



多旋翼飞行器设计与控制 实践

第九讲 姿态控制器设计实验

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



北航可靠飞行控制研究组

BUAA Reliable Flight Control Group



大纲

1. 实验原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 小结



实验原理

□ 基本概念

(1) 系统时域特性

对二阶系统
$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

其中 $0 < \zeta < 1$ ，阶跃响应曲线如左图所示

1) 超调量
$$M_p = \frac{x_o(t_p) - x_o(\infty)}{x_o(\infty)} * 100\% = e^{-\xi\pi/\sqrt{1-\xi^2}} * 100\%$$

2) 调节时间

在初步分析系统时，经常采用下列公式计算调节时间 t_s 。当阻尼比 < 0.8 时

$$t_s = \frac{3.5}{\xi\omega_n} \text{ (取5\%误差带)} \quad t_s = \frac{4.5}{\xi\omega_n} \text{ (取2\%误差带)}$$

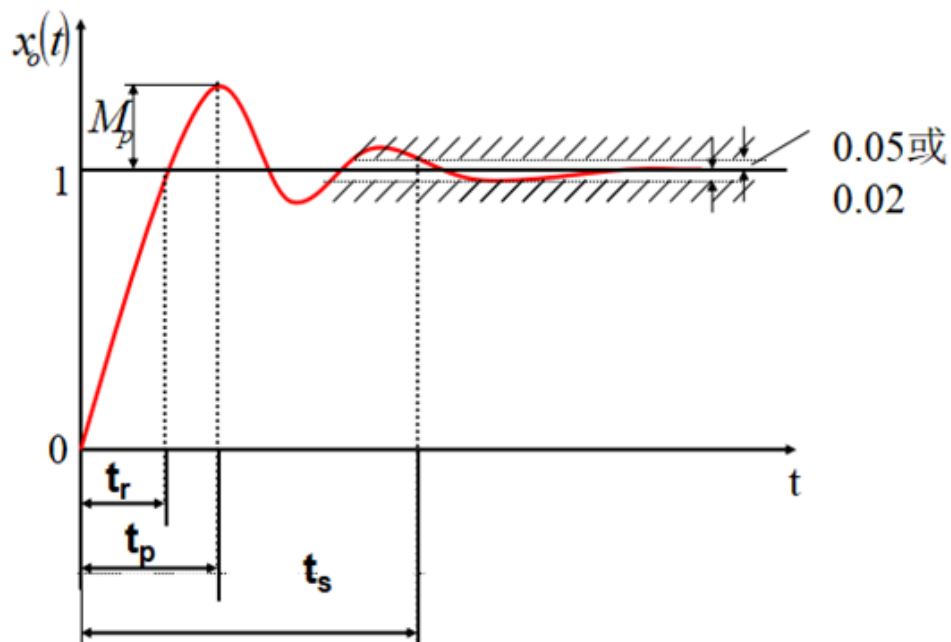


图. 二阶系统阶跃响应



实验原理

□ 基本概念

(2) Bode图 and 稳定裕度

Bode图也叫对数频率特性曲线，它将开环幅相特性画在对数坐标上。对数稳定判据根据开环对数幅频与对数相频曲线的相互关系来判别闭环系统稳定性。

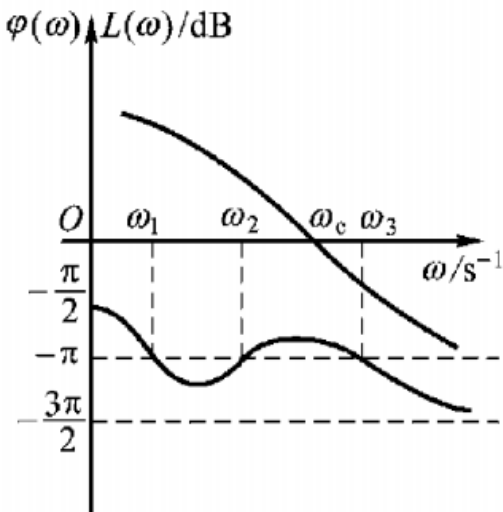


图. 稳定裕度

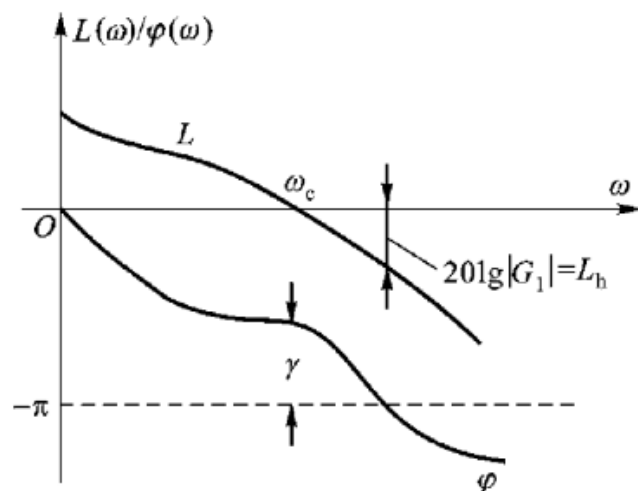


图. 对数频率特性曲线

相角裕度 γ : $L(\omega) = 0$ dB 处的相频与 $-\pi$ 的差角, 即:

$$\gamma = \angle G(j\omega_c)H(j\omega_c) - (-180^\circ)$$

其中: ω_c 表示截止频率, $L(\omega_c) = 0$ dB。

模稳定裕度 h : $\angle G(j\omega_1)H(j\omega_1)$ 为 $-\pi$ 时, 对应的对数幅频的绝对值, 即:

$$h(\text{dB}) = 20 \lg \left| \frac{1}{G(j\omega_1)H(j\omega_1)} \right| = -20 \lg |G(j\omega_1)H(j\omega_1)|$$

在闭环系统稳定的条件下, 系统的 h 越大, 反应系统的稳定程度越高。稳定裕度也间接反映了系统动态过程的平稳性, 裕度大意为着超调小, 振荡弱, “阻尼”大。一般要求:

$$\gamma > 40^\circ$$

$$h > 6\text{dB}$$



实验原理

□ 底层控制框架

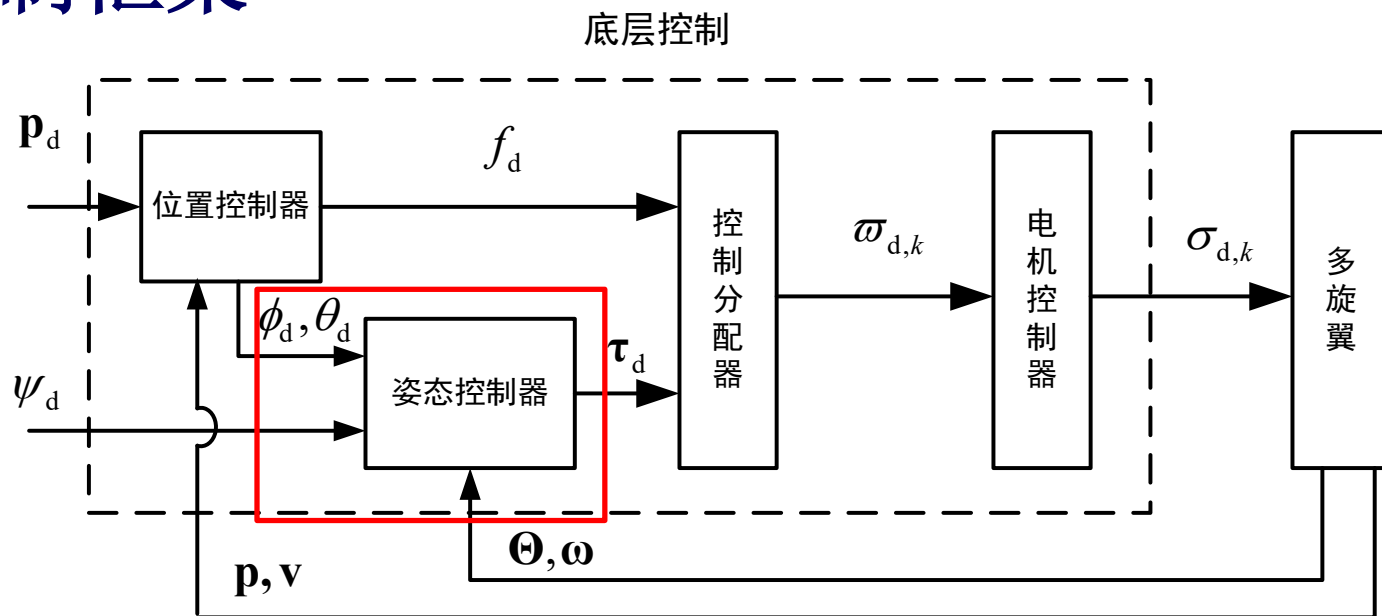


图. 多旋翼全自主控制闭环框图

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。



实验原理

□ 姿态控制

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



实验原理

□ 姿态控制

姿态控制的目标: 已知参考姿态角 $\Theta_d = [\Theta_{hd}^T \ \psi_d]^T$, 设计控制器使得 $\lim_{t \rightarrow \infty} \|\mathbf{e}_\Theta(t)\| = 0$, 其中 $\mathbf{e}_\Theta \triangleq \Theta - \Theta_d$ 。这里 Θ_{hd} 、 ψ_d 是由遥控器给定的。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

$$\omega_d = -\mathbf{K}_\Theta \mathbf{e}_\Theta$$

其中 $\mathbf{K}_\Theta \in \mathbb{R}^{3 \times 3}$ 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

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

设计转矩的期望 τ_d

$$\tau_d = -\mathbf{K}_{\omega p} \mathbf{e}_\omega - \mathbf{K}_{\omega i} \int \mathbf{e}_\omega - \mathbf{K}_{\omega d} \dot{\mathbf{e}}_\omega$$

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



实验原理

□ 控制系统校正

下面主要介绍串联校正。加入串联校正的系统结构图如图所示。其中 $G_c(s)$ 表示串联校正装置的传递函数， $G(s)$ 表示系统不变部分的传递函数。在工程实践中常用的串联校正有超前校正、滞后校正和滞后-超前校正。

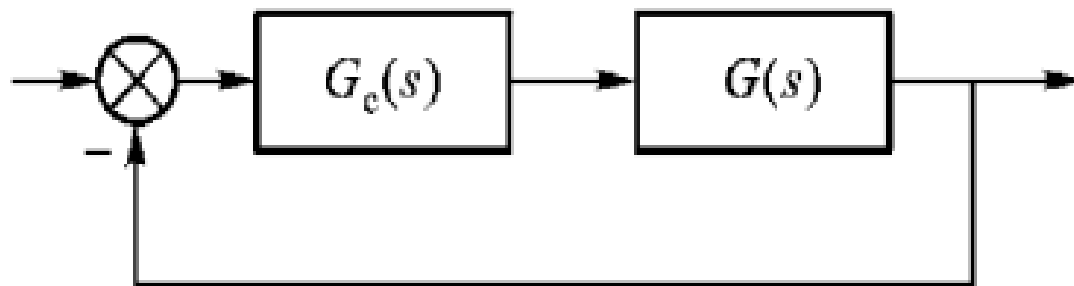


图. 系统串联校正



实验原理

□ 控制系统校正

(1) 相位超前校正

$$G_c(s) = \frac{1 + aTs}{1 + Ts} \quad (a > 1)$$

特点: 提供正的相移。

相位超前校正主要发生在频段 $\left(\frac{1}{aT}, \frac{1}{T}\right)$

而且超前的最大值为: $\varphi_m = \arcsin \frac{a-1}{a+1}$

这一最大值发生在对数频率特性曲线的几何中心处, 对应的角频率为:

$$\omega_m = \frac{1}{\sqrt{aT}}$$

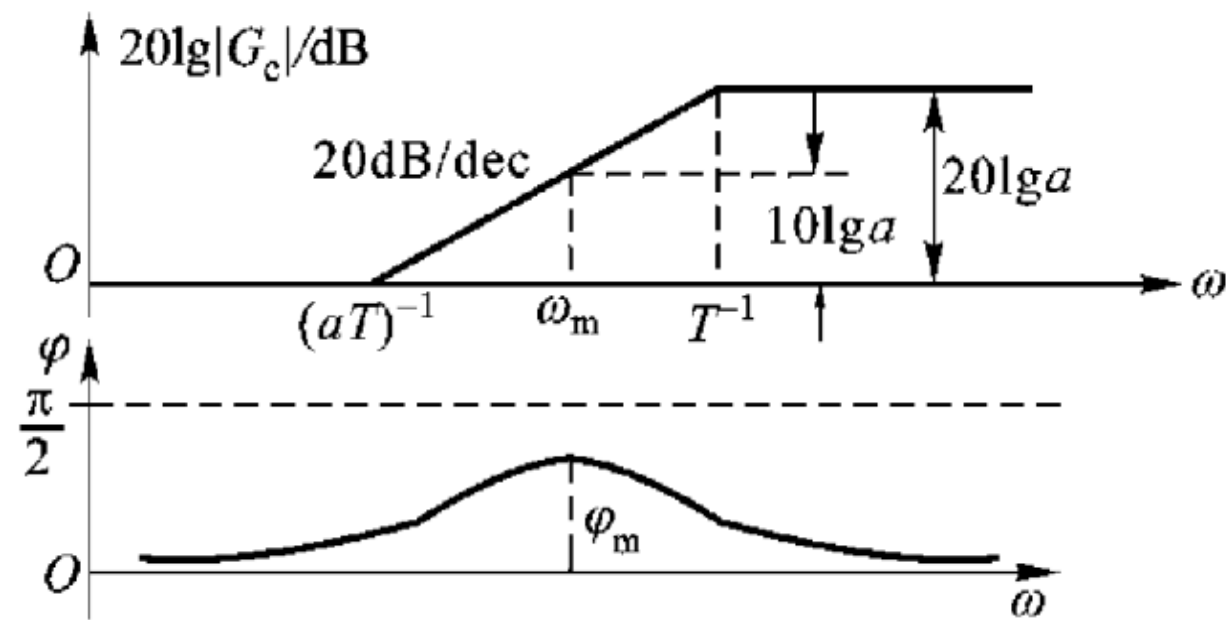


图. 超前校正幅频曲线



实验原理

□ 控制系统校正

(2) 相位滞后校正

$$G_c(s) = \frac{1 + bTs}{1 + Ts} \quad (b < 1)$$

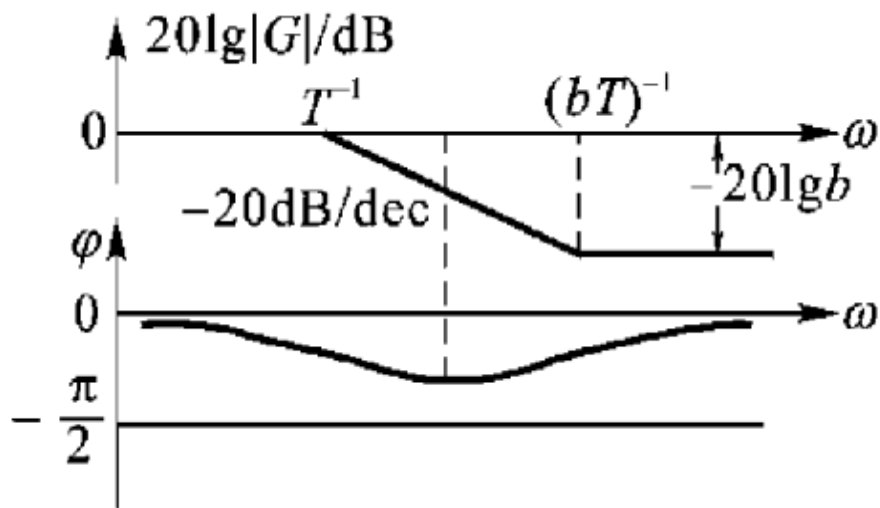


图. 滞后校正幅频曲线

(3) 相位滞后超前校正

$$G_c(s) = \frac{(1 + bT_1s)(1 + aT_2s)}{(1 + T_1s)(1 + T_2s)}$$

其中 $a > 1, b > 1$, 且有 $bT_1 > aT_2$

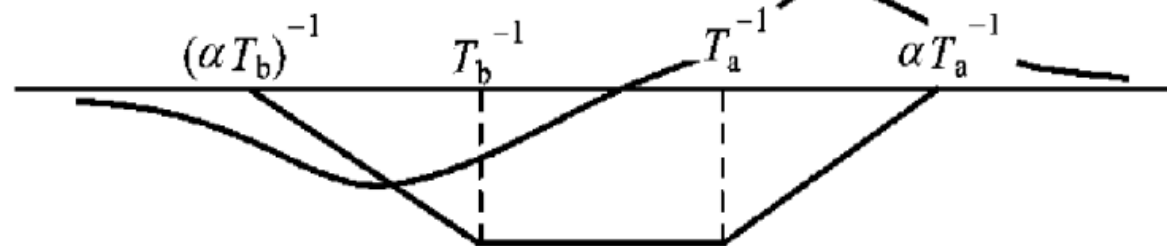


图. 滞后超前校正幅频曲线



实验原理

□ 控制分配

简单来说，控制分配问题可描述为：给定 $\mathbf{u}_v(t)$

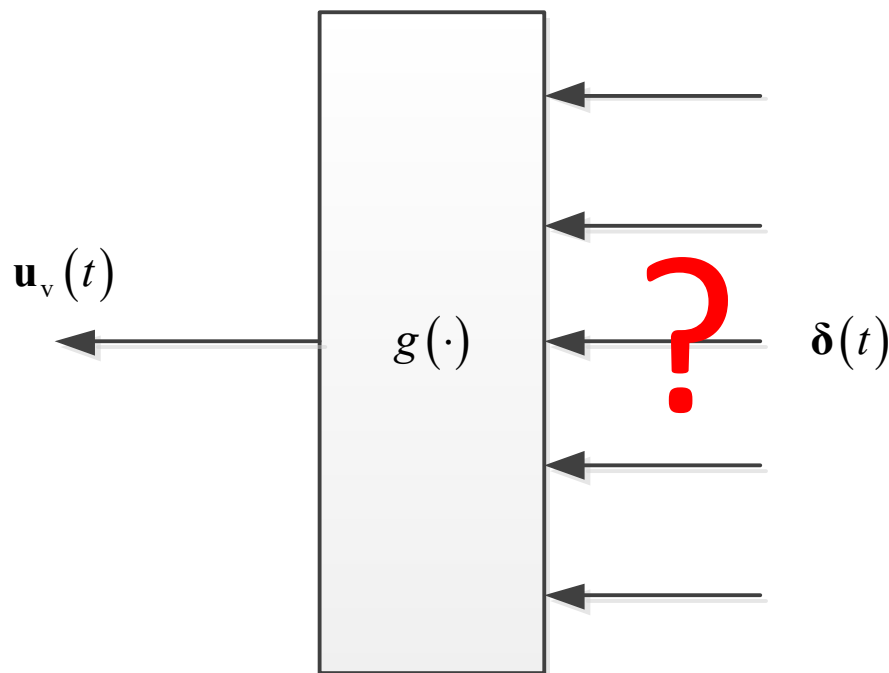
寻找 $\delta(t)$ 使得

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

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。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。



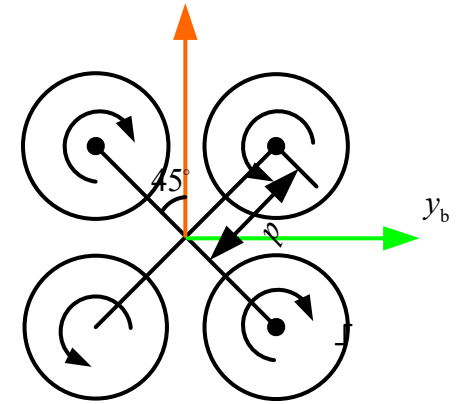
- 上层控制和底层控制分离
- 有效分配，防止饱和
- 通过分配，提高对故障及损伤的鲁棒性
- ...



