

# XIV. Number Theoretic Transform (NTT)

## ◎ 14-A Definition

### ◆ Number Theoretic Transform and Its Inverse

$$F(k) = \sum_{n=0}^{N-1} f(n) \alpha^{nk} \pmod{M}, k = 0, 1, 2, \dots, N-1$$

$$f(n) = N^{-1} \sum_{k=0}^{N-1} F(k) \alpha^{-nk} \pmod{M}, n = 0, 1, 2, \dots, N-1$$

$$f(n) \begin{matrix} \xrightarrow{NTT} \\ \xleftrightarrow{\quad} \\ \xleftarrow{INTT} \end{matrix} F(k)$$

Note :

(1)  $M$  is a **prime number**,  $(\text{mod } M)$ : 是指除以  $M$  的餘數

(2)  $N$  is a factor of  $M-1$

(Note: when  $N \neq 1$ ,  $N$  must be prime to  $M$ )

(3)  $N^{-1}$  is **an integer** that satisfies  $(N^{-1})N \pmod{M} = 1$

(When  $N = M-1$ ,  $N^{-1} = M-1$ )

(4)  $\alpha$  is a root of unity of order  $N$

$$\alpha^N = 1 \pmod{M}$$

$$\alpha^k \neq 1 \pmod{M}, k = 1, 2, \dots, N-1$$

When  $\alpha$  satisfies the above equations and  $N = M-1$ , we call  $\alpha$  the “primitive root”.

$$\alpha^k \neq 1 \pmod{M} \quad \text{for } k = 1, 2, \dots, M-2$$

$$\alpha^{M-1} = 1 \pmod{M}$$

$\alpha^{-1}$  的求法與  $N^{-1}$  相似

$\alpha^{-1}$  is an integer that satisfies  $(\alpha^{-1})\alpha \pmod{M} = 1$

Example 1:

$$M = 5 \quad \alpha = 2 \quad \alpha^1 = 2 \pmod{5} \quad \alpha^2 = 4 \pmod{5} \quad \alpha^3 = 3 \pmod{5} \quad \alpha^4 = 1 \pmod{5}$$

When  $N = 4$

$$\begin{bmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \\ 1 & 3 & 4 & 2 \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \\ f[2] \\ f[3] \end{bmatrix}$$

When  $N = 2$

$$\begin{bmatrix} F[0] \\ F[1] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \end{bmatrix}$$

Example 2:

$M = 7$  ,  $N = 6$  :  $\alpha$  cannot be 2 but can be 3.

$$\alpha = 2: \alpha^1 = 2 \pmod{7} \quad \alpha^2 = 4 \pmod{7} \quad \alpha^3 = 1 \pmod{7}$$

$$\alpha = 3: \alpha^1 = 3 \pmod{7} \quad \alpha^2 = 2 \pmod{7} \quad \alpha^3 = 6 \pmod{7}$$

$$\alpha^4 = 4 \pmod{7} \quad \alpha^5 = 5 \pmod{7} \quad \alpha^6 = 1 \pmod{7}$$

Advantages of the NTT:

Disadvantages of the NTT:

## ◎ 14-B 餘數的計算

(1)  $x \pmod{M}$  的值，必定為  $0 \sim M-1$  之間

$$(2) a + b \pmod{M} = \{a \pmod{M} + b \pmod{M}\} \pmod{M}$$

$$\text{例： } 78 + 123 \pmod{5} = 3 + 3 \pmod{5} = 1$$

(Proof): If  $a = a_1M + a_2$  and  $b = b_1M + b_2$ , then

$$a + b = (a_1 + b_1)M + a_2 + b_2$$

$$(3) a \times b \pmod{M} = \{a \pmod{M} \times b \pmod{M}\} \pmod{M}$$

$$\text{例： } 78 \times 123 \pmod{5} = 3 \times 3 \pmod{5} = 4$$

(Proof): If  $a = a_1M + a_2$  and  $b = b_1M + b_2$ , then

$$a \times b = (a_1 b_1 M + a_1 b_2 + a_2 b_1)M + a_2 b_2$$

在 Number Theory 當中

只有  $M^2$  個可能的加法， $M^2$  個可能的乘法

可事先將加法和乘法的結果存在記憶體當中

需要時再“LUT”

LUT : lookup table

## © 14-C Properties of Number Theoretic Transforms

### P.1) Orthogonality Principle

$$S_N = \sum_{n=0}^{N-1} \alpha^{nk} \alpha^{-n\ell} = \sum_{n=0}^{N-1} \alpha^{n(k-\ell)} = N \cdot \delta_{k,\ell}$$

proof : for  $k = \ell$ ,  $S_N = \sum_{n=0}^{N-1} \alpha^0 = N$

for  $k \neq \ell$ ,  $(\alpha^{k-\ell} - 1) S_N = (\alpha^{k-\ell} - 1) \sum_{n=0}^{N-1} \alpha^{n(k-\ell)} = \alpha^{N(k-\ell)} - 1 = 1 - 1 = 0$

$\because \alpha^{k-\ell} \neq 1 \quad \therefore S_N = 0$

### P.2) The NTT and INTT are exact inverse

proof :

$$g(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) \alpha^{-nk} = \frac{1}{N} \sum_{k=0}^{N-1} \left( \sum_{\ell=0}^{N-1} f(\ell) \alpha^{\ell k} \right) \alpha^{-nk}$$

$$= \frac{1}{N} \sum_{\ell=0}^{N-1} f(\ell) \sum_{k=0}^{N-1} \alpha^{(\ell-n)k} = \frac{1}{N} \sum_{\ell=0}^{N-1} f(\ell) \cdot N \delta_{\ell,n} = f(n)$$



