approximation with successive linearization will be introduced in the following. Specifically, if the nonlinear dynamics in Eq. (42) can be rewritten as (possibly by change of variables):

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t) + B(\boldsymbol{x}, t)\boldsymbol{u} \tag{49}$$

we can linearize $\boldsymbol{f}(\boldsymbol{x}, t)$ at $\boldsymbol{x}^{(k)}$ and approximate $B(\boldsymbol{x}, t)$ with $B(\boldsymbol{x}^{(k)}, t)$ to yield [39]:

$$\begin{aligned}\dot{\boldsymbol{x}} =& \boldsymbol{f}(\boldsymbol{x}^{(k)}, t) + \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)}, t)(\boldsymbol{x} - \boldsymbol{x}^{(k)}) + B(\boldsymbol{x}^{(k)}, t)\boldsymbol{u} \\ =& A(\boldsymbol{x}^{(k)}, t)\boldsymbol{x} + B(\boldsymbol{x}^{(k)}, t)\boldsymbol{u} + \boldsymbol{c}(\boldsymbol{x}^{(k)}, t)\end{aligned} \tag{50}$$

where $A(\boldsymbol{x}^{(k)}, t) = \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)}, t)$ and $\boldsymbol{c}(\boldsymbol{x}^{(k)}, t) = \boldsymbol{f}(\boldsymbol{x}^{(k)}, t) - \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}^{(k)}, t)\boldsymbol{x}^{(k)}$. Since linearization on $\boldsymbol{f}(\boldsymbol{x}, t)$ is used from Eq. (49) to Eq. (50), the following trust-region constraint needs to be imposed:

$$|\boldsymbol{x} - \boldsymbol{x}^{(k)}| \leqslant \boldsymbol{\delta}_{\boldsymbol{x}} \tag{51}$$

which are linear constraints.

Remark 3: The first method is the most standard way to convexify nonlinear dynamics. Nevertheless, the second and third methods have their unique features in that the coefficients of the linear dynamics (see Eqs. (48) and (50)) are independent of $\boldsymbol{u}^{(k)}$, that is to say, the intermediate controls at an iteration (which may have high-frequency jitters [36]) will not affect the dynamics in the next iteration. This is a property that is not owned by the first method, but is likely to be helpful for convergence of the successive solution procedure introduced in Section 5. In addition, compared to the first method, the second and third methods may generate additional constraints when the system dynamics are expressed in the form of Eq. (47) or (49). These additional constraints, such as $u_1^2 + u_2^2 = 1$ in Ref. [36], may be nonconvex, and necessary convexification techniques introduced previously are required to handle the non-convexity.

4.4 Convexification of concave state inequality constraints

Consider a concave inequality constraint in the following form:

$$s(\boldsymbol{x}, t) \leqslant 0 \tag{52}$$

where s is a concave function. This type of nonconvex constraint could be from a collision avoidance constraint which is present in many practical problems. For a circular collision avoidance region, Mueller *et al.* [27,52] chose to replace the constraint with a rotating line which is tangent to the avoidance region. Two parameters, the initial location of the tangent line and a constant rotating speed, need to be determined by trial and error. Infeasibility may occur if the parameters are not appropriately selected. In Ref. [22], Liu and Lu proposed to use successive linearization to convexify Eq. (52) into the following linear constraint:

$$s(\boldsymbol{x}^{(k)}, t) + s_{\boldsymbol{x}}(\boldsymbol{x}^{(k)}, t)(\boldsymbol{x} - \boldsymbol{x}^{(k)}) \leqslant 0 \tag{53}$$

This method is simple and generally applicable to any concave inequality constraint. It should be emphasized that theoretical understanding and assurance to this classical linearization approach has been established in Ref. [22], that is, convergence is theoretically guaranteed in a successive solution procedure in which a sequence of SOCP problems with the constraint (53) is solved.

