

基于多种群蜻蜓算法的四旋翼自抗扰姿态优化控制

李茂林 王勇杰 介丹

(运城职业技术大学电子信息工程系 山西 运城 044000)

摘要 针对四旋翼自抗扰控制器耦合参数较多,依靠人工经验难以整定,导致控制效果和抗扰动能力不佳的问题,提出一种多种群蜻蜓算法的四旋翼自抗扰控制参数整定策略。针对传统蜻蜓算法全局搜索能力和局部勘探能力不平衡、易陷入局部最优早熟收敛的问题,提出一种动态混沌惯性权重,平衡算法的全局搜索能力和局部勘探能力,并通过引入位置遗弃策略和多种群策略提高算法的搜索精度,避免算法陷入局部最优。将改进后的蜻蜓算法在线整定自抗扰参数。实验结果表明,对比其他控制策略,该方法具有更高的控制精度和更好的抗扰动能力。

关键词 四旋翼 自抗扰参数整定 蜻蜓算法 动态混沌权重 位置遗弃 多种群

中图分类号 TP3 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2022.12.049

QUADROTOR DISTURBANCE REJECTION ATTITUDE CONTROL BASED ON MULTIGROUP DRAGONFLY ALGORITHM

Li Maolin Wang Yongjie Jie Dan

(Department of Electronic Information Engineering, School of Yuncheng Vocational and Technical, Yuncheng 044000, Shanxi, China)

Abstract In order to solve the problem that there are many coupling parameters in the quadrotor ADRC, and it is difficult to adjust the parameters by artificial experience, which leads to the poor control effect and anti disturbance ability, this paper proposes a quadrotor ADRC parameter adjustment strategy based on multigroup dragonfly algorithm. Aiming at the problem that the global search ability and local exploration ability of the traditional dragonfly algorithm were unbalanced and easy to fall into the local optimal premature convergence, a dynamic chaos inertia weight was proposed to balance the global search ability and local exploration ability of the algorithm. The search accuracy of the algorithm was improved by introducing the location abandonment strategy and the multi population strategy to avoid the algorithm falling into the local optimum. The improved dragonfly algorithm was optimized to adjust the auto disturbance rejection parameters online. Experimental results show that compared with other control strategies, this method has higher control accuracy and better anti-disturbance ability.

Keywords Quadrotor Auto disturbance rejection parameter setting Dragonfly algorithm Dynamic chaos weight Location abandonment Multigroup

0 引言

四旋翼飞行器是一类多输入多输出的非线性飞行器,具有强耦合性且难以建立精确的数学模型^[1],传统的PID控制^[2]、鲁棒控制^[3]和神经网络控制^[4]等控制策略的控制效果均依赖被控对象的数学模型,因此针

对具有难以建立精确数学模型特点的被控对象,传统的控制策略难以得到最优控制效果。此外,四旋翼飞行器在飞行过程中,易受到外界扰动,偏离预定飞行高度或飞行航线,导致飞行器在飞行过程中对于控制器的抗扰动能力要求极高。因此,针对上述特点,四旋翼飞行器的控制设计成为了国内外研究学者的研究重点。

古训等^[5]提出一种自适应滑模控制策略,实现了飞行器的姿态控制和高度控制,提高了控制精度和响应速度。胡文华等^[6]提出一种基于改进粒子群算法的自抗扰四旋翼控制策略,该策略针对传统粒子群算法求解精度不高的问题进行改进,并用于自抗扰参数整定,提高了控制精度。陈林奇等^[7]提出一种双空间粒子群算法的自抗扰四旋翼控制策略,该策略实现了飞行器到达指令高度的快速无超调,但该策略中粒子群算法难以针对扰动计算出当前有效的控制参数,使得控制器的抗扰动能力不强。Wang 等^[8]提出一种基于干扰观测器的四旋翼自适应容错控制,针对外负载扰动具有较强的观测能力,但该策略的缺点在于系统受到扰动后,再次达到稳态的时间较长。卢凯文等^[9]提出一种具有 LQR 的四旋翼控制策略,该策略较传统 PID 控制的控制精度更高,但抗扰动能力较差。

为提高四旋翼飞行器的控制精度和抗扰动能力,本文首先选取非线性自抗扰作为四旋翼飞行器的控制器。其次针对非线性自抗扰控制器参数难以整定的问题,通过动态混沌权重、位置遗弃和多种群策略对蜻蜓算法进行改进,将改进后的蜻蜓算法用于自抗扰控制器参数的在线整定。最后将优化后的自抗扰控制器对四旋翼飞行器进行姿态控制和高度控制。

1 数学模型建立

1.1 四旋翼飞行器数学模型

设四旋翼飞行器在地面坐标系下的坐标为 $O_L = (x, y, z)$, 飞行器的俯仰角、滚动角和航向角分别记为 α 、 β 和 γ 。因此定义四旋翼飞行器的动力学方程的数学模型为:

$$\begin{cases} \ddot{x} = (\cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha)\frac{U_1}{m} \\ \ddot{y} = (\sin\gamma\sin\beta\cos\alpha - \cos\gamma\sin\alpha)\frac{U_1}{m} \\ \ddot{z} = (\cos\beta\cos\alpha)\frac{U_1}{m} - g \\ \ddot{\alpha} = \frac{I_y - I_z}{I_x}\dot{\beta}\dot{\gamma} + \frac{lU_2}{I_x} \\ \ddot{\beta} = \frac{I_z - I_x}{I_y}\dot{\alpha}\dot{\gamma} + \frac{lU_3}{I_y} \\ \ddot{\gamma} = \frac{I_x - I_y}{I_z}\dot{\alpha}\dot{\beta} + \frac{lU_4}{I_z} \end{cases} \quad (1)$$

