第3章

第3章 离散时间信号的傅里叶变换

3.1 CTFS， CTFT

3.2 DTFT

3.3 CT信号的抽样

3.4 DTFS，DFS

为了引出DFT

3.5 DFT　重点内容

3.6 用DFT计算线性卷积

3.7 与DFT有关的几个问题

3.8 二维傅里叶变换

3.9 Hilbert 变换

傅立叶变换是信号分析与处理的基本工具

例 9．利用离散傅立叶变换可以实现有限脉冲响应滤波器（用圆周卷积计算线性卷积，也可以利用 DFT 来实现无限脉冲响应滤波器，但其实际意义不大，因为对于一个只有单极点的系统 $H(z)=1/D(z)$，在 $D(z)$ 的阶次为约 95 以上时，乘法次数才比直接解差分方程少），涉及到如下运算：用系统单位脉冲响应的离散傅立叶变换 $H(k)$ 乘以输入（或输入各段）的离散傅立叶变换 $X(k)$，得到输出的离散傅立叶变换 $Y(k)$，再计算离散傅立叶反变换。具体实现可用 FFT 算法。

滤波器的频域实现方法

如果信号被一个线性时不变系统 $h(n)$ 过虑后发生了失真，可以设计其逆系统 $h_i(n)$（也是线性时不变的）来恢复信号。要注意：$1/H(k)$ 所对应的时间序列并不是系统的逆系统的单位脉冲响应。

$h_1(n)$ 和 $h_i(n)$ 的关系

设一个线性时不变系统 $h(n)=\delta(n)-\dfrac{1}{2}\delta(n-n_0)$，计算下列各问（考虑 N 为 n_0 的整倍数的情况）：

(a).确定 $h(n)$ 的 N 点 DFT $H(k)$。

(b).若序列 $h_1(n)$ 的 N 点 DFT $H_1(k)$ 为

$$H_1(k)=\frac{1}{H(k)}, \quad k=0,1,\cdots,N-1$$

试求 $h_1(n)$。

(c).当 $N=4n_0$ 时，画出上问中 $h_1(n)$ 的略图。

(d).计算 $h(n)$ 和 $h_1(n)$ 的线性卷积，$h_1(n)$ 是 $h(n)$ 的逆系统的单位脉冲响应吗？

(e).计算 $h(n)$ 和 $h_1(n)$ 的 N 点循环卷积，并画出它的略图。

(f).确定 $h(n)$ 的逆系统的单位脉冲响应 $h_i(n)$。

(g).利用上面例 6 的结论，用该题的具体数值确定和证明$h_1(n)$和 $h_i(n)$ 的关系。

解：

(a). $$H(k)=\sum_{n=0}^{N-1}h(n)W_N^{kn}=1-\frac{1}{2}W_N^{kn_0}$$

(b). $H_1(k)=1/H(k)=1/(1-\frac{1}{2}W_N^{kn_0})$，直接计算 $IDFT$ 比较麻烦，因有 N 为 n_0 的整倍数，可利用长除法可将 $H_1(k)$ 表示为 $W_N^{kn_0}$ 的多项式，即可看出 $h_1(n)$ 就是 W_N^{kn} 的系数

长除法分析

$$1-\frac{1}{2}W_N^{kn_0} = \frac{1+\frac{1}{2}W_N^{kn_0}+\frac{1}{4}W_N^{2kn_0}+\cdots}{1}$$

$$\frac{1-\frac{1}{2}W_N^{kn_0}}{\frac{1}{2}W_N^{kn_0}}$$

$$\frac{\frac{1}{2}W_N^{kn_0}-\frac{1}{4}W_N^{2kn_0}}{\frac{1}{4}W_N^{2kn_0}}$$

能化简成有多少个非零项？

$$H_1(k) = 1+\frac{1}{2}W_N^{kn_0}+\frac{1}{4}W_N^{2kn_0}+\cdots = \sum_{l=0}^{\infty}\left(\frac{1}{2}\right)^l W_N^{lkn_0}$$

