ISBN 978-952-5726-04-6 (Print), 978-952-5726-05-3 (CD-ROM)
Proceedings of the International Symposium on Intelligent Information Systems and Applications (IISA'09)
Qingdao, P. R. China, Oct. 28-30, 2009, pp. 047-050

Design of Trajectory Tracking for Vehicles using PSO Algorithm

Chuanfeng Li[1,2], Lingling Tang[1], Yongji Wang[1], and Zhixiang Deng[1]
[1]Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China
[2]Department of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, China
Email: lichuanfeng@smail.hust.edu.cn

***Abstract*— A trajectory programming algorithm based drag acceleration profile is presented which aiming at the reference trajectory in reentry trajectory optimization for gliding vehicles. Firstly, Motion equations of vehicles are simplified and control variable is parameterized. Secondly, the optimal control problem of trajectory programming is simplified as one-dimensional searching problem from longitudinal and lateral parts. Finally, the tracking controller is designed for tracking the drag acceleration profile, where the particle swarm optimization(PSO) is adopted in order to optimize the gain coefficient of tracking controller. Simulation results show that the obtained reentry trajectory can save the subsequently optimization iteration time and approach the best trajectory.**

***Index Terms*—reentry vehicles, drag acceleration profile, trajectory tracking, particle swarm optimization**

I. INTRODUCTION

When the gliding vehicle gets back to earth, the Earth's atmosphere is utilized as a kind of natural resource to consume its huge kinetic energy and potential energy in the form of thermal energy and at last it reentry to the earth surface. At the same time, compression and friction between vehicle and atmosphere transform a part of energy into heat energy around, and the heat energy partly passes back to the vehicle by convection and radiation. In this whole process the vehicle suffers severe aerodynamic heating and overload[1].

The optimization of reentry trajectory is done by designing an optimized trajectory and making the vehicle move along it[2], which can alleviate the aerodynamic heating of the vehicle during the reentry process as well as greatly reduce the pressure in designing heat-resistance. Furthermore, it can be described as a nonlinear optimal control problem which is subject to some control constraints, terminal constraints and process constraints. It is difficult to find accurate analytic solutions of optimal control variables because of complicated nonlinear model characteristic. The direct method for trajectory optimization is focused in recent research, but it requires enormous iterative time to search the optimal control variables and its corresponding optimal trajectory[3]. In addition, it is very sensitive to the initial position of the reference trajectory because the quality of the reference trajectory will affect convergence and accuracy in subsequent optimization process. Therefore, in order to guarantee the iterative efficiency, the reference trajectory should be as close to the optimal solution as possible[4].

Based on constructing non-dimensional dynamic model, this paper calculates the reentry corridor of drag acceleration vs. velocity according to the constraint conditions. Among the corridor, the reference profile of drag acceleration can be decided by one-dimensional search method, and then a reference trajectory can be planned quickly by designing tracking controller, which can increase the iterative efficiency in subsequent optimization algorithm.

II. MOTION MODEL OF VEHICLE

According to theoretical mechanics and kinematics principle[5], we can deduce the derivation of kinematics equations of hypersonic vehicle. The 3DOF equations of motion through dimensionless method are given by

$$\left\{\begin{aligned}
&\frac{dR}{d\tau}=V\sin\gamma\\
&\frac{d\theta}{d\tau}=\frac{V\cos\gamma\sin\psi}{R\cos\phi}\\
&\frac{d\phi}{d\tau}=\frac{V\cos\gamma\cos\psi}{R}\\
&\frac{dV}{d\tau}=-D-\left(\frac{\sin\gamma}{R^2}\right)+\omega^2R\cos\phi(\sin\gamma\cos\phi-\cos\gamma\sin\phi\cos\psi)\\
&\frac{d\gamma}{d\tau}=\frac{1}{V}[L\cos\sigma+\left(V^2-\frac{1}{R}\right)\left(\frac{\cos\gamma}{R}\right)+2\omega V\cos\phi\sin\psi+\\
&\qquad\omega^2R\cos\phi(\cos\gamma\cos\phi+\sin\gamma\cos\psi\sin\phi)]\\
&\frac{d\psi}{d\tau}=\frac{1}{V}[\frac{L\sin\sigma}{\cos\gamma}+\frac{V^2\cos\gamma\sin\psi\tan\phi}{R}-2\omega V(\tan\gamma\cos\psi\cos\phi-\sin\phi)+\\
&\qquad\frac{\omega^2}{\cos\gamma}\sin\psi\sin\phi\cos\phi]
\end{aligned}\right.\tag{1}$$