式中: I_x 、 I_y 和 I_z 分别为飞行器绕 x 、 y 、 z 三轴的惯性张量; m 为飞行器质量; U_1 、 U_2 、 U_3 和 U_4 分别为飞行器高度、俯仰角、滚动角和航向角的系统控制量; g 为重力

加速度; l 表示四旋翼飞机质心与机翼中心点的距离。定义飞行器四个旋翼的转速分别为 Ω_1 、 Ω_2 、 Ω_3 和 Ω_4 。因此系统控制量和四个旋翼转速之间关系的数学表达式如下:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(\Omega_1^2 - \Omega_3^2) \\ U_3 = b(\Omega_2^2 - \Omega_4^2) \\ U_4 = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{cases} \quad (2)$$

式中: b 为升力系数; d 为反扭矩系数。

由式(1)和式(2)可知,飞行器中,航向角、俯仰角和滚动角之间相互耦合,自抗扰控制器可将三通道之间的耦合量作为控制系统中的内部扰动,并与外部与扰动一起作为观测器观测的总扰动,对总扰动进行观测和补偿。因此可将式(1)和式(2)改写成与自抗扰控制数学模型的相对形式:

$$\begin{cases} \dot{z} = f_1(z, \dot{z}) + b_1 U_1 + \omega_1 - g \\ \ddot{\alpha} = f_2(\alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma}) + b_2 U_2 + \omega_2 \\ \ddot{\beta} = f_3(\alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma}) + b_3 U_3 + \omega_3 \\ \ddot{\gamma} = f_4(\alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma}) + b_4 U_4 + \omega_4 \end{cases} \quad (3)$$

式中: $\omega_i = (\omega_1, \omega_2, \omega_3, \omega_4)$ 为系统四通道所受外负载扰动; f_i 为系统内部扰动; $b_1 \approx m^{-1}$, $b_2 \approx I_x^{-1}$, $b_3 \approx I_y^{-1}$, $b_4 \approx I_z^{-1}$ 。

1.2 自抗扰控制器数学模型

自抗扰控制器是一类不基于精确数学模型且抗扰动能力强的非线性控制器^[10]。控制器由微分跟踪器、扩张状态观测器和非线性反馈控制率三部分构成。以航向角为例,设计航向角自抗扰控制器如下。被控对象可表示为:

$$\begin{cases} \ddot{\gamma} = f(\alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma}) + \omega + bu \\ \dot{y} = \gamma \end{cases} \quad (4)$$

1) 微分跟踪器的数学表达式如下:

$$\begin{cases} v_1(k+1) = v_1(k) + hv_2(k) \\ v_2(k+1) = v_2(k) + h \times fst(v_1(k) - \gamma_d, v_2(k), r, h_0) \end{cases} \quad (5)$$

式中: γ_d 为航向角期望输入; h_0 为滤波因子; h 为步长; r 为速度因子,调节微分跟踪器对输入信号的跟踪速度; $fst()$ 为最速综合函数,安排系统的过渡过程。

2) 扩张状态观测器的数学表达式如下:

$$\begin{cases} e_1 = z_1(k) - y(k) \\ z_1(k+1) = z_1(k) + h(z_2(k) - \beta_1 e_1) \\ z_2(k+1) = z_2(k) + h(z_3(k) - \beta_2 fal(e_1, \alpha_1, \delta) + bu(k)) \\ z_3(k+1) = z_3(k) - h\beta_3 fal(e_1, \alpha_2, \delta) \end{cases} \quad (6)$$

式中: β_1 、 β_2 和 β_3 为反馈增益参数,决定扩张状态观测器对扰动的观测和补偿效果; z_1 和 z_2 为被动对象输入值和其微分信号的观测值; z_3 为系统总扰动观测值; α_1 通常取0.5; α_2 通常取0.25; δ 通常取0.1。 $fal()$ 函数的定义如下:

$$fal(e, \alpha, \delta) = \begin{cases} |e|^\alpha \text{sgn}(e) & |e| > \delta \\ e \times \delta^{\delta-1} & |e| \leq \delta \end{cases} \quad (7)$$

式中: e 表示系统反馈误差。

3) 非线性误差反馈控制率根据输入输出反馈所得误差和扰动观测值 z_3 ,计算被控对象的最终控制量 U_4 。其数学表达式如下所示:

$$\begin{cases} e_1 = v_1(k) - z_1(k) \\ e_2 = v_2(k) - z_2(k) \\ U_0(k) = \beta_{11} fal(e_1, \alpha_1, \delta) + \beta_{12} fal(e_2, \alpha_2, \delta) \\ U_4(k) = U_0(k) - \frac{z_3(k)}{b} \end{cases} \quad (8)$$

式中: β_{11} 和 β_{12} 为控制器增益参数, β_{11} 越大响应速度越快,但会引起系统震荡, β_{12} 越大,系统不易出现超调,但会导致系统调节时间过长; α_1 一般取0.5; α_2 一般取1.5; b 为补偿系数。因此,非线性自抗扰控制器中,待整定参数为 β_1 、 β_2 、 β_3 、 β_{11} 、 β_{12} 和 b 。

本文将改进的蜻蜓算法在线整定自抗扰控制参数,并将优化的自抗扰控制器对四旋翼飞行器进行姿态控制和高度控制,进行实验。系统分为四个独立回路通道,分别控制高度、航向角、俯仰角和滚动角。设系统高度的期望输入为 z_d 、航向角期望输入为 γ_d 、俯仰角期望输入为 α_d 、滚动角期望输入为 β_d 。则基于改进蜻蜓算法的自抗扰四旋翼控制系统的控制结构如图1所示。

