



多旋翼飞行器设计与控制

第十三讲 任务决策

全权 副教授

qq_buaa@buaa.edu.cn

自动化科学与电气工程学院

北京航空航天大学



东方智慧



博弈



核心问题

多旋翼在全自主控制和半自主控制下的
任务决策机制分别是什么？



大纲

1. 全自主控制
2. 半自主控制
3. 本讲小结



1. 全自主控制

□ 总体介绍

(1) **任务规划** 通常由操作人员离线完成，主要进行任务的阶段划分，以及在各航路点的任务分配。

(2) **路径规划** 根据飞行任务点给定期望的可飞路径，即期望的飞行器空间三维坐标。可以是连续的路径，也可以是离散的目标点。

(3) **底层控制** 控制电机转动，实现对可飞路径的跟踪。

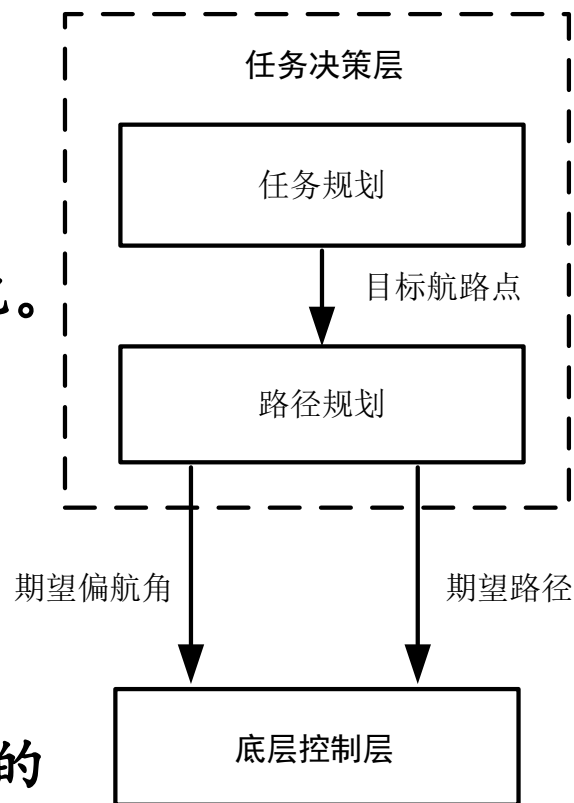


图13.1 多旋翼全自主控制层次



1. 全自主控制

□ 总体介绍

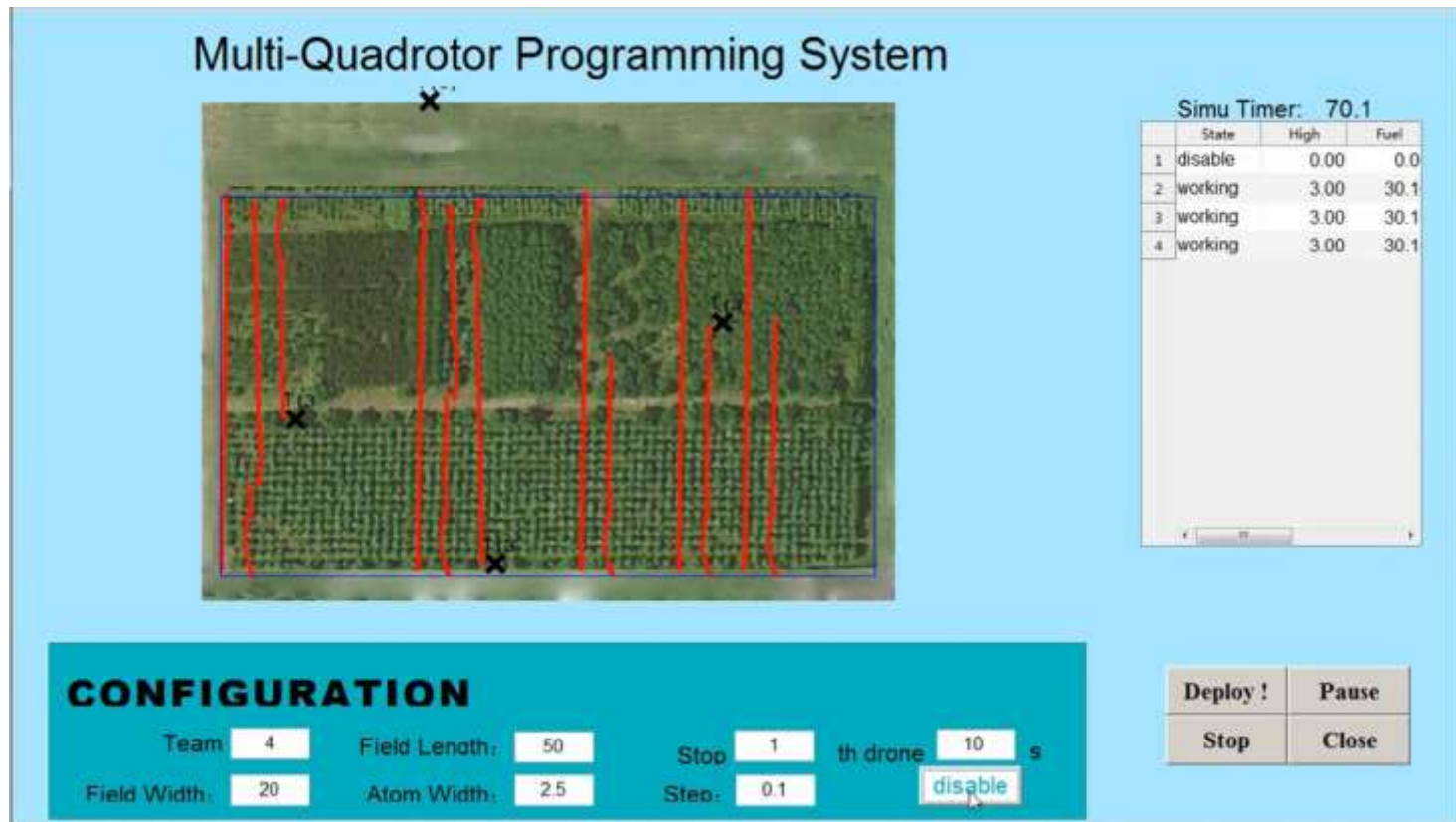


图13.1 Mission Planner地面站多旋翼航路规划



1. 全自主控制

□ 总体介绍



视频：多无人机协同农田喷洒协议设计, <https://youtu.be/fkp043jE5fQ>



1. 全自主控制

□ 任务规划

(1) 任务需求

1) 给定条件

① 已知期望航路点

$\mathbf{p}_{wp,0}, \mathbf{p}_{wp,1}, \dots, \mathbf{p}_{wp,k}, \dots, \mathbf{p}_{wp,n_{wp}}$
 $\in \mathbb{R}^3, k = 0, \dots, n_{wp};$

② 已知多旋翼实时位置 \mathbf{p} 和电量 b 。

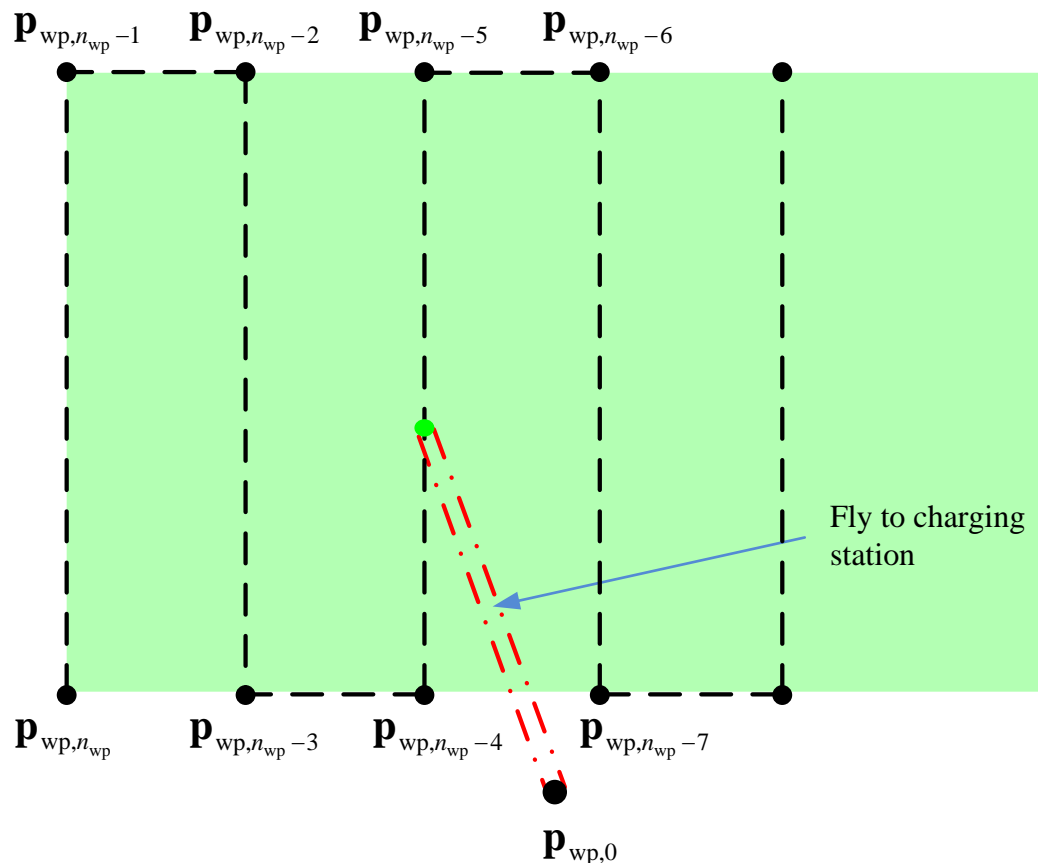


图13.2 S形遍历任务规划示意图



1. 全自主控制

□ 任务规划

2) 任务需求

- ①多旋翼从充电站 $p_{wp,0}$ 出发，按S型遍历所有期望航路点；
- ②期望偏航角指向下一个航路点；
- ③当电量低于门限值时，多旋翼保存当前位置并返回充电站充电，充电完成后飞回原地继续完成任务；
- ④遍历完所有航路点后多旋翼飞回充电站。

