



第三章 离散信号的分析

浙江大学 电气工程学院 杨欢 yanghuan@zju.edu.cn





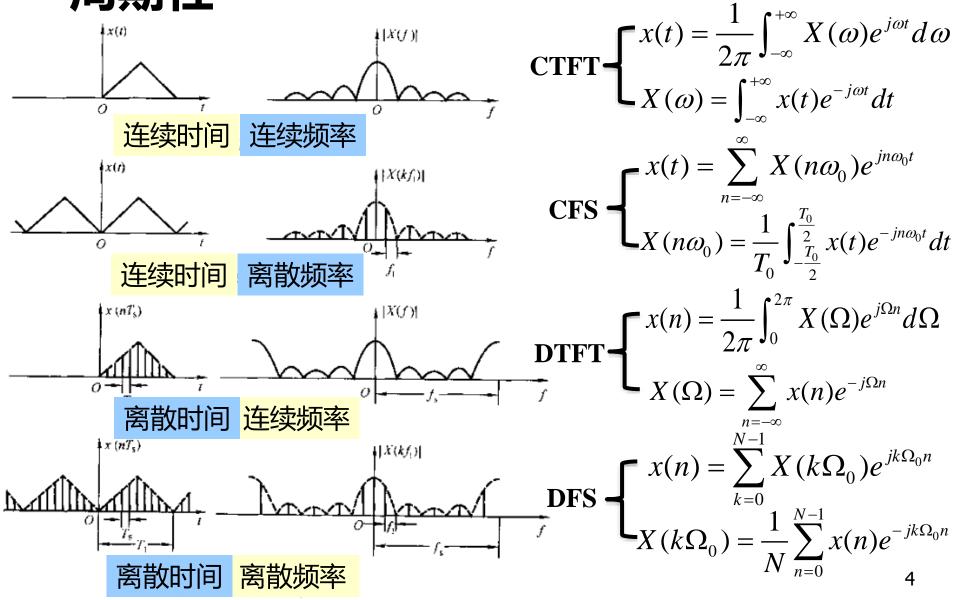
时域周期性——频域离散性时域离散性——频域周期性

时域非周期——频域连续性 时域连续性——频域非周期





时域		频域(频谱	形式)	对应的傅里叶级数/变换
<mark>连续</mark> 时间,		非周期,	1. 级数形 离散	
<mark>周期</mark> , <i>T</i> ₀	CFS	<mark>离散</mark> 频率 ω_0	2. 谐波分 无穷, 期	非問 $X(n\omega_0) = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t) e^{-jn\omega_0 t} dt$
离散序列,		周期, N=2	1. 级数形式 离散	$ x(n) = \sum X(k\Omega_0)e^{jk\Omega_0 n} $
<mark>周期</mark> ,T ₀ N点采样	DFS	离散频率,	2. 谐波分量 有限, 周期	$X(k\Omega_0) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)e^{-jk\Omega_0 n}$
<mark>连续</mark> 时间,	CTFT	非周期, 1.	积分形式-续	$X(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{s} d\omega$
非周期,		连续频率	谐波分量- 穷,非周期	$V(\alpha) = V(t) \alpha J^{(\alpha)} dt$
离散序列,	DTFT	周期 2π 1.	积分形式-	$\gamma_{\pi} \int_{0}^{\pi} \Lambda(3z)e^{-\pi 3z}$
非周期,		连续频率 2.	谐波分量-有限,周期	$\Delta (\Delta \Delta) = A(B)$



时域 频域	连续	离散
<u>连续</u>	时域 均为 <u>非周期</u> 、连续 频域	$x(n) = \frac{1}{2\pi} \int_0^{2\pi} X(\Omega) e^{j\Omega n} d\Omega$ DTFT $X(\Omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\Omega n}$ 时域非周期、离散 频域连续、周期
离散	CFS $x(t) = \sum_{n=-\infty}^{\infty} X(n\omega_0) e^{jn\omega_0 t}$ $X(n\omega_0) = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t) e^{-jn\omega_0 t} dt$ 时域周期、连续 频域离散、非周期	$X(n) = \sum_{k=0}^{N-1} X(k\Omega_0) e^{jk\Omega_0 n}$ DFS $X(k\Omega_0) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-jk\Omega_0 n}$ 时域 均为局期、离散

三、离散傅立叶变换 (DFT)







- ⇒ 从有限长序列的 DTFT 到 DFT
- ⇒ 从周期序列的 DFS 到有限长序列 DFT
- **⇒** DFT 的性质

从有限长序列的DTFT到DFT





$$X(\Omega) = \sum_{n=0}^{N-1} x(n)e^{-j\Omega n}$$

$$x(n) = \frac{1}{2\pi} \int_0^{2\pi} X(\Omega) e^{j\Omega n} d\Omega$$

离散信号的 DTFT,它是 Ω 的连续周期函数,尽管在理论上有重要意义,但在计算机上实现有困难。为此,需要一种时域和频域上都是离散的傅里叶变换对,实现计算机的快速计算,这就是离散傅里叶变换 DFT。

能量有限、时间点度为 L 的有限长序列的 DTFT 为





$$X(\Omega) = \sum_{n=0}^{L-1} x(n)e^{-j\Omega n}$$

频率离散化
$$\int_{0}^{\infty} k\Omega_{0} = k2\pi/N \rightarrow \Omega$$

$$X(k\frac{2\pi}{N}) = X(k\Omega_0) = \sum_{n=0}^{L-1} x(n)e^{-jk2\pi n/N}$$

频率采样点数N已 知, 2π/N为定数

$$X(k) = \sum_{n=0}^{L-1} x(n)e^{-jk2\pi n/N}$$

N点DFT是有限长序列 (L≤N,由频率采样定理决定)的TFT的N点均匀取样值,也就是非周期序列频谱的样值。

从DFS到DFT





DFS:
$$\tilde{X}_N(k\Omega_0) = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{x}_N(n) e^{-jnk\Omega_0}$$

$$\tilde{x}_N(n) = \sum_{k=0}^{N-1} \tilde{X}_N(k\Omega_0) e^{jk\Omega_0 n}$$

为了计算的方便,通常将 1/N 移到 $N_N(k\Omega_0)$ 中,而且

二者所具有的物理意义和性质都相同

$$\tilde{X}_N(k) = \sum_{n=0}^{N-1} \tilde{x}_N(n) e^{-j\frac{2\pi}{N}nk}$$

$$\tilde{x}_N(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_N(k) e^{j\frac{2\pi}{N}kn}$$

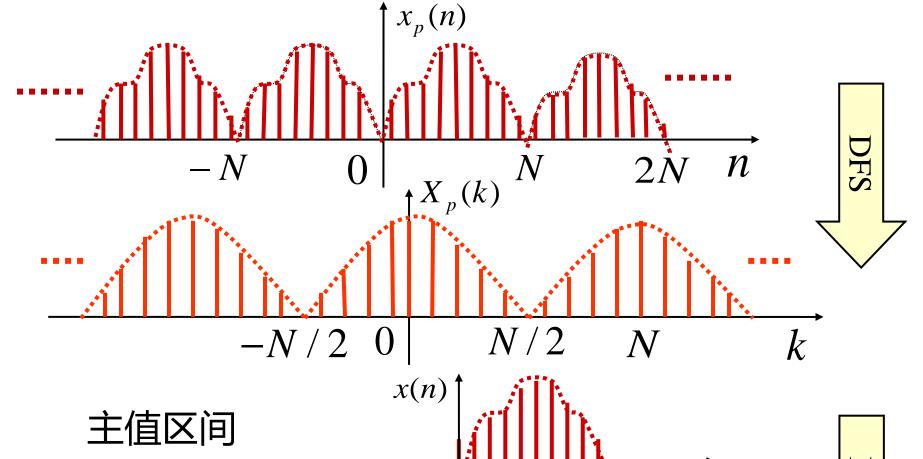
从DFS到DFT

DFT 又可看作以有限 长序列 x(n) 为一个周期, 进行周期延拓后所形成的周期序列 $x_p(n)$ 的离散频谱

② 设
$$\tilde{\chi}_N(n) \stackrel{DFS}{\longleftrightarrow} \tilde{X}_N(k)$$
 , 令

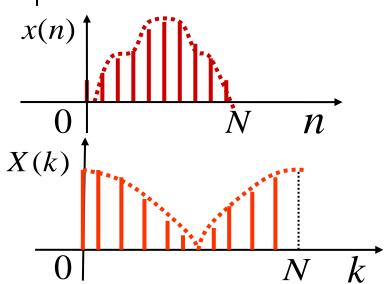
• x(n)、X(k) 分别称作 $\tilde{x}_N(n)$ 、 $\tilde{X}_N(k)$ 的主值

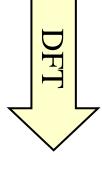
$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}nk} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi}{N}nk}$$
IDFT

