



# MATLAB与信号处理

数字信号处理



# 4、基于MATLAB的数字信号处理

- 4.1 信号产生
- 4.2 离散系统的MATLAB实现
- 4.3 变换的MATLAB实现
- 4.4 数字滤波器设计
- 4.5 功率谱估计

# 4.1 信号产生

#### • 信号的表示

 x(n)={x(n)}={...,x(-2),x(-1),x(0),x(1),x(2) ...}

 Matlab中,信号都是用矩阵(向量)描述

 若不需要位置信息:x=[4 3 7 -9 1];

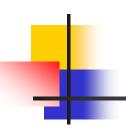
 若需要位置信息:n=[-2,-1,0,1,2];x=[4 3 7 -9 1];

#### • 典型离散信号表示

> 单位抽样序列

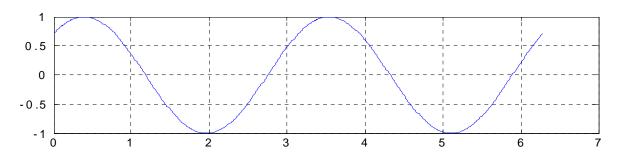
$$x=zeros(1,N);$$
  
 $x(1)=1;$   $\delta$  (n)  
 $x(k)=1;$   $\delta$  (n-k)

▶ 单位阶跃序列: x=ones(1,N);



# > 波形的产生

首先产生时间:如t=0:0.01:2\*pi; 产生信号:如 y=sin(2\*t+pi/4);



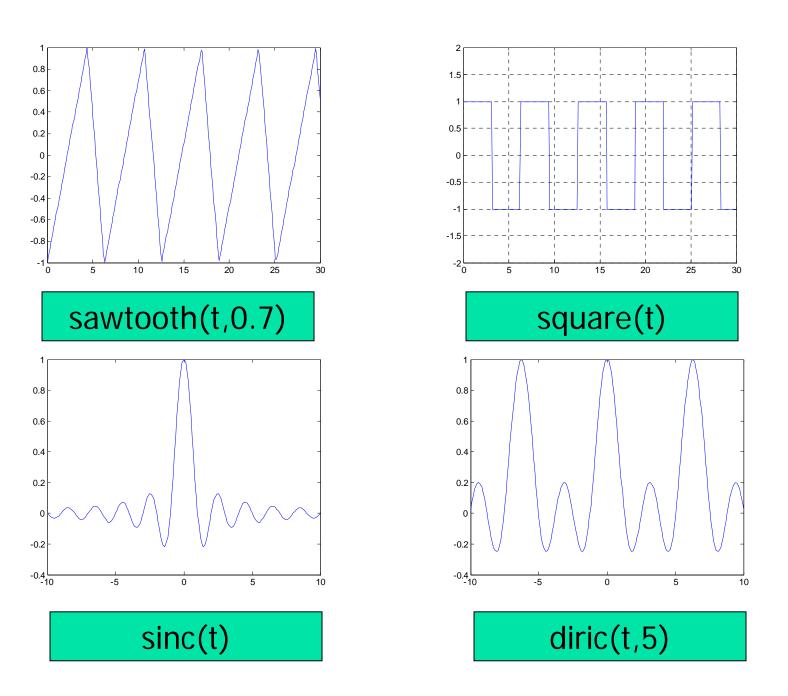
square sawtooth sinc

➤ 噪声的产生 rand, randn



# 周期波形产生

函数名	功能
Sawtooth(t,width)	产生锯齿波或三角波. Width确定最大值位置
Square(t,duty)	产生方波。Duty指定正半周期的比例
Sinc(t)	产生sinc或sin(pi*t)/(pi*t)
Diric(t,n)	产生Dirichlet函数





• 信号的常规操作

信号加

x=x1+x2

信号必须 等长对齐

信号乘

x = x1.\*x2

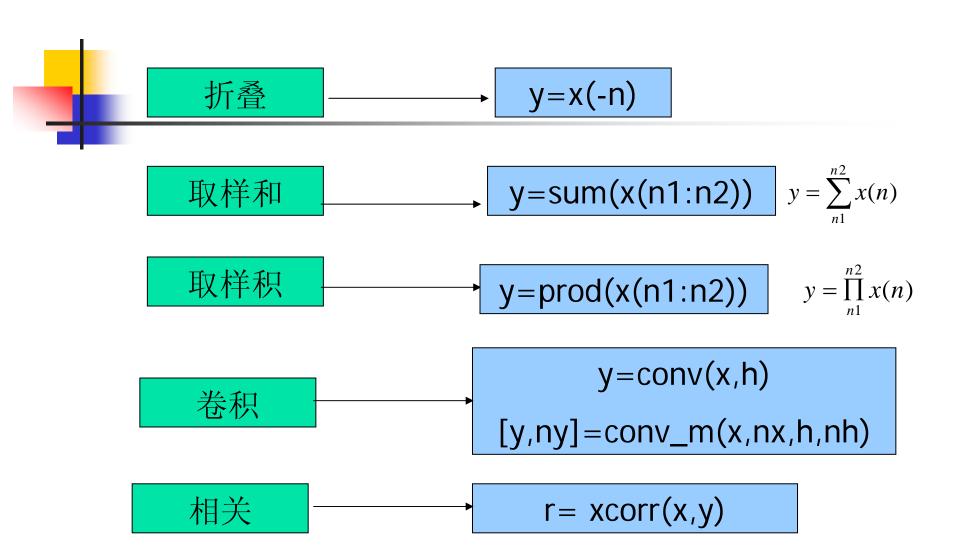
信号必须 等长对齐

改变比例

x=a\*x1

移位

x=x1(n-k)





# • 产生数据

- ▶直接键盘输入和程序中赋值
- ▶ 利用函数产生,如sin,cos,sawtooh等
- > 用Matlab从.Mat文件导入
- ▶ 通过文件读写产生数据,如fopen,fread等
- ▶ 直接利用函数读入数据(wavrecord)



# 4.2 离散系统的MATLAB实现

# 离散系统的表示方法:

#### LSI系统的时域表示

常系数线性差分方程 单位抽样响应h(n)

#### LSI系统的频域表示

频率响应 (DFT)

转移函数 (z变换)

零极点增益

二次分式

#### 离散系统的内部描述



# • 离散系统的输入输出的差分方程:

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{r=0}^{M} b_r x(n-r)$$

$$y(n) - 0.4y(n-1) - 0.5y(n-2) = 0.2x(n) + 0.1x(n-1)$$

• 单位抽样响应 h(n)?

