第十一章 DFT和FFT

1. DFT基础
2. DFT窗效应
3. 频谱图
4. FFT基础

信号处理的“双子星”：DFT 与 FFT

在数字信号处理中，DFT（离散傅里叶变换）宛如一颗璀璨的启明星。它像是一把神奇的钥匙，将离散的数字信号，从时域的平凡世界，带入频域的奇幻天地。每一个数字信号，都仿佛是一首无声的乐章，而 DFT 则赋予它们旋律，让我们能洞悉信号中隐藏的频率奥秘，知晓每个音符的强弱与频率。

然而，DFT 的计算过程有时稍显繁琐。此时，FFT（快速傅里叶变换）翩然而至，它是 DFT 的高效化身。FFT 运用独特的算法，大幅缩短计算时间，让我们能更迅速地在频域中翱翔，去探索那些被时间掩埋的频率细节，成为数字世界不可或缺的瑰宝。

DFT 基础

- **计算周期数字信号频谱**

傅里叶级数的系数

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j2\pi n\frac{k}{N}}$$

非周期数字信号

$$\Omega \Leftrightarrow 2\pi\frac{k}{N}$$

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-jn\Omega}$$

DFT 基础

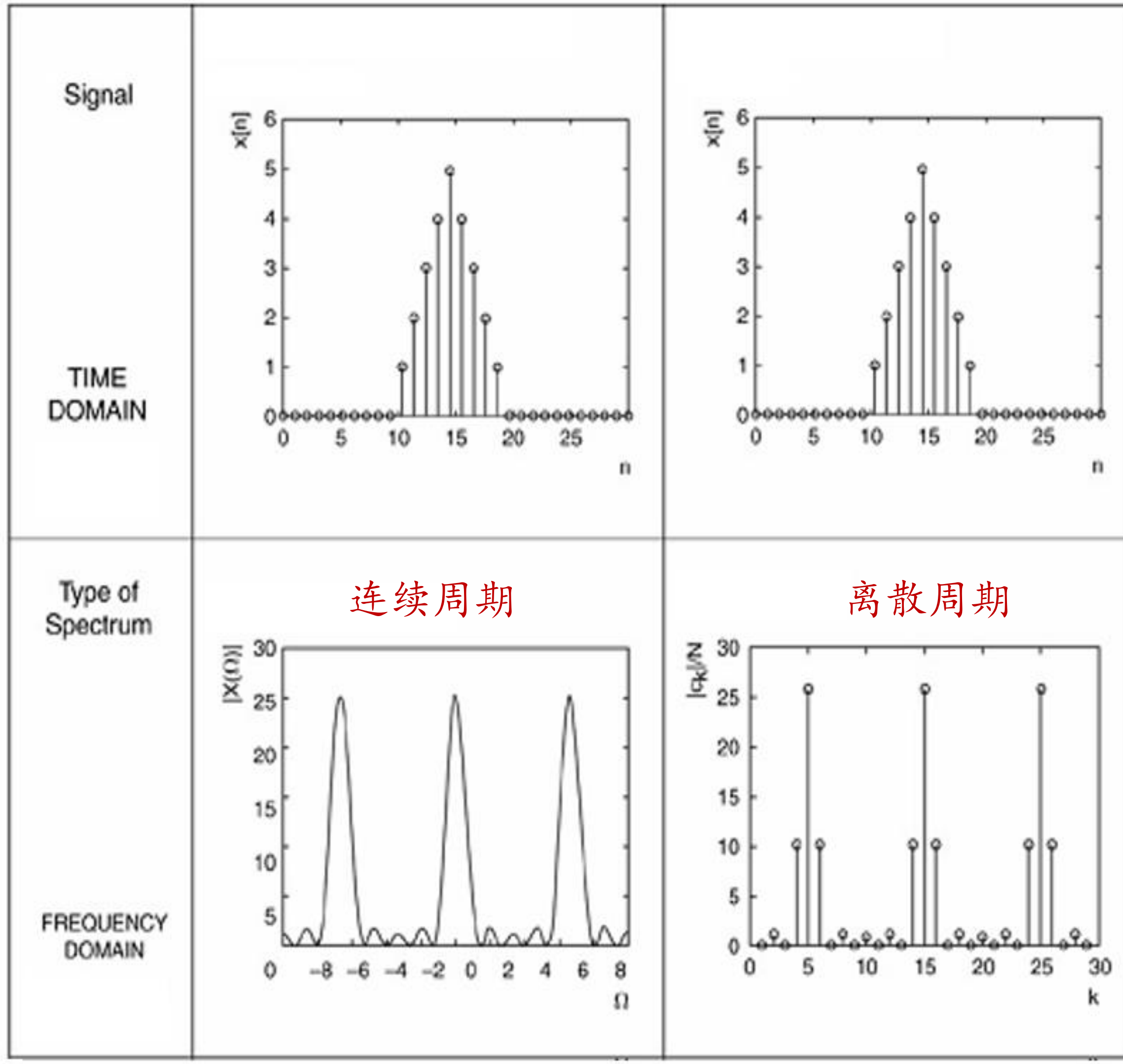
- 对数字信号 x[n] 的离散傅里叶变换 DFT（Discrete Fourier Transform）为

$$X(k)=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k}{N}n}$$

K=0, 1, 2……, N-1　　**N为采样点数**

计算有限个求和点

- 傅里叶变换把信号从时间域变换到频率域，进而研究信号的频谱结构和变化规律。
- DFT在时域和频域上都呈现离散的形式；
- DTFT在时域上呈现离散的形式，频域上呈现连续的形式。
  - ✓ 它将以离散时域的信号变换到连续的频域，即产生这个离散时域信号的连续频谱，值得注意的是这一**频谱是周期的**。



DTFT
离散时间傅里叶变换

DFT
离散傅里叶变换

DTFT

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

DFT

$$X(k)=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k}{N}n}$$

$$\Omega=2\pi k/N$$

$|X(k)|$ 可以表示如下：

$$X(k)=|X(k)|e^{j\theta}$$

$$X(k)=X(k+N)$$

$X(k)$是以 N为周期的

N为采样点数

- $|X(k)|$ 对k的关系成为幅度频谱
- θ对k的关系成为相位频谱

$X(k)$是以N为周期的

$$X(k+N)=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k+N}{N}n}$$

$$=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k}{N}n}e^{-j2\pi n}$$

$$=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k}{N}n}$$

$$=X(k)$$

以采样点数 N 为周期

例 11.1 求图 11.2 中所选采样信号的幅度频谱和相位频谱

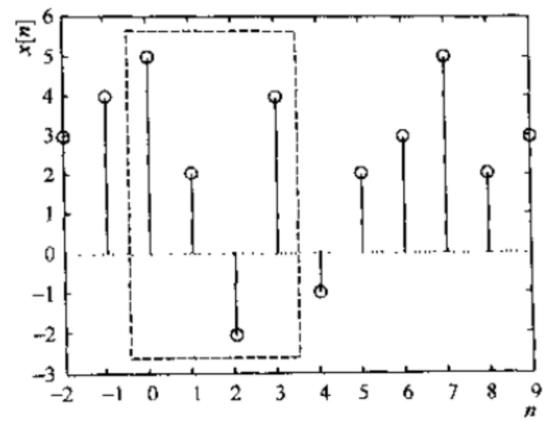
解：

DFT 窗选定了 4 个采样值进行分析，即 $N=4$，用式(11.1)的 DFT：

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k}{N}n}=x[0]+x[1]e^{-j2\pi\frac{k}{4}}+x[2]e^{-j2\pi\frac{k}{4}2}+x[3]e^{-j2\pi\frac{k}{4}3}$$

