

多旋翼飞行器远程控制实践

第4章 系统辨识实验

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

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.基本原理

□ 频域系统辨识



???

系统辨识主要包括**频域辨识**和时域辨识

输入/输出数据

系统辨识



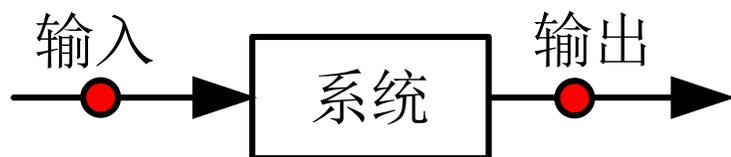
系统模型

1. 系统稳定性
2. 动态系统仿真, 系统参数获取
3. 控制器设计
4. 系统频率特性分析

1.基本原理

□ 两种辨识

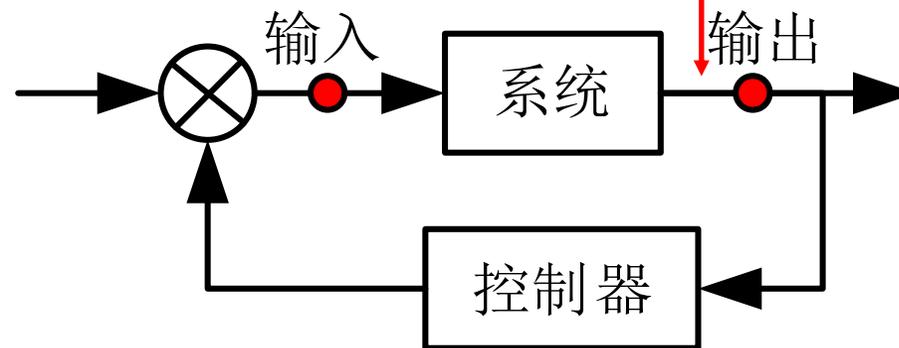
开环辨识



系统**稳定时**采用的方法。输入信号可以任意选择，因此开环实验相比于闭环实验能获得更多的系统信息

如果系统可以脱离控制器情况下工作，开环实验是更好的选择！

闭环辨识



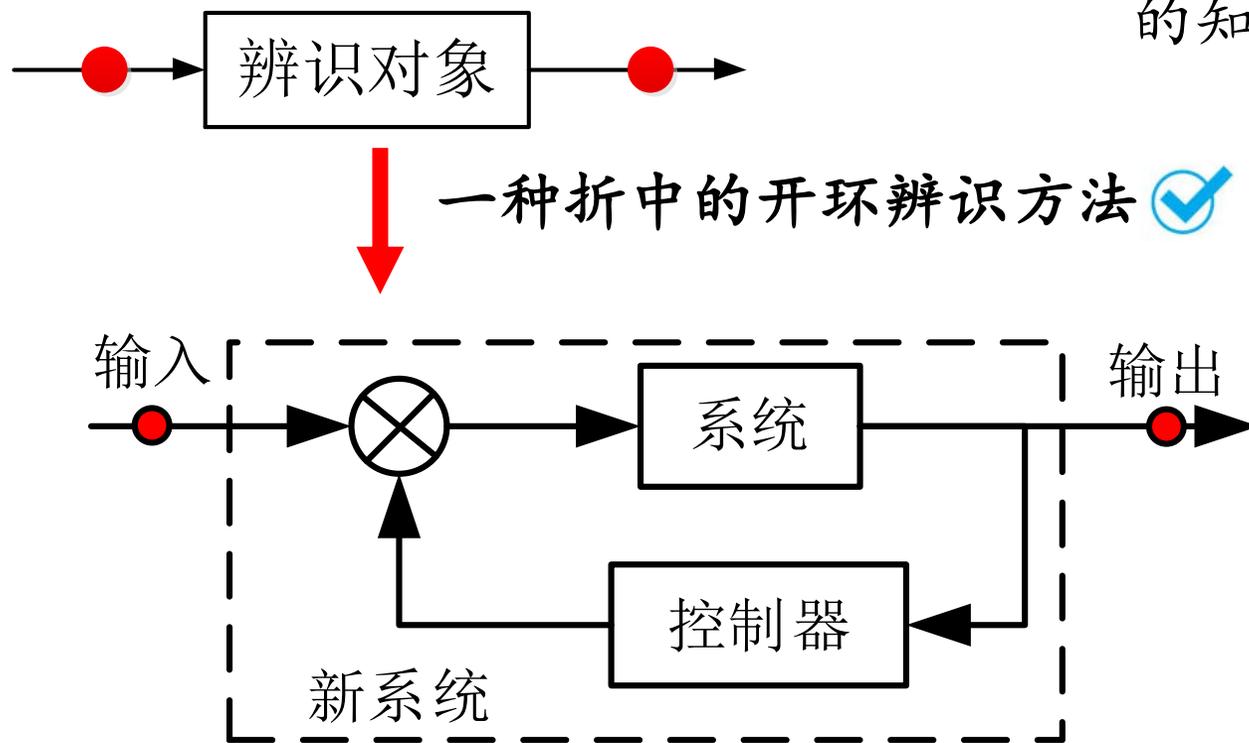
一些系统是**不稳定**的，它们需要在反馈控制器的控制下工作，因此这些系统的输入信号由控制器决定

1.基本原理



□ 两种辨识

可以应对**不稳定**系统，
激励充分，但不能精确的知道原来的辨识对象



1. 基本原理

□ 系统辨识

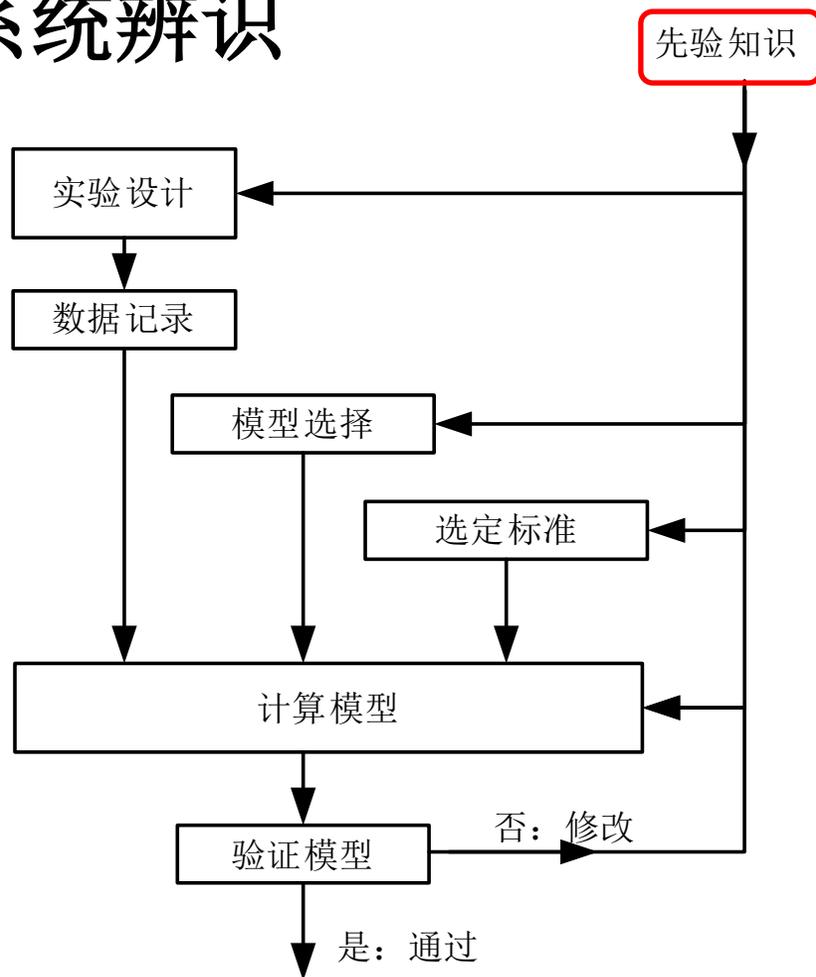


图. 系统辨识过程

1. 先验知识：包括系统特性、数据采集方法以及待辨识系统其它方面的已有知识。这些知识对选择备选模型、设计实验、决定计算方法和检验准则等都有重要作用。由于辨识目的不同，即使是对同一系统进行辨识，其应用的先验知识也可能有很大差别。

1. 基本原理

□ 系统辨识

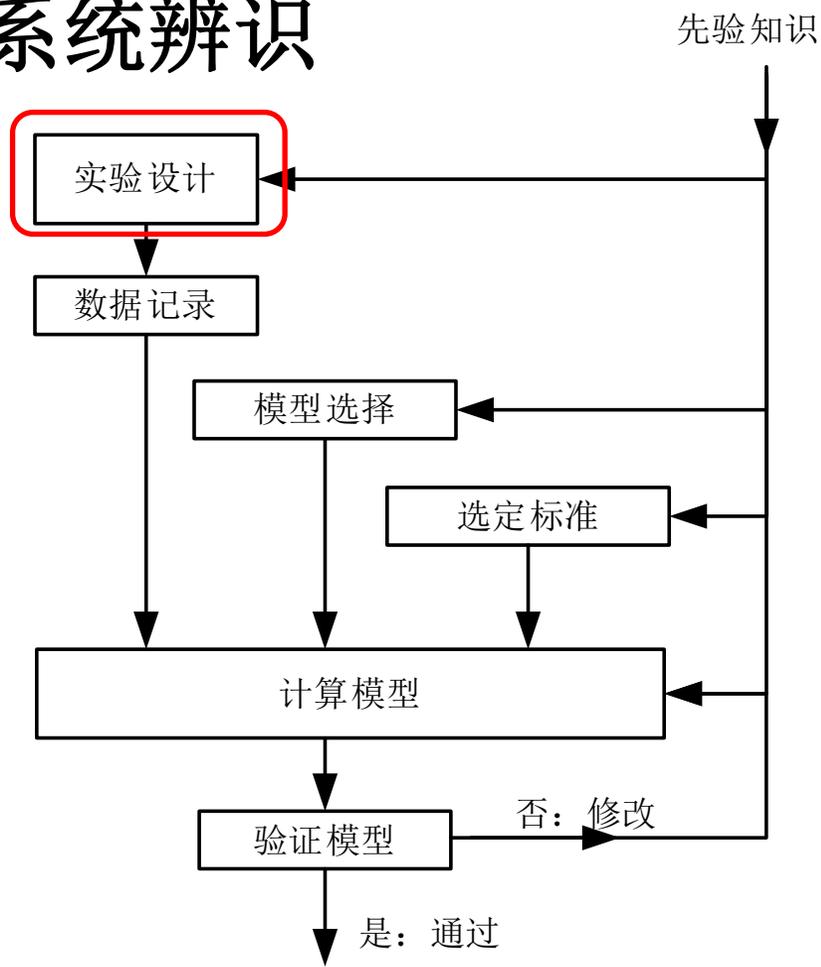


图. 系统辨识过程

2. 实验设计：目的是在已知条件下，获得能反映系统性能的输入 / 输出数据。在输入 / 输出数据的采集过程中，有时需要针对辨识过程进行特殊的实验设计。在实验中，可能需要用户确定测量**哪些信号**，以及什么时候**测量**这些信号。

1.基本原理

□ 频率信号测量注意点

1. 扫频信号需要对进行辨识的通道产生**足够激励**，即**幅值足够大**。频率范围与飞行系统通道的可用频率范围**相符合**。

选翼飞行器系统的频率范围： $0.3\text{rad/s} \sim 12\text{rad/s}$ 。

2. 开始和结尾处于**配平**状态

3. 需要加入适当的**高斯白噪声**来丰富其频谱。

4. 单个扫频输入信号的持续时间，称为“长度” $T_{\text{rec}} \geq T_{\text{max}}$ ， T_{max} 单个扫频输入信号中所存在的最大周期

5. 相应的时间历程数据大致是对称的

6. 真实实验中，还需要考虑传感器采集频率

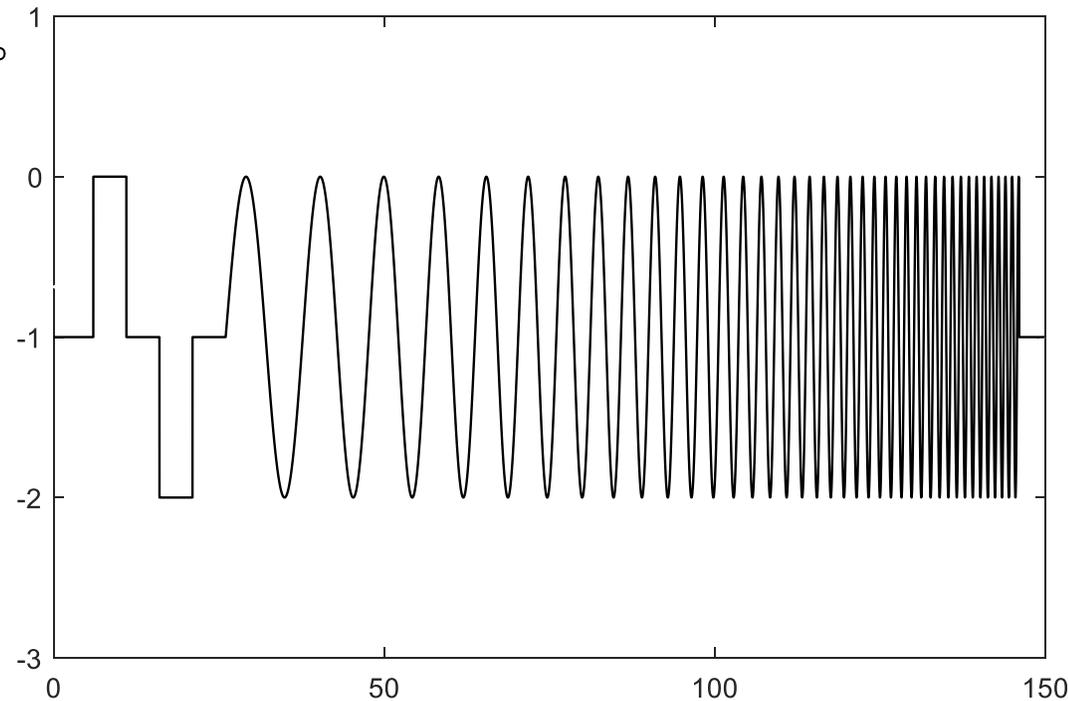


图. 扫频信号示例

1. 基本原理



□ 系统辨识

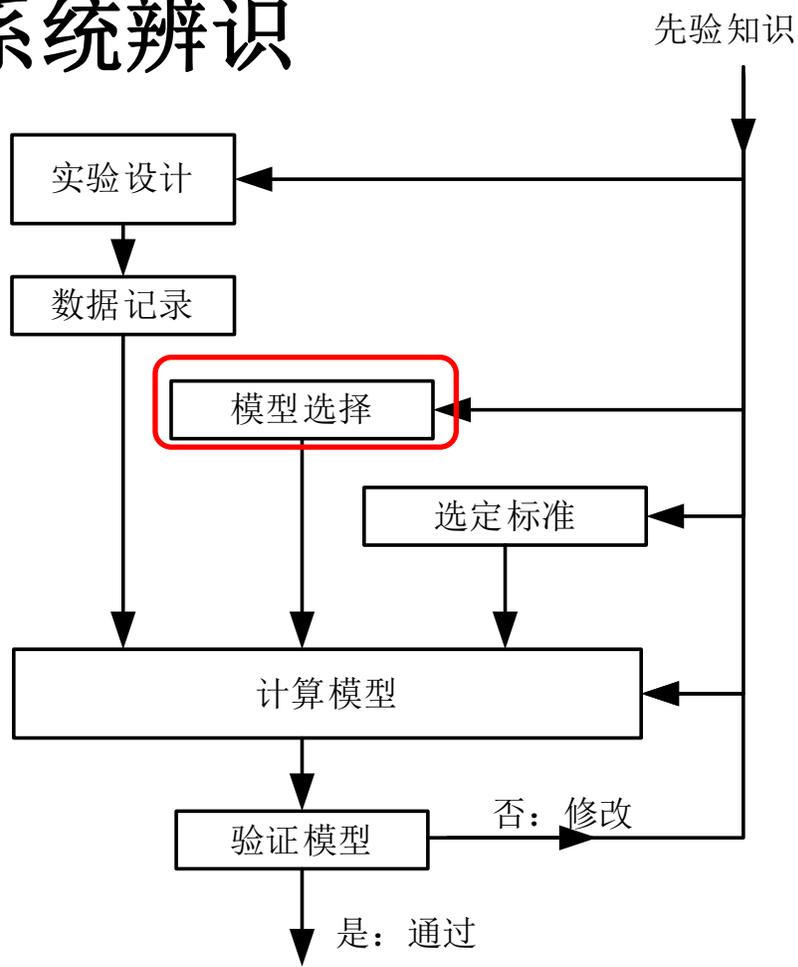


图. 系统辨识过程

3. 模型选择：选择一系列的备选模型，通过后续验证，从中确定最合适的模型。通过数学建模，可以得到一个未知参数化模型，然后通过参数辨识方法计算出模型中的未知参数。

线性/非线性？连续/离散？
传递函数/状态空间？系统阶数？……

4. 数据记录：通过合理的实验来获得输入/输出数据。

1. 基本原理



□ 系统辨识

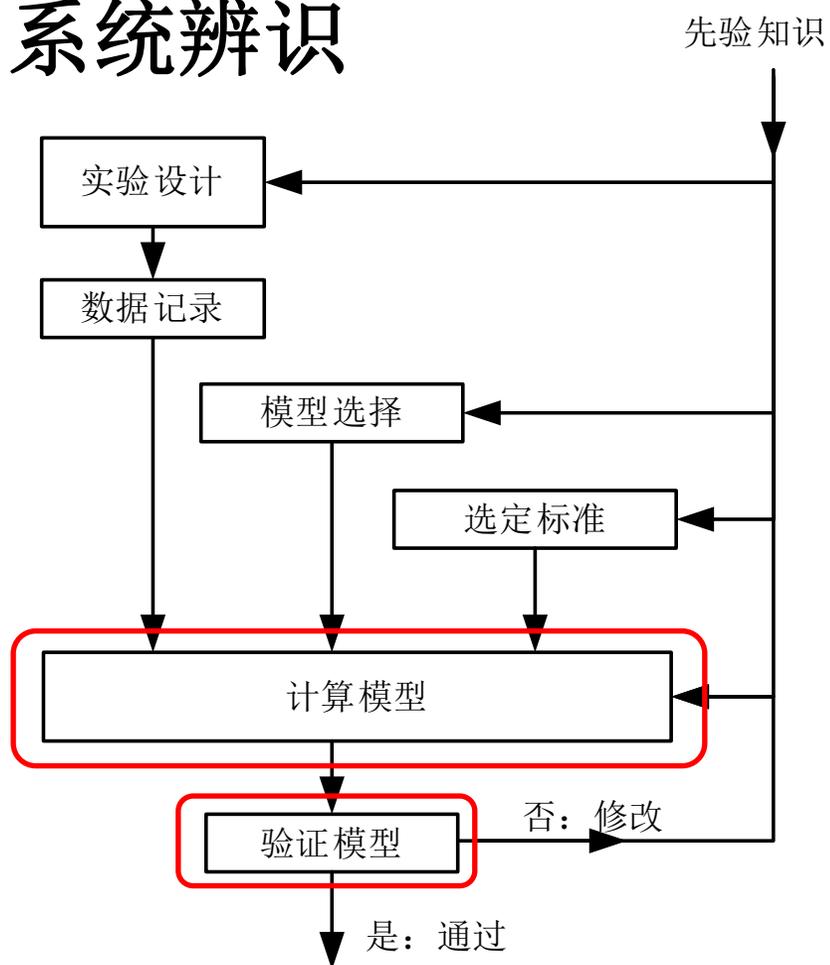


图. 系统辨识过程

5. 计算模型：采用合适的优化方法来计算备选模型的未知参数。

6. 验证模型：建立一个标准来检验备选模型与计算出的参数是否满足设计要求。通常，该标准的确定依赖于观测到的数据、先验知识以及待辨识模型的用途。如果模型及其参数可以通过校验，则可以将其作为最终辨识出来的模型，否则需要重复上述步骤，直至模型通过验证。

