第四讲

离散傅立叶变换

《数字信号处理》第一部分

提纲

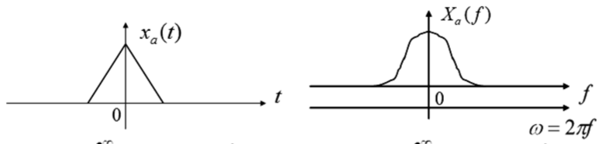
- 离散傅里叶变换(DFT)的导出
- 离散傅里叶变换(DFT)的性质
- 快速傅立叶变换
- 利用DFT计算线性卷积
- 利用DFT进行频谱分析

离散傅里叶变换(DFT)的导出

- 离散傅里叶变换(DFT)的导出
- 离散傅里叶变换(DFT)的性质
- 快速傅立叶变换
- 利用DFT计算线性卷积
- 利用DFT进行频谱分析

回顾

- 连续非周期信号的傅立叶变换(FT)



$$x_a(t) = \int_{-\infty}^{\infty} X_a(f) e^{j2\pi ft} df = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_a(\omega) e^{j\omega t} d\omega$$

$$X_a(f) = \int_{-\infty}^{\infty} x_a(t) e^{-j2\pi ft} dt$$

$$X_a(\omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\omega t} dt$$

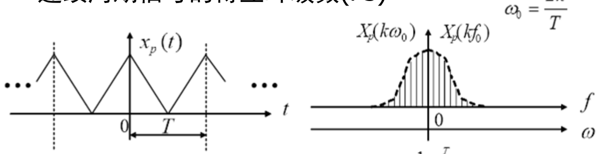
时域连续 ⟷ 频域非周期

时域非周期 ⟷ 频域连续

回顾

- 连续周期信号的傅立叶级数(FS)

$$\omega_0 = \frac{2\pi}{T}$$



$$x_p(t) = \sum_{k=-\infty}^{\infty} X_p(kf_0) e^{j2\pi kf_0 t} = \sum_{k=-\infty}^{\infty} X_p(k\omega_0) e^{jk\omega_0 t}$$

$$X_p(kf_0) = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x_p(t) e^{-j2\pi kf_0 t} dt$$

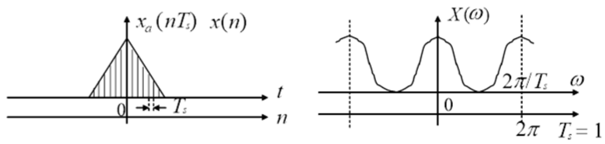
$$X_p(k\omega_0) = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x_p(t) e^{-jk\omega_0 t} dt$$

时域连续 ⟷ 频域非周期

时域周期 ⟷ 频域离散

回顾

- 离散时间信号的傅立叶变换(DTFT)



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

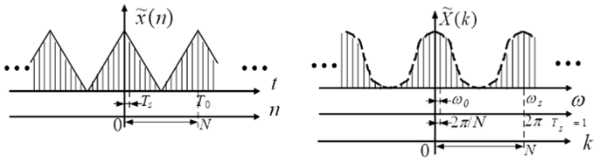
$$X_D(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

时域离散 ⟷ 频域周期

时域非周期 ⟷ 频域连续

拓展

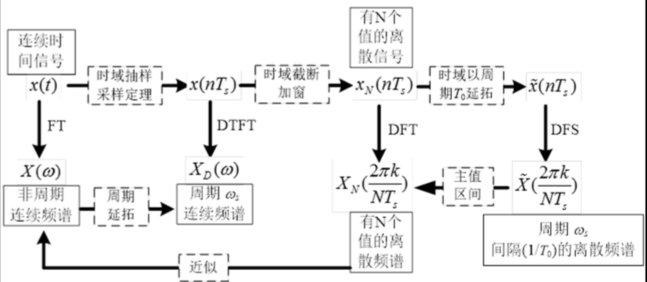
- 离散时间周期信号的频谱
  - 特点：时域离散、周期；频域离散、周期



$$\tilde{x}(n)=\tilde{x}(n+mN) \qquad \boxed{\begin{aligned}\omega_0&=\frac{2\pi}{T_0}\\ \omega_s&=\frac{2\pi}{T_s}\end{aligned}} \qquad \tilde{X}(k)=\tilde{X}(k+lN)$$

$$T_0=NT_s \qquad \omega_s=N\omega_0$$

离散傅里叶变换(DFT)的推导



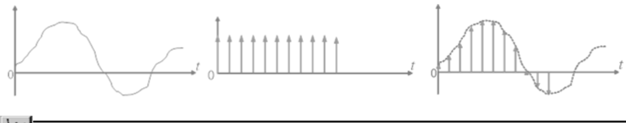
离散傅里叶变换(DFT)的推导

- 时域抽样
  - 在满足采样定理的前提下，取采样周期为$T_s=1/f_s$，因此用于抽样的周期脉冲序列可以表示为

$$p_1(t)=\Delta T_s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s)$$

  - 信号的抽样波形

$$x(t)\cdot p_1(t)=\sum_{n=-\infty}^{\infty}x(t)\cdot\delta(t-nT_s)=x(nT_s)$$



离散傅里叶变换(DFT)的推导

- 抽样脉冲序列$p_1(t)$的傅里叶变换
  - 抽样脉冲序列$p_1(t)$为周期T_s的周期序列，其可写为傅里叶级数

$$p_1(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s)=\frac{1}{T_s}\sum_{m=-\infty}^{\infty}e^{im\omega_s t}$$

  - 抽样脉冲序列$p_1(t)$的傅里叶变换

$$p_1(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s)=\frac{1}{T_s}\sum_{m=-\infty}^{\infty}e^{im\omega_s t}$$

$$\boxed{\int_{-\infty}^{\infty}e^{-i\omega t}\,\mathrm{d}t=2\pi\delta(\omega)} \Longrightarrow$$

$$\mathrm{FT}\left[p_1(t)\right]=P_1(\omega)=\frac{2\pi}{T_s}\sum_{m=-\infty}^{\infty}\delta(\omega\text{-}m\omega_s)$$

离散傅里叶变换(DFT)的推导

- 抽样信号$x(nT_s)$的傅里叶变换

$$\begin{aligned}\mathrm{FT}\left[x(nT_s)\right]&=X_D(\omega)\\ &=\mathrm{FT}\left[x(t)\cdot p_1(t)\right]=\frac{1}{2\pi}X(\omega)*P_1(\omega)=X(\omega)*\frac{1}{T_s}\sum_{m=-\infty}^{\infty}\delta(\omega\text{-}m\omega_s)\\ &=\frac{1}{T_s}\sum_{m=-\infty}^{\infty}X(\omega)*\delta(\omega\text{-}m\omega_s)=\frac{1}{T_s}\sum_{m=-\infty}^{\infty}\int_{-\infty}^{\infty}X(u)*\delta(\omega\text{-}m\omega_s-u)du\\ &=\frac{1}{T_s}\sum_{m=-\infty}^{\infty}X(\omega\text{-}m\omega_s)\end{aligned}$$

$\Uparrow$ 周期延拓

$$\mathrm{FT}\left[x(t)\right]=X(\omega)=\int_{-\infty}^{\infty}x(t)e^{-i\omega t}dt$$

离散傅里叶变换(DFT)的推导

- 时域截断
  - 信号截断处理等同于信号乘以一窗函数对信号进行截取
  - 不失一般性，我们定义窗函数的时间长度为$T_0=NT_s$，并假设使用矩形窗进行截断

$$w(t)=\begin{cases}1 & -T_s/2\le t<T_0--T_s/2\\ 0 & \text{others}\end{cases}$$

  - 窗函数$w(t)$与抽样信号$x(nT_s)$相乘，就可以把信号截断到区间$[-T_s/2, T_0\text{-}T_s/2\,)$内，而且$T_0$的选取使得第一周期的第$N$点和下一周期的第0点重合在一起

$$\begin{aligned}x_N(nT_s)&\triangleq x(nT_s)\cdot w(t)=x(t)\cdot p_1(t)\cdot w(t)\\ &=\left[\sum_{n=-\infty}^{\infty}x(t)\cdot\delta(t-nT_s)\right]\cdot w(t)=\sum_{n=0}^{N-1}x(t)\cdot\delta(t-nT_s)\end{aligned}$$

离散傅里叶变换(DFT)的推导

- 时域周期延拓
  - 经过时域离散、截断步骤后，信号在时域是：离散的，并且是时间受限的；在频域是：周期的，但是连续的
  - 考虑到周期信号的频谱离散特点，要得到离散频谱，只要是时域变成周期信号即可
  - 实现时把区间$[-T_s/2, T_0-T_s/2)$内函数按照周期T_0进行周期延拓，即把该区间内的波形平移nT_0 $(0\neq n\in Z)$后叠加形成
  - 周期延拓中的搬移可以通过与$\delta(t-nT_0)$的卷积来实现，因此周期延拓后的波形在数学上可以表示为原始波形与冲击串序列的卷积

$$\tilde{x}_N(nT_s)=x_N(nT_s)*p_2(t)$$

离散傅里叶变换(DFT)的推导

- 时域周期延拓
  - $x(nT_s)$以T_0为周期延拓后，根据周期信号频谱的特点，离散谱线间隔为$\omega_0=2\pi/T_0$，可设计频域冲激序列为

$$P_2(\omega)=\sum_{m=-\infty}^{\infty}\delta(\omega\text{-}m\omega_0)$$

$$p_1(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT_s)\underset{IFT}{\overset{FT}{\rightleftarrows}}P_1(\omega)=\frac{2\pi}{T_s}\sum_{m=-\infty}^{\infty}\delta(\omega\text{-}m\omega_s)$$

  - 频域冲激序列$P_2(\omega)$对应的时域信号

$$p_2(t)=\frac{T_0}{2\pi}\sum_{n=-\infty}^{\infty}\delta(t-nT_0)$$

离散傅里叶变换(DFT)的推导

- 时域周期延拓
  - $x_N(nT_s)$以T_0为周期延拓，等效于$x_N(nT_s)$与$p_2(t)$ 时域卷积

$$\begin{aligned}\tilde{x}_N(nT_s)&=x_N(nT_s)*p_2(t)\\&=\left[\sum_{n=0}^{N-1}x(t)\cdot\delta(t-nT_s)\right]*\left[\frac{T_0}{2\pi}\sum_{m=-\infty}^{\infty}\delta(t-mT_0)\right]\\&=\frac{T_0}{2\pi}\sum_{m=-\infty}^{\infty}\sum_{n=0}^{N-1}\left[\left(x(t)\cdot\delta(t-nT_s)\right)*\delta(t-mT_0)\right]\\&=\frac{T_0}{2\pi}\sum_{m=-\infty}^{\infty}\sum_{n=0}^{N-1}\left[x(t)\cdot\delta(t-nT_s-mT_0)\right]\end{aligned}$$

