XIII. Walsh Transform (Hadamard Transform)

◎ 13-A Ideas of Walsh Transforms

- 8-point Walsh transform

$$W[m,n] = \begin{array}{cc} & m \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} & \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{array}$$

- Advantages of the Walsh transform:

(1) Real

(2) No multiplication is required

(3) Some properties are similar to those of the DFT

- Forward and inverse Walsh transforms are similar.

forward: $F[m] = \sum_{n=0}^{N-1} f[n] W[m,n]$, inverse: $f[m] = \frac{1}{N} \sum_{n=0}^{N-1} W[m,n] F[n]$

- Alternative names of the Walsh transform:

Hadamard transform, Walsh-Hadamard transform

- Orthogonal Property $\sum_{n=0}^{N-1} W[m_0,n] W[m_1,n] = 0$ if $m_0 \neq m_1$
- Zero-Crossing Property
- Even / Odd Property
- Fast Algorithm

Useful for spectrum analysis

Sometimes also useful for implementing the convolution

Walsh transform 和 DFT, DCT 有許多相似處

$$W[m,n]=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&1&-1&1&-1&1&-1\end{bmatrix},\ DFT[m,n]=\exp(-j2\pi m\,n/N),$$

$$\mathbf{DCT}=\begin{bmatrix}1.0000&1.0000&1.0000&1.0000&1.0000&1.0000&1.0000&1.0000\\1.3870&1.1759&0.7857&0.2759&-0.2759&-0.7857&-1.1759&-1.3870\\1.3066&0.5412&-0.5412&-1.3066&-1.3066&-0.5412&0.5412&1.3066\\1.1759&-0.2759&-1.3870&-0.7857&0.7857&1.3870&0.2759&-1.1759\\1.0000&-1.0000&-1.0000&1.0000&1.0000&-1.0000&-1.0000&1.0000\\0.7857&-1.3870&0.2759&1.1759&-1.1759&-0.2759&1.3870&-0.7857\\0.5412&-1.3066&1.3066&-0.5412&-0.5412&1.3066&-1.3066&0.5412\\0.2759&-0.7857&1.1759&-1.3870&1.3870&-1.1759&0.7857&-0.2759\end{bmatrix}$$

References for Walsh Transforms

[1] K. G. Beanchamp, *Walsh Functions and Their Applications*, Academic Press, New York, 1975.

[2] B. I. Golubov, A. Efimov, and V. Skvortsov, *Walsh Series and Transforms: Theory and Applications*, Kluwer Academic Publishers, Boston, 1991.

[3] H. F. Harmuth, "Applications of Walsh functions in communications," *IEEE Spectrum*, vol. 6, no. 11, pp. 82-91, Nov. 1969.

[4] H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, New York, 1972.

◎ 13-B Generate the Walsh Transform

2-point Walsh transform

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

4-point Walsh transform

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

How do we obtain the 2^{k+1}-point Walsh transform from the 2^k-point Walsh transform ?

Step 1 $$\mathbf{V}_{2^{k+1}} = \begin{bmatrix} \mathbf{W}_{2^k} & \mathbf{W}_{2^k} \\ \mathbf{W}_{2^k} & -\mathbf{W}_{2^k} \end{bmatrix}$$

Step 2 根據 sign changes 將 rows 的順序重新排列

$$\mathbf{V}_{2^{k+1}} \xrightarrow{\;permutation\;} \mathbf{W}_{2^{k+1}}$$

已知 $\mathbf{W}_{2^k}$ 每個 row 的 sign change 數，由上到下分別為

$0, 1, 2, 3, \ldots.., 2^k-1$

則 $\mathbf{V}_{2^{k+1}}$ 每個 row 的 sign change 數，由上到下分別為

$0, 3, 4, 7, \ldots.., 2^{k+1}-1, 1, 2, 5, 6, \ldots.., 2^{k+1}-2,$

若 row 的index 由0 開始

則 $\mathbf{V}_{2^{k+1}}$ 第 n 個 row (n is even and $n < N/2$) 的 sign change 為 $2n$

(n is odd and $n < N/2$) 的 sign change 為 $2n + 1$

(n is even and $n \geq N/2$) 的 sign change 為 $2n+1-N$

(n is odd and $n \geq N/2$) 的 sign change 為 $2n-N$

要根據 sign change 的數目將 $\mathbf{V}_{2^{k+1}}$ 的 row 重新排列

$$\mathbf{V}_{2^{k+1}} \xrightarrow{\textit{permutation}} \mathbf{W}_{2^{k+1}}$$

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \mathbf{V}_4 = \begin{bmatrix} \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{array}{c} \text{sign changes} \\ 0 \\ 3 \\ 1 \\ 2 \end{array}$$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad \mathbf{V}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{array}{c} \text{sign changes} \\ 0 \\ 3 \\ 4 \\ 7 \\ 1 \\ 2 \\ 5 \\ 6 \end{array}$$

Constraint for the number of points of the Walsh transform:

N must be a power of 2 (2, 4, 8, 16, 32, ……..)