### P.3) Symmetry

$$f(n) = f(N-n) \quad \stackrel{\text{NTT}}{\Leftrightarrow} \quad F(k) = F(N-k)$$

$$f(n) = -f(N-n) \quad \stackrel{\text{NTT}}{\Leftrightarrow} \quad F(k) = -F(N-k)$$

### P.4) INNT from NTT

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) \alpha^{-nk} = \frac{1}{N} \sum_{(-k)=0}^{N-1} F(-k) \alpha^{nk} = \text{NTT of } \frac{1}{N} F(-k)$$

Algorithm for calculating the INNT from the NTT

(1)  $F(-k)$  : time reverse

$$F_0, F_1, F_2, \dots, F_{N-1} \xrightarrow[\text{reverse}]{\text{time}} F_0, F_{N-1}, \dots, F_2, F_1$$

(2) NTT[  $F(-k)$  ]

(3) 乘上  $\frac{1}{N} = M - 1$

### P.5) Shift Theorem

$$f(n + \ell) \leftrightarrow F(k) \alpha^{-\ell k}$$

$$f(n) \alpha^{n\ell} \leftrightarrow F(k + \ell)$$



### P.6) Circular Convolution (the same as that of the DFT)

If  $f(n) \leftrightarrow F(k)$

$$g(n) \leftrightarrow G(k)$$

then  $f(n) \otimes g(n) \leftrightarrow F(k)G(k)$

i.e.,  $f(n) \otimes g(n) = INTT \{ NTT[f(n)] NTT[g(n)] \}$

$$f(n) \cdot g(n) \leftrightarrow \frac{1}{N} F(k) \otimes G(k)$$

### P.7) Parseval's Theorem

$$N \sum_{n=0}^{N-1} f(n) f(-n) = \sum_{k=0}^{N-1} F^2(k)$$

$$N \sum_{n=0}^{N-1} f(n)^2 = \sum_{k=0}^{N-1} F(k)F(-k)$$

**P.8) Linearity**

$$a f(n) + b g(n) \leftrightarrow a F(k) + b G(k)$$

**P.9) Reflection**

$$\text{If } f(n) \leftrightarrow F(k) \quad \text{then } f(-n) \leftrightarrow F(-k)$$

## © 14-D Efficient FFT-Like Structures for Calculating NTTs

- If  $N$  (transform length) is a power of 2, then the radix-2 FFT butterfly algorithm can be used for efficient calculation for NTT.

Decimation-in-time NTT

Decimation-in-frequency NTT

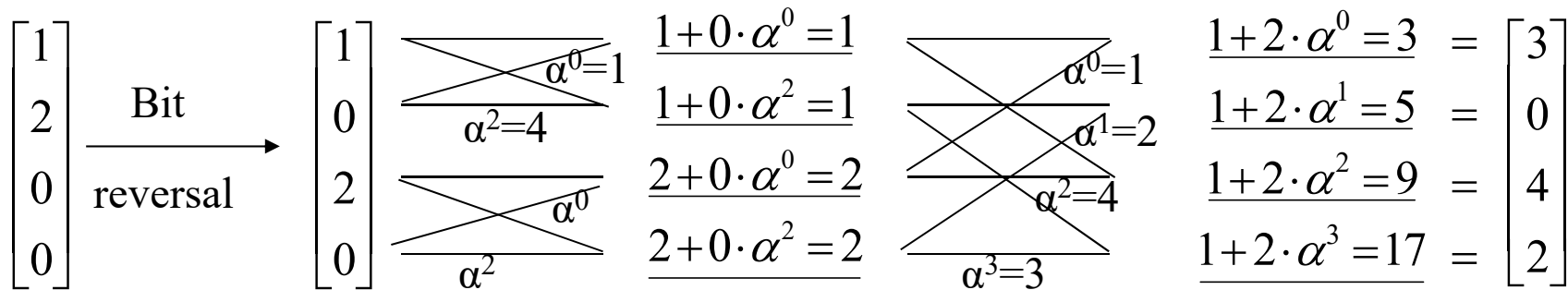
- The prime factor algorithm can also be applied for NTTs.

$$\begin{aligned}
F(k) &= \sum_{n=0}^{N-1} f(n) \alpha^{nk} = \sum_{r=0}^{\frac{N-1}{2}} f(2r) \alpha^{2rk} + \sum_{r=0}^{\frac{N-1}{2}} f(2r+1) \alpha^{(2r+1)k} \\
&= \sum_{r=0}^{\frac{N-1}{2}} f(2r) (\alpha^2)^{rk} + \alpha^k \sum_{r=0}^{\frac{N-1}{2}} f(2r+1) (\alpha^2)^{rk} \\
&= \begin{cases} G(k) + \alpha^k H(k) & , 0 \leq k \leq \frac{N}{2} - 1 \\ G(k - \frac{N}{2}) + \alpha^k H(k - \frac{N}{2}) & , \frac{N}{2} \leq k \leq N \end{cases}
\end{aligned}$$

where  $G(k) = \sum_{r=0}^{N/2-1} f(2r) (\alpha^2)^{rk}$      $H(k) = \sum_{r=0}^{N/2-1} f(2r+1) (\alpha^2)^{rk}$