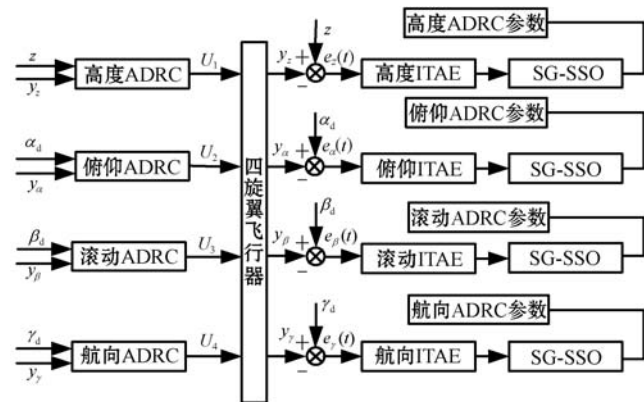


图1 控制结构

2 算法设计

2.1 基本蜻蜓优化算法

蜻蜓优化算法^[11]在寻优过程中会分为静态和动

态两个群体,其中静态群体在食物源附近进行小规模的范围性搜索,不断寻找其他猎物,直到找到最优食物源,这类行为被称为捕食行为。动态群体不断向最优食物源靠近,并沿当前寻优路径进行位置更新,寻找全局最优解,这类行为被称为迁徙行为。按照捕食和迁徙行为,蜻蜓算法主要通过以下五个方面进行迭代寻优。

1) 蜻蜓算法在寻优过程中,种群中的蜻蜓个体会不断与相邻个体进行分离,其数学表达式如下:

$$S_i = - \sum_{j=1}^N (x_i - x_j) \quad (9)$$

式中: S_i 为第 i 个个体与其相邻个体的分离度; N 为相邻个体的数量; $i=1,2,\dots,NP$, NP 为种群规模; x_i 为蜻蜓个体的位置; x_j 为相邻个体的位置, $j=1,2,\dots,N$ 。

2) 蜻蜓个体在寻优过程中,位置会不断发生改变,并与相近个体进行速度匹配,防止蜻蜓个体脱离群体,其数学表达式如下:

$$A_i = \frac{\sum_{j=1}^N v_j}{N} \quad (10)$$

式中: A_i 为第 i 个个体与相邻个体的速度对齐度; v_j 为相邻个体的速度。

3) 蜻蜓种群在寻优过程中,会形成一个小集体在寻优路径上进行小范围寻优,寻优完成后,小集体中的全部个体会向种群聚拢,其数学表达式如下:

$$C_i = \frac{\sum_{j=1}^N x_j}{N} - x_i \quad (11)$$

式中: C_i 为第 i 个个体与其相邻个体的聚拢度。

4) 蜻蜓个体会朝最优解方向不断靠近,并在路径中寻找更优质的解,其数学表达式如下:

$$F_i = x_{\text{best}} - x_i \quad (12)$$

式中: x_{best} 为食物源,即当前迭代产生的最优解; F_i 为食物源对第 i 个个体的吸引力。

5) 蜻蜓个体在捕猎过程中,会不断躲避天敌,这种行为的数学表达式如下:

$$E_i = x_{\text{worst}} - x_i \quad (13)$$

式中: x_{worst} 为蜻蜓个体的天敌,即当前迭代中的最差解; E_i 为第 i 个个体对天敌的排斥力。

综合以上五种蜻蜓在捕猎时的行为,定义在蜻蜓算法中,蜻蜓个体的位置更新公式如下:

$$X(t+1) = X(t) + \Delta X(t+1) \quad (14)$$

式中: $t=1,2,\dots,T_{\text{max}}$ 为当前迭代次数, T_{max} 为最大迭代次数; $X(t)$ 为蜻蜓种群中个体的位置; ΔX 表示蜻蜓个体的更新步长。 ΔX 更新公式如下:

$$\Delta X(t+1) = \alpha_s S_i + \alpha_A A_i + \alpha_C C_i + \alpha_F F_i + \alpha_E E_i + \omega \Delta X(t) \quad (15)$$

式中: α_s 为个体的分离权重; α_A 为速度匹配权重; α_C 为聚拢权重; α_F 为食物因子; α_E 为天敌因子; ω 为步长更新权重。

2.2 动态混沌权重

蜻蜓算法在位置更新过程中,步长更新权重 ω 会随机更改搜索步长,其优点在于搜索过程中不会丧失种群多样性,但缺点在于随机搜索极大概率会破坏算法全局收敛和局部搜索的平衡性。根据大量的实验研究表明,算法在迭代前期,应具有较大的搜索步长,扩大种群的搜索范围,保证全局最优解可以包含在解空间范围内,提高算法的全局搜索能力。同时大范围的搜索会使得算法的收敛精度降低,因此算法在迭代后期,应缩短搜索步长,减小搜索范围,使得算法可以在当前最优解附近进行精确搜索,提高算法的局部收敛精度。因此本文通过提出一种动态混沌步长更新权重对蜻蜓算法的位置更新方式进行改进。改进后的步长更新权重如下:

$$\omega_c(t+1) = 4\omega_c(t) \times (1 - \omega_c(t)) \times e^{\frac{T_{\max}}{T_{\max}-t}} \quad (16)$$

