



北京理工大学
Beijing Institute of Technology

本科生实验报告

课程名称：控制理论基础

实验名称：控制理论基础

任课教师：				实验教师：			
实验日期：				实验地点：			
实验类型：	<input checked="" type="checkbox"/> 原理验证 <input type="checkbox"/> 综合设计 <input type="checkbox"/> 自主创新						
学生姓名：		班级：		学号：			
学 院：	信息与电子学院			专 业：	电子信息工程（徐特立英才班）		
组 号：		同 组 同 学：					
成 绩：							



信息与电子学院

SCHOOL OF INFORMATION AND ELECTRONICS

目录

实验 1 控制系统的模型建立	3
一、实验目的	3
二、实验原理	3
三、实验内容	4
四、实验总结	10
实验 2 控制系统的暂态特性分析.....	11
一、实验目的	11
二、实验原理	11
三、实验内容	11
四、实验总结	19
实验 3 根轨迹分析.....	21
一、实验目的	21
二、实验原理	21
三、实验内容	21
四、实验总结	28
实验 4 系统的频率特性分析	30
一、实验目的	30
二、实验原理	30
三、实验内容	30
四、实验总结	36
实验 6 极点配置与全维状态观测器的设计	38
一、实验目的	38
二、实验原理	38
三、实验内容	38
四、实验总结	41
总结与反思	43

实验 1 控制系统的模型建立

一、 实验目的

1. 掌握利用 MATLAB 建立控制系统模型的方法。
2. 掌握系统的各种模型表述及相互之间的转换关系。
3. 学习和掌握系统模型连接的等效变换。

二、 实验原理

2.1 系统模型的 MATLAB 描述

系统的模型描述了系统的输入、输出变量以及内部各变量之间的关系，可以用微分方程、传递函数、状态空间方程等方式表征系统模型。在 MATLAB 中，通常采取多项式型传递函数（TF）模型、零极点型传递函数（ZPK）模型和状态空间方程（SS）模型实现对一个系统模型的描述，并且这三种模型可以实现相互转换。

2.2 系统模型的连接

在实际应用中，整个系统是由多个单一模型组合而成，基本的连接方式有串联连接、并联连接和反馈连接，结构框图及其等效传递函数如下图所示：

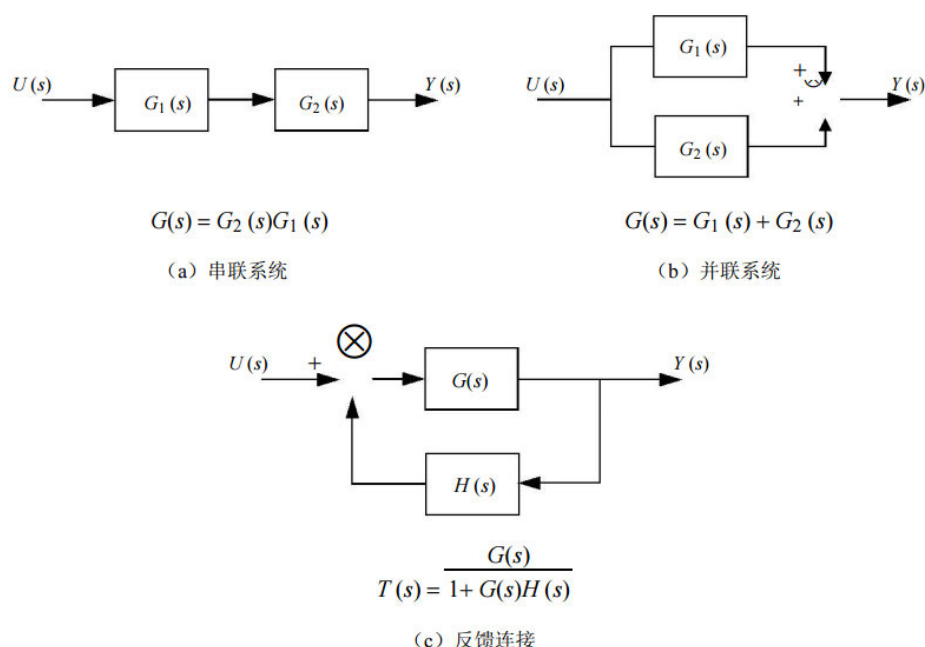


图 1-1：串联连接、并联连接和反馈连接

在 MATLAB 中可以直接使用 “*” 运算符实现串联连接，使用 “+” 运算符实现并联连接。反馈系统传递函数求解可以通过命令 `feedback` 实现。

三、 实验内容

3.1 第 1 题

1. 实现代码

```
1. % 定义系统传递函数的分子和分母多项式系数
2. num = [2, 18, 40];    % 分子多项式系数: 2s^2 + 18s + 40
3. den = [1, 5, 8, 6];   % 分母多项式系数: s^3 + 5s^2 + 8s + 6
4.
5. Gtf = tf(num, den)    % 创建传递函数对象
6.
7. Gzpk = zpk(Gtf)       % 转换为零极点增益形式
8.
9. Gss = ss(Gtf)         % 转换为状态空间形式
10.
11. % 绘制零极点图
12. figure;              % 创建新图形窗口
13. pzmap(Gzpk);          % 绘制零极点图
14. title('系统零极点图'); % 添加标题
15. xlabel('实轴');       % x 轴标签
16. ylabel('虚轴');       % y 轴标签
17. grid on;             % 显示网格
```

2. 实验结果

系统的传递函数模型如图 1-2 所示:

$$G_{tf} = \frac{2s^2 + 18s + 40}{s^3 + 5s^2 + 8s + 6}$$

连续时间传递函数。

图 1-2: 传递函数模型

零极点增益模型如图 1-3 所示:

$$G_{zpk} = \frac{2(s+5)(s+4)}{(s+3)(s^2 + 2s + 2)}$$

连续时间零点/极点/增益模型。

图 1-3: 零极点增益模型

系统的状态空间方程模型如图 1-4 所示：

Gss =

A =

	x1	x2	x3
x1	-5	-2	-1.5
x2	4	0	0
x3	0	1	0

B =

	u1
x1	4
x2	0
x3	0

C =

	x1	x2	x3
y1	0.5	1.125	2.5

D =

	u1
y1	0

连续时间状态空间模型。

图 1-4：状态空间方程模型

系统的零极点图如图 1-5 所示：

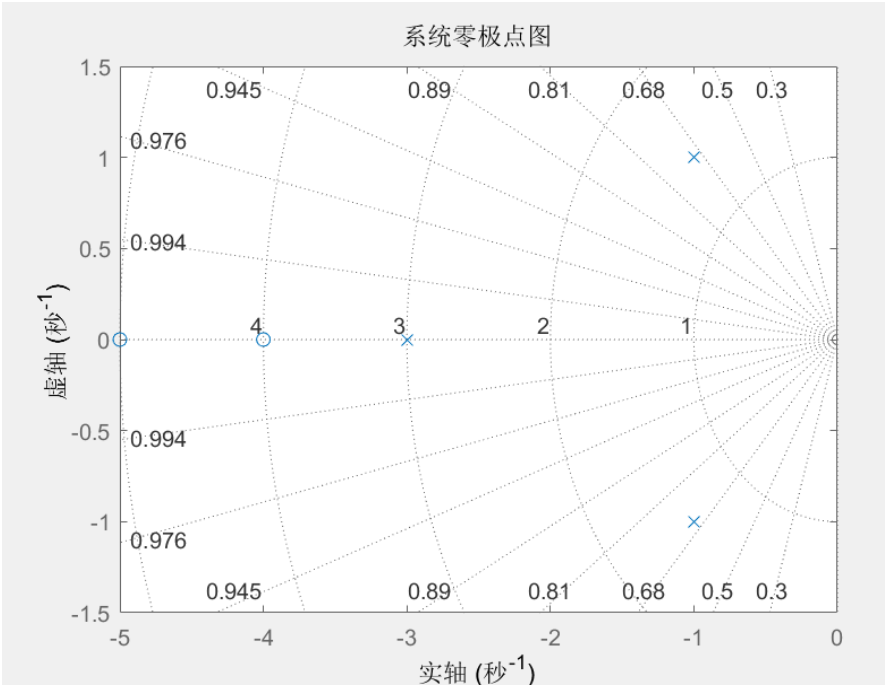


图 1-5：系统零极点图

3.2 第2题

1. 实现代码

```
1. % 定义状态空间模型的矩阵
2. A = [0, 1, 0, 0;
3.      0, 0, 1, 0;
4.      0, 0, 0, 1;
5.      -1, -2, -3, -4];
6.
7. B = [0; 0; 0; 1];
8. C = [10, 2, 0, 0];
9. D = 0;
10.
11. % 创建状态空间对象
12. Gss = ss(A, B, C, D)
13.
14. % 转换为零极点增益形式
15. Gzpk = zpk(Gss)
16.
17. % 绘制零极点图
18. figure;
19. pzmap(Gzpk);
20. title('系统零极点分布图');
21. xlabel('实轴');
22. ylabel('虚轴');
23. grid on;
24.
25. % 转换为传递函数形式
26. Gtf = tf(Gzpk)
```

2. 实验结果

系统的传递函数模型如图 1-6 所示：

$$G_{tf} = \frac{2s + 10}{s^4 + 4s^3 + 3s^2 + 2s + 1}$$

连续时间传递函数。

图 1-6：传递函数模型

零极点增益模型如图 1-7 所示：

Gzpk =

$$\frac{s^2 (s+5)}{(s+3.234) (s+0.6724) (s^2 + 0.0936s + 0.4599)}$$

连续时间零点/极点/增益模型。

图 1-7：零极点增益模型

系统的状态空间方程模型如图 1-8 所示：

Gss =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	0	1	0
x3	0	0	0	1
x4	-1	-2	-3	-4

B =

	u1
x1	0
x2	0
x3	0
x4	1

C =

	x1	x2	x3	x4
y1	10	2	0	0

D =

	u1
y1	0

图 1-8：状态空间方程模型

系统的零极点图如图 1-9 所示：

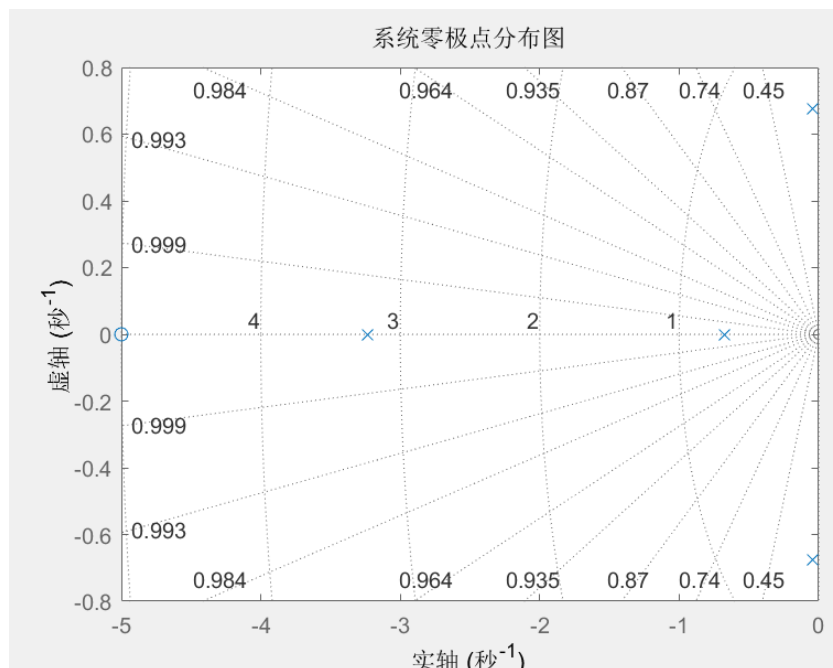


图 1-9：系统的零极点图

3.3 第 4 题

1. 实现代码

```

1. % 1. 定义各子系统的传递函数分母
2. den1 = [1, 1];
3. den2 = [0.5, 1];
4. den3 = [1, 0];
5.
6. % 2. 创建各子系统传递函数
7. G1 = tf(1, den1);
8. G2 = tf(1, den2);
9. H = tf(1, den2);
10. G3 = tf(3, den3);
11.
12. % 3. 构建前向通道传递函数
13. G0 = (G1 + G2) * G3;
14. disp('前向通道传递函数 G0:');
15. disp(G0);
16.
17. % 4. 构建闭环系统传递函数
18. G = feedback(G0, H)

