XI. Discrete Fourier Transform 的替代方案

◎ 11-A Why Should We Use Other Operations?

Discrete Fourier Transform (DFT):

$$X_F[m] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi mn}{N}}$$

優點： 有 fast algorithm (complexity 為 $O(N\log_2 N)$).

適合做頻譜分析和 convolution implementation

問題： (1) complex output

(2) The exponential function is irrational.

Applications of the DFT: ① Spectrum analysis ② Performing the convolution

For **spectrum analysis**, the DFT can be replaced by:

(1) DCT, even (2) DST, odd (3) DHT, real

(4) Walsh (Hadamard) transform,

(5) Haar transform,

(6) orthogonal basis expansion,

(including orthogonal polynomials and CDMA),

(7) wavelet transform,

(8) time-frequency distribution

When **performing the convolution**, the DFT can be replaced by:

(1) DCT, (2) DST, (3) DHT,

(4) Directly Computing,

(5) Sectioned DFT convolution,

IFFT(FFT(x[n])FFT(h[n]))

(6) Winograd algorithm,

(7) number theoretic transform (NTT)

(8) Z-transform based recursive method

◉ 11-B Discrete Sinusoid Transforms

DCT (discrete cosine transform) has 8 types

DST (discrete sine transform) has 8 types

DHT (discrete Hartley transform) has 4 types

共通的特性：皆為 real, 且和 DFT 密切相關

Reference

- N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan 1974.
- Z. Cvetkovic and M. V. Popovic, "New fast recursive algorithms for the computation of discrete cosine and sine transforms," *IEEE Trans. Signal Processing*, vol. 40, pp. 2083-2086, Aug. 1992.
- R. N. Bracewell, *The Hartley Transform*, New York, Oxford University Press, 1986.
- S. C. Chan and K. L. Ho, "Prime factor real-valued Fourier, cosine and Hartley transform," *Proc. Signal Processing VI*, pp. 1045-1048, 1992.

- **Case 1: 當 $x[n]$ 為 even function**，$x[n] = x[N-n]$

在做頻譜分析時，

N-point DFT 可以被 (floor(N/2) +1)-point DCT (type 1) 取代

$$X_C[m] = \sum_{n=0}^{Q} k_n x[n] \cos\left(\frac{\pi m n}{Q}\right), \qquad Q = floor(N/2),$$

$$\begin{cases} k_n = 1 & \text{,when } n = 0 \text{ or } N/2 \\ k_n = 2 & \text{,otherwise} \end{cases}$$

可以證明，當 $x[n]$ 為 even，$X_C[m] = X_F[m]$

（運算量減少將近一半）

Recover: $$x[n] = \frac{1}{N} \sum_{n=0}^{Q} k_m X_C[m] \cos\left(\frac{\pi m n}{Q}\right)$$

注意：和 JPEG 所用的 DCT (type 2) 並不相同

$$F[m] = \sqrt{\frac{2}{N}} C_m \sum_{n=0}^{N-1} f[n] \cos\frac{(n+1/2)m\pi}{N} \qquad \begin{array}{ll} C_0 = 1/\sqrt{2} & \\ C_m = 1 & \text{otherwise} \end{array}$$

(Proof) $$X_F[m] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi m n}{N}}$$

When $x[n] = x[N-n]$ ， N is even

(The case where N is odd can be proved in the similar way)

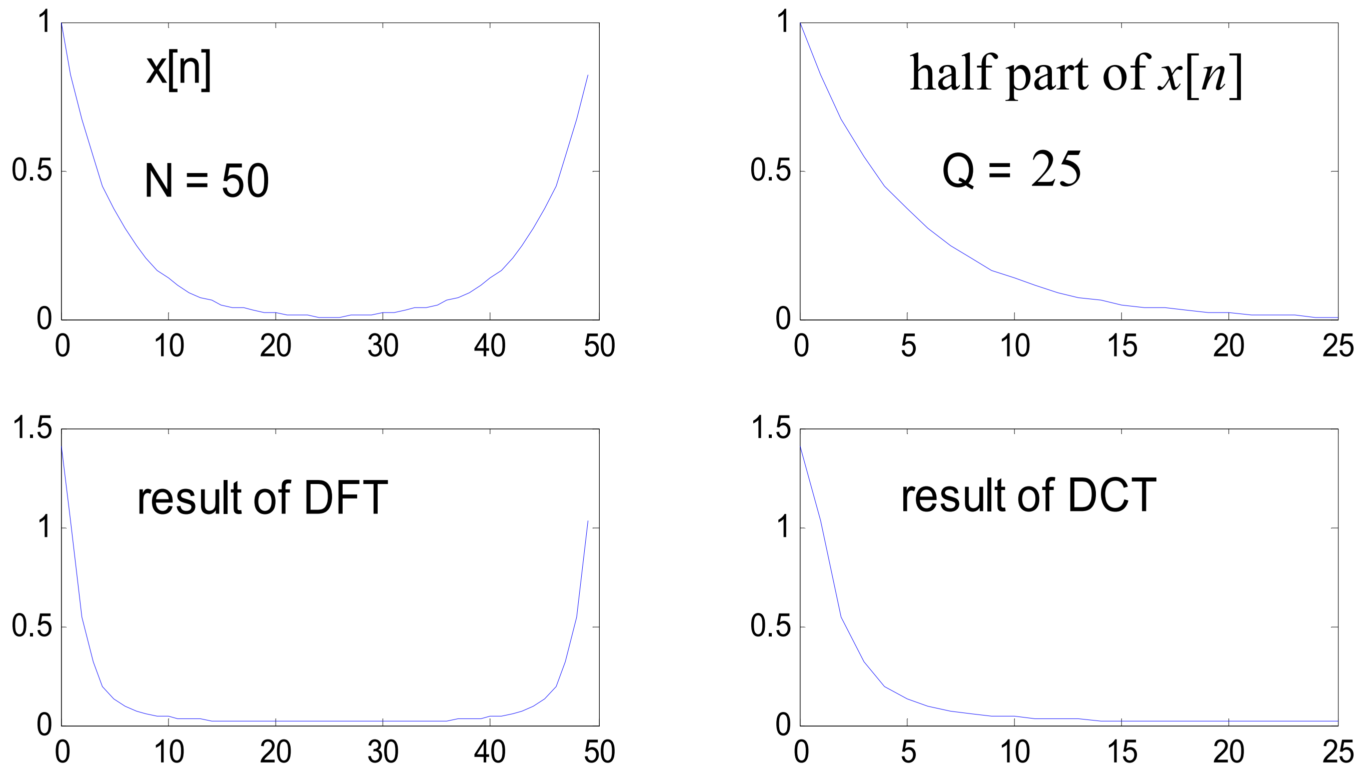
$$X_F[m] = x[0] + \sum_{n=1}^{N/2-1} x[n]\, e^{-j\frac{2\pi m n}{N}} + x\left[\frac{N}{2}\right] e^{-j\pi m} + \sum_{n=1}^{N/2-1} x[N-n]\, e^{-j\frac{2\pi m (N-n)}{N}}$$

$$= x[0] + \sum_{n=1}^{N/2-1} x[n]\, e^{-j\frac{2\pi m n}{N}} + x\left[\frac{N}{2}\right](-1)^m + \sum_{n=1}^{N/2-1} x[n]\, e^{j\frac{2\pi m (n)}{N}}$$

$$= x[0] + 2\sum_{n=1}^{N/2-1} x[n] \cos\left(\frac{2\pi m n}{N}\right) + x\left[\frac{N}{2}\right](-1)^m$$

$$= \sum_{n=0}^{N/2} k_n x[n] \cos\left(\frac{2\pi m n}{N}\right) \qquad \begin{cases} k_n = 1 & \text{,when } n = 0 \text{ or } N/2 \\ k_n = 2 & \text{,otherwise} \end{cases}$$

$$= X_C[m]$$


x[n]
N = 50
half part of $x[n]$
Q = 25
result of DFT
result of DCT
1
0.5
0
0
10
20
30
40
50
5
15
25
1.5

- **Case 2: 當 $x[n]$ 為 odd function**， $x[n] = -x[N-n]$

在做頻譜分析時，

N-point DFT 可以被 (N/2 −1)-point DST (type 1) 取代

$$X_S[m] = 2\sum_{n=1}^{Q-1} x[n] \sin\left(\frac{\pi m n}{Q}\right), \quad Q = N/2.$$

可以證明，當 $x[n]$ 為 odd， $X_S[m] = jX_F[m]$

(運算量減少將近一半)

Recover: $$x[n] = \frac{2}{N}\sum_{m=1}^{Q-1} X_S[m] \sin\left(\frac{\pi m n}{Q}\right)$$

- **Case 3: 當 *x*[*n*] 為 real function**，在做頻譜分析時，

Only when $N=2^k$, there is some fast algorithms.