特点：无穷项，kn_0的整数倍项，N为n_0的整数倍；

对比 DFT，$H_1(k)$的特点：有限项、W_N^k的连续幂。

而 $H_1(k)=\sum_{n=0}^{N-1} h_1(n) W_N^{kn}$ ，与长除法的商比较可知， $H_1(k)$ 应为

参照$H_1(k)$的标准公式形式

怎么化简？

$$H_1(k)=\sum_{l=0}^{\infty}\left(\frac{1}{2}\right)^{l} W_N^{lkn_0}$$ （无穷项要归纳于 N 项，从结果看又可归纳于 N/n_0 项）

$$=\sum_{n=0}^{N/n_0-1}\sum_{r=0}^{\infty}\left(\frac{1}{2}\right)^{n+r\frac{N}{n_0}} W_N^{k(n+r\frac{N}{n_0})n_0}$$

令$l=n+rN/n_0$
一个变量变成两个变量，各自的作用？

$$=\sum_{n=0}^{N/n_0-1}\left(\frac{1}{2}\right)^{n} W_N^{knn_0}\sum_{r=0}^{\infty}\left(\frac{1}{2}\right)^{r\frac{N}{n_0}} W_N^{krN}, \quad W_N^{krN}=1$$

$$=\frac{1}{1-(1/2)^{N/n_0}}\sum_{n=0}^{N/n_0-1}\left(\frac{1}{2}\right)^{n} W_N^{knn_0}$$

$$=\underbrace{\frac{1}{1-2^{-N/n_0}}\left[1+\frac{1}{2}W_N^{kn_0}+\frac{1}{4}W_N^{2kn_0}+\cdots+\frac{1}{2^{N/n_0-1}}W_N^{(N/n_0-1)kn_0}\right]}$$

$H_1(k)$的指数项系数

所以 $h_1(n)$ 为

以n_0为单位而不是以1为单位的整倍数点上的值

$$h_1(n)=\frac{1}{1-2^{-N/n_0}}\sum_{k=0}^{N/n_0-1}\left(\frac{1}{2}\right)^{n/n_0}\delta(n-kn_0)$$

(c). $N=4n_0$， $h_1(n)$ 为

$$h_1(n)=\frac{1}{1-2^{-4}}\left[1+\frac{1}{2}\delta(n-n_0)+\frac{1}{4}\delta(n-2n_0)+\frac{1}{8}\delta(n-3n_0)\right]$$

$$=\frac{16}{15}+\frac{8}{15}\delta(n-n_0)+\frac{4}{15}\delta(n-2n_0)+\frac{2}{15}\delta(n-3n_0)$$

(d). $h(n)$ 和 $h_1(n)$ 的线性卷积为

计算具体的$y_1(n)$时n为常数，l为变量；
掌握与$\delta(n)$的乘积运算和卷积运算，都可以化简计算。

$$y_1(n)=h(n)*h_1(n)=\sum_{l=0}^{\infty}h(l)h_1(n-l)$$

$h(n)=\delta(n)-\delta(n-n_0)/2$

$$=\sum_{l=0}^{\infty}\left[\delta(l)-\frac{1}{2}\delta(l-n_0)\right]\left[\frac{16}{15}\sum_{k=0}^{3}\frac{1}{2^{(n-l)/n_0}}\delta(n-l-kn_0)\right]$$

$$=\frac{16}{15}\sum_{k=0}^{3}\frac{1}{2^{n/n_0}}\delta(n-kn_0)-\frac{1}{2}\frac{16}{15}\sum_{k=0}^{3}\frac{1}{2^{(n-n_0)/n_0}}\delta(n-n_0-kn_0)$$

$$y_1(0)=16/15\neq 1$$

$$y_1(in_0)=\frac{16}{15}\frac{1}{2^i}-\frac{1}{2}\frac{16}{15}\frac{1}{2^{i-1}}=0, i\neq 0$$

$$y_1(N)=-\frac{1}{2}\frac{16}{15}\frac{1}{2^{i-1}}\neq 0$$

$h(n)$：n的取值区间，$[0, n_0]$
$h_1(n)$：n的取值区间，$[0, (N/n_0-1)n_0]$
$y_1(n)$：n的取值区间，$[0, N]$
计算$y_1(0)$时，？有贡献？
计算$y_1(N)$时，？有贡献？

可见，$h(n)*h_1(n)\neq\delta(n)$，$h_1(n)$也就不是$h(n)$的逆系统的单位脉冲响应。

(e). 计算$h(n)$和$h_1(n)$的N点循环卷积为

循环卷积，非零点为：in_0，$i=0,\ldots,N/n_0-1$；
h(n)不动，按N周期化，周期内两个非零点；
h_1(n)翻转转之后按N周期化，有N/n_0个非零点。

$$y_2(n)=h(n)\otimes h_1(n)=\left[\sum_{l=0}^{N-1}h((l))_N h_1((n-l))_N\right]R_N(n)$$

