

## Research Article

# Efficient Convex Optimization of Reentry Trajectory via the Chebyshev Pseudospectral Method

Chun-Mei Yu,<sup>1,2</sup> Dang-Jun Zhao<sup>1b</sup>,<sup>3</sup> and Ye Yang<sup>2</sup>

<sup>1</sup>College of Aerospace and Material Engineering, National University of Defense Technology, Changsha 100854, China

<sup>2</sup>Beijing Aerospace Automatic Control Institute, Beijing 100854, China

<sup>3</sup>School of Aeronautics and Astronautics, Central South University, Changsha 410083, China

Correspondence should be addressed to Dang-Jun Zhao; zhao\_dj@csu.edu.cn

Received 5 November 2018; Revised 18 February 2019; Accepted 7 April 2019; Published 2 May 2019

Guest Editor: Xiang Li

Copyright © 2019 Chun-Mei Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel sequential convex (SCvx) optimization scheme via the Chebyshev pseudospectral method is proposed for efficiently solving the hypersonic reentry trajectory optimization problem which is highly constrained by heat flux, dynamic pressure, normal load, and multiple no-fly zones. The Chebyshev-Gauss Legend (CGL) node points are used to transcribe the original dynamic constraint into algebraic equality constraint; therefore, a nonlinear programming (NLP) problem is concave and time-consuming to be solved. The iterative linearization and convexification techniques are proposed to convert the concave constraints into convex constraints; therefore, a sequential convex programming problem can be efficiently solved by convex algorithms. Numerical results and a comparison study reveal that the proposed method is efficient and effective to solve the problem of reentry trajectory optimization with multiple constraints.

## 1. Introduction

The trajectory optimization problem for a hypersonic vehicle constrained by heat rating, dynamic pressure, normal load, and other constraints related to the specified mission is often a highly constrained nonlinear dynamic programming problem which, in general, can be solved by two types of methods: direct and indirect methods [1]. Indirect methods rely on solving the necessary conditions which is analytically derived from the Pontryagin minimum principles [2]. On the contrary, indirect methods, the analytical necessary conditions do not have to be derived, while the parameterization techniques are used to convert the original infinite-dimensional dynamic optimization problem to a finite-dimensional nonlinear programming problem (NLP) which can be solved by some nonlinear programming algorithms such as the well-known sequential quadratic programming (SQP) [1, 3]. There are some software packages such as GPOPS [4] and GPOCS [5] based on direct methods for addressing the trajectory optimization problems. However, these mentioned

nonlinear programming algorithms cannot provide an a priori guarantee on the convergence and acquisition of a global optimal solution [6].

In recent years, convex optimization methods have been introduced to solve complex trajectory optimization problems because of their unique theoretical advantages: (1) rapid convergent rate and (2) insensitivity to the initial guess solution [7]. In [6, 8], Açıkmeşe and his coauthors proposed a lossless convexification method for solving the soft landing problem in the Mars exploration, then the highly constrained nonlinear dynamic programming problem is converted into its convex version which is efficiently solved by the second-order conic programming (SOCP) algorithm. Further, in [9], they improved their convex optimization algorithm based on the interior-point methods and thereby an efficient online algorithm for the guidance of soft landing. In [10], an SCvx optimization framework is proposed for solving nonconvex optimal control problems, in which the concave inequality constraint is successively approximated by linearization on the iterated solution rendering a

convex optimization problem suited to be solved by SOCP algorithms. This convex optimization method has been successfully used for addressing trajectory optimization problems of hypersonic vehicles [11].

Actually, as for the aforementioned SCvx optimization methods, the appropriate techniques of linearization and discretization are the key factors ensuring that the solution of the convexified problem is still the solution of the original problem. Hence, additional constraints of the trust region are applied to guarantee the validity of linearization in [10, 11]. Our previous study [7] reveals that the trust region should be carefully selected: if the trust region is given too large, the conditions of linearization will be violated and therefore the solution of the convexified problem is not that of the original problem; on the contrary, a smaller trust region will result in larger iterations and make the convergent rate slower.

In the procedure of discretization mentioned in [6–11], the uniformly distributed discretized points are chosen to transform the original infinite-dimensional optimization problem into a finite-dimensional parameter optimization problem. With such numerical discretization scheme, the discretized interval should be sufficiently small in order to obtain a sufficiently accurate solution, while this leads a high-dimensional transformed problem which is time-consuming to be solved. Actually, the pseudospectral methods have been widely used to solve the trajectory optimization problems, such as the Radau-Gauss pseudospectral method in [4, 5] and the Chebyshev method in [12–14]. The Chebyshev pseudospectral method is a special case of the general spectral methods, in which the functions can be expanded in terms of interpolating polynomials and the expansion coefficients are the values of the function at the Gauss quadrature node points, thereby having the best accuracy in interpolation of a function [12]. It has been shown that interpolation at the Chebyshev-Gauss Legend (CGL) nodes presents a closest result to the optimal polynomial approximation to a given function [15].

In this study, we develop a new SCvx optimization algorithm based on the Chebyshev pseudospectral method to improve the SCvx optimization method proposed in our previous work [7], in which the dynamic programming problem of reentry trajectory optimization is transcribed into a nonlinear programming problem by using the equispace discretizing technique, and then the convexification method and SCvx algorithm are employed to efficiently obtain the optimal trajectory. However, in order to obtain a sufficient well solution, the discretization points in the manner of equispace must be sufficiently large due to the requirement of discretizing accuracy. Based on the advantages of the Chebyshev pseudospectral method, CGL node points are used to discretize and approximate the state and control variables, therefore having more accurate approximations and less discretized points than using equispace discretized points. The less discretized points mean that less decision variables in the transformed SCvx problem and, consequently, the computational cost will be dramatically reduced, and this is validated by numerical study results in Section 4.

## 2. Problem Formulation

The reentry trajectory optimization problem, including the reentry motion model, various constraints, and the performance index, is formulated in this section. Further, the corresponding convexification techniques are particularly demonstrated.

