

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## DOE ROV 的动力学建模与 运动控制



学生姓名: 简心语

学生学号: 515021910260

指导老师: 刘旌扬

专    业: 船舶与海洋工程

学    院: 船舶海洋与建筑工程学院

## Contents

<b>1</b>	<b>摘要</b>	<b>2</b>
<b>2</b>	<b>ROV 模型介绍</b>	<b>3</b>
2.1	DOE ROV 运动学模型 . . . . .	3
2.2	DOE ROV 动力学模型 . . . . .	3
2.3	推进器模型 . . . . .	4
2.3.1	螺旋桨模型 . . . . .	4
2.3.2	电机模型 . . . . .	4
<b>3</b>	<b>Simulink 模型</b>	<b>4</b>
<b>4</b>	<b>求解传递函数</b>	<b>4</b>
4.1	机理法 . . . . .	4
4.2	系统辨识法 . . . . .	6
<b>5</b>	<b>控制器设计</b>	<b>10</b>
5.1	深度控制器设计 . . . . .	10
5.2	艏向控制器设计 . . . . .	10
<b>6</b>	<b>非线性仿真验证</b>	<b>11</b>
<b>7</b>	<b>总结</b>	<b>13</b>
<b>8</b>	<b>附录</b>	<b>14</b>



## 1 摘要

根据给定的 ROV 数学模型和 Simulink 模型, 分别采用机理法和系统辨识法求取从垂向推进器控制电压到 ROV 深度变化的传递函数  $G_1(s)$  和从左右联动对推进器控制电压差 (控制电压之和为零以减小对 ROV 纵向位置的扰动) 到 ROV 艏向角变化的传递函数  $G_2(s)$ , 其中, 机理法由于非线性未能继续得到结果。根据得到的传递函数, 导入 Matlab PID Tuner 中进行 PID 参数整定。再将 PID 控制器加入到 Simulink 模型中形成闭环控制系统。最后对该系统进行深度与艏向运动自动控制仿真, 给定目标值, 观察响应曲线。

关键词: ROV 模型, Simulink, PID Controller, 系统辨识

## 2 ROV 模型介绍

### 2.1 DOE ROV 运动学模型

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ r \end{bmatrix} \quad (1)$$

### 2.2 DOE ROV 动力学模型

$$(M_{RB} + M_A)\dot{V} + [C_{RB}(v) + C_A(v)]V + D(v)V + g(\eta) = \tau \quad (2)$$

其中：

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & -my_G \\ 0 & m & 0 & mx_G \\ 0 & 0 & m & 0 \\ -my_G & mx_G & 0 & I_Z \end{bmatrix} \quad (3)$$

$$M_A = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 \\ 0 & 0 & 0 & I_z \end{bmatrix} \quad (4)$$

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & 0 & -m(x_G r + v) \\ 0 & 0 & 0 & -m(y_G r - u) \\ 0 & 0 & 0 & 0 \\ m(x_G r + v) & m(y_G r - u) & 0 & 0 \end{bmatrix} \quad (5)$$

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & Y_{\dot{v}}v \\ 0 & 0 & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & 0 \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & 0 \end{bmatrix} \quad (6)$$

$$D(v) = - \begin{bmatrix} X_u + X_{u|u}|u| & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v}|v| & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w}|w| & 0 \\ 0 & 0 & 0 & N_r + N_{r|r}|r| \end{bmatrix} \quad (7)$$

$$g(\eta) = [0 \ 0 \ 0 \ 0]^T \quad (8)$$

$$V = [u \ v \ w \ r]^T \quad (9)$$

$$\dot{V} = [\dot{u} \ \dot{v} \ \dot{w} \ \dot{r}]^T \quad (10)$$

$$\tau = [T_x \ 0 \ T_z \ M_r]^T \quad (11)$$

其中  $T_x$  为四个纵向推进器推力之和； $T_z$  为垂向推进器推力； $M_r$  为通过左右两对纵向推进器差动产生的水平面控制力矩。由于定深、定向、前进过程过程中不需要单独控制侧向运动，且侧向推进器推进时由于位置原因会影响艏向角，故在动力学建模时可省去侧向推进器，也不产生侧向控制力。

