

1. 实验名称及目的

1.1. 实验名称

系统辨识实验

1.2. 实验目的

基础实验：（1）了解固定翼无人机制导模型，掌握数据时间配准预处理方法（包括起止时间对齐和采样频率统一），学习 MATLAB 系统辨识工具箱用法。（2）使用速度、航向和高度三个通道采集好的输入输出数据，使用 MATLAB 系统辨识工具箱完成固定翼制导

$$\text{模型的辨识, 即式} \begin{cases} \dot{p}_{x_e} = V_e \cos \psi + v_{wx} \\ \dot{p}_{y_e} = V_e \sin \psi + v_{wy} \\ \psi = \chi + f(v_{wx}, v_{wy}) \\ V_e(s) = G_{V_e}(s)V_{ed}(s) \\ \chi(s) = G_\chi(s)\chi_d(s) \\ H(s) = G_H(s)H_d(s) \end{cases} \quad (1)$$

分析实验：以俯仰力矩通道为例，辨识式

$$\begin{aligned} m_{ax} &= \frac{1}{2} \rho V_a^2 S b \left(C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{b}{2V_a} \omega_{x_b} + C_{l_r} \frac{b}{2V_a} \omega_{z_b} + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \right) \\ m_{ay} &= \frac{1}{2} \rho V_a^2 S c \left(C_{m_0} + C_{m_\alpha} \alpha + C_{m_p} \frac{c}{2V_a} \omega_y + C_{m_{\delta_e}} \delta_e \right) \quad \text{中的 } C_{m_0}, C_{m_\alpha}, \\ m_{az} &= \frac{1}{2} \rho V_a^2 S b \left(C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{b}{2V_a} \omega_{x_b} + C_{n_r} \frac{b}{2V_a} \omega_{z_b} + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \right) \end{aligned}$$

$C_{m_q}, C_{m_{\delta_e}}$ ，了解气动参数辨识的流程和思想。对比分析不同幅值或持续时间的偶极方波或频率变化范围的扫频信号作为激励信号的辨识效果。

设计实验：在固定翼无人机 Simulink 模型中，设计合适的输入信号，激励相应通道，采集数据并处理，辨识滚转力矩通道的气动导数，即式

$$\begin{aligned} m_{ax} &= \frac{1}{2} \rho V_a^2 S b \left(C_{l_0} + C_{l_\beta} \beta + C_{l_p} \frac{b}{2V_a} \omega_{x_b} + C_{l_r} \frac{b}{2V_a} \omega_{z_b} + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \right) \\ m_{ay} &= \frac{1}{2} \rho V_a^2 S c \left(C_{m_0} + C_{m_\alpha} \alpha + C_{m_p} \frac{c}{2V_a} \omega_y + C_{m_{\delta_e}} \delta_e \right) \quad \text{中的} \\ m_{az} &= \frac{1}{2} \rho V_a^2 S b \left(C_{n_0} + C_{n_\beta} \beta + C_{n_p} \frac{b}{2V_a} \omega_{x_b} + C_{n_r} \frac{b}{2V_a} \omega_{z_b} + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \right) \end{aligned}$$

$C_{l_0}, C_{l_\beta}, C_{l_p}, C_{l_r}, C_{l_{\delta_a}}, C_{l_{\delta_r}}$ ，并验证得到的非线性模型的吻合程度。

硬件在环仿真实验：在硬件在环仿真环境下，完成固定翼无人机的俯仰力矩、滚转力矩和偏航力矩通道的气动参数辨识，与分析实验和设计实验类似，区别在于需要使用 Pixhawk 自驾仪系统进行实验得到所需的数据。

气动参数辨识硬件在环仿真实验：在硬件在环仿真环境下，完成固定翼无人机的滚转力矩通道的气动参数辨识。与分析实验和设计实验类似，区别在于需要使用 CubePilot/Pixhawk 自驾仪进行实验得到所需的数据。

1.3. 关键知识点

关键知识点 1: [Cm_DataPreprocess.mlx](#) 代码讲解

数据处理代码位于“Cm_DataPreprocess.mlx”中，如下代码段 5.2，第 1 行为清空工作区，第 2 行为从“.mat”文件中读取数据并存储为“data”，第 4-9 行为将“data”拆分为舵偏角（修正前）、俯仰角速率、迎角、空速、动压、俯仰角加速度数据。第 11-17 行为分别对各个状态数据进行窗口长度为 15 的滑动平均滤波处理，其中“movmean”函数为 MATLAB 提供的移动均值函数，其语法为：“M = movmean(A,k)”。该函数返回由局部“k”个数据点的均值组成的数组，其中每个均值是基于“A”的相邻元素的长度为“k”的滑动窗计算得出。当“k”为奇数时，窗口以当前位置的元素为中心。当“k”为偶数时，窗口以当前元素及其前一个元素为中心。当没有足够的元素填满窗口时，窗口将自动在端点处截断。当窗口被截断时，只根据窗口内的元素计算平均值。

```
clear;clc;
data = load("Cm_Data.mat");
% Split data
alpha = data.data(2,:);
dE = data.data(3,:);
q = data.data(4,:);
Airspeed = data.data(5,:);
alpha_dot = data.data(6,:);
wby_dot = data.data(7,:);
% Moving average filter
window = 15;
data_filtered.alpha=movmean(alpha,window);
data_filtered.dE=movmean(dE,window);
data_filtered.wby=movmean(q,window);
data_filtered.Airspeed=movmean(Airspeed,window);
data_filtered.alpha_dot=movmean(alpha_dot,window);
data_filtered.wby_dot=movmean(wby_dot,window);
```

关键知识点 2: [Cm_Identification.mlx](#) 代码讲解

第 1 行为清空工作区，第 2 行为从“.mat”文件中读取数据并存储为“data”；第 3-8 行为将“data”拆分为俯仰角速率、俯仰角加速度、空速、舵偏角、迎角和迎角变化率等数据；第 10-14 行输入了该通道辨识所需的无人机转动惯量以及几何参数等；第 15-24 行为计算动

压，根据固定翼姿态动力学方程
$$\begin{aligned} \mathbf{J} \cdot {}^b \omega &= -{}^b \omega \times (\mathbf{J} \cdot {}^b \omega) + {}^b \mathbf{m} \\ {}^b \mathbf{m} &= \mathbf{g}_y + {}^b \mathbf{m}_a + {}^b \mathbf{m}_p \end{aligned}$$
 计算俯仰力矩系数，以及对

迎角变化率和俯仰角速率进行修正；第 26 行生成零升力矩项；第 27 行将用于辨识的状态量数据合并成矩阵；第 28 行调用 MATLAB 提供的“regress”函数进行辨识。这里“regress”函数为多元线性回归函数，其语法为“[b,bint,r,rint,stats]=regress(y,X)”，该函数返回：向量“b”为“y”中的响应对矩阵“X”中的预测变量的多元线性回归的系数估计值；“bint”为系数估计值的 95 % 置信区间的矩阵；“r”为残差向量；“rint”为可用于诊断离群值的区间，离群值是指与集合中其它观测值差异较大的任何数据点；“stats”包含 R^2 统计量、以及其它统计量指标（本实验中未用到）。需要注意的是要计算具有常数项（截距）的模型的系数估计值，在矩阵 X 中要包含一个由 1 构成的列，在此处即为零升力矩项。第 30 行为使用“stepwise”函数进行辨识。“stepwise”函数本质上与“regress”函数一样，也是多元线性回归，但是前者进行了可视化显示，可以逐次加入被辨识的数据并进行显著性检测，也可以显示“regress”函数得出的各种结果。第 31 行为使用辨识出的气动参数计算俯仰力矩系数。

```
clear;
load('Cm_DataFiltered.mat'); %load data
wby=data_filtered.wby(5500:7500)'; %split data
wby_dot=data_filtered.wby_dot(5500:7500)';
Va=data_filtered.Airspeed(5500:7500)';
deltaE=data_filtered.dE(5500:7500)';
alpha=data_filtered.alpha(5500:7500)';
dalpha=data_filtered.alpha_dot(5500:7500)';
% Aircraft geometric parameters
Iy=0.144;
c=0.3;
b=1.2;
S=0.32;
rho = 1.225;
qbar=zeros(2001,1);
Cm=zeros(2001,1);
dalpha_after=zeros(2001,1);
q_after=zeros(2001,1);
for i=1:2001 %calculate qbar Cm dalpha_after q_after
    qbar(i)=0.5*rho*Va(i)^2;
    Cm(i)=(Iy*wby_dot(i))/(qbar(i)*c*S);
    dalpha_after(i)=dalpha(i)*c/(2*Va(i));
    q_after(i)=wby(i)*c/(2*Va(i));
end
% regress
Cm0=ones(2001,1);
XCm=[Cm0,alpha,dalphi_after,q_after,deltaE];
[bCm,bintCm,rCm,rintCm,statsCm]=regress(Cm,XCm);
% stepwise
stepwise(XCm,Cm);
Cm_identify=XCm*bCm;
```

关键知识点 3: [CI_DataPreprocess.mlx](#) 代码讲解

与分析实验相同，选择窗口长度为 15 的滑动平均滤波器对数据进行处理。相应的数据处理代码位于“CI_DataPreprocess.mlx”，如下代码段所示。前 2 行和 12-19 行与分析实验中相应代码别无二致。需要说明的是，第 4-10 行为将“data”拆分为辨识需要的侧滑角、副翼