Although in Matlab it is possible to define the $12 \cdot 2^k$ point or the $20 \cdot 2^k$ point Walsh transform, the inverse transform require the floating-point operation.

◉ 13-C Alternative Forms of the Walsh Transform

標準定義 from zero-crossing

- Sequency ordering (i.e., Walsh ordering) ……. using for signal processing
- Dyadic ordering (i.e., Paley ordering) ……….. using for control
- Natural ordering (i.e., Hadamard ordering) ……using for mathematics

Sequency ordering	Dyadic ordering	Natural ordering	$W[m, n]$
←→	(Gray Code) ←→	(Bit Reversal)	
row 0	row 0	row 0	[1, 1, 1, 1, 1, 1, 1, 1]
row 1	row 1	row 4	[1, 1, 1, 1, −1, −1, −1, −1]
row 2	row 3	row 6	[1, 1, −1, −1, −1, −1, 1, 1]
row 3	row 2	row 2	[1, 1, −1, −1, 1, 1, −1, −1]
row 4	row 6	row 3	[1, −1, −1, 1, 1, −1, −1, 1]
row 5	row 7	row 7	[1, −1, −1, 1, −1, 1, 1, −1]
row 6	row 5	row 5	[1, −1, 1, −1, −1, 1, −1, 1]
row 7	row 4	row 1	[1, −1, 1, −1, 1, −1, 1, −1]

- Dyadic ordering Walsh transform

$$W[m,n]=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

- Natural ordering Walsh transform

$$W[m,n]=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

- binary code $n = \sum_{p=1}^{k} b_p 2^{p-1}$ to gray code

  When $N = 2^k$

$g_k = b_k$, $g_q = \mathrm{XOR}(b_{q+1}, b_q)$ for $q = k-1, k-2, \ldots., 1$ $\qquad m = \sum_{q=1}^{k} g_q 2^{q-1}$

- gray code to binary code

  When $N = 2^k$

$b_k = g_k$, $b_q = \mathrm{XOR}(b_{q+1}, g_q)$ for $q = k-1, k-2, \ldots., 1$

⦿ 13-D Properties

(1) Orthogonal Property

(2) Zero-Crossing Property

(3) Even / Odd Property

(4) Linear Property

If $f[n] \Rightarrow F[m]$, $g[n] \Rightarrow G[m]$, ($\Rightarrow$ means the Walsh transform)

then $a\,f[n] + b\,g[n] \Rightarrow a\,F[m] + b\,G[m]$

(5) Addition Property

$$W[m,n]\cdot W[l,n]=W[m\oplus l,n]$$

註： Addition modulo 2 (denoted by ⊕)

$0 \oplus 0 = 1 \oplus 1 = 0, \quad 0 \oplus 1 = 1 \oplus 0 = 1,$

$$(\sum_{p=0}^{k} a_k 2^p)\oplus(\sum_{p=0}^{k} b_k 2^p)=\sum_{p=0}^{k}(a_k \oplus b_k)2^p$$

Example:

3	0	1	1
7	1	1	1
4	1	0	0

, therefore $3 \oplus 7 = 4$

(6) Special functions

$\delta[n] = 1$ when $n = 0$, $\delta[n] = 0$ when $n \neq 0$

$\delta[n] \Rightarrow 1$, $1 \Rightarrow N \cdot \delta[n]$

(7) Shifting property

If $f[n] \Rightarrow F[m]$, then $f[n \oplus k] \Rightarrow W(k, m) \cdot F[m]$

(8) Modulation property

If $f[n] \Rightarrow F[m]$, then $W(k, n) \cdot f[n] \Rightarrow F[m \oplus k]$

(9) Parseval's Theorem

If $f[n] \Rightarrow F[m]$,

$$\sum_{n=0}^{N-1} \left| f[n] \right|^2 = \frac{1}{N} \sum_{n=0}^{N-1} \left| F[m] \right|^2 ,$$