## 2.3 推进器模型

### 2.3.1 螺旋桨模型

$$T = \rho D^4 K_T n^2 \quad (12)$$

$$Q = \rho D^5 K_Q n^2 \quad (13)$$

$$K_T = \alpha_1 + \alpha_2 J_0 \quad (14)$$

$$K_Q = \alpha_3 + \alpha_4 J_0 \quad (15)$$

$$V_a = (1 - \omega)V \quad (16)$$

$$J_0 = \frac{V_a}{nD} \quad (17)$$

$$(18)$$

### 2.3.2 电机模型

$$e = V_s - u_f \quad (19)$$

$$u_f = k_e \omega \quad (20)$$

$$L \frac{di_a}{dt} + R_a i_a = e \quad (21)$$

$$J \frac{d\omega}{dt} = k_t i_a - B\omega - T_L \quad (22)$$

$$T_L = Q \quad (23)$$

## 3 Simulink 模型

如图 1 所示，原始模型由老师给出。对比文档“DOE ROV 四自由度模型方程及参数”后，对模型中一些错误参数做出了修正。

1. “Vertical Thruster” 中的电机电感改为  $L = 0.0468H$ ，螺旋桨转动惯量改为  $J = 9.275 \times 10^{-4} kg \cdot m^2$ 。
2. 将三个推进器的输入电压的范围改为  $[-100V, 100V]$ 。
3. 由角速度  $\omega$  到电压反馈值  $u_f$  的增益参数改为  $K_e = 42 * 60 / 1000 / (2 * \pi)$ 。

## 4 求解传递函数

### 4.1 机理法

机理法是根据系统组件的工作原理列出部件数学模型，再根据信号传递关系融合为控制系统的数学模型。针对从输入电压  $V_s$  到电机转轴角速度  $\omega$  的电机这一段工作过程，得到如图 2 所示的部件数学模型。使用这一方法的思路是，以“Vertical Thruster”为例，先根据机理法得到输入电压  $V_s$  到电机转速  $\omega$  的传递函数  $G_m(s)$ ，再根据螺旋桨模型得到电机转速  $\omega$  到螺旋桨推力  $T$  的传递函数  $G_p(s)$ ，然后直接使用 Matlab 线性化分析得到从螺旋桨推力  $T$  到垂向位

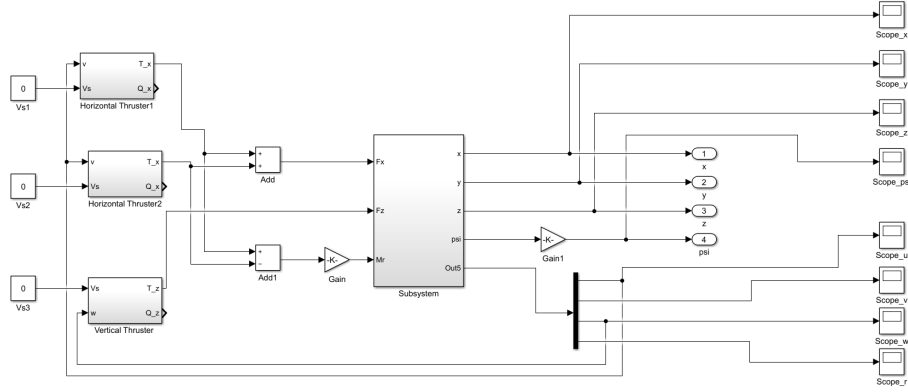


Figure 1: Simulink 原始模型

移  $z$  的传递函数  $G_r(s)$ ，最后将三者相乘得到从输入电压  $V_s$  到垂向位移  $z$  的传递函数  $G_2(s)$ 。这里要解释一下为什么不用 Matlab 线性化分析直接得到  $G_p(s)$ ，因为只要在零初始条件下，无论给什么阶跃信号，得到的结果都为 0 且无法改进，而对于  $G_r(s)$  却是适用的，遂放弃。由电机模型得到：