离散傅里叶变换(DFT)的推导

- 时域离散、截断和周期延拓后的信号频谱

$$\tilde{x}_N(nT_s)=\frac{T_0}{2\pi}\sum_{m=-\infty}^{\infty}\sum_{n=0}^{N-1}\left[x(t)\cdot\delta(t-nT_s-mT_0)\right]=\sum_{k=-\infty}^{\infty}a_ke^{ik\omega_0t}=\sum_{k=-\infty}^{\infty}a_ke^{ik2\pi t/T_0}$$

$$\boxed{\text{傅里叶级数展开}}$$

$$\begin{aligned}a_k&=\frac{1}{T_0}\int_{-T_s/2}^{T_0-T_s/2}\tilde{x}_N(nT_s)e^{-i2\pi kt/T_0}dt\\&=\frac{1}{T_0}\int_{-T_s/2}^{T_0-T_s/2}\left[\frac{T_0}{2\pi}\sum_{m=-\infty}^{\infty}\sum_{n=0}^{N-1}\left[x(t)\cdot\delta(t-nT_s-mT_0)\right]\right]e^{-i2\pi kt/T_0}dt\\&=\frac{1}{2\pi}\int_{-T_s/2}^{T_0-T_s/2}\sum_{n=0}^{N-1}x(t)\cdot\delta(t-nT_s)e^{-i2\pi kt/T_0}dt=\frac{1}{2\pi}\sum_{n=0}^{N-1}\int_{-T_s/2}^{T_0-T_s/2}x(t)e^{-i2\pi kt/T_0}\cdot\delta(t-nT_s)dt\\&=\frac{1}{2\pi}\sum_{n=0}^{N-1}x(nT_s)e^{-i2\pi knT_s/T_0}=\frac{1}{2\pi}\sum_{n=0}^{N-1}x(nT_s)e^{-i2\pi nk/N}\end{aligned}$$

离散傅里叶变换(DFT)的推导

- 时域离散、截断和周期延拓后的信号频谱

$$\tilde{x}_N(nT_s)=\sum_{k=-\infty}^{\infty}a_ke^{ik\omega_0t}=\sum_{k=-\infty}^{\infty}a_ke^{ik2\pi t/T_0}\qquad a_k=\frac{1}{2\pi}\sum_{n=0}^{N-1}x(nT_s)e^{-i2\pi nk/N}$$

$$\begin{aligned}\mathrm{FT}\left[\tilde{x}_N(nT_s)\right]&=\int_{-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty}a_ke^{ik\omega_0t}\right)e^{-i\omega t}dt\qquad\boxed{\text{傅里叶变换}}\\&=2\pi\sum_{k=-\infty}^{\infty}a_k\delta(\omega\text{-}k\omega_0)\\&=\sum_{k=-\infty}^{\infty}\left[\sum_{n=0}^{N-1}x(nT_s)e^{-i2\pi nk/N}\right]\delta(\omega\text{-}k\omega_0)\\&=\tilde{X}_N(k\omega_0)=\tilde{X}_N\left(\frac{2\pi k}{T_0}\right)=\tilde{X}_N\left(\frac{2\pi k}{NT_s}\right)\end{aligned}$$

离散傅里叶变换(DFT)的推导

- 时域离散、截断和周期延拓后的信号频谱

$$\begin{aligned}\mathrm{FT}\left[\tilde{x}_N(nT_s)\right]&=\sum_{k=-\infty}^{\infty}\left[\sum_{n=0}^{N-1}x(nT_s)e^{-i2\pi nk/N}\right]\delta(\omega\text{-}k\omega_0)\\&=\tilde{X}_N(k\omega_0)=\tilde{X}_N\left(\frac{2\pi k}{T_0}\right)=\tilde{X}_N\left(\frac{2\pi k}{NT_s}\right)\end{aligned}$$

  - 显然，$\tilde{x}_N(nT_s)$的频谱$\tilde{X}_N\left(\frac{2\pi k}{NT_s}\right)$是一个离散函数，它仅在离散频率点$\frac{2\pi k}{NT_s}$处取值，频率间隔为$\frac{2\pi}{NT_s}$，或$\frac{2\pi}{T_0}$，或$\omega_0$；
  - 同时由于$\tilde{X}_N\left(\frac{2\pi(k+N)}{NT_s}\right)=\tilde{X}_N\left(\frac{2\pi k}{NT_s}\right)$，则$\tilde{X}_N\left(\frac{2\pi k}{NT_s}\right)$是一个周期函数，周期为$N\omega_0=\omega_s$。

离散傅里叶变换及其逆变换的定义

■ DFT定义

- 设$x_N(nT_s)$是连续函数$x(t)$的N个抽样值，$n = 0,1,\ldots, N-1$，这N个点DFT定义为

$$\mathrm{DFT}\left(x_N(nT_s)\right)=\sum_{n=0}^{N-1} x_N(nT_s) e^{-i2\pi nk/N} \triangleq X_N\left(\frac{2\pi k}{NT_s}\right), \quad (k=0,1,\ldots,N-1)$$

■ IDFT定义

- 设$X_N(2\pi k/NT_s)$是连续频率函数$X(\omega)$的N个抽样值，$k = 0,1,\ldots, N-1$，这N个点IDFT定义为

$$x_N(nT_s) \overset{\mathrm{DFT}}{\Leftrightarrow} X_N\left(\frac{2\pi k}{NT_s}\right)$$

$$\mathrm{IDFT}\left(X_N\left(\frac{2\pi k}{NT_s}\right)\right)=\frac{1}{N}\sum_{k=0}^{N-1} X_N\left(\frac{2\pi k}{NT_s}\right) e^{i2\pi nk/N} \triangleq x_N(nT_s), \quad (n=0,1,\ldots,N-1)$$

离散傅里叶变换及其逆变换的定义

■ 复指数函数集$\{e^{jn\omega_0 t}, n \in Z\}$在$(t_0, t_0+T_0)$区间内是完备的正交函数集

$$\int_{t_0}^{t_0+T_0}\left(e^{jm\omega_0 t}\right)\left(e^{jn\omega_0 t}\right)^* dt = T_0\delta(m-n)$$

■ 复指数 $e^{-j2\pi nk/N}$ 称为DFT的变换核函数，具有如下正交性

$$\sum_{k=0}^{N-1} e^{-j2\pi kn/N}\left(e^{-j2\pi kr/N}\right)^* = \sum_{k=0}^{N-1} e^{-j2\pi kn/N} e^{j2\pi kr/N} = N\delta(n-r)$$

离散傅里叶变换及其逆变换的定义

■ 复指数 $e^{-j2\pi nk/N}$ 称为DFT的变换核函数，具有如下正交性

$$\sum_{k=0}^{N-1} e^{-j2\pi kn/N}\left(e^{-j2\pi kr/N}\right)^* = \sum_{k=0}^{N-1} e^{-j2\pi kn/N} e^{j2\pi kr/N} = N\delta(n-r)$$

- 如果$r = n$，显然有

$$\sum_{k=0}^{N-1} e^{-j2\pi kn/N}\left(e^{-j2\pi kn/N}\right)^* = \sum_{k=0}^{N-1} 1 = N$$

- 如果$r \neq n$，设$q = e^{j2\pi(r-n)/N}$，则在$0 \le r, n \le N-1$时，$q \neq 1$，从而有

$$\sum_{k=0}^{N-1} e^{-j2\pi kn/N} e^{j2\pi kr/N} = \sum_{k=0}^{N-1} q^k = \frac{1-q^N}{1-q} = 0$$

离散傅里叶变换及其逆变换的定义

■ 根据变换核函数的正交性，我们可以验证IDFT公式

$$\begin{aligned}\frac{1}{N}\sum_{k=0}^{N-1} X_N\left(\frac{2\pi k}{NT_s}\right) e^{i2\pi nk/N} &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{m=0}^{N-1} x_N(mT_s) e^{-i2\pi mk/N}\right] e^{i2\pi nk/N} \\ &= \frac{1}{N}\sum_{m=0}^{N-1} x_N(mT_s)\left[\sum_{k=0}^{N-1} e^{i2\pi nk/N} e^{-i2\pi mk/N}\right] \\ &= \frac{1}{N}\sum_{m=0}^{N-1} x_N(mT_s) N\delta(n-m) \\ &= x_N(nT_s)\end{aligned}$$

离散傅里叶变换及其逆变换的定义

■ 简记$W_N = e^{-j2\pi/N}$

- 核函数的正交性可表示为

$$\sum_{k=0}^{N-1} W_N^{kn}\left(W_N^{kr}\right)^* = N\delta(n-r)$$

- DFT可以表示为

$$X_N\left(\frac{2\pi k}{NT_s}\right) = \sum_{n=0}^{N-1} x_N(nT_s) W_N^{nk} \quad (k=0,1,\ldots,N-1)$$

- IDFT可以表示为

$$x_N(nT_s) = \frac{1}{N}\sum_{k=0}^{N-1} X_N\left(\frac{2\pi k}{NT_s}\right) W_N^{-nk}, \quad (n=0,1,\ldots,N-1)$$

DFT离散谱的性质

■ 通常称$X_N(2\pi k/NT_s)(k\in Z)$为离散序列$x_N(nT_s)$ $(0\le n<N)$的DFT离散谱；满足采样定理的条件下，通常简记$x_N(nT_s)$为$x(n)$，$X_N(2\pi k/NT_s)$简记为$X(k)$

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

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

$$x(n) \underset{\mathrm{IDFT}}{\overset{\mathrm{DFT}}{\rightleftharpoons}} X(k)$$

■ W_N的性质：关于k呈现一定的周期性和对称性

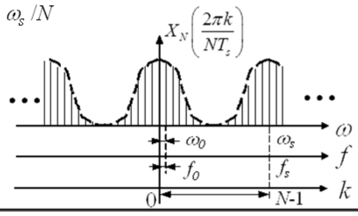
$$W_N^N = e^{-j2\pi} = 1 \qquad W_N^{N/2} = e^{-j\pi} = -1$$

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

$$\left|W_N^m\right| = 1\ (\forall m \in Z) \qquad W_{mN}^{mn} = e^{-j2\pi mn/mN} = e^{-j2\pi n/N} = W_N^n\ (\forall m, n \in Z)$$

DFT离散谱的性质

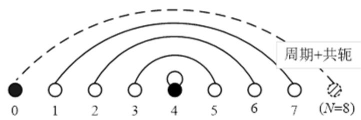
- 离散谱的周期性
  - DFT的定义是针对任意的离散序列$x_N(nT_s)$中的有限个离散抽样($0\le n<N$)，它并不要求该序列具有周期性。而由DFT求出的离散谱是离散的周期函数，周期为$N\omega_0 = 2\pi N/T_0 = 2\pi N/NT_s = 2\pi/T_s = \omega_s$，离散间隔为$\omega_0 = 2\pi/T_0 = 2\pi/NT_s = \omega_s/N$



DFT离散谱的性质

- 偶数点实序列离散谱的共轭对称性
  - 如果$x(n)$ ($0\le n<N$)为实序列，则其N点的DFT $X(k)$关于原点和$N/2$都具有共轭对称性

$$X(-k)=X^*(k) \qquad X(N-k)=X^*(k) \qquad X\left(\frac{N}{2}\pm k\right)=X^*\left(\frac{N}{2}\mp k\right)$$



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

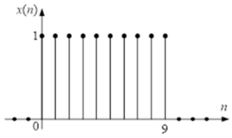
$$X\left(\frac{N}{2}\right)=\sum_{n=0}^{N-1}(-1)^n x(n)$$

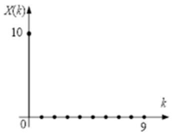
$$\|X(-k)\|=\|X(k)\| \qquad \|X(N-k)\|=\|X(k)\| \qquad \left\|X\left(\frac{N}{2}\pm k\right)\right\|=\left\|X\left(\frac{N}{2}\mp k\right)\right\|$$