N-point DFT 可以被 *N*-point DHT (type 1) 取代

$$X_H[m]=\sum_{n=0}^{N-1}x[n]cas\left(\frac{2\pi mn}{N}\right),\qquad \text{where } \operatorname{cas}(k)=\cos(k)+\sin(k)$$

比較： $\exp(-jk)=\cos(k)-j\sin(k)$

可以證明，若 $x[n]$ 為 real， $X_H[m]=real\{X_F[m]\}-imag\{X_F[m]\}$

(運算量減少將近一半)

Recover: $x[n]=\sum_{m=0}^{N-1}X_H[m]cas\left(\frac{2\pi mn}{N}\right)$

- 大部分的 convolution 仍然使用 DFT。

$$y[n] = x[n] * h[n]$$

$$y[n] = \mathrm{IDFT}\{\ \mathrm{DFT}(x[n]) \times \{\mathrm{DFT}(h[n])\ \}$$

思考：何時適合用 DCT 做 convolution ？
何時適合用 DST 做 convolution ？
何時適合用 DHT 做 convolution ？

附錄十二：論文英文常見的文法錯誤

(1) *** transform, *** equation, *** method, *** algorithm 在論文當中，當成是可數名詞，而非專有名詞 (除非是所有格的形態)。

可數名詞單數時，前面要要冠詞 (a 或 the)

Fourier transform is important for signal processing. （錯誤）

The Fourier transform is important for signal processing. （正確）

A Fourier transform is important for signal processing. （正確）

Fourier transforms are important for signal processing. （正確）

I have written the Matlab program of Parks-McClellan algorithm （錯誤）

I have written the Matlab program of the Parks-McClellan algorithm （正確）

(2) 若是所有格的形態，不必加冠詞

I have written the Matlab program of the Parks-McClellan's algorithm （錯誤）

I have written the Matlab program of Parks-McClellan's algorithm （正確）

(3) 論文視同正式的文件，對 not, is, are 不用縮寫

they’re（錯誤） they are（正確）

he’s（錯誤） he is（正確）

aren’t（錯誤） are not（正確）

don’t（錯誤） do not（正確）

can’t（錯誤） cannot（正確）

(4) Suppose, assume 後面要加關係代名詞

Suppose x is a large number.（錯誤）

Suppose that x is a large number.（正確）

(5) 每一個子句都有一個動詞，而且只有一個動詞

(6) In this paper, in this section, in this chapter 開頭的句子，應該用現在式，而非未來式

In this paper, the fast algorithm of DCT will be introduced.（錯誤）

In this paper, , the fast algorithm of DCT is introduced.（正確）

(7) 在 conclusion 當中回顧文章一內容，用<u>過去式</u>

(8) 敘述所引用的論文的內容，用<u>過去式</u>

In [10], the number theoretic transform was proposed.

(9) time domain, frequency domain 前面也加冠詞

in time domain（錯誤） in the time domain（正確）

(10) 不以 "this paper", "section *", "Ref. [*]" 當主詞用

This paper describes several concepts.（錯誤）

In this paper, several concepts are described.（正確）

Ref. [1] proposed the method.（錯誤）

In Ref. [1], Parks and McClellan proposed the method.（正確）

(11) 提及某個 equation 時，直接括號加數字即可

in equation (3)（錯誤） in (3)（正確）

提及某個 section, table, or figure 時，前面不加冠詞，而且常用大寫

in the section 4（錯誤） in Section 4（正確）

in the table 5（錯誤） in Table 4（正確）

(12) 寫科技論文不是寫文學作品，不要用高明、漂亮、但沒有保握的文法。

儘量用簡單而有把握的文法。

(13) 科技論文英文講求「長話短說」，儘量用精簡的文字來表達意思

(14) 用字儘量避免重覆

(15) Equations 也當成是文章的一部分，所以通常也要加標點符號

The formula of Newton's 2nd law is

$$F = ma.$$

要加標點符號

(16) 解釋 parameters 和 symbols 時，用 where 當關係代名詞

$x = 10t$ where x is the location of the object and t is time.

(17) 很重要的論文，投稿至國際學術期刊，又對自己的英文文法沒有十足的把握時

可以用網路上的論文編修服務，來修改文法上的錯誤

本系以及台大語言中心也經常有英文論文寫作相關的訓練課程，有志將來在學術界奮鬥的同學，可以多參與相關的課程

附錄十三：論文的標準格式與編輯論文技巧

註：這裡指的是一般 journal papers 和 conference papers 的格式。

然而，不同的 journals 和 conferences，對於格式的規定，也會稍有不同。投稿前，還是要細讀相關的規定。

(1) 變數使用斜體，矩陣或向量使用粗體

$f(x) = x^2 + 3x + 2.$ (f, x 皆用斜體)

$\mathbf{y} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ (**y**, **A** 皆用粗體)

(2) 段落的經常用「左右對齊」的格式

如果使用 Word ，可以按 常用 → 段落 → 對齊方式→ 左右對齊

或是按工具列中的 ☰

(3) Equation 的標號，經常用「定位點」的功能，讓標號的位置固定

如果使用 Word ，可以按 常用 → 段落 → 定位點 (在對話框左下角)，再設定定位點的位置

(4) 至於 equations 本身，通常置於這一行的中間，例如

$$F = ma. \tag{1}$$

Equations 和前一行以及後一行，皆要有足夠的距離。而且，equations 的後方常常要加逗號或句號 (以下一行是否為新的句子而定)。

(5) 標題(包括 papers 的標題以及每個 chapters 和 sections的標題) 當中，

每個單字的開頭一定要大寫，除了 (a) 介係詞 (b) 連詞 (c) 冠詞 以外。

若為第一個單字，即使是介係詞 ， 連詞，或冠詞，也要大寫

The Applications of the Fourier Transform in Daily Life

Fast Algorithms of the Wavelet Transform and JPEG2000

(6) 文章一定要包括

(a) Abstract,

(b) Introduction (通常是第一個 section)

(c) 內文

(d) Conclusions 或 Conclusions and Future Works (通常是最後一個 section)

(e) References

(7) 每一張圖 (figures)，每一張表 (tables) 都要編號，而且要附加文字說明。

如 Fig. 3 The result of the Fourier transform for a chirp signal.

若一張圖當中有很多個小圖，每個小圖也要編號 (a), (b), (c), (d) …..

(8) 同一個 equation，同一張圖，要放在同一頁，不分散於兩頁。

(9) 一般而言，Journal papers 的初稿，是 one column, double space 的格式。

在 Word 當中， double space 可以用後下的方法設定

常用 → 段落 → 行距 → 2倍行高

但有時， 2倍行高會讓初稿過於稀疏，在 Word 2007 當中可以用

版面配置→ 版面設定 → 文件格線 → 沒有格線

來讓文件看起來不會那麼稀疏，且不易超過規定的頁數。

(10) Conference papers 是 two columns, one space 的格式。有時 Journal papers 被接受後，也會要求改成 two columns, one space 的格式。

在Word 2007 ， two columns 可以用

版面配置 → 欄 → 二 (W)

來設定

(11) References 的編號，通常是按照在文章中出現的順序來排序

或者也可按照第一作者的 last name 的英文字母順序排序

(12) Reference 的寫法

(以 IEEE Transactions on Signal Processing 為例)

(A) Journal papers and conference papers

Authors (first name 或 middle name 只用一個字母代表), "title ," name of the journal (縮寫為佳), vol. *, no. *, pp. **~**, month, year.

使用縮寫

逗號在引號之前

加句號

只有第一個字母、專有名詞開頭、和縮為用大寫

範例：

S. Abe and J. T. Sheridan, "Optical operations on wave functions as the Abelian subgroups of the special affine Fourier transformation," *Opt. Lett.*, vol. 19, no. 22, pp. 1801-1803, 1994.

(B) Books

Authors (first name 或 middle name 只用一個字母代表), *title* (斜體, 字開頭大寫，不加引號), 第幾版 (非必需), 出版社, 出版地, year.

範例

H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing*, 1st Ed., John Wiley & Sons, New York, 2000.

(C) Websites

Authors, "title," available in http://網址.

範例

張智星, "Utility toolbox," available in http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/utility/.

◉ 11-C 計算 Linear Convolution

output: N+M-1 points

We know that when $y[n] = \overset{N}{x[n]} * \overset{M}{h[n]} = \sum_k x[n-k]h[k]$

then $y[n] = IFFT\left(FFT\{x[n]\}\, FFT\{h[n]\}\right)$

(*N* points) (*M* points)

(*P* points)

$\boxed{P \geq M + N - 1}$

Ex: M=100
N=10
P≥109

When P=112
x[n]=0 for 100≤n≤111
h[n]=0 for 10≤n≤111

2 個 FFT_{112} + 1個 112-point complex muls.

(FFT_{112}(h[n]) 可先算) $= 2MUL_{112} + 112 \times 3$

$= 2 \times 396 + 336$

$= \underline{\underline{1128}}$

But how do we implement it correctly?

How do we choose *P* ?

If length(x)= 256
length(h)=19 P=?

