

# 浙江大学实验报告

专业： 电子信息工程

姓名： 邢毅诚

学号： 3190105197

日期： 2021-11-26

地点： 教二-213

课程名称： 控制理论（乙）

指导老师： 姚维、韩涛

成绩：

实验名称： 控制系统的时域分析

实验类型： 验证实验

同组学生姓名： 无

## 一、 实验目的

1. 掌握用 MATLAB 进行系统的时域分析的方法
2. 熟悉 Simulink 仿真环境

## 二、 实验内容

1. 二阶系统，状态方程模型如下：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.5572 & -0.7814 \\ 0.7814 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$
$$y = [1.9691 \quad 6.4493] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

- (1) 画出系统的单位阶跃相应曲线
- (2) 画出系统的冲击相应曲线
- (3) 当系统的初始状态为  $X_0 = [1, 0]$
- (4) 当系统的初始状态为零时，画出系统斜坡输入相应

要求：

- (1) 编程画出单位单位阶跃响应曲线、冲击响应曲线、系统的零输入响应、斜坡输入响应。
- (2) 在 Simulink 仿真环境中，组成系统的仿真框图，得到系统的单位阶跃响应曲线

2. 二阶系统标准传递函数形式如下：

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (1)$$

使用阶跃相应曲线分析特征参量  $\xi$  和  $\omega_n$  对二阶系统性能的影响。

要求：

- (1) 在参量  $\omega_n$  (自由振荡频率) 为 1 时，参量  $\xi$  (阻尼比) 在无阻尼 ( $\xi = 0$ )，欠阻尼 ( $\xi = 0.5$ )，临界阻尼 ( $\xi = 1$ ) 和过阻尼 ( $\xi = 2$ ) 状态下对二阶系统性能的影响。

- (2) 在参量  $\omega_n$  在欠阻尼 ( $\xi = 0.5$ ) 情况下,  $\omega_n$  分别取 1、2、3 时对二阶系统性能的影响。
3. 在初始状态为  $[1, 1, -1]^T$ , 求三阶系统的单位阶跃相应, 单位脉冲相应以及零输入响应:

$$H(s) = \frac{5(s^2 + 5s + 6)}{s^3 + 6s^2 + 10s + 8} \quad (2)$$

### 三、 实验内容与实验数据

#### 1. 实验 1

##### 1. MATLAB 编程

首先, 我们需要获得系统的空间状态模型, 并利用 ss 函数进行转换, 具体代码如下:

```
1 A = [-0.5572, -0.7814; 0.7814, 0];  
2 B = [1; 0];  
3 C = [1.9691 6.4493];  
4 D = [0];  
5 t = 0:0.04:20; % 201 points  
6 sys = ss(A,B,C,D);  
7 x0 = [1, 0];
```

再据此获得各响应曲线:

- 单位阶跃响应曲线

利用 step 函数, 我们可以获得并绘制出对应系统的单位阶跃响应曲线, 具体代码如下:

```
1 %单位阶跃响应曲线  
2 figure;  
3 grid on;  
4 [y0] = step(sys,t);  
5 plot(t,y0,'linewidth',1.2);  
6 title('系统的单位阶跃响应曲线','fontsize',24)  
7 xlabel('t/s','fontsize',24);  
8 ylabel('c(t)','fontsize',24);  
9 set(gca,'FontSize',24);  
10 grid on;
```

运行程序, 可以获得单位阶跃响应曲线的输出图像:

