Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3676810, 19 pages
https://doi.org/10.1155/2021/3676810

Research Article

Robust Trajectory Planning for Hypersonic Glide Vehicle with Parametric Uncertainties

Chunyun Dong,[1] Zhi Guo,[2] and Xiaolong Chen[1]

[1] *School of Electro-mechanical Engineering, Xidian University, Xi'an, Shaanxi 710071, China*
[2] *Xi'an Satellite Control Center, Xi'an, Shaanxi 710043, China*

Correspondence should be addressed to Xiaolong Chen; xlchen@mail.xidian.edu.cn

Received 6 August 2020; Revised 19 November 2020; Accepted 20 December 2020; Published 16 January 2021

Academic Editor: Luis Payá

A hybrid double-loop optimization algorithm combing particle swarm optimization (PSO) and nonintrusive polynomial chaos (NIPC) is proposed for solving the robust trajectory optimization of hypersonic glide vehicle (HGV) under uncertainties. In the outer loop, the PSO method searches globally for the robust optimal control law according to a penalized fitness function that contains the system robustness considerations. In the inner loop, uncertainty propagation of the stochastic system is performed using the NIPC method, to provide statistical moments for the iterative scheme of the PSO method in the outer loop. Only control variables are discretized, and the state constraints are satisfied implicitly through the numerical integration process, which reduces the number of decision variables as well as the huge amount of computation increased by NIPC. In the end, the robust optimal control law is achieved conveniently. Numerical simulations are carried out considering a classical time-optimal trajectory optimization problem of HGV with uncertainties in both initial states and aerodynamic coefficients. The results demonstrate the feasibility and effectiveness of the proposed method.

1. Introduction

Hypersonic glide vehicle (HGV) is generally released from solid rocket boosters and then reentry glides through the atmosphere at hypersonic speed without power. Depending on the high lift-to-drag (L/D) ratio shape design and proper aerodynamic control techniques, it can achieve long flight distance and strong maneuverability. In recent years, HGV has attracted worldwide attention for its broad application prospects in both military and civilian fields [1–3].

However, the reentry region of HGV is quite narrow and always suffers from complex uncertainties, due to large space span, long flight time, changeable aerodynamic environment, and shortage of flight experience [4]. Therefore, the reentry trajectory design, as a core problem of the guidance and control, has become a challenge and hotspot for HGV [5, 6]. The performance of trajectory design is the essential section of a flight. The trajectory design framework of offline trajectory optimization and online tracking guidance is usually adopted in engineering, because of the limitation of the current vehicle-borne computing capacity and the high real-time calculation requirements [6, 7]. Under this framework, the reference trajectory is optimized offline and then stored in the onboard computer for the guidance and control system to track.

The reference trajectory optimization process affords an overall analysis of multidisciplinary designs and provides a theoretical basis for many other disciplines, such as guidance, control, defense penetration, and thermal protection. It is essentially a nonlinear optimal control problem with multiple constraints, including the control limitation, dynamic pressure, heating rate, and aerodynamic load. It is difficult to solve this problem analytically; thus, many numerical optimization algorithms have been developed and applied by now [8–12]. According to Betts' survey on numerical trajectory optimization methods, these algorithms can be divided into two categories [13]. The first one is the indirect method, in which the trajectory optimization problem is converted to a Hamiltonian boundary value problem using the variation method or Pontryagin's maximum principle [11]. These methods can obtain accurate analytical solutions but their initial states are difficult to guess. The other one is the direct methods, in which the optimal control problem is transformed into a nonlinear programming problem and then solved by some numerical methods. These methods are weakly dependent on their initial states and require no system derivative information. In the field of direct methods for trajectory optimization, the metaheuristic algorithms, especially the swarm intelligence algorithms, have received widespread attention [10]. However, the deterministic optimization methods lack robustness considerations for uncertainties and, therefore, will increase the design burden of the online tracking guidance system for the case of unpredictable fluctuations or deviations from the original reference trajectory [14].

To solve this problem, trajectories that are statistically less sensitive to uncertainties are preferred, and robust reference trajectory optimization is needed. The improvement of robust trajectory optimization over the conventional trajectory design is to integrate the independent design steps into an interactive whole, by means of building a reverse information flow between the offline trajectory optimization and the online reentry environment. Compared with some online trajectory optimization and tracking guidance techniques [15, 16], robust trajectory optimization displays obvious advantages in reducing the burden of the guidance and control system effectively.

Robust trajectory optimization of HGV is a complex stochastic process with multidisciplinary attributes such as flight system modeling, uncertainty analysis, and trajectory numerical solution. Since the deterministic optimization technology is relatively mature, the commonly used robust strategy is to transform the stochastic system into an equivalent deterministic system. Generally, this is achieved by adding appropriate statistical information (such as the mean and the standard deviation) of the user-defined qualities to the original objective function as well as the constraints. To this end, the efficient and accurate uncertainty propagation (UP) techniques [17] are necessary, which usually include Monte Carlo (MC) simulation, linear methods, and nonlinear methods. MC simulation relies on a large random sample space to approach the true value and is usually used to verify the effectiveness of the target algorithm. Janson et al. proposed an MC motion planning method for the robot trajectory optimization problem under uncertainty and proved its feasibility and effectiveness [18]. Linear methods simplify the uncertainty propagation analytically but are inapplicable for highly nonlinear hypersonic vehicles. As one of the popular nonlinear methods, polynomial chaos (PC) uses the sum of orthogonal polynomial chaos expansions to fit the probabilistic model output response. It is based on the spectral representation of uncertainties and can obtain the system's high-order statistical moments conveniently [19, 20]. The common implementation forms of PC include the generalized polynomial chaos (gPC) and the nonintrusive polynomial chaos (NIPC) that produce almost the same accuracy. However, the NIPC is more promising for it treats the system as a black box and requires no modification on the existing code [21].

In recent years, the PC method has been implemented to some aerospace engineering problems. Cottrill and Harmon [22] proved that the gPC algorithm displayed improved calculation efficiency while it maintained the same accuracy with Monte Carlo simulation. Subsequently, they proposed a robust trajectory optimization model that used the polynomial chaos method and the Gaussian pseudospectral method to transform the stochastic optimization problem into a set of similar deterministic optimization problems. Furtherly, Okamoto and Tsuchiya [23] combined the gPC method with the Legendre-Gaussian pseudospectral method to study the robust optimization of aircraft landing trajectories in a microburst environment. Boutselis et al. [24] combined the gPC method with the dynamic programming algorithm for the trajectory optimization of quadcopters with uncertain model parameters, and they adopted variational integrators to improve the performance of the proposed framework.

The robust optimization methods reviewed above are concise. However, when the target system is complicated, the repeated solution of the deterministic optimization problem will be unacceptable. Therefore, Fisher and Bhattacharya [25] transformed the stochastic optimization problem into an expanded deterministic optimization problem in the higher-dimensional state space and ensured the robustness considerations by adding optimization constraints. Afterward, Li et al. [26] conducted a preliminary study on the robust dynamic trajectory optimization of the aircraft for a small number of random variables. Huang et al. [27, 28] combined the NIPC with the most probable point search algorithm and the evidence theory, respectively, to study the robust trajectory optimization of Mars entry. Wang et al. [29] also addressed the Mars entry problem by combining the gPC with convex optimization techniques, which improved the efficiency and accuracy of the robust trajectory optimization. These methods avoid the repeated solution of corresponding deterministic optimization problems, but a larger number of constraints, especially the expanded ordinary differential equations, increase the difficulty of optimization convergence.

Consequently, how to imply the PC method to solve the complex HGV trajectory optimization problem with uncertainties efficiently is still a crucial problem. The key difficulty in the application of polynomial chaos is that it is easy to cause dimension disaster with the increase of random variables. As a result, the computational burden is unacceptable. The sparse tensor-product theory is a potential solution to this problem [30, 31]. However, if proper mathematical techniques can be used to simplify the application process of polynomial chaos in trajectory optimization, these difficulties will be alleviated in a wider scope.

In this paper, a stochastic trajectory optimization model of HGV is studied in the aspect of uncertainties in both initial states and aerodynamic coefficients. Specifically, it is assumed that the exact values of these uncertainties are unknown, but their probability density functions can be accessed. On this basis, the objective function and the constraint function are augmented to include the statistical moments of the system, and the dynamic robust optimization model is established. Then, the NIPC method is utilized to perform the uncertainty propagation over time for the statistical information and transform the original stochastic ordinary differential equations into deterministic ones. Afterward, an expanded deterministic optimal control problem with a stochastic flavor is obtained depending on the NIPC representation of the dynamics. Since the dimension of deterministic states is expanded, the control parameterization method is preferred in this paper for reducing the number of decision variables. Therefore, a novel version of the PSO method [32, 33] is incorporated to solve the obtained problem, and a hybrid double-loop optimization algorithm called NIPC-PSO is developed. In the proposed algorithm, the PSO particles which represent different control laws are iteratively optimized in the outer loop, based on their probabilistic behaviors that are calculated via NIPC in the inner loop. Moreover, the computing efficiency of the proposed framework is further improved by incorporating the concept of a numerical integration scheme for solving the complex dynamic equations.

The main contribution of this work lies in developing a concise formulation for robust reference trajectory optimization of HGV with uncertainty in dynamics. Only control variables are discretized, and the expanded state constraints are satisfied implicitly utilizing the benefits of the numerical integration scheme. The number of decision variables as well as the huge amount of computation for uncertainty propagation using NIPC is extremely reduced. These above properties of the proposed approach indicate its high efficiency and potential applicability for robust trajectory optimization problems of HGV.

The remainder of this paper is organized as follows. In Section 2, the robust trajectory optimization model is established. Then, the double-loop NIPC-PSO framework is described step by step in Section 3. In Section 4, a classical trajectory optimization problem of HGV with uncertainties is introduced. The numerical simulation examples are included in Section 5, and conclusions are drawn in Section 6.

2. Problem Formulation

The trajectory optimization is a typical nonlinear constrained optimal control problem. A commonly used optimal control formulation of the deterministic trajectory optimization P^D is as follows:

$$\text{Problem: } P^D$$
$$\begin{cases} \text{find} & \mathbf{U}(t) \\ \min & J = M\left(\mathbf{U}(t_0), \mathbf{X}(t_0), \mathbf{U}(t_f), \mathbf{X}(t_f), t_0, t_f\right) + \int_{t_0}^{t_f} L(\mathbf{U}(t), \mathbf{X}(t), t)\mathrm{d}t \quad \text{(a)} \\ \text{s.t.} & \dot{X}(t) = \mathbf{F}(\mathbf{U}(t), X(t), t) \quad \text{(b)} \\ & \mathbf{C}(\mathbf{U}(t), \mathbf{X}(t), t) \le \mathbf{0} \quad \text{(c)} \\ & \mathbf{B}\left(\mathbf{X}(t_0), t_0, \mathbf{X}(t_f), t_f\right) = \mathbf{0}, \quad \text{(d)} \end{cases} \tag{1}$$

where $\mathbf{U}(t)$ is the control variable vector and t is the time; J is the objective function, in which M is the cost of boundary constraints, and L is the cost of Lagrange (integral) term; t_0 and t_f denote the initial and final time, respectively; $\mathbf{X}(t)$ is the state variable vector and $\dot{X}(t) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t), t)$ is the system ordinary differential equations, i.e., ODEs; $\mathbf{C}$ and $\mathbf{B}$ are the path constraints and boundary constraints, respectively.

With the consideration of uncertainties, random variables obeying some probability distribution functions (PDFs) should be included [23]. In the present study, the control variables are assumed as deterministic functions of only t [23, 25], whereas the performance objective J, the ODEs, and the constraints C and B are all stochastic functions of the random variables. The goal of robust trajectory optimization is to obtain the optimal control law, which is insensitive to possible disturbances. Therefore, when uncertainty occurs, the expected trajectory dispersion would be minimized statistically.

