4D Trajectory Tracking Control for Aircraft Based on Point-to-Point Iterative Learning Control with Current-Cycle Feedback

JIANG Gaoyang[1*], *WANG Hongyong*[2]

1. College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, P. R. China;
2. Tianjin Key Laboratory for Air Traffic Operation Planning and Safety Technology, Tianjin 300300, P. R. China

(Received 31 March 2021; revised 7 September 2021; accepted 10 December 2021)

Abstract: A point-to-point iterative learning control method with the current-cycle feedback is proposed to enable aircraft to achieve an accurate four-dimensional (4D) trajectory tracking. To this end, the 4D trajectory tracking control problem is formulated into a point-to-point tracking control issue with an external disturbance. Then, the optimal point-to-point iterative learning control law is derived based on the successive projection method. Further, the current-cycle feedback error is added to the control law, so that the tracking error is reduced in both time and iteration domains. Finally, a numerical simulation is carried out using the kinematic model of an unmanned aerial vehicle and 4D trajectory data. Obtained results demonstrate that the proposed method can quickly reduce the trajectory tracking error even in the presence of gust interferences. Compared with the commonly used average velocity method and the velocity correction method, the proposed method makes full use of the past and current running data, and can continuously improve the accuracy of 4D trajectory tracking with the repetitive operation of aircraft between city pairs.

Key words: four-dimentional(4D) trajectory; trajectory tracking; iterative learning control; trajectory-based operation; controlled time of arrival

CLC number: TP273 **Document code:** A **Article ID:** 1005-1120(2021)06-0937-11

0 Introduction

With the rapid development of civil aviation industry, the number of airlines is rapidly increasing. Accordingly, the air traffic has been characterized by large flow, high flight density and small intervals between flights. To improve the utilization rate of airspace resources and achieve an accurate flight control, the concept of trajectory-based operation (TBO), also known as four-dimensional(4D) trajectory, has been put forward in Europe and the United States. This concept can also be interpreted by adding a controlled time of arrival (CTA) on a three-dimensional(3D) trajectory. Generally, CTA is attached to the key route points, indicating that the aircraft should reach the designated way-points at the specified time along the planned route. In this concept, it is essential to study the 4D trajectory tracking control problem. It is of great significance to the trajectory-based operation paradigm, including the future air traffic operation mode.

Unlike conventional problems in 4D trajectory planning[1-3] and trajectory prediction[4-6], the 4D trajectory tracking control focuses on how to control the aircraft accurately to achieve an appropriate flight along the planned route. From the perspective of equipment, the flight management system with the required time of arrival (RTA) function can control the aircraft to fly along the 4D trajectory. However, not all existing aircraft are equipped with the RTA function[7]. From the algorithm perspective, existing control algorithms, such as the backstep-

*Corresponding author, E-mail address: gy-jiang@cauc.edu.cn.

How to cite this article: JIANG Gaoyang, WANG Hongyong. 4D trajectory tracking control for aircraft based on point-to-point iterative learning control with current-cycle feedback[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2021, 38(6): 937-947.

http://dx.doi.org/10.16356/j.1005-1120.2021.06.004

ping control [8], the adaptive control[9-10], the model predictive control[11] and the sliding mode control[12] algorithms, can achieve a high-precision trajectory tracking control. However, these algorithms ignore the value of historical data, so that the control performance cannot be further improved by historical data generated during the repeated operation of the system.

It is worth noting that there are periodicities and similarities in the flight operations between city pairs. In other words, flights must strictly follow the established flight schedule from one city to another, and the flight cycle is usually about a few days or weeks. It should be indicated that a large amount of historical data is generated in the continuous round-trip operation, which contains useful information for improving flight quality. However, conventional control algorithms cannot learn and utilize historical data. In today's big data era, using data as a driving resource in the process of smart civil aviation has become a concern in the aviation industry[13]. Furthermore, temporary or non-repetitive interference inevitably appears in the round-trip operation. Accordingly, removing non-repetition interference on the time axis is another concern in the repetitive system control.

Based on the foregoing analysis, the point-to-point iterative learning control with current-cycle feedback (PTP-ILC-CF) is adopted in this study to resolve the aircraft 4D trajectory tracking control problem. The advantages of the proposed method over conventional methods are in three aspects. First, it improves the control accuracy of the aircraft to the designated waypoint at a specific time. To this end, previous iteration data are used. Second, the proposed method can reduce the trajectory tracking error along the time axis in comparison with the conventional iterative learning control (ILC) method[14]. Third, the proposed method can be directly applied to point-to-point iterative learning control. This is not practical in real-time ILC[15] and current-cycle ILC methods[16].

The rest of the paper is organised as follows. The problem formulation is presented in section 1. The point-to-point iterative learning control law with current-cycle feedback is derived in section 2. A case study is illustrated in section 3 to evaluate the performance of the proposed method, and results from two commonly used methods and the proposed method are compared. Finally, the main achievements and conclusions are presented in section 4.

1 Problem Formulation

In the 4D trajectory operation, different arrival times will be attached to key waypoints, and the operation goal is to control the aircraft to arrive at the designated waypoint at the designated time, as illustrated in Fig.1.

Normally, the pilot (or autopilot) will control the aircraft in each flight segment to fly at a constant speed that is calculated from the distance traveled and the corresponding traveling time. However, when there is an en-route wind interference, this simple calculation cannot guarantee that the aircraft will strictly follow the specified time to each waypoint. Therefore, it is essential to find an effective way to comply with the operation control goal of 4D trajectory.

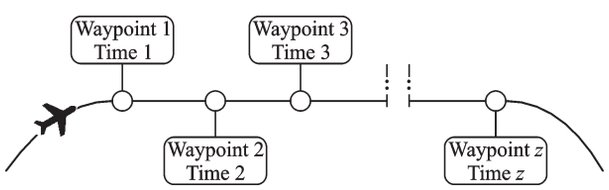

Fig.1 Waypoints with controlled times of arrival

From the perspective of tracking control, 4D trajectory operation is a typical point-to-point tracking control problem. Specifically, an appropriate control algorithm needs to be designed to continuously generate aircraft control inputs (such as speed), so that the aircraft can arrive at various waypoints on time. To achieve this goal, the kinematic equation of the aircraft is expressed by the following discrete-time linear system

$$\boldsymbol{x}(t+1)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}(t)u(t)+\boldsymbol{d}(t) \quad (1)$$

where t represents the tth sampling time; $\boldsymbol{x}(t)\in\mathbf{R}^m$ the state variable; $u(t)\in\mathbf{R}$ the control input; $\boldsymbol{d}(t)\in\mathbf{R}^m$ the repetitive external interference; $\boldsymbol{A}\in\mathbf{R}^{m\times m}$ the system matrix, and $\boldsymbol{B}(t)\in\mathbf{R}^m$ the in-

put matrix.

