International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:8, 2011

Reentry Trajectory Optimization Based on Differential Evolution

Songtao Chang, Yongji Wang, Lei Liu, and Dangjun Zhao

Abstract—Reentry trajectory optimization is a multi-constraints optimal control problem which is hard to solve. To tackle it, we proposed a new algorithm named CDEN(Constrained Differential Evolution Newton-Raphson Algorithm) based on Differential Evolution(DE) and Newton-Raphson. We transform the infinite dimensional optimal control problem to parameter optimization which is finite dimensional by discretize control parameter. In order to simplify the problem, we figure out the control parameter's scope by process constraints. To handle constraints, we proposed a parameterless constraints handle process. Through comprehensive analyze the problem, we use a new algorithm integrated by DE and Newton-Raphson to solve it. It is validated by a reentry vehicle X-33, simulation results indicated that the algorithm is effective and robust.

Keywords—reentry vehicle, trajectory optimization, constraint optimal, differential evolution.

I. Introduction

As one of critical technologies of advanced vehicle design, reentry trajectory optimization is an important component of vehicle design. It is formulated to a nonlinear multi-constraints optimal control problem. Optimal control problem can be solved by applying the calculus of variance and Pontryagin's maximum principle usually. Then the problem becomes a two-point boundary value problem. It is difficult to solve analytically, therefore numerical techniques are required to determine an approximation to the continuous solution. Numerical methods are normally divided into indirect methods and direct methods.

Indirect methods[1, 2] transform the original optimal problem to two-point boundary problem, and using discrete points to approximate the continuous solution. It has high accuracy and assurances the solution satisfies the necessary optimality conditions. However the radius of convergence is small, the co-states is difficult to guess.

Direct methods overcome some of the deficiencies of indirect methods by transform the continuous optimal control problem into a parameter optimization problem, which can be solved by nonlinear programming algorithm. There are two mainly types of transform methods: collection method[3] , pseudospectral method[4] . Collection method disperse control variable and states simultaneously, transform state equations and constrains to algebraic equations. Pseudospectral method using orthogonal polynomials to approximate the differential equations at collocation points.

The authors are with the Department of Control Science and Engineering, Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, Huazhong University of Science and Technology Wuhan, 430074, China. (e-mail: Wangyjch@hust.edu.cn)

Due to determinate algorithm is sensitive and stagnated at local optimal point easily, some researchers applied intelligent algorithm to solve reentry trajectory optimization problem[5, 6, 7] . Arora set bank angle as zero, only angle of attack is optimized and using genetic algorithm to solve it[5] . Zhang transformed control variable optimization to drag optimization to simplify original problem, then using ant colony algorithm to solve it[6] . Chen optimal angle of attack and bank angle simultaneously using genetic algorithm, but bank angle is always positive, without considering lateral motion[7] .

Differential evolution[8] is a simple, efficient and robust evolutionary algorithm. It is widely used in a number of scientific and engineering fields.

First, we transform the infinite dimensional optimal control problem to parameter optimization which is finite dimensional by discretize control parameters. It is difficult to optimize angle of attack and bank angle simultaneously, so many researches only optimize one of the two control variables. This paper, we address the problem by figure out the scope of angle of attack to simplify the transformed nonlinear programming problem. Considering lateral motion, we use two bank angle reverse points to control lateral motion, and add them to the decision vector. Classic differential evolution can only solve nonrestraint parameter optimal problem. We incorporated a parameterless constraints handle process to classic differential evolution to solve constrained parameter optimization problem. Lateral motion is sensitive to bank angle reverse points, and terminal lateral distance is monotone to the first reverse point, terminal flight head angle is monotone to the second reverse point. We use Newton-Raphson algorithm to optimal the two reverse points. Our algorithm named CDEN incorporated a parameterless constrained handle process and Newton-Raphson algorithm to DE.