Note: When $y_1[n] = IFFT_P\left(FFT_P\{x_1[n]\}\, FFT_P\{h_1[n]\}\right)$

then $y_1[n] = \sum_{k=0}^{P-1} x_1[((n-k))_P]h_1[k]$ (circular convolution)

FFT_P: the *P*-point FFT $IFFT_P$: the *P*-point inverse FFT

$((a))_P$: *a* 除以 *P* 的餘數

When P=120

$2MUL_{120} + 120 \times 3$

$= 2 \times 380 + 360$

$= \underline{\underline{1120}}$

Linear convolution $\quad y[n]=\sum_{k=0}^{M-1} x[n-k]h[k]$

Circular convolution $\quad y_1[n]=\sum_{k=0}^{P-1} x_1[((n-k))_P]h_1[k]$

$$\boxed{x_1[n]=x[n] \quad for\ 0\le n<N, \quad x_1[n]=0 \quad for\ N\le n\le P-1}$$

$$\boxed{h_1[n]=h[n] \quad for\ 0\le n<M, \quad h_1[n]=0 \quad for\ M\le n<P-1}$$

When $\quad P\ge M+N-1$

$$y[n]=y_1[n] \quad for\ 0\le n<N+M-2,$$

$$y[n]=0 \quad for\ N+M-1\le n<P-1$$

When $y_1[n] = IFFT_P\left(FFT_P\{x_1[n]\}FFT_P\{h_1[n]\}\right)$

then $y_1[n] = \sum_{k=0}^{P-1} x_1[((n-k))_P]h_1[k]$

(Proof) Suppose that $X_1[m] = FFT_P\{x_1[n]\}$, $H_1[m] = FFT_P\{h_1[n]\}$

$$\frac{1}{P}\sum_{m=0}^{P-1} X_1[m]H_1[m]e^{j\frac{2\pi m}{P}n} = \frac{1}{P}\sum_{m=0}^{P-1}\sum_{k=0}^{P-1} x_1[k]e^{-j\frac{2\pi m}{P}k}\sum_{s=0}^{P-1} h_1[s]e^{-j\frac{2\pi m}{P}s}e^{j\frac{2\pi m}{P}n}$$

$$= \frac{1}{P}\sum_{k=0}^{P-1}\sum_{s=0}^{P-1} x_1[k]h_1[s]\sum_{m=0}^{P-1} e^{j\frac{2\pi(n-s-k)}{P}m}$$

$$= \frac{1}{P}\sum_{k=0}^{P-1}\sum_{s=0}^{P-1} g[k]h[s]P\delta_d[((n-s-k))_P]$$

$$= \sum_{k=0}^{P-1}\sum_{s=0}^{P-1} g[k]h[((n-k))_P]$$

Here we apply $\sum_{n=0}^{P-1} e^{j\frac{2\pi a}{P}n} = P\delta_d[((a))_P]$

$((a))_P$: the remainder of a after divided by P

[Discrete Circular Convolution and Discrete Linear Convolution]

A discrete linear time-invariant (LTI) system can always be expressed a discrete linear convolution:

$$\boxed{y[n] = x[n] * h[n] = \sum_{k=0}^{N-1} x[k]h[n-k]}$$

However, the convolution implemented by the DFT is the discrete circular convolution:

If

$$y_1[n] = IDFT\left(DFT\{x_1[n]\}\, DFT\{h_1[n]\}\right) = IDFT\left(X_1[m]H_1[m]\right)$$

then

$$\boxed{y_1[n] = x_1[n] *_c h_1[n] = \sum_{k=0}^{P-1} x_1[k]h_1[((n-k))_P]}$$

$((a))_P$: the remainder of a after divided by P

linear convolution: $y[n]=x[n]*h[n]=\sum_{k=0}^{N-1}x[k]h[n-k]$

circular convolution: $y_1[n]=x_1[n]*_c h_1[n]=\sum_{k=0}^{P-1}x_1[k]h_1[((n-k))_P]$

For example,

$$y[2]=x[0]h[2]+x[1]h[1]+x[2]h[0]+x[3]h[-1]+x[4]h[-2]+\cdots\cdots$$

$$y_1[2]=x_1[0]h_1[2]+x_1[1]h_1[1]+x_1[2]h_1[0]+x_1[3]h_1[P-1]+x_1[4]h_1[P-2]$$
$$+\cdots\cdots+x[P-1]h[3]$$

The condition where <u>the circular convolution</u> is equal to <u>the linear convolution</u>:

(i) $x[n]=0 \quad for\ n<0 \ \ or \ \ n\geq N$

$\boxed{x_1[n]=x[n] \quad for\ 0\leq n<N, \quad x_1[n]=0 \quad for\ N\leq n<P-1}$

(ii) $h[n]=0 \quad for\ n<0 \ \ or \ \ n\geq M$

$\boxed{h_1[n]=h[n] \quad for\ 0\leq n<M, \quad h_1[n]=0 \quad for\ M\leq n<P-1}$

(iii) $P\geq N+M-1$

The condition where the circular convolution is equal to the linear convolution:

(i) $x[n]=0 \quad for\ n<0\ \ or\ \ n\geq N$

$$\boxed{x_1[n]=x[n] \quad for\ 0\leq n<N, \quad x_1[n]=0 \quad for\ N\leq n<P-1}$$

(ii) $h[n]=0 \quad for\ n<0\ \ or\ \ n\geq M$

$$\boxed{h_1[n]=h[n] \quad for\ 0\leq n<M, \quad h_1[n]=0 \quad for\ M\leq n<P-1}$$

(iii) $P\geq N+M-1$

(Proof): $y_1[n]=\sum_{k=0}^{N-1} x_1[k]h_1[((n-k))_P]$

$$y_1[n]=x_1[0]h_1[n]+x_1[1]h_1[n-1]+\cdots+x_1[n]h_1[0]+\cancel{x_1[n+1]h_1[P-1]+}$$
$$\cancel{x_1[n+2]h_1[P-2]+\cdots}+x_1[P+n+1-M]h_1[M-1]+\cdots+x_1[P-1]h_1[n+1]$$
$$=x_1[0]h[n]+x_1[1]h[n-1]+\cdots+x_1[n]h[0]$$
$$\cancel{+x_1[P+n+1-M]h[M-1]+\cdots+x_1[P-1]h[n+1]}$$
$$=x[0]h[n]+x[1]h[n-1]+\cdots+x[n]h[0]=y[n]$$

(Since $P+n+1-M \geq P+1-M \geq N$)

$$y[n] = x[n] * h[n] = \sum_k x[n-k]h[k]$$

Linear Convolution 有幾種 Cases

Case A: Both $x[n]$ and $h[n]$ have infinite lengths. (impossible)

Case B: Both $x[n]$ and $h[n]$ have finite lengths.

Case C: $x[n]$ has infinite length but $h[n]$ has finite length.

Case D: $x[n]$ has finite length but $h[n]$ has infinite length.

We focus on Case B.

Case C and Case D can also be computed.

Case B: Both $x[n]$ and $h[n]$ have finite lengths.

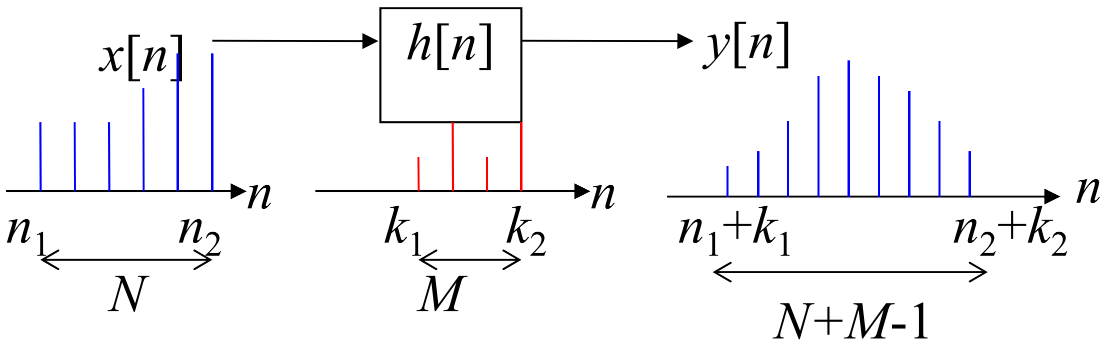
$x[n]$ 的範圍為 $n \in [n_1, n_2]$，大小為 $N = n_2 - n_1 + 1$

$h[n]$ 的範圍為 $n \in [k_1, k_2]$，大小為 $K = k_2 - k_1 + 1$

(n_1, k_1 can be nonzero)

$$y[n] = x[n] * h[n] = \sum_{k=k_1}^{k_2} x[n-k]h[k]$$

$y[n]$ 的範圍?



