



论文

基于反射鸽群优化算法的无人机自主空中加油自抗扰控制方法

巩嘉玮, 范彦铭*

中国航空工业集团公司, 沈阳飞机设计研究所, 沈阳 110035

* E-mail: michaelfan@yeah.net

收稿日期: 2024-09-06; 接受日期: 2024-12-03; 网络版发表日期: 2025-04-11

国家自然科学基金(批准号: 62473025)资助项目

摘要 本文针对空中加油过程中受油机的姿态控制问题, 提出了基于反射鸽群优化(reflected pigeon-inspired optimization, RPIO)算法的无人机(unmanned aerial vehicle, UAV)姿态控制律设计方法, 首先设计一种基于自抗扰控制(active disturbance rejection control, ADRC)的无人机姿态控制律, 然后再用反射鸽群优化算法对该控制律参数进行优化, 形成最终控制律。同时, 本文对该反射鸽群优化算法与经典鸽群优化算法以及粒子群算法的优化结果进行了对比分析。仿真结果表明, 本文所提出的反射鸽群优化算法在设计效果上表现更佳, 可显著提升自主空中加油过程中无人机姿态控制的精度和稳定性。

关键词 空中加油, 反射鸽群优化, 自抗扰控制

1 引言

随着无人机(unmanned aerial vehicle, UAV)技术的发展, 越来越多的无人机逐步应用于民用及军事领域。面对复杂和长时间的飞行任务, 无人机自主空中加油技术的应用可以有效增加飞机的行程和载荷。无人机自主空中加油是指无人机在没有驾驶员控制的情况下, 通过自主的导航、感知、决策和控制与空中加油机进行精准对接, 完成燃油补给的过程。在空中加油对接阶段, 无人机需要通过一系列传感器来精确定位加油管或锥套的位置。此时无人机的控制系统将结合传感器数据, 实时计算相对位置、速度和姿态, 并精

确调整自身位置和姿态, 确保与加油机的精确对接。这一过程需要极高的精度和响应速度, 无人机需要在燃油补给期间保持高度稳定的姿态控制, 以确保燃油输送的顺利完成。与此同时, 受油机还要抵抗加油机尾流以及各种复杂自然气流的干扰, 这些气流可能会对无人机的稳定性、姿态控制 and 对接精度产生显著影响, 因此需要高度精确的控制系统进行补偿和调节^[1]。

为解决空中加油抗干扰精确控制问题, 国内外许多学者应用了自适应控制、主动抗干扰控制、最优控制等方法^[2,3]。弗吉利亚理工大学的Wang等^[4]采用L1自适应控制方法来解决空中加油过程中气流扰动下的受油机精确控制的问题, 既抑制了尾流的扰动又减小了

引用格式: 巩嘉玮, 范彦铭. 基于反射鸽群优化算法的无人机自主空中加油自抗扰控制方法. 中国科学: 技术科学, 2025, 55: 893–905

Gong J W, Fan Y M. Active disturbance rejection control method for UAV autonomous aerial refueling based on reflected pigeon-inspired optimization (in Chinese). Sci Sin Tech, 2025, 55: 893–905, doi: [10.1360/SST-2024-0228](https://doi.org/10.1360/SST-2024-0228)

稳态误差. 北京航空航天大学Su等^[5,6]根据受油机的仿射非线性模型, 将控制框架分为五个回路, 并分别应用自抗扰、反步高阶滑膜等控制方法, 实现空中加油过程中的快速精确控制.

自抗扰控制(active disturbance rejection control, ADRC)最早由Han^[7]提出, 相较于传统的PID控制, ADRC具有较好的抗干扰能力以及较高的控制精度, 并且不依赖于被控对象的具体数学模型. Duan等^[8]利用自抗扰控制的架构设计了一个受油机参数自适应等效滑模控制律, 可以在抑制复杂扰动的同时实现椎管-锥套的相对位置精确控制. 自抗扰控制具有较好的控制性能和鲁棒性, 但是由于其参数较多, 参数整定较为困难, 难以达到最好的控制效果.

鸽群优化(pigeon-inspired optimization, PIO)算法最早由段海滨教授在2014年首次提出^[9]. 鸽群优化算法模拟了鸽子的导航能力以及鸽群归巢机制, 能较好地解决算法寻优过程中探索与开发之间的平衡. 自鸽群优化算法首次提出至今, 已针对各种问题衍生出许多变型, 并且得到广泛应用. 文献^[10]中Duan等提出了一种捕食-逃逸鸽群优化(predator-prey pigeon-inspired optimization, PPPIO)算法, 通过将鸽群分成逃逸者和捕食者来增强全局搜索能力. Hai等^[11,12]提出了一种进化博弈鸽群优化(evolutionary game theory-based pigeon-inspired optimization, EGPIO)算法, 鸽子在迭代过程中通过进化博弈不断调整策略, 提高了鸽群的适应性.

针对ADRC控制律参数优化等多维优化问题, 经典鸽群优化算法容易陷入局部最优, 难以达到较好的优化效果. 因此, 本文提出了一种反射鸽群优化(reflected pigeon-inspired optimization, RPIO)算法, 用于ADRC控制律参数优化. RPIO算法相较于经典鸽群优化算法加入了镜面反射学习机制, 通过该机制生成一一对应的镜像个体, 增加了鸽子种群的多样性和适应性, 能有效地防止鸽子陷入局部最优解.

本文主要研究内容是: 首先利用自抗扰技术设计了一个用于自主空中加油的无人机姿态控制律, 然后为了解决ADRC参数整定困难的问题, 提出了一种基于反射鸽群优化算法的控制律参数优化设计方法, 完成了设计与分析, 并与经典鸽群优化算法以及粒子群(particle swarm optimization, PSO)算法的结果进行了比较. 本文的主要创新点可以总结如下.

1) 利用自抗扰技术设计了一个无人机姿态控制律, 以实现复杂气流干扰下的无人机自主空中加油.

2) 提出了一种反射鸽群优化算法, 并利用基于反射鸽群优化算法的控制律参数优化设计方法, 完成了无人机自主空中加油自抗扰控制律的设计与分析.

2 自抗扰控制律及无人机建模

2.1 自抗扰控制律总体结构

自抗扰控制律主要由三部分组成, 包括微分跟踪器(tracking differentiator, TD)、扩张状态观测器(extended state observer, ESO)以及非线性状态误差反馈律(nonlinear state error feedback, NLSEF), 其具体结构如图1所示^[13]. 其中, v 为输入的目标期望值, v_1 为输入值 v 的跟踪值, v_2 为 v_1 的导数; e_1 和 e_2 为经过ESO反馈之后反馈值与期望值的跟踪误差; u 为控制量, u_0 表示虚拟控制量, b_0 为反馈增益, y 表示系统的输出. ADRC能有效地利用ESO来精确地估计和减少系统模型中的不确定性以及外部扰动, 同时还利用了NLSEF来提供反馈并有效地抑制系统干扰.

2.2 跟踪微分器

跟踪微分器可以使输入的信号更加平滑, 防止信号突变, 从而可以提高实际系统中信号的实时跟踪能力^[14]. TD主要有一个输入 v 和两个输出 v_1 , v_2 , 其中 v_1 是对于输入信号 v 的跟踪, v_2 是 v_1 的导数. 在跟踪过程中 v_1 会逐渐爬升到 v 的值, 这个过程中不会产生静差和超调. TD的主要数学表达式如下:

$$\begin{cases} e_v = v_1 - v_0, \\ v_1 = v_1 + Tv_2, \\ v_2 = v_2 + Tf_{han}, \\ f_{han} = fhan(v_1, v_2, r_0, h_0), \end{cases} \quad (1)$$

式中, v_1 用于跟踪TD的输入 v_0 , v_2 为 v_1 的导数, e_v 表示跟

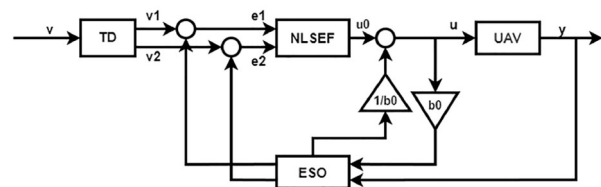


图1 ADRC控制律的结构图
Figure 1 Structure of ADRC controller.

