东南大学 SOUTHEAST UNIVERSITY

4 频域分析与滤波器设计

讲 解 人： 王俊波

E-mail: jbwang@seu.edu.cn

Phone: 13770681926

QQ:308322767

主要内容

- 频域分析
  - ✓ 傅里叶分析
  - ✓ 离散时间傅里叶变换和离散傅里叶变换
  - ✓ 快速傅里叶变换分析
  - ✓ 离散时间序列和连续时间信号
  - ✓ 栅栏效应和泄露现象
- 离散时间系统简介
- 滤波器设计
  - ✓ 基本知识
  - ✓ 窗函数法
  - ✓ 频率采样法
  - ✓ 最优等波纹法

信号的频域分析概述

- 分析信号的不同角度称为域，通常包括时域和频域
- 时域即时间域，自变量是时间，即横轴是时间，纵轴是信号的变化
  - ✓ 时域分析是以时间轴为坐标表示动态信号的关系
  - ✓ 时域的表示较为形象与直观
  - ✓ 时域表示的信号通过线性时不变（LTI）系统时，时域卷积积分可给出LTI系统的输出
- 频域即频率域，自变量是频率，即横轴是频率，纵轴是该频率信号的幅度或相位，也就是通常说的幅度谱（频谱图）和相位谱
  - ✓ 信号频谱：信号的频域表示通常描述信号的功率或能量如何分配到不同的频率分量上，这种在频率上的分布称为信号频谱
  - ✓ 幅度谱更为常见
  - ✓ 周期性信号的频谱是离散的，因为其功率集中在基频的倍频上，与信号的基频直接相关
  - ✓ 非周期信号的频谱是频率的连续函数

信号频谱分析

- 频域分析的特性
  - ✓ 当LTI系统的输入信号是某个频率的复指数（或余弦和正弦的组合）时，输入信号乘以一个复常数就可以得到系统的输出信号，该复常数与系统对输入频谱的响应有关
  - ✓ 例：如果$x(t)=e^{j2\pi f_0 t}$ $(-\infty<t<\infty)$对于某一频率f_0，是具有脉冲响应$h(t)$的因果和稳定LTI系统的输入，稳态输出为
$$y(t)=e^{j2\pi f_0 t}H(f_0)$$
  - ✓ 若将信号在频域表示，也即将信号表示为正弦波的线性组合，就可以利用线性和叠加的优势，将每个正弦波输入到系统中，然后将每各个正弦波的响应相加，得到整体响应，也即其通过系统的输出信号可由一系列的系统频域响应刻画。
- 利用上述特性，信号和系统的频率表示在信号处理和通信中非常重要
  - ✓ 通信系统中消息的调制、带宽的含义
  - ✓ 滤波器设计
  - ✓ 模拟信号的采样，是模拟信号和数字信号处理之间的桥梁

傅立叶简介记及信号分析工具

- 让•巴普蒂斯•约瑟夫•傅里叶（Baron Jean Baptiste Joseph Fourier，1768-1830），男爵，法国数学家、物理学家，1768年3月21日生于欧塞尔，1830年5月16日卒于巴黎。1817年当选为科学院院士，1822年任该院终身秘书，后任法兰西学院终身秘书和理工科大学校务委员会主席
- 傅立叶于1807年在法国科学学会上发表一篇论文：任何连续周期信号都可以表示为一组适当加权的正弦曲线之和
- 审查这篇论文的人，包括了两位著名的数学家拉格朗日（Joseph Louis Lagrange）和拉普拉斯（Pierre Simon de Laplace）
  - ✓ 拉普拉斯和其他审查者同意发表，但拉格朗日坚决反对，他认为傅立叶的方法无法表示带有棱角的信号
  - ✓ 法国科学学会屈服于拉格朗日的威望，拒绝了傅立叶的论文。知道拉格朗日死后15年，这篇论文才被发表出来
- 信号频谱分析的工具
  - ✓ 周期连续时间信号：傅里叶级数（Fourier Series)
  - ✓ 非周期连续时间信号：傅里叶变换(Fourier Transform)
  - ✓ 离散时间信号：离散时间傅里叶变换(Discrete Time Fourier Transform)和离散傅里叶变换(Discrete Fourier Transform)

	Continuous Frequency	Discrete Frequency
Continuous Time	CTFT	CTFS
Discrete Time	DTFT	DFT

傅里叶变换

- 周期信号是一种特殊信号，而一般信号大多是非周期信号
- 傅里叶变换是在频域中表示在时域中一般非周期时域信号的方法，也是研究非周期信号频谱的工具
- 傅里叶变换和反变换

$$X(\omega)=\int_{-\infty}^{+\infty} x(t)e^{-j\omega t}\mathrm{d}t=\int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}\mathrm{d}t$$

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty} X(\omega)e^{j\omega t}\mathrm{d}\omega=\int_{-\infty}^{+\infty} X(f)e^{j2\pi ft}\mathrm{d}f$$

  - ✓ 傅里叶变换存在的充分条件是在无限区间内绝对值可积 $\int_{-\infty}^{+\infty}|x(t)|\mathrm{d}t<\infty$
- $F(j\omega)$称为$f(t)$的频谱密度函数，简称频谱，可表示为
  - ✓ $F(j\omega)=R(\omega)+jX(\omega)=|F(j\omega)|e^{j\varphi(\omega)}$
  - ✓ $R(\omega)=\int_{-\infty}^{\infty} f(t)cos(\omega t)\,dt$,为实部谱，$X(\omega)=-\int_{-\infty}^{\infty} f(t)sin(\omega t)\,dt$，为虚部谱
  - ✓ $|F(j\omega)|=\sqrt{R^2(\omega)+X^2(\omega)}$为幅度谱，$\varphi(\omega)=arctan(\frac{X(\omega)}{R(\omega)})$为相位谱
  - ✓ 工程上，一般使用幅度谱及相位谱表示频谱

符号计算实现傅里叶变换和逆变换

- Symbolic Math Toolbox 提供了能直接求解傅里叶变换及逆变换的函数 fourier()及 ifourier()
  - ✓ F=fourier(f)：默认变量为x，返回是关于ω的函数
  - ✓ F=fourier(f,v)：返回函数F是关于符号对象v的函数

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

  F=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

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

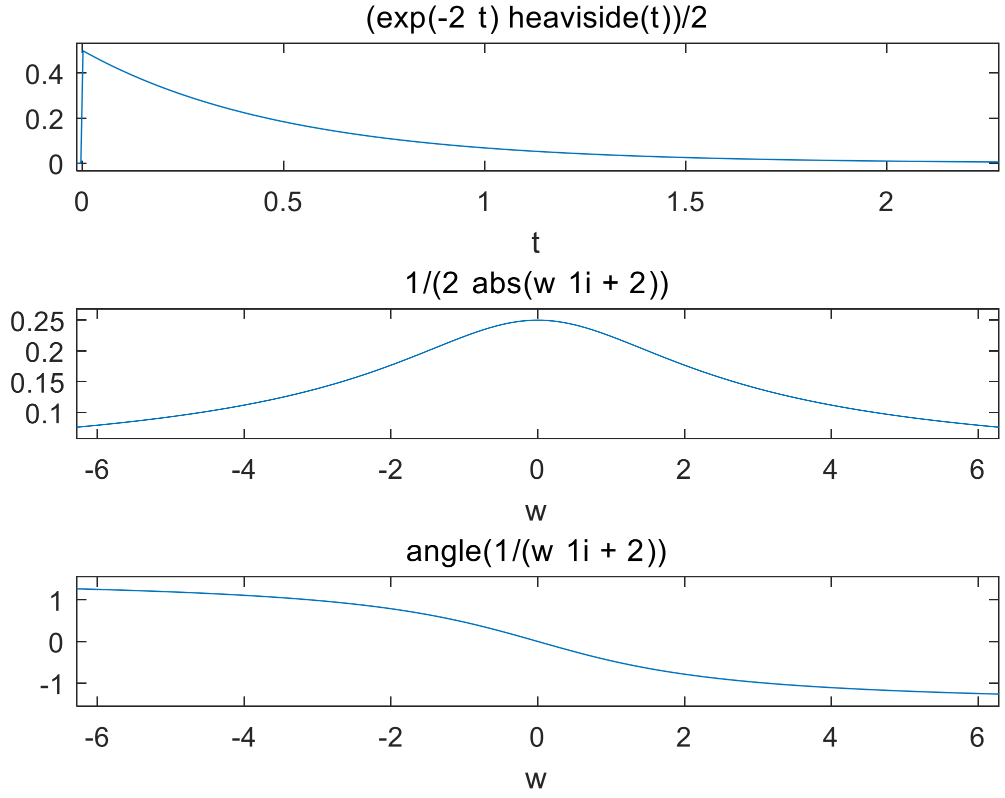
- 反傅里叶函数
  - ✓ f=ifourier(F):缺省的独立变量为ω，缺省返回是关于x的函数
  - ✓ f=ifourier(F,u):返回函数f是u的函数
  - ✓ f=ifourier(F,v,u):对关于v的函数F进行变换，返回关于u的函数f
- 注意：
  - ✓ 调用函数前，要用syms命令对所用到的变量（ 如 t、u、v、w）等申明成符号变量
  - ✓ 对函数f及F，也要用申明为符号表达式
  - ✓ 若f或F是MATLAB中的通用函数表达式，则不必用sym 加以说明

Time Domain	Frequency Domain	Commands	Results
$x(t)$	$X(\Omega)$	`syms t w w0 t0`	
$\delta(t)$	1	`x=dirac(t);` `fourier(x,w)`	`ans = 1`
1	$2\pi\delta(\Omega)$	`fourier(1,w)`	`ans = 2*pi*dirac(w)`
$u(t)$	$(1/j\Omega)+\pi\delta(\Omega)$	`X=1/(j*w)+pi*dirac(w);` `ifourier(X,t)`	`ans = heaviside(t)`
$\delta(t-t_0)$	$e^{-j\Omega t_0}$	`x=dirac(t-t0);` `fourier(x,w)`	`ans = exp(-i*t0*w)`
$e^{j\Omega_0 t}$	$2\pi\delta(\Omega-\Omega_0)$	`X=2*pi*dirac(w-w0);` `ifourier(X,t)`	`ans = exp(i*w0*t)`
$\cos(\Omega_0 t)$	$\pi\delta(\Omega-\Omega_0)$ $+\pi\delta(\Omega+\Omega_0)$	`X=pi*(dirac(w-w0)+dirac(w+w0));` `ifourier(X,t)`	`ans = cos(w0*t)`
$\sin(\Omega_0 t)$	$(\pi/j)\delta(\Omega-\Omega_0)$ $-(\pi/j)\delta(\Omega+\Omega_0)$	`X=(pi/j)*(dirac(w-w0)` `-dirac(w+w0));` `x= ifourier(X,t)`	`x= sin(w0*t)`
$e^{-at}u(t)$, $\mathrm{Re}(a)>0$	$1/(j\Omega+a)$	`a=8;` `x=exp(-a*t)*heaviside(t);` `X= fourier(x,w)`	`X = 1/(8+i*w)`
$te^{-at}u(t)$, $\mathrm{Re}(a)>0$	$1/(j\Omega+a)^2$	`x=t*exp(-a*t)*heaviside(t);` `fourier(x,w)`	`ans = 1/(8+i*w)^2`
$(t^{n-1}/(n-1)!)e^{-at}u(t)$, $\mathrm{Re}(a)>0$	$1/(j\Omega+a)^n$	`n=4;` `X=1/(j*w+a)^n;` `ifourier(X,t)`	`ans = 1/6*t^3` `*exp(-8*t)` `*heaviside(t)`

课堂小练习

- 设$f(t) = e^{-2t}\varepsilon(t)$,试画出$f(t)$及其频谱（幅度谱和相位谱）
  - ✓ 程序：simFouier.m

```
syms t v w x;
x=exp(-2*t)*heaviside(t); %符号定义的时域信号
X=fourier(x);             % 傅里叶变换
subplot(311);
fplot(x);                 %时域信号波形
title(char(x))
xlabel('t')
subplot(312);
fplot(abs(X))             %幅度谱
title(char(abs(X)))
xlabel('w')
subplot(313);
fplot(angle(X))           %相位谱
title(char(angle(X)))
xlabel('w')
```



离散时间傅里叶分析

- 将连续时间傅里叶变换（CTFT）的积分替换成求和符号，就得到离散时间傅里叶变换（DTFT）

$$X(\omega)=\int_{-\infty}^{+\infty} x(t)e^{-j\omega t}\mathrm{d}t \quad \Rightarrow \quad X(\omega)=\sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$$

  - ✓ DTFT是循环频率ω的连续复值函数，是周期为2π的周期函数
  - ✓ 信号$x[n]$的DTFT $X(\omega)$存在的一个充分条件是 $\sum_{n=-\infty}^{+\infty}|x[n]|<\infty$

- 反离散时间傅里叶变换为

$$x[n]=\frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}d\omega=\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega.$$