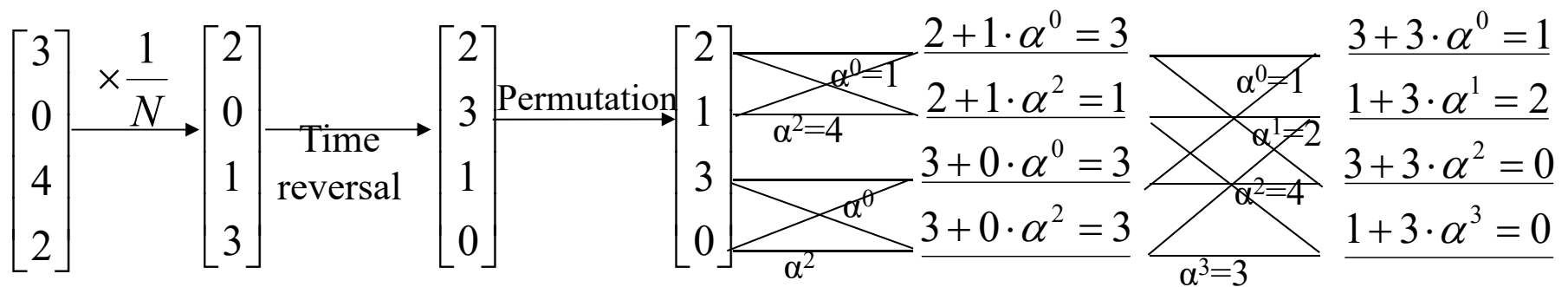
One  $N$ -point NTT  $\longrightarrow$  Two  $(N/2)$ -point NTTs  
plus twiddle factors

Original sequence  $f(n) = (1, 2, 0, 0)$   $N = 4, M = 5$   
 Permutation  $(1, 0, 2, 0)$   
 After the 1<sup>st</sup> stage  $(1, 1, 2, 2)$   
 After the 2<sup>nd</sup> stage  $F(k) = (3, 0, 4, 2)$



Inverse NTT by Forward NTT :

- 1)  $1/N$   $F(-k) \times \frac{1}{N}$  ( $4^{-1} = 4$ )
- 2) Time reversal
- 3) permutation
- 4) After first stage
- 5) After 2<sup>nd</sup> stage



## ◎ 14-E Convolution by NTT

假設  $x[n] = 0$  for  $n < 0$  and  $n \geq K$ ,  $h[n] = 0$  for  $n < 0$  and  $n \geq H$

要計算  $x[n] * h[n] = z[n]$

且  $z[n]$  的值可能的範圍是  $0 \leq z[n] < A$  (more general,  $A_1 \leq z[n] < A_1 + T$ )

(1) 選擇  $M$  (the prime number for the modulus operator), 滿足

(a)  $M$  is a prime number, (b)  $M \geq \max(H+K, A)$

(2) 選擇  $N$  (NTT 的點數), 滿足

(a)  $N$  is a factor of  $M-1$ , (b)  $N \geq H+K - 1$

(3) 添 0:  $x_1[n] = x[n]$  for  $n = 0, 1, \dots, K-1$ ,  
 $x_1[n] = 0$  for  $n = K, K+1, \dots, N-1$   
 $h_1[n] = h[n]$  for  $n = 0, 1, \dots, H-1$ ,  
 $h_1[n] = 0$  for  $n = H, H+1, \dots, N-1$



$$(4) X_1[m] = \text{NTT}_{N,M}\{x_1[n]\}, \quad H_1[m] = \text{NTT}_{N,M}\{h_1[n]\}$$

$\text{NTT}_{N,M}$  指  $N$ -point 的 DFT (mod  $M$ )

$$(5) Z_1[m] = X_1[m]H_1[m], \quad z_1[n] = \text{INTT}_{N,M}\{Z_1[m]\},$$

$$(6) z[n] = z_1[n] \text{ for } n = 0, 1, \dots, H+K-1$$

(移去  $n = H+K, H+K+1, \dots, N-1$  的點)

(More general, if we have estimated the range of  $z[n]$  should be  $A_1 \leq z[n] < A_1 + T$ , then

$$z[n] = ((z_1[n] - A_1))_M + A_1$$

適用於 (1)  $x[n]$  ,  $h[n]$  皆為整數

(2)  $\text{Max}(z[n]) - \text{min}(z[n]) < M$  的情形。

Consider the convolution of  $(1, 2, 3, 0) * (1, 2, 3, 4)$

Choose  $M = 17, N = 8$  , 結果為 :

•  $\text{Max}(z[n]) - \text{min}(z[n])$  的估測方法

假設  $x_1 \leq x[n] \leq x_2$ ,  $z[n] = x[n] * h[n] = \sum_{m=0}^{H-1} h[m]x[n-m]$

則  $\text{Max}(z[n]) - \text{min}(z[n]) = (x_2 - x_1) \sum_{n=0}^{H-1} |h[n]|$

(Proof):  $\text{Max}(z[n]) = \sum_{m=0}^{H-1} h_1[m]x_2 + \sum_{m=0}^{H-1} h_2[m]x_1$

where  $h_1[m] = h[m]$  when  $h[m] > 0$ ,  $h_1[m] = 0$  otherwise

$h_2[m] = h[m]$  when  $h[m] < 0$ ,  $h_2[m] = 0$  otherwise

$$\text{min}(z[n]) = \sum_{m=0}^{H-1} h_1[m]x_1 + \sum_{m=0}^{H-1} h_2[m]x_2$$

$$\text{Max}(z[n]) - \text{min}(z[n]) = \sum_{m=0}^{H-1} h_1[m](x_2 - x_1) + \sum_{m=0}^{H-1} h_2[m](x_1 - x_2)$$

$$= (x_2 - x_1) \left\{ \sum_{m=0}^{H-1} h_1[m] - \sum_{m=0}^{H-1} h_2[m] \right\} = (x_2 - x_1) \sum_{m=0}^{H-1} |h[m]|$$

## © 14-F Special Numbers

**Fermat Number** :  $M = 2^{2^p} + 1$

$$P = 0, 1, 2, 3, 4, 5, \dots$$

$$M = 3, 5, 17, 257, 65537, \dots$$

**Mersenne Number** :  $M = 2^p - 1$

$$P = 1, 2, 3, 5, 7, 13, 17, 19$$

$$M = 1, 3, 7, 31, 127, 8191, \dots$$

If  $M = 2^p - 1$  is a prime number,  $p$  must be a prime number.