*2.1. Reentry Dynamics of CAV.* For simplicity, the nonrotating earth with constant gravitational acceleration is applied during modelling the motion of CAV; meanwhile, it assumes that CAV's motion is with small flight path angle and limited control input (bank angle and angle of attack). Then, the nondimensional equation of CAV is given as [16].

$$\begin{cases} \dot{x} = v \cos \theta, \\ \dot{y} = v \sin \theta, \\ \dot{h} = v\gamma, \\ \dot{v} = \frac{-Bv^2 e^{-\beta R_e h} (1 + \lambda^2)}{2E^*}, \\ \dot{\gamma} = Bv e^{-\beta R_e h} \lambda \cos \sigma - 1/v + \nu, \\ \dot{\theta} = Bv e^{-\beta R_e h} \lambda \sin \sigma. \end{cases} \quad (1)$$

Here, the independent variable is dimensionless time  $t$  which is normalized by  $\sqrt{R_e/g_e}$  ( $R_e$  is the earth's radius,  $g_e$  is the gravitational acceleration at sea level), states  $x$  and  $y$  are the horizontal positions normalized by the earth's radius  $R_e$ , states  $h$  and  $v$  are the dimensionless altitude and velocity of the vehicle, and  $\gamma$  and  $\theta$  are the flight path angle and heading angle, respectively. The vehicle-specific constant  $B$  is defined by  $B = \rho_0 R_e S_{\text{ref}} C_L^* / (2m)$ , in which  $\rho_0$  is the atmospheric density at sea level,  $S_{\text{ref}}$  is the aerodynamic reference area,  $m$  is the vehicle mass, and  $C_L^*$  and  $C_D^*$ , respectively, are the coefficients of lift and drag which produce the maximum lift-to-drag ratio  $E^* = C_L^* / C_D^*$  for CAV. The control variables in (1) are the bank angle  $\sigma$  and the normalized coefficient of lift  $\lambda = C_L / C_L^*$  where  $C_L$  is the lift coefficient of the vehicle. Thus, the coefficient of lift and drag can be represented by  $C_L = \lambda C_L^*$  and  $C_D = C_L^* (1 + \lambda^2) / 2E^*$ , respectively. The further details of the aerodynamics of CAV can be found in the literature [17]. For convenience, we rewrite (1) as the following general nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \quad (2)$$

where the state vector  $\mathbf{x} = [x, y, h, v, \gamma, \theta]^T$  and the control vector is  $\mathbf{u} = [\lambda, \sigma]^T$ .

*2.2. Flight Constraints.* During the entry flight of CAV, the strong path constraints, such as heating flux, dynamic pressure, and load factor, should be satisfied for guaranteeing the safety of the vehicle. Moreover, in many cases, to avoid the enemy's detection and interception, no-fly zone constraints should be considered in the trajectory planning such that the vehicle keeps away from the area with the deployment of missile defense systems.

**2.2.1. Path Constraints.** In this research, the path constraints including the maximal heating rate, dynamic pressure, and load factor are considered for the safe flight of the vehicle. According to [7], the dimensional atmospheric density  $\rho$  is reasonably assumed to be an exponential function of the nondimensional altitude  $h = (R - R_e)/R_e$  with the form

$$\rho = \rho_0 e^{-\beta R_e h}. \quad (3)$$

Then, the normalized path constraints are defined by

$$\mathbf{P}_1(\mathbf{x}(t), t) = \begin{bmatrix} \bar{k}_Q e^{-0.5\beta R_e h} v^3 - 1 \\ \bar{k}_q e^{-\beta R_e h} v^2 - 1 \\ \bar{k}_n e^{-\beta R_e h} v^2 \sqrt{4E^* \lambda^2 + (1 + \lambda^2)^2} - 1 \end{bmatrix} \leq \mathbf{0}_{3 \times 1}, \quad (4)$$

in which the normalized coefficients are defined by  $\bar{k}_Q = k_Q \rho_0 (\sqrt{g_e R_e})^3 / \dot{Q}_{\max}$ ,  $\bar{k}_q = 0.5 \rho_0 g_e R_e / q_{\max}$ , and  $\bar{k}_n = 0.5 \rho_0 R_e S_{\text{ref}} C_L^{*2} / (m n_{\max})$ , where  $\dot{Q}_{\max}$  (W/m<sup>2</sup>),  $q_{\max}$  (N/m<sup>2</sup>), and  $n_{\max}$ , respectively, are the maximum allowable heating rate, dynamic pressure, and loading factor.

**2.2.2. No-Fly Zone Constraints.** No-fly zones (NFZs) are considered in this paper which are defined as circular exclusion zones in the horizontal plane with infinite altitude; thus,  $n_{\text{NFZ}}$  NFZs can be specified by their centre  $\mathbf{C}_j^{\text{NFZ}} = [x_j, y_j]^T$  and the corresponding radius  $R_j^{\text{NFZ}}$  for  $j = 1, \dots, n_{\text{NFZ}}$ . Then, NFZs constraints are formulated as

$$N_j(\mathbf{x}(t), \mathbf{C}_j^{\text{NFZ}}, R_j^{\text{NFZ}}) = \left\| \begin{bmatrix} x - x_j \\ y - y_j \end{bmatrix} \right\|^T \geq R_j^{\text{NFZ}}. \quad (5)$$

**2.2.3. Boundary Constraints.** According to the mission profile of CAV, the vehicle's entry start and the target point are prescribed; therefore, the following boundary constraints are

$$\mathbf{\Gamma}(\mathbf{x}(t_0), \mathbf{x}_0) = \mathbf{0}, \quad \mathbf{\Psi}(\mathbf{x}(t_f), \mathbf{x}_f) = \mathbf{0}, \quad (6)$$

where  $\mathbf{x}_0$  and  $\mathbf{x}_f$ , respectively, are the given initial states and terminal states which should be satisfied in the optimized trajectory [16].