- MATLAB中的计算方法
  - ✓ 符号计算
  - ✓ 数值计算
  - ✓ fft函数（稍后介绍）

DTFT的符号计算和数值计算

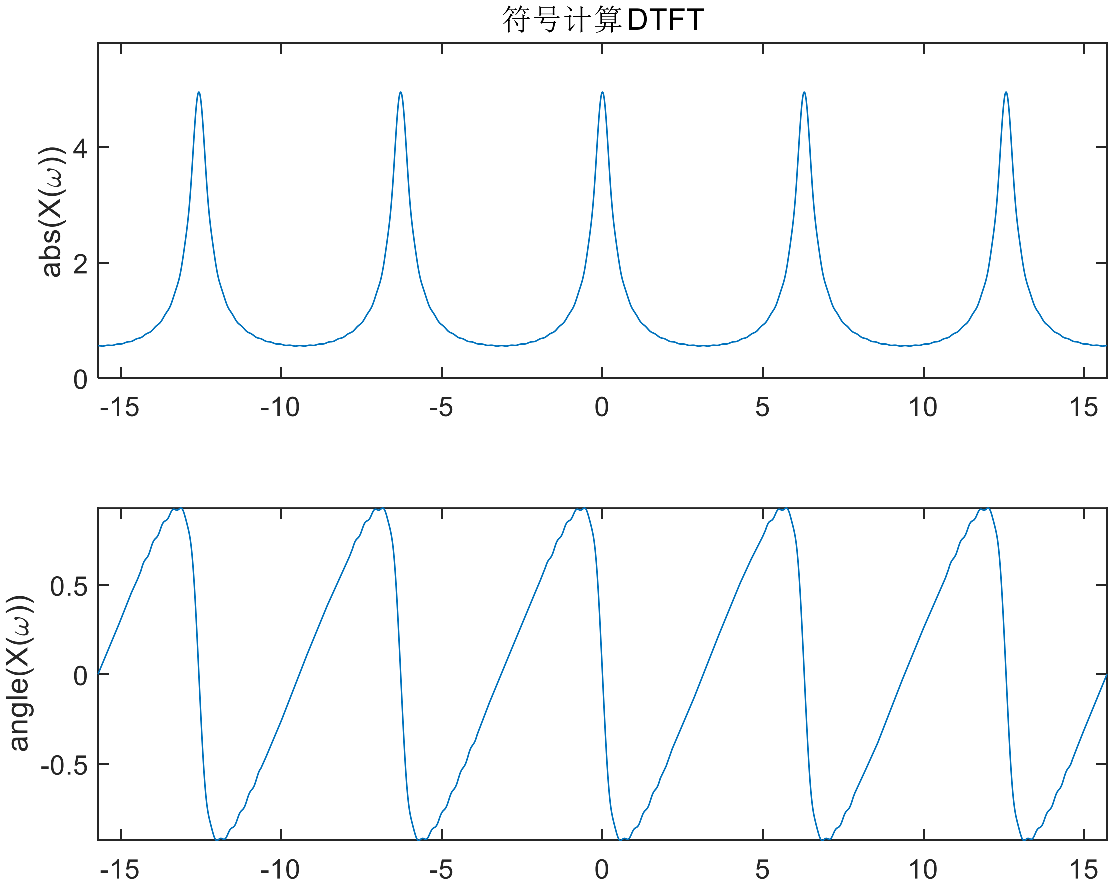
- 问题：给定时域信号x[n]，绘制-N*pi~N*pi上的DTFT
- 符号计算
  - ✓ 给定符号变量w，计算 $X(\omega)=\sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$
  - ✓ 利用fplot，绘制[-N*pi, N*pi]上abs(X)或phase(X)
  - ✓ 利用eval计算数值结果
- 数值计算
  - ✓ 给定变量w，在[-pi,pi)区间上均匀地取K个样点 $\omega = k\frac{2\pi}{K}, k = 0,1,\cdots,K-1$
  - ✓ 若频率跨度[-N*pi~N*pi),则k的取值为 $k = \mathrm{floor}((-N*K/2+0.5):(N*K/2-0.5));$
  - ✓ 对应的频率矢量为 $\omega = k\frac{2\pi}{K} = k\cdot \mathrm{d}\omega$
  - ✓ 根据w的数值，计算DTFT： $X(\omega)=\sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$

基于符号计算的DTFT

- 例：计算信号$x[n]=0.8^n\ 0\le n\le 20$X[n]的DTFT $X(\omega)$并在频率范围$[-5\pi,5\pi]$内绘制
  - ✓ 程序:simDTFT.m

$$X(\omega)=\sum_{n=-\infty}^{+\infty}x[n]e^{-j\omega n}$$

```
N=5;      %频率跨度, -N*pi~N*pi
%符号计算
syms w
n= 0:20;
x=0.8.^n;
X=sum(x.*exp(-1i*w*n));
subplot(211)
fplot(abs(X),[-N*pi N*pi]);
ylabel('abs(X(\omega))')
title('符号计算 DTFT')
ylim([0 5.8])
subplot(212)
fplot(phase(X),[-N*pi N*pi]);
xlim([-N*pi N*pi])
ylabel('angle(X(\omega))')
```



基于数值计算的DTFT

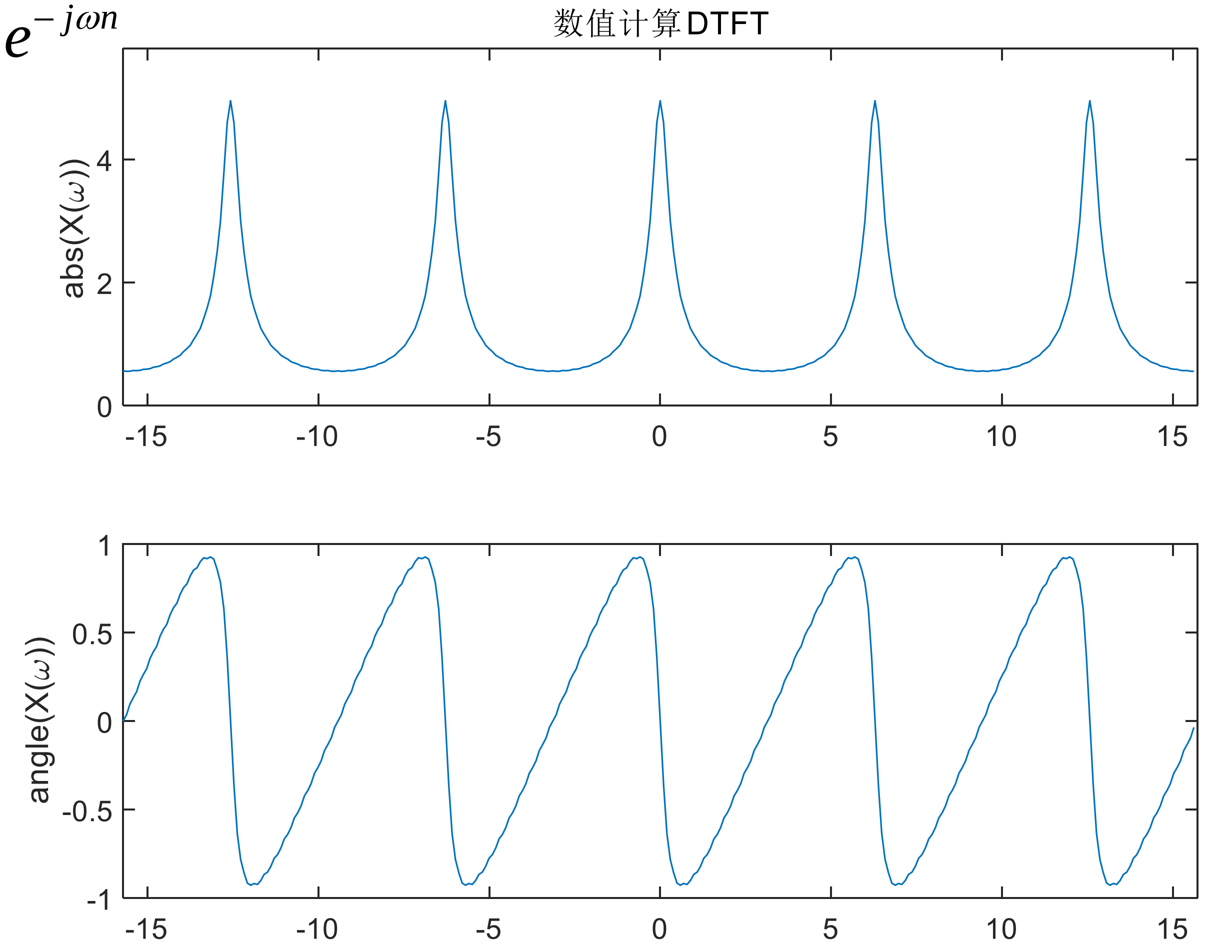
程序：simDTFT.m

$$\omega = k\frac{2\pi}{K} = k \cdot \mathrm{d}\omega$$

$$k = \mathrm{floor}((\text{-}N * K \,/\, 2 + 0.5) : (N * K \,/\, 2 \text{ - } 0.5));$$

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\omega n}$$

```
%数值计算
K=64;         %2pi 区间内的采样点数
dw=2*pi/K;
k=floor((-N*K/2+0.5):(N*K/2-0.5));
X=x*exp(-1i*dw*n'.*k);
w=k*dw;
figure
subplot(211)
plot(w,abs(X));
xlim([-N*pi N*pi])
ylabel('abs(X(\omega))')
title('数值计算 DTFT')
ylim([0 5.8])
subplot(212)
plot(w,phase(X));
xlim([-N*pi N*pi])
ylabel('angle(X(\omega))')
```



离散傅里叶变换

- 为了在计算机上实现对信号的DTFT分析，需要从频率ω中获取样本$\omega_k = 2\pi k/N$,这引出了适合于离散时间信号的第二种傅里叶变换，即离散傅里叶变换（DFT）
- 离散时间信号x[n]的N点DFT表示为

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\omega n} \Rightarrow X_k = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}}, \quad k = 0, 1, \ldots, N-1.$$

✓ X_k通常是复数。因此，它们可以用极性形式表示为

$$X_k = |X_k| e^{j\angle X_k}, \quad k = 0, 1, \ldots, N-1,$$

➢ 或者

$$X_k = \mathrm{Re}\{X_k\} + j\,\mathrm{Im}\{X_k\}, \quad k = 0, 1, \ldots, N-1,$$

$$\mathrm{Re}\{X_k\} = x(0) + \sum_{n=1}^{N-1} x[n] \cos\left(\frac{2\pi nk}{N}\right) = \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi nk}{N}\right) \qquad \mathrm{Im}\{X_k\} = -\sum_{n=0}^{N-1} x[n] \sin\left(\frac{2\pi nk}{N}\right)$$

反离散傅里叶变换

- 假设离散时间信号 $x[n]$ 的DFT X_k 是已知的，信号 $x[n]$ 可以从n个DFT点 X_k，$k=1,2,\ldots,N$ 的逆离散傅里叶变换（IDFT）得到。序列X_k的IDFT为

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X_k e^{j\frac{2\pi nk}{N}},\quad n=0,1,\ldots,N-1$$

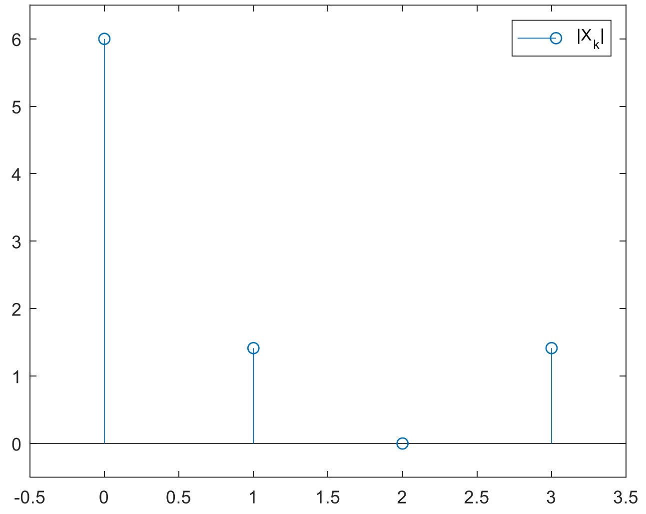
- 例：计算序列的IDFT $X_k=[6,\ -1-j,\ 0,\ -1+j],\ 0\le k\le 3.$
  - ✓ 程序：simIDFT.m

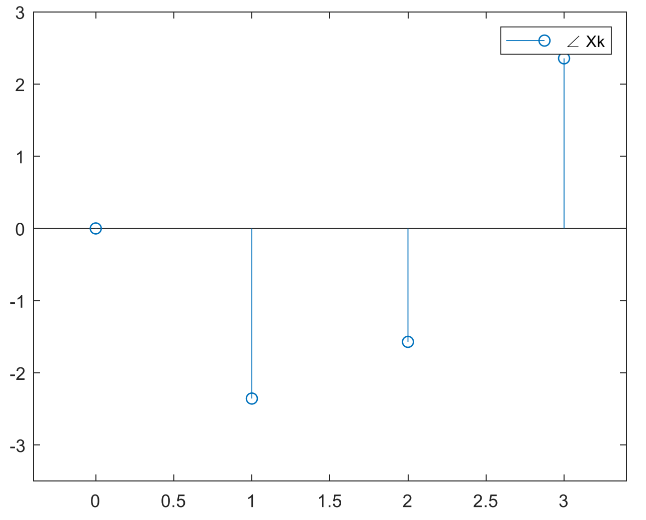
```
Xk=[6, -1-j,0,-1+j];
N=length(Xk);
for n=0:N-1
    for k=0:N-1
        xn(k+1)=Xk(k+1)*exp(j*2*pi*n*k/N);
    end
    x(n+1)=sum(xn);
end
x=(1/N)*x
```

