

基于协作神经动力学优化的四旋翼轨迹规划

郑天昂

浙江师范大学数学科学学院, 浙江 金华

收稿日期: 2026年7月6日; 录用日期: 2026年7月29日; 发布日期: 2026年8月12日

摘要

针对四旋翼轨迹规划中的非凸约束与局部最优问题, 本文提出一种基于协作神经动力学优化的轨迹规划方法。利用微分平坦性将问题转化为样条参数优化, 并将动力学约束与飞行走廊统一建模为可微惩罚函数。采用分布式实现多节点协同优化, 并引入快速随机跳跃算法增强全局搜索能力。结果表明, 该方法可生成满足速度与加速度约束的平滑轨迹, 并有效降低目标函数, 验证了方法的有效性与可行性。

关键词

四旋翼, 轨迹规划, 神经动力学优化

Quadrotor Trajectory Planning Based on Collaborative Neurodynamic Optimization

Tian'ang Zheng

School of Mathematical Sciences, Zhejiang Normal University, Jinhua Zhejiang

Received: July 6, 2026; accepted: July 29, 2026; published: August 12, 2026

Abstract

To address the non-convex constraints and local optima problems in quadcopter trajectory planning, this paper proposes a trajectory planning method based on collaborative neurodynamic optimization. By utilizing differential flatness, the problem is transformed into a spline parameter optimization problem, and the dynamic constraints and flight corridor are uniformly modeled as differentiable penalty functions. A distributed implementation is used to achieve multi-node collaborative optimization, and a Fast Stochastic Jump algorithm is introduced to enhance global search capabilities. The results demonstrate that this method can generate smooth trajectories that satisfy velocity and acceleration constraints while effectively minimizing the objective function, thereby validating the method's effectiveness and feasibility.

Keywords

Quadrotor, Trajectory Planning, Neurodynamic Optimization

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

无人飞行器(Unmanned Aerial Vehicle, UAV), 专指通过远程操控或机载程控的不载人飞行器。传统意义下, 无人机常根据气动原理的不同而分为固定翼无人机、旋翼无人机、扑翼无人机和复合翼无人机四类。其中, 旋翼无人机——尤其是多旋翼类中的四旋翼无人机(本文简称四旋翼)——近年来已成为民用领域应用最广泛的机型。与其他无人机相比, 四旋翼具有更简单的物理结构和精确的动力学模型, 因此备受国内外研究界的广泛关注。

以往的无人机应用场景多在高空领域, 在这种场景下, 无人机的自主导航过程中只涉及任务导向的航迹规划, 规划的约束至多考虑可行空域, 通常不会遇到避障的问题, 传统定位系统足以满足飞行任务需求。然而, 随着现代应用场景的变化, 无人机更多在建筑群、森林甚至室内环境中作业。在这些环境下, 系统受多种遮蔽物影响, 定位稳定性和准确性大幅下降。且相较于高空障碍稀疏的情况, 低空飞行更需要对稠密的障碍物环境进行感知和规划, 从而保证其航行的安全性和时间效率。具备完全自主导航能力的多旋翼无人机通常能够借助各种各样的传感器才能实现在三维环境的实时感知、定位、建图和规划。

对于无人机的动态模型, Fliess 等提出了微分平坦特性[1], 在相关领域引起极大关注[2]-[4]。特性表明, 满足特定条件的动态系统可通过一个仅包含部分信息的平坦输出及其有限阶导数反推完整状态和输入变量。Mellinger 和 Kumar 用机体合推力和三维扭矩为输入, 验证了四旋翼的微分平坦性[5]。Watterson 和 Kumar 利用 Hopf 纤维分解四轴飞行器的姿态旋转变换, 从而在平坦映射的搜索中实现了奇点数的最小化[6]。本文将基于此特性建立四旋翼轨迹优化问题的基础模型, 并提出一种分布式的协作神经动力学优化(Collaborative Neurodynamic Optimization, CNO)方法加以求解。CNO 主要由两个层级组成, 下层级负责搜索解, 上层级则跳出局部最优以改进解。传统的上层级受收敛性影响, 需要采用的改进策略相当复杂, 而本文则针对无人机飞行的工程需求, 提出了一种名为快速随机跳跃的改进策略, 在收敛性需求与工程执行效率需求间实现新的权衡。

2. 准备工作

2.1. 微分平坦性

考虑标准动力系统 $\dot{x} = f(x) + g(x)u$, 其中状态变量 $x \in \mathbb{R}^n$, 输入 $u \in \mathbb{R}^m$, 函数 $f(\cdot): \mathbb{R}^n \mapsto \mathbb{R}^n$, $g(\cdot): \mathbb{R}^n \mapsto \mathbb{R}^{m \times n}$, 且 $\text{rank}(g) = m$ 。称该系统是微分平坦的[1], 如果存在 $z \in \mathbb{R}^n$, 使得 x 和 u 均可表示为 z 及其有限阶导数的函数:

$$(x, u) = \Psi(z, \dot{z}, \ddot{z}, \dots, z^{(s)}) \quad (1)$$

其中, s 为控制阶数, $\Psi(\cdot)$ 由 $f(\cdot)$ 和 $g(\cdot)$ 诱导。四旋翼的微分平坦性已得到充分研究, 平坦映射 $\Psi(\cdot)$ 的具体形式在[5]-[8]中有详细阐述。简而言之, 四旋翼通常有 12 个待控状态:

$$[p_x, p_y, p_z, \dot{p}_x, \dot{p}_y, \dot{p}_z, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \quad (2)$$

其中, $[p_x, p_y, p_z]^\top$ 为四旋翼质心位置, $[\dot{p}_x, \dot{p}_y, \dot{p}_z]^\top$ 为质心在世界坐标系 $\mathcal{W}$ 中的线速度。 $[\phi, \theta, \psi]^\top$ 为欧拉角(三个分量分别为横滚、俯仰、偏航), $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^\top$ 为机体坐标系 $\mathcal{B}$ 相对于 $\mathcal{W}$ 的角速度。借助该系统的微分平坦性, 可以用平坦输出来简化控制的维数, 常用的平坦输出是

$$z = [p_x, p_y, p_z, \psi]^\top \quad (3)$$

得益于此良好性质, 我们可以首先在平坦输出空间中规划轨迹, 然后通过平坦映射将其映射到完整的状态-输入轨迹。本文的模型正是针对平坦输出轨迹 $z(t)$ 来构建, 飞行走廊、航点和动态限制等主要需求可通过可微惩罚函数纳入优化目标, 额外的用户定义的硬不等式约束则可显式保留。优化完成后再利用平坦映射 $\Psi(\cdot)$ 来恢复相应的状态-输入轨迹 $(x(t), u(t))$ 以供跟踪。