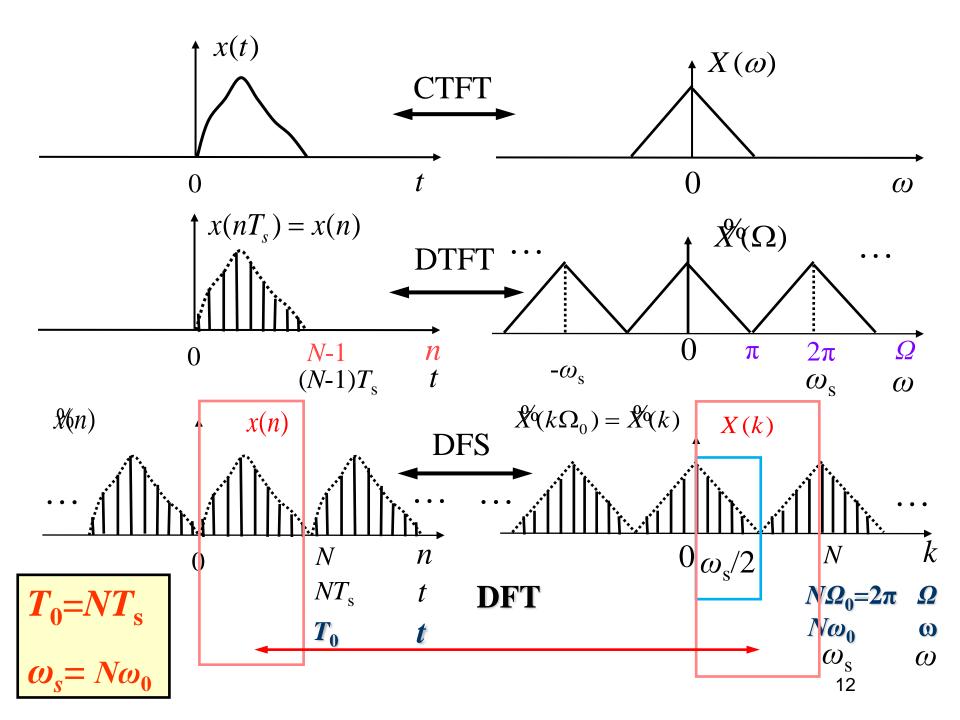


$$0 \le n \le N - 1$$

$$0 \le k \le N - 1$$





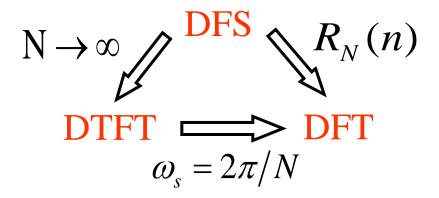


DFT小结





- ⇒ DFT 是 DFS 的主值序列
- ⇒ DFS 是严格按傅立叶分析的概念得来的
- ⇒ DFT 只是一种借用形式,一种算法
- → 用DFT 计算信号的频谱时,采样频率必须大于两倍的信号最高截止频率,对周期信号至少要取一个整周期



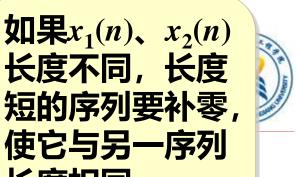
3、DFT的性质





- ⇒线性
- ⇒ 对称性
- ⇒ 圆周位移
- ⇒ 圆周卷积

线性



长度相同

⇒若

$$x_1(n) \stackrel{DFT}{\longleftrightarrow} X_1(k)$$

$$x_2(n) \stackrel{DFT}{\longleftrightarrow} X_2(k)$$

⇒ 那么

$$ax_1(n) + bx_2(n) \stackrel{DFT}{\longleftrightarrow} aX_1(k) + bX_2(k)$$

圆周移位的概念

有限长序列

$$\chi(n)$$
 $0 \le n \le N-1$

周期延拓

$$\mathcal{X}_{\mathcal{N}}(n)$$

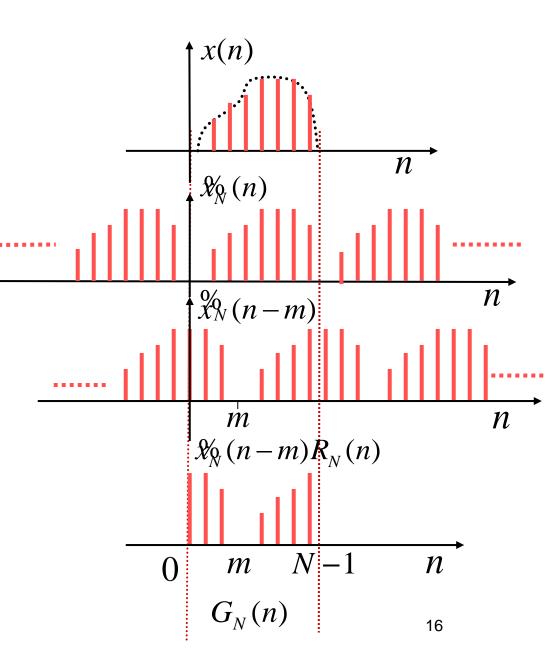
线性位移

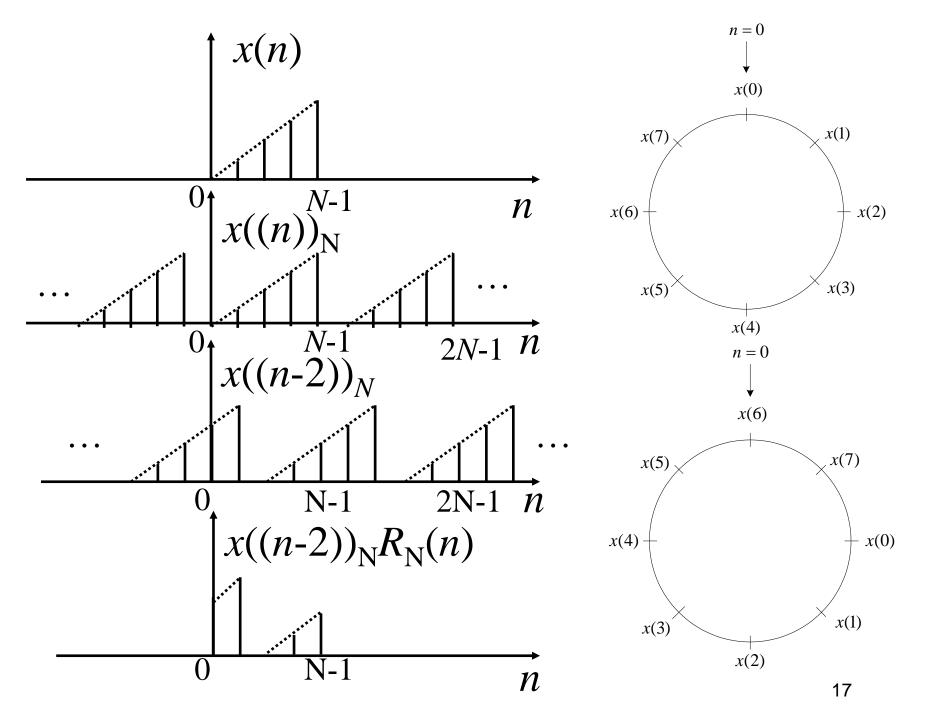
$$\chi_N^{0}(n-m)$$

加窗

得到圆周位移序列

$$\mathcal{K}_N(n-m)R_N(n)$$





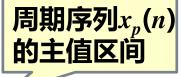




例: 已知 $x_p(n)$ 是周期 N=5 的周期序列,则

$$x_p(19) = x((19))_N = ? \quad x_p(-2) = x((-2))_N = ?$$

周期序列
$$x_p(n)$$
的
$$x(n_1) = \begin{cases} x_p(n_1) & 0 \le n_1 \le N-1 \\ 0 &$$
其他



$$x_p(rN + n_1) = x_p(n_1) = x(n_1) \quad 0 \le n_1 \le N - 1$$

$$x((n-m))_{N} = \sum_{r=-\infty}^{\infty} x(n-m+rN)$$





例: 有N=5有限长序列 $x(n) = \{1,2,3,4,5\}$

求 $x((n))_5$, $x((n))_4$, $x((n))_6 R_6(n)$, $x((n+3))_5 R_5(n)$

$$x((n))_5 = \{\cdots 4, 5, \underline{1}, 2, 3, 4, 5, 1, 2, 3, 4, 5, \cdots\}$$

$$x((n))_4 = \sum_{r=0}^{\infty} x(n+4r) = \{\cdots, 2, 3, 4, \underline{6}, 2, 3, 4, 6, 2, \cdots\}$$

r=-∞ 周期延拓点数 N1 小于有限长序列点数 N 时? 1, 2, 3, 4, 5





例: 有N=5有限长序列 $x(n) = \{1,2,3,4,5\}$

求 $x((n))_5$, $x((n))_4$, $x((n))_6 R_6(n)$, $x((n+3))_5 R_5(n)$

$$x((n))_6 = \sum_{r=-\infty}^{\infty} x(n+6r) = \{?\cdot, 4, 5, 0, \underline{1}, 2, 3, 4, 5, 0, ?\cdot\}$$

$$x((n))_6 R_6(n) = \{\underline{1}, 2, 3, 4, 5, 0\}$$

周期延拓点数 N1 大于有限长序列点数 N 时?