To achieve the goal of robust trajectory optimization, the objective and constraint functions are often augmented to include robustness considerations, usually the statistical moments of the stochastic system, such as the mean and the standard deviation. In this study, the robust trajectory optimization problem P^R is formulated as follows:

$$\text{Problem}: \ P^R$$
$$\begin{cases} \text{find} & \mathbf{U}(t) \\ \text{min} & J_A = k_\mu \cdot J_\mu(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}), t) + k_\sigma \cdot J_\sigma(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}), t) \quad \text{(a)} \\ \text{s.t.} & \dot{X}(t, \boldsymbol{\xi}) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}), t, \boldsymbol{\xi}) \quad \text{(b)} \\ & \begin{cases} \mathbf{C}_\mu(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}), t) \leq \mathbf{0} \\ \mathbf{B}_\mu\left(\mathbf{X}(t_0, \boldsymbol{\xi}), t_0, \mathbf{X}\left(t_f, \boldsymbol{\xi}\right), t_f\right) = \mathbf{0} \end{cases} \quad \text{(c)} \\ & \begin{cases} \mathbf{C}_\sigma(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}), t) \leq \boldsymbol{\varepsilon}_C \\ \mathbf{B}_\sigma\left(\mathbf{X}(t_0, \boldsymbol{\xi}), t_0, \mathbf{X}\left(t_f, \boldsymbol{\xi}\right), t_f\right) \leq \boldsymbol{\varepsilon}_B, \end{cases} \quad \text{(d)} \end{cases} \tag{2}$$

where ξ is the random variable vector; J_A is the augmented objective function; J_μ and J_σ are the expectation and standard deviation of J, respectively; k_μ and k_σ are the weight factors of J_μ and J_σ; $\mathbf{X}(t, \xi)$ denotes the stochastic state variable vector; $\dot{X}(t, \xi) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \xi), t, \xi)$ denotes the stochastic ODEs; the variables $\mathbf{C}_\mu$, $\mathbf{B}_\mu$ and $\mathbf{C}_\sigma$, $\mathbf{B}_\sigma$ are the expectations and standard deviations of $\mathbf{C}$ and $\mathbf{B}$, respectively; ε_C and ε_B are allowable standard deviation thresholds of $\mathbf{C}$ and $\mathbf{B}$.

For the above robust optimization model, the deterministic demands are satisfied using mean values and the stability requirements are guaranteed with standard deviations. Robustness of both the objective function and the constraint functions is carried out simultaneously. When this model is performed for trajectory planning problems of HGV, it is necessary to solve the complex stochastic ODEs as well as system robustness considerations.

3. The Proposed Robust Trajectory Optimization Scheme

To solve the above robust trajectory optimization problem, a step-by-step description of the proposed hybrid procedure using NIPC and PSO is given as follows.

Step 1. Transform the stochastic ODEs into augmented higher-dimensional deterministic ODEs using the NIPC method.

For a $q-$dimensional random variable vector ξ, the exact solution $\mathbf{X}(t, \xi)$ to the original stochastic ODEs of equation (2b) can be represented as the following NIPC model:

$$\mathbf{X}(t, \boldsymbol{\xi}) \approx \widetilde{X}(t, \boldsymbol{\xi}) = \sum_{i=0}^{p} \widetilde{X}_i(t)\phi_i(\boldsymbol{\xi}), \tag{3}$$

where $\widetilde{X}(t, \xi)$ is the approximation of $\mathbf{X}(t, \xi)$; p is the number of expansion terms that is determined by $p + 1 = (((q + r)!)/(q!r!))$ and r is the desired approximation order of orthogonal polynomials; $\widetilde{X}_i(t)$ are the PC expansion coefficients; $\phi_i(\xi)$ are the orthogonal PC basis functions belonging to the Askey family [20] and satisfy the following orthogonal properties:

$$< \phi_i(\boldsymbol{\xi}), \phi_j(x) > = \delta_{ij} < \phi_i(\boldsymbol{\xi})^2 >, \tag{4}$$

where δ_{ij} is the Kronecker delta function, and $\langle \cdot, \cdot \rangle$ denotes the inner product, defined by

$$< f(\boldsymbol{\xi}), g(\boldsymbol{\xi}) > = \int f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) W(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}, \tag{5}$$

where $f(\xi)$ and $g(\xi)$ are functions of ξ and $W(\xi)$ is a weight function. There are a corresponding polynomial basis function and weight function for a random variable with a given distribution. Moreover, for the $q-$dimensional random vector ξ, $\phi_i(\xi)$ and $W(\xi)$ can be obtained based on the product of corresponding functions related to random variables in each dimension.

According to the NIPC expansion scheme, the unknown PC expansion coefficients $\widetilde{X}_i(t)$ of equation (3) can be calculated by the following integral [26]:

$$\widetilde{X}_i(t) = \frac{1}{\langle \phi_i(\boldsymbol{\xi})^2 \rangle} \int \mathbf{X}(t, \boldsymbol{\xi}) \phi_i(\boldsymbol{\xi}) W(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad i = 0, 1, 2, \ldots, p. \tag{6}$$

Based on the above formulation, the full tensor-product numerical quadrature rule can be used to approximate $\widetilde{X}_i(t)$, i.e.,

$$\widetilde{X}_i(t) = \sum_{k_1=1}^{m} \cdots \sum_{k_q=1}^{m} \mathbf{X}\left(t, \xi_{k_1}, \ldots, \xi_{k_q}\right) \frac{\phi_i\left(\xi_{k_1}, \ldots, \xi_{k_q}\right)}{\langle \phi_i^2 \rangle} \prod_{j=1}^{q} \omega_{k_j}, \tag{7}$$

where m is the number of quadrature points along each dimension of ξ; $\xi_{k_j}, k = 1, \ldots, m, j = 1, \ldots, q$, are the kth integration points of the jth component of ξ, and ω_{k_j} is the multidimensional quadrature weight corresponding to the jth dimension of the quadrature point $\xi_{k_j}, k = 1, \dots, m$.

To implement the above numerical approximation of equation (7), $(\xi_{k_1}, \dots, \xi_{k_q})$ and ω_{k_j} need to be determined firstly according to a proper quadrature rule, such as Gaussian quadrature. Then, compute $\phi_i(\xi_{k_1}, \dots, \xi_{k_q})$ and $\langle \phi_i^2 \rangle$. Afterward, calculate $\mathbf{X}(t, \xi_{k_1}, \dots, \xi_{k_q})$ by substituting $(\xi_{k_1}, \dots, \xi_{k_q})$ into equation (2b). According to equation (7), there are $\prod_{i=1}^{q} m = m^q$ quadrature points of $(\xi_{k_1}, \dots, \xi_{k_q})$, i.e., $\{\xi_1, \xi_2, \dots, \xi_n\}, n = m^q$. Therefore, m^q deterministic ODEs are generated and need to be solved, i.e.,

$$\begin{cases} \dot{X}(t, \boldsymbol{\xi}_1) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}_1), t, \boldsymbol{\xi}_1) \\ \dot{X}(t, \boldsymbol{\xi}_2) = \mathrm{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}_2), t, \boldsymbol{\xi}_2) \\ \vdots \\ \dot{X}(t, \boldsymbol{\xi}_n) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}_n), t, \boldsymbol{\xi}_n), \quad n = m^q, \end{cases} \tag{8}$$

where $\dot{X}(t, \xi_n) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \xi_n), t, \xi_n)$ is the augmented deterministic ODEs with ξ_n.

Step 2. Establish the equivalent deterministic trajectory optimization model P^{ED} of P^R based on NIPC.

According to the NIPC model of equation (3), the original stochastic state $\mathbf{X}(t, \xi)$ can be represented as $\widetilde{X}(t, \xi)$. When the PC expansion coefficients $\widetilde{X}_i(t), i = 0, 1, \dots, p$ have been calculated, its first two statistical moments can be easily computed as follows:

$$\begin{cases} \mathbf{X}_\mu(t, \boldsymbol{\xi}) = \widetilde{X}_0(t), \\ \mathbf{X}_\sigma(t, \boldsymbol{\xi}) = \sqrt{\sum_{i=1}^{p} \widetilde{X}_i(t)^2 < \phi_i^2 >} \end{cases}, \tag{9}$$

where $\mathbf{X}_\mu(t, \xi)$ and $\mathbf{X}_\sigma(t, \xi)$ are the mean and the standard deviations of $\mathbf{X}(t, \xi)$.

The original performance objective J and the constraints C and B are all functions with respect to the approximate stochastic state $\widetilde{X}(t, \xi)$. Therefore, their statistical moments can also be calculated deterministically based on the PC expansion coefficients and their specific function forms using corresponding computation rules, such as addition, subtraction, multiplication, and division [29].

Afterward, the original stochastic optimal control problem P^R is approximately transcribed into a deterministic optimal control problem P^{ED}, i.e.,

$$\text{Problem}: P^{ED}$$
$$\begin{cases} \text{find} \quad \mathbf{U}(t) \\ \min \quad J_A = k_\mu \cdot J_\mu(\mathbf{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) + k_\sigma \cdot J_\sigma(\mathbf{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) & \text{(a)} \\ \text{s.t.} \quad \begin{cases} \dot{X}(t, \boldsymbol{\xi}_1) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}_1), t, \boldsymbol{\xi}_1) \\ \dot{X}(t, \boldsymbol{\xi}_2) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}_2), t, \boldsymbol{\xi}_2), \ n = m^q \\ \vdots \\ \dot{X}(t, \boldsymbol{\xi}_n) = \mathbf{F}(\mathbf{U}(t), \mathbf{X}(t, \boldsymbol{\xi}_n), t, \boldsymbol{\xi}_n) \end{cases} & \text{(b)} \\ \qquad \begin{cases} \mathbf{C}_\mu(\mathbf{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) \le \mathbf{0} \\ \mathbf{B}_\mu(\widetilde{X}(t_0, \boldsymbol{\xi}), t_0, \widetilde{X}(t_\mathrm{f}, \boldsymbol{\xi}), t_\mathrm{f}) = \mathbf{0} \end{cases} & \text{(c)} \\ \qquad \begin{cases} \mathbf{C}_\sigma(\mathbf{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) \le \boldsymbol{\varepsilon}_\mathrm{C}, \\ \mathbf{B}_\sigma(\widetilde{X}(t_0, \boldsymbol{\xi}), t_0, \widetilde{X}(t_\mathrm{f}, \boldsymbol{\xi}), t_\mathrm{f}) \le \boldsymbol{\varepsilon}_\mathrm{B}. \end{cases} & \text{(d)} \end{cases} \tag{10}$$

The main difficulty in the application of the above model lies in the repeated solutions of the augmented deterministic ODEs to obtain the PC expansion coefficients. In particular, for the complex flight systems of HGV, this process requires a lot of calculation and time. Some mathematical techniques are adopted to improve this problem in this study.

Step 3. Discretize the continuous optimal control problem to P^{EDZ} and use the numerical integration scheme to simplify the calculation of PC expansion coefficients.

To solve the above continuous optimization problem P^{ED} using direct optimization algorithms, discretization strategies are necessary and the number of decision variables always affects the calculation amount of the discrete optimization process significantly. In the problem of P^{ED}, there are m^q deterministic ODEs and the number of state variables is quite a lot. Thus, for reducing the number of decision variables, the control parameterization method [33] instead of the state and control parameterization method is adopted here to discretize the continuous optimization problem.

The time interval from t_0 to t_f is first divided into N subintervals as follows:

$$\mathbf{Z} = \{t_0, t_1, \dots, t_N\}, \quad t_0 < t_1 < \cdots < t_{N-1} < t_N = t_f, \tag{11}$$

where $\mathbf{Z}$ is the discretized sampling time.