2.2. 经典神经动力学优化方案

协作式 CNO 由上下两层级组成。下层利用多个具有不同初始状态的循环神经网络(Recurrent Neural Network, RNN)进行散射搜索, 以寻找局部最优解[9]-[11]。当 RNN 达到局部收敛时, 上层采用元启发式规则对其进行重新初始化。下层级有界优化问题通常表述如下:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad (3a)$$

$$\text{s.t. } g(\mathbf{x}) \leq \mathbf{0} \quad (3b)$$

$$h(\mathbf{x}) = 0 \quad (3c)$$

其中, $g(\cdot)$ 被假设为二阶可微的, $\Omega = \prod_i [a_i, b_i]$ 是一个有界超盒。此类问题的最优解等价于一个变分不等式的解, 且该解被广泛认为满足卡鲁什-库恩-塔克(Karush-Kuhn-Tucker, KKT)条件:

$$\mathbf{x} - \mathcal{P}_\Omega(\mathbf{x} - \nabla f(\mathbf{x}) - \lambda^\top \nabla g(\mathbf{x})) = \mathbf{0} \quad (4a)$$

$$\lambda \geq \mathbf{0}, g(\mathbf{x}) \leq \mathbf{0}, \lambda^\top g(\mathbf{x}) = 0 \quad (4b)$$

$$h(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \Omega \quad (4c)$$

其中, $\mathcal{P}_\Omega(x) = \arg \inf_{y \in \Omega} \|x - y\|$ 为投影函数。现有研究为解决该问题而设计的投影神经网络如下:

$$\epsilon \dot{\mathbf{x}} = -\mathbf{x} + \mathcal{P}_\Omega(\mathbf{x} - \nabla f(\mathbf{x})) \quad (5a)$$

$$\epsilon \dot{\lambda} = -\lambda + (\lambda + g(\mathbf{x}))_+ \quad (5b)$$

$$\epsilon \dot{\mu} = h(\mathbf{x}) \quad (5c)$$

通常而言, 由于超盒约束, CNO 必然包含投影层, 从而增加计算负担; 但在四旋翼模型中, 飞行走廊本身对轨迹参数取值范围做出了比超盒更精细的限制, 所以在本文的 CNO 中, 投影层便可省去。

3. 问题建模

本文将平坦输出的轨迹由一条均匀样条曲线参数化, 并采用控制成本函数结合时间正则化项提出规划问题模型:

$$\min_{B, T} f_{\text{ctrl}}(B, T) + \xi_{\text{time}}(T) \quad (6a)$$

$$\text{s.t. } z(t) = z(t; B, T) \in \mathcal{C}, \forall t \in [0, T] \quad (6b)$$

$$z^{(j-1)}(t_i) = z_{ij}, i = 0, \dots, n_T; j = 1, \dots, s \quad (6c)$$

$$\mathcal{G}(\dot{z}(t), \dots, z^{(s)}(t)) \leq \mathbf{0}, \forall t \in [0, T] \quad (6d)$$

$$g_h(B, T) \leq \mathbf{0} \quad (6e)$$

$$T > 0 \quad (6f)$$

其中, B 为样条参数向量, $T > 0$ 为总飞行时间, $z(t; B, T)$ 是由 (B, T) 表征的平滑输出轨迹, f_{ctrl} 为二次控制成本, ξ_{time} 为时间正则项, 集合 $\mathcal{C}$ 为安全飞行走廊, t_i 是第 i 个时间节点, z_{ij} 是轨迹在 t_i 时刻的第 $(j-1)$ 阶导数的预设期望值, s 称为控制阶数, $\mathcal{G}(\cdot)$ 概括了连续时间动态极限约束, $g_h(B, T)$ 表示由用户定义的其他需求(取决于具体飞行场景)。

具体而言, 输出的样条轨迹表述为 $z(t; B, T) = \Phi(\tau)B$, 其中 $\Phi(\cdot)$ 是基函数矩阵, 可预先离线计算, 记优化变量 $x = [B^\top, T]^\top \in \mathbb{R}^{n \times m+1}$, 采用二次形式控制成本 f_{ctrl} 以确保运动平滑性[12], 并根据梯度进行归一化:

$$f_{\text{ctrl}}(x) = \int_0^T \|z^{(s)}(t)\|^2 dt = \frac{1}{T^{2s-1}} B^\top M B \quad (7a)$$

$$M = \int_0^1 \Phi^{(s)}(\tau)^\top \Phi^{(s)}(\tau) d\tau \quad (7b)$$

$$\frac{\partial f_{\text{ctrl}}}{\partial B} = \frac{2}{T^{2s-1}} M B \quad (8a)$$

$$\frac{\partial f_{\text{ctrl}}}{\partial T} = -\frac{2s-1}{T^{2s}} B^\top M B \quad (8b)$$

对称半正定矩阵 M 其也可以离线计算。于是, 目前可将目标函数改写为如下形式:

$$F(x) = f_{\text{ctrl}}(x) + \xi_{\text{time}}(T) + \sum_{r \in \mathcal{I}_p} \mu_r \xi_r(c_r(x)) \quad (9)$$

其中, 集合 $\mathcal{I}_p$ 是罚函数指标集, $c_r(x)$ 是 $r \in \mathcal{I}_p$ 对应的违反函数, $\xi_r(\cdot)$ 为罚函数, μ_r 为惩罚权重。

4. 约束处理

4.1. 时间正则项

时间惩罚主要有两类: 刚性惩罚和柔性惩罚。刚性惩罚将时间约束视为不可违背的要求, 任何违反限制的解均被视为不可行, 对应的惩罚函数值为无穷大, 尽管这种做法能保证强可行性, 但在实际应用中存在若干缺点: 其缺乏灵活性, 无法在轻微的约束违反与目标函数的改进之间取得平衡, 这可能会排除一些可接受的近可行解, 还会破坏目标函数的可微性。因此, 本文采用多项式形式的柔性惩罚:

$$\xi_{\text{time}} = \sum_{i=1}^{m_T} k_i (T - T_{\text{desire}})^i \quad (10a)$$

$$\frac{\partial \xi_{\text{time}}}{\partial T} = \sum i k_i (T - T_{\text{desire}})^{i-1} \quad (10b)$$

其中, $k_{m_T} > 0$ 。显然, 时间正则项是光滑的且梯度是利普希茨连续的。在实际工程中, 其通常选为线性项 $\xi_{\text{time}}(T) = kT$ 或二次项 $\xi_{\text{time}}(T) = k(T - T_{\text{desire}})^2$ 即可。

