

多旋翼飞行器远程控制实践

第7章 跟随控制器设计实验

全 权

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
BEIHANG UNIVERSITY



可靠飞行控制研究组

RELIABLE FLIGHT CONTROL GROUP

大纲

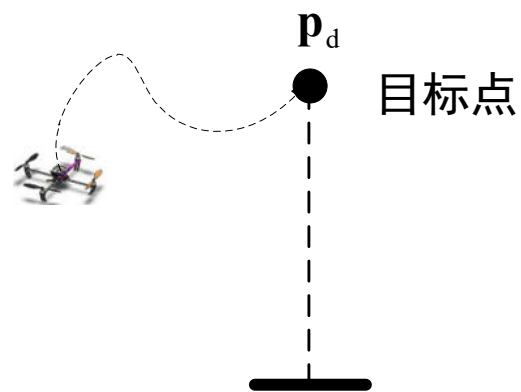


- 1. 基本原理**
- 2. 基础实验**
- 3. 分析实验**
- 4. 设计实验**
- 5. 实飞实验**
- 6. 本章小结**

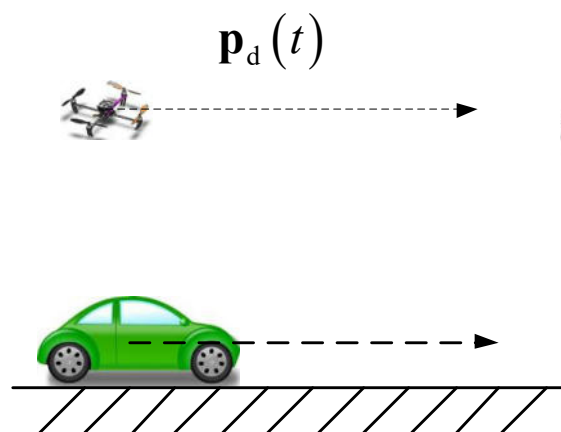
1. 实验原理

□ 分类

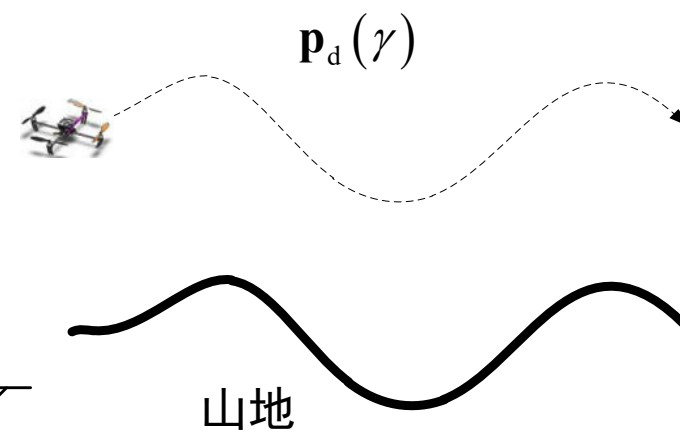
根据**给定目标轨迹** p_d 的不同，可将位置控制分为三类：定点控制、轨迹跟踪和路径跟随。



(a) 定点控制



(b) 轨迹跟踪



(c) 路径跟随

图. 位置控制三类形式



1. 实验原理

□ 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)。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. 实验原理

□ 人工势场法

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

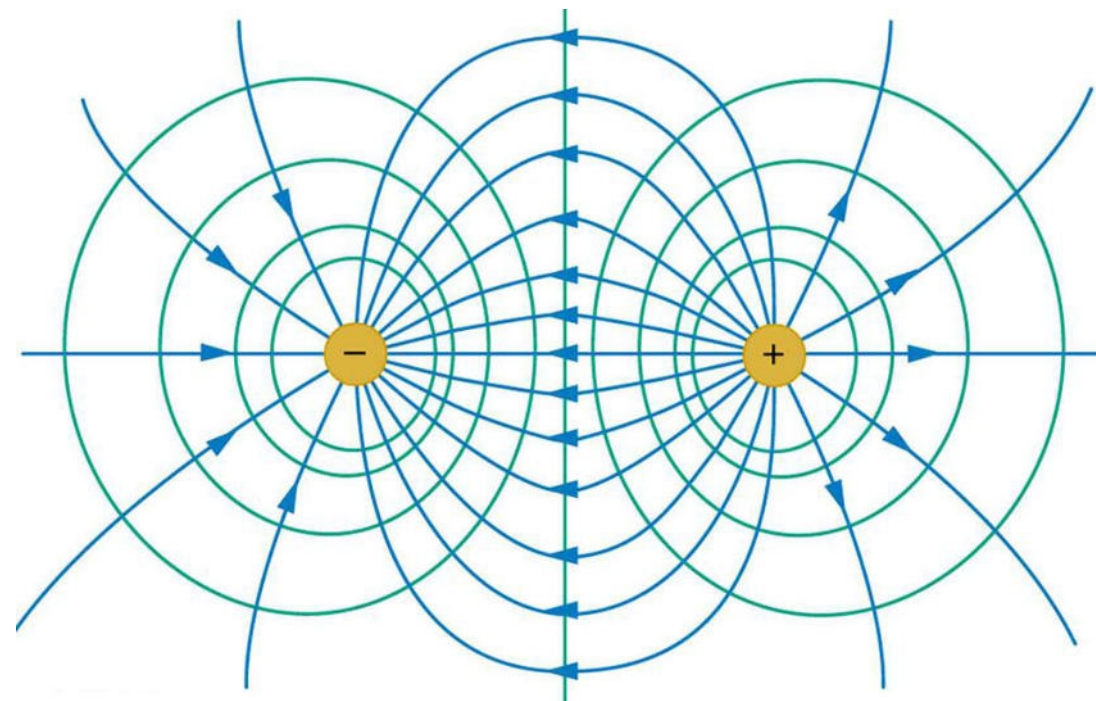


图. 势场示意图



1. 实验原理

□ 直线路径跟随

(1) 问题描述：令 $\mathbf{p} \in \mathbb{R}^2$ 表示多旋翼当前位置， $\mathbf{p}_{wp} \in \mathbb{R}^2$ 表示目标航路点， $\mathbf{p}_{wp,last} \in \mathbb{R}^2$ 表示上一个航路点。设计局部路径规划器实时地生成期望位置 $\mathbf{p}_d \in \mathbb{R}^2$ ，引导多旋翼沿着 $\mathbf{p}_d \in \mathbb{R}^2$ 和 $\mathbf{p}_{wp,last} \in \mathbb{R}^2$ 之间的直线路径飞行直到到达目标航路点 $\mathbf{p}_{wp} \in \mathbb{R}^2$ 。

(2) 算法设计

将多旋翼看作质点模型，其满足牛顿第二定律

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{u}\end{aligned}$$

1. 实验原理

□ 直线路径跟随

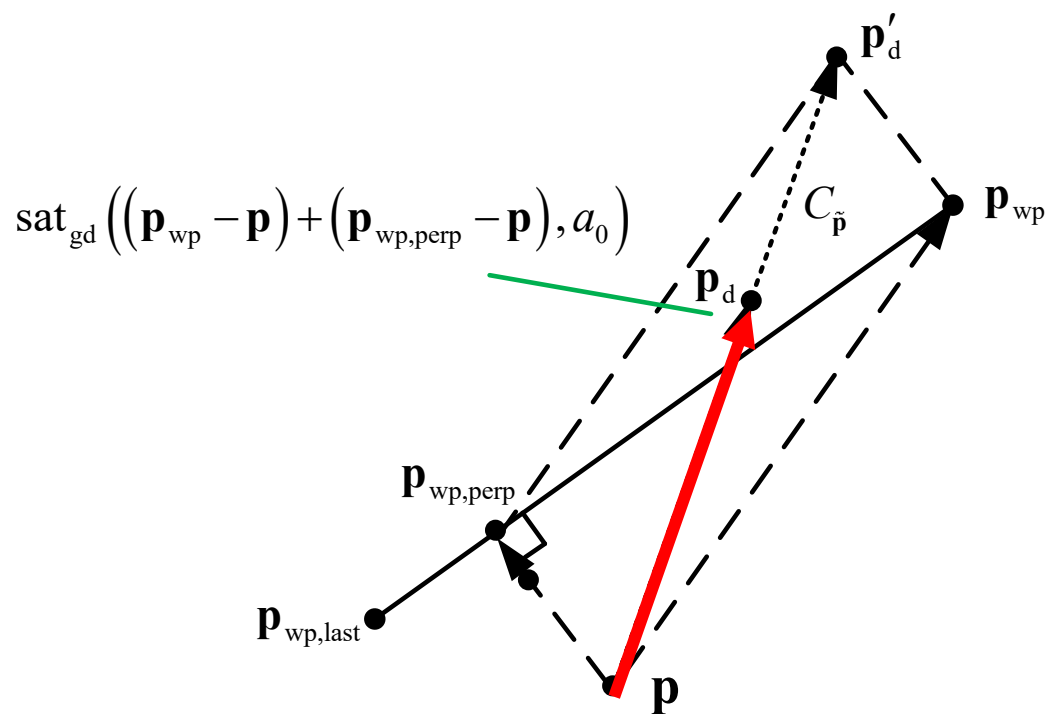


图. 直线路径跟随几何关系

垂足可以表示为

$$\mathbf{p}_{wp,perp} = \mathbf{p}_{wp} + (\mathbf{p}_{wp,last} - \mathbf{p}_{wp}) \frac{(\mathbf{p} - \mathbf{p}_{wp})^T (\mathbf{p}_{wp,last} - \mathbf{p}_{wp})}{\|\mathbf{p}_{wp} - \mathbf{p}_{wp,last}\|^2}$$

那么

$$\mathbf{p} - \mathbf{p}_{wp,perp} = \mathbf{A}(\mathbf{p} - \mathbf{p}_{wp})$$

其中

$$\mathbf{A} = \mathbf{I}_3 - \frac{(\mathbf{p}_{wp,last} - \mathbf{p}_{wp})(\mathbf{p}_{wp,last} - \mathbf{p}_{wp})^T}{\|\mathbf{p}_{wp} - \mathbf{p}_{wp,last}\|^2}$$

方程 $\mathbf{A}(\mathbf{p} - \mathbf{p}_{wp}) = 0$ 表示 $\mathbf{p} \in \mathbb{R}^2$ 位于直线航路上。

