

实验一 产生信号波形的仿真实验

一、实验目的：熟悉 MATLAB 软件的使用，并学会信号的表示和以及用 MATLAB 来产生信号并实现信号的可视化。

二、实验时数：3 学时+3 学时（即两次实验内容）

三、实验内容：

信号按照自变量的取值是否连续可分为连续时间信号和离散时间信号。对信号进行时域分析，首先需要将信号随时间变化的规律用二维曲线表示出来。对于简单信号可以通过手工绘制其波形，但对于复杂的信号，手工绘制信号波形显得十分困难，且难以绘制精确的曲线。

在 MATLAB 中通常用三种方法来产生并表示信号，即（1）用 MATLAB 软件的 funtool 符合计算方法（图示化函数计算器）来产生并表示信号；（2）用 MATLAB 软件的信号处理工具箱（Signal Processing Toolbox）来产生并表示信号；（3）用 MATLAB 软件的仿真工具箱 Simulink 中的信号源模块。

（一）用 MATLAB 软件的 funtool 符合计算方法（图示化函数计算器）来产生并表示信号

在 MATLAB 环境下输入指令 funtool，则回产生三个视窗。即

figure No.1：可轮流激活，显示 figure No.3 的计算结果。

figure No.2：可轮流激活，显示 figure No.3 的计算结果。

figure No.3：函数运算器，其功能有：f, g 可输入函数表达式；x 是自变量，在缺省时在 $[-2\pi, 2\pi]$ 的范围内；自由参数是 a；在分别输入完毕后，按下面四排的任一运算操作键，则可在 figure No.1 或 figure No.2 产生相应的波形。

➤ 学生实验内容：

产生以下信号波形

$3\sin(x)$ 、 $5\exp(-x)$ 、 $\sin(x)/x$ 、 $1-2\text{abs}(x)/a$ 、 $\sqrt{a*x}$

（二）用 MATLAB 软件的信号处理工具箱（Signal Processing Toolbox）来产生并表示信号

一种是用向量来表示信号，另一种则是用符合运算的方法来表示信号。用适当的 MATLAB 语句表示信号后，可以利用 MATLAB 的绘图命令绘制出直观的信号波形。

1. 向量表示法

对于连续时间信号 $f(t)$ ，可以用两个行向量 f 和 t 来表示，其中向量 t 是形如 $t=t_1:p:t_2$ 的 MATLAB 命令定义的时间范围向量， t_1 为信号起始时间， t_2 为信号终止时间， p 为时间间隔。向量 f 为连续信号 $f(t)$ 在向量 t 所定义的时间点上的样值。

下面分析连续时间信号 $f(t)=Sa(t)=\sin(t)/t$ ，可用如下的两个变量表示：

$$t=-10:1.5:10$$

$$f=\sin(t)./t$$

命令运行结果为：

```

t =
Columns 1 through 8
-10.0000    -8.5000    -7.0000    -5.5000    -4.0000    -2.5000    -1.0000     0.5000
Columns 9 through 14
     2.0000     3.5000     5.0000     6.5000     8.0000     9.5000
f =
Columns 1 through 8
-0.0544    0.0939    0.0939   -0.1283   -0.1892    0.2394    0.8415    0.9589
Columns 9 through 14
     0.4546    -0.1002    -0.1918     0.0331     0.1237    -0.0079

```

用上述向量对连续信号进行表示后，就可以用 `plot` 命令来绘制出信号的时域波形。`plot` 命令可将点与点间用直线连接，当点与点间的距离很小时，绘出的波形就成了光滑的曲线。

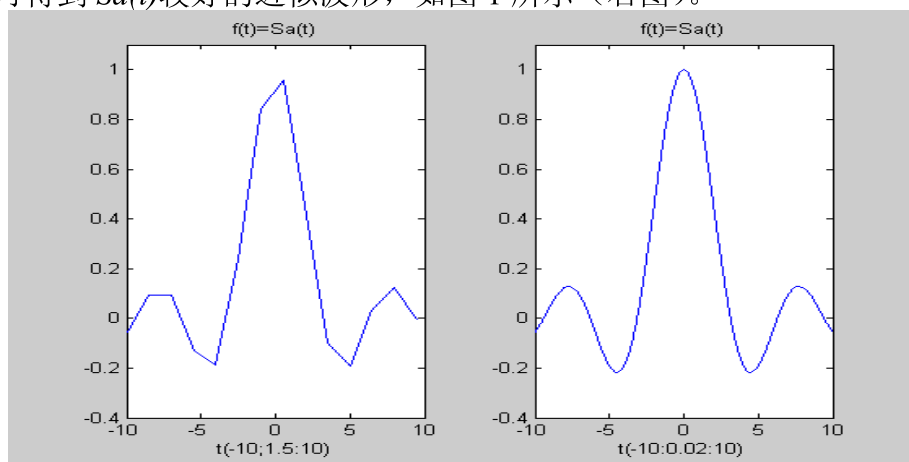
MATLAB 命令如下：

```

plot(t,f);
title('f(t)=Sa(t)');
xlabel('t');
axis([-10,10,-0.4,1.1]);

```

绘出的信号波形如图 1 所示(左图)。当把时间间隔 p 取得更小(例如为 0.02)时，就可得到 $Sa(t)$ 较好的近似波形，如图 1 所示(右图)。



如图 1 $Sa(t)$ 的近似波形

➤ 用以下程序可产生正弦波：

```

t=0:0.001:50;
y=sin(2*pi*50*t);
plot(t(1:50),y(1:50));

```

➤ 用以下程序可产生加入随机噪声的正弦波：

```

t=0:0.001:50;
y=sin(2*pi*50*t);
s=y+randn(size(t));
plot(t(1:50),s(1:50));

```

➤ 用以下程序可产生周期方波：

```

t=0:0.001:2.5;
y=square(2*pi*30*t);
plot(t(1:50),y(1:50));

```

- 用以下程序可产生周期锯齿波：

```
t=0:0.001:2.5;
y=sawtooth(2*pi*30*t);
plot(t,y);
axis([0 0.2 -1 1]);
```

- 用以下程序可产生 sinc 函数：

```
x=linspace(-5,5);
y=sinc(x);
plot(x,y);
```

- 用以下程序可产生 Dirichlet 函数：

```
x=linspace(0,4*pi,300);
y1=diric(x,7);
y2=diric(x,8);
subplot(1,2,1);plot(x,y1);
subplot(1,2,2);plot(x,y2);
```

2. 符合运算表示法

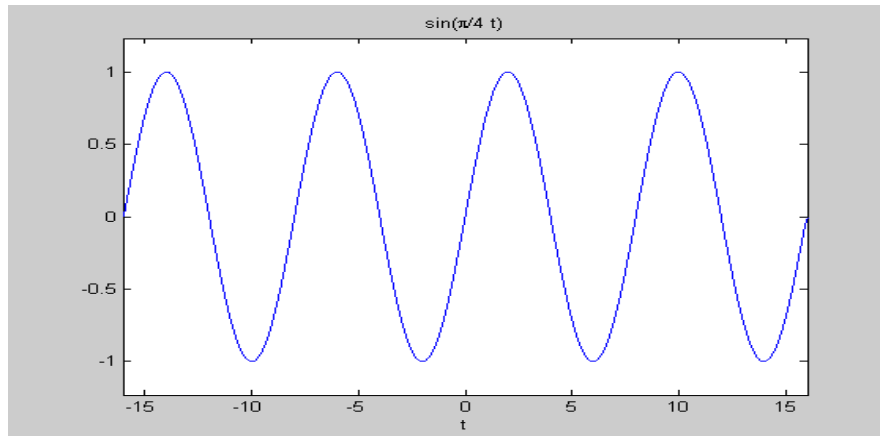
如果信号可以用一个符号表达式来表示它，则我们可用 `ezplot` 命令（缺省的区间为 $[-2\pi, 2\pi]$ ）绘制出信号的波形，例如对于连续信号 $f(t)=\sin(\pi t/4)$ ，我们可以用符号表达式表示为：

```
f=sym('sin(pi/4*t)');
f=
sin(pi/4*t);
```

然后用 `ezplot` 命令绘制其波形：

```
ezplot(f, [-16, 16]);
```

该命令绘制的信号波形如图 2 所示，



如图 2 正弦信号波形图

- 学生实验内容

- 1、改用其它的信号来练习使用向量表示法和符号运算表示法来绘制信号波形，达到对两种方法的熟练掌握。

（三）用 MATLAB 软件的仿真工具箱 Simulink 中的信号源模块

在 MATLAB 的命令视窗下输入 `simulink` 指令，则会打开 `untitled` 和 `library simulink` 两个视窗。`library simulink` 有 7 个子库，其中 `source` 是信号源子库，`sinks`

是显示器子库。子库中的任何模块都可以拖动到 **untitled** 视窗中，用鼠标把模块用连线按输入输出关系连接起来，就构成了仿真系统。在 **untitled** 视窗的菜单选 **simulation** 中的 **start**，开始进行仿真，仿真执行完毕后，示波器上会显示出信号波形。

