

# 基于数字地图的亚音速飞行器最优地形跟随

董超<sup>1</sup>, 王彪<sup>1</sup>, 黄鹏程<sup>1</sup>, 唐超颖<sup>1</sup>, 魏东辉<sup>2</sup>

(1. 南京航空航天大学 自动化学院, 南京 211106;

2. 复杂系统控制与智能协同技术重点实验室, 北京 100074)

**摘要:** 针对真实高山地形设计了一套地形跟随系统; 首先, 根据数字地图结合航迹规划算法生成参考航迹; 接着设计了一个基于滚动时域优化的航迹控制器, 用于跟踪航迹、稳定速度并处理约束, 将预测时域内的航迹点作为参考输入, 通过最小化跟踪误差以及滚动优化实时求解控制量; 仿真结果表明, 与基于适应角导引的地形跟随系统相比, 文章的方法具有更好的性能, 能够满足亚音速飞行器地形跟随任务需求。

**关键词:** 地形跟随; 三次样条; 航迹跟踪; 滚动时域控制; 撞地概率

## Optimal Terrain Following for Subsonic Aircraft Based on Digital Map

DONG Chao<sup>1</sup>, WANG Biao<sup>1</sup>, HUANG Pengcheng<sup>1</sup>, TANG Chaoying<sup>1</sup>, WEI Donghui<sup>2</sup>

(1. School of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;

2. Lab. of Science and Technology on Complex System Control and Intelligent Agent Cooperation, Beijing 100074, China)

**Abstract:** This paper designs a terrain-following system based on the real mountain terrain. First, the reference path is generated using the digital map and the path planning algorithm. Then a controller based on receding horizon optimization is designed for simultaneous path following, speed stabilization and constraint handling. The waypoints in the predictive horizon are used as reference input, and the control input is solved in real time by minimizing the tracking error. The simulation results show that compared with the terrain following system based on the adaptive angle guidance, the method in this paper has better performance and can meet the requirements of the terrain following task for subsonic aircraft.

**Keywords:** terrain following; cubic spline; path tracking; receding horizon control; clobber probability

## 0 引言

军用飞行器常常需要贴地起伏飞行以利用地形躲避防空系统的探测, 这种飞行方式称为“地形跟随”。地形跟随技术大致可以分为两类<sup>[1]</sup>, 一类是角指令法, 另一类为参考航迹法。早期地形跟随系统多采用角指令法<sup>[2]</sup>, 利用飞行器与地形的相对位置关系生成航迹控制指令。参考航迹法基于数字地图生成飞行参考航迹, 再通过航迹跟踪实现地形跟随, 航迹跟踪是该地形跟随方案的关键部分, 控制系统需要具备较高的跟踪精度以保证地形跟随的安全性。在低空高速状态下, PID 难以提供良好的跟踪性能, 也很难处理实际系统的物理限制<sup>[3]</sup>。L1<sup>[4]</sup>、矢量场<sup>[5]</sup>等非线性导引律对变曲率航迹的跟踪效果较差。针对这些问题, 滚动时域控制 (RHC) 在航迹跟踪领域正受到越来越多的关注。RHC 又称“模型预测控制 (MPC)”, 作为一种基于模型的控制策略, 它可以显著提高控制性能, 其控制目标是使跟踪误差最小化<sup>[6]</sup>, 从而使航迹控制成为有约束系统的输出跟踪问题。文献 [7] 采用基于状态扩展的双反馈 MPC 设计固定翼无人机航迹跟踪控制器, 取得了比较精确的跟踪效果。文献 [8-9] 则使用非线性 MPC 作为控制策略, 为