However, if  $p$  is a prime number,  $M = 2^p - 1$  may not be a prime number.

The modulus operations for Mersenne and Fermat prime numbers are very easy for implementation.

$$2^k \pm 1$$

Example: 25 mod 7

$$\begin{array}{r}
 11 \\
 100a \overline{)11001} \\
 \underline{100a} \\
 1011 \\
 \underline{100a} \\
 12 \\
 \downarrow \\
 100
 \end{array}
 \qquad a = -1$$

## © 14-G Complex Number Theoretic Transform (CNT)

The integer field  $Z_M$  can be extended to complex integer field

If the following equation does not have a sol. in  $Z_M$

$$x^2 = -1 \pmod{M} \quad \text{無解}$$

This means (-1) does not have a square root

When  $M = 4k + 1$ , there is a solution for  $x^2 = -1 \pmod{M}$ .

When  $M = 4k + 3$ , there is no solution for  $x^2 = -1 \pmod{M}$ .

For example, when  $M = 13$ ,  $8^2 = -1 \pmod{13}$ .

$$2^1 = 2, \quad 2^2 = 4, \quad 2^3 = 8, \quad 2^4 = 3, \quad 2^5 = 6, \quad 2^6 = 12 = -1,$$

$$2^7 = 11, \quad 2^8 = 9, \quad 2^9 = 5, \quad 2^{10} = 10, \quad 2^{11} = 7, \quad 2^{12} = 1$$

When  $M = 11$ , there is no solution for  $x^2 = -1 \pmod{M}$ .

If there is no solution for  $x^2 = -1 \pmod{M}$ , we can define an imaginary number  $i$  such that

$$i^2 = -1 \pmod{M}$$

Then, “ $i$ ” will play a similar role over finite field  $Z_M$  such that plays over the complex field.

$$(a + i b) \pm (c + i d) = (a \pm c) + i (b \pm d)$$

$$\begin{aligned} (a + i b) \cdot (c + i d) &= ac + i^2 bd + i bc + i ad \\ &= (ac - bd) + i (bc + ad) \end{aligned}$$

## ◎ 14-H Applications of the NTT

NTT 適合作 convolution

但是有不少的限制

新的應用： encryption (密碼學)

CDMA



## References:

- (1) R. C. Agavard and C. S. Burrus, "Number theoretic transforms to implement fast digital convolution," *Proc. IEEE*, vol. 63, no. 4, pp. 550-560, Apr. 1975.
- (2) T. S. Reed & T. K. Truoay, "The use of finite field to compute convolution," *IEEE Trans. Info. Theory*, vol. IT-21, pp.208-213, March 1975
- (3) E. Vegh and L. M. Leibowitz, "Fast complex convolution in finite rings," *IEEE Trans ASSP*, vol. 24, no. 4, pp. 343-344, Aug. 1976.
- (4) J. H. McClellan and C. M. Rader, *Number Theory in Digital Signal Processing*, Prentice-Hall, New Jersey, 1979.
- (5) 華羅庚, "數論導引," 凡異出版社, 1997。

# XIV. Orthogonal Transform and Multiplexing

## © 14-A Orthogonal and Dual Orthogonal

Any  $M \times N$  discrete linear transform can be expressed as the matrix form:

$$\begin{array}{c}
 \left[ \begin{array}{c} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[M-1] \end{array} \right] = \left[ \begin{array}{ccccc} \phi_0^*[0] & \phi_0^*[1] & \phi_0^*[2] & \cdots & \phi_0^*[N-1] \\ \phi_1^*[0] & \phi_1^*[1] & \phi_1^*[2] & \cdots & \phi_1^*[N-1] \\ \phi_2^*[0] & \phi_2^*[1] & \phi_2^*[2] & \cdots & \phi_2^*[N-1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{M-1}^*[0] & \phi_{M-1}^*[1] & \phi_{M-1}^*[2] & \cdots & \phi_{M-1}^*[N-1] \end{array} \right] \left[ \begin{array}{c} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{array} \right] \\
 \mathbf{Y} = \qquad \qquad \qquad \mathbf{A} \qquad \qquad \qquad \mathbf{X}
 \end{array}$$

$$y[m] = \langle x[n], \phi_m[n] \rangle = \sum_{n=0}^{N-1} x[n] \phi_m^*[n]$$

$\uparrow$   
 inner product

Orthogonal:  $\langle \phi_k[n], \phi_h[n] \rangle = \sum_{n=0}^{N-1} \phi_k[n] \phi_h^*[n] = 0$  **when  $k \neq h$**

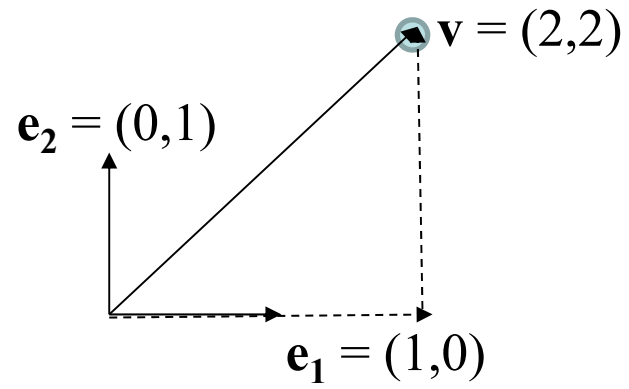
orthogonal transforms 的例子：

- discrete Fourier transform
- discrete cosine, sine, Hartley transforms
- Walsh Transform, Haar Transform
- discrete Legendre transform
- discrete orthogonal polynomial transforms  
Hahn, Meixner, Krawtchouk, Charlier

為什麼在信號處理上，我們經常用 orthogonal transform?