where, the dimensionless parameters of geocentric vector *R*, velocity of vehicle *V*, time τ and rotational angular velocity of the Earth ω represent the average vector of the earth R_0, $\sqrt{g_0R_0}$, $\sqrt{R_0/g_0}$, $\sqrt{g_0/R_0}$ respectively. g_0 represents the sea level gravitational acceleration. θ and ϕ represent geographical longitude and latitude respectively. γ represents flight path angle, which is the angle between velocity vector and the local horizontal level; ψ represents heading angle, which is the angle between local longitude line and the projection of horizontal plane of velocity vector, along the clockwise rotation toward north, its value is positive.

The control law of attack angle can be obtained by qualitative analysis of reentry process. According to flight mission, angle of attack can be preselected. Commonly, the curve of attack angle will change with flight velocity. Then, we can decide the reentry trajectory by deciding the control law of bank angle σ only.

III. TRAJECTORY PROGRAMMING ALGORITHM

A. Longitudial Trajectory Programming

Generally speaking, the reentry trajectory programming can be divided into the longitudinal profile programming and lateral profile programming. In this paper, we simplify those algorithms and turn them into the two independent parameter optimization problems of one-dimensional search.

1) Aerodynamic Heating Rate Constraint

In reentry process, the friction between the vehicle and the atmosphere will leads to severe aerodynamic heating problem, and the surface temperatures of different parts of the vehicle are different. The effect of aerodynamic heating of the critical heating region in the vehicle head must be considered as it is relatively serious. So the limits on the surface temperature can be transferred to the constraint for the aerodynamic heating rate in engineering practice. Q_s is determined by the following formula[6]:

$$Q_s = k\left(\frac{\rho}{\rho_0}\right)^{0.5} V^{3.25} \tag{2}$$

The drag acceleration is given by

$$D \le \frac{C_D S_r}{2mg_0}\left(\frac{Q_{smax}^2}{k^2 V^{4.5}}\right) \tag{3}$$

2) Dynamic Pressure Constraint

There should be some restrictions on the dynamic pressure in order to reduce the weight of actuator. The maximum dynamic pressure restriction depends on the strength of defending heat materials for aircraft surface and the aerodynamic hinge moment that augments with the change of dynamic pressure. And it should satisfy:

$$q \le q_{max} \tag{4}$$

where

$$q = \frac{1}{2}\rho (VV_0)^2 \tag{5}$$

According to the relationship between dynamic pressure and the velocity as well as the relationship between velocity and drag acceleration, we can conclude the constraint on the drag acceleration:

$$D \le \frac{C_D q_{max} S_r}{mg_0} \tag{6}$$

3) Overload Constraint

In 3DOF reentry trajectory, the overload constraint quantity is the maximum overload and it depends on structural strength of the vehicle, the total overload constraint satisfies:

$$\sqrt{(L^2 + D^2)} \le n_{max} \tag{7}$$

The constraint to the drag acceleration can be transformed into

$$D \le \frac{n_{max} C_D}{\sqrt{{C_L}^2 + {C_D}^2}} \tag{8}$$

4) The Quasi-Equilibrium Glide Condition

Under normal circumstances, there will be some oscillation and bounce for reentry trajectory of vehicle in the reentry process and they should be avoided. The ideal reentry trajectory should have no bounce and its flight path angle should change smoothly, which means $\gamma \approx 0$ and $d\gamma/dt \approx 0$. According to (1) and if we ignore the Earth's rotation, then we will conclude that the balance of gliding constraint should satisfy:

$$L\cos\sigma + \left(V^2 - \frac{1}{R}\right)\frac{1}{R} = 0 \tag{9}$$

Through the calculation under constraints, the upper and lower boundaries of drag acceleration which satisfy constraints can be obtained under every velocity, and form the reentry corridor of drag acceleration and velocity[7].

B. Lateral Trajectory Programming

The lateral programming should satisfy two requirements of flight, one is the terminal heading angle ψ and the other is cross-range of vehicle.