Convolution output 的範圍以及點數，
是學信號處理的人必需了解的常識

FFT implementation for Case B

$x_1[n] = x[n+n_1]$ for $n = 0, 1, 2, \ldots, N-1$

$x_1[n] = 0$ for $n = N, N+1, \ldots, P-1$ $P \geq N+M-1$

$h_1[n] = h[n+k_1]$ for $n = 0, 1, 2, \ldots, M-1$

$h_1[n] = 0$ for $n = M, M+1, \ldots, P-1$

$$y_1[n] = IFFT_P\left(FFT_P\{x_1[n]\}\,FFT_P\{h_1[n]\}\right)$$

$y[n] = y_1[n-n_1-k_1]$ for $n = n_1+k_1, n_1+k_1+1, n_1+k_1+2, \ldots\ldots, k_2+n_2$

i.e., $n-n_1-k_1 = 0, 1, \cdots\cdots, N+M-2$

取 output 的前面 $N+M-1$ 個點

Case C: $x[n]$ has finite length but $h[n]$ has infinite length

$x[n]$ 的範圍為 $n \in [n_1, n_2]$，範圍大小為 $N = n_2 - n_1 + 1$

$h[n]$ 無限長

$y[n] = \sum_{k} x[n-k]h[k]$ $y[n]$ 每一點都有值 (範圍無限大)

但我們只想求出 $y[n]$ 的其中一段

希望算出的 $y[n]$ 的範圍為 $n \in [m_1, m_2]$，範圍大小為 $M = m_2 - m_1 + 1$

$h[n]$ 的範圍？

要用多少點的 FFT？

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[k]$$

改寫成 $y[n] = x[n] * h[n] = \sum_{s=n_1}^{n_2} x[s]h[n-s]$

$$y[n] = x[n_1]h[n-n_1] + x[n_1+1]h[n-n_1-1] + x[n_1+2]h[n-n_1-2] + \cdots\cdots + x[n_2]h[n-n_2]$$

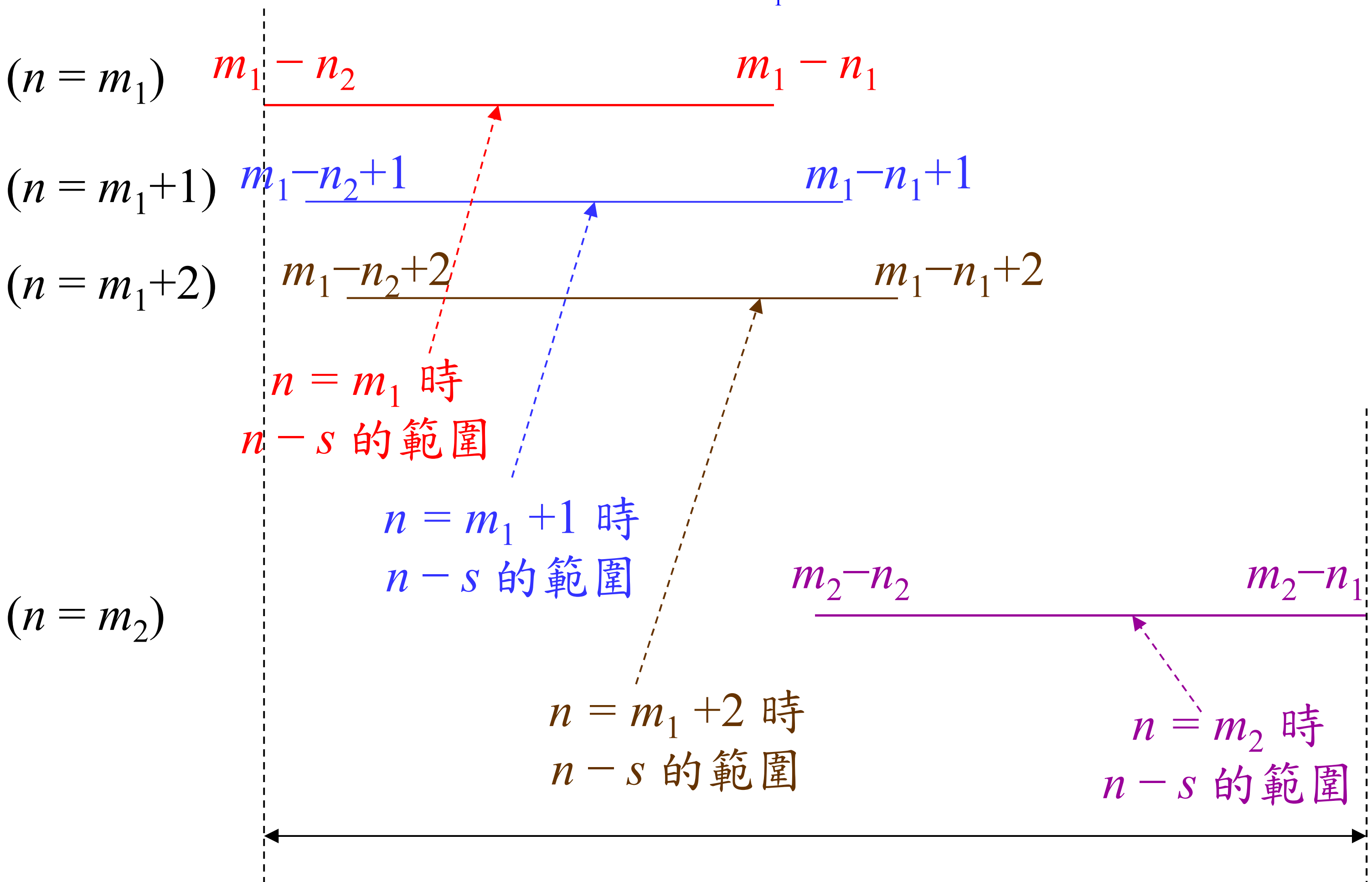
當 $n = m_1$

$$y[m_1] = x[n_1]h[m_1-n_1] + x[n_1+1]h[m_1-n_1-1] + x[n_1+2]h[m_1-n_1-2] + \cdots\cdots + x[n_2]h[m_1-n_2]$$

當 $n = m_2$

$$y[m_2] = x[n_1]h[m_2-n_1] + x[n_1+1]h[m_2-n_1-1] + x[n_1+2]h[m_2-n_1-2] + \cdots\cdots + x[n_2]h[m_2-n_2]$$

此圖為 $n-s$ 範圍示意圖 $$y[n]=\sum_{s=n_1}^{n_2} x[s]h[n-s]$$



有用到的 $h[k]$ 的範圍：$k \in [m_1 - n_2, m_2 - n_1]$

所以有用到的 $h[k]$ 的範圍是 $k \in [m_1 - n_2\,,\, m_2 - n_1\,]$

範圍大小為 $m_2 - n_1 - m_1 + n_2 + 1 = N + M - 1$

FFT implementation for Case 3

$x_1[n] = x[n + n_1]$ for $n = 0, 1, 2, \ldots, N-1$

$x_1[n] = 0$ for $n = N, N+1, N+2, \ldots\ldots, P-1$ $P \geq N + M - 1$

$h_1[n] = h[n + m_1 - n_2]$ for $n = 0, 1, 2, \ldots, L-1$

$$y_1[n] = IFFT_P\left(FFT_P\{x_1[n]\}\,FFT_P\{h_1[n]\}\right)$$

$y[n] = y_1[n - m_1 + N - 1]$ for $n = m_1, m_1+1, m_1+2, \ldots, m_2$

$$n - m_1 + N - 1 = N - 1, N, \cdots\cdots, N + M - 2$$

注意：$y[n]$ 只選 $y_1[n]$ 的第 N 個點到第 $N+M-1$ 個點

◉ 11-D Relations between the Signal Length and the Convolution Algorithm

Suppose that

$x[n]$: input, $h[n]$: the impulse response of the filter

length($x[n]$) = N, length($h[n]$) = M (Both of them have finite lengths)

number of real mul.:
$3NM$ Ex: $N=100$, $M=10$, $3MN=3000$

We want to compute

However, if
x, h are real

$$y[n]=\sum_{m=0}^{M-1} x[\overset{N}{n-m}]\,h[\overset{M}{m}] \quad , \qquad y[n] = x[n] * h[n].$$

The above convolution needs the P-point DFT, $P \geq M + N - 1$. MN

complexity: $O(P \log_2 P)$

$N=100$, $M=10$,
$MN=1000$

[Case 1]: When *M* is a very small integer:

Directly computing

Number of multiplications for directly computing: $N \times M$

- Number of real multiplications for directly computing:

$$3N \times M$$

real case NM

- When using $y[n] = IFFT_P\left(FFT_P\{x[n]\}\, FFT_P\{h[n]\}\right)$

Number of real multiplications

$$2MUL_P + 3P$$

2MUL$_p$+3P

MUL$_P$: the number of multiplications for the P-point DFT

When $3N \times M \leq 2MUL_P + 3P$,

it is proper to do directly computing instead of applying the DFT.

Example: $N = 126$, $M = 3$, (difference, edge detection)

$(3/2)\log_2 N = 10.4659$

When compute the number of real multiplications explicitly,

using direct implementation: $3N \times M = 1134$,

using the 128-point DFT:

using the 144-point DFT:

Although in usual "directly computing" is not a good idea for convolution implementation, in the cases where

(a) M is small

(b) The filter has some symmetric relation

using the directly computing method may be efficient for convolution implementation.