5 Validity of convexification

With the convexification techniques introduced in the preceding section and a discretization process (let $\boldsymbol{z}$ denote the controls and states from the discretized nodes plus all other decision variables), the original nonconvex Problem $\mathcal{O}$ can finally be converted into a convex optimization problem whose parameters are dependent on $\boldsymbol{z}^{(k)}$. In general, approximation is used in the convexification, hence it necessitates a successive solution procedure of solving a sequence of convex optimization problems to approach the solution of the original problem. This procedure can be described as follows:

(1) Set $k = 0$, and select an appropriate value for $\boldsymbol{z}^{(0)}$. Initialize the parameters in the convex optimization problem using $\boldsymbol{z}^{(0)}$.

(2) At the $(k+1)$th iteration ($k \geqslant 0$), solve the convex optimization problem to get a solution denoted by $\boldsymbol{z}^{(k+1)}$.

(3) Check whether the following stopping criterion for convergence is satisfied:

$$\max_i |\boldsymbol{z}_i^{(k+1)} - \boldsymbol{z}_i^{(k)}| \leqslant \epsilon \tag{54}$$

where the subscript i is the ith element of the vector $\boldsymbol{z}$ and $\epsilon \in \mathbb{R}$ is a user-defined small tolerance. If the condition in Eq. (54) holds, go to Step (4); otherwise, use $\boldsymbol{z}^{(k+1)}$ to update the parameters in the convex optimization problem, set $k = k + 1$, and return to Step (2).

(4) The solution to Problem $\mathcal{O}$ is found to be $\boldsymbol{z}^{(k+1)}$. Stop.

It should be pointed out that if the relaxation technique is used in convexification of the original problem, the statement in Step (4) is based on the assumption that the convex relaxation is exact, i.e., the solution meeting the relaxed constraint is also feasible to its corresponding (original) nonconvex constraint. Nevertheless, exact convex relaxation may not be satisfied. Hence, validity of convexification should concern how to ensure exact convex relaxation. In addition, it also concerns how to achieve convergence of the successive solution procedure. All these will be discussed in the following two subsections.

5.1 Exact convex relaxation

Ensuring exact convex relaxation is of great importance for validity of the convexification process. In many cases, when a nonconvex constraint is relaxed into a convex constraint, exact convex relaxation can be theoretically proved, and optimal control theory turns out to be a significant tool for completing the proof [10, 18, 19, 36]. Nevertheless, in some cases proof of the exactness may be unachievable and numerical simulations indeed support that the convex relaxation is not exact, which means that the relaxation technique used is invalid. Note that this may be caused by existence of some path constraints. For example, in Ref. [39] exact convex relaxation is guaranteed only when the dynamic pressure constraint is inactive (or equivalently the constraint does not exist).

When the convex relaxation is not exact, it should be emphasized that exact convex relaxation may still be achieved by using certain techniques. In the literature, a regularization technique has been proposed to serve the purpose. It chooses to just add a small regularization term to the objective function in Eq. (4), i.e.

$$\min \quad \varphi(\boldsymbol{x}(t_\mathrm{f}), t_\mathrm{f}) + \int_{t_0}^{t_\mathrm{f}} \ell(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,\mathrm{d}t + c_\mathrm{r}\int_{t_0}^{t_\mathrm{f}} \nu(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,\mathrm{d}t \tag{55}$$

where the third term is the regularization term. Note that c_r is generally selected to make the value of the regularization term relatively small such that the original optimization objective is hardly affected. Though the modification is slight, its effects are miraculous in that the convex relaxation is now guaranteed to be exact [36,38,39]. In addition, another technique is used in the problem of multiple spacecraft rendezvous using drag plates which are either deployed or retracted. The discrete control set $\{-1, 0\}$ is relaxed to a continuous set $[-1, 0]$. A constructive procedure is proposed to replace non-bang-bang singular controls (which are values in the interior of $[-1, 0]$) with bang-bang controls so that the corresponding convex relaxation is exact [53].

If the convex relaxation is not exact and we can not find effective techniques to make it exact, this implies that relaxing the original nonconvex constraint is inappropriate, and instead, we may need to consider using other convexification techniques to deal with the original constraints.

5.2 Convergence of the successive solution procedure

In general, it is difficult to theoretically prove convergence of the successive solution procedure. The reason is that convergence may not be mathematically guaranteed in all cases or it is still too difficult to prove the convergence when a very complicated problem (highly nonlinear and nonconvex) is solved by convex optimization. Nevertheless, in some special cases convergence can be proved. For instance, when concave constraints are successively linearized within the framework of SOCP, the corresponding successive solution procedure is theoretically proved to converge [20]. Another representative example is that convergence is proved when an iterative rank minimizing approach is used to approximate the nonconvex rank-one constraint [31, 54].

