《自动控制原理》上机指导书

（能源与动力学院用）

南京航空航天大学能源与动力学院

二〇〇五年八月

1 概述

Matlab 是 Mathworks 公司于 1984 年推向市场的一套高性能的数值计算和可视化软件，它集数值分析、矩阵运算、信号处理和图形显示于一体。经过二十年的发展，Matlab 已成为一个国际公认的最优秀的科技应用软件，并且其强大的扩展功能更是为各个工程领域提供了分析和设计的基础，成为国际控制界应用最广的首选计算机工具。

Matlab 提供了框图式动态系统仿真工具 Simulink，用它可以方便地建立控制系统原型和控制对象模型，通过仿真不断地优化和改善你的设计。无论是离散的、连续的、条件执行的、多采样的或混杂的系统，Simulink 都是描述动态系统模型的最佳工具。

运行 Matlab 软件后出现界面如图 1.1 所示。

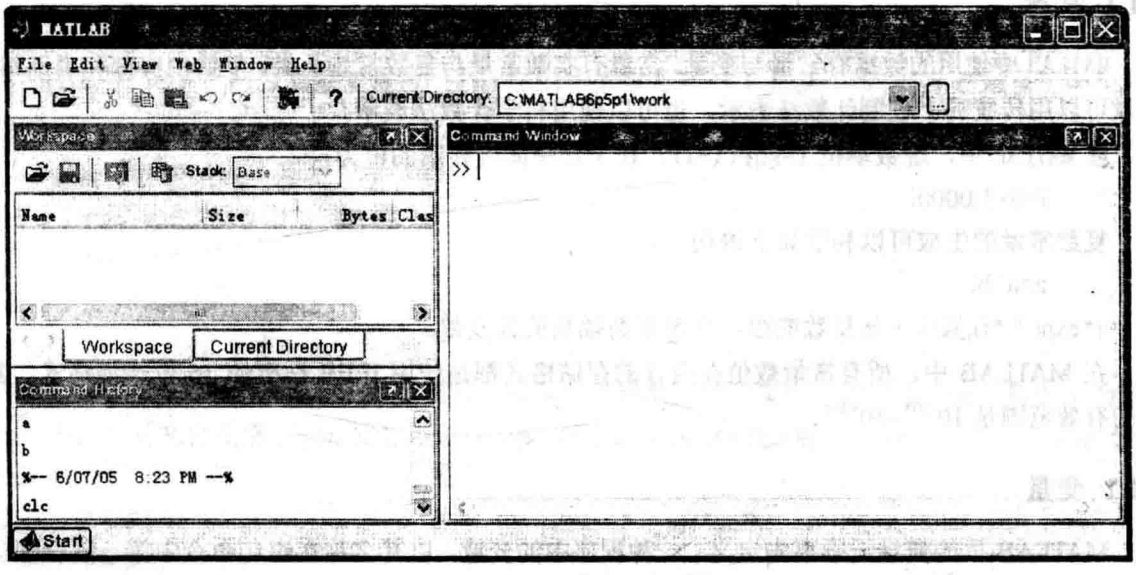

图 1.1 Matlab 命令窗口界面

主窗口为 Command Window，可将 Matlab 命令直接输入到本窗口中，按 Enter 键后就会执行相应的操作。左上侧的窗口是 Workspace（工作空间，即用来存储曾经使用过的 MATLAB 变量的内存空间）窗口，显示当前可应用变量的运算结果信息。左下侧是 Command History 窗口，显示你在 Command Window 中曾经输入过的 Matlab 命令。当你不需要某些窗口时，可直接点击本窗口上的☒关闭，若你想要的窗口没有显示出来，也可通过选取菜单 View 的下拉子菜单来打开窗口。你可以在 Command Window 中直接输入 Matlab 命令来显示一些结果和帮助信息，对于较长的程序，建议选取 File 菜单中的 new 命令，新建一个扩展名为.m 的文件，或直接在 Command Window 中键入命令 edit 并回车，命令执行后出现图形界面如图 1.2 所示。