If $f[n] \Rightarrow F[m]$, $g[n] \Rightarrow G[m]$,

$$\sum_{n=0}^{N-1} f[n] g[n] = \frac{1}{N} \sum_{n=0}^{N-1} F[m] G[m]$$

(10) Convolution Property

If $f[n] \Rightarrow F[m]$, $\quad g[n] \Rightarrow G[m]$,

then $h[n] = f[n] \star g[n] \Rightarrow F[m]\, G[m]$

$\star$ means the "logical convolution"

$$h[n] = f[n] \star g[n] = \sum_{l=0}^{N-1} f\left[l\right] g\left[((n \oplus l))_N\right] = \sum_{l=0}^{N-1} f\left[((n \oplus l))_N\right] g\left[l\right]$$

For example, when $N = 8$,

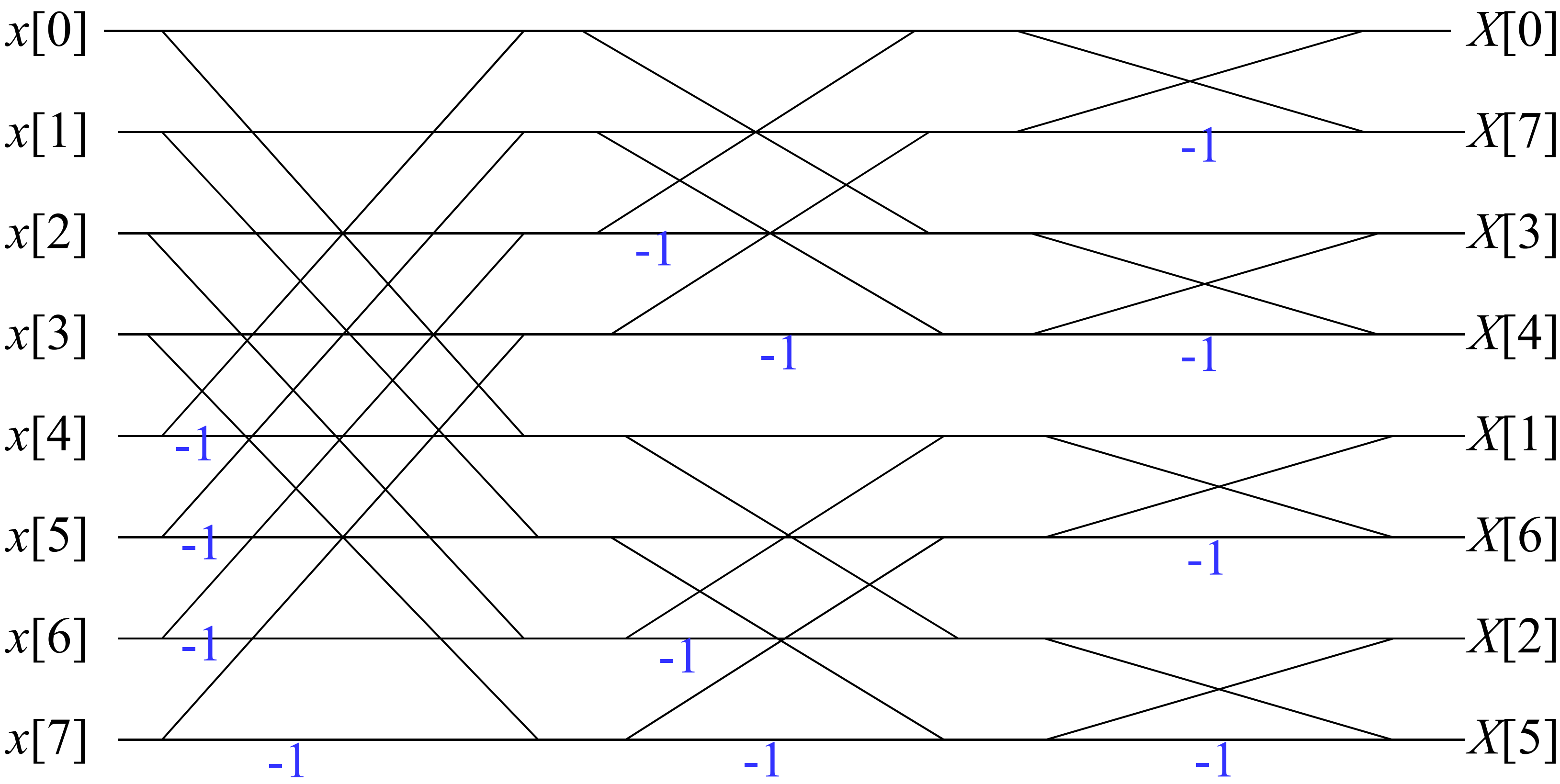
$$h[3] = f[0]g[3] + f[1]g[2] + f[2]g[1] + f[3]g[0] + f[4]g[7] + f[5]g[6] + f[6]g[5] + f[7]g[4]$$

$$h[2] = f[0]g[2] + f[1]g[3] + f[2]g[0] + f[3]g[1] + f[4]g[6] + f[5]g[7] + f[6]g[4] + f[7]g[5]$$