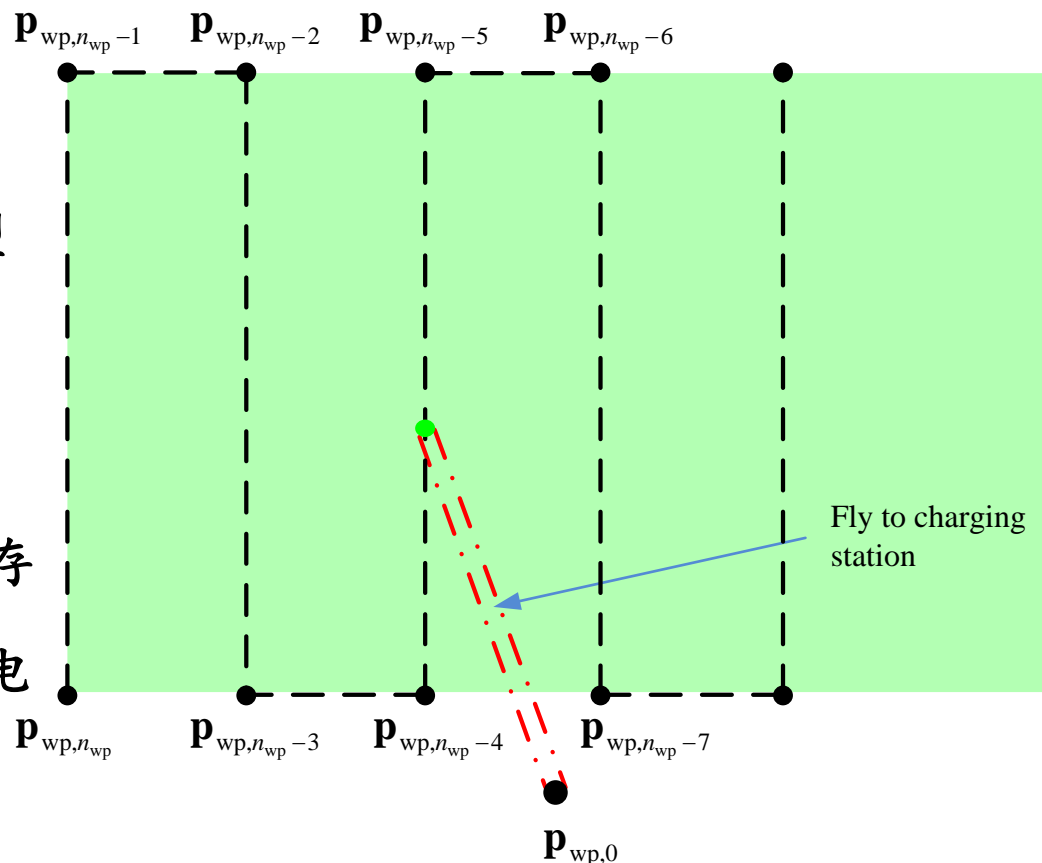


图13.3 S形遍历任务规划示意图



1. 全自主控制

□ 任务规划

(2) 算法设计

输入：多旋翼实时位置信息 p ，实时电量信息 b 。

输出：实时目标位置 p_d （作为位置控制器期望输入）。

初始化阶段

1) 确定矩形区域坐标系；

2) 根据任务需求生成需要遍历的航路点

$p_{wp,1}, \dots, p_{wp,k}, \dots, p_{wp,n_{wp}}$ 坐标及充电站坐标 $p_{wp,0}$ ，**并将所有航路点坐标依次压入栈中；**

3) 令 $k = n_{wp}$ ，并令当前目标航路点为 $p_d = p_{wp,k}$ ，设置 $\varepsilon > 0, b_{th} > 0$ ，其中 ε 用于判断多旋翼是否到达航路点， b_{th} 是电量门限值；

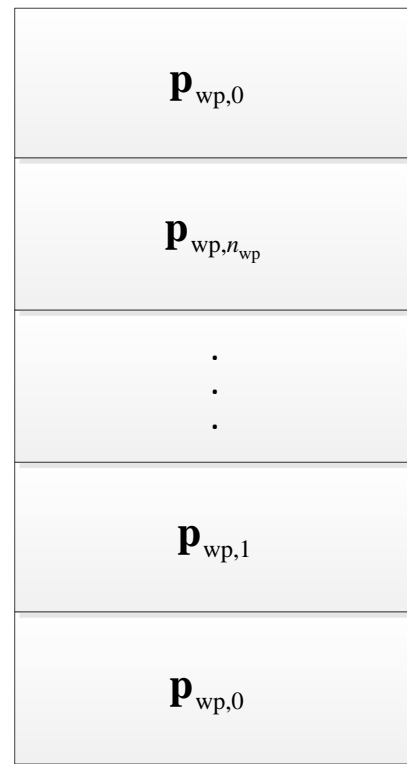


图13.4 航路点入栈示意图



1. 全自主控制

□ 任务规划

(2) 算法设计

4) 如果 $\|p_d - p\| \leq \varepsilon$, 说明多旋翼已进入目标航路点, 转至 5) 否则转至 6) ;

5) 如果 $p_d = p_{wp,0} \& k = 0$, 说明多旋翼已遍历完航路点, 结束任务; 如果 $p_d = p_{wp,0} \& k \neq 0$, 说明任务还没完成, 只是电池电量过低返回充电。当多旋翼充电完成后, 停在原地等待起飞指令。一旦接收到起飞指令, 则执行 $k = k - 1$, $p_d = p_{wp,k}$ 并跳转至 4) 继续运行; 如果 $p_d \neq p_{wp,0}$, 说明多旋翼任务还未完成且电量充足, 执行 $k = k - 1$, $p_d = p_{wp,k}$ 并跳转至 6) 继续执行;

算法执行过程

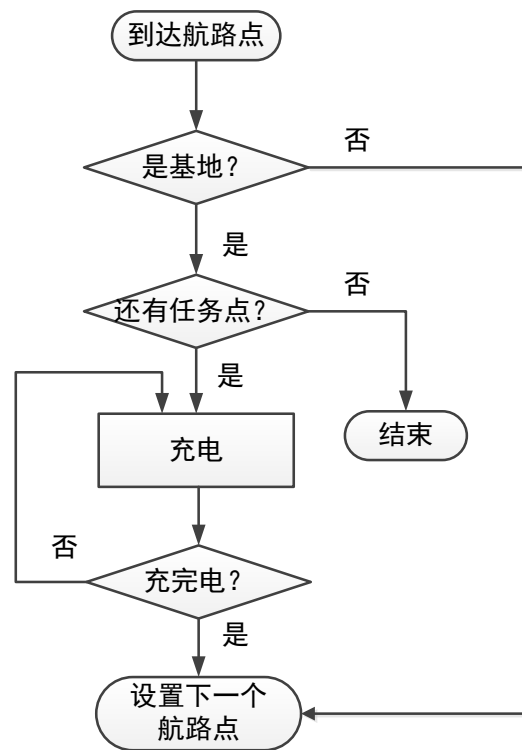


图13.5 算法流程图



1. 全自主控制

□ 任务规划

(2) 算法设计

6) 如果电池电量 $b < b_{th}$ & $p_d \neq p_{wp,0}$, 说明多旋翼在飞向航路点的过程中电量过低, 需要将当前位置作为航路点加入栈中, 再将充电站位置加入栈中, 以使多旋翼先飞回充电站充电再返回当前位置继续完成任务, 该过程通过执行以下指令实现:

$$k = k + 1, p_{wp,k} = p$$

$$k = k + 1, p_{wp,k} = p_{wp,0}$$

$$p_d = p_{wp,k}$$

否则跳转至 4) 继续运行。

电量不足, 保护现场

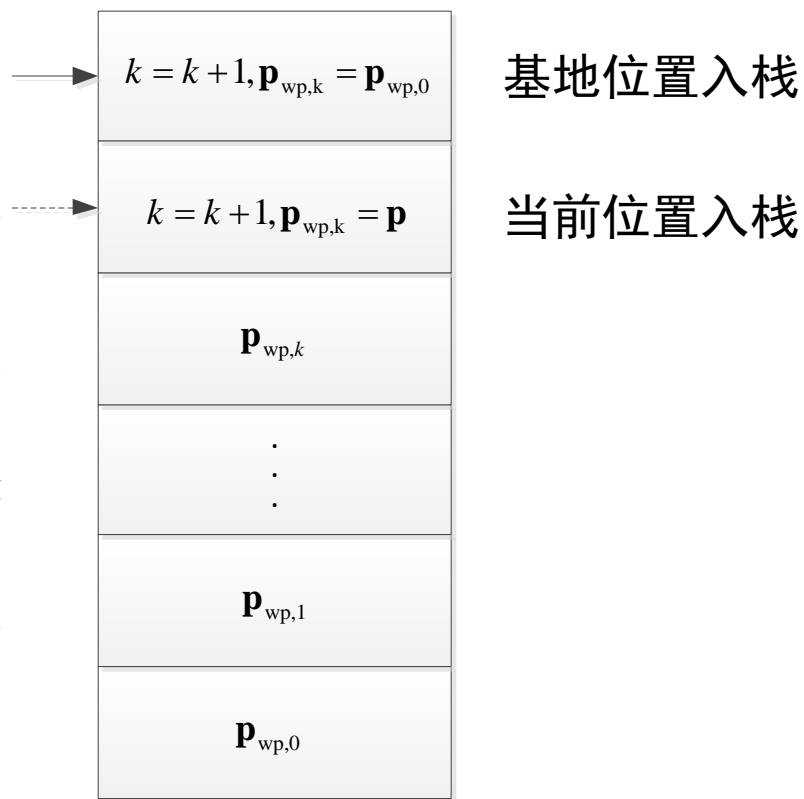


图13.6 电量不足时算法的处理过程示意图



1. 全自主控制

□ 路径规划

(1) 基本概念

1) 路径规划：多旋翼在飞行中不仅受到自身约束的限制，还会受到环境约束的影响。路径规划是为了给多旋翼提供既满足自身约束又满足环境约束的可飞路径。

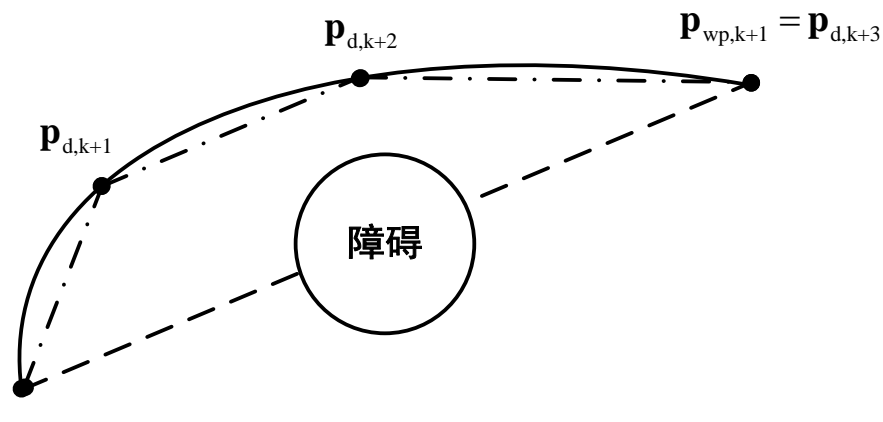


图13.7 全局路径规划和局部路径规划

2) 全局路径规划和局部路径规划[1]：根据任务生成航路点，并用直线将航路点连接起来，这一过程称为全局路径规划；根据局部环境条件和多旋翼的动力学运动学约束将直线路径提炼为可飞路径，这一过程称为局部路径规划。

[1] Tsourdos A, White B, Shanmugavel M. Cooperative path planning of unmanned aerial vehicles[M]. John Wiley & Sons, 2010. 2-12.



1. 全自主控制

□ 路径规划

(1) 基本概念

3) **人工势场法**：给目标航路点和直线路径分配吸引势场，给障碍物分配排斥势场，使三维飞行区域处于目标航路点、直线路径和障碍物的叠加势场中，飞行在该区域中的多旋翼将会被目标航路点和直线路径同时吸引，而被障碍物排斥，使得多旋翼在飞向目标航路点的同时沿着直线飞行，且能避开障碍物[2]。

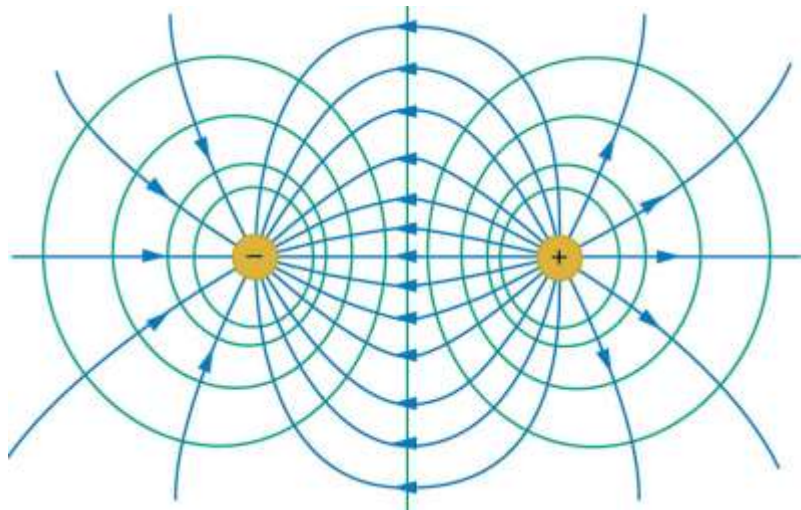


图13.8 势场示意图

[2] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, 1986, 5(1): 90-98



1. 全自主控制

□ 路径规划

(2) 直线路径跟随

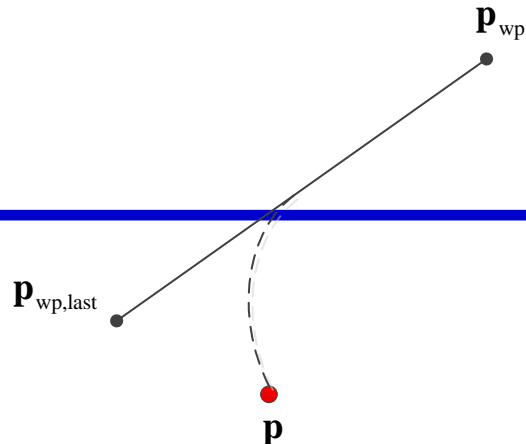


图13.9 直线路径跟随示意图

1) 问题描述：令 $\mathbf{p} \in \mathbb{R}^3$ 表示多旋翼当前位置， $\mathbf{p}_{wp} \in \mathbb{R}^3$ 表示目标航路点， $\mathbf{p}_{wp,last} \in \mathbb{R}^3$ 表示上一个航路点。设计局部路径规划器实时地生成期望位置 $\mathbf{p}_d \in \mathbb{R}^3$ ，引导多旋翼沿着 $\mathbf{p}_{wp,last} \in \mathbb{R}^3$ 和 $\mathbf{p}_{wp} \in \mathbb{R}^3$ 之间的直线路径飞行直到到达目标航路点 \mathbf{p}_{wp} 。

2) 算法设计

将多旋翼看作质点模型，其满足牛顿第二定律

$$\dot{\mathbf{p}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \mathbf{u}$$



全自主控制

□ 路径规划

(2) 直线路径跟随

2) 算法设计

若定义李雅普诺夫函数为

$$V_1 = \underbrace{k_0 \tilde{\mathbf{p}}_{\text{wp}}^T \tilde{\mathbf{p}}_{\text{wp}}}_{\text{靠近目标航路点}} + \underbrace{k_1 \tilde{\mathbf{p}}_{\text{wp}}^T \mathbf{A} \tilde{\mathbf{p}}_{\text{wp}}}_{\text{靠近期望路径}} + \frac{k_2}{2} \mathbf{v}^T \mathbf{v}$$

其中 $\tilde{\mathbf{p}}_{\text{wp}} \triangleq \mathbf{p} - \mathbf{p}_{\text{wp}}$ ，那么 $k_0, k_1, k_2 \in \{0\} \cup \mathbb{R}_+$ 意味着多旋翼能同时靠近目标航路点和期望路径。



全自主控制

□ 路径规划

(2) 直线路径跟随

2) 算法设计

然而，这里需要考虑控制量的饱和，因而需要对李雅普诺夫函数进行修改。定义 $\text{sat}_{\text{gd}}(k_0 \tilde{\mathbf{p}}_{\text{wp}} + k_1 \mathbf{A} \tilde{\mathbf{p}}_{\text{wp}}, a_0)$ 为沿着 $C_{\tilde{\mathbf{p}}}$ 分布的向量场，其中 $C_{\tilde{\mathbf{p}}}$ 表示由 $\tilde{\mathbf{p}}_{\text{wp}}$ 参数化的平滑曲线。**李雅普诺夫函数**可以设计为

$$V_1' = \underbrace{\int_{C_{\tilde{\mathbf{p}}}} \text{sat}_{\text{gd}}(k_0 \tilde{\mathbf{p}}_{\text{wp}} + k_1 \mathbf{A} \tilde{\mathbf{p}}_{\text{wp}}, a_0)^T d\tilde{\mathbf{p}}_{\text{wp}}}_{\text{靠近目标航路点并靠近直线航路}} + \frac{k_2}{2} \mathbf{v}^T \mathbf{v}$$

其中 $k_0, k_1, k_2 \in \{0\} \cup \mathbb{R}_+$, $a_0 \in \mathbb{R}_+$ 。 V_1 对时间求一阶导数得

$$\dot{V}_1' = \text{sat}_{\text{gd}}(k_0 \tilde{\mathbf{p}}_{\text{wp}} + k_1 \mathbf{A} \tilde{\mathbf{p}}_{\text{wp}}, a_0)^T \mathbf{v} + k_2 \mathbf{v}^T \mathbf{u}$$



全自主控制

□ 路径规划

(2) 直线路径跟随

2) 算法设计

如果虚拟控制量 $\mathbf{u} = -\frac{1}{k_2} \text{sat}_{\text{gd}}(k_0 \tilde{\mathbf{p}}_{\text{wp}} + k_1 A \tilde{\mathbf{p}}_{\text{wp}}, a_0) - \frac{1}{k_2} \text{sat}_{\text{gd}}(\mathbf{v}, a_1)$

那么 $\dot{V}'_1 = -\mathbf{v}^T \text{sat}_{\text{gd}}(\mathbf{v}, a_0) \mathbf{v}$, 当且仅当 $\mathbf{v} = \mathbf{0}_{3 \times 1}$ 时 $\dot{V}'_1 = 0$ 。 $\mathbf{v} = \mathbf{0}_{3 \times 1}$ 意味着

$$\frac{1}{k_2} \text{sat}_{\text{gd}}(k_0 \tilde{\mathbf{p}}_{\text{wp}} + k_1 A \tilde{\mathbf{p}}_{\text{wp}}, a_0) = \mathbf{0}_{3 \times 1} \quad (1)$$

根据不变集理论可知系统将全局收敛到 (\mathbf{p}, \mathbf{v}) , 其中 \mathbf{p} 是方程 (1) 的解, $\mathbf{v} = \mathbf{0}_{3 \times 1}$ 。调节参数 k_0, k_1 可以调整多旋翼靠近到直线航路和靠近到目标航路点的相对快慢



全自主控制

□ 路径规划

(2) 直线路径跟随

2) 算法设计

$$\mathbf{u} = -\frac{1}{k_2} \text{sat}_{\text{gd}} \left(k_0 \tilde{\mathbf{p}}_{\text{wp}} + k_1 A \tilde{\mathbf{p}}_{\text{wp}}, a_0 \right) - \frac{1}{k_2} \text{sat}_{\text{gd}} \left(\mathbf{v}, a_1 \right)$$