翻转和
周期化

$$y_2(0)=\left[\sum_{l=0}^{N-1}h((l))_N h_1((-l))_N\right]R_N(n)=\frac{16}{15}+\frac{\overset{2}{\cancel{16}}}{15}\left(-\frac{1}{2}\right)=1$$

对应两点
相乘相加

$$y_2(in_0)=\frac{16}{15}\frac{1}{2^i}-\frac{1}{2}\frac{16}{15}\frac{1}{2^{i-1}}=0, i\neq 0$$

$y_2(n)$取N点长，所以有$y_2(n)=\delta(n)$

循环卷积不同于线性卷积，周期延拓后对应非零点上相乘相加，周期值为N。
$h(n)$：n的取值区间，$[0, N-1]$
$h_1(n)$：n的取值区间，$[0, N-1]$
$y_1(n)$：n的取值区间，$[0, N-1]$

序列$h(l)$ 和 $h_1(-l)$ 的图示

(f). 确定 $h(n)$ 逆系统的 $h_i(n)$ 有几种方法。本题因有 $H_i(z)=1/H(z)=1/(1-\frac{1}{2}z^{-n_0})$ 关系，及 N 为 n_0 的整倍数，所以可以用长除法计算 $H_i(z)$ 的反变换。另外一个简单的方法是利用 Z 变换的时域扩展性质。

利用长除法，可以得到：

$$H_i(z)=\sum_{n=0}^{\infty}\left(\frac{1}{2}\right)^n z^{-nn_0}$$，所以 $h_i(n)$ 为

$$h_i(n)=\begin{cases}\left(\frac{1}{2}\right)^{n/n_0}, & n\text{为}n_0\text{的整倍数}\\ 0, & \text{其它}\end{cases}$$

(g). 利用例 6 的结果，可以得出

$$h_1(n)=\sum_{r=-\infty}^{\infty}h_i(n+rN)=\sum_{r=0}^{\infty}h_i(n+rN)=\sum_{r=0}^{\infty}\left(\frac{1}{2}\right)^{(n+rN)/n_0}$$

$$=\sum_{r=0}^{\infty}\left(\frac{1}{2}\right)^{n/n_0}\left(\frac{1}{2}\right)^{rN/n_0}=\frac{1}{1-2^{-N/n_0}}\left(\frac{1}{2}\right)^{n/n_0}$$

$$=\frac{1}{1-2^{-N/n_0}}\sum_{k=0}^{N/n_0-1}\left(\frac{1}{2}\right)^{n/n_0}\delta(n-kn_0)$$

$h_i(n)$为无限时宽序列，其ZT为$H_i(z)$，在单位圆上对$H_i(z)$等间隔取N点值，所对应取样值序列为本题目的$H_1(k)$；

$H_1(k)$的IDFT结果，为所求的时间序列$h_1(n)$。可以利用例6的结论。

<u>总结：信号间关系的图示。</u>

另外，利用ZT时域扩展性质求解$h_i(n)$的方法

已知： $$h_0(n)=\left(\frac{1}{2}\right)^n u(n)\Leftrightarrow H_0(z)=\frac{1}{1-\frac{1}{2}z^{-1}}$$

根据ZT时域扩展性质，$h_0(n)$的连续点之间插入n_0-1个零，有：

$$h_i(n)=\begin{cases}\left(\frac{1}{2}\right)^{n/n_0} & n\text{为}n_0\text{的整倍数}\\ 0 & \text{其它}\end{cases}$$

$$H_i(z)=\frac{1}{1-\frac{1}{2}(z^{n_0})^{-1}}=\frac{1}{1-\frac{1}{2}z^{-n_0}}$$

根据本题目已知条件，即给出了$H_i(z)$的表达式，所以，所求的$h_i(n)$，就是$h_0(n)$的时域扩展——相邻点之间插入了n_0-1个零。

扩展：同态滤波及复倒谱，回波噪声滤波，参见9.7节

同态滤波及复倒谱简介

$$\text{加性噪声}\begin{cases} y(n) = \underbrace{[s(n) + u(n)]}_{x(n)} * h(n) \\ Y(e^{\mathrm{j}\omega}) = S(e^{\mathrm{j}\omega})H(e^{\mathrm{j}\omega}) + U(e^{\mathrm{j}\omega})H(e^{\mathrm{j}\omega}) \end{cases}$$