Eq.(1) can be rewritten as follows by using the lift technique

$$\boldsymbol{r}=\boldsymbol{E}\boldsymbol{x}_0+\boldsymbol{G}\boldsymbol{u}+\boldsymbol{F}\boldsymbol{d} \tag{2}$$

where

$$\boldsymbol{r}=[\boldsymbol{x}^{\mathrm{T}}(1),\boldsymbol{x}^{\mathrm{T}}(2),\boldsymbol{x}^{\mathrm{T}}(3),\cdots,\boldsymbol{x}^{\mathrm{T}}(n)]^{\mathrm{T}}\in\mathbf{R}^{mn} \tag{3}$$

$$\boldsymbol{E}=\begin{bmatrix}\boldsymbol{A} & & & & \\ & \boldsymbol{A}^2 & & & \\ & & \boldsymbol{A}^3 & & \\ & & & \ddots & \\ & & & & \boldsymbol{A}^n\end{bmatrix}\in\mathbf{R}^{mn\times mn} \tag{4}$$

$$\boldsymbol{x}_0=[\boldsymbol{x}^{\mathrm{T}}(0),\boldsymbol{x}^{\mathrm{T}}(0),\boldsymbol{x}^{\mathrm{T}}(0),\cdots,\boldsymbol{x}^{\mathrm{T}}(0)]^{\mathrm{T}}\in\mathbf{R}^{mn} \tag{5}$$

$$\boldsymbol{G}=\begin{bmatrix}\boldsymbol{B}(0) & & & & \\ \boldsymbol{A}\boldsymbol{B}(0) & \boldsymbol{B}(1) & & & \\ \boldsymbol{A}^2\boldsymbol{B}(0) & \boldsymbol{A}\boldsymbol{B}(1) & \boldsymbol{B}(2) & & \\ \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{A}^{n-1}\boldsymbol{B}(0) & \boldsymbol{A}^{n-2}\boldsymbol{B}(1) & \cdots & & \boldsymbol{B}(n-1)\end{bmatrix}\in\mathbf{R}^{mn\times n} \tag{6}$$

$$\boldsymbol{u}=[u(0),u(1),u(2),\cdots,u(n-1)]^{\mathrm{T}}\in\mathbf{R}^{n} \tag{7}$$

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{I} & & & & \\ \boldsymbol{A} & \boldsymbol{I} & & & \\ \boldsymbol{A}^2 & \boldsymbol{A} & \boldsymbol{I} & & \\ \vdots & \vdots & & \ddots & \\ \boldsymbol{A}^{n-1} & \boldsymbol{A}^{n-2} & \cdots & & \boldsymbol{I}\end{bmatrix}\in\mathbf{R}^{mn\times mn} \tag{8}$$

$$\boldsymbol{d}=[\boldsymbol{d}^{\mathrm{T}}(0),\boldsymbol{d}^{\mathrm{T}}(1),\boldsymbol{d}^{\mathrm{T}}(2),\cdots,\boldsymbol{d}^{\mathrm{T}}(n-1)]^{\mathrm{T}}\in\mathbf{R}^{mn} \tag{9}$$

In Eq.(2), $\boldsymbol{r}$ is the set of aircraft states at each time in the interval $[1, n]$; $\boldsymbol{x}_0$ the set of aircraft initial states; $\boldsymbol{u}$ and $\boldsymbol{F}$ are the set of control input and the external repetitive interference at each time in the interval $[0, n-1]$, respectively; $\boldsymbol{E}$, $\boldsymbol{G}$, and $\boldsymbol{F}$ the coefficient matrices; and $\boldsymbol{d}$ is the repetitive disturbance along the iterative axis, and can be well eliminated by iterative learning control. When non-repetitive interference is added on this basis, the traditional iterative learning control cannot achieve an accurate tracking. This issue is explained in section 2.3.

According to the description of the 4D trajectory tracking control problem, the control goal of system in Eq.(2) is to find a series of control input $\boldsymbol{u}$, so that the system output $\boldsymbol{r}$ can reach the designated position at the designated time. From the perspective of tracking control, when the control input $\boldsymbol{u}$ approaches the optimal control input $\boldsymbol{u}^*$, the tracking error of the system in Eq.(2) at the designated waypoint will tend to 0. Since we do not care about the aircraft flight status between two waypoints, but only consider the tracking errors at several waypoints, the 4D trajectory tracking issue could be addressed using the point-to-point iterative learning control method.

2 Point-to-Point Iterative Learning Control

2.1 Objective

Eq.(3) shows that the output vector $\boldsymbol{r}$ contains n output points. Without loss of generality, we suppose that among the n output points, only z points have an arrival time constraint ($z<n$). Let the control arrival time be τ_1, τ_2, …, τ_z, the expected reference trajectory is

$$\boldsymbol{r}_d=[\boldsymbol{x}_d^{\mathrm{T}}(\tau_1),\boldsymbol{x}_d^{\mathrm{T}}(\tau_2),\cdots,\boldsymbol{x}_d^{\mathrm{T}}(\tau_z)]^{\mathrm{T}}\in\mathbf{R}^{mz} \tag{10}$$

In order to extract the positions at the required arrival time, a matrix $\boldsymbol{\Phi}\in\mathbf{R}^{mz\times mn}$ is defined as

$$\boldsymbol{\Phi}=\begin{bmatrix}\phi_{11} & \cdots & \phi_{1n}\\ \vdots & \ddots & \vdots\\ \phi_{z1} & \cdots & \phi_{zn}\end{bmatrix} \tag{11}$$

where

$$\phi_{ij}:=\begin{cases}\boldsymbol{I}_{m\times m} & i=1,2,\cdots,z;\ j=\tau_i\\ \boldsymbol{0}_{m\times m} & \text{Otherwise}\end{cases} \tag{12}$$