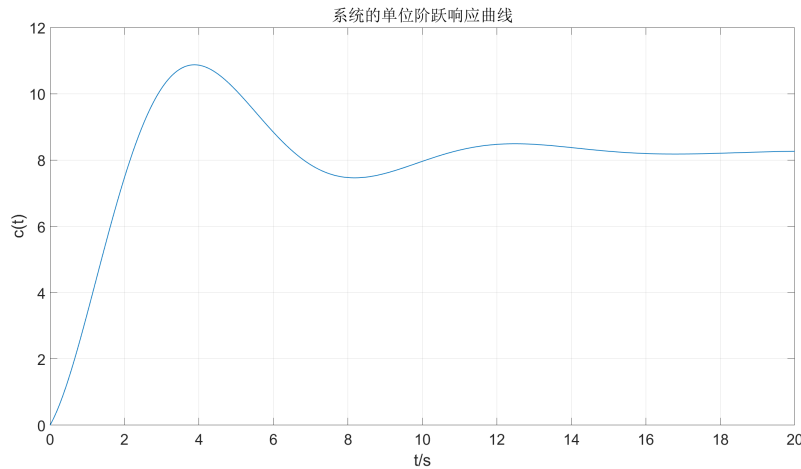


图 1: 单位阶跃响应曲线

- 冲击响应曲线

利用 `impulse` 函数，我们可以获得并绘制出对应系统的单位阶跃响应曲线，具体代码如下：

```
1 %冲击响应曲线
2 figure;
3 grid on;
4 [y0] = impulse(sys,t);
5 plot(t,y0,'linewidth',1.2);
6 title('系统的冲击响应曲线','fontsize',24)
7 xlabel('t/s','fontsize',24);
8 ylabel('c(t)','fontsize',24);
9 set(gca,'FontSize',24);
10 grid on;
```

运行程序，可以获得冲击响应曲线的输出图像：

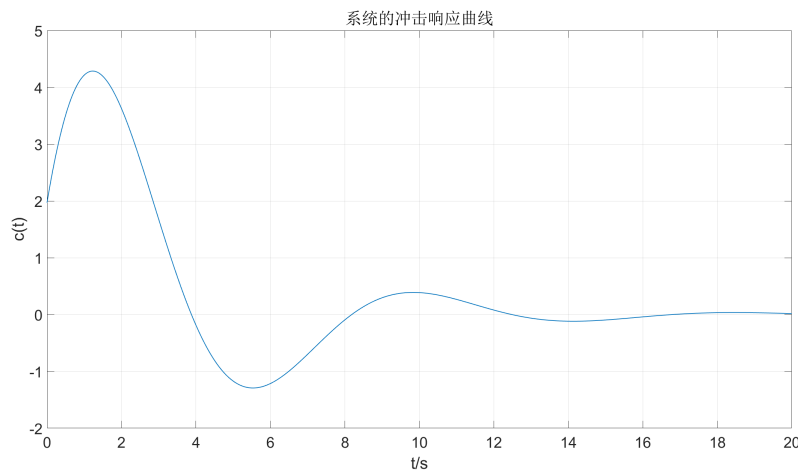


图 2: 冲击响应曲线

- 系统的零输入响应

利用 `initial` 函数，我们可以获得并绘制出对应系统的单位阶跃响应曲线，具体代码如下：

```
1 %系统的零输入响应
2 figure;
3 [y0] = initial(sys,x0,t);
4 plot(t,y0,'linewidth',1.2);
5 title('系统的零输入响应曲线','fontsize',24)
6 xlabel('t/s','fontsize',24);
7 ylabel('c(t)','fontsize',24);
8 set(gca,'FontSize',24);
9 grid on;
```

