

中图分类号：TP249

论文编号：10006SY2003533

北京航空航天大学
硕士学位论文

基于模型预测的无人机
密集编队跟踪控制算法研究

作者姓名 李谨杰

学科专业 导航、制导与控制

指导老师 任章 教授

培养院系 自动化科学与电气工程学院

**Close Formation Tracking of Quadrotors
with Model Predictive Control**

A Dissertation Submitted for the Degree of Master

Candidate: LI, Jinjie

Supervisor: Prof. REN, Zhang

School of Automation Science and Electrical Engineering

Beihang University, Beijing, China

中图分类号: TP249

论文编号: 10006SY2003533

硕士学位论文

基于模型预测的无人机 密集编队跟踪控制算法研究

作者姓名	李谨杰	申请学位级别	工学硕士
指导教师姓名	任章	职 称	教授
学科专业	控制科学与工程	研究方向	导航、制导与控制
学习时间自	2020 年 9 月 1 日	起至	2023 年 6 月 30 日止
论文提交日期	2023 年 5 月 24 日	论文答辩日期	2023 年 5 月 15 日
学位授予单位	北京航空航天大学	学位授予日期	年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期： 年 月 日

指导教师签名：_____ 日期： 年 月 日

摘 要

多旋翼无人机集群常常以编队飞行的方式移动。当无人机集群在城市、森林等充满障碍的复杂环境中穿行时，需要收缩队形进行密集编队以提高在狭窄空间中的穿越能力。然而，多旋翼无人机通过旋翼向下方吹动空气以产生升力，这种下洗气流气动效应容易干扰近距离飞行的其他无人机，从而威胁密集编队飞行安全。为解决上述问题，本文以四旋翼无人机为研究对象，通过神经网络建模机间下洗气流扰动，同时结合非线性模型预测控制（Model Predictive Control, MPC）方法提出密集编队跟踪控制算法，最终通过仿真与实物飞行验证算法有效性。论文主要研究成果如下：

（1）针对无人机受到的强非线性下洗气流扰动问题开展研究。首先设计多层感知机网络结构，确定输入输出变量。接着设计数据收集实验，操纵双无人机飞行以收集扰动数据。然后通过扰动数据训练神经网络，同时引入谱归一化机制提升网络的鲁棒性。最后通过线性加和的方式将扰动观测扩展到多无人机。在扰动数据测试集上的结果显示网络观测器能较好拟合下洗气流扰动现象，同时鲁棒性较强，可有效支持控制器设计。

（2）针对基于模型预测控制的轨迹跟踪问题开展研究。首先实现 Minimum Snap 多项式轨迹生成算法，并基于微分平坦性质转化为全状态轨迹。接着基于具有预测能力的非线性 MPC 方法设计轨迹跟踪控制算法，同时针对其运算量大的特点进行算法优化。通过仿真与实物飞行验证控制器可在机载计算机实时运行并以较高精度跟踪轨迹。

（3）针对存在下洗气流扰动的密集编队飞行问题开展研究。首先将双无人机扰动观测器与跟踪控制器结合以实现轨迹跟踪飞行。接着在 MPC 优化目标中加入协同项以设计编队跟踪控制器。最后将多无人机扰动观测器与编队控制器结合以实现密集编队飞行。通过仿真验证控制器可实现多无人机密集编队飞行；通过实物验证控制器可在两架无人机上实时运行，并能在下洗气流扰动中以较高精度跟踪轨迹。

（4）面向密集编队问题，开发六自由度四旋翼集群仿真平台，可模拟下洗气流效应。对实物四旋翼进行参数辨识，使得仿真验证过的算法可直接向实物迁移。采用面向数据编程范式，利用 GPU 实现并行计算加速，可支持上千非线性四旋翼节点的仿真。

关键词：多旋翼飞行器，编队跟踪，模型预测控制，下洗气流，神经网络

Abstract

Swarm multi-rotor drones often move in formation during flight. When navigating through complex environments such as cities and forests, swarm drones contract into a close formation to enhance their maneuverability in narrow spaces. However, the downward airflow generated by the drones' rotors can interfere with nearby ones during close-proximity flights and threaten flight safety. To address this problem, this thesis models inter-vehicle downwash disturbance using a neural network and proposes a close formation tracking approach based on nonlinear model predictive control (MPC), which are verified through simulated and physical flights. The main contributions of this thesis are as follows:

(1) This part focuses on the high nonlinear property of downwash disturbance experienced by quadrotors. First, a multi-layer perceptron neural network is designed to determine the input and output variables. Then, data collection experiments are designed to collect disturbance data. Next, the neural network is trained using the disturbance data, and spectral normalization is implemented to improve the network's robustness. Finally, the disturbance observer is extended to multiple drones through linear assumption. The network observations on disturbance test data show that it can fit the downwash disturbance accurately with strong robustness, which can effectively support controller design.

(2) This part focuses on the trajectory tracking problem using model predictive control methods. First, the Minimum Snap algorithm is implemented to generate polynomial trajectories, which are then transformed into a full-state trajectory based on differential flatness. Then, a trajectory tracking algorithm is designed based on the nonlinear MPC method, and the computational acceleration are introduced to address the MPC's computational burden. Finally, simulated and physical flights show that the controller can accurately track trajectories in real time on onboard computers.

(3) This part focuses on close formation tracking with downwash disturbances. First, the two drones' disturbance observer and tracking controller are integrated to achieve trajectory tracking. Then, a formation tracking controller is designed by adding a cooperation term to the MPC optimization objective. Finally, the disturbance observer for multiple quadrotors and the

formation tracking controller are combined to achieve close formation flight. Simulation shows that the controller can achieve close formation flight, and physical flights demonstrate that the controller can track trajectories with high accuracy under downwash disturbances.

(4) To verify the above algorithms, a six-degree-of-freedom multi-quadrotor simulator is developed, which supports the simulation of the downwash effect. The physical parameters of the quadrotors are identified to enable direct algorithm migration from simulation to physical systems. Data-driven programming is utilized with parallel computation using GPUs, which can support the simulation of thousands of nonlinear quadrotors.

Keywords: Quadrotors, Formation Tracking, Model Predictive Control, Downwash Effect, Neural Network

目 录

第一章	绪论	1
1.1	研究背景及意义	1
1.2	国内外研究现状	2
1.2.1	多无人机集群系统研究现状	2
1.2.2	气动扰动建模方法研究现状	5
1.2.3	模型预测控制方法研究现状	9
1.3	解决的关键问题	12
1.4	论文主要工作及内容安排	13
第二章	四旋翼无人机动力学建模及参数辨识	15
2.1	常用坐标系及其转换	15
2.2	四旋翼无人机动力学模型	16
2.2.1	六自由度刚体动力学	16
2.2.2	旋翼动力系统建模	17
2.2.3	旋翼气动阻力建模	18
2.2.4	全量动力学模型	19
2.3	角速度控制器与油门推力映射	20
2.4	四旋翼无人机参数辨识	23
2.4.1	参数辨识过程介绍	24
2.4.2	旋翼系统数据分析	26
2.4.3	惯性属性数据分析	32
2.4.4	模型及仿真参数汇总	33
2.5	本章小结	35
第三章	多无人机机间下洗气流扰动估计	36
3.1	下洗气流扰动仿真模型	36
3.2	神经网络下洗气流扰动观测器	38
3.3	机间下洗气流扰动数据收集	39
3.3.1	扰动数据的收集原理	39
3.3.2	扰动数据的收集过程	40
3.4	谱归一化神经网络训练方法	42
3.5	神经网络观测器的训练和测试	43

3.6	多无人机下洗气流扰动观测	46
3.7	本章小结	47
第四章	无人机非线性 MPC 轨迹跟踪控制算法	48
4.1	Minimum Snap 轨迹生成算法	48
4.2	基于微分平坦生成全状态轨迹	52
4.3	非线性 MPC 轨迹跟踪控制算法	53
4.3.1	非线性 MPC 采用的四旋翼模型	53
4.3.2	非线性 MPC 轨迹跟踪控制算法	55
4.3.3	非线性 MPC 的运算加速技巧	57
4.3.4	基于卡尔曼滤波的悬停油门观测器	58
4.4	仿真验证	59
4.4.1	轨迹规划与轨迹跟踪	59
4.4.2	卡尔曼滤波悬停油门观测	64
4.5	实物验证	65
4.5.1	实验过程	65
4.5.2	实验结果	66
4.6	本章小结	67
第五章	多无人机非线性 MPC 密集编队跟踪控制算法	68
5.1	气动补偿下的无人机轨迹跟踪控制算法	68
5.2	分布式非线性 MPC 编队跟踪控制算法	69
5.3	气动补偿下的多机密集编队跟踪控制算法	72
5.4	仿真验证	73
5.4.1	多机非线性 MPC 编队跟踪	73
5.4.2	气动补偿下的密集编队跟踪	75
5.5	实物验证	77
5.5.1	实验过程	77
5.5.2	实验结果	78
5.6	本章小结	79
第六章	多无人机下洗气流仿真环境与实物系统	80
6.1	无人机软件系统介绍	80
6.2	基于 ROS 的多无人机仿真平台	81

6.2.1 六自由度仿真平台系统架构	81
6.2.2 面向数据编程与 GPU 并行计算加速	83
6.2.3 基于 ROS 的通讯结构介绍	85
6.3 多无人机硬件系统设计	87
6.4 本章小结	89
结 论	90
参考文献	92
附 录	97
攻读硕士学位期间取得的学术成果	100
致 谢	101
作者简介	103

图清单

图 1 DARPA 挑战赛冠军团队 CERBERUS 使用的机器人 ^[4]	2
图 2 各种类型的无人机 ^[1]	3
图 3 第一个全自主无人机集群系统及其性能指标 ^[13]	4
图 4 机器学习结合物理模型的结合方式	7
图 5 动态建模框图	7
图 6 控制学科发展历程 ^[49]	9
图 7 机器学习改善系统模型分类	11
图 8 本文各章节间关系	13
图 9 坐标系定义图	15
图 10 本文使用的实物无人机系统	23
图 11 连接支架及连接后的测试状态	24
图 12 电机参数测试环境	25
图 13 采用双线摆的转动惯量测量装置	26
图 14 转速平方随油门的变化曲线	26
图 15 电机动力为 16% 下的无人机三轴受力	27
图 16 电机动力为 44% 下的无人机三轴受力	27
图 17 推力随转速平方的变化曲线	28
图 18 12% 油门下由辨识得到的电机转速响应曲线	29
图 19 60% 油门下由辨识得到的电机转速响应曲线	30
图 20 92% 油门下由辨识得到的电机转速响应曲线	30
图 21 电机动力为 60% 下的无人机三轴所受力矩	31
图 22 力矩随转速平方的变化曲线	31
图 23 转动惯量测量装置示意图 ^[59]	32
图 24 仿真中的角速度跟踪效果	34
图 25 第三章内容逻辑图	36
图 26 不同相对高度下仿真产生的下洗气流扰动力	37
图 27 下洗气流仿真模型在仿真环境中的影响	38

图 28 实物飞行中机体对 Z 轴所受扰动力的观测	41
图 29 下洗气流扰动数据收集过程	41
图 30 下洗气流收集数据按相对位置关系进行可视化	42
图 31 $\gamma = 4$ 时神经网络的训练曲线	44
图 32 不同谱归一化系数、不同高度下神经网络对 Z 轴的扰动力预测	45
图 33 下洗气流水平叠加示意图	46
图 34 下洗气流垂直叠加示意图及仿真效果	47
图 35 第四章内容逻辑图	48
图 36 由多个多项式曲线组成的轨迹	50
图 37 模型预测控制原理图	55
图 38 非线性 MPC 算法的实现流程	57
图 39 低速场景下无人机轨迹跟踪仿真效果图	62
图 40 低速场景下轨迹跟踪仿真结果	62
图 41 高速场景下无人机轨迹跟踪仿真效果图	63
图 42 高速场景下轨迹跟踪仿真结果	64
图 43 仿真中的悬停油门观测量随时间的变化	65
图 44 非线性 MPC 轨迹跟踪实物飞行实验图	66
图 45 非线性 MPC 轨迹跟踪实物飞行信息可视化	66
图 46 第五章内容逻辑图	68
图 47 神经网络气动补偿对非线性 MPC 模型的改进	69
图 48 分布式 MPC 的基本思想	70
图 49 通讯拓扑示意图	70
图 50 两架无人机队形切换仿真效果图	73
图 51 三架无人机编队跟踪仿真效果图	74
图 52 三架无人机编队切换仿真效果图	74
图 53 三架无人机编队跟踪仿真结果	75
图 54 气动补偿下的三架无人机编队仿真效果图	76
图 55 加入气动扰动补偿的三架无人机编队跟踪仿真结果	76

图 56 气动补偿下的非线性 MPC 轨迹跟踪实物实验	77
图 57 气动补偿下的非线性 MPC 轨迹跟踪实物实验结果	78
图 58 本文涉及的三个 ROS 仓库的序列图	80
图 59 dop_qd_sim 仿真平台的程序结构	82
图 60 RVIZ 多无人机显示界面	82
图 61 dop_qd_sim 并行计算的核心思想	84
图 62 每一轮仿真的运行时间随智能体数量变化的曲线	85
图 63 单机轨迹跟踪采用的 ROS 通讯架构	86
图 64 三机编队跟踪采用的 ROS 通讯架构	86
图 65 实验室之前的无人机硬件系统	87
图 66 升级之后的无人机硬件系统	87
图 67 实验平台硬件框图	88
图 68 实验平台硬件连线图	88

表清单

表 1 MPC 发展历程 ^[49]	10
表 2 六轴力传感器 SI-165-15 的量程	24
表 3 无人机的转动惯量测试结果	32
表 4 对无人机辨识的参数	33
表 5 仿真中的气动阻力模型参数设置	34
表 6 仿真中的角速度控制器采用参数	34
表 7 仿真中采用的下洗气流模型参数	37
表 8 神经网络扰动观测器的训练参数	44
表 9 在不同谱归一化系数下的测试集 Loss	44
表 10 Minimum Snap 仿真路径点设置	60
表 11 仿真中非线性 MPC 采用的控制参数	60
表 12 低速轨迹参数	61
表 13 高速轨迹参数	63
表 14 不同语言版本和 PyTorch 版本下的仿真运行时间	84
表 15 TX2 NX 软件环境配置	89
表 16 旋翼转速随油门变化的原始数据	97
表 17 无人机的扭转摆动周期	98
表 18 无人机硬件选型	98

主要符号表

\mathbb{R}	实数集合
\mathbb{R}^n	n 维实列向量集合
$\mathbb{R}^{n \times m}$	$n \times m$ 阶实矩阵集合
$x \in \mathbb{R}$	标量
$\mathbf{x} \in \mathbb{R}^n$	向量
$\mathbf{X} \in \mathbb{R}^{n \times m}$	矩阵
$[\cdot]$	排列, 如向量或矩阵
(\cdot)	函数
$\hat{\cdot}$	估计变量
$\ \mathbf{x}\ $	向量 \mathbf{x} 的 2-范数 (欧几里得范数)
$\ \mathbf{x}\ _n$	向量 \mathbf{x} 的 n -范数
$\text{clip}(x, [x_{\text{MIN}}, x_{\text{MAX}}])$	将变量 x 限制在 $[x_{\text{MIN}}, x_{\text{MAX}}]$ 之间

第一章 绪论

1.1 研究背景及意义

本论文受科技创新 2030——“新一代人工智能”重大项目《面向群体行为的群智激发汇聚研究》中的课题五《面向异构无人集群的协同对抗环境应用示范》支撑，主要关注无人集群中的密集编队跟踪问题。

随着电子元件的小型化和芯片计算能力的提升，无人机（Drone），又称为空中机器人（Aerial Robots），已经成为机器人与自动控制领域的研究热点。其中多旋翼飞行器以多个电动旋翼为动力，通过旋翼旋转产生的升力飞行，具备垂直起降能力，且体积较小、机动性强，对实验场地要求较低，因而在学界与业界得到广泛的关注^[1]。随着大疆、Skydio 等商业公司逐渐将单个多旋翼无人机推向成熟化、产品化，学界逐渐将研究目光转向无人集群。无人集群是一种灵活的分布式智能自主系统，与单无人机相比，无人集群可以显著增强灵活性（包括适应性、可扩展性和可维护性）和鲁棒性（包括可靠性、生存能力和容错性）^[2]。因此，无人集群逐渐被认为是一项可以在军事和民用领域产生重大影响的颠覆性技术。

与单体机器人类似，集群机器人也存在感知、决策、规划、控制等方面的问题，而编队跟踪控制属于控制领域较为基础的问题。与地面机器人集群系统相比，气动扰动是影响空中机器人集群系统的重要问题。当无人机进行密集编队飞行时，相互之间复杂的气动扰动会给编队跟踪控制器的设计带来较大挑战，因此对下洗气流扰动进行建模对于提升密集编队跟踪控制性能具有重要意义。此外，经典的串级 PID 或几何控制方法无法对上述扰动提前作出反应，补偿扰动的速度较为有限。而基于滚动优化的模型预测控制方法具有预测性质，可以在控制中提前考虑未来可能发生的扰动，同时能够对系统的状态和输入进行限制，对无人集群在扰动下的安全飞行具有重要意义。

综上所述，本文将从拓展多无人机编队技术实际应用场景出发，研究包含气动扰动建模补偿的多无人集群密集编队跟踪问题。本文通过神经网络对下洗气流进行建模，将所建模型与传统四旋翼模型融合，结合模型预测控制框架，设计面向无人机轨迹跟踪和多无人集群密集编队跟踪的算法，最后在仿真环境和实物平台验证算法效果。

1.2 国内外研究现状

本节首先介绍多无人机集群系统的国内外研究现状，接着介绍多无人机编队跟踪的研究现状，得出机间存在的下洗气流扰动是密集编队跟踪的主要难点。然后介绍国内外文献对气动扰动的建模方法，其中着重调查基于机器学习（Machine Learning, ML）的建模技术研究现状。最后介绍模型预测控制（Model Predictive Control, MPC）方法的研究现状，这种方法可以处理上述包含机器学习的高度非线性模型，也可以处理状态和输入的约束，因而较适合被用来设计多无人机密集编队跟踪控制器。

1.2.1 多无人机集群系统研究现状

随着机载设备计算能力的增强和计算机视觉、人工智能、数值优化、微型传感器等技术的进步，机器人技术在过去的十几年中迅速发展，在学术界产生了层出不穷的研究，在工业界也吸引了大量的关注。在各式各样的机器人当中，自主移动机器人（Autonomous Mobile Robots）指的是一类可在无人参与的情况下用于空间移动的机器人^[3]，因其较强的移动能力而受到广泛关注。如下图所示，在 2018~2021 年由美国 DARPA 举办的地下挑战赛中，来自世界各国的团队使用轮式、腿式、无人机等多种移动机器人共同完成在地下山洞内的种种任务，代表了移动机器人当前研究的最高水平^[4, 5]。



图 1 DARPA 挑战赛冠军团队 CERBERUS 使用的机器人^[4]

作为移动机器人的重要分支，无人机可以将机器人的运动能力从地面拓展到天空，大大增加了运动速度与运动范围，因而在物流运输^[6]、地形探测^[7]、环境监测^[8]、精准农业^[9]等领域具有广阔的应用前景。如下图所示，无人机可以分为固定翼类型、旋翼类型、扑翼类型和其他类型^[1]。其中固定翼无人机是有人驾驶飞行器微型化和智能化的结果，由快速移动的机翼产生升力维持飞行。这种类型的无人机续航时间长，飞行距离远，但机动能力较差，受限于滑跑起降所需的飞行场地与较大的转弯半径，只能在具有广阔飞行环境的任务中使用。旋翼无人机是近二十年蓬勃发展的一类飞行器，仅由多个旋翼产生升力维持飞行。这种类型的无人机续航时间相对较短，但可以垂直起降，原地悬停，飞行非常灵活，可以在较为局限的室外飞行场地甚至室内环境进行飞行，得到了更广泛的关注。扑翼无人机一般受生物启发，模仿昆虫或者鸟类的飞行原理进行设计。由于材料和能量等限制，这种飞行器目前还不太成熟，有待学界进一步研究。此外，还有大量介于上述三者之间的无人机，如倾转旋翼垂直起降无人机^[10]、尾座式垂直起降无人机^[11]、可变翼型无人机^[12]等。旋翼类飞行器结构设计成熟、应用广泛，是本文主要关注的飞行器类型。

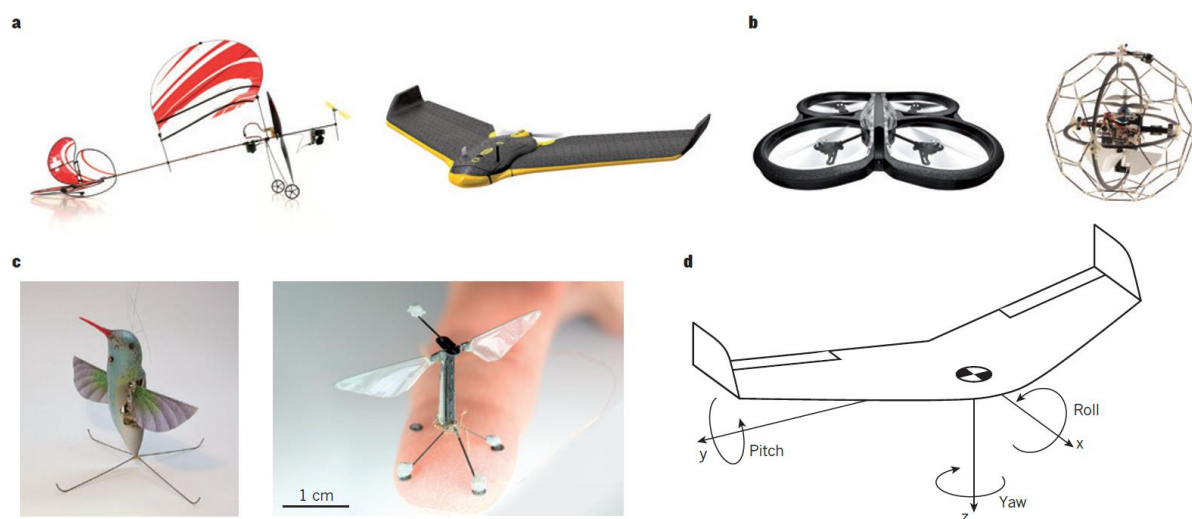


图 2 各种类型的无人机^[1]

近二十年学术界对于四旋翼飞行器已开展大量研究，奠基性研究团队包括由 Vijay Kumar 教授领导的宾夕法尼亚大学 GRASP 实验室、由 Roland Siegwart 教授领导的苏黎世联邦理工学院 ASL 实验室、由 Raffaello D'Andrea 教授领导的苏黎世联邦理工学院动态系统实验室、由 Jonathan P. How 教授领导的麻省理工学院 ACL 实验室、由 Claire Tomlin 教授领导的动态系统实验室，中生代研究团队包括由 Davide Scaramuzza 教授领

导的苏黎世大学 RPG 实验室、由沈劭劼教授领导的香港科技大学空中机器人实验室、由 Angela Schoellig 教授领导的慕尼黑工业大学 LSY 实验室等。

随着大疆等公司将单体无人机的性能推向极致，目前学术界对无人机的研究方向逐渐转向了无人机集群系统。如下图所示，由高飞教授领导的浙江大学 FAST 实验室近期基于优化框架实现了世界上第一个全自主无人机集群系统，发表在机器人领域顶级刊物 *Science Robotics* 封面上^[13]，代表了这一领域的最新研究成果。与单体无人机类似，集群无人机也存在感知、决策、规划、控制等方面的问题。本文主要聚焦于集群无人机的编队控制问题，希望实现密集编队飞行，拓展集群无人机的运动能力。

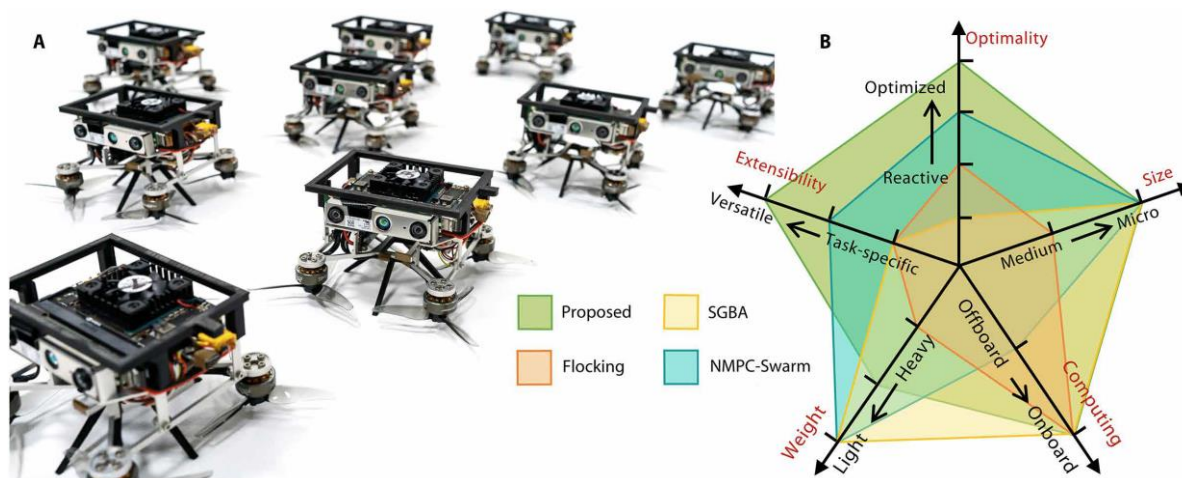


图 3 第一个全自主无人机集群系统及其性能指标^[13]

编队控制问题自上世纪 90 年代开始就受到了机器人和控制领域的关注，主要包括基于行为^[14]、基于虚拟结构^[15]、基于领导者-跟随者^[16]、基于一致性理论^[17, 18]的经典编队控制方法和基于深度强化学习^[19]等理论的智能编队控制方法。

在基于行为的编队控制策略中，集群系统的每个智能体都具有几种预定的行为模式，这些行为模式构成一个行为集。每一种行为都可以产生相应的控制作用，智能体的最终控制器由这些行为的控制作用通过加权求和得到。基于虚拟结构的编队控制的基本思想是把期望的编队看作一个刚性的虚拟结构，集群系统中的每个智能体对应虚拟结构上的一个节点，当编队移动时，智能体跟踪虚拟结构上对应的节点进行运动即可。基于领导者-跟随者的编队控制方法是将集群系统中一个或者多个智能体作为领导者，其余智能体作为跟随者，所有跟随者都与领导者进行直接通信，领导者按照指定的路径运动，跟随者与领导者保持特定的相对位置及角度关系进行运动。基于一致性理论的方法根据任务要求和性能指标对各个智能体设计局部控制策略，使得智能体间相互作用，最终使某

些状态量趋于一致，形成编队。

为简化推导与证明，这些理论往往采用不含相互影响的独立模型进行分析，当应用到真实的四旋翼无人机时，个体之间的飞行距离较远，以避免无人机之间发生碰撞^[20]。与理论中的假设不同，实物四旋翼存在下洗气流扰动，当距离较近时无人机之间可能存在较强的气动干扰，影响飞行安全^[21]。对于基于深度强化学习等理论的端到端智能编队算法，虽然部分文献会将气动扰动考虑在内，将算法在实物上进行验证，但受限于神经网络的不可解释性和弱泛化性，存在实验相撞的风险，其面向的任务场景也较为单一^[19]。只有少量集群研究将无人机间的气动扰动包含在内，且大多对象为固定翼，如文献^[21]对固定翼的密集编队问题进行分析，考虑了执行器故障和翼尖湍流的影响；文献^[22]将四旋翼的下洗气流作用区域作为整体进行集群轨迹规划，但分析较为粗略。

总体来说，虽然已有大量关于编队的理论成果，但传统的编队控制方法没有将多旋翼无人机之间存在的气动扰动纳入控制律的设计，导致在实际飞行时无法实现无人机密集编队，限制了无人机集群的运动能力与灵活性，不利于充分发挥集群技术的优势。而端到端智能算法虽然可以考虑气动扰动带来的问题，但数据需求大，训练时间长，泛化性差，也较难在实际工程中开展实践。因此，如何在较少的数据需求下对无人机和环境、无人机之间的气动扰动进行建模，是解决无人机密集编队跟踪控制问题的关键。

1.2.2 气动扰动建模方法研究现状

对气动扰动进行准确、高效的建模属于非线性系统建模问题，而深度学习的发展为气动建模提供新的工具。本小节第一部分分析气动扰动现象，介绍传统分析方法；第二部分介绍使用机器学习建模的研究现状；第三部分介绍近三年的最新相关研究。

(1) 气动扰动现象分析

排除风等外部环境的影响，四旋翼无人机在集群飞行时存在的气动扰动主要包含地面效应、飞行阻力和来自于其他无人机下洗气流的影响。针对旋翼飞行器，地面效应指的是飞行器飞行高度较低时，螺旋桨向下喷射的气流在地面附近发生的复杂相互作用。飞行阻力指的是飞行器在空旷环境飞行时，由自身与空气相互运动产生的阻力。而下洗气流是集群飞行中的独特现象，指当一个四旋翼从另外一个四旋翼下方经过时，会受到上方四旋翼向下喷射气流的影响。

气动扰动属于非高斯过程，因而不能将其近似为随机扰动处理。流体属于动力学系

统，常采用纳维-斯托克斯微分方程进行描述^[23]。此外，由四旋翼飞行产生的气动扰动与飞机本身的姿态关联性很高，使用被动抗扰动控制算法，通过反馈的方式进行扰动抑制响应较慢^[24]，无法满足四旋翼大机动快速飞行的需求。因此，对气动扰动进行建模是解决密集编队问题的重要途径。

对于这三种气动现象，只有地面效应得到了较为透彻的研究^[25]，而其余两种现象受飞机实际状态与飞行环境的影响显著，很难用传统物理方法进行建模。有的研究采用风洞对气动现象进行测量^[26]，有的研究使用计算流体力学进行建模^[27]，这两种方法虽然准确，但对实验设备或者计算能力有很高的要求。

(2) 基于机器学习的动态系统建模研究现状

近年来深度学习等机器学习领域新方法的蓬勃发展^[28, 29]为复杂现象的建模分析带来新的契机，控制学科也走到了控制理论、机器人学和人工智能的交叉点。深度学习的主要工具为深度神经网络，这本质上是一种效果极好的非线性拟合工具。因此，很多研究者开始尝试使用深度神经网络去解决传统控制理论难以解决的控制难题。这种尝试可以分为两个方向：一部分学者将深度神经网络应用于强化学习范式，抛弃传统控制方法，将控制问题转化为马尔科夫过程，使用深度强化学习以试错的方法学习状态到动作的映射，这已经在游戏^[30]、围棋^[31]、无人机^[32]、足式机器人^[33]、机械臂抓取^[34]等领域取得显著的成果。然而，由于深度强化学习学到的策略完全由神经网络表达，可解释性差，泛化性弱，数据需求量大，安全性也难以得到保障。因此另一部分学者尝试将基于物理现象的先验知识与深度神经网络方法进行耦合，以同时发挥传统方法与深度学习的优点，抵消彼此的缺点，达到更好的控制效果。传统方法往往基于模型（Model-Based）进行相应的分析与控制，因此这种结合主要使用深度神经网络对模型进行改进。文献^[35]对这种思路在地球系统、气候科学、湍流建模、材料发现、量子化学、生物科学、工程等领域的应用做了系统而全面的叙述。从具体目标来说，分为改进或替代物理模型、数据降采样、模型参数化、模型降阶、前向求解偏微分方程 PDE、发现未知物理现象中的关键方程、数据生成、不确定性量化等目标。从结合方式来说，分为以机器学习为主的方式和以物理模型为主的方式，具体分类如下图所示。

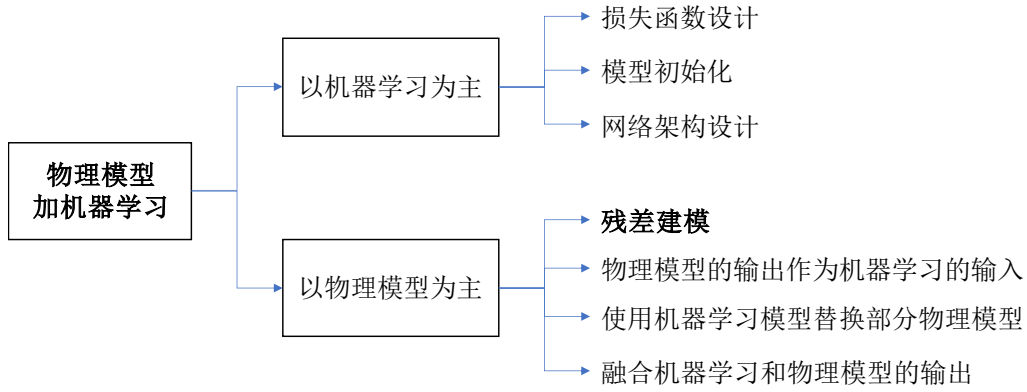


图4 机器学习结合物理模型的结合方式

以机器学习为主的方法又可以细分为在损失函数中融入物理知识、在模型初始化阶段融入物理知识、在神经网络架构设计阶段融入物理知识三个类别。纯机器学习获得模型的方法在对直升机^[36]、四旋翼^[37]的模型学习上取得了一定的成果。然而，因为纯机器学习具备数据量大、可解释性差等缺点，不适合在机器人尤其是空中机器人领域的多数应用场景中使用，因而不作为调研重点。以物理模型为主的方法又可以分为残差建模、将物理模型的输出作为机器学习的输入、使用机器学习模型替换部分物理模型、融合物理模型和机器学习模型输出四种方式。其中，使用机器学习对物理模型的误差进行建模是学界历史最悠久也是最常用的一种方式，也是本文调查的重点，其具体流程如下图。

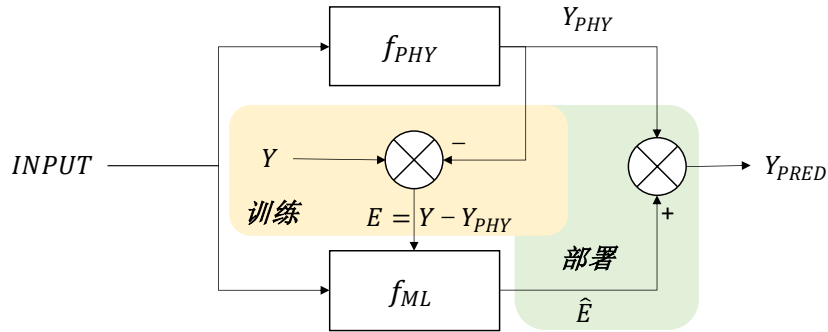


图5 动态建模框图

其中 $INPUT$ 表示输入， f_{PHY} 表示基于物理知识的传统模型， Y_{PHY} 表示传统模型的输出， f_{ML} 表示基于机器学习的补偿模型， Y 表示系统的实际输出， Y_{PRED} 表示补偿后的模型输出。在训练阶段，将系统输入 $INPUT$ 作为数据，将传统模型输出 Y_{PHY} 和系统实际输出 Y 的偏差 E 作为标签，输入机器学习模型对模型误差进行训练。在部署阶段，将机器学习模型预测的误差 \hat{E} 加到物理模型的输出 Y_{PHY} 上以进行修正，增加模型输出的准确性。

采用机器学习补偿误差的方式有如下优点：首先，与纯用神经网络训练相比，物理模型作为主导减少了对数据的要求；其次，神经网络仅对模型误差进行预测，这保证了主体规律的可预测性，确保了收集数据过程的安全；最后，神经网络的灵活性弥补了传统模型的不足，可以对气动扰动、摩擦和电池压降等复杂现象进行描述，对基于模型的控制效果有显著提升。

(3) 基于机器学习的误差建模技术在机器人领域的应用

近三年机器人领域已经有很多研究应用了这种基于机器学习对误差进行建模的范式，明显提升了控制效果。文献^[38]为赛车设计了一种可以在线学习的数据驱动模型预测控制器，使用高斯过程学习系统模型的误差，在确保安全的前提下将每一圈的行驶时间减少了 10%。文献^[39]将高斯过程融合模型预测控制的思路总结为一种规范方法，并给出了基于高斯过程的噪声传递关系，在汽车轨迹跟踪的控制效果和安全性上都有较大提升。文献^[40]提出 GPMPc 方法，使用高斯过程和扩展卡尔曼滤波共同对机械臂末端的模型误差进行在线估计，最后结合 MPC 进行轨迹跟踪控制，显著提升了跟踪控制的精度。在空中机器人领域，文献^[41]使用一组仿生气流传感器感知空气中的气流，并采用长短期记忆网络 (LSTM) 对四旋翼所受的空气阻力进行建模。文献^[42]使用数据驱动模型结合经典模型，以离线训练的方式使用高斯过程对误差进行建模以生成补偿控制信号，提升一架全向飞行机器人的姿态控制效果。具体到四旋翼轨迹跟踪问题，文献^[43]使用高斯过程对无人机高速大机动飞行时机体产生的气动阻力进行建模，并把模型融入 MPC 架构中进行控制，减少了高速飞行时 70% 的轨迹跟踪误差。在此基础上，文献^[44]更进一步，将高斯过程更换为时域卷积编码器，使用桨叶元素理论 (Blade Element Theory) 改进传统四旋翼模型，最终跟踪误差比高斯过程模型又减少了 50%，并通过大量实验说明了误差预测的效果。综上，大量的研究已经说明这种传统模型组合机器学习模型范式的可行性。

然而，与传统的高斯噪声相比，流体的动态特性更为复杂，一般使用纳维-斯托克斯微分方程进行描述，需要更独特的机器学习方法进行拟合。文献^[45]使用相等不变网络 (Equivariant Networks) 和权重相等 (Weight Symmetry) 将对称性融入进深度动态模型，大大提升了流体预测的泛化性。针对无人机飞行过程中产生的一系列气动现象，加州理工的 Soon-Jo Chung 教授团队近三年发表了一系列使用神经网络进行气动拟合的研究。文献^[46]提出 Neural Lander，使用深度神经网络拟合地面效应和飞行阻力，并在训练过程

中加入谱正则 (Spectral Normalization, SN) 方法确保利普希茨 (Lipschitz) 条件, 并使用反馈线性化原理实现了新模型的稳定控制。文章用降落实验和绕桌飞行实验验证了深度神经网络对地面效应的拟合效果和对未知数据的泛化性。文献^[47]提出 Neural-Swarm, 首次使用神经网络对同构多旋翼在接近飞行时受到的气动力进行建模。文章将一个标准模型与一个置换不变 (Permutation-Invariant) 深度神经网络组合, 使用谱正则方法确保利普希茨条件, 同时采用 Deep Sets 思想将气动扰动拟合能力拓展到多机, 即将其他飞机的影响先用一个神经网络表征, 求和后经过另外一个神经网络以得到对本机的影响。文献^[48]提出 Neural-Swarm2, 对异构飞机间的气动扰动进行考虑, 同时基于改进模型设计了运动规划算法。

总的来说, 这一系列研究为使用神经网络拟合气动扰动创造了网络结构、训练流程等方面的研究范式。然而, 其网络结构可以进一步提升, 训练过程可以从离线改为在线, 控制器也可以选用模型预测控制以提前预判扰动的到来, 增加对扰动的响应速度。

1.2.3 模型预测控制方法研究现状

在控制学科的发展历程中, 模型预测控制 (Model Predict Control, MPC) 方法是一种较为年轻的控制技术, 如下图所示。随着计算机性能的提升和数值优化方法的成熟, 模型预测控制方法兴起于上个世纪 80 年代, 并被广泛应用于工业领域的过程控制中。模型预测控制是一种控制思想, 通过模型预测、反馈校正和滚动优化更新控制律。本小节前半部分逐次介绍 MPC 的概念和发展历程; 后半部分主要介绍 MPC 的最新分支——数据驱动 MPC 的优点、分类和应用现状。