1. 实验原理

□ 直线路径跟随

(2) 算法设计

定义 $\tilde{\mathbf{p}}_{wp} = \mathbf{p} - \mathbf{p}_{wp}$ ，那么路径跟随控制设计为

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

将虚拟控制量写成PD控制器的形式

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

其中

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

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

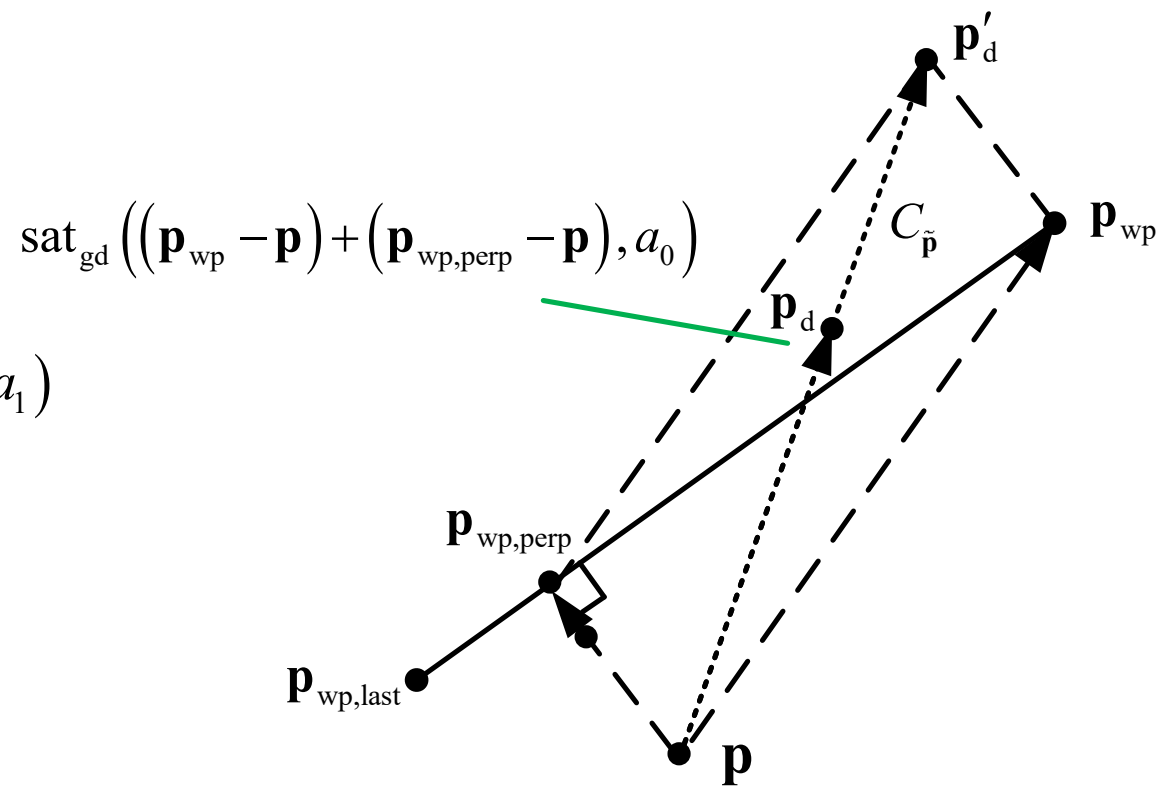


图. 直线路径跟随物理意义

大纲



1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 实飞实验
6. 本章小结



2. 基础实验-实验目标

■ 准备

- 软件：MATLAB R2017b 及以上版本，基于Simulink 的控制器设计与仿真平台和实验指导包“e4.1”，FlightGear
- 硬件：计算机

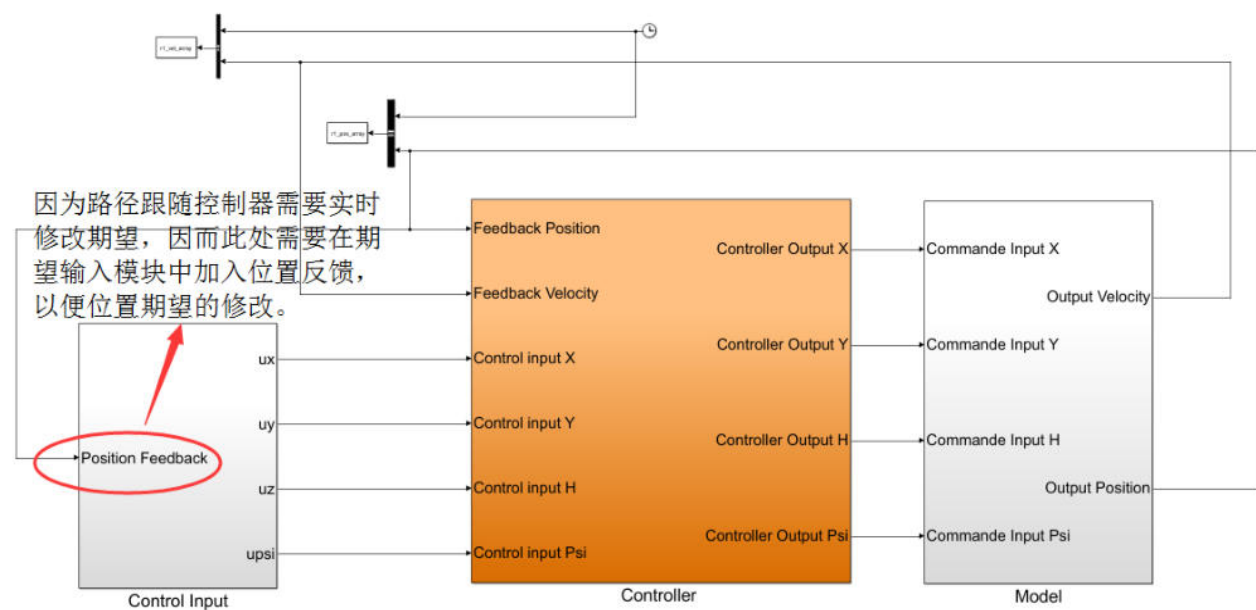
■ 目标

-给定多旋翼传递函数模型，该模型已经包含第六章设计的跟踪控制器。设计路径跟随控制器让传递函数模型输出进行直线路径跟随。直线初始点 $(5, -3)$ ，终点 $(5, 10)$ ，多旋翼初始位置随机，可以定为 $(0, 0)$ 。此外，偏航角通道需要始终保持为0，高度始终保持在2。

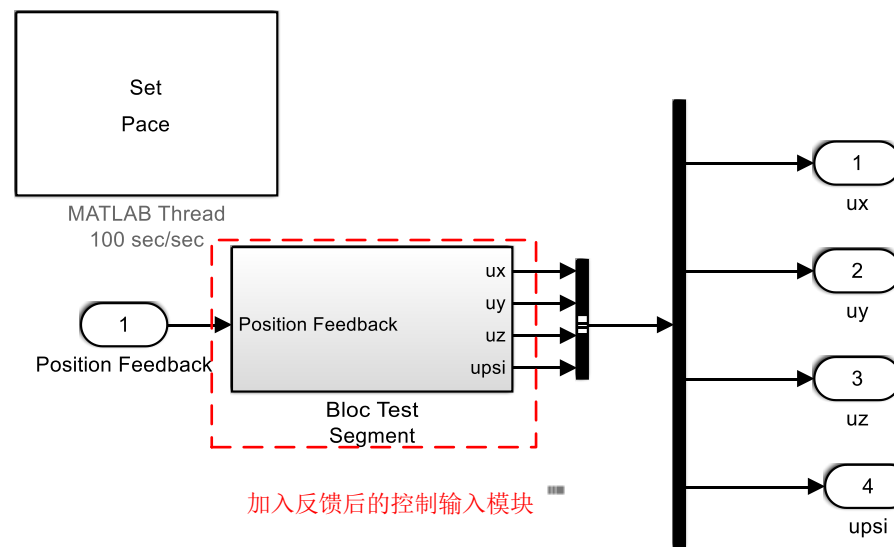
1. 学习并掌握路径跟随的理论基础及原理；
2. 实现多旋翼直线路径跟随控制。

2. 基础实验-实验模块介绍

(1) 打开Simulink 文件“e4\4.1\sim1.0\4_1_TF_TrajectoirePlanning_Segment.slx”，得到整体模型如左下图所示。打开控制输入模块，内部如右下图所示。