舵偏角、方向舵舵偏角、空速、滚转角速率、偏航角速率以及滚转角加速度数据。

```
clear;clc;
data = load("Cl_Data.mat");
% split data
beta = data.data(2,:);
dA = data.data(3,:);
dR = data.data(4,:);
Airspeed = data.data(5,:);
wbx = data.data(6,:);
wbz = data.data(7,:);
wbx_dot = data.data(8,:);
% Average sliding filter
window = 15;
data_filtered.beta=movmean(beta,window);
data_filtered.dA=movmean(dA,window);
data_filtered.dR=movmean(dR,window);
data_filtered.Airspeed=movmean(Airspeed,window);
data_filtered.wbx=movmean(wbx,window);
data_filtered.wbz=movmean(wbz,window);
data_filtered.wbx_dot=movmean(wbx_dot,window);
```

关键知识点 4: [HeightChannel OffboardController.py](#) 代码讲解

第 1 行为设定路径点参考速度，第 2-13 行为构造循环生成航点队列，其中第 3 行为设定北向航点，第 4 行为设定东向航点，第 5-6 行为设定第 0-5 个航点高度为 100m，第 7-8 行为设定第 6-21 个航点高度为 120m，第 9-10 行为设定第 22-41 个航点高度为 80m，第 11-12 行为设定其余航点高度为 100m，第 13 行为在每次循环中用“append”函数将三维航点行向量按行拼接，从而得到一个 22 行 3 列的航点列表。

```
v = 40
for i in range(41):
    x = -v*i
    y = 0
    if i <= 5:
        z = -100
    elif 5 < i <= 21:
        z = -120
    elif 21 < i <= 41:
        z = -80
    else:
        z = -100
    missionPoints.append([x, y, z])
```

2. 实验效果

基础实验：(1) 导入固定翼飞行日志文件，并进行预处理。(2) 使用 MATLAB 系统辨识工具箱进行系统辨识。(3) 验证辨识结果。

分析实验：分析不同幅值或持续时间的偶极方波或频率变化范围的扫频信号作为激励信号的辨识效果。

设计实验：使用最小二乘法和逐步回归法辨识气动参数。

硬件在环仿真实验：在硬件在环仿真环境下，完成固定翼无人机的俯仰力矩、滚转力矩和偏航力矩通道的气动参数辨识。

3. 文件目录

	文件夹/文件名称	说明
HeightChannel 1	log_trajectory_setpoint_0.csv	飞行日志中高度等指令信号和实际信号的记录。
	log_vehicle_local_position_	
	HeightChannel_Input.mat	从日志文件导出的数据。
	HeightChannel_Output.mat	
	HeightChannel_DataPreprocess.mlx	数据预处理脚本。
	HeightChannel_Identification.sid	存储了辨识结果。
SpeedChannel	log_airspeed_0.csv	飞行日志中速度等指令信号和实际信号的记录。
	log_tecs_status_0.csv	
	SpeedChannel_Input.mat	从日志文件导出的数据。
	SpeedChannel_Output.mat	
	SpeedChannel_DataPreprocess.mlx	数据预处理脚本。
	SpeedChannel_Identification.sid	存储了辨识结果。
HeadingChan nel	log_trajectory_setpoint_0.csv	飞行日志中航向等指令信号和实际信号的记录。
	log_vehicle_local_position_0.csv	
	HeadingChannel_Input.mat	从日志文件导出的数据。
	HeadingChannel_Output.mat	
	HeadingChannel_DataPreprocess.mlx	数据预处理脚本。
	HeadingChannel_Identification.sid	存储了辨识结果。
e3-2	model	小型固定翼模型
	Cm_Data.mat	运行模型得到的辨识所需的原始数据
	Cm_DataFiltered.mat	使用“Cm_DataFiltered.mat”进行俯仰力矩通道气动参数辨识的脚本
	Cm_DataPreprocess.mlx	直接用于辨识的“Cm_DataFiltered.mat”数据
	Cm_Identification.mlx	处理数据的代码
e3-3	model	小型固定翼模型
	Cm_Data.mat	运行模型得到的辨识所需的原始数据
	Cm_DataFiltered.mat	使用“Cm_DataFiltered.mat”进行俯仰力矩通道气动参数辨识的脚本
	Cm_DataPreprocess.mlx	直接用于辨识的“Cm_DataFiltered.mat”数据
	Cm_Identification.mlx	处理数据的代码
HeightChanne 1	log.ulg	飞控硬件中导出的日志。
	log_vehicle_local_position_0.csv	飞行日志中高度等指令信号和实际信号的记录。
	log_vehicle_local_position_0.csv	

	HeightChannel_Input.mat	从日志文件导出的数据。
	HeightChannel_Output.mat	
	HeightChannel_DataPreprocess.mlx	数据预处理脚本。
	HeightChannel_Identification.sid	存储了辨识结果。
Aerodynamic Parameter_Ide ntification	AircraftMathworksAttCtrl.py	固定翼无人机硬件在环实验控制脚本
	SmallFixedWingUAVIdentificationHIL.dll	待辨识的固定翼无人机模型 dll 文件
	UAVIdentification.bat	硬件在环实验一键启动脚本
	log	飞行日志数据文件夹
	CI_DataPreprocess.mlx	输入输出数据预处理脚本
	CI_Identification.mlx	滚转力矩通道气动参数辨识脚本

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	Pixhawk 自驾仪	1
3	MATLAB R2022b 及以上	遥控器	1
4		遥控器接收机	1

①：推荐配置请见：<https://doc.rflysim.com>

5. 实验步骤

5.1. 基础实验

(1) 导入固定翼飞行日志文件，并进行预处理

1. 实验步骤以高度通道辨识为例进行说明。高度通道制导模型辨识的目的是辨识式 (1) 中的 $G_H(s)$ 。使用 MATLAB 自带的“数据导入”功能可以方便地将“log_trajectory_setpoint_0.csv”文件导入 MATLAB 工作区并储存为“[HeightChannel_Input.mat](#)”文件，需要注意的是，在如图所示的导入数据界面中，将“输出类型”选为“数值矩阵”，对于“无法导入的单元格”设置为“排除具有以下项的行”，然后选定代表时间戳的“timestamp”和代表高度设定值（输入）的“z”列，点击“导入所选内容”即可将数据导入到工作区中，再进行适当的命名并保存为“.mat”格式。对“log_vehicle_local_position_0.csv”文件的处理与上述过程相似。

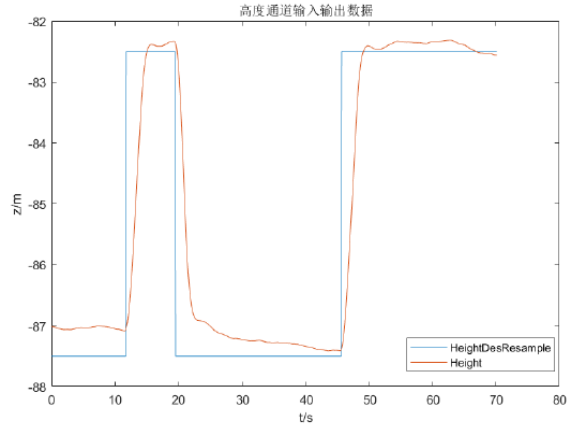
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值
1	timestamp	x	y	yaw	yawspeed	vx	vy	vz	accelerati...	accelerati...	accelerati...	jerk[0]	jerk[1]	jerk[2]	thrust[0]	thrust[1]	thrust[2]	
2	5588000	0.0011952...	-0.005644...	0.0560281...	1.5777836	0.0	-0.000187...	0.0007393...	0.69920003	-0.009995...	0.0579746...	0.0	0.0	0.0	-4.0	nan	nan	
3	6608000	0.0011553...	-0.005424...	0.0556089...	1.577726	0.0	0.0001881...	-0.000740...	0.69920003	0.0100383...	-0.058220...	-0.079996...	0.0	0.0	0.0	-4.0	nan	nan
4	7788000	-0.0003816...	-0.002349...	0.0551871...	1.578058	0.0	0.000193...	-0.0007510...	0.69920015	-0.0103369...	0.060233...	-0.079992...	0.0	0.0	0.0	-4.0	nan	nan
5	8988000	-0.003768...	0.0012328...	0.0566704...	1.5778481	0.0	4.957846e...	-0.0007619...	0.69920015	2.4795532...	0.08258106	-0.079992...	0.0	0.0	0.0	-4.0	nan	nan
6	7188000	-0.005215...	-0.003645...	0.0608831...	1.5778263	0.0	8.389741e...	-0.0007719...	0.69920003	4.196167e...	-0.065030...	-0.079996...	0.0	0.0	0.0	-4.0	nan	nan
7	7388000	-0.006215...	-0.0061653...	0.06474302	1.5776709	0.0	1.0486302...	-0.0007806...	0.6992001	5.2452097...	0.0675830...	-0.079994...	0.0	0.0	0.0	-4.0	nan	nan
8	7588000	-0.006609...	-0.0069887...	0.0697711...	1.5777553	0.0	2.8968421...	-4.781644...	0.6672223	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
9	7788000	-0.006599...	-0.0069712...	0.07238165	1.5773376	0.0	7.5315096...	-0.000127...	0.60055846	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
10	7988000	-0.006579...	-0.0069377...	0.07423448	1.5770252	0.0	0.0001216...	-0.000207...	0.5338955	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
11	8188000	-0.006550...	-0.0068883...	0.07492957	1.5770103	0.0	0.0001680...	-0.000286...	0.4672286	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
12	8388000	-0.006512...	-0.0068229...	0.0668696	1.5774044	0.0	0.0002143...	-0.000365...	0.4005623	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
13	8588000	-0.006464...	-0.0067416...	0.0588177...	1.5775279	0.0	0.0002607...	-0.000446...	0.33389568	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
14	8788000	-0.006408...	-0.0066443...	0.0523980...	1.577529	0.0	0.0003070...	-0.000526...	0.26722908	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
15	8988000	-0.006341...	-0.0065311...	0.04914136	1.5776047	0.0	0.0003534...	-0.000605...	0.20056248	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
16	9188000	-0.006266...	-0.0064026...	0.0422293	1.5779335	0.0	0.0003997...	-0.000685...	0.13389587	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
17	9388000	-0.006182...	-0.0062569...	0.0369466	1.5780526	0.0	0.0004481...	-0.000765...	0.0672292	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
18	9588000	-0.006088...	-0.0060958...	0.0337519	1.5782366	0.0	0.0004924...	-0.000845...	0.0005625	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
19	9788000	-0.005985...	-0.0059188...	0.03215249	1.5782241	0.0	0.0005387...	-0.000924...	0.066104...	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
20	9988000	-0.005872...	-0.0057959...	0.0279584	1.5779749	0.0	0.0005851...	-0.001004...	0.1327708	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
21	10188000	-0.005751...	-0.0055176...	0.02333974	1.5777618	0.0	0.0006314...	-0.001094...	-0.199437...	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
22	10388000	-0.005620...	-0.0052922...	0.0169208	1.577472	0.0	0.0006778...	-0.001163...	-0.266104	0.0002317...	-0.000398...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
23	10588000	-0.005481...	-0.0050534...	0.0120214	1.5773691	0.0	0.0007006...	-0.001203...	-0.33270...	-4.768371...	7.1525574...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
24	10788000	-0.005341...	-0.00481305	0.00785004	1.5772434	0.0	0.0006989...	-0.001200...	-0.399437...	-1.215934...	1.9311905...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
25	10988000	-0.005201...	-0.0045734...	0.0045060	1.5772492	0.0	0.0006956...	-0.001195...	-0.466103...	-1.978874...	3.2901764...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
26	11188000	-0.005062...	-0.0043351...	0.0033159	1.5773844	0.0	0.0006907...	-0.001187...	-0.53270...	-2.884864...	4.839897e...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
27	11388000	-0.004925...	-0.0040987...	0.0040940	1.5772805	0.0	0.0006838...	-0.001175...	-0.599437...	-3.910064...	6.604195e...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan
28	11588000	-0.004789...	-0.0038651...	0.0036906	1.5771601	0.0	0.0006747...	-0.001160...	-0.666103...	-5.125999...	8.6545944...	-0.333333...	0.0	0.0	0.0	0.0	nan	nan