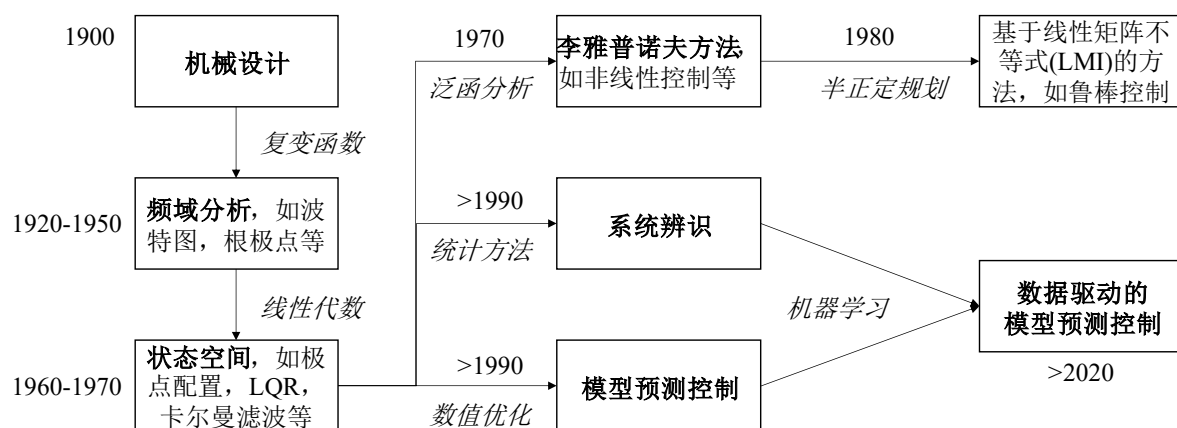


图 6 控制学科发展历程^[49]

(1) 模型预测控制 (MPC) 概况

MPC 所要解决的是一个在有限时间步长内的优化问题。与传统的最优控制方法 LQR 相比, LQR 是在控制开始针对全时间周期进行的优化, 提前计算好全部控制序列并进行执行; 而 MPC 是在每个时段都根据当前的状态重新优化, 只执行第一步动作。这给 MPC 带来更大的计算量, 但可以更好地处理模型误差, 也可以将约束包含在优化目标中, 有效处理带约束的优化问题^[50]。

经过近四十年的发展, MPC 发展出了线性 MPC、动态 MPC、线性时变 MPC、分布式 MPC、随机 MPC 和数据驱动 MPC 等种种方法, 已经逐渐在各行各业得到应用, 其发展史如下表。

表 1 MPC 发展历程^[49]

应用领域	MPC 方法	时间
过程控制	线性 MPC	1970-2000
汽车控制	显式 MPC, 动态 MPC	2001-2010
航空航天系统	线性时变 MPC	>2005
信息和通信技术	分布式 MPC	>2005
能源, 金融, 水	随机 MPC	>2010
工业生产	MPC 嵌入式优化求解器	>2010
机器学习	数据驱动 MPC	今天

线性 MPC 可转化为一个凸优化问题, 存在全局最优解, 且控制学界已经发展出较完备的稳定性分析理论。非线性 MPC 由于优化目标或者状态方程是一个非凸函数, 只能转化为非凸优化问题, 计算量较大, 难以求得全局最优解, 也没有完备的理论证明。然而, 由于“热启动”、“多触发”、“实时迭代技术”等工程技巧的出现, 非线性 MPC 已经可以在动态系统中开展应用。除去模型的线性特性, 上述 MPC 没有考虑模型中的不确定性, 而 2010 年出现的随机 MPC 将模型的随机性考虑在内, 进一步提高了 MPC 的适用性。最后, 近几年机器学习的快速进步推动了 MPC 的最新类别——数据驱动 MPC

的发展。

(2) 数据驱动 MPC 的研究现状

从 MPC 的角度来说,将机器学习和 MPC 结合具有如下优点:首先,机器学习领域已经有大量算法用于从数据中提取数学模型,如回归、分类等方法;其次,人工智能、统计和优化等领域有较多的数学基础理论,为 MPC 的发展提供一定的理论基础;再次,尽管数据驱动 MPC 需要解决一个非凸优化问题,往往无法获得理论上的全局最优解,但该方法已经大量应用于实际问题,极大提升了控制效果;最后,机器学习有 PyTorch、TensorFlow 等大量开源工具可供选择,便于数据驱动 MPC 的开发。从机器学习的角度来说,MPC 作为带约束控制的主要代表,可以为机器学习获得的模型提供安全性约束,推动基于学习的技术在高风险行业的应用。从 MPC 的角度来说,机器学习大大丰富了模型的灵活性,有利于 MPC 在复杂问题上的应用。数据驱动 MPC 利用了两种技术的优势,因此近几年在机器人等领域得到了充分关注。

关于数据驱动 MPC 的研究现状主要参考了综述文献^[50]。MPC 与机器学习的结合方式主要有三类:1)使用机器学习技术改进模型的准确性;2)使用机器学习技术推断 MPC 的参数;3)使用 MPC 为基于机器学习的方法提供安全性约束。第二类和第三类不是本文关注的重点,因此主要关注第一类别的应用。对于第一种类,根据学到的模型类别可以分为鲁棒模型和统计模型两类,如下图所示。

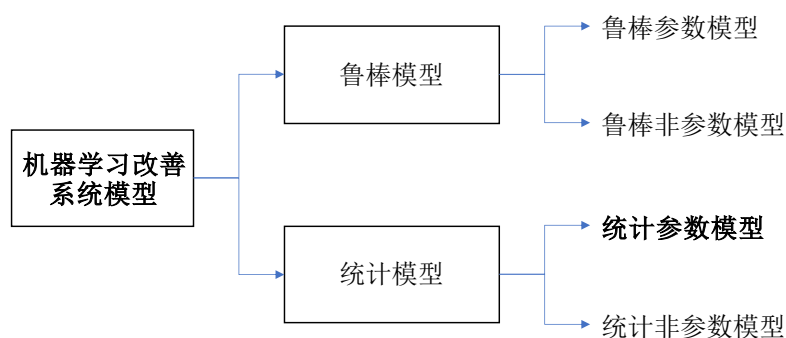


图 7 机器学习改善系统模型分类

对于鲁棒模型,MPC 中的约束对所有不确定元素和噪声都应满足约束条件。这其中又分为鲁棒参数模型和鲁棒非参数模型。对于神经网络拟合模型,是典型的非参数模型。而鲁棒非参数模型需要满足的数学条件是导数有界,因此在使用神经网络进行拟合的过程中,要着重满足导数有界条件,即 Lipschitz 条件,这样可参考鲁棒 MPC 的理论成果

进行分析。对于统计模型，需要借助模型不确定性的概率分布，将经典 MPC 的硬约束条件转变为一个概率约束条件。对于统计参数模型，主要使用贝叶斯方法对参数进行估计。对于统计非参数模型，主要采用高斯过程进行估计。高斯过程可以通过一个核函数将不确定性模型映射为某个超空间的高斯分布，目前已经被大量应用于模型误差建模，但参数或数据点较多时其计算量较大。

具体到应用领域，MPC 在无人机编队领域的应用较少，已有文献主要利用 MPC 的不等式约束来处理避障问题^[51]。对于无人机单体控制和 MPC 的结合，苏黎世大学的 Davide Scaramuzza 教授团队进行了一系列深入的研究。文献^[43]和^[44]使用机器学习方法来改进模型准确性，最终融合到 MPC 框架中以提升高速大机动飞行状态下的轨迹跟踪精度。文献^[52]使用自适应非线性 MPC 方法在线估计模型的不确定性，将估计结果迅速补偿到控制回路中，在大风、未知载荷和大过载的飞行状况下进行测试，验证了自适应非线性 MPC 算法在有误差模型下的控制效果。文献^[53]系统研究了两种四旋翼常用控制器，微分平坦（Differential-Flatness Based）控制器和 MPC 控制器在高速飞行下的轨迹跟踪控制性能。高速飞行对过载有更高要求，较易出现电机输入饱和现象，影响控制效果。MPC 的约束处理能力可以最大程度避免电机输入饱和，在物理不可实现的轨迹上有更小的轨迹跟踪误差。

总的来说，得益于机器学习领域的快速发展，数据驱动的 MPC 算法近几年得到较多关注。将前述结合深度学习气动扰动建模的组合模型包含在模型预测控制的架构内，既可以利用神经网络的拟合能力，又可以使用 MPC 处理输入饱和问题，且这种结合可以借用鲁棒 MPC 的一些结论进行理论分析，这些技术共同为实现四旋翼密集编队快速飞行提供了可行的途径。

1.3 解决的关键问题

本文主要解决以下问题与难点：

(1) 无人机集群间存在的下洗气流非线性程度高、变化迅速，很难用数理方法进行建模。如果采用流体力学中常用的计算流体力学仿真，又会消耗大量时间，难以融入进控制闭环。如何从真实机体上采集数据，并采用深度神经网络拟合下洗气流扰动，同时避免神经网络输出不可解释性对飞行安全的影响，是本文解决的关键技术问题。

(2) 针对单无人机的非线性模型预测控制轨迹跟踪算法已经得到较为广泛的研究。如何将上述针对单无人机的算法拓展到无人机编队领域，同时考虑无人机间存在的下洗气流效应以实现密集编队跟踪任务，是本文解决的关键技术问题。

(3) 在算法的验证上，仿真和实物存在较大差异。如何通过参数辨识等方法缩小仿真环境与实物个体的差异，从而将算法实时部署在机载计算机上进行密集编队飞行，是本文解决的工程难点。

1.4 论文主要工作及内容安排

本文以四旋翼无人机及其集群为研究对象，以下洗气流扰动下的密集编队跟踪为研究目标，重点围绕无人机间的下洗气流建模、传统方法对下洗气流的产生缺乏预见性等问题，对四旋翼无人机的系统建模与参数辨识、下洗气流扰动建模、单无人机轨迹跟踪控制、多无人机密集编队跟踪控制技术进行了研究。本文主干内容共分六章，主要章节安排及各章间关系如下图所示。

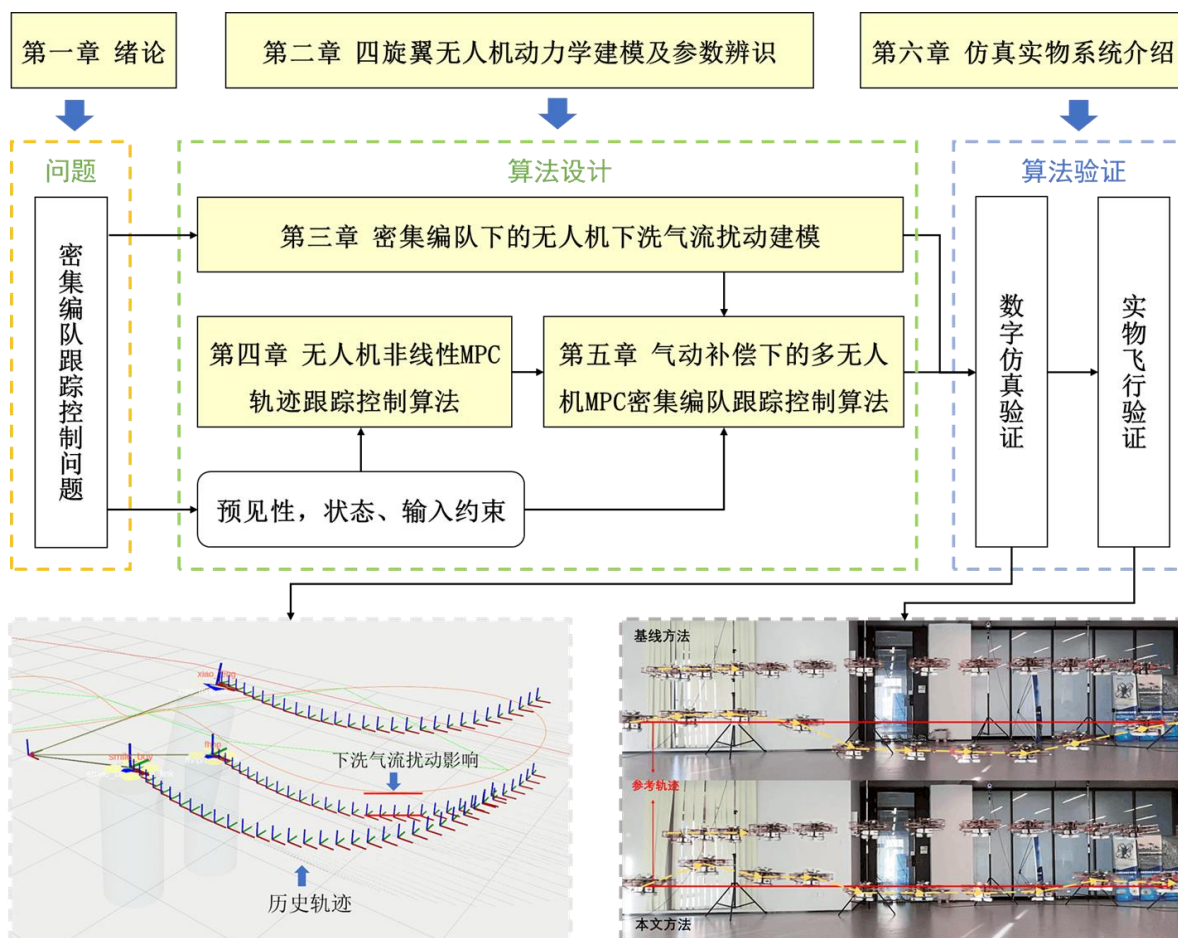


图 8 本文各章节间关系

第一章阐述了本文的研究背景、研究意义和国内外研究现状，分别对多无人机集群系统研究现状、气动扰动建模研究现状和模型预测控制研究现状进行调研，给出本文拟解决的关键问题，最后给出本文的内容安排。

第二章首先介绍全文采用的坐标系及变换关系，接着对四旋翼进行详细的动力学建模，同时给出角速度控制器和油门推力映射的模型。最后详细介绍对实物无人机进行参数辨识与数据分析的全过程，获得完整系统参数。本章建立的模型与辨识的参数将会用于第三、四、五章设计扰动估计与控制算法，同时也会用于第六章构建仿真环境。

第三章研究了多无人机机间下洗气流扰动估计问题。首先建立了一种简易的下洗气流模型用于仿真环境。接着设计神经网络下洗气流扰动观测器，并进行双机实物飞行，基于网络的输入输出收集数据。然后训练神经网络，并采用谱归一化方法提高网络的鲁棒性。之后测试神经网络，确认网络输出的安全性和在数据集上的有效性。最后采用线性叠加拓展到多无人机场景。

第四章研究了针对单无人机的非线性 MPC 轨迹跟踪控制问题。首先实现 Minimum Snap 轨迹生成算法，并利用四旋翼的微分平坦性质将其转换为动力学可行的全状态轨迹，以作为轨迹跟踪的控制目标。接着设计非线性 MPC 轨迹跟踪控制器，包括模型选择、优化过程和运算加速技巧。同时设计一种悬停油门观测器用于油门值的转化。最后进行仿真和实物实验验证算法效果。

第五章研究了存在下洗气流扰动的多无人机密集编队跟踪控制问题。首先将第三章设计的扰动观测器与第四章结合，设计气动补偿下的无人机轨迹跟踪控制算法。接着将第四章设计的控制器拓展到多无人机场景，设计非线性 MPC 编队跟踪控制算法。然后结合第三章设计的扰动观测器，设计气动补偿下的非线性 MPC 密集编队跟踪控制算法。最后进行仿真和实物实验验证算法效果。

第六章介绍了为验证上述算法开发的多无人机仿真环境与实物系统。首先介绍无人机软件系统，接着介绍基于 ROS 的仿真平台系统架构，最后介绍为实物实验搭建的多无人机硬件系统。所开发的仿真平台与实物平台有效支撑了本文算法的验证。

最后总结了本文的研究内容与成果，指出研究的不足以及可进一步改进的方向。

第二章 四旋翼无人机动力学建模及参数辨识

本章主要包括符号系统介绍、四旋翼动力学建模、PID 角速度控制器的实现、参数辨识等内容。角速度控制器本属于控制范畴，但因本文直接采用底层飞控内部的角速度控制器，故将其放在本章进行介绍。后续章节的控制算法将以角速度作为控制指令，这也与实际系统中的控制框架一致。本章节建立的模型与辨识的参数被用于开发多无人机仿真环境，同时也被用于设计后文的模型预测控制算法。

2.1 常用坐标系及其转换

本文采用的坐标系如下图所示，包括惯性坐标系 I 和机体坐标系 B 。本文选用东北天 (East, North, and Up, ENU) 惯性坐标系和前左上 (Front, Left, and Up, FLU) 机体坐标系以与机器人操作系统 (Robot Operating System, ROS) 保持一致，以便于通过 MAVROS 等工具包与底层飞控进行通信。在上述符号体系下，一个在 I 系下的向量将被记录为 ${}^I \mathbf{p}$ ，从 B 系向 I 系的旋转将被记录为 ${}^I \mathbf{R}$ (旋转矩阵) 或 ${}^I \mathbf{q}$ (姿态四元数)。图中采用 α 表示机架张开的角度， L 表示从质心到电机的距离，电机的标号顺序与旋转方向与 PX4 飞控的默认定义保持一致。

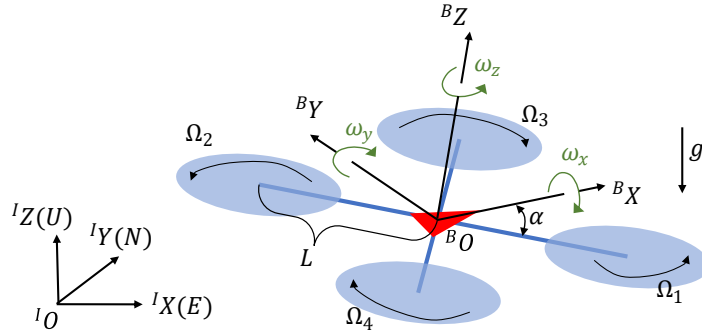


图 9 坐标系定义图

本文采用四元数表示飞机的姿态，为与 ROS 保持一致，采用 $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T \in \mathbb{H}$ 表示在汉密尔顿传统 (Hamilton-Convention^[54]) 下的四元数，采用 $\mathbf{q}^* = [q_w \ -q_x \ -q_y \ -q_z]^T$ 表示四元数的共轭，使用 \circ 符号表示四元数相乘。姿态四元数属于单位四元数，因而其模长 $\|\mathbf{q}\| = 1$ 且有 $\mathbf{q}^{-1} = \mathbf{q}^*$ 。本文使用 $\mathcal{V}(\cdot)$ 表示四元数的向量部分 $\mathcal{V}(\mathbf{q}) := [q_x \ q_y \ q_z]^T, \mathbb{H} \rightarrow \mathbb{R}^3$ ，使用 $\mathcal{V}^*(\cdot)$ 表示从位置向量到四元数的反向映射

$\mathcal{V}^*(\mathbf{p}) := [0 \ \mathbf{p}]^T, \mathbb{R}^3 \rightarrow \mathbb{H}$, 那么从坐标系 B 向坐标系 I 的 SE3 变换可以被表示为 ${}^I \mathbf{p} = \mathcal{V}({}^I_B \mathbf{q} \circ \mathcal{V}^*({}^B \mathbf{p}) \circ {}^I_B \mathbf{q}^*) + {}^I \mathbf{p}_{Bo} = {}^I_B \mathbf{R}(\mathbf{q})^B \mathbf{p} + {}^I \mathbf{p}_{Bo}$, 其中 ${}^I \mathbf{p}_{Bo}$ 表示 B 系原点在 I 系中的位置, $\mathbf{R}(\mathbf{q})$ 是从四元数计算得到的旋转矩阵:

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1 - 2q_y^2 - 2q_z^2 & 2q_x q_y - 2q_w q_z & 2q_x q_z + 2q_w q_y \\ 2q_x q_y + 2q_w q_z & 1 - 2q_x^2 - 2q_z^2 & 2q_y q_z - 2q_w q_x \\ 2q_x q_z - 2q_w q_y & 2q_y q_z + 2q_w q_x & 1 - 2q_x^2 - 2q_y^2 \end{bmatrix}. \quad (2.1)$$

虽然在状态方程和控制算法中均采用四元数表示姿态, 但本文也会生成欧拉角姿态信息以便于算法调试。欧拉角采用 Z-Y-X 顺序进行表示, 即先沿 Z 轴转动 ψ 角, 再沿新坐标系的 Y 轴转动 θ 角, 最后沿新坐标系的 X 轴转动 ϕ 角。注意偏航角 ψ 和滚转角 ϕ 的幅度都是 $[-\pi, \pi]$, 而俯仰角 θ 的幅度是 $[-\pi/2, \pi/2]$ 。四元数转换欧拉角的公式为:

$$\begin{cases} \phi = \operatorname{atan} 2\left(2(q_w q_x + q_y q_z), (q_w^2 + q_z^2 - q_x^2 - q_y^2)\right) \\ \theta = \operatorname{asin}\left(2(q_w q_y - q_x q_z)\right) \\ \psi = \operatorname{atan} 2\left(2(q_w q_z + q_x q_y), (q_w^2 + q_x^2 - q_y^2 - q_z^2)\right) \end{cases}. \quad (2.2)$$

2.2 四旋翼无人机动力学模型

四旋翼无人机的动力学模型指的是机体状态随时间与控制输入变化的动态过程。建模完成后, 输入是四个电机的实际转速共 4 个变量, 状态是机体的位置、速度、四元数姿态、角速度, 共 13 个变量 (电机模型不在状态方程中考虑)。

2.2.1 六自由度刚体动力学

本文假设坐标系 B 的原点位于无人机的质心 (Center of Gravity, CoG), 同时一个飞机的四个旋翼放置在坐标系 B 的 XY 平面上。为与动作捕捉装置等设备的坐标系和操纵手的习惯保持一致, 本文选用惯性系下的速度 ${}^I \mathbf{v}$ 作为状态量。六自由度刚体的运动可以分为质心运动 (平动) 和绕质心运动 (转动), 前者为位置及其导数的变化, 后者为姿态及其导数的变化。

首先介绍质心运动 (平动)。根据速度的定义可以得到运动学公式为

$${}^I \dot{\mathbf{p}} = {}^I \mathbf{v}, \quad (2.3)$$

展开为 $[\dot{p}_e \ \dot{p}_n \ \dot{p}_u]^T = [v_x \ v_y \ v_z]^T$; 根据牛顿第二定律可以得到动力学公式为

$${}^I \dot{\mathbf{v}} = \left({}^I \mathbf{R}(\mathbf{q}) \cdot \left({}^B \mathbf{f}_u + {}^B \mathbf{f}_a \right) + {}^I \mathbf{f}_d \right) / m + {}^I \mathbf{g}, \quad (2.4)$$

其中 m 为机体质量, ${}^I \mathbf{g} = [0 \ 0 \ -g]^T$ 为重力向量, ${}^B \mathbf{f}_u$ 、 ${}^B \mathbf{f}_a$ 和 ${}^I \mathbf{f}_d$ 分别表示由螺旋桨造成的推力、由空气摩擦造成的空气阻力、和由下洗气流造成的扰动力。

接着介绍绕质心运动 (转动)。根据四元数刚体运动学可以得到姿态的更新公式为

$${}^B \dot{\mathbf{q}} = \frac{1}{2} \cdot {}^B \mathbf{q} \circ \mathcal{V}^* ({}^B \boldsymbol{\omega}), \quad (2.5)$$

展开为

$$\begin{bmatrix} \dot{q}_w \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \cdot \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}, \quad (2.6)$$

其中 ${}^B \boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ 为在机体系下表示的角速度; 根据刚体动力学可以得到角速度的更新公式为

$${}^B \dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} \cdot \left(-{}^B \boldsymbol{\omega} \times (\mathbf{I} \cdot {}^B \boldsymbol{\omega}) + {}^B \boldsymbol{\tau}_u + {}^B \boldsymbol{\tau}_d \right), \quad (2.7)$$

其中 \mathbf{I} 表示惯性矩阵, ${}^B \boldsymbol{\tau}_u$ 和 ${}^B \boldsymbol{\tau}_d$ 分别表示由螺旋桨造成的力矩和由下洗气流造成的扰动力矩。我们假设四旋翼无人机是三轴对称的, 则惯性矩阵可简化为对角阵的形式 $\mathbf{I} = \text{diag}(I_{xx}, I_{yy}, I_{zz})$ 。

力和力矩的具体公式将在本节后两条进行介绍。

2.2.2 旋翼动力系统建模

无人机的动力系统包括两个部分: 旋翼模型和电机模型。建模完成后的输入是四个电机的转速指令 $\Omega_{\text{cmd},i} \ i=1,2,3,4$, 输出是 ${}^B \mathbf{f}_u$ 和 ${}^B \boldsymbol{\tau}_u$ 。

旋翼模型构建的是从电机实际转速到合力、合力矩的关系, 除由旋翼旋转带来的力和力矩, 还应包括由旋翼转动惯量对系统的影响、旋翼的陀螺效应^[53]、和四旋翼高速飞行时的桨叶变形产生的影响^[44]。由于场地限制, 本文所控制的四旋翼飞行速度较慢, 姿态变化较缓和, 因而建模时忽略旋翼转动惯量、旋翼陀螺效应和桨叶变形的影响, 采用二次形式去拟合从每个旋翼 i 产生的力和力矩, 即为:

$$f_i = k_f \cdot \Omega_i^2, \quad \tau_i = k_\tau \cdot \Omega_i^2. \quad (2.8)$$

假设一个飞机的四个旋翼放置在坐标系 B 的 XY 平面上, 且电机产生的推力与 XY 平面

垂直，则有 ${}^B \mathbf{f}_u = [0 \ 0 \ f_c]^T$ 和 ${}^B \boldsymbol{\tau}_u = [\tau_x \ \tau_y \ \tau_z]^T$ ，那么每个电机推力的关系为

$$\begin{bmatrix} f_c & \tau_x & \tau_y & \tau_z \end{bmatrix}^T = \mathbf{G} \cdot [f_1 \ f_2 \ f_3 \ f_4]^T, \quad (2.9)$$

其中

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -L \sin \alpha & L \sin \alpha & L \sin \alpha & -L \sin \alpha \\ -L \cos \alpha & L \cos \alpha & -L \cos \alpha & L \cos \alpha \\ -k_q/k_t & -k_q/k_t & k_q/k_t & k_q/k_t \end{bmatrix}, \quad (2.10)$$

这里 α 表示机架张开的角度， L 表示从质心到电机的距离。注意控制分配矩阵 \mathbf{G} 与四旋翼坐标系、旋翼顺序、旋翼旋转方向的定义密切相关。

电机模型建立的是从电机期望转速到电机实际转速的关系，根据文献^[55]，无刷直流电机的动态过程可以简化为一阶惯性环节，具体建模如下

$$\Omega_i = \frac{1}{t_m \cdot s + 1} \Omega_{\text{cmd},i} \quad (i = 1, 2, 3, 4), \quad (2.11)$$

其中 t_m 为电机响应的时间常数。若选定状态为 Ω_i ，则上述传递函数模型可以转化为状态空间模型

$$\dot{\Omega}_i = \frac{1}{t_m} \cdot (\Omega_{\text{cmd},i} - \Omega_i) \quad (i = 1, 2, 3, 4), \quad \text{clip}(\Omega_i, [\Omega_{\text{MIN}}, \Omega_{\text{MAX}}]), \quad (2.12)$$

其中 $A = -1/t_m$ ， $B = 1/t_m$ ， $C = 1$ ， $D = 0$ 。

至此便完成了对无人机动力系统的建模。

2.2.3 旋翼气动阻力建模

多旋翼无人机在飞行中存在四种气动效应：风吹影响、气动阻力、地面效应、下洗气流。本文主要研究的是多无人机间下洗气流的影响，因而将下洗气流建模放到第三章进行介绍。在仿真中假设无人机处于无风环境，且飞行高度距离地面较高，因而忽略风吹和地面效应的影响，此处主要介绍单无人机在飞行时存在的气动阻力。

无人机飞行时存在与空气摩擦产生的气动阻力 ${}^B \mathbf{f}_a$ 和阻力矩 ${}^B \boldsymbol{\tau}_a$ 。本文参考文献^[56]对无人机飞行时的气动阻力进行建模。气动阻力与机体系下的飞行速度有关，因此要先得到某一时刻机体系下的速度 ${}^B \mathbf{v} = [u \ v \ w]^T = {}^B \mathbf{R}(\mathbf{q}) \cdot {}^I \mathbf{v}$ ，进而得到气动阻力

$${}^B \mathbf{f}_a = \begin{bmatrix} -k_{d,x} u & -k_{d,y} v & -k_{d,z} w + k_h (u^2 + v^2) \end{bmatrix}^T. \quad (2.13)$$

由于飞行姿态变化较为缓和，忽略气动阻力矩的影响。

2.2.4 全量动力学模型

将 2.2.2 条建立的动力模型和 2.2.3 条建立的阻力模型代入到 2.2.1 条中的四旋翼无人机刚体动力学模型中，可以汇总得到如下的四旋翼无人机六自由度全量动力学模型：

$$\begin{cases} {}^I \dot{\mathbf{p}} = {}^I \mathbf{v} \\ {}^I \dot{\mathbf{v}} = \left({}^I \mathbf{R}(\mathbf{q}) \cdot ({}^B \mathbf{f}_u + {}^B \mathbf{f}_a) + {}^I \mathbf{f}_d \right) / m + {}^I \mathbf{g} \\ {}^B \dot{\mathbf{q}} = \frac{1}{2} \cdot {}^B \mathbf{q} \circ \mathcal{V}^* ({}^B \boldsymbol{\omega}) \\ {}^B \dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} \cdot \left(-{}^B \boldsymbol{\omega} \times (\mathbf{I} \cdot {}^B \boldsymbol{\omega}) + {}^B \boldsymbol{\tau}_u + {}^B \boldsymbol{\tau}_d \right) \end{cases}, \quad (2.14)$$

动力系统部分为

$$\begin{cases} \dot{\boldsymbol{\Omega}} = \frac{1}{t_m} (\boldsymbol{\Omega}_{\text{cmd}} - \boldsymbol{\Omega}), \text{clip}(\boldsymbol{\Omega}_i, [\Omega_{\text{MIN}}, \Omega_{\text{MAX}}]) \\ f_i = k_t \cdot \Omega_i^2, i = 1, 2, 3, 4 \\ \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -L \sin \alpha & L \sin \alpha & L \sin \alpha & -L \sin \alpha \\ -L \cos \alpha & L \cos \alpha & -L \cos \alpha & L \cos \alpha \\ -k_q / k_t & -k_q / k_t & k_q / k_t & k_q / k_t \end{bmatrix}, \\ \begin{bmatrix} f_c & \tau_x & \tau_y & \tau_z \end{bmatrix}^T = \mathbf{G} \cdot \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^T \\ {}^B \mathbf{f}_u = \begin{bmatrix} 0 & 0 & f_c \end{bmatrix}^T \\ {}^B \boldsymbol{\tau}_u = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \end{cases}, \quad (2.15)$$

空气阻力部分为

$$\begin{cases} {}^B \mathbf{v} = \begin{bmatrix} u & v & w \end{bmatrix}^T = {}^I \mathbf{R}(\mathbf{q})^T \cdot {}^I \mathbf{v} \\ {}^B \mathbf{f}_a = \begin{bmatrix} -k_{d,x} u & -k_{d,y} v & -k_{d,z} w + k_h (u^2 + v^2) \end{bmatrix}^T \end{cases} \quad (2.16)$$

接下来需要对上述连续模型进行离散化以进行计算机仿真。对于复杂模型，可以采用数值方法直接进行仿真；对于简单模型，可以采用 Z 变换获得时域下的解析解进行仿真。

上述六自由度模型(2.14)为复杂的多变量非线性常微分方程，适合采用数值方法进行仿真，这里采用固定步长四阶龙格-库塔法（4th Order Runge-Kutta Method, RK4）。给定方程 $y' = f(t, y)$ 与初值 $y(t_0) = y_0$ ，则下一时刻的状态为

$$y_{n+1} = y_n + \frac{t_s}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad (2.17)$$

其中 t_s 为仿真步长, $k_1 = f(t_n, y_n)$, $k_2 = f\left(t_n + \frac{t_s}{2}, y_n + \frac{t_s}{2}k_1\right)$, $k_3 = f\left(t_n + \frac{t_s}{2}, y_n + \frac{t_s}{2}k_2\right)$, $k_4 = f(t_n + t_s, y_n + t_s k_3)$ 。需要注意的是, 根据采样定理, 仿真步长应小于系统最小时间常数的二分之一。选定合适的仿真步长后, 动态系统的下一个值由现在的值加上仿真时间间隔 t_s 和一个估算的斜率的乘积所决定, 从而可由计算机进行数值仿真。

上述动力系统(2.15)存在的一阶系统为 $\dot{y} = -ay + au$ 形式的单变量线性常微分方程, 可直接求得采样情况下的解析解^[57]为

$$y(t+t_s) = e^{-at_s} y(t) + a \cdot \int_0^{t_s} e^{-a(t_s-\tau)} u(\tau) d\tau. \quad (2.18)$$

假定采样周期间 $u(t)$ 不变, 可以得到

$$y_{n+1} = e^{-at_s} y_n + (1 - e^{-at_s}) u_n. \quad (2.19)$$

因此, 电机模型可以采用该时域低通滤波器的形式进行离散仿真:

$$\boldsymbol{\Omega}_{n+1} = e^{-t_s/t_m} \boldsymbol{\Omega}_n + (1 - e^{-t_s/t_m}) \boldsymbol{\Omega}_{cmd,n}. \quad (2.20)$$

2.3 角速度控制器与油门推力映射

上文建立四旋翼无人机动力学模型后, 可以通过控制四个电机的转速来控制无人机飞行。然而, 一般四旋翼需要运行一个安装有实时操作系统的底层控制器来进行状态估计、电机控制等任务。本文选用的 PX4 飞行控制器运行在 Pixhawk 4 飞控上, 所支持的最底层控制指令是角速度指令, 其内部运行着一个与陀螺仪相同频率 (约 1 kHz) 的 PID 角速度控制器, 直接以陀螺仪量测信息作为反馈信号。为确保在仿真中运行的控制代码可以不加修改地迁移到实物飞机上, 本文在仿真中也集合了一个基于 PID 的角速度控制器, 使得控制接口与 PX4 保持一致。该角速度控制器的输入是期望的三轴角速度 ${}^B \boldsymbol{\omega}_{cmd} = [\omega_{x,cmd} \quad \omega_{y,cmd} \quad \omega_{z,cmd}]^T$ 和油门 *throttle*, 输出是期望的三轴力矩和合推力 $[\tau_{x,cmd} \quad \tau_{y,cmd} \quad \tau_{z,cmd} \quad f_{c,cmd}]^T$, 经逆控制分配得到四个电机的期望转速 $\boldsymbol{\Omega}_{cmd,i} \quad i=1,2,3,4$ 。本节首先介绍 PID 角速度控制器, 接着介绍油门换算, 最后介绍逆控制分配过程。

比例-积分-微分 (Proportion Integration Differentiation, PID) 控制是控制领域最为常用的反馈控制器, 将输出误差进行比例、积分和微分运算并求和得到控制信号, 对于二阶及以下线性系统可实现指数收敛。以前连续域的 PID 可通过放大电路实现, 现在大多

将 PID 离散化后通过计算机实现。本文所要处理的角速度到力矩的关系属于一阶系统，因而可以对 XYZ 三轴分别采用离散 PID 进行控制：

$$\begin{cases} \tau_{x,\text{cmd}} = \text{PID}(\omega_{x,\text{cmd}}, \omega_x) \\ \tau_{y,\text{cmd}} = \text{PID}(\omega_{y,\text{cmd}}, \omega_y) \\ \tau_{z,\text{cmd}} = \text{PID}(\omega_{z,\text{cmd}}, \omega_z) \end{cases} \quad (2.21)$$

现简要介绍离散 PID 算法流程^[57]。连续域的 PID 控制律可写为

$$u(t) = k_p e(t) + k_i \int_{-\infty}^t e(\tau) d\tau + k_d \frac{de}{dt}(t), \quad (2.22)$$

其中 $e(t) = y^c(t) - y(t)$ 是期望输出 $y^c(t)$ 和实际输出 $y(t)$ 的误差。对上式进行拉氏变换可得

$$U(s) = k_p E(s) + k_i \frac{E(s)}{s} + k_d s E(s). \quad (2.23)$$

因为纯微分项是非因果的，这里采用有限带宽积分器进行近似：

$$U(s) = k_p E(s) + k_i \frac{E(s)}{s} + k_d \frac{s}{\tau s + 1} E(s). \quad (2.24)$$

为将上述控制律转化到离散时间系统，可以采用双线性变换（Tustin 变换）进行离散化，将系统从拉普拉斯域转化为 Z 域。转化后的积分项与微分项分别变为

$$I(z) = \frac{t_{sc}}{2} \left(\frac{1+z^{-1}}{1-z^{-1}} \right) E(z), \quad (2.25)$$

$$D(z) = \frac{\left(\frac{2}{2\tau + t_{sc}} \right) (1-z^{-1})}{1 - \left(\frac{2\tau - t_{sc}}{2\tau + t_{sc}} \right) z^{-1}} E(z). \quad (2.26)$$

转化为时域后为

$$I[n] = I[n-1] + \frac{t_{sc}}{2} (E[n] + E[n-1]), \quad (2.27)$$

$$D[n] = \left(\frac{2\tau - t_{sc}}{2\tau + t_{sc}} \right) D[n-1] + \left(\frac{2}{2\tau + t_{sc}} \right) (E[n] - E[n-1]), \quad (2.28)$$

其中 t_{sc} 为采样周期（一般与控制周期相同）， τ 为微分项的低通滤波器增益。

除上述公式外，还应考虑积分项的抗饱和处理（Anti-Windup）和对控制输入的限幅处理。具体算法流程总结如下：

Input: y_c , y , k_p , k_i , k_d , limit , T_s , τ , $(y_c \text{ dot})$, $(y \text{ dot})$

Output: u_{sat}

初始化 $I[0]=0$, $D[0]=0$, $E[0]=0$

While 正在仿真, 第 n 步时:

$$E[n] = y_c - y$$

$$I[n] = I[n-1] + \frac{t_{sc}}{2} (E[n] + E[n-1]) \quad // \text{双线性 (Tustin) 变换}$$

If y_c_dot 和 y_dot 存在:

$$D[n] = y_c' - y'$$

Else:

$$D[n] = \frac{2\tau - t_{sc}}{2\tau + t_{sc}} D[n-1] + \frac{2}{2\tau + t_{sc}} (E[n] - E[n-1])$$

End If

$$u[n] = K_p \cdot E[n] + K_I \cdot I[n] + K_D \cdot D[n] \quad // \text{PID}$$

$$u_{sat}[n] = \text{sat}(u[n], \text{limit}) \quad // \text{饱和约束}$$

If $\|K_I\| > 0$:

$$I[n] \leftarrow I[n] + \frac{t_{sc}}{K_I} (u_{sat} - u) \quad // \text{积分器抗饱和}$$

End If

$$E[n-1] \leftarrow E[n]$$

End While

下面介绍油门的换算过程。为与 MAVROS 工具包的接口保持一致, 此部分输入的油门是归一化到 $[0,1]$ 的油门变量 $throttle$, 输出为合推力 $f_{c,cmd}$, 根据文献^[55]可知上述变量满足线性关系:

$$f_{c,cmd} = \max(k_{th} \cdot throttle + b_{th}, 0) \cdot 4 \quad (2.29)$$

在实物飞行时, 发现从油门到合力的映射是一个随电池电压改变的量, 当电池电压下降时, 相同油门对应的合推力也逐渐减小。本文假设每节锂电池的电压从满电 4.2V 线性降低到没电 3.6V, 降低的时间是飞机的续航时间, 同时假设飞机相同油门产生的推力与电压呈线性关系。则有

$$U(t) = \begin{cases} 4.2 - t \cdot \frac{4.2 - 3.6}{t_{all}} & t \leq t_{all} \\ 0 & t > t_{all} \end{cases}, \quad (2.30)$$

其中 t_{all} 表示飞机的续航时间。那么油门与合推力的关系变为

$$f_{c,\text{cmd}} = \max\left(\frac{U(t)}{4.2} \cdot (k_{th} \cdot throttle + b_{th}), 0\right) \cdot 4。 \quad (2.31)$$

现在我们已经得到三轴期望力矩和合推力，下面通过控制分配的逆过程得到

$$\begin{bmatrix} f_{1,\text{cmd}} & f_{2,\text{cmd}} & f_{3,\text{cmd}} & f_{4,\text{cmd}} \end{bmatrix}^T = \mathbf{G}^{-1} \cdot \begin{bmatrix} f_{c,\text{cmd}} & \tau_{x,\text{cmd}} & \tau_{y,\text{cmd}} & \tau_{z,\text{cmd}} \end{bmatrix}^T, \quad (2.32)$$

其中

$$\mathbf{G}^{-1} = \begin{bmatrix} \frac{1}{4} & -\frac{\text{Csc}(\alpha)}{4L} & -\frac{\text{Sec}(\alpha)}{4L} & -\frac{k_t}{4k_q} \\ \frac{1}{4} & \frac{\text{Csc}(\alpha)}{4L} & \frac{\text{Sec}(\alpha)}{4L} & -\frac{k_t}{4k_q} \\ \frac{1}{4} & \frac{\text{Csc}(\alpha)}{4L} & -\frac{\text{Sec}(\alpha)}{4L} & \frac{k_t}{4k_q} \\ \frac{1}{4} & -\frac{\text{Csc}(\alpha)}{4L} & \frac{\text{Sec}(\alpha)}{4L} & \frac{k_t}{4k_q} \end{bmatrix}。 \quad (2.33)$$

