

Research Article

Minimum-Fuel Ascent of Hypersonic Vehicle considering Control Constraint Using the Improved Pigeon-Inspired Optimization Algorithm

Jiahao He, Yanbin Liu , Shuanglin Li, and Yue Tang

College of Astronautics, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, China

Correspondence should be addressed to Yanbin Liu; nuaa_liuyanbin@139.com

Received 13 January 2020; Revised 18 March 2020; Accepted 19 May 2020; Published 8 June 2020

Academic Editor: Antonio Viviani

Copyright © 2020 Jiahao He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Trajectory optimization problem for hypersonic vehicles has long been recognized as a difficult problem. This paper brings control constraints into the trajectory optimization to make the optimal trajectory meet the requirements of control performance. The strong nonlinear characteristic of the ascent phase aerodynamics makes the trajectory optimization problem difficult to be solved by the optimal control theory. A trajectory optimization algorithm based on the improved pigeon-inspired optimization (PIO) algorithm is proposed to solve the complex trajectory optimization problem under multiple constraints. To overcome the obstacle of premature convergence and deceptiveness, the evolutionary strategy of qubit in quantum evolutionary algorithm (QEA) is introduced into the PIO to maintain population diversity and judge the optimal solution. To handle constraints, the penalty function is used to construct the fitness function. The optimal ascent trajectory is obtained by utilizing the improved PIO algorithm. Then, the trajectory inverse algorithm is used to verify the feasibility of the optimal trajectory to ensure that a feasible optimal trajectory is obtained. The comparison results show that the proposed algorithm outperforms particle swarm optimization (PSO) and standard PIO on trajectory optimization. Meanwhile, the simulation result shows that the performance of the optimal ascent trajectory with control constraints is improved and the trajectory is feasible. Therefore, the method is potentially feasible for solving the ascent trajectory optimization problem under control constraint for hypersonic vehicles.

1. Introduction

The hypersonic vehicle has received wide attention as it has high speed and large flight range. As an essential part, trajectory optimization for the hypersonic vehicle can help the design of the vehicle shape and a flight plan. The ascent trajectory optimization, due to its highly nonlinear dynamics and strong constraints, becomes a hot and difficult issue. Meanwhile, the reduction of fuel consumption in the ascent phase can help the hypersonic vehicle to increase the cruising distance.

A great deal of research has been done on the optimization of the hypersonic vehicle's trajectory, and a lot of new optimization methods have been proposed [1–3]. Kumar et al. presented a trajectory optimization of an aerodynamically controlled hypersonic boost-glide class of flight vehicles [4]. Xie et al. proposed a multiobjective gliding trajectory

optimization scheme for a hypersonic vehicle with complicated constraints [5]. Chai et al. have done a lot of research on trajectory optimization and optimal control problem [6–8]; they considered the highly constrained trajectory optimization problems, applied a specific multiple-shooting discretization technique with the newest NSGA-III optimization algorithm, and constructed a new evolutionary optimal control solver to address the multiobjective trajectory planning problem [6]. Fu et al. solved the ascent trajectory optimization problem for hypersonic vehicles with the improved chicken swarm optimization (ICSO) algorithm [9]. Cheng et al. presented an iterative convex programming algorithm for the complex ascent trajectory planning problem, transcribed the dynamic equations into linearized algebraic equality constraints with a given initial guess based on the Newton–Kantorovich/Pseudospectral (N–K/PS) approach, and formulated the ascent trajectory planning problem as a

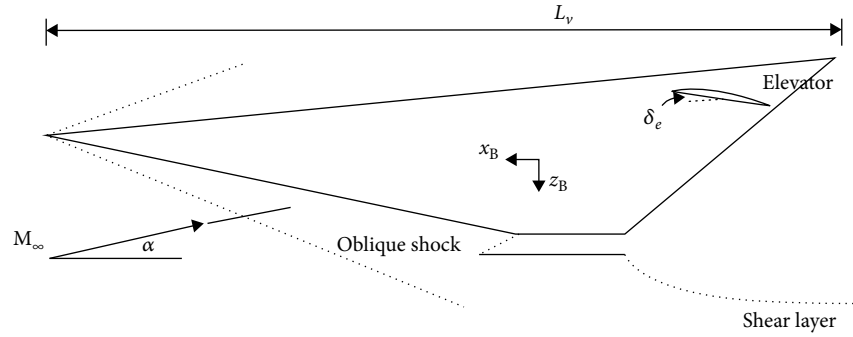


FIGURE 1: Geometry of the hypersonic vehicle.

convex programming problem [10]. Mahmoud et al. described the ascent and descent optimal trajectories with two different dynamics in each stage and the relation between the stages under the constraints of each phase [11].

However, the hypersonic vehicle is a strong coupling system; its plant performance, trajectory performance, and control performance are not independent but affect each other. Due to the limitation of control performance, the optimal trajectory that only considers trajectory performance may not be realized, so it is necessary to add control constraints to trajectory optimization.

At present, the numerical method is widely studied and applied in trajectory optimization [12–14], and it can be divided into indirect method and direct method. The indirect method uses optimal-control theory to obtain an analytical solution to the trajectory optimization problem. The merit of this method is that it can obtain an accurate analytical solution by the first-order necessary optimal condition; meanwhile, the initial costate is very difficult to guess. The direct method first transforms the trajectory optimization problem into nonlinear programming problem, then solves the nonlinear programming problem. Compared with the indirect method, the direct method has the advantages of low precision of the initial guess and simple derivation process. Recently, bioinspired heuristic algorithms have received widespread attention in the field of direct methods for solving the nonlinear programming problem. The advantage of the bioinspired heuristic algorithm is that the optimal solution is usually the global optimal solution, and these algorithms have strong robustness.

Due to the effectiveness and ease of implementation, more and more optimization algorithms based on swarm intelligence are proposed and researched nowadays [15–17]. The PIO algorithm performed better in a lot of popular benchmark optimization problems compared with DE [18] and PSO method and has proved effective in many areas. For example, three-dimensional path planning [19] controls parameter optimization [20] and predicts 3D protein structure [21]. Therefore, this paper chooses the PIO algorithm as the root-finding algorithm to solve the nonlinear programming problem.

Although the PIO algorithm is able to converge quickly, it is often inaccurate in dealing with complex optimization problems and easily falls into local optimization owing to premature convergence. Premature convergence mostly originates

from a loss of diversity and deceptiveness. The lack of diversity means that the difference between all solution candidates is small, which weakens the exploration. Meanwhile, a deceptive direction of convergence forestalls the exploration.

To overcome these obstacles and obtain better solutions, this paper introduces quantum representation and quantum rotation gate in the quantum evolutionary algorithm (QEA) to improve the PIO algorithm. The QEA is a probabilistic evolutionary algorithm that integrates concepts from quantum computing for robust search [22]. The QEA uses a qubit as the probabilistic representation, which represents a linear superposition of binary solutions [23]. The quantum rotation gate is the evolutionary strategy of qubit in the QEA [24], which is adopted as a mutation operator to update the pairs of probability amplitudes toward the one with the best fitness. In this paper, the position of pigeons is represented by qubit; the current best pigeon is considered to be a linear superposition of positive state and deceptive state. Other pigeons make their own judgment by observation. If the current optimal solution still exists after iteration, the deceptive probability amplitude will decrease. In this way, the accuracy of the PIO algorithm is improved.

The rest of this paper is structured as follows. Section 2 briefly introduces the hypersonic vehicle model adopted in this paper; Section 3 establishes the optimization problem; constraints and performance index in trajectory optimization are determined; Section 4 proposes a new trajectory optimization based on the improved PIO algorithm and trajectory inverse algorithm in allusion to optimize ascent trajectory under complex constraints. Section 5 shows the simulation results. In this section, the comparison results of the proposed algorithm, PIO, and PSO on trajectory optimization are given. Meanwhile, the optimal trajectories with and without control constraints are compared; then, the control variables are obtained by using the trajectory inverse algorithm to determine whether the optimal trajectory is feasible. Finally, we conclude in the last section.

2. Vehicle Model

The hypersonic vehicle model used in this paper was developed by Bolender and Doman [25] and Parker et al. [26] as an attempt to extend earlier work done by Chavez and Schmidt [27]. The basic geometry of the vehicle model is shown in Figure 1, which contains only longitudinal dynamics

(in the figure, α is the angle of attack; L_v is vehicle length; δ_e is elevator deflection; and M_∞ is freestream Mach number). A detailed explanation of the model derivation is given in [25]. Rather than using the relatively simple approach of Newtonian Impact Theory [27], the model was derived using a compressible flow theory. The combination of oblique shock wave and Prandtl-Meyer flow theory is used to determine the pressure within the range of possible angles of attack and structural bending conditions. The engine model is a scramjet taken directly from the paper by Chavez and Schmidt [27]. Since the angle of attack plays an important role in determining the characteristics of the flow characteristics into the scramjet, the thrust becomes very dependent on it.

2.1. Dynamic Equation. The dynamic equations of motion for the longitudinal dynamics of the hypersonic vehicle are as follows:

$$\begin{cases} \dot{v} = \frac{T \cos \alpha - D}{m} - g \sin(\theta - \alpha), \\ \dot{\alpha} = q_p - \frac{T \sin \alpha + L}{mv} + \left(\frac{g}{v} - \frac{v}{R_e + h} \right) \cos \gamma, \\ \dot{h} = v \sin(\theta - \alpha), \\ \dot{\theta} = q_p, \\ \dot{q}_p = \frac{M_y}{I_y}, \end{cases} \quad (1)$$

where v is the velocity; h is the altitude; α is the angle of attack; θ is the pitch angle; q_p is the pitch rate; γ is the flight path angle; I_y is the moment of inertia; g is the gravitational acceleration; R_e is the earth radius; m is the mass of the vehicle; and T , L , D , and M_y are thrust, lift, drag, and moment, respectively, expressed as

$$\begin{cases} L = 0.5\rho v^2 s C_L, \\ D = 0.5\rho v^2 s C_D, \\ T = 0.5\rho v^2 A_e C_T, \\ M_y = 0.5\rho v^2 \bar{c} C_M, \end{cases} \quad (2)$$

where v is the velocity; s is the reference area; A_e is the air inlet area; ρ is the atmospheric density; \bar{c} is the mean aerodynamic chord; and C_L , C_D , C_T , and C_M are lift coefficient, drag coefficient, thrust coefficient, and moment coefficient, respectively. The coefficients determined by the angle of attack α and Mach number are as follows:

$$\begin{cases} C_L = C_{L_0} + C_{L_\alpha} \times \alpha + C_{L_{\delta_e}} \times \delta_e, \\ C_D = C_{D_0} + C_{D_\alpha} \times \alpha + C_{D_{\delta_e}} \times \delta_e, \\ C_T = C_{T_0} + C_{T_\alpha} \times \alpha + C_{T_\phi} \times \phi, \\ C_M = C_{M_0} + C_{M_\alpha} \times \alpha + C_{M_{\delta_e}} \times \delta_e, \end{cases} \quad (3)$$

where C_{L_0} , C_{L_α} , $C_{L_{\delta_e}}$, C_{D_0} , C_{D_α} , $C_{D_{\delta_e}}$, C_{T_0} , C_{T_α} , C_{T_ϕ} , C_{M_0} , C_{M_α} , and $C_{M_{\delta_e}}$ are determined by experimental and simulation data; α is the angle of attack; δ_e is elevator deflection; and ϕ is fuel equivalent ratio. The exact analytical expressions of forces and moments are described in detail in [25].