Let $\boldsymbol{u}_k$ and $\boldsymbol{x}_k$ be the input and the state vector at the kth iteration, respectively, and the position error at the required arrival time at the kth iteration can be defined as the difference between the reference trajectory and the actual position

$$\bar{\boldsymbol{e}}_k=(\boldsymbol{r}_d-\boldsymbol{\Phi}\boldsymbol{r}_k) \tag{13}$$

The objective of the point-to-point iterative learning control is to continuously update the control input by using the data from the last operation, so that when k tends to infinity, the point-to-point tracking error tends to 0.

$$\lim_{k\to\infty}\left\|(\boldsymbol{r}_d-\boldsymbol{\Phi}\boldsymbol{r}_k)\right\|=0 \tag{14}$$

When the point-to-point tracking error tends to 0, the control input tends to the ideal value, as expressed in Eq.(15). It should be noted that the ideal control input $\boldsymbol{u}^*$ certainly exists, because the 4D ref-

erence trajectory is generated with full consideration of aircraft performance. This means that there must be an optimal control input for the pilot or the autopilot to enable the precise 4D trajectory tracking.

$$\lim_{k\to\infty}\left\|\boldsymbol{u}^{*}-\boldsymbol{u}_{k}\right\|=0 \tag{15}$$

In practice, it is obviously not feasible to perform infinite iterations to obtain the optimal value. Therefore, it is necessary to adopt a suitable method to design the iterative learning control law in order to obtain the best tracking effect through the least number of iterations. Next, the successive projection method is adopted to design a point-to-point iterative learning control law with a rapid error convergence.

2.2 Point-to-point ILC control law

The successive projection method was proposed by Owens and Jones[17] in order to find a point in the intersection of two closed, convex sets S_1 and S_2 in certain real Hilbert space H. The basic idea is to define a successive point through the projection of previous iterates onto the convex sets. To be specific, given an initial point k_0 in H, subsequent points are obtained successively by projection of the last point onto one and then the other of the two convex sets. This algorithm is illustrated in Fig.2 and formally expressed as

$$k_{i+1}=\arg\min_{k\in S}\left\|k-k_{i}\right\|^{2}\quad i\geqslant 0 \tag{16}$$

where $S=\{S_1, S_2\}$. Specifically, Eq.(16) helps us quickly find point k_{i+1} from point k_i.

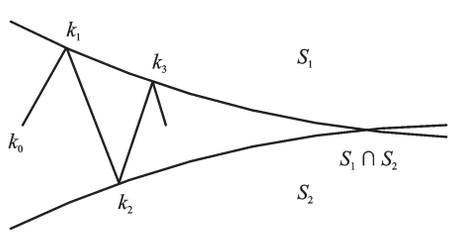

Fig.2 Illustration of the successive projection algorithm

According to the research of Owens and Jones[17], the successive projection algorithm has the following properties. Given the initial point $k_0\in\mathcal{H}$, the sequence $\{k_i\}_{i\geqslant 0}$ satisfies

$$\left\|k_{i+1}-k_{i}\right\|\leqslant\left\|k_{i}-k_{i-1}\right\|\quad i\geqslant 2 \tag{17}$$

and for any $x\in S_1\cap S_2$

$$\left\|x-k_{i}\right\|^{2}\geqslant\left\|x-k_{i+1}\right\|^{2}+\left\|k_{i+1}-k_{i}\right\|^{2} \tag{18}$$

These properties indicate that the point obtained by successive projection is continuously close to the intersection of the two closed convex sets, and finding the intersection in this way is faster than searching directly.

The task of point-to-point ILC is to find a sequence of u_k, such that $u_k\to u^{*}$ when k tends to infinite, and at the same time, the tracking errors at the intermediate pass point $\bar{\boldsymbol{e}}_k\to\boldsymbol{0}$. This is equivalent to alternatively finding a sequence of points between two set S_1 and S_2, so that $(\bar{\boldsymbol{e}}_k,\boldsymbol{u}_k)\to(\boldsymbol{0},\boldsymbol{u}^{*})$ when k tends to infinite.

Based on the above analysis, let S_1 and S_2 be the closed convex sets composed of $(\bar{\boldsymbol{e}}_k,\boldsymbol{u}_k)$ and $(\boldsymbol{0},\boldsymbol{u}_k)$ in real Hilbert space H, respectively, and they are expressed as

$$S_1=\{(\bar{\boldsymbol{e}}_k,\boldsymbol{u}_k)\in\mathcal{H}:\bar{\boldsymbol{e}}_k=(\boldsymbol{r}_d-\boldsymbol{\Phi r}_k)\} \tag{19}$$

$$S_2=\{(\bar{\boldsymbol{e}}_k,\boldsymbol{u})\in\mathcal{H}:\bar{\boldsymbol{e}}_k=\boldsymbol{0}\} \tag{20}$$

$$S_1\bigcap S_2=(\boldsymbol{0},\boldsymbol{u}^{*}) \tag{21}$$

Then according to Eqs.(16,19,20) we have

$$\boldsymbol{u}_{k+1}=\arg\min_{\boldsymbol{u}}\left\{\left\|\boldsymbol{0}-\bar{\boldsymbol{e}}_k\right\|_{\boldsymbol{Q}}^{2}+\left\|\boldsymbol{u}-\boldsymbol{u}_k\right\|_{\boldsymbol{R}}^{2}\right\} \tag{22}$$

Rewrite Eq.(22) as

$$\boldsymbol{u}_{k+1}=\arg\min_{\boldsymbol{u}}\left\{\left\|(\boldsymbol{r}_d-\boldsymbol{\Phi r})\right\|_{\boldsymbol{Q}}^{2}+\left\|\boldsymbol{u}-\boldsymbol{u}_k\right\|_{\boldsymbol{R}}^{2}\right\} \tag{23}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weight matrices. To solve this optimization problem, let

$$J=\left\|\boldsymbol{r}_d-\boldsymbol{\Phi r}\right\|_{\boldsymbol{Q}}^{2}+\left\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\right\|_{\boldsymbol{R}}^{2} \tag{24}$$

By setting $\partial J/\partial\boldsymbol{u}_{k+1}=0$, we can derive the point-to-point iterative learning control law based on the successive projection as

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+[(\boldsymbol{\Phi G})^{\mathrm{T}}\boldsymbol{Q\Phi G}+\boldsymbol{R}]^{-1}\times(\boldsymbol{\Phi G})^{\mathrm{T}}\boldsymbol{Q}\bar{\boldsymbol{e}}_k \tag{25}$$

Using Eq.(25), the control objective of system (2) can be gradually reached

$$\lim_{k\to\infty}\bar{\boldsymbol{e}}_k=0 \tag{26}$$

The error sequence monotonously converges, and the input signal $\boldsymbol{u}_k$ monotonously approximates the optimal input signal $\boldsymbol{u}^{*}$

$$\left\|\bar{\boldsymbol{e}}_{k+1}\right\|\leqslant\left\|\bar{\boldsymbol{e}}_k\right\|\quad\forall k\geqslant 0 \tag{27}$$

$$\left\|\boldsymbol{u}_{k+1}-\boldsymbol{u}^{*}\right\|\leqslant\left\|\boldsymbol{u}_k-\boldsymbol{u}^{*}\right\|\quad\forall k\geqslant 0 \tag{28}$$

Next, the convergence proof of the algorithm is given based on the successive projection.