Despite the difficulty of proving convergence, we can use proper techniques to improve the robustness of convergence of the successive solution procedure. The first and foremost technique is the selection of an appropriate method to convexify the nonlinear dynamics. Different convexification methods are previously introduced in Section 4–Section 4.3, and which one to choose depends on which one has better convergence property. If the first method (pure successive linearization) makes the successive solution procedure hard/slow to converge, the second (pure successive approximation) or the third method (combined successive linearization and successive approximation) should be considered. It should be pointed out that though all methods finally get linear dynamics, the first method removes all nonlinearities inherent in the original dynamics, whereas the second and third methods may generate additional constraints when change of variables is used. These constraints are generally nonlinear and can be considered as the nonlinearities preserved. This is usually helpful for improving the robustness of convergence or rate of convergence of the successive solution procedure.

Other techniques include a line search approach. Note that certain constraints in the convex optimization problem are just approximations to their original ones which can be expressed in the following general form:

$$\boldsymbol{\psi}(\boldsymbol{z}) = 0 \tag{56}$$

This means that at the intermediate iterations the solution of the convex optimization problem does not satisfy Eq. (56). We can define an ℓ_1 merit function to measure the constraint violation of a solution $\boldsymbol{z}^{(k+1)}$ as follows:

$$\phi_1(\boldsymbol{z}^{(k+1)}, \boldsymbol{\mu}) = \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\psi}(\boldsymbol{z}^{(k+1)}) \tag{57}$$

where $\boldsymbol{\mu}$ is a positive weighting vector. We hope the value of ϕ_1 keeps decreasing as k increases and finally the converged solution makes $\phi_1 = 0$ (no constraint violation). To achieve this, a line search can be applied, which considers using:

$$\hat{\boldsymbol{z}}^{(k+1)} = \boldsymbol{z}^{(k)} + \alpha \boldsymbol{p} \tag{58}$$

where $\boldsymbol{p} = \boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}$ is the search direction and α is the step size, to update the parameters in the convex optimization problem. Note that originally $\boldsymbol{z}^{(k+1)}$ is used for the updates in Step (3) of the successive solution procedure, and $\hat{\boldsymbol{z}}^{(k+1)} = \boldsymbol{z}^{(k+1)}$ only when $\alpha = 1$. In Eq. (58), α is initially set as 1 and then decreased successively with a contraction factor κ, i.e., $\alpha \leftarrow \kappa\alpha$, until sufficient reduction is obtained in the constraint violation, that is, the following condition is satisfied for some positive constant ϱ:

$$\phi_1(\boldsymbol{z}^{(k)} + \alpha \boldsymbol{p}; \boldsymbol{\mu}) \leqslant \phi_1(\boldsymbol{z}^{(k)}) + \varrho\, \alpha\, D(\phi_1(\boldsymbol{z}^{(k)}; \boldsymbol{\mu}); \boldsymbol{p}) \tag{59}$$

where $D(\phi_1; \boldsymbol{p})$ is the directional derivative of $\phi_1(\boldsymbol{z}^{(k)}; \boldsymbol{\mu})$ along the search direction $\boldsymbol{p}$ (see the definition in Ref. [55]). Such a line search approach worked well in enhancing the robustness of convergence of the successive solution procedure in the maximum-crossrange problem [37].

6 Implementation

In this section, we will discuss some implementation issues related to the applications of convex optimization.

6.1 Numerical algorithms

In the successive solution procedure introduced in Section 5, a convex optimization problem needs to be solved in each iteration by numerical algorithms. Such algorithms are included in both commercial software such as MOSEK [56] and CPLEX and free software such as SDPT3 [57], SeDuMi [58], and ECOS [59]. Different software may have different interfaces. To facilitate the process of programming, a few software are also available for modeling a problem in a simple and natural language, such as YALMIP [60], CVX [61], and CVXGEN [62].

