基于MATLAB的线性控制系统的计算机辅助设计

燕山大学

电气工程学院

电工电能新技术研究室

2018/05

目 录

Contents Page

http://iee.ysu.edu.cn/~eepc/board.php?sub=3

目 录

Contents Page

目 录

Contents Page

一、常微分方程问题的MATLAB求解

线性常微分方程的解析解

- MALAB中提供了 dsolve()函数求解线性常系数微分方程的解析解，首先要用syms声明符号变量，以区别于MATLAB语言的常规数值变量，然后就可以使用dsolve(表达式)直接求解。

- 例1-1: $$\frac{d^4y(t)}{dt^4}+11\frac{d^3y(t)}{dt^3}+41\frac{d^2y(t)}{dt^2}+61\frac{dy(t)}{dt}+30y(t)=e^{-6t}\cos 5t$$

Command Window

```
>> syms t y; %声明符号变量
y=dsolve('D4y+11*D3y+41*D2y+61*Dy+30*y=exp(-6*t)*cos(5*t)');
>> pretty(simplify(y))  %以更好看的形式显示解析解
(exp(-6 t) (sin(5 t) 109 - cos(5 t) 79 + 181220 C2 exp(t) + C1 exp(3 t) 181220 + C3 exp(4 t) 181220

  + C4 exp(5 t) 181220))/181220
```

一、常微分方程问题的MATLAB求解

一阶常微分方程组的数值解法

- 对于形如：$x_i=f_i(t, x)$,   i=1,2,…,n
- MATLAB提供了一些不同方法的数值解函数：ode23()(二阶三极RK算法），ode45()（四阶五极RK算法），ode15s( )（变阶次刚性方程求解算法）调用格式都是一致的:

[t, *x*]=ode45(方程函数名，tspan,*x*0,选项，附加参数）

- 其中，t为仿真结果的自变量构成的向量（一般变步长），返回的x是一个矩阵，其列数为n，即微分方程的阶次，行数等于t的行数。“方程函数名”为用MATLAB编写的固定格式的M-函数，描述一阶微分方程组，tspan 为数值解时的初始和终止时间等信息，x0为初始状态变量，“选项”为求解微分方程的一些控制参数，还可以将一些“附加参数”在求解函数和方程描述函数之间传递。

一、常微分方程问题的MATLAB求解

一阶常微分方程组的数值解法

- 例1-2:

$$\begin{cases} \dot{x}(\mathrm{t}) = -y(\mathrm{t}) - z(\mathrm{t}) \\ \dot{y}(\mathrm{t}) = x(\mathrm{t}) + ay(\mathrm{t}) \\ \dot{z}(\mathrm{t}) = \mathrm{b} + [x(\mathrm{t}) - c]z(\mathrm{t}) \end{cases}$$

且有

$$a = b = 0.2$$
$$c = 5.7$$
$$x(0) = y(0) = z(0) = 0$$

- 选状态变量x1=x，x2=y，x3=z。

$$\begin{cases} x1'(\mathrm{t}) = -x2(\mathrm{t}) - x3(\mathrm{t}) \\ x2'(\mathrm{t}) = x1(\mathrm{t}) + ax2(\mathrm{t}) \\ x3'(\mathrm{t}) = \mathrm{b} + [x1(\mathrm{t}) - c]x3(\mathrm{t}) \end{cases}$$

若要求该方程，需先编写一个M函数描述方程组:

一、常微分方程问题的MATLAB求解

一阶常微分方程组的数值解法

■ 编写M函数，保存为rossler.m

rossler.m

```
function dx=rossler(t,x)  %虽然不显含时间，还应该写出占位符。
dx=[-x(2)-x(3);  % 对应方程第一行，直接将参数代入。
x(1)+0.2*x(2);  % 对应方程第二行，直接将参数代入。
0.2+(x(1)-5.7)*x(3)];% 对应方程第三行，直接将参数代入。
```

$$\begin{cases} x1'(t) = -x2(t) - x3(t) \\ x2'(t) = x1(t) + ax2(t) \\ x3'(t) = \mathrm{b} + [x1(t) - c]x3(t) \end{cases}$$

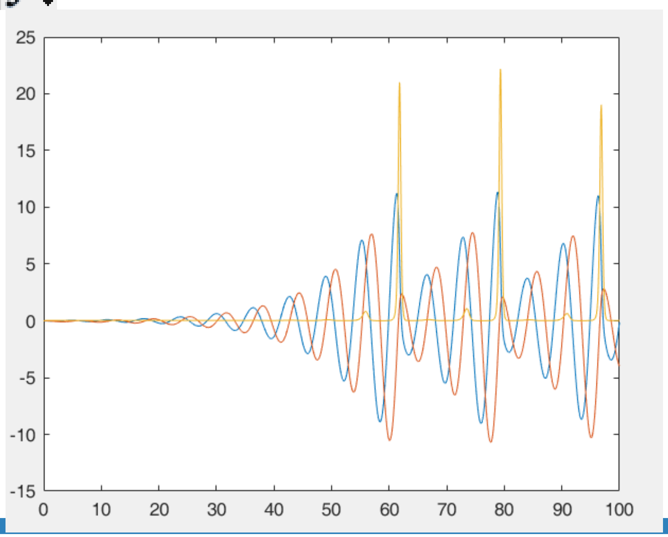
一、常微分方程问题的MATLAB求解

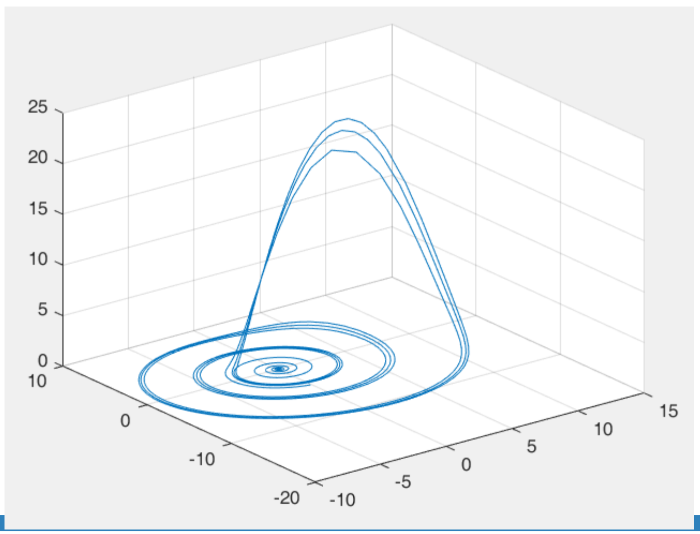
一阶常微分方程组的数值解法

■ 在命令窗口输入：

Command Window

```
>> x0=[0;0;0];%微分方程的初值
[t,y]=ode45('rossler',[0,100],x0);%求解微分方程
plot(t,y)%绘制各个状态变量的时间响应
figure;plot3(y(:,1),y(:,2),y(:,3)),grid,%绘制相空间图形
fx >>
```





燕山大学 YANSHAN UNIVERSITY

一、常微分方程问题的MATLAB求解

一阶常微分方程组的数值解法

■ 如果a,b,c三个参数需要用外部命令给出，则可以按右面的格式写出一个新的M函数来描述微分方程组，其中用变量flag占位。

Editor - G:\电力电子电路仿真\修改后课件180514\Example\chapter2\rossler1.m

rossler1.m | rossler.m

```matlab
function dx=rosslerabc(t,x,flag,a,b,c)  %加入附加参数。
dx=[-x(2)-x(3); % 对应方程第一行，直接将参数代入。
x(1)+a*x(2);  % 对应方程第二行，直接将参数代入。
b+(x(1)-c)*x(3)];% 对应方程第三行，直接将参数代入。
```

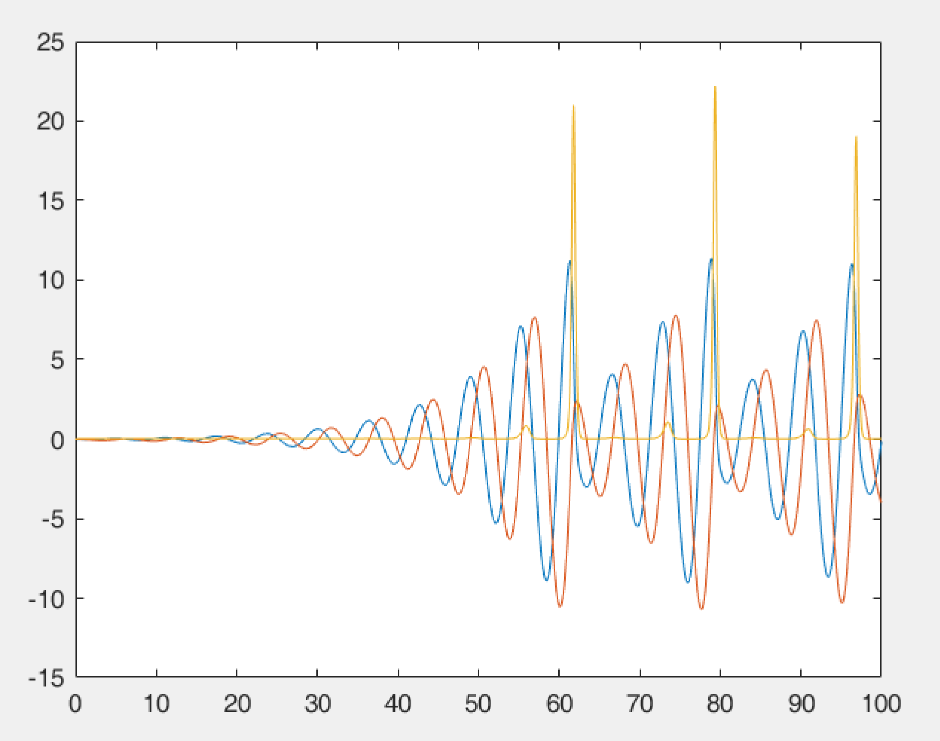
Command Window

```matlab
>> x0=[0;0;0];%微分方程的初值
>> a=0.2;b=0.2;c=5.7;%从函数外部定义三个变量，无需修改函数本身
>> [t,y]=ode45('rossler1',[0,100],x0,[],a,b,c);%用附加参数求解微分方程
>> plot(t,y)%绘制各个状态变量的时间响应
```