运行程序，可以获得零输入响应曲线的输出图像：

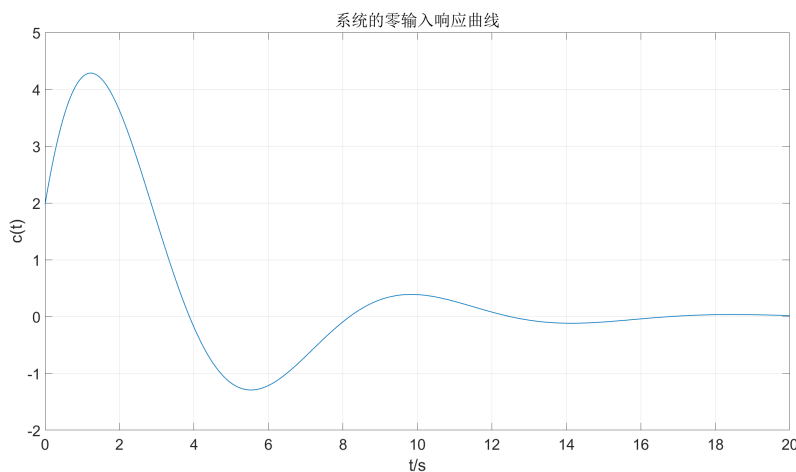


图 3: 零输入响应曲线

- 斜坡输入响应

设置斜坡函数为单位斜坡函数，利用 `lsim` 函数便可以获得对应的斜坡响应曲线，具体代码如下：

```
1 %斜坡输入响应
2 figure;
3 t = 0:0.04:20; % 201 points
4 u = t;
5 [y0] = lsim(sys,u,t);
6 plot(t,y0,'linewidth',1.2);
7 grid on;
8 title('系统的斜坡输入响应曲线','fontsize',24)
9 xlabel('t/s','fontsize',24);
10 ylabel('c(t)','fontsize',24);
11 set(gca,'FontSize',24);
```