```

2. 实验结果

该系统的闭环传递函数如图 1-10 所示：

$G =$

$$\frac{2.25 s^2 + 7.5 s + 6}{0.25 s^4 + 1.25 s^3 + 2 s^2 + 5.5 s + 6}$$

连续时间传递函数。

图 1-10: 闭环传递函数模型

3.4 第 5 题

1. 实现代码

```
1. % 1. 定义各子系统的传递函数分子和分母
2. % 前向通道传递函数 1: 10/(s+1)
3. num1 = 10;
4. den1 = [1, 1];
5. Gtf1 = tf(num1, den1);
6.
7. % 前向通道传递函数 2: 2/(s^2+s)
8. num2 = 2;
9. den2 = [1, 1, 0];
10. Gtf2 = tf(num2, den2);
11.
12. % 反馈环节 H2: (s+3)/(s+2)
13. num3 = [1, 3];
14. den3 = [1, 2];
15. H2 = tf(num3, den3);
16.
17. % 反馈环节 H1: (5s)/(s^2+6s+8)
18. num4 = [5, 0];
19. den4 = [1, 6, 8];
20. H1 = tf(num4, den4);
21.
22. % 2. 构建内环反馈系统 (Gtf2 与 H2 组成内环)
23. G1 = feedback(Gtf2, H2, 1); % 正反馈连接
24.
25. % 3. 串联前向通道传递函数
26. G2 = G1 * Gtf1;
27.
28. % 4. 构建外环反馈系统 (G2 与 H1 组成外环)
29. G = feedback(G2, H1) % 负反馈连接
```

2. 实验结果

该系统的闭环传递函数如图 1-11 所示：

$$G = \frac{20 s^3 + 160 s^2 + 400 s + 320}{s^6 + 10 s^5 + 35 s^4 + 44 s^3 + 82 s^2 + 116 s - 48}$$

连续时间传递函数。

图 1-11：闭环传递函数模型

四、实验总结

4.1 实验中遇到的问题及解决方法

问题：在系统连接时混淆了各种连接方式，导致传递函数计算错误。

解决方案：明确控制系统串联连接、并联连接和反馈连接这三种连接方式的特点以及传递函数计算公式，在利用命令 `feedback` 实现反馈连接时要特别注意反馈方式，在正反馈时要显式指明 `sign = 1`。

4.2 收获与体会

通过此次实验，我系统学习了使用 **MATLAB** 建立控制系统模型的方法。我熟练掌握了使用 `tf`、`zpk`、`ss` 函数建立控制系统的传递函数模型、零极点增益模型和状态空间模型的方法，同时学会了使用 **MATLAB** 进行这三种模型转换。与此同时，我也学会了使用 **MATLAB** 进行模型连接的等效变换的方法，也掌握了控制系统串联连接、并联连接和反馈连接的 **MATLAB** 实现。最后我还学到了利用 **MATLAB** 进行零极点配置的方法，根据闭环系统的零极点分布我们可以对系统的稳定性进行分析和判断。

在实验过程中，我进一步加深了对课上所学理论知识的理解，体会到了各种控制系统描述方式的特点及其适用场景，并学会了选择适当的控制系统模型并在需要时进行模型转换的方法。与此同时，我也进一步认识到各种复杂的模型都是由若干简单模型连接而成，进行适当的设计和组合我们可以实现任意复杂的控制系统，在设计时要特别注意复杂控制系统的连接方式。我在实验时因搞错正反馈和负反馈导致最终传递函数计算错误，这也提醒我在实验中要更加注意这一方面的细节。本次实验也增强了我的编程能力以及分析、解决问题的能力，同时我也学会了在设计时不断调整和修改控制系统的方法，这对于我之后的课程学习有很大帮助。

实验 2 控制系统的暂态特性分析

一、实验目的

1. 学习和掌握利用 MATLAB 进行系统时域响应求解和仿真的方法。
2. 考察二阶系统的时间响应，研究二阶系统参数对系统暂态特性的影响。

二、实验原理

2.1 系统的暂态性能指标

控制系统的暂态性能指标常以一组时域量值的形式给出，这些指标通常根据系统的单位阶跃响应给出定义，指标分别为：延迟时间 t_d 、上升时间 t_r 、峰值时间 t_p 、调整时间 t_s 、超调量 σ 。

2.2 LTI Viewer 工具

在 MATLAB 中提供了线性时不变系统仿真的工具 LTI Viewer，可以方便地观察系统的响应曲线和性能指标。在命令窗口中键入 `litview` 即可启动 LTI Viewer。

LTI Viewer 工具包含【File】菜单、【Edit】菜单、右键菜单等三部分。

三、实验内容

3.1 第 1 题

1. 实现代码

```
1. % 1. 定义开环传递函数
2. num = [8];
3. den = [1, 2, 0];
4. G = tf(num, den);
5.
6. % 2. 构建单位反馈闭环系统
7. T = feedback(G, 1, -1);
8.
9. % 3. 绘制阶跃响应曲线
10. step(T);
11. title('闭环系统阶跃响应');
12. xlabel('时间(秒)');
13. ylabel('幅值');
14. grid on;
```

2. 实验结果

系统的单位阶跃响应如图 2-1 所示：

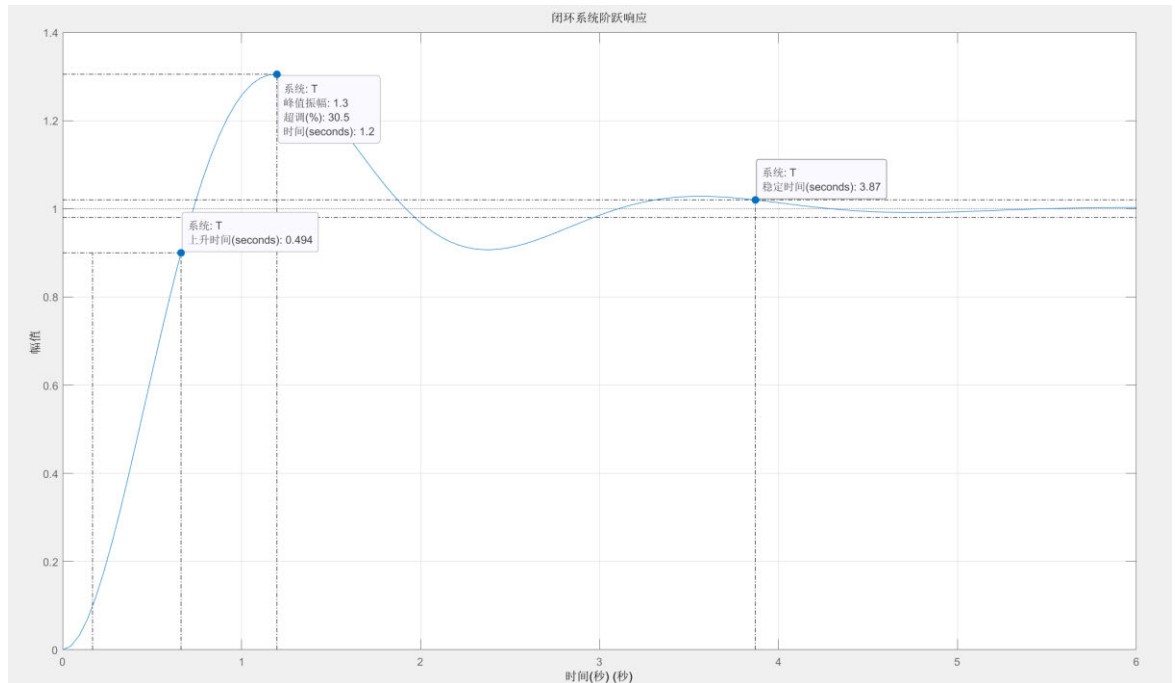


图 2-1：单位阶跃响应

- ✧ 上升时间：0.494s
- ✧ 峰值时间：1.2s，超调量：30.5%
- ✧ 稳定时间：3.87s

3.2 第 2 题

1. 实现代码

```
1. wn1 = 6; % 设定自然频率为 6 rad/s
2. ks1 = [0.1, 0.2, 0.707, 1.0, 2.0]; % 不同阻尼系数数组
3.
4. figure(1)
5. hold on
6.
7. % 循环生成不同阻尼系数的二阶系统
8. for i = 1:5
9.     num1 = wn1^2;
10.    den1 = [1, 2*wn1*ks1(i), wn1^2];
11.    G(i) = tf(num1, den1);
12.    step(G(i))
13.    title('固定 wn=6, 变化 ζ 时的阶跃响应')
14.    xlabel('时间')
```

```

15.     ylabel('幅值')
16.     legend('ζ=0.1', 'ζ=0.2', 'ζ=0.707', 'ζ=1.0', 'ζ=2.0', 'Location', 'best')
17.     grid on
18. end
19.     hold off
20.
21.
22. %% 第二部分：固定阻尼系数，变化自然频率
23. ks2 = 0.5; % 设定阻尼系数为 0.5
24. wn2 = [1, 3, 5]; % 不同自然频率数组
25.
26. figure(2)
27. hold on
28.
29. % 循环生成不同自然频率的二阶系统
30. for j = 1:3
31.     num2 = wn2(j)^2;
32.     den2 = [1, 2*wn2(j)*ks2, wn2(j)^2];
33.     T(j) = tf(num2, den2);
34.     step(T(j))
35.     title('固定 ζ=0.5, 变化 wn 时的阶跃响应')
36.     xlabel('时间')
37.     ylabel('幅值')
38.     legend('wn=1', 'wn=3', 'wn=5', 'Location', 'best')
39.     grid on
40. end
41. hold off
42.
43.
44. %% 第三部分：不同增益的三阶系统分析
45. k = [1.2, 2.25, 4]; % 不同增益值数组
46. den1 = [1, 0];
47. den2 = [0.5, 1];
48. den3 = [4, 1];
49.
50. % 创建各子系统传递函数
51. N1 = tf(1, den1);
52. N2 = tf(1, den2);
53. N3 = tf(1, den3);
54.
55. % 分析不同增益下的闭环系统
56. hold on

```

```

57. for n = 1:3
58.     figure(n+2)
59.     N = k(n) * N1 * N2 * N3;
60.     N = feedback(N, 1, -1);
61.
62.     % 根据增益值调整仿真时间
63.     if n == 2
64.         step(N, 0:0.1:50) % 中等增益使用更长时间仿真
65.     else
66.         step(N) % 默认仿真时间
67.     end
68.
69.     title(['增益 K=', num2str(k(n)), '时的闭环系统响应'])
70.     xlabel('时间')
71.     ylabel('幅值')
72.     grid on
73. end
74. hold off

```

2. 实验结果

当 $\omega_n=6$ 时, $\xi=0.1, 0.2, 0.707, 1.0, 2.0$ 的单位阶跃响应如图 2-2 所示:

