MATLAB实验报告

实验内容1

exp1

代码实现

```
A = [-2.8 -1.4 0 0;
     1.4 0 0 0;
     -1.8 -0.3 -1.4 -0.6;
     0 0 0.6 0];
B = [1;0;1;0];
C = [0 0 0 1];
D = 0;
[num,den] = ss2tf(A,B,C,D);%传递函数分子分母
G = tf(num,den)          % 传递函数
[z,p,k] = tf2zp(num,den) ;  %  零极点
zpModel = zpk(z,p,k)
```

输出结果

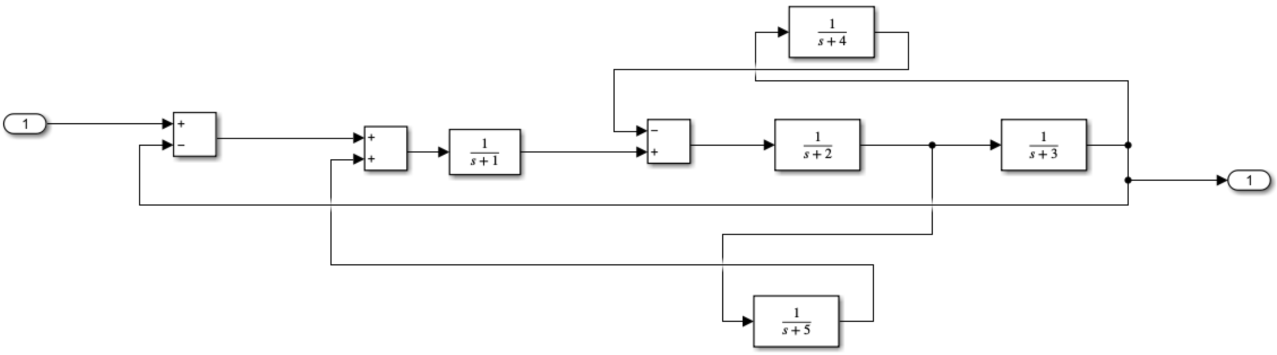
```
G =

              0.6 s^2 + 0.6 s + 0.924
  -------------------------------------------
  s^4 + 4.2 s^3 + 6.24 s^2 + 3.752 s + 0.7056

Continuous-time transfer function.


zpModel =

       0.6 (s^2 + s + 1.54)
  ------------------------------
  (s+1.4)^2 (s+1.061) (s+0.3394)

连续时间零点/极点/增益模型。
```

稳定性判别

```
所有极点都在左半平面（实部小于0），故系统稳定。
```

exp2

simulink模型搭建



输出结果

```
[A,B,C,D] = linmod('exp1_2');
[num,den] = ss2tf(A,B,C,D);
G = tf(num,den)
```

```
G =

                   s^2 + 9 s + 20
  ---------------------------------------------
  s^5 + 15 s^4 + 85 s^3 + 226 s^2 + 282 s + 133

Continuous-time transfer function.
```