$$=5+2e^{-j\frac{\pi k}{2}}-2e^{-j\pi k}+4e^{-j\frac{\pi k3}{2}}$$

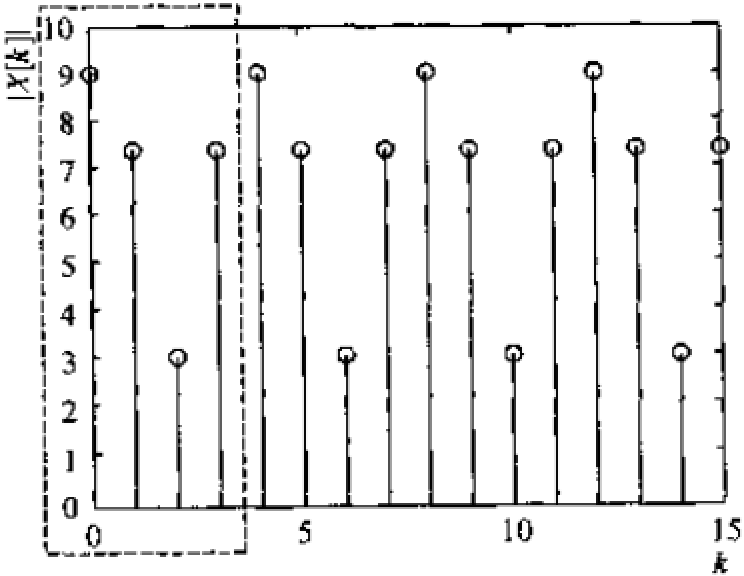
N=4，只研究4个点



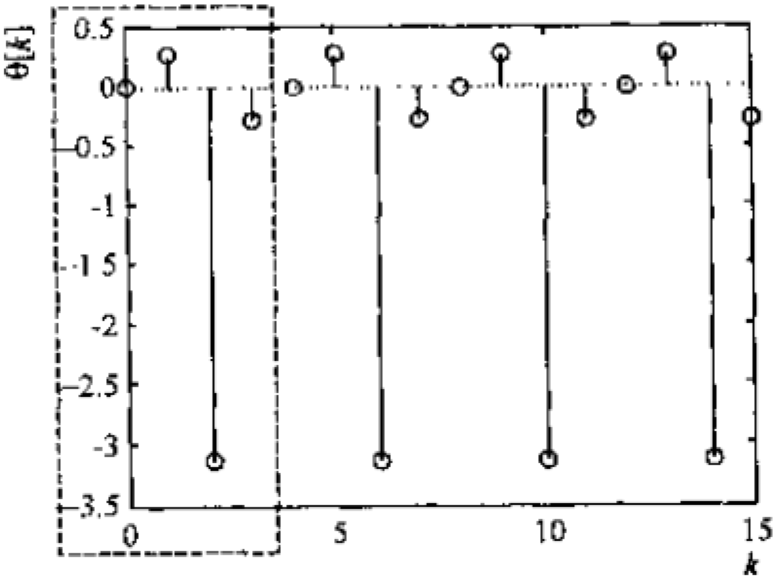
每个极坐标形式的复数 $re^{j\theta}$ 可简写成 $r\angle\theta$，记录了此数的幅度$\angle$相位。$e^{j\theta}$ 可简记为 $1\angle\theta$，使用这种标记法，$X[k]$ 可写成：

$$X[k]=5\angle 0+2\angle-\frac{\pi k}{2}-2\angle-\pi k+4\angle-\frac{\pi k3}{2}$$

角度为弧度。



(a) DFT 幅度频谱



(b) DFT 相位频谱

$X(k)$是以N为周期的

这里$N=4$

以　4个样点　为周期

例 11.2　用 DTFT 和 DFT 求图 11.5 所示信号的幅度频谱。

解：

DTFT 是：

$$X(\Omega)=\sum_{n=-\infty}^{\infty}x[n]e^{-jn\Omega}=2-e^{-j\Omega}+3e^{-j2\Omega}+3e^{-j3\Omega}$$

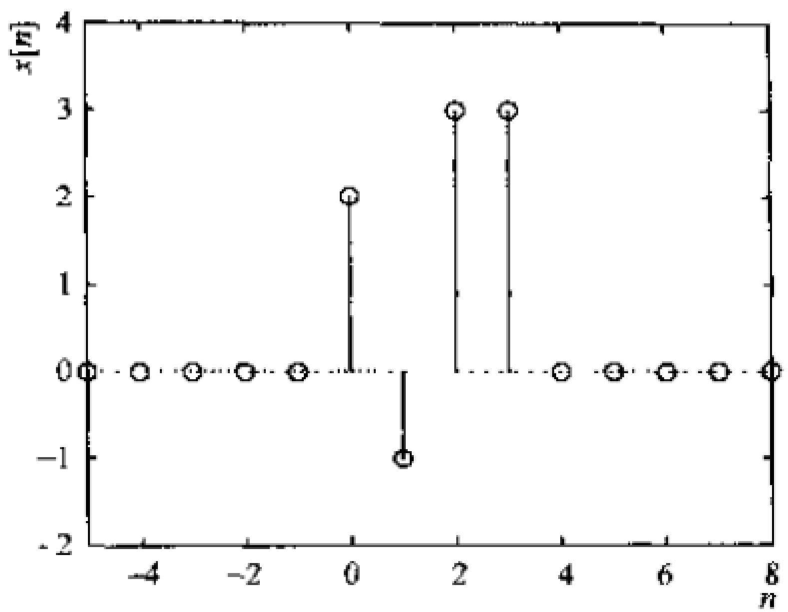


图 11.5　例 11.2 的信号

对图 11.5 中 4 个非零采样值，$N=4$ 点 DFT 可由下式计算：

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j2\pi\frac{k}{N}n}=2-e^{-j2\pi\frac{k}{4}}+3e^{-j2\pi\frac{k}{4}}+3e^{-j2\pi\frac{k}{4}3}$$

$$=2-e^{-j\frac{\pi k}{2}}+3e^{-j\pi k}+3e^{-j\frac{\pi 3k}{2}}$$

对比分析一下

$$X(\Omega)=\sum_{n=-\infty}^{\infty}x[n]e^{-jn\Omega}=2-e^{-j\Omega}+3e^{-j2\Omega}+3e^{-j3\Omega}$$

$$\Omega=2\pi k/N \qquad k=\frac{\Omega N}{2\pi}$$

$$\Omega=-2\pi \quad\Rightarrow\quad k=\frac{\Omega N}{2\pi}=\frac{-2\pi N}{2\pi}=-N=-4 \qquad N=4$$

$$\Omega=4\pi \quad\Rightarrow\quad k=\frac{\Omega N}{2\pi}=\frac{4\pi N}{2\pi}=2N=8$$