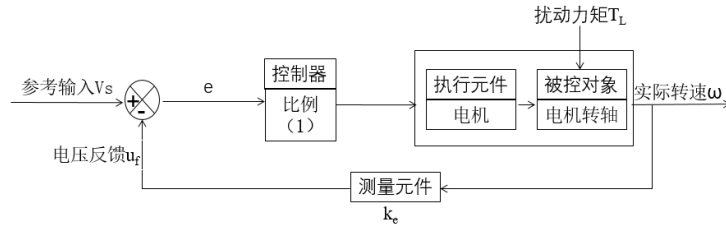


Figure 2: 电机部件数学模型

$$\dot{i}_a = -\frac{R_a}{L}i_a - \frac{K_e}{L}\omega + \frac{1}{L}i_a \quad (24)$$

$$\dot{\omega} = \frac{K_t}{J}i_a - \frac{B}{J}\omega - \frac{K_e}{L}\omega \quad (25)$$

以  $i_a$  和  $\omega$  为状态变量，得到状态变量方程 26 及输出方程 27。

$$\frac{d}{dt} \begin{bmatrix} i_a \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L} & -\frac{K_e}{L} \\ \frac{K_t}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} V_s \\ T_L \end{bmatrix} \quad (26)$$

$$\omega = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix} \quad (27)$$

状态变量方程 26 可简记为：

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (28)$$

其中：

$$x = \begin{bmatrix} i_a \\ \omega \end{bmatrix}, u = \begin{bmatrix} V_s \\ T_L \end{bmatrix}, y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

$$A = \begin{bmatrix} -\frac{R_a}{L} & -\frac{K_e}{B} \\ \frac{K_t}{J} & -\frac{1}{J} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

令初始条件为零（电机电流为零，转速为零），可得到电机的传递函数：

$$G_m(s) = C(sI - A)^{-1}B + D$$

$$= \frac{1}{JLs^2 + (BL + JR_a)s + BR_a + K_e K_t} \begin{bmatrix} K_t & -(Ls + R_a) \end{bmatrix} \quad (29)$$

由是可以发现，该系统是双输入，负载扰动  $T_L$  是一个独立的输入，在求解输入输出传递函数时，无需考虑这一项。

代入有关数值，得到：

$$G_m(s) = \frac{9192}{s^2 + 253s + 3959} \quad (30)$$

接下来，需要求得电机角速度  $\omega$  到螺旋桨推力  $T$  的传递函数。这里，我们认为电机的角速度即为螺旋桨的角速度。由螺旋桨模型可知，推力  $T$  与角速度  $\omega$  的平方成正比。从数学角度，可以在工作点附近泰勒展开取一次项进行线性化，但是这样与工作点（初始条件） $\omega_0$  的选取有关，产生的结果是，每一个工作点对应一个传递函数；另外，笔者对线性化的精度表示怀疑，因此放弃继续采用数学推导的方法求取传递函数。

## 4.2 系统辨识法

系统辨识是根据系统的输入输出时间函数来确定描述系统行为的数学模型，用当前可测量的系统的输入和输出建立一个能模仿真实系统行为的模型。输入信号设计为采样频率  $500Hz$ ，总时长  $50s$ ，幅度为  $100$ ，扫频频率  $0-10Hz$  的扫频电压信号，其时间序列曲线见图 3，频域特性曲线见图 4，生成代码见附件。在 Simulink 中的“Vertical Thruster”添加一个名为 Signal 的扫频信号输

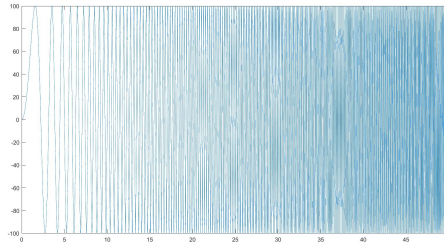


Figure 3: 扫频信号时序曲线

入，在其输出端对深度  $z$  以  $500Hz$  频率采样，并导出至 Workspace，如图 7 所示。之后将输入输出信号导入 System Identification 工具箱中，估计其传递函数。该信号产生的深度  $z$  及垂向速度  $w$  分别如图 5, 图 6 所示。严谨起见，先对