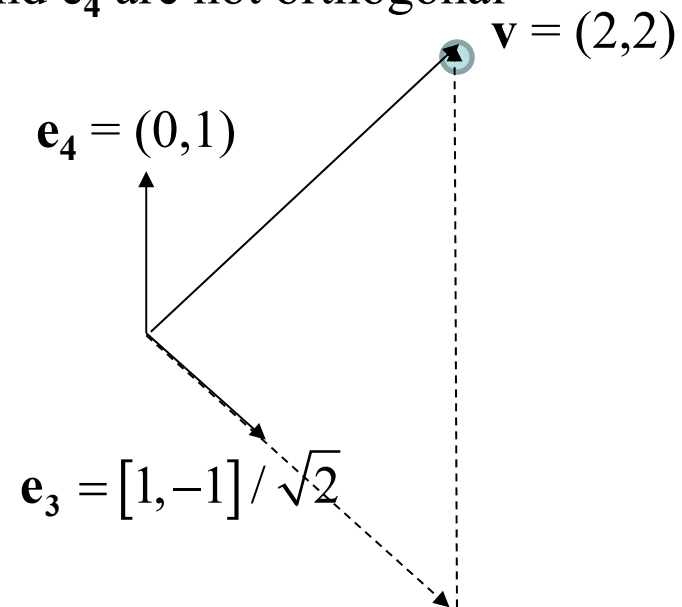
**Orthogonal transform** 最大的好處何在？

$\mathbf{e}_1$  and  $\mathbf{e}_2$  are orthogonal



$$\mathbf{v} = 2\mathbf{e}_1 + 2\mathbf{e}_2$$

$\mathbf{e}_3$  and  $\mathbf{e}_4$  are not orthogonal



$$\mathbf{v} = 2\sqrt{2}\mathbf{e}_3 + 4\mathbf{e}_4$$

- If partial terms are used for reconstruction

for orthogonal case,

perfect reconstruction: 
$$x[n] = \sum_{m=0}^{N-1} C_m^{-1} y[m] \phi_m[n]$$

partial reconstruction: 
$$x_K[n] = \sum_{m=0}^{K-1} C_m^{-1} y[m] \phi_m[n] \quad K < N$$

reconstruction error of partial reconstruction

$$\begin{aligned} \|x[n] - x_K[n]\|^2 &= \sum_{n=0}^{N-1} \left\| \sum_{m=K}^{N-1} C_m^{-1} y[m] \phi_m[n] \right\|^2 \\ &= \sum_{n=0}^{N-1} \sum_{m=K}^{N-1} C_m^{-1} y[m] \phi_m[n] \sum_{m_1=K}^{N-1} C_{m_1}^{-1} y^*[m_1] \phi_{m_1}^*[n] \\ &= \sum_{m=K}^{N-1} \sum_{m_1=K}^{N-1} C_m^{-1} y[m] C_{m_1}^{-1} y^*[m_1] \sum_{n=0}^{N-1} \phi_m[n] \phi_{m_1}^*[n] \\ &= \sum_{m=K}^{N-1} \sum_{m_1=K}^{N-1} C_m^{-1} y[m] C_{m_1}^{-1} y^*[m_1] C_m \delta[m - m_1] = \sum_{m=K}^{N-1} C_m^{-1} |y[m]|^2 \end{aligned}$$

由於  $C_m^{-1} |y[m]|^2$  一定是正的，可以保證  $K$  越大, reconstruction error 越小

For non-orthogonal case,

perfect reconstruction:  $x[n] = \sum_{m=0}^{N-1} B[n, m] y[m] \quad \mathbf{B} = \mathbf{A}^{-1}$

partial reconstruction:  $x_K[n] = \sum_{m=0}^{K-1} B[n, m] y[m] \quad K < N$

reconstruction error of partial reconstruction

$$\begin{aligned} \|x[n] - x_K[n]\|^2 &= \sum_{n=0}^{N-1} \left\| \sum_{m=K}^{N-1} B[n, m] y[m] \right\|^2 \\ &= \sum_{n=0}^{N-1} \sum_{m=K}^{N-1} B[n, m] y[m] \sum_{m_1=K}^{N-1} B^*[n, m_1] y^*[m_1] \\ &= \sum_{m=K}^{N-1} \sum_{m_1=K}^{N-1} y[m] y^*[m_1] \sum_{n=0}^{N-1} B[n, m] B^*[n, m_1] \end{aligned}$$

由於  $y[m] y^*[m_1] \sum_{n=0}^{N-1} B[n, m] B^*[n, m_1]$  不一定是正的，  
無法保證  $K$  越大, reconstruction error 越小

## ◎ 14-B Frequency and Time Division Multiplexing

傳統 Digital Modulation and Multiplexing : 使用 Fourier transform

### • Frequency-Division Multiplexing (FDM)

$$z(t) = \sum_{n=0}^{N-1} X_n \exp(j2\pi f_n t) \quad X_n = 0 \text{ or } 1$$

$X_n$  can also be set to be  $-1$  or  $1$

When (1)  $t \in [0, T]$  (2)  $f_n = n/T$

$$z(t) = \sum_{n=0}^{N-1} X_n \exp\left(j \frac{2\pi n t}{T}\right)$$

it becomes the **orthogonal frequency-division multiplexing (OFDM)**.