**2.2.4. Control Constraints.** During the entry of CAV, due to the vehicle's aerodynamic characteristics, the normalized lift coefficient  $\lambda$  is confined in a certain range (for example [0,2]). Further, the bank angle  $\sigma$  is bounded to guarantee the stability. Then, the inequality constraints imposed on the controls are formulated as

$$\mathbf{C}(\mathbf{u}, t) = \begin{bmatrix} |\sigma| - \sigma_{\max} \\ \lambda - 2 \\ -\lambda \end{bmatrix} \leq \mathbf{0}. \quad (7)$$

It is noteworthy that if we directly use  $\sigma$  and  $\lambda$  as control variables in the followed SCvx algorithm, the sinusoidal function of  $\sigma$  in (1) will result in the chattering phenomenon in the solution. The detailed reason can be found in [11]. Consequently, in accordance with the treatment proposed in [18], three new control variables are defined as follows to replace the original control variables in order to conveniently convexify the control constraints.

$$\begin{aligned} u_1 &= \lambda \cos \sigma, \\ u_2 &= \lambda \sin \sigma, \\ u_3 &= \lambda^2 = u_1^2 + u_2^2, \\ \mathbf{u} &= [u_1, u_2, u_3]^T. \end{aligned} \quad (8)$$

Then, the constraints in (7) are reformulated as

$$\tilde{\mathbf{C}}(\mathbf{u}, \tau) = \begin{bmatrix} -u_3 \\ u_3 - 4 \\ -u_2 - u_1 \tan \sigma_{\max} \\ u_2 - u_1 \tan \sigma_{\max} \end{bmatrix} \leq \mathbf{0}, \quad (9)$$

with an auxiliary equality constraint

$$C_e = u_1^2 + u_2^2 - u_3 = 0. \quad (10)$$

**2.3. Reentry Trajectory Optimization Problem.** In the general trajectory optimization problem, there are several choices of performance indices to specify different optimization objectives such as maximum range, minimum heat load, and minimum time. In this research, minimum time is chosen as the performance index; thus, the reentry trajectory optimization problem is formulated as

$$\begin{aligned} P0 : \quad & \min \int_0^{t_f} 1 dt, \\ & \text{subject to (2), (4), (5), (6), (9), (10)}. \end{aligned} \quad (11)$$

### 3. Sequential Convex Optimization Based on the Chebyshev Pseudospectral Method

In this section, the Chebyshev pseudospectral method is introduced to reformulate the original dynamic optimization problem (11) as an SCvx programming problem in order to be efficiently solved by the convex optimization algorithms. First, the infinite-dimensional trajectory optimization problem is discretized by the Chebyshev pseudospectral method and henceforth a finite-dimensional parameter optimization problem. Then, the sequential convex optimization problem is formulated to be solved by the SOCP algorithm.

**3.1. Chebyshev Pseudospectral Method.** In the classical Chebyshev pseudospectral method, the CGL points are given by  $\tau_k = \cos(\pi k/N)$  for  $k = 0, \dots, N$ , then the node points range from 1 to -1. Since such order is inconvenient for the

trajectory optimization problem, then the modified CGL points proposed by [12] with the form,

$$\tau_k = -\cos\left(\frac{\pi k}{N}\right), \quad k = 0, \dots, N, \quad (12)$$

are employed in this work. The corresponding  $i^{\text{th}}$  order Chebyshev polynomial is

$$P_i = \cos(i \cos^{-1}\tau), \quad i = 0, \dots, N, \quad (13)$$

where  $\tau \in [-1, 1]$  can be mapped from the independent variable  $t$  by the equation  $t = (t_f - t_0)\tau/2 + (t_f + t_0)/2$ . If  $t_0 = 0$ , then  $t = t_f(\tau + 1)/2$ . Consequently, problem P0 is equivalent to the following problem P1.

$$\text{P1 : min} \quad t(1), \quad (14)$$

$$\text{subject to} \quad \frac{d\mathbf{X}}{d\tau} = \frac{t_f}{2} \begin{bmatrix} \mathbf{f}(\mathbf{x}, \mathbf{u}, \tau) \\ 1 \end{bmatrix} = \mathbf{F}(\mathbf{X}, \mathbf{U}, \tau), \quad \tau \in [-1, 1], \quad (15)$$

$$\begin{aligned} \Gamma(\mathbf{X}(-1), \mathbf{X}_0) &= \mathbf{0}, \\ \Psi(\mathbf{X}(1), \mathbf{X}_f) &= \mathbf{0}, \end{aligned} \quad (16)$$

$$\mathbf{G}(\mathbf{X}, \mathbf{U}, \tau) = \begin{bmatrix} \mathbf{P}_1(\mathbf{x}(\tau), \tau) \\ -N_j \left( \mathbf{x}(\tau), \mathbf{C}_j^{\text{NFZ}}, R_j^{\text{NFZ}} \right) \\ \leq 0, \quad j = 1, \dots, n_{\text{NFZ}}, \end{bmatrix} \quad (17)$$

$$\mathbf{C}(\mathbf{U}, \tau) = \begin{bmatrix} \tilde{\mathbf{C}}(\mathbf{u}, \tau) \\ -t_f \end{bmatrix} \leq 0, \quad (18)$$

$$C_e = 0, \quad (19)$$

where the augment state  $\mathbf{X} = [\mathbf{x}^T, t]^T$  and the control variable  $\mathbf{U} = [\mathbf{u}^T, t_f]^T$  are used in (15) to reformulate the original time-free problem (11) as a time-fixed problem for the convenience of the following discretized manipulations; meanwhile, Lagrange's problem P0 is converted into Mayer's problem, which is suited to be solved by convex programming since its performance index is linear in nature. In (16),  $\mathbf{X}_0 = [\mathbf{x}_0^T, t_0]^T$  and  $\mathbf{X}_f = [\mathbf{x}_f^T, t_f]^T$ . The constraints in (17) are the general form of the aforementioned nonlinear constraints (4) and (5). The augmented control variable  $t_f$  is constrained by  $t_f \geq 0$  in (18). The state and control vectors of (15) can be approximated by

$$\mathbf{X}(\tau) = \sum_{i=0}^N \mathbf{X}(\tau_i) \Phi_i(\tau), \quad (20)$$

$$\mathbf{U}(\tau) = \sum_{i=0}^N \mathbf{U}(\tau_i) \Phi_i(\tau), \quad (21)$$

where the basis  $N$ -order Lagrange interpolating polynomials  $\Phi_i(\tau)$  for  $i = 0, 1, \dots, N$  are given by

$$\Phi_i(\tau) = \frac{(-1)^{i+1} (1-t^2) \dot{P}_N(\tau)}{N^2 c_i (\tau - \tau_i)}, \quad (22)$$

with

$$c_i = \begin{cases} 2, & i = 0, N, \\ 1, & 1 \leq i \leq N-1. \end{cases} \quad (23)$$