实验原理

控制分配

x字形四旋翼的控制分配和多旋翼的控制效率模型如下：



$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T \\ \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T \\ c_M & c_M & -c_M & -c_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & 0 & 0 & 0 \\ 0 & dc_T & 0 & 0 \\ 0 & 0 & dc_T & 0 \\ 0 & 0 & 0 & c_M \end{bmatrix} \begin{matrix} M_4 \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 1 & 1 & -1 & -1 \end{bmatrix} \end{matrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

$$\begin{cases} \omega_1^2 = \frac{f}{4c_T} - \frac{\sqrt{2}\tau_x}{4dc_T} + \frac{\sqrt{2}\tau_y}{4dc_T} + \frac{\tau_z}{4c_M} \\ \omega_2^2 = \frac{f}{4c_T} + \frac{\sqrt{2}\tau_x}{4dc_T} - \frac{\sqrt{2}\tau_y}{4dc_T} + \frac{\tau_z}{4c_M} \\ \omega_3^2 = \frac{f}{4c_T} + \frac{\sqrt{2}\tau_x}{4dc_T} + \frac{\sqrt{2}\tau_y}{4dc_T} - \frac{\tau_z}{4c_M} \\ \omega_4^2 = \frac{f}{4c_T} - \frac{\sqrt{2}\tau_x}{4dc_T} - \frac{\sqrt{2}\tau_y}{4dc_T} - \frac{\tau_z}{4c_M} \end{cases}$$

未知参数可以使用
控制器来进行补偿，

假设油门、俯仰、横
滚、偏航四个通道的
输出分别为

σ_1 、 σ_2 、 σ_3 、 σ_4

$$\begin{cases} \omega_1^2 = \sigma_1 - \sigma_3 + \sigma_2 + \sigma_4 \\ \omega_2^2 = \sigma_1 + \sigma_3 - \sigma_2 + \sigma_4 \\ \omega_3^2 = \sigma_1 + \sigma_3 + \sigma_2 - \sigma_4 \\ \omega_4^2 = \sigma_1 - \sigma_3 - \sigma_2 - \sigma_4 \end{cases}$$



实验原理

□ 控制分配

多旋翼的控制效率模型如下：

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

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



实验原理

□ 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 未知时。那这又是如何进行控制分配的呢？



实验原理

□ 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

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

已知参数 未知参数



实验原理

□ 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。

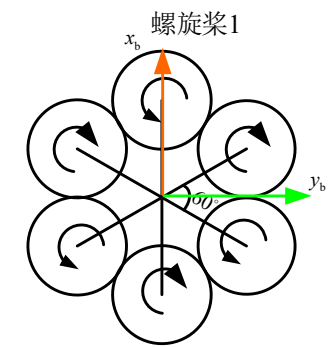


图. 常规布局的六旋翼



实验原理

以上原理可以详细参考“Quan Quan. *Introduction to Multicopter Design and Control*. Springer, Singapore, 2017”或者“全权著. 杜光勋、赵峙尧、戴训华、任锦瑞、邓恒译。《多旋翼飞行器设计与控制》，电子工业出版社，2018.”的第11章。



基础实验

□ 实验目标

■ 已知

- (1) 硬件：Pixhawk 自驾仪系统，多旋翼硬件系统。
- (2) 软件：MATLAB 2017b 或以上的版本，基于 Simulink 的控制器设计与仿真平台，硬件在环平台，实验指导包“e5.1”（下载地址：<https://rflysim.com/course>）。

■ 目标

- (1) 复现四旋翼飞行器的 Simulink 仿真，分析控制分配器的作用；
- (2) 记录姿态的阶跃响应，并对开环姿态控制系统进行扫频以绘制 Bode 图，分析闭环姿态控制系统的稳定裕度；
- (3) 完成四旋翼硬件在环仿真。



基础实验

□ 实验步骤

(1) 步骤一：软件在环仿真——控制分配器功能

1) 参数初始化

运行“e5\5.1\Init_control.m”文件进行参数初始化。同时Simulink文件“AttitudeControl_Sim.slx”将会自动打开，如图所示。

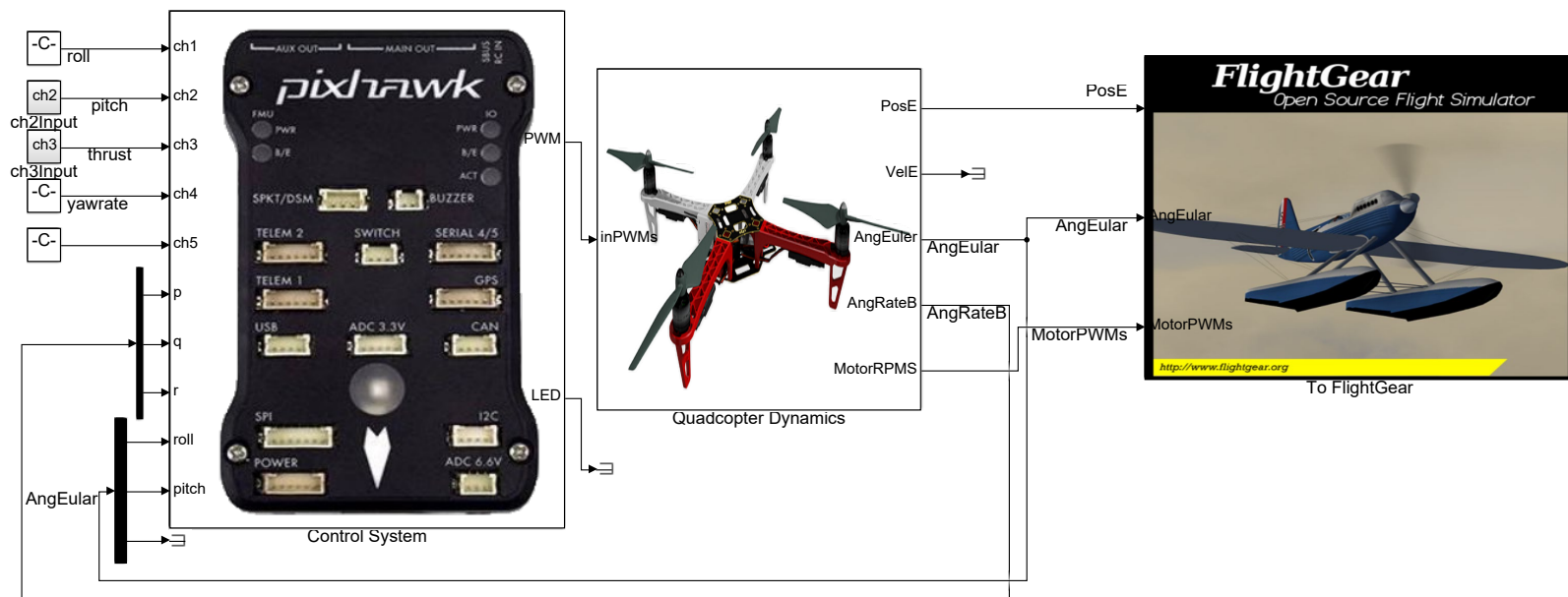


图. AttitudeControl_Sim.slx截图



基础实验

□ 实验步骤

2) 开始仿真

打开FlightGear-F450程序，然后点击Simulink“开始仿真”按钮开始仿真。此时可以在FlightGear中观察多旋翼的状态，如图。

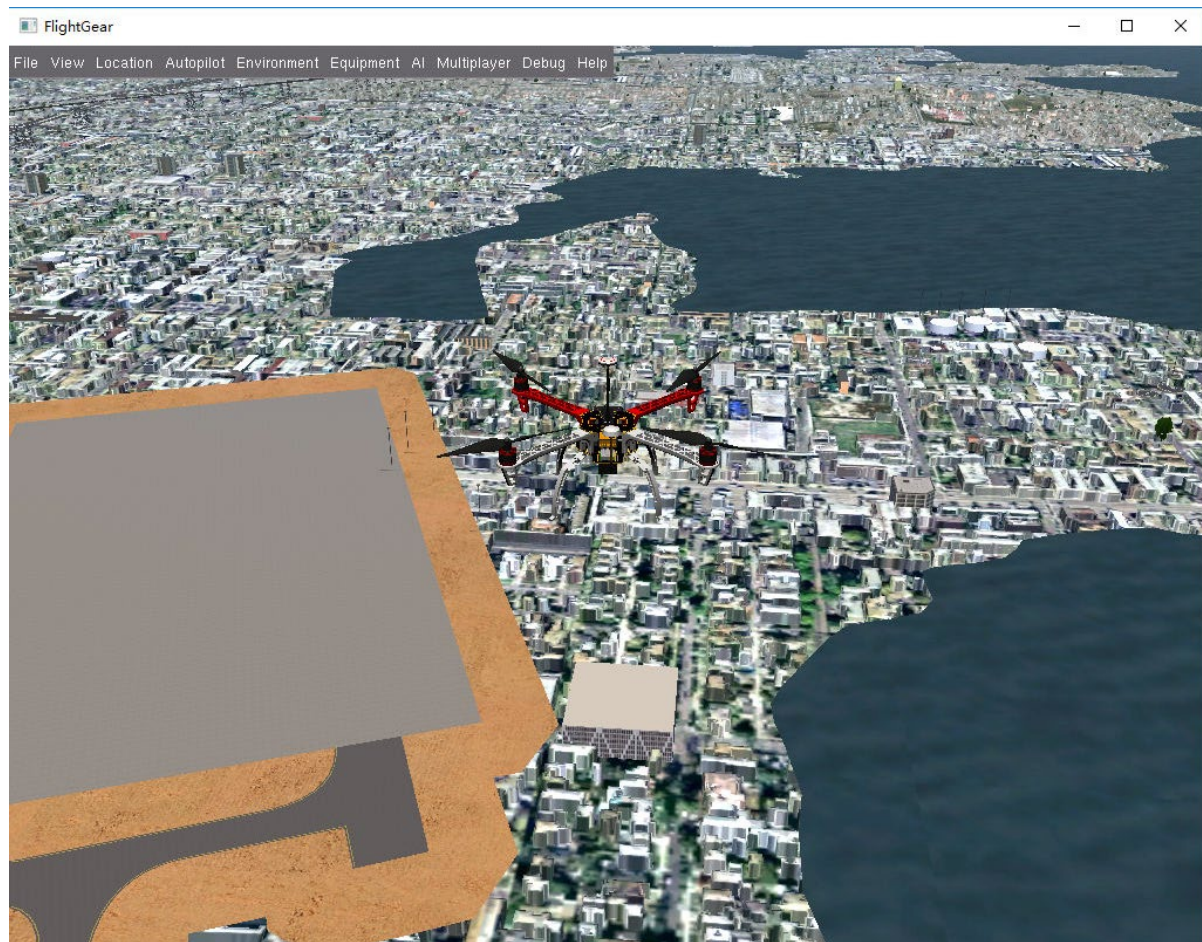


图. FlightGear中的多旋翼



基础实验

□ 实验步骤

3) 仿真效果

给pitch通道遥控器延迟输入1600，
得到相应曲线如图所示。

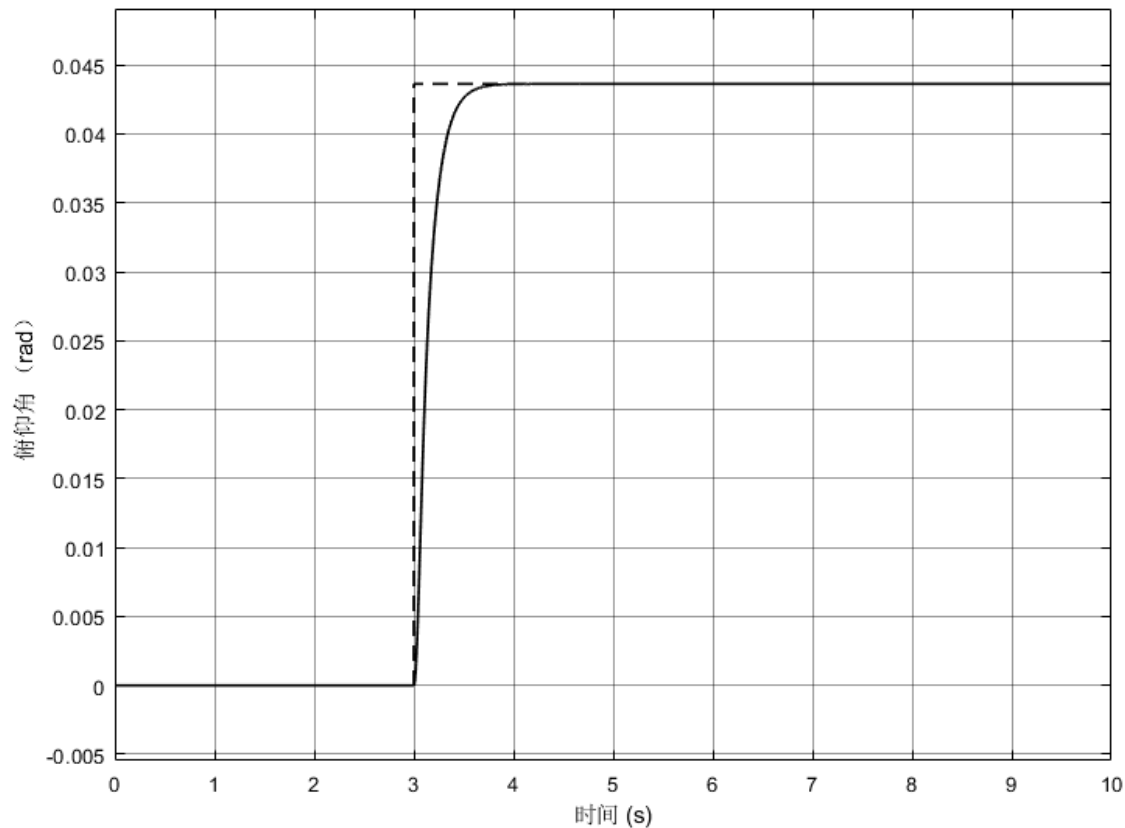


图. Pitch角度响应



基础实验

实验步骤

4) 控制分配器作用

分配器接收横滚、俯仰、偏航、油门的控制量分配给四个电机。

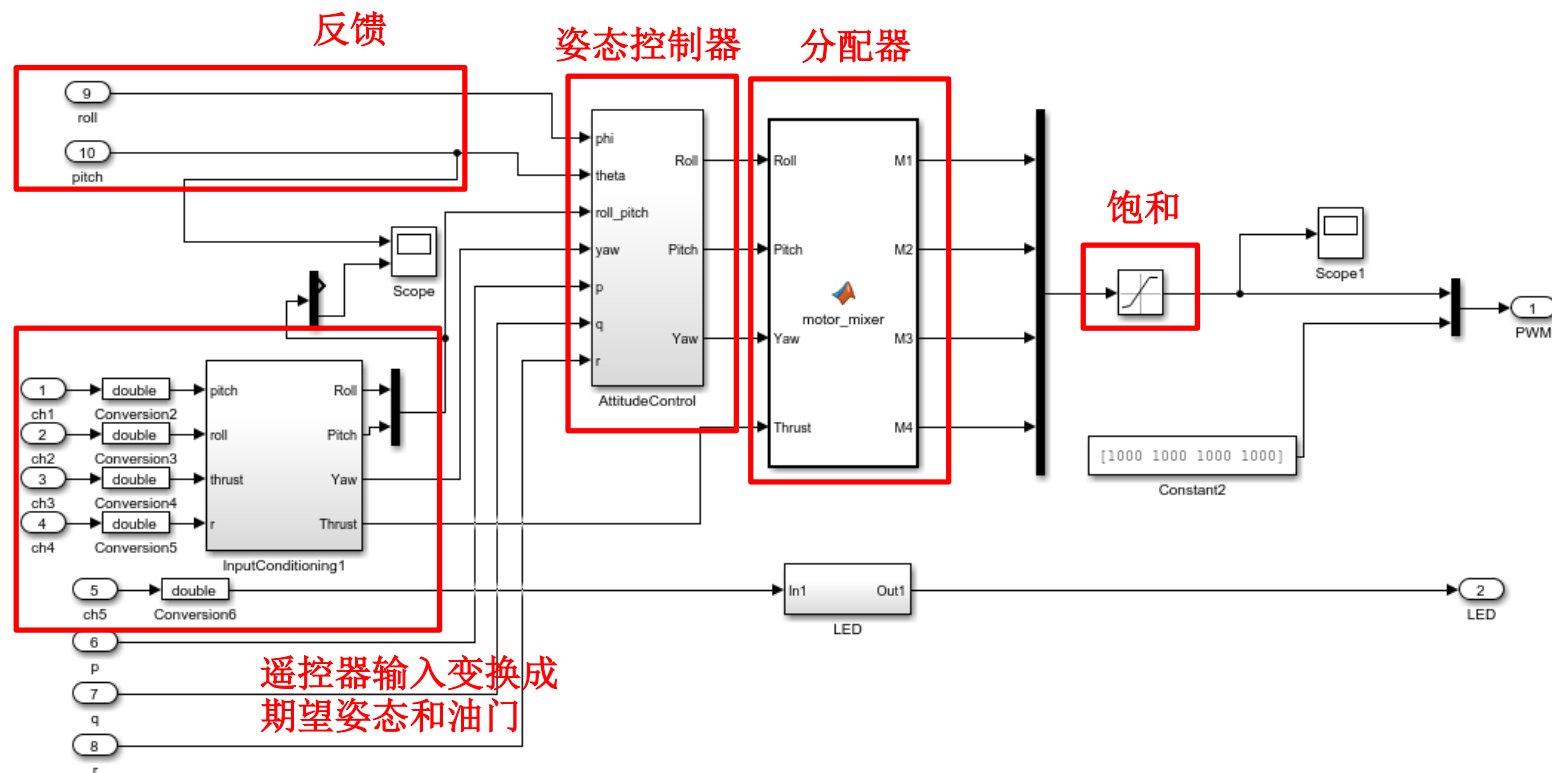


图. 控制分配器的位置



基础实验

□ 实验步骤

4) 控制分配器作用

当俯仰通道给值1600，在姿态调节阶段，控制器的输出为 $\tau_{dx} = 0, \tau_{dy} > 0, \tau_{dz} = 0, f_d = \text{constant}$ ，所以电机提供了 $o_b y_b$ 轴方向的力矩，螺旋桨 1 和螺旋桨 3 电机控制量增加，螺旋桨 2 和螺旋桨 4 电机控制量减少，与控制分配原理相符。

$$\omega_{d,1}^2 \propto (f_d - \tau_{dx} + \tau_{dy} + \tau_{dz})$$

$$\omega_{d,2}^2 \propto (f_d + \tau_{dx} - \tau_{dy} + \tau_{dz})$$

$$\omega_{d,3}^2 \propto (f_d + \tau_{dx} + \tau_{dy} - \tau_{dz})$$

$$\omega_{d,4}^2 \propto (f_d - \tau_{dx} - \tau_{dy} - \tau_{dz})$$

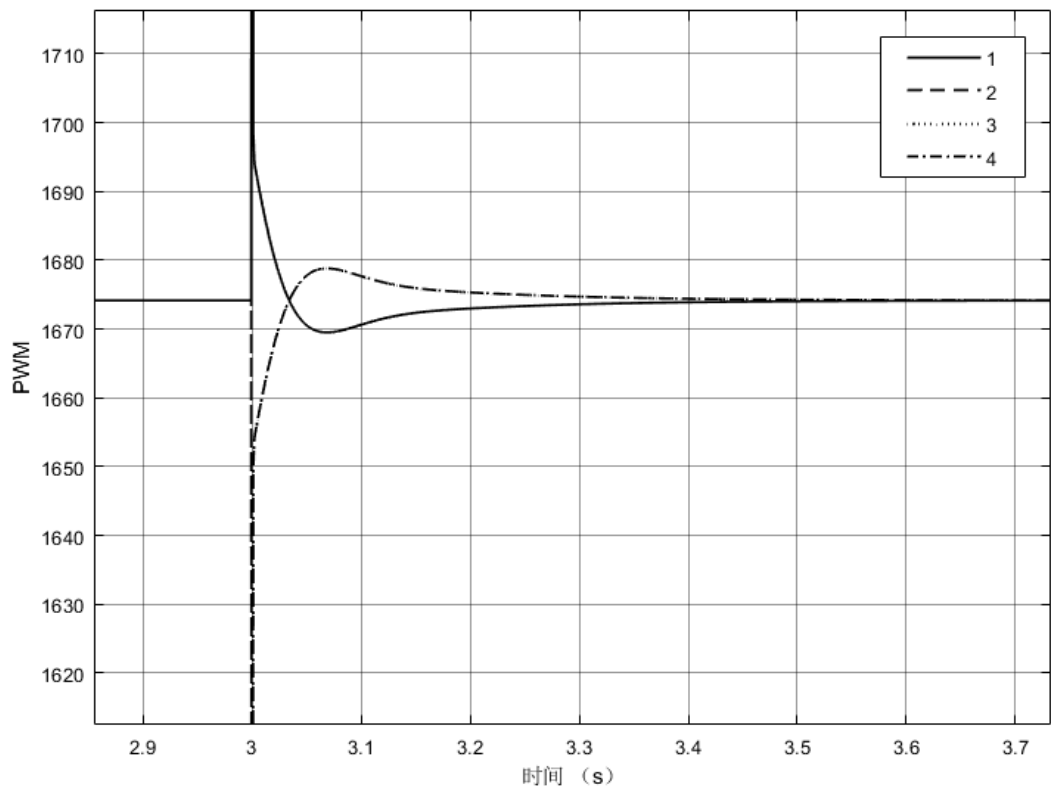


图 经过控制分配后的PWM值



基础实验

□ 实验步骤

(2) 步骤二：软件在环仿——稳定裕度

1) 打开文件

“e5\e5.1\tune\AttitudeControl_tune.slx” ,

如右图所示。

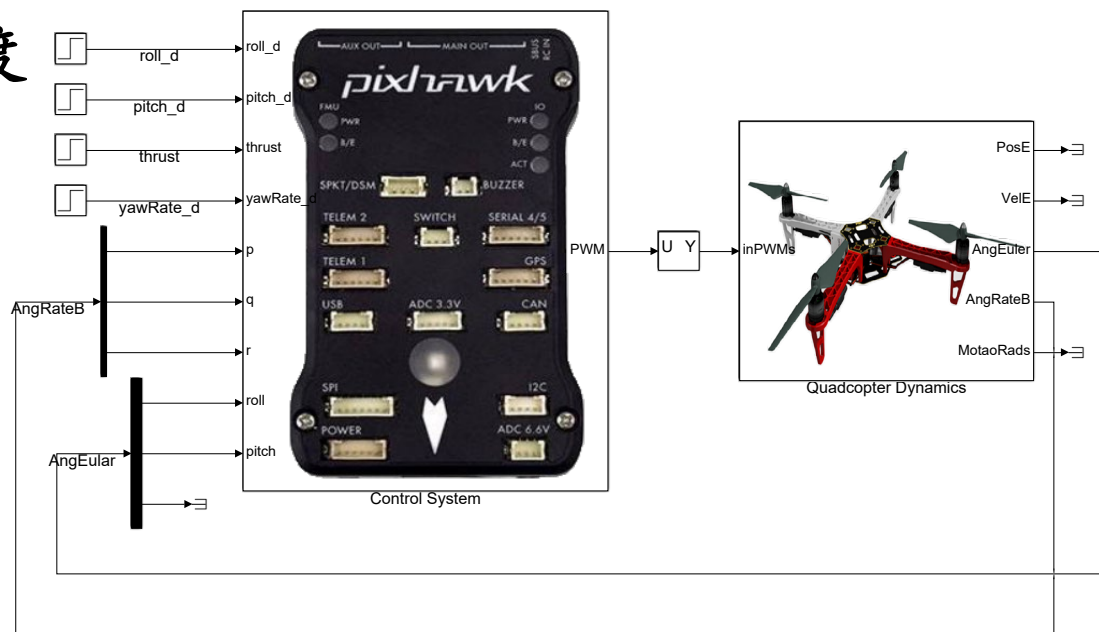


图. AttitudeControl_tune.slx截图



基础实验

实验步骤

2) 设置信号的输入输出点

在“Control System” - “AttitudeControl”子模块中可见如右图所示的姿态控制系统模型。

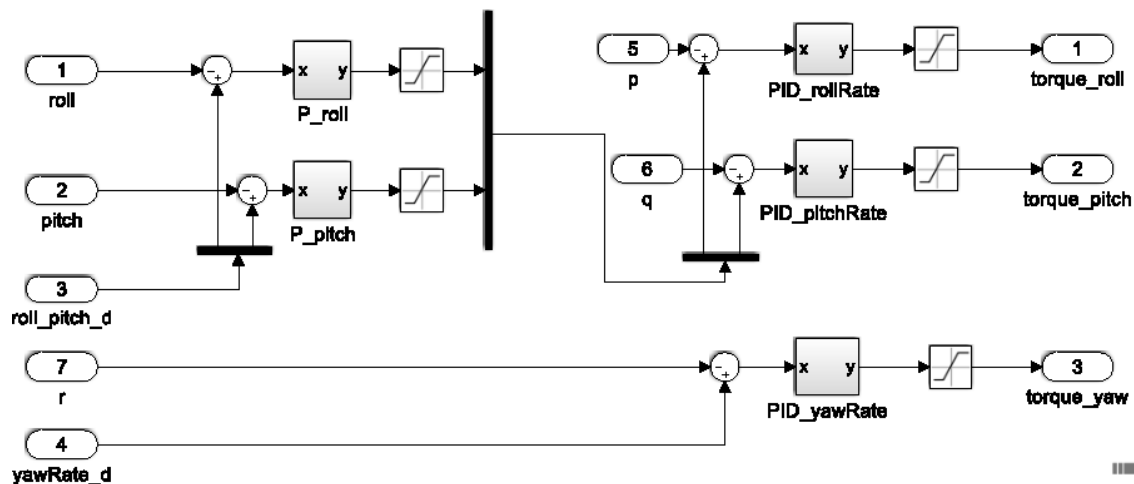


图. 姿态控制系统

姿态控制系统中有4个通道，即roll、pitch、yaw、thrust，这里设定thrust通道为定值0.6085，该值为悬停时的油门值。这里只对剩下的三个通道中的一个进行扫频，如pitch通道，其他通道的扫频步骤与此类似。