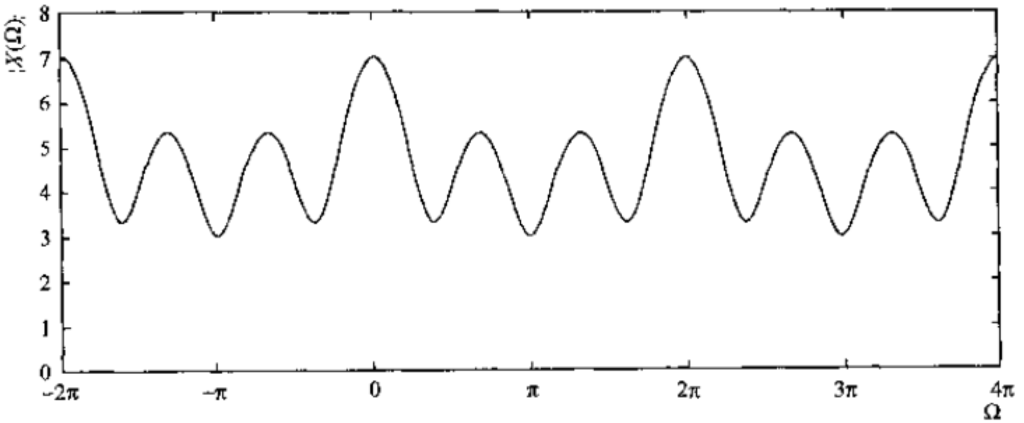


图 11.6 例 11.2 的 DTFT 幅度频谱

$$\Omega = -2\pi \rightarrow \quad 4\pi$$

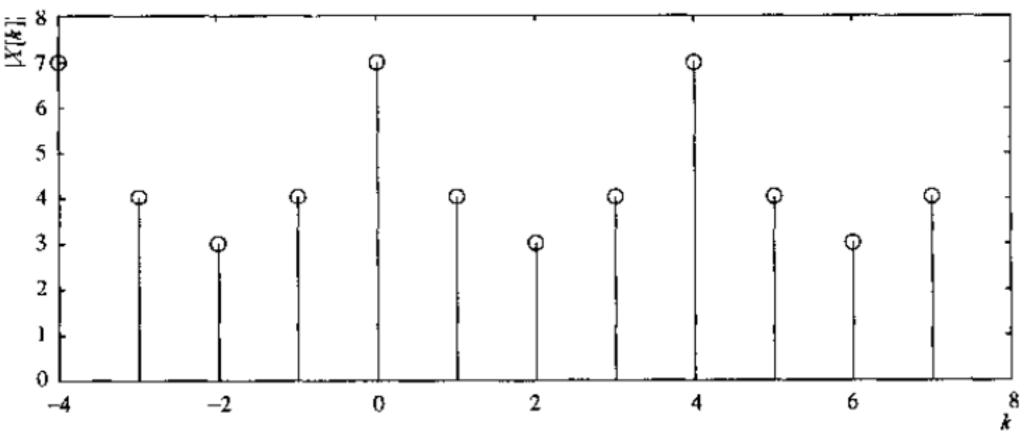
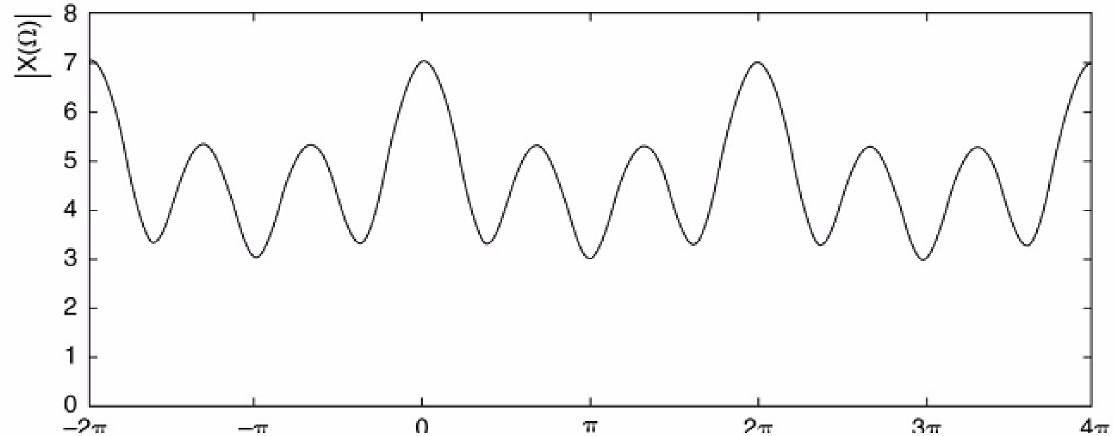