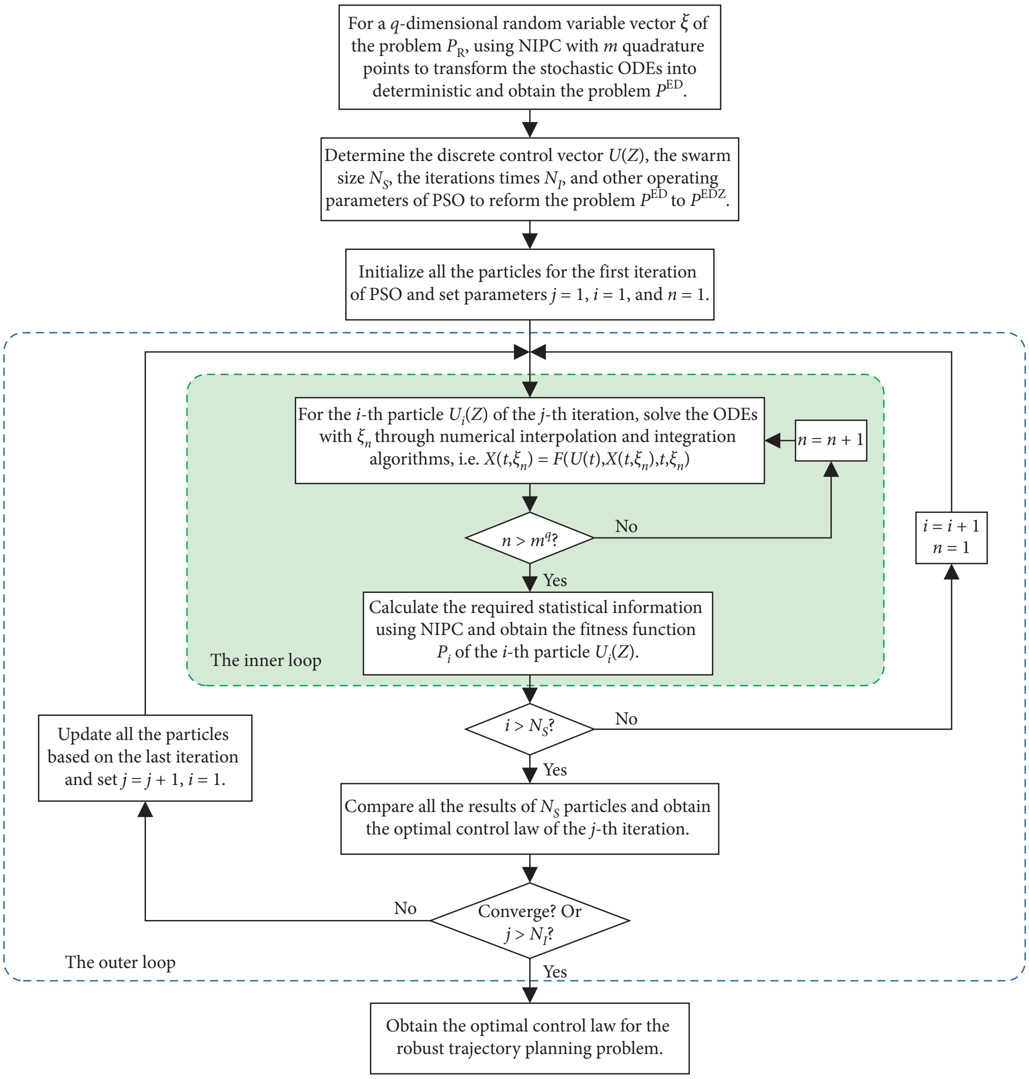
For a q-dimensional random variable vector ξ of the problem P_R, using NIPC with m quadrature points to transform the stochastic ODEs into deterministic and obtain the problem P^{ED}.
Determine the discrete control vector $U(Z)$, the swarm size N_S, the iterations times N_I, and other operating parameters of PSO to reform the problem P^{ED} to P^{EDZ}.
Initialize all the particles for the first iteration of PSO and set parameters $j = 1$, $i = 1$, and $n = 1$.
For the i-th particle $U_i(Z)$ of the j-th iteration, solve the ODEs with ξ_n through numerical interpolation and integration algorithms, i.e. $X(t,\xi_n) = F(U(t),X(t,\xi_n),t,\xi_n)$
$n = n + 1$
$n > m^q$?
No
Yes
Calculate the required statistical information using NIPC and obtain the fitness function P_i of the i-th particle $U_i(Z)$.
The inner loop
$i = i + 1$
$n = 1$
$i > N_S$?
No
Yes
Update all the particles based on the last iteration and set $j = j + 1$, $i = 1$.
Compare all the results of N_S particles and obtain the optimal control law of the j-th iteration.
Converge? Or $j > N_I$?
No
Yes
The outer loop
Obtain the optimal control law for the robust trajectory planning problem.

Figure 1: Flow chart of the algorithm program.

Then, the control variables at the discrete nodes are selected as the decision variables to be optimized, that is,

$$\mathbf{U}(\mathbf{Z}) = \left[\mathbf{U}_{t_0}^T, \mathbf{U}_{t_1}^T, \ldots, \mathbf{U}_{t_N}^T\right]^T, \tag{12}$$

where $\mathbf{U}(\mathbf{Z})$ is the discretized control vector.

The discrete-time dynamic optimization problem is established by fewer design variables; however, the m^q expanded ODEs are still not resolved. In this study, the numerical integration scheme is incorporated to simplify the calculation of PC expansion coefficients.

For the discretized control vector $\mathbf{U}(\mathbf{Z})$, during each segment $\Delta t \in [t_i, t_{i+1}]$, it can be approximated by linear interpolation, i.e.,

$$\begin{gathered}\mathbf{U}(\Delta t_i) \approx \mathbf{U}_{t_i} + \frac{\mathbf{U}_{t_{i+1}} - \mathbf{U}_{t_i}}{t_{i+1} - t_i}(\Delta t_i - t_i), \\ \Delta t_i \in [t_i, t_{i+1}], i = 0, 1, \ldots, N - 1.\end{gathered} \tag{13}$$

The approximate continuous control variable $\widetilde{U}(t)$ can be represented as follows:

$$\widetilde{U}(t) = \sum_{i=0}^{N-1} \mathbf{U}(\Delta t_i). \tag{14}$$

For the convenience of description, represent the expanded ODEs of equation (8) in a brief form as follows:

$$\dot{X}_D(t) = \mathbf{F}_D(\widetilde{U}(t)t, n\mathbf{X}_{\mathrm{D}}q(t)h,_t), \tag{15}$$

where $\mathbf{X}_D$ denotes the deterministic states $\{\mathbf{X}(t,\xi_1), \mathbf{X}(t,\xi_2), \ldots, \mathbf{X}(t,\xi_n)\}$ and $\mathbf{F}_{\mathrm{D}}$ denotes the augmented deterministic ODEs. According to the numerical integration theory, the state variables propagate from t_0 to t_f and their values of any time can be obtained. Herein, by providing control inputs $\widetilde{U}(t)$ and necessary initial conditions, a brief fixed-step fourth-order Runge-Kutta method is applied to effectively reduce the amount of calculation; i.e., when

Table 1: Initial conditions and terminal constraints of the states.

Constraints	y (km)	θ (°)	φ (°)	V (m/s)	γ (°)	ψ (°)
Initial conditions	100	160	25	7200	-2	58
Terminal constraints	20	232	37	1000		

Table 2: Gauss Legendre quadrature points and weights.

Points	Weights
−0.93247	0.17132
−0.66121	0.36076
−0.23862	0.46791
0.23862	0.46791
0.66121	0.36076
0.93247	0.17132

Table 3: Complexity analysis of the three cases.

Methods	Particles	Iterations	Number of ODEs	Number of integrations	Running time (s)
DO	50	100	1	5000	213.9
TRO	50	100	50	250000	10550.2
NIPC-PSO	50	100	36	180000	7670.4

$\mathbf{X}_D(t_k)$ is already obtained, $\mathbf{X}_D(t_{k+1}) = \mathbf{X}_D(t_k + h)$ is iteratively computed by

$$\mathbf{X}_D(t_{k+1}) = \mathbf{X}_D(t_k) + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \quad (16)$$

where

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{F}_D(\widetilde{U}(t_k), \mathbf{X}_\mathrm{D}(t_k), t_k),\\
\mathbf{k}_2 &= \mathbf{F}_D\left(\widetilde{U}\left(t_k + \frac{h}{2}\right), \mathbf{X}_\mathrm{D}(t_k) + \frac{h}{2}\mathbf{k}_1, t_k + \frac{h}{2}\right),\\
\mathbf{k}_3 &= \mathbf{F}_D\left(\widetilde{U}\left(t_k + \frac{h}{2}\right), \mathbf{X}_\mathrm{D}(t_k) + \frac{h}{2}\mathbf{k}_2, t_k + \frac{h}{2}\right)\\
\mathbf{k}_4 &= \mathbf{F}_D(\widetilde{U}(t_k + h), \mathbf{X}_\mathrm{D}(t_k) + h\mathbf{k}_3, t_k + h),
\end{aligned} \quad (17)$$

where $h = t_{k+1} - t_k$ is the fixed-step length of integrations. This approximation makes it possible to represent the PC expansion coefficients as a function of the discrete control variables $\mathbf{U}(\mathbf{Z})$. Moreover, the solution to the m^q deterministic ODEs of the PC expansion coefficients is simplified through numerical integrations. The discrete-time dynamic optimization problem P^{EDZ} is as follows:

$$\text{Problem: } P^{EDZ}$$

$$\begin{cases}
\text{find} & \mathbf{U}(\mathbf{Z}) \\
\min & J = k_\mu \cdot J_\mu(\widetilde{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) + k_\sigma \cdot J_\sigma(\widetilde{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) \quad \text{(a)} \\
\text{s.t.} & \dot{X}_D(t) = \mathbf{F}_D(\widetilde{\mathbf{U}}(t), \mathbf{X}_D(t), t) \quad \text{(b)} \\
& \begin{cases} \mathbf{C}_\mu(\widetilde{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) \le \mathbf{0} \\ \mathbf{B}_\mu\left(\widetilde{X}(t_0, \boldsymbol{\xi}), t_0, \widetilde{X}\left(t_f, \boldsymbol{\xi}\right), t_f\right) = \mathbf{0} \end{cases} \quad \text{(c)} \\
& \begin{cases} \mathbf{C}_\sigma(\widetilde{U}(t), \widetilde{X}(t, \boldsymbol{\xi}), t) \le \boldsymbol{\varepsilon}_C, \\ \mathbf{B}_\sigma\left(\widetilde{X}(t_0, \boldsymbol{\xi}), t_0, \widetilde{X}\left(t_f, \boldsymbol{\xi}\right), t_f\right) \le \boldsymbol{\varepsilon}_\mathrm{B}. \end{cases} \quad \text{(d)}
\end{cases} \quad (18)$$

Table 4: Statistical results of the three cases.

	Mean values			Standard deviations		
	DO	TRO	NIPC-PSO	DO	TRO	NIPC-PSO
$\|P\|$	80.57	92.19	41.15	42.61	23.78	13.99
$\dot{Q}_{\max}$ (kW/m^2)	854.13	776.84	815.97	27.33	28.55	25.26
$q_{\max}$ (kPa)	77.36	47.68	41.10	14.86	48.62	5.12
$n_{\max}$	6.08	3.80	3.10	0.68	0.44	0.57
$y_{(t_f)}$ (m)	19726	22833	20088	2405.61	1803.71	780.09
$\theta_{(t_f)}$ (°)	231.75	231.05	231.53	3.03	2.61	1.83
$\varphi_{(t_f)}$ (°)	37.01	37.15	36.89	0.27	0.29	0.18

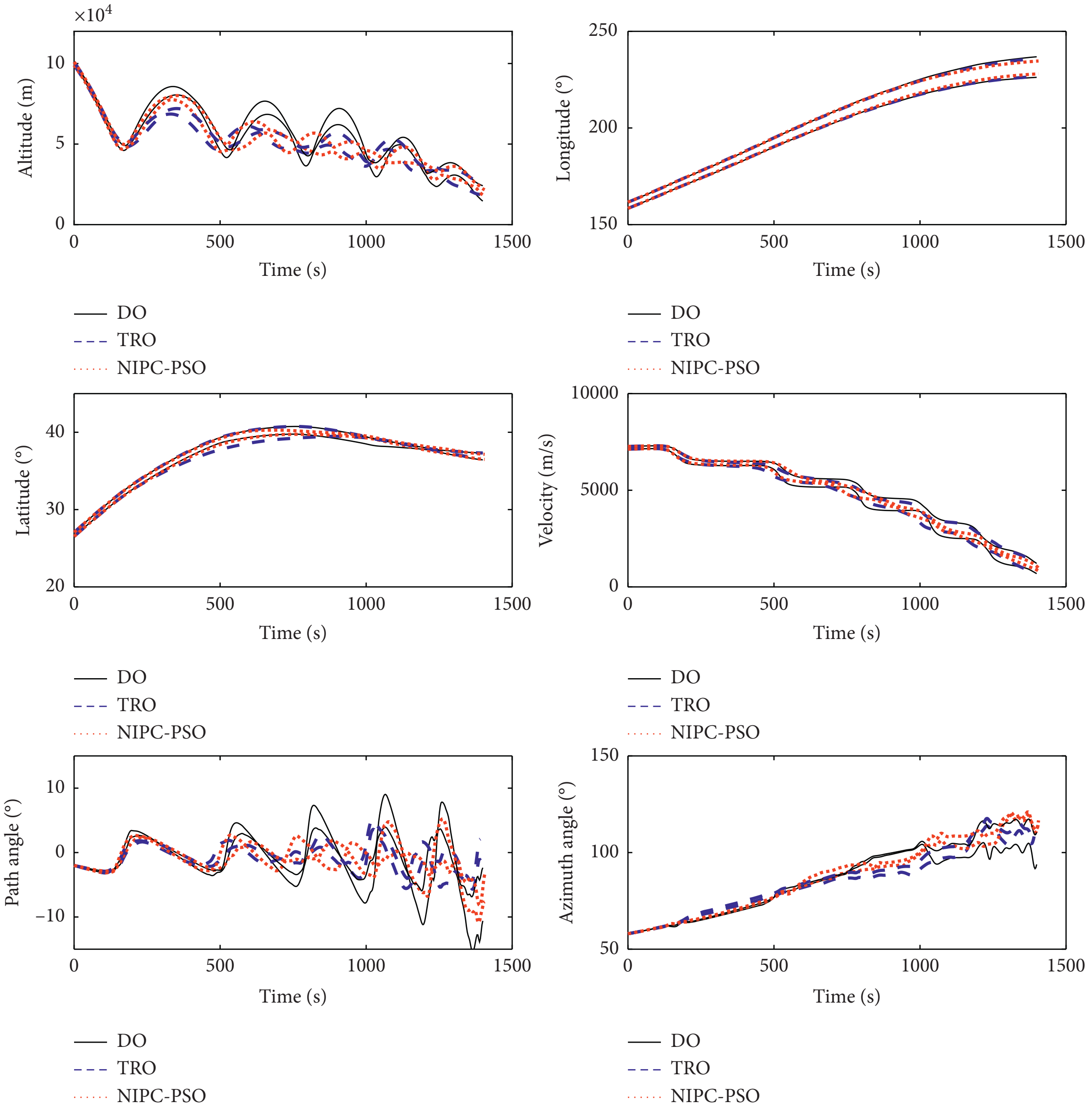

Figure 2: Boundaries of the trajectory states in MC tests of the three cases.

