**实验三、MATLAB/Simulink在时域分析法中的应用**

**实验时间：2020-12-30 实验地点：线上实验**

**姓名学号： 成绩：**

**一、实验目的**

1、掌握时域分析中Matlab/Simulink的函数

2、掌握Matlab/Simulink在稳定性分析中的应用

**二、实验设备**

微型电子计算机，MATLAB软件

**三、预习要求及思考题**

1、自动控制原理中的时域分析法

2、时域分析中MATLAB/Simulink的函数

**四、讲授内容**

1、简要介绍自动控制原理中的时域分析法

2、介绍时域分析中MATLAB/Simulink的函数

3、系统稳定性分析

**五、实验内容**

1、已知单位负反馈系统的开环传递函数为，用MATLAB求其单位阶跃响应，绘制出响应曲线，并对单位阶跃响应求其性能指标。

2、已知控制系统的传递函数为 ， 试求

（1）转化成使用零点、极点表示的传递函数；

（2）绘制零极点图，并求其零点、极点；

（3）试求在输入函数为sin(t)的时间响应，时间0~5s。

3、已知某二阶系统的传递函数为，**分子项为平方**

（1）将自然频率固定为ωn=1，ζ=0,0.1,0.5,1,2,3,5，利用MATLAB计算并分析ζ变化时系统的单位阶跃响应；

（2）将阻尼比固定为ζ=0.55，利用MATLAB计算并分析自然频率ωn变化时系统的阶跃响应（ωn的变化范围为0.1~1）。

4、已知4阶系统的闭环传递函数为 ，利用MATLAB求其二阶近似系统的单位阶跃响应c2(t)，并与4阶系统的单位阶跃响应c4(t)进行比较，时间是0~4s，简要说明此简化是否合理。

5、某随动系统结构如图所示。利用MATLAB完成如下工作：

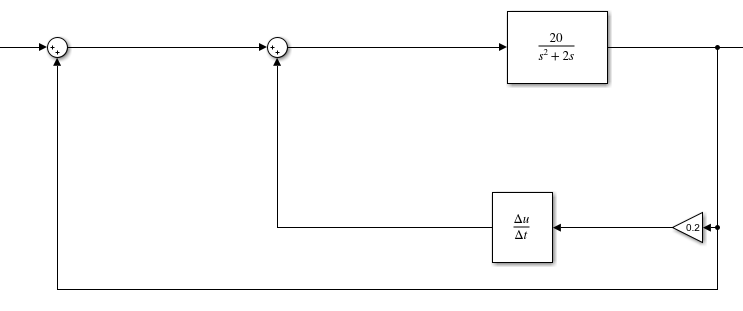
（1）求系统的传递函数；

（2）绘制零极点位置，分析系统的稳定性；

（3）分别利用MATLAB和Simulink建模来绘制阶跃响应曲线；

（4）分析系统的总体性能，包括系统的上升时间，峰值时间和调整时间，并给出理论解释；

（5）计算或观察系统的稳态误差。



**六、实验数据与结果**

**第一题：**

**代码：**

>> num=[4];

>> den=[1 2 2];

>> sys=tf(num,den);

>> step(sys)

>> [y t x]=step(sys);

>> mp=max(y);

>> tp=spline(y,t,mp)

tp = 3.1315

>> cs=length(t)

cs = 139

>> yss=y(cs)

