



多旋翼飞行器设计与控制

第十一讲 多旋翼的底层飞行控制

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

2016年5月19日 北航主南401



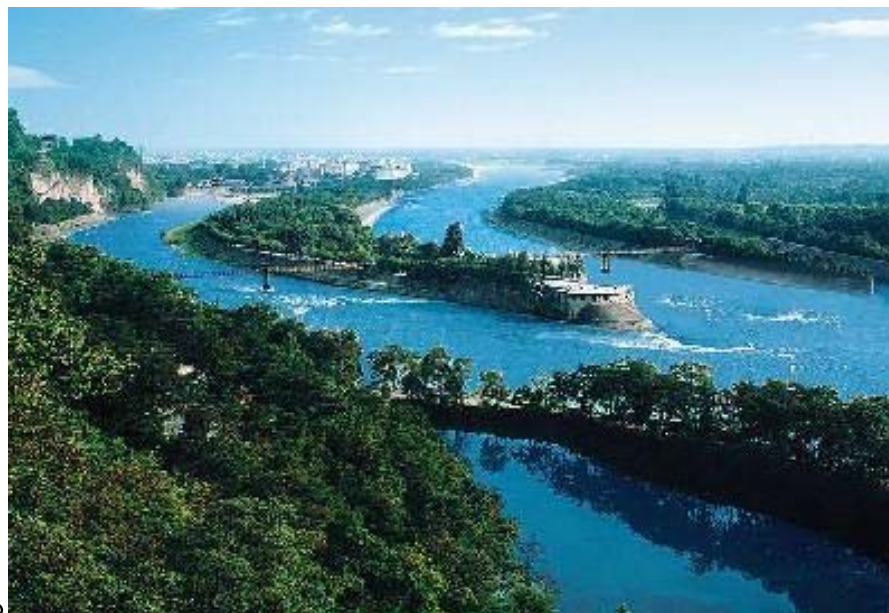
北航可靠飞行控制研究组

BUAA Reliable Flight Control Group



前言

东方哲理：中国古人很早就开始利用自动控制思想了。都江堰位于四川省成都市都江堰市城西，坐落在成都平原西部的岷江上，始建于秦昭王末年(约公元前256~前251)，是蜀郡太守李冰父子在前人鳖灵开凿的基础上组织修建的大型水利工程，两千多年来一直发挥着防洪灌溉的作用，是全世界迄今为止，年代最久、唯一留存、仍在一直使用，以无坝引水为特征的宏大水利工程。它由分水鱼嘴、飞沙堰、宝瓶口等部分组成，充分利用了大自然的原理，实现了**自动分流，自动排沙，控制进水流量**，在两千多年前便完成了自动化。



都江堰，图片来源 <http://baike.sogou.com/v90370.htm>



前言

给定期望的位置，如何控制多旋翼的电机使其能够飞到目标位置？



大纲

1. 多旋翼底层控制框架和模型化简
2. 位置控制
3. 姿态控制
4. 控制分配
5. 电机控制
6. 综合仿真
7. 飞控原理与调参
8. 总结
9. 作业



1. 多旋翼底层控制框架和模型化简

□ 底层控制框架

多旋翼的底层控制可以分为四个层次，分别为位置控制、姿态控制、控制分配和电机控制

位置控制：期望的三维位置 \mathbf{p}_d \rightarrow 解算期望姿态角 Θ_d (滚转、俯仰和偏航 ϕ_d, θ_d, ψ_d) 以及期望总拉力 f_d

姿态控制：期望姿态角 Θ_d \rightarrow 解算期望力矩 τ_d

控制分配：期望力矩和升力 τ_d, f_d \rightarrow 解算 n 个电机的期望转速 $\omega_{d,k}$, $k=1, 2, \dots, n$

电机控制：期望转速 $\omega_{d,k}$ \rightarrow 解算电机输入油门 (电调PWM占空比) $\sigma_{d,k}$, $k=1, 2, \dots, n$

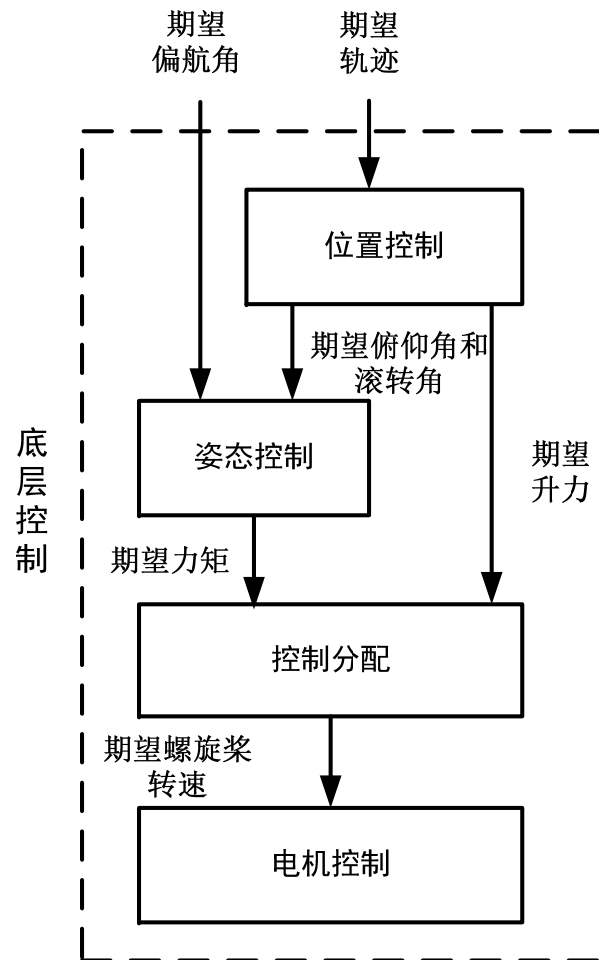


图.多旋翼底层控制层次



1. 多旋翼底层控制框架和模型化简

□ 底层控制框架

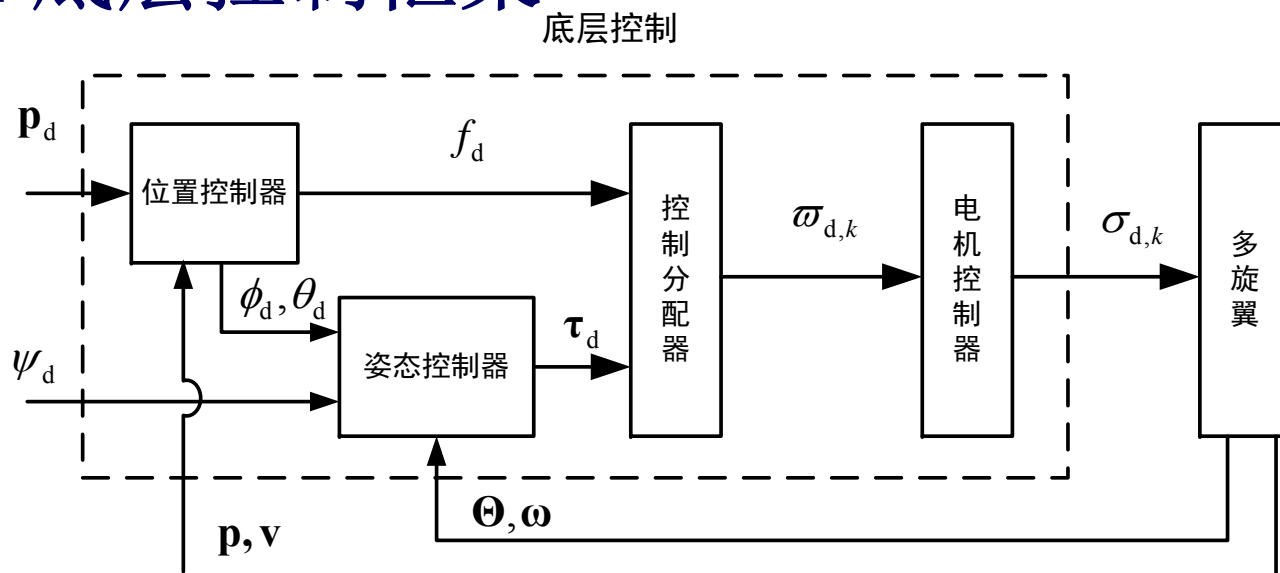


图. 多旋翼全自主控制闭环框图

欠驱动系统：4个输入（总拉力 f 和三轴力矩 τ ）控制6个输出（位置 p 和姿态角 Θ ）。设计多旋翼飞行控制器时，可以采用**内外环的控制策略**，其中**内环对多旋翼飞行器姿态角进行控制**，而**外环对多旋翼飞行器的位置进行控制**。由内外环控制实现多旋翼飞行器的升降、悬停、侧飞等飞行模态。



1. 多旋翼底层控制框架和模型化简

□ 线性简化模型

基于第六章的多旋翼非线性模型，忽略 $-{}^b\boldsymbol{\omega} \times (\mathbf{J} \cdot {}^b\boldsymbol{\omega}) + \mathbf{G}_a$ ，得到简化的多旋翼模型如下：

$$\begin{aligned} {}^e\dot{\mathbf{p}} &= {}^e\mathbf{v} \\ {}^e\dot{\mathbf{v}} &= g\mathbf{e}_3 - \frac{f}{m}\mathbf{R}\mathbf{e}_3 \\ \dot{\boldsymbol{\Theta}} &= \mathbf{W} \cdot {}^b\boldsymbol{\omega} \\ \mathbf{J} \cdot {}^b\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} \end{aligned} \quad \left\} \longleftrightarrow \begin{aligned} \ddot{p}_{x_e} &= -\frac{f}{m}(\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi) \\ \ddot{p}_{y_e} &= -\frac{f}{m}(-\cos\psi \sin\phi + \cos\phi \sin\theta \sin\psi) \\ \ddot{p}_{z_e} &= g - \frac{f}{m}\cos\theta \cos\phi \end{aligned}$$

其中， ${}^e\mathbf{p} = [p_{x_e}, p_{y_e}, p_{z_e}]^T \in \mathbb{R}^3$ 为对地位置， ${}^e\mathbf{v}$ 为对地速度， ${}^b\boldsymbol{\omega}$ 为飞机机体角速度， \mathbf{R} 为旋转矩阵， \mathbf{J} 为转动惯量矩阵， $\mathbf{e}_3 = [0, 0, 1]^T$ ，其他符号定义详见第六章。



1. 多旋翼底层控制框架和模型化简

□ 线性简化模型

假设多旋翼处于悬停状态，即 $\mathbf{p} \approx \mathbf{p}_d, \theta \approx \phi \approx 0, \psi \approx \psi_d, \dot{\mathbf{p}} \approx 0, \dot{\theta} \approx \dot{\phi} \approx \dot{\psi} \approx 0$
标称输入给定为 $f \approx mg, \tau \approx 0$ 。则此时有 $\sin \phi \approx \phi, \cos \phi \approx 1, \sin \theta \approx \theta, \cos \theta \approx 1$ 。
通过在平衡点的小角度线性化，此时，

$$\mathbf{Re}_3 \approx \begin{bmatrix} \theta \cos \psi + \phi \sin \psi \\ \theta \sin \psi - \phi \cos \psi \\ 1 \end{bmatrix}$$

可得到多旋翼的线性模型：

(1) 水平通道模型

$$\begin{aligned} {}^c\dot{\mathbf{p}} &= {}^c\mathbf{v} \\ {}^c\dot{\mathbf{v}} &= g\mathbf{e}_3 - \frac{f}{m}\mathbf{Re}_3 \end{aligned} \quad \rightarrow \quad \begin{aligned} \ddot{x}_i &= -\frac{f}{m}(\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \\ \ddot{y}_i &= -\frac{f}{m}(-\cos \psi \sin \phi + \cos \phi \sin \theta \sin \psi) \end{aligned}$$

可以认为 Θ_h 是输入。
进一步，因为 $-g\mathbf{A}_\psi$ 可以得到，所以可以认为输入是 $-g\mathbf{A}_\psi\Theta_h$ 。水平通道模型为线性的。

$$\begin{aligned} \dot{\mathbf{p}}_h &= \mathbf{v}_h \\ \dot{\mathbf{v}}_h &= -g\mathbf{A}_\psi\Theta_h \end{aligned}$$

其中 $\mathbf{p}_h = \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \mathbf{R}_\psi = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix}, \mathbf{A}_\psi = \mathbf{R}_\psi \text{diag}(1, -1), \Theta_h = \begin{bmatrix} \theta \\ \phi \end{bmatrix}$



1. 多旋翼底层控制框架和模型化简

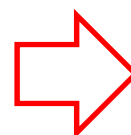
□ 线性简化模型

(2) 高度通道模型

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= g\mathbf{e}_3 - \frac{f}{m}\mathbf{R}\mathbf{e}_3\end{aligned}$$



$$\ddot{z}_i = g - \frac{f}{m}\cos\theta\cos\phi$$



线性模型

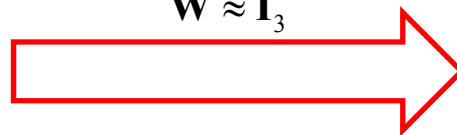
$$\begin{aligned}\dot{p}_z &= v_z \\ \dot{v}_z &= g - \frac{f}{m}\end{aligned}$$

(3) 姿态模型

$$\begin{aligned}\dot{\boldsymbol{\Theta}} &= \mathbf{W} \cdot {}^b\boldsymbol{\omega} \\ \mathbf{J} \cdot {}^b\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau}\end{aligned}$$

小角度假设

$$\mathbf{W} \approx \mathbf{I}_3$$



线性模型

$$\begin{aligned}\dot{\boldsymbol{\Theta}} &= \boldsymbol{\omega} \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau}\end{aligned}$$



