北京理工大學
Beijing Institute of Technology

本科实验报告

实验名称： 控制理论基础实验

课程名称:	控制理论基础 B	实验时间:	2018.4.24-5.3
任课教师:	王卫江	实验地点:	图书馆机房
实验教师:	王卫江	实验类型:	■ 原理验证 □ 综合设计 □ 自主创新
学生姓名:	施念		
学号/班级:	1120161302/05011609	组　　号:	
学　　院:	信息与电子学院	同组搭档:	
专　　业:	电子信息工程	成　　绩:	

信息与电子学院
SCHOOL OF INFORMATION AND ELECTRONICS

目录

实验 1 控制系统的模型建立

一、实验目的

1. 掌握利用 MATLAB 建立控制系统模型的方法。
2. 掌握系统的各种模型表述及相互之间的转换关系。
3. 学习和掌握系统模型连接的等效变换。

二、实验原理

1. 系统模型的 MATLAB 描述

系统的模型描述了系统的输入、输出变量以及内部各变量之间的关系，表征一个系统的模型有很多种，如微分方程、传递函数模型、状态空间模型等。这里主要介绍系统传递函数（TF）模型、零极点增益（ZPK）模型和状态空间（SS）模型的 MATLAB 描述方法。

1）传递函数（TF）模型

传递函数是描述线性定常系统输入-输出关系的一种最常用的数学模型，其表达式一般为

$$G(s)=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s^1+b_0}{a_n s^n+a_{n-1}s^{n-1}+\cdots+a_1 s^1+a_0}$$

在 MATLAB 中，直接使用分子分母多项式的行向量表示系统，即

num = [bm, bm-1, … b1, b]

den = [an, an-1, … a1, a0]

调用 tf 函数可以建立传递函数 TF 对象模型，调用格式如下：

Gtf = tf(num,den) Tfdata

函数可以从 TF 对象模型中提取分子分母多项式，调用格式如下：

[num,den] = tfdata(Gtf) 返回 cell 类型的分子分母多项式系数

[num,den] = tfdata(Gtf,'v') 返回向量形式的分子分母多项式系数

2）零极点增益（ZPK）模型

传递函数因式分解后可以写成

$$G(s)=\frac{k(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

式中， $z1,z2\cdots zm$ 称为传递函数的零点，$p1,p2\cdots pn$ … 称为传递函的极点，k 为传递系数（系统增益）。

在 MATLAB 中，直接用[z,p,k]矢量组表示系统，其中 z，p，k 分别表示系统的零极点及其增益，即：

z=[$z1,z2\cdots zm$];

p=[$p1,p2\cdots pn$];

k=[k];

调用 zpk 函数可以创建 ZPK 对象模型，调用格式如下：

G= $zpk(z,p,k)$

同样，MATLAB 提供了 zpkdata 命令用来提取系统的零极点及其增益，调用格式如下：

[z,p,k] = $zpkdata(Gzpk)$ 返回 cell 类型的零极点及增益

[z,p,k] = $zpkdata$ ($Gzpk$,’ v’) 返回向量形式的零极点及增益

函数 pzmap 可用于求取系统的零极点或绘制系统得零极点图，调用格式如下： $pzmap(G)$ 在复平面内绘出系统模型的零极点图。

[p,z] = $pzmap(G)$ 返回的系统零极点，不作图。

3）状态空间（SS）模型

由状态变量描述的系统模型称为状态空间模型，由状态方程和输出方程组成：

$$\begin{cases}\dot{x}=Ax+Bu\\ y=Cx+Du\end{cases}$$

其中：x 为 n 维状态向量；u 为 r 维输入向量； y 为 m 维输出向量； A 为 n×n 方阵，称为系统矩阵； B 为 n×r 矩阵，称为输入矩阵或控制矩阵；C 为 m×n 矩阵，称为输出矩阵； D 为 m×r 矩阵，称为直接传输矩阵。

在 MATLAB 中，直接用矩阵组[A,B,C,D]表示系统，调用 ss 函数可以创建 ZPK 对象模型，调用格式如下：

Gss = ss(A,B,C,D)

同样，MATLAB 提供了 ssdata 命令用来提取系统的 A、B、C、D 矩阵，调用格式如下：

[A,B,C,D] = ssdata (Gss) 。它返回系统模型的 A、B、C、D 矩阵。

4）三种模型之间的转换

上述三种模型之间可以互相转换，MATLAB 实现方法如下

TF 模型→ZPK 模型：zpk(SYS)或 tf2zp(num,den)

TF 模型→SS 模型：ss(SYS)或 tf2ss(num,den)

ZPK 模型→TF 模型：tf(SYS)或 zp2tf(z,p,k)

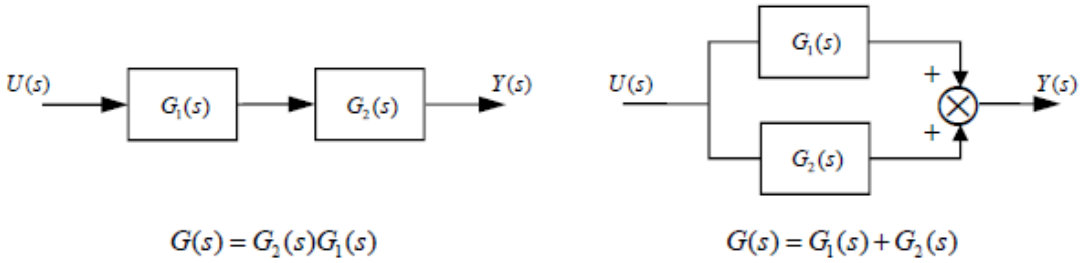
ZPK 模型→SS 模型：ss(SYS)或 zp2ss(z,p,k)

SS 模型→TF 模型：tf(SYS)或 ss2tf(A,B,C,D)

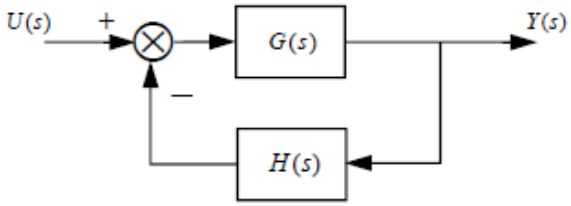
SS 模型→ZPK 模型：zpk(SYS)或 ss2zp(A,B,C,D)

2. 系统模型的连接

在实际应用中，整个控制系统是由多个单一的模型组合而成，基本的组合方式有串联连接、并联连接和反馈连接。图 1-2 分别为串联连接、并联连接和反馈连接的结构框图和等效总传递函数。



（a）串联系统　　（b）并联系统



（c）反馈连接

图 1-2 串联连接、并联连接和反馈连接

在 MATLAB 中可以直接使用“*”运算符实现串联连接，使用“＋”运算符实现并联连接。反馈系统传递函数求解可以通过命令 feedback 实现，调用格式如下：

T = feedback(G,H)

T = feedback(G,H,sign)

其中，G 为前向传递函数，H 为反馈传递函数；当 sign = +1 时，GH 为正反馈系统传递函数；当 sign = -1 时，GH 为负反馈系统传递函数；默认值是负反馈系统。

