Aerospace Science and Technology 92 (2019) 314–325

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Robust trajectory optimization using polynomial chaos and convex optimization

Fenggang Wang [a,b], Shuxing Yang [a,b], FenFen Xiong [a,b,*], Qizhang Lin [a,b], Jianmei Song [a,b]

[a] School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China
[b] Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education, Beijing 100081, China

ARTICLE INFO

Article history:
Received 27 July 2018
Received in revised form 15 March 2019
Accepted 5 June 2019
Available online 10 June 2019

Keywords:
Robust trajectory optimization
Uncertainty
Convex optimization
Polynomial chaos

ABSTRACT

The polynomial chaos (PC) theory and direct collocation method have been integrated to solve a lot of robust trajectory optimization problems. However, the computational cost and memory consumption increase significantly with the increase of the dimension of uncertain factors, and the nonlinearity of dynamic equations. To address this issue, a novel robust trajectory optimization procedure combining PC with the convex optimization technique is proposed in this paper. With the proposed procedure, the trajectory optimization can be implemented with high accuracy and efficiency, by taking advantage of the high accuracy of PC in addressing UP for highly nonlinear dynamics and the high efficiency of convex optimization in solving optimal control. The proposed robust trajectory optimization procedure is applied to two examples and compared with the existing method employing PC and the pseudospectral method. The results show that the proposed procedure can obtain highly accurate results similar to the existing method. Moreover, with the increase of random dimension, the optimal trajectory can still be generated very efficiently without significant increase of computational cost. These results demonstrates the effectiveness of the proposed procedure.

1. Introduction

Trajectory optimization is an important process in flight vehicle design and has gained a lot of attention during recent years [1–6]. However, these works have generally focused on nominal trajectory performance without considering trajectory robustness. Realistic flight dynamic models established for trajectory computation have uncertainties such as those in aerodynamic datasets obtained from experimental tests and initial states [7], which may induce significant deviation of the actual trajectory from the nominal trajectory. For spacecraft rendezvous and Mars reentry, some constraints may become violated as the satisfaction margin is very small for deterministic trajectory optimization, resulting in mission failure. For certain aircraft with small control surfaces, even if feedback control is employed, large deviation from the nominal trajectory is difficult to eliminate owing to their limited maneuvering capability. Therefore, robust trajectory optimization has been proposed to make the nominal trajectory inherently less sensitive to uncertainties by using the scenario of robust optimization [8], which is also more beneficial to closed-loop control [9–12].

The basic idea of solving the robust trajectory optimization problem is to transform the stochastic optimal control problem with stochastic ordinary differential equations (ODEs) into a deterministic optimal control problem with deterministic ODEs. One key step in dealing with robust trajectory optimization is how to solve the transformed deterministic optimal control problem. The two commonly used approaches are the indirect and direct methods. The indirect method is highly sensitive to initial estimates of the solution and the necessary conditions are generally very complicated for nonlinear problems with many constraints [8]. Thus, direct methods such as direct shooting [13], direct collocation [14], and the pseudo-spectral method [15,16] are much more commonly used in practice. With the direct method, the optimal control problem is converted into a nonlinear programming (NLP) problem by discretizing the system controls and states.

The other key step is the transformation from the stochastic optimal control problem into the deterministic version, i.e. dynamic uncertainty propagation (UP). As a highly efficient UP method, the linear covariance analysis (LCA) method has been employed for UP in many robust trajectory optimization problems. Saunders employed LCA in robust trajectory planning of atmospheric reentry in conjunction with the Legendre pseudo-spectral method [10]. Tang et al. implemented optimal robust linearized impulsive rendezvous using LCA and a genetic algorithm [11]. Chen et al. applied LCA

* Corresponding author at: School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China.
E-mail address: fenfenx@bit.edu.cn (F. Xiong).

https://doi.org/10.1016/j.ast.2019.06.011
1270-9638/

Nomenclature

d	dimension of a random vector	$\boldsymbol{w}$	random parameter vector
g_j	the jth constraint function	$\boldsymbol{W}$	polynomial chaos coefficients of random parameters $\boldsymbol{w}$
J	objective function	μ	mean value
$P+1$	number of polynomial chaos coefficients	σ	standard deviation value
T	sampling period	Φ	d-dimensional orthogonal polynomial
$\boldsymbol{x}$	state vector	$\boldsymbol{\Delta}$	standard random vector
$\boldsymbol{X}$	polynomial chaos coefficients of random states $\boldsymbol{x}$	$\|\bullet\|$	2-norm of a vector
$\boldsymbol{u}$	control vector		

and the Chebyshev pseudo-spectral method to robust trajectory optimization of a gliding guided projectile [17]. As is well known, the LCA method assumes the system is a linear Gaussian process model. However, in reality, the sources of uncertainty do not necessarily follow a Gaussian distribution and the dynamic model may be highly nonlinear. Thus, large errors may be induced with the LCA method during dynamic UP [18,19]. An alternative approach to dynamic UP is the polynomial chaos (PC) method [20–22], which has been frequently utilized in trajectory robustness analysis and robust design in recent years due to its rigor and high accuracy. Ren and Zhu adopted PC to analyze the impact of uncertain factors in the entry dynamics on the landing precision of a representative Mars entry scenario [23]. Prabhakar et al. employed PC to obtain the evolution of the state uncertainty of a hypersonic air vehicle [24].

Considering the advantages of PC and the direct approach, many works have combined them to solve robust trajectory optimization problems. Fisher and Bhattacharya developed a PC-based framework for solving optimal robust trajectory generation problems with probabilistic uncertainty in the system parameter, in which the direct collocation method is employed to solve the transformed deterministic optimal control problem [25]. Similar to the work of Prabhakar et al., Li et al. applied PC and the direct Gaussian pseudo-spectral method to robust trajectory optimization of supersonic aircraft short-time climb with uncertainties in the aerodynamic data [26]; Xiong et al. employed them to robust trajectory optimization considering uncertainties in both the random parameter and initial states [27]. Although the PC-based direct approach has been commonly used in robust trajectory optimization, the computational cost may be very large and unaffordable. Firstly, the NLP problem transcribed by the direct method is generally non-deterministic polynomial-time hard (NP-hard). With the increase in nonlinearity in system dynamics, the dimensions of states, as well as the flight time, the number of discrete nodes for control and states may increase significantly, resulting in a large-scale NLP problem that is very difficult and computationally expensive to solve. On the other hand, when uncertainties are further considered, the increase in the number of discrete nodes is much more serious compared with the deterministic optimization case, especially when the dimension of uncertain factors is large, and so is the computational cost. During our study, it has been frequently observed that when solving robust trajectory optimization using PC and the direct method, the intensive memory consumption and large computational cost seriously impede the optimization process, inducing failure of optimization [22,27]. The intensive computational issue has also been indicated in [26,28]. It is also noted that in the existing works on robust trajectory optimization using PC and the direct method mentioned previously, only a few uncertain factors ($\leq$3) are considered [25–27], which is less common in reality.

In contrast to NLP, the convex optimization problem is computationally tractable in the sense that it can be solved by polynomial-time algorithms [29,30], which have been demonstrated to be much more efficient than the direct method in solving trajectory optimization. In recent years, many researchers have made significant efforts to solve the optimal control problems in aerospace engineering using convex optimization [31–34]. This is done by formulating the optimal control problem as a convex optimization problem whenever possible or applying various convexification techniques to convert the original non-convex formulation into a convex optimization formulation. It is the objective of this work to propose a new robust trajectory optimization strategy by combining the rigorous PC method with the efficient convex optimization technique to ensure both high accuracy and efficiency of the trajectory generation. With the proposed strategy, the PC method is employed for dynamic UP to calculate the mean and variance of the states, constraints, and performance objective by transforming the original stochastic optimal control into the deterministic version in a higher-dimensional state space. Then, the convex optimization technique is used to solve the transformed deterministic optimal control problem, during which the convexification techniques for transforming the non-convex dynamics, constraints, and objective are presented.

The remainder of this article is organized as follows. In Section 2, the robust trajectory optimization formulation is presented. In Section 3, the proposed robust trajectory optimization scheme using PC and convex optimization is introduced, with the PC for UP and the convexification process described in detail. In Section 4, comparative studies on the Van Der Pol oscillator and a Mars pinpoint landing problem is presented to verify the effectiveness of the proposed method. Conclusions are drawn in the final section.

2. Robust trajectory optimization formulation

A commonly used optimal control formulation of the deterministic trajectory optimization (P^{D}) is as follows:

$$
\text{Problem } P^{\mathrm{D}}
\begin{cases}
\text{find} & \boldsymbol{u}(t) \\
\min & J\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{w}\big) \\
\text{s.t.} & \dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{w}\big) \\
& g_j\big(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{w}\big) \leq 0, \quad j=1,\ldots,N_{ieq} \\
& \boldsymbol{u}^{\mathrm{LB}} \leq \boldsymbol{u}(t) \leq \boldsymbol{u}^{\mathrm{UB}} \\
& \boldsymbol{x}^{\mathrm{LB}} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}^{\mathrm{UB}} \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\
& \boldsymbol{x}(t_f) = \boldsymbol{x}_f
\end{cases}
\tag{1}
$$

where $\boldsymbol{x} = [x_1, \ldots, x_n]$ and $\boldsymbol{u} = [u_1, \ldots, u_m]$ are the state and control variables, respectively; $\boldsymbol{w}$ is the parameter vector $\boldsymbol{w} = [w_1, \ldots, w_s]$; $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w})$ is the system dynamic equation, *i.e.* ODEs; $g_j(\boldsymbol{x}(t), \boldsymbol{u}(t), w) \leq 0$ is the path constraint; $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ and $\boldsymbol{x}(t_f) = \boldsymbol{x}_f$ are boundary constraints at the initial (t_0) and terminal (t_f) time instant, respectively; $\boldsymbol{u}^{\mathrm{LB}}$ and $\boldsymbol{u}^{\mathrm{UB}}$ represent the lower and upper bounds of the control variables, respectively; $\boldsymbol{x}^{\mathrm{LB}}$ and $\boldsymbol{x}^{\mathrm{UB}}$ denote the lower and upper bounds of the state variables, respectively.