对比

```
经过手算后发现，与方框法结果相同。
```

实验内容2

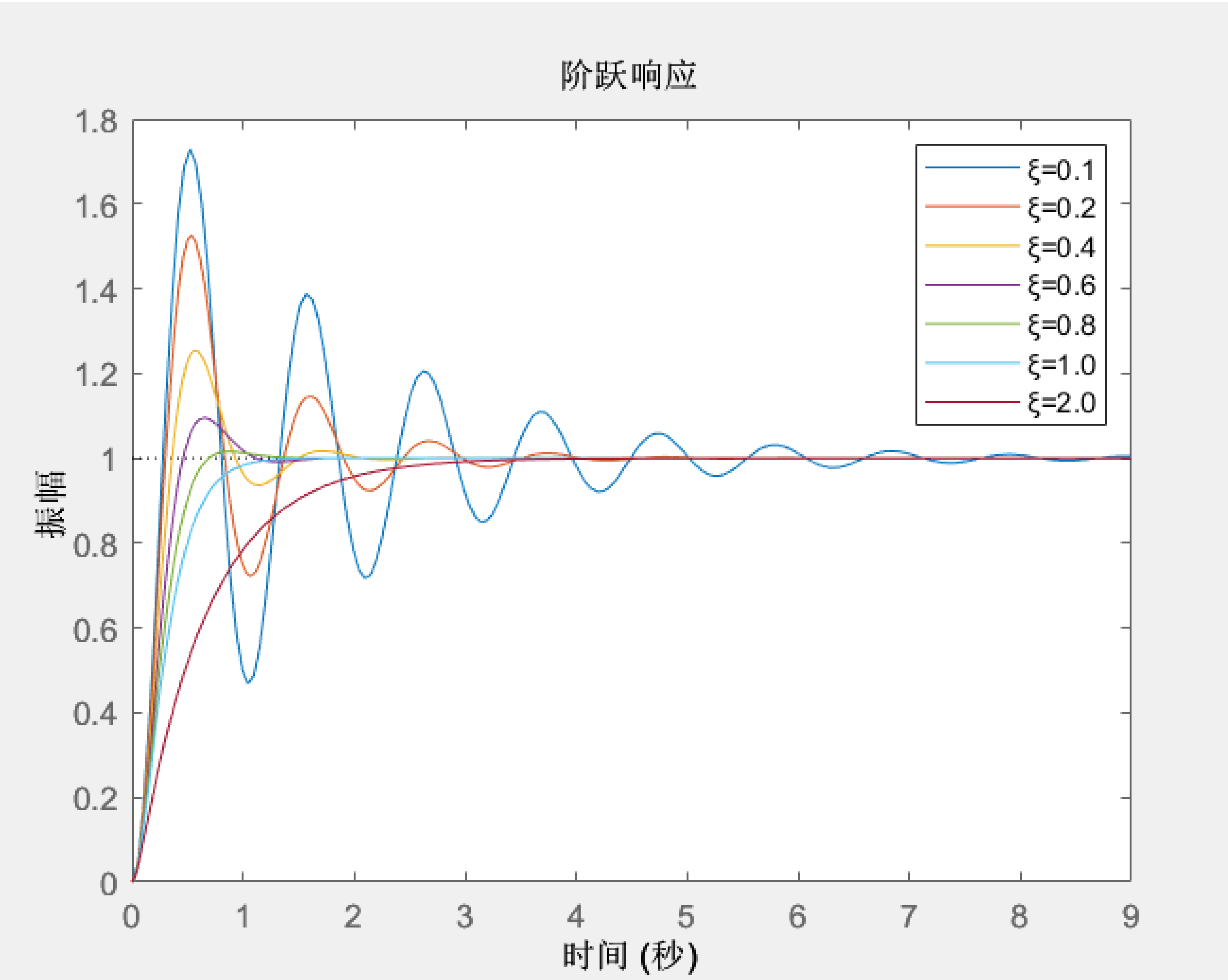
exp1

第一小题代码实现

为了不让图像过于拥挤，中间取点0.4,0.6,0.8。

```
kexi=[0.1 0.2 0.4 0.6 0.8 1.0 2.0];
wn=6;
hold on;
for kexii=kexi
    num=[wn^2];
    den=[1 2*kexii*wn wn^2];
    G=tf(num,den);

    step(G);    %可以直接得到阶跃响应图像
end
legend('ξ=0.1','ξ=0.2','ξ=0.4','ξ=0.6','ξ=0.8','ξ=1.0','ξ=2.0');
hold off
```