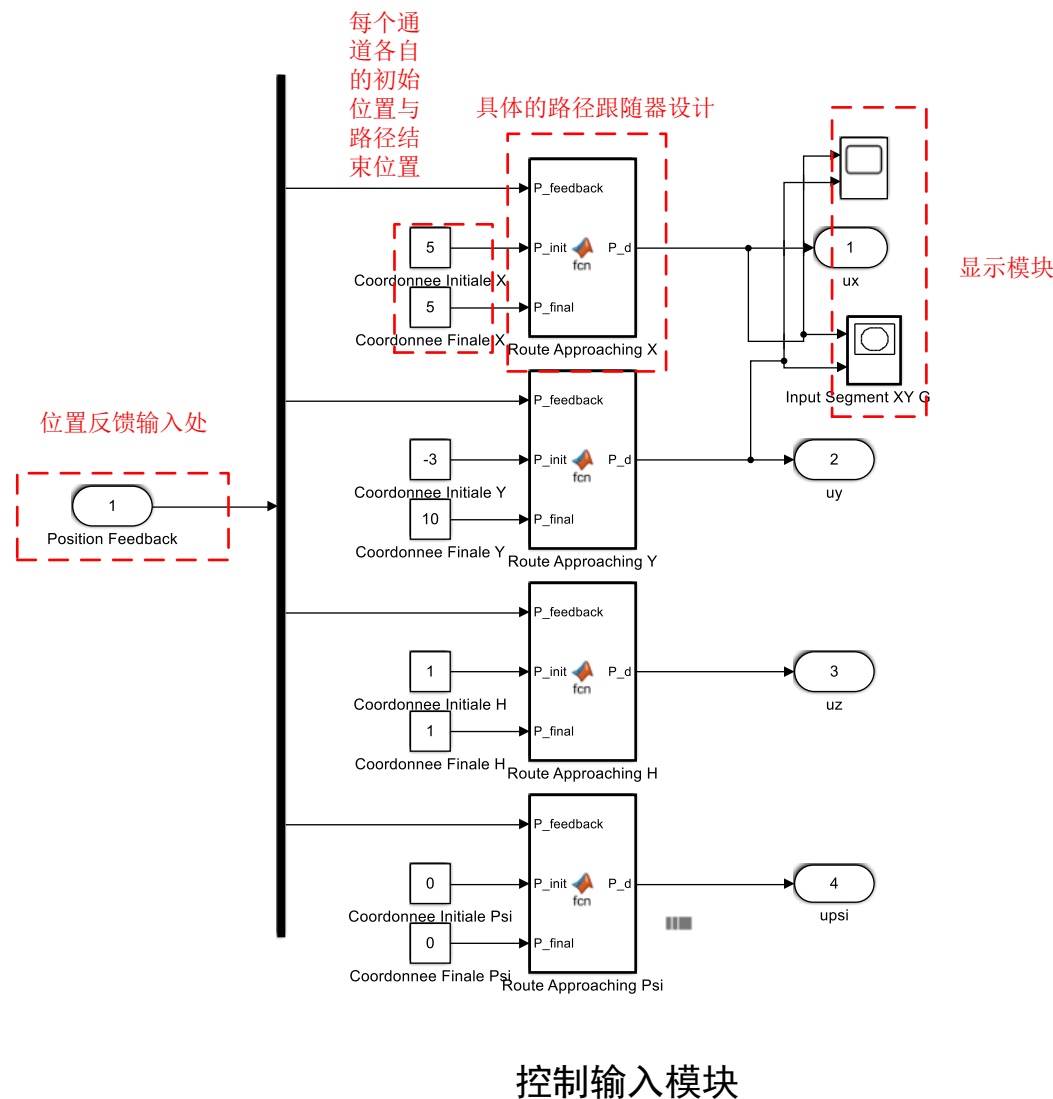
整体模块



控制输入模块

2. 基础实验-实验模块介绍

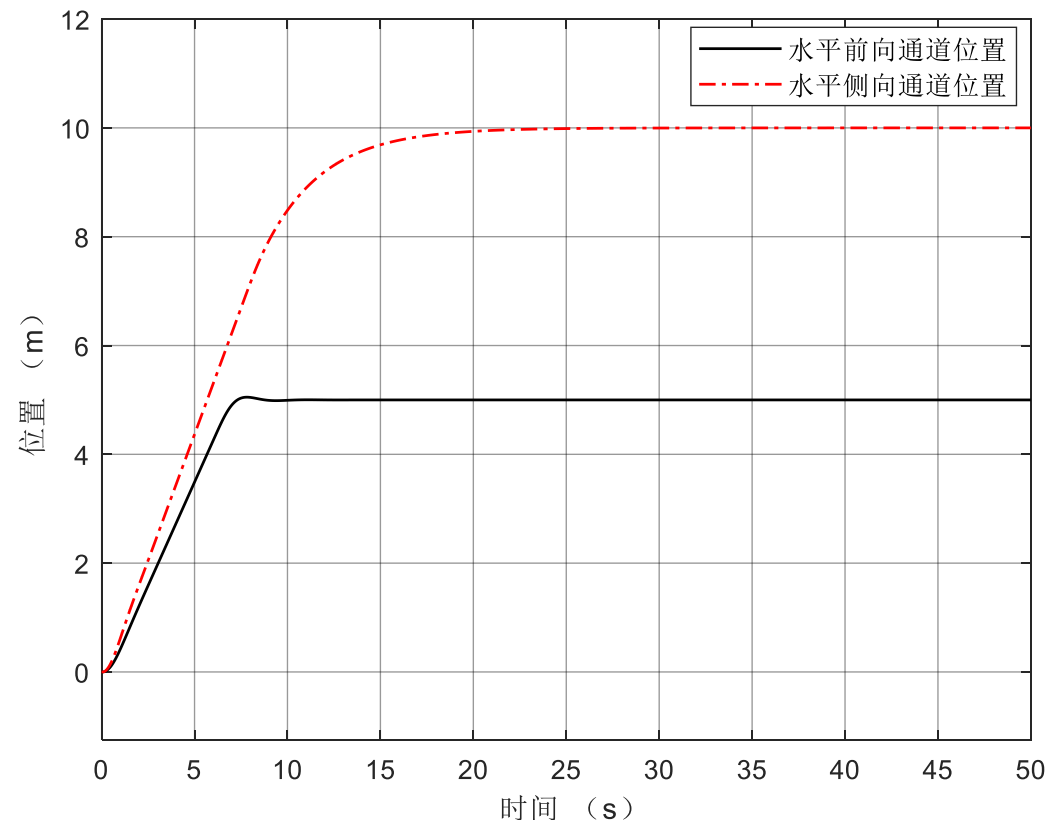
(2) 进一步打开路径跟随控制器模块，可以看到如右图所示的内部结构。对于四通道中的每一个通道，都设计了独立的路径跟随控制器。由于四个通道的设计几乎相同，这里仅以水平前向通道为例进行介绍。右图中加入了位置反馈，并且可以设置多旋翼需要跟随路径的初始坐标与终点坐标。设置完之后，将它们传入路径跟随控制器，产生位置期望，输出到跟踪控制器模块。此外这里还需要一些数据的保存与显示。



2. 基础实验-仿真1.0

(1) 运行文件

“e4\4.1\sim1.0\StartSimulation.m”，并打开相应的模型，按照实验目标要求设定初始与终点坐标进行参数初始化。单击Simulink“开始仿真”按钮进行仿真，水平前向通道以及水平侧向通道的路径跟随控制器响应时间曲线如右图所示。可以看出，多旋翼水平两通道轨迹都是从初始位置开始，然后逐渐逼近期望的轨迹，最终到达目标终点。



水平通道响应

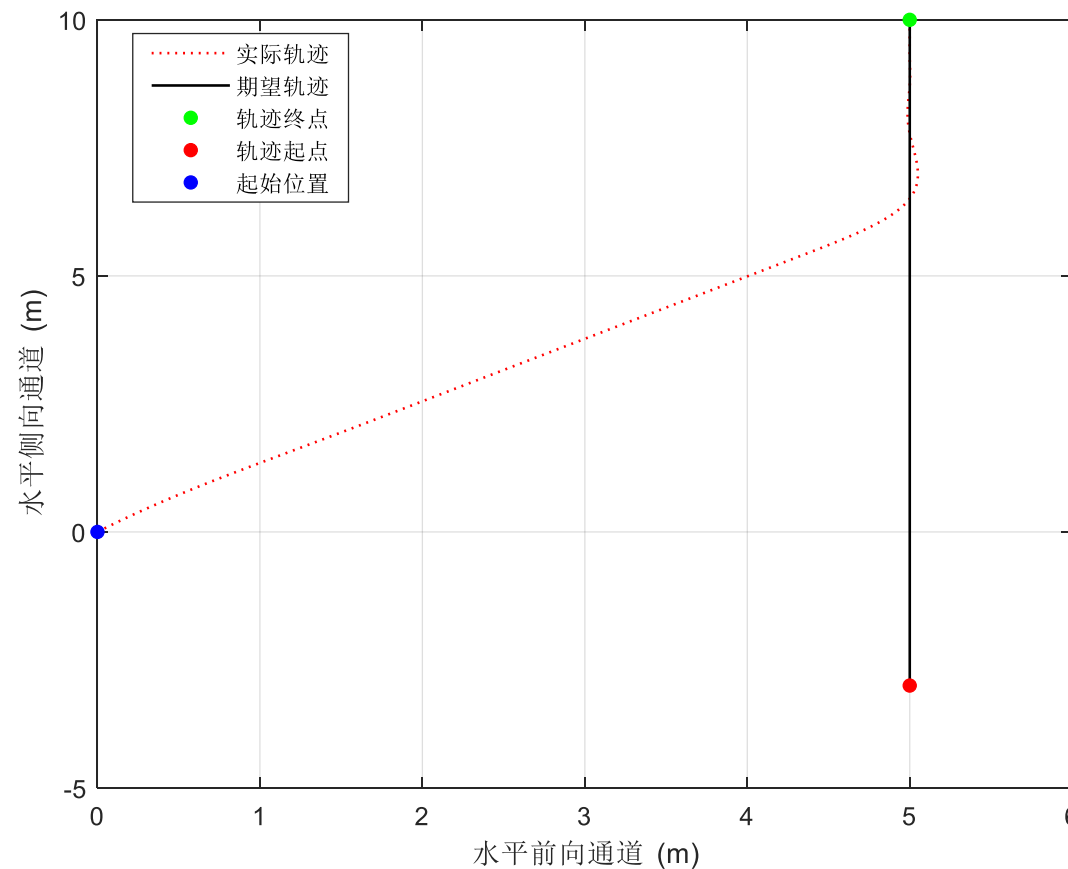
2. 基础实验-仿真1.0

(2) 此外，路径跟随控制器输出轨迹与期望轨迹

之间对比图如图所示。可以看到，从初始位置开始，

多旋翼非常迅速地逼近了期望轨迹，可以认为路径

跟随控制器的设计是可行的。



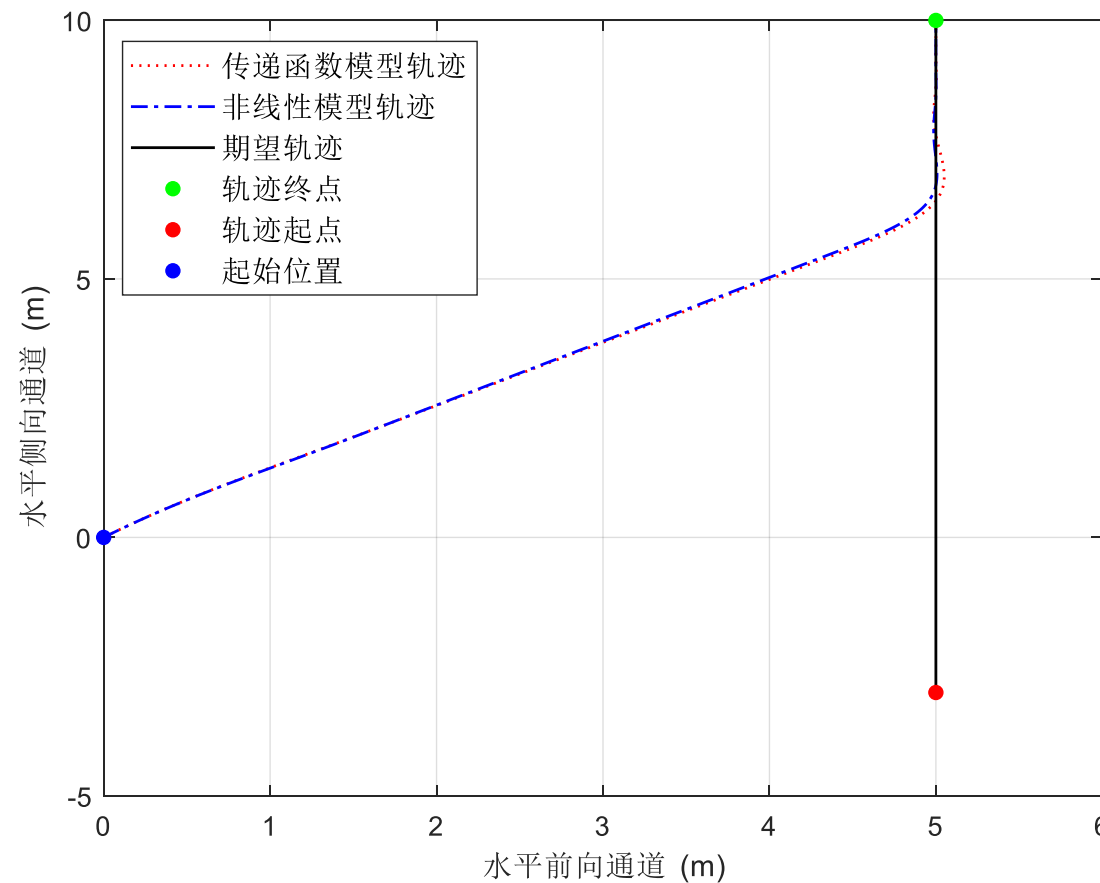
水平通道响应

第七章跟随控制器设计

基础实验：仿真1.0

2. 基础实验-仿真2.0

打开Simulink 文件“e4\4.1\sim2.0”，其中的文件与“e4\4.1\sim1.0”基本相同，区别在于这里替换多旋翼模型为非线性模型。采用与仿真1.0 完全相同的实验步骤，将两种模型的关键结果进行对比，对比结果如右图所示。从对比结果可以看出，我们采用系统辨识获得的传递函数模型进行跟踪控制器设计，可以产生与基于非线性模型设计非常接近的效果。同时，也可以在FlightGear 中查看仿真飞行效果。



水平通道响应

第七章跟随控制器设计

基础实验：仿真2.0

大纲



1. 基本原理
2. 基础实验
- 3. 分析实验**
4. 设计实验
5. 实飞实验
6. 本章小结



3. 分析实验-实验目标

■ 准备

- 软件：MATLAB R2017b 及以上版本基于Simulink 的控制器设计与仿真平台和实验指导包“e4.2”，FlightGear
- 硬件：计算机

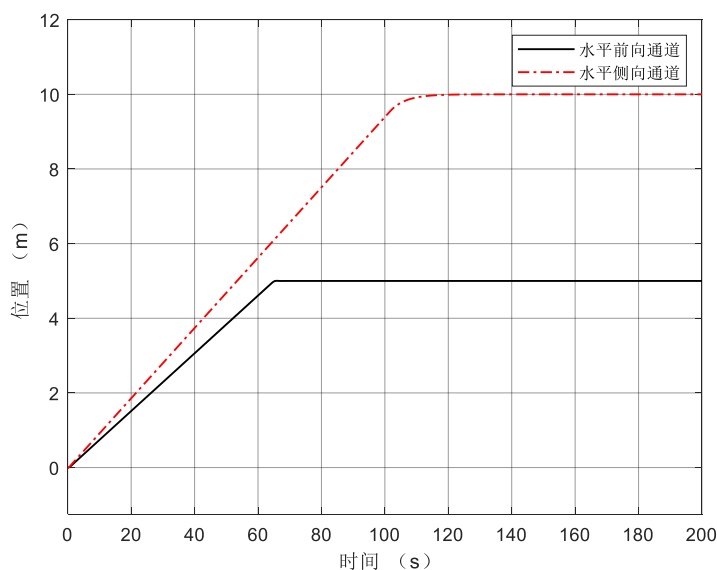
■ 目标

-对于直线路径跟随，修改参数 a_0 ， k_1 ，对比所获得的路径跟随轨迹的变化，总结规律并得出结论。

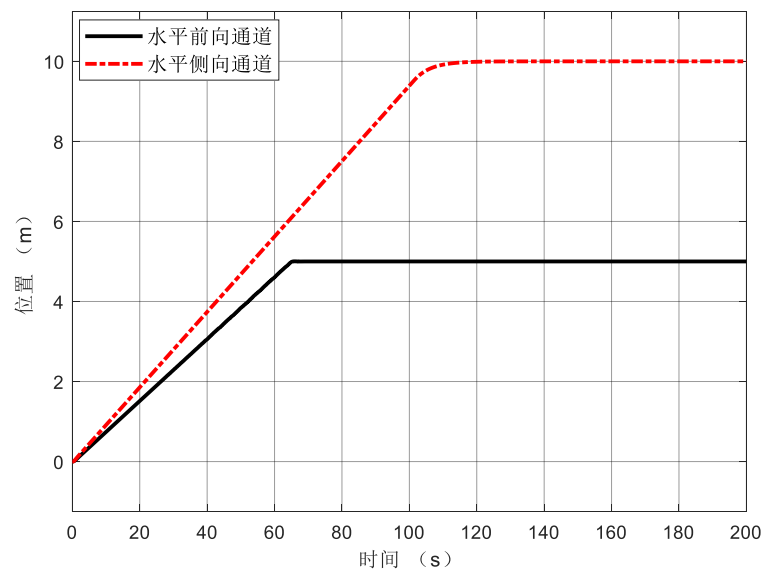
此处仍采用与基础实验相同的路径跟随控制器。直线初始点为 $(5, -3)$ ，到达点 $(5, 10)$ ，无人机初始位置随机，可以定为 $(0, 0)$ 。此外偏航角需要始终保持为0，高度始终保持在2。