Differentiating (20) at each CGL point yields the derivative approximation with the following form

$$\dot{\mathbf{X}}(\tau_k) = \sum_{i=0}^N \mathbf{X}(\tau_i) \dot{\Phi}_i(\tau_k) = \sum_{i=0}^N D_{ki} \mathbf{X}(\tau_i), \quad (24)$$

where  $D_{ki}$  are the entries of the differentiation matrix  $\mathbf{D} \in \mathbf{R}^{(N+1) \times (N+1)}$  and can be obtained by

$$\mathbf{D} := [D_{ki}] := \begin{cases} -\left(\frac{c_k}{c_i}\right) \left[ \frac{(-1)^{i+k}}{\tau_k - \tau_i} \right], & j \neq k, \\ \frac{\tau_k}{2(1-\tau_k^2)}, & 1 \leq i = k \leq N-1, \\ -\frac{2N^2+1}{6}, & i = k = 0, \\ \frac{2N^2+1}{6}, & i = k = N. \end{cases} \quad (25)$$

Thus, the dynamic constraint of (15) is transcribed into an algebraic constraint as

$$\sum_{i=0}^N D_{ki} \mathbf{X}(\tau_i) - \mathbf{F}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) = 0, \quad (k = 0, \dots, N), \quad (26)$$

where  $\mathbf{X}_k \equiv \mathbf{X}(\tau_k)$  and  $\mathbf{U}_k \equiv \mathbf{U}(\tau_k)$ . It is to be noted that if the function  $\mathbf{F}(\mathbf{X}, \mathbf{U}, \tau)$  in (15) is a linear function of  $\mathbf{X}$  and  $\mathbf{U}$ , then (26) will be a series of linear equality constraints which are convex in nature. Unfortunately,  $\mathbf{F}(\mathbf{X}, \mathbf{U}, \tau)$  is a strong nonlinear function; hence, the further convexification technique should be employed in order to formulate an SCvx problem.

**3.2. Convexification.** In the above subsection, the dynamic constraint is transformed into an algebraic constraint via the Chebyshev pseudospectral method, and the trajectory optimization problem can be reformulated as an NLP problem which cannot be directly solved by convex optimization algorithms. Hence, appropriate convexification techniques are required to reformulate problem P1 as an SCvx problem. It is obvious that the constraints in (15) and (17)–(19) are concave and should be convexified.

3.2.1. *Convexification of Dynamic Constraints.* Due to the iterative nature of sequential optimization algorithms, we suppose a reference trajectory denoted by  $\{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)\}$  which is provided by the  $n$ -th iteration solution of the SCvx algorithm, then in the  $(n+1)$ -th iteration, the dynamic equation (15) can be linearized along the reference trajectory  $\{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)\}$  as

$$\frac{d\mathbf{X}}{d\tau} = \mathbf{A}^n(\mathbf{X}^n, \mathbf{U}^n, \tau)\mathbf{X} + \mathbf{B}^n(\mathbf{X}^n, \mathbf{U}^n, \tau)\mathbf{U} + \mathbf{C}^n(\mathbf{X}^n, \mathbf{U}^n, \tau), \quad (27)$$

where  $\mathbf{A}^n$  and  $\mathbf{B}^n$ , the Jacobian matrices of  $\mathbf{F}$  with respect to  $\mathbf{X}$  and  $\mathbf{U}$ , are given by

$$\mathbf{A}^n = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{X}} \right|_{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & t_f \cos \theta & 0 & -t_f v \sin \theta & 0 \\ 0 & 0 & 0 & t_f \sin \theta & 0 & t_f v \cos \theta & 0 \\ 0 & 0 & 0 & t_f \gamma & t_f v & 0 & 0 \\ 0 & 0 & \frac{t_f \beta R_c B v^2 e^{-\beta R_c h} (1 + u_3)}{2E^*} & \frac{-t_f B v e^{-\beta R_c h} (1 + u_3)}{E^*} & 0 & 0 & 0 \\ 0 & 0 & -t_f \beta R_c B v e^{-\beta R_c h} u_1 & t_f (B e^{-\beta R_c h} u_1 + 1/v^2 + 1) & 0 & 0 & 0 \\ 0 & 0 & -t_f \beta R_c B v e^{-\beta R_c h} u_2 & t_f B e^{-\beta R_c h} u_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Big|_{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)},$$

$$\mathbf{B}^n = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \right|_{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & v \cos \theta \\ 0 & 0 & 0 & v \sin \theta \\ 0 & 0 & 0 & v \gamma \\ 0 & 0 & \frac{-t_f B v^2 e^{-\beta R_c h}}{2E^*} & \frac{-B v^2 e^{-\beta R_c h} (1 + u_3)}{2E^*} \\ t_f B v e^{-\beta R_c h} & 0 & 0 & B v e^{-\beta R_c h} u_1 - 1/v + v \\ 0 & t_f B v e^{-\beta R_c h} & 0 & B v e^{-\beta R_c h} u_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Big|_{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)} \quad (28)$$

The residue term can be calculated by

$$\mathbf{C}^n(\mathbf{X}^n, \mathbf{U}^n, \tau) = \mathbf{F}(\mathbf{X}^n, \mathbf{U}^n, \tau) - \mathbf{A}^n(\mathbf{X}^n, \mathbf{U}^n, \tau)\mathbf{X}^n - \mathbf{B}^n(\mathbf{X}^n, \mathbf{U}^n, \tau)\mathbf{U}^n. \quad (29)$$

