



---

# 多旋翼飞行器设计与控制 实践

## 第十讲 定点位置控制器设计实验

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



# 大纲

---

1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 小结



# 基本原理

## □ 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

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

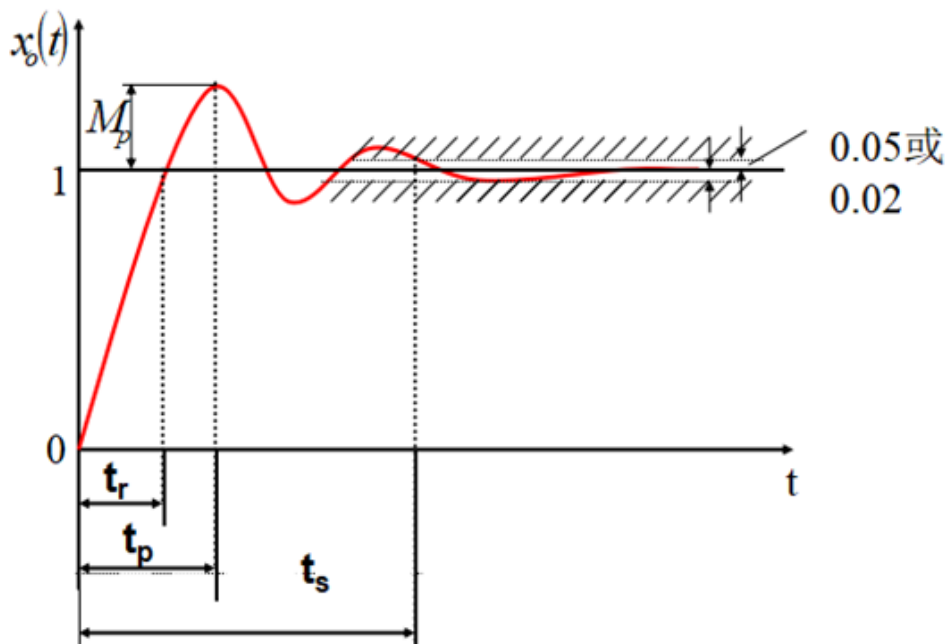


图. 二阶系统阶跃响应





# 基本原理

## □ 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。

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

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

其中： $\omega_c$ 表示截止频率， $L(\omega_c)=0\text{dB}$ 。

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 $\gamma$ 和 $h$ 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：

$$\gamma > 40^\circ$$

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

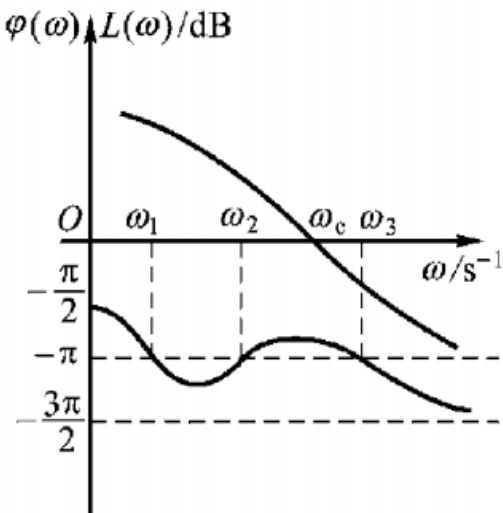


图. 稳定裕度

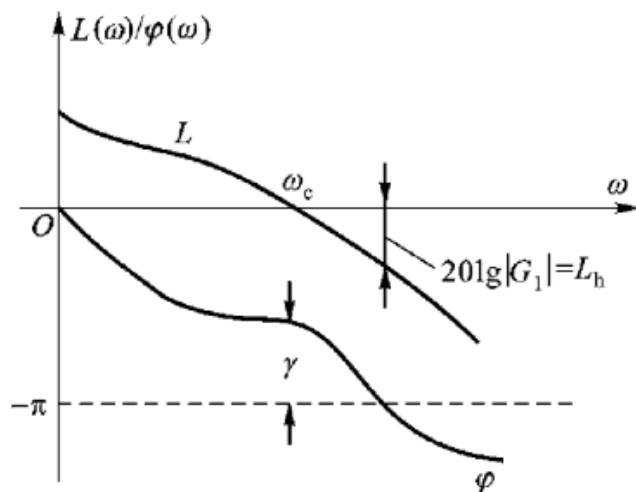


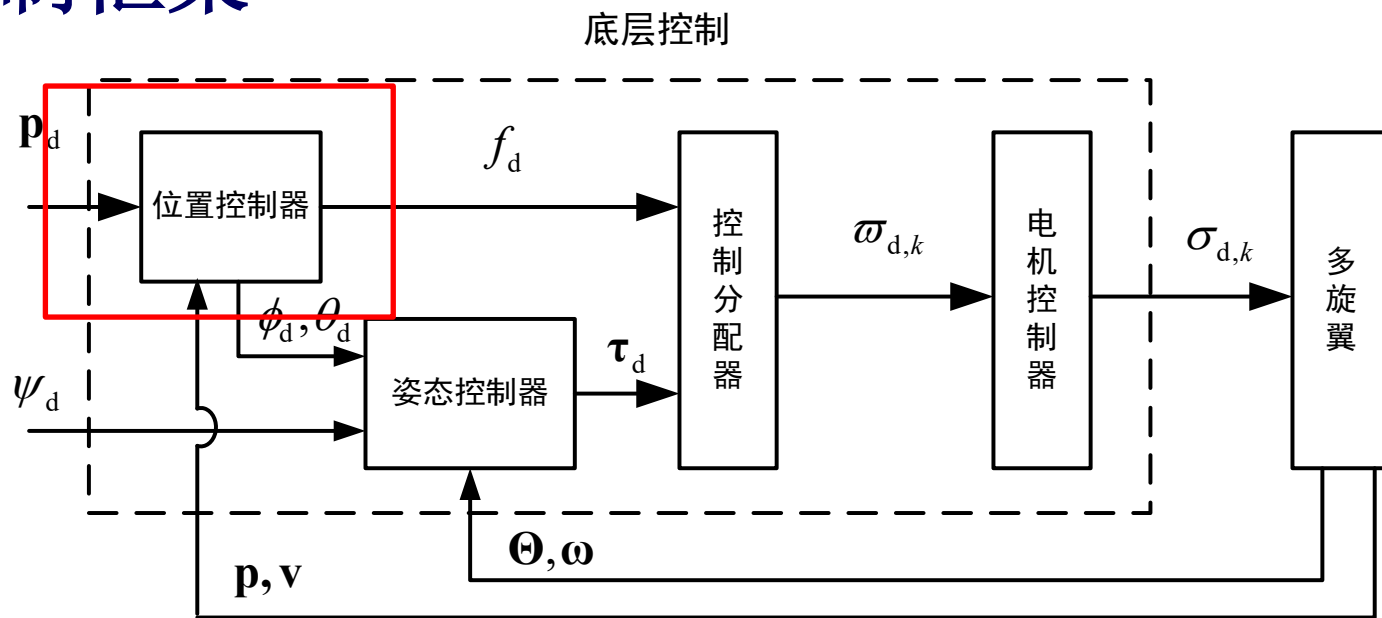
图. 对数频率特性曲线





# 基本原理

## □ 底层控制框架



欠驱动系统：4个输入（总拉力 $f$ 和三轴力矩 $\tau$ ）控制6个输出（位置 $p$ 和姿态角 $\Theta$ ）。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。





# 基本原理

## □ 位置控制

### (1) 传统的PID设计

#### ■ 水平通道模型

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

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

#### ■ 期望水平位置动态

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

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

当考虑定点控制时

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



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

其中  $\mathbf{K}_{(\cdot)}$  表示参数。





# 基本原理

## □ 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

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



# 基本原理

## □ 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

$$\dot{\mathbf{p}}_h = \mathbf{v}_h \quad \Rightarrow \quad \mathbf{v}_{hd} = \mathbf{K}_{\mathbf{p}_h} (\mathbf{p}_{hd} - \mathbf{p}_h)$$

在  $\dot{\mathbf{p}}_{hd} = 0$  的前提下如果

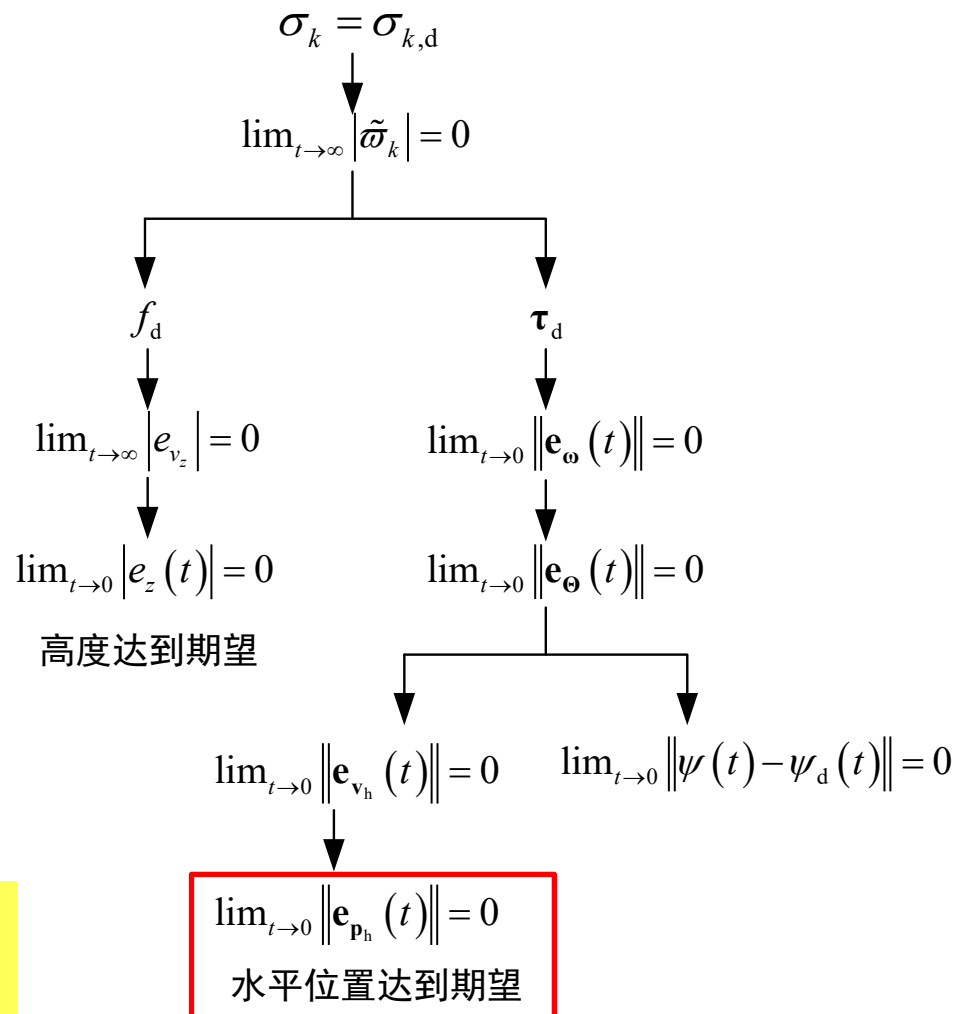
$$\lim_{t \rightarrow \infty} \|\mathbf{e}_{\mathbf{v}_h}(t)\| = 0$$

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







# 基本原理

## □ 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

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

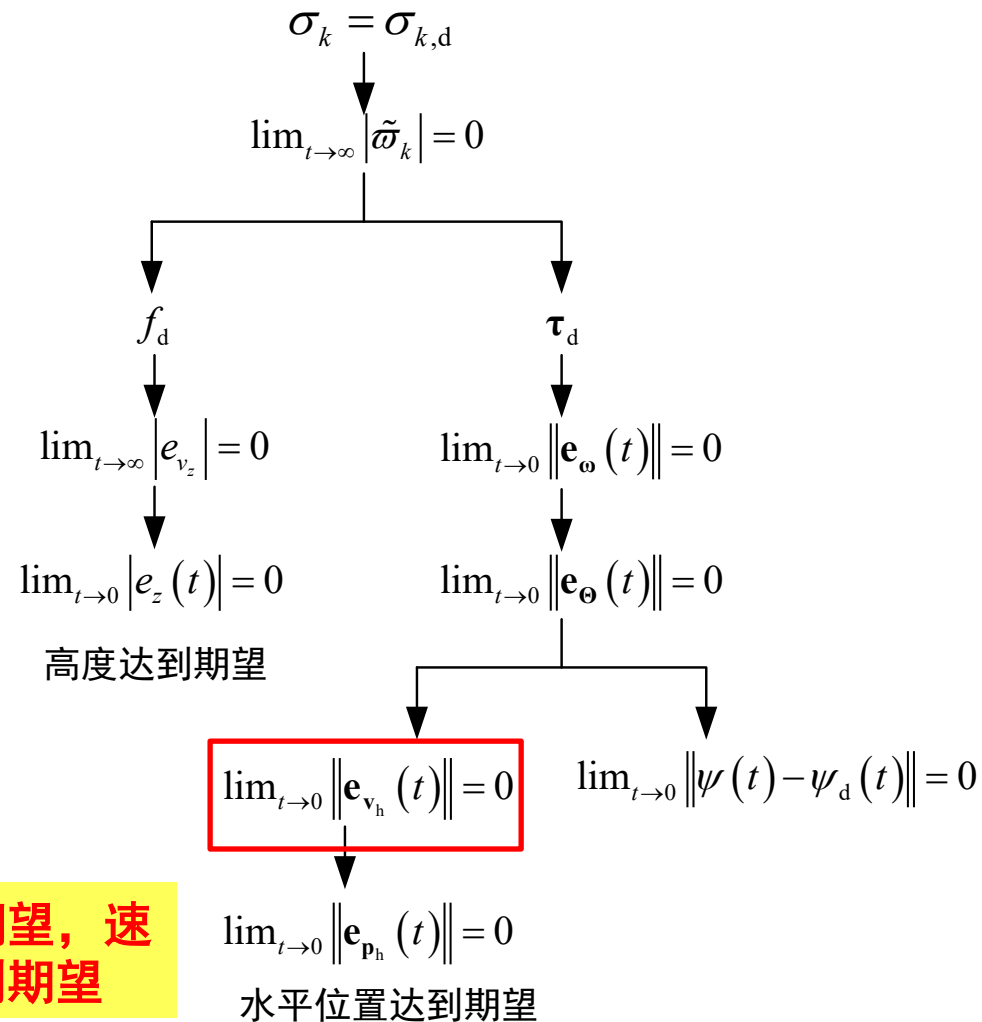
$$-g\mathbf{A}_\psi \Theta_{hd} = -\mathbf{K}_{vhp} \mathbf{e}_{v_h} - \mathbf{K}_{vhi} \int \mathbf{e}_{v_h} - \mathbf{K}_{vhd} \dot{\mathbf{e}}_{v_h}$$

$$\Theta_{hd} = g^{-1} \mathbf{A}_\psi^{-1} \left( \mathbf{K}_{vhp} \mathbf{e}_{v_h} + \mathbf{K}_{vhi} \int \mathbf{e}_{v_h} + \mathbf{K}_{vhd} \dot{\mathbf{e}}_{v_h} \right)$$

