

---

# 1. 实验名称及目的

## 1.1. 实验名称

系统辨识实验 (分析实验)

## 1.2. 实验目的

对给定的多旋翼非线性模型进行辨识,在假设存在延时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。

(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  $J \leq 50$ , 详情参考“CIFER 软件使用指南”(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。

## 1.3. 关键知识点

4.1.1 节建立了多旋翼各通道的线性模型,这些模型会在辨识传递函数模型时用到。因为辨识是为了获取传递函数的参数,所以各通道的线性模型为系统辨识提供了先验知识。这里用传递函数来描述各通道从输入到输出的动态过程,这些传递函数的形式可以根据线性模型得到。本节将对多旋翼的高度通道模型、偏航通道模型和水平位置通道模型分别进行系统辨识。在具体工作之前,先介绍系统辨识及其常用工具。

### 1) 系统辨识步骤

系统辨识步骤如图 4.2 所示,主要包含以下六部分:先验知识、实验设计、数据采集、模型选择、模型计算和模型检验。

(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他方面的已有知识。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。

(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。如图 4.3 所示,目前有两类实验设计方案:开环实验和闭环实验。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,见图 4.3 (b);而在开环实验中,见图 4.3 (a),输入信号可以任意选择,因此开环实验比闭环实验能获得更多的系统信

息,系统模态更容易被激励出来。显然,如果系统可以在脱离控制器的情况下工作,那么开环实验是更好的选择。

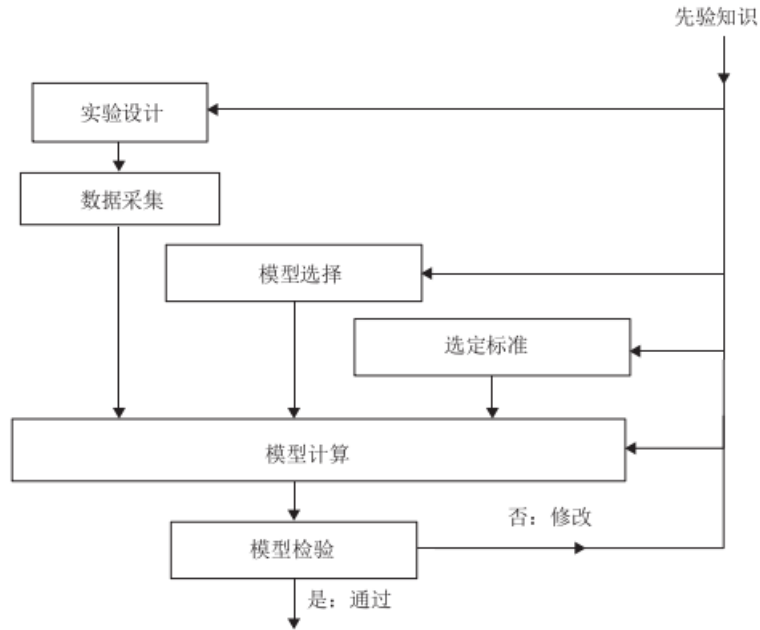


图 4.2 系统辨识过程

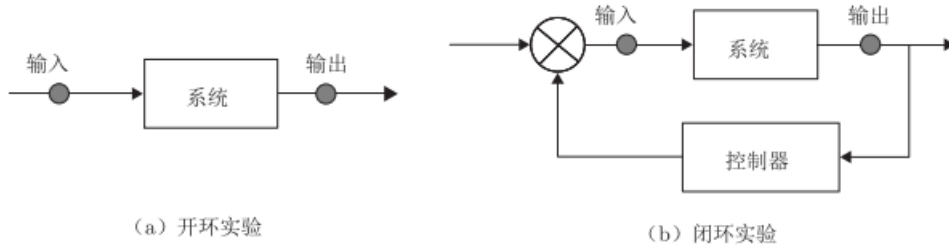


图 4.3 两类实验设计方案

(3) **数据采集**：通过合理的实验来获得输入/输出数据。

(4) **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。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。