Example: edge detection

$\sum x[n-m]h[m]$ n: -3 -2 -2 0 1 2 3

$h[n] = [-0.1, -0.3, -0.6, 0, 0.6, 0.3, 0.1]$ for $n = -3 \sim 3$

N=100, MUL=300

$y[n] = x[n] * h[n]$

$= 0.1x[n-3] + 0.3x[n-2] + 0.6x[n-1] - 0.6x[n+1] - 0.3x[n+2] - 0.1x[n+3]$

$= 0.1(x[n-3] - x[n+3]) + 0.3(x[n-2] - x[n+2]) + 0.6(x[n-1] - x[n+1])$

$= 0.1(x[n-3] - x[n+3] - x[n-1] + x[n+1])$

$+ 0.3(x[n-2] - x[n+2] - x[n-1] + x[n+1]) + x[n-1] - x[n+1]$

Example: smooth filter

$h[n] = [0.1,\ 0.2,\ 0.4,\ 0.2,\ 0.1]$ for n = -2 ~ 2

$x[n] * [1\ \ 2\ \ 4\ \ 2\ \ 1]/10$

$= \left(x_1[n-2] + x_1[n+2] + 2\left(x_1[n-1] + x_1[n+1]\right) + 4x_1[n]\right)$

$x_1[n] = x[n]/10$

sectioned Convolution

[Case 2]: When M is not a very small integer but much less than N ($N >> M$):

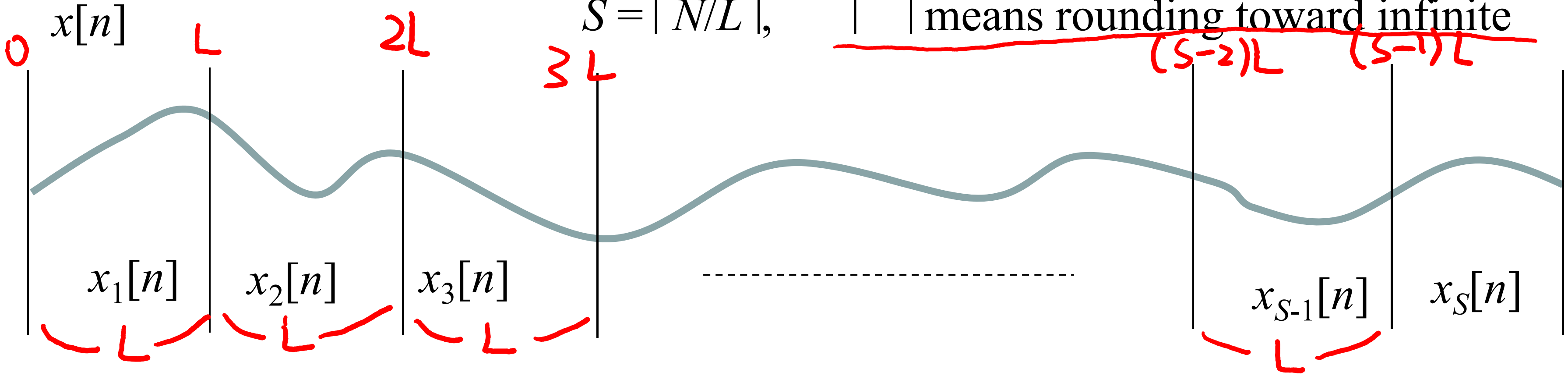
It is proper to divide the input $x[n]$ into several parts:

Each part has the size of L ($L > M$).

$x[n]$ ($n = 0, 1, \ldots, N-1$) $\rightarrow$ $x_1[n]$, $x_2[n]$, $x_3[n]$, ………, $x_S[n]$

ceil(N/L)

$S = \lceil N/L \rceil$, $\lceil \ \rceil$ means rounding toward infinite



Section 1 $\quad x_1[n] = x[n]$ $\quad$ for $n = 0, 1, 2, \ldots, L-1$,

n+L= L ~ 2L-1

Section 2 $\quad x_2[n] = x[n+L]$ $\quad$ for $n = 0, 1, 2, \ldots, L-1$,

:

Section s $\quad x_s[n] = x[n+(s\text{-}1)L]$ $\quad$ for $n = 0, 1, 2, \ldots, L-1$,

$s = 1, 2, 3, \ldots, S$

$$x[n] = \sum_{s=1}^{S} x_s[n-(s-1)L]$$

$$y[n] = x[n] * h[n] = \sum_{s=1}^{S} x_s[n-(s-1)L] * h[n] = \sum_{s=1}^{S} y_s[n-(s-1)L]$$

where $y_s[n] = x_s[n] * h[n] = \sum_{m=0}^{M-1} \underbrace{x_s[n-m]}_{\text{length} = L} \underbrace{h[m]}_{\text{length} = M}$

$s = 1, 2, \cdots S$

$S = \lceil \frac{N}{L} \rceil$

It should perform the P-point FFTs $2S$ times. Why?

$P \geq L+M-1$

Detail of Implementation

Suppose that the P-point DFT is applied for each section

$$x[n] = 0 \quad \text{when } n < 0 \text{ and } n \geq N$$

(1) First, determine $L = P - M + 1$

(2) $x_s[n] = x[(s-1)L + n]$ for $n = 0, 1, 2, \ldots, L-1,$ $\quad s = 1, 2, 3, \ldots, S$

$x_s[n] = 0$ for $n = L, L+1, \ldots, P-1$ $\quad S = \lceil N/L \rceil$

$h_1[n] = h[n]$ for $n = 0, 1, 2, \ldots, M-1,$

$h_1[n] = 0$ for $n = M, M+1, \ldots, P-1$

(3) Then calculate

MULP MULP 3·P known in prior

$$y_s[n] = IDFT_P\{DFT_P(x_s[n])DFT_P(h_1[n])\}$$

(4) Then, apply "overlapped addition"

$$y[n] = \sum_{s=1}^{S} y_s[n-(s-1)L]$$

sectioned convolution

運算量： $2S \times \mathrm{MUL}_P + 3S \times P$ $\quad S \approx N/L, \quad P \approx L+M-1$

MUL_P: the number of multiplications for the P-point DFT

運算量：(理論值) $2S \times \boxed{[(3P/2)\log_2 P]} + 3S \times P \quad S \approx N/L, \quad P \approx L+M-1$

$$\approx \frac{N}{L} 3(L+M-1)\left[\log_2(L+M-1)+1\right]$$

(linear with N) $\Theta(N)$

If L is fixed

運算量大約等於 $= N \times$ constant

math advantages

① complexity $\Theta(N)$ 運算量比較少

② fixed hardware (FFT 點數固定)

何時為 optimal? $\rightarrow \dfrac{\partial 運算量}{\partial L} = 0$

每一段長度都是固定的L，硬體的架構與需求量就是會固定的

$$N\frac{L-(L+M-1)}{L^2}\left[\log_2(L+M-1)+1\right] + N\frac{L+M-1}{L}\frac{1}{(L+M-1)\log 2} = 0$$

$$L = (M-1)\left[\log(L+M-1)+\log 2\right]$$

In practice, a computer program is applied to determine the optimal L.

How do we estimate the optimal L?

(1) Find L_0 to minimize $\dfrac{N}{L}3(L+M-1)\left[\log_2(L+M-1)+1\right]$

(2) Estimate P_0 from $P_0 = L_0+M-1$. Then several values of P around P_0 to make MUL_P smaller

(3) Calculate L, S, and the number of real multiplications for each possible P to find the optimal P and L.