如果  $\lim_{t \rightarrow \infty} \|\Theta_h(t) - \Theta_{hd}(t)\| = 0$

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





# 基本原理

## □ 位置控制

### (2) 开源自驾仪PID设计

#### 2) 高度通道模型

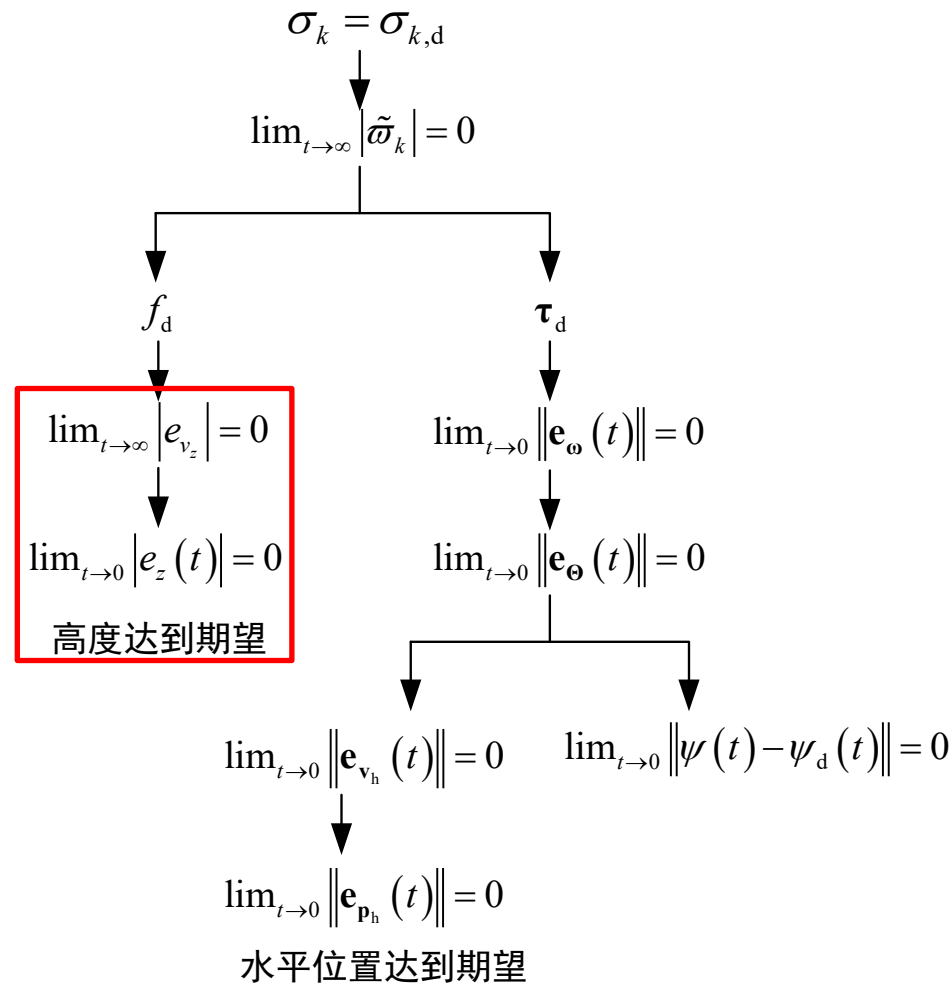
类似于水平通道设计，高度通道

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





# 基本原理

## □ 位置控制

### (3) 加饱和的PID控制

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

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

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





# 基本原理

## □ 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 $e_{v_h}$ 和控制器右端的限幅

$$\mathbf{e}_{v_h} = \text{sat}_{\text{gd}}(\mathbf{v}_h - \mathbf{v}_{hd}, a_1)$$

$$\mathbf{\Theta}_{hd} = \text{sat}_{\text{gd}}\left(\mathbf{g}^{-1}\mathbf{A}_{\psi}^{-1}\left(\mathbf{K}_{v_h p}\mathbf{e}_{v_h} + \mathbf{K}_{v_h i}\int \mathbf{e}_{v_h} + \mathbf{K}_{v_h d}\dot{\mathbf{e}}_{v_h}\right), a_2\right)$$

其中 $a_1, a_2 \geq 0$ 。保方向饱和函数 $\text{sat}_{\text{gd}}(\mathbf{u}, a)$ 的定义如下

$$\text{sat}_{\text{gd}}(\mathbf{u}, a) \triangleq \begin{cases} \mathbf{u}, & \|\mathbf{u}\|_{\infty} \leq a \\ a \frac{\mathbf{u}}{\|\mathbf{u}\|_{\infty}}, & \|\mathbf{u}\|_{\infty} > a \end{cases}$$

保方向饱和函数 $\text{sat}_{\text{gd}}(\mathbf{u}, a)$ 与传统的饱和函数 $\text{sat}(\mathbf{u}, a)$ 的区别：饱和函数限制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，不仅可以限制最终分量每个分量的绝对值不大于 $a$ ，还可以保证它的方向与 $\mathbf{x}$ 相同。



# 基本原理

## □ 位置控制

### (3) 加饱和的PID控制

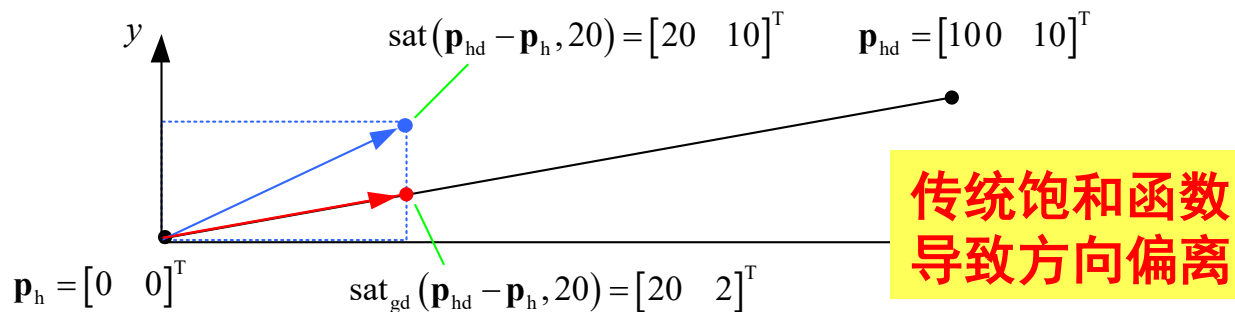


图. 两种饱和函数的作用效果对比图

传统饱和函数

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

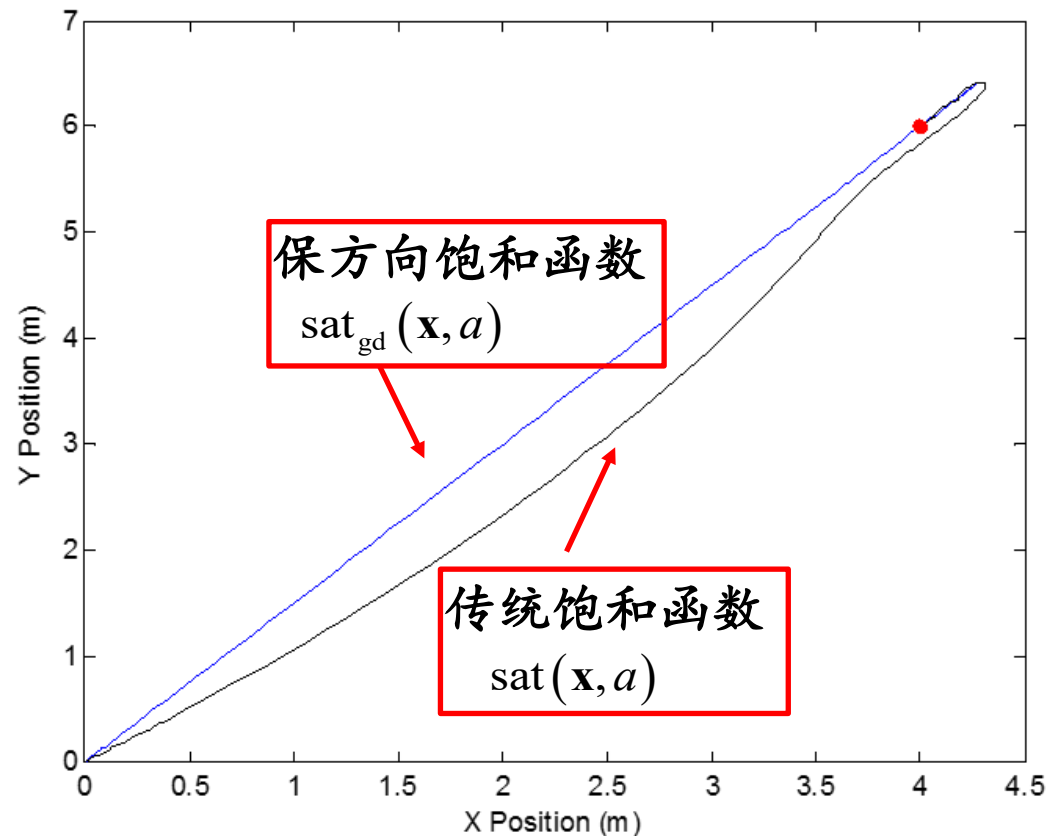


图. 饱和情况下多旋翼定点跟踪水平面运动轨迹

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





# 基本原理

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

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

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

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



# 基本原理

## □ 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。

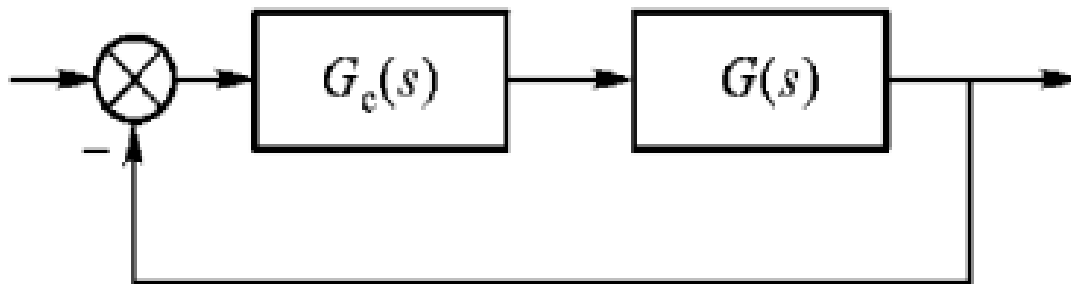


图. 系统串联校正



# 基本原理

## □ 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:

$$\omega_m = \frac{1}{\sqrt{a}T}$$

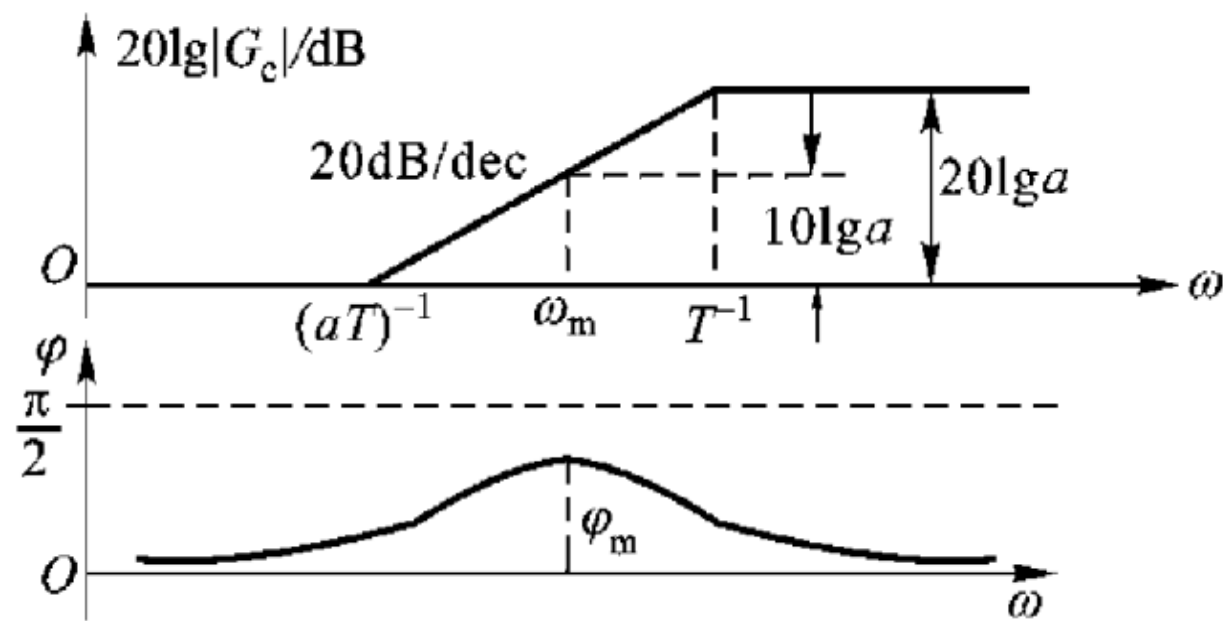


图. 超前校正幅频曲线







# 基本原理

## □ 控制系统校正

### (2) 相位滞后校正

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

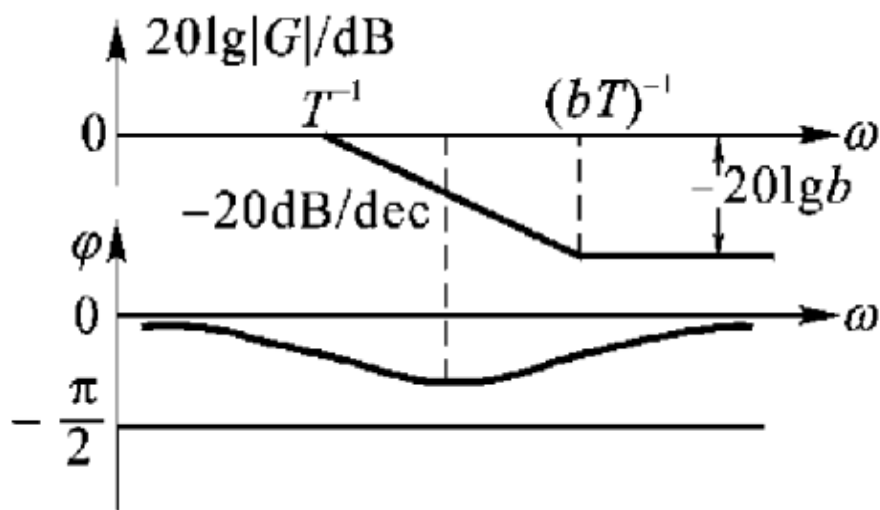


图. 滞后校正幅频曲线

### (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  $bT_1 > aT_2$

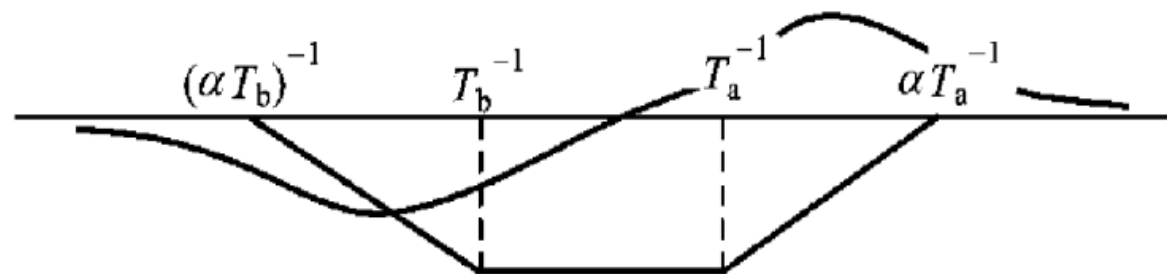


图. 滞后超前校正幅频曲线





# 基本原理

---

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：Matlab2017b，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“e6.1” (<https://flyeval.com/course>)。