最后得到四个电机的期望转速为

$$\Omega_i = \sqrt{f_{i,\text{cmd}} / k_t}, \quad i = 1, 2, 3, 4。 \quad (2.34)$$

2.4 四旋翼无人机参数辨识

上文建立了四旋翼模型后，需要通过对实物飞机进行辨识以获得相应参数。本文进行辨识的四旋翼实物平台如下图所示，本节仅介绍参数辨识的过程，后续将在第六章对实物系统进行详细的介绍。



图 10 本文使用的实物无人机系统

四旋翼有两类参数需要进行辨识，一类是旋翼参数，另一类是惯性参数。本文采用六轴力传感器对旋翼参数进行辨识，辨识过程中给电机施加不同转速的控制信号，测量旋翼产生的力和力矩随转速的变化。同时采用双线摆法对转动惯量进行辨识。在本节最后总结辨识得到与仿真用到的全部参数。

2.4.1 参数辨识过程介绍

(1) 旋翼参数辨识

旋翼参数的辨识设备采用 ATI 公司的 NET-DELTA 六轴力传感器，具体型号为 SI-165-15，其量程如下表，可以满足旋翼对测试精度的要求。

表 2 六轴力传感器 SI-165-15 的量程

名称	F_x, F_y	F_z	T_x, T_y	T_z
量程	165 N	495 N	15 Nm	15 Nm
分辨率	1/32 N	1/16 N	1/528 Nm	1/528 Nm

辨识参数时，将一对斜对角的旋翼安装到机体上以考虑机身对旋翼气流的影响。采用 3D 打印加工四旋翼与六轴力传感器的连接支架，将连接支架与四旋翼底部的电池安装板使用螺丝相连。支架本身和安装四旋翼后的形态如下图所示。

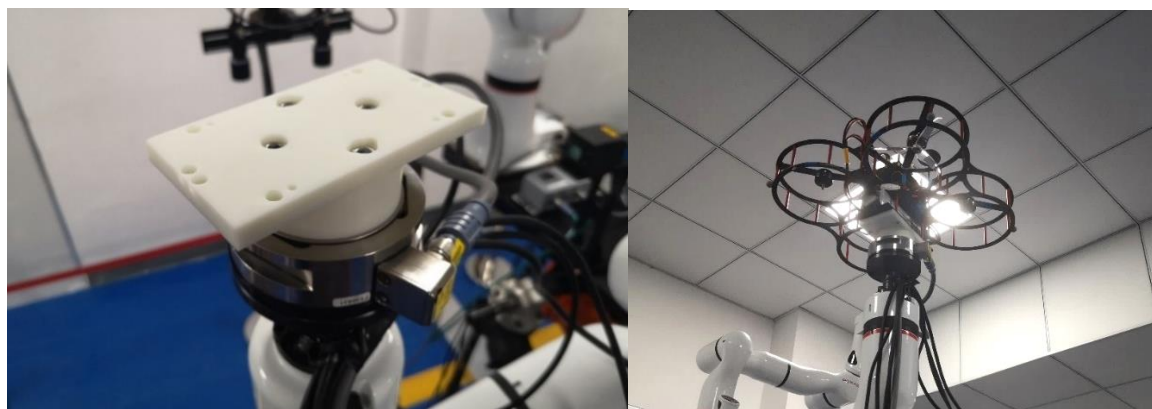


图 11 连接支架及连接后的测试状态

四旋翼安装完毕的电机参数测试环境如下图所示。该实验环境位于北航余杭创新研究院，六轴力传感器安装在机械臂末端，旁边计算机内有配套软件读取力传感器的量测信息。

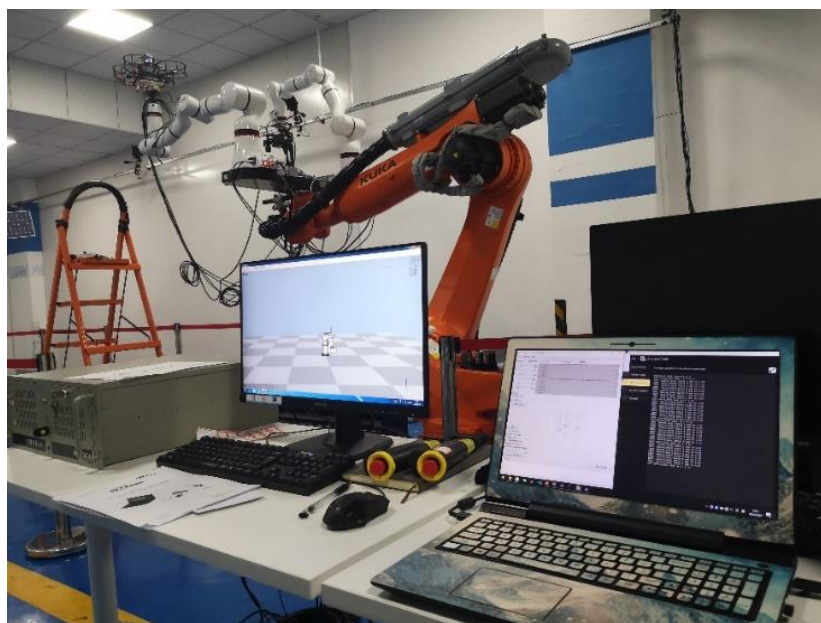


图 12 电机参数测试环境

辨识开始后,首先测试电机推力和力矩随油门的变化。从 QGroundControl 地面站向 PX4 发送电机测试指令,从 0%-100%按照 4%的区间等间隔发送,每次发送后等待 10 s,然后等待旋翼停转,再开启下一次测试。保存 ATI 传感器测得的力和力矩以待处理。接着测试电机转速随油门的变化,往螺旋桨上粘上反光条,从 QGroundControl 向 PX4 发送电机测试指令,测试螺旋桨转速并保存。后续我们检测了旋翼 DShot 协议测量的转速,发现与转速计测得的结果基本相同,因而后续改为采用 DShot 测量转速。上述测试均按照油门从低到高和从高到底的顺序测量两次。

(2) 惯性参数辨识设备

惯性参数包括机体质量与转动惯量。机体质量采用电子秤测量即可,而转动惯量的测量经查阅资料有三种方法。第一种方法是使用 CAD 软件进行计算,但这种方式无法考虑线材等装置的影响,计算误差较大。第二种方法如文献^[58]所述采用平行轴定理和单摆的方法进行测量,但这种方法需要安装光电码盘以测量单摆的角度。第三种方法采用双线摆法^[59]进行量测,只需要用两根线将机体吊起来记录摆动的周期即可(如下图所示),方案执行简单有效,因此本文采用这种方法测量转动惯量。辨识开始后,双手施加扭矩使得无人机以重力方向为转轴偏转一个较小的角度,之后松手让无人机开始自然摆动。记录自然摆动的周期,即可计算得到当前重力方向对应轴的转动惯量。

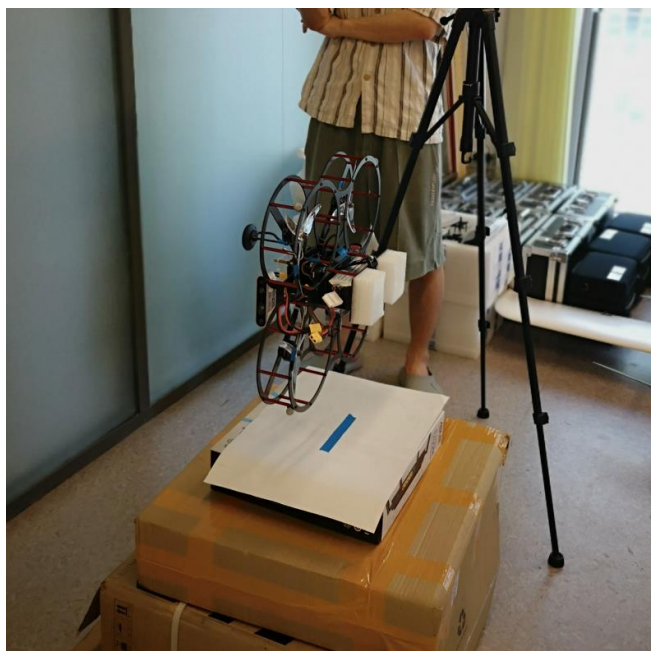


图 13 采用双线摆的转动惯量测量装置

2.4.2 旋翼系统数据分析

(1) 旋翼转速数据分析

改变油门，测量转速，原始数据请见附录。转速平方随油门的变化绘制如下图。从图中可以发现转速的平方与油门约为线性关系，但在 0~10%油门附近的线性度较差。从实际测量数据点可以得到油门为 0%时电机转速 Ω_{MIN} 约为 2600 RPM，油门为 100%时电机转速 Ω_{MAX} 约为 24000 RPM。

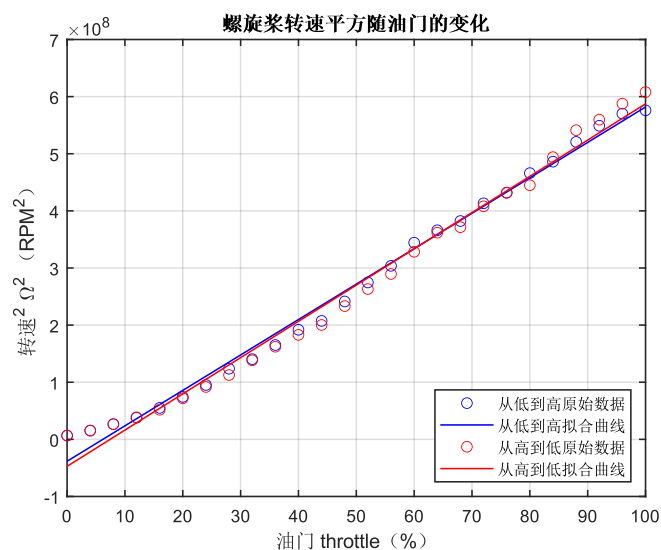


图 14 转速平方随油门的变化曲线

使用最小二乘法，得到从低到高的曲线为 $\Omega_i^2 = 6.2020 \times 10^8 \times throttle - 3.8354 \times 10^7$ ；从高

到低的曲线为 $\Omega_i^2 = 6.3458 \times 10^8 \times throttle - 4.7263 \times 10^7$ 。平均可以得到转速平方随油门的变化关系为

$$\Omega_i^2 = 6.2739 \times 10^8 \times throttle - 4.2809 \times 10^7。 \quad (2.35)$$

(2) 旋翼推力数据分析

给定油门下电机推力的测量数据如下图所示。

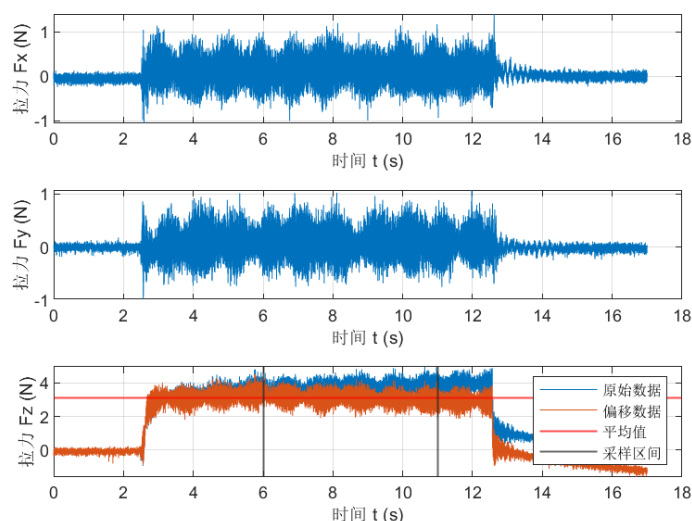


图 15 电机动力为 16% 下的无人机三轴受力

可以看出三轴受力的原始数据（蓝色曲线）存在一定的零漂。经过分析，造成此现象的原因应该是机体振动导致传感器温度升高所致。经观察，零漂近似为线性增加，因此加入补偿得到图中橙色曲线。除此以外，在部分油门下还发现了共振的现象，如下图所示：

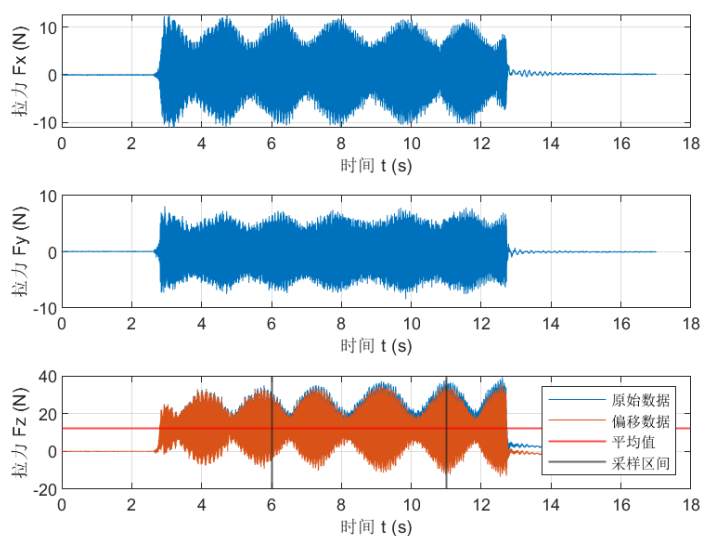


图 16 电机动力为 44% 下的无人机三轴受力

在油门等于 44%附近，机体的振动非常剧烈，峰峰值甚至超过了电机的推力，并且既有低频的基波又有高频的谐波。当油门超过这个值后，该现象迅速消失。这种振动现象可能与四旋翼与测试平台的固连有关，当飞机起飞后，并未观察到如此强的振动，因此暂不考虑这里的振动现象。

后续的推力测量数据均采用上述方法对 Z 轴拉力数据进行补偿，同时选定数据较为稳定的 5s 区间，取平均值作为该油门大小下的最终结果。记录所有油门下的平均值，同时使用转速数据作为横坐标，绘图如下：

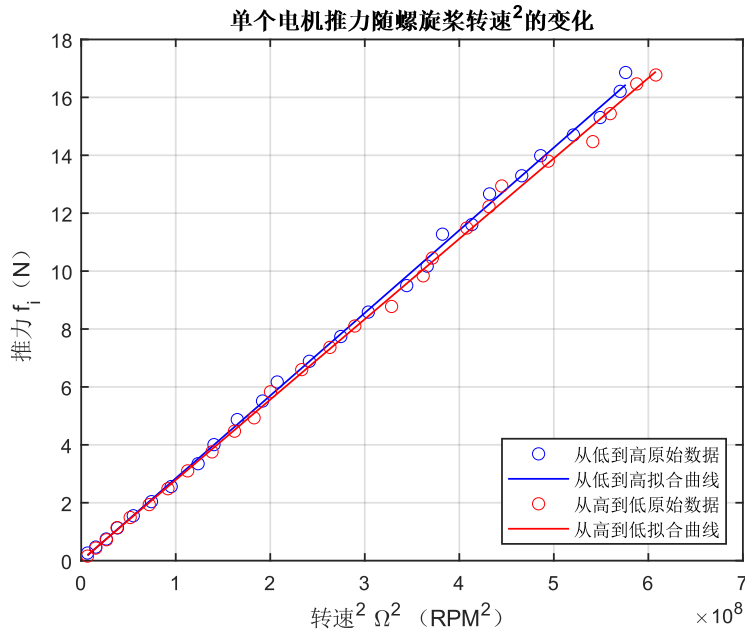


图 17 推力随转速平方的变化曲线

使用最小二乘法，可以得到从低到高的测试曲线为 $f_i = 2.8551 \times 10^{-8} \times \Omega_i^2 - 0.01096$ ；从高到低的测试曲线为 $f_i = 2.7765 \times 10^{-8} \times \Omega_i^2 + 0.00975$ 。两曲线取平均，最终获得旋翼推力随转速平方的变化公式为

$$f_i = 2.8158 \times 10^{-8} \times \Omega_i^2 - 0.00061, \quad (2.36)$$

从而获得旋翼推力系数为 $k_t = 2.8158 \times 10^{-8} \text{ N/RPM}^2 = 2.8158 \times 10^{-2} \text{ N/kRPM}^2$ 。将上述公式代入(2.35)可以得到推力与油门的关系

$$f_i = 17.6660 \times throttle - 1.2060, \quad (2.37)$$

故 $k_{th} = 17.6660$ ， $b_{th} = -1.2060$ 。这个公式在飞行时得到了悬停油门的验证，基本是准确的。当油门为 1 时，单个电机产生的最大推力为 16.46 N，则四个电机产生的最大推力

为 65.84 N，又量得飞机重量为 1.5344 kg，计算可得飞机的推重比为 4.37。然而考虑到电池电压衰减等因素，将电机的实际最大转速限制在 24000 RPM，此时的推重比为 4.31。

(3) 旋翼动态特性分析

下面使用 MATLAB 的系统辨识工具箱对电机的时间常数 t_m 进行辨识。由于测量的结果是电机推力的响应，而需要得到的结果是电机转速的响应，因此需要通过公式(2.36)对推力测量结果进行转换，转换中需要开方因而要舍去负数结果。根据文献^[55]，旋翼电机的响应曲线近似为一阶系统，因而采用 MATLAB 的系统辨识工具箱，采用 n4sid 函数设置辨识模型为 1 阶状态空间函数进行辨识。一阶系统的时间常数不会随着参考信号的幅值而改变，因而对低、中、高三档油门下的电机响应曲线进行辨识，最终时间常数为三者的均值。辨识结果如下图所示。

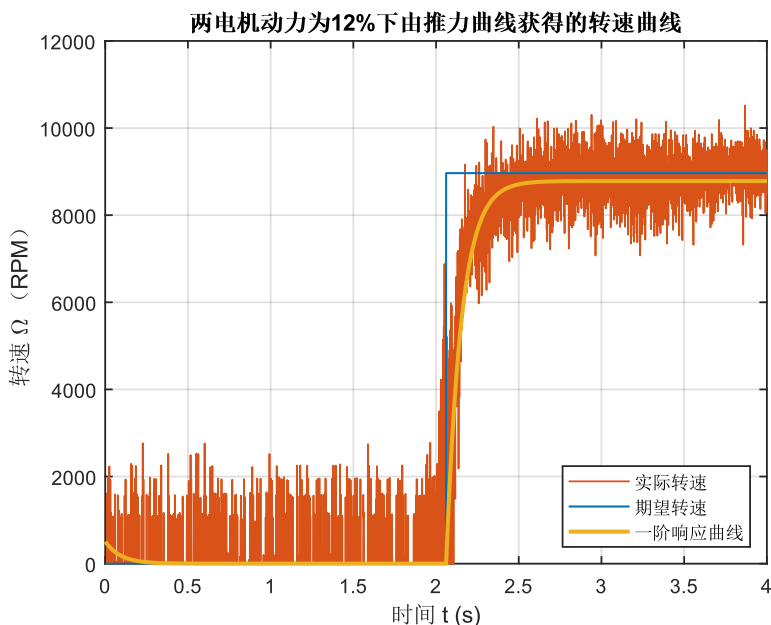


图 18 12%油门下由辨识得到的电机转速响应曲线

辨识后得到 $t_m = 1/10.84=0.092$ s，对原始数据拟合准确率为 82.84%。采用其他油门数据进行辨识，根据数据情况在系统辨识工具箱中设置频率范围为 [1, 200] rad/s 进行滤波处理，得到

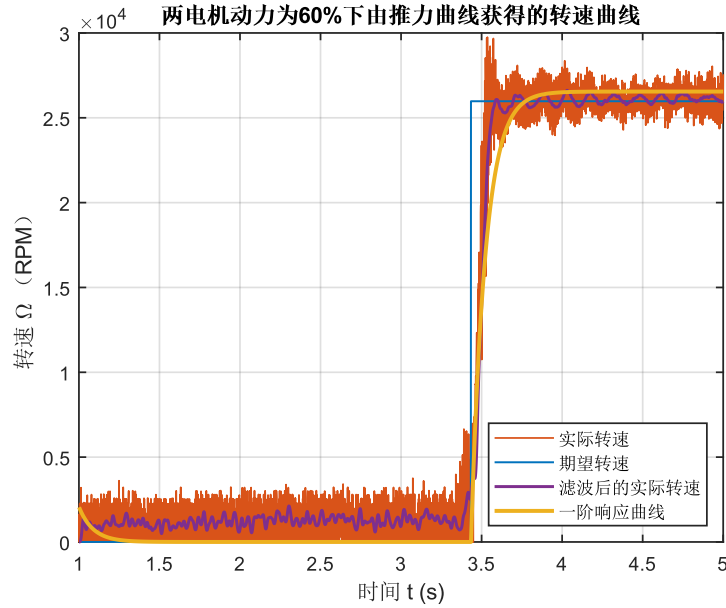


图 19 60%油门下由辨识得到的电机转速响应曲线

辨识后得到 $t_m = 1/11.33=0.088\text{s}$ ，对原始数据拟合准确率为 85.50%。

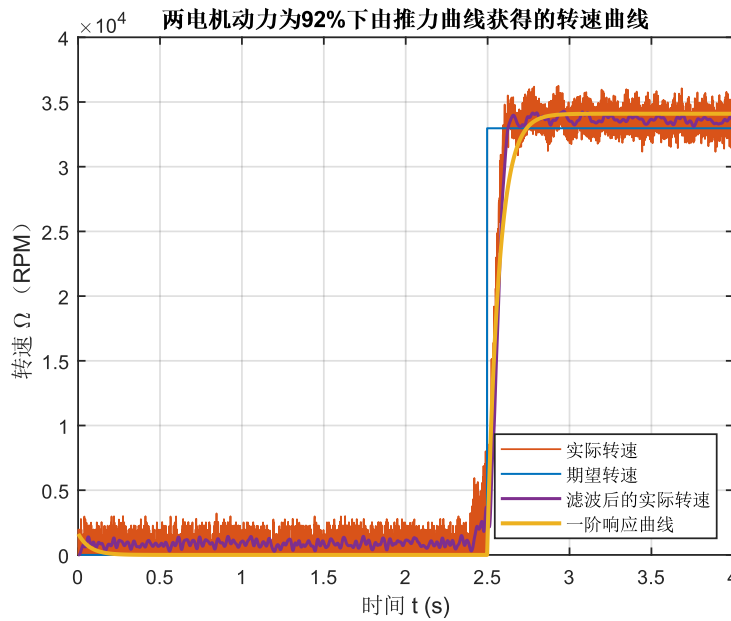


图 20 92%油门下由辨识得到的电机转速响应曲线

辨识后得到 $t_m = 1/14.13=0.071\text{ s}$ ，拟合准确率为 90.21%。将三次辨识结果取平均，可以得到最终的 $t_m = 0.084\text{ s}$ ，与文献^[55]中记录数值相近。实践中发现拟合结果与控制信号的输入时刻紧密相关，因而在辨识时最好同时记录输入与响应曲线。

(4) 旋翼力矩数据分析

绘制力矩数据如下图，可以看出力矩几乎没有零漂，因此可以直接处理。

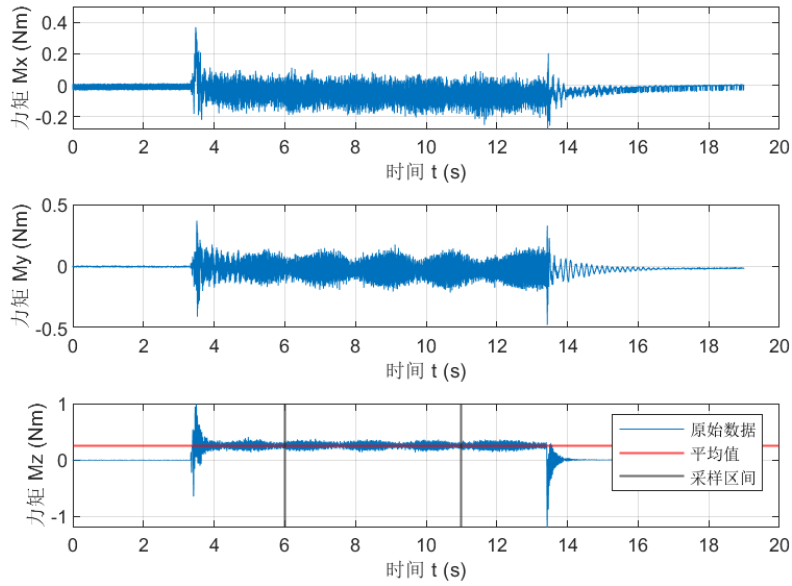


图 21 电机动力为 60% 下的无人机三轴所受力矩

按照测试旋翼推力的方法，对力矩随油门的变化进行数据处理，得到下图。

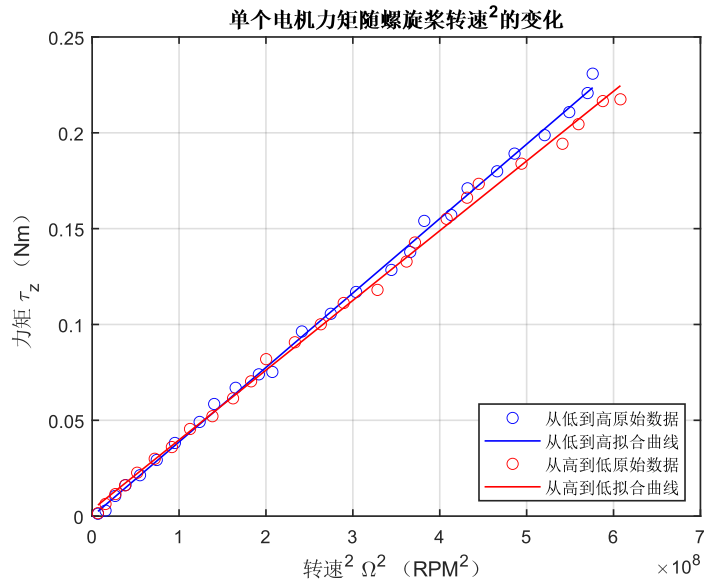


图 22 力矩随转速平方的变化曲线

同样使用最小二乘法，可以得到从低到高的测试曲线为 $\tau_z = 3.8833 \times 10^{-10} \times \Omega_i^2 - 0.00011$ ；从高到低的测试曲线为 $\tau_z = 3.6388 \times 10^{-10} \times \Omega_i^2 - 0.00337$ 。两曲线取平均，最终获得旋翼力矩随转速平方的变化公式为

$$\tau_z = 3.7611 \times 10^{-10} \times \Omega_i^2 - 0.00174, \quad (2.38)$$

从而获得旋翼力矩系数为 $k_q = 3.7611 \times 10^{-10} \text{ N/RPM}^2 = 3.7611 \times 10^{-4} \text{ N/kRPM}^2$ 。至此获得了无人机的全部旋翼参数。

2.4.3 惯性属性数据分析

假设物体质量为 m ，重力加速度为 g ，从上悬挂点到下悬挂点的距离（即绳长）为 L ，两线距离为 $D = 2r$ （如下图所示），旋转的周期为 T ，则当扰动角度 θ 很小时（ 10° 以内，适用小角度线性化），根据拉格朗日力学可以计算得到物体绕竖直向下轴的转动惯量^[59]为

$$I = \frac{mgD^2}{4L\omega^2} = \frac{mgD^2}{4L(2\pi f)^2} = \frac{mgD^2T^2}{16\pi^2L}。 \quad (2.39)$$

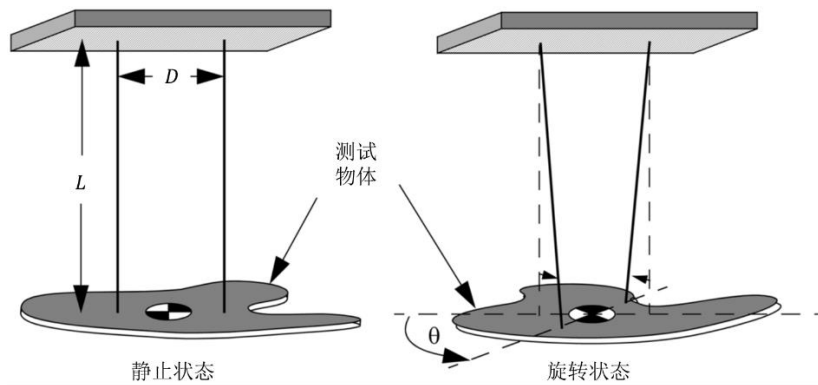


图 23 转动惯量测量装置示意图^[59]

忽略摩擦力等影响，扭摆的运动周期可看作不变。分别悬挂 XYZ 三轴，每一轴进行三组实验，每一组记录 50 次扭转运动的时间，具体时间数据请见附录。测试时的实验数据和根据(2.39)计算的转动惯量结果见下表。

表 3 无人机的转动惯量测试结果

	符号	数值	单位	符号	数值	单位
惯性参数	m	1.5344	kg	g	9.81	m/s^2
X 轴数据	D	0.365	m	L	2.06	m
	T	1.2361	s	I_{xx}	0.0094	$\text{kg} \cdot \text{m}^2$
Y 轴数据	D	0.365	m	L	2.06	m
	T	1.4757	s	I_{yy}	0.0134	$\text{kg} \cdot \text{m}^2$

	符号	数值	单位	符号	数值	单位
Z 轴数据	D	0.29	m	L	2.05	m
	T	1.9271	s	I_{zz}	0.0145	kg·m ²

至此获得了无人机的全部惯性参数。

2.4.4 模型及仿真参数汇总

对无人机辨识后的参数汇总如下表：

表 4 对无人机辨识的参数

参数符号	数值	单位
L	0.1372	m
α	45	deg
m	1.5344	kg
g	9.81	m/s ²
I_{xx}	0.0094	kg·m ²
I_{yy}	0.0134	kg·m ²
I_{zz}	0.0145	kg·m ²
t_m	0.084	s
k_{th}	17.666	N
b_{th}	-1.206	N
k_q	3.7611 e-4	N·m/kRPM ²
k_t	2.8158 e-2	N/kRPM ²
Ω_{MIN}	2.6	kRPM
Ω_{MAX}	24.0	kRPM
推力/重力 (TWR)	4.3100	—

参数符号	数值	单位
力矩/转动惯量 (TIR)	469.6976	—
飞行时间 t_{all}	705	s

上述参数将用于仿真环境和模型预测控制算法中，力求减少仿真与实物系统间的差异。

在仿真环境中，本文设置仿真步长 $t_s = 0.01s$ ，可视化步长 $t_{s,viz} = 0.05s$ 。仿真中采用的气动阻力参数主要参考论文^[53]进行设置，如下表所示。

表 5 仿真中的气动阻力模型参数设置

变量	数值	单位	变量	数值	单位
$k_{d,x}$	0.26	kg/s	$k_{d,y}$	0.28	kg/s
$k_{d,z}$	0.42	kg/s	k_h	0.01	kg/m

仿真中角速度环控制器采用的参数根据系统响应调节而成，如下表所示。

表 6 仿真中的角速度控制器采用参数

名称	数值	名称	数值
角速度 ω_x 、 ω_y 的 K_P	0.3	角速度 ω_z 的 K_P	0.13
角速度 ω_x 、 ω_y 的 K_I	0.02	角速度 ω_z 的 K_I	0.02
角速度 ω_x 、 ω_y 的 K_D	0.0025	角速度 ω_z 的 K_D	0.0025
控制周期 t_{sc}	0.01 s	微分项的低通滤波器增益 τ	0.05

仿真中的角速度环跟踪效果如下图所示，对于每张图内的图例，上方颜色曲线表示实际曲线，下方颜色表示期望曲线。可以看到三轴的角速度指令均可以被较快跟踪上。

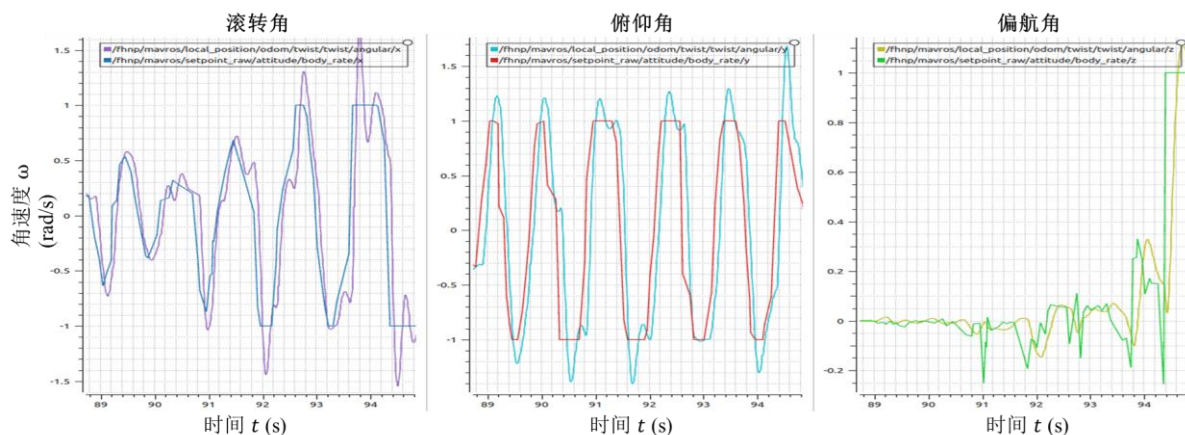


图 24 仿真中的角速度跟踪效果

2.5 本章小结

本章首先给出了四旋翼常用的坐标系，并给出了坐标系间的转换计算式。接着建立了四旋翼的刚体动力学模型、旋翼动力模型与气动阻力模型，并在全量动力学模型部分给出了公式汇总。然后基于离散 PID 给出了角速度环的控制器，同时给出随电池电压变化的油门换算公式。之后介绍了对上述所建模型的参数辨识过程，主要包括旋翼系统参数和惯性系统参数，并给出详细的数据分析过程，最后汇总无人机模型参数以进行后续的仿真与控制。

第三章 多无人机机间下洗气流扰动估计

在集群飞行时多架无人机彼此间存在气动扰动，这种扰动主要由在上层飞行的无人机通过旋翼向下吹风导致，因而被称为下洗气流（Downwash）。本章主要介绍为下洗气流设计的神经网络估计器，具体内容如下图所示。

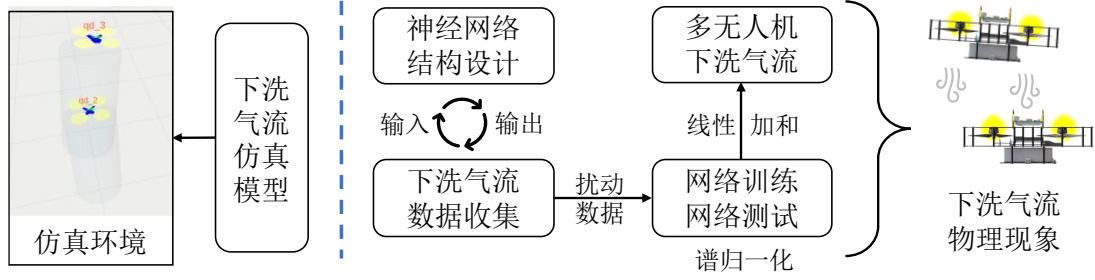


图 25 第三章内容逻辑图

本章首先介绍一种文献中采用的简单模型，用于在仿真环境中模拟下洗气流以验证整个算法流程。接着确定输入输出变量，介绍全连接神经网络估计器的具体结构。然后介绍扰动数据的收集原理，同时记录在实物飞行中机间气动扰动的数据收集过程。为提高神经网络的鲁棒性，在训练网络时引入谱归一化机制，可以将网络输出的变化率限定在一定范围内。最后介绍了神经网络的训练与测试过程，并将基于双无人机的扰动预测拓展到多无人机。测试结果表明，神经网络成功预测了扰动力，同时谱归一化策略可以有效提升神经网络的鲁棒性。

3.1 下洗气流扰动仿真模型

为在仿真环境中模拟无人机间的气动扰动以初步验证算法的有效性，需要建立机间气动扰动受力模型。仿真中仅考虑无人机 Z 轴方向上的受力 ${}^l f_d = [0 \ 0 \ f_z]^T$ 随惯性系相对位置的变化。对两架无人机，参考文献^[60]建立下洗气流作用于质心的模型如下

$$f_z = \begin{cases} -k_{D_1} \left(\frac{r_p}{4 \cdot \delta p_z} \right)^2 \exp \left(-\frac{1}{2} \left(\frac{\sqrt{\delta p_x^2 + \delta p_y^2}}{k_{D_2} \cdot \delta p_z + k_{D_3}} \right)^2 \right) & r_{dw,v} > \delta p_z > 0 \text{ 且} \\ & r_{dw,h} > \sqrt{\delta p_x^2 + \delta p_y^2}, \\ 0 & \text{其余情况} \end{cases}, \quad (3.1)$$

其中 r_p 为螺旋桨的半径， $\delta p_{x,y,z} = p_{\text{others}} - p_{\text{ego}}$ 为三轴方向上的相对位置， $r_{dw,v}$ 表示垂直方向下洗气流的作用范围， $r_{dw,h}$ 表示水平方向下洗气流的作用范围， k_{D_1} 、 k_{D_2} 、 k_{D_3} 为待辨

识参数，通过文献^[58]获得并修改适应本文尺寸飞机得到。仿真参数具体如下表所示。

表 7 仿真中采用的下洗气流模型参数

名称	数值	名称	数值
r_p	3.05 in = 0.0775 m	k_{D_1}	4000
k_{D_2}	0.65	k_{D_3}	-0.10
$r_{dw,v}$	4 m	$r_{dw,h}$	1.5 m

将上述公式进行可视化呈现如下图。从这三张图可以看出，扰动力沿水平方向的变化趋势是两架飞机越重叠，扰动力越大，同时沿垂直方向的变化趋势是高度差异越小，扰动力越大。虽然上述模型没有将四个旋翼产生的下洗气流分开考虑，且没有考虑无人机相对速度对下洗气流的影响，但总的变化趋势符合认知。

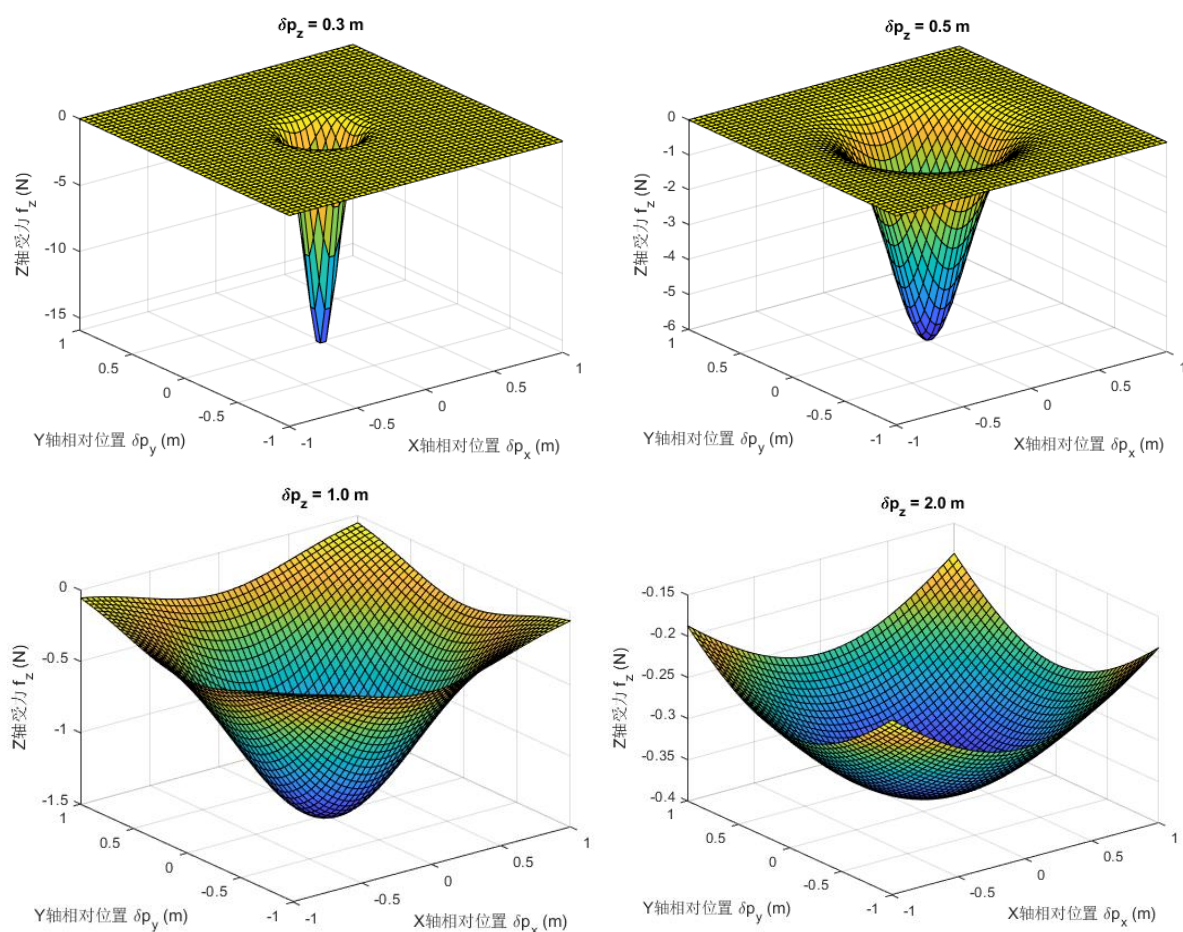


图 26 不同相对高度下仿真产生的下洗气流扰动力

该下洗气流的简易模型可用于仿真环境以初步验证神经网络气动扰动算法的可行

性。对于多个无人机造成的影响，仿真实现时将其以加和的方式呈现，即

$$f_{zi} = \sum_{j \in N_i^p} f_{zj}, \quad (3.2)$$

其中 N_i^p 表示第 i 架无人机周围处于下洗气流作用范围的无人机集合。如下图所示将上述下洗气流模型融入仿真环境，可见右下角的无人机被上边的无人机产生的气流影响，偏离了用红色五角星标示的期望位置。