一、常微分方程问题的MATLAB求解

一阶常微分方程组的数值解法

- 如果a,b,c三个参数需要用外部命令给出，则可以按右面的格式写出一个新的M函数来描述微分方程组，其中用变量flag占位。



- 这样编写M函数，可以方便的修改a,b,c等参数。在修改参数时，无需修改M函数文件本身，只需在求解该方程时将参数代入即可。

目 录

Contents Page

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 线性系统的传递函数模型

$$G(s)=\frac{b_1s^m+b_2s^{m-1}+\ldots+b_ms+b_{m+1}}{a_1s^n+a_2s^{n-1}+\ldots+a_ns+a_{n+1}}$$

- 可用两种方法进行表示
  - 第一种：可利用tf()函数表示，调用格式为：

```
>> num=[b1, b2,… bm, bm+1]; den =[a1, a2,… an, an+1];
>>G=tf(num,den);   %分子: numerator ,分母: denominator
```

  - 第二种：也可用s=tf('s')先定义传递函数的算子，然后用类似数学表达式的形式直接输入系统的传递函数模型。

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

■ 例2-1:

第一种方法

```
>> num=[12 24 12 20];den=[2 4 6 2 2];
>> G=tf(num,den)

Transfer function:
  12 s^3 + 24 s^2 + 12 s + 20
--------------------------------
2 s^4 + 4 s^3 + 6 s^2 + 2 s + 2
```

```
>> a=[2 2 2];b=[3 3 3 3];
>> f=tf(a,b)

Transfer function:
    2 s^2 + 2 s + 2
-----------------------
3 s^3 + 3 s^2 + 3 s + 3
```

函数调用

第二种方法

```
>> s=tf('s');
>> l=(s+1)/(s^2-s+5)

Transfer function:
   s + 1
-----------
s^2 - s + 5
```

S不能改成其它字母

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 对已建立的传递函数，可以有两种方法提取系统的分子和分母多项式。
- 第一种方法使用tfdata()函数，以上例传递函数G为例用法如下:

Command Window

```
>> num=[12 24 12 20];
>> den=[2 4 6 2 2];
>> G=tf(num,den)

G =

   12 s^3 + 24 s^2 + 12 s + 20
  -----------------------------
  2 s^4 + 4 s^3 + 6 s^2 + 2 s + 2
```

```
>> [num,den]=tfdata(G,'v') %"  "v"表示想获得的数值

num =

    0    12    24    12    20


den =

    2     4     6     2     2
```

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

■ 例2-3：以上例传递函数l为例用法如下：

Command Window

```
>> s=tf('s');
>> l=(s+1)/(s^2-s+5)

l =

    s + 1
  -----------
  s^2 - s + 5
```

```
>> [num,den]=tfdata(l)

num =

    [1x3 double]


den =

    [1x3 double]
```

```
>> [num,den]=tfdata(l,'v')

num =

     0     1     1


den =

     1    -1     5
```

```
>> [c,d]=tfdata(l,'v')

c =

     0     1     1


d =

     1    -1     5
```

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 第二种方法：通过下列语句直接提取传递函数分子和分母多项式
  - >> num=G.num{1}; den=G.den{1};%可以直接提取分子和分母多项式。

```
Command Window
>> num=[12 24 12 20];
>> den=[2 4 6 2 2];
>> G=tf(num,den)

G =

  12 s^3 + 24 s^2 + 12 s + 20
  -------------------------------
  2 s^4 + 4 s^3 + 6 s^2 + 2 s + 2
```

```
>> num=G.num{1}%直接提取分子多项式。

num =

     0    12    24    12    20

>> den=G.den{1}%直接提取分母多项式。

den =

     2     4     6     2     2
```

- 其中{1}实际上为{1,1}，表示第1输入和第1输出之间的传递函数，该方法直接适用于多变量系统的描述。

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 线性系统的状态方程模型

$$\begin{cases} x(\mathrm{t}) = \mathrm{A}\,x(\mathrm{t}) + \mathrm{B}u(\mathrm{t}) \\ y(\mathrm{t}) = \mathrm{C}\,x(\mathrm{t}) + \mathrm{D}u(\mathrm{t}) \end{cases}$$

- MATLAB使用G=ss(A,B,C,D)来建立状态方程，如下例2-5:

```
>> A=[-1,-7,-6,-9;6,8,4,6;6,7,8,6;-5,-8,-8,-6];B=[1,2;1,3;2,1;0,5];
>> C=[2,5,0,8;3,3,2,1];D=zeros(2,2);G=ss(A,B,C,D)
a =
          x1   x2   x3   x4
     x1   -1   -7   -6   -9
     x2    6    8    4    6
     x3    6    7    8    6
     x4   -5   -8   -8   -6
```

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 续上页程序

```
b =
          u1   u2
     x1    1    2
     x2    1    3
     x3    2    1
     x4    0    5
c =
          x1   x2   x3   x4
     y1    2    5    0    8
     y2    3    3    2    1
d =
          u1   u2
     y1    0    0
     y2    0    0

Continuous-time model.
```

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 读取已建立的一个状态方程的参数可使用ssdata()或者G.a来实现，调用方法如下例2-6:

```
>> [A,B,C,D]=ssdata(G)

A =

   -1   -7   -6   -9
    6    8    4    6
    6    7    8    6
   -5   -8   -8   -6

B =

    1    2
    1    3
    2    1
    0    5

C =

    2    5    0    8
    3    3    2    1

D =

    0    0
    0    0
```

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 采用G.a实现，调用方法如下例2-7:

```
>> G.a

ans =

    -1    -7    -6    -9
     6     8     4     6
     6     7     8     6
    -5    -8    -8    -6
```

```
>> G.B

ans =

     1     2
     1     3
     2     1
     0     5
```

```
>> G.c

ans =

     2     5     0     8
     3     3     2     1
```

```
>> G.d

ans =

     0     0
     0     0
```

- A、B、C、D不区分大小写。

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 线性系统的零极点模型

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

- 可用两种方法进行表示
  - 第一种：可利用zpk()函数表示，调用格式为:

    \>\> z=[z1； z2； …； zm];p=[p1； p2； …； pn];

    \>\> G=zpk(z,p,k);

  - 第二种：也可用s=zpk(‘s’)先定义传递函数的算子，然后用类似数学表达式的形式直接输入系统的传递函数模型。

    \>\> s=zpk(‘s’)　%定义零极点形式的Laplace算子

    \>\> G=…　　%输入G零极点表达式

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 例2-8:

```
>> p=[-1;-2;-3;-4];z=[-5;-2+2i;-2-2i];
G=zpk(z,p,6)

G =

  6 (s+5) (s^2 + 4s + 8)
  -----------------------
  (s+1) (s+2) (s+3) (s+4)

Continuous-time zero/pole/gain model.
```

```
>> s=zpk('s')
G=6*(s-2)*(s-7)/(s-3)*(s-5)*(s+5)

s =

  s

Continuous-time zero/pole/gain model.


G =

  6 (s-2) (s-5) (s-7) (s+5)
  -------------------------
           (s-3)

Continuous-time zero/pole/gain model.
```

二、线性控制系统的数学模型及MATLAB表示

线性连续系统模型及MATLAB表示

- 例2-8:

```
>> s=zpk('s')
>> G=6*(s-2)(s+7)/(s-3)(s-5)(s+5)
??? G=6*(s-2)(s+7)/(s-3)(s-5)(s+5)
                |
Error: Unbalanced or unexpected parenthesis or bracket.
>> G=6*(s-2)*(s+7)/(s-3)*(s-5)*(s+5)

Zero/pole/gain:
6 (s-2) (s-5) (s+5) (s+7)
-------------------------
          (s-3)
>> G=6*(s-2)*(s+7)/((s-3)*(s-5)*(s+5))

Zero/pole/gain:
  6 (s-2) (s+7)
-----------------
(s-3) (s-5) (s+5)
>>
```

缺少乘号

括号的作用

二、线性控制系统的数学模型及MATLAB表示

线性离散系统模型及MATLAB表示

- 离散系统的传递函数模型,可用两种方法进行表示

$$H(z)=\frac{b_0z^n+b_1z^{n-1}+\ldots+b_{n-1}z+b_n}{a_1z^n+a_2z^{n-1}+\ldots+a_nz+a_{n+1}}$$

- 第一种：可利用tf()函数表示，调用格式为：

  >> num=[b_0,b_1,…b_{n-1},b_n]; den=[a_1,a_2,…a_n,a_{n+1}];

  >> H=tf(num,den,’Ts’,T);　% T为实际的采样周期数值

- 第二种：也可用z=tf(‘z’,T)先定义传递函数的算子，然后用类似数学表达式的形式直接输入系统的传递函数模型。

  >> z=tf(‘z’,T);　%定义离散传函算子

  >> H=…　　　　%输入离散传函表达式

二、线性控制系统的数学模型及MATLAB表示

线性离散系统模型及MATLAB表示

■ 例2-9

```
>> num=[6 -0.6 -0.12];den=[1 -1 0.25 0.25 -0.125];
H=tf(num,den,'Ts',0.1) %输入并显示系统的传递函数模型

H =

        6 z^2 - 0.6 z - 0.12
  --------------------------------------
  z^4 - z^3 + 0.25 z^2 + 0.25 z - 0.125

Sample time: 0.1 seconds
Discrete-time transfer function.
```

```
>> z=tf('z',0.1);
H=(6*z^2-0.6*z-0.12)/(z^4-z^3+0.25*z^2+0.25*z-0.125)

H =

        6 z^2 - 0.6 z - 0.12
  -------------------------------------
  z^4 - z^3 + 0.25 z^2 + 0.25 z - 0.125

Sample time: 0.1 seconds
Discrete-time transfer function.
```

二、线性控制系统的数学模型及MATLAB表示

线性离散系统模型及MATLAB表示

- 如果以z^{-1}为算子（即以z的负幂表示）时，即

$$\hat{H}(z^{-1})=\frac{b_0+b_1z^{-1}+\ldots+b_{n-1}z^{-n+1}+b_nz^{-n}}{a_1+a_2z^{-1}+\ldots+a_nz^{-n+1}+a_{n+1}z^{-n}}$$

  - 这时可以按照z的负幂的表示形式定义该传函:

    >>num=[b_0 ,b_1,…b_{n-1},b_n]; den=[a_1, a_2,… a_n , a_{n+1}];

    >>H=tf(num,den,'Ts',T,'variable','z^-1');