1. 输入相同的数据，对比系统输入的相似度；
2. 辨识工具箱中的评价指标

1. 基本原理

□ 数学模型

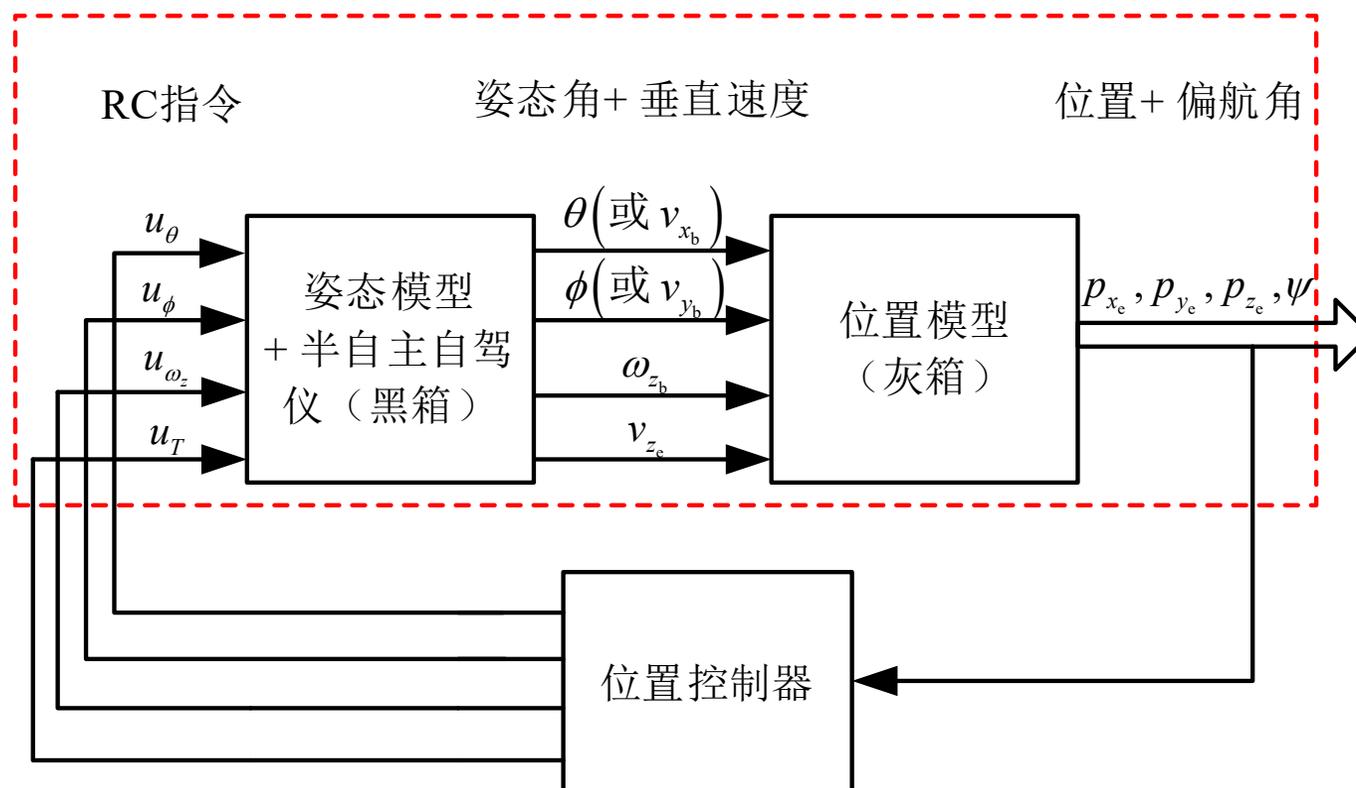


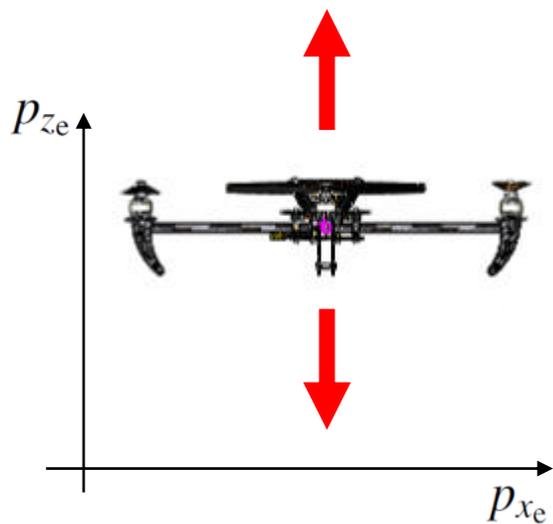
图. 基于半自主自驾仪的位置控制闭环图

1.基本原理

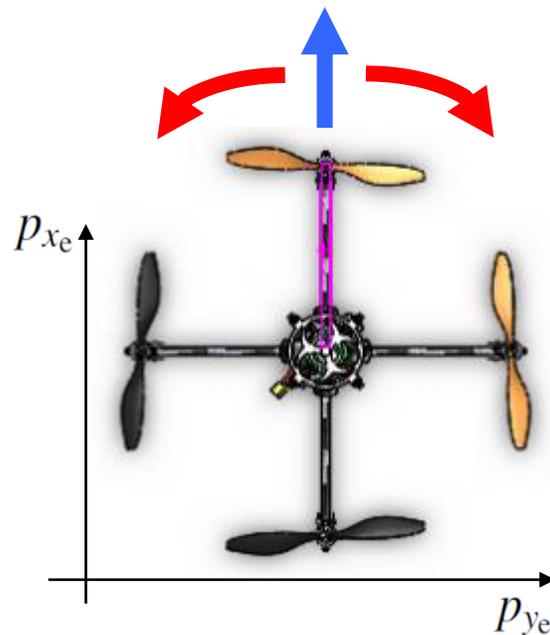


□ 数学模型

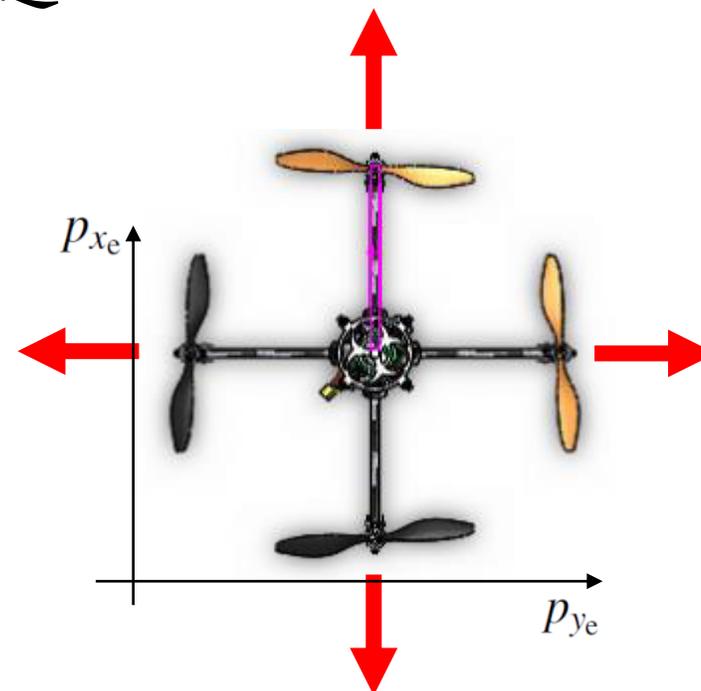
在半自主飞控平台的作用下解耦为三个相对独立的通道



高度通道



偏航通道



水平通道

1. 基本原理

□ 数学模型

高度通道

$$\begin{aligned}\dot{p}_{ze} &= v_{ze} \\ \dot{v}_{ze} &= -k_{v_z} v_{ze} - k_{u_T} u_T\end{aligned}$$



$$p_{ze}(s) = \boxed{G_{p_z u_T}(s)} u_T(s) \xrightarrow{\text{比例控制器}} u_T = k_{p_z} p_{ze} + u_{p_z}$$

包含一个一阶积分环节，是一个临界稳定的传递函数

$$\begin{aligned}\dot{p}_{ze} &= v_{ze} \\ \dot{v}_{ze} &= -k_{u_T} k_{p_z} p_{ze} - k_{v_z} v_{ze} - k_{u_T} u_{p_z}\end{aligned}$$



$$p_{ze}(s) = G_{p_z u_{p_z}}(s) u_{p_z}(s)$$

仿真、实飞

??? 垂直速度可测量

$$v_{ze}(s) = G_{v_{ze} u_T}(s) u_T(s)$$



$$p_{ze}(s) = \frac{1}{s} G_{v_{ze} u_T}(s) u_T(s)$$

仿真

1. 基本原理

□ 数学模型（假设）

偏航通道

$$\dot{\psi} = \omega_z$$

$$\dot{\omega}_z = -k_{\omega_z} \omega_z + k_{u_{\omega_z}} u_{\omega_z}$$

$$\dot{\psi} = \omega_z$$

$$\dot{\omega}_z = -k_{\omega_z} \omega_z - k_{u_{\omega_z}} k_{\psi} \psi + k_{u_{\omega_z}} u_{\psi}$$

??? 偏航角速度可测量

$$\omega_z(s) = G_{\omega_z u_{\omega_z}}(s) u_{\omega_z}(s)$$

$$\psi(s) = G_{\psi u_{\omega_z}}(s) u_{\omega_z}(s) \xrightarrow{\text{比例控制器}} u_{\omega_z} = -k_{\psi} \psi + u_{\psi}$$

包含一个一阶积分环节，是一个临界稳定的传递函数

$$\psi(s) = G_{\psi u_{\psi}}(s) u_{\psi}(s)$$

$$\psi(s) = \frac{1}{s} G_{\omega_z u_{\omega_z}}(s) u_{\omega_z}(s)$$

1. 基本原理

□ 数学模型（假设）

水平位置通道

耦合现象 $\mathbf{R}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$

$$\dot{\mathbf{p}}_h = \mathbf{R}_\psi \mathbf{v}_{hb}$$

$$\dot{\mathbf{v}}_{hb} = -\mathbf{K}_{v_{hb}} \mathbf{v}_{hb} - g \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Theta_h$$

$$\dot{\Theta}_h = \omega_{hb}$$

$$\dot{\omega}_{hb} = -\mathbf{K}_{\Theta_h} \Theta_h - \mathbf{K}_{\omega_{hb}} \omega_{hb} + \mathbf{K}_{u_h} \mathbf{u}_h$$

假设 ψ_d 为常数



$$\begin{aligned} \mathbf{p}_h(s) &= \text{diag}\left(\frac{1}{s}, \frac{1}{s}\right) \mathbf{R}_\psi \mathbf{G}_{v_{hb}} \mathbf{u}_h(s) \\ &= \mathbf{R}_\psi \text{diag}\left(\frac{1}{s}, \frac{1}{s}\right) \mathbf{G}_{v_{hb}} \mathbf{u}_h(s) \end{aligned}$$

1. 基本原理

□ 数学模型（假设）

水平位置通道

没有速度反馈

$$\begin{aligned} \dot{\mathbf{p}}_h &= \mathbf{R}_\psi \mathbf{v}_{hb} \\ \dot{\mathbf{v}}_{hb} &= -\mathbf{K}_{v_{hb}} \mathbf{v}_{hb} - g \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Theta_h \\ \dot{\Theta}_h &= \omega_{hb} \\ \dot{\omega}_{hb} &= -\mathbf{K}_{\Theta_h} \Theta_h - \mathbf{K}_{\omega_{hb}} \omega_{hb} + \mathbf{K}_{u_h} \mathbf{u}_h \end{aligned} \quad \xrightarrow{\mathbf{u}_h = -\mathbf{K}'_{v_{hb}} \mathbf{v}_{hb} + \mathbf{u}_{vh}} \quad \begin{aligned} \dot{\mathbf{v}}_{hb} &= -\mathbf{K}_{v_{hb}} \mathbf{v}_{hb} - g \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Theta_h \\ \dot{\Theta}_h &= \omega_{hb} \\ \dot{\omega}_{hb} &= -\mathbf{K}_{u_h} \mathbf{K}'_{v_{hb}} \mathbf{v}_{hb} - \mathbf{K}_{\Theta_h} \Theta_h - \mathbf{K}_{\omega_{hb}} \omega_{hb} + \mathbf{K}_{u_h} \mathbf{u}_{vh} \end{aligned}$$

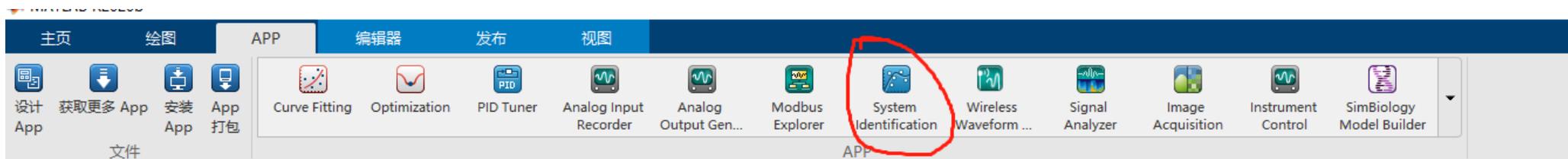
比例控制器

传递函数形式： $\mathbf{v}_{hb}(s) = \mathbf{G}_{v_{hb}} \mathbf{u}_{vh}(s)$

1.基本原理

□ 辨识工具箱

1. **MATLAB系统辨识工具箱**：该工具箱包括了MATLAB 函数、Simulink 函数块以及一个可利用测得的输入/输出数据构造动态系统的应用程序。



2. **CIFER工具箱**：该工具箱是解决复杂的系统辨识问题，可控制器设计相关的系统辨识问题



1. 辨识算法被广泛采用
 2. 采用线性调频Z变换和复合窗口优化技术获得高质量的响应辨识结果；
 3. 基于频率响应精度的全自动加权函数选取
 4. 可靠的参数精度度量准则
-

1.基本原理

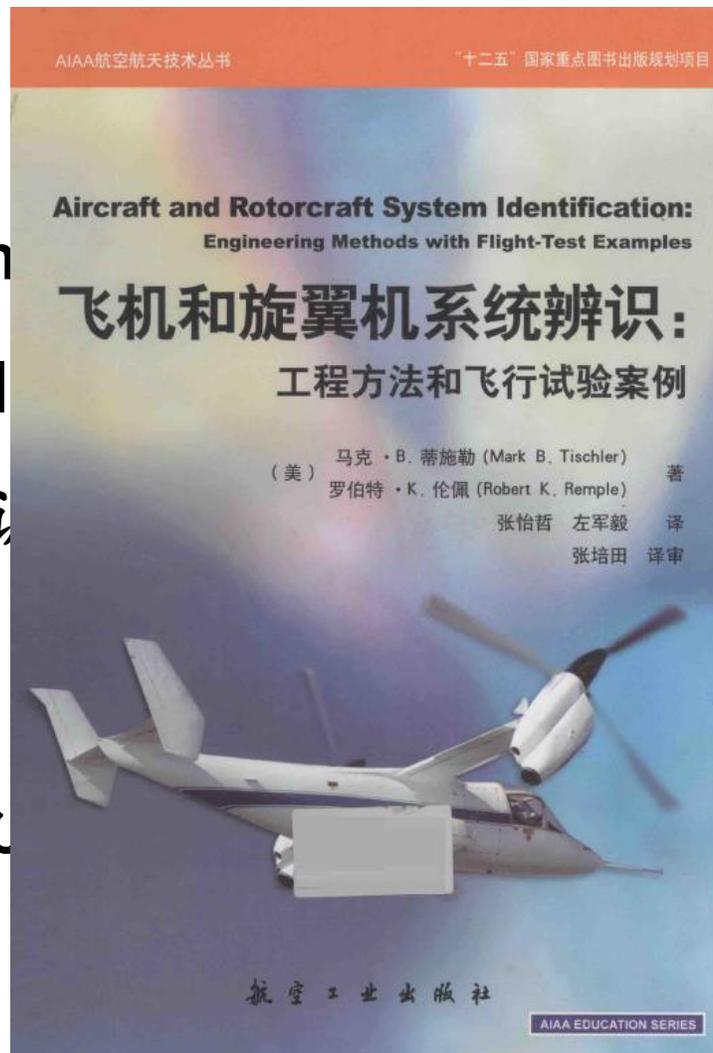


以上原
Multicopter
或者“全权
器设计与控
Tischl
行实验案例

多旋翼飞行器 设计与控制

全权 著
杜光勋 赵峙尧 戴训华 任锦瑞 邓恒 译
郭祥 插图设计

中国工信出版集团 电子工业出版社
http://www.gdpc.com.cn



to
re, 2017”
《多旋翼飞行
工程方法和飞
出版社, 2012.

CIFER软件的获取: <http://uarc.ucsc.edu/flight-control/cifer>

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“e1.1”，CIFER软件及使用文档（见附录A）
- 硬件：计算机

■ 目标

将已给出的**传递函数模型**与**非线性模型**进行**对比**；在外加控制器的情况下，用相同扫频信号对两种模型进行激励，对比仿真结果。具体有如下三个部分：

- 1) 理解与熟悉建模过程；
- 2) 理解与熟悉获得的多旋翼传递函数模型；
- 3) 对系统辨识的传递函数模型和多旋翼非线性模型输入相同的扫频信号，对比其输出结果。

2.基础实验

□ 实验步骤

(1) 步骤一：首先阅读本书第二章的内容，对于Simulink模型有一个初步的了解。

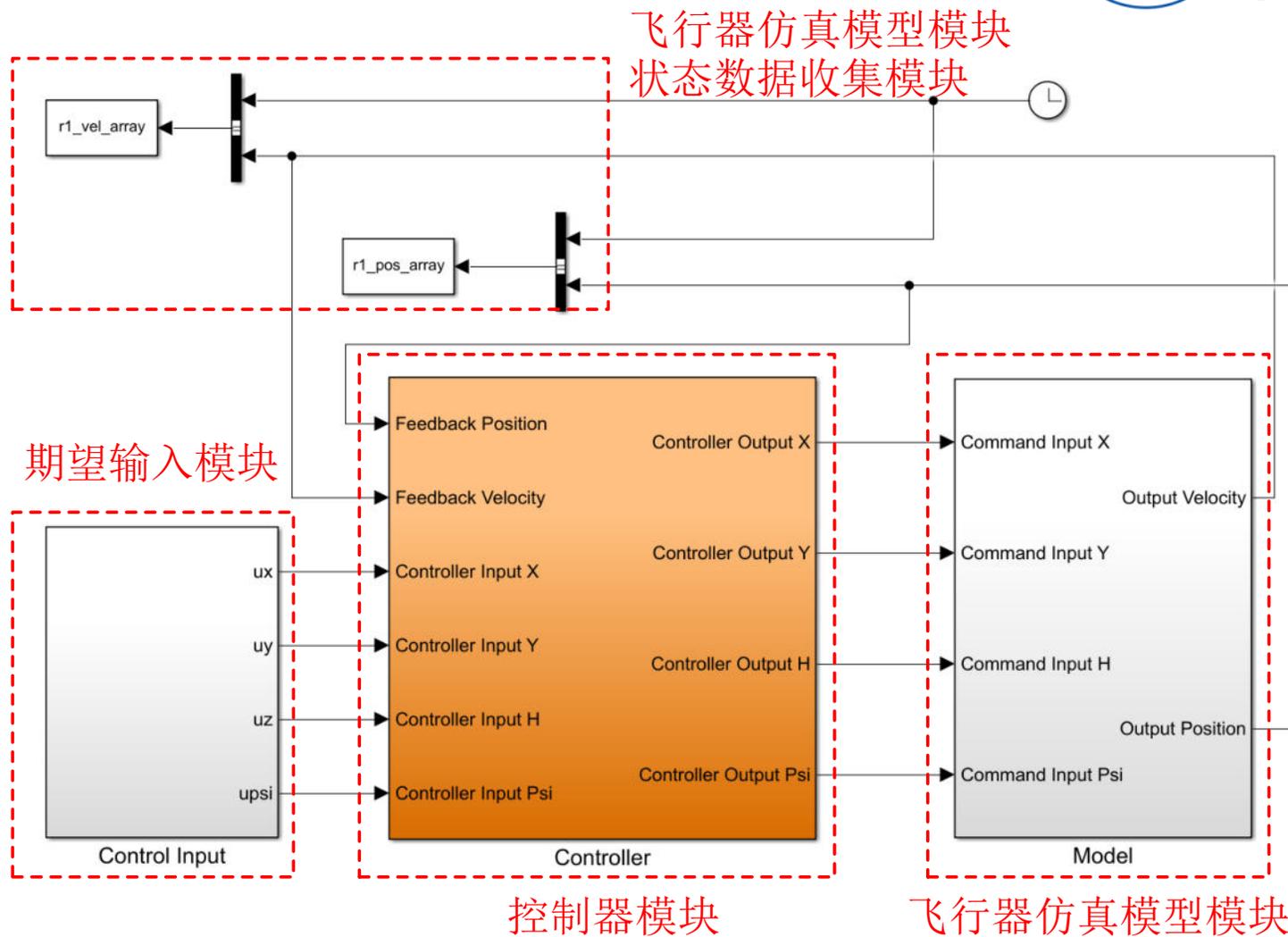
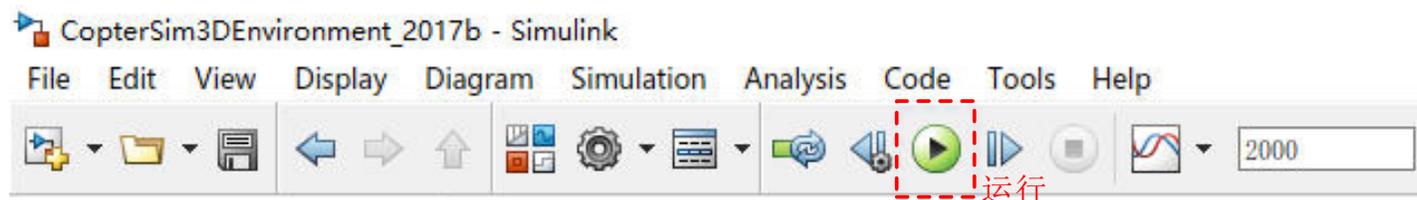