3. 分析实验-仿真1.0

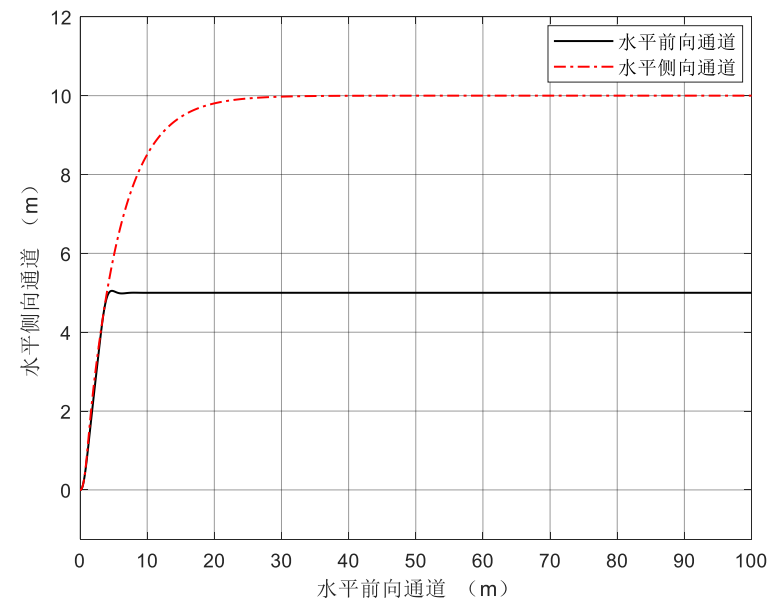
(1) 运行“e4\4.2\sim1.0\StartSimulation.m”进行参数初始化。选取如下参数进行实验，分别为 $k_0 = 0.2$ ， $k_1=1$ ， $a_0=0.1$ 、 0.5 和 2 。每次运行模型前改变 a_0 参数，得到水平前向通道和水平侧向通道的路径跟随控制器响应，如下图所示。



$a_0=0.1$ 路径跟随控制器输出时间曲线对比图



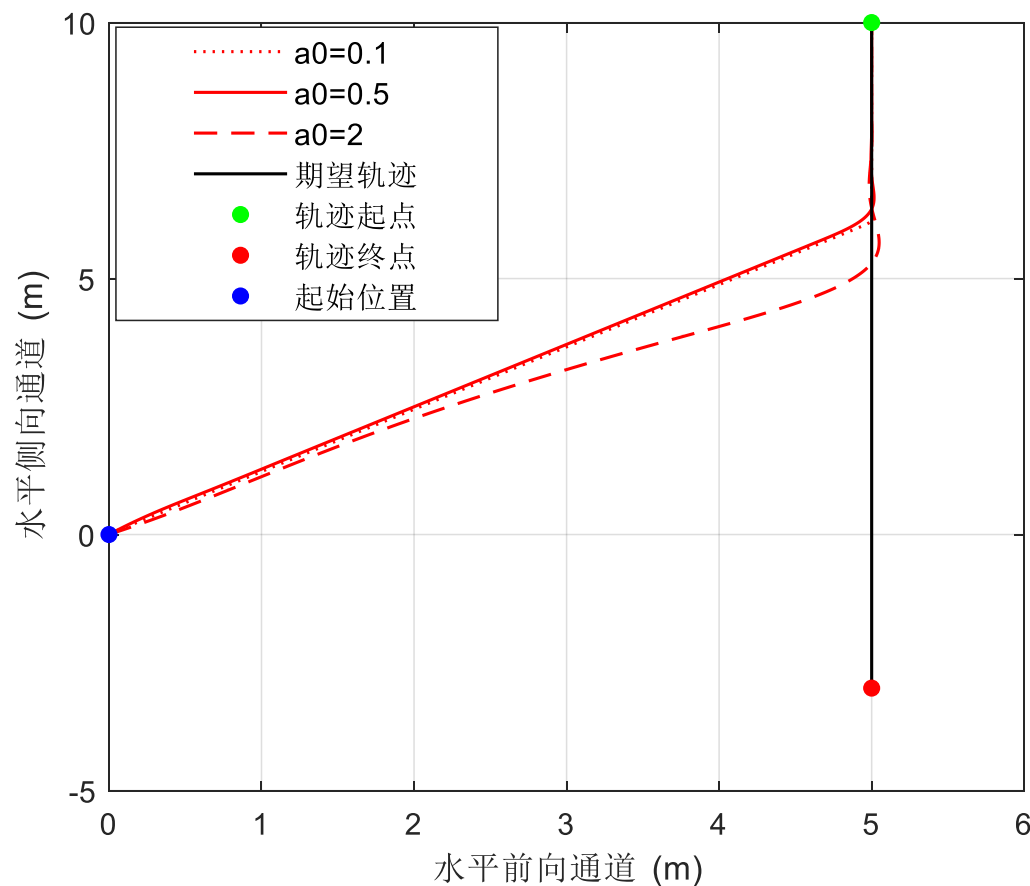
$a_0=0.5$ 路径跟随控制器输出时间曲线对比图



$a_0=2$ 路径跟随控制器输出时间曲线对比图

3. 分析实验-仿真1.0

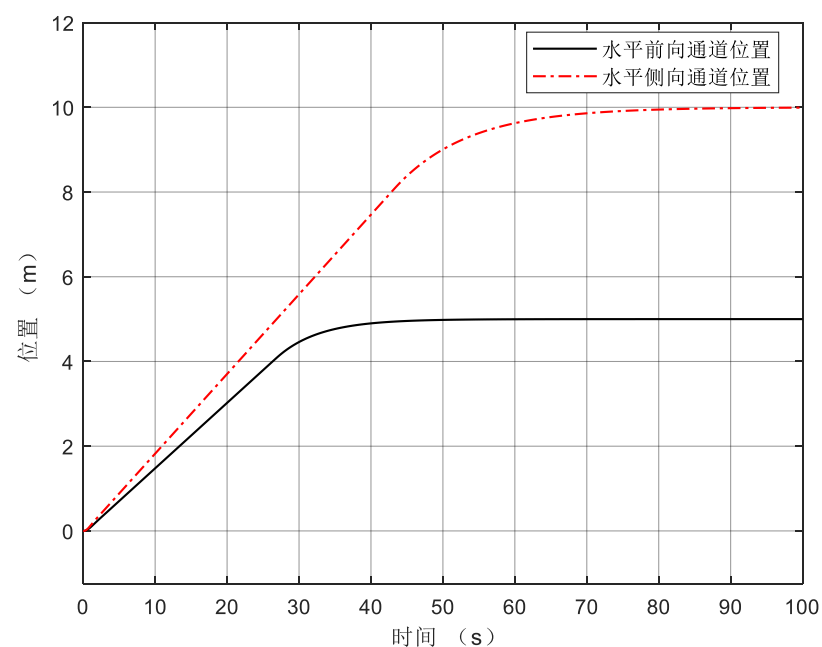
(2) 可以看出，在保持其它两个参数不变的情况下，改变饱和上限 a_0 的值，多旋翼贴合到期望轨迹上的速度越来越快。三种不同饱和上限的情况下，路径跟随控制器输出与期望轨迹对比图如右图所示。另外，随着饱和上限的增大，多旋翼靠近期望轨迹的斜率增加，甚至在 $a_0=2$ 时出现了一种类似“超调”的现象。



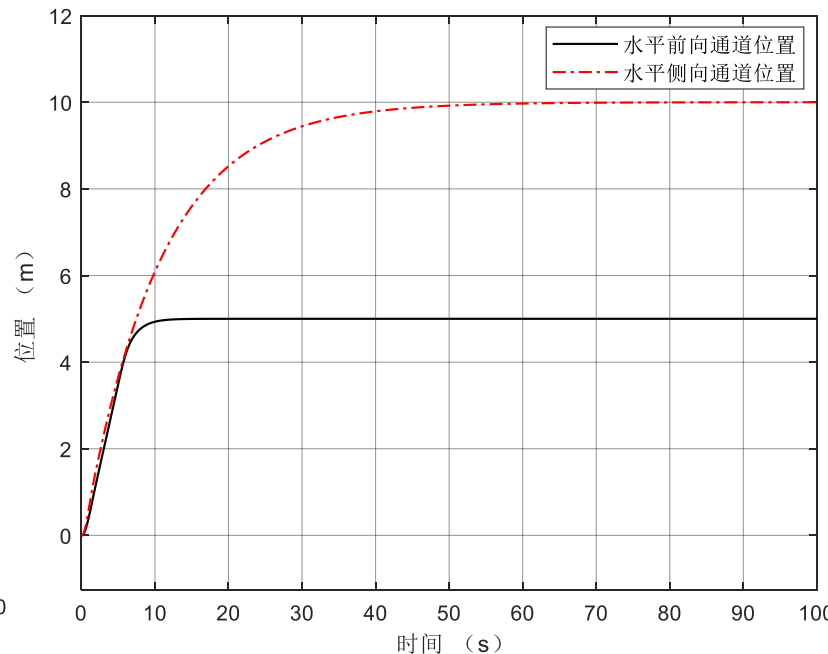
水平通道响应

3. 分析实验-仿真1.0

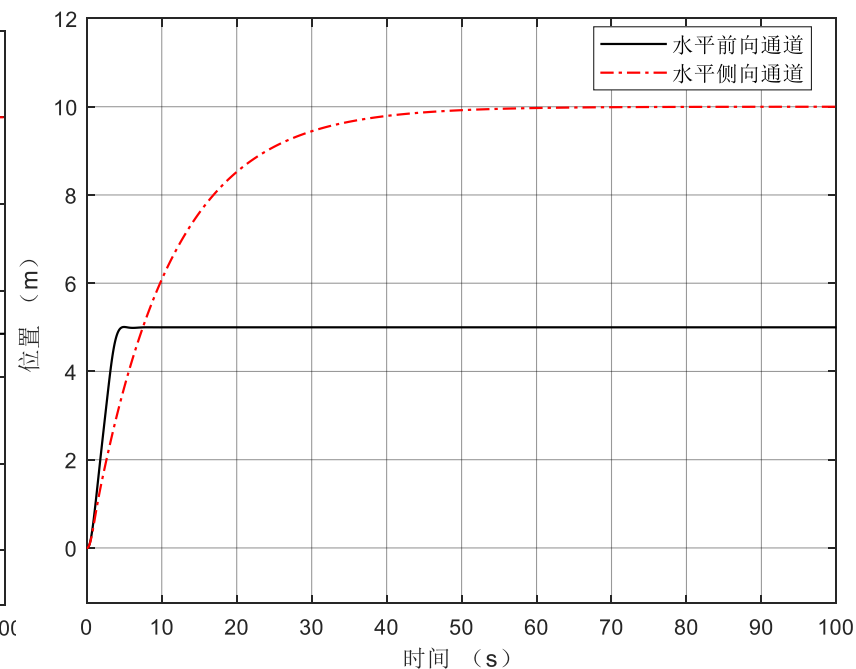
(3) 选取如下参数进行实验，分别为 $k_0 = 0.1$ ， $a_0=2$ ， $k_1=0.1$ 、 0.5 和 1 。每次运行模型前改变表7.1中的 k_1 参数，得到水平前向通道和水平侧向通道的路径跟随控制器响应，如下图所示。