  - 也可以先用q取代z^{-1}，$\hat{H}(q)=\dfrac{b_nq^n+b_{n-1}q^{n-1}+\ldots+b_1q+b_0}{a_{n+1}q^n+a_nq^{n-1}+\ldots+a_2q+a_1}$
  - 然后按照q的正幂表示形式定义该传函，方法是:

    >>num=[b_n,b_{n-1},…b_1,b_0]; den=[a_{n+1},a_n, a_{n-1},…,a_1];

    >>H=tf(num,den,'Ts',T,'variable','q');

二、线性控制系统的数学模型及MATLAB表示

线性离散系统模型及MATLAB表示

- 例2-10

```
>> num=[6 -0.6 -0.12];den=[1 -1 0.25 0.25 -0.125];
>> H=tf(num,den,'Ts',0.1,'variable','z^-1')

H =

          6 - 0.6 z^-1 - 0.12 z^-2
  -----------------------------------------
  1 - z^-1 + 0.25 z^-2 + 0.25 z^-3 - 0.125 z^-4

Sample time: 0.1 seconds
Discrete-time transfer function.
```

```
>> num=[6 -0.6 -0.12];
>> den=[1 -1 0.25 0.25 -0.125];
>> H=tf(num,den,'Ts',0.1,'variable','q')

H =

        6 q^2 - 0.6 q - 0.12
  ----------------------------------
  q^4 - q^3 + 0.25 q^2 + 0.25 q - 0.125

Sample time: 0.1 seconds
Discrete-time transfer function.
```

二、线性控制系统的数学模型及MATLAB表示

线性离散系统模型及MATLAB表示

- 离散系统的状态方程模型

$$\begin{cases} x[(k+1)T] = Fx(kT) + Gu(kT) \\ y(kT) = Cx(kT) + Du(kT) \end{cases}$$

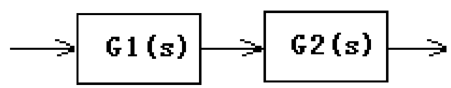
- 可使用以下函数调用方式进行定义:

>> H=ss(F, G, C, D, ’Ts’, T);

二、线性控制系统的数学模型及MATLAB表示

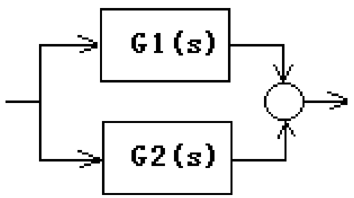
方框图描述系统的化简

- 方框图描述系统（适用于连续和离散系统）
- 控制系统的典型连接结构



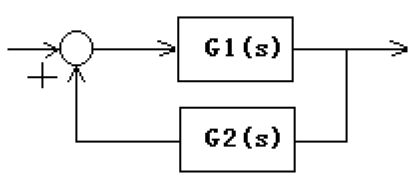
串联结构

G=G2*G1
（对于多变量系统注意顺序）



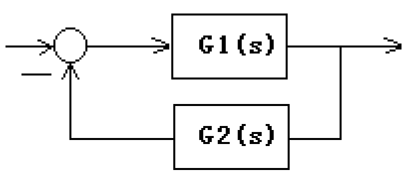
并联结构

G=G1+G2



正反馈系统

G=feedback(G1,G2,1)



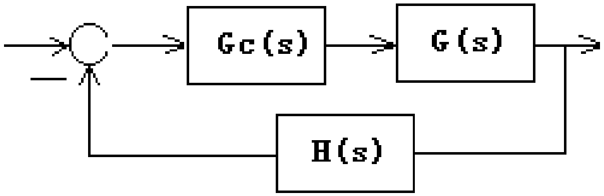
负反馈系统

G=feedback(G1,G2)

二、线性控制系统的数学模型及MATLAB表示

方框图描述系统的化简

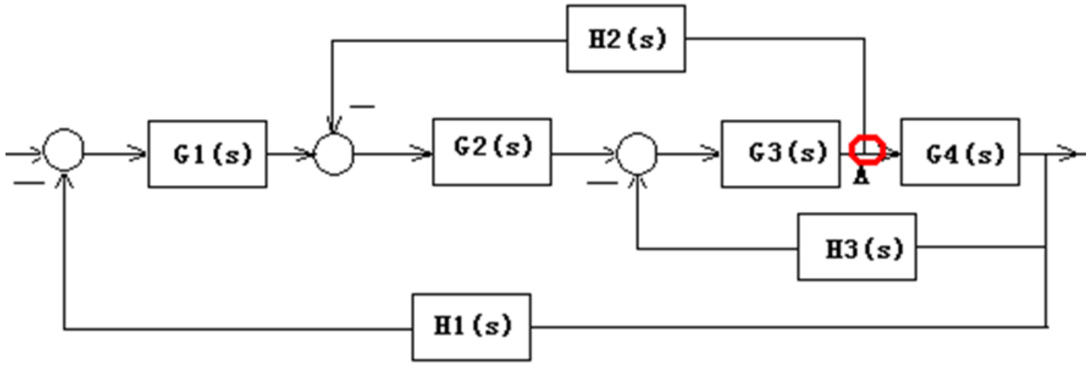
- 节点移动时的等效变换
- 例2-11:



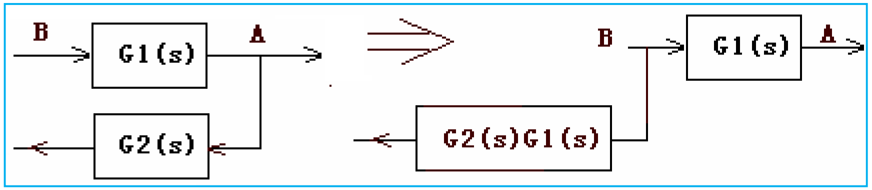

```
>> s=tf('s');G=(12*s^3+24*s^2+12*s+20)/(2*s^4+4*s^3+6*s^2+2*s+2);
>> Gc=(5*s+3)/s;H=1000/(s+1000);
>> GG=feedback(G*Gc,H)

Transfer function:
     60 s^5 + 60156 s^4 + 156132 s^3 + 132136 s^2 + 136060 s + 60000
------------------------------------------------------------------------
2 s^6 + 2004 s^5 + 64006 s^4 + 162002 s^3 + 134002 s^2 + 138000 s + 60000
```

二、线性控制系统的数学模型及MATLAB表示

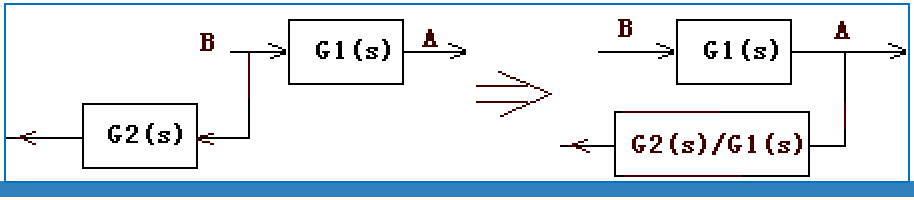
方框图描述系统的化简



- 对于如上图，比较复杂的结构图，在简化的时候，需要将某支路从一个节点移动到另一个节点上（如从A点移动到输出位置）



前向移动节点
G2*G1



后向移动节点
G2/G1或G2*inv(G1)

目 录

Contents Page

三、线性系统稳定性分析及稳定性直接判定方法

线性系统稳定性分析

- 控制理论发展初期：受传统数学理论的影响，认为高阶系统不能求出所有特征根，而出现了各种各样的间接判定方法：Routh表、Hurwitz矩阵法、离散系统Jury判据等。
- 线性系统稳定的充要条件：系统状态方程中A矩阵的特征根均具有负实部。由控制理论知，系统A的特征根和系统的极点是一致的，所以若能获得系统的极点，则可以直接判定稳定性。
- Z域中，线性定常离散系统稳定的充要条件：离散特征方程D(z)=1+GH(z)=0的全部特征根分布在Z平面上的单位圆内。
- 目前通过MALAB求解特征根轻而易举：基于直接求解方法的控制系统稳定性判定方法——**eig()函数求系统特征根。**

三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

■ 假设系统模型已知为G，不管模型G是传递函数，状态方程还是零极点模型，都可以用控制系统工具箱函数：eig(G)来求取连续线性定常系统的特征根。

E = eig(X) is a vector containing the eigenvalues of a square matrix X.

■ pzmap(G)函数能用图形方式直观的绘出系统所有特征根在s-复平面上的位置，是否稳定只要看一下系统所有极点是不是在s-复平面上均位于虚轴左侧即可。

■ 还可采用pole(G)和zero(G)分别求出系统的极点和零点。

三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

■例3-1：用图形显示一个单位负反馈系统的全部零极点位置，系统开环传递函数为

$$G=(10s^4+50s^3+100s^2+100s+40)/(s^7+21s^6+184s^5+870s^4+2384s^3+3664s^2+2496s)$$

Command Window

```
>> num=[10,50,100,100,40];
>> den=[1,21,184,870,2384,3664,2496,0];
>> G=tf(num,den);
>> GG=feedback(G,1);
>> eig(GG)
```

```
ans =

  -6.9223 + 0.0000i
  -3.6502 + 2.3020i
  -3.6502 - 2.3020i
  -2.0633 + 1.7923i
  -2.0633 - 1.7923i
  -2.6349 + 0.0000i
  -0.0158 + 0.0000i
```

