

```
1 %% 实验一a)
2 num = 10;
3 den = conv([1,1],[0.1,1]);
4 sys = tf(num,den);
5 figure('Position', [100, 100, 800, 600]);
6 margin(sys);
7 %% 实验一 a) 实际值
8 clear;
9 num = 9.21;
10 den = conv([1.02 1],[0.1 1]);
11 sys = tf(num,den);
12 figure('Position', [100, 100, 800, 600]);
13 margin(sys);
14 %% 实验一 b) 设计补偿器前的阶跃响应
15 clear;
16 num = 10;
17 den = [0.1 1.1 11];
18 sys = tf(num,den);
19 x = stepinfo(sys);
20 disp(x);
21 figure('Position', [100, 100, 800, 600]);
22 step(sys);
23 %% 实验一 b) 不同K值对应的阶跃响应
24 clear;
25 for k=[0.1 0.2025 1]
26     num = 10;
27     den = [0.1 1.1 10*k+1];
28     sys = tf(k*num,den);
29     hold on
30     step(sys);
31 end
32 hold off
33 %% 实验一 b) 补偿器前根轨迹
34 clear;
35 num = 10;
36 den = conv([1 1], [0.1 1]);
37 sys = tf(num, den);
38 figure; % 确保创建一个新的图形窗口
39 rlocus(sys);
40 ylim([-13 13]);
41 hold on;
42 p1 = plot(-8.05, 11.454, 'rx');
43 text(-8.05, 11.454, '(-8.05, 11.454)', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
44 p2 = plot(-8.05, -11.454, 'rx');
45 text(-8.05, -11.454, '(-8.05, -11.454)', 'VerticalAlignment', 'top', 'HorizontalAlignment', 'right');
46 legend([p1, p2], {'Dominant root'}, 'Location', 'Best');
47 hold off;
48 %% 实验一 b) 加入补偿器后的阶跃响应及数据
49 clear;
50 G = tf(10,[0.1 1.1 1]);
51 K=1.8;
```

```
52 C = tf([1 14.668],[1 22.772]);
53 Gc = K * C * G;
54 T = feedback(Gc, 1);
55 step(T);
56 % x = stepinfo(T);
57 % disp(x);
58 %% 实验一 b) 加入补偿器后的根轨迹图
59 rlocus(T);
60 hold on;
61 p1 = plot(-8.05, 11.454, 'rx');
62 text(-8.05, 11.454, '(-8.05, 11.454)', 'VerticalAlignment', 'bottom', ↵
'HorizontalAlignment', 'right');
63 p2 = plot(-8.05, -11.454, 'rx');
64 text(-8.05, -11.454, '(-8.05, -11.454)', 'VerticalAlignment', 'top', ↵
'HorizontalAlignment', 'right');
65 legend([p1, p2], {'Dominant root'}, 'Location', 'Best');
66 hold off;
67 %% 实验一 c) 加入状态反馈控制器后的阶跃响应
68 clear;
69 num = 100;
70 den = [1 16.1 196];
71 sys= tf(num,den);
72 figure;
73 step(sys);
74 I = stepinfo(sys);
75 disp(I);
76 %% 实验一 c) 状态反馈控制器和相位超前补偿器对比图
77 clear;
78 % 相位超前补偿器
79 G = tf(10,[0.1 1.1 1]);
80 K=1.8;
81 C = tf([1 14.668],[1 22.772]);
82 Gc = K * C * G;
83 T = feedback(Gc, 1);
84
85 % 状态反馈控制器
86 num = 100;
87 den = [1 16.1 196];
88 sys= tf(num,den);
89
90 figure('Position', [100, 100, 800, 600]);
91 hold on
92 step(T);
93 step(sys);
94 hold off
95 legend("Phase-lead", "state-feedback");
```

```
1 %% 求解K为不同取值时, 对应特征方程的跟。即为闭环传输函数的极点
2 K_value = [1 2 3 4 5 6 7.5 8 9 10];
3 all_roots = [];
4 for K = K_value
5     coefficients = [0.01 0.15 0.5 K];
6     roots_values = roots(coefficients);
7     all_roots = [all_roots; roots_values.'];
8     fprintf('当K=%.1f时, 多项式的根为: \n', K);
9     disp(round(roots_values,2));
10 end
11 %% 绘制极点的图
12 [r,c] = size(all_roots);
13 figure('Position', [100, 100, 800, 600]);