基础实验

实验步骤

2) 设置信号的输入输出点

- **输入点设置:** 选中信号线, 点击鼠标右键, 选择“Linear Analysis Points”-“Open-loop Input”。
- **输出点设置:** 与输入点设置步骤相同, 最后选择“Open-loop Output”。如右图所示。

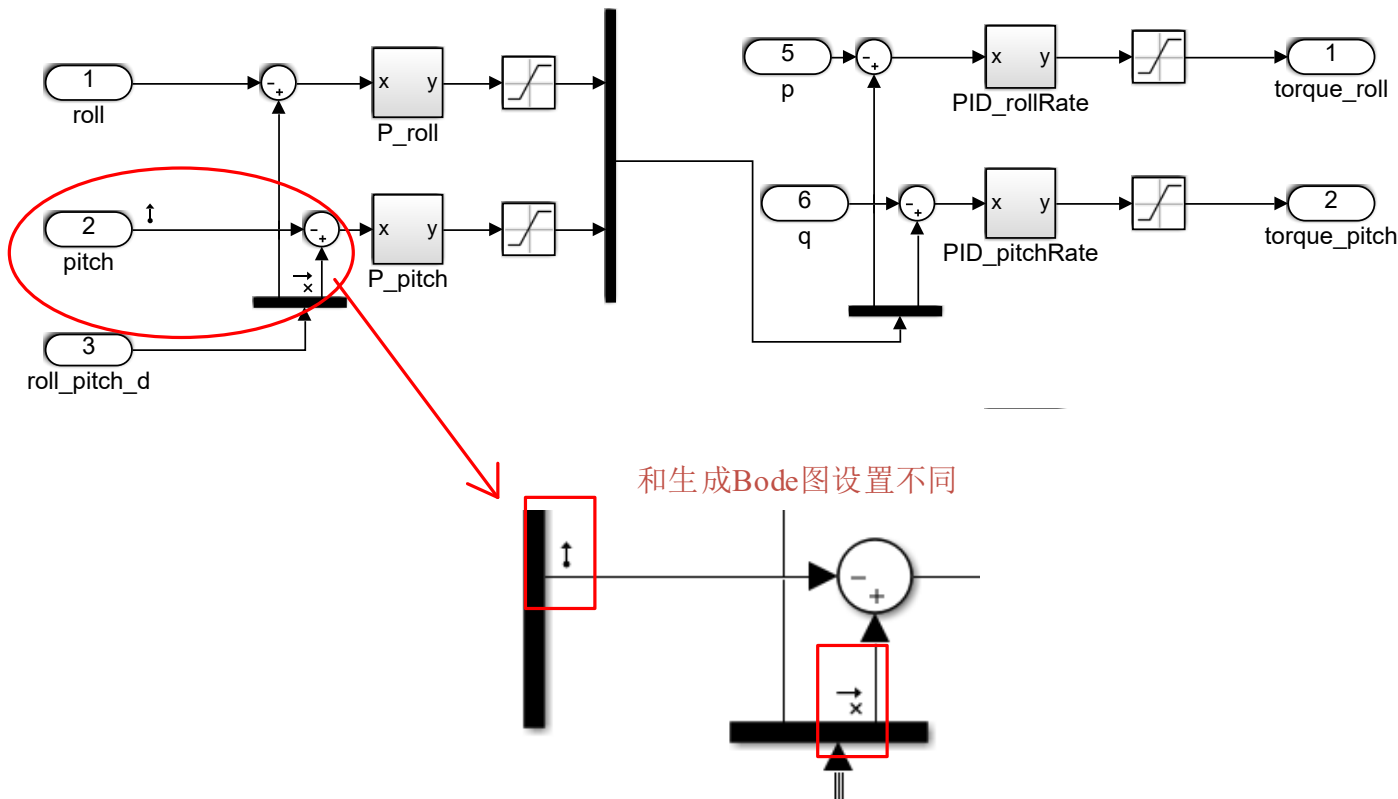


图. 设置输入输出点示例



基础实验

□ 实验步骤

3) 得到Bode图

选择Simulink 上面菜单中的“Analysis” - “Control Design” - “Linear Analysis”，如左图所示。

在弹出的窗口中选择“LINEAR ANALYSIS”，点击“Bode”，即可得到Bode图，如右图所示。

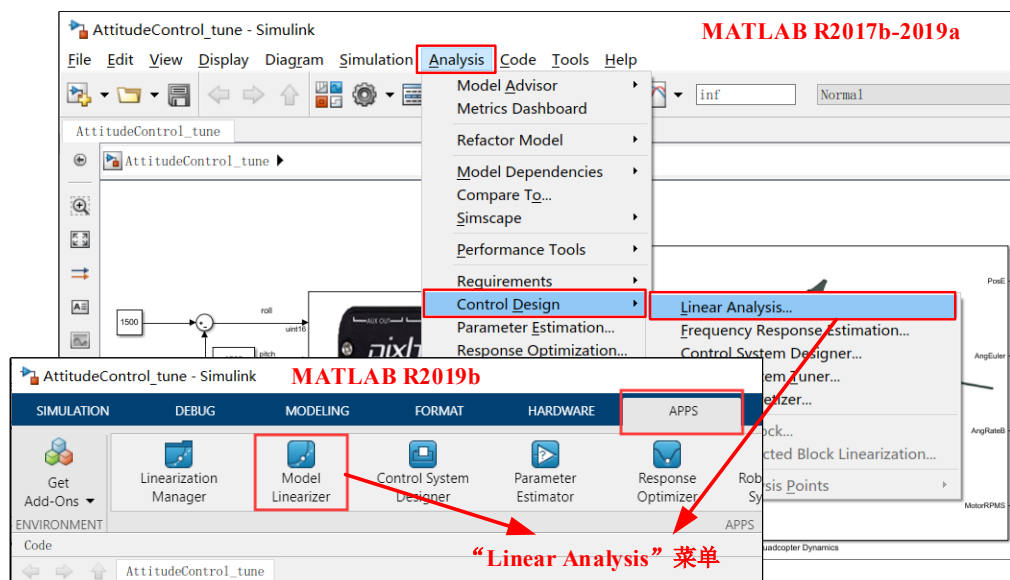


图. “Linear Analysis” 菜单路径

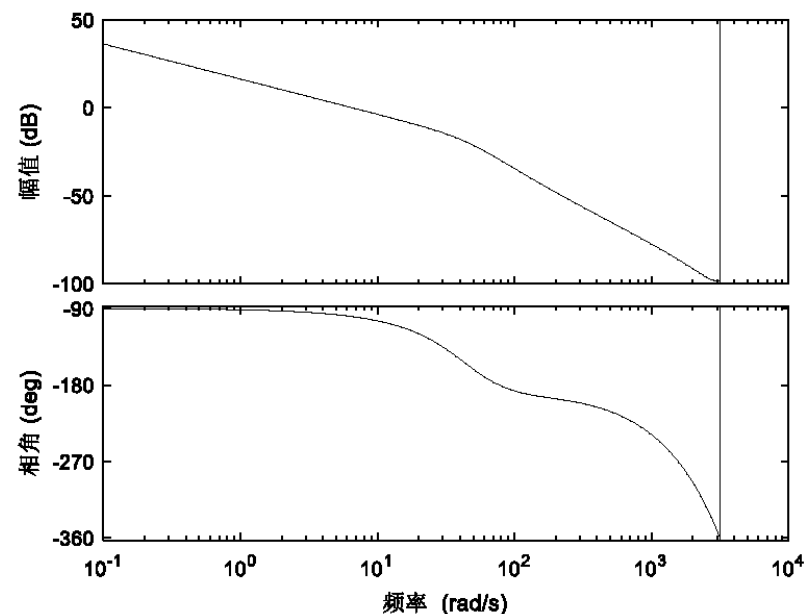
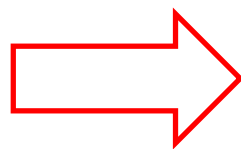


图. 扫频得到Bode图



基础实验

□ 实验步骤

(3) 步骤三：软件在环仿真——阶跃响应

1) 设置信号的输入输出点

设定信号的输入点：选中信号线，点击鼠标右键，选择“Linear Analysis Points” – “Open-loop Input”。

输出点设置：与输入点设置步骤相同，最后选择Output Measurement。

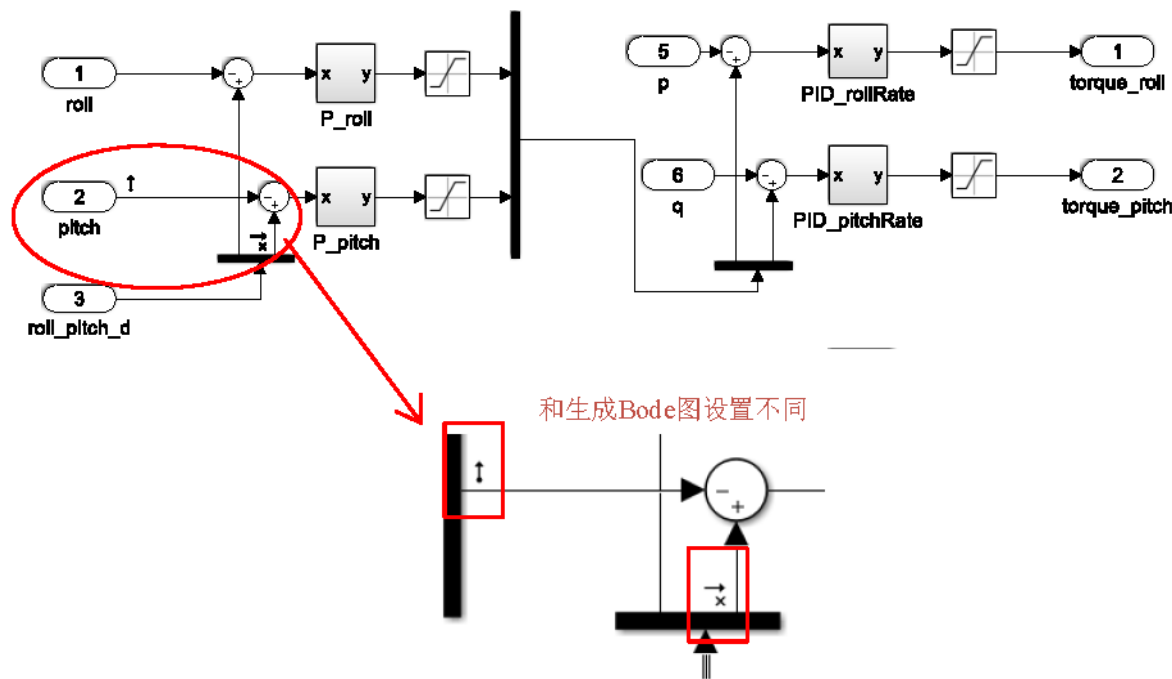


图. 阶跃响应曲线设置



基础实验

□ 实验步骤

2) 得到阶跃响应曲线

进入Simulink的“Linear Analysis”界面，
点击“Step”即可得到阶跃响应曲线，如右
图。

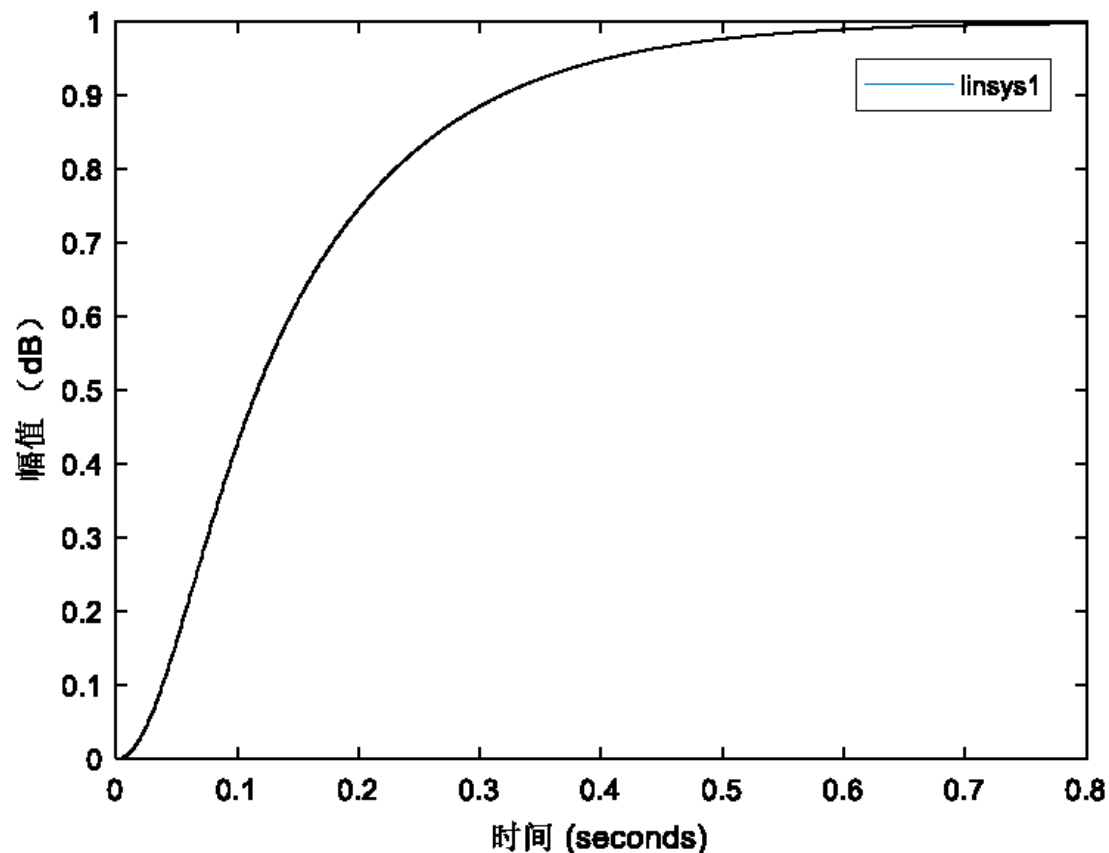


图. pitch角阶跃响应曲线



基础实验

实验步骤

(4) 步骤四：硬件在环仿真

1) 打开Simulink仿真模型

打开“e5\5.1\HIL\AttitudeControl_HIL.slx”