(5) **模型计算**：采用合适的优化方法来计算备选模型的未知参数。

(6) **模型检验**：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。

## 2) 系统辨识工具箱

(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。利用该工具箱，用户可

以用系统的时域或频域输入/输出数据来辨识其连续或离散的传递函数、过程模型、状态空间模型等，得到利用物理规律很难进行建模的系统动态模型。该工具箱还提供了很多辨识方法，如极大似然法、预测误差最小化方法以及子空间辨识方法等。如果考虑非线性系统辨识，还可以使用 Hammerstein-Wiener 模型、带有小波网络的非线性 ARX 模型以及其他非线性模型。该工具箱同时提供“灰箱”系统辨识，即使用自定义的模型进行参数估计。辨识出来的模型可用于建模及系统响应预测。

(2) CIFER 工具箱<sup>[11]</sup>：该工具箱基于综合频域响应方法进行系统辨识，适用于解决复杂的系统辨识问题，可以用于辨识不稳定模态，尤其是与飞机建模和控制器设计相关的系统辨识问题。CIFER 的基本原理是通过提取一组高质量完备的非参数多输入多输出频域响应数据，在不需要先验假设的前提下，直接由这些响应数据提取得到系统的耦合特性。CIFER 以先进的线性调频 Z 变换以及组合优化窗技术作为理论基础，相比于标准的快速傅里叶变换而言，其频域响应质量有显著的提高。在完整频域响应数据集已知的前提下，可以用多种复杂的非线性搜索算法来提取状态空间模型。因此，该工具箱在多种真实飞机的系统辨识中得到广泛应用，具体使用参见附录 A。

## 2. 实验效果

通过实验分析水平前向及侧向通道输入\输出对比图。

## 3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\6.RflySimExtCtrl\1.BasicExps\e7\\_MutUAVRemoteCtrl\1.System identification experiment\el.2\](#)

文件夹/文件名称	说明
e1_2_SystemIdentification_Vel_2017b.slx	多旋翼 simulink 仿真文件
startSimulation.m	初始化参数文件
e1_2_SystemIdentification_Vel_Verification_2017b.slx	模型验证文件
e1_2_GetFinalData.m	保存数据文件

## 4. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 <sup>①</sup>	1
2	RflySim 工具链		
3	MATLAB 2022b 及以上版本		

①：推荐配置请见：<https://doc.rflysim.com>

---

## 5. 实验步骤

### 5.1. 仿真 1.0(sim1.0)实验步骤:

#### Step 1:

首先阅读第 2 章的内容, 对于 Simulink 模型有一个初步的了解。阅读“CIFER 软件使用指南”(见附录 A), 学习如何进行系统辨识。

#### Step 2:

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 0.08~1.18Hz;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 0.08~1.18Hz 得到的时间历程文件数据 R);(2)“e1\_2\_SystemIdentification\_Vel.slx”为模型文件,由非线性系统、控制器以及信号输入构成,用于对模型进行扫频操作;(3)“e1\_2\_SystemIdentification\_vel\_Verification.slx”是模型验证文件,用于对比多旋翼非线性模型和传递函数在相同输入情况下的输出;(4)“startSimulation.m”是初始化文件,对 Simulink 里面的相关参数进行初始化操作。

#### Step 3:

打开“startSimulation.m”初始化文件,在其中可以找到如表 4.1 所示的一段代码。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,分别用于仿真时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。