输出结果

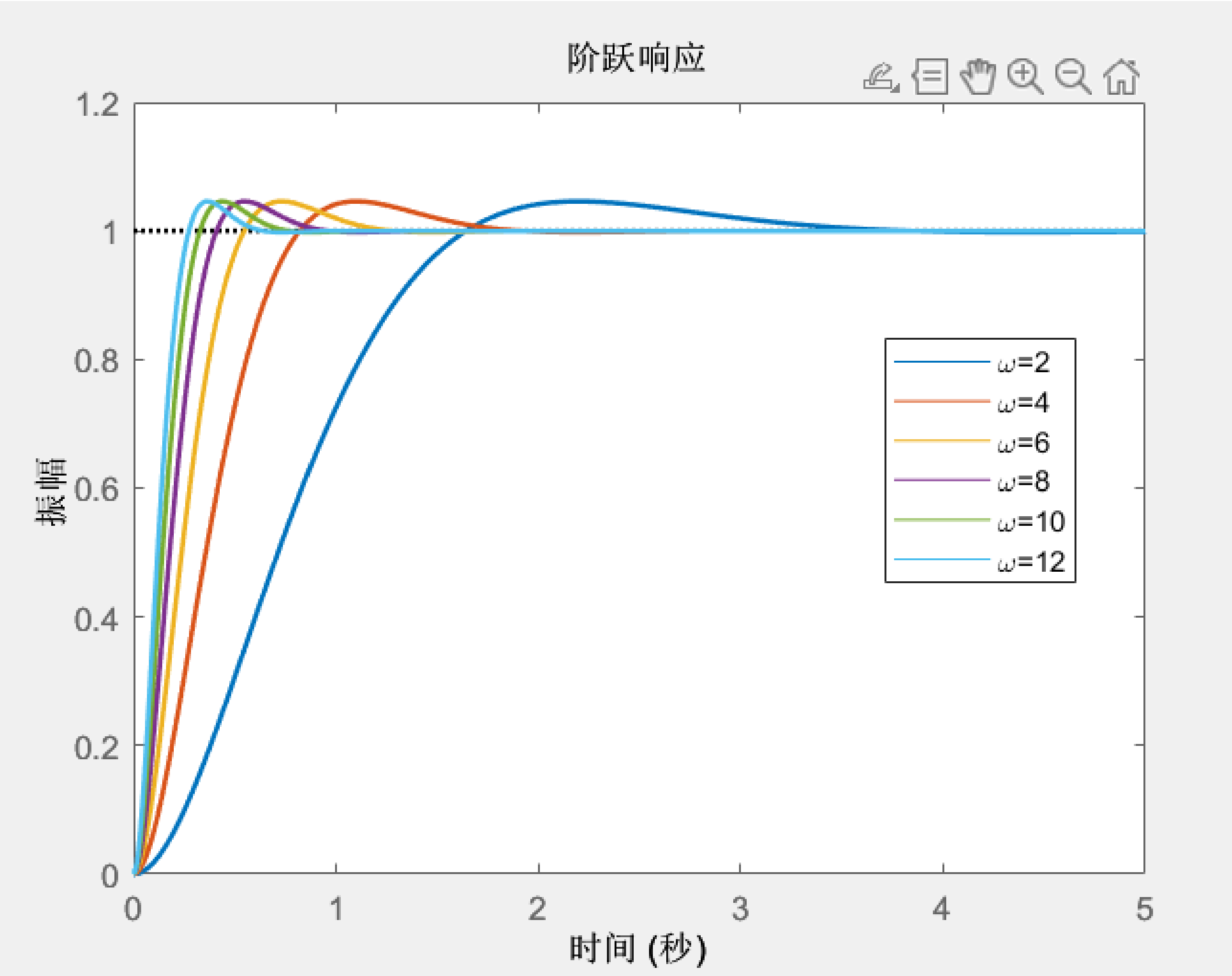


第二小题代码实现

```
wn=2:2:12;
kexi=0.7;
hold on;
```

```
for wni=wn
    num=[wni^2];
    den=[1 2*kexi*wni wni^2];
    G=tf(num,den);

    step(G);    %可以直接得到阶跃响应图像
end
legend('\omega=2','\omega=4','\omega=6','\omega=8','\omega=10','\omega=12');
hold off
set(findall(gcf,'Type','line'),'LineWidth',1.5);
```