2.按照以上步骤，从“log_trajectory_setpoint_0.csv”和“log_vehicle_local_position_0.csv”文件中分别导出“HeightChannel Input.mat”和“HeightChannel Output.mat”，分别对应式(1)中的 $H_a(s)$ 和 $H(s)$ 。接下来需要对输入输出数据进行时间配准处理。此处的时间配准需要解决两方面的问题，一是起始和终止时间的配准，二是进行采样频率的配准。对于起始和终止时间的配准，需要观察输入输出数据，选取适用于辨识的数据段，例如高度通道选取无人机结束爬升飞行，进行上下阶跃飞行的阶段，从而确定起始时间和终止时间，进而在输入输出数据中截取出相应的数据。对于采样频率的配准，已知飞控日志时间戳的单位为微秒，观察输入输出数据时间戳，可以知道输入数据的采样频率是5Hz(由于飞控硬件刚启动时会进行初始化，前几个数据可能采样频率可能小于5Hz，初始化完成后会稳定在5Hz)，输出数据的采样频率是10Hz。因此需要通过重采样的方式使输入输出数据时间戳对准，即增大输入数据的采样频率至10Hz，由于输入数据在下次更新前会保持上次的值，因此只需要使增加出的输入数据采样点的值与上一时刻保持一致，即可得到辨识所需的“Height”(实际高度)与“HeightDesResample”(重采样的期望高度)。

在“HeightChannel DatePreprocess.mlx”中的关键代码如下代码段表，其中：第1行为清空工作区；第2-3行为导入数据；第4-5行为截取数据并转换为矩阵格式；6-10行为重采样。

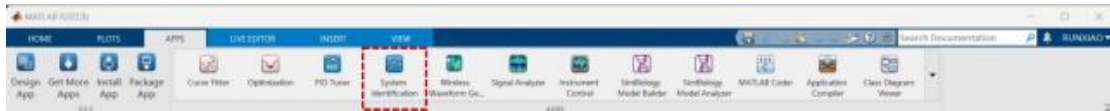
```
clear
HeightData1=importdata("HeightChannel_Input.mat");
HeightData2=importdata("HeightChannel_Output.mat");
HeightDes=table2array(HeightData1(320:671,2));
Height=table2array(HeightData2(800:1501,2));
HeightDesResample=[];
for i=1:2:size(Height)-1
    HeightDesResample(i,1)=HeightDes((i+1)/2,1);
    HeightDesResample(i+1,1)=HeightDes((i+1)/2,1);
end
```

3.使用“plot”指令绘制出“HeightDesResample”和“Height”变量，即可得到的用于辨识的输入输出数据曲线，如图所示。从图可以看出，高度期望“HeightDesResample”曲线为在-82.5与-87.5两个值之间阶跃变化的，而实际高度“Height”曲线则是跟踪期望高度曲线变化，但是存在一定的过渡过程和跟踪误差。

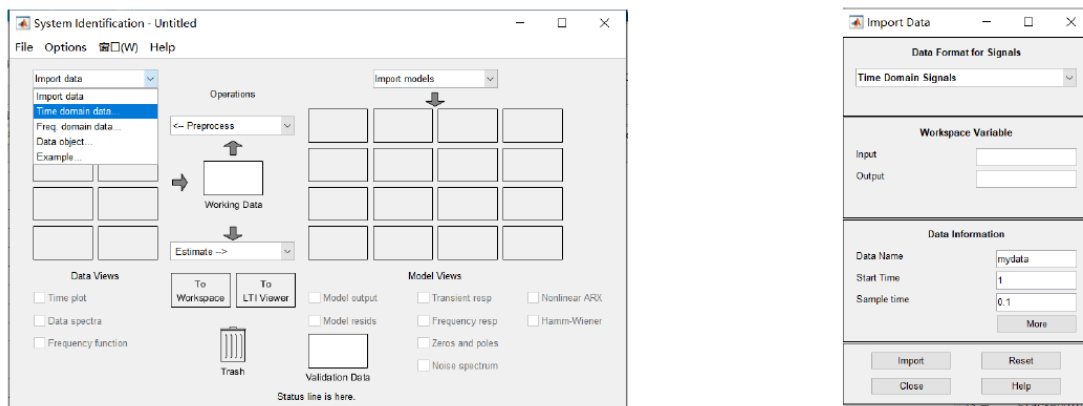


(2) 使用 MATLAB 系统辨识工具箱进行系统辨识

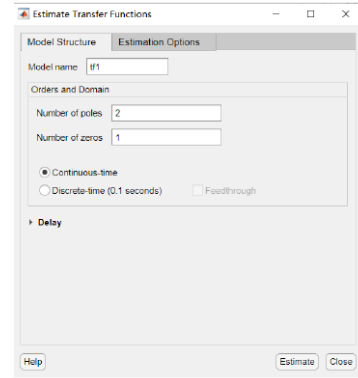
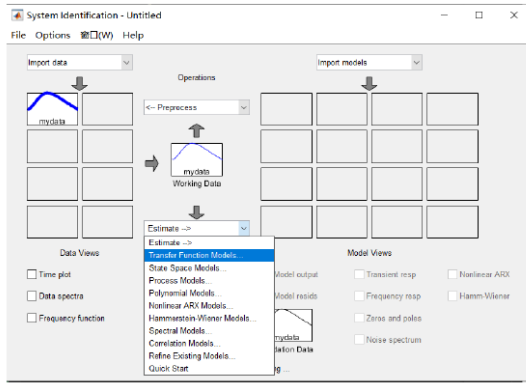
1. 如图, 打开 MATLAB 的系统辨识工具箱“APP-System Identification”。



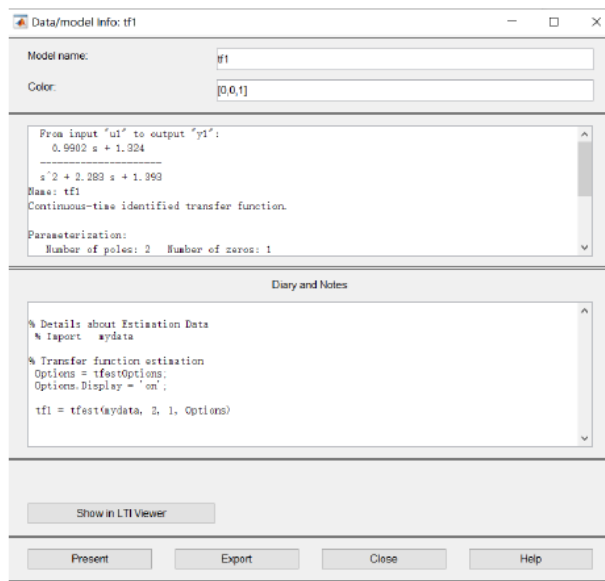
2. 如图, 点击“Import data”, 弹出窗口后, 选择时域数据“Time domain data”, 设置输入“input”为“HeightDesResample”, 输出“output”为“Height”, 设置采样时间“Sample time”为 0.1s, 点击“Import”即可导入数据。