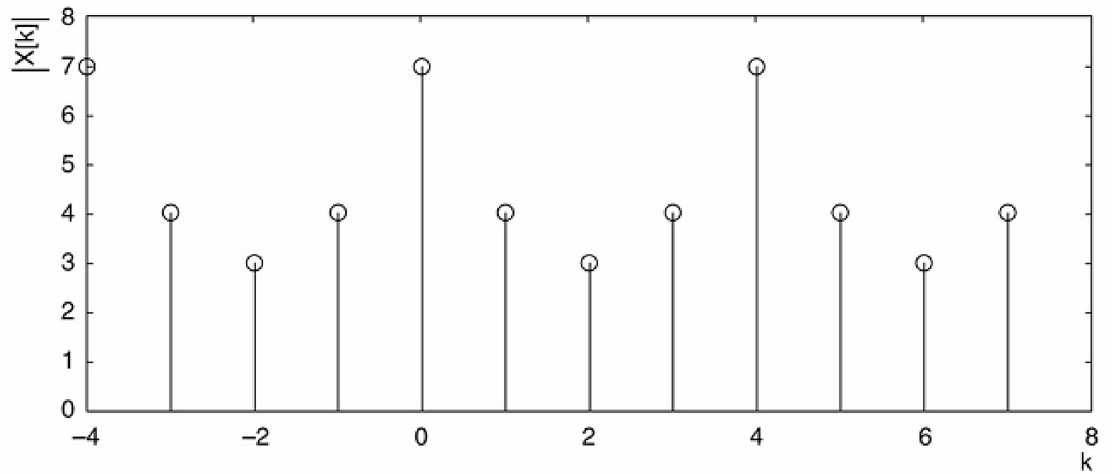
图 11.7 例 11.2 的 DFT 幅度频谱

$$k = -4 \quad \rightarrow \quad 8$$

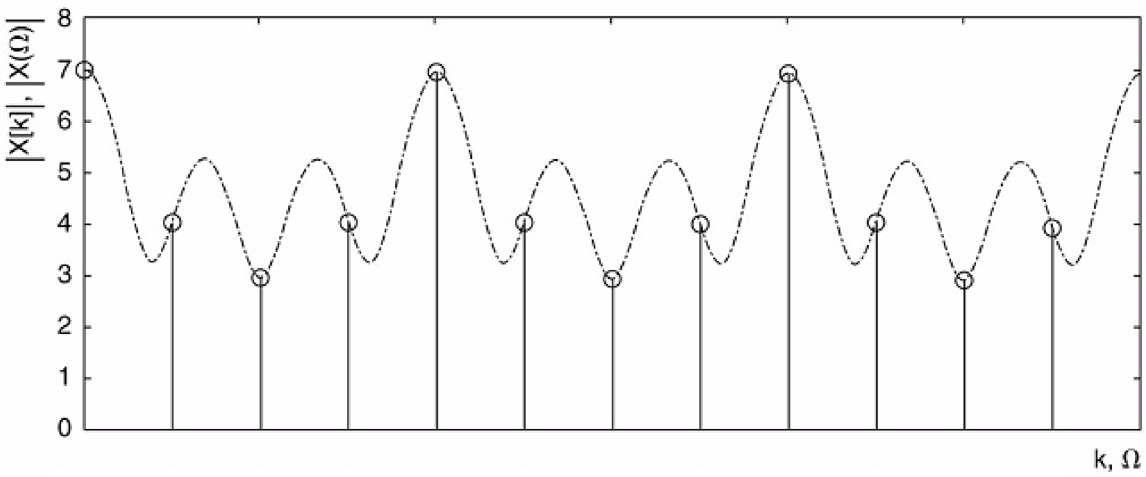
DTFT与DFT



|X(Ω)|
8
7
6
5
4
3
2
1
0
−2π
−π
0
π
2π
3π
4π



|X[k]|
8
7
6
5
4
3
2
1
0
−4
−2
0
2
4
6
8
k



|X[k]|, |X(Ω)|
8
7
6
5
4
3
2
1
0
k, Ω

11.2 DFT窗效应 (只研究窗内点)

$$X(k)=\sum_{n=0}^{N-1}x[n]w[n]e^{-j2\pi\frac{k}{N}n}$$

K=0, 1, 2……, N-1

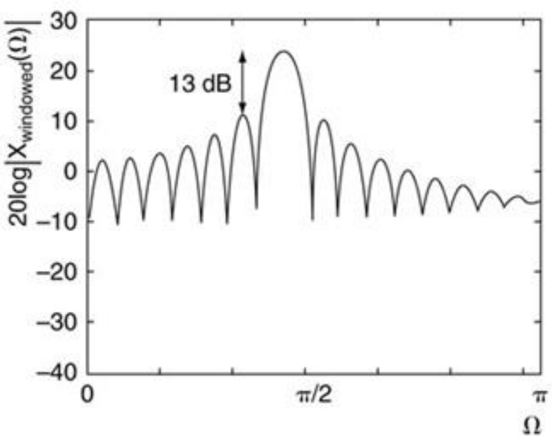
N为采样点数

$w[n]$：DFT窗，在时域内选择有限的采样值

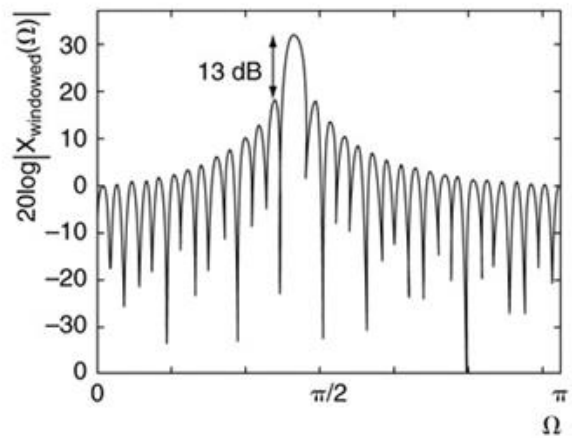
通常选择矩型窗：

窗对所有窗内采样值的权值为1，对所有窗外采样值的权值为0, DFT内的窗内采样点数为N。

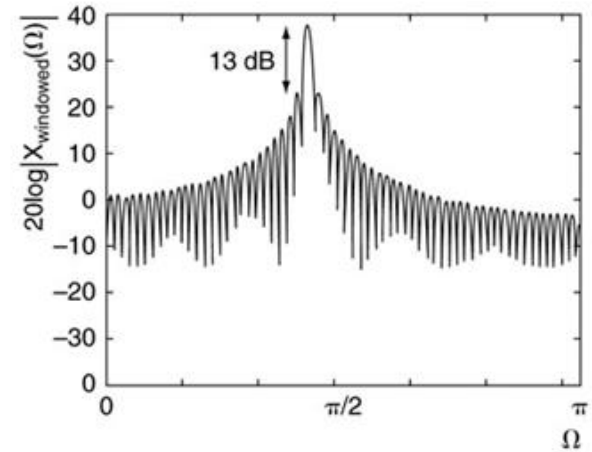
不同长度矩形窗频谱效应



(c) 32 samples



(b) 64 samples



(a) 128 samples

窗口长度一定时，窗内的采样点数N越多,可捕获的信息越详细。

11.3 频谱图

- 能否在一张图上同时显示信号-时间、信号-频率?