运行程序，可以获得斜坡输入响应曲线的输出图像：

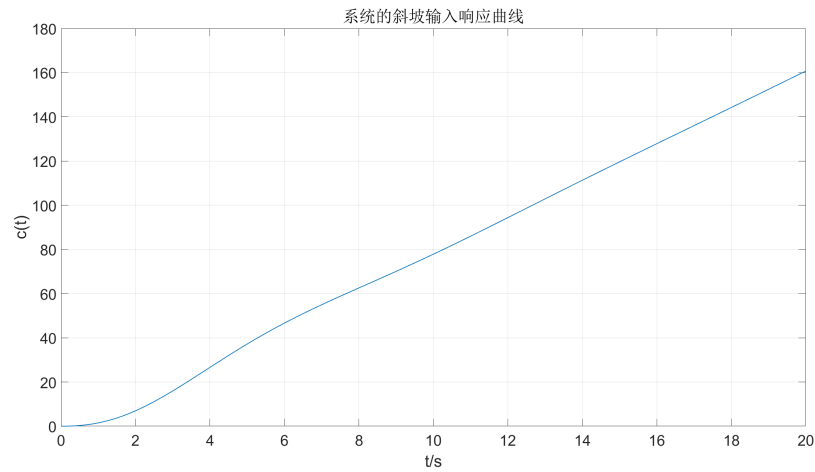


图 4: 斜坡输入响应曲线

## 2. Simulink 系统仿真

使用 Simulink 进行系统仿真，构建对应的响应框图，如下图所示：

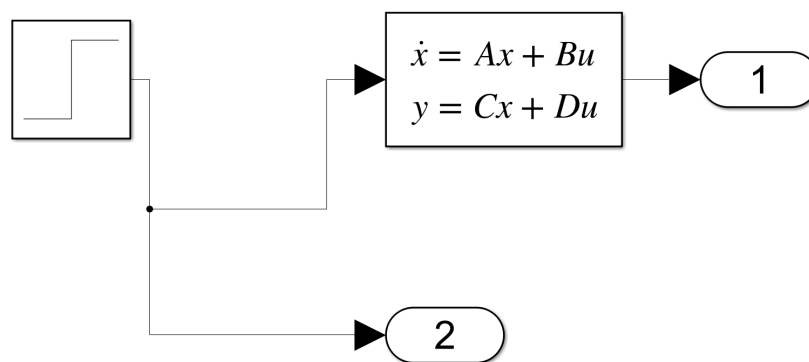


图 5: 系统仿真框图

运行程序，可以获得响应结果：

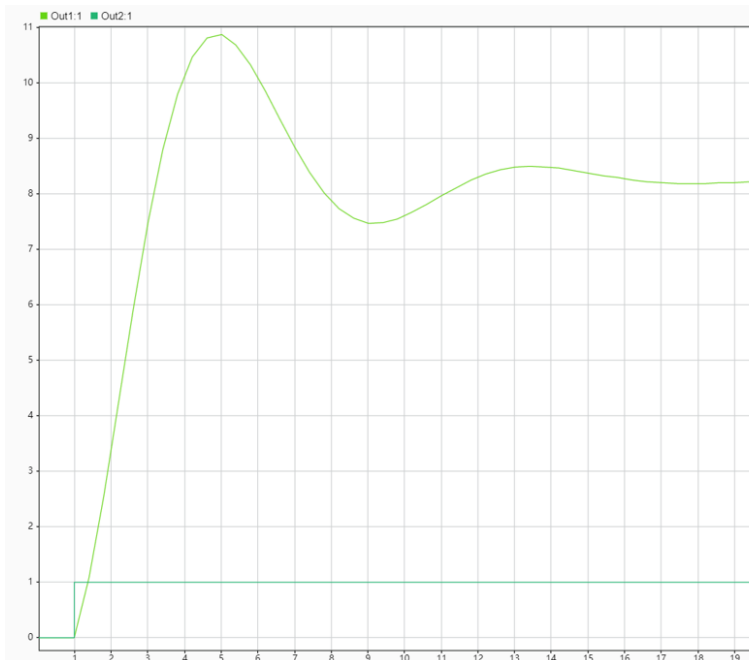


图 6: Simulink 仿真曲线

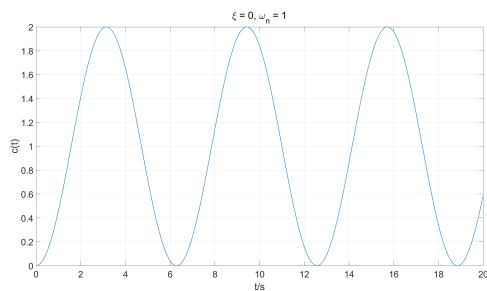
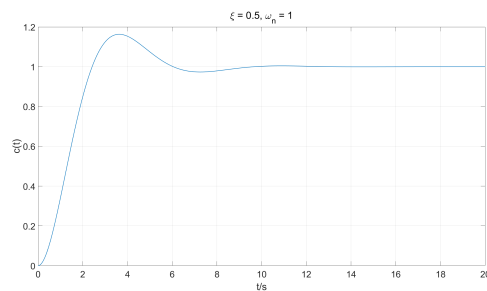
可以看到与 MATLAB 编程获得的响应曲线相同，认为实验结果正确。

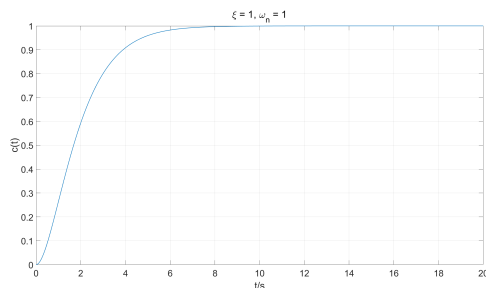
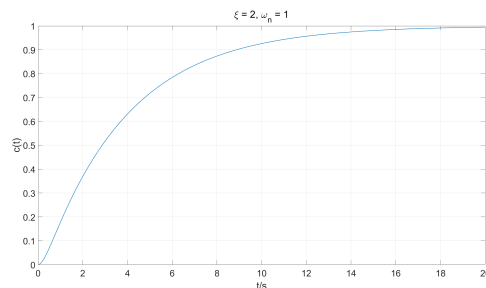
## 2. 实验 2

利用 `tf2ss()` 函数以及 `ss()` 函数，我们可以获得系统的空间状态模型，再根据 `step` 函数，我们便可以获得系统的响应曲线：

### 1. 参量 $\omega_n = 1$

设定  $\omega_n = 1$  不变，分别设置  $\xi = 0, 0.5, 1, 2$ ，获得响应曲线如下所示：

图 7:  $\omega_n = 1, \xi = 0$ 图 8:  $\omega_n = 1, \xi = 0.5$

图 9:  $\omega_n = 1, \xi = 1$ 图 10:  $\omega_n = 1, \xi = 2$ 

根据系统传递函数：

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

我们可以计算出，当  $\xi = 1$  时，系统处于临界阻尼状态，而当  $\xi < 1$  时，系统处于欠阻尼状态，当  $\xi > 1$  时，系统处于过阻尼状态，当  $\xi = 0$  时，系统处于无阻尼状态。