三、实验内容

1.已知控制系统的传递函数如下

$$G(s)=\frac{2s^2+18s+40}{s^3+5s^2+8s+6}$$

试用 MATLAB 建立系统的传递函数模型、零极点增益模型及系统的状态空间方程模型，并绘制系统零极点图。

代码及结果分析：

代码：

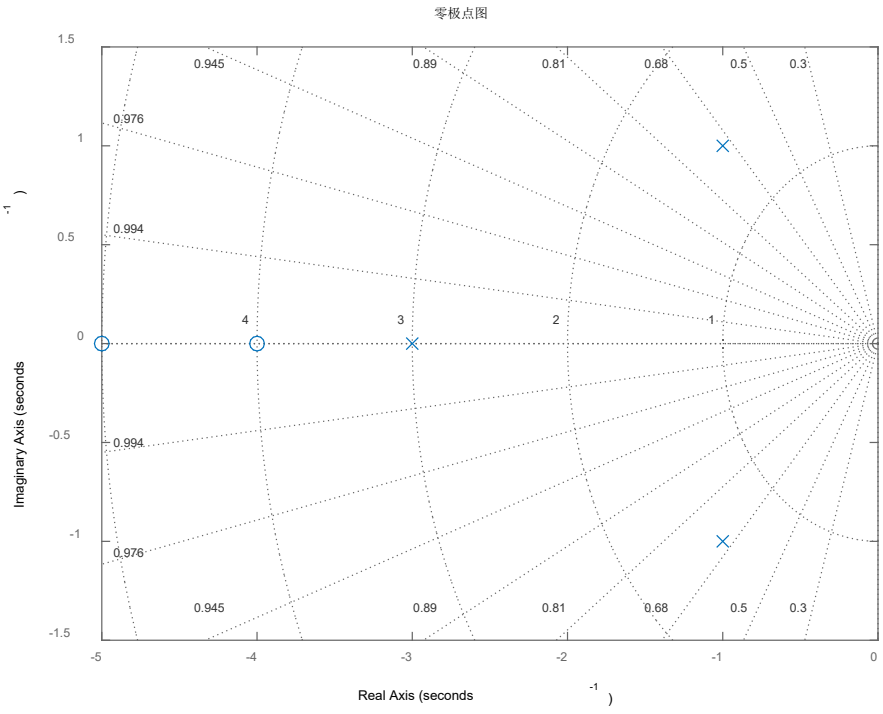
```
num = [2 18 40];
den = [1 5 8 6];
Gtf = tf(num,den) %系统传函
Gzpk = zpk(Gtf) %零极点增益
pzmap(Gzpk) %绘制零极点
title('零极点图');
grid on;
Gss = ss(Gtf) %空间方程模型
```

结果：

```
Gtf =
    2 s^2 + 18 s + 40
  --------------------
  s^3 + 5 s^2 + 8 s + 6

Gzpk =
     2 (s+5) (s+4)
  --------------------
  (s+3) (s^2 + 2s + 2)

Gss =

  A =
         x1     x2     x3
   x1    -5     -2   -1.5
   x2     4      0      0
```

```
  x3       0       1       0

B =
       u1
  x1    4
  x2    0
  x3    0

C =
          x1       x2       x3
  y1     0.5    1.125      2.5

D =
       u1
  y1    0
```

零极点图：



2. 已知控制系统的状态空间方程如下

$$\dot{x}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -1 & -2 & -3 & -4\end{bmatrix}x+\begin{bmatrix}0\\0\\0\\1\end{bmatrix}u$$
$$y=\begin{bmatrix}10 & 2 & 0 & 0\end{bmatrix}x$$

试用 MATLAB 建立系统的传递函数模型、零极点增益模型及系统的状态空间方程模型，并绘制系统零极点图。

代码及结果分析：

代码：

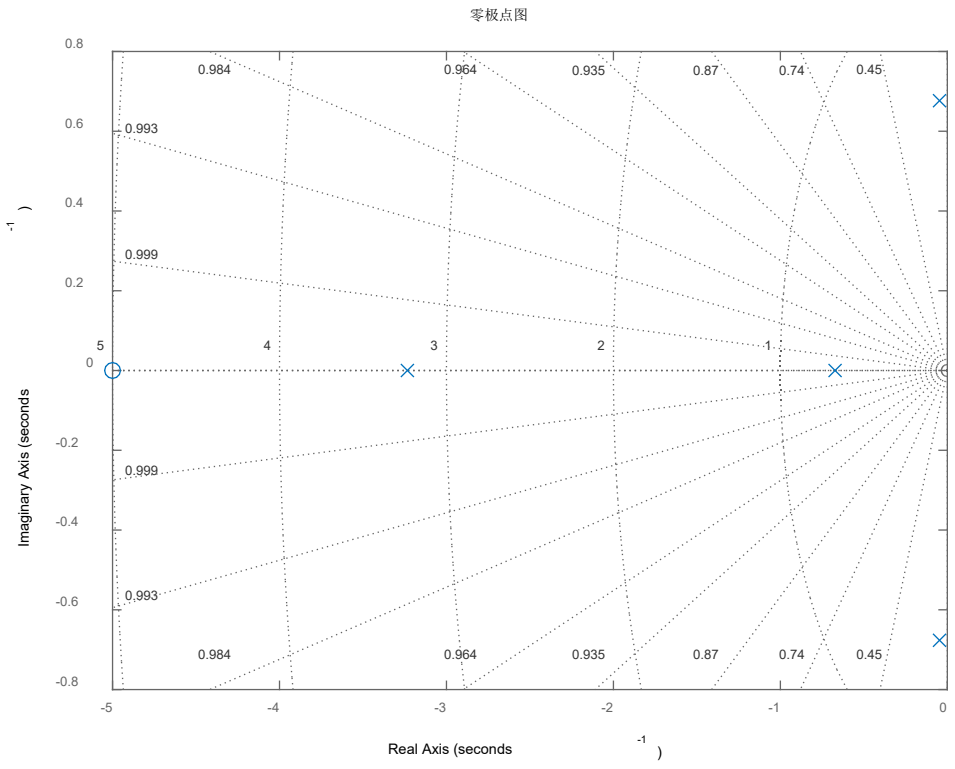
```
A = [0 1 0 0;0 0 1 0;0 0 0 1;-1 -2 -3 -4];
B = [0;0;0;1];
C = [10 2 0 0];
D = [0];
Gss = ss(A,B,C,D)
Gtf = tf(ss)
Gzpk = zpk(Gss)
pzmap(Gzpk);
grid on;
title('零极点图');
grid on;
```

结果：

```
Gss =

  A =
       x1  x2  x3  x4
   x1   0   1   0   0
   x2   0   0   1   0
   x3   0   0   0   1
   x4  -1  -2  -3  -4

  B =
       u1
   x1   0
   x2   0
   x3   0
   x4   1

  C =
       x1  x2  x3  x4
   y1  10   2   0   0
```

```
D =
       u1
  y1    0

Gtf =

              2 s + 10
  ----------------------------
  s^4 + 4 s^3 + 3 s^2 + 2 s + 1

Gzpk =

                        2 (s+5)
  ---------------------------------------------
  (s+3.234) (s+0.6724) (s^2 + 0.0936s + 0.4599)
```

零极点图：



3.已知三个系统的传递函数分别为

$$G_1(s)=\frac{2s^2+6s+5}{s^3+4s^2+5s+2}$$

$$G_2(s)=\frac{s^2+4s+1}{s^3+9s^2+8s}$$

$$G_3(s)=\frac{5(s+3)(s+7)}{(s+1)(s+4)(s+6)}$$

试用 MATLAB 求上述三个系统串联后的总传递函数。

代码及结果分析：

代码：

```
a1 = [2 6 5];
b1 = [1 4 5 2];
a2 = [1 4 1];
b2 = [1 9 8];
a3 = [1 10 21]*5;
b3 = [1 11 34 24];

Gtf1 = tf(a1,b1);
Gtf2 = tf(a2,b2);
Gtf3 = tf(a3,b3);

Gtf = Gtf1*Gtf2*Gtf3
Gzpk = zpk(Gtf)
```