文件，如右图所示。值得注意的是，

“Control System”模块和软件在环仿真

相同。

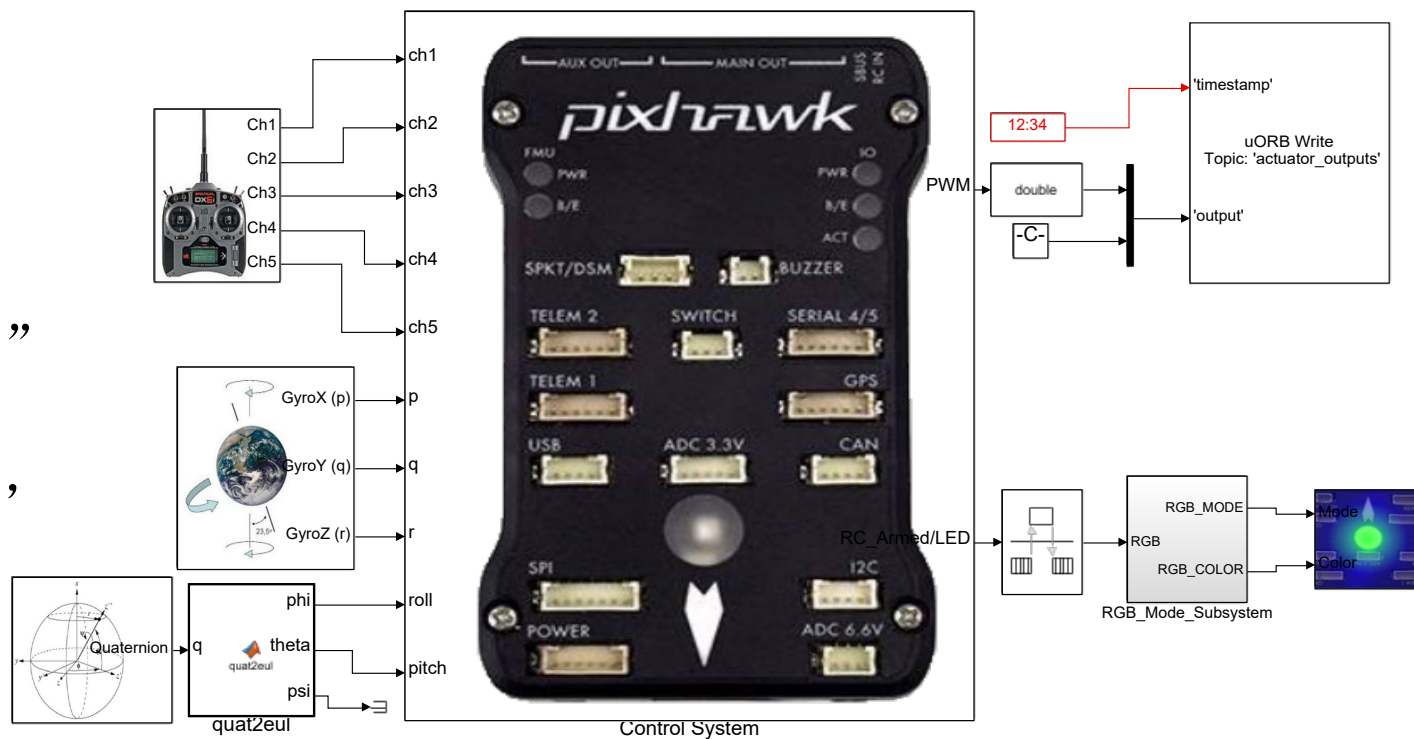


图. AttitudeControl_HIL.slx截图



基础实验

□ 实验步骤

2) 硬件连接

需要注意的是，在进行硬件在环仿真时**机架类型**应该是“**HIL Quadcopter X**”。

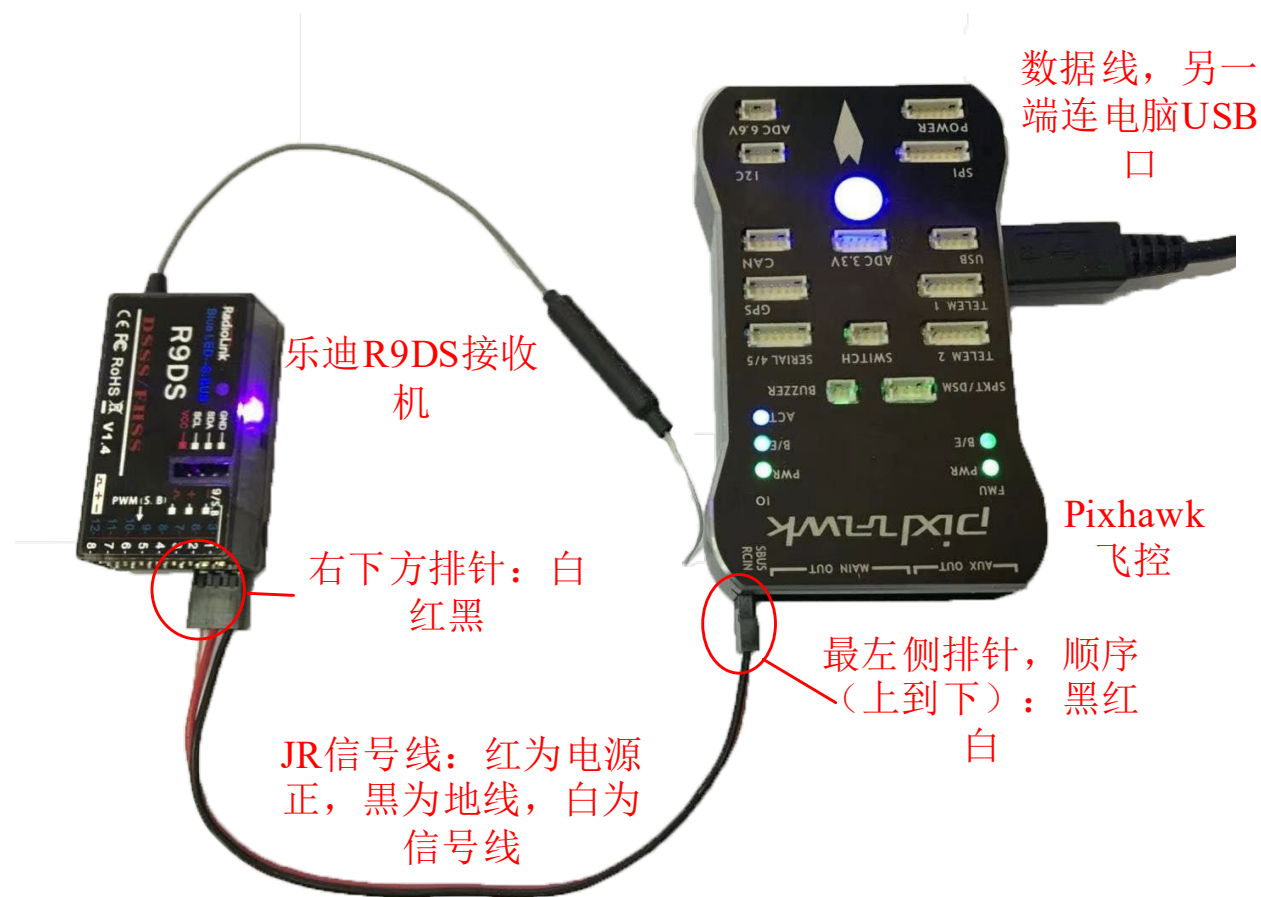


图. 硬件系统连接



基础实验

实验步骤

3) 代码编译及下载

将硬件在环仿真模型编译并下载文件到Pixhawk自动驾驶仪中。这样就可以在Pixhawk自动驾驶仪中运行我们自己设计的姿态控制程序。

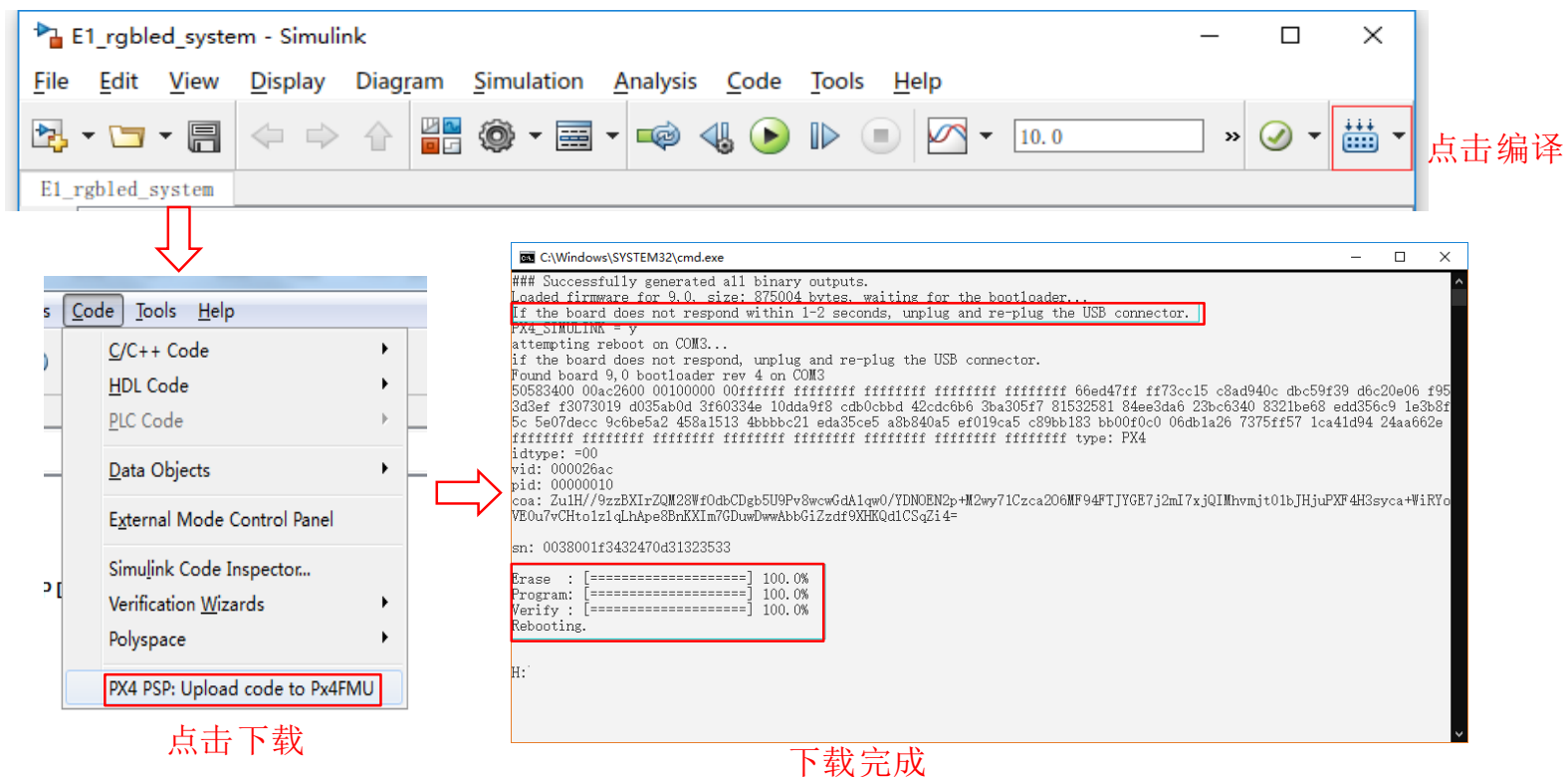


图. 代码编译及下载流程



基础实验

实验步骤

4) 模型仿真器软件配置。

双击 CopterSim 桌面快捷方式即可打开 CopterSim。读者可以选择不同的动力系统模型，步骤如下：单击“模型参数”自定义参数，然后单击“存储并使用参数”存储并使用参数。软件会自动匹配串口号，单击“开始仿真”按钮就可以进入硬件在环仿真模式。此时可以看到如所示的界面左下角收到自驾仪返回的相关消息。

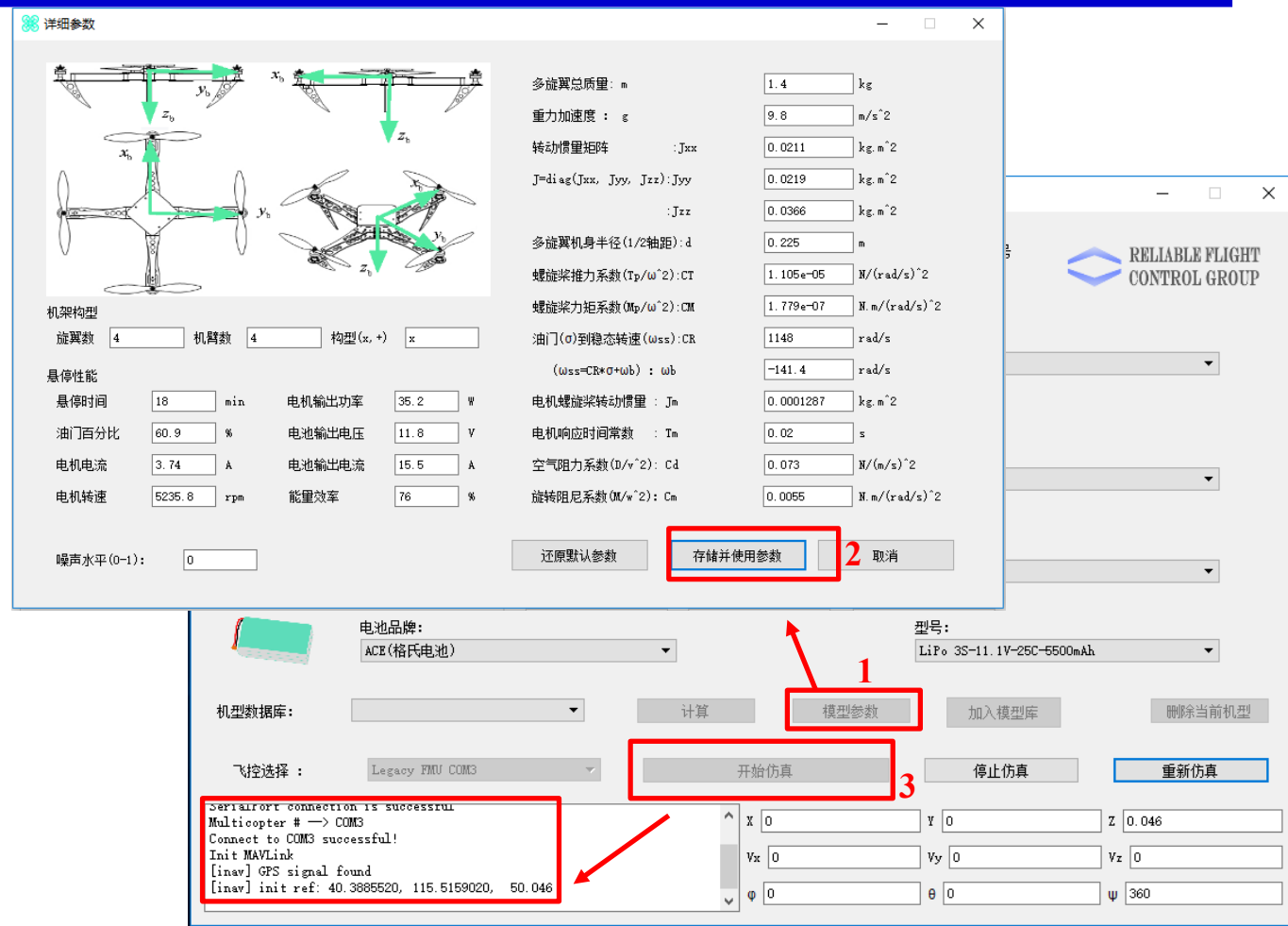


图. 模型仿真器软件配置



基础实验

□ 实验步骤

5) 3D Display三维显示软件配置

双击3DDisplay桌面快捷方式打开

3DDisplay。

6) 仿真效果

用遥控器解锁多旋翼，实现手动控制。

在“3DDisplay”软件界面左侧观察多旋翼状态变化，界面右上角观察实时飞行数据，界面右下角观察多旋翼飞行轨迹。

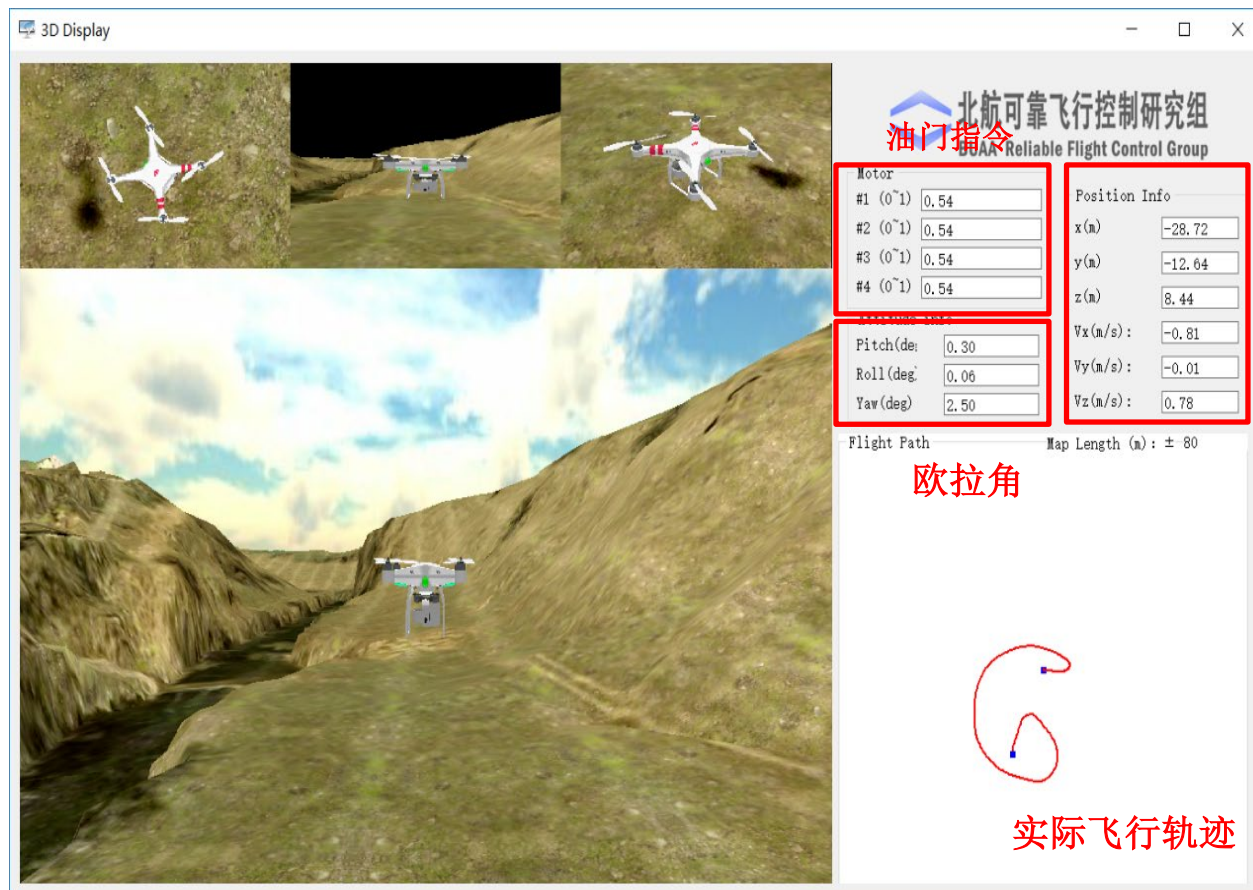


图. 3D Display



分析实验

□ 实验目标

■ 已知

(1) 软件：MATLAB 2017b或以上的版本，基于Simulink的软件在环仿真平台，实验指导包“e5.2”（下载地址：<https://rflsim.com/course>）。

■ 目标

(1) 调节PID控制器相关参数以改善控制性能并记录超调量和调节时间，得到一组恰当参数；

(2) 使用调试后的参数，对系统进行扫频以绘制Bode图，观察系统幅频响应，相频响应曲线，分析其稳定裕度。



分析实验

□ 实验步骤

(1) 步骤一：模型初始设置

在四旋翼对称的前提下，滚转角和俯仰角的PID 参数是一样的。下面以俯仰角的PID 参数调节为例。简化基础实验中的基于Simulink 的软件在环模型。

调节参数的初始状态应是飞行器处于高空悬停状态，油门值设置为0.6085，将初始高度设置为100m，电机的初始转速设置为557.1420rad/s，修改“Init_control.m” 文件中的对应参数。

`ModelInit_PosE=[0,0,-100];`

`ModelInit_VelB=[0,0,0];`

`ModelInit_AngEuler=[0,0,0];`

`ModelInit_RateB=[0,0,0];`

`ModelInit_RPM=557.1420;`



图. AttitudeControl_tune. slx截图



分析实验

□ 实验步骤

(2) 步骤二：角速度环参数调节

在“AttitudeControl_tune.slx”文件中将角速度期望部分换成阶跃输入，并将“q”信号线和阶跃信号设置为“Enable Data Logging”，如图所示。

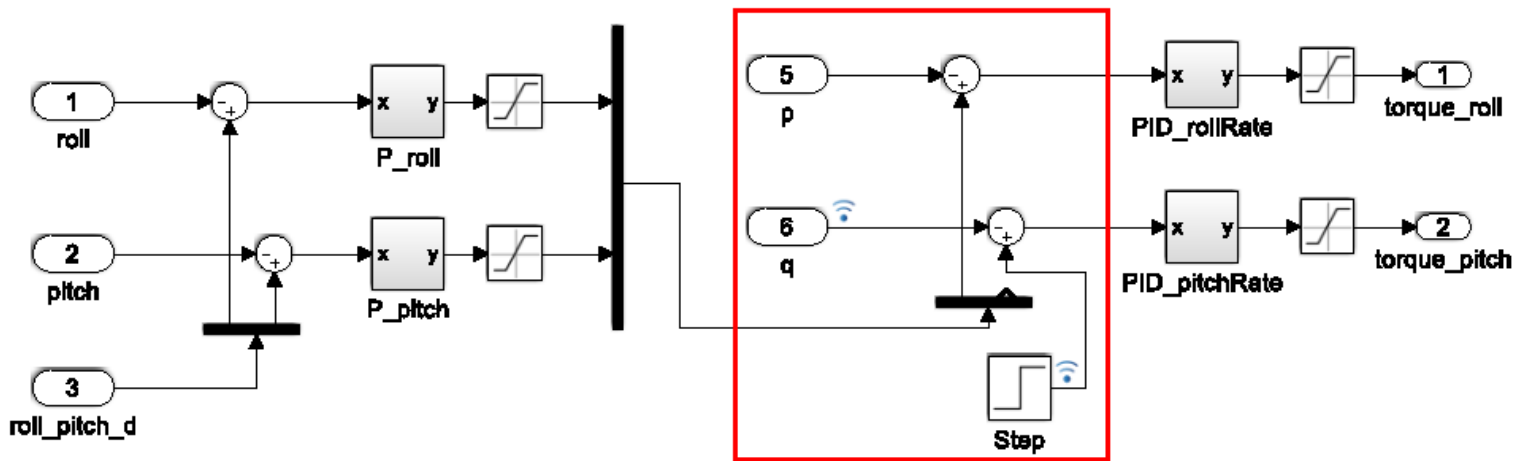


图. 角速度环调试模型



分析实验

□ 实验步骤

(2) 步骤二：角速度环参数调节

在“Init_control.m”文件中修改PID参数的值。

先设定比例项参数，积分和微分参数设为0，运行

“Init_control.m”文件。点击Simulink的开始仿真按钮，在“Simulation Data Inspector”中查看输入输出波形。

由小到大逐渐增大比例项系数，得到阶跃响应曲线如图。

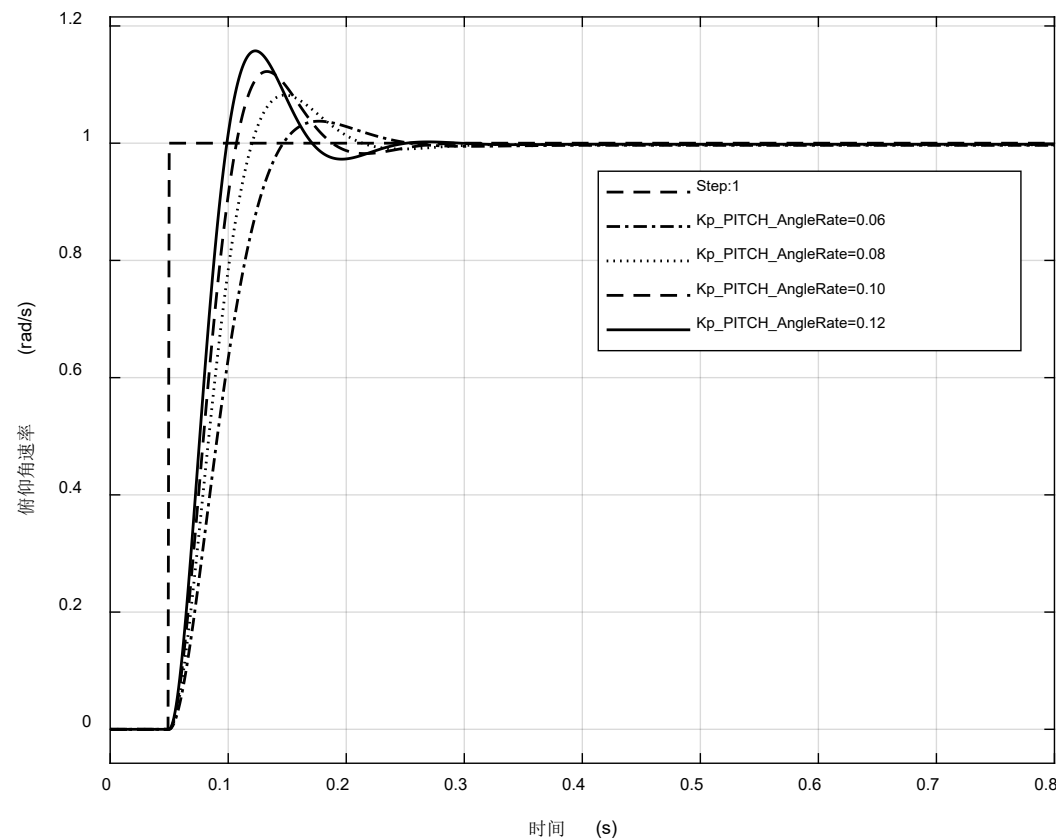


图. 阶跃响应调试



分析实验

□ 实验步骤

(2) 步骤二：角速度环参数调节

加入积分项和微分项系数，并且微调比例项系数，最终得到的响应曲线如右图。

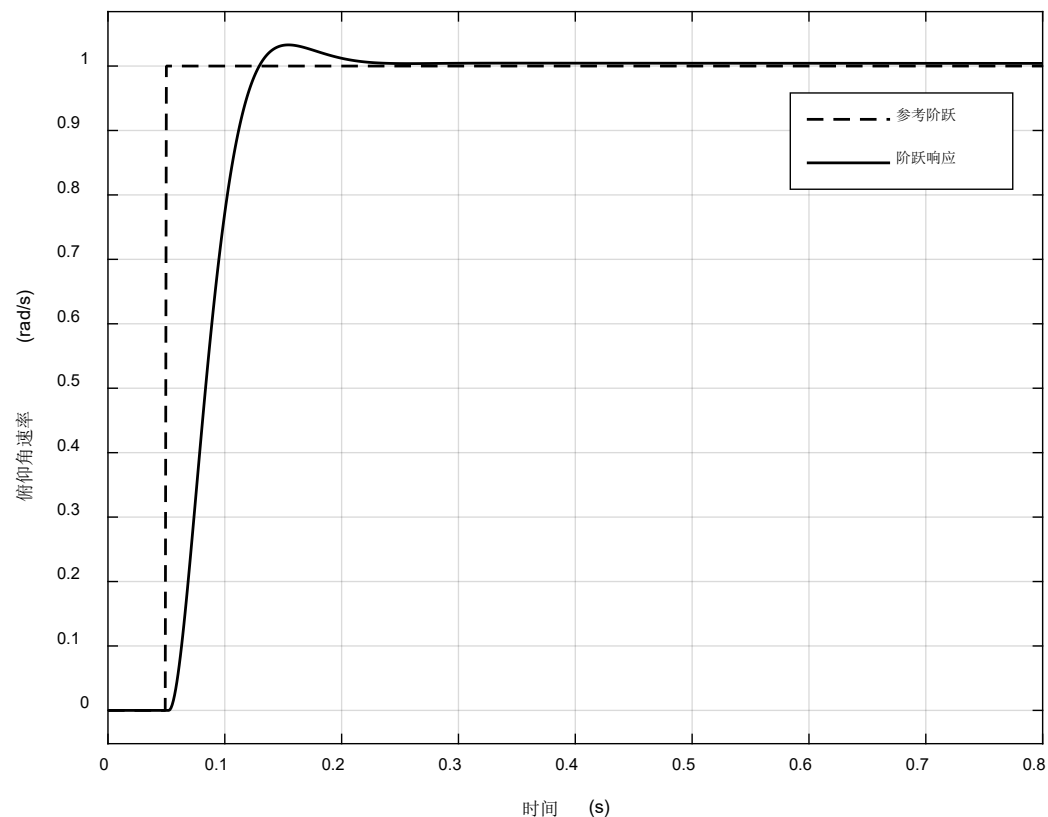


图. 阶跃响应



分析实验

□ 实验步骤

(3) 步骤三：角度环参数调节

采用上一步所得到的角速度环参数，调节角度环比例项系数。将期望俯仰角换成阶跃输入，并将阶跃输入和期望角实际输出设置为“Enable Data Logging”，如右图所示。