filter函数

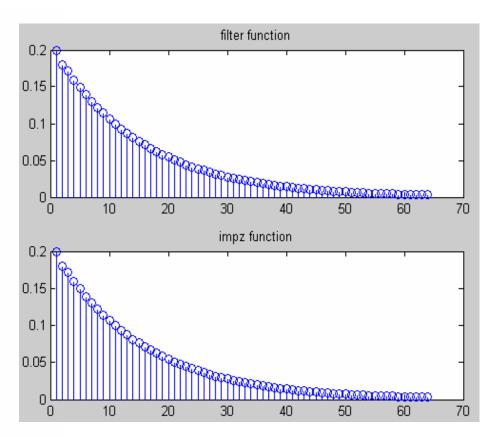
y=filter(b,a,x)

滤波函数 (系统实现)

impz函数

y=impz(b,a)

```
pul=[1 zeros(1,63)];
b=[0.2 \ 0.1];
a=[1 -0.4 -0.5];
h=filter(b, a, pul);
h1=impz(b, a, 64);
subplot(2,1,1);
stem(h):
title ('filter function');
subplot(2,1,2);
stem(h1):
title('impz function');
```



4

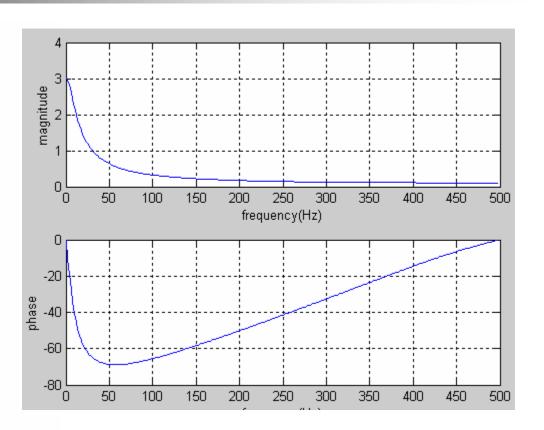
频率响应 H(e<sup>jw</sup>) ?

$$H(e^{jw}) = \frac{\sum_{r=0}^{M} b(r)e^{-jwr}}{1 + \sum_{k=1}^{N} a(k)e^{-jwk}}$$

[h,f]=freqz(b,a,n,fs)



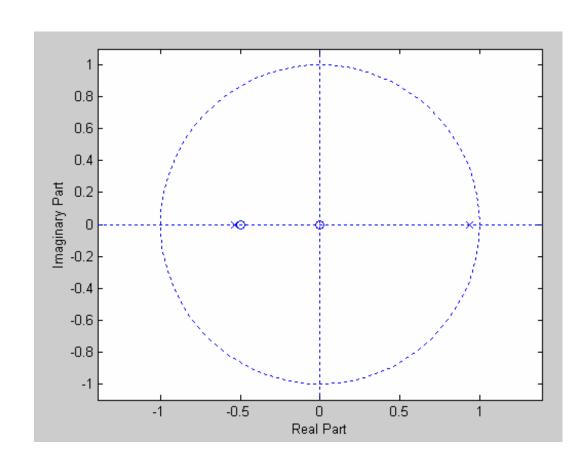
```
fs=1000;
b=[0.2 0.1];
a=[1 -0.4 -0.5];
[h, f]=freqz(b, a, 256, fs);
mag=abs(h);
ph=angle(h);
ph=ph*180/pi;
subplot(2,1,1);
plot(f, mag); grid;
xlabel('frequency(Hz)');
ylabel('magnitude');
subplot(2,1,2);
plot(f, ph); grid;
xlabel('frequency(Hz)');
ylabel('phase');
```



### • 零极点增益?

```
b=[0.2 0.1];
a=[1 -0.4 -0.5];
zr=roots(b)
pk=roots(a)
g=b(1)/a(1)
zplane(b, a)
    -0.5000
_{\mathbf{P}}\mathbf{k}=
      0.9348
    -0.5348
 E
```

0.2000





# • 离散系统的表示方法转换函数

tf2zp sos2tf

tf2ss sos2zp

zp2tf sos2ss

zp2sos ss2tf

zp2ss ss2zp

ss2sos



# 4.3 变换的MATLAB实现

#### 1、DFT

# 4

# 计算 $X(n) = \cos(\frac{n\pi}{6}), N = 12$ DFT并画图

```
N=12:
n=0:N-1:
xn=cos(pi*n/6);
k=0: N-1:
WN=exp(-j*2*pi/N);
nk=n'*k:
WNnk=WN. nk:
XK=xn*WNnk
figure(1):
stem(n, xn);
figure(2);
stem(k, abs(XK))
```

#### KΚ =

Columns 1 through 5

-0.0000

6.0000 + 0.0000i -0.0000 - 0.0000i -0.0000 - 0.0000i -0.0000 - 0.0000i

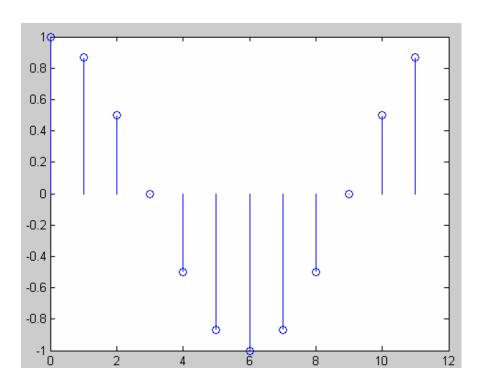
Columns 6 through 10

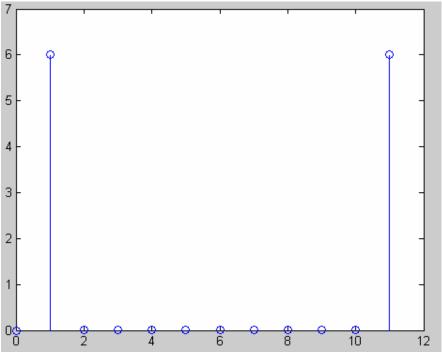
-0.0000 - 0.0000i -0.0000 - 0.0000i 0.0000 - 0.0000i 0.0000 - 0.0000i 0.0000i 0.0000 - 0.0000i

Columns 11 through 12

0.0000 - 0.0000i 6.0000 + 0.0000i









#### 2、FFT

可利用内部函数fft进行计算,速度快。 y=fft(x,n) x的n点fft

$$x = ifft(y,n)$$



# 3、z变换

$$X (z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$

$$\chi(n) = \frac{1}{2\pi i} \oint_C \chi(z) z^{n-1} dz$$
 f=iztrans(F)