结果：

```
Gtf =

          10 s^6 + 170 s^5 + 1065 s^4 + 3150 s^3 + 4580 s^2 + 2980 s + 525
  ------------------------------------------------------------------------

  s^8 +24 s^7+ 226 s^6+ 1084 s^5+ 2905 s^4 +4516 s^3 + 4044 s^2 + 1936 s + 384


Gzpk =
```

$$\frac{10\ (s+7)\ (s+3.732)\ (s+3)\ (s+0.2679)\ (s^2+3s+2.5)}{(s+8)\ (s+6)\ (s+4)\ (s+2)\ (s+1)^4}$$

4.已知如图 E2-1 所示的系统框图

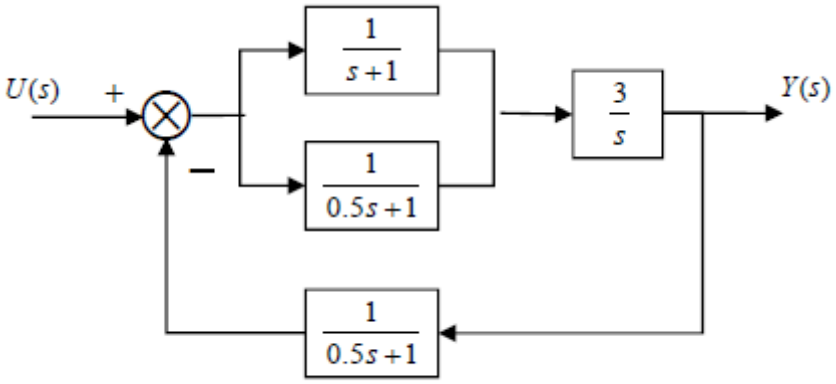


图 E2-1

试用 MATLAB 求该系统的闭环传递函数。

代码及结果分析：

代码：

```
clear all;
a_1 = [1];
b_1 = [1 1];
a_2_4 = [1];  %2 and 4 is same
b_2_4 = [0.5 1];
a_3 = [3];
b_3 = [1 0];

Gtf1 = tf(a_1,b_1);
Gtf2 = tf(a_2_4,b_2_4);
Gtf3 = tf(a_3,b_3);
H = tf(a_2_4,b_2_4);

Gtf = (Gtf1+Gtf2)*Gtf3;
F = feedback(Gtf,H,-1) % -1 -> negative
```

结果：

```
F =
```

```
            2.25 s^2 + 7.5 s + 6
-------------------------------------
0.25 s^4 + 1.25 s^3 + 2 s^2 + 5.5 s + 6
```

5. 已知如图 E2-2 所示的系统框图

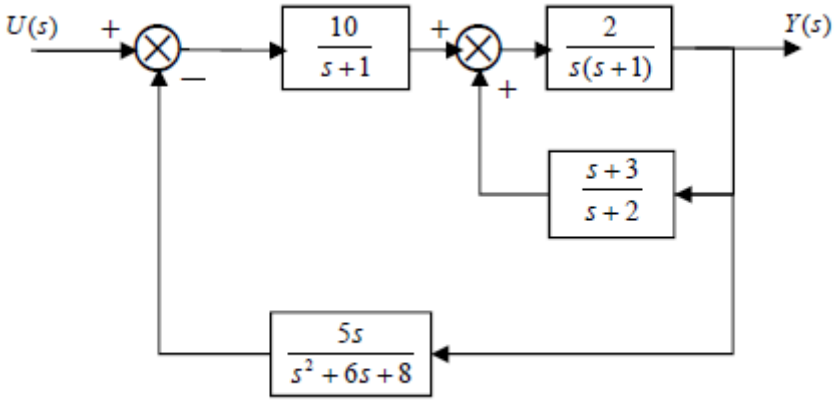


图 E2-2

试用 MATLAB 求该系统的闭环传递函数。

代码及结果分析：

代码：

```
a_1 = [10];
b_1 = [1 1];
a_2 = [2];
b_2 = [1 1 0];
a_3 = [1 3];
b_3 = [1 2];
a_4 = [5 0];
b_4 = [1 6 8];

G1 = tf(a_1,b_1);
G2 = tf(a_2,b_2);
H1 = tf(a_3,b_3);
H2 = tf(a_4,b_4);

F1 = feedback(G2,H1,1) % 支路上的正反馈闭环传函
G = G1*F1; % 总体框图开环传函
F = feedback(G,H2,-1) % 总闭环传函
```

结果：

正反馈网络

```
F1 =
```

```
     2 s + 4
--------------
s^3 + 3 s^2 - 6
```

总负反馈网络：

```
F =
                20 s^3 + 160 s^2 + 400 s + 320
-------------------------------------------------
s^6 + 10 s^5 + 35 s^4 + 44 s^3 + 82 s^2 + 116 s - 48
```

分析：

需要注意的是 feedback（）的第三个参数默认为-1，即默认为负反馈，所以当反馈为正反馈时，应注意将第三个参数设置为 1。

四．实验收获与心得

此次实验主要学习了系统传函模型，零极点模型以及状态空间模型这三种模型（tf，ss,zpk）的建立，同时也学习了这三种模型之间的相互转化。除此之外，也学到了一个绘制零极点图形的函数 pzmap。最后，将学到的这些方法应用到系统模型的连接中（串联，并联以及反馈连接），运用 MATLAB 很好的解决问题。

实验 2 控制系统的暂态特性分析

一、实验目的

1.学习和掌握利用 MATLAB 进行系统时域响应求解和仿真的方法。

2.考察二阶系统的时间响应，研究二阶系统参数对系统暂态特性的影响。

二、实验原理

1.系统的暂态性能指标

控制系统的暂态性能指标常以一组时域量值的形式给出，这些指标通常由系统的单位阶跃响应定义出来，这些指标分别为：

（1）延迟时间 td ：响应曲线首次到达稳态值的 50%所需的时间。

（2）上升时间 tr ：响应曲线从稳态值的 10%上升到 90%所需要的时间长，对于欠阻尼系统，通常指响应曲线首次到达稳态值所需的时间。

（3）峰值时间 tp ：响应曲线第一次到达最大值的时间。

（4）调整时间 ts ：响应曲线开始进入并保持在允许的误差（±2%或±5%）范围内所需要的时间。

（5）超调量 σ：响应曲线的最大值和稳态值之差，通常用百分比表示

$$\sigma = \frac{y(t_p) - y(\infty)}{y(\infty)} \times 100\%$$

其中 y(t) 为响应曲线。

在 MATLAB 中求取单位阶跃响应的函数为 step，其使用方法如下

step(sys)在默认的时间范围内绘出系统响应的时域波形

step(sys,T)绘出系统在 0 - T 范围内响应的时域波形

step(sys,ts:tp:te)绘出系统 ts - te 范围内，以 tp 为时间间隔取样的响应波形。

[y,t]=step(„)该调用格式不绘出响应波形，而是返回响应的数值向量及其对应的时间向

量。

系统的暂态性能指标可以根据上述定义，在响应曲线上用鼠标读取关键点或通过搜索曲线对应的数值向量中关键点来确定。

2.LTIViewer 工具

在 MATLAB 中提供了线性是不变系统仿真的工具 LT Viewer，可以方便地观察系统的响应曲线和性能指标。在命令窗口中键入 litview 即可启动 LTIViewer。这里简要介绍 LTI Viewer 工具。