- 频谱图：它是频率对时间的图形，是DFT的集和。。

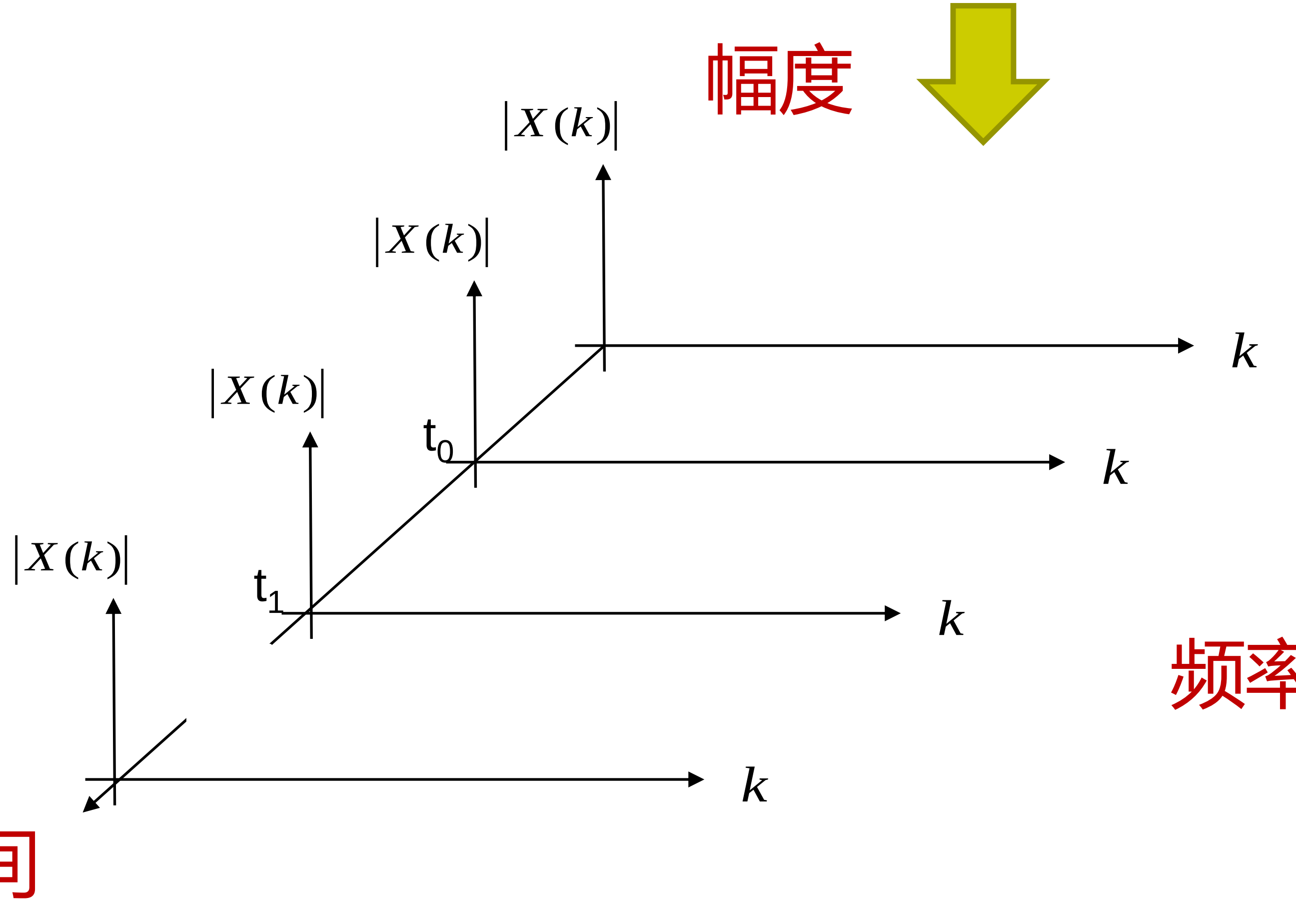
某一时刻 t_0

某一时刻 t_1

10.3 频谱图

$$X(k)=\sum_{n=0}^{N-1}x[n]w[n]e^{-j2\pi\frac{k}{N}n}$$

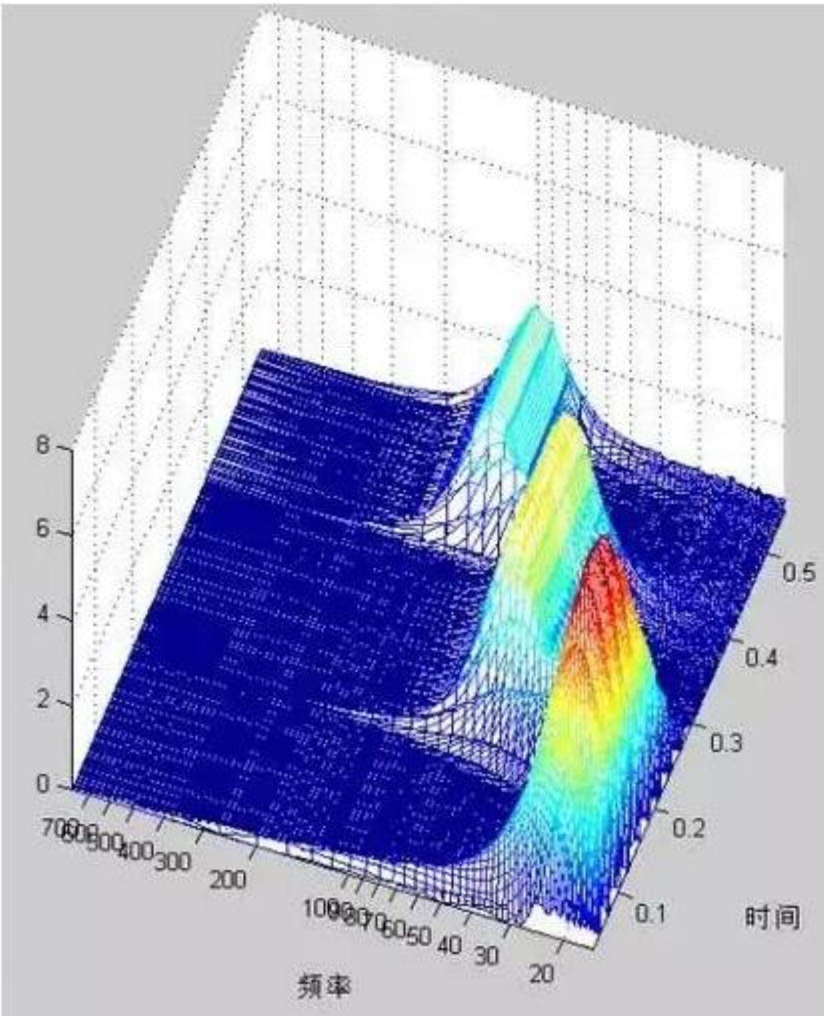
K=0, 1, 2……, N-1

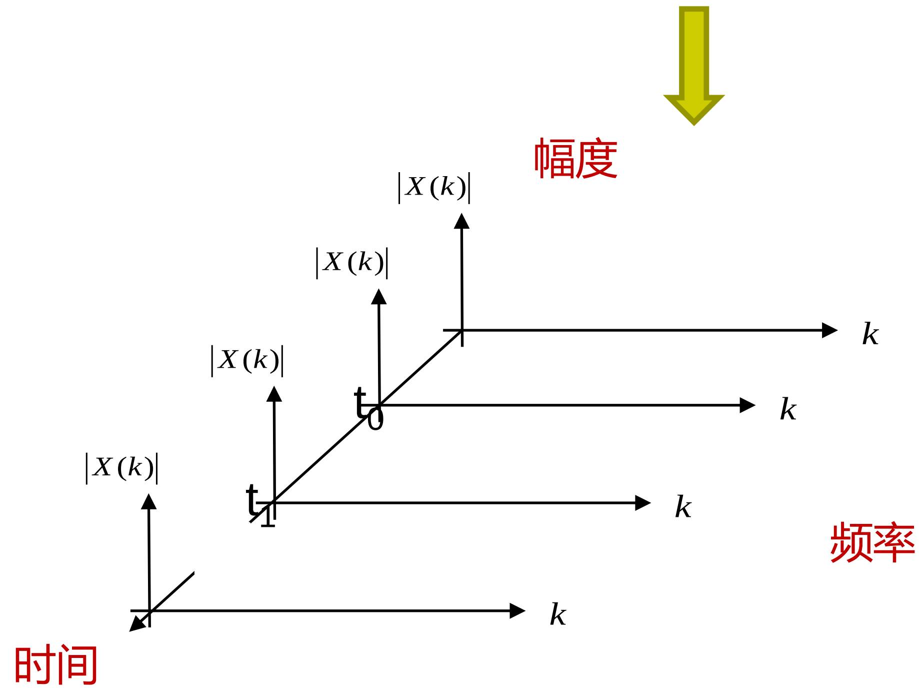


11.3 频谱图

$$X(k)=\sum_{n=0}^{N-1}x[n]w[n]e^{-j2\pi\frac{k}{N}n}$$

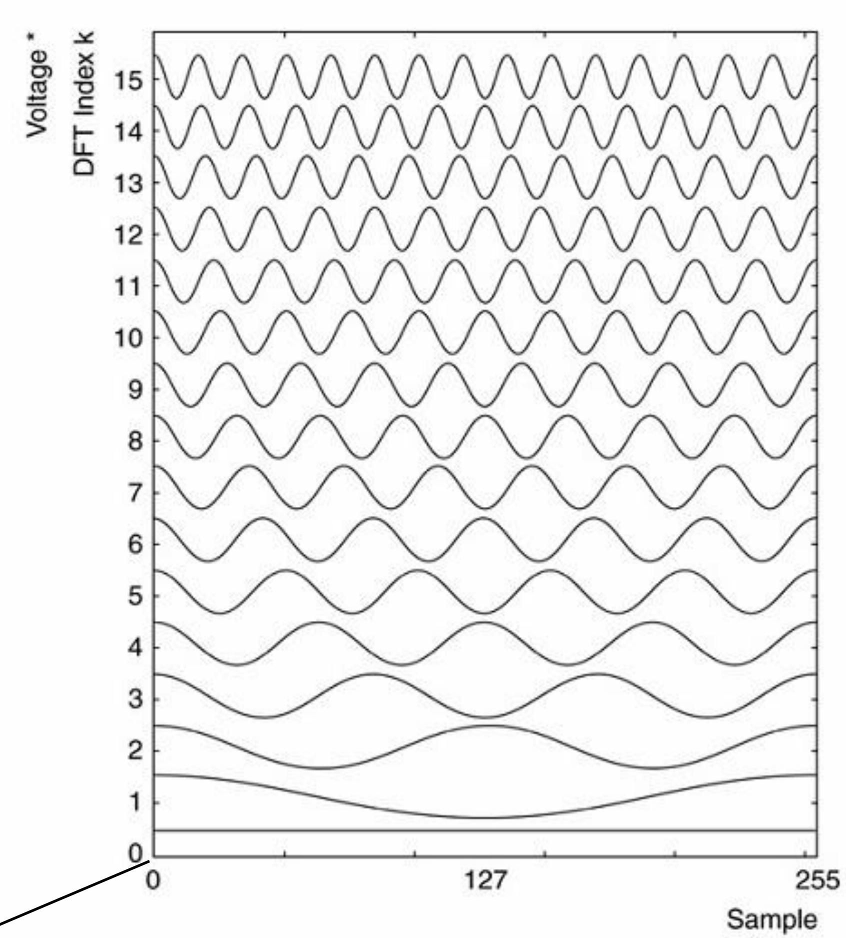
K=0, 1, 2……, N-1