To further improve the efficiency of the convex optimization algorithms for potential onboard and embedded applications, it is important to develop customized algorithms for faster computational speed, which have been gained much attention in recent years [59, 63, 64]. Specifically, customized algorithms for second-order cone programming problems were introduced and successfully used for onboard powered descent guidance in a practical vertical-takeoff and vertical-landing rocket [65, 66].

6.2 Closed-loop optimization

Closed-loop optimization means that the optimal control problem is solved repeatedly every certain amount of time (or in each cycle) with the current condition as the initial condition of the problem. If controls are obtained in each cycle, this is much like a numerical feedback control law since the control law involves numerically solving an optimal control problem. Note that the horizon of the problem in each cycle keeps shrinking as the final time is approaching.

It is desirable to implement closed-loop optimization since it is advantageous for a system to be robust to uncertainties and disturbances, and can thus provide superior performance. It can also make a system adaptable to different mission requirements, especially in emergency situations. To realize closed-loop optimization, the optimal control problem should be quite efficiently solved in each cycle. The interval of the cycle depends on how fast the problem can be solved. In the literature, convex optimization-based closed-loop optimization was successfully applied to design optimized proportional navigation gain [67]. To decrease the length of the cycle, customized convex optimization algorithms are highly desired. In addition, the solution obtained in the previous cycle can be used to initialize the parameters of the optimization problem in the current cycle, which, when the successive solution procedure introduced in Section 5 is used, is beneficial to reduce the number of iterations for convergence.

7 Conclusions

With strong theoretical properties on existence and uniqueness of solution and appealing computational advantages ensured by polynomial complexity, convex optimization has in recent years found increasing applications in aerospace engineering. This paper provides a survey on representative literature of aerospace applications of convex optimization over the last 15 years. Moreover, various convexification techniques used to convexify an original nonconvex problem into a convex optimization problem are reviewed. Especially, the relaxation technique and different methods on convexifying nonlinear dynamics are discussed in detail. To ensure validity of the convexification process, techniques are introduced to achieve exact convex relaxation and improve the robustness of convergence of the successive solution procedure.

An emerging trend in aerospace guidance and control is what is known as computational guidance and control (CG&C) [68]. In CG&C algorithms replace the traditional guidance and control laws. Intensive online computation is the hallmark in CG&C. Convex optimization-based methods represent a prominent part of CG&C. To be certain, challenges in applying convex optimization in aerospace guidance and control applications remain. Chief among the challenges is the non-convexity that often exists in a realistic problem. The relaxation techniques to convexify the problem that offers the same solution for the relaxed and original problem are still specific, and it may require certain threshold of expertise to devise an effective relaxation. For closed-loop online applications, the efficiency/speed of the solution is of critical importance, and a customized algorithm tailored to a specific problem may be necessary. We hope that this paper has provided an informative overview to this exciting and relatively new research area. In the final analysis, we will have accomplished our goal if this effort stimulates further research and bring about advances in applying convex optimization to aerospace problems.

Acknowledgements

The author at Beijing Institute of Technology gratefully acknowledges the support to this work by the National Natural Science Foundation of China (Grant No. 61603017).

References

[1] Betts, J. T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd end. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, **2010**: 247–255.

[2] Nesterov, Y., Nemirovski, A. *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, **1994**.

[3] Boyd, S., Vandenberghe, L. *Convex Optimization*. New York: Cambridge University Press, **2004**.

[4] Mueller, M. W., D'Andrea, R. A model predictive controller for quadrocopter state interception. In: Proceedings of the 2013 European Control Conference, **2013**: 1383–1389.

[5] Augugliaro, F., Schoellig, A. P., D'Andrea, R. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, **2012**: 1917–1922.

[6] Chen, Y., Cutler, M., How, J. P. Decoupled multiagent path planning via incremental sequential convex programming. In: Proceedings of the IEEE International Conference on Robotics and Automation, **2015**: 5954–5961.

[7] Alonso-Mora, J., Baker, S., Rus, D. Multi-robot navigation information via sequential convex programming. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, **2015**: 4634–4641.

[8] Alonso-Mora, J., Montijano, E., Schwager, M., Rus, D. Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus. In: Proceedings of the IEEE International Conference on Robotics and Automation, **2016**.