Proof According to Eqs.(19, 20), let $k_0=$

$(0, \boldsymbol{u}_0)\in S_2$, $k_1=(\bar{\boldsymbol{e}}_1, \boldsymbol{u}_1)\in S_1$, $k_2=(\bar{\boldsymbol{e}}_2, \boldsymbol{u}_2)\in K_2$.

Starting from the initial point $(0, \boldsymbol{u}_0)$, the next point $(\bar{\boldsymbol{e}}_1, \boldsymbol{u}_1)$ can be obtained by one projection, and expressed by

$$(\bar{\boldsymbol{e}}_1, \boldsymbol{u}_1)=\arg\min_{(\boldsymbol{e},\boldsymbol{u})\in S_1}\left\{\|\boldsymbol{e}-0\|_{\boldsymbol{Q}}^2+\|\boldsymbol{u}-\boldsymbol{u}_0\|_{\boldsymbol{R}}^2\right\} \tag{29}$$

Continuing to project, we can get

$$(\bar{\boldsymbol{e}}_2, \boldsymbol{u}_2)=\arg\min_{(\boldsymbol{e},\boldsymbol{u})\in S_2}\left\{\|\boldsymbol{e}-\bar{\boldsymbol{e}}_1\|_{\boldsymbol{Q}}^2+\|\boldsymbol{u}-\boldsymbol{u}_1\|_{\boldsymbol{R}}^2\right\} \tag{30}$$

$(\bar{\boldsymbol{e}}_2, \boldsymbol{u}_2)=(0, \boldsymbol{u}_2)$ because $(\bar{\boldsymbol{e}}_2, \boldsymbol{u}_2)\in S_2$. A series of points can be obtained by successive projection as

$$\{(0, \boldsymbol{u}_0), (\bar{\boldsymbol{e}}_1, \boldsymbol{u}_1), (0, \boldsymbol{u}_1), (\bar{\boldsymbol{e}}_2, \boldsymbol{u}_2), (0, \boldsymbol{u}_2), (\bar{\boldsymbol{e}}_3, \boldsymbol{u}_3), \cdots\}$$

Then, the general representation of the ith point k_i can be obtained as

$$\begin{cases} k_i=(\bar{\boldsymbol{e}}_k, \boldsymbol{u}_k) \\ k_{i+1}=(0, \boldsymbol{u}_k) \\ k_{i+2}=(\bar{\boldsymbol{e}}_{k+1}, \boldsymbol{u}_{k+1}) \\ k_{i+3}=(0, \boldsymbol{u}_{k+1}) \end{cases} \tag{31}$$

where $i>1$ and i is odd.

From Eq.(17), we have

$$\|k_{i+3}-k_{i+2}\|^2\leqslant\|k_{i+2}-k_{i+1}\|^2\leqslant\|k_{i+1}-k_i\|^2 \tag{32}$$

Substituting Eq.(31) into Eq.(32), we can get

$$\|\bar{\boldsymbol{e}}_{k+1}\|^2\leqslant\|\bar{\boldsymbol{e}}_{k+1}\|^2+\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\|^2\leqslant\|\bar{\boldsymbol{e}}_k\|^2 \tag{33}$$

This indicates that the point-to-point tracking error has monotonic convergence.

It can be further derived from Eq.(18) that

$$\|x-k_{i+2}\|^2+\|k_{i+2}-k_{i+1}\|^2\leqslant\|x-k_{i+1}\|^2 \tag{34}$$

Denote $x\in S_1\bigcap S_2=(0, \boldsymbol{u}^*)$, Substituting $(0, \boldsymbol{u}^*)$ and Eq.(31) into Eq.(34), we can get

$$\|\boldsymbol{u}_{k+1}-\boldsymbol{u}^*\|^2\leqslant 2\|\bar{\boldsymbol{e}}_{k+1}\|^2+\|\boldsymbol{u}_{k+1}-\boldsymbol{u}^*\|^2+\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\|^2\leqslant\|\boldsymbol{u}_k-\boldsymbol{u}^*\|^2 \tag{35}$$

This denotes that the input signal $\boldsymbol{u}_k$ can monotonically approximate the optimal input signal $\boldsymbol{u}^*$.

Proof completed.

2.3 Point-to-point ILC control law with current-cycle feedback

The point-to-point iterative learning control law in Eq.(25) based on the successive projection can effectively use the data from the previous run and quickly improve the control performance of the system. However, the algorithm lacks error feedback on the time axis, so that it cannot adjust the control input in time to overcome external non-repetitive disturbances. In this regard, a current-cycle error is designed as

$$\tilde{\boldsymbol{e}}_k(t)=\frac{\bar{\boldsymbol{e}}_k(\tau_i)}{\tau_{i+1}-\tau_i}$$
$$\forall\, t\in[\tau_i, \tau_{i+1}); i=1, 2, \cdots, z-1 \tag{36}$$

where $\tilde{\boldsymbol{e}}_k(\tau_i)$ represents the average tracking error generated at the required arrival time in the kth iteration. In point-to-point tracking control, only a few limited tracking error information at the required arrival time can be obtained, while the tracking error at other times is unknown. This brings difficulties to current-cycle feedback control. Therefore, in Eq.(36), the tracking error at time τ_i is averaged in the interval $[\tau_i, \tau_{i+1}]$, and the average error value will be added to the point-to-point ILC control framework as the current-cycle feedback error.

The complete point-to-point ILC control method with current-cycle feedback can be described as follows.