1, 2, 3, 4, 5, 0

1, 2, 3, 4, 5, 0

 \cdots , 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, \cdots





例: 有N=5有限长序列 $x(n) = \{1,2,3,4,5\}$

求 $x((n))_5$, $x((n))_4$, $x((n))_6 R_6(n)$, $x((n+3))_5 R_5(n)$

$$x((n+3))_5 = \sum_{r=-\infty}^{\infty} x(n+3+5r) = \{?\cdot, 4, 5, 1, 2, 3, 4, 5, ?\cdot\}$$

$$x((n+3))_5 R_5(n) = \{1, 2, 3, \underline{4}, 5\}$$

练习: $x((-n+2))_4 R_4(n)$

时移特性





22

⇒若

$$DFT[x(n)] = X(k)$$
$$y(n) = \tilde{x}_N(n - m)R_N(n)$$

⇒则

$$DFT[y(n)] = e^{-j\Omega_0 mk} X(k)$$

- \Rightarrow 时域序列的圆周位移的 **DFT** 为原来的 **DFT** 乘 以一个因子 $e^{-j\Omega_0mk}$
- ⇒ 有限长序列的圆周移位,在离散频域中移入和频率成正比的线性相移,对频谱的幅度没有影响。

频移特性





⇒若

$$DFT[x(n)] = X(k)$$

⇒则

$$Y(k) = X_N^0(k-l)R_N(n)$$

$$IDFT[Y(k)] = x(n)e^{j\Omega_0 ln}$$

⇒ IDFT 在时域 x(n) 乘以一个 $e^{j\Omega_0 ln}$

时域圆周卷积定理





⇒若

$$Y(k) = X(k)H(k)$$

⇒ 则

$$y(n) = x(n) \otimes h(n)$$

$$x(n) \otimes h(n) = \sum_{m=0}^{N-1} x(m) h_N^{0}(n-m) R_N(n)$$

定义为 圆周卷积

$$f(x(n) \otimes h(n) = \sum_{m=0}^{N-1} h(m) \mathcal{X}_{N}(n-m) R_{N}(n)$$

x(n)和h(n)都 需是N点 x(n) \mathbb{N} h(n)

N点的 圆周卷积

频域圆周卷积定理





⇒若

$$y(n) = x(n)h(n)$$

⇒则

$$Y(k) = DFT[y(n)]$$

$$= \frac{1}{N} \sum_{l=0}^{N-1} X(l) \tilde{H}_{N}(k-l) R_{N}(k)$$

$$= \frac{1}{N} \sum_{l=0}^{N-1} H(l) \tilde{X}_{N}(k-l) R_{N}(k)$$

例 计算矩形脉冲序列 $x_1(n)$ 、 $x_2(n)$ 的 N 点圆周

乘积。
$$x_1(n) = x_2(n) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & 其他 \end{cases}$$

解: $x_1(n)$ 、 $x_2(n)$ 的 N 点 DFT 为

$$X_1(k) = X_2(k) = \sum_{n=0}^{N-1} W_N^{nk} = \begin{cases} N & k = 0 \\ 0 & \sharp \text{ th} \end{cases}$$

因此,有

$$X(k) = X_1(k)X_2(k) == \begin{cases} N^2 & k = 0 \\ 0 &$$
其他

 $x_1(n)$ 、 $x_2(n)$ 的 N 点圆周卷积是 X(k) 的反 **DFT** 变换

$$x(n) = \begin{cases} N & 0 \le n \le N - 1 \\ 0 &$$
其他

DFT





$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jnk\Omega_0}$$

$$= \sum_{n=0}^{N-1} x(n)e^{-jnk\frac{2\pi}{N}} = \sum_{n=0}^{N-1} x(n)W_N^{nk}$$

W_N 指数/旋转/加权 因子

$$W_N^{nk}=e^{-\mathrm{j}\Omega_0nk}=e^{-\mathrm{j}rac{2\pi}{N}nk}$$

(1) 周期性

$$W_N^0 = 1$$
, $W_N^N = W_N^0 = 1$, $W_N^{mN} = 1$

$$W_N^{r+mN} = W_N^r, W_N^{N/4} = e^{-j\frac{2\pi}{N} \times \frac{N}{4}} = -j$$

$$W_{2N}^k = W_N^{k/2}$$

DFT

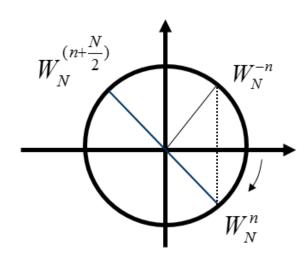




(2) 对称性

$$W_N^{\frac{N}{2}} = e^{-j\frac{2\pi N}{N}} = e^{-j\pi} = -1, \ W_N^{(mN+\frac{N}{2})} = -1$$

$$W_N^{r+\frac{N}{2}} = -W_N^r$$



回顾





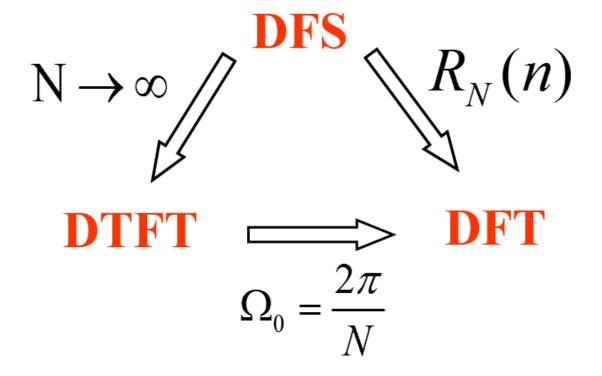
	时域	频域
DFS	离散、周期	离散、周期
DTFT	离散、非周期	连续、周期
DFT	离散、非周期	离散、非周期

DFT具有时域和频域均离散化的特点,且可用于处理时间有限的离散信号(周期与非周期均可),是借用DFS形式/DTFT离散化的算法。

回顾







数字实验二





实验三 离散傅立叶变换

一、实验目的

- 1. 掌握 DFT 变换
- 2. 掌握 DFT 性质
- 3. 掌握利用 DFT 计算线性卷积
- 4. 掌握快速傅立叶变换 (FFT)

二、实验内容

1. 求以下有限长离散时间信号的离散时间傅立叶变换 $X(e^{j\Omega})$

•
$$\exists \exists x(n) = 2^n$$
 $-10 \le n \le 10$

四、快速傅立叶变换 (FFT)





- **⇒** DFT 的计算
- **⇒** FFT

DFT的运算

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \cdot R_N(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \cdot R_N(k)$$

$$| \mathbf{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}(k) e^{j\frac{2\pi}{N}kn} \cdot R_N(n) = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}(k) W_N^{-kn} \cdot R_N(n)$$

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^{1 \times 1} & W_N^{2 \times 1} & \cdots & W_N^{(N-1) \times 1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ W_N^0 & W_N^{1 \times (N-1)} & W_N^{2 \times (N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^{-1 \times 1} & W_N^{-2 \times 1} & \cdots & W_N^{-(N-1) \times 1} \\ W_N^0 & W_N^{-(N-1) \times 1} & W_N^{-(N-1) \times 2} & \cdots & W_N^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

DFT





$\Rightarrow W_N = e^{-j\Omega_0} = e^{-j2\pi/N}$

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jnk\Omega_0}$$

$$= \sum_{n=0}^{N-1} x(n)e^{-jnk\frac{2\pi}{N}} = \sum_{n=0}^{N-1} x(n)W_N^{nk}$$

W_N 指数/旋转/加权 因子

$$W_N^{nk} = e^{-j\Omega_0 nk} = e^{-j\frac{2\pi}{N}nk}$$

(1) 周期性

$$W_N^0 = 1$$
, $W_N^N = W_N^0 = 1$, $W_N^{mN} = 1$

$$W_N^{r+mN} = W_N^r, W_N^{N/4} = e^{-j\frac{2\pi}{N} \times \frac{N}{4}} = -j$$

$$W_{2N}^k = W_N^{k/2}$$

DFT

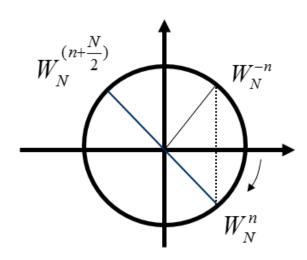




(2) 对称性

$$W_N^{\frac{N}{2}} = e^{-j\frac{2\pi N}{N}\frac{N}{2}} = e^{-j\pi} = -1, \ W_N^{(mN+\frac{N}{2})} = -1$$

$$W_N^{r+\frac{N}{2}} = -W_N^r$$

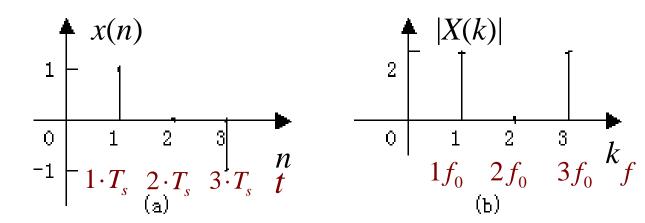


例:设有正弦信号 $x(t) = \sin(\omega_0 t) = \sin(2\pi f_0 t)$,对其一周内

采样N点得到正弦序列 $x(n) = \sin(\omega_0 n T_s) = \sin(2\pi \frac{f_0}{f_s} n)$,若

$$f_s = 4f_0$$
(即一周内采样4点),于是 $x(n) = \sin(\frac{2\pi}{4}n)$ 。

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -2j \\ 0 \\ 2j \end{bmatrix}$$