### ■ 目标

- (1) 复现四旋翼Simulink仿真，分析控制作用在  $o_b x_b$  轴和  $o_b y_b$  轴的解耦。
- (2) 对系统进行扫频以绘制bode图，分析闭环位置控制系统稳定裕度；
- (3) 完成硬件在环仿真。





# 基础实验

## □ 实验步骤

### (1) 步骤一：Simulink仿真——通道解耦

1) 参数初始化。打开“e6/e6.1/SIM/Init\_control.m” Simulink文件进行参数初始化，“PosControl\_Sim.slx”将会自动打开，如下图。

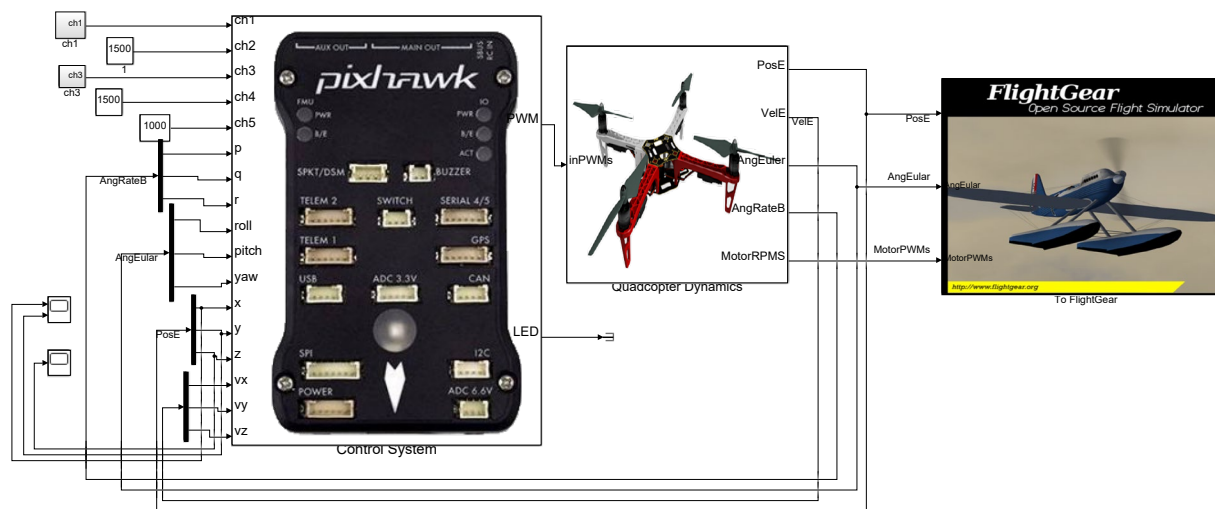


图. 位置控制仿真截图



# 基础实验

## □ 实验步骤

### 2) 开始仿真

打开FlightGear-F450程序，然后点击Simulink“Run”按钮开始仿真。此时可以在FlightGear中观察四旋翼的状态，可以看到四旋翼先上升到空中，然后向 $o_e y_e$ 轴方向飞行（视野中向右），最后悬停。



图. FlightGear中的多旋翼







# 基础实验

## □ 实验步骤

### 3) 控制效果

高度状态变化如图所示。多旋翼先上升后保持悬停状态。

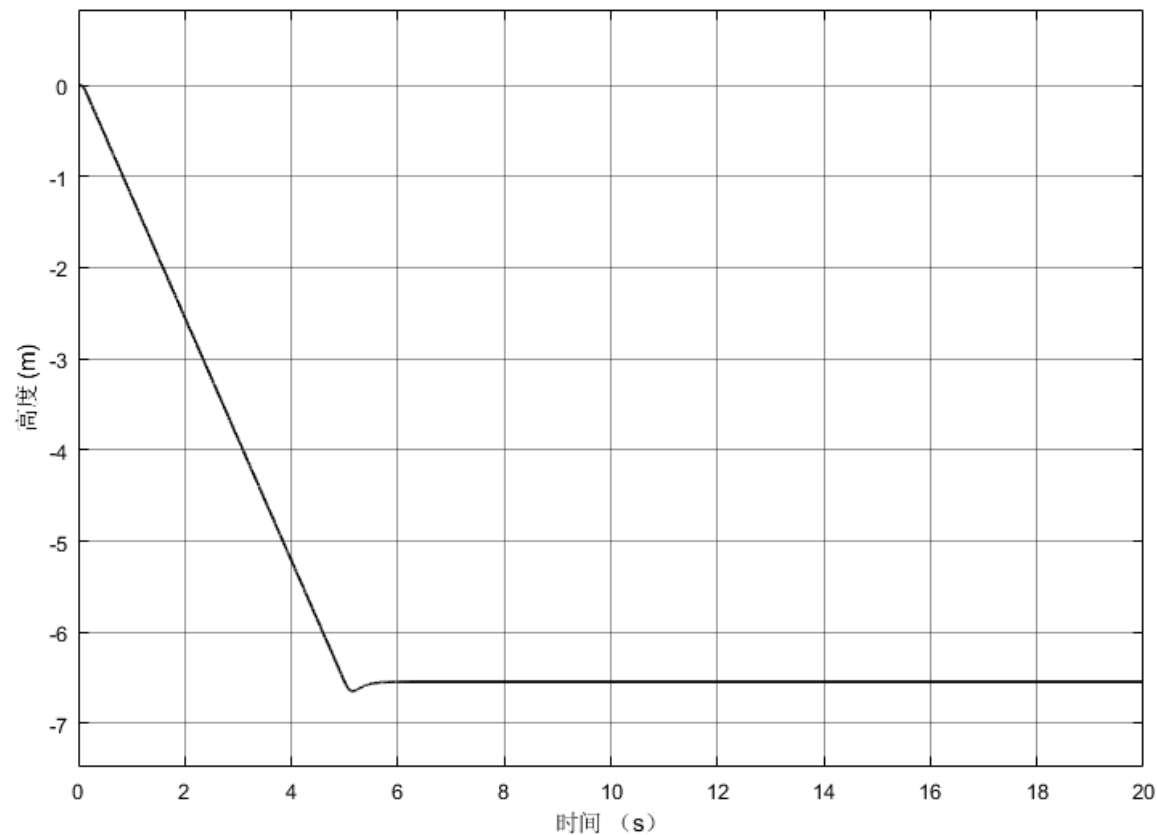


图. 高度响应





# 基础实验

## □ 实验步骤

### 4) 通道解耦分析

四旋翼的水平位置如图所示。可以看到只对  $O_b y_b$  轴位置进行控制不会对  $O_b x_b$  轴的状态产生影响, 说明  $O_b x_b$ 、 $O_b y_b$  轴的控制已解耦。

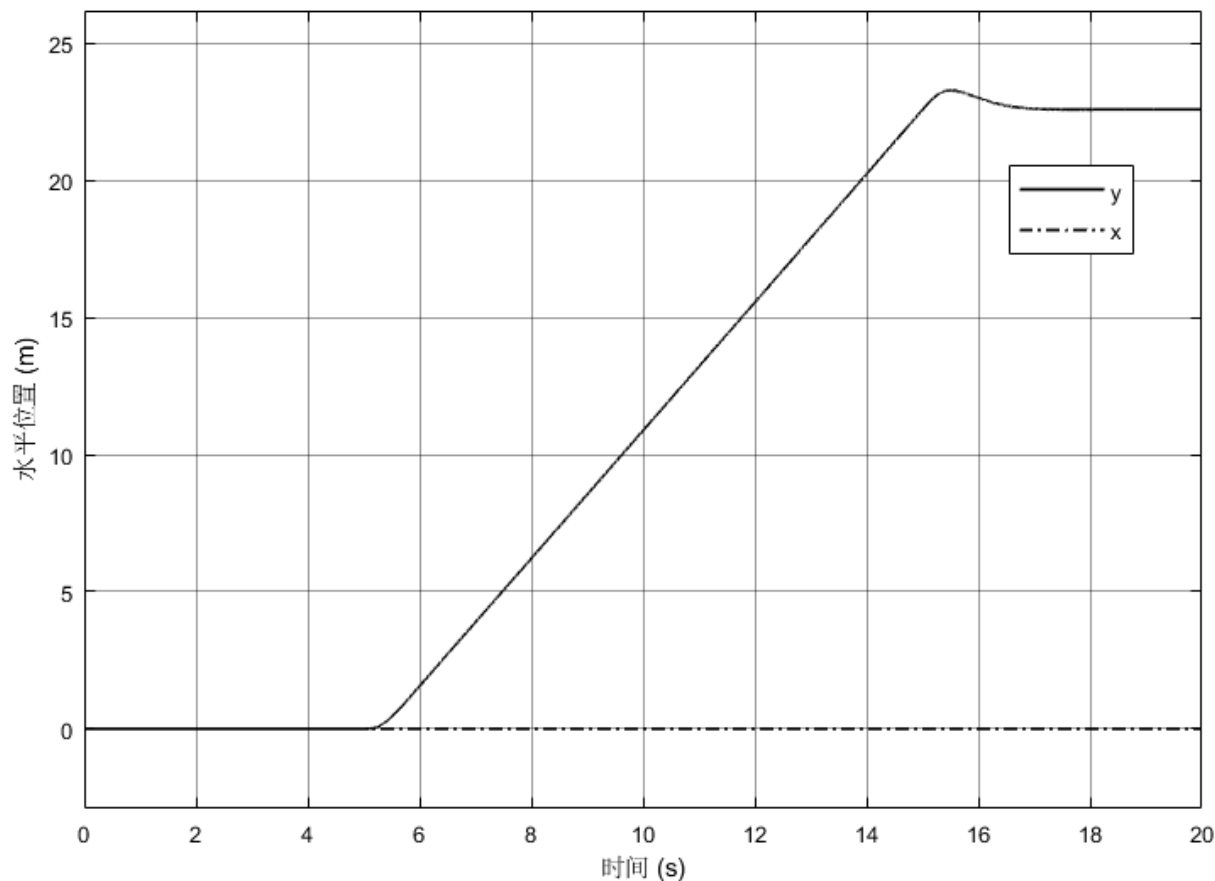


图. 水平位置响应





# 基础实验

## □ 实验步骤

### (2) 步骤二：Simulink仿真——稳定裕度

#### 1) 打开文件

“e6/e6.1/tune/Init\_control.m”文件初始化参数，“PosControl\_tune.slx”将会自动打开。

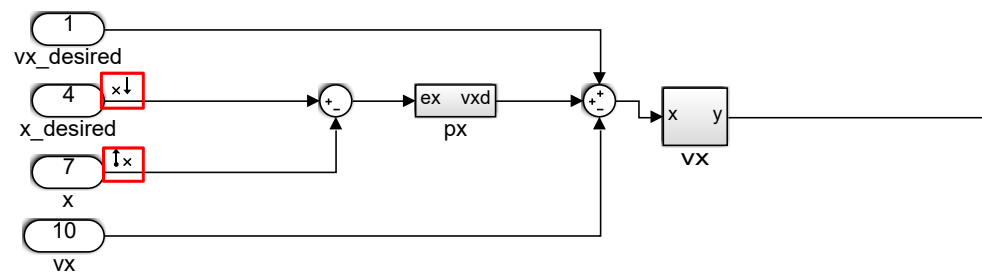


图. 输入输出点设置

打开上述文件的“Control System”子模块中的“position\_control”模块，将期望x通道输入线设为Open-loop Input，x通道的实际输出设置为Open-loop Output。







# 基础实验

## □ 实验步骤

- 2) 选择Simulink上面菜单中的“Analysis”，在下拉菜单中选择“Control Design”，选择“Linear Analysis”。
- 3) 在弹出的窗口中选择“LINEAR ANALYSIS”，点击Bode得到Bode图。
- 4) 在曲线中点击鼠标右键，选择“Characteristics” - “All Stability Margins”，可得到幅值裕度为15.3dB, 此时的频率为3.97rad/s; 相位裕度为65.5, 频率为1.04rad/s。