With the consideration of uncertainties, such as the ones from the initial states $\boldsymbol{x}(t_0)$ and parameters $\boldsymbol{w}$ in the system dynamic model (ODEs), the states $\boldsymbol{x}(t)$, constraints g_j, performance objective J, and the ODEs in (1) all become stochastic. It should be noted that if J is only a function of the control vector $\boldsymbol{u}$, it is deterministic as $\boldsymbol{u}$ is assumed to be deterministic, and thus UP is not required for J. In our work, it is assumed that J is also related to the state variables $\boldsymbol{x}$. The robust trajectory optimization problem (P^{R}) can be correspondingly formulated as follows:

$$
\text{Problem } P^{\mathrm{R}}
\begin{cases}
\text{find} & \boldsymbol{u}(t) & \\
\text{min} & J_\mu + k^* J_\sigma & \text{(a)} \\
\text{s.t.} & \dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}\big) & \text{(b)} \\
& g_{j\mu} + k^* g_{j\sigma} \le 0, \quad j = 1, ..., N_{ieq} & \text{(c)} \\
& \boldsymbol{u}^{\mathrm{LB}} \le \boldsymbol{u} \le \boldsymbol{u}^{\mathrm{UB}} & \text{(d)} \\
& \boldsymbol{x}^{\mathrm{LB}} + k^* \boldsymbol{x}_\sigma \le \boldsymbol{x}_\mu \le \boldsymbol{x}^{\mathrm{UB}} - k^* \boldsymbol{x}_\sigma & \text{(e)} \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_0 & \text{(f)} \\
& \boldsymbol{x}_\mu(t_f) = \boldsymbol{x}_f & \text{(g)} \\
& \boldsymbol{x}_\sigma(t_f) \le \boldsymbol{\varepsilon}_f & \text{(h)}
\end{cases}
\tag{2}
$$

where the variables with subscript μ and σ represent the expectation and standard deviation of them, respectively; k^* is a constant coefficient determined by designers; and $\boldsymbol{\varepsilon}_f$ is the allowable standard deviation of the stochastic states.

3. The proposed robust trajectory optimization scheme

A step-by-step description of the proposed robust trajectory optimization procedure using PC and the convex optimization technique is given as follows.

Step 1. Formulate the optimal control problem in (2) as a convex optimization problem whenever possible using techniques such as equivalent transformation, change in variables, and lossless relaxation. For a more detailed description of these techniques, the reader can refer to Ref. [35].

It is not mandatory to perform Step 1 for all the problems especially when the system ODEs are linear, which will not impact the final optimal results of the robust trajectory optimization.

Step 2. Transform the original stochastic ODEs into an expanded higher-dimensional deterministic ODE using the generalized PC method.

Represent each stochastic state variable $x_i(t)$ $(i = 1, ..., n)$ and parameter w_i $(i = 1, ..., s)$ as a generalized PC model.

$$
x_i(t) \approx \sum_{j=0}^{P} x_{ij}(t) \Phi_j(\boldsymbol{\Delta}) \tag{3}
$$

$$
w_i \approx \sum_{j=0}^{P} w_{ij} \Phi_j(\boldsymbol{\Delta}) \tag{4}
$$

where $P + 1 = \frac{(p+d)!}{p!d!}$ with d as the dimension of the uncertain factors and p as the order of the PC model, $\boldsymbol{\Delta}$ represents the standard random variable belonging to the Askey scheme [21], and $\Phi_j(\boldsymbol{\Delta})$ is the orthogonal polynomial term and is simplified as Φ hereafter.

All the PC coefficients of the random parameter w_i (w_{ij}) can be obtained based on the distribution of w_i, whereas the coefficients x_{ij} for the random state $x_i(t)$ are unknown. In some cases, if the uncertain factor does not follow the random distributions in the Askey scheme or exists in some raw data, the data-driven PC method [22] can be employed here. Substitute Eqs. (3) and (4) into the stochastic ODE in Eq. (2-b), based on which the Galerkin projection is conducted to transform the original stochastic ODE into an expanded deterministic ODE in a higher-dimensional state space including $(P+1)^*n$ coupled equations as follows:

$$
\frac{\mathrm{d}x_{il}(t)}{\mathrm{d}t} = \frac{1}{< \Phi_l^2 >} \left\langle f\left(\sum_{j=0}^{P} x_{ij}(t)\Phi_j, \boldsymbol{u}, \sum_{j=0}^{P} w_{ij}\Phi_j, t\right), \Phi_l \right\rangle,
$$
$$
i = 1, 2, ..., n \text{ and } l = 0, 1, ..., P \tag{5}
$$

Eq. (5) can be rewritten in a compact form as

$$
\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) \tag{6}
$$

where the vectors $\boldsymbol{X}$ and $\boldsymbol{W}$ respectively represent all the PC coefficients of the random states $\boldsymbol{x}$ and parameters $\boldsymbol{w}$,

$$
\boldsymbol{X} = \big[x_{10}(t), \cdots, x_{1P}(t), x_{20}(t), \cdots, x_{2P}(t), \cdots, x_{n0}(t), \cdots, x_{nP}(t)\big]^{\mathrm{T}} \tag{7}
$$

$$
\boldsymbol{W} = [w_{10}, \cdots, w_{1P}, w_{20}, \cdots, w_{2P}, \cdots, w_{s0}, \cdots, w_{sP}]^{\mathrm{T}} \tag{8}
$$

Step 3. Set up the equivalent deterministic trajectory optimization problem P^{ED} of P^{R} based on PC.

The robust trajectory optimization P^{R} is transformed into the equivalent deterministic trajectory optimization problem P^{ED} as follows.

$$
\text{Problem } P^{\mathrm{ED}}
\begin{cases}
\text{find} & \boldsymbol{u}(t) & \\
\text{min} & J_\mu(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) + k^* J_\sigma(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) & \text{(a)} \\
\text{s.t.} & \dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) & \text{(b)} \\
& g_{j\mu}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) + k^* g_{j\sigma}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) \le 0, \\
& \quad j = 1, ..., N_{ieq} & \text{(c)} \\
& \boldsymbol{u}^{\mathrm{LB}} \le \boldsymbol{u} \le \boldsymbol{u}^{\mathrm{UB}} & \text{(d)} \\
& \boldsymbol{x}^{\mathrm{LB}} + k^* \boldsymbol{x}_\sigma(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) \\
& \quad \le \boldsymbol{x}_\mu(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) \le \boldsymbol{x}^{\mathrm{UB}} - k^* \boldsymbol{x}_\sigma(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) & \text{(e)} \\
& x_{i0}(t_0) = x_{i0}, \quad i = 1, ..., n & \text{(f)} \\
& x_{i0}(t_f) = x_{if}, \quad i = 1, ..., n & \text{(g)} \\
& \boldsymbol{x}_{\sigma, t_f}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) \le \boldsymbol{\varepsilon}_f & \text{(h)}
\end{cases}
\tag{9}
$$

Step 4. Convexification of the nonlinear ODEs in Eq. (9-b).

In general, the ODEs of a dynamic system shown in Eq. (9-b) are nonlinear, which should be converted into linear ones firstly to facilitate the employment of convex optimization. The commonly used method is the successive linearization technique [35], and its good performance in convergence and accuracy have been demonstrated by substantial applications in aerospace engineering [31–35]. Let one reference trajectory sequence as $\{\boldsymbol{X}_r, \boldsymbol{u}_r\}$, at which the first-order Taylor expansion of Eq. (9-b) is performed as follows:

$$
\dot{\boldsymbol{X}} = \boldsymbol{A}_1 \boldsymbol{X} + \boldsymbol{B}_1 \boldsymbol{u} + \boldsymbol{C}_1 \tag{10}
$$

where $\boldsymbol{A}_1 = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}\big|_{\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}}$ $(\boldsymbol{A}_1 \in \mathbb{R}^{n(P+1) \times n(P+1)})$, $\boldsymbol{B}_1 = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\big|_{\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}}$ $(\boldsymbol{B}_1 \in \mathbb{R}^{n(P+1) \times m})$, $\boldsymbol{C}_1 = \boldsymbol{F}(\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}) - \boldsymbol{A}_1 \boldsymbol{X}_r - \boldsymbol{B}_1 \boldsymbol{u}_r$ $(\boldsymbol{C}_1 \in \mathbb{R}^{n(P+1) \times 1})$, $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}\big|_{\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}}$ and $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\big|_{\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}}$ respectively represent the partial derivatives of $\boldsymbol{F}$ with respect to $\boldsymbol{X}$ and $\boldsymbol{u}$ at the reference trajectory.

It should be noted that the first-order approximation is employed for linearization here, which is only reasonable within a small region. Therefore, the trust region constraints are correspondingly added as the convergence condition.

$$
\|\boldsymbol{X} - \boldsymbol{X}_r\| \le \delta_{\boldsymbol{X}}, \qquad \|\boldsymbol{u} - \boldsymbol{u}_r\| \le \delta_{\boldsymbol{u}} \tag{11}
$$

where $\delta_{\boldsymbol{x}}$ and $\delta_{\boldsymbol{u}}$ are constants.

