Chinese Journal of Aeronautics, (2015), 28(3): 822–831

CSAA

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Particle swarm optimization applied to hypersonic reentry trajectories

Zhao Jiang *, Zhou Rui

School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

Received 6 May 2014; revised 24 October 2014; accepted 22 January 2015
Available online 15 April 2015

KEYWORDS

Footprint;
Hypersonic vehicles;
Particle swarm optimization (PSO);
Reentry;
Trajectories

Abstract This paper presents the novel use of the particle swarm optimization (PSO) to generate the end-to-end trajectory for hypersonic reentry vehicles in a quite simple formulation. The velocity-dependent bank angle profile is developed to reduce the search space of unknown parameters based on the constrained PSO algorithm. The path constraints are enforced by setting the fitness function to be infinite on condition that the particles violate the maximum allowable values. The PSO algorithm also provides a much easier means to satisfy the terminal conditions by adding penalty terms to the fitness function. Furthermore, the approximate reentry landing footprint is fast constructed by incorporating an interpolation model into the standardized bank angle profiles. Numerical simulations demonstrate that the PSO method is a feasible and flexible tool to generate the end-to-end trajectory and landing footprint for hypersonic reentry vehicles.

1. Introduction

Since human started the space era, global strike and space transportation have spurred a great interest in hypersonic vehicles for both civilian and military applications. The need for an effective and reliable access to the space is promoting a rapid development of hypersonic vehicles.[1,2] The progress is witnessed by the experimental success of NASA's scramjet-powered X-43A in 2004, US Air Force's X-51 in 2010 and DARPA's Falcon HTV-2 in 2011. Although the X-51 went through serious test failures after its first flight, the recent test missions in 2013 succeeded in covering a large downrange more than 230 nautical miles. In addition, the X-37B orbital test vehicle also completed a successful flight test in 2012, which lasts for 469 days and demonstrates the great capability of the flight inspection and data analysis.

The reference trajectory is one of key components of the reentry guidance for hypersonic vehicles. Therefore, the design of reference trajectory plays an important role in steering a safe and efficient flight. In general, the reference trajectory is generated offline and preloaded on the hypersonic vehicle before its launching. The vehicle enters the atmosphere of the Earth at an altitude of about 100–120 km. The full trajectory typically expands to the range of the terminal area at about 20–30 km in altitude. It is a challenging task to design the reference trajectory for hypersonic vehicles, since the reentry dynamics is highly nonlinear with limited control authority. Besides, hypersonic vehicles must be subject to a great many

* Corresponding author. Tel.: +86 10 82316849.
E-mail address: jzhao@buaa.edu.cn (J. Zhao).

Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.cja.2015.04.007
1000-9361

path constraints in the complex environment such as the heating rate, dynamic pressure and aerodynamic load.[3,4]

In the current literature, three typical classes of approaches have been proposed to design the constrained reentry trajectory for hypersonic vehicles. One is to reduce the complexity of the problem by using a reduced-order model. The evolved acceleration guidance logic for entry (EAGLE) that includes a trajectory planner to generate the atmospheric reentry trajectory was presented in Refs.[5–7]. The design of both the feasible and optimal trajectories is based on the drag planning technique for space shuttles. The second type of approaches employs the quasi-equilibrium glide phenomenon for lifting vehicles. Shen and Lu[8,9] proposed the quasi-equilibrium glide condition (QEGC) to generate the constrained reentry trajectory for hypersonic vehicles. The nonlinear trajectory design is decomposed into two sequential one-parameter search problems. The third class of approaches uses the direct trajectory optimization technique, in which the reentry trajectory planning is based on pseudospectral methods (PSM).[10–13] By approximating the state and control at a set of discretization points, the optimal control problem is transcribed into the nonlinear programming (NLP) problem that can be solved by using much more approaches.

In this paper, the particle swarm optimization (PSO) is used to generate the end-to-end trajectory for hypersonic reentry vehicles in a quite simple formulation. The contributions of the paper are as follows: (1) a standardized bank angle profile is developed to reduce the search space of unknown parameters based on the PSO method; (2) two interpolation models are incorporated into the control profiles for rapid construction of the reentry landing footprint.