本文通过混沌映射所具有的遍历性和随机性,调整蜻蜓个体的搜索步长,使得蜻蜓个体可以在最优解附近进行精确搜索,同时通过指数函数 $e^{\frac{T_{\max}}{T_{\max}-t}}$ 动态调整步长的更新权重,使得算法在迭代前期具有较大的混沌权重,扩大算法的搜索范围,提高算法的全局搜索能力。在迭代后期具有较小的混沌权重,缩短蜻蜓个体的搜索范围,在当前最优解附近进行精确搜索,提高算法的收敛精度和局部搜索能力。更改后的蜻蜓步长更新公式如下:

$$\Delta X(t+1) = (\alpha_s S_i + \alpha_A A_i + \alpha_C C_i + \alpha_F F_i + \alpha_E E_i) + \omega_c \Delta X(t) \quad (17)$$

2.3 最差粒子位置遗弃

在蜻蜓个体寻优过程中,由于局部极值对其附近个体具有较强的吸引力,动态群体中的全部个体会向局部极值点不断靠近,使得个体无法跳出当前位置,导致算法早熟收敛,陷入局部最优,降低算法的收敛精度。传统的变异策略,通过使陷入局部最优的个体获得一个较大的扰动力,帮助个体跳出局部最优,但不能保证跳出局部最优的个体不会再次受到局部极值的吸引,重新陷入局部最优。因此本文引入一种新的最差个体位置遗弃策略,对算法中适应度值较差的群体进行改进。最差个体位置遗弃是指,若当前个体在同一个位置被搜索 L_{\max} 次,且适应度值并没有得到有效的改善,则认为该个体在当前位置陷入局部最优,算法在之后的若干次迭代过程中,不会对当前位置进行搜

索,此位置被标记为遗弃位置,并被生成的新位置所代替,新位置生成的数学表达式如下:

$$X_{\text{new}} = (1 - \eta(t)) X_{\text{best}}(t) + \lambda (X_{\text{best}}(t) - X(t)) \quad (18)$$

式中: X_{new} 为遗弃位置后产生的新位置; $\lambda = 2 - (\text{rand} - 0.5) \times e^{-0.35t}$ 为位置因子,是一个随机数; $X_{\text{best}}(t)$ 为当前迭代计算所得最优个体的位置; $\eta(t)$ 表示第 t 次迭代时位置更新权重。 $\eta(t)$ 计算公式如下:

$$\eta(t) = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \times e^{-0.35t} \quad (19)$$

式中: η_{\max} 和 η_{\min} 为位置更新权重的最大值和最小值。

2.4 多种群寻优

本文通过引入最差粒子位置遗弃策略的基本蜻蜓算法进行改进,可以有效避免个体陷入局部最优,早熟收敛。但新位置生成的条件,是指个体在一个位置被多次搜索到,但适应度值并没有被改善。该策略的优点在于,若被搜索到的个体适应度值较差,帮助该个体跳出局部最优,避免算法早熟收敛陷入局部最优。缺点在于,若被搜索到的个体适应度值较优,但该位置同样被搜索多次,且未改变该个体的适应度函数值,说明该个体很有可能为全局最优解,若此时将该个体的位置进行遗弃,则会导致算法跳出极值点,难以找到全局最优解。因此本文通过多种群策略对算法进行改进,双种群寻优过程如下:

Step1 种群初始化后,计算全部种群的适应度函数值,并进行排序。选择前 10% 的个体作为一个子种群,记为 S_1 。其余个体作为另一个子种群,记为 S_2 。

Step2 在每次迭代过程中,通过动态混沌惯性权重对子种群 S_1 中的个体进行位置更新,通过最差粒子位置遗弃策略对子种群 S_2 中的个体进行位置更新。

Step3 计算子种群 S_1 和 S_2 中的全部个体的适应度函数值。将种群 S_2 中的个体按照适应度函数值进行排序,并比较子种群 S_2 中前 10% 的个体与子种群 S_1 中个体的适应度函数值,若优于 S_1 中的个体,则进行替换。

2.5 改进蜻蜓算法的寻优步骤

Step1 蜻蜓种群初始化,包括种群规模 NP 、维数为 D 、 η_{\max} 、 η_{\min} 、 L_{\max} 、 T_{\max} 、种群中全部个体的位置 X_i 和初始步长 ΔX 。

Step2 计算蜻蜓种群中全部个体的适应度函数值并进行排序,选取前 10% 的个体作为一个子种群,记为 S_1 。其余个体作为另一个子种群,记为 S_2 。

Step3 通过式(9) - 式(13)计算 S_i 、 A_i 、 C_i 、 F_i 和 E_i 。

Step4 通过式(10)、式(11)和式(12)对子种群

S_1 中个体的位置和补偿进行更新。

Step5 通过式(10)、式(11)和式(13)对子种群 S_2 中个体的位置和补偿进行更新,判断是否进行位置遗弃并产生新位置。

Step6 计算子种群 S_1 和 S_2 中的全部个体的适应度函数值,并比较子种群 S_2 中前 10%的个体与子种群 S_1 中个体的适应度函数值,若优于 S_1 中的个体,则进行替换。