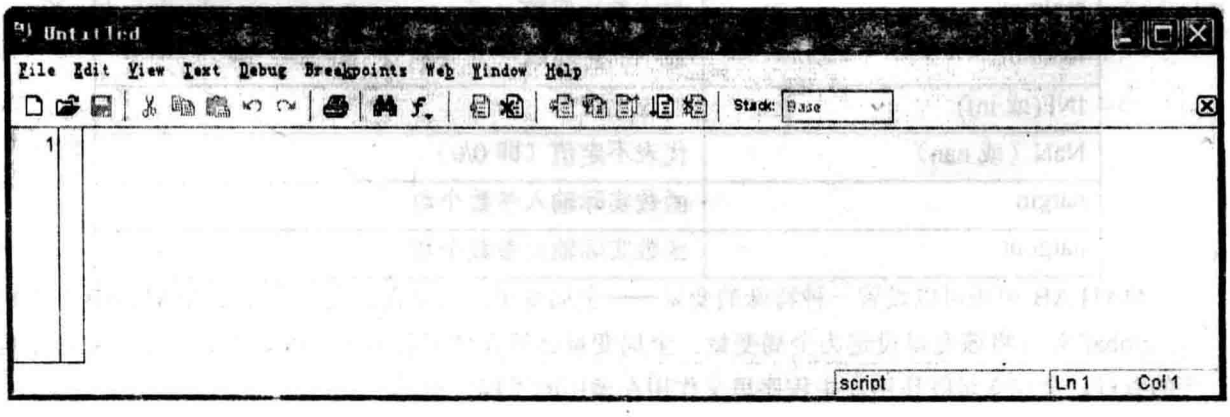

图 1.2 Matlab 程序编辑界面

可在这个窗口输入程序后保存，系统会自动将其扩展名存为.m，如果要运行此文件，直接选取菜单中的 Debug 菜单，并选其子菜单中的 run，或直接按 运行程序，运行结果或出错信息会显示在 Command Window 中，并且在 WorkSpace 中也会显示出变量名和变量的运行结果。在编写程序的过程中，要适当地加注释，注释语句前用“%”来表示，注释语句不参与程序编译和运行。

2 MATLAB 矩阵运算基础

2.1 常量与变量

2.1.1 常量

MATLAB 中使用的数据有常量与变量。常量有实数常量与复数常量两类，复数由实部和虚部组成。常量可以用传统的十进制计数法表示，也可以使用科学计数法表示。

在 MATLAB 中，虚数单位 i=sqrt(-1)，在工作空间内显示的值为：

i=0+1.0000i

复数常量的生成可以利用如下语句

z=a+bi

或 z=r*exp(θ *i),其中 r 是复数的模， θ 是复数幅角的弧度数。

在 MATLAB 中，所有常量数值在内存的存储格式都是使用 IEEE 标准的 16 位长型格式，其数值的有效范围是 10^{-308}~10^{+308}。

2.1.2 变量

MATLAB 里的变量无须事先定义。一种程序中的变量，以其名称在语句命令中第一次合法出现而定义。MATLAB 是大小写敏感的语言，例如，Abc 和 ABC 是不同的变量。

MATLAB 中设置了一些特殊的变量与常量，如表 2.1 所示。

表 2.1 MATLAB 的特殊变量与常量

变量名	功能说明
ANS（或 ans）	缺省变量名，以应答最近一次操作运算结果
i 或 j	虚数单位，定义为 $\sqrt{-1}$
pi	圆周率
eps	浮点的相对误差
realmax	最大的正实数
realmin	最小的正实数
INF(或 inf)	代表无穷大
NaN（或 nan）	代表不定值（即 0/0）
nargin	函数实际输入参数个数
nargout	函数实际输出参数个数

MATLAB 中还可以设置一种特殊的变量——全局变量，只要在该变量前添加 MATLAB 的关键字(“global”就可将该变量设定为全局变量。全局变量必须在使用前声明，即这个声明必须放在主程序的首行。全局变量既作用在主程序里又作用在调用的子程序里。

2.2 矩阵的输入方法

MATLAB 最初是作为矩阵运算而开发的软件，MATLAB 以复数矩阵为基本运算单元，既可以对它进行整体性处理，也可以对它的某个或某些元素进行处理。矩阵有多种输入方法：

（1）直接在表达式中列出元素，下面是常用的几种方式。若输入：

A=[1,2;3,4],B=[10,20,30],C= [1;2;3]

则得到

```
A=1 2
  3 4
B=10 20 30
C=1
  2
  3
```

MATLAB 定义了两个复数变量：i 和 j。它们的值均为 $\sqrt{-1}$。如果 i 和 j 不重新赋值，则保留这种定义；如果想把变量 a 变成 $\sqrt{-1}$，则用 a=sqrt(-1)。

