

1. 实验名称及目的

1.1. 实验名称

系统辨识实验 (实飞实验)

1.2. 实验目的

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, 获得四个带延时的通道传递函数, 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 $J \leq 100$ 。

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, 输出也是如此。

1.3. 关键知识点

本节先介绍高度通道的辨识, 然后介绍偏航通道。只有在这两个通道稳定后, 水平位置通道的实验才能获得较好的输入/输出数据。因此, 水平位置通道将在最后介绍。注意, 如果想得到效果较好的系统辨识结果, 待辨识的通道需要尽可能稳定。因此, 在实验设计阶段, 分析待辨识的通道是否稳定显得尤为重要。

如果待辨识的通道是稳定的, 可以直接辨识; 如果不稳定, 可以先引入一个 P 或 PD 控制器来确保该通道稳定, 然后参考图 4.4 为该通道设计实验。例如, 对于式 (4.1) 中的高度通道, 如果油门摇杆量 u_r 到垂直速度 v_{z_c} 通道是稳定的, 则在辨识前不需要设计控制器。然而, 如果考虑从油门摇杆量 u_r 到垂直位置 p_{z_c} 构成的通道, 那么需要在系统辨识前设计一个控制器。因为从垂直速度 v_{z_c} 到垂直位置 p_{z_c} 存在积分环节, 所以从油门摇杆量 u_r 到垂直位置 p_{z_c} 构成的通道是不稳定的, 辨识前需要附加一个 P 控制器。这里要说明的是, 并不需要反复调节控制器参数使系统性能变好, 只需达到稳定系统的效果即可。至于具体选择什么样的控制器, 可以通过先验知识确定出来。最终, 把带有控制器的闭环系统整体作为新的辨识对象。

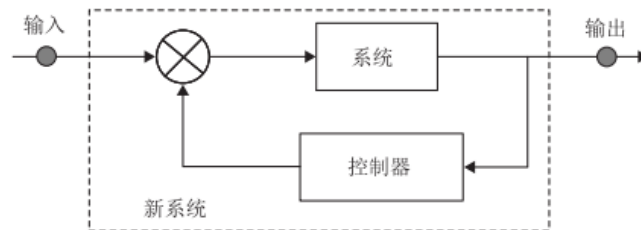


图 4.4 改进的开环实验方法

2) 高度通道

半自主自驾仪的高度通道一般以控制高度方向的速度为主，即遥控指令 u_T 控制的是 v_{z_c} 。当 $u_T = 0$ 时，可以使多旋翼保持当前高度。根据先验知识可知，带有半自主飞控的高度通道可以由式 (4.1) 表达。那么，根据式 (4.1)，相应的传递函数模型记为

$$p_{z_c}(s) = G_{p_{z_c}u_T}(s)u_T(s) \quad (4.4)$$

其中， $G_{p_{z_c}u_T}(s)$ 包含有一个一阶积分环节，是一个临界稳定的传递函数。该通道不适合进行直接辨识，因此需要一个 P 控制器，形式如下

$$u_T = k_{p_{z_c}}p_{z_c} + u_{p_{z_c}} \quad (4.5)$$

其中， $k_{p_{z_c}} > 0$ ，而 $u_{p_{z_c}}$ 定义为新的输入。注意，式 (4.1) 的输入系数是负的，即 $-k_{u_T}$ ，所以控制器的形式才如式 (4.4) 所示。在该控制器的作用下，式 (4.1) 表示为

$$\begin{aligned} \dot{p}_{z_c} &= v_{z_c} \\ \dot{v}_{z_c} &= -k_{u_T}k_{p_{z_c}}p_{z_c} - k_{v_{z_c}}v_{z_c} - k_{u_T}u_{p_{z_c}} \end{aligned} \quad (4.6)$$