DFT的计算示例

- 计算序列x[n]=[1，2，2，1] $0 \le n \le 3$的DFT X_k，绘制X_k的大小、相位。
  - ✓ 程序：simDFT.m

```
x=[1 2 2 1];
N=length(x);
for k=0:N-1
    for  n=0:N-1
        X(n+1)=x(n+1)*exp(-j*2*pi*k*n/N);
    end
    Xk(k+1)=sum(X);
end
mag=abs(Xk);
stem(0:N-1,mag);
legend ('|X_k|')
figure
phas=angle(Xk);
stem(0:N-1,phas);
legend ('\angle Xk')
```





自定义函数化DFT和IDFT

□ DFT函数

$$X_k=\sum_{n=0}^{N-1}x[n]e^{-j\frac{2\pi nk}{N}},\quad k=0,1,\ldots,N-1.$$

```
function Xk = dft(x)
% 离散傅里叶变换
% 输入参数
% x - 时域信号
% 输出参数
% Xk -频域信号

N=length(x);
Xk=zeros(1,N);
X=zeros(1,N);
 for k=0:N-1
     for n=0:N-1
       X(n+1)=x(n+1)*exp(-1i*2*pi*k*n/N);
     end
     Xk(k+1)=sum(X);
end
```

□ IDFT函数

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X_k e^{j\frac{2\pi nk}{N}},\quad n=0,1,\ldots,N-1$$

```
function x=idft(Xk)
% 逆离散傅里叶变换
% 输入参数
% Xk - 频域信号
% 输出参数
% x -  时域信号

N=length(Xk);
xn=zeros(1,N);
x=zeros(1,N);
for n=0:N-1
   for k=0:N-1
     xn(k+1)=Xk(k+1)*exp(1i*2*pi*n*k/N);
   end
x(n+1)=sum(xn);
end
x=(1/N)*x;
```

快速傅里叶变换（FFT）

- 直接从N点DFT或IDFT的定义计算它可能是一个计算代价很高的过程
  - ✓ $\mathrm{O}(N^2)$乘法操作
  - ✓ 如果离散时间信号$x[n]$是复数，事情就会变得困难，因为两个复数的一个乘法需要实数之间的四个乘法
  - ✓ 直接从定义计算DFT（或IDFT）对于实时应用来说通常太慢
- 快速傅里叶变换是计算DFT的一种高效算法
  - ✓ FFT是基于“分而治之”技术的
  - ✓ 把N个点的原始问题分成两个N/2点的对称子问题，如果n/2是偶数，则将n/2分为两个n/4的子问题
  - ✓ $O(N * \log(N))$
- 使用方法
  - ✓ X=fft(x,N) :x为离散的时域信号，N为采样点数（可省略），X为离散的频域信号
  - ✓ x=ifft(X,N)：

验证fft

✓ 程序：**simTime4DFT_FFT.m**

```
N=2^12;
n=0:N-1;
x=(-1).^n;

disp('DFT')
tic
X=dft(x);
toc

disp('FFT')
tic
X=fft(x);
toc
```

```
N =

        4096

DFT
时间已过 3.941886 秒。
FFT
时间已过 0.005693 秒。
```

应用：用FFT计算有限长离散时间序列的频谱

有限长离散时间序列频谱的求解过程

- ✓ 计算一个序号从$n_1 \sim n_2$的有限长序列的频谱
- ✓ 若$n_1 \neq 0$，则通过补零，构造从$n = 0$的主值区间序列
- ✓ 若主值区间序列的序列长度为N,调用N 点fft函数，输出的数据为在$\omega_k = 2\pi k/N$上的频谱数据
- ✓ 调用fftshift，得到$\left[-\frac{\pi}{T}, \frac{\pi}{T}\right)$区间上$N$个频点的频谱

关于频谱分辨率

- ✓ 频谱分辨率：频点之间的最小间隔
- ✓ 若N 较小，频点之间的间距较大，频谱分辨率较低
- ✓ 通过在原序列后面补零，增加序列长度，提高频谱分辨率

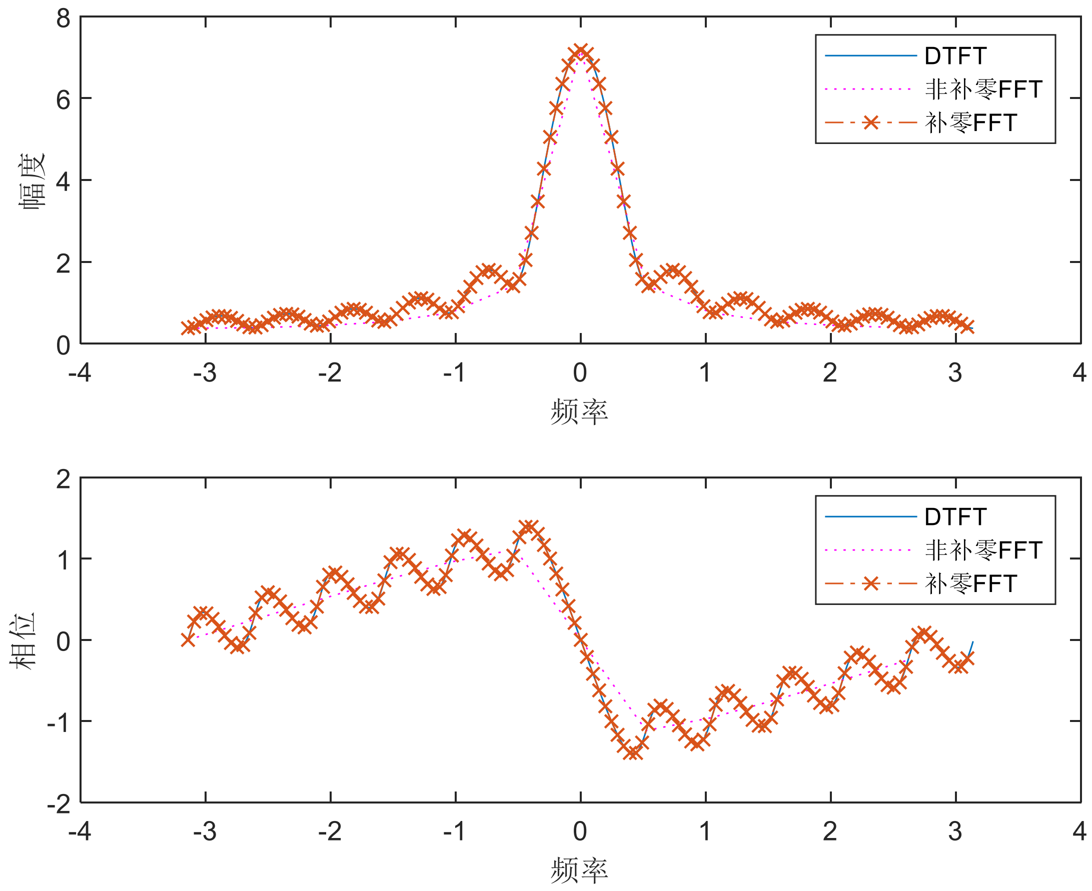
例：用FFT计算有限长离散时间序列的频谱

□ 例：用FFT计算序列$x[n]=0.9^n, n=0,1,\cdots,11$ 的频谱

✓ 程序simFFT4DTFT.m

$$k=\text{floor}((-N*K/2+0.5):(N*K/2-0.5));$$

```
nx=0:11;
x=0.9.^nx;
nx=0:11;

%% fft 求解
N1=length(x);
dw1=2*pi/(N1);                    % 求出序列长度及频率分辨率
k1=floor((-(N1-1)/2):((N1-1)/2)); % 求对称于零频率的 FFT 位置向量
X1=fftshift(fft(x,N1));           % 求对称于零频率的 FFT 序列值

%% 补零 fft，提高频率分辨率
N2=128;
dw2=2*pi/(N2);                    % 求出序列长度及频率分辨率
k2=floor((-(N2-1)/2):((N2-1)/2)); % 求对称于零频率的 FFT 位置向量
X2=fftshift(fft(x,N2));           % 求对称于零频率的 FFT 序列值

%% DTFT 计算法
w=-pi:0.01:pi;                  % 生成连续频谱的细分频率
X=x*exp(-1i*nx'*w);               % 计算连续频谱
```



应用：用FFT计算无限长离散时间序列的频谱

- 对无限长离散时间序列，由于存在的数据无限多，因此频率分辨率过低的问题是不存在的
- 在计算机上进行计算时，总需要吧一个无限序列截断成一个有限序列，过短的截断会导致频谱分析的失真，过长的截断又会导致计算量过大
- 如何确定截取序列长度?
  - ✓ 1.初始化$N_1 = N$ 和$N_2 = 2N$;
  - ✓ 2.计算N_1和N_2两种截断长度下的频谱;
  - ✓ 3.计算两个长度下共同频点上最大的相对误差
  - ✓ 4.若相对误差大于门限（如1%），则更新$N_1 = 2N$ 和 $N_2 = 2N_1$，转到步骤2;
  - ✓ 5.输出长度为N_2时的频谱作为无限长离散时间序列的近似频谱。

傅里叶变换与离散傅里叶变换的关系

- 理论上，连续时间信号的频谱是由傅里叶变换求取。但在实际中，大多数信号都不能采用这种方法。通常是对连续时间信号进行采样和截断
- 连续时间信号的傅里叶变换可以通过离散傅里叶变换近似
  - ✓ 用采样时间t采样信号$x(t)$，得到离散时间信号$x[nT]$，$n = 0,1,2 \ldots N-1$
  - ✓ 计算$x[nT]$的离散傅里叶变换，得到X_k，$\boldsymbol{k} = 0,1,2 \ldots N-1$
  - ✓ 原来连续时间信号在$\boldsymbol{\omega}_k = 2\boldsymbol{\pi} k/NT$处的频谱为

$$X(\omega_k) = NT_s \frac{1-e^{-j\frac{2\pi k}{N}}}{j2\pi k} X_k \quad k = 0,1,\cdots,N-1$$

  - ✓ 随着N变大，连续时间信号的频谱$X(\boldsymbol{\omega})$可由$X(\boldsymbol{\omega}_k)$近似表示
- 在Matlab中，对模拟信号进行采样，采样频率要满足奈奎斯特定律
  - ✓ 带限连续信号：提高采样频率使之满足时域采样定理
  - ✓ 非带限连续信号：增加抗混叠滤波（一个低通滤波器），使之成为带限信号，减少混叠误差

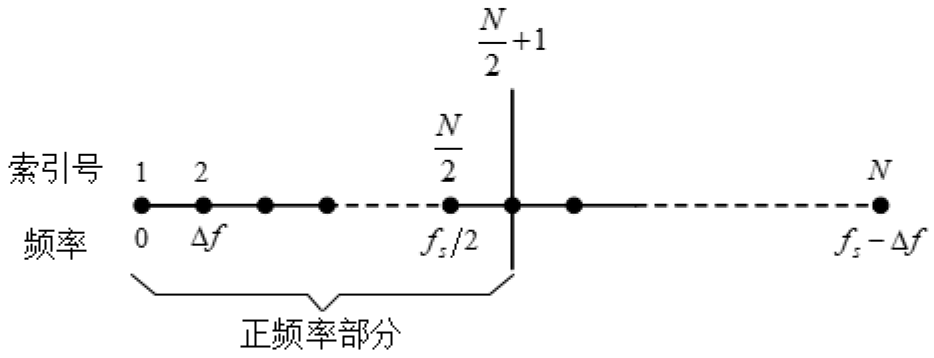
应用：用FFT计算连续时间信号的频谱

- 信号采样
- 调用fft函数
- 设置频谱刻度（频谱图的横坐标）
- 若为双边谱，调用fftshift，将零频分量调整到频谱中心

```
X =[0   1   2   3   4   5   6   7   8   9   10   11   12   13   14]
>> fftshift(X)
ans =[8   9   10   11   12   13   14   0   1   2   3   4   5   6   7]
```