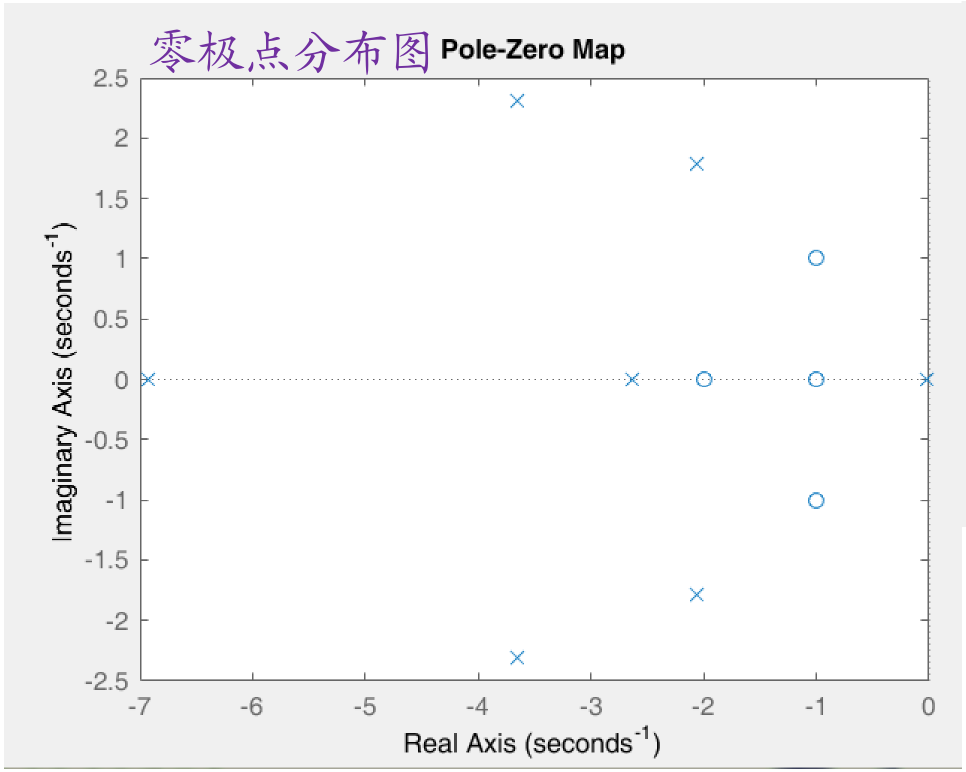
系统特征根

由系统特征根可以看出：

闭环系统所有极点都在S-左半平面，系统是稳定的。

三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根



系统极点

```
>> pole(GG)

ans =

  -6.9223 + 0.0000i
  -3.6502 + 2.3020i
  -3.6502 - 2.3020i
  -2.0633 + 1.7923i
  -2.0633 - 1.7923i
  -2.6349 + 0.0000i
  -0.0158 + 0.0000i
```

系统零点

```
>> zero(GG)

ans =

  -2.0000 + 0.0000i
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i
  -1.0000 + 0.0000i
```

由零极点分布图可以看出：

闭环系统所有极点都在S-左半平面，系统是稳定的。

三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

■ 使用转换语句

```
Command Window
>> num=[10,50,100,100,40];
>> den=[1,21,184,870,2384,3664,2496,0];
>> G=tf(num,den);
>> GG=feedback(G,1);
>> zpk(GG)

ans =

              10 (s+2) (s+1) (s^2 + 2s + 2)
  --------------------------------------------------------------------------
  (s+6.922) (s+2.635) (s+0.01577) (s^2 + 4.127s + 7.47) (s^2 + 7.3s + 18.62)

Continuous-time zero/pole/gain model.
```

三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

■ 例3-2：离散系统稳定性分析

```
Command Window
>> num=[6,-0.6,-0.12];
>> den=[1 -1 0.25 0.25 -0.125];
>> H=tf(num,den,'Ts',0.1);
>> z=tf('z','Ts',0.1);
>> Gc=0.3*(z-0.6)/(z+0.8);
>> GG=feedback(H*Gc,1);
>> abs(eig(GG)')

ans =

   1.1644    1.1644    0.5536    0.3232    0.3232
```

求取闭环系统的特征根的模，并将结果转置显示

可以看出，前两个特征根的模均大于1，所以可判定该闭环系统是不稳定的。

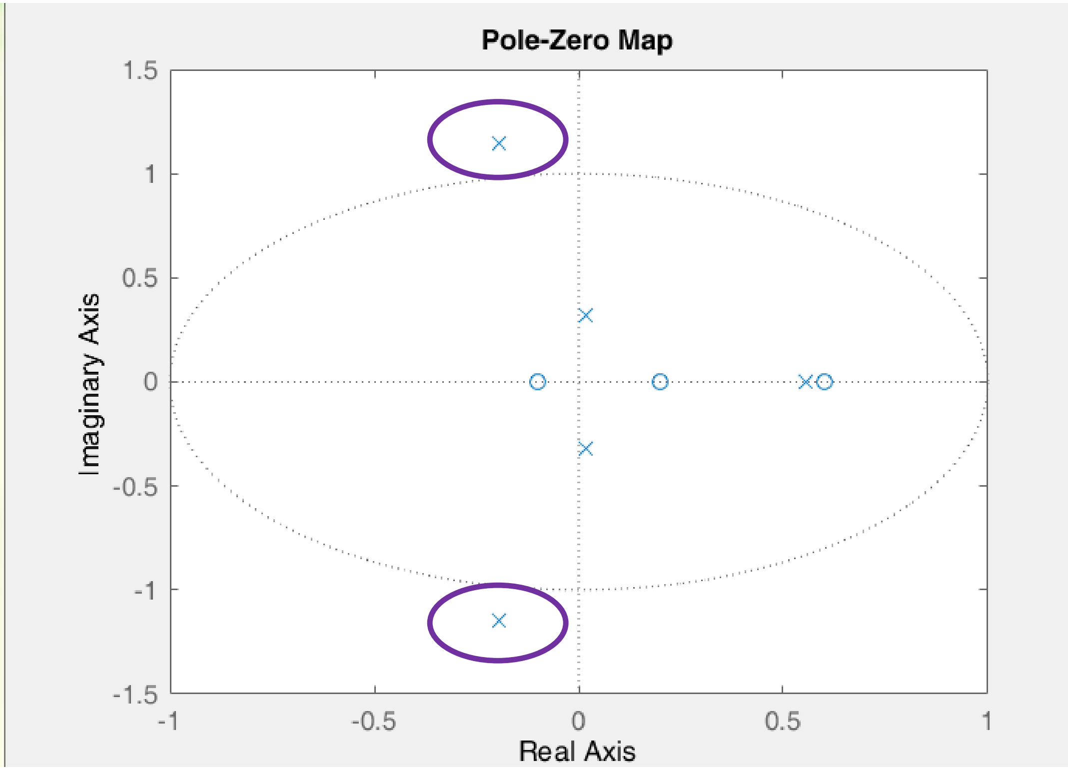
三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

■ 零极点分布图

Command Window

```
>> num=[6,-0.6,-0.12];
den=[1 -1 0.25 0.25 -0.125];
H=tf(num,den,'Ts',0.1);
z=tf('z','Ts',0.1)
Gc=0.3*(z-0.6)/(z+0.8);
GG=feedback(H*Gc,1);
pzmap(GG)

z =


  z

Sample time: 0.1 seconds
Discrete-time transfer function.

>>
>>
```