快速傅立叶变换





- N点DFT的计算量:
 - 每计算一个 X(k) 值需要进行 N 次复数相乘, N-1 次复数相加;
 - 对于 $N \cap X(k)$ 点,完成全部**DFT**运算共需 \mathbb{N}^2 次复数相乘和 N(N-1) 次复数加法。
 - 一次复数乘(a+jb)(c+jd)=(ac-bd)+j(ad+bc)共需 2 次实数 加法和 4 次实数乘法。
 - 一次复数加a+jb+c+jd=(a+c)+j(b+d)共需 2 次实数加法。
 - 所以完成整个DFT运算需要 $4N^2$ 次实数乘法和 $2N^2 + 2N(N-1)\approx 4N^2$ 次实数加法。

当 N 较大时,所需的运算工作量相当大。例如当 $N = 2^{10} = 1024$ 时,总共约需 400 万次实数乘法运算,对于实时信号处理来说,将对计算速度有十分苛刻的要求。

基2FFT算法





主要思路: 将 x(n) 或 X(k) 的次序重排,并利用 W_N 的特性,将长序列的离散傅里叶变换运算逐次分解成较短序列的离散傅里叶变换计算,从而提高运算效率。

若算法是对输入序列 x(n) 进行逐次分解进行的,则叫做**时间抽选**(Decimation In Time, DIT)FFT**算法。**

若序列 x(n) 的长度 $N=2^{\beta}$, β 是整数(可以对序列增补零值点来达到)。则通过分解,其最小的DFT运算单元是2点。通常将FFT运算中最小DFT运算单元称为**基**(radix),因而把这种算法称为**基2时间抽选FFT算法**。

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x(n)W_N^{nk} + \sum_{n=0}^{N-1} x(n)W_N^{nk}$$

$$= \sum_{n=0}^{N-1} x(2r)W_N^{2r.k} + \sum_{n=0}^{N-1} x(2r+1)W_N^{(2r+1)k}$$

$$= \sum_{n=0}^{N-1} x(2r)(W_N^2)^{r.k} + \sum_{n=0}^{N-1} x(2r+1)(W_N^2)^{r.k}W_N^k$$

$$= \sum_{n=0}^{N-1} x(2r)W_N^{rk} + W_N^k \sum_{n=0}^{N-1} x(2r+1)W_N^{rk} = G(k) + W_N^k H(k)$$

x(n)的序号为偶数的序列 x(n)的序号为奇数的序列

G(k)、H(k)符合DFT的定义,它们是以 N/2 为周期的序列

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_{\frac{N}{2}}^{rk} + W_{N}^{k} \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_{\frac{N}{2}}^{rk} \qquad (0 \le k \le N-1)$$
$$= G(k) + W_{N}^{k} H(k)$$

(3)
$$X(k) = G(k) + W_N^k H(k)$$





$$X(0)=G(0) + W_8^0 H(0)$$

$$X(1)=G(1) + W_8^1 H(1)$$

$$X(2)=G(2) + W_8^2 H(2)$$

$$X(3)=G(3) + W_8^3 H(3)$$

$$X(4)=G(4) + W_8^4 H(4) = G(0) - W_8^0 H(0)$$

$$X(5)=G(5) + W_8^5 H(5) = G(1) - W_8^1 H(1)$$

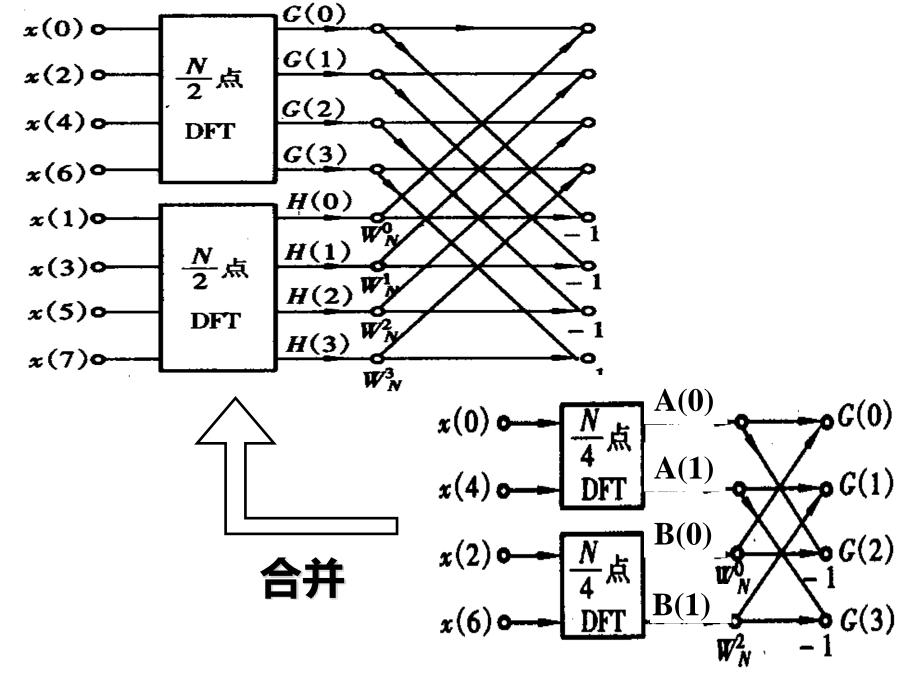
$$X(6)=G(6) + W_8^6 H(6) = G(2) - W_8^2 H(2)$$

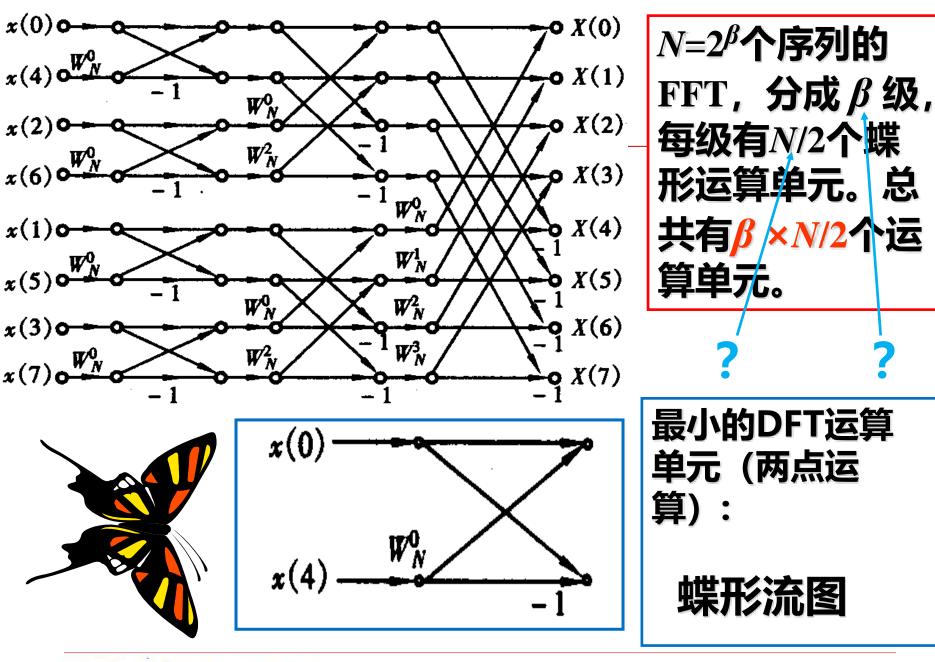
$$X(7)=G(7) + W_8' H(7) = G(3) - W_8^3 H(3)$$

$$W_N^k = W_N^{(k+rN)}$$
 $W_N^{rk} = W_{\frac{N}{r}}^k$ $W_N^k = -W_N^{(k+\frac{N}{2})}$

 $X(k) = G(k) + W_N^k H(k)$ (k=0...N-1) G(0) $X(0) = G(0) + W_N^0 H(0)$ x(0) o G(1) $o X(1) = G(1) + W_N^1 H(1)$ x(2) \circ G(2) $\alpha X(2) = G(2) + W_N^2 H(2)$ x(4)o**DFT** G(3) $X(3) = G(3) + W_N^3 H(3)$ x(6) \circ H(0) $X(4) = G(0) - W_N^0 H(0)$ x(1)o H(1) $x(5) = G(1) - W_N^1 H(1)$ x(3)o $H(2)^{\mathbf{W}_{3}}$ $\sum_{0} X(6) = G(2) - W_N^2 H(2)$ x(5)o **DFT** H(3) $\lambda_{o} X(7) = G(3) - W_{N}^{3}H(3)$ x(7) \circ W_N^3

图3.15 N点的DFT计算分解成两个N/2点的DFT计算的时间抽选流程图 (N=8)





FFT的基本思想





⇒ 把原始的N点序列,依次分解成一系列短序列。同时充分利用DFT计算式中所具有的对称性和周期性,求出这些短序列相应的DFT,并进行适当组合,最终达到删除重复运算,减少乘法运算,提高速度的目的。

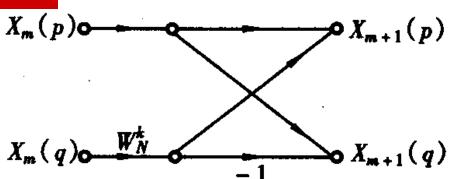
DIT FFT算法的规律和特点





(1) 蝶形运算

$$\begin{cases} X_{m+1}(p) = X_m(p) + W_N^k X_m(q) \\ X_{m+1}(q) = X_m(p) - W_N^k X_m(q) \end{cases} X_m(q) \bullet W_N^k$$