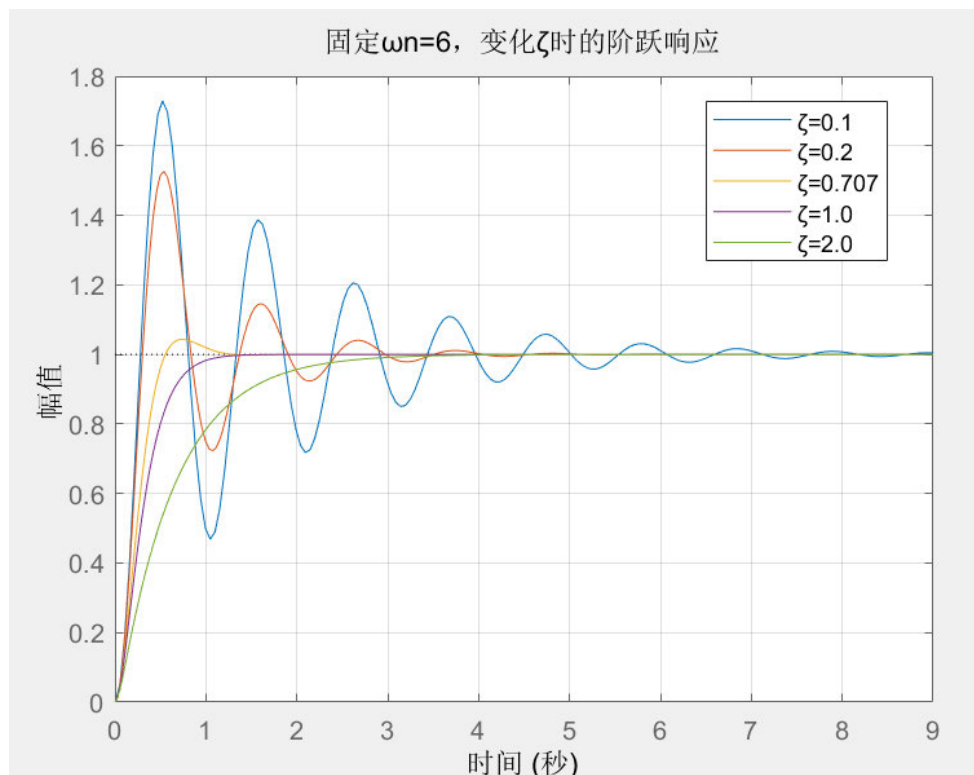


图 2-2: 单位阶跃响应

当 $\xi=0.5$ 时， $\omega_n=1,3,5$ 的单位阶跃响应如图 2-3 所示：

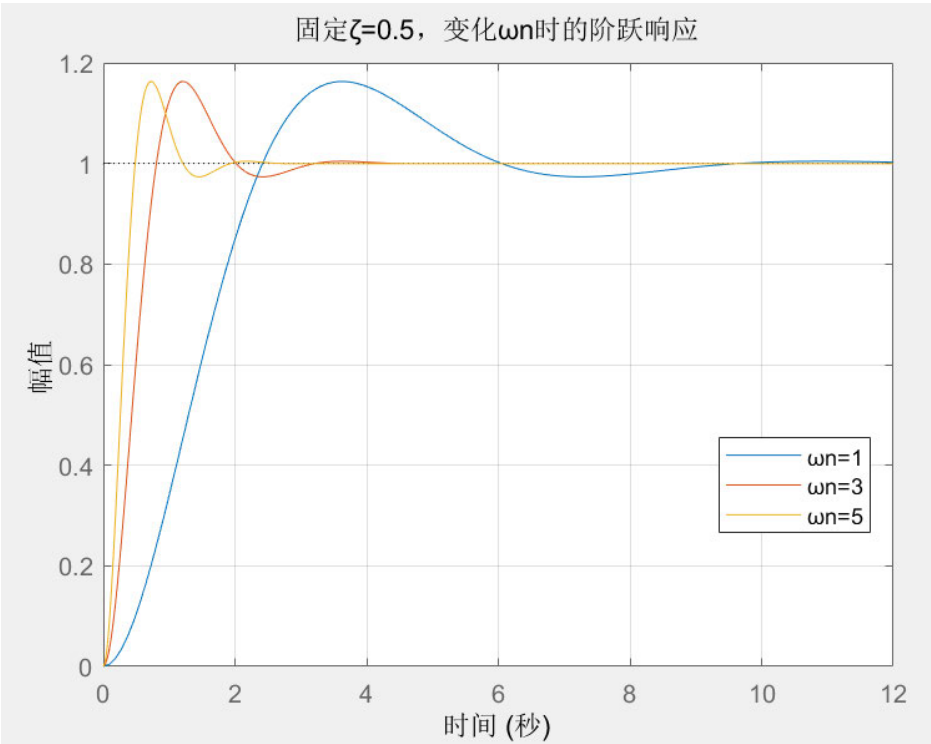


图 2-3：单位阶跃响应

由上述实验可以看出：当 ω_n 固定时， ξ 越大，超调量越小。当 $\xi \geq 1$ 时，系统处于过阻尼、临界阻尼状态，系统无超调。 ξ 越小，上升时间、峰值时间越短，系统响应速度越快。但当 ξ 越小，系统振荡次数越多、幅值越大，平稳性越差，因此实际设计二阶系统时要综合考虑。当 $\xi=0.707$ 为最佳阻尼比，系统调节时间最短。

当 ξ 固定时， ω_n 越大，系统响应速度越快，上升时间、峰值时间越短，系统超调量不会发生明显变化。

当系统开环增益 $K=1.2, 2.25, 4$ 时，单位阶跃响应如图 2-4~2-6 所示：

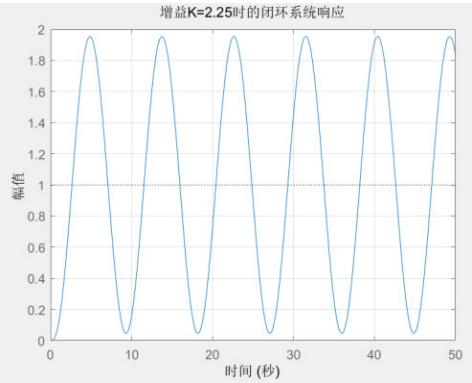


图 2-4：K =1.2 单位阶跃响应

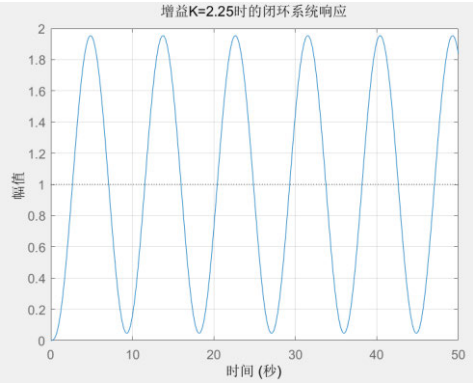


图 2-5：K =2.25 单位阶跃响应

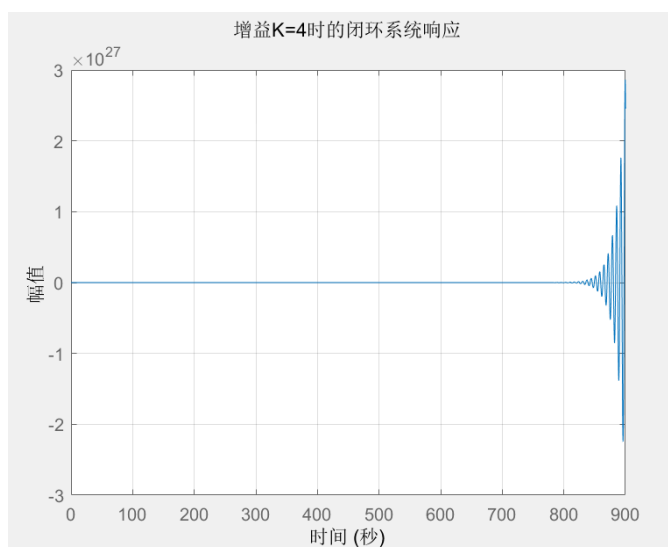


图 2-6: $K=4$ 单位阶跃响应

由上图可以看出，对于控制系统的稳定性，当 K 不断增大时，系统有可能会发生稳定→临界稳定→不稳定的现象。对于控制系统的暂态性能，当 K 较小时（如 $K=1.2$ ），系统响应速度慢，超调量较小甚至无超调。当 K 不断增大（如 $K=2.25$ ）快、超调可能上升。 K 过大若系统仍稳定，超调进一步增大、振荡加剧；若 K 超过临界值（ $K=4$ ），系统发散。

3.3 第3题

1. 实现代码

```
1. %% 初始系统
2. num = 1;
3. den = [5, 1, 0];
4. G = tf(num, den);
5.
6. %% 比例控制
7. figure(2);
8. G2 = 5 * G;
9. G2 = feedback(G2, 1, -1);
10.
11. % 绘制阶跃响应曲线(0-50 秒, 步长 0.01 秒)
12. step(G2, 0:0.01:50);
13. xlabel('时间');
14. ylabel('幅值');
15. grid on;
16.
17. %% 比例-微分控制
```



```

18. figure(3);
19. T = tf([4, 5], 1);
20. G3 = T * G;
21. G3 = feedback(G3, 1, -1);
22.
23. % 绘制阶跃响应曲线(0-50 秒, 步长 0.02 秒)
24. step(G3, 0:0.02:50);
25. xlabel('时间');
26. ylabel('幅值');
27. grid on;
28.
29. %% 速度负反馈控制
30. figure(4);
31. T1 = tf(5, [5, 1]);
32. T1 = feedback(T1, 0.8, -1);
33. T2 = tf(1, [1, 0]);
34. G4 = T1 * T2;
35. G4 = feedback(G4, 1, -1);
36.
37. % 绘制阶跃响应曲线(0-50 秒, 步长 0.02 秒)
38. step(G4, 0:0.02:50);
39. xlabel('时间(秒)');
40. ylabel('幅值');
41. grid on;

```

2. 实验结果

比例控制的单位阶跃响应如图 2-7 所示：

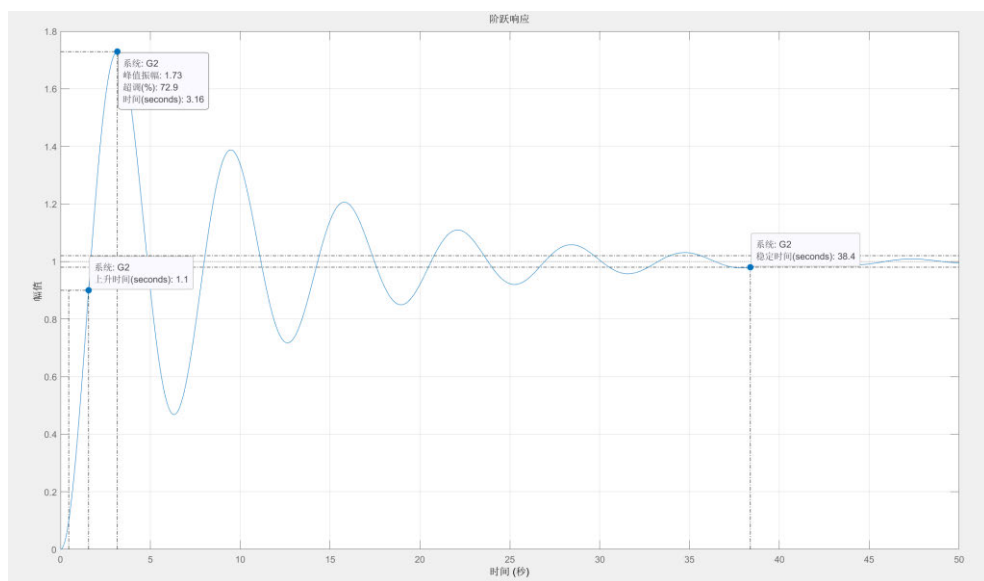


图 2-7：比例控制的单位阶跃响应

- ✧ 上升时间: 1.1s
- ✧ 峰值时间: 3.16s, 超调量: 72.9%
- ✧ 稳定时间: 38.4s

比例-微分控制的单位阶跃响应如图 2-8 所示:

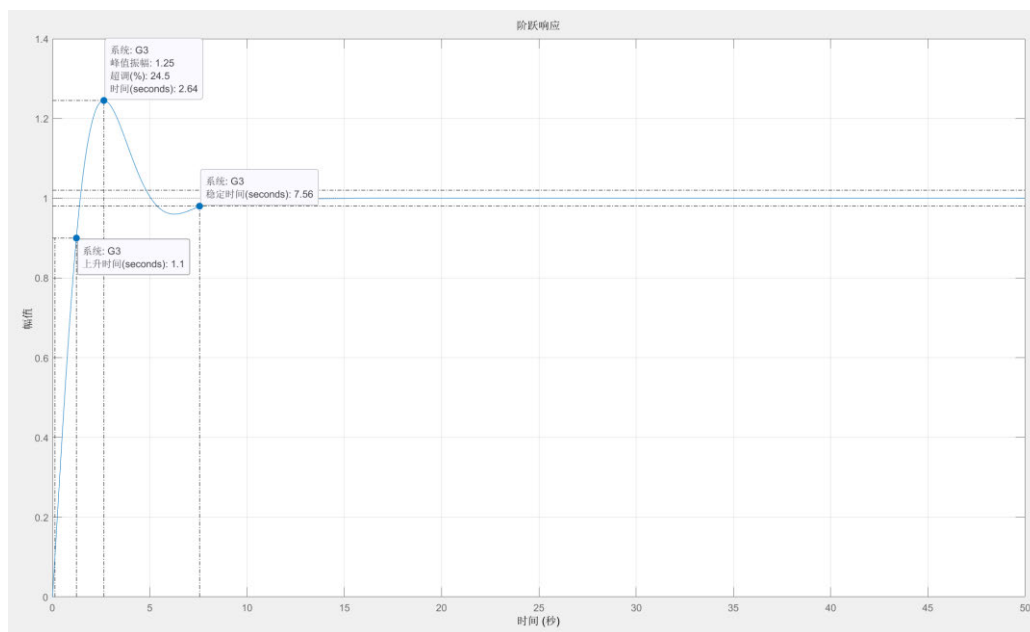


图 2-8: 比例-微分控制的单位阶跃响应

- ✧ 上升时间: 1.1s
- ✧ 峰值时间: 2.64s, 超调量: 24.5%
- ✧ 稳定时间: 7.56s

速度负反馈控制的单位阶跃响应如图 2-9 所示:

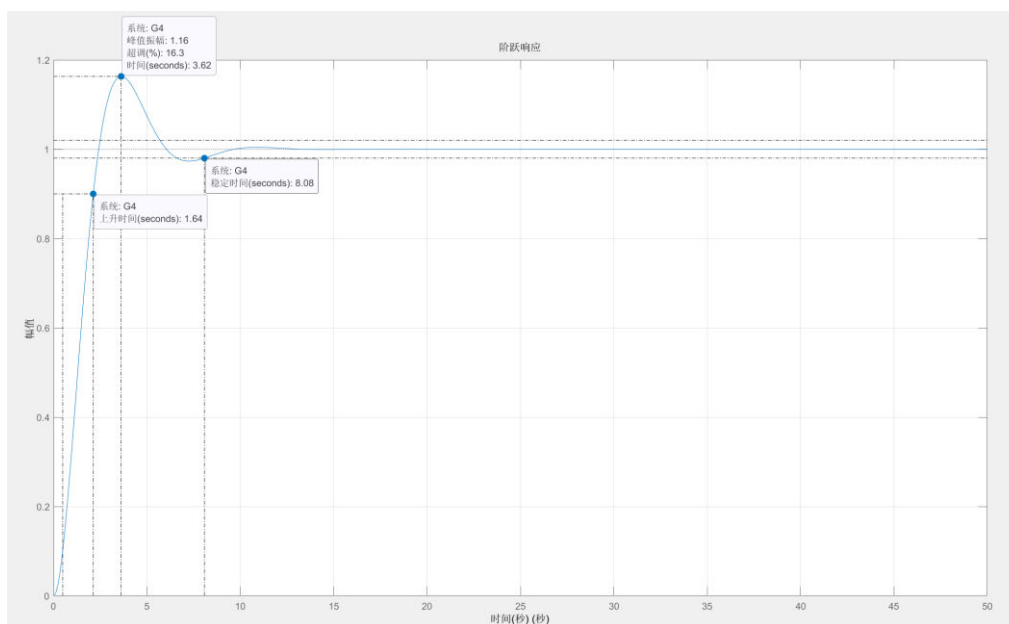


图 2-9：速度负反馈控制的单位阶跃响应

- ✧ 上升时间：1.64s
- ✧ 峰值时间：3.62s，超调量：24.9%
- ✧ 稳定时间：8.08s

由上述分析可得：

比例控制：当增大开环增益 K ，可以加快系统响应速度，但可能增大超调量，不利于系统的平稳性。

比例-微分控制：引入微分环节（0.8s），可以对系统的误差进行提前预判，从而减小超调量，加快系统响应速度，有利于提升系统的平稳性与快速性。

速度负反馈控制：通过反馈速度信号，增加系统阻尼，减小振荡剧烈程度，降低超调量。但同时可能会改变系统固有频率，影响响应速度，可使系统更加平稳。

四、实验总结

4.1 实验中遇到的问题及解决方法

问题 1：在绘制阶跃响应曲线时，曲线显式不完整。

解决方案：该问题的出现是由于没有设置 `step` 函数的时间范围，在系统默认的时间间隔内控制系统的响应尚未达到稳定，因此无法绘制出完整的曲线。解决方法是根据系统参数估算调整时间，对 `step` 函数的时间范围和时间间隔进行手动设置，让响应曲线完整呈现稳态过程。

问题 2：混淆上升时间、峰值时间、稳定时间等相关概念，

解决方法：认真理清相关概念的内涵和联系，并通过 MATLAB 进行作图求解。

4.2 收获与体会

通过实验二，我对控制系统的时域响应有了更深刻的认识。在实验中我学会了使用 MATLAB 绘制阶跃响应曲线，并通过 LTI Viewer 工具求解系统的延迟时间 t_d 、上升时间 t_r 、峰值时间 t_p 、调整时间 t_s 、超调量 σ 等暂态性能指标。在这里我更直观地认识到这些物理量的深刻含义与相互之间的关系，进一步加深了对课上所学理论知识的理解。与此同时，通过实验中分别固定阻尼比和自然频率来研究其中一个因素对系统响应的影响，我亲自验证了课本上的“阻尼比越大，系统稳定性越差”、“自然频率越大，响应速度越快”等相关结论。最后，通过在控制系统中引入比例控制、比例-微分控制和速度负反馈控制，我深刻认识到这些控制方式在改善二阶欠阻尼系统性能中所起到的重要作用，为控制系统的设计和性能改善提供了具体可行的指导。

在这次实验中，我不仅学到了很多控制原理方面的理论知识，也提高了我利用 MATLAB 分析和设计控制系统的能力，做到了理论和实践相融合，这些知识和经验将在我以后的学习科研中发挥重要作用。

实验 3 根轨迹分析

一、实验目的

1. 学习和掌握利用 MATLAB 绘制根轨迹图的方法。
2. 学习和掌握利用系统根轨迹图分析系统的性能。

二、实验原理

3.1 根轨迹分析的 MATLAB 实现

根轨迹是指系统某一参数变化时，闭环特征根在 s 平面上运动的轨迹。在 MATLAB 中，提供了用于根轨迹分析的专门函数，这些函数包括 `rlocus` 函数、`rlocfind` 函数、`sgrid` 函数等。

3.2 Rltool 工具

MATLAB 提供了一个根轨迹设计工具 `Rltool`，在命令窗口输入 `rltool` 命令即可启动该工具，也可输入 `rltool(sys)` 命令打开带系统模型 `sys` 的根轨迹设计器。`Rltool` 工具既可以分析系统根轨迹也能对系统进行设计。

三、实验内容

3.1 第 1 题

1. 实现代码

```
1. % 定义系统零点位置
2. z = [-5]; % 系统零点在  $s = -5$  处
3.
4. % 定义三种不同的极点配置
5. p1 = [-1 -3 -12];
6. p2 = [-1 -3 -35];
7. p3 = [-1 -3 -7];
8.
9. % 创建三种零极点形式的传递函数
10. G1 = zpk(z, p1, 1);
11. G2 = zpk(z, p2, 1);
12. G3 = zpk(z, p3, 1);
13.
14. % 绘制第一种系统的根轨迹图
15. figure(1)
16. rlocus(G1)
```

```

17. title('根轨迹图 (极点: -1, -3, -12)')
18. xlabel('实轴')
19. ylabel('虚轴')
20. grid on
21.
22. % 绘制第二种系统的根轨迹图
23. figure(2)
24. rlocus(G2)
25. title('根轨迹图 (极点: -1, -3, -35)')
26. xlabel('实轴')
27. ylabel('虚轴')
28. grid on
29.
30. % 绘制第三种系统的根轨迹图
31. figure(3)
32. rlocus(G3)
33. title('根轨迹图 (极点: -1, -3, -7)')
34. xlabel('实轴')
35. ylabel('虚轴')
36. grid on

```

2. 实验结果

该系统的根轨迹图，如图 3-1 所示：

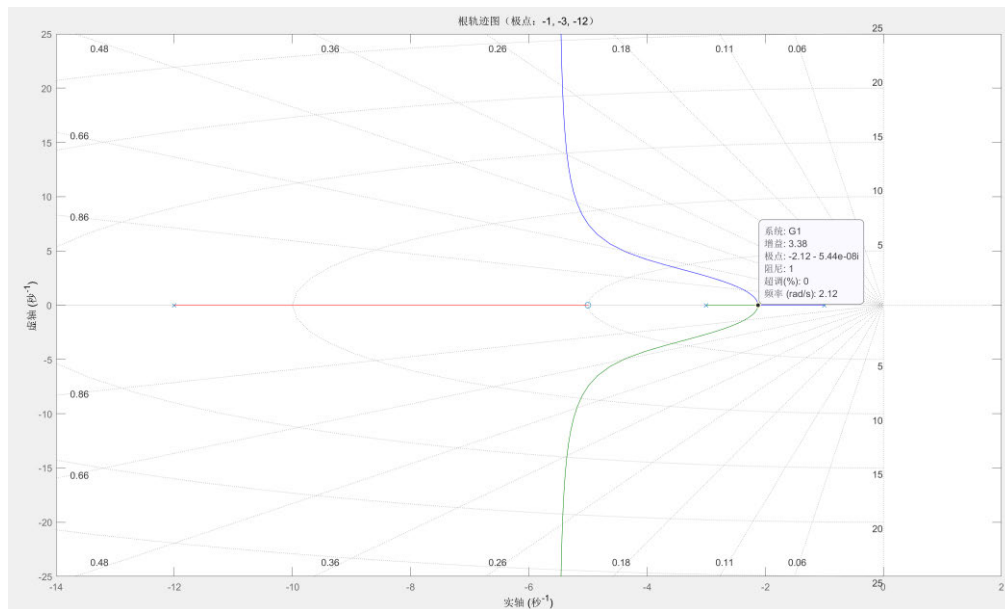


图 3-1：根轨迹图

由上图可知，根轨迹的两条分支离开实轴时 $K=3.38$ ，对应的所有闭环极点为 -2.12 。

以-35、-7 替代 $s=-12$ 处的极点，重新绘制根轨迹图，如图 3-2、3-3 所示：

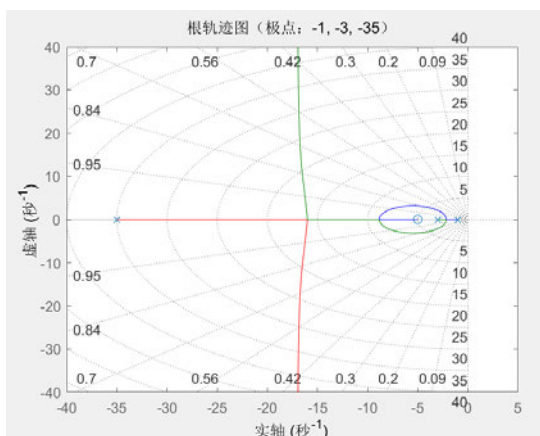


图 3-2: s 替换为-35 根轨迹图

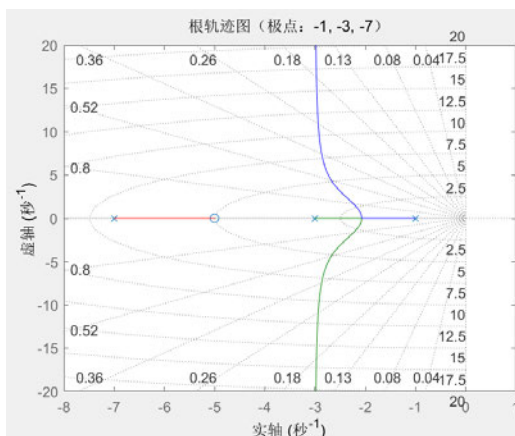


图 3-3: s 替换为-7 根轨迹图

观察以上根轨迹图可知：

当 $s=-12$ 变为 $s=-35$ 时，渐近线与实轴的交点变为-17，夹角仍保持 $\pm 90^\circ$ 不变，与原先相比根轨迹发生了整体左移。与此同时，根轨迹在实轴多了一对分离点和会合点，从而导致一段闭合曲线的形成。极点左移也使得达到临界稳定时 K 值增大，在一定程度上改善了系统的稳定性。