- 基于输出，计算信号的频谱，包括幅度和相位（频谱图的纵坐标）
- 画图展示（可选）

设置频谱刻度——N为偶数



- 信号长为N，采样频率为f_s，在DFT（FFT）以后信号的频谱在-f_s/2和f_s/2之间，谱线之间的间隔为

$$\Delta f = \frac{f_s}{N} = \frac{1}{NT_s}$$

  - ✓ 频谱刻度从0开始，最大频率f_s/2
  - ✓ 注意：实际不存在大于f_s/2的频率分量，大于f_s/2的频率分量实际是负频率的分量

- 双边谱的频谱刻度

$$frcq = (0:N-1)\times\frac{f_s}{N}-\frac{f_s}{2}$$

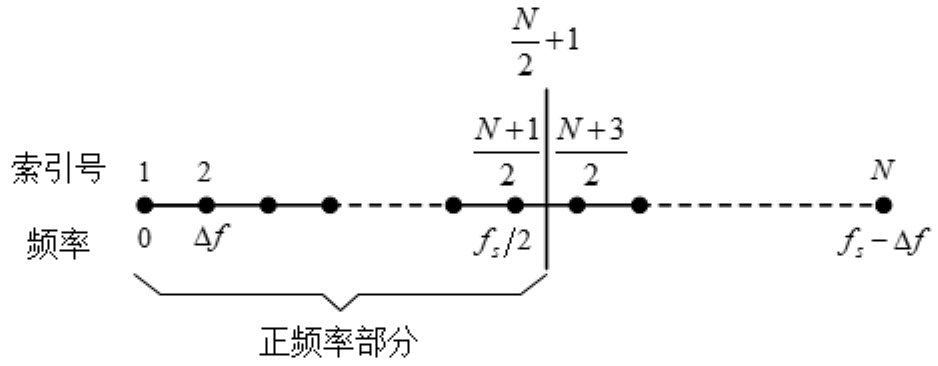
  - ✓ 注意：不包括f_s/2

- 单边谱的频谱刻度

$$frcq = \left(0:\frac{N}{2}-1\right)\times\frac{f_s}{N}$$

  - ✓ 不包括f_s/2，一共N/2个点

设置频谱刻度——N为奇数



□ 双边谱的频谱刻度

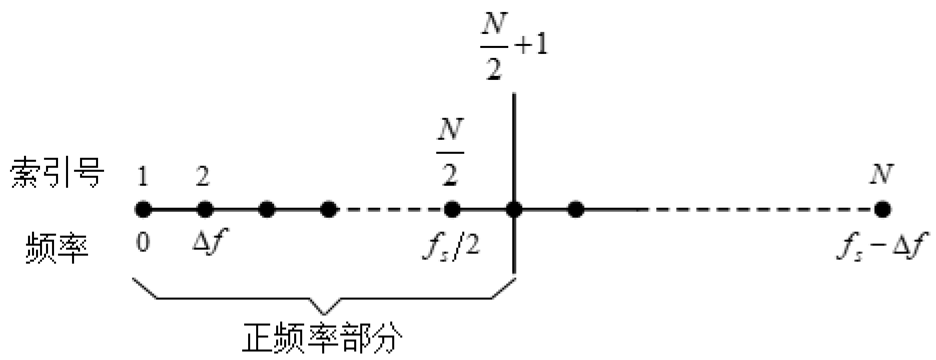
$$frcq=\left(0:N-1\right)\times\frac{f_s}{N}-\frac{\left(N-1\right)f_s}{2N}$$

□ 单边谱的频谱刻度

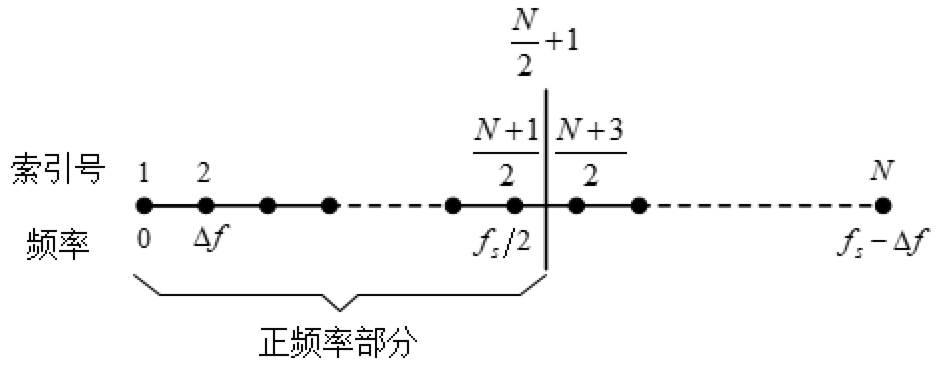
$$frcq=\left(0:\frac{N}{2}\right)\times\frac{f_s}{N}$$

设置频谱刻度——统一表示

N为偶数



N为奇数



- **双边谱的频谱刻度**

$$freq = floor\left(\left(-\frac{N}{2}+0.5\right):\left(\frac{N}{2}-0.5\right)\right)\times\frac{f_s}{N}$$

- **单边谱的频谱刻度**

$$freq = floor\left(0:\left(\frac{K}{2}-0.5\right)\right)\times\frac{f_s}{N}$$

幅度谱和相位谱（纵坐标）

- 双边谱（幅度谱和相位谱）

$$\begin{cases} A = \dfrac{\left|X_k\right|}{N} \\ \theta = \arctan X_k \end{cases}$$

- 单边谱（幅度谱和相位谱）

$$\begin{cases} A = \begin{cases} \dfrac{\left|X_k\right|}{N} & k = 0 \\ 2\dfrac{\left|X_k\right|}{N} & k \neq 0 \end{cases} \\ \theta = \begin{cases} 0 & k = 0 \\ \arctan X_k & k \neq 0 \end{cases} \end{cases}$$

示例：连续时间信号的频谱分析

- 例：有一信号为$x(t) = 0.8 + 2cos(50\pi t + \pi/6)$，试用FFT分析该信号的单边谱和双边谱
  - ✓ 程序：simCosSpectrum.m

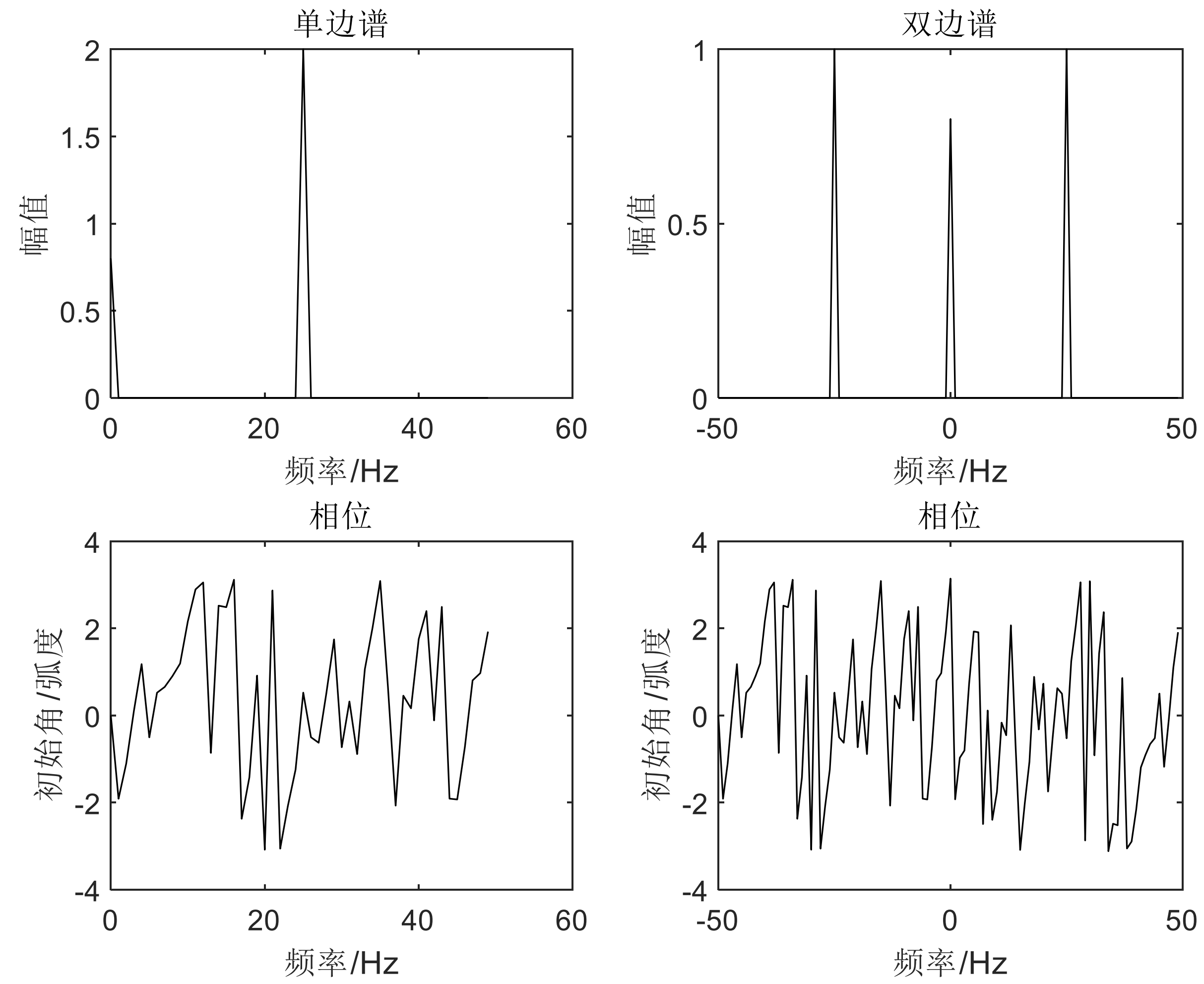
$$freq = floor\left(0:\left(\frac{K}{2}-0.5\right)\right)\times\frac{f_s}{N}$$

$$freq = floor\left(\left(-\frac{N}{2}+0.5\right):\left(\frac{N}{2}-0.5\right)\right)\times\frac{f_s}{N}$$

```
fs=100;                         % 采样频率
N=100;                          % 信号长度
t=(0:N-1)/fs;                   % 设置时间序列
f1=25;                          % 信号频率

x=0.8+2*cos(2*pi*f1*t+pi/6);      % 信号采样
X=fft(x);                       % FFT

% 单边谱
Y=zeros(1,N);
freq=(0:N/2-1)*fs/N;            % 设置频率刻度
Y(1)=abs(X(1))/N;               % 直流分量的幅度调整
Y(2:end)=abs(X(2:end))*2/N;     % 计算幅值
Theta=angle(X);                 % 计算相位
```

```
subplot 221; plot(freq,Y(1:N/2),'k');
xlabel('频率/Hz'); ylabel('幅值'); title('单边谱');
subplot 223; plot(freq,Theta(1:N/2),'k');
xlabel('频率/Hz'); ylabel('初始角/弧度'); title('相位');

% 双边谱
freq=(0:N-1)*fs/N-fs/2;                    % 设置频率刻度
Y=fftshift(abs(X)/N);                      % 计算幅值
subplot 222; plot(freq,Y,'k');
xlabel('频率/Hz'); ylabel('幅值'); title('双边谱');
subplot 224; plot(freq,Theta,'k');
xlabel('频率/Hz'); ylabel('初始角/弧度'); title('相位');
```

示例：连续时间信号的频谱分析（续）

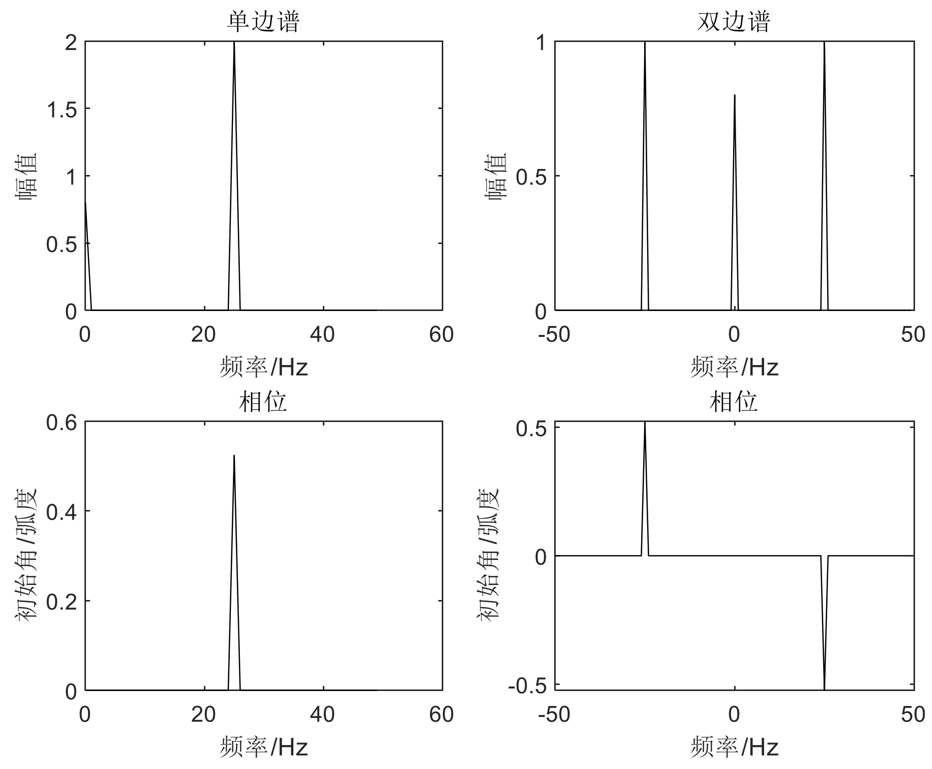


为什么相位谱混乱？？

混乱相位谱的噪声处理

- 对数据进行观察，发现X(k)很多点都很小，因此混乱的相位谱是由于计算误差产生
  - ✓ FFT运算后，在非信号频率的位置，频率分量都非常小，通常都在10负十几次幂的量级。但这些微小的频率分量还是可以计算出相位
- 可在程序中设置一个阈值Th，大于这个值才计算相位
  - ✓ 程序：simCosSpectrum2.m

```
%消除相位噪声
Th=0.1;                        % 设置阈值
thetadex=find(Y<Th);          % 寻找小于阈值的那线谱线的索引
Theta(thetadex)=0;            % 对于小于阈值的那线谱线初始相位都为 0
```