踪误差; T 为时间步长; r_0 表示跟踪速度因子; h_0 表示滤波因子; $\text{fhan}(\cdot)$ 为最速控制函数, 其具体表达式如式(2)所示, 其中 $\text{sign}(\cdot)$ 为符号函数.

$$\begin{cases} \text{fhan}(v_1, v_2, r_0, h_0) \\ = \begin{cases} v_2 + \frac{1}{2}\text{sign}(y) \cdot (a_0 - d), & |y| > d_0, \\ v_2 + \frac{y}{h_0}, & |y| \leq d_0, \end{cases} \\ d = r_0 h_0, \\ d_0 = h_0 d, \\ y = v_1 + h_0 v_2, \\ a_0 = \sqrt{d^2 + 8r_0|y|}. \end{cases} \quad (2)$$

跟踪速度因子 r_0 越大, TD的跟踪效果越好, 但与此同时较小的 r_0 可以减小超调量, 因此需要选择一个合适的速度因子来达到最好的跟踪效果. 滤波因子 h_0 可以过滤掉输入信号中的噪声, 使跟踪信号更加平滑.

2.3 扩张状态观测器

扩张状态观测器可以实时地跟踪系统的状态变量, 并利用反馈的方式对系统给予补偿, 通过补偿消除扰动的影响, 从而达到抗干扰的效果, 提高系统的控制性能^[15]. 从图1中可以看出ESO的输入值为 y 和 b_0u , 输出值有三个, 分别为 z_1 , z_2 和 z_3 . z_1 为目标输出的估计值, z_2 为 z_1 的一阶导数, z_3 是系统的一个扩张状态, 用于观测系统的总扰动. ESO的具体数学模型如式(3)所示:

$$\begin{cases} e = z_1 - y, \\ z_1 = z_1 + T(z_2 - \beta_{01}e), \\ z_2 = z_2 + T(z_3 - \beta_{02}f_{e0} + b_0u), \\ z_3 = z_3 + T(-\beta_{03}f_{e1}), \\ f_{e0} = \text{fal}(e, \alpha_{01}, \delta), \\ f_{e1} = \text{fal}(e, \alpha_{02}, \delta), \end{cases} \quad (3)$$

式中, e 为目标输出的估计值 z_1 与实际输出 y 之间的误差; T 为时间步长; β_{01} , β_{02} 和 β_{03} 是ESO的增益, 经过多次调试发现 β_{01} 取30时控制效果最好; u 为控制量; b_0 为反馈增益; $\text{fal}(\cdot)$ 是一个非线性函数, 其非线性程度由 α_{01} 和 α_{02} 决定, 通常情况下 $\alpha_{01}=0.5$, $\alpha_{02}=0.25$ ^[13]. f_{e0} 和 f_{e1} 是根据 $\text{fal}(\cdot)$ 函数得到的, 他们可以根据误差 e 的大小来调整ESO的增益. δ 表示 $\text{fal}(\cdot)$ 函数线性区间的宽度, 通常取0.02^[13]. $\text{fal}(\cdot)$ 的具体表达式如下:

$$\text{fal}(e, \alpha, \delta) = \begin{cases} |e|^\alpha \text{sign}(e), & |e| > \delta, \\ \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta. \end{cases} \quad (4)$$

2.4 非线性状态误差反馈律

非线性状态误差反馈律主要根据TD得出的给定信号及其微分, 与ESO观测到的系统输出及其导数进行对比得到误差, 进而进行控制和扰动补偿, 其表达式如下:

$$\begin{cases} e_{01} = v_1 - z_1, \\ e_{02} = v_2 - z_2, \\ u_0 = \beta_1 \cdot \text{fal}(e_{01}, \alpha_{11}, \delta_1) + \beta_2 \cdot \text{fal}(e_{02}, \alpha_{12}, \delta_1), \end{cases} \quad (5)$$

式中, e_{01} 和 e_{02} 为反馈得到的误差; β_1 和 β_2 分别为NLSEF的比例增益和微分增益; δ_1 与ESO中的 δ 类似, 此处取0.5; u_0 为虚拟控制量, 其与实际控制量 u 之间的关系可以表示为

$$u = u_0 - \frac{z_3}{b_0}. \quad (6)$$

通过NLSEF可以得到虚拟控制量 u_0 , 之后根据式(6)得到的控制量对ESO进行补偿, 从而使控制律达到抗扰的效果.

综上所述, 本文中所设计的ADRC控制律共有9个参数需要优化调整, 分别为TD中的 r_0 和 h_0 , ESO中的 β_{02} 和 β_{03} , NLSEF中的 β_1 , β_2 , α_{11} 和 α_{12} , 以及反馈增益 b_0 .

2.5 无人机运动模型

无人机姿态控制主要包括对俯仰、滚转和偏航三个姿态角的控制, 无人机可以通过控制舵机偏转实现对其自身姿态的控制. 在无人机空中受油过程中, 完整动力学模型涉及复杂的气动耦合效应、姿态和位置的协调控制、燃油质量和力矩的动态变化以及环境扰动的影响^[16]. 加油机与受油机之间的气动耦合效应会导致非线性和时变性的增加, 对受油机的姿态控制造成影响, 使动力学模型更加复杂. 本文中简化了飞机运动模型, 利用无人机的姿态角运动公式建立了一个六自由度的非线性无人机模型, 具体公式如下^[2]:

$$\begin{cases} \dot{\phi} = p + (q \sin \phi + r \cos \phi) \tan \theta, \\ \dot{\theta} = q \cos \phi - r \sin \phi, \\ \dot{\psi} = (q \sin \phi + r \cos \phi) \sec \theta, \end{cases} \quad (7)$$

$$\begin{cases} \dot{p} = \frac{1}{I_{xz} - I_{xz}^2} [I_{xz}N + I_zL + qr(I_zI_y - I_z^2 - I_{xz}^2) \\ \quad + pq(I_{xz}I_x - I_{xz}I_y - I_{xz}I_z)], \\ \dot{q} = \frac{1}{I_y} [M + r^2I_{xz} - p^2I_{xz} + pr(I_z - I_x)], \\ \dot{r} = \frac{1}{I_xI_z - I_{xz}^2} [pq(I_x^2 - I_yI_x + I_{xz}^2) \\ \quad - qr(I_{xz}I_x - I_{xz}I_z - I_{xz}I_y) + NI_x + LI_{xz}], \end{cases} \quad (8)$$

式中, ϕ 为无人机的滚转角, θ 为俯仰角, ψ 为偏航角; p 为滚转角速度, q 为俯仰角速度, r 为偏航角速度; I_x , I_y , I_z , I_{xz} 为无人机的转动惯量; L , M , N 分别为滚转力矩、俯仰力矩和偏航力矩, 具体公式可以表示如下:

$$\begin{cases} L = \frac{1}{2}\rho V^2 S b \left(C_{l,0} + C_{lp}\beta + \frac{b}{2V}C_{lp} \right. \\ \quad \left. + \frac{b}{2V}C_{lr}r + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r \right), \\ M = \frac{1}{2}\rho V^2 S b_A \left(C_{m,0} + C_{m\delta_e}\delta_e + C_{m\alpha}\alpha + \frac{b_A}{2V}C_{mq}q \right), \\ N = \frac{1}{2}\rho V^2 S b \left(C_{n,0} + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r \right. \\ \quad \left. + C_{n\beta}\beta + \frac{b}{2V}C_{np}p + \frac{b}{2V}C_{nr}r \right), \end{cases} \quad (9)$$

式中, ρ 为空气密度; V 为飞行速度; S 为机翼面积; b 为翼展; b_A 为平均气动弦长; δ_a , δ_e 和 δ_r 为舵机偏转角; $C_{l,0}$, C_{lp} , C_{lr} , $C_{l\delta_a}$, $C_{l\delta_r}$, $C_{m,0}$, $C_{m\delta_e}$, $C_{m\alpha}$, C_{mq} , $C_{n,0}$, $C_{n\delta_a}$, $C_{n\delta_r}$, $C_{n\beta}$, C_{np} , C_{nr} 为无人机的气动参数。

在空中加油过程中, 受油机在受到加油机尾流等复杂气流扰动的同时, 必须实时跟踪加油机上快速移动的锥套。为了确保无人机在空中加油过程中姿态角控制的快速性及准确性, 对自抗扰控制律的设计提出了较高的要求。