表 4.1 “startSimulation.m” 中扫频数据加载代码

```

1 %% 加载仿真文件数据
2 load('0.08/F434test_psi_0.08_1.18.mat')
3 load('0.08/F434test_x_0.08_1.18.mat')
4 load('0.08/F434test_y_0.08_1.18.mat')
5 load('0.08/F434test_z_0.08_1.18.mat')
6
7 % load('0.09/F434test_psi_0.09_1.21.mat')
8 % load('0.09/F434test_x_0.09_1.21.mat')
9 % load('0.09/F434test_y_0.09_1.21.mat')
10 % load('0.09/F434test_z_0.09_1.21.mat')
11
12 % load('1.11/F434test_psi_0.11_1.25.mat')
13 % load('1.11/F434test_x_0.11_1.25.mat')
14 % load('1.11/F434test_y_0.11_1.25.mat')
15 % load('1.11/F434test_z_0.11_1.25.mat')
16
17 %% 保存时间历程文件.mat
18 load('0.08/F434loaddata_psi_0.08_1.18.mat')
19 load('0.08/F434loaddata_x_0.08_1.18.mat')
20 load('0.08/F434loaddata_y_0.08_1.18.mat')
21 load('0.08/F434loaddata_z_0.08_1.18.mat')
22
23 % load('0.09/F434loaddata_psi_0.09_1.21.mat')
24 % load('0.09/F434loaddata_x_0.09_1.21.mat')
25 % load('0.09/F434loaddata_y_0.09_1.21.mat')
26 % load('0.09/F434loaddata_z_0.09_1.21.mat')
27
28 % load('1.11/F434loaddata_psi_0.11_1.25.mat')
29 % load('1.11/F434loaddata_x_0.11_1.25.mat')
30 % load('1.11/F434loaddata_y_0.11_1.25.mat')
31 % load('1.11/F434loaddata_z_0.11_1.25.mat')

```

#### Step 4:

在完成步骤三之后，运行“startSimulation.m”初始化文件，初始化实验参数。

#### Step 5:

打开模型文件“el\_2\_SystemIdentification\_Vel\_2017b.slx”，并在“Control Input”模块中修改相应的扫频输入通道，详细步骤请参考 4.2.2.1 节。

#### Step 6:

模型文件“el\_2\_SystemIdentification\_Vel.slx\_2017b”是主要的实验模型，该模型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相应的扫频通道，将其中的“Switch”开关拨到相应位置，进行扫频即可。详细步骤请参考 4.2.2.1 节。

#### Step 7:

在每个通道仿真完毕后，需要保存数据。保存数据需要打开“el\_2\_GetFinalData.m”文件，其代码如表 4.2 和表 4.3 所示。表 4.2 表示在 MATLAB 工作区保存相关数据，表 4.3 表示保存数据并生成用于 CIPHER 软件进行辨识的时间历程文件。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。在本例中，表 4.2 中第 10 行、第 16 行和第 22 行分别表示在 MATLAB 工作区保存水平前向的位置反馈、水平前向通道期望速度和水平前向通道速度反馈；表 4.3 中第 5 行和第 22 行分别表示生成频率范围为 0.08~1.18Hz 的期望输入“ux”一位置输出“outputx”时间历程文件和频率范围为 0.08~1.18Hz 的期望输入“ux”一速度输出“outputvx”时间历程文件。

表 4.2 “e1\_2\_GetFinalData.m” 中加载数据部分

```
1 %% 加载数据区域
2
3 % 保存期望位置数据
4 ux=4*uxreal_array(:,1);
5 % uy=4*uyreal_array(:,1);
6 % uh=4*(uhreal_array(:,1)-1)+1;
7 % upsi=4*upsireal_array(:,1);
8
9 % 保存反馈位置数据
10 outputx=r1_pos_array(:,2);
11 % outputy=r1_pos_array(:,3);
12 % outputh=r1_pos_array(:,4);
13 % outputps=r1_pos_array(:,5);
14
15 % 保存期望速度数据
16 ux=uxreal_array(:,1);
17 % uy=uyreal_array(:,1);
18 % uh=1-uhreal_array(:,1);
19 % upsi=upsireal_array(:,1);
20
21 % 保存速度反馈数据
22 outputvx=r1_vel_array(:,2);
23 % outputvy=r1_vel_array(:,3);
24 % outputvh=r1_vel_array(:,4);
25 % outputvp=r1_vel_array(:,5);
```