Similar to references [7, 10], additional constraints of the trust region are denoted by the following element-wise inequalities

$$\begin{aligned} |\mathbf{X} - \mathbf{X}^n| &< \boldsymbol{\varepsilon}_X, \\ |\mathbf{U} - \mathbf{U}^n| &< \boldsymbol{\varepsilon}_U, \end{aligned} \quad (30)$$

where  $\boldsymbol{\varepsilon}_X \in \mathbf{R}^7$  and  $\boldsymbol{\varepsilon}_U \in \mathbf{R}^4$  are imposed to make the linearized system sufficiently approximate to the original nonlinear system. Actually, the constraints of the trust region (30)

confine the deviated trajectory in a prescribed neighbourhood about the reference trajectory  $\{\mathbf{X}^n(\tau), \mathbf{U}^n(\tau)\}$ .

Discretizing (27) on the CGL node points like (26) yields the following convex (linear) algebraic equality constraints:

$$\begin{aligned} \sum_{i=0}^N D_{ki} \mathbf{X} &= \mathbf{A}_k^n(\mathbf{X}_k^n, \mathbf{U}_k^n, \tau) \mathbf{X} + \mathbf{B}_k^n(\mathbf{X}_k^n, \mathbf{U}_k^n, \tau) \mathbf{U} \\ &+ \mathbf{C}_k^n(\mathbf{X}_k^n, \mathbf{U}_k^n, \tau), \quad (k=0, \dots, N), \end{aligned} \quad (31)$$

which will enforce the solution at  $N+1$  CGL node points exactly satisfying the dynamic constraint (15).

3.2.2. *Convexification of Path Constraints.* Each entry of the concave path constraints (17) can be represented by a generalized inequality as

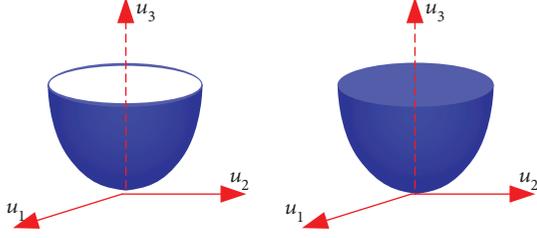


FIGURE 1: [18] Relaxation of control constraints. (Left) original nonconvex set: blue surface (10). (Right) relaxed convex set: blue cone includes its surface (34) Reproduced from Liu et al. [18] (under the Creative Commons Attribution License/public domain).

TABLE 1: CAV mission description.

	Latitude (deg)	Longitude (deg)	Radius (km)
Initial point	N 28.5	W -80.6	—
NFZ1	N 20.1	W -3.4	177.7920
NFZ2	N 55.6	E 58.5	277.8000
Target	N 31.2	E 65.7	—

$$g_p(\mathbf{X}, \mathbf{U}, \tau) \leq 0, \quad p = 1, \dots, m, \quad (32)$$

where  $m$  is the number of path constraints.

In the scenario of trajectory linearization and discretization, the abovementioned concave inequality constraints can be reformulated as

$$g_p(\mathbf{X}_k^n, \mathbf{U}_k^n) + \nabla g_p^T(\mathbf{X}_k^n)(\mathbf{X} - \mathbf{X}_k^n) + \nabla g_p^T(\mathbf{U}_k^n)(\mathbf{U} - \mathbf{U}_k^n) \leq 0, \quad k = 0, 1, 2, \dots, N, \quad (33)$$

where  $\nabla g_p(\mathbf{X}_k^n)$  and  $\nabla g_p^T(\mathbf{U}_k^n)$ , respectively, are the gradients of  $g_p$  at  $\mathbf{X}_k^n$  and  $\mathbf{U}_k^n$ . Note that the trust region defined by (30) is necessary in order to guarantee the linearized constraints in (33) reasonably approximate to the original constraints in (17). Lemma 1 of [10] presented a theoretical illustration in which the feasible solution of P1 comes from the linearized inequality constraints. Equation (33) is also the feasible solution of the original problem.

**3.2.3. Convexification of Control Constraints.** The equality constraint on the control variables in (19) is obviously concave. We relax the strong equality constraints to a convex inequality constraint as follows [7, 18]:

$$u_1^2 + u_2^2 \leq u_3, \quad (34)$$

which is shown by the blue cone including its surface on the right side of Figure 1. Auxiliary inputs  $u_1$  and  $u_2$  are limited by  $u_3$  and the maximum bank angle  $\sigma_{\max}$  as illustrated by the left image in Figure 1. And the relaxed constraint illustrated by the right image in Figure 1 is obviously convex.

**3.3. Sequential Convex Optimization Problem.** After the convexifications of path constraints and the control constraints,

TABLE 2: Comparison of the minimum-time solution for CAV.

Method	Number of discretized points	Iteration number	Optimal index (s)	CPU time (s)
Cheb-SCvx	25	7	3056.4	5.59
	30	18	2860.5	14.20
	40	8	2862.6	6.75
	60	6	2909.5	5.68
Eq-SCvx	80	6	2968.4	12.21
	300	7	2869.2	11.15
GPOPS	250	10	2893.8	15.42
	-	-	3100.0	20.430

the optimal solution of P1 is obtained by solving the following relaxed sequential convex optimal problems defined on the CGL node points for  $n = 0, 1, 2, \dots$ :

$$P2 : \min t(1), \quad (35)$$

$$\begin{aligned} \text{subject to} \quad & \sum_{i=0}^N D_{ki} \mathbf{X} = \mathbf{A}_k^n(\mathbf{X}_k^n, \mathbf{U}_k^n, \tau) \mathbf{X} + \mathbf{B}_k^n(\mathbf{X}_k^n, \mathbf{U}_k^n, \tau) \mathbf{U} \\ & + \mathbf{C}_k^n(\mathbf{X}_k^n, \mathbf{U}_k^n, \tau), \\ & (k = 0, \dots, N, n = 1, 2, \dots), \end{aligned} \quad (36)$$