1. 多旋翼底层控制框架和模型化简

□ 线性简化模型

原模型

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

$${}^e\dot{\mathbf{v}} = g\mathbf{e}_3 - \frac{f}{m}\mathbf{R}\mathbf{e}_3$$

$$\dot{\mathbf{\Theta}} = \mathbf{W} \cdot {}^b\boldsymbol{\omega}$$

$$\mathbf{J} \cdot {}^b\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}$$



(1) 水平通道模型

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

$$\dot{\mathbf{v}}_h = -g\mathbf{A}_{\psi}\mathbf{\Theta}_h$$

(2) 高度通道模型

$$\dot{p}_z = v_z$$

$$\dot{v}_z = g - \frac{f}{m}$$

(3) 姿态模型

$$\dot{\mathbf{\Theta}} = \boldsymbol{\omega}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}$$

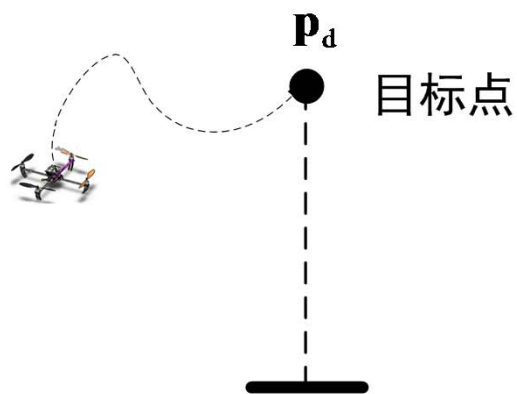
每个通道都为
线性模型！



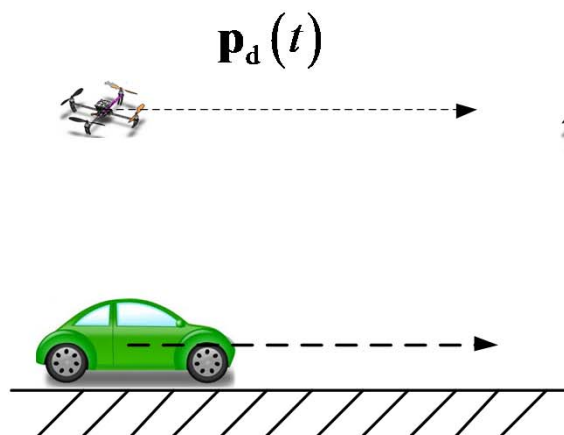
2. 位置控制

□ 分类

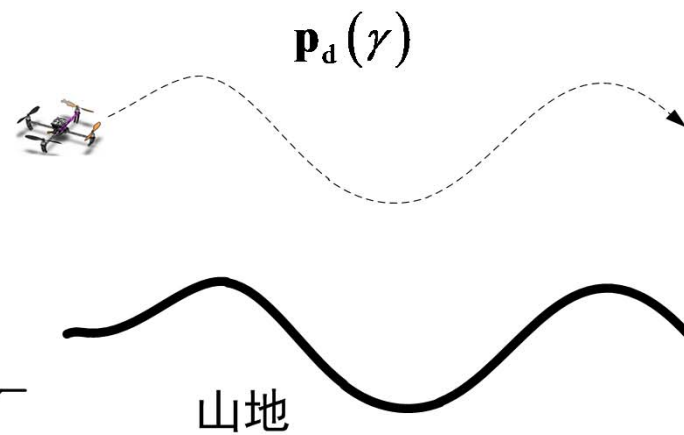
根据给定目标轨迹 \mathbf{p}_d 的不同，可将位置控制分为三类：定点控制、轨迹跟踪和路径跟随。



(a) 定点控制



(b) 轨迹跟踪



(c) 路径跟随

图. 位置控制三类形式



2. 位置控制

□ 分类

- (1) **定点控制** (Set-Point Control)。期望目标点 $\mathbf{p}_d \in \mathbb{R}^3$ 是一个常值。设计控制器使得, 当 $t \rightarrow \infty$ 时, $\|\mathbf{p}_d - \mathbf{p}(t)\|$ 收敛到0或者在0足够小的邻域内。
- (2) **轨迹跟踪** (Trajectory Tracking)。期望轨迹 $\mathbf{p}_d(t): [0, \infty) \rightarrow \mathbb{R}^3$ 是一条与时间相关的轨迹。设计控制器使得, 当 $t \rightarrow \infty$ 时, $\|\mathbf{p}_d(t) - \mathbf{p}(t)\|$ 收敛到0或者在0足够小的邻域内。
- (3) **路径跟随** (Path Following) [1]。期望轨迹 $\mathbf{p}_d(\gamma) \in \mathbb{R}^3$ 是一条由参数 γ 直接决定的路径, 而不是时间。设计控制器使得, 当 $t \rightarrow \infty$ 时, $\|\mathbf{p}_d(\gamma(t)) - \mathbf{p}(t)\|$ 收敛到0或者在0足够小的邻域内。

[1] Hespanha J P. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. Automatic Control, IEEE Transactions on, 2007, 52(8): 1362–1379.



2. 位置控制

□ 分类

根据解算出的姿态信息的不同，可将位置控制分为两类：

(1) 产生期望欧拉角的位置控制。

针对线性系统设计控制器。这种类型的位置控制器最终产生的期望值为欧拉角和拉力 ϕ_d, θ_d, f_d 。

(2) 产生期望旋转矩阵的位置控制。

直接针对非线性耦合系统设计控制器。这种类型的位置控制器最终产生的期望值为旋转矩阵和拉力 R_d, f_d 。



2. 位置控制

□ 产生期望欧拉角的位置控制

(1) 传统的PID设计

- 水平通道模型

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

$$\dot{\mathbf{v}}_h = -g\mathbf{A}_\psi \Theta_h$$

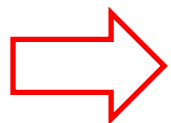
- 期望水平位置动态

$$\ddot{\mathbf{p}}_h = \ddot{\mathbf{p}}_{hd} - \mathbf{K}_{p_{hd}} (\dot{\mathbf{p}}_h - \dot{\mathbf{p}}_{hd}) - \mathbf{K}_{p_{hp}} (\mathbf{p}_h - \mathbf{p}_{hd})$$

$$-g\mathbf{A}_\psi \Theta_{hd} = \ddot{\mathbf{p}}_{hd} - \mathbf{K}_{p_{hd}} (\dot{\mathbf{p}}_h - \dot{\mathbf{p}}_{hd}) - \mathbf{K}_{p_{hp}} (\mathbf{p}_h - \mathbf{p}_{hd})$$

当考虑定点控制时

$$\dot{\mathbf{p}}_{hd} = \ddot{\mathbf{p}}_{hd} = \mathbf{0}_{2 \times 1}$$



$$\Theta_{hd} = -g^{-1} \mathbf{A}_\psi^{-1} \left(\ddot{\mathbf{p}}_{hd} - \mathbf{K}_{p_{hd}} (\dot{\mathbf{p}}_h - \dot{\mathbf{p}}_{hd}) - \mathbf{K}_{p_{hp}} (\mathbf{p}_h - \mathbf{p}_{hd}) \right)$$

其中 $\mathbf{K}_{(.)}$ 表示参数。



2. 位置控制

□ 产生期望欧拉角的位置控制

(1) 传统的PID设计

- 高度通道模型

$$\dot{p}_z = v_z$$

$$\dot{v}_z = g - \frac{f}{m}$$

- 期望高度动态

$$\ddot{p}_z = \ddot{p}_{z_d} - k_{p_z d}(\dot{p}_z - \dot{p}_{z_d}) - k_{p_z p}(p_z - p_{z_d})$$

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

当考虑定点控制时, $\dot{p}_{z_d} = \ddot{p}_{z_d} = 0$ 上式就变为

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



2. 位置控制

产生期望欧拉角的位置控制

(2) 开源自驾仪PID设计

1) 水平通道模型

为了使 $\lim_{t \rightarrow \infty} \|\mathbf{e}_{\mathbf{p}_h}(t)\| = 0$ ，我们先针对

期望速度 $\dot{\mathbf{p}}_h = \mathbf{v}_h$ \rightarrow $\mathbf{v}_{hd} = \mathbf{K}_{\mathbf{p}_h} (\mathbf{p}_{hd} - \mathbf{p}_h)$

在 $\dot{\mathbf{p}}_{hd} = 0$ 的前提下如果

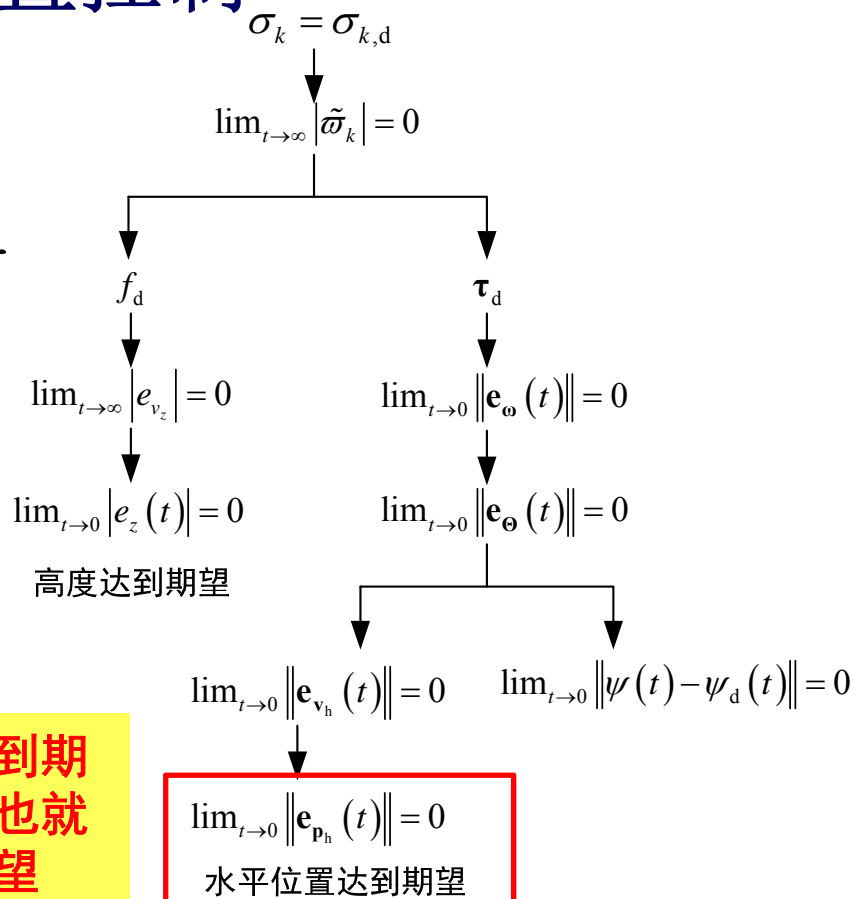
$$\lim_{t \rightarrow \infty} \|\mathbf{e}_{\mathbf{v}_h}(t)\| = 0$$

那么

$$\lim_{t \rightarrow \infty} \|\mathbf{e}_{\mathbf{p}_h}(t)\| = 0$$

其中 $\mathbf{e}_{\mathbf{v}_h} \triangleq \mathbf{v}_h - \mathbf{v}_{hd}$

速度能达到期望，位置也就能达到期望





2. 位置控制

产生期望欧拉角的位置控制

(2) 开源自驾仪PID设计

1) 水平通道模型

为了使 $\lim_{t \rightarrow \infty} \|\mathbf{e}_{v_h}(t)\| = 0$, 我们先针对

$$\dot{\mathbf{v}}_h = -g\mathbf{A}_\psi \boldsymbol{\Theta}_h$$

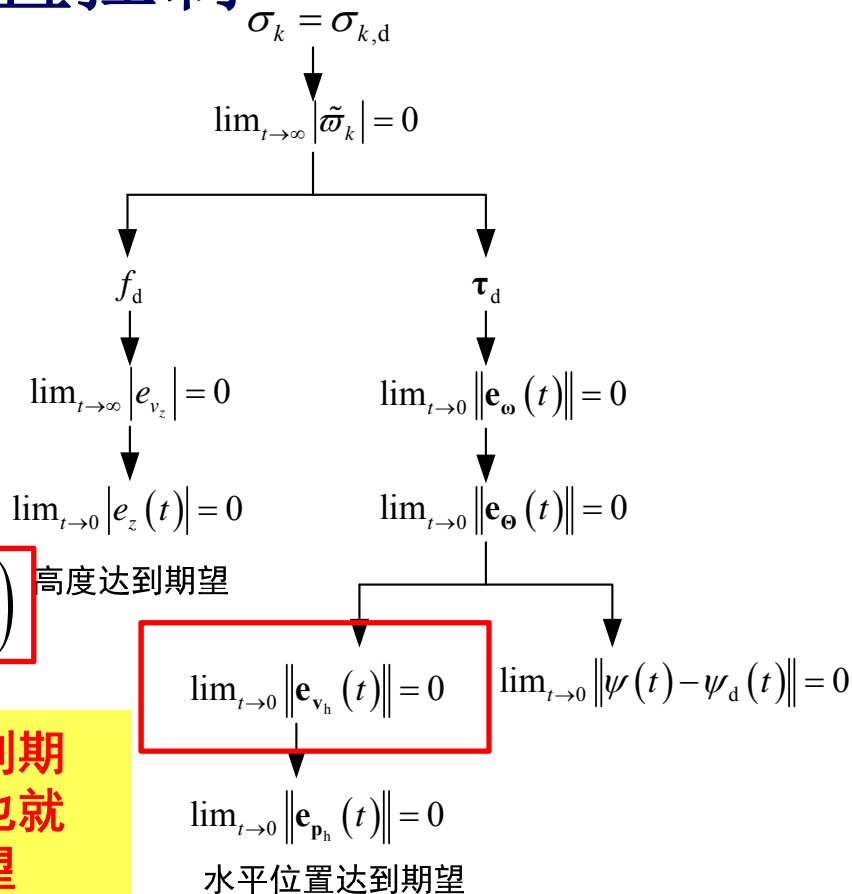
$$-g\mathbf{A}_\psi \boldsymbol{\Theta}_{hd} = -\mathbf{K}_{vhp} \mathbf{e}_{v_h} - \mathbf{K}_{vhi} \int \mathbf{e}_{v_h} - \mathbf{K}_{vhd} \dot{\mathbf{e}}_{v_h}$$

$$\boldsymbol{\Theta}_{hd} = g^{-1} \mathbf{A}_\psi^{-1} \left(\mathbf{K}_{vhp} \mathbf{e}_{v_h} + \mathbf{K}_{vhi} \int \mathbf{e}_{v_h} + \mathbf{K}_{vhd} \dot{\mathbf{e}}_{v_h} \right) \quad \text{高度达到期望}$$