Ex: $N=1000$, $M=19$

① FT method

$P \geq 1018$

$P=1152$, $\text{MUL}_{1152}=7088$

$2\text{MUL}_{1152}+3\times1152$

$=2\times7088+3456$

$=17632$

② direct

$3MN=57000$

③ $L_0=98$

$L_0+M-1=116$

(i) $P=120$

$L=P-M+1=102$

$S=\left\lceil\frac{N}{L}\right\rceil=10$

$20\times\text{MUL}_{120}+3\times10\times120$

$=20\times380+3600$

$=11200$

(ii) $P=144$, $L=P-M+1=126$

$S=\left\lceil\frac{N}{L}\right\rceil=8$

$16\times436+24\times144$

$=6976+3456=10432$

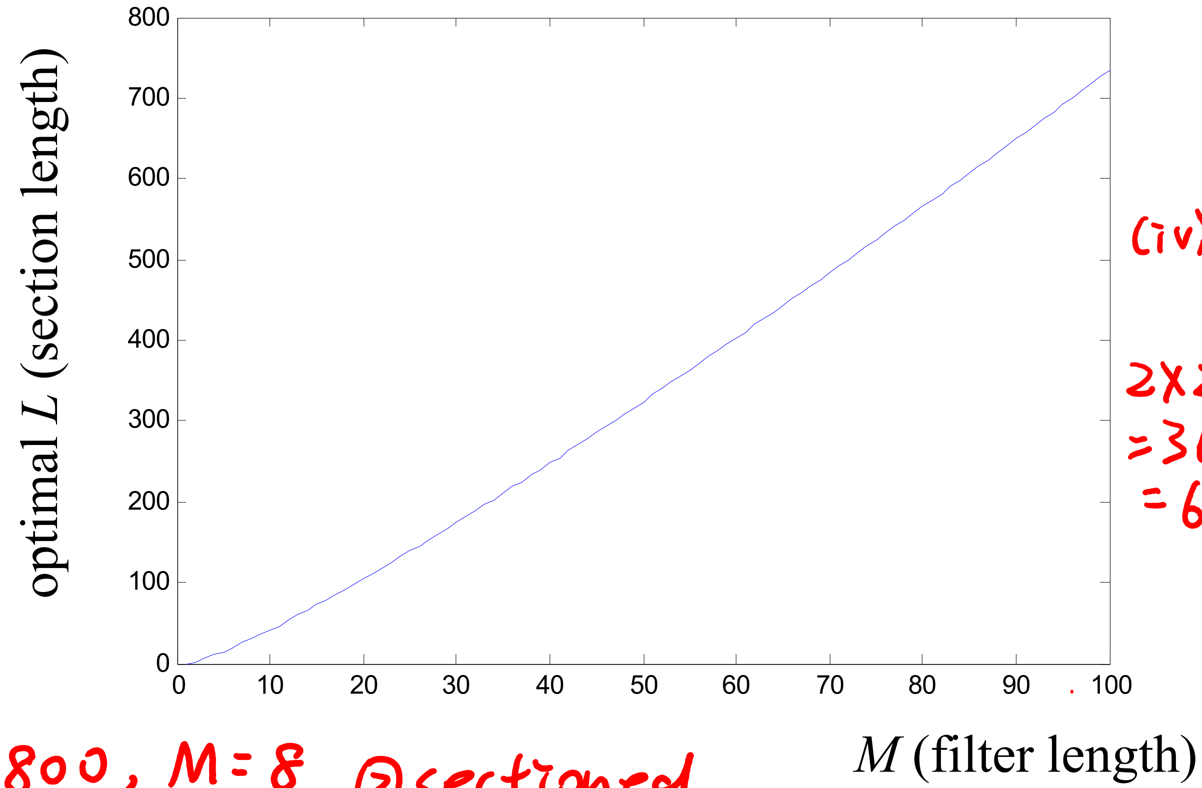
(iii) $P=168$, $L=P-M+L=150$

$S=\left\lceil\frac{1000}{L}\right\rceil=7$

$14\times580+21\times168$

$=8120+3528$

$=11648$



Ex: $N=800$, $M=8$

① FT based

$P \geq 807$

$P = 840$

$2\,MUL_{840} + 3 \times 840$

$= 2 \times 4580 + 2520 = 11680$

② direct

$3MN = 19200$

③ sectioned

$L_0 = 30$

$P_0 = L_0 + M - 1 = 37$

(i) $P=36$, $L = P-M+1 = 29$

$S = \lceil \frac{800}{29} \rceil = 28$

$2 \times 28 \times MUL_{36} + 3 \times 28 \times 36$

$= 56 \times 64 + 56 \times 54 = 6608$

(ii) $P=24$, $L = P-M+1 = 17$

$S = \lceil \frac{800}{17} \rceil = 48$

$2 \times 48 \times 28 + 3 \times 48 \times 24$

$= 2688 + 3456 = \underline{6144}$

(iii) $P=16$, $L = P-M+1 = 9$

$S = 89$

$2 \times 89 \times 20 + 3 \times 89 \times 16$

$= 89 \times 88 = 7832$

(iv) $P=48$, $L=41$

$S = 20$

$2 \times 20 \times 92 + 3 \times 20 \times 48$

$= 3680 + 2880$

$= 6560$

注意：

(1) Optimal section length is independent to N

(2) If M is a fixed constant, then the complexity is linear with N, i.e.,

$$O(N)$$

比較：使用原本方法時， complexity = $O((N+M-1)\log_2(N+M-1))$

(3) 實際上，需要考量 P-point FFT 的乘法量必需不多

$$P = L+M-1$$

例如，根據 page 442 的方法，算出當 $M = 10$ 時， $L = 41.5439$ 為 optimal

但實際上，應該選 $L = 39$，因為此時 $P = L+M-1 = 48$ 點的 DFT 有較少的乘法量

[Case 3]: When M has the same order as N

[Case 4]: When M is much larger than N

[Case 5]: When N is a very small integer

$x[n]$: N points $h[n]$: M points

How to compute $x[n] * h[n]$?

Case 1	Case 2	Case 3	Case 4	Case 5
direct implementation	sectioned convolution		sectioned convolution	direct implementation
M is a very small integer and x, h are real	$M << N$	$M \approx N$	$M >> N$	N is a very small integer x, h are real
		IFFT(FFT($x[n]$)FFT($h[n]$))		

M →

- **Sectioned Convolution for the Condition where One Sequence is Finite and the Other One is Infinite**

$$y[m] = \sum_n x[n]h[m-n]$$

$x[n] \neq 0$ for $n_1 \leq n \leq n_2$, length of $x[n] = N = n_2 - n_1 + 1$,

length of $h[n]$ is infinite,
and we want to calculate $y[m]$ for $m_1 \leq m \leq m_2$, $M = m_2 - m_1 + 1$.

Suppose that $M << N$.

In this case, we can try to partition $x[n]$ into several sections.

section 1: $x_1[n] = x[n]$ for $n = n_1 \sim n_1 + L - 1$, $x_1[n] = 0$ otherwise,
section 2: $x_2[n] = x[n]$ for $n = n_1 + L \sim n_1 + 2L - 1$, $x_2[n] = 0$ otherwise,

$\vdots$

section q: $x_q[n] = x[n]$ for $n = n_1 + (q-1)L \sim n_1 + qL - 1$, $x_q[n] = 0$ otherwise,

$\vdots$

Then we perform the convolution of $x_q[n] * h[n]$ for each of the sections by the method on pages 431-433.

(Since the length of $x_q[n]$ is L, it requires the P-point DFT,

$$P \geq L+M-1.$$

Its complexity and the optimal section length can also be determined by the formulas on page 442.

◉ 11-E Recursive Method for Convolution Implementation

$$y[n]=\sum_{m=0}^{N-1}x[n-m]a\cdot b^{m}=x[n]*a\cdot b^{n}u[n]$$

$u[n]$: unit step function

$$Y(z)=X(z)\frac{a}{1-bz^{-1}}$$

$$(1-bz^{-1})Y(z)=aX(z)$$

$$y[n]=by[n-1]+ax[n]$$

Only two multiplications required for calculating each output.

$$y[n] = x[n] * h[n]$$

$$h[n] = 0.25 \cdot 0.6^{|n|}$$

$$H(z) = \frac{0.25}{1-0.6z^{-1}} + \frac{0.25}{1-0.6z} - 0.25$$
$$= 0.25 \frac{1}{\frac{1.36}{0.64} - \frac{0.6}{0.64}\left(z^{-1} + z\right)}$$

12. Fast Algorithm 的補充

◎ 12-A Discrete Fourier Transform for Real Inputs

DFT: $$F[m]=\sum_{n=0}^{N-1} f[n]e^{-j\frac{2\pi}{N}mn}$$

當 $f[n]$ 為 real 時， $F[m] = F^*[N-m]$ *: conjugation

$$F^*(N-m) = \sum_{n=0}^{N-1} \overline{f[n]\, e^{-j\frac{2\pi}{N}n(N-m)}}$$

$$= \sum_{n=0}^{N-1} f[n]\, e^{j\frac{2\pi}{N}n(N-m)} = \sum_{n=0}^{N-1} f[n]\, e^{-j\frac{2\pi}{N}mn}\, \cancel{e^{j2\pi n}}$$

$$= F[m]$$

若我們要對兩個 real sequences $f_1[n]$ ，$f_2[n]$ 做 DFTs

Step 1: $f_3[n] = f_1[n] + j f_2[n]$

Step 2: $F_3[m] = \mathrm{DFT}\{f_3[n]\}$

Step 3: $F_1[m] = \dfrac{F_3[m] + F_3^*[N-m]}{2}$ $\qquad F_2[m] = \dfrac{F_3[m] - F_3^*[N-m]}{2j}$

只需一個 DFT

證明：由於 DFT 是一個 linear operation

$$F_3[m] = F_1[m] + jF_2[m]$$

又 $\quad F_1[m] = F_1{}^*[N-m] \qquad F_2[m] = F_2{}^*[N-m]$

$$F_3[m] + F_3^*[N-m] = F_1[m] + jF_2[m] + F_1^*[N-m] - jF_2^*[N-m]$$
$$= 2F_1[m]$$

$$F_3[m] - F_3^*[N-m] = j2F_2[m]$$

同理，當兩個 inputs 為

(1) pure imaginary

(2) one is real and another one is pure imaginary

時，也可以用同樣的方法將運算量減半

- 若 input sequence 為 even $f[n] = f[N\text{-}n]$，
  則 DFT output 也為 even $F[n] = F[N\text{-}n]$
- 若 input sequence 為 odd $f[n] = \text{-}f[N\text{-}n]$，
  則 DFT output 也為 odd $F[n] = \text{-}F[N\text{-}n]$

若有四個 input sequences，二個為 even and real，二個為 odd and real，
則乘法量可減為 1/4

[Corollary 1]

If we have known that the IDFTs of $F_1[m]$ and $F_2[m]$ are real, then the IDFTs of $F_1[m]$ and $F_2[m]$ can be implemented using only one IDFT:

(Step 1) $F_3[m] = F_1[m] + j\,F_2[m]$

(Step 2) $f_3[n] = \mathrm{IDFT}\{F_3[m]\}$

(Step 3) $f_1[n] = \mathcal{Re}\{f_3[n]\},\ f_2[n] = \mathcal{Im}\{f_2[n]\},$

[Corollary 2]

When $x[n]$ and $h[n]$ are both real, the computation loading of the convolution (or the sectioned convolution) of $x[n]$ and $h[n]$ can be halved.

