

# 过驱动六旋翼无人机的自适应模型预测控制

刘越千

学 院：机电工程与自动化学院 专 业：自动化

学 号：180320123 指导教师：陈浩耀教授

2022 年 6 月

# 哈尔滨工业大学深圳校区

# 毕业设计（论文）

题        目 过驱动六旋翼无人机的  
自适应模型预测控制

姓        名 刘越千

学        号 180320123

学        院 机电工程与自动化学院

专        业 自动化

指 导 教 师 陈浩耀教授

答 辩 日 期 2022年06月09日

## 摘要

在需要与环境有物理交互的场景下，过驱动无人机由于可以在机体坐标系下产生六自由度的合力和合力矩，故在完成某些如带臂精确操作物体、慢速穿越狭窄缝隙等复杂任务时，比欠驱动无人机有优势。上述复杂任务都对无人机的飞行性能有较高的要求，而轨迹跟踪精度是衡量其飞行性能的一个重要参考。模型预测控制是解决轨迹追踪问题的优秀方案，但是其依赖于精确的系统模型，在机械结构较为复杂的倾转机臂型过驱动六旋翼无人机上是难以获得的。本文针对此问题提出了自适应模型预测控制方法，期望在模型预测控制器仅考虑简化的系统模型、存在较大模型误差以及时变的外部干扰的情况下，能够控制过驱动六旋翼无人机获得理想的轨迹追踪精度。

本文提出的“自适应模型预测控制”的主要思想是：在模型预测控制器与实际被控对象之间加入 $\mathcal{L}_1$ 自适应控制器。 $\mathcal{L}_1$ 自适应控制器的作用是使“自身与被控对象串联的整体”向模型预测控制器中参考的名义模型收敛，如此对于模型预测控制器来说，被控对象的动态特性与名义模型保持一致，排除了模型误差和外部干扰对模型预测控制器的负面影响。这种思路曾见于四旋翼无人机上，但并没有工作将其用于本课题所研究的对象上，故本文从运动学、动力学系统建模和模型预测控制理论、 $\mathcal{L}_1$ 自适应控制基础理论出发，独立且完整地推导了适用于此种无人机的自适应模型预测控制算法。

本文还详细叙述了以下辅助工作：无人机的硬件设计和装配、PID 备份控制器和控制分配算法设计与实现、仿真环境的搭建等，这些工作是课题研究的重要组成部分。实验验证方面，本文设计了多组针对“自适应”特性的对照试验：在仿真中人为引入较大的模型误差和干扰，以测试算法性能；在实物实验中不进行模型参数的辨识和调整，而直接测试算法是否能够跟踪参考轨迹。本文还对数据进行了分析与解释、展示了算法的优劣所在，并成功验证了算法的有效性。

**关键词：**模型预测控制； $\mathcal{L}_1$ 自适应控制；过驱动无人机；控制分配；

## Abstract

In the scenario where micro aerial vehicles (MAVs) need to interact with the world physically, over-actuated MAVs (OMAVs) are more capable of completing tasks such as flying with a robotic arm, manipulating items precisely, and crossing narrow gaps at low speeds, thanks to its ability to generate 6DOF wrenches in the body frame.

Most physical-interaction-involved tasks demand high MAV performance, and trajectory tracking performance is one crucial benchmark in assessing how well a MAV can fly. In literature, model predictive control (MPC) is an excellent solution to the trajectory tracking problem. MPC depends on an accurate reference model to achieve high tracking performance. An ACCURATE reference model, however, is hard or even impossible to obtain, especially in the OMAV case. This work proposes an adaptive model predictive control method to achieve desirable tracking performance on a tilt-rotor OMAV with unmodeled dynamics, modeling error, and time-variant external disturbance.

The idea of the proposed adaptive model predictive controller is to insert an  $\mathcal{L}_1$  adaptive controller between the MPC and the plant. The  $\mathcal{L}_1$  adaptive controller makes the combination of itself and the plant similar to the reference model. Then from the MPC perspective, modeling error no longer presents an issue. This method has been seen on quadcopters but not on the OMAV studied in this work. So, this work derives the appropriate algorithms for the OMAV from the basics of system modeling, MPC theory, and  $\mathcal{L}_1$  adaptive control theories.

There are other topics to consider before testing the proposed method on a physical OMAV system: designing and assembling the hardware, designing a backup controller, implementing the control allocation algorithm, and building the simulation environment. The investigation of the topics is an indispensable part of the thesis project. Targeted experiments are then designed to verify the effectiveness of the proposed method: in simulated experiments, we introduce significant modeling error and external disturbance; and in real-world experiments, we deploy our controller directly onto the OMAV without identifying or tuning model parameters and see if the OMAV can track a designed trajectory. This work gives interpretation and statistical analysis of the results, shows the advantages and disadvantages of the adaptive MPC algorithm, and, most importantly, verifies that it works both in the simulation environment and in real-world systems.

**Keywords:** Model Predictive Control,  $\mathcal{L}_1$  Adaptive Control, OMAV, Control Allocation

## 目 录

|                           |    |
|---------------------------|----|
| 摘要                        | I  |
| ABSTRACT                  | II |
| 第1章 绪论                    | 1  |
| 1.1 课题背景及研究的目的和意义         | 1  |
| 1.1.1 课题背景和来源             | 1  |
| 1.1.2 研究目的和意义             | 1  |
| 1.2 相关研究工作和成果             | 2  |
| 1.2.1 过驱动无人机              | 2  |
| 1.2.2 欠驱动无人机的模型预测控制       | 3  |
| 1.2.3 研究现状小结              | 4  |
| 1.3 研究内容和方法               | 4  |
| 第2章 无人机硬件和系统建模            | 5  |
| 2.1 引言                    | 5  |
| 2.2 无人机硬件                 | 5  |
| 2.2.1 飞行平台概述              | 5  |
| 2.2.2 机臂倾转结构              | 6  |
| 2.2.3 电子和计算设备             | 7  |
| 2.3 系统建模                  | 8  |
| 2.3.1 坐标系设置               | 8  |
| 2.3.2 符号记号及说明             | 9  |
| 2.3.3 系统运动学和动力学           | 10 |
| 2.4 本章小结                  | 11 |
| 第3章 自适应模型预测控制             | 12 |
| 3.1 引言                    | 12 |
| 3.2 基于名义模型的模型预测控制         | 12 |
| 3.2.1 最优控制问题              | 12 |
| 3.2.2 模型预测控制              | 14 |
| 3.2.3 潜在问题                | 16 |
| 3.3 $\mathcal{L}1$ 自适应控制器 | 16 |

|  |           |
|--|-----------|
| 3.3.1 主要思想和意义 .....                      | 16        |
| 3.3.2 自适应算法 .....                        | 17        |
| 3.3.3 实时运行优化 .....                       | 19        |
| 3.4 本章小结 .....                           | 20        |
| <b>第 4 章 带前馈的 PID 控制 .....</b>           | <b>21</b> |
| 4.1 引言 .....                             | 21        |
| 4.2 控制器设计 .....                          | 22        |
| 4.2.1 位置控制 .....                         | 22        |
| 4.2.2 姿态控制 .....                         | 22        |
| 4.2.3 合力与合力矩的计算 .....                    | 25        |
| 4.3 控制分配 .....                           | 25        |
| 4.3.1 执行器建模 .....                        | 26        |
| 4.3.2 效率最优控制分配 .....                     | 28        |
| 4.3.3 奇异点姿态处理 .....                      | 29        |
| 4.4 本章小结 .....                           | 30        |
| <b>第 5 章 实验与分析 .....</b>                 | <b>31</b> |
| 5.1 实验设置 .....                           | 31        |
| 5.1.1 OmniHex 系统架构 .....                 | 31        |
| 5.1.2 参考轨迹的生成 .....                      | 33        |
| 5.2 数值仿真实验 .....                         | 36        |
| 5.3 GAZEBO 仿真实验 .....                    | 43        |
| 5.4 实物实验 .....                           | 48        |
| 5.4.1 遥控飞行 .....                         | 49        |
| 5.4.2 Offboard 模式轨迹追踪 .....              | 50        |
| 5.4.3 $\mathcal{L}1$ -MPC 的基本飞行实验 .....  | 52        |
| 5.5 本章小结 .....                           | 56        |
| <b>结 论 .....</b>                         | <b>57</b> |
| <b>参考文献 .....</b>                        | <b>59</b> |
| <b>哈尔滨工业大学深圳校区本科生毕业设计（论文）原创性声明 .....</b> | <b>62</b> |
| <b>致 谢 .....</b>                         | <b>63</b> |

## 第1章 绪论

### 1.1 课题背景及研究的目的和意义

#### 1.1.1 课题背景和来源

多旋翼无人机敏捷灵活，是移动机器人中的常见研究对象。随着感知、规划和控制相关技术的发展，无人机与环境的交互正成为研究热点之一。在无人机与环境进行交互的场景下，相比于欠驱动无人机，过驱动无人机由于可以在机体坐标系下产生完整的六自由度的合力和合力矩（wrench），无姿态和位置上的耦合，有较大的潜力完成更复杂的工作。

在过驱动无人机中，一种较为综合的机型结构设计是：在欠驱动六旋翼无人机的基础上，赋予每个旋翼矢量推力的能力。这样的结构设计既考虑了飞行效率，也考虑到了力的全向完整性。尽管如此，与欠驱动型相比，它的机械机构更复杂、重量更大、系统建模和算法设计需要考虑的方面也更多。

无人机的轨迹跟踪精度是衡量基本飞行性能的重要指标。模型预测控制在轨迹跟踪上的性能普遍优于 PID 控制，它基于优化的框架使设计者能更统一地综合考虑系统模型、约束条件以及上层的优化目标。模型预测控制的性能依赖模型的精度，但是完美的模型几乎是不可能获得的，再考虑到飞行平台有携带质量较大且非恒定的负载的需求，故需要在控制器设计中引入自适应的特性。

#### 1.1.2 研究目的和意义

课题意义在于搭建高性能的飞行平台以及面向未来的、易用的开发框架，为后续的感知、规划、机械臂整合、视觉伺服以及自动智能飞行的研究打下基础。本文的研究成果能用于无人机带臂飞行、精确穿越狭缝、空中抓取和操控等场景中，提升无人机的飞行性能，减轻上游模块算法的负担和复杂度。

目前国内外并没有在此种过驱动无人机上使用自适应模型预测控制的工作和成果，故本课题希望填补这项空白。

## 1.2 相关研究工作和成果

### 1.2.1 过驱动无人机

苏黎世联邦理工学院的 Autonomous Systems Lab (ASL) 是最早开始研究倾转旋翼型过驱动无人机的团队之一<sup>[1]</sup>。文献[2]提出了这种无人机的动力学模型和基于伪逆的控制分配算法，并使用 PID 配合前馈控制实现了姿态和位置控制的解耦，同时提出了解决控制分配问题中奇异位姿和线缆缠绕的方法。文献[3]的主要工作是给出了导数控制分配矩阵，实现了机臂线缆的反缠绕，并设计了基于导数分配的 LQR 和 PID 两种控制器，得到了比文献[2]提出的控制器更好的控制效果。



图 1-1 ASL 开发的倾转旋翼型过驱动无人机<sup>[2]</sup>

文献[4]首次在图 1-1 所示的无人机上实现了模型预测控制（Model Predictive Control, MPC），使用了 EKF（Extended Kalman Filter）对合力和力矩的误差、外力扰动进行观测，但是最终控制器性能和 PD 控制器相当，并没有显著提升。在文献[5]和文献[6]中，作者使用了基于高斯过程的离线辨识来捕捉未建模动态，使姿态追踪性能明显提高。ASL 接着研究了多飞行模式下对不同目标进行追踪的 MPC<sup>[7]</sup>：提出了自由飞行、末端不动、追踪末端的运动和力，这三种模式下的控制方案和切换逻辑。文献[4]和[7]并没有使用文献[3]所推导的导数分配算法，而是沿用了文献[2]中的基本分配算法；MPC 的输出为合力和力矩，依靠分配矩阵映射到动力电机的转速，以及驱动机臂倾转的矢量舵机的转角。

文献[8]在倾转型四旋翼无人机上使用了自抗扰控制理论（Active Disturbance Rejection Control, ADRC）的扩张状态观测器（Extended State Observer, ESO），将未建模动态和建模误差以及外界扰动合为一项总扰动进行估计，并在 PD 控制器的基础上做补偿。仿真中追踪轨迹数据很理想，但是缺少实物实验，实物效果有

待验证。

文献[9]将多种过驱动机型抽象为一类对象，提出在 MPC 框架下同时解决局部路径规划和跟踪的方法，控制器直接输出动力电机的角加速度。文中提到的“在 MPC 框架下同时解决控制和其他上层问题”的思想与文献[10]和文献[11]的思想类似。这些文献反映了基于优化的控制架构的灵活性。

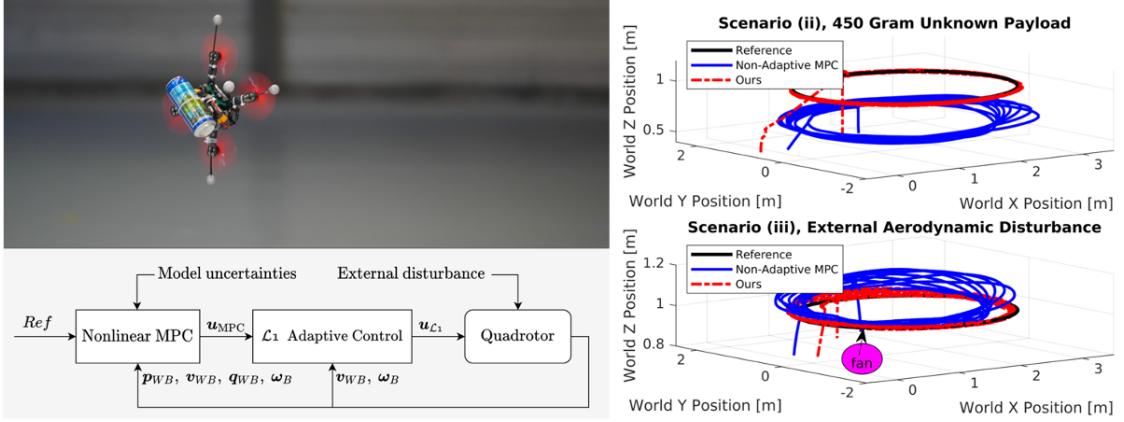
### 1.2.2 欠驱动无人机的模型预测控制

文献[12]将 MPC 与在线系统辨识相结合，提出了基于学习的 MPC (Learning Based MPC, LBMPC)，LBMPC 可根据传感器数据更新模型参数，捕捉未建模动态以及模型误差；相比于不进行模型参数更新的 MPC，LBMPC 的性能更好。

苏黎世大学 Robotics and Perception Group (RPG) 在竞速四旋翼无人机上使用了数据驱动的 MPC：文献[13]从数据中估计出模型误差以提高追踪精度。但[13]更接近[5]、[6]的组合，并没有在线更新模型参数，而是先得到名义模型，随后在实际飞行中采集状态数据，然后通过学习得到一个高斯过程用于补偿模型误差；这种方法比名义模型的 MPC 效果好不少，特别是在空气动力学效果显著的高速情况下。

几乎所有依赖于数据采集和离线辨识的方法都很难在有额外负载和外界干扰比如大风，电池电压下降等情况下保持良好的性能，毕竟它们只是一定程度上解决了模型误差的问题。基于这个痛点，RPG 提出了[14]，将 MPC 和  $\mathcal{L}_1$  自适应控制器串联 ( $\mathcal{L}_1$ -MPC)，在无需事先收集数据进行学习的情况下实现了令人满意的追踪效果以及抗干扰能力。 $\mathcal{L}_1$ -MPC 计算更新一次需要的时间明显比[13]短；作者还提到了 INDI-MPC<sup>[15]</sup>，并将其与  $\mathcal{L}_1$ -MPC 进行对比，得出的结论是：在模型精确且没有附加干扰的情况下 INDI-MPC 略有优势，但是若要考虑附加干扰，那么  $\mathcal{L}_1$ -MPC 性能显著好于 INDI-MPC。文献[14]结尾处提到： $\mathcal{L}_1$ -MPC 的劣势在于 MPC 与  $\mathcal{L}_1$  控制器的串联结构可能会破坏约束条件，故未来工作可考虑用自适应控制的方法在线更新模型参数以避免这个问题。

文献[14]中  $\mathcal{L}_1$  自适应控制与预测控制相结合的工作其实首次出现在文献[16]中，但是文献[16]并没有在实物上验证算法的可行性，且其使用的 Model Predictive Path Integral Control (MPPI) 对计算力的需求远高于文献[14]中使用的非线性模型预测控制器 (Nonlinear Model Predictive Control, NMPC)。 $\mathcal{L}_1$ -MPC 和  $\mathcal{L}_1$  自适应控制的原始思路和算法来自于文献[16]，但是文献[14]对这种思路进行了进一步验证和拓展。

图 1-2  $\mathcal{L}_1$ -MPC 相关成果<sup>[14]</sup>

### 1.2.3 研究现状小结

针对过驱动六旋翼无人机的轨迹追踪问题，文献[4]和[7]已经给出了现成的解决方案，但是方案中对模型误差和外力扰动的估计方法，即 EKF，存在估计速度不理想的问题。而文献[14]和[16]所展示的自适应控制方案提供了另一种思路，效果在四旋翼无人机上得到了验证。但是目前并没有工作将 $\mathcal{L}_1$ -MPC 运用于过驱动六旋翼无人机上，本课题希望填补此项空白。

## 1.3 研究内容和方法

主要研究内容是自适应模型预测控制算法的设计和验证。算法设计涉及研究 MPC 基本框架和 $\mathcal{L}_1$ 自适应控制理论、对系统进行建模、推导数学公式并将其用代码实现；算法的验证包括数值仿真、Gazebo 仿真和实物飞行实验。

为了完成主线研究，需要完成一系列辅助工作：实物无人机硬件的设计、零件加工和装配、Gazebo 仿真环境的搭建、PID 备份控制器和控制分配算法的设计、PX4 飞控固件的代码修改、动作捕捉系统的调试和使用、执行器驱动程序的编写等。笔者期望在 ROS2 框架下，使用 PX4 项目作为子模块，为六旋翼过驱动无人机设计一种自适应模型预测控制器。使用 ROS2 是主要为了减少在特定工程问题花费的时间，专注于算法的实现和验证，同时期望利用上 ROS2 提供的更加优秀的低延迟特性。

## 第 2 章 无人机硬件和系统建模

### 2.1 引言

本章节首先对倾转机臂型过驱动六旋翼无人机的硬件系统进行详细叙述：无人机硬件部分包括机械结构设计和电子、计算设备的选取，其中机械结构方面将重点叙述机臂倾转结构的设计。随后系统建模部分则将复杂的系统简化为单个刚体进行分析，得到运动学和动力学层面的常微分方程数学模型。

严格来说，“旋翼”和“螺旋桨”指代的其实是不同类型的执行器，但本文不对二词进行区分，均指代固定螺距的螺旋桨。

### 2.2 无人机硬件

#### 2.2.1 飞行平台概述

本课题所设计并研究的无人机的执行器由六个无刷电机-桨叶动力系统和六个驱动碳纤维管机臂的减速电机组成，又因其可以实现任意姿态的悬停和全向飞行（Omnidirectional Flight），故将其取名为 OmniHex。

OmniHex 的轴距为 680 毫米。设计时选用此数值的考虑如下：

- 一、零部件的通用性：因为市售的许多欠驱动六旋翼无人机轴距也为 680 毫米，设定轴距为此数值则便于使用现成的动力电机安装座以及为此尺寸优化的起落架，简化了设计过程，减少了硬件迭代的次数。
- 二、因为有一定的运输负载需求，所以轴距不能太小，否则无法安装高效率的大桨叶；另一方面又不希望飞行平台太大太重，否则将难以在拥挤的室内环境使用。

桨叶选用直径 13 英寸的高效型正反桨，由型号为 SunnySky V3508 580KV 的直流无刷电机驱动，此尺寸的桨叶是 680 毫米轴距的六旋翼无人机上可安装的最大桨叶。

OmniHex 机体主要结构为碳纤维材质，在保证机体刚性的情况下可使重量处于一个合理的数值范围内，整机带电池质量为 3.78 千克。关键部位，如电机座、机臂轴承夹座以及碳纤维板间的支撑柱的材质为航空铝材。机臂的倾转角度范围是  $[-7,7]$  弧度。图 2-1 展示了 OmniHex 在机臂倾转角度为 0 和  $\pm 1$  弧度下的外观。



图 2-1 OmniHex 不同的机臂倾转状态

### 2.2.2 机臂倾转结构

OmniHex 共有六个机臂倾转结构单元，位于两片中心碳纤维板的夹层处，且主要部件靠近机体中心，布局紧凑，有利于减小 OmniHex 的转动惯量，提高机动性。图 2-2 详细标注了单个机臂倾转单元的重要组件。

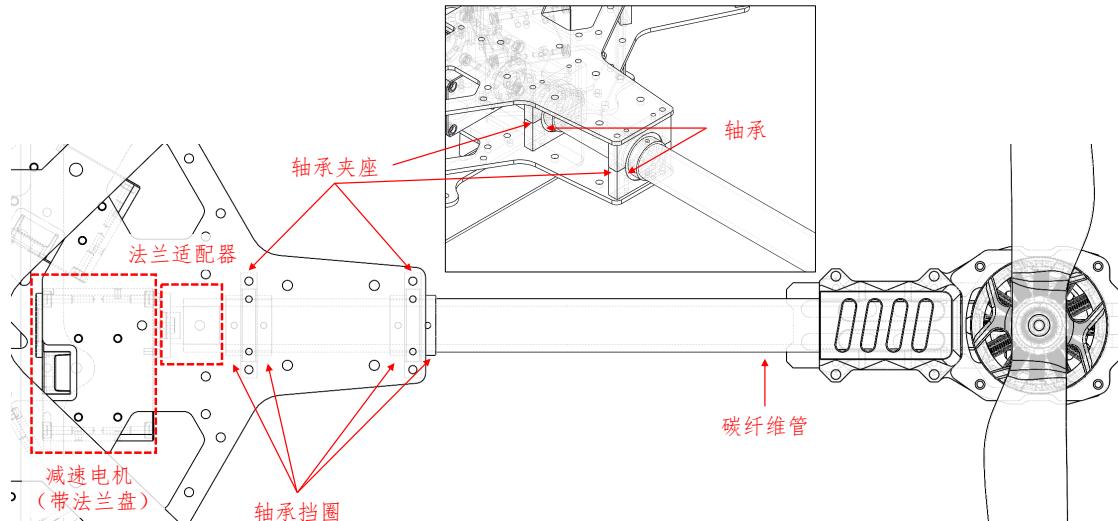


图 2-2 机臂倾转单元的组件

其中减速电机的型号是 Dynamixel-XH430-W210-T，工作电压为 12V，减速比为 210:1，通信接口为 TTL。此型号的减速电机总成与其他同体积的产品相比，力矩和转速性能优良，工作电压合适，且与机载电子设备通信方便。产品附带了有螺纹安装孔的法兰盘和对应图纸，为法兰适配器的设计和安装提供了便利和支持。

法兰适配器用于连接减速电机法兰和带孔的碳纤维管机臂，其与电机法兰盘通过法兰盘上的螺纹孔和适配器上的通孔用螺栓进行连接，与碳纤维管通过通孔

和螺栓螺母进行连接。由于 OmniHex 原型机上的此零件由三维打印制造，精度和耐用性低于期望，在经过多次飞行调试和碰撞后，装配孔位磨损导致机臂倾转运动出现松动和死区，影响了控制器的控制效果。

碳纤维管的外径为 16 毫米；对应地，轴承内径为 16 毫米，外径为 22 毫米。轴承与碳纤维管之间先由过盈配合装配，随后将轴承挡圈由通孔和螺栓螺母固定在轴承两侧，如此进一步限制轴承在碳纤维管上的轴向移动。考虑到轴承允许一定的偏转角，所以为了尽量减少机臂上径向力（比如旋翼产生的拉力）对减速电机的影响，在一个机臂单元中设计了两个轴承共同承受径向力。

轴承夹座的主要作用有二：一是夹紧机臂上轴承的外缘，二是作为中心两层碳纤维板中间的支撑。用于施加夹紧力的零件为长螺栓和防松螺母。图 2-3 展示了 OmniHex 实物上的机臂倾转结构，图中可以清楚地看到减速电机、法兰适配器、轴承夹座、挡圈、碳纤维管等组件，以及他们的装配方式。

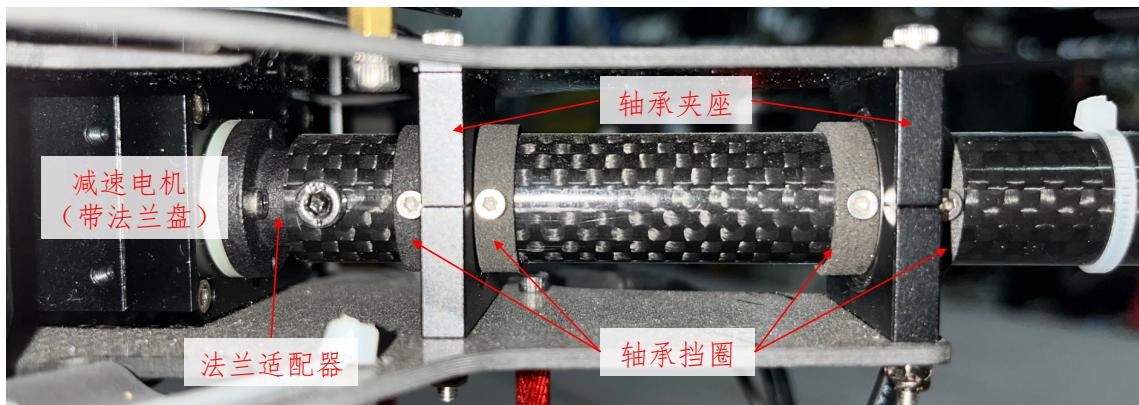


图 2-3 OmniHex 实物机臂倾转结构

### 2.2.3 电子和计算设备

OmniHex 上主要的电子和计算设备包括机载电脑、底层飞行控制器、无刷电机电子调速器和减速电机驱动器。机载电脑的型号是 INTEL NUC11PAHi7，主要运行计算负载比较大的任务；底层飞行控制器的型号是 Cube Orange，运行开源飞控 PX4 固件；而电子调速器依据底层飞行控制器的 PWM 输出指令控制直流无刷电机的转速，后者转动旋翼为 OmniHex 提供力和力矩。减速电机驱动器一方面为减速电机提供电力，另一方面通过 TTL 转 USB 模块连接至机载电脑，后者运行驱动程序控制减速电机的力矩、速度和位置。