当 $s=-12$ 变为 $s=-7$ 时，渐近线与实轴的交点变为-3，夹角仍保持 $\pm 90^\circ$ 不变，与原先相比根轨迹发生了整体右移，并且实轴上分离点和汇合点的情况基本保持不变。极点右移也使得达到临界稳定时 K 值减小，导致系统的稳定性下降。

3.2 第 2 题

1. 实现代码

```
2. % 定义三种不同的零点位置
3. z1 = -8;
4. z2 = -18;
5. z3 = -3;
6.
7. % 定义共同的极点配置（包含实数极点和共轭复数极点）
8. p = [0 -2 -4+4i -4-4i]; % 极点在  $s = 0, -2, -4\pm 4i$ 
9.
10. % 创建三种零极点形式的传递函数（增益均为 1）
11. G1 = zpk(z1, p, 1);
12. G2 = zpk(z2, p, 1);
13. G3 = zpk(z3, p, 1);
14.
```

```

15. % 绘制第一种系统的根轨迹图
16. figure(1)
17. rlocus(G1) % 绘制 G1 的根轨迹
18. title('根轨迹图 (零点: -8, 极点: 0, -2, -4±4i)')
19. xlabel('实轴 (Re)')
20. ylabel('虚轴 (Im)')
21. grid on
22.
23. % 绘制第二种系统的根轨迹图
24. figure(2)
25. rlocus(G2)
26. title('根轨迹图 (零点: -18, 极点: 0, -2, -4±4i)')
27. xlabel('实轴 (Re)')
28. ylabel('虚轴 (Im)')
29. grid on
30.
31. % 绘制第三种系统的根轨迹图
32. figure(3)
33. rlocus(G3)
34. title('根轨迹图 (零点: -3, 极点: 0, -2, -4±4i)')
35. xlabel('实轴 (Re)')
36. ylabel('虚轴 (Im)')
37. grid on

```

2. 实验结果

该系统的根轨迹图，如图 3-4 所示：

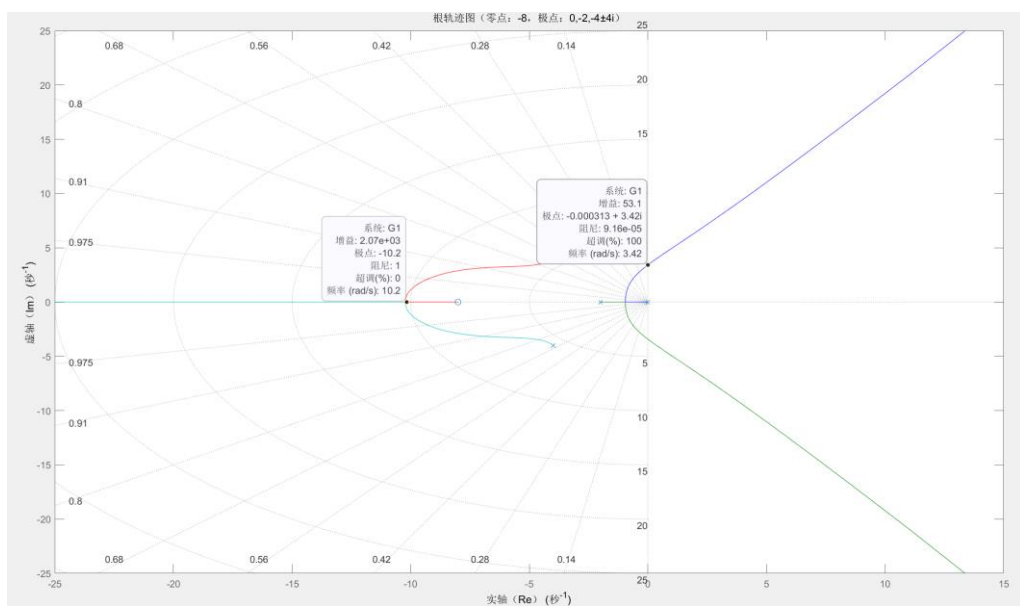


图 3-4: 根轨迹图

由上图可知，根轨迹的两条分支进入右半平面 $K=53.1$ ，两条分支复数极点出发在实轴相交处 $K=2070$ 。

以-18、-3 替代零点的位置，重新绘制根轨迹图，如图 3-5、3-6 所示：

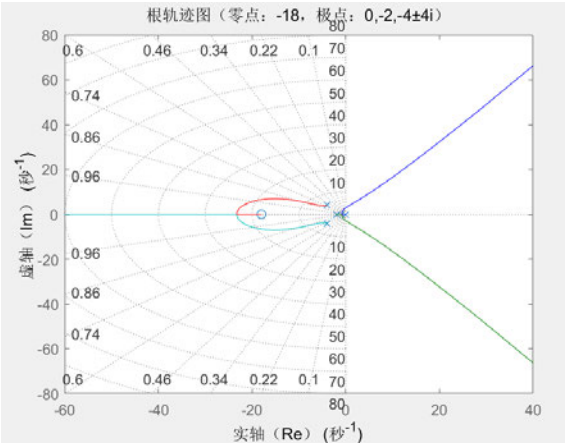


图 3-5: $s=-18$ 根轨迹图

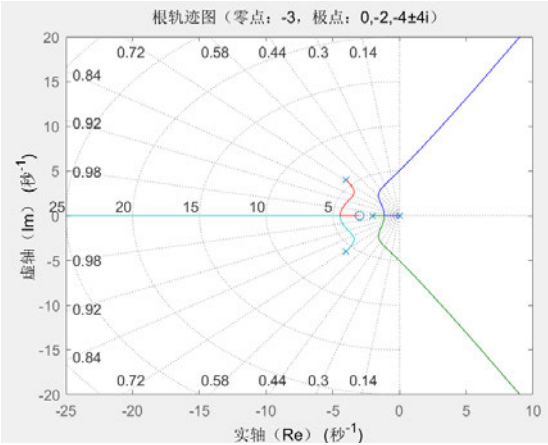


图 3-6: $s=-3$ 根轨迹图

观察以上根轨迹图可知：

当 $s=-8$ 变为 $s=-35$ 时，渐近线与实轴的交点变为 2.67，夹角仍保持 $\pm 60^\circ$ 不变，与原先相比根轨迹发生了整体右移。并且实轴上分离点和汇合点的情况基本保持不变。零点左移也使得达到临界稳定时 K 值减小，可能会导致系统的稳定性下降。

当 $s=-8$ 变为 $s=-3$ 时，渐近线与实轴的交点变为-3，夹角仍保持 $\pm 90^\circ$ 不变，与原先相比根轨迹发生了整体右移，并且实轴上分离点和汇合点的情况基本保持不变。但是由于零点距离虚轴较近，导致根轨迹出现了一定的波动。零点的右移使得临界稳定时 K 值增大，在一定程度上可以改善系统的稳定性。

3.3 第3题

1. 实现代码

```
2. %% (1) 绘制系统的根轨迹图
3. % 定义开环传递函数  $G(s) = K/[s(s^2 + 4s + 5)]$ 
4. num = 1;
5. den = conv([1 0], [1 4 5]);
6.
7. G = tf(num, den);
8.
9. % 绘制根轨迹图
10. figure(1)
11. rlocus(G)
```

```

12. title('系统根轨迹图  $G(s)=K/[s(s^2+4s+5)]$ ')
13. grid on
14.
15. %% 验证不同 K 值的阶跃响应
16. K_values = [10 20 30]; % 分别选择稳定、临界稳定和不稳定的 K 值
17.
18. for i = 1:length(K_values)
19.     K = K_values(i);
20.     G_closed = feedback(K*G, 1);
21.
22.     figure(i+1)
23.     step(G_closed, 10) % 仿真 10 秒
24.     title(sprintf('阶跃响应 (K=%d)', K))
25.     grid on
26.
27. end
28. %% 专门验证无超调情况
29. for K_no_overshoot = 5:6
30.     figure(K_no_overshoot)
31.     G_closed_no_overshoot = feedback(K_no_overshoot*G, 1);
32.
33.     step(G_closed_no_overshoot, 10)
34.     title(sprintf('无超调验证 (K=%1f)', K_no_overshoot))
35.     grid on
36. end

```

2. 实验结果

系统的根轨迹图，如图 3-7 所示：

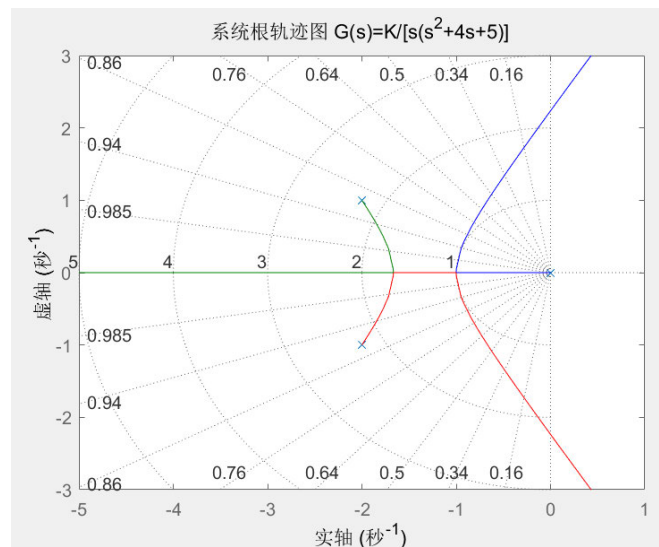


图 3-7：根轨迹图

使用劳斯判据判断系统稳定性，过程如下：

计算闭环特征方程： $1 + G(s) = 0 \rightarrow s^3 + 4s^2 + 5s + K = 0$

使用 Routh 判据分析稳定性：

$s^3 \mid 1 \quad 5$

$s^2 \mid 4 \quad K$

$s^1 \mid (20-K)/4$

$s^0 \mid K$

稳定条件：

1. 所有系数为正 $\rightarrow K > 0$
2. 第一列全为正 $\rightarrow K < 20$

因此稳定范围： $0 < K < 20$

分别画出 $K=10, 20, 30$ 的阶跃响应曲线，如图 3-8, 3-9, 3-10 所示：

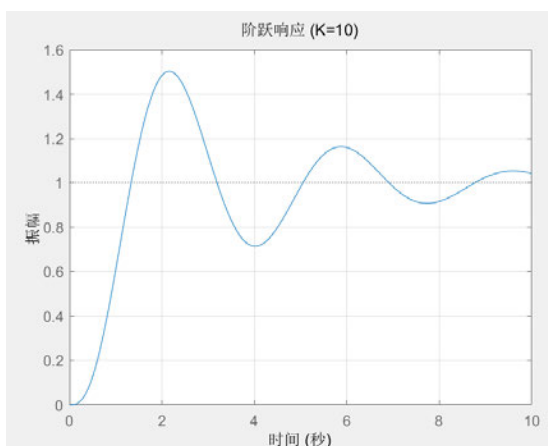


图 3-8：K=10 单位阶跃响应曲线

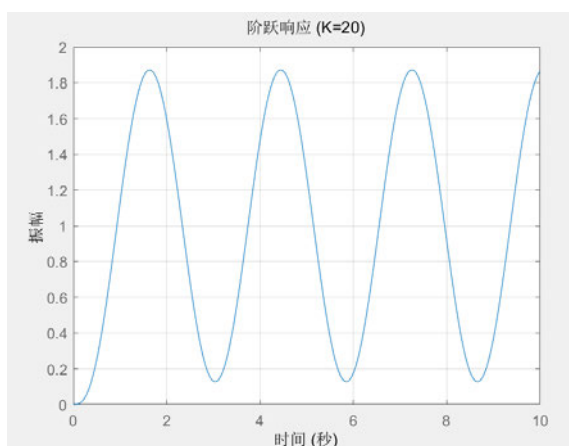


图 3-9：K=20 单位阶跃响应曲线

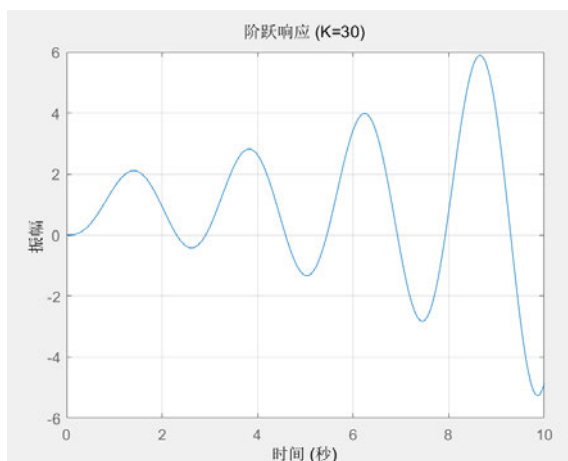


图 3-10：K=30 单位阶跃响应曲线

接下来对系统超调情况进行分析。无超调条件：所有极点均为实数且位于负实轴，通过根轨迹观察可得，当主导极点位于实轴时无超调。通过阻尼比 $\xi \geq 1$ ，得到系统无超调的 K 取值范围： $0 < K < 5$ 。画出 $K=5$ 和 $K=6$ 的阶跃响应曲线，如图 3-11,3-12 所示：