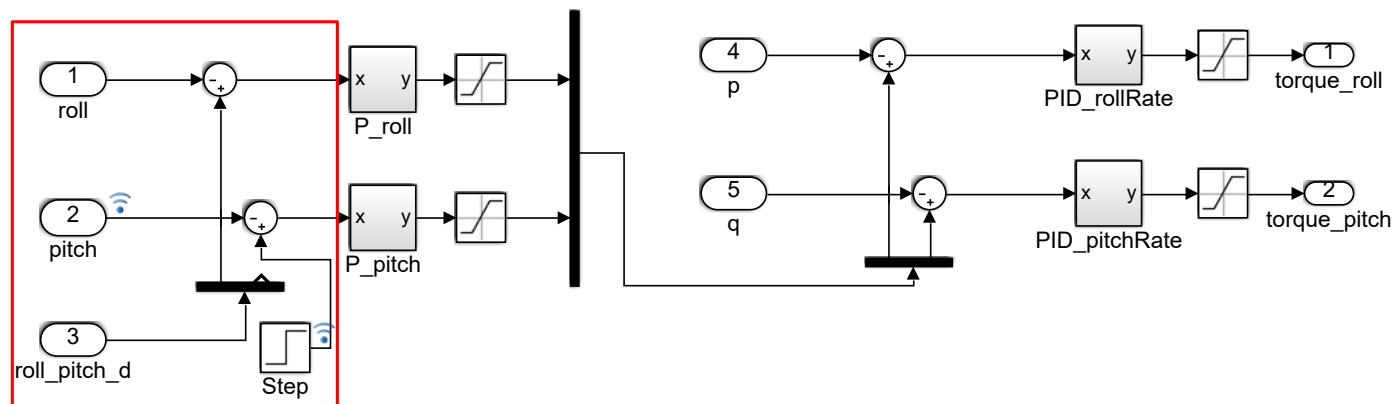


图. 角度环调试模型



分析实验

□ 实验步骤

(3) 步骤三：角度环参数调节

由小增大外环比例项参数，在“Simulation Data Inspector”中观察阶跃响应。得到如下波形如图，再进一步微调比例项参数得到最终的响应曲线。

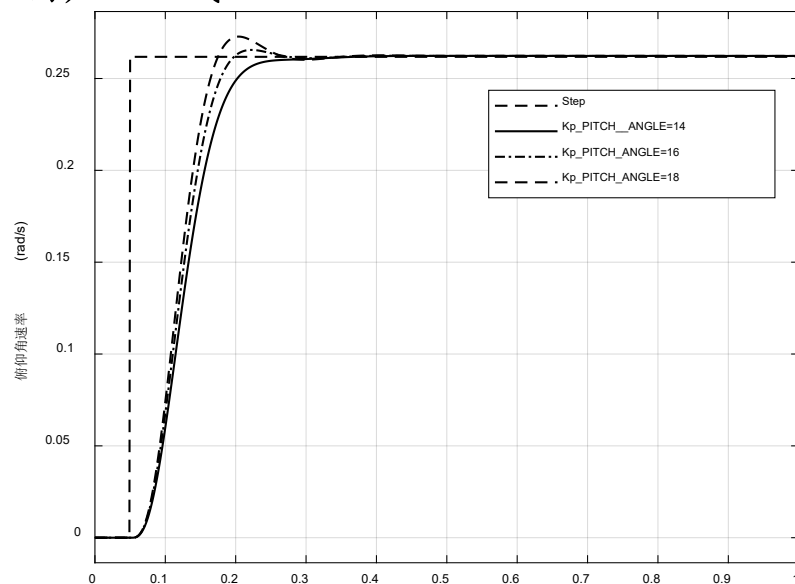


图. pitch角阶跃响应调试 时间 (s)

$Kp_RP_ANGLE = 14;$
 $Kp_RP_AngleRate = 0.10;$
 $Ki_RP_AngleRate = 0.02;$
 $Kd_RP_AngleRate = 0.001;$



分析实验

□ 实验步骤

(4) 步骤四：扫频得到Bode图

设定信号输入输出点。将期望俯仰角输入线设为“Open-loop Input”，俯仰角的实际输出设置为“Open-loop Output”。得到Bode图。

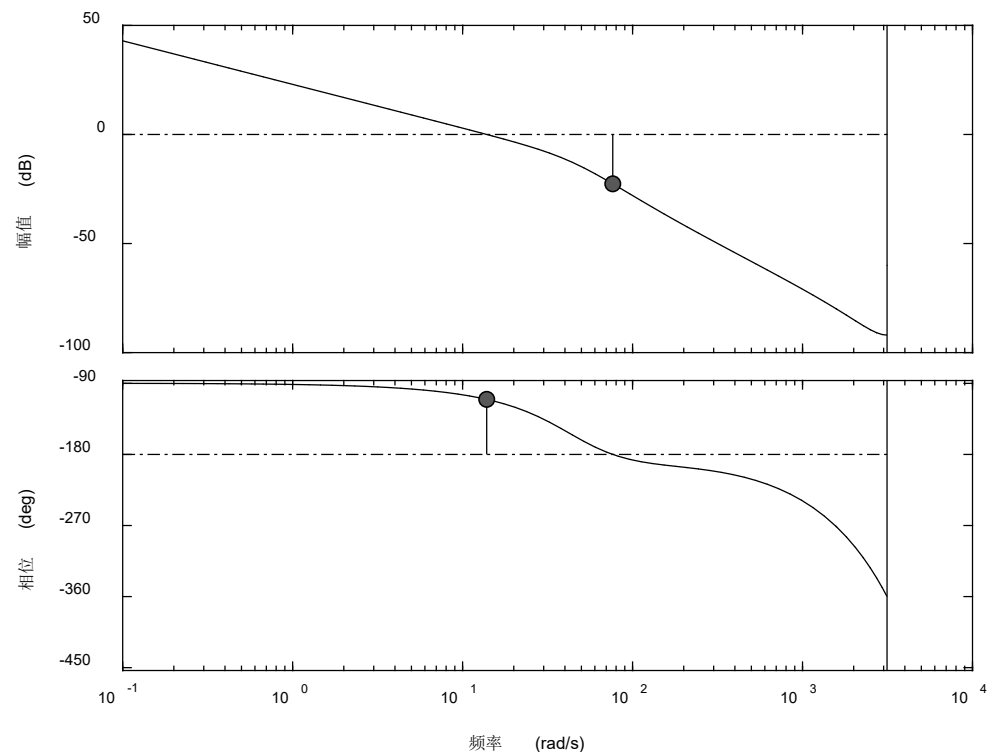


图. pitch角控制系统开环Bode图



设计实验

□ 实验目标

■ 已知:

- (1) 硬件: Pixhawk 自驾仪系统, 多旋翼硬件系统。
- (2) 软件: MATLAB 2017b 或以上版本, 基于 Simulink 的控制器设计与仿真平台, 硬件在环仿真平台, 实验指导包 “e5.3” 和 “e5.4” (下载地址: <https://rflysim.com/course>)。

■ 目标:

- (1) 建立姿态控制通道的传递函数模型, 设计校正控制器, 使得姿态角速度环稳态误差 $e_{rSS} \leq 0.01$, 相位裕度 $> 65^\circ$, 截至频率 $> 10 \text{ rad/s}$ 。姿态角度环截至频率 $> 5 \text{ rad/s}$, 相位裕度 $> 60^\circ$;
- (2) 使用自己设计的控制器进行软件在环和硬件在环仿真实验;
- (3) 使用自己设计的控制器进行实飞实验。



设计实验

□ 实验设计

(1) 步骤一：对整体结构进行简化

首先只对一个通道进行分析，这里选择俯仰角通道，简化后的模型如下图所示。

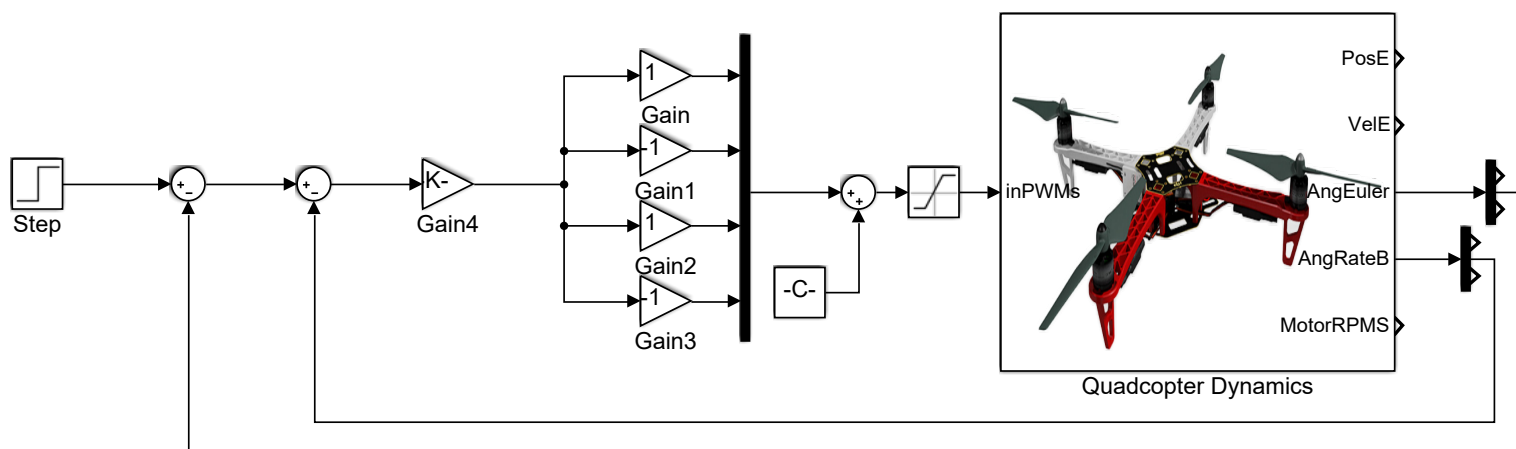


图. 俯仰角控制系统简化模型



设计实验

□ 实验设计

(2) 步骤二：角速度环分析

输入为期望角速度，输出为实际角速度。设置输入输出点如图所示。

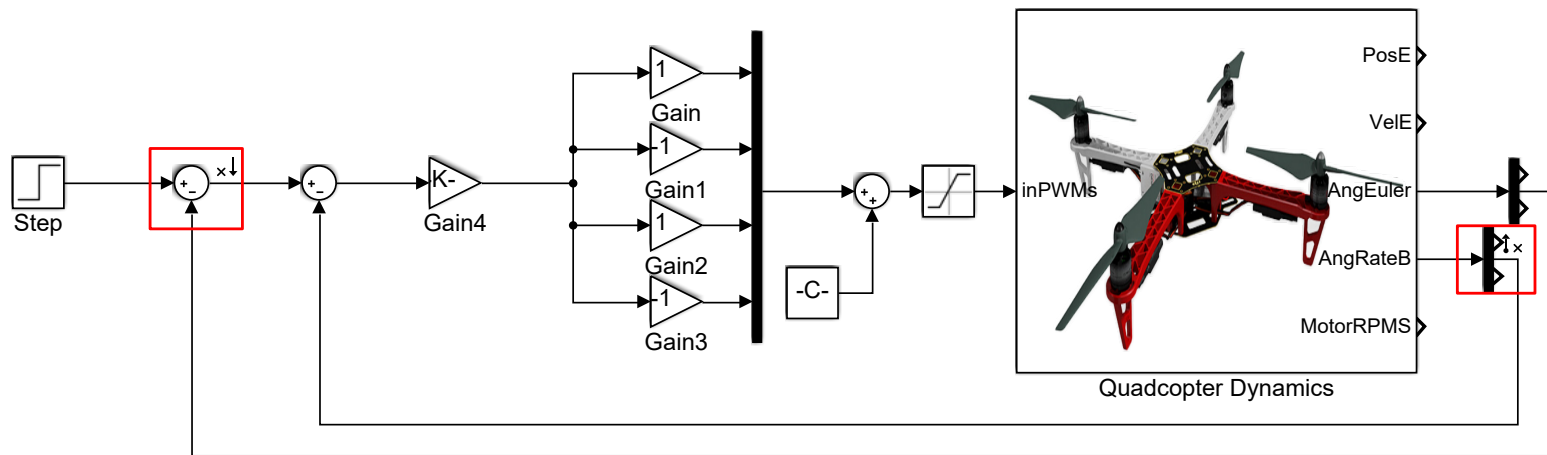


图. 设置输入输出点



设计实验

□ 实验设计

(3) 步骤三：得到传递函数模型

生成 Bode 图后，在左侧 Linear Analysis Workspace 中会出现 linsys1 变量，按图示操作即可得到传递函数模型。

$$\frac{20538}{(s + 9.132e - 07)(s + 50)} \quad \longrightarrow \quad \frac{410.76}{s(0.02s + 1)}$$

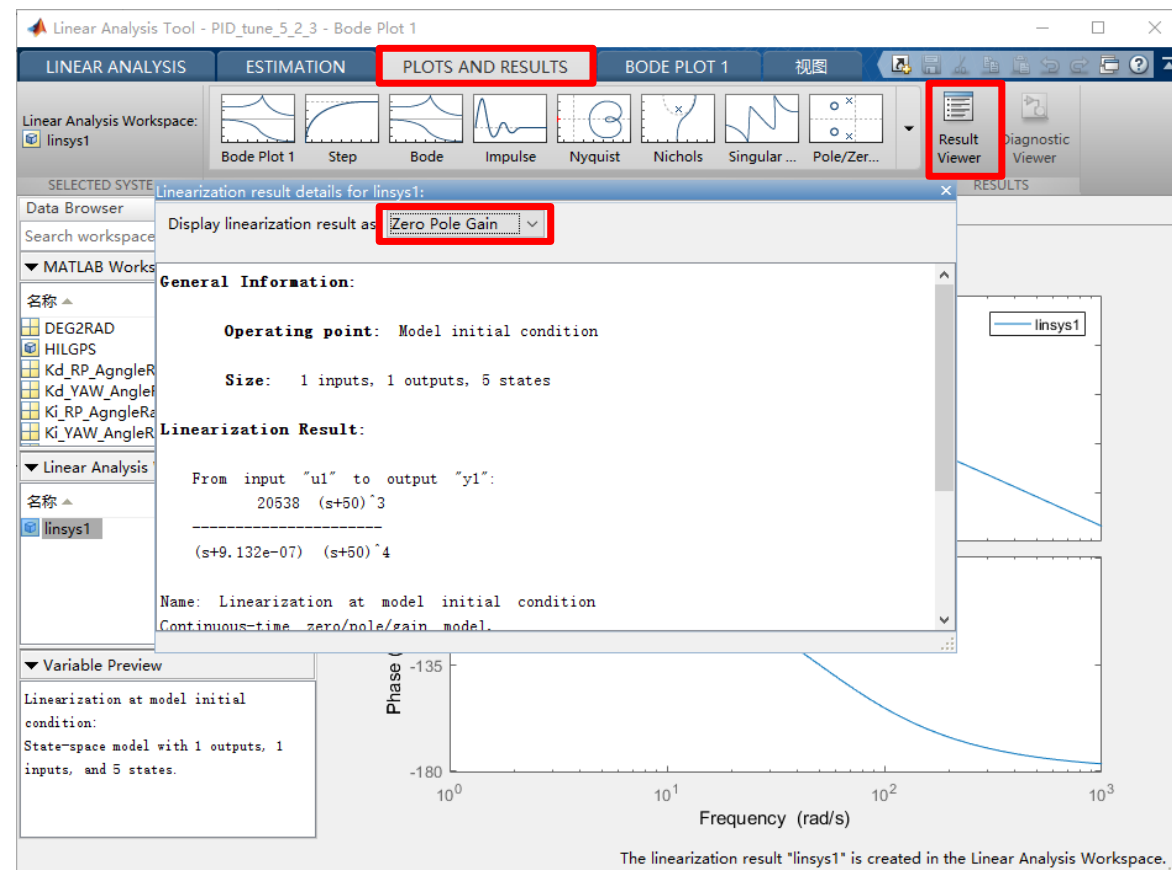


图. 得到传递函数



设计实验

□ 实验设计

(4) 步骤四：调整开环增益

首先根据稳态误差调整开环增益。未加校正时，系统在 $r(t)=t$ 作用下的稳态误差，可由终值定理求出。

$$e_{r_{ss}} = \frac{1}{K}$$

因此要 $e_{r_{ss}} \leq 0.01$ ，就有 $K \geq 100$ 。现在 $K = 410.76$ 已经大于100，故无需调整。



设计实验

(5) 步骤五：设计角速度环校正器

做出未校正系统的Bode图，如图

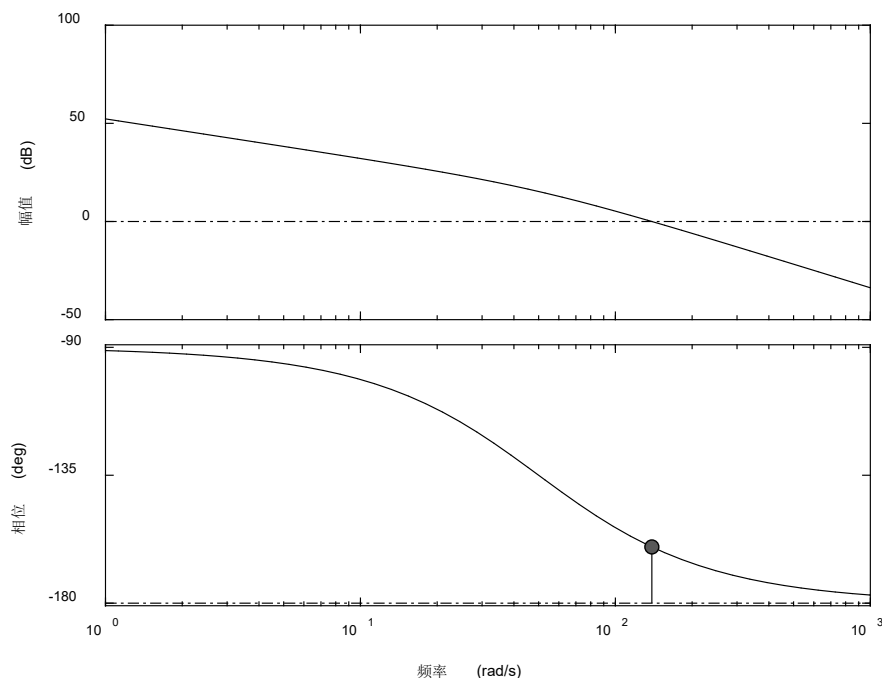


图. 角速度环未做校正系统开环Bode图

稳定裕度 $\gamma = 19.8^\circ$ 不符合要求。考虑使用相位滞后校正，这样既可以使相位裕度达到要求，又能提高系统对高频干扰的抑制能力。考虑设置截止频率在 $\omega'_c = 12.9 \text{ rad/s}$ 处，由Bode图可知该频率下相位裕度为76度，符合要求，并由Bode图幅频特性曲线得知该频率下幅度响应为29.8dB。