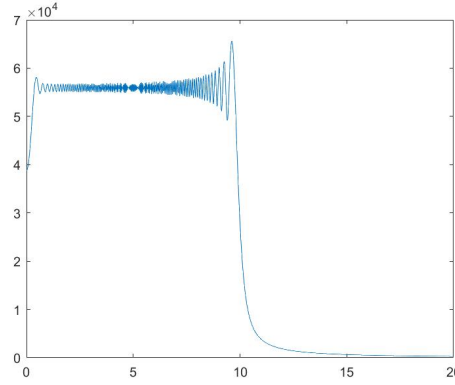


Figure 4: 扫频信号频率曲线

角速度进行验证，在角速度添加采样并导入 Workspace，计算得传递函数如式 31 所示。

$$G_m(s) = \frac{8285}{s^2 + 216.6s + 5204} \quad (31)$$

对比之前计算结果，误差可以接受。于是直接计算输入电压  $V_s$  到深度  $z$  的传递函数，输入输出信号见图 8，结果如式 32 所示，图 9 展示了辨识模型的输出和 ROV 真实的深度变化，其中黑线为实际输出，蓝线为辨识模型输出，拟合度为 96.59%，误差曲线如图 10 所示。

$$G_1(s) = \frac{0.002083s + 0.0003037}{s^3 + 0.3784s^2 + 0.02072s + 4.425e - 14} \quad (32)$$

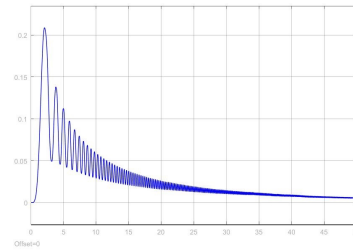
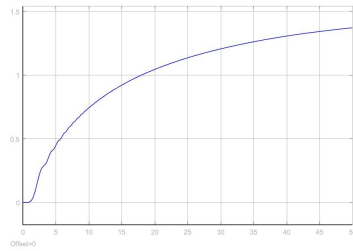


Figure 5: Vertical Thruster 深度响应 Figure 6: Vertical Thruster 垂向速度响应

相同的扫频信号，添加  $180^\circ$  的相位差，这保证了控制电压之和为零，以减小对 ROV 纵向位置的扰动。将两个扫频信号分别输入“Horizontal Thruster”1 和 2，如图 7 所示。艏向角  $\psi$  响应如图 11 所示，Yaw 角速度  $r$  响应如图 12 所示。导入 System Identification 工具箱中的输入输出数据如图 13 所示，结果如式 33 所示，图 14 展示了辨识模型的输出和 ROV 真实的深度变化，其中黑线为实际输出，蓝线为辨识模型输出，拟合度为 94.68%，误差曲线如图 15 所示。

$$G_2(s) = \frac{0.001417s - 0.1519}{s^3 + 17.47s^2 + 25.14s + 5.308e - 11} \quad (33)$$