(1) Generate the control input of the kth iteration.

$$\boldsymbol{u}_k=\boldsymbol{u}_{k-1}+[(\boldsymbol{\Phi G})^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\Phi G}+\boldsymbol{R}]^{-1}\times(\boldsymbol{\Phi G})^{\mathrm{T}}\boldsymbol{Q}\bar{\boldsymbol{e}}_{k-1} \tag{37}$$

(2) Add the average current-cycle feedback to the control input, that is

$$\tilde{u}_k(t)=u_k(t)+\boldsymbol{K}_{\mathrm{RT}}\tilde{\boldsymbol{e}}_k(t) \tag{38}$$

where $\tilde{u}_k(t)$ is the control input with current-cycle feedback, and $\boldsymbol{K}_{\mathrm{RT}}$ the gain of current-cycle feedback error.

Remark 1 Eq.(38) is essentially a feedback control, thus the convergence of Eq.(38) along the time axis can be guaranteed as long as the feedback gain $\boldsymbol{K}_{\mathrm{RT}}$ is chosen properly. In the following simulation, the parameter $\boldsymbol{K}_{\mathrm{RT}}$ is obtained by trial-and-error method. In this method, the parameter $\boldsymbol{K}_{\mathrm{RT}}$ is adjusted at each simulation until the output error is reduced or eliminated satisfactorily.

3 Numerical Simulation

3.1 Existing methods

In order to evaluate the effectiveness of the proposed method, two methods are applied in the simulation and the obtained results are compared.

First, the average velocity method is adopted,

where the average flight velocity of each flight segment is calculated, and then the aircraft is controlled, accordingly to fly at the calculated average velocity. The mathematical expression is

$$V_j = \frac{\left\| x_d(\tau_j) - x_d(\tau_{j-1}) \right\|}{\tau_j - \tau_{j-1}} \quad j = 1, 2, \cdots, z \tag{39}$$

where V_j and τ_j are the average velocity and the required arrival time of flight segment j, respectively. $x_d(\tau_j)$ means the desired position at the required arrival time τ_j

Second, the velocity correction method[18] is used, where the average flight speed is revised at regular intervals to ensure that the aircraft can reach the designated waypoint at the required arrival time. The mathematical expression is

$$S(n\Delta t) = \frac{x_{\mathrm{CTA}} - x_{\mathrm{pred}}}{\mathrm{CTA} - n\Delta t} \quad n\Delta t \leqslant \mathrm{CTA} \tag{40}$$

where S is the velocity correction; x_{CTA} the position at the controlled time of arrival (CTA); x_{pred} the predicted aircraft position; Δt the velocity correction period, and n an integer.

3.2 Kinematic model

The kinematic model of unmanned aviation vehicles[19] is used in this simulation, and its state space equation is in Eq.(1), where

$$\boldsymbol{x}(t) = [x_E(t), y_E(t), z_E(t)]^{\mathrm{T}} \tag{41}$$

$$u(t) = V_T(t) \tag{42}$$

$$\boldsymbol{A} = \boldsymbol{I}_{3\times 3} \tag{43}$$

$$B(t) = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix} \begin{bmatrix} \cos(\alpha(t))\cos(\beta(t)) \\ \sin(\beta(t)) \\ \sin(\alpha(t))\cos(\beta(t)) \end{bmatrix} \Delta t \tag{44}$$

$$B_1 = \begin{bmatrix} \cos(\psi(t))\cos(\theta(t)) \\ \sin(\psi(t))\cos(\theta(t)) \\ -\sin(\theta(t)) \end{bmatrix} \tag{45}$$

$$B_2 = \begin{bmatrix} -\sin(\psi(t))\cos(\varphi(t)) + \cos(\psi(t))\sin(\theta(t))\sin(\varphi(t)) \\ \cos(\psi(t))\cos(\varphi(t)) + \sin(\varphi(t))\sin(\theta(t))\sin(\psi(t)) \\ \cos(\theta(t))\sin(\varphi(t)) \end{bmatrix} \tag{46}$$

$$B_3 = \begin{bmatrix} \sin(\psi(t))\sin(\varphi(t)) + \cos(\psi(t))\cos(\varphi(t))\sin(\theta(t)) \\ -\cos(\psi(t))\sin(\varphi(t)) + \sin(\theta(t))\sin(\psi(t))\cos(\varphi(t)) \\ \cos(\theta(t))\cos(\varphi(t)) \end{bmatrix} \tag{47}$$

$$\boldsymbol{d}(t) = B(t)w(t) \tag{48}$$

where x_E, y_E, z_E represent the x, y, z coordinates in the ground-fixed coordinate system, respectively (m). V_T is the velocity of aircraft (m/s); Δt the sampling time (s); α the angle of attack (rad); β the slide angle (rad); ψ the yaw angle (rad); θ the pitch angle (rad); φ the roll angle (rad); and w the wind speed along the trajectory (m/s), and defined as[20]

$$w(t) = \bar{w} + C\frac{h(t) - \bar{h}}{\Delta h}\cos\left(\pi\frac{t}{t_{\mathrm{A}}}\right) + w_{\mathrm{G},k}(t) \tag{49}$$

where C is speed change coefficient; h the flight altitude; $\bar{h}$ the reference altitude; Δh the altitude increment; t_{A} the total flight time; $\bar{w}$ the wind speed at the reference altitude; and $w_{\mathrm{G},k}$ the gust, which occurs only at the kth iteration.

3.3 Reference trajectory

Fig. 3 shows the designated waypoints and related arrival times along the given heading that the aircraft should reach. Coordinates and controlled arrival time are shown in Table 1[21].