图. Simulink仿真平台

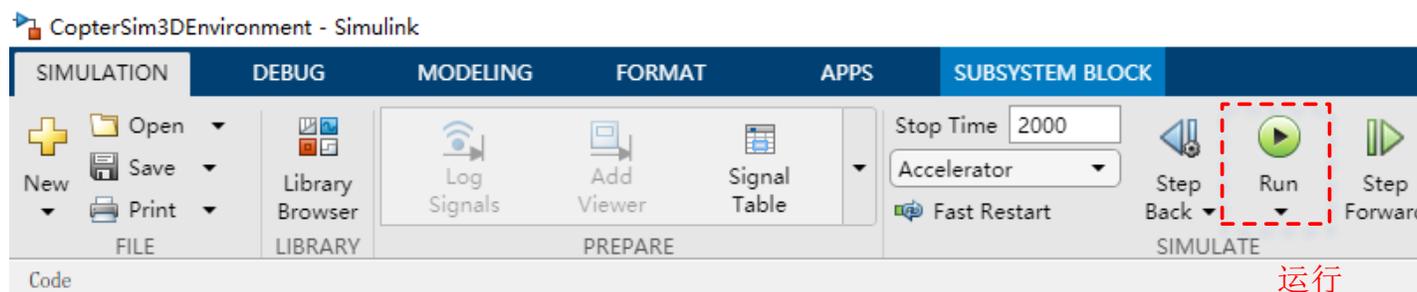
2.基础实验

□ 实验步骤

(2) 步骤二：打开“e1\e1.1\startSimulation.m”文件，点击MATLAB中的“运行”按钮



(a) Simulink “运行”按钮（MATLAB 2017b-2019a）



(b) Simulink “运行”按钮（MATLAB 2019b及更高版本）

图. 不同版本MATLAB的“运行”按钮

2. 基础实验

□ 实验步骤

(3) 步骤三：打开“e1\ e1.1\ e1_1_ModelUnderstanding.slx”文件后，可以看到一个整体模型。

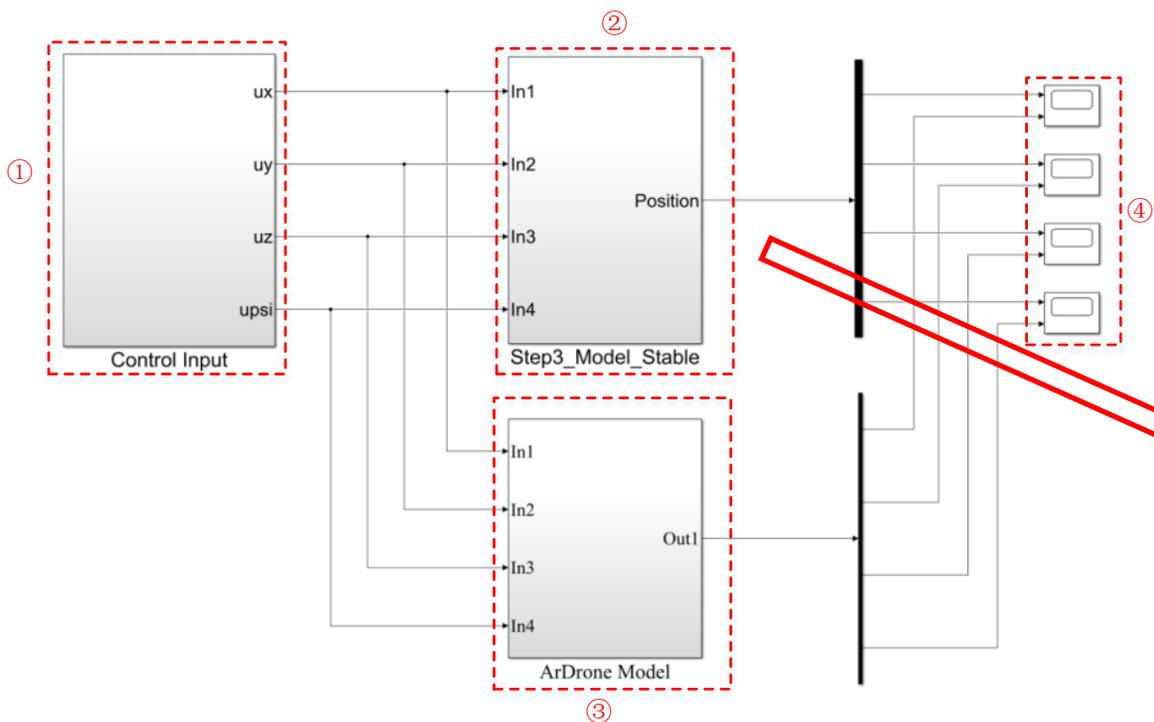


图. 整体模块示意图

- a) 虚线框①第二章2.2小节所提到的各通道位置期望输入模块；
- b) 虚线框②中是传递函数模型，该模型是对虚线框③里面多旋翼非线性模型的辨识结果，里面添加了个“比例-微分”控制器。

水平前向通道：

$$G_{p_{x_e}u_{v_x}} = \frac{6.48651}{s^3 + 3.72616s^2 + 12.6598s + 6.55711} e^{-0.0429s}$$

水平侧向通道：

$$G_{p_{y_e}u_{v_y}} = \frac{6.48638}{s^3 + 3.72607s^2 + 12.66s + 6.55534} e^{-0.0429s}$$

高度通道：

$$G_{p_{z_e}u_T} = \frac{12.275}{s^2 + 13.0573s + 12.3932} e^{-0.0271s}$$

偏航通道：

$$G_{\psi u_{\omega_z}} = \frac{5.28369}{s^2 + 5.07814s + 5.37552} e^{-0.0281s}$$

2.基础实验

□ 实验步骤

(3) 步骤三：打开“e1\e1.1\e1_1_ModelUnderstanding.slx”文件后，可以看到一个整体模型。

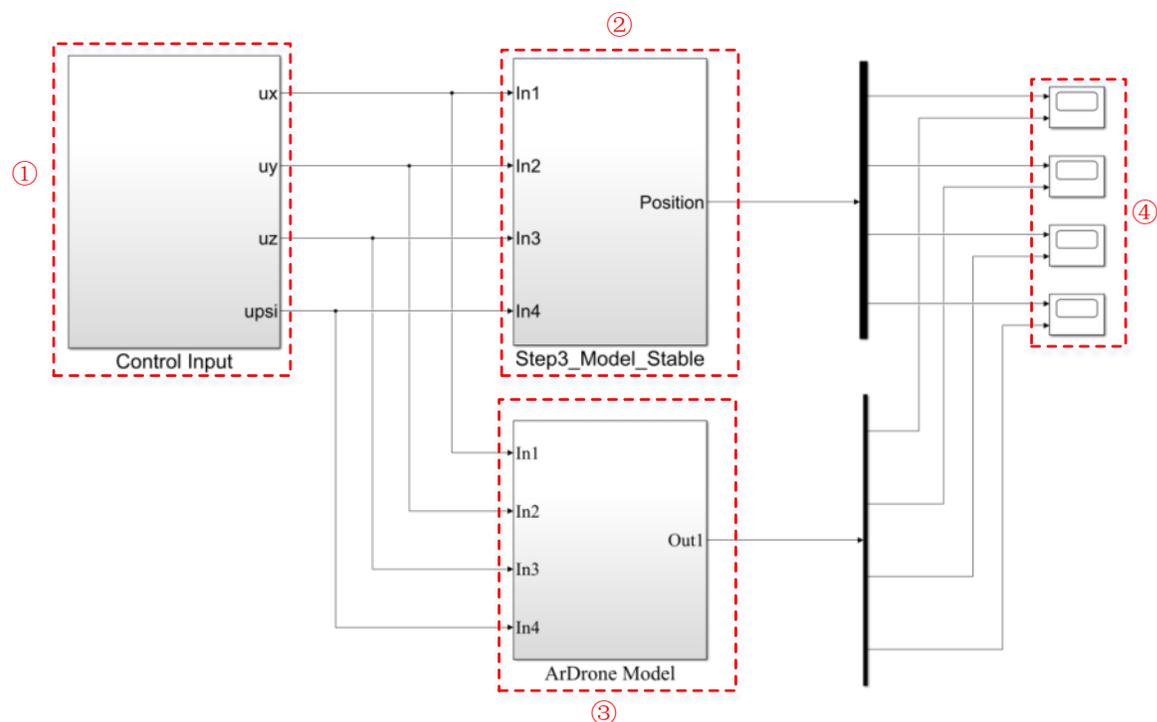


图. 整体模块示意图

- c) 虚线框③中的多旋翼模型则是多旋翼的非线性模型；
- b) 虚线框④为显示模块，可以显示虚线框②和虚线框③两种模型中相同通道反馈的实时多旋翼位置信息的对比。

2. 基础实验

□ 实验步骤

(4) 了解该模型文件之后，打开期望输入模块（虚线框①）。在该模块中，如果需要修改相应的各通道期望扫频输入，则需要打开虚线框①所示模块。从上到下四个模块依次为四个通道的扫频输入，以最上边的水平前向通道“X Swept”模块为例。

①不同通道加入扫频信号

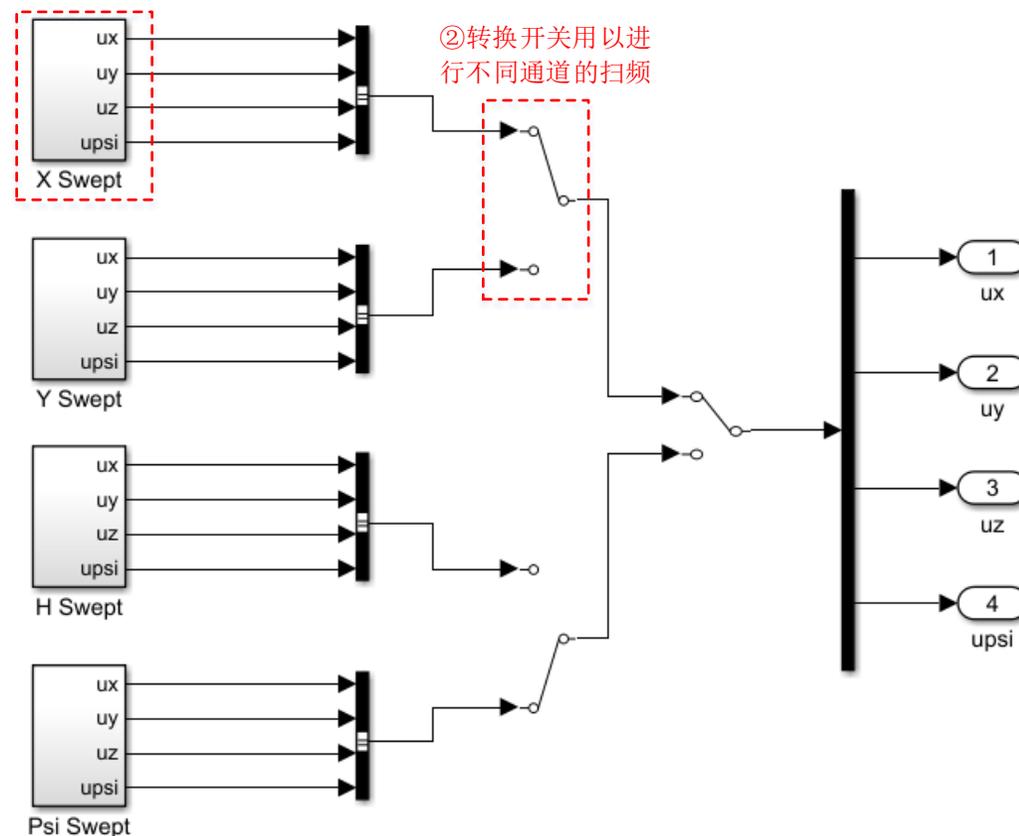


图. 期望输入模块示意图

2. 基础实验

实验步骤

①不同通道加入扫频信号

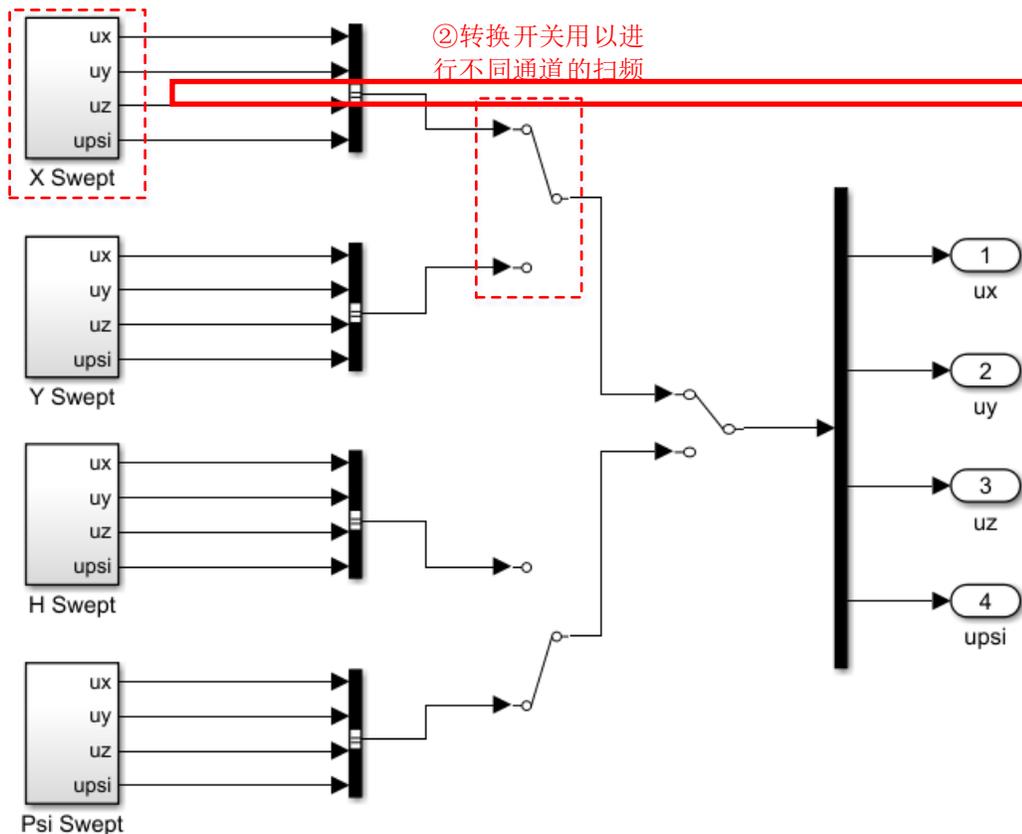
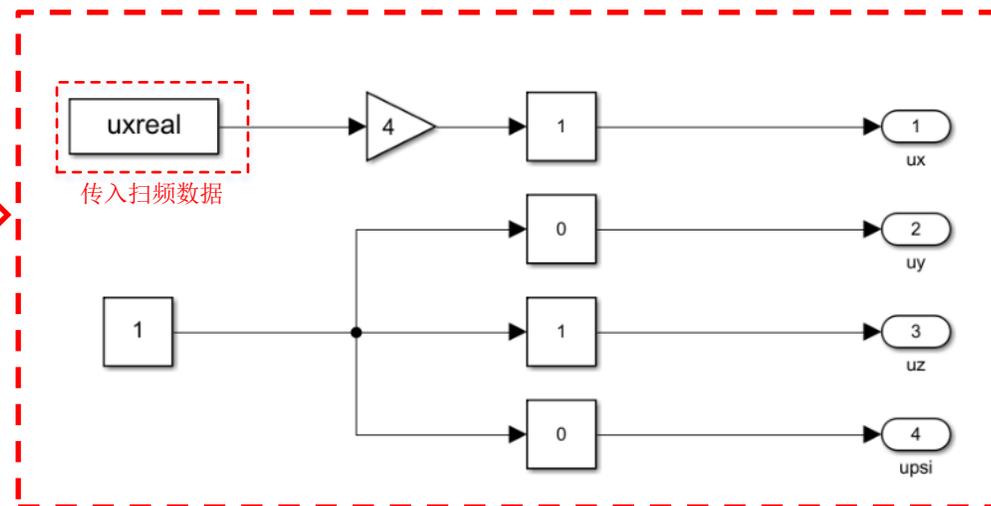


图. 期望输入模块示意图



这里需要注意的是，进行扫频时不可以同时进行多通道扫频，需要单个通道依次扫频。这时，需要相应调节左图虚线框②所示的转换开关，并且除需要扫频的通道外，其它通道均设置为稳定值，即多旋翼在该通道的“平衡点”附近。点击Simulink中“Run”按钮运行该模型，并通过多旋翼非线性模型观察仿真结果。