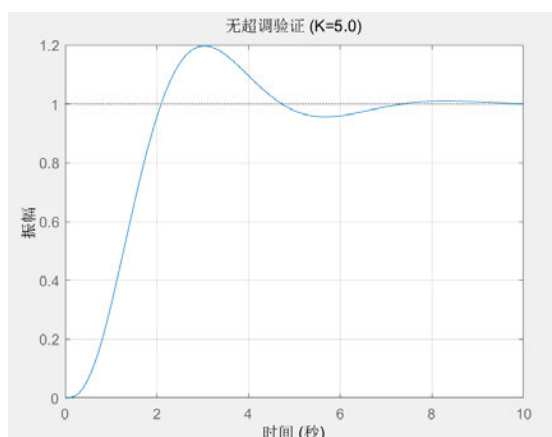


图 3-11: $K=5$ 单位阶跃响应曲线

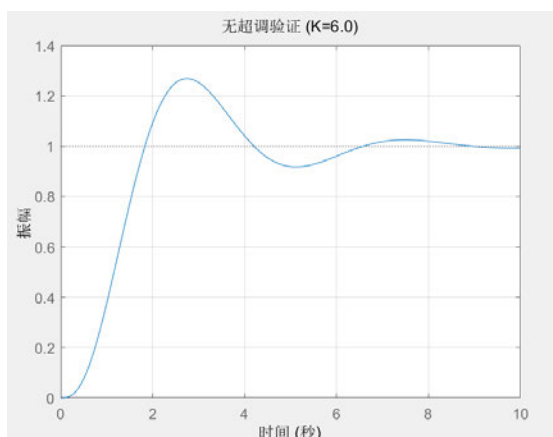


图 3-12: $K=6$ 单位阶跃响应曲线

由上图可以看出， $K=5$ 系统单位阶跃响应峰值不超过 1.2，无超调现象； $K=6$ 系统单位阶跃响应峰值超过 1.2，有超调。

四、实验总结

4.1 实验中遇到的问题及解决方法

问题 1：根轨迹形态与理论分析不符

解决方案：通过理论分析判断出根轨迹应当具有的形态，检查开环传递函数的求解是否正确，为了便于分析可采用零极点增益模型构建控制系统。

问题 2：开环增益计算出错

解决方法：明确相关概念，在根轨迹图上认真寻找对应点即可。

4.2 收获与体会

在这次的实验中，我学会了使用 MATLAB 进行根轨迹绘制的方法。通过实验中对根轨迹形态的理论分析，并且使用 MATLAB 工具对其进行绘制，我进一步加深了对根轨迹的形态特点和物理意义的认识，也提高了我利用 MATLAB 进行控制系统设计的能力。在这次实验中，尽管在绘制根轨迹的过程中出现了形态异常的问题，但我充分利用课上所学的理论知识，对根轨迹应有的形态进行分析判断，并逐步调整控制系统的相关参数，绘制出形态正确的根轨迹。这种理论与实践相结合，

分析问题解决问题的方法与能力是我这次实验中最大的收获。最后，我还利用劳斯定律对系统稳定性进行分析判断，并求解系统无超调开环增益取值范围，这使我进一步加深了对控制系统稳定性等相关知识的理解，也使我意识到这些方法对于控制系统分析与设计中所起到的重大作用。这次实验的收获也必将会为我之后的专业学习带来带来很大的一处，也为我之后进一步学习控制系统相关知识打下了基础。

实验 4 系统的频率特性分析

一、实验目的

1. 学习和掌握利用 MATLAB 绘制系统 Nyquist 图和 Bode 图的方法。
2. 学习和掌握利用系统的频率特性分析系统的性能。

二、实验原理

系统的频率特性是一种图解方法，运用系统的开环频率特性曲线，分析闭环系统的性能，如系统的稳态性能、暂态性能。常用的频率特性曲线有 Nyquist 图和 Bode 图。在 MATLAB 中，提供了绘制 Nyquist 图和 Bode 图的专门函数。其中 nyquist 函数可以用于计算或绘制连续时间 LTI 系统的 Nyquist 频率曲线，bode 函数可以用于计算或绘制连续时间 LTI 系统的 Bode 图。margin 函数可以用于从频率响应数据中计算出幅值裕度、相位裕度及其对应的角频率。

三、实验内容

3.1 第 1 题

1. 实现代码

```
1. % 定义传递函数的分子多项式
2. num = 1000;
3.
4. % 定义第一个传递函数的分母多项式
5. den1 = [1, 3, 2];
6. % 定义第二个传递函数的分母多项式
7. den2 = [1, 5];
8.
9. % 创建两个传递函数对象
10. G1 = tf(num, den1);
11. G2 = tf(1, den2);
12.
13. % 计算串联系统的总传递函数（两个传递函数相乘）
14. G = G1 * G2;
15.
16. % 绘制 Nyquist 图（频域分析）
17. nyquist(G)
18. title('系统 Nyquist 图  $G(s)=1000/[(s^2+3s+2)(s+5)]'$ ')
19. grid on
```

2. 实验结果

系统的 Nyquist 图，如图 4-1 所示：

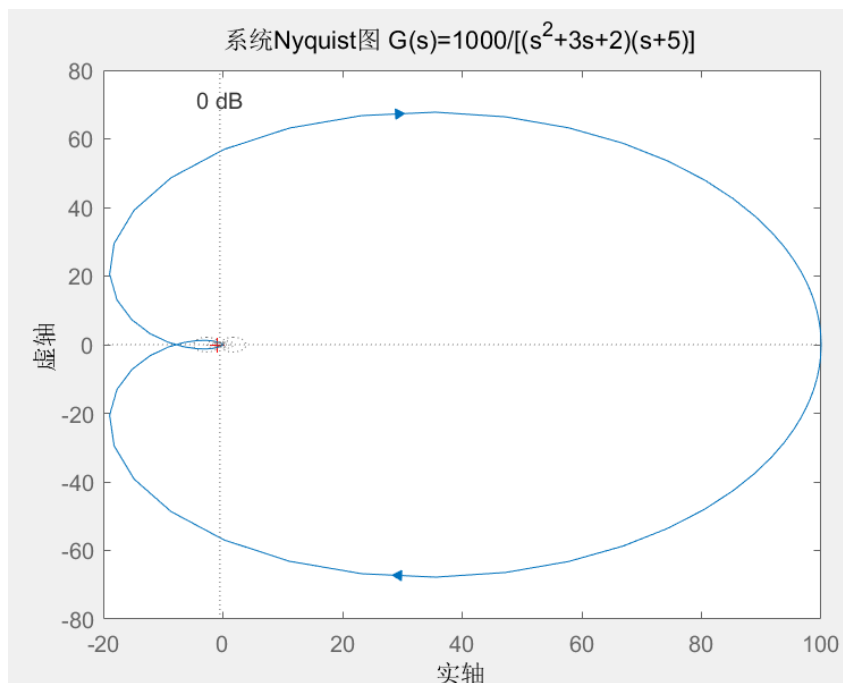


图 4-1: Nyquist 图

由上图可知，系统没有位于右边平面的开环极点，并且 Nyquist 曲线未包围 $(-1, j0)$ 点，所以系统闭环稳定。

3.2 第 2 题

1. 实现代码

```
1. % 定义第一个传递函数（比例+微分环节）
2. G1 = tf([100, 400], 1);
3.
4. % 定义第二个传递函数（积分环节）
5. G2 = tf(1, [1, 0]);
6.
7. % 定义第三个传递函数（一阶惯性环节）
8. G3 = tf(1, [1, 0.5]);
9.
10. % 定义第四个传递函数（二阶振荡环节）
11. G4 = tf(1, [1, 100, 2500]);
12.
13. % 计算串联系统的开环传递函数（各环节相乘）
14. G = G1 * G2 * G3 * G4;
15.
```

```
16. % 绘制 Bode 图并显示幅值裕度和相位裕度
```

```
17. margin(G)
```

```
18. grid on
```

2. 实验结果

该系统的 Bode 图，如图 4-2 所示：

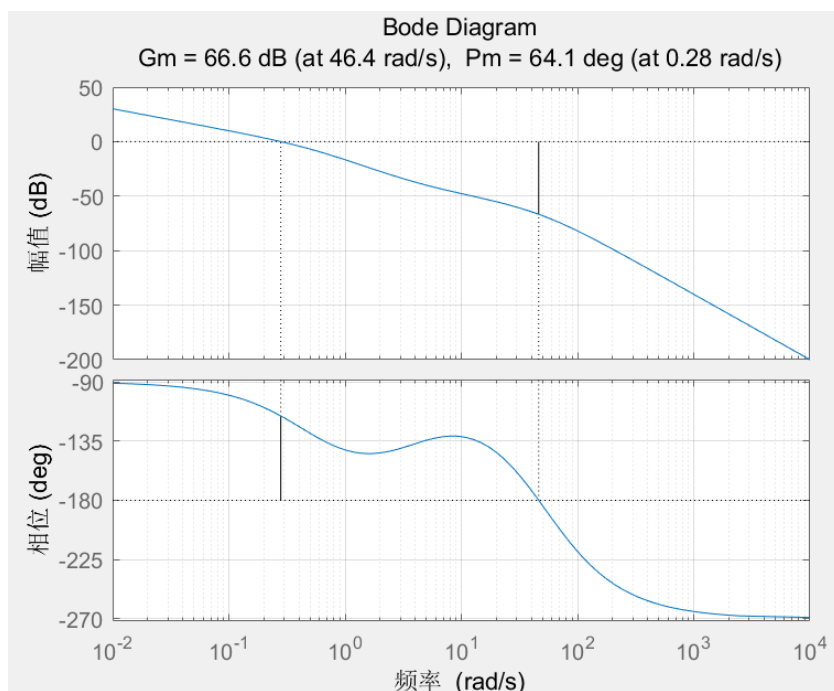


图 4-2: Bode 图

由上图可以看出，该系统幅值裕度为：66.6dB；相位裕度为：64.1°，由于系统的幅值裕度和相位裕度均大于零，因此该闭环系统稳定。

3.3 第 3 题

1. 实现代码

```
2. % 定义两个不同的分子系数（系统增益）
```

```
3. num1 = 5;
```

```
4. num2 = 20;
```

```
5.
```

```
6. % 定义共同的分母多项式（通过 conv 函数计算多项式乘积）
```

```
7. den = conv([1 1 0 0], [0.1 1]);
```

```
8.
```

```
9. % 创建两个传递函数对象（不同增益，相同极点配置）
```

```
10. G1 = tf(num1, den);
```

```
11. G2 = tf(num2, den);
```

```
12.
```

```
13. % 绘制第一个系统的 Bode 图（含幅值/相位裕度）
```



```

14. figure(1);
15. margin(G1);
16. grid on;
17.
18. % 绘制第二个系统的 Bode 图（含幅值/相位裕度）
19. figure(2);
20. margin(G2);
21. grid on;