Furthermore, if the time-axis is also sampled

$$t = mT/N, \quad m = 0, 1, 2, \dots, N-1$$

$t \in [0, T]$   
sampling for t-axis

$$z\left(m \frac{T}{N}\right) = \sum_{n=0}^{N-1} X_n \exp\left(j \frac{2\pi nm}{N}\right)$$

then the OFDM is equivalent to the transform matrix of the **inverse discrete Fourier transform (IDFT)**, which is one of the discrete orthogonal transform.

Modulation:  $Y_m = z\left(m \frac{T}{N}\right) = \sum_{n=0}^{N-1} A[m, n] X_n$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{j\frac{2\pi}{N}} & e^{j\frac{4\pi}{N}} & \dots & e^{j\frac{2(N-1)\pi}{N}} \\ 1 & e^{j\frac{4\pi}{N}} & e^{j\frac{8\pi}{N}} & \dots & e^{j\frac{4(N-1)\pi}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j\frac{2(N-1)\pi}{N}} & e^{j\frac{4(N-1)\pi}{N}} & \dots & e^{j\frac{2(N-1)(N-1)\pi}{N}} \end{bmatrix}$$

Modulation: 
$$Y_m = \sum_{n=0}^{N-1} A[m, n] X_n$$

Demodulation: 
$$X_n = \frac{1}{N} \sum_{m=0}^{N-1} A^*[m, n] Y_m$$

Example:  $N = 8$

$$X_n = [1, 0, 1, 1, 0, 0, 1, 1] \quad (n = 0 \sim 7)$$

- **Time-Division Multiplexing (TDM)**

$$z(0) = X_0, \quad z\left(\frac{T}{N}\right) = X_1, \quad z\left(2\frac{T}{N}\right) = X_2, \quad \dots, \quad z\left((N-1)\frac{T}{N}\right) = X_{N-1}$$

$$y(m) = z\left(m\frac{T}{N}\right) = \sum_{n=0}^{N-1} A[m,n]X_n$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

(also a discrete orthogonal transform)

思考：

既然 time-division multiplexing 那麼簡單

那為什麼要使用 frequency-division multiplexing  
和 orthogonal frequency-division multiplexing (OFDM)?

## ◎ 14-C Code Division Multiple Access (CDMA)

除了 **frequency**-division multiplexing 和 **time**-division multiplexing，是否還有其他 multiplexing 的方式？

使用其他的 orthogonal transforms  
即 code division multiple access (CDMA)

CDMA is an important topic in **spread spectrum** communication

參考資料

[1] M. A. Abu-Rgheff, *Introduction to CDMA Wireless Communications*, Academic, London, 2007

[2] 邱國書, 陳立民譯, “CDMA 展頻通訊原理”, 五南, 台北, 2002.

CDMA 最常使用的 orthogonal transform 為 Walsh transform

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

channel 1  
 channel 2  
 channel 3  
 channel 4  
 channel 5  
 channel 6  
 channel 7  
 channel 8

channel 1

當有兩組人在同一個房間裡交談 (A 和B交談) , (C 和D交談) ,  
如何才能夠彼此不互相干擾?

(1) Different Time

(2) Different Tone

(3) Different Language

CDMA 分為：

- (1) Orthogonal Type      (2) Pseudorandom Sequence Type

Orthogonal Type 的例子： 兩組資料  $[1, 0, 1]$      $[1, 1, 0]$

(1) 將 0 變為 -1       $[1, -1, 1]$      $[1, 1, -1]$

(2)  $1, -1, 1$  modulated by  $[1, 1, 1, 1, 1, 1, 1, 1]$  (channel 1)

→  $[1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1]$

$1, 1, -1$  modulated by  $[1, 1, 1, 1, -1, -1, -1, -1]$  (channel 2)

→  $[1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1]$

(3) 相合

$[2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, -2, -2, -2, -2, 0, 0, 0, 0, 2, 2, 2, 2]$



demodulation

[2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, -2, -2, -2, -2, 0, 0, 0, 0, 2, 2, 2, 2]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

↓  
内積 = 8

注意：

- (1) 使用  $N$ -point Walsh transform 時，總共可以有  $N$  個 channels
- (2) 除了 Walsh transform 以外，其他的 orthogonal transform 也可以使用
- (3) 使用 Walsh transform 的好處

- Orthogonal Transform 共通的問題: 需要同步 synchronization

$$\mathbf{R}_1 = [1, 1, 1, 1, 1, 1, 1, 1]$$

$$\mathbf{R}_2 = [1, 1, 1, 1, -1, -1, -1, -1]$$

$$\mathbf{R}_5 = [1, -1, -1, 1, 1, -1, -1, 1]$$

$$\mathbf{R}_8 = [1, -1, 1, -1, 1, -1, 1, -1]$$

但是某些 basis, 就算不同步也近似 orthogonal

$$\langle \mathbf{R}_1[n], \mathbf{R}_1[n] \rangle = 8, \quad \langle \mathbf{R}_1[n], \mathbf{R}_k[n] \rangle = 0 \text{ if } k \neq 1$$

$$\langle \mathbf{R}_1[n], \mathbf{R}_k[n-1] \rangle = 2 \text{ or } 0 \quad \text{if } k \neq 1.$$