# 4

# 求f(n)=sin(ak)u(k)的Z变换

```
f=sym('sin(a*k)');
F=ztrans(f)
f1=iztrans(F)
结果:
    z*sin(a)/(z^2-2*z*cos(a)+1)
   sin(a*n)
```

$$X(z) = A_0 + \sum_{i=1}^{N} \frac{A_i}{1 - p_i z^{-1}}$$

$$X(z) = A_0 + \sum_{i=1}^{N} \frac{A_i}{1 - p_i z^{-1}}$$
 [R,P,K]=residuez(b,a);

$$R =$$

$$-0.0625$$

$$b=1$$
;

$$a=[1 \ 0 \ -0.4000 \ 0 \ 0.0144];$$

$$P =$$

$$X(z) = \frac{0.5625}{1 - 0.6z^{-1}} + \frac{0.5625}{1 + 0.6z^{-1}} - \frac{0.0625}{1 - 0.2z^{-1}} - \frac{0.0625}{1 + 0.2z^{-1}}$$

$$-0.6000$$

$$x(n) = [0.5625(0.6)^n + 0.5625(-0.6)^n - 0.0625(0.2)^n - 0.0625(-0.2)^n]u(n)$$

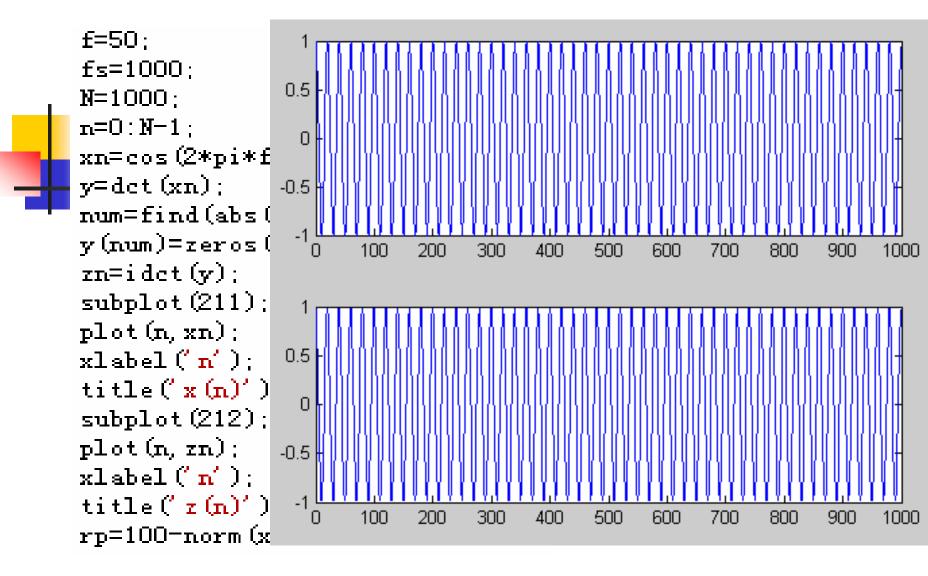
$$K =$$

# 4、DCT

$$X_{c}(0) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X(n)$$

$$X_{c}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} X(n) \cos \frac{(2n+1)k\pi}{2N}$$

- y = dct(x,n)
- x=idct(y,n)



rp=84.3566



# 5、Hilbert变换

· 解析信号

$$Z(n) = X(n) + j x(n)$$

- DFT方法求解
- 函数y=Hilbert(x)



- Function [zn,hxn]=yhilbert(xn)
- **•** {%...
- N=length(xn);
- Xk=fft(xn);
- K1=0;
- Zk(k1+1)=Xk(k1+1);
- K2=1:N/2-1;
- Zk(k2+1)=2\*Xk(k2+1);
- K3=N/2:N-1;
- Zk(k3+1)=zeros(size(k3));
- Zn=ifft(Zk);
- Hxn=-j\*(Zn-Xn);
- }



# 4.4 基于MATLAB的数字滤波器设计

- (1) 数字滤波器的基本概念
- (2) 基于MATLAB的IIR滤波器设计

(3) 基于MATLAB的FIR滤波器设计



# (1) 数字滤波器的基本概念

数字滤波器对数字信号进行滤波的线性时不变系统。

# • 数字滤波器的分类

IIR数字滤波器:设计经典选频滤波器,性能好,阶数低。

FIR数字滤波器:设计线性相位经典滤波器,适用范围广、

阶数高。



### • 数字滤波器的设计步骤

- 1) 按照实际任务的要求,确定滤波器的技术指标。
- 2)用一个因果、稳定的离散线性时不变系统的系统函数去逼近这一网络性能。根据不同的要求可选择IIR或者FIR系统函数去逼近。
- 3)利用有限精度算法实现系统函数。



# • IIR数字滤波器一般有两种设计方法:

- 模拟原型法
   先设计一个合适的模拟滤波器,然后再变 换为满足要求的数字滤波器。
- 2)直接法直接在离散时域和离散频域进行数字滤波器的设计



• FIR一般有三种设计方法:

窗函数法 频率取样法 切比雪夫逼近法



# (2) 基于MATLAB的IIR滤波器设计

Butterworth数字滤波器设计

- 椭圆数字滤波器设计
- Chebyshev-I数字滤波器设计
- Chebyshev-II数字滤波器设计



#### Butterworth数字滤波器设计

#### MATLAB语句:

$$[b,a]$$
=butter $(n,wn)$  (1)

$$[b,a]=butter(n,wn,'ftype')$$
 (2)

$$[z,p,k] = butter(...)$$
 (3)

$$[a,b,c,d] = butter(...)$$
 (4)

语句(1)~(4)都可以设计一个阶数为n,截止频率为wn的数字低通滤波器。对于语句(2),若参数ftype取'high'或'stop',则可以来设计相应的高通或带阻滤波器。返回值a和b为系统函数的分子和分母的系数。



系统函数为:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \dots + a(n+1)z^{-n}}$$

返回值z,p,k分别是滤波器的零点、极点和增益。

返回值A、B、C、D构造滤波器的状态方程。状态方程为:

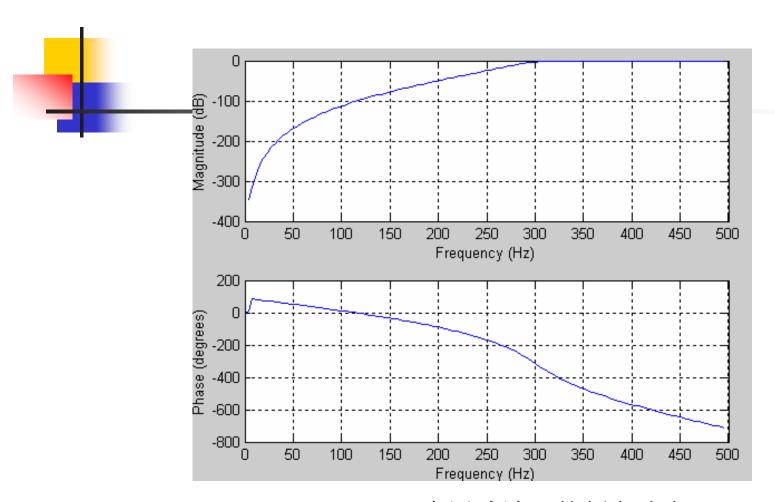
$$x(n+1) = Ax(n) + Bu(n)$$
$$y(n) = Cx(n) + Du(n)$$