如果 $\lim_{t \rightarrow \infty} \|\boldsymbol{\Theta}_h(t) - \boldsymbol{\Theta}_{hd}(t)\| = 0$

那么 $\lim_{t \rightarrow \infty} \|\mathbf{e}_{v_h}(t)\| = 0$

角度能达到期望，速度也就能达到期望





2. 位置控制

产生期望欧拉角的位置控制

(2) 开源自驾仪PID设计

2) 高度通道模型

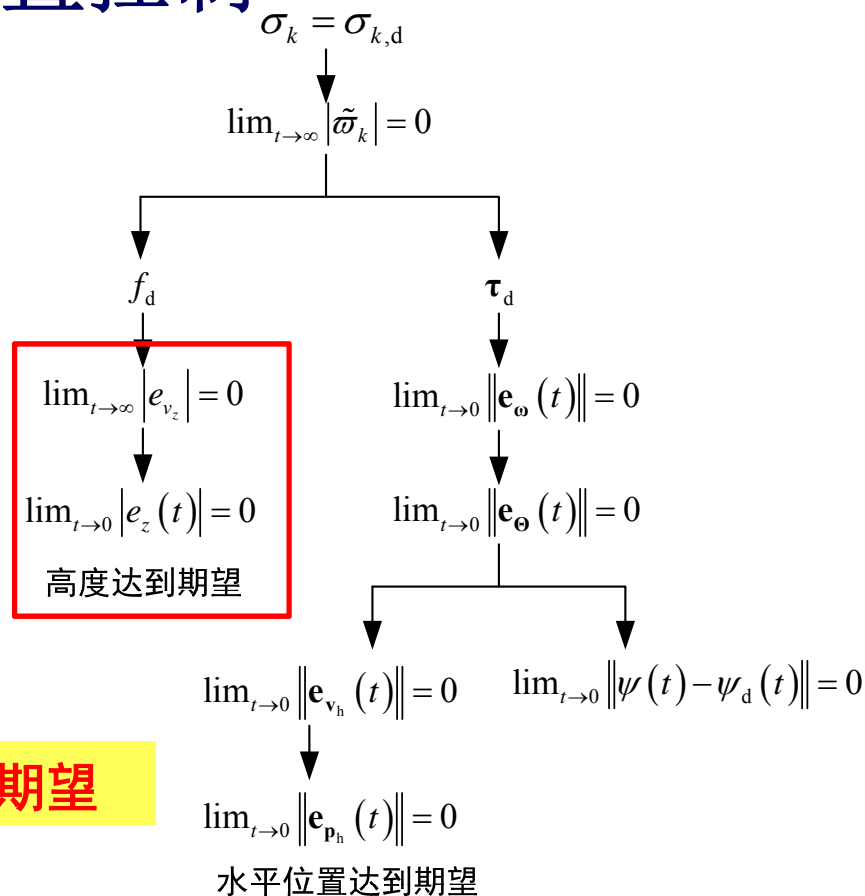
类似于水平通道设计，高度通道

控制器设计如下

$$v_{zd} = -k_{p_z} (p_z - p_{zd})$$

$$f_d = m \left(g + k_{v_zp} e_{v_z} + k_{v_zi} \int e_{v_z} + k_{v_zd} \dot{e}_{v_z} \right)$$

竖直方向速度能达到期望，高度也就能达到期望





2. 位置控制

□ 产生期望欧拉角的位置控制

(3) 加饱和的PID控制

传统PID $\Theta_{hd} = -g^{-1} A_{\psi}^{-1} \left(\ddot{p}_{hd} - K_{p_{hd}} (\dot{p}_h - \dot{p}_{hd}) - K_{p_{hp}} (p_h - p_{hd}) \right)$

自驾仪PID $\Theta_{hd} = g^{-1} A_{\psi}^{-1} \left(K_{v_{hp}} e_{v_h} + K_{v_{hi}} \int e_{v_h} + K_{v_{hd}} \dot{e}_{v_h} \right)$

控制器设计
就没意义了

位置误差很大



角度误差很大

$$\Theta_{hd} \gg 2\pi$$

小角度假设被破坏!

加饱和非常必要



2. 位置控制

□ 产生期望欧拉角的位置控制

(3) 加饱和的PID控制

同样地，在开源自驾仪 PID 设计中，我们增加对 $\mathbf{e}_{\mathbf{v}_h}$ 和控制器右端的限幅

$$\begin{aligned}\mathbf{e}_{\mathbf{v}_h} &= \text{sat}_{\text{gd}}(\mathbf{v}_h - \mathbf{v}_{\text{hd}}, a_1) \\ \Theta_{\text{hd}} &= \text{sat}_{\text{gd}}\left(\mathbf{g}^{-1}\mathbf{A}_{\psi}^{-1}\left(\mathbf{K}_{\mathbf{v}_h\text{p}}\mathbf{e}_{\mathbf{v}_h} + \mathbf{K}_{\mathbf{v}_h\text{i}}\int\mathbf{e}_{\mathbf{v}_h} + \mathbf{K}_{\mathbf{v}_h\text{d}}\dot{\mathbf{e}}_{\mathbf{v}_h}\right), a_2\right)\end{aligned}$$

其中 $a_1, a_2 \in \mathbb{R}_+$ 。保方向饱和函数 $\text{sat}_{\text{gd}}(\mathbf{x}, a)$ 的定义见第 10 讲。

下面讨论保方向饱和函数 $\text{sat}_{\text{gd}}(\mathbf{x}, a)$ 与传统的饱和函数 $\text{sat}(\mathbf{x}, a)$ 的区别。虽然饱和函数限制 $\text{sat}(\mathbf{x}, a)$ 的每个分量的绝对值不大于 a ，但它与 \mathbf{x} 的方向可能不同。而保方向饱和函数 $\text{sat}_{\text{gd}}(\mathbf{x}, a)$ ，不仅可以限制最终向量每个分量的绝对值不大于 a ，还可以保证 $\text{sat}_{\text{gd}}(\mathbf{x}, a)$ 的方向与 \mathbf{x} 相同。



2. 位置控制

产生期望欧拉角的位置控制

(3) 加饱和的PID控制

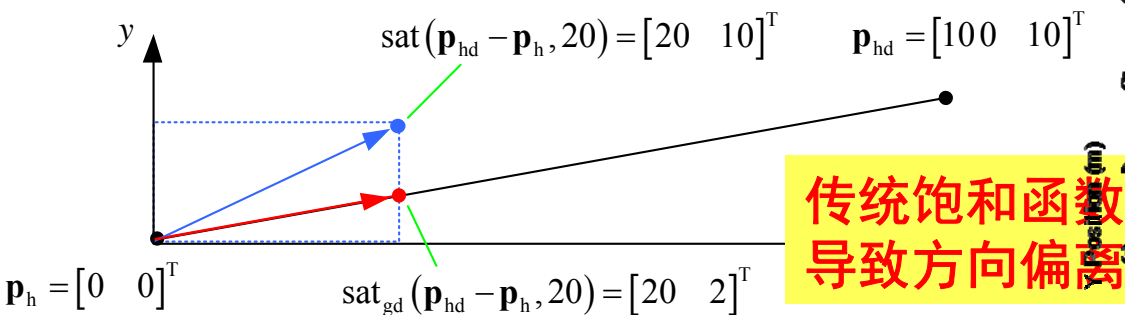


图. 两种饱和函数的作用效果对比图

传统饱和函数

$$\text{sat}(\mathbf{x}, a) \triangleq \begin{bmatrix} \text{sat}(x_1, a) \\ \vdots \\ \text{sat}(x_n, a) \end{bmatrix}, \text{sat}(x_k, a) \triangleq \begin{cases} x_k & |x_k| \leq a \\ a \cdot \text{sign}(x_k) & |x_k| > a \end{cases}$$

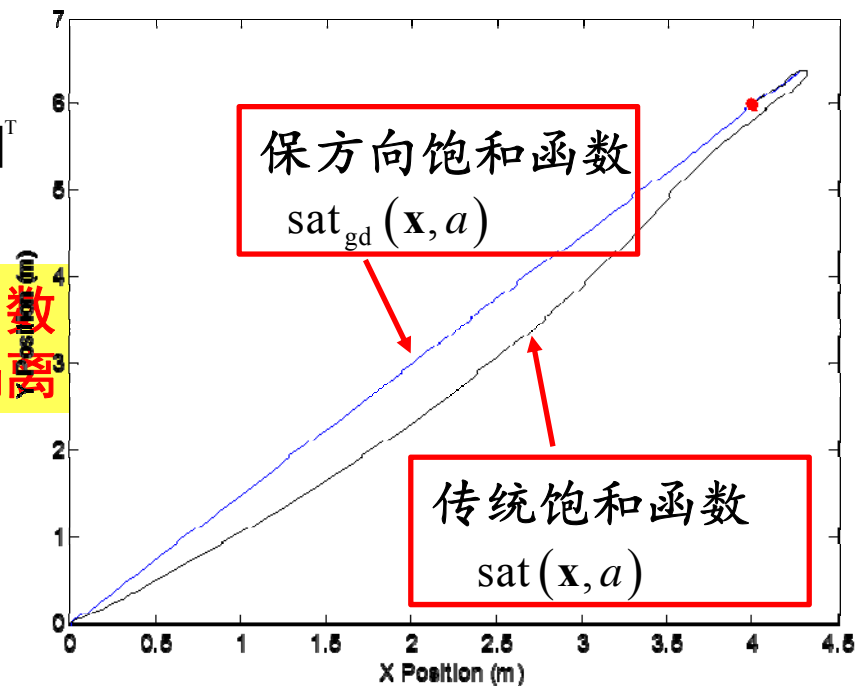


图. 饱和情况下多旋翼定点跟踪水平面运动轨迹

保方向的饱和函数可以保证多旋翼直线飞行，而传统饱和函数则不行



2. 位置控制

□ 产生期望欧拉角的位置控制

(3) 加饱和的PID控制

为了避免油门过大，同样需要加入饱和。在传统 PID 控制器设计中，我们需要对控制器的右端进行如下限幅

$$f_d = \text{sat}_{\text{gd}} \left(m \left(g + k_{p_z d} \dot{p}_z + k_{p_z p} (p_z - p_{z_d}) \right), a_3 \right)$$

其中 $a_3 \in \mathbb{R}_+$ 。同样地，在开源自驾仪 PID 设计中，我们增加对 e_{v_z} 和控制器的右端的限幅

$$e_{v_z} = \text{sat}_{\text{gd}} (v_z - v_{z_d}, a_4)$$

$$f_d = \text{sat}_{\text{gd}} \left(m \left(g + k_{v_z p} e_{v_z} + k_{v_z i} \int e_{v_z} + k_{v_z d} \dot{e}_{v_z} \right), a_5 \right)$$

其中 $a_4, a_5 \in \mathbb{R}_+$ 。对于一维变量，保方向饱和函数 sat_{gd} 和与传统饱和函数 sat 的作用相同。



2. 位置控制

□ 产生期望旋转矩阵的位置控制

因为姿态运动学方程也可以利用旋转矩阵表述，所以当位置与姿态采取内外环控制时，我们自然期望位置控制器可以直接给出期望的旋转矩阵 \mathbf{R}_d 。因为旋转矩阵 \mathbf{R}_d 需要直接通过向量进行设计，所以外环的位置控制可直接针对非线性耦合模型进行控制器设计。

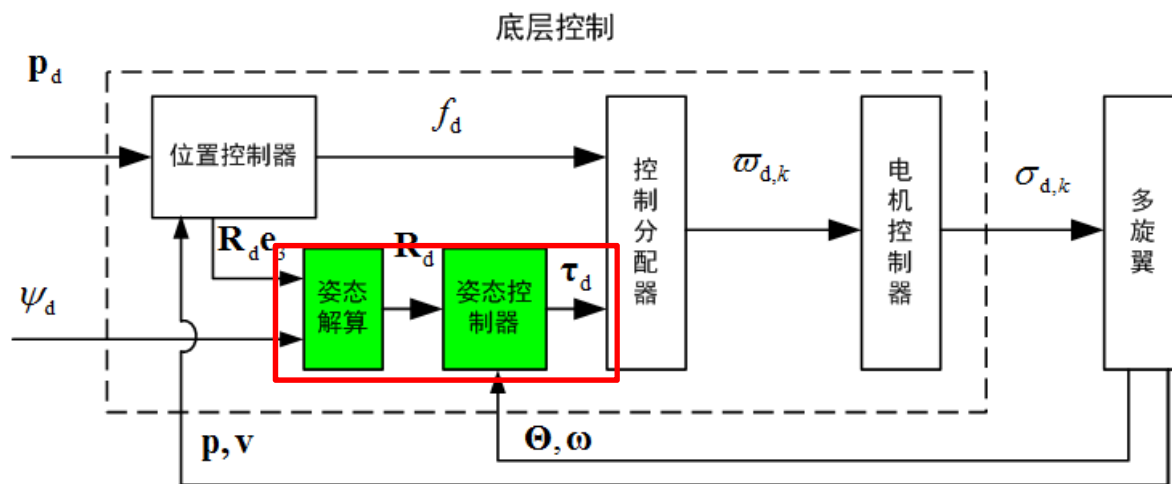


图. 位置控制器给出 \mathbf{R}_d 情况下的多旋翼总体控制框图



2. 位置控制

□ 产生期望旋转矩阵的位置控制 [2]

针对非线性耦合模型进行如下控制器设计

$$g\mathbf{e}_3 - \frac{f_d}{m}\mathbf{r}_{d,3} = \mathbf{a}_d$$

其中 $\mathbf{b}_{d,3} = \mathbf{R}_d\mathbf{e}_3$ 以及

$$\mathbf{a}_d = \begin{bmatrix} \text{sat}_{\text{gd}} \left(-\mathbf{K}_{vhp}\mathbf{e}_{v_h} - \mathbf{K}_{vhi}\int \mathbf{e}_{v_h} - \mathbf{K}_{vhd}\dot{\mathbf{e}}_{v_h}, a_1 \right) \\ \text{sat}_{\text{gd}} \left(-k_{vzp}e_{v_z} - k_{vzi}\int e_{v_z} - k_{vzd}\dot{e}_{v_z}, a_2 \right) \end{bmatrix}$$

多旋翼位置模型

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

$$\dot{\mathbf{v}} = g\mathbf{e}_3 - \frac{f}{m}\mathbf{R}\mathbf{e}_3$$

期望的加速度，需要加饱和

先求解期望的旋转矩阵 \mathbf{R}_d 。由上式可以得到

$$\mathbf{r}_{d,3} = \frac{m(g\mathbf{e}_3 - \mathbf{a}_d)}{f_d}$$

[2] Lee, T, Leoky M, and McClamroch N H. Geometric tracking control of a quadrotor UAV on SE(3). In: Proceedings of the 49th IEEE Conference on Decision and Control. Atlanta, Georgia, USA: IEEE, 2010. 5420–5425.