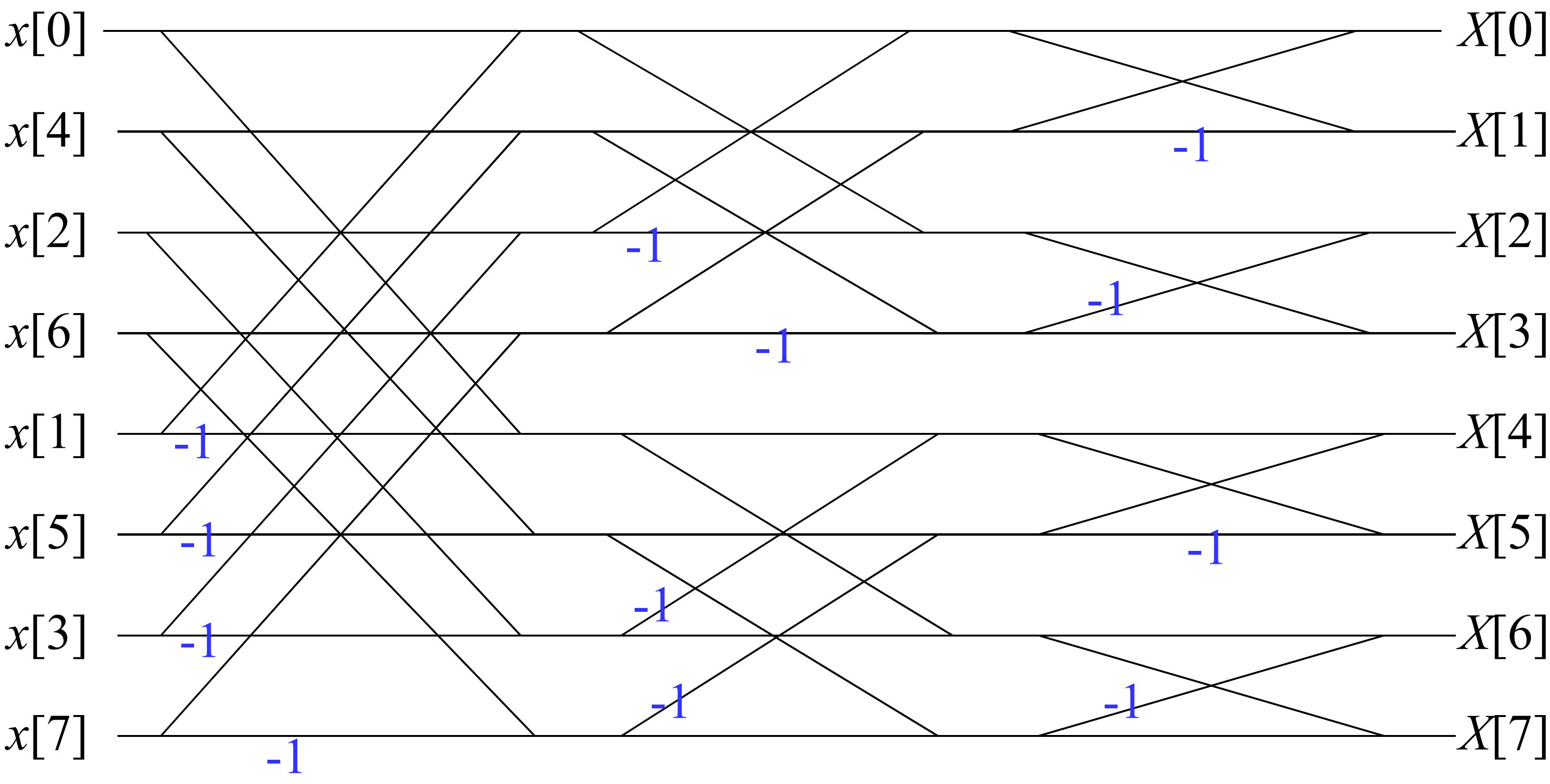
和 linear convolution 比較

⦿ 13-E Butterfly Fast Algorithm

(Method 1) John L. Shark's Algorithm



(Method 2) Manz's Sequence Algorithm



There are other fast implementation algorithm for the Walsh transform.