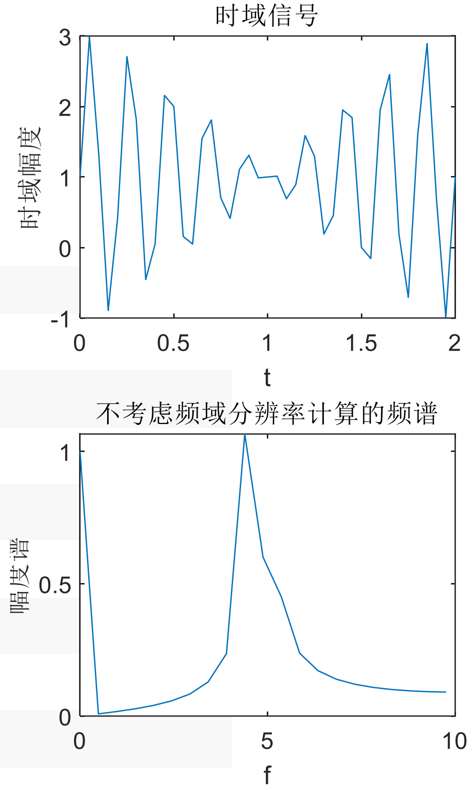


频谱分析的栅栏现象

- 栅栏现象：DFT/FFT得到的频谱只是对时间连续信号的连续频谱在有限离散频点上的采样值，无法反映频率采样点之间频率上的细节。有时有些重要的信息正好在频率采样点之间，因其强度过小而检测不到。
  - ✓ 解决办法：缓解该问题的主要方法时在序列后补零，以增加频域上的分量数目，减少频率间隔，提高频率分辨率。
- 两种含义：物理分辨率和计算分辨率
  - ✓ 物理分辨率：谱分析中将信号中两个靠得很近的谱峰仍然能保持分辨的能力
    - ➢ 假定信号截至频率为f_c，为满足采样定理，采样频率满足$f_s \geq 2f_c$
    - ➢ 若要能分辨间隔为f_δ的两个谱峰，数据序列长度M应满足$M \geq f_s/f_\delta$
    - ➢ 物理分辨率定义为f_s/M
  - ✓ 计算分辨率：在DFT时，频率轴上所能得到的最小频率间隔$\Delta f = f_s/N$
- 注意:N通常大于M，是通过对原序列补零获得
  - ✓ 通常序列的长度M不是2的幂
  - ✓ 为提高运算效率，在序列后补零，使得长度变为2的幂

示例：栅栏现象及其处理

- 例：有一信号为$x(t) = 1 + sin(2\pi f_1 t) + sin(2\pi f_2 t)$，且$f_1 = 5$，$f_2 = 4.5$，试分析其频谱
  - ✓ 主程序: simPicketFence.m
  - ✓ 支持程序：SimpleSpectrumViewer.m




```
%% 不考虑频率分辨率的频率分析
Fs = 20; Ts = 1/Fs; f1=5; f2=4.5;
t = 0:Ts:2;
xt = 1+sin(2*pi*f1*t) + sin(2*pi*f2*t);
[Xf, f] = SimpleSpectrumViewer(xt, Fs);

subplot(221)
plot(t, xt)
xlabel('t'); ylabel('时域幅度'); title('时域信号')
subplot(223)
plot(f, abs(Xf));xlabel('f');ylabel('幅度谱')
title('不考虑频域分辨率计算的频谱')
```

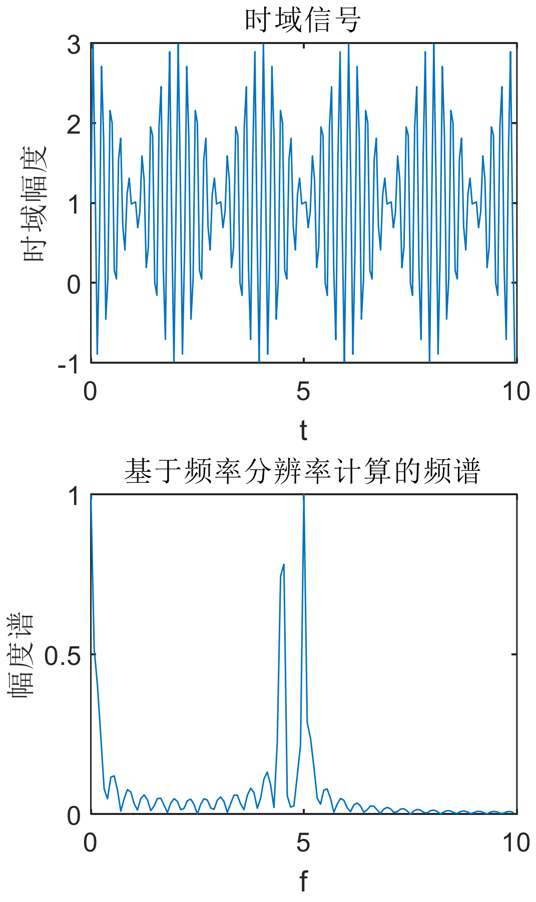
```
function [Xf, f] = SimpleSpectrumViewer(xt, Fs)
%不考虑频域分辨率计算的频谱（单边谱）
% 输入参数
% xt:时序信号采样序列; Fs:采样频率
% 输出参数
% Xf:频域分量; % f: 频域刻度

N =length(xt);                    %序列长度
X= fft(xt,N);                     %fft 计算

Xf=zeros(1,floor(N/2-0.5)+1);     %频域分量存储空间，长度为 floor(N/2-0.5)+1
Xf(1)=X(1)/N;                     %直流分量
Xf(2:end) = 2*X(floor(1:N/2-0.5)+1)/N; %非直流分量频谱调整（单边谱）
f =floor(0:N/2-0.5)*Fs/N;         %频域刻度
end
```

示例：栅栏现象及其处理（续）

◻ 设定频率分辨率

- ✓ 根据采样率和频率分辨率设定最低点数：$M \geq f_s/f_\delta$
- ✓ 为了加速计算，再调整到2的幂：N =2^(max(nextpow2(M),nextpow2(L)))
- ✓ 主程序: simPicketFence.m
- ✓ 支持程序：SpectrumViewer.m

```
%% 设定频率分辨率的频率分析
Fdelta=0.1;                                         %频率分辨率
M=Fs/Fdelta;                                        %实现频率分辨率的序列长度
t=0:Ts:(M-1)*Ts;
xt =1+sin(2*pi*f1*t) + sin(2*pi*f2*t);               %信号采样
[Xf, f] = SpectrumViewer(xt,Fs,'onesided',Fdelta);   %设定频率分辨率的频域分析

subplot(222)
plot(t, xt);xlabel('t');ylabel('时域幅度');title('时域信号')
subplot(224)
plot(f, abs(Xf));xlabel('f');ylabel('幅度谱');title('基于频率分辨率计算的频谱')
```



频谱分析函数

```
function [Xf, f] = SpectrumViewer(xt, Fs, freqRange,df)
% 频谱分析（单边谱）
% 输入参数
% xt:时序信号采样序列; Fs:采样频率; freqRange:单边/双边分析; df:最小频率分辨率（计算）
% 输出参数
% Xf:频域分量; f: 频域刻度

if nargin == 2
  M=0;
  freqRange='onesided';
elseif nargin == 3
  M=0;
else
  M=Fs/df;                    %根据计算频谱分辨率要求，设定 FFT 的点数
end
L = length(xt);
N =2^(max(nextpow2(M),nextpow2(L))); %实际 FFT 的点数
X = fft(xt,N);

if strcmp(freqRange,'onesided') % 单边分析
    Xf=zeros(1,floor(N/2-0.5)+1);   %频域分量存储空间，长度为 floor(N/2-0.5)+1
    Xf(1)=X(1)/L;                         %直流分量，注意，除以实际数据长度
    Xf(2:end) = 2*X(floor(1:N/2-0.5)+1)/L; %非直流频谱分量调整（单边谱）
    f =floor(0:N/2-0.5)*Fs/N;             %频域刻度
elseif strcmp(freqRange,'twosided') % 双边分析
    Xf = fftshift(X)/L;           频谱分量调整（双边谱），注意，除以实际数据长度
    f =floor(-N/2+0.5:N/2-0.5)*Fs/N;            %频域刻度
end
end
```

泄露现象及其处理

- 泄露现象：对时间连续信号，由于FFT频率分析时，信号被截断成有限长度序列，将使信号的带宽被扩展了
- 对连续信号进行截断，相当于对连续时间信号乘以一个矩形窗函数

$$x_T(t)=x(t)w_T(t) \Longleftrightarrow X_T(t)=2\pi A_m\int W_T(f-\tau)\delta_{\omega_m}(\tau)d\tau=2\pi A_m T\frac{\sin\left(\frac{\omega-\omega_m}{2}T\right)}{\frac{\omega-\omega_m}{2}T}$$

$$w_T(t)=\begin{cases}1 & -\frac{T}{2}<t\le\frac{T}{2}\\ 0 & \text{其他}\end{cases} \Longleftrightarrow W_T=T\frac{\sin(\pi Tf)}{\pi Tf}=\frac{\sin(\pi Tf)}{\pi f}=\frac{\sin(\omega T/2)}{\omega/2}$$

- 解决这种情况的主要方法是对截断后的序列，再乘以一个窗函数，也就是“加窗”
  - ✓ 例如：汉宁（Hanning）窗、海明（Hamming）窗、布莱克曼（Blackman）窗
  - ✓ 一般来说，选择第一旁瓣衰减大、旁瓣峰值衰减快的窗函数有利于缓解截断产生的频谱泄露问题。选择何种窗，需要具体情况具体分析，加窗会导致频谱幅值发生变换，需要修正。修正系数根据具体选择的窗函数设定。

常用窗函数和恢复系数

表 1　常用的窗函数的数学表达式、幅值相等恢复系数和功率相等恢复系数

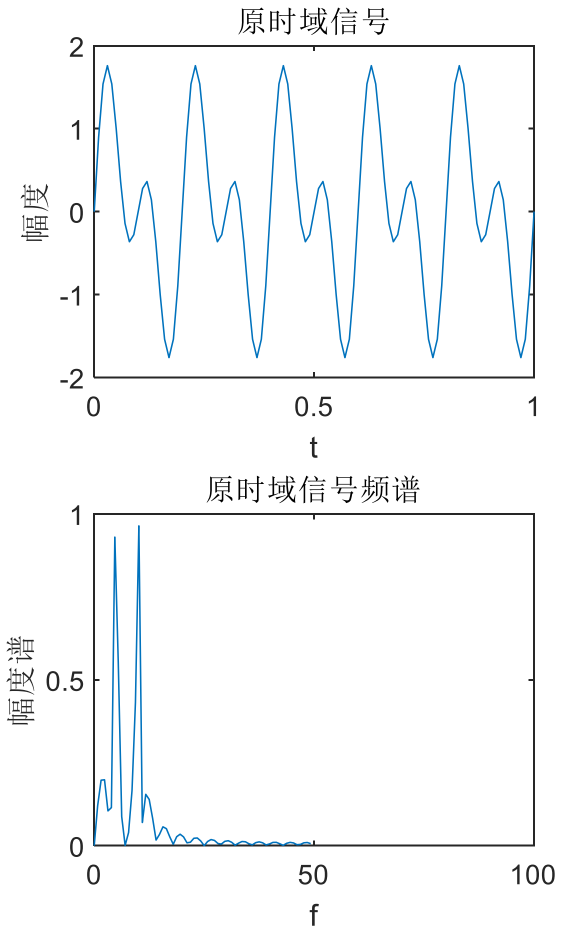
窗函数	数学表达式 (n＝1、2、3…、N)	幅值相等恢复系数	功率相等恢复系数
矩形窗	$W(n)=1$	1	1
汉宁窗	$W(n)=0.5-0.5\cos(2\pi n/N)$	2	1.633
海明窗	$W(n)=0.54-0.46\cos(2\pi n/N)$	1.852	1.586
三角窗	$W(n)=\begin{cases}\dfrac{n}{(N/2)}, & n=1,\cdots,N/2\\ W(N-n), & n=N/2,\cdots,N-1\end{cases}$	2	1.732
高斯窗	$W(n)=e^{-\frac{1}{2}[3(\frac{2n}{N-1})]^2}$	2.396	1.840
指数窗	$W(n)=e^{-\lvert n-1-\frac{N}{2}\rvert/N}$	1.582	1.521
布拉克曼窗	$W(n)=0.42-0.5\cos(2\pi\frac{n-1}{N-1})+0.008\cos(4\pi\frac{n-1}{N-1})$	2.381	1.812
平顶窗	$W(n)\begin{cases}\frac{1}{2}[1-\cos(10\pi n/N)], & n=1,2,3,\cdots,N/10\\ 1, & n=N/10+1,\cdots,9N/10-1\\ \frac{1}{2}\{1+\cos[10\pi(n-\frac{9N}{10})/N]\}, & n=9N/10,\cdots,N\end{cases}$	1.110	1.069
凯泽—贝塞尔窗	$W(n)=I_0\left(3\pi\sqrt{-(2n/N-1)^2}\right)/I_0(3\pi)$	2.480	1.854