三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

- 使用转换语句

```
>> num=[6,-0.6,-0.12];
>> den=[1 -1 0.25 0.25 -0.125];
>> H=tf(num,den,'Ts',0.1);
>> z=tf('z','Ts',0.1);
>> Gc=0.3*(z-0.6)/(z+0.8);
>> GG=feedback(H*Gc,1);
>> zpk(GG)

ans =

                1.8 (z-0.6) (z-0.2) (z+0.1)
  -----------------------------------------------------------
  (z-0.5536) (z^2 - 0.03727z + 0.1045) (z^2 + 0.3908z + 1.356)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

三、线性系统稳定性分析及稳定性直接判定方法

稳定性直接判定方法——求系统特征根

- 采用直接判定的方法，除了能获得稳定性信息外，还可以直接看出零极点分布，从而对系统有个更好点的了解。比如，
  - 对于连续系统来说，如果存在距离虚轴特别近的复极点，则可能会使系统有很强的振荡；
  - 对于离散系统来说，如果复极点距离单位圆较近，也有可能得出较强的振荡，这样用间接判据（例如Routh判据）是不可能得到的。

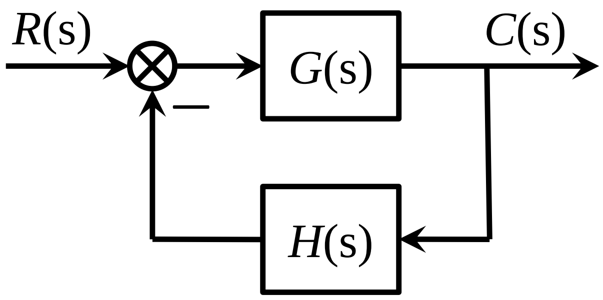
目 录

Contents Page

四、线性系统的根轨迹分析方法

根轨迹分析

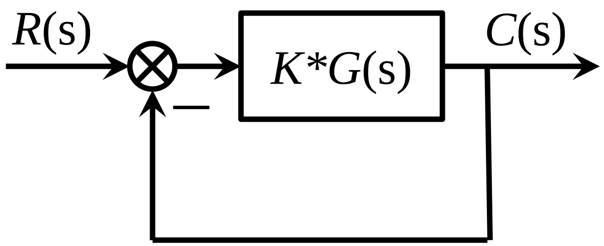
- 根轨迹：当系统开环传递函数中某一参数从零变化到无穷时，闭环特征方程式的根在s平面上运动的轨迹。
  - 单变量开环传递函数$G(s)$。
  - 控制器增益为K。
  - 单位负反馈系统。
  - 闭环系统特征方程。
  - 对 K 的不同取值，则可能绘制出每个特征根变化的曲线，这样的曲线称为系统的根轨迹。
  - 根轨迹用开环信息研究闭环特性。



四、线性系统的根轨迹分析方法

根轨迹分析

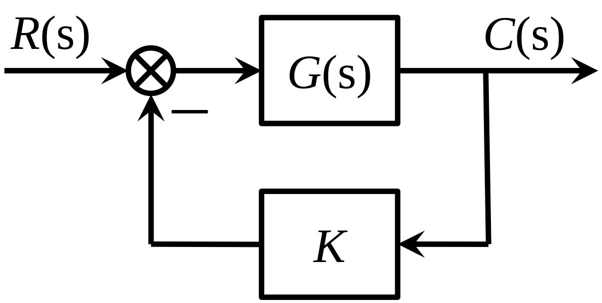
- 根轨迹：当系统开环传递函数中某一参数从零变化到无穷时，闭环特征方程式的根在s平面上运动的轨迹。
  - 单变量开环传递函数$G(s)$。
  - 控制器增益为K。
  - 单位负反馈系统。
  - 闭环系统特征方程。
  - 对 K 的不同取值，则可能绘制出每个特征根变化的曲线，这样的曲线称为系统的根轨迹。
  - 根轨迹用开环信息研究闭环特性。



四、线性系统的根轨迹分析方法

根轨迹分析

- 根轨迹：当系统开环传递函数中某一参数从零变化到无穷时，闭环特征方程式根在s平面上运动的轨迹。**MATLAB中的描述。**
  - 单变量开环传递函数$G(s)$。
  - 反馈回路增益为K。
  - 负反馈系统。
  - 闭环系统特征方程。
  - 对 K 的不同取值，则可能绘制出每个特征根变化的曲线，这样的曲线称为系统的根轨迹。
  - 根轨迹用开环信息研究闭环特性。



四、线性系统的根轨迹分析方法

线性系统的计算机辅助设计——根轨迹分析

- MATLAB 求解

  >> rlocus(G);　　　　%不返回变量将自动绘制根轨迹曲线

  >> rlocus(G,k);　　　%给定增益向量，绘制根轨迹曲线

  >> [R, k]=rlocus(G) ;　%R为闭环特征根构成的复数矩阵

  >> rlocus(G1, ’-’, G2, ’-.b’, G3, ’:r’);　%同时绘制若干系统的根轨迹

- 该函数可以用于单变量不含有时间延迟的连续、离散系统的根轨迹绘制，也可以用于带有时间延迟的单变量离散系统的根轨迹绘制。
- 在绘制的根轨迹上，用鼠标点击某点，将显示出关于该点的有关信息，包括该点的增益值，对应的系统特征根的值和可能的闭环系统阻尼比和超调量等。
- 绘制根轨迹后，若给出**grid**命令，则将在根轨迹曲线上叠印出等阻尼线和等自然频率线，根据等阻尼线可以进行基于根轨迹的系统设计。

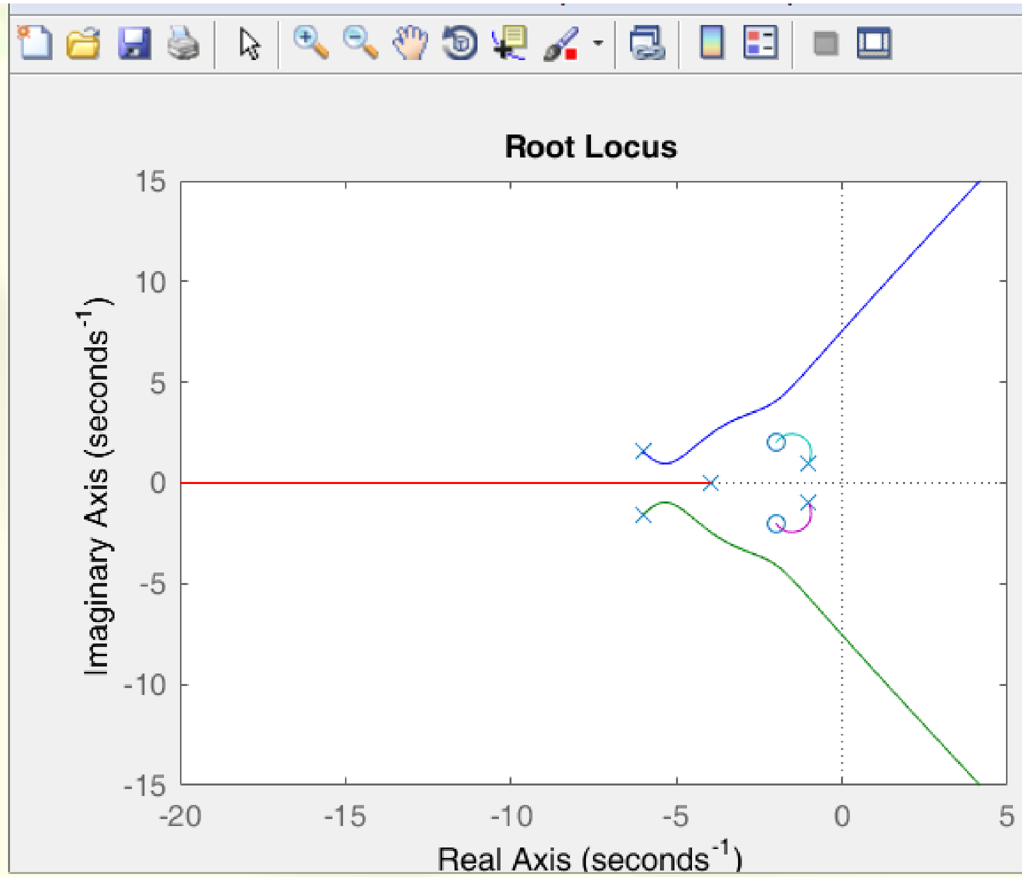
四、线性系统的根轨迹分析方法

线性系统的计算机辅助设计——根轨迹分析

■ 例4-1：已知开环传函G(s)

Command Window

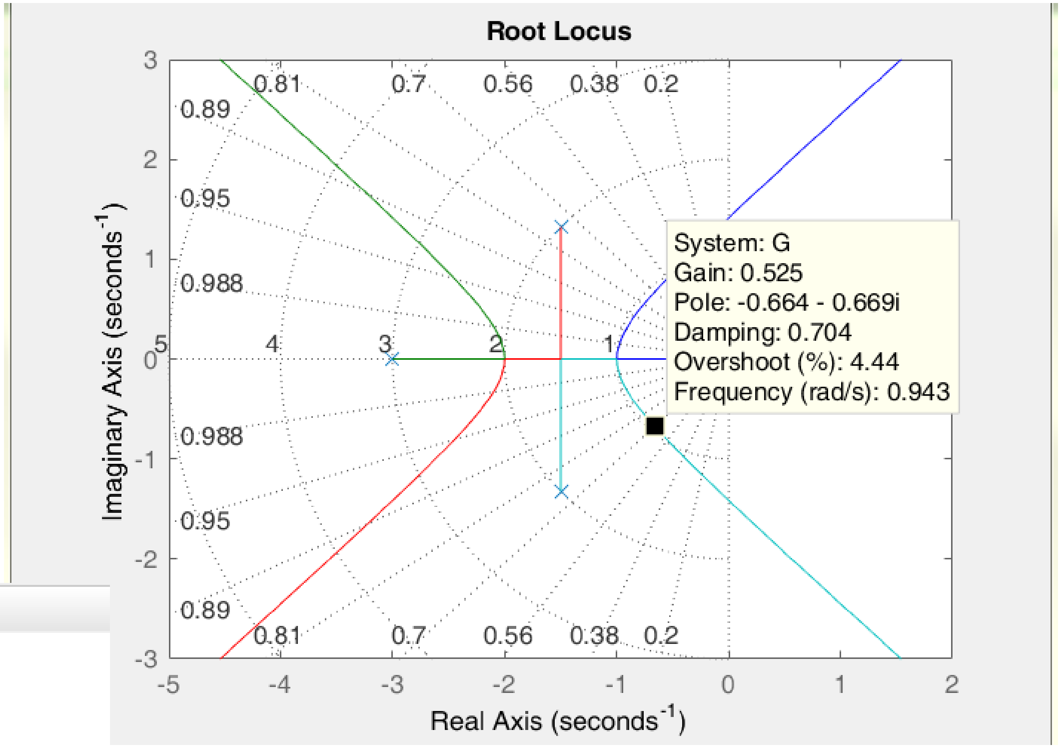
```
>> clear
>> num=[1 4 8];
>> den=[1 18 120.3 357.5 478.5 306];
>> G=tf(num,den);
>> rlocus(G)
fx >>
```



四、线性系统的根轨迹分析方法

线性系统的计算机辅助设计——根轨迹分析

■ 例4-2：已知开环传函G(s)，根据等阻尼线，单击阻尼比为0.707附近的点，看出增益约为0.525，这样可绘制系统的阶跃响应曲线。



Command Window

```
>> clear
>> s=tf('s');
>> G=10/(s*(s+3)*(s^2+3*s+4));
>> rlocus(G),grid
```

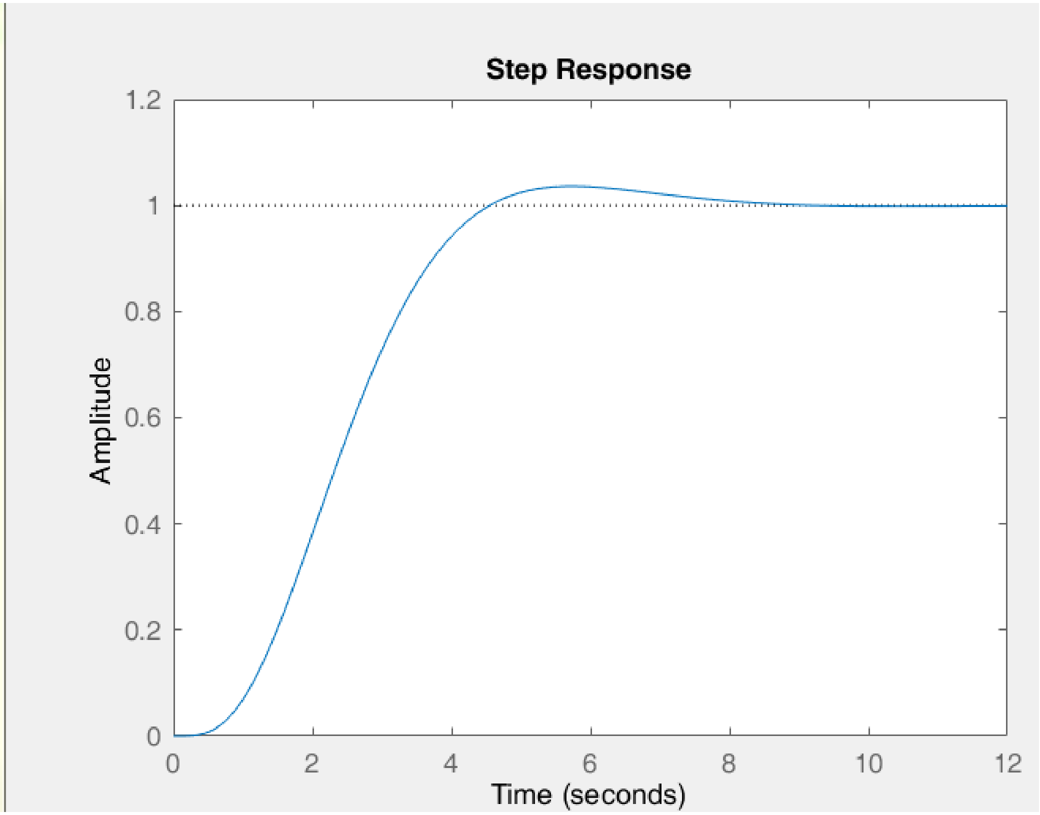
四、线性系统的根轨迹分析方法

线性系统的计算机辅助设计——根轨迹分析

■ 可以看出这样设计的动态性能比较令人满意

Command Window