yss = 1.9963

>> ct=(mp-yss)/yss

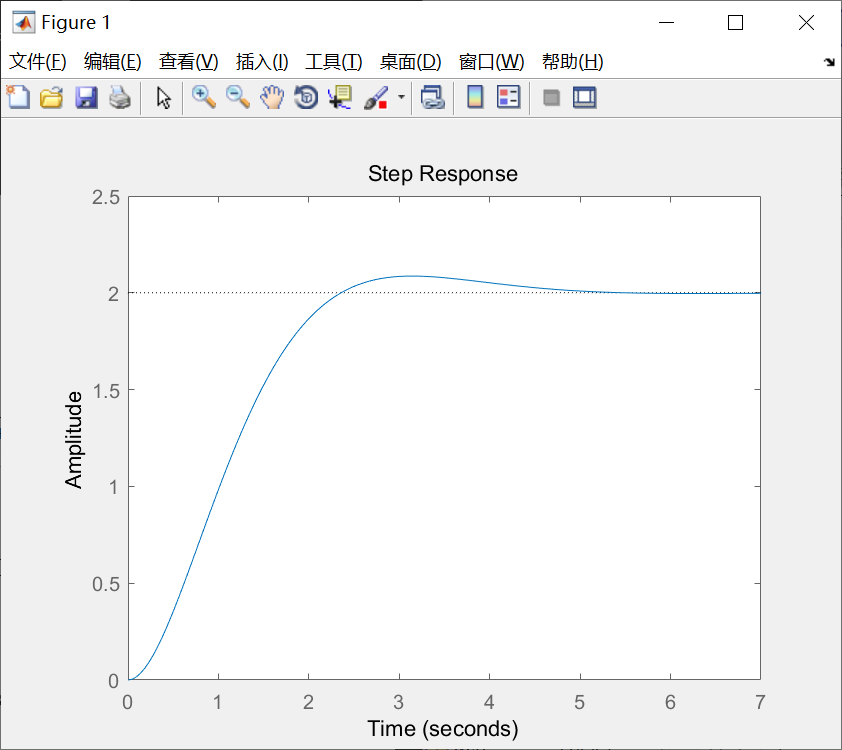
ct = 0.0452

**峰值时间： 3.1315**

**稳定值： 1.9963**

**超调量： 0.0452**

**图像如下：**



**第二题：**

**（1）、代码：**

>> num=[1 2];

>> den=[1 5 10 8];

>> sys=tf(num,den);

>> [z p k]=tf2zp(num,den);

>> zpk(sys)

ans =

(s+2)

--------------------

(s+2) (s^2 + 3s + 4)

Continuous-time zero/pole/gain model.

**使用零点极点表示的传递函数为：**

**(s+2)**

**--------------------------**

**(s+2) (s^2 + 3s + 4)**

**（2）、代码：**

pzmap(sys)

[z p]=tf2zp(num,den)