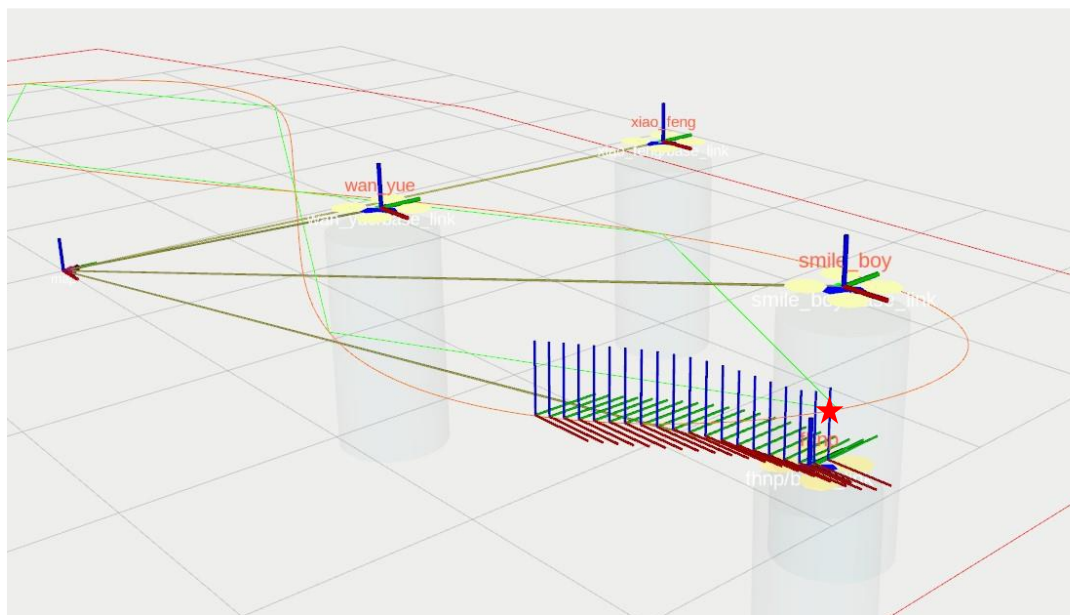


图 27 下洗气流仿真模型在仿真环境中的影响

3.2 神经网络下洗气流扰动观测器

由于下洗气流的非线性程度较高，随无人机状态的变化剧烈，很难用数理方法进行建模。如果使用计算流体力学方法（Computational Fluid Dynamics, CFD）分析，计算时间较长，难以用于实时控制。因而本文采用神经网络（Neural Network, NN）中的多层感知机（Multi-Layer Perceptron, MLP）去估计下洗气流对机体产生的扰动。本节介绍的估计器适用于两架飞机之间的扰动估计。

首先确定网络的输入和输出。对于双机机间气动扰动估计器，其输入向量包括自身和另外一个无人机的状态误差 $\mathbf{x} = [{}^I \delta \mathbf{p} \quad {}^I \delta \mathbf{v} \quad {}^B \delta \mathbf{q} \quad {}^B \delta \boldsymbol{\omega}]^T$ ，输出向量包括扰动力和力矩 $\mathbf{y} = [{}^I \mathbf{f}_d \quad {}^B \boldsymbol{\tau}_d]^T$ 。由于本文无人机的飞行速度较慢且飞行轨迹较为平缓，无人机姿态变化较小，因而将输入向量改为 $\mathbf{x} = [{}^I \delta \mathbf{p} \quad {}^I \delta \mathbf{v}]^T$ 。另一方面，在实验中发现扰动力矩可由底层飞控包含的高频率角速度控制器（1 kHz）及时补偿，无需单独估计，因而

将输出向量改为扰动力 $\mathbf{y} = {}^I \mathbf{f}_d$ 。

下面介绍网络结构。一个训练好的多层感知机可看作从输入向量 \mathbf{x} 到输出向量 \mathbf{y} 的映射函数 $f(\cdot; \boldsymbol{\theta}): \mathbb{R}^i \rightarrow \mathbb{R}^o$ ，其中 $\boldsymbol{\theta} := \{\mathbf{W}^1, \dots, \mathbf{W}^{H+1}; \mathbf{b}^1, \dots, \mathbf{b}^{H+1}\}$ 表示隐藏层权重参数和隐藏层偏置参数，里边 H 表示隐藏层的数量。将神经网络的激活函数(Activation Function)确定为逐个元素相乘的 ReLU 函数 $\phi(x) = \max(0, x)$ ，那么上述多层感知机可以被写为

$$\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{W}^{H+1} \cdot \phi(\mathbf{W}^H \cdot \phi(\dots \phi(\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1) \dots) + \mathbf{b}^H) + \mathbf{b}^{H+1} [61]。 \quad (3.3)$$

收集扰动训练数据，依据梯度下降对上式进行迭代优化，即可得到神经网络扰动观测器

$${}^I \mathbf{f}_d = \text{DNN}({}^I \delta \mathbf{p}, {}^I \delta \mathbf{v})。 \quad (3.4)$$

3.3 机间下洗气流扰动数据收集

3.3.1 扰动数据的收集原理

为训练神经网络，需要收集无人机机体在下洗气流扰动下受到的扰动力和力矩。本文假设在训练阶段无人机存在电机转速测量装置，而在实际应用时没有该装置。在存在转速测量装置的前提下，无人机的期望合力和力矩可以由四个电机的转速与第二章建立的无人机模型计算得出；另一方面，无人机的实际受力和力矩可由状态估计模块根据量测值的导数直接得出。将两者作差，即可得到当前机体因下洗气流产生的扰动力和力矩。

下面介绍期望合力和力矩的估算过程。给定四个电机的转速量测 $\hat{\Omega}_i, i=1,2,3,4$ ，则根据四旋翼动力系统公式

$$\left\{ \begin{array}{l} f_i = k_t \cdot \hat{\Omega}_i^2 \\ \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -L \sin \alpha & L \sin \alpha & L \sin \alpha & -L \sin \alpha \\ -L \cos \alpha & L \cos \alpha & -L \cos \alpha & L \cos \alpha \\ -k_q / k_t & -k_q / k_t & k_q / k_t & k_q / k_t \end{bmatrix} \\ \begin{bmatrix} f_c & \tau_x & \tau_y & \tau_z \end{bmatrix}^T = \mathbf{G} \cdot \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^T \\ {}^B \mathbf{f}_u = \begin{bmatrix} 0 & 0 & f_c \end{bmatrix}^T \\ {}^B \boldsymbol{\tau}_u = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \end{array} \right. \quad (3.5)$$

可以获得 ${}^B \mathbf{f}_u$ 和 ${}^B \boldsymbol{\tau}_u$ 。进而结合重力可以得到期望合力 ${}^I \mathbf{f}_n$ 和力矩 ${}^B \boldsymbol{\tau}_n$

$$\begin{cases} {}^I \mathbf{f}_n = {}^I \mathbf{R}(\mathbf{q}) \cdot {}^B \mathbf{f}_u + m \cdot {}^I \mathbf{g} \\ {}^B \boldsymbol{\tau}_n = {}^B \boldsymbol{\tau}_u \end{cases} \quad (3.6)$$

接着介绍对无人机实际受力和力矩的计算过程。无人机底层飞控包含一个惯性测量单元 (Inertial Measurement Unit, IMU)，可通过加速度计测量三轴加速度，通过陀螺仪测量三轴角速度。此外，底层飞控还包含测量高度的气压计、测量航向的磁罗盘等传感器，同时还会接收到外界传来的位置量测信息。在底层飞控中运行着一个高频率的扩展卡尔曼滤波器 (Extended Kalman Filter, EKF)，可以将上述量测信息进行融合以产生更高精度的组合导航结果，包括位置、速度、姿态、角速度等信息。以该滤波结果为基础，机体所受的合力与力矩可由速度和角速度的微分得到，如下式所示

$$\begin{cases} {}^I \mathbf{f} = m \cdot {}^I \dot{\mathbf{v}} \\ {}^B \boldsymbol{\tau} = \mathbf{I} \cdot {}^B \dot{\boldsymbol{\omega}} \end{cases} \quad (3.7)$$

其中数值微分由双线性变换 (Tustin 变换) 将微分项进行离散化得到，细节可以参考离散 PID 的实现过程(2.28)。笔者同时测试了通过加速度计的量测直接获得所受合力的方案

$${}^I \mathbf{f} = m \cdot \left({}^I \mathbf{R}(\mathbf{q}) \cdot {}^B \mathbf{a}_{\text{imu}} + {}^I \mathbf{g} \right), \quad (3.8)$$

但获取的合力噪声过大，舍弃。

根据公式(3.6)和公式(3.7)即可获得扰动力和力矩

$$\begin{cases} {}^I \mathbf{f}_d = {}^I \mathbf{f} - {}^I \mathbf{f}_n \\ {}^B \boldsymbol{\tau}_d = {}^B \boldsymbol{\tau} - {}^B \boldsymbol{\tau}_n \end{cases} \quad (3.9)$$

现在无人机的状态误差 $\mathbf{x} = [{}^I \delta \mathbf{p} \quad {}^I \delta \mathbf{v} \quad {}^I_B \delta \mathbf{q} \quad {}^B \delta \boldsymbol{\omega}]^T$ 和扰动力和力矩 $\mathbf{y} = [{}^I \mathbf{f}_d \quad {}^B \boldsymbol{\tau}_d]^T$ 即可作为输入和输出信息对以训练神经网络。因实际飞行速度较慢，舍弃姿态对下洗气流扰动的影响，输入改为 $\mathbf{x} = [{}^I \delta \mathbf{p} \quad {}^I \delta \mathbf{v}]^T$ ；因底层飞控对力矩扰动补偿较好，舍弃扰动力矩，将输出改为 $\mathbf{y} = {}^I \mathbf{f}_d$ 。

3.3.2 扰动数据的收集过程

在正式收集数据之前，于实物飞行中对扰动估计算法进行测试，测试结果如下图所示。从图中可见在机体重叠处存在明显的 Z 轴受力，验证了估计算法的正确性。

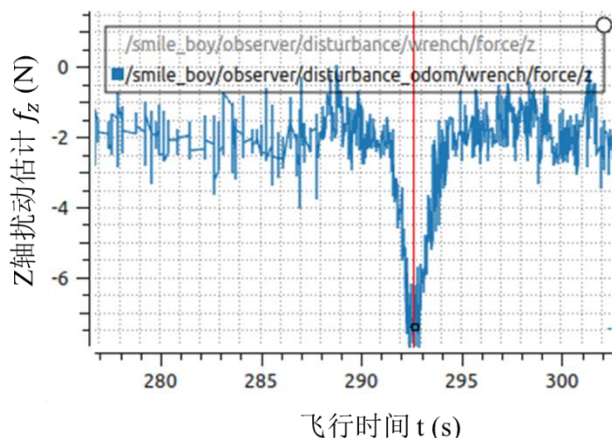


图 28 实物飞行中机体对 Z 轴所受扰动力的观测

在实物实验过程中，由两位操纵手分别控制两架无人机来收集下洗气流扰动数据，实验过程如下图所示。固定下方飞机，同时改变上方飞机来制造下洗气流的影响，其中红线为下方飞机在没有干扰时的悬停位置。在收集数据时，尽可能改变上方飞机的高度和速度以增加数据的多样性。本文通过 rosbag 工具包按照 100 Hz 收集数据，数据内容具体包括时间戳、相对状态和估计的扰动。



图 29 下洗气流扰动数据收集过程

最终对两架飞机共收集了 570 秒的数据。由于两架飞机是完全相同的，因而收集到的数据可以同时使用，所以数据点共有 $570 \times 100 \times 2 = 114,000$ 个。接着需要对这些数据进行时间对齐、减去悬停状态平均扰动偏差、插值处理。收集数据对应的相对位置可视化如下图所示，可见基本覆盖了水平位置正负 1 米和垂直位置正负 0.5~1.5 米的空间。

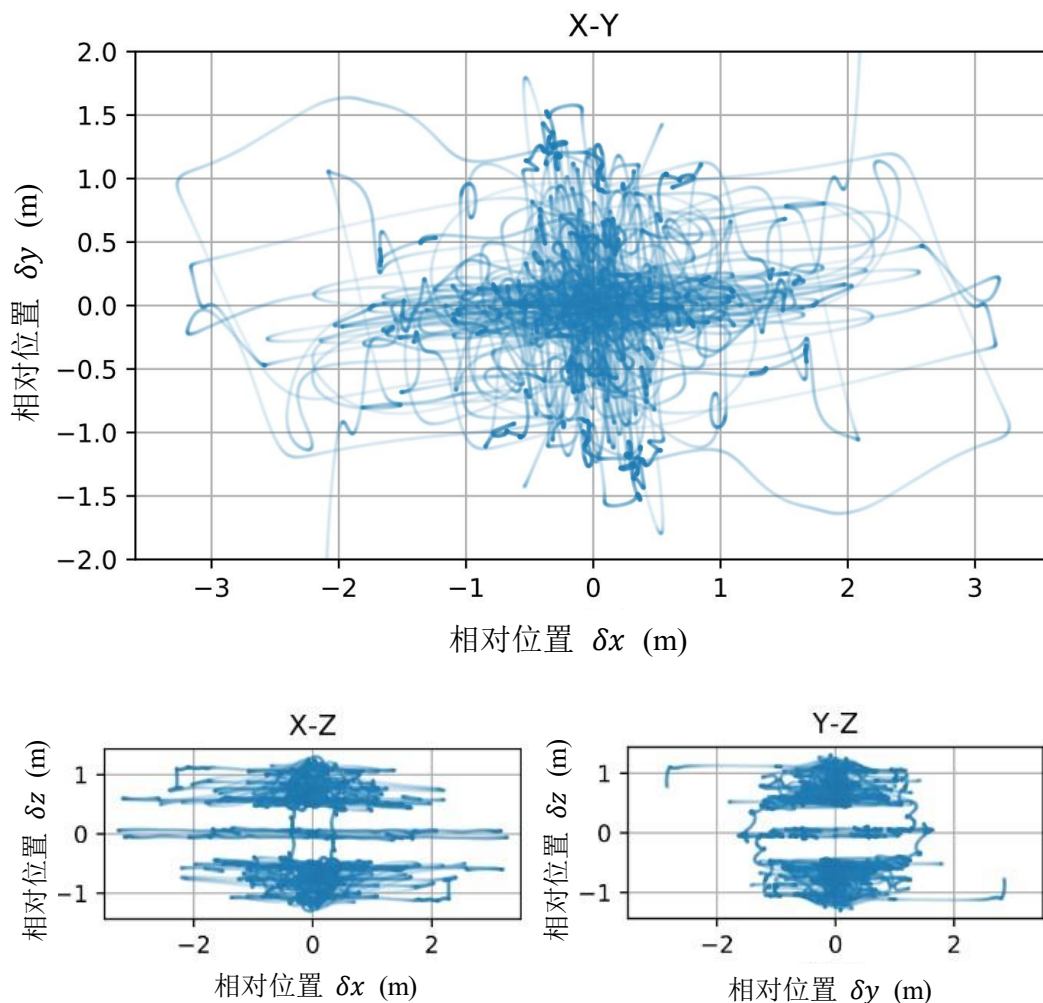


图 30 下洗气流收集数据按相对位置关系进行可视化

下一节使用这些数据训练神经网络下洗气流扰动观测器。

3.4 谱归一化神经网络训练方法

在上一节收集的数据不可能覆盖整个状态空间。因此，若使用上述数据去训练神经网络，很可能导致网络在没有收集到数据的状态上输出异常值，对飞行安全造成影响。为了提高神经网络对气动扰动预测的鲁棒性，本文在网络训练过程中引入谱归一化（Spectral Normalization, SN）技术。谱归一化最初在生成式对抗网络（Generative Adversarial Network, GAN）领域提出^[62]以处理训练不稳定的问题，近期已被应用于无人机领域并证明可以显著提升神经网络在预测气动效应时的鲁棒性和泛化性^[46]。下面介绍谱归一化的具体原理。

谱归一化可以通过限制神经网络的李普希兹常数（Lipschitz Constant）来提升网络

的鲁棒性与泛化性。对于一个李普希兹连续的函数 $f(\cdot)$ ，给定定义域里的 $\forall \mathbf{x}, \mathbf{x}'$ 且 $\mathbf{x} \neq \mathbf{x}'$ ，其李普希兹常数 $\|f\|_{\text{Lip}}$ 在数学上被定义为满足下式的最小值

$$\|f(\mathbf{x}) - f(\mathbf{x}')\| / \|\mathbf{x} - \mathbf{x}'\| \leq \|f\|_{\text{Lip}}, \quad (3.10)$$

其中 $\|\cdot\|$ 表示 l_2 范数。该定义可直观理解为函数的变化速度存在界限，而李普希兹常数就是这个界限对应的数值。对于复合的两个李普希兹连续函数，它们的李普希兹常数具有如下性质

$$\|f_1(f_2(x))\|_{\text{Lip}} \leq \|f_1(x)\|_{\text{Lip}} \cdot \|f_2(x)\|_{\text{Lip}}。 \quad (3.11)$$

将上式代入多层感知机的定义(3.3)，则有

$$\|f(\mathbf{x}; \boldsymbol{\theta})\|_{\text{Lip}} \leq \|\mathbf{W}^{H+1} \mathbf{x} + \mathbf{b}^{H+1}\|_{\text{Lip}} \cdot \|\phi\|_{\text{Lip}} \cdot \|\mathbf{W}^H \mathbf{x} + \mathbf{b}^H\|_{\text{Lip}} \cdots \|\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1\|_{\text{Lip}}。 \quad (3.12)$$

对于 ReLU 函数，根据定义可知其 $\|\phi(\cdot)\|_{\text{Lip}} = 1$ 。对于仿射函数 $\mathbf{W}\mathbf{x} + \mathbf{b}$ ，则有

$$\|\mathbf{W}\mathbf{x} + \mathbf{b} - \mathbf{W}\mathbf{x}' - \mathbf{b}\| / \|\mathbf{x} - \mathbf{x}'\| \leq \|\mathbf{W}\| \|\mathbf{x} - \mathbf{x}'\| / \|\mathbf{x} - \mathbf{x}'\| = \|\mathbf{W}\| = \|\mathbf{W}\mathbf{x} + \mathbf{b}\|_{\text{Lip}}, \quad (3.13)$$

而矩阵的 l_2 范数被称为谱范数，其值等于该矩阵最大的奇异值，可记为 $\|\mathbf{W}\| = \sigma_{\max}(\mathbf{W})$ 。

综上，则有

$$\|f(\mathbf{x}; \boldsymbol{\theta})\|_{\text{Lip}} \leq \|\mathbf{W}^{H+1}\| \cdots \|\mathbf{W}^1\| = \sigma_{\max}(\mathbf{W}^{H+1}) \cdots \sigma_{\max}(\mathbf{W}^1) = \prod_{l=1}^{H+1} \sigma_{\max}(\mathbf{W}^l)。 \quad (3.14)$$

因此，只要在训练中采用矩阵的谱范数去归一化神经网络的每一层

$$\bar{\mathbf{W}}^l := \gamma \cdot \mathbf{W}^l / \sigma_{\max}(\mathbf{W}^l), \quad (3.15)$$

那么就可以将归一化后神经网络的李普希兹常数限制在某个范围内

$$\|f(\mathbf{x}; \bar{\boldsymbol{\theta}})\|_{\text{Lip}} \leq \gamma^{H+1}。 \quad (3.16)$$

在实际训练时，每迭代一轮就用式(3.15)归一化一次，即可得到满足式(3.16)的神经网络。

谱归一化将网络输出的变化率限制在某个范围内，使得网络的输出更为平均，避免出现剧烈变化，以此来增强网络的鲁棒性。这种特性在后续实验中也得到了验证。谱归一化对网络泛化性的影响可见文献^[46]，本文不作进一步说明。

3.5 神经网络观测器的训练和测试

在 3.3 节收集数据的基础上，本文按照 3.2 节建立的神经网络结构与 3.5 节介绍的

谱归一化机制对机间扰动观测器网络开展训练。网络采用含有三个隐藏层的多层感知机，其输入是 6 维的相对位置和速度，输出是 3 维的扰动力，基础参数如下表所示。

表 8 神经网络扰动观测器的训练参数

参数名称	H	网络结构	训练轮数	学习率
数值	3	6-128-64-128-3	10000	1e-4

该网络选用 ReLU 函数作为激活函数，选用 Adam^[63]作为优化器，选用均方误差 (MSE) 作为损失函数。收集的数据被随机混杂 (Shuffle) 在一起，其中 70%为训练集，30%为测试集。整个网络使用 PyTorch 库进行实现，从下图可以看出，上述网络结构可以较好地拟合气动扰动。在后续部署到 MPC 控制算法时，需要对一序列的相对状态进行预测，此时可以通过批处理 (Batch) 的方式实现。

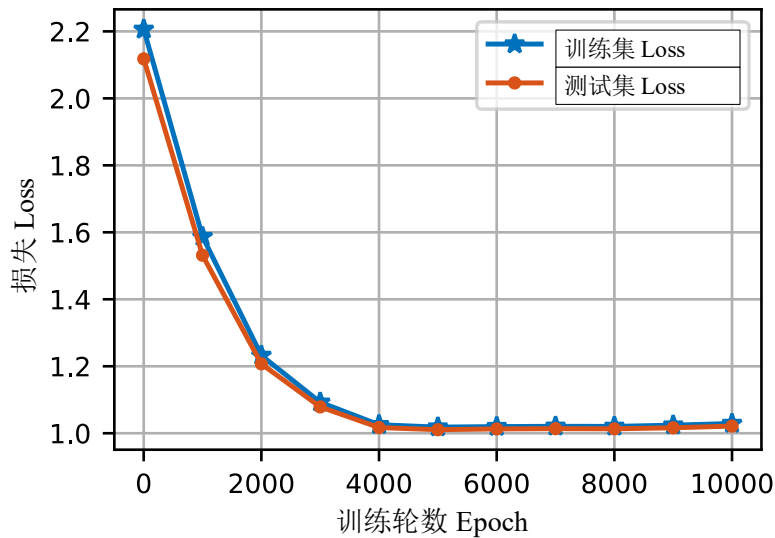


图 31 $\gamma = 4$ 时神经网络的训练曲线

谱归一化系数 γ 的选取对于网络预测结果在准确度和鲁棒性间的权衡至关重要。本文开展实验以选取最优的 γ 值，并将不同 γ 值下网络在测试集上的损失记录于下表。

表 9 在不同谱归一化系数下的测试集 Loss

γ	2	4	6	8	∞ (无 γ)
损失 Loss	1.6288	1.0523	0.6645	0.6250	0.6185

从表中可以看出，网络在测试集上的损失随 γ 的下降而增大，这意味着谱归一化的影响越强，神经网络对数据的拟合能力就越弱。

为探究经过谱归一化处理的神经网络在没有收集过数据的状态上的表现，本文将神经网络在不同高度下的预测呈现如下图所示。图中没有呈现 X 轴和 Y 轴的扰动力输出结果，因为它们的量级相对于 Z 轴的输出来说是可以忽略的。

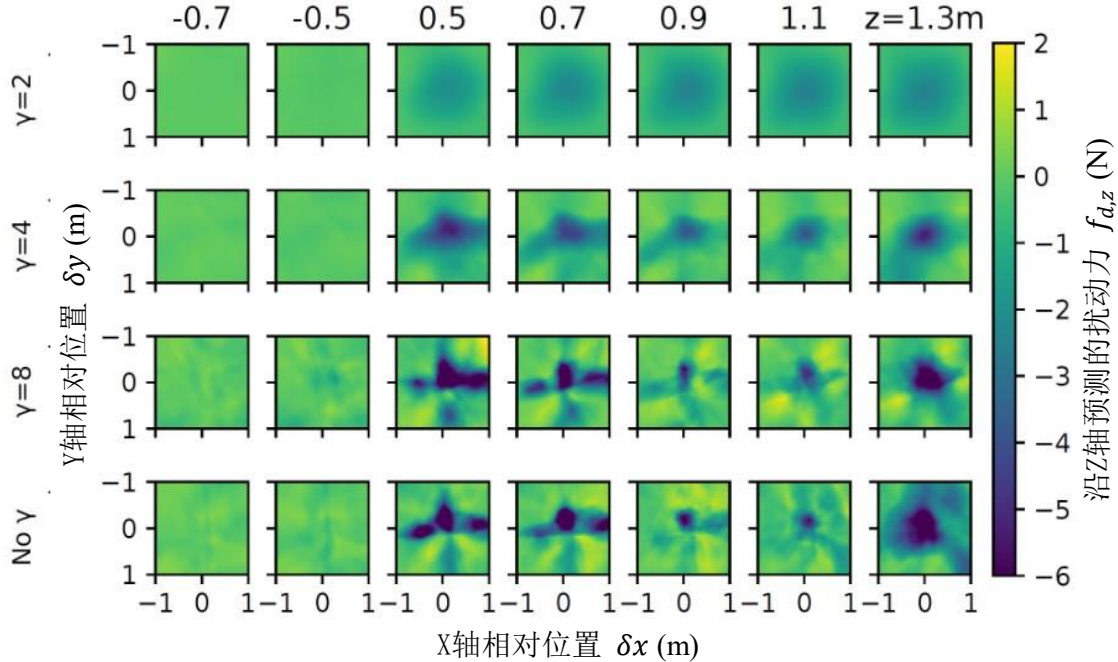


图 32 不同谱归一化系数、不同高度下神经网络对 Z 轴的扰动力预测

预测时固定无人机的相对速度为 0, 改变 γ , 观察神经网络在不同高度上的输出扰动力。图中每一个方块代表了神经网络在水平相对位置 $[-1,1]$ 米区域上的输出结果。对于同一列数据, 可以观察到 γ 越小, 对数据的拟合能力越保守, 这与表 10 中读出的结果相一致。然而, 这种保守的拟合能力也避免了网络输出的迅速变化, 提升了扰动力预测的安全性。例如, 相对高度为 1.3 米的数据很少, 可以看作是没有收集到数据的状态。从图中可以看出在没有谱归一化 ($\gamma = \infty$) 和 $\gamma = 8$ 的情况下 (第三、四行), 当相对高度达到 1.3 米时, 网络输出突然大幅增大, 这不符合常理。而在 γ 较小的情况下 (第一、二行), 谱归一化减小了这种趋势, 确保了飞行的安全性。考虑到预测准确性和鲁棒性之间的取舍, 最终选择 $\gamma = 4$ 时对应的神经网络作为飞行中采用的机间扰动神经网络观测器。

上述神经网络扰动观测的有效性分两步进行说明。首先在实物飞行时, 在 RVIZ 工具中可视化估计到的扰动力, 观察大小和方向是否与期望相符。接着将扰动估计纳入轨迹跟踪控制过程, 如果下洗气流扰动下的跟踪效果得到显著改善, 则说明扰动观测有效。

3.6 多无人机下洗气流扰动观测

上文训练的扰动观测器可以对两架无人机间的下洗气流扰动进行估计，现将其拓展为对多个无人机的下洗气流扰动进行估计。文献^[48]采用深度集合（Deep Sets）的方式对多机扰动进行估计，然而上述方法对实验数据采集的要求较高。而本文认为线性加和的方式足以描述扰动的主要成分，因而可直接将双机扰动估计拓展到多机，无需额外收集数据。下面分析线性加和关系的合理性。

多无人机产生的下洗气流叠加效应可分为两类，水平叠加和竖直叠加。对于多架处于水平飞行关系的无人机，如下图所示，因下洗气流的作用范围较为有限，大多数情况为一架重叠率较高的无人机和其他重叠率较低的无人机共同作用。线性加和方式较好保留了高重叠率无人机的影响，同时少量引入低重叠率飞机的影响，因而可以捕捉扰动的主要信息。

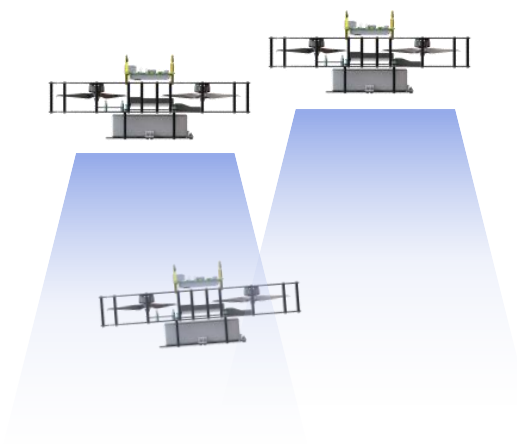


图 33 下洗气流水平叠加示意图

对于多架处于垂直飞行关系的无人机，如下图所示，可以看到最上方的无人机对最下方的无人机距离较远，影响较为有限，还是以中间飞行的无人机产生的影响为主。线性加和方式较好保留了中间无人机影响，同时少量引入顶部无人机的作用，也可以捕捉扰动的主要信息。

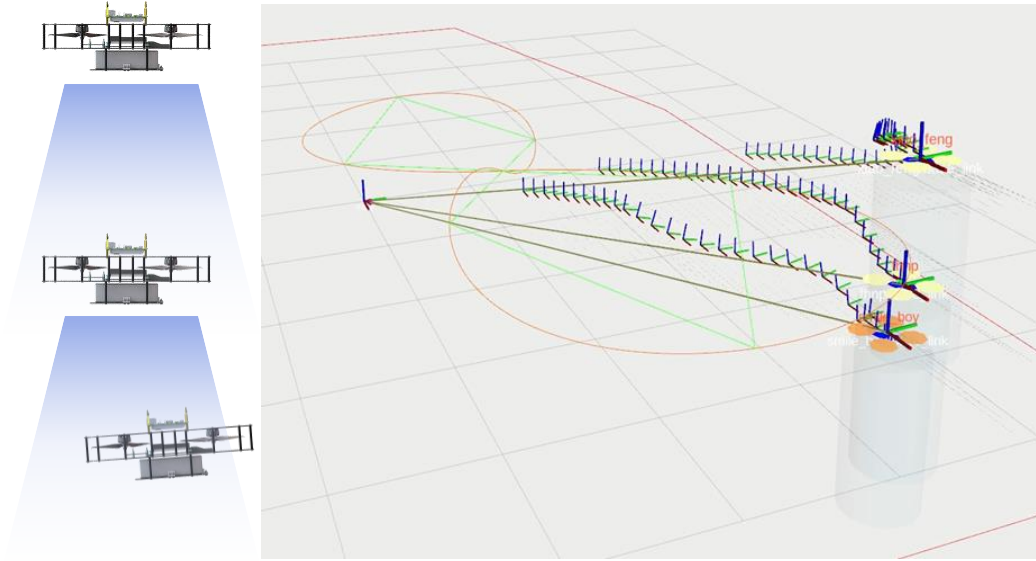


图 34 下洗气流垂直叠加示意图及仿真效果

综合上述分析，本文采用线性加和关系

$${}^i f_{d,i} = \sum_{j \in N_i^D} \text{DNN}({}^i \delta p_j, {}^i \delta v_j) \quad (3.17)$$

描述多无人机的扰动影响，其中 $N_i^D, i=1, \dots, M, \text{s.t. } \|p_i - p_j\| \leq r_{\text{Downwash}}$ 表示与无人机 i 存在下洗气流作用的全部邻居节点组成的集合， M 表示无人机个数，而 r_{Downwash} 为人为设置的下洗气流作用半径。通过这种方式，可以对多无人机产生的气动扰动进行观测。

3.7 本章小结

本章介绍了一种神经网络观测器，可以对无人机集群飞行时存在的下洗气流进行扰动估计。为提升神经网络的鲁棒性，在网络训练时引入谱归一化方法以限制神经网络输出的变化率。此外本章详细介绍了扰动数据收集、数据处理、训练和测试的过程，证明该神经网络可以较好地观测无人机飞行时存在的扰动。最后将对两架无人机下洗气流进行的扰动观测以线性加和的方式拓展到多无人机的情况。

第四章 无人机非线性 MPC 轨迹跟踪控制算法

本章主要实现了一种非线性模型预测控制轨迹跟踪控制器，算法基本流程如下。

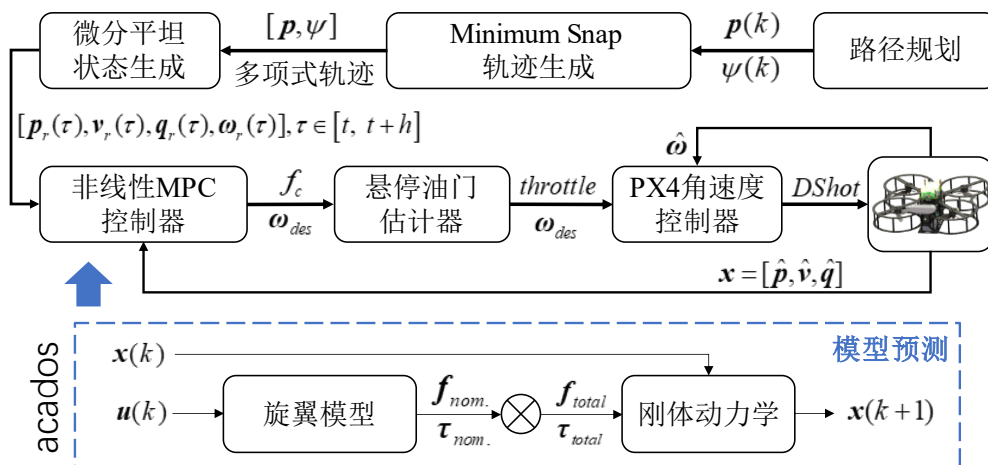


图 35 第四章内容逻辑图

按照图中流程，本章首先介绍 Minimum Snap 轨迹生成算法，通过给定的路径点生成位置与偏航角的多项式轨迹。接着介绍如何通过四旋翼的微分平坦性质将多项式轨迹转换为动力学可行的全状态轨迹，包括速度、位置、四元数姿态和角速度，从而作为控制目标输入控制器。然后介绍标准非线性 MPC 轨迹跟踪控制器的实现流程，包括 MPC 采用的非线性模型、MPC 算法、非线性 MPC 的运算加速技巧。上述非线性 MPC 得到的合推力指令将会通过一个悬停油门估计器转换为油门值，进而传递给底层飞控。最后，给出算法在本文开发的仿真环境中的仿真效果，以及算法在实物上的实现效果。

4.1 Minimum Snap 轨迹生成算法

轨迹跟踪控制算法以输入的轨迹为控制目标。然而，大多数有关轨迹跟踪控制的研究没有特别关注轨迹的可执行性，而是以圆形或双纽线等几何曲线直接作为期望跟踪的轨迹，这意味着部分轨迹可能是动力学上无法执行的。而模型预测控制算法以优化的思想执行控制任务，其优化的初值和目标（即期望轨迹）间的差异对优化的速度和最优性影响极大，因而有必要关注产生轨迹的合理性。

首先需要区分路径（Path）和轨迹（Trajectory）两个概念。路径指的是位置的几何曲线，不包含时间变量；而轨迹指的是位置随时间变化的参数方程，与时间密切相关^[64]。轨迹生成（Trajectory Generation）指的是给定初始状态、目标状态以及一系列需要经过

的路径点，生成一条光滑、动力学可行、无碰撞的轨迹方程，同时可以满足一定的性能指标。从上述定义可以看出，轨迹生成的本质是满足一定等式约束和不等式约束的优化问题。对于具有非线性动力学的移动个体，为使轨迹满足动力学约束，往往需要通过高斯伪谱法、序列凸优化等直接优化方法进行处理，计算量较大。而四旋翼无人机具备微分平坦的性质，其高维非线性状态可以转化为位置和偏航角及其各阶导数的代数组合，可以大幅简化其轨迹优化过程。

Vijay Kumar 团队已经在 2011 年证明了四旋翼是一种微分平坦的个体，这表明其输入和状态（位置、速度、加速度、姿态、角速度、角加速度）均可以由四个平坦输入（三轴位置及偏航角）及其各阶导数唯一确定^[65]。根据上述特性，任何在平坦空间得到的导数有界光滑轨迹都可以被四旋翼跟踪，因此只要分别对这四个平坦输入对应的轨迹进行运动学优化，即可以得到动力学可行的全状态飞行轨迹。在此基础上，Kumar 团队以最小控制输入为优化目标，将具有四阶非线性方程的四旋翼简化为四阶线性方程，提出了四旋翼无人机领域经典的 Minimum Snap 轨迹生成方法^[65]。本章根据人为指定的路径点，采用 Minimum Snap 方法生成动力学可行的期望轨迹，下面对该方法进行介绍。

对于优化问题，首先要指定优化目标。在最优控制中常常将系统输入的平方作为优化项以尽可能减少一段时间内的能量消耗，这被称为最小能量控制。对于单变量高阶线性系统，若以状态的最高阶导数 $x^{(n)}$ 为控制量，给定控制时间为 $0 \sim T$ ，则上述优化问题可被记为

$$x^*(t) = \operatorname{argmin}_{x(t)} \int_0^T (x^{(n)})^2 dt。 \quad (4.1)$$

确定优化目标后，需要确定优化后的函数形式，这实际上是一个泛函极值问题。针对该类问题的求解需要用到欧拉-拉格朗日方程，即对于优化指标 \mathcal{L} ，满足优化问题

$$x^*(t) = \operatorname{argmin}_{x(t)} \int_0^T \mathcal{L}(x^{(n)}, x^{(n-1)}, \dots, \dot{x}, x, t) dt \quad (4.2)$$

的函数必然满足下列等式（必要条件）

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{x}} \right) + \frac{d^2}{dt^2} \left(\frac{\partial \mathcal{L}}{\partial \ddot{x}} \right) + \dots + (-1)^n \frac{d^n}{dt^n} \left(\frac{\partial \mathcal{L}}{\partial x^{(n)}} \right) = 0。 \quad (4.3)$$

在上一段定义的轨迹优化问题中， $\mathcal{L}(x^{(n)}, x^{(n-1)}, \dots, \dot{x}, x, t) = (x^{(n)})^2$ ，代入计算可得结果为

$$x^{(2n)} = 0。 \quad (4.4)$$

对上式不断积分，可知满足上述最优条件的函数形式为 $2n-1$ 阶多项式的形式

$$x(t) = c_{2n-1}t^{2n-1} + c_{2n-2}t^{2n-2} + \cdots + c_1t + c_0。 \quad (4.5)$$

上述方程的系数需要通过边界条件进行求解。

上述结论是针对 $t=0$ 初始状态和 $t=T$ 目标状态优化的轨迹函数，下面将该结论拓展到存在多个中间点的情况。多个路径点可以看作是由多段多项式轨迹组合起来的情况，如下图所示。假设有 $M+1$ 个路径点，记为 w_0, w_1, \dots, w_M ，则优化生成的轨迹是由 M 段 $N=2n-1$ 阶多项式曲线组成的曲线。用 $x_i(t)$ 表示从 w_{i-1} 点到 w_i 的多项式曲线，将经过这一段轨迹的时间记为变量 T_i ，其中 $i=1, \dots, M$ ， T_i 由时间分配得到。

