ELEG 467/667 - Imaging and Audio Signal Processing

Gonzalo R. Arce

Chapter IV(d)

Department of Electrical and Computer Engineering
University of Delaware
Newark, DE, 19716
Spring 2013

Let **x** and **y** be two vectors:

$$\mathbf{x} = [x_1, \cdots, x_n]^T, \quad \mathbf{y} = [y_1, \cdots, y_n]^T$$

Their *inner product* is defined as

$$(\mathbf{x}, \mathbf{y}) \stackrel{\triangle}{=} \mathbf{x}^{*T}\mathbf{y} = \sum_{k=1}^{n} x_k^* y_k$$

where T and $*$ represent transpose and complex conjugate, respectively. The *norm* (magnitude, length) of a vector x is defined as

$$\|\mathbf{x}\| \stackrel{\triangle}{=} (\mathbf{x}, \mathbf{x})^{1/2} = \sqrt{\sum_{k=1}^{n} |x_k|^2}$$

where $|x|$ represents the absolute value if (real x) or norm (complex x) of x. **x** is normalized if $\|\mathbf{x}\| = 1$.

Two vectors **x** and **y** are *orthogonal* to each other if and only if their inner product is zero. For normalized orthogonal vectors, we have

$$(\mathbf{x}, \mathbf{y}) = \delta_{xy} \triangleq \begin{cases} 1 & \textit{if } \mathbf{x} = \mathbf{x} \\ 0 & \textit{if } \mathbf{x} \neq \mathbf{y} \end{cases}$$

Rank, Trace, Determinant, Transpose and Inverse of a Matrix

Let $\mathbf{A}$ be an $N \times N$ square matrix:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdot & a_{1N} \\ a_{21} & a_{22} & \cdot & a_{2N} \\ \cdot & \cdot & \cdot & \cdot \\ a_{N1} & a_{N2} & \cdot & a_{NN} \end{bmatrix}_{N \times N}$$

where

$$\begin{bmatrix} a_{1j} \\ a_{2j} \\ \cdots \\ a_{Nj} \end{bmatrix}$$

is the jth column vector and

$$[a_{i1} \ a_{i2} \ \cdots \ a_{iN}]$$

is the ith row vector.

The N rows span the *row space* of **A** and the N columns span the *column space* of **A**. The dimensions of these two spaces are the same and called the *rank* of **A**:

$$R = rank(\mathbf{A}) \leq N$$

The *determinant* of A is denoted by $det(\mathbf{A}) = |\mathbf{A}|$ and we have

$$|\mathbf{AB}| = |\mathbf{A}|\,|\mathbf{B}|$$

$rank(\mathbf{A}) < N$ if and only if $det(\mathbf{A}) = 0$.
The *trace* of A is defined as the sum of its diagonal elements:

$$tr(\mathbf{A}) = \sum_{i=1}^{N} a_{ii}$$

The *transpose* of a matrix **A**, denoted by $\mathbf{A}^T$, and

For any two matrices **A** and **B**, we have

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$

If $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$, where **I** is an identity matrix, then $\mathbf{B} = \mathbf{A}^{-1}$ is the *inverse* of **A**.
$\mathbf{A}^{-1}$ exists iff $det(\mathbf{A}) \neq 0$, i.e., $rank(\mathbf{A}) = N$.
For any two matrices **A** and **B**,

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

and

$$(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1}$$

Hermitian Matrix and Unitary Matrix

$\mathbf{A}$ is a *Hermitian matrix* iff $\mathbf{A}^{*T} = \mathbf{A}$.
When a Hermitian matrix $\mathbf{A}$ is real ($\mathbf{A}^* = \mathbf{A}$), it becomes *symmetric*, $\mathbf{A}^T = \mathbf{A}$.
$\mathbf{A}$ is a *unitary matrix* iff $\mathbf{A}^{*T}\mathbf{A} = \mathbf{I}$, i.e., $\mathbf{A}^{*T} = \mathbf{A}^{-1}$.
When a unitary matrix $\mathbf{A}$ is real ($\mathbf{A}^* = \mathbf{A}$), it becomes an *orthogonal matrix*, $\mathbf{A}^T = \mathbf{A}^{-1}$.