◉ 13-F Applications

Walsh transform 適合作 spectrum analysis，但未必適合作convolution

↓

may not be better than DFT, DCT

Applications of the Walsh transform

Bandwidth reduction

High resolution

Multiplexing

Information coding

Feature extraction

ECG signal (in medical signal processing) analysis

Hadamard spectrometer

Avoiding quantization error

- The Walsh transform is suitable for the function that is a combination of Step functions

New Applications: CDMA (code division multiple access)

◉ 13-G Jacket Transform

把部分的 1 用 $\pm 2^k$ 取代

4-point Jacket transform

$$\mathbf{J}_4 = \begin{bmatrix} 1 & x & x & 1 \\ 1 & w & -w & -1 \\ 1 & -x & -x & 1 \\ 1 & -w & w & 1 \end{bmatrix} \qquad w = 2^k, \quad x = 2^h,$$

2^{k+1}-point Jacket

$$\mathbf{J}_{2^{k+1}} = \mathbf{P}\begin{bmatrix} \mathbf{J}_{2^k} & \mathbf{J}_{2^k} \\ \mathbf{J}_{2^k} & -\mathbf{J}_{2^k} \end{bmatrix}$$

P: row permutation

[Ref] M.H. Lee, "A new reverse Jacket transform and its fast algorithm", *IEEE Trans. Circuits Syst.-II* , vol 47, pp.39-46, 2000.

◉ 13-H Haar Transform

y = Hx

$N=2$ $\quad \mathbf{H}=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$

$N=4$ $\quad \mathbf{H}=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & -1\end{bmatrix}$

$N=8$ $\quad \mathbf{H}=\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1\\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix}$

[Ref] H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, New York, 1972

$N = 16$

$$\mathbf{H} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}$$

$H[m, n]$ 的值 ($m = 0, 1, \ldots, 2^k-1$, $n = 0, 1, \ldots, 2^k-1$)：

$H[0, n] = 1$ for all n

If $2^h \le m < 2^{h+1}$

$H[m, n] = 1$ for $(m - 2^h)2^{k-h} \le n < (m - 2^h + 1/2)2^{k-h}$

$H[m, n] = -1$ for $(m - 2^h + 1/2)2^{k-h} \le n < (m - 2^h + 1)2\mathrm{k}^{-h}$

$H[m, n] = 0$ otherwise

運算量比 Walsh transforms 更少

Applications: localized spectrum analysis, edge detection

Transforms	Running Time	terms required for NRMSE $< 10^{-5}$
DFT	9.5 sec	43
Walsh Transform	2.2 sec	65
Haar Transform	0.3 sec	128

Main Advantage of the Haar Transform

(1) Fast (but this advantage is no longer important)

(2) Analysis of the local high frequency component
(The wavelet transform is a generalization of the Haar transform)

(3) Extracting local features
(Example: Adaboost face detection)

附錄十三 SCI Papers 查詢方式

我們經常聽到 SCI 論文，impact factor….那麼什麼是 SCI 和 impact factor？什麼樣的論文是 SCI Papers? Impact factor 號如何查詢？