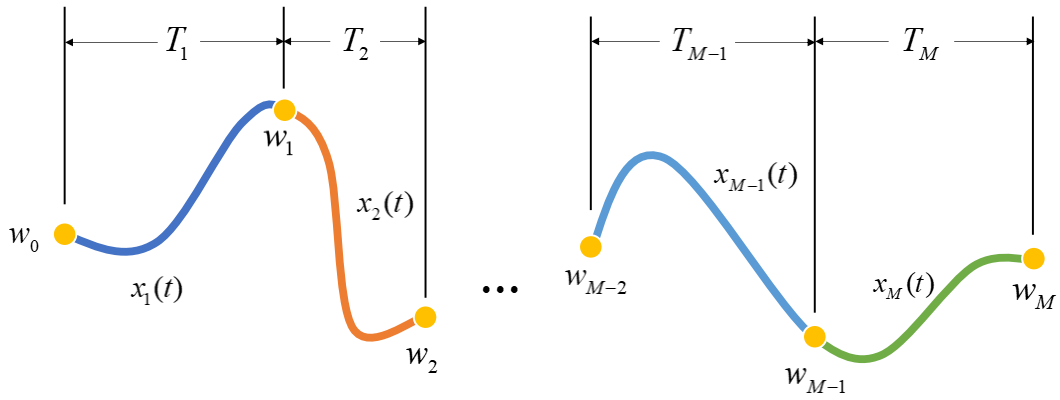


图 36 由多个多项式曲线组成的轨迹

对于多段线轨迹优化问题，给每段轨迹分配的时间将会极大影响轨迹质量。文献^[65]采用一种梯度下降迭代优化的方式获得最优时间分配，而本文采用一种简化的时间分配方式

$$T_i = \frac{\|w_i - w_{i-1}\|}{\bar{v}}, \quad (4.6)$$

其中 \bar{v} 是经过这两个路径点速度的平均值，注意除开始和结束路径点的速度均为 0 m/s 外，其他路径点的速度可人为指定。此外，为提升优化过程中的数值稳定性，需要将多项式的自变量归一化到 $0\sim 1$ 区间。若使用记号 S_i 表示从路径点 w_0 到路径点 w_i 的总时间，则有 $S_0 = 0$ ，同时对于 $i=1, \dots, M$ ，有 $S_i = \sum_{k=1}^i T_k$ 。然后，多项式 $x_i(t)$ 便有了如下形式

$$x_i(t) = c_{i0} + c_{i1} \frac{t - S_{i-1}}{T_i} + c_{i2} \left(\frac{t - S_{i-1}}{T_i} \right)^2 + \cdots + c_{iN} \left(\frac{t - S_{i-1}}{T_i} \right)^N, i=1, \dots, M, t \in [S_{i-1}, S_i]。 \quad (4.7)$$

令 $\tilde{t} = \frac{t - S_{i-1}}{T_i}$ ，则上式变为

$$x_i(\tilde{t}) = c_{i0} + c_{i1} \cdot \tilde{t} + c_{i2} \cdot \tilde{t}^2 + \cdots + c_{iN} \cdot \tilde{t}^N, i=1, \dots, M, \tilde{t} \in [0, 1]。 \quad (4.8)$$

下面构造约束条件以求解 M 段 $N = 2n - 1$ 阶多项式曲线对应的 $M \cdot (N + 1)$ 个未知数。若无法给定全部约束, 则将该问题转化为 QP 问题进行迭代求解。若可以给定全部约束, 则该问题转化为闭式求解形式, 可直接通过矩阵求逆得出解析解。本章介绍闭式求解形式, 下面构造关于这些未知数的约束方程。首先, 每段多项式的首尾必须经过所有的路径点, 这给定了 $2M$ 个约束

$$\begin{cases} x_i(S_{i-1}) = w_{i-1}, & i = 1, \dots, M. \\ x_i(S_i) = w_i \end{cases} \quad (4.9)$$

其次, 在开始和结束时候的 k 阶导数均为 0, 给定了 $2(n-1) = N - 1$ 个额外约束

$$x_1^{(k)}(S_0) = x_M^{(k)}(S_M) = 0, \quad k = 2, \dots, n, \quad (4.10)$$

注意在计算导数时有 $x_i^{(k)}(t) = \tilde{x}_i^{(k)}(\tilde{t}) / T_i^k$ 。接着为确保轨迹是光滑的, 设定每两段多项式之间各阶不为 0 的导数必须连续, 这提供了额外 $(N - 1) \cdot (M - 1)$ 个约束

$$x_i^{(j)}(S_i) = x_{i+1}^{(j)}(S_i), \quad i = 1, \dots, M, \quad j = 1, \dots, N - 1. \quad (4.11)$$

按照上述方案即获得了 $M \cdot (N + 1)$ 个等式。然后将上述约束写成如下矩阵相乘的形式

$$\mathbf{A} \cdot \mathbf{c} = \mathbf{b}, \quad (4.12)$$

其中 \mathbf{A} 是多项式求导系数阵, \mathbf{c} 包括所有待求系数 c_{ij} , \mathbf{b} 是常量约束。由于轨迹生成的随机性, \mathbf{A} 在一般情况下是满秩的, 因而可以求得系数为

$$\mathbf{c} = \mathbf{A}^{-1} \cdot \mathbf{b}. \quad (4.13)$$

注意在实际工程实现时, 采用数值法求解 $\text{solve}(\mathbf{A}, \mathbf{b})$ 要比通过(4.13)计算的速度快 100 倍。当得出系数 \mathbf{c} 后, 可以直接通过

$$x_i^{(m)}(t) = x_i^{(m)}(\tilde{t}) \cdot (\partial \tilde{t} / \partial t)^m = x_i^{(m)}(\tilde{t}) \cdot (1 / T_i)^m \quad (4.14)$$

从归一化后的时间 \tilde{t} 与标准多项式(4.8)的 m 阶导数获得实际状态的 m 阶导数 $x_i^{(m)}(t)$ 。

最后介绍如何将上述方法应用于四旋翼个体。根据第二章所建模型可知, 忽略电机模型的情况下, 从位置到加速度为二阶关系, 而四旋翼个体的加速度与其姿态相对应, 其姿态与角加速度又呈现二阶关系, 因而四旋翼可看作一个非线性四阶系统。四旋翼的旋翼转速可以直接影响角加速度, 因此可将角加速度作为控制输入。若将四旋翼近似为四阶线性模型, 则位置的四阶导数 (Snap) 可被视为控制量。延续最优控制中的思想将最小输入作为优化目标, 那么对于四旋翼优化的目标就是 Snap, 这便是 Minimum Snap

方法名称的由来。对于位置变量，设定方程(4.1)中的 $n=4$ 。对于偏航角，由旋翼直接产生的偏航力矩可以看作是偏航角的二阶导数，因而采用 Minimum Acceleration 方案，设定 $n=2$ 。由于三轴位置和偏航角相互解耦，所以分别对三轴位置采用 Minimum Snap 优化，对偏航角采用 Minimum Acceleration 优化，最后将四个平坦输出轨迹组合在一起即可获得四旋翼无人机在三维空间中的一条满足动力学约束的可行轨迹。

4.2 基于微分平坦生成全状态轨迹

生成三轴位置和偏航角的多项式轨迹后，需要借由四旋翼的微分平坦特性将其转化为动力学可行的全状态轨迹。微分平坦是四旋翼无人机具有的一种数学性质，即四旋翼的全部状态和输入都可以写成平坦输出（三轴位置和偏航角）及其导数的几何方程形式。这实际上建立了一种映射，即只要获得三轴位置和偏航角随时间的变化函数及其多阶导数，就可以生成一条四旋翼动力学可行的飞行轨迹。这种特性将四旋翼的轨迹生成问题简化为对三轴位置和偏航角的光滑曲线搜索问题，极大简化了轨迹生成的难度。下面介绍借助微分平坦特性的转换过程^[65]。

当获得平坦输出及其导数 $[\mathbf{}^I\mathbf{p} \ \mathbf{}^I\mathbf{v} \ \mathbf{}^I\mathbf{a} \ \mathbf{}^I\mathbf{j} \ \psi \ \dot{\psi}]^T$ 后，四旋翼除重力外的加速度方向可写为

$$\mathbf{t} = [a_x, a_y, a_z + g]^T, \quad (4.15)$$

那么机体的 Z 轴朝向为 $\mathbf{z}_B = \frac{\mathbf{t}}{\|\mathbf{t}\|}$ 。定义偏航向量为

$$\mathbf{x}_C = [\cos\psi \ \sin\psi \ 0]^T, \quad (4.16)$$

那么姿态矩阵可以表示为

$${}^I\mathbf{R} = [\mathbf{x}_B \ \mathbf{y}_B \ \mathbf{z}_B], \quad (4.17)$$

其中 $\mathbf{y}_B = \frac{\mathbf{z}_B \times \mathbf{x}_C}{\|\mathbf{z}_B \times \mathbf{x}_C\|}$ ， $\mathbf{x}_B = \mathbf{y}_B \times \mathbf{z}_B$ 。再将旋转矩阵转化为四元数 ${}^I\mathbf{e}$ ，这样就获得了期望的姿态信息。下面计算期望的角速度信息。定义

$$\mathbf{h}_\omega = {}^B\boldsymbol{\omega} \times \mathbf{z}_B = \frac{m}{u_1} (\mathbf{j} - (\mathbf{z}_B \cdot \mathbf{j}) \mathbf{z}_B), \quad (4.18)$$

由于除重力外四旋翼的机体受力来自旋翼，式(4.18)中推力 $u_1 = \|\mathbf{t}\| \cdot m$ 。根据角速度定义

$${}^B\boldsymbol{\omega} = p \cdot \mathbf{x}_B + q \cdot \mathbf{y}_B + r \cdot \mathbf{z}_B, \quad (4.19)$$

则期望角速度信息为

$$\begin{cases} p = -\mathbf{h}_\omega \cdot \mathbf{y}_B \\ q = \mathbf{h}_\omega \cdot \mathbf{x}_B \\ r = \dot{\psi} \mathbf{z}_W \cdot \mathbf{z}_B \end{cases} \quad (4.20)$$

这样, 就从平坦输出及其导数 $[\mathbf{}^I \mathbf{p} \ \mathbf{}^I \mathbf{v} \ \mathbf{}^I \mathbf{a} \ \mathbf{}^I \mathbf{j} \ \psi \ \dot{\psi}]^T$ 获得了无人机的期望 13 维全状态轨迹 $\mathbf{x} = [\mathbf{}^I \mathbf{p} \ \mathbf{}^I \mathbf{v} \ \mathbf{}^I \mathbf{e} \ \mathbf{}^B \boldsymbol{\omega}]^T$ 。如还需得到机体的角加速度 $\mathbf{}^B \dot{\boldsymbol{\omega}}$, 可参考文献^[56]求得解析解, 也可以对角速度进行数值微分得到数值解。由 $\mathbf{}^B \dot{\boldsymbol{\omega}}$ 和合推力 u_1 经由控制分配可计算得到四个电机的期望转速。

最后需要计算未来一段时间序列下的全状态轨迹, 以作为控制目标输入非线性 MPC 控制器。假设运行一次非线性 MPC 控制器需要预测 N 个点, 如果在每一次控制前都执行 N 次微分平坦转换, 那么对控制目标的计算速度较慢, 无法满足实时性需求。因此在工程实现上, 此处要求非线性 MPC 的预测步长 Δt_{pred} 是控制周期 Δt_{nmPC} 的整数倍, 这样可以维持一个长度为 $N \cdot \Delta t_{\text{pred}} / \Delta t_{\text{nmPC}} + 1$ 的数组。每次控制前, 去掉数组最开始的元素, 同时运算一次 $t = t_{\text{now}} + N \cdot \Delta t_{\text{pred}}$ 时刻的微分平坦转换并添加到数组的末尾, 进而按照 $\Delta t_{\text{pred}} / \Delta t_{\text{nmPC}}$ 的步长取出 $N + 1$ 个点作为控制目标。这样每次控制前只需要运算一次微分平坦转换, 计算量小, 可以满足实时需求。

4.3 非线性 MPC 轨迹跟踪控制算法

以上文得到的轨迹作为控制器的参考输入, 本节主要介绍非线性 MPC 轨迹跟踪控制器的实现过程。与经典的最优控制或者 PID 控制相比, MPC 有很多有用的特性: MPC 可以直接处理多输入多输出 (Multiple-Input Multiple-Output, MIMO) 系统, 无需像 PID 一样必须解耦处理, 参数调节较少; MPC 可以利用未来的预测信息以提前做出反应; MPC 具有显式约束处理特性, 可以直接对状态与输入加以限制, 避免饱和问题; MPC 的结构还可以处理系统的时延, 同时对模型的误差较为鲁棒。本节先介绍 MPC 采用的模型, 再介绍算法过程, 最后介绍算法加速技巧。

4.3.1 非线性 MPC 采用的四旋翼模型

模型预测控制是一种基于模型的控制方法, 首先需要确定被控对象的模型。根据选用模型的特性, 可以将 MPC 方法分为线性 MPC 和非线性 MPC。对于四旋翼, 文献中

采用的线性 MPC 方法往往需要对悬停状态的无人机采用小扰动线性化，当无人机大机动飞行时模型偏差较大，因而本文采用非线性 MPC 执行轨迹跟踪以提高轨迹跟踪精度。本文采用的无人机非线性模型有两种，均根据第二章建立的全量动力学模型作不同程度的简化得到，现分别介绍如下。

(1) 角速度控制模型

第一种模型考虑到无人机在底层飞控中还存在一个控制器进行角速度控制，因而只考虑到输出为角速度的模型，即

$$\begin{cases} {}^I \dot{\boldsymbol{p}} = {}^I \boldsymbol{v} \\ {}^I \dot{\boldsymbol{v}} = {}^I \boldsymbol{R}(\boldsymbol{q}) \cdot {}^B \boldsymbol{f}_u / m + {}^I \boldsymbol{g}, \\ {}^B \dot{\boldsymbol{q}} = 1/2 \cdot {}^B \boldsymbol{q} \circ \mathcal{V}^*({}^B \boldsymbol{\omega}) \end{cases} \quad (4.21)$$

其中 ${}^B \boldsymbol{f}_u = [0 \ 0 \ f_c]^T$ 。与全量动力学模型相比，该模型舍去了对机体气动阻力的建模和角速度环动力学，以 ${}^B \boldsymbol{\omega}$ 和 f_c 作为控制输入。

(2) 电机转速控制模型

上述模型符合本文设计的控制系统架构，但无法充分利用 MPC 的约束处理能力以对电机转速进行限制。然而电机转速作为四旋翼无人机实际上的控制输入，完全有达到饱和的风险。为利用 MPC 对电机输入进行约束处理，可以采用以下模型

$$\begin{cases} {}^I \dot{\boldsymbol{p}} = {}^I \boldsymbol{v} \\ {}^I \dot{\boldsymbol{v}} = {}^I \boldsymbol{R}(\boldsymbol{q}) \cdot {}^B \boldsymbol{f}_u / m + {}^I \boldsymbol{g} \\ {}^B \dot{\boldsymbol{q}} = \frac{1}{2} \cdot {}^B \boldsymbol{q} \circ \mathcal{V}^*({}^B \boldsymbol{\omega}) \\ {}^B \dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1} \cdot (-{}^B \boldsymbol{\omega} \times (\boldsymbol{I} \cdot {}^B \boldsymbol{\omega}) + {}^B \boldsymbol{\tau}_u) \\ \begin{bmatrix} f_c & \tau_x & \tau_y & \tau_z \end{bmatrix}^T = \boldsymbol{G} \cdot \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^T \end{cases}, \quad (4.22)$$

该模型以四个电机的推力 f_i 为控制输入，可通过推力与转速的对应公式由转速上下限对其进行限制，以避免旋翼转速出现饱和现象。在实际执行时，选用 MPC 生成状态中的角速度发送给底层飞控。这种思路的优点是，虽然无法将经过输出限制的转速直接发送给电机，但因为在控制过程中考虑到了电机的执行能力，其角速度变化就不会过于剧烈以造成电机转速饱和。这种模型的缺点是对轨迹要求较高，需要由角加速度计算得到电机输入序列作为参考轨迹。

本文主要实现的是第一种模型。将上述模型经四阶龙格库塔方法离散后，可简记为

$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ ，从而作为下一节非线性 MPC 采用的控制模型。

4.3.2 非线性 MPC 轨迹跟踪控制算法

模型预测控制的思想如下图所示。在每个控制周期内，MPC 控制器接收参考轨迹和对被控对象状态的估计。接着 MPC 将会以当前状态为初值，通过预测模型不断向前迭代，试图最小化一段有限预测时域上的损失函数，得到一组状态和输入序列。然后 MPC 选取第一个计算得到的控制输入应用于被控对象，同时舍弃其余的状态与输入。在下一个控制周期，MPC 的预测时域会向前滚动一小步，同时循环上述过程。

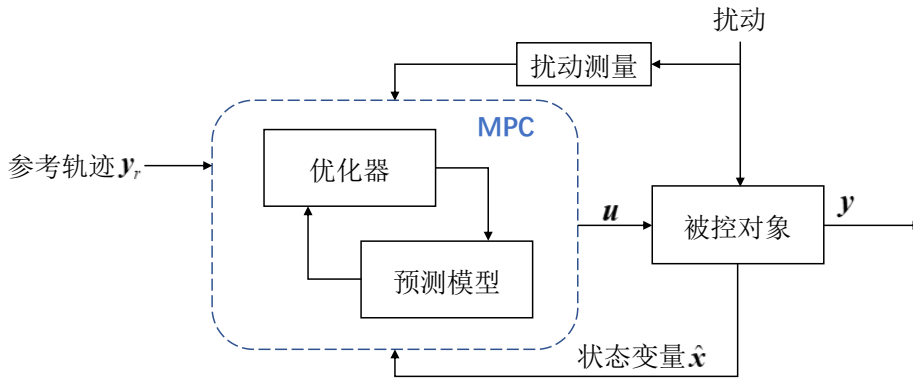


图 37 模型预测控制原理图

非线性 MPC 以滚动时域的方式求解一个有限时间最优控制问题（Optimal Control Problem, OCP）从而获得控制指令。本文将非线性 MPC 的控制周期记为 Δt_{nmmpc} ，即每过 Δt_{nmmpc} ，求解一次优化问题。参考最优控制里的线性二次型（Linear-Quadratic Regulator, LQR）控制器，定义 MPC 优化问题的损失函数包含系统误差、控制能量和终端误差三项。模型预测控制是在离散空间定义的误差，因而需要将状态和输入序列在预测时域 $\tau \in [t, t + t_{\text{pred}}]$ 以步长 $\Delta t_{\text{pred}} = t_{\text{pred}} / N$ 离散为 N 个点，其中 h 表示预测时域的长度。那么非线性 MPC 轨迹跟踪可定义为如下控制序列的优化问题

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} \sum_{k=1}^{N-1} (\bar{\mathbf{x}}_k^T \mathbf{Q}_x \bar{\mathbf{x}}_k + \bar{\mathbf{u}}_k^T \mathbf{R}_u \bar{\mathbf{u}}_k) + \bar{\mathbf{x}}_N^T \mathbf{Q}_N \bar{\mathbf{x}}_N, \quad (4.23)$$

其中横线 $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}_r$ 表示实际状态与参考状态的误差，注意姿态误差应该用四元数误差 $\mathcal{V}({}_B^I \mathbf{q}_o \circ {}_B^I \mathbf{q}_r^{-1})$ 来表示，其中符号 \mathcal{V} 表示取 q_x, q_y, q_z 的数据； \mathbf{Q}_x 、 \mathbf{R}_u 、 \mathbf{Q}_N 均为对角阵，分别表示系统误差矩阵、控制能量矩阵、终端误差矩阵，其相对数值表征了优化过程中某个变量对结果的影响大小。

由于姿态四元数本身具有很多独特性质，需要进一步介绍在实际实现时对损失函数的处理。首先，由于四元数误差是非线性的，因而在使用求解器时需要整体模型采用“NONLINEAR_LS”误差类型，其对应的过程误差与终端误差分别定义如下：

$$\text{过程: } l(x, u, z, p) = \|y(x, u, z, p) - y_{\text{ref}}\|_w^2, \text{ 终端: } m(x, p) = \|y^e(x, p) - y_{\text{ref}}^e\|_{w^e}^2. \quad (4.24)$$

对于四元数以外的变量，其损失函数是线性的，因而定义 $y(x, u, z, p) = x$ ；对于四元数姿态，定义 ${}^B\bar{q} = {}^Bq \circ {}^Bq_r^{-1}$ ，则其 y 函数可定义为非线性函数

$$y(x, u, z, p) = \begin{bmatrix} 0 \\ \mathcal{V}({}^B\bar{q}) \end{bmatrix} + {}^Bq_r, \quad (4.25)$$

这样既保证了 MPC 优化问题中有关四元数的状态是四维的，可通过“NONLINEAR_LS”误差类型实现；又确保了 \bar{q}_w 会被减为 0，只有 $\bar{q}_x, \bar{q}_y, \bar{q}_z$ 起实际作用，与基于四元数的非线性 PD 控制器^[66]相符。与此相匹配的是，虽然误差矩阵 Q_x 的对角元素个数与状态个数相同，但 \bar{q}_w 所对应的那一个对角元素并不起作用，只是为了满足“NONLINEAR_LS”形式而起到占位的作用。

其次，四元数表示的空间双倍覆盖了物理旋转空间^[66]，因而每一个旋转都可以被一对单位四元数表示，这在实际应用中可能会造成“unwinding”现象，即机体从超过 180° 的一侧旋转到期望姿态，而不是小于 180° 的一侧。为了解决这个现象，PD 控制中常采用 $\text{sgn}(\bar{q}_w) \cdot \bar{q}_{x,y,z}$ 的方案解决^[66]。将这种思路应用于非线性 MPC 中，可定义四元数在误差项对应的 y 函数为

$$y(x, u, z, p) = \begin{bmatrix} 0 \\ \text{sgn}({}^B\bar{q}_w) \cdot \mathcal{V}({}^B\bar{q}) \end{bmatrix} + {}^Bq_r. \quad (4.26)$$

在非线性 MPC 的运算过程中， Bq_r 会以参数的形式不断传入优化器。

定义完成优化目标后，上述优化问题还应该满足如下约束：

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \text{ 动力学等式约束;} \quad (4.27)$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}}, \text{ 初值等式约束, 引入反馈;} \quad (4.28)$$

$$\mathbf{u}_k \in [\mathbf{u}_{\text{MIN}}, \mathbf{u}_{\text{MAX}}], \text{ 控制输入限幅, 不等式约束。} \quad (4.29)$$

此外还可以出于安全考虑，对系统状态做不同程度的不等式约束。

优化完成后，最终将第一个控制量 \mathbf{u}_0^* 作为控制输入发送给执行机构。当将上述优化

过程应用于四旋翼，不管是采用角速度控制模型还是电机转速控制模型，都可以从输入或状态中获得控制指令

$$\mathbf{u} = [f'_c \quad \omega'_x \quad \omega'_y \quad \omega'_z]。 \quad (4.30)$$

如果希望将优化得到的状态序列 \mathbf{x}_k^* 用于编队控制或 RVIZ 可视化，最好对序列中的四元数再进行一次归一化以确保其为单位四元数。

4.3.3 非线性 MPC 的运算加速技巧

上述非线性 MPC 优化属于非线性二次型优化问题，可以通过序列二次规划（Sequential Quadratic Programming, SQP）方法求解。然而若直接对非线性模型进行优化，计算量非常大，难以确保在嵌入式设备上实时运行。为加速算法运行，本节介绍非线性 MPC 在实现中的技巧，包括热启动（Warm-Start）、多触发（Multi-Shooting）和实时迭代策略（Real-Time Iteration, RTI）三种方法。整个加速的核心思想就是利用次优替代最优，用速度来弥补最优性下降带来的问题，最终使用 ACADOS^[67]优化工具包完成非线性 MPC 在嵌入式设备上的实时运行。整个流程如下图所示。

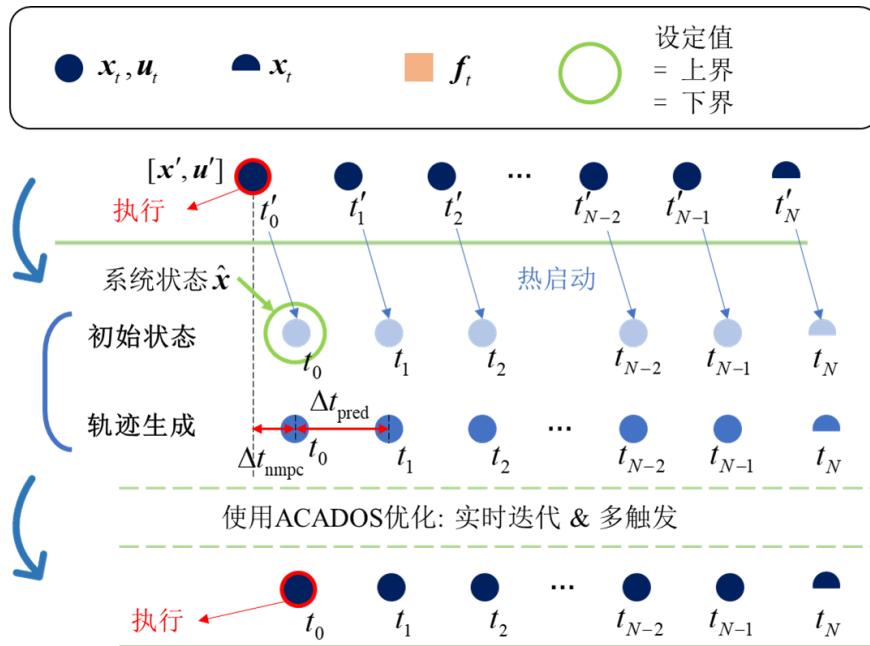


图 38 非线性 MPC 算法的实现流程

首先介绍热启动方法。优化问题的初值对于优化速度至关重要，如果初值离最优值较近，那么优化速度较快；反之如果较远，那么优化速度较慢。热启动就是通过改变初值来加速优化过程的，指的是每一轮优化的初始状态等同于上一次优化的结果。如果

MPC 的运行速度很快，那么每两次优化间无人机状态的改变是很有限的，因而很快就可以完成优化过程。

接着介绍多触发方法。在多触发方法中，MPC 问题的预测时间段被划分为多个子时间段，每个时间段内的状态轨迹使用一个单独的状态变量集合在动态模型和系统约束条件下进行优化。优化过程中需要确保不同时间段的状态轨迹彼此首尾相等，将优化后的多段轨迹组合起来即可得到最终结果。与单触发 (Single-Shooting) 相比，多触发具有更好的数值稳定性、更强的非凸问题处理能力和更高的计算效率，因而可以加速算法运行。

最后是实时迭代策略，该策略的核心是为提升计算速度而牺牲最优性。求解一个非线性优化问题需要运行多轮 SQP 算法，但运用 RTI 策略的 MPC 在优化过程中只运行一轮 SQP 算法，这样显著提升了计算速度。对于运动个体，当计算速度较快时，不同时刻间的状态变化较小，因而使用 SQP 迭代一次得到的次优解也可以满足要求。

应用上述策略，可以在 NVIDIA 的 TX2 NX 机载计算机上实现 100 Hz 的非线性 MPC 运行速率。

4.3.4 基于卡尔曼滤波的悬停油门观测器

尽管在上一节计算得到了合力 f_c ，但在实际飞行时，需要将 f_c 通过悬停油门归一化到 $[0,1]$ 区间以发送给底层飞控。然而在实验中笔者发现悬停油门会随着电池电量的下降而上升，因此若保持悬停油门不变，无人机会逐渐降低高度。为了让无人机维持高度，需要设计观测器对悬停油门进行估计。

本文基于卡尔曼滤波器 (Kalman Filter, KF) 设计悬停油门观测器，设计过程主要参考论文^[68]，该设计同样被用在 PX4 飞行控制器之中。如果忽略气动阻力的影响，假设 IMU 测量得到的比力均来自于四个旋翼，那么无人机的状态方程可被写为下式

$$\begin{cases} f_c(k+1) = h(k+1) \cdot mg / h_v(k) + w_1 = h(k+1) \cdot \gamma_h(k) + w_1, \\ \gamma_h(k+1) = \gamma_h(k) + w_2 \end{cases}, \quad (4.31)$$

而量测方程可被写为

$${}^B a_{sp,z}(k+1) = f_c(k+1) / m + e_1, \quad (4.32)$$

其中 f_c 表示所有电机产生的合推力， h 表示归一化后作用于飞控的油门值， h_v 表示悬停油门， $\gamma_h := mg / h_v$ 为重力与悬停油门 h_v 的比值， ${}^B a_{sp,z}$ 是 IMU 量测到的 Z 轴比力。如果令观测器采用 $\mathbf{x} = [\hat{f}_c \quad \hat{\gamma}_h]^T$ 作为状态量，采用 $z = {}^B a_{sp,z}$ 作为量测量，那么可以得到

$$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} 0 & h(k+1) \\ 0 & 1 \end{bmatrix} \cdot \mathbf{x}(k) \\ z(k+1) = \begin{bmatrix} 1 \\ m, 0 \end{bmatrix} \cdot \hat{\mathbf{x}}(k+1) \end{cases} \quad (4.33)$$

根据状态估计理论，卡尔曼滤波离散方程的形式为

$$\begin{cases} X(k+1) = \Phi(k+1, k)X(k) + G(k+1, k)u(k) + \Gamma(k+1, k)W(k) \\ Z(k+1) = H(k+1)X(k+1) + Y(k+1) + V(k+1) \end{cases}, \quad (4.34)$$

将系统模型(4.34)代入可得 $\Phi(k+1, k) = \begin{bmatrix} 0 & h(k+1) \\ 0 & 1 \end{bmatrix}$ ， $G(k+1, k) = 0$ ， $\Gamma(k+1, k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ， $H(k+1) = \begin{bmatrix} 1 \\ m, 0 \end{bmatrix}$ ， $Y(k+1) = 0$ 。给定过程噪声协方差 Q 阵和量测噪声协方差 R 阵，接下来按照如下卡尔曼滤波的经典流程进行迭代。

首先初始化 $X(0|0)$ ， $P(0|0)$ 。 $X(0|0)$ 采用第一组量测数据初始化， $P(0|0)$ 采用单位阵初始化。接着不断循环下述五步公式

$$\begin{cases} P(k+1|k) = \Phi(k+1, k)P(k|k)\Phi^T(k+1, k) + \Gamma(k+1, k)Q_k\Gamma^T(k+1, k) \\ K(k+1) = P(k+1|k)H^T(k+1)[H(k+1)P(k+1|k)H^T(k+1) + R_{k+1}]^{-1} \\ \hat{X}(k+1|k) = \Phi(k+1, k)\hat{X}(k|k) \\ \hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)[Z(k+1) - H(k+1)\hat{X}(k+1|k)] \\ P(k+1|k+1) = [I - K(k+1)H(k+1)]P(k+1|k) \end{cases}, \quad (4.35)$$

从状态量中即可获得重力与悬停油门的比值 $\hat{\gamma}_h$ 。最后通过 $\hat{\gamma}_h$ 得到发送给底层飞控的油门值为

$$h' = f'_c / \hat{\gamma}_h, \quad (4.36)$$

进而与角速度指令组成 $\mathbf{u} = [h' \quad \omega'_x \quad \omega'_y \quad \omega'_z]$ 发送给底层飞控。当两飞机存在下洗气流干扰时，对于观测器系统方程的假设并不成立。因此该滤波器仅在飞机附近没有其他无人机时起作用。

4.4 仿真实验

4.4.1 轨迹规划与轨迹跟踪

(1) Minimum Snap 轨迹规划

为验证 Minimum Snap 轨迹规划算法的正确性，在系统中进行仿真测试。发送的路径点信息如下表所示，路径点在三维位置上均有变化。设置平均速度为 0.5 m/s，经过每一段

路径的时间由直线距离除以平均速度得到。经验证，生成的轨迹能够经过各个路径点。

表 10 Minimum Snap 仿真路径点设置

序号	X 轴位置 (m)	Y 轴位置 (m)	Z 轴位置 (m)	偏航角 (deg)
0	1.0	1.0	0.5	0
1	3.0	2.0	1.0	10
2	5.0	1.0	1.0	20
3	3.0	0.0	1.0	10
4	1.0	1.0	0.5	0
5	-1.0	2.0	0.0	-10
6	-3.0	1.0	0.0	-20
7	-1.0	0.0	0.0	-10
8	1.0	1.0	0.5	0

(2) 非线性 MPC 控制参数的选取

以上述轨迹为参考，在仿真中实现非线性 MPC 轨迹跟踪控制器进行仿真。综合参考文献^[53]和仿真表现，设定仿真中的控制参数如下表。

表 11 仿真中非线性 MPC 采用的控制参数

参数	数值	参数	数值
N	20	t_{pred}	2 s
Δt_{nmpc}	1/50 s	Δt_{pred}	0.1 s
$Q_{p,xy}$	300	$Q_{p,z}$	400
$Q_{v,xyz}$	1	$Q_{q,xyz}$	0.1
$R_{\omega,xyz}$	10	R_{f_c}	10
v_{MAX}	20 m/s	v_{MIN}	- 20 m/s
ω_{MAX}	6 rad/s	ω_{MIN}	- 6 rad/s
QP 求解器	PARTIAL_CONDENSING_HPIPM	积分类型	ERK

参数	数值	参数	数值
Hessian 近似法	GAUSS_NEWTON	NLP 类型	SQP_RTI

简要介绍参数的意义及调节过程。首先在机载计算机算力允许的情况下，非线性 MPC 的控制周期 Δt_{nmpc} 越短越好。经过测试在 TX2 NX 上运行执行实时迭代策略的非线性 MPC 可以稳定在 100 Hz 的运行速率。预测时间 $t_{\text{pred}} = N \times \Delta t_{\text{pred}}$ ，其中预测长度 N 与预测步长 Δt_{pred} 表征了非线性 MPC 的预测能力。如果 N 过大则计算量过大，过小则无法充分发挥非线性 MPC 的优势。而 Δt_{pred} 一般选择的比控制周期 Δt_{nmpc} 要长，这样可以尽可能发挥非线性 MPC 向前预测的优势。而优化目标里 Q 阵和 R 阵的选取方案与 LQR 里类似，其大小是一种相对值，表征了不同状态的重要程度。对于轨迹跟踪控制任务，位置误差是评价效果好坏的最重要因素，因而对于 Q 阵中的位置给了较大参数，对于其他状态量则参数值较小。对于速度与角速度的限幅，可根据实验环境与轨迹参数进行设计。

(3) 轨迹跟踪效果测试

为验证非线性 MPC 的轨迹跟踪效果，分别在低速和高速轨迹上进行测试。低速轨迹针对实验室飞行环境进行设计，以表 10 为路径点，生成轨迹的参数如下表所示。

表 12 低速轨迹参数

属性	数值	单位	属性	数值	单位
最大 X 轴位置	5.0920	m	最小 X 轴位置	-3.0921	m
最大 Y 轴位置	2.2319	m	最小 Y 轴位置	-0.2317	m
最大 Z 轴位置	1.1984	m	最小 Z 轴位置	-0.1985	m
最大速度	1.0309	m/s	最大加速度	0.4569	m/s ²
最大偏航角	20.0052	°	最小偏航角	-20.0041	°

在仿真中使用非线性 MPC 跟踪该轨迹，飞行图像如下图所示。

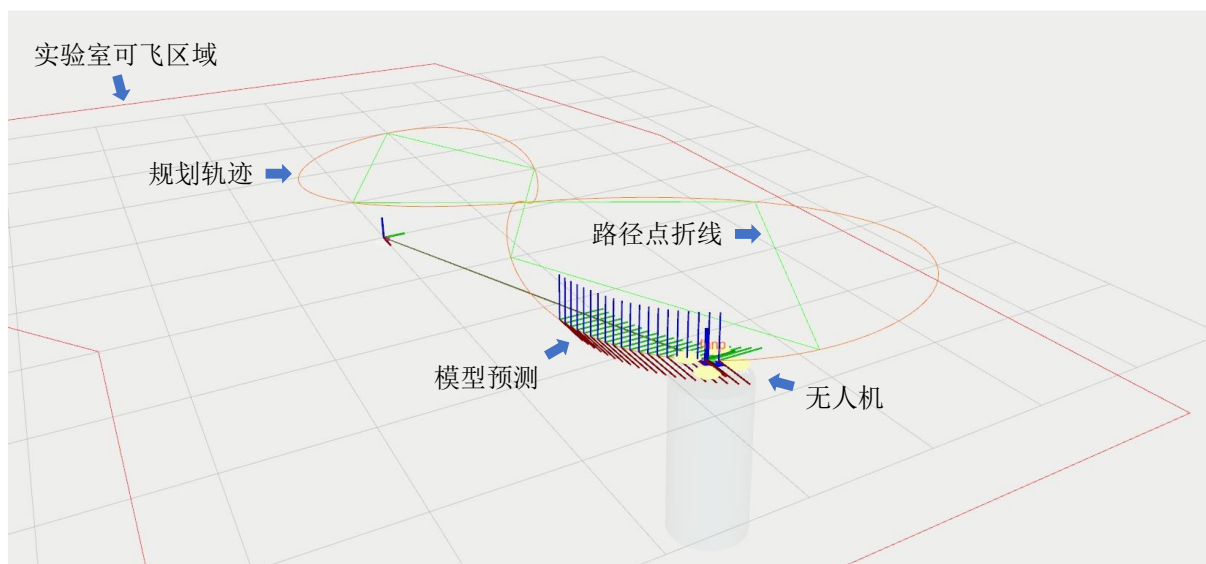


图 39 低速场景下无人机轨迹跟踪仿真效果图

在低速场景下对轨迹的跟踪结果如下图所示。

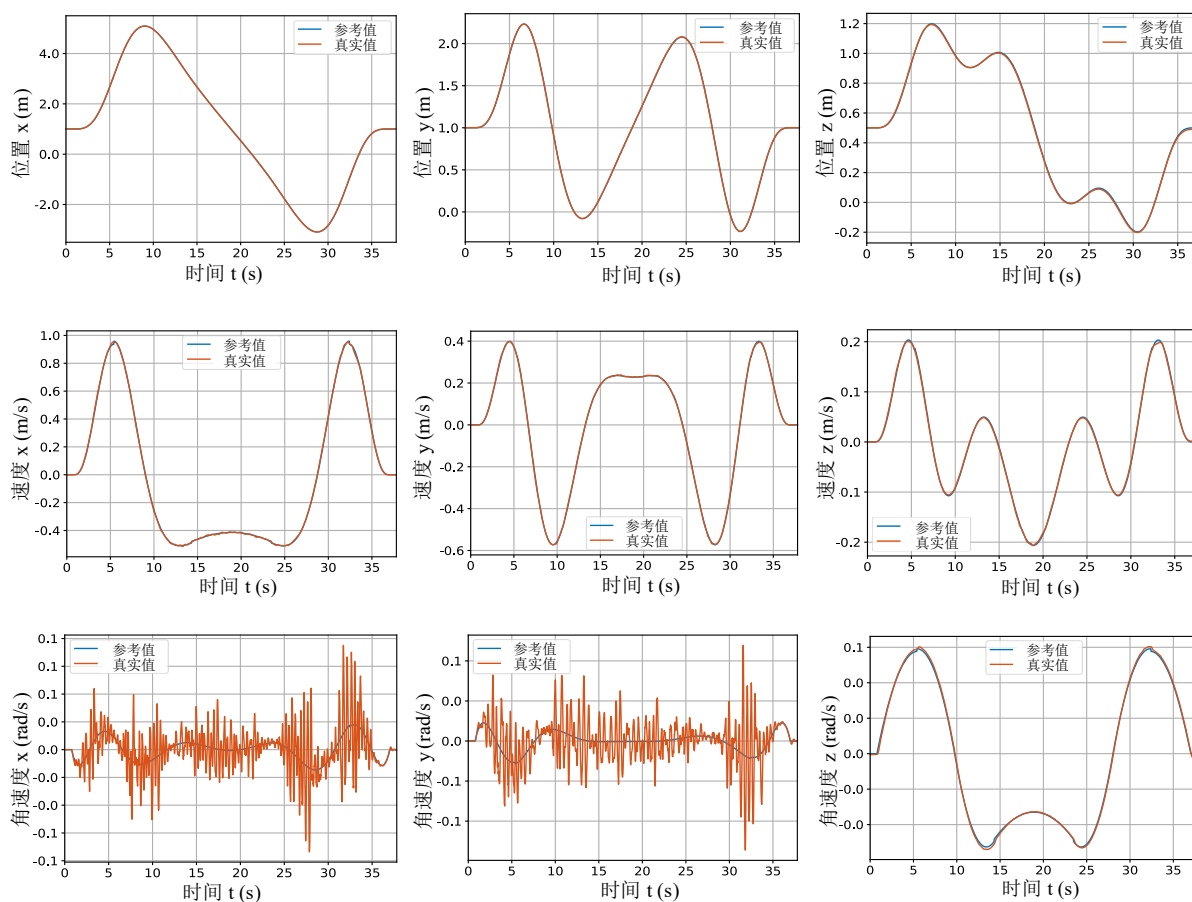


图 40 低速场景下轨迹跟踪仿真结果

从图中可以看出非线性 MPC 可以非常精确地跟踪轨迹。最终对跟踪结果计算均方根误差(Root Mean Square Error, RMSE),得到位置误差为 0.0149 m, 偏航角误差为 0.1547° 。