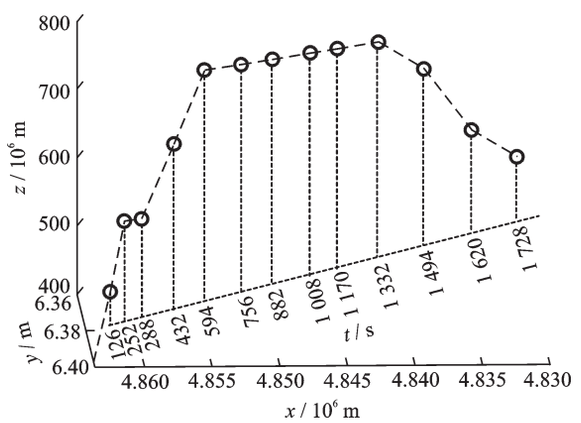

Fig.3 Reference trajectory with a controlled time of arrival

3.4 Simulation results

The initial setting of parameters are as follows. Let the reference altitude be 300 m and the wind speed at this height is 0 m/s. The altitude increment is 1 m, and the wind speed change coefficient C is

Table 1 List of waypoints

Waypoint	Longitude	Latitude	Altitude/m	Time/s
P_0	7°29′35.00″W	39°49′25.71″N	400	0
P_1	7°29′37.00″W	39°50′34.82″N	500	126
P_2	7°29′39.00″W	39°51′33.38″N	600	252
P_3	7°29′41.00″W	39°52′39.86″N	600	288
P_4	7°29′41.50″W	39°54′50.26″N	700	432
P_5	7°29′42.00″W	39°56′55.38″N	800	594
P_6	7°29′45.00″W	39°59′15.04″N	800	756
P_7	7°29′47.00″W	40°01′17.12″N	800	882
P_8	7°29′49.00″W	40°03′45.92″N	800	1 008
P_9	7°29′51.00″W	40°05′31.38″N	800	1 170
P_{10}	7°29′53.00″W	40°08′12.56″N	800	1 332
P_{11}	7°29′55.00″W	40°11′06.43″N	750	1 494
P_{12}	7°30′00.00″W	40°14′07.43″N	650	1 620
P_{13}	7°30′02.00″W	40°17′02.02″N	600	1 728

0.01. Let the gain of the current-cycle feedback be $\boldsymbol{K}_{RT}=[1,1,5]$. Next, the simulation will be carried out under two cases: Case 1 without gust component, and case 2 with gust component.

3.4.1 Case 1 without gust component

The wind speed along the trajectory without gust component is shown in Fig.4. A positive wind speed indicates a tailwind, and a negative value indicates a headwind. This kind of wind can be seen as a repetitive interference in the simulation.

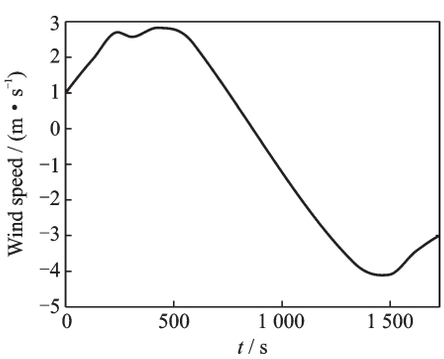

Fig.4 Along-track wind speed variation with time

The average velocity method, the velocity correction method and the proposed method are used for simulation. The velocity correction method updates the velocity every 1 min. It should be noted that if the velocity correction cycle is too short, it will cause a large amount of calculation; and if it is too long, it will cause a large position tracking error. The simulation results of the three methods are shown in Figs.5—9.

Fig.5 shows the position tracking results of the three methods on the x-y plane, where "×" "◇" and "●" represent the position of the aircraft at the required arrival time under the three methods. It can be seen that in the presence of wind speed interference, the average velocity method cannot achieve accurate tracking, while the velocity correction method and the proposed method can achieve a good position tracking. But at individual waypoints (such as the sixth and ninth waypoint), the velocity correction method has a certain degree of tracking error.

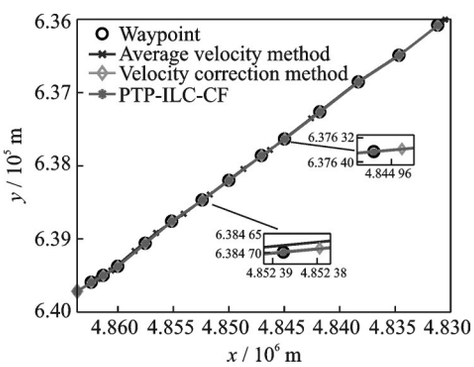

Fig.5 x-y position tracking result

Fig.6 shows the altitude tracking results performed by the three methods. The average velocity method fails to track the specified altitude at the required arrival time due to the influence of wind speed. The other two methods achieve good tracking results.

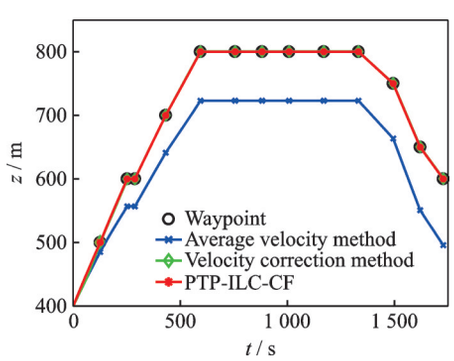

Fig.6 Altitude tracking results

Fig.7 shows the position tracking error of each waypoint of the three methods. Among them, the error produced by the average velocity method is the largest, as shown in Fig.7(a). Since the wind speed

changes with time, and the wind direction is different in the first half and the second half of the flight, this method produces a large tracking error. The velocity correction method periodically adjusts the average flight speed, so there is no accumulation of position tracking errors. However, at individual waypoints, the velocity correction is not timely, resulting in larger tracking errors, as shown in Fig.7(b). Compared with the other two methods, the proposed method has the smallest position tracking error that tends to 0. Since the proposed method learns and utilizes the tracking errors generated in the last operation, it shows higher tracking accuracy, as shown in Fig.7(c).

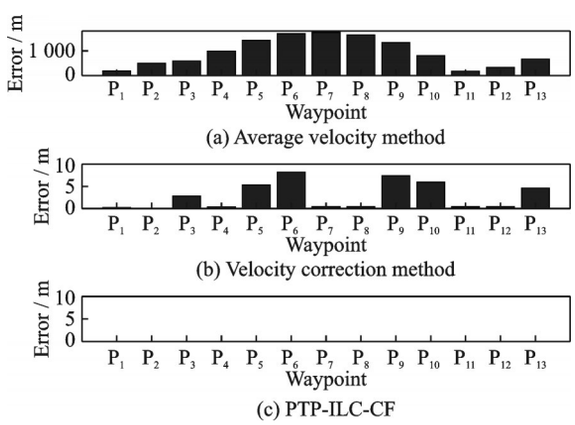

Fig.7 Position tracking error at each way point

Fig.8 shows the error convergence of the three algorithms along the iteration axis. It can be seen that the average velocity method produces relatively large errors, while the velocity correction method periodically modifies the average velocity, resulting in relatively small errors. However, the errors produced by these two methods are fixed and cannot be further reduced with the iterative operation. In contrast, the proposed method rapidly reduces the tracking error in the second run by learning from the first run data, and the tracking error is close to zero during the fourth iteration. This means that aircraft that travel between city pairs in a certain period of time can use operational data to automatically learn and continuously improve the accuracy of 4D trajectory tracking.