表中：$I_0(x)=\sum_{k=0}^{\infty}[(x/2)^k/k!]^2$

示例：泄露现象及其处理

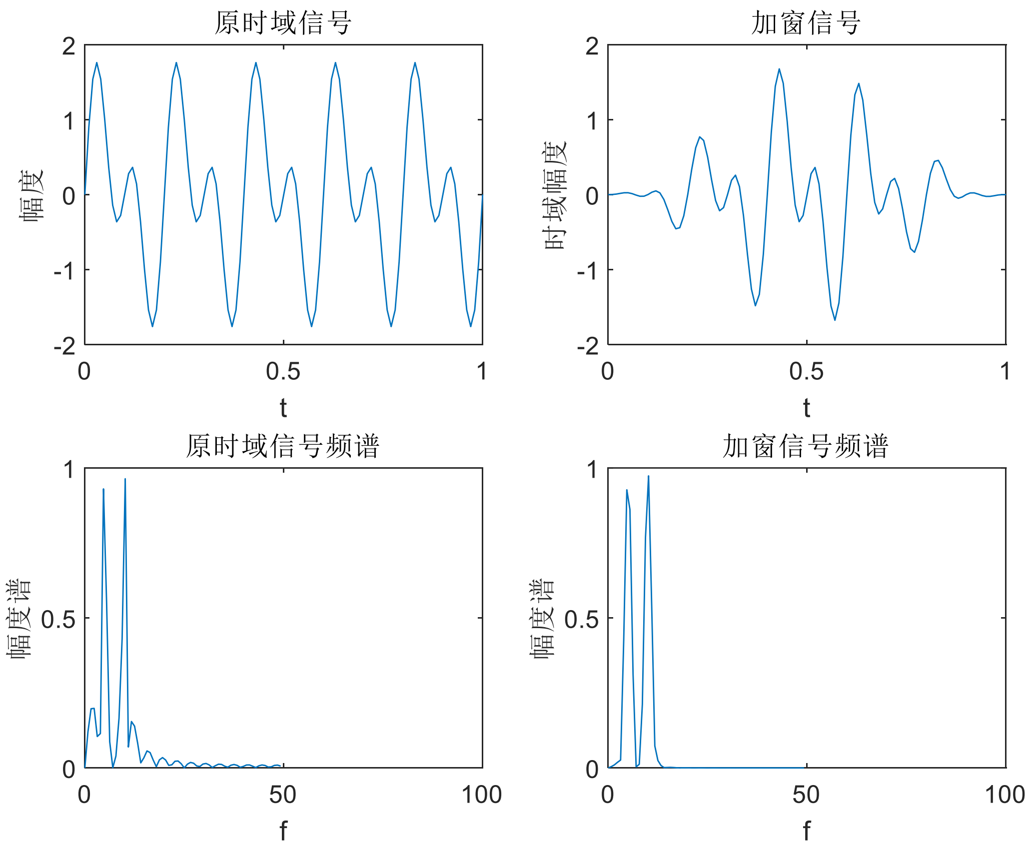
- 有一信号为$x(t)=sin(2\pi f_1 t)+sin(2\pi f_2 t)$，且$f_1=5$，$f_2=10$，试分析其频谱，并用汉宁窗缓解频谱泄露现象
  - ✓ 程序: simAddWindow.m

```
Ts = 0.01;
Fs = 1/Ts;
%% 原始信号
t = 0:Ts:1;
xt = sin(2*pi*5*t) + sin(2*pi*10*t);    %信号采样
[Xf, f] = SpectrumViewer(xt, Fs);       %频谱分析

subplot(221)
plot(t, xt);xlabel('t');ylabel('幅度');title('原时域信号')
subplot(223)
plot(f, abs(Xf));xlabel('f');ylabel('幅度谱')
xlim([0 100]);ylim([0 1])
title('原时域信号频谱')
```



示例：泄露现象及其处理（续）

```
%% 加汉宁窗
win = hann(length(t));
xt1= xt.*win';                          % 加汉宁窗
[X1, f1] = SpectrumViewer(xt1, Fs); % 频谱分析
Xf1=2*X1;                               % 加窗后的幅度修正

subplot(222)
plot(t, xt1);xlabel('t');ylabel('时域幅度')
title('加窗信号')
subplot(224)
plot(f1, abs(Xf1))
xlabel('f');ylabel('幅度谱');title('加窗信号频谱')
xlim([0 100]);ylim([0 1])
```



离散时间系统的描述

- 离散系统：系统的激励和响应均为离散信号
- LTI离散时间系统完全由一个常系数线性差分方程描述，即由一个n阶形式的微分方程

$$\sum_{k=0}^{N} a_k\, y[n-k] = \sum_{k=0}^{M} b_k\, x[n-k].$$

  - ✓ 系数a_n，b_m是常数
- 如果等式两边序列是因果的，应用z变换（$X(z) = \sum_{-\infty}^{+\infty} x[n]z^{-n}$）方程的两边，得到

$$\sum_{k=0}^{N} a_k z^{-k}\, Y(z) = \sum_{k=0}^{M} b_k z^{-k}\, X(z).$$

- 传递函数为

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_M z^M + b_{M-1} z^{(M-1)} + \cdots + b_1 z^1 + b_0}{a_N z^N + a_{N-1} z^{(N-1)} + \cdots + a_1 z^1 + a_0}$$

  - ✓ 注意：离散系统的系统函数可能有两种形式，一种是分子和分母多项式均按Z的降幂次序排列；另一种是的分子多项式和分母多项式均按Z^{-1}的升幂次序排列

MATLAB中离散时间系统的描述方式

- **传递函数通常以有理形式给出，即以两个多项式的比值表示**

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_M z^M+b_{M-1}z^{(M-1)}+\cdots+b_1 z^1+b_0}{a_N z^N+a_{N-1}z^{(N-1)}+\cdots+a_1 z^1+a_0}$$

- **MATLAB通过tf定义离散时间系统**

```
H=tf(num,den, Ts)
```

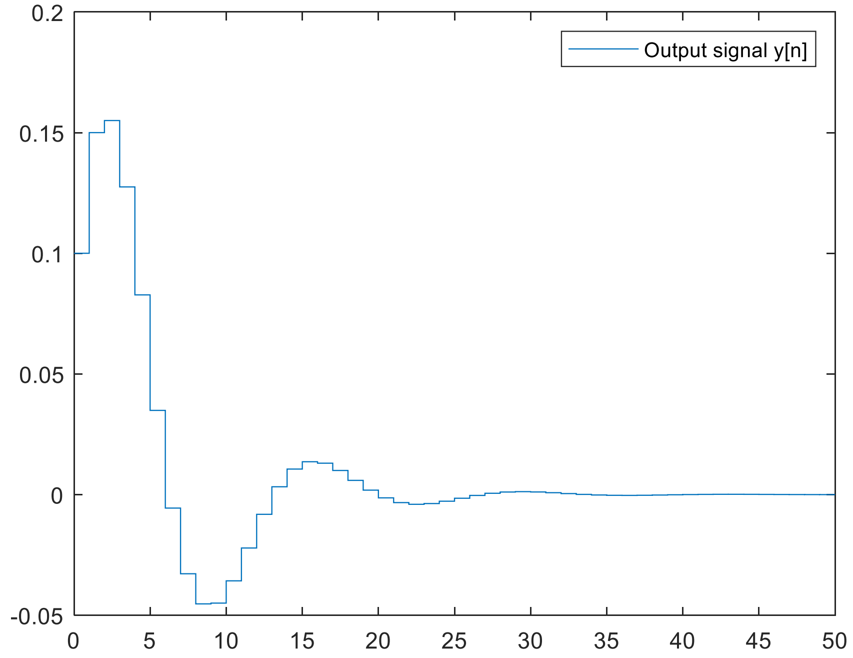
  - ✓ **其中num=**$[b_m,\cdots,b_1,b_0]$**和den=**$[a_n,\cdots,a_1,a_0]$**分别是分子和分母的多项式系数**
  - ✓ **Ts是采样时间**

- **例:**
  - ✓ num= [2 1];
  - ✓ den= [1 3 2];
  - ✓ Ts=0.5
  - ✓ Hs=tf(num,den,Ts)

```
  2 z + 1
-------------
z^2 + 3 z + 2

Sample time: 0.5 seconds
Discrete-time transfer function.
```

离散时间系统的时域响应

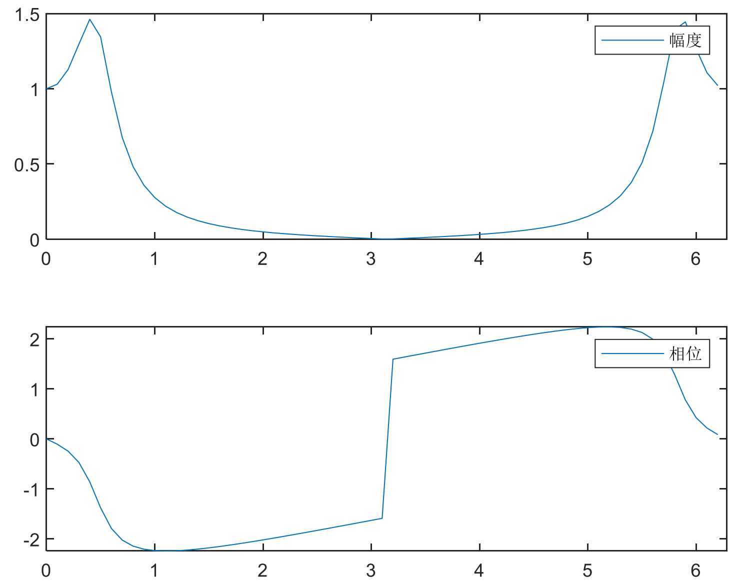
- 阶跃响应
  - ✓ 离散时间系统的阶跃响应是系统对单位阶跃序列的响应，阶跃序列为 $u[n]=\begin{cases}1, & n\geq 0\\ 0, & n<0\end{cases}$
  - ✓ 计算方式为 y=dstep(num,den) 或 y=stepz(num,den)
- 冲击响应
  - ✓ 离散时间系统的冲击响应是系统对单位脉冲序列的响应，单位脉冲序列为 $\delta[n]=\begin{cases}1, & n=0\\ 0, & n\neq 0\end{cases}$
  - ✓ 计算方式为 y=dimpulse (num,den) 或 y=impz (num,den)
- 对任意序列的响应，使用filter函数计算
  - ✓ num=[0.1 0.1];
  - ✓ den=[1 -1.5 0.7];
  - ✓ n=0:50;
  - ✓ x=(-1).^n;
  - ✓ y2=filter(num,den,x);
  - ✓ stairs(n,y2);
  - ✓ legend ('Output signal y[n] ')



离散时间系统的频域响应

- 在MATLAB中，可使用内置函数freqz计算并绘制
  - ✓ freqz(num,den,w),
  - ✓ w式频率矢量，num和den分别为传递函数的分子和分母系数矢量
  - ✓ 当输入为[H,w]=freqz(num,den,N)，则输出0~pi上均匀分布的N个频点的频率响应，若没有N，则默认输出512个频点。若希望输出0~2pi上的频率响应，可使用[H,w]=freqz(num,den,N,‘whole’)
- 例：例 绘制传递函数为 $H(z)=\dfrac{0.1z-0.1}{z^2-1.5z+0.7}$ 的频率响应

```
w =0:.1:2*pi;
num=[0.1 0.1];
den=[1 -1.5 0.7];
H=freqz(num,den,w);
subplot(211)
plot(w,abs(H))
xlim([0 2*pi]);legend('幅度');
subplot(212)
plot(w,angle(H))
xlim([0 2*pi]);legend('相位');
```



滤波器概述

- 在通信系统中，滤波器广泛用于从接收信号中分离出目标信号
  - ✓ 在掌握了目标信号的频域特征后，也就确定了所需滤波器的幅度和相位特性
- 滤波器设计，就是求解一个频率响应逼近需求值的因果滤波器的系数，通常包括三步：
  - ✓ 确定指标：设定滤波器特性的核心指标，这些指标需要根据应用确定。
  - ✓ 模型逼近：利用一些滤波器设计算法，逼近给定的指标。
  - ✓ 实现：以差分方程来实现得到的滤波器。
- 数字滤波器大致分为有限冲激响应滤波器（FIR）和无限冲激响应滤波器（IIR）
  - ✓ FIR滤波器可使通带内相位线性
  - ✓ 而IIR滤波器所需的抽头系数较少，从而降低了实现的复杂度
  - ✓ 由于通信系统通常对相位失真较敏感，因此实际使用的大多是FIR滤波器。