3 反射鸽群优化算法

3.1 鸽群优化算法简述

鸽群优化算法是一种通过模拟鸽子归巢行为和导航机制来进行全局优化的智能算法。根据鸽子归巢过程中不同阶段的飞行特点, 鸽群优化算法分为地图和指南针算子、地标算子两个阶段^[17]。

1) 地图和指南针算子。在归巢的初始阶段, 鸽子可以通过地球磁场以及太阳高度角进行导航和调整方

向。在鸽群优化算法中, 每只鸽子的位置向量代表寻优问题中解空间的一个解。若鸽群数量为 N_p , 搜索空间的维数是 D , 第 i 只鸽子在第 t 次迭代中的速度和位置分别表示为 V_i^t 和 X_i^t , 则每只鸽子的速度和位置更新规则可以表示如下^[17]:

$$V_i^{t+1} = e^{-R(t+1)} \cdot V_i^t + rand \cdot (X_g^t - X_i^t), \quad (10)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \quad (11)$$

式中, R 为地图和指南针算子; $rand$ 是一个 0 到 1 之间的随机数; X_g^t 代表第 t 次迭代中的全局最优解。当迭代次数达到 N_{c1max} 时, 这一阶段停止并进入地标算子阶段。

2) 地标算子。当鸽子逐渐接近目标位置时, 会利用周围的地标进行导航。算法的第二阶段由地标算子主导, 在这一阶段鸽群会在每次迭代中抛弃适应度值较差的一半鸽子。若用 X_c^t 代表每次迭代中鸽群的中心位置, 则鸽子的位置更新规则可以表示如下:

$$N_p^t = [N_p^t / 2], \quad (12)$$

$$X_c^t = \frac{\sum_{i=1}^{N_p^t} X_i^t \cdot f(X_i^t)}{\sum_{i=1}^{N_p^t} f(X_i^t)}, \quad (13)$$

$$X_i^{t+1} = X_i^t + rand \cdot (X_c^t - X_i^t), \quad (14)$$

式中, $f(X_i^t)$ 代表鸽子的位置代价函数值, 具体公式如式 (15) 所示, 其中 $fitness(X_i^t)$ 为各自的适应度函数值, ε 是一个任意非零常数, 本文中取 0.5。当地标算子的迭代次数达到 N_{c2max} 时, 算法结束。

$$f(X_i^t) = \frac{1}{fitness(X_i^t) + \varepsilon}. \quad (15)$$

3.2 反射鸽群优化算法

经典鸽群优化算法虽然在解决一般优化问题时具有较快的收敛速度, 但当优化问题的维度较高时, 容易过早地陷入局部最优解。因此为了优化 ADRC 控制律的参数, 本文提出了一种反射鸽群优化算法。

镜面反射学习策略的灵感源自物理学领域中的镜面反射现象, 这种策略已被证明是一种增强元启发式算法优化性能的有效手段^[18]。本文提出的反射鸽群优化算法, 其核心便是运用镜面反射学习策略对传统的经典鸽群优化算法进行改进。反射鸽群优化算法的具

体原理如图2所示.

在每一轮迭代过程中, 每一只鸽子都会通过反射操作获得其在对应位置的反射鸽子. 图2中, $X_{i \text{ reflected}}^t$ 表示第 t 次迭代中第 i 只鸽子的反射鸽子, 而 X_g^t 代表第 t 次迭代的全局最优解. 反射的目的是在搜索空间中找到新的解, 这个新解可能比反射前的位置更好. 通过将反射前后的两个解进行对比, 保留处于更优位置的解. 反射学习策略的具体数学表达式如下^[19]:

$$X_{i \text{ reflected}}^t = (1 + \alpha)X_g^t - \alpha X_i^t, \quad (16)$$

式中, α 代表反射度, 是决定反射后新位置的关键, 取值范围为 $[1, 5]$, 本文中取1. 通过反射策略得到的新解可以有效增加搜索空间中解的个数, 将反射得到的新解 $X_{i \text{ reflected}}^t$ 与原始解 X_i^t 进行比较, 保留适应度值更优的解.

经典鸽群优化算法在解决多维优化问题时容易陷入局部最优解, 因此本文提出的RPIO算法引入了一种反射策略来解决这一问题. 这种策略的核心思想是在搜索过程中利用反射来适当地调整鸽群位置, 以此来增加鸽子种群的多样性. 这种反射策略能够在鸽群陷入局部最优解时, 帮助它们跳出当前的区域, 并继续在更大的搜索空间中进行探索. 因此, 这种改进策略能够降低算法陷入局部最优的概率, 提高算法的全局搜索能力, 使其在面对更加复杂的优化问题时, 能够更高效地得到最优解. 反射鸽群优化算法的伪代码如算法1所示.

3.3 反射鸽群优化算法收敛性分析

对于本文提出的RPIO算法, 接下来利用鞅理论对其收敛性进行分析和证明. 证明算法的收敛性, 需要证明算法在经过连续无穷次迭代后, 收敛到全局最优解

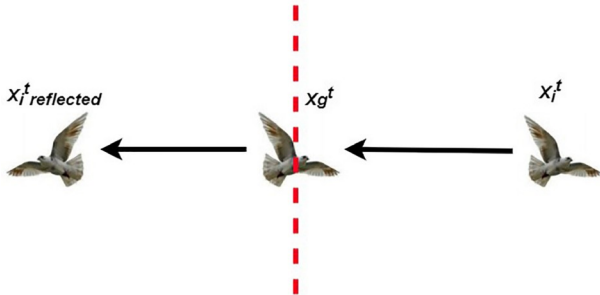


图2 (网络版彩图)反射鸽群优化算法原理示意图

Figure 2 (Color online) The schematic diagram of RPIO.

算法 1: RPIO算法

```

1. 初始化
   设置鸽群的  $N_p, D, N_{c1max}, N_{c2max}$ . 初始化每只鸽子的初始位置  $X_i^1$  和速度  $V_i^1$ , 计算鸽子的适应度值, 得到全局最优解  $X_g^1$  和个体最优解  $P_{best}^1$ .

2. 地图和指南针算子
   For  $t=1$  to  $N_{c1max}$  do
     For  $i=1$  to  $N_p$  do
       根据式(10)和式(11)更新鸽子的速度  $V_i^t$  和位置  $X_i^t$ ;
       根据式(16)得到反射解  $X_{i \text{ reflected}}^t$ ;
       比较  $X_{i \text{ reflected}}^t$  与  $X_i^t$  的适应度函数值, 保留适应度值较低的个体
     End for
     用适应度函数评价  $X_i^t$ , 并更新  $X_g^t$  和  $P_{best}^t$ 
   End for

3. 地标算子
   For  $t=1$  to  $N_{c2max}$  do
     根据适应度函数值对鸽子排序, 保留适应度值较低的个体  $N_p' = \lfloor N_p/2 \rfloor$ ;
     根据式(13)计算  $X_c^t$ ;
     For  $i=1$  to  $N_p'$  do
       根据式(14)更新  $X_i^t$ , 并根据式(16)得到反射解  $X_{i \text{ reflected}}^t$ ;
       比较  $X_{i \text{ reflected}}^t$  与  $X_i^t$  的适应度函数值, 保留适应度值较低的个体;
     End for
     用适应度函数评价  $X_i^t, X_g^t, P_{best}^t$ 
   End for

4. 输出
   输出全局最优解以及适应度函数值

```

的概率为1. 本文首先对反射鸽群期望位置的收敛性进行分析, 证明反射鸽群经过连续无穷次迭代后能够收敛到同一位置. 但这一位置并不一定是全局最优解, 因此还需要证明RPIO算法的全局收敛性.

RPIO算法的基本过程包括地图和指南针算子以及地标算子两个阶段, 第一阶段的有限迭代不影响收敛性, 因此可以直接从第二阶段开始分析其收敛性^[17].

1) 鸽群期望位置的收敛性.

