

求传递函数 num1=[1];den1=[0.5 1];num2=[1];den2=[1 2 2];num3=[1];den3=[1 1];[numc,denc]=feedback(num2,den2,num3,den3);[nums,dens]=series(num1,den1,numc,denc);[numt,dent]=cloop(nums,dens,-3);G=tf(numt,dent)

求零点极点增益 [z,p,k]=tf2zp(numt,dent)

绘制系统的零极点图figure(1);G1=zpk(G);z=G1.z;p=G1.p;Z=z{:};P=p{:};k=G1.k;pzmap(G);pzmap(G1);grid on

绘制系统的奈奎斯特曲线 figure(2);nyquist(G);grid on

状态空间模型 [A,B,C,D]=tf2ss(numt,dent)

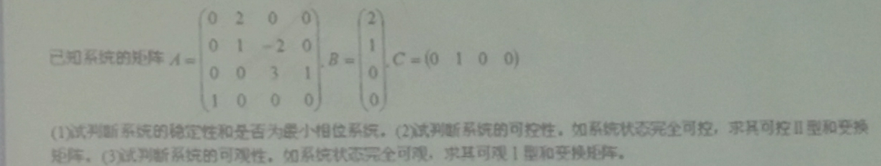
程序：num1=[1];den1=[0.5 1];num2=[1];den2=[1 2 2];num3=[1];den3=[1 1];[numc,denc]=feedback(num2,den2,num3,den3);[nums,dens]=series(num1,den1,numc,denc);[numt,dent]=cloop(nums,dens,-3);G=tf(numt,dent)

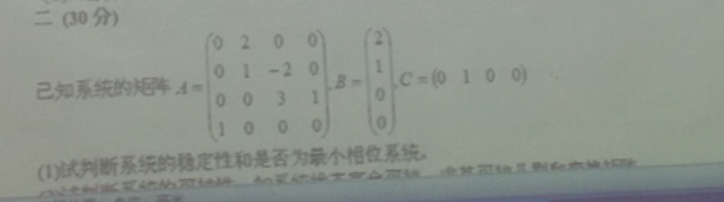
[z,p,k]=tf2zp(numt,dent)

figure(1);G1=zpk(G);z=G1.z;p=G1.p;Z=z{:};P=p{:};k=G1.k;pzmap(G);pzmap(G1);grid

figure(2);nyquist(G);grid

[A,B,C,D]=tf2ss(numt,dent)





1. 判断系统的稳定性 A=[0 2 0 0;0 1 -2 0;0 0 3 1;1 0 0 0];B=[2;1;0;0];C=[0 1 0 0];D=[0];[num,den]=ss2tf(A,B,C,D,1);[z,p]=tf2zp(num,den)

for i=1;4

if(real(p(i))>0)

disp('系统不稳定')

else disp('系统稳定')

end

end

for x=1;4

if(real(z(i))>0)

disp('系统不是最小相位系统')

else disp('系统是最小相位系统')

end

end

（2）判断系统的可控性 M=ctrb(A,B);RM=rank(M)

可控二型[Ac2,Bc2,Cc2,Dc2]=ss2ss(A,B,C,D,inv(M))

(3)判断系统的可观性 N=obsv(A,C);RN=rank(N)

[Ao1,Bo1,Co1,Do1]=ss2ss(A,B,C,D,N)

A=[0 2 0 0;0 1 -2 0;0 0 3 1;1 0 0 0];

B=[2;1;0;0];

C=[0 1 0 0];

D=[0];

[num,den]=ss2tf(A,B,C,D,1);%构建函数

[z,p]=tf2zp(num,den);%提取零极点

%（第一问）判断稳定性

ss=find(real(p) > 0);

tt=length(ss);

if(tt > 0)

disp('系统不稳定')

else

disp('系统稳定')

end

%判断最小相位

xx=find(real(z) > 0);

yy=length(xx);

if(yy>0)

disp('系统不是最小相位系统')

else

disp('系统是最小相位系统')

end

%（第二问）判断可控性

ctrl\_matrix=ctrb(A,B);%检验控制系统的可控性

if(rank(ctrl\_matrix)==3)%根据题意，可知能控矩阵的秩为3（于A满秩相同），系统完全可控p270)

disp('系统完全可控！')

[A,B,C,D]=ss2ss(A,B,C,D,inv(T))%p263可控2型

else

disp('系统不完全可控！')

end

%（第三问）判断可观性

observe\_matrix=obsv(A,C);%检验控制系统的可观性

if(rank(observe\_matrix)==3)%根据题意，可知能控矩阵的秩为3（于A满秩相同），系统完全可观p270

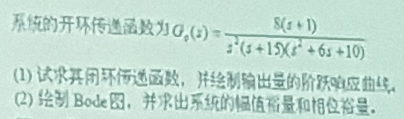
disp('系统完全可观！')

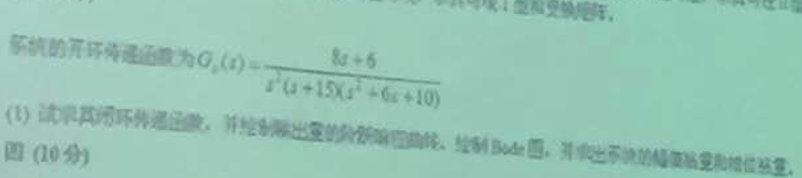
[A,B,C,D]=ss2ss(A,B,C,D,T)%p264可观1型

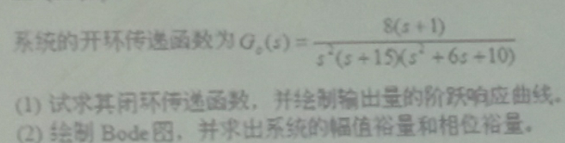
else

disp('系统不完全可观！')

end







1. 计算传递函数numc=[8,8];denc=conv([1,0,0],conv([1,15],[1,6,10]));[numc,denc]=cloop(numc,denc,-1);GC=tf(numc,denc)