而观察响应曲线，当  $\xi = 0$  时，系统等幅振荡，为无阻尼状态， $\xi = 0.5$  时，系统存在超调量，为欠阻尼状态， $\xi = 1$  时，系统恰好没有超调量，为临界阻尼状态， $\xi = 2$  时，为过阻尼状态，与理论结果相符。

具体实现代码如下所示：

```

1  close all;
2  %xi = 0时的阶跃响应曲线
3  wn = 1;
4  k = 0;
5  [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
6  sys = ss(A,B,C,D);
7  figure;
8  [y0] = step(sys,t);
9  plot(t,y0,'linewidth',1.2);
10 xlabel('t/s','fontsize',24);
11 ylabel('c(t)','fontsize',24);
12 title('\xi = 0, \omega_n = 1','fontsize',24);
13 set(gca,'FontSize',24);
14 grid on;
15
16 %xi = 0.5时的阶跃响应曲线
17 wn = 1;
18 k = 0.5;
19 [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
20 sys = ss(A,B,C,D);
21 figure;
22 [y0] = step(sys,t);
23 plot(t,y0,'linewidth',1.2);
24 xlabel('t/s','fontsize',24);
25 ylabel('c(t)','fontsize',24);
26 title('\xi = 0.5, \omega_n = 1','fontsize',24);

```

```

27 set(gca, 'FontSize', 24);
28 grid on;
29
30 %xi = 1时的阶跃响应曲线
31 wn = 1;
32 k = 1;
33 [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
34 sys = ss(A,B,C,D);
35 figure;
36 [y0] = step(sys,t);
37 plot(t,y0,'linewidth',1.2);
38 xlabel('t/s','fontsize',24);
39 ylabel('c(t)','fontsize',24);
40 title('\xi = 1, \omega_n = 1','fontsize',24)
41 set(gca, 'FontSize', 24);
42 grid on;
43
44 %xi = 2时的阶跃响应曲线
45 wn = 1;
46 k = 2;
47 [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
48 sys = ss(A,B,C,D);
49 figure;
50 [y0] = step(sys,t);
51 plot(t,y0,'linewidth',1.2);
52 xlabel('t/s','fontsize',24);
53 ylabel('c(t)','fontsize',24);
54 title('\xi = 2, \omega_n = 1','fontsize',24)
55 set(gca, 'FontSize', 24);
56 grid on;

