

# 1. 实验名称及目的

## 1.1. 实验名称

静拉力整合模型辨识实验

## 1.2. 实验目的

动力整合模型为辨识电机油门/PWM 输入至螺旋桨拉力与反扭力矩输出的量化关系，可分为静态模型（用静态函数描述）与动态模型（用传递函数描述）。

## 1.3. 关键知识点

曲线拟合与系统辨识的区分

# 2. 实验效果

# 3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\4.RflySimModel\2.AdvExps\e2\\_MultiModelCtrl\1.MultiModelCtrl\](#)

文件夹/文件名称	说明
细分 PWM 数据.xlsx	测试台实验数据
Static_data.mat	由阶梯测油门/PWM 测试出来的单个电机+螺旋桨动力数据
FX450_integrated_power_model.mlx	实时脚本，包含数据处理和模型辨识的逻辑
No2_0.2to1.mat	扫频测试数据，频段为 0.2~xHz，由于动力测试台限制，我们测试的 PWM 输入信噪比较高，测量中高频信号的意义被认为不大，因为只有 30s 扫频时间

# 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 2022A 及以上 <sup>③</sup>		

①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

### 5.1. 必做实验：导入测试台实验数据

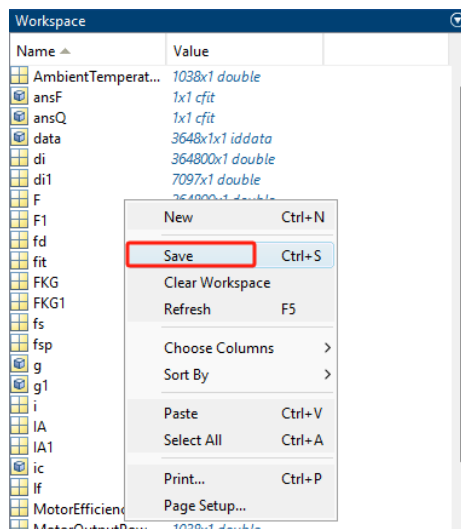
#### Step 1: 导入数据

如图 1 所示，打开 MATLAB 并将 MATLAB 当前工作路径指定到数据存放位置。用 MATLAB 打开动力测试台阶梯测试的实验数据“细分 PWM 数据.xlsx”，选中所需导入的数据的列（注意最好不导入第一行变量名称，默认输出“NaN”），选择“输出类型”为“列向量”并导入（之后的数据导入流程类似）；