下面给出复数矩阵的输入方法，例如键入：

B=[1+2i,1+5j;3+7j,5+8i]

则

```
B= 1.0+2.0i    1.0+5.0j
   3.0+7.0j    5.0+8.0i
```

(2) 建立在 M 文件中，如果矩阵过大，可以用文件来产生，文件可命名为 x.m。其中 x 为矩阵名，m 为特有的文件后缀。x.m 文件可以用 load 命令装入到 MATLAB 中，例如，有一个 b.m 文件，它的内容为：

```
3 5 6 8
1 4 5 6
```

如果用命令：load b.m，则可以得到矩阵 b:

```
b= 3 5 6 8
   1 4 5 6
```

(3)从小矩阵扩充成大矩阵，或用“:”从大矩阵中提取出小矩阵，例如键入

A=[1,2;3,4],B=[A;[10,14]],C=B(2:3,1:2),D=B(1,:),E=B(:,2)

则

```
A=1 2
  3 4
B=1   2
  3   4
 10   14
C= 3   4
  10   14
D=1 2
E=2
  4
 14
```

2.3 MATLAB 运算符

2.3.1 MATLAB 算术运算符

（1）加减：C=A+B，C=A-B。A，B 为两个 $n \times m$ 的矩阵。

（2）转置：C=A'，如果 A 是复阵，则 C 为 A 的共轭转置阵。

（3）翻转：翻转命令有 fliplr,flipud 和 rot90，下面举例说明它们的效果：

若键入

```
A=[1,4;9,7],B=fliplr(A),C=flipup(A),D=rot90(A)
```

则

```
A=1 4
  9 7
B=4 1
  7 9
C=9 7
  1 4
D=4 7
  1 9
```

（4）乘法：C=A*B。

（5）Kronecker 积：C=kron(A,B)，例 A=[1,2], B=[-1;2]，则 C=kron(A,B)的结果为：

```
C=-1  -2
   2   4
```

（6）除法：分左除和右除两种。左除用“\”符号实现，其格式为 A\B，相当于 $A^{-1}B$；右除用“/”符号实现，其格式为 A/B，相当于 BA^{-1}。

（7）乘方：如果 A 是方阵，则 A^x 表示 A 的乘方，其格式为 A^x，其中 x 可以是正整数、负整数和分数；MATLAB 也有另一种乘方运算：a^B，其中 a 是一个标量，B 是一个方阵。

（8）矩阵的点运算：这是 MATLAB 定义了一种特殊的运算，两个矩阵之间的点运算是矩阵对应元素的直接运算，例如：

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \quad B=\begin{bmatrix}7 & 8\\ 9 & 10\end{bmatrix} \quad A\bullet B=\begin{bmatrix}7 & 16\\ 27 & 40\end{bmatrix}$$

对矩阵部分元素的赋值与运算，如键入

```
A=[1,2;4,5],B=A;B(2,2)=100,C=B;C(2,3)=200
```

则

```
A=1   2
  4   5
B=1   2
  4   100
C=1 2   0
  4 100 200
```

上面的语句 B(2,2)=100 和 C(2,3)=200 分别用于对矩阵 B 和 C 的单个元素进行赋值，如果给出的行数或列数超出了原来矩阵的范围，MATLAB 会自动地扩展原来的矩阵，并将扩展后未赋值的矩阵元素设置为 0。

2.3.2 MATLAB 关系运算符和逻辑运算符

如表 2.2 和表 2.3 所示。

表 2.2 MATLAB 关系运算符

操作符	说明
==	等于
~=	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于

表 2.3 MATLAB 逻辑关系运算符

逻辑运算符	说明
&	逻辑和 and
|	逻辑或 or
~	逻辑非 not
	逻辑异或 xor

2.3.3 数组函数

在 MATLAB 中，对数组运算的函数可以分为基本数组函数和特殊数组函数两类。基本数组函数是指三角函数、对数函数、指数函数、复数运算函数等。这些函数执行时，遵循数组运算规则，即对数组每个元素进行同等操作。特殊数组函数满足标量运算规则。

2.4 控制语句

MATALB 提供的控制语句与一般高级语言类似。

（1）循环语句：for 和 while 语句，它们允许多级嵌套和互相嵌套。for 和 while 语句的基本结构是：

```
for 循环变量=表达式 1：表达式 2：表达式 3
  循环语句组
end

while(条件式)
  循环语句组
end
```