了减小计算量, 对模型进行了简化以确保实时性。文献 [10] 所设计的线性 MPC 可将闭环系统的极点配置在单位圆内, 保证系统的稳定性。目前多数文献在设计跟踪控制器时均使用直线或圆弧航迹作为参考输入, 虽然这种做法简单便于应用, 但航迹在航迹点处是不光滑的, 跟踪误差会发生突变, 从而使舵角容易产生饱和。文献 [11] 针对变曲率曲线航迹设计了一个 MPC 控制器, 但简单的运动学模型并不适合高速飞行器, 为了提高飞行器在高速飞行时的可靠性, 控制策略必须考虑飞行动力学的影响, 在保证系统稳定运行的同时, 充分发挥飞行器的潜力。

本文针对亚音速飞行器地形跟随问题, 首先基于数字地图进行最优航迹规划, 再将滚动时域优化策略用于航迹跟踪控制器设计, 提高高速飞行器对复杂航迹的跟踪能力。在真实山区地形数据上进行仿真验证, 通过与基于适应角导引的地形跟随方法对比, 验证本文地形跟随方法的有效性。图 1 为本文地形跟随系统框图。

## 1 问题建立

### 1.1 飞行器模型与参考航迹

使用六自由度非线性动力学方程描述飞行器的运动,

收稿日期: 2021-03-25; 修回日期: 2021-05-12。

基金项目: 装备预研航天科工联合基金 (6141B07090114)。

作者简介: 董超 (1996-), 男, 江苏扬州人, 硕士研究生, 主要从事飞行控制与航迹规划方向的研究。

通讯作者: 王彪 (1975-), 男, 辽宁铁岭人, 副教授, 博士, 主要从事飞行控制与视觉导引, 鲁棒与最优控制方向的研究。

引用格式: 董超, 王彪, 黄鹏程, 等. 基于数字地图的亚音速飞行器最优地形跟随[J]. 计算机测量与控制, 2021, 29(11): 95-99.

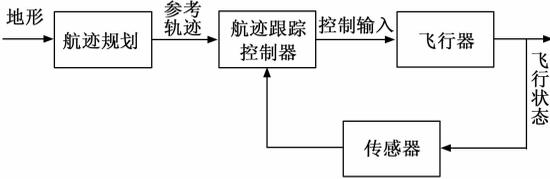


图 1 地形跟随系统结构图

忽略机体  $y$  轴所受的力以及滚转、偏航力矩，其纵向运动方程为<sup>[12-13]</sup>：

$$\begin{aligned} \dot{u} &= qw - g\sin\theta + \frac{1}{m}(X + T) \\ \dot{w} &= qu + g\cos\varphi\cos\theta + \frac{1}{m}Z \end{aligned} \quad (1)$$

$$\dot{q} = \frac{M}{I_y}, \ddot{\theta} = q \quad (2)$$

$$\begin{aligned} \dot{x} &= u\cos\theta + w\sin\theta \\ \dot{z} &= -u\sin\theta + w\cos\theta \end{aligned} \quad (3)$$

式中,  $u, w$  分别是飞行器速度沿机体  $x, z$  轴的分量;  $\theta$  为俯仰角,  $q$  为机体俯仰角速度;  $T, X, Z$  分别为推力、机体  $x$  轴和  $z$  轴的气动力, 推力沿机体纵轴,  $M$  为俯仰力矩。有关飞行器的参数详见<sup>[14]</sup>。

由图 1 可知, 需要根据数字地图生成参考航迹。本文参考文献 [15], 采用最优航迹规划算法生成参考航迹, 参考航迹由分段三次样条表示, 优化目标为航迹与地形的贴合程度。与 [15] 不同的是, 本文在目标函数中添加了法向加速度平方项, 从而减少飞行器大幅度机动的次数, 使飞行更加平稳。图 2 显示了这两种不同规划方法的区别。