若$s(n)$和$u(n)$的频谱互不重叠，那么可以通过合理设计的滤波器（线性时不变系统），按线性滤波方式去除噪声$u(n)$。

$$\text{线性滤波}\begin{cases} y(n) = x(n) * h(n) \\ Y(e^{\mathrm{j}\omega}) = X(e^{\mathrm{j}\omega})H(e^{\mathrm{j}\omega}) \end{cases}$$

乘性噪声 $\begin{cases} x(n) = s(n)u(n) \\ X(e^{j\omega}) = S(e^{j\omega}) * U(e^{j\omega}) \end{cases}$

卷积噪声 $\begin{cases} x(n) = s(n) * u(n) \\ X(e^{j\omega}) = S(e^{j\omega})U(e^{j\omega}) \end{cases}$

调制信号就是乘法性的，即传输出去的信号是待调信号和载波信号的乘积。

卷积性信号多出现在有回波的场合，如语音、雷达、声纳及超声成像等领域。

在这两种情况下信号的频谱和噪声的频谱混叠在一起，不能简单地用线性滤波的方法去出噪声。

同态滤波可以实现上述类型的去噪问题。又称广义线性滤波。

基本思路：先想办法把信号和噪声的混合方式变换为加性方式，然后借助于线性滤波的方法来去除噪声

对乘法性噪声

$$x(n) = s(n)u(n)$$

步骤1
取对数

$$\ln x(n) = \ln s(n) + \ln u(n)$$

$$x'(n) = s'(n) + u'(n)$$

$$X'(e^{j\omega}) = S'(e^{j\omega}) + U'(e^{j\omega})$$

令$H(e^{j\omega})$在$U'(e^{j\omega})$的有效频率范围内接近为零

步骤2
线性滤波

$$y'(n) = x'(n) * h(n) \Rightarrow s'(n)$$

步骤3
取指数

$$y(n) = \exp[y'(n)] = \exp[s'(n)]$$
$$= \exp[\ln s(n)] = s(n)$$

对卷积性噪声

$$x(n) = s(n) * u(n)$$

步骤1

取Z变换

$$X(z) = S(z)U(z)$$

步骤2

取对数

$$\ln X(z) = \ln S(z) + \ln U(z)$$

$$\hat{X}(z) = \hat{S}(z) + \widehat{U}(z)$$

步骤3

取Z反变换

$$\hat{x}(n) = \hat{s}(n) + \hat{u}(n)$$

步骤4
线性滤波

$$\hat{y}(n) = \hat{x}(n) * h(n) \Rightarrow \hat{s}(n)$$

步骤5
取Z变换

$$\hat{Y}(z) = \hat{S}(z)$$

步骤6
取指数

$$Y(z) = \exp[\hat{Y}(z)] = \exp[\hat{S}(z)]$$
$$= \exp[\ln S(z)] = S(z)$$

$$y(n) = s(n)$$

在上述步骤中，有

$$\hat{X}(z) = \ln X(z)$$

$$\hat{x}(n) = Z^{-1}[\hat{X}(z)] = Z^{-1}[\ln X(z)]$$

$$\hat{x}(n) = F^{-1}[\ln X(e^{j\omega})]$$

$\hat{x}(n)$ 是 $x(n)$ 的傅里叶变换取自然对数后的傅里叶反变换，称其为倒谱，由于其一般为复数，故称之为复倒谱（Complex Cepstrum）

例：声源发出信号$s(n)$，接收器收到的信号$x(n)$是由信号$s(n)$和其不同时间延迟、幅度减小的反射信号的叠加，这些反射信号称之为$s(n)$的回波（ehco)。

$$x(n) = s(n) + \sum_{k=1}^{M} \alpha_k s(n - n_k)$$

式中 $$0 < n_1 < n_2 < \cdots < n_M, \quad |\alpha_k| < 1$$

若记 $$p(n) = \delta(n) + \sum_{k=1}^{M} \alpha_k \delta(n - n_k)$$

则 $$x(n) = s(n) * p(n)$$

对单一延迟，有 $p(n) = \delta(n) + \alpha_1 \delta(n - n_1)$

$$x(n)=s(n)+\alpha_1 s(n-n_1)$$

从$x(n)=s(n)*p(n)$中分离出$s(n)$

$$X(z)=S(z)P(z)$$

$$P(z)=1+\alpha_1 z^{-n_1}$$

属同态滤波问题

取Z变换

$$\hat{X}(z)=\ln X(z)=\ln S(z)+\ln(1+\alpha_1 z^{-n_1})$$
$$=\hat{S}(z)+\hat{P}(z)$$