其中，表达式 1，2，3 分别给出了循环变量的初始值、增量、最终值。当表达式 2 为 1 时，表达式 2 可省略。例如 for i=1:1:5 可写成 for i=1:5。对于条件式来说，逻辑运算符是必要的，MATLAB 使用“&”，“|”，“~”分别表示“与”、“或”、“非”。

下面是两段等价的 MALAB 程序，这两段程序执行结束后，sum 的值都是 5.

```
sum=0;
for i=1:5
  sum=sum+1;
end

sum=0;i=1;
while(i<=5)
 sum=sum+1;i=i+1;
end
```

（2）条件转移语句：条件转移语句的基本结构是：

```
if(条件式)
```

```
语句组
end

if(条件式)
 语句组 1
else
 语句组 2
end

if(条件式)
 语句组 1
else if(条件式)
 语句组 2
end
```

（3）break 语句：循环过程的终止由 break 来完成。break 用来终止包含它的最里一层的循环过程，例如下面的程序段运行结束为 sum=6.

```
sum=0;
for i=1:10
 if(sum>5) break;end
 sum=sum+1;
end
```

2.5 图形功能

MATLAB 的二维图形功能：常用的一种是 plot()函数。其调用格式为：

plot(x1,y1,选项 1, x2,y2,选项 2，……)

其中，MATLAB 提供的选项见表 2.4，这些选项对其它图形函数也是适用的。

表 2.4 MATLAB 绘图命令的各种选项

选项	意义	选项	意义	选项	意义	选项	意义
'-'	实线	'*'	星号	'g'	绿色	'b'	蓝色
':'	点线	'o'	圆圈	'y'	黄色	'-.'	点划线
'r'	红色	'_'	虚线	'.'	点号	'x'	叉号

下面是一段语句串实例：

```
t=0:0.1:10;x=sin(t);y=cos(t);plot(t,x,t,y,':');grid;xlabel('X axis');
ylabel('Y axis');title('My plot');text(2.4,0.85,'x=sin(t)');
```

其执行结果见图 2-1。其中，正弦曲线为实线，而余弦曲线为虚线型。另外，grid()函数是在图中加网格线，xlabel()和 ylabel()函数是在 x 轴和 y 轴上标注字符串，title()函数标注图形的标题，text()是在图中标注文字。

在二维图形函数中，还在其他常用的函数：

polar(theta,tho,选项)：绘制极坐标曲线。theta 和 tho 分别为角度和幅值向量。

bar(x,y,选项)：绘制直方图。

subplot(m,n,k)：在一个窗口中开出 $m \times n$ 个图形窗口，k 为窗口序号。

loglog(x,y,选项)，semilogx(x,y,选项)，semilogy(x,y,选项)：绘制对数和半对数坐标曲线图。loglog()绘制对数坐标的曲线，semilogx()绘制横坐标为对数坐标的曲线，semilogy()绘制纵坐标为对数坐标的

曲线。

x=logspace(x1,x2,n)：按对数等间距地产生一个向量，例如要产生 10^{-2}~10^{2} 的 100 个点，则用 x=logspace(-2,2,100)。