```

## 2. 参量 $\xi = 0.5$

设定  $\xi = 0.5$  不变，分别设置  $\omega_n = 1, 2, 3$ ，获得响应曲线如下所示：

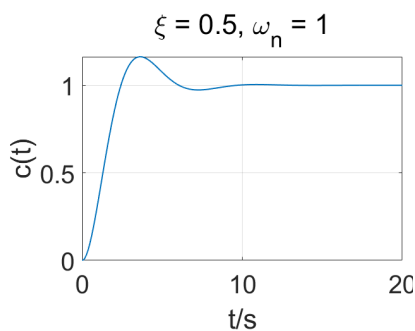


图 11:  $\omega_n = 1, \xi = 0$

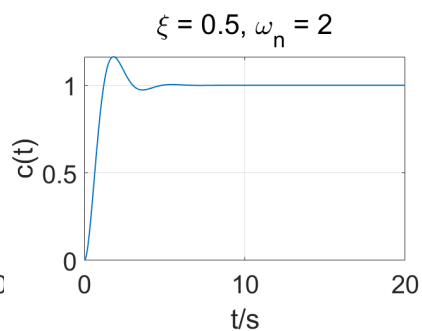


图 12:  $\omega_n = 2, \xi = 0.5$

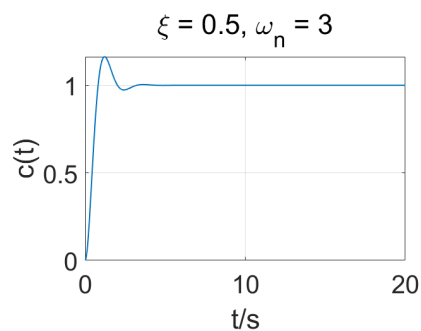


图 13:  $\omega_n = 3, \xi = 0.5$

可以看到，当处于过阻尼状态下， $\omega_n$  越大，系统的调整时间以及峰值时间便会越短，而超调量则没有明显的变化，这与我们的实验结果相符。



具体代码如下所示：

```
1  close all;
2
3  % $\omega_n = 1$ 时的阶跃响应曲线
4  wn = 1;
5  k = 0.5;
6  [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
7  sys = ss(A,B,C,D);
8  figure;
9  [y0] = step(sys,t);
10 plot(t,y0,'linewidth',1.2);
11 xlabel('t/s','fontsize',24);
12 ylabel('c(t)','fontsize',24);
13 title('\xi = 0.5, \omega_n = 1','fontsize',24)
14 set(gca,'FontSize',24);
15 grid on;
16
17 % $\xi = 1$ 时的阶跃响应曲线
18 wn = 2;
19 k = 0.5;
20 [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
21 sys = ss(A,B,C,D);
22 figure;
23 [y0] = step(sys,t);
24 plot(t,y0,'linewidth',1.2);
25 xlabel('t/s','fontsize',24);
26 ylabel('c(t)','fontsize',24);
27 title('\xi = 0.5, \omega_n = 2','fontsize',24)
28 set(gca,'FontSize',24);
29 grid on;
30
31 % $\xi = 2$ 时的阶跃响应曲线
32 wn = 3;
33 k = 0.5;
34 [A,B,C,D] = tf2ss([wn^2],[1,2*k*wn,wn^2]);
35 sys = ss(A,B,C,D);
36 figure;
37 [y0] = step(sys,t);
38 plot(t,y0,'linewidth',1.2);
39 xlabel('t/s','fontsize',24);
40 ylabel('c(t)','fontsize',24);
41 title('\xi = 0.5, \omega_n = 3','fontsize',24)
42 set(gca,'FontSize',24);
43 grid on;
```

### 3. 实验 3

首先，利用 `tf()` 函数，`tf2ss()` 函数，`ss()` 函数，我们可以获得系统的状态方程以及零输入时对应的方程，具体代码如下：

```
1 close all;
2 s = tf('s');
3 num = [5,25,30];
4 den = [1,6,10,8];
5 [A,B,C,D] = tf2ss(num,den);
6 sys = ss(A,B,C,D);
7 sys1 = tf(num, den);
8 HS=ss(sys1);
9 x0 = [1;1;-1];
10 t = [0:0.01:10];
```

再据此获得各响应曲线

- 单位阶跃响应曲线

单位阶跃响应曲线的代码如下所示：

```
1 %系统的单位阶跃响应曲线
2 figure;
3 [y0] = step(sys,t);
4 [y1]=initial(HS,x0,t);
5 plot(t,y0+y1,'linewidth',1.2);
6 title('系统的单位阶跃响应曲线','fontsize',24)
7 xlabel('t/s','fontsize',24);
8 ylabel('c(t)','fontsize',24);
9 set(gca,'FontSize',24);
```

运行程序，便可以获得单位阶跃响应曲线的输出图像：

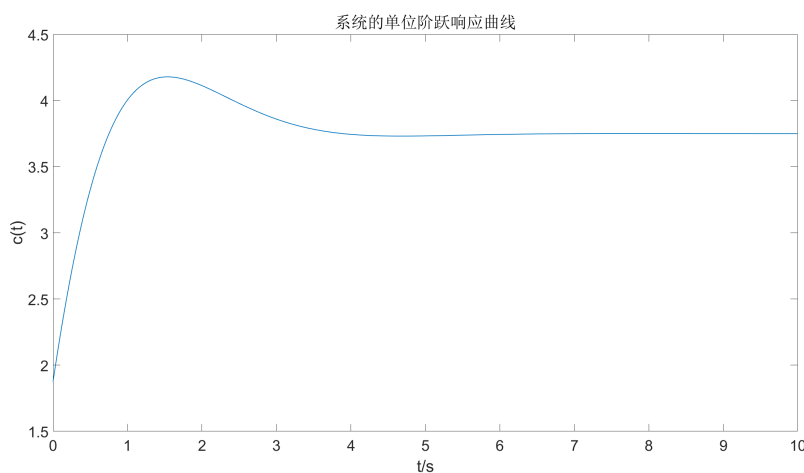


图 14: 单位阶跃响应曲线

- 单位脉冲响应曲线

单位脉冲响应曲线的代码如下所示：

```
1 %系统的单位脉冲响应曲线
2 figure;
3 [y0] = impulse(sys,t);
4 plot(t,y0+y1,'linewidth',1.2);
5 title('系统的单位脉冲响应曲线','fontsize',24)
6 xlabel('t/s','fontsize',24);
7 ylabel('c(t)','fontsize',24);
8 set(gca,'FontSize',24);
```