2.2. Fuel Consumption. In the ascent phase of the hypersonic vehicle, fuel mass is time-varying. Fuel consumption is calculated to determine the fuel mass at each moment. Fuel consumption per unit time of the vehicle is

$$\dot{m} = -\frac{T}{g_0 \cdot I_{sp}}, \quad (4)$$

where I_{sp} is the specific impulse of the engine; T is thrust; and g_0 is the normal acceleration of gravity. The fuel consumption between two adjacent points is

$$dm = \dot{m} \cdot \frac{\Delta v}{a}, \quad (5)$$

where a is the acceleration of vehicle and Δv is the velocity difference between two adjacent points.

3. Optimization Problem Establishment

The hypersonic vehicle model is a complex dynamic system owing to the coupling between subsystems. Therefore, the trajectory performance and control performance are also coupled with each other during the ascent phase. The fully integrated system model can be more simply expressed as

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad (6)$$

where $\mathbf{x}(t)$ is the vehicle states during the ascent phase and $\mathbf{u}(t)$ is the control input. In this way, the hypersonic vehicle ascent trajectory becomes the multidisciplinary system model given in Equation (6) [28]. Hence, the control performance constraint has to be taken into account when optimizing the ascent trajectory.

3.1. Optimization Problem Formulation. The optimal control problem can be simply described as seeking the optimal control variable and the corresponding state variable, so that the performance index achieves the extreme value under the constraints of equality and inequality. The optimization problem of the multidisciplinary system in Equation (6) is as follows:

$$\begin{cases} \min & J = \Phi(\mathbf{x}(t_f), t_f), \\ \text{s.t.} & \begin{cases} \dot{\mathbf{x}}(t) = F(\mathbf{x}(t), \mathbf{u}(t), t), \\ C(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0, \\ \varphi(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f) = 0, \end{cases} \end{cases} \quad (7)$$

where t_0 and t_f are the initial time and the terminal time, respectively; $\mathbf{x}(t)$ is the state variable; $\mathbf{u}(t)$ is input control variable; $J = \Phi(\mathbf{x}(t_f), t_f)$ is performance index which is obtained by the objective function; $\varphi(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f) =$

0 is given by boundary value constraint condition, which is converted to $C(x(t), u(t), t) \leq 0$ by constraints during the flight.

3.2. Construction of Performance Index. The mission in the ascent phase is to make the vehicle move to the cruise window safely. The reduction of fuel consumption can help the vehicle to increase the cruise distance. In this paper, the maximum residual mass at the end of the ascent phase is set as the optimal objective function.

$$\min J_{\text{opt}} = -m_f, \quad (8)$$

where J_{opt} is the performance index of the optimal trajectory in this paper; m_f is the mass of the vehicle at the trajectory termination point, which is obtained by integrating Equation (4).

To introduce equality constraints and inequality constraints into the optimization problem, this paper uses penalty function, which is usually used to solve constrained optimization problems, to construct objective function. The basic idea is that, according to the characteristics of constraint conditions, it can be converted into some punishment function and added to the objective function, to transform the constrained optimization problem into an unconstrained optimization problem to be solved, and the optimal solution can be obtained by solving the unconstrained optimization problem. The penalty function $\bar{P}(x)$ constructed in this paper is as follows:

$$\bar{P}(x) = \sum h_i^2(x) + \sum [\min \{0, g_i(x)\}]^2, \quad (9)$$

where $h(x)$ and $g(x)$ represent equality constraints and inequality constraints, respectively, and x is the state variable.

Each term in the penalty function is multiplied by a certain penalty parameter and then combined with the original objective function to obtain the augmented objective function.

$$\min J_{\text{opt}} = -m_f + \sigma \bar{P}(x), \quad (10)$$

where σ is the penalty matrix and x is the state variable.

3.3. Constraints for Ascent Phase. In this part, various constraints included in $h(x)$ and $g(x)$ are given. In addition to the common constraint, such as path constraints and terminal constraints, the impact of control constraints and state constraints on trajectory will also be taken into consideration.

3.3.1. Path Constraints. During the flight, intense friction with the atmosphere will produce high temperatures. Therefore, the heat flux constraint must be considered to prevent the excessive surface temperature of the vehicle. The heat flux constraint is as follows:

$$\dot{Q} = C\sqrt{\rho}v^{3.15} \leq \dot{Q}_{\text{max}}, \quad (11)$$

where C is constant; ρ is air density; and v is the velocity. Dynamic pressure generated by high-speed flight provides

aerodynamic force and control torque to adjust attitude. However, if the dynamic pressure exceeds the limit, it will have a great impact on the vehicle. The dynamic pressure constraint is as follows:

$$q_{\text{min}} \leq q = \frac{1}{2}\rho v^2 \leq q_{\text{max}}, \quad (12)$$

where ρ is air density and v is the velocity. In order to ensure the structural stability of the vehicle, the overload constraint should be considered in the ascent trajectory optimization, and the overload constraint is as follows:

$$n = \frac{\sqrt{L^2 + D^2}}{mg} \leq n_{\text{max}}, \quad (13)$$

where g is the gravitational acceleration; m is the mass of the vehicle; and L and D are lift and drag, respectively.

3.3.2. Terminal Constraints. The terminal constraint is related to flight missions. In this paper, the flight height, the velocity, and the flight path angle are required to meet specific constraint as follows:

$$\begin{aligned} v_t &= v_f + \Delta v, \\ h_t &= h_f + \Delta h, \\ \gamma_t &= \gamma_f + \Delta \gamma, \end{aligned} \quad (14)$$

where v_t , h_t , and γ_t are actual cruising state; v_f , h_f , and γ_f are the predetermined cruising state; and Δv , Δh , and $\Delta \gamma$ are acceptable error.

3.3.3. Control Constraints and State Constraints. Although the ascent trajectory without considering control constraints and state constraints is theoretically feasible, in practical application, due to the influence of ascending requirement and manoeuvring performance, it is necessary to control the state variables within a certain range and consider the control feasibility of the vehicle. As mentioned before, the control performance is affecting the ascent trajectory. Consequently, control constraints and state constraints are needed in trajectory optimization.

In this paper, for the state constraint, the constraints of the flight altitude, the velocity, and the flight path angle are granted.

$$\begin{aligned} v_{\text{min}} &\leq v \leq v_{\text{max}}, \\ h_{\text{min}} &\leq h \leq h_{\text{max}}, \\ \gamma_{\text{min}} &\leq \gamma \leq \gamma_{\text{max}}, \end{aligned} \quad (15)$$

where v_{min} and v_{max} are the lower and upper limits of the velocity; h_{min} and h_{max} are the lower and upper limits of the flight altitude; and γ_{min} and γ_{max} are the lower and upper limits of the flight path angle.

For the control constraint, the constraints of the angle of attack and the elevator deflection are granted.

$$\begin{aligned} \alpha_{\min} &\leq \alpha \leq \alpha_{\max}, \\ \delta_{e_{\min}} &\leq \delta_e \leq \delta_{e_{\max}}, \end{aligned} \quad (16)$$

where α_{\min} and α_{\max} are the lower and upper limits of the angle of attack and $\delta_{e_{\min}}$ and $\delta_{e_{\max}}$ are the lower and upper limits of elevator deflection.

Moreover, considering the attitude stability of the vehicle, the rate of change of the flight path angle cannot be too high or too low during the ascent phase. In this paper, the absolute value of the rate of change of the flight path angle never exceeds 0.002 rad/s.

4. Trajectory Optimization Algorithm

As mentioned in the previous section, trajectory performance and control performance are both considered in trajectory optimization. As the constraints increase, the complexity of trajectory optimization increases. An effective trajectory optimization algorithm is needed to solve the trajectory optimization problem under complex constraints.

The optimization can be divided into two steps: (1) identifying feasible designs and (2) identifying the optimal design from the set of feasible candidates [29]. This paper proposes an ascent trajectory optimization algorithm to solve a complex optimization problem under multiple constraints, as shown in Figure 2. The algorithm first establishes the constraints and fitness function of the ascent trajectory, optimizes the ascent trajectory using the optimization algorithm, and then performs a feasibility analysis on the optimal trajectory. When the trajectory is not feasible, the penalty value of fitness function is adjusted until the feasible optimal ascent trajectory is obtained. The specific methods included in the algorithm are introduced, respectively, in this section.

4.1. The Improved PIO Algorithm. In the global optimization problem, especially for multimodal problems, when the PIO algorithm cannot generate new children, it is easy for the algorithm to converge to local optimization prematurely. The main reason for premature convergence is the low diversity of the algorithm. Also, in the convergence stage, when the population aggregates at a certain point, even if this is not the correct global optimal solution, it will be regarded as the global optimal solution. Therefore, to overcome these weaknesses in the PIO algorithm, this paper represents the position of pigeons by qubit, the current best pigeon is considered to be a linear superposition of positive state and deceptive state. The other pigeon makes its own judgment by observation. Meanwhile, the quantum rotation gate is adopted in the PIO algorithm as a mutation operator to enhance positive probability.

4.1.1. Map/Compass Operator. In the basic PIO algorithm, the spatial position of the pigeon is the possible solution to the optimization problem. The position of the pigeon's home is the global optimal solution. The fitness of the pigeon is the

value of the objective function. The position \mathbf{X}_i and the speed \mathbf{V}_i of the i th pigeon are

$$\begin{aligned} \mathbf{X}_i^T &= [X_{i,1}, X_{i,2}, \dots, X_{i,n}], \quad i = 1, 2, \dots, N_p, \\ \mathbf{V}_i^T &= [V_{i,1}, V_{i,2}, \dots, V_{i,n}], \quad i = 1, 2, \dots, N_p, \end{aligned} \quad (17)$$

where N_p is the population size.

For the sake of the optimal solution to meet the multiple constraints presented in the previous section, strong constraints need to be applied to the value range of each possible solution element; i.e., the position and speed constraints of the pigeon are as follows:

$$\begin{cases} \mathbf{X}_{\text{low}} \leq \mathbf{X}_i^{(i)} \leq \mathbf{X}_{\text{up}}, \\ \mathbf{V}_{\text{low}} \leq \mathbf{V}_i^{(i)} \leq \mathbf{V}_{\text{up}}, \end{cases} \quad (18)$$

where $i = 1, 2, \dots, N_p$; \mathbf{X}_{low} and \mathbf{X}_{up} are the lower and upper limits of each element in the position, respectively; and \mathbf{V}_{low} and \mathbf{V}_{up} are the lower and upper limits of each element in the speed, respectively.

In the map compass operator, each individual in the population updates its speed and position through geomagnetic, solar height information and the optimal information in the population.

$$\mathbf{V}_i(t) = \mathbf{V}_i(t-1) \cdot e^{-Rt} + \text{rand} \cdot (\mathbf{X}_g - \mathbf{X}_i(t-1)), \quad (19)$$

$$\mathbf{X}_i(t) = \mathbf{X}_i(t-1) + \mathbf{V}_i(t), \quad (20)$$

where t is the current number of iterations, R is the map compass factor, \mathbf{X}_g is the global optimal solution in the current population, and rand is the uniform random value between the [0,1] regions. According to the map compass operator, each pigeon in the population adjusts its direction according to Equation (19).

The map compass operator of the PIO algorithm is improved in this paper, so as to solve the problem that the basic PIO algorithm falls into local optimal due to premature convergence.