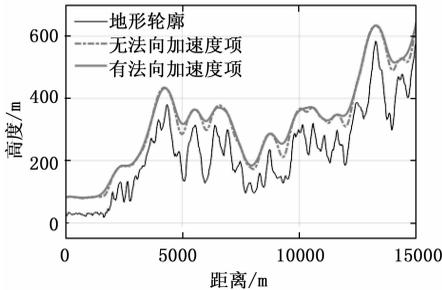


图 2 最优航迹规划结果

### 1.2 航迹跟踪控制问题描述

如图 3 所示, 航迹跟踪控制的目的是选择合适的控制输入, 使位置误差向量  $e_p$  渐进收敛至 0:

$$e_p = (d - p) \times T_d \quad (4)$$

式中,  $T_d$  为参考航迹点  $d$  处的单位切线矢量,  $p$  为飞行器当前位置, 同时速度方向与航迹的方向  $T_d$  一致, 使:

$$e_y = \gamma_d - \gamma \quad (5)$$

为 0, 其中  $\gamma_d = \text{atan2}(T_{dx}, T_{dy}) \in (-\frac{\pi}{2}, \frac{\pi}{2})$ ,  $\text{atan2}(\cdot)$  是 MATLAB 中的函数。这里的  $e_p$  被视为高度偏差, 将航迹倾斜角  $\gamma$  也作为被控变量可以改善航迹跟踪的性能, 它的作用是增加高度响应的阻尼, 减小高度跟踪上的超调。

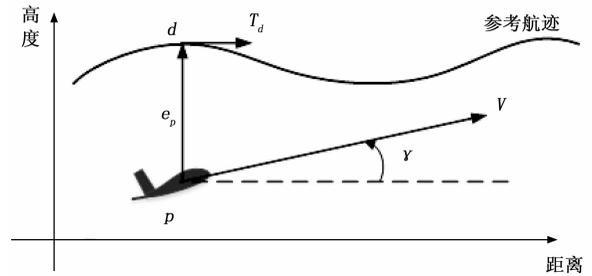


图 3 航迹跟踪示意图

式 (4) 和式 (5) 表明, 只要根据飞行器当前位置选择一个参考点, 并求出沿航迹切线方向的向量, 就可以使用此目标公式设计滚动时域控制的目标函数, 因此它适合于任何航迹, 而不局限于直线或圆弧等特定形式的航迹。

## 2 航迹跟踪控制器设计

### 2.1 控制目标

根据 1.2 节的讨论, 航迹控制的目标是最小化控制输入与跟踪误差:

$$J = \sum_{i=1}^{N_p} \|r(k+i) - \hat{y}(k+i)\|_Q^2 + \sum_{j=1}^{N_c} \|\Delta u(k+j-1)\|_R^2 \quad (6)$$

式中,  $k$  表示当前采样时刻,  $N_p, N_c$  分别为预测时域和控制时域,  $\hat{y}(k+i)$  是预测输出,  $\Delta u$  是控制增量,  $Q, R$  分别是跟踪误差与控制输入的正定加权矩阵,  $r(k+i)$  为参考值。该目标函数的第一项用于最小化航迹倾斜、高度与速度大小误差, 反映系统对参考航迹的跟随要求, 第二项反映对控制量平稳变化的要求, 避免控制量变化过大。整个表达式的目标就是在未来有限时域内使飞行器能够尽快且平稳地跟踪参考航迹, 同时稳定速度。

由图 3 可知, 为了达到控制目标, 正确选取预测时域内的参考航迹点  $d$  是提高控制器性能的关键, 本文根据当前位置  $p(k)$  和速度矢量  $V(k)$ , 使用欧拉法对未来的位置  $p(k+i)$  进行预测, 并在惯性系下定义未来有限时域内的参考航迹点, 如图 4 所示。这种方法的思想来源于驾驶员对车辆的操作: 驾驶员总是相对于车辆纵轴来预判未来的位置<sup>[16]</sup>。

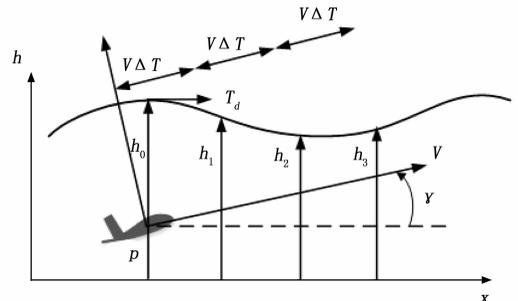


图 4 预测时域内参考航迹点的选取

## 2.2 滚动时域控制

RHC 航迹控制器是系统稳定、快速跟踪参考航迹的关键, 在进行控制器设计时, 首先将飞行器方程 (1) ~ (3) 在平衡点处进行线性化, 再按一定采样周期进行离散化处理, 得离散状态方程:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (7)$$