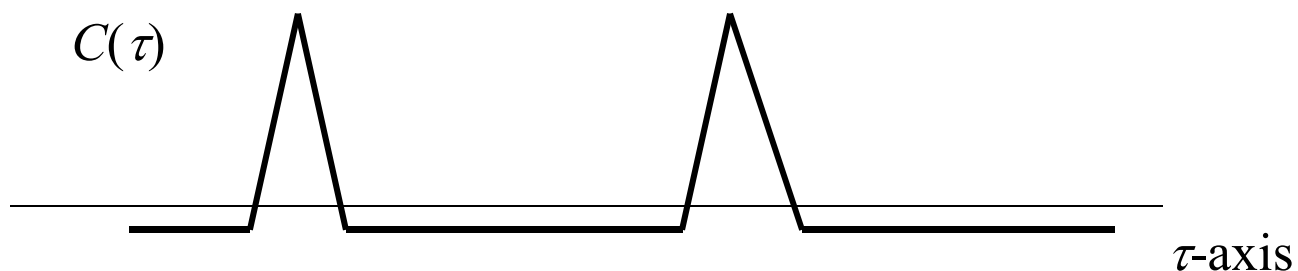
這裡的 shift 為 circular shift

## Pseudorandom Sequence Type

不為 orthogonal，capacity 較少

但是不需要同步 (asynchronous)

## Pseudorandom Sequence 之間的 correlation



$$b_1 p(t + \tau_1) + b_2 p(t + \tau_2)$$

$$\text{recovered: } \int (b_1 p(t + \tau_1) + b_2 p(t + \tau_2)) p(t + \tau_1) dt = b_1 C(0) + b_2 C(\tau_2 - \tau_1) \approx b_1$$

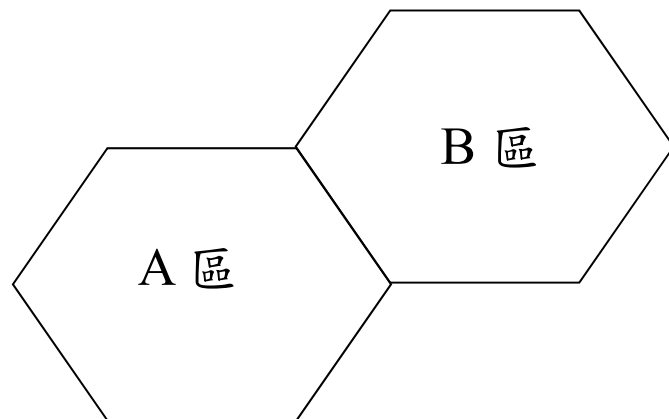
$$(\text{若 } C(0) = 1, C(\tau_2 - \tau_1) \approx 0)$$

$\tau_1, \tau_2$  不必一致

CDMA 的優點：

- (1) 運算量相對於 frequency division multiplexing 減少很多
- (2) 可以減少 noise 及 interference 的影響
- (3) 可以應用在保密和安全傳輸上
- (4) 就算只接收部分的信號，也有可能把原來的信號 recover 回來
- (5) 相鄰的區域的干擾問題可以減少

相鄰的區域，使用差距最大的「語言」，則干擾最少



假設 A 區使用的 orthogonal basis 為  $\phi_k[n], k = 0, 1, 2, \dots, N-1$

B 區使用的 orthogonal basis 為  $\mu_h[n], h = 0, 1, 2, \dots, N-1$

設法使  $\max \left( \left| \frac{\langle \phi_k[n], \mu_h[n] \rangle}{\langle \phi_k[n], \phi_h[n] \rangle} \right| \right)$  為最小