II. Problem formulation

A. Model

Three degree of freedom dynamic equations of reentry vehicle is:

$$\frac{dr}{dt} = v \sin\gamma \tag{1}$$

$$\frac{d\theta}{dt} = \frac{v\cos\gamma\sin\psi}{r\cos\phi} \tag{2}$$

$$\frac{d\phi}{dt} = \frac{v\cos\gamma\cos\psi}{r} \tag{3}$$

$$\begin{aligned}\frac{dv}{dt} =&\omega^2 r\cos\phi\left(\sin\gamma\cos\phi-\cos\gamma\sin\phi\cos\psi\right)\\&-D-g\sin\gamma\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{d\gamma}{dt} =&\frac{L\cos\sigma}{v}-\frac{g}{v}\cos\gamma+\frac{v}{r}\cos\gamma+2\omega\cos\phi\sin\psi\\&+\frac{\omega^2 r}{v}\cos\phi\left(\cos\gamma\cos\phi+\sin\gamma\cos\psi\sin\phi\right)\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{d\psi}{dt} =&\frac{v}{r}\cos\gamma\sin\psi\tan\phi+\frac{\omega^2 r}{v\cos\gamma}\sin\psi\sin\phi\cos\phi\\&+\frac{L\sin\sigma}{v\cos\gamma}-2\omega\left(\tan\gamma\cos\psi\cos\phi-\sin\phi\right)\end{aligned} \tag{6}$$

$$\begin{aligned}g=\frac{\mu}{r^2}, \qquad \rho=\rho_0 e^{-\frac{r-r_0}{h_s}},\\ L=\frac{\rho v^2 S}{2m}C_L, \qquad D=\frac{\rho v^2 S}{2m}C_D.\end{aligned} \tag{7}$$

where r is the radial distance from the Earth center to the vehicle, θ and ϕ are the longitude and latitude, v is the Earth-relative velocity, γ is the relative flight-path angle, ψ is the relative velocity heading angle measured clockwise from the north, ω is Earth self-rotation rate, m is vehicle's mass, S is reference area, ρ is density of atmosphere, ρ_0 is density of atmosphere at sealevel, r_0 is Earth's radius, C_L and C_D are lift and drag coefficient determined by angle of attack α and Ma, g is gravity acceleration, t is time, control variables are angle of attack α and bank angle σ.

Reentry process must satisfy heating rate, dynamic pressure and normal acceleration constraints:

$$\dot{Q}=c\sqrt{\rho}v^{k_q}\le\dot{Q}_{max}. \tag{8}$$

$$q=\frac{\rho v^2}{2}\le q_{max}. \tag{9}$$

$$n=L\cos\alpha+D\sin\alpha\le n_{max}. \tag{10}$$

where $\dot{Q}_{max}$, q_{max}, n_{max} represent allowable maximum heating rate, dynamic pressure and acceleration separately.

In order to convenience design controller, reentry trajectory should not oscillate acutely($\gamma\approx 0$, $\dot{\gamma}\approx 0$), should satisfy Quasi-Equilibrium Glide Condition(QEGC) as Eq. (11). Let $\gamma=0$, $\dot{\gamma}=0$, ignore earth rotation(let $\omega=0$), from Eq. (5), we can get Eq. (11).

$$L\cos\sigma-g+\frac{v^2}{r}=0. \tag{11}$$

Heating rate, dynamic pressure and normal acceleration constraints are hard constraints, must be satisfied. QEGC is soft constraint, may be violated a little.

Following reentry phase is Terminal Area Energy Management(TAEM), reentry terminate point must be restrained:

$$\begin{aligned}|r(t_f)-r_f|\le\Delta r,\ |\theta(t_f)-\theta_f|\le\Delta\theta,\\ |\phi(t_f)-\phi_f|\le\Delta\phi,\ |v(t_f)-v_f|\le\Delta v,\\ |\gamma(t_f)-\gamma_f|\le\Delta\gamma,\ |\psi(t_f)-\psi_f|\le\Delta\psi.\end{aligned} \tag{12}$$

where t_f is reentry terminate time, $r_f,\theta_f,\phi_f,v_f,\gamma_f,\psi_f$ represent ideal reentry terminate state.

B. Trajectory optimization formulation

Let J represent objective function:

$$J=\chi\left(\mathbf{S}\left(t_f\right),t_f\right)+\int_{t_0}^{t_f} g\left(\mathbf{S},t\right)dt \tag{13}$$

where $\mathbf{S}=[r,\theta,\phi,v,\gamma,\psi]$, t_0,t_f represent reentry time and reentry terminate time respectively.

Reentry trajectory optimization formulate to an optimal control problem which satisfy constraints Eq. (8)-Eq. (12), dynamic characterize as Eq. (1)-Eq. (6) and optimal objective described by Eq. (13).