$$\begin{aligned} \Gamma(\mathbf{X}(-1), \mathbf{X}_0) &= \mathbf{0}, \\ \Psi(\mathbf{X}(1), \mathbf{X}_f) &= \mathbf{0}, \end{aligned} \quad (37)$$

$$g_p(\mathbf{X}_k^n, \mathbf{U}_k^n) + \nabla g_p^T(\mathbf{X}_k^n)(\mathbf{X} - \mathbf{X}_k^n) + \nabla g_p^T(\mathbf{U}_k^n)(\mathbf{U} - \mathbf{U}_k^n) \leq 0, \quad k = 0, 1, 2, \dots, N, \quad (38)$$

$$u_1^2 + u_2^2 \leq u_3, \quad (39)$$

$$0 \leq u_3 \leq 4, \quad (40)$$

$$-\tan(\sigma_{\max})u_1 \leq u_2 \leq \tan(\sigma_{\max})u_1, \quad (41)$$

$$\begin{aligned} |\mathbf{X} - \mathbf{X}^n| &< \boldsymbol{\varepsilon}_X, \\ |\mathbf{U} - \mathbf{U}^n| &< \boldsymbol{\varepsilon}_U. \end{aligned} \quad (42)$$

**3.4. SCvx Algorithm.** According to the solution procedure of the sequential convex programming method, the SCvx algorithm to find the original problem P0 is given as follows.

*Step 1.* Set  $n = 0$ , and choose an initial control profile  $\mathbf{U}^0(\tau_k)$  ( $k = 0, 1, 2, N$ ) on the CGL node points. An easy choice is set  $\lambda_k = 1$  and bank angle  $\sigma_k = 0$ . Then driven by  $\mathbf{U}^0(\tau_k)$ , numerical integration is conducted according to the dynamical model in (1); we henceforth have the initial state profile  $\mathbf{X}^0(\tau_k)$ .

*Step 2.* At the  $(n + 1)^{\text{th}}$  iteration, problem P2 is set up by using  $\{\mathbf{X}^n(\tau_k), \mathbf{U}^n(\tau_k)\}$ , meanwhile point-wisely checking

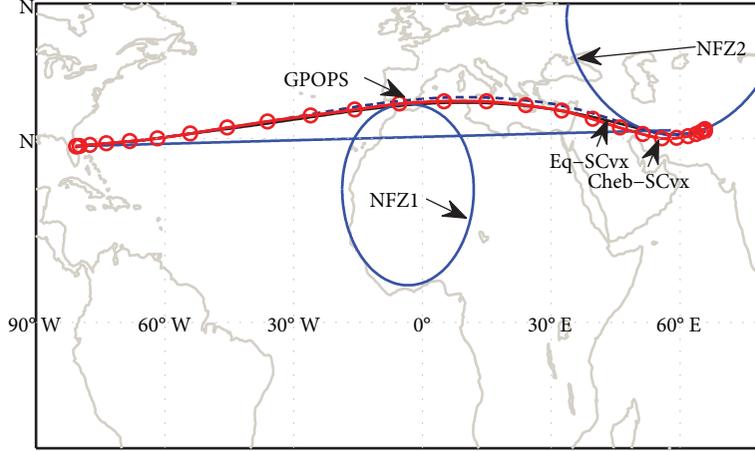


FIGURE 2: Ground tracks of CAV.

the violation states of the heat flux, dynamic pressure, load factor, and NFZ constraints defined in (4). If all or some of these constraints are violated, the corresponding convexified constraints in (38) are set up. Then solving problem P2 by the SOCP algorithm yields the solution denoted by  $\{\mathbf{X}^{n+1}(\tau_k), \mathbf{U}^{n+1}(\tau_k)\}$ .

*Step 3.* Check whether the convergence condition as follows is satisfied:

$$\max \quad |\mathbf{X}^{n+1} - \mathbf{X}^n| \leq \boldsymbol{\delta}, \quad (43)$$

which consists of a series of component-wise inequalities;  $\boldsymbol{\delta}$  is a prescribed sufficiently small constant vector. Moreover, the constraints defined in (4) are point-wisely calculated. If all these constraints on the CGL node points and the convergence condition in (43) are satisfied, go to Step 4; otherwise, set  $n = n + 1$  and go back to Step 2.

*Step 4.* The solution for P0 is found to be  $\{\mathbf{X}^{n+1}(\tau_k), \mathbf{U}^{n+1}(\tau_k)\}$ . Stop.

#### 4. Numerical Results

In this section, the hypersonic gliding vehicle (CAV-H) described by (1) is used to validate the proposed algorithm in the above section. The MATLAB modelling tool YALMIP [19] is used to formulate the SOCP problem P2, and a light embedded SOCP solver ECOS [20] is used to obtain the solution. The all-solution procedure is executed on a laptop computer with Intel Core i5-4200 at 2.5 GHz.

The CAV mission is described in Table 1, which defines the horizontal positions of the initial and target points, as well as two waypoints during the flight. The radius of the first NFZ is chosen to be much smaller than the turn capability of the CAV, while the second NFZ has a large enough radius. The path constraints imposed on the trajectory are given by  $\dot{Q}_{\max} = 4000 \text{ kW/m}^2$ ,  $q_{\max} = 5 \times 10^5 \text{ N/m}^2$ , and  $n_{\max} = 2.5g_e$ . The boundary conditions are as follows:  $h_0 = 122 \text{ km}$ ,  $v_0 =$

$7315.2 \text{ m/s}$ ,  $\gamma_0 = -1.5^\circ$ ,  $\theta_0 = 4^\circ$ ,  $h_f \geq 20 \text{ km}$ ,  $v_f \geq 2000 \text{ m/s}$ , and  $\gamma_f = -4^\circ$ . The terminal flight heading angle is not constrained. In addition, the bank angle are limited by  $\sigma_{\max} = \pi/3$ . The trust region in (42) is given as

$$|\mathbf{X} - \mathbf{X}^n| \leq \left[ 0.2 \quad 0.2 \quad \frac{10000}{R_e} \quad \frac{5000}{\sqrt{g_e R_e}} \quad \frac{\pi}{6} \quad \frac{\pi}{6} \right]^T, \quad (44)$$

$$|\mathbf{U} - \mathbf{U}^n| \leq [0.5 \quad 0.5 \quad 1 \quad 0.5]^T,$$

which is sufficiently large to satisfy Assumption 1 presented by [7]. The stopping criterion  $\boldsymbol{\delta}$  in (43) is set as  $\boldsymbol{\delta} = [0.03 \quad 0.03 \quad 0.003 \quad 0.04 \quad 0.06 \quad 0.03]^T$ .