2.基础实验

□ 结果分析

在对各个通道进行扫频信号激励之后，这里以水平通道为例，其结果如下图所示。

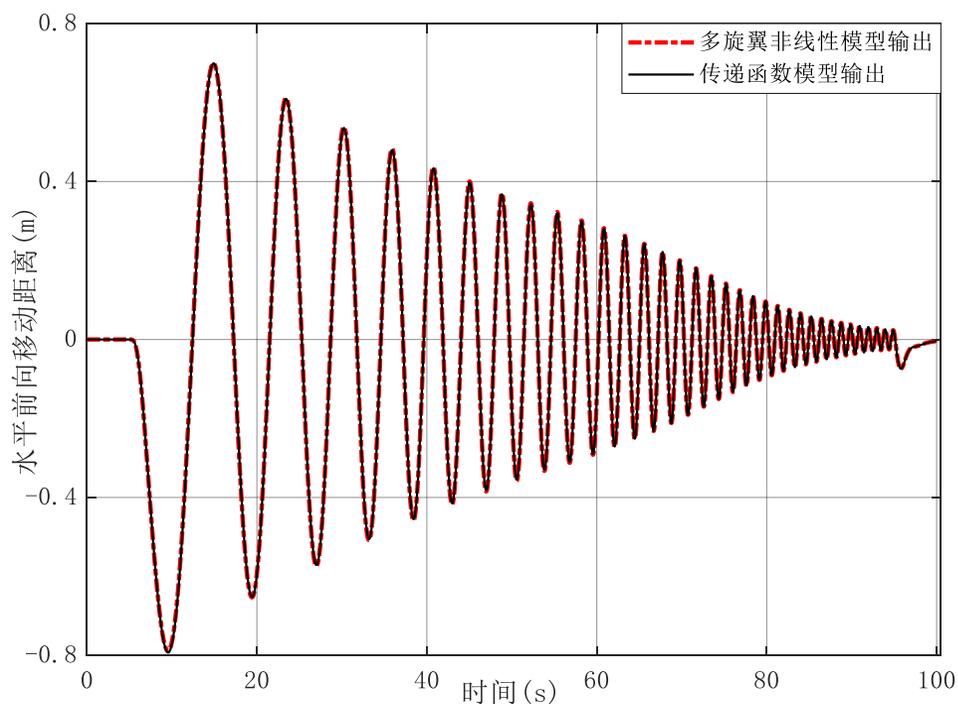


图. 水平前向通道对比图

其通道扫频响应的峰值均为1，激励时间长度为100s。由此可以看出，利用传递函数获得的输出与多旋翼非线性模型获得的输出**几乎是一致的**。当给定的扫频信号的扫频频率范围在0.1~1Hz，即**低频**时，两者的输出**近似程度更高**；而在**高频**时，扫频输出的幅值出现了**差异**，但是仍然可以吻合。因此，**该传递函数模型可以较为完整地反映系统的真实性能**。对于其它通道，读者可以自己实验操作验证。

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“e1.2”，CIFER软件及使用文档（见附录A）
- 硬件：计算机

■ 目标

对给定的多旋翼非线性模型进行辨识，在假设存在时延的情况下获得四个通道的传递函数模型。注意四个通道建立的均是由**输入**到**机体速度**的传递函数，这里将水平通道的两个方向分解为水平前向通道和水平侧向通道，因此是四通道的。

- 1) 了解利用**CIFER软件**进行系统辨识对系统进行建模的大致流程。
- 2) 要求各通道辨识结果的**代价函数 $J < 50$** ，详情参考“CIFER使用指南”（见附录A）。
- 3) 对于通过系统辨识获得的**速度模型传递函数**，加入**积分**环节，使之成为完整的多旋翼机体模型传递函数，即输入到**机体位置的传递函数模型**。

3.分析实验

□ 实验步骤：仿真1.0

(1) 步骤一：了解辨识相关流程

首先阅读本书第二章的内容，对于Simulink模型有一个初步的了解。

阅读“CIFER使用指南”（见附录A），学习如何进行系统辨识。

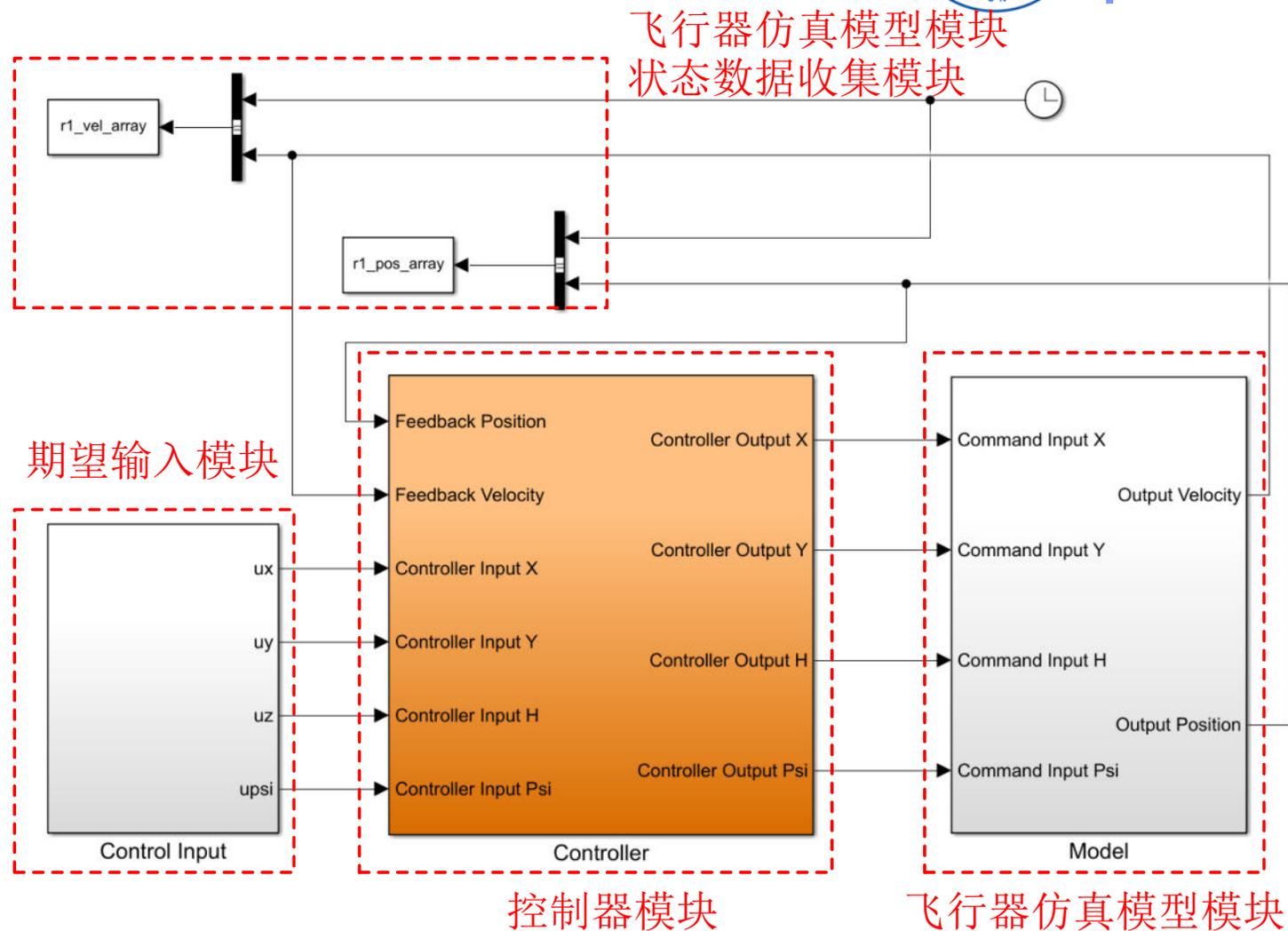


图. Simulink仿真平台

3.分析实验

□ 实验步骤：仿真1.0

(2) 步骤二：扫频相关文档介绍

打开文件夹“e1”，该文件夹内容如下：1) 文件夹“e1\Data_SystemIdentification”存放有所需要的系统辨识扫频数据，按照扫频数据的初始频率不同分为了三个文件夹，每个文件夹中前缀为“F434loaddata”的mat数据文件加载后，**作为辨识过程的期望输入**（例如：

“F434loaddata_psi_0.08_1.18.mat”代表扫频信号频率范围是0.08Hz~1.18Hz，时间历程的详细介绍请参考附录A）；前缀为“F434test”的mat数据文件加载后**用于仿真实验中**，CIFER软件的扫频数据输入（例如：“F434test_psi_0.08_1.18.mat”代表扫频信号频率范围是0.08Hz ~ 1.18Hz得到的时间历程文件数据

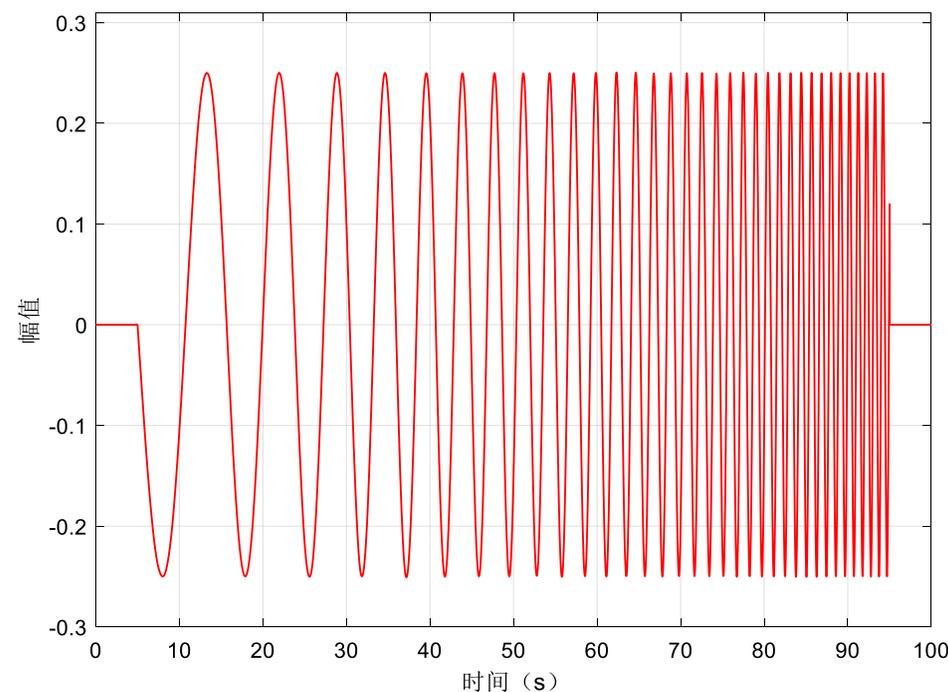


3.分析实验

□ 实验步骤：仿真1.0

(3) 步骤三：初始化参数

打开“startSimulation.m”初始化文件。在其中可以找到右图所示代码。与前面所叙述的一致，该段代码中有“加载仿真文件数据.mat”和“保存时间历程文件.mat”两个部分，分别用于仿真的**扫频输入**和保存时间历程文件（详情请参考附录A）所需要的数据，在使用不同频率范围的扫频数据之前，需要同时将对应的代码去除注释标记并运行。



扫频输入

3.分析实验

□ 实验步骤：仿真1.0

(3) 步骤三：初始化参数

打开“startSimulation.m”初始化文件。在其中可以找到右图所示代码。与前面所叙述的一致，该段代码中有“加载仿真文件数据.mat”和“保存时间历程文件.mat”两个部分，分别用于仿真的扫频输入和保存**时间历程文件**（详情请参考附录A）所需要的数据，在使用不同频率范围的扫频数据之前，需要同时将对应的代码去除注释标记并运行。

时序名称: =

时间	数据:1
5.5200	-0.0647
5.5300	-0.0659
5.5400	-0.0671
5.5500	-0.0684
5.5600	-0.0696
5.5700	-0.0708
5.5800	-0.0720
5.5900	-0.0733
5.6000	-0.0745
5.6100	-0.0757

时间历程文件数据：指代包括“时间”和“输入”（或者“输出”）一一对应的数据文件

图. 时间历程文件数据



3.分析实验

□ 实验步骤：仿真1.0

(3) 步骤三：初始化参数

打开“startSimulation.m”初始化文件。在其中可以找到右图所示代码。与前面所叙述的一致，该段代码中有“加载仿真文件数据.mat”和“保存时间历程文件.mat”两个部分，分别用于仿真的扫频输入和保存时间历程文件（详情请参考附录A）所需要的数据，在使用不同频率范围的扫频数据之前，需要同时将对应的代码去除**注释标记**并运行。

%% 加载仿真文件数据

```
load('0.08/F434test_psi_0.08_1.18.mat')
load('0.08/F434test_x_0.08_1.18.mat')
load('0.08/F434test_y_0.08_1.18.mat')
load('0.08/F434test_z_0.08_1.18.mat')
```

载入用于扫频的数据

```
% load('0.09/F434test_psi_0.09_1.21.mat')
% load('0.09/F434test_x_0.09_1.21.mat')
% load('0.09/F434test_y_0.09_1.21.mat')
% load('0.09/F434test_z_0.09_1.21.mat')
```

注释的代码

```
% load('1.11/F434test_psi_0.11_1.25.mat')
% load('1.11/F434test_x_0.11_1.25.mat')
% load('1.11/F434test_y_0.11_1.25.mat')
% load('1.11/F434test_z_0.11_1.25.mat')
```

%% 保存时间历程文件.mat

```
load('0.08/F434loaddata_psi_0.08_1.18.mat')
load('0.08/F434loaddata_x_0.08_1.18.mat')
load('0.08/F434loaddata_y_0.08_1.18.mat')
load('0.08/F434loaddata_z_0.08_1.18.mat')
```

载入用以保存输入输出结果的数据

```
% load('0.09/F434loaddata_psi_0.09_1.21.mat')
% load('0.09/F434loaddata_x_0.09_1.21.mat')
% load('0.09/F434loaddata_y_0.09_1.21.mat')
```

3. 分析实验

□ 实验步骤：仿真1.0

(4) 步骤四：进行扫频实验

在完成上一步之后，运行“startSimulation.m”初始化文件，初始化实验参数。

(5) 步骤五：修改相应扫频通道

打开模型文件“e1_2_SystemIdentification_Vel.slx”，并在“Control Input”模块中修改相应的扫频通道信息。

①不同通道加入扫频信号

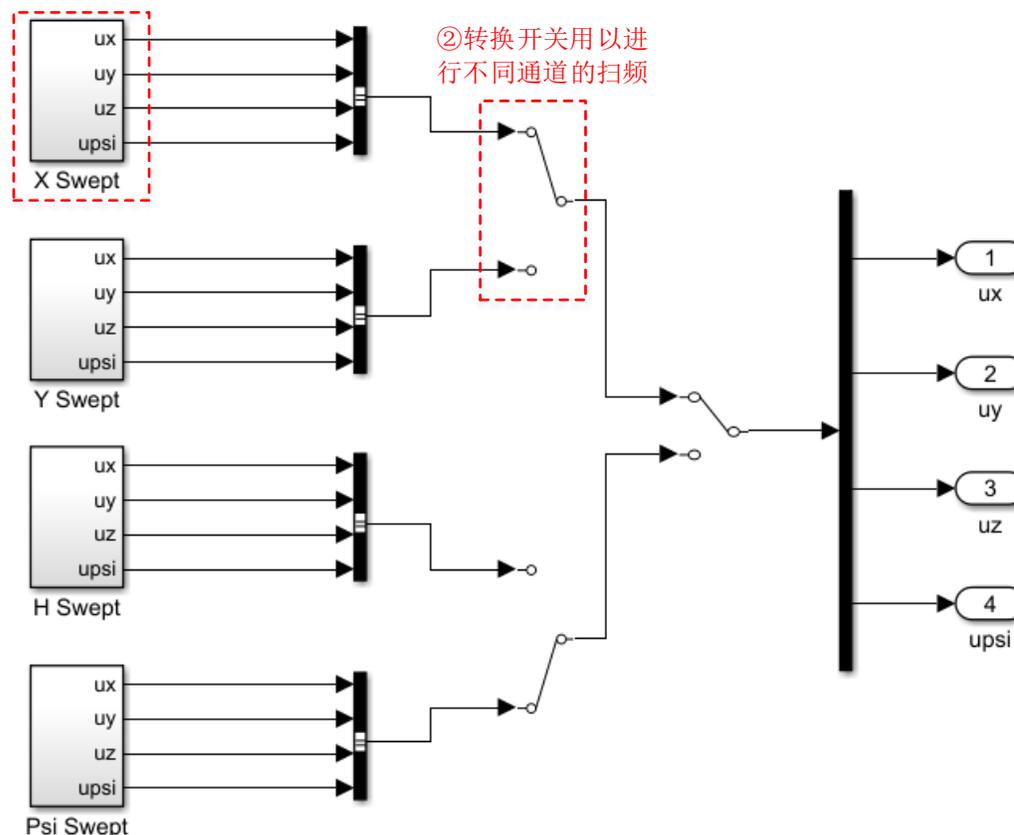


图. 期望输入模块“Control Input”，Simulink 模型“e1_1_ModelUnderstanding.slx”

3.分析实验

□ 实验步骤：仿真1.0

(6) 步骤六：各个通道分别扫频

模型文件

“e1_2_SystemIdentification_Vel.slx” 是主要的实验模型，该模型中可以按通道对多旋翼非线性模型进行扫频，同时获得实时的多旋翼非线性模型速度以及位置信息。因此，在需要辨识某个通道时，需要根据步骤五选择扫频相应扫频通道，将其中的“Switch”开关拨到相应位置，进行扫频即可。

①不同通道加入扫频信号

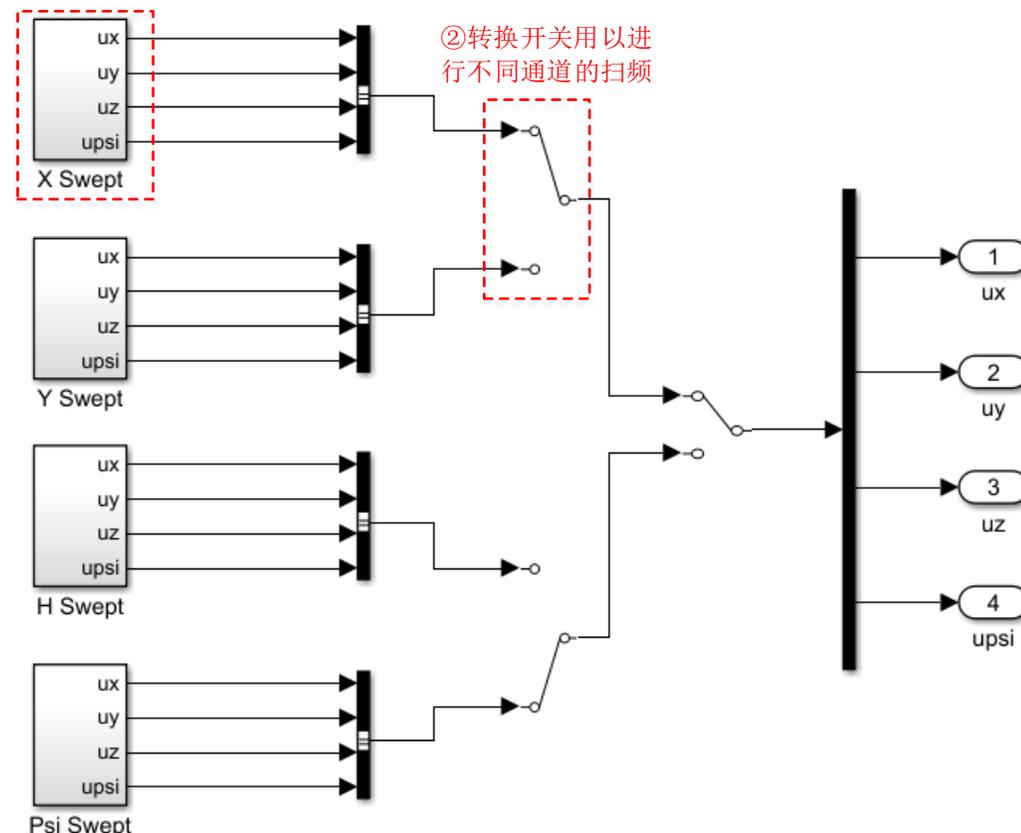


图. 期望输入模块“Control Input”，Simulink 模型“e1_1_ModelUnderstanding.slx”