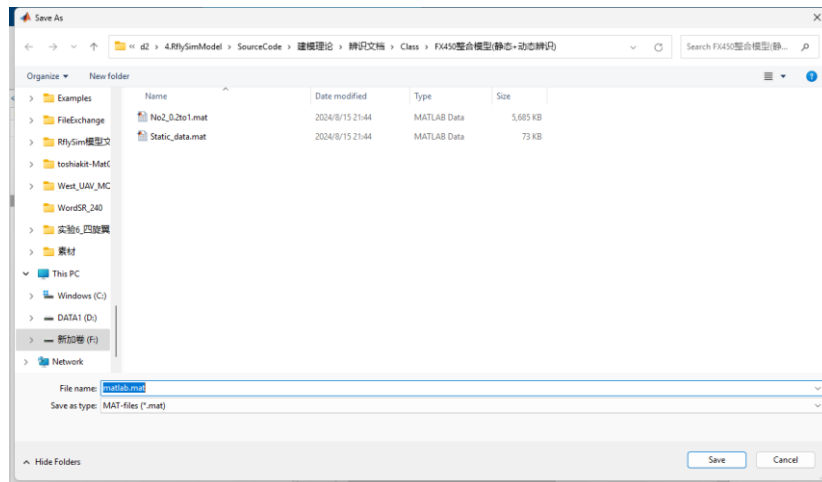


图 1 数据导入

#### Step 2: 保存数据



在工作空间右击，选择保存，在弹出的对话框选择保存



## 5.2. 必做实验：静态模型辨识

### Step 1: 加载数据

加载“Static\_data.mat”

```
1 clear
2 clc
3 load('Static_data.mat'); % 数据导入
```

静态模型的辨识无数据采样频率要求，无须对数据的进行滤波处理，同时认为测量数据无偏差或偏差极小，所以无须进行去趋势处理；

### Step 2: 统一单位并截取测试数据段

对所需数据单位预处理，并且取上升阶梯段数据进行拟合，MATLAB 代码为：

```
4 PWM=PWMus; % 电机 PWM 输入值(us)
5 w=NRPM; % 电机/螺旋桨转速值(rpm)
6 T=FKG*9.8; % 把拉力单位从 kg 转换至 N
7 Q=TNM; % 反扭力矩(N.m)
8 % 取上升阶梯段数据进行拟合
9 PWMtest=PWM(1:1138);
10 Ttest=T(1:1138);
11 Qtest=Q(1:1138);
```

### Step 3: 模型拟合

采用 `lsqcurvefit` 函数对电机 PWM 输入至螺旋桨拉力输出进行拟合（这里考虑到 PWM 值为 1000 时拉力输出应该为 0，因此待拟合函数形式为为“ $c_1x^3 + c_2x^2 + c_3x - 1000^3c_1 - 1000^2c_2 - 1000c_3$ ”，电机 PWM 输入至螺旋桨扭矩输出代码同理，只需将“Test”更换为“Qest”），代码为：

```
12 x=PWMtest; % 横坐标变量（输入）
13 y=Ttest; % 纵坐标变量（输出）
```

```

14 f = @(c, x) c(1).*x.^3 + c(2).*x.^2 + c(3).*x -(c(1)*1000^3 + c(2)*1000^2 + c(3)*1000); % 待拟合曲线方程
15 c0= [0 0 0]; % 变量初值
16 [ct, fval]= lsqcurvefit(f, c0, x, y); %进行拟合

```

#### Step 4: 查看拟合效果及参数

代码为:

```

17 xx=1000:1:2000; % 给定横坐标范围
18 yy=f(ct, xx); % 对应给定横坐标范围的拟合函数值
19 plot(x, y, 'r.', xx, yy, 'b-'); %画出拟合效果
20 disp(ct); % 显示拟合的参数值

```

#### Step 5: 静态模型辨识结果

运行 FX450\_integrated\_power\_model.mlx 脚本

➤ PWM→拉力模型;

经过模型拟合, 可以得到拉力的静态整合模型为:

$$T_s = -0.000000040824166\rho^3 + 0.000184086137229\rho^2 - 0.248603166860739\rho + 105.3411956994972$$

模型与数据的拟合效果如图 2 所示。

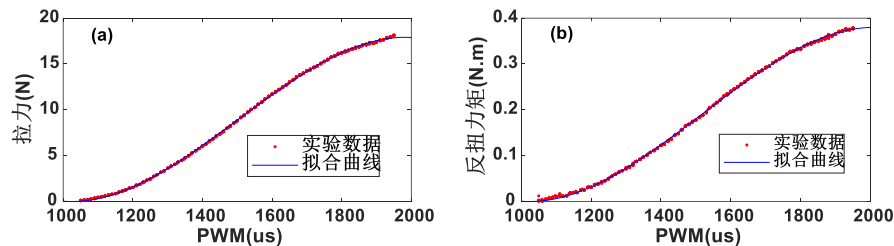


图 2 PWM-拉力模型与 PWM-扭矩模型拟合

➤ PWM→扭矩模型。

同理, 经过模型拟合, 可以得到拉力的静态整合模型为:

$$Q_s = -0.000000000795457\rho^3 + 0.000003619447532\rho^2 - 0.004910485951841\rho + 2.086495261312058$$