4.2. 飞行走廊

设 $\mathcal{C} \subset \mathbb{R}^3$ 表示位置空间中的走廊, 其由一系列 $\mathcal{H}$ -表示[13]的凸多面体近似表征:

$$\mathcal{C} = \bigcup_{i=1}^{n_C} \mathcal{C}_i \quad (11a)$$

$$\mathcal{C}_i = \{p \in \mathbb{R}^3 \mid A_i p \leq b_i\} \quad (11b)$$

走廊基元沿前端路径排列且相邻段之间有足够重叠，从而使连续轨迹能够在基元之间平滑过渡。理想的连续时间走廊约束为 $z(t; B, T) \in \mathcal{C}$ 。每个走廊基元分配相同数量的多项式段，设此数量为 k_C ，则索引映射 $j(k)$ 决定使用哪个多面体来评估第 k 个样本，采样索引 k 与走廊索引 j 间的关系如下：

$$j(k) = \left\lceil \frac{k+1}{k_C} \right\rceil \quad (12)$$

其中， $\lceil \cdot \rceil$ 表示高斯上取整。对任意节点 $t_i \sim \tau_i$ ，其位置 $p_i = \Phi(\tau_i)B$ ；令 $(x)_+ = \max\{x, 0\}$ 表示逐元素取正函数，则惩罚项及其梯度信息如下：

$$\xi_{\text{obs}}(x) = \sum_{k=0}^{n_s} \left\| \left(A_{j(k)} \Phi(\tau_k) B - b_{j(k)} \right)_+ \right\|^2 \quad (13)$$

$$\frac{\partial \xi_{\text{obs}}}{\partial B} = 2 \sum_{k=0}^{n_s} \left(A_{j(k)} \Phi(\tau_k) \right)^\top \left(A_{j(k)} \Phi(\tau_k) B - b_{j(k)} \right)_+ \quad (14)$$

本文的飞行走廊由快速迭代区域膨胀(Fast Iterative Region Inflation, FIRI)算法[14]生成。FIRI 以种子集和障碍点作为输入，并在下述两个模块间交替迭代。

- **限制性膨胀(Restrictive Inflation, RsI)**: 将椭球体膨胀至与障碍物接触，同时严格包含种子点，从而生成一个半空间。所有此类半空间的交集构成一个凸多面体。

- **最大内接椭球(Maximum Volume Inscribed Ellipsoid, MVIE)**: 计算多面体内部体积最大的椭球体，并将之作为 RsI 的下一个椭球体。内接椭球的体积单调递增，当相对改进率低于阈值 ρ 时，迭代终止。FIRI 可在毫秒级时间内保证可控性(包含种子)和高体积。

由 FIRI 生成的凸多面体表示为 H-结构，即有限个半空间的交集，因此可直接用于计算上一节中建模的惩罚函数。飞行走廊的结构和采样惩罚的设置如图 1 所示。

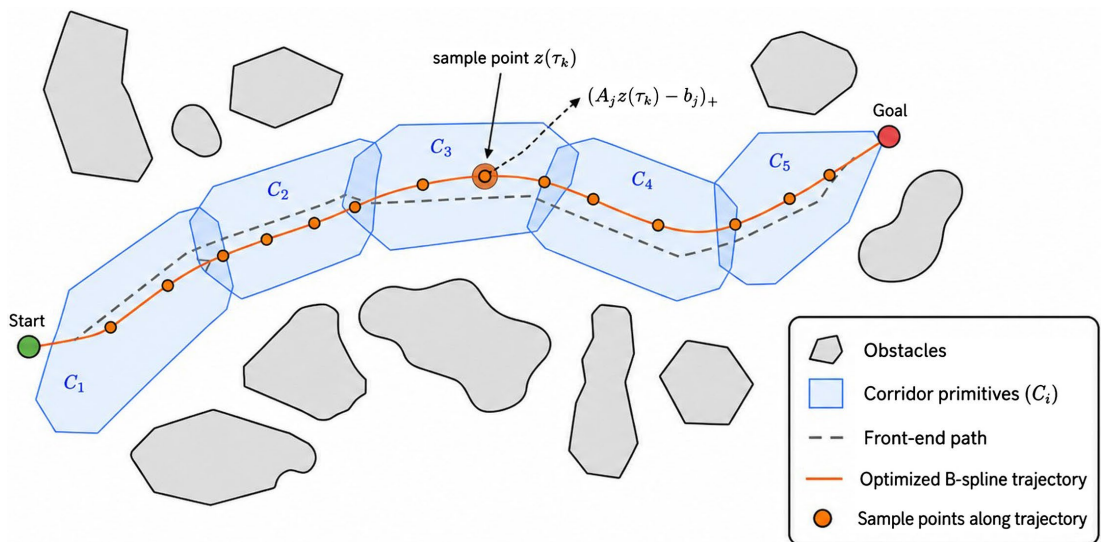


Figure 1. Illustration of flight corridors and sampled corridor penalties
图 1. 飞行走廊及采样惩罚示意图

4.3. 基本动力学约束

本节介绍两种常见的用户指定规划需求：路径点和基本动态限制。在本模型中，其被转换为可微惩罚项并纳入惩罚目标函数 $\tilde{F}(x)$ 。其他自定义约束则保留在 $g_h(x) \leq 0$ 中。路径点需求规定轨迹应在选定的时间点通过一组预定的位置，即 $z(t_i) = z_i (i = 1, \dots, n_w)$ ，此处采用二次损耗来定义位置约束惩罚：

$$\xi_{\text{way}}(x) = \sum_{i=1}^{n_w} \|\Phi(\tau_i)B - z_i\|^2 \quad (15)$$

其中， z_i 为时间 t_i 处的目标路径点，与 t_i 对应的标准化时间即上式中的 τ_i ， n_w 为总路径点个数。直观上看， $\xi_{\text{way}}(\cdot)$ 衡量的是轨迹在“感兴趣的点”处的位置偏差的平方和，它是关于 B 的二次型，并通过归一化时间 τ 非线性地依赖于 T ，只要 $\Phi(\cdot)$ 光滑，则 $\xi_{\text{way}}(\cdot)$ 也是光滑的，导出如下梯度信息：

$$\frac{\partial \xi_{\text{way}}}{\partial B} = 2 \sum_{i=1}^{n_w} \Phi(\tau_i)^\top (\Phi(\tau_i)B - z_i) \quad (16a)$$

$$\frac{\partial \xi_{\text{way}}}{\partial T} = -2 \sum_{i=1}^{n_w} (\Phi(\tau_i)B - z_i)^\top \Phi'(\tau_i)B \quad (16b)$$

事实上，路径点约束正是最初模型中 $z^{(j-1)}(t_i) = z_{ij} (i = 0, \dots, n_T; j = 1, \dots, s)$ 这一约束当 $j = 1$ 时的特殊情况；对于更一般的 $j > 1$ 的情况，同样的惩罚项构造也依然可行，但在工程上略显严格，因为此时约束的是速度、加速度、急推等动力学信息，在实际飞行中通常只需要限制其不超过某一上界即可。本文的实验依照最小急推准则(Minimum Jerk)，取控制阶数 $s = 3$ ，因此实际受到上述上界限制的动力学信息恰好正是四旋翼的速度和加速度，对应的需求即

$$\|\dot{z}(t)\| \leq v_{\max}, \quad \forall t \in [0, T] \quad (17a)$$

$$\|\ddot{z}(t)\| \leq a_{\max}, \quad \forall t \in [0, T] \quad (17b)$$

分别定义两个标量不等式：

$$g_v(\tau_k, x) = \frac{1}{T} \|\Phi'(\tau_k)B\| - v_{\max} \geq 0 \quad (18a)$$

$$g_a(\tau_k, x) = \frac{1}{T^2} \|\Phi''(\tau_k)B\| - a_{\max} \geq 0 \quad (18b)$$