3.分析实验

□ 实验步骤：仿真1.0

(5) 步骤七：保存数据

“e1_2_SystemIdentification_Vel.slx”，并在“Control

在每个通道仿真完毕后，需要

保存数据。保存数据需要打开

“e1_2_GetFinalData.m”文件，

其代码如右图所示。

生成频率范围为0.08Hz ~ 1.18Hz的期望输入
“ux”-速度输出“outputvx”时间历程文件。
读者需要注释不必要的代码，保存符合本次实
验通道和频率的数据文件，注意这里的**频率范
围**需要保持和**步骤二**的注释相对应。

```
%% 保存在.mat文件的数据

% CIPHER要求变量名最多8个字符（不带下划线）

% save data_x_0.08_1.18.mat time ux outputx;
% save data_y_0.08_1.18.mat time uy outputy;
% save data_z_0.08_1.18.mat time uh outputh;
% save data_psi_0.08_1.18.mat time upsi outputps;

% save data_x_0.09_1.21.mat time ux outputx;
% save data_y_0.09_1.21.mat time uy outputy;
% save data_z_0.09_1.21.mat time uh outputh;
% save data_psi_0.09_1.21.mat time upsi outputps;

% save data_x_0.11_1.25.mat time ux outputx;
% save data_y_0.11_1.25.mat time uy outputy;
% save data_z_0.11_1.25.mat time uh outputh;
% save data_psi_0.11_1.25.mat time upsi outputps;

% CIPHER要求变量名最多8个字符（不带下划线）

save data_vx_0.08_1.18.mat time ux outputvx;
```

```
% save data_vy_0.08_1.18.mat time uy outputvy;
% save data_vz_0.08_1.18.mat time uh outputvh;
% save data_vpsi_0.08_1.18.mat time upsi outputvp;
```

```
%% 加载数据区域

% 存储期望位置数据
% ux=4*uxreal_array(:,1);
% uy=4*uyreal_array(:,1);
% uh=4*(uhreal_array(:,1)-1)+1;
% upsi=4*upsireal_array(:,1);

% 存储位置反馈数据
% outputx=r1_pos_array(:,2);
% outputy=r1_pos_array(:,3);
% outputh=r1_pos_array(:,4);
% outputps=r1_pos_array(:,5);

% 存储期望速度数据
ux=uxreal_array(:,1);
uy=uyreal_array(:,1);
uh=1-uhreal_array(:,1);
upsi=upsireal_array(:,1);

% 存储速度反馈数据
outputvx=r1_vel_array(:,2);
outputvy=r1_vel_array(:,3);
outputvh=r1_vel_array(:,4);
outputvp=r1_vel_array(:,5);

%% 保存在.mat文件的数据
```



3.分析实验

□ 实验步骤：仿真1.0

(8) 步骤八：进行系统辨识，获得相应结果

保存三组不同初始频率下的各通道扫频输入输出数据，将其复制到CIFER软件的目录下，进行系统辨识，获得相应的结果。

(9) 步骤九：进行非线性模型对比验证

在获得了四个通道各自的传递函数模型之后，运行“e1\e1.2\ e1_2_SystemIdentification_Vel_Verification.slx”模型文件验证所获得的传递函数模型的效果。