$k = 0, 1, 2, \dots, N-1, h = 0, 1, 2, \dots, N-1$

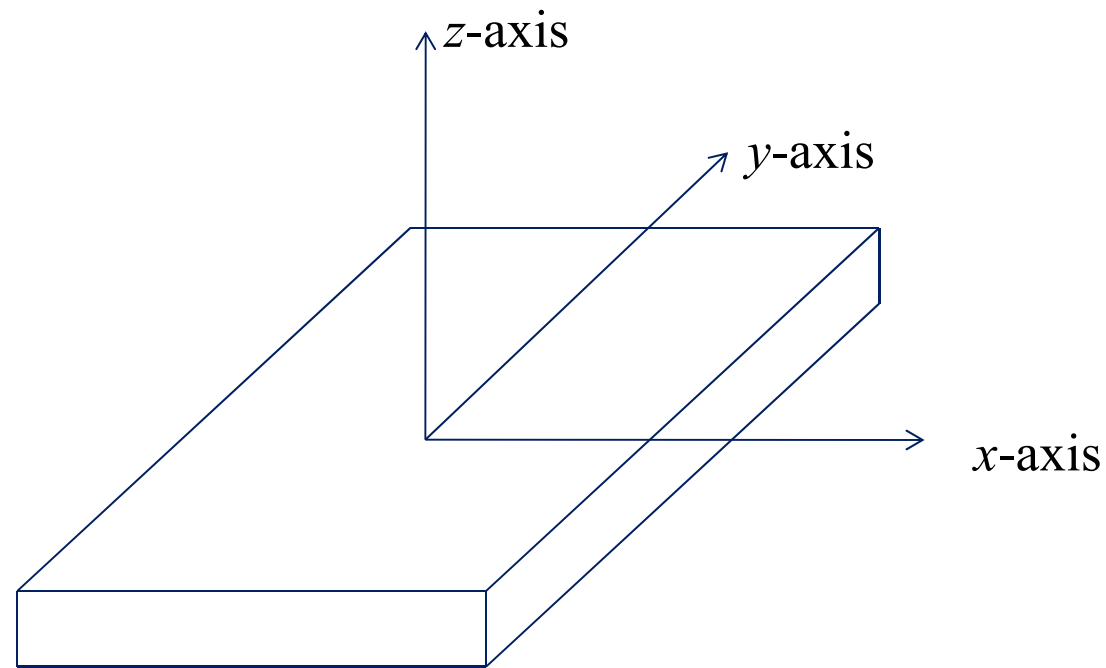
## 附錄十四 3-D Accelerometer 的簡介

**3-D Accelerometer:** 三軸加速器，或稱作加速規

許多儀器(甚至包括智慧型手機)都有配置三軸加速器

可以用來判別一個人的姿勢和動作

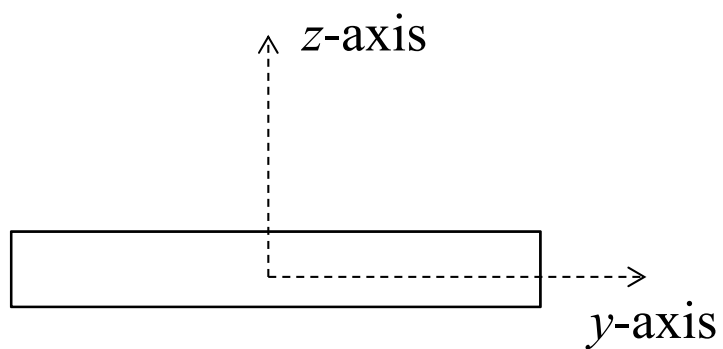
註：**Gyrator** (陀螺儀)可以用來量測物體旋轉之方向，可補 3-D Accelerometer 之不足，許多儀器(包括智慧型手機)也內建陀螺儀之裝置，3-D Accelerometer Signal Processing 和 gyration signal processing 經常並用



根據  $x, y, z$  三個軸的加速度的變化，來判斷姿勢和動作

平放且靜止時， $z$ -axis 的加速度為  $-g = -9.8$

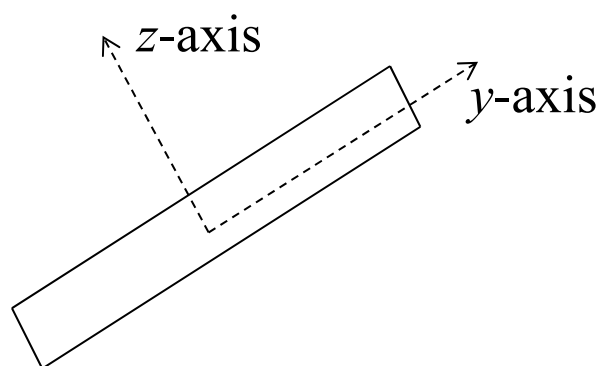




$$y: 0$$

$$z: -9.8$$

tilted by  $\theta$



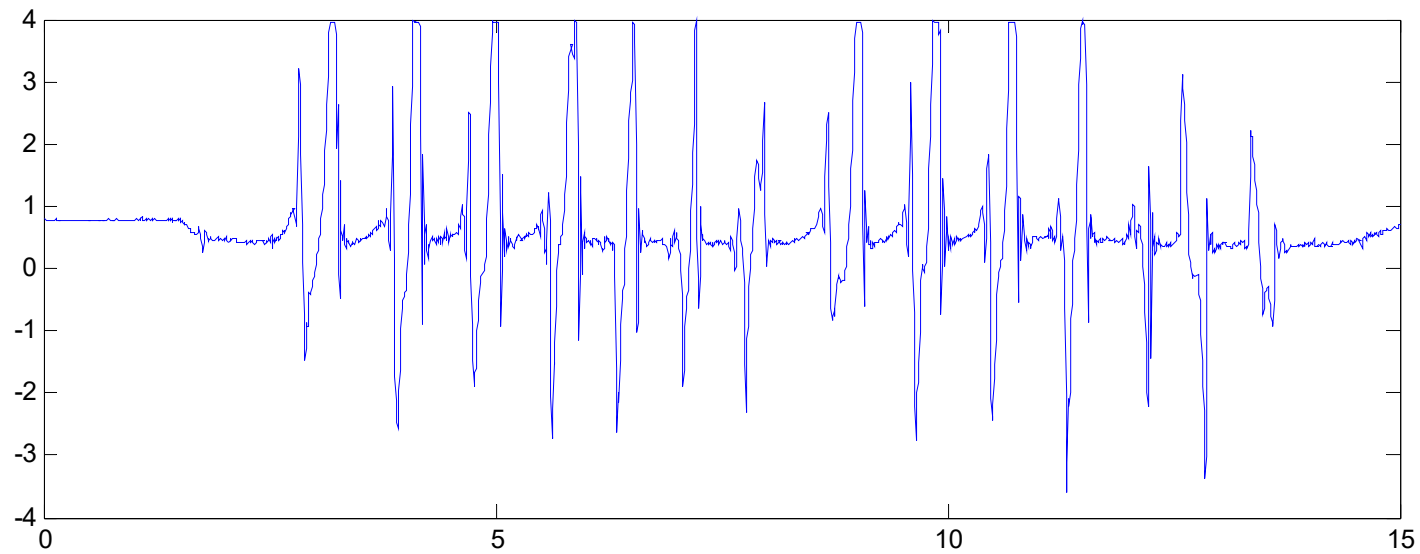
$$y: -9.8\sin \theta$$

$$z: -9.8\cos \theta$$

可藉由加速規傾斜的角度，來判斷姿勢和動作

例子：若將加速規放在腳上.....

走路時，沿著其中一個軸的加速度變化



應用： 動作辨別

運動 (訓練，計步器)

醫療復健，如 Parkinson 患者照顧，傷患復原情形

其他 (如動物的動作，機器的運轉情形的偵測)

3-D Accelerometer Signal Processing 是訊號處理的重要課題之一

一方面固然是因為應用多，另一方面， 3-D Accelerometer Signal 容易受 noise 之干擾，要如何藉由 3-D Accelerometer Signal 來還原動作以及移動速度，仍是個挑戰

## 祝各位同學暑假愉快！

各位同學在研究上或工作上，有任何和 digital signal processing 或 time frequency analysis 方面的問題，歡迎找我來一起討論。