◉ 12-B Converting into Convolution

一般的 linear operation:

$$z[m]=\sum_{n=0}^{N-1}x[n]k[m,n]$$ (習慣上，把 $k[m, n]$ 稱作 “kernel”)

$n = 0, 1, \ldots., N-1, \quad m = 0, 1, \ldots., M-1$

可以用矩陣 (matrix) 來表示

運算量為 MN

若為 linear time-invariant operation:

$$z[m]=\sum_{n=0}^{N-1}x[n]h[m-n]$$ $k[m, n] = h[m-n]$ (dependent on m, n 之間的差)

$n = 0, 1, \ldots., N-1, \quad m = 0, 1, \ldots., M-1$

$m-n$ 的範圍： 從 $1-N$ 到 $M-1$， 全長 $M+N-1$

運算量為 $L\log_2 L$, $\quad L \geq M+2N-2$

大致上，變成 convolution 後 總是可以節省運算量

例子A $z[m]=\sum_{n=0}^{N-1}x[n]2^{mn}$

可以改寫為

$$z[m]=2^{\frac{m^2}{2}}\sum_{n=0}^{N-1}\left\{x[n]2^{\frac{n^2}{2}}\right\}2^{\frac{-(n-m)^2}{2}}$$

convolution

運算量為 $M+N+L\log_2 L$

例子B Linear Canonical Transform

$$y(k)=A\int_{-\infty}^{\infty}e^{j\alpha k^2+j\beta kt+j\gamma t^2}x(t)\,dt$$

$$y(k)=A\int_{-\infty}^{\infty}e^{j(\alpha+\beta/2)k^2-j\frac{\beta}{2}(k-t)^2+j(\gamma+\beta/2)t^2}x(t)\,dt$$

$$y(k)=Ae^{j(\alpha+\beta/2)k^2}\int_{-\infty}^{\infty}e^{-j\frac{\beta}{2}(k-t)^2}\left[e^{j(\gamma+\beta/2)t^2}x(t)\right]dt$$

General rules：

當$k[m, n]$ 可以拆解成 $A[m] \times B[m-n] \times C[n]$

或 $$k[m,n] = \sum_s A_s[m] B_s[m-n] C_s[n]$$

即可以使用 convolution

◎ 12-C LUT

LUT (lookup table)