SCI 全名： Science Citation Index

(A) SCI 相關網站：ISI Web of Knowledge

連結至 ISI Web of Knowledge

http://admin-apps.webofknowledge.com/JCR/JCR?RQ=HOME

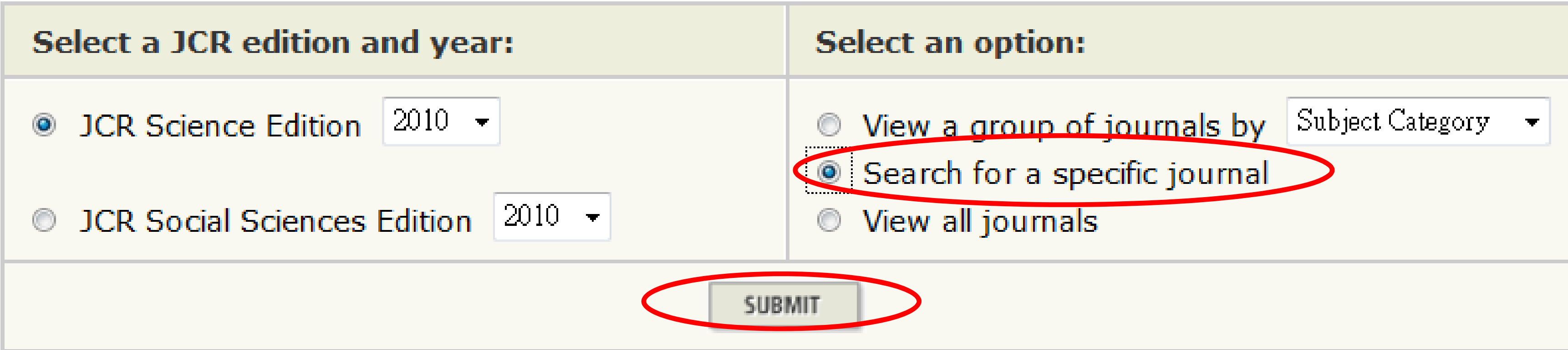
若出現「You do not have a session」

則按 establish a new session

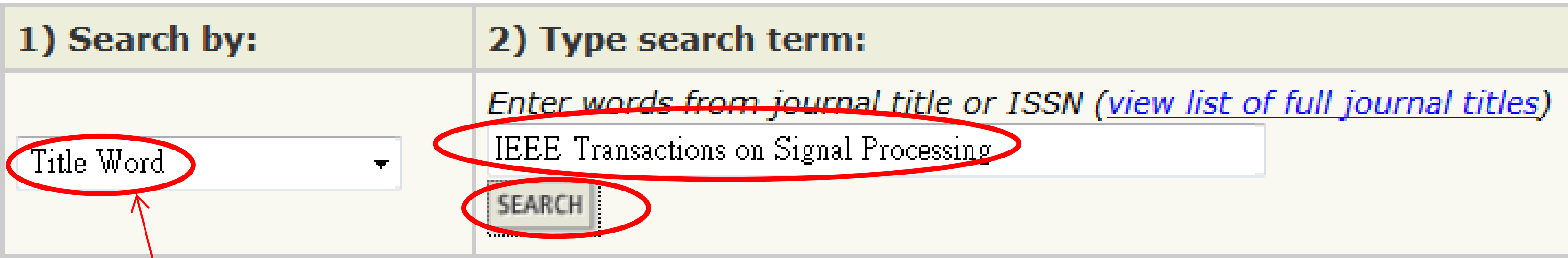
註：必需要在台大上網，或是在其他有付錢給 ISI 的學術單位上網，才可以使用 ISI Web of Knowledge

(B) 若要找某個期刊是否為 SCI journal，以及它的 impact factor

先點選 Search for a specific journal，再按 SUBMIT



再輸入期刊的名稱，再按 SEARCH



建議用 Title Word，這樣只需知道部分期刊名稱即可查詢

若有搜尋到，則代表這個期刊是 SCI 期刊

並且會顯示出這個期刊的 impact factor

Mark	Rank	Abbreviated Journal Title *(linked to journal information)*	ISSN	JCR Data						Eigenfactor™ Metrics	
				Total Cites	Impact Factor	5-Year Impact Factor	Immediacy Index	Articles	Cited Half-life	*Eigenfactor*™ Score	*Article Influence*™ Score
☐	1	IEEE T SIGNAL PROCES	1053-587X	16725	2.651	2.968	0.340	571	7.2	0.04989	1.111

Impact Factor (影響係數)

(C) 關於 impact factor (影響係數)：

若一個 journal 裡面的文章，被別人引用的次數越多，則這個 journal 的 impact factor 越高

一般而言， impact factor 在 1.5 以上的 journals，已經算是高水準的期刊

Nature 的 impact factor 為 36.1
Science 的 impact factor 為 31.4

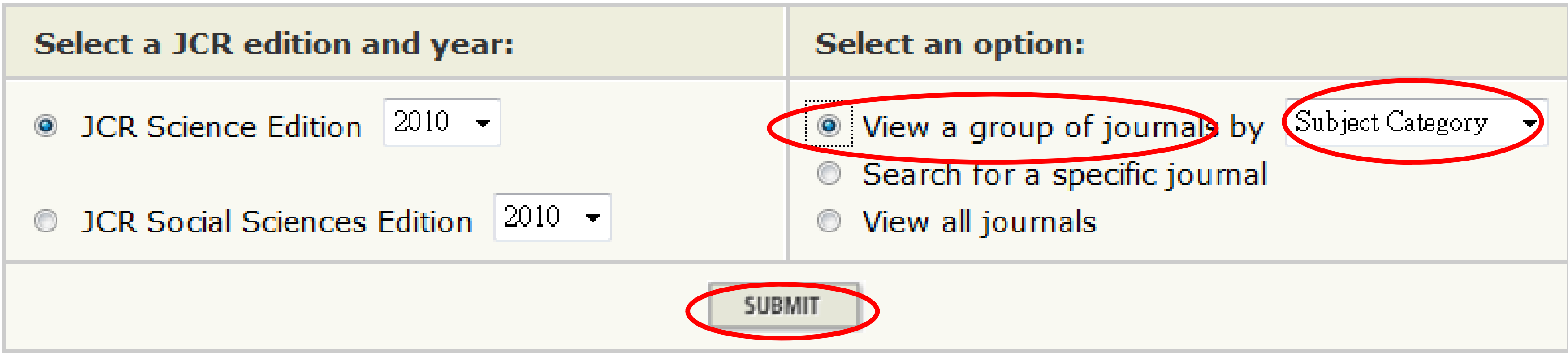
IEEE 系列的期刊的 impact factors 通常在 1 到 5 之間
IEEE Trans. Image Processing的 impact factors 在 3.0 左右
IEEE Trans. Signal Processing的 impact factors 在 3.0 左右