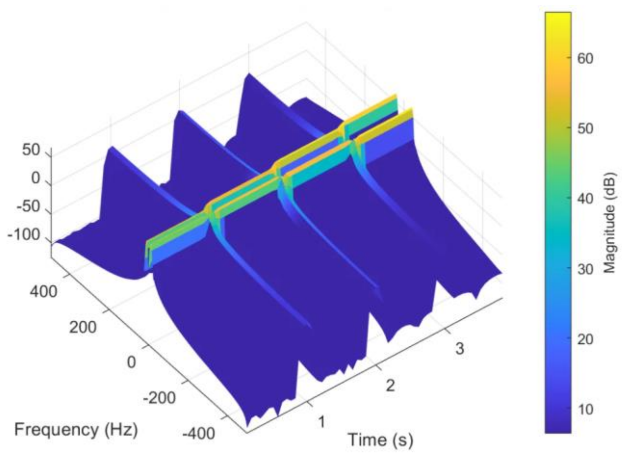




11.3　频谱图

频率





时间

幅度

频谱图

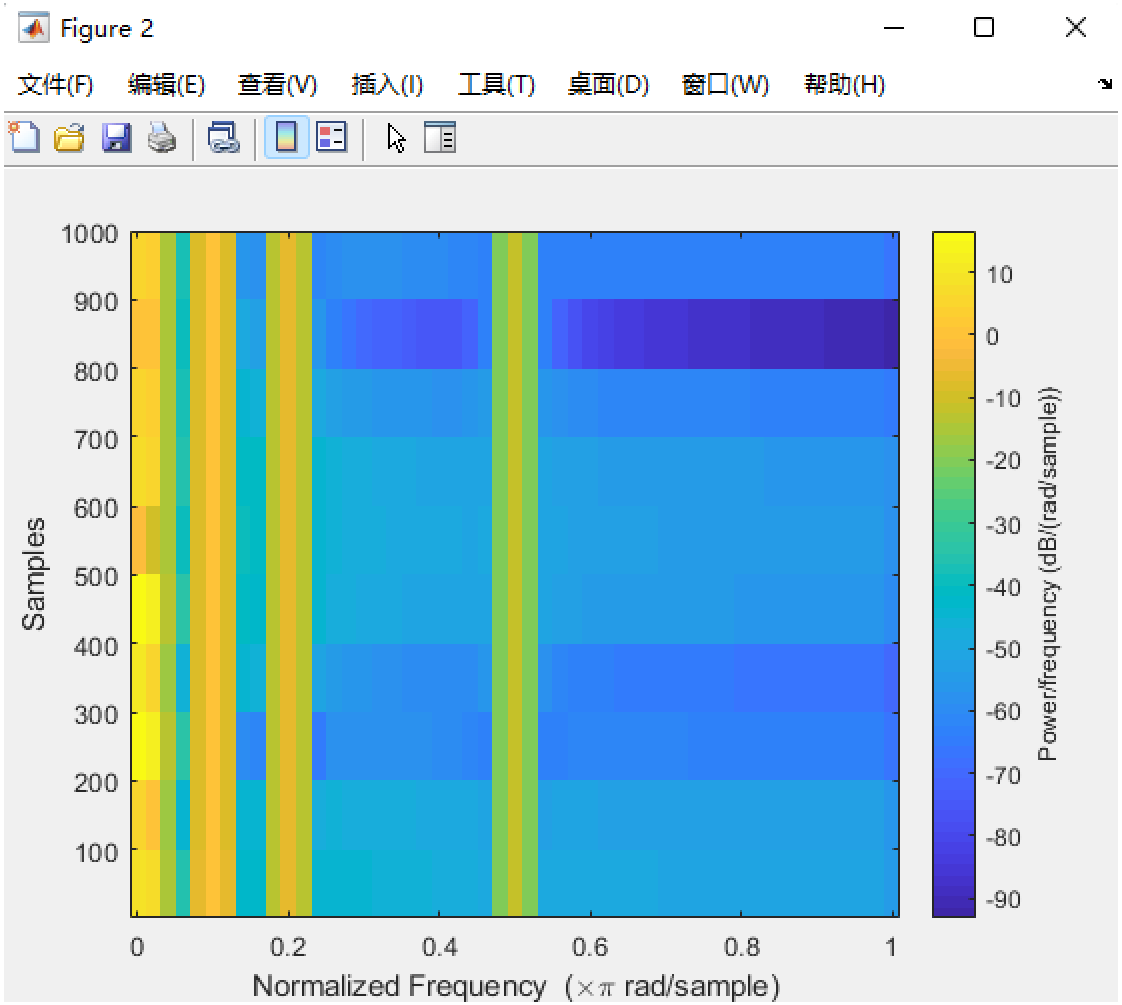


Figure 2
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
1000
900
800
700
600
500
400
300
200
100
Samples
0
0.2
0.4
0.6
0.8
1
Normalized Frequency (×π rad/sample)
10
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
Power/frequency (dB/(rad/sample))