式中, 状态量、被控量和控制量依次为:

$$\begin{aligned} \mathbf{x} &= [V, \alpha, q, \theta, pow, h]^T \\ \mathbf{y} &= [V, \gamma, h]^T \\ \mathbf{u} &= [\delta_e, \delta_c]^T \end{aligned} \quad (8)$$

式中,  $\gamma = \theta - \alpha$  为航迹倾斜角,  $pow$  为发动机功率水平。V 为飞行器总速度,  $\delta_e, \delta_c$  分别表示升降舵偏转角和油门开度。为了对控制增量进行限制, 避免控制量突变, 需要将式 (7) 转换成以控制增量  $\Delta\mathbf{u}(k)$  为输入的形式, 定义状态增量与控制增量:

$$\begin{aligned} \Delta\mathbf{x}(k) &= \mathbf{x}(k) - \mathbf{x}(k-1) \\ \Delta\mathbf{u}(k) &= \mathbf{u}(k) - \mathbf{u}(k-1) \end{aligned} \quad (9)$$

增广状态向量:

$$\boldsymbol{\xi}(k) = \begin{bmatrix} \Delta\mathbf{x}(k) \\ \mathbf{y}(k) \end{bmatrix} \quad (10)$$

建立以  $\boldsymbol{\xi}(k)$  为状态量、 $\Delta\mathbf{u}(k)$  为控制输入的积分增广状态空间表达式<sup>[17]</sup>:

$$\begin{aligned} \boldsymbol{\xi}(k+1) &= \tilde{\mathbf{A}}\boldsymbol{\xi}(k) + \tilde{\mathbf{B}}\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \tilde{\mathbf{C}}\boldsymbol{\xi}(k) \end{aligned} \quad (11)$$

式中,  $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}\mathbf{A} & \mathbf{I} \end{bmatrix}$ ,  $\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{C}\mathbf{B} \end{bmatrix}$ ,  $\tilde{\mathbf{C}} = [0 \quad \mathbf{I}]$ ,  $\mathbf{0}, \mathbf{I}$  分别为维数相容的零矩阵和单位矩阵。式 (11) 用于预测系统未来输出, 在已知当前飞行器状态  $\boldsymbol{\xi}(k)$  的情况下, 输出预测值为:

$$\mathbf{Y}(k) = \mathbf{F}\boldsymbol{\xi}(k) + \boldsymbol{\Phi}\Delta\mathbf{U}(k) \quad (12)$$

$$\begin{aligned} \text{式中, } \mathbf{Y} &= \begin{bmatrix} \hat{\mathbf{y}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) \\ \vdots \\ \hat{\mathbf{y}}(k+N_p|k) \end{bmatrix}, \Delta\mathbf{U} = \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+N_c-1) \end{bmatrix}, \\ \mathbf{F} &= \begin{bmatrix} \tilde{\mathbf{C}}\tilde{\mathbf{A}} \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^2 \\ \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N_p} \end{bmatrix}, \boldsymbol{\Phi} = \begin{bmatrix} \tilde{\mathbf{C}}\tilde{\mathbf{B}} & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{C}}\tilde{\mathbf{B}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N_p-1}\tilde{\mathbf{B}} & \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N_p-2}\tilde{\mathbf{B}} & \cdots & \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{N_p-N}\tilde{\mathbf{B}} \end{bmatrix} \end{aligned}$$