道理和背九九乘法表一樣

記憶體容量夠大時可用的方法

Problem: memory requirement

wasting energy

九九乘法表的例子

附錄十四　創意思考

New ideas 聽起來偉大，但大多是由既有的 ideas 變化而產生

(1) Combination

(2) Analogous

(3) Connection

(4) Generalization

(5) Simplification

(6) Reverse

註：感謝已過逝的李茂輝教授，他開的課「創造發明工程」，
　　讓我一生受用無窮

(7) Key Factor Analysis

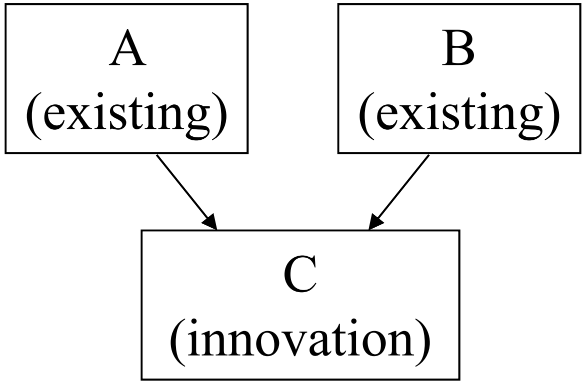
(8) 胡思亂想，純粹意外

(7) Key Factor

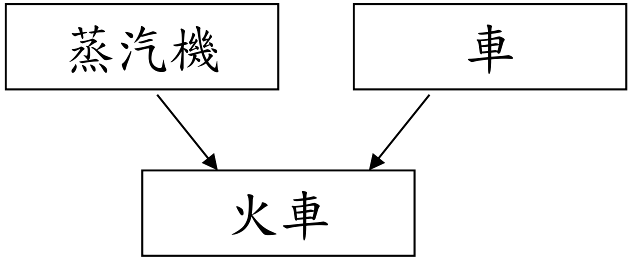
(8) Imagination

(9) 純粹意外

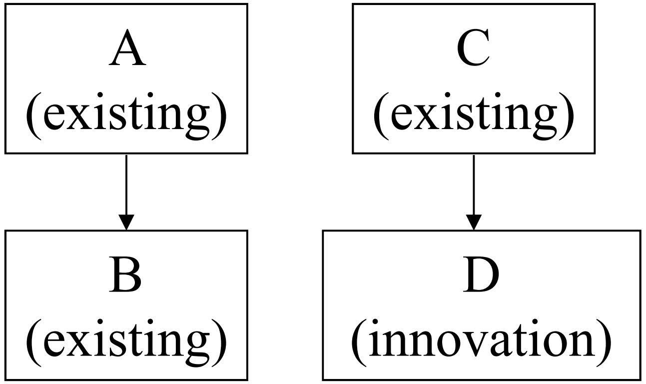
(1) Combination



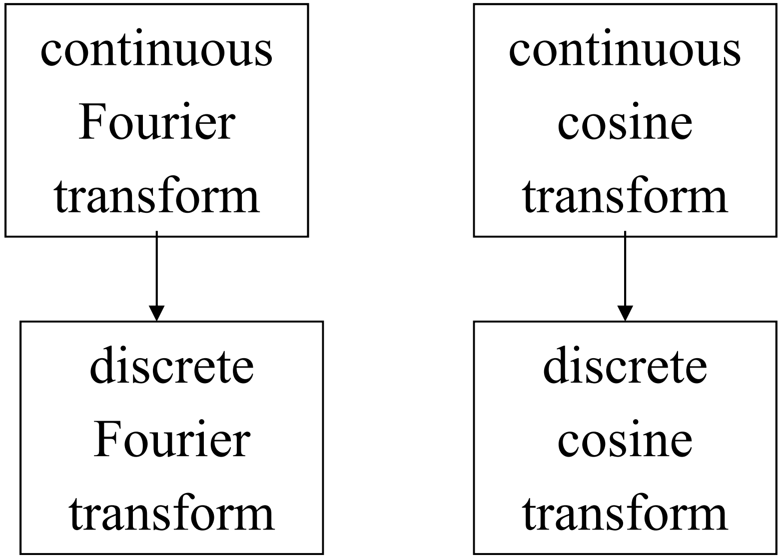
例：



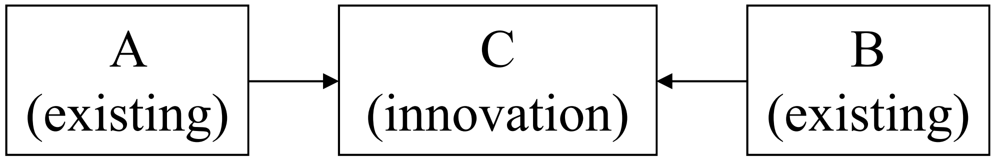
(2) Analog



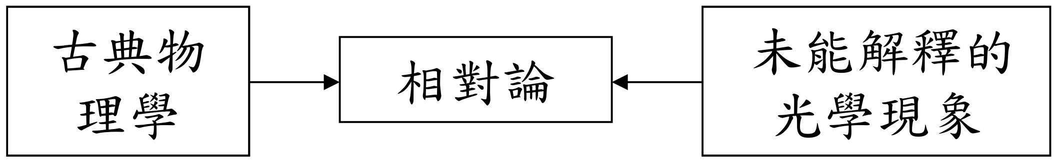
例：



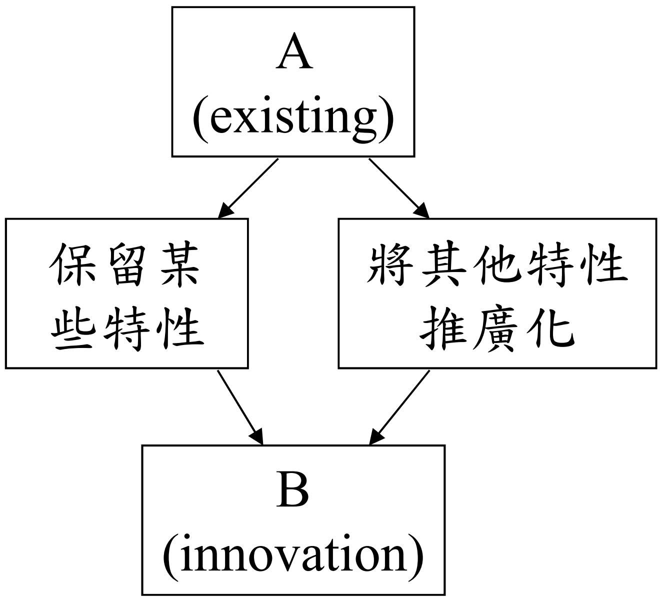
(3) Connection



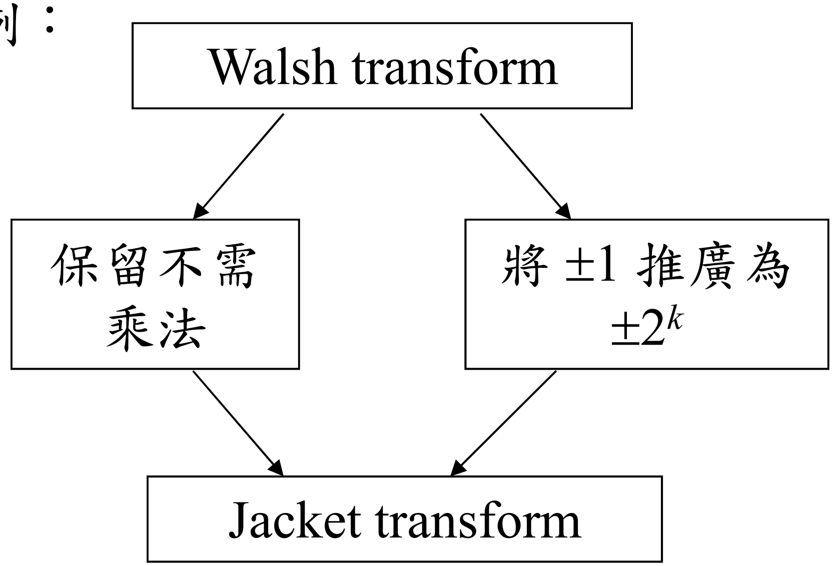
例：



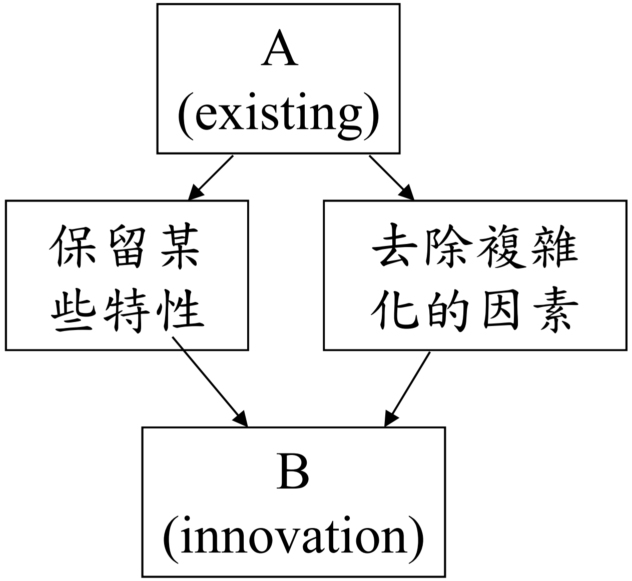
(4) Generalization



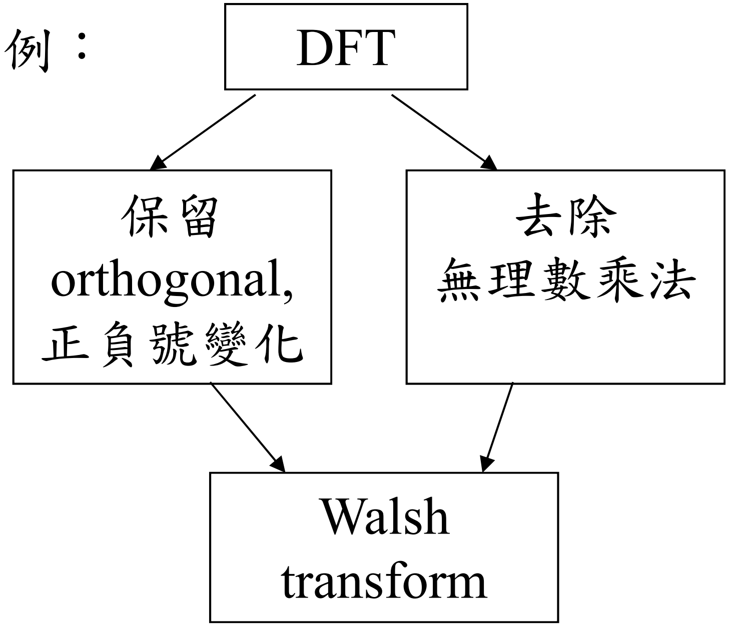
例：



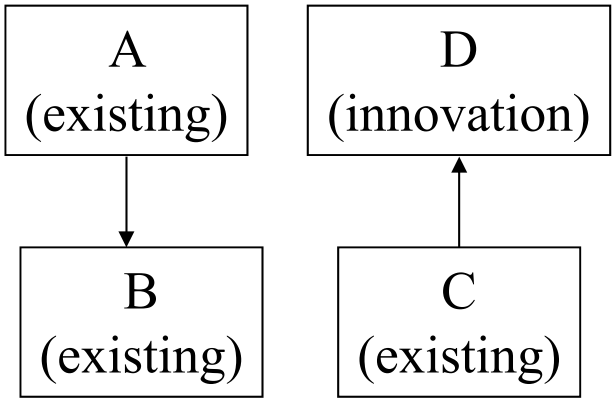
(5) Simplification

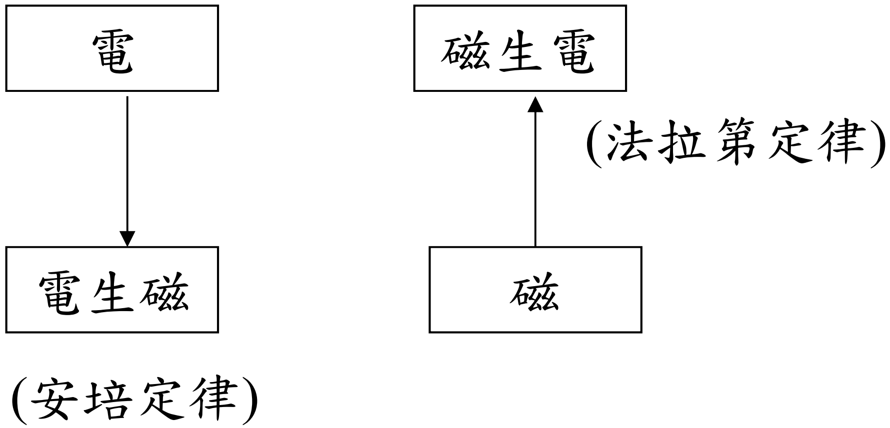


例：



(6) Reverse





個人做研究時，關於發明創造的心得

(1) 研究問題的第一步，往往是先想辦法把問題簡化

如果不將複雜的經濟問題簡化成二維供需圖，經濟學也就無從發展起來

如果不將電子學的問題簡化常小訊號模型，電子電路的許多問題都將難以解決

個人在研究影像處理時，也常常先針對 size 很小，且較不複雜的影像來做處理，成功之後再處理 size 較大且較複雜的影像

問題簡化之後，才比較容易對問題做分析，並提出改良之道

(2) 如果有好點子，趕快用筆記下來，

好點子是很容易稍縱即逝而忘記的。

(3) 練習多畫系統圖

系統圖畫得越多，越容易發現新的點子

(4) 其實，對台大的同學而言，提出 new ideas 並不難，但是要把 ideas 變成有用的、成功的 ideas，不可以缺少分析和解決問題的能力

很少有一個 new idea 一開始就 works well for any case，任何一個成功的創意，都是經由問題的分析，解決一連串的技術上的問題，才產生出來的

(5) 當心情放鬆時，想像力特別強，有助於發現意外的點子。

(6) 就短期而言，技術性的問題固然重要

但是就長期而言，不要因為技術上的困難，而否定了一個偉大的構想

大學以前的教育，是學習前人的智慧結晶

研究所的教育，是訓練創造發明和解決問題的能力