DFT计算示例

- N点矩形序列 $x(n)$=1($0\le n\le N$-1)的N点DFT (取N=10)

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}=\sum_{n=0}^{N-1}W_N^{nk}=\begin{cases}\dfrac{1-W_N^{Nk}}{1-W_N^{k}}=0 & k\neq 0\\ \sum_{n=0}^{N-1}1=N & k=0\end{cases}, \qquad (k=0,1,\ldots,N-1)$$

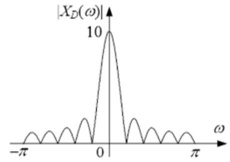


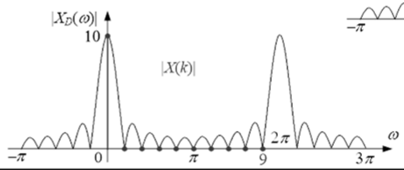


DFT计算示例

- N点矩形序列 $x(n)$=1($0\le n\le N$-1)的N点DFT (取N=10)
  - N=10点矩形序列的DTFT

$$X_D(\omega)=e^{-j(N-1)\omega/2}\frac{\sin(N\omega/2)}{\sin(\omega/2)}$$





DFT计算示例

- N点矩形序列 $x(n)$=1($0\le n\le N$-1)的2N点DFT (取N=10)
  - N点序列$x(n)$的后面N点用零填充

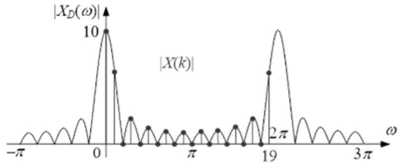
$$X(k)=\sum_{n=0}^{2N-1}x(n)W_{2N}^{nk}=\sum_{n=0}^{N-1}W_{2N}^{nk}=\sum_{n=0}^{N-1}e^{-j2\pi nk/2N}=\sum_{n=0}^{N-1}e^{-j\pi nk/N}$$

$$=\frac{1-\left(e^{-j\pi k/N}\right)^N}{1-e^{-j\pi k/N}}=\frac{1-e^{-j\pi k}}{1-e^{-j\pi k/N}}=\frac{e^{-j\pi k/2}\left(e^{j\pi k/2}-e^{-j\pi k/2}\right)}{e^{-j\pi k/2N}\left(e^{j\pi k/2N}-e^{-j\pi k/2N}\right)}$$

$$=e^{-j\pi(N-1)k/2N}\frac{\sin(\pi k/2)}{\sin(\pi k/2N)}, \qquad (k=0,1,\ldots,N-1)$$

DFT计算示例

- N点矩形序列 $x(n)$=1($0\le n\le N$-1)的2N点DFT (取N=10)
  - 10点序列$x(n)$ 20点DFT，与10点矩形序列的DTFT比较



离散傅里叶变换(DFT)的性质

- 离散傅里叶变换(DFT)的导出
- 离散傅里叶变换(DFT)的性质
- 快速傅立叶变换
- 利用DFT计算线性卷积
- 利用DFT进行频谱分析

离散傅里叶变换(DFT)的性质

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

$$X(k) = \mathrm{DFT}\left[x(n)\right] = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad 0 \le k \le N-1$$

$$x(n) = \mathrm{IDFT}\left[X(k)\right] = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-nk} \qquad 0 \le n \le N-1$$

- DFT的性质—线性 $\mathrm{DFT}[ax(n)+by(n)] = aX(k)+bY(k)$
  - 式中，a, b为任意常数
  - 若$x(n)$和$y(n)$的长度不等，分别为$N1$和$N2$，则$ax(n)+by(n)$的长度应为N=max[$N1$,$N2$]，故DFT必须按长度N计算，将长度短的补零后再做DFT.

离散傅里叶变换(DFT)的性质

- DFT的性质—正交性 $X(k) = \mathrm{DFT}\left[x(n)\right] = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad 0 \le k \le N-1$

$$\mathbf{W_N} = \left[W_N^{kn}\right] = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

$$\mathbf{X_N} = \left\{X(0), X(1), ..., X(N-1)\right\}^T \qquad \mathbf{x_N} = \left\{x(0), x(1), ..., x(N-1)\right\}^T$$

$$\mathbf{X_N} = \mathbf{W_N}\mathbf{x_N}$$

离散傅里叶变换(DFT)的性质

- DFT的性质—正交性

$$\mathbf{W_N^*}\mathbf{W_N} = [\sum_{k=0}^{N-1} W_N^{mk} W_N^{-nk}]_{N\times N}$$

$$\sum_{k=0}^{N-1} W_N^{mk} W_N^{-nk} = \sum_{k=0}^{N-1} W_N^{(m-n)k} = \begin{cases} N & m = n \\ 0 & m \ne n \end{cases}$$

或 $\mathbf{W_N^{-1}} = \frac{1}{N}\mathbf{W}_N^*$

$$\mathbf{X_N} = \mathbf{W_N}\mathbf{x_N} \longrightarrow \mathbf{x_N} = \mathbf{W_N^{-1}}\mathbf{X_N} = \frac{1}{N}\mathbf{W}_N^*\mathbf{X_N}$$

离散傅里叶变换(DFT)的性质

- 周期序列的主值区间与主值序列
  - 对于周期序列$\tilde{x}(n)$，定义其周期n=0~N-1，为$\tilde{x}(n)$的“主值区间”，主值区间上的序列为主值序列$x(n)$。

$$\begin{cases} \tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n+rN) \\ x(n) = \begin{cases} \tilde{x}(n) & 0 \le n \le N-1 \\ 0 & 其它n \end{cases} \end{cases} \longrightarrow \begin{cases} \tilde{x}(n)是x(n)的周期延拓 \\ x(n)是\tilde{x}(n)的"主值序列" \end{cases}$$

  - 数学表示

余数运算表达式

$$\begin{cases} \tilde{x}(n) = x((n))_N \\ x(n) = \tilde{x}(n)R_N(n) = x((n))_N R_N(n) \end{cases}$$

矩形序列

离散傅里叶变换(DFT)的性质

- 余数运算表达式
  - 如果$n = n_1 + mN$, $0 \leqq n_1 \leqq N\text{-}1$, m为整数，则有：

$$((n))_N = (n_1)$$

  - 此运算符表示n被N除，商为m，余数为n_1
  - (n_1)是$((n))_N$的解,或称作取余数
  - 例

$$n = 25, N = 9 \dashrightarrow \begin{array}{l} n = 25 = 2\times 9 + 7 = 2N + n_1 \\ ((25))_9 = (7) \end{array}$$

$$n = -4, N = 9 \dashrightarrow \begin{array}{l} n = -4 = -9 + 5 = -N + 5 \\ ((-4))_9 = (5) \end{array}$$

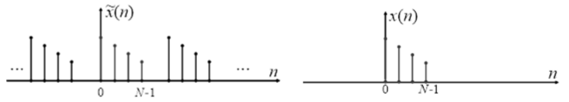
离散傅里叶变换(DFT)的性质

- 有限长序列$x(n)$和周期序列$\tilde{x}(n)$ 的关系
  - 周期序列$\tilde{x}(n)$是有限长序列$x(n)$的周期延拓

$$\tilde{x}(n)=\sum_{m=-\infty}^{\infty}x(n+mN)=x((n))_N$$

  - 有限长序列$x(n)$是周期序列$\tilde{x}(n)$的主值序列

$$x(n)=\tilde{x}(n)R_N(n)$$



离散傅里叶变换(DFT)的性质

- 频域上的主值区间与主值序列
  - 周期序列$\tilde{x}(n)$的离散傅立叶级数$\tilde{X}(k)$也是一个周期序列，也可给它定义一个主值区间 $0\le k\le N-1$，以及主值序列$X(k)$
  - 数学表示

$$\begin{cases}\tilde{X}(k)=X((k))_N\\ X(k)=\tilde{X}(k)R_N(k)\end{cases}$$

  - 周期序列$\tilde{X}(k)$是有限长序列$X(k)$的周期延拓
  - 有限长序列$X(k)$是周期序列$\tilde{X}(k)$的主值序列

离散傅里叶变换(DFT)的性质

- 循环移位 (圆周移位)

$$\boxed{y(n)=x((n+m))_N R_N(n)}$$

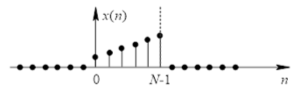
  - $x((n+m))_N$ 表示 $x(n)$ 的周期延拓序列$\tilde{x}(n)$左移m位

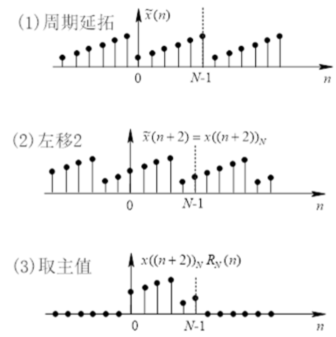
$$x((n+m))_N=\tilde{x}(n+m)$$

  - $x((n+m))_N R_N(n)$ 表示对移位的周期序列 $x((n+m))_N$ 取主值序列，所以$y(n)$仍然是一个长度为N的有限长序列。$y(n)$实际上可看作序列 $x(n)$排列在一个N等分圆周上，并顺时针旋转 m 位(右移对应逆时针)

离散傅里叶变换(DFT)的性质

- 循环移位示例
  - $x(n)$循环左移2位





离散傅里叶变换(DFT)的性质

- 循环移位运算 $\boxed{y(n)=x((n-m))_N R_N(n)}$
  - 循环移位的MATLAB代码

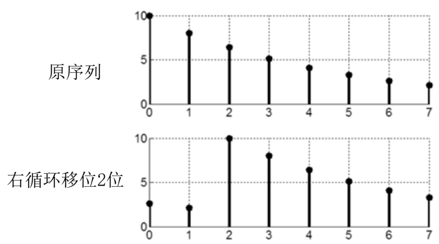
```
function y = cirshift(x,m,N)
%功能将序列x右循环移位m位，存入新序列y中
%输入要求x为行向量，长度小于等于N

  x = [x  zeros(1,N-length(x))];     %x序列补零使长度为N
  n = [0:1:N-1];
  n = mod(n-m,N);                     %余数运算
  y = x(n+1);

end
```

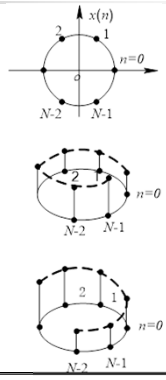

离散傅里叶变换(DFT)的性质

- 循环移位运算
  - 已知一个8点序列$x(n)=10*(0.8)^n$, $0\le n\le 7$,求出并画出 $x((n-2))_N$：y=cirshift(x,2,8)



离散傅里叶变换(DFT)的性质

- 循环移位可看作圆周移位
  - 可以想象$x(n)$是排列在一个N等分的圆周上，序列$x(n)$的循环移位，就相当于$x(n)$在此圆周上旋转
  - 若将$x(n)$向左循环移位时，此圆是顺时针旋转；将$x(n)$向右循环移位时，此圆是逆时针旋转
  - 左图表示序列向左循环移位2