z = -2

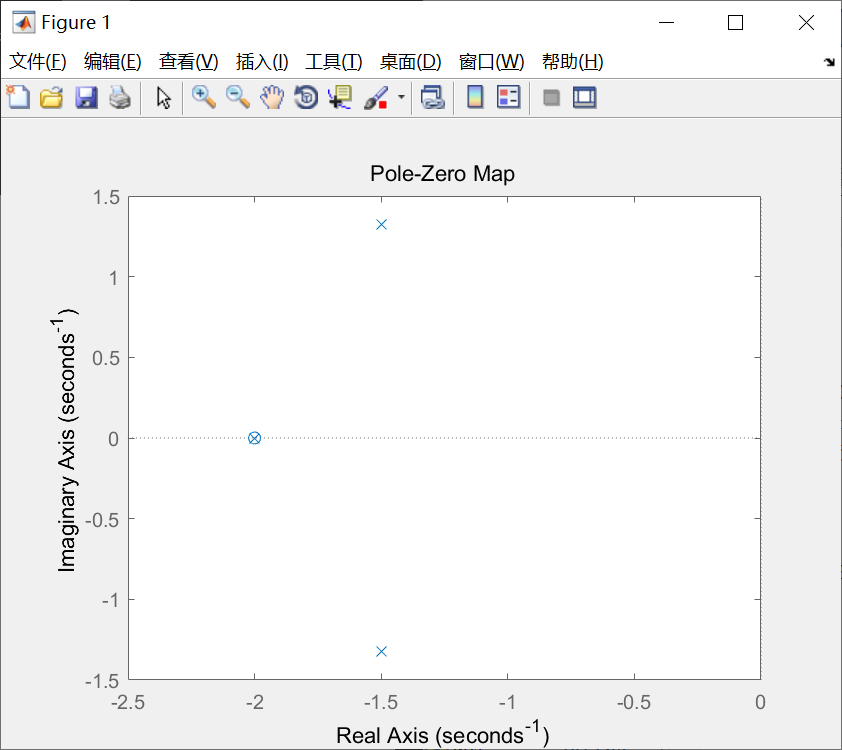
p =

-2.0000 + 0.0000i

-1.5000 + 1.3229i

-1.5000 - 1.3229i

**图像如下：**



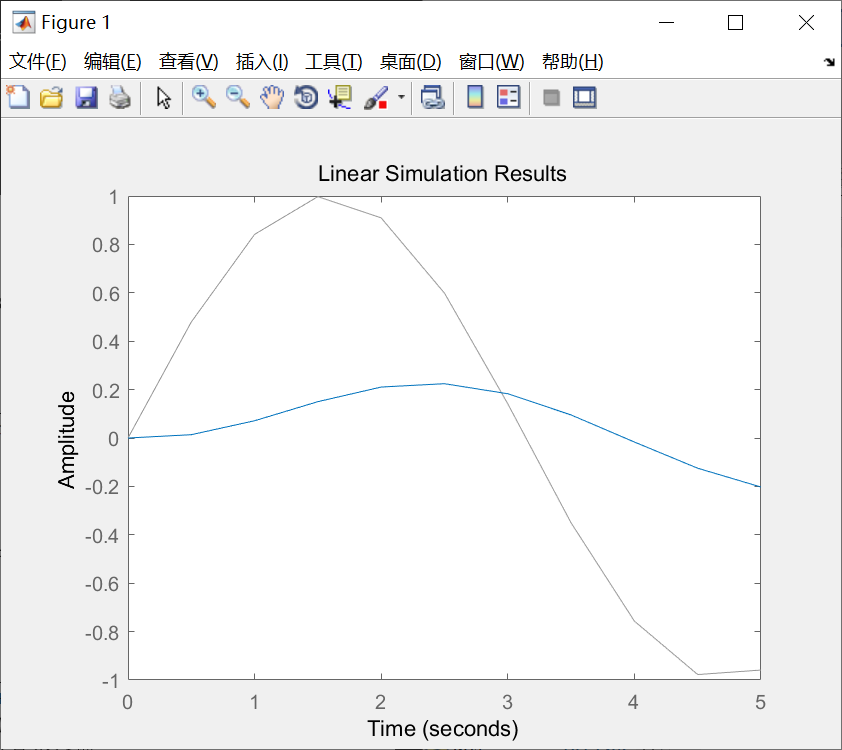
**（3）、代码：**

>> t=0:0.5:5;

>> y=sin(t);

>> lsim(sys,y,t)

**图像如下：**



**第三题：**

**代码：**

>> num=[1];

>> den1=[1 0 1];

>> den2=[1 0.2 1];

>> den3=[1 1 1];

>> den4=[1 2 1];

>> den5=[1 4 1];

>> den6=[1 6 1];

>> den7=[1 10 1];

>> sys1=tf(num,den1);

>> sys2=tf(num,den2);

>> sys3=tf(num,den3);

>> sys4=tf(num,den4);

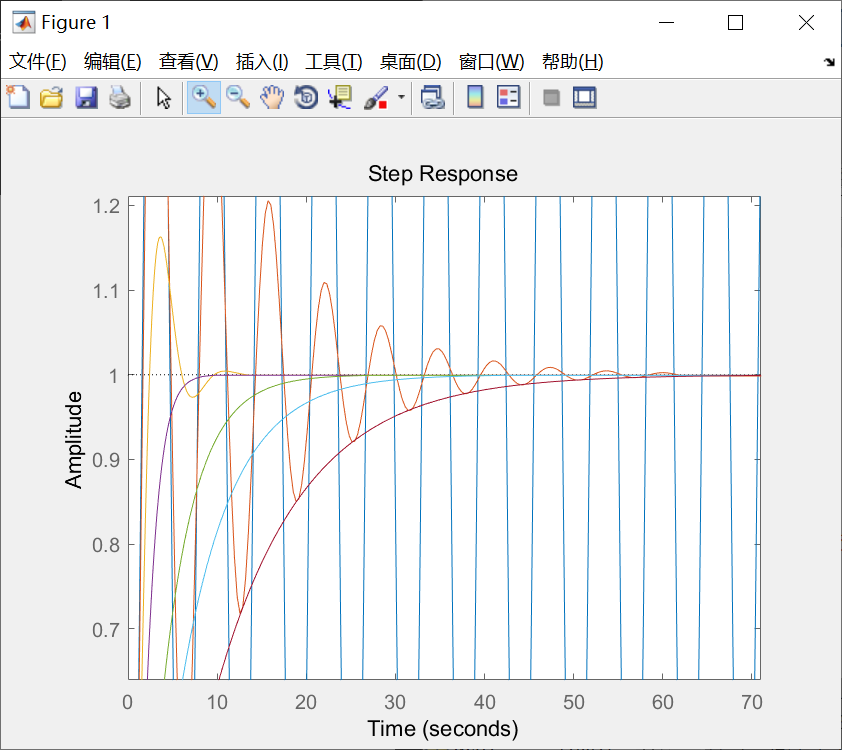
>> sys5=tf(num,den5);

>> sys6=tf(num,den6);

>> sys7=tf(num,den7);

>> step(sys1,sys2,sys3,sys4,sys5,sys6,sys7);