The columns (or rows) of a unitary matrix **A** are *orthonormal*, i.e. they are both orthogonal and normalized, i.e.,

$$(\mathbf{a}_i, \mathbf{a}_j) = \sum_k a_{ik}^* a_{jk} = \delta_{ij} \triangleq \begin{cases} 1 & \textit{if } i = j \\ 0 & \textit{if } i \neq j \end{cases}$$

Any Hermitian matrix **A** (symmetric if real) can be converted to a diagonal matrix Λ by a particular unitary (orthogonal if real) matrix Φ:

$$\Phi^{*T}\mathbf{A}\Phi = \Lambda$$

where Λ is a diagonal matrix.

Unitary Transforms

For the unitary matrix $\mathbf{A}$ ($\mathbf{A}^{-1} = \mathbf{A}^{*T}$), define a *unitary transform* $\mathbf{x} = [x_1, \cdots, x_n]^T$:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix} = \mathbf{A}^{*T}\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^{*T} \\ \mathbf{a}_2^{*T} \\ \cdots \\ \mathbf{a}_N^{*T} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{bmatrix}, \quad \text{(forw. transf.)}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{bmatrix} = \mathbf{A}\mathbf{y} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix} = \sum_{i=1}^{n} y_i\, \mathbf{a}_i \quad \text{(inv. trans.)}$$

When $\mathbf{A} = \mathbf{A}^*$ is real, $\mathbf{A}^{-1} = \mathbf{A}^T$, this is an orthogonal transform.

The first equation above is the **forward transform** and can be written as:

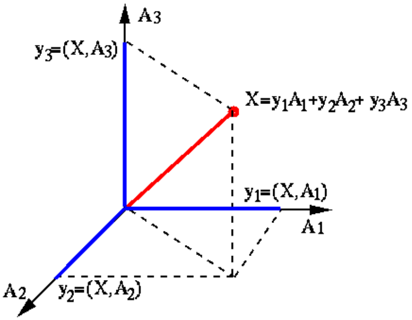
$$y_i = \mathbf{a}_i^{*T}\mathbf{x} = (\mathbf{a}_i, \mathbf{x}) = \sum_{j=1}^{N} a_{i,j}^{*}\mathbf{x}_j$$

The transform coefficient $y_i = (\mathbf{a}_i, \mathbf{x})$ (an inner product) represents the projection of vector $\mathbf{x}$ onto the ith column vector $\mathbf{a}_i$ of the transform matrix **A**.

The second equation is the **inverse transform**

$$x_j = \sum_{i=1}^{N} a_{j,i}\, y_i$$

x is a linear combination of the N column vectors $\mathbf{a}_i, \mathbf{a}_2, \cdots, \mathbf{a}_N$ of the matrix **A**. Geometrically, **x** is a point in the N-D space spanned by these N orthonormal basis vectors. Each coefficient y_i is the projection of **x** onto the corresponding basis $\mathbf{a}_i$.



A N-dimensional space can be spanned by the column vectors of *any* unitary matrix.

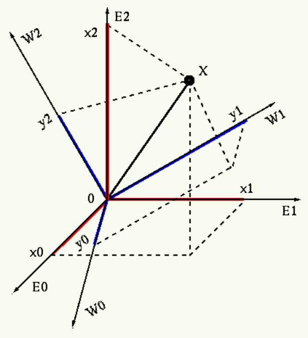
Examples:

- When $\mathbf{A} = \mathbf{I} = [\cdots, \mathbf{e}_i, \cdots]$ is an identity matrix, we have

$$\mathbf{x} = \sum_{i=1}^{N} y_i \mathbf{a}_i = \sum_{i=1}^{N} x_i \mathbf{e}_i$$

  where $\mathbf{e}_i = [0, \cdots, 0, 1, 0, \cdots, 0]^T$ is the ith column of $\mathbf{I}$ with the ith element equal 1 and all other 0.

- When $a_{m,n} = w[m,n] = e^{-j2\pi mn/N}$, we obtain the DFT. The nth column vector $\mathbf{w}_n$ of $\mathbf{W} = [\mathbf{w}_0, \cdots, \mathbf{w}_{N-1}]$ represents a sinusoid of a frequency nf_0, and the corresponding $y_n = (\mathbf{x}, \mathbf{w}_n)$ represents the magnitude $|y_n|$ and phase $\angle y_n$ of this nth frequency component. The Fourier transform $\mathbf{y} = \mathbf{W}\mathbf{x}$ represents a rotation of the coordinate system.