其中u是输入信号,x是状态变量,y是输出信号。



例:设计一个阶数为9阶、截止频率为300Hz的高通 Butterworth滤波器,采样频率为1000。并画出它的频 率特性。

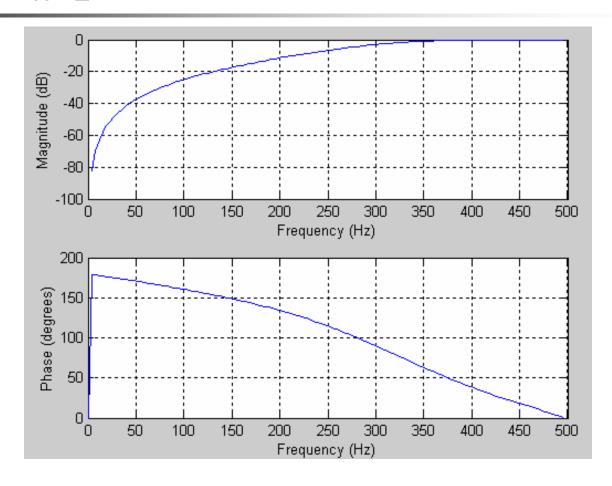
```
程序: n=9;
wn=300/500;
[b,a]=buffer(n,wn,'high');
freqz(b,a,128,1000);
```



Buttordworth高通滤波器的频率响应

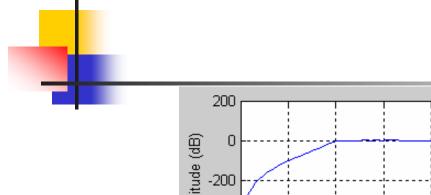


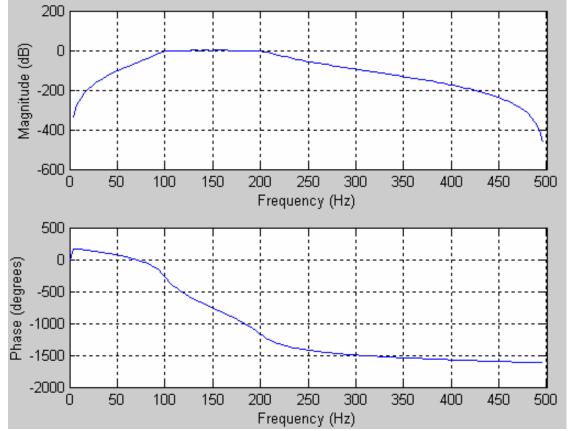
### n=2



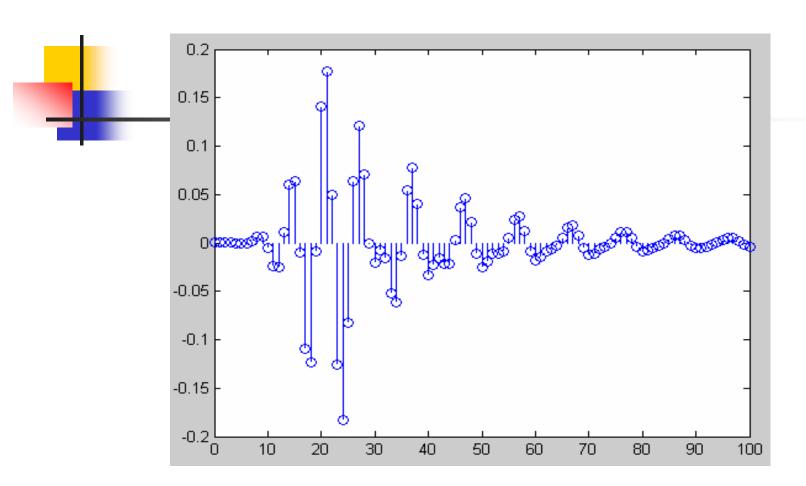
例 设计一个10阶的带通Butterworth滤波器,它的通带范围是100~200Hz,采样频率为1000。并画出它的频率特性和冲击响应。

```
程序: n=10;
wn=[100 200]/500;
[b,a]=butter(n,wn);
freqz(b,a,128,1000);
[y,t]=impz(b,a,101);
stem(t,y);
```





Buttordworth带通滤波器的频率响应



Buttordworth滤波器的时域冲击响应



### • 椭圆数字滤波器设计

### MATLAB语法:

$$[b,a] = ellip(n,Rp,Rs,wn)$$
 (5)

$$[z,p,k] = ellip(...)$$
 (7)

$$[a,b,c,d] = ellip(...)$$
 (8)

语句(5)~(8)都可以设计一个阶数为n,截止频率为wn,通带波纹最大衰减为Rp、阻带波纹最小衰减为Rs的数字低通滤波器。对于语句(6),若参数ftype取'high'或'stop',则可以来设计相应的高通或带阻滤波器。

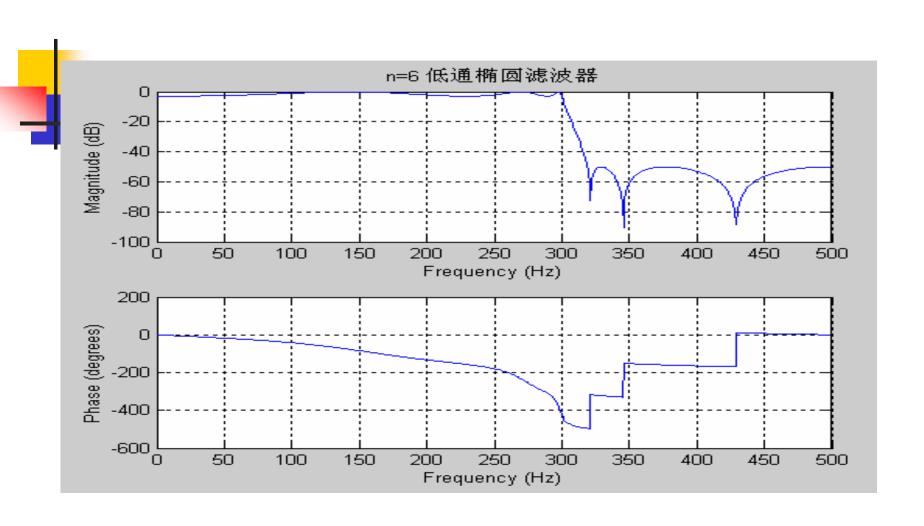
返回值a和b为系统函数的分子和分母的系数。返回



值z,p,k分别是滤波器的零点、极点和增益。返回值A、B、C、D可以用来构造滤波器的状态方程。

例 采样频率为1000Hz,请设计一个阶数为6阶、截止频率为300Hz的低通ellip数字滤波器。其中滤波器在通带的波纹为3dB,阻带的波纹为50dB。

程序: [b,a]=ellip(6,3,50,300/500); freqz(b,a,512,1000); title('n=6 低通椭圆滤波器');