**图像如下：**



**由于图像混在一起很难看清具体的变化趋势，所以我将每个ζ取值单独做图，获得了以下所示的各个图像：**

**各图像分开如下所示：**

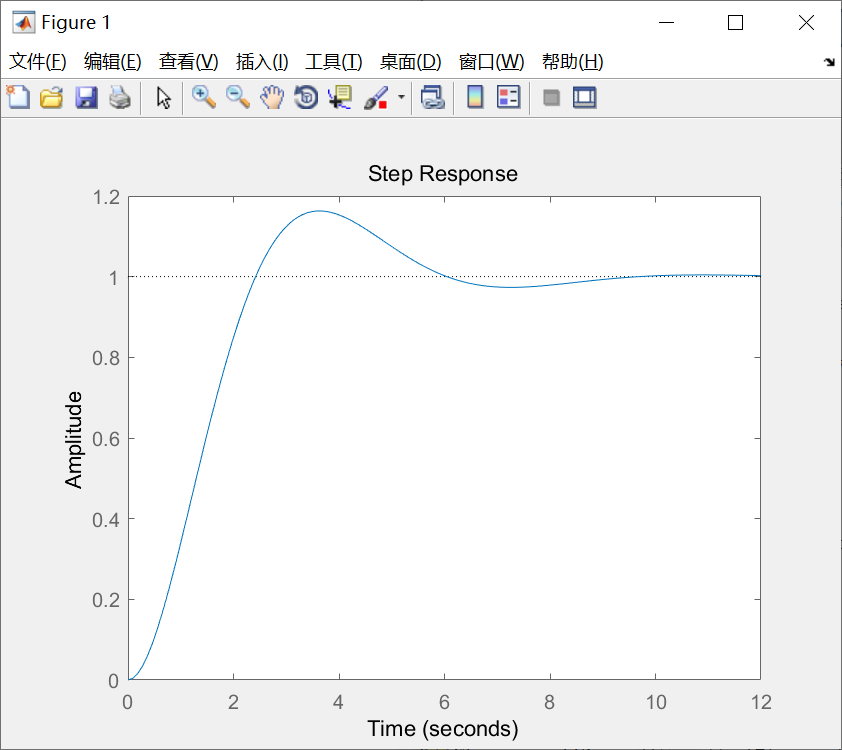
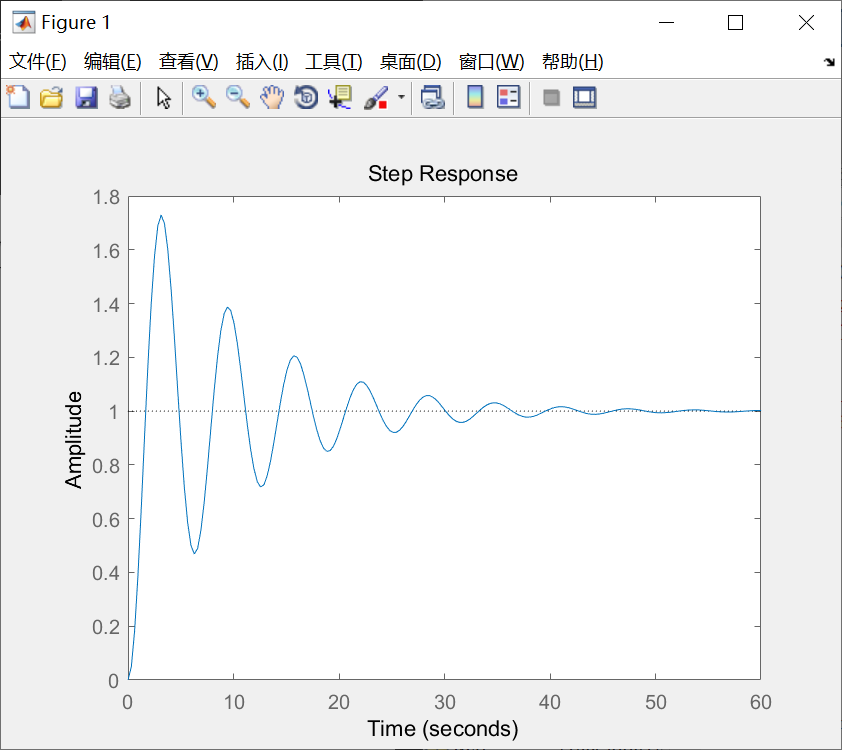
**ζ= 0.2:**