```

2. 实验结果

K=5 和 20 的 Bode 图，如图 4-3、4-4 所示：

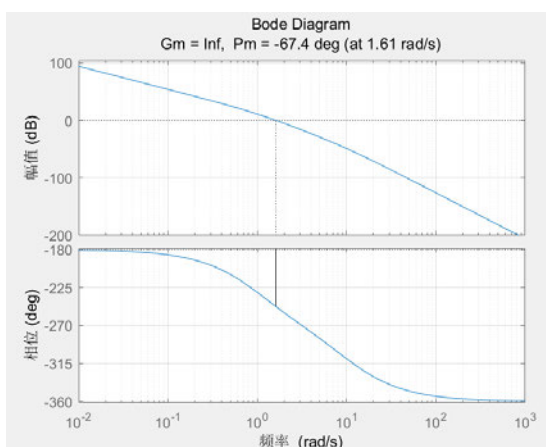


图 4-3: K=5 Bode 图

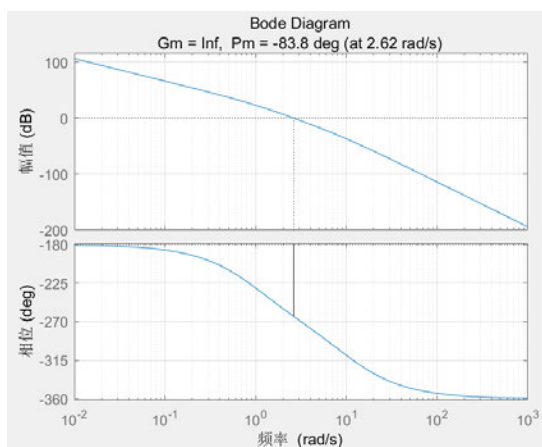


图 4-4: K=20 Bode 图

K=5 系统幅值裕度为 $-\infty$ ；相位裕度为： -67.4° 。K=20 系统幅值裕度为： $-\infty$ ；相位裕度为： -83.8° 。系统幅值裕度和相位裕度均小于零，故闭环系统不稳定。

3.4 第 4 题

1. 实现代码

```

1. % 系统参数
2. K = 1; % 初始增益
3. num = K;
4. den = [1, 2, 4]; % s^2 + 2s + 4
5.
6. % 1. 满足稳态误差要求
7. K_req = 36;
8. sys_uncorrected = tf(K_req, den);
9.
10. % 2. 检查未校正系统的相位裕度
11. [Gm_uncorr, Pm_uncorr] = margin(sys_uncorrected);
12. fprintf('未校正系统相位裕度: %.2f°\n', Pm_uncorr);
13.

```

```

14. % 3. 设计超前校正器（满足相位裕度 45°）
15. % 目标相位裕度
16. Pm_desired = 45;
17.
18. % 计算需要增加的相位超前量
19. phi_m = (Pm_desired - Pm_uncorr) + 10; % 增加 10°补偿量
20. phi_m_rad = deg2rad(phi_m);
21.
22. % 计算超前校正器的参数
23. alpha = (1 + sin(phi_m_rad))/(1 - sin(phi_m_rad));
24.
25. % 找到未校正系统增益为-10log10(alpha)的频率
26. [mag, phase, w] = bode(sys_uncorrected);
27. mag_db = 20*log10(squeeze(mag));
28. gain_target = -10*log10(alpha);
29. w_c = interp1(mag_db, w, gain_target, 'spline');
30.
31. % 计算时间常数
32. T = 1/(w_c*sqrt(alpha));
33.
34. % 创建超前校正器
35. num_lead = [alpha*T, 1];
36. den_lead = [T, 1];
37. lead_compensator = tf(num_lead, den_lead);
38.
39. % 4. 校正后系统
40. sys_corrected = series(lead_compensator, sys_uncorrected);
41.
42. % 5. 验证校正后系统性能
43. [Gm_corr, Pm_corr] = margin(sys_corrected);
44. fprintf('校正后系统相位裕度: %.2f°\n', Pm_corr);
45.
46. % 6. 绘制 Bode 图
47. figure('Position', [100, 100, 1000, 800]);
48.
49. % 未校正系统 Bode 图
50. subplot(2,1,1);
51. hold on
52. margin(sys_uncorrected);
53. grid on;
54.
55. % 校正后系统 Bode 图

```

```

56. margin(sys_corrected);
57. grid on;
58. title('Bode 图');
59. legend('未校正系统', '校正后系统');
60. hold off
61.
62. % 7. 绘制单位阶跃响应
63. subplot(2,1,2);
64. hold on;
65.
66. % 未校正系统闭环阶跃响应
67. sys_cl_uncorr = feedback(sys_uncorrected, 1);
68. step(sys_cl_uncorr);
69.
70. % 校正后系统闭环阶跃响应
71. sys_cl_corr = feedback(sys_corrected, 1);
72. step(sys_cl_corr);
73.
74. grid on;
75. title('单位阶跃响应');
76. legend('未校正系统', '校正后系统');
77. xlim([0, 5]);
78.
79. % 8. 显示控制器参数
80. fprintf('\n 超前校正器设计:\n');
81. fprintf('传递函数: Gc(s) = %.4f * (%.4fs + 1)/(%.4fs + 1)\n', ...
82.         alpha, alpha*T, T);
83. fprintf('参数: alpha = %.4f, T = %.4f\n', alpha, T);
84.
85. % 9. 分析校正效果
86. fprintf('相位裕度从 %.2f° 提高到 %.2f°\n', Pm_uncorr, Pm_corr);

```

2. 实验结果

首先进行控制器的设计。由稳态误差的要求可求得 $K = 36$ ，根据校正前后的相位裕度并且增加 10° 补偿量，可求得 $\alpha = 3.6931$ ，将未校正系统增益为 $-10\log_{10}(\alpha)$ 的频率作为校正后系统的幅值穿越频率，可求得 $\omega_c = 8.4311$ ， $T = 0.0617$ ，最终可得控制器的传递函数为 $G_c(s) = 3.6931 \cdot (0.2279s + 1) / (0.0617s + 1)$ 。

超前校正前后系统的 Bode 图和系统的单位阶跃响应如图 4-5 所示：

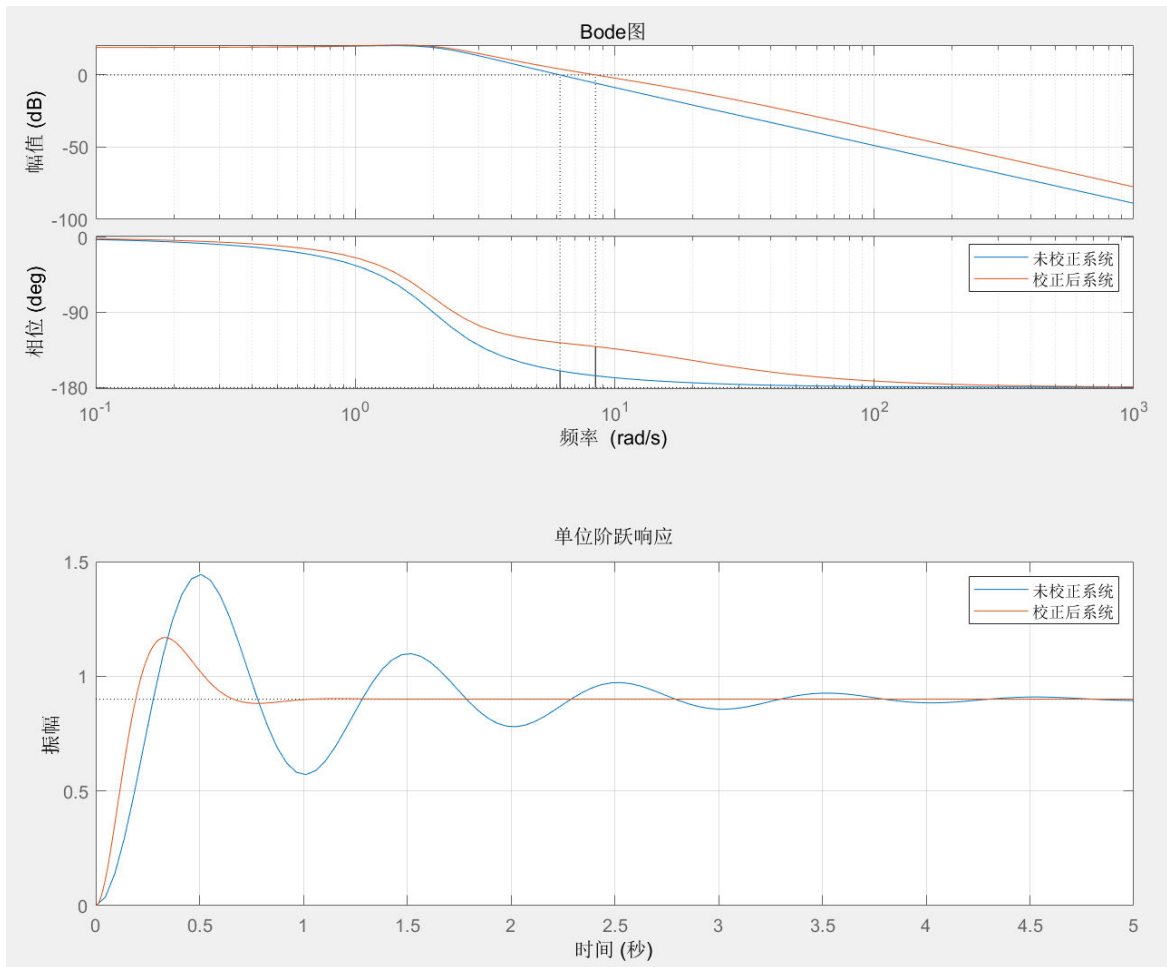


图 4-5: 系统的 Bode 图和单位阶跃响应曲线

通过验证，系统的相位裕度从 19.98° 提高到 49.13° ，稳态误差保持不变，符合设计的要求。

由上图可知，系统经过超前校正后，相位裕度显著提高，系统稳定性增强。系统的幅值穿越频率增大，阶跃响应曲线更加平稳，调节时间缩短，可以显著改善控制系统的性能。

四、实验总结

4.1 实验中遇到的问题及解决方法

问题：超前校正系统参数设计不合理。

解决方案：认真对照超前校正系统的设计方法进行设计，注意选择合适的补偿相角，在设计完成之后要进行相位裕度的验证，判断该超前校正系统是否满足设计要求。

4.2 收获与体会

在这次实验中，我进一步掌握了控制系统的频率设计分析方法，学会了使用 MATLAB 绘制系统 Nyquist 图和 Bode 图的方法。在实验中我通过对控制系统 Nyquist 图的绘制，更深刻地理解了使用开环传递函数的 Nyquist 图进行闭环系统稳定性的判断的方法。通过对 Bode 图的绘制，我也进一步体会到对数幅频特性和相频特性对于分析控制系统频域性能指标带来的极大便利，也熟练掌握了幅值裕度和相位裕度的计算并通过其判断闭环系统稳定性的方法。最后我通过 MATLAB 进行超前校正网络的设计，也深刻体会到实际应用中控制系统的复杂性以及进行校正的必要性，学会了使用基于 Bode 图频率特性进行超前校正网络设计的方法，也看到了校正网络在提高系统稳定性方面所发挥的重大作用。在这次实验中我继续将理论与实际相结合，不断提升自己分析设计复杂控制系统的能力，为以后的专业学习打下良好的基础。