一次蝶形运算需要1次复数乘,2次复数加运算。

复数乘法总数: $m_c = \frac{N}{2}\beta = \frac{N}{2}\log_2 N$

复数加法总数: $a_c = 2(\frac{N}{2}\beta) = N\log_2 N$

比较:完成全部 N 点 X(n) 计算,DFT需要 \mathbb{N}^2 次复数乘法和 N(N-1) 次复数加法。

直接DFT与DIT FFT所需复数乘法次数的比较

β	N	直接 DFT(N²)	$\operatorname{FFT}\left(\frac{N}{2}\log_2 N\right)$	改善比值 $\left(\frac{2N}{\log_2 N}\right)$
1	2	4	1	4
2	4	16	4	4
3	8	64	12	5.3
4	16	256	32	8
5	32	1 024	80	12.8
6	64	4 096	192	21.3
7	128	16 384	448	36.6
8	256	65 536	1 024	64
9	512	262 144	2 304	113.8
10	1 024	1 048 576	5 120	204.8
11	2 048	4 194 304	11 264	372.4

例: 如果一台通用计算机的速度为平均每次复数乘要5μs,每次复数加需要0.5μs,用它来计算512点的**DFT**,问直接计算需要多少时间,用**FFT**运算需要多少时间。

解: (1) 直接计算

复乘所需时间: $T_1 = 5 \times 10^{-6} \times N^2 = 5 \times 10^{-6} 512^2 = 1.31072s$

复加所需时间:

$$T_2 = 0.5 \times 10^{-6} \times N \times (N-1) = 0.5 \times 10^{-6} \times 512 \times (512-1) = 0.130816s$$

$$T = T_1 + T_2 = 1.441536s$$

(2) FFT计算

复乘所需时间:

$$T_1 = 5 \times 10^{-6} \times \frac{N}{2} \log_2 N = 5 \times 10^{-6} \times \frac{512}{2} \times \log_2 512 = 0.01152s$$

复加所需时间:

$$T_2 = 0.5 \times 10^{-6} \times N \times \log_2 N = 0.5 \times 10^{-6} \times 512 \times \log_2 512 = 0.002304s$$

$$T = T_1 + T_2 = 0.013824s$$

如果采用专用**DSP**处理芯片**TMS320**系列,其 每次复数乘需400ns,每次复数加需要100ns,则 计算512点的**FFT**所需时间为:

复乘所需时间:

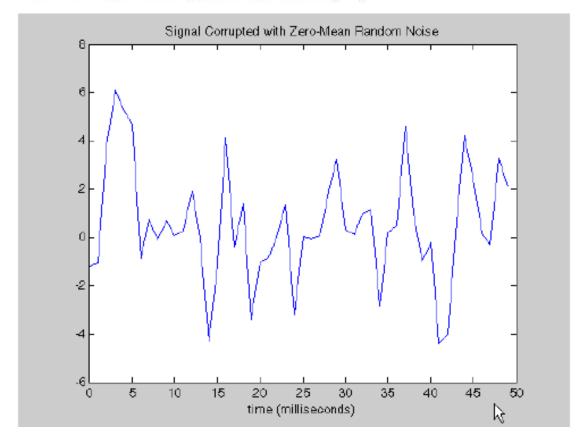
$$T_1 = 0.4 \times 10^{-6} \times \frac{N}{2} \log_2 N = 0.4 \times 10^{-6} \times \frac{512}{2} \times \log_2 512 = 0.0009216s$$

复加所需时间:

$$T_2 = 0.1 \times 10^{-6} \times N \times \log_2 N = 0.1 \times 10^{-6} \times 512 \times \log_2 512 = 0.0004608s$$

$$\therefore T = T_1 + T_2 = 0.0013824s$$

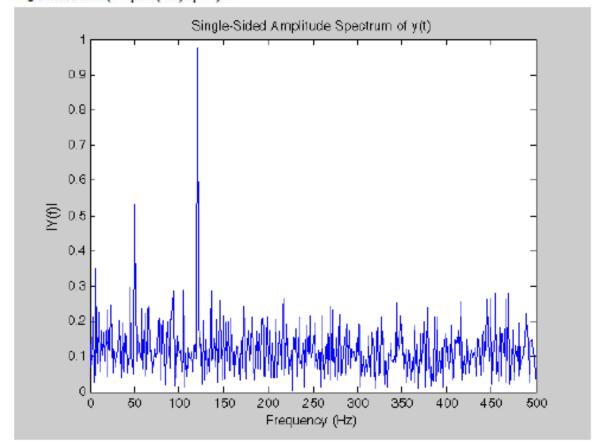
```
Fs = 1000; % Sampling frequency
T = 1/Fs; % Sample time
L = 1000; % Length of signal
t = (0:L-1)*T; % Time vector
% Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
x = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(size(t)); % Sinusoids plus noise
plot(Fs*t(1:50),y(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
```



Matlab实例: 输入(时域)信号= 两个频率分量+零均 值随机变量 L点采样

```
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
plot(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
```



Matlab实例:

输出(频域)信号 两个频率分量的幅 值并不严格等于原 设定值。 通过增加L(即较小 采样间隔),可以 减少频谱混叠现象。 **例:** 用FFT算法处理一幅N×N点的二维图像,如用每秒可做10万次复数乘法的计算机,当N=1024时,问需要多少时间(不考虑加法运算时间)?

解 当N=1024点时, FFT算法处理—幅二维图像所需复数乘法次数约为:

$$\frac{N^2}{2}\log_2 N^2 = 10N^2 \approx 10^7$$

DFT算法处理所需复数乘法次数约为:

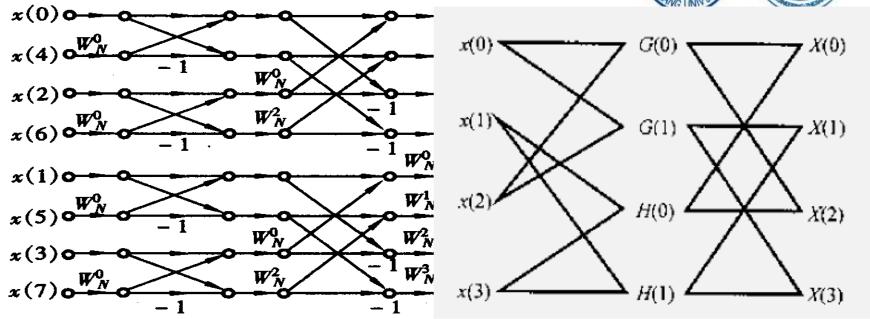
$$N^4 \approx 10^{12}$$

所以: 当N=1024点时,**FFT**算法处理一幅二维图像所需复数乘法仅为直接计算**DFT**所需时间的**10万分之一**。即原需要3000小时,现在只需要2分钟。

同址运算(原位运算)







在编制FFT算法时,当数据输入到存储器之后,每级运算结果仍然储存在原有的同一组存储器中,直到最后一级算完,中间不需要增设其它存储器(节省存储单元)。

每个蝶形运算需 2个序列存储单元和1个系数存储单元,每一列的 N/2个蝶形运算共需 (N+N/2) 个存储单元(也是整个 FFT 运算所需的存储单元)。

倒位序 (bit reversed order)



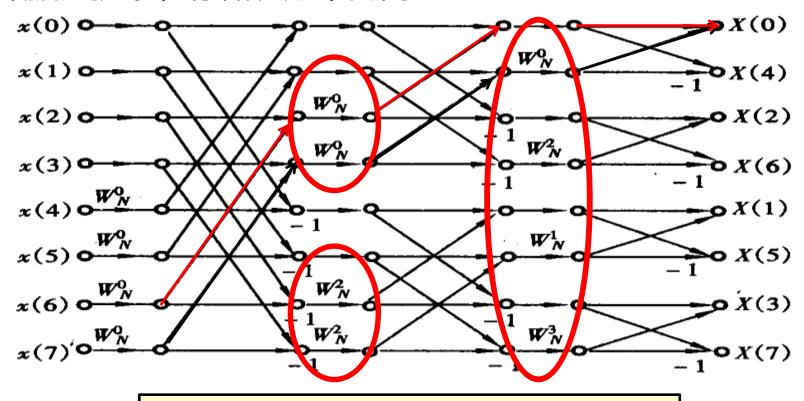


将序列 x(n) 码位倒置后作为FFT程序运算的输入。

$x(n) = x(2^2$	$n_2 + 2n$	$(n_1 + n_0)$)		N=	= 8 <u>,</u> 三	次奇值	禺分组			
I		п		Ш							
000	000		000	n ₂ =0	000	0					
001	010	n ₁ =0	100	n ₂ =1	100	4					
010	100	n ₁ =1	010	n ₂ =0	010	2	倒				
011 n₀=0	110		110	n ₂ =1	110	6	位				
100 n₀=1	001		001	n ₂ =0	001]	序				
101	011	n ₁ =0	101	n ₂ =1	101	5					
110	101	$n_i=1$	011	n ₂ =0	011	3					
111	111		111	n ₂ =1	111	7					
白然序			v(10) -	$-v(2^2)$		10 10	`				