14 hold on
15 for i = 1:r
16     x = all_roots(i,:);
17     plot(real(x(1)), imag(x(1)), 'rx');
18     plot(real(x(2)), imag(x(2)), 'gx');
19     plot(real(x(3)), imag(x(3)), 'bx');
20 end
21 title('Complex Roots Plot');
22 xlabel('Real Axis');
23 ylabel('Imaginary Axis');
24 % 添加虚线
25 xline(0, 'LineStyle', '--');
26 yline(0, 'LineStyle', '--');
27 grid on
28 hold off
29 %% 绘制根轨迹的图形
30 clear;
31 num = 1;
32 den = [0.01 0.15 0.5 0];
33 sys = tf(num,den);
34 figure('Position', [100, 100, 800, 600]);
35 rlocus(sys);
36 %% a) 不同K值对应的阶跃响应
37 clear;
38 num = 1;
39 den = conv(conv([0.5 0],[0.2 1]),[0.1 1]);
40 sys = tf(num,den);
41 figure('Position', [100, 100, 800, 600]);
42 hold on
43 legendInfo = cell(1,6); % 创建一个空的字符串数组
44 for K = [1 2 3 4 5 6]
45     Gc = sys * K;
46     H = feedback(Gc, 1);
47     legendInfo{K} = ['K = ', num2str(K)]; % 将每个K值添加到字符串数组中
48     step(H);
49 end
50 legend(legendInfo); % 使用字符串数组作为legend函数的参数
51 hold off
52
53 %创建K=8 9 10 时的阶跃响应
```

```
54 figure('Position', [100, 100, 800, 600]);
55 hold on
56 legendInfo = cell(1,3); % 创建一个大小为3的单元数组
57 index = 1; % 创建一个索引变量
58 for K = [8 9 10]
59     Gc = sys * K;
60     H = feedback(Gc, 1);
61     legendInfo{index} = ['K = ', num2str(K)]; % 使用索引变量来存储数据
62     step(H);
63     index = index + 1; % 更新索引变量
64 end
65 legend(legendInfo);
66 hold off
67 xlim([0 10]);
68 %% b) PID反馈系统
69 clear;
70 num = 1;
71 den = conv(conv([0.5 0],[0.2 1]),[0.1 1]);
72 sys = tf(num, den);
73 PID = tf([0.5 4.50 10.13],[1 0]);
74 Gc = PID * sys;
75 H = feedback(Gc, 1);
76 %step(H);
77 x = stepinfo(H);
78 disp(x);
79
80 %% a) 鲁棒性分析
81 clear;
82 z_values = [0.5855, 0.5750];
83 w_values = [14.6793, 16.5317];
84
85 figure('Position', [100, 100, 800, 600]);
86 hold on
87 for i = 1:2
88     z = z_values(i);
89     w = w_values(i);
90     num = w^2;
91     den = [1 2*z*w w^2];
92     sys = tf(num, den);
93     legend("State-feedback", "Phase-lead");
94     margin(sys);
95 end
96 hold off
97 % Phase-lead compensator: 112度(at 11.6rad/s)
98 % State-feedback controller: 109度(at 13.6rad/s)
99 %% 实验二 b) PID系统的评价
100 % 存在PID的系统
101 num = 1;
102 den = conv(conv([0.5 0],[0.2 1]),[0.1 1]);
103 sys = tf(num, den);
104 PID = tf([0.5 4.50 10.13],[1 0]);
105 Gc = PID * sys;
106 HPID = feedback(Gc, 1);
```

```
107
108 figure('Position', [100, 100, 800, 600]);
109 hold on
110 step(HPID);
111
112 %不存在PID的系统
113 legendInfo = cell(1,4); % 创建一个大小为4的单元数组
114 legendInfo{1} = 'PID'; % 将'PID'添加到图例条目中
115 index = 2; % 创建一个索引变量
116 for K = [1 3 5]
117     Gc = sys * K;
118     H = feedback(Gc, 1);
119     legendInfo{index} = ['K = ', num2str(K)]; % 使用索引变量来存储数据
120     step(H);
121     index = index + 1; % 更新索引变量
122 end
123 legend(legendInfo); % 在循环结束后调用一次legend函数
124 hold off
125
```