2. Preliminaries

The PSO method is one of the swarm intelligence methods that take the original inspiration from the natural phenomena. It mimics the motion of the bird flocks while searching for a food source. The idea of the PSO was first proposed by Eberhart and Kennedy[14] in 1995 and then modified by Shi and Eberhart.[15] As a population-based optimization tool, the PSO has a main strength that each particle uses the experience of the whole particles in the search space rather than only the experience of its own. This feature of the PSO results in a fast convergence.[16]

The initial set of the particles is randomly distributed in the searching space. At a given iteration, each particle has a position vector, a velocity vector and a vector of its previous best position. Each particle in the swarm represents a possible solution and corresponds to a specific value of the objective (fitness) function. Both the position vectors and velocity vectors are updated using the following information:

(1) The distance between its current position and the best position so far of its own.
(2) The distance between its current position and the best position so far in the group.

At the end of the iteration, the best particle in the swarm is selected.[17]

In the current literature, two classes of particle swarms are typically used including the local particle swarm and the global particle swarm.[18] The local particle swarm selects the collective best position among the particles in a given neighborhood of the particle itself, while the global particle swarm takes into account the entire swarm. In this paper, we adopt the basic version of the global particle swarm algorithm since it is well suited for finding the optimal solution to trajectory optimization problems. In addition, it is quite easy to define the search space for the unknown parameters such that a simple MATLAB or C/C++ code can be implemented and applied to the trajectory optimization problem. In the following text, the unconstrained and constrained PSO algorithms are delineated.

2.1. Unconstrained PSO

The rationale of the unconstrained parameter optimization is to determine the optimal unknown parameters such that the objective function is minimized. Assume that $\{x_1, x_2, \ldots, x_n\}$ are the n unknown parameters that have their own bounds in terms of

$$x_i \in [a_i, b_i] \quad (i = 1, 2, \ldots, n) \tag{1}$$

where a_i and b_i are the lower and upper bounds of the ith unknown parameter. The population in the PSO is represented by a swarm of N particles. Then, each particle k is associated with a position vector $\boldsymbol{x}(k)$ and a velocity vector $\boldsymbol{v}(k)$ as

$$\boldsymbol{x}(k) = [x_1(k),\ x_2(k),\ \ldots,\ x_n(k)]^{\mathrm{T}} \quad (k = 1, 2, \ldots, N) \tag{2}$$

$$\boldsymbol{v}(k) = [v_1(k),\ v_2(k),\ \ldots,\ v_n(k)]^{\mathrm{T}} \quad (k = 1, 2, \ldots, N) \tag{3}$$

where the terms $\boldsymbol{x}(k)$ and $\boldsymbol{v}(k)$ are referred to the search space of the n unknown parameters without any physical meaning. The elements of the two vectors are represented by $x_i(k)$ and $v_i(k)$ $(i = 1, 2, \ldots, n)$. According to the bounds of the n unknown parameters, the related position and velocity components are limited to

$$\begin{cases} a_i \leqslant x_i(k) \leqslant b_i \\ |v_i(k)| \leqslant |a_i - b_i| \end{cases} (i = 1,\ 2, \ldots, n;\ \ k = 1, 2, \ldots, N) \tag{4}$$

Each particle described by the terms $\boldsymbol{x}(k)$ and $\boldsymbol{v}(k)$ represents a possible solution to the unconstrained optimization problem and results in a specific value of the objective function. The swarm evolution to the global optimal location is determined by the position and velocity update. Suppose that the PSO algorithm terminates at the maximum number of the iterations N_{IT}. In a generic iteration j (j = 1, 2, …, N_{IT}), the fitness function is evaluated with the particle k. The best position $\boldsymbol{p}_{\mathrm{best}}^{(j)}(k)$ ever visited by the particle k is determined. Then, determine the global best position $\boldsymbol{g}_{\mathrm{best}}^{(j)}(k)$ ever visited by the swarm such that the update of the velocity vector for each particle k can be described as[15]

$$\begin{aligned} \boldsymbol{v}^{(j+1)}(k) = {} & w\boldsymbol{v}^{(j)}(k) + c_1 r_1(0,1)\left(\boldsymbol{p}_{\mathrm{best}}^{(j)}(k) - \boldsymbol{x}^{(j)}(k)\right) \\ & + c_2 r_2(0,1)\left(\boldsymbol{g}_{\mathrm{best}}^{(j)}(k) - \boldsymbol{x}^{(j)}(k)\right) \quad (k = 1, 2, \ldots, N) \end{aligned} \tag{5}$$

where $\boldsymbol{x}^{(j)}(k)$ and $\boldsymbol{v}^{(j)}(k)$ are the position vector and velocity vector in each iteration; w is the inertial weight; c_1 and c_2 represent the influences of the cognitive and social components, respectively; r_1 (0, 1) and r_2 (0, 1) are independent random numbers