不等号左侧式的各梯度信息如下：

$$\frac{\partial g_v(\tau_k, x)}{\partial B} = \frac{\Phi'(\tau_k)^\top \Phi'(\tau_k)B}{T \|\Phi'(\tau_k)B\|} \quad (19a)$$

$$\frac{\partial g_v(\tau_k, x)}{\partial T} = -\frac{|\Phi'(\tau_k)B|}{T^2} \quad (19b)$$

$$\frac{\partial g_a(\tau_k, x)}{\partial B} = \frac{\Phi''(\tau_k)^\top \Phi''(\tau_k)B}{T^2 \|\Phi''(\tau_k)B\|} \quad (19c)$$

$$\frac{\partial g_a(\tau_k, x)}{\partial T} = -\frac{2 \|\Phi''(\tau_k)B\|}{T^3} \quad (19d)$$

矩阵 $\Phi'(\tau_k)$ 和 $\Phi''(\tau_k)$ 分别是样条基矩阵对归一化时间的一阶和二阶导数。因子 $1/T$ 和 $1/T^2$ 将归一化时间的导数转换为物理时间下的速度和加速度，于是总惩罚项即设置为：

$$\xi_{\text{dyn}}(x) = \mu_v \sum_{k=1}^{n_s} (g_v(\tau_k, x))_+^3 + \mu_a \sum_{k=1}^{n_s} (g_a(\tau_k, x))_+^3 \quad (20)$$

$$\frac{\partial \xi_{\text{dyn}}}{\partial B} = 3\mu_v \sum_{k=1}^{n_s} (g_v(\tau_k, x))_+^2 \frac{\partial g_v(\tau_k, x)}{\partial B} + 3\mu_a \sum_{k=1}^{n_s} (g_a(\tau_k, x))_+^2 \frac{\partial g_a(\tau_k, x)}{\partial B} \quad (21a)$$

$$\frac{\partial \xi_{\text{dyn}}}{\partial T} = 3\mu_v \sum_{k=1}^{n_s} (g_v(\tau_k, x))_+^2 \frac{\partial g_v(\tau_k, x)}{\partial T} + 3\mu_a \sum_{k=1}^{n_s} (g_a(\tau_k, x))_+^2 \frac{\partial g_a(\tau_k, x)}{\partial T} \quad (21b)$$

其中, 常数 $\mu_v > 0$ 和 $\mu_a > 0$ 分别是速度和加速度超限的惩罚权重。此式仅对正超限情况 $(\cdot)_+$ 进行惩罚; 次数取为 3 是旨在确保对超限情况响应的平滑性。

5. 神经动力学优化

5.1. 问题重建模

首先将以上基于惩罚的受约束重构写成以下集中式优化问题的形式:

$$\min_x \tilde{f}(x) = \sum_{i=1}^N \tilde{f}_i(x) \quad (22a)$$

$$\text{s.t. } g_h(x) \leq 0 \quad (22b)$$

该问题可以分布式求解, 为实现并行计算, 引入了决策变量的本地副本, 并在求解器节点间施加一致性共识约束:

$$\min_{\{x_i\}_{i=1}^N} \sum_{i=1}^N \tilde{f}_i(x_i) \quad (23a)$$

$$\text{s.t. } g_{h,i}(x_i) \leq 0, \quad i = 1, \dots, N \quad (23b)$$

$$x_i = x_j, \quad (i, j) \in \mathcal{E} \quad (23c)$$

其中, x_i 是节点 i 处决策变量的局部副本, $\tilde{f}_i(x_i)$ 是节点 i 处的局部惩罚目标函数, $g_{h,i}(x_i)$ 表示节点 i 处保留的局部硬不等式约束, 边集 $\mathcal{E}$ 表示节点间的通信拓扑, 等式 $x_i = x_j$ 迫使相邻节点间达成共识。设 $\mathcal{G}$ 表示 N 个求解节点的信息交换图, L_G 表示其加权拉普拉斯矩阵, 若图 $\mathcal{G}$ 无向连通, 则共识条件可等价表示为 $(L_G \otimes I)\mathbf{x} = \mathbf{0}$ [15]。受约束重构的拉格朗日函数定义为 $\mathcal{L}(\mathbf{x}, \lambda, \mu) = \tilde{f}(\mathbf{x}) + \lambda^\top g_h(\mathbf{x}) + \mu^\top L\mathbf{x}$, 相应的 KKT 条件如下:

$$\nabla \tilde{f}(\mathbf{x}^*) + \nabla g_h(\mathbf{x}^*)^\top \lambda^* + L^\top \mu^* = \mathbf{0} \quad (24a)$$

$$\lambda^* \geq \mathbf{0} \quad (24b)$$

$$g_h(\mathbf{x}^*) \leq \mathbf{0} \quad (24c)$$

$$(\lambda^*)^\top g_h(\mathbf{x}^*) = \mathbf{0} \quad (24d)$$

$$L\mathbf{x}^* = \mathbf{0} \quad (24e)$$

其中, $\mathbf{x}^*$ 为某个局部最优解, λ^* 为约束 $g_h(\mathbf{x}) \leq \mathbf{0}$ 相关乘子, μ^* 为共识等式约束 $L\mathbf{x} = \mathbf{0}$ 相关乘子, 互补条件也可通过关系 $\lambda = (\lambda + g_h)_+$ 表示。定义增广拉格朗日函数:

$$\bar{\mathcal{L}}(\mathbf{x}, \lambda, \mu) = \tilde{f}(\mathbf{x}) + \lambda^\top (\mathbf{x}) + \mu^\top L\mathbf{x} + \frac{\alpha}{2} g_h(\mathbf{x})^\top \text{diag}(\lambda^2) g_h(\mathbf{x}) \quad (25)$$

其中, $\text{diag}(\lambda^2)$ 表示以 λ 各分量平方为对角元的对角矩阵, 增广项梯度 $\alpha \nabla g_h(\mathbf{x})^\top \text{diag}(\lambda^2) g_h(\mathbf{x})$, 在实际实验中, α 可通过试错法或自适应策略来选择。该 CNO 系统可被解释为与扩充拉格朗日结构相关的原-对偶梯度流。在满足互补松弛条件的 KKT 点处, 增广项取值为零, 因此基于惩罚项的受限重构的 KKT 条件保持不变。