Step 4. Incorporate the NIPC with PSO and solve the problem $P^{\mathrm{E}DZ}$ by the double-loop NIPC-PSO framework.

Since the iteration optimization process of PSO depends on the performance of its fitness function, it is not directly available for the constrained optimization

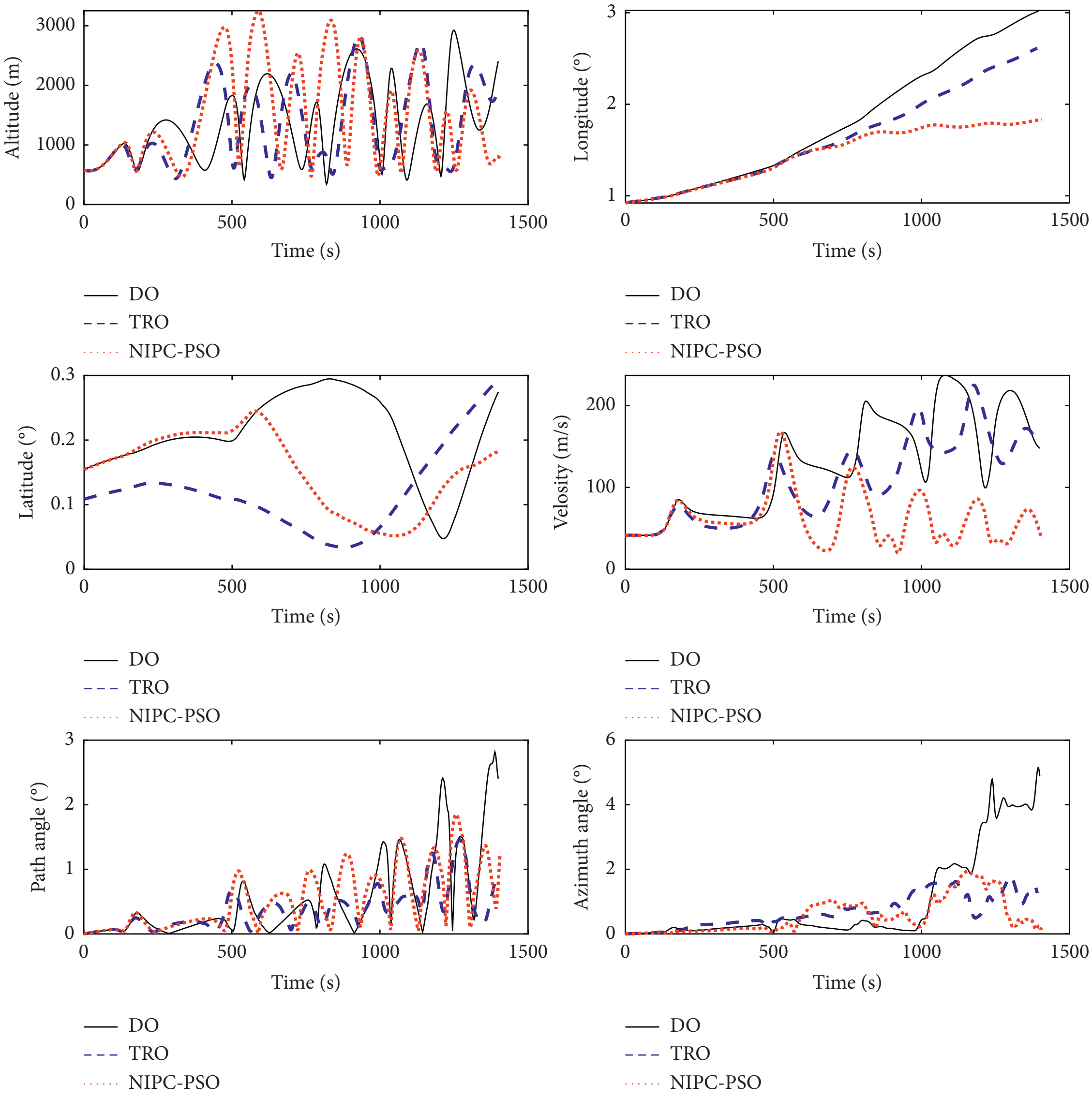

Figure 3: Comparison of the trajectory state standard deviations in MC tests of the three cases.

problem of equation (18). To solve this problem, a PSO application model is established here using the penalty function method, i.e.,

$$\text{Problem: } P^{EDZ-PSO}$$

$$\begin{cases} \text{find} & \mathbf{U}(\mathbf{Z}) \\ \min & P = p_{J_A} \cdot J_A + p_{C\mu} \cdot \mathbf{C}_\mu + p_{B\mu} \cdot \mathbf{B}_\mu + p_{C\sigma} \cdot \mathbf{C}_\sigma + p_{B\sigma} \cdot \mathbf{B}_\sigma \quad \text{(a)} \\ \text{s.t.} & \dot{X}_D(t) = \mathbf{F}_D(\tilde{U}(t), \mathbf{X}_D(t), t), \quad \text{(b)} \end{cases} \tag{19}$$

where P is the integrated penalty function and p_* denote the coefficients of different penalty components. These coefficients can be determined according to the importance of different penalty components using some evaluation methods, such as the analytic hierarchy process (AHP) [34]. The original equality constraint $\mathbf{B}_\mu$ can be expressed by a pair of inequality constraints. It should be noted that significant

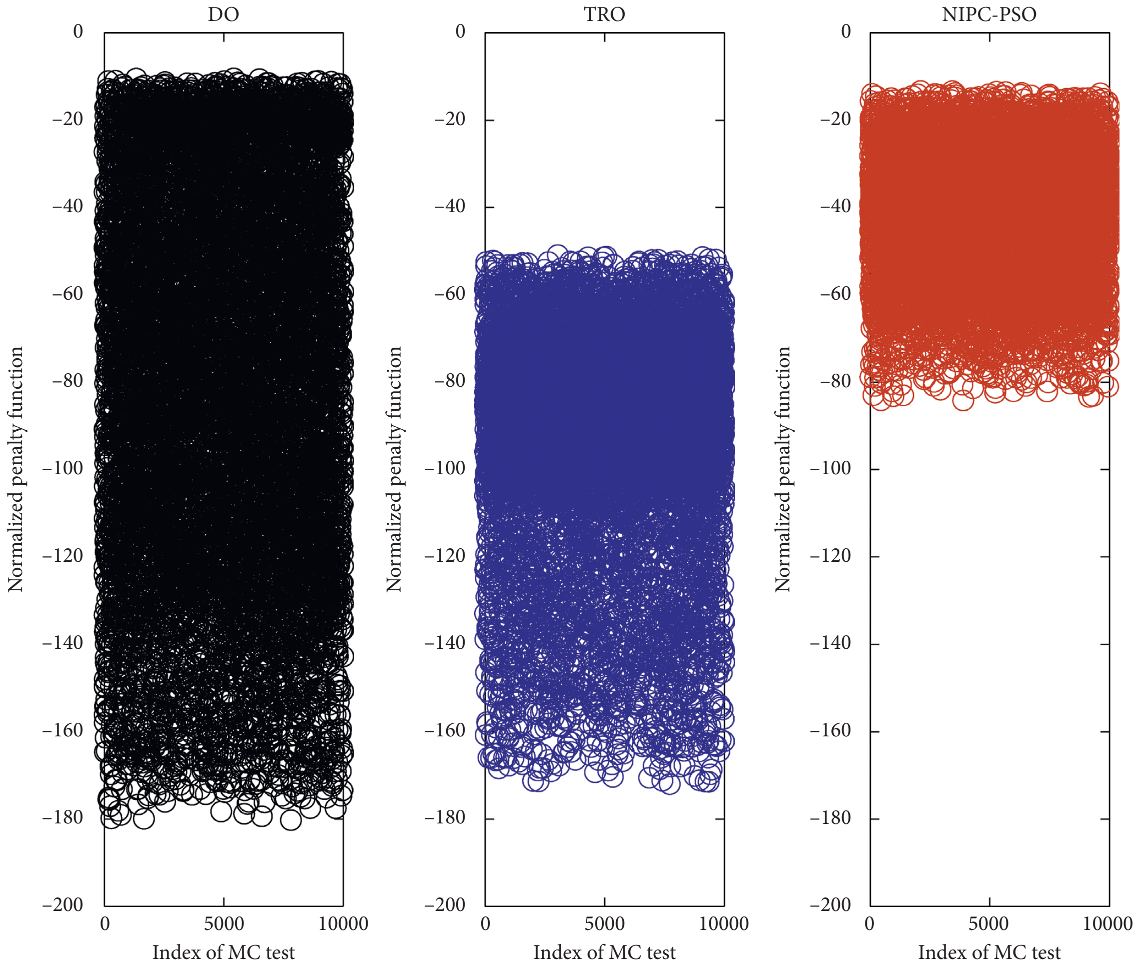

Figure 4: Comparison of the normalized penalty functions in MC tests of the three cases.

differences between the magnitudes of the penalty components will result in calculation difficulties. For fairness, the normalization process should be applied to map different objectives into comparable scales in some cases.

For PSO, its population is represented by a swarm of N_S particles, i.e., $\Omega_p = \{(\mathbf{u}_k, \mathbf{v}_k)\}, k = 1, 2, \ldots, N_S$, and the maximum number of iterations is set to be N_I. Each particle represents a potential solution to the optimization problem $P^{EDZ-PSO}$ and consists of a position vector and a velocity vector as follows:

$$\begin{cases} \mathbf{u}_k = \left(u_{k,1}, u_{k,2}, \ldots, u_{k,n_Z}\right)^{\mathrm{T}}, \\ \mathbf{v}_k = \left(v_{k,1}, v_{k,2}, \ldots, v_{k,n_Z}\right)^{\mathrm{T}}, \end{cases} \tag{20}$$

where $\mathbf{u}_k$ and $\mathbf{v}_k$ denote the position vector and velocity vector of particle k, respectively; n_Z is the number of discrete points of decision parameters in equation (12); namely, $n_Z = \dim(\mathbf{U}(\mathbf{Z}))$.

For the first iteration, a swarm of N_S particles $\Omega_p^0 = \{(\mathbf{u}_k^0, \mathbf{v}_k^0), k = 1, 2, \ldots, N_S\}$ is generated randomly, i.e.,

$$\begin{cases} \mathbf{u}_k^0 = \mathbf{u}_{\mathrm{L}} + r_u \cdot (\mathbf{u}_{\mathrm{U}} - \mathbf{u}_{\mathrm{L}}) \\ \mathbf{v}_k^0 = \mathbf{v}_{\mathrm{L}} + r_v \cdot (\mathbf{v}_{\mathrm{U}} - \mathbf{v}_{\mathrm{L}}) \end{cases}, \quad k = 1, 2, \cdots, N_S, \tag{21}$$

where $\mathbf{u}_L$ and $\mathbf{u}_U$ are, respectively, the lower and upper bounds of the design parameters $\mathbf{U}(\mathbf{Z})$; r_u and r_v are random coefficients between 0 and 1; $\mathbf{v}_L$ and $\mathbf{v}_U$ are the lower and upper bounds for the velocity vector.