$k_1=0.1$ 路径跟随控制器输出时间曲线对比图



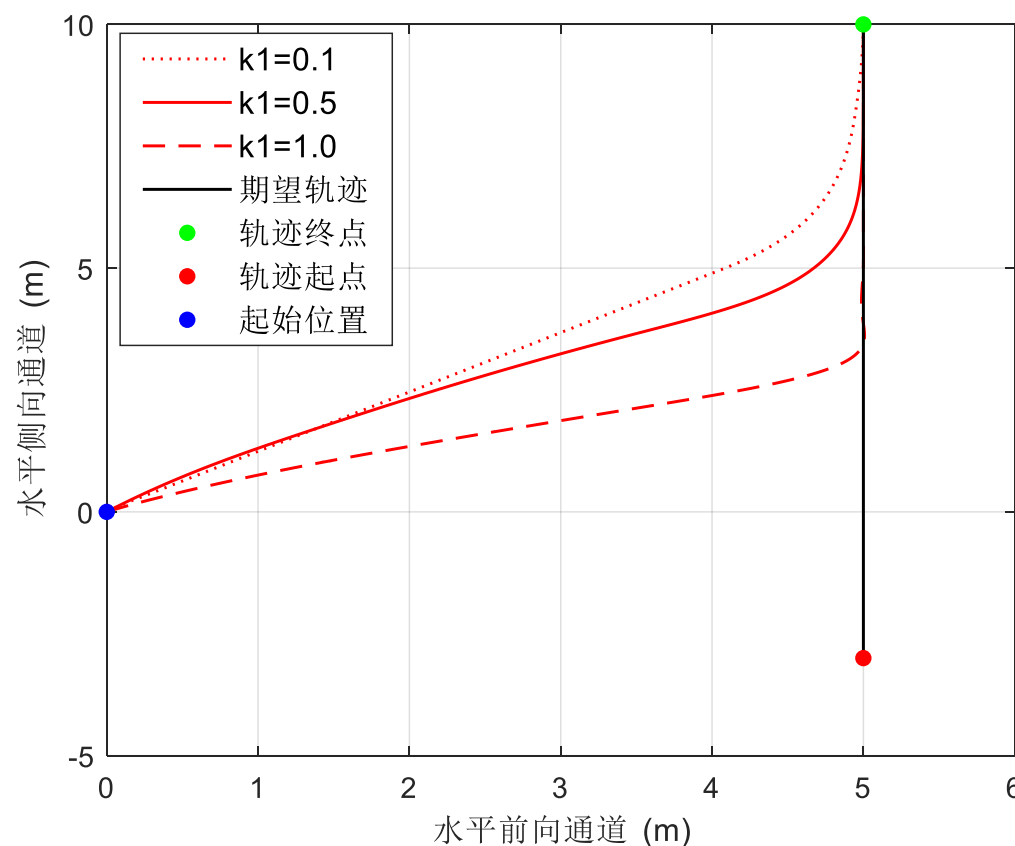
$k_1=0.5$ 路径跟随控制器输出时间曲线对比图



$k_1=1$ 路径跟随控制器输出时间曲线对比图

3. 分析实验-仿真1.0

(2) 可以看出，在保持其它两个参数不变的情况下，改变参数 k_1 的值，多旋翼可以更“直接”地贴合到期望轨迹上。这也就说明参数 k_1 的作用便是使多旋翼更快地靠近期望轨迹。在三种不同参数 k_1 的情况下，路径跟随控制器输出与期望轨迹对比图，如右图所示，该图也说明了这个规律。



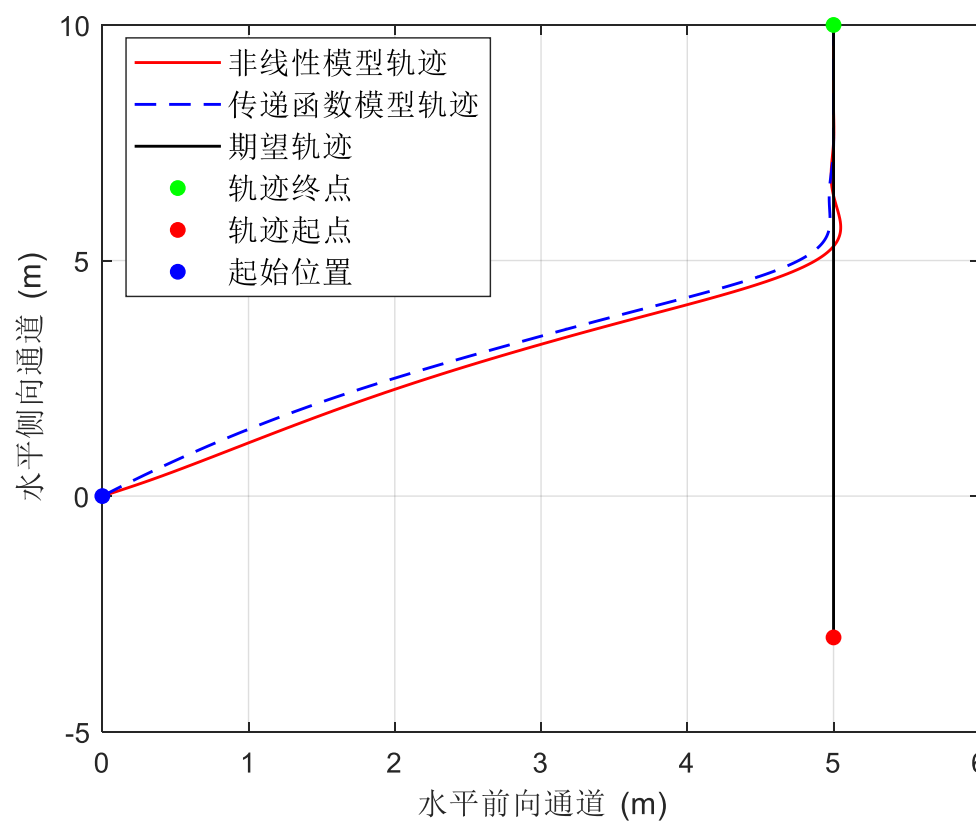
不同 k_1 水平通道响应

第七章跟随控制器设计

分析实验：仿真1.0

3. 分析实验-仿真2.0

(1) 打开文件夹“e4\4.2\sim2.0”，其中的文件与“e4\4.2\sim1.0”相同，区别在于其中的模型为非线性模型。采用与传递函数模型实验中完全相同的实验步骤，将两种模型的关键结果进行对比。为了尽可能简略且清楚地显示对比效果，这里选取 $a_0=2$ 与 $k_1=1$ 时的跟随轨迹情况进行对比。从对比结果图可以看出，我们采用系统辨识获得的传递函数模型进行实验时，可以产生与基于非线性模型设计非常接近的效果。同时也可以在FlightGear 中查看仿真飞行效果。



非线性模型水平通道响应

第七章跟随控制器设计

分析实验：仿真2.0

大纲



1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 实飞实验
6. 本章小结



4. 设计实验-实验目标

■ 准备

- 软件：MATLAB R2017b 及以上版本，基于Simulink 的控制器设计与仿真平台和实验指导包“e4.3”，CopterSim, RflySim3D
- 硬件：计算机，自驾仪

■ 目标

-在直线路径跟随理论的基础上，设计圆轨迹路径跟随算法。已知所跟随圆的圆心位于 $(0,0)$ ，半径为3，无人机初始位置随机（可定为 $(0,0)$ ）。此外偏航角需要始终保持为0，高度始终保持在2。在完成圆轨迹路径跟随算法的设计和验证之后，改变参数 k_1 的值，分析其作用。最后进行仿真1.0，仿真2.0 和硬件在环仿真。



4. 设计实验-实验设计

可以将多旋翼看作质点，其满足牛顿第二定律，表示如下

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{u}\end{aligned}$$

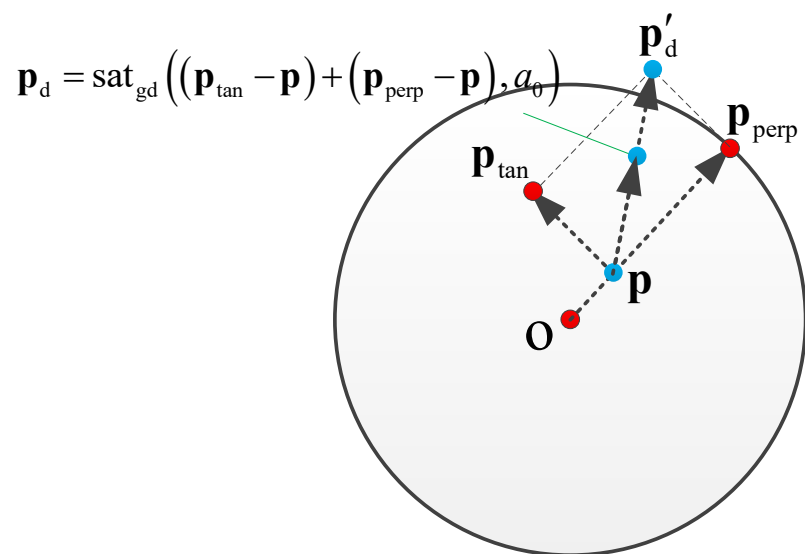
其中 $\mathbf{u} \in \mathbb{R}^2$ 表示虚拟控制， $\mathbf{v} \in \mathbb{R}^2$ 表示多旋翼的速度， $\mathbf{p} \in \mathbb{R}^2$ 表示多旋翼的当前位置。

设计实验问题可以描述为：当前多旋翼位置为 $\mathbf{p} \in \mathbb{R}^2$ ，路轨迹为圆，圆心为 $\mathbf{o} \in \mathbb{R}^2$ ，