表 4.3 “e1\_2\_GetFinalData.m” 中保存相关数据部分

```

1 %% 保存在.mat文件的数据
2
3 % CIPHER软件要求变量名最多8个字符（不带下划线）。
4
5 save data_x_0.08_1.18.mat time ux outputx;
6 % save data_y_0.08_1.18.mat time uy outputy;
7 % save data_z_0.08_1.18.mat time uh outputh;
8 % save data_psi_0.08_1.18.mat time upsi outputps;
9
10 % save data_x_0.09_1.21.mat time ux outputx;
11 % save data_y_0.09_1.21.mat time uy outputy;
12 % save data_z_0.09_1.21.mat time uh outputh;
13 % save data_psi_0.09_1.21.mat time upsi outputps;
14
15 % save data_x_0.11_1.25.mat time ux outputx;
16 % save data_y_0.11_1.25.mat time uy outputy;
17 % save data_z_0.11_1.25.mat time uh outputh;
18 % save data_psi_0.11_1.25.mat time upsi outputps;
19
20 % CIPHER软件要求变量名最多8个字符（不带下划线）。
21
22 save data_vx_0.08_1.18.mat time ux outputvx;
23 % save data_vy_0.08_1.18.mat time uy outputvy;
24 % save data_vz_0.08_1.18.mat time uh outputvh;
25 % save data_vpsi_0.08_1.18.mat time upsi outputvps;
26
27 % save data_vx_0.09_1.21.mat time ux outputvx;
28 % save data_vy_0.09_1.21.mat time uy outputvy;
29 % save data_vz_0.09_1.21.mat time uh outputvh;
30 % save data_vpsi_0.09_1.21.mat time upsi outputvps;
31
32 % save data_vx_0.11_1.25.mat time ux outputvx;
33 % save data_vy_0.11_1.25.mat time uy outputvy;
34 % save data_vz_0.11_1.25.mat time uh outputvh;
35 % save data_vpsi_0.11_1.25.mat time upsi outputvps;

```

## Step 8:

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 CIPHER 软件的目录下，进行系统辨识，获得相应的结果。利用 CIPHER 软件进行辨识的具体步骤请参考附录 A。

## Step 9:

在获得了四个通道各自的传递函数模型之后，运行“elle1.2 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。

（具体操作步骤可以观看每章相对应的 PPT，其中由各个实验相对应的演示视频。）

## 5.2. 仿真 2.0(sim2.0)实验步骤:

打开附带的“elle1.2\el\_2\_SystemIdentification\_Vel\_Verification.slx”模型文件，其内部如图 4.9 所示。在图 4.9 中，虚线框①部分代表系统辨识的传递函数模型，②部分的作用是载入之前通过扫频所获得的各通道扫频输出数据，虚线框③部分的作用是将传递函数所获得的扫频输出保存至工作空间，从而进行定量的拟合度计算。

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“elle1.2\el\_2\_SystemIdentification\_vel\_Verification\_Corr2.m”文件;接下来运行“el\_2\_System

“Identification\_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输入相同信号,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软件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。

在通过模型仿真获得时间历程数据后,利用CIFER软件进行辨识,得到各通道的传递函数。由于水平通道较为复杂,因此将对水平通道结果进行分析,高度通道和偏航通道读者可以自行分析。