在 ω'_c 处，要求加入校正后幅频特性为0dB，根据滞后校正环节幅频特性得到

$$20 \lg b + 20 \lg |G(j\omega'_c)| = 0$$

得 $b = 0.0324$ 。

为了使滞后校正部分的相位滞后特性对 ω'_c 处影响不大，所以校正环节的转折频率 $(bT)^{-1}$ 应设在远离 ω'_c 十倍频程的地方，即

$$(bT)^{-1} = 0.1\omega'_c$$

将 b, ω'_c 代入，可得 $T = 23.9558$ 。此时校正系统的传递函数为：

$$G_c(s) = \frac{1 + bTs}{1 + Ts} = \frac{1 + 0.775194s}{1 + 23.955779s}$$

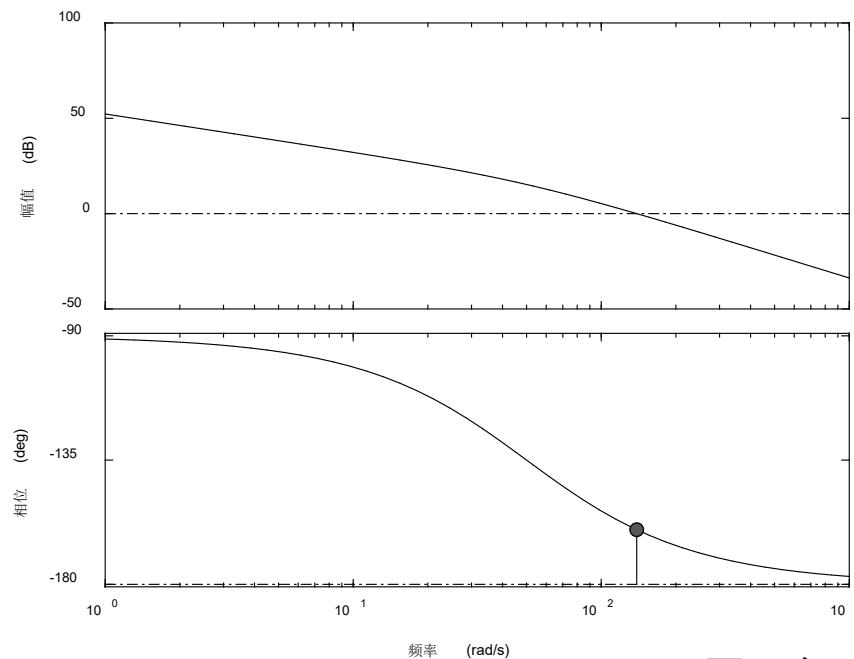


设计实验

□ 实验设计

(6) 步骤六：将校正器加入到角速度控制环

加入校正环节后的Bode图如图所示，相位裕度为 70° 。



$$G_c(s) = \frac{1+0.775194s}{1+23.955779s}$$

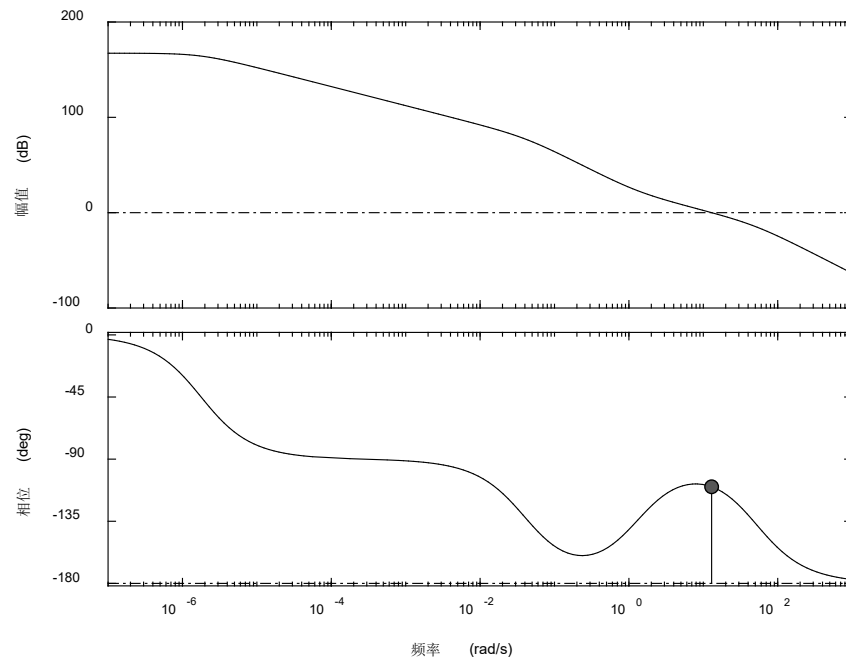


图. 角速度环校正前后的开环Bode图



设计实验

□ 实验设计

(7) 步骤七：设计角度环校正器。

角度环也使用反馈进行闭环控制，然后设计校正器对其进行校正。与角速度环类似，先得到角度环的开环Bode图。

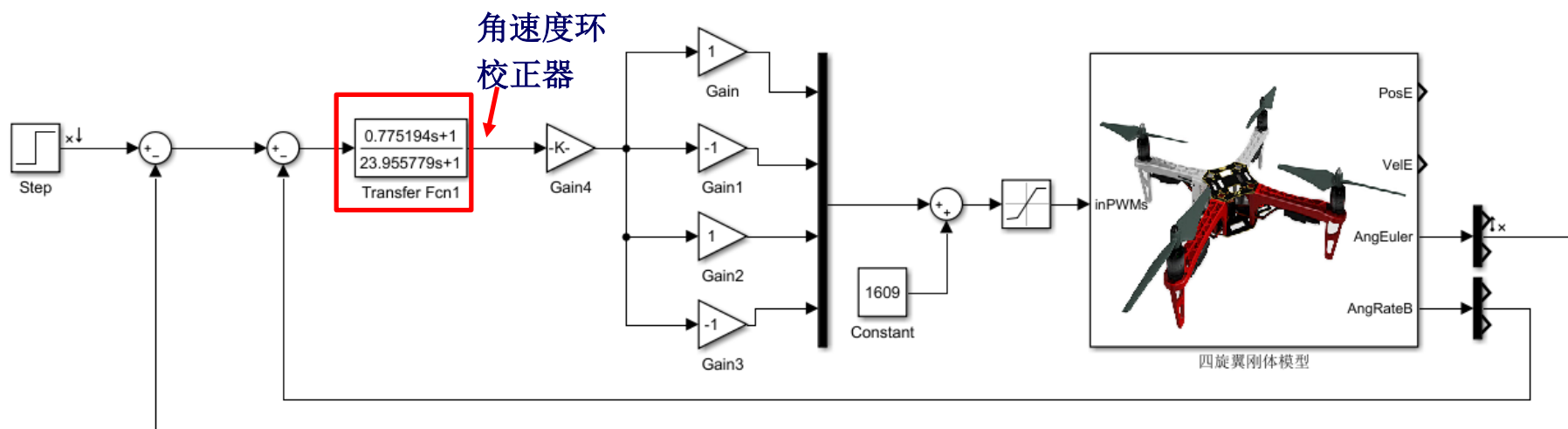


图. 角度环未校正模型



设计实验

□ 实验设计

(6) 步骤六：设计角度环校正器

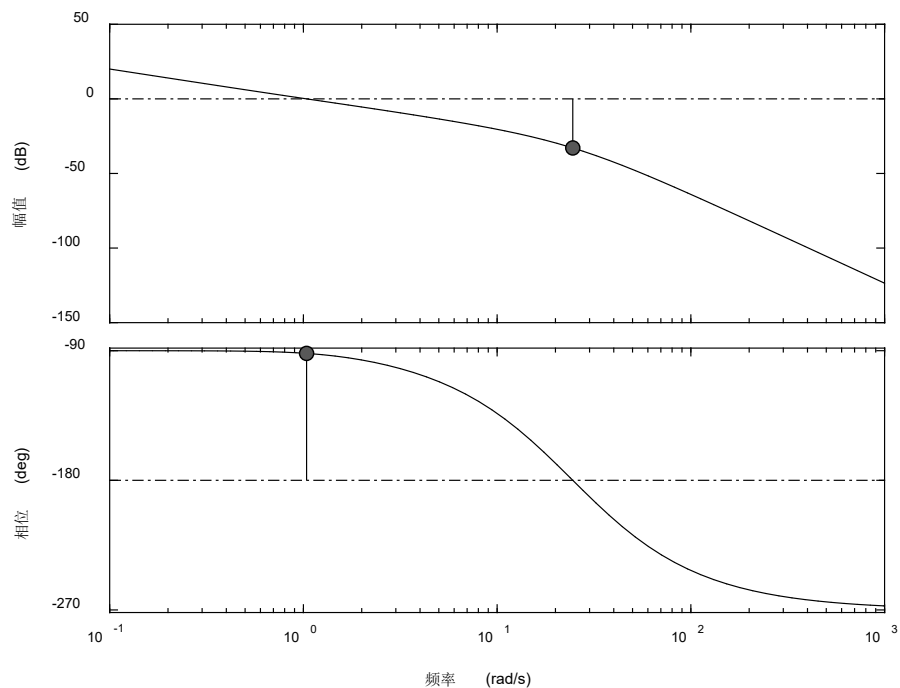


图. 角度环校正前Bode图

得到系统Bode图如图。可以看到截至频率为 1.04rad/s 不符合要求，相位裕度为 88.1° 。根据姿态角度环截至频率 $>5\text{rad/s}$,相位裕度 $>60^\circ$ 的设计要求，考虑选择增大开环增益 K 。从图中可以看出， $\omega=5.3\text{rad/s}$

时，幅频特性曲线的高度为 -14dB ，要求加入校正环节后幅频特性曲线高度为 0dB ,则有：

$$20\lg(K_2) = 14$$

可得 $K_2 = 5.0119$ 。

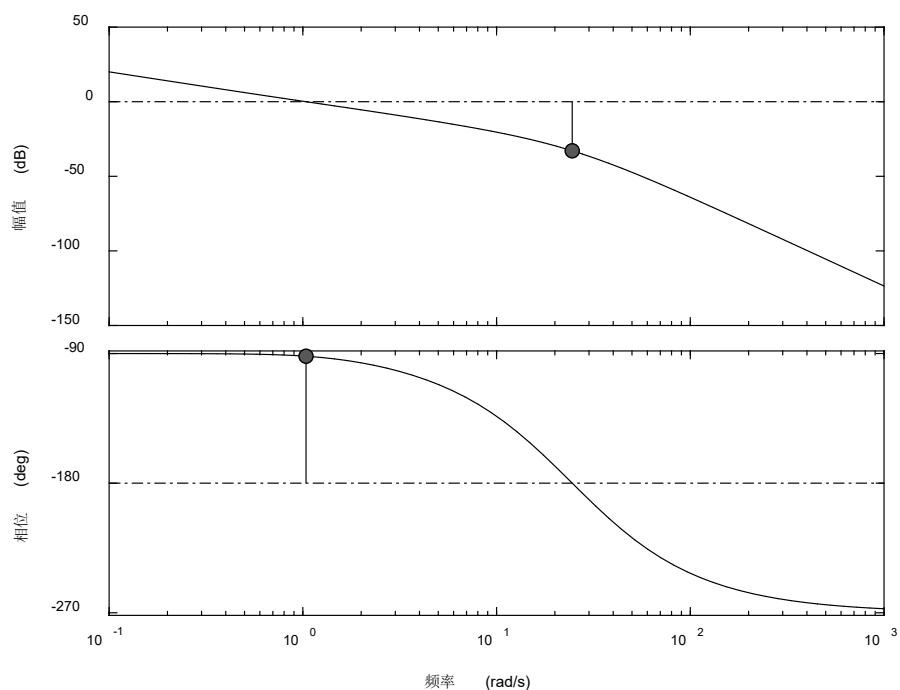


设计实验

□ 实验设计

(6) 步骤六：设计角度环校正器

加入校正环节后的Bode图如下，截止频率为 5.3rad/s ，幅值裕度为 67.1°



$$K_2 = 5.0119$$

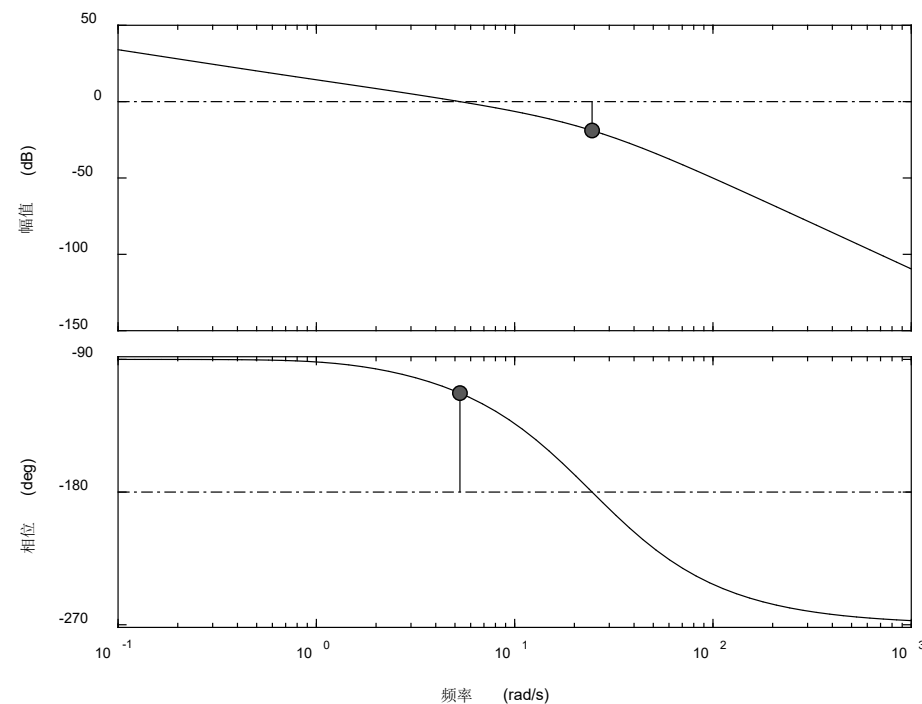
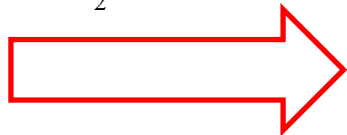


图. 角度环校正前后Bode图



设计实验

□ 仿真实验步骤

(1) 步骤一：将连续时间校正环节离散化

前述设计的校正环节是时域连续的环节，应将其变为时域离散的环节以便在自驾仪上运行。
使用c2d函数将s域的传递函数变为z域：

$$H = \text{tf}([\text{num}], [\text{den}])$$

$$H_d = \text{c2d}(H, T_s, \text{'foh'})$$

其中“num”为传递函数分子系数向量，“den”为传递函数分母系数向量，“Ts”为仿真步长，本例中为0.004s。

替换后的传递函数为：

$$G_c(s) = \frac{1 + 0.775194s}{1 + 23.955779s} \rightarrow G_c(z) = \frac{0.03236z - 0.03219}{z - 0.9998}$$



设计实验

□ 仿真实验步骤

(2) 步骤二：替换控制模型。

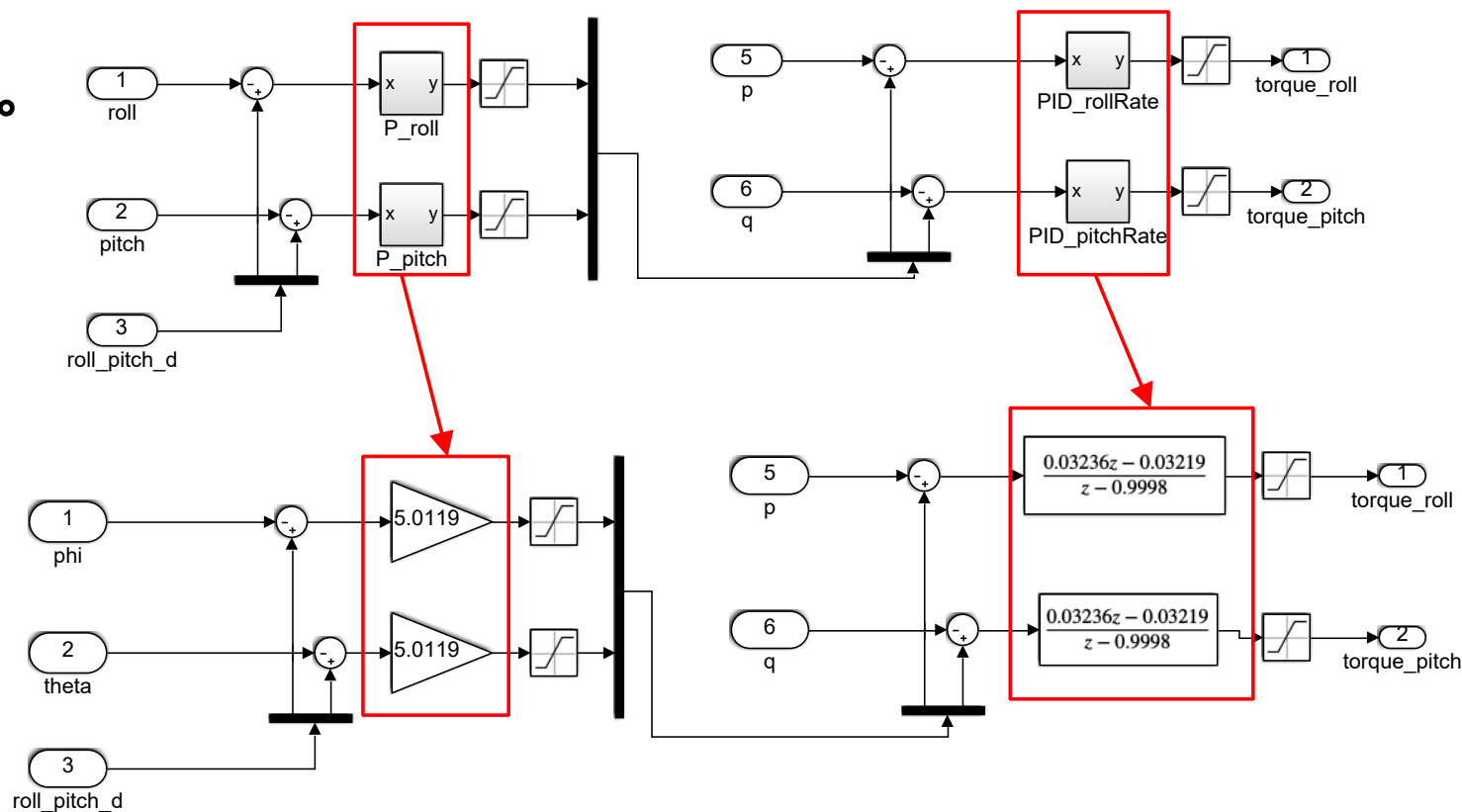


图. 替换原硬件在环仿真中的PID校正器



设计实验

□ 仿真实验步骤

(3) 步骤三：进行硬件在环仿真

观察发现，随着被拨动摇杆的放开，多旋翼能很快的返回到之前平稳的状态，并无明显振荡。故可认为校正环节设计可以满足要求。

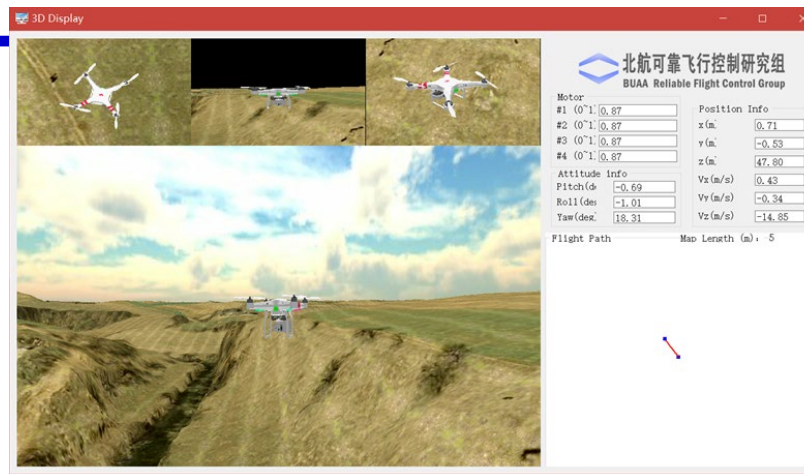


图. 3D显示界面—无人机解锁并起飞



图. 拨动摇杆使四旋翼无人机滚转角变化

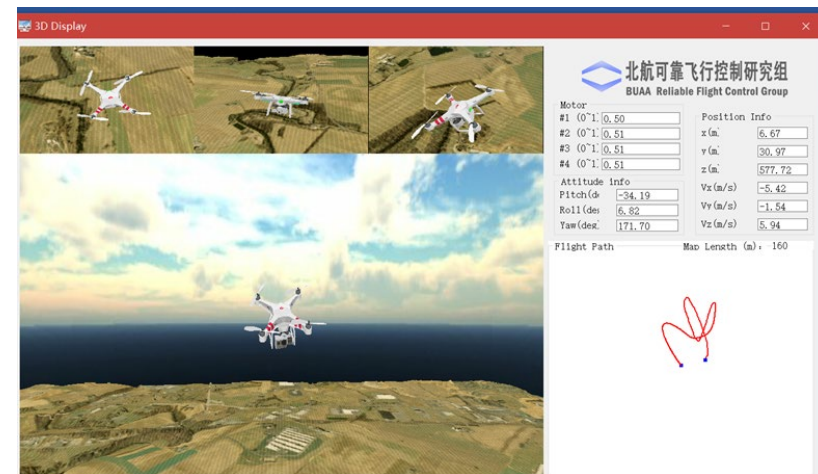


图. 拨动摇杆使四旋翼无人机俯仰角变化



设计实验

□ 实飞实验步骤

(1) 步骤一：四旋翼准备

实际飞行试验所采用的多旋翼为F450 四旋翼，如图所示，在实际飞行时需要在QGC 中将Pixhawk 的机架类型从“HIL Quadcopter X”修改为“DJI Flame Wheel F450”，并完成传感器校准。



图. F450 机架示意图



设计实验

□ 实飞实验步骤

(2) 步骤二： Simulink 实飞模型

相比设计实验中的硬件在环仿真模型，这里只是将其PWM 输出部分替换了。为了记录飞行中的数据，我们在模型中加入了一种新的数据记录的方式，在打开模型时会提示“invalid .msg specified”的错误对话框。

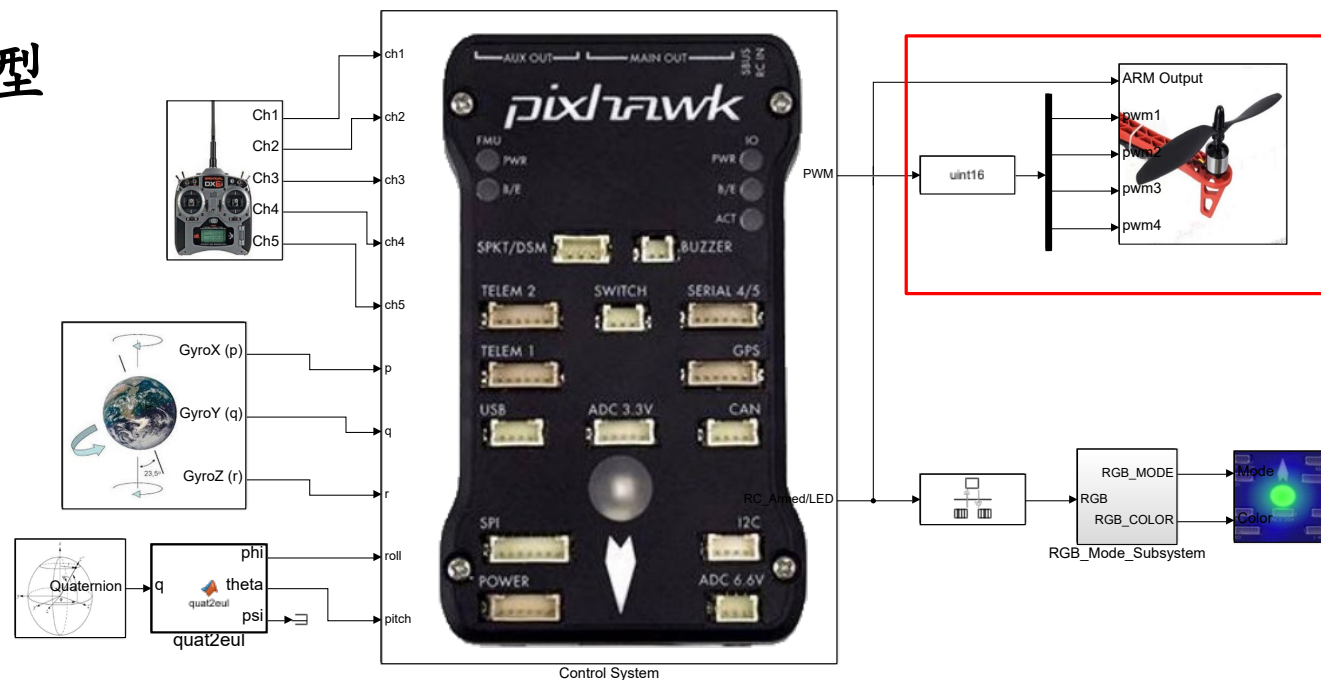


图. 实飞模型, Simulink 模型 “AttitudeControl_FLY.slx”



设计实验

□ 实飞实验步骤

(2) Simulink 实飞模型—添加自定义logger数据

1) 自定义message 文件。建立costom_attctrl_e5.msg 文件如表所示，例程可以参考“e5.4\PSPfile”内文件（如果使用的是PX4 1.9及以上固件，还需要在第一行增加“uint64 timestamp”变量，）。然后将创建的costom_attctrl_e5.msg 文件复制到“Firmware\msg”文件夹（默认在C:\PX4PSP目录内下）。