**tp =3.1416**

**yss =1.0044**

**ct =0.7216**

**图像见下左图：**



**ζ= 1:**

**tp = 3.5920**

**yss = 1.0040**

**ct = 0.1583**

**图像见上右图**

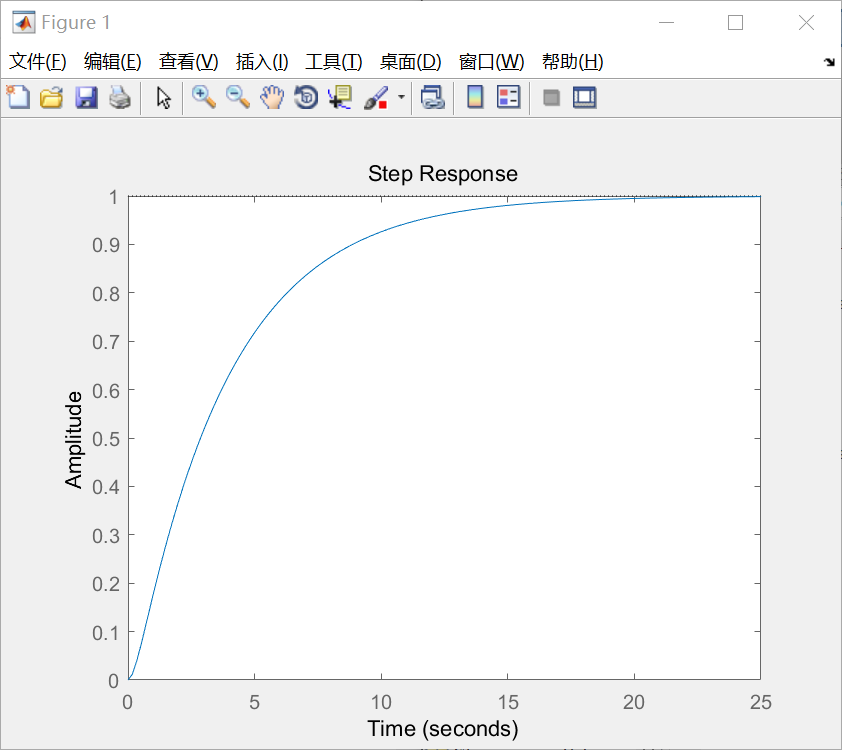
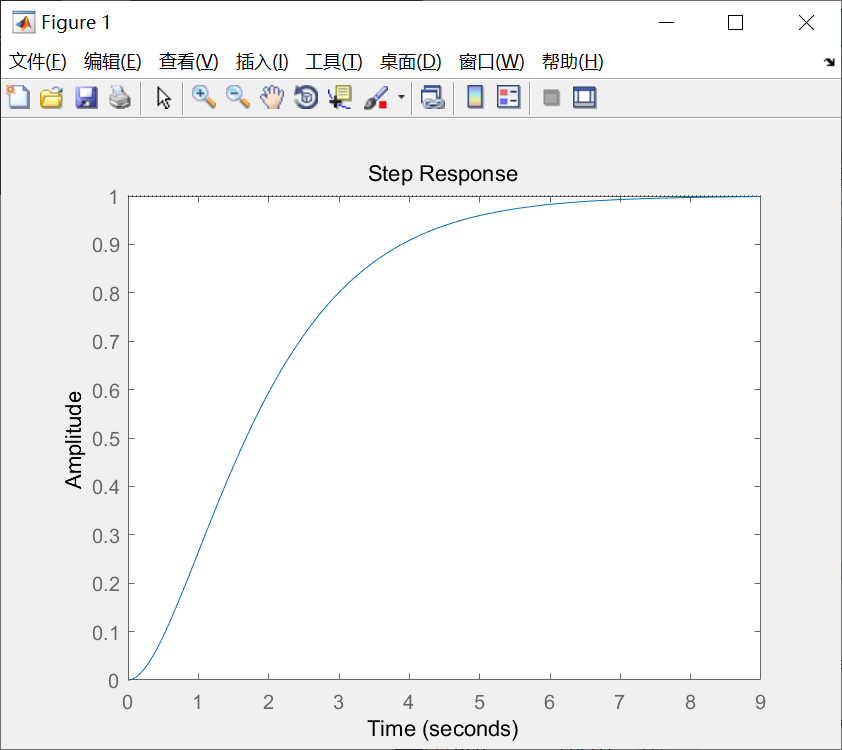
**ζ= 2:**