其相应的传递函数可表示为

$$p_{z_c}(s) = G_{p_{z_c}u_{p_{z_c}}}(s)u_{p_{z_c}}(s) \quad (4.7)$$

这里，从 $u_{p_{z_c}}$ 到 p_{z_c} 的通道是一个待辨识的参数化模型。当记录该模型的数据时，高度数据 p_{z_c} （即该通道的输出）是由高度传感器测量得到的，如气压计或超声波测距仪。同时，新的输入 $u_{p_{z_c}}$ （即该通道的输入）也要记录下来。这时，具体的模型参数就可以通过系统辨识工具箱得到了，具体的操作过程将在接下来的部分进行介绍。此外， u_T 到 v_{z_c} 的通道是稳定的，如果垂直速度 v_{z_c} 是可以直接测量的，那么不需要额外设计控制器就可以直接辨识。在这种情况下，该通道的传递函数为

$$v_{z_c}(s) = G_{v_{z_c}u_T}(s)u_T(s) \quad (4.8)$$

进而有

$$p_{z_c}(s) = \frac{1}{s}G_{v_{z_c}u_T}(s)u_T(s) \quad (4.9)$$

3) 偏航通道

在可以得到半自主自驾仪反馈的偏航角速率时，偏航通道可以由式 (4.2) 表达，相应的传递函数为

$$\psi(s) = G_{\psi u_{\omega_z}}(s)u_{\omega_z}(s) \quad (4.10)$$

其中， $G_{\psi u_{\omega_z}}(s)$ 含有一阶积分环节，是不稳定的。因此，该通道需要用 P 控制器进行稳定。控制器形式为

$$u_{\omega_z} = -k_{\psi}\psi + u_{\psi} \quad (4.11)$$

其中， $k_{\psi} > 0$ ， u_{ψ} 为新的输入，那么式 (4.2) 变为

$$\begin{aligned}\dot{\psi} &= \omega_z \\ \dot{\omega}_z &= -k_{\omega_z}\omega_z - k_{\psi}\psi + k_{u_{\omega_z}}u_{\psi}\end{aligned}\quad (4.12)$$

相应的传递函数可以表达为

$$\psi(s) = G_{\psi u_{\psi}}(s)u_{\psi}(s) \quad (4.13)$$

辨识过程中需要同时记录偏航角 ψ 以及新输入 u_{ψ} 。磁力计和运动捕捉系统可以用来测量偏航角 ψ 。同时，模型参数可以通过系统辨识工具获得，具体实施过程将在接下来的章节中进行介绍。从 u_{ω_z} 到 ω_z 的通道是稳定的，所以如果能直接测量到 ω_z ，对此通道可以直接辨识，不需要额外设计控制器。在这种情况下，该通道相应的传递函数为

$$\omega_z(s) = G_{\omega_z u_{\omega_z}}(s)u_{\omega_z}(s) \quad (4.14)$$

进而有

$$\psi(s) = \frac{1}{s}G_{\omega_z u_{\omega_z}}(s)u_{\omega_z}(s) \quad (4.15)$$

4) 水平位置通道

半自主自驾仪会通过自身的角度传感器对姿态角 θ 和 ϕ 进行反馈，使水平指令 u_{θ} 和 u_{ϕ} 到 θ 和 ϕ 的通道分别是稳定的。该通道可以由式 (4.3) 来表达，这是先验知识。

水平位置通道的辨识是在对高度通道和偏航通道实现控制后进行的，因此在控制器的作用下，偏航通道满足 $\psi \approx \psi_d$ 。为了得到更好的辨识结果，通常将偏航角控制到合理的期望偏航角 ψ_d 上（最好假定 $\psi_d = 0$ ），因此可将 ψ_d 看成一个常数。这时， \mathbf{R}_{ψ} 为一个常数矩阵。可得到如下传递函数形式：

$$\begin{aligned}\mathbf{p}_h(s) &= \text{diag}\left(\frac{1}{s}, \frac{1}{s}\right)\mathbf{R}_{\psi}\mathbf{G}_{\mathbf{v}_{hb} \mathbf{u}_h}(s)\mathbf{u}_h(s) \\ &= \mathbf{R}_{\psi}\text{diag}\left(\frac{1}{s}, \frac{1}{s}\right)\mathbf{G}_{\mathbf{v}_{hb} \mathbf{u}_h}(s)\mathbf{u}_h(s)\end{aligned}\quad (4.16)$$

其中， $\mathbf{G}_{\mathbf{v}_{hb} \mathbf{u}_h}(s)$ 是一个待辨识的参数模型。由于 \mathbf{R}_{ψ} 引入了耦合，水平位置通道较其他两个通道的控制难度更大。因此，下面将为水平位置通道设计速度控制器。