飞行器的控制量在控制时域内定义为:

$$\begin{aligned} \mathbf{u}_{\min}(k+j) &\leq \mathbf{u}(k+j) \leq \mathbf{u}_{\max}(k+j) \\ \Delta\mathbf{u}_{\min}(k+j) &\leq \Delta\mathbf{u}(k+j) \leq \Delta\mathbf{u}_{\max}(k+j) \\ j &= 0, 1, \dots, N_c - 1 \end{aligned} \quad (13)$$

类似地, 状态约束在预测时域内定义为:

$$\begin{aligned} \mathbf{x}_{\min}(k+j) &\leq \mathbf{x}(k+j) \leq \mathbf{x}_{\max}(k+j) \\ j &= 0, 1, \dots, N_p - 1 \end{aligned} \quad (14)$$

定义了目标函数 (6) 和约束 (13) ~ (14) 后, 可使用二次规划算法求解该优化问题, 生成最优控制指令序列后使用其中第一个作为最优控制输入, 而下一时刻的控制量则根据下一时刻的状态反馈重新计算。

## 3 仿真结果与分析

### 3.1 撞地概率计算

使用撞地概率指标对地形跟随系统进行评价。假设  $t$  时刻飞行器的海拔高度为  $h(t)$ , 地形海拔高度为  $T(t)$ , 飞行器与地形之间的净高度差为  $h_e(t) = h(t) - T(t)$ , 预设的安全高度为  $H_0$ , 定义高度误差为  $\Delta h(t) = H_0 - h_e(t)$ , 将飞行器看作一个质点, 则撞地事件可以用如下公式表示<sup>[18]</sup>:

$$\Delta h(t_i) \geq H_0 \quad (15)$$

满足式 (15) 的所有时刻  $t_i$  均视为撞地。统计全过程中所有的  $t_i$ , 得到每次撞地事件所经历的时间  $t_{pi}$ , 最后根据总仿真时间  $t_{\text{total}}$ , 得到本文使用的撞地概率计算公式:

$$p_{\text{hit}} = \frac{\sum t_{pi}}{t_{\text{total}}} \quad (16)$$

除了计算相对实际地形的撞地概率, 本文还考虑了离地  $H_{\text{safe}}$  处“虚拟地形”的撞地概率, 即选取不同的离地高度  $H_{\text{safe}}$ , 当飞行器满足:

$$\Delta h(t_i) \geq H_0 - H_{\text{safe}} \quad (17)$$

时表示飞行器撞上了离地  $H_{\text{safe}}$  处的“地形”, 其撞地概率同样用公式 (16) 计算。

### 3.2 仿真结果

本文的地形数据从地理空间数据云平台上得到, 如图 5 所示, 该地形在起伏度分级上属于“高山起伏”<sup>[19]</sup>。飞行器初始速度  $V=200$  m/s, 初始高度  $h=80$  m, RHC 采样周期  $T_s=0.1$  s, 预测时域  $N_p=20$ , 控制时域  $N_c=20$ , RHC 控制器的权矩阵设置为:

$$\begin{aligned} \mathbf{Q} &= \text{diag}[0.01, 200, 0.01] \\ \mathbf{R} &= \text{diag}[2000, 15] \end{aligned} \quad (18)$$

升降舵以一阶惯性环节表示其动态, 增益为  $1/0.049$  5, 偏转幅度为  $\pm 25^\circ$ , 偏转速率限制为  $\pm 60^\circ/\text{s}$ 。

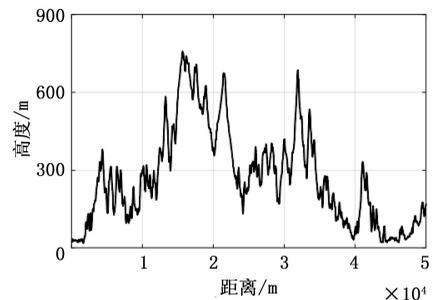


图 5 50 km 地形剖面

仿真 1: 图 6 显示了 50 km 的地形上进行地形跟随的结果, 可以看出, 飞行器的实际飞行航迹与参考航迹贴十分紧密, 显示了较高的跟踪精度, 即使在某些坡度较大的