3. 如图, 点击“Estimate”, 选择传递函数模型“Transfer Function Models”, 弹出界面后, 设置模型的零点个数“Number of zeros”为 0, 极点个数“Number of poles”为 1,

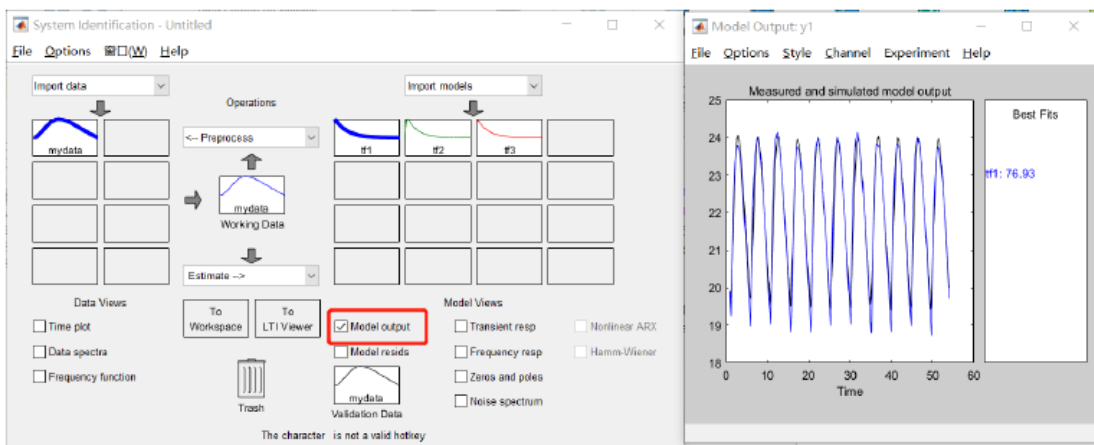


选择连续传递函数“Continuous-time”，点击“Estimate”，辨识传递函数模型，返回工具箱界面，双击建立好的系统模型就可以显示得到传递函数以及匹配度，如图。



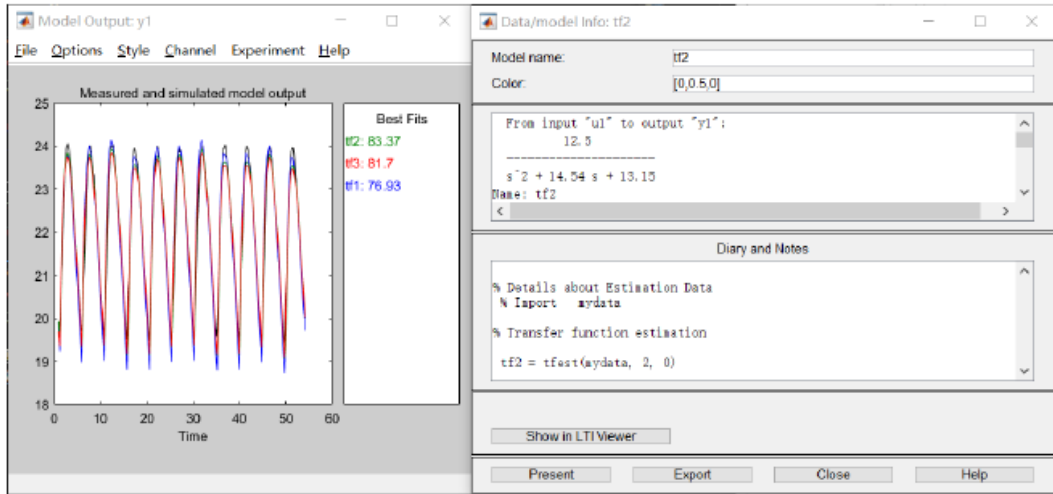
(3) 验证辨识结果

通过(2)得到传递函数后，选择已建立好的模型，在工具箱界面右下角勾选“Model output”即可得到模型在输入信号作用下的响应曲线和匹配度，如图所示。通常来说，匹配度大于70%则认为辨识效果较好。



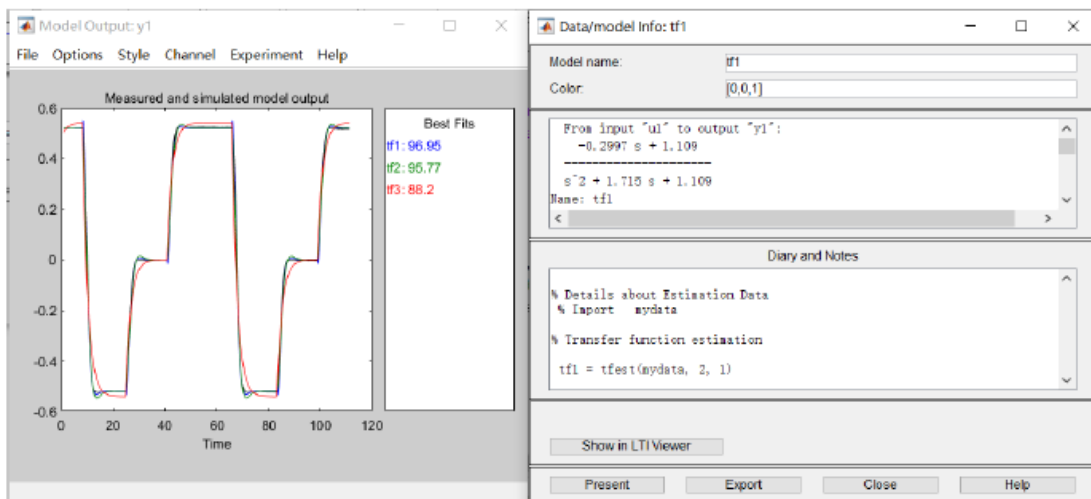
1. 速度通道

速度通道辨识结果如图，其中“tf1”、“tf2”、“tf3”曲线分别代表选择不同数量的零极点得到的辨识结果，对比可得，“tf2”即 2 个极点 0 个零点建立的传递函数模型匹配度最高，达到了 83.37%，符合预期。



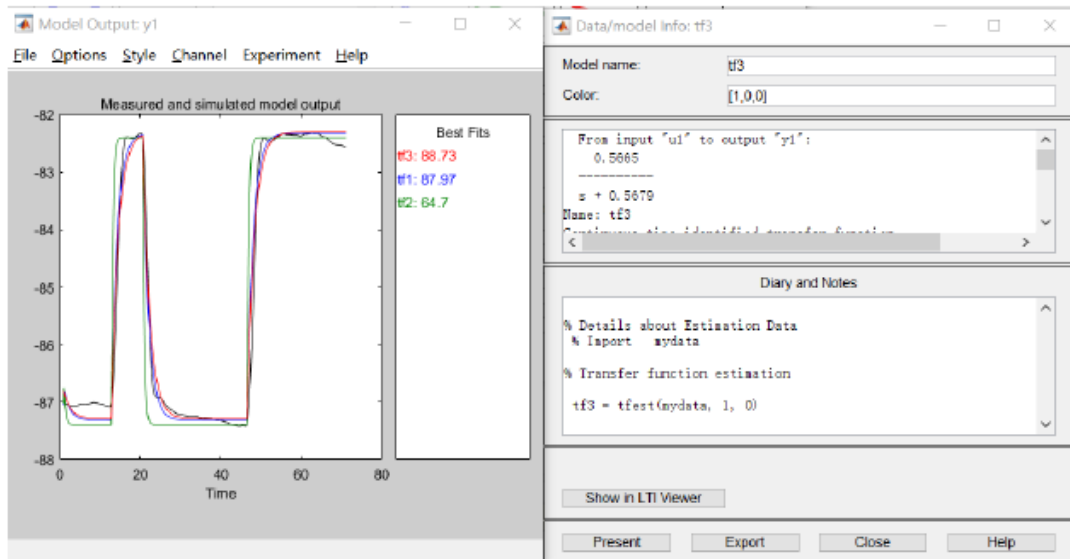
2. 航向通道

航向通道辨识结果如图，其中“tf1”、“tf2”、“tf3”曲线分别代表选择不同数量的零极点得到的辨识结果，对比可得，“tf1”即 2 个极点 1 个零点建立的传递函数模型匹配度最高，达到了 96.95%，符合预期。



3. 高度通道

高度通道辨识结果如图，其中“tf1”、“tf2”、“tf3”曲线分别代表选择不同数量的零极点得到的辨识结果，对比可得，“tf3”即 1 个极点 0 个零点建立的传递函数模型匹配度最高，达到了 88.73%，符合预期。