```
>> clear
>> s=tf('s');
>> G=10/(s*(s+3)*(s^2+3*s+4));
>> rlocus(G),grid
>> k=0.523;
>> step(feedback(G*k,1))
>>
```



目 录

Contents Page

五、线性系统的频域分析方法

线性系统的频域分析方法

- 频域分析法主要适用于线性定常系统，可以根据系统的开环频率特性去判断闭环系统的稳定性。
- Nyquist提出用于系统稳定性分析的**Nyquist定理**，Bode提出了另一种频率响应的分析方法，同时可以分析系统的幅值相位与频率之间的关系（**Bode图**），Nichols在Bode图的基础上又进行了重新定义，构成了**Nichols图**。这些方法是单变量系统频域分析最重要的方法。而对于多变量系统，由于信号之间存在相互耦合，如希望对某对输入输出单独设计控制器，则需要引入解耦。

五、线性系统的频域分析方法

单变量系统的频域分析方法

■ 对于系统的传函模型$G(s)$来说若用jω取代s，则描述这个复数变量有不同方法，根据表示方法的不同，就可以构造出不同的频率响应曲线:

- $G(j\omega)=P(\omega)+jQ(\omega)$若横轴表示实数，纵轴表示虚数，则可以将$G(j\omega)$在复数平面上表示出来，即Nyquist图，是分析系统稳定性和一些性能的有效工具。<u>但传统的Nyquist图不能提供频率信息。</u>
- $G(j\omega)=A(\omega)e^{-j\Phi(\omega)}$ 以频率ω为横轴，幅值$A(\omega)$的对数为纵轴，或以频率ω为横轴，相位 $\Phi(\omega)$为纵轴，即Bode图。
- $G(j\omega)=A(\omega)e^{-j\Phi(\omega)}$ 以相位$\Phi(\omega)$为横轴，以单位为dB的幅值为纵轴，即Nichols图。

五、线性系统的频域分析方法

Nyquist图的绘制方法

- 在MATLAB下提供了一个nyquist()函数，可以绘制系统的Nyqusit图，该函数常用的调用格式如下:

  >>nyquist(G)　　%不返回变量将自动绘制Nyquist图

  >>nyquist(G,{ω_{min}, ω_{max}})　　%给定频率范围绘制Nyquist图

  >>nyquist(G, ω)　　%给定频率向量ω绘制Nyquist图

  >>[R, I, ω]=nyquist(G)　　%计算Nyquist响应数值

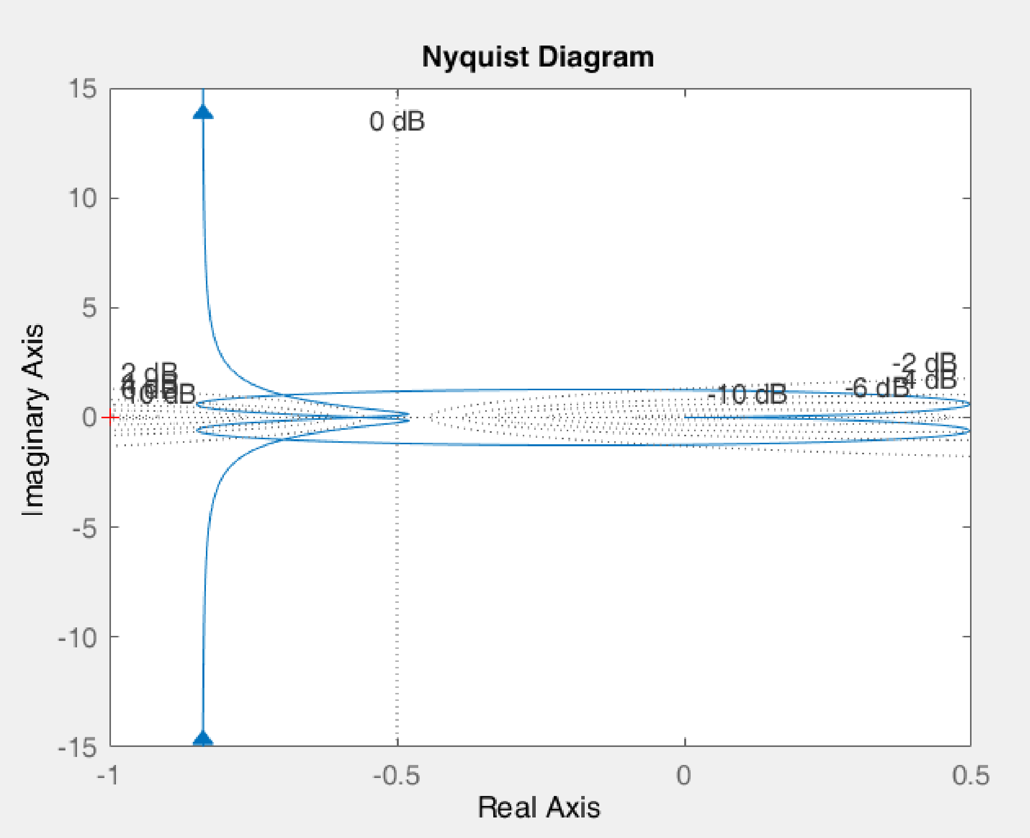
  >> nyquist(G1,'-',G2,'-.b',G3,':r')　%绘制几个系统的 Nyquist图

- 单击Nyqusit图上的点，可显示该点处增益与频率之间的关系。
- 改写的grid命令可以在Nyquist图上叠印出等M圆。

五、线性系统的频域分析方法

Nyquist图的绘制方法

- 例5-1：绘制Nyquist图



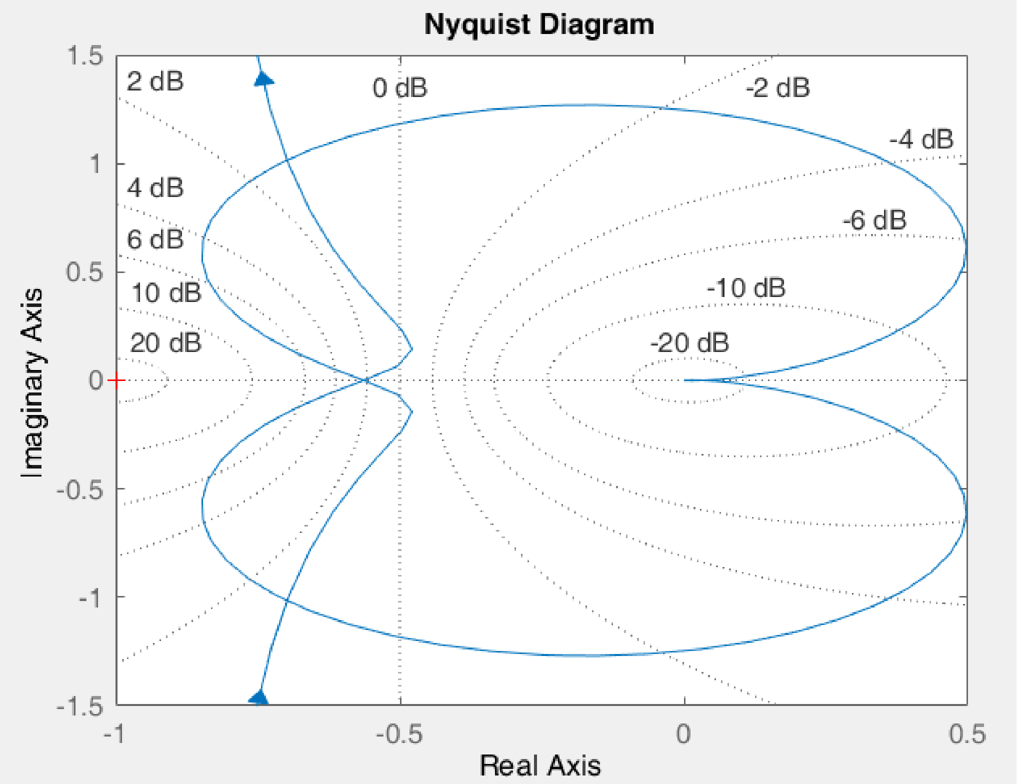
Command Window

```
>> clear
>> s=tf('s');
>> G=(s+8)/(s*(s^2+0.2*s+4)*(s+1)*(s+3));
>> nyquist(G),grid  %绘制Nyquist图并叠印等幅值圆
```

五、线性系统的频域分析方法

Nyquist图的绘制方法

- 例5-1：绘制Nyquist图，调整图形显示范围。



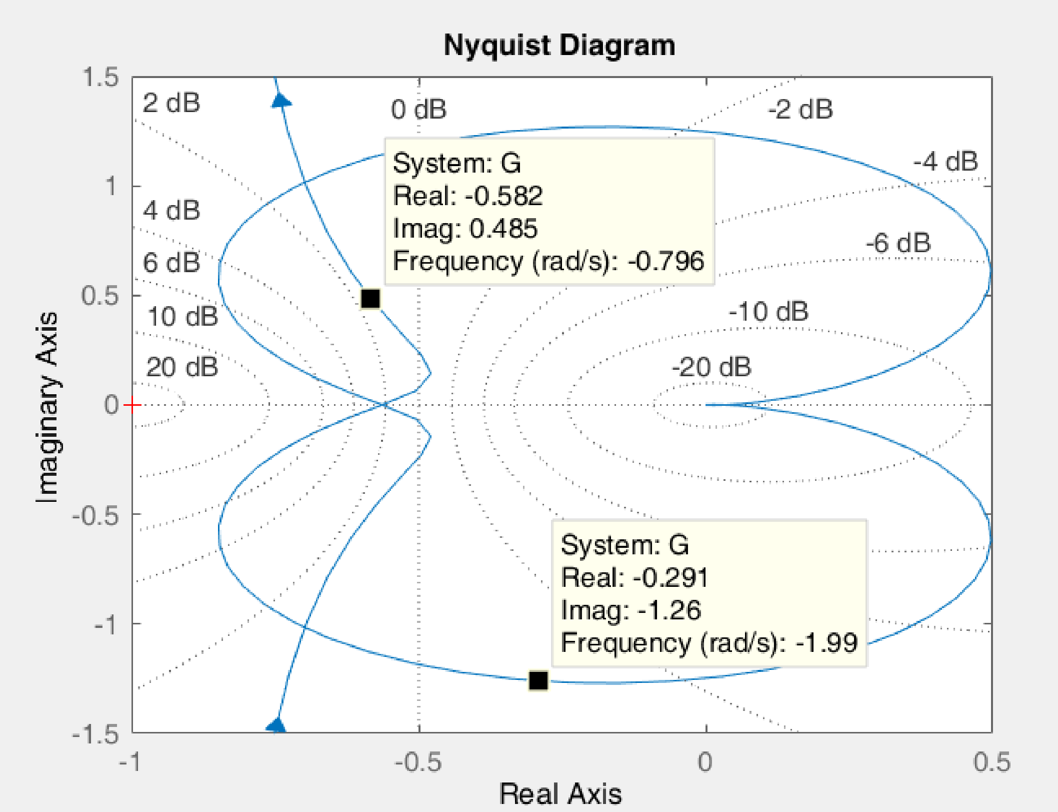
Command Window