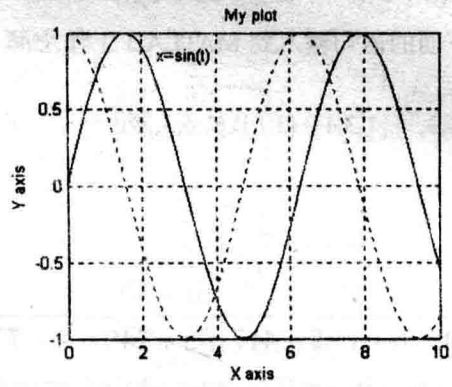

图 2.1 图形实例结果

3 用 Matlab 来表示传递函数

线性定常系统的传递函数（transfer function），定义为零初始条件下，系统输出量的拉氏变换与输入量的拉氏变换之比。

3.1 分子分母多项式模型

Matlab 可以方便的用于分析由传递函数模型描述的系统，由于传递函数具有多项式之比的形式，因此应首先学会如何应用 Matlab 进行多项式运算。

在 MATLAB 中采用行向量表示多项式，行向量内存存放按降幂排列的多项式系数。在 MATLAB 中建立多项式就是输入多项式系数行向量

多项式 $P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$ 的系数行向量为：

$$P=[a_0 \quad a_1 \quad \cdots \quad a_{n-1} \quad a_n]$$

函数 tf 根据给定的分子和分母多项式，创建系统模型为传递函数（TF）对象。

【例 3-1】以一个简单的传递函数(transfer function)模型为例

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

在 Matlab 中，传递函数的分子(numerator)和分母(denominator)多项式是分别给定的。可以由下面的命令输入到 MATLAB 工作空间去，其中 num 和 den 分别为系统分子和分母多项式系统的降幂排列。

```
num=[1,5];                %分子多项式行向量
den=[1,2,3,4,5];          %分母多项式行向量
G=tf(num,den)             %返回G为TF对象
```

输出结果为：

Transfer function

$$G=\frac{s+5}{s\ ^4+2s^3+3s^2+4s+5}$$

【例 3-2】下面给出一个稍微复杂一些的传递函数模型

$$G(s)=\frac{6(s+5)}{(s^2+3s+1)^2(s+6)(s^3+6s^2+5s+3)}$$

该传递函数模型可以通过下面的语句输入到 MATLAB 工作空间

```
num=6*[1,5];
den=conv(conv(conv([1,3,1],[1,3,1]),[1,6]),[1,6,5,3]);
tf(num,den)
```

输出结果为：

Transfer function

$$G(s)=\frac{6s+30}{s\text{ ^}8+18s\text{^}7+124s\text{^}6+s\text{ ^}8+417s\text{^}5+740s\text{^}4+729s\text{^}3+437s\text{^}2+141s+18}$$

其中 conv()函数为标准的 MATLAB 函数。若给定两个包含两个多项式系数的行向量，conv 返回一个含有两个多项式乘积多项式系数的行向量。conv()函数允许任意地多层嵌套，从而表示复杂的计算。用户应该注意其括号的匹配，否则会得出错误信息与结果。

3.2 零极点模型

函数 zpk 根据给定的零点(zero)、极点(pole)和增益(gain)，创建系统模型为 zpk 对象。

【例 3-3】假设系统的零极点模型为

$$G(s)=6\frac{(s+1.9294)(s+0.0353\pm0.9287i)}{(s+0.9567\pm1.2272i)(s-0.0433\pm0.6412i)}$$

则该模型可以由下面的语句输入到 MATLAB 工作空间中

```
j=sqrt(-1);    Kgain=6;
Z=[-1.9294;-0.0353+0.9287*j;-0.0353-0.9287*j];
P=[-0.9567+1.2272*j;-0.9567-1.2272*j;+0.0433+0.6412*j;+0.0433-0.6412*j];
zpk(Z, P, Kgain)
```

程序输出为：

Zero/pole/gain:

$$G(s)=\frac{6(s+1.929)(s\text{^}2+0.0706s+0.8637)}{(s\text{^}2-0.0866s+0.413)(s\text{^}2+1.1913s+2.421)}$$

3.3 结构图模型

(1) 串联连接结构（图 3.1）

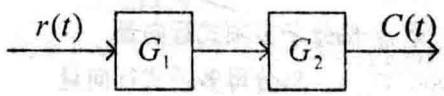

图 3.1 串联

在串联连接下，整个系统的传递函数为 $G(s)=G_1(s)G_2(s)$。对单变量系统来说，这两个模块是可以互换的，亦即 $G_1G_2=G_2G_1$，对多变量系统来说，一般不具备这样的关系。

【例 3-3】已知系统 $G_1(s)=\frac{1}{s^2+2s+1}$，$G_2(s)=\frac{1}{s+1}$，则 $G_1(s)$ 和 $G_2(s)$ 串联后系统的模型可以由下列 MATLAB 命令得出：

```
G1=tf(1, [1, 2, 1]);
G2=tf(1, [1, 1]);
G=G1*G2;
```

(2) 并联连接结构（图 3.2）

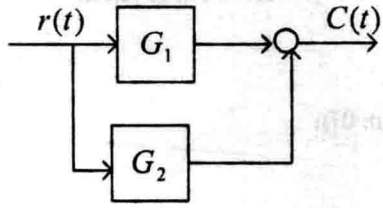

图 3.2 并联

在并联连接下，整个系统的传递函数为 $G(s)=G_1(s)+G_2(s)$。

【例 3-4】已知系统 $G_1(s)=\dfrac{1}{s^2+2s+1}$，$G_2(s)=\dfrac{1}{s+1}$，则 $G_1(s)$ 和 $G_2(s)$ 并联后系统的模型可以由下列 MATLAB 命令得出：

```
G1=tf(1, [1, 2, 1]); G2=tf(1, [1, 1]);
G=G1+G2;
```