频谱图

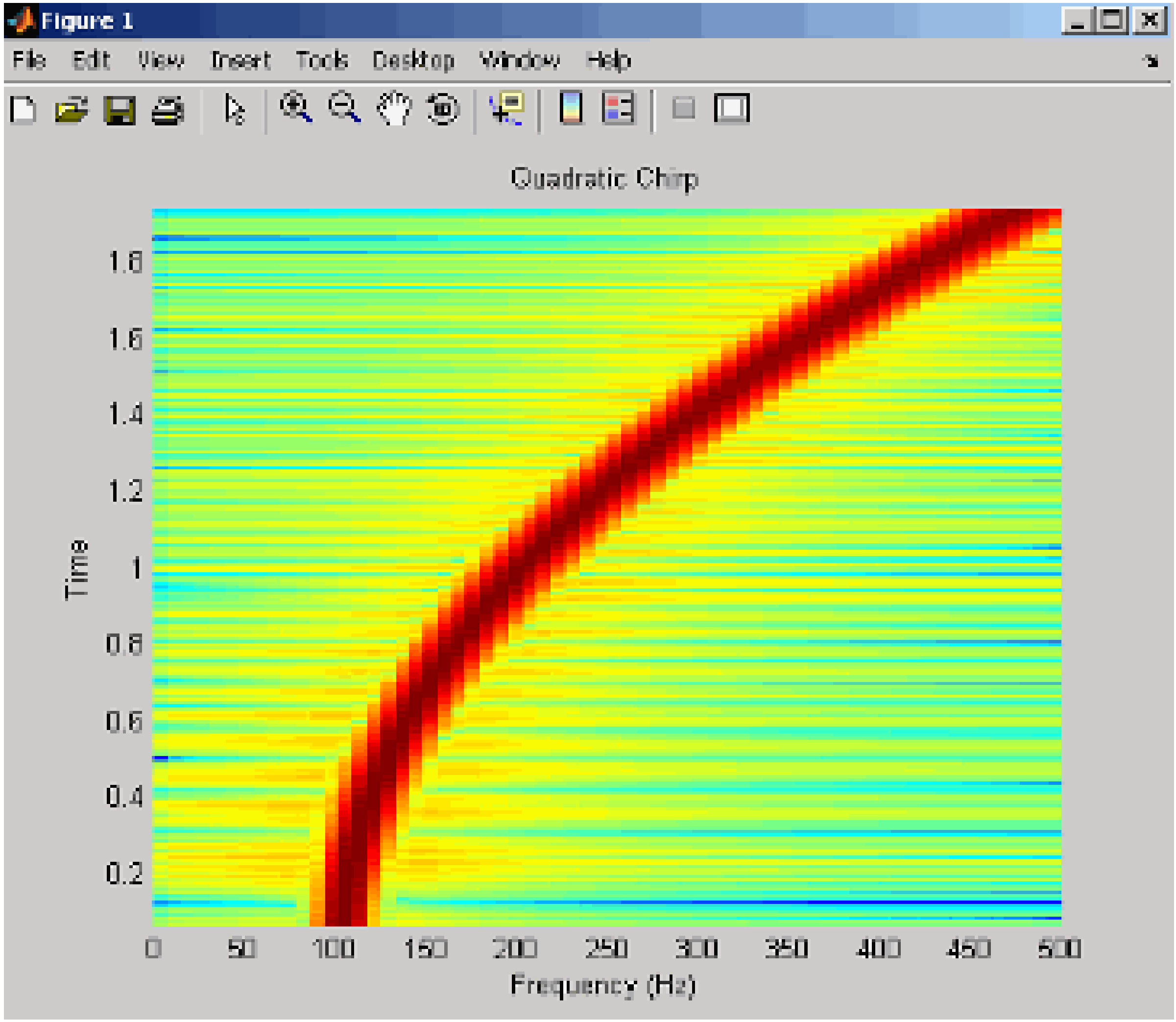
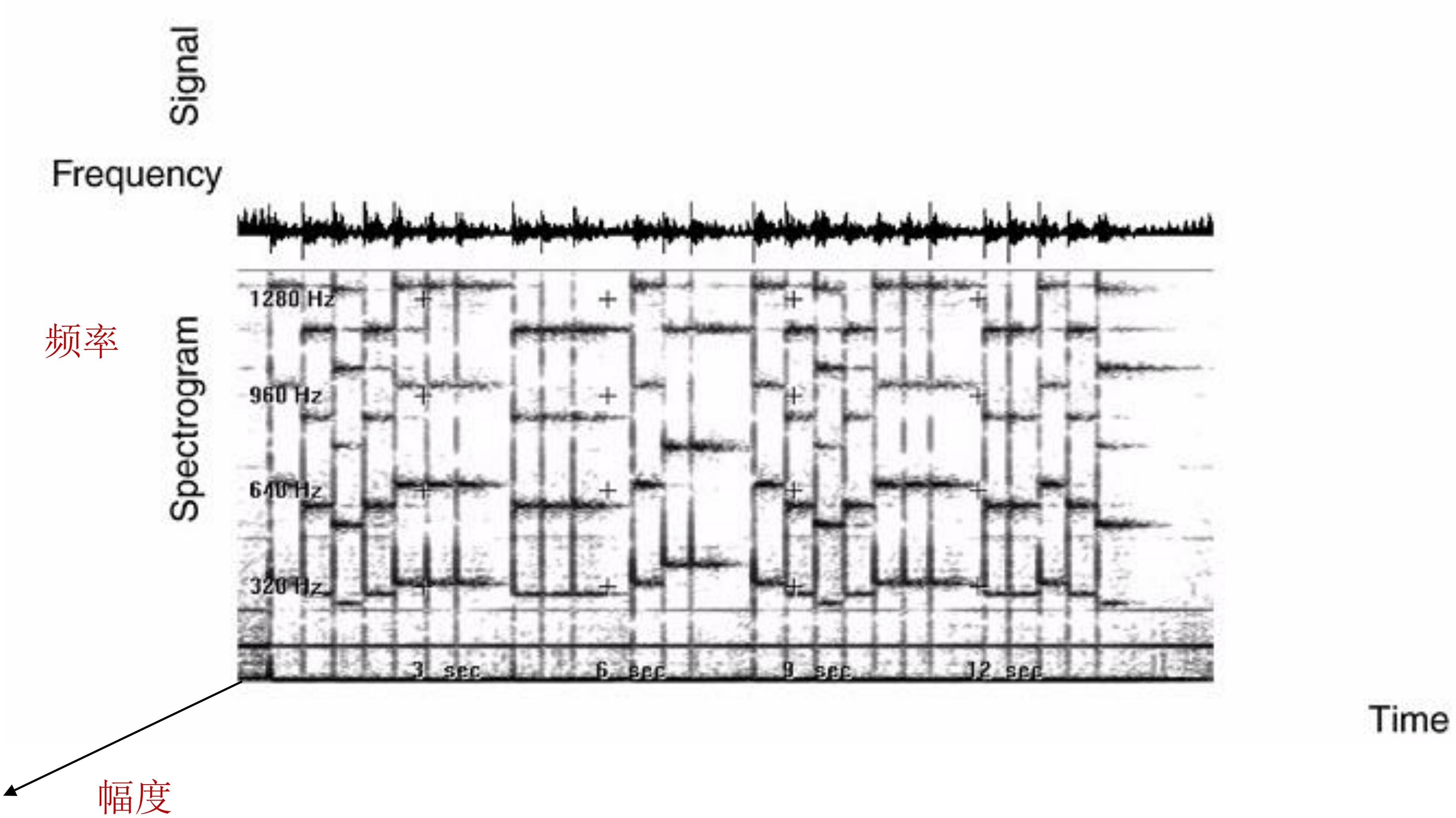
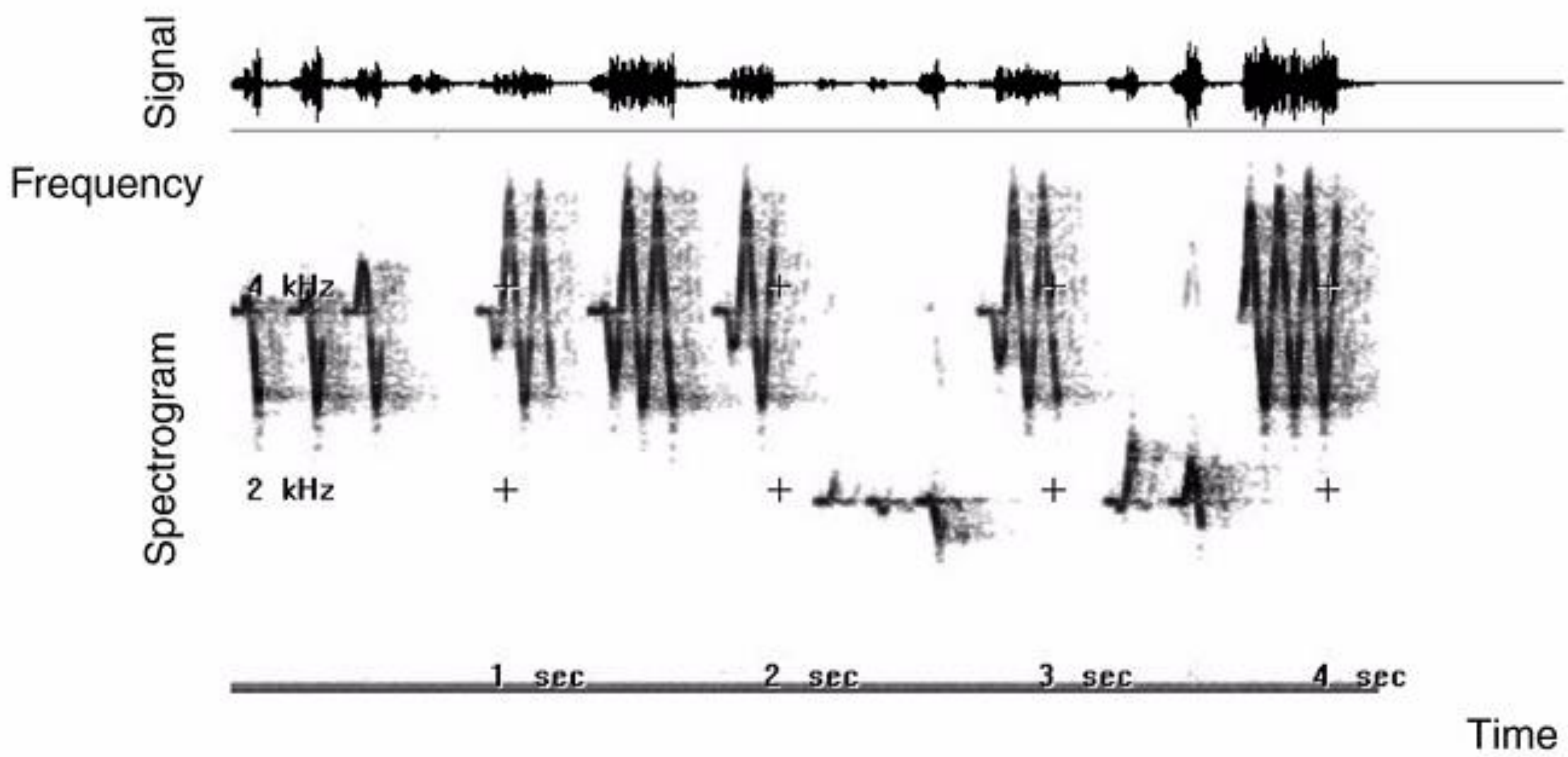