Geometrically, a unitary transform $\mathbf{y} = \mathbf{A}\mathbf{x}$ is a rotation of the vector X about the origin. It also does not change the vector's length:

$$|\mathbf{y}|^2 = \mathbf{y}^{*T}\mathbf{y} = (\mathbf{A}^{*T}\mathbf{x})^{*T}(\mathbf{A}^{*T}\mathbf{x}) = \mathbf{x}^{*T}\mathbf{A}\mathbf{A}^{*T}\mathbf{x} = \mathbf{x}^{*T}\mathbf{x} = |\mathbf{x}|^2$$

as $\mathbf{A}\mathbf{A}^{*T} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$.
Parseval's relation: the total energy of the signal is preserved under a unitary transform.

Some other features of the signal may be changed. If $\mathbf{x}$ is a random vector with mean $\mathbf{m}_x$ and covariance Σ_x:

$$\mathbf{m}_x = E(\mathbf{x}), \quad \Sigma_x = E(\mathbf{x}\mathbf{x}^T) - \mathbf{m}_x\mathbf{m}_x^T$$

then $\mathbf{y} = \mathbf{A}^T\mathbf{x}$ has the following

$$\mathbf{m}_y = E(\mathbf{y}) = E(\mathbf{A}^T\mathbf{x}) = \mathbf{A}^T E(\mathbf{x}) = \mathbf{A}^T\mathbf{m}_x$$

$$\begin{aligned}
\Sigma_y &= E(\mathbf{y}\mathbf{y}^T) - \mathbf{m}_y\mathbf{m}_y^T = E[(\mathbf{A}^T\mathbf{x})(\mathbf{A}^T\mathbf{x})^T] - (\mathbf{A}^T\mathbf{m}_x)(\mathbf{A}^T\mathbf{m}_x)^T \\
&= E[\mathbf{A}^T(\mathbf{x}\mathbf{x}^T)\mathbf{A}] - \mathbf{A}^T\mathbf{m}_x\mathbf{m}_x^T\mathbf{A} = \mathbf{A}^T[E(\mathbf{x}\mathbf{x}^T) - \mathbf{m}_x\mathbf{m}_x^T]\mathbf{A} \\
&= \mathbf{A}^T\Sigma_x\mathbf{A}
\end{aligned}$$

Eigenvalues and Eigenvectors

For any matrix **A**, if there exist a vector ϕ and a value λ such that

$$\mathbf{A}\phi = \lambda\phi$$

then λ and ϕ are called the *eigenvalue* and *eigenvector* of **A**. To obtain λ, rewrite the above equation as

$$(\lambda\mathbf{I} - \mathbf{A})\phi = 0$$

which is a homogeneous equation system. To find its non-zero solution for ϕ, we require

$$|\lambda\mathbf{I} - \mathbf{A}| = 0$$

Solving this *N*th order equation of λ, we get *n* eigenvalues $\{\lambda_1, \cdots, \lambda_N\}$.

Substituting each λ_i back into the equation system, we get the corresponding eigenvector $\phi_{\mathbf{i}}$.

$$\mathbf{A}[\phi_{\mathbf{1}}, \cdots, \phi_N] = [\lambda_1 \phi_1, \cdots, \lambda_N \phi_N]$$
$$= [\phi_1, \cdots, \phi_N] \begin{bmatrix} \lambda_1 & 0 & \cdot & 0 \\ 0 & \lambda_2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \lambda_N \end{bmatrix}$$