取对数

$$\hat{x}(n)=\hat{s}(n)+\hat{p}(n)$$

设计一滤波器h，将 $\hat{p}(n)$ 去除

取Z反变换

线性滤波

$$\hat{p}(n)=\sum_{k=1}^{\infty}(-1)^{k+1}\frac{{\alpha_1}^k}{k}\delta(n-kn_1)$$

在滤波得到$\hat{s}(n)$之后，再做Z变换、指数运算，就得到$s(n)$

3.6 用 DFT 计算线性卷积

- 圆周卷积/循环卷积分析
  - 参与圆周卷积的两个序列是等长度的，设都是N点序列
  - 圆周卷积的结果序列也是N点序列
  - 圆周卷积是建立在DFT之上的
  - DFT隐含着周期性
    - 时域和频域对应着各自以N为周期的周期序列

$$x(n) \otimes h(n) = \mathrm{IDFT}\{X(k)H(k)\} = y_{\otimes}(n)$$

回顾：循环卷积定义

三个序列都是周期为N的周期序列

$$y(n,\ \mathrm{mod}\ \ N) = x(n) * h(n)$$
$$= \sum_{i=0}^{N-1} x(i,\ \mathrm{mod}\ \ N)h(n-i,\ \mathrm{mod}\ \ N)$$

循环卷积定理

$$y(n) = \sum_{i=0}^{N-1} x(i)h(n-i), \quad y(n): N\text{点序列}$$

$$\boxed{Y(k) = X(k)H(k)}$$

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} \underbrace{[X(k)H(k)]}_{Y(k)} W_N{}^{-nk}$$