四、离散时间序列波形产生并绘制

一般来说，离散时间信号用 $f(k)$ 表示，其中变量 k 为整数，代表离散的采样时间点， $f(k)$ 可表示为： $f(k)=\{\cdots f(-2), f(-1), f(0), f(1), f(2) \cdots\}$ 。在用 MATLAB 绘制离散时间信号波形时，要使用专门绘制离散数据的 **stem** 命令，而不用 **plot** 命令。

如序列 $f(k)=\{1, 2, -1, 3, 2, 4, -1\}$

↑
 $k=0$

在 MATLAB 中应表示为：

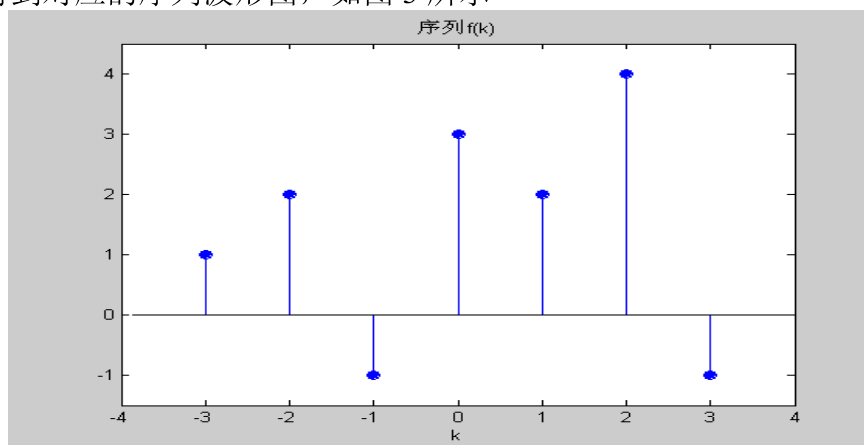
$k=[-3,-2,-1,0,1,2,3]$ 或是 $k=-3:3$;

$f=[1,2,-1,3,2,4,-1]$;

用如下 **stem** 命令绘图：

`stem(k,f,'filled');`
`axis([-4,4,-1.5,4.5]);`

则得到对应的序列波形图，如图 3 所示



如图 3 离散时间信号波形

➤ 学生实验内容：

试用 MATLAB 绘制两正弦序列

$$f_1(k)=\cos(k \pi/8), f_2(k)=\cos(2k)$$

的时域波形，观察它们的周期性，并验证是否与理论分析结果相符？（提示：并非所有的离散时间正弦序列信号都是周期的，不同于连续时间正弦信号）

五、实验要求：

- 1、按照实验指导书中的内容自己练习一遍；重新设置参数，例如：频率、周期、幅值、相位、显示时间段、步长、加噪等等，再按实验指导书上的内容做一遍；
- 2、自己选择一些典型信号进行练习；
- 3、熟悉 MATLAB 软件使用环境、启动及退出等；熟悉 MATLAB 软件的常用命令的

使用；

4、规范化地书写实验报告。

实验二 连续时间信号卷积及 MATLAB 实现

一、实验目的：熟悉使用 MATLAB 软件来分析连续时间信号的卷积积分运算并用图形可视化相关结果。

二、实验时数：3 学时

三、实验内容：

1. 卷积积分

卷积积分在信号与线形系统分析中具有非常重要的意义，是信号与系统分析的基本方法之一。

连续时间信号 $f_1(t)$ 和 $f_2(t)$ 的卷积积分（简称为卷积） $f(t)$ 定义为：

$$f(t) = f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(t) f_2(t - \tau) d\tau$$

由此可得到两个与卷积相关的重要结论，即是：

(1) $f(t) = f(t) * \delta(t)$ ，即连续信号可分解为一系列幅度由 $f(t)$ 决定的冲激信号 $\delta(t)$ 及其平移信号之和；

(2) 线形时不变连续系统，设其输入信号为 $f(t)$ ，单位响应为 $h(t)$ ，其零状态响应为 $y(t)$ ，则有： $y(t) = f(t) * h(t)$ 。

可见，连续信号卷积的计算对我们进行连续信号与系统的分析具有重要的意义。

用 MATLAB 实现连续信号 $f_1(t)$ 与 $f_2(t)$ 卷积的过程如下：

(1) 将连续信号 $f_1(t)$ 与 $f_2(t)$ 以时间间隔 Δ 进行取样，得到离散序列 $f_1(k\Delta)$ 和 $f_2(k\Delta)$ ；

(2) 构造与 $f_1(k\Delta)$ 和 $f_2(k\Delta)$ 相对应的时间向量 k_1 和 k_2 ；

(3) 调用 `conv()` 函数计算卷积积分 $f(t)$ 的近似向量 $f(n\Delta)$ ；

(4) 构造 $f(n\Delta)$ 对应的时间向量 k 。

下面即是利用 MATLAB 实现连续时间卷积的通用函数 `sconv()`，该程序在计算出卷积积分的数值近似的同时，还绘出 $f(t)$ 的时域波形图。需要注意的是，程序中是如何构造 $f(t)$ 的对应时间向量 k 的？另外，程序在绘制 $f(t)$ 波形图时采用的是 `plot` 命令而不是 `stem` 命令。

```
function [f,k]=sconv(f1,f2,k1,k2,p)
%计算连续信号卷积积分 f(t)=f1(t)*f2(t)
% f: 卷积积分 f(t)对应的非零样值向量
% k: f(t)的对应时间向量
% f1: f1(t)非零样值向量
% f2: f2(t)的非零样值向量
% k1: f1(t)的对应时间向量
% k2: f2(t)的对应时间向量
% p: 取样时间间隔
```

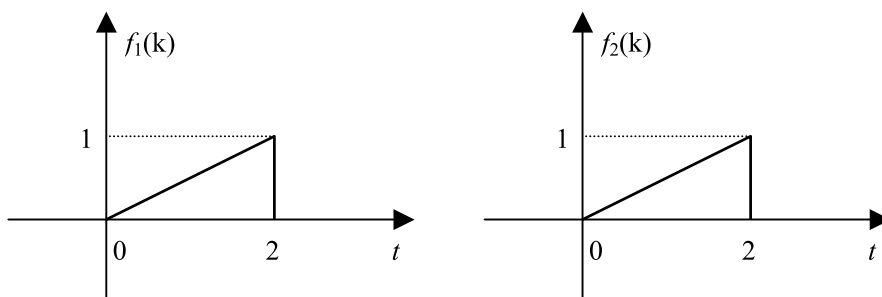
```

f=conv(f1,f2);                                %计算序列 f1 与 f2 的卷积和 f
f=f*p;                                         %计算序列 f 非零样值的起点位置
k0=k1(1)+k2(1);                               %计算卷积和 f 的非零样值的宽度
k3=length(f1)+length(f2)-2;                  %确定卷积和 f 非零样值的时间向量
k=k0:p:k3*p;
subplot(2,2,1)
plot(k1,f1)                                   %在子图 1 绘 f1(t)时域波形图
title('f1(t)')
xlabel('t')
ylabel('f1(t)')
subplot(2,2,2)
plot(k2,f2)                                   %在子图 2 绘 f2(t)时域波形图
title('f2(t)')
xlabel('t')
ylabel('f2(t)')
subplot(2,2,3)
plot(k,f);                                    %画卷积 f(t)的时域波形
h=get(gca,'position');
h(3)=2.5*h(3);                                %将第三个子图的横坐标范围扩为原来的 2.5 倍
set(gca,'position',h)
title('f(t)=f1(t)*f2(t)')
xlabel('t')
ylabel('f(t)')

```

四、学生实验部分：

1. 已知两连续时间信号如下图所示，试用 MATLAB 求 $f(t) = f_1(t) * f_2(t)$ ，并绘出 $f(t)$ 的时域波形图。（设定取样时间间隔为 p ）



◆ 参考程序：

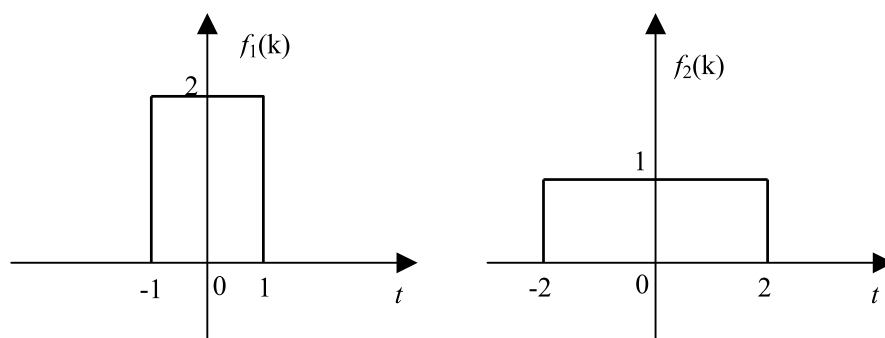
```

p=0.5;
k1=0:p:2;
f1=0.5*k1;
k2=k1;
f2=f1;
[f,k]=sconv(f1,f2,k1,k2,p)

```

【实验思考】：通过不断改变 p 的取值并对比所得到的实验效果，观察当取样时间 p 为多大时，函数 `sconv()` 的计算结果就是连续时间卷积 $f(t) = f_1(t) * f_2(t)$ 的较好近似结果？