For the$s-th$ iteration, the performance of the swarm of N_S particles is represented by their fitness functions $P_k^s, k = 1, 2, \ldots, N_S$, which are calculated, respectively, in the inner loop based on the NIPC method and the numerical integration scheme. When the calculation of the inner loop is all finished, the best position $\mathbf{p}_k^s$ of each particle k is recorded and the global best position $\mathbf{p}_g^s$ of the whole swarm up to the current iteration is updated, i.e.,

$$\begin{aligned} \mathbf{p}_k^s &= \mathbf{u}_k^{l_m}, l_m = \arg\min\nolimits_{p_1=1,2,\ldots,s} P_k^{p_1}, \quad k = 1, 2, \ldots, N_S, \\ \mathbf{p}_g^s &= \mathbf{u}_{q_m}^{z_m}, (z_m, q_m) = \arg\min_{\substack{z_1=1,2,\ldots,s \\ q_1=1,2,\ldots,N_S}} P_{q_1}^{z_1}. \end{aligned} \tag{22}$$

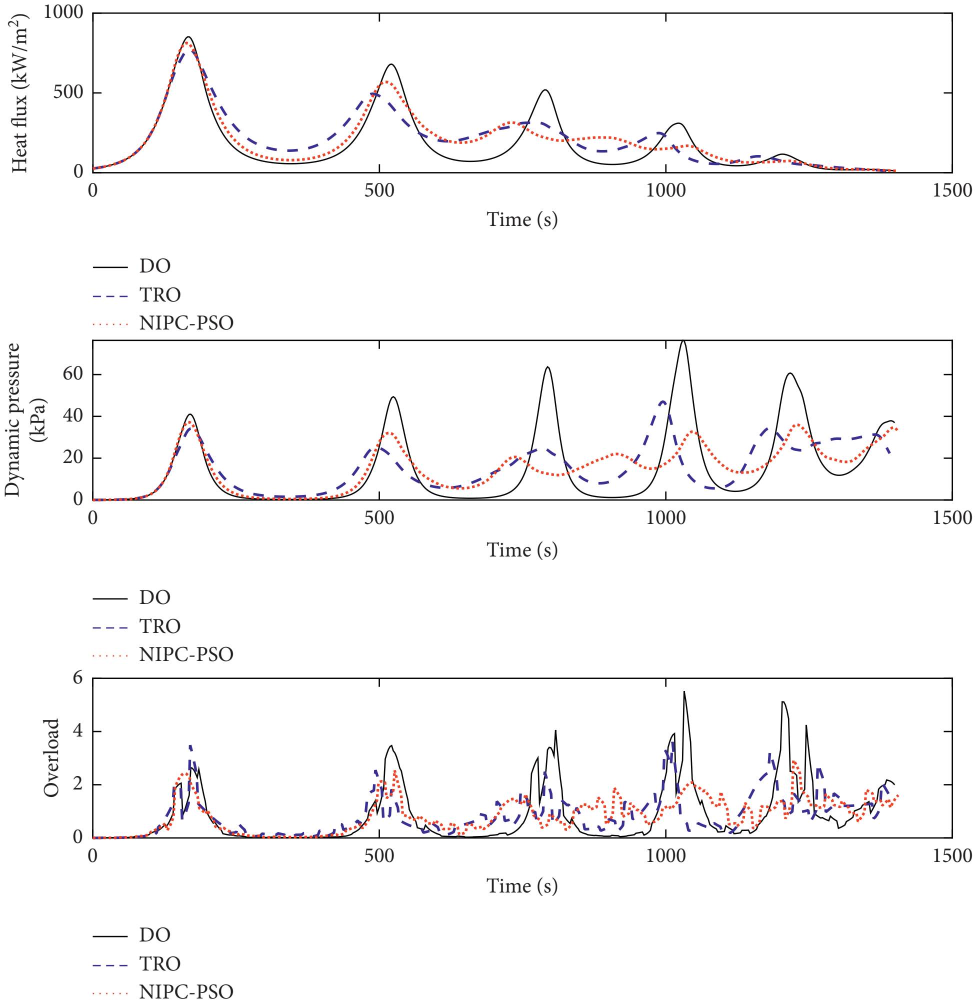

Figure 5: Mean values of path constraints in MC tests of the three cases.

Afterward, the velocity and position vector of each particle are updated in the outer loop as follows:

$$\begin{cases} \mathbf{v}_k^{s+1} &= \omega \cdot \mathbf{v}_k^s + c_1 \cdot r_1 \cdot \left(\mathbf{p}_k^s - \mathbf{u}_k^s\right) + c_2 \cdot r_2 \cdot \left(\mathbf{p}_g^s - \mathbf{u}_k^s\right) \\ \mathbf{u}_k^{s+1} &= \mathbf{u}_k^s + \mathbf{v}_k^{s+1} \end{cases}, \quad k = 1, 2, \ldots, N_S, \tag{23}$$

where ω denotes the inertia parameter and can be linearly decreased with iteration for better optimization effect [35], i.e., $\omega^{s+1} = \omega_{\max} - s\left(\left(\omega_{\max} - \omega_{\min}\right)/N_I\right), s = 1, 2, \ldots, N_I$; c_1 and c_2 are two trust parameters, which represent confidence the particle has in itself and in the swarm, respectively; r_1 and r_2 are two independent random variables between 0 and 1. To keep the position vector of the next generation within the feasible region, $\mathbf{v}_L$ and $\mathbf{v}_U$ are chosen as follows:

$$\begin{cases} \mathbf{v}_L = \mathbf{u}_L - \mathbf{u}_U, \\ \mathbf{v}_U = \mathbf{u}_U - \mathbf{u}_L. \end{cases} \tag{24}$$

Moreover, if any element in the updated velocity vector and position vector violates its limits, it is set as the closest boundary value before the next iteration starts.

To show the double-loop optimization algorithm more clearly, a flow chart of its implementation program is shown in Figure 1.

According to the above double-loop optimization framework, the PSO iterates for the global optimal solution in the outer loop, while the corresponding fitness functions with robustness considerations are calculated in the inner loop by the NIPC method and numerical integrations. Finally, the best particle of PSO which represents the optimal solution to the original robust optimization problem is obtained conveniently.

4. The Robust Trajectory Optimization Problem of HGV

In this section, a typical robust trajectory optimization problem of HGV is introduced, and the reference aircraft is chosen as the CAV-H which has been widely used in the research of hypersonic vehicles [9, 36].

Considering the Earth rotation model, the specific trajectory optimization formulation is as follows. Find the design variables

$$\mathbf{U} = \left[\alpha_{\mathrm{c}}\boldsymbol{\sigma}_{\mathrm{c}}\right]^{\mathrm{T}}, \tag{25}$$

which minimize the objective function,

$$J = t_f, \tag{26}$$

and subject to the dynamic constraints

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}t} = V\sin\gamma, \\[2ex] \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dfrac{V\cos\gamma\sin\psi}{y\cos\varphi}, \\[2ex] \dfrac{\mathrm{d}\varphi}{\mathrm{d}t} = \dfrac{V\cos\gamma\cos\psi}{y}, \\[2ex] \dfrac{\mathrm{d}V}{\mathrm{d}t} = -\overline{D} - \dfrac{\sin\gamma}{y^2} + \Omega^2 y\cos\varphi\left(\sin\gamma\cos\varphi - \cos\gamma\sin\varphi\cos\psi\right), \\[2ex] \dfrac{\mathrm{d}\gamma}{\mathrm{d}t} = \dfrac{1}{V}\left(\overline{L}\, cost\, n\sigma_{\mathrm{c}}q + h\dfrac{\cos\gamma}{y}\left(V^2 - \dfrac{1}{y}\right)x + 72C\Omega; V\, cos\,\varphi\, sin\,\psi + \Omega^2 y\, cos\,\varphi\left(\cos\gamma\cos\varphi + \sin\gamma\sin\varphi\cos\psi\right)\right), \\[2ex] \dfrac{\mathrm{d}\psi}{\mathrm{d}t} = \dfrac{1}{V}\left(\dfrac{\overline{L}\sin\sigma_{\mathrm{c}}}{\cos\gamma} + \dfrac{V^2}{y}\cos\gamma\sin\psi\tan\varphi - 2\Omega V\left(\tan\gamma\cos\varphi\cos\psi - \sin\varphi\right) + \dfrac{\Omega^2 y}{\cos\gamma}\sin\varphi\cos\varphi\sin\psi\right). \end{cases} \tag{27}$$

The path constraints are

$$\mathbf{C} = \begin{cases} \dot{Q} = \dfrac{C_1}{\sqrt{R}}\left(\dfrac{\rho}{\rho_s}\right)^{0.5} V^{3.15} \le \dot{Q}_{\max}, \\[2ex] q_r = \dfrac{1}{2}\rho\left(V\cdot V_c\right)^2 \le q_{\max}, \\[2ex] n_r = \sqrt{\overline{L}^2 + \overline{D}^2} \le n_{\max}, \end{cases} \tag{28}$$

and the boundary constraints are

$$\mathbf{B} = \begin{cases} \left|y\left(t_f\right) - y_f\right| \le \varepsilon_{f_y}, \\ \left|\theta\left(t_f\right) - \theta_f\right| \le \varepsilon_{f_\theta}, \\ \left|\varphi\left(t_f\right) - \varphi_f\right| \le \varepsilon_{f_\phi}, \\ \left|V\left(t_f\right) - V_f\right| \le \varepsilon_{f_V}, \end{cases} \tag{29}$$

where α_{c} is the angle of attack and σ_{c} is the bank angle; θ, φ, γ, and ψ are the longitude, the latitude, the path angle, and the azimuth, respectively; y, V, t, and Ω are dimensionless variables of the geocentric distance, the velocity, the time, and the angular rate of Earth rotation, and their dimensionless parameters are R_0, $V_{\mathrm{c}} = (g_0 R_0)^{0.5}$, $(R_0/g_0)^{0.5}$, and $(g_0/R_0)^{0.5}$, respectively; R_0 is the average radius of the Earth and g_0 is the gravitational acceleration at sea level; $\overline{L}$ and $\overline{D}$ represent the dimensionless lift and drag acceleration; $\dot{Q}$, q_r, and n_r are the heat flux, the dynamic pressure, and the overload, respectively, in which R is the vehicle head radius of curvature, ρ is the atmospheric density, ρ_s is the atmospheric density at sea level, and C_1 is a constant; ε_{f_y}, ε_{f_θ}, ε_{f_ϕ}, and ε_{f_V} are error bounds for corresponding terminal states.

In this paper, the path constraints are given as $\dot{Q}_{\max} = 1000\,\mathrm{kW/m^2}$, $q_{\max} = 400\,\mathrm{kPa}$, and $n_{\max} = 6$, respectively. The initial conditions and desired terminal states are shown in Table 1. In the proposed method, the system state variables are obtained using numerical integrations based on given initial conditions; thus, only terminal conditions are required as boundary constraints. As described in equation (29), the corresponding error bounds of the terminal constraints are $\varepsilon_{f_y} = 0$, $\varepsilon_{f_\theta} = 0$, $\varepsilon_{f_\phi} = 0$, and $\varepsilon_{f_V} = 0$, respectively.