输出结果



exp2

代码实现

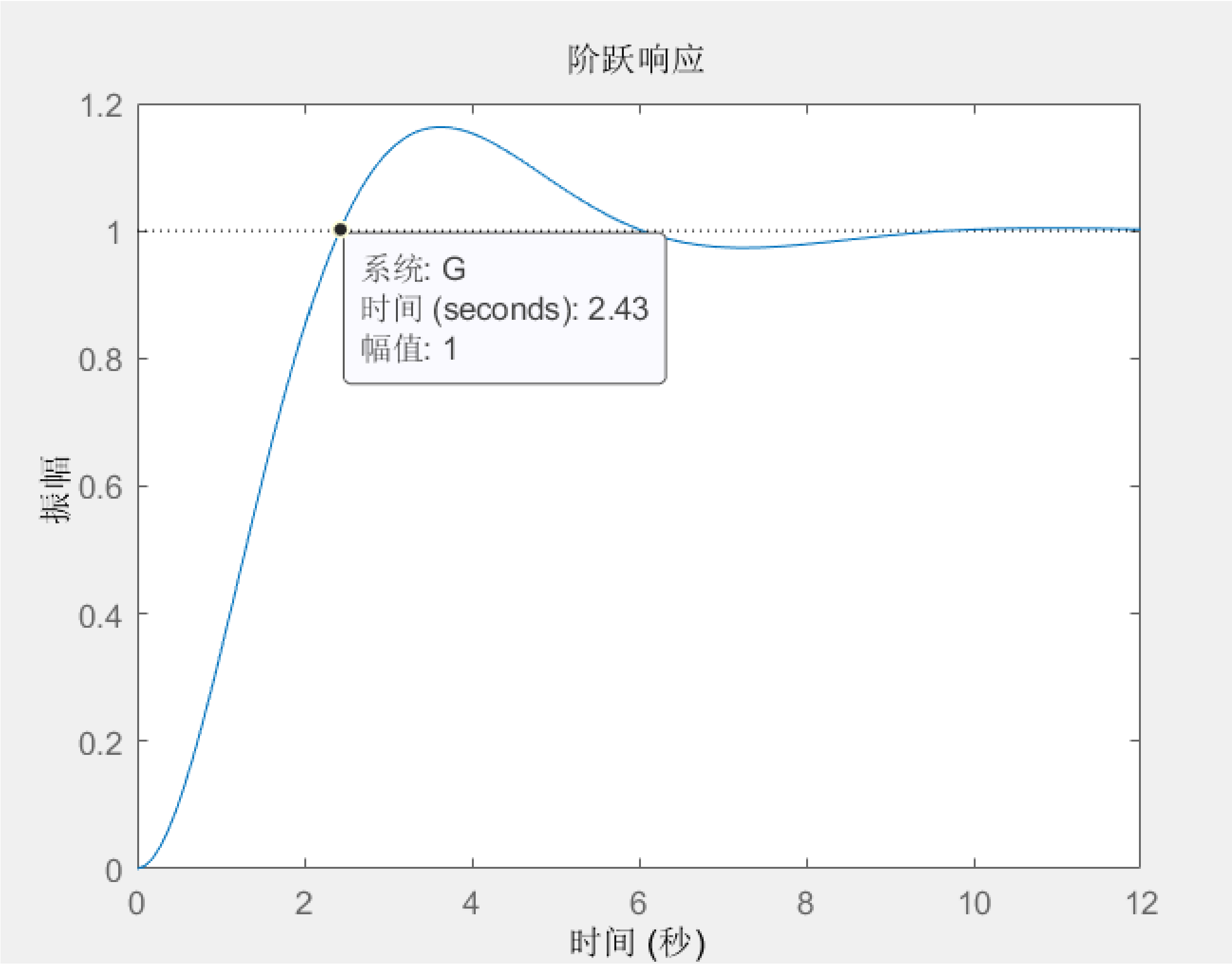
```
kexi=0.5;
wn=1;
G=tf(1,[1 2*kexi*wn,wn^2]);
[y,t] = step(G);
step(G)
[y_max,i] = max(y);
```

```
Mp = y_max-1;%超调量
T_p = t(i);%峰值时间
minn = 100;
for i = 1:length(t)
    if abs(y(i)-1)<=0.01
        ti = i;
        break
    end
end
T_r = t(ti);

for i = length(t):-1:1
    if y(i) <= 0.98 || y(i)>= 1.02 % 取2%的误差
        ti = i;
        break
    end
end

T_s = t(ti);
fprintf('上升时间:%f s\n峰值时间：%f s\n稳态时间：%f s\n超调量：%f \n', T_r,
T_p,T_s,Mp);
```