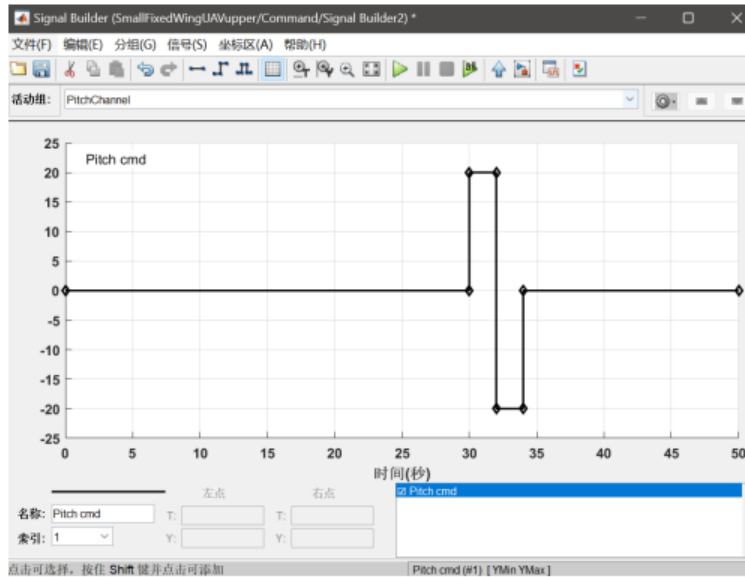
综上所述，可以得到辨识出的制导模型

$$\left\{ \begin{array}{l} \dot{p}_{x_e} = V_e \cos \psi + v_{wx} \\ \dot{p}_{y_e} = V_e \sin \psi + v_{wy} \\ \psi = \chi + f(v_{wx}, v_{wy}) \\ V_e(s) = G_{V_e}(s)V_{ed}(s) = \frac{12.5}{s^2 + 14.54s + 13.15} V_{ed}(s) \\ \chi(s) = G_{\chi}(s)\chi_d(s) = \frac{-0.2997s + 1.109}{s^2 + 1.715s + 1.109} \chi_d(s) \\ H(s) = G_H(s)H_d(s) = \frac{0.5665}{s + 0.5679} H_d(s) \end{array} \right.$$

5.2. 分析实验

(1) 设置不同幅值和持续时间的偶极方波信号输入

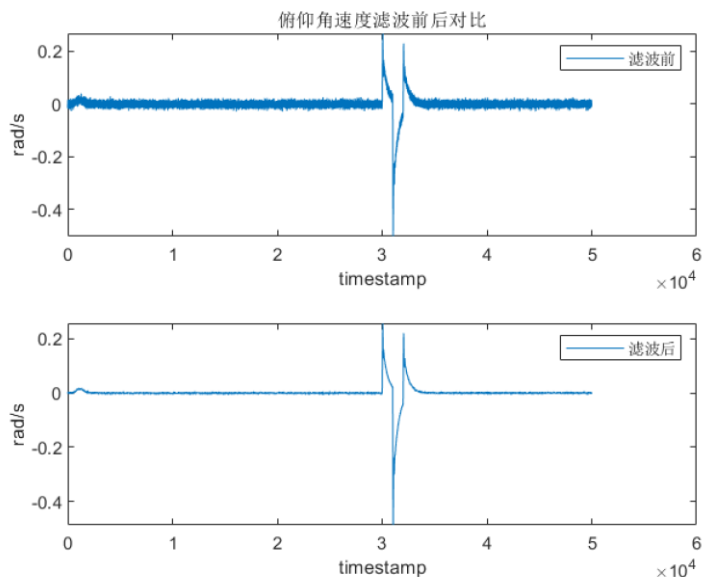
1. 在“SmallFixedWingUAVupper.slx”的“Command”路径下中，可以使用“Signal Builder (PitchChannel)”模块向俯仰通道注入激励。对于俯仰通道辨识，可以使用不同幅值和持续时间的偶极方波信号作为激励，图中的激励信号为幅值 20° 持续时间 4s 的偶极方波信号。为了防止其它通道对俯仰辨识效果产生影响，需要保持其它通道不变，仅给俯仰通道注入激励信号。



(2) 运行软件仿真并记录所需的各状态量, 并进行滤波处理

1. 打开“model”文件夹, 首先运行初始化文件“[InitDatactrl.m](#)”, 然后运行“[SmallFixedWingUAVupper.slx](#)”程序。该程序运行完成后可以自动导出俯仰力矩通道辨识所需的数据, 并保存为原始数据文件“Cm_Data.mat”。文件“Cm_Data.mat”中各行数据依次为: 迎角 α (“alpha”)、升降舵舵偏角 δ_e (“dE”)、俯仰角速率 ω_{y_b} (“wby”)、空速 V_a (“Airspeed”)、俯仰角加速度 $\dot{\omega}_{y_b}$ (“wby_dot”)。

2. 由于采集到的各个状态信号包含噪声, 因此在进行气动参数辨识前需要对采集到的数据进行滤波。在本实验中, 选择 MATLAB 自带的滑动平均滤波器进行处理, 该滤波器具有“通低频阻高频”的效果, 以此消除高频噪声影响, 可以通过使用 MATLAB 信号分析器观察各信号频谱特征选择滤波器参数, 为了保证滤波器造成的各信号时延相同, 综合考虑滤波效果, 此处选择平均滑动滤波窗口长度为 15, 效果如图(以俯仰角速度为例), 其中 ω_{y_b} 为原始数据, $\omega_{y_b,f}$ 为滤波后数据可见, 滤波效果明显。

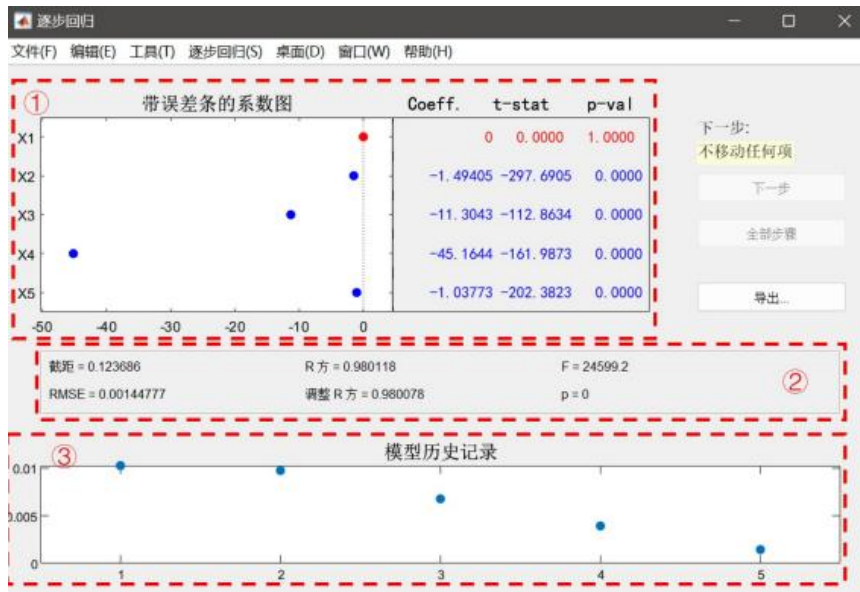


(3) 使用最小二乘法和逐步回归法辨识气动参数

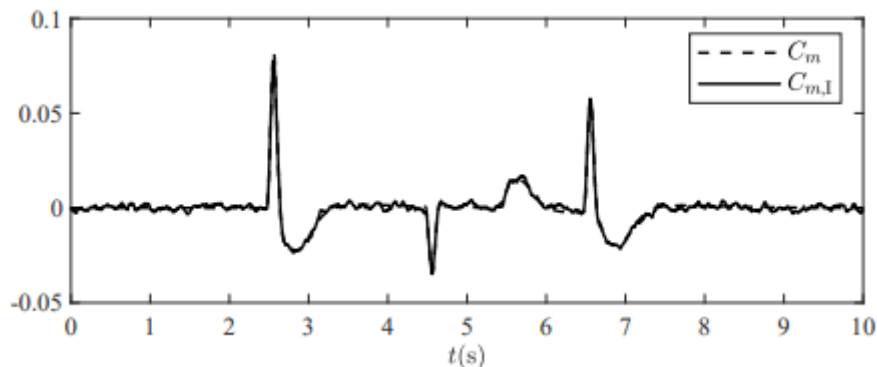
辨识结果: 以幅值 20 度持续时间 4 秒的偶极方波信号辨识结果为例进行说明。

图为“stepwise”函数的界面, 主要包括三个部分, ①中左侧的图为回归系数带有误差条的辨识结果, 纵坐标的“X1”至“X4”为参与逐步回归的自变量, 横坐标为相应的误差条数值; ①中右侧的部分与左侧的各行相对应, 共有 3 列, 其中只需要关心为回归系数“Coeff.”列, 例如“-1.49405”、“-11.3043”、“-45.1644”和“-1.03773”即为本例辨识出的气动参数。

其余列为其它的统计学量。②为最终的回归结果, 包括“截距”; 拟合优度 R^2 , R^2 越接近 1 说明结果越好; “F 统计量”, 该指标越大说明拟合的结果越好; “RMSE”指均方根误差, 该指标越小越好; “调整 R 方”此处并没有被用到, 不需要关注; “p”表示统计学上的 p 值, 此处为接受拟合方程后出错的概率大小。③为依次引入自变量后, “RMSE”的变化情况。



本例中, “X2”至“X5”为实验中所求的气动参数 $C_{m\alpha}$, $C_{m\dot{\alpha}}$, C_{mq} , $C_{m\delta_e}$, 而截距则对应 C_{m0} , 拟合优度 R^2 为 0.98, 说明回归结果比较精确。在本章的实验中, 只需要关注辨识出的截距、回归系数以及拟合优度 R^2 就足够了。下图为气动参数辨识结果的拟合效果, 其中 C_m 曲线为俯仰力矩系数真值, $C_{m,l}$ 曲线为根据辨识出的气动参数拟合出的俯仰力矩系数。二者对比, 说明辨识出的气动参数较为准确。