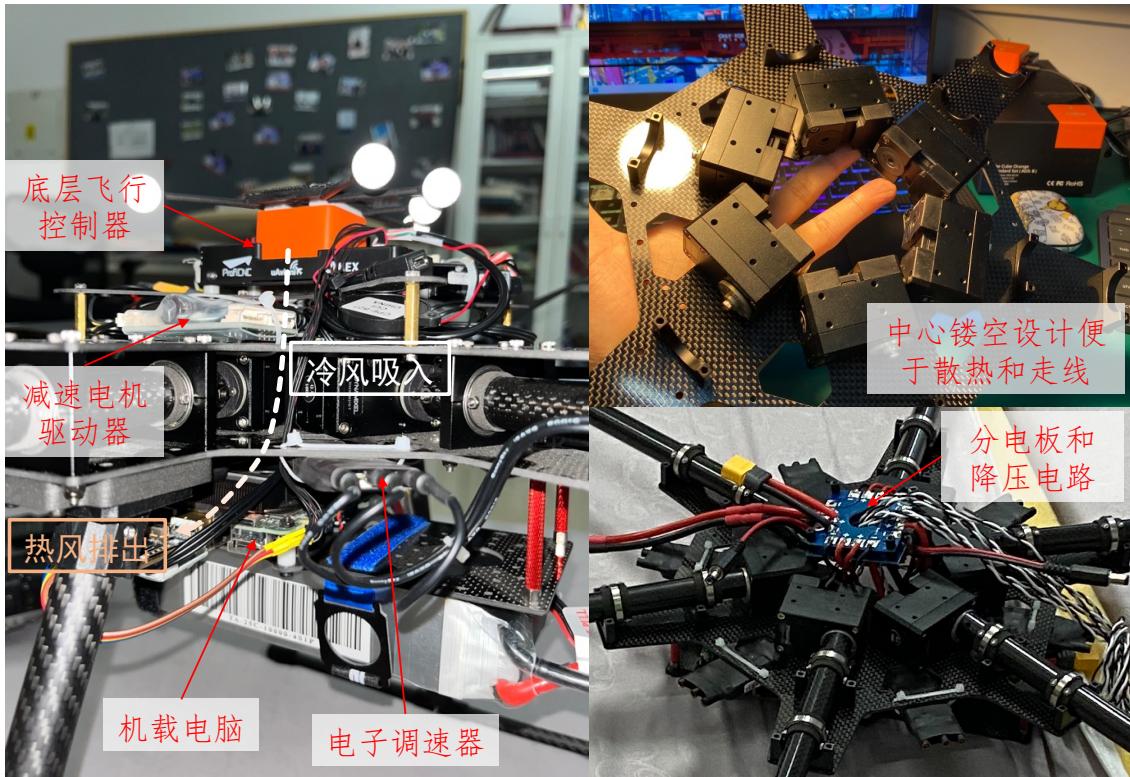


图 2-4 主要电子和计算设备的布局

如图 2-4 所示，底层飞行控制器处于 OmniHex 最顶端，此位置便于开发者频繁插拔信号线以及进行微处理器固件的烧录；机载电脑处于中心板下层的设备载板上，其与中心板有一定的垂直距离，较大的空间为插拔如 USB 和 HDMI 等大型接口提供了便利。机载电脑上用于散热的离心式风扇从上往下抽风，侧边出风，可为中心板间的减速电机以及降压电路提供一定程度的散热环境。

另外，因为机臂需要倾转，又无法采购到满足尺寸和电流需求的滑环，故在 OmniHex 上应对机臂线缆缠绕的方式是：为机臂旋转范围预留出足够长度的线缆。通过合理的整理和固定，这些线缆可以做到不阻碍机臂倾转，同时不干扰旋翼的正常工作。

## 2.3 系统建模

### 2.3.1 坐标系设置

世界坐标系为 XYZ-NED 坐标系，与 PX4 中的世界坐标系保持一致；机体坐标系为 XYZ-FRD 坐标系，原点设置在 OmniHex 中心板夹层空间的几何中心处，如此设置可方便 4.3 节控制分配关系的推导。图 2-5 直观地展示了世界坐标系的设置以及机体坐标系在 OmniHex 上的位置和姿态绑定关系。

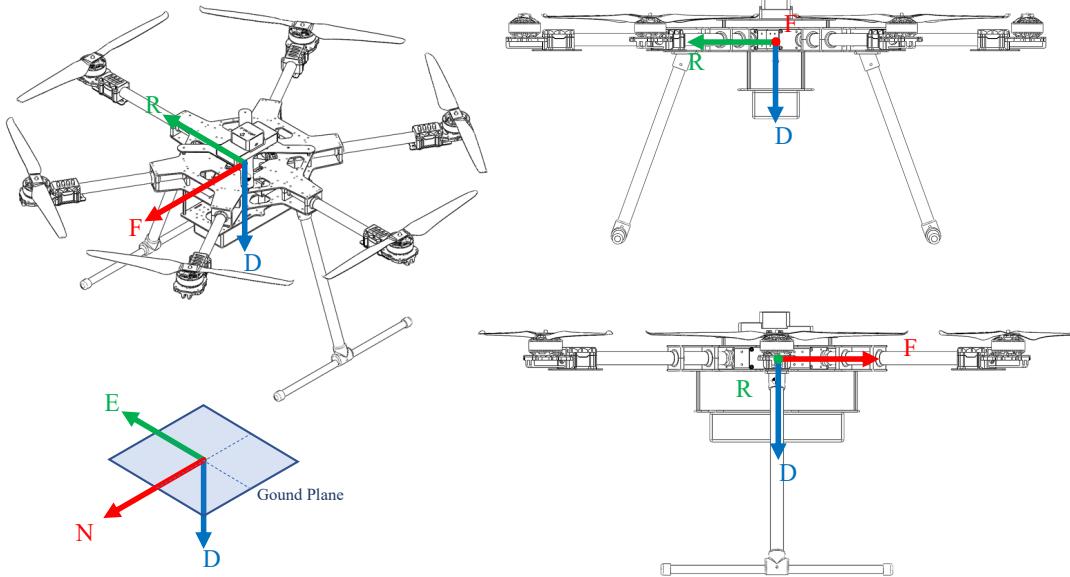


图 2-5 世界坐标系和机体坐标系设置

### 2.3.2 符号记号及说明

此条内容为后续章节所用到的部分符号及其对应的意义，详情见表 2-1，此表内容在后续章节均有效并将得到补充。

表 2-1 符号的意义和说明

| 符号                                    | 维度           | 意义                |
|---------------------------------------|--------------|-------------------|
| $\mathbf{p}$                          | $3 \times 1$ | 世界坐标系下位置 (m)      |
| $\mathbf{q} = [\omega, x, y, z]^\top$ | $4 \times 1$ | 表示姿态的单位四元数        |
| ${}_W\mathbf{v}$                      | $3 \times 1$ | 世界坐标系下线速度 (m/s)   |
| ${}_B\boldsymbol{\omega}$             | $3 \times 1$ | 机体坐标系下角速度 (rad/s) |
| ${}_B\mathbf{f}_a$                    | $3 \times 1$ | 机体坐标系下执行器合力 (N)   |
| ${}_B\boldsymbol{\tau}_a$             | $3 \times 1$ | 机体坐标系下执行器合力矩 (Nm) |
| $m$                                   | $1 \times 1$ | 机体质量 (kg)         |
| ${}_B\mathbf{d}_{com}$                | $3 \times 1$ | 机体坐标系下质心位置 (m)    |
| $J$                                   | $3 \times 3$ | 转动惯量矩阵            |
| $\mathbf{g} = [0, 0, 9.8]^\top$       | $3 \times 1$ | 世界坐标系下重力加速度       |
| ${}_W\mathbf{f}_{com}$                | $3 \times 1$ | 世界坐标系下对质心的主矢 (N)  |
| ${}_B\boldsymbol{\tau}_{com}$         | $3 \times 1$ | 机体坐标系下对质心的主矩 (Nm) |

### 2.3.3 系统运动学和动力学

运动学关系是系统位置姿态和其导数之间的关系，动力学关系是线速度与角速度和力与力矩之间的关系。本条先给出运动学关系，再结合单刚体的受力分析给出动力学关系。

由速度的定义可直接得到：

$$\dot{\mathbf{p}} = {}_W \mathbf{v} \quad (2-1)$$

姿态由四元数表示，四元数对时间的导数存在以下结论：

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \begin{bmatrix} 0 \\ {}_B \boldsymbol{\omega} \end{bmatrix} \quad (2-2)$$

其中 $\otimes$ 符号代表四元数乘法。上述二式共同构成了系统的运动学模型。

推导动力学模型的主要思路是：以对质心的主矢和主矩为桥梁，将牛顿-欧拉公式，即式(2-3)，与受力分析联系起来，得到 ${}_W \dot{\mathbf{v}}$ 、 ${}_B \dot{\boldsymbol{\omega}}$ 与力和力矩之间的关系。

$$\begin{bmatrix} {}_W \mathbf{f}_{com} \\ {}_B \boldsymbol{\tau}_{com} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \cdot {}_B \dot{\boldsymbol{\omega}} + {}_B \boldsymbol{\omega} \times (\mathbf{J} \cdot {}_B \boldsymbol{\omega}) \\ {}_B \boldsymbol{\tau}_a \end{bmatrix} \quad (2-3)$$

式(2-3)的第一行中使用的参考系是世界坐标系，通常牛顿-欧拉公式中的平动描述使用的是机体坐标系，但因为使用世界坐标系对后续章节的控制器设计有简化作用，所以此处使用了世界坐标系作为参考。

对系统的受力分析使用图 2-6 进行辅助说明。

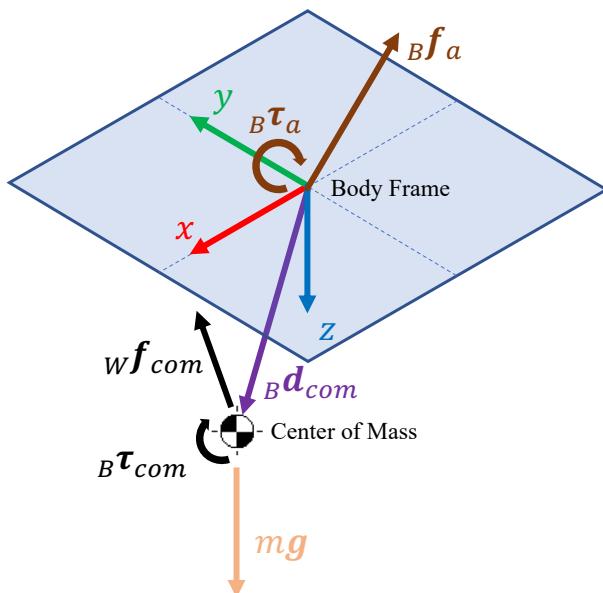


图 2-6 系统受力示意图

如图 2-6 所示,  $w\mathbf{f}_{com}$  和  $_B\boldsymbol{\tau}_{com}$  分别是对系统质心的主矢和主矩。根据力的平移定理, 作用于机体坐标系原点的  $_B\mathbf{f}_a$  可以等效为作用于质心的、同等的力, 外加一个力矩, 这个力矩等于  $-_B\mathbf{d}_{com} \times _B\mathbf{f}_a$  (此处  $_B\mathbf{d}_{com}$  是指向质心的向量), 于是主矢包括两个部分: 一是重力, 二是执行器产生的合力, 后者需要通过机体姿态四元数旋转到世界坐标系下。而力矩的作用效果与作用点无关, 故  $_B\boldsymbol{\tau}_a$  可直接等效于作用于质心的同等力矩。用数学表达式描述此段关系则是:

$$\begin{bmatrix} w\mathbf{f}_{com} \\ {}_B\boldsymbol{\tau}_{com} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} + \text{rotate}({}_B\mathbf{f}_a, \mathbf{q}) \\ {}_B\boldsymbol{\tau}_a - {}_B\mathbf{d}_{com} \times {}_B\mathbf{f}_a \end{bmatrix} \quad (2-4)$$

式中的旋转满足如下关系:

$$\begin{bmatrix} w_{DC} \\ \text{rotate}({}_B\mathbf{f}_a, \mathbf{q}) \end{bmatrix} = \mathbf{q} \otimes \begin{bmatrix} 0 \\ {}_B\mathbf{f}_a \end{bmatrix} \otimes \mathbf{q}^{-1} \quad (2-5)$$

即  $\text{rotate}(\mathbf{v}, \mathbf{q})$  表示基于单位四元数  $\mathbf{q}$  对三维向量  $\mathbf{v}$  进行旋转: 将  $\mathbf{v}$  扩展成  $\mathbf{v}_q = [0, \mathbf{v}]^\top$ , 四元数乘法运算得到  $\mathbf{v}'_q = \mathbf{q} \otimes \mathbf{v}_q \otimes \mathbf{q}^{-1}$ , 取  $\mathbf{v}'_q$  的向量部分作为输出。变量  $w_{DC}$  表示四元数中不关心的标量部分。

联立式(2-3)和(2-4), 得到  $w\dot{\mathbf{v}}$ 、 ${}_B\dot{\boldsymbol{\omega}}$  与力和力矩之间的关系:

$$\begin{bmatrix} w\dot{\mathbf{v}} \\ {}_B\dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{g} + \frac{\text{rotate}({}_B\mathbf{f}_a, \mathbf{q})}{m} \\ J^{-1}({}_B\boldsymbol{\tau}_a - {}_B\mathbf{d}_{com} \times {}_B\mathbf{f}_a - {}_B\boldsymbol{\omega} \times (J \cdot {}_B\boldsymbol{\omega})) \end{bmatrix} \quad (2-6)$$

## 2.4 本章小结

本章从 OmniHex 实物的概述出发, 对飞行器的外观、质量、材质和尺寸进行了简要介绍; 随后重点叙述了机臂倾转结构的设计和思路, 以及电子计算设备的配置和布局。

在之后的小节中, 本章定义了坐标系, 以及用于描述模型的变量符号, 再通过运动学公理、牛顿-欧拉公式、受力分析和力的平移定理, 得到单刚体的运动学和动力学公式, 整理如下。

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \\ w\dot{\mathbf{v}} \\ {}_B\dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{g} + \frac{\text{rotate}({}_B\mathbf{f}_a, \mathbf{q})}{m} \\ \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ {}_B\boldsymbol{\omega} \end{bmatrix} \\ J^{-1}({}_B\boldsymbol{\tau}_a - {}_B\mathbf{d}_{com} \times {}_B\mathbf{f}_a - {}_B\boldsymbol{\omega} \times (J \cdot {}_B\boldsymbol{\omega})) \end{bmatrix} \quad (2-7)$$

## 第3章 自适应模型预测控制

### 3.1 引言

本章先基于第二章的建模结论，结合 OmniHex 的实际特性设计约束，构造出最优控制问题（Optimal Control Problem, OCP）。在后继小节中，又以 OCP 为基础得到模型预测控制（Model Predictive Control, MPC）算法，并讨论 MPC 中由建模误差带来的问题，考虑到这个潜在问题，自然可以引出自适应控制器的主要思想和意义。本章节的最后推导了 OmniHex 适用的  $\mathcal{L}_1$  自适应算法，给出了增强名义模型 MPC 输出的具体方案，并对算法的实时性进行了优化。

如标题所示，本章是课题的重点研究内容，其中结论是主要研究成果。即便如此，后续章节却并非可有可无，而是作为本章结论的补充、辅助和验证的角色，重要性不减。另外，纵观课题的研究历程，对本章内容进行研究时已处于研究历程的后半段，前期准备工作占了较多工作量，而这些前期准备工作也将会以后续章节内容的方式呈现。

### 3.2 基于名义模型的模型预测控制

#### 3.2.1 最优控制问题

最优控制问题是模型预测控制中的求解单元。它包括以下几项属性：时域窗口（Time Horizon）、最优化目标、约束条件。求解最优控制问题可简洁地概括为：于给定的时域窗口内，在满足约束条件的情况下，求解为达到最优化目标所需要的系统控制量序列。为了构造最优控制问题并方便叙述，定义如表 3-1 的符号，作为表 2-1 的补充。

表 3-1 符号的意义和说明（补充 1）

| 符号              | 维度            | 意义                                   |
|-----------------|---------------|--------------------------------------|
| $\mathbf{x}_k$  | $19 \times 1$ | 第 $k$ 个离散节点的状态向量                     |
| $\mathbf{u}_k$  | $6 \times 1$  | 第 $k$ 个离散节点的输入向量                     |
| $\mathbf{e}_k$  | $18 \times 1$ | 第 $k$ 个离散节点的代价向量                     |
| $\mathbf{h}_k$  | $12 \times 1$ | 第 $k$ 个离散节点的约束向量                     |
| $(\cdot)_{err}$ | $3 \times 1$  | 测量值与参考值之差 ( $\mathbf{q}_{err}$ 单独说明) |
| $\mathbf{B}$    | $6 \times 12$ | 控制分配矩阵                               |

表 3-1 (续表)

| 符号  | 维度             | 意义                         |
|---|----------------|----------------------------|
| $\mathbf{F} = [f_1^2, f_2^2 \cdots f_6^2]^\top$ | $6 \times 1$   | 电机推力平方向量 ( $N^2$ )         |
| $N$   | $1 \times 1$   | 时域窗口离散化等分数                 |
| $\mathbf{Q}$                                    | $18 \times 18$ | 中间节点对 $\mathbf{e}$ 的代价权重矩阵 |
| $\mathbf{Q}_N$                                  | $18 \times 18$ | 终止节点对 $\mathbf{e}$ 的代价权重矩阵 |
| $\mathbf{R}$                                    | $6 \times 6$   | 对 $\mathbf{u}$ 的代价权重矩阵     |

表 3-1 中列举了状态向量、输入向量、代价向量和约束向量，他们的定义如下：

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ {}^W\mathbf{v} \\ {}^B\boldsymbol{\omega} \\ {}^B\mathbf{f}_a \\ {}^B\boldsymbol{\tau}_a \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} {}^B\dot{\mathbf{f}}_a \\ {}^B\dot{\boldsymbol{\tau}}_a \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \mathbf{p}_{err} \\ \mathbf{q}_{err} \\ {}^W\mathbf{v}_{err} \\ {}^B\boldsymbol{\omega}_{err} \\ {}^B\mathbf{f}_a \\ {}^B\boldsymbol{\tau}_a \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} {}^W\mathbf{v} \\ {}^B\boldsymbol{\omega} \\ \mathbf{F} \end{bmatrix} \quad (3-1)$$

状态向量是包括了位置、姿态、速度、角速度以及执行器合力与合力矩 (wrench, 后文均用 wrench 指代“合力与合力矩”) 的列向量。系统输入向量是 wrench 对时间的导数，这样便于在代价函数里对 wrench 的变化率进行惩罚。代价向量主要由各状态的误差和 wrench 组成，各项值越大，则最终代价越大，这点在后文的代价函数的叙述中也可体现。位置误差  $\mathbf{p}_{err}$ 、线速度误差  ${}^W\mathbf{v}_{err}$  和角速度误差  ${}^B\boldsymbol{\omega}_{err}$  均由当前值减去参考值得到。姿态误差则满足下式：

$$\mathbf{q} \otimes \mathbf{q}_{ref}^{-1} = \begin{bmatrix} {}^W \\ \mathbf{q}_{err} \end{bmatrix} \quad (3-2)$$

约束向量中是被上下界约束的各变量，其中的  $\mathbf{F}$  可由控制分配矩阵的伪逆  $\mathbf{B}^\dagger$  求出：

$$\mathbf{y} \triangleq \mathbf{B}^\dagger \begin{bmatrix} {}^B\mathbf{f}_a \\ {}^B\boldsymbol{\tau}_a \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{12} \end{bmatrix} \quad (3-3)$$

$$\mathbf{F} = [y_1^2 + y_2^2 \quad \dots \quad y_{11}^2 + y_{12}^2]^\top \quad (3-4)$$

控制分配相关内容将在第 4 章中作详细讨论。

代价函数采用常用的离散二次型形式，为每个离散时间节点的代价向量与输入向量的加权二次型：

$$\sum_{k=0}^{N-1} (\mathbf{e}_k^\top \mathbf{Q} \mathbf{e}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k) + \mathbf{e}_N^\top \mathbf{Q}_N \mathbf{e}_N \quad (3-5)$$

期望式(3-5)所示的代价在时域窗口内达到最小，即误差向量尽量小：用最小的 wrench 尽可能跟踪上目标轨迹；输入向量尽可能小：wrench 的变化率不要太大，对系统稳定性有好处。

在使代价函数往最小值逼近的同时，期望满足下列约束条件：约束向量在一定范围内、输入向量在一定范围内、系统满足微分方程约束、系统的初始状态为当前开始求解 OCP 时的状态，前两项属于不等式约束，后两项属于等式约束。其中微分方程约束可由式(2-7)扩充得到：

$$\dot{\mathbf{x}} = \begin{bmatrix} {}^W\boldsymbol{\nu} \\ \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ {}_B\boldsymbol{\omega} \end{bmatrix} \\ \boldsymbol{g} + \frac{\text{rotate}({}_B\boldsymbol{f}_a, \boldsymbol{q})}{m} \\ J^{-1}(\boldsymbol{\tau}_a - {}_B\boldsymbol{d}_{com} \times {}_B\boldsymbol{f}_a - {}_B\boldsymbol{\omega} \times (J \cdot {}_B\boldsymbol{\omega})) \\ {}_B\dot{\boldsymbol{f}}_a \\ {}_B\dot{\boldsymbol{\tau}}_a \end{bmatrix} \triangleq \text{dyn}(\mathbf{x}, \mathbf{u}) \quad (3-6)$$

基于上述定义，可以构造 OCP 求解时域窗口内的最优输入序列  $[\mathbf{u}_0 \cdots \mathbf{u}_{N-1}]$ ，在满足约束的情况下使代价函数最小：

$$\begin{aligned} & \min_{[\mathbf{u}_0 \cdots \mathbf{u}_{N-1}]} \sum_{k=0}^{N-1} (\mathbf{e}_k^\top \mathbf{Q} \mathbf{e}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k) + \mathbf{e}_N^\top \mathbf{Q}_N \mathbf{e}_N \\ & \text{s.t. } \begin{cases} \mathbf{h}_k \in [\mathbf{h}_{lb}, \mathbf{h}_{ub}] \\ \mathbf{u}_k \in [\mathbf{u}_{lb}, \mathbf{u}_{ub}] \\ \dot{\mathbf{x}} = \text{dyn}(\mathbf{x}, \mathbf{u}) \\ \mathbf{x}_0 = \mathbf{x}(t) \end{cases} \end{aligned} \quad (3-7)$$

至此，OCP 已被完整定义，如式(3-7)所示。OCP 的具体求解不在本课题的研究范围内，实验中使用开源工具 acados<sup>[19]</sup>，用序列二次规划（Sequential Quadratic Programming, SQP）的方法求解上述非线性优化问题（Non-linear Programming, NLP）。SQP 方法具有效率高、收敛性好、边界搜索能力强等优点，适用于需要快速求解 NLP 的场景。

### 3.2.2 模型预测控制

MPC 以 OCP 的求解为基础：每一个控制周期内调用求解器求解 OCP，得到解序列  $[\mathbf{u}_0 \cdots \mathbf{u}_{N-1}]$ ，取  $\mathbf{u}_0$  作为当前控制周期的系统输入，使系统状态发生变化。一个周期结束，进入下一个周期，时域窗口往未来移动一个控制周期，重复上述操作。即 MPC 用滚动窗口的方式不断进行优化，以此形成反馈闭环，使系统状态往使代价函数最小的最优状态上收敛。

若 MPC 控制周期为 0.01 秒，OCP 的时域窗口为 1 秒，时域窗口离散化等分数  $N = 20$ ，则在这种设置下，相邻代价节点之间的时间间隔为 0.05 秒，参考轨迹之间的采样周期至多为 0.01 秒。

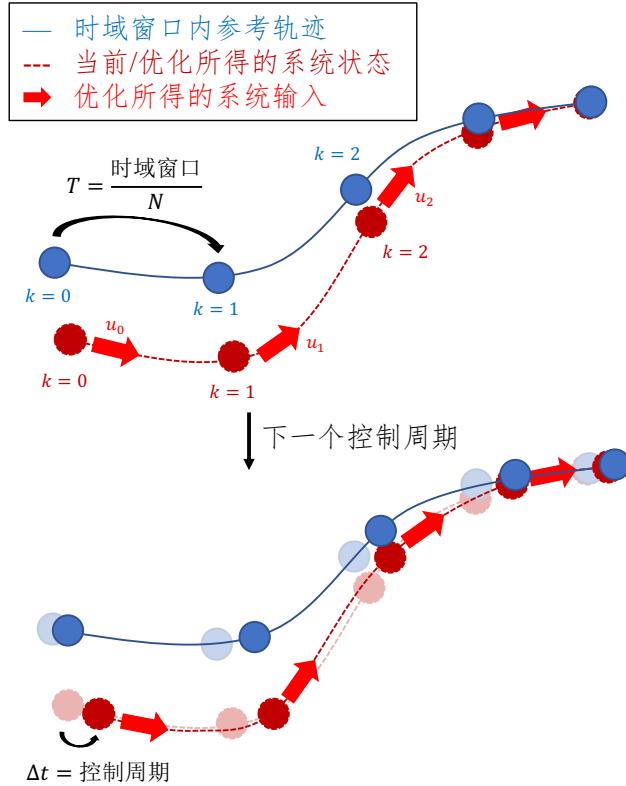


图 3-1 MPC 的工作流程

现结合图 3-1 对 MPC 的工作流程进行详细叙述。图中展示了两个周期，在第一个周期中，OCP 的输入为当前 ( $k = 0$ ) 的系统状态和时域窗口内的参考轨迹，输出为解序列  $[u_0 \cdots u_{N-1}]$ ，以及用这些解（亮红色箭头）参考系统模型所预测出的未来状态（暗红线和点）。MPC 取  $u_0$  为真正输入至物理系统的控制量，随后系统进入下一周期，时域窗口相应地向后滑动一个控制周期。

由于扰动和模型误差的存在，进入下一周期时，系统状态往往与上一周期用  $u_0$  和系统模型预测得到的状态不同，即上一周期得到的  $[u_1 \cdots u_{N-1}]$  往往不再是最优解了，故丢弃。这时用新的参考轨迹和当前状态作为输入再次求解 OCP，同样取  $u_0$  输入至物理系统。正是这样的算法将负反馈引入系统：就算有小幅度扰动和模型误差的存在，系统状态也不至于发散。

式(3-1)中定义的系统输入向量是 wrench 对时间的导数，是为了方便对其施加不等式约束，但在实际 OmniHex 系统中，底层的执行系统所需的输入是 wrench 本身，其数值等于上一周期的 wrench 加上本周期  $u_0$  与控制周期的乘积。

通常，控制周期小于相邻代价节点之间的时间间隔；较短的控制周期有利于提升 MPC 的鲁棒性和稳定性；较长的时域窗口则提供良好的预测性，通常体现在参考轨迹有剧烈变化时系统也能较为平滑且准确地进行追踪。

### 3.2.3 潜在问题

设计 MPC 时，很难得到系统精确的数学模型，其原因一方面是系统参数较难估计准确，另一方面是建模时的前提假设和简化使得未建模动态存在。换言之，欲使 MPC 有较好的性能，一方面需要精细调节模型参数，另一方面需要尽可能完整地捕捉实际物理系统的动态特性。考虑到参数调节的繁琐程度以及建模时面面俱到的难度，本课题处理模型误差的方法是使用  $\mathcal{L}_1$  自适应控制理论对基于名义模型的 MPC 进行增强，结合成名为  $\mathcal{L}_1$ -MPC 的方法，此种方法将在下一小节中详细讨论。

另一个潜在问题是在状态偏离参考轨迹太多，或参考轨迹在约束条件下是无法完成的时候，求解器无法求出当前控制周期的解，此时则应该考虑用何种控制量代替 OCP 的解作为系统的控制输入。本课题给出的处理方法是使用成熟的带前馈的 PID 控制器作为备份控制器，在 MPC 失效时获取 OmniHex 的控制权，确保其飞行安全。这些内容将在第 4 章中得到更具体的阐述。

## 3.3 $\mathcal{L}_1$ 自适应控制器

### 3.3.1 主要思想和意义

模型中的不确定性来源于参数误差、未建模动态、执行系统对指令的跟踪误差以及外力扰动，而状态估计的准确性不在此课题的讨论范围内。基于上述假设，可对系统动态方程进行降维，只考虑式(3-6)的动力学部分。再由于对系统输入的 wrench 可以改变系统的所有状态量，于是可以认为将所有不确定性表达成 wrench 的形式是合理的，自然可以在 wrench 上对系统进行补偿。

本课题设计的  $\mathcal{L}_1$  自适应控制器基于上述思想，以 MPC 的输出量、系统当前状态为输入，迭代地计算出系统所需的 wrench 补偿量，把补偿量和 MPC 的输出量之和作为最终指令输出至执行器系统。期望在名义模型与实际模型差别很大的情况下，自适应控制器也能使系统按照 MPC 所期望的那样进行运动。

使用  $\mathcal{L}_1$  自适应控制是解决 MPC 中模型误差的一个有效方法，此种方法不需要提前采集数据进行拟合或学习。此小节的后续条目将会给出具体的算法，其有效性则将在第 5 章得到验证。