Step7 判断是否达到最大迭代次数 T_{max} ,是则输出最优解,否则返回 Step3。

3 算法仿真实验

为了验证本文所提多种群蜻蜓算法的性能,选取 10 个国际通用的标准测试函数进行实验验证。其中 F1 - F5 为单峰测试函数,主要验证算法的收敛速度和全局搜索能力。F6 - F10 为多峰测试函数,主要验证算法的局部收敛能力和收敛精度。其中测试函数的具体信息如表 1 所示。为了可以更加直观地表现出多种群蜻蜓算法的寻优性能,本文将多种群蜻蜓算法的实验结果与双空间粒子群算法^[7]、改进樽海鞘群算法^[12]、交叉保优粒子群算法^[6]的实验结果进行对比,为了保证实验的公平性,四种算法独立运行 50 次并对全部实验结果去平均值(Ave)、最小值(Min)、标准差(Std)、寻优成功率(SR),种群规模均为 100,迭代次数均为 100。具体实验结果如表 2 所示,其中最优解用加粗字体表示。四种算法的参数设置如下:其中本文所提多种群蜻蜓算法的位置更新权重的最大值 η_{max} 、最小值 η_{min} 和最大搜索次数 L_{max} 通过多次数值仿真实验所得最优取值。

1) 多种群蜻蜓算法: $\eta_{min}=0.25, \eta_{max}=0.9, L_{max}=5$ 。

2) 双空间粒子群算法:加速度因子 $c_1=c_2=2$,惯性权重 $\omega=0.6$,搜索范围因子 $r_1=0.75$,搜索范围因子 $r_2=0.25$ 。

3) 交叉保优粒子群算法:加速度因子 $c_1=2$,加速度因子 $c_2=2$,惯性权重最小值 $\omega_{min}=0.4$,惯性权重最大值 $\omega_{max}=0.9$ 。

表 1 测试函数

函数	函数名称	求解区间	维数	理论极值
F1	Sphere	[-100,100]	10	0
F2	Step	[-100,100]	20	0
F3	Schwefel2.21	[-100,100]	50	0
F4	Schwefel1.2	[-100,100]	50	0
F5	Rosenbrock	[-30,30]	100	0

续表 1

函数	函数名称	求解区间	维数	理论极值
F6	Ackley	[-32,32]	50	0
F7	Rastrigin	[-5.12,5.12]	10	0
F8	Griewank	[-600,600]	100	0
F9	Penalized1	[-50,50]	100	0
F10	Penalized2	[-50,50]	200	0

表 2 测试结果

函数	类别	多种群蜻蜓算法	双空间粒子群算法	交叉保优粒子群算法	改进樽海鞘群算法
F1	Min	3.79E-39	1.07E-34	5.86E-28	2.45E-29
	Ave	4.94E-33	9.29E-30	2.72E-25	2.83E-25
	Std	4.48E-33	8.71E-30	2.05E-25	2.03E-25
	SR/%	100	100	100	100
F2	Min	3.48E-36	3.49E-32	3.51E-00	3.52E-00
	Ave	6.72E-30	9.42E-28	7.13E-00	4.84E-00
	Std	1.31E-30	6.75E-28	1.19E-01	4.63E-01
	SR/%	100	100	0	0
F3	Min	3.52E-23	5.33E-17	3.55E-07	5.36E-11
	Ave	4.84E-18	7.55E-13	8.06E-03	2.97E-05
	Std	2.63E-18	5.06E-13	5.03E-03	1.94E-05
	SR/%	100	100	76	53
F4	Min	3.57E-34	8.53E-33	6.10E-21	3.61E-26
	Ave	5.68E-30	3.88E-28	8.09E-17	8.13E-21
	Std	4.03E-30	1.08E-28	1.25E-17	2.75E-21
	SR/%	100	100	100	100
F5	Min	2.63E-27	2.92E-13	5.36E+01	3.64E+03
	Ave	6.51E-24	2.56E-09	1.93E+02	4.39E+04
	Std	4.12E-24	1.63E-09	1.02E-01	3.44E-01
	SR/%	100	88	0	0
F6	Min	7.87E-17	3.36E-15	7.43E-11	7.41E-11
	Ave	7.33E-14	9.03E-12	2.64E-07	5.49E-08
	Std	6.35E-14	6.31E-12	2.03E-07	3.19E-08
	SR/%	100	100	100	100
F7	Min	0	0	7.89E-26	4.26E-26
	Ave	0	0	5.24E-22	5.29E-23
	Std	0	0	5.01E-22	4.92E-23
	SR/%	100	100	100	100
F8	Min	4.77E-19	953E-17	3.61E-05	9.20E+01
	Ave	8.25E-16	2.27E-4	5.43E-02	1.96E+01
	Std	7.33E-16	2.19E-14	2.39E-02	1.06E-01
	SR/%	100	95	46	0

续表 2

函数	类别	多种群蜻蜓算法	双空间粒子群算法	交叉保优粒子群算法	改进樽海鞘群算法
F9	Min	5.49E-37	7.22E-31	9.19E-04	6.38E-08
	Ave	6.37E-33	9.06E-28	5.59E-01	5.40E-03
	Std	6.02E-33	8.38E-28	5.24E-01	5.16E-03
	SR/%	100	100	57	89
F10	Min	5.07E-28	7.41E-31	5.79E-14	7.49E-08
	Ave	9.61E-25	6.97E-28	6.39E-11	4.58E-05
	Std	8.27E-25	6.49E-28	6.15E-11	4.06E-05
	SR/%	100	100	100	82

从表 2 中可知,对于单峰测试函数而言,本文所提多种群蜻蜓优化算法求解的平均值和最小值最小,说明多种群蜻蜓优化算法的全局收敛精度和寻优能力要优于其他三种算法。同时对于单峰测试函数 F2 和 F5 而言,交叉保优粒子群算法和改进樽海鞘群算法均早熟收敛,无法找到全局最优解,但多种群蜻蜓算法仍可找到全局最优解,说明动态混沌权重在寻优初期给予算法较大的搜索步长,扩大算法的寻优范围并成功覆盖了全局极值点。此外随求解维数增加,算法求解难度也急剧增加,双空间粒子群算法、交叉保优粒子群算法和改进樽海鞘群算法均很大程度地降低了算法的须有精度,寻优成功率也不同程度地降低,但多种群蜻蜓算法的寻优成功率均为 100% 且收敛精度变化不大,说明动态混沌权重在迭代后期有效缩短搜索步长,使得算法可以小范围精确搜索,提高了算法的收敛精度。