In a more compact form $\mathbf{A}\Phi = \Phi\Lambda$ or

$$\Phi^{-1}\mathbf{A}\Phi = \Lambda$$

where

$$\Phi = [\phi_1, \cdots, \phi_N]$$

and

$$\Lambda = diag[\lambda_1, \cdots, \lambda_N]$$

The trace and determinant of **A** can be obtained from its eigenvalues

$$tr(\mathbf{A}) = \sum_{k=1}^{N} \lambda_k$$

and

$$det(\mathbf{A}) = \prod_{k=1}^{N} \lambda_k$$

$\mathbf{A}^T$ has the same eigenvalues and eigenvectors as $\mathbf{A}$.
$\mathbf{A}^m$ has the same eigenvectors as $\mathbf{A}$, but its eigenvalues are $\{\lambda_1^m, \cdots, \lambda_n^m\}$, where m is a positive integer.
This is also true for $m = -1$, i.e., the eigenvalues of $\mathbf{A}^{-1}$ are $\{1/\lambda_1, \cdots, 1/\lambda_N\}$.
If $\mathbf{A}$ is Hermitian (symmetric if $\mathbf{A}$ is real), all the λ_i's are real and all eigenvectors ϕ_i's are orthogonal:

$$(\phi_i, \phi_j) = \delta_{ij}$$

If all ϕ_i's are normalized, matrix Φ is unitary (orthogonal if **A** is real):

$$\Phi^{-1} = \Phi^{*T}$$

and we have

$$\Phi^{-1}\mathbf{A}\Phi = \Phi^{*T}\mathbf{A}\Phi = \Lambda$$

The matrix **A** can be decomposed to be expressed as

$$\mathbf{A} = \Phi\Lambda\Phi^T = [\phi_1, \cdots, \phi_N] \begin{bmatrix} \lambda_1 & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \lambda_N \end{bmatrix} \begin{bmatrix} \phi_1^T \\ \ldots \\ \phi_N^T \end{bmatrix} = \sum_{i=1}^{N} \lambda_i \phi_i \phi_i^T$$

Hadamard Matrix

The *Kronecker product* of two matrices $\mathbf{A} = [a_{ij}]_{m \times n}$ and $\mathbf{B} = [b_{ij}]_{k \times l}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} \triangleq \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \cdots & \cdots & \cdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}_{mk \times nl}$$

In general, $\mathbf{A} \otimes \mathbf{B} \neq \mathbf{B} \otimes \mathbf{A}$.

The *Hadamard Matrix* is defined recursively as below:

$$\mathbf{H}_1 \triangleq \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

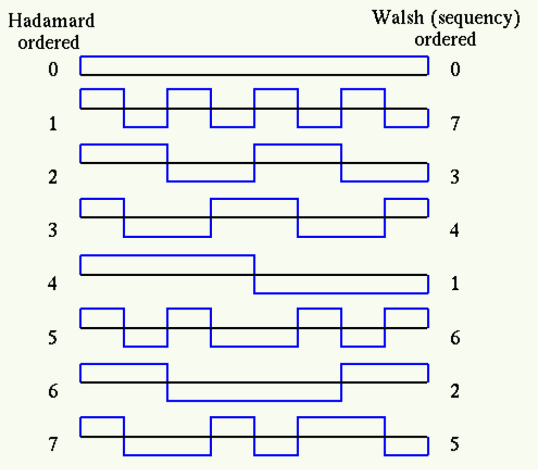
$$\mathbf{H}_n = \mathbf{H}_1 \otimes \mathbf{H}_{n-1} = \begin{bmatrix} \mathbf{H}_{n-1} & \mathbf{H}_{n-1} \\ \mathbf{H}_{n-1} & -\mathbf{H}_{n-1} \end{bmatrix}$$