半径为 R ，设计虚拟控制 $\mathbf{u} \in \mathbb{R}^2$ ，使得多旋翼最终能够绕圆周按逆时针飞行。

4. 设计实验-实验设计

前两个实验实现了直线的路径跟随，对于圆的路径跟随，也可以采用相同的思路。



(a) 飞行器在圆内

如图所示, $p_{\text{perp}} \in \mathbb{R}^2$ 表示多旋翼与圆的最近距离点, 可以表示为

$$p_{\text{perp}} = o + (p - o) \frac{R}{\|p - o\|}$$

那么

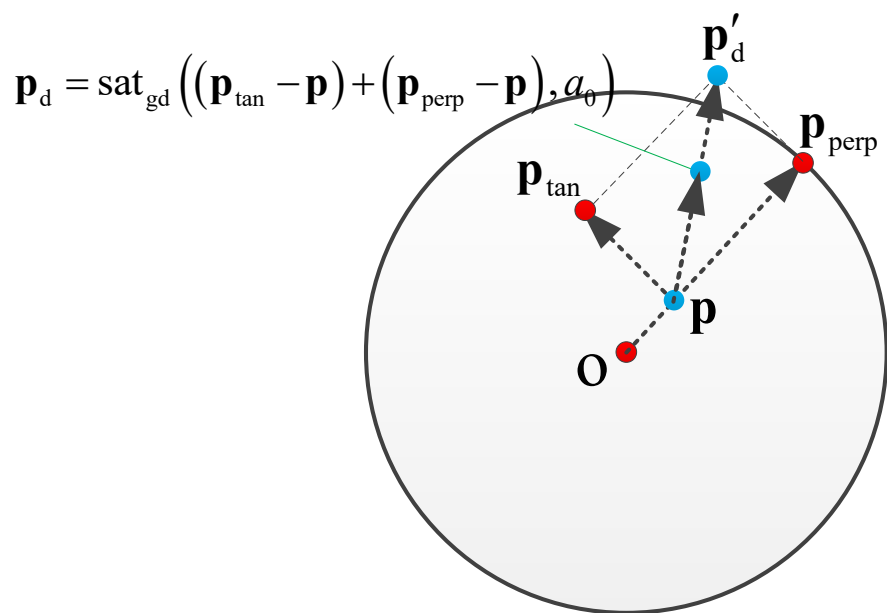
$$p - p_{\text{perp}} = \lambda (p - o)$$

其中

$$\lambda = 1 + \frac{R}{\|p - o\|}$$

令 p_{perp} 能够产生吸引多旋翼的引力。这样飞行器就会趋近圆。

4. 设计实验-实验设计



(a) 飞行器在圆内

另外一方面，我们希望多旋翼能够绕圆周进行逆时针飞行，那么需要能够绕圆周进行逆时针飞行切向方向的吸引力。我们希望这个切向方向引导点 p_{tan} 。到当前多旋翼的位置具有单位长度，那么

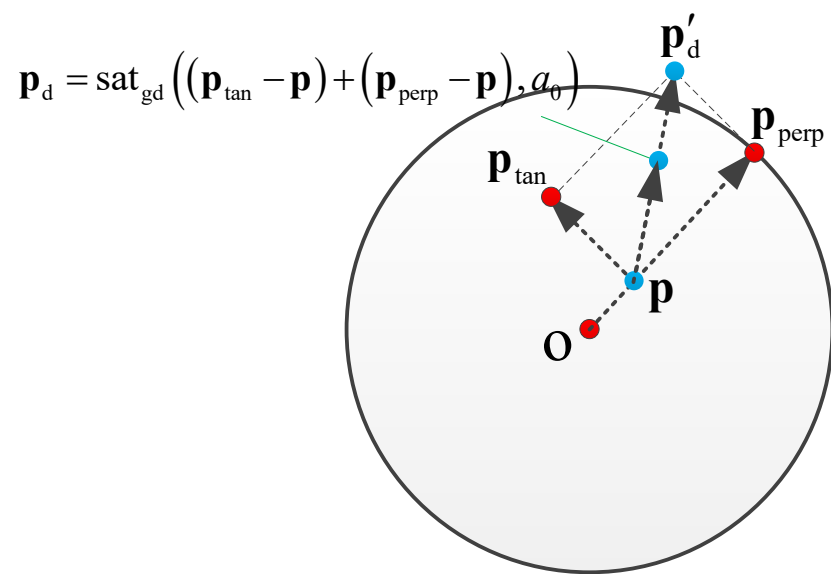
$$p_{tan} - p = R_{\alpha=90^\circ} \frac{p - p_{perp}}{\|p - p_{perp}\|}$$

其中 R_α 表示旋转矩阵，表示为

$$R_\alpha = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$$

一向量与其相乘后，向量可以逆时针旋转 α 角度。

4. 设计实验-实验设计



(a) 飞行器在圆内

圆路径跟随的控制器为
$$\mathbf{u} = -\frac{1}{k_2}(\mathbf{p} - \mathbf{p}_d) - \frac{1}{k_2} \mathbf{v}$$

其中
$$\mathbf{p}_d = \text{sat}_{\text{gd}}(k_0(\mathbf{p}_{\text{tan}} - \mathbf{p}) + k_1(\mathbf{p}_{\text{perp}} - \mathbf{p}), a_0)$$
。

这里的 $k_0, k_1 > 0$ 分别表示趋向 \mathbf{p}_{tan} 和 \mathbf{p}_{perp} 的增益大小。

若飞行器离圆很远，那么

$$\mathbf{p}_d \approx \text{sat}_{\text{gd}}(k_1(\mathbf{p}_{\text{perp}} - \mathbf{p}), a_0)$$

此时飞行器的绝大部分控制是趋近圆。若飞行器已经在圆上，那么

$$\mathbf{p}_d = \text{sat}_{\text{gd}}(k_0(\mathbf{p}_{\text{tan}} - \mathbf{p}), a_0)$$

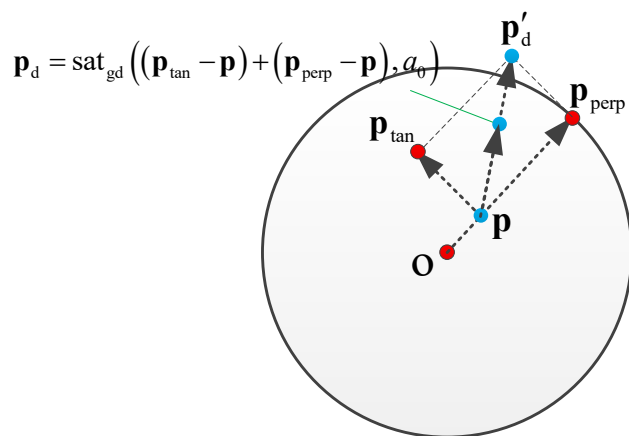
此时飞行器的绝大部分控制是绕圆飞行。

4. 设计实验-实验设计

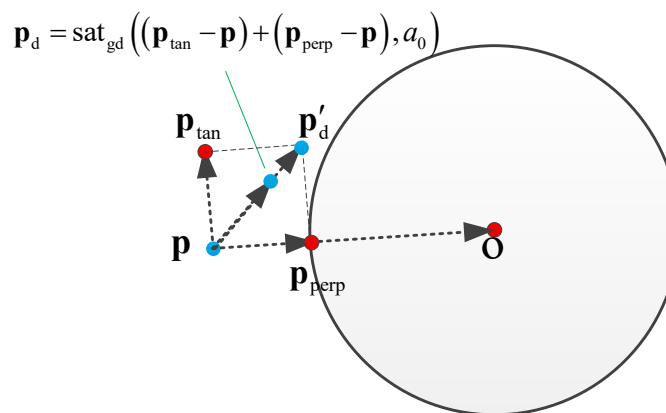
为了更好地理解物理意义。令 $k_0 = k_1 = 1$, 那么

$$\mathbf{p}_d = \text{sat}_{\text{gd}} \left((\mathbf{p}_{\text{tan}} - \mathbf{p}) + (\mathbf{p}_{\text{perp}} - \mathbf{p}), a_0 \right)$$

其中 \mathbf{p}_d 的物理意义如图所示。由图可见，无论飞行器在给定的圆内还是圆外都有趋向圆的趋势。



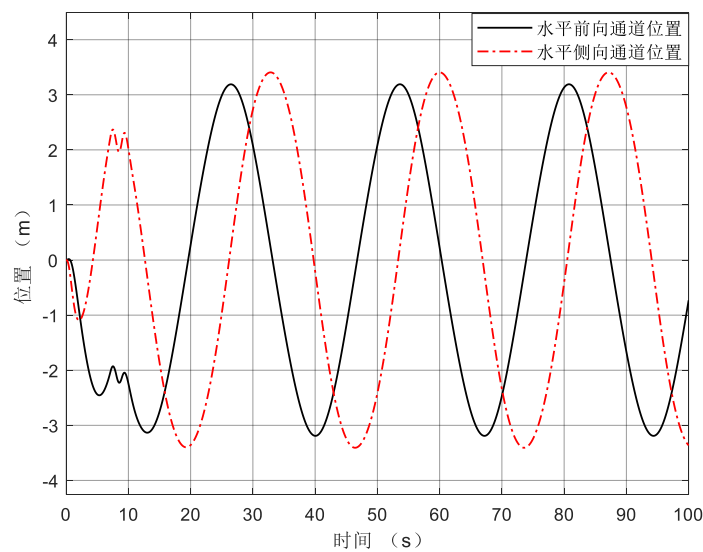
(a) 飞行器在圆内



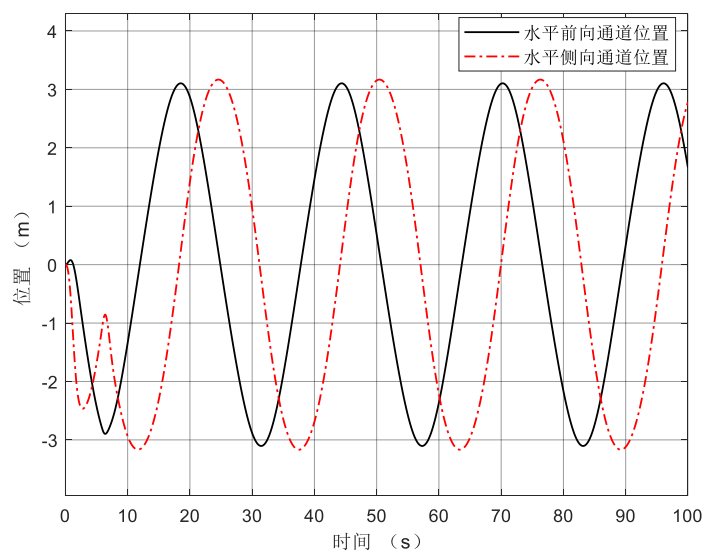
(b) 飞行器在圆外

4. 设计实验-仿真1.0

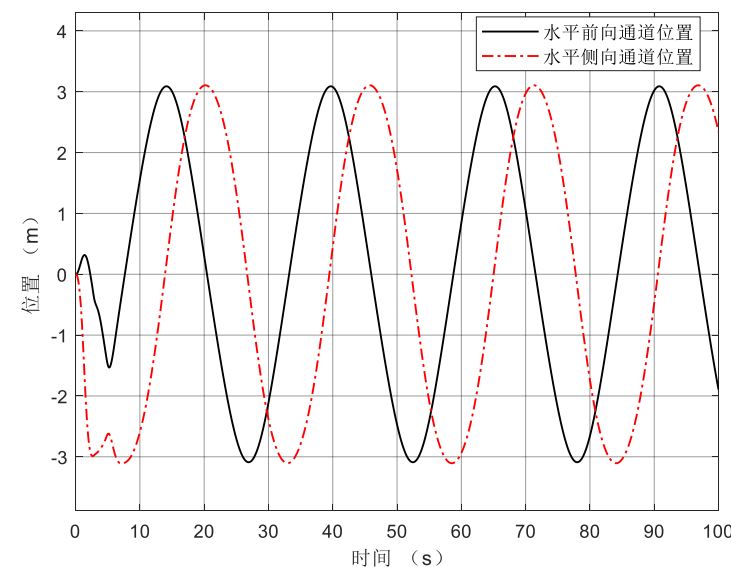
(1) 运行“e4\4.3\sim1.0\StartSimulation.m”初始化参数。选取如下参数进行实验，分别为 $k_0 = 0.3$, $a_0=3$, $k_1=0.3, 0.7, 1$, 获得不同的结果，分别进行保存。水平前向通道以及水平侧向通道的路径跟随控制器输出时间曲线的对比图所示。可以看出，无论参数如何修改，两通道的期望输出曲线均明显存在大约为半个周期的相位差，达到了实验目标的要求。