```
>> clear
>> s=tf('s');
>> G=(s+8)/(s*(s^2+0.2*s+4)*(s+1)*(s+3));
>> nyquist(G),grid  %绘制Nyquist图并叠印等幅值圆
>> set(gca, 'ylim',[-1.5 1.5]) %根据需要手动选择纵坐标范围
```

五、线性系统的频域分析方法

Nyquist图的绘制方法

■ 例5-1：在nyquist曲线上单击鼠标可同时显示该点处的频率等信息。这样MATLAB赋予了Nyquist图新的功能，有助于系统的频域分析。



五、线性系统的频域分析方法

Bode图和Nichols图的绘制方法

- Bode图是一种常用的频率响应的分析方法，同时可以分析系统的幅值相位与频率之间的关系。
- 在MATLAB的控制系统工具箱中提供了bode()函数，可以直接绘制系统的Bode图，该函数常用的调用格式如下:

  >> bode(G)　　%不返回变量将自动绘制bode图

  >> bode (G,{ωm, ωM})　　%给定频率范围绘制bode图

  >> bode (G, ω)　　%给定频率向量ω绘制bode图

  >> [A, Φ, ω]=bode (G)　　%计算bode响应数值

  >> bode (G1,’-’,G2,’-.b’,G3,’:r’)　　%绘制几个系统的 bode图

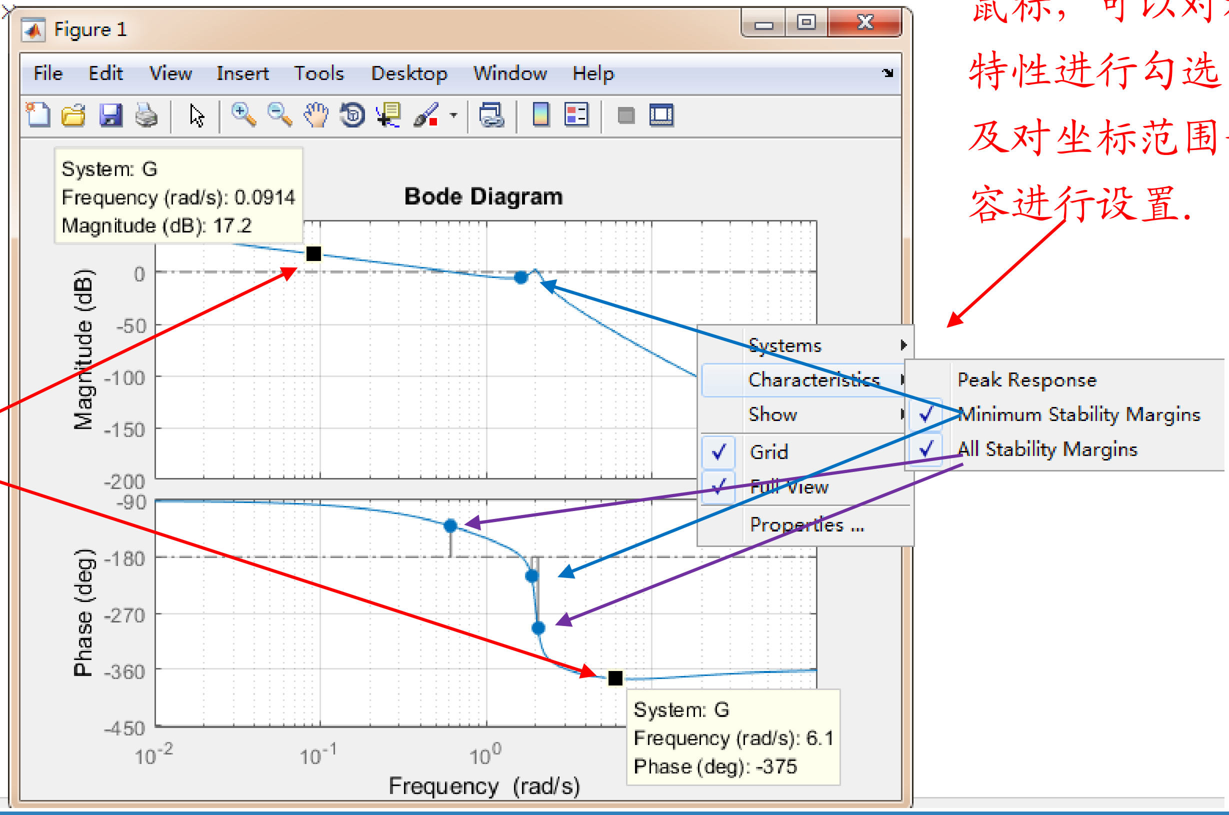
- Nichols图 nichols()函数的调用格式与Bode图调用格式完全一致。这时的**grid函数**可以叠印出等幅值曲线和等相位曲线。

五、线性系统的频域分析方法

Bode图和Nichols图的绘制方法

■ 例5-2:

```
>> s=tf('s');
>> G=(s+8)/(s*(s^2+0.2*s+4)*(s+1)*(s+3));
>> bode(G),grid
```



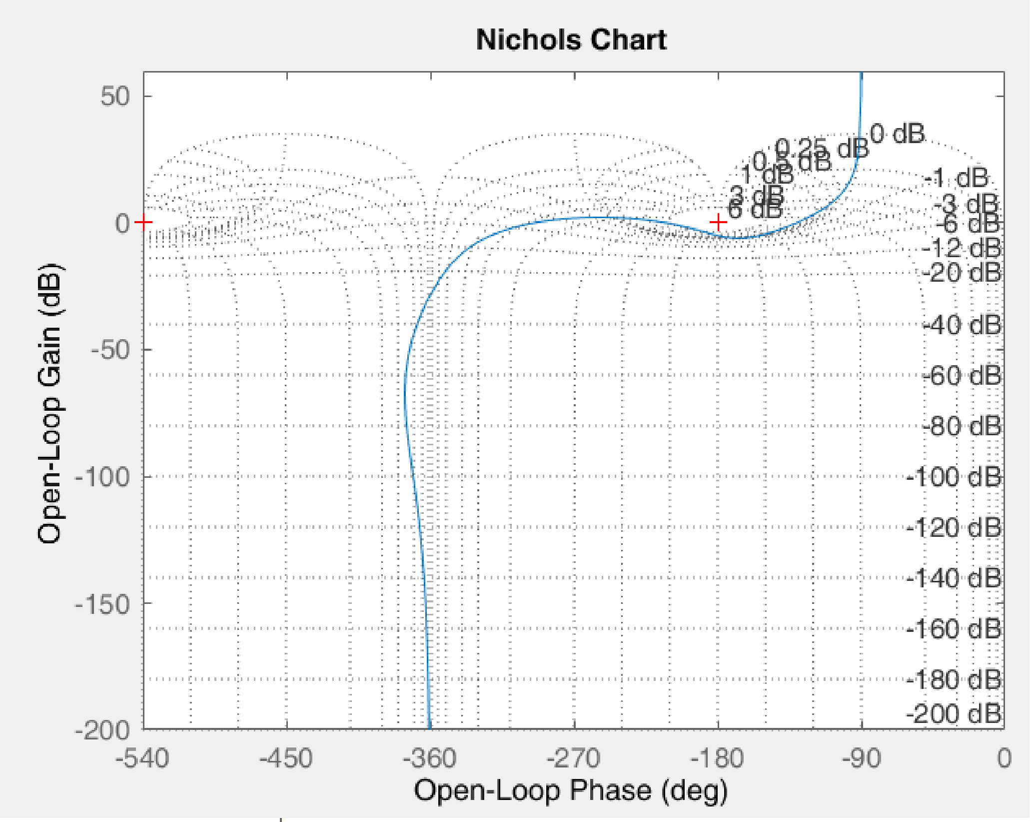
在bode图界面右击鼠标，可以对相关特性进行勾选，以及对坐标范围等内容进行设置.

曲线上单击可同时显示该点处的频率等信息。

五、线性系统的频域分析方法

Bode图和Nichols图的绘制方法

■ 例5-2:



Command Window