1)【File】菜单

Import 选项：可以从 Workspace 或文件中导入系统模型。

Export 选项：将当前窗口中的对象模型保存到 Workspace 或文件中。

Toolbox preferences 选项：属性设置功能，可以设置控制系统中得各种属性值。 Page Setup 选项：页面设置功能，可以对打印输出和显示页面进行设置。

2)【Edit】菜单

Plot Configuration 选项：对显示窗口及显示内容进行配置。

Line Style 选项：线型设置功能，可以对输出响应曲线的线型进行设置。

Viewer Preferences 选项：对当前窗口的坐标、颜色、字体、响应曲线的特性参数等属性进行设置。

3）右键菜单

在运行界面上点击鼠标右键，将会弹出一个弹出式菜单，菜单上个选项的功能分别为：

Plot Types：选择绘制的系统曲线类型，可选的类型有单位阶跃响应、单位冲击响应、波特图、奈奎斯特图、零极点图等。

System：选择需要仿真的系统。

Characteristic：系统的性能指标选项。

Grid：显示和关闭网格。

Normalize：正常显示模式。

Full View：满界面显示模式。

Properties：性能编辑器选项，可以设置画面的标题、坐标标志、坐标范围、线型、颜色、性能指标等。

三、实验内容

1.已知单位负反馈系统前向通道的传递函数为

$$G(s)=\frac{80}{s^2+2s}$$

试用 MATLAB 绘制系统的单位阶跃响应曲线。

代码及结果分析：

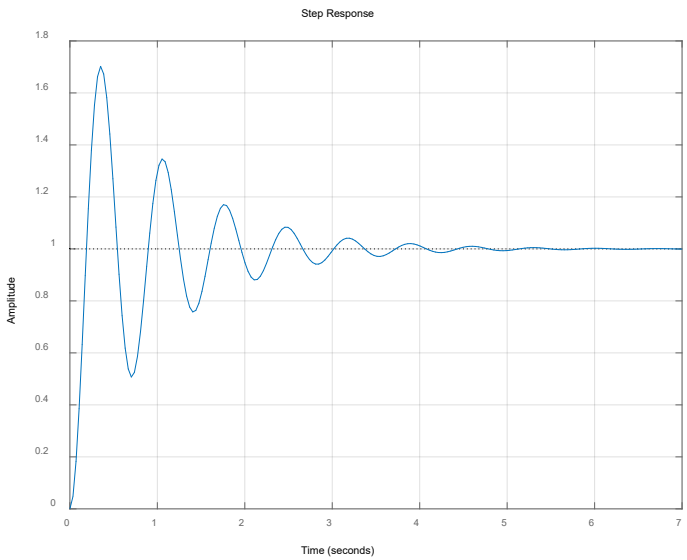
代码：

```
clear all;
a = [80];
b = [1 2 0];
G = tf(a,b);
T = feedback(G,1,-1)
step(T);
grid on;
```

结果：

```
T =
          80
  ---------------
  s^2 + 2 s + 80
```

图形：



2. 已知二阶系统

$$G(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

（1）ζ=0.6，ωn=5，试用 MATLAB 绘制系统单位阶跃响应曲线，并求取系统的暂态性能指标。

代码及结果分析：

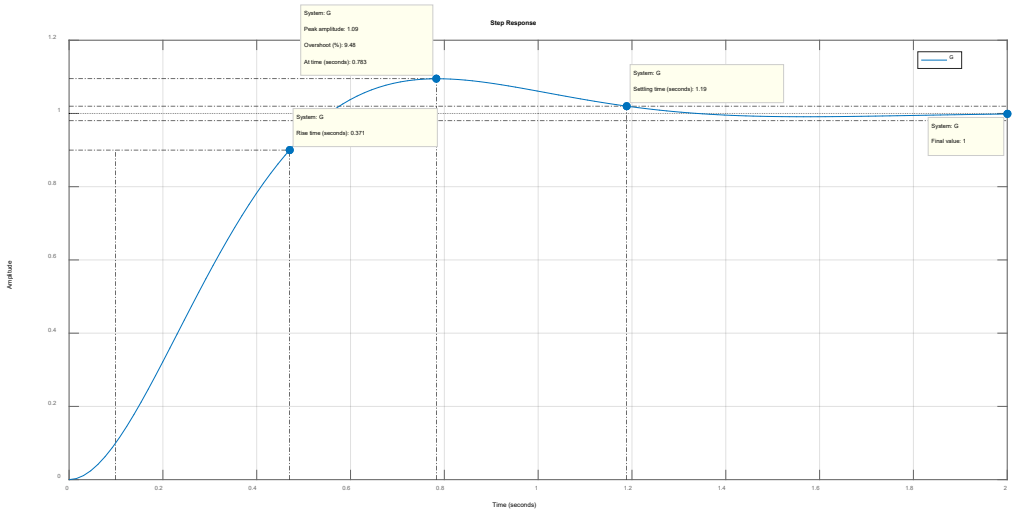
代码：

```
wn = 5;
sn = 0.6;
a = [wn^2];
b = [1 2*wn*sn wn^2];
G = tf(a,b)
step(G)
grid on;
```

结果：

```
G =
          25
  -------------
  s^2 + 6 s + 25
```

图像：



分析：

由图可知，系统的$t_p=0.785s$ $t_s=1.19s$ $t_r=0.371s$ $\sigma_p=9.48\%$ $c(t_p)=1.09$

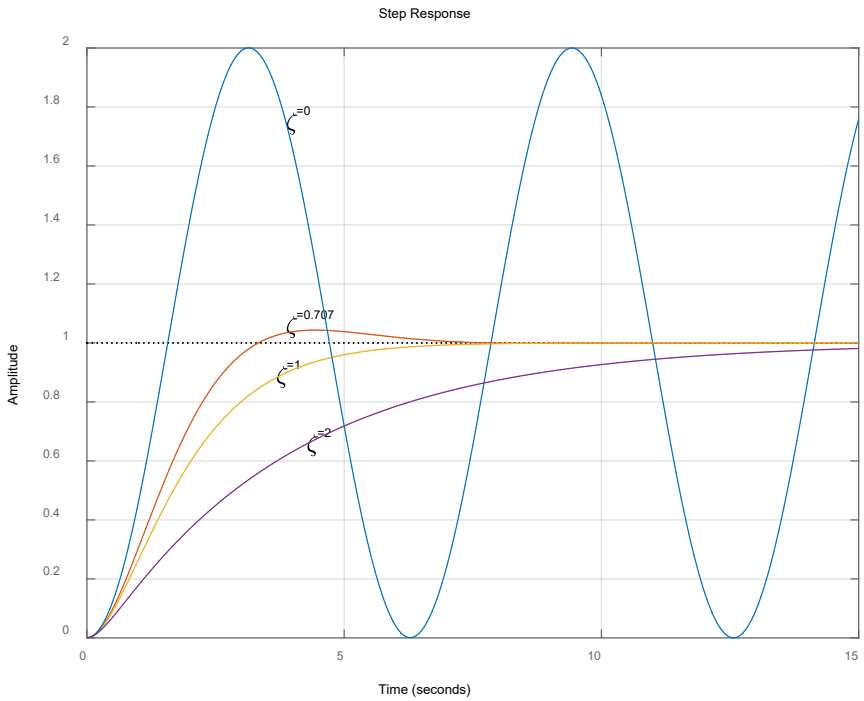
（2）ωn=1，ζ 位 0、0.707、1、2，求此系统的单位阶跃响应。

代码及结果分析：

代码：

```
clear all;
wn = 1;
sn = [0 0.707 1 2];
t = [0:0.05:15];
for i=1:length(sn)

    a = [wn^2];
    b = [1 2*wn*sn(i) wn^2];
    G = tf(a,b);
    step(G,t);
    grid on;
    hold on;
    gtext(['\zeta=' num2str(sn(i))]);
end;
hold off;
```