3.分析实验

□ CIFER软件辨识步骤

(1) 步骤一：建立数据库Database

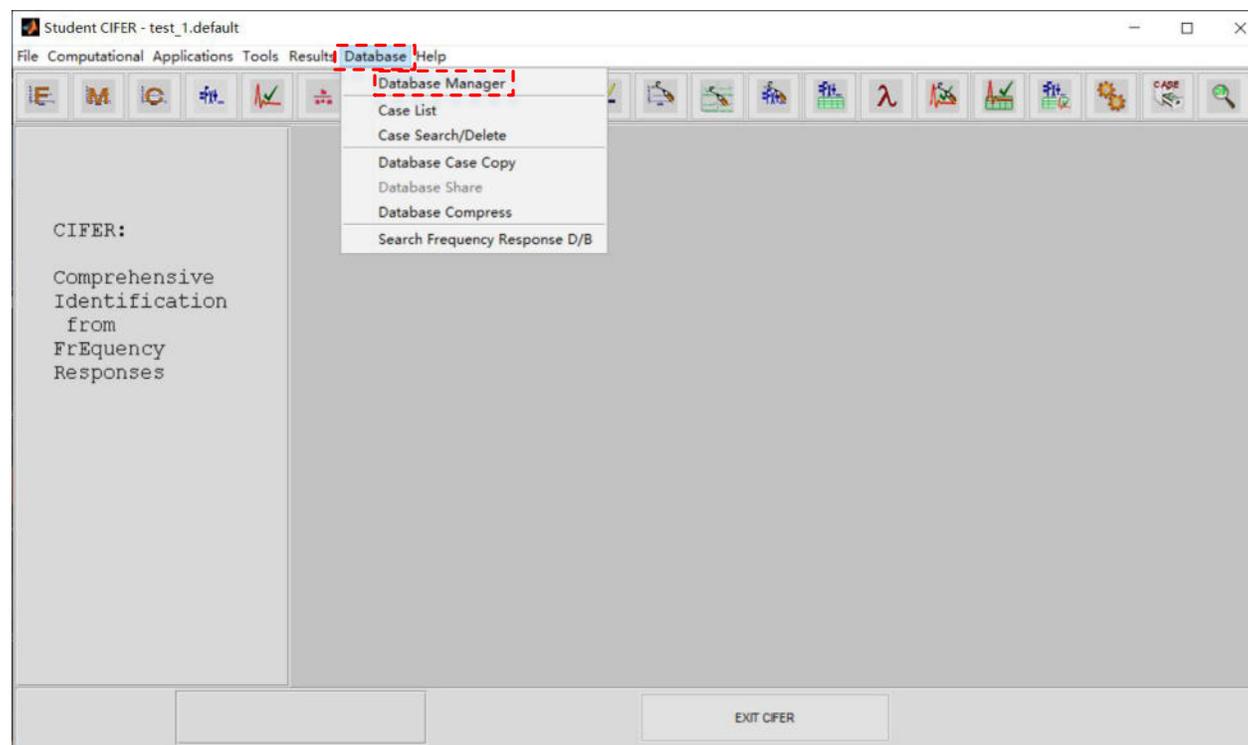


图. CIFER 软件用户主界面

3.分析实验

□ CIFER软件辨识步骤

(2) 步骤二：频域响应辨识模块FRESPID

“FRESPID”模块主要功能是将所获得的时域数据（时间历程数据）转化为频域数据，即输入/输出数据的频域响应

针对单输入单输出系统而言！

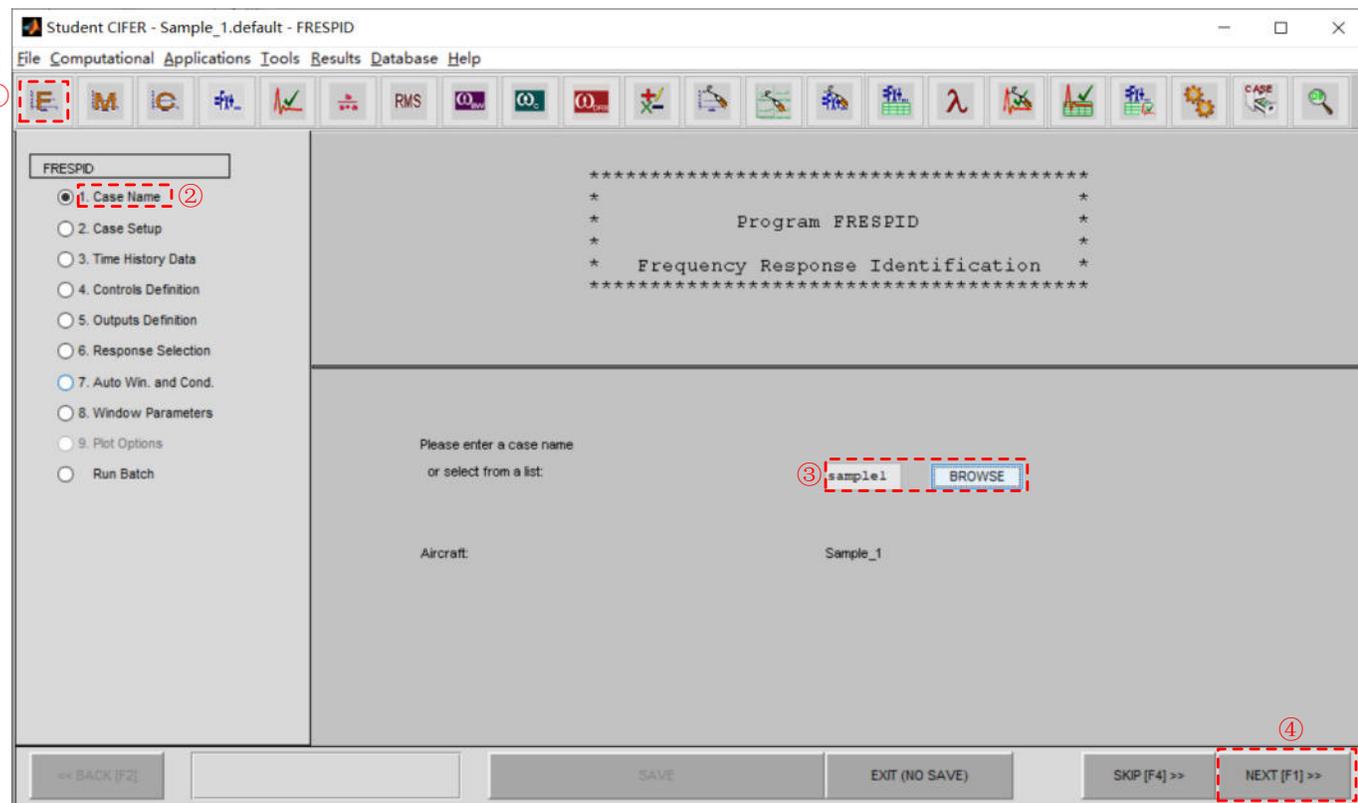


图. “FRESPID” 用户主界面

3.分析实验

□ CIFER软件辨识步骤

(2) 步骤二：频域响应辨识模块：
FRESPID——“Auto Win. And Cond.”
频率参数设置模块

滤波器截止频率，选取值至少应
为时间历程文件中所使用的最大频率
(或者说模型应用的最大可能频率)
的**五倍**。

例如时间历程文件中所使用的最大
频率为1Hz，那么这一栏设置频率
应该大于 $1\text{Hz} * 5 = 5\text{Hz}$ 。

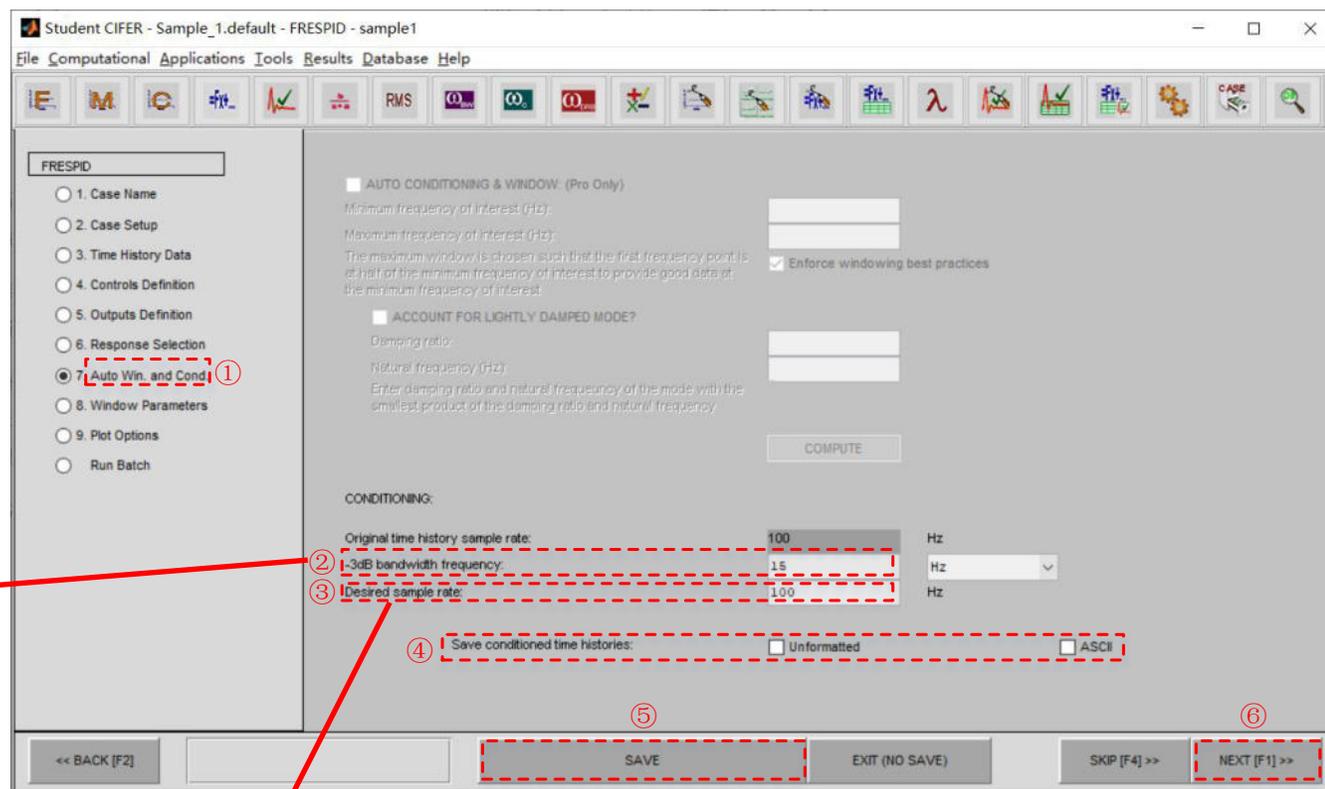


图. “Auto Win. And Cond.” 用户显示界面
大于带宽频率的5倍

3.分析实验

□ CIFER软件辨识步骤

(2) 步骤二：频域响应辨识模块： FRESPID——“Windows Parameter” 抽取的时间历程数据长度模块

1. 最佳窗口尺寸：使用扫频信号中最小扫频信号对应周期的两倍

2. 最大的窗口尺寸应该小于单个时间历程数据文件的时间长度的一半，也应该小于所有时间历程数据连接之后总长度的1/5，取两者中的较小值。

3. 最小的窗口尺寸应该大于所使用扫频信号最大频率（最大感兴趣频率）所代表的周期的二十倍。

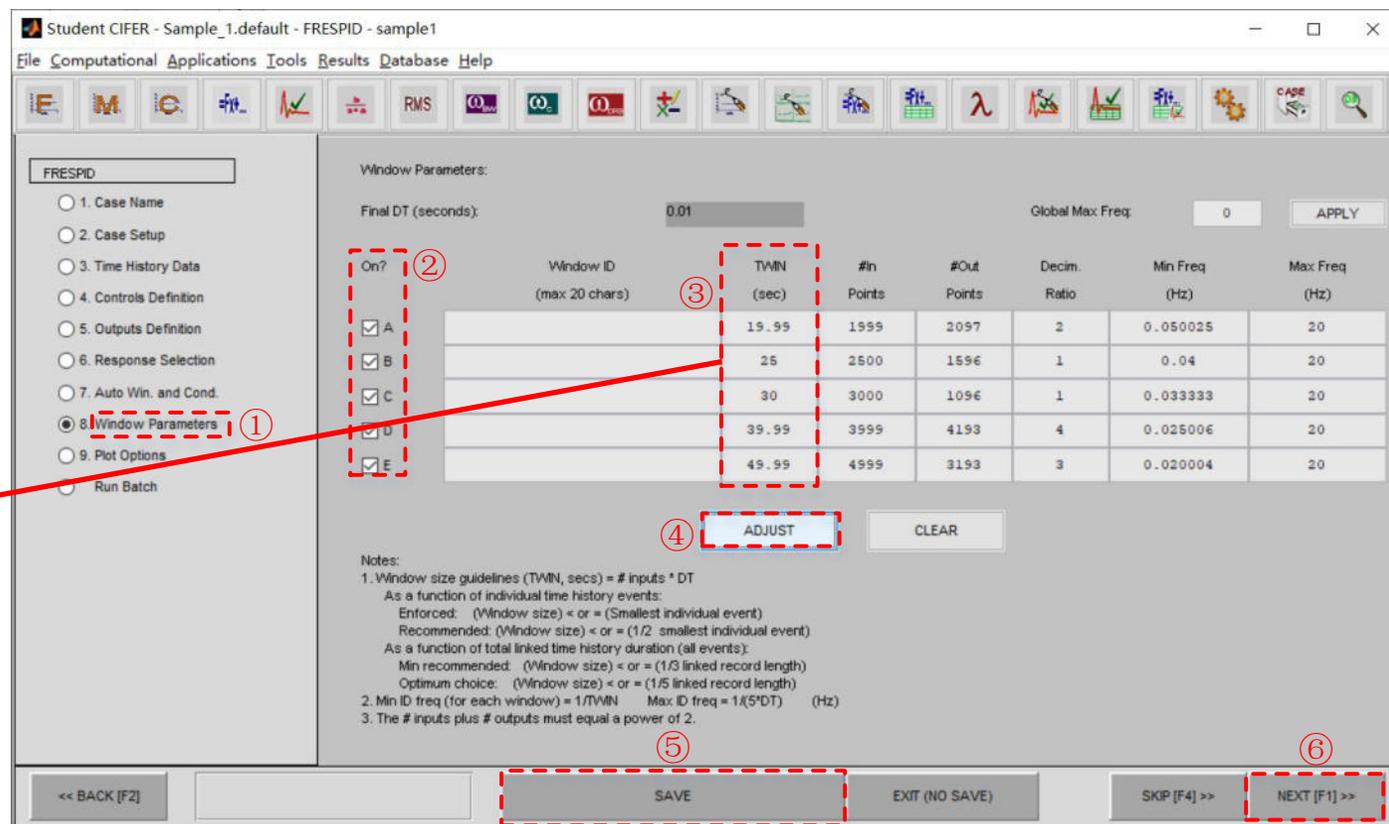


图. “Windows Parameter” 用户显示界面

3.分析实验

□ CIFER软件辨识步骤

(3) 步骤三：复合分窗模块：COMPOSITE

复合分窗模块用来将之前设置的各

个分窗通过加权理论结合在一起将各自的优点结合起来，用以削弱大尺寸分窗与小尺寸分窗各自的缺点，提高辨识的准确度。

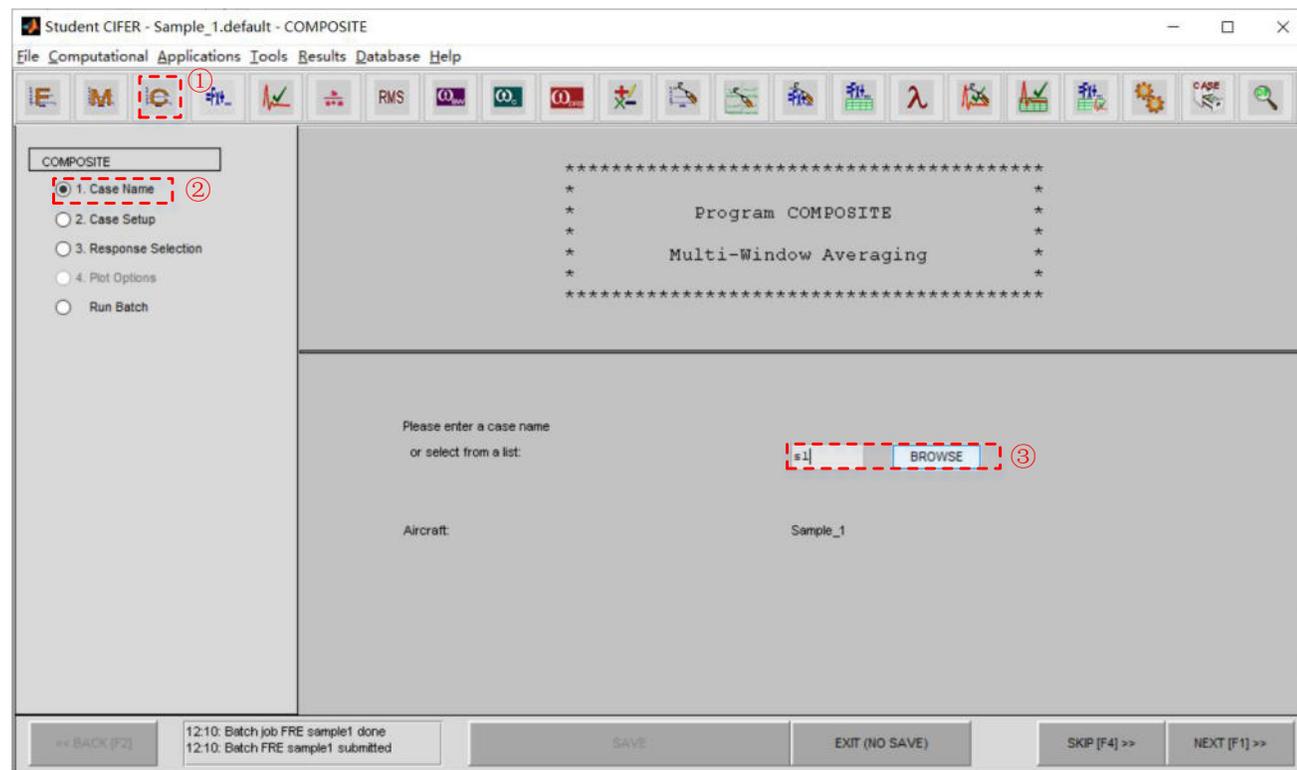


图. “Composite” 模块中的 “Case Name” 用户界面

3.分析实验

□ CIFER软件辨识步骤

(4) 步骤四：传递函数模型建立模块：NAVFIT

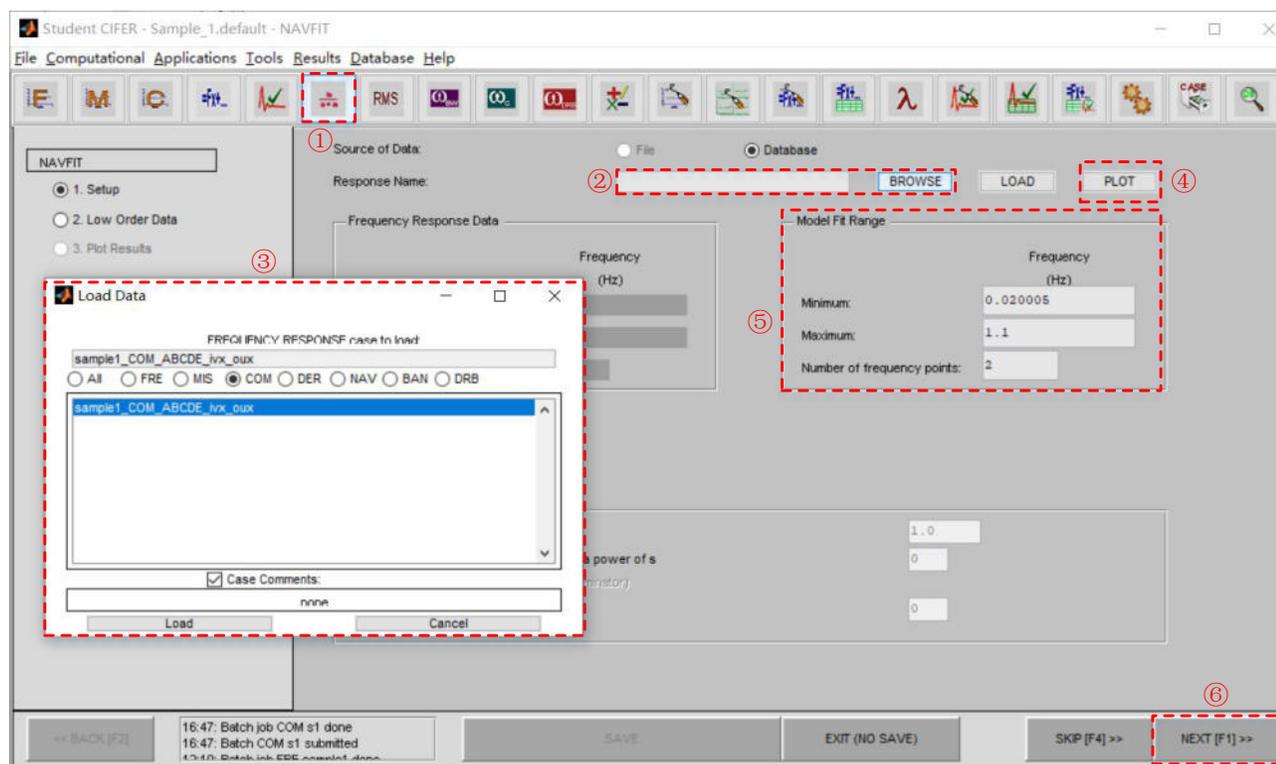


图. “Navfit”模块中的“Setup”用户界面

第四章系统辨识实验

分析实验：仿真1.0

3.分析实验

□ 实验步骤：仿真2.0

打开附带的“e1\ e1.2\ e1_2_SystemIdentification_Vel_Verification.slx”模型文件，其内部下图所示。

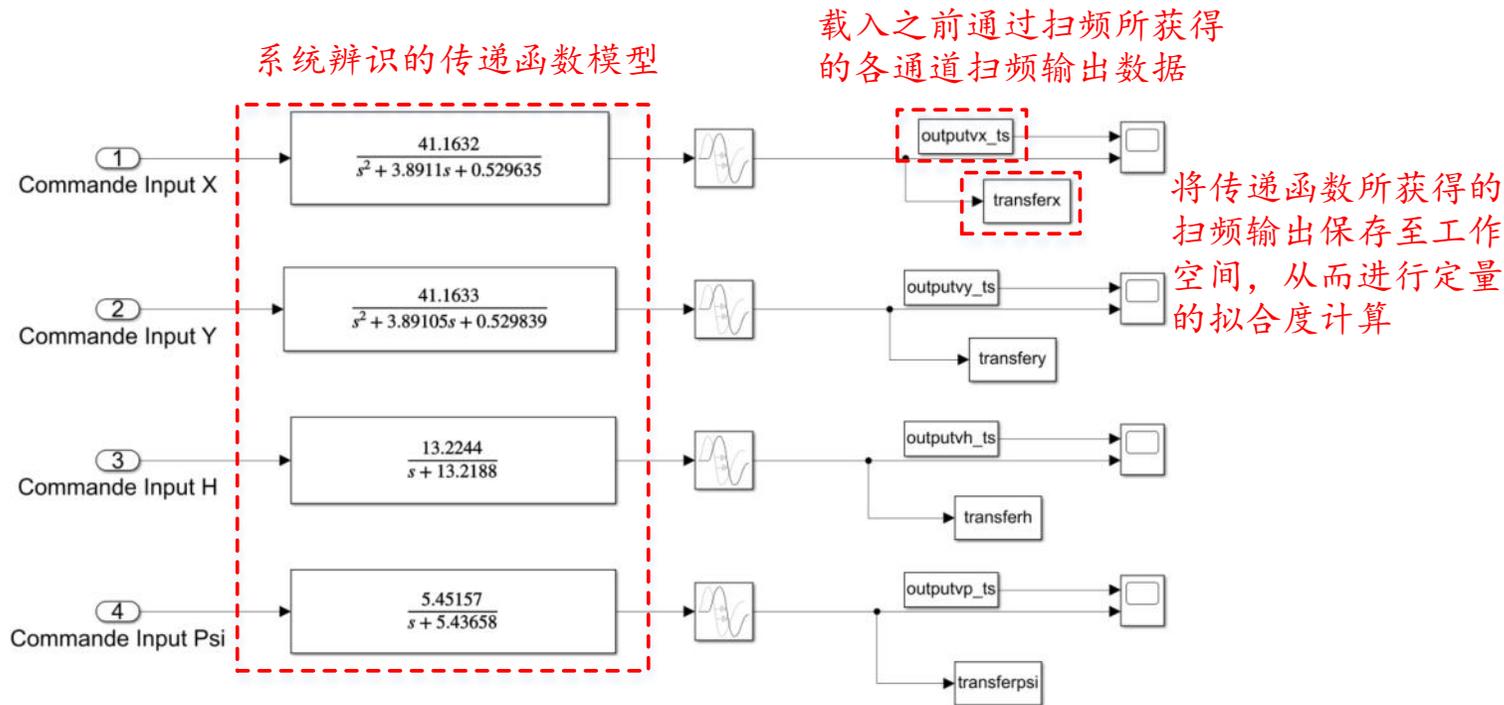


图. 扫频输入验证模型示意图，Simulink 文件“e1_2_SystemIdentification_Vel_Verification.slx”

需要注意到，如果要正确运行这里的验证模型，需要进行一些数据的载入与转换，运行“e1\ e1.2\ e1_2_SystemIdentification_Vel_Verification_Corr2.m”文件；接下来运行“e1_2_SystemIdentification_Vel_Verification.slx”模型验证文件；之后，可以在示波器中查看相应的对比结果，也能够利用“corr2”函数获得定量的拟合度数据。

最后，需要对获得的传递函数模型进行验证。验证思路是对传递函数模型以及多旋翼非线性模型进行相同信号输入，比较二者所获得的输出结果是否一致。如果二者结果一致，同时CIFER辨识获得的代价函数指标也较小，可以认为辨识成功。对于输出结果的比较，可以使用MATLAB内置的“corr2”函数，该函数可以对两个列向量进行比较，实际使用时可以将两个输出结果传入工作空间，再将二者利用该函数进行比较。比较结果越接近于100%则表明进行比较的二者越相似。

3.分析实验

□ 结果分析

非线性模型以及传递函数模型各个通道的扫频输出结果对比分析如下所示。

(1) 水平前向通道

$$G_{v_{x_b} u_{v_x}} = \frac{41.1632}{s^2 + 3.8911s + 0.529635} e^{-0.0289s} \quad J = 0.267$$

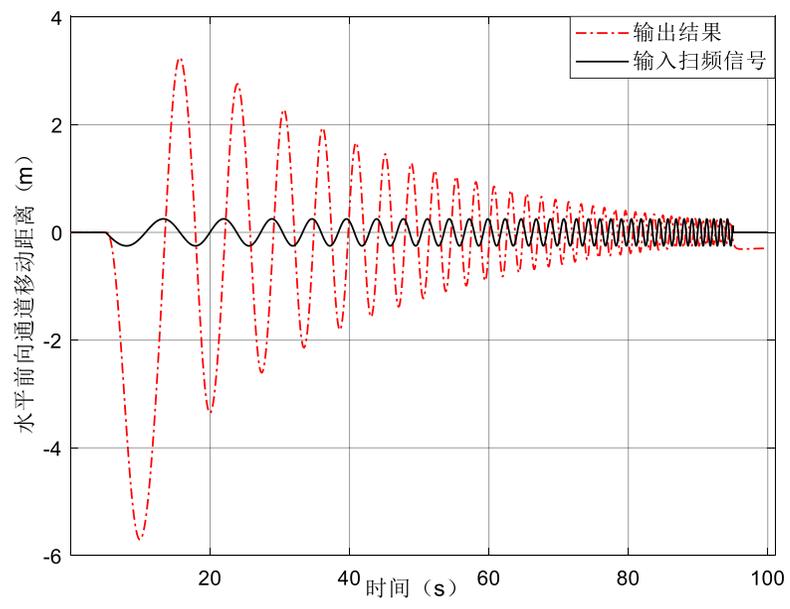


图. 水平前向通道输入输出对比图

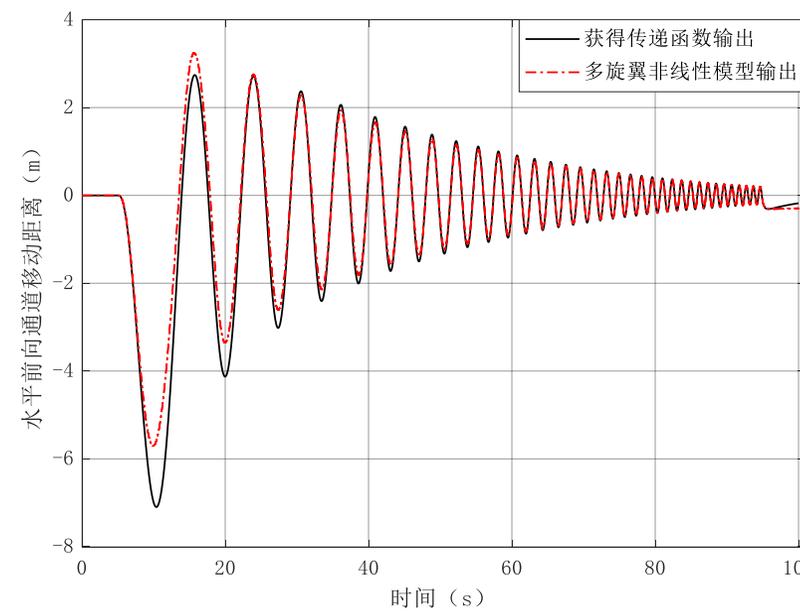


图. 水平前向通道验证对比图

3.分析实验

□ 结果分析

非线性模型以及传递函数模型各个通道的扫频输出结果对比分析如下所示。

(2) 水平侧向通道

$$G_{y_b u_{vy}} = \frac{41.1633}{s^2 + 3.89105s + 0.529839} e^{-0.0289s} \quad J = 0.267$$

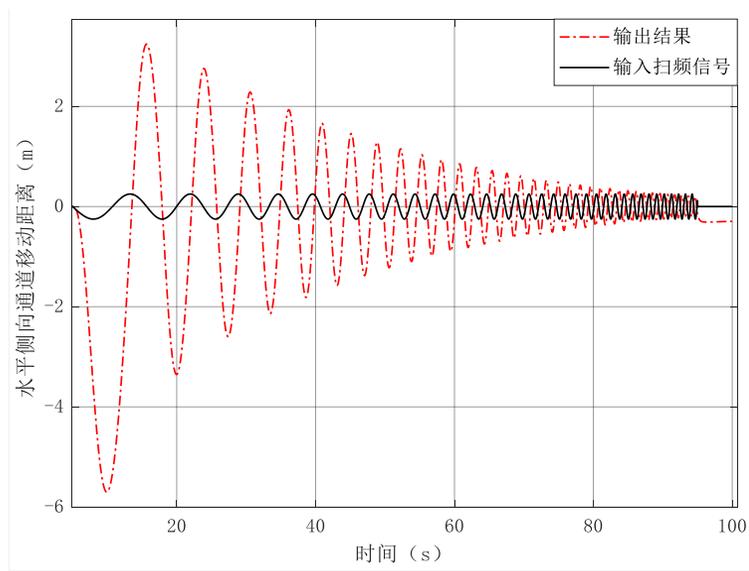


图. 水平侧向通道输入输出对比图

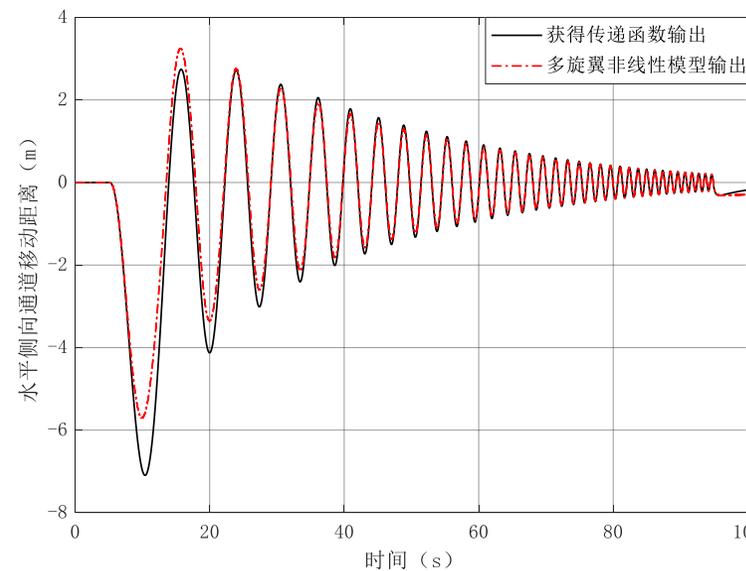


图. 水平侧向通道验证对比图

3.分析实验

□ 结果分析

对于本实验中的水平通道模型来说，需要获得一个由遥控器输入到地球坐标系下位置输出的模型，而之前所获得的仅仅是一个由遥控器输入到机体速度输出的模型。因此，为了要获得完整系统模型的传递函数，需要加入由速度到位置的积分环节。在加入积分环节和偏航相关旋转矩阵之后，水平通道通道传递函数可表示如下

$$\begin{bmatrix} G_{p_x e u_{vx}} \\ G_{p_y e u_{vy}} \end{bmatrix} = \mathbf{R}_\psi \begin{bmatrix} \frac{41.1632}{s^3+3.8911s^2+0.529635s} e^{-0.0289s} \\ \frac{41.1633}{s^3+3.89105s^2+0.529839s} e^{-0.0289s} \end{bmatrix}$$

水平通道获得的传递函数模型**代价函数值**均较小，**都小于0.3**；经过验证获得的传递函数与多旋翼非线性模型的**匹配度**也非常高，**均大于98.00%**。可以认为辨识是成功的。

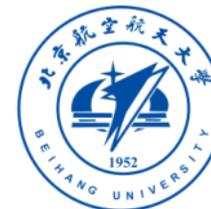
第四章系统辨识实验

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“e1.3”，CIFER软件及使用文档（见附录A），CopterSim, RflySim3D
- 硬件：计算机，自驾仪

■ 目标

对于给定的多旋翼非线性模型，通过对模型外加控制器的方式对其进行辨识，在假设存在时延的情况下获得四个通道的模型传递函数。注意四个通道建立的均是输入到地球系下位置/角度的传递函数，这里将水平通道可以分为水平前向通道和水平侧向通道，因此是四通道的。具体要求如下。

- 1) 了解利用CIFER软件进行系统辨识的大致流程。
- 2) 在各通道辨识后，获得的传递函数判定标准代价函数满足 $J < 50$ 。
- 3) 这里需要注意的是，为了获得位置通道的传递函数，需要设计控制器，获得的四通道非线性化传递函数模型。各通道期望输入为位置与偏航角，输出亦是如此。
- 4) 在获得辨识的传递函数模型之后，将分析实验中获得的速度模型加入与设计实验中相同的位置控制器，对比设计实验和分析实验所得模型的Bode图。

4.设计实验

□ 实验步骤

水平通道的两通道模型是解耦的（解耦条件则是偏航角为零，参考实验原理中的内容）。这一点在对模型进行线性化时可以看出，水平通道模型的位置与速度分别为 $\mathbf{p}_{hb} = \mathbf{R}_\psi^T \mathbf{p}_h$ 和 $\mathbf{v}_{hb} = \mathbf{R}_\psi^T \mathbf{v}_h$ ，而 $\mathbf{p}_h = [p_{xe} \ p_{ye}]^T$, $\mathbf{v}_h = [v_{xe} \ v_{ye}]^T$ 为多旋翼在地球坐标系下的位置和速度。因此，水平两通道解耦的条件即为 $\psi = 0$ ，即 $\mathbf{R}_\psi^T = \mathbf{I}$ 。需要注意的是，水平前向通道和水平侧向通道的输出量是两个方向的位置，而输入量则为加入控制器之后的新的控制变量。