```
>> clear
>> s=tf('s');
>> G=(s+8)/(s*(s^2+0.2*s+4)*(s+1)*(s+3));
>> nichols(G),grid
```

五、线性系统的频域分析方法

离散系统的频域分析方法

- 对于离散系统H(Z)来说，可以将$z=e^{j\omega T}$代入传递函数模型，就可得出频率和增益$\hat{H}(j\omega)$之间的关系。MATLAB中提供的各种频率响应分析函数，如nyquist()等，同样直接适用于离散的系统模型。
- 例5-3：考虑离散系统的传递函数模型

$$G(z)=\frac{0.2*(0.3124z^3-0.5743z^2+0.3879z-0.0889)}{z^4-3.233z^3+3.9869z^2-2.2209z+0.4723}$$

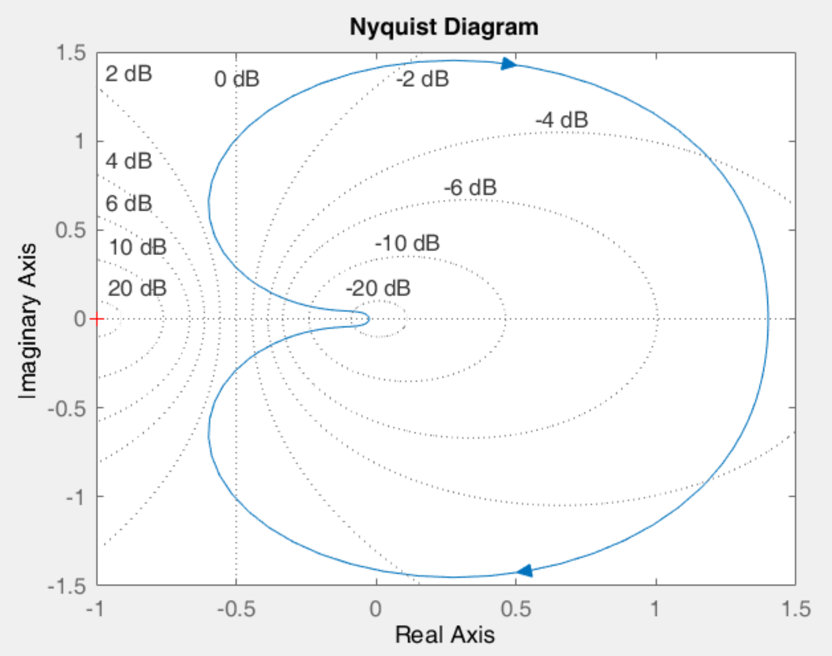
且已知系统的采样周期T=0.1s，则可以用下面的语句将其输入到MATLAB工作空间，并将系统的Nyquist图，Bode图直接绘制出来。

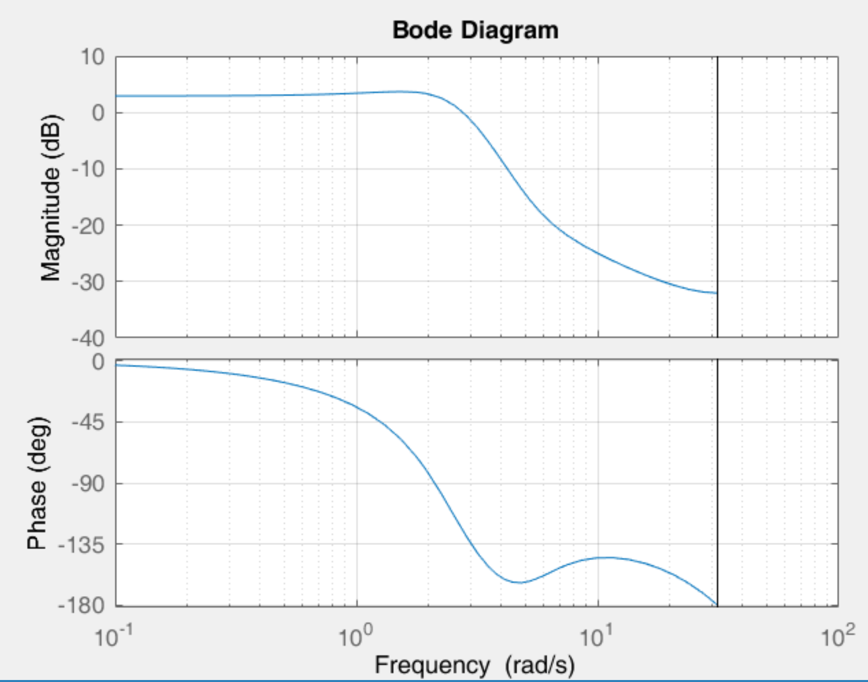
Command Window

```
>> num=0.2*[0.3124 -0.5743 0.3879 -0.0889];
>> den=[1 -3.233 3.9869 -2.2209 0.4723 ];
>> G=tf(num, den, 'Ts', 0.1);
>> nyquist(G); grid
>> bode(G),grid
```

五、线性系统的频域分析方法

离散系统的频域分析方法

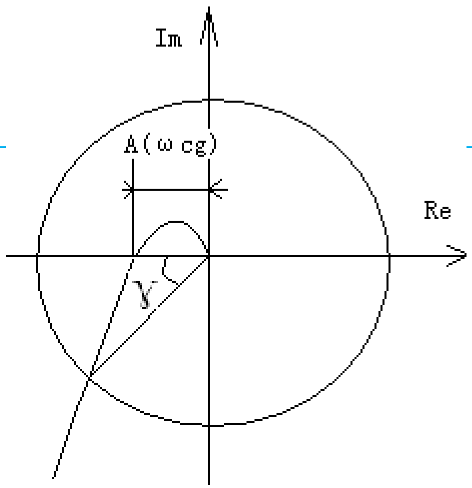




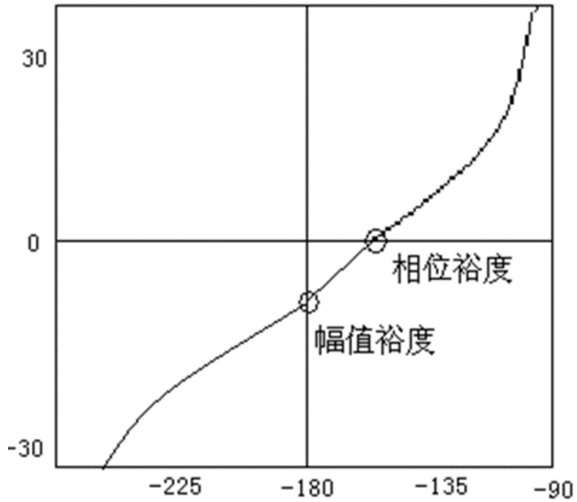
五、线性系统的频域分析方法

系统的幅值裕度和相位裕度

- 基于频域响应裕度的定量分析方法。
- 对于Nyquist图，若在频率ω_{cg}时与负实轴相交，则将该频率下幅值的倒数，即$G_m=1/A(\omega_{cg})$定义为系统的幅值裕度。
- 若与单位圆在ω_{cp}处相交，且记该频率下的相位角度为$\Phi(\omega_{cp})$，则系统的相位裕度定义为$\gamma=\Phi(\omega_{cp})-180°$ 。



Nyquist图



Nichols图

五、线性系统的频域分析方法

系统的幅值裕度和相位裕度

- 一般幅值裕度Gm越大，系统对扰动的抑制能力越强。
- 如果Gm < 1，则闭环系统不稳定。同样，相位裕度越大，系统对扰动的抑制能力越强。如果γ < 0，则闭环系统不稳定。
    - 如果Nyquist图不与负实轴相交，则系统的幅值裕度为无穷大。
    - 如果Nyquist图与负实轴在（-1，j0)与(0, j0)两点之间有若干个交点，则系统的幅值裕度以离（-1，j0),点最近的点为准。
    - 如果Nyquist图不与单位圆相交，则系统的相位裕度为无穷大。
    - 如果Nyquist图在第三象限与单位圆有若干个交点，则系统的相位裕度以离负实轴最近的点为准。

五、线性系统的频域分析方法

系统的幅值裕度和相位裕度

■ MATLAB中提供了margin()函数，可直接求取系统的幅值和相位裕度。调用格式为:

[Gm,γ, ωcg, ωcp]=margin(G);

- Gm幅值裕度,γ相位裕度, ωcg频率, ωcp剪切频率
- 若某个裕度为无穷大，则返回Inf，相应的频率值为NaN

五、线性系统的频域分析方法

系统的幅值裕度和相位裕度

■ 例5-3

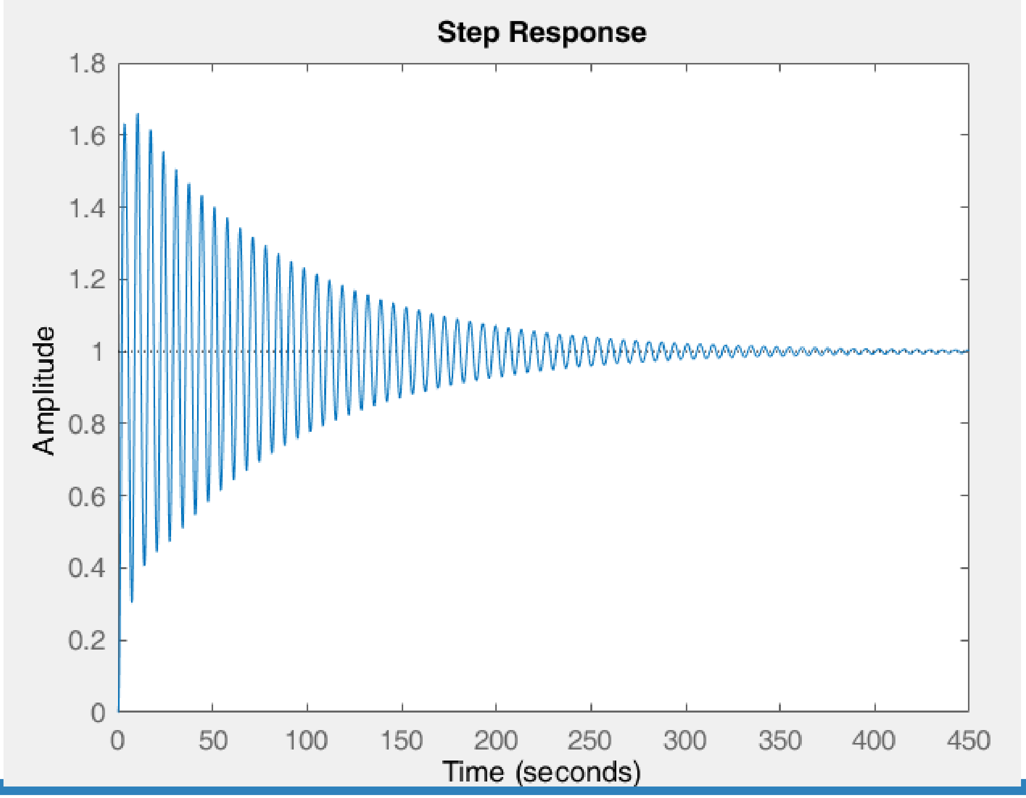
```
>> s=tf('s');
>> G=2.7778*(s^2+0.192*s+1.92)/(s*(s+1)^2*(s^2+0.384*s+2.56));
>> [gm,pm,wg,wp]=margin(G)
gm =
     1.1050

pm =
     2.0985

wg =
     0.9621

wp =
     0.9261
```

➢ 则系统的幅值裕度为1.105，频率为0.9621rad/s，相位裕度为2.0985，剪切频率为0.9261，由于幅值、相位裕度偏小，故系统的闭环响应将有强振荡。

五、线性系统的频域分析方法

Command Window

```
>> clear
>> s=tf('s');
>> G=2.7778*(s^2+0.192*s+1.92)/(s*(s+1)^2*(s^2+0.384*s+2.56));
>> step(feedback(G,1))
```



Thank you !

Question and Answer?