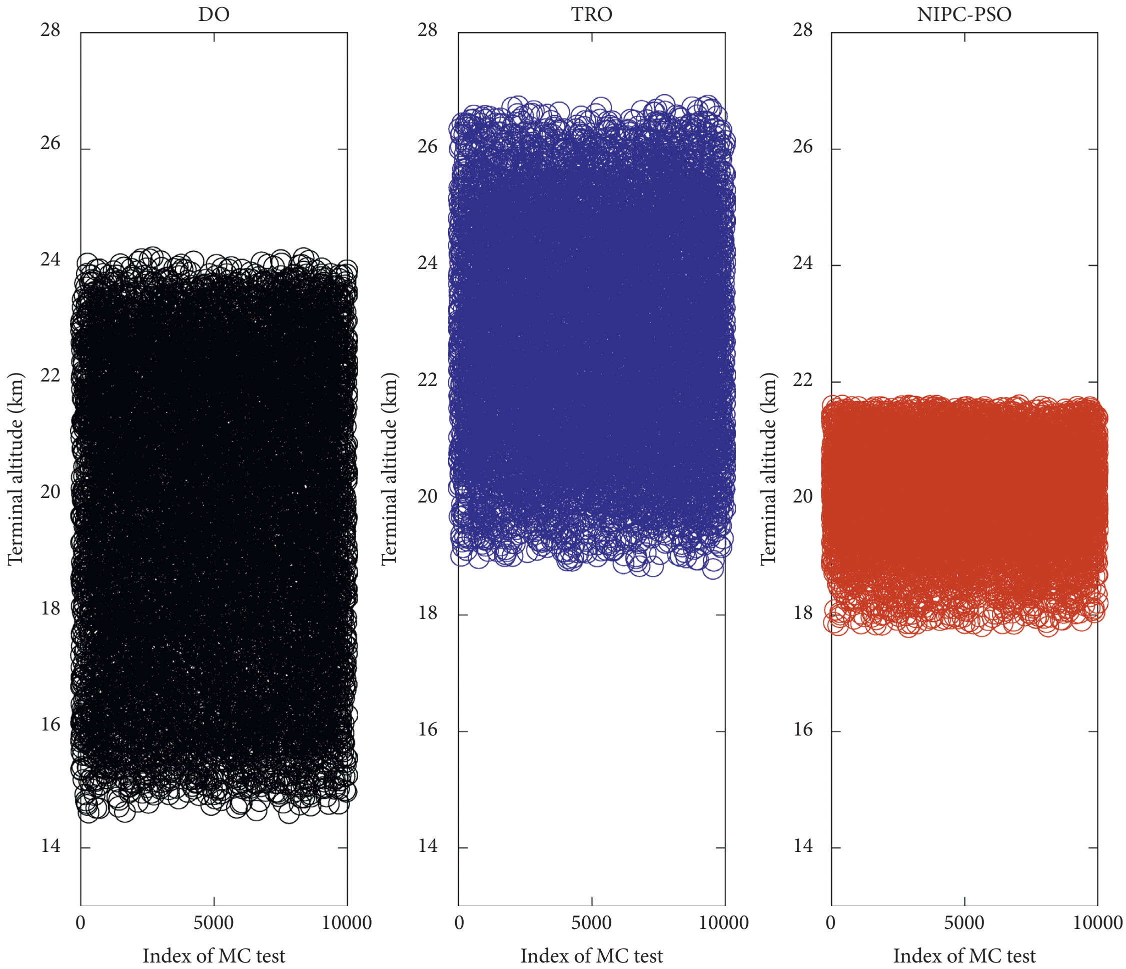

Figure 6: Distributions of terminal altitude in MC tests of the three cases.

Aerodynamic forces play an important role in the unpowered reentry process of HGV. The dimensionless lift and drag acceleration shown in equation (27) is expressed as follows:

$$\begin{cases} \overline{L} = \dfrac{\rho \cdot (V \cdot V_c)^2 \cdot S_V \cdot C_L}{(2m_V \cdot g_0)}, \\ \\ \overline{D} = \dfrac{\rho \cdot (V \cdot V_c)^2 \cdot S_V \cdot C_D}{(2m_V \cdot g_0)}, \end{cases} \tag{30}$$

where $m_V = 907.2\,\text{kg}$ is the vehicle mass; $S_V = 0.4839\,\text{m}^2$ is the pneumatic reference area; C_L and C_D are the lift and drag coefficients, which are functions of the angle of attack and Mach number.

Generally, there are no analytical expressions for aerodynamic coefficients C_L and C_D. Instead, they are calculated through numerical fitting based on experimental aerodynamic data. However, due to the complex atmospheric environment, these experimental data cannot be as accurate as anticipated. Therefore, it is assumed that the variation in C_L and C_D is a kind of uncertainty that can be represented by a single random variable [14, 26]. Otherwise, for the hypersonic vehicle, it is always not easy to achieve the designed reentry point accurately [36]. As a result, random deviations are also unavoidable at the beginning of the reentry process, i.e., the uncertain initial conditions. Based on the above discussions, two kinds of uncertainties are included in this paper, and the uncertain parameters are as follows:

$$\boldsymbol{\xi} = [\xi_C, \xi_I]^T, \tag{31}$$

where ξ_C indicates the aerodynamic uncertainty in the lift and drag coefficients, and ξ_I indicates the state deviation uncertainty in the initial reentry conditions. It is assumed that both ξ_C and ξ_I are uniformly distributed, and they are independent of each other. In summary, the corresponding stochastic model is as follows:

$$\begin{cases} \widetilde{C}_L = C_L(1 + \xi_C), \\ \widetilde{C}_D = C_D(1 + \xi_C), \\ \widetilde{X}_0 = \mathbf{X}_0(1 + \xi_I), \end{cases} \tag{32}$$

where $\widetilde{C}_L$ and $\widetilde{C}_D$ are the stochastic lift and drag coefficients, respectively; $\widetilde{X}_0$ is the stochastic initial reentry states. It is assumed that the aerodynamic uncertainty ξ_C lies in the interval of $[-0.2, 0.2]$ and the state deviation uncertainty ξ_I lies in the interval of $[-0.01, 0.01]$. This means that the

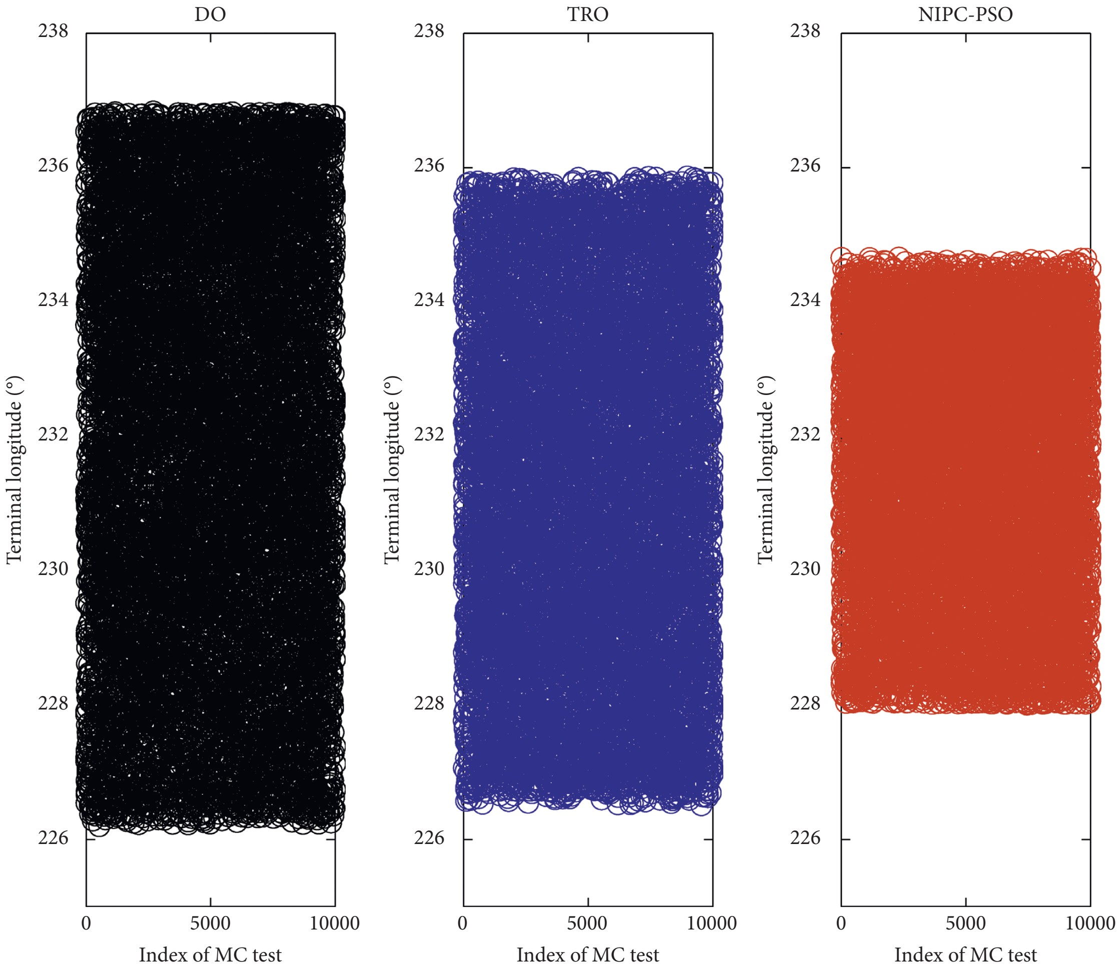

Figure 7: Distributions of terminal longitude in MC tests of the three cases.

deviation of the aerodynamic coefficients around their nominal values is about [−20%, +20%], while the deviation of the initial reentry conditions around their nominal values is about [−1%, +1%].

It should be noted that although the state vector includes six variables, we only introduce uncertainty to the altitude, the longitude, the latitude, and the speed due to their more demanding requirements. When equation (32) is incorporated, the deterministic trajectory optimization problem is now reformulated to a robust trajectory optimization problem.

The penalty components of the PSO application model in equation (19) are as follows:

$$\mathbf{J} = \left(t_f, \dot{Q}, q_r, n_r, \left|y\left(t_f\right) - y_f\right|, \left|\theta\left(t_f\right) - \theta_f\right|, \left|\varphi\left(t_f\right) - \varphi_f\right|, \left|V\left(t_f\right) - V_f\right|\right)^{\mathrm{T}}. \tag{33}$$

A maximum normalization method is adopted to map magnitudes of different penalty components into comparable scales, i.e.,

$$\overline{J}_j = \frac{J_j}{J_{jU}} \times 100, \quad j = 1, 2, \ldots, h_0, \tag{34}$$

where $J_j = \mathbf{J}(j)$, $j = 1, 2, \ldots, h_0$, indicate the values of diffident components in equation (33) and h_0 is the number of all the terms; J_{jU} is the corresponding upper limit of J_j and $\overline{J}_j$ is its normalized value. After normalization, all the objective functions are mapped to the interval of [0, 100]. The objective function turns out to be

$$\min P = k_\mu \sum_{j=1}^{h_0} \mu\left(\overline{J}_j\right) k_{\mu_j} + k_\sigma \sum_{j=1}^{h_0} \sigma\left(\overline{J}_j\right) k_{\sigma_j}, \tag{35}$$

where all the penalty coefficients are user-defined parameters determined through the AHP method.

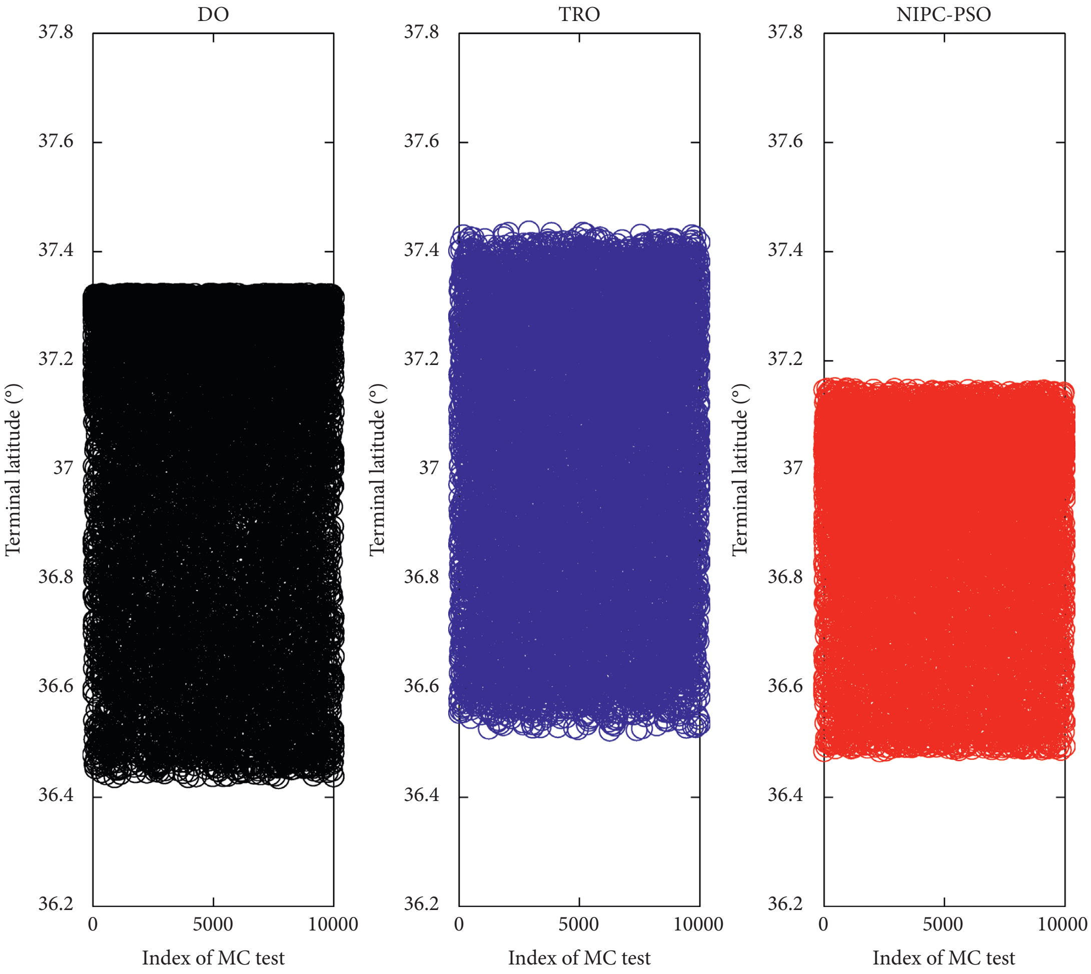

Figure 8: Distributions of terminal latitude in MC tests of the three cases.