### 3.3.2 自适应算法

为方便叙述，先对表 3-1 所定义的符号进行补充。

表 3-2 符号的意义和说明（补充 2）

| 符号                               | 维度           | 意义                            |
|----------------------------------|--------------|-------------------------------|
| ${}_B f_{\mathcal{L}_1}$         | $3 \times 1$ | $\mathcal{L}_1$ 控制器的输出力 (N)   |
| ${}_B f_{\Delta}$                | $3 \times 1$ | 力不确定性 (N)                     |
| ${}_B \tau_{\mathcal{L}_1}$      | $3 \times 1$ | $\mathcal{L}_1$ 控制器的输出力矩 (Nm) |
| ${}_B \tau_{\Delta}$             | $3 \times 1$ | 力矩不确定性 (Nm)                   |
| $u_{\mathcal{L}_1}$              | $6 \times 1$ | $\mathcal{L}_1$ 控制器的输出向量      |
| $\sigma, \hat{\sigma}$           | $6 \times 1$ | 不确定性与其估计值                     |
| $z, \hat{z}$                     | $6 \times 1$ | 降维状态向量与其估计值                   |
| $\mathbb{I}_i$                   | $i \times i$ | 单位矩阵                          |
| $\mathbb{O}_i$                   | $i \times i$ | 零矩阵                           |
| $\mathcal{A}, \hat{\mathcal{A}}$ | $6 \times 1$ | 理想系统动态与其估计值                   |
| $\mathcal{B}$                    | $6 \times 6$ | 输入增益矩阵                        |
| $R$                              | $3 \times 3$ | 姿态旋转矩阵                        |
| $D$                              | $3 \times 3$ | ${}_B d_{com}$ 对应的斜对称矩阵       |
| $A$                              | $6 \times 6$ | 自适应增益矩阵                       |
| $T_s$                            | $1 \times 1$ | 离散周期（控制周期）                    |

对式(3-6)进行降维，令  $\dot{z} = [{}_W \dot{v}, {}_B \dot{\omega}]^T$ ，并补充不确定性和补偿项，得到下式：

$$\begin{bmatrix} {}_W \dot{v} \\ {}_B \dot{\omega} \end{bmatrix} = \begin{bmatrix} g + \frac{\text{rotate}({}_B f_a + {}_B f_{\mathcal{L}_1} + {}_B f_{\Delta}, q)}{m} \\ J^{-1}({}_B \tau_a + {}_B \tau_{\mathcal{L}_1} + {}_B \tau_{\Delta} - {}_B d_{com} \times ({}_B f_a + {}_B f_{\mathcal{L}_1} + {}_B f_{\Delta}) - {}_B \omega \times (J \cdot {}_B \omega)) \end{bmatrix} \quad (3-8)$$

并将其写成理想动态项和不确定性项的形式：

$$\dot{z} = \underbrace{\begin{bmatrix} g + \frac{\text{rotate}({}_B f_a, q)}{m} \\ J^{-1}({}_B \tau_a - {}_B d_{com} \times {}_B f_a - {}_B \omega \times (J \cdot {}_B \omega)) \end{bmatrix}}_{\mathcal{A}} + \underbrace{\begin{bmatrix} R & \mathbb{O}_3 \\ \frac{m}{m} & J^{-1} \end{bmatrix}}_{\mathcal{B}} \left( \underbrace{\begin{bmatrix} {}_B f_{\mathcal{L}_1} \\ {}_B \tau_{\mathcal{L}_1} \\ u_{\mathcal{L}_1} \end{bmatrix}}_{\sigma} + \underbrace{\begin{bmatrix} {}_B f_{\Delta} \\ {}_B \tau_{\Delta} \end{bmatrix}}_{\sigma} \right) \quad (3-9)$$

为了表示方便，考虑了不确定性之后，系统动态可以更简洁地写成：

$$\dot{z} = \mathcal{A} + \mathcal{B}(u_{\mathcal{L}_1} + \sigma) \quad (3-10)$$

其中  $\mathcal{A}$  和  $\mathcal{B}$  是每周期更新的，执行器 wrench 来源于 MPC 的输出，其他变量的值来源于传感器和状态估计器。

在欠驱动多旋翼上，不确定性一般被分为匹配的（matched）不确定性和不匹配（unmatched）的不确定性，因为欠驱动多旋翼无法产生机体坐标系 XY 平面上的力，于是出现在 XY 平面上的不确定性无法直接被抵消，这一部分的不确定性被称作为不匹配的不确定性。而 OmniHex 则可以通过执行器的作用直接抵消 XY 平面上的不确定性，所以不存在不匹配的不确定性。

$\mathcal{L}_1$  自适应控制的出发点是对  $\sigma$  进行估计，即获取  $\hat{\sigma}$ ，然后用某种方法通过  $u_{\mathcal{L}_1}$  将其抵消。将  $\hat{\sigma}$  带入式(3-10)得到：

$$\dot{\hat{\mathbf{z}}} = \hat{\mathcal{A}} + \mathcal{B}(\mathbf{u}_{\mathcal{L}_1} + \hat{\sigma}) \quad (3-11)$$

希望  $\hat{\mathbf{z}} \rightarrow \mathbf{z}$ ,  $\hat{\sigma} \rightarrow \sigma$ , 即希望不确定性估计值向真实值收敛，同时状态的估计值也向真实值收敛，伴随而来同理有  $\hat{\mathcal{A}} \rightarrow \mathcal{A}$ ，这一连串代表的意义是系统动态方程向名义动态方程收敛。引入记号  $\tilde{\mathbf{z}} = \hat{\mathbf{z}} - \mathbf{z}$ ,  $\tilde{\sigma} = \hat{\sigma} - \sigma$  构造如下线性系统：

$$\dot{\tilde{\mathbf{z}}} = \mathcal{A}\tilde{\mathbf{z}} + \mathcal{B}\tilde{\sigma} \quad (3-12)$$

其中  $\mathcal{A}$  为 Hurwitz 矩阵，这样能保证这个系统的平衡点  $(\tilde{\mathbf{z}}, \tilde{\sigma}) = (0,0)$  为渐进稳定的，以此作为能估计出不确定性的一个先决条件。注意此处仅有  $\mathcal{A}$  为 Hurwitz 矩阵是不能保证  $(\tilde{\mathbf{z}}, \tilde{\sigma})$  的有界性的，也不能决定系统状态转移过程的表现和性质。

现假设能得到可靠的  $\hat{\sigma}$ ，即  $\hat{\sigma} = \sigma$ ，并以此为基础讨论消去不确定性影响的方法。朴素的思想是令  $\mathbf{u}_{\mathcal{L}_1} + \hat{\sigma} = 0$ ，这样式(3-12)变为  $\dot{\tilde{\mathbf{z}}} = \mathcal{A}$ ，完全服从理想动态。但是考虑到现实的系统，执行器是没办法对超出自身能力范围的不确定性补偿的：比如有些  $\sigma$  频率太高，系统从物理上没办法补偿它们，若算法却还将它们一起纳入  $\mathbf{u}_{\mathcal{L}_1}$  的计算，则易引起系统震荡，限制了算法对  $\sigma$  感知的增益，也就间接地使对不确定性的估计速度下降。 $\mathcal{L}_1$  自适应理论给出的解决办法是令：

$$\mathbf{u}_{\mathcal{L}_1} = \text{LPF}(\hat{\sigma}) \quad (3-13)$$

即对  $\hat{\sigma}$  低通滤波之后作为  $\mathbf{u}_{\mathcal{L}_1}$ ，通过适当调节截止频率，能够剔除上述负面影响，并允许设置更大的自适应增益，以此提高自适应性能。

接下来讨论获取  $\hat{\sigma}$  的方法，式(3-11)本身无法完成对  $\hat{\mathbf{z}}$  和  $\hat{\sigma}$  的更新，因为输入  $\tilde{\sigma}$  无从获取。而利用式(3-10)形式的状态预测器，则可在式(3-12)，和自适应控制律  $\hat{\sigma} = \text{adapt}(\tilde{\mathbf{z}})$  的帮助下更新  $\hat{\mathbf{z}}$  和  $\hat{\sigma}$ 。但式(3-10)中存在的  $\hat{\mathcal{A}}$  会使分析较为复杂，所以利用式(3-11)进行一次代换，将式(3-11)和式(3-12)相加得到另一种状态预测器的形式：

$$\dot{\hat{\mathbf{z}}} = \mathcal{A} + \mathcal{B}(\mathbf{u}_{\mathcal{L}_1} + \hat{\sigma}) + \mathcal{A}\tilde{\mathbf{z}} \quad (3-14)$$

利用此式可对降维状态的估计值进行更新。

自适应控制律  $\hat{\sigma} = \text{adapt}(\tilde{\mathbf{z}})$  本质是由状态误差估计系统不确定性的方法，其严格推导不在本课题的研究范围内，此处仅给出离散系统适用的结论<sup>[16]</sup>：

$$\hat{\sigma}_k = -\mathbf{B}^{-1}(e^{AT_s} - \mathbb{I}_6)^{-1}\mathbf{A}e^{AT_s}\tilde{\mathbf{z}}_k \quad (3-15)$$

式(3-15)中的  $e^{AT_s}$  表示计算  $\mathbf{A}$  对应的以  $T_s$  为周期的离散化矩阵：

$$e^{AT_s} = \mathcal{L}^{-1}((s\mathbb{I}_6 - \mathbf{A})^{-1}, T_s) \quad (3-16)$$

其中  $\mathcal{L}^{-1}(\mathbf{M}, T_s)$  是矩阵  $\mathbf{M}$  作拉氏逆变换，并令时域变量  $t = T_s$ 。

在每一个  $\mathcal{L}_1$  自适应控制的控制周期  $T_s$  内，控制器将会按顺序进行以下运算或操作：(1) 根据当前系统状态更新  $\mathbf{B}$  和  $\mathbf{z}$ ；(2) 由式(3-15)更新  $\hat{\sigma}$ ；(3) 由式(3-13)更新  $\mathbf{u}_{\mathcal{L}_1}$ ；(4) 由式(3-14)和  $T_s$  得到下一个周期的  $\hat{\mathbf{z}}$ ：

$$\hat{\mathbf{z}}_{k+1} = \hat{\mathbf{z}}_k + T_s \dot{\hat{\mathbf{z}}} \quad (3-17)$$

算法初始化使  $\mathbf{z} = \hat{\mathbf{z}}$ 。

至此， $\mathcal{L}_1$ -MPC 设计完毕。 $\mathcal{L}_1$ -MPC 的输出为  $\mathbf{u}_{\mathcal{L}_1} + \mathbf{u}_{mpc}$ ，由下游的控制分配器接收并计算每个执行单元的指令。

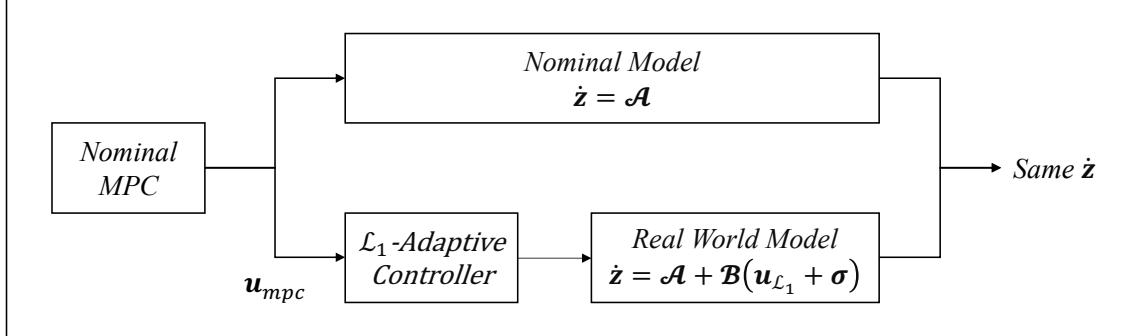


图 3-2 名义模型 MPC 和自适应环节在系统中的角色

如图 3-2 所示， $\mathcal{L}_1$  自适应控制器处于被控对象和名义模型 MPC 之间。 $\mathcal{L}_1$  自适应控制的角色是使“自适应-真实模型”串联系统对于 MPC 来说，与名义模型保持一致。故  $\mathcal{L}_1$  自适应控制是一种“模型参考自适应控制”。

### 3.3.3 实时运行优化

如前文所述，自适应控制器的每个周期内均需要计算式(3-15)，涉及到矩阵求逆以及拉氏逆变换符号运算，在算法实现时应当避免。

利用分块矩阵求逆相关结论和  $\mathbf{R} \in \text{SO}(3)$  可得：

$$\mathbf{B}^{-1} = \begin{bmatrix} \mathbf{R} & \mathbb{O}_3 \\ \frac{m}{\mathbf{R}} & \mathbb{O}_3 \\ -\mathbf{J}^{-1}\mathbf{D} & \mathbf{J}^{-1} \end{bmatrix}^{-1} = \begin{bmatrix} (\frac{\mathbf{R}}{m})^{-1} & \mathbb{O}_3 \\ m\mathbf{D}\mathbf{R}^{-1} & \mathbf{J} \end{bmatrix} = \begin{bmatrix} m\mathbf{R}^\top & \mathbb{O}_3 \\ m\mathbf{D}\mathbf{R}^\top & \mathbf{J} \end{bmatrix} \quad (3-18)$$

同时注意到对  $\mathbf{B}$  本身的更新涉及到  $\mathbf{J}^{-1}$ ，在程序初始化时计算一次即可。同理，在按需更新的思想下，式(3-15)中的常值  $(e^{\mathbf{A}T_s} - \mathbb{I}_6)^{-1}\mathbf{A}e^{\mathbf{A}T_s}$  也只需在初始化时计算一次即可。采用上述方法进行优化后，自适应控制器的每一步迭代耗时从 1ms 左右下降到 0.005ms。

### 3.4 本章小结

模型预测控制凭借 OCP 的最优化、预测性和灵活性，是解决精确轨迹跟踪问题的优秀方案。但因为模型误差的存在，单使用名义模型的 MPC 往往性能达不到理想水平，需要借某种手段弥合名义和实际模型之间的差距。针对这个问题，本章提出了适用于过驱动六旋翼无人机的  $\mathcal{L}_1$ -MPC，完整地给出了自适应模型预测控制的数学公式和算法流程，并详细地介绍了其背后的设计动机和思路，是本课题的核心研究成果。

## 第 4 章 带前馈的 PID 控制

### 4.1 引言

在系统当前状态与参考轨迹偏差较大，或是有较大扰动的时候，OCP 易求解失败。对于地面机器人来说，OCP 求解失败并不是严重的问题，处理方式也相对简单，只需要将执行器输出设为零即可避免严重事故，但是对于 OmniHex 这种昂贵且维护代价高的空中机器人来说，一旦求解失败，则可能出现失去稳定并坠毁的严重后果，所以设计备份控制并不是无关紧要的事情。

本章将基于 PX4 中包含的相关算法，给出带前馈的 PID 控制的数学表达式，其中包括相互独立无耦合的位置控制和姿态控制器。位置控制器包括外层位置环和内层速度环，输出加速度设定点。姿态控制器包括外层姿态环和内层角速度环，输出角加速度设定点。

除了给出控制器的设计，本章还会给出相应的 wrench 的计算方法，以及 wrench 到底层执行器指令的转换。wrench 到底层执行器指令的转换又被称为控制分配（Control Allocation, CA），涉及到执行器建模、分配算法的设计等内容。为方便叙述，需对符号说明表进行如下补充。

表 4-1 符号的意义和说明（补充 3）

| 符号  | 维度           | 意义                           |
|---|--------------|------------------------------|
| $\mathbf{p}_{ref}$  | $3 \times 1$ | 参考位置 (m)                     |
| $w\mathbf{v}_{sp}$  | $3 \times 1$ | 位置环输出速度设定点 (m/s)             |
| $w\mathbf{v}_{ref}$   | $3 \times 1$ | 参考速度 (m/s)                   |
| $w\dot{\mathbf{v}}_{sp}$  | $3 \times 1$ | 速度环输出的加速度设定点 ( $m/s^2$ )     |
| $\mathbf{P}_{pos}$  | $3 \times 1$ | 位置环比例增益                      |
| $\mathbf{P}_{vel}, \mathbf{I}_{vel}, \mathbf{D}_{vel}, \mathbf{F}_{vel}$        | $3 \times 1$ | 速度环比例、积分、微分、前馈增益             |
| $\mathbf{q}_{ref}$  | $4 \times 1$ | 参考姿态的四元数表示                   |
| $B\boldsymbol{\omega}_{sp}$   | $3 \times 1$ | 姿态环输出角速度设定点 (rad/s)          |
| $B\boldsymbol{\omega}_{ref}$  | $3 \times 1$ | 参考机体角速度 (rad/s)              |
| $B\dot{\boldsymbol{\omega}}_{sp}$   | $3 \times 1$ | 角速度环输出的角加速度设定点 ( $rad/s^2$ ) |
| $\mathbf{P}_\omega, \mathbf{I}_\omega, \mathbf{D}_\omega, \mathbf{F}_\omega, K$ | $3 \times 1$ | 角速度环增益                       |
| $\tau_x, \tau_y, \tau_z$  | $1 \times 1$ | 姿态环输出增益                      |
| $p_z \in [0,1]$   | $1 \times 1$ | 姿态环中机体 z 轴权重                 |
| $\mathbf{q}_{sp}$   | $4 \times 1$ | 姿态环中的四元数设定点                  |

## 4.2 控制器设计

### 4.2.1 位置控制

位置控制器包括外层位置环和内层速度环，以世界坐标系为参考，输出加速度设定点。

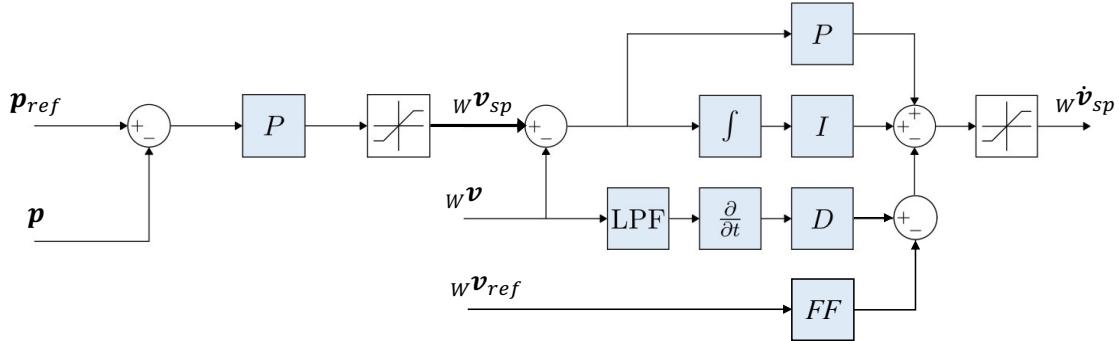


图 4-1 位置控制器框图

如图 4-1 所示，位置环为比例控制，速度环为比例-积分-微分-前馈控制。记图中的饱和环节为  $\text{sat}(\cdot)$ ，并定义  $\odot$  运算：

$$A_{m \times n} \odot B_{m \times n} = C_{m \times n} \text{ s.t. } C_{i,j} = A_{i,j} B_{i,j} \quad (4-1)$$

则位置控制器算法可以表达为下式：

$$\begin{cases} {}^w v_{sp} = \text{sat}\left(P_{pos} \odot (p_{ref} - p)\right) \\ e_v \triangleq {}^w v_{sp} - {}^w v \\ {}^w \dot{v}_{sp} = \text{sat}\left(P_{vel} \odot e_v + I_{vel} \odot \int_0^t e_v dt - D_{vel} \odot \frac{\text{LPF}({}^w v)}{dt} + F_{vel} \odot {}^w v_{ref}\right) \end{cases} \quad (4-2)$$

### 4.2.2 姿态控制

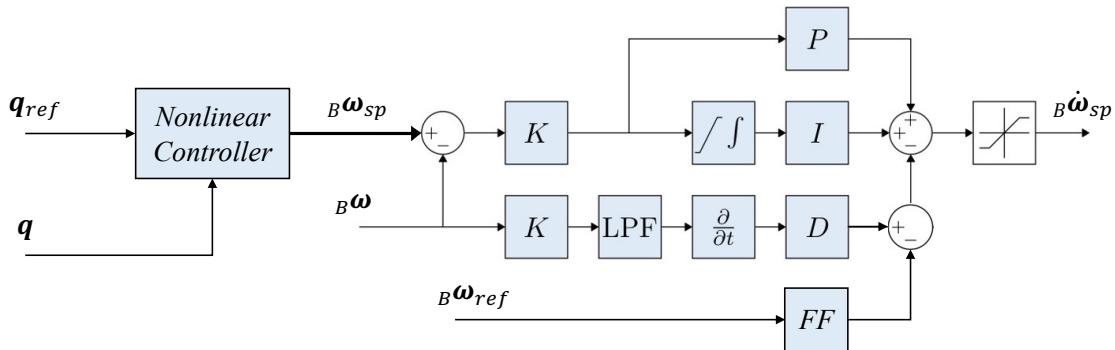


图 4-2 姿态控制器框图

如图 4-2 所示，姿态控制包括基于四元数的非线性姿态环，和比例-积分-微分-前馈控制的角速度环，最终的输出是机体坐标系下的角加速度。

类似位置控制器的速度环，角速度环的表达式如下：

$$\begin{cases} \boldsymbol{e}_\omega \triangleq {}_B\boldsymbol{\omega}_{sp} - {}_B\boldsymbol{\omega} \\ {}_w\dot{\boldsymbol{\omega}}_{sp} = \text{sat}\left(K\left(\boldsymbol{P}_\omega \odot \boldsymbol{e}_\omega + \boldsymbol{I}_\omega \odot \int_0^t \boldsymbol{e}_\omega dt - \boldsymbol{D}_\omega \odot \frac{\text{LPF}({}_B\boldsymbol{\omega})}{dt} + \boldsymbol{F}_\omega \odot {}_B\boldsymbol{\omega}_{ref}\right)\right) \end{cases} \quad (4-3)$$

姿态环算法本身又分为两大部分： $\boldsymbol{q}_{sp}$ 的生成和 ${}_B\boldsymbol{\omega}_{sp}$ 的计算。朴素的设计并不包括 $\boldsymbol{q}_{sp}$ 的生成，直接令 $\boldsymbol{q}_{sp} = \boldsymbol{q}_{ref}$ ；且输出增益 $\tau_x = \tau_y = \tau_z \triangleq \tau$ 。此时 ${}_B\boldsymbol{\omega}_{sp}$ 的计算如下：

$$\begin{cases} \boldsymbol{q}_e \triangleq \boldsymbol{q}_{sp} \otimes \boldsymbol{q}^{-1} = [q_{ew}, q_{ex}, q_{ey}, q_{ez}]^\top \\ {}_B\boldsymbol{\omega}_{sp} = \frac{2}{\tau} \text{sgn}(q_{ew}) \begin{bmatrix} q_{ex} \\ q_{ey} \\ q_{ez} \end{bmatrix}, \quad \text{sgn}(x) \triangleq \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \end{cases} \quad (4-4)$$

其中 $\boldsymbol{q}_e$ 表示的是从当前姿态到期望姿态的旋转， $\tau$ 的意义是将式(4-4)代入式(2-2)所得的一阶系统的时间常数。这样的控制设计使 $\pm \boldsymbol{q}_{sp}$ 在式(2-2)所描述的非线性旋转系统中是全局渐进稳定的平衡点，同时 $\text{sgn}(x)$ 的引入确保了控制器在一个物理姿态下（对应 $\pm \boldsymbol{q}$ ）有且只有一种输出值。

使用式(4-4)中的控制器的问题在于各轴的姿态响应速度不同，所以使用同一个输出增益值 $\tau_x = \tau_y = \tau_z = \tau$ 时，要么机体 z 轴姿态控制超调很大，要么其他两轴调节很慢。为了解决这个问题，需要引入不同的输出增益、同时优先确保机体的 z 轴与期望 z 轴重合，再尽量让机体绕 z 轴旋转，对齐剩余二轴。在这种思路下， $\boldsymbol{q}_{sp}$ 不一定等于 $\boldsymbol{q}_{ref}$ ，从当前姿态到 $\boldsymbol{q}_{sp}$ 可以被分为：一、使 z 轴与期望 z 轴对齐的最短旋转，二、绕 z 轴的旋转，旋转的角度由 $p_z$ 作为缩放因子。式(4-4)这时被重写成：

$${}_B\boldsymbol{\omega}_{sp} = 2 \text{sgn}(q_{ew}) \begin{bmatrix} \frac{1}{\tau_x} \\ \frac{1}{\tau_y} \\ \frac{1}{\tau_z p_z} \end{bmatrix} \odot \begin{bmatrix} q_{ex} \\ q_{ey} \\ q_{ez} \end{bmatrix} \quad (4-5)$$

通常式中参数满足如下关系：

$$\frac{1}{\tau_x} = \frac{1}{\tau_y} = \frac{1}{\tau_z p_z} > \frac{1}{\tau_z} \quad (4-6)$$

记使 z 轴与期望 z 轴对齐的最短旋转为  $\mathbf{q}_{e,red}$ , 此处借用下图描述这样的旋转:

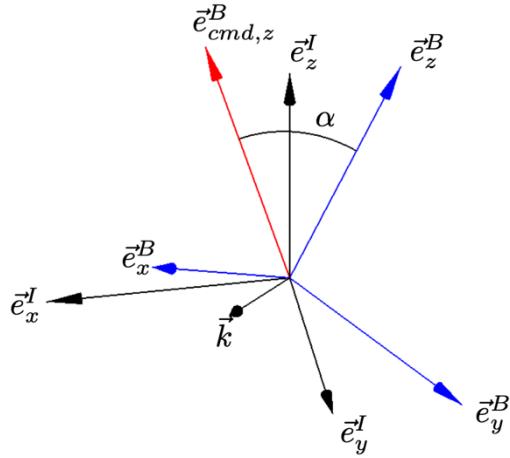


图 4-3 使 z 轴与期望 z 轴对齐的最短旋转<sup>[17]</sup>

图 4-3 中蓝色为当前机体坐标轴在世界坐标系中所对应的单位向量, 红色向量  $\vec{e}_{cmd,z}^B$  是期望的机体 z 轴朝向,  $\alpha$  为具体旋转的角度。 $\vec{k}$  为旋转轴, 因为是最短旋转, 所以  $\vec{k} \perp \vec{e}_{cmd,z}^B$  且  $\vec{k} \perp \vec{e}_z^B$ , 即旋转过程中机体 z 轴上的角速度为 0。由向量点乘和叉乘的定义可知:

$$\begin{cases} \alpha = \cos^{-1}(\vec{e}_z^B \cdot \vec{e}_{cmd,z}^B) \\ \vec{k} = \frac{\vec{e}_z^B \times \vec{e}_{cmd,z}^B}{\|\vec{e}_z^B \times \vec{e}_{cmd,z}^B\|} \end{cases} \quad (4-7)$$

于是可计算  $\mathbf{q}_{e,red}$  以及仅对齐 z 轴的姿态设定点  $\mathbf{q}_{sp,red}$ :

$$\mathbf{q}_{sp,red} = \mathbf{q}_{e,red} \otimes \mathbf{q} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \vec{k} \sin\left(\frac{\alpha}{2}\right) \end{bmatrix} \otimes \mathbf{q} \quad (4-8)$$

记从  $\mathbf{q}_{sp,red}$  到  $\mathbf{q}_{ref}$  的绕机体 z 轴 (此时为  $\vec{e}_{cmd,z}^B$ ) 的旋转为  $\mathbf{q}_{e,full}$ , 对应的旋转角度为  $\alpha_z$ , 而从  $\mathbf{q}_{sp,red}$  到  $\mathbf{q}_{sp}$  的绕 z 轴的旋转为  $\mathbf{q}_{e,p}$ , 对应的旋转角度为  $\alpha_z p_z$ , 那么直接可以写出下式:

$$\mathbf{q}_{e,full} = \begin{bmatrix} \cos\left(\frac{\alpha_z}{2}\right) \\ \vec{e}_{cmd,z}^B \sin\left(\frac{\alpha_z}{2}\right) \end{bmatrix}, \quad \mathbf{q}_{e,p} = \begin{bmatrix} \cos\left(\frac{\alpha_z p_z}{2}\right) \\ \vec{e}_{cmd,z}^B \sin\left(\frac{\alpha_z p_z}{2}\right) \end{bmatrix} \quad (4-9)$$