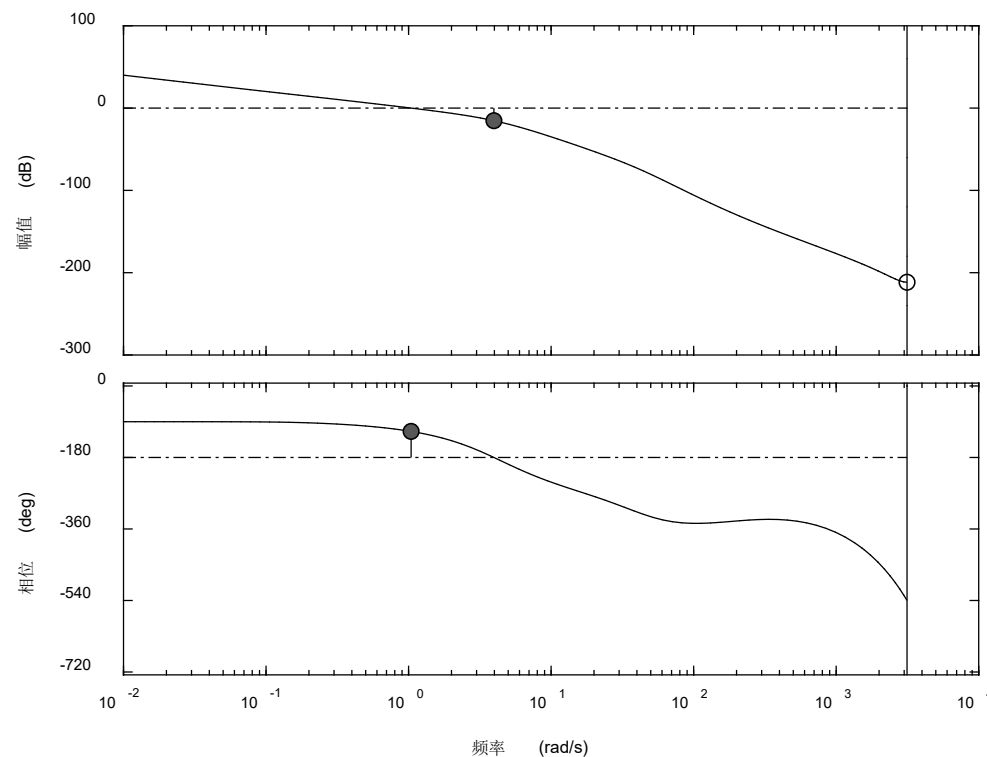


图. x通道控制系统开环Bode图





# 基础实验

## □ 实验步骤

### (3) 步骤三：硬件在环仿真

#### 1) 打开硬件在环仿真文件

运行文件

“e6/e6.1/HIL/Init\_control.m” 进行参数初始化，“PosControl\_HIL.slx” 将会自动打开，如右图所示。

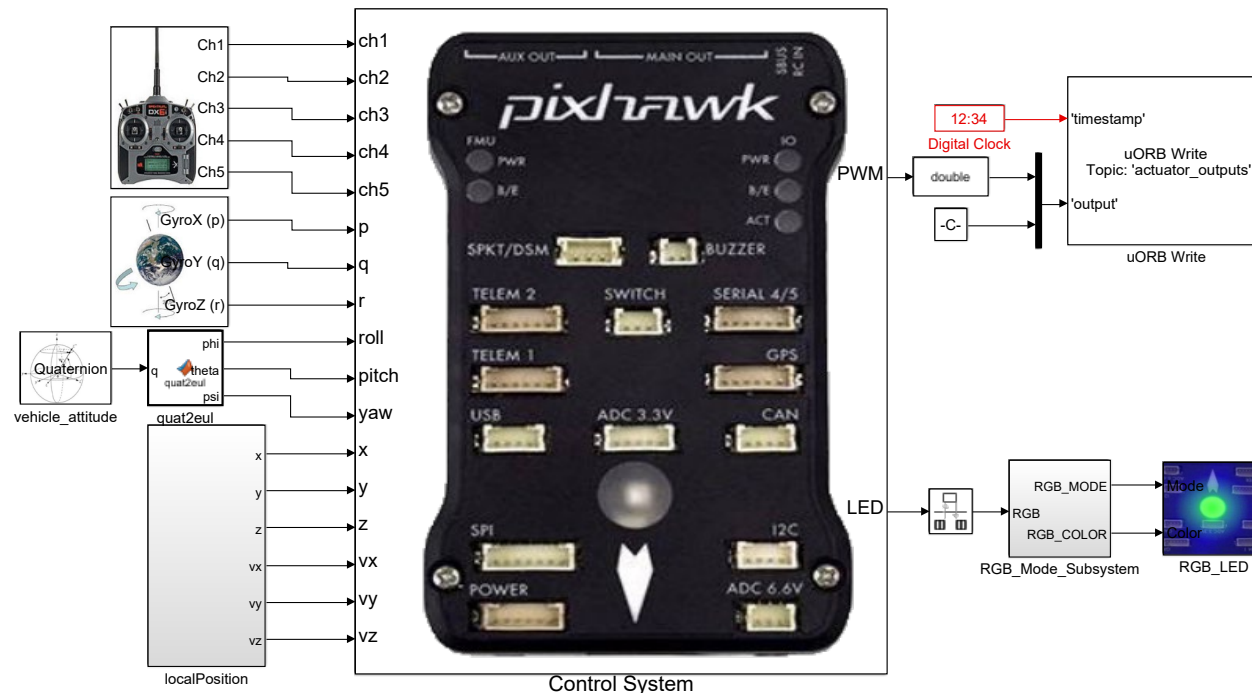


图. 验证通道解耦的硬件在环仿真模型





# 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**”。

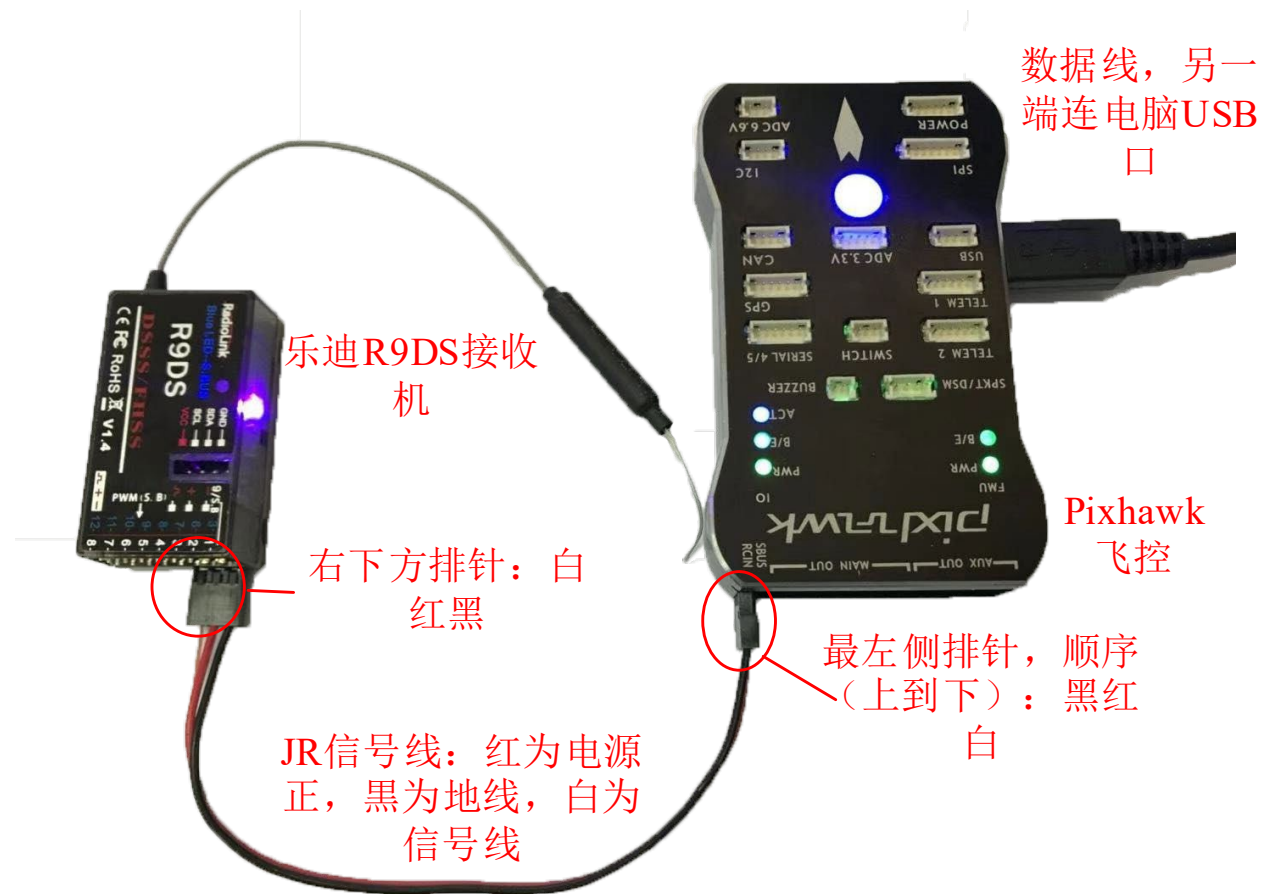


图. 硬件系统连接



# 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自驾仪中。这样就可以在Pixhawk自驾仪中运行我们自己设计的姿态控制程序。

The image shows the process of compiling and downloading code from a Simulink model to a Pixhawk autopilot. It consists of three main parts:

- Top Window:** A Simulink window titled "E1\_rgbled\_system - Simulink". The "Code" menu is open, and the "PX4 PSP: Upload code to Px4FMU" option is highlighted. A red arrow points from this option to the command window below.
- Bottom Left Window:** A command window titled "C:\Windows\SYSTEM32\cmd.exe" showing the output of the compilation process. The output includes the following text:

```
### Successfully generated all binary outputs.  
Loaded firmware for 9.0, size: 875004 bytes, waiting for the bootloader...  
if the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.  
PX4_SIMULINK = y  
attempting reboot on COM3...  
if the board does not respond, unplug and re-plug the USB connector.  
Found board 9.0 bootloader rev 4 on COM3  
50583400 00ac2600 00100000 00ffffff ffffffff ffffffff ffffffff 66ed47ff ff73cc15 c8ad940c dbc59f39 d6c20e06 f95  
3d3ef f3073019 d035ab0d 3f60334e 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8f  
5c 5e07decc 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e  
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4  
idtype: =00  
vid: 000026ac  
pid: 00000010  
coa: ZuiH//9zzBXIrZQM28Wf0dbCDgb5U9Pv8wcwGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2m17xjQIMhvmjt01bJHjuPXF4H3syca+W1RYo  
VEOu7vCHto1z1qLhApe8BnKXIm7GDuwDwwAbbG1Zzdf9XHKQd1CSqZi4=  
sn: 0038001f3432470d31323533  
Erase : [=====] 100.0%  
Program: [=====] 100.0%  
Verify : [=====] 100.0%  
Rebooting.  
H:
```
- Bottom Right Window:** A command window titled "C:\Windows\SYSTEM32\cmd.exe" showing the output of the download process. The output includes the following text:

```
### Successfully generated all binary outputs.  
Loaded firmware for 9.0, size: 875004 bytes, waiting for the bootloader...  
if the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.  
PX4_SIMULINK = y  
attempting reboot on COM3...  
if the board does not respond, unplug and re-plug the USB connector.  
Found board 9.0 bootloader rev 4 on COM3  
50583400 00ac2600 00100000 00ffffff ffffffff ffffffff ffffffff 66ed47ff ff73cc15 c8ad940c dbc59f39 d6c20e06 f95  
3d3ef f3073019 d035ab0d 3f60334e 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8f  
5c 5e07decc 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e  
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4  
idtype: =00  
vid: 000026ac  
pid: 00000010  
coa: ZuiH//9zzBXIrZQM28Wf0dbCDgb5U9Pv8wcwGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2m17xjQIMhvmjt01bJHjuPXF4H3syca+W1RYo  
VEOu7vCHto1z1qLhApe8BnKXIm7GDuwDwwAbbG1Zzdf9XHKQd1CSqZi4=  
sn: 0038001f3432470d31323533  
Erase : [=====] 100.0%  
Program: [=====] 100.0%  
Verify : [=====] 100.0%  
Rebooting.  
H:
```

Red arrows indicate the flow from the Simulink window to the command window, and from the command window to the download completion message.

点击编译

点击下载

下载完成

图. 代码编译及下载流程





# 基础实验

## 实验步骤

### 4) CopterSim配置

双击CopterSim桌面快捷方式即可以打开多旋翼模拟器软件。依次点击按钮“模型参数”-“存储并使用参数”，软件会自动匹配串口号，如果串口号有错请手动选择，再点击“开始仿真”按钮就可以进入半物理仿真模式。此时可以看到如图所示的界面左下角收到飞控返回的相关消息，以及PixHawk飞控上的灯光从蓝色变为绿色闪烁。

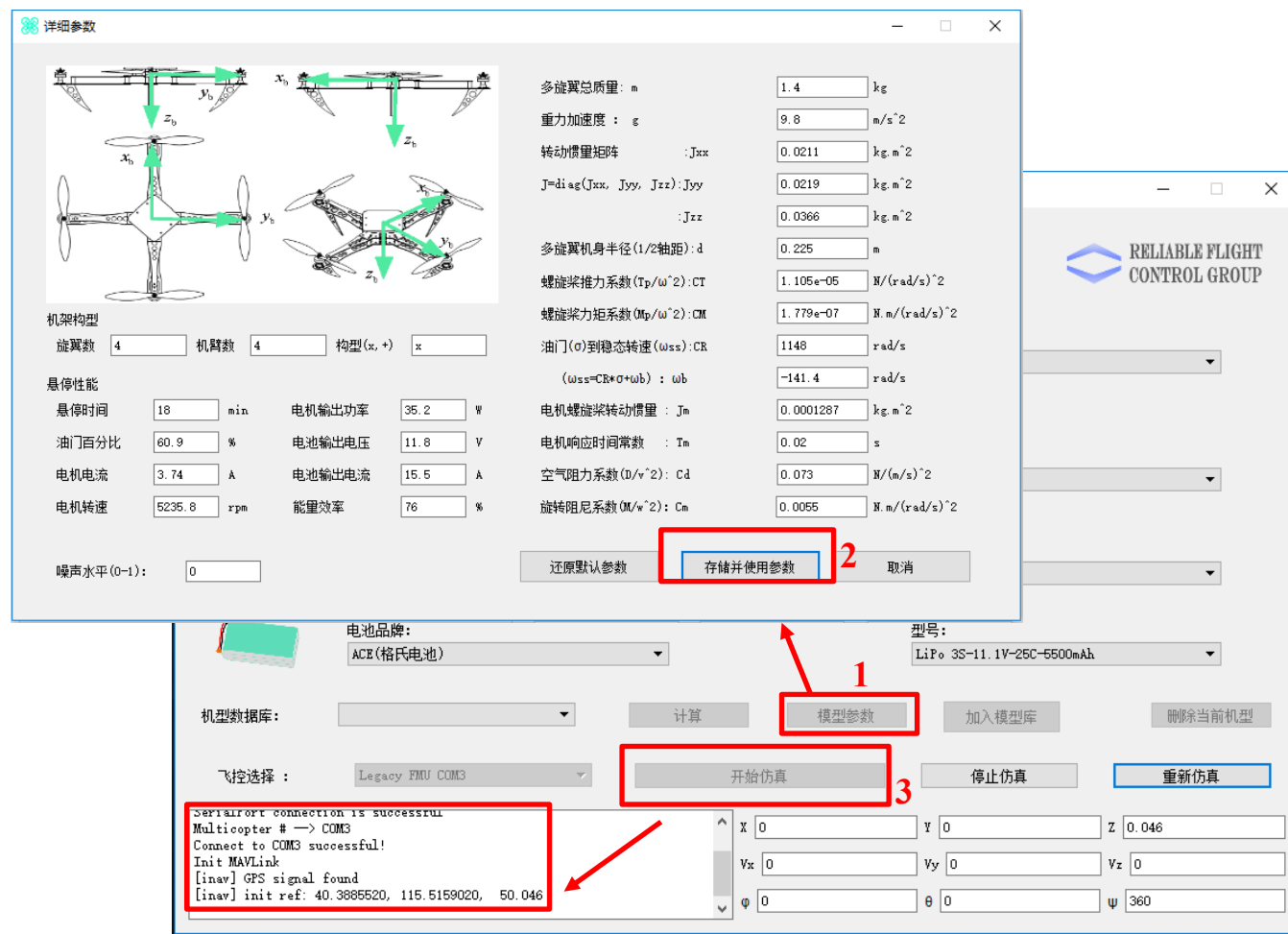


图. CopterSim用户界面





# 基础实验

## 实验步骤

### 5) 3D Display配置

双击文件3DDisplay桌面快捷方式打开三维显示软件。

### 6) 仿真效果

使用遥控器解锁多旋翼，可以控制多旋翼悬停，以指定速度飞行。当全部摇杆都在中间位置时，四旋翼保持悬停。



图. 3DDisplay用户界面





# 分析实验

---

## □ 实验目标

### ■ 已知

(1) 软件: Matlab2017b, 基于Simulink的控制器设计与仿真平台, 实验指导包“e6.2” (<https://flyeval.com/course>)。