地方也不会有撞地的危险，这是因为 RHC 可以根据前方航迹的变化提前动作以减小过峰谷时的过冲。图 7 为飞行器的各个状态随时间的变化曲线，包括航迹倾斜角  $\gamma$ 、迎角  $\alpha$  和速度  $V$ ，可以看到迎角处于约束范围  $[-10^\circ, 45^\circ]$  内，速度保持在 200 m/s 附近。图 8 为高度、航迹角误差曲线及执行器输入，高度误差最大 5.80 m，平均值 0.10 m，标准差 1.26 m，结果表明航迹控制方法有效完成了地形跟随任务。

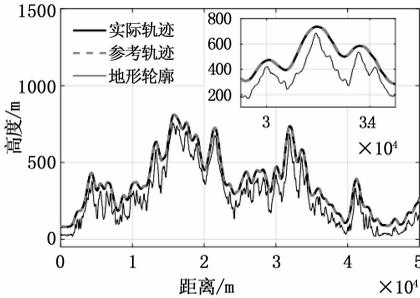


图 6 地形跟随结果

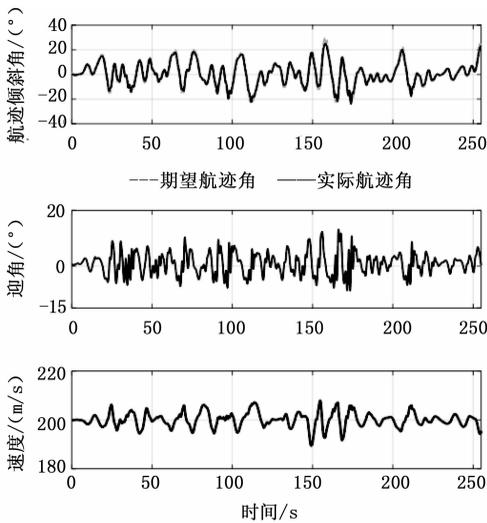


图 7 状态曲线

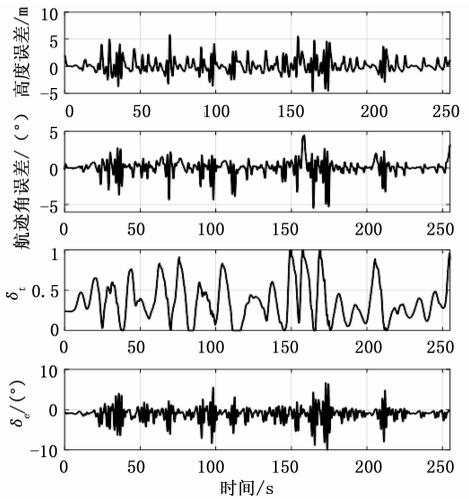


图 8 误差与控制量

到 10 m，但对于实际地形，撞地概率依旧为 0，从图 9 中也可以看出这一点。

表 1 水平与高度方向均存在导航误差的地形跟随结果

导航误差水平 /m	高度误差平均值/m	高度误差标准差/m	高度误差最大值/m	0 m 撞地概率 / (%)	10 m 撞地概率 / (%)	20 m 撞地概率 / (%)
0	0.10	1.26	5.80	0	0	0
+10	-9.82	1.94	-21.35	0	0	0
+20	-19.70	3.37	-31.72	0	0	0
+30	-29.71	5.42	-47.96	0	0.09	4.32

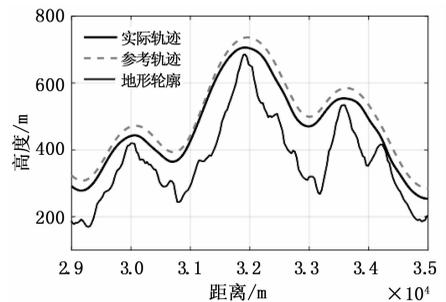


图 9 30 m 导航误差水平下的地形跟随结果 (29~35 km)