结果：



分析：

从上图结果可以看出，随着ζ最贱增大，频率响应逐步减弱。在ζ= 0时属于无阻尼振荡，0 <ζ< 1 时属于欠阻尼振荡，ζ> 1时属于过阻尼振荡，此时已经没有振荡发生。当ζ= 1时属于临界振荡。

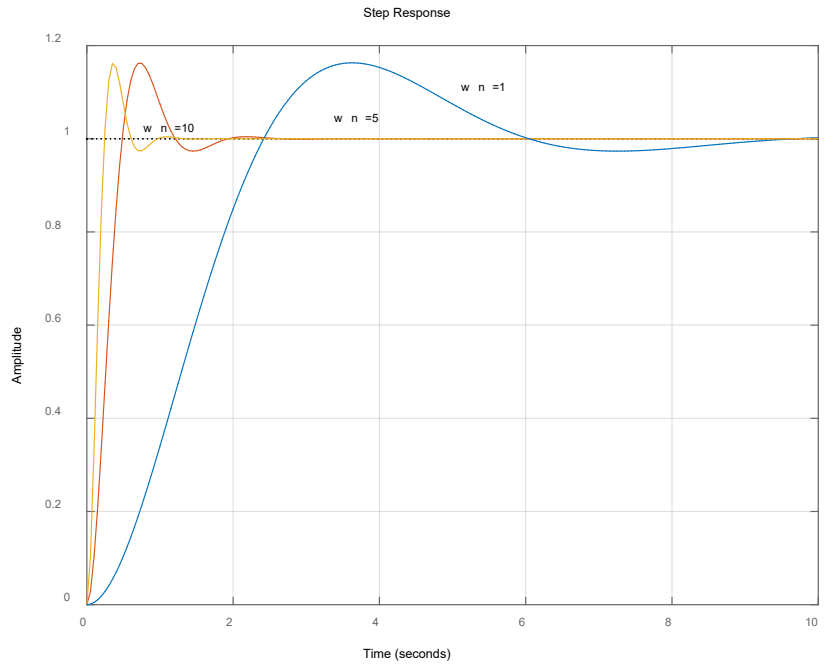
（3）ζ=0.5，ωn 为 1、5、10 时，求此系统的单位阶跃响应。

代码及结果分析：

代码：

```
clear all;
wn = [1 5 10];
sn = 0.5;
t = [0:0.05:10];
for i=1:length(wn)

    a = [wn(i)^2];
    b = [1 2*wn(i)*sn wn(i)^2];
    G = tf(a,b);
    step(G,t);
    grid on;
    hold on;
    gtext(['w{n}=' num2str(wn(i))]);
end;
hold off;
```

结果：



分析：

wn 的增大使得时间 ts、tp、tr 都减小，但是超调量、峰值却没有改变。wn 可以增强系统的快速性，但不能增强系统的稳定性。

（4）观察上述实验结果，分析这两个特征参数对系统暂态特性的影响。

从上述结果和分析可以看出ζ的增大会使 tr、tp 增加，ts 减小，超调量降低，峰值降低，振荡逐渐减小。ζ的调整会改变系统的快速性和稳定性。wn 的增大使得时间 ts、tp、tr 都减小，但是超调量、峰值却没有改变。wn 可以增强系统的快速性，但不能增强系统的稳定性。

四．实验收获与心得

通过 MATLAB 不仅可以很直观的画出系统的单位冲激响应，而且可以运用 ltiview 工具箱很好地对系统的参数进行测量。可以借助此方法帮助我们检验我们的系统是否符合我们的设定要求。

此次试验让我直观的看到了ζ和 wn 对系统各个动态指标（tr、tp、ts、σp）的影响，从而判断其对系统快速性和稳定性的影响。

除此之外，这次实验过程中让我知道，在“参数改变观察影响”的类似实验中，可以将结果放在一个图像中，便于直接比较。

实验 4 系统的频率特性分析

一、实验目的

1. 学习和掌握利用 MATLAB 绘制系统 Nyquist 图和 Bode 图的方法。

2. 学习和掌握利用系统的频率特性分析系统的性能。

二、实验原理

系统的频率特性是一种图解方法，分析运用系统的开环频率特性曲线，分析闭环系统的 性能，如系统的稳态性能、暂态性能常用的频率特性曲线有 Nyquist 图和 Bode 图。在 MATLAB 中，提供了绘制 Nyquist 图和 Bode 图的专门函数。

1.Nyquist 图

nyquist 函数可以用于计算或绘制连续时间 LTI 系统的 Nyquist 频率曲线，其使用方法如下： nyquist(sys) 绘制系统的 Nyquist 曲线。

nyquist(sys,w) 利用给定的频率向量 w 来绘制系统的 Nyquist 曲线。

[re,im]=nyquist(sys,w) 返回 Nyquist 曲线的实部 re 和虚部 im，不绘图。

2.Bode 图

bode 函数可以用于计算或绘制连续时间 LTI 系统的 Bode 图，其使用方法如下：

bode(sys) 绘制系统的 Bode 图。

bode(sys,w) 利用给定的频率向量 w 来绘制系统的 Bode 图。

[mag,phase]=bode(sys,w) 返回 Bode 图数据的幅度 mag 和相位 phase，不绘图。

3.幅度和相位裕度计算

margin 函数可以用于从频率响应数据中计算出幅度裕度、相位裕度及其对应的角频率，其使用方法如下： margin(sys)

margin(sys)

margin(mag,phase,w)

[Gm,Pm,Wcg,Wcp] = margin(sys)

[Gm,Pm,Wcg,Wcp] = margin(mag,phase,w)