The improved PIO algorithm adopts qubit to describe the current state of the pigeon. To maintain population diversity, the position state of the pigeon is obtained according to Monte Carlo random simulation:

$$\mathbf{X}_i(t) = \mathbf{P}_i \mathbf{P}_g(t) \pm \frac{L}{2} \ln \left(\frac{1}{u} \right), \quad (21)$$

where

$$\mathbf{P}_i \mathbf{P}_g(t) = f(t) \times \mathbf{P}_i(t) + (1 - f(t)) \times \mathbf{P}_g(t), \quad (22)$$

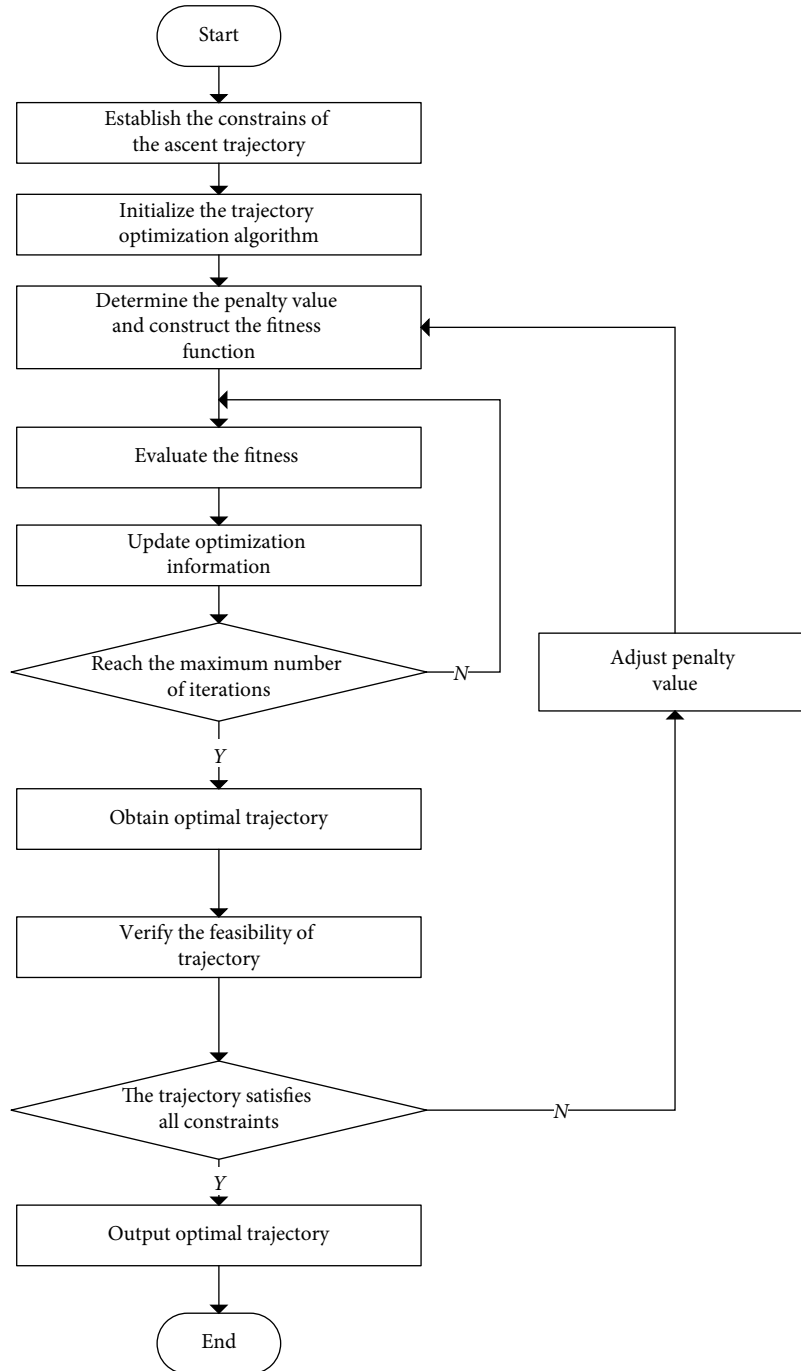


FIGURE 2: Ascent trajectory optimization algorithm.

where t is the number of iterations; u and f are uniformly distributed random numbers on $[0,1]$; $\mathbf{P}_i(t)$ is the historical optimal position at t th iteration; $\mathbf{P}_g(t)$ is the global optimal position at the t th iteration; and L is defined as follows:

$$L = 2\omega(t)|m_{\text{best}}(t) - \mathbf{X}_i(t)|, \quad (23)$$

where $\omega(t)$ is inertia weight which has great influence on the convergence of the algorithm; $m_{\text{best}}(t)$ is the average optimal

position of all pigeons in the population at t th iteration and $\omega(t)$ and $m_{\text{best}}(t)$ are defined as follows:

$$\omega(t) = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) \times \frac{t}{T}, \quad (24)$$

$$m_{\text{best}}(t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{P}_i(t),$$

where N_p is the population size; ω_{max} and ω_{min} are lower and upper limits of inertia weight; and T is the maximum number of iterations.

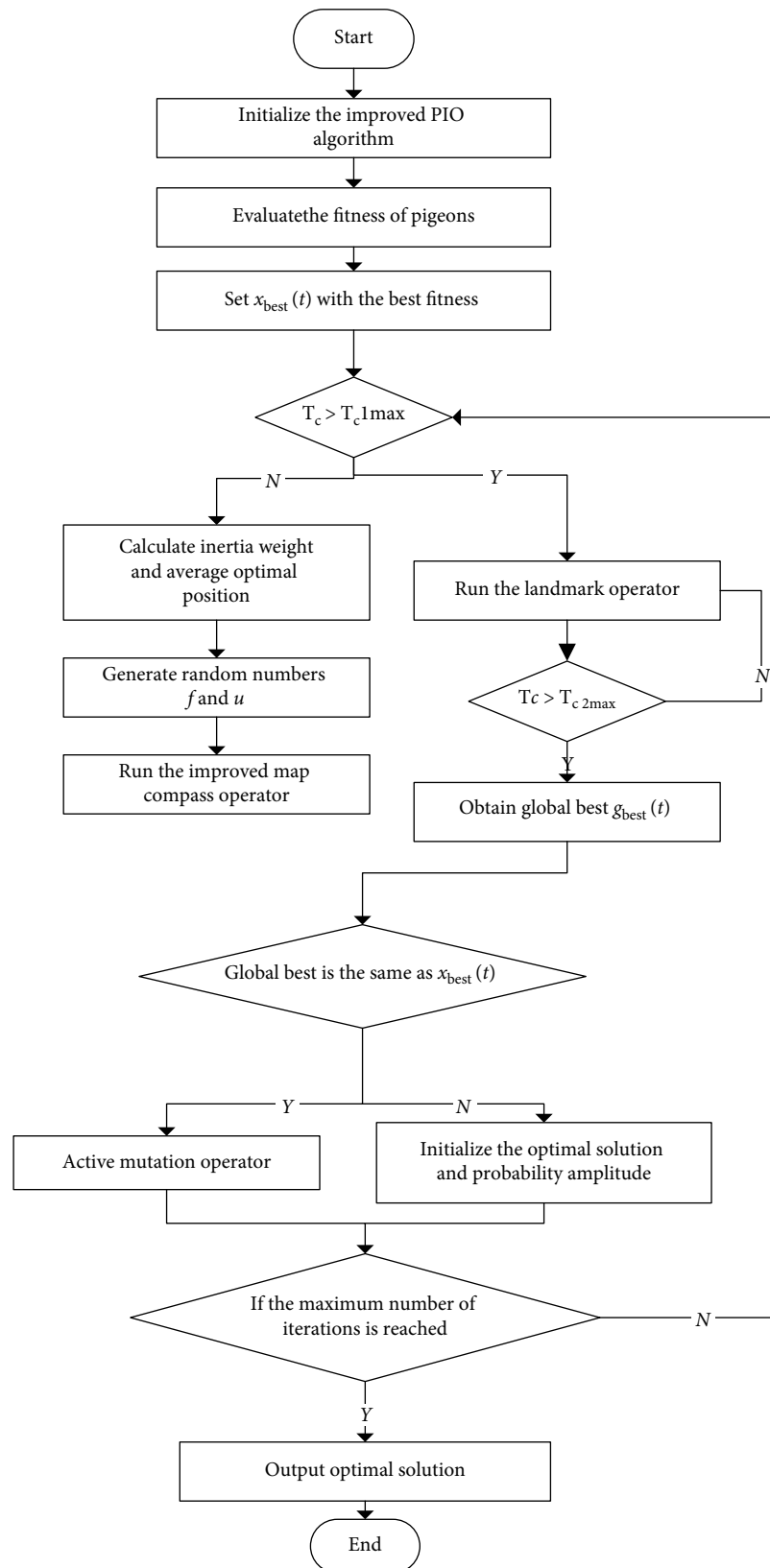


FIGURE 3: The procedure of the improved PIO algorithm.

TABLE 1: Test function optimization results.

Test function	D	Algorithm	Optimal value	Time (s)
Ackley	5	PSO	7.1431E-04	0.11
		PIO	0.0137	0.12
		The improved PIO	5.0849E-15	0.09
Rastrigin	5	PSO	5.7730E-06	0.13
		PIO	0.0121	0.12
		The improved PIO	0	0.09
Rosenbrock	2	PSO	1.0878E-09	0.11
		PIO	1.0900E-08	0.12
		The improved PIO	7.9993E-33	0.09

Hence, the improved PIO algorithm updates pigeons' position according to the following principle:

$$X_i(t+1) = \begin{cases} \mathbf{P}_i \mathbf{P}_g(t) + \omega(t) \times |\mathbf{m}_{\text{best}}(t) - X_i(t)| \times \ln \frac{1}{f(t)}, f(t) \geq 0.5, \\ \mathbf{P}_i \mathbf{P}_g(t) - \omega(t) \times |\mathbf{m}_{\text{best}}(t) - X_i(t)| \times \ln \frac{1}{f(t)}, f(t) < 0.5. \end{cases} \quad (25)$$

4.1.2. Landmark Operator. In the landmark operator, some individuals with better fitness are known as pigeons familiar with the landmarks, and the remaining pigeons may follow the elite pigeons or be abandoned by the population. In each iteration, pigeons are sorted according to the fitness, and half of the pigeons with poor fitness are eliminated. Then, the population centre point $\mathbf{C}(t)$ is selected as the reference direction of the remaining pigeons and the position of the pigeon is updated.

$$\begin{aligned} N_p(t) &= \frac{N_p(t-1)}{2}, \\ X_c(t) &= \frac{\sum X_i(t) \cdot \text{fitness}(X_i(t))}{N_p \sum \text{fitness}(X_i(t))}, \\ X_i(t) &= X_i(t-1) + \text{rand} \cdot (X_c - X_i(t-1)), \end{aligned} \quad (26)$$

where N_p is the population size; t is the number of iterations; X_c is the position of centre point; and $\text{fitness}()$ is a fitness function, which is constructed by the objective function in Equation (10), i.e.,

$$\text{fitness}(x) = J_{\text{opt}_{\min}} = -m_f + \sigma \bar{P}(x). \quad (27)$$

4.1.3. Judgment on Global Optimal Solution. To avoid the problem that the wrong global optimal solution was obtained by reason of the aggregation of pigeons, herein, this paper adopts the quantum rotating gate in the QEA [24] to solve this problem.

TABLE 2: Parameters of global optimization methods.

Parameters	PSO	PIO	The improved PIO
Population size (N_p)	100	100	100
Maximum number of iterations (T)	50	50	50
Map factor (R)	N/A	0.2	0.2
Learning factor ($[c_1, c_2]$)	[2,2]	N/A	N/A
Dimension (D)	10	10	10
Rotating angle ($^\circ$)	N/A	N/A	-11

Qubit chromosomes can be represented as follows:

$$q = \begin{bmatrix} \alpha_1 \cdots \alpha_i \cdots \alpha_m \\ \beta_1 \cdots \beta_i \cdots \beta_m \end{bmatrix}, \quad (28)$$

where q is qubit chromosomes and $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, \dots, m$, m , is the length of the chromosome. $|\alpha_i|^2$ and $|\beta_i|^2$ represent the probability that the state of the quantum bit is 0 and 1, respectively. By comparing the random number generated within the interval with $|\alpha_i|^2$, if the random number is larger than $|\alpha_i|^2$, the corresponding qubit of the chromosome is set as 1, otherwise set as 0 [30].