如果半自主自驾仪考虑了速度反馈，并且 $\mathbf{K}_{\mathbf{v}_{hb}}$ 设置合理，那么传递函数 $\mathbf{G}_{\mathbf{v}_{hb} \mathbf{u}_h}(s)$ 是稳定的，可以直接进行系统辨识。如果没有引入速度反馈，有 $\mathbf{K}_{\mathbf{v}_{hb}} \approx \mathbf{0}_{2 \times 2}$ ，那么 $\mathbf{G}_{\mathbf{v}_{hb} \mathbf{u}_h}(s)$ 含有一个积分环节，是不稳定的。因此，在进行系统辨识之前，需要设计一个控制器来稳定这个系统。控制器形式如下：

$$\mathbf{u}_h = -\mathbf{K}'_{\mathbf{v}_{hb}}\mathbf{v}_{hb} + \mathbf{u}_{v_h} \quad (4.17)$$

其中， $\mathbf{u}_{v_h} = [u_{v_x} \quad u_{v_y}]^T$ 是一个新输入， $\mathbf{K}'_{\mathbf{v}_{hb}} \in \mathbb{R}^{2 \times 2}$ ，将控制器代入式 (4.3) 可得

$$\begin{aligned}\dot{\mathbf{v}}_{hb} &= -\mathbf{K}_{\mathbf{v}_{hb}}\mathbf{v}_{hb} - g\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\Theta_h \\ \dot{\Theta}_h &= \omega_{hb} \\ \dot{\omega}_{hb} &= -\mathbf{K}_{u_h}\mathbf{K}'_{\mathbf{v}_{hb}}\mathbf{v}_{hb} - \mathbf{K}_{\Theta_h}\Theta_h - \mathbf{K}_{\omega_{hb}}\omega_{hb} + \mathbf{K}_{u_h}\mathbf{u}_{v_h}\end{aligned}\quad (4.18)$$

对应的传递函数为

$$\mathbf{v}_{hb}(s) = \mathbf{G}_{\mathbf{v}_{hb}, \mathbf{u}_{vh}}(s) \mathbf{u}_{vh}(s) \quad (4.19)$$

水平位置可以由全球定位系统（Global Positioning System, GPS）接收机或室内定位系统（如室内运动捕捉系统）进行测量。速度信息可由 GPS 接收机直接得到，或通过对 GPS 接收机给出的位置信息进行滤波得到 \mathbf{v}_h 。进一步由 $\mathbf{v}_{hb} = \mathbf{R}_\psi^T \mathbf{v}_h$ 可以得到 \mathbf{v}_{hb} 。同时，新输入 \mathbf{u}_{vh} 也需要记录下来。模型参数可以利用系统辨识工具箱得到，具体实施过程可参考接下来的章节。实际上，期望的水平速度 $\mathbf{v}_{hb,d}$ 可以由期望的水平位置 $\mathbf{p}_{hb,d}$ 间接得到，因此它是控制器设计的间接目标。

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\7_MutUAVRemoteCtrl\1.System_identification_experiment\el.4\](#)

文件夹/文件名称	说明
battery_status_receive_test_2017b.slx	多旋翼 simulink 仿真文件
start_tello.m	初始化参数文件
el_3_TF_Bode_Verification_2017b.slx	多旋翼 simulink 仿真文件
generate_cifer_datapack.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 ^①	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：

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，如图 4.14 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，这里多旋翼各个通道控制器可以采用简单的 PD 控制器使其保持稳定即可。

Step 1:

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

Step 2:

打开本章所附带的文件夹“elle1.4”，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，用于与给定多旋翼进行交互所使用的 Simulink 模型；“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。

Step 3:

打开“elle1.4\ start_tello.m”初始化文件。在其中可以找到如表 4.4 所示的扫频数据加载代码。启动多旋翼的步骤在第 3 章 3.2 节中已经进行了详细说明，唯一需要注意的是：表 4.4 中第 3 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）。例如，表 4.4 所示的代码表示保存偏航通道的扫频数据。在完成上述的准备工作后，运行“elle1.4 start_tello.m”初始化文件。

Step 4:

打开“elle1.4\ e1_4_cifer_model.slx”模型文件，其期望输入模块如图 4.24 所示。该模块用于生成单通道扫频所使用的位置指令，其中虚线框①所示模块用于生成扫频数据；虚线框②表示是否引入扫频信号，图 4.24 中所示案例为水平前向通道引入扫频信号，其他通道均设置为稳定值，即多旋翼在该通道的“平衡点”附近。利用 4.1.1 节中的控制器设计方法，在“elle1.4\ 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 PD 控制器即可，如图 4.25 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。特别需要注意的是，因为偏航角的控制更加底层也就更加容易，因此只需一个 P 控制器就可以使得该通道稳定。

表 4.4 “start_tello.m” 中扫频数据加载代码

```

1 %扫频信号选择
2 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);
3 model_channel = Model_Channel.YAW;
4
5 PARAM_time_start = 3;%3s后产生扫频信号
6                 % ns后多旋翼稳定
7 PARAM_C1 = 4.0;
8 PARAM_C2 = 0.0187;
9 PARAM_w_min = 0.5;%0.3 0.4 0.5
10 PARAM_w_max = 12;%10 11 12
    
```

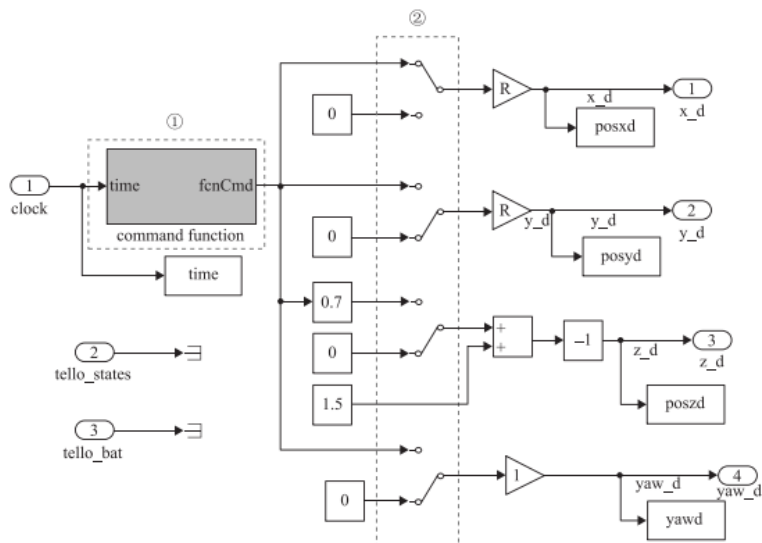


图 4.24 期望输入模块示意图, Simulink 文件详见 “e1_4_cifer_model.slx”

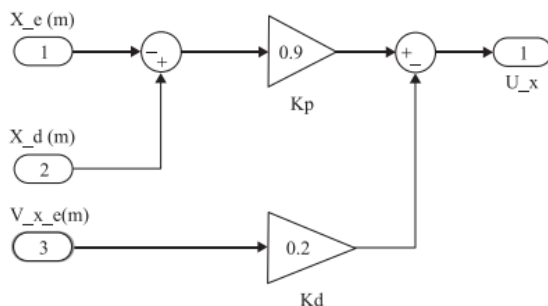


图 4.25 实飞实验水平前向通道 PD 控制器, Simulink 文件详见 “e1_4_cifer_model.slx” 中 “control system” 模块

Step 5:

(1) 启动 OptiTrack

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

(2) 启动 tello_driver

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

(3) 起飞 Tello

打开一个新终端, 运行命令 “roslaunch tello Tello_takeoff_all”, 可以看到一架多旋翼起飞

并保持悬停在正上方高度 1m 的位置。

(4) 运行 MATLAB 控制程序

运行“el_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，需要修改表 4.4 中第 3 行代码“Model_Channel.YAW”，与需要辨识通道相对应。

(5) 降落 Tello

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

Step 6:

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 软件的使用方法请参考附录 A。

以水平前向通道和水平侧向通道为例，由式(4.16)可知，水平通道的传递函数是一个三阶系统，考虑延时的影响，不断调节分子的阶数，使代价函数最小，最终得到传递函数模型如式(4.31)和式(4.32)所示。

(1) 水平前向通道

$$G_{p_{x_c}u_{v_x}} = \frac{1.24431s + 1.75269}{s^3 + 2.63008s^2 + 2.37405s + 1.95897} e^{-0.3778s} \quad J = 52.074 \quad (4.31)$$