掌握证明过程

线性卷积分析

$$x(n), \quad n = 0,1,\cdots,N-1$$

$$h(n), \quad n = 0,1,\cdots,M-1$$

都是非周期

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

$n = 0,\ldots,N+M-2$ 长度 $L = N+M-1$

为什么用DFT计算线性卷积？

DFT有快速算法FFT

如何用DFT来实现？

存在什么矛盾？

DFT实现线性卷积的步骤

$x(n)$

$n=0,1,\cdots,N-1$

补零

$x'(n)$

$n=0,1,\cdots,L-1$

DFT

$X'(k)$

$k=0,1,\cdots,L-1$

$$L=N+M-1$$

$y(n)$

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

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

IDFT

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

相乘

$h(n)$

$n=0,1,\cdots,M-1$

补零

$h'(n)$

$n=0,1,\cdots,L-1$

DFT

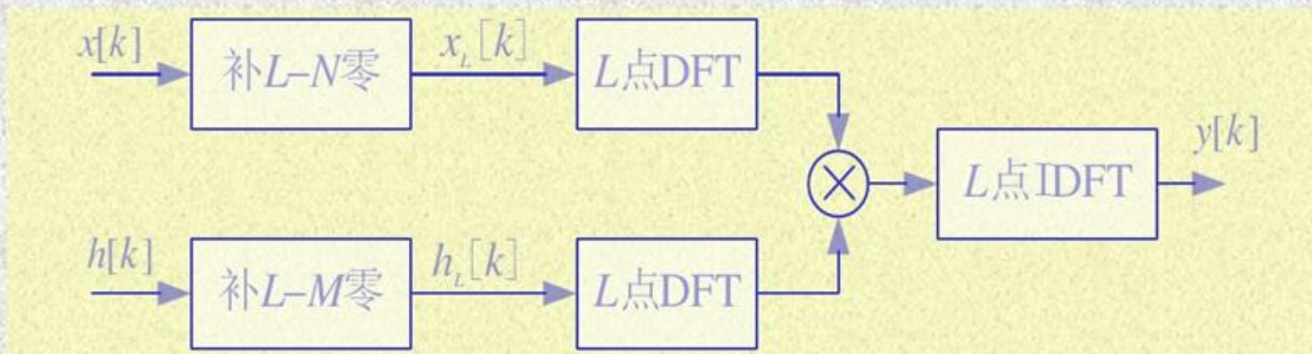
$H'(k)$

$k=0,1,\cdots,L-1$

利用DFT计算序列线性卷积的步骤

若$x[k]$的长度为N，$h[k]$的长度为M，则

$L=N+M-1$点循环卷积等于$x[k]$ 与$h[k]$的线性卷积。



$$x_L[k] \otimes h_L[k] = x[k] * h[k]$$

循环卷积，既可以在时域直接实现，也可以在频域借助于DFT计算实现。

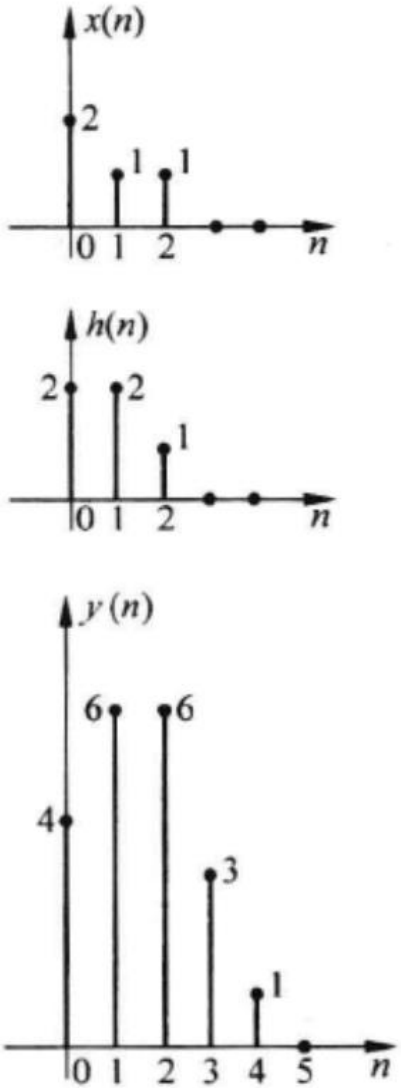
DFT有快速算法FFT，当信号长度N很大时，频域计算速度比时域计算快很多。

循环卷积可以用来实现线性卷积，再借助FFT，可以实现快速线性卷积。

- 补零并没有增加信号本质不同的信息
  - 补零序列的线性卷积与未补零序列的线性卷积的结果是一样的
  - 补零序列的圆周圆周卷积却不等于先前序列的圆周卷积
    - 序列长度发生了变化；上述做法中长度正好等于线性卷积长度
  - 线性卷积与圆周卷积
    - 线性卷积有明确的物理意义
    - 圆周卷积操作时，可对序列的长度进行改变
      - 可以通过补零的方法进行更长序列的圆周卷积
      - 只是一种计算手段（频域抽样加密就对应时域序列补零）
    - 设参与卷积的两个序列的时宽分别是$N1$、$N2$
      - 则线性卷积结果y_L的时宽为$N1+N2-1$
      - 进行L点圆周卷积，结果是y_L以L为周期的周期延拓序列的主值序列