In the QEA, the probability amplitude update is calculated by

$$\alpha_i(t+1) = [\cos(\Delta\theta) - \sin(\Delta\theta)] \left[\frac{\alpha_i(t)}{\sqrt{1 - |\alpha_i(t)|^2}} \right], \quad (29)$$

where $\Delta + \theta$ is the rotation angle and t is the number of iterations. Unlike the quantum bit evolution strategy in QEA, it is used as a mutation operator to enhance positive probability.

The improved PIO algorithm will get a global optimal $x_{\text{best}}(t)$ after each iteration, but for the reasons mentioned above, this global optimal may not be the accurate global optimal. Probability amplitudes are initialized as $\alpha_i = \beta_i = \sqrt{2}/2$, $i = 1, 2, \dots, n$. If the current global optimal solution remains after iteration, run the mutation operator to increase α_i ; this means that $x_{\text{best}}(t)$ is more likely to be a global optimal

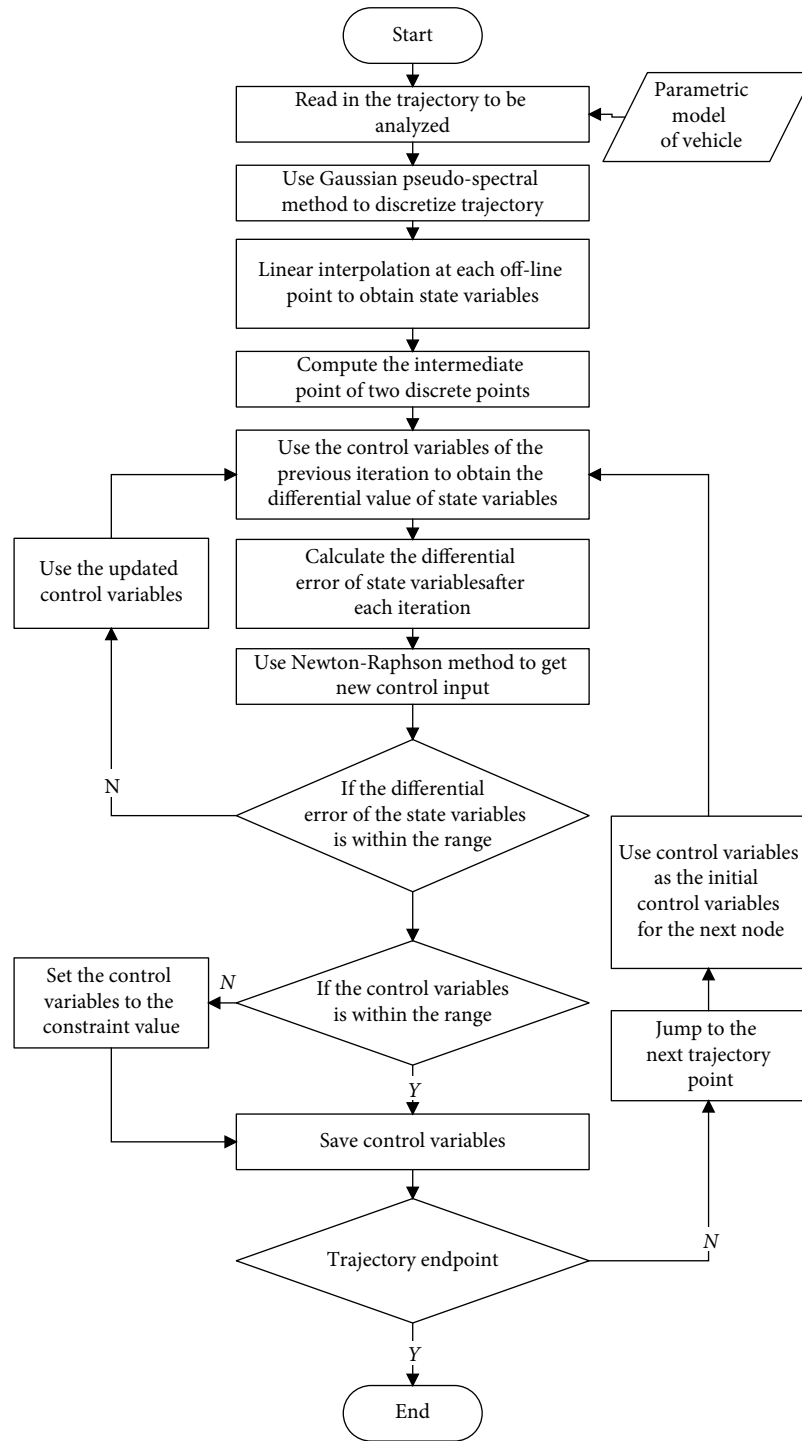


FIGURE 4: The procedure of the trajectory inverse algorithm.

solution. Otherwise, the probability amplitude is reset to initial values to maintain vigilance against the deceptiveness.

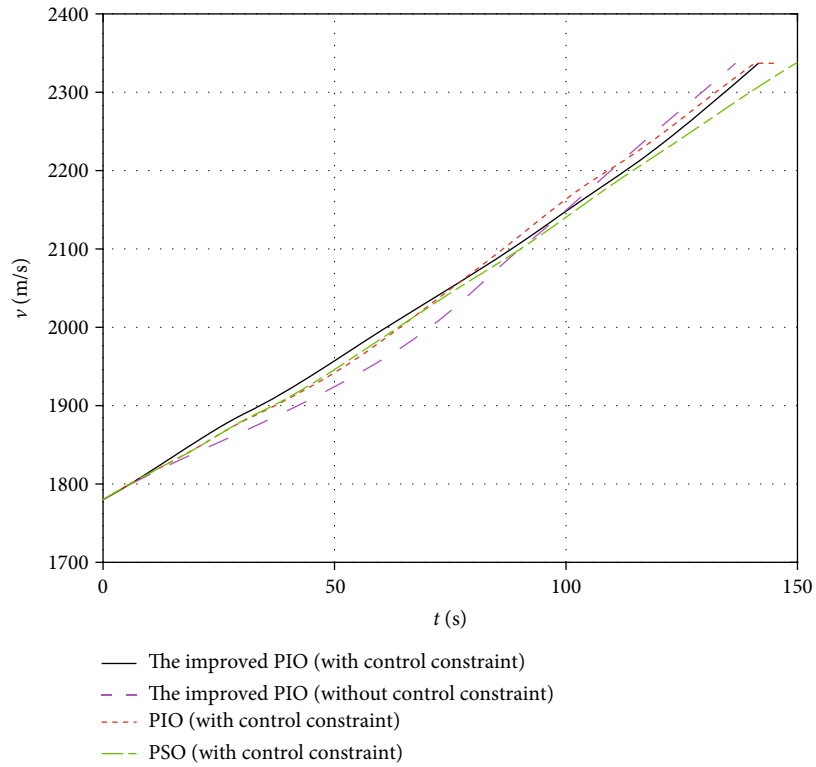
4.1.4. *The Procedure of the Improved PIO Algorithm.* The procedure of the improved PIO algorithm is as follows.

Step 1. Initialize population information and parameters of the improved PIO algorithm. Set a random speed and path for each pigeon.

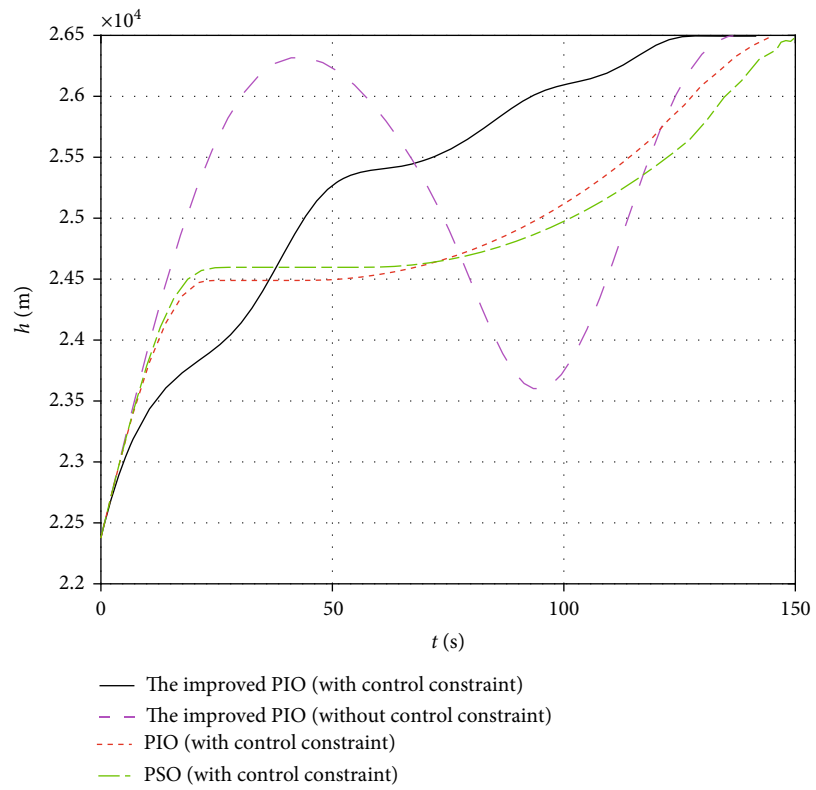
Step 2. Compare the fitness of each pigeon, set $x_{best}(t)$ with the current best fitness.

Step 3. Use the improved map compass operator to update; then use the landmark operator to convergence; obtain $g_{best}(t)$.

Step 4. Compare $g_{best}(t)$ with $x_{best}(t)$. If they are the same, go to (a); otherwise, go to (b).