对于多峰测试函数 F6 - F9 而言,本文所提多种群蜻蜓算法求解的平均值和最小值相较其他三种算法而言,同样最小,仅在测试函数 F10 上,双空间粒子群算法的求解精度要略优于本文算法,说明本文所提多种群蜻蜓算法的局部寻优能力要优于其他三种算法。同样,随求解维数的增加,四种算法的局部搜索能力均有所下降,但本文算法的局部寻优精度变化不大,且成功率为 100%。值得注意的是,对于测试函数 F7 而言,本文算法可以求解到理论最优值,说明本文算法在迭代后期通过在极值点附近进行小范围搜索可以有效提高算法的寻优精度。

同时,对于全部测试函数而言,本文所提多种群蜻蜓算法求解的标准差较其他三种算法而言同样最小,说明本文算法的寻优稳定性要强于其他三种算法。随求解维数的增加,双空间粒子群算法、交叉保优粒子群算法和改进樽海鞘群算法在求解稳定性上表现的能力较差,但本文所提多种群蜻蜓算法求解的标准差变化不大,说明多种群蜻蜓算法对于求解具有复杂非线性数学模型的问题上,具有较强的寻优精度和寻优稳定性。

4 自抗扰四旋翼控制实验

本文通过多种群蜻蜓算法优化后的自抗扰控制器对飞行器进行姿态控制和高度控制实验,其中飞行器参数如表 3 所示。优化后的自抗扰控制器参数如表 4 所示。

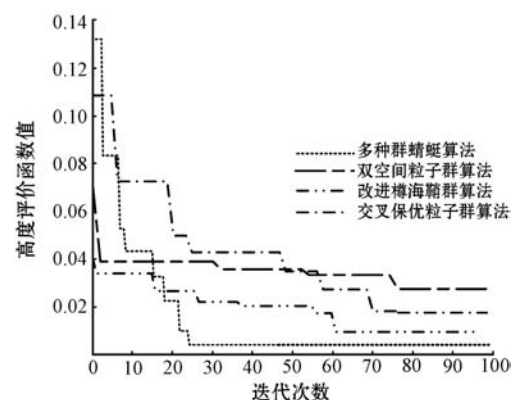
表 3 飞行器参数

参数	数值	参数	数值
惯性张量 $I_x / (\text{kg} \cdot \text{m}^2)$	0.023 9	飞行器质量 m/kg	1.5
惯性张量 $I_y / (\text{kg} \cdot \text{m}^2)$	0.023 9	旋翼到质心的距离 l/m	0.24
惯性张量 $I_z / (\text{kg} \cdot \text{m}^2)$	0.018 7	升力系数 $b / (\text{m} \cdot \text{s}^{-2})$	5.67E-05
重力加速度 $g / (\text{m} \cdot \text{s}^{-2})$	9.857	反扭矩系数 $d / (\text{N} \cdot \text{s}^2)$	1.1E-06

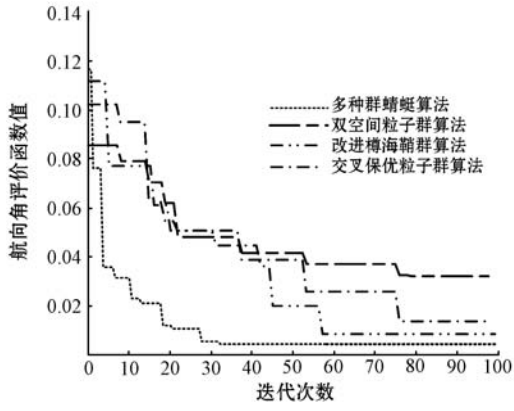
表 4 优化后的自抗扰控制器参数

高度 ADRC	航向 ADRC	俯仰 ADRC	滚动 ADRC
$\beta_1 = 98.35$	$\beta_1 = 80.55$	$\beta_1 = 95$	$\beta_1 = 95$
$\beta_2 = 298.55$	$\beta_2 = 284.35$	$\beta_2 = 355.64$	$\beta_2 = 355.64$
$\beta_3 = 2\ 500.75$	$\beta_3 = 2\ 200$	$\beta_3 = 2\ 300$	$\beta_3 = 2\ 300$
$\beta_{11} = 95.35$	$\beta_{11} = 85$	$\beta_{11} = 35.45$	$\beta_{11} = 25.45$
$\beta_{12} = 80.45$	$\beta_{12} = 80.45$	$\beta_{12} = 45.55$	$\beta_{12} = 40.35$
$b = 5$	$b = 5$	$b = 5$	$b = 5$