输出结果

```
上升时间:2.394688 s
峰值时间：3.592033 s
稳态时间：8.012996 s
超调量：0.162929
```

值得一提的是，在上升时间这里，因为t值的离散化，导致难以得到一个精确的值，但从图像可以发现



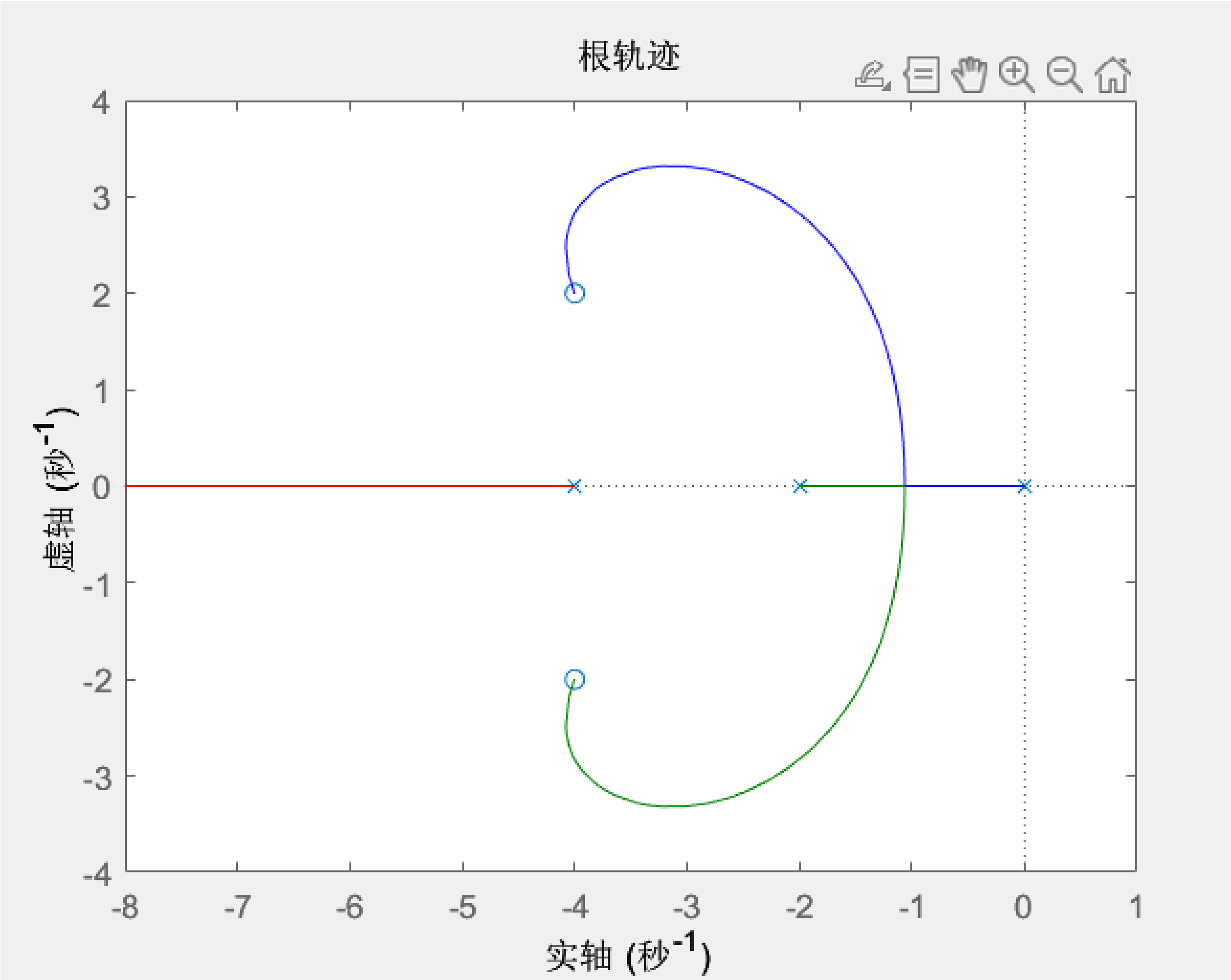
上升时间大约为为2.43s

实验内容3

exp1

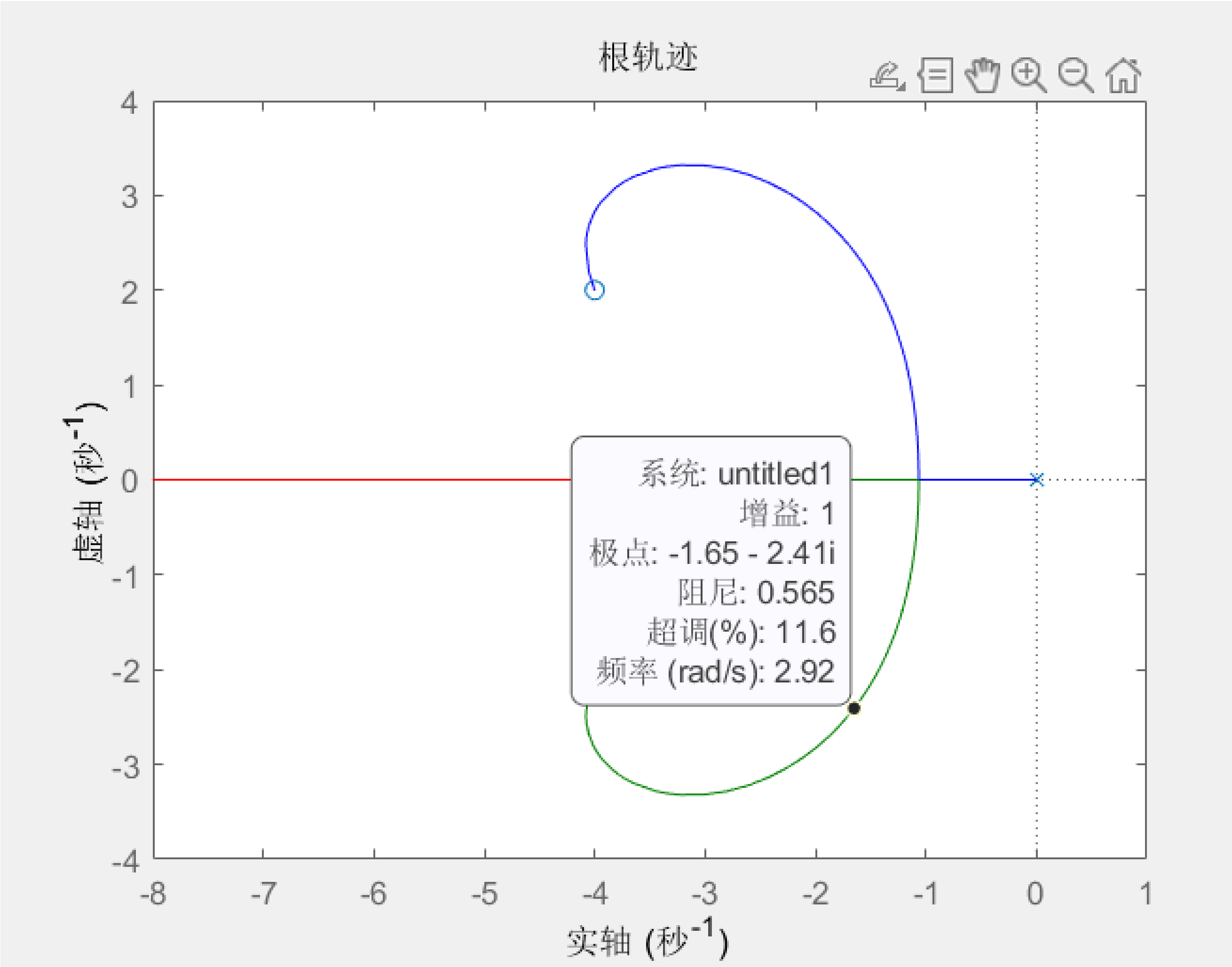
绘制根轨迹

```
numG = [10,80,200];
denG = [1,4,0];
G = tf(numG,denG);
numH = 0.2;
denH = [1 2];
H = tf(numH,denH);
rlocus(G*H);
```



放大系数

通过拖动图上的点得到小阻尼系数




```
此时A = 1, ξ = 0.565
```

闭环函数

```
sys = G/(1+G*H);
[num, den] = tfdata(sys);  % 获取传递函数的系数
[z, p, k] = tf2zp(num{1}, den{1});  % 计算零极点和增益
zpkModel = zpk(z, p, k)  % 创建零极点增益模型
```

```
zpkModel =

      10 s (s+4) (s+2) (s^2 + 8s + 20)
  ------------------------------------------
  s (s+4.706) (s+4) (s^2 + 3.294s + 8.499)

连续时间零点/极点/增益模型。
```