III. Trajectory optimization algorithm

The original optimal control problem has infinite dimensions, we transformed it to finite dimensional parameter optimal problem by discretize angle of attack and bank angle simultaneous. It is difficult to optimize angle of attack and bank angle simultaneous, we figure out the angle of attack's scope by process constraints(Eq. (8)-Eq. (11)) to simplify the problem. In order to handle constraints, we introduce a parameterless constraints handle process to DE algorithm. Lateral motion include terminal cross-range and terminal flight heading angle. We used two bank angle reverse points to adjust lateral motion. Lateral motion is sensitive to bank angle reverse points, and terminal lateral distance is monotone to the first reverse point, terminal flight head angle is monotone to the second reverse point. Through the analysis, we find the two bank angle reverse points are difficult for DE to optimal, here we using Newton-Raphson algorithm to solve it efficiently. So we incorporated parameterless constraints handle process and New-Raphson algorithm to DE form a new algorithm CDEN to solve the transformed parameter optimal problem.

A. Control disperse algorithm

From Eq. (7)-Eq. (9) we have:

$$\begin{aligned}r &\ge r_0-2h_s\ln\left(\frac{\dot{Q}_{max}}{\sqrt{\rho_0}cv^{k_q}}\right)\triangleq r_Q\left(v\right)\\ r &\ge r_0-h_s\ln\frac{2q_{max}}{\rho_0 v^2}\triangleq r_q\left(v\right)\end{aligned} \tag{14}$$

Let $\sigma=0$, from Eq. (11) we have $\alpha=\alpha(v,r)$. Assign $y=L-g+\frac{v^2}{r}$, then

$$\frac{\partial y}{\partial r}=\frac{2g}{r}-\frac{v^2}{r^2}-\frac{L}{h_s}+\frac{L}{C_L}\frac{\partial C_L}{\partial r},\quad \frac{\partial y}{\partial\alpha}=\frac{\rho v^2 S}{2m}\frac{\partial C_L}{\partial\alpha}$$

Usually lift coefficient C_L is increase as α increase, so $\partial y/\partial\alpha>0$. $\partial C_L/\partial r$ is comparable small, we can ignore it, $2g/r$ is relatively small compare to L/h_s, we can also ignore it, so $\partial y/\partial r<0$, $\dfrac{\partial\alpha\left(v,r\right)}{\partial r}=-\dfrac{\partial y/\partial r}{\partial y/\partial\alpha}>0$, then

$$\alpha=\alpha\left(v,r\right)\ge\alpha\left(v,r_l\right)\ge\alpha\left(v,\max\left(r_Q\left(v\right),r_q\left(v\right)\right)\right) \tag{15}$$

Suppose maximum allowable angle of attack and bank angle are $\alpha_{\max}$ and $\sigma_{\max}$, number of angle of attack discrete points is N_1, bank angle discrete points number is N_2, then angle of attack and bank angle are described by Eq. (16):

$$\begin{cases} v_k = v_i + k\frac{v_f - v_i}{N_1 - 1}, k = 0, 1, ..., N_1 - 1. \\ \eta = \eta_{k-1} + \frac{\eta_k - \eta_{k-1}}{v_k - v_{k-1}}\left(v - v_{k-1}\right), v \in [v_k, v_{k-1}] \\ \alpha\left(v\right) = \eta.\alpha_{\max} + (1 - \eta) \\ \qquad .\alpha\left(v, \max\left(r_Q\left(v\right), r_q\left(v\right)\right)\right) \end{cases}$$
$$\begin{cases} v_k = v_i + k\frac{v_f - v_i}{N_2 - 1}, k = 0, 1, ..., N_2 - 1. \\ \xi = \xi_{k-1} + \frac{\xi_k - \xi_{k-1}}{v_k - v_{k-1}}\left(v - v_{k-1}\right), v \in [v_k, v_{k-1}] \\ |\sigma\left(v\right)| = \xi.\sigma_{\max} + \left(1 - \xi\right).\sigma_{\min} \\ \sigma = \begin{cases} -|\sigma\left(v\right)|, & e_1 \le e \le e_2 \\ |\sigma\left(v\right)|, & \text{otherwise} \end{cases} \end{cases} \quad (16)$$