高速轨迹希望充分发挥无人机的潜力，使得飞行的最大速度达到 10 m/s，其轨迹参数如下表。

表 13 高速轨迹参数

属性	数值	单位	属性	数值	单位
最大 X 轴位置	11.2037	m	最小 X 轴位置	-9.2085	m
最大 Y 轴位置	7.1114	m	最小 Y 轴位置	-5.1225	m
最大 Z 轴位置	8.0611	m	最小 Z 轴位置	1.9375	m
最大速度	9.6141	m/s	最大加速度	16.5302	m/s ²
最大偏航角	0.1021	°	最小偏航角	-0.1006	°

在仿真中使用非线性 MPC 跟踪该轨迹，飞行图像如下图所示。图中除标示规划轨迹、模型预测等元素外，还用坐标轴序列的形式标示了飞行的历史轨迹。

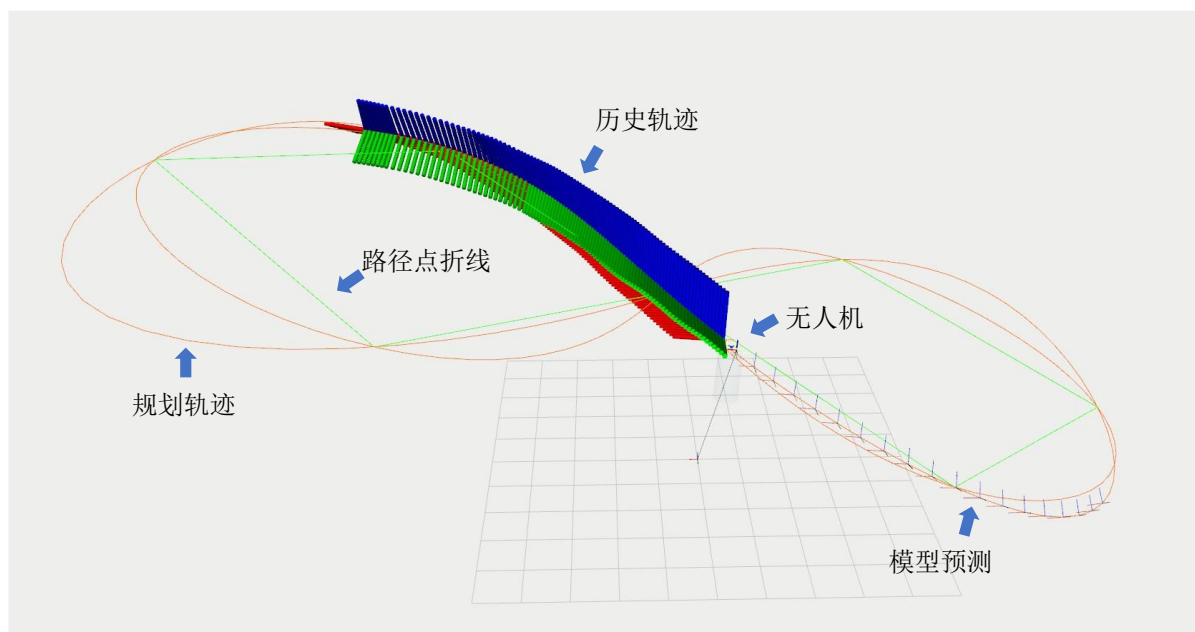


图 41 高速场景下无人机轨迹跟踪仿真效果图

从上图可以看出，与低速轨迹对比，对高速轨迹的预测时间虽然没有改变，但预测的距离延长了许多。这种预测的特性可以帮助无人机提前调整状态，有利于其跟踪曲率较大的轨迹。在高速场景下对轨迹的跟踪结果如下图所示。

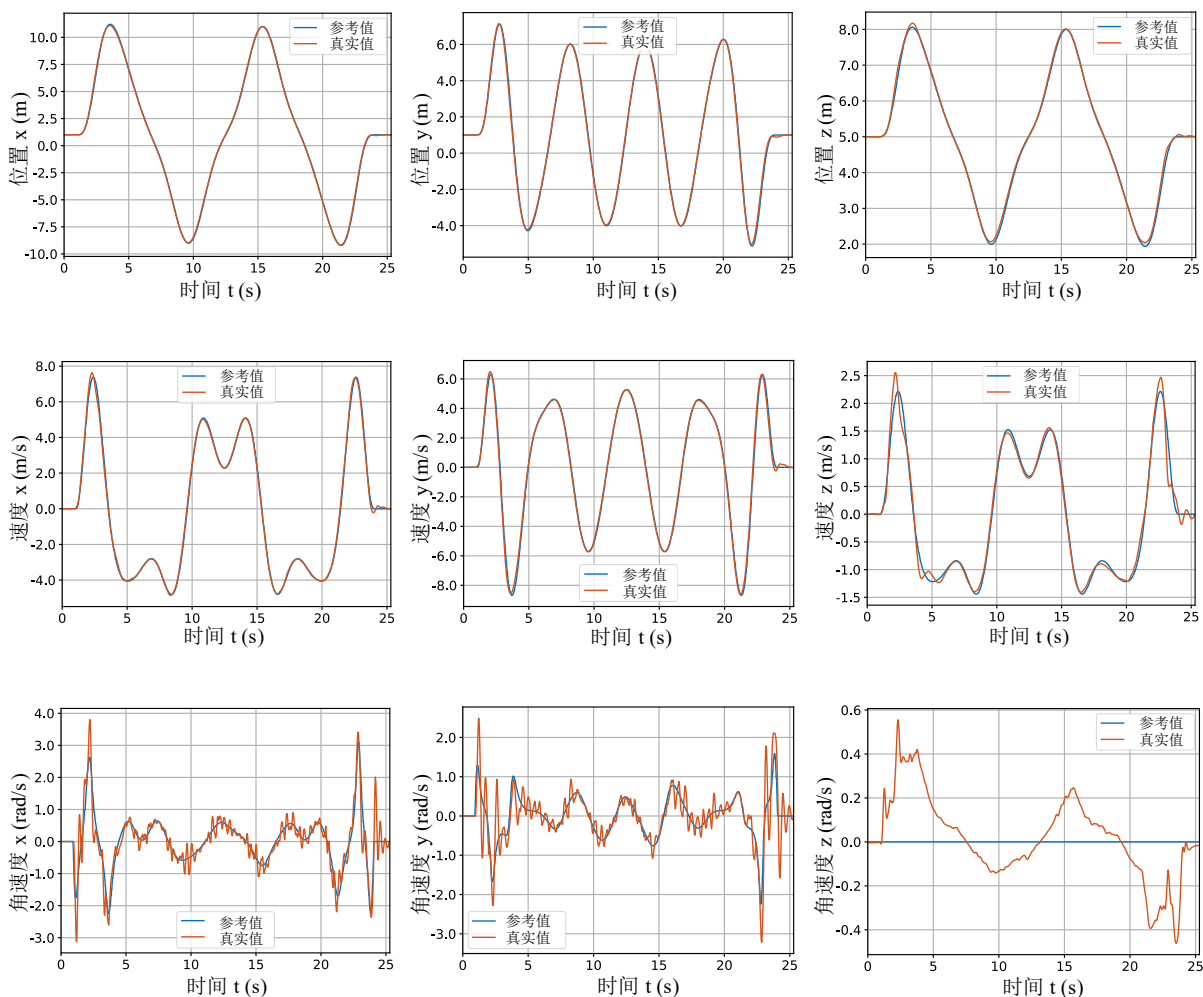


图 42 高速场景下轨迹跟踪仿真结果

从图中可以看出非线性 MPC 可以在高速情况下较为精确地跟踪轨迹。最终对跟踪结果计算 RMSE 误差，得到位置误差为 0.1262 m，偏航角误差为 8.9610° 。与低速轨迹相比，高速轨迹的最大速度扩大约 10 倍，最大加速度扩大约 36 倍，因而位置误差也扩大较多。该跟踪误差与最新研究^[53]较为接近，这表明当前非线性 MPC 的参数较为合适。

在仿真中非线性 MPC 可以较准确地对四旋翼实现轨迹跟踪，后续可继续实物飞行。

4.4.2 卡尔曼滤波悬停油门观测

系统噪声协方差矩阵 Q 和量测噪声协方差矩阵 R 对于卡尔曼滤波的效果至关重要，下面简要介绍一下 Q 阵和 R 阵的参数选取过程。从物理意义上说，两者取值的相对大小表明更相信系统还是更相信量测。假设各个过程噪声之间相互独立， Q 阵可转化为对角矩阵。对于状态方程，油门值和机体重量均较为准确，因而两个方程产生的误差均产生

于对悬停油门的估计误差，进而选取相同数量级的参数，即 $q_{11} = q_{22} = 0.1$ 。而 R 阵与量测噪声有关，PX4 固件中将加速度计的噪声协方差设为 0.1225，但根据实际表现将其扩大十倍，设为 $r_{11} = 1.225$ 。

按照上述设置实现卡尔曼滤波悬停油门观测器，在仿真中测试上述参数对应的滤波结果如下图所示。

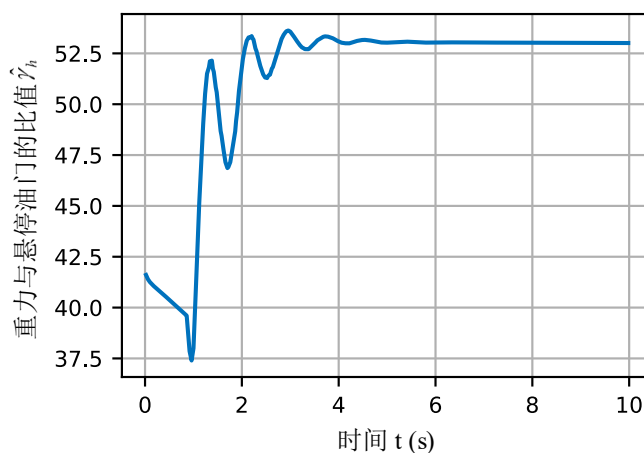


图 43 仿真中的悬停油门观测量随时间的变化

可以看到经过几秒钟时间，观测器成功估计了重力与悬停油门的比值。在实际飞行中，同样验证上述参数可以较好地估计无人机的悬停油门。

4.5 实物验证

4.5.1 实验过程

在北航杭州创新研究院开展实物试验，实验飞行场景如下图所示。实验采用动作捕捉装置（Motion Capture System, MoCap）提供位置和姿态信息，控制相关的程序均运行在机载计算机 TX2 NX 上。试验开始后，首先由操纵手通过遥控器将 PX4 的模式切换为 OFFBOARD，接着无人机会在程序的控制下飞到起始点。当控制程序检测到无人机飞到起始点后，会开始向无人机发送控制目标点，控制无人机飞行。

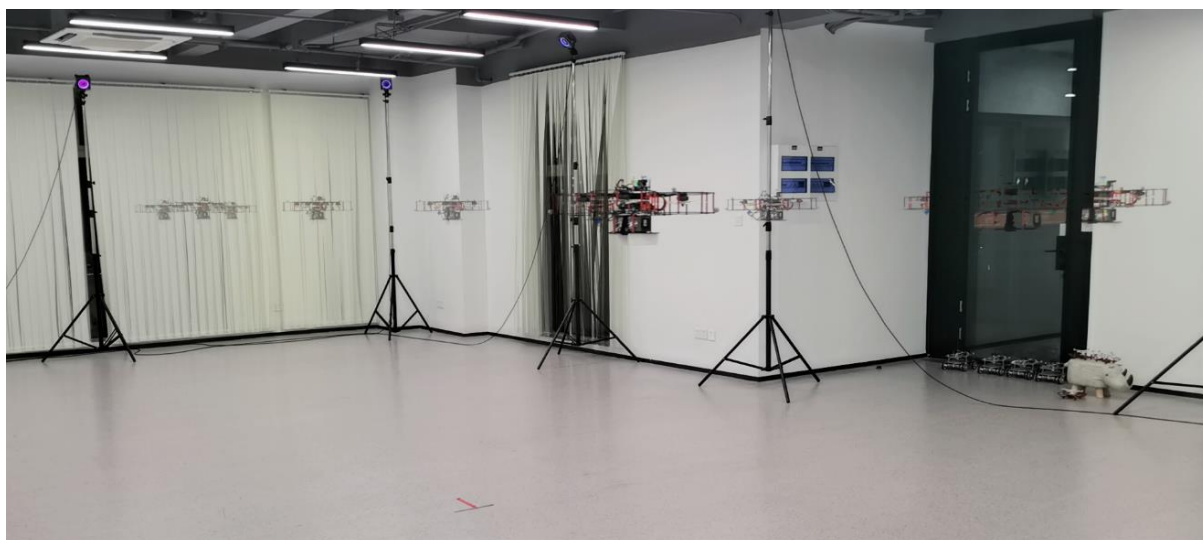


图 44 非线性 MPC 轨迹跟踪实物飞行实验图

4.5.2 实验结果

下图是实物飞行过程中在 RVIZ 上展现的场景。本实验采用的路径点为一长八米，宽四米的八字轨迹，如下图中绿线所示。根据 Minimum Snap 生成的轨迹如下图黄线所示。非线性 MPC 使用 100 Hz 运行，其余控制参数与仿真相同，每次预测的模型如下图红色箭头所示。此外控制目标点的位置用粉色球表示，实物四旋翼用大坐标轴表示，实验室可飞区域用红色折线表示。

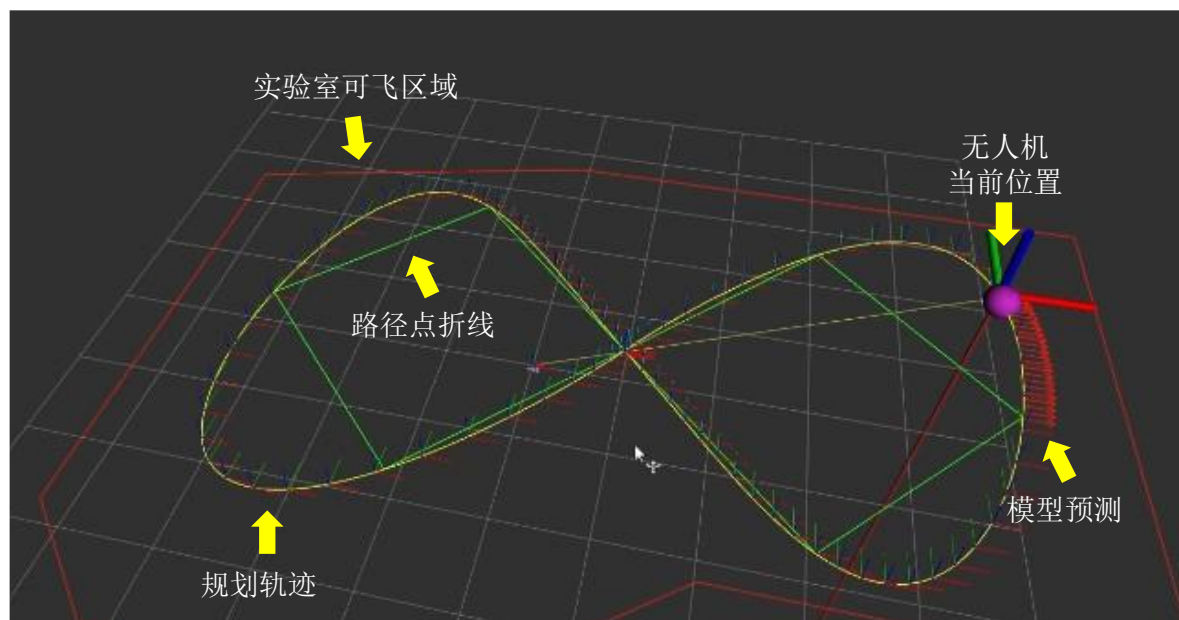


图 45 非线性 MPC 轨迹跟踪实物飞行信息可视化

经过多次飞行取平均值，轨迹跟踪的位置误差约为 0.07 m。经过实物验证，本章介

绍的轨迹生成、非线性 MPC 轨迹跟踪均可以正常发挥作用,产生较好的轨迹跟踪效果。

4.6 本章小结

本章介绍了针对单无人机的非线性 MPC 轨迹跟踪控制方法。首先介绍四旋翼领域经典的 Minimum Snap 轨迹生成算法,该算法可以生成三轴位置和偏航角随时间变化的光滑曲线。接着介绍如何通过四旋翼的微分平坦特性将多项式轨迹转换为全状态轨迹以作为非线性 MPC 的跟踪目标。然后介绍非线性 MPC 轨迹跟踪控制方法,包括采用的控制模型、构造的滚动时域优化问题和非线性 MPC 的运算加速技巧。之后描述悬停油门观测器的设计,将合力指令转化为油门指令。最后展示三维仿真和实物实验的效果,说明非线性 MPC 可在实物飞机上以 100 Hz 速率运行。

第五章 多无人机非线性 MPC 密集编队跟踪控制算法

如果仅采用上一章的轨迹跟踪控制算法，当多架无人机飞行穿越狭小的空间时可能会受到下洗气流干扰，然后偏离原始路径撞到障碍。然而，如果将第三章的下洗气流估计纳入控制器设计，便可以保持原始飞行轨迹穿越障碍。本章主要将第三章和第四章结合以设计算法，同时将非线性 MPC 轨迹跟踪控制算法拓展到多无人机编队飞行领域，具体内容如下图所示。

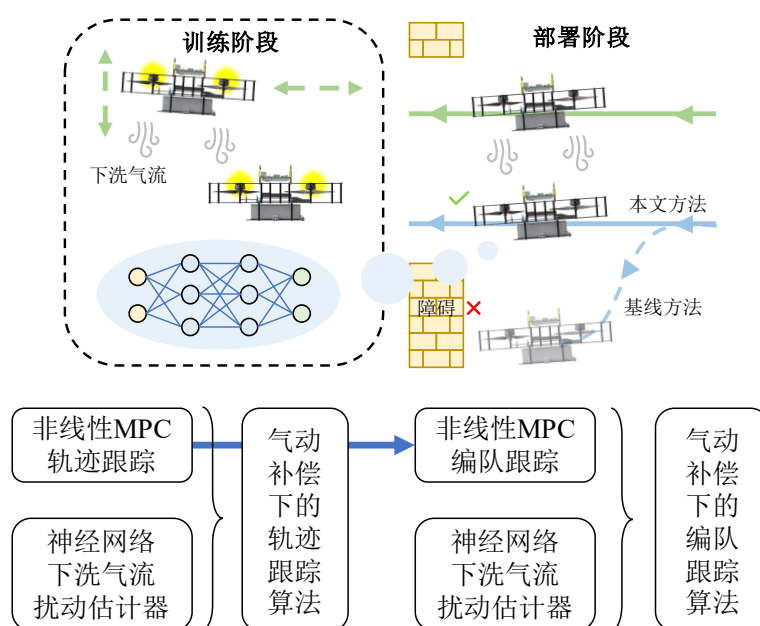


图 46 第五章内容逻辑图

首先将第三章设计的神经网络气动扰动估计器和第四章设计的轨迹跟踪算法结合，提出气动补偿下的无人机轨迹跟踪控制算法。接着定义编队向量、通讯拓扑关系、编队机制，将第四章设计的单无人机轨迹跟踪算法拓展为多无人机编队跟踪算法。然后将气动补偿引入编队跟踪，提出气动补偿下的多无人机编队跟踪控制算法。最后展示了仿真与实物实验的结果。

5.1 气动补偿下的无人机轨迹跟踪控制算法

尽管在第三章使用神经网络对下洗气流扰动的三轴力和力矩均进行了估计，但在本章只用到了三轴扰动力，这是出于两个原因：第一，本文设计的非线性 MPC 算法以角速度指令为控制输入，而底层的 PX4 飞控仍然以 1 kHz 运行着一个与陀螺仪量测数据同

频率的 PID 角速度控制器。在实际飞行中发现该控制器可以很快补偿掉力矩的影响。如果去掉它转而采用非线性 MPC 对力矩进行补偿，控制频率会降低从而不能充分利用陀螺仪量测数据，可能造成控制效果下降。第二，与扰动力相比，扰动力矩的影响较小。

气动补偿下的非线性 MPC 轨迹跟踪算法主要通过修改模型的方式来将下洗气流扰动估计纳入进算法框架，核心思想如下图所示。

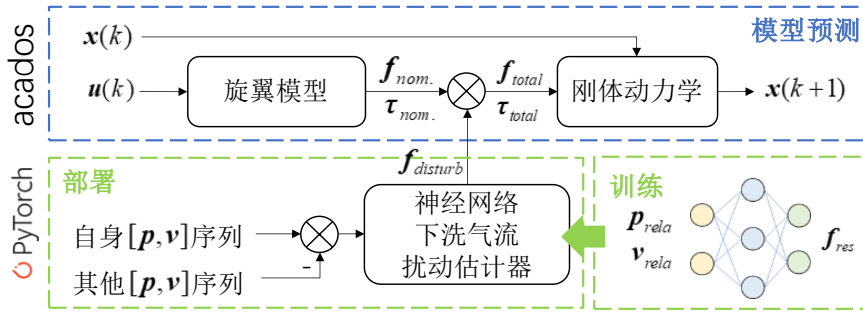


图 47 神经网络气动补偿对非线性 MPC 模型的改进

假设两架无人机均采用非线性 MPC 轨迹跟踪控制方法。在算法运行过程中，每架无人机可以通过通讯得到其他无人机经由 MPC 预测的未来轨迹，并和自身由 MPC 预测的未来轨迹作差，得到未来一段时间的相对状态关系，接着使用神经网络通过该关系预测未来的下洗气流扰动。假设控制速率较高，那么由上一轮状态得到的神经网络扰动预测在下一轮可看作是不变的，因而可以直接将该扰动以参数的方式传入下一轮非线性 MPC 的优化过程。扰动力作用于模型中的刚体动力学，从而实现向控制算法引入扰动的目的。

总结为公式，气动补偿下的非线性 MPC 的控制模型变为

$$\begin{cases} {}^I \dot{\mathbf{p}} = {}^I \mathbf{v} \\ {}^I \dot{\mathbf{v}} = ({}^I \mathbf{R}(\mathbf{q}) \cdot {}^B \mathbf{f}_u + {}^I \mathbf{f}_d) / m + {}^I \mathbf{g}, \\ {}^B \dot{\mathbf{q}} = 1/2 \cdot {}^B \mathbf{q} \circ \mathcal{V}^* ({}^B \boldsymbol{\omega}) \end{cases} \quad (5.1)$$

其中 ${}^I \mathbf{f}_d$ 由第三章设计的神经网络观测器根据两飞机的相对状态产生。注意在实际运行中，需要针对 MPC 的预测长度 N 生成一系列的扰动预测 ${}^I \mathbf{f}_{d,i} = \text{DNN}({}^I \delta \mathbf{p}_i, {}^I \delta \mathbf{v}_i)$, $i = 1, 2, \dots, N$ ，其余算法流程与第四章相同。

5.2 分布式非线性 MPC 编队跟踪控制算法

本节介绍使用分布式(Distributed)非线性 MPC 进行编队跟踪的思想，如下图所示，

每架飞机通过与邻近节点的信息交互和状态观测共同完成控制任务。与去中心化 (Decentralized) MPC 不同, 分布式 MPC 包含了算法间的信息交互。

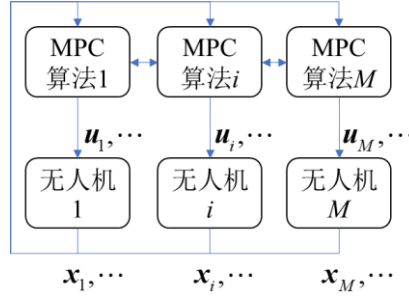


图 48 分布式 MPC 的基本思想

对于 M 个同构个体组成的编队跟踪控制问题, 其控制目标为每个个体在位置层面保持队形的同时跟踪一个期望轨迹。这个队形对应的数学表达式被称为编队向量, 记做 $\mathbf{h}_i(t) \in \mathbb{R}^3, i=1, \dots, M$ 。本文定义的编队参考以领导者作为坐标原点。

编队控制的另一个重要元素是个体之间的通讯关系, 即作用拓扑 (Interaction Topology), 可以用邻接矩阵 (Adjacency Matrix) $\mathbf{S}(t) \in \mathbb{R}^{M \times M}$ 来表示, 其中每个元素 $[\mathbf{S}(t)]_{ij} = s_{ij}$ 表示个体 i 能否接收到个体 j 的信息, 如果能收到即为 1, 不能收到即为 0。例如, 针对下图这样的通讯关系,

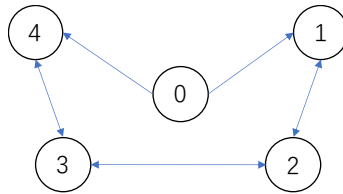


图 49 通讯拓扑示意图

其邻接矩阵可写为

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}。 \quad (5.2)$$

本文认为集群系统的通讯关系与通讯距离 r_{Comm} 有关, 若两架飞机间的距离满足 $\|\mathbf{p}_i - \mathbf{p}_j\| \leq r_{\text{Comm}}$, 则有 $[\mathbf{S}(t)]_{ij} = [\mathbf{S}(t)]_{ji} = 1$, 因此本文的邻接矩阵是时变的。将个体 i 满足 $\|\mathbf{p}_i - \mathbf{p}_j\| \leq r_{\text{Comm}}$ 的邻居节点集合记为 $N_i, i=1, \dots, M$ 。此外, 将 N_i 中距离自身最近的两个

个体组成的集合记为 $N_i^{\min 2}$ 。

本文采用领导者-跟随者的方案实现编队跟踪。假设集群中存在一个计算能力较强的领导者可以接收由轨迹规划生成的全状态轨迹 $\mathbf{x}_r(t)$ 并按照第四章的内容进行轨迹跟踪，同时该领导者可将从决策模块接收到的编队向量 $\mathbf{h}_i(t), i=1, \dots, N$ 以 f_h 的频率传递给所有跟随者。在编队过程中，领导者只负责向邻居节点发送自身由 MPC 产生的预测轨迹信息，因而其非线性 MPC 控制律与上一章相同。而跟随者需要获得邻居的状态信息，并将彼此的状态差异与期望实现的编队相对位置 $\mathbf{h}_j(t) - \mathbf{h}_i(t)$ 进行比对以调整自身行为。

参考一致性理论中为编队跟踪设定的控制目标

$$\lim_{t \rightarrow \infty} (\mathbf{x}_i(t) - \mathbf{h}_i(t) - \mathbf{r}(t)) = \mathbf{0}, \quad (5.3)$$

将个体 i 跟踪的期望轨迹和期望状态修改为邻居实际轨迹与实际状态的平均值：

$$\begin{cases} \mathbf{p}_r = \sum_{j \in N_i^{\min 2}} \mathbf{p}_j / 2 + \mathbf{h}_i - \mathbf{h}_j \\ \mathbf{x}_r = \sum_{j \in N_i^{\min 2}} \mathbf{x}_j / 2, \text{ 除 } \mathbf{p}_r \text{ 外的 } \mathbf{x}_r, \\ \mathbf{u}_r = \sum_{j \in N_i^{\min 2}} \mathbf{u}_j / 2 \end{cases} \quad (5.4)$$

其中 \mathbf{p}_r 表示状态中的位置序列，加入了编队参考 $\mathbf{h}_i - \mathbf{h}_j$ 以进行偏移。注意此处的邻居节点集合 $N_i^{\min 2}$ 仅包含距离最近的两个节点，这样既可以减少通讯负担，又可以避免编队误差逐层传递。在非线性 MPC 方面，依然按照上一章介绍的模型

$$\begin{cases} {}^I \dot{\mathbf{p}} = {}^I \mathbf{v} \\ {}^I \dot{\mathbf{v}} = {}^I \mathbf{R}(\mathbf{q}) \cdot {}^B \mathbf{f}_u / m + {}^I \mathbf{g} \\ {}^B \dot{\mathbf{q}} = 1/2 \cdot {}^B \mathbf{q} \circ \mathcal{V}^* ({}^B \boldsymbol{\omega}) \end{cases} \quad (5.5)$$

和优化目标

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} \sum_{k=1}^{N-1} (\bar{\mathbf{x}}_k^T \mathbf{Q}_x \bar{\mathbf{x}}_k + \bar{\mathbf{u}}_k^T \mathbf{R}_u \bar{\mathbf{u}}_k) + \bar{\mathbf{x}}_N^T \mathbf{Q}_N \bar{\mathbf{x}}_N \quad (5.6)$$

进行优化，其余设置与上一章完全相同。总结跟随者的编队跟踪算法如下：

Step 0: 跟随者获得编队向量 $\mathbf{h}(t)$

While Loop:

Step 1: 根据机间相对距离 $\delta \mathbf{p}$ 和通信半径 r_{Comm} 获得最近邻居节点集合 $N_i^{\min 2}$

Step 2: 从 $N_i^{\min 2}$ 所含机体获得由 MPC 预测的轨迹集合 $\{\mathbf{x}_{jk} \mid j \in N_i^{\min 2}\}$

Step 3: 按照公式(5.4)构造期望的控制目标, 包括期望轨迹和期望状态

Step 4: 按照模型(5.5)和优化目标(5.6)进行非线性 MPC 编队跟踪

End While

按照上述编队方案, 在飞行过程中跟随者无需知道全局轨迹指令即可形成队形。飞行中的队形不是一成不变的, 需要根据环境信息调整队形, 因而编队向量是时变函数。在工程实践中发现, 若编队向量的切换为阶跃函数, 那么编队形成过程中的振荡较大, 因而加入一个时域低通滤波器来缓和编队向量的改变, 具体形式如下:

$$\mathbf{h}(k) = \alpha \cdot \mathbf{h}(k-1) + (1-\alpha) \cdot \hat{\mathbf{h}}(k), \quad (5.7)$$

其中 $\alpha = 0.8$ 为滤波系数, $\hat{\mathbf{h}}(k)$ 为编队向量目标函数, $\mathbf{h}(k)$ 为滤波后的编队向量函数。该低通滤波器的频率与编队向量广播频率 f_h 相同, 仿真验证该设置可以较好实现编队切换。

5.3 气动补偿下的多机密集编队跟踪控制算法

上一节根据分布式非线性 MPC 设计了编队跟踪控制算法。然而当无人机为穿越障碍实现密集编队时, 无人机间距离较近, 极有可能受到机间下洗气流的影响。本节将第三章设计的气动扰动观测器与编队跟踪控制算法结合, 提出气动补偿下的多无人机编队跟踪控制算法。

与两架飞机不同, 多无人机间存在下洗气流的叠加影响。根据 3.6 节的内容, 本文将这种叠加影响简化为线性加和关系, 即对第 i 架无人机, 将其下洗气流扰动定义为

$${}^i \mathbf{f}_d = \sum_{j \in N_i^D} \text{DNN}({}^i \delta \mathbf{p}_j, {}^i \delta \mathbf{v}_j), \quad (5.8)$$

里边的 $N_i^D, i=1, \dots, M, \text{s.t. } \|\mathbf{p}_i - \mathbf{p}_j\| \leq r_{\text{Downwash}}$ 表示与无人机 i 存在下洗气流作用的全部邻居节点。在实物系统中, 下洗气流的作用范围 r_{Downwash} 往往远小于通讯作用范围 r_{Comm} , 因而存在机间气动扰动的个体必定可以经过通讯获得彼此由 MPC 预测的轨迹。

将考虑机间气动扰动的跟随者编队跟踪算法梳理如下:

Step 0: 跟随者获得编队向量 $\mathbf{h}(t)$

While Loop:

Step 1: 根据无人机相对距离 $\delta \mathbf{p}$ 和通信半径 r_{Comm} 获得最近邻居节点集合 $N_i^{\text{min}2}$

Step 2: 根据下洗气流作用半径 r_{Downwash} 获得存在气动扰动的邻居节点集合 N_i^D

Step 3: 从 $N_i^{\min 2}$ 和 N_i^D 所含机体获得由 MPC 预测的轨迹集合 $\{x_{jk} \mid j \in N_i^{\min 2} \cup N_i^D\}$

Step 4: 按照公式(5.4)构造期望的控制目标, 包括期望轨迹和期望状态

Step 5: 按照公式(5.8)计算多无人机存在的机间下洗气流扰动 $^l f_d$

Step 6: 按照模型(5.1)和优化目标(5.6)进行非线性 MPC 编队跟踪

End While

按照上述编队方案, 跟随者无需知道全局轨迹指令即可形成队形, 同时可以在下洗气流干扰下完成密集编队跟踪飞行。

5.4 仿真验证

仿真主要验证编队算法, 领导者沿着上一章介绍的慢速轨迹进行飞行, 同时向跟随者发送编队参考。跟随者全过程接收领导者通过非线性 MPC 生成的预测状态进行编队保持。在飞行过程中, 为实现下洗气流叠加效果, 无人机的队形会不断变化, 当其水平位置与轨迹原点产生 2 米的偏移时, 无人机会组成存在气动扰动的密集编队。

5.4.1 多机非线性 MPC 编队跟踪

两架无人机进行编队跟踪的仿真效果如下图所示。图中上方飞机在队形切换点从侧上方飞到下方飞机的正上方, 造成下洗气流扰动影响, 使得下方飞机偏离原始轨迹。编队算法和队形切换算法得到验证。

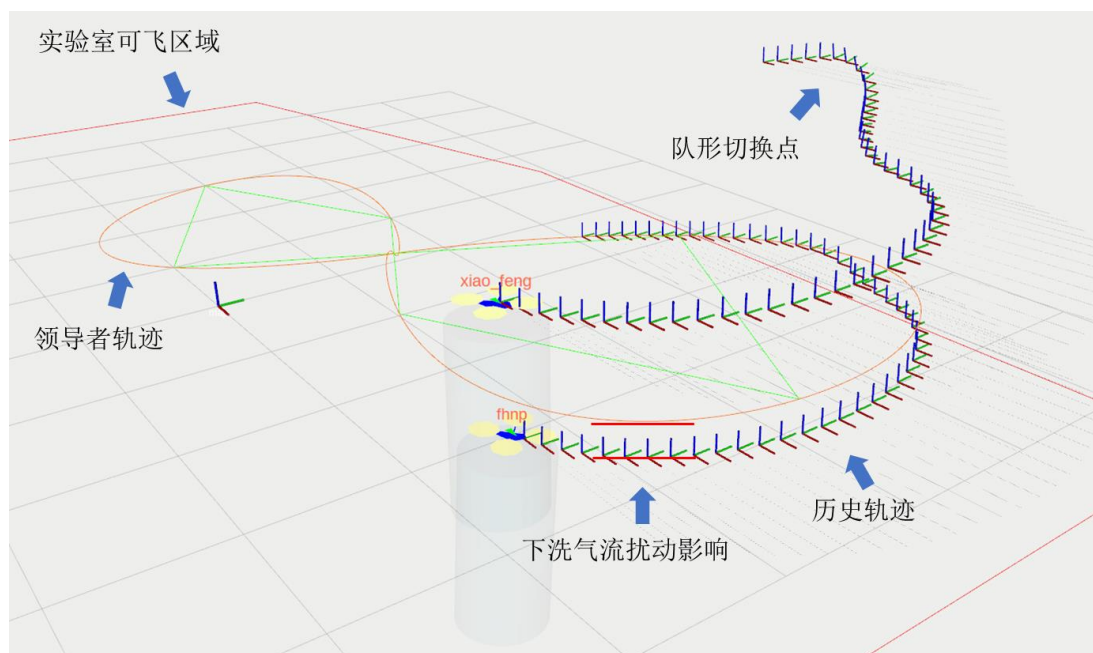


图 50 两架无人机队形切换仿真效果图

为验证多无人机编队跟踪，开展对三架无人机的仿真实验如下图所示。

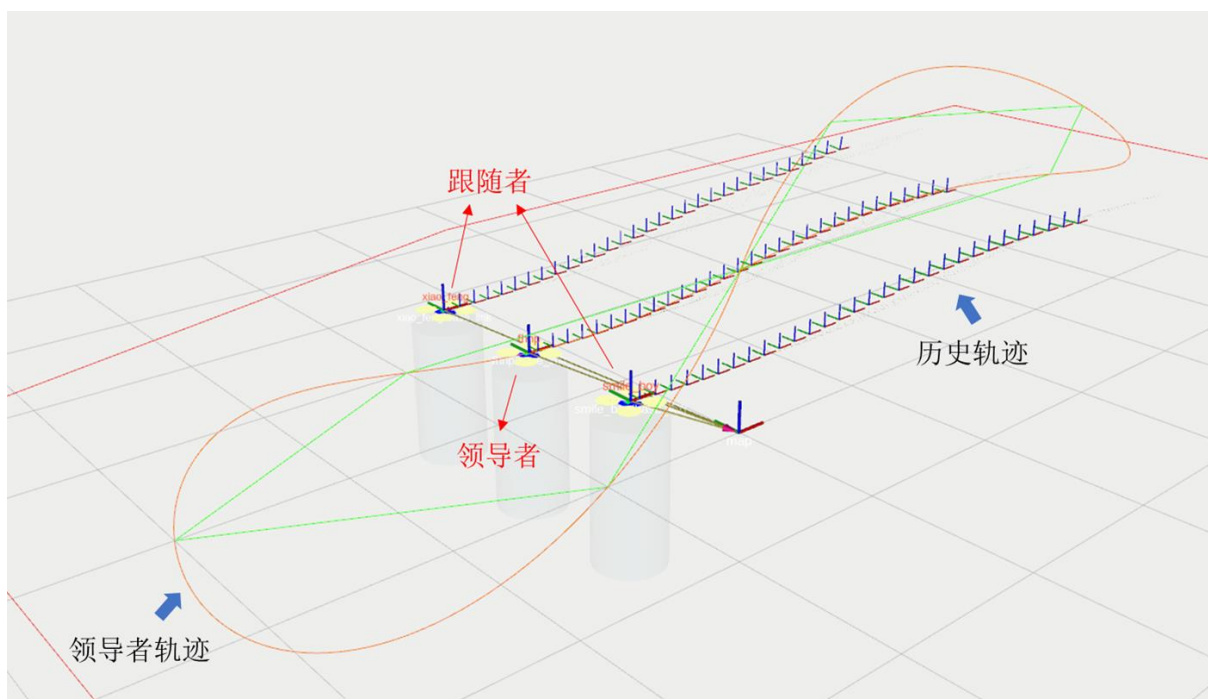


图 51 三架无人机编队跟踪仿真效果图

领导者同样沿着上一章介绍的慢速 8 字轨迹进行跟踪，两个跟随者会在与双机飞行相同的位置进行编队切换，切换后一架跟随者无人机会飞到领导者无人机上方造成下洗气流影响，如下图所示。