### ■ 目标

- (1) 调节PID控制器的相关参数改善系统控制性能, 并记录超调量和调节时间, 试得到一组满意的参数。
- (2) 在得到满意参数后, 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

PID参数步骤与姿态控制的参数调试步骤相同。先调试内环速度环，再调试外环的位置环，先调高度再调水平位置。调试文件在” e6/e6.2/tune” 文件夹中。调节参数的初始状态应是飞行器处于高空悬停状态，将初始高度设置为100m, 电机的初始转速设置为557.1420rad/s，这个初始条件对应于飞行器在空中100m处悬停。修改 “Init\_control.m” 文件中的对应参数如右。

ModelInit\_PosE=[0,0,-100];

ModelInit\_VelB=[0,0,0];

ModelInit\_AngEuler=[0,0,0];

ModelInit\_RateB=[0,0,0];

ModelInit\_RPM=557.1420;







# 分析实验

## □ 实验步骤

### (2) 步骤二：速度控制环参数调节

首先调节内环PID参数。打开

“e6/e6.2/tune/PosControl\_tune.slx” 文件中的Control System子模块中的position\_control模块，即为位置控制系统模型。将其中x通道的速度期望部分换成阶跃输入，并将输入输出设置为“Enable Data logging”。

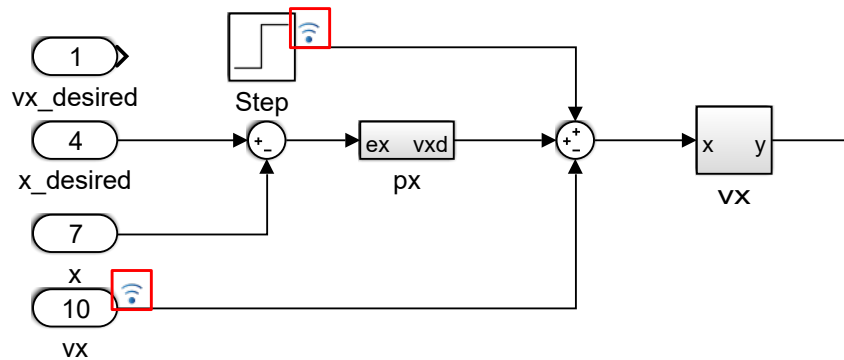


图. 速度控制环调试模型





# 分析实验

## □ 实验步骤

### (2) 步骤二：速度控制环参数调节

在“Init\_control.m”文件中修改内环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“Run”按钮开始仿真，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值，得到阶跃响应曲线如图。

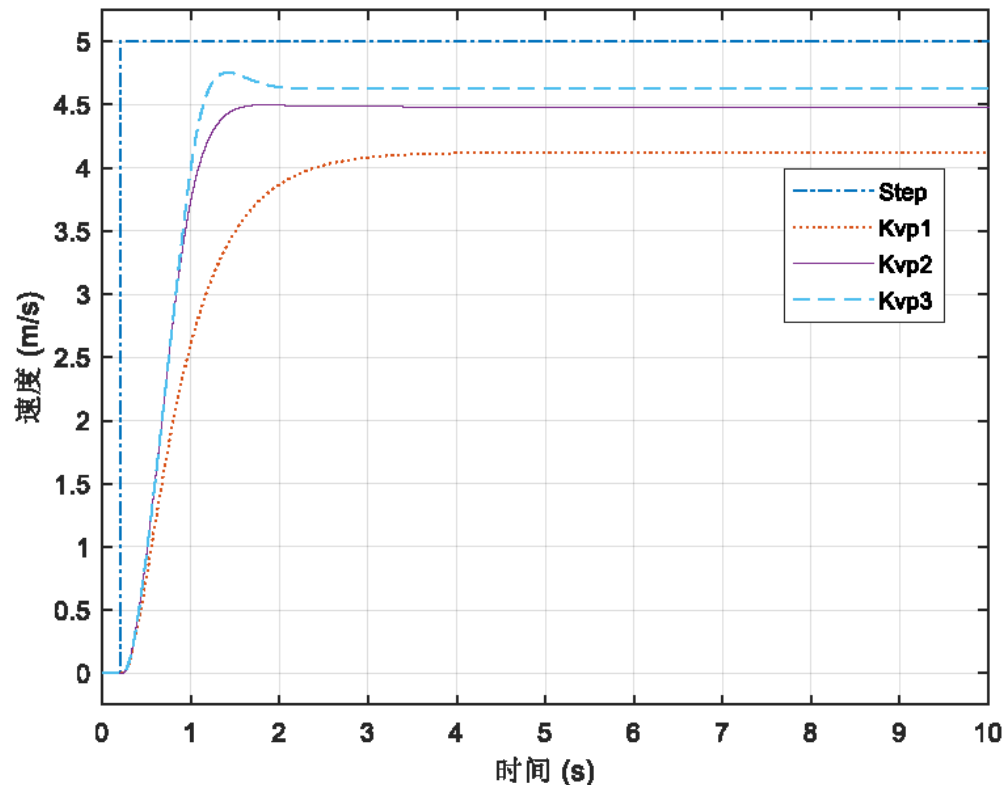


图. 不同比例项系数下的阶跃响应





# 分析实验

## □ 实验步骤

### (2) 步骤二：速度控制环参数调节

加入微分项和积分项，并且微调比例项参数，最终得到的响应曲线如图

$$K_{vxp}=2.5;$$

$$K_{vxi}=0.4;$$

$$K_{vxd}=0.01;$$

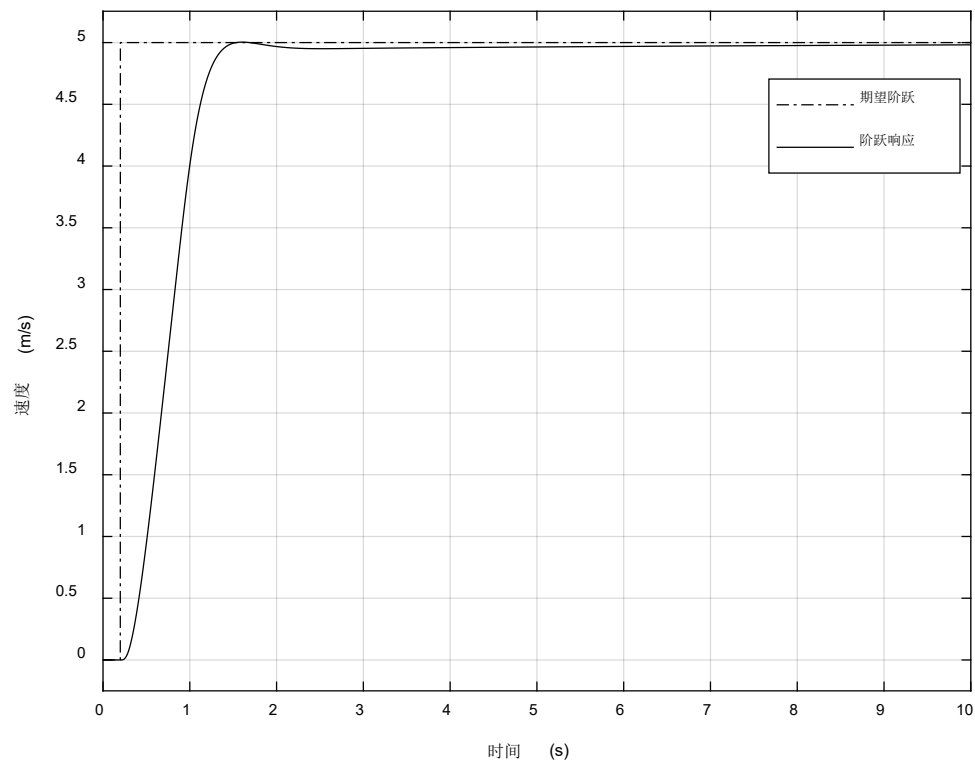


图. 满意参数下的阶跃响应





# 分析实验

## □ 实验步骤

### (3) 步骤三：位置环参数调节

使用步骤二中得到的速度环参数，在“PosControl\_tune.slx”文件中，将“x\_desired”换为阶跃输入，并将阶跃输入和“x”信号线设置为“Enable Data Logging”。

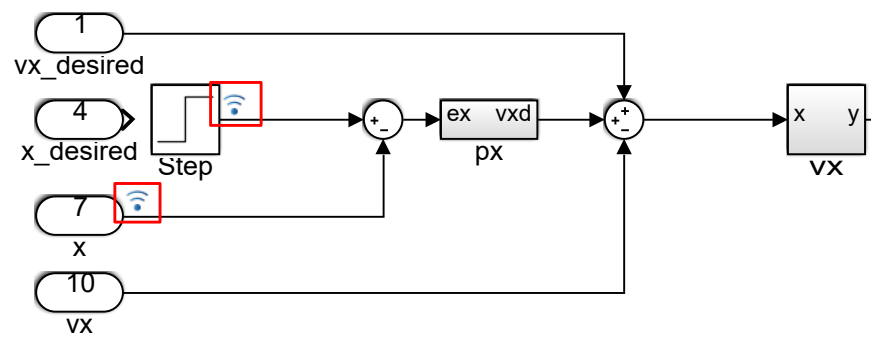


图 位置环调试模型



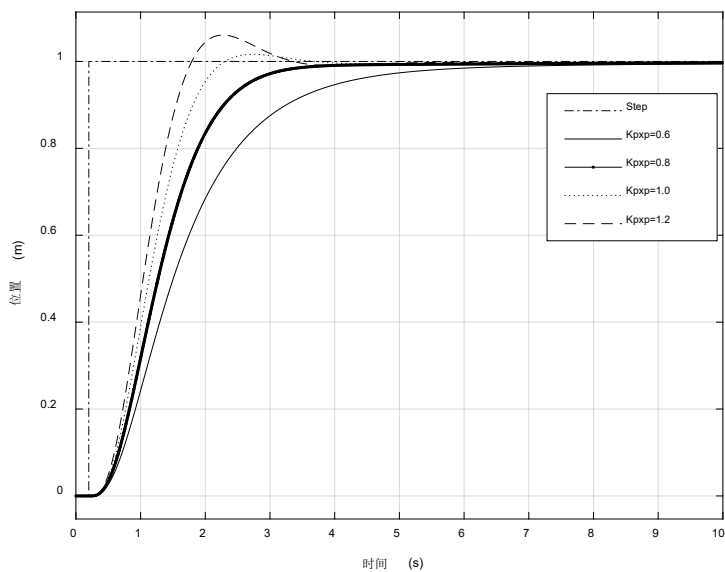


# 分析实验

## □ 实验步骤

### (3) 步骤三：位置环参数调节

在“Init\_control.m”由小增大位置环比例项系数，即“Kp<sub>xp</sub>”的值，在“Simulation Data Inspector”观察阶跃响应。得到如下波形如图，再进一步微调比例项系数得到最终的响应曲线。



$K_{p_{xp}}=1.0;$   
 $K_{v_{xp}}=2.5;$   
 $K_{v_{xi}}=0.4;$   
 $K_{v_{xd}}=0.01;$

图 调节位置环比例项系数得到的阶跃曲线

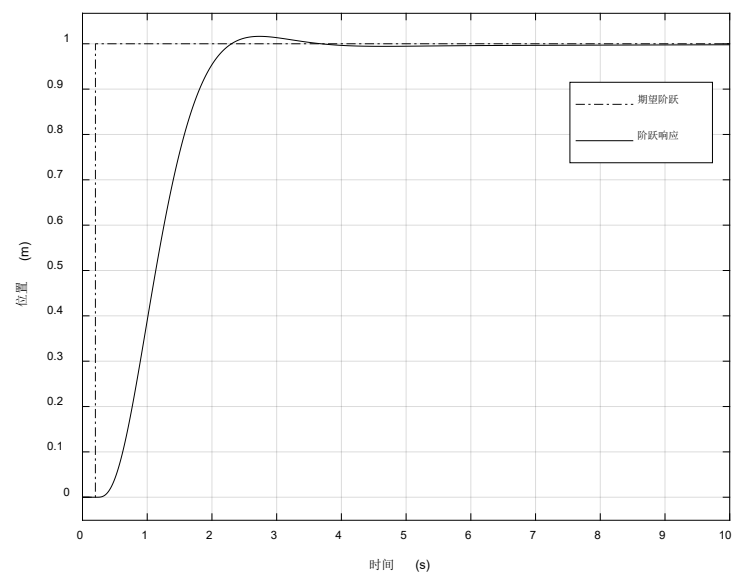


图 最终参数下的阶跃响应曲线





# 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“x\_desired”输入线设为“Open-loop Input”，“x”设置为“Open-loop Output”得到Bode图如右图。