椭圆低通滤波器的频率响应

例设计一个10阶的带通滤波器,通带范围为100~200Hz,

信号的采样频率为1000Hz,滤波器在通带的波纹为0.5 dB,阻带的波纹为20dB。设计出这个滤波器并画出它的冲击响应。

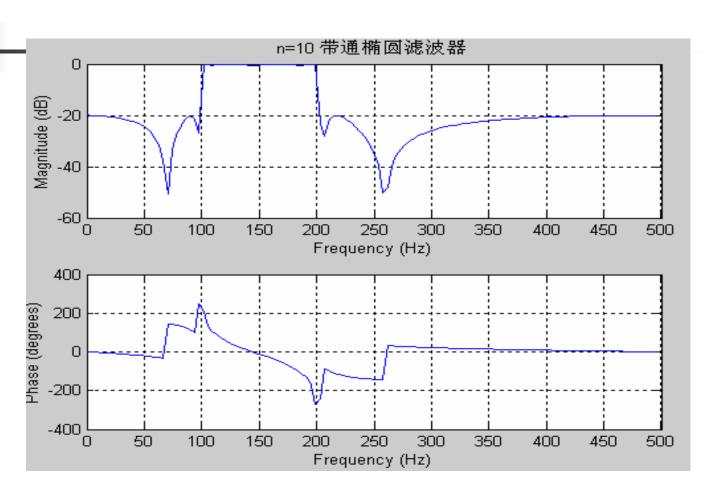
```
程序: [b,a]=ellip(10,0.5,20,[100 200]/500);

[y,t]=impz(b,a,101);

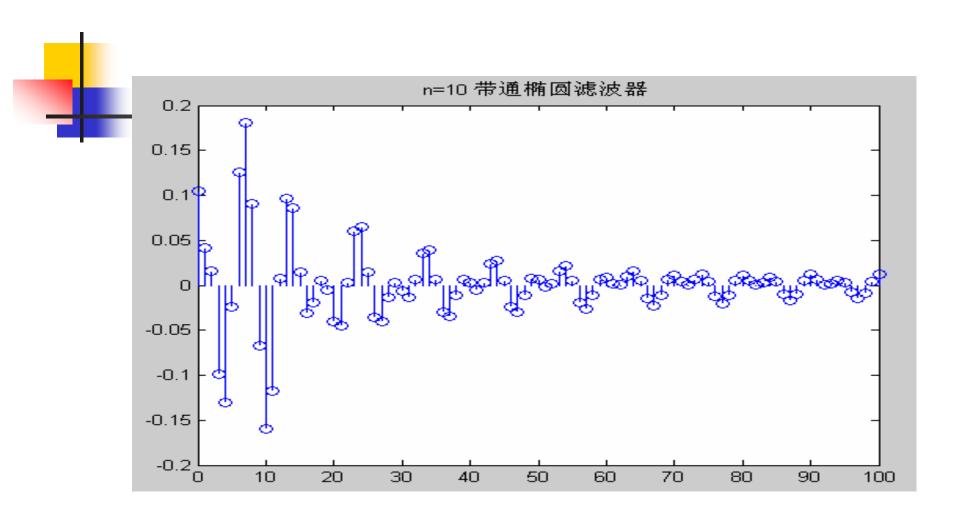
freqz(b,a,128,1000);

stem(t,y);

title('n=10 带通椭圆滤波器');
```



椭圆带通滤波器的频率响应



椭圆滤波器的冲击响应



## • Chebyshev-I数字滤波器设计

### MATLAB语法:

$$[b,a] = cheby1(n,Rp,wn)$$
 (9)

$$[b,a] = cheby1(n,Rp,wn,'ftype')$$
 (10)

$$[z,p,k] = cheby1(...)$$
 (11)

$$[a,b,c,d] = cheby1(...)$$
 (12)

语句(9)~(12)都可以设计一个阶数为n,截止频率为wn,通带波纹最大衰减为Rp的数字低通滤波器。对于语句(10),若参数ftype取'high'或'stop',则可以来设计相应的高通或带阻滤波器。

返回值a和b为系统函数的分子和分母的系数。返回

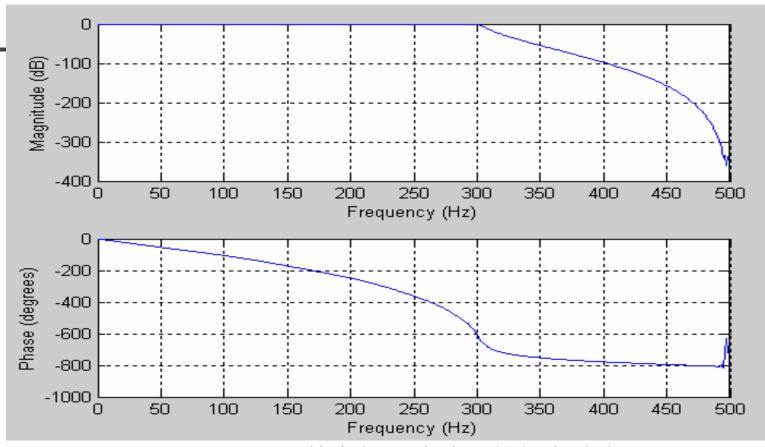


值z,p,k分别是滤波器的零点、极点和增益。返回值A、B、C、D可以用来构造滤波器的状态方程。

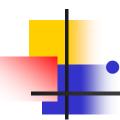
例 对采样频率为1000Hz的采样信号,设计一个阶数为9 阶,截止频率为300Hz的低通cheby1数字滤波器,其中滤波器在通带的波纹为0.5dB。

程序: [b,a]=cheby1(9,0.5,300/500); freqz(b,a,512,1000);





Chebyshev-I数字低通滤波器的频率响应



## Chebyshev-II数字滤波器设计

### MATLAB语法:

$$[b,a] = cheby2(n,Rs,wn)$$
 (13)

$$[b,a] = cheby2(n,Rs,wn,'ftype')$$
 (14)

$$[z,p,k] = cheby2(...)$$
 (15)

$$[a,b,c,d] = cheby2(...)$$
 (16)

语句(13)~(16)都可以设计一个阶数为n,截止频率为wn,阻带纹波最小衰减为Rs的数字低通滤波器。对于语句(13)式,若参数ftype取'high'或'stop',则可以来设计相应的高通或带阻滤波器。