其中,  $\rho$  表示输入的电机 PWM,  $T_s$  和  $Q_s$  分别为静态拉力与静态扭矩。模型与数据的拟合效果如图 2 所示。

注: 上述模型也可以根据电机的力效表数据得到, 前提是电调螺旋桨等设备需一致。不同产商电机力效表信息不一致, 但也有信息不全和较为粗略等情况。

### 5.3. 必做实验: 动态模型辨识

#### Step 1: 加载数据

加载 “No2\_0.2to1.mat”, 此处导入扫频范围为 0.2~1Hz 的时域数据;

## Step 2: 统一单位

统一单位代码与静态模型一致（见静态模型中的②）；

## Step 3: 求解静态模型输出

根据静态模型求解扫频测试 PWM 输入对应的静态拉力与静态扭矩输出，即：

```
21 Ts = ct(1)*PWM.^3 + ct(2)*PWM.^2 + ct(3)*PWM -(ct(1)*1000^3 + ct(2)*1000^2 +  
ct(3)*1000); % 拉力模型  
22 Qs = cq(1)*PWM.^3 + cq(2)*PWM.^2 + cq(3)*PWM -(cq(1)*1000^3 + cq(2)*1000^2 +  
cq(3)*1000); % 扭矩模型
```

## Step 4: 数据滤波处理

由于动态模型的数据采样频率为 10000Hz，为了避免出现频率混叠及适应辨识的频率范围，首先对采集数据进行低通滤波处理（此处以截止频率为 5Hz 为例）：

```
1 % 低通滤波  
2 fs=1e4; % 采样频率  
3 lf=5; % 滤波截止频率  
4 PWMf=lowpass(PWM,lf,fs); % 滤波 PWM  
5 Tf=lowpass(T,lf,fs); % 滤波动态拉力  
6 Tsf=lowpass(Ts,lf,fs); % 滤波静态拉力  
7 Qf=lowpass(Q,lf,fs); % 滤波动态扭矩  
8 Qsf=lowpass(Qs,lf,fs); % 滤波静态扭矩
```

## Step 5: 数据降采样处理

为提升辨识速度，减小辨识数据长度，将滤波后的所需数据进行降采样处理（这里取降采样频率为最大考察频率范围的 20 倍）。处理代码为：

```
1 % 降采样  
2 fd=20*lf; % 降采样频率  
3 PWMfd=resample(PWMf,fd,fs);  
4 Tfd=resample(Tf,fd,fs);  
5 Tsf=resample(Tsf,fd,fs);  
6 Qfd=resample(Qf,fd,fs);  
7 Qsf=resample(Qsf,fd,fs);
```

**注 1：**降采样处理可能会导致数据前端和末端出现混乱情况，为避免此情况，辨识中可截取中间段合适数据进行辨识，对于等时采样数据也可以采用等距降采样实现。

**注 2：**实验中 PWM 选择给定输入了 20Hz 以内的信噪比较高的信号，原因一方面是无法判断小范围波动的 PWM 值具体范围，另一方面是由于扫频时间只有 30s，若给定较高频率信号，扫频辨识的结果也会比较粗糙。

## Step 6: 动力模型的频域辨识

- 估计数据在所需辨识频域带宽内的响应函数。如辨识动态 PWM→动态拉力的处理代码为：

```
1 data=iddata(Tfd,PWMfd,1/fd); % 定义数据
2 fsp = linspace(0,lf*2*pi,500); % 线性化频率范围，也可以采用 logspace 函数
3 g = spafdr(data,[],fsp); % 估计对应频率带宽的频率响应
```

- 辨识对应带宽内的频域降阶模型。代码为：

```
1 % 逐阶辨识至 10 阶传函，计算每阶传函的拟合优度传至 fit
2 for i=0:10
3 sys=tfest(g,i);
4 [y,fit(i+1),ic]=compare(sys,data);
5 end
6 sys1=tfest(g,find(fit==max(fit))-1); % 取时域拟合优度最好的那组
7 compare(sys1,data); % 比较拟合优度最好的时域响应与测量数据进行比较
8 g1=idfrd(sys1); % 转换辨识模型为频率响应
9 bode(g,g1) % 比较辨识所得模型频率响应与高度近似的频率响应的伯德图
```