FIR滤波器

- **FIR滤波器的单位冲激响应是有限长的，假设** $h[n]=[b_0,\cdots,b_{N-1}]$ **在** $0\le n\le N-1$ **上有值，则其传递函数为**

$$H(z)=\sum_{n=0}^{N-1}h[n]z^{-n}$$

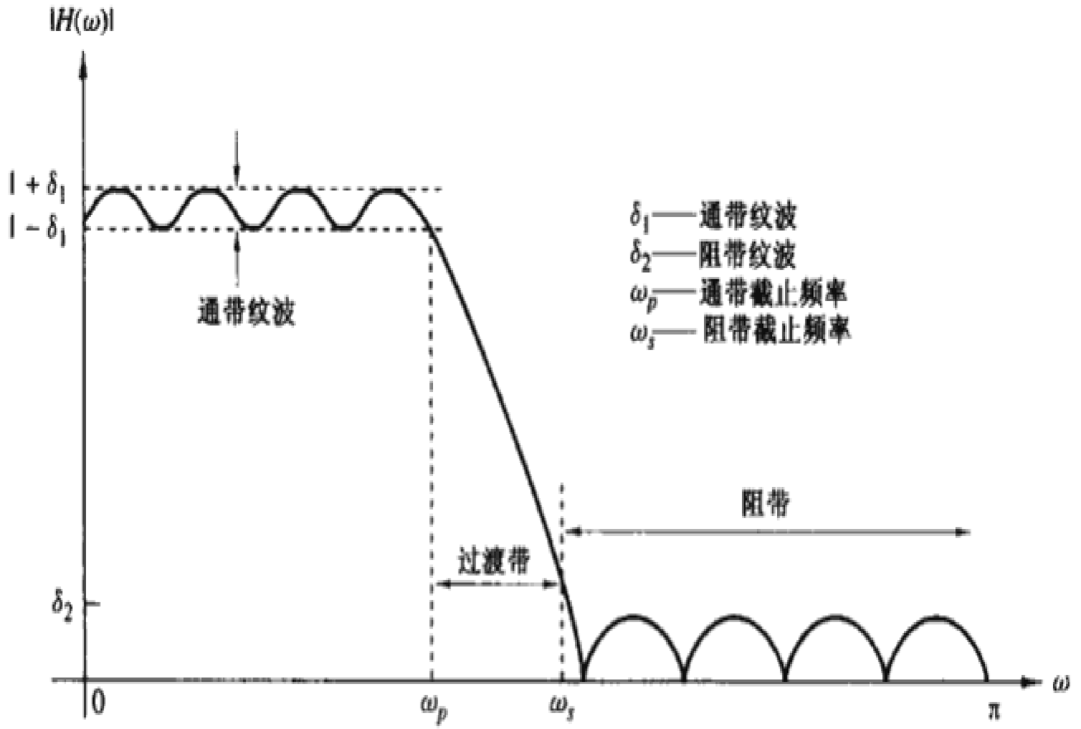
  - ✓ **其对应的差分方程为** $y[n]=b_0+b_1x[n-1]+\cdots+b_{N-1}x[n-N+1]$
  - ✓ **N-1阶FIR滤波器有N个存储器（也即滤波器长度或滤波器系数）**
- **由于只有零点、没有反馈环，FIR滤波器是稳定的**
- **当FIR系数满足奇对称或偶对称时，FIR滤波器可实现严格的线性相位特性**

$$h[n]=-h[N-1-n]\ (\text{奇对称})\qquad h[n]=h[N-1-n]\ (\text{偶对称})$$

- **线性相位 FIR 滤波器的相位延迟和群延迟在整个频带内相等且恒定**
  - ✓ **对于长度为N的线性相位 FIR 滤波器，群延迟为(N-1)/2 ，滤波后的信号延迟(N-1)/2个时间周期。**
  - ✓ **注意：群延迟可以是整数，也可以是分数（若要为整数，阶数为偶数，或滤波器长度为奇数）**

滤波器的主要指标

- **典型的低通滤波器通常包括4个指标**
  - ✓ **通带频率** $f_p = \frac{\omega_p}{2\pi}$ **，阻带频率** $f_s = \frac{\omega_s}{2\pi}$
  - ✓ **通带波动幅度** δ_1 **阻带波动幅度** δ_2
  - ✓ **通带波动幅度和阻带波动幅度实际上描述了实现的滤波器与设计目标频率特性之间的误差容限**
- **其他概念**
  - ✓ **滤波器带宽：** f_p
  - ✓ **过渡带：** $f_p \sim f_s$
  - ✓ **通带波纹：** $R_p = 20\lg\frac{1+\delta_1}{1-\delta_1}$
  - ✓ **阻带衰减：** $A_s = 20\lg\frac{1+\delta_1}{\delta_2}$



- **给定通带波纹和阻带衰减，可以计算通带和阻带波动幅度**

$$\delta_1 = \frac{1-10^{-\frac{R_p}{20}}}{1+10^{-\frac{R_p}{20}}}$$

$$\delta_2 = 1+\delta_1\ 10^{-\frac{A_s}{20}} \approx 10^{-\frac{A_s}{20}}$$

滤波器设计的窗函数法（1/2）

- 窗函数法是一种时域设计法，它以理想数字滤波器的设计为基础，用因果、有限长的冲击响应去逼近非因果无限长的理想滤波器
  - ✓ 可设计经典低通、高通、带通、带阻滤波器
  - ✓ 优点：窗函数法的设计过程比较简单
  - ✓ 缺点：但不能设计具有复杂频率响应特性的滤波器，包括窗函数形状不一定满足给定的特数频率响应指标，不能准确控制通带及阻带的截止频率，不能准确控制通带和阻带的波纹等。

- 常用窗函数

窗函数	过渡带宽（d_w/M）	最小阻带衰减(dB)	函数格式
矩形窗	$1.8\pi/\boldsymbol{M}$	21	wind=boxcar(M+1)
汉宁窗	$6.2\pi/M$	44	wind=hanning(M+1)
海明窗	$6.6\pi/M$	53	wind=hamming(M+1)
布莱克曼窗	$11\pi/M$	74	wind=blackman(M+1)
凯泽窗($\beta = 7.865$)	$10\pi/M$	80	wind=kaiser(M+1,beta)

滤波器设计的窗函数法（2/2）

- **步骤**
  - ✓ 根据阻带衰减选择窗函数
  - ✓ 据过渡带宽d_w与阶数之间的关系确定滤波器的阶数（选为偶数） $M = \left\lceil \frac{d_w}{f_s - f_p} \right\rceil$
  - ✓ 调用MATLAB函数，获得FIR滤波器系数
- **窗函数法设计FIR滤波器的内置函数fir1(n,Wn,Window)**
  - ✓ n是阶数
  - ✓ Wn是归一化截止角频率（ Wn=1对应于0.5fs，采样频率的一半，即奈奎斯特频率)
  - ✓ Window是窗函数的系数

示例：滤波器设计的窗函数法

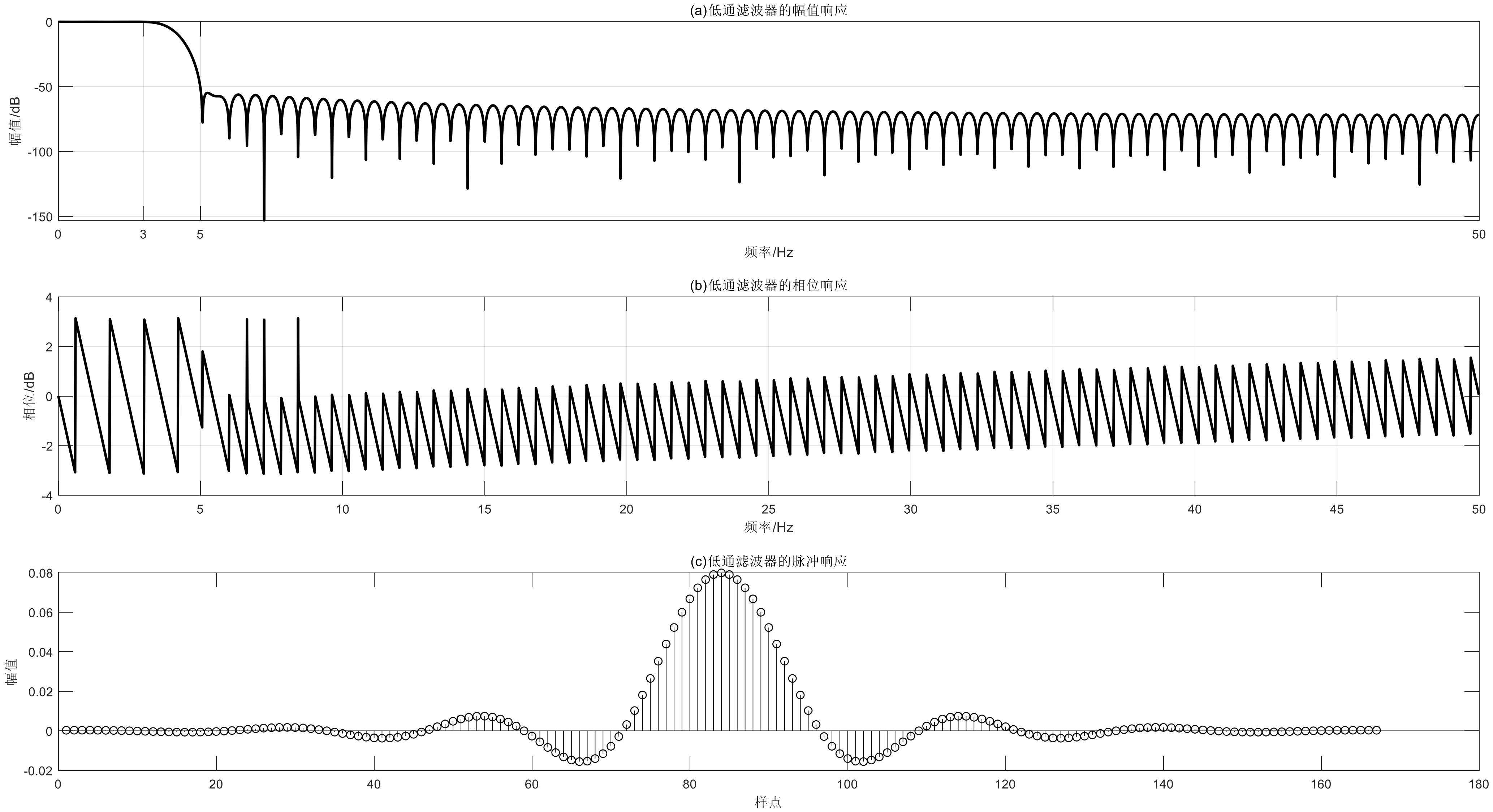
- 例：用窗函数法设计一个低通滤波器，采样频率为100Hz，通带频率fp=3Hz，阻带频率为fs=5Hz，通带波纹 Rp=3dB，阻带衰减为As=50dB
  - ✓ 由于As=50dB ，查表可知，选用海明窗。
  - ✓ 程序: simFIRWindFunc.m

```
Fs=100;                          % 采样频率
halfFs=Fs/2;                     % 奈奎斯特频率
fp=3; fs=5;                      % 通带和阻带频率
Rp=3; As=50;                     % 通带波纹和阻带衰减
wp = fp*pi/halfFs;               % 通带归一化角频率
ws = fs*pi/halfFs;               % 阻带归一化角频率
deltaw= ws - wp;                 % 过渡带宽 Δω 的计算
M = ceil(6.6*pi/ deltaw);        % 按海明窗计算所需的滤波器阶数 N
M = M + mod(M,2);                % 使滤波器阶数为偶数
wind = (hamming(M+1))';          % 海明窗计算
Wn=(fp+fs)/Fs;                   % 计算截止频率
b=fir1(M,Wn,wind);               % 用 fir1 函数设计 FIR 第 1 类滤波器
```

```
% 计算滤波器的频率响应
w=0:0.001:pi;
H=freqz(b,1,w);                   %滤波器响应
mag=abs(H);
db=20*log10((mag+eps)/max(mag));  %幅度的相对值
pha=angle(H);                     %相位谱
grd=grpdelay(b,1,w);              %群时延
```

示例：滤波器设计的窗函数法

- M=166, 群时延恒为83



滤波器设计的频率采样法

- 频率采样法是一种频域设计法，其基本思路是把目标滤波器频率特性进行等间隔采样，再采样值的傅里叶反变换作为滤波器系数
  - ✓ 优点：设计比较简单，适用于分段常数的滤波器设计，特别是窄带滤波器的设计
  - ✓ 缺点：滤波器边界频率不易控制，不能控制通带波纹等。
- 频率采样法设计FIR滤波器的内置函数fir2(n, f, m, Window)
  - ✓ n是阶数，f和m是频率和滤波器幅值矢量
  - ✓ Window是窗函数的系数（默认是海明窗)
- 设计方法
  - ✓ 当设计低通滤波器时，f=[fp fs], m=[1 0]
  - ✓ 当设计带通滤波器时，f=[fs1 fp1 fp2 fs2], m=[0 1 1 0]
  - ✓ 对高通滤波器可参照低通滤波器的设计给出f和m
  - ✓ 注意，f是归一化的频率，f=1对应于0.5fs（采样频率的一半，也即奈奎斯特频率）。
  - ✓ 此外，f和m的值不止限于低通、高通、带通和带阻滤波器，也可以表示出任意的幅值响应。