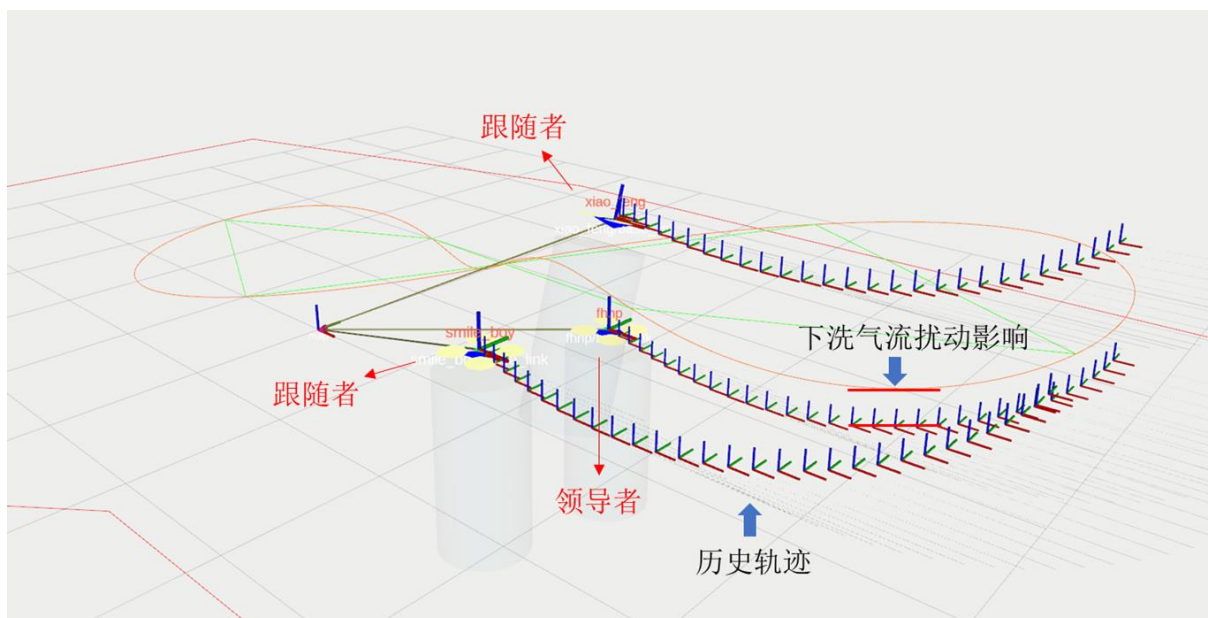


图 52 三架无人机编队切换仿真效果图

三架无人机飞行中的编队跟踪仿真结果如下图所示。

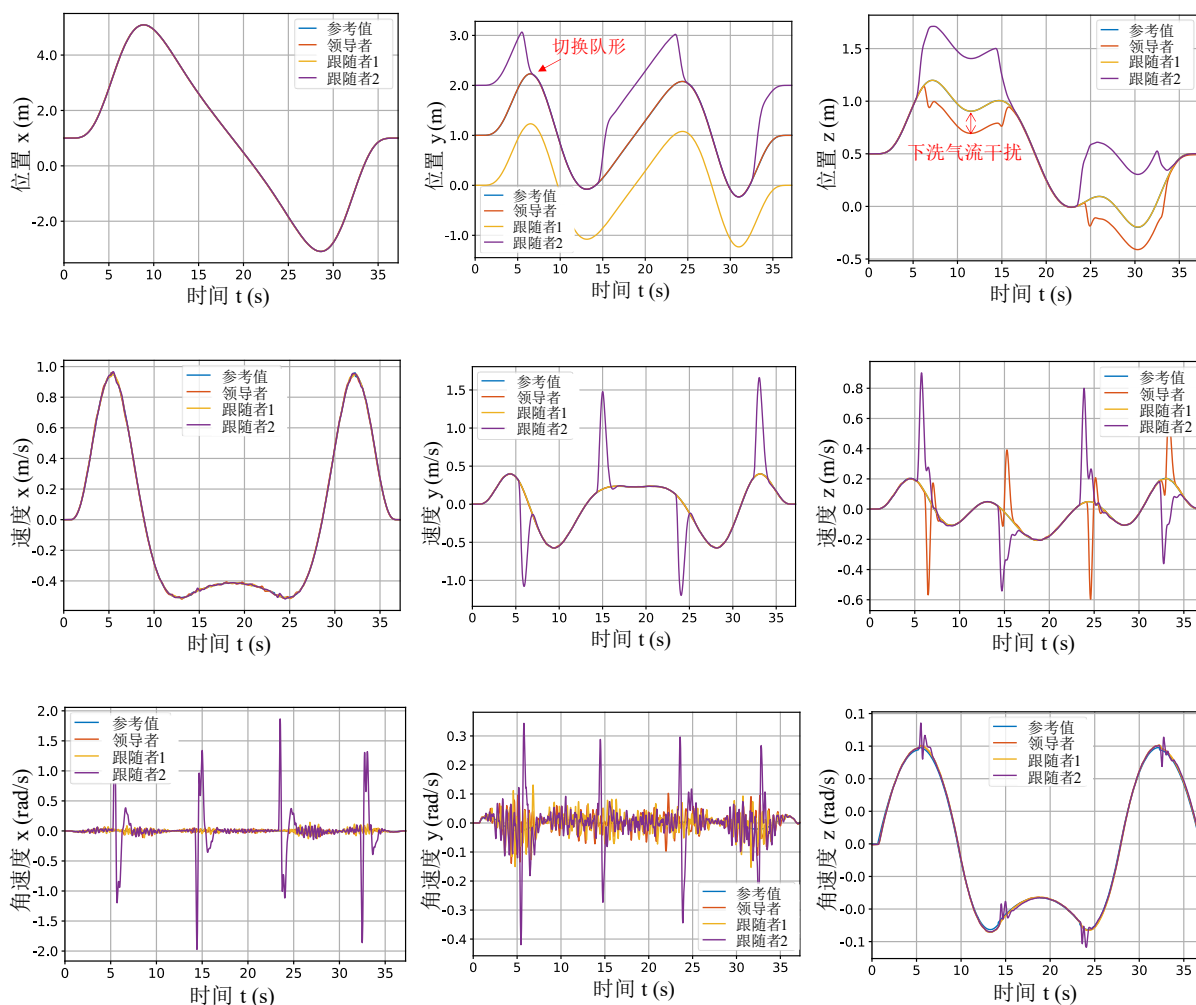


图 53 三架无人机编队跟踪仿真结果

可以看到切换队形后，领导者受到下洗气流的干扰偏离参考轨迹。计算得到领导者位置跟踪的 RMSE 均方根误差为 0.1506 m ，偏航角 RMSE 为 0.1559° ，与上一章得到的结果 0.0149 m 、 0.1547° 相比位置误差较大，这由下洗气流干扰造成。对于存在编队切换的跟随者 1 无人机，编队跟踪的 RMSE 为 0.2121 m ；对于不存在编队切换的跟随者 2 无人机，编队跟踪的 RMSE 为 0.0384 m 。前者误差较大，推测由编队切换的过程产生。

5.4.2 气动补偿下的密集编队跟踪

在编队跟踪算法中加入对下洗气流的扰动估计，此外三架无人机的编队跟踪过程与上一条完全相同。切换队形后，一架跟随者无人机会飞到领导者无人机的上方造成下洗气流影响。由于扰动估计的存在，如下图所示被影响的无人机依然能够以较高精度跟踪轨迹。

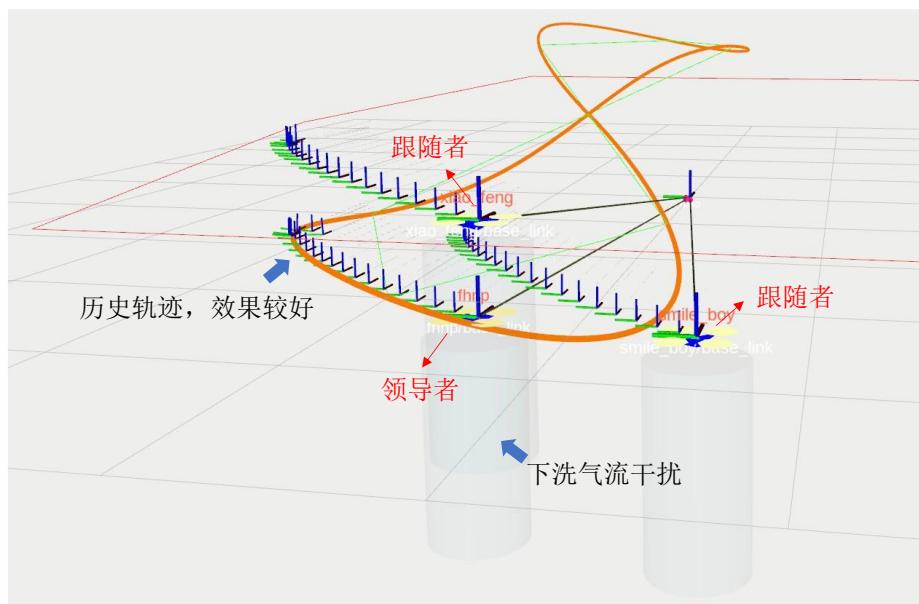


图 54 气动补偿下的三架无人机编队仿真效果图

三架无人机飞行中的编队跟踪仿真结果如下图所示。

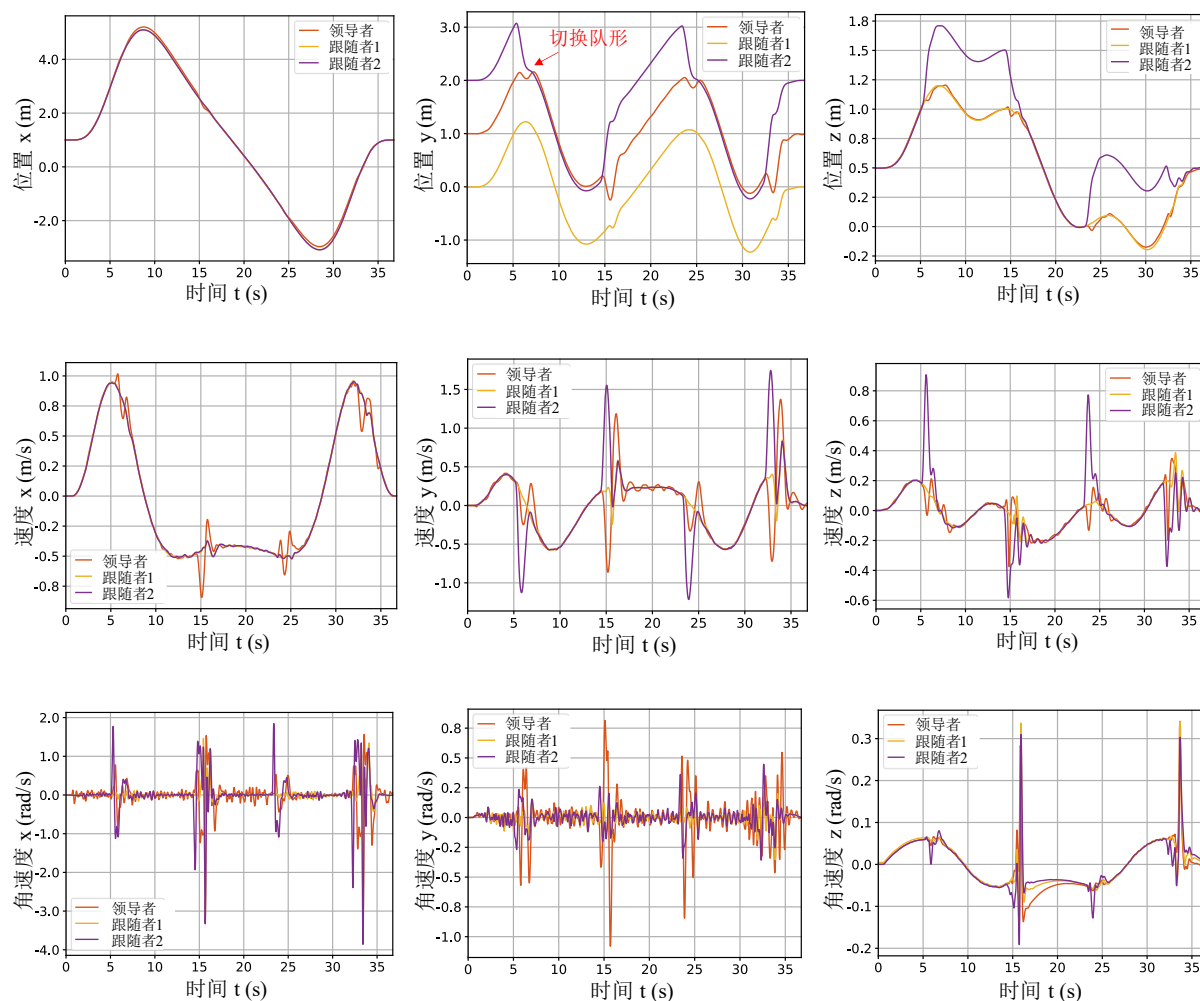


图 55 加入气动扰动补偿的三架无人机编队跟踪仿真结果

可以看到切换队形后，领导者受到下洗气流的干扰，但因为扰动估计的存在，Z轴位置跟踪精度依然较高。然而神经网络对扰动的估计非常依赖彼此的相对位置，在实际飞行中很有可能出现扰动力估计与相对位置错配的情况，造成精度下降。同时，仿真中采用的神经网络经由实际飞行数据训练得到，其输出与仿真中的下洗气流区别较大，因而增大了编队跟踪误差。仿真中计算得到领导者位置跟踪的 RMSE 均方根误差为 0.1593 m，偏航角 RMSE 为 3.1672° ；对于飞到上方的跟随者 1 无人机，编队跟踪位置 RMSE 为 0.2511 m；对于不存在编队切换的跟随者无人机，编队跟踪的 RMSE 为 0.1591 m。

接下来在实物中验证气动扰动观测器与非线性模型预测控制器的结合效果。

5.5 实物验证

5.5.1 实验过程

在北航杭州创新研究院开展实物试验，实物飞行场景如下图所示，其中红线为参考轨迹，黄线为实际飞行轨迹。实验采用动作捕捉装置提供位置和姿态信息，控制相关程序和神经网络扰动估计程序均运行在机载计算机上。实验开始后，首先由两个操纵手将飞机飞到起点附近，同时通过遥控器将 PX4 的模式切换为 OFFBOARD，接着无人机会在程序的控制下飞到起始点。当控制程序检测到无人机飞到起始点后，开始向无人机发送目标点，控制无人机飞行。

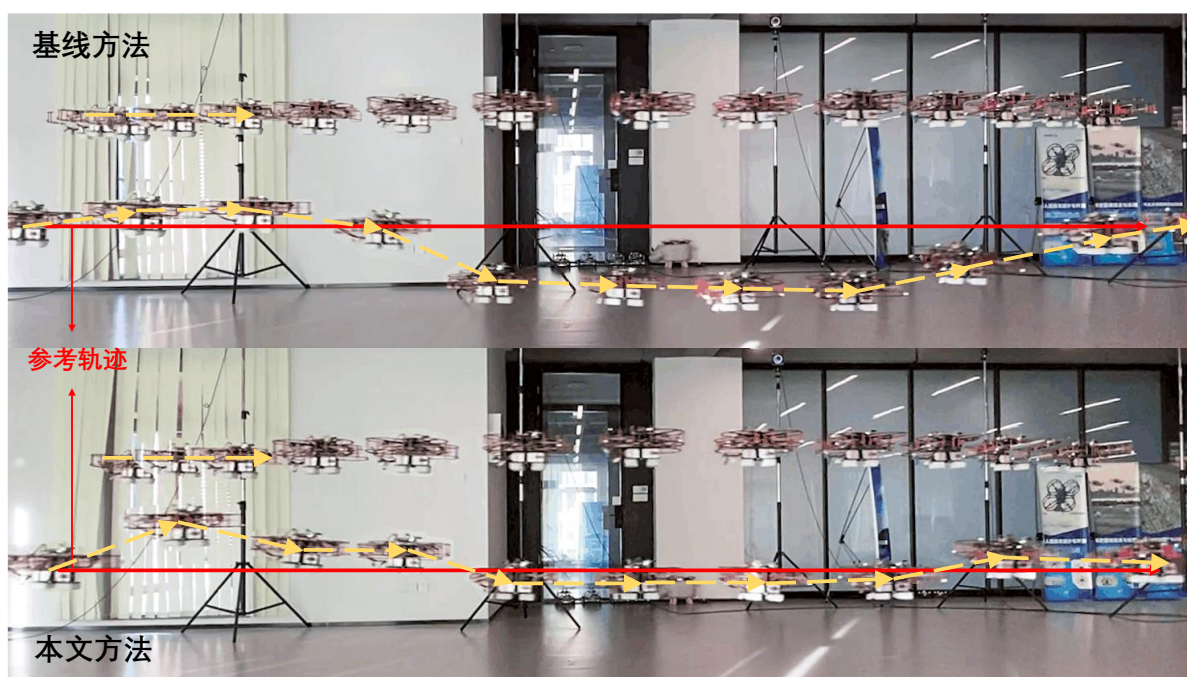


图 56 气动补偿下的非线性 MPC 轨迹跟踪实物实验

5.5.2 实验结果

两架无人机的飞行轨迹如下图所示。无人机沿着不同高度的轨迹从左向右同向飞行，在中段会出现一段交叠区域（橙色阴影），下方无人机会受到较强的下洗气流影响。无人机飞行所跟踪的参考轨迹如下图所示虚线所示，采用基线方法与本章方法所执行的实际飞行轨迹由蓝色线与红色线表示。

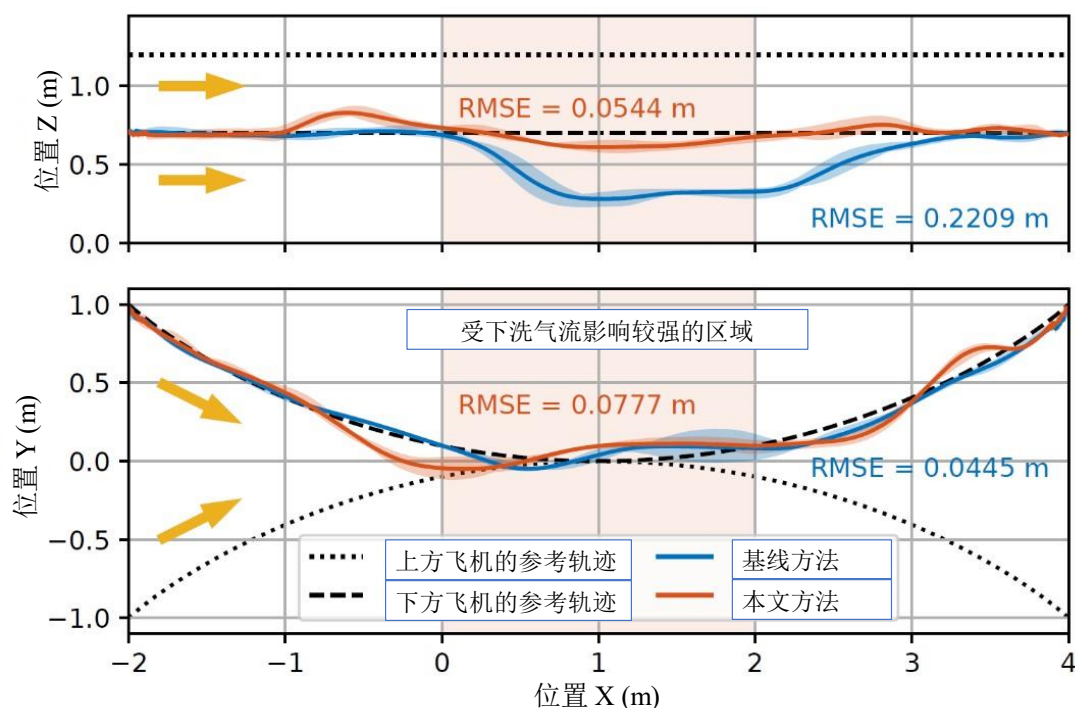


图 57 气动补偿下的非线性 MPC 轨迹跟踪实物实验结果

图中无人机的飞行轨迹由三次实验取平均值得到，实线表示平均值，阴影表示这三段轨迹的极值。本文实验采用的基线方法为不含扰动观测器的非线性 MPC 轨迹跟踪控制器。从上图可以看出，使用基线方法的四旋翼受到严重干扰并偏离参考轨迹，而本文所提出的方法显著减少了干扰的影响，将轨迹跟踪的 RMSE 均方根误差降低了 75.37%，这验证了该方法对下洗气流扰动飞行的有效性。从图中还可以观察到，下方的四旋翼在进入下洗区域之前突然出现了跳跃，然后被压到了参考高度。这种现象是由于神经网络预测的不准确性造成的：网络预测的干扰实际没有发生，因而没有相应的平衡力，这导致下方的四旋翼出现了意外的升高。这种不准确性还会导致水平方向上轨迹跟踪的波动，同样可以在上图观察到。因此在尝试将神经网络预测与非线性 MPC 结合以提高跟踪精度时，必须将神经网络预测可能产生的不准确性考虑在内。

5.6 本章小结

本章是对第三章和第四章算法的融合和扩展。首先直接将第三章设计的神经网络扰动观测器以模型的方式与第四章进行融合，对单无人机设计下洗气流扰动下的轨迹跟踪控制算法。接着将第四章的内容扩展到多无人机编队跟踪，设计分布式非线性 MPC 编队跟踪控制算法。然后将第三章设计的神经网络扰动观测器与编队跟踪控制结合，提出密集编队下的多无人机编队跟踪算法，完成了本文题目对应的内容。最后进行三维仿真和实物实验验证算法效果。

第六章 多无人机下洗气流仿真环境与实物系统

本章主要介绍为验证本文算法开发的仿真环境与实物系统。本章开头介绍了机载软件系统，包括仿真、控制和规划三个 ROS 软件仓库及其相互关系。接着介绍笔者在研究生阶段开发的仿真平台，包括应用于本文的基于 Python、ROS 和 RVIZ 开发的开源多无人机仿真平台，仿真中使用的 GPU 并行计算技术，以及 ROS 通讯结构。介绍完仿真平台后，本章继续介绍无人机硬件实物系统构成，详细介绍了实验用无人机的硬件系统和配件信息。本章开发的仿真环境以第二章所建模型与辨识参数为基础，同时上述仿真环境与实物系统为第三、四、五章节的算法验证提供支撑。

6.1 无人机软件系统介绍

本文使用机器人操作系统（Robot Operating System, ROS）进行通讯，最终开发三个 ROS 包仓库，分别对应仿真、控制、规划三个环节。每个 ROS 包仓库对应一个 ROS 节点，均采用 git 进行版本控制并托管到 Github 上。这三个仓库的相互关系如下图所示。

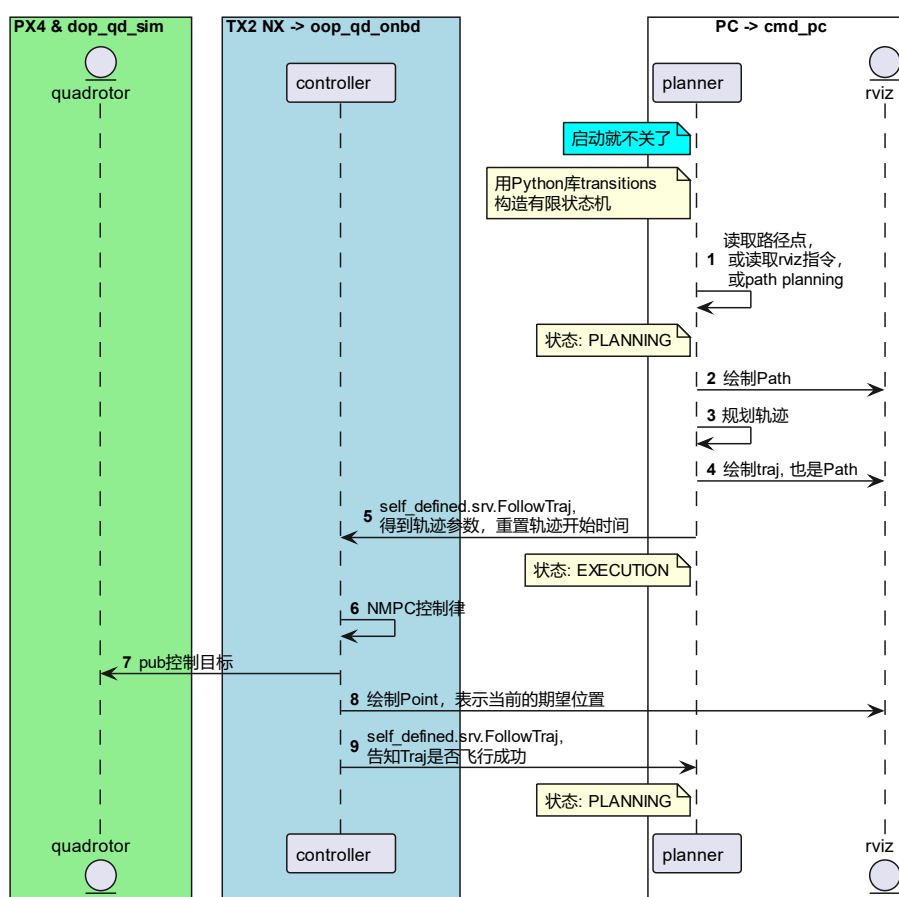


图 58 本文涉及的两个 ROS 仓库的序列图

第一个仓库为 `dop_qd_sim`，内容为使用面向数据编程编写的四旋翼仿真环境。其仿真模型采用第二章建立的模型；其算法接口采用 `mavros_msg`，与实物 PX4 飞控完全相同，具体采用角速度和归一化油门值作为输入，同时回传转速信息和飞控状态信息。这个仓库仅在仿真时运行，在实物飞行时无需启用，仓库地址为 https://github.com/Li-Jinjie/dop_qd_sim。

第二个仓库为 `oop_qd_onbd`，内容为各种控制相关的算法。这个仓库是本文算法代码的核心仓库，主要存放第三章、第四章、和第五章涉及控制和扰动估计的代码，采用面向对象的模式编写，既可以在仿真中运行，又可以不加修改便上传到飞机上运行。其输入为多项式轨迹参数，内部会通过微分平坦将轨迹转换为全状态轨迹点，进而传送给非线性 MPC 控制器作为控制目标以进行轨迹跟踪。非线性 MPC 控制器产生的角速度和油门指令会发送给底层飞控以控制无人机飞行。该仓库在仿真时运行在地面站上，在实物飞行时运行在机载计算机上，仓库地址为 https://github.com/Li-Jinjie/oop_qd_onbd。

第三个仓库为 `cmd_pc`，内容为规划算法。这个仓库的输入是人为指定的若干路径点，输出是经 Minimum Snap 算法优化后的多项式轨迹。这个仓库起辅助作用，在仿真和实物飞行时均运行在地面站上，内部采用有限状态机（Finite State Machine, FSM）编写以控制整个系统的飞行过程，仓库地址为 https://github.com/Li-Jinjie/cmd_pc。

在这三个仓库运行时，`cmd_pc` 为整个系统的总控模块，而 `oop_qd_onbd` 内部的 `controller` 被封装为 ROS Action Server 的形式以待 `cmd_pc` 调用。当 `cmd_pc` 中的 `planner` 规划完毕并将轨迹参数以 Goal 的形式发送给 `controller` 后，等待 `controller` 执行。当 `controller` 跟踪完毕当前的轨迹后，向 `planner` 发送 Success，告知其飞行成功，使得 `planner` 进入下一个规划周期，如此循环。三个模块运行时均会将信息发送到 RVIZ 以可视化。

仿真环境 `dop_qd_sim` 的实现细节将在 6.2 节进行介绍；控制算法 `oop_qd_onbd` 运行在机载计算机上，`cmd_pc` 运行在地面站上，关于这部分的硬件细节和软件运行环境将在 6.3 节进行介绍。

6.2 基于 ROS 的多无人机仿真平台

6.2.1 六自由度仿真平台系统架构

机器人操作系统 ROS 提供了定时、通讯等多种基础功能，已经成为机器人领域的

标准，因而整个仿真环境基于 ROS 开发。本文采用 ROS 生态提供的可视化界面 RVIZ 对全部算法进行可视化。程序结构如下图所示，仿真运行的主节点按照仿真频率定时执行仿真循环，同时以固定频率定时向控制算法与 RVIZ 显示节点发送消息。此外，仿真节点从控制节点接收控制指令以驱动无人机飞行。

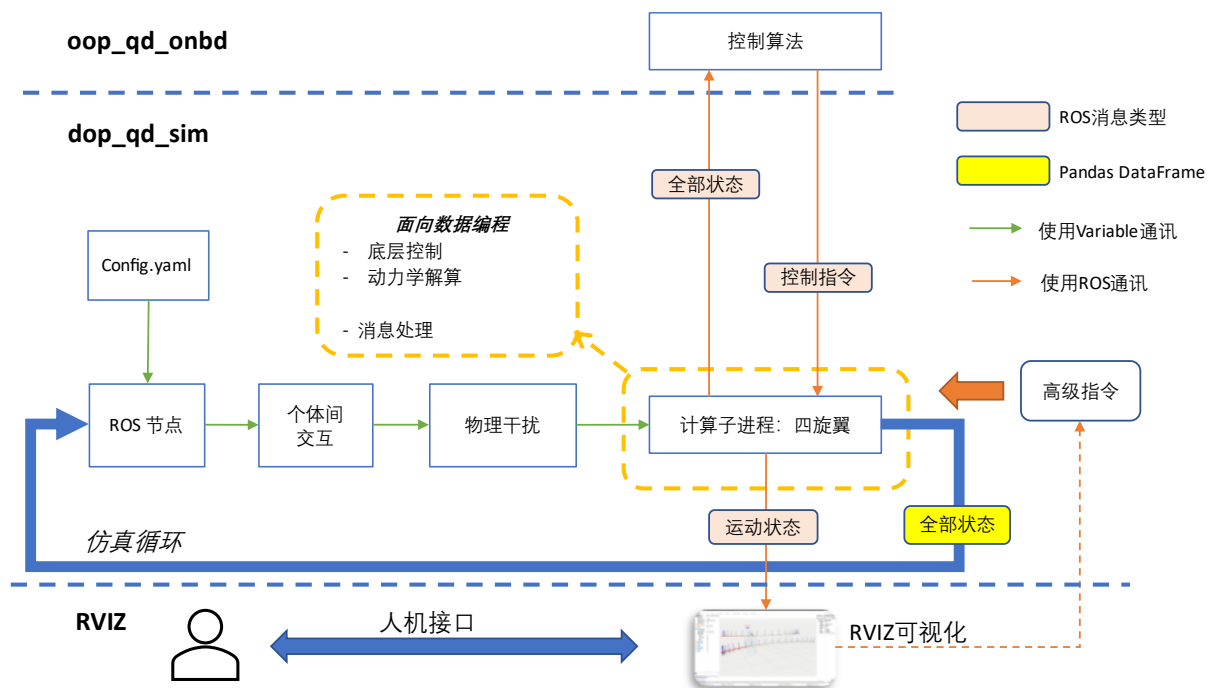


图 59 dop_qd_sim 仿真平台的程序结构

仿真采用的物理模型与第二章完全相同。仿真采用的 RVIZ 可视化界面如下图所示。

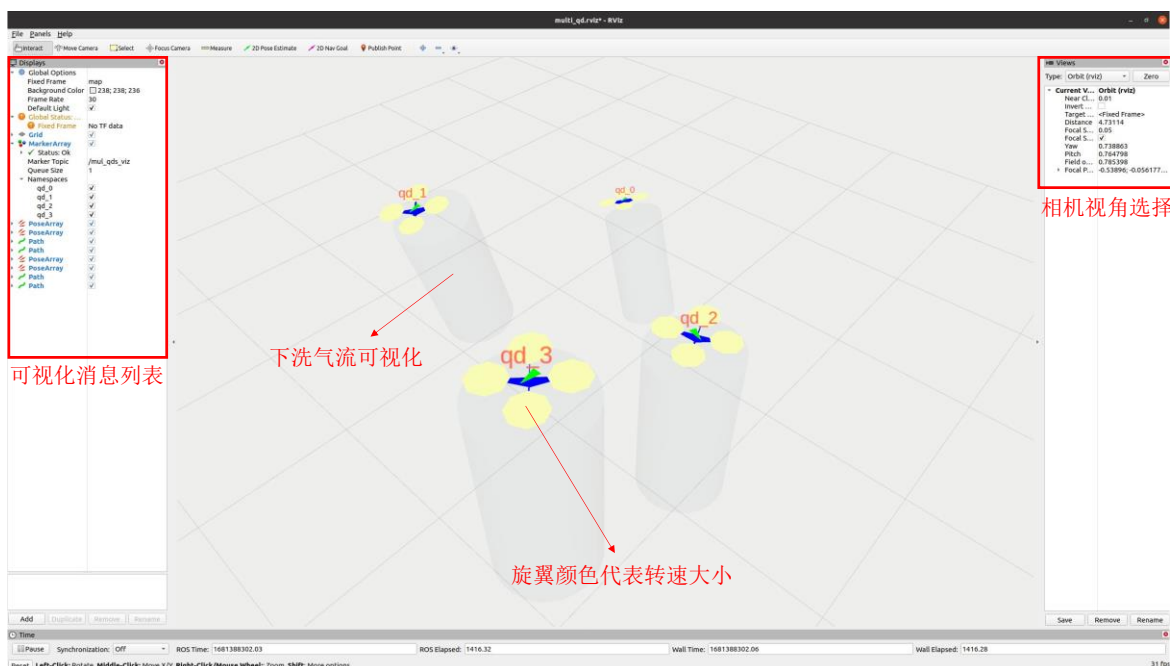


图 60 RVIZ 多无人机显示界面

仿真采用的无人机视觉模型由笔者使用一系列三角形拼接而成，同时采用 Markers 消息类型向 RVIZ 发送。除无人机机体外，还可以显示无人机的名称与下洗气流作用效果。无人机的旋翼颜色与 RPM 转速相关，采用 Matplotlib 中的“Spectral”色系，转速越快则越偏红色，转速越慢则越偏蓝色，中间转速则偏浅黄色。显示效果较为直观。

6.2.2 面向数据编程与 GPU 并行计算加速

面向多无人机的集群仿真对于集群算法的开发至关重要。集群仿真平台有两个维度的要求：第一个维度是仿真数量，第二个维度是仿真模型的逼真度，现有的平台往往只能向一个维度进行侧重。部分集群仿真平台追求大规模而牺牲逼真度，采用简单的运动学模型对系统进行大规模仿真，例如 ARGoS^[69]采用速度模型，SCRIMAGE^[70]采用自行车模型。另外一些常见的机器人仿真平台具有非常逼真的模型，如 Gazebo^[71]、AirSim^[72]、pybullet^[60]等，但在单台计算机上支持的节点数仅为 50 个左右，无法进行大规模（超过 1000 架）的集群仿真。

上述仿真平台主要采用 CPU 进行动力学解算，而近期 GPU 计算能力的发展使得研究者开始思考使用 GPU 进行大规模高精度仿真的可能性。在该方向上，NVIDIA 的 Isaac 仿真环境^[73]使用 GPU 来进行动力学仿真，实现了复杂模型的大规模并行运算。然而，Isaac 主要为深度强化学习领域服务，其开发的接口与集群仿真的需求差异较大。此外，Isaac 底层采用 CUDA 和 C++开发，对开发难度要求较高，且只能在 Linux 系统上使用。而且 Isaac 为商业闭源软件，内部难以修改，很难用于集群算法的科研用途。

本节开发了一种基于 GPU 的大规模机器人仿真平台。与传统基于面向对象（Object-Oriented Programming, OOP）的仿真平台不同，该平台采用面向数据（Data-Oriented Programming, DOP）的思路进行设计，天然支持大规模节点的并行仿真，并且可以借助 GPU 进行并行加速。与同样基于 GPU 的 Isaac 仿真平台相比，本平台基于 Python 和 PyTorch 开发，开发难度低，同时可以跨平台使用。开发完成的模型可以借用 PyTorch 中用于深度学习加速的工具 TorchScript 编译成 C++代码进一步加速。

本节并行加速的核心思想如下图所示。经典的机器人仿真环境采用面向对象的方式编写程序，一个个体是一个实例，多个个体就是多个实例，通过 for 循环或者多线程、多进程等思路进行运算。然而，由于目前家用台式机 CPU 的线程数大概在 10~20 个，当仿真节点数量达到 1000 量级时，每个 CPU 线程实际上是以循环的方式运算多个实例，

因此仿真运算速度随节点数量的增加呈现几乎线性的增长。而本节开发的仿真平台采用面向数据的方式编写程序，将同类型智能体的计算放到一起，用张量（`torch.Tensor`）的方式一批一批地并行计算，整个过程体现为数据在计算模块之间的流转。在此基础上，可以将每个计算模块写成 `nn.Module` 的形式，调用 PyTorch 的 TorchScript 将代码的内部运算编译为 C++ 以进一步提升计算速度。

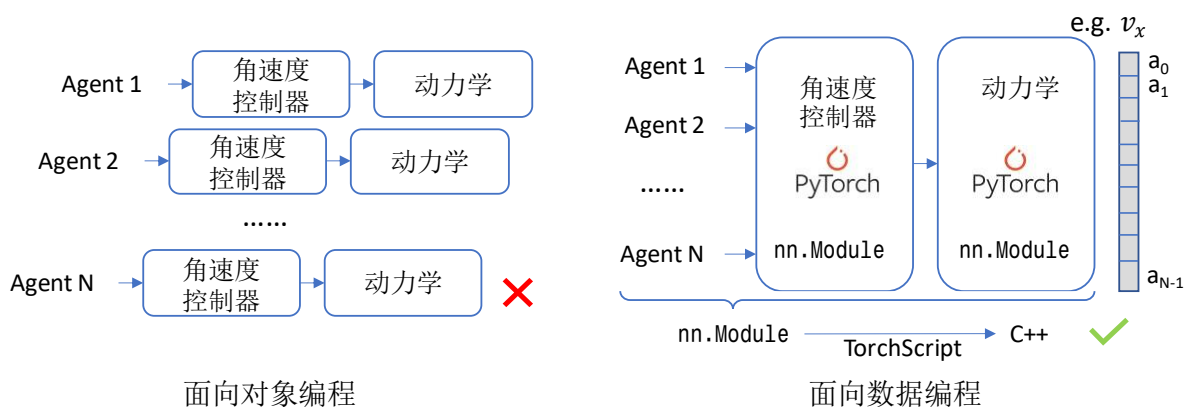


图 61 `dop_qd_sim` 并行计算的核心思想

为验证这种面向数据并行计算的优越性，本文采用第二章建立的四旋翼模型对性能进行测试。测试环境是一台个人台式计算机，CPU 为 Intel i7-10700，GPU 为 NVIDIA GTX 1660 SUPER。测试程序运行在 Ubuntu 20.04 操作系统上。所提出的方法依赖于 PyTorch 提供的 TorchScript 来加快计算速度，整个仿真循环可以用 Python 或 C++ 实现，因此本文测试了不同语言和 PyTorch 版本下每一轮的运行时间。仿真运行的四旋翼数量为 1000，先运行 500 轮使得程序稳定运行，再运行 2000 轮以计算平均耗时。本文对每个项目执行三轮测试，取平均值作为结果展示如下表。

表 14 不同语言版本和 PyTorch 版本下的仿真运行时间

语言版本	PyTorch 版本	是否出现 FALLBACK 报错	每轮的运行时间 (ms)
原生 Py3.8	1.10.0+cu102	无	0.8452
Conda Py3.8	1.10.0+cu102	无	0.8550
Conda Py3.9	1.13.0+cu116	有	1.6973
Conda Py3.9	2.0.0+cu117	无	1.4171
Conda Py3.10	1.12.0+cu116	有	1.1738

语言版本	PyTorch 版本	是否出现 FALLBACK 报错	每轮的运行时间 (ms)
Conda Py3.9	1.12.0+cu116	有	1.1857
Conda Py3.8	1.12.0+cu116	无	0.8678
C++ Release	1.12.0+cu116	有 (很多)	2.8181
C++ Debug	1.12.0+cu116	有 (很多)	2.8017

从表中可以看出, C++版本的仿真没有比 Python 版本的更快, 这可能是因为 C++版本的 TorchScript 在加载模型时报了很多 FALLBACK 警告。

然后本文选择 PyTorch 1.12.0+cu116 的 Conda Python 3.8 环境来测试运行时间随四旋翼数量的变化, 如下图所示。从图中可以看出, 当仿真数量在 5000 以下时, 计算时间几乎不随仿真数量上升而改变。即使在 10000 个仿真节点数, 计算时间也保持在 2 ms 以下, 这说明了面向数据编程且利用 GPU 加速的仿真实实现方法在大规模智能体仿真方面的优势。本文的仿真平台 dop_qd_sim 采用上述思路进行实现。

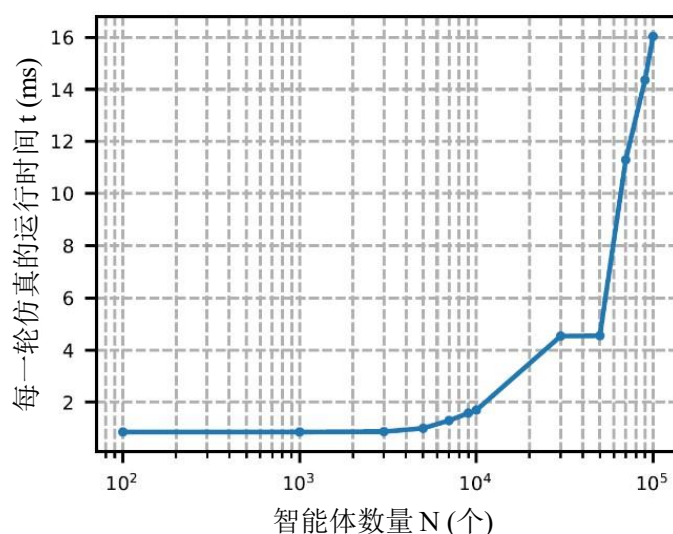


图 62 每一轮仿真的运行时间随智能体数量变化的曲线

6.2.3 基于 ROS 的通讯结构介绍

本部分介绍系统采用的 ROS 通讯架构。使用 ROS 开发程序时, 常常会陷入构建新节点和新话题的误区, 很容易创建很多节点和话题, 将系统越开发越复杂。为避免上述情况, 应该对 ROS 节点与话题的创建保持相当的克制, 并尽量采用多样的通讯方式。本文即按照该思路进行设计, 整体只区分仿真/qd_sim, 控制/nmpc_node, 和规划/planner_node 三个节点, 每个节点相关的功能都以 Python 的方式内部实现。单机轨迹跟