(3) 反馈连接结构（图 3.3）

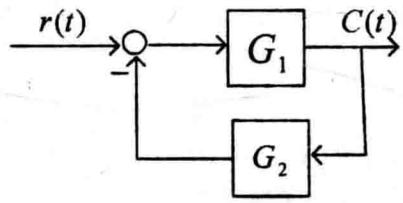

图 3.3 反馈

控制系统工具箱中提供了 feedback()函数，用来求取反馈连接下总的系统模型，即系统的闭环传递函数模型。该函数的调用格式为：

```
G=feedback(G1, G2, Sign)
```

其中变量 Sign 用来表示正反馈结构，若 Sign=－1 则将求出负反馈系统的模型，若省略 Sign 变量，即 G=feedback(G1, G2)，则仍将表示负反馈结构。G1 和 G2 分别表示前向模型和反向模型。

4 采用部分分式分解法求 Laplace 反变换及系统时域分析

4.1 Laplace 反变换及系统时域响应的解析解

MATLAB 语言提供了一个 residue( )函数来对有理传递函数(num, den)进行部分分式展开，根据系统的部分分式展开则可方便地得出系统时域响应的解析解。该函数的调用格式为

```
[R, P, K]=residue(num, den)
```

P 代表传递函数的极点，R 代表相应极点的部分分式分解系数（即留数），而 K 为部分分式展开后的余项。

【例 4. 1】考虑如下的传递函数模型

$$G(s)=\frac{s^3+7s^2+24s+24}{s^4+10s^3+35s^2+50s+24}$$

若输入 $R(s)=1(t)$，则系统的输出为：

$$C(s)=G(s)R(s)=\frac{s^3+7s^2+24s+24}{s^4+10s^3+35s^2+50s+24}\cdot\frac{1}{s}=\frac{s^3+7s^2+24s+24}{s^5+10s^4+35s^3+50s^2+24s}$$

由下面的 MATLAB 语句可以将 $C(s)$ 展开为部分分式：

```
num=[1, 7, 24, 24];
den=[1, 10, 35, 50, 24];
[r, p, k]=residue(num, [den, 0]);
```

在命令窗口运行后得：

```
r=
−1.0000
2.0000
−1.0000
−1.0000
1.0000
p=
−4.0000
−3.0000
−2.0000
−1.0000
     0
k=
     [ ]
```

即展开后的部分分式为：

$$C(s)=\frac{-1}{s+4}+\frac{2}{s+3}+\frac{-1}{s+2}+\frac{-1}{s+1}+\frac{1}{s+0}$$

对上式进行拉氏反变换，可得出系统的阶跃响应解析解：

$$c(t)=-e^{-4t}+2e^{-3t}-e^{-2t}-e^{-t}+1$$

【例 4.2】考虑一个不稳定有重极点的数学模型模型的响应

$$G(s)=\frac{s+1}{(s-1)(s+3)(s+4)^2}$$

若输入 $R(s)=1(t)$，则系统的输出为：

$$C(s)=G(s)R(s)=\frac{s+1}{(s-1)(s+3)(s+4)^2}\frac{1}{s}$$

由下面的 MATLAB 语句可以将 $C(s)$ 展开为部分分式，

```
num=[1, 1];
den=conv([1, -1],conv([1,3],conv([1,4],[1,4])));
[r, p, k]=residue(num, [den, 0]);
```

在命令窗口运行后得

```
r=
   0.1675
   0.1500
  -0.1667
```

```
0.0200
-0.0208
p=
   -4.0000
   -4.0000
   -3.0000
    1.0000
         0
k=
    []
```

即展开后的部分分式为：

$$C(s)=\frac{0.1675}{s+4}+\frac{0.1500}{(s+4)^2}+\frac{-0.1667}{s+3}+\frac{0.02}{s-1}+\frac{-0.0208}{s}$$