(4) 验证辨识结果，并比较不同输入信号的辨识精确度。

1. 辨识结果：更换不同幅值和持续时间的偶极方波，完成 6 组对比实验，得到表中的数据。

幅值	20°			5°		
长度	4s	6s	8s	4s	6s	8s
R^2	0.980118	0.98354	0.97199	0.956337	0.958191	0.930923

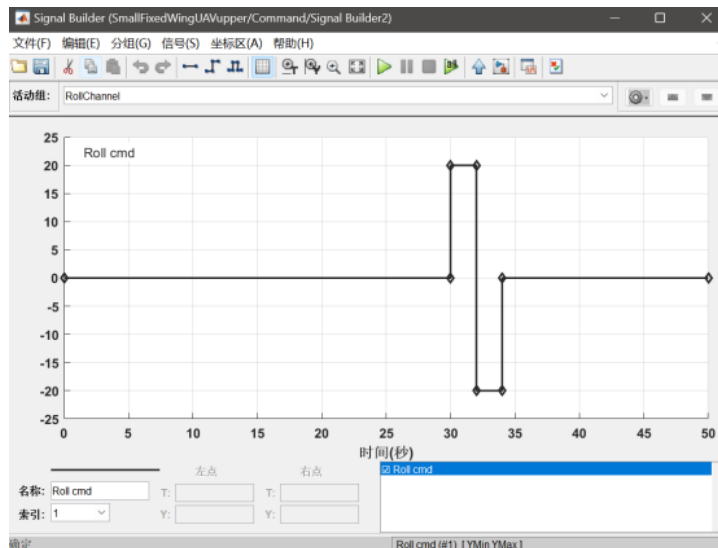
2. 比较不同输入信号的辨识精确度：通过对比可以看出采用偶极方波输入的长度为 6s 时的 R^2 比 4s 和 8s 时的 R^2 更大；随着偶极方波幅值的增大， R^2 更接近于 1，有较好的拟合效果。

3. 分析：首先分析输入信号的幅值对辨识结果的影响，在保证无人机稳定的前提下，输入信号的幅值越大，意味着无人机的作动幅度越大，一方面有利于更充分地激发无人机相应通道的运动模式，有利于提高辨识精度；另一方面产生的姿态角、姿态角速度等数据的变动更大，有利于将其从噪声数据中清洗出来，从而提高辨识的精度。其次分析输入偶极方波信号的长度对辨识结果的影响，偶极方波信号的长度会影响其在频谱上的能量分布，合适的频率能量分布可以更好地激发无人机相应频段的响应，从而提高辨识的精度。

5.3. 设计实验

(1) 设计激励信号

对于激励信号设计，保证无人机以待辨识通道运动为主，以此来提高辨识精确度。在“SmallFixedWingUAVupper.slx”的“Command”路径下中，可以使用“Signal Builder (RollChannel)”模块向滚转通道注入激励。为了防止其它通道对滚转通道辨识效果产生影响，需要保持其它通道不变，仅给滚转通道注入激励信号，注入的激励信号如图所示。



(2) 运行 Simulink 软件仿真，完成辨识所需的作动并记录相应数据

打开“model”文件夹，首先运行初始化文件“[InitDatactrl.m](#)”，然后运行“[SmallFixedWingUAVupper.slx](#)”程序。该程序运行完成后可以自动导出滚转力矩通道辨识所需的数据，并保存为原始数据文件“CI_Data.mat”。文件“CI_Data.mat”中各行数据依次为：侧滑角 β (“b

eta”)、副翼舵偏角 δ_a (“dA”)、方向舵舵偏角 δ_r (“dR”)、空速 Va (“Airspeed”)、滚转角速率 $\dot{\omega}_{x_b}$ (“wbx”)、偏航角速率 $\dot{\omega}_{z_b}$ (“wbz”) 以及滚转角加速度 $\dot{\omega}_{x_b}$ (“wby_dot”)。

(3) 对各个状态量数据进行滤波等预处理

运行“Cl_DataPreprocess.mlx”文件。

(4) 使用最小二乘法和逐步回归法辨识气动参数

选取飞机进行平稳飞行阶段后，对应的各项数据进行辨识，基于固定翼姿态动力学方程，计算出实际滚转力矩系数力矩系数，通过 MATLAB 自带的“regress”和“stepwise”函数进行最小二乘法估计和逐步回归，得到滚转通道气动参数。

滚转通道代码位于“Cl_Identification.mlx”，如下面的代码块所示。第 1 行为清空工作区，第 2 行为从“.mat”文件中读取数据并存储为“data”，第 4-10 行为将“data”拆分为动压、侧滑角、滚转角加速度、副翼舵偏角（修正前）、方向舵舵偏角（修正前）、滚转角速度、偏航角速度，第 12-16 行输入了该通道辨识所需的飞机转动惯量以及几何参数等，第 17-20 行为根据刚体转动方程计算滚转力矩系数，第 21-28 行为计算空速、计算滚转角速度和偏航角速度相关的状态量，第 30 行生成零升力矩项，第 31 行将用于辨识的状态量数据合并成矩阵，第 32 行调用 MATLAB 提供的“regress”函数进行辨识。第 34 行为使用“stepwise”函数进行辨识。对于“regress”函数和“stepwise”函数已经在分析实验中进行了说明，不再赘述。第 35 行为使用辨识出的气动参数计算滚转力矩系数。

```
clear;
load('Cl_DataFiltered.mat'); %load data
% split data
wbx=data_filtered.wbx(5500:7500)';
wbx_dot=data_filtered.wbx_dot(5500:7500)';
Va=data_filtered.Airspeed(5500:7500)';
dA=data_filtered.dA(5500:7500)';
dR=data_filtered.dR(5500:7500)';
beta=data_filtered.beta(5500:7500)';
wbz=data_filtered.wbz(5500:7500)';
% Aircraft geometric parameters
Ix=0.0894;
c=0.3;
b=1.2;
S=0.32;
rho = 1.225;
qbar=zeros(2001,1); %calculate qbar Cl wbx_compensation wbz_compensation
Cl=zeros(2001,1);
dalpha_after=zeros(2001,1);
q_after=zeros(2001,1);
wbx_compensation=zeros(2001,1);
wbz_compensation=zeros(2001,1);
for i=1:2001
    qbar(i)=0.5*rho*Va(i)^2;
    Cl(i)=(Ix*wbx_dot(i))/(qbar(i)*b*S);
    wbx_compensation(i)=0.5*wbx(i)*b/(Va(i));
    wbz_compensation(i)=0.5*wbz(i)*b/(Va(i));
```

```

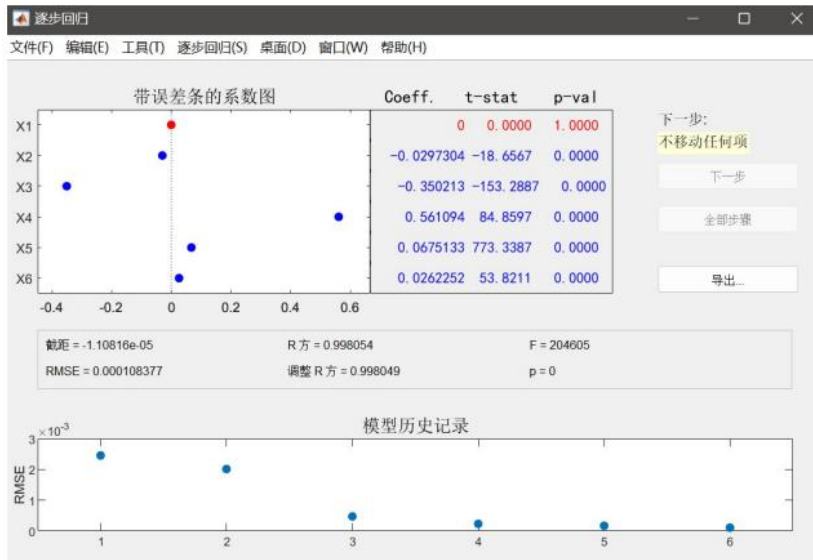
end
% regress
Cl0=ones(2001,1);
XCl=[Cl0,beta,wbx_compensation,wbz_compensation,dA,dR];
[bCl,bintCl,rCl,rintCl,statsCl]=regress(Cl,XCl);
% stepwise
stepwise(XCl,Cl);
Cl_identify=XCl*bCl;

```

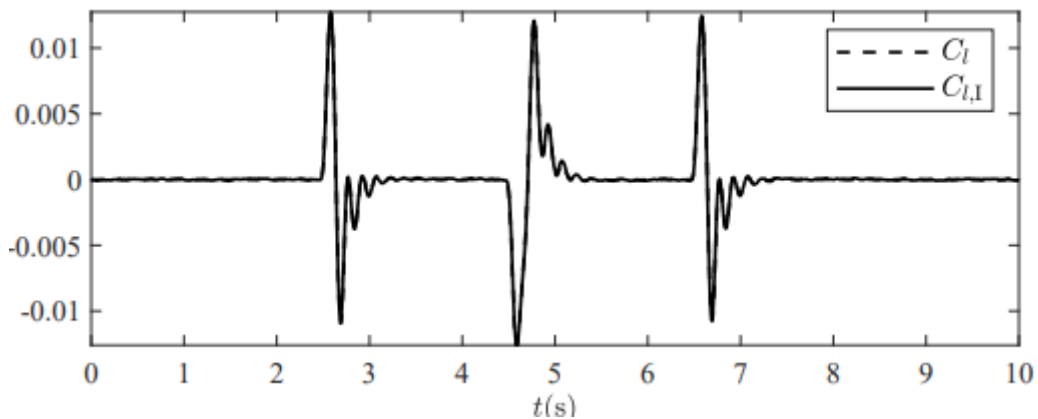
(5) 验证辨识结果

滚转力矩通道辨识结果如图： R^2 达到 0.998，回归效果较好。如表所示，辨识出的气动参数与真值进行对比，辨识结果与真值基本吻合。

气动参数	C_{l_0}	C_{l_β}	C_{l_p}	C_{l_r}	$C_{l_{\delta_a}}$	$C_{l_{\delta_r}}$
真值	0	-0.04	-0.414	0.399	0.0677	0.0168
辨识结果	0.00001	-0.0297304	-0.350213	0.561094	0.0675133	0.0262252