2. 已知两连续时间信号如下图所示，试用 MATLAB 求 $f(t) = f_1(t) * f_2(t)$ ，并绘出 $f(t)$ 的时域波形图。（设定取样时间间隔为 p ）



◆参考程序：

```
p=0.1;
k1=-1:p:1
f1=2*ones(1,length(k1))
k2=-2:p:2
f2=ones(1,length(k2))
[f,k]=sconv(f1,f2,k1,k2,p)
```

【实验思考】：通过不断改变 p 的取值并对比所得到的实验效果，观察当取样时间 p 为多大时，函数 `sconv()` 的计算结果就是连续时间卷积 $f(t) = f_1(t) * f_2(t)$ 的较好近似结果？

实验三 系统时域特性的仿真分析实验

一、实验目的：通过使用 MATLAB 仿真软件对 LTI 系统的时域特性进行仿真分析，使学生对系统的冲激响应和零状态响应等有更深入的理解和掌握。

二、实验时数：3 学时+1 学时

三、连续系统的冲激响应、阶跃响应及 MATLAB 实现

对于 LTI 连续系统，求解系统的冲激响应 $h(t)$ 和阶跃响应 $g(t)$ 对我们进行连续系统的分析具有非常重要的意义。MATLAB 为用户提供了专门用于求连续系统冲激响应和阶跃响应并绘制其时域波形的函数 `impulse()` 和 `step()`。

在调用 `impulse()` 和 `step()` 函数时，我们需要用向量来对连续系统进行分析。

设描述连续系统的微分方程为：

$$\sum_{i=0}^N a_i y^{(i)}(t) = \sum_{j=0}^M b_j x^{(j)}(t)$$

则我们可用向量 **a** 和 **b** 来表示该系统，即：

$$\mathbf{a} = [a_N, a_{N-1}, \dots, a_1, a_0]$$

$$\mathbf{b} = [b_M, b_{M-1}, \dots, b_1, b_0]$$

注意，向量 **a** 和 **b** 的元素一定要以微分方程中时间求导的降幂次序来排列，且缺项要用 0 来补齐。例如，对微分方程 $y''(t) + 3y'(t) + 2y(t) = f''(t) + f(t)$ ，则表示该系统的对应向量应为 $\mathbf{a} = [1 \ 3 \ 2]$ ， $\mathbf{b} = [1 \ 0 \ 1]$ 。

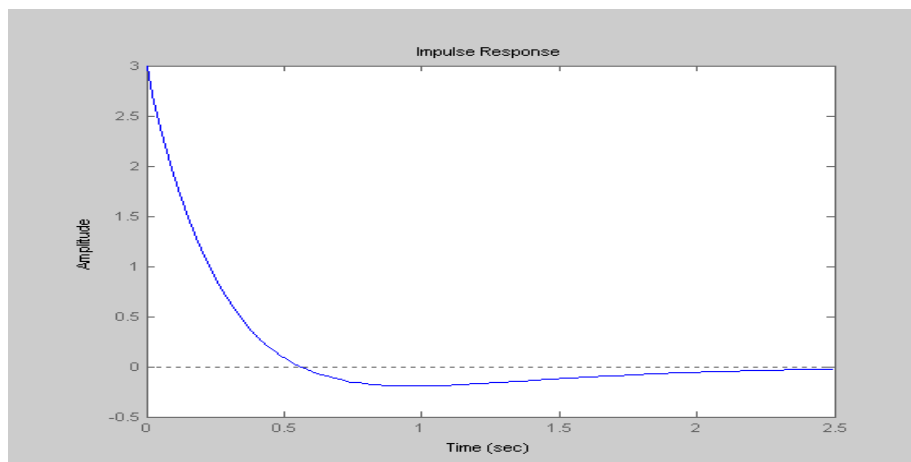
1. `impulse()` 函数

函数 `impulse()` 将绘出由向量 **a** 和 **b** 表示的连续系统在指定时间范围内的冲激响应 $h(t)$ 的时域波形图，并能求出指定时间范围内冲激响应的数值解。`impulse()` 函数有如下几种调用格式：

(1) `impulse(b,a)`：该调用格式以默认方式绘出向量 **a** 和 **b** 定义的连续系统的冲激响应的时域波形。例如描述连续系统的微分方程为 $y''(t) + 5y'(t) + 6y(t) = 3f'(t) + 2f(t)$ ，运行如下 MATLAB 命令：

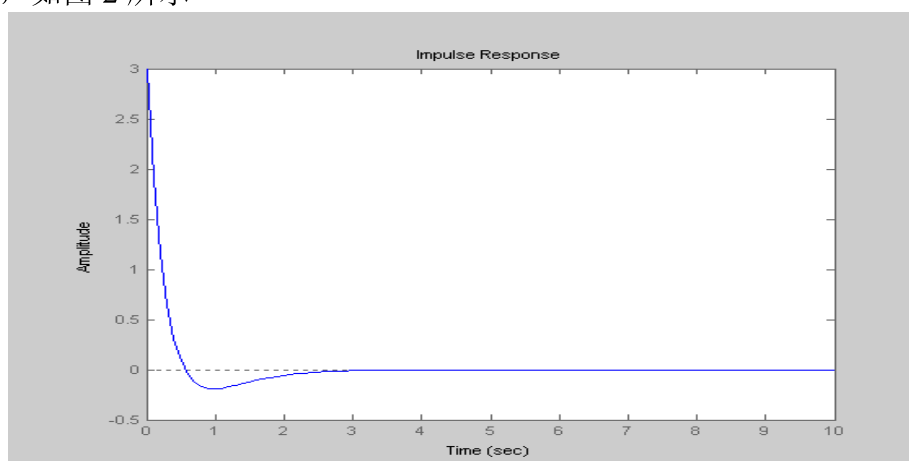
```
a=[1 5 6];  
b=[3 2];  
impulse(b,a);
```

则绘出系统的冲激响应波形，如图 1 所示。



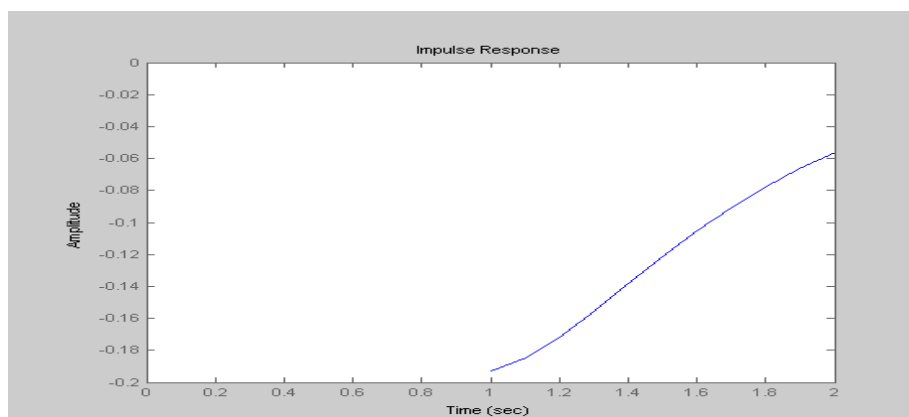
如图 1 连续系统的冲激响应 1

(2) `impulse(b,a,t)`: 绘出系统在 $0 \sim t$ 时间范围内冲激响应的时域波形。对上例，若运行命令 `impulse(b,a,10)`，则绘出系统在 $0 \sim 10$ 秒范围内冲激响应的时域波形，如图 2 所示



如图 2 连续系统的冲激响应 2

(3) `impulse(b,a,t1:p:t2)`: 绘出在 $t1 \sim t2$ 时间范围内，且以时间间隔 p 均匀取样的 激响应波形。对上例，若运行命令 `impulse(b,a,1:0.1:2)`，则绘出 $1 \sim 2$ 秒内，每隔 0.1 秒取样的冲激响应的时域波形，如图 3 所示



如图 3 连续系统的冲激响应 3

(4) `y=impulse(b,a,t1:p:t2)`: 不绘出波形, 而是求出系统冲激响应的数值解。
对上例, 若运行命令 `y=impulse(b,a,0:0.2:2)`, 则运行结果为:

```
y =  
    3.0000  
    1.1604  
    0.3110  
   -0.0477  
   -0.1726  
   -0.1928  
   -0.1716  
   -0.1383  
   -0.1054  
   -0.0777  
   -0.0559
```

2. `step()` 函数: 可绘出连续系统的阶跃响应 $g(t)$ 在指定时间范围的时域波形并能求出其数值解, 和 `impulse()` 函数一样, 也有四种调用格式。

3. 学生实验内容:

已知描述某连续系统的微分方程为:

$$2y''(t) + y'(t) + 8y(t) = f(t),$$

试用 MATLAB: (1) 绘出该系统在 0~30 秒范围内, 并以时间间隔 0.01 秒取样的冲激响应和阶跃响应的时域波形; (2) 求出系统在 0~30 秒范围内, 并以时间间隔 0.01 秒取样的冲激响应和阶跃响应的数值解。

四、离散系统的单位脉冲响应及 MATLAB 实现

1. 求解离散系统的单位脉冲响应 $h(k)$ 对我们进行离散时间系统的分析具有重要意义;