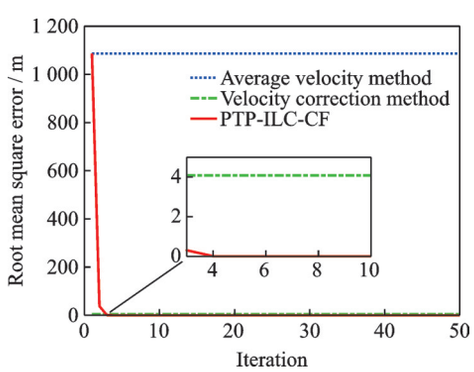

Fig.8 Variation of position tracking error with iteration

Fig.9 shows the control input signal. It can be seen that the aircraft is flying at a fixed velocity in each flight segment, and this control scheme is easy to implement in practice.

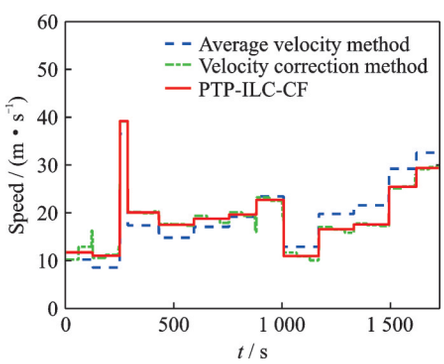

Fig.9 Control input

3.4.2 Case 2 with gust component

Next, the impact of gust interference is considered in the simulation. Let the gust appears at the 700th second of the 5th iteration and the last for 30 s, and the maximum wind speed is 5 m/s. The mathematical expression is as

$$w_{G,i}(t) = -0.5 \times \left(5 - 5 \times \cos\left(2\pi \frac{t-700}{30}\right)\right)$$
$$i = 5,\ 700 \leqslant t \leqslant 730 \tag{50}$$

Fig.10 shows the wind speed of gusts over time, where the negative sign indicates headwind.

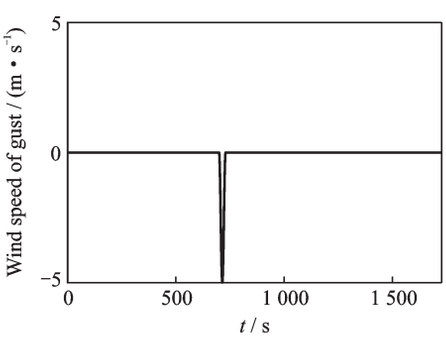

Fig.10 Gust at the 5th iteration

This kind of wind can be seen as a non-repetitive interference in the simulation.

After considering the influence of wind gust, the tracking errors produced by the three methods at each waypoint are shown in Fig. 11. It can be seen that the tracking error produced by the average velocity method is still large, while the velocity correction method has a small tracking error except at the P_6 waypoint. In contrast, the tracking error generated by the proposed method is the smallest, and due to the current-cycle feedback in the algorithm, the tracking error generated at the P_6 waypoint can be gradually eliminated at the subsequent waypoints.

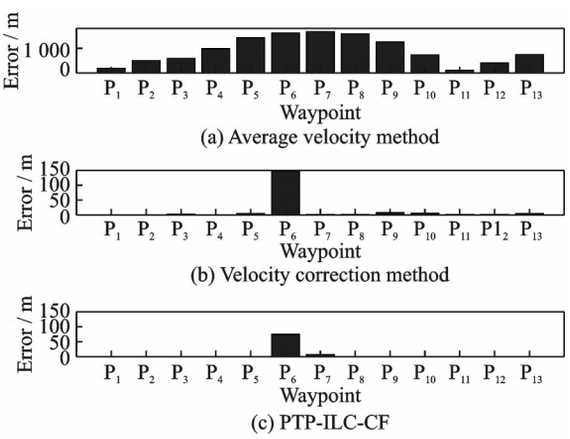

Fig.11 Tracking errors of three mehods at the waypoints under gust influence

From the perspective of the iteration axis, due to the interference of wind gust at the 5th iteration, the tracking error is produced, and it has an impact on the next iteration. But at the 7th iteration, the tracking error quickly approaches 0 again, as shown in Fig.12.

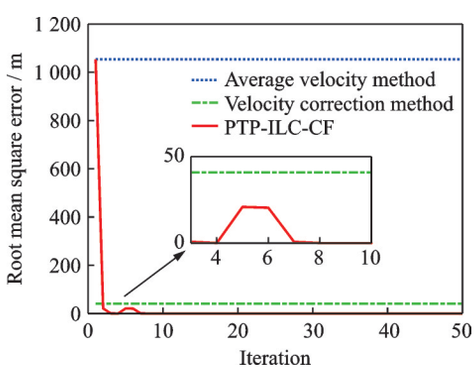

Fig.12 Tracking error of the iterative learning method along the iteration axis under gust influence

4 Conclusions

The ILC method is used to investigate the 4D trajectory tracking with a controlled time of arrival. Considering the flight characteristics between city pairs, the 4D trajectory problem is first formulated as a point-to-point tracking problem. Then for repetitive and non-repetitive disturbances that normally occur in the flight, a point-to-point iterative learning control algorithm with the current-cycle feedback is established. In the simulation, the average velocity method, velocity correction method and the proposed method are compared, and certain conclusions can be drawn.

(1) The average velocity method cannot achieve accurate 4D trajectory tracking control. Due to the influence of wind speed changes, there are some degrees of position tracking error at each waypoint. However, this method can provide a reference for the selection of initial values and the realization of other methods.

(2) The velocity correction method can effectively reduce the position tracking error at each waypoint. However, due to the periodicity of the velocity correction, the velocity correction may not be timely, which may lead to a larger position tracking error at individual waypoints.

(3) The proposed method can effectively learn from historical data, quickly reduce the position tracking errors, and continuously improve the control accuracy in subsequent iterations. These are the capabilities that the other two methods do not have.