2. 位置控制

□ 产生期望旋转矩阵的位置控制

但是为了满足 $\mathbf{r}_{d,3}^T \mathbf{r}_{d,3} = 1$ ，我们只需要得到方向

$$\mathbf{r}_{d,3} = \frac{g\mathbf{e}_3 - \mathbf{a}_d}{\|g\mathbf{e}_3 - \mathbf{a}_d\|}$$

然而 $\mathbf{R}_d = [\mathbf{r}_{d,1} \quad \mathbf{r}_{d,2} \quad \mathbf{r}_{d,3}]$ ，因此还要确定 $\mathbf{r}_{d,1}, \mathbf{r}_{d,2}$ 中的一个，我们就可以恢复

出 \mathbf{R}_d 了。根据旋转矩阵的定义， $\mathbf{r}_1 \in \mathbb{R}^3$ 为

$$\mathbf{r}_1 = \begin{bmatrix} \cos \theta \cos \psi \\ \cos \theta \sin \psi \\ -\sin \theta \end{bmatrix}$$



2. 位置控制

□ 产生期望旋转矩阵的位置控制

因此，我们定义向量

$$\bar{\mathbf{r}}_{d,1} = \begin{bmatrix} \cos \psi_d \\ \sin \psi_d \\ 0 \end{bmatrix}$$

在姿态角变化不大的情况下，可认为 $\bar{\mathbf{r}}_{d,1} \approx \mathbf{r}_{d,1}$ 。这样通过 $\bar{\mathbf{r}}_{d,1}$ 和 $\mathbf{r}_{d,3}$ 可以将 $\mathbf{r}_{d,2}$ 定义：

$$\mathbf{r}_{d,2} = \frac{\mathbf{r}_{d,3} \times \bar{\mathbf{r}}_{d,1}}{\|\mathbf{r}_{d,3} \times \bar{\mathbf{r}}_{d,1}\|}$$

根据叉乘的定义，可以知道 $\mathbf{r}_{d,2} \perp \mathbf{r}_{d,3}$ 。进一步，定义 $\mathbf{r}_{d,1} = \mathbf{r}_{d,2} \times \mathbf{r}_{d,3}$ 。这样

$$\mathbf{r}_{d,1} \perp \mathbf{r}_{d,3}, \mathbf{r}_{d,1} \perp \mathbf{r}_{d,2}。$$



2. 位置控制

□ 产生期望旋转矩阵的位置控制

至此，可得期望的姿态矩阵 \mathbf{R}_d 为

$$\mathbf{R}_d = \begin{bmatrix} \mathbf{r}_{d,2} \times \mathbf{r}_{d,3} & \mathbf{r}_{d,2} & \mathbf{r}_{d,3} \end{bmatrix}$$

其满足 $\mathbf{R}_d^T \mathbf{R}_d = \mathbf{I}_3$ 。下面进一步考虑期望升力 f_d 的计算，因为变量 f_d 和 $\mathbf{b}_{d,3}$ 耦合在一起，因此我们

进行如下处理。首先在 $g\mathbf{e}_3 - \frac{f_d}{m}\mathbf{b}_{d,3} = \mathbf{a}_d$ 两边同乘 $\mathbf{r}_{d,3}^T$ ，利用 $\mathbf{r}_{d,3}^T \mathbf{r}_{d,3} = 1$ 可以得到

$$f_d = m\mathbf{r}_{d,3}^T (g\mathbf{e}_3 - \mathbf{a}_d)$$

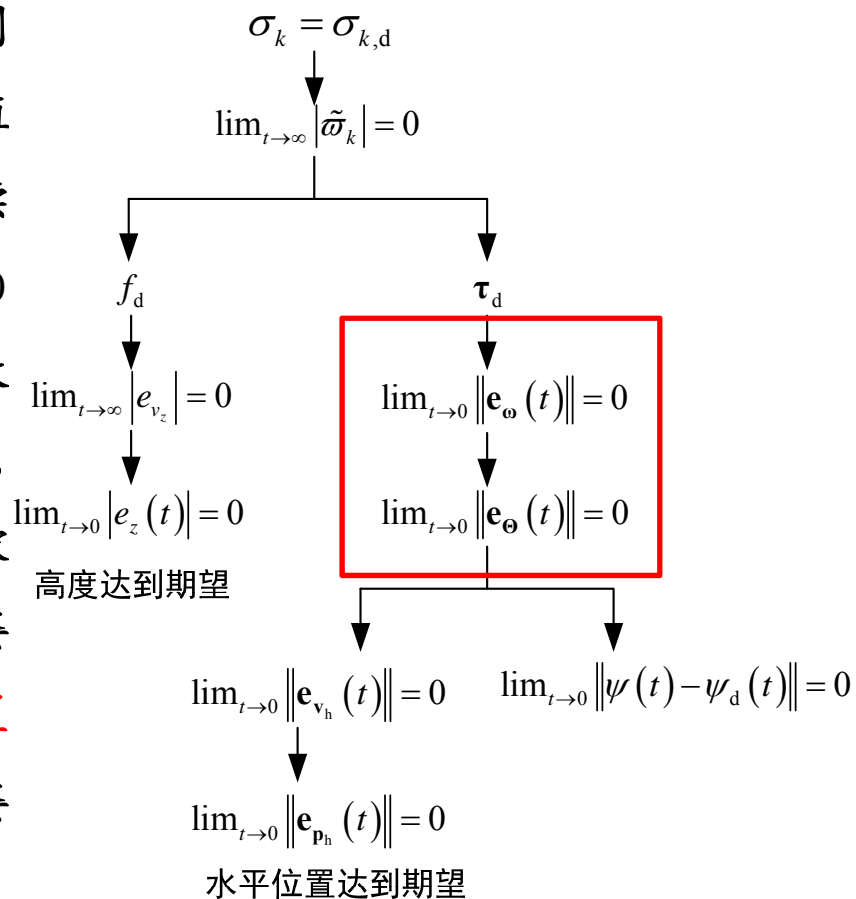
如果限定 $f_d \in [f_{\min}, f_{\max}]$ ，那么姿态控制器可以写为

$$f_d = \text{sat}_{\text{gd}} \left(m\mathbf{r}_{d,3}^T (g\mathbf{e}_3 - \mathbf{a}_d) - \frac{f_{\min} + f_{\max}}{2}, \frac{f_{\max} - f_{\min}}{2} \right) + \frac{f_{\min} + f_{\max}}{2}$$



3. 姿态控制

多旋翼采用**分层控制**，外层控制器为内层控制器提供指令，即把外环控制器得到的值当作理想值。这里就是将 Θ_{hd} 或者 R_d 作为姿态控制的期望。后续的姿态控制的目标就是完成 $\lim_{t \rightarrow \infty} \|\Theta_h(t) - \Theta_{hd}(t)\| = 0$ 或者 $\lim_{t \rightarrow \infty} \|R^T R_d - I_3\| = 0$ 。不仅如此，一般要求收敛速度比水平通道动态快 4~5 倍。从水平通道看来，可以认为姿态控制目标 $\Theta_h = \Theta_{hd}$ 或 $R(t) = R_d(t)$ 被“瞬间”实现。因此，目标像**击鼓传花**一样，传给姿态控制了。**只要姿态控制被很好地实现，水平位置跟踪的问题也就被解决了**。下面，我们再来介绍姿态控制。





3. 姿态控制

□ 基本概念

多旋翼姿态控制是飞行器位置控制的基础。常见的刚体姿态描述方法为欧拉角描述方法和旋转矩阵描述方法。

表.姿态参数性能对比

姿态参数	优点	缺点
欧拉角	无冗余参数，物理意义明确	俯仰角为90度时存在奇异，存在大量超越函数运算，死锁；偏航角从 2π 到0的过渡问题。
旋转矩阵	无奇异，无超越函数运算，可用于连续旋转表示，全局且唯一，便于插值	六个冗余参数



3. 姿态控制

□ 基于欧拉角的控制器设计

姿态控制的目标是已知参考姿态角 $\Theta_d = [\Theta_{hd}^T \ \psi_d]^T$ ，设计控制器 τ_d 使得 $\lim_{t \rightarrow \infty} \|e_\Theta(t)\| = 0$ ，其中 $e_\Theta \triangleq \Theta - \Theta_d$ 。这里 Θ_{hd} 是由位置控制器给定的，而 ψ_d 是任务规划给定的。为了达到这个目的，我们先针对

$$\dot{\Theta} = \omega$$

设计角速度的期望 ω_d 为

$$\omega_d = -K_\Theta e_\Theta$$

其中 $K_\Theta \in \mathbb{R}^{3 \times 3}$ 是正定的常值对角矩阵，所有元素都大于 0。以上两式构成了角度控制环。接下来的任务是针对

$$J\dot{\omega} = \tau$$

设计转矩的期望 τ_d ：

$$\tau_d = -K_{\omega p} e_\omega - K_{\omega i} \int e_\omega - K_{\omega d} \dot{e}_\omega$$

如何加入饱和？

其中 $e_\omega \triangleq \omega - \omega_d$ ， $K_{\omega p}, K_{\omega i}, K_{\omega d} \in \mathbb{R}^{3 \times 3}$ 。以上两式构成了角速度控制环。



3. 姿态控制

□ 基于旋转矩阵的控制器设计[2]

控制器设计的思路为：首先定义期望机体坐标系以及相应的期望坐标旋转矩阵，利用期望坐标旋转矩阵与实际坐标旋转矩阵定义姿态跟踪误差，并通过跟踪误差设计姿态控制器。

首先定义姿态误差矩阵为

$$\tilde{\mathbf{R}} = \mathbf{R}^T \mathbf{R}_d$$

由旋转矩阵的定义可知：当且仅当 $\mathbf{R} = \mathbf{R}_d$ 时， $\tilde{\mathbf{R}} = \mathbf{I}_3$ 。基于以上分析可以将基于旋转矩阵的

多旋翼姿态跟踪控制的控制目标总结为 $\lim_{t \rightarrow \infty} \|\tilde{\mathbf{R}}(t) - \mathbf{I}_3\| = 0$ 。

定义姿态跟踪误差为

$$\mathbf{e}_R = \frac{1}{2} \text{vex}(\mathbf{R}_d^T \mathbf{R} - \mathbf{R}^T \mathbf{R}_d)$$

$$\text{其中 } \mathbf{x} = [x_1 \quad x_2 \quad x_3]^T \in \mathbb{R}^3, [\mathbf{x}]_{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \text{vex}([\mathbf{x}]_{\times}) = \mathbf{x}$$



3. 姿态控制

□ 基于旋转矩阵的控制器设计

姿态角速度跟踪误差定义为

$$\mathbf{e}_\omega = \omega - \mathbf{R}^T \mathbf{R}_d \omega_d$$

小角度假设下 $\omega_d = \dot{\Theta}_d$ 为期望的角速度, 一般情况下可忽略, 则 $\mathbf{e}_\omega = \omega$ 。此时, 通过计算可得 $\mathbf{e}_R = \Theta - \Theta_d$ 。

可以设计如下的 PD 控制器

$$\tau_d = -\mathbf{K}_R \mathbf{e}_R - \mathbf{K}_\omega \mathbf{e}_\omega$$

其中 $\mathbf{K}_R, \mathbf{K}_\omega \in \mathbb{R}^{3 \times 3}$ 为正定增益矩阵。上面设计的 PD 控制器只可以在悬停位置的小范围内保证系统稳定。为了获得更大范围的稳定性, 通过引入误差校正项, 设计非线性控制器为:

$$\tau_d = -\mathbf{K}_R \mathbf{e}_R - \mathbf{K}_\omega \mathbf{e}_\omega - \mathbf{J} \left([\omega]_\times \mathbf{R}^T \mathbf{R}_d \omega_d - \mathbf{R}^T \mathbf{R}_d \dot{\omega}_d \right)$$

此控制器可以保证在几乎任意旋转情况下系统指数稳定。实际中控制器的后两项的值很小, 直接忽略也可以获得满意的性能。但对于大机动飞行中后两项的值很大, 对系统性能的影响很大, 则必须予以考虑。