1	#attitude data
2	float32[2] euler_rp
3	float32 yawrate
4	#desired attitude data
5	float32[2] euler_rp_d
6	float32 yawrate_d
7	

2) 打开“Firmware\msg\CmakeLists.txt”文件，在set(msg_files 函数的下一行添加costom_attctrl_e5.msg。

3) 打开“Firmware\src\modules\logger\logger.cpp”文件（对于PX4 1.11及以上版本请打开“Firmware/src/modules/logger/logged_topics.cpp文件），在add_default_topics()最后加入

```
add_topic(“costom_attctrl_e5”,4);
```

其中“costom_attctrl_e5”为消息名称，4为写入周期，代表4ms（即250Hz）。

完成后重新打开“AttitudeControl_HIL.slx”文件就不会提示错误了。



设计实验

□ 实飞实验步骤

(3) 步骤三：下载代码

与硬件在环仿真过程中编译下载代码过程完全一致。

(4) 步骤四：无桨测试

1) 油门通道

● 拆掉螺旋桨，然后连接电源

- 打开遥控器，向上拨遥控器CH5 通道实现解锁
- 向上推CH3 油门通道，观察电机是否正常转动。

2) 滚转通道

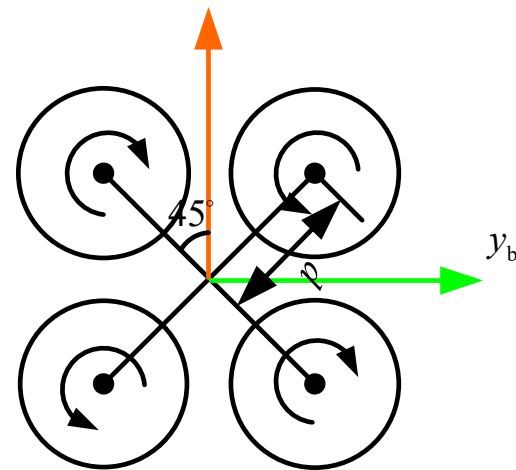
- 向左拨动CH1 滚转通道
- 观察此时1、4 号电机的转速是否变快，2、3 号电机的转速是否变慢。

3) 俯仰通道

- 向前拨动CH2 俯仰通道
- 观察2、4 号电机转速是否变快，1、3 号电机转速是否变慢。

4) 偏航通道

- 向右拨CH4 偏航通道
- 观察1、2 号电机转速是否变快，3、4 号电机转速是否变慢。





设计实验

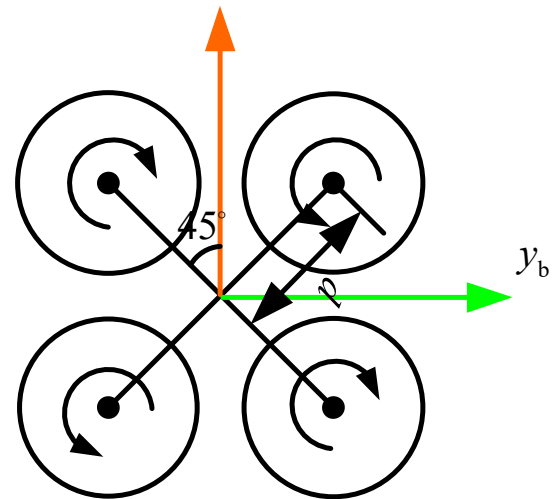
□ 实飞实验步骤

(4) 无桨测试

5) 自动控制检验

- 保持油门居中，手拿飞机向左侧倾斜（绕 ob_xb 轴旋转），观察#2、3 号电机转速是否变快，#1、4 号电机转速是否变慢。
- 飞机向前倾斜（绕 $obyb$ 轴旋转），观察#1、3 号电机转速是否变快，#2、4 号电机转速是否变慢。
- 飞机向右偏航（绕 ob_zb 轴旋转），观察#1、2 号电机转速是否变慢，#3、4 号电机转速是否变快。

至此说明自驾仪对姿态的控制响应正常。





设计实验

□ 实飞实验步骤

(5) 步骤五：室内台架飞行测试

将多旋翼安装在台架上，并安装上螺旋桨，如图所示。解锁多旋翼，测试飞行。



图. 台架上的多旋翼



设计实验

□ 实飞实验步骤

(6) 步骤六：室外实飞测试

为确保安全，可在飞机上系上安全绳，并将安全绳的另一端固定在重物上，如图所示。飞行时人在安全半径以外，在姿态模式下，高度可能比较难控，注意不要急推油门，让油门在中位附近，缓慢推油门。



图. 室外实飞



设计实验

□ 实飞实验步骤

(7) 步骤七：飞行数据分析

- 1) 取出Pixhawk 中的SD 卡，使用读卡器读出 logger 文件“log001.ulg”，并将该文件复制到“e5\e5.4\”目录下；
- 2) 使用“ulog2csv”函数即可将“ulg”文件转换成“csv”文件进行分析。如运行

```
ulog2csv('log001.ulg', 'log001')
```

就可将“log001.ulg”文件内容提取到“log001”文件夹中。自定义消息数据在“log001_custom_attctrl_e5_0.csv”文件中。

台架飞行数据和室外飞行数据如图所示。

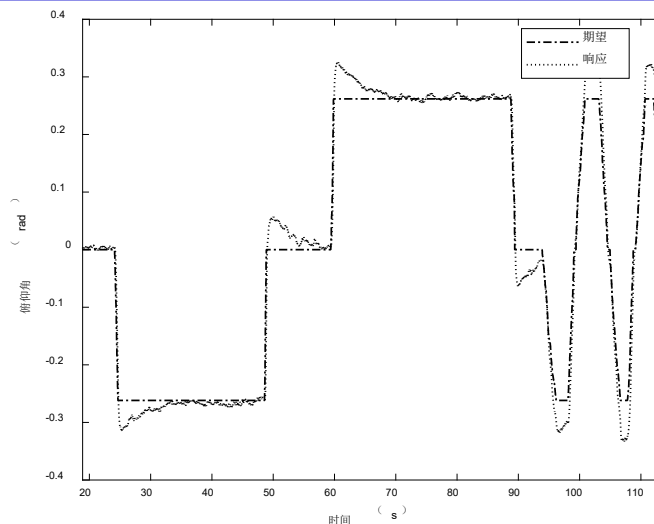


图. 台架飞行俯仰响应

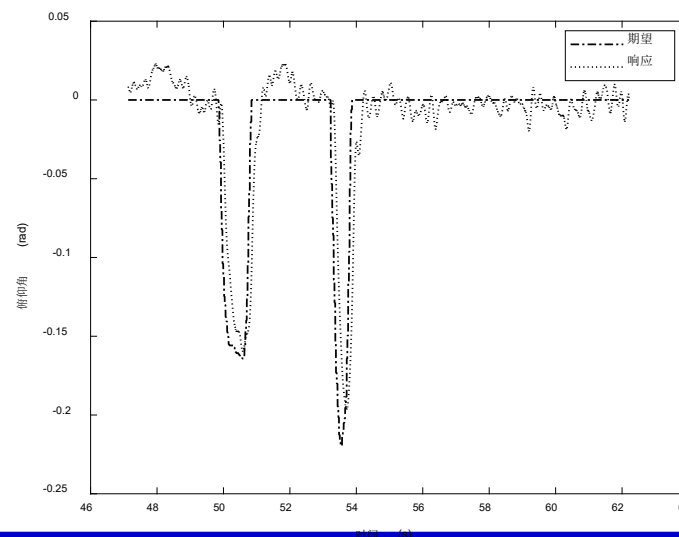


图. 室外实飞俯仰响应



小结

- (1) 以多旋翼的姿态模型为依据，建立了常见的PID控制方法，并在MATLAB的Simulink中完成姿态控制器的设计，并在FlightGear中显示仿真效果。
- (2) 使用Simulink中的PSP工具箱生成代码下载到PixHawk飞控中进行硬件在环仿真实验。
- (3) 调节PID控制器的参数，尝试得到最优的参数，并使用MATLAB系统分析工具得到整个开环系统的Bode图，并查看系统的稳定裕度和幅值裕度。
- (4) 使用自动控制原理中的系统校正方法对多旋翼系统进行校正，设计了超前和滞后超前环节分别对角度环和角速度环实施控制，并达到设计指标。完成设计后，连接硬件进行在环仿真验证了设计的效果。

如有疑问，请到<https://rflysim.com>查询更多信息。



多旋翼飞行器设计与控制 实践

第九讲 姿态控制器设计实验拓展 ——基于ADRC的姿态控制器设计

王帅

wsh_buaa@buaa.edu.cn

自动化科学与电气工程学院

北京航空航天大学



北航可靠飞行控制研究组

BUAA Reliable Flight Control Group



大纲

1. ADRC的提出
2. ADRC的结构及参数整定
3. ADRC在多旋翼姿态控制中的应用实例
4. 控制器调试



ADRC的提出

把系统的模型作用当做系统的内扰，那么它连同系统的外扰一起，均可作为对系统的扰动。这个补偿分量并不区分内扰和外扰，直接检测并补偿他们的总和作用——对系统的总扰动。由于这个分量的补偿作用，被控对象实际上被化成积分器串联型而易于构造出理想的控制器，这个补偿分量的补偿作用实质上是一种抗扰作用。因此我们将此控制器称为“自抗扰控制器”（ADRC）。

韩京清. 自抗扰控制器及其应用. 控制与决策, 1998, 13 (1) : 19-23.



ADRC的提出

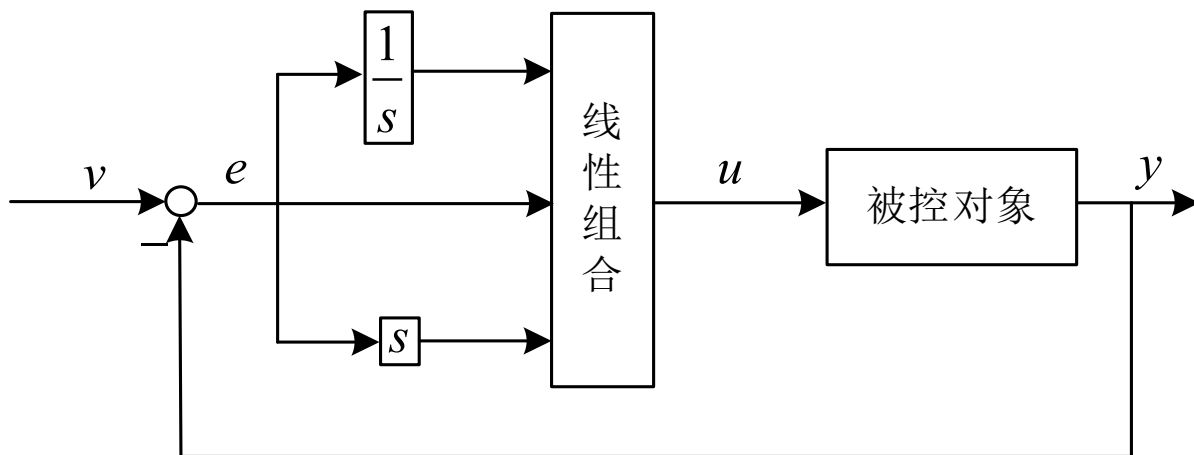


图. PID基本结构

PID控制器的缺陷：

- 直接接取目标与实际行为之间的误差并不是完全合理的。控制目标是在过程中可以“跳变”，但是对象输出 y 的变化都有惯性，不可能跳变，要求让缓变的变量 y 来跟踪能够跳变的变量 v 本身就不合理。
- 产生误差信号的微分信号没有太好的办法。控制界有一个紧箍咒——微分器物理不可实现，只能近似实现。PID 控制器除特殊情形之外，实际上都是 PI 控制器。

韩京清. 自抗扰控制技术[J]. 前沿科学, 2007(01):25-32.



ADRC的提出

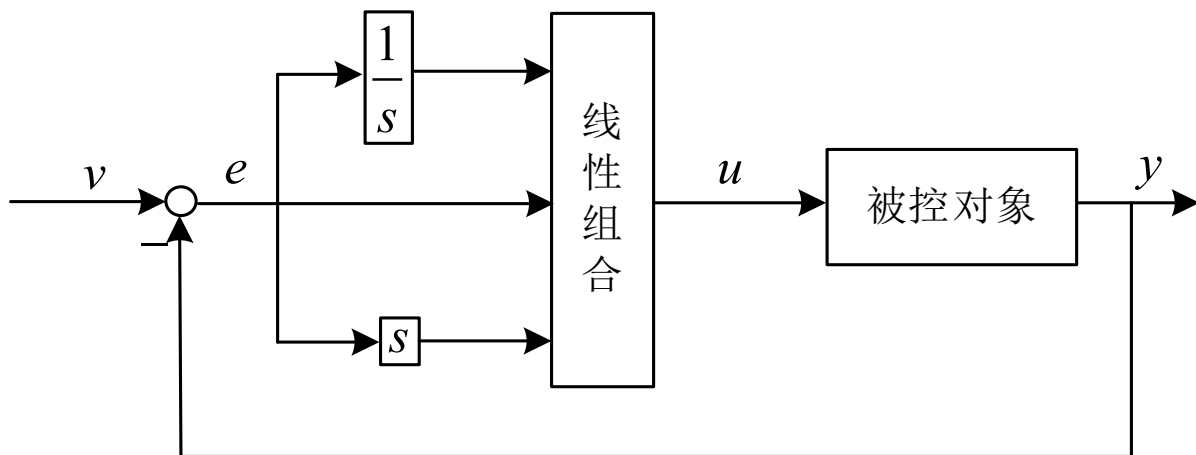


图. PID基本结构

PID控制器的缺陷:

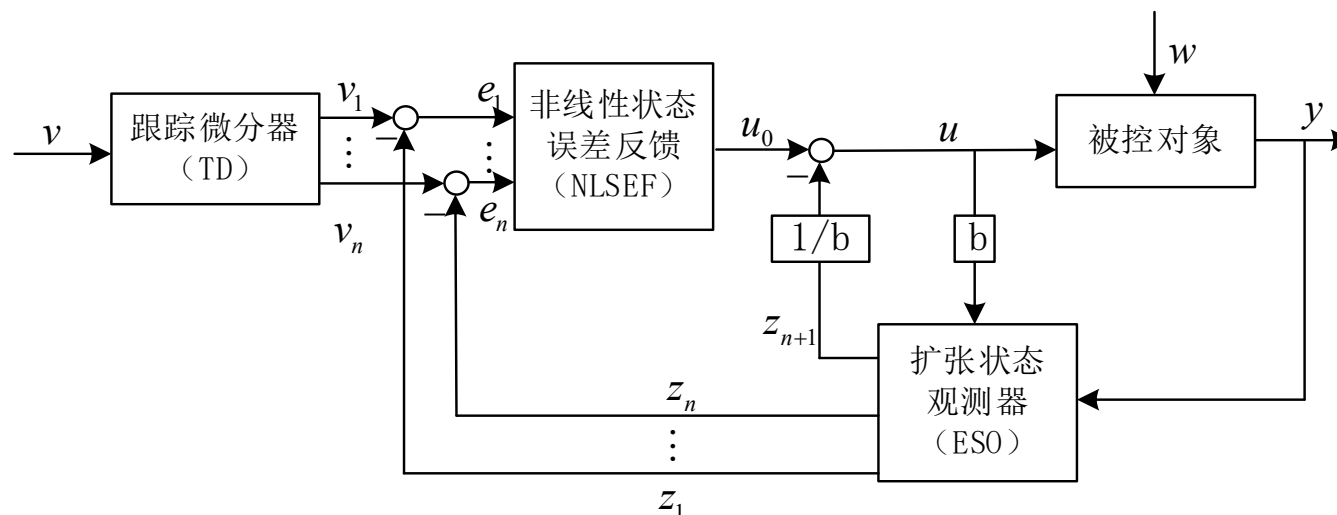
- 线性组合不一定是最好的组合方式。PID控制器给出的控制量是误差的现在、误差的过去、误差的将来三者的加权和,是这三者的线性组合。大量工程实践表明,三者的线性组合不一定是最好的组合方式。
- 误差信号的积分反馈的引入有很多副作用。大量工程实践表明,误差积分反馈的引入,使闭环变得迟钝,容易产生振荡,积分饱和引起的控制量饱和等。

韩京清. 自抗扰控制技术[J]. 前沿科学, 2007(01):25-32.



ADRC的结构及参数整定

PID控制器的缺陷	解决方法
直接接取目标与实际行为之间的误差并不是完全合理的	安排过渡过程和微分信号提取
产生误差信号的微分信号没有太好的办法	
线性组合不一定是最好的组合方式	非线性状态误差反馈
误差信号的积分反馈的引入有很多副作用	扩张状态观测器



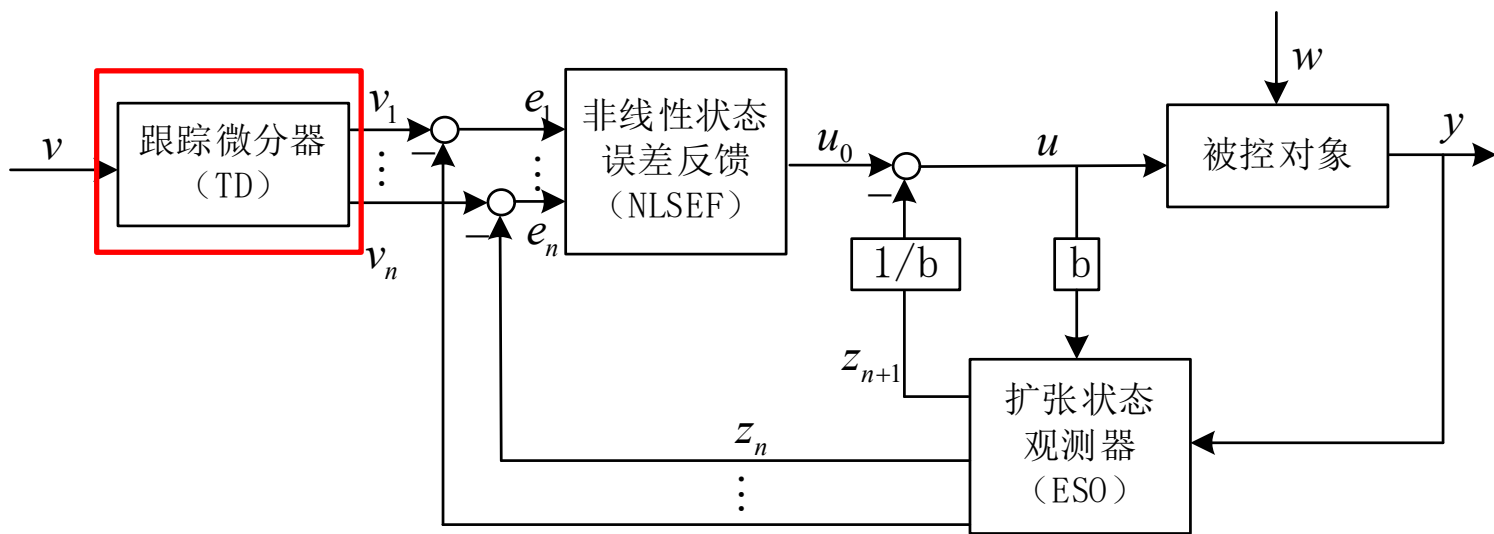


ADRC的结构及参数整定

□ 安排过渡过程和微分信号提取

(1) 定义及作用

期望值经过跟踪微分器 (TD) 会输出一个变化相对较慢且时刻跟踪期望值的过渡信号。被控对象时刻跟踪这个过渡信号，这样不仅可以解决超调与快速性矛盾，还可以使误差反馈增益和误差微分反馈增益的选取范围扩大，增强控制器的鲁棒性。另一方面TD可以得到输入的微分信号。





ADRC的结构及参数整定

□ 安排过渡过程和微分信号提取

(2) 跟踪微分器形式

二阶跟踪微分器的离散形式如下：

$$\begin{cases} u = fhan(v_1(k) - v(k), v_2(k), r, h_0) \\ v_1(k+1) = v_1(k) + hx_2(k) \\ v_2(k+1) = v_2(k) + hu \end{cases}$$

r 为可调参数， h 为控制器的执行周期，

$fhan(x_1, x_2, r, h_0)$ (最速反馈函数)的表达式如下：

$$\begin{cases} d = rh_0^2 \\ a_0 = h_0x_2 \\ y = x_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + \text{sign}(y)(a_1 - d) / 2 \\ a = (a_0 + y)\text{fsg}(y, d) + a_2(1 - \text{fsg}(y, d)) \\ u = -r(a/d)\text{fsg}(a, d) - r\text{sign}(a)(1 - \text{fsg}(a, d)) \end{cases}$$

其中 $\text{fsg}(x, d) = (\text{sign}(x + d) - \text{sign}(x - d)) / 2$



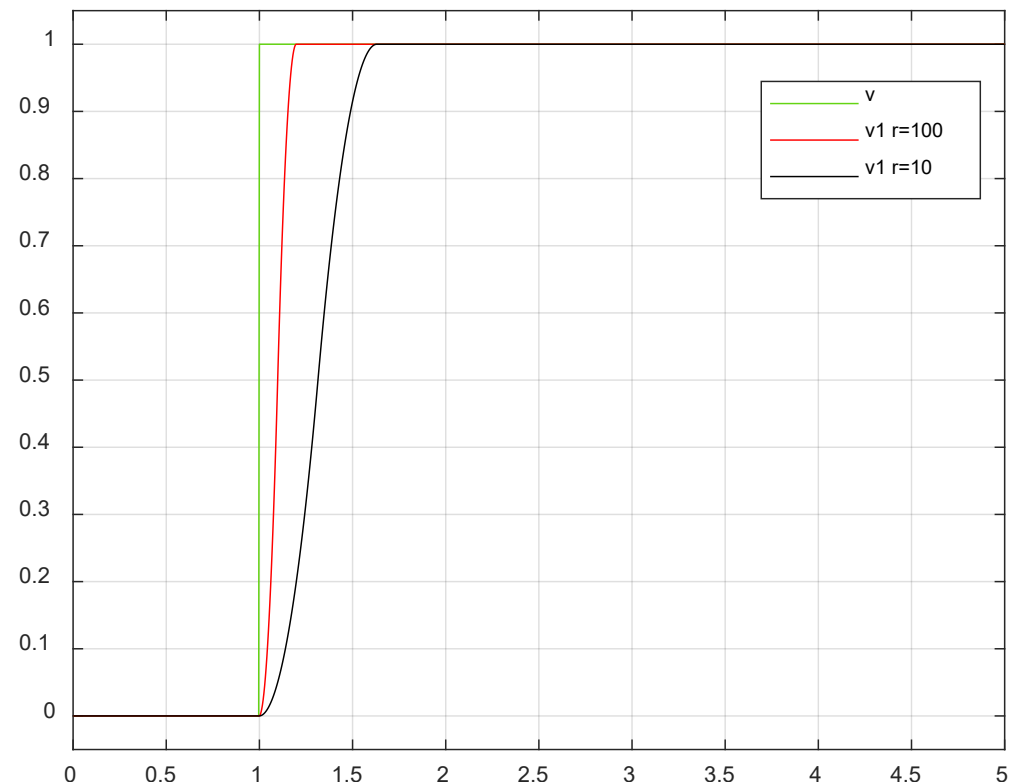
ADRC的结构及参数整定

□ 安排过渡过程和微分信号提取

(3) 参数性质

- r 越大，越快达到设定值，因此 r 被称为**速度因子**。
- 在 r 的限制下，这个方法对目标的跟踪是**时间最优**的。

$$\begin{cases} u = fhan(v_1(k) - v(k), v_2(k), r, h_0) \\ v_1(k+1) = v_1(k) + hx_2(k) \\ v_2(k+1) = v_2(k) + hu \end{cases}$$