Step 5. Perform dynamic UP on the linearized expanded ODEs shown in Eq. (10) to calculate the mean and standard derivation

of states ($\boldsymbol{x}_\mu$ and $\boldsymbol{x}_\sigma$), path constraints ($g_{j\mu}$ and $g_{j\sigma}$), and performance functions (J_μ and J_σ).

The mean $x_{i\mu}(t)$ and standard deviation $x_{i\sigma}(t)$ of the original random state variable $x_i(t)$ $(i = 1, 2, ..., n)$ can be calculated analytically as follows [22], which are functions with respect to its PC coefficients.

$$x_{i\mu}(t) = \mathrm{E}\left[\sum_{j=0}^{P} x_{ij}(t)\Phi_j\right] = x_{i0}(t) \tag{12}$$

$$x_{i\sigma}(t) = \sqrt{\mathrm{E}\left[x_i^2(t)\right] - \mathrm{E}^2\left[x_i(t)\right]} = \sqrt{\sum_{j=1}^{P}\left(x_{ij}(t)\right)^2 \mathrm{E}\left[\Phi_j^2(\xi)\right]} \tag{13}$$

As the path constraint g_j and performance function J are generally functions with respect to the stochastic state variables $\boldsymbol{x}$, their mean and variance are also related to the PC coefficients $\boldsymbol{X}$. In general, the path constraint g_j and performance function J are composed of or can be transformed into some basic common functions with respect to the stochastic state variables $\boldsymbol{x}$, such as addition and subtraction, product, reciprocal, exponential, and trigonometric functions. Therefore, the calculation of mean and variance of these basic functions is firstly introduced as follows.

1) Addition and subtraction functions $c = x_k \pm x_l$

$$c = \sum_{i=0}^{P} c_i\Phi_i = \sum_{i=0}^{P} x_{ki}\Phi_i \pm \sum_{i=0}^{P} x_{li}\Phi_i = \sum_{i=0}^{P}(x_{ki} \pm x_{li})\Phi_i \tag{14}$$

Then, the PC coefficients for c can be expressed as

$$c_i = x_{ki} \pm x_{li}, \quad i = 0, ..., P \tag{15}$$

2) Product function $c = x_k x_l$

$$c \approx \sum_{i=0}^{P} c_i\Phi_i = \left(\sum_{j=0}^{P} x_{kj}\Phi_j\right)\left(\sum_{s=0}^{P} x_{ls}\Phi_s\right) \tag{16}$$

Then, the PC coefficients for c can be expressed as

$$c_i = \sum_{j=0}^{P}\sum_{s=0}^{P} \frac{\langle\Phi_j\Phi_s\Phi_i\rangle}{\langle\Phi_i^2\rangle} x_{kj}x_{ls}, \quad i = 0, \ldots, P \tag{17}$$

3) Reciprocal function $c = 1/x_k$

$$c \approx \sum_{i=0}^{P} c_i\Phi_i = 1\Big/\left(\sum_{j=0}^{P} x_{kj}\Phi_j\right) \tag{18}$$

Then

$$\sum_{i=0}^{P}\sum_{j=0}^{P} c_i x_{kj}\Phi_i\Phi_j = 1 \tag{19}$$

Finally, the PC coefficients for c can be obtained by solving Eq. (19).

4) Exponential function $c = e^{x_k}$

Using the Taylor expansion, c can be expressed as

$$c \approx \sum_{i=0}^{P} c_i\Phi_i \approx e^{x_{k0}}\left(1 + \sum_{q=1}^{N_T} \frac{\left(\sum_{i=1}^{P} x_{ki}\Phi_i\right)^q}{q!}\right) \tag{20}$$

where N_T is the number of terms in the Taylor expansion.

Then, the PC coefficients for c can be obtained by using the Galerkin projection method as follows.

$$c_i = \frac{\langle e^{x_{k0}}\left(1 + \sum_{q=1}^{N_T} \frac{\left(\sum_{i=1}^{P} x_{ki}\Phi_i\right)^q}{q!}\right), \Phi_i\rangle}{\langle\Phi_i^2\rangle} \tag{21}$$

5) Sine function $c = \sin(x_k)$

$$\begin{aligned} c \approx \sum_{i=0}^{P} c_i\Phi_i &= \sin\left(x_{k0} + \sum_{i=1}^{P} x_{ki}\Phi_i\right) \\ &= \sin(x_{k0})\cos\left(\sum_{i=1}^{P} x_{ki}\Phi_i\right) \\ &\quad + \cos(x_{k0})\sin\left(\sum_{i=1}^{P} x_{ki}\Phi_i\right) \end{aligned} \tag{22}$$

Using the Taylor expansion one has

$$\sin\left(\sum_{i=0}^{P} x_{ki}\Phi_i\right) \approx \sum_{j=0}^{N_T}(-1)^{j-1}\frac{\left(\sum_{i=1}^{P} x_{ki}\Phi_i\right)^{2j+1}}{(2j+1)!} \tag{23}$$

$$\cos\left(\sum_{i=0}^{P} x_{ki}\Phi_i\right) \approx \sum_{j=0}^{N_T}(-1)^{j}\frac{\left(\sum_{i=1}^{P} x_{ki}\Phi_i\right)^{2j}}{(2j)!} \tag{24}$$

Through substituting Eqs. (23) and (24) into Eq. (22), and then using the Galerkin projection method on Eq. (22), the PC coefficients for c can be obtained.

It should be pointed out that if the performance function J contains an integral form, the PC model of the integrand should be firstly constructed using the way introduced above. Then the PC coefficients of the integral can be calculated as follows.

$$\begin{aligned} \int_{t_0}^{t_f} L\big(\boldsymbol{X}(t), \boldsymbol{u}(t), \boldsymbol{W}\big)\mathrm{d}t &= \int_{t_0}^{t_f}\left(\sum_{i=0}^{P} l_i\big(\boldsymbol{X}(t), \boldsymbol{u}(t)\big)\Phi_i\right)\mathrm{d}t \\ &= \sum_{i=0}^{P}\left(\int_{t_0}^{t_f} l_i\big(\boldsymbol{X}(t), \boldsymbol{u}(t)\big)\mathrm{d}t\right)\Phi_i \end{aligned} \tag{25}$$

$$c_i = \int_{t_0}^{t_f} l_i\big(\boldsymbol{X}(t), \boldsymbol{u}(t)\big)\mathrm{d}t \tag{26}$$

where $\int_{t_0}^{t_f} L(\boldsymbol{X}(t), \boldsymbol{u}(t), \boldsymbol{W})\mathrm{d}t$ is the integral form in the performance function J, $l_i(\boldsymbol{X}(t), \boldsymbol{u}(t))$ is the PC coefficient of the integrand $L(\boldsymbol{X}(t), \boldsymbol{u}(t), \boldsymbol{W})$, c_i is the PC coefficient of the integral.

Using the above method, the PC coefficients of the constraint g_j and performance function J can be obtained, and then their mean and standard deviation can be analytically calculated using Eqs. (12) and (13), which are also functions with respect to $\boldsymbol{X}$ and $\boldsymbol{u}$.

Step 6. Convexification of the non-convex constraints in Eqs. (9-c) and (9-e), and the non-convex performance function in Eq. (9-a)

In general, after UP by PC, the constraints in Eq. (9-c) and performance function in Eq. (9-a) are non-convex, which can be converted into linear functions using the successive linearization method [35]. At the reference trajectory $\{\boldsymbol{X}_r, \boldsymbol{u}_r\}$, the first-order Taylor expansion of Eq. (9-c) is used:

$$\begin{aligned} &g_{j\mu}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) + k^* g_{j\sigma}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{W}) \\ &\approx g_{j\mu}(\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}) + k^* g_{j\sigma}(\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}) \\ &\quad + \left(\frac{\partial g_{j\mu}}{\partial \boldsymbol{X}} + k^*\frac{\partial g_{j\sigma}}{\partial \boldsymbol{X}}\right)\Big|_{\boldsymbol{X}_r, \boldsymbol{u}_r, \boldsymbol{W}} (\boldsymbol{X} - \boldsymbol{X}_r) \end{aligned}$$

$$+\left(\frac{\partial g_{j\mu}}{\partial \boldsymbol{u}}+k^*\frac{\partial g_{j\sigma}}{\partial \boldsymbol{u}}\right)\Big|_{\boldsymbol{X}_r,\boldsymbol{u}_r,\boldsymbol{W}}(\boldsymbol{u}-\boldsymbol{u}_r)\le 0 \tag{27}$$

where $\frac{\partial g_{j\mu}}{\partial \boldsymbol{X}}$, $\frac{\partial g_{j\sigma}}{\partial \boldsymbol{X}}$, $\frac{\partial g_{j\mu}}{\partial \boldsymbol{u}}$, and $\frac{\partial g_{j\sigma}}{\partial \boldsymbol{u}}$ are the partial derivatives with respect to $\boldsymbol{X}$ and $\boldsymbol{u}$.