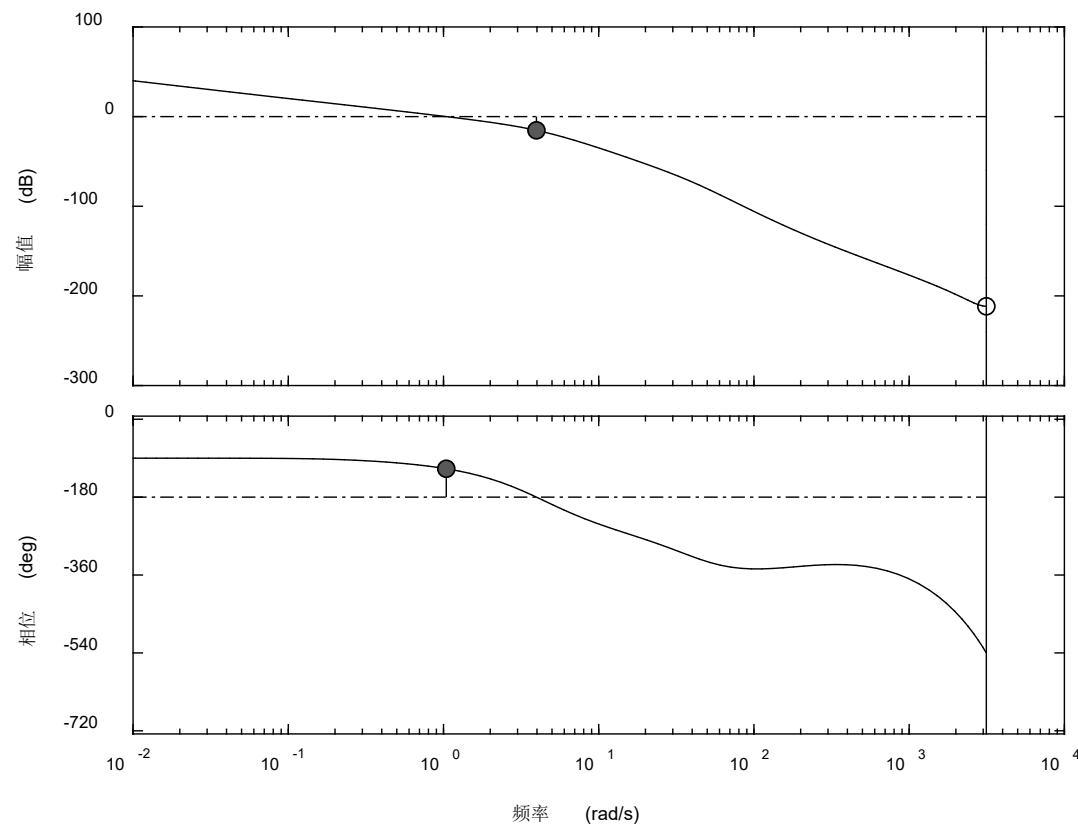


图. 沿  $O_e x_e$  轴位置控制系统开环Bode图





# 分析实验

---

## □ 注意事项

(1) 频率响应曲线的前提是四旋翼处于平衡态，即四旋翼保持悬停，所以要得到正确的频率响应曲线要保持四旋翼处于平衡态。

(2) 理清系统的输入输出，正确选择输入输出点，测试开环系统时选择输出点类型为“Open-loop output”，测试闭环时使用“Output Measurement”，具体介绍可以参考官方文档<https://ww2.mathworks.cn/help/slcontrol/ug/specify-portion-of-model-to-linearize-in-simulink-model.html>。





# 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: Matlab2017b, 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“e6.3”“e6.4” (<https://flyeval.com/course>)。

### ■ 目标

- 1) 建立位置控制通道的传递函数模型, 使用MATLAB “ControlSystemDesigner” 设计校正控制器, 使得加入校正环节后系统速度控制环阶跃响应稳态误差  $e_{r_{ss}} \leq 0.01$ , 相位裕度  $> 75^\circ$ , 截止频率  $> 2.0 \text{ rad/s}$ 。位置控制环截止频率  $> 1 \text{ rad/s}$ , 相位裕度  $> 60^\circ$ 。
- 2) 使用自己设计的控制器进行软件在环仿真实验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x通道。简化后的模型如下图。

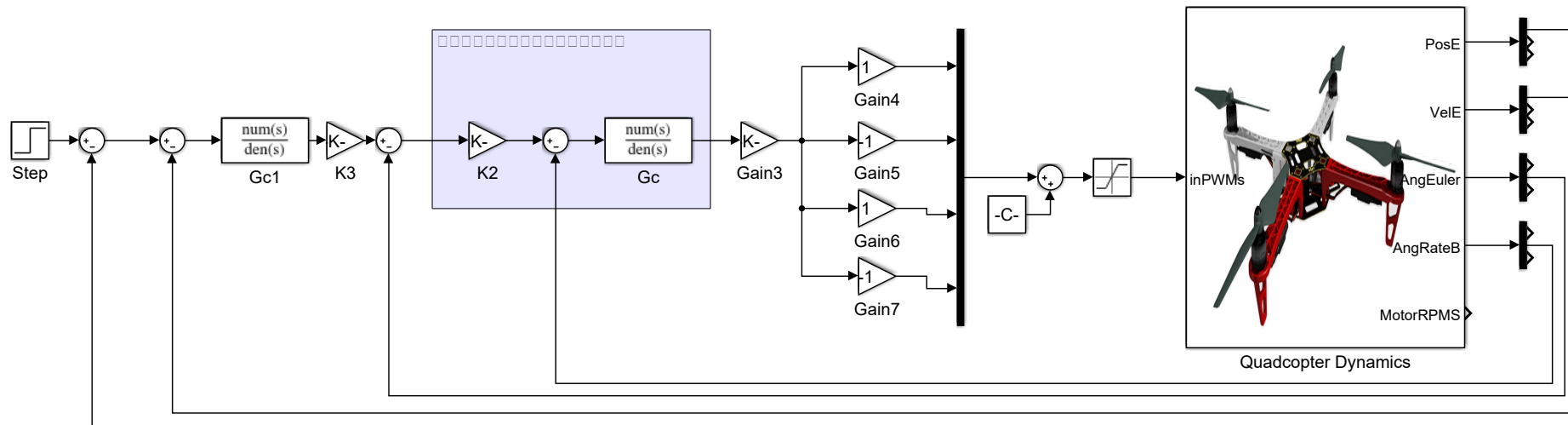


图. X轴位置控制系统简化模型





# 设计实验

## □ 实验设计

### (2) 步骤二：速度环分析

输入为期望速度，输出为实际速度。步骤与生成Bode图相同。设置输入输出点如图所示。

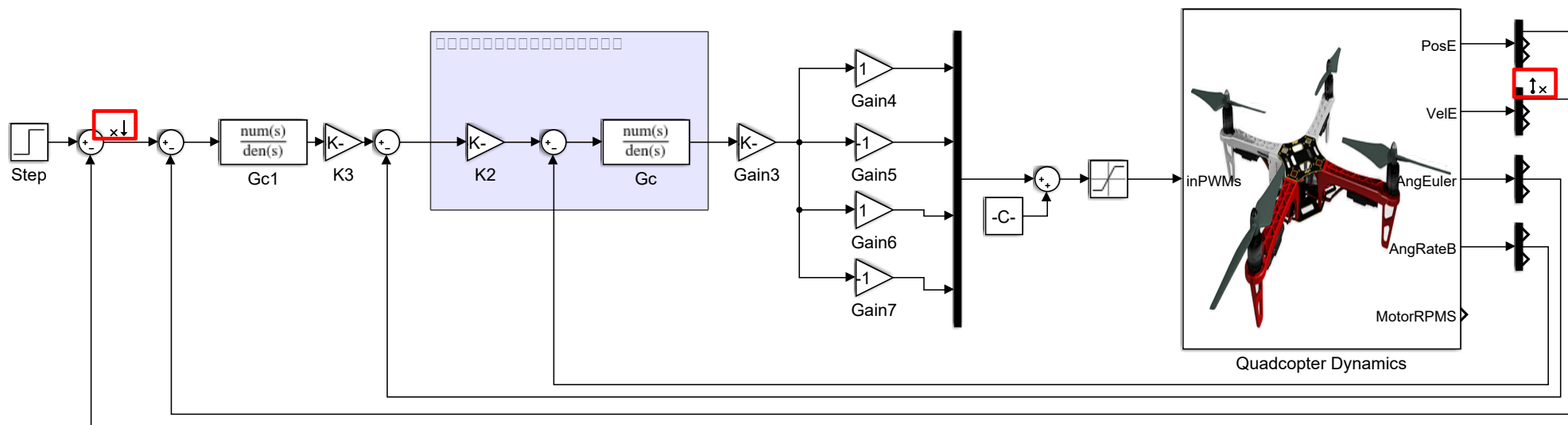


图. 设置输入输出点



# 设计实验

## □ 实验步骤

(3) 步骤三：得到X通道的传递函数

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。

$$\frac{3330.9(s+1.29)}{(s+5.214e-07)(s+1.253)(s+33.92)(s^2+14.87s+101.1)}$$



$$\frac{3331s^4 + 5.039e05s^3 + 2.563e07s^2 + 4.486e08s + 5.371e08}{s^8 + 200s^7 + 1.567e04s^6 + 6.045e05s^5 + 1.189e07s^4 + 1.154e08s^3 + 5.557e08s^2 + 5.371e08s + 280.1}$$

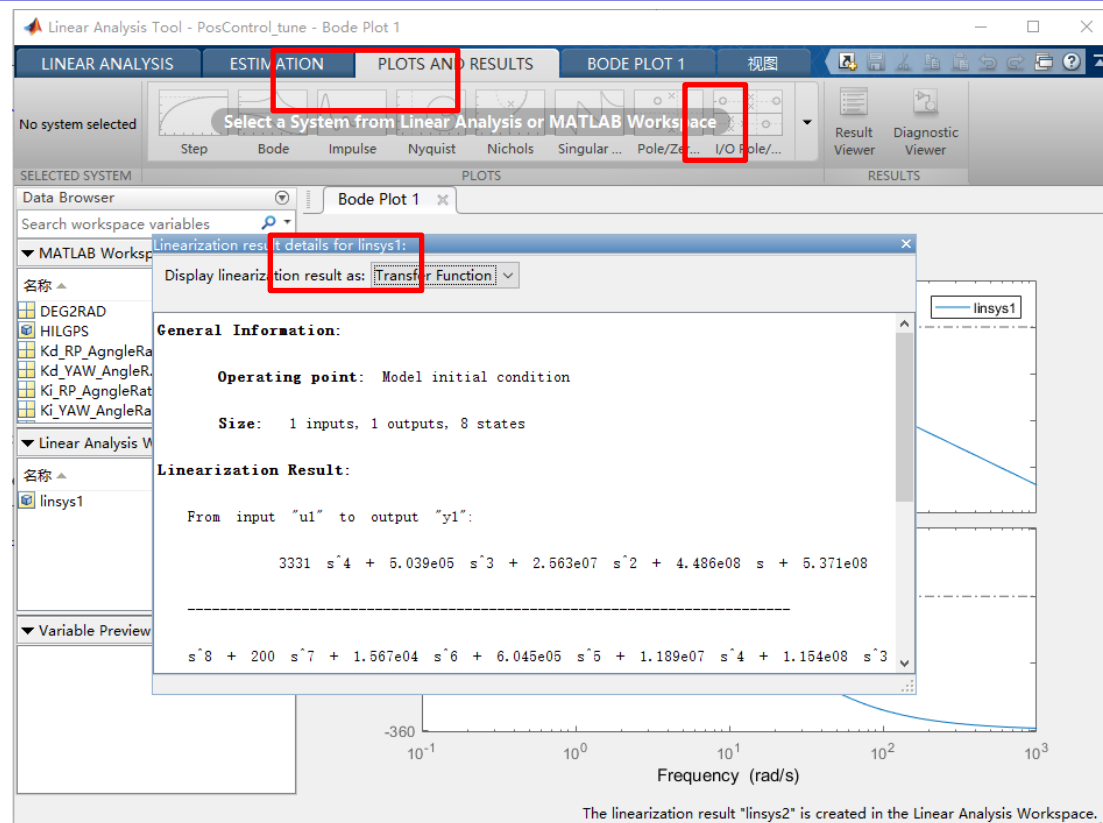


图. 得到传递函数





# 设计实验

## (4) 步骤四：建立环境

由传递函数建立如下m文件，运行即可使用Matlab 基于Bode图的控制系统设计。

1	<code>num=[3331 5.039e05 2.563e07 4.486e08 5.371e08];%分子</code>
2	<code>den=[1 200 1.567e04 6.045e05 1.189e07 ...</code>
3	<code>1.154e08 5.557e08 5.371e08 280.1];%分母</code>
4	<code>G=tf(num,den);%传递函数</code>
5	<code>controlSystemDesigner('bode',G);</code>