To verify the proposed SCvx algorithm via Chebyshev pseudospectral method (denoted by Cheb-SCvx), a comparison study between the proposed SCvx method in [7] (denoted by Eq-SCvx) as well as GPOPS is conducted. In the Eq-SCvx method, the equal-space discretized scheme is employed, while GPOPS uses the hp-adaptive Radau pseudospectral method to solve optimal control problems. Similar to the proposed method in Section 3, we use the new controls  $u_1$ ,  $u_2$ , and  $u_3$  to replace the original controls ( $\lambda$  and  $\sigma$ ) in (1).

The comparison of numerical solutions obtained by Cheb-SCvx, Eq-SCvx, and GPOPS is presented in Table 2, which reveals that the proposed Cheb-SCvx method is more efficient and effective than Eq-SCvx and GPOPS while dealing with the highly constrained entry trajectory optimization problems. It is to be noted that, in the Eq-SCvx algorithm, when the discretized point number is less than 250, the algorithm cannot converge, and no solution can be obtained. The reason lies in the fact that less discretized points result in a larger distance between two adjacent points; thus, the linearization condition in (27) will be violated. And after multiple numerical experiments, we can ascertain that the appropriate number of CGL node points will extremely reduce the computational time and improve the performance index.

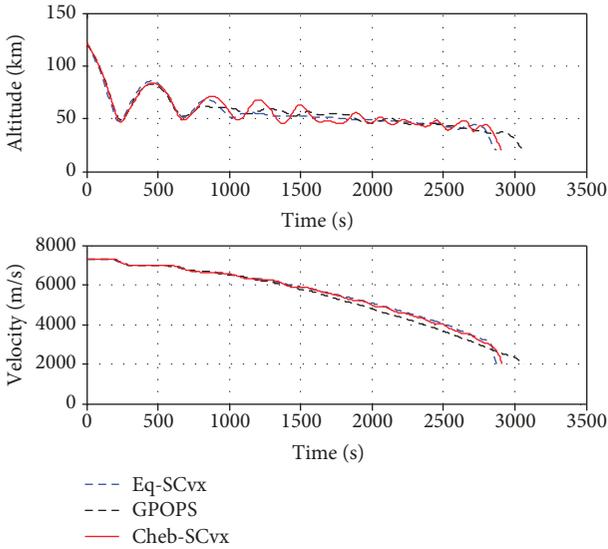


FIGURE 3: Altitude and velocity profile.

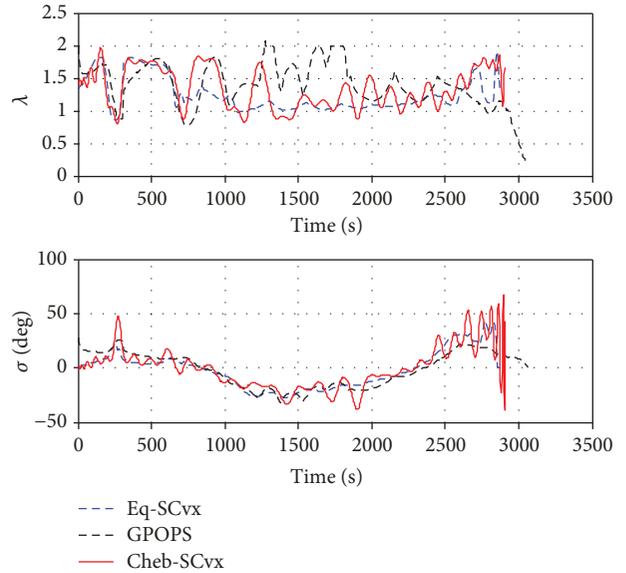


FIGURE 5: Control variables (normalized lift coefficient and bank angle) histories.

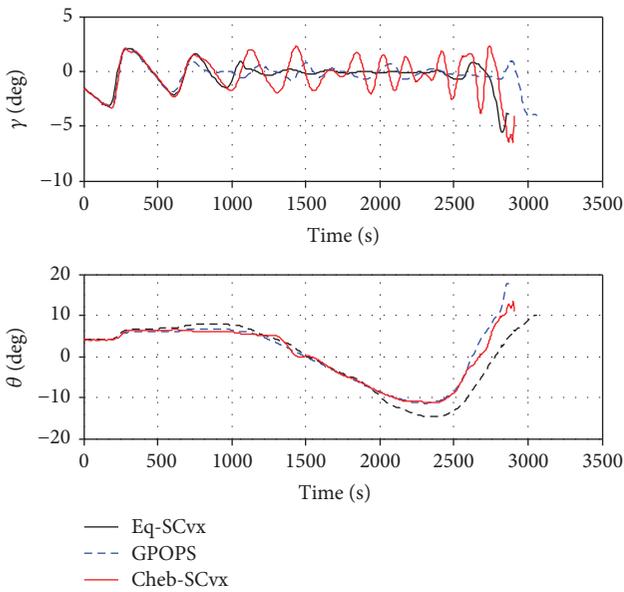


FIGURE 4: Flight path angle and heading angle histories.