2. MATLAB 为用户提供专门求离散系统单位脉冲响应 $h(k)$ 的函数, 即 `impz()`。调用 `impz()` 函数时, 类似连续系统, 也需要用向量来对离散系统进行表示。

设描述离散系统的差分方程为:

$$\sum_{i=0}^N a_i y(k-i) = \sum_{j=0}^M b_j f(k-j)$$

则可以用向量 a 和 b 表示该系统, 即:

$$a = [a_0, a_1, \dots, a_{N-1}, a_N]$$

$$b = [b_0, b_1, \dots, b_{M-1}, b_M]$$

在用向量来表示差分方程描述的离散系统时, 缺项要用 0 来补齐。

函数 `impz()` 能绘出向量 a 和 b 定义的离散系统在指定时间范围内单位脉冲响应 $h(k)$ 的时域波形, 并能求出系统单位脉冲响应 $h(k)$ 在指定时间范围内的

数值解。

impz()函数有如下几种调用格式：

- impz(b,a) 以默认方式绘出向量 a 和 b 定义的离散系统 h(k)的时域波形；
- impz(b,a,n) 绘出向量 a 和 b 定义的离散系统在 0~n (n 必须为整数) 离散时间范围内单位脉冲响应 h(k)的时域波形；
- impz(b,a,n1:n2) 绘出向量 a 和 b 定义的离散系统在 n1~n2 (n1、n2 必须为整数，且 n1<n2) 离散时间范围内单位脉冲响应 h(k)的时域波形；
- y=impz(b,a,n1:n2) 不绘出系统的 h(k)的时域波形，而是求出向量 a 和 b 定义的离散系统在 n1~n2 (n1、n2 必须为整数，且 n1<n2) 离散时间范围内单位脉冲响应 h(k)的数值解。

3. 学生实验内容：

已知某离散系统的差分方程为 $y(k)-y(k-1)+0.9y(k-3)=f(k)$ ，试作出：(1) 以默认方式绘出系统 h(k)的时域波形；(2) 绘出系统在 0~60 取样点范围内 h(k)的时域波形；(3) 绘出系统在 -10~40 离散时间范围内 h(k)的时域波形；(4) 求出系统在 -5~10 离散时间范围内 h(k)的数值解。

五、利用 MATLAB 中的 Simulink 进行系统时域特性仿真

(一) 仿真系统的冲激响应

已知或求解出系统传输函数 H(s)，分子和分母都写出多项式的形式，例如：

$$H(s) = \frac{s+1}{s^2+1.3s+0.8}$$

在 continuous 的子库中选择传输函数 (Transfer Fcn) 子库，用鼠标把传输函数模块拖动 untitled 视窗中，置于激励信号源和示波器之间，然后用鼠标拖出的连线将信号源、传输函数模块和示波器等按照系统的要求连接起来即可。

如果需要对传输函数的参数进行设置，则可通过双击传输函数模块打开它的设置环境窗，可以分别设置分子、分母多项式的系数和阶数等。

在 untitled 窗口的菜单中选 simulation 的 start 功能则可执行仿真，待仿真完毕后，则可通过激活示波器观察到激励信号和响应信号的波形。

由于 source 库中没有单位冲激信号模块，所以由阶跃信号模块经过微分来产生。

(二) 仿真系统在任意激励下的零状态响应

采用的方法类似 (一) 中仿真系统的冲激响应，只是输入是任意波形例如周期的矩形波等。

(三) 学生实验内容：

对如下连续时间系统

$$H(s) = \frac{s+1}{s^2+1.3s+0.8}$$

通过仿真分别观察其单位冲激响应波形和在周期矩形信号作用下的零状态响应波形。

实验四 连续时间信号的频域特性仿真实验

一、用 MATLAB 函数实现连续信号的频域分析

MATLAB 的 Symbolic Math Toolbox 提供了能直接求解傅立叶变换及其逆变换的函数 `fourier()` 和 `ifourier()`。

(1) Fourier 变换函数 `fourier()` 调用格式

- $F = \text{fourier}(f)$ 是符号函数 f 的 Fourier 变换，默认返回是关于 ω 的函数。如果 $f = f(\omega)$ ，则 `fourier` 函数返回关于 t 的函数。
- $F = \text{fourier}(f, v)$ 返回函数 F 是关于符号对象 v 的函数，而不是默认的 ω 。

$$F(v) = \int_{-\infty}^{\infty} f(x) e^{-jvx} dx$$

- $F = \text{fourier}(f, u, v)$ 对关于 u 的函数 f 进行变换，返回函数 F 是关于 v 的函数。

$$F(v) = \int_{-\infty}^{\infty} f(u) e^{-jvu} du$$

(2) Fourier 逆变换 `ifourier()` 调用格式

- $f = \text{ifourier}(F)$ 是函数 F 的 Fourier 逆变换，默认的独立变量是 ω ，默认返回是关于 x 的函数。如果 $f = f(\omega)$ ，则 `fourier` 函数返回关于 t 的函数。
- $f = \text{ifourier}(F, u)$ 返回函数 f 是 u 的函数，而不是默认的 x 的函数。
- $f = \text{ifourier}(F, v, u)$ 对关于 v 的函数 F 进行变换，返回关于 u 的函数。

注意：• 在调用函数 `fourier()` 和 `ifourier()` 之前，需用 `syms` 命令对所用到的变量进行说明，即要将这些变量说明成符号变量。

- 采用 `fourier()` 和 `ifourier()` 得到的返回函数，仍然是符号表达式。如需对返回的函数作图，则应用 `ezplot()` 绘图命令而不是用 `plot()` 命令。如果返回函数中含有诸如狄拉克函数 $\delta(t)$ 等的项，则用 `ezplot()` 也无法作图。

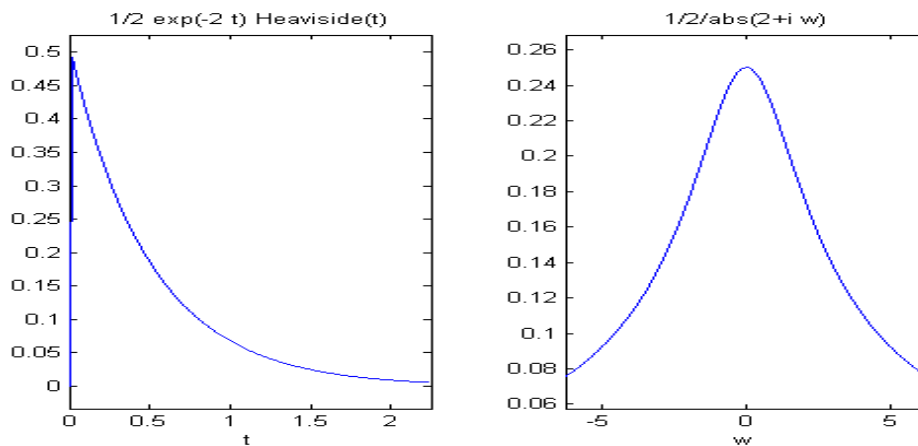
(3) 应用举例：

例1 试绘出连续时间信号 $f(t) = \frac{1}{2} e^{-2t} \varepsilon(t)$ 的时域波形 $f(t)$ 及相应的幅频特性图。参考 MATLAB 语言代码如下：

```
clear;
syms t;
x=1/2*exp(-2*t)*sym('Heaviside(t)');
subplot(1,2,1);
ezplot(x);
F=fourier(x);
```

```
Subplot(1,2,2);
ezplot(abs(F));
```

程序绘制的信号时域波形及幅频特性图为：



注：Heaviside(t)函数即为单位阶跃函数 $\varepsilon(t)$ 。在调用 Heaviside(t)函数之前一定要在你的当前工作目录下创建该函数。创建 Heaviside(t)函数方法如下：

```
function f=Heaviside(t)
f=(t>0);
```

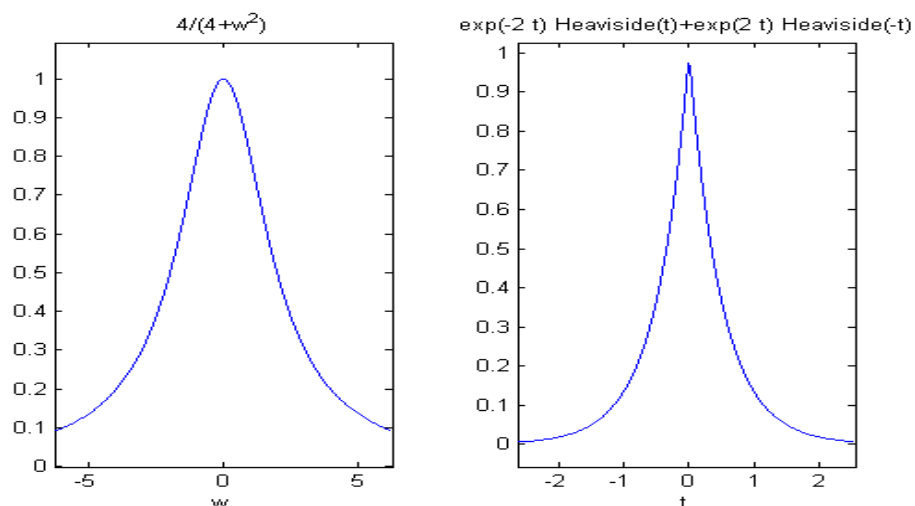
且以 Heaviside.m 文件名保存。

例2 若某信号 $f(t)$ 的傅立叶变换， $F(j\omega)$ 试画出该信号的时域波形和响应频谱图。

参考 MATLAB 语言代码如下：

```
clear;
syms t w;
F=4/(4+w^2);
subplot(1,2,1);
ezplot(F);
f=ifourier(F,t);
subplot(1,2,2);
ezplot(f);
```