将虚拟控制量写成PD控制器的形式

$$\mathbf{u} = -\frac{k'_1}{k_2} (\mathbf{p} - \mathbf{p}_d) - \frac{1}{k_2} \text{sat}_{\text{gd}} (\mathbf{v}, a_1)$$

其中

$$\mathbf{p}_d = \mathbf{p} + \frac{1}{k'_1} \text{sat}_{\text{gd}} \left(k_0 (\mathbf{p}_{\text{wp}} - \mathbf{p}) + k_1 (\mathbf{p}_{\text{wp,perp}} - \mathbf{p}), a_0 \right)$$

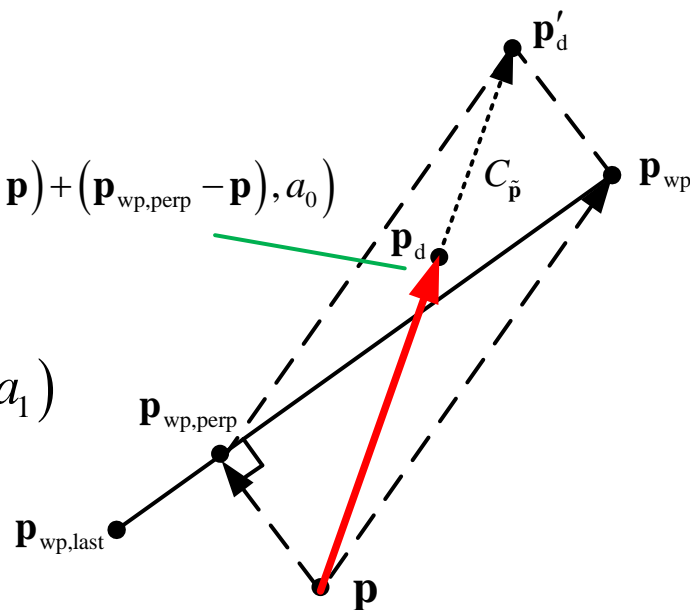


图13.11 直线路径算法跟随物理意义

令 $1/k'_1 = k_0 = k_1 = 1$ ，则 \mathbf{p}_d 的物理意义如右图所示。



全自主控制

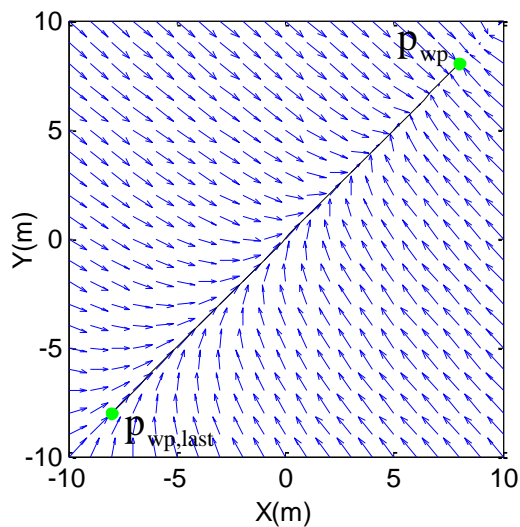
□ 路径规划

(2) 直线路径跟随

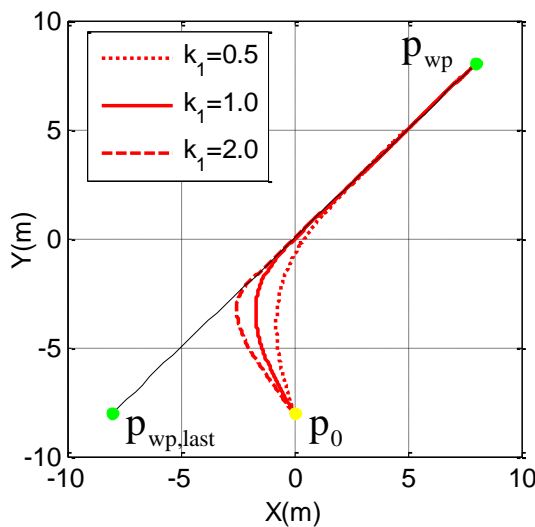
3) 算法仿真

$$\mathbf{p}_d = \mathbf{p} + \frac{1}{k_1'} \text{sat}_{\text{gd}} \left(k_0 (\mathbf{p}_{\text{wp}} - \mathbf{p}) + \boxed{k_1} (\mathbf{p}_{\text{wp,perp}} - \mathbf{p}), a_0 \right)$$

k1 越大，多旋翼靠近到直线航路的速度越快！



(a) 人工势场



(b) 从 \mathbf{p}_0 出发在 \mathbf{p}_{wp} 结束的三条飞行路径

图13.12 直线路径跟随算法仿真结果



全自主控制

□ 路径规划

(3) 避障

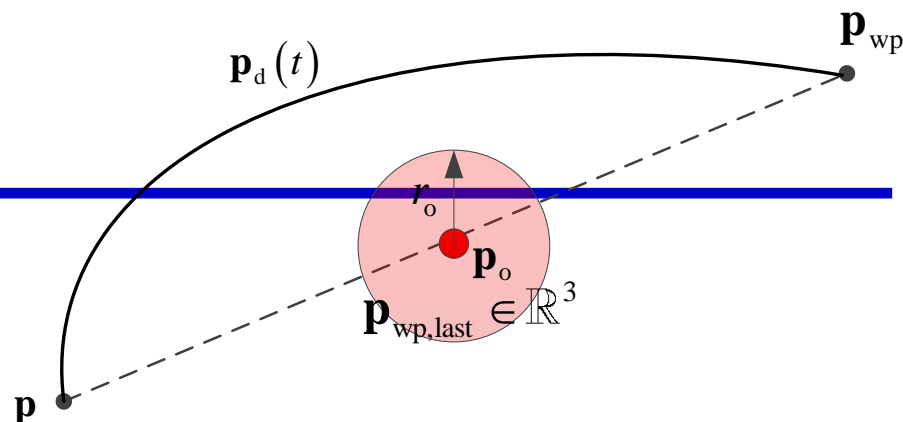


图13.13 避障示意图

1) 问题描述 令 $p \in \mathbb{R}^3$ 表示多旋翼当前位置, $p_{wp} \in \mathbb{R}^3$ 表示目标航路点。设计局部路径规划器实时地生成期望位置 $p_d \in \mathbb{R}^3$, 引导多旋翼飞向目标航路点 $p_{wp} \in \mathbb{R}^3$ 并避开半径为 $r_o \in \mathbb{R}_+$, 球心位于 $p_o \in \mathbb{R}^3$ 处的障碍物。

2) 两个假设

假设1: 多旋翼的初始位置 $p_o \in \mathbb{R}^3$ 满足 $(p_o - p_o)^T (p_o - p_o) - r_o^2 > 0$;

假设2: 目标航路点 p_{wp} 满足 $\frac{1}{(p_o - p_{wp})^T (p_o - p_{wp}) - r_o^2} \approx 0$ 。

1. 飞行器刚开始不在障碍区域内;
2. 目标位置离障碍足够远。



全自主控制

□ 路径规划

(3) 避障

3) 算法设计

定义 $\tilde{\mathbf{p}}_o = \mathbf{p} - \mathbf{p}_o$ ，考虑到饱和处理，定义函数

$$V_2 = \underbrace{k_0 \int_{C_{\tilde{\mathbf{p}}}} \text{sat}_{\text{gd}}(\tilde{\mathbf{p}}_{\text{wp}}, a_0)^T d\tilde{\mathbf{p}}_{\text{wp}}}_{\text{靠近目标航路点}} + \underbrace{\frac{k_1}{2} \frac{1}{\tilde{\mathbf{p}}_o^T \tilde{\mathbf{p}}_o - r_o^2}}_{\text{避障}} + \frac{k_2}{2} \mathbf{v}^T \mathbf{v}$$

根据假设1，“避障”项在初始时刻是有界的，如果函数一直有界，
则 $(\mathbf{p}_o - \mathbf{p})^T (\mathbf{p}_o - \mathbf{p}) > r_o^2$ ，即多旋翼不会与障碍物发生碰撞。



全自主控制

□ 路径规划

(3) 避障

3) 算法设计

对 V_2 求一阶导数得 $\dot{V}_2 = k_0 \text{sat}_{\text{gd}}(\tilde{\mathbf{p}}_{\text{wp}}, a_0)^T \mathbf{v} - \frac{k_1}{(\tilde{\mathbf{p}}_o^T \tilde{\mathbf{p}}_o - r_o^2)^2} \tilde{\mathbf{p}}_o^T \mathbf{v} + k_2 \mathbf{v}^T \mathbf{u}$

如果虚拟控制量

$$\mathbf{u} = -\frac{k_0}{k_2} \text{sat}_{\text{gd}}(\tilde{\mathbf{p}}_{\text{wp}}, a_0) + \frac{k_1}{k_2 (\tilde{\mathbf{p}}_o^T \tilde{\mathbf{p}}_o - r_o^2)^2} \tilde{\mathbf{p}}_o - \frac{1}{k_2} \text{sat}_{\text{gd}}(\mathbf{v}, a_1)$$

那么 $\dot{V}_2 = -\mathbf{v}^T \text{sat}_{\text{gd}}(\mathbf{v}, a_1) \leq 0$ 。由于 $V_2(0) > 0$ 且 $\dot{V}_2 \leq 0$ ，函数满

足 $V_2(t) \leq V_2(0), t \in [0, \infty)$ ，这意味着 $0 < \frac{k_1}{2V_2(0)} \leq \tilde{\mathbf{p}}_o^T \tilde{\mathbf{p}}_o - r_o^2$ ，即，在虚拟控制量 \mathbf{u} 的控制下，多旋翼能避开障碍物。