[9] Alonso-Mora, J., Naegeli, T., Siegwart, R., Beardsley, P. Collision avoidance for aerial vehicles in multi-agent scenarios. *Autonomous Robots*, **2015**, 39(1): 101–121.

[10] Acikmese, B., Ploen, S. R. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, **2007**, 30(5): 1353–1366.

[11] Acikmese, B., Scharf, D., Blackmore, L., Wolf, A. Enhancements on the convex programming based powered descent guidance algorithm for mars landing. In: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences, **2008**: AIAA 2008-6426.

[12] Blackmore, L., Acikmese, B., Scharf, D. P. Minimum landing error powered descent guidance for mars landing using convex optimization. *Journal of Guidance, Control, and Dynamics*, **2010**, 33(4): 1161–1171.

[13] Acikmese, B., Carson, J., Blackmore, L. Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Transactions on Control Systems Technology*, **2013**, 21(6): 2104–2113.

[14] Harris, M. W., Acikmese, B. Maximum divert for planetary landing using convex optimization. *Journal of Optimization Theory and Applications*, **2013**, 162(3): 975–995.

[15] Scharf, D. P., Ploen, S. R., Acikmese, B. A. Interpolation-enhanced powered descent guidance for onboard nominal, off-nominal, and multi-x scenarios. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, **2015**: AIAA 2015-0850.

[16] Pinson, R., Lu, P. Rapid generation of optimal asteroid powered descent trajectories. In: Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, **2015**: AAS 15-616.

[17] Pinson, R., Lu, P. Trajectory design employing convex optimization for landing on irregularly shaped asteroids. In: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, **2016**: AIAA 2016-5378.

[18] Acikmese, B., Blackmore, L. Lossless convexification for a class of optimal problems with nonconvex control constraints. *Automatica*, **2011**, 47(2): 341–347.