Similarly, the performance function Eq. (9-a) that is non-convex with respect to $\boldsymbol{X}$ and $\boldsymbol{u}$ can also be linearized as

$$\begin{aligned}&J_\mu(\boldsymbol{X},\boldsymbol{u},\boldsymbol{W})+k^*J_\sigma(\boldsymbol{X},\boldsymbol{u},\boldsymbol{W})\\&=J_\mu(\boldsymbol{X}_r,\boldsymbol{u}_r,\boldsymbol{W})+k^*J_\sigma(\boldsymbol{X}_r,\boldsymbol{u}_r,\boldsymbol{W})\\&\quad+\left(\frac{\partial J_\mu}{\partial \boldsymbol{X}}+k^*\frac{\partial J_\sigma}{\partial \boldsymbol{X}}\right)\Big|_{\boldsymbol{X}_r,\boldsymbol{u}_r,\boldsymbol{W}}(\boldsymbol{X}-\boldsymbol{X}_r)\\&\quad+\left(\frac{\partial J_\mu}{\partial \boldsymbol{u}}+k^*\frac{\partial J_\sigma}{\partial \boldsymbol{u}}\right)\Big|_{\boldsymbol{X}_r,\boldsymbol{u}_r,\boldsymbol{W}}(\boldsymbol{u}-\boldsymbol{u}_r)\end{aligned} \tag{28}$$

For the non-convex constraint in Eq. (9-e), the variance of the stochastic state variable x_i can be expressed as

$$x_{i\sigma^2}=\boldsymbol{Y}_i^{\mathrm{T}}\boldsymbol{Q}\,\boldsymbol{Y}_i \tag{29}$$

where $\boldsymbol{Y}_i=[\boldsymbol{X}((i-1)(P+1)+2),\boldsymbol{X}((i-1)(P+1)+3),\ldots,\boldsymbol{X}(i(P+1))]^{\mathrm{T}}$ and $\boldsymbol{Q}\in \mathrm{R}^{P\times P}$ is a diagonal matrix with elements as $\langle\Phi_j^2\rangle$ $(j=1,2,\ldots,P)$.

As $\langle\Phi_j^2\rangle>0$, $\boldsymbol{Q}$ can be expressed as follows using the Cholesky decomposition:

$$\boldsymbol{Q}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}\left(\boldsymbol{L}\in\mathbb{R}^{P\times P}\right) \tag{30}$$

Then, the standard deviation of the stochastic state variable x_i can be expressed as

$$x_{i\sigma}=\sqrt{\boldsymbol{Y}_i^{\mathrm{T}}\boldsymbol{Q}\,\boldsymbol{Y}_i}=\|\boldsymbol{L}\boldsymbol{Y}_i\| \tag{31}$$

By substituting Eq. (31) into Eq. (9-e), the non-convex constraint Eq. (9-e) is converted into the classic second-order cone constraint shown in Eq. (32), which can be directly handled by the convex optimization technique:

$$\|\boldsymbol{L}\boldsymbol{Y}_i\|\le-\frac{1}{k^*}x_{i0}+\frac{1}{k^*}x_i^{UB},\qquad \|\boldsymbol{L}\boldsymbol{Y}_i\|\le\frac{1}{k^*}x_{i0}-\frac{1}{k^*}x_i^{LB} \tag{32}$$

At this point, the equivalent deterministic trajectory optimization problem P^{ED} has been transformed into a classic second-order cone programming formulation $P^{\mathrm{ED\text{-}SOCP}}$ as follows.

Problem $P^{\mathrm{ED\text{-}SOCP}}$

$$\begin{cases}\text{find} & \boldsymbol{u}(t)\\ \min & \text{Eq. (28)}\\ \text{s.t.} & \text{Eq. (10)}\\ & \text{inequalities (11)}\\ & \text{inequalities (27)}\\ & \text{inequalities (32)}\\ & \text{Eqs. (9-f), (9-g)}\\ & \text{inequalities (9-h)}\\ & \boldsymbol{u}^{\mathrm{LB}}\le\boldsymbol{u}\le\boldsymbol{u}^{\mathrm{UB}}\end{cases} \tag{33}$$

Step 7. Discretize the above continuous optimal control problem $P^{\mathrm{ED\text{-}SOCP}}$ and solve it by the successive convex optimization technique [35].

Discretize the linearized the dynamic equations (Eq. (10)) with the sampling period T with the classic trapezoidal rule as follows. Other method such the Euler method can also be employed for discretization.

$$\begin{aligned}\boldsymbol{X}(k+1)=\boldsymbol{X}(k)+\frac{T}{2}\big[&\big(\boldsymbol{A}(k)\boldsymbol{X}(k)+\boldsymbol{B}(k)\boldsymbol{u}(k)+\boldsymbol{C}(k)\big)\\&+\big(\boldsymbol{A}(k+1)\boldsymbol{X}(k+1)+\boldsymbol{B}(k+1)\boldsymbol{u}(k+1)\\&+\boldsymbol{C}(k+1)\big)\big],\quad k=1,2\ldots,K\end{aligned} \tag{34}$$

where $\boldsymbol{A}(k)=\boldsymbol{A}_1(\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W})$, $\boldsymbol{B}(k)=\boldsymbol{B}_1(\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W})$, $\boldsymbol{C}(k)=\boldsymbol{C}_1(\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W})$, $T=(t_f-t_0)/K$ and K represents the total steps of discretization.

The performance function can be discreted as the summation of the performance values at the discrete nodes.

$$\begin{aligned}J=\sum_{k=1}^{K+1}\big[&J_\mu(k)+k^*J_\sigma(k)+J_{\boldsymbol{X}}(k)\big(\boldsymbol{X}(k)-\boldsymbol{X}_r(k)\big)\\&+J_{\boldsymbol{u}}(k)\big(\boldsymbol{u}(k)-\boldsymbol{u}_r(k)\big)\big]\end{aligned} \tag{35}$$

where $J_\mu(k)=J_\mu(\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W})$, $J_\sigma(k)=J_\sigma(\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W})$, $J_{\boldsymbol{X}}(k)=(\frac{\partial J_\mu}{\partial \boldsymbol{X}}+k^*\frac{\partial J_\sigma}{\partial \boldsymbol{X}})|_{\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W}}$ and $J_{\boldsymbol{u}}(k)=(\frac{\partial J_\mu}{\partial \boldsymbol{u}}+k^*\frac{\partial J_\sigma}{\partial \boldsymbol{u}})|_{\boldsymbol{X}_r(k),\boldsymbol{u}_r(k),\boldsymbol{W}}$.

The constraints shown in inequalities (27) and (32) and the constraint on control $\boldsymbol{u}^{LB}\le\boldsymbol{u}\le\boldsymbol{u}^{UB}$ are discreted to be satisfied at each discrete nodes, respectively; while the constraints shown in Eqs. (9-f) and (9-g), and inequalities (9-h) are discreted to be satisfied at the first $(k=1)$ or last $(k=K+1)$ discrete node.

Based on Steps 1–7, $P^{\mathrm{ED\text{-}SOCP}}$ is transformed into a parametric convex optimization problem, which can be easily and efficiently solved by the classic interior point method [29]. With the iteration of the convex optimization, the reference trajectory $\{\boldsymbol{X}_r,\boldsymbol{u}_r\}$ will be updated by the current obtained optimal trajectory till convergence condition ($\|\boldsymbol{u}-\boldsymbol{u}_r\|\le\varepsilon_u$ and ε_u is a small constant) is satisfied.

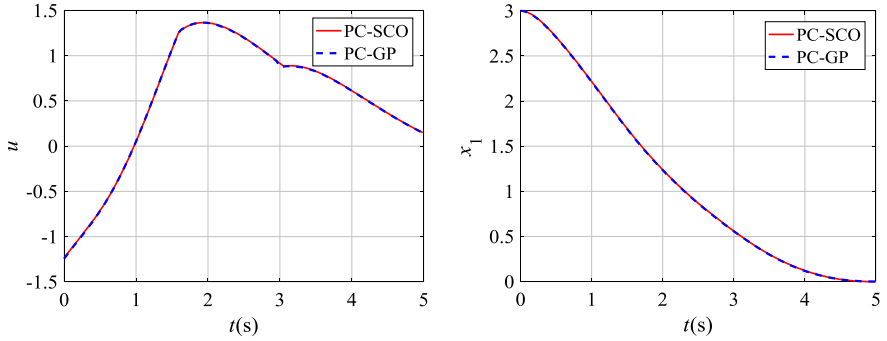
4. Comparative studies

In this section, the Van Der Pol oscillator problem and an engineering Mars entry problem are used to verify the effectiveness of the proposed robust trajectory optimization procedure combining PC with successive convex optimization (denoted as PC-SCO) in this paper. The Gaussian pseudo-spectral method in conjunction with PC (denoted as PC-GP) is employed for comparison. MOSEK [36] called by the YALMIP toolkit [37] in Matlab is used to solve the convex optimization in the study. For the Gaussian pseudo-spectral method, the GPOPS software is employed for implementation [38]. All the simulations are performed on a personal computer with Intel Core i7-4770 3.40 GHz processor and 16 GB of RAM.

During robust optimization, the user-defined parameter k^* is set as $k^*=3$. Meanwhile, the deterministic trajectory optimization without considering any uncertainness (denoted as DO) is also conducted for comparison to verify the advantage of robust optimization. MCS (1000 runs) is also employed to obtain the confirmed results by substituting the optimal control variables from RO and DO into the ODEs considering the same uncertainties.