仿真 2：为了验证系统在存在导航误差时的跟踪效果，在水平与高度方向均存在导航误差的情况下进行了多次仿真。导航误差来自 GPS，呈高斯分布，方差为 1，误差水平（均值）有不同等级。一般 GPS 的导航误差最大为 10 m，本文增设了 20 m 与 30 m 的误差水平。RHC 参数设置保持不变，每个误差水平下进行 50 次仿真，记录其跟踪误差数据及撞地概率。除了计算相对实际地形的撞地概率，还计算离地 10 m、20 m 处“虚拟地形”的撞地概率，取所记录数据的平均值作为最终结果，将不同误差水平下的平均误差及撞地概率数据记录在表 1 中。图 9 为某一次仿真的结果。

从表 1 中可以看出，随着导航误差增大，跟踪误差也随之增大，但在误差水平达到 30 m 时才有可能出现撞地事故，这意味着飞行器实际飞行过程中的某些位置离地面不

表 2 为只有水平方向的导航误差时的统计数据，在 30 m 误差范围内均不存在撞地危险，显然只有水平导航误差时的系统性能要比存在高度导航误差的情况好得多。从表 1 与表 2 的数据可以看出，高度方向的导航误差对地形跟随撞地概率的影响远远大于水平方向，因此从地形跟随的安

表 2 只有水平方向存在导航误差的撞地概率统计

导航误差水平 /m	高度误差平均值/m	高度误差标准差/m	高度误差最大值/m	0 m 撞地概率 / (%)	10 m 撞地概率 / (%)	20 m 撞地概率 / (%)
0	0.10	1.26	5.80	0	0	0
+10	0.18	1.83	-8.00	0	0	0
+20	0.25	3.21	-11.72	0	0	0
+30	0.32	4.75	15.76	0	0	0

全角度来说, 应尽量减少高度测量误差, 除了使用 GPS 外, 最好添加气压计或无线高度表等额外传感器作为辅助。

仿真 3: 与基于适应角导引的地形跟随系统进行对比实验。适应角法是传统地形跟随中使用最广泛的方法, 在利用前视传感器获得前方地形信息后, 通过适应角可以得到航迹倾斜角的指令。角指令由如下公式给出<sup>[20]</sup>:

$$\theta_{FL} = K_{\theta} \left( \theta + \lambda + \frac{H_0}{L} - F_s \right) \quad (19)$$

式中,  $K_{\theta}$  为增益,  $F_s$  为抑制函数。本文的抑制函数分为三段, 分别对应飞行器在地形跟随中的 3 个阶段, 以飞行器与障碍物/地形的相对位置划分为: 末段冲刺、中程拉起和远程下滑。斜距在 3 个阶段有着不同的值, 分别为 3 500 m、3 900 m、4 627 m, 雷达对前方地形的探测角度为  $\pm 8^\circ$ , 参数  $K_{\theta}$  为 1.12。

图 10 显示了适应角法地形跟随的结果, 可以发现, 与本文基于数字地图的地形跟随系统相比, 该方法的贴地效果较差, 在过峰后的过冲较大, 增加了飞行器暴露的时间, 而在某些地方飞行器离地面非常近, 增加了撞地概率。在实际应用中, 基于适应角导引的地形跟随系统通常会牺牲贴地性能以保证安全性, 这是其缺陷所在。对比仿真表明, 本文所使用的地形跟随方法具有更优异的贴地性能以及更好的安全性。