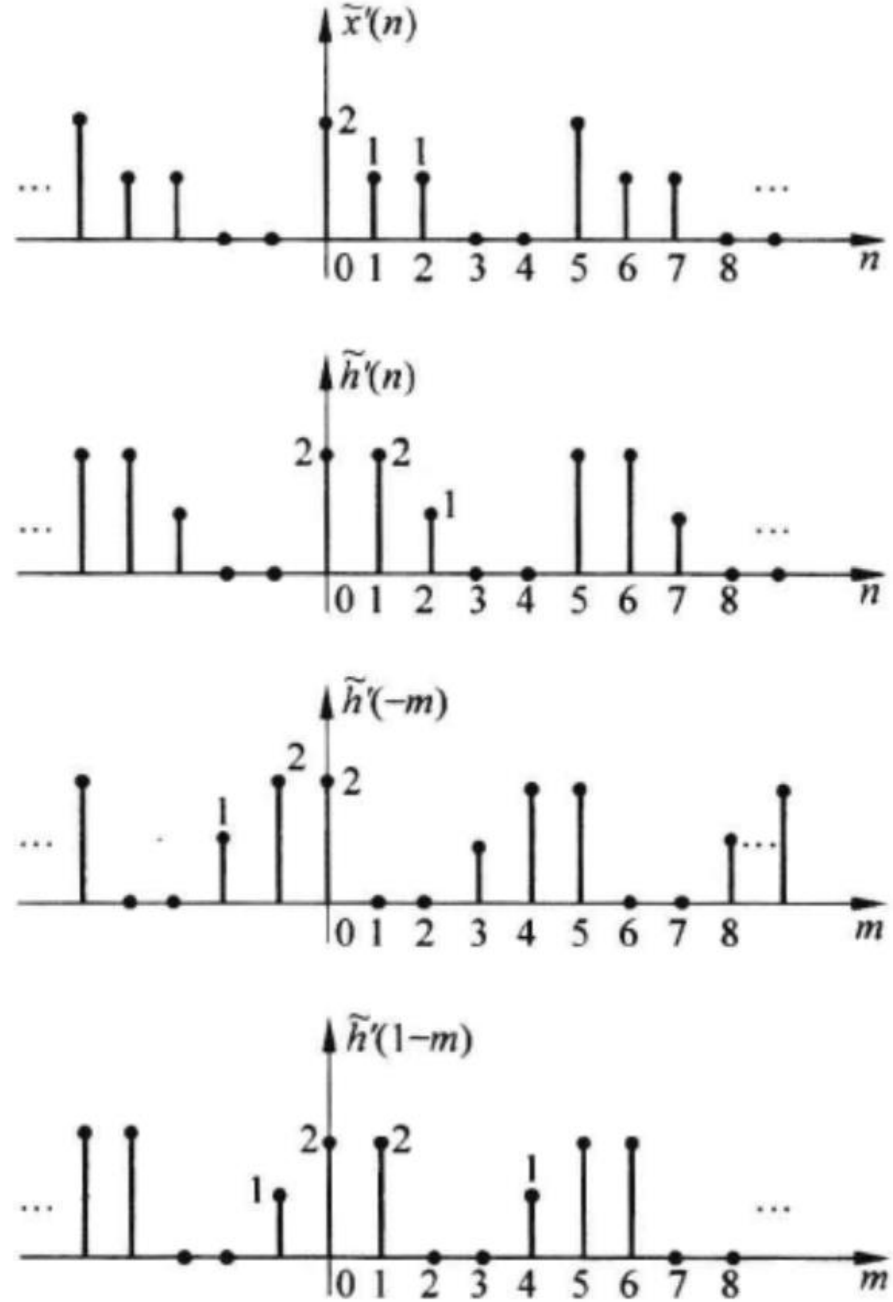
$$y_{\otimes}(n) = \left[\sum_{r=-\infty}^{\infty} y_L(n+rL)\right] \cdot R_L(n)$$

      - 当$L \geq N1+N2-1$时，$y_{\otimes}$的前（$N1+N2-1$）个点正好是$y_{\otimes}$的全部非零值，也正好是线性卷积结果y_L。



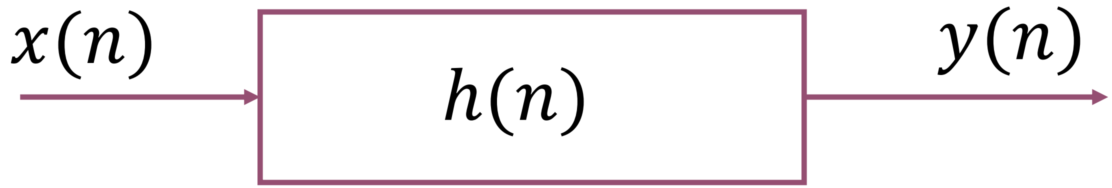
左图：
计算[2,1,1]与[2,2,1]的线性卷积

右图：
用循环卷积计算线性卷积



按点处理
按帧处理

长序列卷积的计算



数字信号处理的优势是“实时实现”，即信号进来后，经处理理后马上输出出去。然而：

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

$x(n)$ 没有全部进入，如何实现卷积？

全部进入再卷积，又如何保证实时实现？

关键是将 $x(n)$ **分段**和 $h(n)$ **卷积**

$$
\begin{array}{ll}
x(n): & N \\
h(n): & M \\
y(n): & N+M-1
\end{array}
$$

将 $x(n)$ 分成 L 段，每段长 $K=N/L$

$$
\left.
\begin{array}{l}
x_1(n), x_2(n), \cdots, x_L(n) \\
y_1(n), y_2(n), \cdots, y_L(n) \\
\quad K+M-1
\end{array}
\right\}
\qquad
\begin{array}{l}
L(K+M-1) \\
= N+LM-L \\
\neq N+M-1
\end{array}
$$