4.1. Van Der Pol oscillator problem

The deterministic optimal control formulation of this example is



(a) Optimal control variables (b) Optimal state x_1

(c) Optimal state variable x_2 (d) Constraint

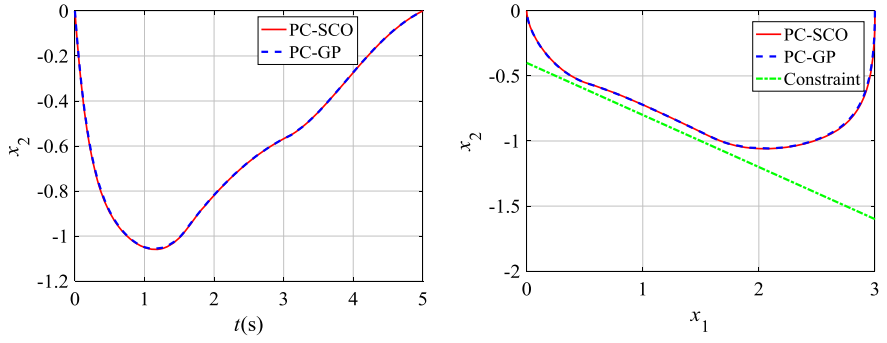


Fig. 1. Results of robust trajectory optimization ($d = 1$).

$$
\begin{cases}
\min \quad J = \int_{t_0}^{t_f} (x_1^2 + x_2^2 + u^2)dt \\
\text{s.t.} \quad \dot{x}_1 = x_2 \\
\qquad \dot{x}_2 = -x_1 + a(1 - x_1^2)x_2 + u, \quad a = 0.5 \\
\qquad g(x) = 0.4x_1 + x_2 + 0.4 \geq 0 \\
\qquad \boldsymbol{x}_0 = [3; 0], \quad \boldsymbol{x}_f = [0; 0] \\
\qquad [-5; -5] \leq \boldsymbol{x} \leq [5; 5] \\
\qquad -5 \leq u \leq 5 \\
\qquad t_0 = 0, \quad t_f = 5
\end{cases} \tag{36}
$$

Robust trajectory optimization is performed considering uncertainties from the parameter a, which follows uniform distribution with ± 0.01 variation around its nominal values. The PC order is set as $p = 2$ and thus the number of PC coefficients for each random variable is $P + 1 = \frac{(1+2)!}{1!2!} = 3$. With the consideration of uncertainties, each stochastic dynamic equation is expanded into 3 coupled deterministic equations in a higher-dimensional state space. Meanwhile, the constraints on the states are converted to $g(x)_\mu - 3g(x)_\sigma \geq 0$.

The optimal control variables and states of robust trajectory optimization obtained by PC-SCO and PC-GP are illustrated in Fig. 1, from which it is observed that both approaches can generate almost the same results and all the states lie within the constrained boundaries. Meanwhile, it is noticed from Fig. 1(d) that the optimal trajectory obtained by robust optimization is clearly deviated from the boundaries and has certain satisfaction margin under uncertainties.

The evolutions of the confirmed trajectory with MCS (1000 runs) for RO and DO are shown in Fig. 2. It is observed that for RO, the trajectory clearly lies within the boundary; while for DO, it is clearly beyond the boundary. These results further demonstrate the effectiveness and advantage of the proposed robust trajectory optimization procedure.

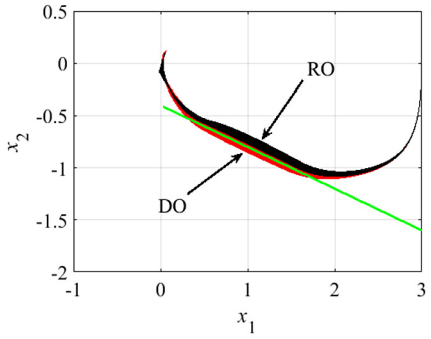
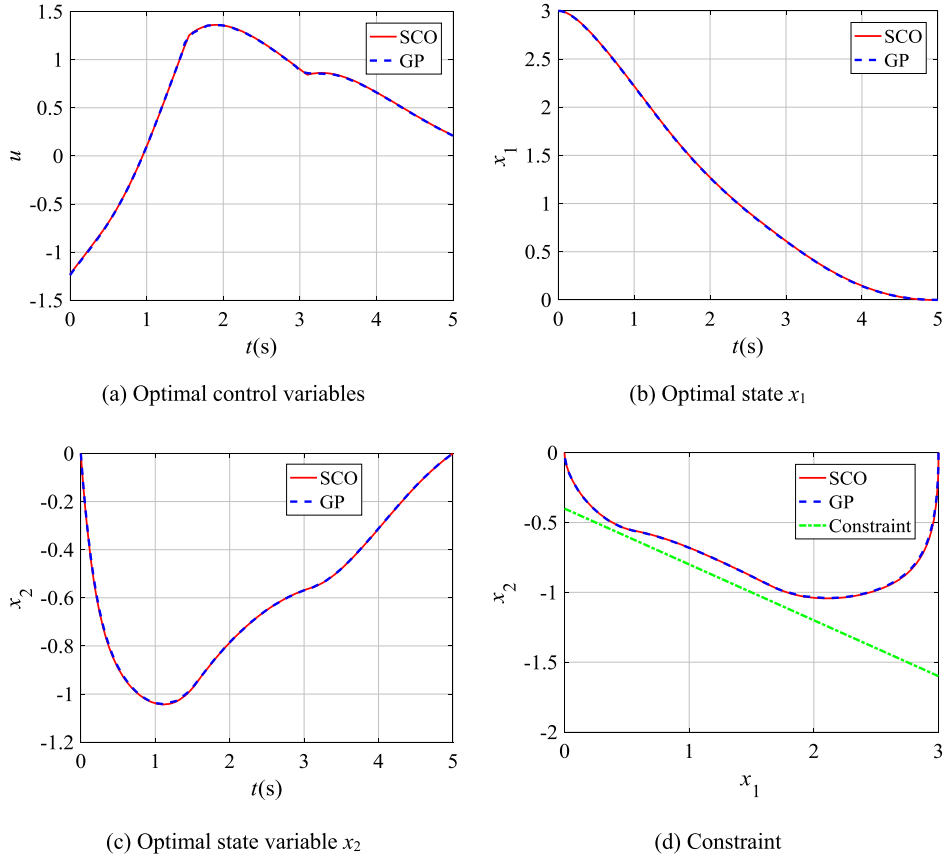


Fig. 2. Evolution of trajectory with MCS ($d = 1$).

Further, the robust trajectory optimization is done considering uncertainties from the parameter a and all the initial states $\boldsymbol{x}_0$ simultaneously. It is assumed that a follows uniform distribution with ± 0.01 variation, the initial states of x_1 and x_2 follow normal distributions with standard deviation of 0.02. Accordingly, each stochastic dynamic equation is expanded into 10 coupled deterministic equations.

The optimal control variables and states of robust trajectory optimization obtained by PC-SCO and PC-GP are illustrated in Fig. 3. Similar to the results with $d = 1$, it is noticed that all the states lie within the constrained boundaries and the optimal trajectory obtained by robust optimization is clearly deviated from the boundaries with certain satisfaction margin under uncertainties.

Similarly, the evolutions of confirmed trajectory with MCS (1000 runs) for RO and DO are shown in Fig. 4. It is observed that for RO, the trajectory lies within the boundary; while for DO, it is beyond the boundary. These results further demonstrate the effectiveness and advantage of the proposed robust trajectory optimization procedure.



(a) Optimal control variables

(b) Optimal state x_1

(c) Optimal state variable x_2

(d) Constraint

Fig. 3. Results of robust trajectory optimization for example 1 ($d = 3$).

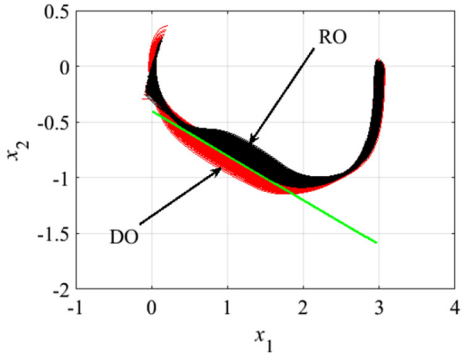


Fig. 4. Evolution of trajectory with MCS ($d = 3$).

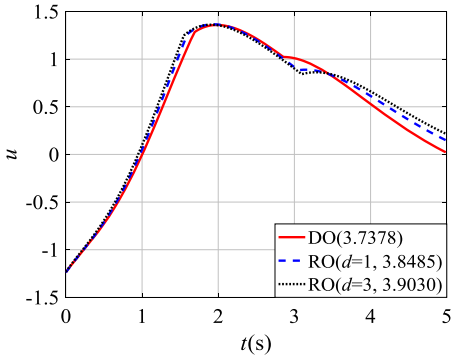


Fig. 5. Optimal control variables.

Table 1
Time cost and iteration number.

Methods	PC-SCO		PC-GP	
	T_{cost} (s)	D_{design}	T_{cost} (s)	D_{design}
Deterministic	0.48	1200	0.55	267
Robust ($d = 1$)	1.04	2800	4.53	623
Robust ($d = 3$)	1.22	8400	30.68	1533

In addition, the control variables for robust optimization with $d = 1$ and $d = 3$, as well as deterministic optimization are illustrated in Fig. 5. The control effort during the whole time span is also calculated by integrating $u^2(t)$ with respect to time t, of which the values are also shown in Fig. 5. From Fig. 5, it is noticed that the magnitude of the control variables is increased with the consideration of uncertainties, which is also increased with the increase of d. Meanwhile, it is found that the control effort with $d = 3$ (3.9030) is only slightly increased compared to that with $d = 1$ (3.8485). The reason is that the impact of uncertainties from the initial states on the optimal solution is small for this example. These results are consistent with the scenario of robust optimization that the robustness is achieved at the expense of certain performance.