离散傅里叶变换(DFT)的性质

- DFT的性质—时域循环移位定理

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

$$y(n) = x((n+m))_N R_N(n) \longrightarrow Y(k) = DFT\left[y(n)\right] = W_N^{-km} X(k)$$

  - 证明

$$Y(k) = \sum_{n=0}^{N-1} x((n+m))_N W_N^{kn} = \sum_{r=m}^{N-1+m} x((r))_N W_N^{k(r-m)} = W_N^{-km} \sum_{r=m}^{N-1+m} x((r))_N W_N^{kr}$$

$$= W_N^{-km}\left[\sum_{r=m}^{N-1} x((r))_N W_N^{kr} + \sum_{r=N}^{N-1+m} x((r))_N W_N^{kr}\right]$$

$$= W_N^{-km}\left[\sum_{r=m}^{N-1} x((r))_N W_N^{kr} + \sum_{r=0}^{m-1} x((r))_N W_N^{kr}\right]$$

$$= W_N^{-km} X(k)$$

离散傅里叶变换(DFT)的性质

- DFT的性质—时域循环移位定理

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

  - 有限长序列的时域循环移位相当于在离散频域中引入一个和频率成正比的线性相移$W_N^{-km} = e^{\left(j\frac{2\pi}{N}k\right)m}$，而对频谱的幅度没有影响

- DFT的性质—频域循环移位定理

$$X(k) = \mathrm{DFT}[x(n)] \longrightarrow \mathrm{IDFT}[X((k+l))_N R_N(k)] = W_N^{nl} x(n)$$

  - 频域的循环移位等效于时域序列的调制

离散傅里叶变换(DFT)的性质

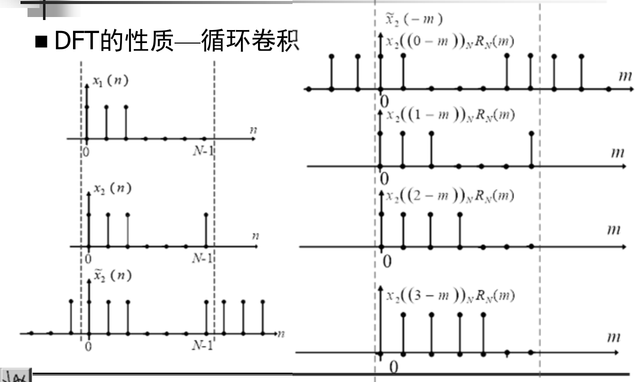
- DFT的性质—循环卷积
  - 设$x_1(n)$和$x_2(n)$都是点数为N的有限长序列，其循环卷积定义为

$$y(n) = x_1(n) \otimes x_2(n) = \sum_{m=0}^{N-1} x_1(m) x_2((n-m))_N R_N(n)$$

  - 步骤：
    - 将$x_2(m)$周期化，形成$x_2((m))_N$，再翻转形成$x_2((-m))_N$，取主值序列得到$x_2((-m))_N R_N(m)$
    - 循环右移n，形成$x_2((n-m))_N R_N(m)$
    - 当n=0,1,2,…,N-1时，分别将$x_1(m)$与$x_2((n-m))_N R_N(m)$相乘，并在m=0 ~N-1 区间内求和，得到循环卷积$y(n)$

离散傅里叶变换(DFT)的性质

- DFT的性质—循环卷积



离散傅里叶变换(DFT)的性质

- DFT的性质—时域循环卷积
  - 设$x_1(n)$和$x_2(n)$都是点数为N的有限长序列($0 \le n \le N$-1)，且有$\mathrm{DFT}[x_1(n)] = X_1(k)$, $\mathrm{DFT}[x_2(n)] = X_2(k)$, 若 $y(n) = x_1(n) \otimes x_2(n)$

    则：$Y(k) = \mathrm{DFT}[y(n)] = X_1(k) X_2(k)$

证明：

$$Y(k) = \sum_{n=0}^{N-1}\left[\sum_{m=0}^{N-1} x_1(m) x_2((n-m))_N\right] W_N^{kn}$$

$$= \sum_{m=0}^{N-1} x_1(m) \sum_{n=0}^{N-1} x_2((n-m))_N W_N^{kn}$$

$$= \sum_{m=0}^{N-1} x_1(m) \sum_{r=-m}^{N-1-m} x_2((r))_N W_N^{k(r+m)}$$

$$\sum_{r=-m}^{N-1-m} [\,] = \sum_{r=N-m}^{N-1} [\,] + \sum_{r=0}^{N-1-m} [\,]$$

$$= \sum_{m=0}^{N-1} x_1(m) W_N^{km} \sum_{r=-m}^{N-1-m} x_2((r))_N W_N^{kr} = X_1(k) X_2(k)$$

离散傅里叶变换(DFT)的性质

- DFT的性质—频域循环卷积
  - 设$x_1(n)$和$x_2(n)$都是点数为N的有限长序列($0\le n\le N$-1)，

若 $\quad y(n)=x_1(n)x_2(n)$

则 $\quad Y(k)=\mathrm{DFT}[y(n)]$

$$=\frac{1}{N}\sum_{l=0}^{N-1}X_1(l)X_2((k-l))_N R_N(k)$$

$$=\frac{1}{N}\sum_{l=0}^{N-1}X_2(l)X_1((k-l))_N R_N(k)$$

$$=\frac{1}{N}X_1(k)\otimes X_2(k)$$

  - 时域序列相乘，乘积的DFT等于各个DFT的循环卷积再乘以$1/N$

离散傅里叶变换(DFT)的性质

- DFT的性质—共轭对称性
  - 设 $x^*(n)$为 $x(n)$的共轭复数序列，则DFT[$x^*(n)$]=$X^*(N-k)$
- DFT的性质—Parseval定律
  - 可证明DFT形式下的Parseval定律：

$$\sum_{n=0}^{N-1}x(n)y^*(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)Y^*(k)$$

  - 当$y(n)=x(n)$时，即为有限长序列的能量：

$$\sum_{n=0}^{N-1}|x(n)|^2=\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2$$

快速傅立叶变换

- 离散傅里叶变换(DFT)的导出
- 离散傅里叶变换(DFT)的性质
- 快速傅立叶变换
- 利用DFT计算线性卷积
- 利用DFT进行频谱分析

快速傅立叶变换

- 引言
  - 1965年库利(J. W. Cooley)和图基(J. W. Tukey)在《计算数学》上发表了著名的“机器计算傅里叶级数的一种算法”的文章、提出了DFT的一种快速算法
  - 自此，人们开始认识到DFT运算的一些内在规律，并对算法进行了改进，从而很快地发展和完善了一套高速有效的运算方法，这就是快速傅里叶变换(FFT)算法。FFT使DFT的计算大大简化，运算时间一般可缩短一、二个数量级，从而使DFT在实际中真正得到了广泛的应用
  - 快速傅里叶变换(FFT)并不是一种新的变换，而是离散傅里叶变换(DFT)的一种快速算法

快速傅立叶变换

- DFT的运算量分析
  - 一次复数乘法需用四次实数乘法和二次实数加法；一次复数加法则需两次实数加法

$$(a+jb)(c+jd)=(ac-bd)+j(ad+bc)$$

$$(a+jb)+(c+jd)=(a+c)+j(b+d)$$

  - N点DFT的运算量分析

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

	复数乘法	复数加法	实数乘法	实数加法
一个$X(k)$	N	$N-1$	$4N$	$2(2N-1)$
N个$X(k)$	N^2	$N(N-1)$	$4N^2$	$2N(2N-1)$

快速傅立叶变换

- 减少DFT的运算量的途径
  - 利用$W_N^{nk}=e^{-j\frac{2\pi}{N}nk}$的特性

(1)对称性 $\quad \left(W_N^{nk}\right)^*=W_N^{-nk}=W_N^{(N-n)k} \quad W_N^{(k+N/2)}=-W_N^k$

(2)周期性 $\quad W_N^{nk}=W_N^{(N+n)k}=W_N^{n(N+k)}$

(3) 可约性 $\quad W_N^{nk}=W_{mN}^{mnk} \quad W_N^{nk}=W_{N/m}^{nk/m}$

(4) 特殊点 $\quad W_N^0=1 \quad W_N^{N/2}=-1$

  - 把N 点DFT化为几组点数较少的DFT运算

快速傅立叶变换

- FFT的两个重要算法
  - 基-2按时间抽取的FFT算法 [Decimation in Time (DIT)]
    - 设输入序列长度为$N=2^M$(M为正整数)，将该序列按时间顺序的奇偶分解为越来越短的子序列，称为基2按时间抽取的FFT算法，也称为Coolkey-Tukey算法
    - 若不满足$N=2^M$这个条件，可以人为地加上若干零值（加零补长）使其达到$N=2^M$
  - 基-2按频率抽取的FFT算法 [Decimation in Frequency (DIF)]
    - 设输入序列长度为$N=2^M$(M为正整数)，将该序列的频域的输出序列$X(k)$(也是N点序列)，按其频域顺序的奇偶分解为越来越短的子序列，称为基2按频率抽取的FFT算法。也称为Sander-Tukey算法

快速傅立叶变换

- 基-2按时间抽取FFT算法的推导
  - 设$N=2^M$，将$x(n)$按 n 的奇偶分为两组：

$$X(k)=[\sum_{n=0}^{N-1}x(n)W_N^{nk}]R_N(k)=\sum_{n为偶}^{N-2}x(n)W_N^{nk}+\sum_{n为奇}^{N-1}x(n)W_N^{nk}$$

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

$$=\sum_{r=0}^{\frac{N}{2}-1}x_1(r)W_{\frac{N}{2}}^{rk}+W_N^k\sum_{r=0}^{\frac{N}{2}-1}x_2(r)W_{\frac{N}{2}}^{rk}$$

$$=X_1(k)+W_N^kX_2(k)$$

$X_1(k)$和$X_2(k)$均为$N/2$点DFT

时间抽取**FFT**算法

- 基-2按时间抽取FFT算法的推导
  - $k\geq N/2$的情形　　可利用 W_N^{rk} 的周期性

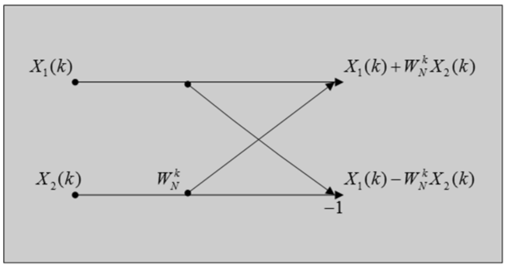
$$\because\ W_{\frac{N}{2}}^{rk}=W_{\frac{N}{2}}^{r(k+\frac{N}{2})}\quad\therefore\ X_1(k+\frac{N}{2})=X_1(k)\quad X_2(k+\frac{N}{2})=X_2(k)$$

这样，可以写出N点 $X(k)$ 与 $\frac{N}{2}$点$X_1(k)$、$X_2(k)$ 的关系式：

$$\left.\begin{aligned}X(k)&=X_1(k)+W_N^kX_2(k)\\X(k+\frac{N}{2})&=X_1(k+\frac{N}{2})+W_N^{(k+\frac{N}{2})}X_2(k+\frac{N}{2})\\&=X_1(k)-W_N^kX_2(k)\end{aligned}\right\}\quad k=0,1,\cdots,\frac{N}{2}-1$$