用 X_{ij}^t 和 X_{cj}^t 代表 X_i^t 和 X_c^t 的第 j 个分量, $j=1, 2, \dots, D$. 当不考虑反射策略时, 根据地标算子阶段的位置更新公式(14), 可得期望序列 $\{E(X_{ij}^t)\}_{t=1}^{+\infty}$ 的方程如下:

$$E(X_{ij}^t) = \frac{1}{2}E(X_{ij}^{t-1}) + \frac{1}{2}E(X_{cj}^t). \quad (17)$$

式(17)可视为线性非齐次一阶差分方程, 其通解可以表示为

$$E(X_{ij}^t) = \frac{\frac{1}{2}E(X_{cj}^t)}{1 - \frac{1}{2}} + A\left(\frac{1}{2}\right)^t = E(X_{cj}^t) + A\left(\frac{1}{2}\right)^t, \quad (18)$$

其中 A 是一个常数. 当 $t \rightarrow +\infty$ 时, 式(18)可表示为

$$\lim_{t \rightarrow +\infty} E(X_{ij}^t) = \lim_{t \rightarrow +\infty} E(X_{cj}^t). \quad (19)$$

则每只鸽子的平均收敛位置可表示为

$$\begin{aligned} & \lim_{t \rightarrow +\infty} E(X_i^t) \\ &= \lim_{t \rightarrow +\infty} (E(X_{i1}^t), E(X_{i2}^t), \dots, E(X_{iD}^t)) \\ &= (\lim_{t \rightarrow +\infty} E(X_{i1}^t), \lim_{t \rightarrow +\infty} E(X_{i2}^t), \dots, \lim_{t \rightarrow +\infty} E(X_{iD}^t)) \\ &= (\lim_{t \rightarrow +\infty} E(X_{c1}^t), \lim_{t \rightarrow +\infty} E(X_{c2}^t), \dots, \lim_{t \rightarrow +\infty} E(X_{cD}^t)) \\ &= \lim_{t \rightarrow +\infty} E(X_c^t), k \in \{1, 2, \dots, N_p\}. \end{aligned} \quad (20)$$

由于每次迭代会减少一半的鸽子, 当经过有限的 t 次迭代后, 种群数量 N_p 会减小到1, 由RPIO的地标中心位置计算可得

$$X_c^t = \operatorname{argmin} \{f(X_1^t), f(X_2^t), \dots, f(X_{N_p}^t)\}. \quad (21)$$

此时, X_c^t 即为当前第 t 代鸽群的最优位置 $\mathbf{X}_g^t$. 由式(20)和式(21), 只要存在这个极限, 当 t 趋近于无穷时, 每个个体都将收敛到同一个平均位置 $\lim_{t \rightarrow +\infty} E(X_c^t)$, 亦即 $\lim_{t \rightarrow +\infty} E(X_g^t)$.

当考虑反射策略时, 由式(15)可以得到

$$X_i^t = X_{i\text{reflected}}^t = (1 + \alpha)X_g^t - \alpha X_i^t, \quad (22)$$

$$(1 + \alpha)X_i^t = (1 + \alpha)X_g^t. \quad (23)$$

本文中 α 代表反射度, 是一个非负数. 用 X_{ij}^t 和 X_{gj}^t 代表 X_i^t 和 X_g^t 的第 j 个分量, $j=1, 2, \dots, D$. 则期望序列 $\{E(X_{ij}^t)\}_{t=1}^{+\infty}$ 的差分方程的极限可以表示为

$$\lim_{t \rightarrow +\infty} E(X_{ij}^t) = \lim_{t \rightarrow +\infty} E(X_{gj}^t). \quad (24)$$

从而可以得到每只鸽子的平均收敛位置

$$\begin{aligned} & \lim_{t \rightarrow +\infty} E(X_i^t) \\ &= \lim_{t \rightarrow +\infty} (E(X_{i1}^t), E(X_{i2}^t), \dots, E(X_{iD}^t)) \\ &= (\lim_{t \rightarrow +\infty} E(X_{i1}^t), \lim_{t \rightarrow +\infty} E(X_{i2}^t), \dots, \lim_{t \rightarrow +\infty} E(X_{iD}^t)) \\ &= (\lim_{t \rightarrow +\infty} E(X_{g1}^t), \lim_{t \rightarrow +\infty} E(X_{g2}^t), \dots, \lim_{t \rightarrow +\infty} E(X_{gD}^t)) \\ &= \lim_{t \rightarrow +\infty} E(X_g^t), k \in \{1, 2, \dots, N_p\}. \end{aligned} \quad (25)$$

因此, 由式(20)和式(25)可以得出, 鸽群中每个个体的平均位置都将收敛于当前最优解 $\lim_{t \rightarrow +\infty} E(X_g^t)$.

2) 反射鸽群全局收敛性分析.

前文中指出, 当 t 趋近于无穷时每只鸽子的平均位

置都将收敛于当前的最优解, 但这并不能说明收敛位置为目标函数的全局最优解, 因此下面利用鞅理论研究RPIO算法的全局收敛性.

鞅是一个随机的过程, 在给定上一个值以及当前值的情况下, 下一个值的条件期望等于当前值. 超鞅是鞅的一种推广形式, 表示随机过程的期望值在给定过去的信息条件下是非递增的^[20]. 超鞅的定义以及收敛定理如下.

定义1 超鞅^[20] 对于两个随机过程 $\{Y_k\}_{k=0}^{+\infty}$ 和 $\{Z_k\}_{k=0}^{+\infty}$, 对于任意的 $k > 0$, 当满足下面三个条件时, 将 $\{Y_k\}_{k=0}^{+\infty}$ 称为 $\{Z_k\}_{k=0}^{+\infty}$ 的超鞅:

- (1) $E(|Y_k|) < +\infty$;
- (2) $E(Y_{k+1} | Z_0, Z_1, \dots, Z_j) \leq Y_j$;
- (3) Y_j 是 $Z_0, Z_1, \dots, Z_k$ 的函数.

定理1 超鞅收敛定理^[20] 假设 $\{Y_k\}_{k=0}^{+\infty}$ 是一个关于 $\{Z_k\}_{k=0}^{+\infty}$ 的超鞅, 且存在上界, 那么存在一个几乎必然有限的随机变量 Y_∞ , 使得 $\{Y_k\}_{k=0}^{+\infty}$ 收敛到 Y_∞ 且概率为1, 即, $P(\lim_{j \rightarrow +\infty} Y_k = Y_\infty) = 1$.

对于任意鸽群, 其鸽群状态可表示为 $\eta(t) = (X_1^t, \dots, X_{N_p}^t, X_g^t)$, $t = 0, 1, \dots$, 适应度函数记为 $F(\eta(t))$. 假设 $f(\cdot)$ 为最小化问题的目标函数, 定义为

$$F(\eta(t)) \triangleq \min \{f(X_1^t), \dots, f(X_{N_p}^t), f(X_g^t)\}, \quad (26)$$

式中, $\mathbf{X}_g^t$ 是种群在第 t 次迭代时的全局最优位置, 定义为 $F(\eta(t)) = f(X_g^t)$.

因此, $\{F(\eta(t))\}_{t=0}^{+\infty}$ 是一个单调非递增序列. 根据超鞅的定义可知, 随机过程 $\{F(\eta(t))\}_{t=0}^{+\infty}$ 是相对于 $\{\eta(t)\}_{t=0}^{+\infty}$ 的超鞅过程. 则根据定理1, 存在一个几乎必然有限的随机变量 $F(\eta(\infty))$, 使 $\{F(\eta(t))\}_{t=0}^{+\infty}$ 随趋于无穷而收敛到全局最优解的概率为1. 同样, $\{\eta(t)\}_{t=0}^{+\infty}$ 也满足该定理, 即 $P(\lim_{t \rightarrow +\infty} \eta(t) = \eta(\infty)) = 1$.

定义2 基于RPIO的全局收敛性 假设与RPIO相关的离散时间随机过程为 $\{\eta(t)\}_{t=0}^{+\infty}$. 若 $\lim_{t \rightarrow +\infty} P(\eta(t) \in \Omega^*(\varepsilon)) = 1$ 对任意的 $0 < \varepsilon < \mathbb{R}$ 成立, 则RPIO全局收敛.

引理1^[17] 若 $\sum_{k=0}^{\infty} a_k b_k < +\infty (0 \leq a_k, b_k \leq 1, k=0, 1, \dots)$,

存在两种可能:

(1) 若 $\sum_{k=0}^{\infty} a_k = +\infty$, 则 $\lim_{k \rightarrow \infty} b_k = 0$.