The computational time T_{cost} and the number of design variables D_{design} during the deterministic and robust optimization processes are listed in Table 1. The comparison between PC-SCO and PC-GP is made in the case that the two approaches yield similar accuracy of optimal solution. For PC-GP, the Gaussian pseudospectral method implemented in the GPOPS-II software is used for solving the optimal control problem, and the discrete points are adaptively allocated according to the convergence tolerance and initial setting. For PC-SCO, as the highly efficient convex opti-

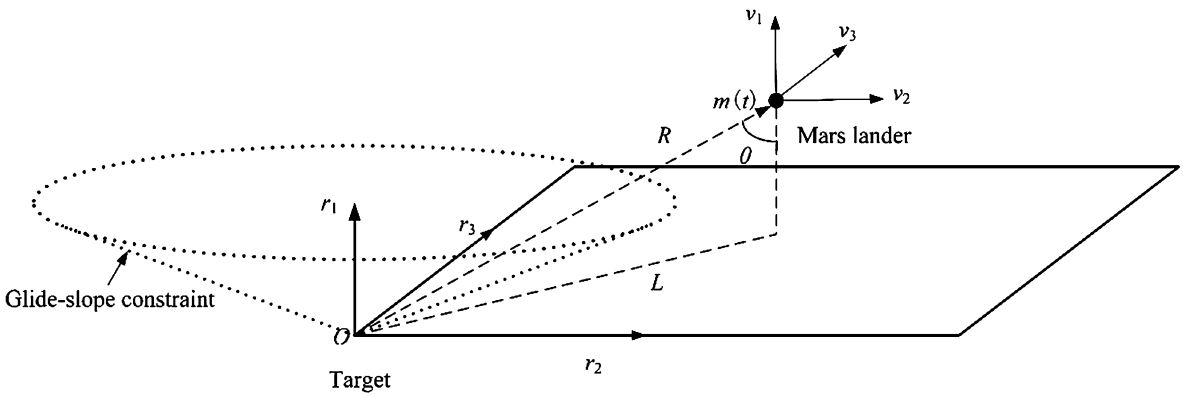
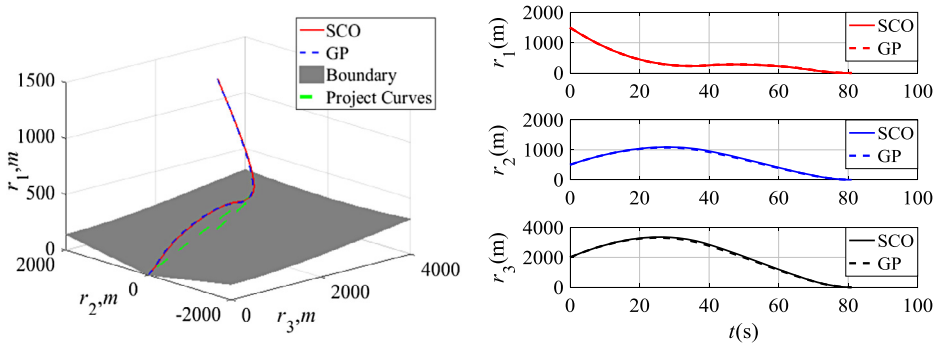


Fig. 6. Surface fixed coordinate frame.



(a) Optimal trajectory (b) State variable $\boldsymbol{r}$

Fig. 7. Optimal results of deterministic optimization.

mization method is used, a large number of uniformly distributed discrete points is set to ensure high accuracy as is commonly done in practice. Therefore, D_{design} of PC-SCO is much larger than that of PC-GP. It is noticed that for DO, the SCO method (0.48 s) is slightly more efficient than the GP technique (0.55 s). However, when uncertainties are considered, the computational cost for PC-GP is increased significantly, which is about 8.24 times ($d = 1$) and 61.89 times ($d = 3$) of that taken by the deterministic optimization; while for PC-SCO, it is increased to 2.17 ($d = 1$) and 2.54 times ($d = 3$), respectively. Clearly, PC-SCO is much more efficient than PC-GP for robust trajectory optimization. The reason is that with the consideration of uncertainties, the dimension of the expanded ODEs is greatly increased, resulting in significant increase of nodes (*i.e.*, design variables) in the transcribed NLP for the PC-GP method, so as the computational cost. However, for PC-SCO, as the optimal control is ultimately transcribed into a convex optimization problem, it is can be efficiently solved even for large-scale problem. Thus, even though D_{design} of PC-SCO is larger than PC-GP, PC-SCO is still much more efficient. These results demonstrate the high efficiency of the proposed robust trajectory optimization procedure.

4.2. Robust trajectory optimization of Mars pinpoint landing

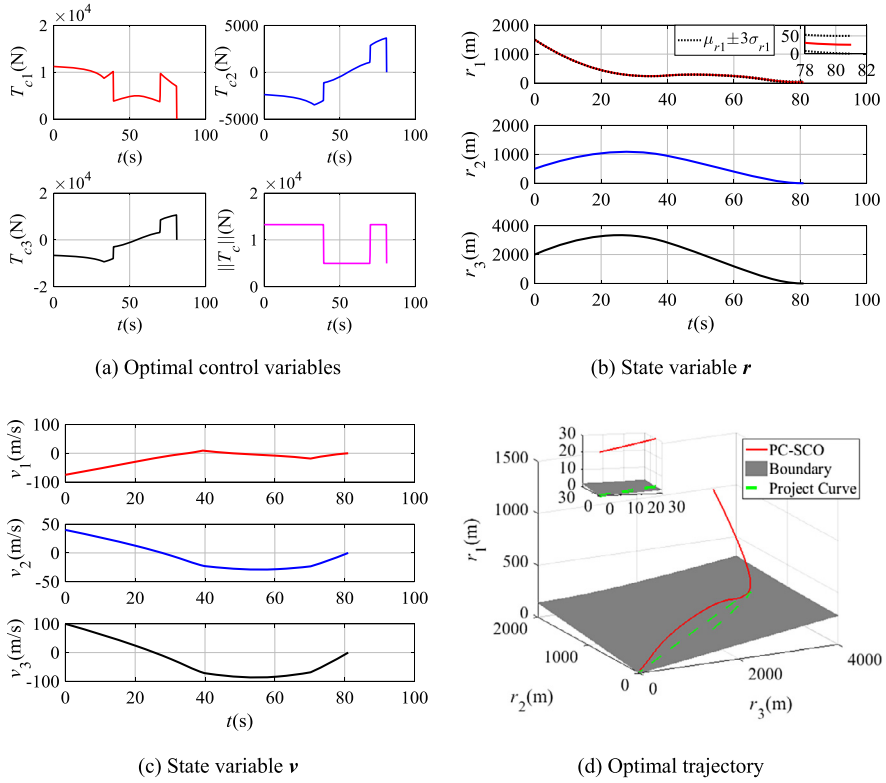
The Mars landing problem can be described as a trajectory optimization problem for a generic spacecraft performing pinpoint landing on a planet with uniform gravitational field. The deterministic optimal control formulation of the Mars landing problem adopted from [31] is shown as follows with the surface fixed coordinate frame shown in Fig. 6.

$$
\begin{cases}
\text{find} & \boldsymbol{T_c}(t) \\
\min & J = -m(t_f) \\
\text{s.t.} & \dot{\boldsymbol{r}} = \boldsymbol{v} \\
& \dot{\boldsymbol{v}} = \dfrac{\boldsymbol{T_c}}{m} - \boldsymbol{g} \\
& \dot{m} = -\alpha \|\boldsymbol{T_c}\| \\
& \boldsymbol{r}(t_0) = [1500;\, 500;\, 2000]\ \text{m} \\
& \boldsymbol{v}(t_0) = [-75;\, 40;\, 100]\ \text{m/s} \\
& r_2(t_f) = 0\ \text{m},\ r_3(t_f) = 0\ \text{m} \\
& \boldsymbol{v}(t_f) = [0;\, 0;\, 0]\ \text{m/s} \\
& m(t_0) = 1905\ \text{kg} \\
& 0.3T_{\max} \le \|\boldsymbol{T_c}\| \le 0.8T_{\max} \\
& \dfrac{\sqrt{r_2^2 + r_3^2}}{r_1} \le \tan(86^\circ) \\
& r_1(t) \ge 0\ \text{m},\ t_0 = 0\ \text{s},\ t_f = 81\ \text{s}
\end{cases}, \tag{37}
$$

where $\boldsymbol{r} = [r_1, r_2, r_3] \in \mathbb{R}^3$ is the position vector relative to the target, $\boldsymbol{v} = [v_1, v_2, v_3] \in \mathbb{R}^3$ is the velocity vector, $\boldsymbol{g} = [-3.7114\ \text{m/s}^2, 0, 0]^{\mathrm{T}}$ is the constant gravitational acceleration vector of the planet, $\boldsymbol{T_c} \in \mathbb{R}^3$ is the net thrust vector, m is the spacecraft mass, and $\alpha = 5.09 \times 10^{-4}$ s/m is a positive constant describing the fuel consumption rate.

In (37), $r_1(t) \ge 0$ ensures that the trajectory does not go below the surface during the maneuver and $\sqrt{r_2^2 + r_3^2}/r \le \tan(86^\circ)$ represents another state constraint exerted on the "altitude angle" or "glide slope", which is used to prevent the trajectory from descending at an angle shallower than 4° and ensure that the target can be observed in the rugged surface of Mars.

For deterministic trajectory optimization, the results generated by SCO and GP are shown in Fig. 7. It is observed that the two approaches can obtain very similar results. However, it is also noted that the optimal trajectory is right on the boundary in some re-



(a) Optimal control variables (b) State variable $\boldsymbol{r}$

(c) State variable $\boldsymbol{v}$ (d) Optimal trajectory

Fig. 8. Optimal control and states of PC-SCO ($d = 4$).

gions around the initial and middle stages, and so is $\boldsymbol{r}(t_f)$, exhibiting a very small satisfactory margin. Therefore, when uncertainties exist, the flight trajectory and r_1 would probably exceed the boundary and violate the constraint, inducing mission failure.