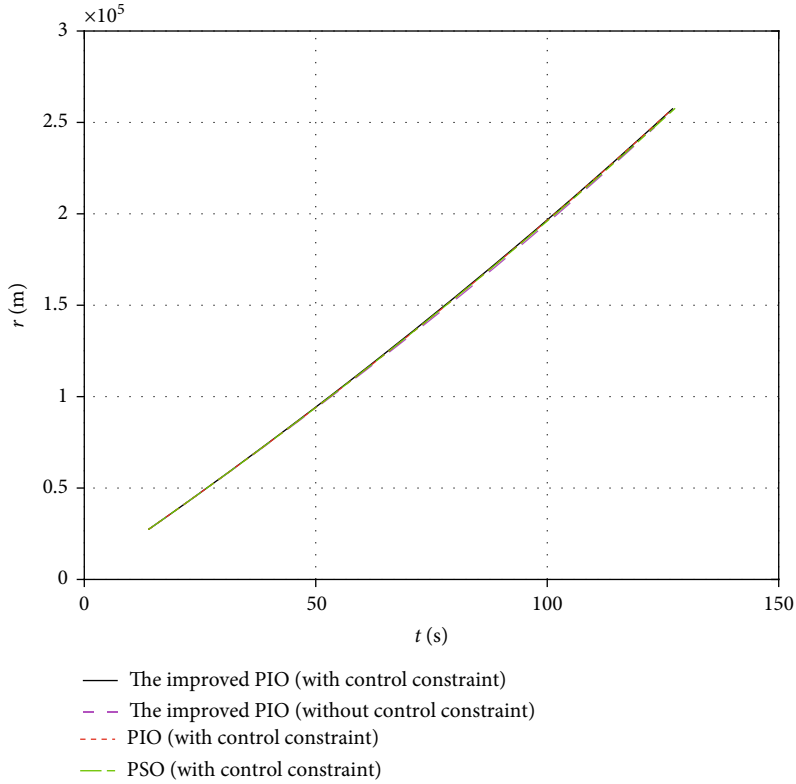


(a) Velocity

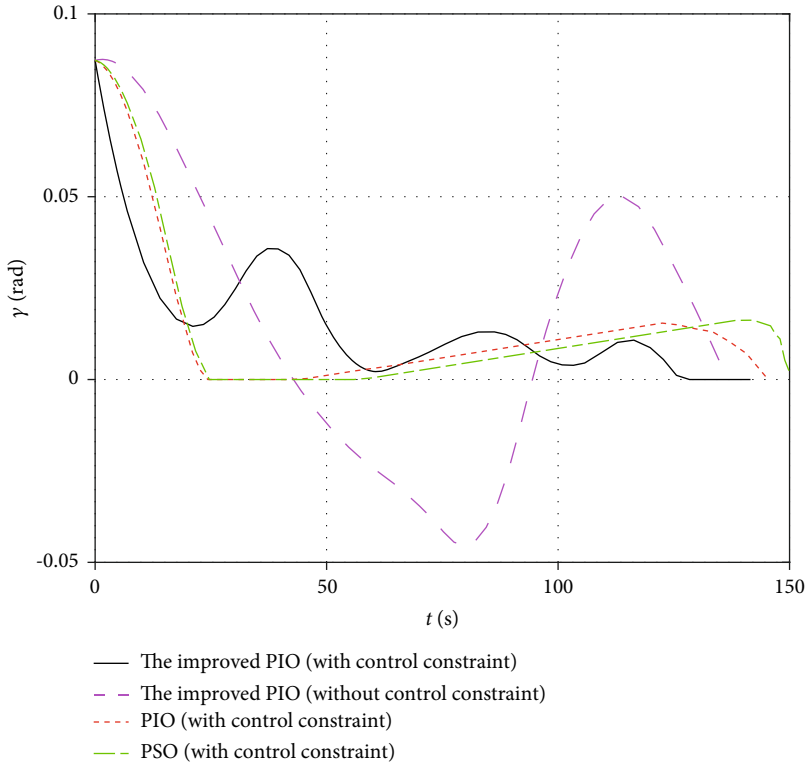


(b) Height

FIGURE 5: Continued.



(c) Flight range



(d) Flight path angle

FIGURE 5: Continued.

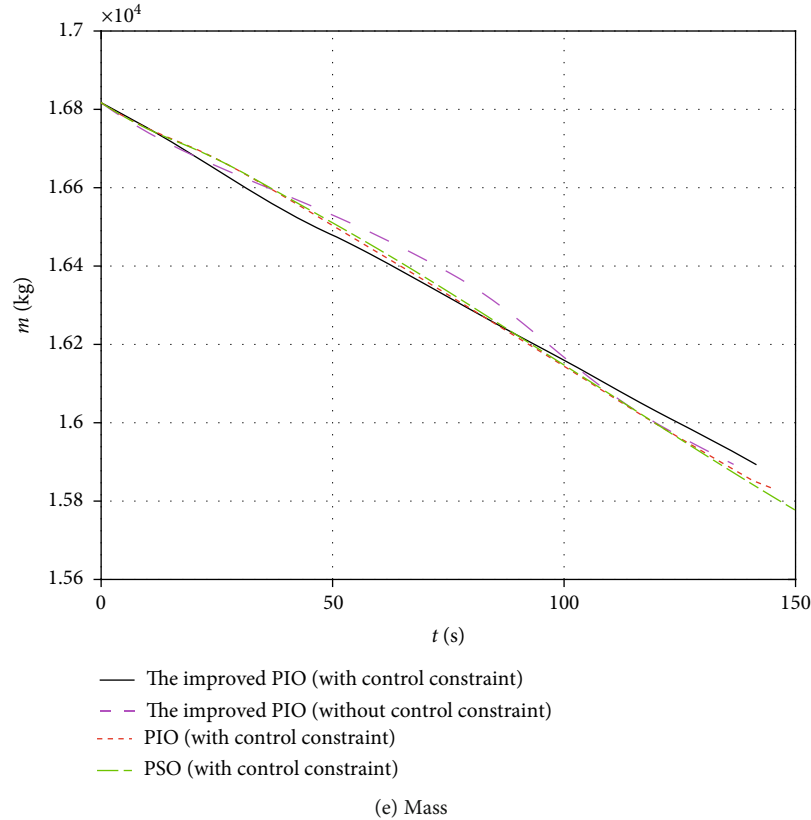


FIGURE 5: State variables of optimal ascent trajectories.

- (a) Conduct the mutation operator by Equation (29)
- (b) Conduct the mutation operator by Equation (29)

Step 5. If the maximum iteration number is reached, get global optimum $x_{\text{best}}(t)$. Otherwise, return to Step 3.

The above steps can be summarized in Figure 3.

To validate the improved PIO algorithm, three test functions are used to compare three algorithms, i.e., PSO [31], PIO [18], and the improved PIO. In order to rule out the effect of a few special values on the results, all experimental data in this section are the average of the values obtained from 40 independent runs of each algorithm. The optimized results for the test functions of these algorithms are listed in Table 1, where D is the dimension of the design variables. To avoid the difference in optimized results originating from the selection of control parameters, all equivalent control parameters are set to be the same (the specific parameters are listed in Table 2).

The improved PIO algorithm shows great potential for solving the multimodal optimization problem in terms of accuracy. The optimal value of the improved PIO algorithm is much smaller than that of other algorithms; the improved PIO algorithm also has the shortest running time. Therefore, the improved PIO algorithm outperforms PSO and PIO on benchmark functions.

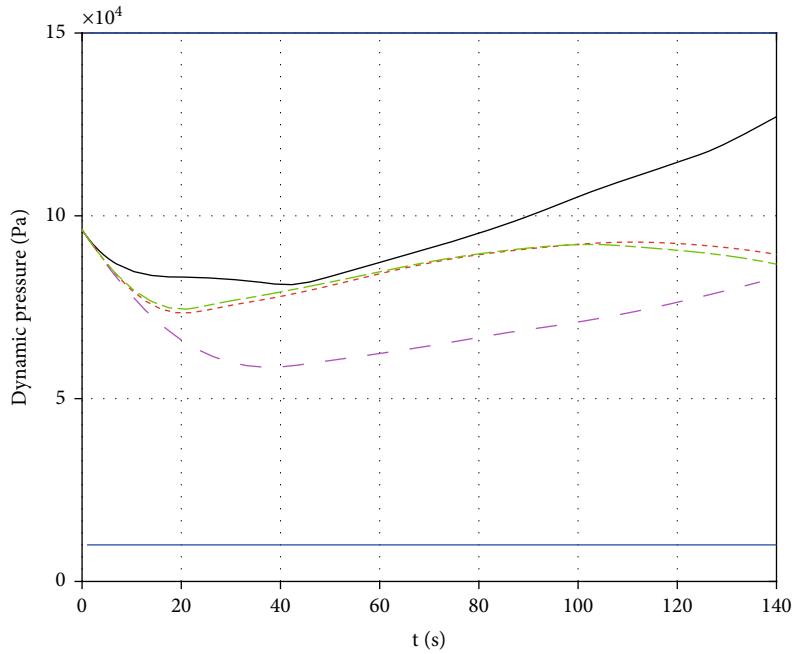
4.2. Trajectory Feasibility Analysis. To verify whether the control variables required by the trajectory conforms to constraints, this paper uses a trajectory inverse algorithm, which can solve the control variables needed according to the state variables at the current time node and the state variables at the next node.

For the vehicle, the known output is an ascent trajectory varying with time, and the algorithm can work out the value of the actual control variable required by the vehicle following a given trajectory. By analyzing the change curve of the control variable, it can be estimated whether the required control variable exceeds the physical limit when the vehicle follows the reference trajectory [32].

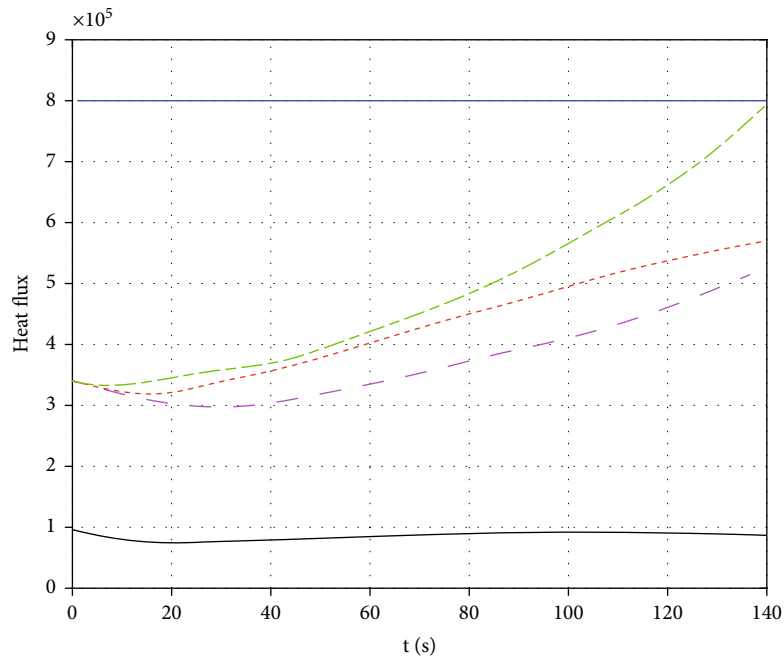
In the trajectory inverse algorithm, the dynamic system to be analyzed can be defined by state variable \mathbf{x} , control variable \mathbf{u} , and an output vector \mathbf{y} . The motion equation is shown as follows:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}), \\ \mathbf{x}(0) &= \mathbf{x}_0, \\ \mathbf{y} &= g(\mathbf{x}), \end{aligned} \quad (30)$$

where f and g represent any functions related to system variables; \mathbf{x}_0 is the initial value of the state variable; and \mathbf{y} is usually used to represent the known trajectory. The above



(a) Dynamic pressure



(b) Heat flux

FIGURE 6: Continued.

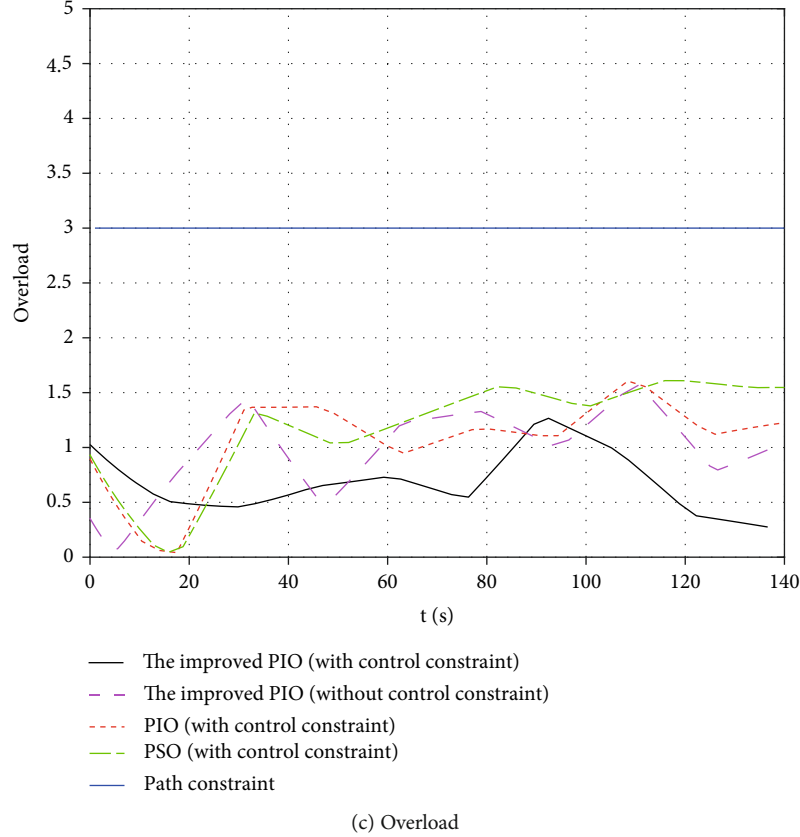


FIGURE 6: Dynamic pressure, heat flux, and overload of optimal ascent trajectories.