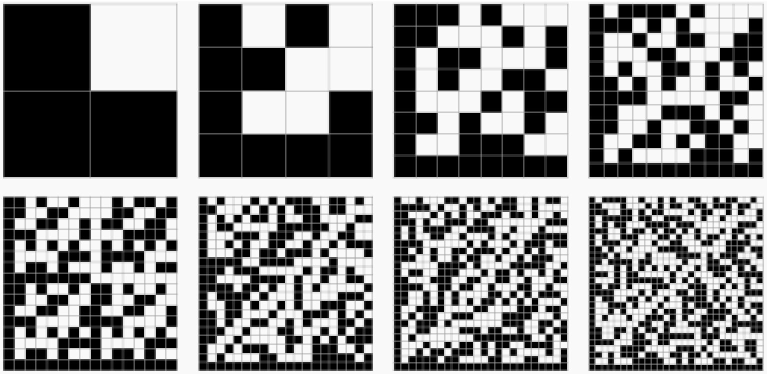
For example,

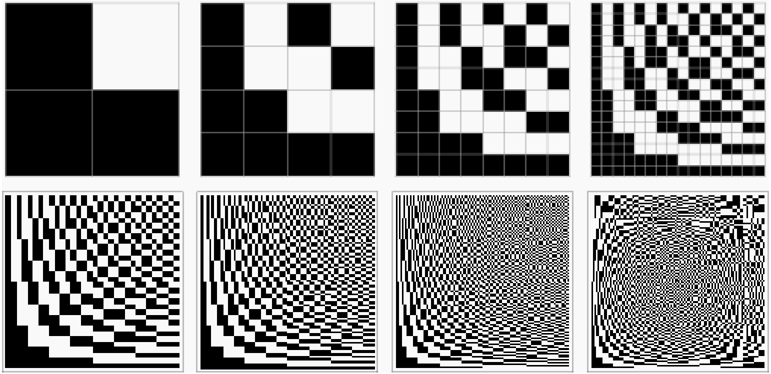
$$\mathbf{H}_2 = \mathbf{H}_1 \otimes \mathbf{H}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_1 \\ \mathbf{H}_1 & -\mathbf{H}_1 \end{bmatrix} = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

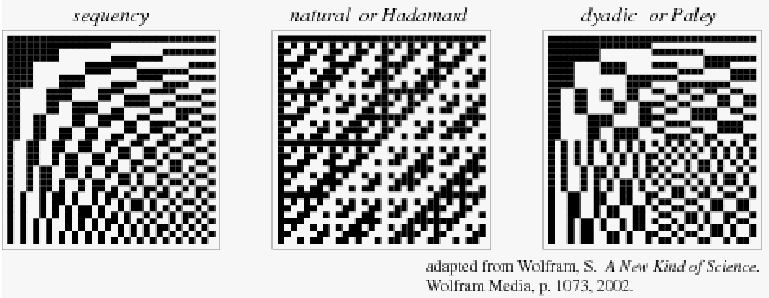
$$\mathbf{H}_3 = \mathbf{H}_1 \otimes \mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}$$

The first column following the array is the index numbers of the $N = 8$ rows, and the second column represents the *sequency* (the number of zero-crossings or sign changes) in each row.


Hadamard
ordered
Walsh (sequency)
ordered
0
0
1
7
2
3
3
4
4
1
5
6
6
2
7
5






sequency
natural or Hadamard
dyadic or Paley
adapted from Wolfram, S. A New Kind of Science.
Wolfram Media, p. 1073, 2002.

The Hadamard matrix can also be obtained by defining its element in the kth row and mth column of H as

$$h[k,m] = (-1)^{\sum_{i=0}^{n-1} k_i m_i} = \prod_{i=0}^{n-1} (-1)^{k_i m_i} = h[m,k] \quad (k,m = 0,1,\cdots,N-1)$$

where

$$k = \sum_{i=0}^{n-1} k_i 2^i = (k_{n-1} k_{n-2} \cdots k_1 k_0)_2 \quad (k_i = 0,1)$$

$$m = \sum_{i=0}^{n-1} m_i 2^i = (m_{n-1} m_{n-2} \cdots m_1 m_0)_2 \quad (m_i = 0,1)$$

i.e., $(k_{n-1} k_{n-2} \cdots k_1 k_0)_2$ and $(m_{n-1} m_{n-2} \cdots m_1 m_0)_2$ are the binary representations of k and m, respectively. Obviously, $n = log_2 N$.

H is real, symmetric, and orthogonal:

$$\mathbf{H} = \mathbf{H}^* = \mathbf{H}^T = \mathbf{H}^{-1}$$

It defines the transform pair:

$$\mathbf{X} = \mathbf{H}\mathbf{x}, \quad \mathbf{x} = \mathbf{H}\mathbf{X}$$

where the forward and inverse transforms are identical.

Fast Walsh-Hadamard Transform (Hadamard Ordered)

Since any orthogonal matrix defines a transform, the Walsh-Hadamard transform pair is

$$\begin{cases} \mathbf{X} = \mathbf{H}\mathbf{x} \\ \mathbf{x} = \mathbf{H}\mathbf{X} \end{cases}$$

where $\mathbf{x} = [x[0], x[1], \cdots, x[N-1]]^T$ and $\mathbf{X} = [X[0], X[1], \cdots, X[N-1]]^T$ are the signal and spectrum vectors. The kth element of the transform is

$$X[k] = \sum_{m=0}^{N-1} h[k,m]x[m] = \sum_{m=0}^{N-1} x[m] \prod_{i=0}^{n-1} (-1)^{m_i k_i}$$

The complexity of WHT is $O(N^2)$. Similar to FFT algorithm, we can derive a fast WHT algorithm with complexity of $O(Nlog_2 N)$.