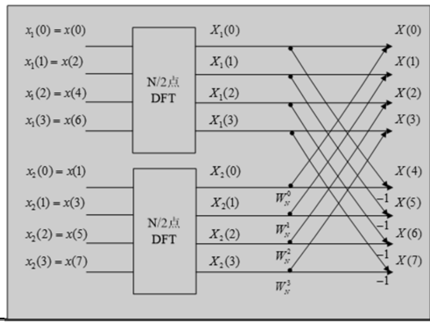
快速傅立叶变换--时间抽取FFT算法

- 蝶形信号流图表示
  - 一次蝶形运算需要一次复数乘法和二次复数加法



快速傅立叶变换--时间抽取FFT算法

- 例：两个4点DFT组成8点DFT



快速傅立叶变换--时间抽取FFT算法

- N 点DFT化为2组N/2点DFT运算的计算量分析

	复数乘法	复数加法
两个N / 2点DFT	$\frac{N^2}{2}$	$N(\frac{N}{2}-1)$
N / 2个蝶形运算	$\frac{N}{2}$	N
总计	$\frac{N^2}{2}+\frac{N}{2}$	$N(\frac{N}{2}-1)+N=\frac{N^2}{2}$
N点DFT	N^2	$N(N-1)$

快速傅立叶变换--时间抽取FFT算法

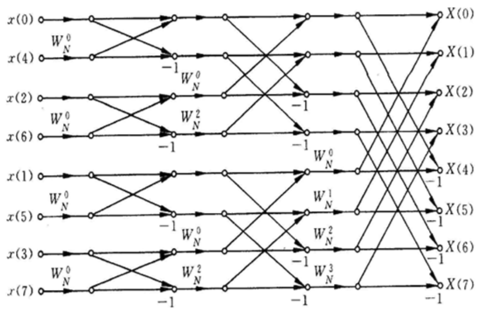
■ 进一步分解，可进一步节省计算量

- N 点DFT化为2组N/2点DFT运算的计算量可以减小一半
- 按照这个办法，继续把$N/2$用2除，由于$N=2^M$，仍然是偶数，可以被2整除，因此可以对两个N/2点的DFT再分别作进一步的分解。即对$X_1(k)$和$X_2(k)$的计算，又可以分别通过计算两个长度为$N/4=2$点的DFT，进一步节省计算量，这样，一个8点的DFT就可以分解为四个2点的DFT
- 2点DFT运算为一蝶形运算

$$X(0)=x(0)+W_2^0x(1)=x(0)+W_N^0x(1)$$
$$X(1)=x(0)+W_2^1x(1)=x(0)-W_N^0x(1)$$

快速傅立叶变换--时间抽取FFT算法

■ 按时间抽取的8点FFT



快速傅立叶变换

■ 时间抽取FFT算法与直接DFT运算量比较

- 对于$N=2^M$，总是可以通过M次分解，最后成为2点的DFT运算。这样构成从$x(n)$到$X(k)$的M级运算过程。从上面的流图可看到，每一级运算都由$N/2$个蝶形运算构成。因此每一级运算都需要$N/2$次复乘和N次复加，这样经过时间抽取后M级运算总共需要的运算
- 复乘 $\frac{N}{2}\bullet M=\frac{N}{2}\log_2 N$ 复加 $N\bullet M=N\log_2 N$
- 而直接运算时与N^2 成正比
  - 例N=2048，N^2=4194304，$(N/2)\log_2 N$=11264， $N^2/[(N/2)\log_2 N]$=392.4。FFT显然要比直接法快得多

快速傅立叶变换

■ 时间抽取FFT的运算特点--原位计算

- 当数据输入到存储器中以后，每一级运算的结果仍然储存在同一组存储器中，直到最后输出，中间无需其它存储器，这叫原位计算
- 每一级运算均可在原位进行，这种原位运算结构可节省存储单元，降低设备成本，还可节省寻址的时间

快速傅立叶变换

■ 时间抽取FFT的运算特点--码位倒置

- 对按时间抽取FFT的原位运算结构，当运算完毕时，正好顺序存放着$X(0), X(1), X(2), \ldots, X(7)$，因此可直接按顺序输出，但这种原位运算的输入$x(n)$却不能按这种自然顺序存入存储单元中，而是按$x(0), x(4), x(2), x(6), \ldots, x(7)$的顺序存入存储单元，这种顺序看起来相当杂乱，然而它也是有规律的。当用二进制表示这个顺序时，它正好是"码位倒置"的顺序
  - 例如，原来的自然顺序应是$x(1)$的地方，现在放着$x(4)$，用二进制码表示这一规律时， 则是在

    x(0 0 1)处放着 x(1 0 0)

    x(0 1 1)处放着 x(1 1 0)

快速傅立叶变换

■ 时间抽取法FFT的运算特点--码位倒置

- 在实际运算中，先按自然顺序输入存储单元，然后再通过变址运算将自然顺序的存储转换成码位倒置顺序的存储，然后进行FFT的原位计算

码位倒置顺序

自然顺序	二进码表示	码位倒置	码位倒置顺序
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	010	3
7	111	111	7

快速傅立叶变换

- 离散傅里叶反变换(IDFT)的快速计算方法
  - 前面讨论的FFT算法同样可以适用于离散傅里叶反变换(1DFT)运算，即快速傅里叶反变换(IFFT)。比较IDFT和DFT公式：

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

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

  - 只要把DFT运算中的每一个系数W_N^{nk}换成W_N^{-nk}，最后再乘以常数$1/N$，则以上所有按时间抽选或按频率抽选的FFT都可以拿来运算IDFT。

快速傅立叶变换

- 离散傅里叶反变换(IDFT)的快速计算方法
  - 还有一种完全不用改变FFT的程序就可以计算IFFT的方法。我们对IDFT公式式取共轭：

$$\left(W_N^{-nk}\right)^*=W_N^{nk}\qquad x^*(n)=\frac{1}{N}\sum_{k=0}^{N-1}X^*(k)W_N^{nk}$$

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

  - 这说明，只要先将$X(k)$取共轭，就可以直接利用FFT子程序，最后再将运算结果取一次共扼，并乘以$1/N$，即得到$x(n)$值

快速傅立叶变换

- 实序列的FFT算法
  - 实际工作中序列一般都是实序列，如果直接按照FFT的算法计算，就是把序列看成是虚部为零的复序列计算，这增加了存储量和运算时间
  - 解决方法一：
    - 一个N点的FFT计算两个N点的实序列FFT,一个序列作实部，另一个序列作虚部，计算完FFT后,有DFT的共轭对称性，由输出X(k)得到两个实序列的N点DFT
  - 解决方法二：
    - 设$x(n)$为$2N$点序列，取$x(n)$的偶数点和奇数点作为新序列的实部和虚部，有DFT的共轭对称性，达到用一个N点的FFT计算一个$2N$点的实序列的DFT的目的

快速傅立叶变换

- 快速傅立叶变换的MATLAB命令
  - $X=\mathrm{fft}(x,N)$ 对离散时间信号$x(n)$计算N点DFT
    - 当$x(n)$的长度小于N, 则在$x(n)$后面补0使$x(n)$的长度等于N;
    - 当$x(n)$的长度大于N, 则取$x(n)$前面N位
    - 缺省N值为$x(n)$的长度；
    - 采用的算法： $N=2^m$ → 基2的快速算法；
      $N\neq 2^m$ → 混合基算法，慢；
      N为质数→ 用原始的DFT算法。
  - $x=\mathrm{ifft}(X,N)$ 用来计算DFT 的反变换 IDFT
  - MATLAB数组从1开始，DFT和IDFT公式为：

$$X(k)=\sum_{n=1}^{N}x(n)W_N^{(n-1)(k-1)};\quad x(n)=\frac{1}{N}\sum_{k=1}^{N}X(k)W_N^{-(n-1)(k-1)}$$

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

$$1\le n,k\le N$$

利用DFT计算线性卷积

- 离散傅里叶变换(DFT)的导出
- 离散傅里叶变换(DFT)的性质
- 快速傅立叶变换
- 利用DFT计算线性卷积
- 利用DFT进行频谱分析

利用**DFT**计算线性卷积

- 引言
  - 大多数的实际问题是求解线性卷积，如信号$x(n)$通过线性时不变系统$h(n)$，其输出就是线性卷积$y(n)=x(n)*h(n)$
  - 循环卷积比起线性卷积，在运算速度上有很大的优越性，它可以采用快速傅里叶变换(FFT)技术，若能利用循环卷积求线性卷积，会带来很大的方便
  - 通过讨论比较$x(n)$与$h(n)$的线性卷积和循环卷积，如果$x(n)$、$h(n)$为有限长序列，分析在什么条件下线性卷积能用循环卷积代替而不产生失真

利用DFT计算线性卷积

■ 循环卷积与线性卷积的性质对比

循环卷积	线性卷积
是针对DFT引出的一种表示方法	信号通过线性系统时，信号输出等于输入与系统单位冲激响应的卷积
两序列长度必须相等，不等时按要求补足零值点	两序列长度可以不等 如$x_1(n)$为 N_1点，$x_2(n)$为 N_2点
卷积结果长度与两信号长度相等皆为N	卷积结果长度为$N=N_1+N_2-1$

利用DFT计算线性卷积

■ 循环卷积计算

$$y(n)=x_1(n)\otimes x_2(n)=\sum_{m=0}^{N-1}x_1(m)x_2((n-m))_N R_N(n)$$

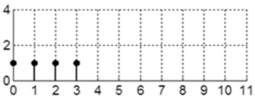
■ MATLAB代码

```
function y = circonv(x1,x2,N)
%序列x1与x2的N点循环卷积，结果存入新序列y中
%输入要求x1与x2为行向量，长度均小于等于N
x1 = [x1 zeros(1,N-length(x1))];   %序列补零使长度为N
x2 = [x2 zeros(1,N-length(x2))];
m = [0:1:N-1];
x2 = x2(mod(-m,N)+1);              % x2((-m))N
H = zeros(N,N);
for n = 1:1:N
   H(n,:)=cirshift(x2,n-1,N);      % x2((-m))N循环右移，存入H对应行中
end
y = x1*H';                         %序列x1与x2对应项相乘并求和
```

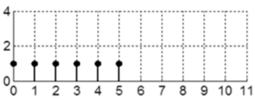

利用DFT计算线性卷积

■ 循环卷积计算

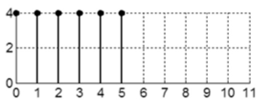
■ 设有两个序列：$x(n)=R_4(n)$，$y(n)=R_6(n)$，观察其N=6, 7, 8, 9, 10, 11点的循环卷积，并对比线性卷积



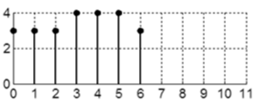
$x(n)$



$y(n)$



N=6的循环卷积结果

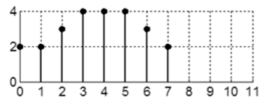


N=7的循环卷积结果

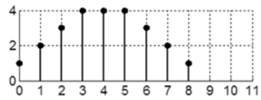
利用DFT计算线性卷积

■ 循环卷积计算(接上)

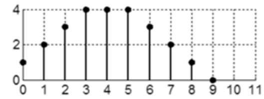
■ 设有两个序列：$x(n)=R_4(n)$，$y(n)=R_6(n)$，观察其N=6, 7, 8, 9, 10, 11点的循环卷积，并对比线性卷积