Robust trajectory optimization is performed considering uncertainties from the initial states $r_1(t_0)$, $r_2(t_0)$, $r_3(t_0)$ induced by navigation errors and the parameter α. It is assumed $r_1(t_0)$, $r_2(t_0)$, $r_3(t_0)$ all follow uniform distributions with ± 2 m variations around their nominal values, and α follow uniform distributions with $\pm 1\%$ variations around its nominal value. The PC order is set as $p = 2$, and thus the number of PC coefficients for each random state is $P + 1 = (4 + 2)!/(4!2!) = 15$. Accordingly, the original stochastic ODEs are expanded into 15×7 coupled deterministic equations in a higher-dimensional state space.

The optimal results obtained from the proposed PC-SCO method is shown in Fig. 8. However, for the PC-GP approach, the optimal results cannot be obtained owing to the intensive computational cost and memory consumption, and there was no result after more than 12 hours of optimization in this work. Fig. 8 shows that all the constrained states as well as the trajectory generated by PC-SCO lie within the boundaries with certain evident gaps from the boundaries, which yield relatively large satisfactory margins.

MCS with 1000 runs is conducted to obtain the confirmed results using the optimal control variables obtained from PC-SCO, and deterministic trajectory optimization. The optimal trajectory and r_1 are plotted in Figs. 9 and 10, respectively. It is shown that r_1 and the trajectory generated by the deterministic trajectory optimization violate the constraint. However, for the robust trajectory optimization using PC-SCO, they both lie within the boundaries, exhibiting good robustness with respect to the uncertainties. These results further demonstrate the effectiveness of the proposed robust trajectory optimization procedure.

Further, robust trajectory optimization is performed considering uncertainties from all the initial states $\boldsymbol{x}_0 = [r_1(t_0), r_2(t_0), r_3(t_0), v_1(t_0), v_2(t_0), v_3(t_0), m(t_0)]$ and the parameter α. It is assumed

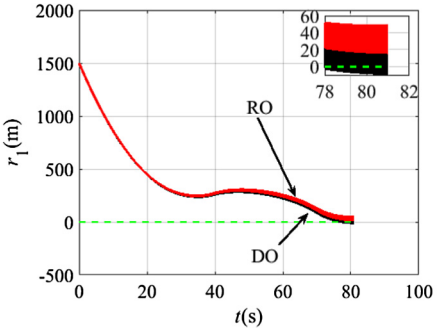


Fig. 9. State r_1 obtained by MCS ($d = 4$).

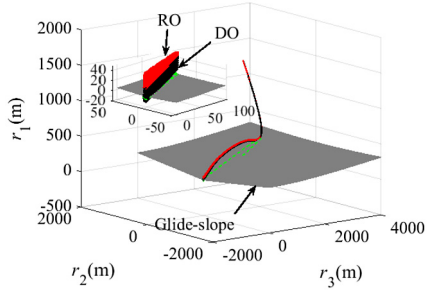


Fig. 10. Trajectories obtained by MCS ($d = 4$).

that α and $\boldsymbol{x}_0$ follow uniform distributions with $\pm 1\%$ and ± 2 (m, m/s, or kg) variations, respectively, around their nominal values. The PC order is also set as $p = 2$, and accordingly the original stochastic ODEs are expanded into 45×7 coupled deterministic equations in a higher-dimensional state space.

The optimal results by the proposed PC-SCO method are shown in Fig. 11. However, for the PC-GP approach, the optimal results

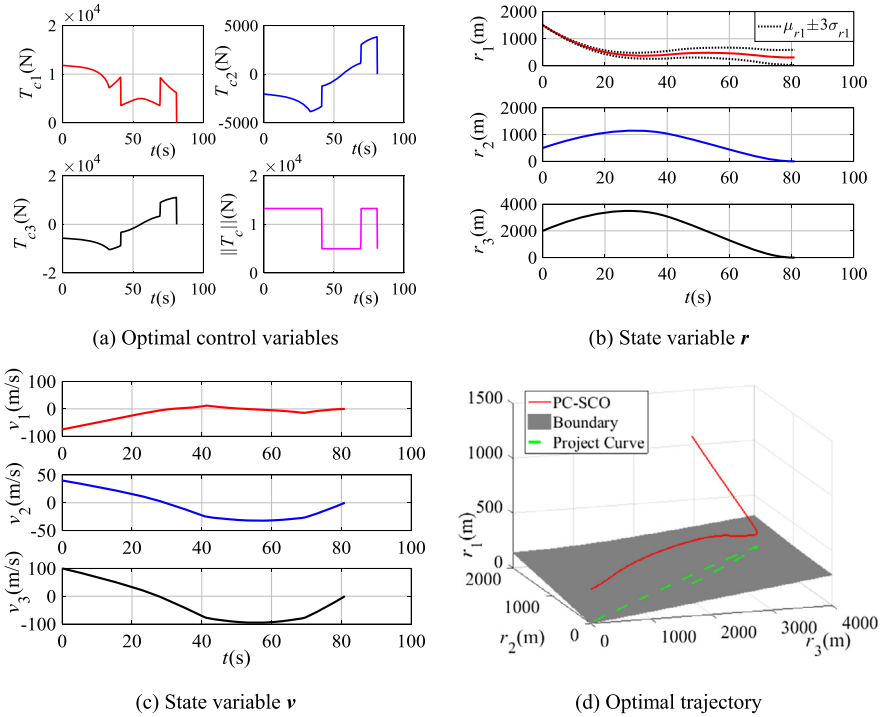


Fig. 11. Optimal control and states of PC-SCO ($d = 8$).

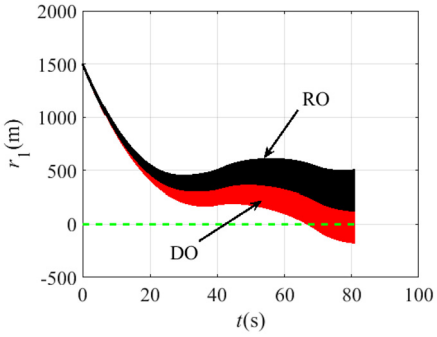


Fig. 12. State r_1 obtained by MCS ($d = 8$).

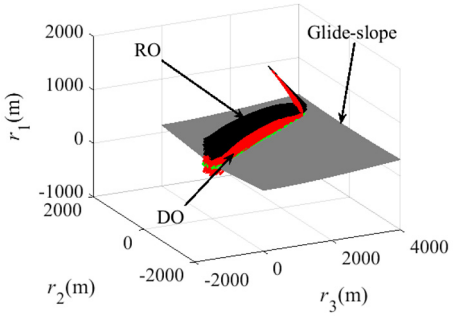


Fig. 13. Trajectories obtained by MCS ($d = 8$).

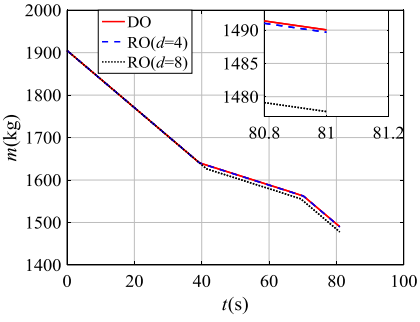


Fig. 14. Comparison of m.

also cannot be obtained after more than 12 hours of optimization in this work. Figs. 12 and 13 illustrate the confirmed results with MCS for both PC-SCO and DO. These results show that the PC-SCO exhibits good robustness to uncertainties and all the constraints are satisfied with certain margin, while the constraints of DO are violated. These results further demonstrate the effectiveness of the proposed robust trajectory optimization procedure.

The performance mass m and the 2-norm of control $\|\boldsymbol{T}_c\|$ (indicating the control effort) for deterministic optimization and the two cases of robust optimization are compared, and the results are shown in Figs. 14 and 15, respectively. It is found that the final mass $m(t_f)$ of RO ($d = 4$) is slightly reduced compared with that of DO, while it is evidently reduced for RO with higher random dimension ($d = 8$). Meanwhile, it is noted that the time that $\|\boldsymbol{T}_c\|$ lies in the upper limit value for RO with $d = 8$ is longer than that of DO, indicating that more control effort is required. The robustness of this problem is achieved at the expense of more control effort and smaller final mass, which is consistent with the scenario of robust optimization.

The computational times T_{cost} and the number of the design variables D_{design} during optimization for both RO and DO are listed in Table 2. It is shown that for the deterministic trajectory optimization, the SCO method (0.38 s) is much more efficient than the GP technique (10.52 s). Moreover, when uncertainties are considered, the computational cost for PC-GP is increased significantly and results cannot even be obtained after about 12 hours of computation. The dynamics of this example is more nonlinear and higher-dimensional compared to the above example, and thus the transcribed NLP problem by PC-GP is large scale and very difficult to solve. However, for PC-SCO, owing to the employment of SCO, the computational time is slowly increased to 17.18 s even with

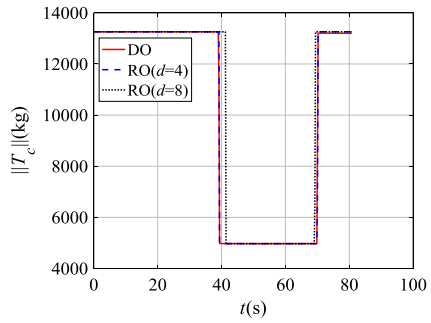


Fig. 15. Comparison of $\|\boldsymbol{T}_c\|$.

Table 2
Time cost and iteration number.