用  $\mathbf{q}_{ref}$  和  $\mathbf{q}_{sp,red}$  可计算  $\mathbf{q}_{e,full}$ :  $\mathbf{q}_{e,full} = \mathbf{q}_{ref} \otimes \mathbf{q}_{sp,red}^{-1}$ , 自然可以得到  $\alpha_z$  然后计算

得到  $\mathbf{q}_{e,p}$ 。考察  $\mathbf{q}_{sp} = \mathbf{q}_{e,p} \otimes \mathbf{q}_{sp,red} = \mathbf{q}_{e,p} \otimes \mathbf{q}_{e,red} \otimes \mathbf{q}$ ,  $p_z = 1$  时,  $\mathbf{q}_{sp} = \mathbf{q}_{ref}$ ;  $p_z = 0$  时,  $\mathbf{q}_{sp} = \mathbf{q}_{sp,red}$ ; 剩下情况  $\mathbf{q}_{sp}$  为姿态  $\mathbf{q}_{ref}$  与  $\mathbf{q}_{sp,red}$  之间的某个值。

在这样的姿态环设计下, 控制器仅仅会考虑  $\alpha_z p_z$  这么多的 z 轴偏差, 也就限制了  ${}_B\boldsymbol{\omega}_{sp}$  的 z 轴分量, 并使各轴时间常数参数解耦, 降低了调参难度。尽管控制器没有以  $\mathbf{q}_{ref}$  设定为目标点, 但是随着  $t \rightarrow \infty$ , 可以保证  $\mathbf{q} \rightarrow \mathbf{q}_{ref}$ 。

姿态环的设计均来自参考文献[17], 但对文中有误的结论在式(4-8)中进行了修正。文献[17]中的控制对象为四旋翼欠驱动无人机, 但是姿态动态模型不失一般性, 均用式(2-2)描述, 故控制律可直接用于 OmniHex 上。

文中设计优先级, 是基于机体 z 轴的旋转动态响应慢, 在四旋翼上这么考虑是合理的, 因为 z 轴方向的旋转力矩来源于桨叶与空气相互作用产生的力矩; 在 OmniHex 上如此考虑同样是合理的, 因为 z 轴上旋转力矩来源于桨叶与空气的相互作用以及机臂的倾转, 后者同样是较慢的, 甚至因为死区和装配的原因是有滞后和误差的。稳定性证明和鲁棒性分析不在本课题的研究范围内, 文献[17]提供了详细的推导和证明。

#### 4.2.3 合力与合力矩的计算

利用机体坐标系下的牛顿-欧拉公式, 引入力与力矩意义下的前馈, 得到执行器需要产生的合力和合力矩:

$$\begin{bmatrix} {}_B\mathbf{f}_a \\ {}_B\boldsymbol{\tau}_a \end{bmatrix} = \begin{bmatrix} m(\text{rotate}({}_W\dot{\mathbf{v}}_{sp} - \mathbf{g}, \mathbf{q}^{-1}) + {}_B\boldsymbol{\omega} \times \text{rotate}({}_W\mathbf{v}, \mathbf{q}^{-1})) \\ \mathbf{J} \cdot {}_B\dot{\boldsymbol{\omega}}_{sp} + {}_B\mathbf{d}_{com} \times {}_B\mathbf{f}_a + {}_B\boldsymbol{\omega} \times (\mathbf{J} \cdot {}_B\boldsymbol{\omega}) \end{bmatrix} \quad (4-10)$$

PID 控制方案中 wrench 的计算基于名义模型, 必然会受到模型误差的影响。模型误差最终反映在被控量的偏差上, 可由控制器修正。

### 4.3 控制分配

控制分配解决的是如何从 wrench 得到执行器单元指令, 如电机转速、机臂倾转角度的问题。本小节中用到的新符号定义如下:

表 4-2 符号的意义和说明 (补充 4)

| 符号         | 维度           | 意义                      |
|------------|--------------|-------------------------|
| $\alpha_i$ | $1 \times 1$ | $i$ 号机臂倾转角度 (rad)       |
| $\Omega_i$ | $1 \times 1$ | $i$ 号动力电机/旋翼的转速 (rad/s) |
| $l$        | $1 \times 1$ | 机臂长度 (m)                |
| $\mu, k$   | $1 \times 1$ | 力和力矩系数                  |

表 4-2 (续表)

| 符号            | 维度           | 意义                     |
|---------------|--------------|------------------------|
| $f_i, \tau_i$ | $1 \times 1$ | 电机-旋翼单元产生的力和力矩 (N, Nm) |
| $\eta$        | $1 \times 1$ | 电机-旋翼单元力效 (g/W)        |

### 4.3.1 执行器建模

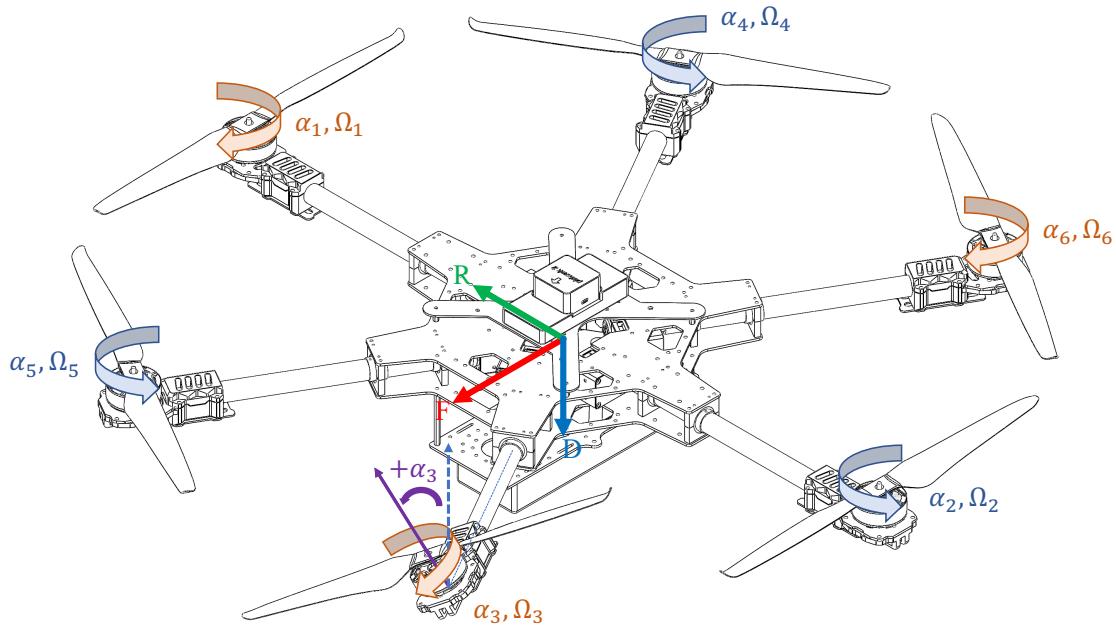


图 4-4 OmniHex 的执行器配置

OmniHex 上电机编号如图 4-4 所示，与 PX4 中欠驱动型六旋翼无人机的设置相同。在图 4-4 所示视角下，1、3、6 号电机顺时针转动，2、4、5 号逆时针转动。机臂倾转角度的正向由右手法则决定：右手拇指由中心沿轴指向机体外侧，剩余手指方向为机臂倾转角度的正方向。

本课题中忽略机臂倾转角度的动态性能，因为在 MPC 设计中可以通过限制 wrench 的变化率以限制机臂倾转的速度  $\dot{\alpha}_i$ ，于是在执行器建模时可以假设  $\alpha_i$  可以瞬间达到期望值，文献[3]和文献[4]对此有更加详细的说明。同时本课题假设电机-旋翼单元产生的陀螺效应可以被忽略。

电机-旋翼单元产生的力和力矩满足如下关系：

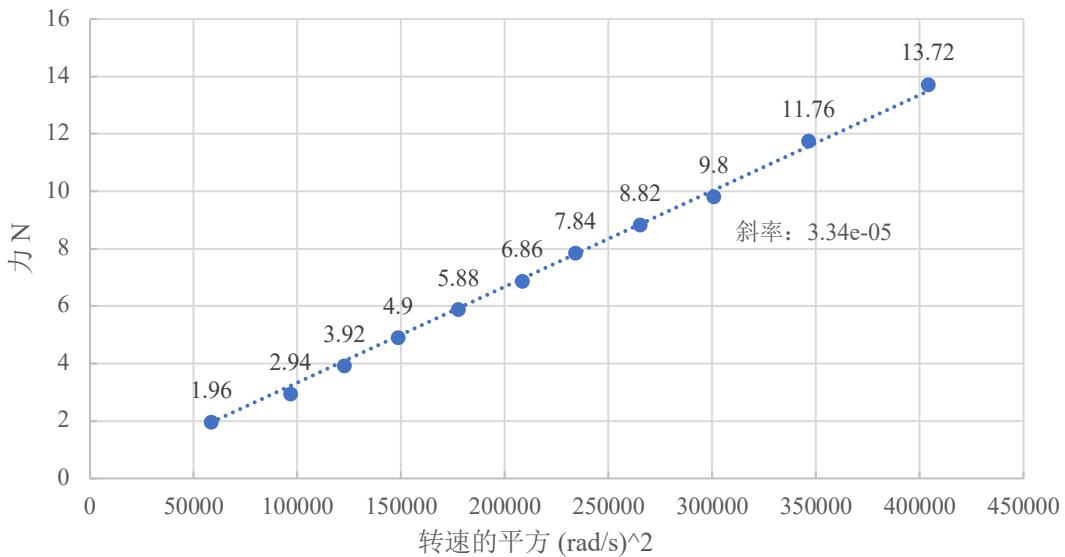
$$\begin{cases} f_i = \mu \Omega_i^2 \\ \tau_i = k \Omega_i^2 \end{cases} \quad (4-11)$$

力的关系可以由电机或旋翼的产品文档得到验证。以 SunnySky 提供的 V3508 KV580 电机在 14.8V 电压下搭配 EOLO1350 桨叶测得的数据为例：

表 4-3 SunnySky V3508 KV580 14.8V EOLO1350 力效

| 转速 (RPM) | 拉力 (g) | 力效 (g/W) |
|----------|--------|----------|
| 2307     | 200    | 15.02    |
| 2969     | 300    | 13.51    |
| 3344     | 400    | 11.75    |
| 3680     | 500    | 10.90    |
| 4024     | 600    | 10.14    |
| 4359     | 700    | 9.27     |
| 4618     | 800    | 8.72     |
| 4918     | 900    | 8.33     |
| 5236     | 1000   | 7.77     |
| 5619     | 1200   | 7.05     |
| 6069     | 1430   | 6.40     |

将转速和拉力单位转换为 rad/s 和 N 并进行线性拟合得到下图：

图 4-5 电机-旋翼力系数 $\mu$ 的拟合估计

由图 4-5 可知  $f_i = \mu \Omega_i^2$  的关系基本符合实际产品测试结果，且此例中  $\mu = 3.34 \times 10^{-5}$ 。类似地，力矩关系  $\tau_i = k \Omega_i^2$  往往能由旋翼生产厂商提供的文档得到验证。在无法获得厂商提供的测试数据时，可以根据文献[20]提供的动力系统建模章节的结论，用旋翼和空气动力学参数进行估计。

在图 4-4 所示的执行器配置下，进行力和力矩的合成，可以得到  ${}_B f_a$ 、 ${}_B \tau_a$  与  $\alpha_i$ 、 $\Omega_i$  之间的关系。由于本文坐标系设置和对执行器的建模与文献[18]相同，得到的结论一致。记  $s_i = \sin(\alpha_i)$ ， $c_i = \cos(\alpha_i)$ ，则关系可以描述为下式：

$$\mathbf{M} \triangleq \begin{bmatrix} -\mu s_1 & \mu s_2 & \frac{\mu}{2} s_3 & -\frac{\mu}{2} s_4 & -\frac{\mu}{2} s_5 & \frac{\mu}{2} s_6 \\ 0 & 0 & \frac{\sqrt{3}\mu}{2} s_3 & -\frac{\sqrt{3}\mu}{2} s_4 & \frac{\sqrt{3}\mu}{2} s_5 & -\frac{\sqrt{3}\mu}{2} s_6 \\ -\underline{u} c_1 & -\mu c_2 & -\mu c_3 & -\mu c_4 & -\mu c_5 & -\mu c_6 \\ -\mu l c_1 - k s_1 & u l c_2 - k s_2 & \frac{\mu l}{2} c_3 + \frac{k}{2} s_3 & -\frac{\mu l}{2} c_4 + \frac{k}{2} s_4 & -\frac{\mu l}{2} c_5 + \frac{k}{2} s_5 & \frac{\mu l}{2} c_6 + \frac{k}{2} s_6 \\ 0 & 0 & \frac{\sqrt{3}\mu l}{2} c_3 + \frac{\sqrt{3}k}{2} s_3 & -\frac{\sqrt{3}\mu l}{2} c_4 + \frac{\sqrt{3}k}{2} s_4 & \frac{\sqrt{3}\mu l}{2} c_5 - \frac{\sqrt{3}k}{2} s_5 & -\frac{\sqrt{3}\mu l}{2} c_6 - \frac{\sqrt{3}k}{2} s_6 \\ u l s_1 - k c_1 & u l s_2 + k c_2 & u l s_3 - k c_3 & u l s_4 + k c_4 & u l s_5 + k c_5 & u l s_6 - k c_6 \end{bmatrix}$$

$$\begin{bmatrix} {}^B\mathbf{f}_a \\ {}^B\boldsymbol{\tau}_a \end{bmatrix} = \mathbf{M}[\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2, \Omega_5^2, \Omega_6^2]^\top \quad (4-12)$$

于是执行器建模部分的最终结果是得到了 wrench 与电机转速和机臂倾转角度的非线性函数。

### 4.3.2 效率最优控制分配

通过简单恒等变换，将  $\mathbf{M}$  中的变量去除，得到一个新的矩阵  $\mathbf{M}_c$  和包括了转角和转速的指令向量  $\mathbf{U}$ 。

$$\begin{bmatrix} {}^B\mathbf{f}_a \\ {}^B\boldsymbol{\tau}_a \end{bmatrix} = \mathbf{M}_c \mathbf{U} = \mathbf{M}_c \begin{bmatrix} \Omega_1^2 s_1 \\ \Omega_1^2 c_1 \\ \vdots \\ \Omega_6^2 s_6 \\ \Omega_6^2 c_6 \end{bmatrix} \quad (4-13)$$

变换的本质是重构了基底向量，使  $\mathbf{M}_c$  为常值矩阵，于是上式为线性方程组。

因为  $\mathbf{M}_c$  是 6 行 12 列的矩阵，线性方程组有无数组解。效率最优分配的核心是使用  $\mathbf{M}_c$  的伪逆  $\mathbf{M}_c^\dagger$  求得  $\mathbf{U}$ ：

$$\mathbf{U} = \mathbf{M}_c^\dagger \begin{bmatrix} {}^B\mathbf{f}_a \\ {}^B\boldsymbol{\tau}_a \end{bmatrix} \quad (4-14)$$

这样求得的解使：

$$\|\mathbf{U}\| = \sqrt{\sum_{i=0}^6 \Omega_i^4} \quad (4-15)$$

最小，而  $\|\mathbf{U}\|$  和所需的能量有正相关的关系，故这种方式被称为效率最优的控制分配。从一对  $(\Omega_i^2 s_i, \Omega_i^2 c_i)$  获取  $\Omega_i$  和  $\alpha_i$  的具体算法是：

$$\Omega_i = \sqrt[4]{(\Omega_i^2 s_i)^2 + (\Omega_i^2 c_i)^2}, \quad \alpha_i = \tan^{-1} \left( \frac{\Omega_i^2 s_i}{\Omega_i^2 c_i} \right) \quad (4-16)$$

### 4.3.3 奇异点姿态处理

考虑式(4-12)所示的线性方程组，当机臂倾转角度不同时， $\mathbf{M}$ 的条件数不同，意味着在不同倾转角配置下，电机转速的分配结果对 wrench 设定点的灵敏度不同。在前述分配策略下，wrench 中的力设定点垂直或处于机体 XY 平面内时所对应的机臂倾转角度会使  $\mathbf{M}$  的条件数非常大，即微小的 wrench 设定点变化会使电机转速产生较为剧烈的变化，这些情况在文献[2]中被称为降秩奇异点 (rank-reduction singularities)。文献[2]同时也给出了修改  $\alpha_i$  的算法，以减小  $\mathbf{M}$  在降秩奇异点处的条件数。但考虑到动力电机的响应较为迅速，故采取措施处理  $\mathbf{M}$  的条件数收益并不明显。

另一种奇异点被称为运动学奇异点 (kinematic singularities<sup>[2]</sup>)。出现在机体合力与某机臂轴重合时，此时此机臂和其对面机臂的  $(\Omega_i^2 s_i, \Omega_i^2 c_i) = (0,0)$ 。一种处理方法是使用带阻尼的伪逆矩阵<sup>[21]</sup>，但是此方法对计算力要求较高，故文献[2]提出了一种在奇异点附近限制机臂倾转速率的解决方案。实际控制分配算法在离散时段中运行，记  $\delta\alpha$  为两次运行时分配倾转角度之差， $\delta\alpha_{new}$  为进行奇异点处理后的角度差， $\varphi_d$  与  $\varphi_0$  分别是两个临界点参数，用于衡量合力向量与机臂向量的夹角  $\varphi$  与奇异点的靠近程度，直观的描述可见下图：

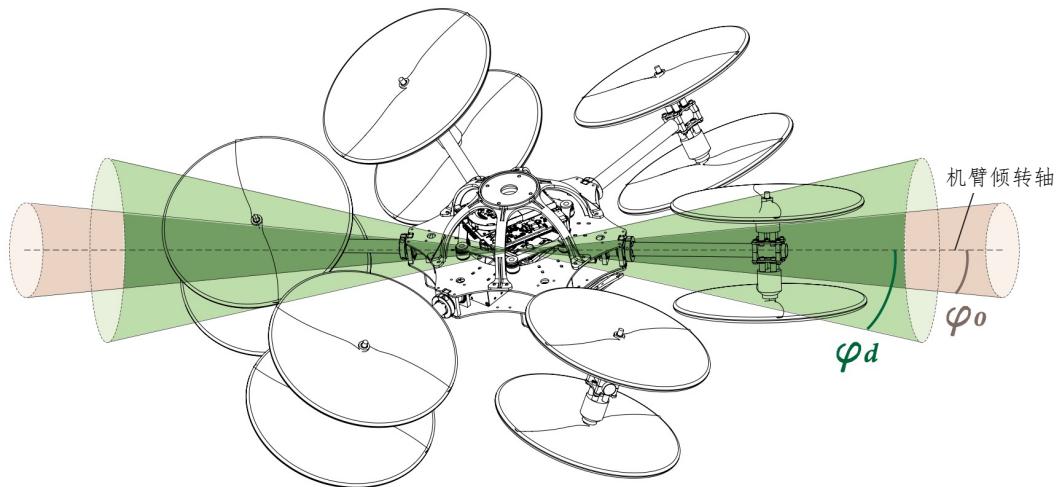


图 4-6 运动学奇异点附近区域<sup>[2]</sup>

具体算法是：

$$\delta\alpha_{new} = \delta\alpha(1 - k)$$

$$k = \begin{cases} 0, & \varphi_d < \varphi \\ \left(1 - \frac{\varphi - \varphi_0}{\varphi_d - \varphi_0}\right), & \varphi_0 < \varphi < \varphi_d \\ 1, & \varphi_0 > \varphi \end{cases} \quad (4-17)$$

算法实现时  $\varphi$  可由向量点乘和反余弦函数得到。上述阻尼算法只对接近奇异点的两臂生效。

在运动学奇异点附近，对应电机-旋翼的转速接近 0，基本不会输出力和力矩，可利用这一性质规划机臂的倾转角度，使机臂回到倾转零位，一定程度上解决线缆缠绕的问题。

#### 4.4 本章小结

本章叙述了带前馈的 PID 备份控制器以及底层的控制分配算法。PID 控制器完全运行于 PX4 微型控制器上，既可以作为 MPC 的备份控制，也可以作为单独的完整的飞行控制器。在前馈通路的存在下，此控制器既可以由遥控指令驱动，完成基本的手动操作飞行，也可以对一段 6 自由度的轨迹进行追踪。控制分配器也运行于 PX4 微型控制器上，根据不同的飞行模式，选择以 MPC 的 wrench 指令或是 PID 控制器的 wrench 指令作为待分配值，是 MPC 和 PID 控制器共同使用的下游组件。

## 第 5 章 实验与分析

### 5.1 实验设置

#### 5.1.1 OmniHex 系统架构

下图从整体上展示了系统架构。一个圆角矩形框代表一个模块：订阅话题（Topic），对消息（Message）进行处理并把处理结果作为消息发布于某些话题上的功能单元。ROS1 和 ROS2 模块运行在 NUC11i7PAH 机载电脑上；PX4FMU 和 PX4IO 的实体为 PX4 飞控产品的两片 MCU，集成于飞控 PCB 上。

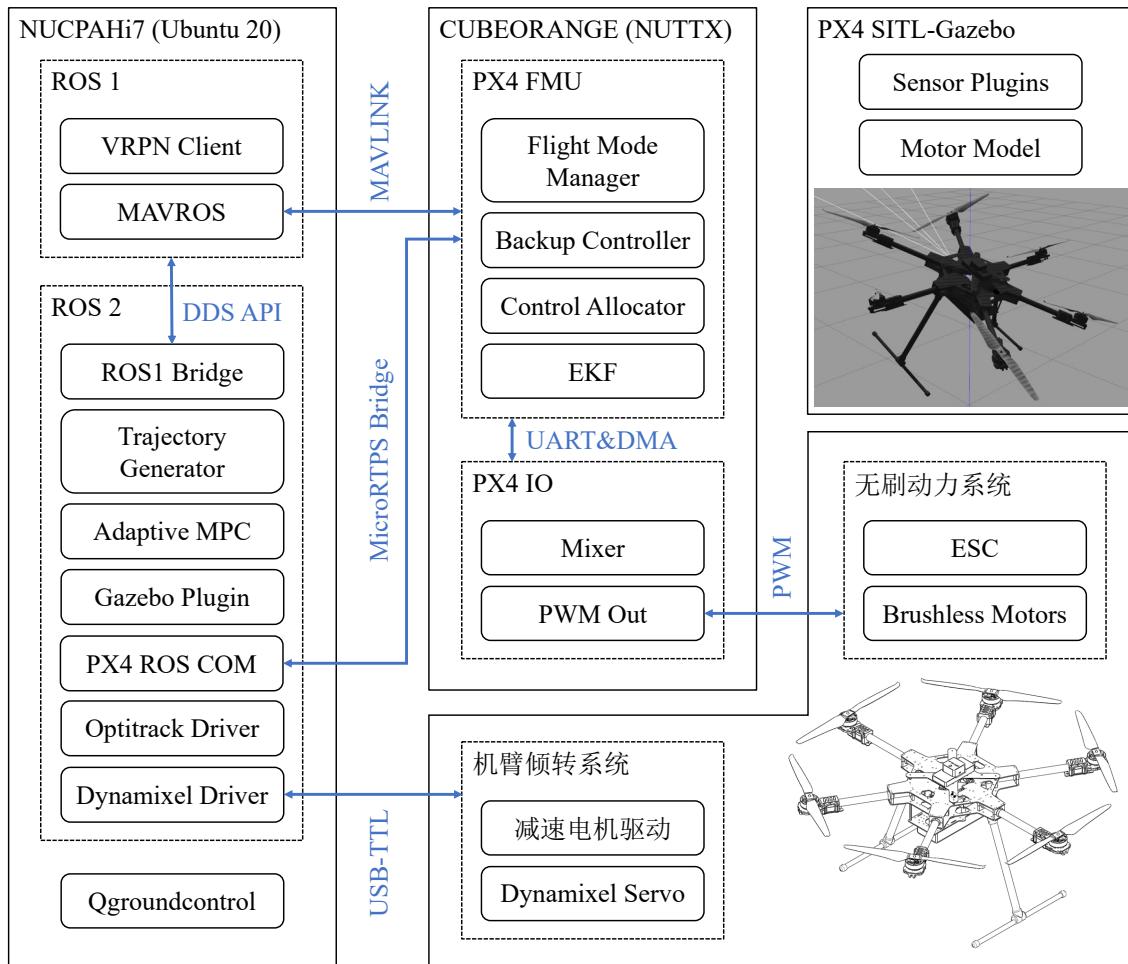


图 5-1 系统架构示意图

ROS1 环境下的模块是 VRPN Client 和 MAVROS。VRPN Client 用于接受 UDP 协议下动作捕捉数据并将其发布为 ROS1 话题中的数据。MAVROS 用于在 ROS1

下联合 VRPN Client 对飞行数据进行可视化，它和 Qgroundcontrol 地面站软件均通过 MAVLINK 协议与 PX4 进行通信。

ROS1 与 ROS2 的数据互通依靠 ROS2 官方提供的包 ROS1 Bridge，这样 ROS2 下的 Optitrack Driver 就能获取机器人的位姿数据，然后把数据转换为 PX4 系统中所定义的格式然后发布，这些数据随后通过 MicroRTPS Bridge 传输到 PX4 中成为 uORB 消息，并被飞控系统使用：EKF 模块接收动捕位姿数据，进行数据融合，产生最终的状态估计。

关键数据，比如执行器指令、动作捕捉位姿信息等，在系统中的通讯由 MicroRTPS Bridge 处理，主要是考虑到传输带宽和延迟性能；而 MAVLINK 只作为监控数据或者低频率数据的通信通道。

运行  $\mathcal{L}_1$ -MPC 的模块是 Adaptive MPC。MPC 和自适应在同一个循环中先后运行，频率为 100Hz。OCP 设置中设置时域窗口为 1 秒，时域窗口离散化等分数  $N = 20$ 。在机载电脑上 OCP 平均求解耗时 1.5ms (C++, Core i7-1165G7)。

PX4 采用飞行管理和指令输出分离的设计，主要逻辑和飞行管理程序运行在 FMU 中，而 IO 处理遥控器的指令输入、控制量的混合、PWM 信号输出等底层任务。FMU 和 IO 之间通过 UART (Universal Asynchronous Receiver/Transmitter) 进行通讯，此外还实现了 DMA (Direct Memory Access) 以降低通信延迟。

Flight Mode Manager 模块根据飞行模式，和遥控器输入，设定控制器的期望值。位姿的期望值由位置控制器处理，得到归一化的力期望值和姿态期望值，进入姿态控制器。姿态控制器不对力期望值作任何处理，但其中姿态设定点与当前姿态相运算后得到角速度期望值。这些数据随后由角速度控制器处理，其中归一化的力期望值被转化为单位为 N 的实际力，并根据机体坐标系下的牛顿-欧拉公式被  ${}_B\omega \times {}_Bv$  补偿；姿态方面，角速度控制器根据角速度期望值和当前值获得力矩期望值，并由角速度和质心项补偿，具体方法见 4.2.3 条。

期望的 wrench 由 Control Allocator 控制分配器解算获得机臂倾转角度和 actuator\_controls\_0 话题中的电机控制量。电机控制量随着 actuator\_controls\_0 被送入 IO 中，由 Mixer 混合后得到实际的 PWM 信号值。PWM 信号值随后由 PWM Out 模块输出给电子调速器 (Electronic Speed Controller, ESC)，控制无刷电机的转速。机臂倾转角度被 ROS2 中的 Dynamixel Driver 舵机驱动接收，后者驱动机臂按照设定点进行倾转。在仿真中，则是 Gazebo 的插件控制仿真环境中的机臂进行倾转。目前在舵机驱动方面调用的是以位置指令为接口的位置-速度-电流三环控制。

PX4 原本实现的算法并没有考虑如 OmniHex 这样的过驱动六旋翼机型，故需要对其代码进行修改，实现以下目标：一、按照牛顿-欧拉公式实现更精确的、带

有物理意义的刚体运动控制；二、为无人机设计一种新的飞行模式，此模式下期望用遥控器指令改变俯仰角；三、实现控制分配算法，将指令输出管线调试通畅；四、在相关的模块中修复由欠驱动转换为过驱动而产生的自由度缺失问题。关键模块的修改见下表。