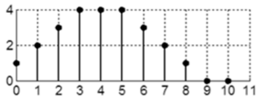
N=8的循环卷积结果



N=9的循环卷积结果；线性卷积结果



N=10的循环卷积结果



N=11的循环卷积结果

利用DFT计算线性卷积

■ 有限长序列的循环卷积与线性卷积的比较

$$y(n)=x_1(n)\otimes x_2(n)=\sum_{m=0}^{N-1}x_1(m)x_2((n-m))_N R_N(n)$$

$$y(n)=x_1(n)*x_2(n)=\sum_{m=-\infty}^{\infty}x_1(m)x_2(n-m)$$

■ 由前面的分析可知，当循环卷积的点数$N<N_1+N_2-1=9$时，循环卷积与线性卷积结果不相等，循环卷积结果是线性卷积卷积结果作N点周期延拓后混叠结果

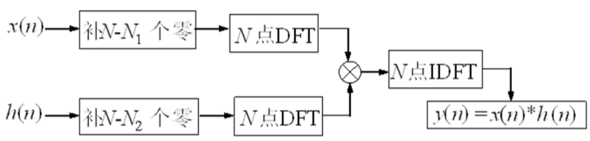
■ 使循环卷积等于线性卷积而不产生混叠的必要条件是：循环卷积的点数$N\geqslant N_1+N_2-1$，N_1和N_2分别为两个作循环卷积的有限长序列的长度

利用DFT计算线性卷积

■ 利用DFT的线性卷积算法

■ 已知线性时不变系统的输入$x(n)$和脉冲响应$h(n)$均为有限长序列，其长度分别为N_1和N_2，输入$y(n)$为$x(n)$和$h(n)$的线性卷积。利用DFT的时域循环卷积定理及循环卷积和线性卷积的等效关系，有以下计算流程图：

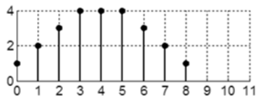
$N=N_1+N_2-1$



利用DFT计算线性卷积

- 利用DFT的线性卷积算法
  - 序列$h(n)=R_4(n)$，$x(n)=R_6(n)$，利用DFT计算线性卷积

```
h = ones(1,4); x = ones(1,6);
N=length(x)+length(h)-1;
n=0:N-1;
X=fft(x,N);
H=fft(h,N);
Y=X.*H;
y=real(ifft(Y,N));
figure;
stem(n,y);
```



N=9的循环卷积结果；

线性直接卷积结果(conv);

利用DFT的线性卷积结果

利用DFT计算线性卷积

- 利用DFT的线性卷积算法与直接线性卷积算法比较

$$x(n)*h(n)=ifft[fft(x(n))\bullet fft(h(n))]$$

$$x(n)*h(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)$$

  - 利用DFT进行N点线性卷积计算，共需两次N点FFT和一次IFFT运算，由于DFT的快速算法，当N较长时，算法所需时间会明显比直接线性卷积算法节省时间
  - 作为验证，通过编制程序比较两种方法的执行时间：现有两个长度均为N点的随机序列，当数据点数N从1点增加到1000点时，通过100次随机序列线性卷积，计算两种线性卷积方法对不同数据点数的平均计算时间

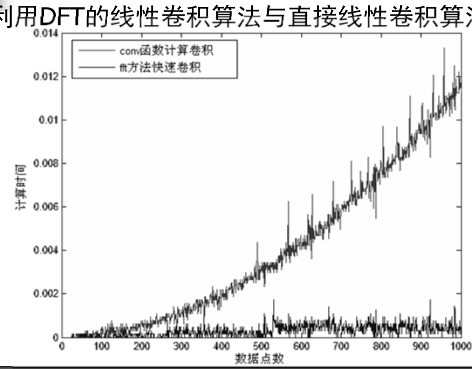
利用DFT计算线性卷积

利用**DFT**的线性卷积算法与直接线性卷积算法比较

```
conv_time = zeros(1,1000); fft_time = zeros(1,1000);
for N = 1:1000
   tc = 0; tf=0;                              %确保L为2的整数次幂，且L≥N
   L = 2*N-1; nu = round((log10(L)/log10(2))+0.45); L = 2^nu;
   for I=1:100
      h = randn(1,N);       x = rand(1,N);             %直接线性卷积
      t0 = clock; y1 = conv(h,x); t1=etime(clock,t0);
      tc = tc+t1;                                      %利用DFT的线性卷积
      t0 = clock; y2 = ifft(fft(h,L).*fft(x,L)); t2=etime(clock,t0);
      tf = tf+t2;
   end
conv_time(N)=tc/100;    fft_time(N)=tf/100;
end
plot([25:1000],conv_time(25:1000),'r',[25:1000],fft_time(25:1000),'b')
```


利用DFT计算线性卷积

- 利用DFT的线性卷积算法与直接线性卷积算法比较



利用DFT计算线性卷积

- 分块线性卷积算法
  - 前面提到的补零算法适用于$x(n)$、$h(n)$两序列长度比较接近或相等的情况；如果$x(n)$、$h(n)$长度相差较多，例如，$h(n)$为某滤波器的单位脉冲响应，长度有限，用来处理一个很长的输入信号$x(n)$，或者处理一个连续不断的信号，按前述方法，$h(n)$要补许多零再进行计算,计算量有很大的浪费
  - 长输入序列的卷积计算可以将长序列分成较小的段，每一小段长度与$h(n)$接近，通过计算每一小段的卷积然后相加来得到完整的输出

利用DFT计算线性卷积

- 分块线性卷积示例

$$x(n)=\{x(0)\quad x(1)\quad x(2)\quad x(3)\quad x(4)\quad x(5)\quad x(6)\quad x(7)\quad x(8)\quad x(9)\}$$

$$h(n)=\{h(0)\quad h(1)\quad h(2)\}$$

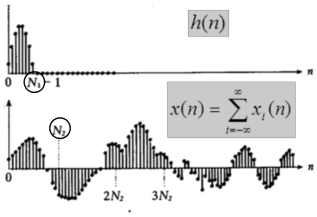
$$y(n)=x(n)*h(n)$$

$$x_1(n)=\{x(0)\quad x(1)\quad x(2)\quad x(3)\quad x(4)\}\qquad y_1(n)=x_1(n)*h(n)$$

$$x_2(n)=\{x(5)\quad x(6)\quad x(7)\quad x(8)\quad x(9)\}\qquad y_2(n)=x_2(n)*h(n)$$

利用DFT计算线性卷积

■ 分块线性卷积算法—重叠相加(overlap-add)方法



$$x_i(n)=\begin{cases}x(n) & iN_2\le n\le (i+1)N_2-1\\ 0\end{cases}$$

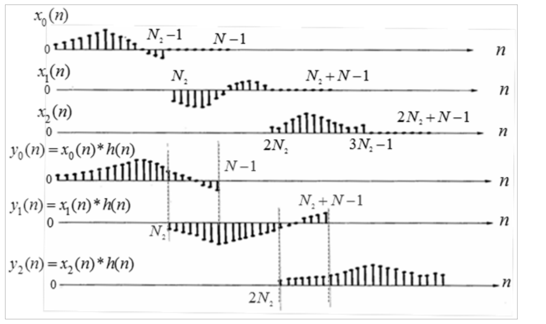
- 1)先对$h(n)$及$x_i(n)$补零，补到具有N点长度，$N=N_1+N_2-1$。一般选$N=2^M$
- 2) 用DFT快速算法计算 $y_i(n)=x_i(n)*h(n)$
- 3)由于$y_i(n)$的长度为N，而$x_i(n)$的长度为N_2，因此相邻两段$y_i(n)$序列必然有$N-N_2=N_1-1$点发生重叠。
- 4)重叠部分相加构成最后的输出序列。

利用DFT进行频谱分析

- 离散傅里叶变换(DFT)的导出
- 离散傅里叶变换(DFT)的性质
- 快速傅立叶变换
- 利用DFT计算线性卷积
- 利用DFT进行频谱分析

利用DFT计算线性卷积

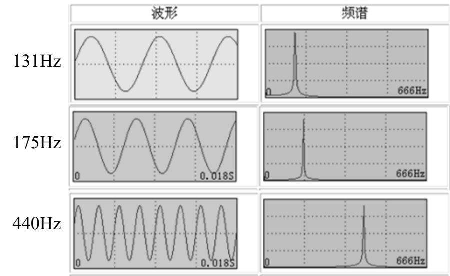
■ 分块线性卷积算法—重叠相加(overlap-add)方法



利用DFT进行频谱分析

■ 频谱分析的物理意义

- 时域和频域的对应关系



频域参数对应于设备转速、固有频率等参数，物理意义更明确。

利用DFT计算线性卷积

■ 分块线性卷积算法—重叠相加(overlap-add)方法

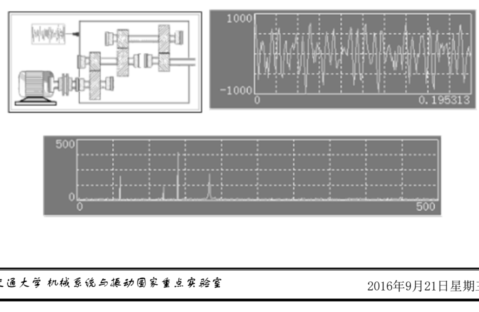
- Matlab函数实现：y=fftfilt(h,x,N)
  - h为长度有限的滤波器单位脉冲响应
  - X为长序列输入信号；N为x的分段长度，缺省时程序自动选取
- 示例：$x(n)=(n+1), 0\le n\le 9$, $h(n)=\{1,0,-1\}$，计算线性卷积

```
h = [1,0,-1]; x = [1:1:10];           % 卷积输入序列
L = length(h) + length(x) - 1;
x = [x zeros(1,L-length(x))];         % 注意长序列x要补零
N = 4;                                % 不同N选择，其结果不变
y1 = fftfilt(h,x,N);                  % 使用分块线性卷积快速算法
y2 = real(ifft(fft(h,L).*fft(x,L)));  % 使用不分块的卷积快速算法
```


利用DFT进行频谱分析

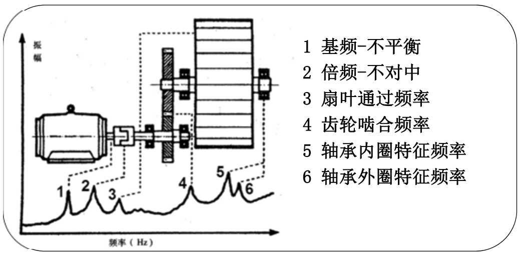
■ 频谱分析的物理意义

- 齿轮箱信号的频谱分析



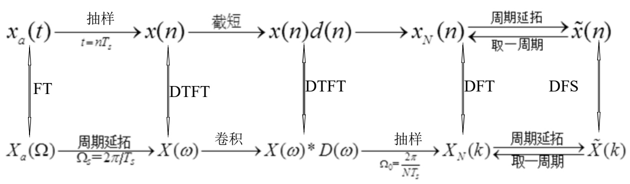
利用DFT进行频谱分析

- 频谱分析的物理意义
  - 鼓风机组的频率与机组故障的对应关系



利用DFT进行频谱分析

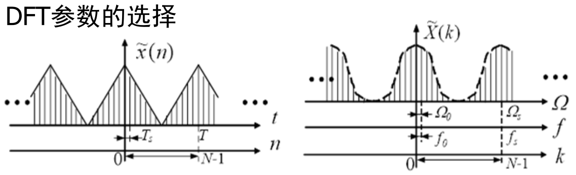
- 用DFT对一连续时间信号作频谱分析的过程



  - DFT参数选择：消除混叠现象以及提高频率分辨率
  - 频谱泄漏：信号截短所带来的问题，如何降低其影响
  - 栅栏效应：频谱离散所带来的问题

利用DFT进行频谱分析

- DFT参数的选择



  - T_s 信号时域采样间隔；f_s信号采样频率；N采样点数；T 信号记录长度；f_0频率分辨率
  - 参数间的关系

$$T = NT_s \qquad f_s = \frac{1}{T_s} \qquad f_0 = \frac{1}{T} \qquad f_s = Nf_0$$