5.2. 分布网络组

为求解上述问题, 可以设计如下的网络组:

$$\epsilon \dot{\mathbf{x}} = -\nabla \tilde{f}(\mathbf{x}) - \nabla_h(\mathbf{x})^\top \lambda - L^\top \mu - \alpha \nabla g_h(\mathbf{x})^\top \text{diag}(\lambda^2) g_h(\mathbf{x}) \quad (26a)$$

$$\epsilon \dot{\lambda} = -\lambda + (\lambda + g_h(\mathbf{x}))_+ \quad (26b)$$

$$\epsilon \dot{\mu} = L\mathbf{x} \quad (26c)$$

其中, $\epsilon > 0$ 是系统的时标参数。其中, 对于第二式的乘子动力学, 非负象限具有正不变性; 具体而言, 若 $\lambda(0) \succeq \mathbf{0}$, 则对于所有 $t \geq 0$, 都有 $\lambda(t) \succeq \mathbf{0}$ 。这是因为在边界 $\lambda_i = 0$ 处, 由于时标参数的严格正性和初值的非负性, 相应的向量场总是满足 $\epsilon \dot{\lambda}_i = g_{h,i}(\mathbf{x}) \succeq \mathbf{0}$, 使得乘子轨迹无法越过此边界离开非负象限。

该系统的每个平衡点都满足基于惩罚的受约束重构形式的 KKT 条件。进一步地, 由于通信图无向连通且拉普拉斯矩阵稀疏, 每个节点只需使用自身的局部目标函数和硬约束, 以及与邻节点交换的变量, 即可计算拉普拉斯矩阵, 因此网络群可以完全以分布式的方式实现。各节点动力学方程如下:

$$\epsilon \dot{x}_i = -\nabla f_i(x_i) - \sum_{j \in \mathcal{N}_i} A_{ij}(\mu_i - \mu_j) - \nabla g_{h,i}(x_i)^\top \lambda_i - \alpha \nabla g_{h,i}(x_i)^\top \text{diag}(\lambda_i^2) g_{h,i}(x_i) \quad (27a)$$

$$\dot{\lambda}_i = -\lambda_i + (\lambda_i + g_{h,i}(x_i))_+ \quad (27b)$$

$$\epsilon \dot{\mu}_i = \sum_{j \in \mathcal{N}_i} A_{ij}(x_i - x_j) \quad (27c)$$

其中, $\mathcal{N}_i$ 表示节点 i 的邻接集; A_{ij} 是通信图邻接矩阵的第 (i, j) 个元素。设 $(\mathbf{x}^*, \lambda^*, \mu^*)$ 表示 CNO 系统的平衡点, 则

$$L\mathbf{x}^* = \mathbf{0} \quad (28a)$$

$$\lambda^* = (\lambda^* + g_h(\mathbf{x}^*))_+ \quad (28b)$$

$$\lambda^* \succeq \mathbf{0} \quad (28c)$$

$$g_h(\mathbf{x}^*) \leq \mathbf{0} \quad (28d)$$

$$(\lambda^*)^\top g_h(\mathbf{x}^*) = 0 \quad (28e)$$

$$\nabla \tilde{f}(\mathbf{x}^*) + \nabla g_h(\mathbf{x}^*)^\top \lambda^* + L^\top \mu^* = \mathbf{0} \quad (28f)$$

在平衡点处互补松弛条件下, 稳态方程中的增项为零, 因此平衡条件恰好归结为基于惩罚项的受限重构问题的 KKT 条件; 反之, 任何满足固定点乘子关系的重新表述问题的 KKT 点, 都是神经动力学系统的平衡点。因此, 神经动力学系统的平衡点与重新表述问题的 KKT 点之间存在一一对应关系。在适当的二阶充分条件下, 相应的原始变量是重新表述问题的严格局部极小值。

定理: 设 $(\mathbf{x}^*, \lambda^*, \mu^*)$ 是 CNO 系统的平衡点, 则 $(\mathbf{x}^*, \lambda^*, \mu^*)$ 满足基于惩罚项的受限重构问题的 KKT 条件。此外, 若二阶充分条件在相应的临界锥上成立, 则 $\mathbf{x}^*$ 是原问题的严格局部极小解。

证明: 在平衡状态下, $\dot{\mu} = \mathbf{0}$ 意味着 $L\mathbf{x}^* = \mathbf{0}$; $\dot{\lambda} = \mathbf{0}$ 意味着 $\lambda^* = (\lambda^* + g_h(\mathbf{x}^*))_+$, 这等价于 $\lambda^* \succeq \mathbf{0}$,

$g_h(\mathbf{x}^*) \leq \mathbf{0}$, $(\lambda^*)^\top g_h(\mathbf{x}^*) = 0$ 。最后, $\dot{\mathbf{x}} = \mathbf{0}$ 给出了驻点条件。由于在互补松弛条件下, 增广项为零, 因此稳态方程简化为普通的 KKT 稳态条件。因此, 每个平衡点都满足重新表述的问题的 KKT 条件。在二阶充分条件下, 相应的原问题变量是严格的局部极小值。后端轨迹优化结构如图 2 所示。