**tp = 8.3900**

**yss = 0.9979**

**ct = 0**

**图像见下左图：**



**ζ= 4:**

**tp =21.6553**

**yss =0.9967**

**ct =0**

**图像见上右图**

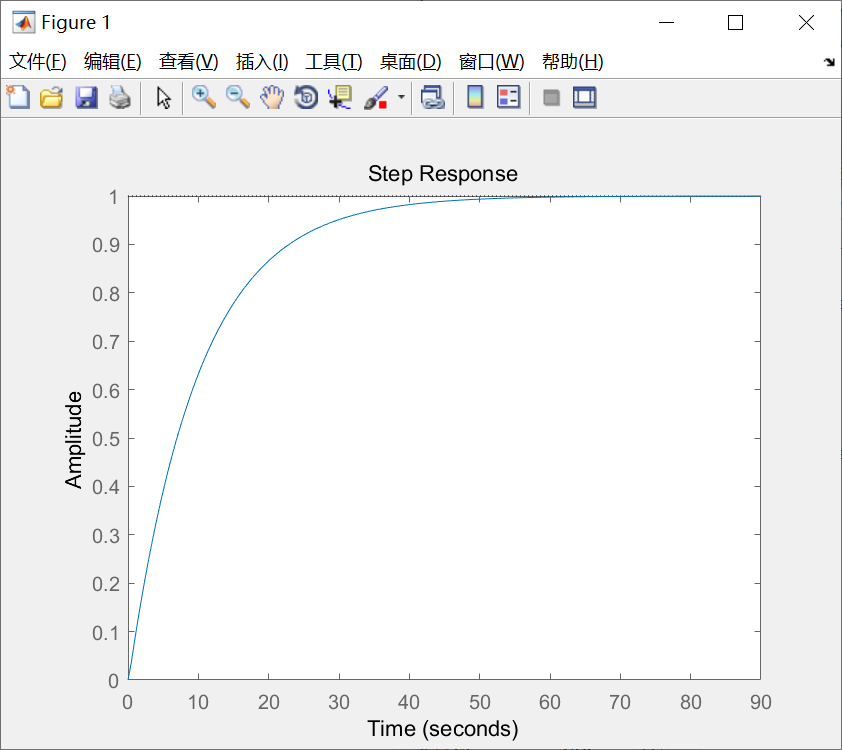
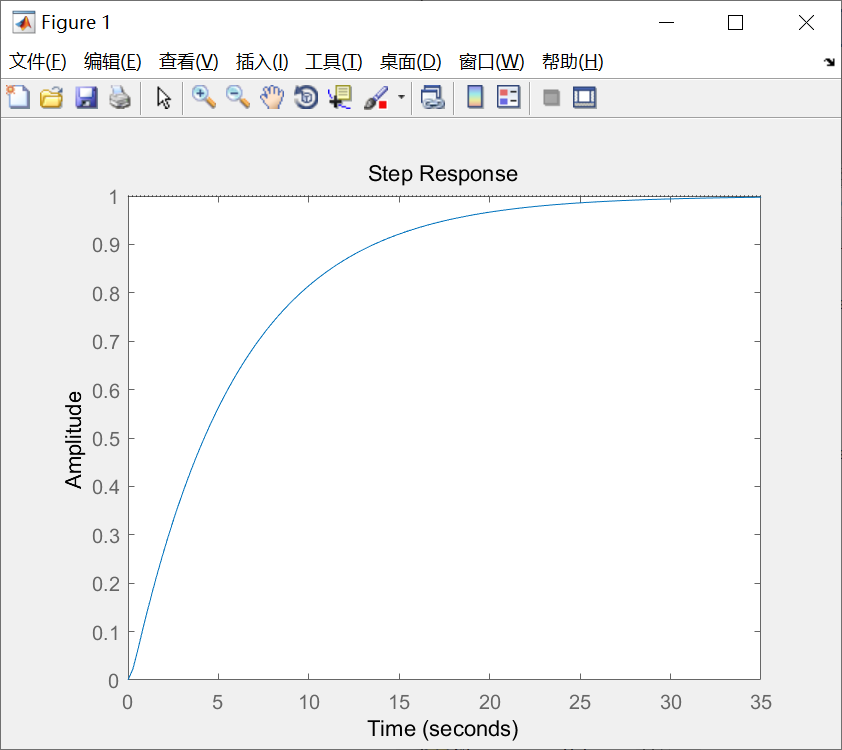
**ζ= 6:**

**tp =33.8195**

**yss =0.9969**

**ct =0**

**图像见下左图：**



**ζ= 10:**

**tp =88.8936**

**yss =0.9999**

**ct =0**

**图像见上右图**

**(2)、代码：**

a=0.55;

wn=0.1:0.1:1;