利用DFT进行频谱分析

- DFT参数的选择(例1)
  - 对如下信号，取采样间隔T_s=1ms，观察时间记录长度T=NT_s 取不同值时的频谱计算结果

$$x(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t)$$
$$f_1 = 100Hz, f_2 = 200Hz, f_3 = 205Hz,$$

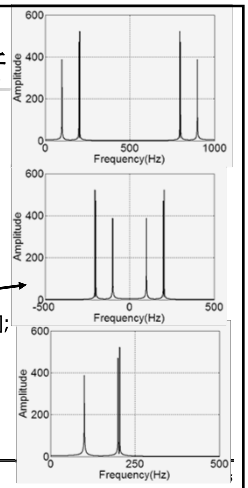
  - MATLAB代码

```
Ts = 0.001;                    %采样间隔
fs = 1/Ts;                     %采样频率
N = 1024;                      %采样点数
t = [0:Ts:(N-1)*Ts];           %时间序列建立(N点)
DataVec = sin(2*pi*100*t)+sin(2*pi*200*t)+sin(2*pi*205*t);
FreqVec = fft(DataVec);        %计算得到的频率数据(N点)
```


利用DFT进行频谱分析

- DFT参数的选择(例1)
  - 频域数据的表示方法

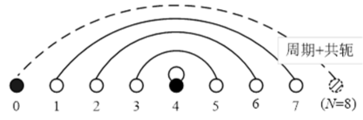
```
FreqVec = fft(DataVec);
%全部N点表示（0：N-1）
f = [0:fs/N:(N-1)*fs/N];
figure; plot(f,abs(FreqVec))
%全部N点对称表示（-N/2：N/2-1）
f_sym = [-N*fs/(2*N):fs/N:(N-1)*fs/(2*N)];
figure; plot(f_sym,abs(fftshift(FreqVec)))
%显示N/2点正频率（0：N/2-1）
f_half = [0:fs/N:(N-1)*fs/(2*N)];
figure; plot(f_half,abs(FreqVec(1:N/2)))
```



DFT离散谱的性质

- 偶数点实序列离散谱的共轭对称性
  - 如果$x(n)$ ($0 \le n < N$)为实序列，则其N点的DFT $X(k)$关于原点和$N/2$都具有共轭对称性

$$X(-k) = X^*(k) \qquad X(N-k) = X^*(k) \qquad X\left(\frac{N}{2} \pm k\right) = X^*\left(\frac{N}{2} \mp k\right)$$



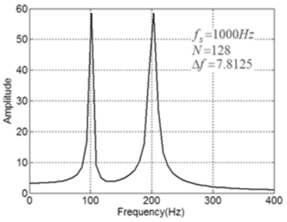
$$X(0) = \sum_{n=0}^{N-1} x(n)$$
$$X\left(\frac{N}{2}\right) = \sum_{n=0}^{N-1} (-1)^n x(n)$$

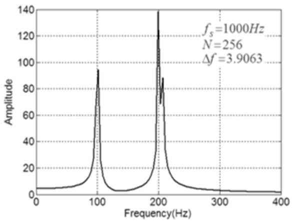
$$\|X(-k)\| = \|X(k)\| \qquad \|X(N-k)\| = \|X(k)\| \qquad \left\|X\left(\frac{N}{2} \pm k\right)\right\| = \left\|X\left(\frac{N}{2} \mp k\right)\right\|$$

利用DFT进行频谱分析

■ DFT参数的选择(例1) $x(t)=\sin(2\pi f_1 t)+\sin(2\pi f_2 t)+\sin(2\pi f_3 t)$
$f_1=100Hz, f_2=200Hz, f_3=205Hz,$

■ 采样间隔T_s=1ms一定，不同采样点数时的讨论：采样点数较少时，频率分辨率不够；注意频率幅值

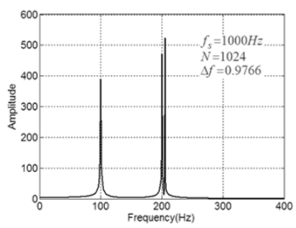


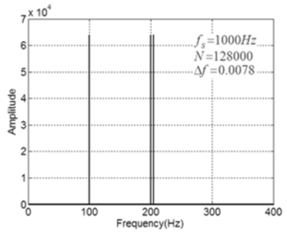


利用DFT进行频谱分析

■ DFT参数的选择(例1) $x(t)=\sin(2\pi f_1 t)+\sin(2\pi f_2 t)+\sin(2\pi f_3 t)$
$f_1=100Hz, f_2=200Hz, f_3=205Hz,$

■ 采样间隔T_s=1ms一定，不同采样点数时的讨论：采样点数增大时，3个峰值可以分出；注意频率幅值





利用DFT进行频谱分析

■ DFT参数选择的一般原则

- ■ ① 若已知信号的最高频率f_c，为防止混叠，则选定抽样频率$f_s \geq 2f_c$
- ■ ② 根据实际需要选定频率分辨率f_0(或Δf)，即可确定做DFT所需要的点数$N = f_s / f_0$，N一般取为2的整数次幂
- ■ ③ f_s和N确定以后，即可确定相应信号的所需记录长度

$$T = N / f_s = NT_s$$

利用DFT进行频谱分析

■ DFT参数的选择(例)

有一频谱分析仪用的FFT处理器，其抽样点数必须是2的整数幂。假定没有采用任何特殊的数据处理措施，已给条件为：

（1）频率分辨率≤10Hz

（2）信号的最高频率≤4kHz

试确定以下参量：

（1）最小记录长度T；

（2）抽样点的最大时间间隔T_s；

（3）在一个记录中的最少点数N

利用DFT进行频谱分析

■ DFT参数的选择(例2)

（1）由分辨力的要求确定最小记录长度T

$$\Delta f = \frac{1}{T} \Rightarrow T = \frac{1}{\Delta f} = 0.1s$$

（2）从信号的最高频率确定最大的抽样时间间隔T_s

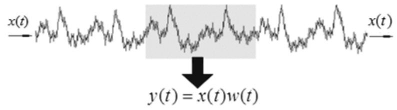
$$T_s = \frac{1}{f_s} = \frac{1}{2f_c} = \frac{1}{8000} = 0.125ms$$

（3）最小记录点数N，它应满足 $N \geq \frac{f_s}{\Delta f} = \frac{2f_c}{\Delta f} = \frac{8000}{10} = 800$

该处理器所需最少采样点数为$N=2^{10}$=1024点

利用DFT进行频谱分析

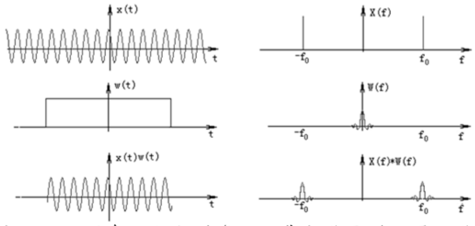
■ 信号的截断、能量泄漏



- ■ 用计算机进行数字信号处理时，不可能对无限长的信号进行测量和运算，而是取其有限的时间片段进行分析，这个过程称信号截断；时域的截断在数学上的意义为原无限长时间信号乘上一个窗函数
- ■ 时域两函数相乘，在频域是其频谱的卷积。由于窗函数不可能取无限宽,即其频谱不可能为一冲激函数,信号的频谱与窗函数的卷积必然产生拖尾现象.造成 频谱泄漏

利用DFT进行频谱分析

- 信号的截断、能量泄漏(例)



- 由于与矩形窗函数频谱卷积，截断信号谱与原始信号谱相比，它已不是原来的两条谱线，而是两段振荡的连续谱。原来集中在f_0处的能量被分散到两个较宽的频带中去了，这种现象称之为频谱能量泄漏

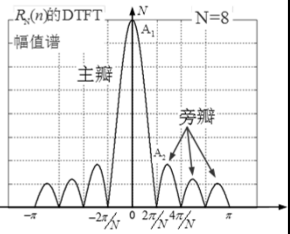
利用DFT进行频谱分析

- 矩形窗的DTFT

$$\mathrm{DTFT}(R_N(n)) = \sum_{n=0}^{N-1} e^{j\omega n}$$

  - MATLAB近似实现

```
N = 8; xn = ones(1,N);
Nfft = 8192  %较密的谱线近似连续谱
w = [-pi:2*pi/Nfft:(pi-2*pi/Nfft)];
Xk = fft(xn,Nfft);
figure
plot(w,abs(fftshift(Xk)))
```

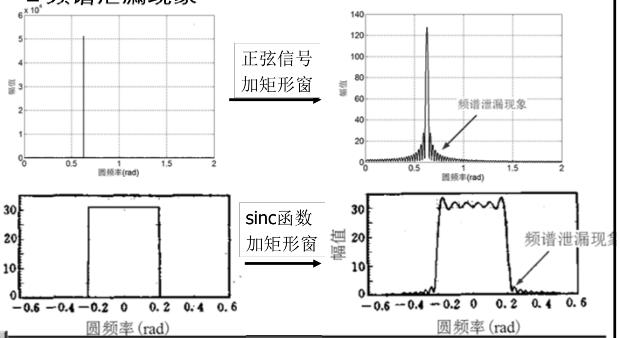


主瓣宽度为 $\frac{4\pi}{N}$　$N\uparrow \Rightarrow$ 宽度减小

旁瓣峰值衰耗 $20\log_{10}(\frac{A_2}{A_1}) \approx 13dB$

利用DFT进行频谱分析

- 频谱泄漏现象

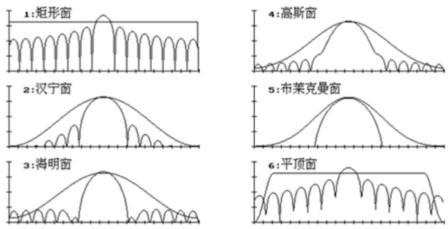


利用DFT进行频谱分析

- 减小频谱泄漏的措施
  - 在进行离散傅立叶变换时，时域的截断是必然的，因而频谱泄漏也是必然存在的。泄漏将导致频谱的扩展，从而使最高频率有可能超过折叠频率而造成混叠失真，为了减少频率泄漏可采用：
  - (1) 加大窗口宽度，即增加信号采样点数N
  - (2) 采用适当形状的窗函数截断
    - 窗函数的频谱主瓣宽度应尽可能小，以提高频率分辨率
    - 主瓣与第一旁瓣的幅度差值应尽可能大，并且旁瓣衰减越快越好，从而减小频谱泄漏提高频谱幅值精度