全自主控制

□ 路径规划

(3) 避障

3) 算法设计

$$\mathbf{u} = -\frac{k_0}{k_2} \text{sat}_{\text{gd}}(\tilde{\mathbf{p}}_{\text{wp}}, a_0) + \frac{k_1}{k_2 (\tilde{\mathbf{p}}_o^T \tilde{\mathbf{p}}_o - r_o^2)^2} \tilde{\mathbf{p}}_o - \frac{1}{k_2} \text{sat}_{\text{gd}}(\mathbf{v}, a_1)$$

当且仅当 $\mathbf{v} = \mathbf{0}_{3 \times 1}$ 时 $\dot{V}_2 = 0$ 。 $\mathbf{v} = \mathbf{0}_{3 \times 1}$ 意味着 $\mathbf{u} = \mathbf{0}$ 进一步

$$a\tilde{\mathbf{p}}_{\text{wp}} + b\tilde{\mathbf{p}}_o = \mathbf{0} \quad (2)$$

其中

$$a = -\frac{k_0}{k_2} \kappa_{a_0}(\tilde{\mathbf{p}}_{\text{wp}}), b = \frac{k_1}{k_2 (\tilde{\mathbf{p}}_o^T \tilde{\mathbf{p}}_o - r_o^2)^2}, \kappa_a(\mathbf{x}) = \begin{cases} 1 & \|\mathbf{x}\|_\infty \leq a \\ \frac{a}{\|\mathbf{x}\|_\infty} & \|\mathbf{x}\|_\infty > a \end{cases}$$

根据不变集原理，系统将全局收敛到 (\mathbf{p}, \mathbf{v}) ，其中 $\mathbf{v} = \mathbf{0}_{3 \times 1}$ ， \mathbf{p} 是方程 (2) 的解。参数 k_0, k_1 可以调整多旋翼靠近到目标航路点和避障的相对快慢。



全自主控制

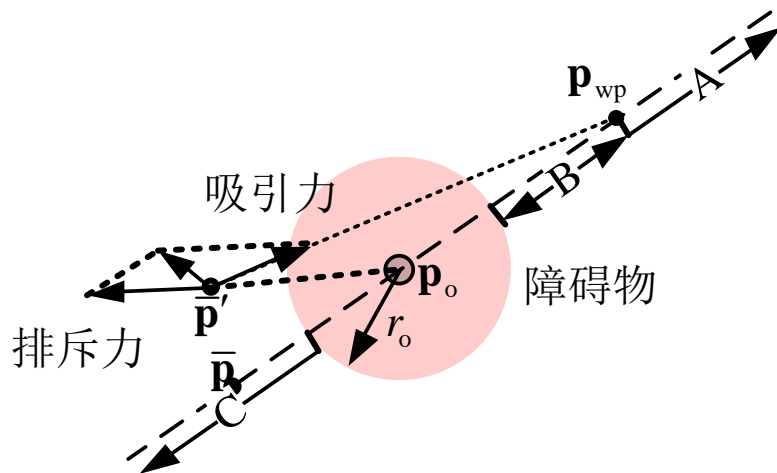
□ 路径规划

(3) 避障

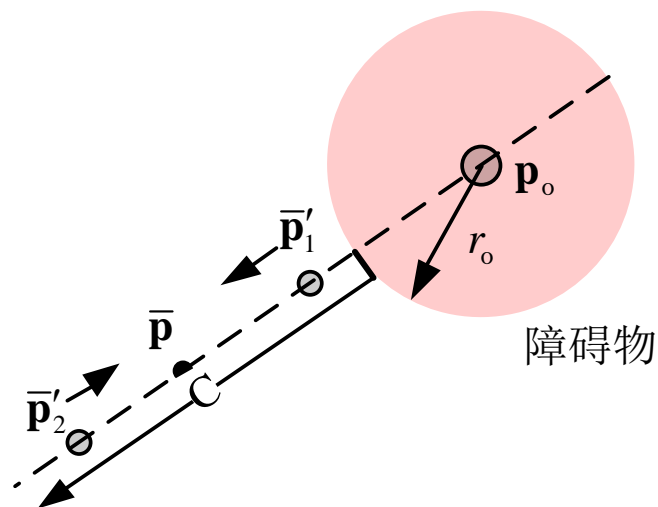
$$a\tilde{\mathbf{p}}_{wp} + b\tilde{\mathbf{p}}_o = 0 \quad (2)$$

$\bar{\mathbf{p}}$ 是不稳定的平衡点

\mathbf{p}_{wp} 是唯一稳定的平衡点



(a) 多旋翼不在“射线C”上运动



(b) 多旋翼在“射线C”上运动

图13.14 平衡点示意图



□ 路径规划

(3) 避障

3) 算法设计



图13.15 避障算法物理意义

把虚拟控制量写成PD控制器的形式

$$\mathbf{u} = -\frac{k_1'}{k_2}(\mathbf{p} - \mathbf{p}_d) - \frac{1}{k_2} \text{sat}_{\text{gd}}(\mathbf{v}, a_1)$$

其中

$$\mathbf{p}_d = \mathbf{p} + \frac{k_0}{k_1'} \text{sat}_{\text{gd}}(\mathbf{p}_{\text{wp}} - \mathbf{p}, a_0) - \frac{k_1}{k_1'} \frac{1}{\left((\mathbf{p} - \mathbf{p}_o)^T (\mathbf{p} - \mathbf{p}_o) - r_o^2 \right)^2} (\mathbf{p}_o - \mathbf{p})$$

令 $k_0/k_1' = k_1/k_1' = 1$, 则 \mathbf{p}_d 的物理意义如图 (b) 所示。



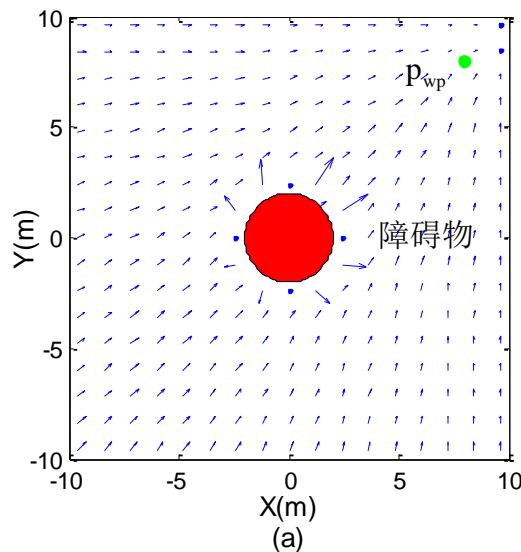
全自主控制

□ 路径规划

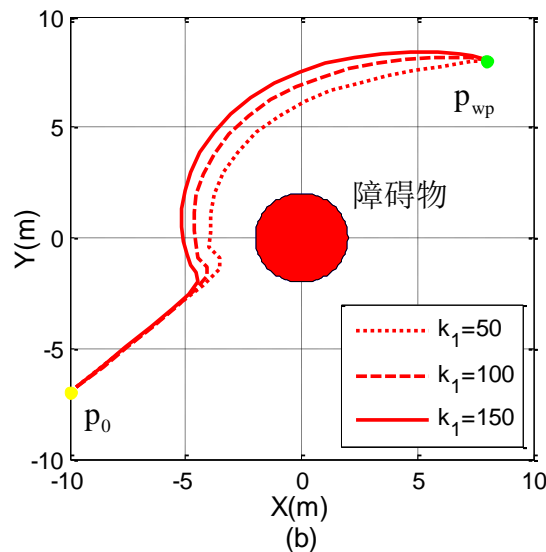
(3) 避障

4) 算法仿真

k_1 越大，多旋翼越早做出避障行为！



(a) 人工势场



(b) 从 P_0 出发在 P_{wp} 结束的3条避障路径

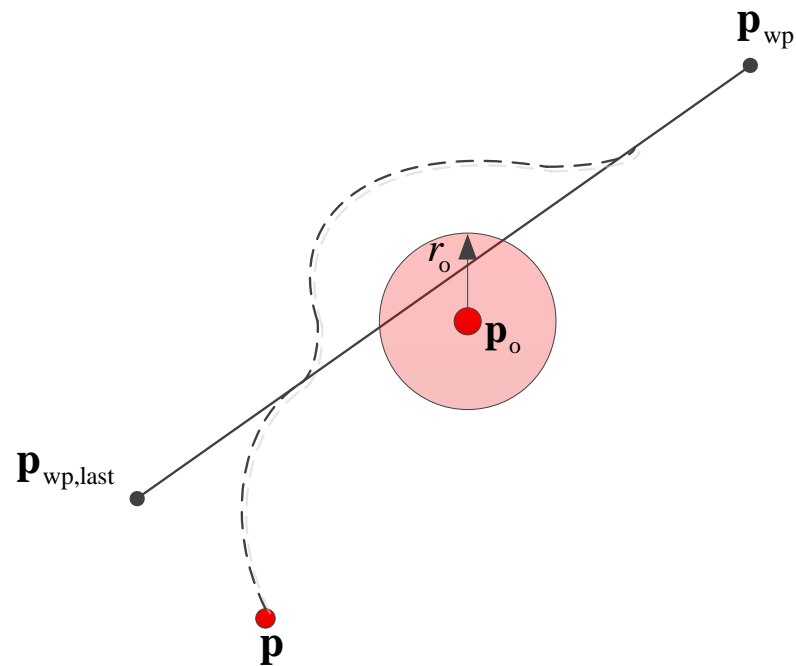
图13.16 避障算法仿真结果



全自主控制

□ 综合

实际的任务中需要多旋翼沿直线飞行，遇到障碍时还能避开。通过以上讲解，我们可以通过**叠加**直线路径、目标位置和障碍物的**势场**来设计算法。具体可以参考本门课程给定的参考书。



综合算法示意图



半自主控制

□ 半自主控制的三种模态

通常，根据自驾仪自主控制的程度，把处于半自主控制下的多旋翼分为3种模态：

- 自稳定模态（Stabilize Mode）
- 高度保持模态（Altitude Hold Mode）
- 定点模态（Loiter Mode）



半自主控制

□ 半自主控制的三种模态

通常，根据自驾仪自主控制的程度，把处于半自主控制下的多旋翼分为3种模态：

- 自稳定模态 (Stabilize Mode)
- 高度保持模态 (Altitude Hold Mode)
- 定点模态 (Loiter Mode)