其中不带输出参数时，可绘制出标有幅度裕度和相位裕度值的 Bode 图，带输出参数时，

返回幅度裕度 Gm、相位裕度 Pm 及其对应的角频率 Wcg 和 Wcp。

三、实验内容

1. 已知系统开环传递函数为

$$G(s)=\frac{1000}{(s^2+3s+2)(s+5)}$$

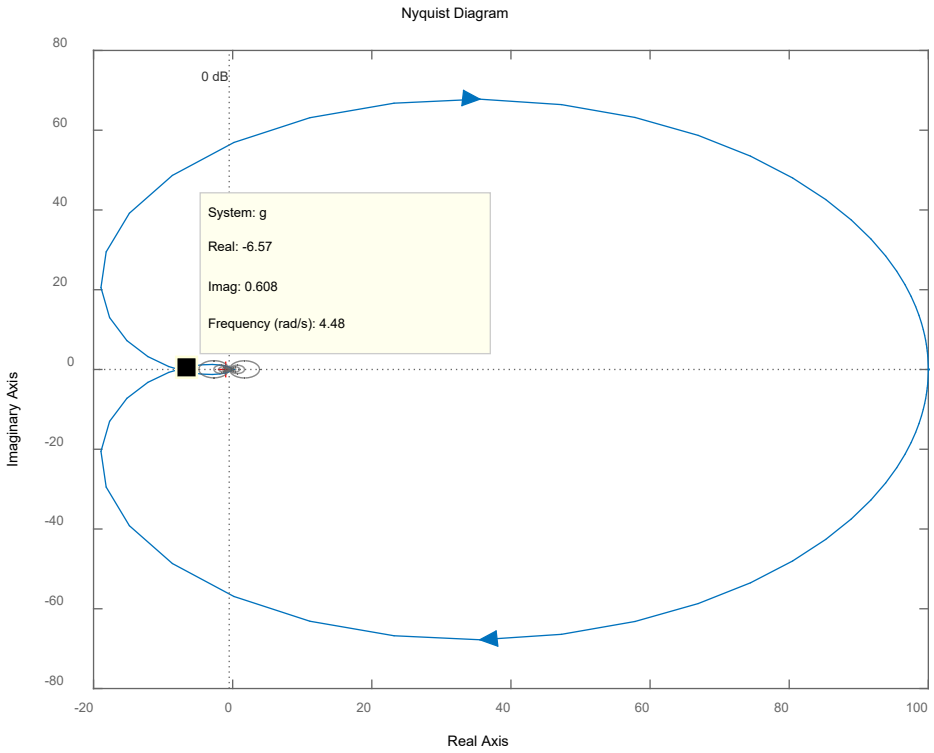
绘制系统的 Nyquist 图，并讨论其稳定性。

代码及结果：

代码：

```
p = [-1 -2 -5];
z = [];
k = 1000;
g = zpk(z,p,k);
nyquist(g);
grid on;
```

结果：



分析：

系统逆时针包围（-1，j0）0 圈，顺时针包围 2 圈，而开环无正实部极点即 P=0，所以不满足稳定判据，闭环不稳定。

2. 已知系统的开环传递函数为

$$G(s)=\frac{10\left[\left(\frac{5}{4}s\right)^2+\frac{5}{4}s+1\right]}{s^2\left(\frac{10}{3}s+1\right)\left(\frac{0.2}{3}s+1\right)\left(\frac{1}{40}s+1\right)}$$

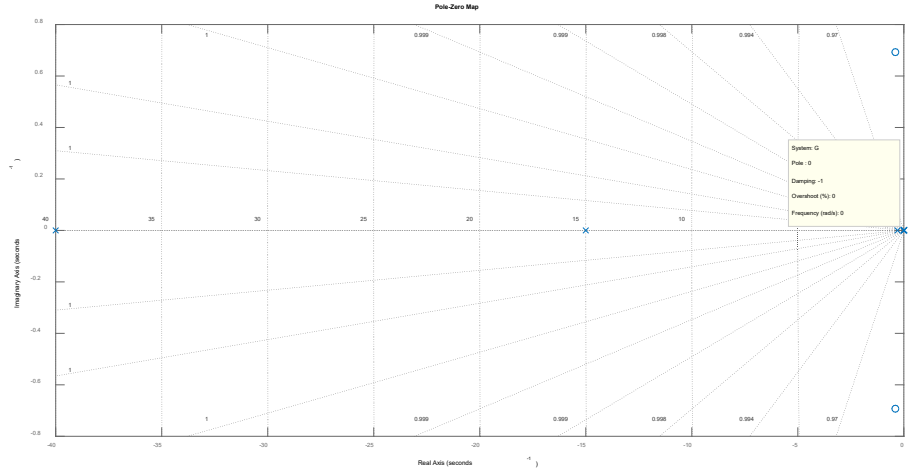
（1）绘制系统的零极点图，根据零极点分布判断系统的稳定性。

代码及结果：

代码：

```
G1 = tf([25/16 5/4 1],[1]);
G2 = zpk([],[0 0 -0.3 -15 -40],1800); %注意零极点
G = G1 * G2;
pzmap(G);
grid on;
```

结果：



分析：

由图像可以看出，其部分极点在虚轴上，故其处于临界稳定状态。

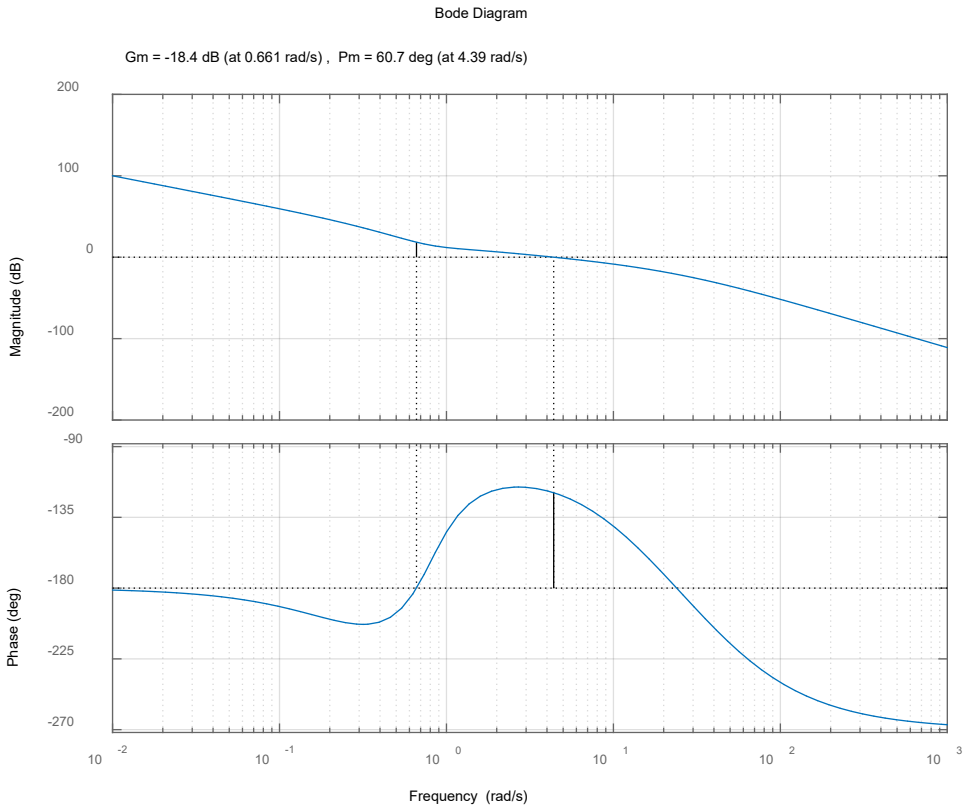
（2）绘制系统 Bode 图，求出幅度裕度和相位裕度，判断闭环系统的稳定性。

代码及结果：

代码：

```
clear all;
G1 = tf([25/16 5/4 1],[1]);
G2 = zpk([],[0 0 -0.3 -15 -40],1800);
G = G1 * G2;
[Gm, Pm, Wcg, Wcp]=margin(G)
grid on;
```