表 5-1 PX4 关键模块的修改

| 模块                          | 修改                                   | 备注   |
|-----------------------------|--------------------------------------|--|
| Angular Velocity Controller | 在扭矩计算中添加了质心的补偿模型，在合力的计算中添加了角速度和线速度补偿 | Newton-Euler 公式                            |
| Commander                   | 添加了 ACRO 飞行模式                        | 在 ACRO 模式中，本来用于前进后退的摇杆用于抬头和低头              |
| Control Allocator           | 添加了 OmniHex 需要的分配算法，修改了控制量的输出话题      | actuator_controls_0 是 FMU 和 IO 通信时被优先考虑的信息 |
| Flight Mode Manager         | 配合 Commander 实现 ACRO 模式              | 补上缺失的自由度                                   |
| MC Attitude Controller      | 正确处理 ACRO 模式下带来的角速度前馈                | 考虑缺失的自由度                                   |
| MC Position Controller      | 添加了世界坐标系下的合力到机体坐标系下合力的转换步骤           | 因为控制分配要使用机体坐标系下的合力/力矩                      |
| PX4IO                       | 修改 Allocated Mixer 的输入话题             | ESC 校准功能暂未测试                               |

### 5.1.2 参考轨迹的生成

如式(3-1)所示，参考轨迹形式应为：

$$\boldsymbol{x}_{ref} = \begin{bmatrix} \boldsymbol{p}_{ref} \\ \boldsymbol{q}_{ref} \\ w\boldsymbol{v}_{ref} \\ {}_B\boldsymbol{\omega}_{ref} \end{bmatrix} \quad (5-1)$$

但人为设计参考轨迹的时候往往习惯使用更直观的欧拉角表示：先绕机体 z 轴旋转  $\psi$  作为 Yaw 角，随后绕机体 y 轴旋转  $\theta$  作为 Pitch 角，最后绕机体 x 轴旋转  $\phi$  作为 Roll 角。实验中用于测试的参考轨迹生成方法是：一、给定  $\boldsymbol{p}_{ref}$ 、 $\phi$ 、 $\theta$ 、 $\psi$  的解析形式；二、求出它们对时间的导数： $\dot{\boldsymbol{p}}_{ref} = w\boldsymbol{v}_{ref}$ 、 $\dot{\phi}$ 、 $\dot{\theta}$ 、 $\dot{\psi}$ ；三、根据  $\phi$ 、 $\theta$ 、 $\psi$  和  $\dot{\phi}$ 、 $\dot{\theta}$ 、 $\dot{\psi}$  求出  $\boldsymbol{q}_{ref}$  和  ${}_B\boldsymbol{\omega}_{ref}$ 。 $\boldsymbol{q}_{ref}$  可以直接由欧拉角转换得到：

$$\mathbf{q}_{ref} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) \end{bmatrix} \quad (5-2)$$

${}_B\boldsymbol{\omega}_{ref}$ 则和欧拉角和其导数均有关:

$${}_B\boldsymbol{\omega}_{ref} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (5-3)$$

本章后续测试中使用的轨迹以  $T$ 、 $t_1$ 、 $t_2$ 、 $r$ 、 $d_h$ 、 $h$ 、 $k_{roll}$ 、 $k_{pitch}$ 、 $x_0$ 、 $y_0$  为参数, 解析形式如下:

$$\mathbf{p}_{ref,x}(t) = \begin{cases} x_0, & t \in [0, t_1) \\ x_0 - r + r \cos\left(\frac{3\pi}{2t_2}(t - t_1)\right), & t \in [t_1, t_1 + t_2) \\ x_0 - r + \frac{r}{t_1}(t - t_1 - t_2), & t \in [t_1 + t_2, 3t_1 + t_2) \\ x_0 + r + r \sin\left(\frac{3\pi}{2t_2}(t - 3t_1 - t_2)\right), & t \in [3t_1 + t_2, T - t_1) \\ x_0, & t \in [T - t_1, T) \end{cases} \quad (5-4a)$$

$$\mathbf{p}_{ref,y}(t) = \begin{cases} y_0 + \frac{rt}{t_1}, & t \in [0, t_1) \\ y_0 + r + r \sin\left(\frac{3\pi}{2t_2}(t - t_1)\right), & t \in [t_1, t_1 + t_2) \\ y_0, & t \in [t_1 + t_2, 3t_1 + t_2) \\ y_0 - r + r \cos\left(\frac{3\pi}{2t_2}(t - 3t_1 - t_2)\right), & t \in [3t_1 + t_2, T - t_1) \\ y_0 - r + \frac{r(t - T + t_1)}{t_1}, & t \in [T - t_1, T) \end{cases} \quad (5-4b)$$

$$\mathbf{p}_{ref,z}(t) = -h - d_h + d_h \cos\left(\frac{2\pi t}{2t_1 + t_2}\right), \quad t \in [0, T) \quad (5-4c)$$

$$\phi(t) = \begin{cases} 0, & t \in [0, t_1) \\ k_{roll} \left( \frac{\pi}{4} - \frac{\pi}{4} \cos\left(\frac{2\pi}{t_2}(t - t_1)\right) \right), & t \in [t_1, t_1 + t_2) \\ 0, & t \in [t_1 + t_2, 3t_1 + t_2) \\ k_{roll} \left( -\frac{\pi}{4} + \frac{\pi}{4} \cos\left(\frac{2\pi}{t_2}(t - 3t_1 - t_2)\right) \right), & t \in [3t_1 + t_2, T - t_1) \\ 0, & t \in [T - t_1, T) \end{cases} \quad (5-4d)$$

$$\theta(t) = k_{pitch} \dot{\mathbf{p}}_{ref,z} \quad (5-4e)$$

$$\psi(t) = \begin{cases} \frac{\pi}{2}, & t \in [0, t_1) \\ \frac{\pi}{2} + \frac{3\pi}{2t_2}(t - t_1), & t \in [t_1, t_1 + t_2) \\ 0, & t \in [t_1 + t_2, 3t_1 + t_2) \\ -\frac{3\pi}{2t_2}(t - 3t_1 - t_2), & t \in [3t_1 + t_2, T - t_1) \\ \frac{\pi}{2}, & t \in [T - t_1, T) \end{cases} \quad (5-4fYes)$$

在某组参数下可视化为：

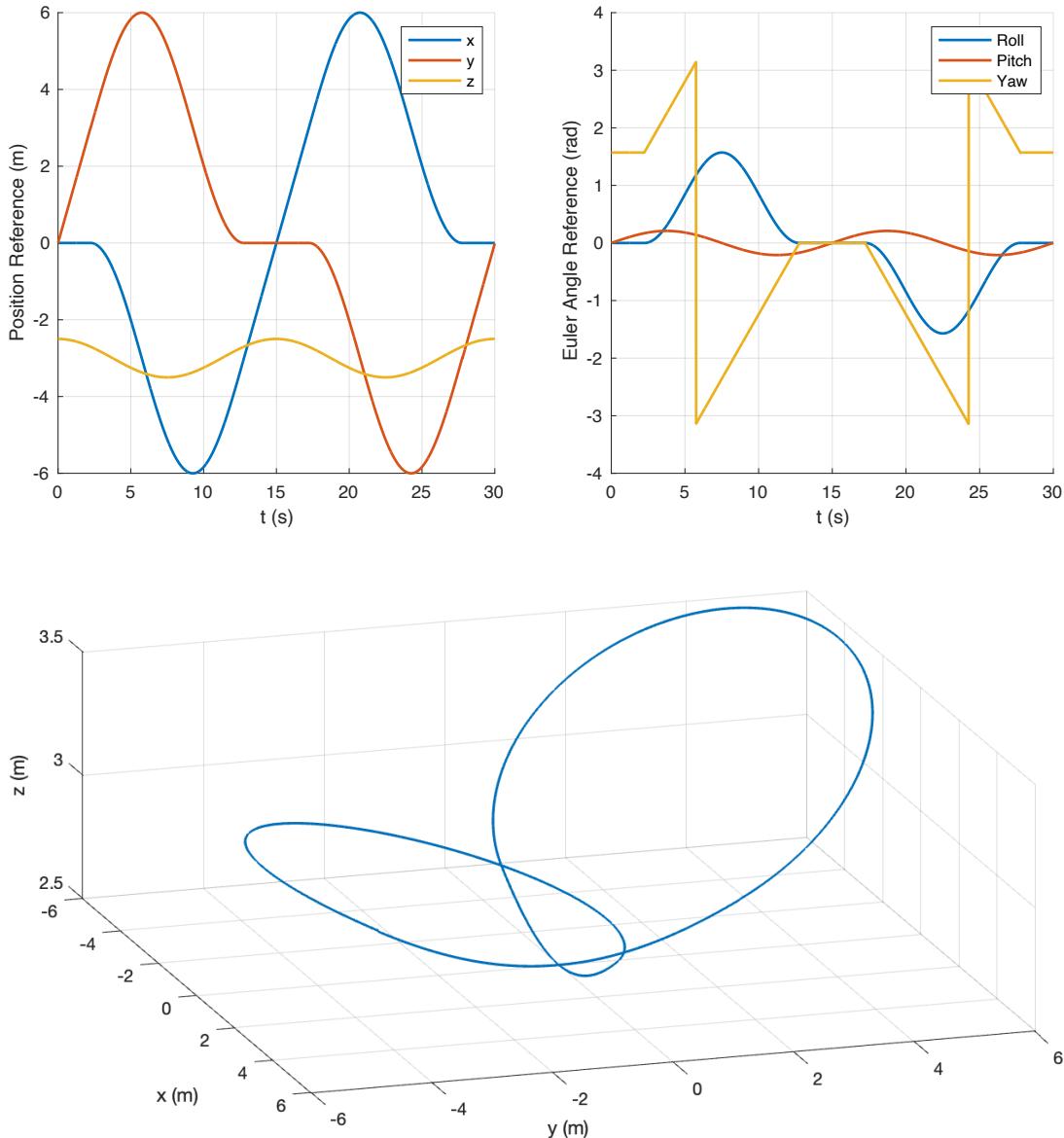


图 5-2 式(5-4)轨迹示意图

$\mathbf{p}_{ref}$  和  $\dot{\mathbf{p}}_{ref}$  均为时间的分段连续函数，而在旋转上因为  $\dot{\theta}(t)$ 、 $\dot{\psi}(t)$  不连续，所以  ${}_B\omega_{ref}$  也不连续，在某些时刻对于控制器来说为阶跃输入：

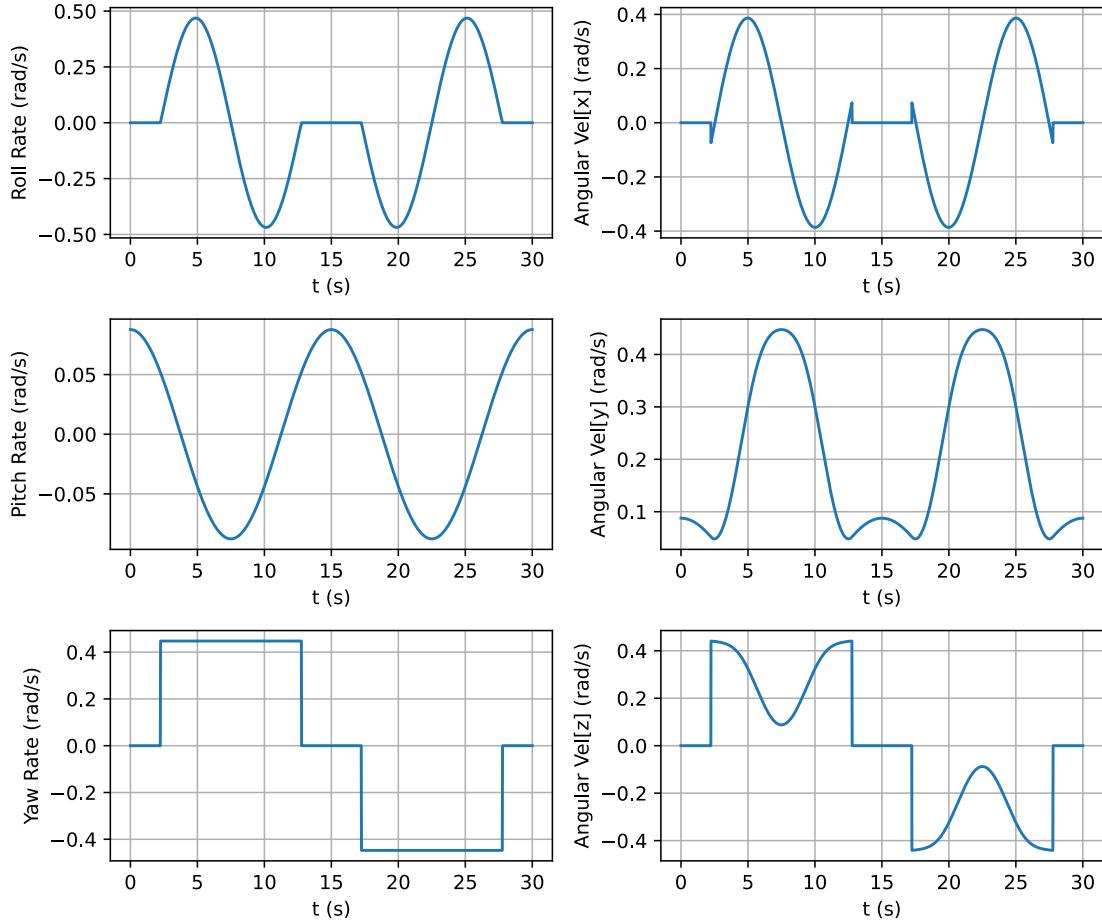


图 5-3 式(5-4)轨迹旋转部分

## 5.2 数值仿真实验

数值仿真实验是用 acados 数值积分器来模拟物理实体，测试 MPC 和自适应效果的实验。实验分八组测试，对比名义模型 MPC (Nominal MPC, NMPC) 和自适应 MPC (Adaptive MPC, AMPC 即  $\mathcal{L}_1$ -MPC) 的轨迹追踪效果，组别设置如下：

表 5-2 数值仿真实验组别设置

| 组别 | 控制器  | 积分器模型                      |
|----|------|----------------------------|
| A  | NMPC | 名义模型                       |
| B  | AMPC | 名义模型                       |
| C  | NMPC | 质量减少 0.5kg, 转动惯量减半, 改变质心位置 |
| D  | AMPC | 质量减少 0.5kg, 转动惯量减半, 改变质心位置 |

表 5-2 (续表)

| 组别 | 控制器  | 积分器模型                            |
|----|------|----------------------------------|
| E  | NMPC | 质量增加 0.5kg, 转动惯量加倍, 改变质心位置       |
| F  | AMPC | 质量增加 0.5kg, 转动惯量加倍, 改变质心位置       |
| G  | NMPC | 质量增加 0.5kg, 转动惯量加倍, 改变质心位置, 施加干扰 |
| H  | AMPC | 质量增加 0.5kg, 转动惯量加倍, 改变质心位置, 施加干扰 |

G 和 H 中施加的干扰的具体形式是改变动力学模型的表达式：将名义模型的动态方程改变为下式：

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{w}^{\boldsymbol{v}} \\ \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ {}_B\boldsymbol{\omega} \end{bmatrix} \\ \boldsymbol{g} + \frac{\text{rotate}(0.95({}_B\boldsymbol{f}_a + \sin({}_B\boldsymbol{f}_a)), \boldsymbol{q})}{m} \\ \boldsymbol{J}^{-1}(0.9({}_B\boldsymbol{\tau}_a + \sin({}_B\boldsymbol{\tau}_a)) - {}_B\boldsymbol{d}_{com} \times (0.95{}_B\boldsymbol{f}_a) - {}_B\boldsymbol{\omega} \times (\boldsymbol{J} \cdot {}_B\boldsymbol{\omega})) \end{bmatrix} \quad (5-5)$$

其中正弦项表示外力干扰，wrench 的系数表示控制分配产生的误差。

每组实验分别让系统追踪参数  $T$  改变 (15, 30, 45, 60, 75) 而其他参数不变的参考轨迹，记录每次实验的位置均方根误差  $p_{RMSE}$ 、四元数均方根误差  $q_{RMSE}$  和优化用时  $t_{opt}$ 。均方根误差的计算如下：

$$\begin{cases} p_{RMSE} = \sqrt{\frac{\sum_{i=0}^N \|\boldsymbol{p}_{err,i}\|^2}{N}} \\ q_{RMSE} = \sqrt{\frac{\sum_{i=0}^N \|\boldsymbol{q}_{err,i}\|^2}{N}} \end{cases} \quad (5-6)$$

下表列举了实验中设置的关键参数：

表 5-3 实验中关键参数的设置

| 参数               | 值                           | 参数                                 | 值                           |
|------------------|-----------------------------|------------------------------------|-----------------------------|
| $\boldsymbol{A}$ | $-5\mathbb{I}_6$            | $\boldsymbol{Q}_e$                 | $\text{diag}(5,20,5,5,0,0)$ |
| LPF              | $\frac{5}{s+5}\mathbb{I}_6$ | $\boldsymbol{R}$                   | $\mathbb{O}_6$              |
| $T_s$            | 0.01s (同 MPC)               | $N$ 、 $T_{ocp}$                    | 20、1s                       |
| $\boldsymbol{Q}$ | $\text{diag}(5,20,5,5,0,0)$ | $m$ 、 $J$ 、 $\boldsymbol{d}_{com}$ | 由 Solidworks 估得             |

得到的结果由图 5-4 至图 5-7 所示。

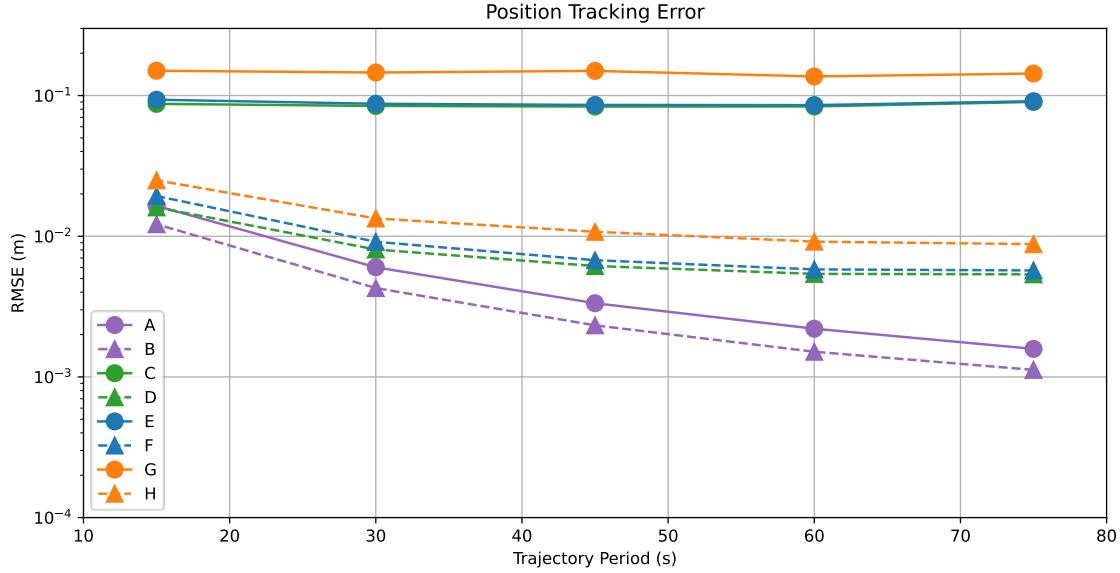


图 5-4 位置追踪误差

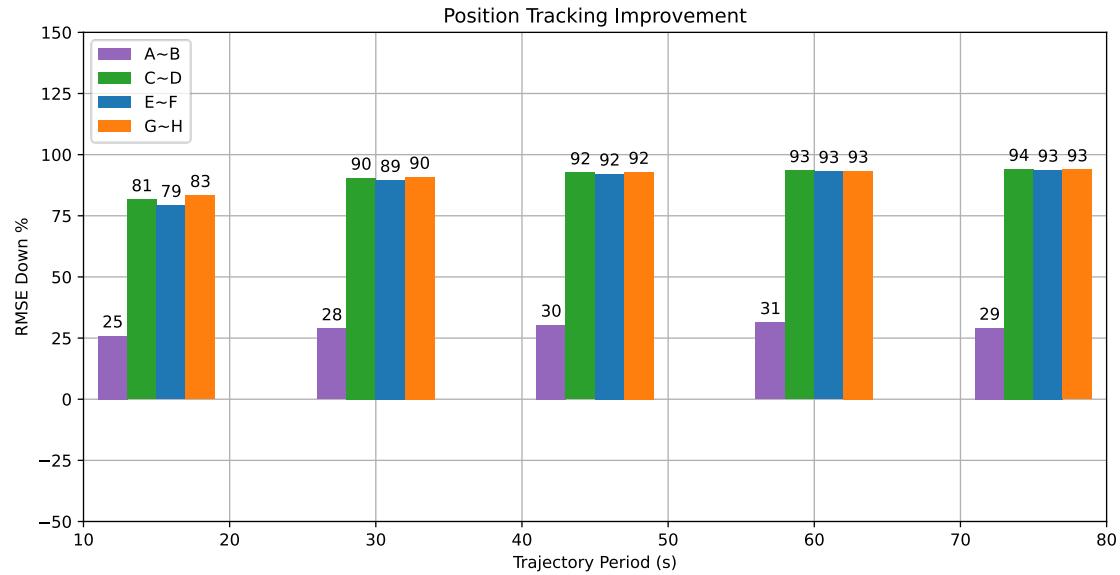


图 5-5 位置追踪误差下降百分比

图 5-4 和图 5-5 展示了位置追踪的效果。数据整体趋势为：一、轨迹周期越小，平均速度越快，则追踪误差越大；二、有自适应补偿的结果均比无自适应补偿的结果要好；三、随着模型误差的增大和外部干扰的引入，追踪误差越大。

除了 A 和 B 组，AMPC 均使追踪误差下降了一个数量级。以误差下降百分比为衡量标准，则可认为轨迹追踪效果有 90% 的提升。而 A 组和 B 组使用的是名义模型，故误差本身很小，尽管提升幅度不如其他几组明显，但可以表明 AMPC 足够灵敏，可以对微小的误差做出补偿。

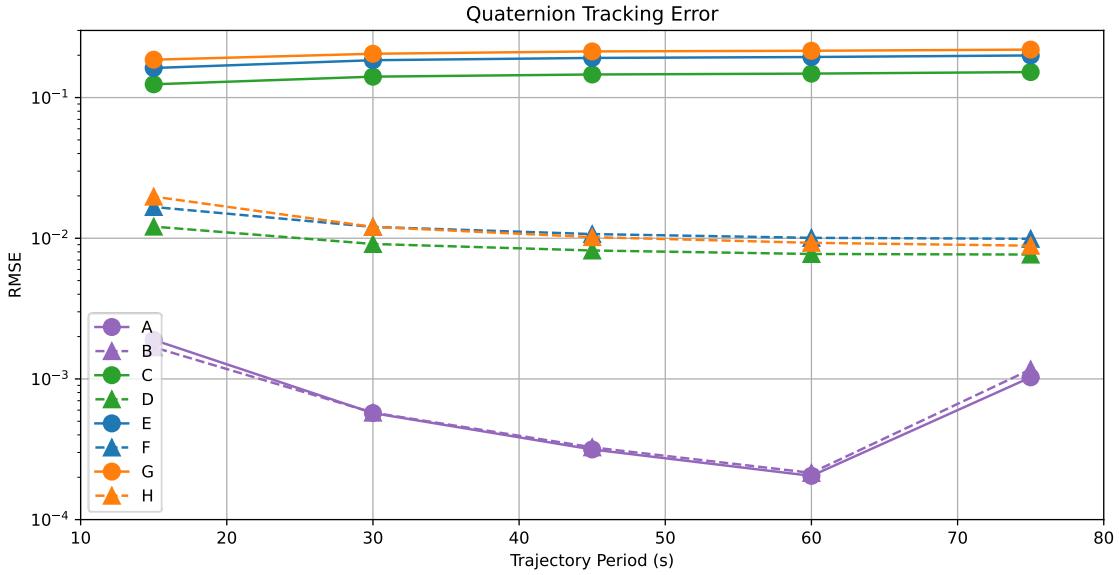


图 5-6 姿态追踪误差

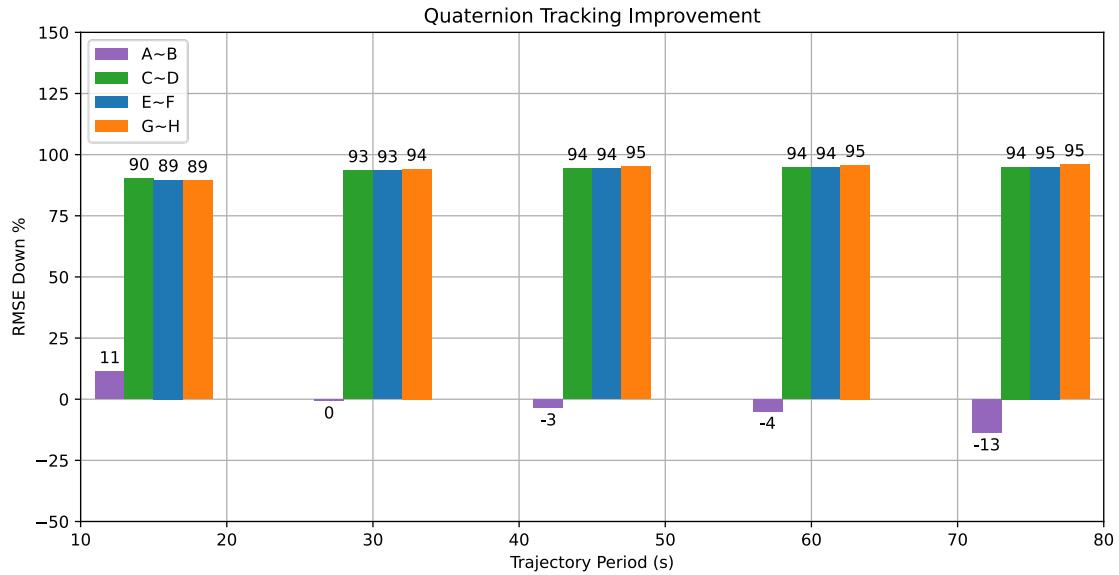


图 5-7 姿态追踪误差下降百分比

图 5-6 和图 5-7 展示了姿态上的轨迹追踪效果，效果与位置追踪类似。AMPC 在引入了模型误差的组别中，均使追踪误差下降了一个数量级，追踪效果有接近 95% 的提升。在完全理想的情况下，AMPC 中的自适应部分虽然引入了更多误差，但是考虑到误差本身的数量级，可以认为这部分的负优化是可以被忽略的；更何况完全理想的情况并不存在，所以并不影响对算法有效性的验证。