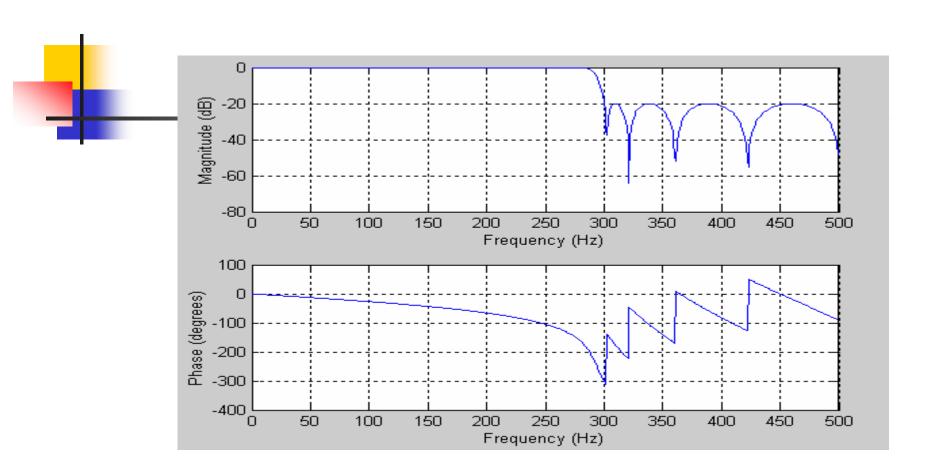
返回值a和b为系统函数的分子和分母的系数。返回



值z,p,k分别是滤波器的零点、极点和增益。返回值A、B、C、D可以用来构造滤波器的状态方程。

例 对于采样频率为1000Hz的信号,设计一个阶数为9 阶,截止频率为300Hz的低通Chebyshev-II数字滤波器,其中滤波器在阻带的波纹为20dB。

程序: [b,a]=cheby2(9,20,300/500); freqz(b,a,512,1000);



Chebyshev-II数字低通滤波器的频率响应

## (3) 基于MATLAB的FIR滤波器设计

- 窗函数法设计FIR数字滤波器
- 频率取样法
- 切比雪夫逼近法设计FIR滤波器



### • 窗函数法设计FIR数字滤波器

- 时域中进行的
- 常见的窗函数

w=boxcar(n)

w=triang(n)

w=hanning(n)

w=hamming(n)

w=blackman(n)

w=chebwin(n,R)

w=bartlett(n)

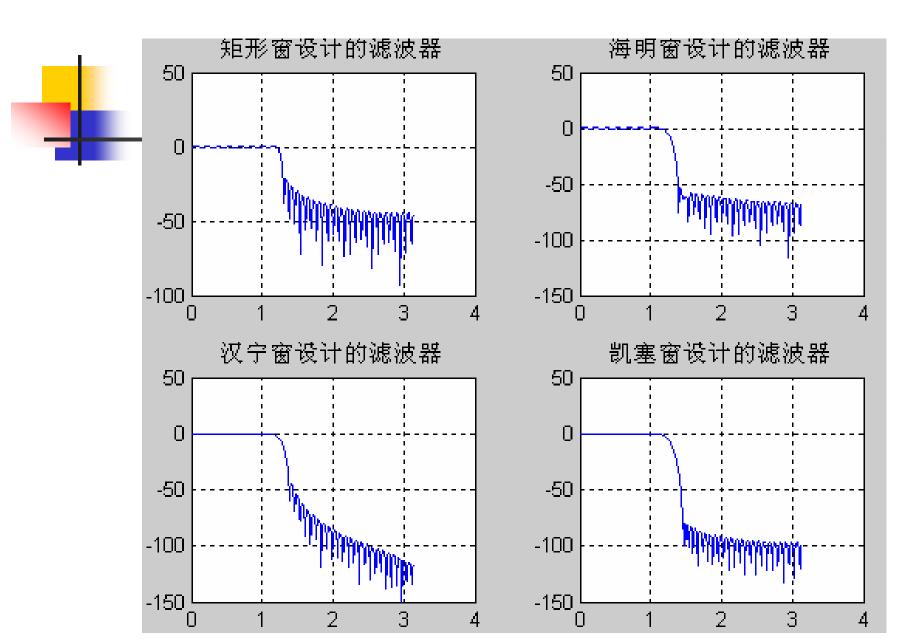
 $w=kaiser(n,\beta)$ 



例:用8种窗函数设计数字低通滤波器。采样频率 1000Hz,截止频率200Hz,滤波器阶数81。

### 程序:

```
passrad=0.4*pi;
w=boxcar(81);
n=1:1:81;
hd=sin(passrad*(n-41))./(pi*(n-41));
hd(41)=passrad/pi;
h=hd.*rot90(w);
[mag,rad]=freqz(h);
plot(rad,20*log10(abs(mag)));
grid;
```





## > 利用fir1函数设计滤波器(窗函数法)

### MATLAB语法:

b=fir1(n, wn)

b=fir1(n, wn, 'ftype')

b=fir1(n, wn, window)

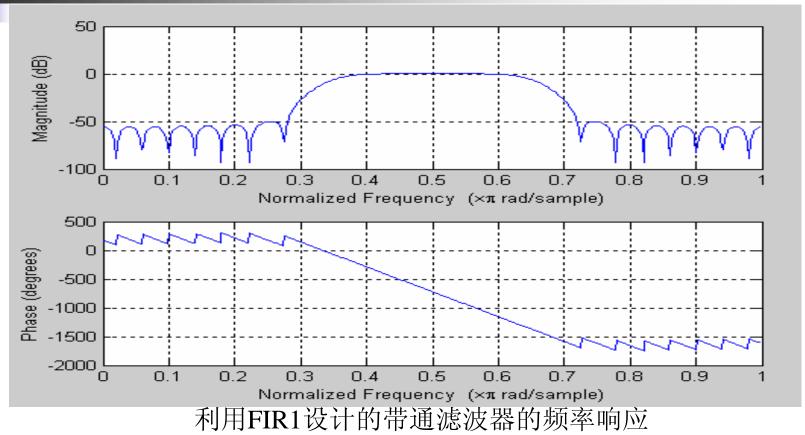
b=fir1(n, wn, 'ftype', window)



例:设计一个阶数为48,通带范围为0.35~0.65的带通FIR线性相位滤波器,并分析它的频率特性。

程序: b=fir1(48,[0.35,0.65]); freqz(b);







## 利用kaiserord函数求凯赛窗函数的参数

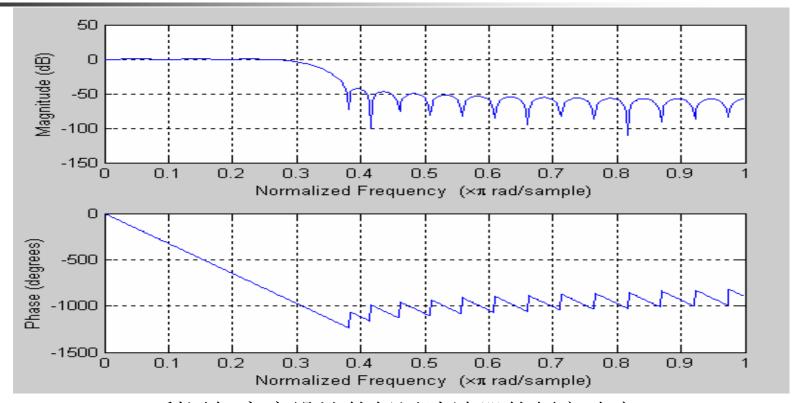
### MATLAB语法:

[n,wn,beta,ftype]=kaiserord(f,a,dev) [n,wn,beta,ftype]=kaiserord(f,a,dev,Fs)

例 利用凯赛窗函数设计一个低通FIR数字滤波器,通带范围是0~1000Hz,阻带范围是1500~4000Hz,通带的波纹最大为0.05,阻带的波纹最大为0.01。采样频率为8000Hz。

```
程序: fsamp=8000;
fcuts=[1000 1500];
mags=[1 0];
devs=[0.05 0.01];
[n,wn,beta,ftype]=kaiserord(fcuts,mags, devs,fsamp);
b=fir1(n,wn,ftype,kaiser(n+1,beta));
freqz(b);
```





利用凯赛窗设计的低通滤波器的频率响应



## ▶ 利用fir2函数设计任意响应FIR数字滤波器

### MATLAB语法:

b=fir2(n, f, m)

b=fir2(n, f, m, window)

b=fir2(n, f, m, npt)

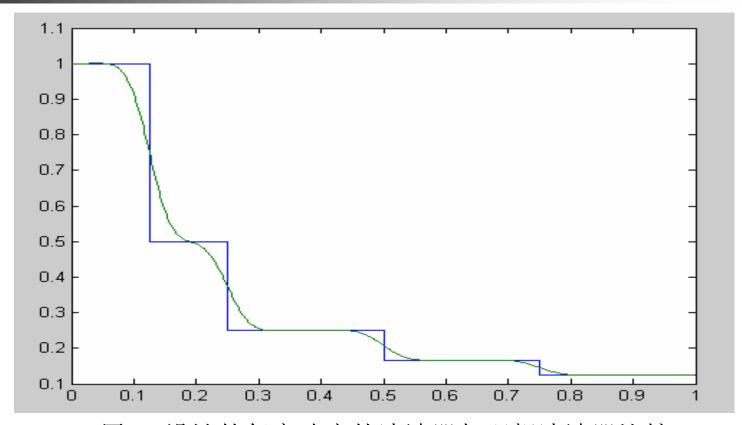
b=fir2(n, f, m, npt, window)

b=fir2(n, f, m, npt, lap)

设计一个60阶的滤波器,要求设计的滤波器在 $0-\pi/8$ 的幅度响应为1, 在  $\pi/8$  到 $2\pi/8$  的幅度响应为1/2, 在  $2\pi/8$  到 $4\pi/8$  的幅度响应为1/4, 在 $4\pi/8$ 到 $6\pi/8$ 的 幅度响应为1/6, 在 $6\pi/8$ 到 $\pi$ 的幅度响应为1/8。并画 出该滤波器和理想滤波器的幅频响应曲线,进行比较。

> 程序: f=[0,0.125,0.125,0.25,0.25,0.5,0.5,0.75,0.75,1]; m=[1,1,0.5,0.5,0.25,0.25,1/6,1/6,0.125,0.125];b = fir2(60, f, m);[h,w] = freqz(b);plot(f,m,w/pi,abs(h));





用fir2设计的任意响应的滤波器与理想滤波器比较



- ·切比雪夫逼近法设计FIR滤波器
- > 采用最大误差最小准则方法设计滤波器

**优化问题** 

► 雷米兹 (REMEZ)算法 较成功的设计线性相位滤波器的方法



# 利用remez函数设计FIR数字滤波器 MATLAB语法:

b=remez(n, f, m)

b=remez(n, f, m, w)

b=remez(n, f, m, 'ftype')

b=remez(n, f, m, w, 'ftype')

• • • • •



## 其它的MATLAB函数:

firls remezord

fires firels1

cremez

firrcos

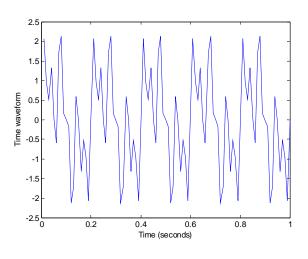


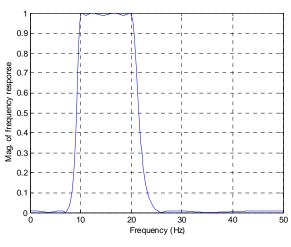
## IIR与FIR数字滤波器的比较

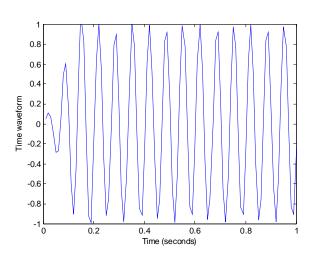
- ▶ 在相同的技术指标下,IIR可以用比FIR 较少的阶数来满足要求;
- > FIR可得到严格的线性相位;
- > IIR的设计可利用模拟滤波器的结果
- ▶ IIR主要设计规格化的滤波器,而FIR较 灵活,可是任意响应的滤波器,适应性 较广。



## ■一维信号处理:信号滤波







滤波后的结果

### 信号:

$$S=s1+s2+s3$$

### 滤波器:

$$[b,a] = ellip(4,0.1,40,[10 20]*2/Fs);$$
 sf = filter(b,a,s);  
 $[H,w] = freqz(b,a,512);$ 