Therefore, we can conclude that the trajectory-based operation between city-pairs could be implemented under the ILC framework. Future work is to eliminate the effect of real-time interference on the iterative axis, thus making the method more robust.

References

[1] ZHANG Xianfeng, ZHOU Qizhong, WANG Changqing. 4D route rapidly planning based on route-segment tree[J]. Journal of Beijing University of Aeronautics and Astronautics, 2013, 39(3): 310-314. (in Chinese)

[2] DING Qiang, TAO Weiming. Improved Tau-H strategy for four-dimensional cooperative route planning of

multi-UAVs[J]. Journal of Zhejiang University (Engineering Science), 2018,52(7): 1398-1405. (in Chinese)

[3] LIAO Wenjing, HAN Songchen, LI Wei, et al. 4D conflict-free trajectory planning for fixed-wing UAV[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020,37(2): 209-222.

[4] WANG Chao, GUO Jiuzhen, SHEN Zhipeng. Prediction of 4D trajectory based on basic flight models[J]. Journal of Southwest Jiaotong University, 2009, 44(2): 295-300. (in Chinese)

[5] ZHANG Junfeng, JIANG Haixing, WU Xiaoguang, et al. 4D trajectory prediction based on BADA and aircraft intent[J]. Journal of Southwest Jiaotong University, 2014,49(3): 553-558. (in Chinese)

[6] ZHANG Junfeng, GE Tengteng, CHEN Qiang, et al. 4D trajectory prediction and uncertainty analysis for departure aircraft[J]. Journal of Southwest Jiaotong University, 2016,51(4): 800-806. (in Chinese)

[7] DE SMEDT D, BERZ G. Study of the required time of arrival function of current FMS in an ATM context[C]//Proceedings of IEEE/AIAA 26th Digital Avionics Systems Conference. Dallas, USA: IEEE, 2007.

[8] SU Z, WANG H, SHAO X, et al. A robust backstepping based trajectory tracking controller for the tanker with strict posture constraints under unknown flow perturbations[J]. Aerospace Science and Technology, 2016,56: 34-45.

[9] ANTONELLI G, CATALDI E, ARRICHIELLO F, et al. Adaptive trajectory tracking for quadrotor mavs in presence of parameter uncertainties and external disturbances[J]. IEEE Transactions on Control Systems Technology, 2018,26(1): 248-254.

[10] FAN Y, SHAO X, LI Q, et al. Integrated 4D trajectory and attitude adaptive controller for civil aircraft[J]. Acta Aeronautica et Astronautica Sinica, 2019,40(2): 161-170.

[11] STASTNY T, DASH A, SIEGWART R. Nonlinear MPC for fixed-wing UAV trajectory tracking: Implementation and flight experiments[C]//Proceedings of AIAA Guidance, Navigation, and Control Conference. Grapevine, USA: AIAA, 2017.

[12] YANG Y, YAN Y. Neural network approximation-based nonsingular terminal sliding mode control for trajectory tracking of robotic airships[J]. Aerospace Science and Technology, 2016,54: 192-197.

[13] FENG Z. Shaping a new future for the civil aviation industry with intelligence[N]. CAAC News, 2019-05-20.

[14] TONG D S, NGUYEN D H, AHN H S. Iterative learning control for optimal multiple-point tracking[C]//Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference. Orlando, USA: IEEE, 2011.

[15] CHIN I, QIN S J, LEE K S, et al. A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection[J]. Automatica, 2004,40(11): 1913-1922.

[16] HUANG D, XU J X, VENKATARAMANAN V, et al. High-performance tracking of piezoelectric positioning stage using current-cycle iterative learning control with gain scheduling[J]. IEEE Transactions on Industrial Electronics, 2013,61(2): 1085-1098.

[17] OWENS D H, JONES R P. Iterative solution of constrained differential/algebraic systems[J]. International Journal of Control, 1976,27(6): 957-974.

[18] SMEDT D D, BRONSVOORT J, MCDONALD G. Model for longitudinal uncertainty during controlled time of arrival operations[C]//Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015). Lisbon, Portugal: [s.n.], 2015.

[19] FOSSEN T I. Mathematical models for control of aircraft and satellites[M]. 3rd Edition. Norway: Norwegian University of Science and Technology, 2013.

[20] VALENZUELA A, RIVAS D. Analysis of along-track variable wind effects on optimal aircraft trajectory generation[J]. Journal of Guidance, Control, and Dynamics, 2016,39(9): 2149-2156.

[21] BOUSSON K, MACHADO P F F. 4D trajectory generation and tracking for waypoint-based aerial navigation[J]. WSEAS Transactions on Systems and Control, 2013,8(3): 105-119.

Acknowledgement This work was supported by the Fundamental Research Funds for the Central Universities (No. 3122019131).

Author Dr. **JIANG Gaoyang** received his Ph.D. degree in traffic information engineering and control from Beijing Jiaotong University, Beijing, China, in 2021. He is currently a lecturer with Civil Aviation University of China. His research interests include data-driven control, iterative learning control, and model predictive control.

Author contributions Dr. **JIANG Gaoyang** designed the study, complied the models, conducted the analysis, interpreted the results and wrote the manuscript. Dr. **WANG Hongyong** contributed to the background and discussion of

the study. All authors commented on the manuscript and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor:ZHANG Bei)

基于当前迭代反馈点到点迭代学习控制的航空器4D航迹跟踪控制

姜高扬[1], 王红勇[2]

(1.中国民航大学空中交通管理学院,天津 300300,中国; 2.天津市空管运行规划与安全技术重点实验室,天津300300,中国)

摘要:为了实现航空器的四维航迹精确跟踪控制,提出一种带有当前迭代反馈的点到点迭代学习控制方法。首先,将四维航迹跟踪控制问题抽象成具有外部干扰的点到点跟踪控制问题。其次,基于连续投影法得到最优点到点迭代学习控制律。进一步,在控制律中加入当前迭代反馈,使跟踪误差在迭代域和时间域均能减小。最后,利用无人机运动学模型和四维航迹数据开展了数值仿真。仿真结果显示,即使是在有阵风干扰的情况下,所提方法依然能够快速减小航迹跟踪误差。与常用的平均速度法和速度修正法相比,所提方法充分利用了航空器过去和当前的运行数据,并且能够随着城市对之间航空器的重复运行而不断提升四维航迹跟踪精度。

关键词:四维航迹;航迹跟踪;迭代学习控制;基于航迹的运行;控制到达时间