对上式进行拉氏反变换，可得出系统的阶跃响应解析解，

$$c(t)=0.1675e^{-4t}+0.1500te^{-4t}-0.1667e^{-3t}+0.02e^{t}-0.0208$$

其中$0.02e^{t}$在时间趋于无穷大时，也趋于无穷大，使系统的输出信号趋于无穷大，故系统不稳定。

上式中留数 r 是用小数表示的，下面的 Matlab 语句可以将前面的这些系数用分数的形式表示

```
[rn, rd]=rat(r)
```

输出为

```
ans=
    67   400
     3    20
    -1     6
     1    50
    -1    48
```

从上面的解可以看出，其输出的数学表达式为

$$c(t)=\frac{67}{400}e^{-4t}+\frac{3}{20}te^{-4t}-\frac{1}{6}e^{-3t}+\frac{1}{50}e^{t}-\frac{1}{48}$$

【例 4.3】我们再来看一个余项 K 不等于 0 的例子。设系统的模型为

$$G(s)=\frac{s^3+2s^2+3s+4}{(s+1)^3}$$

则其部分分式展开为：

```
[r, p, k]=residue([1, 2, 3, 4], [1, 3, 3, 1])
```

输出

```
r=
   -1.000
    2.000
    2.000
p=
   -1.000
   -1.000
-1.000
k=
    1.000
```

即部分分式展开为：

$$G(s)=\frac{-1}{s+1}+\frac{\cdot 2}{(s+1)^2}+\frac{2}{(s+1)^3}+1$$

其 Laplace 反变换为

$$g(t)=-e^{-t}+2te^{-t}+2t^2e^{-t}+t$$

【例 4.4】带有复极点的传递函数

$$G(s)=\frac{3s+2}{2s^3+4s^2+5s+1}$$

MATLAB 语句：

```
num=[3 2];
den=[2 4 5 1];
residue(num,den);
```

输出结果为

```
r =
  -0.1867 - 0.5526i
  -0.1867 + 0.5526i
   0.3734
p =
  -0.8796 + 1.1414i
  -0.8796 - 1.1414i
  -0.2408
k =
     [ ]
```

可写出部分分式展开为：

$$G(s)=\frac{-0.1867-j0.5526}{s+0.8796-j1.1414}+\frac{-0.1867+j0.5526}{s+0.8796+j1.1414}+\frac{0.3743}{s+0.2408}$$

4.2 系统性能指标的计算

典型的时域指标包括主导极点的阻尼比、阶跃响应的上升时间、超调量、调节时间、对稳态信号的跟踪能力和扰动抑制能力。

RPI 函数 tstats（）可以计算出系统的超调量、峰值时间、上升时间和调节时间（±2%误差带），tstats（）函数带 3 个参数：第一个为一个时间的列向量；第二个为系统阶跃响应测列向量；第三个为系统稳态参考值。其具体用法通过例 4.5 演示。

【例 4.5】设闭环系统传递函数为

$$\phi(s)=\frac{50}{s^3+21.01s^2+20.21s+50.2}$$

若求系统的性能指标，则可通过如下 Matlab 语句来实现，程序名 pfi.m

```
%pfi.m     to calculate the performance index of the system
num=50;
den=[1 21.01 20.21 50.2];
Gf=tf(num, den);
t=0:0.05:20]';
yr=step(Gf, t);
plot(t, yr); grid
[Mo, tp, tr, ts2, ess]=tstats(t, yr, 1)      %Mo 为超调量，tp 为峰值时间，tr 为上升时间，
```

%ts2 为调节时间，ess 为稳态误差

dump(Gf)　%返回闭环系统的极点以及极点处的阻尼比和自然频率

程序运行后除输出阶跃响应曲线外，还输出

Mo=39.4817%;

tp=2.1s

tr=0.827s

ts2=9s

ess=0.40044%

4.3 线性系统的稳定性分析

直接判断法：线性系统稳定的充分必要条件是：闭环系统特征方程的所有根均具有负实部；或者说，闭环传递函数的极点均严格位于 s 的左半平面。

利用 Matlab 我们可以方便地求出系统的闭环特征根或闭环极点。

函数 pole 计算一个给定的线性定常系统的极点；函数 pzmap 显示所有复平面的零极点，函数 roots 根据给定一个包含多项式 P(s)系数的行向量计算 P(s)=0 的根。

【例 4.5】假设系统的传递函数模型为：

$$G(s)=\frac{s^3+7s^2+24s+24}{s^4+10s^3+35s^2+50s+24}$$

则可以采用下面三种方法直接判断系统的稳定性。

```
G=tf([1,7,24,24],[1,10,35,50,24]);        %建立线性定常系统为 TF 对象 G
```