Assume $n = 3$ and $N = 2^n = 8$. An $N = 8$ point WHT_h of the signal $x[m]$ is

$$\begin{bmatrix} X[0] \\ . \\ X[3] \\ X[4] \\ . \\ X[7] \end{bmatrix} = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix} \begin{bmatrix} x[0] \\ . \\ x[3] \\ x[4] \\ . \\ x[7] \end{bmatrix}$$

This equation can be separated into two parts. The first half of the X vector is

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \mathbf{H}_2 \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} + \mathbf{H}_2 \begin{bmatrix} x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix} = \mathbf{H}_2 \begin{bmatrix} x_1[0] \\ x_1[1] \\ x_1[2] \\ x_1[3] \end{bmatrix} \tag{1}$$

where

$$x_1[i] \stackrel{\triangle}{=} x[i] + x[i+4] \quad (i = 0, \cdots, 3) \tag{2}$$

The second half of the X is

$$\begin{bmatrix} X[4] \\ X[5] \\ X[6] \\ X[7] \end{bmatrix} = \mathbf{H}_2 \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} - \mathbf{H}_2 \begin{bmatrix} x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix} = \mathbf{H}_2 \begin{bmatrix} x_1[4] \\ x_1[5] \\ x_1[6] \\ x_1[7] \end{bmatrix} \tag{3}$$

where

$$x_1[i+4] \triangleq x[i] - x[i+4] \quad (i = 0, \cdots, 3) \tag{4}$$

What we have done is converting a *WHT* of size $N = 8$ into two *WHTs* of size $N/2 = 4$. Continuing this process recursively, we can rewrite Eq. (1) as the following

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_1 \\ \mathbf{H}_1 & -\mathbf{H}_1 \end{bmatrix} \begin{bmatrix} x_1[0] \\ x_1[1] \\ x_1[2] \\ x_1[3] \end{bmatrix}$$

This equation can again be separated into two halves. The first half is

$$\begin{bmatrix} X[0] \\ X[1] \end{bmatrix} = \mathbf{H}_1 \begin{bmatrix} x_1[0] \\ x_1[1] \end{bmatrix} + \mathbf{H}_1 \begin{bmatrix} x_1[2] \\ x_1[3] \end{bmatrix} \tag{5}$$

$$= \mathbf{H}_1 \begin{bmatrix} x_2[0] \\ x_2[1] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_2[0] \\ x_2[1] \end{bmatrix} = \begin{bmatrix} x_2[0] + x_2[1] \\ x_2[0] - x_2[1] \end{bmatrix} \tag{6}$$

where

$$x_2[i] \stackrel{\triangle}{=} x_1[i] + x_1[i+2] \quad (i = 0, 1) \tag{7}$$

The second half is

$$\begin{bmatrix} X[2] \\ X[3] \end{bmatrix} = \mathbf{H}_1 \begin{bmatrix} x_1[0] \\ x_1[1] \end{bmatrix} - \mathbf{H}_1 \begin{bmatrix} x_1[2] \\ x_1[3] \end{bmatrix} \tag{8}$$

$$= \mathbf{H}_1 \begin{bmatrix} x_2[2] \\ x_2[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_2[2] \\ x_2[3] \end{bmatrix} = \begin{bmatrix} x_2[2] + x_2[3] \\ x_2[2] - x_2[3] \end{bmatrix} \tag{9}$$

where

$$x_2[i+2] \triangleq x_1[i] - x_1[i+2] \quad (i = 0, 1) \tag{10}$$

$X[4]$ through $X[7]$ of the second half can be obtained similarly.

$$X[0] = x_2[0] + x_2[1] \quad (11)$$

and

$$X[1] = x_2[0] - x_2[1] \quad (12)$$

Summarizing the above steps of Equations we get the Fast WHT algorithm as illustrated below.