另外：

$h(n)$较短（FIR：长度在20 ~ 50之间，IIR：尽管无限长，但一般有限长度要小于50），$x(n)$可能很长，也**不适宜直接卷积**。

Overlap — add method　叠接相加法

Overlap — save method　叠接舍去法

掌握程序设计

叠接相加法分析（1）

分段计算：

$$h(n), 0 \le n \le M-1, M\text{点}$$
$$x_1(n), 0 \le n \le L-1, L\text{点}; y_1(n), 0 \le n \le L+M-2, L+M-1\text{点}$$
$$x_2(n), L \le n \le 2L-1, L\text{点}; y_2(n), L \le n \le 2L+M-2, L+M-1\text{点}$$

从 $y_1(n)$ 、$y_2(n)$ 可以看出，其自变量有重叠部分：$L \le n \le L+M-2$，长为 $M-1$ 点

分析重叠点上各段输出间的关系：

计算 $y_1(n)$ 时，假定输入为 $x_1(n)$，$0 \le n \le L-1$，其余点上输入为 0。事实上不是。并导致在 $L \le n \le L+M-2$ 点上，$y_1(n)$ 计算不完整，因为它忽略了 $n \ge L$ 后的输入。

计算 $y_2(n)$ 时，假定输入为 $x_2(n)$，$L \le n \le 2L-1$，其余点上输入为 0。事实上不是。并导致在 $L \le n \le L+M-2$ 点上，$y_2(n)$ 计算不完整，因为它忽略了 $n < L$ 上的输入。

事实上，在 $L \le n \le L+M-2$ 点上，$y(n) = y_1(n) + y_2(n)$，这就是叠接相加法。

结论：上一段的后过渡过程与本段的前过渡过程的对应点相加。$M-1$ 点对应相加。

叠接相加法分析（2）

n	$y(n)=x(n)*h(n)$	$y_1(n)=x_1(n)*h(n)$	$y_2(n)=x_2(n)*h(n)$
0	$y(0)=h_0x_0$	$y_1(0)=h_0x_0$	
1	$y(1)=h_1x_0+h_0x_1$	$y_1(1)=h_1x_0+h_0x_1$	
2	$y(2)=h_2x_0+h_1x_1+h_0x_2$	$y_1(2)=h_2x_0+h_1x_1+h_0x_2$	
3	$y(3)=h_2x_1+h_1x_2+h_0x_3$	$y_1(3)=h_2x_1+h_1x_2+h_0x_3$	
4	$y(4)=h_2x_2+h_1x_3+h_0x_4$	$y_1(4)=h_2x_2+h_1x_3+h_0x_4$	
5	$y(5)=h_2x_3+h_1x_4+h_0x_5$	$y_1(5)=h_2x_3+h_1x_4$	$y_2(5)=h_0x_5$
6	$y(6)=h_2x_4+h_1x_5+h_0x_6$	$y_1(6)=h_2x_4$	$y_2(6)=h_1x_5+h_0x_6$
7	$y(7)=h_2x_5+h_1x_6+h_0x_7$	$y_1(7)=0$	$y_2(7)=h_2x_5+h_1x_6+h_0x_7$
8	$y(8)=h_2x_6+h_1x_7+h_0x_8$		$y_2(8)=h_2x_6+h_1x_7+h_0x_8$
9	$y(9)=h_2x_7+h_1x_8+h_0x_9$		$y_2(9)=h_2x_7+h_1x_8+h_0x_9$
10	$y(10)=h_2x_8+h_1x_9+h_0x_{10}$		$y_2(10)=h_2x_8+h_1x_9$
11			$y_2(11)=h_2x_9$
12			$y_2(12)=0$

Matlab函数实现：filter、filtic
MATLAB快速计算：fftfilt
设计程序C语言程序实现：filter

据此可以分析出另一种方法

长语音实时滤波，C语言浮点子程序定点化