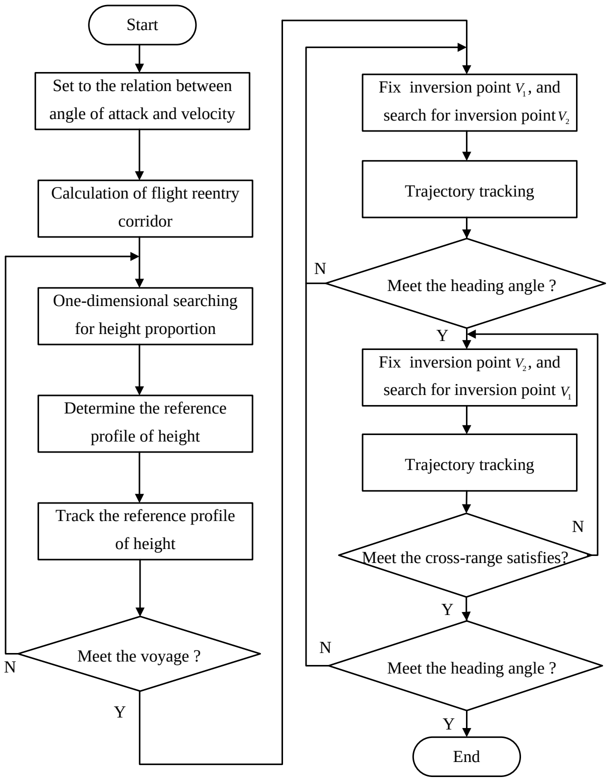

Figure 1. Flowchart of trajectory programming strategy.

The reversal point strategy of bank angle of vehicle can use searching strategy of single inversion point and can also adopt searching strategy of multiple reversal point. If we adopt the single reversal point and meet the requirements above, then the reversal point of bank angle can be found in theory, but it need to consider the longitudinal characteristic at the same time which will become more complicated to solve. Therefore, considering the lateral characteristic, we can search reversal point of bank angle to meet the requirements above by using the searching strategy of two reversal points of bank angle with a short iteration step. The flow chart of the programming algorithm is shown as in Fig.1.

IV. PSO ALGORITHM AND TRAJECTORY TRACKING

A. Priciple of Particle Swarm Optimization

The particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr.Eberhart and Dr.Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling[8].

Every feasible solution is called a "particle" in optimization problems, and a large number of feasible solutions constitute the particle population. Each individual updates its own position constantly by obtaining the search experience of its own and other particles of the population, and they will eventually tend to the same point which is the optimal solution of the solution space. Of course, a large-scale iterative is required if they converge to the same point completely, so we take a certain radius of the particle population converge or maximum iteration time as the termination judgment condition.

We suppose the search space of optimization problem is D-dimensional and the number of particles is N. The position of the ith particle can be represented as a vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and the rate of the position change(velocity) for particle i is represented as the vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, where $1 \le d \le D$ and $1 \le i \le N$. The best previous position (the position giving the best fitness value) of the ith particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$.The position of the best particle among all particles in the population is represented by the symbol P_g or *gBest*. The position and its change rate of every individual updates continuously according to the following equations:

$$v_{id}(t+1) = w \times v_{id} + c_1 \times rand() \times (p_{id}(t) - x_{id}(t)) + c_2 \times rand() \times (p_{gd}(t) - x_{id}(t)) \tag{10}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{11}$$

where, c_1 and c_2 are positive constants which are called acceleration factors. The function rand () is used to generate random numbers between 0 and 1 to ensure a certain degree of diversity for searched solutions. w represents the inertia factor while the d-dimensional of the change scope of location and the change scope of velocity are $[-x_{dmax}\ x_{dmax}]$ and $[-v_{dmax}\ v_{dmax}]$ respectively.

Some optimization parameters have relatively mature common value throughout the optimization process, such as the acceleration factor can be set as $c_1 = c_2 = 2$ and the inertial factor is set as $w=1$. However, among the adjustable parameters, the maximum speed of particles v_{max} affects the searching efficiency of algorithm greatly, especially when there is no inertial factor for the original PSO and it will affect the initialized situation of PSO. The value of v_{max} can be obtained by simulation and debugging in algorithm implementation, but over accuracy is not necessary. The scale of population also has certain effect for the algorithm in that if the number are small, then they will fall into local optimal solution[9], on the other hand, the larger the number of the particles, the slower the convergence speed, which thus lost the advantage of rapidity of PSO.