下面取 G、H 组在  $T = 45$  参数下实验的结果进一步展示 AMPC 对模型误差的适应能力。

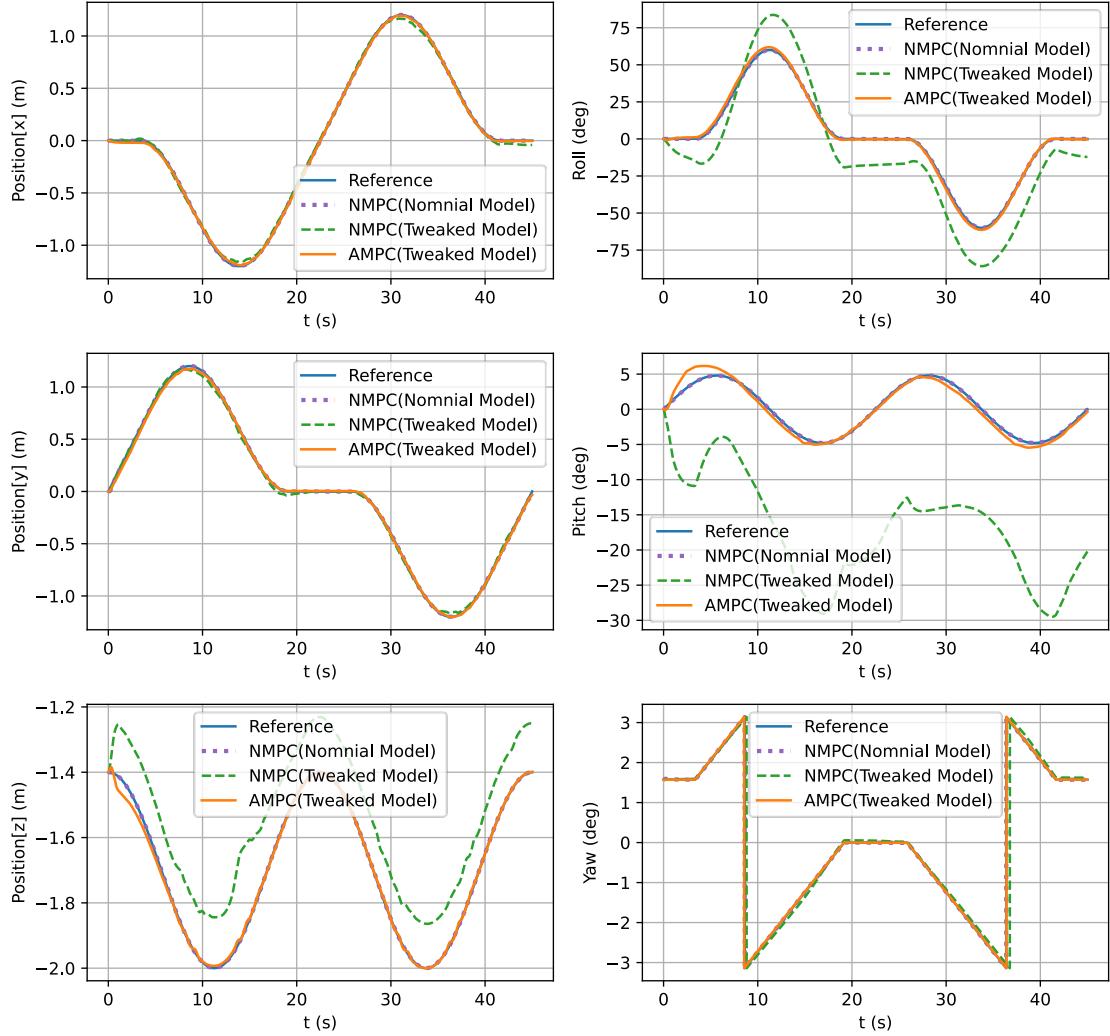


图 5-8 位姿追踪效果对比

位置追踪方面，效果最显著的是对高度的修正：因为 NMPC 并没有积分环节，所以在质量参数  $m$  不准确的时候，实验结果会呈现高度上的稳定偏差，而 AMPC 则能根据状态变化的趋势（图 5-8 中  $t = 0$  附近）迅速对 MPC 的控制量进行补偿，使系统高度向参考值收敛。

姿态追踪方面，由于质心参数误差和外力干扰的引入，NMPC 对 Roll 和 Pitch 的追踪误差已经达到了几乎无法接受的程度，但 AMPC 仍然能较好地处理这部分不确定性。

另外，在状态误差较大时，OCP 求解耗时要更高，而自适应环节可以确保误差不会过大，可认为自适应环节有利于减少 OCP 求解器的计算负担。图 5-9 显示了整个追踪过程中 OCP 的求解用时，从第 10 秒开始，NMPC 的曲线陆续出现了几个明显高于 AMPC 的高峰。

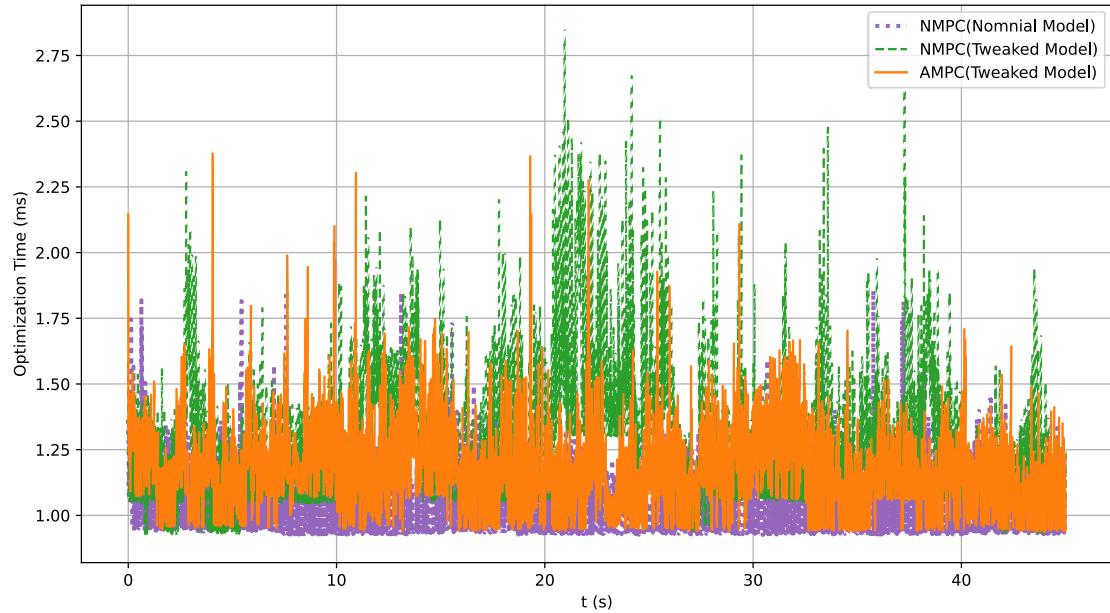


图 5-9 OCP 求解用时 (Python, Ryzen 5900HX)

本文使用的 $\mathcal{L}_1$ 自适应控制本质上属于模型参考自适应控制 (Model Reference Adaptive Control, MRAC)，处于被控对象和上层控制器之间。 $\mathcal{L}_1$ 自适应环节与实际被控对象的整体，对于上层控制器来说，等效于名义模型所描述的对象。下面对控制器输出量的分析，可作为对这种思路有效性的验证。

图 5-10 是实验过程中 wrench 设定点的变化。紫色点线代表是 NMPC 在名义模型上得到的结果，绿色虚线是 G 组的结果，橙色实线是 H 组带补偿的结果。橙色实线与紫色点线几乎重合、意味着对于上层控制 NMPC 来说，控制“由名义模型描述的对象”和“自适应环节串联带误差的实际对象”是等效的，通过相同的控制量使被控对象的状态达到同样的期望值。

图 5-11 展示了无补偿的 MPC 控制输出量和经过补偿后的 MPC 控制输出量分配于电机-旋翼执行器单元上的值。虚线和点线在左右子图中一致，因为均没有开启自适应控制器补偿；实线则因为开启了补偿所以不一致。右图中的实线被整体往上移动了一部分，对应表 5-2 中质量增加 0.5 千克，以及转动惯量加倍的改动；局部的小幅度震荡则对应式(5-5)中对模型施加的外力干扰。

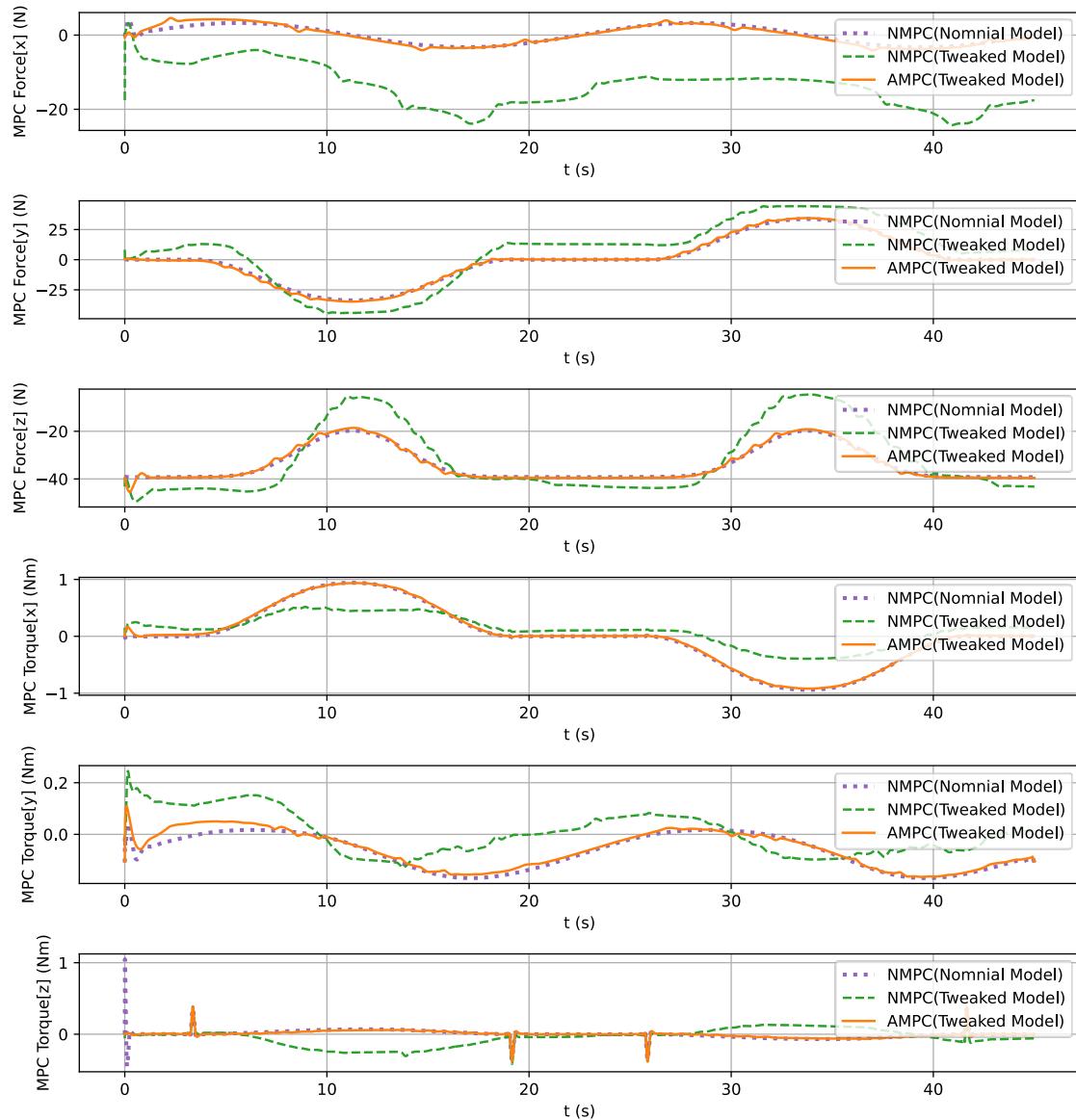


图 5-10 MPC 输出量

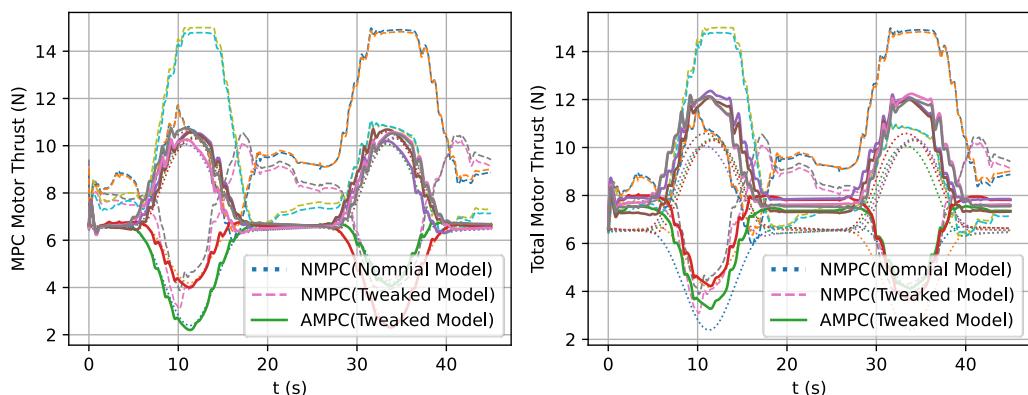


图 5-11 单个电机拉力

图 5-11 还展示了 AMPC 的另一个优点：避免执行器做无用功。图中虚线在 10 到 15 秒以及 30 到 40 秒之间达到饱和，直接原因是式(3-7)中对向量  $\mathbf{h}$  施加了不等式约束所致，本质上则是 NMPC 基于名义模型对状况做出了不符合实际的判断，给出了在理想情况下才合理的输出，输出值在非理想情况下过大则表现为图 5-11 中虚线的饱和。虽然在 MPC 数据下游串联一个补偿环节有可能破坏 MPC 中的约束，但是因为它能使真实系统表现得更像名义系统，使系统状态更接近参考轨迹，所以能避免 MPC 因为追踪误差较大输出饱和或者接近饱和的值，进而避免执行器做无用功。这个性质对于功耗较大且能源有限的空中机器人来说是非常好的。电机-旋翼产生的力之和可以大致反映总能量消耗：把整个过程中每时刻的总拉力加起来，NMPC 结果为 240554，AMPC 为 223208。

但是值得说明的是，“避免执行器做无用功”并不意味着 AMPC 一定比 NMPC 消耗能量更小：考虑仅仅只是轻微低估了质量和转动惯量的情况，此时 NMPC 倾向于输出较小的 wrench 而 AMPC 中的补偿环节则会补上被低估的部分，这时 AMPC 反而消耗的能量更大。从能量角度考虑，AMPC 看似并不理想，但是这部分额外的能量消耗是完全追踪上参考轨迹必要的代价。

### 5.3 Gazebo 仿真实验

类似数值仿真实验，此节将在 Gazebo 仿真环境中对比 PID 控制器、NMPC 和 AMPC 的轨迹追踪效果。

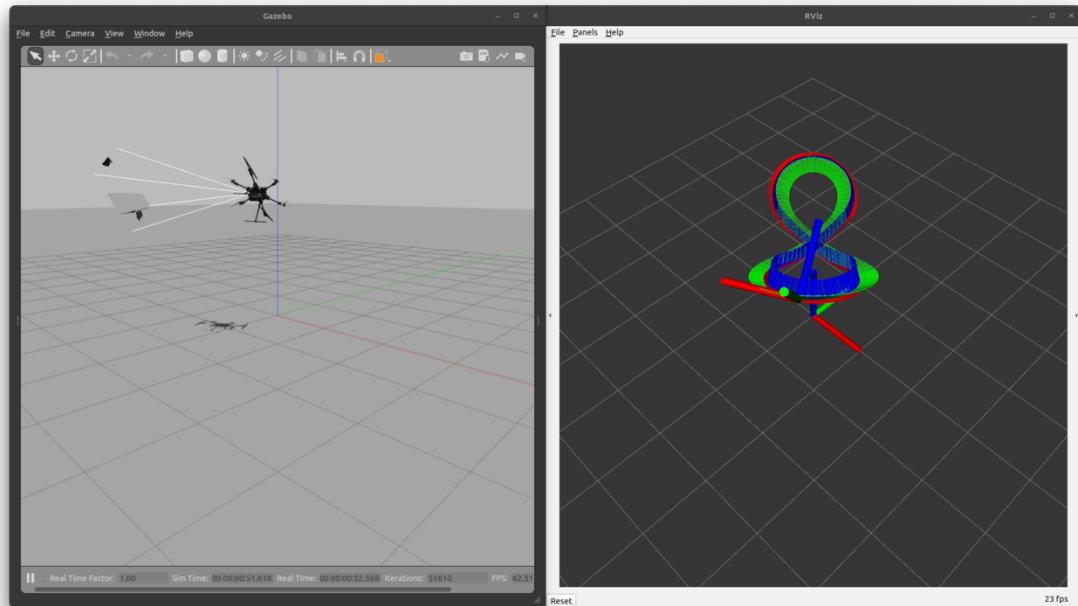


图 5-12 Gazebo 仿真和 RVIZ 界面

使用 PID 控制器进行轨迹追踪实验时，PX4 的飞行模式被设置为 Offboard，以 10Hz 的频率接收机载电脑给出的设定点。设定点包括位置、线速度、欧拉角、机体角速度等信息，与 MPC 的参考轨迹保持一致，但并不包括时间窗口内未来的所有设定点，只有当前时刻的值。

Gazebo 实验中关键参数与表 5-3 相同。本节在环境中引入了陀螺效应，并为了增大控制分配所产生的不确定性，对式(4-16)中  $\Omega_i$  的计算进行了修改，将开四次方运算变为开平方运算：

$$\Omega_i = \sqrt{(\Omega_i^2 s_i)^2 + (\Omega_i^2 c_i)^2}, \quad \alpha_i = \tan^{-1} \left( \frac{\Omega_i^2 s_i}{\Omega_i^2 c_i} \right) \quad (5-7)$$

追踪误差的对比如下图所示。

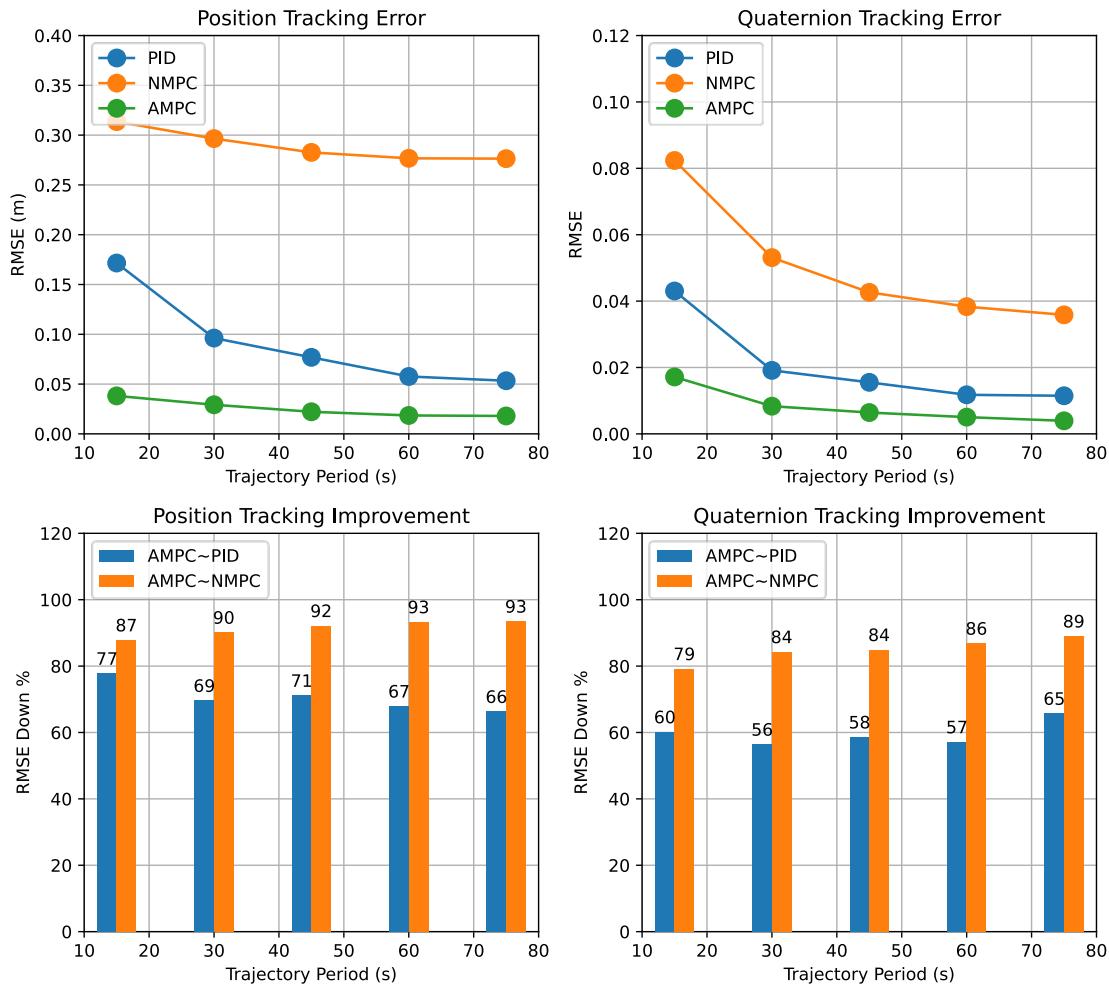


图 5-13 轨迹追踪误差和下降百分比

从图 5-13 可得出的主要结论有：一、基于 PID 控制器轨迹追踪效果总体不错；二、NMPC 因为模型误差的存在而效果甚至不如 PID 控制器；三、AMPC 是三种

算法中效果最好的，追踪效果对比 NMPC 和 PID 均有较大提升；四、追踪效果的整体趋势类似数值仿真结果，轨迹平均速度越大时 RMSE 越大。

PID 控制器的追踪效果可以通过进一步调节相关参数变得更好，笔者所获得的最佳参数在 Gazebo 中的表现与 AMPC 接近，但并没有超越 AMPC；且这组参数在 OmniHex 物理本体上会造成位置曲线大幅度振荡以及姿态曲线的发散，并不具有实际意义。无论 PID 控制器效果如何，图 5-13 依然能够说明 AMPC 能有效地解决模型误差给 NMPC 带来的问题。

下面取轨迹周期为 15 秒和 75 秒的实验组的时域结果进行进一步分析。

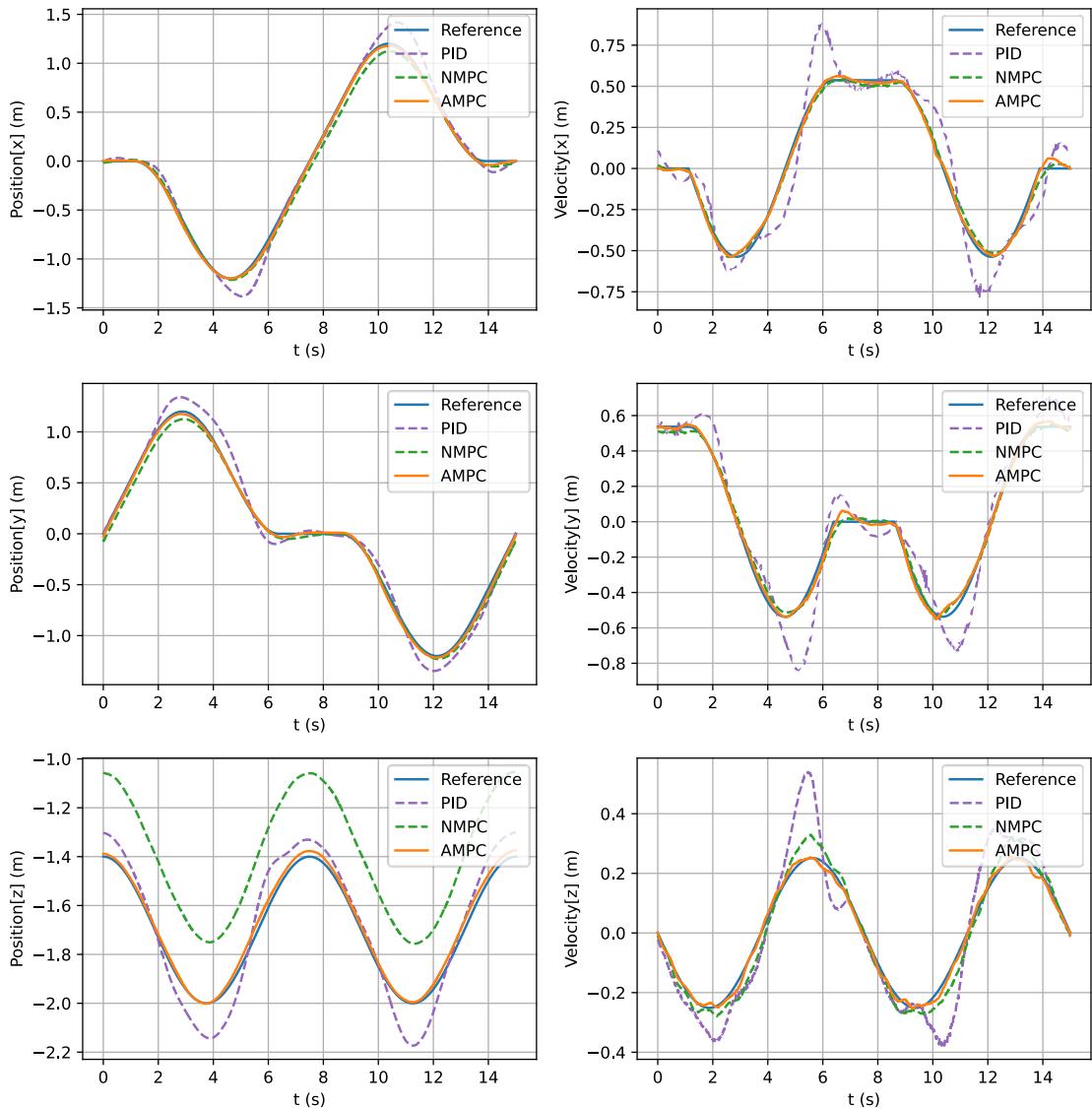


图 5-14 Gazebo 仿真平动追踪效果 ( $T = 15$ )

平动方面：PID 控制器的追踪效果并不理想，在位置和速度变化较为剧烈处

均无法很好追踪参考轨迹。NMPC 的追踪效果整体上优于 PID，但因为高度上有一个恒定的偏差，故反映在图 5-13 中有 0.3m 左右的 RMSE，这些 RMSE 基本由高度误差贡献。AMPC 则是不管在位置还是速度上均能接近完美地对参考轨迹进行跟踪。对比 MPC 和 PID 方法，MPC 得益于预测环节的引入，整体轨迹更加平滑。

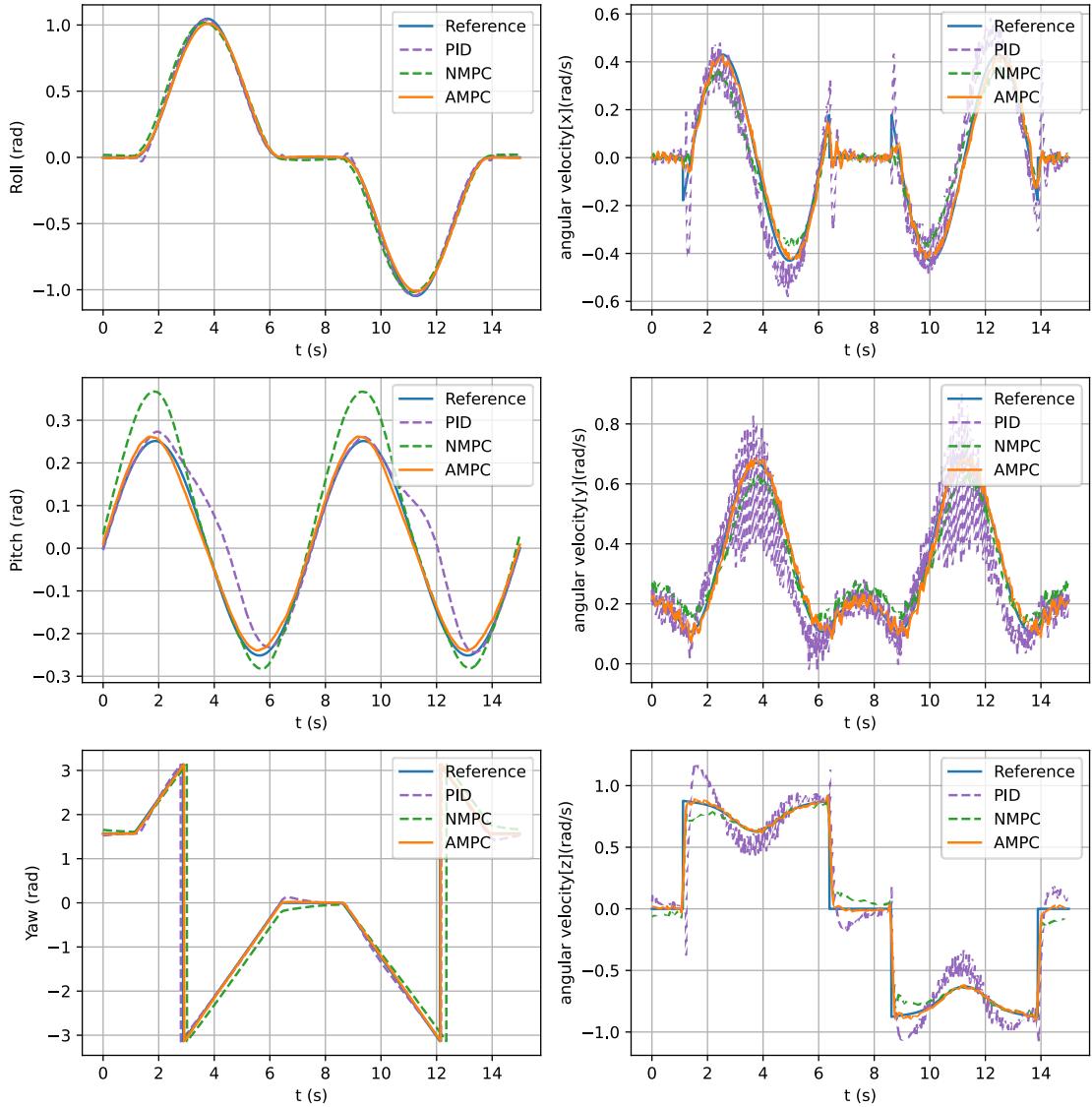


图 5-15 Gazebo 仿真转动追踪效果 ( $T = 15$ )