(2) 若 $\sum_{k=0}^{\infty} a_k < +\infty$, 则 $\lim_{k \rightarrow \infty} b_k = \delta \in [0, 1]$ 或不存在.

若用 Ω 代表 RPIO 鸽群的状态空间, $\Omega^*(\varepsilon)$ 代表鸽群的 ε -全局最优状态, q_t^* 代表鸽群在 t 时进入 ε -全局最优状态的最小概率^[17], 则可以得到定理如下.

定理2 令

$$q_t^* = \min_{y \in \Omega \setminus \Omega^*(\varepsilon)} P(\eta(t+1) \in \Omega^*(\varepsilon) \mid \eta(t) = y), t=0, 1, \dots,$$

因此可得, 只有当 $\sum_{k=0}^{\infty} q_k^* = +\infty$ 时 RPIO 算法的全局收敛性能得到保证.

证明 若 $\eta(t)$ 的概率分布函数为 $F_t(y)$, 当 $\eta(t) = y$ 时 $\eta(t+1)$ 的条件概率分布为 $F_t(z \mid y)$, 因此由条件期望的性质可得

$$\begin{aligned} & E(F(\eta(t+1))) - E(F(\eta(t))) \\ &= E[E(F(\eta(t+1)) \mid \eta(t))] - E(F(\eta(t))) \\ &= \int_{\Omega} E(F(\eta(t+1)) \mid \eta(t) = y) dF_t(y) - \int_{\Omega} F(y) dF_t(y) \\ &= \int_{\Omega} \left[\int_{\Omega} F(z) dF_t(z \mid y) \right] dF_t(y) - \int_{\Omega} F(y) dF_t(y) \\ &= \int_{\Omega} \left[\int_{\Omega} F(z) dF_t(z \mid y) - F_t(y) \right] dF_t(y) \\ &= \int_{\Omega} \left[\int_{\Omega} F(z) dF_t(z \mid y) - \int_{\Omega} F(y) dF_t(z \mid y) \right] dF_t(y) \\ &= \int_{\Omega} \left[\int_{\Omega} (F(z) - F(y)) dF_t(z \mid y) \right] dP(y). \end{aligned} \quad (27)$$

由于 $\Omega^*(\varepsilon) \subset \Omega$, 因此可以得到

$$\begin{aligned} & \int_{\Omega} \left[\int_{\Omega} (F(z) - F(y)) dF_t(z \mid y) \right] dF_t(y) \\ & \geq \int_{\Omega \setminus \Omega^*(\varepsilon)} \left[\int_{\Omega^*(\varepsilon)} (F(z) - F(y)) dF_t(z \mid y) \right] dF_t(y). \end{aligned} \quad (28)$$

设 $S^*(\varepsilon) = \{X \in S \mid f(X) > f^* - \varepsilon\}$ 为全局最优 ε -邻域, 则鸽群的 ε -全局最优状态可表示为 $\Omega^*(\varepsilon) = \{\eta = (X_1, \dots, X_{N_p}, \xi) \mid \exists X_k \in S^*(\varepsilon), k=1, \dots, N_p, \xi \in S\}$. 记 $\alpha = \min\{F(z) - F(y) \mid z \in \Omega^*(\varepsilon), y \in \Omega \setminus \Omega^*(\varepsilon)\}$, 则

$F(y) \leq f^* - \varepsilon < F(z)$, 可得 $\alpha > 0$ 且

$$\begin{aligned} & \int_{\Omega \setminus \Omega^*(\varepsilon)} \left[\int_{\Omega^*(\varepsilon)} (F(z) - F(y)) dF_t(z \mid y) \right] dF_t(y) \\ & \geq \alpha \int_{\Omega \setminus \Omega^*(\varepsilon)} \left[\int_{\Omega^*(\varepsilon)} dF_t(z \mid y) \right] dF_t(y) \\ &= \alpha \int_{\Omega \setminus \Omega^*(\varepsilon)} P(\eta(t+1) \in \Omega^*(\varepsilon) \mid \eta(t) = y) dF_t(y) \\ & \geq \alpha q_t^* \int_{\Omega \setminus \Omega^*(\varepsilon)} dF_t(y) = \alpha q_t^* P(\eta(t) \notin \Omega^*(\varepsilon)). \end{aligned} \quad (29)$$

由式(27)和式(29)可得

$$\begin{aligned} & \sum_{t=0}^M [E(F(\eta(t+1))) - E(F(\eta(t)))] \\ &= E(F(\eta(M+1))) - E(F(\eta(0))) \\ & \geq \alpha \sum_{t=0}^M [q_t^* P(\eta(t) \notin \Omega^*(\varepsilon))]. \end{aligned} \quad (30)$$

已知超鞅过程 $\{F(\eta(t))\}_{t=0}^{+\infty}$ 有上界, 令 M 趋近于 $+\infty$, 则可推出

$$\sum_{t=0}^{\infty} [q_t^* P(\eta(t) \notin \Omega^*(\varepsilon))] < +\infty. \quad (31)$$

利用引理1对式(31)进行分析讨论, 可以得到以下两种情况.

(1) 若 $\sum_{k=0}^{\infty} q_k^* = +\infty$, 结合引理1的第一种情况, 则

$\lim_{t \rightarrow +\infty} P(\eta(t) \notin \Omega^*(\varepsilon)) = 0$, 即 $\lim_{t \rightarrow +\infty} P(\eta(t) \in \Omega^*(\varepsilon)) = 1$. 由于 $\{\eta(t)\}_{t=0}^{+\infty}$ 会收敛到 $\eta(\infty)$, 因此, $\eta(\infty)$ 几乎肯定会在 $\Omega^*(\varepsilon)$ 中取值, 从而保证了 RPIO 的全局收敛性.

(2) 若 $\sum_{k=0}^{\infty} q_k^* \neq +\infty$, 即 $\sum_{k=0}^{\infty} q_k^* < +\infty$, 结合引理1中的

第二种情况, 可以得到 $\lim_{t \rightarrow +\infty} P(\eta(t) \notin \Omega^*(\varepsilon)) = \delta \in [0, 1]$, 或者极限不存在. 当 $\delta = 0$ 时, 与(1)中相同, RPIO 具有全局收敛性. 当 $\delta \neq 0$ 时, $\eta(\infty)$ 可能取到 $\Omega^*(\varepsilon)$ 之外的值, 因此 RPIO 不是全局收敛的. 当极限不存在时, 显然 RPIO 也不是全局收敛的. 因此, 这种情况下无法保证 RPIO 算法的全局收敛性.

综上, 当 $\sum_{k=0}^{\infty} q_k^* = +\infty$ 时, 能够保证 RPIO 算法的全局

收敛性, 而当 $\sum_{k=0}^{\infty} q_k^* < +\infty$ 时无法保证 RPIO 的全局收敛性. 但随着 q_t^* 增大, 鸽群更容易接近全局最优解, 只

要 q_i^* 累积足够大, RPIO算法仍然可以得到全局最优解.

根据以上数学描述, 可以分析RPIO算法的时间复杂度. 在地图和指南针算子阶段, 每轮迭代的时间复杂度为 $O(N_p D)$. 在地标算子阶段, 由于计算地标中心位置时需要鸽子快速排序, 因此地标算子阶段每轮迭代的时间复杂度为 $O(N_p D + N_p \log N_p)$. 加入反射学习策略后, 每轮迭代中每只鸽子都会额外生成一只反射鸽子, 从而可以得到每轮迭代中反射学习策略的复杂度为 $O(N_p D)$. 因此, 地图和指南针算子阶段的时间复杂度为 $O(N_p D N_{c1max})$, 地标算子阶段的时间复杂度为 $O((N_p D + N_p \log N_p) N_{c2max})$, 从而可以得到RPIO算法的时间复杂度为 $O((N_{c1max} + N_{c2max}) N_p D + N_{c2max} N_p \log N_p)$.

3.4 适应度函数选取及控制律优化

在空中加油过程中需要快速准确的控制无人机的姿态, 因此需要保证控制律既能快速地跟踪姿态角目标指令, 又能保证跟踪精度, 即较小的跟踪误差. 因此, 可以选用ITAE积分作为优化算法的适应度函数, ITAE地具体公式如下:

$$ITAE = \int_0^{+\infty} t |e(t)| dt, \quad (32)$$

式中, t 代表调节时间, $e(t)$ 代表跟踪误差. ITAE积分值越小, 相应的调节时间 t 和跟踪误差 $e(t)$ 也越小, 这表明跟踪速度和精度也越高. 总的来说, ITAE值越小, 系统的控制效果越好.