At present, the most popular method is time-varying weight method and this adjustment method is a weights strategy with the linear reduced number of iterations which is presented by Shi[10]. Its effectiveness has been verified in a large number of applications, that is

$$\omega = \omega_1 + \frac{(\omega_0 - \omega_1)}{N}(N - n) \tag{12}$$

where, N is the maximum iteration number set by PSO algorithm, n represents the current iteration, ω_0 and ω_1 are initial and ultimate inertia weight respectively.

B. Selection of Optimization Parameters

After the reference profile of drag acceleration of vehicle has been obtained, we need to track the profile in order to obtain the control parameter of bank angle, in which the PD controller is adopted. The particle swarm optimization algorithm can be used to optimize the parameters of the tracking controller because it is a function optimization problem in essence[11].

First of all, the solution space of parameter application problem should be mapped to population of PSO, moreover, the particle can be set to a two-dimensional real number vector, and the first dimension is the proportional coefficient of controller while the second dimension is the differential coefficient of controller. Here, it is a continuous particle optimization problem. In this optimization program, we select 30 particles, and the maximum displacement and velocity rely on the problem itself. Considering the proportional relation of control parameters and error, and it is reasonable to set the searching range of maximum displacement varies between 0 and 0.1. In this paper, we limit the maximum of particle between 0 and 0.01 in order to find better optimal solution.

Secondly, the problem is about how to select the fitness function. In order to ensure the good performance in the whole tracking process, we take the sum of square error of instruction height and tracking height as the performance index, and at the same time in order to ensure the constraint of smooth shift for control quantity, we also consider adding the constraint of it for the fitness function, that is

$$f = \sum_{i=1}^{n}(H_{com}^{i} - H^{i})^2 + \sum_{i=1}^{n-1}(\sigma^{i+1} - \sigma^{i})^2 \tag{13}$$

where, n represents the total steps of flight simulation, that is, the error value needs to be accumulated in each step of simulation. Therefore, aiming at one computation of fitness function, the whole flight process requires to be calculated for one time in order to obtain the entire flight state.

V. SIMULATION RESULTS

The programming algorithm presented in this paper is implemented in generating entry trajectory with a certain vehicle data. The terminal constraint of target point and satisfying condition are shown in table 1.

For the trajectory programming problem based on reference profile of height, the tracking performances of height is plotted in Fig.2 by PSO algorithm, from which the algorithm has better tracking performance for height.

TABLE I. TERMINAL CONSTRAINTS OF REFERENCE TRAJECTORY

	ΔR (m)	$\Delta \theta$ (°)	$\Delta \phi$ (°)	ΔV (m/s)	$\Delta \gamma$ (°)	$\Delta \psi$ (°)
Constraint requirement	±500	±0.05	±0.05	±20	-3~0	±2
Simulation results	10.4	0.045	0.019	2.7	-1.51	-1.08

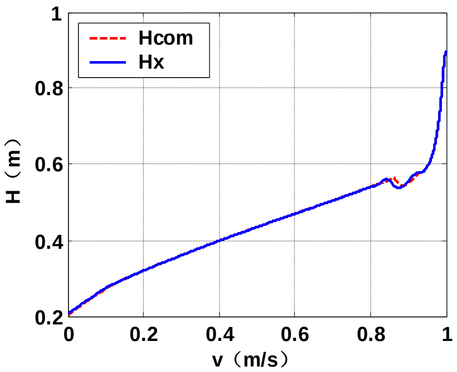

Figure 2. Tracking curves for height using PSO algorithm.

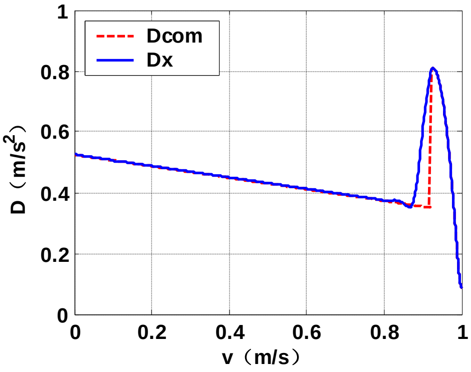

Figure 3. Tracking curves of drag acceleration using PSO algorithm.