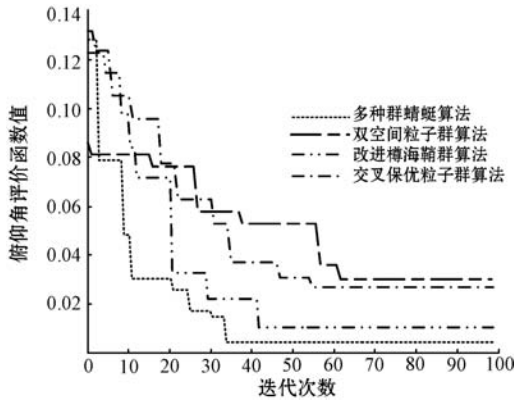
本文选择 ITAE 评价指标作为多种群蜻蜓算法的适应度函数,该函数可较好地抑制长时间存在的误差,是一类具有很好工程实用性和选择性的控制系统性能评价指标。该指标的输入为系统期望值和系统输出值之间的误差。本文将基于多种群蜻蜓算法求解所得的 ITAE 评价函数指标值与双空间粒子群算法、改进樽海鞘群算法、交叉保优粒子群算法求解取得的 ITAE 评价函数指标值进行对比实验,实验结果如图 2 所示。



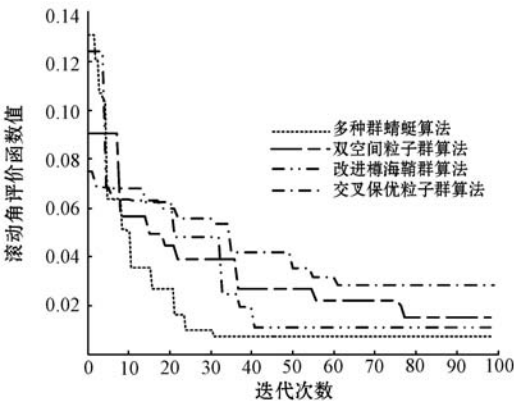
(a) 高度评价函数值



(b) 航向角评价函数值



(c) 俯仰角评价函数值



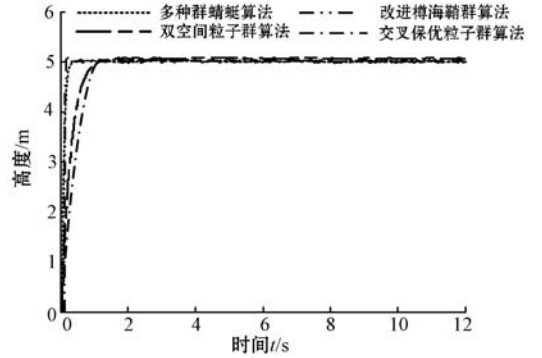
(d) 滚动角评价函数值

图2 评价函数对比

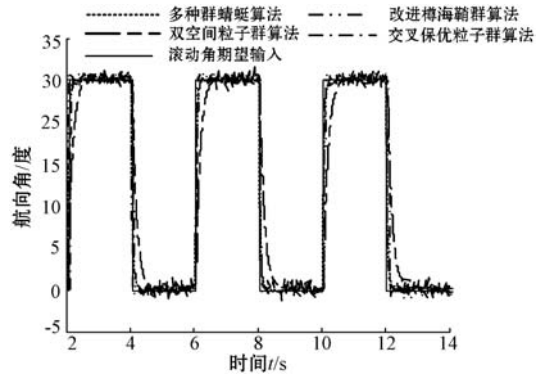
可以看出,在迭代初期,本文所提多种群蜻蜓算法求解的评价函数值下降趋势最快且范围最大,说明本文算法相较于其他三种算法在迭代前期具有较大的搜索范围,全局搜索能力更强。评价函数值下降速度最快说明本文算法的搜索速度最快。此外,在高度 ITAE 值、航向 ITAE 值、俯仰 ITAE 值和滚动 ITAE 值的求解上,多种群蜻蜓算法分别在 23 代、34 代、33 代和 33 代完成收敛,且 ITAE 值相较于其他三种算法均为最小,到达稳态时间最短。说明本文算法在线整定自抗扰参数所需时间最短且精度最高。

本文通过优化后的自抗扰控制器对飞行器进行姿态控制和高度控制,实验结果如图 3 所示。其中高度

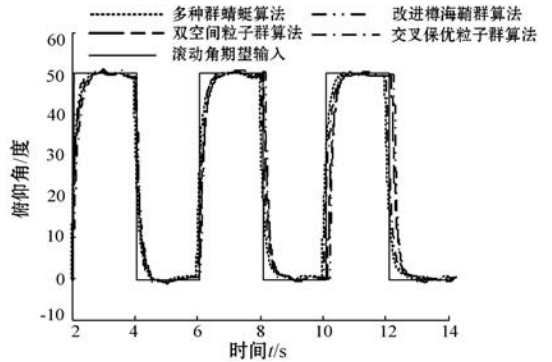
输入为 5 m,即幅值为 5 的阶跃信号。航线角期望输入为方波增益为 15 度的周期性方波,滚动角和俯仰角的期望输入为方波增益为 40 度的周期性方波。在实验过程中,为了对比验证本文所提改进策略的抗扰动能力,在控制过程中加入白噪声信号模拟外界扰动。



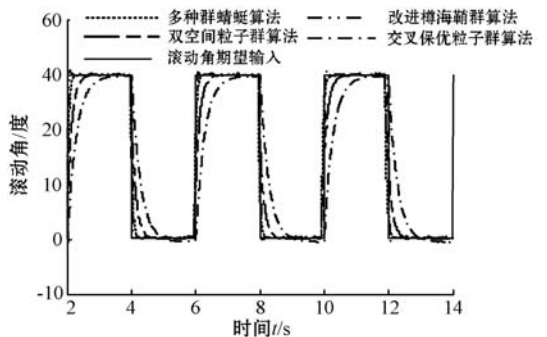
(a) 高度控制



(b) 航向角控制



(c) 俯仰角控制



(d) 滚动角控制

图3 系统控制对比

[D]. 杭州:浙江工业大学.