程序绘制的信号时域波形及响应频谱图为：



- 作业（自己发挥部分）：已知某一连续时间信号为 $f(t) = e^{-2|t|}$ ，试绘出它的时域波形及相应的频谱图。

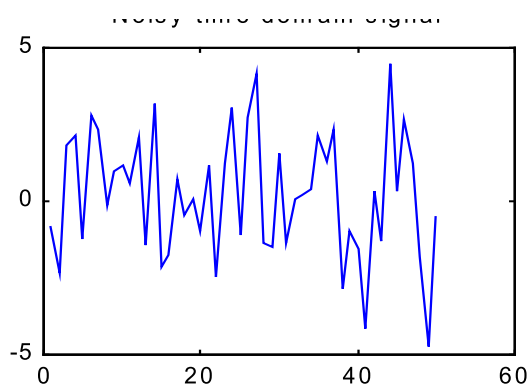
二、用 fft 函数对信号进行频谱分析

例如，要求产生一个时间从 0 到 250ms 的，含有噪声的，频率为 50Hz 和 120Hz 的时域信号，噪声的标准差为 2。离散信号的时间间隔为 2ms。

则 MATLAB 程序如下：

```
t=0:0.001:0.25;  
x=sin(2*pi*50*t)+sin(2*pi*120*t);  
y=x+2*randn(size(t));  
plot(y(1:50));  
title('Noisy time domain signal');
```

运行结果为

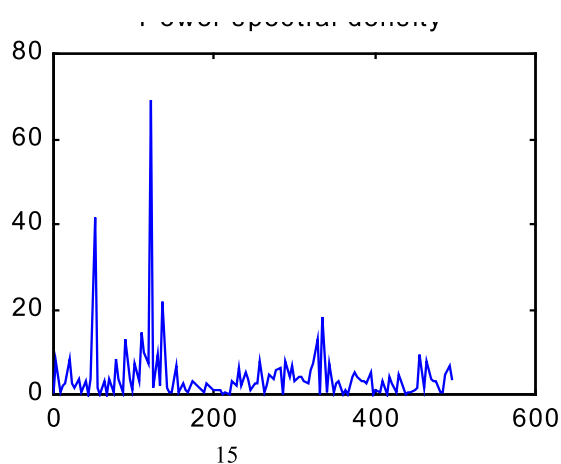


在绘制出的时域波形中我们很难辨认出信号中所含的频率成分，因此有必要进一步做频谱分析。

MATLAB 程序如下：

```
Y=fft(y,256);  
pyy=Y.*conj(Y)/256;  
f=1000/256*(0:127);  
plot(f,pyy(1:128));  
title('power spectral density');
```

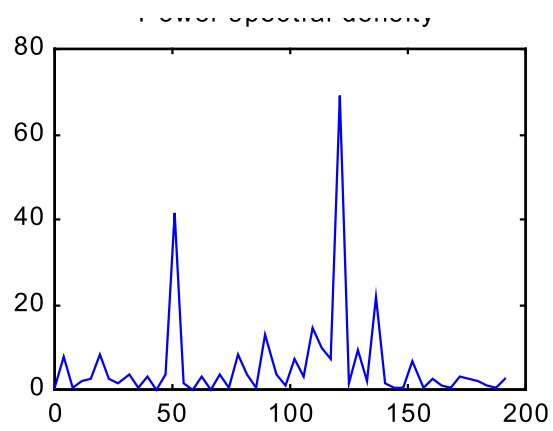
运行结果为，



为了改变显示区域，可使用如下语句，

```
plot(f(1:50),pyy(1:50));
```

运行结果为，



以上 MATLAB 程序请同学们跟着执行一遍，以便加深理解和掌握。

实验五 信号的幅度调制及 MATLAB 实现

一、实验目的：熟悉使用 MATLAB 软件来分析信号的调制问题并可视化相关结果，同时分析和对比有关结果。

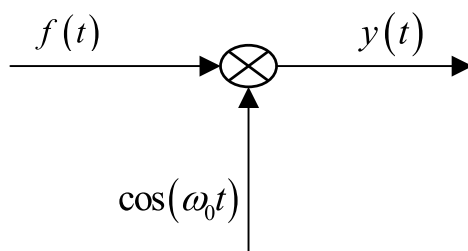
二、实验时数：3 学时

三、实验内容：

设信号 $f(t)$ 的频谱为 $F(j\omega)$ ，现将 $f(t)$ 乘以载波信号 $\cos(\omega_0 t)$ ，得到高频的已调信号 $y(t)$ ，即：

$$y(t) = f(t) \cos(\omega_0 t)$$

其中， $f(t)$ 称为调制信号。实现信号调制的原理图如图所示



图示 幅度调制原理图

从频域上看，已调制信号 $y(t)$ 的频谱为原调制信号 $f(t)$ 的频谱搬移到 $\pm\omega_0$ 处，幅度降为原 $F(j\omega)$ 的 $1/2$ ，即：

$$f(t) \cos(\omega_0 t) \leftrightarrow \frac{1}{2} \{ F[j(\omega + \omega_0)] + F[j(\omega - \omega_0)] \}$$

上式即为调制原理，也是傅立叶变换性质中“频移特性”的一种特别情形。这里采用的调制方法为抑制载波方式，即 $y(t)$ 的频谱中不含有 $\cos(\omega_0 t)$ 的频率分量。

MATLAB 提供了专门的函数 `modulate()` 用于实现信号的调制。调用格式为：

```
y=modulate(x,Fc,Fs,'method')
```

```
[y,t]=modulate(x,Fc,Fs)
```

其中， x 为被调信号， F_c 为载波频率， F_s 为信号 x 的采样频率，`method` 为所采用的调制方式，若采用幅度调制、双边带调制、抑制载波调制，则 `'method'` 为 `'am'` 或 `'amdsd-sc'`。其执行算法为：

$$y = x * \cos(2 * \pi * F_c * t)$$

其中， y 为已调信号， t 为函数计算时间间隔向量。

在 MATLAB 的实现程序中，为了观察 $f(t)$ 及 $y(t)$ 的频谱，可使用“信号处理工具箱函数”中估计信号的功率谱密度函数 `psd()`，其格式是：

`[Px,f]=psd(x,Nfft,Fs>window,noverlap,dflag)`

其中， x 是被调信号（即本例中的 $f(t)$ ），`Nfft` 指定快速付式变换 FFT 的长度， F_s 为对信号的采样频率。后面三个参数的意义将涉及信号处理的更深的知识，在此暂不介绍。

四、学生实验部分：

1. 设信号 $f(t) = \sin(100\pi t)$ ，载波 $y(t)$ 为频率为 400Hz 的余弦信号。试用 MATLAB 实现调幅信号 $y(t)$ ，并观察 $f(t)$ 的频谱和 $y(t)$ 的频谱，以及两者在频域上的关系。

用 MATLAB 实现本例的参考程序如下：

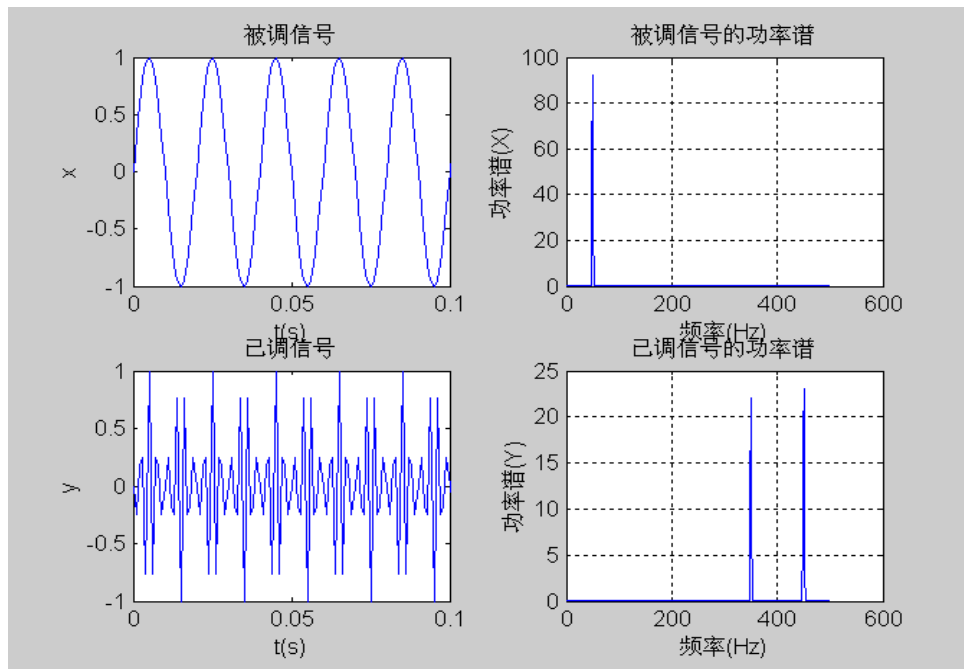
```
Fs=1000;
Fc=400;
N=1000;
n=0:N-2;
t=n/Fs;
x=sin(2*pi*50*t);
subplot(221)
plot(t,x);
xlabel('t(s)');
ylabel('x');
title('被调信号');
axis([0 0.1 -1 1])
Nfft=1024;
window=hamming(512);
noverlap=256;
dflag='none';
[Pxx,f]=psd(x,Nfft,Fs>window,noverlap,dflag);
subplot(222)
plot(f,Pxx)
xlabel('频率(Hz)');
ylabel('功率谱(X)');
title('被调信号的功率谱')
grid
y=modulate(x,Fc,Fs,'am');
subplot(223)
plot(t,y)
xlabel('t(s)');
ylabel('y');
axis([0 0.1 -1 1])
title('已调信号')
```

```

[Pxx,f]=psd(y,1024,Fs>window,noverlap,dflag);
subplot(224)
plot(f,Pxx)
xlabel('频率(Hz)');
ylabel('功率谱(Y)');
title('已调信号的功率谱');
grid