where $e = mv^2/2 - m\mu/r$ is vehicle's mechanical energy at present, e_i and e_f represent vehicle's mechanical energy at reentry point and terminal point. Let $\mathbf{X} = [\eta_0, \eta_1, \ldots, \eta_{N-1}, \xi_0, \xi_1, \ldots, \xi_{N-1}, e_1, e_2]$, $\eta_k, \xi_k \in [0, 1]$, $e_1, e_2 \in [e_f, e_i]$. e_1 and e_2 corresponding to the two bank angle reverse points, are used to adjust lateral motion. Vector $\mathbf{X}$ determine all control variables angle of attack and bank angle. Then we can integrate Eq. (1)-Eq. (6) to figure out objective function Eq. (13) and constraints Eq. (8)-Eq. (12).

B. Classic differential evolution

Unconstraint parameter optimization can be formulated to minimum a single D-dimensional function $f(\mathbf{X})$, $\mathbf{X} = [x_1, x_2, \ldots, x_D]$, $x_i \in [x_{l,i}, x_{u,i}]$. Let $P_G = \{\mathbf{X}_{G,1}, \mathbf{X}_{G,2}, \ldots, \mathbf{X}_{G,NP}\}$ represent G-th population, where i-th individual's position is $\mathbf{X}_{G,i} = [x_{G,i,1}, x_{G,i,2}, \ldots, x_{G,i,D}]$, $i = 1, 2, \ldots NP$, NP is the population's scale.

DE involves two stages: initialization and evolution. Initialization stage using Eq. (17) generates initial population P_0 randomly. Then P_0 evolves to P_1, P_1 evolves to P_2, ..., and so on, until the termination conditions are fulfilled. Each evolve process include three operation, namely differential mutation, crossover and selection.

$$\begin{gathered} x_{G,i,j} = x_{l,j} + \text{rand}[0, 1] \times \left(x_{u,j} - x_{l,j}\right), \\ i = 1, 2, \ldots, NP, \; j = 1, 2, \ldots D \end{gathered} \quad (17)$$

where rand$[0, 1]$ is a uniform random number in $[0, 1]$.

1) Mutation: Let $\mathbf{V}_{G,i} = [v_{G,i,1}, v_{G,i,2}, \ldots, v_{G,i,D}]$ denotes i-th mutant individual. There are many mutation strategies in the literature. Among them, the commonly used operator is "DE/rand/1", which is described as:

$$\mathbf{V}_{G,i} = \mathbf{X}_{G,r_1} + F \times \left(\mathbf{X}_{G,r_2} - \mathbf{X}_{G,r_3}\right) \quad (18)$$

where r_1, r_2, r_3 are mutually exclusive indices randomly chosen in the range $[1, NP]$, which is different from base vector index i. $F > 0$ is mutant scale factor.

2) Crossover: In order to enhance the diversity of the population, a crossover operation comes into play after generating the mutant individual. Mutant individual $\mathbf{V}_{G,i}$ exchanges its components with base vector $\mathbf{X}_{G,i}$ through crossover operation get the trial vector $\mathbf{U}_{G,i} = [u_{G,i,1}, u_{G,i,2}, \ldots, u_{G,i,D}]$. Common crossover operation is described as:

$$u_{G,i,j} = \begin{cases} v_{G,i,j}, & j = j_{rand} \text{ or rand}[0, 1] \le C_R \\ x_{G,i,j}, & \text{otherwise} \end{cases} \quad (19)$$

where j_{rand} is a random index in the range [1,D], it ensure the trial vector has at least one component from mutant vector. $C_R > 0$ is crossover probability.

