

• DFT 泄漏 DFT Leakage

DFT是截断信号与信号和窗函数乘积的结果相等价

$$\text{时: } x(n) = \tilde{x}(n)R(n) \quad n=0,1,2 \dots N-1$$

都进行了截断

$$\text{频: } X(k) = \tilde{X}(k)R(k) \quad k=0,1,\dots,N-1$$

加窗会导致信号的泄漏(频域上的频谱泄漏)

由于窗函数在时域上的长度是有限的, 因而其频谱中包含了较为丰富的高频分量(如图 3.23 所示), 所以即使 $x_a(t)$ 为限带信号且取样频率满足奈奎斯特条件, 在 $X_1(e^{j\omega})$ 和 $W_M(e^{j\omega})$ 的卷积结果 $X(e^{j\omega})$ 中, 也会出现一定的高频分量; 或者说, 因频域上所需完成的卷积运算, $X(e^{j\omega})$ 不仅在取值上有别于 $X_a(j\Omega)$, 而且其频带也被延展了(如图 3.24 所示), 称此现象为频谱泄漏。总之, 频谱泄漏是由时域加窗处理所导致的一个必然结果。

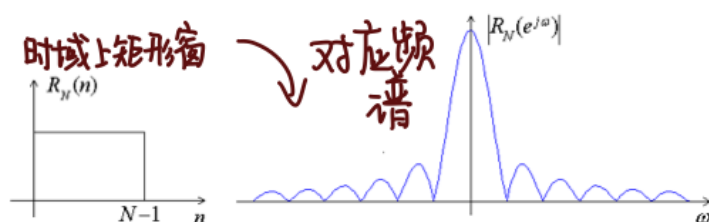


图 3.23 窗函数及其频谱

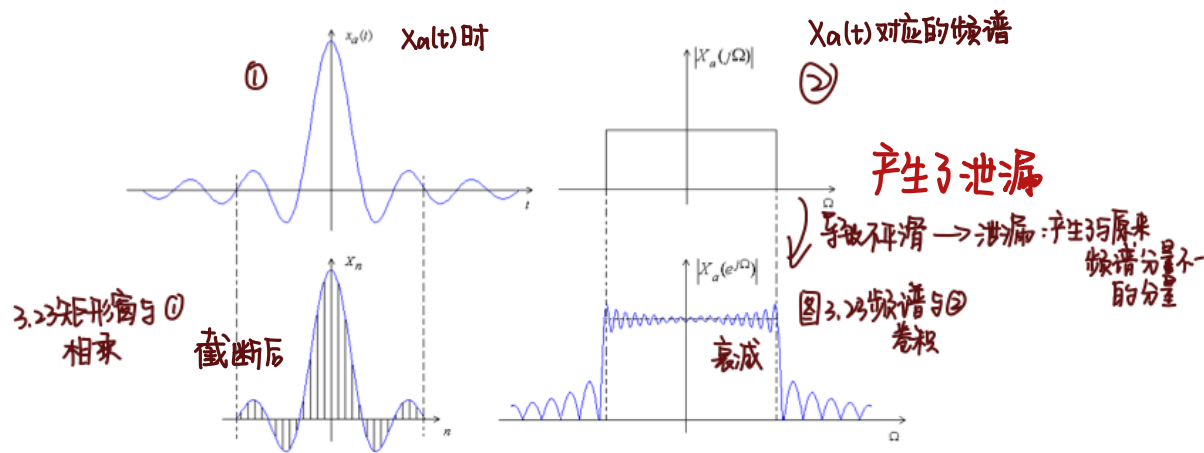


图 3.24 加窗处理与频谱泄漏

由于在利用 DFT 进行信号的频谱分析过程中, 时域加窗处理是一个不可或缺的环节, 因此在最终的分析结果中必然会包含由频谱泄漏所带来的误差, 或者说, 这是采用 DFT 作频谱分析时所遇到的一个固有问题。

由上述分析不难想到, 为了减少这一误差, 需要降低 $W_M(e^{j\omega})$ 中的高频分量。在后续有关 FIR 滤波器的讨论中将指出, 通过调整窗函数的形状可以较为有效地达到这一目的, 这里不再作进一步的分析(更多窗函数的描述, 请参见 5.3.3 节)。

最后还需指出的是，先作时域取样再作加窗处理和先作加窗处理再作时域取样所得到的有限长序列是完全相同的。既然如此，也就不难看出频谱泄漏的存在必然会导致频域混迭。因此，时域取样前的滤波以及取样频率的确定对于提高频谱分析结果的准确程度来讲，起着较为关键的作用。另外，增大窗函数的长度，会导致在所获得的有限长序列中包含更多的 $x_a(t)$ 的内容，而分析结果的准确程度显然也会因此而得到一定程度的提高。

综上所述，由于 DFT 能够与数字系统的两个基本特征相适应，其定义是严格的，物理含义是明确的，又因具有快速计算方法，因此，利用 DFT 完成信号的频谱分析是在实际应用中常见的作法。但是，由于 DFT 自身的局限性（即 DFT 是针对有限长序列所定义的一种变换），分析结果的准确程度会受到频谱泄漏、频域混迭等诸多因素的影响，因此需要采用滤波、调整取样频率、变换窗函数的形状及长度等一系列的措施来改进该方法的性能。可以说，如何更为准确和有效地完成信号的频谱分析是一个较为复杂的问题，也是数字信号处理

• 如何获得用有限长 DFT 得到 DFT $X(e^{j\omega})$ 其他取样点

Solution: 取样点 N 越多 \longrightarrow 越体现连续性

2 种基本思路:

Evidently, our sampling rate should be increased, that is increasing the length of N .