输入到机体位置/角度的传递函数是不稳定的，因此我们采取如下图所示加PD控制器的方式进行辨识（这里我们以水平前向通道为例，水平侧向通道与此类似）。

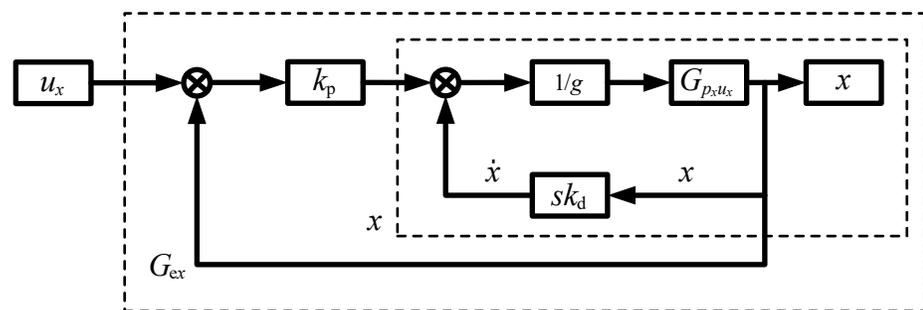


图. 水平前向通道PD控制器

比例微分控制器



$$G_{ex}(s) = \frac{\frac{1}{g} k_p G_{p_x u_x}}{1 + \frac{1}{g} s k_d G_{p_x u_x} + \frac{1}{g} k_d k_p G_{p_x u_x}}$$

$$G_{ey}(s) = \frac{\frac{1}{g} k_p G_{p_y u_y}}{1 + \frac{1}{g} s k_d G_{p_y u_y} + \frac{1}{g} k_d k_p G_{p_y u_y}}$$

4.设计实验

□ 实验步骤：仿真1.0

这里的具体步骤和分析实验的具体步骤类似，请参考4.3.2.1小节。不同的是在“e1_3_SystemIdentification.slx”模型文件中，四通道为合理的位置控制器。在通过对多旋翼非线性模型输入扫频信号获得输入\输出的时间历程数据后，利用CIFER软件获得对应通道的传递函数模型，以水平通道结果为例，其传递函数模型如下图所示。

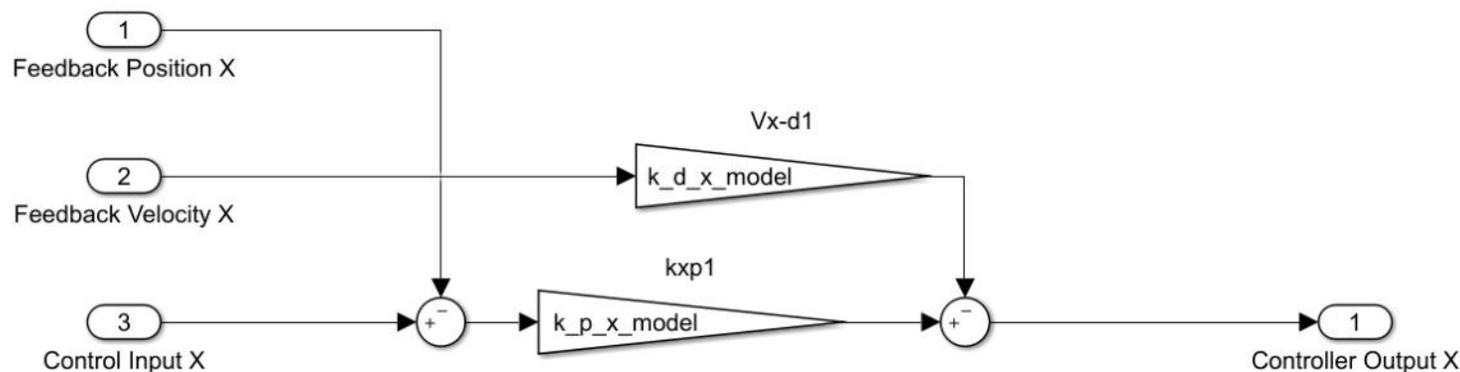


图. 设计实验水平前向通道“比例-微分”控制器，Simulink模型“e1_3_SystemIdentification.slx”中“Controller/P or PD Controller X”模块

4.设计实验

□ 实验结果：仿真1.0

(1) 水平前向通道

$$G_{p_{xe}u_{vx}} = \frac{6.48651}{s^3 + 3.72616s^2 + 12.6598s + 6.55711} e^{-0.0429s} \quad J = 0.099$$

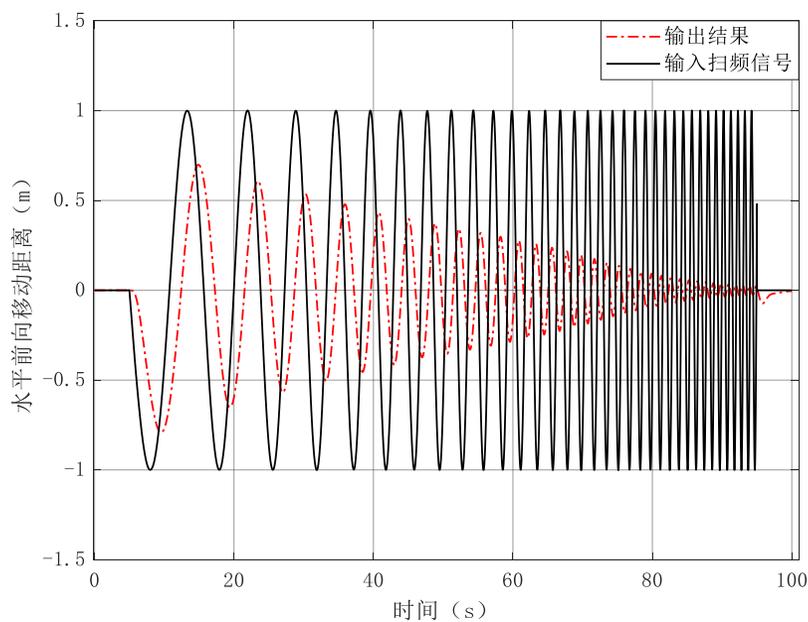


图. 设计实验水平前向通道扫频输入输出对比图

相应地，在水平前向通道模型中，仿真时的扫频输入输出对比如右图所示。可以发现，水平前向通道的输出可以较好的跟随扫频输入，特别是在低频阶段。

(2) 水平侧向通道

$$G_{p_{xe}u_{vy}} = \frac{6.48638}{s^3 + 3.72607s^2 + 12.66s + 6.55534} e^{-0.0429s} \quad J = 0.099$$

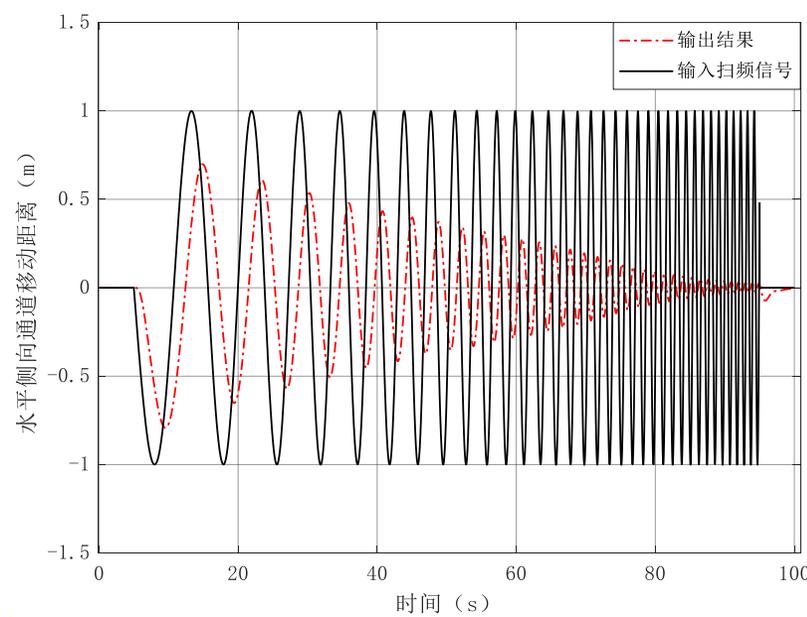


图. 设计实验水平侧向通道扫频输入输出对比图

相应地，在水平前向通道模型中，仿真时的扫频输入输出对比如左图所示。可以发现，水平前向通道的输出可以较好的跟随扫频输入，特别是在低频阶段。

第四章系统辨识实验

设计实验：仿真1.0

4.设计实验

□ 实验步骤：仿真2.0

由4.3.2.1小节结果及分析可以得到机体速度环的模型。在将4.4.2.1小节控制器式(4.27)和式(4.28)放入分析实验的模型之后，利用Simulink 将其与本实验获得的稳定位置闭环系统环模型进行频域响应对比。此处使用了MATLAB 中的“Bode”函数对传递函数的幅频以及相频曲线进行可视化显示，进行比较后的结果如下图所示。

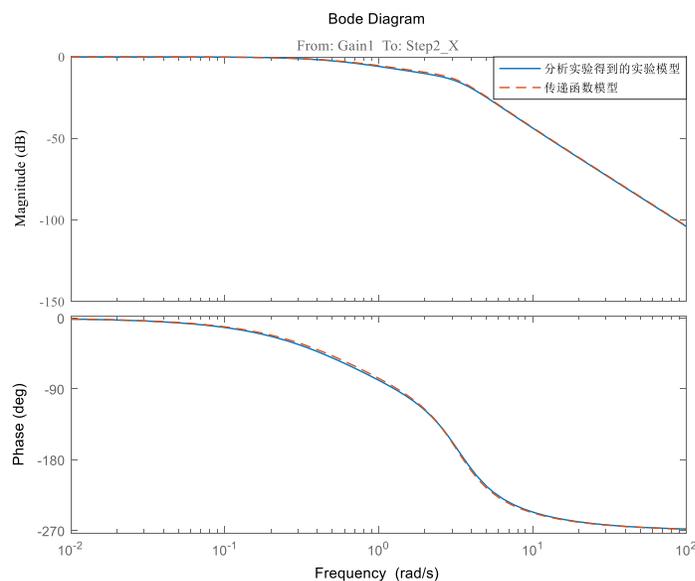


图. 水平前向通道分析实验与设计实验Bode图对比

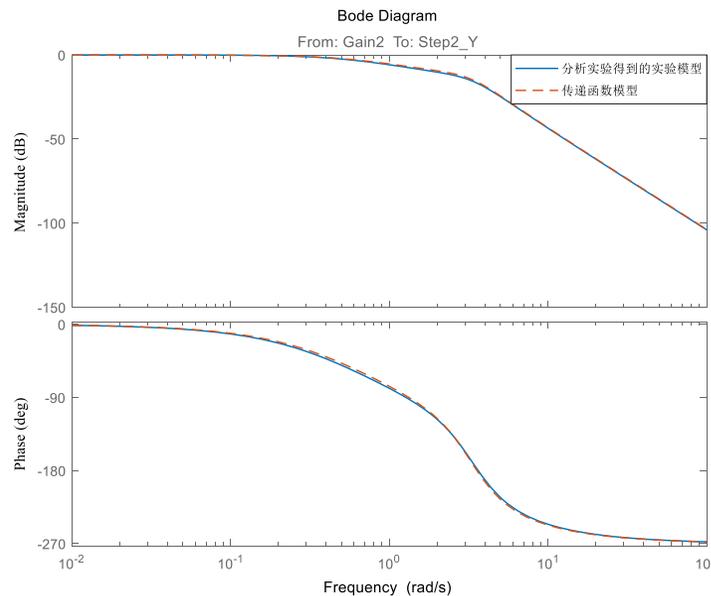


图. 水平侧向通道分析实验与设计实验Bode图对比

可以看出，对于水平通道、高度通道以及偏航通道，本实验中所获得的多旋翼模型与分析实验中的传递函数模型在稳定闭环中的幅频相频曲线几乎一致，表明模型的准确性比较高。

第四章系统辨识实验

设计实验：仿真2.0

4.设计实验

□ 实验步骤：硬件在环仿真

硬件在环仿真内容和目的与设计实验相同，只是从软件仿真变为了硬件在环仿真，首先对硬件在环仿真Simulink模型进行简单介绍，打开文件“e1_3_SystemIdentification_Vel_HITL.slx”文件，如下图所示。

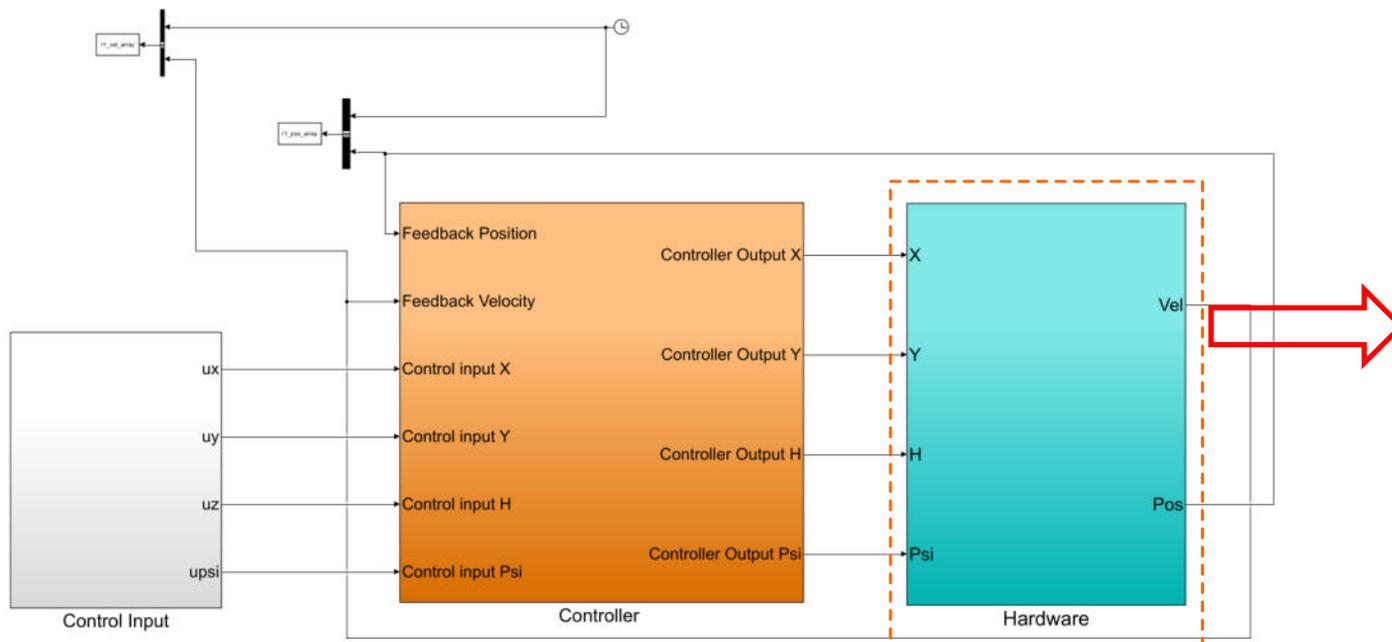


图. 硬件在环仿真接口模块

虚线框部分是硬件在环仿真的接口模块，用于和RflySim进行信息交互，控制器将扫频信号传入到硬件接口中，硬件接口返回响应信号并将记录下来的数据保存到工作区中。模型的其它部分与之前的Simulink模块相同。按照硬件在环仿真实验流程打开RflySim，连接自驾仪，并且按照之前步骤在“startSimulation.m”中对不必要代码进行注释。接着，先后运行“startSimulation.m”文件进行参数初始化及“e1_3_SystemIdentification_Vel_HITL.slx”文件进行系统辨识。在程序运行完成之后，运行“e1_3_GetFinalData.m”文件，保存刚才采集到的数据。将采集到的数据保存到CIFER指定的文件夹中，按照之前所述的辨识流程分别对每个通道进行辨识。根据辨识之后输出的传递函数结果修改“e1_3_SystemIdentification_Vel_Verification_HITL.slx”文件中相应的传递函数模块。点击“运行”便可以对辨识结果和硬件在环仿真结果的差别，从而验证辨识结果的准确性。

4.设计实验

□ 实验步骤：硬件在环仿真

下面以高度通道进行举例说明。

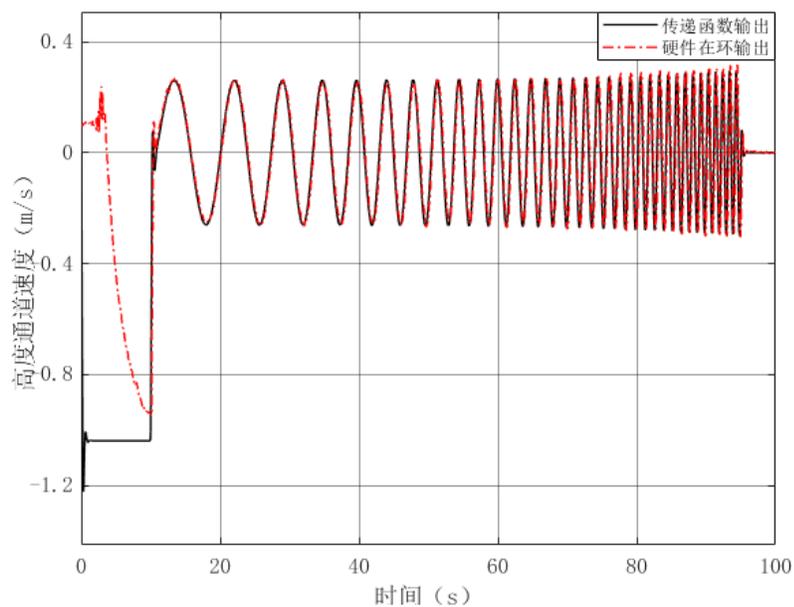


图. 高度通道辨识结果



可以观察到高度通道的信号在**开始阶段误差有些大**。这是因为在硬件在环仿真当中，需要先给四旋翼一个向上飞的速度指令，使得多旋翼起飞；在起飞之后，才能正常进行高度通道的扫频。因此，在图中刚开始的阶段有一个持续10s的起飞速度指令。虚线则表示多旋翼高度方向上的速度由静止到跟随速度指令的连续变化的响应情况。因此，**曲线拟合情况**表明了高度通道辨识结果的**正确性**。



图. 高度通道硬件在环仿真效果图

第四章多旋翼系统辨识 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“e1.4”，CIFER软件及使用文档（见附录A）
- 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

通过对多旋翼外加控制器的方式，对其进行位置以及偏航通道模型的系统辨识，获得四个带时延的通道传递函数，具体目标如下。

- 1) 了解利用CIFER软件进行系统辨识的大致流程。
- 2) 在各通道辨识后，获得的传递函数判定标准代价函数满足 $J < 100$ 。

这里需要注意的是，为了获得位置通道的传递函数，需要设计控制器，获得四通道线性化传递函数模型。各通道期望输入为位置与偏航角，输出亦是如此。

5. 实飞实验

□ 实验步骤

实飞实验所使用的多旋翼平台内置速度环控制器。即使如此，基于速度环的多旋翼室内飞行实验依然非常危险。因此，依据实验目标，需要对多旋翼位置环进行系统辨识实验。从速度环到位置环，需要加入相应的控制器，如下图所示。对于系统辨识实验，这里加入控制器的目标并不是对多旋翼进行精准控制，而是为了使其稳定，从而能够安全地进行室内的飞行实验。因此，这里多旋翼各通道设置的控制器可以简单。

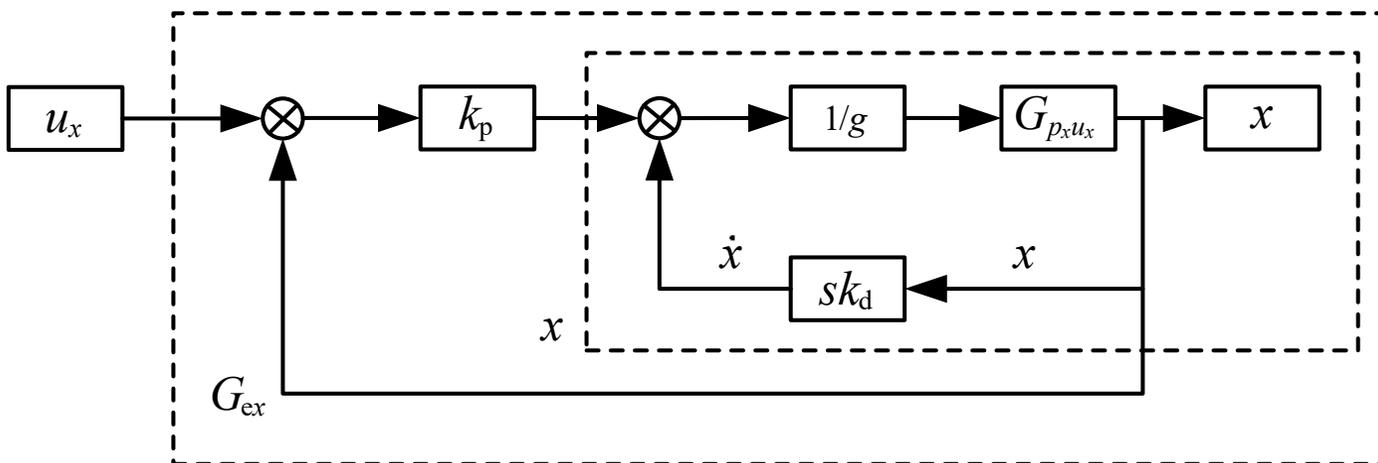


图. 实飞实验PD控制器



5. 实飞实验

□ 实验步骤

(1) 步骤一：阅读相关硬件使用说明文档

首先阅读多旋翼飞行器使用文档(详细内容请参考3.2小节)的内容，熟悉并了解如何操作给定多旋翼飞行器。阅读“CIFER使用指南”(见附录A)，学习如何进行多旋翼的系统辨识。