Methods	PC-SCO		PC-GP	
	T_{cost} (s)	D_{design}	T_{cost} (s)	D_{design}
Deterministic	0.38	3200	10.52	368
Robust ($d=4$)	1.30	42400	>12 hours	/
Robust ($d=8$)	17.18	126400	>12 hours	/

a large number of design variables, which is much more efficient than PC-GP.

5. Conclusions

To address the issue that the computational cost and memory consumption increase significantly with the increase in the dimension of uncertain factors, and the nonlinearity of dynamic equations in the existing robust trajectory optimization method using PC and the direct method, a new robust trajectory optimization procedure using PC in conjunction with the convex optimization technique has been proposed to improve the efficiency of trajectory generation. Firstly, PC is employed to transform the original stochastic optimal control into a deterministic version in a higher-dimensional state space. Then, a series of convexification techniques are introduced to convert the non-convex dynamics, constraints, and objective into convex counterparts, which are then solved by the successive convex optimization technique. Comparative studies on the Van Der Pol oscillator and a Mars landing problem show that the proposed procedure can obtain effective and highly efficient optimal results that are robust to uncertainties even with high-dimensional random factors. Compared with the existing method, the computational efficiency of the proposed method is greatly improved and no significant increase in the computational cost is observed with the increase in dimensions of uncertain factors. The results demonstrate the effectiveness and applicability of the proposed procedure.

Declaration of Competing Interest

There is no competing interest.

Acknowledgements

Grant support from Science Challenge Project (No. TZ2018001) and Hongjian Innovation foundation (No. BQ203-HYJJ-Q2018002) are gratefully acknowledged.

References

[1] J.T. Betts, Survey of numerical methods for trajectory optimization, J. Guid. Control Dyn. 21 (2) (1998) 193–207, https://doi.org/10.2514/2.4231.

[2] C.L. Darby, W.W. Hager, A.V. Rao, An hp-adaptive pseudospectral method for solving optimal control problems, Optim. Control Appl. Methods 32 (4) (2011) 476–502, https://doi.org/10.1002/oca.957.

[3] C.L. Darby, A.V. Rao, Minimum-fuel low Earth-orbit aero-assisted orbital transfer of small spacecraft, J. Spacecr. Rockets 48 (4) (2011) 618–628, https://doi.org/10.2514/1.A32011.

[4] C.L. Darby, Hp-Pseudospectral Method for Solving Continuous-Time Nonlinear Optimal Control Problems, Ph.D. Dissertation, Dept. of Aerospace Engineering, University of Florida, Gainesville, Florida, 2011.

[5] P. Han, J. Shan, X. Meng, Re-entry trajectory optimization using an hp-adaptive Radau pseudospectral method, J. Aerosp. Eng. 227 (10) (2013) 1623–1636, https://doi.org/10.1177/0954410012461745.

[6] H.J. Kelley, S.S. Mckay, S.S.M. Ii, et al., Rocket trajectory optimization by a second-order numerical technique, AIAA J. 7 (5) (2015) 879–884, https://doi.org/10.2514/3.5239.

[7] J.T. Betts, Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Soc. for Industrial and Applied Mathematics Press, Philadelphia, 2009, pp. 257–258.

[8] Z.P. Mourelatos, J. Liang, A methodology for trading off performance and robustness under uncertainty, J. Mech. Des. 128 (4) (2006) 856–863, https://doi.org/10.1115/1.2202883.

[9] Z. Yang, Y.Z. Luo, J. Zhang, Robust planning of nonlinear rendezvous with uncertainty, J. Guid. Control Dyn. 40 (8) (2017) 1954–1967, https://doi.org/10.2514/1.G002319.

[10] B.R. Saunders, Optimal Trajectory Optimization Under Uncertainty, M.S. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Massachusetts, 2012.

[11] G.J. Tang, Y.Z. Luo, H.Y. Li, Optimal robust linearized impulsive rendezvous, Aerosp. Sci. Technol. 11 (7–8) (2015) 563–569, https://doi.org/10.1016/j.ast.2007.04.001.

[12] T. Flanzer, G. Bower, I. Kroo, Robust trajectory optimization for dynamic soaring, in: AIAA Guidance, Navigation, and Control Conference, 13–16 August 2012, Minneapolis, Minnesota, 2012, AIAA 2012-4603.

[13] M. Gerdts, Direct shooting method for the numerical solution of higher-index DAE optimal control problems, J. Optim. Theory Appl. 117 (2) (2003) 267–294, https://doi.org/10.1023/A:1023679622905.

[14] B. Geiger, J. Horn, A. Delullo, et al., Optimal path planning of UAVs using direct collocation with nonlinear programming, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, Colorado, 2013.

[15] L. Zhang, H. Gao, Z. Chen, et al., Multi-objective global optimal parafoil homing trajectory optimization via Gauss pseudospectral method, Nonlinear Dyn. 72 (1–2) (2013) 1–8, https://doi.org/10.1007/s11071-012-0586-9.

[16] G.T. Huntington, A.V. Rao, Optimal reconfiguration of spacecraft formations using the Gauss pseudospectral method, J. Guid. Control Dyn. 31 (3) (2012) 689–698, https://doi.org/10.2514/1.31083.

[17] Q. Chen, Z. Wang, S. Chang, et al., Optimal trajectory optimization under uncertainty for a gliding guidance projectile, Acta Aeronaut. Astronaut. Sin. 35 (9) (2014) 2593–2604 (in Chinese).

[18] G. Terejanu, P. Singla, T. Singh, et al., Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models, J. Guid. Control Dyn. 31 (6) (2008) 1623–1633, https://doi.org/10.2514/1.36247.

[19] Y.Z. Luo, Z. Yang, A review of uncertainty propagation in orbital mechanics, Prog. Aerosp. Sci. 89 (2017) 23–39, https://doi.org/10.1016/j.paerosci.2016.12.002.

[20] S.H. Lee, W. Chen, A comparative study of uncertainty propagation methods for black-box type functions, Struct. Multidiscip. Optim. 37 (3) (2008) 239–253, https://doi.org/10.1007/s00158-008-0234-7.

[21] D. Xiu, G.E. Karniadakis, The Weiner–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644, https://doi.org/10.1137/S1064827501387826.

[22] F. Wang, F. Xiong, H. Jiang, et al., An enhanced data-driven polynomial chaos method for uncertainty propagation, Eng. Optim. 50 (2) (2018) 273–292, https://doi.org/10.1080/0305215X.2017.1323890.

[23] G.F. Ren, S.Y. Zhu, A new landing site uncertainty analysis method for Mars entry mission, J. Harbin Inst. Tech. 44 (7) (2012) 14–20 (in Chinese).

[24] A. Prabhakar, J. Fisher, R. Bhattacharya, Polynomial chaos-based analysis of probabilistic uncertainty in hypersonic flight dynamics, J. Guid. Control Dyn. 33 (1) (2010) 222–234, https://doi.org/10.2514/1.41551.

[25] J. Fisher, R. Bhattacharya, Optimal trajectory generation with probabilistic system uncertainty using polynomial chaos, J. Dyn. Syst. Meas. Control 133 (1) (2011) 014501, https://doi.org/10.1115/1.4002705.

[26] X. Li, P. Nair, Z. Zhang, et al., Aircraft robust trajectory optimization using nonintrusive polynomial chaos, AIAA J. 51 (5) (2014) 592–1603, https://doi.org/10.2514/1.C032474.

[27] F. Xiong, Y. Xiong, B. Xue, Trajectory optimization under uncertainty based on polynomial chaos expansion, in: AIAA Guidance, Navigation, and Control Conference, Florida, 2015.

[28] Z.S. Yu, Research on Design and Optimization Method of Autonomous Navigation Scheme for Mars Entry Phase, Ph.D dissertation, Chapter 4, Beijing Institute of Technology, 2015 (in Chinese).
[29] Y. Nesterov, A. Nemirovski, Interior-point polynomial algorithms in convex programming, J. Soc. Ind. Appl. Math. 13 (1994).
[30] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, New York, 2004.
[31] L. Blackmore, B. Acikmese, D.P. Scharf, Minimum-landing-error powered-descent guidance for Mars landing using convex optimization, J. Guid. Control Dyn. 33 (4) (2010) 1161–1171, https://doi.org/10.2514/1.47202.
[32] X. Liu, Z. Shen, P. Lu, Closed-loop optimization of guidance gain for constrained impact, J. Guid. Control Dyn. 40 (2) (2017) 453–460, https://doi.org/10.2514/1.G000323.
[33] P. Lu, X. Liu, Autonomous trajectory planning for rendezvous and proximity operations by conic optimization, J. Guid. Control Dyn. 36 (2) (2013) 375–389, https://doi.org/10.2514/1.58436.
[34] X. Liu, Autonomous Trajectory Planning by Convex Optimization, PhD Dissertation, Iowa State University, Iowa, 2013.
[35] X. Liu, P. Lu, B. Pan, Survey of convex optimization for aerospace applications, Astrodynamics 1 (1) (2017) 23–40, https://doi.org/10.1007/s42064-017-0003-8.
[36] E.D. Andersen, C. Roos, T. Terlaky, On implementing a primal-dual interior-point method for conic quadratic optimization, Math. Program. 95 (2) (2003) 249–277, https://doi.org/10.1007/s10107-002-0349-3.
[37] J. Lofberg, YALMIP: a toolbox for modeling and optimization in MATLAB, in: 2004 IEEE International Symposium on Computer Aided Control Systems Design, Taiwan, September 2–4, 2004.
[38] M.A. Patterson, A.V. Rao, GPOPS-II: a MATLAB software for solving multiple-phase optimal control problems using HP-adaptive gaussian quadrature collocation methods and sparse nonlinear programming, ACM Trans. Math. Softw. 41 (1) (2014) 1–37.