3) Selection: Using one to one greedy selection operation decide whether trial vector $\mathbf{U}_{G,i}$ substitute $\mathbf{X}_{G,i}$ to next iteration. It is described as:

$$\mathbf{X}_{G+1,i} = \begin{cases} \mathbf{U}_{G,i}, & f(\mathbf{U}_{G,i}) \le f(\mathbf{X}_{G,i}) \\ \mathbf{X}_{G,i}, & \text{otherwise} \end{cases} \quad (20)$$

C. Constraints handling

Many researchers added a penalty function to the objective function to deal with constrained optimization. But it is difficult to choose adequate coefficient of each constraint. Here, we use a simple parameterless constraints handle process. Let $\mathbf{C}_{G,i} = [c_{G,i,1}, c_{G,i,2}, \ldots, c_{G,i,m}], i = 1, 2, \ldots, NP$ represent each individual's m-dimensional constraints. If violate the constraint its corresponding component is positive, otherwise is zero. Due to $\mathbf{C}_{G,i}$ is multidimensional, it is hard to compare between two constraints. We transform it to a scalar using dimension scale method. Define $CM_{G,i}$ as i-th individual's constraint scalar to measure violate constraints. Let $D_G = \{\mathbf{U}_{G,1}, \mathbf{U}_{G,2}, \ldots, \mathbf{U}_{G,NP}\}$ denote offspring population, and i-th offspring's constrain vector is $CD_{G,i} = [cd_{G,i,1}, cd_{G,i,2}, \ldots, cd_{G,i,m}]$, constraint scalar is $CDM_{G,i}$. Tab. I described how to calculate $CM_{G,i}$ and $CDM_{G,i}$.

Tab. I. Pseudo-code to calculate constraint scalar

$CM_{G,i} = 0$, $CDM_{G,i} = 0$, $i = 1, 2, \ldots, NP$
for k = 1 to m a. Find maximum k-th dimension constraint(denoted as $c_{\max,k}$) in P_G and D_G b. for i = 1 to NP $CM_{G,i} = CM_{G,i} + \frac{c_{G,i,k}}{c_{\max,k} + eps}$, $CDM_{G,i} = CDM_{G,i} + \frac{cd_{G,i,k}}{c_{\max,k} + eps}$ where eps is a small positive number to avoid singularity. end for end for

D. CDEN algorithm

Mutation operation is critical to DE algorithm's quality, here we choose a new mutation operation described by Eq. (21)[9] . This mutation operation balanced exploit ability and explore ability, has more robustness than the classical DE mutation operation.

CDEN algorithm is described by Tab. II. We regard the two bank angle reverse points as population's common information shared by each individual. First, initialization the population and calculate the corresponding objective function and constraints. Then evolve the population. Before each evolve process, we figure ...

For each individual do mutation operation and crossover operation to create trial vector and calculate the corresponding objective and constraints, calculate each individual's constraint scalar, using selection operation to decide whether trial vector is substitute base vector to the next iteration.

Tab. II. Pseudo-code of CDEN algorithm

step 1 : Assign DE's parameter NP, F, C_R, $G = 0$.
  for i = 1 to NP
    Initialize population P_0 according to Eq. (17) and calculate the corresponding objective function $f\left(\mathbf{X}_{G,i}\right)$ and constraints $\mathbf{C}_{G,i}$
  end for
step 2 : Using Newton-Raphson algorithm determinate the two bank angle reverse points e_1, e_2 with the best individual in the current population P_G.
  for i = 1 to NP
    step 2.1 : Generate mutant vector $\mathbf{V}_{G,i}$ using Eq. (21),
    step 2.2 : Generate trial vector $\mathbf{U}_{G,i}$ using Eq. (19),
    step 2.3 : Calculate objective function $f\left(\mathbf{U}_{G,i}\right)$ and constraints $\mathbf{CD}_{G,i}$.
  end for
step 3 : Calculate $CM_{G,i}, CDM_{G,i}, i = 1, 2, \ldots, NP$ using the algorithm described by section III-C.
step 4 : for i = 1 to NP
    Selection a individual between trial vector $\mathbf{U}_{G,i}$ and base vector $\mathbf{X}_{G,i}$ to the next iteration using Eq. (22).
  end for
step 5 : $G = G + 1$, if terminate condition is satisfied output the best individual in the current population, otherwise goto step2.

$$\mathbf{V}_{G,i} = \mathbf{X}_{G,n_1} + F \times \left(\mathbf{X}_{G,n_2} - \mathbf{X}_{G,n_3}\right) \tag{21}$$

where r_1, r_2, r_3 are mutually exclusive indices randomly chosen in the range $[1, NP]$,which is different from base vector index i. n_1 is the best index of r_1, r_2, r_3 , n_2, n_3 are different from n_1 of r_1, r_2, r_3 . $F > 0$ is scale factor.