如图, 对比实际的滚转力矩导数和辨识结果拟合出的滚转力矩导数, 二者基本完全重合, 证明辨识精度较高。



5.4. 硬件在环仿真实验

(1) 设计固定翼无人机飞行轨迹

以高度通道硬件在环实验为例进行说明。使用“航点+空速”的指令接口对固定翼无人机进行控制，可以设定路径点参考速度为 40m/s，固定翼在横侧向上不作动，保持直线前飞，而在高度通道进行偶极方波式的运动。

(2) 运行硬件在环仿真，完成辨识所需动作

如图，在电脑上接入 Cubepilot/CubePilot/Pixhawk 自驾仪，打开实验包内的“HeightChannel”文件夹，运行“[AircraftMathworksMavlinkHITLRun.bat](#)”文件，弹出终端窗口，在终端窗口输入合适的串口号，按下回车键，一键启动仿真平台。



切换到 QGC 控制台界面，如图当左上角无人机处于“Ready to fly”状态后，运行提前设计好的轨迹控制程序“HeightChannel_OffboardController.py”。



随后，双击运行 [e3-4\GuidanceModel_Identification\HeightChannel\Python38Run.bat](#) 脚本在其中输入：`python HeightChannel_OffboardController.py`。开始运行航点控制飞行程序。

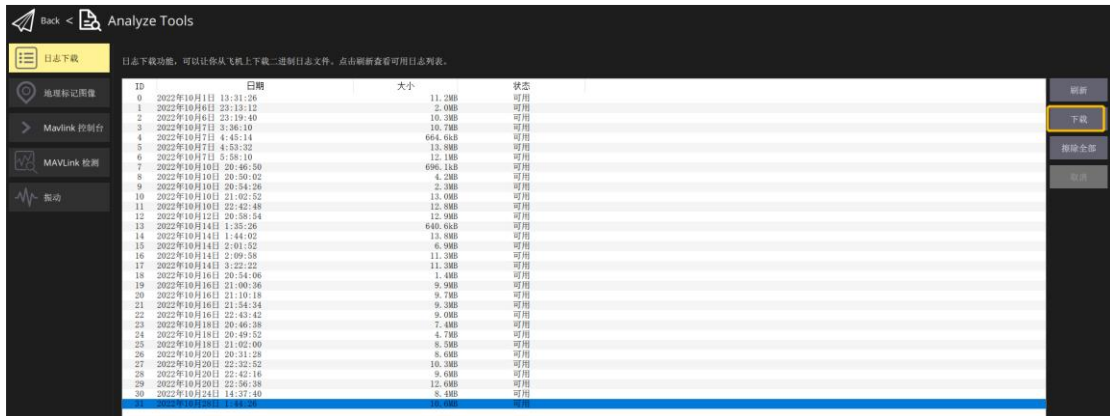


通过 CopterSim 观测到无人机部分实时状态，如图。当无人机飞行到最后一个航迹点位置时，将进行原地盘旋运动。此时，运行 QGC 平台的返航程序即可实现无人机返航，如图，结束飞行运动。

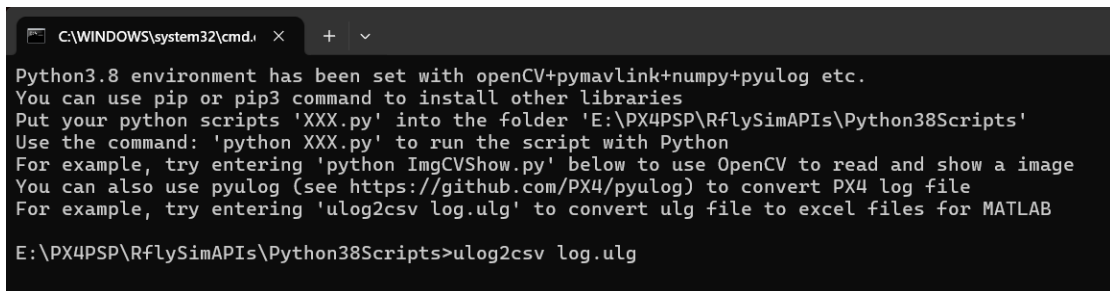


(3) 从自驾仪中导出飞行日志，进行数据预处理

运行结束后，当无人机稳定降落后，单击 QGC 平台左上角图标，选择“Analyse Tools”，在“Log Download”界面可选择最近一次飞行的数据进行下载，如图。



下载得到的数据为“ulg”格式，双击打开“e3-4\GuidanceModel Identification\HeightChannel\Python38Env.bat”的 python 环境，其中输入：`ulog2csv log.ulg`，其中 log.ulg 为你自己下载的日志名称。



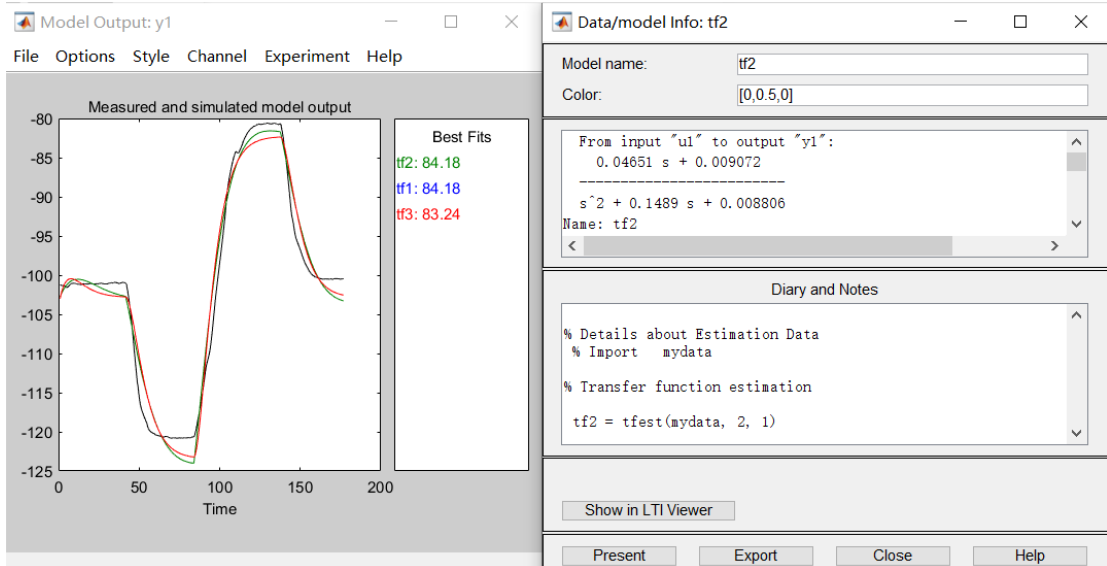
导出“csv”文件之后，接下来的数据预处理步骤与基础实验中基本相同。区别在于，观察高度输入输出数据，可以发现二者的采样频率相同，因此不需要进行采样频率配准，只需要进行起始终止时间对准，并截取相应的数据段即可。过程与基础实验相同，不再赘述。

(4) 使用 MATLAB 系统辨识工具箱完成制导模型辨识

步骤与基础实验相同，使用 MATLAB 系统辨识工具箱完成高度通道制导模型辨识。辨

识效果如图，匹配度达到了 84.18%，符合预期。高度通道的制导模型为：

$$G_H(s) = \frac{H(s)}{H_d(s)} = \frac{0.04651s + 0.009072}{s^2 + 0.1489s + 0.008806}$$



其余通道辨识的流程与高度通道类似，区别在于设计的航迹不同、选取的输入输出数据不同，可以参考实验包内提供的代码进行实践。

5.5. 气动参数辨识硬件在环仿真实验

(1) 设计固定翼无人机滚转通道激励

与制导模型辨识硬件在环实验不同，使用“姿态 + 油门”的指令接口对固定翼无人机进行控制，可以设定固定翼无人机油门固定为 0.8，在纵向上不作动；而在滚转通道进行偶极方波式的运动，幅值为 30 度，持续时间为 20 秒。

相应代码位于“[e3-4\AerodynamicParameter_Identification\AircraftMathworksAttCtrl.py](#)”，关键代码如代码段所示。第 1-2 行为设定固定翼无人机的滚转角为 30 度，油门为 0.8，并持续 10s；第 3-4 行为设定固定翼无人机的滚转角为 -30 度，油门为 0.8，并持续 10s；第 5 行为打印标志行，说明本周期运动结束。

```
mav1.SendAttPX4([30,0,0],0.8)
time.sleep(10)
mav1.SendAttPX4([-30,0,0],0.8)
time.sleep(10)
print("arrive")
```