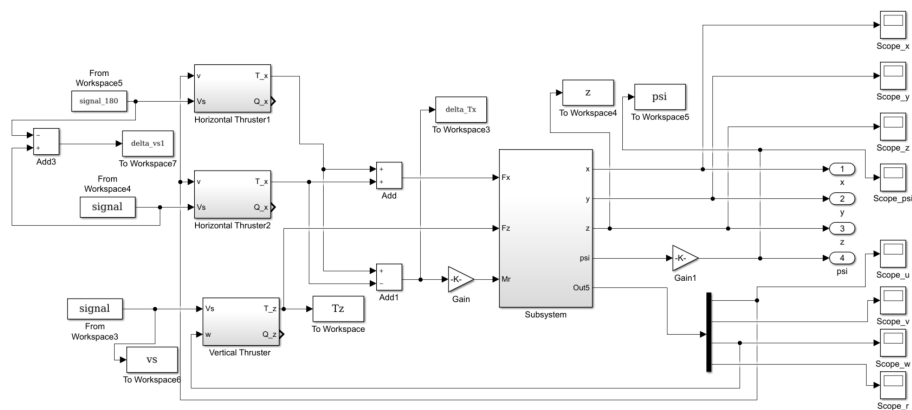


Figure 7: 系统辨识采样设计

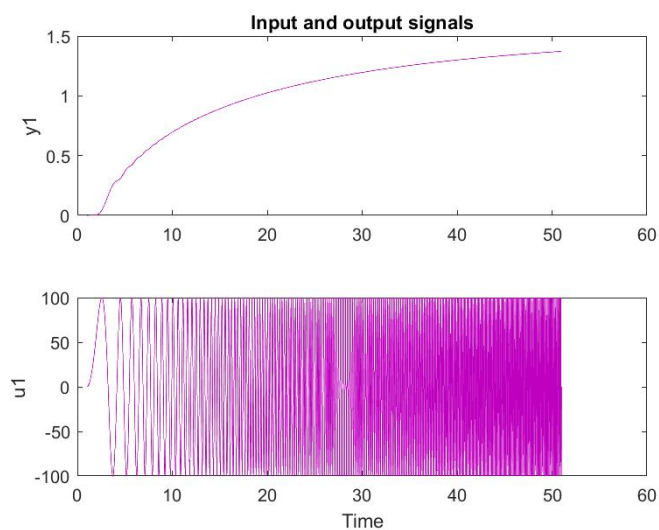


Figure 8: 输入输出信号

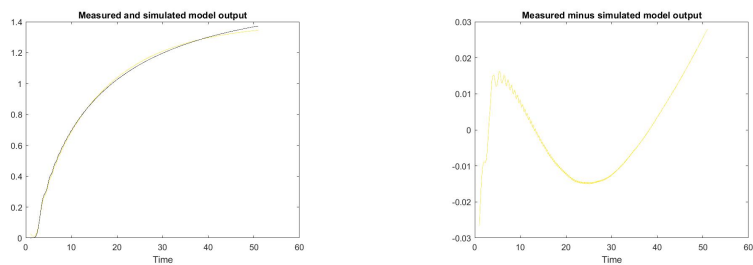


Figure 9: 电压-深度系统辨识结果时域图

Figure 10: 电压-深度系统辨识结果误差曲线

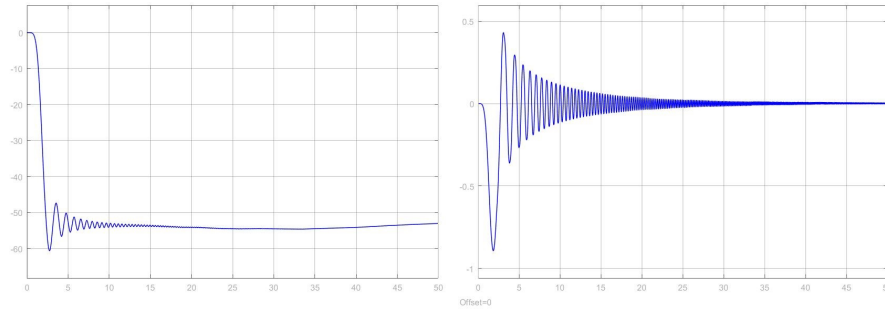


Figure 11: Vertical Thruster 艏向角响应 Figure 12: Vertical Thruster Yaw 角速度响应

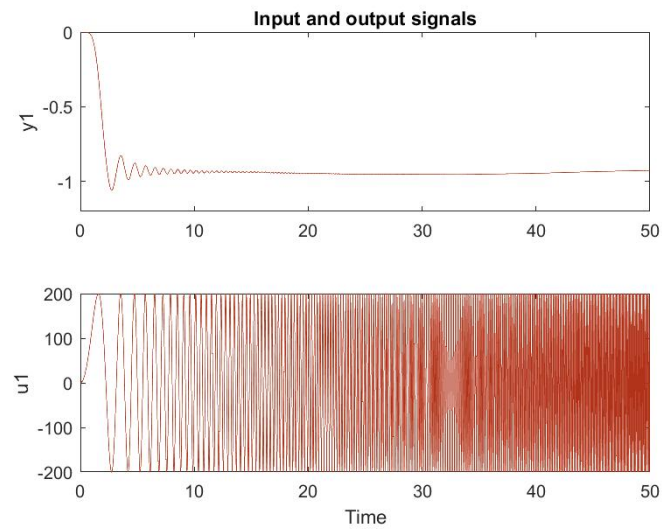


Figure 13: 输入输出信号

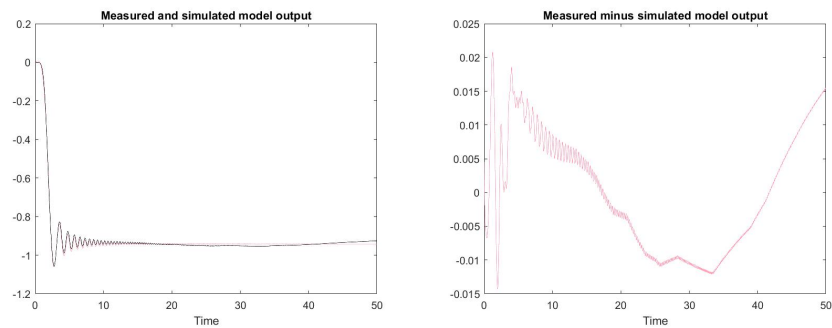


Figure 14: 电压差-艏向角系统辨识结果时域图 Figure 15: 电压差-艏向角系统辨识结果误差曲线