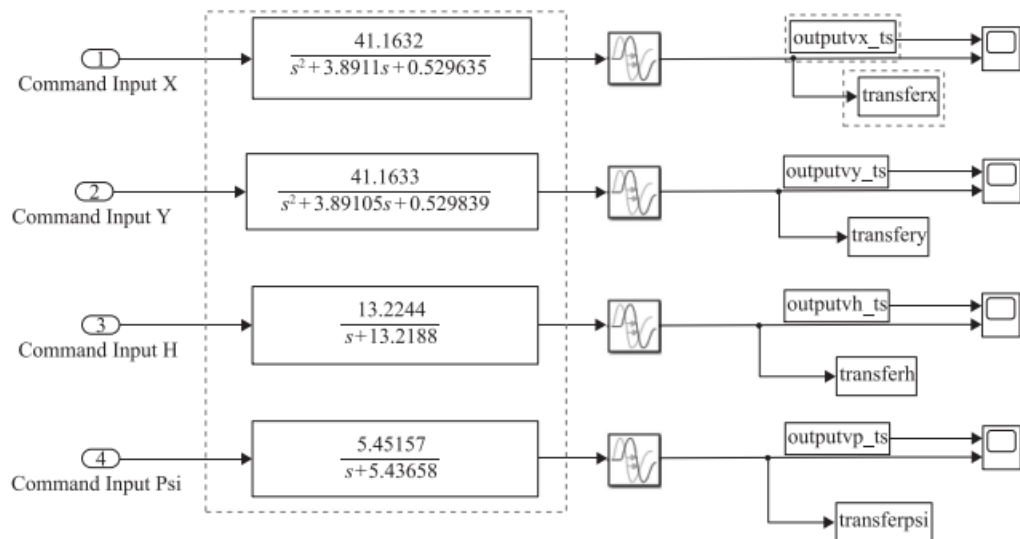


图 4.9 扫频输入验证模型示意图,Simulink 文件详见“e1\_2\_SystemIdentification\_Vel\_Verification.slx”

(1) 水平前向通道模型

$$G_{v_{yb}u_{yx}} = \frac{41.1632}{s^2 + 3.8911s + 0.529635} e^{-0.0289s} \quad J = 0.267 \quad (4.24)$$

其中，代价函数指标  $J = 0.267$ 。相应地，仿真时的扫频输入/输出对比如图 4.10 所示。

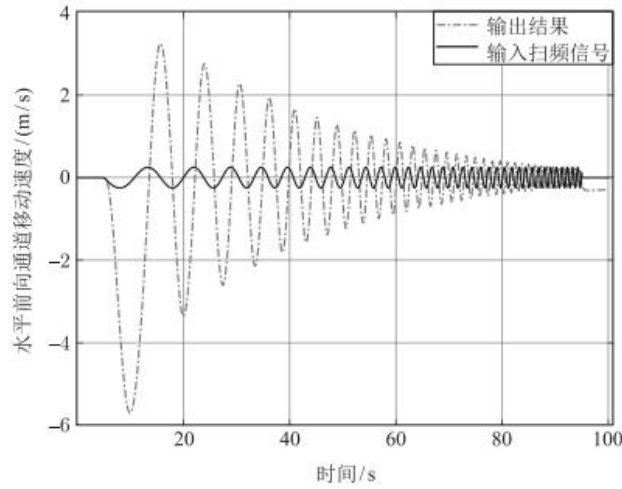


图 4.10 分析实验水平前向通道输入/输出对比图

在对简化的辨识结果进行验证之后，获得的两个输出（传递函数辨识结果输出与多旋翼非线性模型输出）的扫频信号验证对比结果如图 4.11 所示。由图 4.11 可知，二者接近重合。而数据也证明了这一点，利用“corr2”函数获得的相似度为 98.61%，二者相似度很高，验证通过。

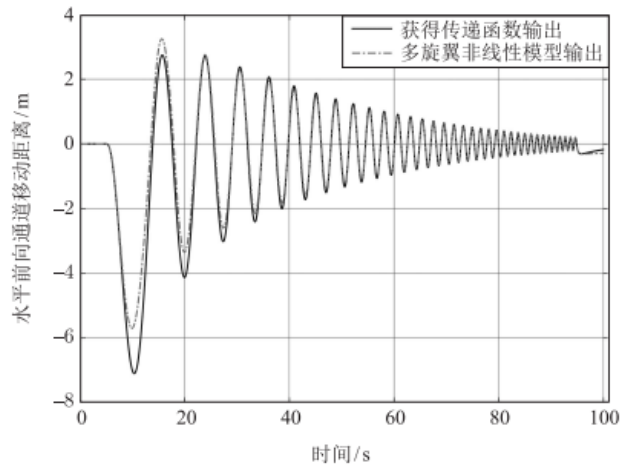


图 4.11 分析实验水平前向通道验证对比图

(2) 水平侧向通道模型

$$G_{y_b u_{y_v}} = \frac{41.1633}{s^2 + 3.89105s + 0.529839} e^{-0.0289s} \quad J = 0.267 \quad (4.25)$$