TABLE 3: Terminal states of trajectories.

Algorithm	Terminal states
PSO (with control constraints)	$[v, h, r, \gamma, m] = [2338 \text{ m/s}, 26485 \text{ m}, 287863 \text{ m}, 0.11^\circ, 15776 \text{ kg}]$
PIO (with control constraints)	$[v, h, r, \gamma, m] = [2337 \text{ m/s}, 26495 \text{ m}, 285903 \text{ m}, 0.03^\circ, 15833 \text{ kg}]$
The improved PIO (with control constraints)	$[v, h, r, \gamma, m] = [2337 \text{ m/s}, 26495 \text{ m}, 287800 \text{ m}, 0^\circ, 15893 \text{ kg}]$
The improved PIO (without control constraints)	$[v, h, r, \gamma, m] = [2337 \text{ m/s}, 26495 \text{ m}, 274864 \text{ m}, 0^\circ, 15894 \text{ kg}]$

TABLE 4: Performance for different methods.

Algorithm	Computational time (s)	Values of the fitness function
PSO	504	-15237
PIO	521	-15612
The improved PIO	327	-15894

formula can associate the control variable \mathbf{u} with the known trajectory to solve the required control variable.

As shown in Equation (31), the state differential value of the trajectory at the discrete point $n - 1$ is determined by the state variable \mathbf{x}_{n-1} at the current point and the current control input variable $\mathbf{u}_{n-1,k}$. The state variable differential of dis-

crete points can obtain the state variable of the next discrete point through the integration of time.

$$\dot{\mathbf{x}}_{n-1,k} = f[\mathbf{x}_{n-1}, \mathbf{u}_{n-1,k}], \quad (31)$$

$$\mathbf{x}_{n,k} = \int_{t_{n-1}}^{t_n} \dot{\mathbf{x}}(t) dt + \mathbf{x}_{n-1,k}, \quad (32)$$

where t_n is the time at node n ; t_{n-1} is the time at node $n - 1$; and k is the number of iterations. Let Δt be the time interval between two nodes, i.e.,

$$\Delta t = t_n - t_{n-1}. \quad (33)$$

When Δt is fairly small, Equation (32) can be approximated as

$$\mathbf{x}_{n,k} \approx \dot{\mathbf{x}}_{n-1,k} \Delta t + \mathbf{x}_{n-1,k}. \quad (34)$$

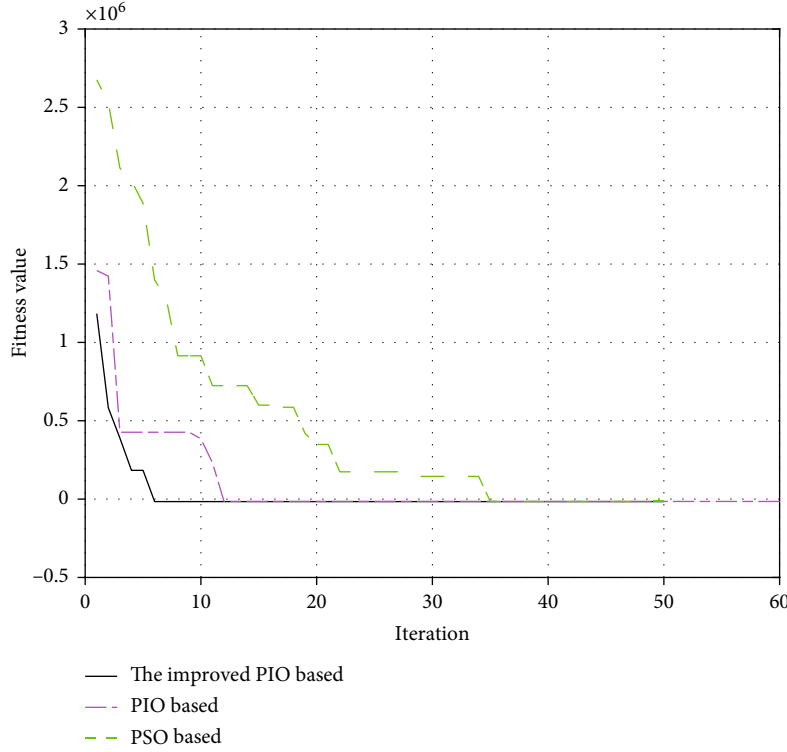


FIGURE 7: Fitness value versus iteration time.

The error vector $\mathbf{e}_{n,k}$ is defined as the error between the state variable differential estimation value at the n th discrete point and the target trajectory point $\dot{\mathbf{x}}_n$ after the trajectory is solved in the k th iteration, as shown in Equation (35). According to the defined error vector, the control variable of the next iteration can be obtained by Newton's method, as shown in Equation (36).

$$\mathbf{e}_{n,k} = \dot{\mathbf{x}}_{n,k} - \dot{\mathbf{x}}_n, \quad (35)$$

$$\mathbf{u}_{n-1,k+1} = \mathbf{u}_{n-1,k} - [\mathbf{J}]^{-1} \mathbf{e}_{n,k}, \quad (36)$$

where \mathbf{J} is the jacobian matrix, which is defined as follows:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{e}_{\dot{x}_v}}{\partial \mathbf{u}_\alpha} & \frac{\partial \mathbf{e}_{\dot{x}_v}}{\partial \mathbf{u}_{\delta_e}} & \frac{\partial \mathbf{e}_{\dot{x}_v}}{\partial \mathbf{u}_\phi} \\ \frac{\partial \mathbf{e}_{\dot{x}_\gamma}}{\partial \mathbf{u}_\alpha} & \frac{\partial \mathbf{e}_{\dot{x}_\gamma}}{\partial \mathbf{u}_{\delta_e}} & \frac{\partial \mathbf{e}_{\dot{x}_\gamma}}{\partial \mathbf{u}_\phi} \\ \frac{\partial \mathbf{e}_{\dot{x}_{qp}}}{\partial \mathbf{u}_\alpha} & \frac{\partial \mathbf{e}_{\dot{x}_{qp}}}{\partial \mathbf{u}_{\delta_e}} & \frac{\partial \mathbf{e}_{\dot{x}_{qp}}}{\partial \mathbf{u}_\phi} \end{bmatrix}, \quad (37)$$

where u_α , u_{δ_e} , and u_ϕ are input of the angle of attack, the elevator deflection, and the fuel equivalent ratio, respectively; and $e_{\dot{x}_v}$, $e_{\dot{x}_\gamma}$, and $e_{\dot{x}_{qp}}$ are the errors of the estimated values of the velocity, the path angle, and the pitch rate, respectively.

Through the iterative solution of Equation (35), the control variable at the $n-1$ discrete point can be obtained, so

that the trajectory can change from the $n-1$ discrete point to the point n .

The trajectory inverse algorithm needs to consider the mass change caused by vehicle fuel consumption. The mass of the vehicle needs to be calculated according to the vehicle mass during the last iteration and the instantaneous fuel flow rate. Assuming that the fuel consumption rate remains unchanged within the time interval Δt , the following equation can be obtained:

$$m_n = m_{n-1} + \dot{m}\Delta t, \quad (38)$$

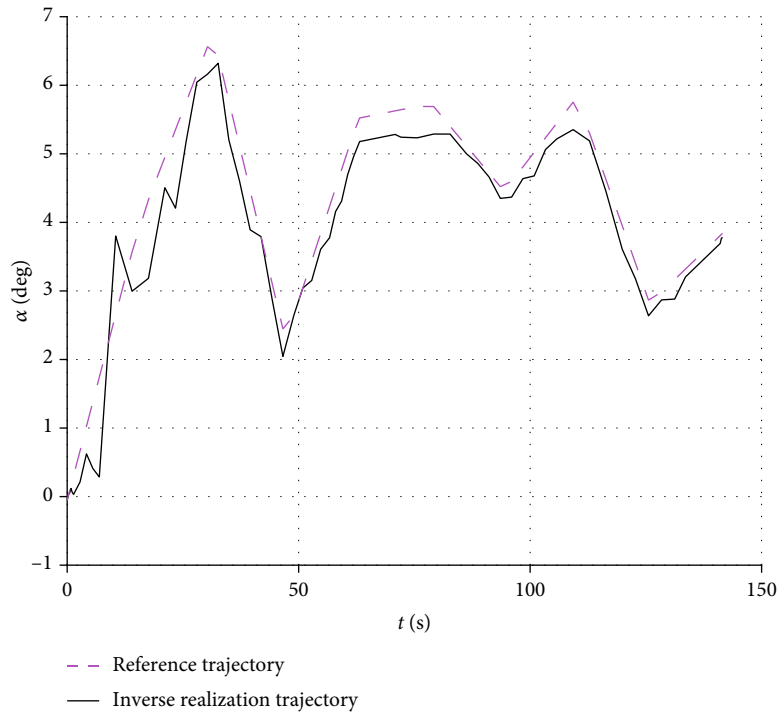
where m_n and m_{n-1} are the mass of the vehicle at the n discrete point and the $n-1$ discrete point, respectively, and \dot{m} is fuel consumption per unit time of the vehicle given in Equation (4).

The entire procedure of trajectory inverse algorithm is shown in Figure 4.

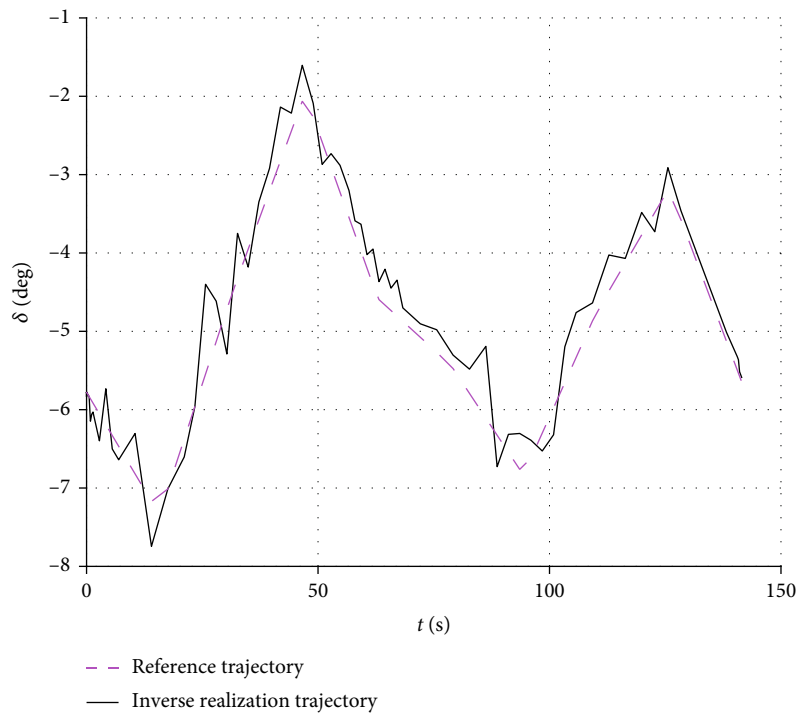
5. Results

The results in this section are mainly divided into two parts. The first part is the comparison of the optimal trajectories obtained by different methods. The second part is to verify the feasibility of the optimal trajectory obtained by the improved PIO algorithm using the trajectory inverse algorithm.