在选取完成合适的适应度函数值后, 可以在SIMULINK中建立无人机自抗扰控制律模型, 如图3所示.

图3中, ADRC控制律中的微分跟踪器、扩张状态观测器和非线性状态误差反馈律三部分都用SIMULINK中的S函数建模, UAV为无人机的六自由度模型. 控制量从NLSEF中输出, 并在经过计算后输入到无人机模型中. 随后无人机模型输出俯仰、滚转和偏航三个通道的响应, 并将输出的角度反馈到扩张状态观测器中. 同时, 可以由输出值与输入值计算得到误差, 进而得到ITAE积分值. ITAE为反射鸽群优化算法的适应度函数, 每次迭代前算法将需要优化的9个参数值输入到模型中, 并通过仿真得到适应度值. 随着算法的迭代, 适应度函数值逐渐减小, 算法逐渐收敛到全局

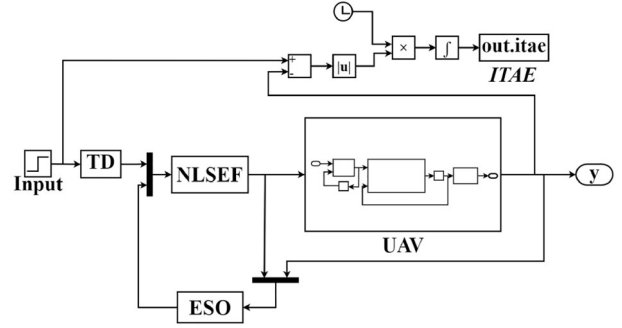


图3 无人机ADRC控制律模型

Figure 3 ADRC control law model of UAV.

最优解, ADRC的9个优化参数也逐渐迭代到最优取值, 最终完成自抗扰控制律的设计.

4 仿真与结果分析

本文利用MATLAB/SIMULINK进行建模仿真, 设计了ADRC控制律, 其中无人机模型采用文献[2]中的模型. 本文首先将ADRC控制方法与传统PID控制方法进行比较, 之后用RPIO算法对ADRC控制律进行了优化设计仿真, 并用PIO和PSO算法对ADRC控制律进行了优化设计仿真, 最后将三种算法的仿真结果进行了对比分析.

4.1 ADRC抗干扰性验证

为了验证ADRC控制方法的抗干扰性, 本文进行了ADRC与传统PID控制方法的对比仿真实验, 同时加入了等幅的高斯噪声, 以模拟自主空中加油时的气流扰动. 本文利用俯仰通道进行测试, 选取的俯仰角指令为 1° , 仿真步长为0.02 s. ADRC控制律共有 r_0 , h_0 , β_{02} , β_{03} , b_0 , α_{11} , α_{12} , β_1 , β_2 等九个参数需要优化, 在本节的对比仿真中使用粗调后的初始参数, 如表1所示.

通过仿真, 两种控制律的控制效果如图4所示. 加入等幅的高斯噪声后, 两种控制律的阶跃响应曲线如图5所示.

从图4中所示的控制效果曲线可以看出, ADRC控制律具有更短的调节时间以及更小的超调量. 从图5中可以看出, ADRC的响应曲线更加平滑, 由此可知, 相比于传统PID控制律, ADRC具有更好的抗干扰能力, 在空中加油的复杂环境中具有更好的控制效果.

表 1 ADRC粗调后的初始参数

Table1 Initial parameters after coarse adjustment of ADRC

参数	粗调后的初始参数
r_0	50
h_0	0.5
β_{02}	10
β_{03}	30000
b_0	1
α_{11}	0.2
α_{12}	0.6
β_1	5
β_2	8

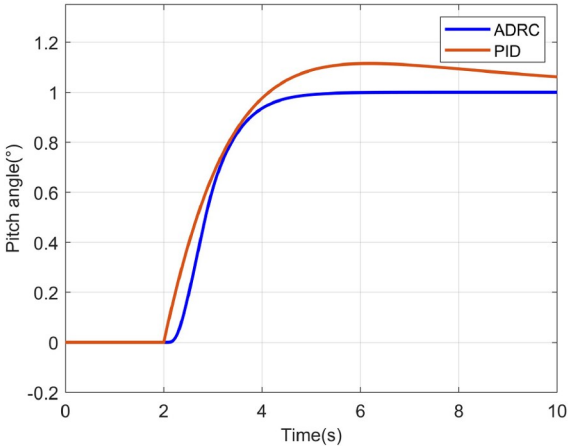


图 4 (网络版彩图)ADRC与PID的控制效果对比图
Figure 4 (Color online) Comparison of control effect of ADRC and PID.

表 2 RPIO, PIO以及PSO的初始参数

Table2 Initial parameters of RPIO, PIO and PSO

算法	参数	含义	取值
RPIO和PIO	N_{c1max}	地图和指南针算子迭代次数	40
	N_{c2max}	地标算子迭代次数	10
	N_p	种群数量	20
	R	地图和指南针算子	0.2
	α	反射度	1
PSO	T	迭代次数	50
	N_p	种群数量	20
	ω	惯性权重系数	0.9
	c_1	自我学习因子	0.5
	c_2	群体学习因子	0.5

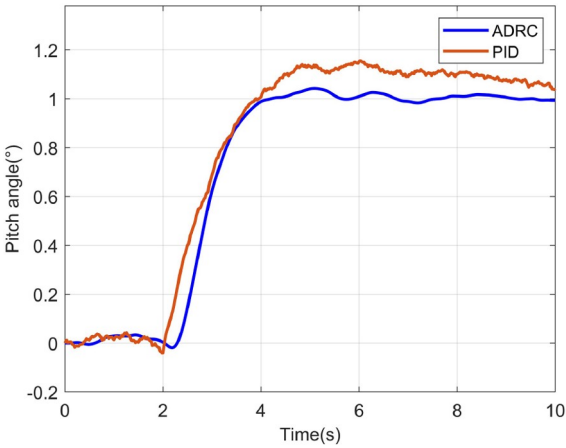


图 5 (网络版彩图)加入干扰后ADRC与PID的控制效果对比图
Figure 5 (Color online) Comparison of control effect of ADRC and PID after adding disturbance.

4.2 初始参数设置

RPIO, PIO以及PSO三种算法的具体初始参数如表2所示.

4.3 三种算法优化设计仿真及结果

本文中对无人机的俯仰、滚转和偏航三个通道的参数进行了优化设计. 仿真过程中采用的仿真步长为 0.02 s, 输入目标信号是一个阶跃信号, 在2秒时刻发生突变, 幅度变化为1°. ADRC控制律共有 r_0 , h_0 , β_{02} , β_{03} , b_0 , α_{11} , α_{12} , β_1 , β_2 等九个参数需要优化. 经过RPIO算法优化设计后, 控制律性能显著得到提高, 控制律参

数优化前后的具体数值如表3所示。

利用PIO和PSO算法对控制律进行优化设计仿真时, 选用的输入信号、仿真步长以及各参数的取值范围与RPIO的相同。俯仰、滚转和偏航三个通道经过PIO和PSO算法优化后的仿真结果如表4所示。

4.4 RPIO算法与其他算法结果对比分析

通过仿真得到了RPIO算法、PIO和PSO算法所生成的三个通道适应度函数的迭代过程曲线, 如图6~8所示; RPIO算法、PIO和PSO算法优化设计后控制律的阶跃响应曲线如图9~11所示。

从图6~8所示的适应度函数曲线可以看出RPIO算法优化设计控制律的适应度函数显著减小(三者最小), 因此表明, RPIO算法更容易跳出局部最优解, 能够更快地到达最优位置, 且最终收敛的位置明显优于其他两种算法。这种快速收敛的特性意味着在实际应用中, RPIO能够更高效地实现控制律的优化, 减少了计算时间和资源消耗, 为实时控制系统的应用提供了更为可靠的支持。