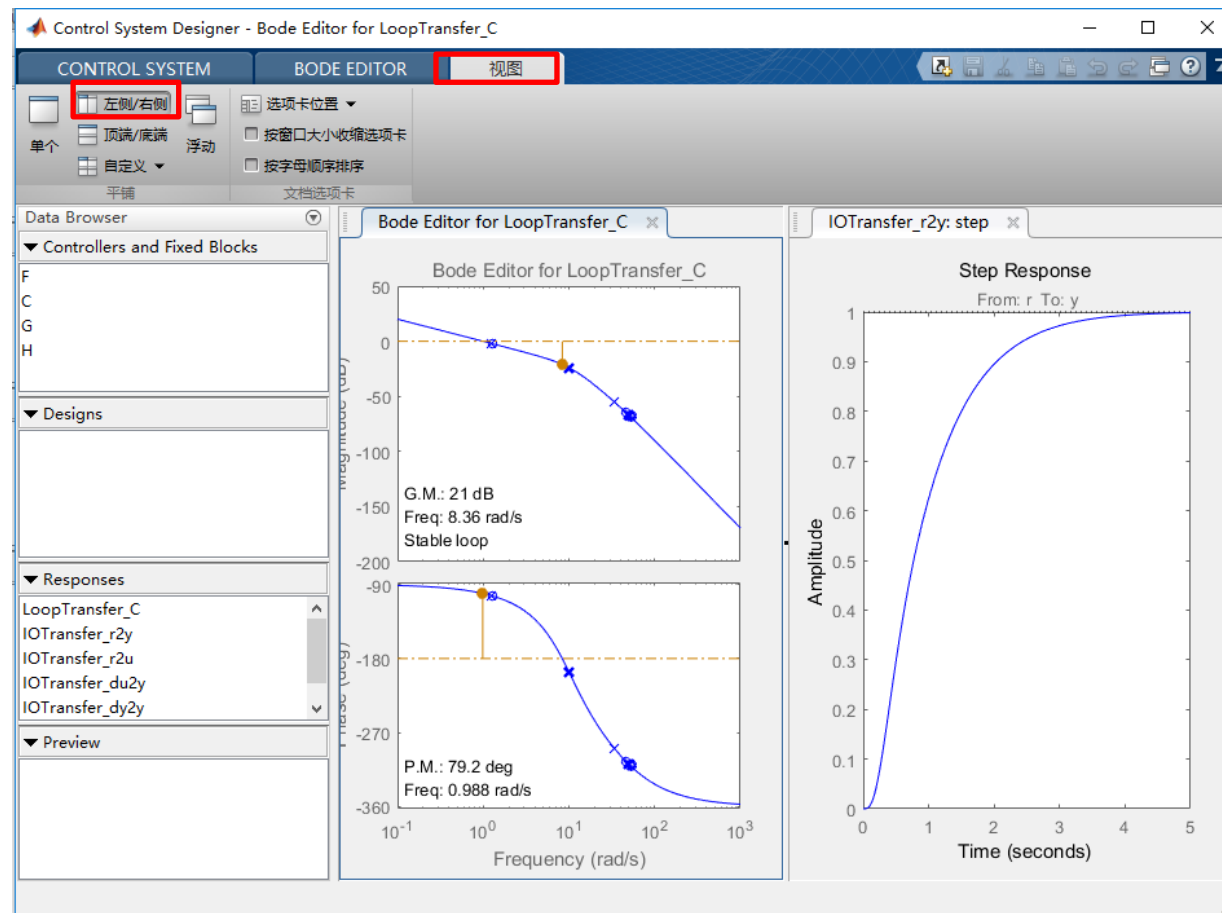


图. 基于Bode图的控制系统设计



# 设计实验

(5) 步骤五：使用工具箱校正  
响应较慢，向上拖动Bode图曲线增大开环增益。

增益增大，从阶跃响应曲线上看，响应时间变短，但是带来了超调。从Bode图上看，相位裕度为 $50.1^\circ$ ，较小。

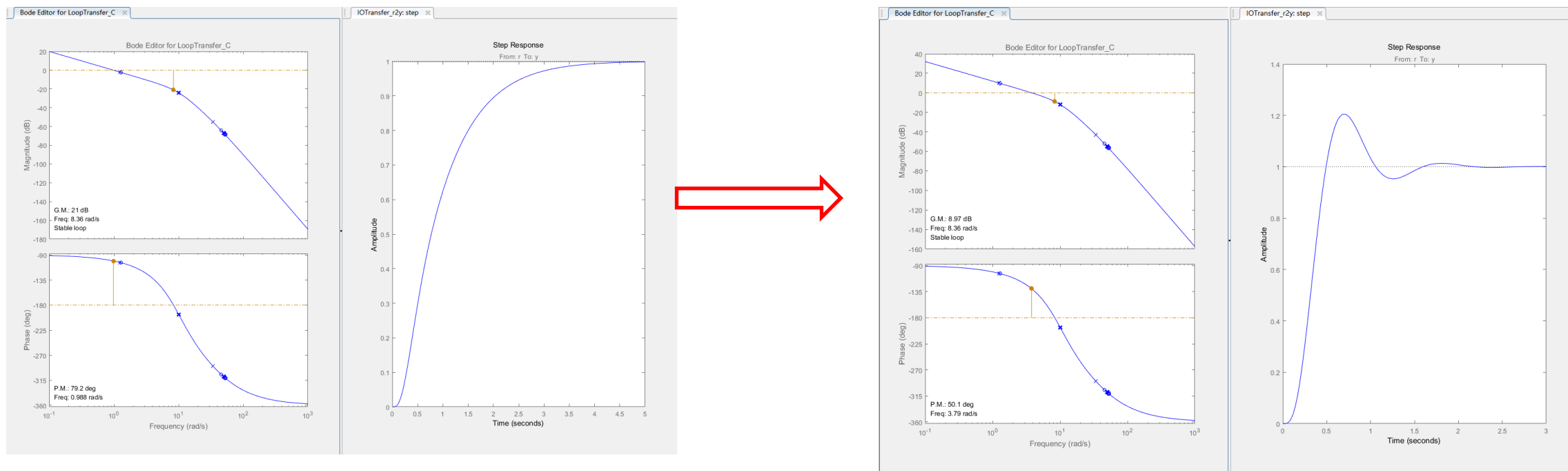


图. 加大增益对Bode图和阶跃响应的影响



# 设计实验

## (5) 步骤五：使用工具箱校正

增加一个超前校正环节，提高相位裕度，还能进一步提高截止频率，增加响应速度。

在Bode图中右键->add Pole/Zero->Lead

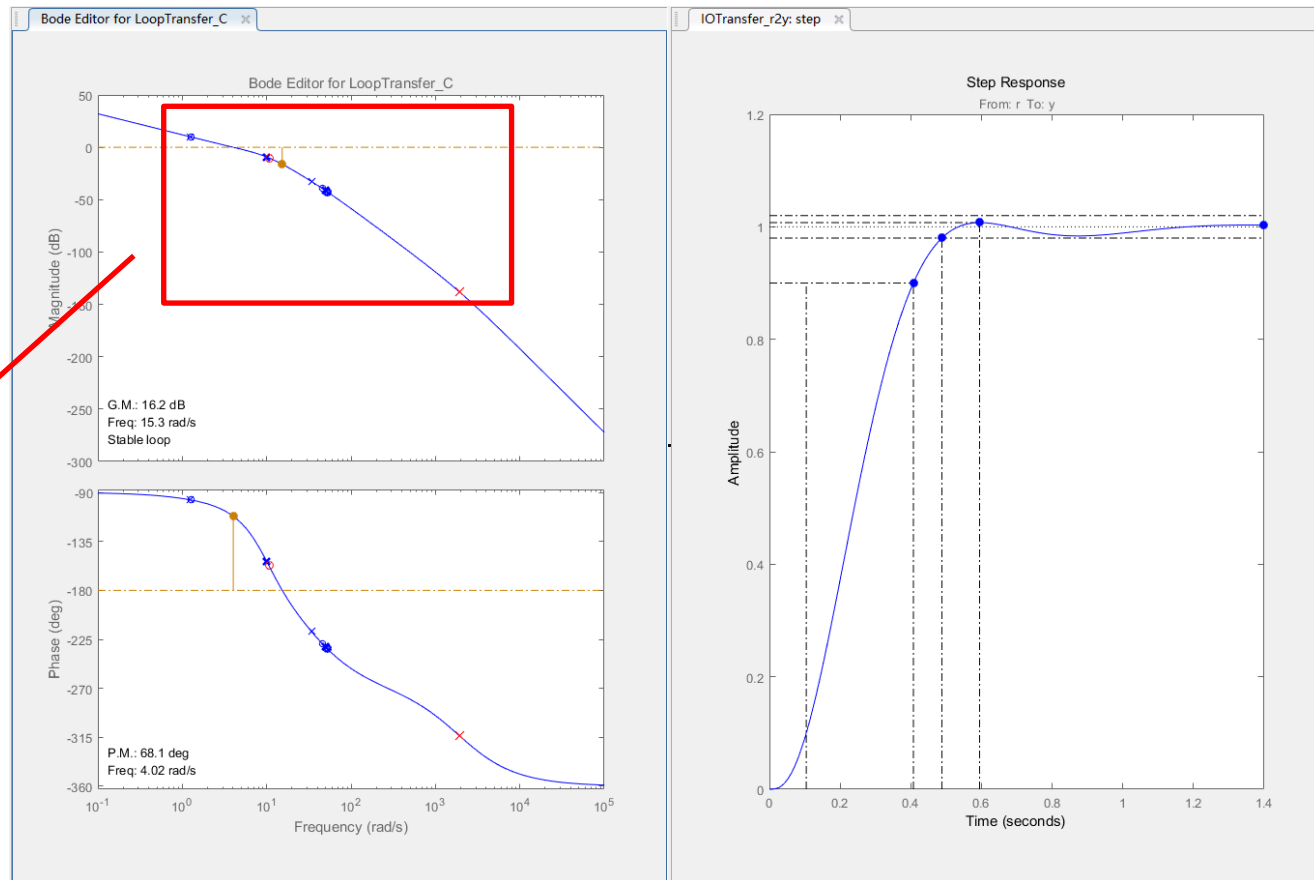
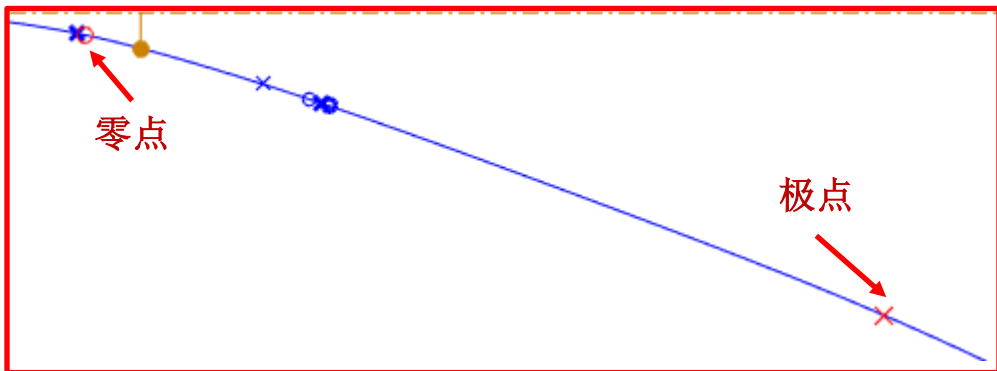


图. 加入超前环节





# 设计实验

## (5) 步骤五：使用工具箱校正

增加一个超前校正环节，提高相位裕度，还能进一步提高截止频率，增加响应速度。在Bode图中右键“add Pole/Zero” -> “Lead”，直接拖动零极点观测响应曲线，得到合适的校正环节。

$$G_c = \frac{4.0741(1 + 0.094s)}{1 + 0.01s}$$

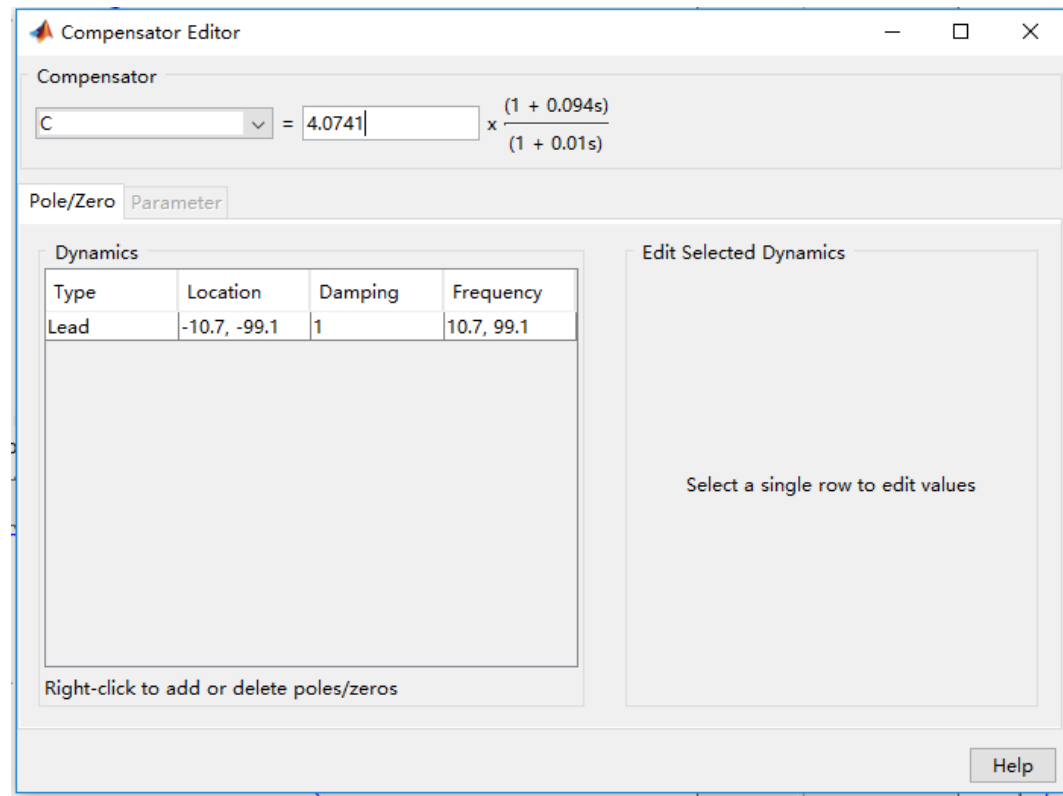


图. 得到校正传递函数



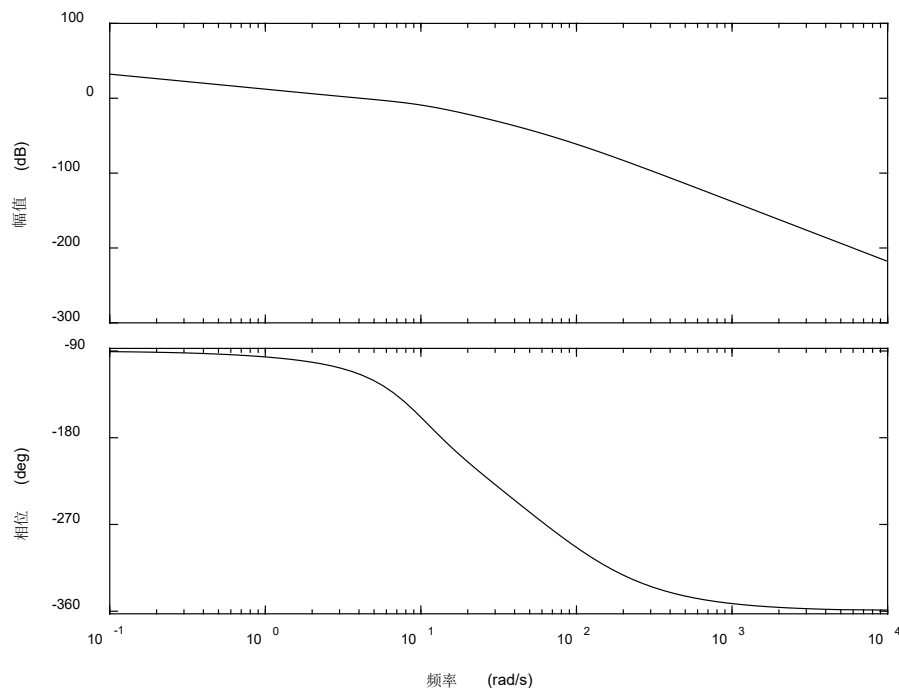




# 设计实验

## (6) 步骤六：对X通道位置环进行校正

加入步骤五得到的速度环校正器，得到位置环的Bode图：



相位裕度 $75.8^\circ$ ，截止频率0.99. 基本满足要求。

可略微增加增益，提高截止频率。

图. 位置环Bode图





# 设计实验

## □ 仿真实验步骤

### (1) 步骤一：将连续校正环节离散化

前述设计的校正环节是时域连续的环节，加入该模型应将其变为时域离散的环节。

使用c2d函数将s域的传递函数变为z域：

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

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

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0.01s。替换后的传递函数为：

$$G_c = \frac{2.0452(1+0.15s)}{1+0.013s} \rightarrow G_c(z) = \frac{22.6z - 22.5}{z - 0.4634}$$