This section uses the Doman model described in Section 2 and studies its optimal ascent trajectory from 6 Mach, altitude 22379 m, to 7.8 Mach, altitude 26495 m. The optimization variables were obtained by using the Chebyshev



(a) Angle of attack



(b) Rudder deflection

FIGURE 8: Continued.

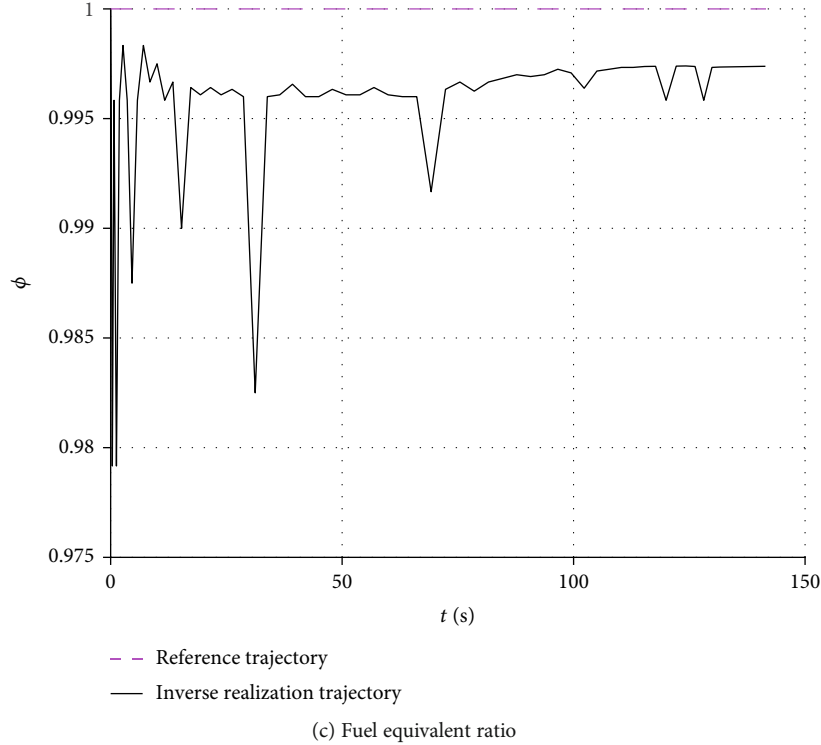


FIGURE 8: Control variables obtained by trajectory inverse algorithm.

interpolation method to discretize the angle of attack and elevator deflection. Then, the optimization variables are expressed as follows:

$$\mathbf{X} = [\alpha_1, \alpha_2, \dots, \alpha_D, \delta_{e_1}, \delta_{e_2}, \dots, \delta_{e_D}, t_f], \quad (39)$$

where α is the angle of attack; δ_e is the elevator deflection; \mathbf{X} is the position vector of pigeons; D is the dimension of the position vector; and t_f is the terminal time;

5.1. Optimal Ascent Trajectory. To invest the feasibility and optimality of the proposed method, comparative studies with other global optimization methods such as PSO and PIO are presented. Besides, to analyze the impact of control constraints, the optimal trajectories with and without control constraints are obtained by the proposed method. The control parameters of global optimization methods are given in Table 2; to avoid the influence of parameters on the effectiveness of the optimization algorithm, all equivalent control parameters are set to be the same.

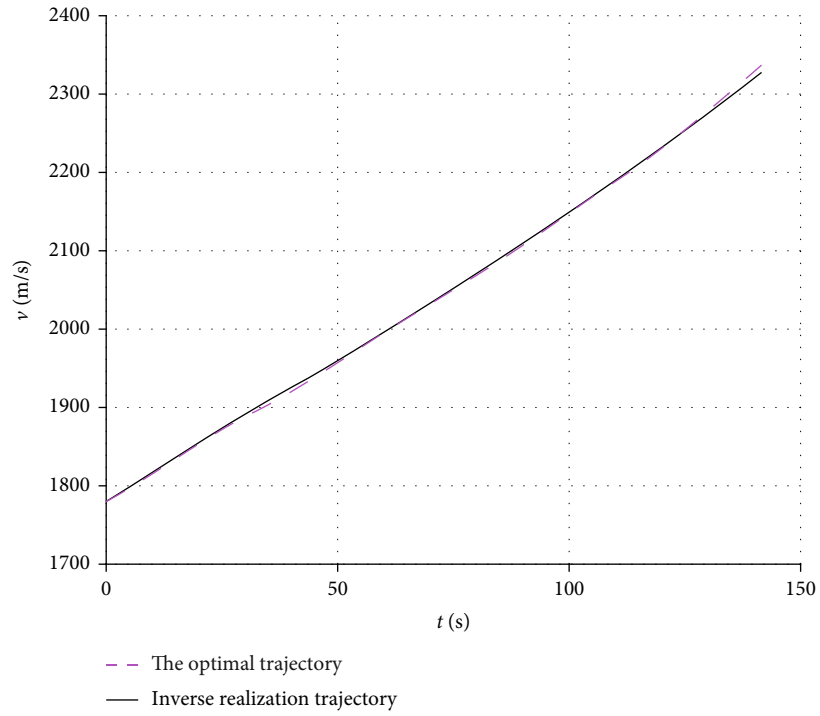
According to the actual situation, select the initial state as $[v, h, r, \gamma, m] = [1780 \text{ m/s}, 22379 \text{ m}, 0, 5^\circ, 16817 \text{ kg}]$. For constraints, set the terminal constraints as $v_f = 2337 \text{ m/s}$, $h_f = 26495 \text{ m}$, and $\gamma_f = 0$; the path constraints are $\dot{Q}_{\max} = 600 \text{ kW/m}^2$, $10 \text{ kPa} \leq q \leq 100 \text{ kPa}$, and $n_{\max} = 3.5$; the state constraints are $1500 \text{ m/s} \leq v \leq 2500 \text{ m/s}$ and $20000 \text{ m} \leq h \leq 30000 \text{ m}$. The above constraints are the basic constraints, and this paper will optimize a standard ascent trajectory based on the basic constraints.

This paper adds control constraints to the basic constraints so that the control performance limitation can be considered into trajectory optimization; the control constraints are as follows:

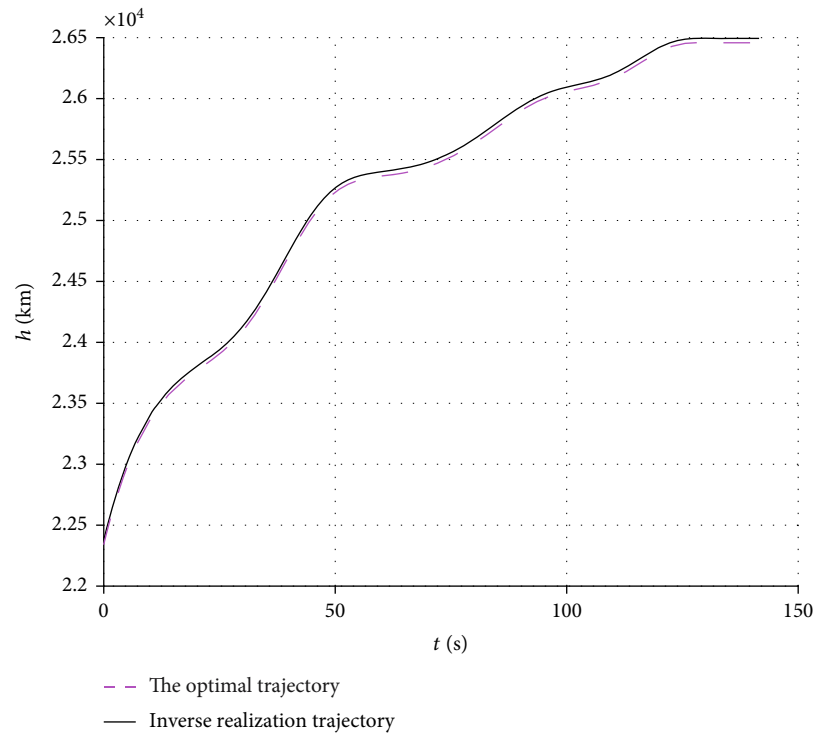
$$\begin{aligned} -6^\circ &\leq \alpha \leq 8^\circ, \\ -10^\circ &\leq \delta_e \leq 10^\circ, \\ 0 &\leq \phi \leq 1, \\ |\dot{\gamma}| &\leq 0.002 \text{ rad/s}, \end{aligned} \quad (40)$$

where α is the angle of attack; δ_e is the elevator deflection; ϕ is the fuel equivalent ratio; and γ is the flight path angle. Besides, if the flight path angle too large or too small, the difficulty of trajectory control will increase. Therefore, this paper also constrains the flight path angle, i.e., $-2^\circ \leq \gamma \leq 5^\circ$. The optimal trajectories generated using different algorithms are plotted in Figures 5 and 6. Table 3 lists the terminal states of optimal trajectories obtained by different algorithms.

When control constraint is considered, it can be seen from Table 3 and Figure 5 that the optimal trajectory obtained by the proposed algorithm is most in line with terminal constraints and has the lowest fuel consumption. Figure 6 shows that compared to the trajectories obtained by other algorithms, the dynamic pressure of the trajectory obtained by the proposed algorithm is largest in the constrained range and the higher dynamic pressure resulted in better performance. The heat flux of the trajectory obtained by the proposed algorithm is smallest, which can reduce the requirement of aerodynamic heating for the vehicle materials