- [13] Chiurtu N, Rimoldi B, Telatar E. On the capacity of multi-antenna Gaussian channels [C]//Proceedings of 2001 IEEE International Symposium on Information Theory. IEEE, 2001.
- [14] Zeng K. Physical layer key generation in wireless networks: Challenges and opportunities [J]. IEEE Communications Magazine, 2015, 53(6):33-39.
- [15] 李佳宸. 基于深度学习的数字调制信号识别方法研究 [D]. 西安:西安电子科技大学.
- [16] Sklar B. Rayleigh fading channels in mobile digital communication systems part I: Characterization [J]. IEEE Communications Magazine, 1997, 35(9):136-146.
- [17] Hinton G E. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313(5786):504-507.
- [18] Alhasanat M, Althunibat S, Darabkh K A, et al. A physical-layer key distribution mechanism for IoT networks [J]. Mobile Networks and Applications, 2020, 25: 173-178.
- [19] 吴彦伦. 基于深度学习的调制识别算法研究 [D]. 杭州:电子科技大学, 2018.
- [20] Tusha A, Dogan S, Arslan H. IQI mitigation for narrowband IoT systems with OFDM-IM [J]. IEEE Access, 2018, 6: 44626-44634.

(上接第 334 页)

可以看出,首先,本文所提改进策略在高度控制和欧拉角控制上,取得的控制精度最高,最大相对误差分别为 0.2%、0.8%、0.93% 和 0.84%,且可以实现快速无超调。虽然基于改进樽海鞘群算法的优化控制策略在高度控制上,到达指定高度的时间略快于本文所提多种群蜻蜓算法的优化控制策略,但存在较大超调,且再次到达稳态的时间要慢于本文所提优化策略。此外,在航向角、俯仰角和滚动角的控制上,本文所提优化策略很好地对期望输入信号进行了跟踪,三个角之间的误差相差不大,说明优化后的自抗扰控制器很好地对三个欧拉角之间的耦合关系进行了解耦,较大程度地提高了控制精度,减小了飞行器的姿态控制误差。其次,加入白噪声后,本文所提多种群蜻蜓算法的自抗扰优化控制策略受扰动影响最小,三个欧拉角受扰动后,最大相对误差为 0.93%、1.1% 和 0.96%,远优于其他三种优化控制策略受到扰动后的控制误差,说明多种群蜻蜓算法优化后的自抗扰控制器相较其他三种优化控制策略,具有较高的抗扰动能力。

5 结 语

本文针对四旋翼飞行器姿态角难以控制的问题,

提出一种多种群蜻蜓算法自抗扰控制策略。对控制器参数难以整定问题,首先通过动态混沌权重、最差粒子遗弃和多种群策略对基本蜻蜓算法进行改进,提高算法的寻优精度,平衡算法全局局部收敛能力。其次将改进后的蜻蜓算法在线优化自抗扰控制器参数。最后将优化后的自抗扰控制器对四旋翼飞行器进行高度控制和姿态控制,在控制过程中加入白噪声扰动。实验结果表明,多种群蜻蜓算法优化后的自抗扰控制策略有效提高了飞行器的控制精度和抗扰动能力。

参 考 文 献

- [1] Ma H J, Liu Y, Li T, et al. Nonlinear high-gain observer-based diagnosis and compensation for actuator and sensor faults in a quadrotor unmanned aerial vehicle [J]. IEEE Transactions on Industrial Informatics, 2019, 15(1):550-562.
- [2] 赵超轮,赵国荣,戴邵武,等. 基于改进型 PID 算法的四旋翼控制研究 [J]. 计算机仿真, 2020, 37(11):48-53, 103.
- [3] 杨立本,章卫国,黄得刚. 基于 ADRC 姿态解耦的四旋翼飞行器鲁棒轨迹跟踪 [J]. 北京航空航天大学学报, 2015, 41(6):1026-1033.
- [4] 王雪娆,孙长银,林晓波,等. 基于神经网络的无人机姿态自适应控制仿真 [J]. 计算机仿真, 2020, 37(3):37-41, 88.
- [5] 古训,郑亚利,陈雨青. 四旋翼飞行器自适应滑模控制设计 [J]. 控制工程, 2020, 27(1):138-142.
- [6] 胡文华,曹仁赢. 改进粒子群优化算法的四旋翼 ADRC 姿态控制 [J]. 电光与控制, 2019, 26(12):12-16, 27.
- [7] 陈林奇,李廷会. 基于双空间 PSO 算法的四旋翼无人机自抗扰控制器优化设计 [J]. 广西师范大学学报(自然科学版), 2019, 37(3):42-49.
- [8] Wang B, Yu X, Mu L, et al. Disturbance observer-based adaptive fault-tolerant control for a quadrotor helicopter subject to parametric uncertainties and external disturbances [J]. Mechanical Systems and Signal Processing, 2019, 120: 727-743.
- [9] 卢凯文,杨忠,许昌亮,等. 基于非线性分离的可倾转四旋翼 LQR 飞行控制研究 [J]. 南京信息工程大学学报(自然科学版), 2019, 11(4):390-397.
- [10] 李家豪,孙洪飞. 自抗扰控制技术的改进和应用 [J]. 厦门大学学报(自然科学版), 2018, 57(5):695-701.
- [11] Mirjalili S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems [J]. Neural Computing and Applications, 2016, 27(4):1053-1073.
- [12] 丁力,高振奇,虞青. 基于改进樽海鞘群算法的四旋翼飞行器姿态优化控制 [J]. 农业机械学报, 2019, 50(10):243-250.