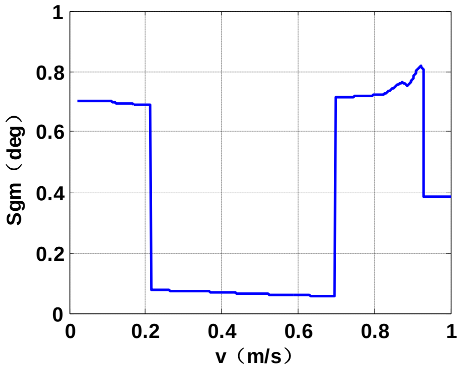

Figure 4. Optimization curve of bank angle using SPSO.

With respect to the tracking controller parameters of trajectory programming problem for drag acceleration profile, which is also optimized by PSO algorithm in Fig.3. Because the PSO algorithm is a random searching method using computer and dependency on human is little, so the algorithm has certain advantages compare with other algorithm such as the one-dimensional search method by trail-and-error. Fig.4 denotes the optimization curve of bank angle using SPSO. The control variable changes smoothly and meets the control constraints well.

VI. CONCLUSIONS

This paper introduces the study of programming algorithm of reentry trajectory on a certain gliding vehicle model. The ultimate aim is planning an entry reference trajectory as initial value to advance the speed of entry trajectory optimization. On the basis of analyzing the profile of drag acceleration vs. velocity, programming trajectory was simplified as one-dimensional search problem from longitudinal and lateral in two ways, especially, the particle swarm optimization algorithm has been adopted to improve the accuracy of trajectory tracking. The simulation results show the effectiveness by the programming algorithm above, and it has certain engineering application value in trajectory programming for vehicles.

ACKNOWLEDGMENTS

We are grateful for the support of the National Nature Science Foundation of China, No. 60674105, the scientific research cultivation project of Ministry of Education, No. 20081383 and the 2008 Spaceflight Support Foundation.

REFERENCES

[1] H.Y,Zhao, *Reentry Dynamics and Guidance of Vehicles*, Changsha:National University of Defense Technology Press,first edition,1997, pp.1–22.

[2] J.A.Leavitt, K. D.Mease, “Feasible Trajectory Generation for Atomospheric Entry Guidance,” *Journal of Guidance, Control, and Dynamics*, vol.30, no.2, 2007, pp.473–481.

[3] R. J.Vanderbei, “Case Studies in Trajectory Optimization: Trains, Planes, and Other Pastimes,” *Optimization and Engineering*, vol. 2, no. 2, 2001, pp. 215–243.

[4] K.P.Bollino,I.M.Ross and D.D.Doman,“Optimal Nonlinear Feedback Guidance for Reentry Vehicle,” In *Proceeding of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone,Volorado, 2006, pp.563–582.

[5] P.Lu, John M.Hanson, “Entry Guidance for the X-33 Vehicle,” *Journal of Spacecraft and Rockets*, vol.35, no.3, 1998, pp. 342-349.

[6] Z.Shen, P.Lu, “On-Board generation of three-dimensional constrained entry trajectories,” In *Proceedings of AIAA Guidance , Navigation, and Control Conference and Exhibit*, Monterey, California, 2002.

[7] K. D.Mease, D. T.Chen and P.Teufel, “Reduced-Order Entry Trajectoy Planning for Acceleration Guidance,” *Journal of Guidance, Control, and Dynamics*, vol.25, no.2, 2002, pp.257–266.

[8] J.Kennedy, R.C.Eberhart, “Partical Swarm Optimization”. In *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp.1942–1948.

[9] F.van den Berh, A.P.Engelbrecht, “A cooperative approach to particle swarm optimization”.*IEEE Transaction on Evolutionary Computation*, vol.8, no.3, 2004, pp.225–239.

[10] Y.H.Shi, R.C.Eberhart, “A modified particle swarm optimizer,” In *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, AK, 1998, pp.69–73.

[11] A.A.A.EI-Gammal, A.A.EI-Samahy, “A modified design of PID controller for DC motor drives using Particle Swarm Optimization PSO,” *International Conference on Power Engineering, Energy and Electrical Drives*, Lisbon, Portugal, 2009, pp.419–424.