```

程序运行结果为



2. 设 $f(t) = u(t+1) - u(t-1)$ ， $f_1(t) = f(t)\cos(10\pi t)$ ，试用 MATLAB 画出 $f(t)$ 、 $f_1(t)$ 的时域波形及其频谱，并观察傅立叶变换的频移特性。

用 MATLAB 实现本例的参考程序如下：

```

R=0.005;t=-1.2:R:1.2;
f=Heaviside(t+1)-Heaviside(t-1);
f1=f.*cos(10*pi*t);
subplot(221)
plot(t,f)
xlabel('t');
ylabel('f(t)');
subplot(222);
plot(t,f1);
xlabel('t');
ylabel('f1(t)=f(t)*cos(10*pi*t)');
W1=40;
N=1000;
k=-N:N;
W=k*W1/N;

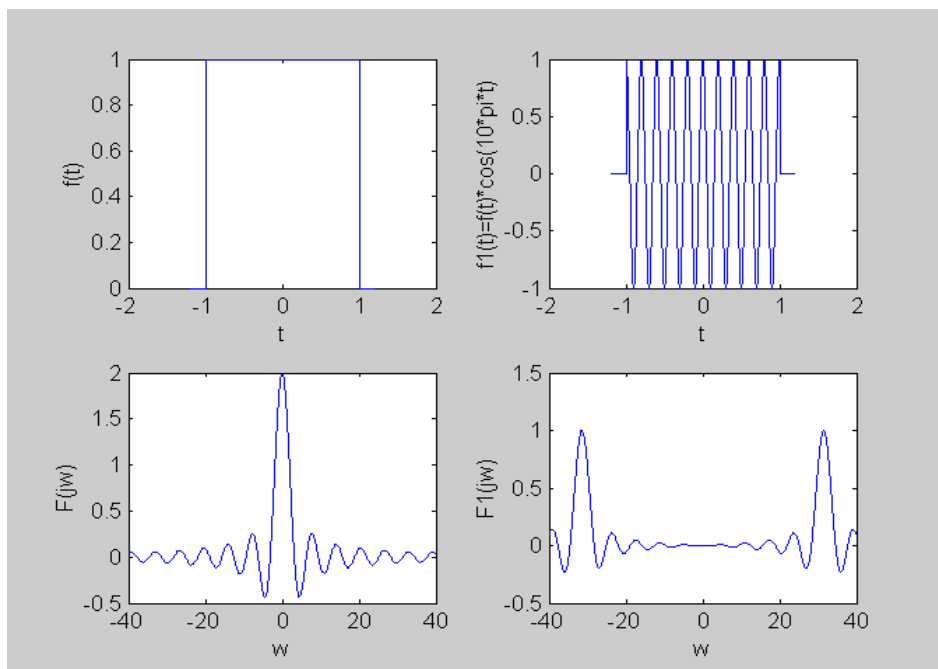
```

```

F=f*exp(-j*t*W)*R;
F=real(F);
F1=f1*exp(-j*t*W)*R;
F1=real(F1);
subplot(223);
plot(W,F);
xlabel('w');
ylabel('F(jw)');
subplot(224);
plot(W,F1);
xlabel('w');
ylabel('F1(jw)');

```

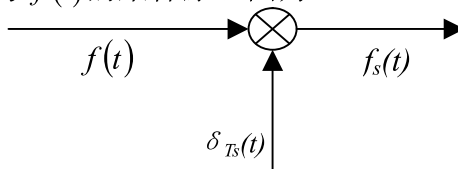
程序运行结果为



实验六 连续信号的采样与恢复（重构）

一、连续信号的采样

对某一连续时间信号 $f(t)$ 的采样原理图



信号采样原理图

由图可知， $f_s(t) = f(t) \cdot \delta_{Ts}(t)$ ，其中，单位冲激采样信号 $\delta_{Ts}(t)$ 的表达式为：

$$\delta_{Ts}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nTs)$$

其傅立叶变换为 $\Omega_s \sum_{n=-\infty}^{\infty} \delta(\Omega - n\Omega_s)$ ，其中 $\Omega_s = 2\pi/T$ ，设 $F(j\Omega)$ 为 $f(t)$ 的傅立叶变换， $f_s(t)$ 的频谱为 $F_s(j\Omega)$ ，由傅立叶变换的频域卷积定理，有：

$$f_s(t) = f(t) \cdot \delta_{Ts}(t) \leftrightarrow F_s(j\Omega) = \frac{1}{2\pi} F(j\Omega) * \Omega_s \sum_{n=-\infty}^{\infty} \delta(\Omega - n\Omega_s) = \frac{1}{Ts} \sum_{n=-\infty}^{\infty} F[j(\Omega - n\Omega_s)]$$

若设 $f(t)$ 是带限信号，带宽为 Ω_m ，即当 $|\Omega| > \Omega_m$ 时， $f(t)$ 的频谱 $F(j\Omega) = 0$ ，则 $f(t)$ 经过采样后的频谱 $F_s(j\Omega)$ 就是 $F(j\Omega)$ 在频率轴上搬移至 $0, \pm\Omega_s, \pm2\Omega_s, \dots, \pm n\Omega_s, \dots$ 处（幅度为原频谱的 $1/T_s$ 倍）。因此，当 $\Omega_s \geq 2\Omega_m$ 时，频谱不会发生混叠；而当 $\Omega_s \leq 2\Omega_m$ 时，频谱发生混叠。

二、信号的恢复

设信号 $f(t)$ 被采样后形成的采样信号为 $f_s(t)$ ，信号的重构是指由 $f_s(t)$ 经内插处理后，恢复出原来的信号 $f(t)$ 的过程，因此又称为信号恢复。设 $f(t)$ 为带限信号，带宽为 Ω_m ，经采样后的频谱为 $F_s(j\Omega)$ 。设采样频率 $\Omega_s \geq 2\Omega_m$ ，则 $F_s(j\Omega)$ 是以 Ω_s 为周期的谱线。现取一个频率特性为

$$H(j\Omega) = \begin{cases} Ts, & |\Omega| < \Omega_c \\ 0, & |\Omega| > \Omega_c \end{cases} \quad (\text{其中，截止频率 } \Omega_c \text{ 满足 } \Omega_m \leq \Omega_c \leq \Omega_s/2)$$

的理想低通滤波器与 $F_s(j\Omega)$ 相乘，得到的频谱即为原信号的频谱 $F(j\Omega)$ 。

根据时域卷积定理，有：

$$f(t) = h(t) * f_s(t)$$

其中， $f_s(t) = f(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nTs) = \sum_{n=-\infty}^{\infty} f(nTs) \cdot \delta(t - nTs)$

$$h(t) = F^{-1}[H(j\Omega)] = Ts \frac{\Omega_c}{\pi} \text{Sa}(\Omega_c t),$$

Ω_c 为 $H(j\Omega)$ 的截止角频率。

因此，得到：

上式即为用 $f(nTs)$ 表达 $f(t)$ 的表达式，其中的抽样函数 $Sa(\Omega_c t)$ 为内插函数。

三、应用举例

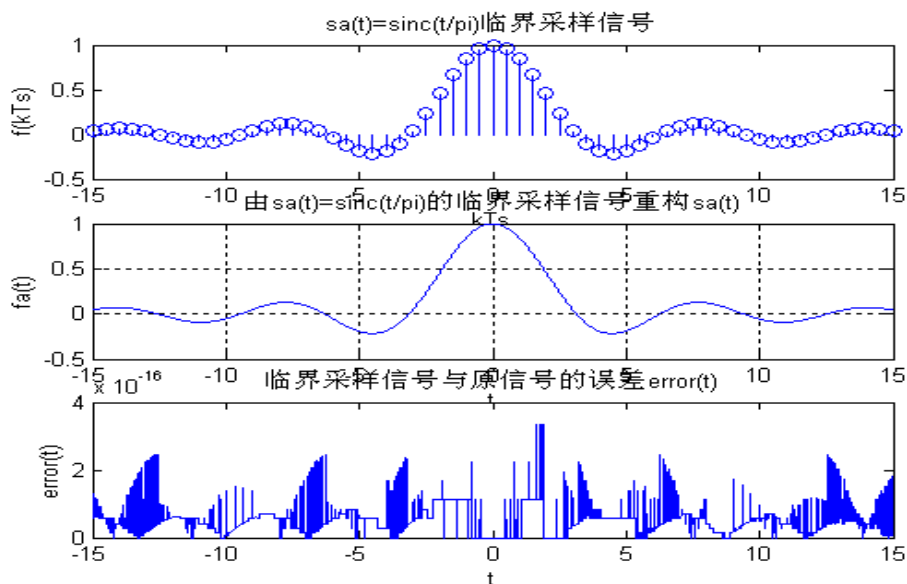
下面我们选取信号 $f(t)=Sa(t)=\sin(t)/t$ 作为被采样的信号，其

$$F(j\Omega)=\begin{cases} \pi, & |\Omega| < 1 \\ 0, & |\Omega| > 1 \end{cases}$$