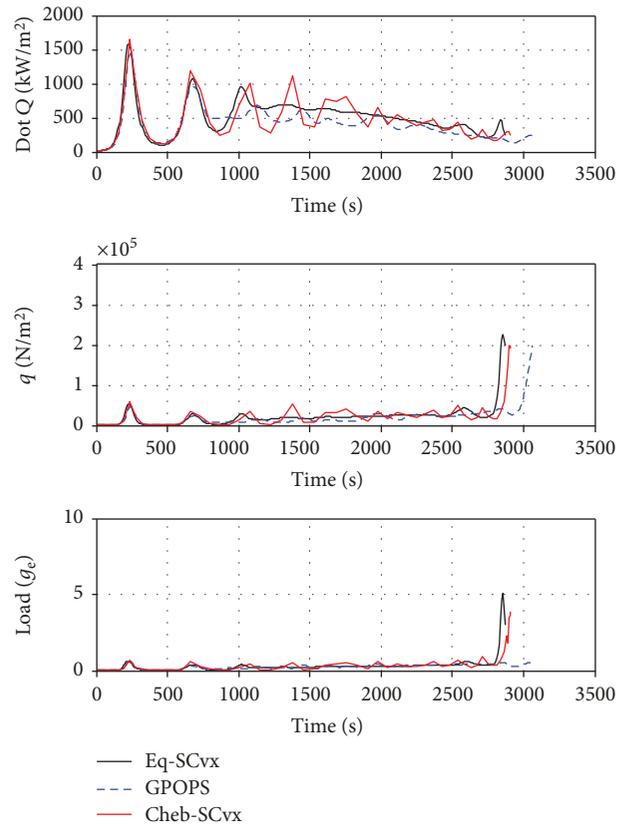


FIGURE 6: Heat flux, dynamic pressure, and normal load histories.

The ground track of CAV constrained by NFZs is presented in Figure 2, which reveals that both target and NFZ constraints are satisfied by three methods. However, the proposed method renders lower cost than the other two methods do. Altitude and velocity profiles are given in Figure 3, while the flight path angle and heading angle histories are illustrated in Figure 4. Although the terminal altitude, velocity, and heading angle are not assigned a fixed value but limited in a prescribed range, the overall trend of the solutions provided by the three methods is similar. The optimized control variables' histories are presented in Figure 5, which shows that the bank angles obtained by the Cheb-SCvx method are with a more chattering phenomenon than those obtained

by the other methods during the final phase, but the constraints on the bank angle are well satisfied. Further, the constraints on the heat rate, dynamic pressure, and normal load are all satisfied (as shown in Figure 6).

## 5. Conclusions

Inspired by the application of convex optimization in aerospace trajectory generation and optimization, a novel sequential convex programming algorithm based on the classical Chebyshev pseudospectral method is proposed to solve a highly constrained entry trajectory optimization problem with free final time. By using sequential linearization, convexification, and discretization on the CGL node points, the original concave optimization problem is approximated by a series of SOCP problems, which are solved by open-source solver ECOS. In this work, we concentrate on the conversion of a nonconvex problem to a convex space, so that the converted problem can be efficiently solved by the SOCP method. An efficient and accurate discretization method based on the Chebyshev interpolating polynomials are proposed to facilitate transcribing the dynamic constraint into an algebraic equality constraint; then, the convexification technique based on the linearization is used to set up the convex version of the original trajectory optimization problem. The numeric results reveal that the proposed method can dramatically reduce computational cost by appropriately choosing the number of discretized points and will be very competitive to fulfill the onboard optimization in the future.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the National Key R&D Program of China (Grant No. 2018YFC0810202).

## References

- [1] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, pp. 1–32, 2010.
- [2] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience, New York, 1962.
- [3] J. T. Betts, "Very low-thrust trajectory optimization using a direct SQP method," *Journal of Computational and Applied Mathematics*, vol. 120, no. 1–2, pp. 27–40, 2000.
- [4] D. Garg, M. Patterson, C. Darby et al., "Direct trajectory optimization and costate estimation of general optimal control problems using a Radau pseudospectral method," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Illinois, 2009.
- [5] G. T. Huntington, "Advancement and analysis of a gauss pseudospectral transcription for optimal control problems," Ph.D dissertation, Massachusetts institute of Technology, 2007.
- [6] B. Açıkmeşe and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.
- [7] D.-J. Zhao and Z.-Y. Song, "Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming," *Acta Astronautica*, vol. 137, pp. 60–69, 2017.
- [8] L. Blackmore, B. Acikmese, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for Mars landing using convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1161–1171, 2010.
- [9] D. Dueri, B. Açıkmeşe, D. P. Scharf, and M. W. Harris, "Customized real-time interior-point methods for onboard powered-descent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 197–212, 2017.
- [10] X. Liu and P. Lu, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 750–765, 2014.
- [11] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, pp. 227–241, 2016, Articles in advance.
- [12] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 160–166, 2002.
- [13] S. Shahmirzaee Jeshvaghani, A. B. Novinzaddeh, and F. Pazooki, "Multiple stage satellite launch vehicle ascent optimization using Chebyshev wavelets," *Aerospace Science and Technology*, vol. 46, pp. 321–330, 2015.
- [14] X. Wei, L. Liu, Y. Wang, and Y. Yang, "Reentry trajectory optimization for a hypersonic vehicle based on an improved adaptive fireworks algorithm," *International Journal of Aerospace Engineering*, vol. 2018, Article ID 8793908, 17 pages, 2018.
- [15] G. N. Elnagar and M. A. Kazemi, "Pseudospectral Chebyshev optimal control of constrained nonlinear dynamical systems," *Computational Optimization and Applications*, vol. 11, no. 2, pp. 195–217, 1998.
- [16] T. R. Jorris and R. G. Cobb, "Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 551–572, 2009.
- [17] T. R. Jorris, "Common aero vehicle autonomous reentry trajectory optimization satisfying waypoint and no-fly zone constraints," Doctor, Engineering and Management, Air University, Ohio, US, 2007.
- [18] X. Liu et al., "Exact convex relaxation for optimal flight of aerodynamically controlled missiles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1881–1892, 2016.
- [19] J. Löfberg, "YALMIP : a toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [20] A. Domahidi, E. Chu, and S. Boyd, "ECOS: an SOCP solver for embedded systems," in *European Control Conference*, Zurich, Switzerland, 2013.