## 5 控制器设计

### 5.1 深度控制器设计

将得到的电压-深度传递函数  $G_1(s)$  导入 PID Tuner 工具箱中, 利用其得到相对较好的 PID 控制器参数 ( $K_p, K_i, K_d$ ). 将响应速度和鲁棒性均调至最大, 生成响应曲线如图 16 所示。其中  $K_p = 200.8, K_i = 11.6, K_d = 872.4$ , 超调量为 3.54%。由此得到深度控制器  $K_1(s)$ 。

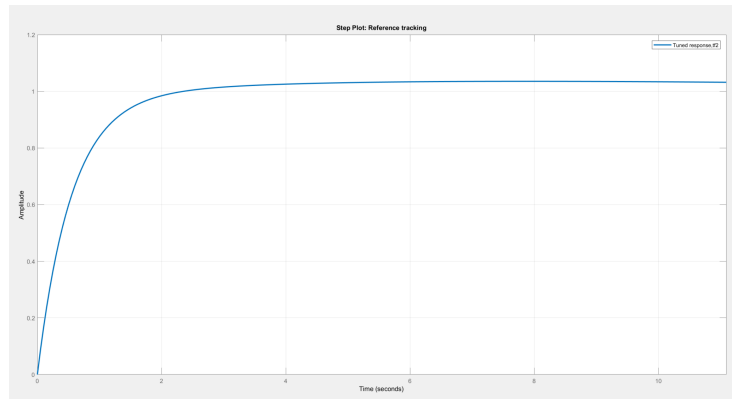


Figure 16: 深度控制器响应曲线

### 5.2 艏向控制器设计

将得到的电压差-艏向角传递函数  $G_2(s)$  导入 PID Tuner 工具箱中。得到响应曲线如图??所示, 其中  $K_p = -0.572, K_i = 0.021, K_d = -0.918$ , 超调量为 10.5%。由此得到艏向控制器  $K_2(s)$ 。