4. 控制分配

□ 基本概念

简单来说，控制分配问题可描述为：给定 $\mathbf{u}_v(t)$,

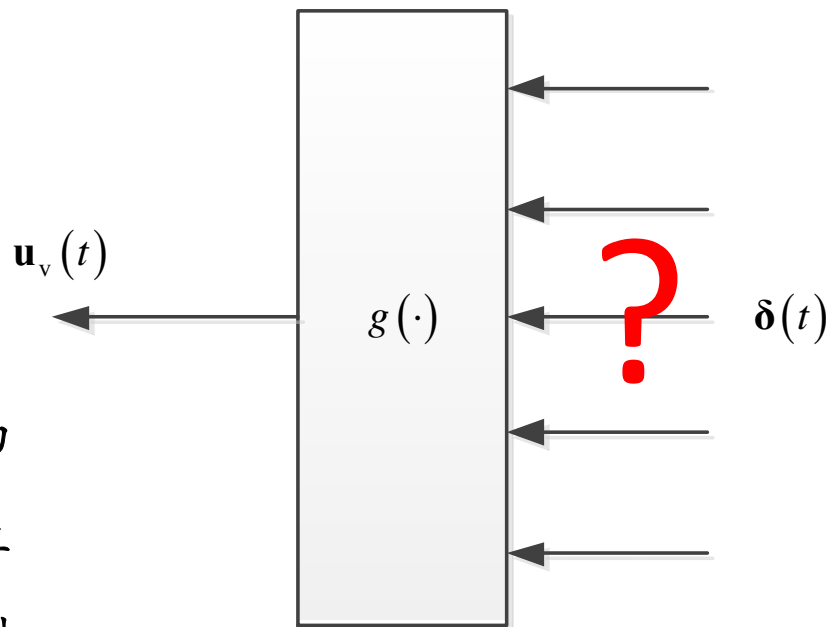
寻找 $\delta(t)$ 使得

$$\mathbf{u}_v(t) = g(\delta(t))$$

其中 $g: \mathbb{R}^m \rightarrow \mathbb{R}^n$ 为被控系统中执行机构控制输入到伪控制输入的映射。通常假设操纵机构偏转量与其产生的控制力矩之间为线性函数关系，则可得到线性控制分配问题

$$\mathbf{u}_v(t) = \mathbf{B}\delta(t)$$

其中 $\mathbf{B} \in \mathbb{R}^{m \times n}$ 为已知控制效率矩阵。



- 上层控制和底层控制分离
- 有效分配，防止饱和
- 通过分配，提高对故障及损伤的鲁棒性
- ...



4. 控制分配

□ 多旋翼控制分配的自驾仪实现

+字形四旋翼的控制分配和多旋翼的控制分配模型见第6讲：

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & -dc_T & 0 & dc_T \\ dc_T & 0 & -dc_T & 0 \\ c_M & -c_M & c_M & -c_M \end{bmatrix}}_{\mathbf{M}_4} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix} \quad \begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_T & c_T & \cdots & c_T \\ -dc_T \sin \varphi_1 & -dc_T \sin \varphi_2 & \cdots & -dc_T \sin \varphi_{n_r} \\ dc_T \cos \varphi_1 & dc_T \cos \varphi_2 & \cdots & dc_T \cos \varphi_{n_r} \\ c_M \delta_1 & c_M \delta_2 & \cdots & c_M \delta_{n_r} \end{bmatrix}}_{\mathbf{M}_{n_r}} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \vdots \\ \varpi_{n_r}^2 \end{bmatrix}$$

对于四旋翼来说， $\mathbf{M}_4 \in \mathbb{R}^{4 \times 4}$ 可逆，可直接求逆得到控制分配矩阵 $\mathbf{P}_4 = \mathbf{M}_4^{-1}$ ，其中 $\mathbf{P}_4 \in \mathbb{R}^{4 \times 4}$ 。这

种分配是唯一的。然而，对于旋翼数大于4的多旋翼，分配可以有无穷多种。分配过程中，可能让某个旋翼的转速达到饱和。因此，好的控制分配算法就十分重要。在开源的自驾仪中，一般的算法是通过**求伪逆**得到控制分配矩阵，即

$$\mathbf{P}_{n_r} = \mathbf{M}_{n_r}^\dagger$$

其中 $\mathbf{P}_{n_r} \in \mathbb{R}^{n_r \times 4}$ ， $\mathbf{M}_{n_r} \in \mathbb{R}^{4 \times n_r}$ 。



4. 控制分配

□ 多旋翼控制分配的自驾仪实现

在实际中，通过控制器得到期望拉力 f_d 和期望力矩 τ_d 后，进一步再通过

$$\begin{pmatrix} \omega_{d,1}^2 \\ \omega_{d,2}^2 \\ \vdots \\ \omega_{d,n_r}^2 \end{pmatrix} = \mathbf{P}_{n_r} \begin{pmatrix} f_d \\ \tau_d \end{pmatrix}$$

与参数相关

可得到每个螺旋桨的期望转速 $\omega_{d,i}, i=1,2,\dots,n_r$ 。

在工程实际中， \mathbf{M}_{n_r} 中的参数未知，即 c_T, c_M, d （具体定义参见第六讲）未知时。

那这又是如何进行控制分配的呢？





4. 控制分配

□ 多旋翼控制分配的自驾仪实现

要说明这个问题，首先定义映射矩阵为

$$\mathbf{M}_{n_r}(c_T, c_M, d) = \begin{bmatrix} c_T & c_T & \cdots & c_T \\ -dc_T \sin \varphi_1 & -dc_T \sin \varphi_2 & \cdots & -dc_T \sin \varphi_{n_r} \\ dc_T \cos \varphi_1 & dc_T \cos \varphi_2 & \cdots & dc_T \cos \varphi_{n_r} \\ c_M \sigma_1 & c_M \sigma_2 & \cdots & c_M \sigma_{n_r} \end{bmatrix}$$

它满足

$$\mathbf{M}_{n_r}(c_T, c_M, d) = \mathbf{P}_a \mathbf{M}_{n_r}(1, 1, 1)$$

其中 $\mathbf{P}_a = \text{diag}(c_T \quad dc_T \quad dc_T \quad c_M)$ 。因此，有如下关系

$$\mathbf{M}_{n_r}^\dagger(c_T, c_M, d) = \underbrace{\mathbf{M}_{n_r}^\dagger(1, 1, 1)}_{\text{已知参数}} \underbrace{\mathbf{P}_a^{-1}}_{\text{未知参数}}$$



4. 控制分配

□ 多旋翼控制分配的自驾仪实现

以六旋翼为例, $\mathbf{M}_6(c_T, c_M, d)$ 可以表示为

$$\mathbf{M}_6(c_T, c_M, d) = \begin{bmatrix} c_T & c_T & c_T & c_T & c_T & c_T \\ 0 & -\frac{\sqrt{3}dc_T}{2} & -\frac{\sqrt{3}dc_T}{2} & 0 & \frac{\sqrt{3}dc_T}{2} & \frac{\sqrt{3}dc_T}{2} \\ dc_T & \frac{dc_T}{2} & -\frac{dc_T}{2} & -dc_T & -\frac{dc_T}{2} & \frac{dc_T}{2} \\ c_M & -c_M & c_M & -c_M & c_M & -c_M \end{bmatrix} \Rightarrow \mathbf{M}_6^\dagger(1,1,1) = \frac{1}{6} \begin{bmatrix} 1 & 0 & 2 & 1 \\ 1 & -\sqrt{3} & 1 & -1 \\ 1 & -\sqrt{3} & -1 & 1 \\ 1 & 0 & -2 & -1 \\ 1 & \sqrt{3} & -1 & 1 \\ 1 & \sqrt{3} & 1 & -1 \end{bmatrix}$$

于是

$$\begin{pmatrix} \omega_{d,1}^2 \\ \omega_{d,2}^2 \\ \vdots \\ \omega_{d,6}^2 \end{pmatrix} = \mathbf{M}_6^\dagger(1,1,1) \mathbf{P}_a^{-1} \begin{pmatrix} f_d \\ \boldsymbol{\tau}_d \end{pmatrix} = \mathbf{M}_6^\dagger(1,1,1) \begin{pmatrix} f_d/c_T \\ \tau_{dx}/dc_T \\ \tau_{dy}/dc_T \\ \tau_{dz}/c_M \end{pmatrix}$$

自驾仪中, 控制器都是PID控制器。可以通过调节PID参数来补偿未知量。

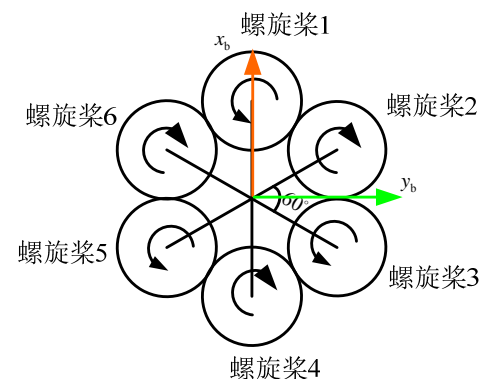


图.常规布局的六旋翼



5. 电机控制

□ 闭环控制

假设多旋翼中每个转子的转速为 $\varpi_k, k=1, 2, \dots, n_r$ 。大多数多旋翼都配置无刷直流电机，其利用反电势力来感知转子的整流，利用高频脉宽调制模块（PWM）来控制电机的电压。下面给出利用电机油门来实现电机控制的控制器设计。**控制目标为：设计每个电机的输入油门 $\sigma_{d,k}$ 使 $\lim_{t \rightarrow \infty} |\varpi_k - \varpi_{d,k}| = 0$ 。**以油门作为输入的电机模型见第六章，对应到多旋翼上每个电机的动态模型为

$$\varpi_k = \frac{1}{T_m s + 1} C_R \sigma_k + \varpi_b$$

其中油门 σ_k 为输入，转子转速 ϖ_k 为输出。电机控制的目标是使控制误差 $\tilde{\varpi}_k = \varpi_k - \varpi_{d,k}$ 最小化，设计

$$\sigma_{d,k} = -k_{\varpi} \tilde{\varpi}_k$$

其中 $k_{\varpi} \in \mathbb{R}_+$ 。 ϖ_k 为实际转速，可以通过嵌入在速度控制器中的电子整流器测得。



5. 电机控制

□ 开环控制

在开源自驾仪中，电机为开环控制，在得到 f_d, τ_d 之后，期望的每个转子的转速为 $\omega_{d,k}, k=1,2,\dots,n_r$ 可通过控制分配直接得到，其与期望油门值成正比。开环控制器可以设计为

$$\sigma_{d,k} = a\omega_{d,k} + b$$

参数 a, b 可以通过位置和姿态控制器中的 PID 参数来补偿。

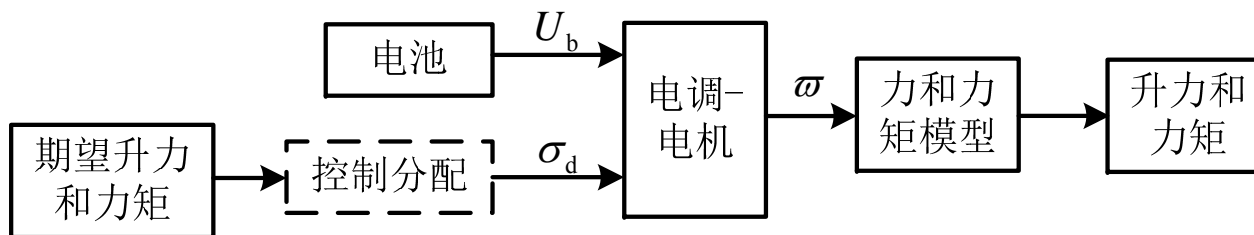


图. 多旋翼动力系统的开环控制及工作框图



6. 综合仿真

为了验证算法的有效性，我们利用MATLAB进行了相关仿真。仿真中采用的四旋翼构型是十字形。它对于X字形构型的四旋翼，除了控制矩阵 M_4 不同外，其余参数均相同，仿真中只需对应修改控制分配模块即可。仿真的搭建本着尽可能贴近实际的原则，考虑了系统输入的限幅，执行器死区和电机动态等因素。

表. 四旋翼主要模型参数

g	$9.8100 \text{ (m/s}^2\text{)}$	J_{yy}	$0.0095 \text{ (kg} \cdot \text{m}^2\text{)}$
d	0.2223 (m)	J_{zz}	$0.0186 \text{ (kg} \cdot \text{m}^2\text{)}$
m	1.0230 (kg)	J_{rp}	$3.7882\text{e-}06 \text{ (kg} \cdot \text{m}^2\text{)}$
c_T	$1.4865\text{e-}07 \text{ (N/rpm}^2\text{)}$	C_b	80.5840 (rpm)
c_M	$2.9250\text{e-}09 \text{ (N} \cdot \text{m/rpm}^2\text{)}$	ϖ_b	976.2000 (rpm)
J_{xx}	$0.0095 \text{ (kg} \cdot \text{m}^2\text{)}$	T_M	0.0760 (s)



6. 综合仿真

控制目标为：以初始位置作为中心点，从中心点出发在水平面飞出一个正方形的轨迹，最后回到中心点并保持定点悬停。系统初值选择为：

$$\omega(0)=[4000 \ 4000 \ 4000 \ 4000]^T (\text{rpm}), \mathbf{p}(0)=[0 \ 0 \ -3]^T (\text{m}), \mathbf{v}(0)=[0 \ 0 \ 0]^T (\text{m/s}),$$

$$\Theta(0)=[0 \ 0 \ 0]^T (\text{deg}), \boldsymbol{\omega}(0)=[0 \ 0 \ 0]^T (\text{deg/s}). \text{期望的航迹指令由时间序列给出。}$$