即信号的带宽 $\Omega_m=1$ 。当采样频率 $\Omega_s=2\Omega_m$ 时，被称为临界采样（取 $\Omega_c=\Omega_m$ ）。在临界采样状态下实现对信号 $Sa(t)$ 的采样及由该采样信号恢复 $Sa(t)$ 的参考程序如下：

```
clear;
wm=1;           %信号带宽
wc=wm;          %滤波器截止频率
Ts=pi/wm;       %采样间隔
ws=2*pi/Ts;     %采样角频率
n=-100:100;     %时域采样点数
nTs=n*Ts;       %时域采样点
f=sinc(nTs/pi);
Dt=0.005;
t=-15:Dt:15;
fa=f*Ts*wc/pi*sinc((wc/pi)*(ones(length(nTs),1)*t-nTs'*ones(1,length(t)))); %信号
                                重构
error=abs(fa-sinc(t/pi));      %求重构信号与原信号的误差
t1=-15:0.5:15;
f1=sinc(t1/pi);
subplot(3,1,1);
stem(t1,f1);
xlabel('kTs');
ylabel('f(kTs)');
title('sa(t)=sinc(t/pi)临界采样信号');
subplot(3,1,2);
plot(t,fa);
xlabel('t');
ylabel('fa(t)');
title('由 sa(t)=sinc(t/pi)的临界采样信号重构 sa(t)');
grid;
subplot(3,1,3);
plot(t,error);
xlabel('t');
ylabel('error(t)');
title('临界采样信号与原信号的误差 error(t)');
```

MATLAB 程序代码运行结果如下：



如图 临界采样、重构信号及两信号的绝对误差

四、实验作业

设信号 $f(t) = Sa(t) = \sin(t)/t$, 在取样间隔分别为 $T_s = 0.7\pi$ (令 $\Omega_m = 1$, $\Omega_c = 1.1\Omega_m$) 和 $T_s = 1.5\pi$ (令 $\Omega_m = 1$, $\Omega_c = \Omega_m$) 的两种情况下, 对信号 $f(t)$ 进行采样, 试编写 MATLAB 程序代码, 并绘制出采样信号波形、由采样信号得到的重构信号波形以及两信号的绝对误差波形。

实验七 用 MATLAB 分析拉普拉斯变换及其曲面

一、实验目的：熟悉使用 MATLAB 软件来分析拉普拉斯变换，并用三维的曲面图来可视化拉氏变换结果。

二、实验时数：3 学时

三、实验内容：

拉普拉斯变换是分析连续时间信号的有效手段，对于当 $t \rightarrow \infty$ 时信号幅度不衰减或增长的时间信号，其傅立叶变换不存在，但我们可以用拉普拉斯变换来分析它们。

连续时间信号 $f(t)$ 的拉普拉斯变换定义为：

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-st} dt$$

其中 $s = \delta + j\omega$ ，若以 δ 为横坐标（实轴）， $j\omega$ 为纵坐标（虚轴），复变量 s 就构成了一个复平面，称为 s 平面。

显然， $F(s)$ 是复变量 s 的复函数，为了便于理解和分析 随 s 的变化规律，我们可以将 $F(s)$ 写成：

$$F(s) = |F(s)|e^{j\varphi(s)}$$

其中 $|F(s)|$ 为复信号 $F(s)$ 的模，而 $\varphi(s)$ 为 $F(s)$ 的相角。

1. 用 MATLAB 绘制拉普拉斯变换的曲面图

从三维几何空间的角度来看， $|F(s)|$ 和 $\varphi(s)$ 对应着复平面的两个曲面，如果我们能绘出它们的三维曲面图，我们就可以直观地分析连续信号的拉普拉斯变换 $F(s)$ 随复变量 s 的变化。

上述过程我们可以利用 MATLAB 的三维绘图功能来实现。现在考虑如何用 MATLAB 来绘制 s 平面的有限区域上连续时间信号 $f(t)$ 的拉普拉斯变换 $F(s)$ 的曲面图，我们以单位阶跃信号 $u(t)$ 为例来说明实现过程。

我们知道，对单位阶跃信号 $f(t) = u(t)$ ，其拉普拉斯变换为 $F(s) = \frac{1}{s}$ 。

首先，我们用两个向量来确定绘制曲面图的 s 平面的横、纵坐标的范围。例如，我们可定义绘制曲面图的横坐标范围向量 $x1$ 和纵坐标范围向量 $y1$ 分别为：

$$x1 = -0.2:0.03:0.2;$$

$$y1 = -0.2:0.03:0.2;$$

然后再调用 meshgrid() 函数来产生矩阵 S ，并用该矩阵来表示绘制曲面图的复平面区域，对应的 MATLAB 命令如下：

$$[x,y] = \text{meshgrid}(x1,y1);$$

$$s = x + i*y;$$

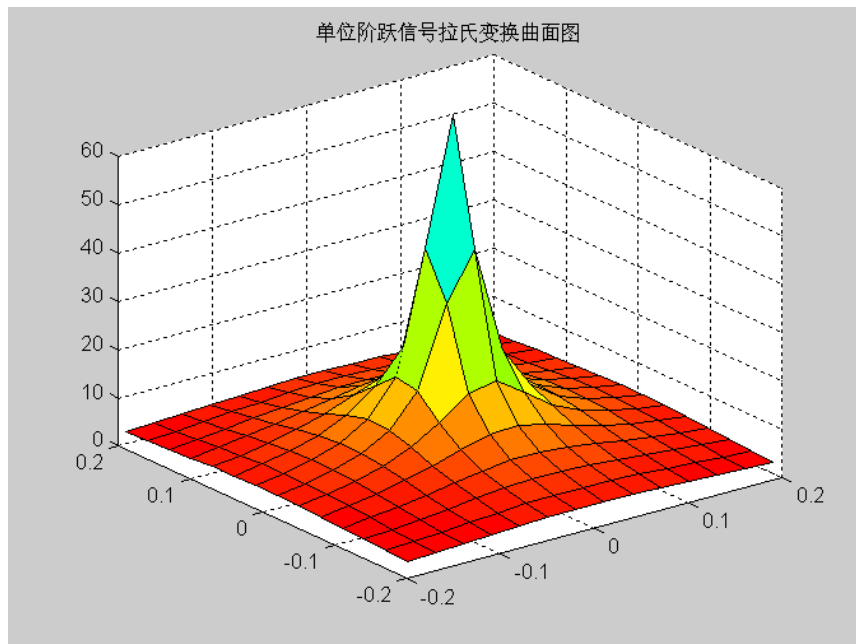
上述命令产生的矩阵 S 包含了复平面 $-0.2 < \delta < 0.2$ 、 $-0.2 < j\omega < 0.2$ 范围内以间

隔 0.03 取样的所有样点。

最后我们再计算出信号拉普拉斯变换在复平面的这些样点上的值，即可用函数 `mesh()` 来绘出其曲面图，对应命令为：

```
fs=abs(1./s);          %计算拉氏变换在复平面上的样点值
mesh(x,y,fs);          %绘制的氏变换曲面图
surf(x,y,fs);
title('单位阶跃信号拉氏变换曲面图');
colormap(hsv);
axis([-0.2,0.2,-0.2,0.2,0,60]);
rotate3d;
```

执行上述命令后，绘制的单位阶跃信号拉普拉斯变换曲面图为



阶跃信号拉氏变换曲面图

【练习内容】:

已知连续时间信号 $f(t) = \sin(t)u(t)$ ，求出该信号的拉普拉斯变换，并用 MATLAB 绘制拉普拉斯变换的曲面图。

实现此功能的 MATLAB 参考程序为：

```
%绘制单边正弦信号拉普拉斯变换曲面图程序
clf;
a=-0.5:0.08:0.5;
b=-1.99:0.08:1.99;
[a,b]=meshgrid(a,b);
d=ones(size(a));
c=a+i*b;                                %确定绘制曲面图的复平面区域
c=c.*c;
c=c+d;
c=1./c;
c=abs(c);                                %计算拉普拉斯变换的样值
mesh(a,b,c);                             %绘制曲面图
```



```
surf(a,b,c);
axis([-0.5,0.5,-2,2,0,15]);
title('单边正弦信号拉氏变换曲面图');
colormap(hsv);
```