相应地，实飞中的水平前向通道扫频输入/输出对比如图 4.26所示，扫频实飞图如图 4.27所示。可以发现水平前向通道的输出可以较好地跟随期望输入，特别是在低频阶段。需要注意的是，我们设计的控制器并不需要输出跟踪上输入，仅需要保持系统稳定即可。

(2) 水平侧向通道

$$G_{p_{y_c}u_{v_y}} = \frac{1.21543s + 2.06629}{s^3 + 2.80018s^2 + 2.54762s + 2.23605} e^{-0.3715s} \quad J = 58.173 \quad (4.32)$$

相应地，实飞实验中的水平侧向通道扫频输入/输出对比如图 4.28 所示，实飞图如图 4.29 所示。从图中可以发现水平侧向通道的输出可以较好地跟随期望输入，特别是在低频阶段。

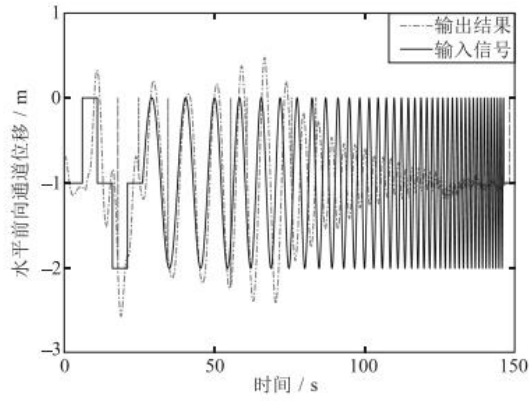


图 4.26 实飞实验水平前向通道扫频输入/输出对比图

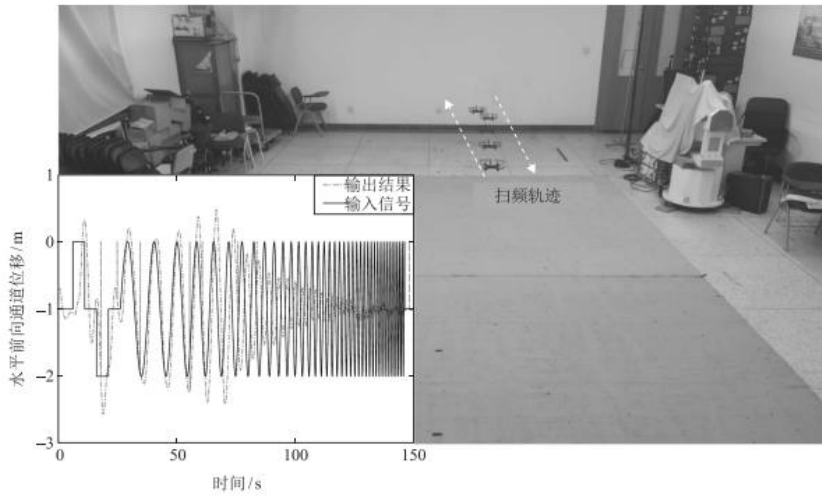


图 4.27 实飞实验水平前向通道扫频实飞图（影像进行了叠加）

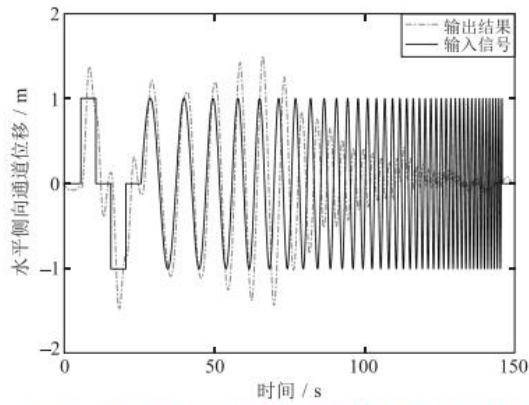


图 4.28 实飞实验水平侧向通道扫频输入/输出对比图

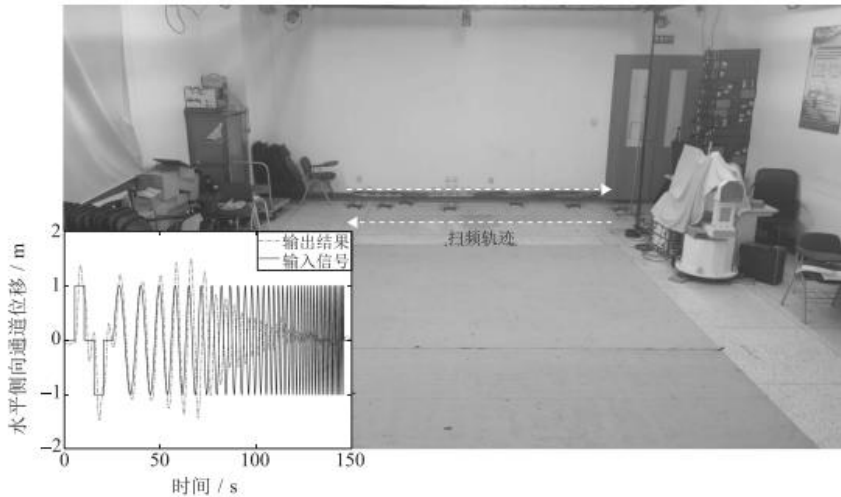


图 4.29 实飞实验水平侧向通道扫频实飞图（影像进行了叠加）

(3) 模型验证