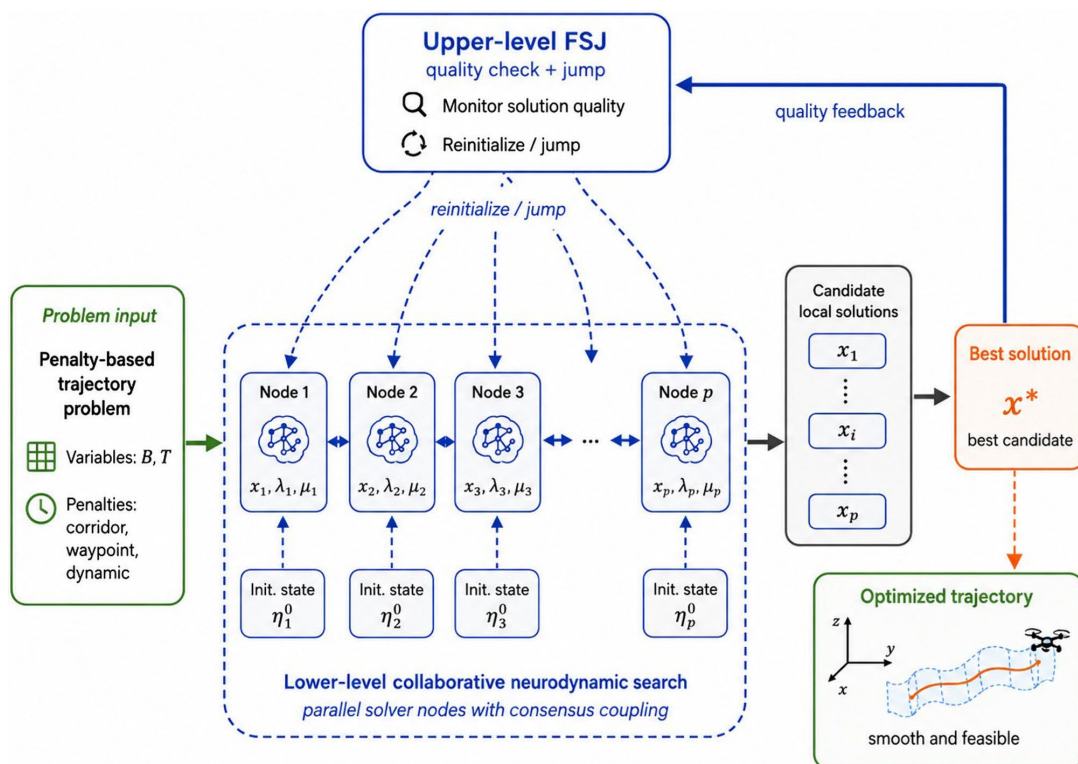


Figure 2. Structure of the back-end trajectory optimizer
图 2. 后端轨迹优化器结构

5.3. 上层级简化与改进

上层级旨在从指定候选点重初始化下层级求解器以改进解。传统基于粒子群的重初始化需要种群演化以及多个超参数, 带来额外的计算开销。鉴于四旋翼运动规划是有界问题, 可行域内必然有解, 且并不强求全局最优性, 摆脱局部最优的主要目标通常是改进性能差的解。改进后的轨迹未必全局最优, 只需工程可行即可。

基于上述性质, 本文提出一种名为快速随机跳跃(Fast Stochastic Jump, FSJ)的轻量算法以实现加速, 见表 1 与图 3。设 $\mathcal{X} = \{x_i\}_{i=1}^{n_x}$ 是包含于可行域的随机变量集。在每次迭代中, x_s 被更新为一个更优的点, 同时移除其他不满足此要求的点。当求解器收敛到较差的局部解后, FSJ 生成随机跳跃, 并选择一个惩罚目标值较低的可行候选解; 然后利用该选定的候选解重新初始化低级求解器, 以执行进一步的优化。

Table 1. Framework of FSJ algorithm

表 1. FSJ 算法框架

Algorithm: Fast Stochastic Jump

Input: set of random variables $\mathcal{X} = \{x_i\}_{i=1}^{n_x}$

Output: improved solution x^*

续表

1	begin
2	$x_s \leftarrow x_0;$
3	$\mathcal{Y} = \mathcal{X} \times f(\mathcal{X});$
4	repeat
5	$x^* \leftarrow \text{FindLocalMinimum}(x_s);$
6	foreach $(x_n, f(x_n)) \in \mathcal{Y}$ do
7	if $f(x_n) - f(x^*) < 0$ then
8	$x_s \leftarrow x_n;$
9	break;
10	end
11	end
12	$\mathcal{Y} \leftarrow \mathcal{Y} - \{(x_i, f(x_i)) \in \mathcal{Y} \mid f(x_i) - f(x^*) \geq 0\};$
13	until $\mathcal{Y} = \emptyset;$
14	return x^*
15	end

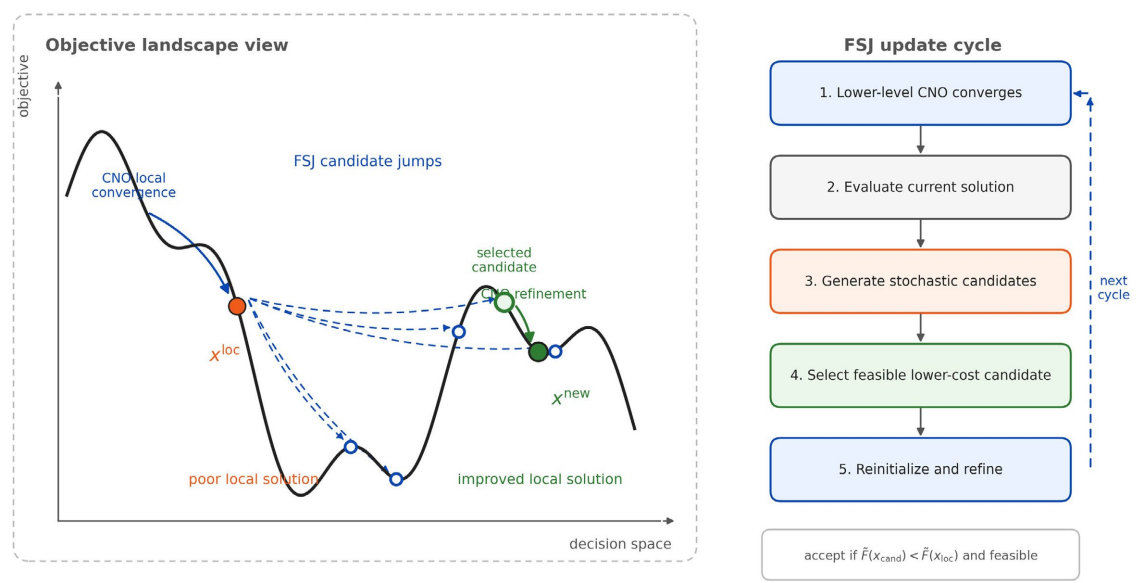


Figure 3. Illustration of the FSJ mechanism
图 3. FSJ 机制示意图

6. 实验

6.1. 主模块仿真测试

- 仿真实验主要针对轨迹优化端开展，其主要验证目标如下：
- (1) 基于 B 样条的轨迹参数化与惩罚函数建模能够形成可计算的后端优化问题；
 - (2) CNO 网络组能够通过多节点协作降低惩罚目标函数并逐渐达成一致；

- (3) FSJ 上层机制能够在局部搜索结果不理想时提供有效跳跃, 从而进一步改善解的质量;
- (4) 所生成轨迹能够在给定场景中满足速度、加速度及障碍物安全距离等基本飞行要求。
- 部分参数设置如表 2 所示, 其中样条次数 $2s-1$ 在[16]中被证明是使轨迹最优的最小次数。

Table 2. Main simulation parameter settings

表 2. 主要仿真参数设置

参数	取值	含义
控制阶数 s	3	Minimum Jerk
样条次数	5	$2s-1$
控制点总数	9	
普通优化采样点数	240	控制成本、路径点等惩罚计算
走廊惩罚采样点数	800	飞行走廊惩罚计算
稠密检查采样点数	1200	可行性检验
最大速度	3	
最大加速度	4	
CNO 协作节点数	4	
CNO 主迭代次数	140	
共识收敛迭代次数	80	
FSJ 候选点个数	30	上层随机候选解数量
FSJ 细化候选数	2	进入下层级 CNO 细化的候选点数