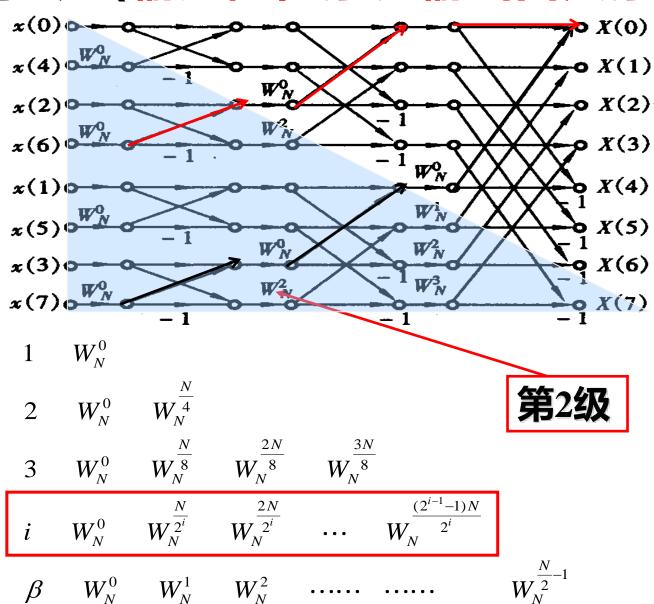
$$x(n) = x(2^2 n_0 + 2n_1 + n_2)$$

事实上,对于任何流程图,只要保持各节点所连支路及其传输系数不变,则不论节点位置怎样排列,所得到的流程图总是等效的,因而都能得到DFT的正确结果,只是数据的提取和存储次序不同而已。

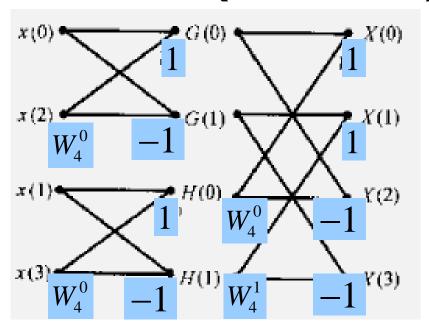


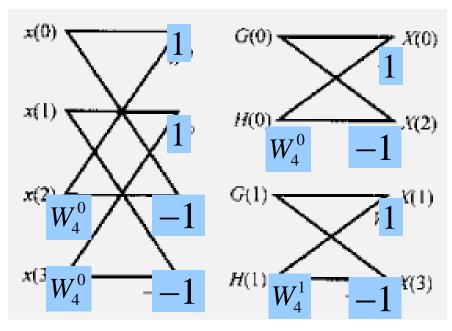
这是输入为自然序,输出为倒位序。

权系数 (输入倒位序、输出自然序)



权系数 (注意区别)





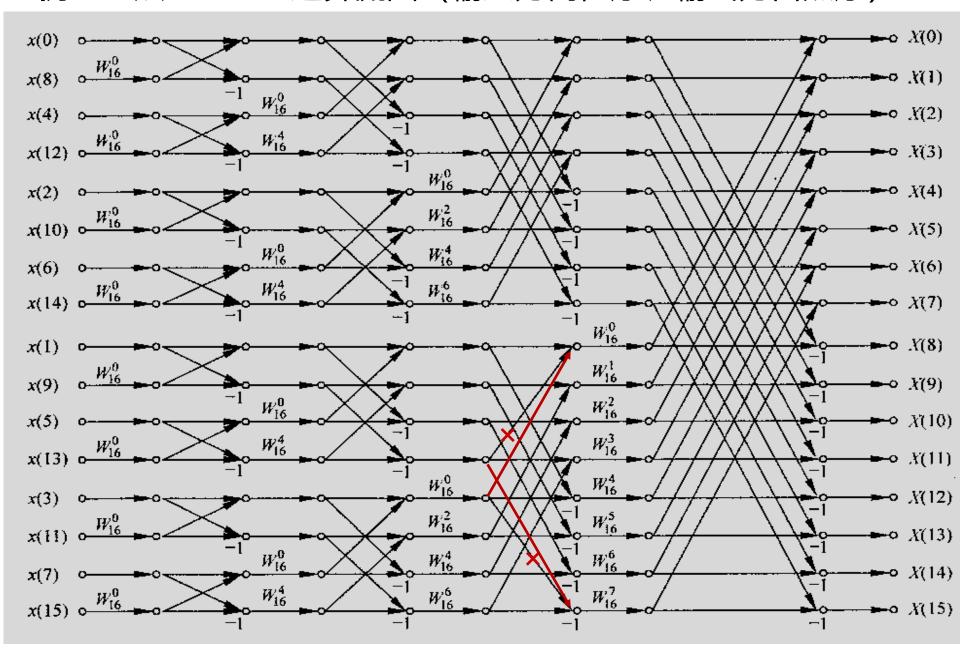
输入为倒位序,输出为自然序

输入为自然序,输出为倒位序

- 1 W_N^0
- $2 \qquad W_N^0 \qquad W_N^{\frac{N}{4}}$
- $3 \quad W_N^0 \quad W_N^{\frac{N}{8}} \quad W_N^{\frac{2N}{8}} \quad W_N^{\frac{3N}{8}}$
- $i \quad W_N^0 \quad W_N^{rac{N}{2^i}} \quad W_N^{rac{2N}{2^i}} \quad \cdots \quad W_N^{rac{(2^{i-1}-1)N}{2^i}}$
 - $\mathcal{G} = W_N^0 = W_N^1 = W_N^2 = \cdots$

 $W_N^{\frac{N}{2}-1}$

例: 16点DIT FFT运算流图 (输入为倒位序、输出为自然序)



例: 已知有限长序列: $x(n) = \{1,2,-1,3\}$ 试利用**DIT FFT**求 X(k)。





$$x(n) = \{1, 2, -1, 3\}$$

$$x(0) = 1$$

$$x(2) = -1$$

$$x(1) = 2$$

$$x(3) = 3$$

$$X(0) = 5$$

$$X(0) = 5$$

$$X(1) = 2 + j$$

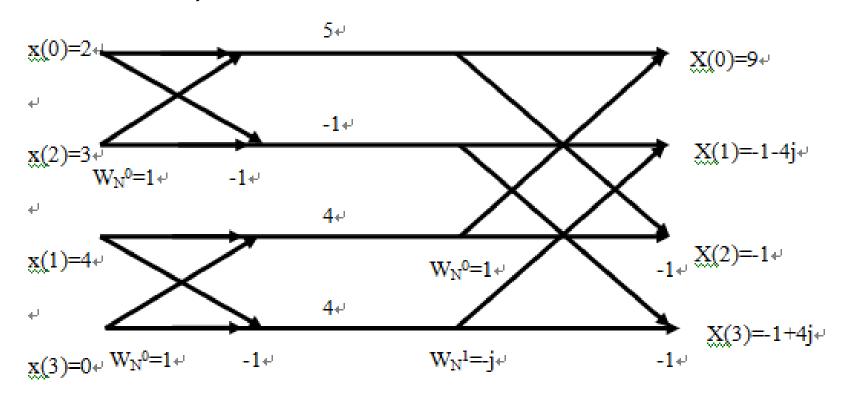
$$X(1) = 2 + j$$

$$X(2) = -5$$

$$X(3) = 3 -1$$

$$X(3) = 2 - j$$

2. (6分)已知有限长序列 x(n)=[2,4,3],试画图,利用基2时间抽选的**FFT**算法求 x(n)的DFT X(k)(要求补全运算流图,代入实际数据,在运算流图上给出计算的中间结果以及最终结果)。



FFT逆变换





FFT算法同样适用于求逆变换(IFFT),其差别在于:加权系数由 W_N^{nk} 改为 W_N^{-nk} ,并且运算结果乘以 1/N。

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \cdot R_N(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \cdot R_N(k)$$

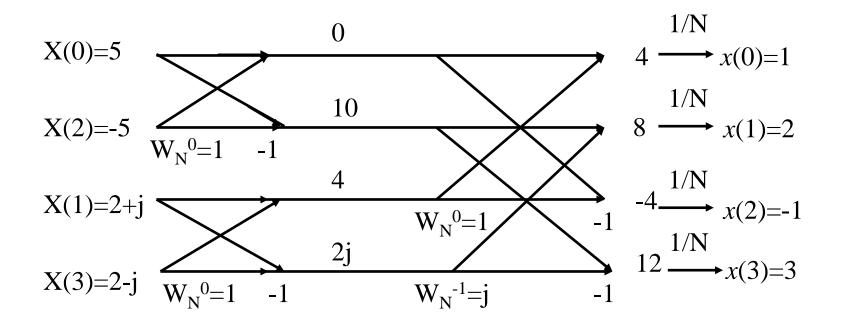
$$\mathbf{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}(k) e^{j\frac{2\pi}{N}kn} \cdot R_N(n) = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}(k) W_N^{-kn} \cdot R_N(n)$$