结果：



分析：

在开环幅频特性大于 0dB 的所有频段内，相频特性曲线对-180 度线的正负穿越次数之差为 0，可以判断出系统是稳定的。

3. 已知系统的开环传递函数为

$$G(s)=\frac{K}{s(s+1)(0.1s+1)}$$

分别判断当开环放大系数 K = 5 和 K = 20 时闭环系统的稳定性，并求出幅度裕度和相位裕度。

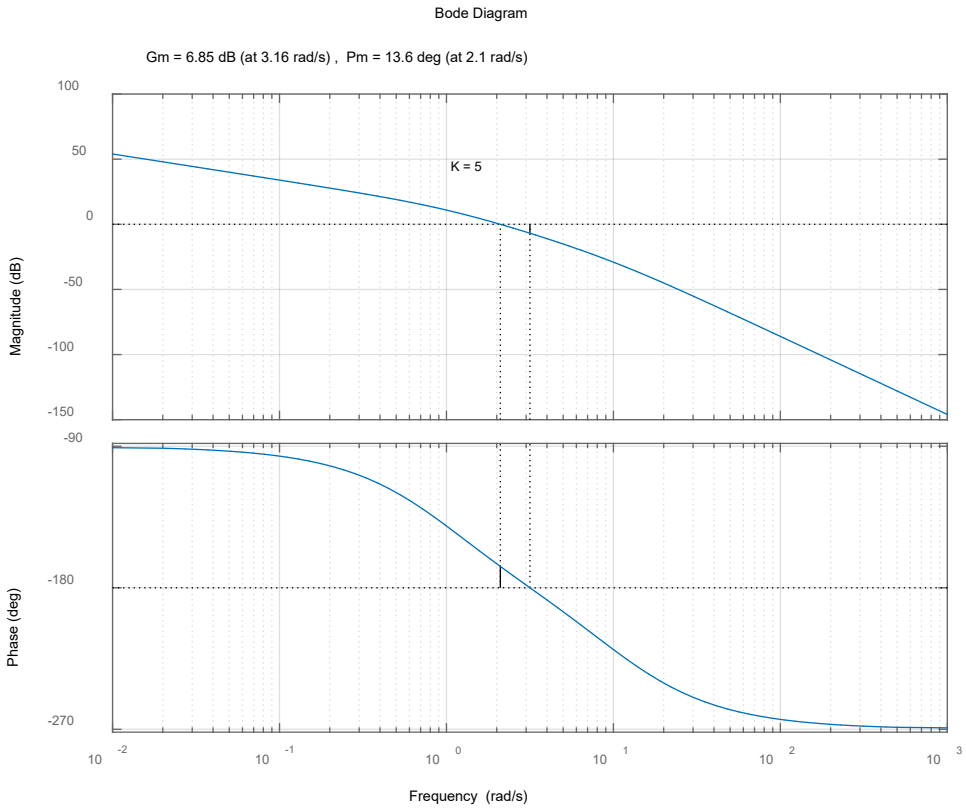
代码及结果：

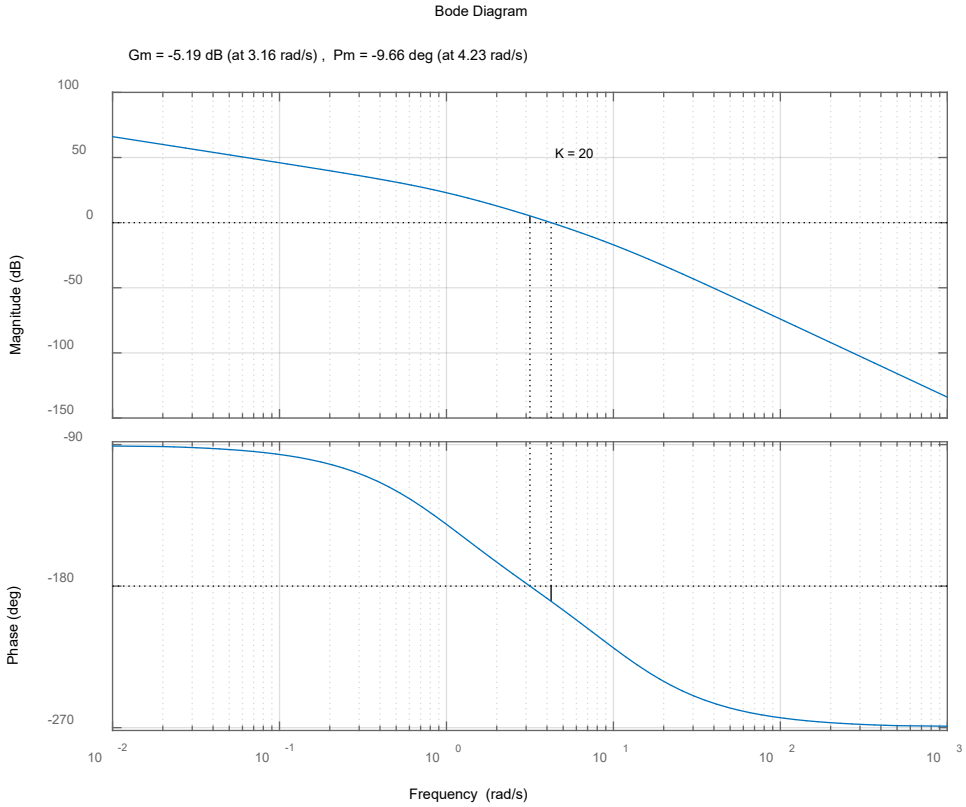
代码：

```
clear all;
for K = [5 20]
    G = zpk([],[0 -1 -10],10*K);
    figure;
    margin(G);
    gtext(['K = ' num2str(K)]);
```

```
    grid on;
end
```

结果：





分析：

由图可知，当 K=5 时，闭环稳定，当 K=20 时，闭环不稳定。

四．实验收获与心得

在学习过程中，我们就知道，实际中闭环函数往往不好求得，但是只要通过系统的开环传递函数的耐奎斯特图和波特图就通过开环函数的特性便可以简便地判定闭环稳定性。MATLAB 直接提供了 bode 图的绘制函数，同时，也可以直接求出相位裕度和幅值裕度，这对我们分析系统的稳定性无疑是很有帮助的。

除此之外，在 MATLAB 的学习过程中，在遇到问题时，我也学会了如何运用 help 函数以及 MATLAB 自带文档的查阅。

实验 6 极点配置与全维状态观测器的设计

一、实验目的

1.加深对状态反馈作用的理解。

2.学习和掌握状态观测器的设计方法。

二、实验原理

在 MATLAB 中，可以使用 acker 和 place 函数进行极点配置，使用方法如下：

K=acker(A,B,P)

K=place(A,B,P)

[K,PREC,MESSAGE]=place(A,B,P)

其中，A,B 为系统的系数矩阵，P 为配置极点，K 为反馈增益矩阵，PREC 为特征 MESSAGE 为配置中的出错信息。

三、实验内容

1.已知系统

$$\dot{x}=\begin{bmatrix}-2 & -1 & 1\\ 1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}x+\begin{bmatrix}1\\1\\1\end{bmatrix}u$$

(1) 判断系统稳定性，说明原因。

代码及结果分析：

代码：

```
clear;
A=[-2 -1 1;1 0 1;-1 0 1];
[V,D]=eig(A)
```

结果：

```
V =[
  -0.4045 + 0.5394i  -0.4045 - 0.5394i  -0.0000 + 0.0000i
   0.6742 + 0.0000i   0.6742 + 0.0000i   0.7071 + 0.0000i
  -0.2697 + 0.1348i  -0.2697 - 0.1348i   0.7071 + 0.0000i]
```

```
D =[
  -1.0000 + 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i  -1.0000 - 1.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   1.0000 + 0.0000i]
```

分析：

由结果易知系数矩阵 A 中有正实部特征值，故系统不稳定。

(2) 若不稳定，进行极点配置，期望极点：-1，-2，-3，求出状态反馈矩阵 k。

代码及结果：

代码：

```
clear;
A = [-2 -1 1; 1 0 1; -1 0 1];
B = [1; 1; 1];
P = [-1; -2; -3];
K = acker(A,B,P)
```

结果：

```
K = [-1 2 4]
```

分析：

由结果可知，反馈矩阵 K=[-1 2 4]

(3) 讨论状态反馈与输出反馈的关系，说明状态反馈为何能进行极点配置？

答：

A.