搭建 CNO 基础模块后, 本文首先在其中验证共识收敛性, 得到如图 4 所示的收敛曲线和共识残差曲线。由左图曲线可见, 随着迭代进行, CNO 网络组的共识残差最终趋于 0, 说明多个求解节点能够通过拉普拉斯耦合逐渐达成一致; 同时如右图所示, 惩罚目标函数下降, 表明系统能有效收敛至局部最优。

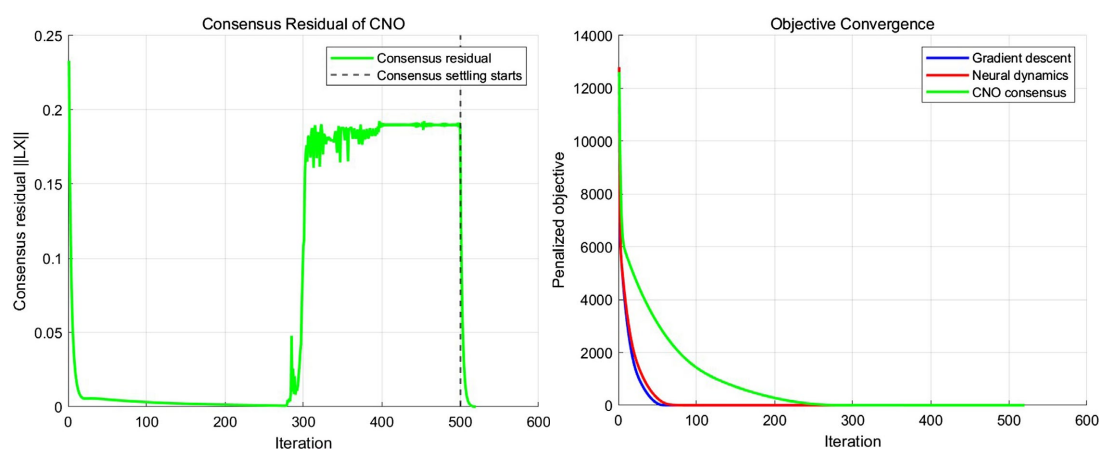


Figure 4. CNO consensus convergence

图 4. CNO 共识收敛

随后设置强调 FSJ 模块, 验证其作用, 实验得如图 5 所示曲线与轨迹示意图。在执行过程中, FSJ 首

先生成一组随机候选点，并对其进行原始惩罚目标函数评估，然后选取目标函数较低的若干候选点调用下层级 CNO 进行局部细化。结果显示，在典型实验中，原始初始点仅经过下层级 CNO 后所得局部目标函数值约 $1.663308e7$ ；而 FSJ 候选点细化后，目标值进一步降低至 $1.657983e7$ ，各候选点均优于原始局部 CNO 结果，表明 FSJ 确实能够通过重新初始化帮助优化过程跳出较差局部吸引域。

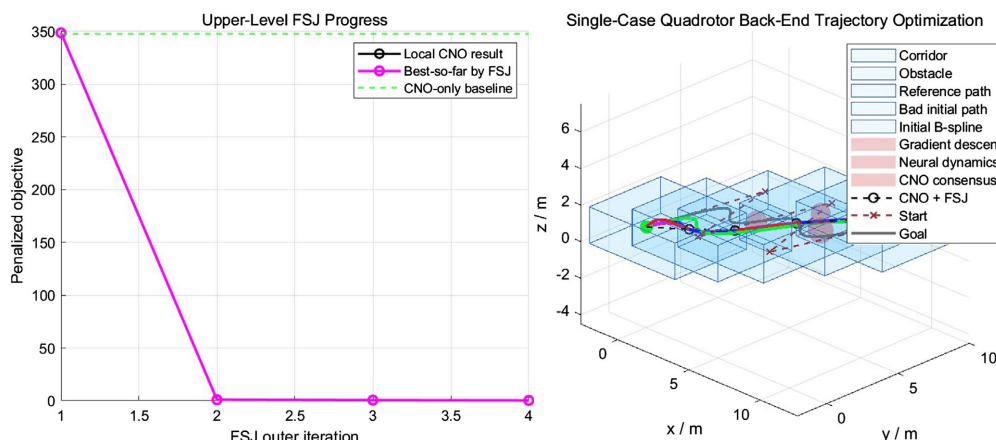


Figure 5. Verification of FSJ jump improvement

图 5. FSJ 跳跃改进验证

此外，动力学上速度与加速度的约束也在实验中得到检查，如图 6 所示的结果表明，速度和加速度不会超出限制，即优化所得的轨迹对四旋翼而言是工程上可执行的。

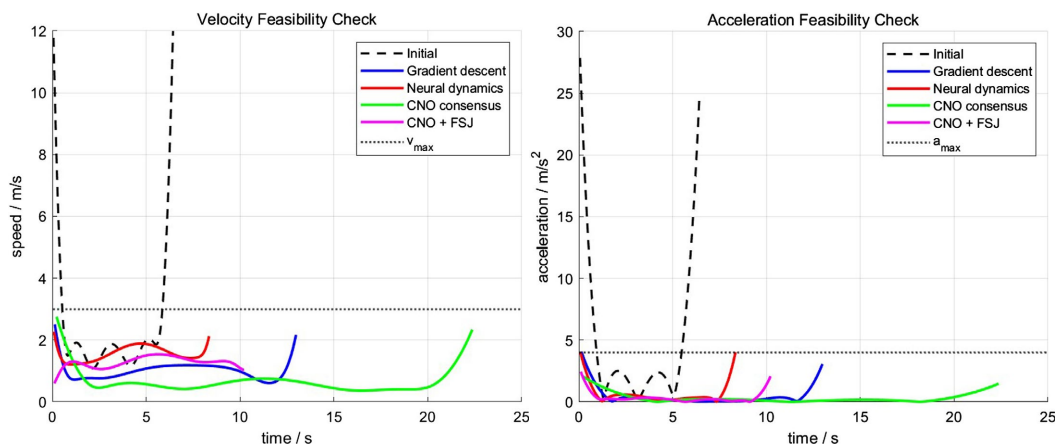


Figure 6. Feasibility check of speed and acceleration

图 6. 速度与加速度可行性检验

6.2. 实机配置

本文实机实验采用的自主研发的四旋翼，第一代机架选用 HMF Totem Q250 型穿越机架，轴距为 250 mm；机载计算单元采用 NVIDIA Jetson Orin NX 8 GB 模组，其搭载 6 核 Arm Cortex-A78AE 处理器与 1024 核 NVIDIA Ampere 架构 GPU，可提供 70 TOPS 的 INT8 稀疏算力；感知系统选用 Intel RealSense D435 深度相机，支持 $1280 \times 720 @ 90 \text{ fps}$ 的深度图像输出，深度视场角为 $87^\circ \times 58^\circ$ ，最小感知距离为 0.105 m；动力系统由 T-MOTOR F60 PRO IV V2.0 无刷电机(KV2550)与 Gem fan 5043 三叶螺旋桨组成，