表.期望指令时间序列

时间 (s)	$x_d (\text{m})$	$y_d (\text{m})$	$z_d (\text{m})$	$\psi_d (\text{deg})$
0	0	0	-3	0
5	10	0	-3	0
10	0	10	-3	0
15	-10	0	-3	0
20	0	-10	-3	0
25	10	0	-3	0
30	0	0	-3	0



6. 综合仿真

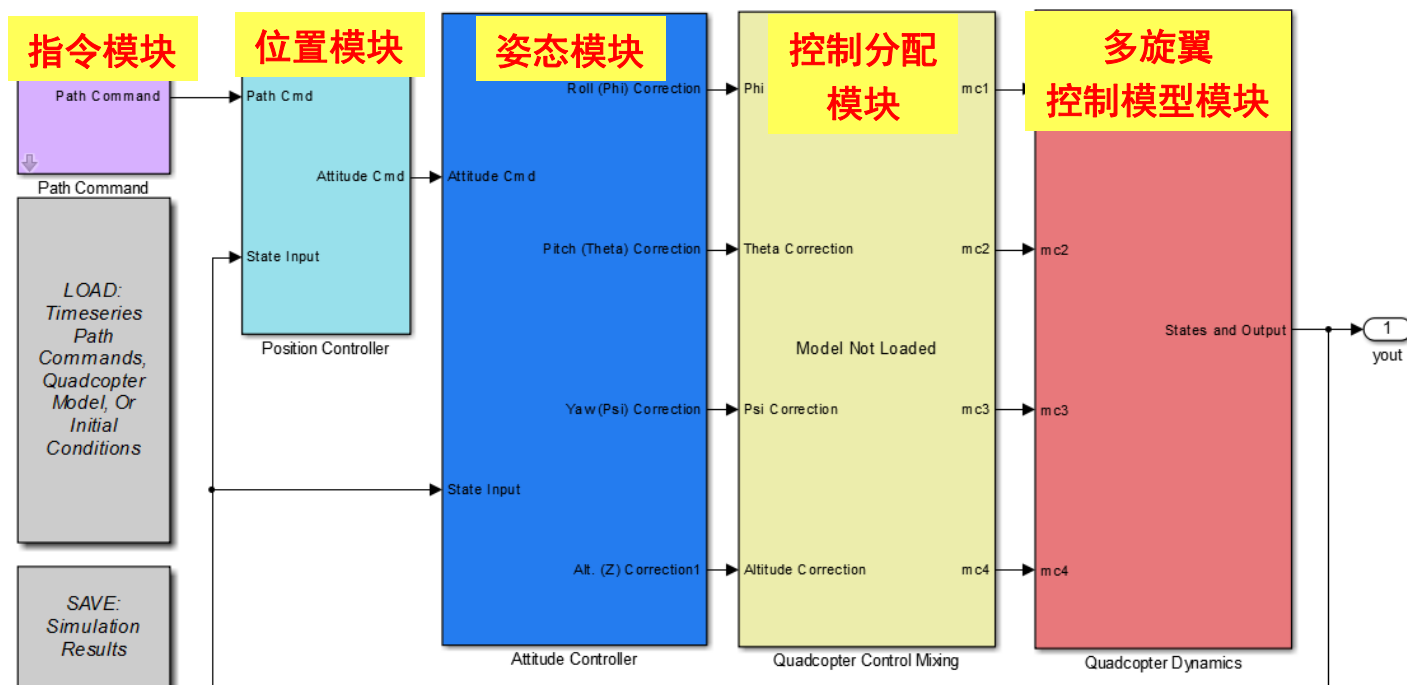


图.综合控制仿真搭建框图

下面将给出基于此仿真搭建的两种控制方案的仿真结果：

- (1) 基于欧拉角的姿态控制结合基于欧拉角的位置控制
- (2) 基于旋转矩阵的姿态控制结合基于旋转矩阵的位置控制



6. 综合仿真

(1) 基于欧拉角的姿态控制结合基于欧拉角的位置控制

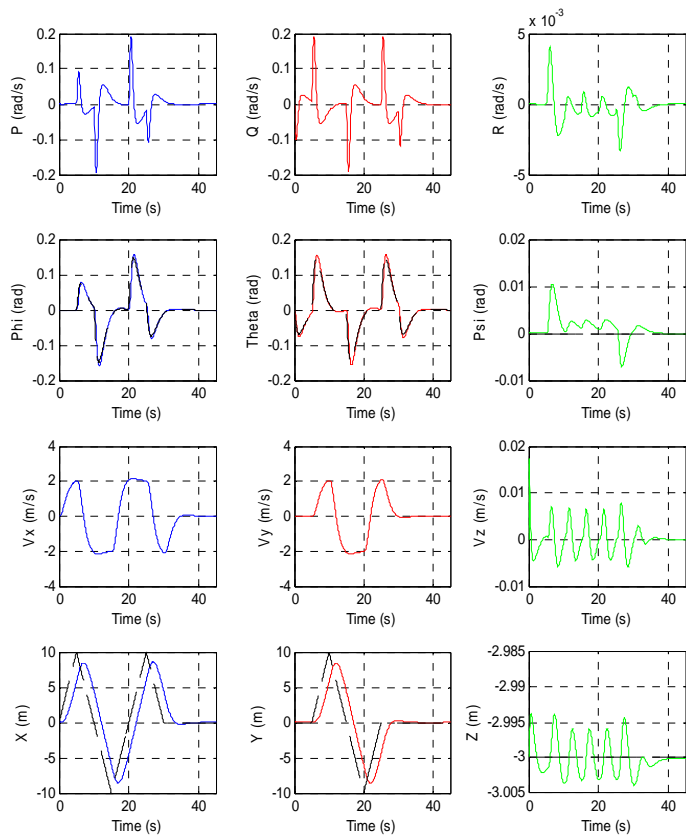


图. 状态响应图

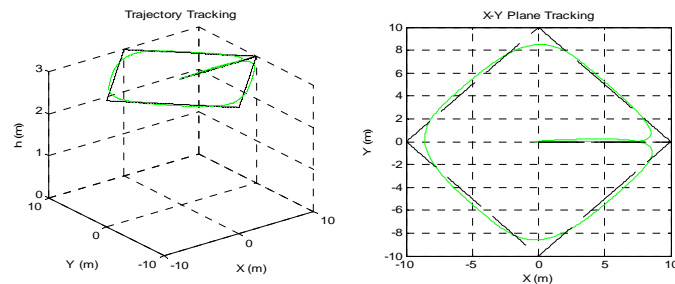


图. 航迹跟踪效果图

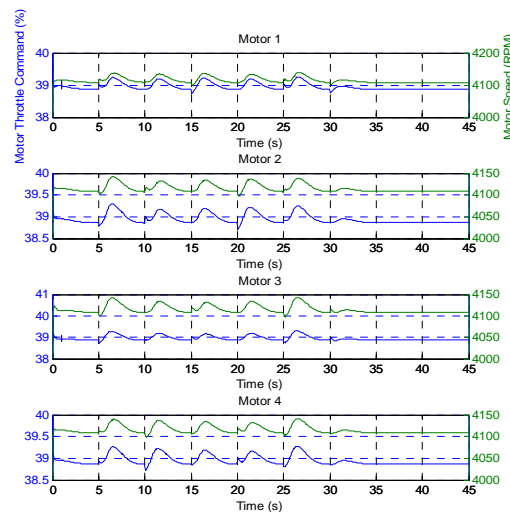


图. 电机油门指令和电机转速响应图



6. 综合仿真

(2) 基于旋转矩阵的姿态控制结合基于旋转矩阵的位置控制

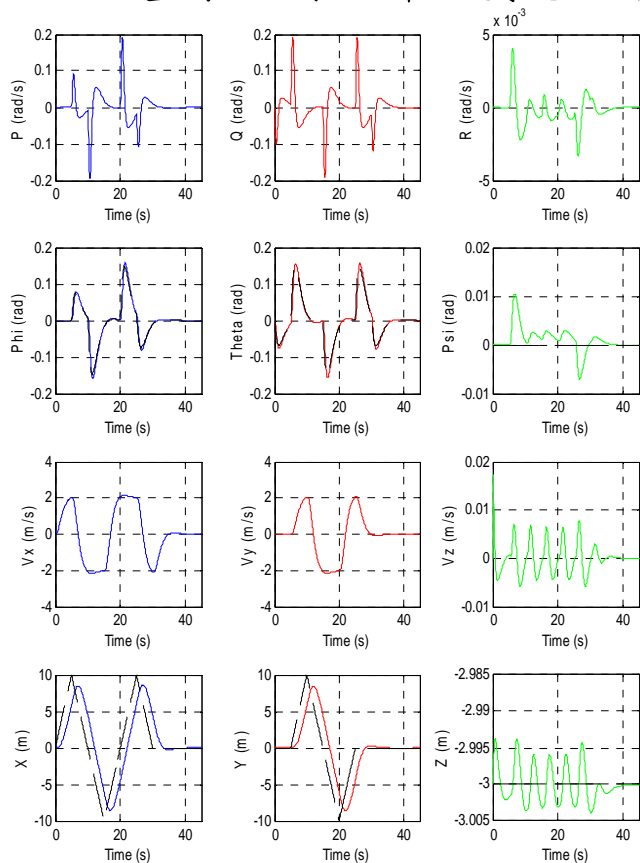


图. 状态响应图

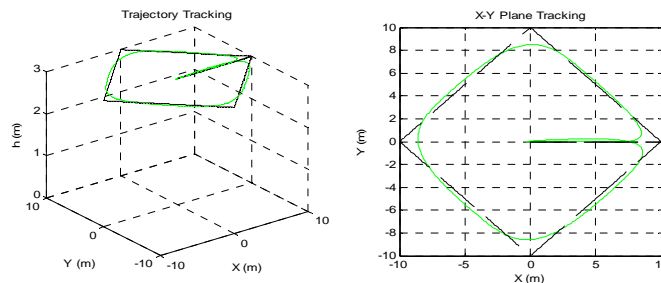


图. 航迹跟踪效果图

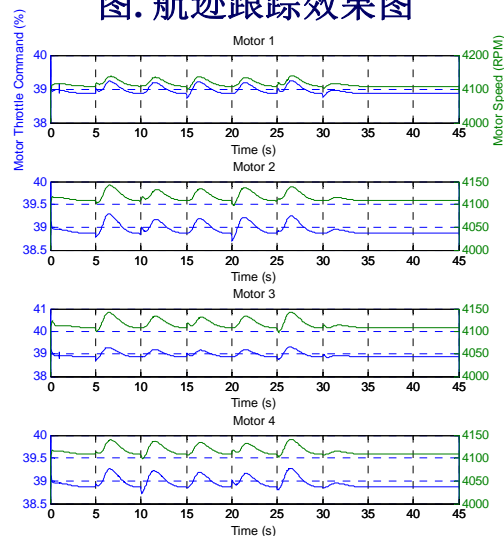


图. 电机油门指令和电机转速响应图



6. 综合仿真

(3) 大机动控制

基于旋转矩阵的控制器相比于基于欧拉角控制器的一大优势就是其可以避免奇异问题，实现全姿态飞行，因此可以实现类似飞行表演中的大机动动作，比如空中筋斗，垂直面画圆等。此处，**基于旋转矩阵实现四旋翼的倒飞翻转**。由下图可见，**滚转角经历了一个由初始正180度（倒飞）到最终的0度（正飞）的过程**。位置经过小幅波动后最终稳定在初始值。