转动方面：PID 和 NMPC、AMPC 之间的显著差别在于机体角速度上 PID 的超调和高频振荡明显，而 NMPC 和 AMPC 的角速度曲线都较为平滑。纵观全部图像，显然 AMPC 是三种方法中欧拉角和角速度上追踪效果最好、超调和振荡最小、平滑性最好的方法。

在轨迹较为平缓的情况下 ( $T = 75$  时), 效果最好的仍然是 AMPC, 在位置、速度、姿态、角速度方面均基本能完全跟踪上参考轨迹。PID 跟踪效果比  $T = 15$  时更好, 因为此时“预测”环节对轨迹追踪的贡献不如  $T = 15$  时大。值得注意的是, 尽管 Pitch 的参考轨迹幅度和频率都变低了, 但是 NMPC 在此维度上的相对误差变大了, 甚至某些时刻达到了 100%, 说明名义模型中的质心位置参数较为不准确。

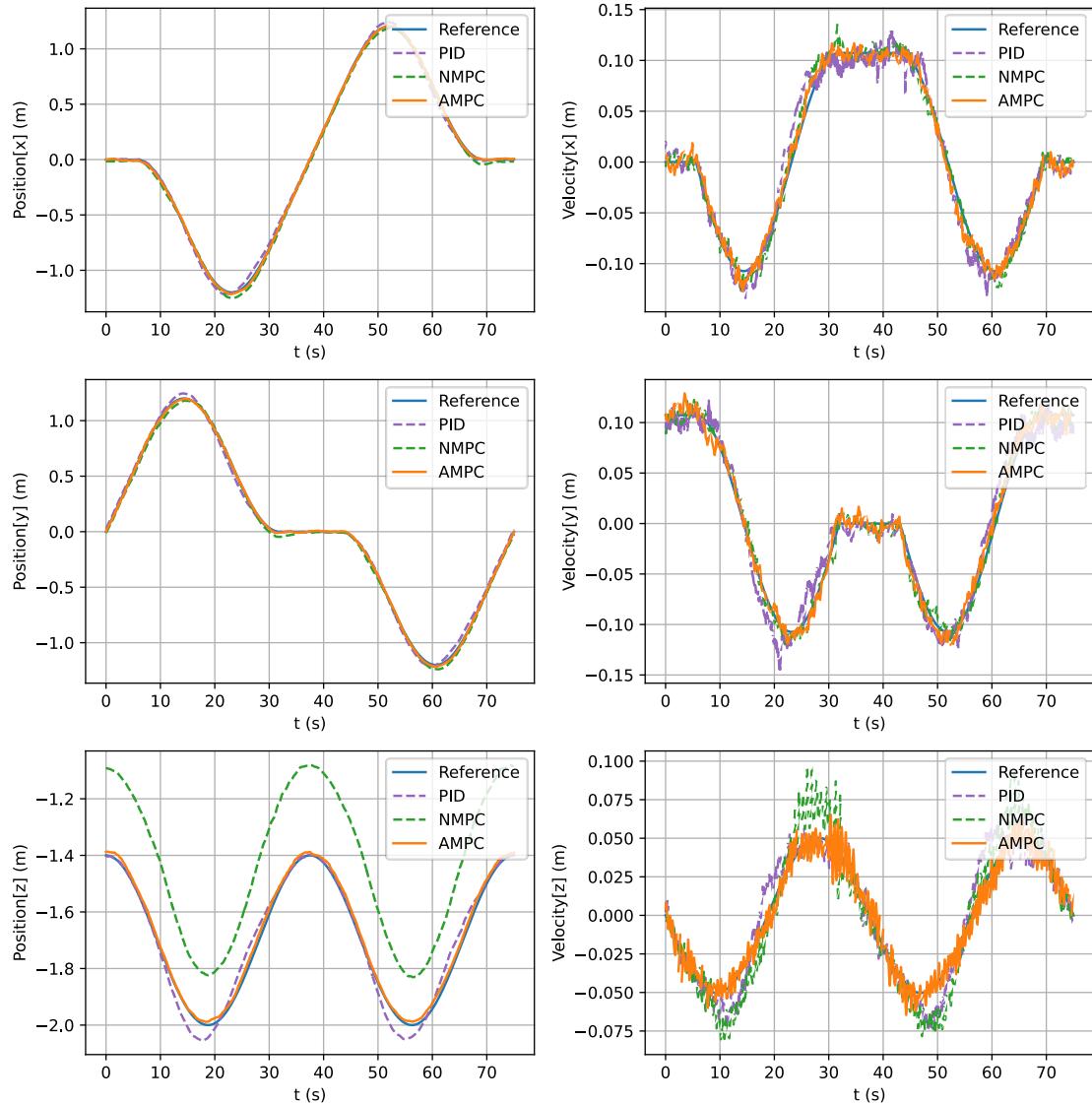
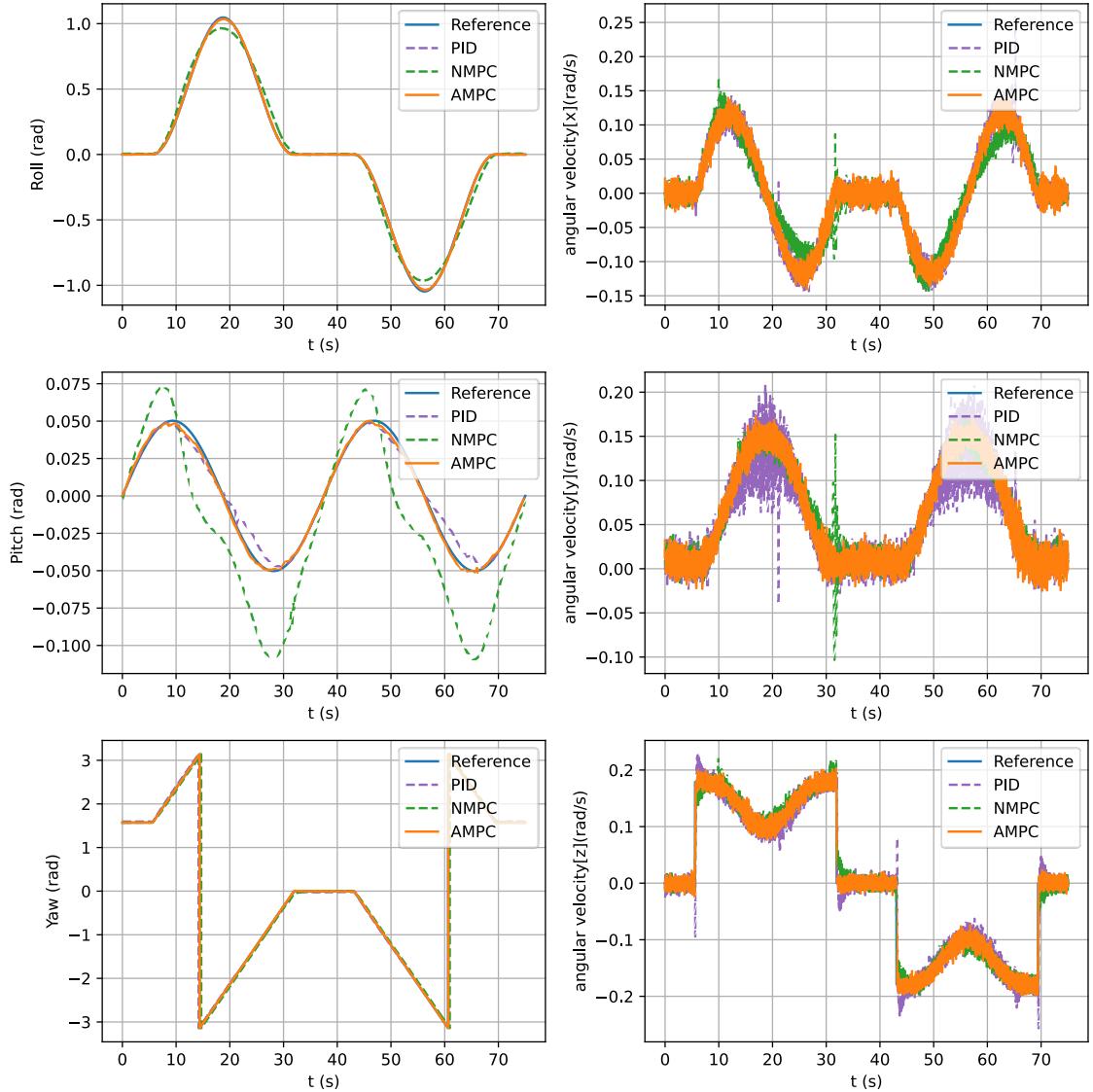


图 5-16 Gazebo 仿真平动追踪效果 ( $T = 75$ )

图 5-17 Gazebo 仿真旋转追踪效果 ( $T = 75$ )

## 5.4 实物实验

按照计划，完整的实物实验本应该是在 OmniHex 物理本体上对 Gazebo 仿真结果（图 5-13）进行重现。但因项目结题将至，剩余时间有限，开展实物实验的条件不足，故本节内容难称完整。即便如此，在实物上已完成的实验已能够在一定程度上验证 PID 控制器和  $\mathcal{L}_1$ -MPC 的可行性。

已完成的实验包括：一、使用 PID 控制器的遥控飞行；二、使用 PID 控制器在 Offboard 模式下的轨迹追踪；三、 $\mathcal{L}_1$ -MPC 的基本飞行实验。这些内容将在本小节后文分条呈现。

### 5.4.1 遥控飞行

在室内动捕提供位姿估计的环境下，OmniHex 可以稳定悬停，位置和姿态控制相互独立。图 5-18 展示了飞行器在不同姿态下的悬停情况。



图 5-18 不同姿态下的悬停飞行

其中左图展示的姿态为  $\phi = 0, \theta = 30^\circ, \psi = 90^\circ$ ，中图为  $\phi = 0, \theta = -20^\circ, \psi = 135^\circ$ ，右图为  $\phi = 0, \theta = 28^\circ, \psi = 180^\circ$ 。可以明显看到机臂倾转的情况，符合效率最优的分配逻辑。一次典型的多模式的遥控飞行测试记录的位姿数据如下图所示。

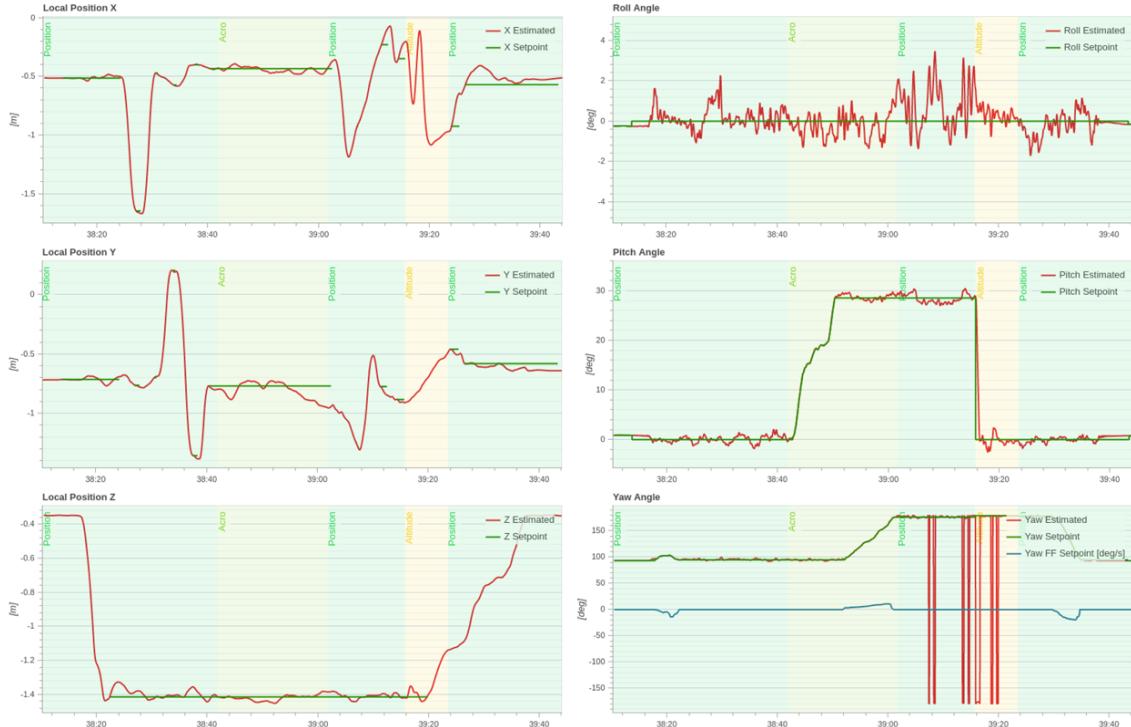


图 5-19 不同飞行模式下的位姿数据（Flight Review）

基于 PX4 的软件框架和其中的算法，本课题为遥控操作设计了三种飞行模式，分别是 Altitude、Position、ACRO（Acrobatic）模式，对应图 5-19 中的浅绿、绿、

黄三种底色。

在 Position 模式下，Roll 和 Pitch 锁定为刚进入此模式的值，Yaw 的期望值可以由摇杆输入改变；位置控制则是：在摇杆不输入信号时，锁定最近一次停止摇杆输入时的位置为期望值，若摇杆有输入则不会设定期望值，而是将摇杆输入换算成加速度直接产生期望合力。

在 ACRO 模式下，原本用于前进和后退的摇杆变为改变 Pitch，其他期望值的产生逻辑与位置控制相同；图 5-19 展示的 ACRO 模式下，没有改变位置和 Roll 的期望值，但是通过摇杆改变了 Yaw 和 Pitch 的期望值。

而 Altitude 模式和 Position 模式的摇杆输入逻辑相同，但不设定位置期望值，即位置控制不起作用；进入 Altitude 模式会使 Roll 和 Pitch 期望值阶跃为 0。

图 5-19 中，实际位置均保持在期望位置 $\pm 0.2\text{m}$  的范围内，实际姿态欧拉角基本保持在期望值的 $\pm 2$  度范围内。Yaw 的数据范围是 $[-\pi, \pi]$ ，所以“Yaw Angle”图中数据在 $\pi$ 附近发生突变，但 OmniHex 物理姿态并未发生突变。若仔细调节控制器参数，可以获得更好的控制效果。

#### 5.4.2 Offboard 模式轨迹追踪

在 Offboard 模式下使用 PID 控制器，对周期为 30、45、60 秒的轨迹进行了追踪， $k_{roll}$  设置为仿真中 $k_{roll}$  的半值。OmniHex 真实飞行轨迹如下图所示：

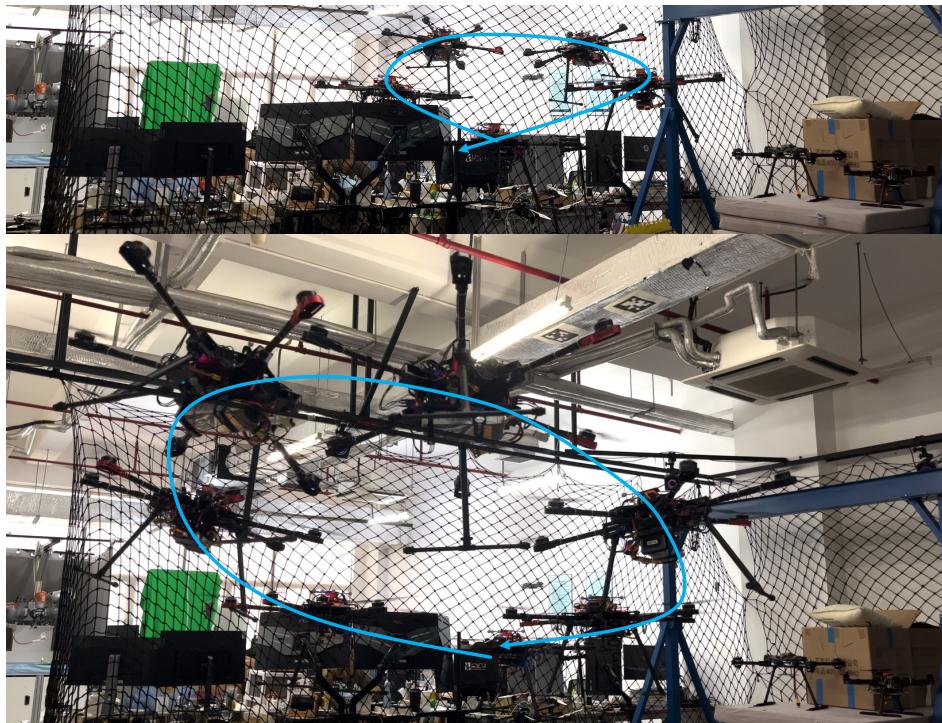


图 5-20 OmniHex 真实飞行轨迹

类似图 5-13，对追踪误差进行统计，结果如下。

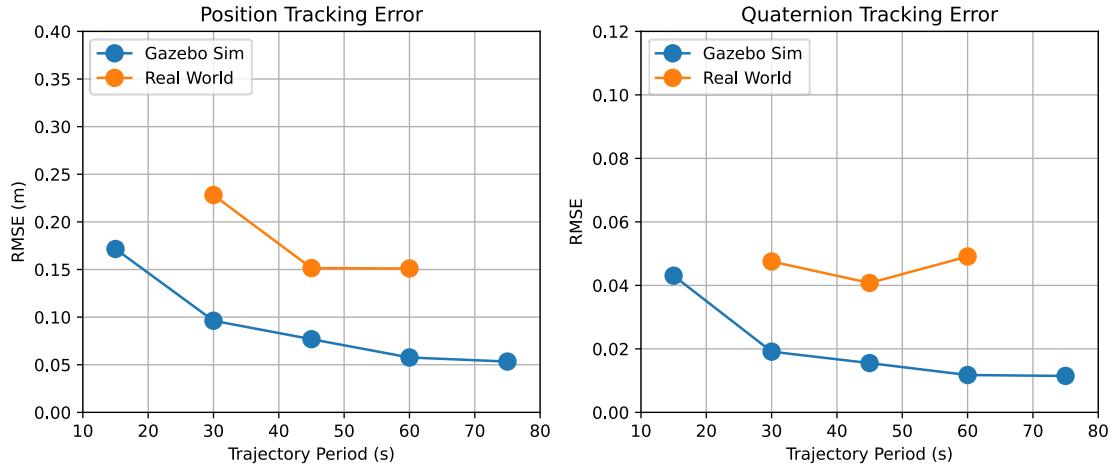


图 5-21 实物实验追踪误差统计

图 5-21 中除了实物实验的误差结果，还画出了图 5-13 中 PID 控制器的结果作为参考。实物实验的结果均差于仿真结果：如前文所述，实物上适用的参数和仿真中参数不一致，但由于时间关系，实物实验时的参数并未经过系统性的调节；若进行科学的、系统的调参，则可获得更好的追踪效果。位置误差数据的整体趋势与仿真结果一致，而姿态误差数据则在周期为 60 秒时反而增大了，为探求其原因，取完整的时域数据进行分析。下面对比轨迹周期为 30 秒和 60 秒的实验结果。

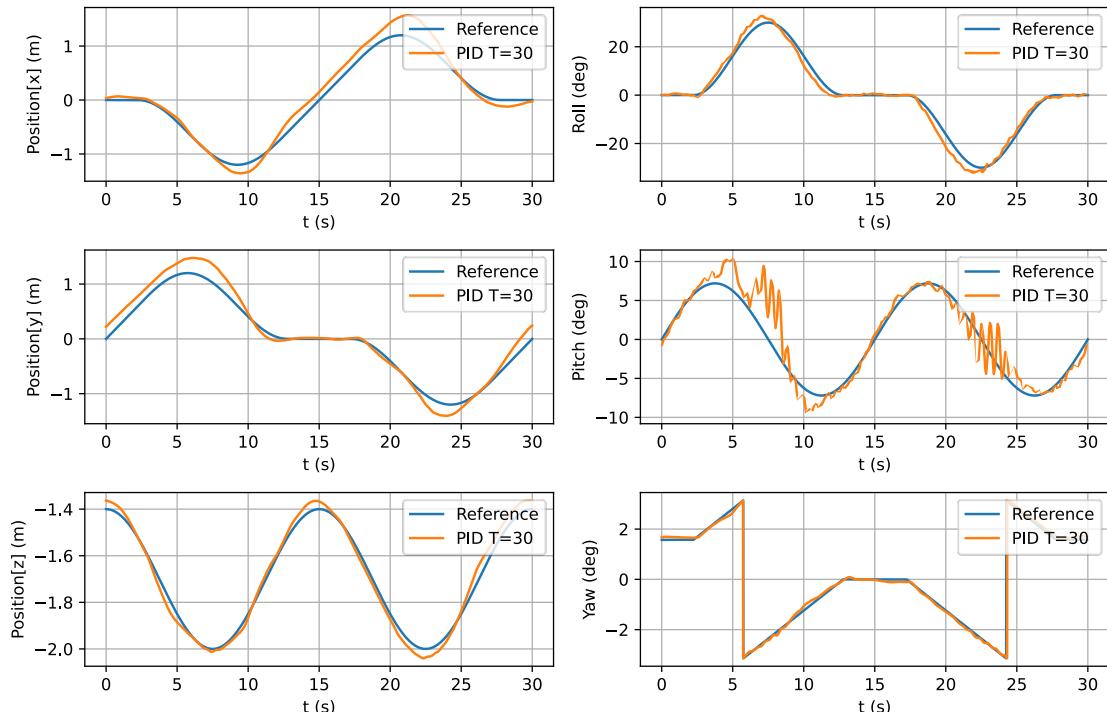
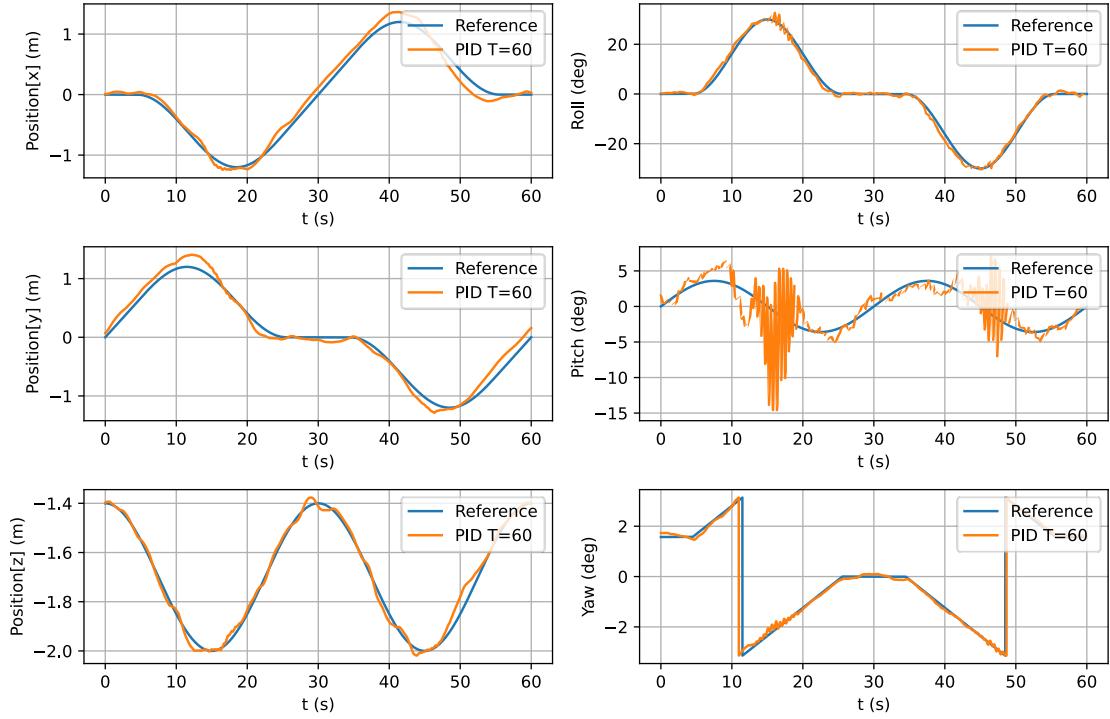


图 5-22 实物实验 PID 位姿追踪效果 ( $T = 30$ )

图 5-23 实物实验 PID 位姿追踪效果 ( $T = 60$ )

对比上两图可知位置上速度越快追踪效果越差，姿态上同样如此，但因为  $T = 60$  时 Pitch 出现了大幅度且高频的振荡，造成了图 5-21 中四元数的 RMSE 反而较大。

实物实验中 Pitch 上追踪效果差的现象与 Gazebo 仿真得到的结果相符。PID 方法在 Pitch 上出现这样有规律的误差模式，猜想是因为在不同的机臂倾转配置下，姿态控制器的控制参数的理想值是不同的，而飞行中并没有自适应的环节去调节姿态控制器的参数，于是出现了较大的超调和振荡。若猜想成立则可从另一方面证明 OmniHex 上自适应环节的重要性。

#### 5.4.3 $\mathcal{L}_1$ -MPC 的基本飞行实验

正如仿真中的 PID 参数无法无缝迁移至 OmniHex 物理本体上那样， $\mathcal{L}_1$ -MPC 的参数也需要重新调节，才能在 OmniHex 实物上达到类似的理想控制效果。在得到满意的参数之前，贸然对图 5-2 所示的轨迹进行追踪实验则可能导致事故发生。而目前暂未获得满意的参数，故此处只有调参过程中记录的基本飞行曲线。尽管如此，这些实验也足以作为对  $\mathcal{L}_1$ -MPC 算法有效性和可行性的初步验证。

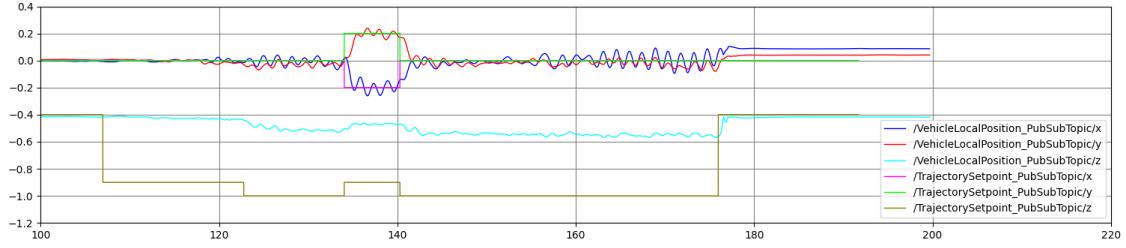


图 5-24 NMPC 的位置控制效果

图 5-24 展示了 NMPC 对阶跃轨迹的响应。其中高度的误差非常大：在设定点为-1 (NED 坐标系) 左右 OmniHex 才能勉强离地 (机体处于地面时高度为-0.4)，猜想是因为控制分配所用的相关参数误差较大。水平位置上控制能对阶跃信号做出响应，但在设定点附近出现了振荡现象，说明需要对参数进行进一步调整。