FFT逆变换





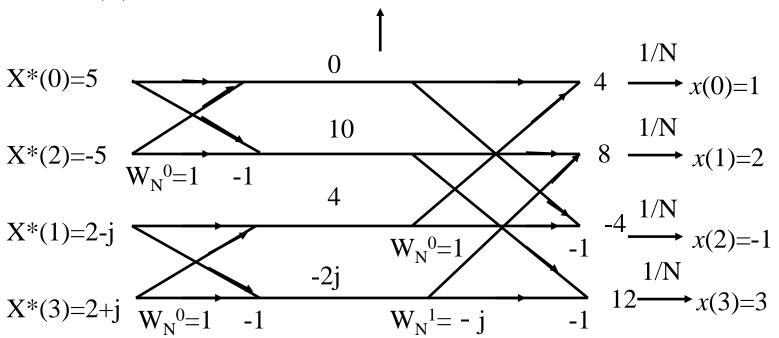
例: 已知有限长序列: $X(k) = \{5,2+j,-5,2-j\}$, 试利用**DIT IFFT**求 x(n)。



另有一种不改变FFT计算程序求IFFT的算法:

$$x^{*}(n) = \left[\frac{1}{N} \sum_{n=0}^{N-1} X(k) W_{N}^{-kn}\right]^{*} = \frac{1}{N} \sum_{n=0}^{N-1} X^{*}(k) W_{N}^{kn}$$
$$= \frac{1}{N} \text{DFT}[X^{*}(k)]$$

例: 已知有限长序列: $X(k)=\{5,2+j,-5,2-j\}$, 试利用**DIT IFFT**求 x(n)。



离散傅里叶变换对(DFT,IDFT)





$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\Omega_0 n} \qquad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jk\Omega_0 n}$$

$$\cdots = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \qquad = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn}$$

$$NX(k\Omega_0) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}$$

$$k, n = 0, 1, 2, 3, \dots, N-1$$

$$\Omega_0 = \frac{2\pi}{N} = \omega_0 T_s = \frac{2\pi}{T_0} T_s$$

基本频率 $f_0(Hz)$ 频率分辨率

 $f_s = Nf_0$ $T_0 = NT_s$ T_0 周期(采样记录长度), N点采样,形成有限长序列 $T_{\rm S}$ 采样间隔

指数/旋转/加权因子

周期性/对称性

$$W_N = e^{-j\frac{2\pi}{N}}$$

DFT(FFT)的应用 谱分析





- → DFT (FFT) 是一种方便计算机求解信号频谱的算法。 一般情况下,待研究的连续时间信号不具备离散性或周期性,也可能有无限长度。为使其适应FFT方法的需要,应对此波形进行抽样和截断,在此情况下,利用FFT计算的结果必然引入误差。下面按几种典型情况说明误差形成的原因。
 - 时限连续信号——先滤波后采样
 - ➤ 带限连续信号———选好"窗"
 - 连续周期信号——注意频率分辨率

65

时间有限信号 (频率无限信号)

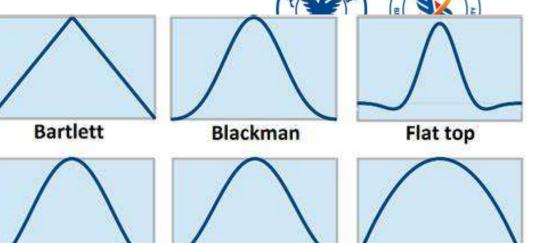




- ⇒ 待分析的信号只出现在某一确定时间之内,也即它是时间受限的。这时,其傅里叶变换不可能是有限带宽的,因此,根据采样定理可知,当时间函数进行采样以后,频谱必然造成混叠。也即,无论怎样减小采样间隔 T_s,混叠虽有可能减弱,但总是不可避免。
- ◆ 处理方法:在进行DFT处理之前,将信号通过一个反混叠滤波器(模拟低通滤波器),将模拟低通滤波器的截止频率设置为不大于抽样频率的一半。

频率有限信号 (时间无限信号)

- → <u>频率有限信号</u>必定是 理必须首先加窗得到
- 时域的截断造成频谱 形扩展开来,称为 "



Hamming

♪ 处理方法: 频谱泄漏不可避免,可以采取改变窗函数的形状(不用矩形窗),以改善频域拖尾、收敛慢的缺陷。例如: 升余弦窗[汉宁(Haning)窗]、改进的升余弦窗[汉明(Hamming)窗]、二阶升余弦窗[布莱克曼(Blackman)窗]等。

Hanning

Welch

频率有限信号 (时间无限信号)





例:复指数信号

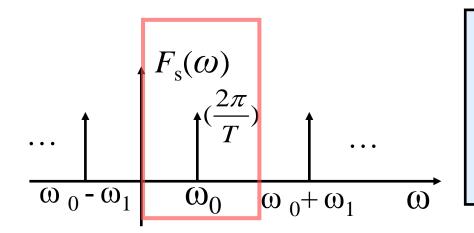
$$f(t) = e^{j\omega_0 t} \leftrightarrow F(\omega) = 2\pi\delta(\omega - \omega_0)$$

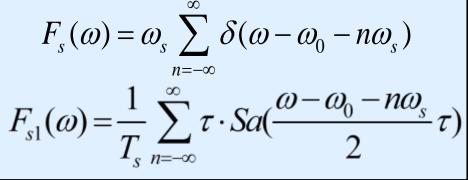
其理想抽样信号的频谱为

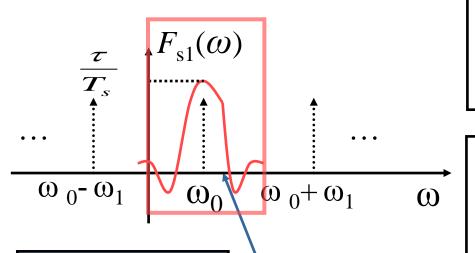
$$F_s(\omega) = \omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - \omega_0 - n\omega_s)$$

对抽样信号乘以一个门函数窗,得到有限长信号 $f_{s1}(t)$ 的频谱

$$F_{s1}(\omega) = \frac{1}{2\pi} F_s(\omega) * [\tau \cdot Sa(\frac{\omega \tau}{2})] = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \tau \cdot Sa(\frac{\omega - \omega_0 - n\omega_s}{2} \tau)$$







由于截断造成原信号的频谱扩展,产生泄漏现象。通常也出现混叠现象。

增加门函数 的宽度能改 善泄漏吗?

$$\omega_0 + \frac{2\pi}{\tau}$$

由于门函数在时域的突变造成 频域拖尾严重,收敛很慢。因 此改变窗函数的形状可以改善 泄漏现象(窗函数的频谱越接 近δ(ω)越好)。

频率分辨率





- \Rightarrow 离散傅里叶变换 X(k) 是对 $X(\Omega)$ 的采样,只给出频 谱在离散点 $\Omega = 2\pi k/N$ 上的值,而无法反映这些点 之间的内容,因此存在栅栏现象。有时谱峰正好位 于两个离散点之间,这样DFT就无法检测出此峰值。

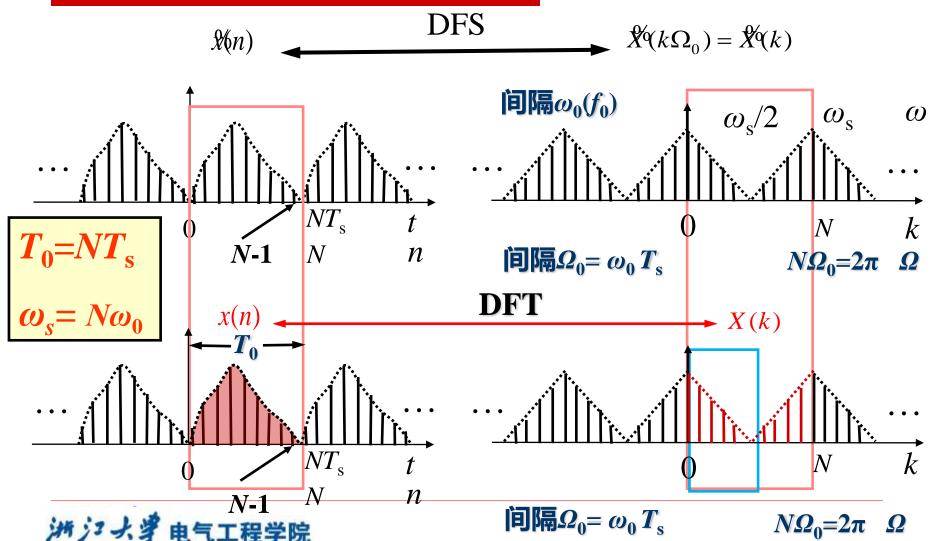
DFT的频率分辨率为
$$\Delta f = f_0 = \frac{1}{T_0} = \frac{f_s}{N}$$

分辨率越高,信号截取的宽度(数据长度 T_0)越长,即必 须增加计算点数 N (若抽样频率 f 不变)。如 N 固定, 则需减少采样频率 f_s (可能使混叠更加严重)。