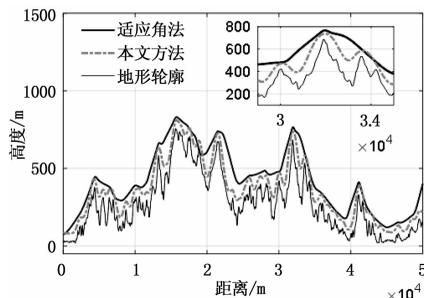


图 10 地形跟随结果比较

## 4 结束语

本文设计了亚音速飞行器地形跟随控制系统。使用最优航迹规划算法生成参考航迹, 该航迹能够在贴近地形的同时减少不必要的飞行机动。针对精确航迹跟踪问题, 基于 RHC 设计了一个航迹跟踪控制器, 即使存在导航误差时该控制器也具有好的跟踪性能。讨论了目标函数的设计, 以及预测时域内参考航迹点的选取, 提高飞行器对复杂曲线航迹的跟踪性能。最后, 与使用适应角法的地形跟随系统进行对比仿真, 结果表明, 本文所设计的系统具有良好的安全性和优异的贴地性能, 非常适合在高山地形上执行地形跟随任务。

### 参考文献:

[1] 魏东辉, 程进, 陈万春. 飞行器超低空飞行制导控制技术发展研究 [J]. 战术导弹技术, 2019 (1): 72-78.  
[2] 谢燕武. 地形跟随技术研究 [D]. 西安: 西北工业大

学, 2007.

[3] 王松涛. 固定翼无人机飞行控制系统设计 [D]. 北京: 北京理工大学, 2015.  
[4] SONG J, GAO K, YANG E. An optimal nonlinear guidance logic for the trajectory tracking of supercavitating vehicles [C] //2017 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2017: 1221-1226.  
[5] SUJIT P B, SARIPALLI S, SOUSA J B. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles [J]. IEEE Control Systems Magazine, 2014, 34 (1): 42-59.  
[6] RAGHAVAN R, THOMAS S. Practically implementable model predictive controller for a twin rotor multi-input multi-output system [J]. Journal of Control, Automation and Electrical Systems, 2017, 28 (3): 358-370.  
[7] 王晓海, 孟秀云, 李传旭. 基于 MPC 的无人机航迹跟踪控制器设计 [J]. 系统工程与电子技术, 2021, 43 (1): 191-198.  
[8] STASTNY T J, DASH A, SIEGWART R. Nonlinear MPC for fixed-wing UAV trajectory tracking: Implementation and flight experiments [C] //AIAA guidance, navigation, and control conference. 2017: 1512-1525.  
[9] STASTNY T, SIEGWART R. Nonlinear model predictive guidance for fixed-wing uavs using identified control augmented dynamics [C] //2018 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2018: 432-442.  
[10] ESKANDARPOUR A, SHARF I. A constrained error-based MPC for path following of quadrotor with stability analysis [J]. Nonlinear Dynamics, 2020, 99 (2): 899-918.  
[11] YANG K, KANG Y, SUKKARIEH S. Adaptive nonlinear model predictive path-following control for a fixed-wing unmanned aerial vehicle [J]. International Journal of Control, Automation and Systems, 2013, 11 (1): 65-74.  
[12] 吴森堂. 飞行控制系统 [M]. 北京: 北京航空航天大学出版社, 2013.  
[13] HUO Y. Model of F-16 fighter aircraft [R]. University of Southern California, Technical Report, 2007.  
[14] SONNEVELDT L. Nonlinear F-16 model description [R]. Delft University of Technology, Netherlands, 2006.  
[15] 王彪, 董超, 黄毅, 等. 地形跟随中的航迹规划与跟踪 [J]. 电光与控制, 2021 (3): 31-35.  
[16] 龚建伟, 姜岩, 徐威. 无人驾驶车辆模型预测控制 [M]. 北京: 北京理工大学出版社, 2014.  
[17] WANG L. Model predictive control system design and implementation using MATLAB [M]. Springer Science & Business Media, 2009.  
[18] 谢邦荣, 尹健, 宋劲松. 空射巡航导弹撞地概率仿真研究 [J]. 系统仿真学报, 2004 (10): 2252-2256.  
[19] 乔森. 基于高差与标准差法的中国地形起伏度研究 [J]. 绿色科技, 2020 (6): 31-33.  
[20] 刘萍. 基于某型导弹的航迹规划与地形匹配技术研究 [D]. 南京: 南京航空航天大学, 2007.