- 在遥控 (RC) 模式下飞控手可以利用遥控器的滚转/俯仰摇杆控制多旋翼的滚转/俯仰角，从而控制其前后左右飞行，此时，如果不同时调整升降摇杆，多旋翼的高度会改变，若想使多旋翼保持定点悬停，飞控手需要不断调整遥控器的所有摇杆。
- 当飞控手释放摇杆时，多旋翼自动进入自动 (AC) 模式，多旋翼会自动保持自身水平，但是位置会漂移。此外，飞控手可以利用遥控器的偏航摇杆控制多旋翼的偏航角。当飞控手释放偏航摇杆时，多旋翼会保持当前机头方向。



半自主控制

□ 半自主控制的三种模式

通常，根据自驾仪自主控制的程度，把处于半自主控制下的多旋翼分为3种模式：

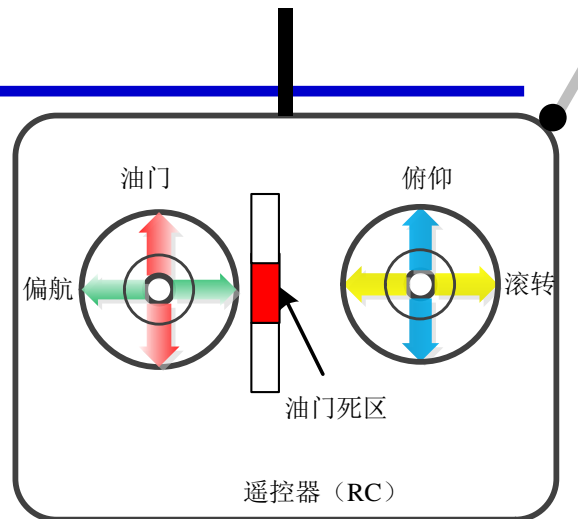


图13.17 遥控器摇杆示意图

- 自稳定模式 (Stabilize Mode)
- 高度保持模式 (Altitude Hold Mode)
- 定点模式 (Loiter Mode)

- 如图所示，当油门摇杆位于死区范围时，多旋翼自动进入**自动模式**，然后油门被自动地调整来保持当前的高度，飞控手需要不断地调整遥控器的滚转/俯仰摇杆保持悬停。
- 一旦油门摇杆离开死区范围，多旋翼将进入**遥控模式**，也就是说此时多旋翼是上升还是下降取决于油门摇杆朝哪个方向偏移。高度保持模式需要高度传感器的支持才能实现，例如气压计、超声波测距仪等。



半自主控制

□ 半自主控制的三种模态

通常，根据自驾仪自主控制的程度，把处于半自主

控制下的多旋翼分为3种模态：

- 自稳定模态 (Stabilize Mode)
- 高度保持模态 (Altitude Hold Mode)
- 定点模态 (Loiter Mode)

- 定点模态下多旋翼能自动保持当前的水平位置、航向和高度。当飞控手释放滚转/俯仰摇杆，偏航摇杆，并将油门摇杆推到死区范围时，多旋翼自动进入**自动模式**，并保持当前的水平位置、航向和高度。高精度的GPS位置信息、磁罗盘信息以及机身低振动对获得良好的定点效果非常重要。
- 飞控手可以通过推动摇杆使多旋翼进入**遥控模式**来实现对多旋翼水平位置、航向和高度的控制。定点模态需要测高仪器和位置传感器的支持才能实现，例如GPS和摄像机等。



半自主控制

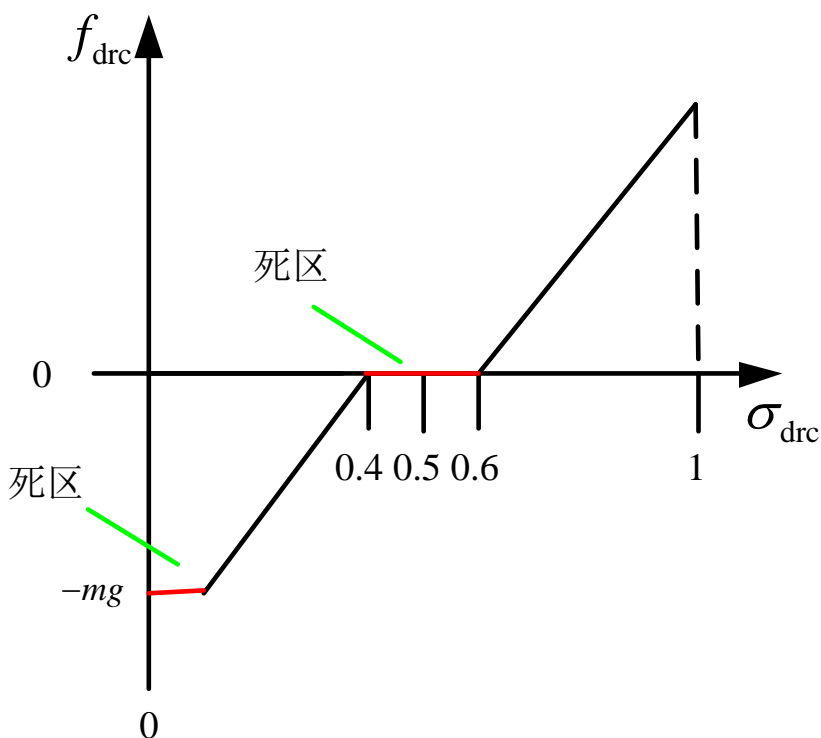
□ 遥控模式

定义 $(\cdot)_d, (\cdot)_{\text{drc}}$ 分别表示期望值和遥控器输出值。在半自主控制下，油门/偏航摇杆和滚转/俯仰摇杆分别用来输出期望总拉力 $f_{\text{drc}} = u_T$ ，期望偏航角速率 $\dot{\psi}_{\text{drc}} = u_{\omega_z}$ ，期望滚转角 $\phi_{\text{drc}} = u_{\phi}$ 和期望俯仰角 $\theta_{\text{drc}} = u_{\theta}$ 。在直接式油门中，遥控器输出的总拉力指令与油门摇杆的偏移量成正比。此外，一些多旋翼产品，例如大疆的小精灵，采用增量式油门，这种油门有两个特点：①飞控手释放油门摇杆时油门摇杆能自动弹回死区位置；②油门摇杆的偏移量与期望的垂直方向速率或者拉力变化率成正比。



半自主控制

□ 遥控模式



这里, $\sigma_{drc} \in [0,1]$ 是油门摇杆的偏移量, 其中 $\sigma_{drc} = 0.5$ 表示油门摇杆的中间位置。如果 $\sigma_{drc} \in [0.4, 0.6]$, 那么 $f_{drc}(\sigma_{drc}) = 0$, 然后自动控制器开始引入高度反馈进行高度保持。在这里, 前馈控制量 mg 由自动控制器给出, 所以, 如图所示, f_{drc} 从 $-mg$ 开始。死区用来消除油门摇杆微弱变化带来的影响。

图13.18 油门摇杆偏移量与油门控制量的关系



半自主控制

□ 自动控制模式

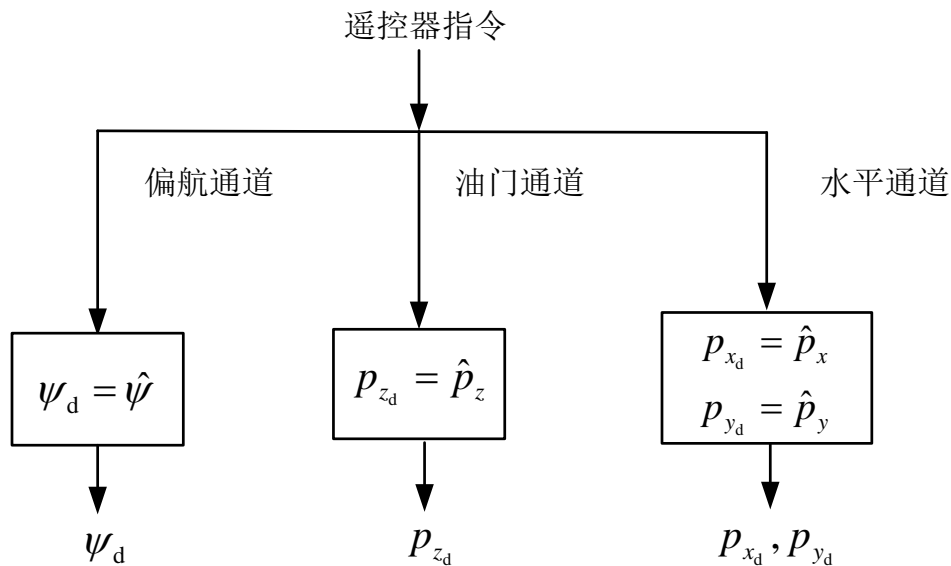
当所有摇杆都处在中间位置的时候，自驾仪的自动控制（AC）将接管多旋翼的控制。事实上，无论多旋翼处于自稳定模态、高度保持模态还是定点模态，任务决策模块都会产生期望姿态或者期望位置 \mathbf{p}_{dac} 和偏航角 ψ_{dac} ，状态估计模块都会产生位置估计量 $\hat{\mathbf{p}} = [\hat{p}_x \ \hat{p}_y \ \hat{p}_z]^T$ 和姿态估计量 $\hat{\Theta} = [\hat{\phi} \ \hat{\theta} \ \hat{\psi}]^T$ ，自动模式的任务都是控制多旋翼飞行使得 $\lim_{t \rightarrow \infty} \|\hat{\mathbf{p}}(t) - \mathbf{p}_{\text{dac}}(t)\| = 0$ 和 $\lim_{t \rightarrow \infty} \|\hat{\psi}(t) - \psi_{\text{dac}}(t)\| = 0$ 。因此，**对于3种模态来说AC控制的结构是一样的**。然而，由于3种模态采用不同的传感器，导航信息的估计精度会有所不同。