中等水準的期刊的 impact factors 在 0.6 到 1.5 之間

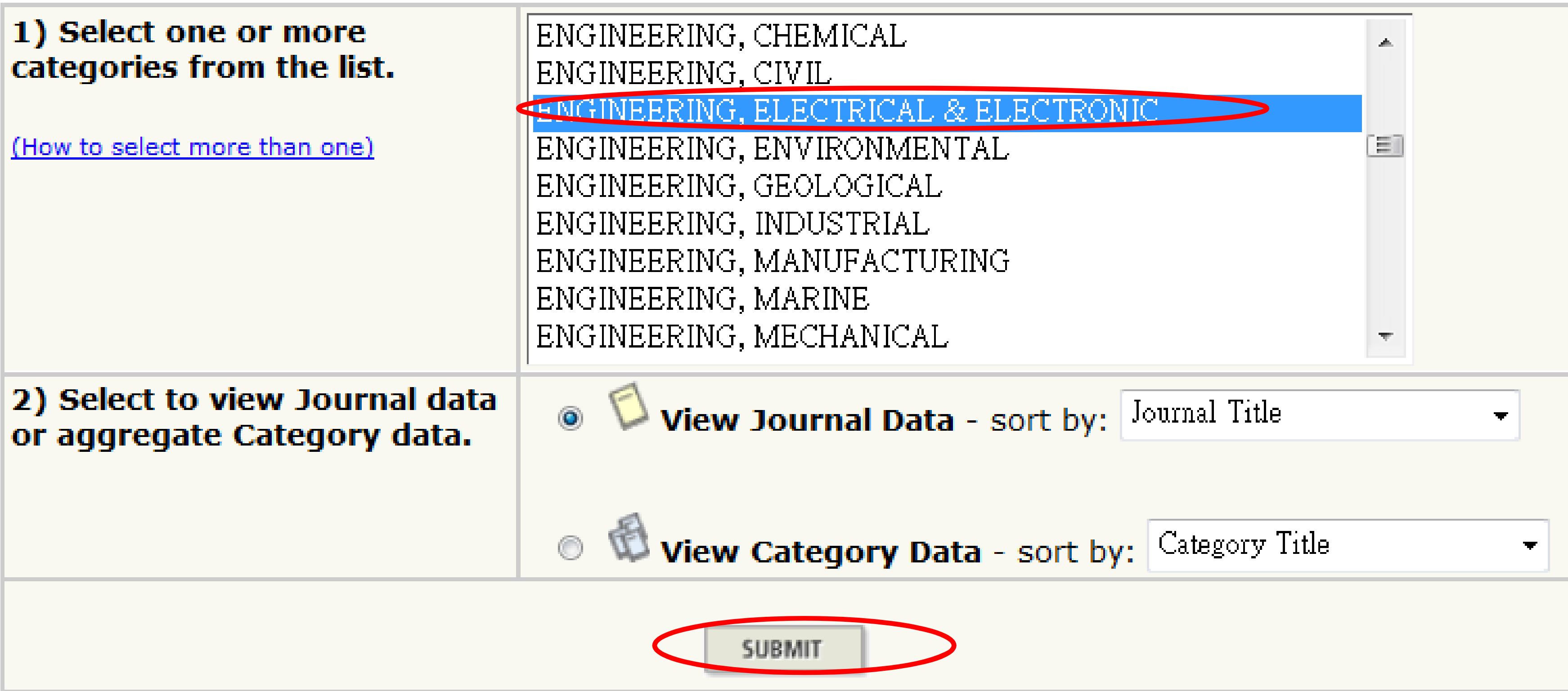
(D) 要查詢一個領域有哪些 SCI journals

方法一

連結至 ISI Web of Knowledge 之後，點選 View a group of journals by 「Subject Category」，再按 SUBMIT