## Step 7: 动态模型辨识结果

运行 FX450\_integrated\_power\_model.mlx 脚本

- PWM→拉力模型；

经频域辨识可得 PWM→拉力的最佳降阶模型（由于 1Hz 以上辨识得到的模型的拟合优度都不算理想，这里采用的是 1Hz 数据拟合出来的 10 阶及以下最佳模型）为：

$$\frac{T_d(s)}{PWM(s)} = \frac{0.6532s^9 + 43.16s^8 + 67.03s^7 + 1166s^6 + 1135s^5 + 1.022e04s^4 + 6781s^3 + 3.521e04s^2 + 1.672e04s + 4.256e04}{s^9 + 35.64s^8 + 69.22s^7 + 973s^6 + 1072s^5 + 8722s^4 + 6102s^3 + 3.131e04s^2 + 1.448e04s + 3.961e04}$$

模型响应与数据的拟合效果如图 3 所示，图 3 (a) 显示为时序的拟合优度，图 3 (b) 为频域特性拟合效果。

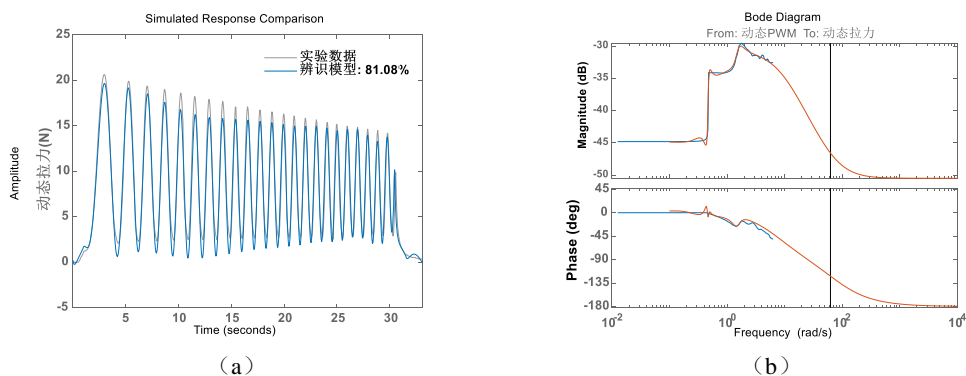


图 3 PWM-拉力模型拟合效果

注：实际当中，此模型在 1Hz 的响应阶型就快达 10 阶，是不适用于各类应用开发中的，且其高阶特性极有可能是由于电机或螺旋非线性特性的影响造成的。

➤ PWM → 反扭力矩模型。

由于在较高频辨识得到的频域模型难以于实际数据拟合上，采用经频域辨识可得 PWM → 反扭力矩的最佳降阶模型（1Hz 以内）为：