$$\mathbf{X}_{G+1,i} = \begin{cases} \mathbf{U}_{G,i}, & CDM_{G,i} < CD_{G,i} \text{ or } \big(CDM_{G,i} = CD_{G,i} \text{ and } f(\mathbf{U}_{G,i}) \le f(\mathbf{X}_{G,i})\big) \\ \mathbf{X}_{G,i}, & \text{otherwise} \end{cases} \tag{22}$$

IV. Simulation

We used X-33[10, 11] to validate our algorithm CDEN, the optimal objective is minimize flight path angle variance, described by Eq. (23). The smaller flight path angle variance, the flatness the trajectory have.

$$J = \sum_{i=1}^{L-1} \left|\gamma_{i+1} - \gamma_i\right| \tag{23}$$

where L is integrate number between reentry point to reentry terminate point.
Parameters Setting:

$$\begin{aligned}[r_i, \theta_i, \phi_i, v_i, \gamma_i, \psi_i] &= [6499518 \text{ m}, -117.01^0, -18.255^0, \\ &\quad 7622 \text{ m/s}, -1.4379^0, 38.329^0] \\ [r_f, \theta_f, \phi_f, v_f, \gamma_f, \psi_f] &= [6408427 \text{ m}, -80.48^0, \\ &\quad 28.6112^0, 908 \text{ m/s}, -7.5^0, 40.0^0] \\ [\Delta r, \Delta\theta, \Delta\phi, \Delta v, \Delta\gamma, \Delta\psi] &= [2000 \text{ m}, 0.08^0, 0.08^0, 20 \text{ m/s}, \\ &\quad 1.0^0, 5.0^0] \end{aligned}$$

$Q_{max} = 431259\,\mathrm{W/m^2},\ q_{max} = 11970\,\mathrm{N/m^2},\ n_{max} = 2.5\,\mathrm{g}$
$5^0 \le \alpha \le 50^0,\ |\dot{\alpha}| \le 5^0/\mathrm{s},\ |\sigma| \le 80^0,\ |\dot{\sigma}| \le 5^0/\mathrm{s}$
Attack angle discrete points number is 5, bank angle discrete points number is 12. DE algorithm parameters setting: $NP = 75, F = 0.6, C_R = 0.9$, maximum iteration is 300.

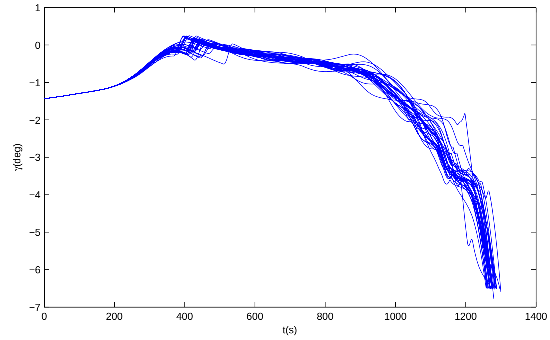

Fig. 1. Flight path angle

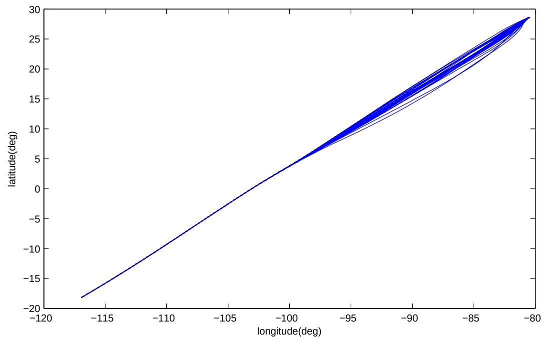

Fig. 2. Longitude and Latitude

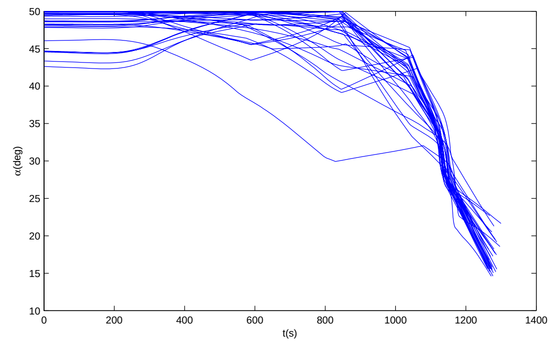

Fig. 3. Attack angle

Because CDEN using random searching algorithm DE as its critical component, each run has different performance. We run 30 times to evaluate it. The results described by Tab. III-Tab. IV, trajectory depicted by Fig. 1-Fig. 4. As can be see from the tables and the figures, all process constraints and terminative