(2) 步骤二：了解相关程序代码

打开本节课所附带的文件夹(“e1\e1.4”), 该文件夹内容如下: 1) “start_tello.m”文件, 用于初始化以及启动相应的Simulink程序; 2) “e1.4_cifer_model.slx”文件, 用于与给定多旋翼飞行器进行交互所使用的Simulink模型; 3) “generate_cifer_datapack.m”文件, 用于保存扫频实验生成的实验数据, 用于模型的系统辨识。

5. 实飞实验

□ 实验步骤

(3) 步骤三：初始化参数

打开“e1\ e1.4\ start_tello.m”初始化文件。在其中可以找到如下图所示代码。启动多旋翼的步骤在第三章3.2节中已经进行了详细说明，唯一需要注意的是：下图中第三行代码用于实验结束后数据的保存，需要根据当前扫频实验的通道而进行更改。各通道分别为：“Pos_X”（水平前向通道），“Pos_Y”（水平侧向通道），“Pos_Z”（高度通道），“Yaw”（偏航通道）。

```
%扫频信号选择
Model_Channel = struct('YAW',1,'YAWRATE',2,'POS_X',3,'POS_Y',4,'POS_Z',5,'VEL_X',6,'
    VEL_Y',7,'VEL_Z',8);
model_channel = Model_Channel.YAW;
PARAM_time_start = 3;%3s后产生扫频信号
    % ns后多旋翼稳定
PARAM_C1 = 4.0;
PARAM_C2 = 0.0187;
PARAM_w_min = 0.5;%0.3 0.4 0.5
PARAM_w_max = 12;%10 11 12
```

表示保存偏航通道的扫频数据

5. 实飞实验

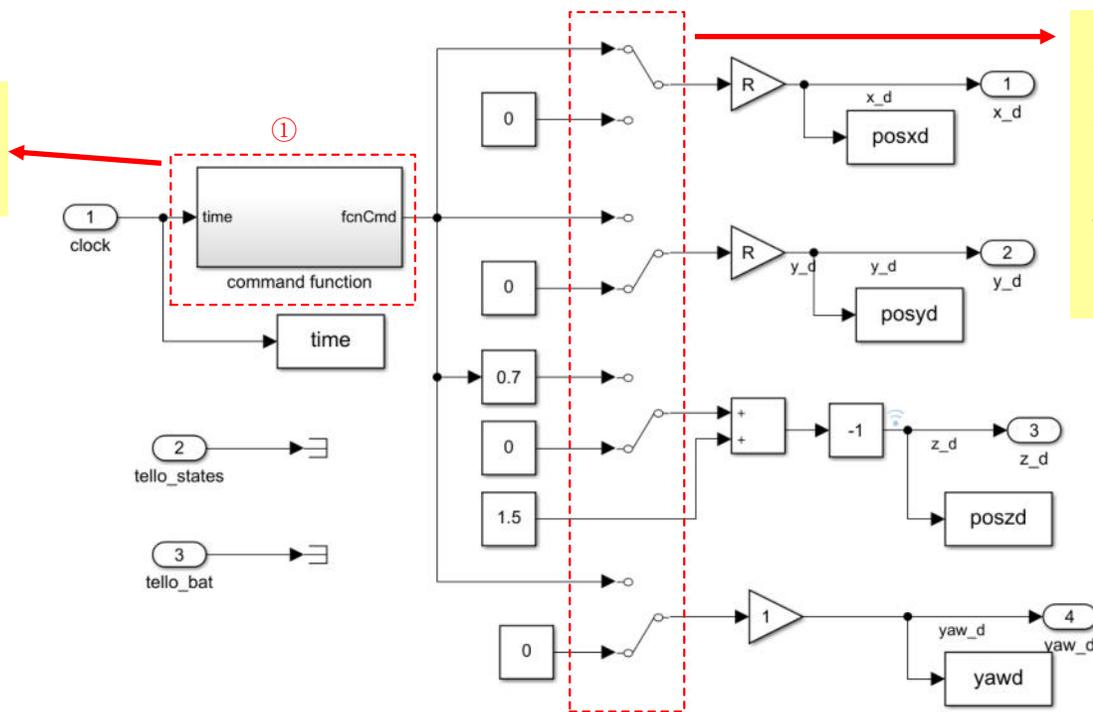
实验步骤

(4) 步骤四：打开Simulink模型

打开“e1\ e1.4\ e1_4_cifer_model.slx”模型文件

②

该模块用于生成扫频数据



表示是否引入扫频信号，图中
所示案例为水平前向通道引入
扫频信号，其它通道均设置为
稳定值，即多旋翼在该通道的
“平衡点”附近。

图. 期望输入模块示意图，Simulink文件“e1_4_cifer_model.slx”

5. 实飞实验

□ 实验步骤

(4) 步骤四：打开Simulink模型

在“e1\e1.4\e1_4_cifer_model.slx”模型文件中为四通道设计合理的位置控制器。这里所设计的控制器只需要用最简单的“比例-微分”控制器即可（如下图所示），同时参数设置也应该尽量简单，达到系统稳定的目的便可以开始辨识。特别注意的是，因为偏航角的控制更加底层也就更加容易，因此只需一个“比例”控制器就可以使得该通道稳定。

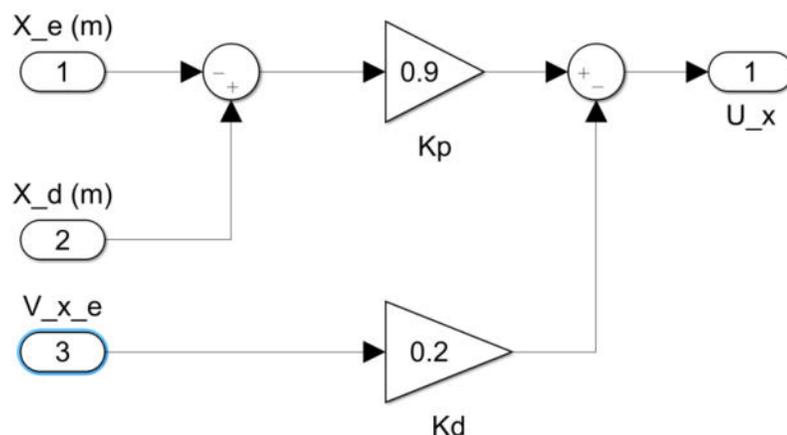


图. 实飞实验水平前向通道“比例-微分”控制器，Simulink文件“e1_4_cifer_model.slx”中“control system”模块



5. 实飞实验

□ 实验步骤

(5) 步骤五：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`”，可以看到两架多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

运行“`e1_4_cifer_model.slx`”模型文件对多旋翼相应通道输入扫频信号，分别进行各通道扫频。需要注意，这里每次更换通道进行扫频时，需要修改步骤三中第三行代码“`Model_Channel.YAW`”，与需要辨识通道相对应。

◆ 降落Tello

打开一个新终端，运行命令“`roslaunch tello Tello_land_all`”，在两架多旋翼降落后，结束所有终端。



5. 实飞实验

□ 实验步骤

(6) 步骤六：保存数据

运行文件“generate_cifer_datapack.m”，保存三组不同初始频率下的各通道扫频输入输出数据，将其复制到CIFER软件的目录下，进行系统辨识，获得相应的结果。CIFER软件的使用请参考附录A。接下来我们给出一段系统辨识过程视频。

实飞实验辨识

5. 实飞实验

□ 实验结果

◆ 水平前向通道

$$G_{p_{x_e} u_{v_x}} = \frac{1.24431s + 1.75269}{s^3 + 2.63008s^2 + 2.37405s + 1.95897} e^{-0.3778s} \quad J = 52.074.$$

可以发现水平前向通道的输出可以较好的跟随期望输入，特别是在低频阶段。注意的是，我们设计的控制器并不需要输出跟踪上输入，仅仅需要保持系统稳定即可。

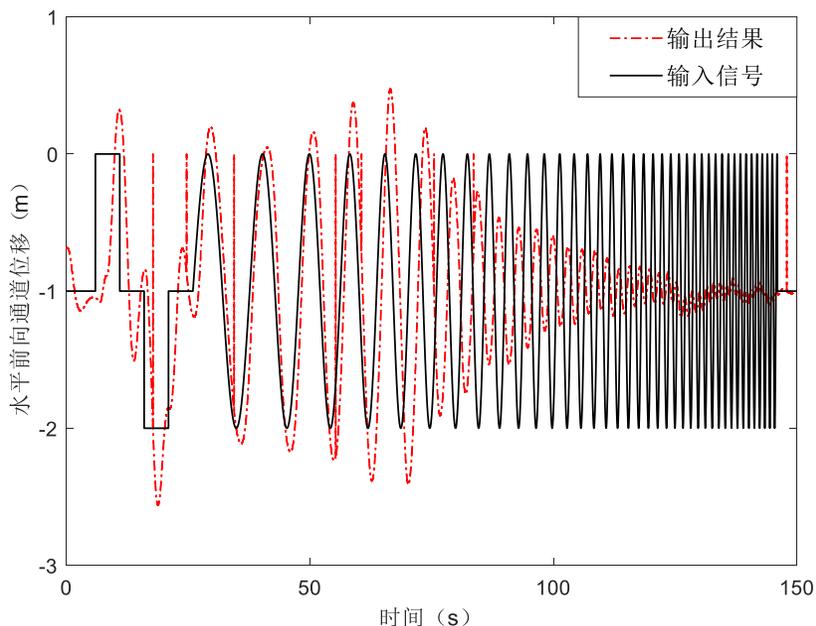


图. 实飞实验水平前向通道扫频输入输出对比图



图. 实飞实验水平前向通道扫频实飞图 (影像进行了叠加)

5. 实飞实验

□ 实验结果

◆ 水平侧向通道

$$G_{p_{ye}u_{vy}} = \frac{1.21543s + 2.06629}{s^3 + 2.80018s^2 + 2.54762s + 2.23605} e^{-0.3715s} \quad J = 58.173.$$

输出可以较好地跟随期望输入，特别是在低频阶段。

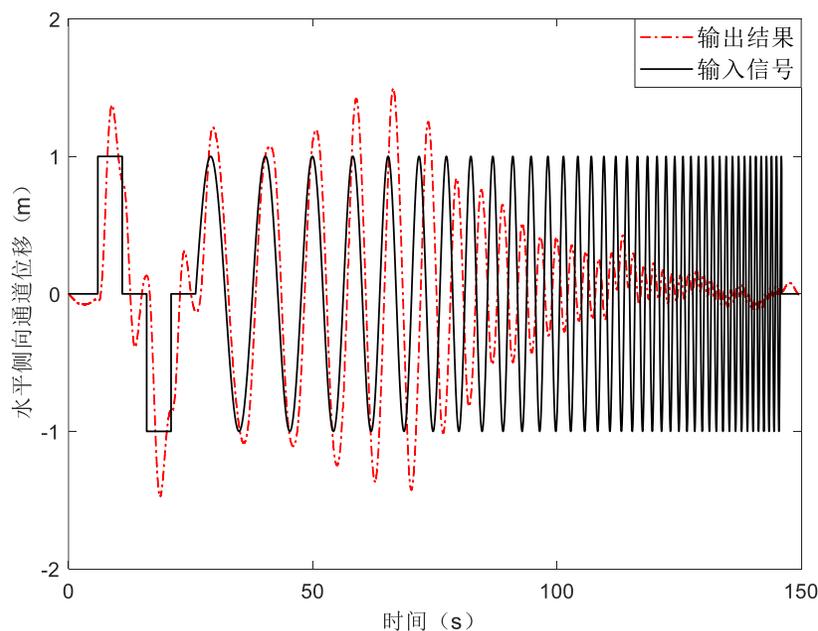


图. 实飞实验水平侧向通道扫频输入输出对比图



图. 实飞实验水平前向通道扫频实飞图 (影像进行了叠加)

5. 实飞实验

□ 实验结果

针对于上面获得的给定多旋翼飞行器模型，还需要重新对其进行实飞验证，以验证模型的准确性。

这里将使用与系统辨识中相似的扫频信号进行验证。我们分别对给定多旋翼飞行器和获得的传递函数模型输入相同的扫频位置控制指令，对比获得的位置输出的拟合效果，如下所示。

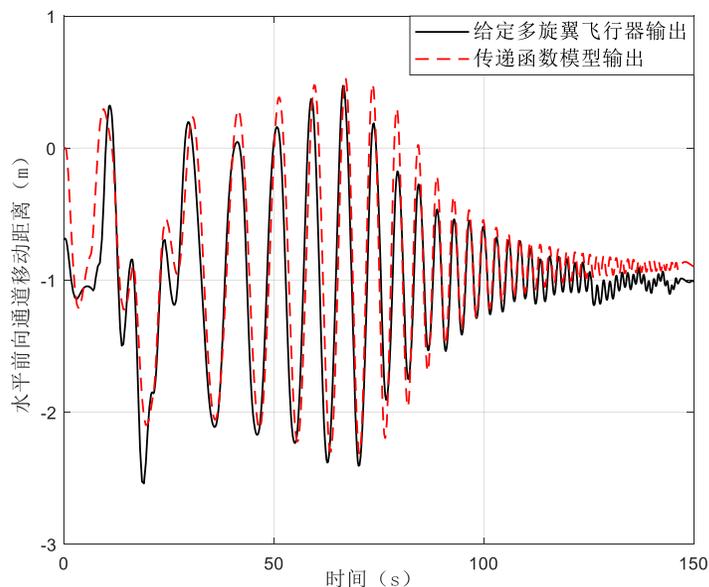


图. 实飞实验水平前向通道扫频模型验证对比图

在水平两通道中给定多旋翼飞行器和获得的传递函数模型的输出基本一致，特别在**低频阶段效果更好**，**高频阶段误差有些大**。

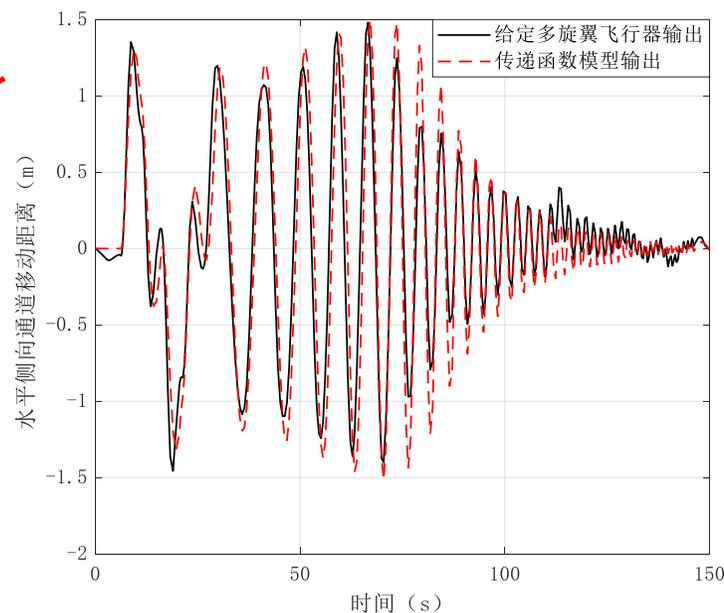


图. 实飞实验水平侧向通道模型验证对比图

通过本章的实飞扫频系统辨识实验，可以获得给定多旋翼飞行器的四通道位置通道模型。经过实飞实验的测试以及验证，此处获得的各个通道模型的传递函数与真实的多旋翼系统的输入输出响应是**基本一致**的，尤其是在**低频响应段**，其与真实飞行效果几乎一致，即通过CIFER频域响应辨识获得的系统模型**准确且有效**。可以利用它来进行Simulink仿真，调节相应控制器的参数，例如第六章的路径跟踪控制器实验，**可大大缩短实验的时间，减少了许多不必要的真机调试过程**。

第四章多旋翼系统辨识 实飞实验

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

6.总结



(1) 通过基础实验，可以理解与熟悉建模过程的理论假设与推导过程，并进一步理解与熟悉获得的四通道线性化多旋翼简化模型。在传递函数模型与非线性模型扫频对比中，可以看出两者在低频段拟合较好，可以利用系统辨识获得的传递函数模型来进行低频段仿真及控制器设计。

(2) 通过分析实验，了解如何利用CIFER软件进行系统辨识，包括对系统进行建模的大致流程及判定的标准代价函数。通过对非线性模型中稳定的速度通道进行系统辨识，直接获得了多旋翼关于速度的传递函数模型。进一步，进行积分可获得位置通道模型。

(3) 在设计实验中，通过设计增稳控制器的方式，直接对非线性模型位置通道进行辨识，并与分析实验中得到的位置通道模型进行比较，验证了分析实验所得到的位置通道模型的正确性。

(4) 在实飞实验，对多旋翼外加控制器，进行位置以及偏航通道模型的系统辨识，验证了系统辨识方法的实用性和辨识模型的准确性。

(5) 通过以上实验，让读者较为深入掌握多旋翼的模型结构以及建模方法。

致谢



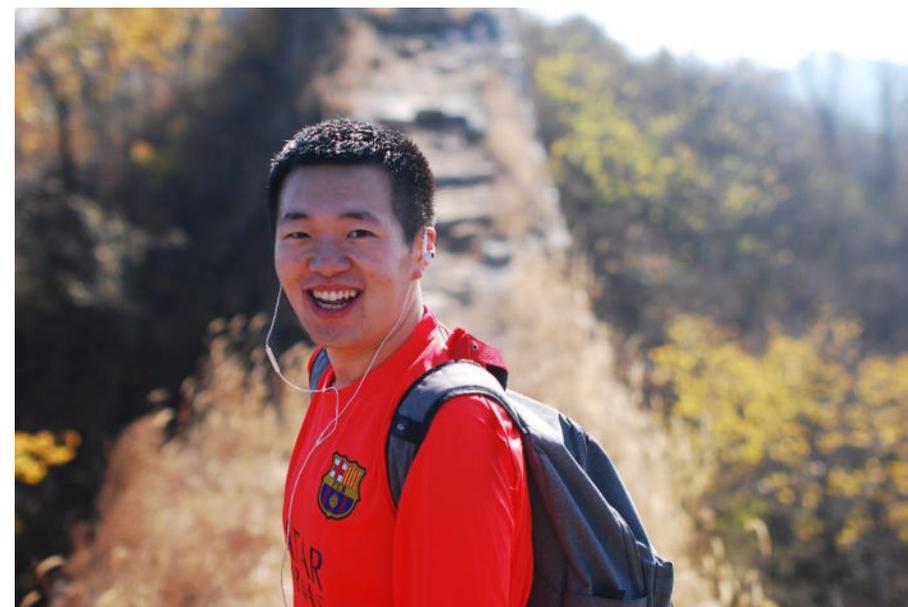
感谢可靠飞行控制研究组同学



宁俊清



毛鹏达



马泽青

为本节课程准备做出的贡献



谢谢!

全 权

qq_buaa@buaa.edu.cn

可靠飞行控制研究组



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