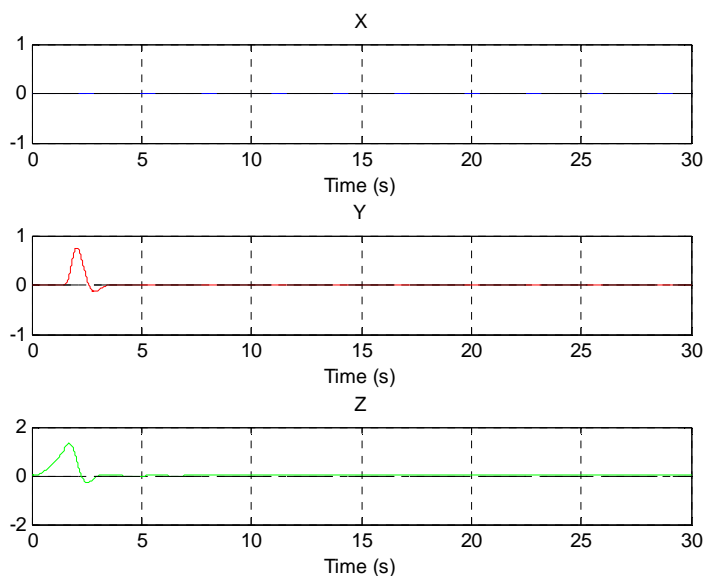


图. 位置响应

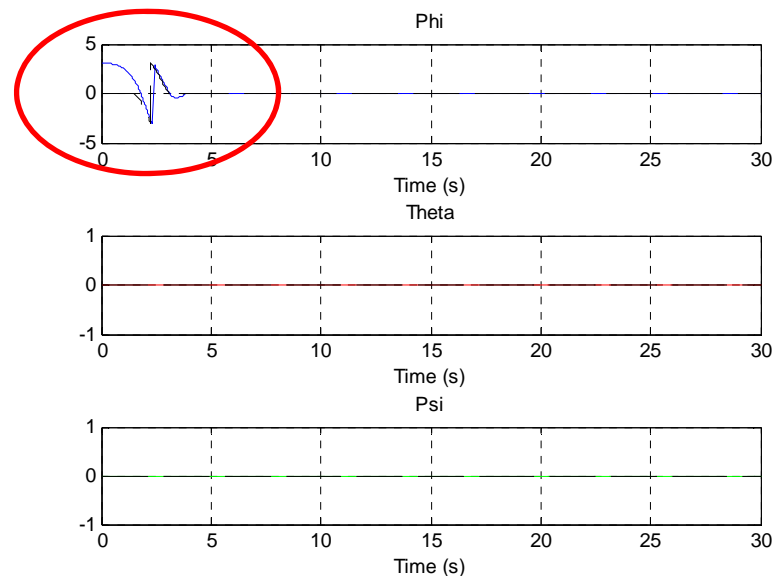


图. 角度响应

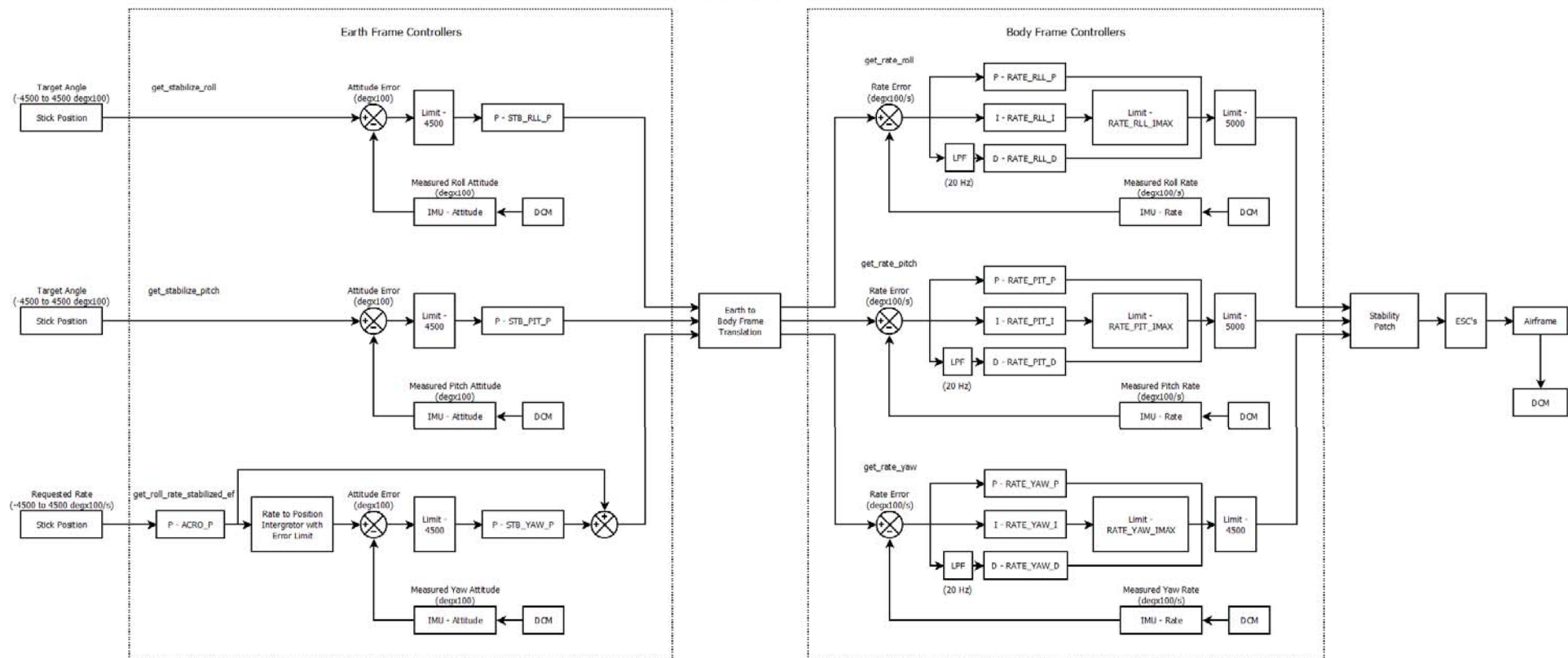


7. 飞控原理与调参

□ 滚转俯仰偏航PID控制

ArduCopter V2.9 STABILIZE Roll, Pitch & Yaw PID's

100 Hz Update Rate

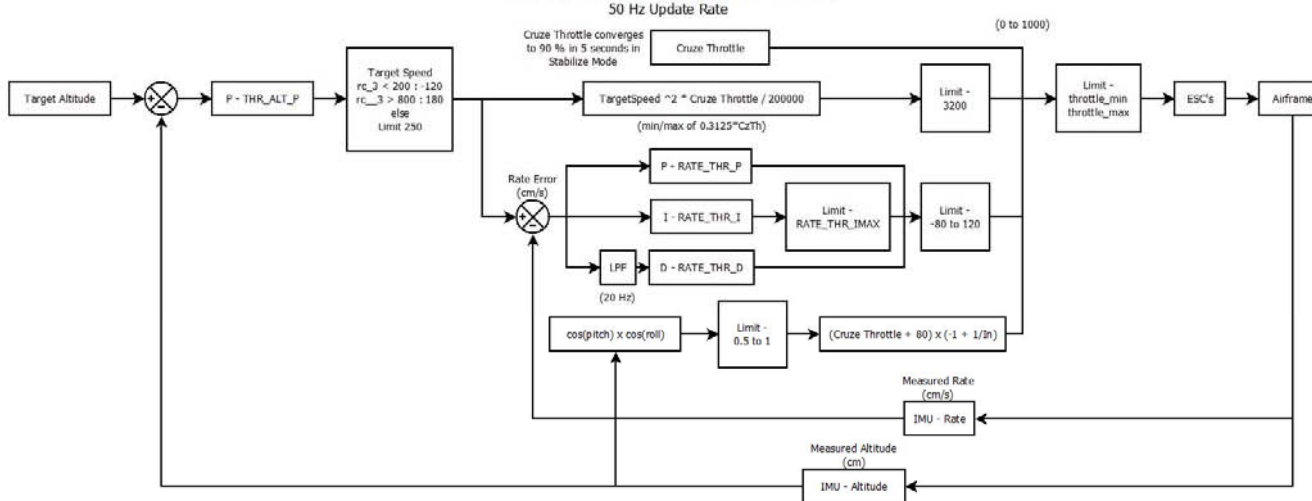




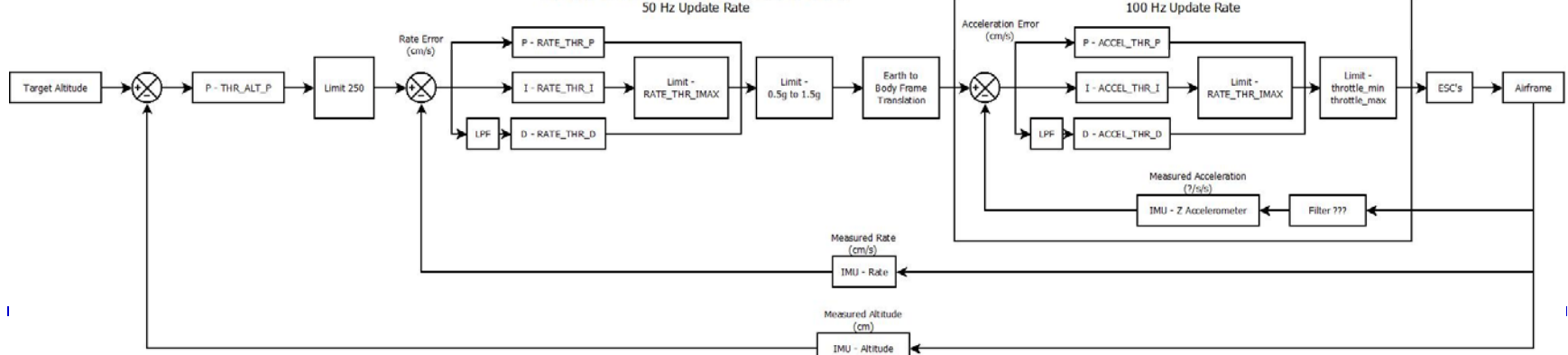
7. 飞控原理与调参

高度控制

CURRENT ALT HOLD PID's



NEW IDEA ALT HOLD PID's





7. 飞控原理与调参

□ Mission Planner调参界面





7. 飞控原理与调参

□ PID控制简介

PID(Proportional Integral Derivative)控制是最早发展起来的控制策略之一，由于其算法简单、鲁棒性好和可靠性高，被广泛应用于工业过程控制，尤其适用于可建立精确数学模型的确定性控制系统。正因为它的简单易用性，因而成为了最早实用化的控制器，应用近百年仍然处于无法取代的位置。

PID控制本质上是一个函数，输入是期望的参考轨线，而输出控制信号，该控制信号再输入到被控系统中，使得被控系统的输出能够较好地达到期望轨线。

连续形式：

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

离散形式：

$$u(k) = K_p e(k) + K_I \sum_{i=0}^k e(i) + K_D [e(k) - e(k-1)]$$



7. 飞控原理与调参

□ PID框图与调参

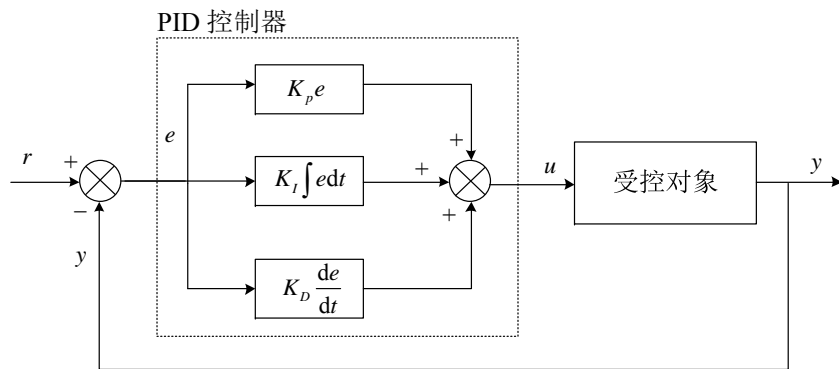


图. PID控制框图

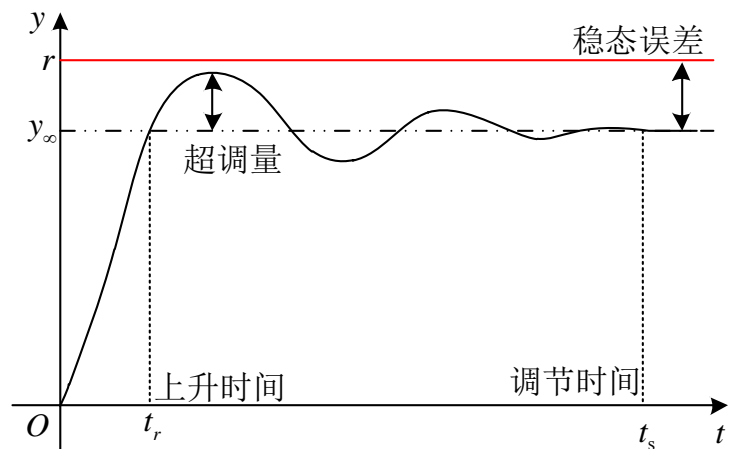


图. 系统阶跃响应曲线

效果指标:

1)上升时间(Rise Time) : 受控对象的输出 从0到第一次增加到稳态输出值 的所消耗的时间(或输入y从10% 增加到90% 所消耗时间)。

2)超调量(Overshoot) : 在响应过程中, 超出稳态值的最大偏离量与稳态值之比, 即

$$\sigma\% = \frac{y_{\max} - y_{\infty}}{y_{\infty}} \times 100\%$$

3)调节时间(Settling Time) : 输出曲线最终收敛于稳态值(5%以内)所用的时间。

4)稳态误差(Steady-state Error) : 稳态值与参考信号输入值之差, 即 $e_{ss} = r - y_{\infty}$ 。



7. 飞控原理与调参

□ PID调参目标与参数意义

任何闭环控制系统的首要任务是要稳（稳定）、快（快速）、准（准确）的响应命令，而“稳快准”会直接反映在上述定义四个指标上。快速性要求上升时间和调节时间尽量短，稳定性要求超调量小无震荡，准确性要求稳态误差尽量小。PID调整的主要工作就是如何实现这一任务。

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

K_p 比例增益。优点：调整系统的开环比例系数，提高系统的稳态精度，加快响应速度；缺点：过大会使系统的超调量增大，而且会使系统稳定裕度变小，甚至不稳定。

K_I 积分增益。优点：消除稳态误差；缺点：影响系统的稳定性，使系统的稳定裕度减小。

K_D 微分增益。优点：使系统的响应速度变快，超调减小，振荡减轻，对动态过程有“预测”作用；缺点：对干扰噪声敏感，使系统抑制干扰的能力降低。一般需要增加惯性滤波环节。



7. 飞控原理与调参

□ PID调节准则

	上升时间	超调量	调节时间	稳态误差	系统稳定性
K 增加	减小	增加	小幅增加	减小	下降
K_I 增加	小幅减小	增加	增加	大幅减小	下降
K_D 增加	小幅减小	减小	减小	几乎不变	提高

更形象的说，**P**——比例控制系统的响应快速性，快速作用于输出，好比“现在”（现在就起作用，快），**I**——积分控制系统的准确性，消除过去的累积误差，好比“过去”（清除过去积怨，回到准确轨道），**D**——微分控制系统的稳定性，具有超前控制作用，好比“未来”（放眼未来，未雨绸缪，稳定才能发展）。当然这个结论也不可一概而论，只是让大家更加快速的理解PID的作用。在调整的时候，你所要做的任务就是在系统结构允许的情况下，在这三个参数之间权衡调整，达到最佳控制效果，实现稳快准的控制特点。



7. 飞控原理与调参

□ PID中积分饱和和

积分饱和(Integral windup): 如果执行机构已经到极限位置, 仍然不能消除偏差时, 由于积分作用, 尽管PID差分方程式所得的运算结果继续增大或减小, 但执行机构已无相应的动作, 这就叫积分饱和。