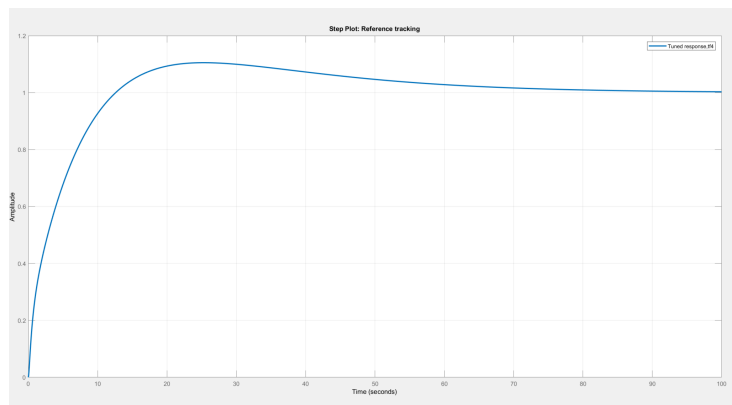


Figure 17: 艏向控制器响应曲线

## 6 非线性仿真验证

将控制器  $K_1(s)$  以及  $K_2(s)$  与运动仿真模型连接为闭环控制系统，并进行深度与艏向运动自动控制仿真：控制器  $K_1(s)$  的输入为深度误差，输出为垂向推进器控制电压；控制器  $K_2(s)$  的输入为艏向误差，输出为左右联动推进器控制电压。添加控制器后的 Simulink 模型如图 18 所示。接下来对控制器的性能进行测试。对于垂向运动，在零初始条件下给定目标垂向移动 1m，得深度响应曲线和垂向速度响应曲线分别如图 19、图 21 所示。对于艏向运动，在零初始条件下给定目标艏向角  $180^\circ$ ，得艏向角响应曲线和角速度响应曲线分别如图 20、图 22 所示。