在 14.46 V 电压下最高转速可达 26,363 RPM，最大功率为 602.31 W；飞行控制器采用 PX4，所有机载运算与通信均基于 ROS (Robot Operating System) 进行统一调度与管理。实机结构如图 7 所示。

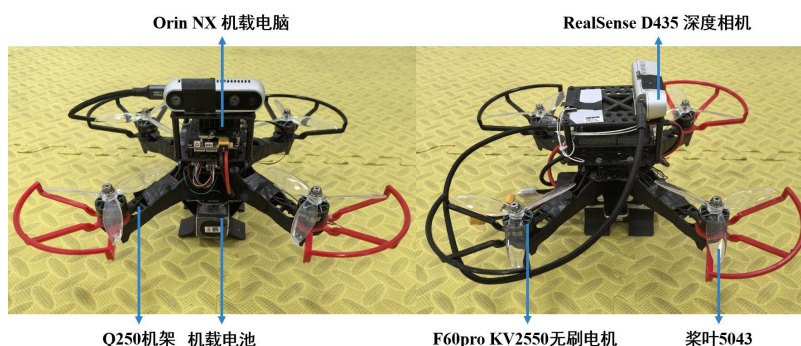


Figure 7. Structure of the quadrotor experimental platform
图 7. 四旋翼实机平台结构

在实机实验的过程中，上级程序首先指定飞行任务，并生成参考路径和轨迹优化问题；随后，采用本文提出的 CNO + FSJ 方法优化样条轨迹；最后通过通信接口传输至飞行控制系统以供执行。完成平台搭建后，在室外环境下进行实机测试，场景主要涉及农田间的半开放巡检(图 8)和林下自主避障(图 9)。



Figure 8. Semi-open field cruising coverage
图 8. 田间半开放巡航覆盖



Figure 9. Obstacle-avoidance flight tests under trees
图 9. 林下避障飞行测试

7. 总结

本文面向四旋翼轨迹规划中的非凸约束处理与局部最优问题,提出了一种基于协作神经动力学优化的后端轨迹规划方法。通过微分平坦性将四旋翼轨迹规划问题转化为样条参数优化问题,并将飞行走廊、速度、加速度等约束统一建模为可微惩罚项;在此基础上构建分布式神经动力学优化网络组实现多节点协同优化,并进一步引入快速随机跳跃算法以改善局部最优解质量。仿真结果表明,所提方法能够有效降低目标函数,生成满足速度与加速度约束的平滑轨迹;实机平台与室外飞行展示进一步说明该方法具备接入真实四旋翼系统的工程应用潜力。

参考文献

- [1] Fliess, M., Lévine, J., Martin, P. and Rouchon, P. (1995) Flatness and Defect of Non-Linear Systems: Introductory Theory and Examples. *International Journal of Control*, **61**, 1327-1361. <https://doi.org/10.1080/00207179508921959>
- [2] Van Nieuwstadt, M.J. and Murray, R.M. (1998) Real-Time Trajectory Generation for Differentially Flat Systems. *International Journal of Robust and Nonlinear Control*, **8**, 995-1020. [https://doi.org/10.1002/\(sici\)1099-1239\(199809\)8:11<995::aid-rnc373>3.0.co;2-w](https://doi.org/10.1002/(sici)1099-1239(199809)8:11<995::aid-rnc373>3.0.co;2-w)
- [3] Martin, P., Murray, R.M. and Rouchon, P. (2003) Flat Systems, Equivalence and Trajectory Generation. https://www.researchgate.net/publication/37267037_Flat_Systems_Equivalence_and_Trajectory_Generation
- [4] Ryu, J.C. and Agrawal, S.K. (2011) Differential Flatness-Based Robust Control of Mobile Robots in the Presence of Slip. *The International Journal of Robotics Research*, **30**, 463-475. <https://doi.org/10.1177/0278364910385586>
- [5] Mellinger, D. and Kumar, V. (2011) Minimum Snap Trajectory Generation and Control for Quadrotors. 2011 *IEEE International Conference on Robotics and Automation*, Shanghai, 9-13 May 2011, 2520-2525. <https://doi.org/10.1109/icra.2011.5980409>
- [6] Watterson, M. and Kumar, V. (2019) Control of Quadrotors Using the Hopf Fibration on $SO(3)$. In: *Springer Proceedings in Advanced Robotics*, Springer, 199-215. https://doi.org/10.1007/978-3-030-28619-4_20
- [7] Ferrin, J., Leishman, R., Beard, R. and McLain, T. (2011) Differential Flatness Based Control of a Rotorcraft for Aggressive Maneuvers. 2011 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, 25-30 September 2011, 298-304. <https://doi.org/10.1109/iros.2011.6048861>
- [8] Faessler, M., Franchi, A. and Scaramuzza, D. (2017) Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories. *IEEE Robotics and Automation Letters*, **3**, 620-626. <https://doi.org/10.1109/lra.2017.2776353>
- [9] Xia, Z.C., Liu, Y. and Wang, J. (2022) A Collaborative Neurodynamic Approach to Distributed Global Optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **53**, 3141-3151. <https://doi.org/10.1109/tsmc.2022.3221937>
- [10] Wang, J.S., Wang, J. and Che, H.J. (2019) Task Assignment for Multivehicle Systems Based on Collaborative Neurodynamic Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 1145-1154. <https://doi.org/10.1109/tnnls.2019.2918984>
- [11] Che, H.J. and Wang, J. (2019) A Collaborative Neurodynamic Approach to Global and Combinatorial Optimization. *Neural Networks*, **114**, 15-27. <https://doi.org/10.1016/j.neunet.2019.02.002>
- [12] Bertsekas, D.P. (2012) Dynamic Programming and Optimal Control: Volume I. Vol. 4. Athena Scientific.
- [13] Toth, C.D., O'Rourke, J. and Goodman, J.E. (2017) Handbook of Discrete and Computational Geometry. CRC Press.
- [14] Wang, Q.H., Wang, Z., Wang, M., Ji, J., Han, Z., Wu, T., et al. (2025) Fast Iterative Region Inflation for Computing Large 2-D/3-D Convex Regions of Obstacle-Free Space. *IEEE Transactions on Robotics*, **41**, 3223-3243. <https://doi.org/10.1109/tro.2025.3562482>
- [15] Wang, J. and Elia, N. (2010) Control Approach to Distributed Optimization. 2010 *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, 29 September-1 October 2010, 557-561. <https://doi.org/10.1109/allerton.2010.5706956>
- [16] Wang, Z.P., Zhou, X., Xu, C. and Gao, F. (2022) Geometrically Constrained Trajectory Optimization for Multicopters. *IEEE Transactions on Robotics*, **38**, 3259-3278. <https://doi.org/10.1109/tro.2022.3160022>