```
%% 实验一a)
num = 10;
den = conv([1,1],[0.1,1]);
sys = tf(num,den);
figure('Position', [100, 100, 800, 600]);
margin(sys);
%% 实验一 a) 实际值
clear;
num = 9.21;
den = conv([1.02 1],[0.1 1]);
sys = tf(num,den);
figure('Position', [100, 100, 800, 600]);
margin(sys);
%% 实验一 b) 设计补偿器前的阶跃响应
clear;
num = 10;
den = [0.1 1.1 11];
sys = tf(num,den);
x = stepinfo(sys);
disp(x);
figure('Position', [100, 100, 800, 600]);
step(sys);
%% 实验一 b) 不同K值对应的阶跃响应
clear;
for k=[0.1 0.2025 1]
    num = 10;
    den = [0.1 1.1 10*k+1];
    sys = tf(k*num,den);
    hold on
    step(sys);
end
hold off
%% 实验一 b) 补偿器前根轨迹
clear;
num = 10;
den = conv([1 1], [0.1 1]);
sys = tf(num, den);
figure; % 确保创建一个新的图形窗口
rlocus(sys);
ylim([-13 13]);
hold on;
p1 = plot(-8.05, 11.454, 'rx');
text(-8.05, 11.454, '(-8.05, 11.454)', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
p2 = plot(-8.05, -11.454, 'rx');
text(-8.05, -11.454, '(-8.05, -11.454)', 'VerticalAlignment', 'top', 'HorizontalAlignment', 'right');
legend([p1, p2], {'Dominant root'}, 'Location', 'Best');
hold off;
%% 实验一 b) 加入补偿器后的阶跃响应及数据
clear;
G = tf(10,[0.1 1.1 1]);
K=1.8;
```

```
C = tf([1 14.668],[1 22.772]);
Gc = K * C * G;
T = feedback(Gc, 1);
step(T);
% x = stepinfo(T);
% disp(x);
%% 实验一 b) 加入补偿器后的根轨迹图
rlocus(T);
hold on;
p1 = plot(-8.05, 11.454, 'rx');
text(-8.05, 11.454, '(-8.05, 11.454)', 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
p2 = plot(-8.05, -11.454, 'rx');
text(-8.05, -11.454, '(-8.05, -11.454)', 'VerticalAlignment', 'top', 'HorizontalAlignment', 'right');
legend([p1, p2], {'Dominant root'}, 'Location', 'Best');
hold off;
%% 实验一 c) 加入状态反馈控制器后的阶跃响应
clear;
num = 100;
den = [1 16.1 196];
sys= tf(num,den);
figure;
step(sys);
I = stepinfo(sys);
disp(I);
%% 实验一 c) 状态反馈控制器和相位超前补偿器对比图
clear;
% 相位超前补偿器
G = tf(10,[0.1 1.1 1]);
K=1.8;
C = tf([1 14.668],[1 22.772]);
Gc = K * C * G;
T = feedback(Gc, 1);

% 状态反馈控制器
num = 100;
den = [1 16.1 196];
sys= tf(num,den);

figure('Position', [100, 100, 800, 600]);
hold on
step(T);
step(sys);
hold off
legend("Phase-lead","state-feedback");
```

```
%% 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
K_value = [1 2 3 4 5 6 7.5 8 9 10];
all_roots = [];
for K = K_value
    cofficients = [0.01 0.15 0.5 K];
    roots_values = roots(cofficients);
    all_roots = [all_roots; roots_values.'];
    fprintf('当K=%.1f时，多项式的根为：\n', K);
    disp(round(roots_values,2));
end
%% 绘制极点的图
[r,c] = size(all_roots);
figure('Position', [100, 100, 800, 600]);
hold on
for i = 1:r
    x = all_roots(i,:);
    plot(real(x(1)), imag(x(1)), 'rx');
    plot(real(x(2)), imag(x(2)), 'gx');
    plot(real(x(3)), imag(x(3)), 'bx');
end
title('Complex Roots Plot');
xlabel('Real Axis');
ylabel('Imaginary Axis');
% 添加虚线
xline(0,'LineStyle','--');
yline(0,'LineStyle','--');
grid on
hold off
%% 绘制根轨迹的图形
clear;
num = 1;
den = [0.01 0.15 0.5 0];
sys = tf(num,den);
figure('Position', [100, 100, 800, 600]);
rlocus(sys);
%% a) 不同K值对应的阶跃响应
clear;
num = 1;
den = conv(conv([0.5 0],[0.2 1]),[0.1 1]);
sys = tf(num,den);
figure('Position', [100, 100, 800, 600]);
hold on
legendInfo = cell(1,6); % 创建一个空的字符串数组
for K = [1 2 3 4 5 6]
    Gc = sys * K;
    H = feedback(Gc, 1);
    legendInfo{K} = ['K = ', num2str(K)]; % 将每个K值添加到字符串数组中
    step(H);
end
legend(legendInfo); % 使用字符串数组作为legend函数的参数
hold off

%创建K=8 9 10 时的阶跃响应
```

```matlab
figure('Position', [100, 100, 800, 600]);
hold on
legendInfo = cell(1,3); % 创建一个大小为3的单元数组
index = 1; % 创建一个索引变量
for K = [8 9 10]
    Gc = sys * K;
    H = feedback(Gc, 1);
    legendInfo{index} = ['K = ', num2str(K)]; % 使用索引变量来存储数据
    step(H);
    index = index + 1; % 更新索引变量
end
legend(legendInfo);
hold off
xlim([0 10]);
%% b) PID反馈系统
clear;
num = 1;
den = conv(conv([0.5 0],[0.2 1]),[0.1 1]);
sys = tf(num, den);
PID = tf([0.5 4.50 10.13],[1 0]);
Gc = PID * sys;
H = feedback(Gc, 1);
%step(H);
x = stepinfo(H);
disp(x);

%% a) 鲁棒性分析
clear;
z_values = [0.5855, 0.5750];
w_values = [14.6793, 16.5317];

figure('Position', [100, 100, 800, 600]);
hold on
for i = 1:2
    z = z_values(i);
    w = w_values(i);
    num = w^2;
    den = [1 2*z*w w^2];
    sys = tf(num, den);
    legend("State-feedback","Phase-lead");
    margin(sys);
end
hold off
% Phase-lead compensator: 112度(at 11.6rad/s)
% State-feedback controller: 109度(at 13.6rad/s)
%% 实验二 b) PID系统的评价
% 存在PID的系统
num = 1;
den = conv(conv([0.5 0],[0.2 1]),[0.1 1]);
sys = tf(num, den);
PID = tf([0.5 4.50 10.13],[1 0]);
Gc = PID * sys;
HPID = feedback(Gc, 1);
```

```
figure('Position', [100, 100, 800, 600]);
hold on
step(HPID);

%不存在PID的系统
legendInfo = cell(1,4); % 创建一个大小为4的单元数组
legendInfo{1} = 'PID'; % 将'PID'添加到图例条目中
index = 2; % 创建一个索引变量
for K = [1 3 5]
    Gc = sys * K;
    H = feedback(Gc, 1);
    legendInfo{index} = ['K = ', num2str(K)]; % 使用索引变量来存储数据
    step(H);
    index = index + 1; % 更新索引变量
end
legend(legendInfo); % 在循环结束后调用一次legend函数
hold off
```