从图9~11所示的RPIO算法、PIO, PSO算法优化设计的和未优化的三个通道阶跃响应曲线中可以明显看出, 经过优化算法优化后自抗扰控制律响应速度大

表 3 RPIO优化前后三个通道的参数

Table 3 Parameters of the three channels before and after RPIO

参数	俯仰通道			滚转通道			偏航通道		
	优化前	优化后	取值范围	优化前	优化后	取值范围	优化前	优化后	取值范围
r_0	50	99.3486	0~100	50	99.1989	0~100	50	100	0~100
h_0	0.5	0.033679	0~1	0.5	0.011276	0~1	0.7	0.043257	0~1.5
β_{02}	10	0.03039	0~20	10	11.5337	0~20	10	4.2169	0~20
β_{03}	30000	34898.1409	25000~35000	15000	15440.8125	14000~16000	15000	15932.075	14000~16000
b_0	1	0.54954	0~2	1	0.53229	0~2	1	1.0806	0~2
α_{11}	0.2	0.38453	0~0.4	0.1	0.15515	0~0.2	0.15	0.099273	0~0.3
α_{12}	0.6	1.0185	0~1.2	0.6	0.9477	0~1.2	0.6	1.0384	0~1.2
β_1	5	4.758	0~10	10	5.1291	0~20	4	4.1713	0~8
β_2	8	12	4~12	14	20	8~20	10	14.9617	5~15
ITAE	2.63	0.803		2.668	0.641		4.185	0.549	

表 4 PIO和PSO优化前后三个通道的参数

Table 4 Parameters of the three channels before and after PIO and PSO

参数	俯仰通道			滚转通道			偏航通道		
	优化前	PIO优化后	PSO优化后	优化前	PIO优化后	PSO优化后	优化前	PIO优化后	PSO优化后
r_0	50	95.8538	79.0574	50	86.5678	58.5039	50	83.4964	100
h_0	0.5	0.010533	0.010827	0.5	0.013902	0.018533	0.7	0.014168	0.019782
β_{02}	10	18.9017	0	10	17.3439	8.0872	10	0.17901	0
β_{03}	30000	34985.6898	26915.2371	15000	14784.7722	15837.7107	15000	15914.0897	14469.0752
b_0	1	0.5043	0.62234	1	0.53255	0.54814	1	1.8375	1.0336
α_{11}	0.2	0.22014	0.4	0.1	0.1247	0.2	0.15	0.26398	0.3
α_{12}	0.6	0.96706	0.99777	0.6	0.9024	1.19913	0.6	1.0529	1.1591
β_1	5	1.6489	10	10	9.4552	5.0843	4	4.3925	7.9446
β_2	8	11.6425	11.8154	14	17.9906	20	10	7.8363	9.6854
ITAE	2.63	0.869	0.895	2.668	0.709	0.728	4.185	0.652	0.664

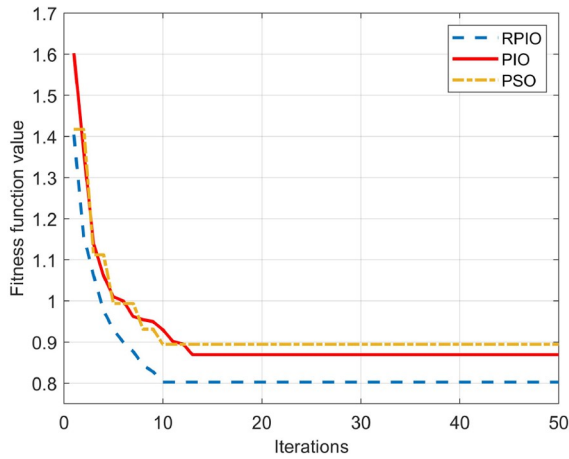


图 6 (网络版彩图)俯仰通道的适应度函数迭代曲线

Figure 6 (Color online) Fitness function iteration curves of pitch channel.

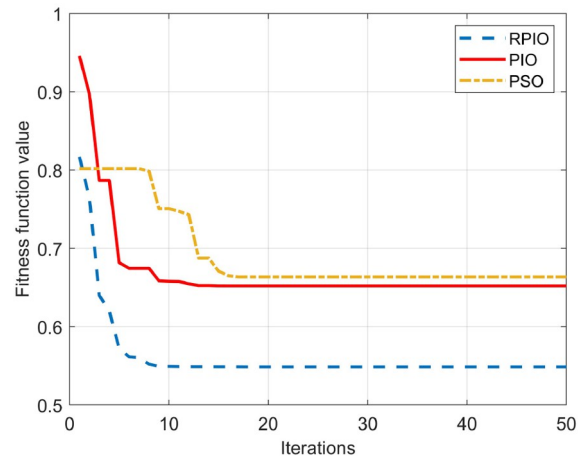


图 8 (网络版彩图)偏航通道的适应度函数迭代曲线

Figure 8 (Color online) Fitness function iteration curves of yaw channel.

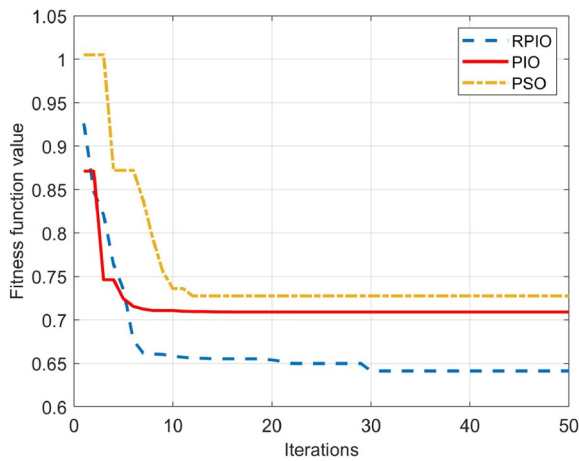


图 7 (网络版彩图)滚转通道的适应度函数迭代曲线

Figure 7 (Color online) Fitness function iteration curves of roll channel.

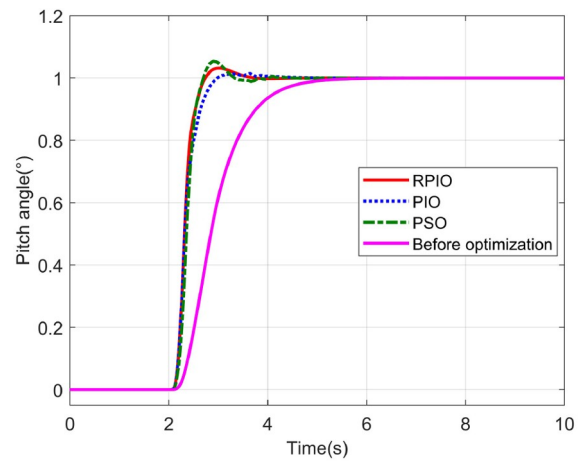


图 9 (网络版彩图)俯仰通道的阶跃响应曲线

Figure 9 (Color online) Step response curves of pitch channel.

大加快, 控制效果得到了显著提升. 在俯仰通道中, RPIO优化后的控制律的超调量为3.2%, 而PIO和PSO算法分别为1.4%和5.4%. RPIO算法优化的控制律虽然超调量稍大于PIO算法, 但其响应速度更快, 调节时间更短. 从图中可以看出RPIO算法优化后的控制律的上升时间为0.7 s, 峰值时间为1 s, 明显快于PIO算法. 在滚转通道中, RPIO算法优化后的控制律的超调量仅为1.6%, 而PIO和PSO算法优化后控制律的超调量分别为7.6%和8.2%, 远大于RPIO. 并且RPIO算法优化后的控制律能在3.3 s左右就稳定在期望角度, 明显快于另外两种算法, 这表明控制律能够在保持系统稳定性的同

时, 快速达到期望状态. 在偏航通道中, RPIO优化后的控制律的超调量虽然稍大于PIO, 但上升时间仅为2.4 s左右, 远小于PIO算法. 此外, 仿真得到RPIO优化后的俯仰、滚转、偏航通道的5%误差范围的调节时间分别为0.61, 0.45和0.36 s, 这表明RPIO优化后的控制律能够快速及时地响应控制指令, 从而在复杂动态环境中实现快速精确的姿态调整, 满足空中加油过程中无人机控制的实时性要求.