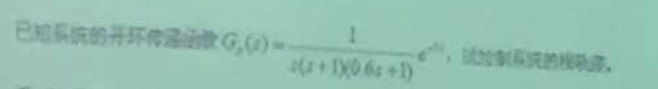
绘制阶跃响应曲线 numc=[8,8];denc=conv([1,0,0],conv([1,15],[1,6,10]));figure(1);step(numc,denc);grid on

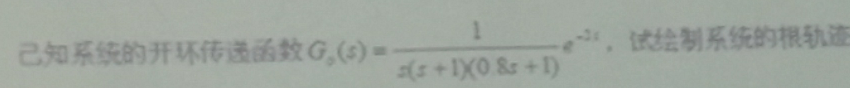
1. 绘制伯德图 figure(2);bode(numc,denc);grid on
2. 求稳定余量 [Gm,Pm]=margin(GC)
3. 程序 numc=[8,8];denc=conv([1,0,0],conv([1,15],[1,6,10]));[numc,denc]=cloop(numc,denc,-1);GC=tf(numc,denc)

numc=[8,8];denc=conv([1,0,0],conv([1,15],[1,6,10]));figure(1);step(numc,denc);grid on

figure(2);bode(numc,denc);grid on

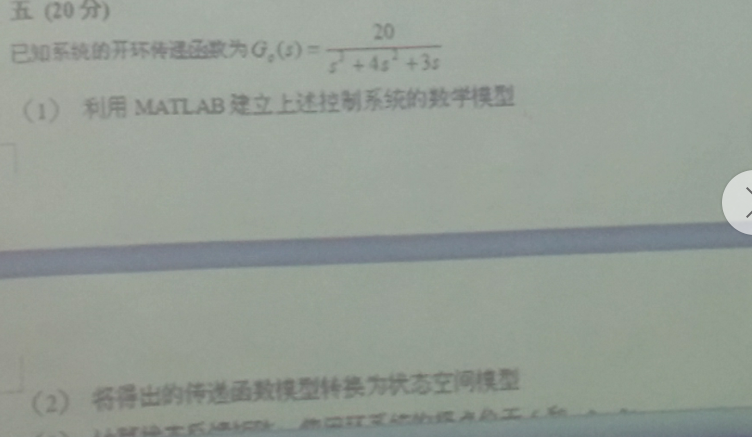
[Gm,Pm]=margin(GC)

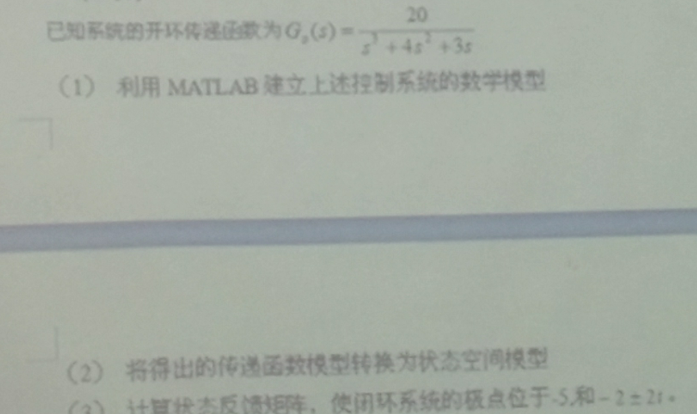




pade(T,N)用于绘制N阶传递函数中exp(-T\*s)延迟的幅值及相角，

根轨迹： num=[1];den=conv([1],conv([1,1],[0.6,1]));sys1=tf(num,den);[np,dp]=pade(2,3);sys=sys1\*tf(np,dp);rlocus(sys);title('时滞系统根轨迹图')





1. 求传递函数 num=[20];den=[1 4 3 0];[numc,denc]=cloop(num,den,-1);sys=tf(numc,denc)

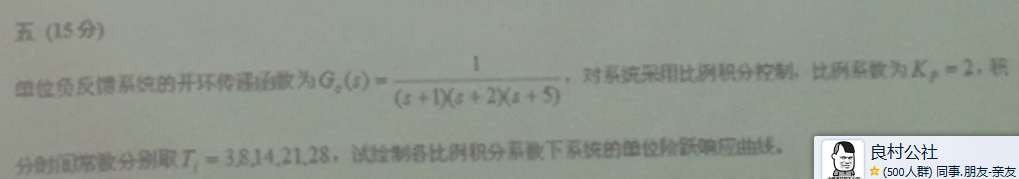
（2）状态空间 [A1,B1,C1,D1]=tf2ss(num,den)

（3）状态反馈矩阵P=[-5,-2+j\*2,-2-2\*j];K=acker(A1,B1,P)

程序 num=[20];den=[1 4 3 0];[numc,denc]=cloop(num,den,-1);sys=tf(numc,denc)

[A1,B1,C1,D1]=tf2ss(num,den)

P=[-5,-2+j\*2,-2-2\*j];K=acker(A1,B1,P)



程序 G=tf(1,conv(conv([1,1],[1,2]),[1,5]));Kp=2;ti=[3,8,14,21,28];

for i=1:5

G1=tf([Kp,Kp/ti(i)],[1,0]);

sys=feedback(G1\*G,1);

step(sys);hold on

end

gtext('ti=3');gtext('ti=8');gtext('ti=14');gtext('ti=21');gtext('ti=28')