针对上面获得的给定多旋翼模型, 还需要重新对其进行模型验证, 以验证模型的准确性。按照 4.3.2.1 节中的步骤重新进行实验, 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和获得的传递函数模型输入相同的位置扫频控制指令, 对比获得的位置输出的拟合效果。

① 水平前向通道

水平前向通道对比结果如图 4.30 所示, 从图中可以看出, 在水平前向通道中, 给定多旋翼和获得的传递函数模型的输出基本上一致, 特别在低频阶段效果更好, 高频阶段误差有些大。

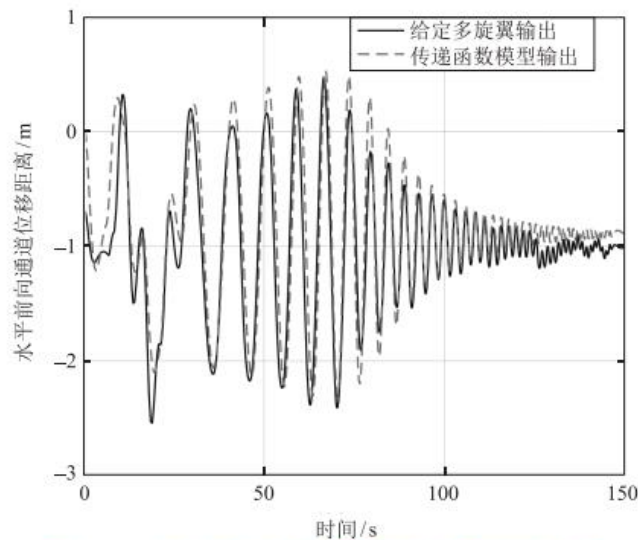


图 4.30 实飞实验水平前向通道扫频模型验证对比图

② 水平侧向通道

水平侧向通道对比结果如图 4.31 所示。从图中可以看出: 在水平侧向通道中给定多旋翼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。

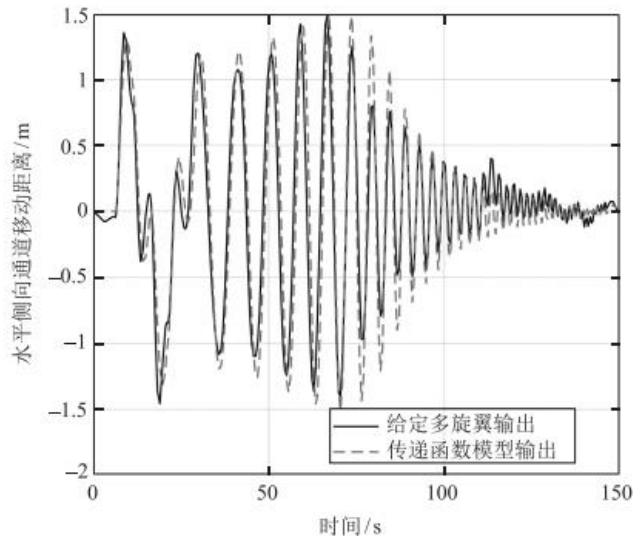


图 4.31 实飞实验水平侧向通道模型验证对比图

实验结论：

本章的实飞扫频系统辨识实验可以获得给定多旋翼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 6 章的跟踪控制器实验，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。

（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: ****