综合来看, 与经典PIO算法以及PSO算法相比, RPIO优化设计的控制律在各个控制通道上均表现出更好的稳定性、鲁棒性和快速性. 因此, RPIO优化设

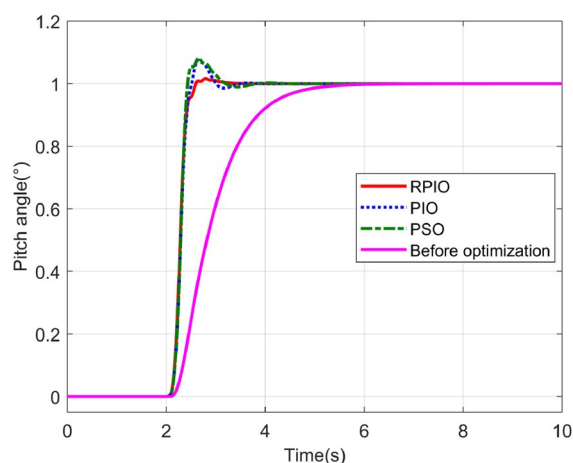


图 10 (网络版彩图)滚转通道的阶跃响应曲线
Figure 10 (Color online) Step response curves of roll channel.

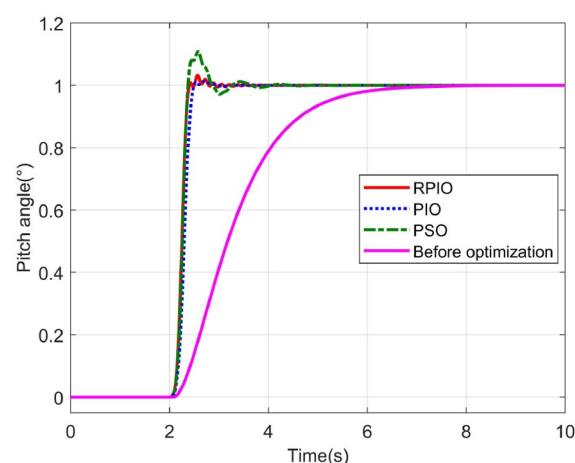


图 11 (网络版彩图)偏航通道的阶跃响应曲线
Figure 11 (Color online) Step response curves of yaw channel.

计的控制律在应用于实际的自主空中加油任务时, 能够更好地满足精确控制和响应速度的需求。

5 结论

本文主要设计了一个基于自抗扰控制的无人机姿态控制律, 进而基于镜面反射策略提出了一种用反射鸽群优化(RPIO)算法实现姿态控制律参数优化设计的方法, 并对其收敛性进行了分析和验证, 文中给出了

RPIO算法、PIO和PSO算法的设计仿真实例。相较于经典鸽群优化算法, RPIO通过引入反射策略, 有效降低了收敛到局部最优的风险, RPIO算法在面对复杂的多维优化问题时能够表现出更强的全局搜索能力, 从而能够更全面地探索解空间, 找到更优的控制参数。仿真结果表明, 经过RPIO优化的姿态控制律具有更好的控制性能, 从而能够更好地保证无人机的响应速度和稳定性, 确保了自主空中加油过程的可靠性与安全性。

参考文献

- Ren J, Quan Q. Progress in modeling and control of probe-and-drogue autonomous aerial refueling. *Chin J Aeronautics*, 2024, 37: 6–26
- Duan H B, Sun Y B. Autonomous Control of UAV Aerial Refueling (in Chinese). Shanghai: Shanghai Jiao Tong University Press, 2024 [段海滨, 孙永斌. 无人机空中加油自主控制. 上海: 上海交通大学出版社, 2024]
- Guo L, Yu X, Zhang X, et al. Safety control system technologies for UAVs: Review and prospect (in Chinese). *Sci Sin Inform*, 2020, 50: 184–194 [郭雷, 余翔, 张霄, 等. 无人机安全控制系统技术: 进展与展望. 中国科学: 信息科学, 2020, 50: 184–194]
- Wang J, Hovakimyan N, Cao C. Verifiable adaptive flight control: Unmanned combat aerial vehicle and aerial refueling. *J Guidance Control Dyn*, 2010, 33: 75–87
- Su Z, Wang H, Li N, et al. Exact docking flight controller for autonomous aerial refueling with back-stepping based high order sliding mode. *Mech Syst Signal Processing*, 2018, 101: 338–360
- Su Z, Wang H. Probe motion compound control for autonomous aerial refueling docking. *Aerospace Sci Tech*, 2018, 72: 1–13
- Han J. From PID to active disturbance rejection control. *IEEE Trans Ind Electron*, 2009, 56: 900–906
- Duan H, Sun Y, Shi Y. Bionic visual control for probe-and-drogue autonomous aerial refueling. *IEEE Trans Aerosp Electron Syst*, 2021, 57: 848–865
- Duan H, Qiao P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int J Intelligent Computing Cybernetics*, 2014, 7: 24–37
- Duan H, Huo M, Yang Z, et al. Predatorprey pigeon-inspired optimization for UAV ALS longitudinal parameters tuning. *IEEE Trans Aerosp*

[Electron Syst](#), 2019, 55: 2347–2358

- 11 Hai X, Wang Z, Feng Q, et al. Mobile robot ADRC with an automatic parameter tuning mechanism via modified pigeon-inspired optimization. [IEEE ASME Trans Mechatron](#), 2019, 24: 2616–2626
- 12 Hai X, Wang Z, Feng Q, et al. A novel adaptive pigeon-inspired optimization algorithm based on evolutionary game theory. [Sci China Inf Sci](#), 2021, 64: 139203
- 13 Fei L, Duan H B, Xu X B, et al. ADRC controller design for UAV based on variable weighted mutant pigeon inspired optimization (in Chinese). *Acta Astronaut*, 2020, 41: 323490 [费伦, 段海滨, 徐小斌, 等. 基于变权重变异鸽群优化的无人机空中加油自抗扰控制器设计. *航空学报*, 2020, 41: 323490]
- 14 Wu Z, Zhou H, Guo B, et al. On the convergence of tracking differentiator with multiple stochastic disturbances. [Sci China Inf Sci](#), 2024, 67: 122203
- 15 Zhang T, Ng J Y, Tai V C. Comparative analysis of attitude control of quadrotor UAVs: PID, ADRC, and NMPC algorithms. In: *Proceedings of 2024 9th International Conference on Electronic Technology and Information Science (ICETIS)*. Hangzhou, 2024. 210–215
- 16 Ren J, Quan Q, Ma H, et al. Additive-state-decomposition-based station-keeping control for autonomous aerial refueling. [Sci China Inf Sci](#), 2021, 64: 219202
- 17 Duan H B, Huo M Z. *Pigeon-Inspired Optimization* (in Chinese). Beijing: Science Press, 2023 [段海滨, 霍梦真. 鸽群优化. 北京: 科学出版社, 2023]
- 18 Zhang Y. Backtracking search algorithm with specular reflection learning for global optimization. [Knowledge-Based Syst](#), 2021, 212: 106546
- 19 Ali Rodan. Reflected PSO for robot path planning. In: *Proceedings of 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. Madrid, 2024. 1–8
- 20 Nguyen H T, Wang T H. *A graduate Course in Probability and Statistics, Volume I, Essentials of Probability for Statistics*. Beijing: Tsinghua University Press, 2008

Active disturbance rejection control method for UAV autonomous aerial refueling based on reflected pigeon-inspired optimization

GONG JiaWei & FAN YanMing

Shenyang Aircraft Design and Research Institute, Aviation Industry Corporation of China (AVIC), Shenyang 110035, China

This study proposes an attitude controller design method for unmanned aerial vehicles (UAVs) based on the reflected pigeon-inspired optimization (RPIO) algorithm, addressing the attitude control problem of the receiver aircraft during aerial refueling. First, a UAV attitude controller based on active disturbance rejection control (ADRC) is designed. Then, the controller parameters are optimized using the RPIO algorithm to form the final controller. The design results obtained using the RPIO algorithm are compared with those of the traditional pigeon-inspired optimization algorithm and the particle swarm optimization algorithm. Simulation results show that the RPIO algorithm introduced in this paper outperforms the traditional algorithm, significantly improving the accuracy and stability of UAV attitude control during autonomous aerial refueling.

aerial refueling, reflected pigeon-inspired optimization, active disturbance rejection control

doi: [10.1360/SST-2024-0228](https://doi.org/10.1360/SST-2024-0228)