[19] Harris, M. W., Acikmese, B. Lossless convexification of non-convex optimal control problems for state constrained linear systems. *Automatica*, **2014**, 50(9): 2304–2311.
[20] Lu, P., Liu, X. Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *Journal of Guidance, Control, and Dynamics*, **2013**, 36(2): 375–389.
[21] Liu, X., Lu, P. Robust trajectory optimization for highly constrained rendezvous and proximity operations. In: Proceedings of the AIAA Guidance, Navigation, and Control (GNC) Conference, Guidance, Navigation, and Control and Co-located Conferences, **2013**: AIAA 2013-4720.
[22] Liu, X., Lu, P. Solving nonconvex optimal control problems by convex optimization. *Journal of Guidance, Control, and Dynamics*, **2014**, 37(3): 750–765.
[23] Liu, X. Autonomous trajectory planning by convex optimization. Ph.D. Thesis. Iowa State University, **2013**.
[24] Grzymisch, J., Fichter, W. Optimal rendezvous guidance with enhanced bearings-only observability. *Journal of Guidance, Control, and Dynamics*, **2015**, 38(6): 1131–1140.
[25] Louembet, C., Arzelier, D., Deaconu, G. Robust rendezvous planning under maneuver execution errors. *Journal of Guidance, Control, and Dynamics*, **2015**, 38(1): 76–93.
[26] Mueller, J. B. Onboard planning of collision avoidance maneuvers using robust optimization. In: Proceedings of the AIAA Infotech@Aerospace Conference, **2009**: AIAA 2009-2051.
[27] Mueller, J. B., Griesemer, P. R., Thomas, S. Avoidance maneuver planning incorporating station-keeping constraints and automatic relaxation. *Journal of Aerospace Information Systems*, **2013**, 10(6): 306–322.
[28] Carson, J. M., Acikmese, B. A model-predictive control technique with guaranteed resolvability and required thruster. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences, **2006**: AIAA 2006-6780.
[29] Tillerson, M., Inalhan, G., How, J. P. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust and Nonlinear Control*, **2002**, 12(2–3): 207–242.
[30] Deaconu, G., Louembet, C., Theron, A. Designing continuously constrained spacecraft relative trajectories for proximity operations. *Journal of Guidance, Control, and Dynamics*, **2015**, 38(7): 1208–1217.
[31] Dai, R., Sun, C. Path planning of spatial rigid motion with constrained attitude. *Journal of Guidance, Control, and Dynamics*, **2015**, 38(8): 1356–1365.
[32] Wu, Y., Cao, X., Xing, Y., Zheng, P., Zhang, S. Relative motion coupled control for formation flying spacecraft via convex optimization. *Aerospace Science and Technology*, **2010**, 14(6): 415–428.
[33] De Bruijn, F., Theil, S., Choukroun, D., Gill, E. Geostationary satellite station-keeping using convex optimization. *Journal of Guidance, Control, and Dynamics*, **2016**, 39(3): 605–616.
[34] De Bruijn, F., Theil, S., Choukroun, D., Gill, E. Collocation of geostationary satellites using convex optimization. *Journal of Guidance, Control, and Dynamics*, **2016**, 39(6): 1303–1313.
[35] Morgan, D., Chung, S. J., Hadaegh, F. Y. Model predictive control of swarms of spacecraft using sequential convex optimization. *Journal of Guidance, Control, and Dynamics*, **2014**, 37(6): 1725–1740.
[36] Liu, X., Shen, Z., Lu, P. Entry trajectory optimization by second-order cone programming. *Journal of Guidance, Control, and Dynamics*, **2016**, 39(2): 227–241.
[37] Liu, X., Shen, Z., Lu, P. Solving the maximum-crossrange problem via successive second-order cone programming with a line search. *Aerospace Science and Technology*, **2015**, 47: 10–20.
[38] Liu, X., Shen, Z. Rapid smooth entry trajectory planning for high lift/drag hypersonic glide vehicles. *Journal of Optimization Theory and Applications*, **2016**, 168(3): 917–943.
[39] Liu, X., Shen, Z., Lu, P. Exact convex relaxation for optimal flight of aerodynamically controlled missiles. *IEEE Transactions on Aerospace and Electronic Systems*, **2016**, 52(4): 1881–1892.
[40] Liu, X. Fuel-optimal rocket landing with aerodynamic controls. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, **2017**: AIAA 2017-1732.
[41] Szmuk, M., Acikmese, B., Berning, A. W. Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, **2016**: AIAA 2016-0378.
[42] Tam, M., Lightsey, E. G. Constrained spacecraft reorientation using mixed integer convex programming. *Acta Astronautica*, **2016**, 127: 31–40.
[43] Richards, A., Schouwenaars, T., How, J. P., Feron, E. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics*, **2002**, 25(4): 755–764.
[44] Tillerson, M. J. Coordination and control of multiple spacecraft using convex optimization techniques. Master Thesis. Massachusetts Institute of Technology, **2002**.