Figure 1
File Edit View Insert Tools Desktop Window Help
Quadratic Chirp
Time
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0 50 100 150 200 250 300 350 400 450 500
Frequency (Hz)

频谱图


Signal
Frequency
频率
Spectrogram
1280 Hz
960 Hz
640 Hz
320 Hz
3 sec
6 sec
9 sec
12 sec
Time
幅度

频谱图



(d) Spectrogram of Bird Song

Matlab

spectrogram(s,w,o,n);

s：信号，

w：时间窗大小（包含的点数），

o：时间窗重合大小，（重合的点数），

n：频率点数（点数越多频率分辨率越高）

```
x=1:1000;
y=sin(x/1000*pi)+0.8*sin(x/500*pi)+0.6*sin(x/100*pi)+0.4*sin(x/10*pi)+0.2*sin(x/5*pi)+0.1*sin(x/2*pi);
figure;
plot(y);%画出信号时域图
figure;
spectrogram(y,100,0,100);%画出信号频谱图
```

Matlab

spectrogram(s,w,o,N);

s：信号，

w：时间窗大小（包含的点数），

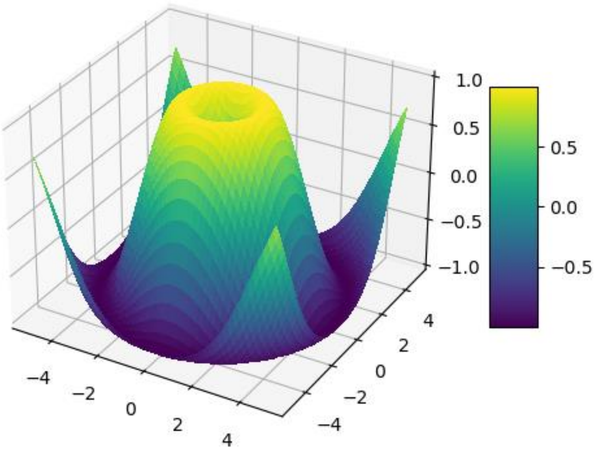
o：时间窗重合大小，（重合的点数），

N：频率点数（点数越多频率分辨率越高）

```
n=1:1000;
Y=sin(n/1000*pi)+0.8*sin(n/500*pi)+0.6*sin(n/100*pi)+0.4*sin(n/10*pi)+0.2*sin(n/5*pi)+0.1*sin(n/2*pi);
figure;
subplot(211),plot(y);%画出信号时域图
subplot(212),spectrogram(y,100,0,100)%画出信号频谱图
```

Python

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# 创建一些数据
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))

# 创建一个新的图像
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# 绘制三维曲面
surf = ax.plot_surface(x, y, z, cmap='viridis',
linewidth=0, antialiased=False)

# 为图像添加一个色条
ax.set_zlim(-1.01, 1.01)
fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()
```



11.4 快速傅里叶变换（FFT）

FFT（fast Fourier transform）快速傅里叶变换

FFT与DFT的输出一样，但运算量要小得多

基本原理：

- 将一个N点的计算分解为两个N/2点的计算，每个N/2点的计算再进一步分解为N/4点的计算，以此类推……。

DFT
N个采样点

FFT
计算
前N/2个采样点的
傅里叶变换

FFT
通过周期性得到
后N/2个样点的
傅里叶变换

- 根据采样编号n，分为偶数采样点和奇数采样点。
- 令：y[n]=x[2n]，z[n]=x[2n+1]

对于前一半计算 （前一个N/2个采样点）：

$$X(k)=\sum_{n=0}^{(N/2)-1} x[2n]e^{-j2\pi\frac{k}{N}(2n)}+\sum_{n=0}^{(N/2)-1} x[2n+1]e^{-j2\pi\frac{k}{N}(2n+1)}$$

$$X(k)=\sum_{n=0}^{(N/2)-1} y[n]e^{-j\frac{4\pi kn}{N}}+\sum_{n=0}^{(N/2)-1} z[n]e^{-j\frac{2\pi k(2n+1)}{N}}$$

$$X(k)=\sum_{n=0}^{(N/2)-1} y[n]e^{-j\frac{4\pi kn}{N}}+\sum_{n=0}^{(N/2)-1} z[n]e^{-j\frac{2\pi k(2n+1)}{N}}$$

$$X(k)=\sum_{n=0}^{(N/2)-1} y[n]e^{-j\frac{2\pi kn}{(N/2)}}+\sum_{n=0}^{(N/2)-1} z[n]e^{-j\frac{2\pi kn}{(N/2)}} * \quad e^{-j\frac{2\pi k}{N}}$$

令：$y[n]$的傅里叶变换为Y(k)，$z[n]$的傅里叶变换为Z(k)

$$X(k)=Y(k)+e^{-j\frac{2\pi k}{N}}Z(k)$$

注意：这里的采样点数是N/2

所以是以N/2为周期的

注意：这里的采样点数是N/2

因为前一半计算 （前一个N/2个采样点）

$$X(k)=Y(k)+e^{-j\frac{2\pi k}{N}}Z(k)$$

那么后一半计算 （后一个N/2个采样点）就有

$$X(k+N/2)=Y(k+N/2)+e^{-j\frac{2\pi(k+N/2)}{N}}Z(k+N/2)$$

因为是以N/2为周期的，所以有

$$Y(k+N/2)=Y(k),\qquad Z(k+N/2)=Z(k)$$

$$e^{-j\frac{2\pi(k+N/2)}{N}}=e^{-j\frac{2\pi k}{N}}*\quad e^{-j\pi}=-e^{-j\frac{2\pi k}{N}}$$

所以后一半计算（后一个N/2个采样点）就有

$$X(k+N/2)=Y(k)-e^{-j\frac{2\pi k}{N}}Z(k)$$

而前一半计算（前一个N/2个采样点）

$$X(k)=Y(k)+e^{-j\frac{2\pi k}{N}}Z(k)$$

比较上面两个式子：

- 将一个N点的计算分解为两个N/2点的计算，只需一次计算出$Y(k)$和$Z(k)$，然后通过不同的组合得到N个点的傅里叶变换，简化了运算。

例：求下面函数的离散傅里叶变换

$x = \sin(10\pi t) + \sin(8\pi t)$

```
t = 0:0.01:10;
x=sin(10*pi*t) + sin(8*pi*t);
y = fft(x);                    % 计算DFT
m = abs(y); % 幅度
p = unwrap(angle(y)); %相角
plot(m);
figure
plot(p)
```

练习：对自己存在电脑中的信号做 fft

```
[x,fs]= audioread ('test.wma'); %写出文件路径
y=x(:,1); % 单声道？
N=10000; % 取确定的样点数N
Y = fft(y);
% Plot single-sided amplitude spectrum.
figure
subplot(211),plot(y);
subplot(212),plot(abs(Y(1:N/2))) %只画的0-N/2的绝对值
```