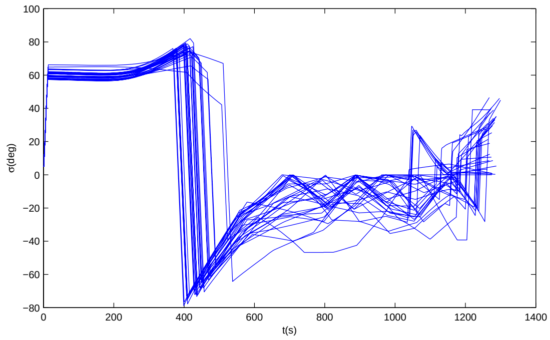

Fig. 4. Bank angle

Tab. III. Objective and Partial Constraints

	J (rad)	$\max(\dot{Q})$ ($\mathrm{W/m^2}$)	$\max(q)$ ($\mathrm{N/m^2}$)	$\max(n)$ (g)	$\|\Delta r(t_f)\|$ (m)
Max	0.1705	423201	10157	1.552	1982
Mean	0.1488	392097	6314	1.329	848
Min	0.1433	375510	5180	1.197	112
Std	0.0058	12288	917	0.097	525

Tab. IV. Terminate Constraints

	$\|\Delta\theta(t_f)\|$ (degree)	$\|\Delta\phi(t_f)\|$ (degree)	$\|\Delta v(t_f)\|$ (m/s)	$\|\Delta\gamma(t_f)\|$ (degree)	$\|\Delta\psi(t_f)\|$ (degree)
Max	0.067	0.053	19.7	1.00	4.9
Mean	0.029	0.019	11.3	0.98	2.7
Min	0.001	0.001	1.4	0.72	0.1
Std	0.018	0.013	4.9	0.05	1.4

constraints satisfied, each trajectory is flatness and objective function has small variance. It is indicated that our algorithm is efficient to handle constraints, has good performance and robustness.

V. Conclusion

In order to tackle the complexity of optimize angle of attack and bank angle simultaneously, we figure out the control parameter's scope. We incorporated a parameterless constraints handle process and Newton-Raphson algorithm to DE algorithm form a new algorithm CDEN to solve reentry trajectory optimization. It is validated by X-33, simulation results indicated that CDEN is effective and robust.

Acknowledgment

This work is partially supported by National Nature Science Foundation of China, NO 60975058; Nature Science Foundation of Hubei, NO 2010CDB01904; 2010 Aerospace Technology Support Foundation.

References

[1] N. X. Vinh, *Optimal Trajectories in Atmospheric Flight*. New York: Elsevier, 1981.

[2] V. Istratie, "Three-dimensional optimal skip entry with terminal maximum velocity." AIAA-97-3483, 1997.

[3] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.

[4] K. Zhang and W. Chen, "Reentry vehicle constrained trajectory optimization." San Francisco, California: AIAA 2011-2231, April 2011.

[5] R. K. Arora, "Reentry trajectory optimization : Evolutionary approach." Atlanta: AIAA 2002-5466, September 2002.

[6] Q. Zhang, C. Liu, B. Yang, and Z. Ren, "Reentry trajectory planning optimization based on ant colony algorithm," in *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*, Sanya, December 2007, pp. 1064–1068.

[7] G. Chen, Z. ming Wan, M. Xu, and S. lu Chen, "Genetic algorithm optimization of RLV reentry trajectory." AIAA 2005-3269, 2005.

[8] R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[9] C. Lin, A. Qing, and Q. Feng, "A new differential mutation base generator for differential evolution," *Journal of Global Optimization*, vol. 49, pp. 69–90, 2011.

[10] P. Lu and J. M. Hanson, "Entry guidance for the X-33 vehicle," *Journal of Spacecraft and Rockets*, vol. 35, no. 3, pp. 342–349, 1998.

[11] Z. Shen and P. Lu, "Onboard generation of three-dimensional constrained entry trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 1, pp. 111–121, 2003.