K1=0.3 水平通道响应



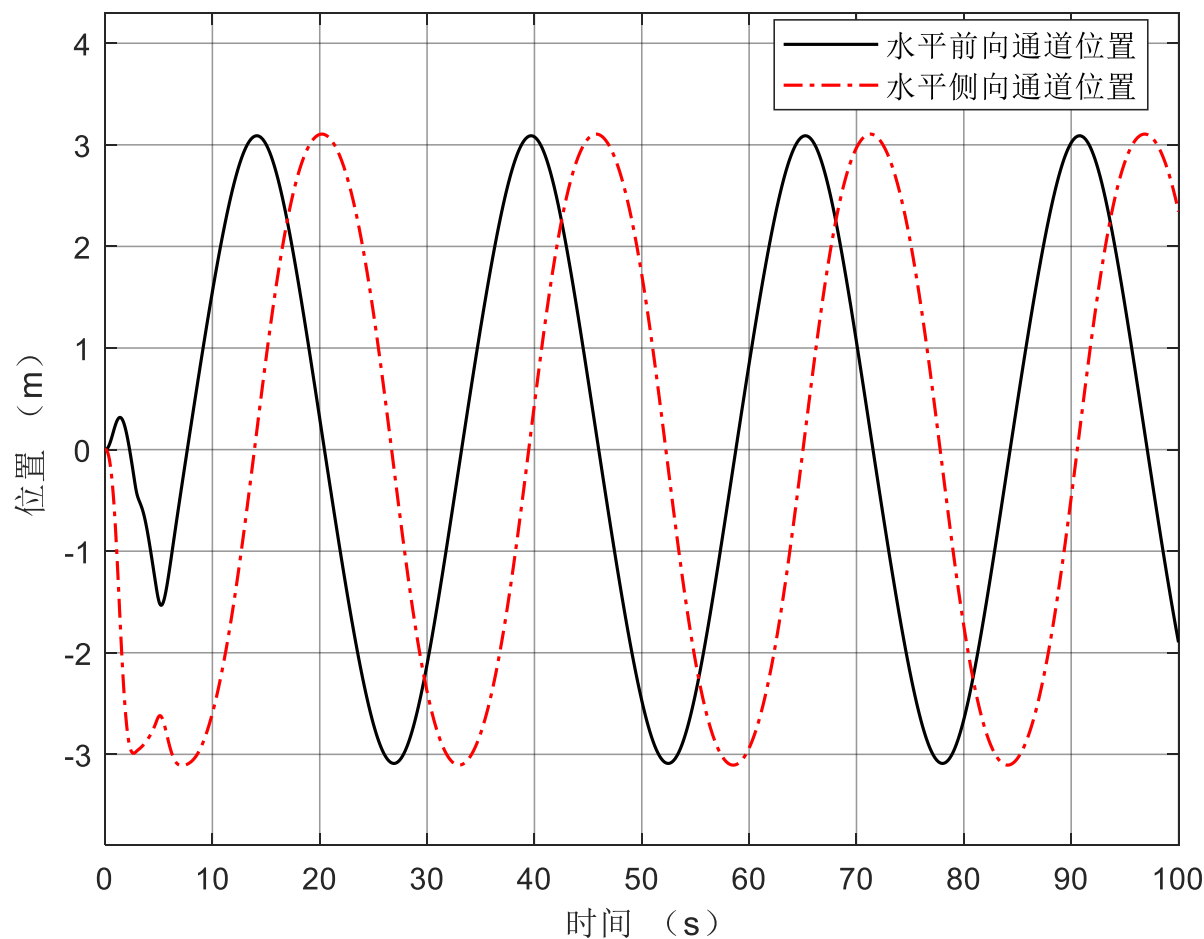
K1=0.7 水平通道响应



K1=1 水平通道响应

4. 设计实验-仿真1.0

(2) 在三种不同参数 k_1 的情况下，路径跟
随控制器输出与期望轨迹之间对比如右图所示。
可以看出，随着参数 k_1 的增大，多旋翼的期望
输入轨迹曲线愈加靠近期望圆轨迹；此外，随
着参数 k_1 的增大，多旋翼靠近期望圆轨迹也会
变得越来越“直接”，靠近轨迹就越精确。更
具体地，多旋翼更期望以优先靠近“当前位置
在期望圆轨迹上的过圆心直线与圆的交点”的
方式靠近期望轨迹。这样的结果与我们设计的
控制器是相符的。



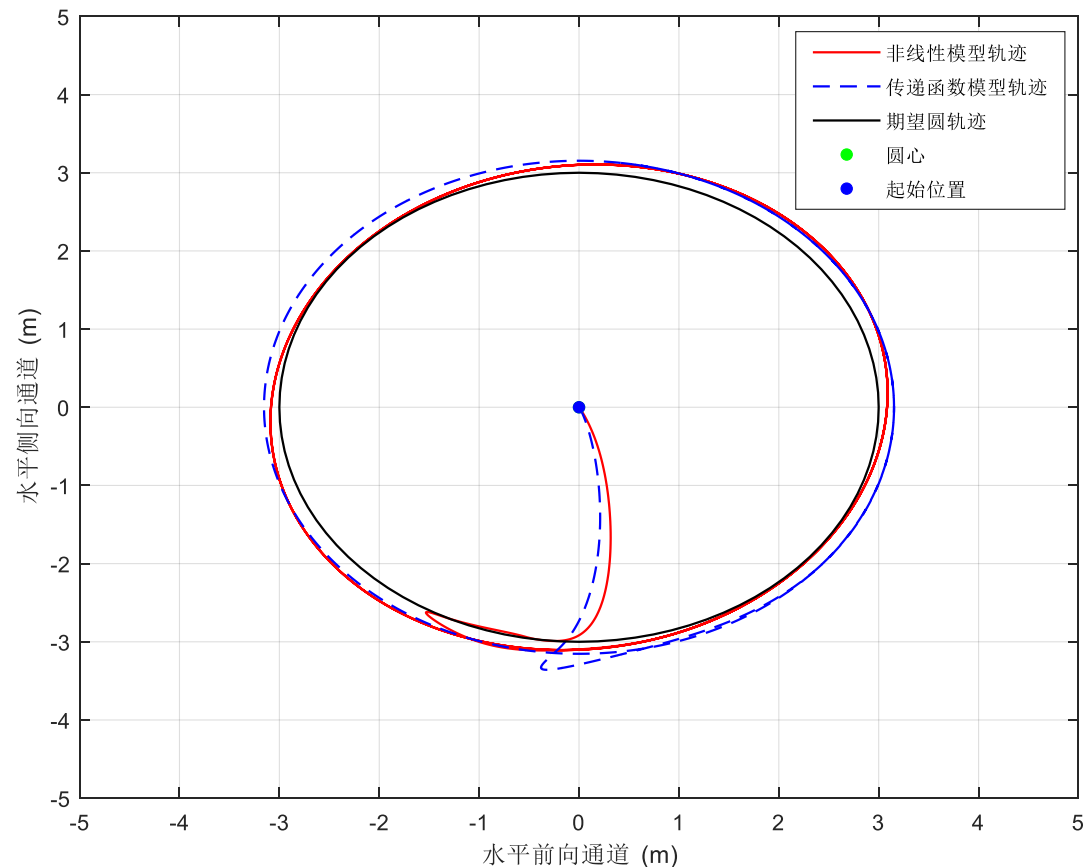
水平通道响应

第七章跟随控制器设计

设计实验：仿真1.0

4. 设计实验-仿真2.0

打开“e4\4.3\sim2.0” Simulink 文件，其中的闭环系统结构与“e4\4.3\sim1.0”中的相同，而区别在于其中的模型为非线性模型。采用与传递函数模型实验中完全相同的实验步骤，将两种模型的关键结果进行对比，如右图所示，从对比结果可以看出，我们采用系统辨识获得的模型进行跟随控制器设计，可以产生与基于非线性模型设计非常接近的效果。另一方面，这里的非线性模型所获得的圆跟随轨迹并不对称，这是因为非线性模型存在着诸多不确定性和限制。同时也可以可以在FlightGear中查看仿真飞行效果。



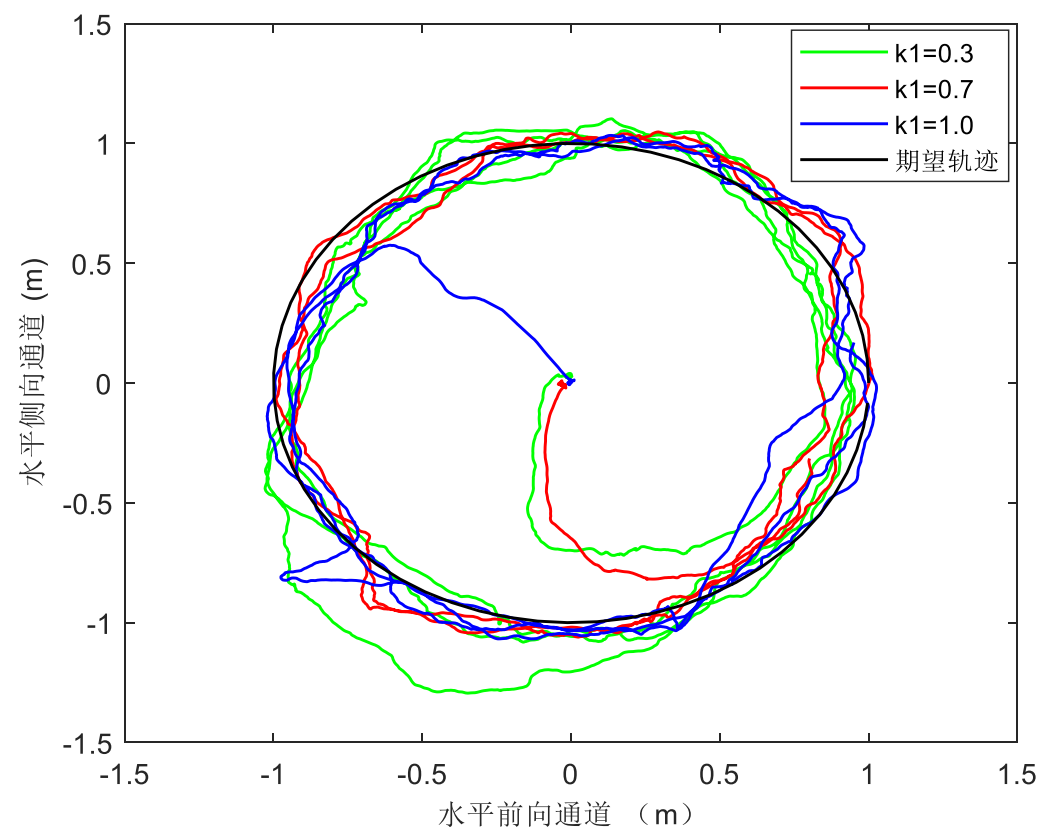
非线性模型水平通道响应

第七章跟随控制器设计

设计实验：仿真2.0

4. 设计实验-硬件在环仿真

首先准备好自驾仪，打开RflySim，同时打开MATLAB2017b，运行“StartSimulation”初始化文件。运行后，再开打并运行Simulink 模型文件“e4_3_Model_TrajectoireFollowing_HTIL_2017b”，便可以在Rflysim3D 软件中看到硬件在环仿真的效果。实验数据记录在“Pos1”文件中，结果如图所示。硬件在环仿真实验分别实现了 $k_1=0.3, 0.7, 1.0$ 的飞行实验，从实验结果可以看出，控制 k_0 不变，改变 k_1 的大小，轨迹随着 k_1 的增大而越来越靠近期望轨迹，但是在期望轨迹附近的抖动也更加明显。