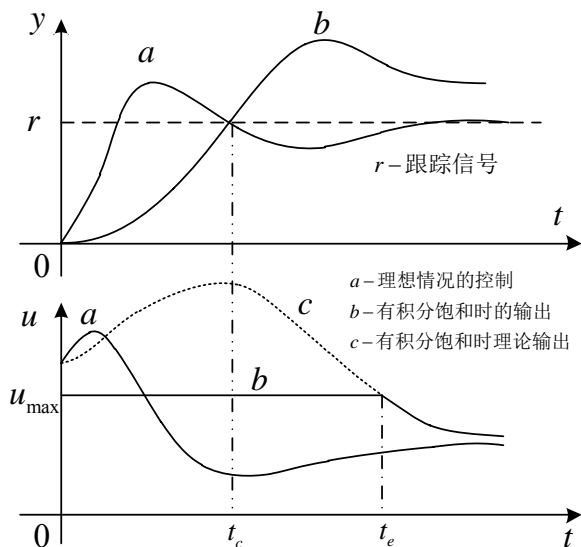


图. PID算法积分饱和示意图

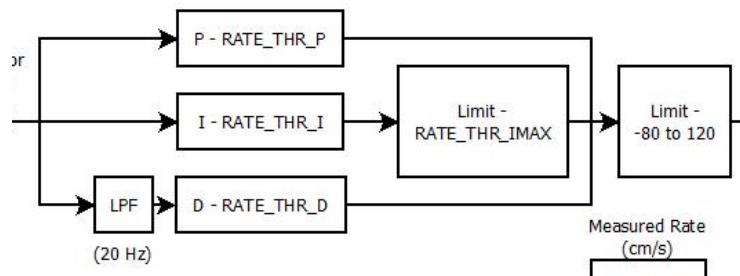


图. PixHawk部分PID算法截图

PixHawk飞控代码, 每个积分项都会增加饱和函数来抑制积分饱和。每个微分项都会增加滤波函数来提高抗干扰能力



7. 飞控原理与调参

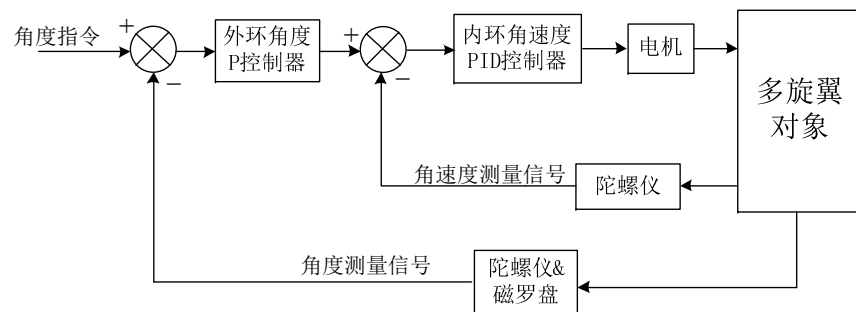
□ PID调参一般思路

- (1) 把P、I和D参数都归零或取固件默认值
- (2) 逐步增大P，一直到输出响应发生振荡，再稍微减小一点P。
- (3) 稍微加入一点积分信号，用于修正存在的稳态误差（一般在偏航通道和定高模式会出现稳态误差）。
- (4) 加入少量的D看看效果。注意：一些飞控板会尽量避免使用D，因为微分项对测量噪声非常敏感，在传感器测量信号本身有较大噪声，且后期信号滤波处理并不好的情况下，应该尽量减小D的使用，否则反而会造成系统的不稳定。
- (5) 如果加入D后对输出响应有改善效果则可以适当增加D，同时调整P使得上升时间较小且超调较小或无超调。
- (6) 反复调整P、I和D的值，直到输出响应达到最佳效果。



7. 飞控原理与调参

□ PixHawk 姿态PID调参方法



- (1) 用飞控自带的参数进行简单试飞（做好安全防护措施），记录飞行操作现象，分析飞控log文件，获取姿态响应曲线
- (2) 调整内环角速度P增益，使得响应快速，无震荡，无超调。
- (3) 略微调整内环角速度I和D增益，进一步优化响应，适当返回去调整速度P增益
- (4) 同理调整偏航内环角速度P增益，接着调整内环I和D增益，最后调整外环P增益
- (5) 根据飞行效果，针对性地微调各个系数



7. 飞控原理与调参

□ PixHawk 姿态PID调参方法

AC 3.0 maiden flight to-do:

1. Tune Roll-Pitch gains
2. Check vibration levels
3. Set throttle mid

视频. ArduCopter 3.0 官方姿态PID调参视频 <https://youtu.be/NOQPrTdrQJM>



7. 飞控原理与调参

□ 地面站Log文件分析与调参

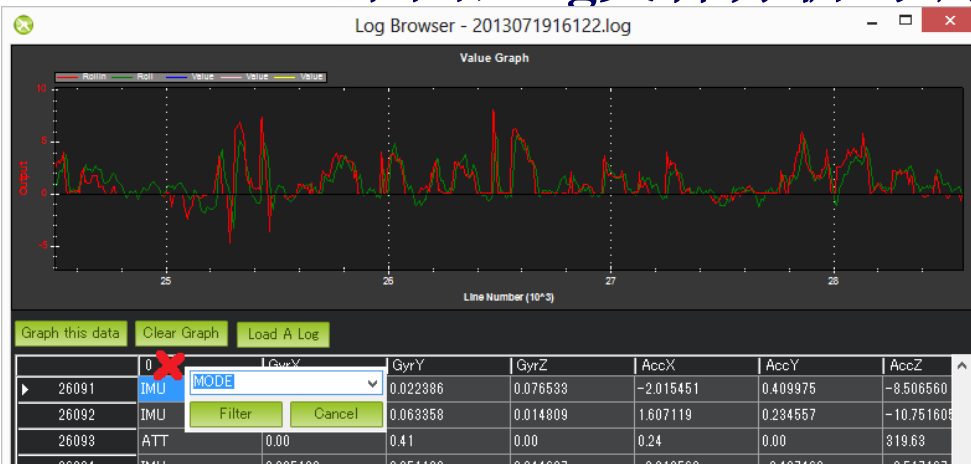


图. MissPlanner 姿态响应分析

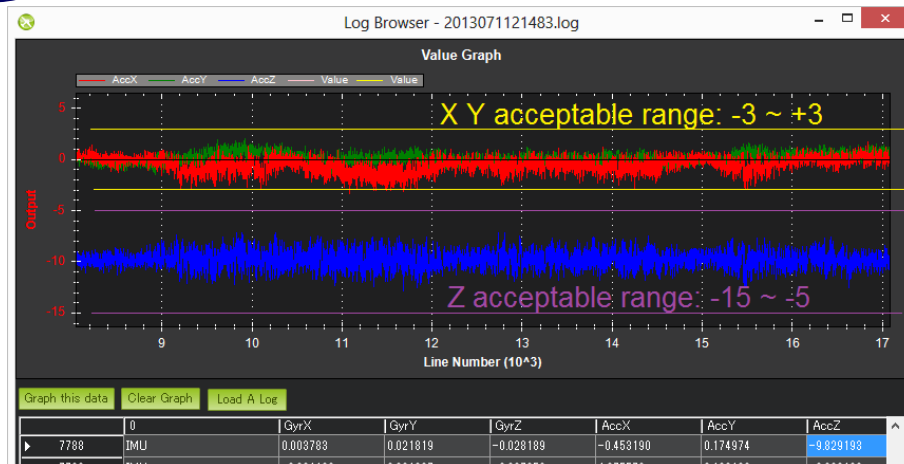


图. 加速度（振动）分析



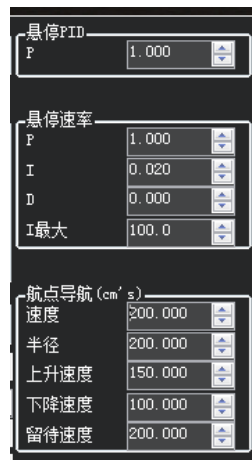
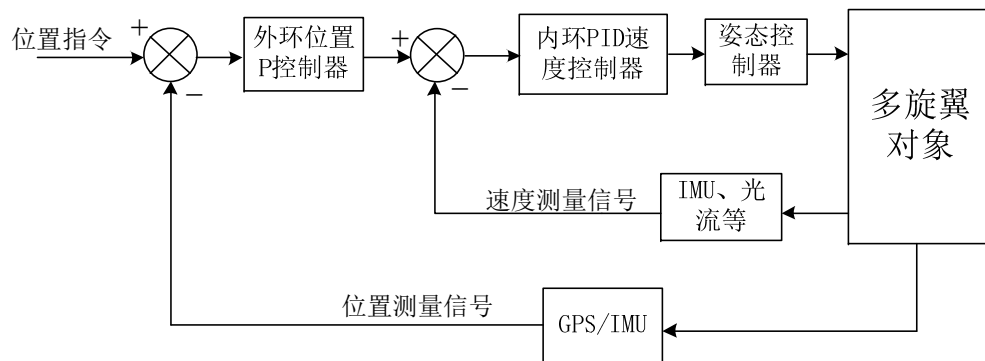
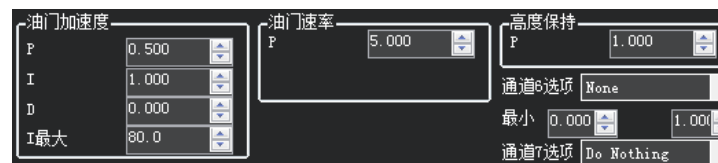
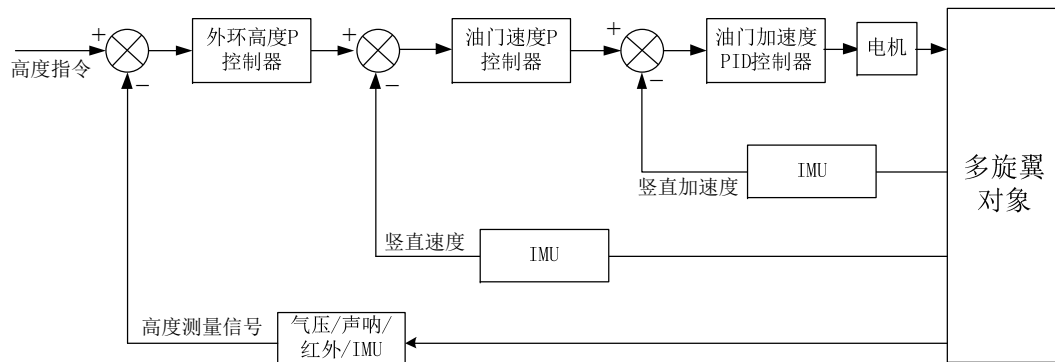
图. 故障分析





7. 飞控原理与调参

□ PixHawk 高度与悬停PID调参



高度调参在定高模式下，位置调参在悬停模式下，调整原则大同小异，从分理解PID各个参数的作用，分析飞行现象与响应曲线，做有针对性地调整



8. 总结

1. 多旋翼控制器的设计基于特定的控制要求，在不同的控制要求下需要选用不同的数学模型和控制方法。
2. 多旋翼的控制器设计是相互嵌套的，最终的控制系统包含多个闭环反馈，各部分相互协调共同实现多旋翼的飞行控制。
3. 对一流的多旋翼来说，这是不够的。对于所有多旋翼都希望**减小振动和节省能量**；对于如海上起飞的多旋翼都希望具有一定的**抗风性**；对于用于拍摄的多旋翼，特别是无云台的多旋翼，需要**高精度悬停**；对于多旋翼表演等场合，需要多旋翼实现大机动飞行。那么，人们不禁会问：如何设计底层控制，使得多旋翼最大可能减少多旋翼振动、节省能量、在大风中起飞、高精度悬停、大机动飞行？最终，**我们需要进一步从实际问题出发，研究和设计适当的底层控制器。**





9. 作业

选做一题：

- 1、第六讲中给出了多旋翼的控制分配模型，结合本节介绍的自驾仪中控制分配矩阵的实现，在已知在参数为：初始情况 $\mathbf{J} = \text{diag}(J_{x,0}, J_{y,0}, J_{z,0})$, $m = m_0$, $d = d_0$, $c_T = c_{T0}$, $c_M = c_{M_0}$ 的情况下，针对虚拟的力和力矩有如下参数的 PD 控制器为

$$\begin{bmatrix} f_0 \\ \tau_{x,0} \\ \tau_{y,0} \\ \tau_{z,0} \end{bmatrix} = \begin{bmatrix} f_0^* + k_{z,p,0}(z - z_d) + k_{z,d,0}\dot{z} \\ k_{\tau_x,p,0}(\phi - \phi_d) + k_{\tau_x,d,0}\omega_x \\ k_{\tau_y,p,0}(\theta - \theta_d) + k_{\tau_y,d,0}\omega_y \\ k_{\tau_z,p,0}(\psi - \psi_d) + k_{\tau_z,d,0}\omega_z \end{bmatrix}$$

请问在参数为新情况 $\mathbf{J} = \text{diag}(J_{x,1}, J_{y,1}, J_{z,1})$, $m = m_1$, $d = d_1$, $c_T = c_{T_1}$, $c_M = c_{M_1}$ ，要达到如同以上控制器的控制效果，PD 控制器为应该如何改进？

- 2、自拟一道与本章节内容相关的题，并回答。（难度与上面题目相当，不能抄袭其他书籍）



下一讲预告

给定期望的位置，如何基于半自主飞控平台控制多旋翼使其能够飞到目标位置？



资源

(1) 课程中心 (课件、资料、作业等)

(2) 可靠飞行控制研究组主页 (课件等)

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

(3) 关注可靠飞行控制研究组公众号 buaarfly (课件等)

(4) 多旋翼设计与控制交流QQ群 183613048





谢谢！

感谢博士生 任锦瑞、戴训华 为本节课课件准备做出的贡献。

如课件有错误之处，请反馈到邮箱 qq_buaa@buaa.edu.cn