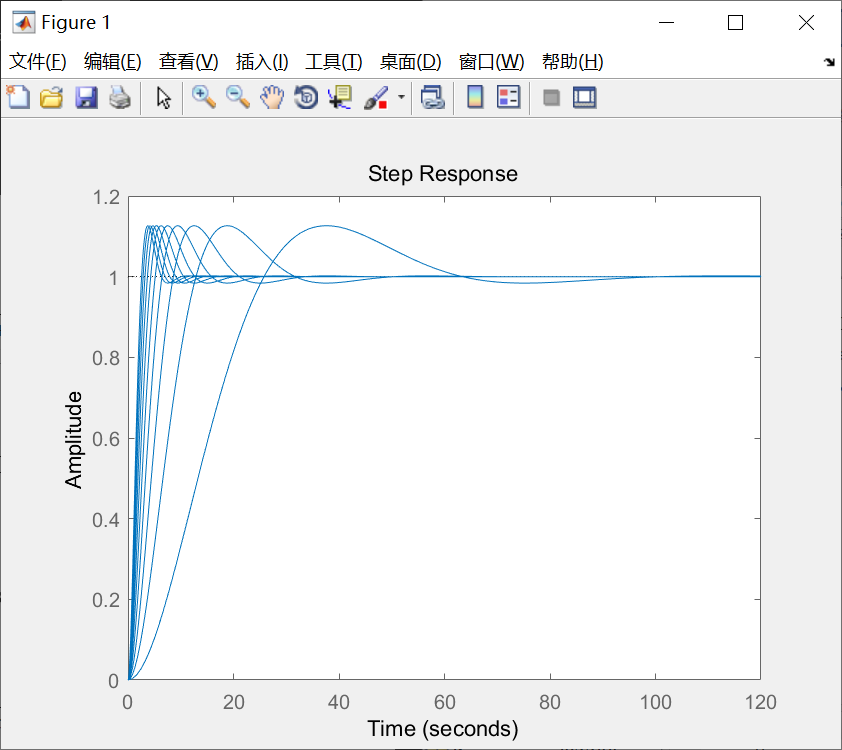
for i=1:length(wn)

sys(:,:,1,i)=tf(wn(i)^2,[1 2\*a\*wn(i) wn(i)^2]);

end

step(sys)

**图像如下：**



**第四题：**

>> num=[1500 30045];

>> den=[1 84 1545 6800 30000];

>> sys=tf(num,den);

>> [z p k]=tf2zp(num,den);

>> zpk(sys)

ans =

1500 (s+20.03)

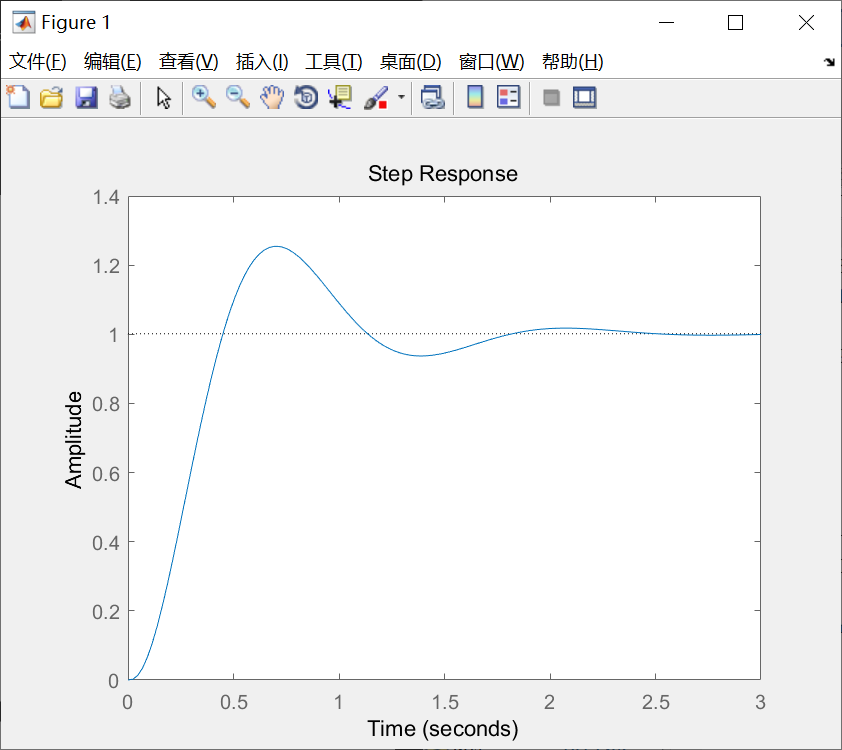
-----------------------------

(s+60) (s+20) (s^2 + 4s + 25)

Continuous-time zero/pole/gain model.