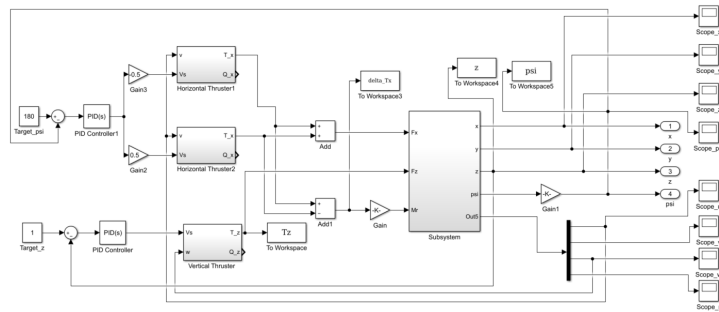


Figure 18: 闭环控制系统

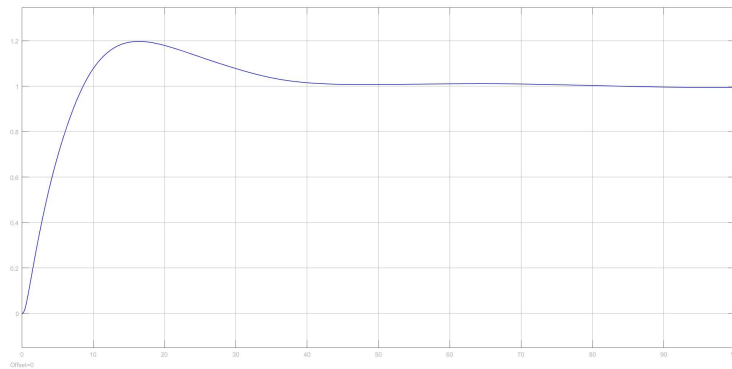


Figure 19: 添加深度控制器后深度响应曲线

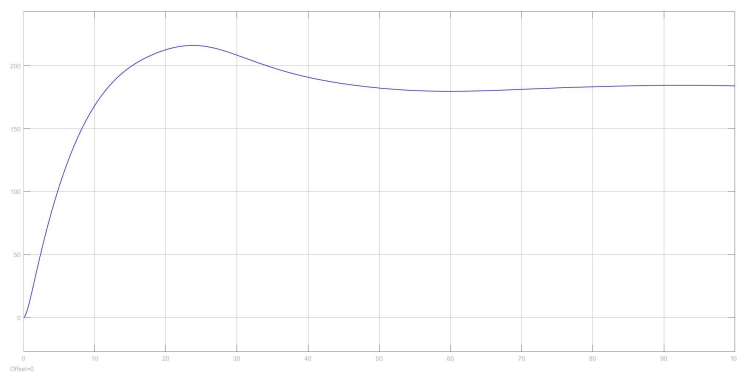


Figure 20: 添加艏向控制器后艏向角响应曲线

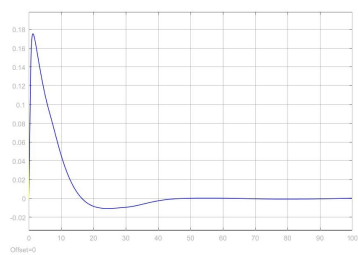


Figure 21: 添加深度控制器后垂向速度响应曲线

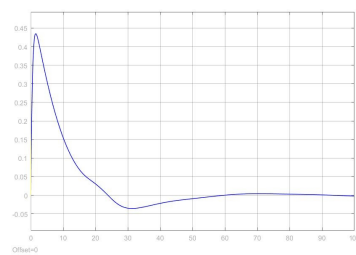


Figure 22: 添加艏向控制器后艏向角速度响应曲线

## 7 总结

观察响应曲线可以看出,无论是深度还是艏向角,超调都不止在 PID Tuner 中的计算量。这说明系统辨识得到的模型与真实模型存在较大差异,这是不可避免的。首先,系统辨识需要选择传递函数分子分母的阶数,问题在于并不知道完全由数学推导得到的正确的传递函数的阶数,于是只能根据拟合度选择。然而拟合度仅仅是对特定的输入数据而言,所以如果按照拟合度越高传递函数越准的原则选取,不可避免地会出现过拟合的情况,导致在真实模型下出现较大偏差。其次,这是一个非线性系统,传递函数本身就是一种线性化,必然会有误差存在。另一方面,在用 PID Tuner 整定 PID 参数时,无法考虑输出阈值,而在 Simulink 中,输入电压最高为 100V,因此这会导致实际情况与在 PID Tuner 中的计算结果有较大差异。

从响应时间来看,下潜 1 米需要 10 秒,转向 180 度需要十几秒,这是比较符合实际的。从稳态响应来看,首先几乎没有稳态误差,令人欣慰,但达到稳态却耗时很长,这一点通过调节 PID 参数几乎得不到改善,说明在求得的传递函数模型下,已经接近 1DOF 的 PID 控制器所能达到的最好效果。

综合来看,如果是用于定性分析该 DOE ROV 模型,系统辨识得到的模型已经足够。遗憾的是,未能采用线性化的方法从数学角度得到传递函数,针对这一点,未来需要温习复变函数与进一步学习控制理论。

## 8 附录

```

1  %% Generate the sweep wave.
2  f_sample = 500; % Sampling frequency is 500Hz
3  Time = 50; %Sampling overall time is 50s
4  N = f_sample*Time;% sweep wave sample numbers
5  t = 0:1/f_sample:(N-1)/f_sample;% sweep wave time instants
6  t = t';% tranpose the array
7  ampl = 100; % the amplitude of the sweep wave
8  sweep_wave = ampl.*chirp(t,0,50,10,'linear',-90); % generate ...
    0-10Hz sweep wave
9  for i = 1:f_sample*Time
10     if abs(sweep_wave(i)) ≤ 1e-9
11         sweep_wave(i) = 0;
12     end
13 end
14 signal = [t,sweep_wave];
15
16 sweep_wave_180 = ampl.*chirp(t,0,50,10,'linear',90); % ...
    generate 0-10Hz sweep wave
17 for i = 1:f_sample*Time
18     if abs(sweep_wave_180(i)) ≤ 1e-9
19         sweep_wave_180(i) = 0;
20     end
21 end
22 signal_180 = [t,sweep_wave_180];%generate signal with 180 ...
    phase angle
23
24 %% Plot time-series data and frequency-domain data.
25 % format short;
26 f = linspace(0,f_sample,N);
27 figure(1);
28 plot(t(1:N), sweep_wave(1:N)); % display the time-series data
29 xlim([0 50]);
30 figure(2);
31 sweep_wave_fft = fft(sweep_wave);
32 plot(f,abs(sweep_wave_fft(1:N)));% display the ...
    frequency-domain data
33 xlim([0 20]);

```