# 设计实验

## □ 仿真实验步骤

### (2) 步骤二：替换控制模型

将Simulink模型中的PID控制器替换成离散模块，如右图所示。

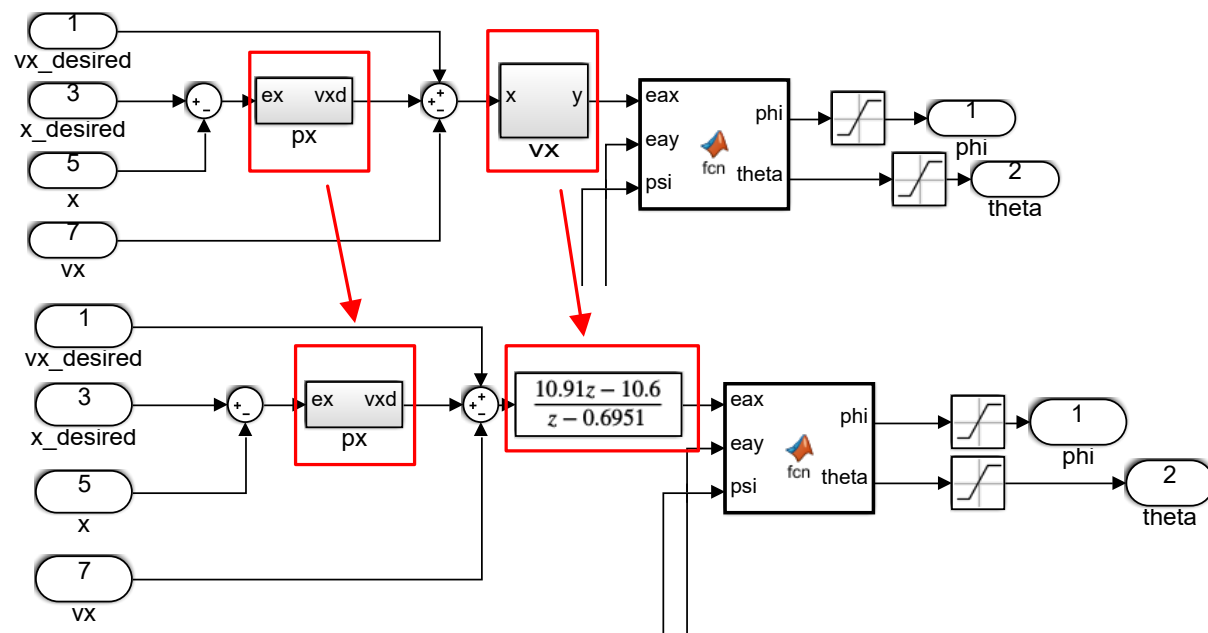


图. 替换原来的PID校正器





# 设计实验

## □ 仿真实验步骤

### (3) 进行硬件在环仿真

多旋翼可以实现基本的直线飞行和悬停。故可认为校正环节设计可以满足要求。

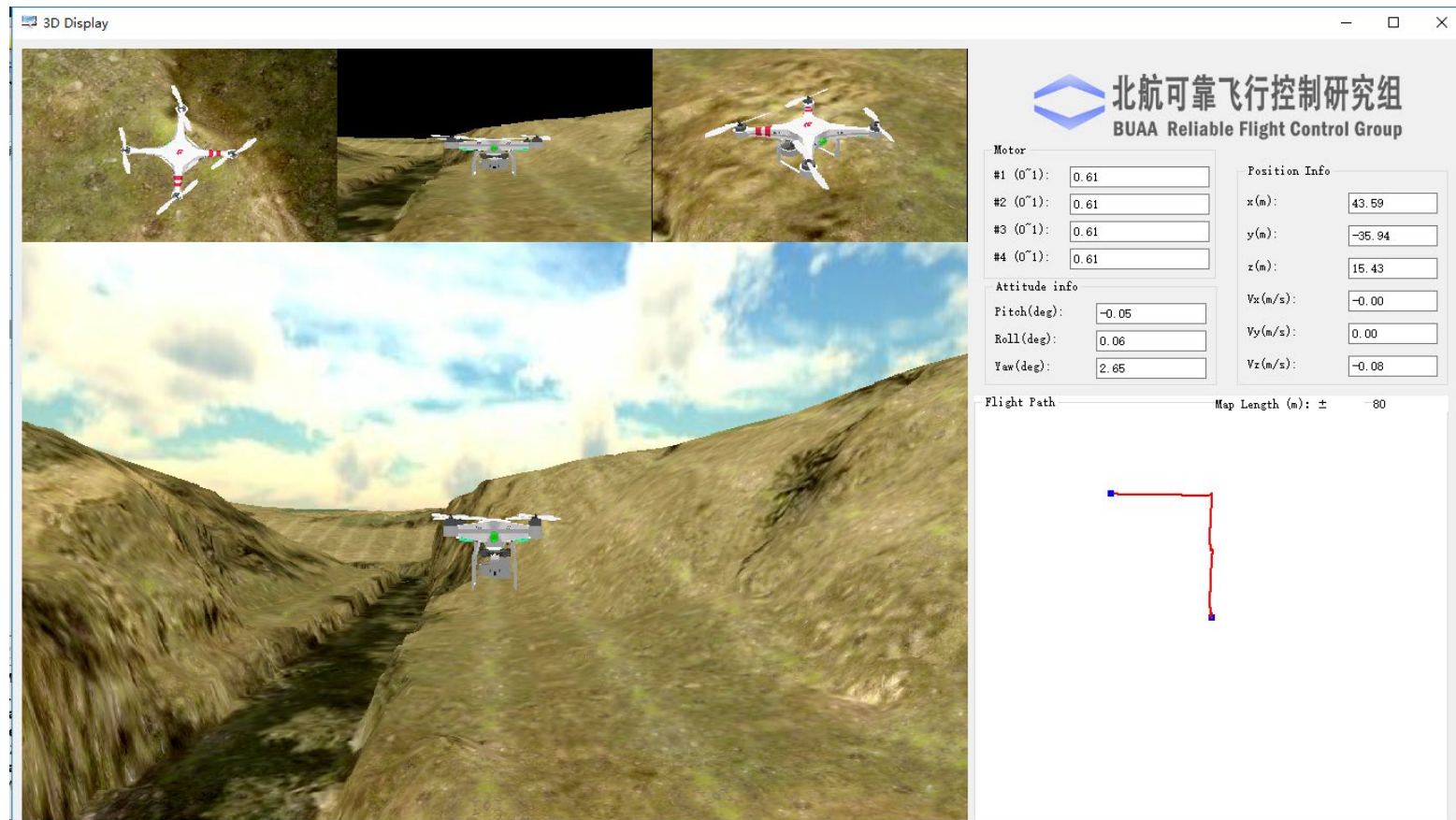


图 硬件在环仿真



# 设计实验

## □ 实飞实验步骤

### (1) 步骤一：四旋翼设置

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。添加自定义logger数据详细步骤见实验五。

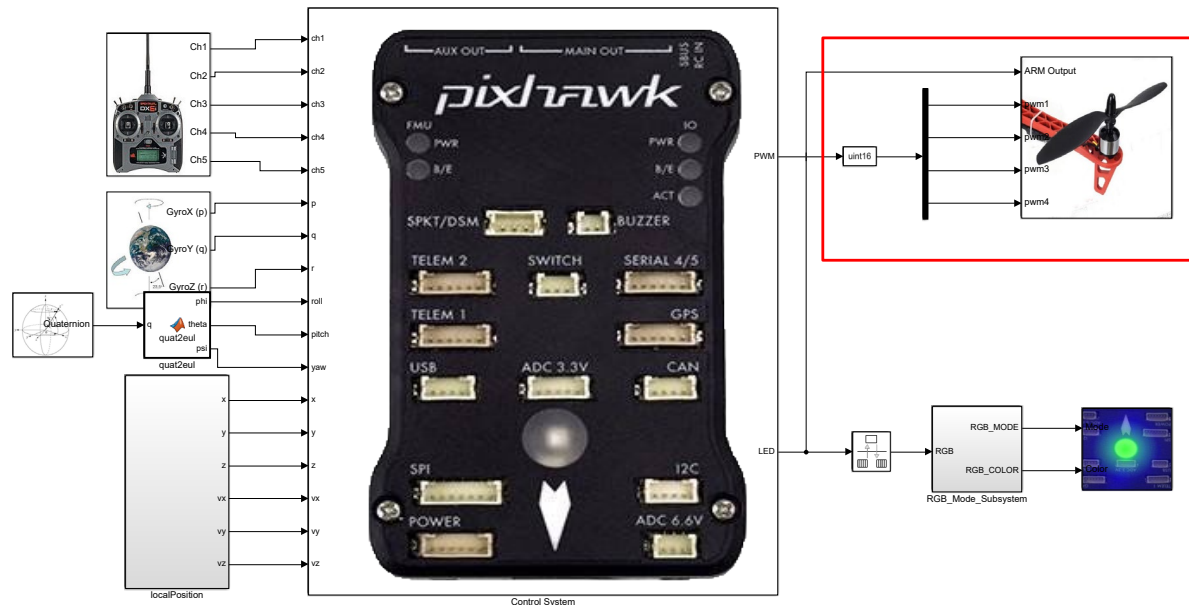


图. 实飞模型, Simulink 模型 “PosControl\_FLY.slx”



# 实飞实验

## □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：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，如右图所示。另外应在空旷场地上进行实验，以确保GPS信号良好。



图. 室外实飞







# 设计实验

## □ 实飞实验步骤

### (5) 步骤五：飞行数据分析

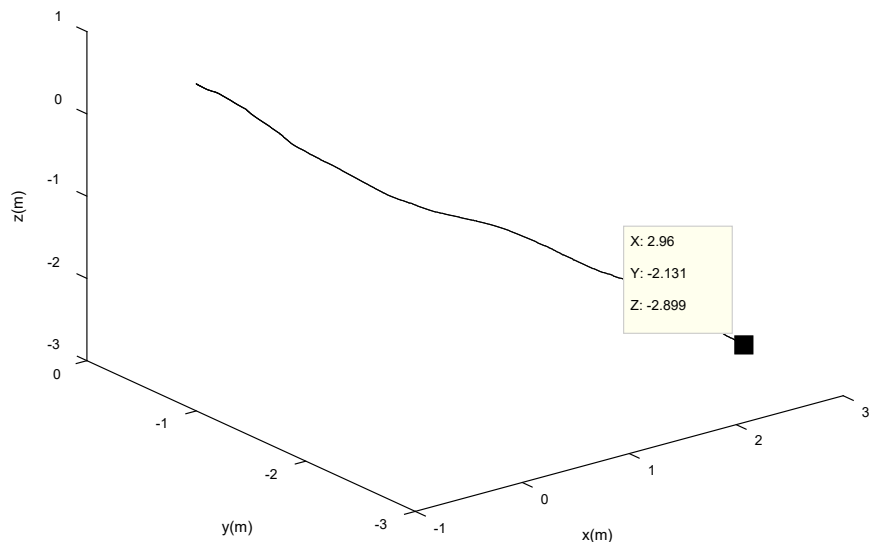


图. 3维位置

左图四旋翼从0点到达指定位置，然后保持不变，四旋翼能较好地实现定点。

右图是速度控制响应曲线，四旋翼按照指定速度飞行，速度响应的快速性较好。

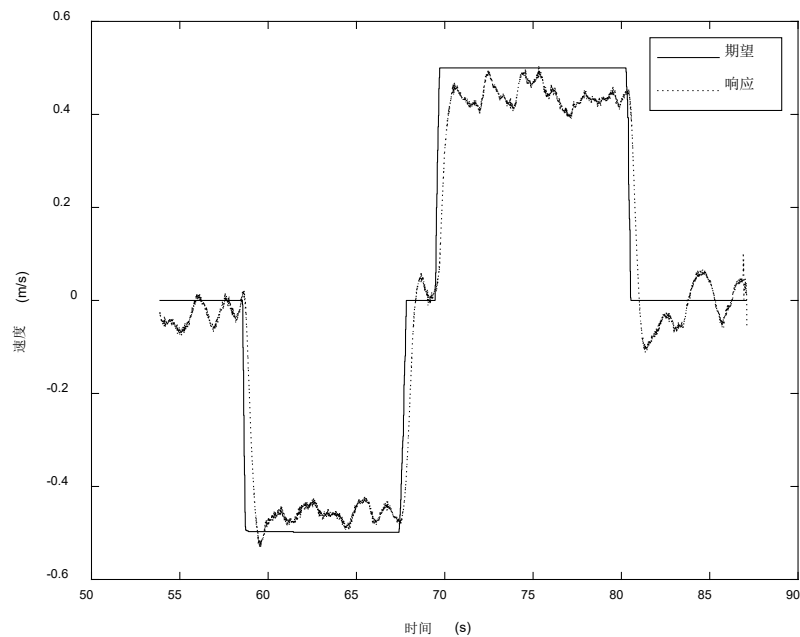


图.  $o_y y_e$ 轴速度响应





## 5. 总结

---

- (1) 以多旋翼的位置模型为依据，建立了常见的PID控制方法，并在MATLAB/Simulink中完成位置控制器的设计，并在FlightGear中显示仿真效果。
- (2) 使用Simulink中的PSP工具箱生成的代码并将其下载到Pixhawk自驾仪中进行硬件在环仿真实验。
- (3) 调节PID控制器的参数，尝试得到一组满意的参数，并使用MATLAB系统分析工具得到整个开环系统的Bode图，查看相应闭环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位置环和速度环实施控制，并达到设计指标。完成设计后，连接硬件进行在环仿真和实飞实验验证了设计的效果。



---

# 谢谢！



北航可靠飞行控制研究组  
BUAA Reliable Flight Control Group