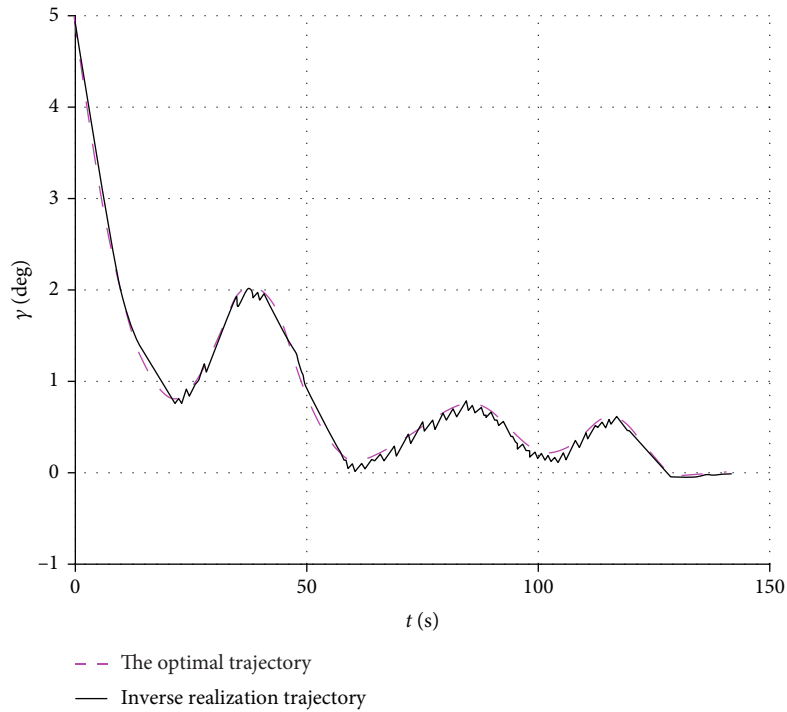


(a) Velocity

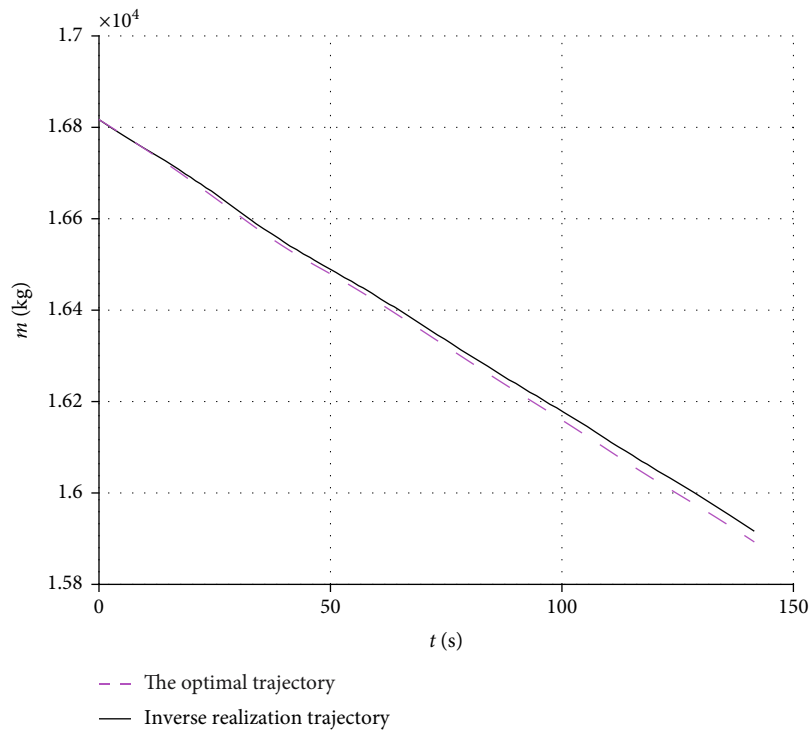


(b) Height

FIGURE 9: Continued.



(c) Flight path angle



(d) Mass

FIGURE 9: Comparison between actual trajectory and optimal trajectory.

to a certain extent. The overload amplitude of the trajectory obtained by the proposed algorithm decreases, which ensures better control performance. Therefore, the proposed trajectory optimization algorithm based on the improved PIO algorithm is feasible and optimal.

The optimal trajectories obtained by the proposed algorithm with and without control constraints in Figures 5 and 6 and Table 3 show that with control constraints having been taken into account, the fuel consumption increases by 0.11% while the range increases by 4.50%, which is of great

practical significance for the long-distance flight. Besides, the ascent trajectory with control constraints avoids the subduction segment and the flight path angle changes more gently; that is, the trajectory control performance is better. From the perspective of dynamic pressure, heat flux, and overload, the trajectory performance with control constraints is also improved.

In order to further analyze the computational complexity of the proposed method, Table 4 illustrates the results of the three methods in terms of the computational time and values of the fitness function, while Figure 7 shows the fitness value versus iteration time.

Table 4 shows that the proposed trajectory optimization algorithm has much less computational time and the minimum fitness value. Besides, it can be seen from Figure 7 that the proposed trajectory optimization algorithm can converge to the global optimal solution fastest. These results further prove that the proposed algorithm is feasible and optimal.

5.2. Trajectory Feasibility. The optimal trajectory is input into the trajectory inverse algorithm to verify the feasibility of trajectory with control constraints. The simulation results are shown in Figure 8.

The control variables obtained by the trajectory inverse algorithm are all within the constraint range. The control variables are brought into the dynamic equations of the vehicle, and the actual ascent trajectory is obtained by numerical integration of state variables by the Runge-Kutta method and compared with the optimal trajectory solved by the proposed algorithm; the results are shown in Figure 9.

The actual trajectory obtained by the trajectory inverse algorithm is consistent with the optimal trajectory solved by the proposed algorithm, thus verifying the feasibility of the optimal trajectory, and the effectiveness of the algorithm is also proved.

6. Conclusion

In this paper, a new trajectory optimization algorithm based on the improved PIO algorithm is proposed to solve the trajectory optimization problem under control constraints.

To get rid of premature convergence, a quantum-based approach is incorporated into the PIO algorithm to perceive deceptiveness and preserve the diversity of the population. To be specific, the position of pigeons is represented by qubit, the quantum representation of the current best solution can effectively maintain the diversity in exploration. Then, the quantum rotation gate is used to decrease the deceptive probability.

By comparing the optimization result with trajectory optimization algorithm based on PSO and the standard PIO, it can be seen that although the trajectories are quite different, the trajectory obtained by the proposed algorithm has a better performance index, and the terminal states are more accurate. Meanwhile, compared with the optimal trajectory without control constraints, the optimal trajectory with control constraints has better control performance. Also, the proposed algorithm has shorter computational time and smaller fitness value and can converge to the optimal solution

faster. Therefore, the proposed algorithm is feasible in trajectory optimization.

In the following research, two problems can be further studied. The first point is how to choose the appropriate interpolation method for the given optimal control problem. Different interpolation methods and interpolation nodes affect the effectiveness and efficiency of the algorithm. The second point is that the efficiency of the PIO algorithm is affected by parameters. How to select appropriate parameters to optimize the efficiency of the algorithm is worth studying.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the Nanjing University of Aeronautics and Astronautics Graduate Innovation Base (Laboratory) Open Fund (Grant No. kfj20191503).

References

- [1] B. Yan, R. Liu, P. Dai, M. Xing, and S. Liu, "A rapid penetration trajectory optimization method for hypersonic vehicles," *International Journal of Aerospace Engineering*, vol. 2019, no. 2, 11 pages, 2019.
- [2] Z. Li, T. Yang, and Z. Feng, "The multiobjective trajectory optimization for hypersonic glide vehicle based on normal boundary intersection method," *Mathematical Problems in Engineering*, vol. 2016, Article ID 9407238, 8 pages, 2016.
- [3] N. Li, H. Lei, L. Shao, T. Liu, and B. Wang, "Trajectory optimization based on multi-interval mesh refinement method," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8521368, 8 pages, 2017.
- [4] G. N. Kumar, M. Ikram, A. K. Sarkar, and S. E. Talole, "Hypersonic flight vehicle trajectory optimization using pattern search algorithm," *Optimization and Engineering*, vol. 19, no. 1, pp. 125–161, 2018.
- [5] Y. Xie, L. Pan, X. Gu, and J. Chen, "Gliding trajectory optimization with multiple objectives and complicated constraints for hypersonic vehicles," *Journal of National University of Defense Technology*, vol. 39, no. 2, pp. 9–17, 2017.
- [6] R. Chai, A. T. Al Savvaris, Y. Xia, and S. Chai, "Solving multi-objective constrained trajectory optimization problem by an extended evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1630–1643, 2020.
- [7] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Optimal fuel consumption finite-thrust orbital hopping of aeroassisted spacecraft," *Aerospace Science and Technology*, vol. 75, pp. 172–182, 2018.
- [8] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 467–480, 2019.

- [9] W. Fu, B. Wang, X. Li, L. Liu, and Y. Wang, "Ascent trajectory optimization for hypersonic vehicle based on improved chicken swarm optimization," *IEEE Access*, vol. 7, pp. 151836–151850, 2019.
- [10] X. Cheng, H. Li, and R. Zhang, "Efficient ascent trajectory optimization using convex models based on the Newton-Kantorovich/Pseudospectral approach," *Aerospace Science and Technology*, vol. 66, pp. 140–151, 2017.
- [11] A. Mahmoud, Z. Hao, C. Wanchun, and L. Yang, "Ascent and glide trajectory optimization for hypersonic vehicle," *2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization*, pp. , 2017133–142, 2017.
- [12] E. Yong, L. Chen, and G. Tang, "A survey of numerical methods for trajectory optimization of spacecraft," *Journal of Astronautics*, vol. 29, no. 2, pp. 397–406, 2008.
- [13] E. O. Nascimento and L. N. Oliveira, "Numerical optimization of flight trajectory for rockets via artificial neural networks," *IEEE Latin America Transactions*, vol. 15, no. 8, pp. 1556–1565, 2017.
- [14] Y. Nie and E. C. Kerrigan, "Efficient and more accurate representation of solution trajectories in numerical optimal control," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 61–66, 2019.
- [15] H. Zhang, X. Wang, P. Memarmoshrefi, and D. Hogrefe, "A survey of ant colony optimization based routing protocols for mobile ad hoc networks," *IEEE Access*, vol. 5, pp. 24139–24161, 2017.
- [16] S. Kuo and Y. Chou, "Entanglement-enhanced quantum-inspired tabu search algorithm for function optimization," *IEEE Access*, vol. 5, pp. 13236–13252, 2017.
- [17] V. Sankardoss and P. Geethanjali, "PMDC motor parameter estimation using bio-inspired optimization algorithms," *IEEE Access*, vol. 5, pp. 11244–11254, 2017.
- [18] H. Duan and P. Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning," *International Journal of Intelligent Computing and Cybernetics*, vol. 7, no. 1, pp. 24–37, 2014.
- [19] B. Zhang and H. Duan, "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 99, pp. 97–107, 2012.
- [20] R. Dou and H. Duan, "Lévy flight based pigeon-inspired optimization for control parameters optimization in automatic carrier landing system," *Aerospace Science and Technology*, vol. 61, pp. 11–20, 2017.
- [21] T. Li, C. Zhou, B. Wang, B. Xiao, and X. Zheng, "A hybrid algorithm based on artificial bee colony and pigeon inspired optimization for 3D protein structure prediction," *Aerospace Science and Technology*, vol. 12, no. 1, pp. 100–108, 2018.
- [22] T. Lu and J. Juang, "A region-based quantum evolutionary algorithm (RQEA) for global numerical optimization," *Journal of Computational and Applied Mathematics*, vol. 239, pp. 1–11, 2013.
- [23] G. Deng, M. Wei, Q. Su, and M. Zhao, "An effective co-evolutionary quantum genetic algorithm for the no-wait flow shop scheduling problem," *Advances in Mechanical Engineering*, vol. 7, no. 12, 2015.
- [24] R. Zhang and H. Gao, "Improved quantum evolutionary algorithm for combinatorial optimization problem," *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3501–3505, 2007.
- [25] M. A. Bolender and D. B. Doman, "Nonlinear longitudinal dynamical model of an air-breathing hypersonic vehicle," *Journal of Spacecraft and Rockets*, vol. 44, no. 2, pp. 374–387, 2007.
- [26] J. T. Parker, A. Serrani, S. Yurkovich, M. A. Bolender, and D. B. Doman, "Control-oriented modeling of an air-breathing hypersonic vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 3, pp. 856–869, 2007.
- [27] F. Chavez and D. Schmidt, "Analytical aeropropulsive-aeroelastic hypersonic-vehicle model with dynamic analysis," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 6, pp. 1308–1319, 1994.
- [28] J. Allison and D. R. Herber, "Special section on multidisciplinary design optimization: multidisciplinary design optimization of dynamic engineering systems," *AIAA Journal*, vol. 52, no. 4, pp. 691–710, 2014.
- [29] B. A. Steinfeldt and R. D. Braun, "Using dynamical systems concepts in multidisciplinary design," *AIAA Journal*, vol. 52, no. 6, pp. 1265–1279, 2014.
- [30] A. N. Al-Rabadi, "New dimensions in non-classical neural computing, part II: quantum, nano, and optical," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 3, pp. 513–573, 2009.
- [31] I. G. Tsoulos and A. Stavrakoudis, "Enhancing PSO methods for global optimization," *Applied Mathematics and Computation*, vol. 216, no. 10, pp. 2988–3001, 2010.
- [32] D. J. Murray-Smith, "The inverse simulation approach: a focused review of methods and applications," *Mathematics and Computers in Simulation*, vol. 53, no. 4-6, pp. 239–247, 2000.