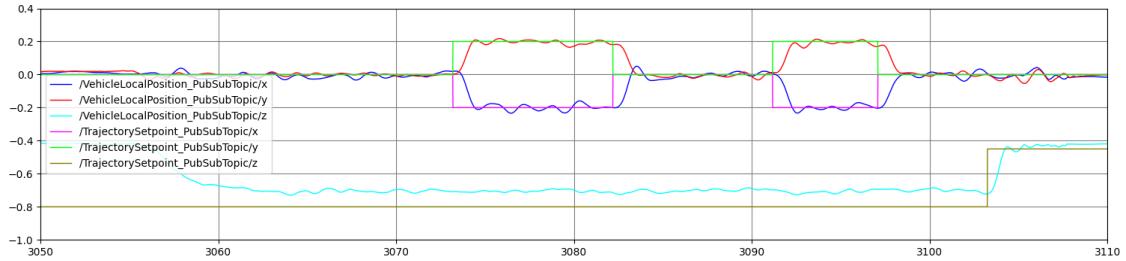


图 5-25 AMPC 的位置控制效果

在同一组 MPC 参数下，打开自适应补偿环节后，系统的阶跃响应如图 5-25 所示。自适应环节不仅对高度进行了补偿，而且使水平位置控制更加平滑，证明了  $\mathcal{L}_1$ -MPC 确实能对模型误差进行估计和补偿，提升总体控制效果。

下面展示系统在设置了不同的  $\mathcal{L}_1$  自适应控制器参数时的表现。 $\mathbf{A}$  矩阵的特征值反映了自适应控制器的增益，其特征值越接近 0，则称为自适应增益越高。注意  $\mathcal{L}_1$  自适应环节中包括一个低通滤波器，其截止频率越高则控制器对误差的估计越快，也更容易使系统状态出现超调和振荡。

图 5-26 至图 5-28 展示的是低增益，低截止频率时的系统表现，图 5-29 至图 5-31 则是调高了自适应增益和低通滤波器截止频率时的系统表现。图 5-30 记录了 Pitch 高频振荡随后发散的数据，说明此时参数不合理，需要降低增益或者 LPF 截止频率。

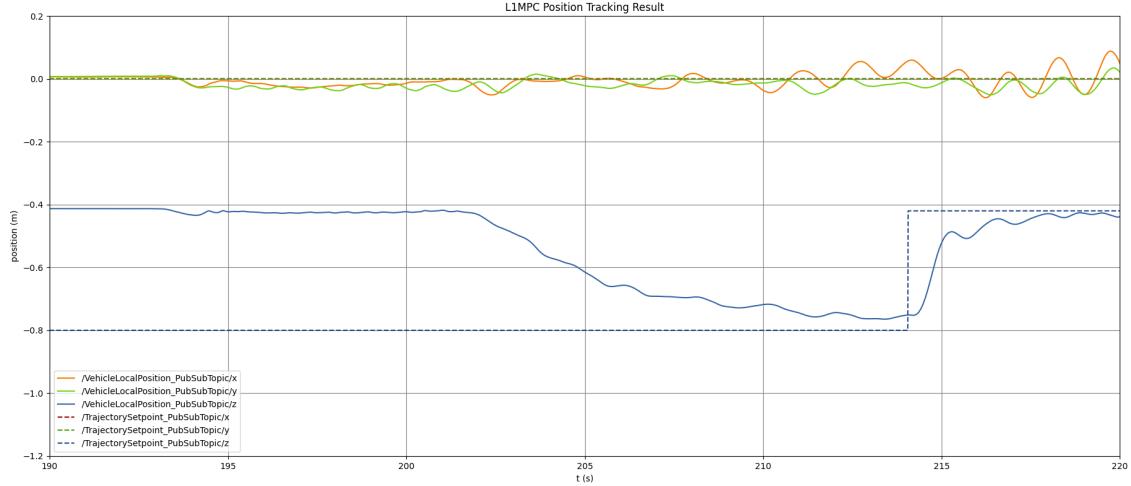


图 5-26 AMPC 的位置控制效果（低增益，低截止频率）

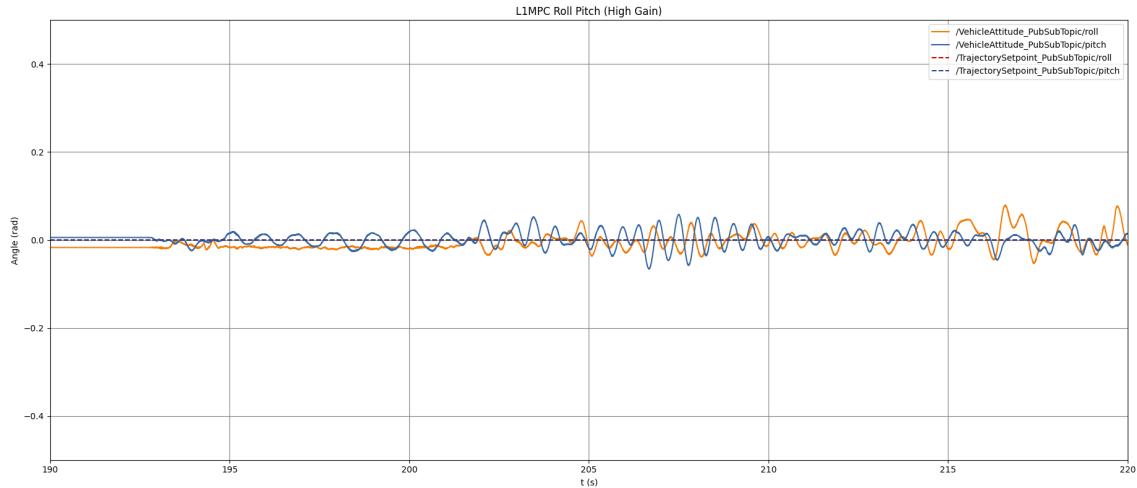


图 5-27 AMPC 的 Roll、Pitch 控制效果（低增益，低截止频率）

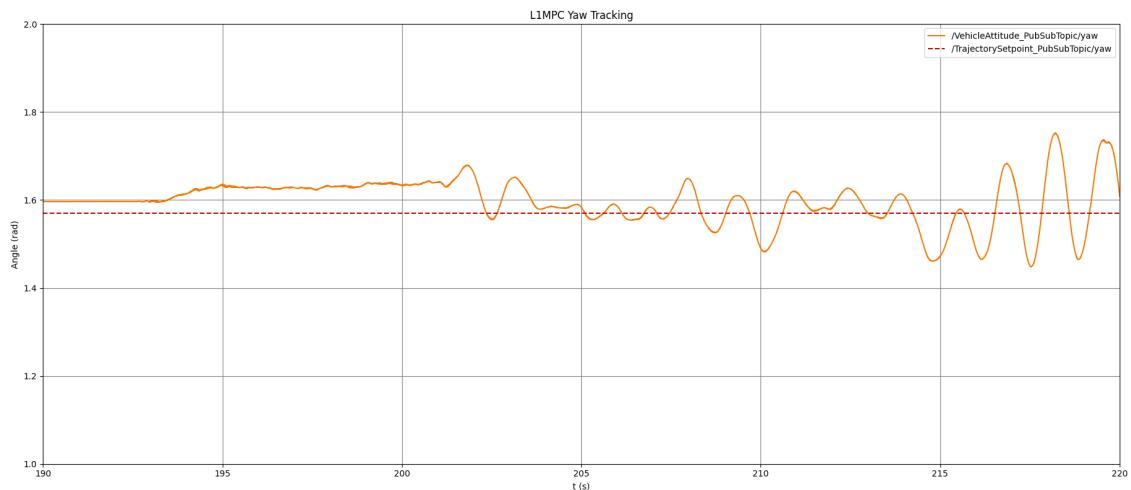


图 5-28 AMPC 的 Yaw 控制效果（低增益，低截止频率）

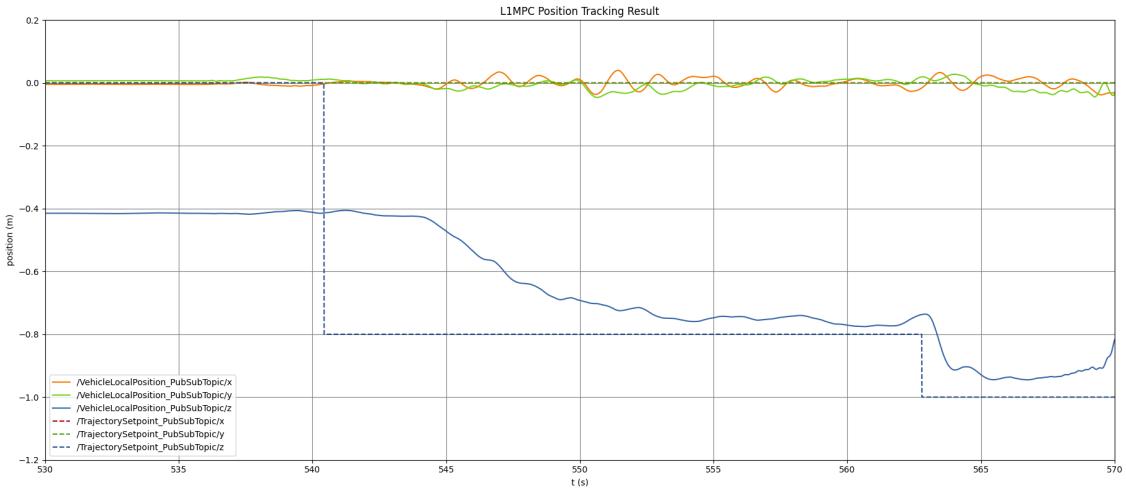


图 5-29 AMPC 的位置控制效果（高增益，高截止频率）

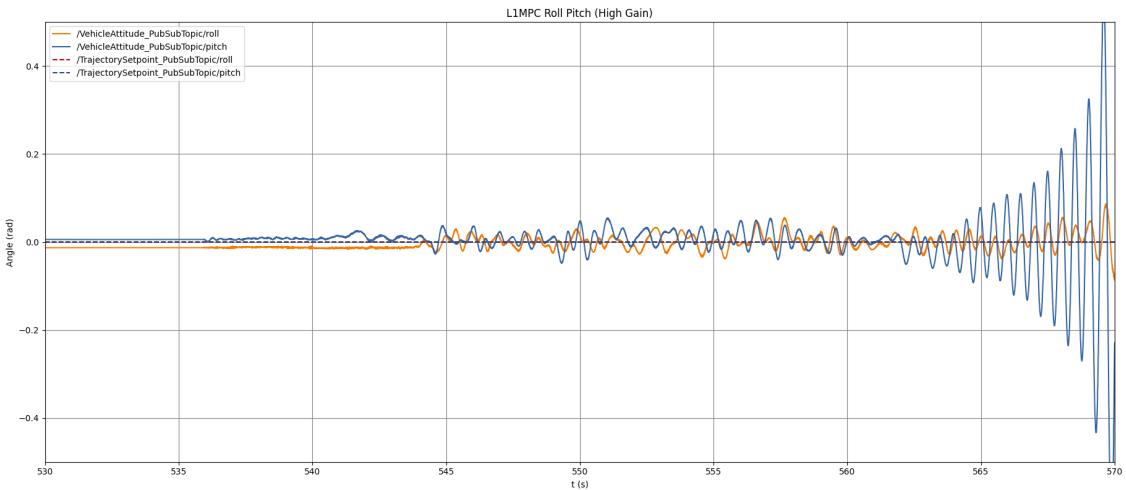


图 5-30 AMPC 的 Roll、Pitch 控制效果（高增益，高截止频率）

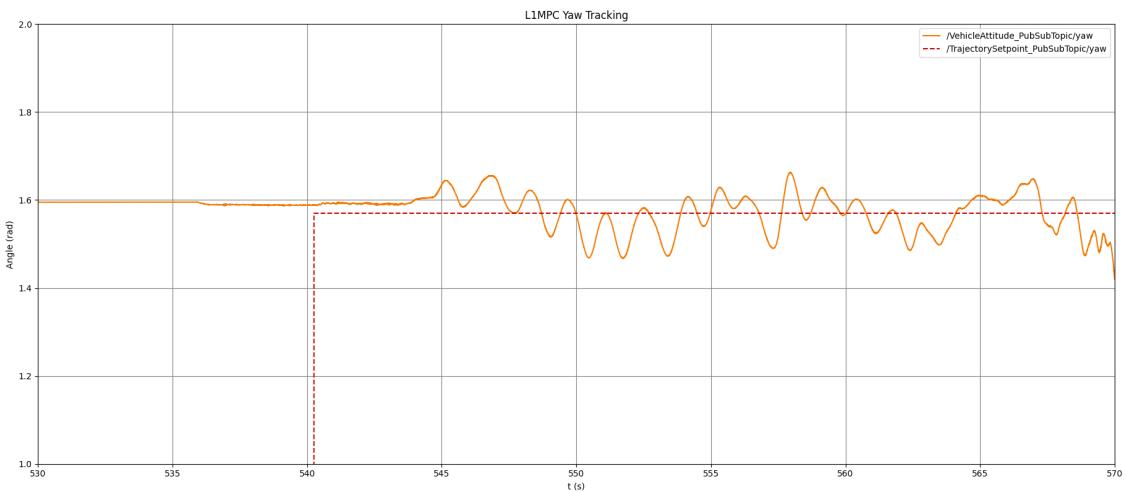


图 5-31 AMPC 的 Yaw 控制效果（高增益，高截止频率）

## 5.5 本章小结

本章首先描述了 OmniHex 的系统架构，介绍了其中重要的软件模块的作用和它们之间的数据流向。随后本章给出了用于测试控制器性能的轨迹的位置和欧拉角的解析式，并列写了生成全维度完整轨迹的算法。之后是在数值仿真和 Gazebo 仿真中对 NMPC、AMPC 包括 PID 控制器的性能进行了测试，证明了 AMPC 在模型误差较大的情况下也能保持很好的轨迹跟踪能力。在 OmniHex 物理本体上的实物实验，由于时间和工作量上的问题，并没有完成全部预定目标，跟踪曲线也难称得上漂亮，但也一定程度上验证了本文前述章节中叙述的 NMPC、 $\mathcal{L}_1$ -MPC、PID 控制、控制分配算法的有效性。

## 结 论

本文的主要成果是为 OmniHex 过驱动六旋翼无人机设计了基于非线性模型预测控制和  $\mathcal{L}_1$  自适应控制的自适应模型预测控制器： $\mathcal{L}_1$ -MPC。 $\mathcal{L}_1$ -MPC 在数值仿真实验、Gazebo 仿真实验中，展现了出色的轨迹跟踪能力和对模型误差和外力干扰的适应性；实物实验结果也初步展示了其有效性。本文还对课题研究过程中重要的辅助工作进行了叙述：无人机硬件和结构设计、基于前馈-PID 的控制器设计、控制分配算法的设计。这些工作对于成功开展仿真和实物实验是不可或缺的。

下面基于已取得的成果和课题研究中的点滴，对未来可开展的工作提出展望。硬件和结构方面：

- 1) 实际飞行时，中心板和机臂出现了肉眼可见的微小形变，对刚体模型假设是一种破坏，故未来可使用更厚的中心碳纤维板以及更粗的机臂。
- 2) 本文 2.2 节中提到法兰适配器精度和耐用性尚低于期望，在经过多次飞行调试和碰撞后，装配孔位磨损导致机臂倾转运动出现松动和死区，影响了控制器的控制效果，后续工作中可以考虑改变安装方法或者使用材料强度更高，加工更精密的零件。
- 3) 考虑使用单轴双电机的设计，以减小陀螺效应，或者是在建模部分考虑陀螺效应产生的力矩。

建模和优化求解方面：

- 1) 如果没有使用单轴双电机的设计，则应该在动态模型中包括陀螺效应。
- 2) 将动力学模型进一步往底层推导，比如将执行器建模部分放入 OCP 中，并加以考虑倾转电机和动力电机的动态，系统输入则设置为动力电机的转速和倾转电机的角度设定点。这样可以省去基于伪逆的控制分配算法，也不用单独考虑机臂线缆的反缠绕，可以解决目前在运动学奇异点处轨迹追踪效果差的问题。但是若这样子修改，则自适应控制部分可能需要重新设计。
- 3) 探索避免自适应环节破坏控制器约束的方案，比如使用自适应控制律更新系统参数的在线系统辨识的方法。
- 4) 本文虽然给出了旋翼转速与其产生的力和力矩的关系，但是并没有完整地建立 ESC 的指令 PWM 脉宽与电机转速的关系，导致控制分配产生的不确定性较大。虽说自适应环节可以对不确定性作出一定的补偿，但建立这部分的映射有助于进一步提高控制器性能。

软件系统实现方面：

- 1) 考虑增加运行时调节 $\mathcal{L}_1$ -MPC 参数的功能，如此能方便调参工作的开展。
- 2) 研究高效科学的调节参数方法。本文中实物实验的完整性正是由于缺乏高效科学的调参方法，且缺乏足够的调参时间而无法以完整的形态呈现。
- 3) 提高并固定自适应控制器的运行频率，本文中的设置是 MPC 和自适应在同一个循环中先后运行，频率为 100Hz，自适应的更新频率受限制，且由于 OCP 求解时间不固定，所以自适应环节的周期并不严格等于设定值。若将自适应部分移动至 NUTTX 实时系统上，并提高运行频率至 400Hz，则应该能有更好的自适应性能和更好的轨迹追踪效果。
- 4) PID 控制方案中，考虑在 ACRO 模式中加入用摇杆改变所有欧拉角设定点的功能。
- 5) 为 MPC 增加用摇杆输入参考轨迹的功能。

## 参考文献

- [1] HAMANDI M, USAI F, SABLE Q, STAUB N, TOGNON M. Design of multirotor aerial vehicles: a taxonomy based on input allocation [C]. HAL, 2021, hal-02433405v2.
- [2] BODIE K, TAYLOR Z, KAMEL M, SIEGWART R. Towards efficient full pose omnidirectionality with overactuated MAVs [C]. arXiv, 2018, 1810.06258v3.
- [3] ALLENSPACH M, BODIE K, BRUNNER M, RINSOZ L, TAYLOR Z, KAMEL M, SIEGWART R, NIETO J. Design and optimal control of a tiltrotor micro-aerial vehicle for efficient omnidirectional flight [J]. The International Journal of Robotics Research, 2020, 39(10-11): 1305–1325.
- [4] BRUNNER M, BODIE K, KANMEL M, PANTIC M, ZHANG W, NIETO J, SIEGWART R. Trajectory tracking nonlinear model predictive control for an overactuated MAV [C]. IEEE International Conference on Robotics and Automation, 2020: 5342.
- [5] ZHANG W, BRUNNER M, OTT L, KAMEL M, SIEGWART R, NIETO J. Learning dynamics for improving control of overactuated flying systems [C]. arXiv, 2020, 2006.13153.
- [6] ZHANG W, TOGNON M, OTT L, SIEGWART R, NIETO J. Active model learning using informative trajectories for improved closed-loop control on real robots [C]. arXiv, 2021, 2101.08100.
- [7] PERIC L, BRUNNER M, BOIDIE K, TOGNON M, SIEGWART R. Direct force and pose NMPC with multiple interaction modes for aerial push-and-slide operations [C]. IEEE International Conference on Robotics and Automation, 2021: 131-137.
- [8] LU K, YANG Z, LIAO L, JIANG Y, XU C, XU H, ZHANG Q. Extended state observer-based robust control of an omnidirectional quadrotor with tiltable rotors [J]. Transactions of the Institute of Measurement and Control, 2021, 43(5): 1143–1155.
- [9] BICEGO D, MAZZETTO J, CARLI R, FARINA M, FRANCHI A. Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs [J]. Journal of Intelligent & Robotic Systems, 2020, 100: 1213–1247.
- [10] NEUNERT M, CROUSAZ C, FURRER R, KAMEL M, FARSHIDIAN F, SIEGWART R, BUCHLI J. Fast nonlinear model predictive control for unified

- trajectory optimization and tracking [C]. IEEE International Conference on Robotics and Automation, 2016: 1398-1404.
- [11] FALANGA D, FOEHN P, LU P, SCARAMUZZA D. PAMPC: perception-aware model predictive control for quadrotors [C]. arXiv, 2018, 1804.04811.
  - [12] BOUFFARD P, ASWANI A, TOMLIN C. Learning-based model predictive control on a quadrotor: onboard implementation and experimental results [C]. IEEE International Conference on Robotics and Automation, 2012: 279-284.
  - [13] TORRENTE G, KAUFMANN E, FOHN P, SCARAMUZZA D. Data-driven MPC for quadrotors [C]. arXiv, 2021, 2102.05773.
  - [14] HANOVER D, FOEHN P, SUN S, KAUFMANN E, SCARAMUZZA D. Performance, precision, and payloads: adaptive nonlinear MPC for quadrotors [C]. IEEE Robotics and Automation Letters, 2021, 7(2): 690-697.
  - [15] SUN S, ROMERO A, FOEHN P, KAUFMANN E, SCARAMUZZA D. A comparative study of nonlinear MPC and differential-flatness-based control for quadrotor agile flight [C]. arXiv, 2021, 2109.01365.
  - [16] PRAVITRA J, ACKERMAN K A, CAO C, HOVAKIMYAN N, THEODOROU E A.  $\mathcal{L}_1$ -Adaptive MPPI Architecture for Robust and Agile Control of Multirotors [C]. arXiv, 2020, 2004.00152.
  - [17] BRESCIANINI D, HEHN M, D'ANDREA R. Nonlinear quadcopter attitude control [R]. ETH Zurich Research Collection, 2013, ethz-a-009970340.
  - [18] ELKHATIB O. Control allocation of a tilting rotor hexacopter [D]. ETH Zurich Research Collection, 2017, ethz-b-000224598.
  - [19] VERSCHUEREN R, KOUZOUPIS D, et al. Acados - a modular open-source framework for fast embedded optimal control [C]. arXiv, 2020, 1910.13753v3.
  - [20] QUAN Q. Introduction to multicopter design and control [M]. Singapore: Springer Nature, 2017, 4: 73-95.
  - [21] WAMPLER C. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1986, 16(1): 93-101.
  - [22] DIEBEL J. Representing attitude: Euler angles, unit quaternions, and rotation vectors [J]. Matrix, 2006, 58(15-16): 1-35.
  - [23] HOVAKIMYAN N, CAO C.  $\mathcal{L}_1$  adaptive control theory: guaranteed robustness with fast adaptation [M]. Society for Industrial and Applied Mathematics, 2010.
  - [24] WANG X, HOVAKIMYAN N.  $\mathcal{L}_1$  adaptive controller for nonlinear time-varying reference systems [J]. Systems & Control Letters, 2012, 61(4): 455-463.
  - [25] LI Z, HOVAKIMYAN N.  $\mathcal{L}_1$  adaptive controller for MIMO systems with unmatched uncertainties using modified piecewise constant adaptation law [C].

- 2012 IEEE 51st IEEE conference on decision and control (CDC). IEEE, 2012: 7303-7308.
- [26] KHARISOV E, HOVAKIMYAN N, ÅSTRÖM K. Comparison of several adaptive controllers according to their robustness metrics [C]. AIAA Guidance, Navigation, and Control Conference. 2010: 8047.
- [27] MEIER L, HONEGGER D, POLLEFEYS M. PX4: A node-based multithreaded open-source robotics framework for deeply embedded platforms [C]. 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015: 6235-6240.
- [28] KOUBÂA A. Robot Operating System (ROS) [M]. Cham: Springer, 2017.
- [29] MARUYAMA Y, KATO S, AZUMI T. Exploring the performance of ROS2 [C]. Proceedings of the 13th International Conference on Embedded Software. 2016: 1-10.
- [30] KIM J S. Recent advances in adaptive MPC [J]. ICCAS 2010, 2010: 218-222.
- [31] VERSCHUEREN R, FRISON G, KOUZOUPIS D, et al. Towards a modular software package for embedded optimization [J]. IFAC, 2018, 51(20): 374-380.
- [32] BARROS CARLOS B, SARTOR T, ZANELLI A, et al. An efficient real-time NMPC for quadrotor position control under communication time-delay [J]. arXiv e-prints, 2020: arXiv: 2010.11264.
- [33] FRISON G, DIEHL M. HPIPM: A high-performance quadratic programming framework for model predictive control [J]. IFAC, 2020, 53(2): 6563-6569.
- [34] FRISON G, FREY J, MESSEMER F, et al. Introducing the quadratically-constrained quadratic programming framework in HPIPM [J]. arXiv preprint arXiv:2112.11872, 2021.
- [35] STEPANYAN V, KRISHNAKUMAR K. MRAC revisited: guaranteed performance with reference model modification [C]. Proceedings of the 2010 American Control Conference. IEEE, 2010: 93-98.

### 哈尔滨工业大学深圳校区本科生毕业设计（论文）原创性声明

本人郑重声明：在哈尔滨工业大学深圳校区攻读学士学位期间，所提交的毕业设计（论文）《过驱动六旋翼无人机的自适应模型预测控制》，是本人在导师指导下独立进行研究工作所取得的成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明，其它未注明部分不包含他人已发表或撰写过的研究成果，不存在购买、由他人代写、剽窃和伪造数据等作假行为。

本人愿为此声明承担法律责任。

作者签名:  日期: 2022 年 6 月 10 日

## 致 谢

首先我想向我的导师陈浩耀教授表达感谢：感谢陈老师提供 NROS 实验室这样优秀的研究平台，并提供经费大力支持这项工作的进行。同时我想感谢理学楼 G305 的师兄们：在进行课题研究这将近一年的时间里，他们包容并帮助我使用实验室的场地和资源，忍受无人机飞行时的各种噪音；其中我想特别感谢全凤宇、王思强、刘博瑜、王巨学长，他们在我灵感迸现之前总能耐心地听我复述不知道哪里出错了的代码思路，比小黄鸭不知道高到哪里去了。当然还有刘健恒、任云帆、栾昊学长，他们也提供了很多经验和帮助。我还想感谢研究过驱动飞行器轨迹规划的同学刘培焱，他为 OmniHex 零件的加工和制造做出了贡献。

在螺旋桨的轰鸣中，在 CPU 占用率和散热风扇声音的律动中，我曾因滑稽可笑的失败怀疑过自己、曾为不断出现的硬件和算法上的小问题苦恼不已，当然也曾有过那么几次取得小成果时的欢呼雀跃。验知真理的喜悦、电光石火的灵感、停滞倒退的苦闷、不甘失败的坚持……是这篇论文背后的点点滴滴。

如果说毕业论文是一个句号，那么没有亲朋好友的陪伴和鼓励，这个句号则是不完整的。我要感谢我的同学陈凌栩、付乐天、李奥淇、赵浩钧、雷轩昂、唐本旭、陈志成、黄俊昌、张大明……只恨“这里空白的地方太小，写不下”心里所有的名字。哈深四年，岁月如歌！我们一起啃过数分高代复变自控、为考试熬夜复习到凌晨，也曾顶着正午烈日在室外采集数据。我们一起看过主楼前的日出和大沙河尽头的日落，一起在球场和跑道上挥汗如雨，一起望向天边的彩霞唱着飞云之下、走在荔园小道上哼着春风十里。感谢我的父母，提供物质上的保障和精神上的支持，让我能够专注于学业而无需考虑繁杂琐事。

北航一位研究空间电力推进的学长在致谢中写道：“我期待着参加一场在地球近地轨道上举办的太空音乐会，那是在一座大型空间站的球形玻璃大厅里，所有的乐团成员漂浮在空中，用矢量推进背包保持着队形，背后是蓝色星球，奏起 *Sogno di Volare*。在歌声中，在蓝色星球的背景里，人类的第一艘核聚变动力飞船拖着耀眼而温暖的光芒，徐徐加速，前往最后的边疆，前往无限未来……此去经年，只愿这真空舱中的星星蓝火，可以燎原”。我的课题和这位学长的课题相去甚远，没有那么看似遥不可及宛若星辰，但我任然想引用这段无比绚烂的文字作为全文的结尾，它有一种不可名状的、直击我心灵的力量，我也想去听一听那群星中的演奏！望今后能对社会和国家作出贡献，努力向那所描述的浪漫未来靠近，哪怕是小小一步……愿永远轻盈、永远滚烫、无限进步。