实验 6 极点配置与全维状态观测器的设计

一、实验目的

1. 加深对状态反馈作用的理解。
2. 学习和掌握状态观测器的设计方法。

二、实验原理

在 MATLAB 中，可以使用 `acker` 和 `place` 函数来进行极点配置。

三、实验内容

3.1 第 1 题

1. 实现代码

```
1. %% 计算系统特征值
2. eig_orig = eig(A);
3. fprintf('系统特征值: %.2f, %.2f, %.2f\n', eig_orig);
4. % 稳定性判断
5. if all(real(eig_orig) < 0)
6.     disp('系统稳定 (所有特征值实部为负)');
7. else
8.     disp('系统不稳定 (存在特征值实部非负)');
9. end
10.
11. %% 极点配置设计 (期望极点: -1, -2, -3)
12. desired_poles = [-1, -2, -3];
13. % 计算状态反馈矩阵 K
14. K = place(A, B, desired_poles);
15. fprintf('状态反馈矩阵 K = [%.4f, %.4f, %.4f]\n', K);
```

2. 实验结果

经计算可得系统特征值为-1.00, -1.00, 1.00，存在正实部特征值，因此系统不稳定。

状态反馈矩阵 $K = [-1.0000, 2.0000, 4.0000]$ 。

由上述实验可知状态反馈利用全部状态变量进行反馈，而输出反馈仅使用可测量输出。状态反馈能实现任意极点配置（当系统完全能控时），输出反馈通常不能。状态反馈通过改变系统矩阵 A 的特征值（ $A-BK$ ）来实现性能调整。

状态反馈进行零极点配置的前提条件是系统完全能控，即能控性矩阵满秩。

3.2 第2题

1. 实现代码

```
1. % 仿真参数设置
2. t = 0:0.01:10;
3. u = ones(size(t));
4. x0 = [0; 1];
5. x0_hat = [0; 0]; % 观测器初始状态（可任意选）
6.
7. A = [0 1; -3 -4];
8. B = [0; 1];
9. C = [2 0];
10. D = 0; % 原系统直接传递项
11.
12. % 构建原系统状态空间模型
13. sys_plant = ss(A, B, C, D);
14.
15. % 仿真原系统响应，得到输出 y 和状态 x
16. [y_plant, t, x_plant] = lsim(sys_plant, u, t, x0);
17.
18. % 期望观测器极点：-12 ± j
19. des_poles_obs = [-12 + 1j, -12 - 1j];
20.
21. % 利用对偶原理计算观测器增益（系统能观测，可任意配置极点）
22. L = place(A', C', des_poles_obs)'; % 计算得 L = [10; 31]（原代码中 L(2)=21 错误）
23.
24. % 观测器状态方程
25. A_obs = A - L * C;
26. B_obs = [B L]; % 输入为[u; y]，故输入矩阵包含 B（对应 u）和 L（对应 y）
27.
28. % 构建观测器状态空间模型（输出为观测器状态  $\hat{x}$ ）
29. sys_obs = ss(A_obs, B_obs, eye(2), zeros(2, 2));
30.
31. % 步骤 1：将 u 转为列向量（与 y_plant 维度统一为 N×1）
32. u_col = u'; % 1×1001 转 1001×1
33.
34. % 步骤 2：按列拼接构建观测器输入（lsim 要求：每列一个输入通道，每行一个时间点）
35. % 最终 u_obs 为 1001×2 矩阵，第 1 列=u，第 2 列=y_plant，匹配观测器 2 个输入通道
36. u_obs = [u_col, y_plant];
37.
38. % 仿真观测器响应（输入 u_obs 为 N×2 矩阵，匹配 sys_obs 的 2 个输入通道）
39. [x_hat, t, ~] = lsim(sys_obs, u_obs, t, x0_hat);
```

```

40.
41. % 第一幅图：单独显示原系统状态
42. figure;
43. % 子图 1: x1 状态
44. subplot(2,1,1);
45. plot(t, x_plant(:,1), 'b-', 'LineWidth', 2);
46. title('原系统状态 x1');
47. xlabel('时间 t/s');
48. ylabel('状态值');
49. legend('x1');
50. grid on;
51.
52. % 子图 2: x2 状态
53. subplot(2,1,2);
54. plot(t, x_plant(:,2), 'b-', 'LineWidth', 2);
55. title('原系统状态 x2');
56. xlabel('时间 t/s');
57. ylabel('状态值');
58. legend('x2');
59. grid on;
60.
61. % 第二幅图：对比原系统状态与观测器估计
62. figure;
63. % 子图 1: x1 与 x1 估计对比
64. subplot(2,1,1);
65. plot(t, x_plant(:,1), 'b-', 'LineWidth', 2);
66. hold on;
67. plot(t, x_hat(:,1), 'r--', 'LineWidth', 2);
68. title('状态 x1 与观测器估计 x1 对比');
69. xlabel('时间 t/s');
70. ylabel('状态值');
71. legend('x1', 'x1 估计');
72. grid on;
73.
74. % 子图 2: x2 与 x2 估计对比
75. subplot(2,1,2);
76. plot(t, x_plant(:,2), 'b-', 'LineWidth', 2);
77. hold on;
78. plot(t, x_hat(:,2), 'r--', 'LineWidth', 2);
79. title('状态 x2 与观测器估计 x2 对比');
80. xlabel('时间 t/s');
81. ylabel('状态值');

```



```
82. legend('x2', 'x2 估计');
```

```
83. grid on;
```

2. 实验结果

原系统的状态曲线如图 6-1 所示：

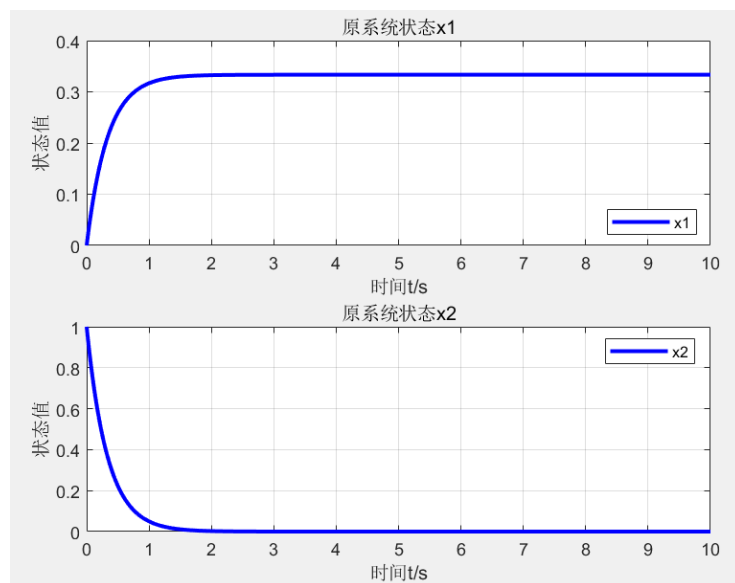


图 6-1：状态曲线

设观测器初始状态为零状态，观测器的状态曲线和原状态对比如图 6-2 所示：

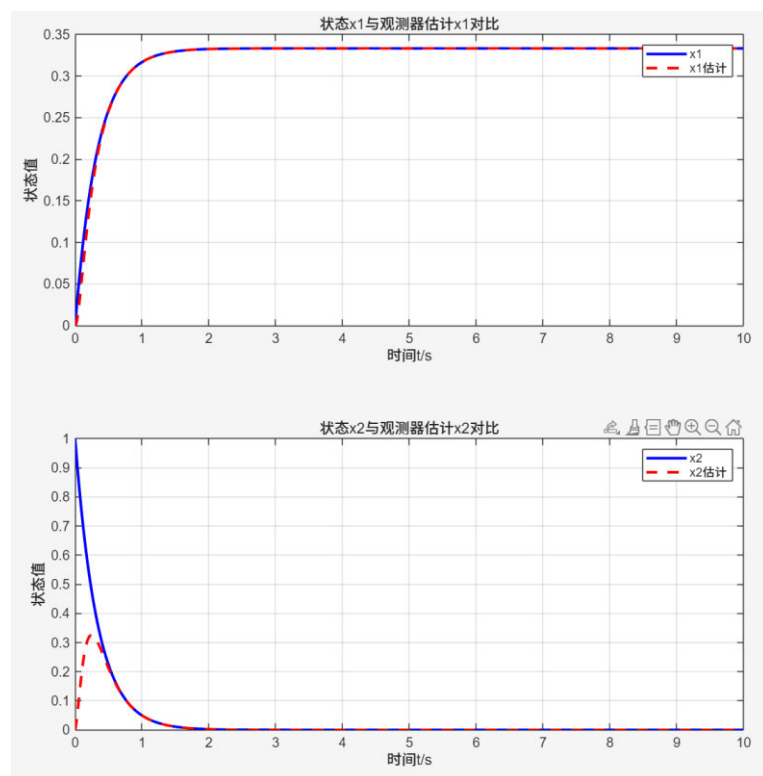


图 6-2：观测器的状态曲线和原状态图

闭环极点选取原则：根据系统动态性能要求（如超调量、调节时间）确定，应当位于左半平面，距离虚轴越远系统响应越快。

观测器极点选取原则：应比最快控制器极点快 3-5 倍，保证状态估计误差快速衰减，同时要避免将观测器极点配置得过快，否则会放大噪声影响。

引入状态观测器后，可使不可测状态变得可用，从而实现状态反馈。根据分离原理，通过合理设计可保持系统稳定性。但状态观测器会引入估计误差，初期存在短暂偏差，在某些情况下可能放大测量噪声。因此状态观测器应当合理设计。

四、 实验总结

4.1 实验中遇到的问题及解决方法

问题：状态估计曲线不收敛，无法跟踪真实状态。

解决方案：检查观测器方程实现是否正确，特别是 $(A-LC)$ 矩阵的计算，并注意观察误差收敛趋势，若误差收敛于零则说明观测器设计正确。

4.2 收获与体会

在本次实验中，我通过状态反馈设计与全维状态观测器的设计，深入理解了现代控制理论的核心原理与应用技巧。实验过程中遇到的系统稳定性判断偏差问题，通过精确计算特征值并验证能控性得以解决，这使我深刻认识到状态反馈对改变系统动态性能的根本性作用。在实验过程中，我观察到了观测器设计中初始误差收敛速度较慢的问题。通过调整增益矩阵使极点合理配置 $(-12 \pm j)$ ，这样就可以保证观测器的快速跟踪，同时避免了噪声敏感性问题，这让我们亲身体会到观测器设计需要在响应速度与抗噪性之间取得平衡的重要性。通过反复调试 MATLAB 代码中的维度匹配与图形标注问题，我们不仅掌握了状态反馈矩阵和观测器增益的计算方法，更深化了对分离原理实践意义的理解——即使存在初始估计误差，闭环系统仍能稳定运行并逐步收敛至理想状态。这些实践中的挑战与解决过程，不仅巩固了理论知识，更培养了在工程应用中识别关键参数、优化系统性能的实战能力。

总结与反思

在本次《控制理论基础》实验中，我通过一系列 MATLAB 仿真实践，系统梳理了控制系统分析与设计的核心知识。从建立系统的传递函数、零极点增益和状态空间模型，到分析暂态特性、根轨迹及频率响应，再到极点配置与状态观测器设计，每个实验都让我对理论知识有了更具象的理解。例如，在暂态特性分析中，通过改变阻尼比和自然频率观察阶跃响应变化，直观验证了参数对系统动态性能的影响；在频率特性分析中，利用 Nyquist 图和 Bode 图判断稳定性，体会到频域分析的便捷性。在这里，每一个实验对我都富有启发意义，使我对控制理论有了更加深入的理解。

实验过程中，我也遇到了很多困难与挑战。起初在系统模型连接时混淆反馈方式，导致传递函数计算错误，后来通过明确不同连接的数学表达和 MATLAB 命令细节才得以解决；在设计超前校正器时，参数选择不合理使相位裕度不达标，经反复调试补偿相角和验证性能才完成优化。这些经历让我深刻认识到，控制系统设计不仅需要扎实的理论基础，更要注重工程实践中的细节处理和误差分析，必须将理论与实践深度结合起来，才能解决复杂控制系统设计和分析中的一系列问题。

此次实验也让我熟练掌握了使用 MATLAB 工具进行控制系统分析与设计的方法，也让我直观地感受到其在控制理论中的重要性，其强大的仿真功能为理论验证提供了直观平台，但也需警惕过度依赖工具可能导致的原理理解偏差。未来学习中，我将更注重理论推导与仿真实践的结合，加强对控制系统设计逻辑的整体把握，提升分析问题和解决问题的能力。