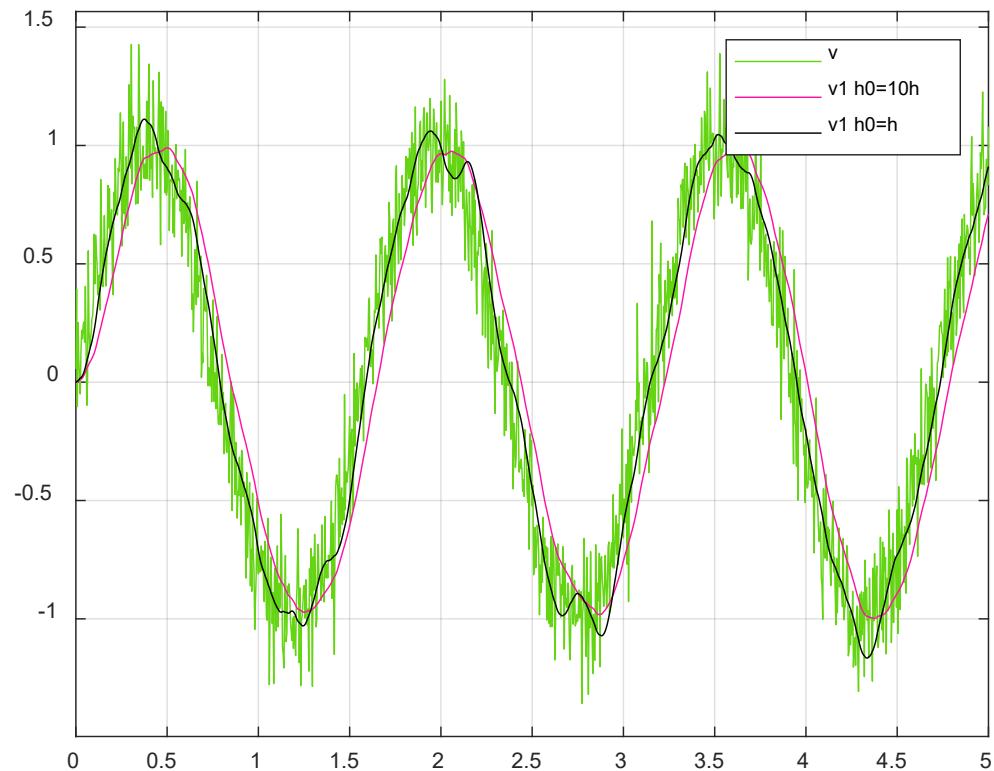
ADRC的结构及参数整定

□ 安排过渡过程和微分信号提取

(3) 参数性质

- 步长 h 越小对噪声抑制作用越强，当步长 h 确定时，增大 h_0 也能提高对噪声的抑制能力。

$$\begin{cases} u = f_{han}(v_1(k) - v(k), v_2(k), r, h_0) \\ v_1(k+1) = v_1(k) + hx_2(k) \\ v_2(k+1) = v_2(k) + hu \end{cases}$$



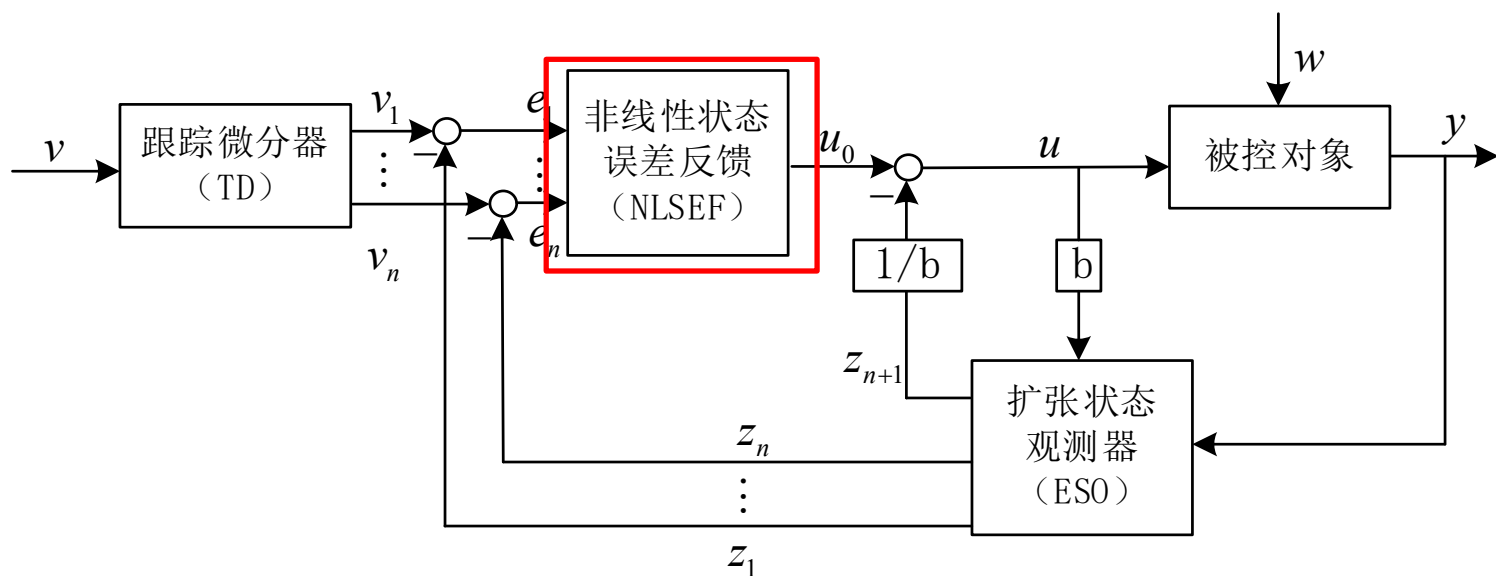


ADRC的结构及参数整定

□ 非线性状态误差反馈

(1) 定义及作用

反馈系统中有意识的引入合适的非线性结构，特别是非光滑结构，将显著改善闭环系统的动态特性。





ADRC的结构及参数整定

□ 非线性状态误差反馈

(2) 非线性状态误差反馈结构

对二阶纯积分器串联系统

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u, |u| < r \end{cases}$$

使用如下反馈控制量

$$\begin{cases} e_1 = v_1 - z_1, e_2 = v_2 - z_2 \\ u_0 = -\text{fhan}(e_1, ce_2, r, h_1) \end{cases}$$

$\text{fhan}(e_1, ce_2, r, h_1)$ 在最速反馈函数中进一步引入阻尼因子

$$\begin{cases} d = rh_0^2 \\ a_0 = h_0ce_2 \\ y = e_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + \text{sign}(y)(a_1 - d) / 2 \\ a = (a_0 + y)\text{fsg}(y, d) + a_2(1 - \text{fsg}(y, d)) \\ u = -r(a/d)\text{fsg}(a, d) - r\text{sign}(a)(1 - \text{fsg}(a, d)) \end{cases}$$



ADRC的结构及参数整定

□ 非线性状态误差反馈

(3) 参数性质

$$\begin{cases} e_1 = v_1 - z_1, e_2 = v_2 - z_2 \\ u_0 = -\text{fhan}(e_1, ce_2, r, h_1) \end{cases}$$

r : 相当于PID中的P, 加大能加快响应速度, 过大可能导致超调和大幅震荡。

c : 相当于PID中的d, 减小 c 能加快响应速度, 太小会导致超调, 大幅振荡甚至发散。

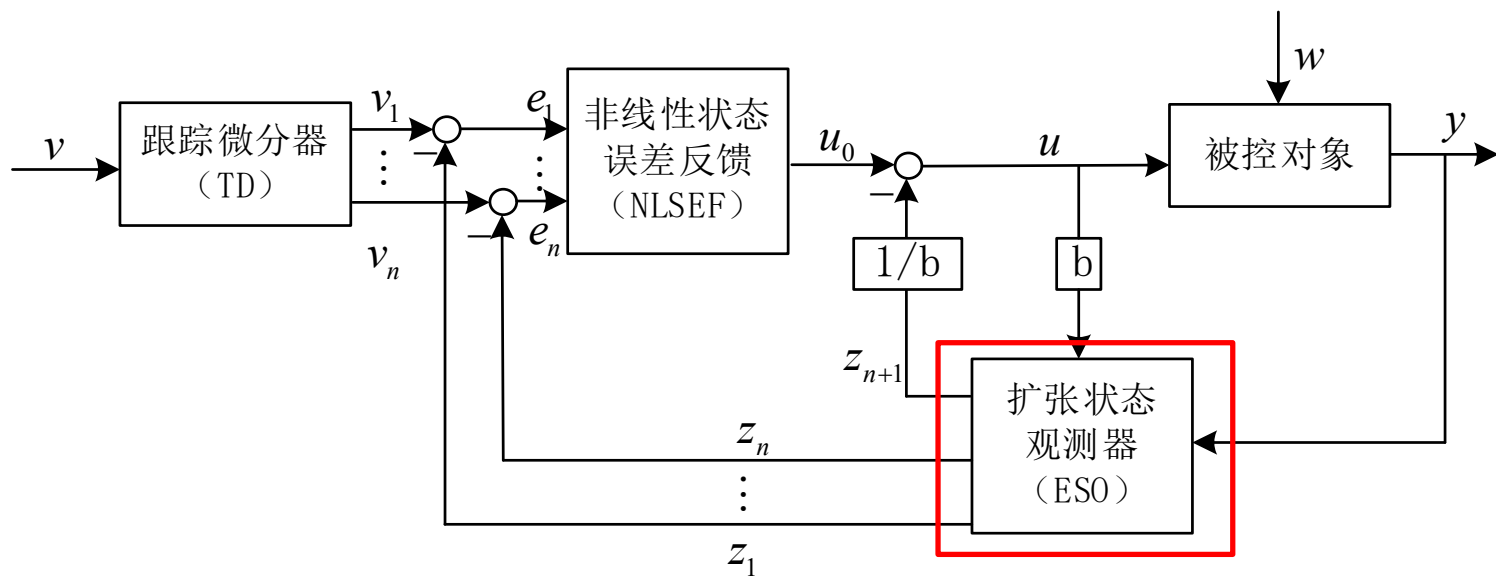


ADRC的结构及参数整定

□ 扩张状态观测器

(1) 定义及作用

自抗扰控制器的核心是扩张状态观测器（ESO）与扰动补偿。扩张状态观测器是将系统的各种扰动（包括建模、未建模动态和外扰）的总和观测出来，并在控制量中添加总和扰动的补偿项，将被控系统转换为简单的积分串联型结构，运用简单的误差反馈控制策略就能实现对系统良好的控制效果。





ADRC的结构及参数整定

□ 扩张状态观测器

(2) 扩张状态观测器结构

对非线性系统

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2) + bu, |u| < r \\ y = x_1 \end{cases}$$

我们把作用于开环系统的加速度 $f(x_1, x_2)$ 的实时作用量扩充成新的状态变量 x_3 ，记作 $x_3 = f(x_1, x_2)$ 并记 $\dot{x}_3 = w$ 。

那么系统可扩张成新的线性控制系统

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dot{x}_3 + bu, |u| < r \\ \dot{x}_3 = w \\ y = x_1 \end{cases}$$

对其建立观测器为

$$\begin{cases} e = z_1 - y, fe = \text{fal}(e, 0.5, \delta), fe_1 = \text{fal}(e, 0.25, \delta) \\ \dot{z}_1 = z_2 - \beta_{01}e \\ \dot{z}_2 = z_3 - \beta_{02}fe + bu \\ \dot{z}_3 = -\beta_{03}fe_1 \end{cases}$$



ADRC的结构及参数整定

□ 扩张状态观测器

(3) 参数性质

$$\begin{cases} e = z_1 - y, fe = \text{fal}(e, \alpha_1, \delta), fe_1 = \text{fal}(e, \alpha_2, \delta) \\ \dot{z}_1 = z_2 - \beta_{01}e \\ \dot{z}_2 = z_3 - \beta_{02}fe + bu \\ \dot{z}_3 = -\beta_{03}fe_1 \end{cases}$$

当 α 、 δ 确定，参数 β_{0i} 基本上与积分步长 h 有关，而积分步长 h 取决于被估计对象的函数 f 的作用范围。如果系统的变化比较灵敏而且快，那么要描述快速灵敏变化的运动，就需要比较小的积分步长。

δ 的取法对ESO的参数影响很大，如 δ 取大时，参数 β_{02} ， β_{03} 增大很多；相反 δ 取小时，参数 β_{02} ， β_{03} 减小很多。



ADRC的结构及参数整定

□ 扩张状态观测器

(3) 参数性质

参数 β_{01} ， β_{02} 和 β_{03} 对扩张状态观测器收敛速度有很大的影响。

- β_{01} 的值越大， z_1 跟踪对状态 y 的跟踪速度越快，通常它的大小和步长的倒数在同一数量级，而且在一定范围内变化对控制品质没有太大影响，但高于该数量级时观测器可能会发散。
- β_{02} 越大， z_2 对状态信号微分的跟踪速度越快，但是如果 β_{02} 太小会引起观测器振荡，过大会造成系统产生高频噪声。
- β_{03} 的大小会对扰动的估计很大影响， β_{03} 越大对扰动估计也就越快，但是太大会造成系统振荡。

ESO的参数 β_{01} ， β_{02} ， β_{03} 与 h 的近似关系：

$\delta=h$ 时

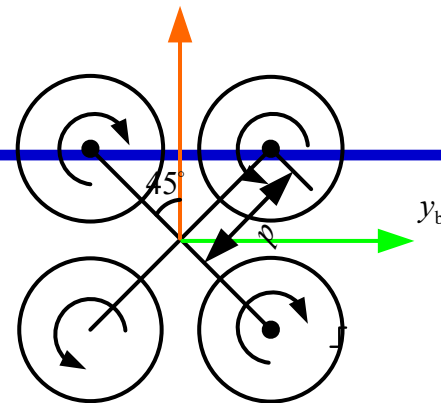
$$\beta_{01} \approx \frac{1}{h}, \beta_{02} \approx \frac{1}{1.6h^{1.5}}, \beta_{03} \approx \frac{1}{8.6h^{2.2}}$$

$\delta=5h$ 时

$$\beta_{01} \approx \frac{1}{h}, \beta_{02} \approx \frac{1}{2.4h^2}, \beta_{03} \approx \frac{1}{15.5h^3}$$



ADRC设计实例



针对如图所示四旋翼，其姿态动力学模型为：

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

$$\mathbf{J} \dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (\mathbf{J} \cdot \boldsymbol{\omega}) + \mathbf{M} + \mathbf{d}$$

其中 $\Theta \triangleq [\phi \ \theta \ \psi]^T$ 为欧拉角， $\boldsymbol{\omega} \in \mathbb{R}^3$ 是机体轴下的角速度， $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ 表示多旋翼的转惯量； $\mathbf{M} \in \mathbb{R}^3$ 表示总力矩，并且有 $\mathbf{M} = \mathbf{G}_a + \boldsymbol{\tau} + \mathbf{M}_d$ ， $\mathbf{G}_a \in \mathbb{R}^3$ 表示陀螺力矩， $\boldsymbol{\tau} \in \mathbb{R}^3$ 表示螺旋桨在机体轴上产生的力矩， $\mathbf{M}_d \in \mathbb{R}^3$ 表示气动力矩。 $\mathbf{d} \in \mathbb{R}^3$ 表示外部扰动。 \mathbf{W} 的形式如下

$$\mathbf{W} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix}$$

在小角度假设下有 $\mathbf{W} = \mathbf{I}_{3 \times 3}$ 。



ADRC设计实例

四旋翼姿态动力学模型变为：

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

$$\dot{\omega} = f(\omega, \mathbf{G}_a, \mathbf{M}_d, \mathbf{d}) + \mathbf{J}^{-1}\boldsymbol{\tau}$$

$f(\omega, \mathbf{G}_a, \mathbf{M}_d, \mathbf{d})$ 表示内部扰动和外部扰动的和。

这里分别对三个欧拉角设计ADRC控制器。
以横滚角为例，若令

$$x_1 = \phi, x_2 = \omega_x, x_3 = f(\omega, \mathbf{G}_{a_x}, \mathbf{M}_{d_x}, \mathbf{d}_x), b = \mathbf{J}_x^{-1}, u = \boldsymbol{\tau}_x$$

将上式表述的二阶系统扩维成三阶系统，则有

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = w(t) \\ y = x_1 \end{cases}$$

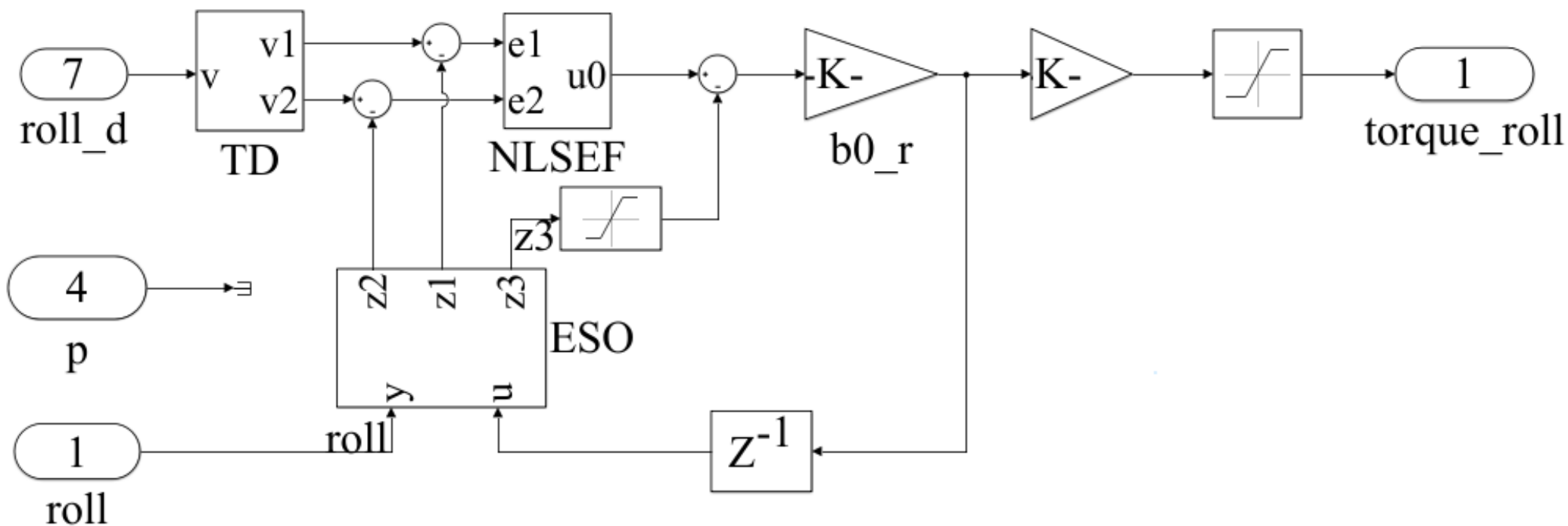
对其设计状态观测器为

$$\begin{cases} e = z_1 - y \\ \dot{z}_1 = z_2 - \beta_{01}e \\ \dot{z}_2 = z_3 - \beta_{02}fal(e, 0.5, \delta) + bu \\ \dot{z}_3 = -\beta_{03}fal(e, 0.25, \delta) \end{cases}$$



ADRC设计实例

Simulink模型：





资源

- 本课程的所有课件、视频和源码将会发布在官方网站：<https://rflysim.com/zh>
- 更详细的内容可以参考我们的教材：全权, 戴训华, 王帅著. 《多旋翼飞行器设计与控制实践》. 北京: 电子工业出版社. 2020
- 扫二维码关注可靠飞行控制研究组公众号 **buarfly** (文章、资讯等)
- 多旋翼控制实践课程交流QQ群: **951534390** (答疑、资料分享等)
- 如果遇到任何问题, 也可以在我们的Github页面查找答案或提问
<https://github.com/RflySim/RflyExpCode/issues>
- 针对无人机/无人车的视觉/集群/AI等顶层控制算法的快速开发与验证, RflySim同时推出了高级版平台和课程, 请访问我们的网站了解更多
https://rflysim.com/zh/4_Pro/Advanced.html
- 我们的官方联系邮箱是: rflysim@163.com



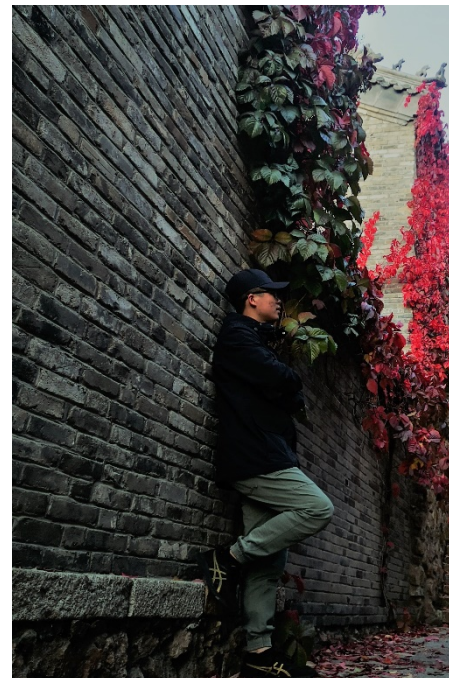


致谢

感谢控制组同学



王帅



刘浩

为本门课程准备做出的贡献



谢谢!