踪采用的 ROS 通讯架构如下图所示。/qd_sim 与/nmpc_node 间存在高频通讯，采用发布-订阅制实现，/nmpc_node 从/qd_sim 获得状态信息，同时向/qd_sim 发送控制指令。
/nmpc_node 与/planner_node 间属于任务执行模式，因而采用 Action 方式，在任务的开始由/planner_node 发送 Goal，在执行过程中由/nmpc_node 向/planner_node 不断回传 Feedback，执行结束后/nmpc_node 向/planner_node 回传一个 Success 信息。

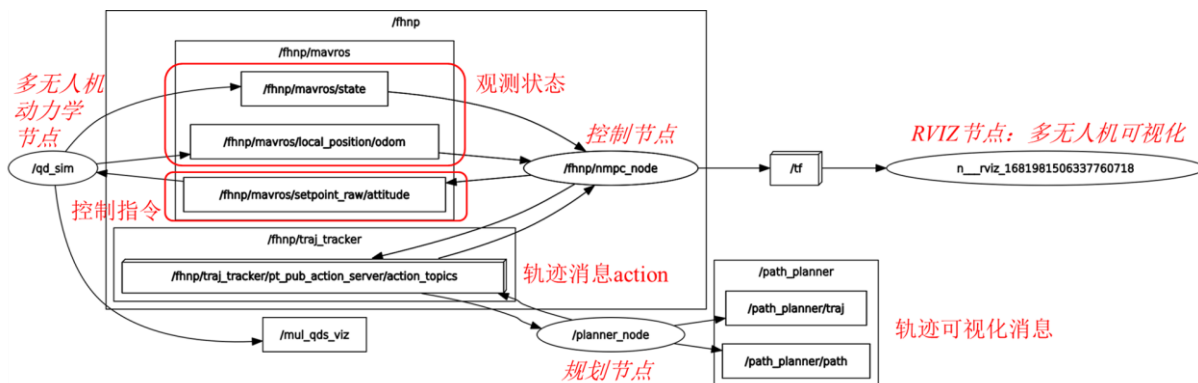


图 63 单机轨迹跟踪采用的 ROS 通讯架构

多机编队跟踪采用的 ROS 通讯架构如下图所示。仿真节点/qd_sim 采用面向数据的方式设计，因而依然保持一个节点。控制节点/nmpc_node 采用面向对象的方式设计，因而有几架飞机就初始化几个节点，每个节点包裹在各自的 namespace 内。编队参考 /formation_ref 以发布-订阅制进行消息传递。经过实践证明，上述架构设计可以优雅地实现多机仿真。

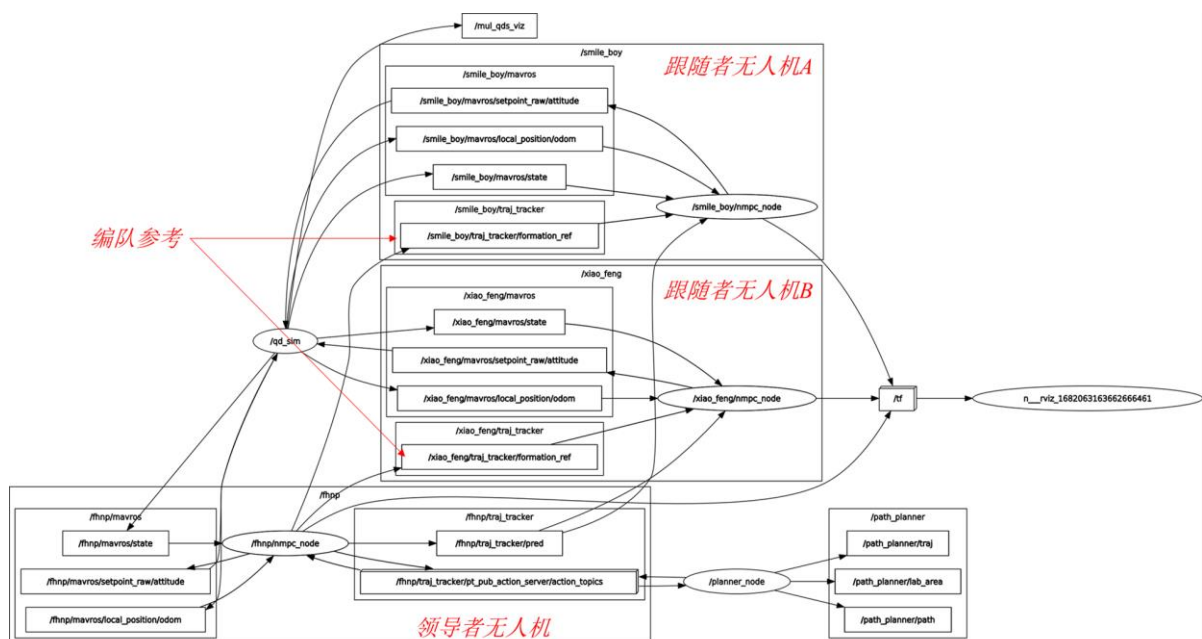


图 64 三机编队跟踪采用的 ROS 通讯架构

6.3 多无人机硬件系统设计

本文研究的无人机密集编队需要在实物上开展测试，因此需要搭建实物硬件平台，配置软件运行环境。实验室本来存在一款如下图所示的四旋翼硬件平台。然而经过测试，其续航能力只有 166 秒，其机载计算机树莓派的计算能力也较差，续航和计算能力远远无法满足本文实验的需要。



图 65 实验室之前的无人机硬件系统

为满足本研究对实物实验的需求，笔者全面升级了该四旋翼的动力系统，同时将机载计算机升级为 NVIDIA TX2 NX 以提升机载计算能力。具体思路为换用 6S 高压电池，选用更强劲的二叶桨，同时修改旋翼方向朝下以增大推力。升级后的四旋翼如下图所示。



图 66 升级之后的无人机硬件系统

经测试，其续航能力可以达到 705 秒，可以满足实验需求。经过 PX4 的振动检测，与 PX4 官网数据对比，振动特性达到“很好”水平，具备开展飞行试验的条件。组建该四旋翼所需的硬件选型请见附录。

实验飞行时，无人机飞行试验平台的外部设备如下图所示。

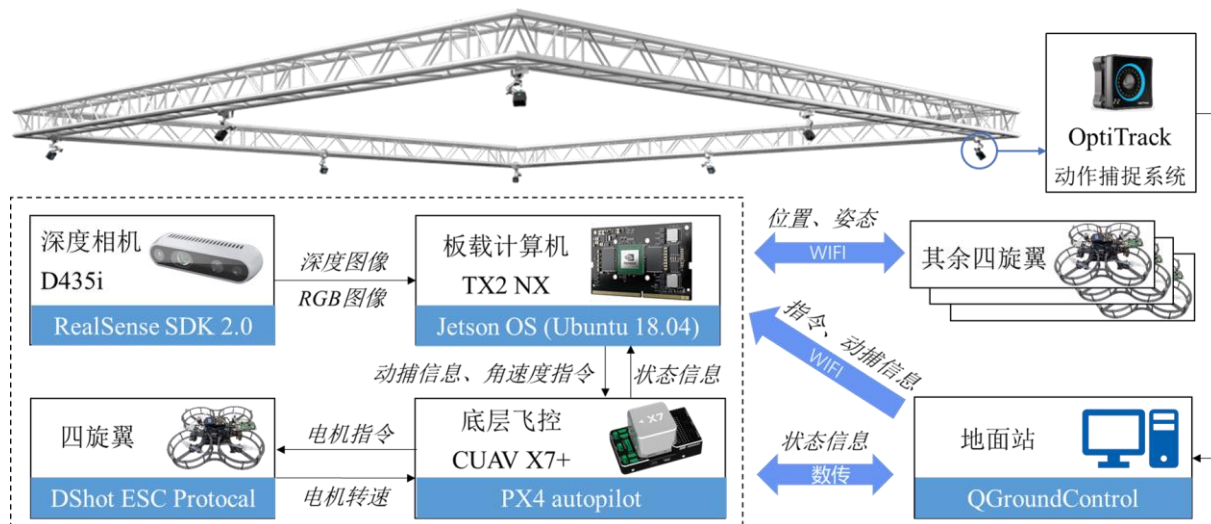


图 67 实验平台硬件框图

整个集群系统使用动作捕捉装置作为外部定位设备。在编队飞行过程中，四旋翼间通过 ROS 传输彼此的状态信息，同时通过数传向地面站电脑运行的 QGroundControl 发送状态信息以监控无人机状态。无人机实验平台的硬件信息流如下图所示。

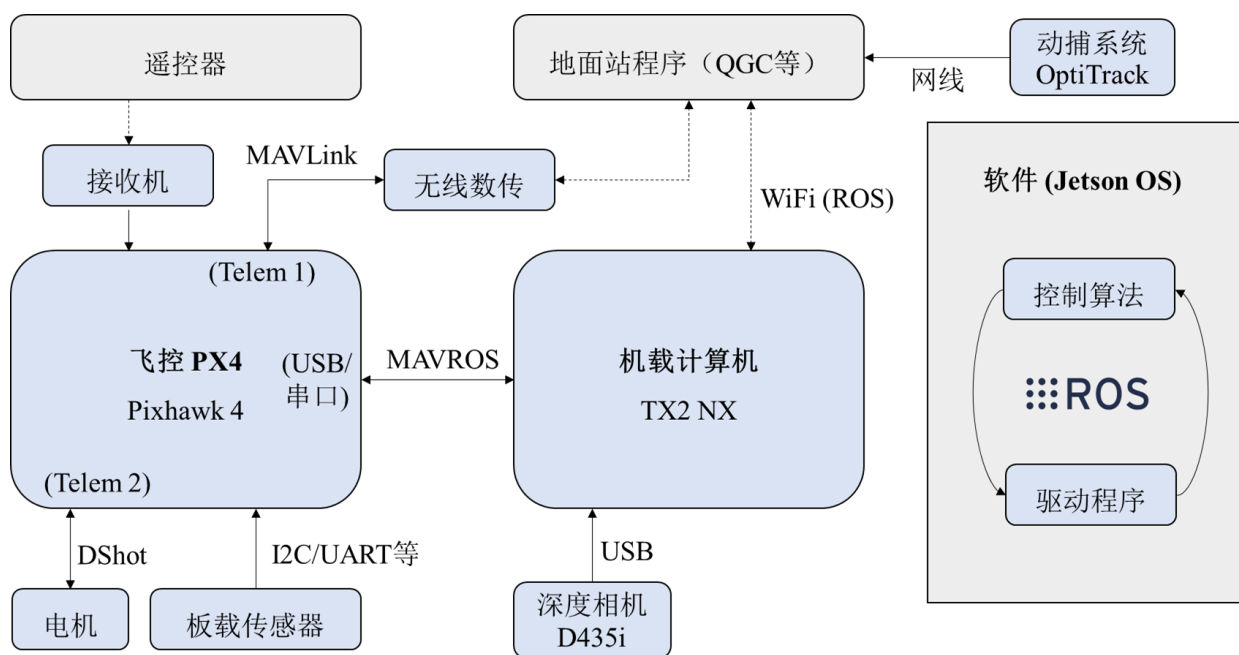


图 68 实验平台硬件连线图

机载计算设备分为运行底层控制算法与状态估计算法的 PX4 飞行控制器，和运行非线性 MPC 与深度学习算法的 TX2 NX 两部分组成，两者之间通过串口进行通讯，采用 MAVROS 消息类型作为通讯内容。动捕系统采集的状态信息经由地面站通过 ROS 转发到 TX2 NX 上以进行无人机控制。TX2 NX 上需要安装算法运行环境，具体如下表所示。

表 15 TX2 NX 软件环境配置

名称	作用
Jetson OS	NVIDIA 为 TX2 NX 开发的操作系统
NVIDIA 组件，如 TensorRT	实现计算机视觉、深度学习等算法加速
rt-kernel	Linux 系统的实时补丁，减少程序运行延迟
Docker	进行环境隔离，配置 ROS 环境
ROS 1 Noetic	进行程序间通信

至此便构建了密集编队飞行试验所需的硬件平台及软件运行环境。

6.4 本章小结

本章介绍了支撑本文算法验证的仿真环境与实物系统。首先介绍本文开发的无人机软件系统及相互关系。接着根据第二章建立的无人机模型与辨识的参数开发多无人机仿真环境，同时利用面向数据编程和 GPU 并行计算技术针对多机仿真进行优化。然后简要介绍系统采用的 ROS 通讯架构。最后介绍本文实物实验采用的硬件系统。

结 论

本文针对四旋翼集群在障碍环境中穿行存在的密集编队飞行问题，研究了四旋翼无人密集编队跟踪方法。针对密集飞行时存在的下洗气流扰动，基于谱归一化神经网络对其进行拟合与观测。针对四旋翼无人机的高精度轨迹跟踪问题，基于非线性模型预测控制设计轨迹跟踪控制律。本文创新性地将两者结合，提出气动扰动补偿下的非线性模型预测控制（MPC）轨迹跟踪控制方法，并将其拓展到编队领域实现基于非线性 MPC 的密集编队跟踪效果。对论文的研究内容总结如下：

（1）研究了使用神经网络对无人机机间下洗气流进行估计的问题。首先确定输入输出，设计神经网络观测器的结构。为提高神经网络的鲁棒性，降低其输出的不可解释性对飞行安全的危害，在网络训练中引入谱归一化方法。基于四旋翼动力学模型和机载传感器信息，控制无人机收集下洗气流扰动数据。最后对神经网络进行训练，验证其对下洗气流观测的有效性与在不同状态下输出的安全性。此外还介绍了一种简易的下洗气流模型，用于在仿真环境中模拟下洗气流效应。

（2）研究了基于非线性 MPC 方法实现无人机轨迹跟踪的问题。首先基于 Minimum Snap 算法实现轨迹生成，并通过四旋翼的微分平坦特性转化为全状态轨迹，该轨迹即为控制器的控制目标。接着设计非线性 MPC 控制器以实现无人机轨迹跟踪，包括模型预测控制采用的非线性无人机模型、优化过程和运算加速技巧。基于仿真与实物实验验证算法效果，证明所提算法可以有效处理状态和输入约束，同时可为大机动飞行提供较精确的轨迹跟踪效果。

（3）研究了下洗气流扰动作用下的无人机轨迹跟踪与多无人机编队控制问题。首先创新性地将神经网络扰动估计器与非线性 MPC 结合，引入预测性质以便于控制器提前处理可能存在的扰动现象，同时利用 MPC 的约束处理特性避免状态和输入达到上限。接着将单机轨迹跟踪拓展到分布式多机编队跟踪，提出包含下洗气流扰动观测的分布式非线性 MPC 密集编队跟踪控制算法。基于仿真与实物实验验证算法效果，证明所提算法在实现编队协同的前提下考虑了无人机间下洗气流的影响，具备较强实用价值。

本文的特点是上述算法均在实物无人机系统上进行验证且可以实时运行，具备较高的实际应用价值。然而本文仍有以下几个方面需要改进和完善：

(1) 本文在使用神经网络进行下洗气流扰动拟合时, 仅考虑了一架飞机对另一架飞机的下洗气流干扰。当集群中存在多架飞机同时干扰时, 本文假设其关系为线性叠加, 这显然较为直接。后续可以考虑采用图神经网络拟合这部分扰动。

(2) 本文采用的非线性 MPC 轨迹跟踪方案直接采用 SQP 序列二次规划对无人机非线性模型进行优化, 计算负担较大。后续可以采用部分线性化或时变线性 MPC 的方案, 在每次控制迭代前对非线性模型进行泰勒展开, 以兼顾模型精度和计算速度。

(3) 本文虽然研究的是障碍环境中的密集编队问题, 但没有考虑从感知障碍到生成轨迹的过程。后续可以考虑为领导者无人机配备同时定位与建图 (Simultaneous Localization and Mapping, SLAM) 模块, 从环境地图中规划轨迹从而带领集群躲避障碍, 实现全自主的无人机编队飞行。

(4) 本文采用的扰动抑制方案没有与其他方法进行充分比较。在控制领域, 已经有大量关于扰动抑制的研究, 包括鲁棒控制、自适应控制等等, 应该将这些方法与本文方法进行比较。此外, 也应该将本文基于 MPC 的控制方案与之前论文中基于非线性串级 PID 的方案进行比较。

参考文献

- [1] Floreano D., Wood R. J. Science, Technology and the Future of Small Autonomous Drones[J]. *Nature*, 2015, 521(7553): 460–466
- [2] Chung S. J., Paranjape A. A., Dames P., et al. A Survey on Aerial Swarm Robotics[J]. *IEEE Transactions on Robotics*, 2018, 34(4): 837–855
- [3] Siegwart R., Nourbakhsh I. R., Scaramuzza D. Introduction to Autonomous Mobile Robots[M]. Cambridge, MA, USA: MIT Press, 2011: 2
- [4] Tranzatto M., Miki T., Dharmadhikari M., et al. CERBERUS in the DARPA Subterranean Challenge[J]. *Science Robotics*, 2022, 7(66): 1–6
- [5] Tranzatto M., Mascarich F., Bernreiter L., et al. CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge[J]. *Field Robotics*, 2022, 2(2022): 274–324
- [6] Omidshafiei S., Agha-Mohammadi A. A., Amato C., et al. Decentralized Control of Multi-Robot Partially Observable Markov Decision Processes Using Belief Space Macro-Actions[J]. *The International Journal of Robotics Research*, 2017, 36(2): 231–258
- [7] Best G., Garg R., Keller J., et al. Resilient Multi-Sensor Exploration of Multifarious Environments with a Team of Aerial Robots[A]. *Proceedings of Robotics: Science and Systems (RSS)[C]*. New York, NY, USA: RSS Foundation, 2022: 1–13
- [8] Jensen-Nau K. R., Hermans T., Leang K. K. Near-Optimal Area-Coverage Path Planning of Energy-Constrained Aerial Robots With Application in Autonomous Environmental Monitoring[J]. *IEEE Transactions on Automation Science and Engineering*, 2021, 18(3): 1453–1468
- [9] Das J., Cross G., Qu C., et al. Devices, Systems, and Methods for Automated Monitoring Enabling Precision Agriculture[A]. *Proceedings of IEEE International Conference on Automation Science and Engineering (CASE)[C]*. Gothenburg, Sweden: IEEE, 2015: 462–469
- [10] Carlson S. J., Arora P., Papachristos C. A Multi-VTOL Modular Aspect Ratio Reconfigurable Aerial Robot[A]. *Proceedings of International Conference on Robotics and Automation (ICRA)[C]*. Philadelphia, PA, USA: IEEE, 2022: 8–15
- [11] Lyu X., Gu H., Wang Y., et al. Design and Implementation of a Quadrotor Tail-Sitter VTOL UAV[A]. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)[C]*. Marina Bay Sands, Singapore: IEEE, 2017: 3924–3930
- [12] Ajanic E., Feroskhan M., Mintchev S., et al. Bioinspired Wing and Tail Morphing Extends Drone Flight Capabilities[J]. *Science Robotics*, 2020, 5(47): 1–12
- [13] Zhou X., Wen X., Wang Z., et al. Swarm of Micro Flying Robots in the Wild[J]. *Science Robotics*, 2022, 7(66): 1–17
- [14] Mataric M. J. Behaviour-Based Control: Examples from Navigation, Learning, and Group Behaviour[J]. *Journal of Experimental & Theoretical Artificial Intelligence*, 1997, 9(2–3): 323–336
- [15] Ren W., Beard R. Virtual Structure Based Spacecraft Formation Control with Formation Feedback[A]. *Proceedings of AIAA Guidance, Navigation, and Control Conference and*

- Exhibit[C]. Monterey, California: AIAA, 2002: 1–8
- [16] Consolini L., Morbidi F., Prattichizzo D., et al. Leader–Follower Formation Control of Nonholonomic Mobile Robots with Input Constraints[J]. *Automatica*, 2008, 44(5): 1343–1349
- [17] Ren W. Consensus Based Formation Control Strategies for Multi-Vehicle Systems[A]. *Proceedings of American Control Conference (ACC)*[C]. Minneapolis, MN, USA: IEEE, 2006: 4237–4242
- [18] 化永朝. 高阶异构集群系统时变编队跟踪控制方法研究[D]. 中国, 北京: 北京航空航天大学, 2019
- [19] Batra S., Huang Z., Petrenko A., et al. Decentralized Control of Quadrotor Swarms with End-to-End Deep Reinforcement Learning[A]. *Proceedings of Conference on Robot Learning (CoRL)*[C]. Auckland, New Zealand: PMLR, 2022: 576–586
- [20] Dong X., Yu B., Shi Z., et al. Time-Varying Formation Control for Unmanned Aerial Vehicles: Theories and Applications[J]. *IEEE Transactions on Control Systems Technology*, 2015, 23(1): 340–348
- [21] Yu Z., Qu Y., Zhang Y. Safe Control of Trailing UAV in Close Formation Flight against Actuator Fault and Wake Vortex Effect[J]. *Aerospace Science and Technology*, 2018, 77(2018): 189–205
- [22] Preiss J. A., Hönig W., Ayanian N., et al. Downwash-Aware Trajectory Planning for Large Quadrotor Teams[A]. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*[C]. Vancouver, BC, Canada: IEEE, 2017: 250–257
- [23] Chorin A. J. Numerical Solution of the Navier-Stokes Equations[J]. *Mathematics of Computation*, 1968, 22(104): 745–762
- [24] Li S., Yang J., Chen W. H., et al. Disturbance Observer-Based Control: Methods and Applications[M]. Boca Raton, FL, USA: CRC Press, 2014: 5
- [25] Kan X., Thomas J., Teng H., et al. Analysis of Ground Effect for Small-Scale UAVs in Forward Flight[J]. *IEEE Robotics and Automation Letters*, 2019, 4(4): 3860–3867
- [26] Carter D. J., Bouchard L., Quinn D. B. Influence of the Ground, Ceiling, and Sidewall on Micro-Quadrotors[J]. *AIAA Journal*, 2021, 59(4): 1398–1405
- [27] Guo Q., Zhu Y., Tang Y., et al. CFD Simulation and Experimental Verification of the Spatial and Temporal Distributions of the Downwash Airflow of a Quad-Rotor Agricultural UAV in Hover[J]. *Computers and Electronics in Agriculture*, 2020, 172(2020): 1–11
- [28] LeCun Y., Bengio Y., Hinton G. Deep Learning Review[J]. *Nature*, 2015, 521(7553): 436–444
- [29] Goodfellow I., Bengio Y., Courville A. Deep Learning[M]. Cambridge, MA, USA: MIT Press, 2016: 1–8
- [30] Mnih V., Kavukcuoglu K., Silver D., et al. Human-Level Control through Deep Reinforcement Learning[J]. *Nature*, 2015, 518(7540): 529–533
- [31] Silver D., Schrittwieser J., Simonyan K., et al. Mastering the Game of Go without Human Knowledge[J]. *Nature*, 2017, 550(7676): 354–359
- [32] Hwangbo J., Sa I., Siegwart R., et al. Control of a Quadrotor With Reinforcement Learning[J]. *IEEE Robotics and Automation Letters*, 2017, 2(4): 2096–2103
- [33] Hwangbo J., Lee J., Dosovitskiy A., et al. Learning Agile and Dynamic Motor Skills for

- Legged Robots[J]. *Science Robotics*, 2019, 4(26): 1–13
- [34] Ibarz J., Tan J., Finn C., et al. How to Train Your Robot with Deep Reinforcement Learning: Lessons We Have Learned[J]. *The International Journal of Robotics Research*, 2021, 40(4–5): 698–721
- [35] Willard J., Jia X., Xu S., et al. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems[J]. *ACM Computing Surveys*, 2022, 55(4): 1–37
- [36] Punjani A., Abbeel P. Deep Learning Helicopter Dynamics Models[A]. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)[C]*. Seattle, WA, USA: IEEE, 2015: 3223–3230
- [37] Bansal S., Akametalu A. K., Jiang F. J., et al. Learning Quadrotor Dynamics Using Neural Network for Flight Control[A]. *Proceedings of IEEE Conference on Decision and Control (CDC)[C]*. Las Vegas, NV, USA: IEEE, 2016: 4653–4660
- [38] Kabzan J., Hewing L., Liniger A., et al. Learning-Based Model Predictive Control for Autonomous Racing[J]. *IEEE Robotics and Automation Letters*, 2019, 4(4): 3363–3370
- [39] Hewing L., Kabzan J., Zeilinger M. N. Cautious Model Predictive Control Using Gaussian Process Regression[J]. *IEEE Transactions on Control Systems Technology*, 2020, 28(6): 2736–2743
- [40] Carron A., Arcari E., Wermelinger M., et al. Data-Driven Model Predictive Control for Trajectory Tracking With a Robotic Arm[J]. *IEEE Robotics and Automation Letters*, 2019, 4(4): 3758–3765
- [41] Tagliabue A., Paris A., Kim S., et al. Touch the Wind: Simultaneous Airflow, Drag and Interaction Sensing on a Multirotor[A]. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[C]*. Las Vegas, NV, USA: IEEE, 2020: 1645–1652
- [42] Zhang W., Brunner M., Ott L., et al. Learning Dynamics for Improving Control of Overactuated Flying Systems[J]. *IEEE Robotics and Automation Letters*, 2020, 5(4): 5283–5290
- [43] Torrente G., Kaufmann E., Föhn P., et al. Data-Driven MPC for Quadrotors[J]. *IEEE Robotics and Automation Letters*, 2021, 6(2): 3769–3776
- [44] Bauersfeld L., Kaufmann E., Föhn P., et al. NeuroBEM: Hybrid Aerodynamic Quadrotor Model[A]. *Proceedings of Robotics: Science and Systems (RSS)[C]*. Held Virtually: RSS Foundation, 2021: 1–11
- [45] Wang R., Walters R., Yu R. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization[A]. *Proceedings of International Conference on Learning Representations (ICLR)[C]*. Vienna, Austria: Elsevier, 2021: 1–12
- [46] Shi G., Shi X., O’Connell M., et al. Neural Lander: Stable Drone Landing Control Using Learned Dynamics[A]. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)[C]*. Montreal, QC, Canada: IEEE, 2019: 9784–9790
- [47] Shi G., Hönig W., Yue Y., et al. Neural-Swarm: Decentralized Close-Proximity Multirotor Control Using Learned Interactions[A]. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)[C]*. Paris, France: IEEE, 2020: 3241–3247
- [48] Shi G., Hönig W., Shi X., et al. Neural-Swarm2: Planning and Control of Heterogeneous

- Multicopter Swarms Using Learned Interactions[J]. *IEEE Transactions on Robotics*, 2021, 38(2): 1063–1079
- [49] Bemporad A. Model Predictive Control Course[EB/OL]. http://cse.lab.imtlucca.it/~bemporad/mpc_course.html, 2021-04-08. (2021-04-08)
- [50] Hewing L., Wabersich K. P., Menner M., et al. Learning-Based Model Predictive Control: Toward Safe Learning in Control[J]. *Annual Review of Control, Robotics, and Autonomous Systems*, 2020, 3(1): 269–296
- [51] Lindqvist B., Mansouri S. S., Agha-mohammadi A. akbar, et al. Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles[J]. *IEEE Robotics and Automation Letters*, 2020, 5(4): 6001–6008
- [52] Hanover D., Foehn P., Sun S., et al. Performance, Precision, and Payloads: Adaptive Nonlinear MPC for Quadrotors[J]. *IEEE Robotics and Automation Letters*, 2022, 7(2): 690–697
- [53] Sun S., Romero A., Foehn P., et al. A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight[J]. *IEEE Transactions on Robotics*, 2022, 38(6): 3357–3373
- [54] Sommer H., Gilitschenski I., Bloesch M., et al. Why and How to Avoid the Flipped Quaternion Multiplication[J]. *Aerospace*, 2018, 5(3): 1–15
- [55] 全权. 多旋翼飞行器设计与控制[M]. 杜光勋, 赵峙尧, 戴训华, 等, 译. 中国, 北京: 电子工业出版社, 2018: 121–124
- [56] Faessler M., Franchi A., Scaramuzza D. Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories[J]. *IEEE Robotics and Automation Letters*, 2018, 3(2): 620–626
- [57] Beard R. W., McLain T. W. *Small Unmanned Aircraft: Theory and Practice*[M]. Princeton, NJ, USA: Princeton University Press, 2012: 114-117,144-145
- [58] Förster J. System Identification of the Crazyflie 2.0 Nano Quadrocopter[D]. Zürich, Switzerland: ETH Zürich, 2015
- [59] Jardin M. R., Mueller E. R. Optimized Measurements of Unmanned-Air-Vehicle Mass Moment of Inertia with a Bifilar Pendulum[J]. *Journal of Aircraft*, 2009, 46(3): 763–775
- [60] Panerati J., Zheng H., Zhou S., et al. Learning to Fly—a Gym Environment with PyBullet Physics for Reinforcement Learning of Multi-Agent Quadcopter Control[A]. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*[C]. Prague, Czech Republic: IEEE, 2021: 7512–7519
- [61] 张阿斯顿, 立顿扎卡里 C., 李沐, 等. 动手学深度学习[M]. 何孝霆, 胡瑞潮儿, 译. 中国, 北京: 人民邮电出版社, 2023: 86–94
- [62] Miyato T., Kataoka T., Koyama M., et al. Spectral Normalization for Generative Adversarial Networks[A]. *Proceedings of International Conference on Learning Representations (ICLR)*[C]. Vancouver, BC, Canada: ICLR, 2018: 1–14
- [63] Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization[A]. *Proceedings of International Conference for Learning Representations (ICLR)*[C]. San Diego, CA, USA: ICLR, 2015: 1–11
- [64] Gasparetto A., Boscariol P., Lanzutti A., et al. Trajectory Planning in Robotics[J]. *Mathematics in Computer Science*, 2012, 6(3): 269–279

-
- [65] Mellinger D., Kumar V. Minimum Snap Trajectory Generation and Control for Quadrotors[A]. Proceedings of IEEE International Conference on Robotics and Automation (ICRA)[C]. Shanghai, China: IEEE, 2011: 2520–2525
- [66] Brescianini D., D’Andrea R. Tilt-Prioritized Quadrocopter Attitude Control[J]. IEEE Transactions on Control Systems Technology, 2020, 28(2): 376–387
- [67] Verschueren R., Frison G., Kouzoupis D., et al. Acados—a Modular Open-Source Framework for Fast Embedded Optimal Control[J]. Mathematical Programming Computation, 2022, 14(1): 147–183
- [68] Grob M. Quaternion Based Estimation and Control for Attitude Tracking of a Quadcopter Using IMU Sensors[D]. Zürich, Switzerland: ETH Zürich, 2016
- [69] Pinciroli C., Trianni V., O’Grady R., et al. ARGoS: A Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems[J]. Swarm intelligence, 2012, 6(2012): 271–295
- [70] DeMarco K., Squires E., Day M., et al. Simulating Collaborative Robots in a Massive Multi-Agent Game Environment (SCRIMMAGE)[A]. Proceedings of Distributed Autonomous Robotic Systems: The 14th International Symposium[C]. Boulder, CO, USA: Springer International Publishing, 2019: 283–297
- [71] Koenig N., Howard A. Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator[A]. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[C]. Sendai, Japan: IEEE, 2004: 2149–2154
- [72] Shah S., Dey D., Lovett C., et al. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles[A]. Proceedings of Field and Service Robotics[C]. Zurich, Switzerland: Springer International Publishing, 2017: 621–635
- [73] Makoviychuk V., Wawrzyniak L., Guo Y., et al. Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning[A]. Proceedings of Conference on Neural Information Processing Systems (NeurIPS)[C]. Held Virtually: NeurIPS Foundation, 2021: 1–21

附 录

表 16 旋翼转速随油门变化的原始数据

油门指令 %	DShot 测量的转速/RPM (从低往高)	DShot 测量的转速/RPM (从高往低)
0	2585	2578
4	3921	3918
8	5150	5171
12	6193	6159
16	7421	7197
20	8621	8484
24	9757	9578
28	11121	10612
32	11850	11764
36	12850	12733
40	13850	13524
44	14400	14150
48	15536	15273
52	16571	16221
56	17428	17014
60	18557	18121
64	19132	19021
68	19550	19273
72	20328	20197
76	20785	20771
80	21588	21092
84	22047	22228
88	22819	23264
92	23431	23657

油门指令 %	DShot 测量的转速/RPM (从低往高)	DShot 测量的转速/RPM (从高往低)
96	23878	24242
100	24000	24656

表 17 无人机的扭转摆动周期

	实验序号	起始时间	终止时间	50T 周期 (s)
✕轴周期	1	0:07.078	1:08.879	61.801
	2	0:02.786	1:04.616	61.83
	3	0:04.171	1:05.955	61.784
∠轴周期	1	0:0.232	1:14.002	73.77
	2	0:02.567	1:16.390	73.823
	3	0:01.603	1:15.372	73.769
∩轴周期	1	0:05.319	1:41.655	96.336
	2	0:02.674	1:39.093	96.419
	3	0:04.540	1:40.845	96.305

表 18 无人机硬件选型

名称	价格¥	名称	价格¥
深度相机 D435i	2580	TX2 NX 核心板	1299
WeAct NX 载板+M.2 转板	821	NVME M.2 SSD 固态硬盘-东芝 BG4 128G	189
Ubuntu 无线网卡 1300M	65.8	格式电池	488
各种线材	20	6S 4000mah 30C	
T-Motor 电调 F55A Pro II	579	各种规格铜螺柱	10
F3		T-Motor F90 电机	180×4
Pixhawk 6X 飞控+电源模块	2228	KV1500	
+M8N GPS		T-Motor 6143 三叶桨	28
数据线	12.90	1 套 4 个	
		USB 拓展坞	28.90

名称	价格¥	名称	价格¥
飞控减震托盘	15	减震圈	1.5
WFLY RF209S 接收机	128	碳板机架	785
总价	9964		

攻读硕士学位期间取得的学术成果

论文

- [1] **Li J.**, Han L., Ren Z. Indoor Localization for Quadrotors using Invisible Projected Tags[A]. IEEE International Conference on Robotics and Automation (**ICRA**)[C]. Philadelphia, PA, USA: IEEE, 2022: 9404-9410. (**EI, CCF-B**, DOI: 10.1109/ICRA46639.2022.9812449)
- [2] Yan Z., Han L., Li X., **Li J.**, Ren Z. Event-Triggered Optimal Formation Tracking Control using Reinforcement Learning for Large-Scale UAV Systems[A]. IEEE International Conference on Robotics and Automation (**ICRA**)[C]. London, UK: IEEE, 2023: to be published. (**EI, CCF-B**, <https://arxiv.org/abs/2301.06749>)
- [3] **Li J.**, Han L., Yu H., et al. Potato: A Data-Oriented Programming 3D Simulator for Large-Scale Heterogeneous Swarm Robotics. **Accepted by ICRA2023 Workshop** on The Role of Robotics Simulators for Unmanned Aerial Vehicles.
- [4] **Li J.**, Han L., Yu H., et al. Nonlinear MPC for Quadrotors in Close-Proximity Flight with Neural Network Downwash Prediction[A]. **Submitted to** the 62nd IEEE Conference on Decision and Control (**CDC**).

专利

- [5] 基于不可见投影二维码的室内定位方法（**发明专利**，老师一作学生二作，已受理，**202111154577.4**）

软件著作权

- [6] 基于不可见投影二维码的定位程序软件 V1.0（**计算机软件著作权**，老师一作学生二作，已授权，**2022SR0123403**）
- [7] 大规模集群仿真平台 V1.0（**计算机软件著作权**，老师一作学生二作，已授权，**2021SR1039534**）

致 谢

硕士三年转瞬即逝，又来到了说再见的时候。我在北航的硕士生涯伴随着新冠疫情开始，又在疫情放开时结束，非常幸运能够在自我成长中度过这曲折的三年。一路走来，我想感谢这一段旅程中遇到的很多人。

感谢我的导师，任章老师。任老师总是能在关键节点给我建议，在重大选择上坚定我的信心。记得刚开学的时候，我问任老师怎么能读好研究生，任老师送了我两个词：认真、专注，这奠定了我读研的主基调。在读研中期面临未来选择时，由于国际形势动荡，我对一直以来的出国读博愿望产生了动摇，于是去询问任老师意见。任老师和我说：“出国读博对你的未来影响很大，甚至可能会影响你一生的成就”，这让我坚定地选择了出国读书的道路。任老师学识渊博，人生阅历丰富，常常和我畅谈种种认识与体会，使我收获良多。衷心感谢任老师！

感谢我的科研领路人，韩亮老师。韩老师是我研究生阶段交流最频繁的老师，直接指导了我的全部科研成果。在科研选题上，我常常会产生游离实验室主线之外的科研想法，不管这些想法是否合理，韩老师都非常尊重我的想法，并愿意与我认真探讨它们的可行性。在科研过程中，我所选择的机器人领域非常看重硬件投入，感谢韩老师的支持，让我在研究生期间从未感受到实验设备的欠缺，还让我去杭州完成了硕士论文所需的实验。在论文撰写上，韩老师总是在 PDF 上标满黄色批注，仔细指出论文中的问题，让论文质量快速提升。科研之外，韩老师还非常关心我的发展，我也从和韩老师的相处中学到很多待人接物的道理。衷心感谢韩老师！

感谢实验室的李清东老师、董希旺老师、李晓多师兄、化永朝师兄和于江龙师兄。李老师富有生活趣味，愿意与学生交流，我读研苦闷时常常喜欢和李老师聊天。董老师做事勤奋刻苦，教会我许多值得回味的道理。李晓多师兄、化永朝师兄和于江龙师兄我虽然接触不多，但都曾在学习和科研上给我以建议。此外还要感谢认真教学的张海老师（状态估计）与王磊老师（矩阵论）。特别感谢张海老师，亲自为我撰写推荐信。衷心感谢在北航遇到的老师们！

感谢我在新主楼 D644、二号楼 218、杭研院 11 幢 512、新主楼 D106 和大运村 1804 遇到的小伙伴们。感谢在 D644 遇到的沈志鹏、姜洋、李清泽、胡思博、白文露、周诗钰、石明慧、李旭恒、康杨睿、张钰清、孙瑜、谭劲昌，和大家在一起读书野餐滑雪看

星星，度过了研究生最快乐的一段时光。感谢在 218 遇到的王建华、廖睿雯、谢雨欣、严紫薇、黄靖宜、于浩洋、郝鹏坤、王越、曹轩、林雨恒、张毅博，你们有着中法工程师学院独特的浪漫与热情，让我明白世界因多样而精彩。感谢在杭研院遇到的李卓然、王兆天、查霁容，一起度过了一段简单又专注的博士申请时光，同时教会我好多句法语。感谢在 D106 遇到的陈奕丞、吕晓康、李庆、郝泽、朱邦钊，在大家的陪伴下完成了硕士论文的写作。感谢我的室友们，1804 的雷洪蔼、张一帆、张金雷，非常怀念和大伙枕边夜话的日子。此外，还要感谢张政、石鹏飞、王建华、王善澎师兄和廖睿雯、谢雨欣、苏飘逸师姐，是你们指引我快速适应研究生生活。还要感谢同级的黄继开、许智印、张扬、赵锋、陈宏旭、曹慧、马文宗，我们共同进步、共同成长。还要感谢黄昂、舒沛玄、石哲鑫、陈旭文、龙彦辰、王嘉璇、火振权、万超凡，真理越辩越明，在和学弟学妹们的讨论中我也提升了很多。衷心感谢在北航遇到的同学们！

感谢我的家人们一如既往的支持与鼓励。读研路上时时有不顺，不管遇到什么事情，每周末与父母聊聊天就会开心起来。感谢路之纤同学的陪伴，和你在一起我时常感觉到灵魂的共鸣，希望余生也可以与你一同度过。

感谢北航自动化学院和北航杭研院为每一位在读学生提供的帮助。感谢人类科技史和开源社区中的无数前辈和同仁，是你们让我接过求知的火炬，走向名为未知的远方。

尼采说，每一个不曾起舞的日子，都是对生命的辜负。感谢努力向前的自己。

作者简介

李谨杰，男，汉族。

1997年08月08日出生于山西省太原市。

2016年09月考入北京航空航天大学高等理工学院（沈元荣誉学院），2020年06月本科毕业并获得自动化专业工学学士学位。

2020年09月免试进入北京航空航天大学自动化科学与电气工程学院，攻读控制科学与工程专业学术型硕士学位至今。

研究方向为自主无人机（空中机器人）的感知、决策、规划、控制问题。担任 The 62nd IEEE Conference on Decision and Control (CDC 2023) 审稿人。

个人主页：www.jinjie.li。