接著，再點選要查詢的 category，再按 SUBMIT，即可顯示出這一類的 SCI journals

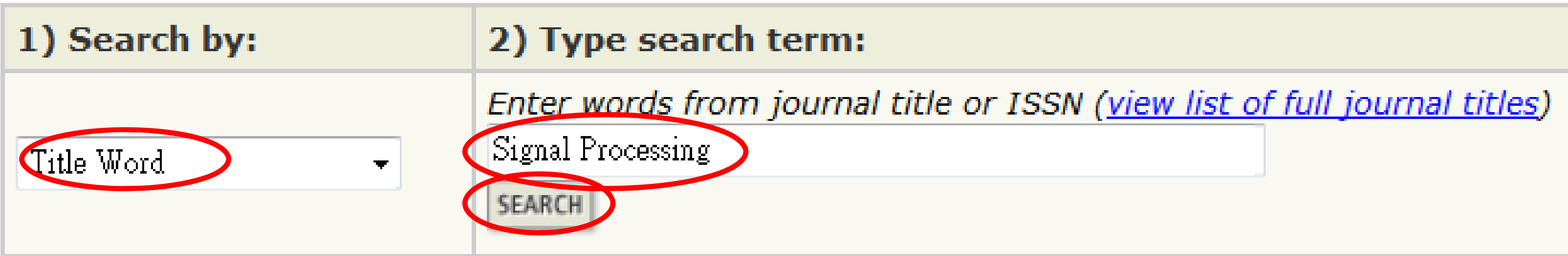


要查詢一個領域有哪些 SCI journals

方法二

連結至 ISI Web of Knowledge 之後，點選 Search for a specific journal，SUBMIT 之後

左邊選用 Title Word，右邊輸入關鍵字



再按 Search 之後，即可找到所有期刊名稱當中有包含這個關鍵字的 journals

(E) EI (Engineering Village)

官方網站： www.engineeringvillage.org

http://www.engineeringvillage.com/search/quick.url

查詢期刊或研討會是否為 EI

http://tul.blog.ntu.edu.tw/archives/4627

(F) SSCI (Social Science Citation Index)

比較偏向於社會科學

http://www.thomsonscientific.com/cgi-bin/jrnlst/jloptions.cgi?PC=J

(G) Conference 排名

Microsoft Academic Search 有列出各領域知名的 conferences 並加以排名 (大致上也是被引用越多的排名越前面)

和通訊與信號處理相關的 conferences，大多排名於

http://academic.research.microsoft.com/RankList?entitytype=3&topDomainID=2&subDomainID=0&last=0&start=1&end=100

或

http://academic.research.microsoft.com/RankList?entitytype=3&topDomainID=8&subDomainID=0&last=0&start=1&end=100

寫論文和投稿的經驗談

研究生經常會寫論文並且投稿。要如何讓論文投稿之後能夠順利的被接受，相信是同學們所期盼的，畢竟每篇論文都是大家花了不少時間的心血結晶，若論文能夠順利的被接受，也代表了自己的成果總算獲得了肯定。然而，影響論文是否被接受的因素很多，一個好的研究成果，還是配合好的編寫技巧，才可以被一流的期刊或研討會所接受。以下是個人關於論文編寫與投稿的經驗談：

(1) 你的論文的「賣點」(優點) 是什麼？人家有沒有辦法一眼看得出來你論文的「賣點」？

寫論文其實就是在推銷商品，而所謂的「商品」，就是你的「研究成果」。要說服人家接受你的商品，首先就是要強調你的商品的「賣點」。

(2) 和既有的方法的比較是否足夠？

要證明你所提出的方法是有效的，最好的方式，就是和既有的方法相比較，而且比較的對象越多越好，越新越好。

(3) 和前人的方法相比，你的方法創新的地方在何處？審稿者是否能看得出來你論文創新的地方？

(4) 就算你的文章和理論相關，最好也多提出實際應用的例子

(5) 參考資料越多越好，越新越好

(6) Previous work (前人已經提出的概念) 精簡介紹即可，多強調自己的貢獻。Introduction 加上 Previous work 最好不要超過一篇論文的四分之一

(7) 英文表達能力要有一定的水準

(8)可以多用數學式和圖來解釋概念，有時會比文字還清楚

通常東方人英文表達能力有限。審稿者經常會看你們的圖表和數學式(而非文字) 來判斷你們論文的品質

(9) 同樣的道理，可以用「條列式」的方式來取代一大段文字來描述方法的觀念、流程、或優點

(10) 可以用 Conference 的期限來要求自己多寫研討會論文，之後再一個一個改成期刊論文投稿，如此一年的論文量將很可觀

(11)多注意格式，不同的期刊或研討會，對格式的要求也不同

(12) 最後，問自己一個問題：

如果你是審稿者，你會滿意你寫的這一篇論文嗎？

若答案是肯定的再投稿