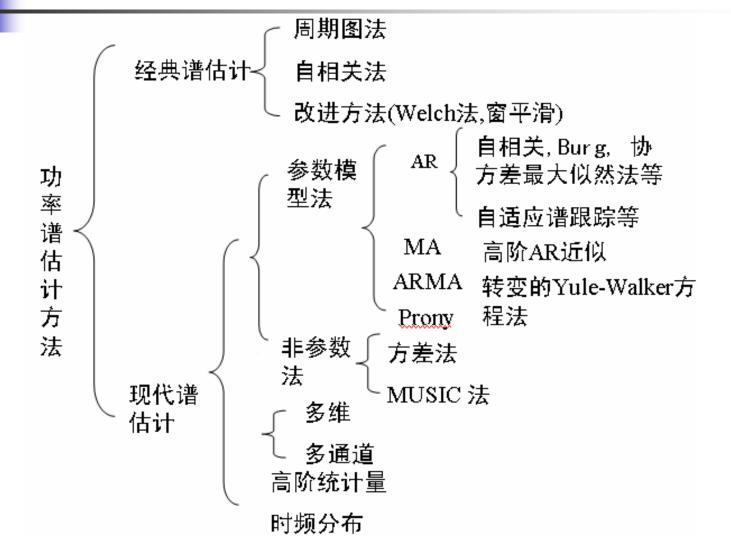
## 5 功率谱估计

### 概念:

平稳随机信号的功率谱密度(PSD)是自相关序列的离散时间傅立叶变换

$$P_{xx}(\varpi) = \sum_{m=-\infty}^{\infty} r_{xx}(m)e^{-j\varpi m}$$







## 经典功率谱估计方法

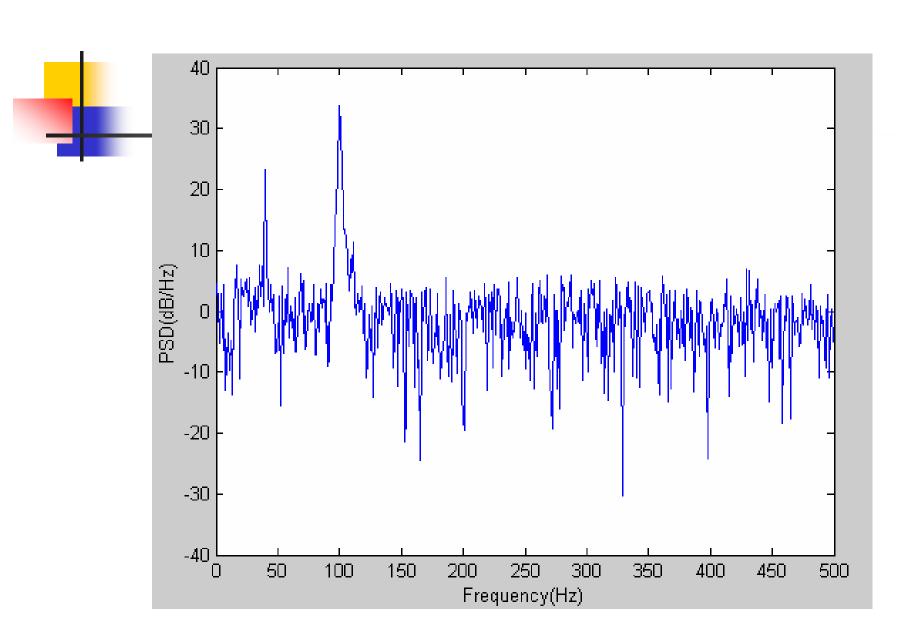
• 直接法(周期图法)

直接由傅立叶变换而得来的。

$$\hat{P}_{PER}(k) = \frac{1}{N} |X_N(k)|^2$$

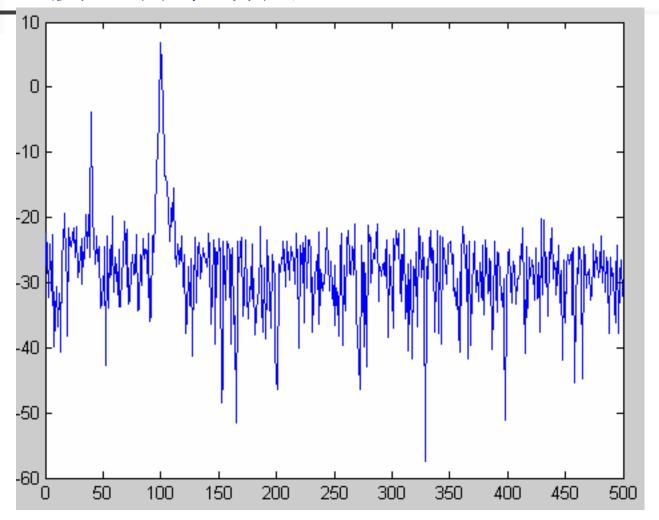
## 例子:直接法计算功率谱

```
利用FFT直接法:
Fs = 1000;
NFFT=1024;
n=0:1/Fs:1:
x=sin(2*pi*40*n)+4*sin(2*pi*100*n)+randn(size(n)); %噪声序列
X=fft(x,NFFT);
Pxx=abs(X).^2/length(n);
                             %求解PSD
t=0:round(NFFT/2-1);
k=t*Fs/NFFT;
P=10*(log10(Pxx(t+1)));
plot(k,P)
xlabel('Frequency(Hz)'); ylabel('PSD(dB/Hz)');
```



```
利用periodogram函数:
Fs=1000;
NFFT=1024;
n=0:1/Fs:1;
x=sin(2*pi*40*n)+4*sin(2*pi*100*n)+randn(size(n)); %噪声序列window=boxcar(length(x));
[Pxx,f]=periodogram(x,window,NFFT,Fs); %求PSD
Plot(f,10*log10(Pxx))
```

## 直接法功率谱图





• 间接法(自相关法或BT法)

其理论基础是维纳-辛钦定理。其方法是先由随机信号N个观察值估计出自相关函数 $\hat{r}(m)$ ,然后再求 $\hat{r}(m)$ 的傅立叶变换。

$$\hat{P}_{BT}(\omega) = \sum_{m=-M}^{M} \hat{r}(m)e^{-j\omega m} \qquad M \le N - 1$$

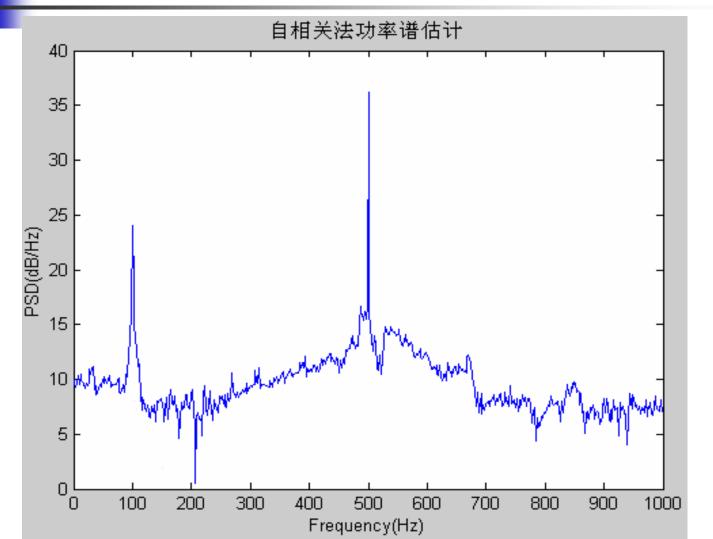
$$x(n) \xrightarrow{\text{dif}} \hat{r}(m)$$

# 4

### 例子: 间接法求功率谱估计

```
Fs = 2000:
NFFT=1024;
n=0:1/Fs:1;
x=sin(2*pi*100*n)+4*sin(2*pi*500*n)+randn(size(n));%噪声序列
Cx=xcorr(x,'unbiased'); %计算序列的自相关函数
Cxk=fft(Cx,NFFT);
Pxx=abs(Cxk); %求解PSD
t=0:round(NFFT/2-1);
k=t*Fs/NFFT;
P=10*(log10(Pxx(t+1)));
plot(k,P)
title('自相关法功率谱估计');
xlabel('Frequency(Hz)');ylabel('PSD(dB/Hz)');
```







• 改进法: Welch法 (psd、pwelch)

• AR模型功率谱估计