示例：滤波器设计的频率采样法

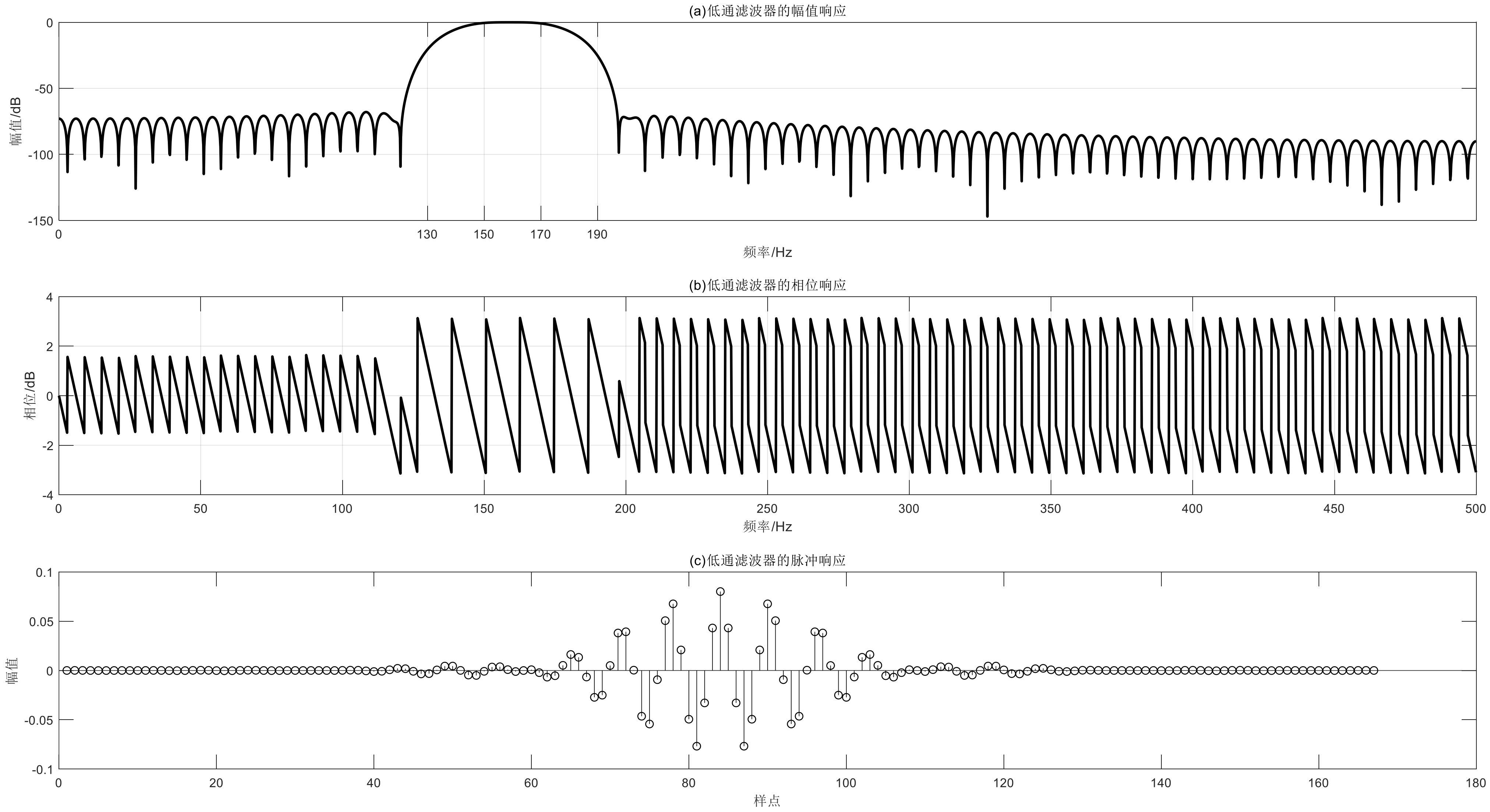
- 例：使用频率采样法设计一个带通滤波器，采样频率1000Hz，通带为fp1=150Hz，fp2=170Hz,fs1=130Hz,fs2=190Hz 。
  - ✓ 程序：simFIRFreqSampling.m

```
Fs=1000;                              % 采样频率
halfFs=Fs/2;                          % 奈奎斯特频率
fp1=150; fp2=170;                     % 通带频率
fs1=130; fs2=190;                     % 阻带频率
deltaw=(fp1-fs1)*pi/halfFs;           % 过渡带宽 Δω
M = ceil(6.6*pi/ deltaw);             % 按海明窗计算所需的滤波器阶数 N
M = M + mod(M,2);                     % 使滤波器阶数为偶数
f=[0 fs1 fp1 fp2 fs2 halfFs]/halfFs; % 通带和阻带的频率点归化值
m=[0 0 1 1 0 0];                      % 通带和阻带的频率点归化值

wind = (hamming(M+1))';               % 海明窗计算
b=fir2(M,f,m,wind);                    % 用 fir2函数设计 FIR 第 1 类滤波器
```

```
% 计算滤波器的频率响应
w=0:0.001:pi;
H=freqz(b,1,w);                       %滤波器响应
mag=abs(H);
db=20*log10((mag+eps)/max(mag));      %幅度的相对值
pha=angle(H);                         %相位谱
grd=grpdelay(b,1,w);                  %群时延
```

示例：滤波器设计的频率采样法

- M=166, 群时延恒为83



滤波器设计的最优等波纹法

- 最优等波纹法是一种频域设计法，其设计思想是使滤波器幅度特性在通带和阻带上的误差峰值均匀分布，从而把波动的幅度控制到最小
  - ✓ 该方法使用的算法为Parks-McClellan算法。
- MATLAB提供了根据赋值偏差指标计算滤波器阶数的内置函数firpmord和使用最优等波纹法设计FIR滤波器的内置函数firpm。
  - ✓ [n,fo,ao,w]=firpmord(f,a,dev,fs)
    - ➢ 输入参数：f和a是频率和滤波器赋值矢量，dev是最大容忍偏差，fs是采样频率；
    - ➢ 输出参数：n是滤波器阶数，fo和ao是频率和滤波器赋值矢量，w是权重矢量。这四个参数即firpm调用时n,f,a,w的参数。
  - ✓ b=firpm (n,f,a,w)
    - ➢ 输入参数：n是滤波器阶数，f和a是频率和滤波器赋值矢量w是权重矢量，用于调整频点上幅度的吻合度；
    - ➢ 输出参数：b是FIR滤波器的抽头系数。

示例：滤波器设计的最优等波纹法

- 例 设计一个等波纹的低通滤波器，采样频率为2000Hz，通带频率fp=200Hz，阻带频率为fs=300Hz，通带波纹Rp=2dB，阻带衰减为As=40dB
  - ✓ 程序：simFIROptimal.m

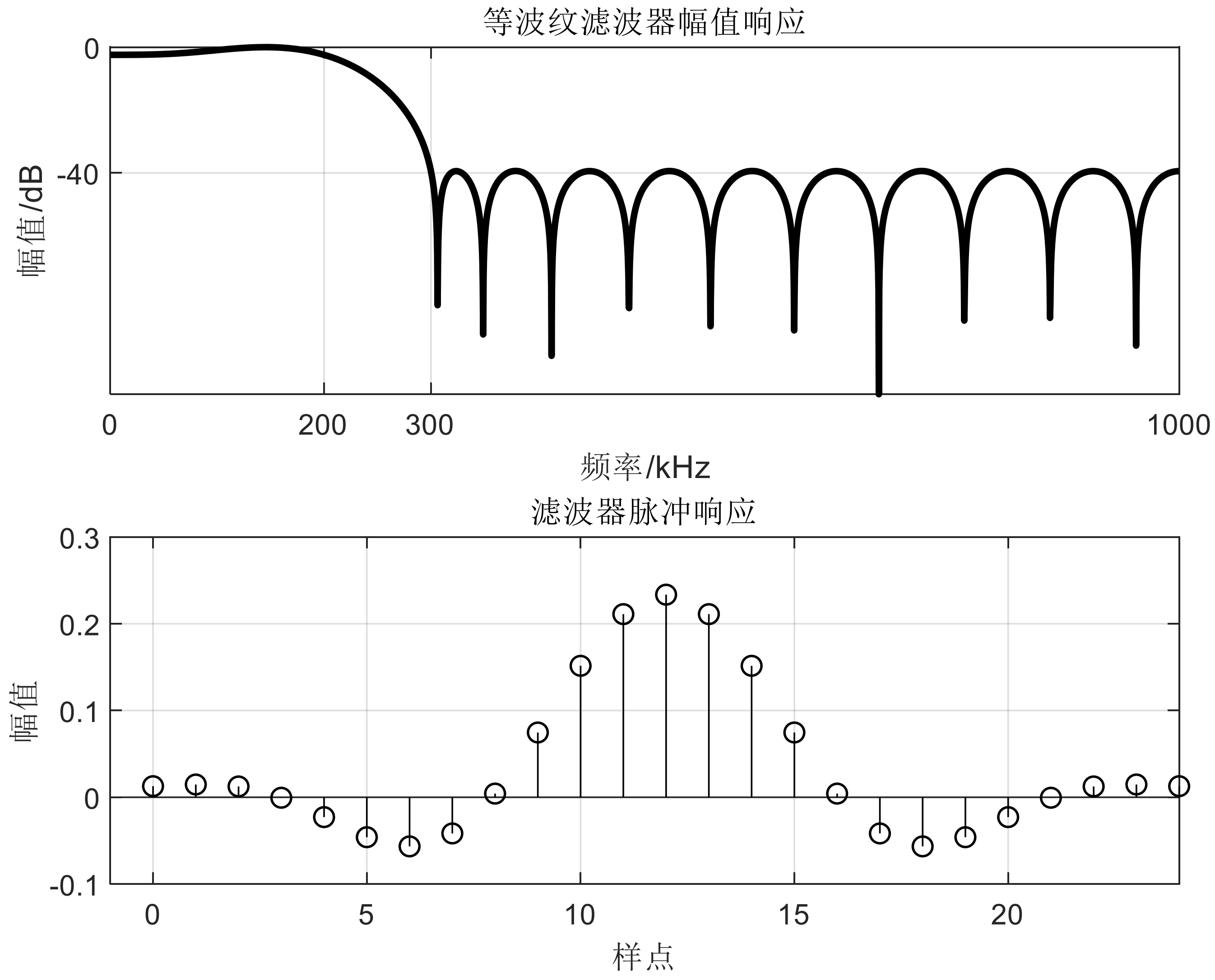
```
Fs=2000;                           % 采样频率
halfFs=Fs/2;                       % 奈奎斯特频率
fp=200; fs=300;                    % 通带和阻带频率
wp=fp*pi/halfFs; ws=fs*pi/halfFs;  % 通带和阻带归一化角频率
Rp=2; As=40;                       % 通带波纹和阻带衰减
F=[wp ws]/pi;                      % 理想滤波器的频率矢量
A=[1,0];                           % 理想滤波器的幅值矢量
```

```
% 通带波纹和阻带衰减线性值
dev1=(1-10^(-Rp/20))/(1+10^(-Rp/20)); dev2=10^(-As/20); %通带和阻带偏差
dev=[dev1,dev2];                   % 与理想滤波器的偏差的矢量

[N,F0,A0,W]=firpmord(F,A,dev);     % 调用 firpmord 函数计算参数
N=N+mod(N,2);                      % 保证滤波器阶数为偶数
b=firpm(N,F0,A0,W);                % 用 firpm 函数设计滤波器
```

```
% 计算滤波器频域响应
w=0:0.001:pi;
H=freqz(b,1,w);                    %滤波器响应
mag=abs(H);
db=20*log10((mag+eps)/max(mag));   %幅度的相对值
pha=angle(H);                      %相位谱
grd=grpdelay(b,1,w);               %群时延
```

示例：滤波器设计的最优等波纹法

- N=24，群时延恒为12



课后作业

- **用FFT来计算下列信号的频谱**
  - ✓ $x_a(t) = e^{-0.01t}\sin 2t - 3e^{-0.02t}\cos 5t$
  - ✓ $x_b(t) = e^{-0.01t}\sin 3t - 3e^{-0.02t}\cos 3.05t$
- **有一个信号$x(t) = \sin 2\pi f_1 t + \sin 2\pi f_2 t$，其中$f_1 = 80$，$f_2 = 150$。使用$f_s = 1500\text{Hz}$对$x(t)$信号进行采样，得到一串样点。请分别设计低通和带通滤波器，从$x(t)$信号中滤出低频和高频信号。（低通滤波器请使用窗函数法，带通滤波器使用频率采样法）**
  - ✓ 程序：hw2.m
  - ✓ Fig1： $x(t)$信号波形和频谱图
  - ✓ Fig2：原低频信号波形、低通滤波器的幅频响应、滤波后得到的低频信号波形
  - ✓ Fig3：原高频信号波形、带通滤波器的幅频响应、滤波后得到的高频信号波形

有问题，随便问！