其中，代价函数指标  $J = 0.267$ 。相应地，模型中仿真时的扫频输入/输出对比如图 4.12 所示。

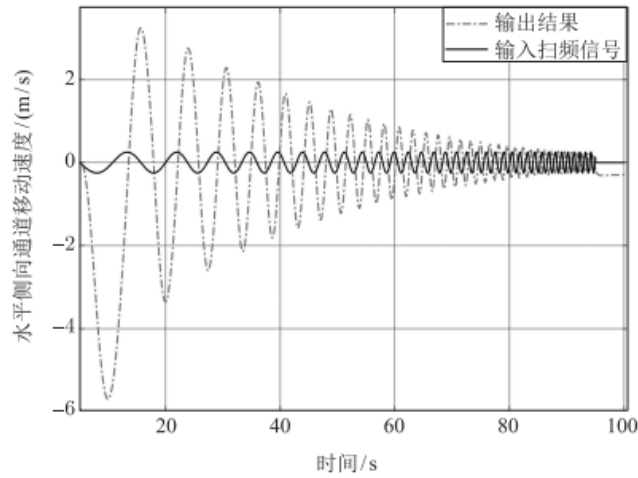


图 4.12 分析实验水平侧向通道输入/输出对比图

在对简化的辨识结果进行验证之后，获得的两个输出（传递函数辨识结果输出与多旋翼非线性模型输出）的扫频信号验证对比结果如图 4.13 所示。由图 4.13 可知，两者接近重合。而数据也证明了这一点，利用 MATLAB 自带的“corr2”函数获得的相似度为 98.61%，两者相似度很高，验证通过。

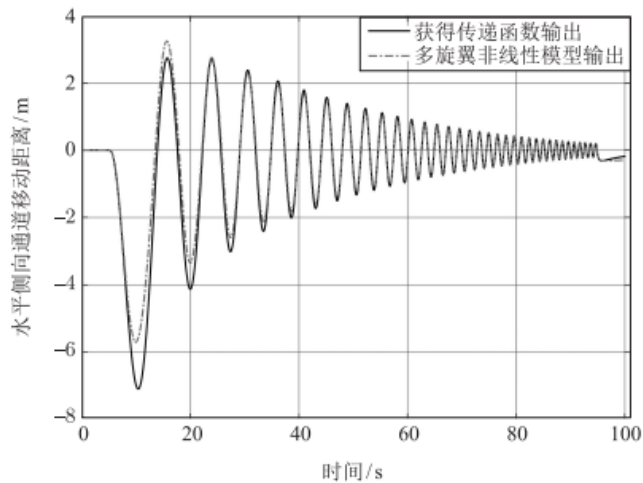


图 4.13 分析实验水平侧向通道验证对比图

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  $\mathbf{R}_\psi$  之后，水平通道通道传递函数可表示如下：

---

$$\begin{bmatrix} G_{p_{\psi} u_{\psi x}} \\ G_{p_{\psi} u_{\psi y}} \end{bmatrix} = \mathbf{R}_{\psi} \begin{bmatrix} \frac{41.1632}{s^3 + 3.8911s^2 + 0.529635s} e^{-0.0289s} \\ \frac{41.1633}{s^3 + 3.89105s^2 + 0.529839s} e^{-0.0289s} \end{bmatrix} \quad (4.26)$$

从式 (4.24) 和式 (4.25) 可以看出, 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%。因此可以认为辨识是成功的。

(具体操作步骤可以观看每章相对应的 PPT, 其中由各个实验相对应的演示视频。)

## 6. 参考资料

- [1]. Quan Quan. Introduction to Multicopter Design and Control. Springer, Singapore, 2017
- [2]. 全权 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 [M] 电子工业出版社 2018.
- [3]. 全权 戴训华 王帅 多旋翼飞行器设计与控制 实践 [M] 电子工业出版社 2020.
- [4]. 全权 等.多旋翼无人机远程控制实践[M].电子工业出版社,2022.

## 7. 常见问题

Q1: \*\*\*\*

A1: \*\*\*\*