$$\frac{Q_d(s)}{PWM(s)} = \exp(-1.5s) (-2.766e-05)。模型响应与数据的拟合效果如图 4 所示。$$

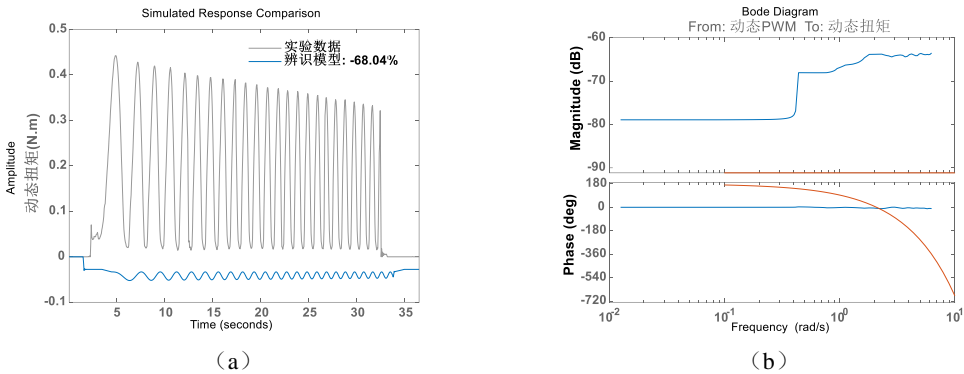


图 4 PWM-扭矩模型拟合效果

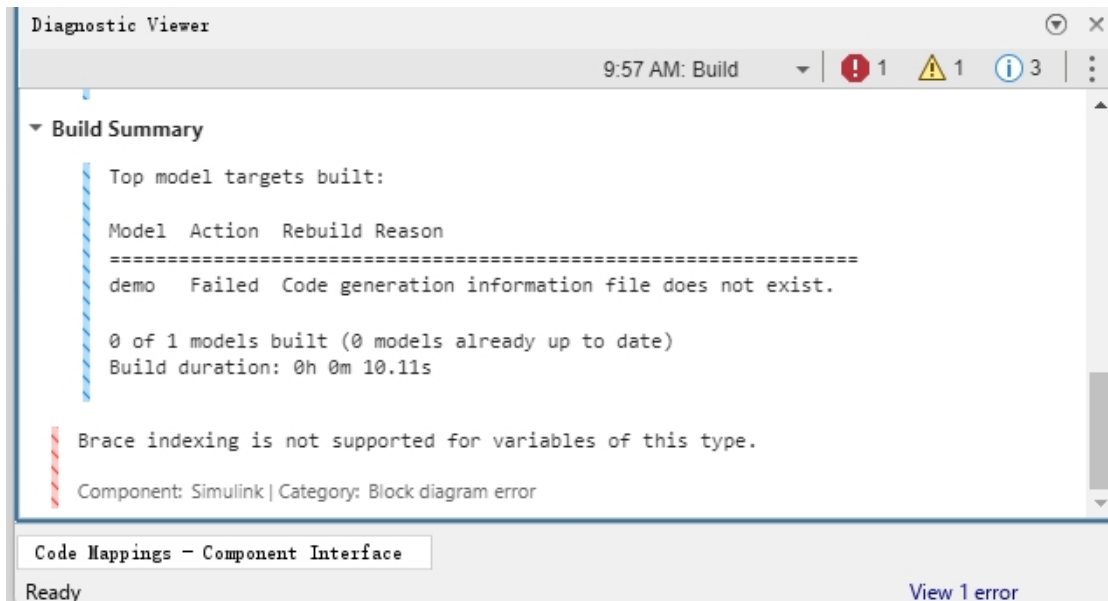
注：该模型阶数虽然不是很高，但是拟合优度较差，由频域特性的拟合效果可知，PWM 至反扭力矩的动态模型较为复杂，需要更高阶的模型来描述，但是这也违背了精简模型应用于控制开发的目的。

## 6. 参考资料

[1].

## 7. 常见问题

Q1: 未正确安装 visual studio c++编译环境并配置 mex，导致 Simulink 文件编译失败



A1: 首先将低于当前 MATLAB 版本的 Visual Studio C++编译环境安装到 VS 默认安装目录，然后在 MATLAB 的命令行窗口中输入指令“mex -setup”，一般来说会自动识别并安装上支持的编译器，命令行显示“MEX 配置使用 ‘Microsoft Visual C++ 2017 (C)’ 以进行编译”的字样说明安装正确。详细环境配置参考” [RflySim 平台安装目



录]RflySimAPIs\4.RflySimModel\API.pdf “中的环境配置

Q2: 编译报错，无法加载库文件



A2: 这可能是由于安装平台时 PX4PSP 工具箱未更新到最新版，更新 RflySim 安装包后按照如下配置重新安装平台即可