半自主控制

□ 自动控制模式

(1) 自稳定模态 (Stabilize Mode)



自稳定模态根据期望位置 $\mathbf{p}_d = \hat{\mathbf{p}}$ 和期望偏航角 $\psi_d = \dot{\psi}$ 生成期望总拉力和力矩。如图所示，自驾仪根据遥控器指令生成期望位置 \mathbf{p}_d 和期望偏航角 ψ_d 。

图13.19 自稳定模态下期望位置和偏航角的生成原理



半自主控制

□ 自动控制模式

(1) 自稳定模态 (Stabilize Mode)

回想第十一讲介绍的控制器，水平通道控制器为

$$\Theta_{hd} = -g^{-1} \mathbf{A}_{\psi}^{-1} \left(-\mathbf{K}_{phd} \dot{\hat{\mathbf{p}}}_h - \mathbf{K}_{php} (\hat{\mathbf{p}}_h - \hat{\mathbf{p}}_{hd}) \right)$$

高度通道控制器为

$$f_d = -m \left(-k_{p_z d} \dot{\hat{p}}_z - k_{p_z p} (\hat{p}_z - p_{z_d}) \right)$$

由于 $\mathbf{p}_d = \hat{\mathbf{p}}$ ，水平通道控制器和高度通道控制器变为

$$\Theta_{hd} = g^{-1} \mathbf{A}_{\psi}^{-1} \mathbf{K}_{phd} \dot{\hat{\mathbf{p}}}_h$$

$$f_d = mg + mk_{p_z d} \dot{\hat{p}}_z$$



半自主控制

□ 自动控制模式

(1) 自稳定模态 (Stabilize Mode)

由于 $\dot{\hat{\mathbf{p}}}_h$ 不可获取或者不够精确或者磁力计不可用，通常令 $\Theta_{hd} = \mathbf{0}_{2 \times 1}$

这意味着自动控制器能使多旋翼保持水平。此外，自动控制器并不能保持高度。这是因为控制器只是把高度通道的速度控制到零。

对于偏航通道，一个简单的控制器具有以下形式

$$\tau_z = -k_{\dot{\psi}} \dot{\psi}$$

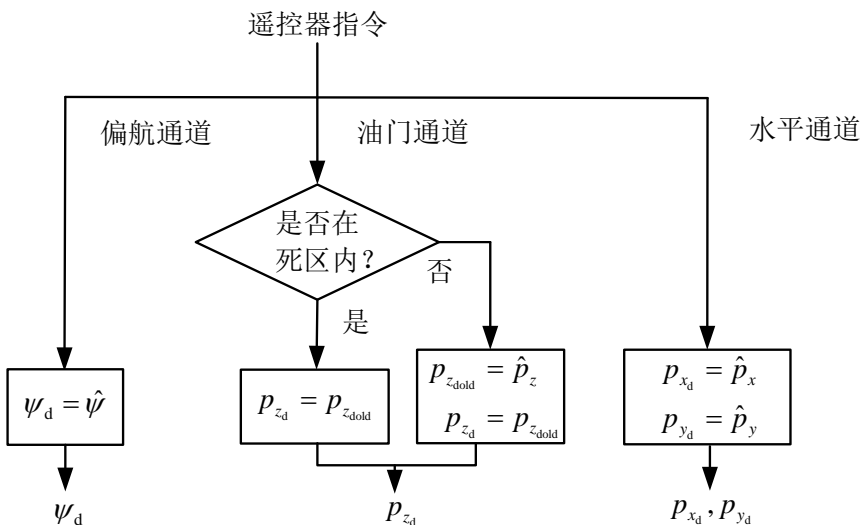
由于缺少位置信息反馈，自稳定模态不能使多旋翼稳定悬停。这个模态通常在没有GPS和高度测量传感器或者这些传感器失效的时候使用。



半自主控制

□ 自动控制模式

(2) 高度保持模态 (Altitude Hold Mode)



高度保持模态根据期望高度 $p_{z_d} = p_{z_{dold}}$,

期望水平位置 $\mathbf{p}_{hd} = \hat{\mathbf{p}}_h$ 和期望偏航角

$\psi_d = \hat{\psi}$ 生成期望总拉力和力矩。其中

, $\mathbf{p}_{hd} = \hat{\mathbf{p}}_h$ 意味着 $\theta_d = \phi_d = 0$ 。

图13.20 高度保持模态下期望位置和偏航角的生成原理



半自主控制

□ 自动控制模式

(2) 高度保持模态 (Altitude Hold Mode)

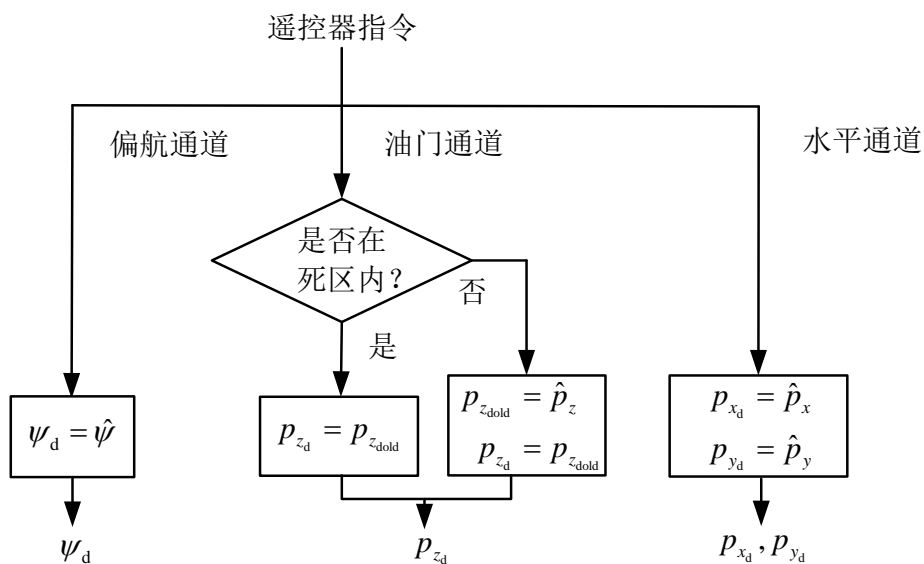


图13.21 高度保持模态下期望位置和偏航角的生成原理

如图所示，将油门摇杆回到中间位置的时刻记为 t_{zd} ，将该时刻的高度估计量记为 $p_{zdold} = \hat{p}_z(t_{zd})$ 。与此同时，高度保持模态将多旋翼的高度保持在 $p_{zd} = p_{zdold}$ 。与自稳定模态一样，由于缺少水平位置信息反馈，高度保持模态也不能使多旋翼稳定悬停。高度保持模态通常在高度测量传感器可用，而缺少位置传感器或电子罗盘不可用的时候使用。



半自主控制

□ 自动控制模式

(3) 定点模态 (Loiter

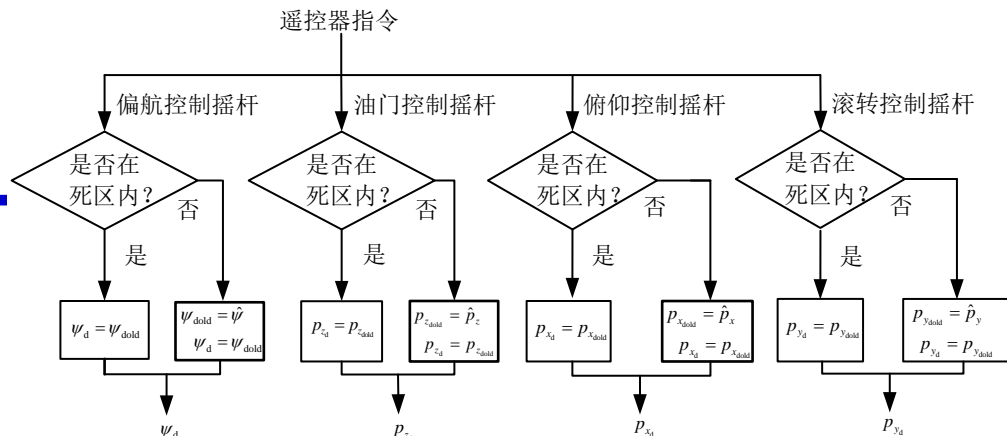


图13.22 定点模态下期望位置和期望偏航角的生成原理

如图所示，定点模态根据期望的位置 $\mathbf{p}_d = \mathbf{p}_{dold}$ 和偏航角 $\psi_d = \psi_{dold}$ 生成期望总拉力和力矩。将4个摇杆回到中间位置的时刻分别记为 $t_{\psi_d}, t_{z_d}, t_{x_d}, t_{y_d}$ ，将这些时刻的估计量分别记为

$$\psi_{dold} = \hat{\psi}(t_{\psi_d})$$

$$p_{z_{dold}} = \hat{p}_z(t_{z_d})$$

$$p_{x_{dold}} = \hat{p}_x(t_{x_d})$$

$$p_{y_{dold}} = \hat{p}_y(t_{y_d})$$

自主控制器将控制多旋翼悬停在 $\mathbf{p}_d = \mathbf{p}_{dold}$ 并保持偏航角为 $\psi_d = \psi_{dold}$ 。定点模态通常在高度测量传感器和位置传感器都可用的时候使用。