状态反馈和输出反馈是控制系统设计中两种主要的反馈策略，其意义分别为将观测到的状态和输出取做反馈量以构成反馈律，实现对系统的闭环控制，以达到期望的对系统的性能指标要求。

在经典控制理论中，一般只考虑由系统的输出变量来构成反馈律，即输出反馈。

在现代控制理论的状态空间分析方法中，多考虑采用状态变量来构成反馈律，即状态反馈。

B.

状态空间分析法所采取的模型为状态空间模型，其状态变量可完全描述系统内部动态特性。

设系统的状态空间表达式为:

$$\dot{x} = Ax + Bu$$

则系统的特征方程为：

$$|sI - A| = 0$$

引入状态反馈后,系统的状态方程为:

$$\dot{x} = (A - BK)x + Bu$$

系统的输入矩阵和输出方程没变,而特征多项式变为:

$$|sI - (A - BK)| = 0$$

因此,状态反馈通过改变系统的系数矩阵,改变了特征多项式,继而改变了系统的极点。因此系统可通过线性状态反馈,实现闭环极点任意的任意配置。

(4) 使用状态反馈的前提条件是什么?

答：使用状态反馈进行零极点配置的前提条件是系统的状态是完全可控的。

2.已知系统

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 2 & 0 \end{bmatrix} x$$

设计全维状态观测器.使观测器的极点配置在-12 土 j。

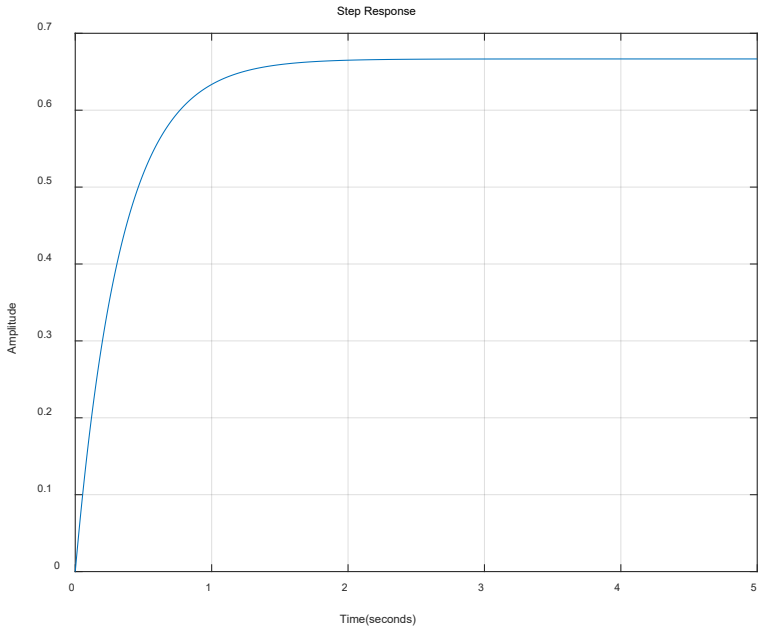
(1) 给出原系统的状态曲线。设：$x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, u(t) = 1(t)$

代码及结果：

代码：

```
clear;
A= [0 1;-3 -4]; B = [0;1];
C = [2 0]; D = [];
sys = ss(A,B,C,D);
t = 0:0.001:5;
x1 = step(sys,t);
x2 = initial(sys,[0;1],t);
plot(t,x1+x2);
title('Step Response');
xlabel('Time(seconds)');
ylabel('Amplitude');
grid on;
```

结果：



(2) 给出观测器的状态曲线并加以对比。（观测器的初始状态可以任意选取）

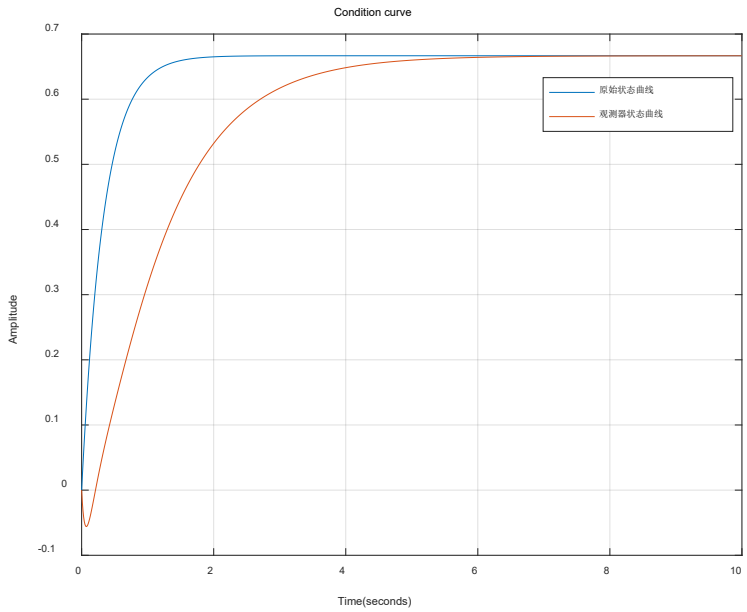
代码及结果：

代码：

```
A=[0 1;-3 -4];
B=[0;1];
C=[2 0];
D=[];
sys=ss(A,B,C,D);
t = 0:0.01:10;
x1 = step(sys,t);
x2 = initial(sys,[0;1],t);
plot(t,x1+x2);
title('Condition curve');
xlabel('Time(seconds)');
ylabel('Amplitude');
grid on;
hold on;
A1 =A';
B1 =C';
C1 =B';
P = [-12+j -12-j];
K = place(A1,B1,P);
G = K';
sys1=ss(A-G*C,[0;0],C,D);
x3=initial(sys1,[0;1],t);
x4=step(sys1,t);
```

```
plot(t,x1-x3-x4);
grid on;
legend('原始状态曲线','观测器状态曲线');
```

结果：



分析：

（见 A、B）

观察实验结果，思考以下问题：

A. 说明反馈控制闭环期望极点和观测器极点的选取原则。

答：由于线性定常系统的特征多项式为实系数多项式,因此考虑到问题的可解性,对期望的极点的选择应注意下列问题:(1)对于 n 阶系统,可以而且必须给出 n 个期望的极点;(2)期望的极点必须是实数或成对出现的共轭复数;(3)期望的极点必须体现对闭环系统的性能品质指标等的要求。

B. 说明观测器的引入对系统性能的影响。

答：由于状态变量是描述系统内部动态运动和特性的,因此对实际控制系统,它可能不能直接测量,更甚者是抽象的数学变量,实际中不存在物理量与之直接对应。若状态变量不能直接测量,则在状态反馈中需要引入所谓的状态观测器来估计系统的状态变量的值,再用此估计值来构成状态反馈律。

四．实验收获与心得

通过实验，加深了对“如何在系统不稳定时对其进行零极点配置”这个问题的理解以及其在 MATLAB 中如何具体实现。掌握了 acker 以及 place 的用法，判断状态反馈矩阵。

实验中，要灵活运用注释以及 clear 函数，防止上一个程序代码的参数残留在工作区影响正在调试的代码的运行。