运行程序，便可以获得单位脉冲响应曲线的输出图像：

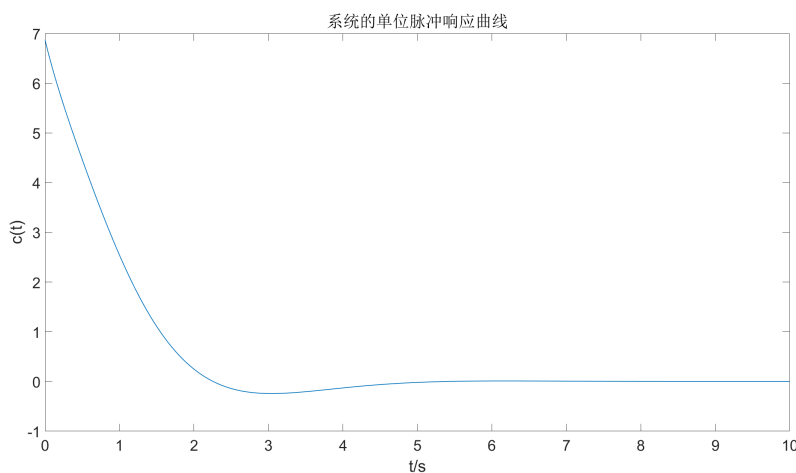


图 15: 单位脉冲响应曲线

- 零输入响应曲线

零输入响应曲线的代码如下所示：

```
1 %系统的零输入相应曲线
2 figure;
3 [y0] = initial(HS,x0,t);
4 plot(t,y0,'linewidth',1.2);
5 title('系统的零输入响应曲线','fontsize',24)
6 xlabel('t/s','fontsize',24);
7 ylabel('c(t)','fontsize',24);
8 set(gca,'FontSize',24);
```

运行程序，便可以获得零输入响应曲线的输出图像：

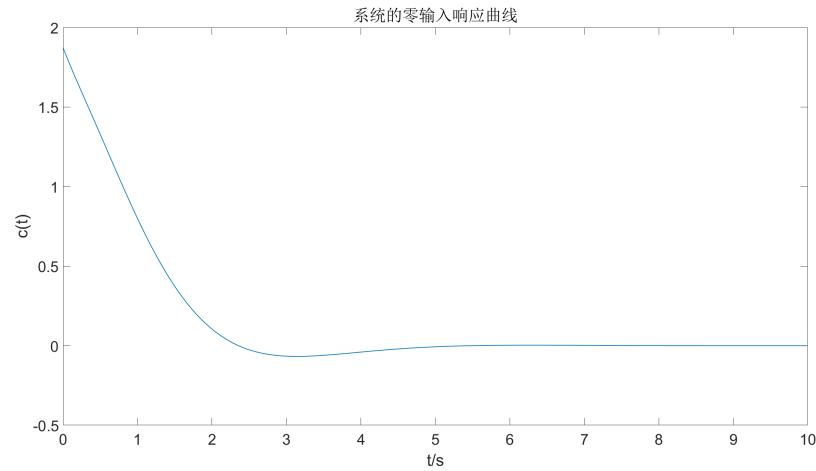


图 16: 零输入响应曲线

#### 四、心得与体会

在本次实验中，我们进行了控制系统的时域分析相关实验，通过这次实验，我理解了零输入响应曲线，初始状态等概念的真正意义，同时，这是我第一次尝试使用 simulink 软件进行系统的建模与仿真，也取得了一定的成效。总体而言，通过本次实验，我收获颇多。