2. 由拉普拉斯曲面图观察频域与复频域的关系

我们知道，若信号 $f(t)$ 的傅立叶变换存在，则其拉普拉斯变换 $F(s)$ 与傅立叶变换 $F(j\omega)$ 存在如下关系：

$$F(j\omega) = F(s) \Big|_{s=j\omega}$$

也即在信号拉普拉斯变换 $F(s)$ 中令 $\delta = 0$ ，就可得到信号的傅立叶变换。从三维几何空间的角度来看，信号 $f(t)$ 的傅立叶变换 $F(j\omega)$ 就是其拉普拉斯曲面图中虚轴（ $\delta = 0$ ）所对应的曲线。我们可以通过将 $F(s)$ 曲面图在虚轴上进行剖面来直观地观察信号的拉普拉斯变换与傅立叶变换的关系。

例如，试利用 MATLAB 绘制信号 $f(t) = u(t) - u(t-2)$ 的拉普拉斯变换的曲面图，观察曲面图在虚轴剖面上的曲线，并将其与信号傅立叶变换 $F(j\omega)$ 绘制的振幅频谱进行比较。

根据拉普拉斯变换和傅立叶变换的定义和性质，我们可以求得该信号的拉普拉斯变换和傅立叶变换如下：

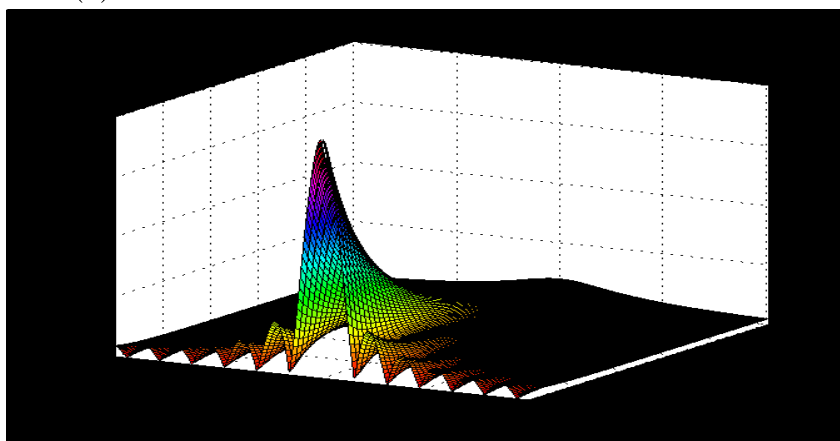
$$F(s) = \frac{1 - e^{-2s}}{s}$$

$$F(j\omega) = 2Sa(\omega)e^{-j\omega}$$

我们可绘制出该信号的拉普拉斯变换曲面图，为了更好地观察曲面图在虚轴剖面上的曲线，我们定义绘制曲面图的 s 平面实轴范围从 0 开始，并用 `view` 函数来调整观察视角。实现上述过程的 MATLAB 程序如下：

```
%绘制矩形信号拉普拉斯变换曲面图程序
clf;
a=-0:0.1:5;
b=-20:0.1:20;
[a,b]=meshgrid(a,b);
c=a+i*b;                                %确定绘图区域
c=(1-exp(-2*c))./c;
c=abs(c);                                %计算拉普拉斯变换
mesh(a,b,c);                             %绘制曲面图
surf(a,b,c);
view(-60,20)                             %调整观察视角
axis([-0.5,5,-20,20,0,2]);
title('拉普拉斯变换（S 域像函数）');
colormap(hsv);
```

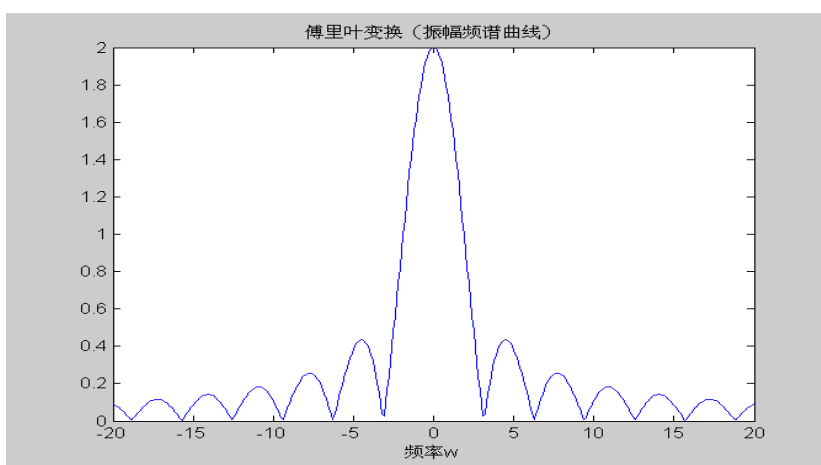
由上述程序绘制的拉普拉斯变换的曲面图如图 1，从该曲面图中我们可以明显地观察到 $F(s)$ 在虚轴剖面上的曲线的变换情况。



现在我们用 MATLAB 来绘制该信号的傅立叶变换曲线（振幅频谱），对应的 MATLAB 程序如下：

```
%绘制矩形时间信号傅里叶变换曲线程序
w=-20:0.1:20; %确定频率范围
Fw=(2*sin(w).*exp(i*w))./w; %计算傅里叶变换
plot(w,abs(Fw)) %绘制信号振幅频谱曲线
title('傅里叶变换（振幅频谱曲线）')
xlabel('频率 w')
```

上述程序绘制的信号傅立叶变换曲线图如下图，



通过对以上信号的拉普拉斯曲面图和傅立叶曲线图进行对比，可以直观地观察到两种变换之间的关系。

3. 拉普拉斯变换零极点分布对曲面图的影响

从单位阶跃信号和单边正弦信号的拉普拉斯变换曲面图中我们可以看出，曲面图中均有突出的尖峰，仔细观察便可得出，这些峰值点在 s 平面的对应点就是信号拉普拉斯变换的极点位置。因此，如果将拉普拉斯变换曲面图比喻为一个地貌图的话，则极点位置就对应着山峰的峰点。

我们再来看一下拉普拉斯变换零点对曲面图的影响，考虑如下信号：

$$F(s) = \frac{2(s-3)(s+3)}{(s-5)(s^2+10)}$$

该信号的零点为 $q_{1,2} = \pm 3$ ，极点为 $p_{1,2} = \pm j3.1623$ ， $p_3 = 5$

我们用前面介绍的方法将其拉普拉斯变换的曲面图绘制出来，对应的 MATLAB 程序如下：

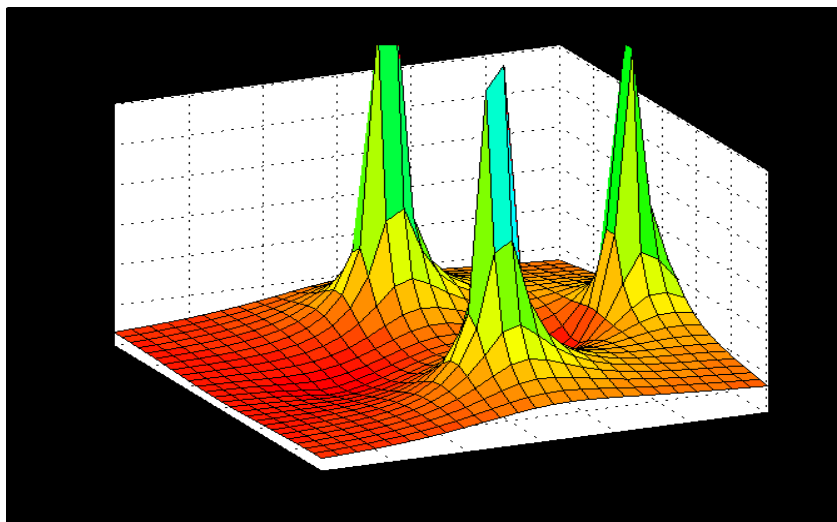
```
%观察拉普拉斯变换零极点对曲面图影响程序
clf;
a=-6:0.48:6;
b=-6:0.48:6;
[a,b]=meshgrid(a,b);
c=a+i*b;
```

```

d=2*(c-3).*(c+3);
e=(c.*c+10).*(c-5);
c=d./e;
c=abs(c);
mesh(a,b,c);
surf(a,b,c);
axis([-6,6,-6,6,0,3]);
title('拉普拉斯变换曲面图');
colormap(hsv);
view(-25,30)

```

由此程序绘制出来的拉氏变换零极点分布曲面图为



从曲面图中我们可以明显看出，曲面图在 $s = \pm j3.1623$ ， $s = 5$ 处有三个峰点，对应着拉普拉斯变换的极点位置，而在 $s = \pm 3$ 处有两个谷点，对应着拉普拉斯变换的零点位置。因此，信号拉普拉斯变换的零极点位置，决定了其曲面图的峰点和谷点位置。