半自主控制

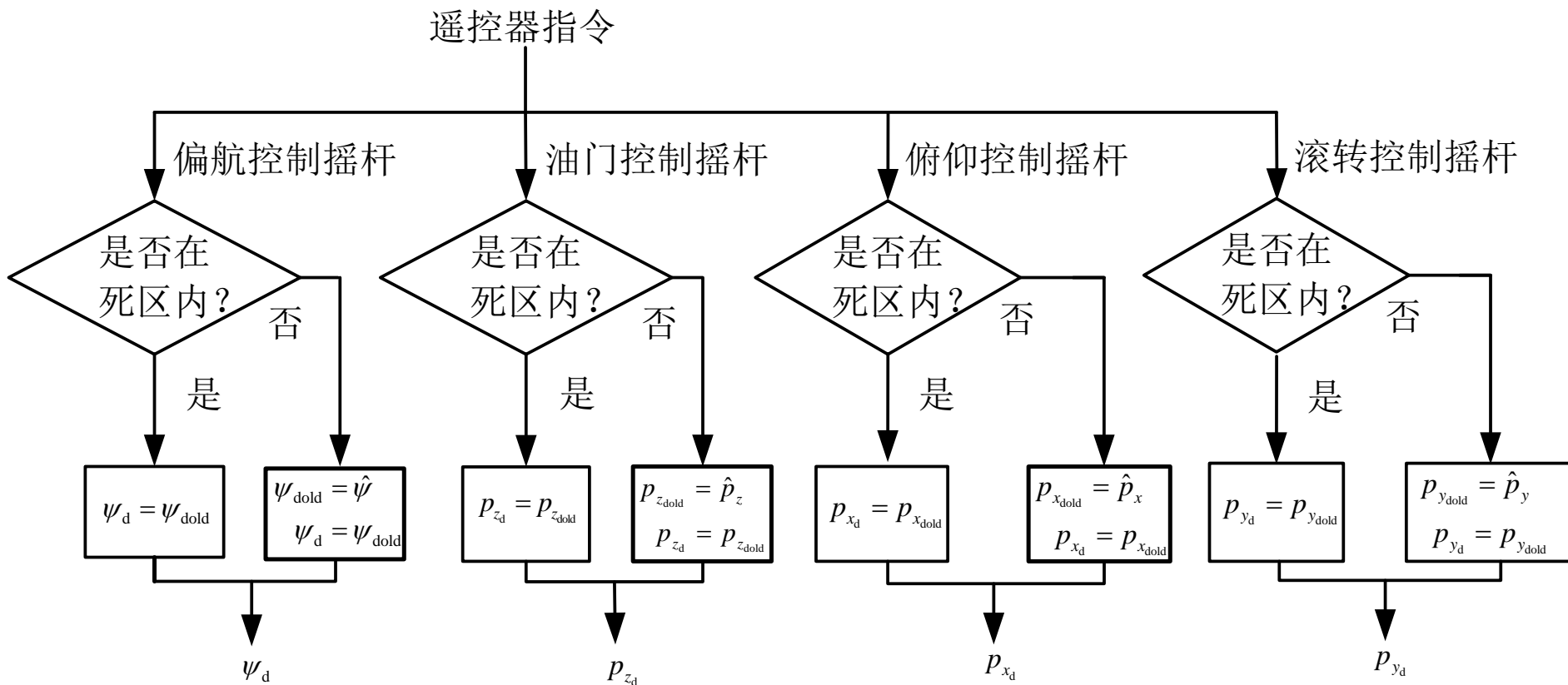


图13.22 定点模态下期望位置和期望偏航角的生成原理



半自主控制

□ 遥控与自主控制间的切换逻辑

(1) 偏航指令的切换逻辑

$$\psi_d = \psi_{dap} + \psi_{drc}$$

$\psi_{dap} = \hat{\psi}$ $\psi_{drc} = \dot{\psi}_{drc} \Delta t$
遥控器的指令

(2) 油门指令的切换逻辑

$$f_d = f_{dap} + f_{drc}(\sigma_{drc})$$

自主控制器的指令 遥控器的指令

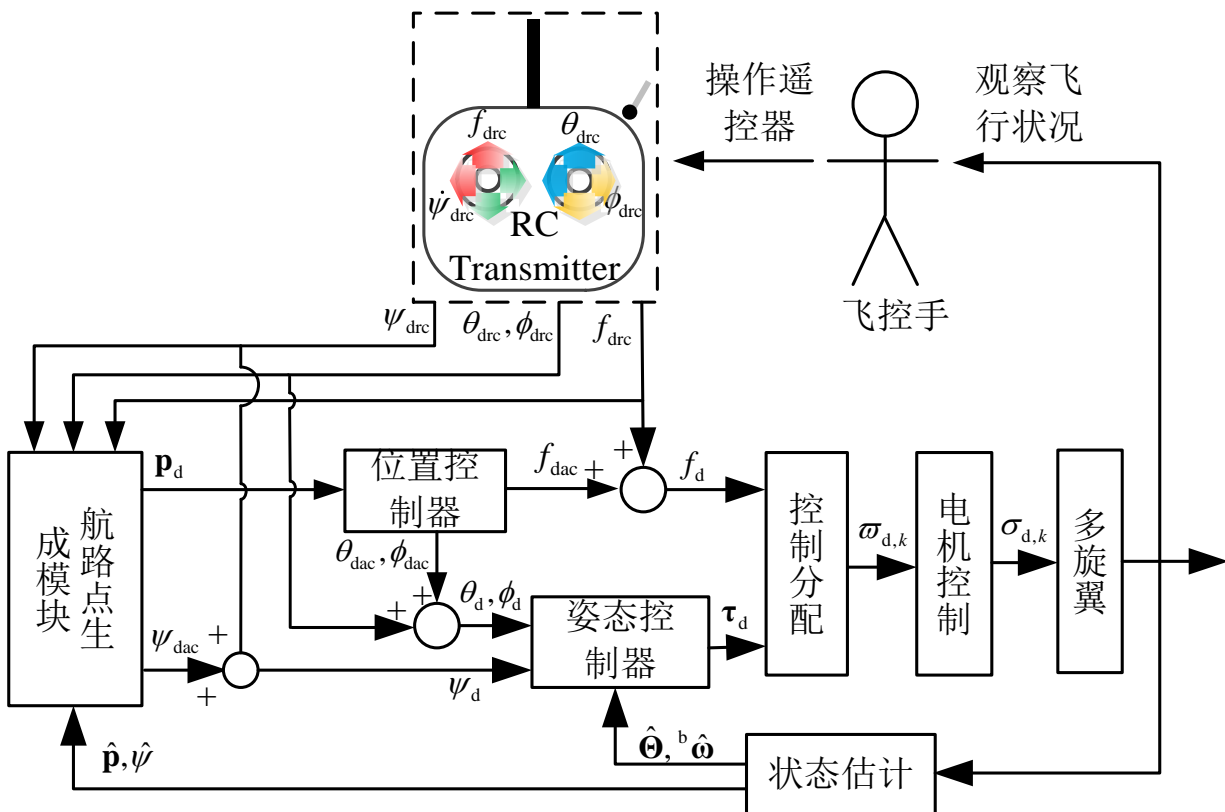


图13.23 半自主控制闭环框图



半自主控制

□ 遥控与自主控制间的切换逻辑

(3) 滚转/俯仰指令的切换逻辑

当滚转/俯仰摇杆处于中间位置附近时，多旋翼切换到AC模式，

(p_{x_d}, p_{y_d}) 将保持常值，AC控制器将控制多旋翼保持在水平位置

$p_{x_d} = p_{x_{dold}} \left(p_{y_d} = p_{y_{dold}} \right)$ ，滚转/俯仰摇杆离开中间位置，则多旋翼切换到RC模式， $(p_{x_{dold}}, p_{y_{dold}})$ 将连续地更新，即 $p_{x_{dold}} = \hat{p}_x \left(p_{y_{dold}} = \hat{p}_y \right)$ 。滚转/俯仰指令的切换逻辑可以表示为

$$\theta_d = \theta_{dap} + \theta_{drc}$$

$$\phi_d = \phi_{dap} + \phi_{drc}$$

当 $\theta_{drc}(\phi_{drc})$ 都非零的时候 $\theta_{dap}(\phi_{dap})$ 应该要等于零。当 $\theta_{drc}(\phi_{drc})$ 等于零的时候 $\theta_{dap}(\phi_{dap})$ 的值由自主控制器确定。



本讲小结

1. 单架多旋翼的任务规划或者自主飞行控制相对简单。
2. 多架多旋翼的协同工作可以更高效地完成给定的任务，近年来，多架多旋翼的协同控制在可控实验环境下已经得以实现。
3. 未来要实现多旋翼的全自主任务规划，必须具备以下条件：
 - (1) 每一架多旋翼都要足够可靠。
 - (2) 位置和姿态的估计要十分精确。
 - (3) 要具有足够长的续航时间，且要装配电池管理系统（BMS）评估电池的健康。
 - (4) 要修建具有自动充电功能的快速充电站。



资源

(1) 可靠飞行控制研究组主页课程中心(全部课件下载)

<http://rfly.buaa.edu.cn/course>

(2) 关注可靠飞行控制研究组公众号 buaarfly(文章、资讯等)

(3) 多旋翼设计与控制交流QQ群:183613048

(4) 视频课程(MOOC)同步发布, 网易云课堂搜索 “多旋翼”

<http://study.163.com/course/introduction/1003715005.htm>

(5) 同名中文书本教材《多旋翼飞行器设计与控制》即将在电子工业出版社出版, 敬请期待

(6) 有疑问可联系课程总助教戴训华, 邮箱: dai@buaa.edu.cn

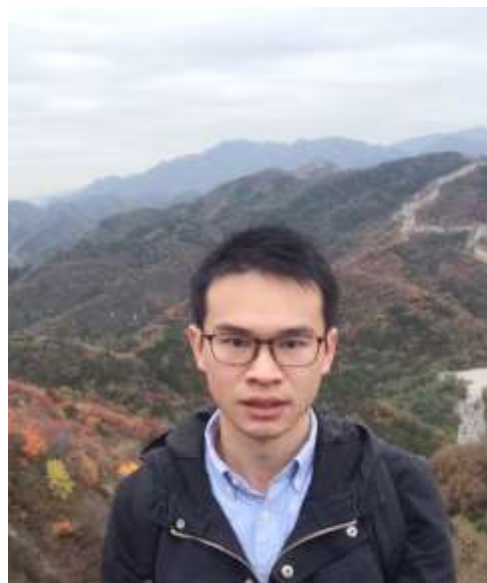


致谢

感谢控制组同学



颜江



戴训华

为本节课程准备作出的贡献。



谢谢！