方法一 直接求分母多项式的根：

```
roots(G.den{1})                  %G.den{1}表示 TF 对象 G 的 den 属性的第一行向量
```

方法二 求其零极点形式的表达式，并将极点表达出来 ：

```
G1=zpk(G); G1.p{1}               %G1 为 zpk 对象，G1.p{1}表示 G1 极点的第一行向量
```

方法三：

```
pole(G)                          %求 G 的极点
```

【例 4.6】假设系统的传递函数模型为：

$$G(s)=\frac{4s^2+36s+32}{s^4+10s^3+30s^2+40s+24}$$

MATLAB 程序：

```
num=[4 36 32];
den=[1 10 30 40 24];
G=tf(num,den);
pzmap(G)                         %绘制 TF 对象的零极点图
```

4.4 系统时域响应的图形解

时域性能指标一般由系统在给定输入信号下的瞬态响应给出。一般情况下，我们用阶跃响应来定义系统的动态性能指标。下面给出在自然频率相同、阻尼系数不同和阻尼系数相同、自然频率不同的情况下，系统的单位阶跃响应。若计算系统的单位脉冲响应，只需将程序中的 step 函数换成 impulse 函数就可以了。

【例 4.7】设二阶系统闭环传递函数为

$$G(s)=\frac{3s+2}{2s^3+4s^2+5s+1}$$

求单位阶跃响应。

```
G=tf([3 2],[2 4 5 1]);
step(G);
```

【例 4.8】设二阶系统闭环传递函数为

$$T(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}=\frac{1}{\frac{s^2}{\omega_n^2}+\frac{2\zeta s}{\omega_n}+1}$$

```
%计算二阶系统的单位阶跃响应
clear          %清除工作空间的所有变量
clc            %清屏
t=[0:0.1:20];  %仿真时间 20s，仿真步长 0.1s
               %生成一个行向量 t，t=[0   0.1   0.2   0.3   0.4 ..... 20.0]
num=1;         %闭环传递函数分子多项式系数
wn=1;
zeta1=0.1; den1=[1/wn/wn   2*zeta1/wn   1];  %闭环传递函数分母多项式系数
zeta2=0.2; den2=[1/wn/wn   2*zeta2/wn   1];
zeta3=0.4; den3=[1/wn/wn   2*zeta3/wn   1];
zeta4=0.7; den4=[1/wn/wn   2*zeta4/wn   1];
zeta5=1.0; den5=[1/wn/wn   2*zeta5/wn   1];
zeta6=2.0; den6=[1/wn/wn   2*zeta6/wn   1];
[y1,x,t]=step(num,den1,t);                %计算系统的单位阶跃响应曲线
[y2,x,t]=step(num,den2,t);                %  x 为与 t 对应的输入单位阶跃函数值
[y3,x,t]=step(num,den3,t);                %  y1,y2,y3,y4,y5,y6 为输出响应值
[y4,x,t]=step(num,den4,t);
[y5,x,t]=step(num,den5,t);
[y6,x,t]=step(num,den6,t);
figure(1)                                 % 打开第一张曲线图
clc
plot(t,y1,t,y2,t,y3,t,y4,t,y5,t,y6);      % 在一张图上同时画出
%      t-y1, t-y2, t-y3, t-y4, t-y5, t-y6 曲线
xlabel('wn t'), ylabel('Y(t)');           % 标明 x 和 y 坐标名称
title('zeta=0.1,0.2,0.4,0.7,1.0,2.0'),grid    % 写上标题，并画上网格
zeta=1;
t=[0:0.1:20];
wn1=0.5;   den1=[1/wn1/wn1 2*zeta/wn1 1];
wn2=1;     den2=[1/wn2/wn2 2*zeta/wn2 1];
wn3=2;     den3=[1/wn3/wn3 2*zeta/wn3 1];
wn4=4;     den4=[1/wn4/wn4 2*zeta/wn4 1];
wn5=8.0;   den5=[1/wn5/wn5 2*zeta/wn5 1];
wn6=16.0;  den6=[1/wn6/wn6 2*zeta/wn6 1];
[y1,x,t]=step(num,den1,t);
[y2,x,t]=step(num,den2,t);
[y3,x,t]=step(num,den3,t);
```