[45] Culligan, K., Valenti, M., Kuwata, Y., How, J. P. Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization. In: Proceedings of the 2007 American Control Conference, **2007**: 5322–5327.
[46] Kamal, W. A., Gu, D.-W., Postlethwaite, I. Real time trajectory planning for UAVs Using MILP. In: Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, **2005**: 3381–3386.
[47] Forsmo, E. J. Optimal path planning for unmanned aerial systems. Master Thesis. Norwegian University of Science and Technology, **2012**.
[48] Culligan, K. F. Online trajectory planning for UAVs using mixed integer linear programming. Master Thesis. Massachusetts Institute of Technology, **2006**.
[49] Mao, Y., Szmuk, M., Ackimese, B. Successive convexification of non-convex optimal control problems and its convergence properties. In: Proceedings of the 2016 IEEE 55th Control and Decision Conference, **2016**: 3636–3641.
[50] Wang, Z., Grant, M. J. Constrained trajectory optimization for planetary entry via sequential convex programming. In: Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA AVIATION Forum, **2016**: AIAA 2016-3241.
[51] Banks, S., Dinesh, K. Approximate optimal control and stability of nonlinear finite- and infinite-dimensional systems. *Annals of Operations Research*, **2000**, 98(1): 19–44.
[52] Mueller, J. B., Larsson, R. Collision avoidance maneuver planning with robust optimization. In: Proceedings of the 7th International ESA Conference on Guidance, Navigation and Control Systems, **2008**.
[53] Harris, M. W., Acikmese, B. Minimum time rendezvous of multiple spacecraft using differential drag. *Journal of Guidance, Control, and Dynamics*, **2014**, 37(2): 365–373.
[54] Sun, C., Dai, R., Lu, P. Solving polynomial optimal control problems via iterative convex optimization. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, **2016**: AIAA 2016-0371.
[55] Nocedal, J., Wright, S. J. *Numerical Optimization*, 2nd end. New York: Springer-Verlag, **2006**.
[56] Andersen, E. D., Roos, C., Terlaky, T. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, **2003**, 95(2): 249–277.
[57] Toh, K. C., Todd, M. J., Tutuncu, R. H. SDPT3—A Matlab software package for semidefinite programming, Version 1.3. *Optimization Methods and Software*, **1999**, 11(1–4): 545–581.
[58] Sturm, J. F. Using SeDuMi 1.02: A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, **1999**, 11(1–4): 625–653.
[59] Domahidi, A., Chu, E., Boyd, S. ECOS: An SOCP solver for embedded system. In: Proceedings of the European Control Conference, **2013**: 3071–3076.
[60] Lofberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB, In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, **2004**: 284–289.
[61] Grant, M., Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.1. **2016**. Available at http://cvxr.com/cvx.
[62] Mattingley, J., Boyd, S. CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, **2010**, 13(1): 1–27.
[63] Chu, E., Parikh, N., Domahidi, A., Boyd, S. Code generation for embedded second-order cone programming. In: Proceedings of the European Control Conference, **2013**.
[64] Dueri, D., Acikmese, B., Scharf, D., Harris, M. W. Customized real-time interior-point methods for onboard powered descent guidance. *Journal of Guidance, Control, and Dynamics*, **2017**, 40(2): 197–212.
[65] Scharf, D. P., Regehr, M. W., Dueri, D., Acikmese, B., Vaughan, G. M., Benito, J. ADAPT: Demonstrations of onboard large-divert guidance with a reusable launch vehicle. In: Proceedings of the 2014 IEEE Aerospace Conference, **2014**: 1–18.
[66] Scharf, D., Acikmese, B., Dueri, D., Benito, J., Casoliva, J. Implementation and experimental demonstration of onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, **2017**, 40(2): 213–229.
[67] Liu, X., Shen, Z., Lu, P. Closed-loop optimization of guidance gain for constrained impact. *Journal of Guidance, Control, and Dynamics*, **2017**, 40(2): 453–460.
[68] Lu, P. Introducing computational guidance and control. *Journal of Guidance, Control, and Dynamics*, **2017**, 40(2): 193.

Xinfu Liu received his B.E. degree in the Department of Automation, Central South University, Changsha, China, in 2008, and Ph.D. degree in aerospace engineering from Iowa State University, Ames, USA, in 2013. From 2013 to 2016 he was a postdoctoral research associate at Beihang University, Beijing, China. Since 2016, he has been an associate professor in the School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China. His research interests are in optimal control, autonomous trajectory planning of flight vehicles, and convex optimization and its applications. E-mail: lau.xinfu@gmail.com.

Binfeng Pan is an associate professor in the School of Astronautics, Northwestern Polytechnical University (NPU). He received his M.S. and Ph.D. degrees in aerospace engineering from NPU in 2007 and 2010 respectively. His research interests are in the trajectory optimization, onboard guidance and control algorithms design. E-mail: panbinfeng@nwpu.edu.cn.

Ping Lu received his baccalaureate degree from the Beijing Institute of Aeronautics, and Ph.D. degree in aerospace engineering from the University of Michigan. He was on the faculty of aerospace engineering at Iowa State University from 1990 to 2016 where his last position was professor. He joined the San Diego State University in 2016 to be a professor and the Chair of the Aerospace Engineering Department. His research interests and expertise are in aerospace guidance, flight control, and autonomous trajectory planning and optimization. Prof. Lu was the recipient of the prestigious American Institute of Aeronautics and Astronautics (AIAA) Mechanics and Control of Flight Award in 2008, "for contributions in advanced guidance algorithms for entry and ascent fligh". He is an AIAA Fellow, and the Editor-in-Chief of *Journal of Guidance, Control, and Dynamics*. E-mail: plu@sdsu.edu.