>> step(sys)

**图像如下：**



**观察可知系统闭环传递函数有一个零点(s+20.03)和一个极点(s+20)距离很近，远离其它零极点。对消此零极点得**

**1500**

**-------------------------------------**

**(s+60) (s^2 + 4s + 25)**

**将高阶系统近似为一个二阶系统：**

**25**

**-------------------------------------**

**(s^2 + 4s + 25)**

**绘制其图像，代码如下：**

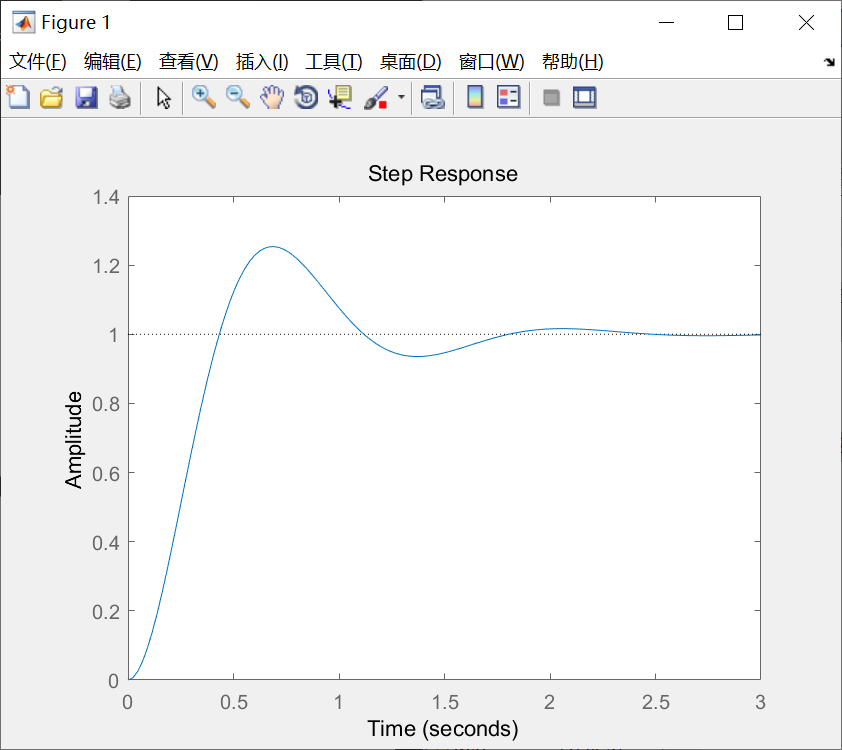
>> num=[25];

>> den=[1 4 25];

>> sys=tf(num,den);

>> step(sys)

**图像如下：**



**显然，对比可知近似前后对于系统影响可以不计，便于计算**

**第五题：**

**代码：**

num1=[20];

>> den1=[1 2 0];

>> sys1=tf(num1,den1);

>> num2=[0.2 0];

>> den2=[ 0 1 ];

>> sys2=tf(num2,den2);

>> sys\_inner=feedback(sys1,sys2);

>> sys\_outer=feedback(sys\_inner,1);

>> sys\_outer=feedback(sys\_inner,1)

sys\_outer =

20

--------------

s^2 + 6 s + 20

Continuous-time transfer function.

>> den=[1 6 20];

>> roots(den)

ans =

-3.0000 + 3.3166i

-3.0000 - 3.3166i

>> pzmap(sys\_outer);

>> grid on;

>> num=20;

>> den=[1 6 20];

>> [y t x]=step(num,den)

>> y\_stable=1;

>> max\_res=max(y);

>> signma=(max\_res-y\_stable)

signma =

0.0583

>> for i=1:length(y)

if y(i)>y\_stable

break;

end

end

>> tr=x(i)

tr =

0.7061

>> [max\_res,index]=max(y);

>> tp=x(index)

tp =

0.9517

>> for i=1:length(y)

if max(y(i:length(y)))<=1.02\*y\_stable

if min(y(i:length(y)))>=0.98\*y\_stable

break;

end

end

end

>> ts=x(i)

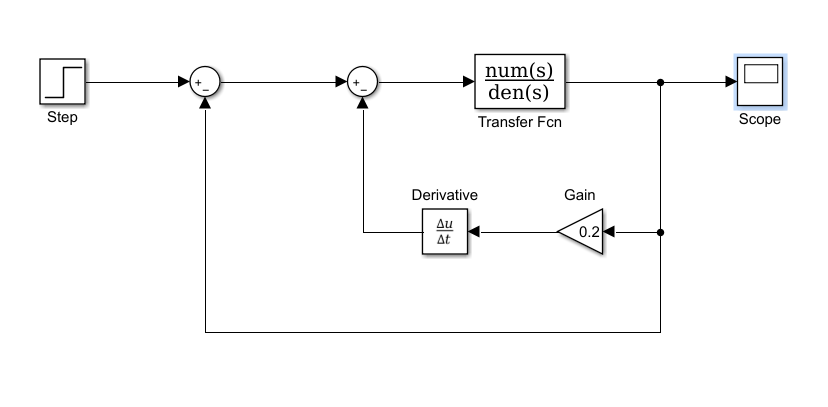
ts =

1.3508

>> dcg=dcgain(num,den)

dcg = 1

**结构图如下：**



**图像如下：**