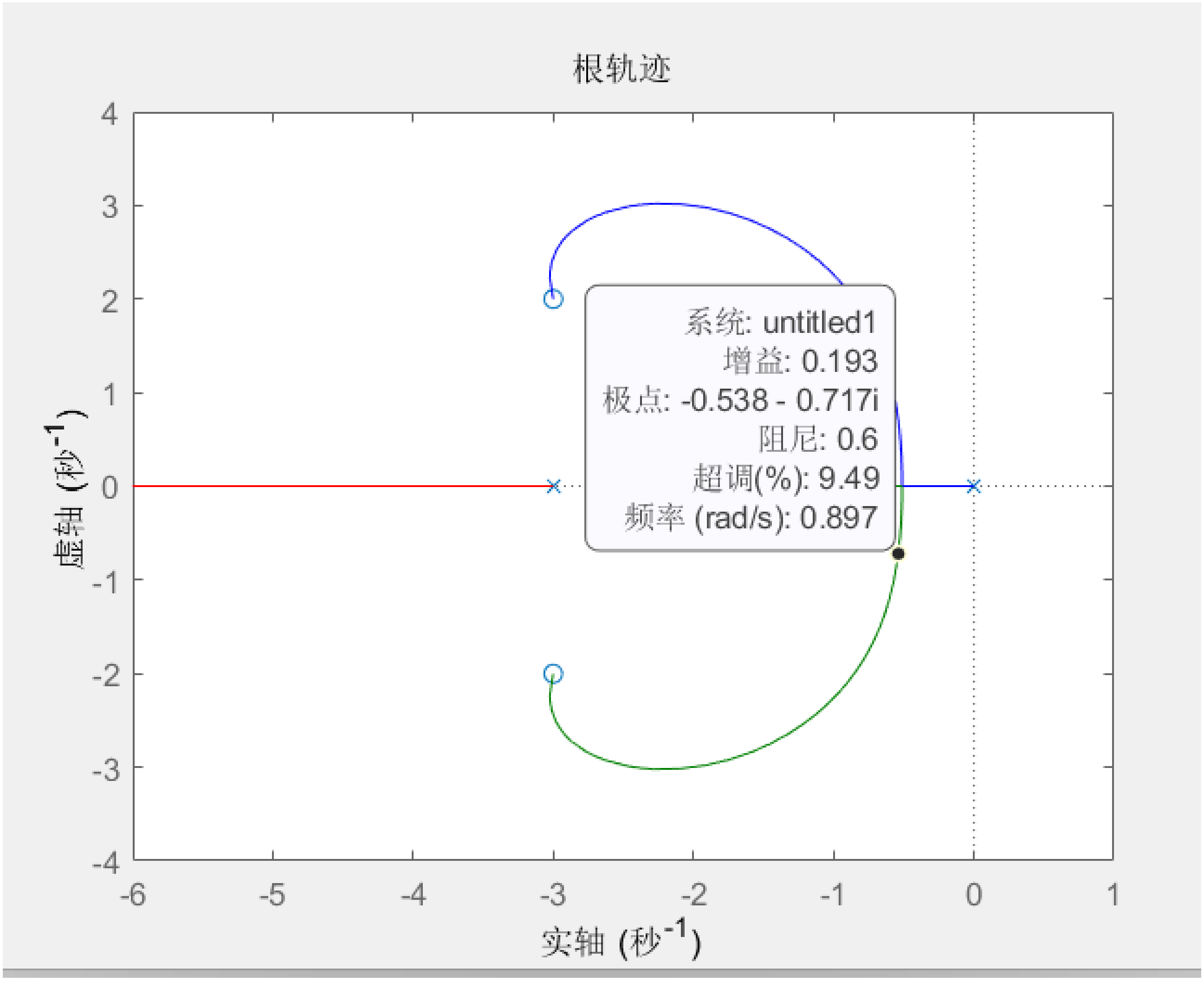
exp2

画出根轨迹

```
numG = [1,6,13];
denG = [1,3,0];
G = tf(numG,denG);
numH = 1;
denH = [1 1];
H = tf(numH,denH);
rlocus(G*H);
```

K值

借助根轨迹图




```
得到  K = 0.193, ξ = 0.6
```

闭环传递函数

```
sys = 0.173*G/(1+0.173*G*H);
[num, den] = tfdata(sys);  % 获取传递函数的系数
[z, p, k] = tf2zp(num{1}, den{1});  % 计算零极点和增益
zpkModel = zpk(z, p, k)  % 创建零极点增益模型
```

```
zpkModel =

     0.173 s (s+3) (s+1) (s^2 + 6s + 13)
  ------------------------------------------
  s (s+3.106) (s+3) (s^2 + 1.067s + 0.7241)

连续时间零点/极点/增益模型。
```

如上所示