频率分辨率







例:设 x(t) 的最高频率 f_m 不超过3Hz,现用 $f_s = 10$ Hz 对其抽样,数据宽度 T_0 取为25.6s,即抽样得到 x(n) 的点数为256。则我们对 x(n) 作**DFT**时,所得到的频率的最大分辨率:

$$\Delta f = \frac{f_s}{N} = \frac{10}{256} = 0.0390625$$
Hz

如信号x(t)由三个正弦组成, f_1 =2 Hz, f_2 =2.02 Hz, f_3 =2.04 Hz。即:

$$x(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t)$$

显然, f2 f3产生的正弦分量无法区分。

若数据宽度 T_0 取为102.4 s, 即抽样得到x(n)的点数为N=1024。

$$\Delta f = \frac{f_s}{N} = \frac{10}{1024} = 0.009765625 \text{ Hz}$$

显然,能正确区分各正弦分量。

利用DFT进行谱分析参数的 选择





- ⇒ 若已知信号的最高频率 f_m ,为防止混叠,选定抽样频率 f_s 满足 $f_s \ge 2f_m$ (或抽样周期 T_s 满足 $T_s \le 1/(2f_m)$)
- ◆ 根据要求确定频率分辨率 Δf , (即记录数据长度 T_0 [有时也写为 T_1] $T_0 = 1/\Delta f$)然后确定做DFT所要求的点数N, 即: $N = f_s/\Delta f = T_0/T_s$,N 的取值还要考虑FFT运算的要求,比如说取2的整数幂。
- ⇒ 根据 N 的最终取值重新选取抽样频率 f_s (或抽样 周期 T_s)

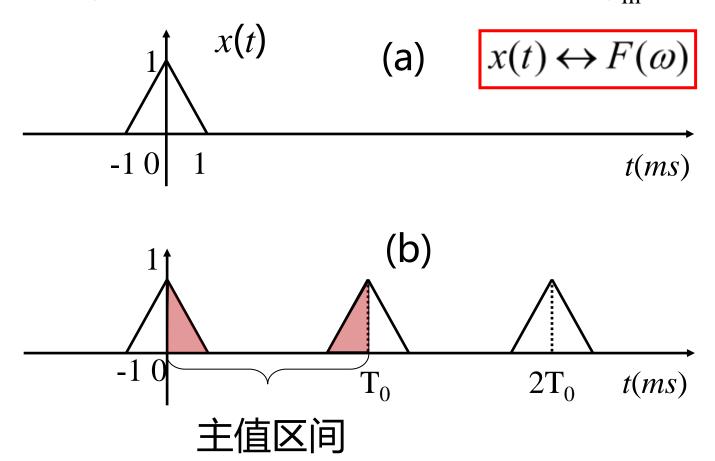
FFT应用中的注意事项





- ⇒ 信号离散时, 采样频率要满足奈奎斯特频率
- ⇒ 对于基2FFT算法, N一定是2的整数次幂, 若不是, 要补若干个零, 凑成2的整数次幂。
- ⇒ 数据长度要取得足够长

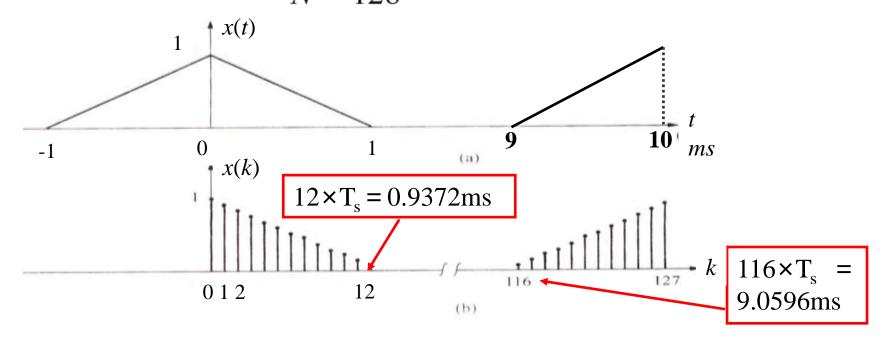
例2: 用**DFT**计算图(a)所示信号的频谱,要求频率分辨率 $\Delta f=100$ Hz,已知信号的最高频率 $f_m=4$ kHz



$$T_s \le \frac{1}{2f_m} = \frac{1}{2 \times 4 \times 10^3} = 0.125(ms)$$

数据长度
$$T_0 = \frac{1}{\Delta f} = \frac{1}{100} = 0.01(s) = 10(ms)$$

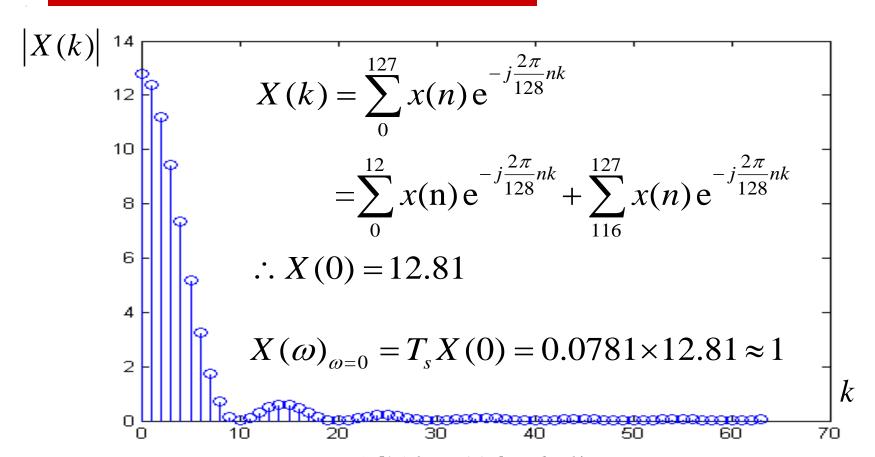
抽样点数
$$N \ge \frac{T_0}{T_s} = \frac{10}{0.125} = 80$$
 取 $N = 2^{\beta}$, $N = 128$. 修正 T_s ,
$$T_s = \frac{T_0}{N} = \frac{10}{128} = 0.0781 (ms)$$



续例2







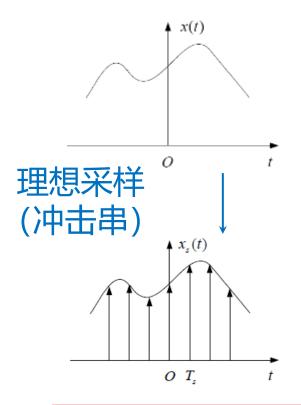
所求信号的幅度谱

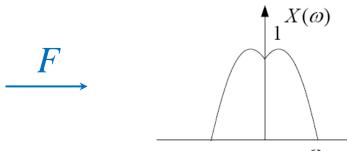
乘 T_s的缘由 (频谱幅值变化)

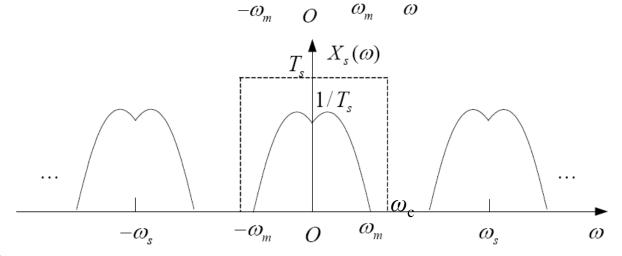




$$X_{s}(\omega) = \frac{1}{T_{s}} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_{s})$$







频谱幅值变化的一般处理办法

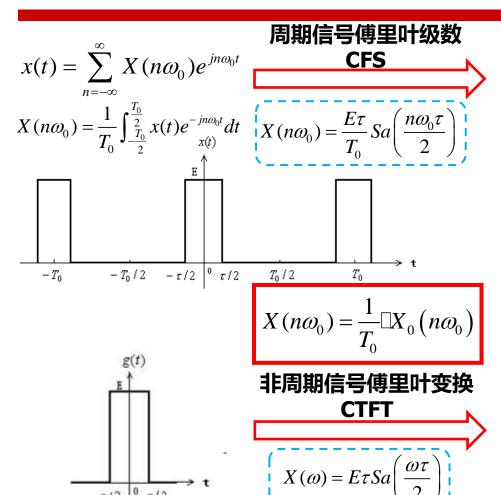


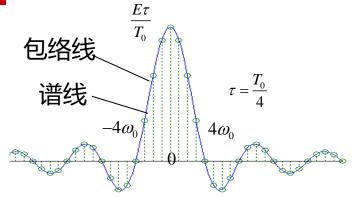
- \Rightarrow 对<mark>时间有限连续信号</mark>进行傅里叶分析,按照 公式求DFT将计算结果乘以系数 T_s 。
- ⇒ 用DFT来求解<mark>连续周期函数</mark>的傅里叶级数近似式。此时,因子1/N可放在正变换公式中,这样无论正、拟DFT变换都可以直接表示所需结果,无需再乘转换系数。

$$\frac{\mathcal{N}_{N}(k\Omega_{0})}{N} = \frac{1}{N} \sum_{n=0}^{N-1} \mathcal{N}_{N}(n) e^{-jk\Omega_{0}n} \qquad X(k) = \sum_{n=0}^{N-1} x(n) W_{N}^{nk}$$

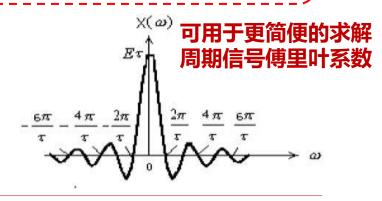
$$\mathcal{N}_{N}(n) = \sum_{k=0}^{N-1} \mathcal{N}_{N}(k\Omega_{0}) e^{jk\Omega_{0}n} \qquad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_{N}^{-kn}$$

求连续周期信号复傅里叶系数 $X(n\omega_0)$





周期脉冲的复傅里叶系数 等于单脉冲傅里叶变换在 $n\omega_0$ 频率点的值除以 T_0 。



- \tau/2 0 \tau/2