Two schemes: 2 种方法让 DFT 点数增加

- ✓ One is obtaining more samples when sampling. ^{① 取样多取点}
- ✓ The other one is appending certain length of zero, namely zero-padding operation. ^{② 补 0}
eg. 取 4 点余 4 点补 0

a) $X(n) = \{1, 1, 1, 1, 0, 0, 0, 0\}$ 在 $X(n)$ 之后补上 4 个点就可以得到 8 点序列

$$\text{8点DFT: } X_8(k) = \sum_{n=0}^7 X(n) W_8^{nk} \quad k=0, 1, 2, \dots, 7 \quad \text{其中 } W_8 = e^{-j\frac{2\pi}{8}} = e^{-j\frac{\pi}{4}}$$

• 周期卷积 (DFS)

已知 $\tilde{x}_1(n) \longleftrightarrow \tilde{x}_1(k)$ $\tilde{x}_2(n) \longleftrightarrow \tilde{x}_2(k)$

若 $\tilde{x}_3(k) = \tilde{x}_1(k) \tilde{x}_2(k)$ 频域相乘 \longleftrightarrow 时域卷积

那么 $\tilde{x}_3(n) = \sum_{m=0}^{N-1} \tilde{x}_1(m) \tilde{x}_2(n-m)$
 $= \sum_{m=0}^{N-1} \tilde{x}_2(m) \tilde{x}_1(n-m)$

若 $\tilde{x}_4(n) = \tilde{x}_1(n) \tilde{x}_2(n)$ 时域相乘 \longleftrightarrow 频域卷积 $\cdot \frac{1}{N}$

$\tilde{x}_4(k) = \frac{1}{N} \sum_{l=0}^{N-1} \tilde{x}_1(l) \tilde{x}_2(k-l)$

周期卷积 PC 与线性卷积 LC 的区别

- ① $\left\{ \begin{array}{l} \text{LC: 对序列没有要求} \\ \text{PC: 需要 2 个序列有相同周期} \end{array} \right.$
- ② $\left\{ \begin{array}{l} \text{LC: 卷积周期取决于 2 个序列的和} \\ \text{PC: 卷积周期是 1 个序列周期} \end{array} \right.$
- ③ $\left\{ \begin{array}{l} \text{LC: 结果仍取决于 2 个序列} \\ \text{PC: 结果仍是有相同周期的序列} \end{array} \right.$

• 循环卷积 Circular Convolution (DFT)

$x_1(n)$ 和 $x_2(n)$ 是 N 点序列, 定义循环卷积: $x_3(n) = x_1(n) \otimes x_2(n)$

结果产生在 $0 \leq n \leq N-1$ 范围内, 也是 N 点序列

与 N 有关, 所以也称为 N 点循环卷积

$$= \left[\sum_{m=0}^{N-1} \tilde{x}_1(m) \tilde{x}_2(n-m) \right] R_N(n)$$

\downarrow
 $x_1(m)$ 需 periodic extended

若 $x_1(n) \xrightarrow{\text{DFT}} X_1(k)$

$x_2(n) \xrightarrow{\text{DFT}} X_2(k)$

那么 $x_1(n) \otimes x_2(n) \xrightarrow{\text{DFT}} X_1(k) X_2(k)$

2 个 N 点序列的循环卷积的 DFT

结果是对应 DFT 的乘积

计算循环卷积的方法

同心圆法 Concentric Method

- $$\left\{ \begin{array}{l} x_1(n) \text{ 顺时针在内} \\ x_2(n) \text{ 逆时针在外} \end{array} \right.$$

顺时针转动

Example:

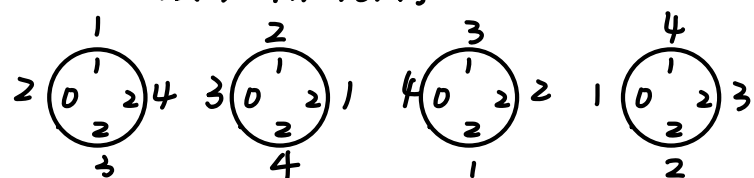
Given $x_1(n) = \{1, 2, 2\}$, $x_2(n) = \{1, 2, 3, 4\}$.

Determine 4-point circular convolution: 4 点循环卷积

$$x_1(n) \otimes x_2(n)$$

对 $x_1(n)$ 补0 $x_1(n) = \{1, 2, 2, 0\}$

$x_2(n) = \{1, 2, 3, 4\}$



$$1+6+8=15 \quad 2+8+2=12 \quad 3+2+4=9 \quad 4+4+6=14$$