(2) 运行硬件在环仿真，完成辨识所需动作

与制导模型辨识硬件在环实验相同，运行硬件在环仿真，可以得到相应的飞行日志文件。

(3) 从 Pixhawk 飞控板中导出飞行日志，进行数据预处理

与制导模型辨识硬件在环实验相同，从 QGC 下载最新的飞行日志，双击打开“[e3-4\Gu](#)

[idanceModel_Identification\HeightChannel\Python38Env.bat](#) "的 python 环境，其中输入：ulog2csv log.ulg，其中 log.ulg 为你自己下载的日志名称。

从“vehicle_local_position_0.csv”中可以获取地面坐标系下的无人机的位置、速度和加速度；从“vehicle_attitude_0.csv”中可以获取四元数表示的无人机姿态；从“vehicle_angular_velocity_0.csv”中可以获取无人机的角速度；从“airspeed_0.csv”中可以获取无人机的空速；从“vehicle_air_data_0.csv”中可以获取无人机的动压和空气密度；从“actuator_controls_0_0.csv”中可以获取无人机作动器的控制信号。

之后需要对以上数据进行时间配准处理，包括起始终止时间配准和采样频率配准。起始终止时间配准是指绘制姿态等原始数据曲线，找到符合参数辨识要求的数据段，从而确定截取其他所有数据的起始终止时间。除此之外，由于各个传感器的采样频率不同，导出的数据采样频率也不同，因此需要使用 MATLAB 内置的“resample”函数进行采样频率配准。

时间配准处理后，需要根据已知的状态数据（地面坐标系下的速度、姿态），推算出参数辨识需要的攻角和侧滑角数据，推算方法与分析实验的实验设计中的描述一致。具体方

$$\phi = \arctan \frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}$$

法为：从四元数可以按照式 $\theta = \arcsin(2(q_0q_2 - q_1q_3))$ 方便地解算出无人机姿态角 (ϕ, θ, ψ) ，

$$\psi = \arctan \frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}$$

结合已知的地面坐标系下的速度 V_g ，按照式

$${}^b\mathbf{v}_g = \begin{bmatrix} v_{gb} \\ v_{gb} \\ v_{gb} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\theta\cos\psi\sin\phi - \sin\psi\cos\phi & \sin\theta\sin\psi\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \sin\theta\cos\psi\cos\phi + \sin\psi\sin\phi & \sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix} \mathbf{v}_g$$

可以得到机体坐标系下的速度，在硬件在环实验中，可以忽略风的影响，由式

$$V_a = \sqrt{v_{ax_b}^2 + v_{ay_b}^2 + v_{az_b}^2}$$

$$\alpha = \arctan \left(\frac{v_{az_b}}{v_{ax_b}} \right)$$

可估计攻角和侧滑角，相应代码在“Cm_DataPreprocess.

$$\beta = \arcsin \left(\frac{v_{ay_b}}{\sqrt{v_{ax_b}^2 + v_{ay_b}^2 + v_{az_b}^2}} \right)$$

mlx”中，关键代码如代码块所示。

```
alpha(i,1)=atan(Vb(i,3)/Vb(i,1));
beta(i,1)=atan(Vb(i,2)/sqrt(Vb(i,3)^2+Vb(i,1)^2));
```

(4) 对各个状态量进行滤波处理

同分析与设计实验，利用 MATLAB 自带的“movmean”函数，选择合适的窗口长度即

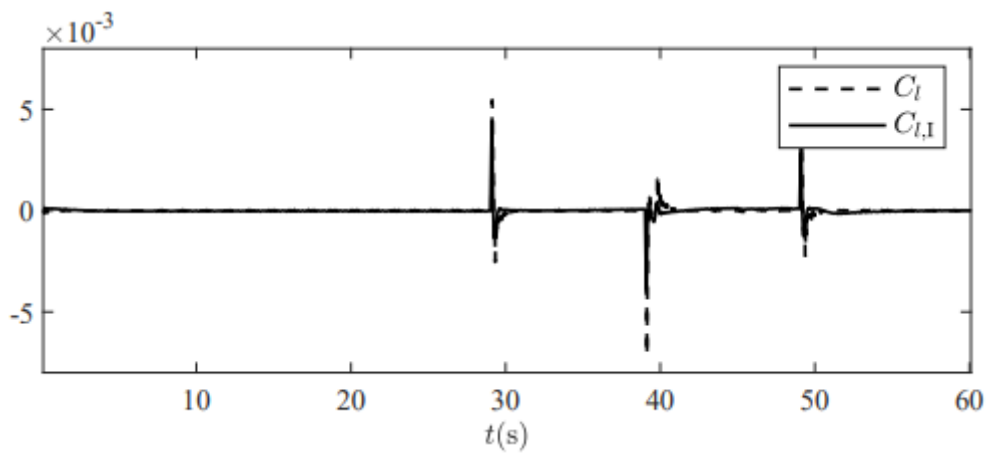
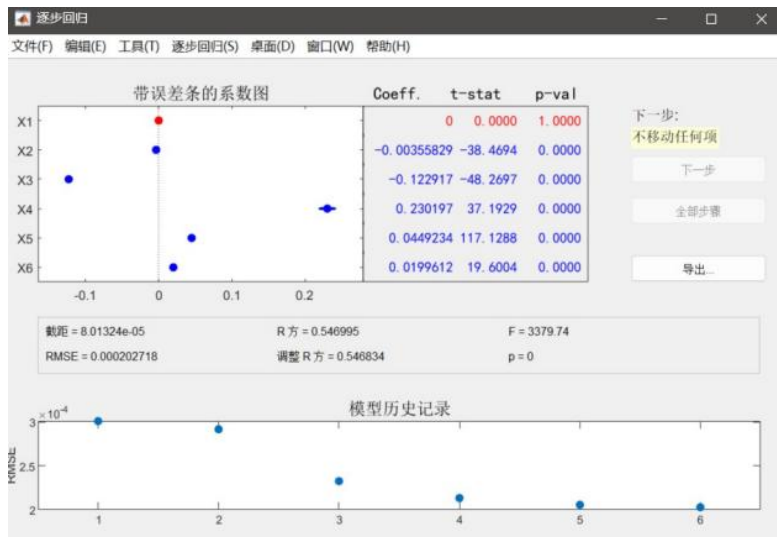
可实现高频噪声滤除（硬件在环实验中注入的噪声较小，因此可以选择较小的窗口长度）。具体只需要运行“Cl_DataPreprocess.mlx”即可得到滤波后的数据“Cl_DataFiltered.mat”。

（5）使用最小二乘法和逐步回归法辨识气动参数并验证结果

与分析、设计实验类似，俯仰通道辨识代码如下表（“Cm_Identification.mlx”）。第 1 行为清空工作区，第 2 行为从“.mat”文件中读取数据，第 3-5 行为设置截取数据起始点、结束点和数量，第 7-12 行为将“data1”拆分为舵偏角（修正前）、俯仰角速度、攻角、空速、动压、俯仰角加速度数据，第 14-18 行输入了该通道辨识所需的飞机转动惯量以及几何参数等，第 19-24 行为根据刚体转动方程计算俯仰力矩系数并对升降舵舵偏进行修正，第 26 行生成零升力矩项，第 27 行将用于辨识的状态量数据合并成矩阵，第 28 行调用 MATLAB 提供的“regress”函数进行辨识。第 30 行为使用“stepwise”函数进行辨识。第 31 行为使用辨识出的气动参数计算俯仰力矩系数。

```
clear;
load('Cl_DataFiltered.mat');% load dataa
startpoint = 7000;% set intercept point
endpoint = 21000;
% split data
wbx=data_filtered.wbx(startpoint:endpoint)';
wbx_dot=data_filtered.wbx_dot(startpoint:endpoint)';
Va=data_filtered.Airspeed(startpoint:endpoint)';
dA=data_filtered.dA(startpoint:endpoint)';
dR=data_filtered.dR(startpoint:endpoint)';
beta=data_filtered.beta(startpoint:endpoint)';
wbz=data_filtered.wbz(startpoint:endpoint)';
% Aircraft geometric parameters
Ix=0.0894;
c=0.3;
b=1.2;
S=0.32;
rho = 1.15;
qbar=zeros(length(Va),1);
Cl=zeros(length(Va),1);
wbx_compensation=zeros(length(Va),1);
wbz_compensation=zeros(length(Va),1);
for i=1:length(Va)
    qbar(i)=0.5*rho*Va(i)^2;
    Cl(i)=(Ix*wbx_dot(i))/(qbar(i)*b*S);
    wbx_compensation(i)=0.5*wbx(i)*b/(Va(i));
    wbz_compensation(i)=0.5*wbz(i)*b/(Va(i));
end
% regress
Cl0=ones(length(Va),1);
XCl=[Cl0,beta',wbx_compensation,wbz_compensation,dA',dR'];
[bCl,bintCl,rCl,rintCl,statsCl]=regress(Cl,XCl);
% stepwise
stepwise(XCl,Cl);
Cl_identify=XCl*bCl;
```

辨识效果如图， R^2 达到了 0.55，符合预期，拟合效果如图。



6. 参考资料

[1] 微小型固定翼无人机飞行控制设计与实践

7. 常见问题

Q1: 在进行分析实验时，运行 [DrawVel.m](#) 所得到的图不一致或是运行失败？

A1: 请观察计算公式中的变量是否进行更改，或是在绘图代码部分中的变量是否进行更改。