5. Simulation and Results

In this section, three simulation cases are carried out to illustrate the proposed algorithm for the above HGV reentry problem, i.e., the deterministic trajectory optimization without consideration of uncertainties (denoted as DO), the traditional robust trajectory optimization using MC (denoted as TRO) [18], and the proposed robust trajectory optimization in this paper (denoted as NIPC-PSO). Herein, TRO is set to a similar structure of NIPC-PSO whereas its uncertainty propagation process is performed using 50 runs of MC simulations instead of the NIPC.

During optimization, the number of control discretization nodes is set as 100; the number of numerical integration points for the states is set as 500. Moreover, the swarm size of the PSO particles is 50; the maximum number of iterations is 100; ω is linearly decreased from 0.9 to 0.4, and $c_1 = c_2 = 2$; the angle of attack is limited within $[10^\circ, 20^\circ]$ and the bank angle is limited within $[-80^\circ, 80^\circ]$. Since the twice-order PC expansion is accurate enough for common applications [37], herein, the NIPC expansion order is set as $r = 2$ and the number of Gauss points is set as $m = 6$ in order to balance the calculation burden and accuracy. The Gauss points and the corresponding weights are shown in Table 2.

All the simulations are performed with Matlab2014 on a personal computer with the Intel Core i5-8400 2.81 GHz processor and 16.0 G of RAM. After abundant numerical simulations, the convergence results of DO, NIPC-PSO, and TRO are achieved. The complexity analysis of the three cases is shown in Table 3.

Since the control parameterization method is adopted in this paper, only control variables are discretized and the number of design variables in the three cases is the same. As introduced in equation (27), HGV's dynamic constraints are complex and time-consuming to solve. Thus, the complexity of the three cases is mainly reflected in their augmented ODEs. As shown in Table 3, DO only needs to deal with the original six-dimensional ODEs, while the number of ODEs in TRO and NIPC-PSO increases to 50 and 36, respectively. The number of numerical integrations as well as the corresponding running time of the three cases is directly proportional to their number of ODEs. It is indicated that both TRO and NIPC-PSO are more complex and time-consuming than DO, but NIPC-PSO is obviously simpler than that TRO of 50 MC simulations. Since the trajectory optimization is carried out in the offline phase of trajectory design, it is worth sacrificing some design time to obtain better robustness performance. Furthermore, under the

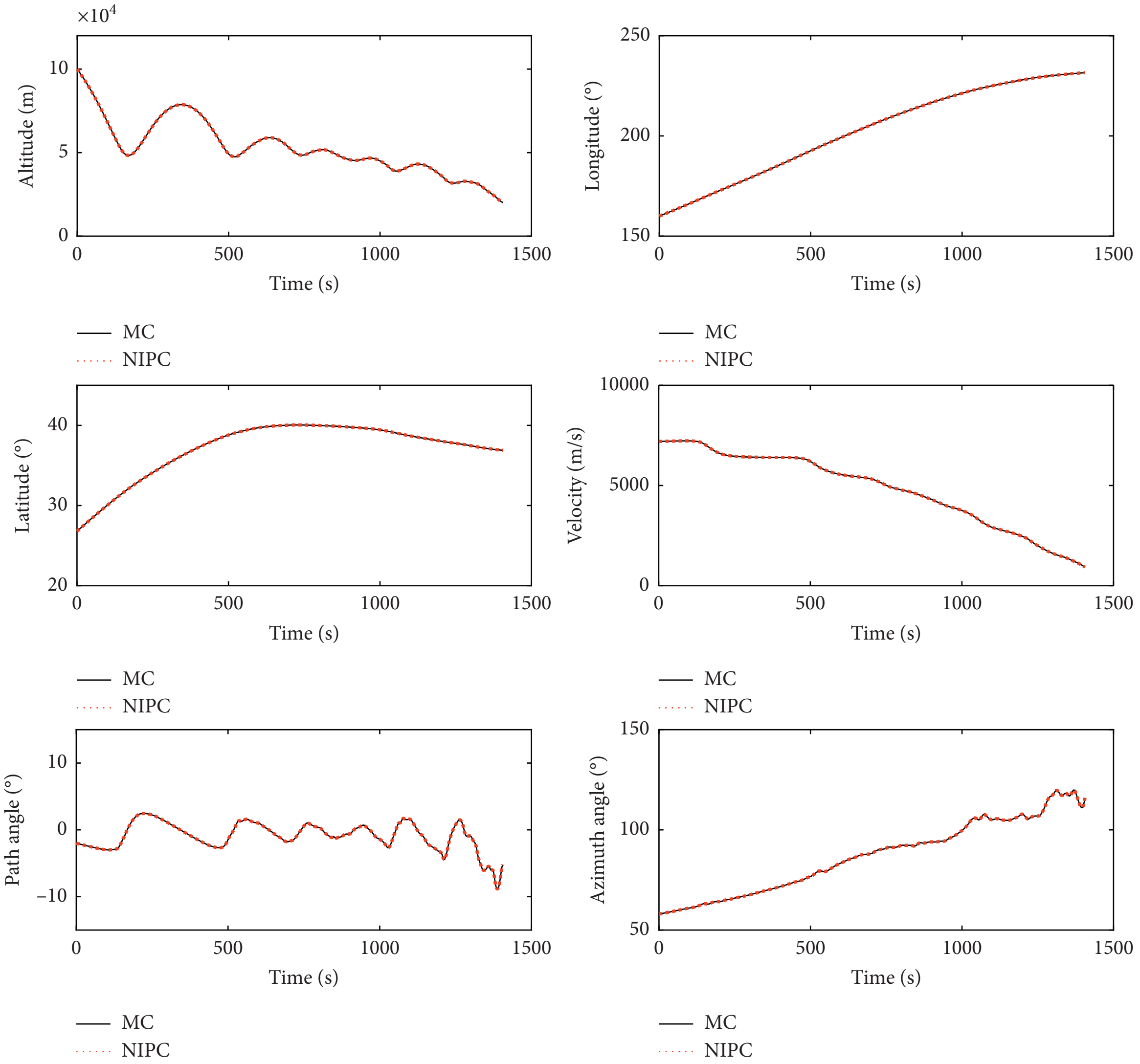

Figure 9: Mean values of states of MC tests and NIPC method in NIPC-PSO case.

double-loop optimization framework of this paper, the main cost of robust optimization is the repeated numerical integrations which is much easier to solve than adding design variables.

To compare the robustness of the designed trajectories in the three cases, the more detailed simulation verification and analysis are carried based on the optimal control laws of DO, TRO, and NIPC-PSO. Specifically, 10,000 times MC tests are implemented through numerical integrations under the same stochastic model of robust optimization. The obtained statistical trajectory data are summarized in Table 4, and the more intuitive comparisons are shown in Figures 2–8 distinctly.

To the statistical data of MC tests, the mean value of different parameters represents corresponding trajectory constraint satisfaction, while the standard deviation is usually regarded as a direct measure of robustness. As shown in Table 4, the mean values of path constraints and boundary constraints for NIPC-PSO all meet the design requirements well, whereas the maximum dynamic pressure of DO exceeds the upper limit of 6 and the terminal altitude of TRO has a large deviation from the reference value of 20,000 m. All the standard deviations of NIPC-PSO are significantly better than those of DO. Moreover, except for the maximum dynamic pressure, they are also better than the standard deviations of TRO. It should be noted that, in some indicators, the standard deviations of TRO are improved, but in others, they become worse. This is because there are only 50 MC simulations in TRO, and the uncertainty propagation process is not accurate enough. To improve this problem, more MC simulations are needed, but the complexity of TRO will be greatly increased.

In summary, the trajectories of DO are more likely to become infeasible with disturbances, and this is because no uncertainty of the online flight is considered by the traditional deterministic optimization. Although there is a little sacrifice in the original objective function of flight time for the inclusion of system robustness considerations, all the trajectory indicators of NIPC-PSO are less sensitive to uncertainties which is exactly what this article prefers. Furthermore, compared with TRO, NIPC-PSO can achieve better performance with much less computation.

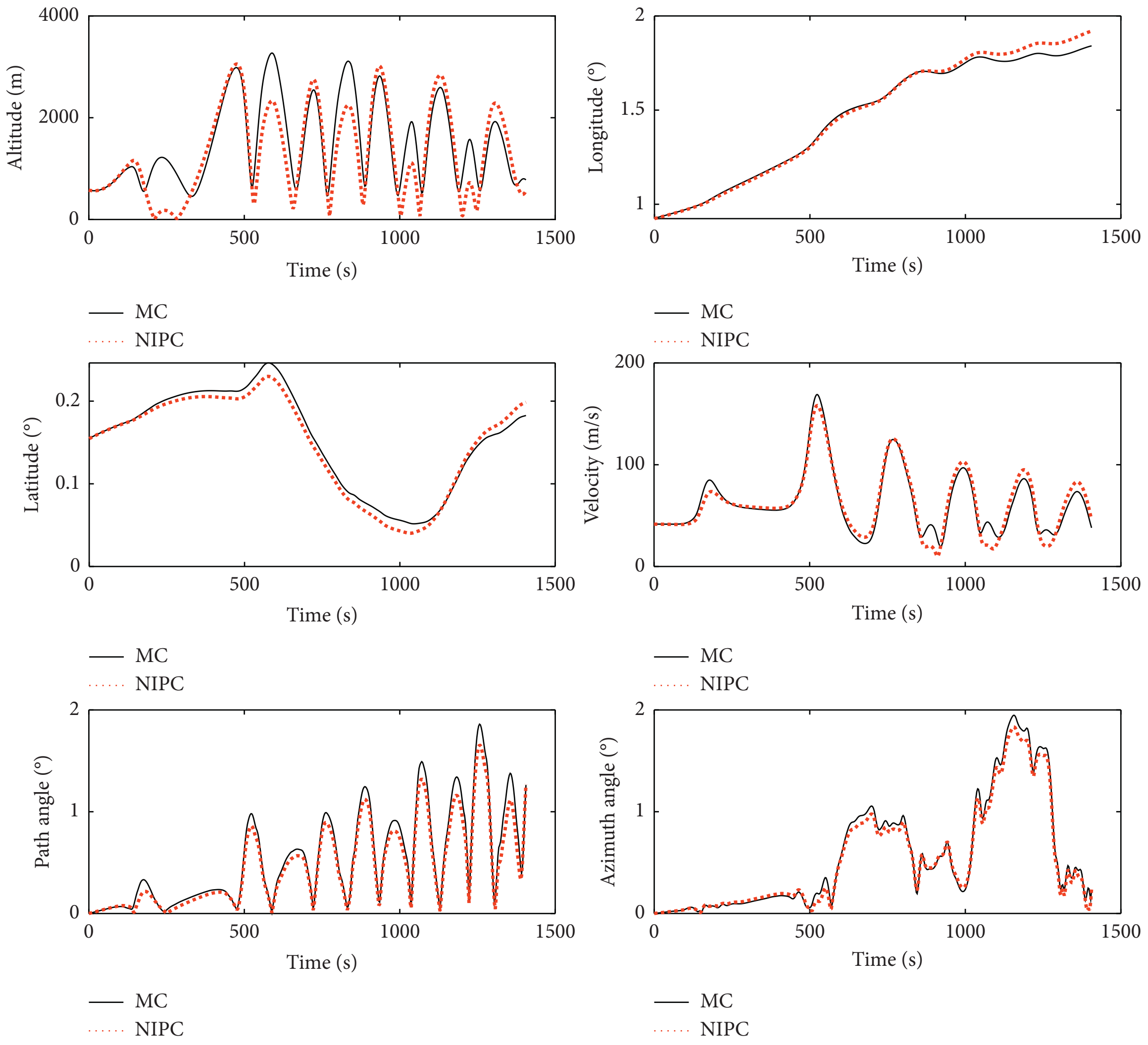

Figure 10: Standard deviations of states of MC tests and NIPC method in NIPC-PSO case.