利用DFT进行频谱分析

- 常用窗函数的主要特性
  - 主瓣宽度最窄的是矩形窗，但其旁瓣衰减慢
  - 主瓣宽度大的窗，旁瓣衰减快



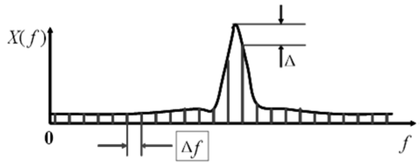
几种常用的窗函数的时域和频域波形

利用DFT进行频谱分析

- 窗函数的选择
  - 主瓣宽度与旁瓣幅度衰减相互制约
    - 主瓣宽度越小，频率分辨率就越高，但旁瓣幅值较大，频谱泄漏增大导致频谱幅值精度降低
    - 旁瓣幅度衰减越大，频谱泄漏越小，提高了频谱幅值精度，但主瓣的宽度也会随之变大，降低了频率分辨率
  - 选择窗函数，应考虑被分析信号的性质与处理要求
    - 如果仅要求精确获得信号频率，而不考虑幅值精度，例如在进行固有频率测量时，则可选用主瓣宽度比较窄而便于分辨的矩形窗
    - 如果分析窄带信号，且有较强的干扰噪声，则应选用旁瓣幅度较小的窗函数，如汉宁窗、三角窗等
    - 实际选取窗函数时，只能以折衷的方式对幅值和频率分辨率进行适当兼顾，权衡处理。一般来说，是以牺牲频率分辨率来换取能量泄漏的减少

利用DFT进行频谱分析

- 栅栏效应
  - N点DFT是在频率区间 $[0, 2\pi]$ 上对信号频谱进行N点等间隔采样，得到的是若干个离散的频谱点 $X(k)$，且它们限制在基频的整数倍上，这就好像在栅栏的一边通过缝隙看另一边的景象一样，只能在离散点处看到真实的景象，其余部分频谱成分被遮挡，所以称之为栅栏效应



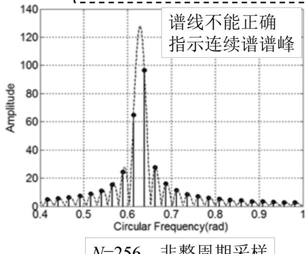
利用DFT进行频谱分析

- 栅栏效应(例)
  - 对一正弦信号 $x(t)=\sin(0.2\pi t)$，以T_s=1s采样，取不同的记录点数N，观察谱线变化情况
  - MATLAB代码

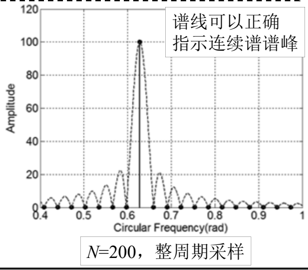
```
N = 256; xn = sin(0.2*pi*[1:N1]');      %N 采样点数
Nfft1 = N;  Xk1 = fft(xn,Nfft1);         %N点fft，得到N条谱线
Nfft2 = 8192;  Xk2 = fft(xn,Nfft2);      %较密的谱线近似连续谱
w1 = [0:2*pi/Nfft1:(2*pi/Nfft1)*(Nfft1/2-1)];
figure；stem(w1,abs(Xk1(1:Nfft1/2)))；           %谱线
w2 = [0:2*pi/Nfft2:(2*pi/Nfft2)*(Nfft2/2-1)];
hold on ; plot(w2,abs(Xk2(1:Nfft2/2)),'r')        %连续谱
```


利用DFT进行频谱分析

- 栅栏效应(例)
  - 对一正弦信号 $x(t)=\sin(0.2\pi t)$，以T_s=1s采样，取不同的记录点数N，观察谱线变化情况



N=256，非整周期采样

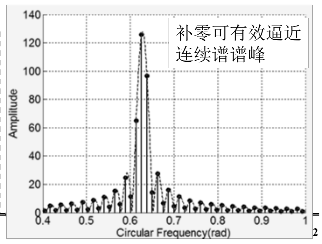


N=200，整周期采样

利用DFT进行频谱分析

- 栅栏效应讨论
  - 通常，对于旋转机械振动信号采集可以做到整周期采样，从而有效减小栅栏效应，提高谱分析精度
  - 对于一般信号，可以通过增加记录点数N来提高谱分析精度；当无法增加记录点数N时，可在采样样本后补零来减小栅栏效应

```
N = 256;
xn = sin(0.2*pi*[1:N1]');
%256采样点数
Nfft = 512;
Xk1 = fft(xn,Nfft);
%512点fft，相当于补256个零点
```

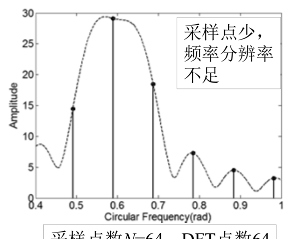


利用DFT进行频谱分析

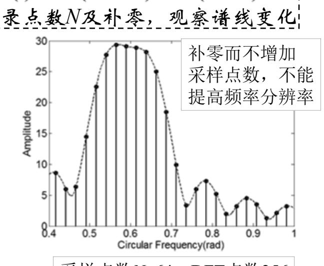
- 关于补零问题的讨论
  - 补零的好处有二：① 可使数据长度N为2的整数次幂，便于利用DFT快速算法；② 克服栅栏效应，使谱的外观得到平滑，提高谱分析的精度
  - 但是补零并不能提高频率分辨率，因为有效的数据长度并没有改变
- 算例
  - 对一双频合成正弦信号 $x(t)=\sin(0.18\pi t)+\sin(0.2\pi t)$，以$T_s$=1s采样，取不同记录点数$N$及补零，观察谱线变化

利用DFT进行频谱分析

- 算例讨论
  - 对一双频合成正弦信号 $x(t)=\sin(0.18\pi t)+\sin(0.2\pi t)$，以$T_s$=1s采样，取不同记录点数$N$及补零，观察谱线变化



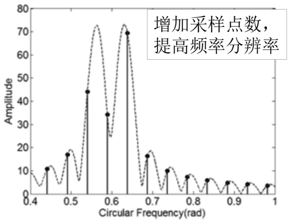
采样点数N=64，DFT点数64



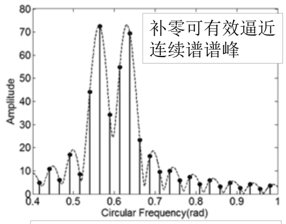
采样点数N=64，DFT点数256

利用DFT进行频谱分析

- 算例讨论
  - 对一双频合成正弦信号 $x(t)=\sin(0.18\pi t)+\sin(0.2\pi t)$，以$T_s$=1s采样，取不同记录点数$N$及补零，观察谱线变化



采样点数N=128，DFT点数128

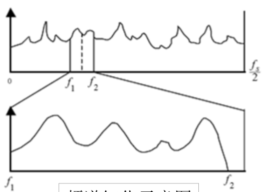


采样点数N=128，DFT点数256

利用DFT进行频谱分析

- 频谱细化(Zoom-FFT)
  - 频谱细化分析是在频谱分析中用来增加频谱中某个频率段中频率分辨率的方法
  - 设数据点数为N，采样频率为f_s,则频谱的频率分辨率为：

$$\Delta f = f_s / N$$



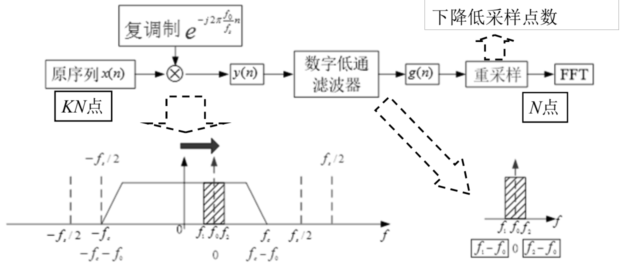
频谱细化示意图

  - 如果我们希望将频段$[f_1\sim f_2]$内频率分辨率提高K倍，直接的办法就是在采样频率不变的情况下，将数据的采样点数从N点增加到KN点，对KN点数据进行DFT运算
  - 运算的结果是频段$[0\sim f_s/2]$内频率分辨率均提高K倍，当K和N比较大时，运算量较大

利用DFT进行频谱分析

- 频谱细化(Zoom-FFT)的基本原理

在保证采样定理的条件下降低采样点数



复调制的结果为坐标原点右移f_0

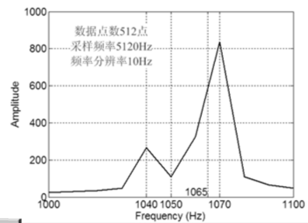
滤除所选频带以外的频率分量

利用DFT进行频谱分析

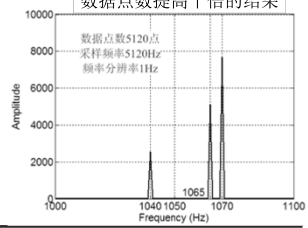
- 频谱细化(Zoom-FFT)讨论
  - 实际进行细化分析时，首先必须保证原始信号采样时有足够的长度：
    - 如果想把所选频段分辨率提高K倍，就得保证原信号的采样长度为KN，这样在每隔K点重采样时，得到新的长度为N的复序列
  - 重采样得到的复序列进行DFT运算得到的正负频率实际都有物理意义，该频率值加上原始频移就得到实际频率
  - Zoom-FFT的意义在于只计算N点DFT就可使所选频段的频率分辨率提高K倍，而不是计算KN点DFT

利用DFT进行频谱分析

- 频谱细化(Zoom-FFT)算例
  - 512点信号 $x(t)=\cos(2\pi f_1 t)+2\cos(2\pi f_2 t)+3\cos(2\pi f_3 t)$，其中$f_1$=1040$Hz$，$f_2$=1065$Hz$，$f_3$=1070$Hz$，采样频率$f_s$=5120$Hz$，频率分辨率10$Hz$



数据点数提高十倍的结果



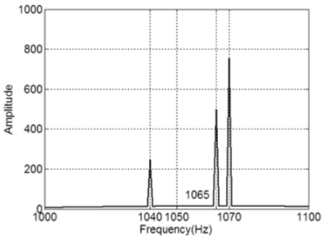
利用DFT进行频谱分析

- 频谱细化(Zoom-FFT)算例
  - MATLAB代码

```
N=5120; n=0:(N-1);                    %数据点数
fs=5120; t=n/fs;                      %采样频率
x=cos(2*pi*1040*t)+2*cos(2*pi*1065*t)+3*cos(2*pi*1070*t);
f0=1050;                              %频移1050Hz
h=exp(-j*2*pi*f0*t); y=x.*h;          %信号x复调制
[b,a] = butter(9,300/fs); g = filter(b,a,y);    %低通滤波
r=resample(g,1,10);                   %重采样，采样频率为fs/10
fs1=fs/10; N1=N/10;                   %数据点数降低10倍
Yk=fftshift(fft(r));                  %512点快速傅里叶变换
w=[f0-N1*fs1/(2*N1):fs1/N1:f0+(N1-1)*fs1/(2*N1)];  %频率对应
figure；plot(w,abs(Yk));
```

利用DFT进行频谱分析

- 频谱细化(Zoom-FFT)算例
  - 通过频谱细化，只计算512点数据就达到频率分辨率为1*Hz*的要求
  - 分析频段以1050Hz为中心，带宽512Hz



The End

谢谢！

请批评指正