硬件在环水平通道响应

第七章路径跟随控制器设计 硬件在环仿真实验

大纲



1. 基本原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 实飞实验
6. 本章小结



5. 实飞实验-实验目标

■ 准备

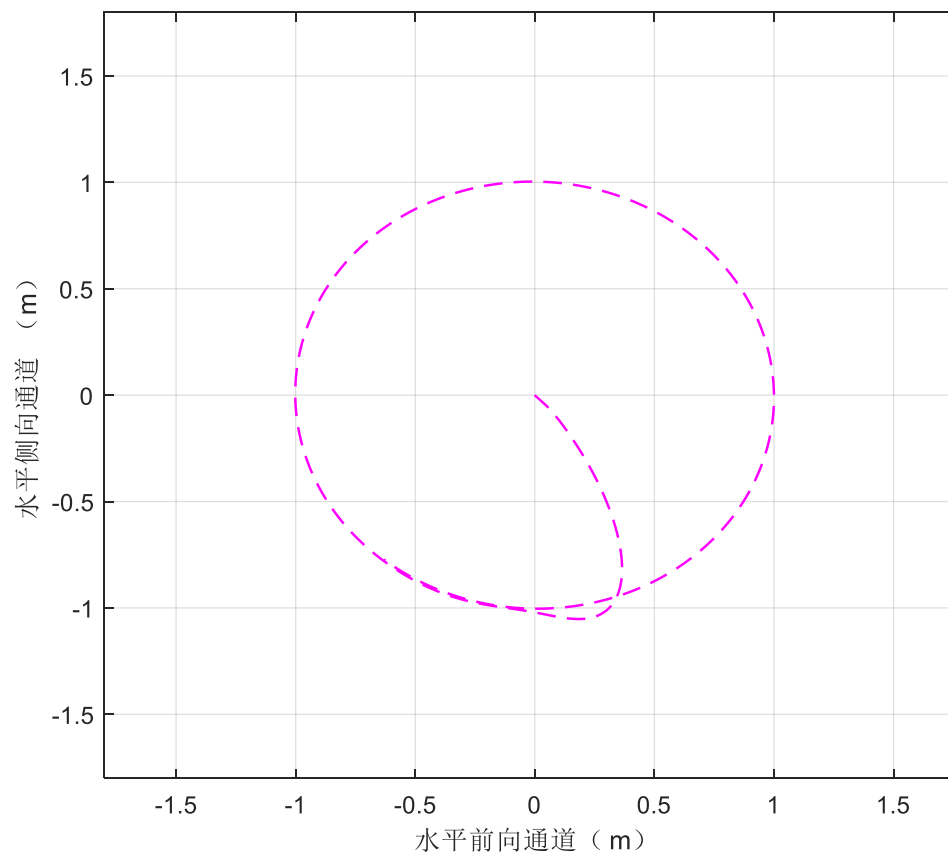
- 软件：MATLAB R2017b 及以上版本，基于Simulink 的控制器设计与仿真平台和实验指导包“e4.4”
- 硬件：计算机，室内定位系统，带半自主飞控的多旋翼飞行器

■ 目标

1. 基于实验第四章中实飞实验辨识出来的传递函数模型，使用设计实验中设计出的圆跟随控制器，查看仿真效果。
2. 将1) 中设计出的圆路径跟随控制器应用于真机，查看实验效果。

5. 实飞实验-仿真1.0

运行“e4\e4.4\sim1.0\start.m”，
“e4_4_trajctoire_following.slx”模型
文件会自动打开。以基础实验中相同的步骤
调节PID 控制器得到较好的控制效果。获得
相应的圆轨迹如右图所示。



PID控制器下水平通道响应

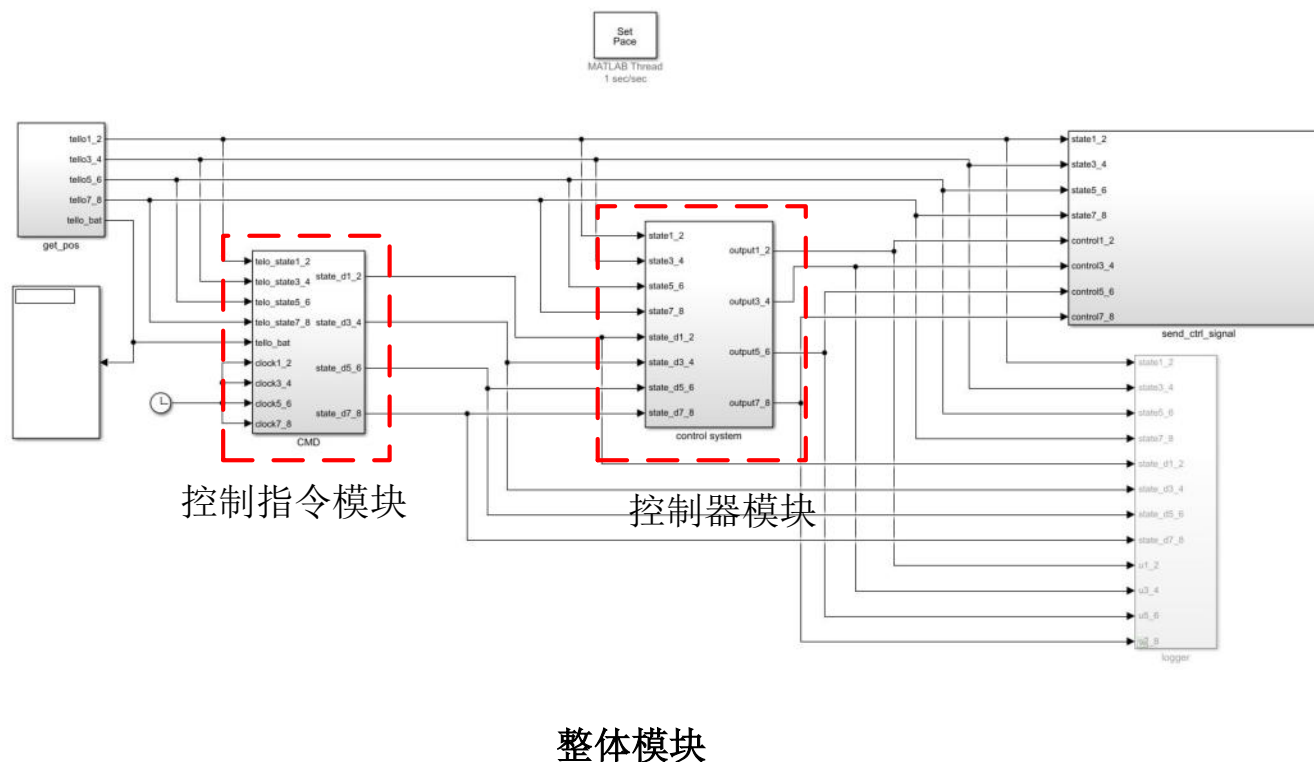
第七章跟随控制器设计

实飞实验：仿真1.0

5. 实飞实验-仿真2.0

(1) 在MATLAB 中点击运行“start_tello.m”文件，进行初始化以及启动相应的Simulink 程序

“e4_4_trajectoire_planning.slx”，模型如下图所示。





5. 实飞实验-仿真2.0

(2) 系统启动流程:

✓ 启动OptiTrack

打开一个新终端，运行命令“`roslaunch mocap_optitrack multi_rigidbody8.launch`”；

✓ 启动tello_driver

打开一个新终端，运行命令“`roslaunch tello_driver tello_node.launch`”；

✓ 起飞Tello 打开一个新终端，运行命令“`roslaunch tello Tello_takeoff_all`”，可以看到四旋翼起飞并保持悬停在正上方高度1 米位置；

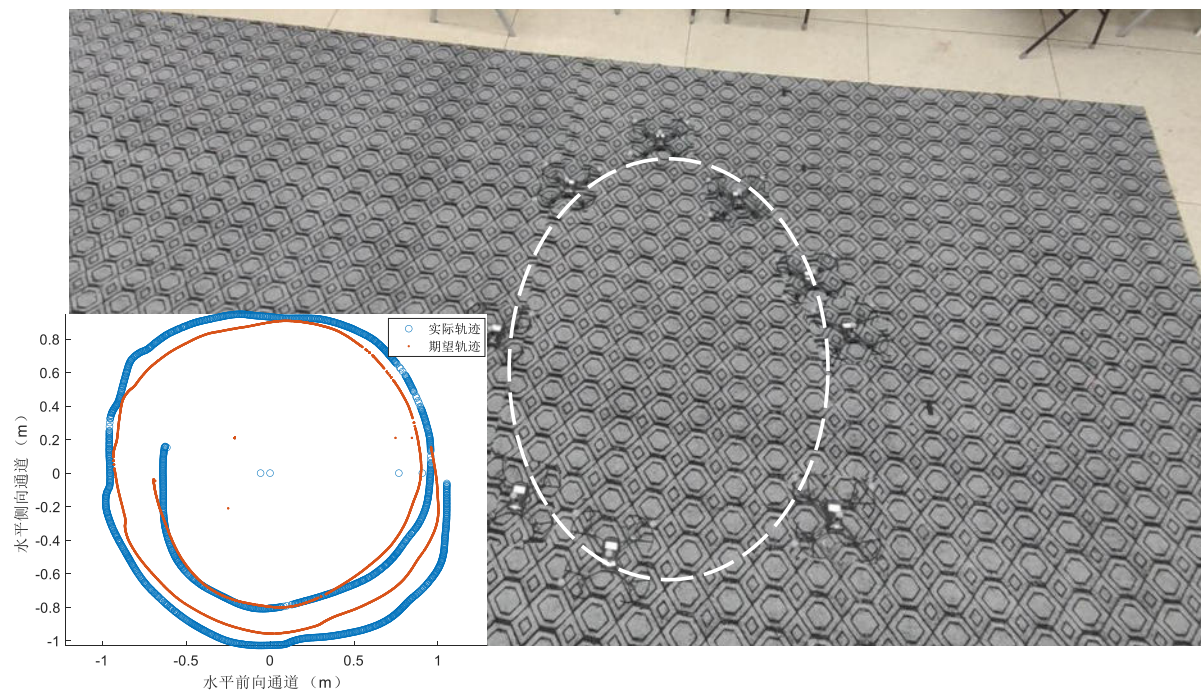
✓ 运行MATLAB 控制程序 点击运行“`e4_4_trajectoire_planning.`”文件；

✓ 降落Tello 打开一个新终端，运行命令“`roslaunch tello Tello_land_all`”，在多旋翼降落后，结束所有终端。

5. 实飞实验-仿真2.0

(3) “e4_4_trajecoire_planning.slx” 文件中包含数据存储模块用于记录结果，见工作区变量 “tello3_states”、

“tello3_states_d” 代表四旋翼飞行过程的期望指令和实际反馈结果。运行 “tello_plot.m” 模型文件，即可得到如右图所示的实飞结果。

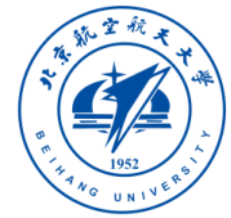


实飞轨迹图

第七章路径跟随控制器设计

实飞实验

6. 总结



- (1) 通过基础实验，了解直线路径跟随算法的设计原理，了解整体的模块框架。
- (2) 通过分析实验，了解直线路径跟随算法相关参数对算法性能的影响，并分析原因。
- (3) 通过设计实验，根据前面两个小实验了解到的原理，设计圆轨迹路径跟随控制器。
- (4) 通过实飞实验，验证设计的圆轨迹跟随控制器算法的正确性。
- (5) 通过以上实验，让读者较为深入掌握不同路径跟随控制器的设计方法。

如有疑问，请到<https://rflysim.com/course> 查询更多信息。

致谢



感谢可靠飞行控制研究组同学为本门课程准备做出的贡献



刘浩



毛鹏达



马泽清



谢谢!

全 权

qq_buaa@buaa.edu.cn

可靠飞行控制研究组



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