Based on the MC tests of the three cases, the evolution of different states along the reentry process over time is shown in Figure 2. As can be seen, all the states fluctuate in a certain range around their nominal values and the states of NIPC-PSO have smaller fluctuations.

More clearly, Figure 3 shows the evolution standard deviations of the system states in the three cases. We can see that the standard deviations of NIPC-PSO are always smaller than those of DO and TRO, and it becomes more obvious when approaching the terminal time. This also suggests that the optimal control law of NIPC-PSO is better in tolerability and stability to the uncertainties. The conclusion is consistent with Table 4.

Figure 4 shows the normalized optimal penalty function values of MC tests in the three cases. As the figure reveals, both the stability and optimal values of the penalty functions in NIPC-PSO are the best in the three cases. According to Table 4, the standard deviations of these normalized penalty functions in DO, TRO, and NIPC-PSO are 42.61, 23.78, and 13.99, respectively. Compared with DO, these standard deviations of TRO and NIPC-PSO have been improved by 44% and 67%, respectively. This indicates that both TRO and NIPC-PSO have better robustness performance than DO, whereas the optimal penalty function value of TRO needs further improvement.

Since path constraints are significant to the trajectory performance, the mean values of the heat flux, the dynamic pressure, and the overload of MC tests in the three cases are shown in Figure 5. As we can see, although under uncertainties, path constraints in TRO and NIPC-PSO still show better capability than DO. This conclusion is in accordance with the performance of their penalty functions.

The terminal constraint distributions of altitude, longitude, and latitude are shown in Figures 6–8, respectively. Compared with DO and TRO, the accuracy of terminal state distributions of NIPC-PSO is improved obviously. The accuracy of TRO is not stable enough, and for better performance, it needs more simulations. Simulation results indicate that the proposed algorithm can achieve better robustness with less computation.

In the proposed algorithm, the NIPC method plays an important role in providing statistical information for the double-loop optimization framework. To verify the efficiency and accuracy of the NIPC method adopted in this paper, the statistical information obtained by the NIPC method is compared with that of MC tests. Specifically, based on the control law of NIPC-PSO and the same stochastic HGV model of Section 4, the uncertainty propagations are carried out using the NIPC method and 1000 MC tests, respectively. Statistical analysis is then conducted to obtain the mean and standard diversions of these six trajectory states. The comparison results of the NIPC method and MC tests are shown in Figures 9 and 10, respectively.

As shown in Figure 9, the mean values of the two methods are almost the same for all the states. In Figure 10, their corresponding standard deviations basically match each other well. As expected, the NIPC method is suitable and accurate enough for our robust trajectory planning problem and it has significantly fewer samples than the MC method. However, it should be noted that there are some errors between the two methods in Figure 10, especially in the altitude. This is because the PC expansions adopted in this paper are only twice-order and the altitude changes sharply during the skip reentry process. The approximation accuracy of NIPC can be improved by using the higher-order PC expansions, but the computational amount will be increased somewhat. Considering the trajectory performance achieved above, these errors are acceptable in this paper and the NIPC has good application potential in such complex robust optimization problems.

6. Conclusion

To improve the robustness of the reference trajectory for HGV with parametric uncertainties, a hybrid double-loop NIPC-PSO methodology is developed in this paper. The stochastic optimal control problem is transformed into a deterministic vision with higher-dimensional ODEs and robustness considerations using the NIPC method. The large number of calculations of the PC expansion coefficients is reduced through the control parameterization method and the numerical integration scheme. Based on an unconstrained model with a penalized fitness function, the PSO method searches globally for the optimal control law in the outer loop, while the penalized fitness function with robustness considerations is calculated repeatedly in the inner loop. The HGV reentry simulation of a classical time-optimal trajectory optimization problem with uncertainties in both initial states and aerodynamic coefficients shows that the NIPC-PSO of our proposed method has better robustness and performance in constraint satisfaction compared with the DO and TRO. The proposed procedure improves the feasibility and effectiveness of NIPC-based optimization methods.

Data Availability

The CAV data used to support the findings of this study are included within the article. The more detailed algorithm parameters are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was financially supported by the Natural Science Foundation of Shaanxi Province (No. 2020JQ-318) and the Fundamental Research Funds for the Central Universities (No. XJS190404 and JB190411). The authors gratefully acknowledge all the supports.

References

[1] X. Yu, P. Li, and Y. Zhang, “The design of fixed-time observer and finite-time fault-tolerant control for hypersonic gliding vehicles,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4135–4144, 2017.

[2] R. Zhang, Q. Chen, and H. Guo, “Chebyshev neural network-based adaptive nonsingular terminal sliding mode control for hypersonic vehicles,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 6830141, 10 pages, 2020.

[3] Y. Zhang, X. Wang, and S. Tang, “A globally fixed-time solution of distributed formation control for multiple hypersonic gliding vehicles,” *Aerospace Science and Technology*, vol. 98, 2020.

[4] Z. Yuan-Long, Y. Xie, S. C. Peng, G. J. Tang, and W. M. Bao, “Entry trajectory generation with complex constraints based on three-dimensional acceleration profile,” *Aerospace Science and Technology*, vol. 91, 2019.

[5] H. Zhou, X. Wang, and N. Cui, “Glide trajectory optimization for hypersonic vehicles via dynamic pressure control,” *Acta Astronautica*, vol. 164, pp. 376–386, 2019.

[6] J. Zhao, R. Zhou, and X. Jin, “Progress in reentry trajectory planning for hypersonic vehicle,” *Journal of Systems Engineering and Electronics*, vol. 25, no. 4, pp. 627–639, 2014.

[7] Z. Wei, C. Huang, D. Ding, H. Huang, and H. Zhou, “UCAV formation online collaborative trajectory planning using hp adaptive pseudospectral method,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 3719762, 25 pages, 2018.

[8] 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.

[9] Z. Li, C. Hu, C. Ding, G. Liu, and B. He, “Stochastic gradient particle swarm optimization based entry trajectory rapid planning for hypersonic glide vehicles,” *Aerospace Science and Technology*, vol. 76, pp. 176–186, 2018.

[10] Y. Wu, B. Yan, and X. Qu, “Improved chicken swarm optimization method for reentry trajectory optimization,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 8135274, 13 pages, 2018.

[11] J. R. Mansell and M. J. Grant, “Adaptive continuation strategy for indirect hypersonic trajectory optimization,” *Journal of Spacecraft and Rockets*, vol. 55, no. 4, pp. 818–828, 2018.

[12] Z. Wang, “Optimal trajectories and normal load analysis of hypersonic glide vehicles via convex optimization,” *Aerospace Science and Technology*, vol. 87, pp. 357–368, 2019.

[13] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[14] A. Prabhakar, J. Fisher, and R. Bhattacharya, “Polynomial chaos-based analysis of probabilistic uncertainty in hypersonic flight dynamics,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 222–234, 2010.

[15] B. Tian, W. Fan, R. Su, and Q. Zong, “Real-time trajectory and attitude coordination control for reusable launch vehicle in reentry phase,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1639–1650, 2015.

[16] L. Zang, D. Lin, S. Chen, H. Wang, and Y. Ji, “An on-line guidance algorithm for high L/D hypersonic reentry vehicles,” *Aerospace Science and Technology*, vol. 89, pp. 150–162, 2019.

[17] Y.-Z. Luo and Z. Yang, “A review of uncertainty propagation in orbital mechanics,” *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, 2017.

[18] L. Janson, E. Schmerling, and M. Pavone, “Monte Carlo motion planning for robot trajectory optimization under uncertainty,” in *Robotics Research*, pp. 343–361, Springer, Cham, Switzerland, 2018.

[19] N. Wiener, “The homogeneous chaos,” *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.

[20] D. Xiu and G. E. Karniadakis, “The Wiener-Askey polynomial chaos for stochastic differential equations,” *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.

[21] F. Xiong, S. Chen, and Y. Xiong, “Dynamic system uncertainty propagation using polynomial chaos,” *Chinese Journal of Aeronautics*, vol. 27, no. 5, pp. 1156–1170, 2014.

[22] G. C. Cottrill and F. G. Harmon, “Hybrid Gauss pseudospectral and generalized polynomial chaos algorithm to solve stochastic trajectory optimization problems,” in *Proceedings of the AIAA Guidance, Navigation, & Control Conference*, Minneapolis, MN, USA, August 2013.

[23] K. Okamoto and T. Tsuchiya, “Optimal aircraft control in stochastic severe weather conditions,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 1, pp. 77–85, 2016.

[24] G. I. Boutselis, Y. Pan, G. D. La Tore, and E. A. Theodorou, “Stochastic trajectory optimization for mechanical systems with parametric uncertainties,” 2017, https://arxiv.org/abs/1705.05506.

[25] J. Fisher and R. Bhattacharya, “Optimal trajectory generation with probabilistic system uncertainty using polynomial chaos,” *Journal of Dynamic Systems Measurement and Control—Transactions of The ASME*, vol. 133, no. 1, 2011.

[26] X. Li, P. B. Nair, Z. Zhang, L. Gao, and C. Gao, “Aircraft robust trajectory optimization using nonintrusive polynomial chaos,” *Journal of Aircraft*, vol. 51, no. 5, pp. 1592–1603, 2014.

[27] Y. Huang and H. Li, “Reliability-based trajectory optimization using nonintrusive polynomial chaos for Mars entry mission,” *Advances in Space Research*, vol. 61, no. 11, pp. 2854–2869, 2018.

[28] Y. Huang, H. Li, X. Du, and X. He, “Mars entry trajectory robust optimization based on evidence under epistemic uncertainty,” *Acta Astronautica*, vol. 163, pp. 225–237, 2019.

[29] F. Wang, S. Yang, F. Xiong, Q. Lin, and J. Song, “Robust trajectory optimization using polynomial chaos and convex optimization,” *Aerospace Science and Technology*, vol. 92, pp. 314–325, 2019.

[30] K. Cheng and Z. Lu, “Adaptive sparse polynomial chaos expansions for global sensitivity analysis based on support vector regression,” *Computers & Structures*, vol. 194, pp. 86–96, 2018.

[31] A. F. Cortesi, G. Jannoun, and P. M. Congedo, “Kriging-sparse polynomial dimensional decomposition surrogate model with adaptive refinement,” *Journal of Computational Physics*, vol. 380, pp. 212–242, 2019.

[32] J. Kennedy and R. Eberhart, “Particle swarm optimization,”vol. 4, pp. 1942–1948, in *Proceedings of the ICNN’95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, WA, Australia, November 1995.

[33] Y. Zhou and X. Liu, “Control parameterization-based adaptive particle swarm approach for solving chemical dynamic optimization problems,” *Chemical Engineering & Technology*, vol. 37, no. 4, pp. 692–702, 2014.

[34] T. L. Saaty, “How to make a decision: the analytic hierarchy process,” *European Journal of Operational Research*, vol. 48, no. 1, pp. 9–26, 1990.

[35] Y. Zhuang and H. Huang, “Time-optimal trajectory planning for underactuated spacecraft using a hybrid particle swarm optimization algorithm,” *Acta Astronautica*, vol. 94, no. 2, pp. 690–698, 2014.

[36] C. Dong and Y. Cai, “Reentry trajectory optimization for hypersonic glide vehicle with flexible initial conditions,” *Journal of Aerospace Engineering*, vol. 30, no. 5, Article ID 04017033, 2017.

[37] S. H. Lee and W. Chen, “A comparative study of uncertainty propagation methods for black-box-type problems,” *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, pp. 239–253, 2009.